**TRƯỜNG ĐẠI HỌC CNTT VÀ TT VIỆT – HÀN**

**KHOA KHOA HỌC MÁY TÍNH**



**ĐỒ ÁN CHUYÊN NGÀNH 1**

**ĐỀ TÀI: ỨNG DỤNG CYPRESS VÀ CUCUMBER TRONG KIỂM THỬ TỰ ĐỘNG**

Sinh viên thực hiện: **NGUYỄN TRUNG HIẾU – 19IT3**

**TRẦN THỊ KIM OANH – 19IT3**

Giảng viên hướng dẫn: **THS. NGUYỄN NGỌC HUYỀN TRÂN**

***Đà Nẵng, tháng 11 năm 2022***

**TRƯỜNG ĐẠI HỌC CNTT VÀ TT VIỆT – HÀN**

**KHOA KHOA HỌC MÁY TÍNH**



**ĐỒ ÁN CHUYÊN NGÀNH 1**

**ĐỀ TÀI: ỨNG DỤNG CYPRESS VÀ CUCUMBER TRONG KIỂM THỬ TỰ ĐỘNG**

Sinh viên thực hiện: **NGUYỄN TRUNG HIẾU – 19IT3**

**TRẦN THỊ KIM OANH – 19IT3**

Giảng viên hướng dẫn: **THS. NGUYỄN NGỌC HUYỀN TRÂN**

***Đà Nẵng, tháng 11 năm 2022***

# LỜI CẢM ƠN

Trong suốt thời gian làm đồ án, nhóm chúng em nhận được rất nhiều sự giúp đỡ. Chúng em xin chân thành cảm ơn Khoa khoa học máy tính - Trường ĐH CNTT & TT Việt Hàn - ĐHĐN đã tạo điều kiện thuận lợi cho em thực hiện đề tài đồ án chuyên ngành 1 này. Xin cảm ơn quý Thầy Cô bộ môn của Khoa khoa học máy tính - Trường ĐH CNTT & TT Việt Hàn - ĐHĐN đã tận tình giảng dạy, trang bị cho chúng em những kiến thức quý báu trong năm học vừa qua.

Chúng em xin chân thành tỏ lòng biết ơn sâu sắc đến Cô Nguyễn Ngọc Huyền Trân đã tận tình hướng dẫn em trong suốt quá trình thực hiện làm đồ án này.

Cảm ơn các bạn trong lớp, trong trường, trong Khoa khoa học máy tính - Trường ĐH CNTT & TT Việt Hàn - ĐHĐN đã hỗ trợ, giúp đỡ, chia sẻ kinh nghiệm và kiến thức cho nhóm trong quá trình thực hiện đề tài.

Trong quá trình làm đồ án, do kiến thức về kiểm thử tự động chưa thực sự sâu sắc nên đồ án của chúng em không thể tránh khỏi những sai sót. Mong thầy cô thông cảm và góp ý thêm cho chúng em để đồ án được hoàn chỉnh hơn.

Chúng em xin chân thành cảm ơn!

# LỜI NÓI ĐẦU

Cùng với sự phát triển không ngừng của các ngành khoa học kỹ thuật, các ngành công nghiệp cũng phát triển nhanh chóng. Việc áp dụng những công nghệ hiện đại vào nhiều lĩnh vực trong cuộc sống đó là điều tất yếu. Song song với sự phát triển đó là sự phát triển về văn hóa, về các phương thức truyền thông mạng. Sự ra đời của công nghệ tự động hóa, đem đến lợi ích và tiện nghi cho xã hội. Chính vì thế việc áp dụng tự động hóa trong kiểm thử phần mềm đó là điều tất yếu. Nó mang đến nhiều lợi ích mà kiểm thử thủ công không thể có được.

Do đó, nhóm chúng em quyết định thực hiện đề tài: “**ỨNG DỤNG CYPRESS VÀ CUCUMBER TRONG KIỂM THỬ TỰ ĐỘNG**”. Qua đề tài này nhóm chúng em muốn lan tỏa cũng như truyền đạt đến mọi người những công nghệ mới nhất hiện nay trong kiểm thử phần mềm tự động. Với những kiến thức tự học, cũng như được học qua học phần kiểm thử phần mềm nhóm chúng em tin chắc rằng đây là một đề tài khá mới mẻ đối với các bạn sinh viên.

Bằng sự cố gắng nỗ lực và đặc biệt là sự giúp đỡ tận tình, chu đáo của giảng viên hướng dẫn ThS. Nguyễn Ngọc Huyền Trân, nhóm chúng em đã hoàn thành đồ án chuyên ngành 1 đúng thời hạn. Do thời gian làm đồ án có hạn và trình độ còn nhiều hạn chế nên không thể tránh khỏi những thiếu sót. Em rất mong nhận được sự đóng góp ý kiến của các thầy cô cũng như là của các bạn sinh viên để bài đồ án này hoàn thiện hơn nữa.

*Đà Nẵng, ngày 10 tháng 11 năm 2022*

# NHẬN XÉT

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

*Đà Nẵng, ngày 20 tháng 11 năm 2022*

Chữ ký của giảng viên hướng dẫn

# MỤC LỤC

[LỜI CẢM ƠN 3](#_Toc124436730)

[LỜI NÓI ĐẦU 4](#_Toc124436731)

[NHẬN XÉT 5](#_Toc124436732)

[MỤC LỤC 6](#_Toc124436733)

[DANH MỤC HÌNH 9](#_Toc124436734)

[DANH MỤC VIẾT TẮT 11](#_Toc124436735)

[MỞ ĐẦU 1](#_Toc124436736)

[Giới thiệu đề tài 1](#_Toc124436737)

[Kế hoạch dự kiến 2](#_Toc124436738)

[Mục tiêu nghiên cứu 3](#_Toc124436739)

[Nhiệm vụ nghiên cứu 3](#_Toc124436740)

[Phương pháp và phạm vi nghiên cứu 3](#_Toc124436741)

[Phương tiện nghiên cứu 3](#_Toc124436742)

[Kết quả đề tài 4](#_Toc124436743)

[Bố cục báo cáo 4](#_Toc124436744)

[CHƯƠNG 1. TỔNG QUAN 5](#_Toc124436745)

[1.1. Ngôn ngữ, công cụ lập trình Website 5](#_Toc124436746)

[1.1.1. Front-End 5](#_Toc124436747)

[1.1.1.1. HTML, CSS, Bootrap 5](#_Toc124436748)

[1.1.2. Back-End 8](#_Toc124436749)

[1.1.2.1. Ngôn ngữ lập trình PHP 8](#_Toc124436750)

[1.1.2.2. Cơ Sở Dữ Liệu XAMPP 10](#_Toc124436751)

[1.1.2.3. Công cụ lập trình Visual Studio Code 11](#_Toc124436752)

[1.2. Công cụ kiểm thử phần mềm 12](#_Toc124436753)

[1.2.1. Công cụ Cypress 12](#_Toc124436754)

[1.2.1.1. Khái Niệm 12](#_Toc124436755)

[1.2.1.2. Các tính năng của Cypress 12](#_Toc124436756)

[1.2.1.3. Ưu điểm nhược điểm của Cypress 13](#_Toc124436757)

[1.2.1.3.1. Ưu điểm 13](#_Toc124436758)

[1.2.1.3.2. Nhược điểm 13](#_Toc124436759)

[1.2.2. Công cụ Cucumber 14](#_Toc124436760)

[1.2.2.1. Khái niệm 14](#_Toc124436761)

[1.2.2.2. Cách hoạt động của Cucumber 15](#_Toc124436762)

[1.2.2.3. So sánh Cucumber với các công cụ khác 16](#_Toc124436763)

[1.2.3. Docker 17](#_Toc124436764)

[1.2.3.1. Khái niệm 17](#_Toc124436765)

[1.2.3.2. Sự ra đời của Docker 17](#_Toc124436766)

[1.2.3.3. Quy trình thực thi của một hệ thống Docker 19](#_Toc124436767)

[1.2.3.4. Một số khái niệm liên quan đến Docker 20](#_Toc124436768)

[CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG WEBSITE 23](#_Toc124436769)

[2.1. Biểu đồ Usecase 23](#_Toc124436770)

[2.1.1. Nhận diện tác nhân và Usecase 23](#_Toc124436771)

[2.1.2. Đặc tả Usecase 24](#_Toc124436772)

[2.1.2.1. Use Case Login 24](#_Toc124436773)

[2.1.2.2. Use Case Order Food 25](#_Toc124436774)

[2.2. Biểu đồ trình tự 25](#_Toc124436775)

[2.2.1. Biểu đồ trình tự đăng nhập 25](#_Toc124436776)

[2.2.2. Biểu đồ trình tự đăng ký 26](#_Toc124436777)

[2.2.3. Biểu đồ trình tự người dùng thực hiện chức năng 26](#_Toc124436778)

[2.2.4. Biểu đồ trình tự admin thực hiện chức năng 27](#_Toc124436779)

[2.3. Cấu trúc cơ sở dữ liệu 28](#_Toc124436780)

[CHƯƠNG 3. XÂY DỰNG WEBSITE QUICK FOOD VÀ KIỂM THỬ TỰ ĐỘNG CHO QUICK FOOD 29](#_Toc124436781)

[3.1. Về website Quick Food 29](#_Toc124436782)

[3.1.1. Giao diện người dùng 29](#_Toc124436783)

[3.1.2. Giao diện Admin 32](#_Toc124436784)

[3.2. Kiểm thử tự động cho Quick Food 36](#_Toc124436785)

[3.2.1. Sử dụng Cypress và Cucumber trong dự án 36](#_Toc124436786)

[3.2.2. Sử dụng Page Object Model (POM) 40](#_Toc124436787)

[3.2.3. Multiple Cucumber HTML Reporter 44](#_Toc124436788)

[3.2.4. Test UI bằng Percy io 45](#_Toc124436789)

[3.2.5. Kết quả của kiểm thử tự động 46](#_Toc124436790)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 47](#_Toc124436791)

[Kết quả đạt được 47](#_Toc124436792)

[Hạn chế 47](#_Toc124436793)

[Hướng phát triển 48](#_Toc124436794)

[TÀI LIỆU THAM KHẢO 49](#_Toc124436795)

# DANH MỤC HÌNH

[Hình 1. 1. HTML (HyperText Markup Language) 6](#_Toc124435954)

[Hình 1. 2. CSS (Cascading Style Sheets) 7](#_Toc124435955)

[Hình 1. 3. Bootstrap 8](#_Toc124435956)

[Hình 1. 4. Ngôn ngữ lập trình PHP 9](#_Toc124435957)

[Hình 1. 5. Cơ sở dữ liệu phpMyAdmin Xampp 11](#_Toc124435958)

[Hình 1. 6. Công cụ lập trình Visual Studio Code 12](#_Toc124435959)

[Hình 1. 7. Công cụ Cucumber 15](#_Toc124435960)

[Hình 1. 8. Mô tả cách thức hoạt động của Cucumber 16](#_Toc124435961)

[Hình 1. 9. So sánh Cucumber với các công cụ khác 17](#_Toc124435962)

[Hình 1. 10. Hình ảnh Docker 18](#_Toc124435963)

[Hình 1. 11. Sự ra đời cửa Docker 19](#_Toc124435964)

[Hình 1. 12. Mô tả cách thức hoạt động của Docker 20](#_Toc124435965)

[Hình 1. 13. Quá trình hoạt động của Docker 22](#_Toc124435966)

[Hình 1. 14. Cách hoạt động của Docker Volume 23](#_Toc124435967)

[Hình 1. 15. Cách hoạt động của Docker Networking 23](#_Toc124435968)

[Hình 2. 1. Biểu đồ Usecase 25](#_Toc124435541)

[Hình 2. 2. Biểu đồ trình tự đăng nhập 26](#_Toc124435542)

[Hình 2. 3. Biểu đồ trình tự đăng ký 27](#_Toc124435543)

[Hình 2. 4. Biểu đồ trình tự người dùng thực hiện các chức năng 28](#_Toc124435544)

[Hình 2. 5. Biểu đồ trình tự Admin thực hiện các chức năng 28](#_Toc124435545)

[Hình 2. 6. Cấu trúc cơ sở dữ liệu 29](#_Toc124435546)

[Hình 3. 1. Giao diện form đăng nhập 30](#_Toc124435547)

[Hình 3. 2. Giao diện form đăng kí 31](#_Toc124435548)

[Hình 3. 3. Giao diện trang chủ 31](#_Toc124435549)

[Hình 3. 4. Danh mục các món ăn trên trang chủ 32](#_Toc124435550)

[Hình 3. 5. Danh mục nhà hàng trên trang chủ 32](#_Toc124435551)

[Hình 3. 6. Danh mục các nhà hàng trên trang Restaurants 32](#_Toc124435552)

[Hình 3. 7. Danh mục các đơn hàng đã đặt tại trang Your Orders 33](#_Toc124435553)

[Hình 3. 8. Trang dashboard 33](#_Toc124435554)

[Hình 3. 9. Xem tất cả tài khoản của những người dùng đã đăng kí thành công 33](#_Toc124435555)

[Hình 3. 10. Thêm tài khoản cho người dùng mới 34](#_Toc124435556)

[Hình 3. 11. Xem tất cả thông tin của những cửa hàng trên website 34](#_Toc124435557)

[Hình 3. 12. Xuất danh sách những cửa hàng trên website 34](#_Toc124435558)

[Hình 3. 13. Giao diện thêm danh mục món ăn 35](#_Toc124435559)

[Hình 3. 14. Giao diện thêm nhà hàng 35](#_Toc124435560)

[Hình 3. 15. Danh mục Menu 36](#_Toc124435561)

[Hình 3. 16. Giao diện thêm món ăn mới 36](#_Toc124435562)

[Hình 3. 17. Xem danh mục các món ăn đã đặt 36](#_Toc124435563)

[Hình 3. 18. Form thay đổi trạng thái của đơn hàng 37](#_Toc124435564)

[Hình 3. 19. Giao diện cypress trong dự án 47](#_Toc124435565)

[Hình 3. 20. Test 2e2 tính năng chính người dùng sử dụng trên website 47](#_Toc124435566)

[Hình 3. 21. Test Order Page 47](#_Toc124435567)

# DANH MỤC VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **STT** | **KÝ HIỆU CHỮ VIẾT TẮT** | **CHỮ VIẾT ĐẦY ĐỦ** |
| 1 | HTML | Hyper Text Markup Language |
| 2 | UML | Unified Modeling Language  (Ngôn ngữ mô hình hóa thống nhất) |
| 3 | UC | UseCase |
| 4 | API | Application Programming Interface  (Giao diện lập trình ứng dụng) |

# MỞ ĐẦU

Cùng với sự phát triển không ngừng của các ngành khoa học kỹ thuật, các ngành công nghiệp cũng phát triển nhanh chóng. Việc áp dụng những công nghệ hiện đại vào nhiều lĩnh vực trong cuộc sống đó là điều tất yếu. Song song với sự phát triển đó là sự phát triển về văn hóa, về các phương thức truyền thông mạng. Sự ra đời của công nghệ tự động hóa, đem đến lợi ích và tiện nghi cho xã hội. Chính vì thế việc áp dụng tự động hóa trong kiểm thử phần mềm đó là điều tất yếu. Nó mang đến nhiều lợi ích mà kiểm thử thủ công không thể có được.

Do đó, nhóm chúng em quyết định thực hiện đề tài: “**ỨNG DỤNG CYPRESS VÀ CUCUMBER TRONG KIỂM THỬ TỰ ĐỘNG**”. Qua đề tài này nhóm chúng em muốn lan tỏa cũng như truyền đạt đến mọi người những công nghệ mới nhất hiện nay trong kiểm thử phần mềm tự động. Với những kiến thức tự học, cũng như được học qua học phần kiểm thử phần mềm nhóm chúng em tin chắc rằng đây là một đề tài khá mới mẻ đối với các bạn sinh viên.

## Giới thiệu đề tài

Hiện nay, công nghệ thông tin được xem là một ngành mũi nhọn của các quốc gia, đặc biệt là các quốc gia đang phát triển, tiến hành công nghiệp hóa và hiện đại hoá như nước ta. Sự bùng nổ thông tin và sự phát triển mạnh mẽ của công nghệ kỹ thuật số, yêu cầu muốn phát triển thì phải tin học hoá, tự động hóa vào tất cả các ngành các lĩnh vực.

Cùng với sự phát triển nhanh chóng về phần cứng máy tính, các ứng dụng ngày càng trở nên đa dạng, phong phú, hoàn thiện hơn và hỗ trợ hiệu quả cho con người. Các ứng dụng hiện nay ngày càng mô phỏng được rất nhiều nghiệp vụ khó khăn, hỗ trợ cho người dùng thuận tiện sử dụng, thời gian xử lý nhanh chóng, và một số nghiệp vụ được tự động hoá cao.

Do vậy mà trong việc phát triển ứng dụng, sự đòi hỏi không chỉ là sự chính xác, xử lý được nhiều nghiệp vụ thực tế mà còn phải đáp ứng các yêu cầu khác như về tốc độ, giao diện thân thiện, mô hình hoá được thực tế vào máy tính để người sử dụng tiện lợi, quen thuộc, tính tương thích cao, bảo mật cao (đối với các dữ liệu nhạy cảm), … Chính vì thế trong quy trình sản xuất một phần mềm. Áp dụng kiến thức đã học để phát triển các nền tảng như trên là cần thiết. Bởi vì điều này cho phép học thêm các công nghệ mới, rèn luyện kỹ năng lập trình, đồng thời nâng cao kinh nghiệm và học thêm các kỹ năng quan trọng khác. Thực hiện kiểm thử tự động bằng Cypress và Cucumber nó không phải là mới lạ gì trong giới phần mềm, nhưng trong trường rất ít sinh viên lựa chọn đề tài này. Qua đề tài này nhóm chúng em muốn nhấn mạnh sự quan trọng của học phần kiểm thử phần mềm, cũng như kiểm thử nói chung và kiểm thử tự động nói riêng.

## Kế hoạch dự kiến

|  |  |
| --- | --- |
| Thời gian | Nội dung thực hiện |
| Tuần thứ 1 | Tìm hiểu xây dựng demo hệ thống và cách thức hoạt động của chương trình mà nhóm hướng đến.  Báo cáo tiến độ đề tài |
| Tuần thứ 2 | Nghiên cứu tiếp cận các công nghệ và công cụ của những dự án kiểm thử đã thành công. Từ đó đưa ra ý tưởng về cách thức cũng như những công nghệ chính phù hợp với đề tài.  Xây dựng và thiết kế website  Báo cáo tiến độ đề tài |
| Tuần thứ 3 | Phân tích, lên kế hoạch chi tiết cho từng phân đoạn cần kiểm thử  Xây dựng và thiết kế website  Báo cáo tiến độ đề tài |
| Tuần thứ 4 | Xây dựng và thiết kế website  Báo cáo tiến độ đề tài |
| Tuần thứ 5 - 9 | Viết bộ Test Case và Test Scenario cho dự án  Từ bộ Test Case và Test Scenario tiến hành viết chương trình kiểm thử tự động cho dự án  Thực thi kiểm tra lại chương trình trong quá trình làm nhằm đam bảo đến lúc gần bảo vệ không xảy ra lỗi hẹ thống.  Tiến hành viết báo cáo đồ án.  Báo cáo tiến độ đề tài |
| Các tuần còn lại | Hoàn thiện website Quick Food  Hoàn thiện báo cáo và slide báo cáo  Nôp báo cáo và sản phẩm hoàn thiện lên hệ thống đào tạo |

## Mục tiêu nghiên cứu

Thực hiện một đề tài đồ án chuyên ngành như một dự án nhỏ bao gồm các phần quan trọng như: lên kế hoạch, đặt ra yêu cầu và mục tiêu, tiến hành triển khai dự án- thiết kế front end and back end, sau đó là kiểm thử thủ công và kiểm thử động.

- Dự án sẽ mô phỏng lại tất cả các thao tác có thể thực hiện trên website

- Giúp cho các bạn đam mê mảng kiểm thử có thể biết đến những công cụ cũng như những công nghệ mới

- Giúp bản thân có thêm khả năng sáng tạo, tư duy thông qua thiết kế giao diện tương tác với người dùng, kiểm thử và vận dụng kiến thức bản thân vào thực tiễn.

## Nhiệm vụ nghiên cứu

Đề tài này có 2 nhiệm vụ chính:

* Về website Quick Food: Là một hệ thống đặt đồ ăn nhanh, tại đây người dung có thể đặt đồ ăn tại các nhà hang cũng như các quán ăn, hiển thị các đơn hang mà người dung đã đặt. Hệ thống quản lý đơn hang dành cho người quản lý.
* Về kiểm thử thủ công và kiểm thử động: Sử dụng bộ test case và test scenario tự viết sau đó sử dụng công cụ cypress, cucumber và ngôn ngữ Gherkin, node js. Ứng dụng những công cụ và ngôn ngữ kể trên vào kiểm thử tự động

## Phương pháp và phạm vi nghiên cứu

**Phương pháp nghiên cứu:**

- Tham khảo, nghiên cứu thêm các tài liệu trên trang web, trang báo, tìm hiểu tình hình thực trạng của đề tài.

- Tìm hiểu các đề tài của các anh chị khóa trước đã làm hoặc tìm hiểu qua sách báo, trang mạng liên quan đến đề tài nghiên cứu.

- Quan sát việc kiểm thử tự động cần những gì và cần làm gì từ đó đưa ra hướng test cụ thể và hợp lý nhất

**Phạm vi nghiên cứu:** Xoay quanh việc kiểm thử từ động là chính, bên cạnh đó còn liên hệ đến việc xây dựng website.

## Phương tiện nghiên cứu

- Nghiên cứu cơ sở lý thuyết về phân tích và thiết kế hệ thống thông tin.

- Sử dụng PHP, javascript, cypress, cucumber để xây dựng website và thực hiện kiểm thử tự động.

- Sử dụng các công cụ Visual studio code, Sublime Text 3… để lập trình, thiết kế giao diện và hoàn thiện chức năng.

## Kết quả đề tài

Sử dụng thành thạo các ngôn ngữ, cũng như các testing framework. Từ đó tích lũy thêm nhiều kinh nghiệm và kĩ năng xây dựng nên nhiều sản phẩm kiểm thử có tính thực tiễn cao.

Sử dụng ngôn ngữ PHP để xây dựng website Quick Food

Sau khi kết thúc một dự án thứ không thể thiếu chính là báo cáo viết về đề tài của mình và slide trình bày dự án. Báo cáo chi tiết trình bày bố cục mục lục hợp lý.

## Bố cục báo cáo

Mở đầu

Chương 1. Tổng quan

Chương 2. Phân tích và thiết kế hệ thống

Chương 3. Xây dựng website Quick Food và kiểm thử tự động cho Quick Food

Kết luận và hướng phát triển.

# CHƯƠNG 1. TỔNG QUAN

## 1.1. Ngôn ngữ, công cụ lập trình Website

### 1.1.1. Front-End

#### 1.1.1.1. HTML, CSS, Bootrap

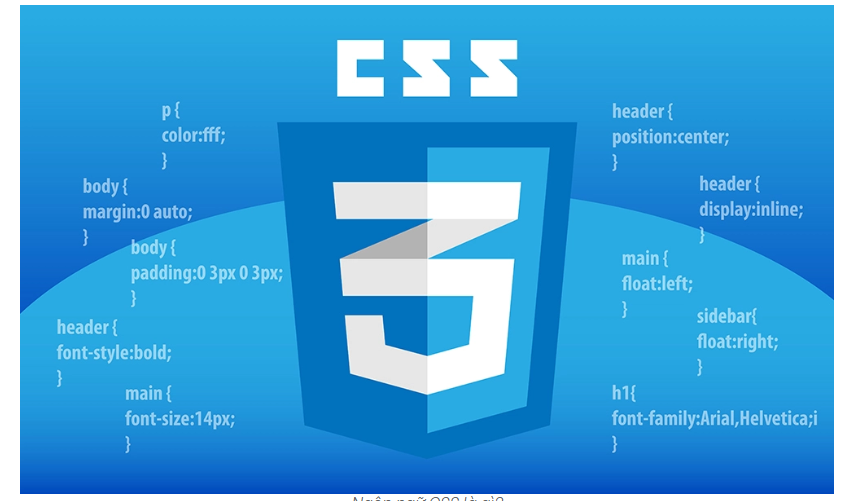
**HTML (HyperText Markup Language)**: là một ngôn ngữ đánh dấu được thiết kế ra để tạo nên các trang web, nghĩa là các mẩu thông tin được trình bày trên World Wide Web



Hình 1. 1. HTML (HyperText Markup Language)

**CSS (Cascading Style Sheets)**: là một ngôn ngữ được sử dụng để **tìm và định dạng** lại các phần tử được tạo ra bởi các ngôn ngữ đánh dấu ([HTML](https://topdev.vn/blog/html-la-gi/)). Nói ngắn gọn hơn là ngôn ngữ tạo phong cách cho trang web. Bạn có thể hiểu đơn giản rằng, nếu HTML đóng vai trò định dạng các phần tử trên website như việc tạo ra các đoạn văn bản, các tiêu đề, bảng,… thì CSS sẽ giúp chúng ta có thể thêm style vào các phần tử HTML đó như đổi bố cục, màu sắc trang, đổi màu chữ, font chữ, thay đổi cấu trúc…

CSS được phát triển bởi **W3C** ([World Wide Web Consortium](https://www.w3.org/)) vào năm 1996, vì HTML không được thiết kế để gắn tag để giúp định dạng trang web.Phương thức hoạt động của CSS là nó sẽ tìm dựa vào các vùng chọn, vùng chọn có thể là tên một thẻ HTML, tên một ID, class hay nhiều kiểu khác. Sau đó là nó sẽ áp dụng các thuộc tính cần thay đổi lên vùng chọn đó. Mối tương quan giữa HTML và CSS rất mật thiết. HTML là ngôn ngữ markup (nền tảng của site) và CSS định hình phong cách (tất cả những gì tạo nên giao diện website), chúng là không thể tách rời.



Hình 1. 2. **CSS** (Cascading Style Sheets)

CSS được dùng để định nghĩa kiểu cách cho các trang trên website của bạn, gồm cả thiết kế, dàn trang và các cách hiển thị khác nhau trên nhiều thiết bị với kích thước màn hình khác nhau.

*CSS giúp giải quyết vấn đề lớn của HTML*

HTML không có phần tử để để định dạng cho trang web. HTML chỉ được dùng để tạo nội dung cho trang. Khi các phần tử như <font> và thuộc tính màu sắc được thêm vào HTML 3.2, cơn ác mộng của các nhà phát triển web bắt đầu. Việc phát triển một website lớn mà thêm thông tin font hay màu sắc vào từng trang đòi hỏi rất nhiều thời gian.

Để giải quyết vấn đề này, World Wide Web Consortium (W3C) đã tạo ra CSS, giúp loại bỏ việc định dạng kiểu cách khỏi trang HTML.Các định nghĩa liên quan đến kiểu cách được đưa vào tập tin .css và nhờ vào tập tin stylesheet ngoài, bạn có thể thay đổi toàn bộ website chỉ bằng một tập tin duy nhất.

**Bootstrap là 1 framework HTML, CSS, và JavaScript** cho phép người dùng dễ dàng thiết kế website theo 1 chuẩn nhất định, tạo các website thân thiện với các thiết bị cầm tay như mobile, ipad, tablet,... Bootstrap bao gồm những cái cơ bản có sẵn như: typography, forms, buttons, tables, navigation, modals, image carousels và nhiều thứ khác. Trong bootstrap có thêm nhiều Component, Javascript hỗ trợ cho việc thiết kế reponsive của bạn dễ dàng, thuận tiện và nhanh chóng hơn.



Hình 1. 3. Bootstrap

Bootstrap là một trong những framework được sử dụng nhiều nhất trên thế giới để xây dựng nên một website. Bootstrap đã xây dựng nên 1 chuẩn riêng và rất được người dùng ưa chuộng. Chính vì thế, chúng ta hay nghe tới một cụm từ rất thông dụng "Thiết kế theo chuẩn Bootstrap".Từ cái "chuẩn mực" này, chúng ta có thể thấy rõ được những điểm thuận lợi khi sử dụng Bootstrap.

- Rất dễ để sử dụng: Nó đơn giản vì nó được base trên HTML, CSS và Javascript chỉ cẩn có kiến thức cơ bản về 3 cái đó là có thể sử dụng bootstrap tốt.

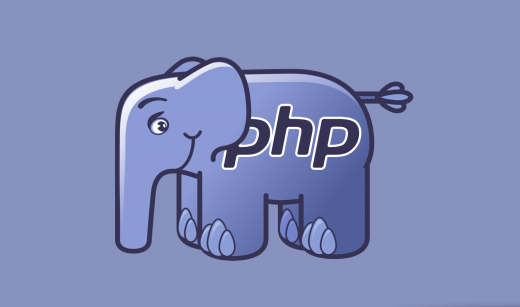
- Responsive: Bootstrap xây dựng sẵn reponsive css trên các thiết bị Iphones, tablets, và desktops. Tính năng này khiến cho người dùng tiết kiệm được rất nhiều thời gian trong việc tạo ra một website thân thiện với các thiết bị điện tử, thiết bị cầm tay.

- Tương thích với trình duyệt: Nó tương thích với tất cả các trình duyệt (Chrome, Firefox, Internet Explorer, Safari, and Opera). Tuy nhiên, với IE browser, Bootstrap chỉ hỗ trợ từ IE9 trở lên. Điều này vô cùng dễ hiểu vì IE8 không support HTML5 và CSS3.

### 1.1.2. Back-End

#### 1.1.2.1. Ngôn ngữ lập trình PHP

PHP (được viết tắt là Hypertext Preprocessor hay Personal Home Page) - là một ngôn ngữ của lập trình sử dụng mã nguồn mở để phát triển các ứng dụng web . Hiện nay, nó được xem như ngôn ngữ lập trình web phổ biến, bởi sự đơn giản thân thiện và tương thích cao với mọi ngôn ngữ, trình duyệt web.PHP được phát triển và tối ưu liên tục từ các nhà phát triển trên toàn thế giới khiến cho nó vô cùng thân thiện với mọi người và ngày càng hoàn thiện hơn. Nó cũng có thể phát triển cấu trúc cao cấp cho các trang web phức tạp, cho ra kết quả tương tự như các ngôn ngữ khác.



Hình 1. 4. Ngôn ngữ lập trình PHP

Những ưu điểm của PHP mang lại khi thiết kế Website:

**- Mã nguồn mở:** Chắc hẳn, đây là điều mà nhiều bạn đã biết PHP khi lựa chọn ngôn ngữ này để học. Nó được biết đến là một sản phẩm mã nguồn mở (Open-source), việc cài đặt và tùy biến PHP đều miễn phí và tự do. Nhờ ưu thế nguồn mở, mà PHP có thể được cài đặt trên hầu hết các Web Server thông dụng hiện nay như Apache, IIS.

**- Tính Cộng đồng của PHP:** Do là ngôn ngữ mã nguồn mở, cộng thêm sự phổ biến của PHP mà PHP có cộng đồng khá lớn và chất lượng khỏi bàn cãi. Với sự phát triển của cộng đồng lớn, nên việc cập nhật các bản vá lỗi phiên bản hiện tại cũng như thử nghiệm các phiên bản mới, làm cho PHP trở nên linh hoạt hơn trong việc hoàn thiện mình. Ngoài ra, PHP còn có Cộng đồng hỗ trợ, chia sẻ kinh nghiệm dồi dào, sở hữu nhiều diễn đàn, blog cả trong và ngoài nước, đã khiến cho quá trình tiếp cận PHP của những người mới bắt đầu được rút ngắn và nhanh chóng.

**- Thư viện phong phú:** Ngoài sự hỗ trợ của cộng đồng, PHP còn có thêm thư viện script PHP cũng rất phong phú và đa dạng. Từ những cái rất nhỏ như chỉ 1 đoạn code, 1 hàm(PHP.net…) cho tới những vấn đề lớn hơn như Framework (Zend, CakePHP, Codeigniter, Symfony…) hay các ứng dụng hoàn chỉnh (Joomla, WordPress, PhpBB…). Cùng với đó là thư viện code phong phú, việc học tập và ứng dụng của PHP trở nên dễ dàng và nhanh chóng. Đây chính là điểm nổi bật làm cho nó trở nên nổi tiếng và ngày càng được nhiều người sử dụng PHP vào quá trình phát triển web.

**- Hỗ trợ kết nối nhiều hệ cơ sở dữ liệu:** Hiện nay, nhu cầu xây dựng web có sử dụng cơ sở dữ liệu là điều tất yếu, PHP là ngôn ngữ đáp ứng rất tốt mọi nhu cầu. Với việc tích hợp sẵn các Database Client trong PHP, làm cho ứng dụng PHP dễ dàng kết nối tới các hệ cơ sở dữ liệu thông dụng khác. Việc cập nhật và nâng cấp các Database Client đơn giản chỉ là thay thế các Extension trong PHP cho phù hợp với hệ cơ sở dữ liệu mà PHP trực tiếp làm việc với chúng như: MySQL, MS SQL, Oracle, Cassandra…

**- Lập trình hướng đối tượng:** Khái niệm lập trình hướng đối tượng (OOP) đã không còn quá xa lạ với lập trình viên. Nhờ khả năng và lợi ích của mô hình lập trình mang lại nên nhiều ngôn ngữ đã triển khai để hỗ trợ OOP. Từ phiên bản PHP 5, PHP đã có khả năng hỗ trợ hầu hết các đặc điểm nổi bật của lập trình hướng đối tượng như là Inheritance, Abstraction, Encapsulation, Polymorphism, Interface, Autoload. Với việc ngày càng có nhiều Framework và ứng dụng PHP viết bằng mô hình OOP nên lập trình viên tiếp cận và mở rộng các ứng dụng này trở nên dễ dàng và nhanh chóng.

**- Tính Bảo mật:** Bản thân PHP là mã nguồn mở và cộng đồng phát triển rất tích cực nên có thể nói PHP khá là an toàn. Nó còn cung cấp nhiều cơ chế cho phép bạn triển khai tính bảo mật cho các ứng dụng của mình như session, các hàm filter dữ liệu, kỹ thuật ép kiểu, thư viện PDO (PHP Data Object) để tương tác với cơ sở dữ liệu an toàn hơn. Kết hợp các kỹ thuật bảo mật ở các tầng khác thì ứng dụng PHP sẽ trở nên chắc chắn hơn và đảm bảo mọi hoạt động cho website.

**- Khả năng mở rộng cho PHP:** Được xây dựng dựa trên nền ngôn ngữ C, là mã nguồn mở nên khả năng mở rộng cho ứng dụng PHP có thể nói là không có giới hạn. Với thư viện phong phú và khả năng mở rộng lớn, ứng dụng PHP có thể tương tác với hầu hết các loại ứng dụng phổ biến như xử lý hình ảnh, nén dữ liệu, mã hóa, thao tác file PDF, Office, Email, Streaming… Bạn cũng hoàn toàn có thể tự xây dựng cho mình các Extension để tối ưu và bổ sung các chức năng cho PHP cũng như tối ưu luôn Core của PHP để phục vụ cho các mục đích mở rộng website của mình.

#### 1.1.2.2. Cơ Sở Dữ Liệu XAMPP

Xampp là chương trình tạo máy chủ Web(Web Server) được tích hợp sẵn Apache, PHP, MySQL, FTP Server, Mail Server và các công cụ như phpMyAdmin. Không như Appserv, Xampp có chương trình quản lý khá tiện lợi, cho phép chủ động bật tắt hoặc khởi động lại các dịch vụ máy chủ bất kỳ lúc nào.

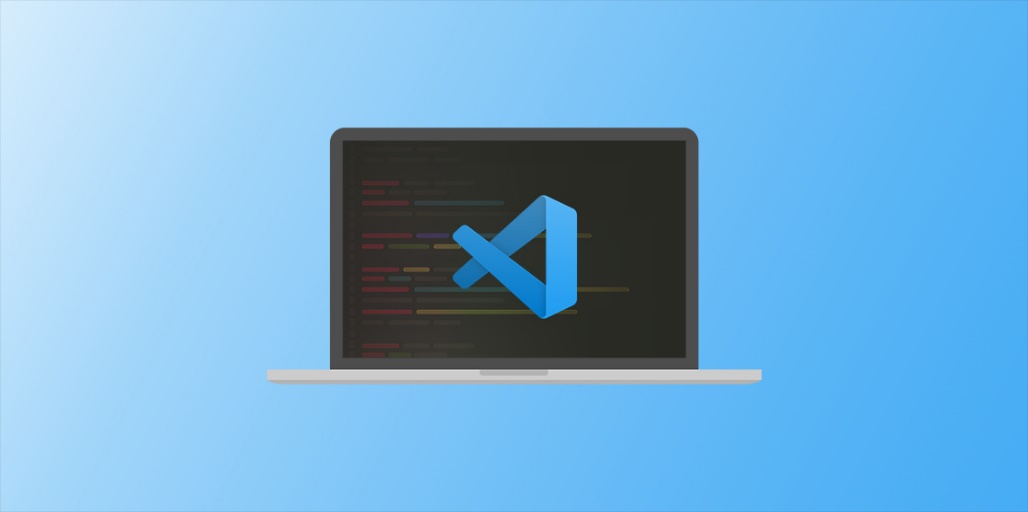


Hình 1. 5. Cơ sở dữ liệu phpMyAdmin Xampp

Xampp là một chương trình mã nguồn mở máy chủ web đa nền được phát triển bởi Apache Friends, bao gồm chủ yếu là Apache HTTP Server, MariaDB database, và interpreters dành cho những đối tượng sử dụng ngôn ngữ PHP và Perl. Xampp là viết tắt của Cross-Platform (đa nền tảng-X), Apache (A), MariaDB (M), PHP (P) và Perl (P). Nó phân bố Apache nhẹ và đơn giản, khiến các lập trình viên có thể dễ dàng tạo ra máy chủ web local để kiểm tra và triển khai trang web của mình. Tất cả mọi thứ cần cho phát triển một trang web - Apache (ứng dụng máy chủ), Cơ sở dữ liệu (MariaDB) và ngôn ngữ lập trình (PHP) được gói gọn trong 1 tệp. Xampp cũng là 1 chương trình đa nền tảng vì nó có thể chạy tốt trên cả Linux, Windows và MacOS. Hầu hết việc triển khai máy chủ web thực tế đều sử dụng cùng thành phần như XAMPP nên rất dễ dàng để chuyển từ máy chủ local sang máy chủ online

#### 1.1.2.3. Công cụ lập trình Visual Studio Code

Visual Studio Code là một trình soạn thảo mã nguồn được phát triển bởi Microsoft dành cho Windows, Linux và macOS. Nó hỗ trợ chức năng debug, đi kèm với Git, có chức năng nổi bật cú pháp (syntax highlighting), tự hoàn thành mã thông minh, snippets, và cải tiến mã nguồn. Nó cũng cho phép tùy chỉnh, do đó, người dùng có thể thay đổi theme, phím tắt, và các tùy chọn khác. Nó miễn phí và là phần mềm mã nguồn mở theo giấy phép MIT,mặc dù bản phát hành của Microsoft là theo giấy phép phần mềm miễn phí.



Hình 1. 6. Công cụ lập trình Visual Studio Code

Visual Studio Code được dựa trên Electron, một nền tảng được sử dụng để triển khai các ứng dụng Node.js máy tính cá nhân chạy trên động cơ bố trí Blink. Mặc dù nó sử dụng nền tảng Electronnhưng phần mềm này không phải là một bản khác của Atom, nó thực ra được dựa trên trình biên tập của Visual Studio Online (tên mã là "Monaco").

## 1.2. Công cụ kiểm thử phần mềm

### 1.2.1. Công cụ Cypress

#### 1.2.1.1. Khái Niệm

Cypress là front end testing tool được xây dựng cho các ứng dụng Web hiện đại. Đây là một công cụ hỗ trợ hữu hiệu cho developers và QA trong kiểm thử ứng dụng Web hiện đại. Cypress được xây dựng trên một kiến trúc mới và chạy trong cùng vòng lặp chạy khi ứng dụng đang được thử nghiệm. Cypress thường được so sánh với Selenium. Nhưng Cypress hoàn toàn khác biệt với Selenium và nó không gặp phải các hạn chế như Selenium. Điều này giúp chúng ta thực thi các Test Cases một cách đơn giản và dễ dàng hơn.

Cypress hỗ trợ chúng ta:

- *Set up tests*: (cài đặt chạy test) dễ dàng, đơn giản: Chúng ta không cần phải cấu hình bất cứ máy chủ, driver, thư viện phụ thuộc nào để sử dụng cypress, hay chạy test.

- *Write tests:* Viết test trên cypress vô cùng dễ dàng và dễ đọc, dễ hiểu. Bởi vì cách đặt tên các API, các ngôn ngữ viết test mà cypress hỗ trợ đều được đặt tên dễ hiểu

- *Run tests:* Cypress có thể chạy test bất cứ lúc nào muốn và sẽ có thể theo dõi việc chạy thực tế như nào trên trình duyệt

- *Debug Tests:* Khi theo dõi chạy test của cypress trên trình duyệt, thì cũng dễ dàng tìm hiểu về bug khi nó bị phát hiện. Cypress cung cấp một giao diện rất trực quan cho việc debug này.

Đằng sau Cypress là một máy chủ Node.js. Quá trình Cypress và Node.js liên tục liên lạc, đồng bộ hóa và thực hiện các nhiệm vụ thay mặt cho nhau. Có quyền truy cập vào cả hai phần (front and back) cho phép khả năng phản hồi các sự kiện của ứng dụng trong thời gian thực, đồng thời hoạt động bên ngoài trình duyệt cho các tác vụ yêu cầu đặc quyền cao hơn.

#### 1.2.1.2. Các tính năng của Cypress

**- Time Travel**: Cypress chụp ảnh khi chạy kịch bản test. Có thể di chuột qua các câu lệnh trong *Command Log* để xem các step hoạt động thế nào.

**- Debuggability**: Dễ dàng biết nơi nào xảy ra lỗi và thông tin cung cấp của nó rất dễ đọc.

**- Automatic Waiting**: Không cần thêm thời gian wait hoặc sleeps vào scripts của bạn, Cypress tự động đợi phần tử xuất hiện trước khi tiếp tục, sẽ không còn địa ngục async nữa.

**- Spies, Stubs, and Clocks**: Xác minh và kiểm soát hoạt động của các chức năng, phản hồi của máy chủ hoặc bộ hẹn giờ.

**- Network Traffic Control:**: Dễ dàng kiểm soát và kiểm tra các trường hợp liên quan đến băng thông mạng, bạn có thể khai báo lưu lượng mạng theo bất kì cách nào bạn mong muốn.

**- Consistent Results**: Kiến trúc không sử dụng Selenium hoặc Webdriver, các kịch bản test được thực hiện nhanh chóng, nhất quán và đáng tin cậy.

**- Screenshots and Videos**: Tự động chụp màn hình khi xảy ra lỗi và quay video toàn bộ quá trình chạy kịch bản test khi chạy từ CLI.

**- Cross browser Testing**: Hỗ trợ chạy trên Firefox, Chrome, Edge, Election và tối ưu trong một quy trình tích hợp liên tục.

#### 1.2.1.3. Ưu điểm nhược điểm của Cypress

#### 1.2.1.3.1. Ưu điểm

- Mã nguồn mở, không mất phí.

- Đơn giản, dễ cài đặt và viết kịch bản test.

- Document đầy đủ, chi tiết, có tích hợp sẵn các examples trong tool giúp cho người mới bắt đầu dễ tiếp cận hơn.

- Debug dễ dàng do các thao tác đã được ghi lại trong quá trình chạy kịch bản test.

- Không cần thêm thời gian wait hoặc sleeps vào scripts của bạn, Cypress tự động đợi phần tử xuất hiện trước khi tiếp tục.

- Dễ dàng kiểm soát và kiểm tra các trường hợp liên quan đến băng thông mạng, bạn có thể khai báo lưu lượng mạng theo bất kì cách nào bạn mong muốn.

- Hỗ trợ chụp ảnh lỗi và quay video toàn bộ quá trình chạy kịch bản test.

- Cung cấp trang *Dashboard service* có thể xem tổng quan, báo cáo kết qủa sau mỗi lần chạy.

#### 1.2.1.3.2. Nhược điểm

- Cộng đồng sử dụng chưa nhiều.

- Không hỗ trợ test trên native mobile app.

- Không thể hỗ trợ tương tác với nhiều tab do chạy bên trong trình duyệt.

- Nếu là bản free thì trang *Dashboard service* có 1 số hạn chế nhất định, nếu muốn sử dụng full dịch vụ thì phải mất phí.

- Chỉ hỗ trợ Javascript.

- Hỗ trợ 1 số ít framwork: Mocha JS.

- Trình duyệt được hỗ trợ còn hạn chế: chỉ hỗ trợ Firefox, Edge, Chrome, Election.

### 1.2.2. Công cụ Cucumber

#### 1.2.2.1. Khái niệm

Cucumber là một công cụ kiểm thử tự động dựa trên việc thực thi các functions được mô tả dướng dạng plain-text, để hỗ trợ cho việc viết Behavior Driven Development(BDD). BDD có khả năng tạo ra các kịch bản test dựa trên góc nhìn của bên phát triển cũng như góc nhìn của bên khác hàng. Ngay từ ban đầu, các thành viên dự án sẽ thảo luận để tạo ra các kịch bản trước, sau đó sẽ cài đặt dựa trên kịch bản đó, tất cả kịch bản test gần gũi với ngôn ngữ tiếng Anh, do đó nó đóng luôn vai trò của tài liệu.



Hình 1. 7. Công cụ Cucumber

Nói theo cách khác Cucumber là công cụ phần mềm được sử dụng bởi tester để thực hiện các trường hợp kiểm thử kiểm tra hành vi của phần mềm.Ban đầu Cucumber chỉ dành riêng cho Ruby nhưng hiện nay đã hỗ trợ nhiều ngôn ngữ lập trình khác nhau như: PHP, Java, JavaScript, C# và Python.Ở Cucumber testing, các trường hợp kiểm thử được viết dưới dạng văn bản theo ngôn ngữ tự nhiên, ai đọc cũng có thể hiểu mà không cần bất kì kiến thức kĩ thuật nào. Ngôn ngữ này được gọi là ngôn ngữ Gherkin.

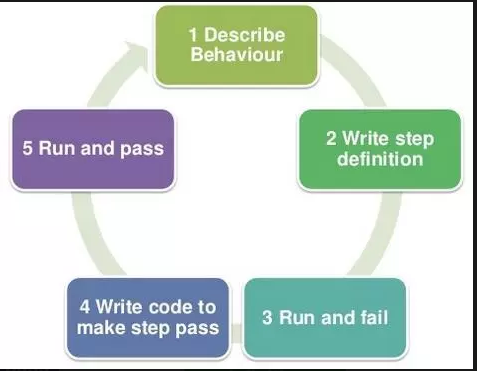
#### 1.2.2.2. Cách hoạt động của Cucumber

Cucumber test cases được viết song song với quá trình phát triển phẩn mềm. Các testcase này được gọi là **steps** trong ngôn ngữ Gherkin.

- Đầu tiên Cucumber đọc các **steps** được viết bằng ngôn ngữ Gherkin trong **feature** file.

- Sau đó, nó tìm kiếm **steps** này trong step definition file( sẽ định nghĩa cách thức hoạt động và assert các kết quả thu được để kiểm tra các step có hoạt động một cách chính xác hay không). Nếu trùng khớp thì sẽ thực hiện chạy testcase và cung cấp kết quả pass/fail sau khi chạy xong.

Ngoài ra Cucumber hỗ trợ một số các phương thức helper điển hình như hooks cho phép chúng ta có thể thực các công việc nhất định tại một thời điểm thực thi các test case như trước khi bắt đầu một Scenario hay sau khi kết thúc một Scenario,…



Hình 1. 8. Mô tả cách thức hoạt động của Cucumber

Lợi ích khi sử dụng Cucumber:

- Giúp cho các bên liên quan đến dự án (stakeholders,…) có thể theo dõi hoạt động test mà không cần kiến thức kĩ thuật chuyên môn.

- Tập trung vào trải nghiệm của người dùng cuối.

- Cho phép tester dễ dàng tái sử dụng lại code trong các trường hợp kiểm thử.

- Hỗ trợ hầu hết tất cả các ngôn ngữ phổ biến khác nhau như [Java.net](http://java.net/)**,** JavaScript Ruby, PHP, v.v.

- Dễ cài đặt và sử dụng.

#### 1.2.2.3. So sánh Cucumber với các công cụ khác

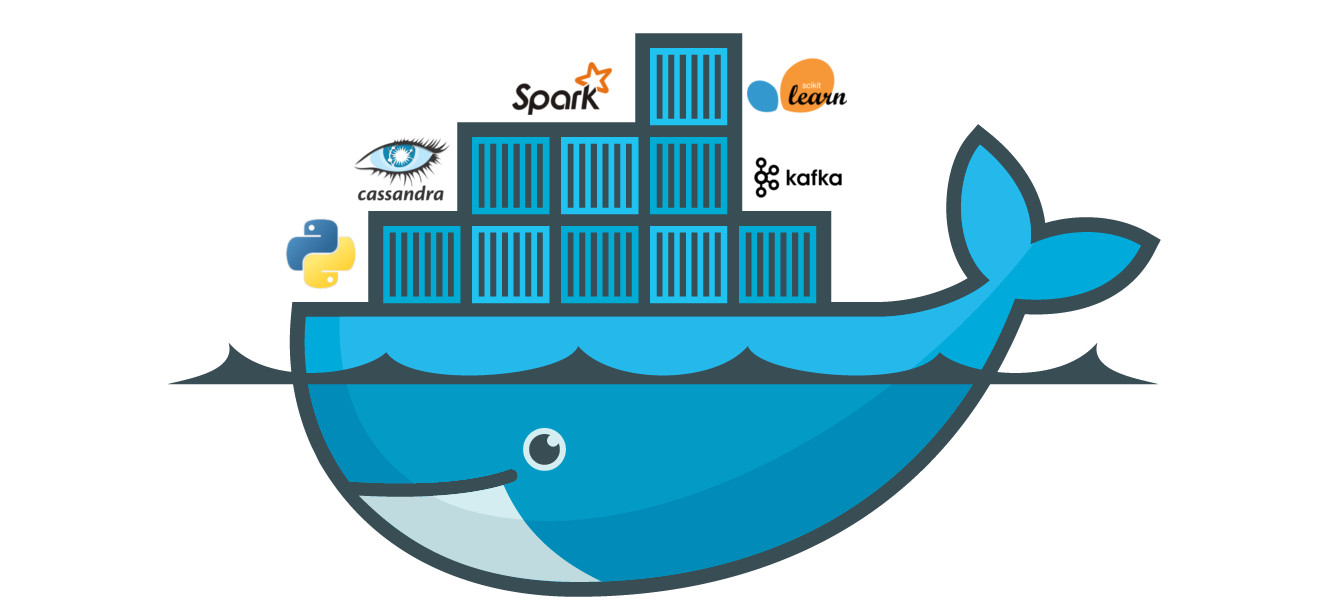


Hình 1. 9. So sánh Cucumber với các công cụ khác

### 1.2.3. Docker

#### 1.2.3.1. Khái niệm

Docker là một nền tảng cho developers và sysadmin để develop, deploy và run application với container. Nó cho phép tạo các môi trường độc lập và tách biệt để khởi chạy và phát triển ứng dụng và môi trường này được gọi là container. Khi cần deploy lên bất kỳ server nào chỉ cần run container của Docker thì application của bạn sẽ được khởi chạy ngay lập tức.



Hình 1. 10. Hình ảnh Docker

#### 1.2.3.2. Sự ra đời của Docker

Trước đây, việc setup và deploy application lên một hoặc nhiều server rất vất vả từ việc phải cài đặt các công cụ, môi trường cần cho application đến việc chạy được ứng dụng chưa kể việc không đồng nhất giữa các môi trường trên nhiều server khác nhau.Vì lý do đó, Docker ra đời để giải quyết vấn đề này.Docker cho phép các developers tạo các môi trường độc lập và tách biệt để khởi chạy và phát triển ứng dụng và môi trường này được gọi là container. Khi cần deploy lên bất kỳ server nào chỉ cần run container của Docker thì application của bạn sẽ được khởi chạy ngay lập tức.

Hãy tưởng tượng bạn code 1 cái ứng dụng, có 1 cái web server được viết bằng NodeJs, có database là PostgreSQL, sử dụng Redis để lưu trữ message,... Mỗi cái như này các bạn phải cài rất là nhiều thư viện, dependencies, đi kèm theo đó là những cái config.. Và khi các bạn code xong, muốn cho người dùng sử dụng thì các bạn phải deploy nó.

Trong môi trường deploy đó thì các bạn cũng phải cài đặt tất cả những cái này kèm với thư viện, config, dependencies của nó. Chỉ cần thiếu 1 thư viện hay 1 dependencies nhỏ thôi thì ứng dụng của các bạn cũng sẽ bị lỗi và k sử dụng được. Và nội cái việc set up và cài đặt tất cả những cái này thôi nó cũng đã mất rất nhiều thời gian rồi.

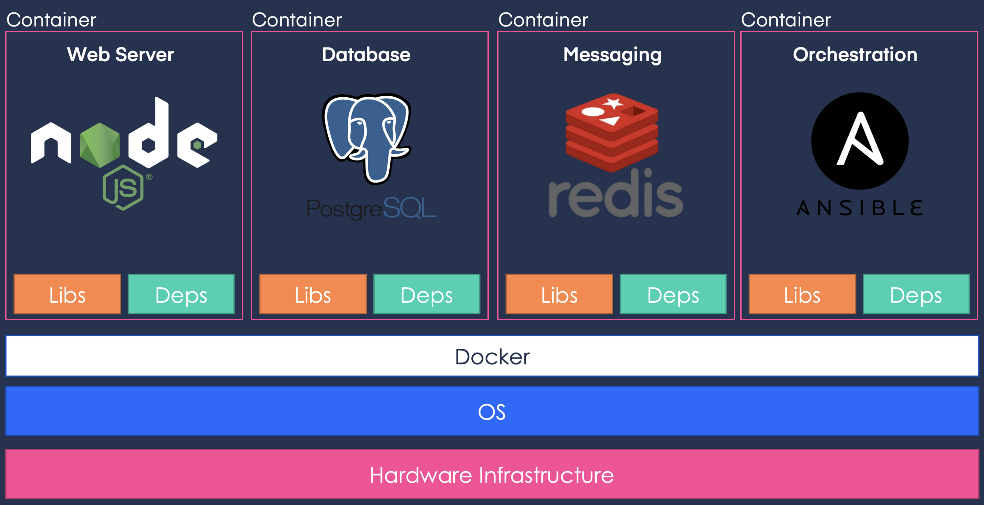
Thì đấy là cái vấn đề, làm sao mình có thể code 1 chỗ mà đưa nó được tới người dùng hay tới nhiều chỗ, nhiều môi trường khác nhau mà vẫn đảm bảo các thư viện, config, dependencies được cài đúng đắn.

Vấn đề này sẽ được giải quyết bởi Docker.

- Docker nó sẽ gom những cái như là NodeJS, PostgreSQL, Redis, Ansible, cài nó khá là phức tạp và làm sao để nó đúng version thì Docker nó gom những cái này lại thành 1 cái container.

- Docker đóng gói phần mềm vào các đơn vị tiêu chuẩn hóa được gọi là container. Các container cho phép nhà phát triển đóng gói một ứng dụng với tất cả các phần mà ứng dụng đó cần, chẳng hạn như thư viện và các phần phụ thuộc khác, và chuyển tất cả nó ra dưới dạng một gói.

- Giống như 1 cái xe container, ở trong có hàng hóa và bạn chỉ bỏ lên xe, chạy đến nơi khác. Tới nơi thì họ sẽ dỡ hàng hóa xuống và sử dụng là được. Docker nó cũng hoạt động với cơ chế như vậy. Nó sẽ gom từng thằng này thành những cái container.



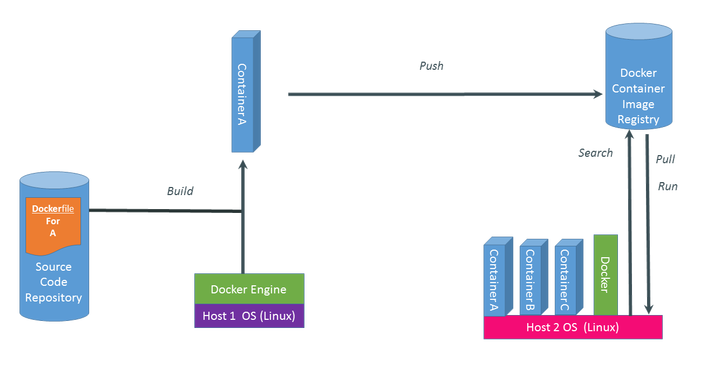
Hình 1. 11. Sự ra đời cửa Docker

- Khi các bạn viết code thì mỗi container nó sẽ bao gồm cả phần cài đặt nodejs, các thư viện, dependencies, code lại thành 1 container. Khi cần chạy thì bạn chỉ cần chạy cái container này lên thì nó sẽ có sẵn tất cả nodejs, thư viện, code... mà bạn không cần phải cài thêm bất kỳ cái gì, Docker nó sẽ lo hết tất cả những việc đó.

- Ứng dụng sau khi được đóng gói có thể hoạt động một cách nhanh chóng và hiệu quả trên các môi trường điện toán khác nhau

- Tóm lại thì Docker nó tách biệt và đóng gói phần mềm thành những đơn vị chuẩn hóa gọi là container để dễ quản lý, dễ di chuyển, share. Từ đó giúp ứng dụng có thể hoạt động một cách nhanh chóng và hiệu quả trên các môi trường điện toán khác nhau.

#### 1.2.3.3. Quy trình thực thi của một hệ thống Docker



Hình 1. . Mô tả cách thức hoạt động của Docker

***Một hệ thống Docker được thực thi với 3 bước chính :***

**- Build**: Đầu tiên tạo một dockerfile, trong dockerfile này chính là code của chúng ta. Dockerfile này sẽ được Build tại một máy tính đã cài đặt Docker Engine. Sau khi build ta sẽ có được Container, trong Container này chứa ứng dụng kèm bộ thư viện của chúng ta.

**- Push**: Sau khi có được Container, chúng ta thực hiện push Container này lên cloud và lưu tại đó.

**- Pull,Run**: Nếu một máy tính khác muốn sử dụng Container chúng ta thì bắt buộc máy phải thực hiện việc Pull container này về máy, tất nhiên máy này cũng phải cài Docker Engine. Sau đó thực hiện Run Container này.

Lợi ích khi dùng Docker

- Không như máy ảo Docker start và stop chỉ trong vài giây.

- Bạn có thể khởi chạy container trên mỗi hệ thống mà bạn muốn.

Container có thể build và loại bỏ nhanh hơn máy ảo.

- Dễ dàng thiết lập môi trường làm việc. Chỉ cần config 1 lần duy nhất và không bao giờ phải cài đặt lại các dependencies. Nếu bạn thay đổi máy hoặc có người mới tham gia vào project thì bạn chỉ cần lấy config đó và đưa cho họ.

- Nó giữ cho word-space của bạn sạch sẽ hơn khi bạn xóa môi trường mà ảnh hưởng đến các phần khác.

#### 1.2.3.4. Một số khái niệm liên quan đến Docker

**- Docker Swarm**: là một service cho phép người dùng tạo và quản lý tập chung cho Docker nodejs. Điều này cũng giống như lịch trình của các containers. Mỗi node của Docker Swarm sẽ là một Docker Daemon. Và tất cả những Docker Daemon này sẽ sử dụng Docker API. Từ đó các Services đều có thể truy cập cũng như vào các nodes như nhau. Việc quản lý bằng docker cơ bản thì các container sẽ được quản lý riêng biệt. Tuy nhiên, nếu quản lý bằng docker Swarm thì các containers sẽ được quản lý chung.

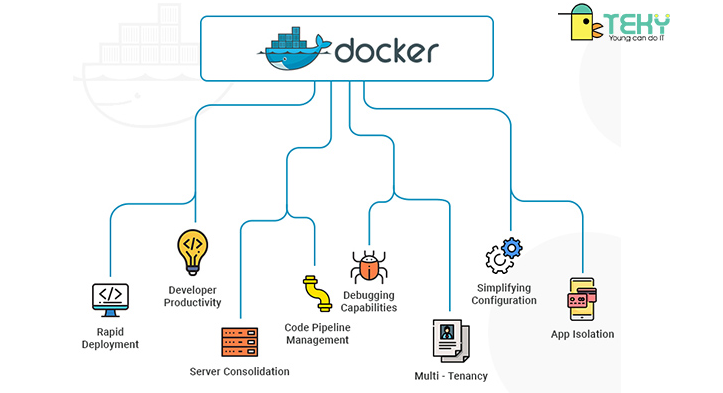
**- Docker Compose**: được dùng để định nghĩa và thực hiện quá trình run multi-container cho Docker Application. Với nền tảng này các bạn có thể sử dụng file YAML để config với các services dành cho Application. Sau đó người dùng có thể dùng command để create và run những Config đó. Để có thể sử dụng, người dùng cần làm ba bước như sau:

- Tiến hành khai báo app’s environment trong Dockerfile.

- Khai báo các services cần thiết để chạy application ở trong file docker-compose.yml.

- Và run docker-compose up để start và run app.

**- Docker Hub**: được biết đến là một dạng github for docker Images. Trên nền tảng này sẽ có hàng ngàn bức ảnh được công khai bởi mọi người. Chúng ta có thể dễ dàng tìm thấy những bức ảnh mà mình cần. Và việc của chúng ta là pull về để sử dụng cùng với một số config khác mà chúng ta mong muốn. Docker Hub được mọi người sử dụng rất nhiều bởi tính tiện lợi.

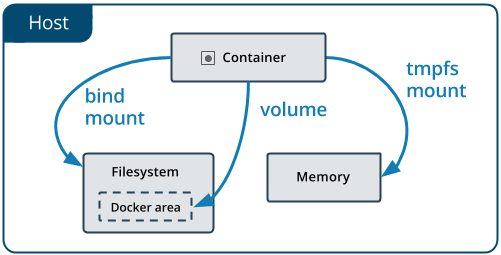


Hình 1. 13. Quá trình hoạt động của Docker

*-* **Docker Client:** là cách mà bạn tương tác với docker thông qua command trong terminal. Docker Client sẽ sử dụng API gửi lệnh tới Docker Daemon.

**- Docker Daemon:** là server Docker cho yêu cầu từ Docker API. Nó quản lý images, containers, networks và volume.

**- Docker Volumes:** là cách tốt nhất để lưu trữ dữ liệu liên tục cho việc sử dụng và tạo apps.

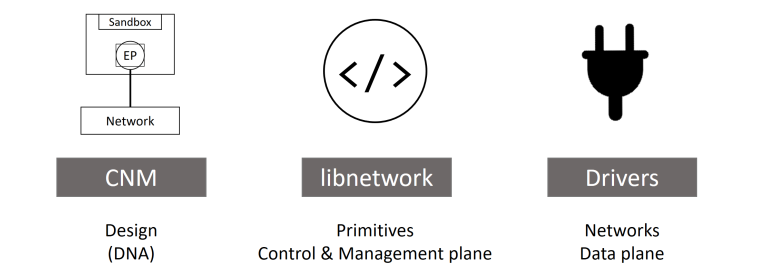


Hình 1. 14. Cách hoạt động của Docker Volume

**- Docker Registry:** là nơi lưu trữ riêng của Docker Images. Images được push vào registry và client sẽ pull images từ registry. Có thể sử dụng registry của riêng bạn hoặc registry của nhà cung cấp như : AWS, Google Cloud, Microsoft Azure.

**- Docker Repository:** là tập hợp các Docker Images cùng tên nhưng khác tags. VD: golang:1.11-alpine.

**- Docker Networking:** cho phép kết nối các container lại với nhau. Kết nối này có thể trên 1 host hoặc nhiều host.



Hình 1. 15. Cách hoạt động của Docker Networking

**- Docker Services:** là các containers trong production. 1 service chỉ run 1 image nhưng nó mã hoá cách thức để run image — sử dụng port nào, bao nhiêu bản sao container run để service có hiệu năng cần thiết và ngay lập tức.

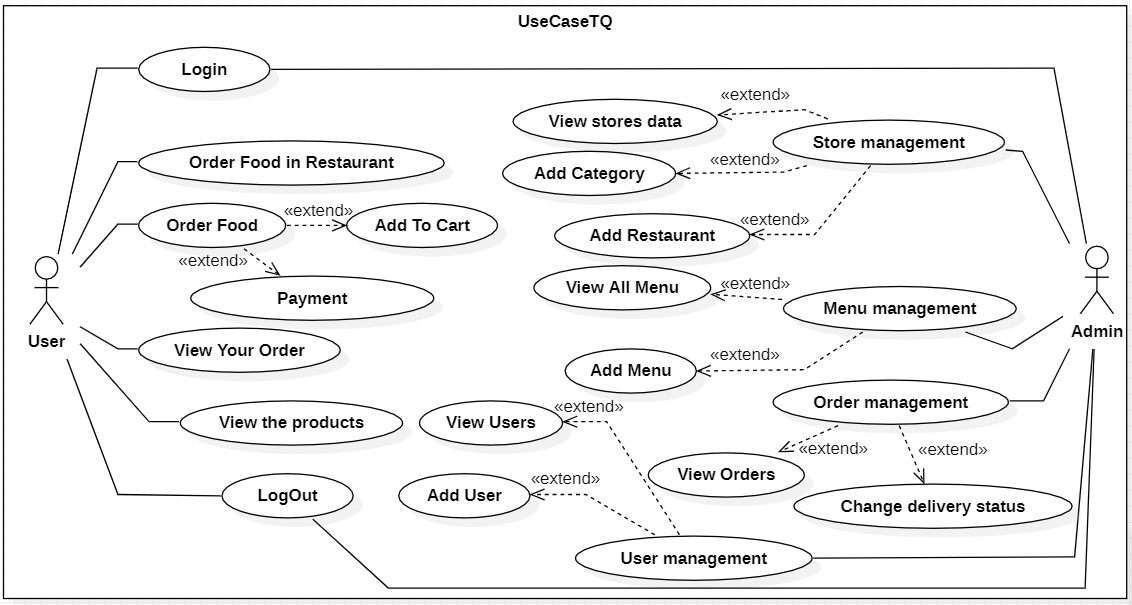
# CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG WEBSITE

## 2.1. Biểu đồ Usecase

### 2.1.1. Nhận diện tác nhân và Usecase

***Bảng 2. 1. Nhận diện tác nhân và UseCase***

|  |  |
| --- | --- |
| Tác nhân | UseCase |
| User | Đăng nhập (Login)  Đặt món trong nhà hàng (Order Food in Restaurant)  Đặt món ngay (Order Food): Thêm vào giỏ hang (Add To Cart) và thanh toán (Payment)  Xem đơn hàng đã đặt  Xem thông tin sản phẩm  Đăng xuất |
| Admin | Quản lý cửa hàng (Store management): Xem dữ liệu của các cửa hàng, them danh mục hàng, thêm nhà hàng  Quản lý thưc đơn (Menu management): Xem tất cả thực đơn, them thực đơn mới  Quản lý đơn hàng (Order management): Xem tất cả thông tin đơn hàng, thay đổi trạng thái giao hàng (đang gửi đi, đang xử lý, đã giao…)  Quản lý người dùng (User management): Xem tất cả thông tin người dùng đã đăng kí, them người dùng mới |



Hình 2. 1. Biểu đồ Usecase

### 2.1.2. Đặc tả Usecase

#### 2.1.2.1. Use Case Login

*Tên UC:* Login

*Tóm tắt:* Use Case này mô tả cách thức người dùng đã thành viên đăng nhập vào ứng dụng.

***Luồng sự kiện chính:***

- User: Từ trang home nếu người dung muốn đặt hang thì nhấn login. Sau khi đăng nhập thành công người dung có thể tiến hành đặt hàng.

- Admin: Link để vào trang admin là ../FoodAndDrink/Admin. Admin sẽ tiến hành đăng nhập với tên là: admin. Mật khẩu là: 1234. Sau khi đăng nhập thành công thì hệ thống sẽ điều hướng bạn đến trang Admin.

***Luồng sự kiện phụ:***

- Người dùng nhấn nút Order now trên trang home thì hệ thống sẽ yêu cầu đăng nhập với tư cách người dùng

#### 2.1.2.2. Use Case Order Food

*Tên UC:* Order Food

*Tóm tắt:* Use Case này mô tả cách thức người dùng đặt món ăn trên website

***Luồng sự kiện chính:***

- User: Từ trang home nếu người dung muốn đặt hàng thì nhấn login. Sau khi đăng nhập thành công người dung nhấn order now – chọn món ăn mà mình muốn đặt hàng- điền số lượng cần mua – thêm vào giỏ hàng – thanh toán

***Luồng sự kiện phụ:***

- Người dùng nhấn nút vào Restaurant và chọn món ăn sau đó them vào giỏ hàng rồi tiến hành thanh toán

- Nếu người dùng không muốn đặt món ăn đã chọn nữa thì có thể nhấn vào biểu tượng xóa để xóa món ăn đó ra khỏi giỏ hàng

## 2.2. Biểu đồ trình tự

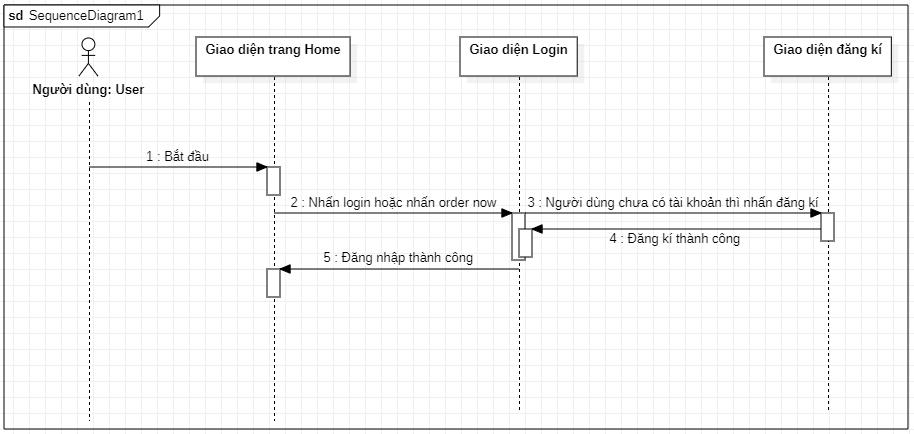
### 2.2.1. Biểu đồ trình tự đăng nhập

Biểu đồ trình tự thực hiện chức năng đăng nhập. Nếu người dùng muốn đặt đồ ăn thì bắt buộc phải đăng nhập nếu chưa có tài khoản thì phải đăng ký tài khoản. Ở biểu đồ này thì người dùng đã có tài khoản và tiến hành đăng nhập. Sau khi đăng nhập xong thì hệ thống sẽ điều hướng người dùng đến trang home với tư cách người dùng đã đăng nhập.



Hình 2. 2. Biểu đồ trình tự đăng nhập

### 2.2.2. Biểu đồ trình tự đăng ký

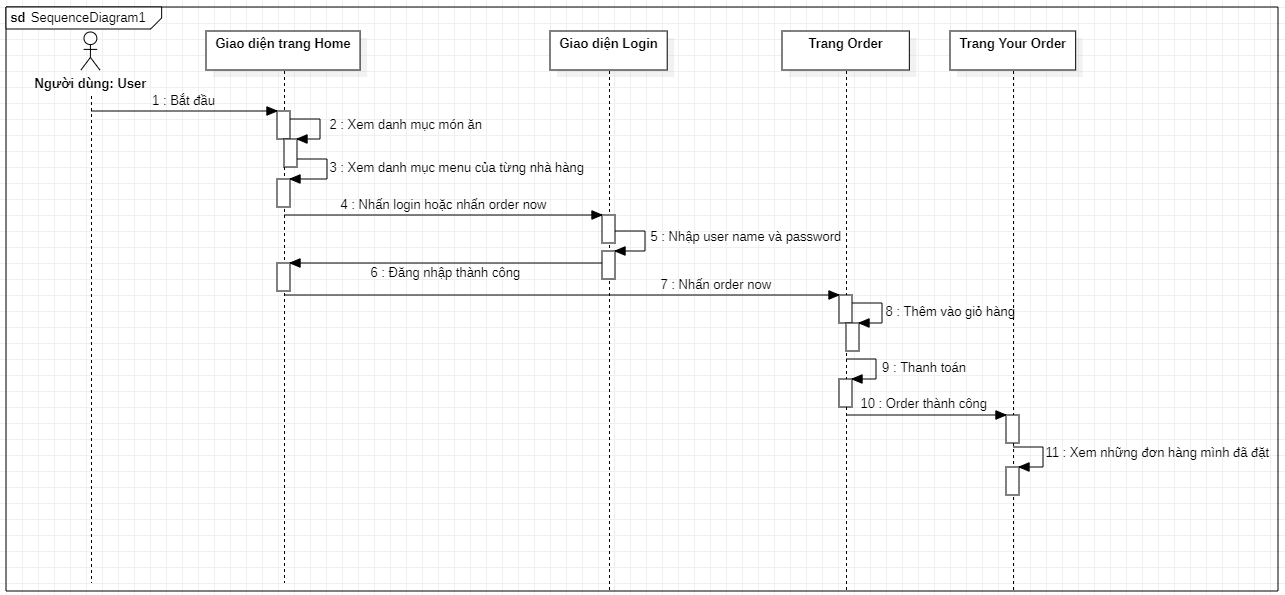


Hình 2. 3. Biểu đồ trình tự đăng ký

### 2.2.3. Biểu đồ trình tự người dùng thực hiện chức năng

Biểu đồ trình tự người dùng thực hiện chức năng bao gồm các chức năng như: xem danh mục món ăn, xem danh mục menu của từng nhà hàng, đặt món ăn (Order), xem lại những đơn hàng mình đã đặt. Vì trình tự thực hiện các chức năng này tương tự nhau nên chúng ta có thể biểu diễn trình tự các chức năng đó trên cùng một biểu đồ trình tự như sau:

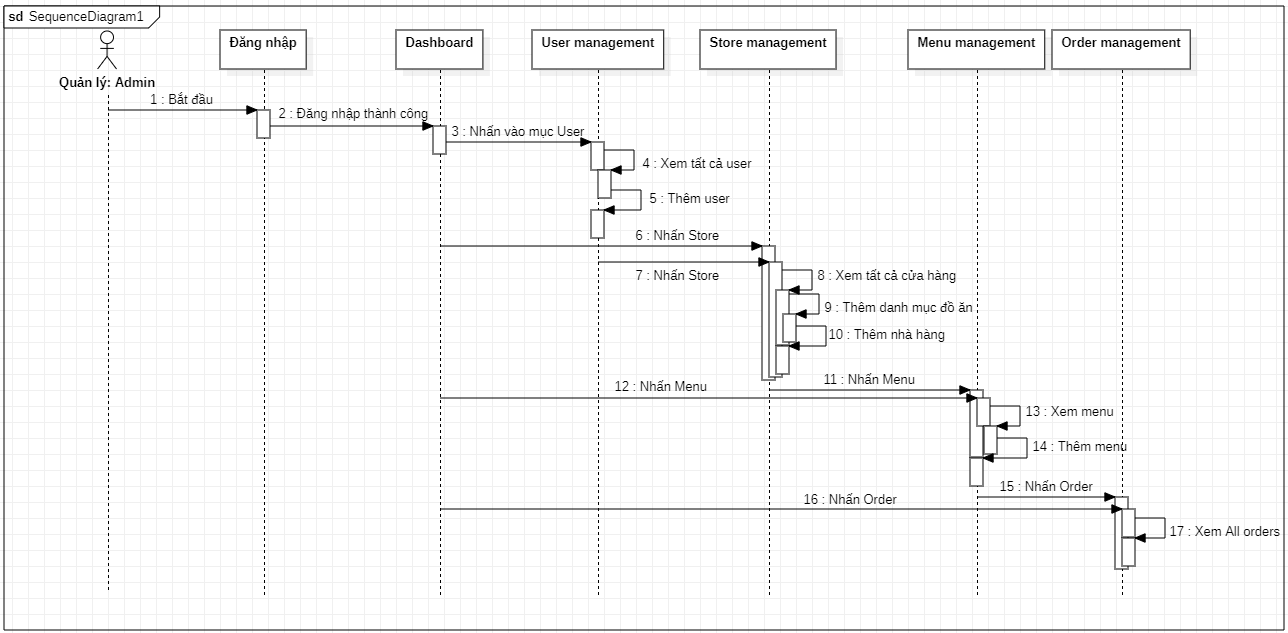
Đầu tiên người dùng sẽ được điều hướng tới trang home ở đây người dùng có thể xem danh mục, thực đơn chi tiết của từng nhà hàng. Sau đó người dùng có thể đăng nhập bằng cách nhấn login, nếu người dùng đã có tài khoản thì tiến hành nhập tên đăng nhập và mật khẩu. Trường hợp nếu người dùng chưa có tài khoản thì tiến hành đăng kí. Đăng nhập thành công người dùng sẽ được điều hướng tới trang chủ. Tại đây người dùng đã có thể tiến hành đặt món ăn, cũng như tiến hành thanh toán món ăn mình đã đặt. Đơn hàng khi chọn phương thức thanh toán và tiến hành thanh toán thành công thì sẽ được lưu tại trang your order. Tại đây người dùng có thể theo dõi đơn hàng của mình ở trạng thái đang xử lý, xác nhận giao hoặc đã giao.



Hình 2. 4. Biểu đồ trình tự người dùng thực hiện các chức năng

### 2.2.4. Biểu đồ trình tự admin thực hiện chức năng

Biểu đồ trình tự Admin thực hiện chức năng bao gồm các chức năng như: quản lý người dùng, quản lý các cửa hàng nhà hàng, quản lý thực đơn, quản lý đơn hàng. Vì trình tự thực hiện các chức năng này tương tự nhau nên chúng ta có thể biểu diễn trình tự các chức năng đó trên cùng một biểu đồ trình tự như sau:



Hình 2. 5. Biểu đồ trình tự Admin thực hiện các chức năng

## 2.3. Cấu trúc cơ sở dữ liệu

******

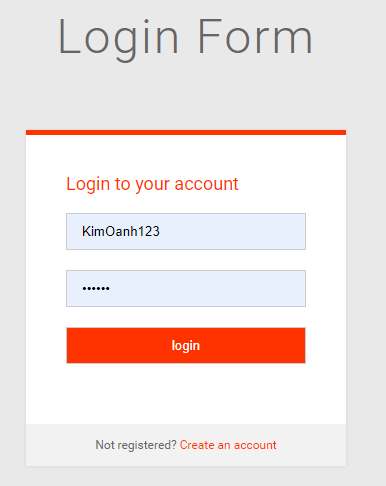
Hình 2. 6. Cấu trúc cơ sở dữ liệu

# CHƯƠNG 3. XÂY DỰNG WEBSITE QUICK FOOD VÀ KIỂM THỬ TỰ ĐỘNG CHO QUICK FOOD

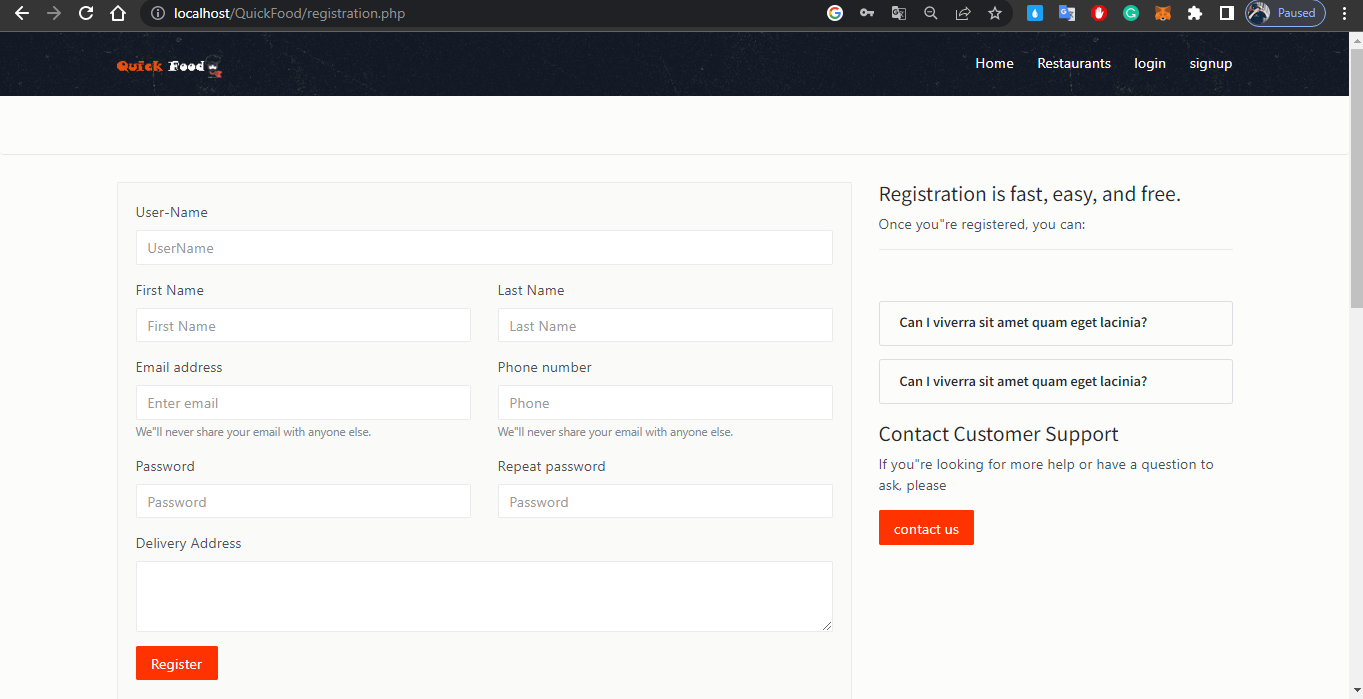
## 3.1. Về website Quick Food

Gồm hai phần luồng chính là người dùng (User) và quản trị viên (Admin). Mỗi phân luồng sẽ có những chức năng và tính năng riêng có liên kết với nhau. Dưới đây là hình ảnh và code chương trình đã được triển khai trong dự án này.

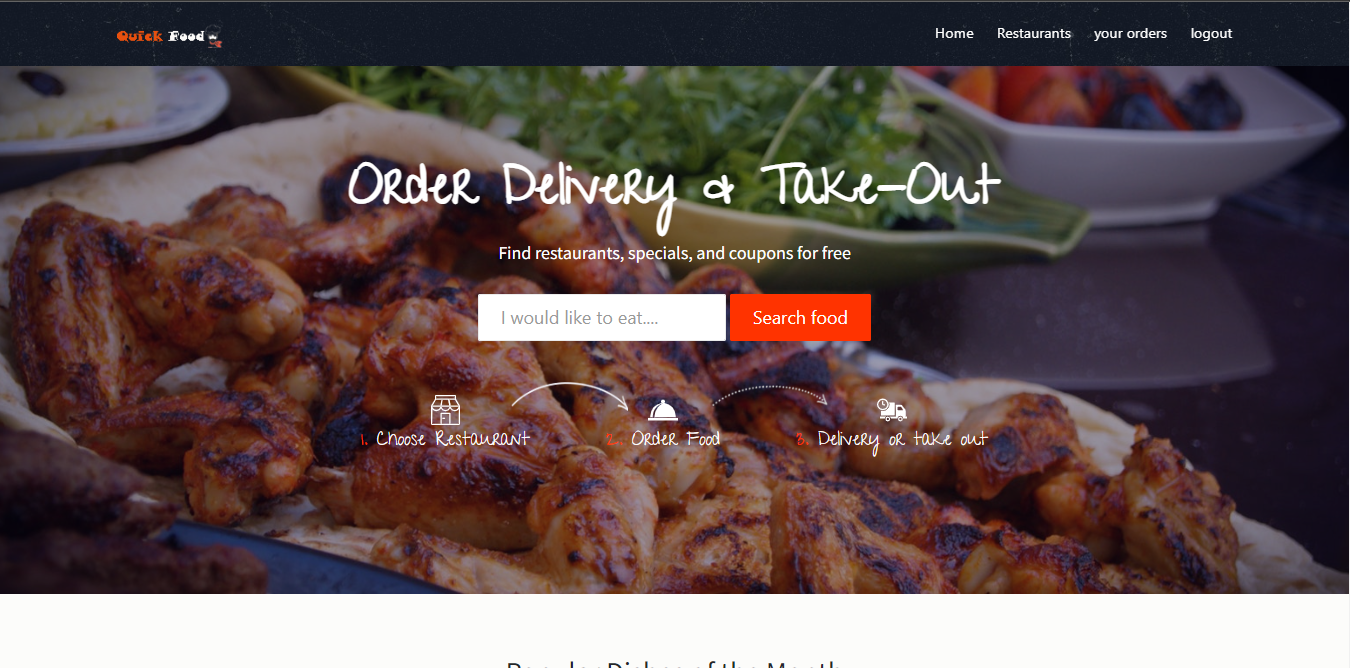
### 3.1.1. Giao diện người dùng



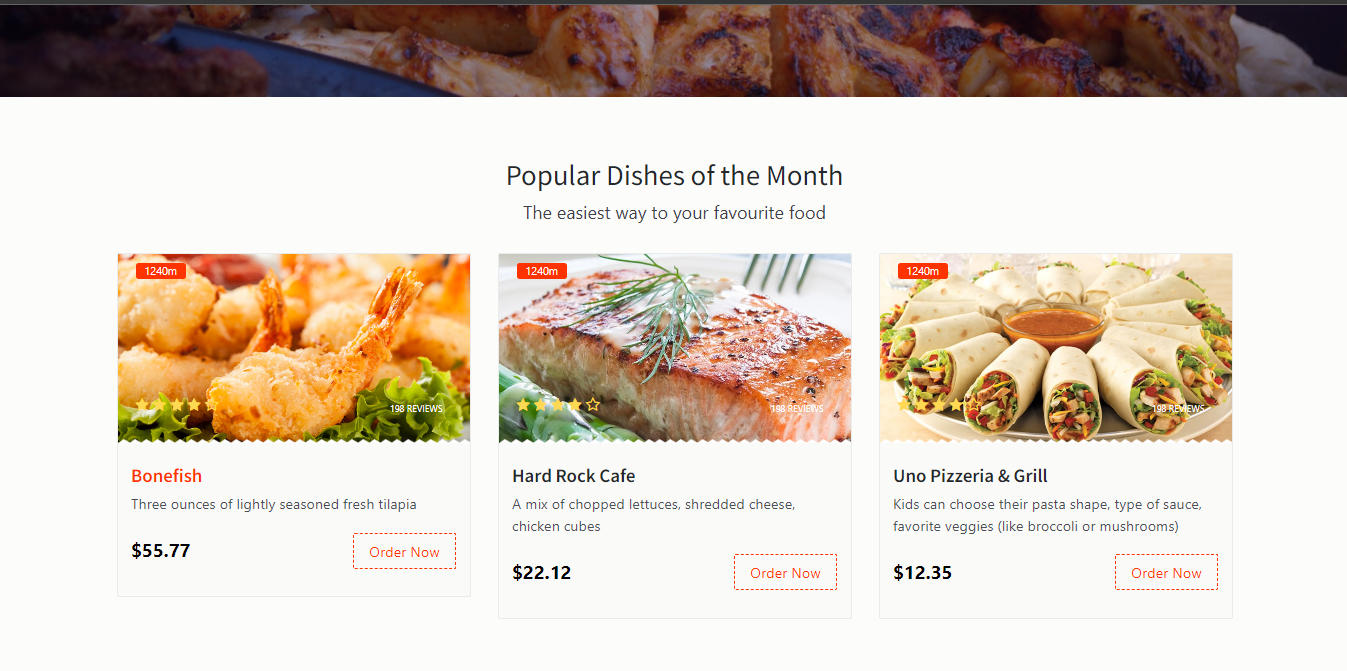
Hình 3. 1. Giao diện form đăng nhập



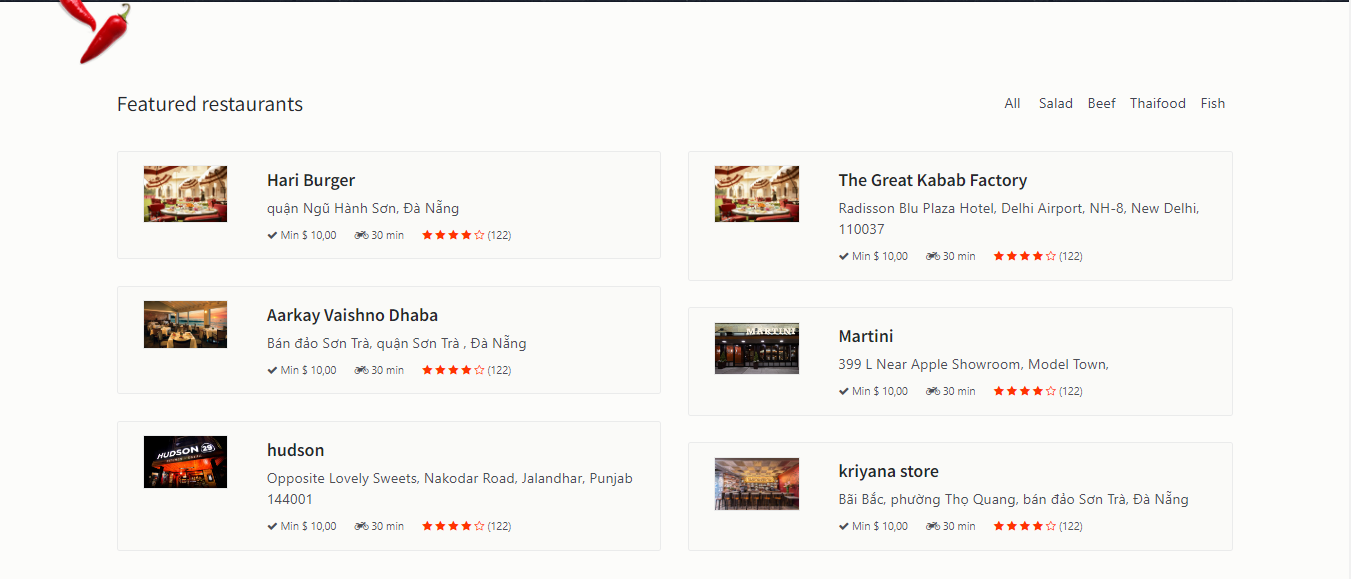
Hình 3. 2. Giao diện form đăng kí



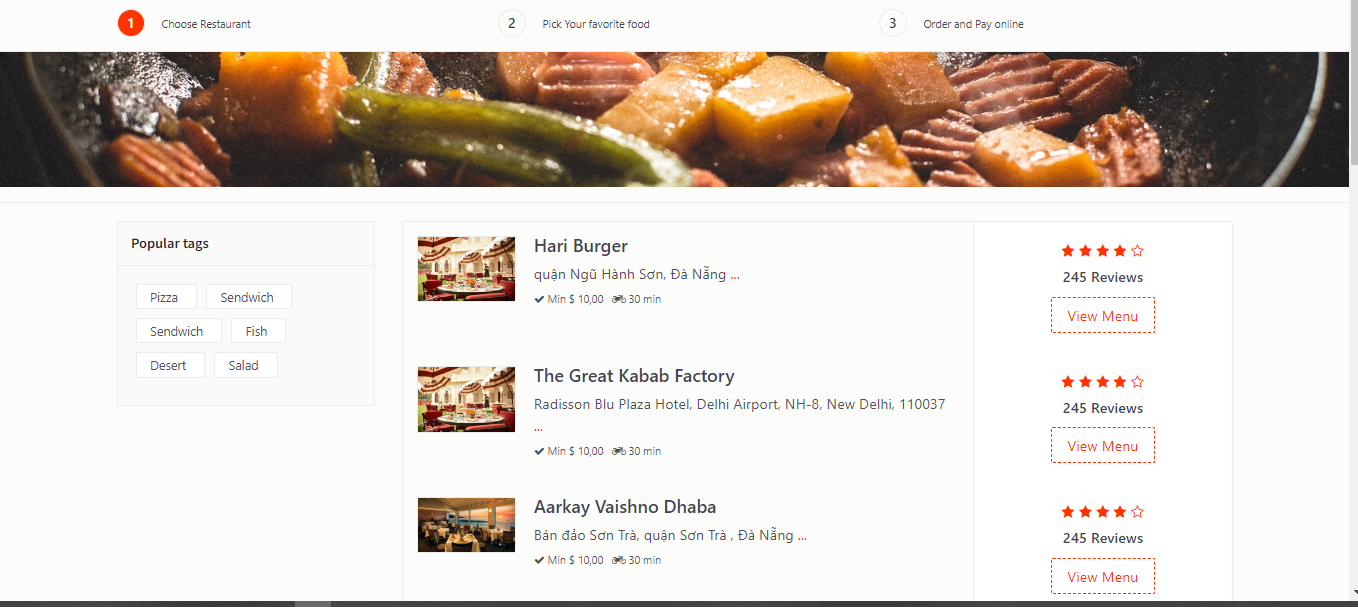
Hình 3. 3. Giao diện trang chủ



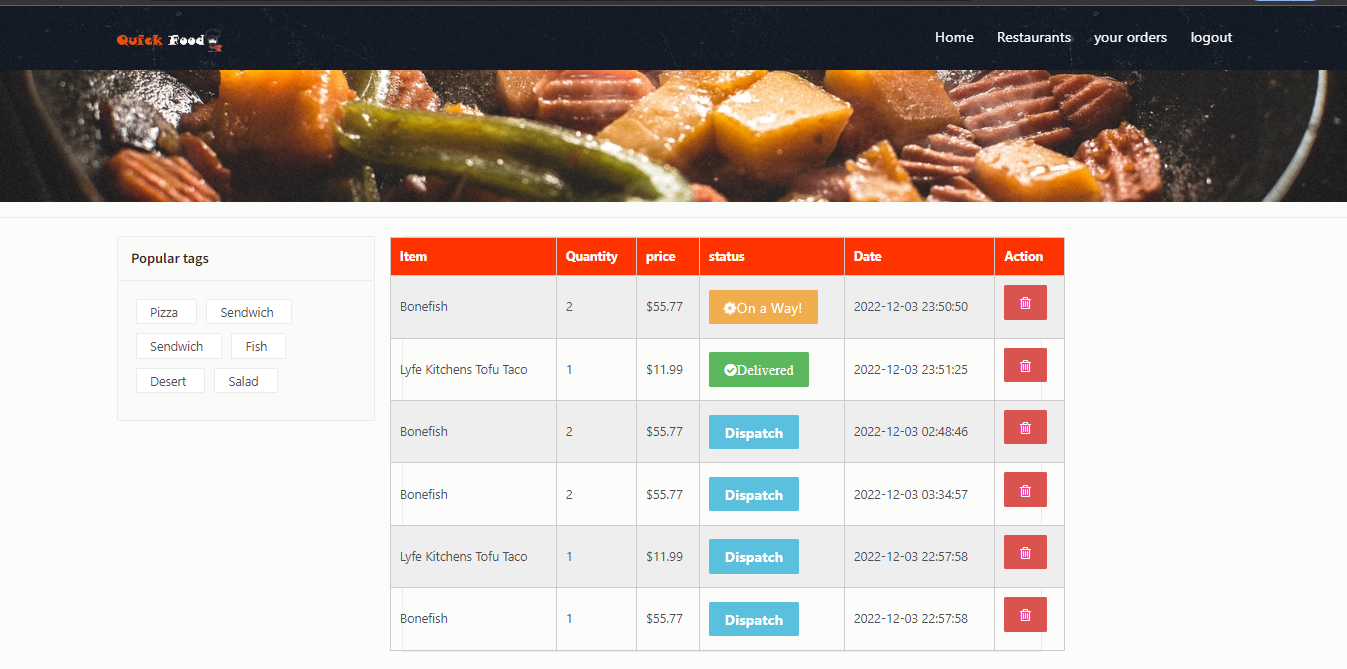
Hình 3. 4. Danh mục các món ăn trên trang chủ



Hình 3. 5. Danh mục nhà hàng trên trang chủ

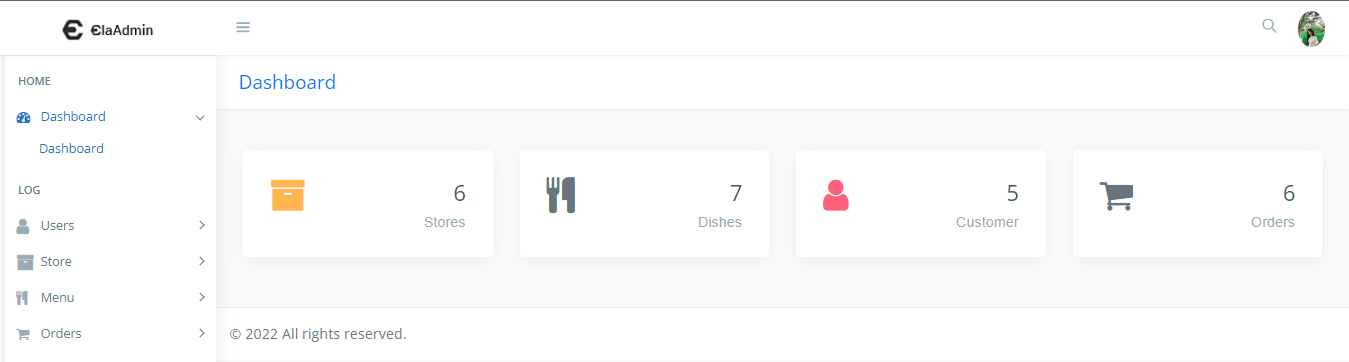


Hình 3. 6. Danh mục các nhà hàng trên trang Restaurants

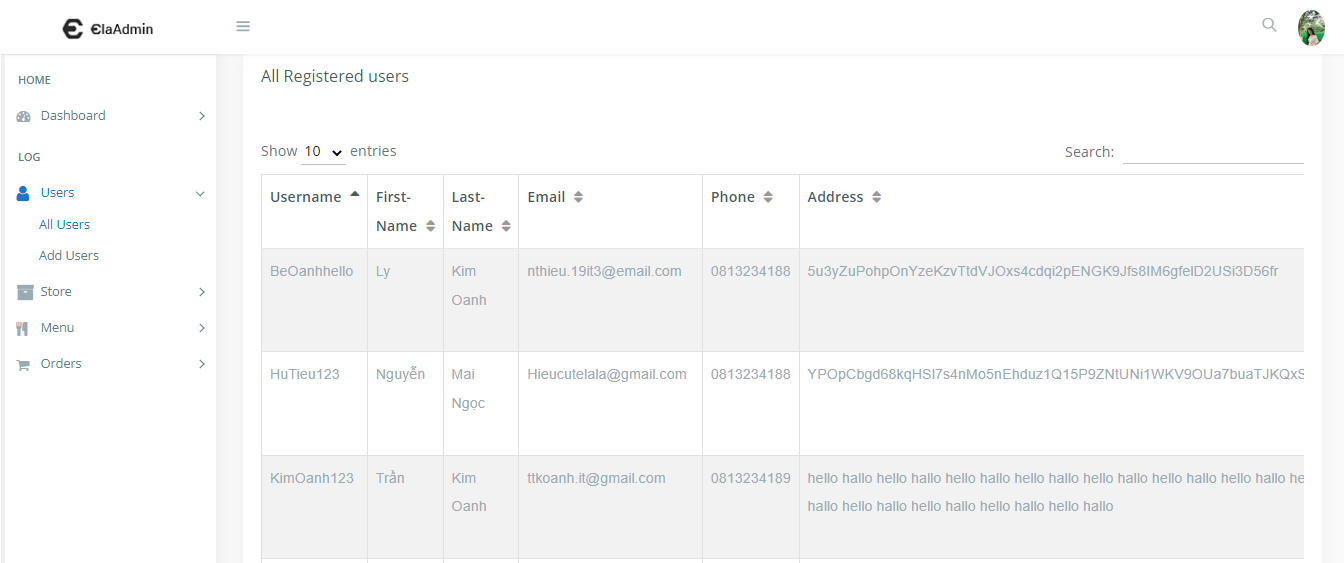


Hình 3. 7. Danh mục các đơn hàng đã đặt tại trang Your Orders

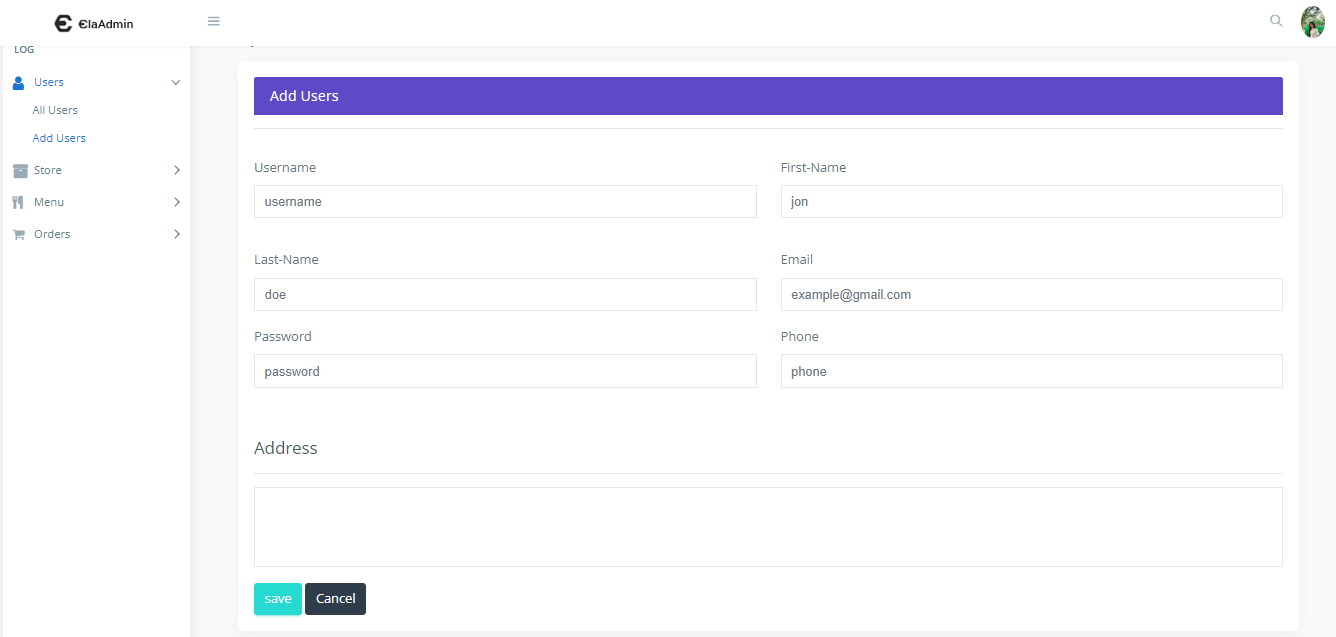
### 3.1.2. Giao diện Admin



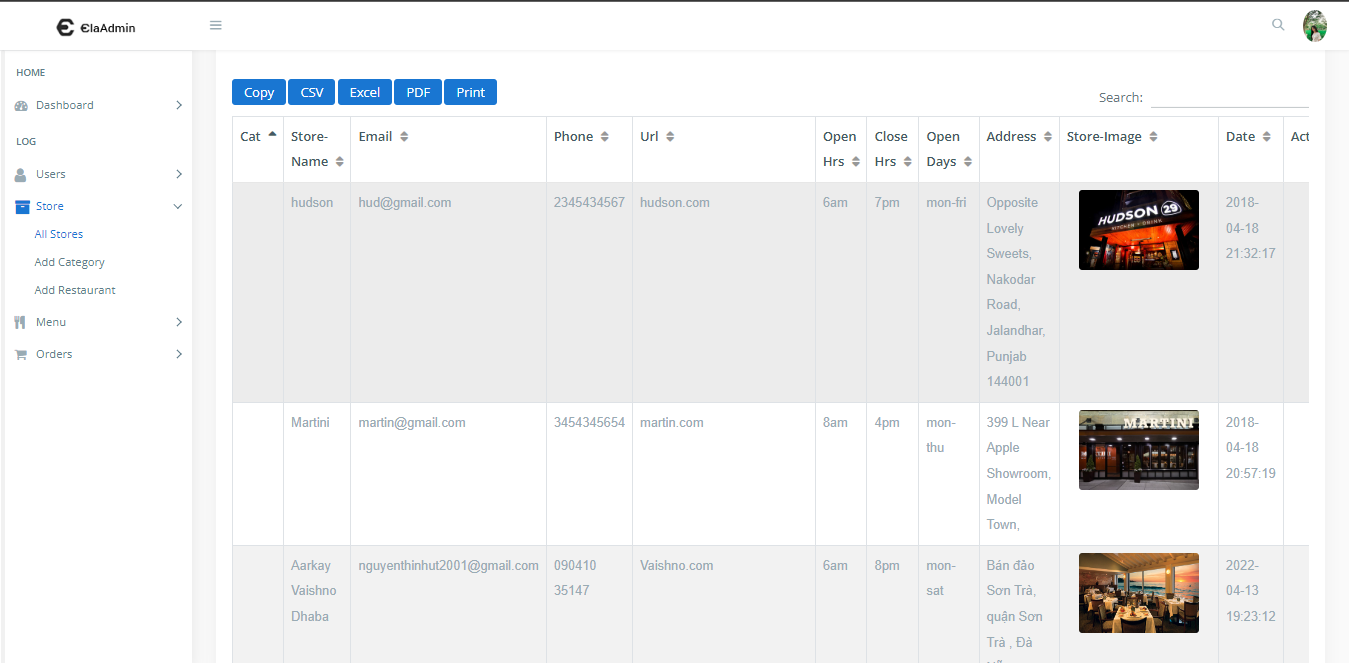
Hình 3. 8. Trang dashboard



Hình 3. 9. Xem tất cả tài khoản của những người dùng đã đăng kí thành công



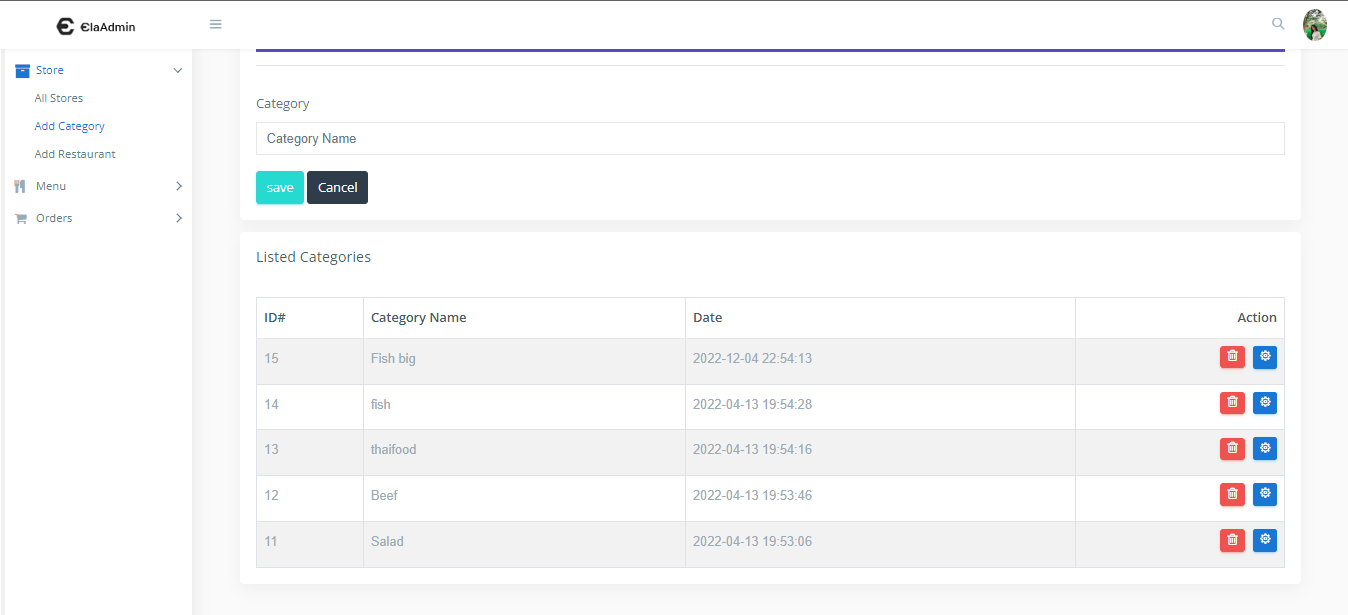
Hình 3. 10. Thêm tài khoản cho người dùng mới



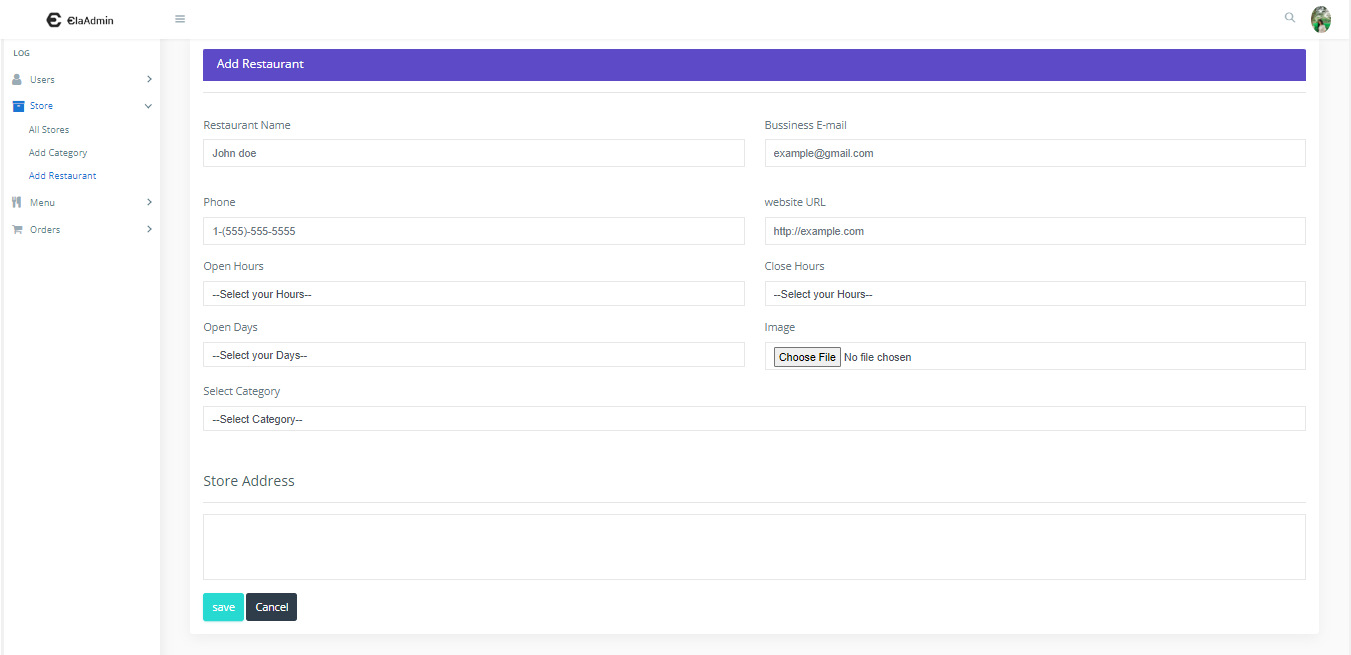
Hình 3. 11. Xem tất cả thông tin của những cửa hàng trên website



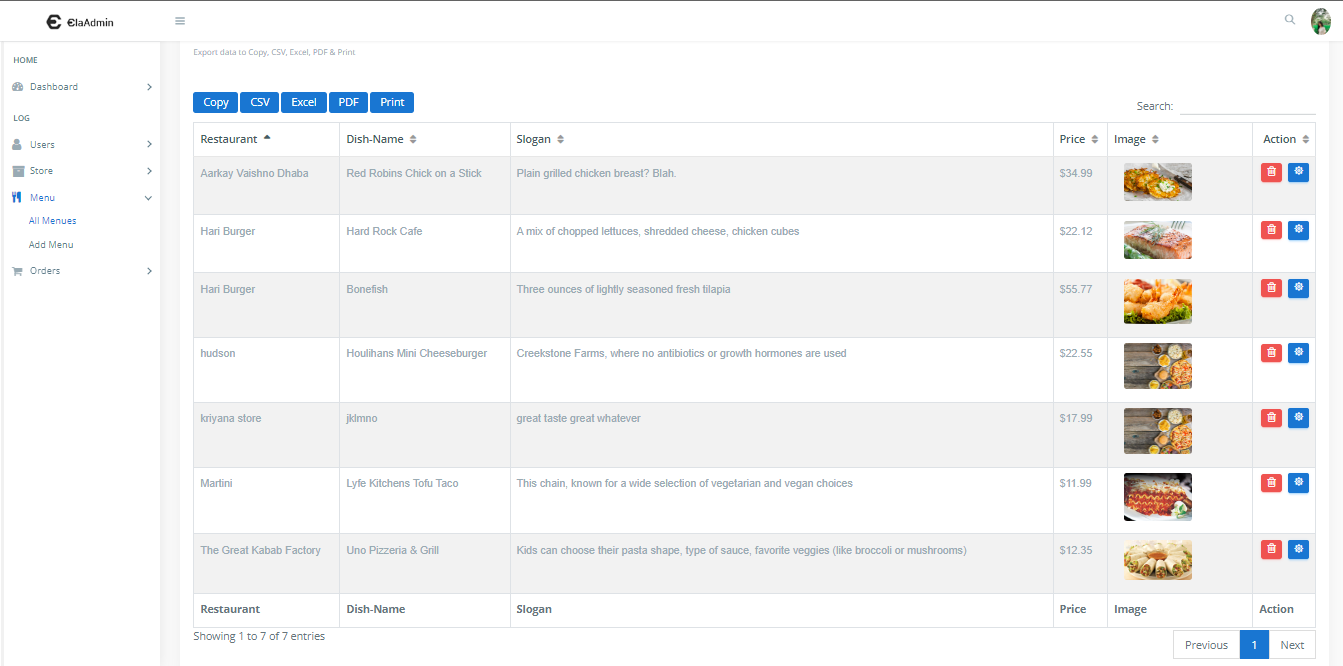
Hình 3. 12. Xuất danh sách những cửa hàng trên website



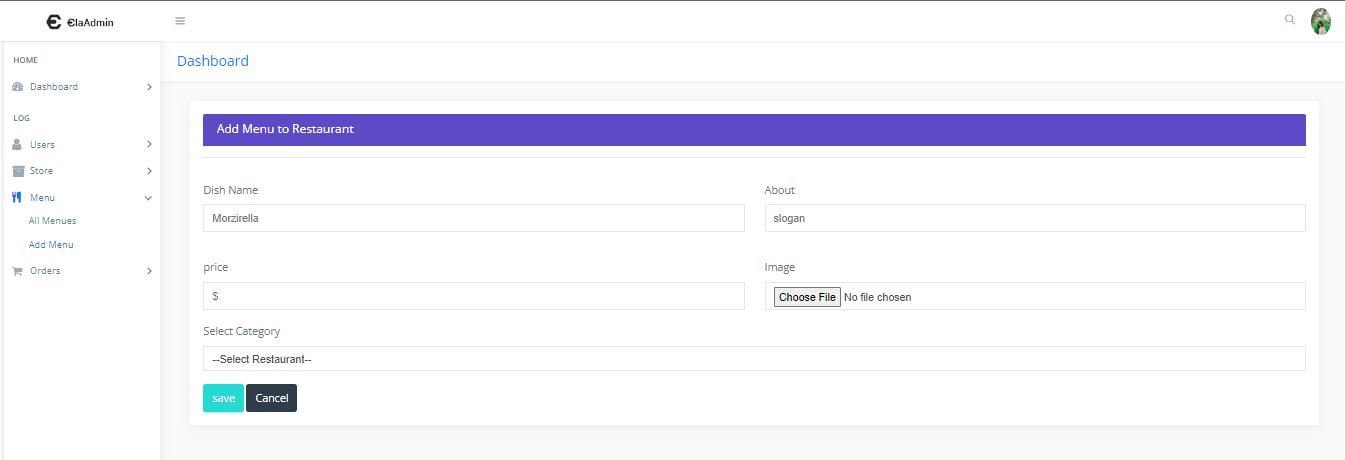
Hình 3. 13. Giao diện thêm danh mục món ăn



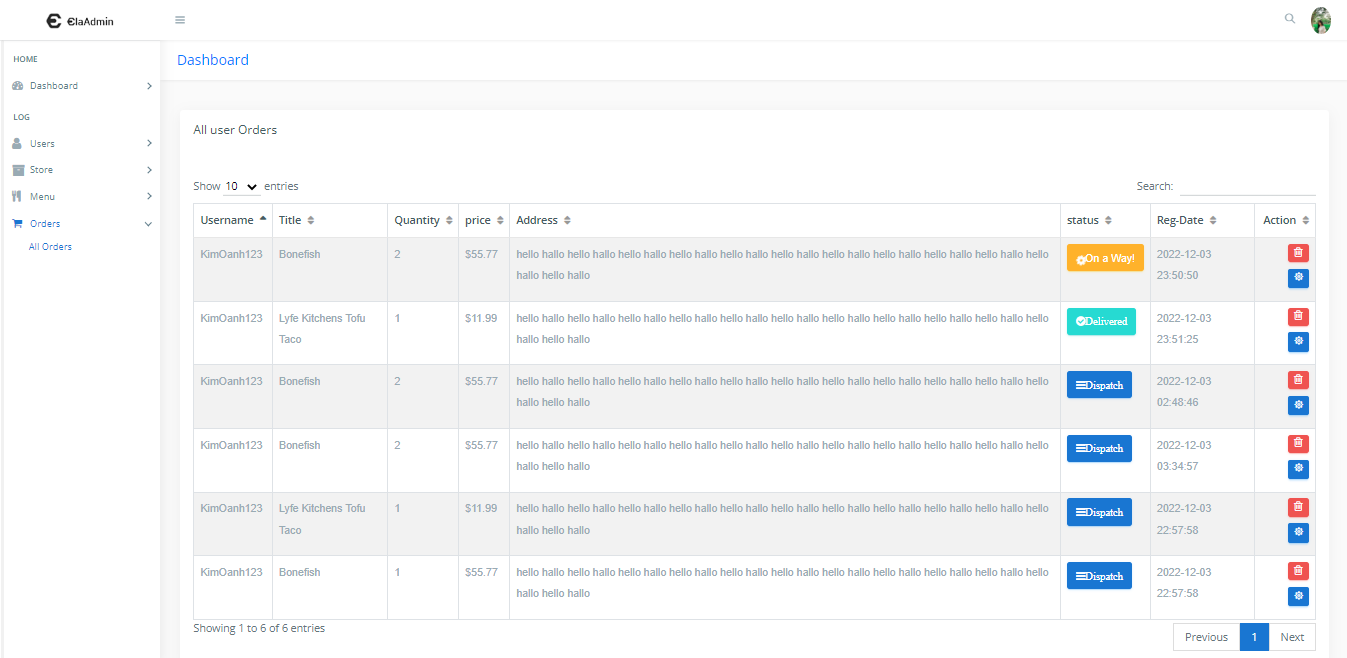
Hình 3. 14. Giao diện thêm nhà hàng



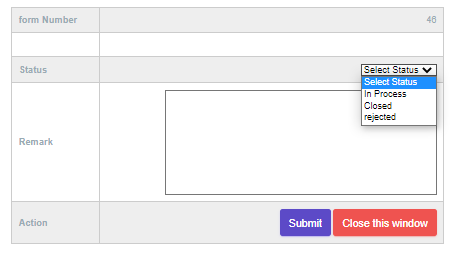
Hình 3. 15. Danh mục Menu



Hình 3. 16. Giao diện thêm món ăn mới



Hình 3. 17. Xem danh mục các món ăn đã đặt



Hình 3. 18. Form thay đổi trạng thái của đơn hàng

## 3.2. Kiểm thử tự động cho Quick Food

### 3.2.1. Sử dụng Cypress và Cucumber trong dự án

*Bước 1: Tạo một folder và cấu hình file package.json*

- Tạo một thư mục mới

- Chạy câu lệnh: *npm init*

*Bước 2: Cài đặt Cypress*

*npm install cypress – save- dev*

*Bước 3: Cài đặt Cucumber*

*npm install -- save-dev cypress-cucumber-preprocessor*

*Bước 4: Cấu hình file* ***index.js***

//For Cucumber Integration

const createEsbuildPlugin =

  require('@badeball/cypress-cucumber-preprocessor/esbuild').createEsbuildPlugin

const createBundler = require('@bahmutov/cypress-esbuild-preprocessor')

const nodePolyfills =

  require('@esbuild-plugins/node-modules-polyfill').NodeModulesPolyfillPlugin

const addCucumberPreprocessorPlugin =

  require('@badeball/cypress-cucumber-preprocessor').addCucumberPreprocessorPlugin

*module*.*exports* = async (*on*, *config*) => {

  await addCucumberPreprocessorPlugin(*on*, *config*) // to allow json to be produced

  // To use esBuild for the bundler when preprocessing

  on(

    'file:preprocessor',

    createBundler({

      plugins: [nodePolyfills(), createEsbuildPlugin(*config*)],

    })

  )

  return *config*

}

*Bước 5: Cấu hình* ***package.json***

{

  "name": "137",

  "version": "1.0.0",

  "description": "",

  "main": "index.js",

  "scripts": {

    "cy:open": "npx cypress open ",

    "cy:run": "yarn run cypress run --browser chrome -e \"TAGS=not @ignore\"",

    "cy:percy-regression": "percy exec -- cypress run --browser chrome -e \"TAGS=not @ignore\" "

  },

  "author": "",

  "license": "ISC",

  "devDependencies": {

    "@percy/cli": "^1.7.0",

    "@percy/cypress": "^3.1.2",

    "csvdata": "^2.0.0",

    "cypress": "^10.6.0",

    "cypress-failed-log": "^2.10.0",

    "cypress-grep": "^3.0.2",

    "husky": "^4.3.8",

    "multiple-cucumber-html-reporter": "^1.21.4",

    "webpack": "^5.73.0"

  },

  "cypress-cucumber-preprocessor": {

    "stepDefinitions": "cypress/e2e/\*\*/\*.{js,ts}",

    "json": {

      "enabled": true

    }

  },

  "dependencies": {

    "@badeball/cypress-cucumber-preprocessor": "^11.5.1",

    "@bahmutov/cypress-esbuild-preprocessor": "^2.1.3",

    "@esbuild-plugins/node-modules-polyfill": "^0.1.4",

    "buffer": "^6.0.3",

    "crypto": "^1.0.1",

    "del": "^7.0.0",

    "esbuild": "^0.15.1",

    "fast-csv": "^4.3.6",

    "json2csv": "^5.0.7",

    "objects-to-csv": "^1.3.6"

  }

}

Chạy các câu lệnh sau để cài đặt các thư viện:

*npm install -D @badeball/cypress-cucumber-preprocessor  
npm install -D @bahmutov/cypress-esbuild-preprocessor  
npm install -D @esbuild-plugins/node-modules-polyfill*

*Bước 6: Cấu hình* ***cypress.config.js***

const { defineConfig } = require("cypress");

*module*.*exports* = defineConfig({

  defaultCommandTimeout: 500,

  e2e: {

    // We've imported your old cypress plugins here.

    // You may want to clean this up later by importing these.

    setupNodeEvents(*on*, *config*) {

      //require('cypress-failed-log/on')(on)

      on('task', {

        log(*message*) {

          console.log(*message*)

          return null

        },

      })

      require('cypress-grep/src/plugin')(*config*);

      return require('./cypress/plugin/index.js')(on, *config*)

    },

    baseUrl: 'http://localhost/FoodAndDrink/',

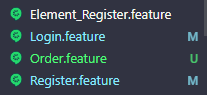
    specPattern:

      ["\*\*/\*.feature", "cypress/e2e/\*\*/\*.cy.{js,jsx,ts,tsx}"]

  },

});

*Bước 7: Viết file* ***\*\*\*.feature***

**

***Login.feature***

Feature: login functionality

  Scenario Outline: Check login is successful with valid credentials

   Given user is on login page and user is a user of nopCommerce

   When user enters userName as '<userName>' and password as '<password>'

   And clicks on login button

  Examples:

     | userName | password |

     | KimOanh123  | 123456 |

     | KimOanh123  | 111111 |

     | MasterFail  | 123456 |

     | MasterFail  | 111111|

***Register.feature***

Feature: Register

    Scenario: Check registration successful

        Given User is on register page

        When User input registration infomation

        And User click Register button

    Scenario: Check registration unsuccessful- Email Already exist

        Given User is on register page

        When User input registration infomation-enter email already

        And User click Register button

    Scenario: Check registration unsuccessful- User name Already exist

        Given User is on register page

        When User input registration infomation-enter user name already

        And User click Register button

***Order.feature***

Feature: Order after user login

  Scenario: Order now

   Given user is login successfully

   When user click order now

   And user check out and check your order

***Element\_Register.feature***

Feature: Search

    Scenario: Check if there is an input search box there

        Given user goes to Register form

        When user search by entering text

        Then user sees search box

    Scenario: Check if there is a title in Register Page

        Given user goes to Register form

        Then user sees Register header

    Scenario: Check the display of fields in the Your Personal Details form

        Given user goes to Register form

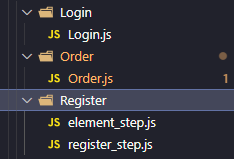
        Then user sees all fields of Your Personal Details form

    Scenario: Check the display of fields in the Company Details form

        Given user goes to Register form

        Then user sees all fields of Company Details form

*Bước 8: Viết các file* ***\*\*\*.js*** *cho các scenario trong file* ***\*\*\*.feature***

**

***Login.js***

import { Given, When, And } from "@badeball/cypress-cucumber-preprocessor";

import LoginPage from "../../pageoject/demoacc/login\_page.js";

Given("user is on login page and user is a user of nopCommerce", () => {

  LoginPage.visit\_login();

  cy.wait(2000);//wait 2s = 2000ms

  });

When('user enters userName as {string} and password as {string}',(*userName*,*password*) =>{

  LoginPage.fill\_enterinformation(*userName*,*password*);

  cy.wait(2000);

});

And("clicks on login button", () => {

  LoginPage.click\_login();

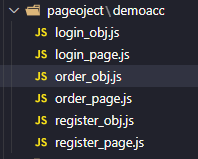
  cy.wait(2000);

});

*Bước 9: Chạy chương trình*

*npx cypress open*

### 3.2.2. Sử dụng Page Object Model (POM)



***login\_obj.js***

*module*.*exports* ={

    login:{

        email\_lbl:'Last Name (Legal)',

        password\_lbl:'Last Name (Legal)',

        login\_button:'buttton login',

        user\_txt:'[type="text"]',

        password\_txt:'[type="password"]',

        btnLogin: '#buttn'

    }

}

***login\_page.js***

export default class LoginPage{

    static visit\_login(){

        cy.visit(Cypress.config().baseUrl+'login.php');

    }

    static fill\_enterinformation(*userName*, *password*) {

        cy.get('[type="text"]').type(*userName*);

        cy.get('[type="password"]').type(*password*);

    }

    static click\_login(){

        cy.get('#buttn').click();

    }

}

***register\_obj.js***

*module*.*exports* ={

    register:{

        firstname\_lbl:'First Name (Legal)',

        lastname\_lbl:'Last Name (Legal)',

        email\_lbl:'Last Name (Legal)',

        country\_lbl:'Country',

        countryinfo\_lbl:'Please ensure your country of residence is accurate. If you want to convert your Demo Trading account to a Live Account in future, we will use this country as reference.',

        userName\_txt: ':nth-child(1) > #example-text-input',

        firstName\_txt:':nth-child(2) > #example-text-input',

        lastName\_txt:'#example-text-input-2',

        email\_txt: '#exampleInputEmail1',

        phoneNumber\_txt:'#example-tel-input-3',

        password\_txt:'#exampleInputPassword1',

        confirmPassword\_txt:'#exampleInputPassword2',

        deliveryAddress\_txt:'#exampleTextarea',

        btnRegister\_txt:'.col-sm-4 > p > .btn'

    }

}

***register\_page.js***

import {register} from '../demoacc/register\_obj.js';

import {randomNumber, randomUserName, randomEmail, randomFirstName,randomLastName,randomText} from '../../support/ramdomdata.js';

export default class RegisterPage{

    //----Action------------------

    static visit\_registerPage(){

        cy.visit(Cypress.config().baseUrl+'registration.php');

    }

    static fill\_Register() {

        let fill\_fields = {};

        fill\_fields.userName\_txt=randomUserName(10);

        fill\_fields.firstName\_txt=randomFirstName(10);

        fill\_fields.lastName\_txt=randomLastName(10);

        fill\_fields.email\_txt=randomEmail();

        fill\_fields.phoneNumber=randomNumber(9);

        fill\_fields.password\_txt=randomText(10);

        fill\_fields.confirmPassword\_txt=fill\_fields.password\_txt;

        fill\_fields.deliveryAddress\_txt = randomText(60);

        cy.get(register.userName\_txt).type(fill\_fields.userName\_txt);

        cy.get(register.firstName\_txt).type(fill\_fields.firstName\_txt);

        cy.get(register.lastName\_txt).type(fill\_fields.lastName\_txt);

        cy.get(register.email\_txt).type(fill\_fields.email\_txt);

        cy.get(register.phoneNumber\_txt).type("0813234188");

        // cy.get('#Newsletter').check();

        cy.get(register.password\_txt).type(fill\_fields.password\_txt);

        cy.get(register.confirmPassword\_txt).type(fill\_fields.confirmPassword\_txt);

        cy.get(register.deliveryAddress\_txt).type(fill\_fields.deliveryAddress\_txt);

        var output = '';

        var currentdate = **new** *Date*();

        var datetime = currentdate.getDate()+"-"+ currentdate.getMonth()+"-"+ currentdate.getFullYear() + "\_" + currentdate.getHours() + "-"+ currentdate.getMinutes() + "-" + currentdate.getSeconds();

        for (const key\_field of *Object*.keys(fill\_fields)) {

            output += key\_field + ': ' + fill\_fields[key\_field]+';'+'\n';

        }

        cy.writeFile('infoRegister/TestResult\_CreatedAccount\_'+datetime+'.csv', output, {

            encoding: 'ascii',

            flag: 'a+',

          })

    }

    static fill\_RegisterEmail() {

        let fill\_fields = {};

        fill\_fields.userName\_txt=randomUserName(10);

        fill\_fields.firstName\_txt=randomFirstName(10);

        fill\_fields.lastName\_txt=randomLastName(10);

        fill\_fields.email\_txt=randomEmail();

        fill\_fields.phoneNumber=randomNumber(9);

        fill\_fields.password\_txt=randomText(10);

        fill\_fields.confirmPassword\_txt=fill\_fields.password\_txt;

        fill\_fields.deliveryAddress\_txt = randomText(60);

        cy.get(register.userName\_txt).type(fill\_fields.userName\_txt);

        cy.get(register.firstName\_txt).type(fill\_fields.firstName\_txt);

        cy.get(register.lastName\_txt).type(fill\_fields.lastName\_txt);

        cy.get(register.email\_txt).type("ttkoanh.19it3@vku.udn.vn");

        cy.get(register.phoneNumber\_txt).type("0813234188");

        // cy.get('#Newsletter').check();

        cy.get(register.password\_txt).type(fill\_fields.password\_txt);

        cy.get(register.confirmPassword\_txt).type(fill\_fields.confirmPassword\_txt);

        cy.get(register.deliveryAddress\_txt).type(fill\_fields.deliveryAddress\_txt);

    }

    static fill\_RegisterUserName() {

        let fill\_fields = {};

        fill\_fields.userName\_txt=randomUserName(10);

        fill\_fields.firstName\_txt=randomFirstName(10);

        fill\_fields.lastName\_txt=randomLastName(10);

        fill\_fields.email\_txt=randomEmail();

        fill\_fields.phoneNumber=randomNumber(9);

        fill\_fields.password\_txt=randomText(10);

        fill\_fields.confirmPassword\_txt=fill\_fields.password\_txt;

        fill\_fields.deliveryAddress\_txt = randomText(60);

        cy.get(register.userName\_txt).type("NBeOan4");

        cy.get(register.firstName\_txt).type(fill\_fields.firstName\_txt);

        cy.get(register.lastName\_txt).type(fill\_fields.lastName\_txt);

        cy.get(register.email\_txt).type(fill\_fields.email\_txt);

        cy.get(register.phoneNumber\_txt).type("0813234188");

        // cy.get('#Newsletter').check();

        cy.get(register.password\_txt).type(fill\_fields.password\_txt);

        cy.get(register.confirmPassword\_txt).type(fill\_fields.confirmPassword\_txt);

        cy.get(register.deliveryAddress\_txt).type(fill\_fields.deliveryAddress\_txt);

}

    static click\_register(){

        cy.get(register.btnRegister\_txt).click();

    }

}

### 3.2.3. Multiple Cucumber HTML Reporter

*Bước 1: Cài đặt*

*npm install multiple-cucumber-html-reporter*

*npm install multiple-cucumber-html-reporter --save*

*hoặc npm install multiple-cucumber-html-reporter --save-dev*

*Bước 2: Viết file cucumber-html-report.js*

const report = require('multiple-cucumber-html-reporter');

report.generate({

    jsonDir: './cypress/reports/',

    reportPath: './cypress/reports/',

    metadata:{

        browser: {

            name: 'chrome',

            version: '103.0.5060.134'

        },

        device: 'Local test machine',

        platform: {

            name: 'ubuntu',

            version: '16.04'

        }

    },

    customData: {

        title: 'Run info',

        data: [

            {label: 'Project', value: 'Custom project'},

            {label: 'Release', value: '1.2.3'},

            {label: 'Cycle', value: 'B11221.34321'},

            {label: 'Execution Start Time', value: 'Dec 19th 2022, 02:31 PM EST'},

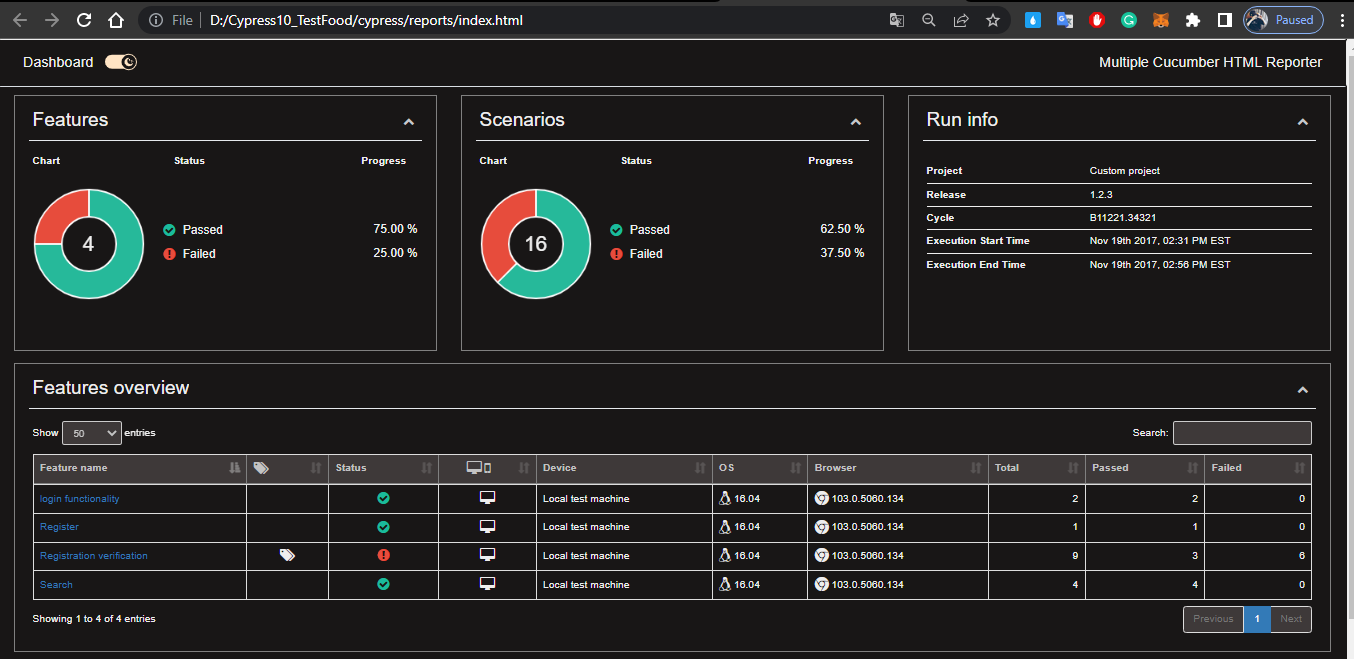
            {label: 'Execution End Time', value: 'Dec 19th 2022, 02:56 PM EST'}

        ]

    }

});

*Bước 3: Sau khi chạy câu lệnh* ***npx cypress run*** *sẽ nhận được kết quả sau*



### 3.2.4. Test UI bằng Percy io

*Bước 1: Cài đặt Percy*

Cài đặt mô-đun này cục bộ bằng lệnh sau:

*npm install multiple-cucumber-html-reporter*

Lưu vào phần phụ thuộc hoặc phần phụ thuộc nhà phát triển:

*npm install multiple-cucumber-html-reporter --save*

*hoặc npm install multiple-cucumber-html-reporter --save-dev*

*Bước 2: Trên trình duyệt percy.io đăng kí tài khoản và lấy token từ dự án được tạo*

*Bước 3: Export token vào dự án bằng câu lệnh:*

$env:PERCY\_TOKEN="7d3ce0918d2812a91240793b386bbd3a5dbf8392d4480f4c010a0b43fe258081"

*Bước 2: Sử dụng câu lệnh:* ***cy.percySnapshot*** *vào trong phần code test*

Given("User is on page Login", () => {

  cy.percySnapshot()

  LoginPage.visit\_login();

  cy.percySnapshot()

  cy.wait(1000);//wait 1s = 2000ms

  });

When('User enters {string} and {string}',(*userName*,*password*) =>{

  cy.percySnapshot()

  LoginPage.fill\_enterinformation(*userName*,*password*);

  cy.percySnapshot()

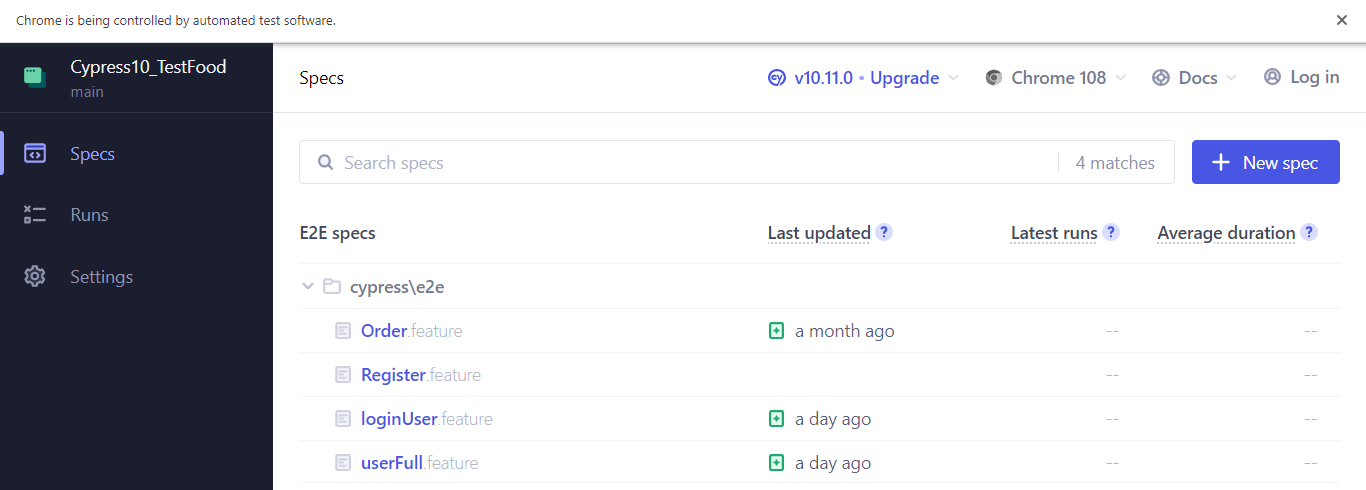
});

And("User click login button", () => {

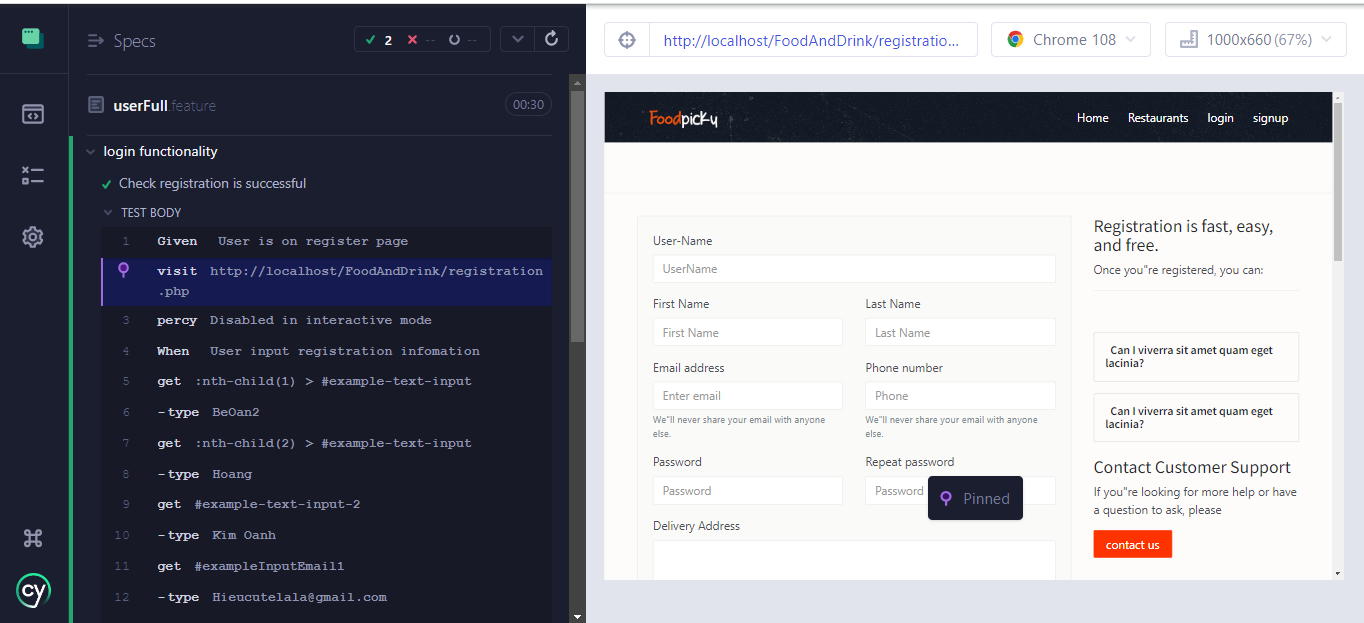
  LoginPage.click\_login();

});

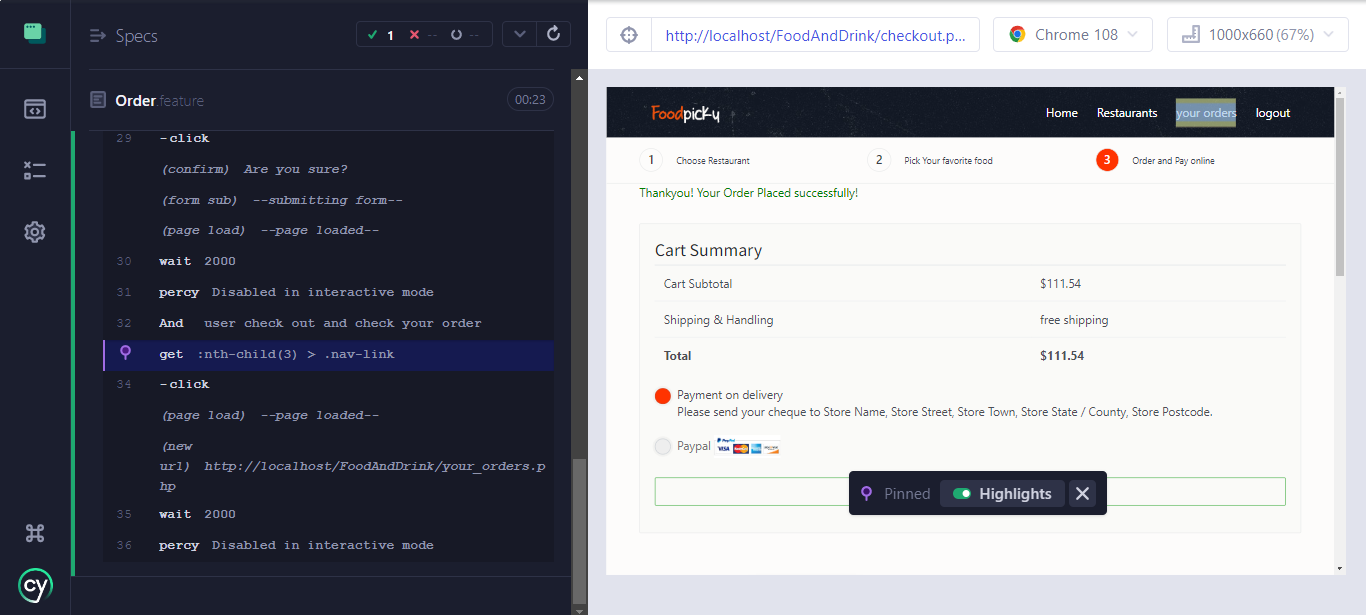
### 3.2.5. Kết quả của kiểm thử tự động



Hình 3. 19. Giao diện cypress trong dự án



Hình 3. 20. Test 2e2 tính năng chính người dùng sử dụng trên website



Hình 3. 21. Test Order Page

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết quả đạt được

***Về kiến thức học tập***

Để xây dựng một hệ thống website nói chung và website đặt đồ ăn nói riêng là một việc không hề đơn giản, nó đòi hỏi chúng ta phải xây dựng theo các giai đoạn: khảo sát, phân tích và thiết kế hệ thống, nghiên cứu tổng quan các công cụ lập trình, thiết kế phần mềm và kiểm thử. Ngày nay chúng ta đang đứng trước nhiều lựa chọn về công nghệ mới vậy nên một hệ thống học tập và làm việc online hiệu quả là điều không thể thiếu. Thông qua nghiên cứu và xây dựng đề tài này, chúng em đã củng cố, học hỏi thêm được nhiều thứ như:

- Củng cố kiến thức về lập trình web bằng PHP, Javascript, NodeJs, Socket.io.... Biết thêm được nhiều kiến thức về xây dựng một hệ thống web thời gian thực cũng như phân tích hệ thống của một ứng dụng web.

- Củng cố kiến thức về kiểm thử phần mềm đặc biệt là kiểm thử tự động. Biết cách áp dụng những công nghệ mới trong kiểm thử tự động

- Cũng cố kiến thức về phân tích thiết kế hệ thống UML, vẽ và thao tác thành thạo các loại biểu đồ cơ bản như: biểu đồ use-case, biểu đồ lớp, biểu đồ tuần tự… Đặc tả được use-case.

***Về phần mềm***

Qua dự án trên nhóm chúng em đạt được một số kết quả như sau:

- Xây dựng được hệ thống website với giao diện phù hợp, chưa được tối ưu hoàn toàn nhưng đủ bắt mắt với người dùng.

- Ứng dụng có một số chức năng căn bản của một ứng dụng đặt đồ ăn

- Ứng dụng giúp người dùng tối ưu hóa dữ liệu người dùng, dễ dàng thao tác dễ dàng sử dụng.

## Hạn chế

  Dự án Xây dựng website Quick Food và kiểm thử tự động cho Quick Food tuy chưa được hoàn thiện hoàn toàn nhưng nó cũng đã có những chức năng căn bản của một ứng dụng hệ thông website cần có. Do thời gian hoàn thiện dự án còn hạn chế nên một số chức năng khác nhóm chúng em chưa kịp hoàn thiện trong thời gian trước khi bảo vệ, nên dự án vẫn có một số chức năng chưa kịp hoàn thiện mong thầy cô giáo châm chước nhiều. Song nếu có thời gian nhiều hơn nhóm chúng em sẽ hoàn thiện và bảo trì dự án tốt hơn từ đó có thể đưa ứng dụng vào thực tiễn.

## Hướng phát triển

Để dự án có thể hoạt động hiệu quả hơn cần có dữ liệu, từ đó tăng tính bảo mật hệ thống. Dự án sẽ được hoàn thiện trong thời gian tới và sẽ được ứng dụng vào thực tế nếu nó mang tính thuyết phục cao. Vì vậy trong khuôn khổ đồ án này chúng em chỉ có thể đưa ra những chức năng cơ bản, mô phỏng một số yêu cầu trong thực tế. Đồ án này của chúng em còn rất nhiều thiếu sót, rất mong nhận được sự góp ý từ quý thầy cô để chúng em hoàn thiện đồ án này hơn.

Em xin chân thành cảm ơn!

# TÀI LIỆU THAM KHẢO

**[1]** [1:56AM 13/11/2022] Giới thiệu và cách cài đặt Cypress

***[online]*** *https://www.cypress.io/*

***[online]***[*https://docs.cypress.io/guides/getting-started/installing-cypress*](https://docs.cypress.io/guides/getting-started/installing-cypress)

***[online]***[*https://docs.cypress.io/api/commands/clear*](https://docs.cypress.io/api/commands/clear)

***[online]*** *https://www.npmjs.com/package/cypress*

**[2]** [3:00AM 13/11/2022] Giới thiệu và cách cài đặt Cucumber

***[online]*** https://cucumber.io/docs/installation/javascript/

**[3]** [4:00AM 13/11/2022] Cách cài đặt Percy

***[online]*** [*https://www.npmjs.com/package/@percy/cypress*](https://www.npmjs.com/package/@percy/cypress)

***[online]*** *https://docs.percy.io/docs/cypress*

**[3]** [4:30AM 13/11/2022] Cách cài đặt Multiple Cucumber HTML Reporter

***[online]*** *https://www.npmjs.com/package/multiple-cucumber-html-reporter*