|  |  |
| --- | --- |
|  |  |
| Bùi Việt Hoàng | **BỘ CÔNG THƯƠNG**  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **---------------------------------------** |
|  |
| ĐA, KLTN ĐẠI HỌC  NGÀNH CÔNG NGHỆ THÔNG TIN |
|  |
| **XÂY DỰNG WEBSITE BÁN NƯỚC HOA SỬ DỤNG**  **ASP.NET CORE WEB API VÀ VUE.JS** |
|  |
|  |
| **CBHD:** **ThS. Vũ Duy Giang** |
| NGÀNH CÔNG NGHỆ THÔNG TIN | **Sinh viên: Bùi Việt Hoàng** |
| **Mã số sinh viên: *2020605607*** |
|  |
|  |
|  |
| Hà Nội – Năm 2024 |
|  |

# **MỤC LỤC**

[**MỤC LỤC** 2](#_Toc167095707)

[Danh mục các ký hiệu chữ viết tắt 4](#_Toc167095708)

[Danh mục các bảng 4](#_Toc167095709)

[Danh mục các hình vẽ ,đồ thị 4](#_Toc167095710)

[MỞ ĐẦU 5](#_Toc167095711)

[Chương 1. TỔNG QUAN VỀ WEBSITE THƯƠNG MẠI ĐIỆN TỬ 9](#_Toc167095712)

[1.1 Giới thiệu chung về thương mại điện tử 9](#_Toc167095713)

[1.1.1 Khái niệm về thương mại điện tử 9](#_Toc167095714)

[1.1.2 Điều kiện để phát triển thương mại điện tử 10](#_Toc167095715)

[1.1.3 Thuận lợi của thương mại điện tử 10](#_Toc167095716)

[1.1.4 Khó khăn của thương mại điện tử 10](#_Toc167095717)

[1.1.5 Mô hình thương mại điện tử phổ biến 10](#_Toc167095718)

[Chương 2. CÔNG NGHỆ SỬ DỤNG ASP.NET CORE WEB API VÀ VUE JS 11](#_Toc167095719)

[2.1 Giới thiệu về ASP.NET core Web API và Vue.js 11](#_Toc167095720)

[2.1.1 ASP.NET Core Web API 11](#_Toc167095721)

[2.1.2 Vue.js: 12](#_Toc167095722)

[2.1.3 Kiến trúc hệ thống 12](#_Toc167095723)

[2.1.4 Xây dựng Backend với ASP.NET Core Web API: 14](#_Toc167095724)

[2.1.5 Phát triển giao diện ggười dùng với Vue.js: 14](#_Toc167095725)

[2.1.6 Tích hợp giữa ASP.NET Core Web API và Vue.js: 16](#_Toc167095726)

[2.2 Giới thiệu về chatbot AI Dialogflow 17](#_Toc167095727)

[2.2.1 Chatbot AI Dialogflow 17](#_Toc167095728)

[2.2.2 Kịch bản xây dựng cho chatbot 17](#_Toc167095729)

[2.2.3 Hướng dẫn tạo và tích hợp chatbot AI Dialogflow vào website 21](#_Toc167095730)

[Chương 3.XÂY DỰNG WEBSITE BÁN NƯỚC HOA 24](#_Toc167095731)

[3.1 Phân tích đề tài 24](#_Toc167095732)

[3.1.1 Yêu cầu hệ thống 24](#_Toc167095733)

[3.1.2 Yêu cầu chức năng 24](#_Toc167095734)

[3.2 Thiết kế mô hình cơ sở dữ liệu 25](#_Toc167095735)

[3.2.1 Các thực thể cơ sở dữ liệu 25](#_Toc167095736)

[3.2.2 Mô hình ERD 25](#_Toc167095737)

[3.3 Biểu đồ usecase 26](#_Toc167095738)

[3.3.1 Biểu đồ usecase người dùng 26](#_Toc167095739)

[3.3.2 Biểu đồ usecase người quản trị hệ thống 27](#_Toc167095740)

[3.4 Mô tả chi tiết usecase 28](#_Toc167095741)

[3.4.1 Mô tả use case “Đăng nhập “ 28](#_Toc167095742)

[3.4.2 Mô tả usecase “Xem sản phẩm theo thương hiệu” 29](#_Toc167095743)

[3.4.3 Mô tả usecase “Xem giỏ hàng” 30](#_Toc167095744)

[3.4.4 Mô tả usecase “Đăng kí” 31](#_Toc167095745)

[3.4.5 Mô tả usecase “Đặt hàng” 32](#_Toc167095746)

[3.4.6 Mô tả usecase “Quên mật khẩu “ 33](#_Toc167095747)

[3.4.7 Mô tả usecase “Bảo trì sản phẩm” 34](#_Toc167095748)

[3.4.8 Mô tả usecase “Xem thống kê” 37](#_Toc167095749)

[3.4.9 Mô tả usecase “Xem chi tiết sản phẩm” 38](#_Toc167095750)

[3.4.10 Mô tả usecase “Thêm sản phẩm vào giỏ hàng” 39](#_Toc167095751)

[3.4.11 Mô tả usecase “Quản lý giỏ hàng” 39](#_Toc167095752)

[3.4.12 Mô tả usecase “Tìm kiếm sản phẩm theo tên” 41](#_Toc167095753)

[3.4.13 Mô tả usecase “Bảo trì thương hiệu” 42](#_Toc167095754)

[3.4.14 Mô tả usecase “Quản lý thông tin tài khoản” 44](#_Toc167095755)

[3.4.15 Mô tả usecase “Quản lý tài khoản” 45](#_Toc167095756)

[3.4.16 Mô tả usecase “Quản lý đơn hàng” 46](#_Toc167095757)

[3.5 Biểu đồ trình tự 49](#_Toc167095758)

[3.5.1 Biểu đồ trình tự usecase “Đăng nhập“ 49](#_Toc167095759)

[3.5.2 Biểu đồ trình tự usecase “Xem sản phẩm theo thương hiệu“ 49](#_Toc167095760)

[3.5.3 Biểu đồ trình tự usecase “Xem giỏ hàng“ 50](#_Toc167095761)

[3.5.4 Biểu đồ trình tự usecase “Đăng kí“ 51](#_Toc167095762)

[3.5.5 Biểu đồ trình tự usecase Đặt hàng 52](#_Toc167095763)

[3.5.6 Biểu đồ trình tự usecase ‘Quên mật khẩu’ 53](#_Toc167095764)

[3.5.7 Biểu đồ trình tự usecase ‘Bảo trì sản phẩm” 53](#_Toc167095765)

[3.5.8 Biểu đồ trình tự usecase “Xem thống kê” 55](#_Toc167095766)

[3.5.9 Biểu đồ trình tự “Xem chi tiết sản phẩm“ 55](#_Toc167095767)

[3.5.10 Biểu đồ trình tự “ Thêm sản phẩm vào giỏ hàng“ 55](#_Toc167095768)

[3.5.11 Biểu đồ trình tự “Quản lý giỏ hàng“ 56](#_Toc167095769)

[3.5.12 Biểu đồ trình tự “Tìm kiếm sản phẩm theo tên” 57](#_Toc167095770)

[3.5.13 Biểu đồ trình tự usecase “bảo trị thương hiệu” 58](#_Toc167095771)

[3.5.14 Biểu đồ trình tự “Quản lý thông tin tài khoản” 60](#_Toc167095772)

[3.5.15 Biểu đồ trình tự “Quản lý tài khoản” 60](#_Toc167095773)

[3.5.16 Biểu đồ trình tự “Quản lý đơn hàng“ 61](#_Toc167095774)

[3.6 Cài đặt chương trình 62](#_Toc167095775)

[3.6.1 Cài đặt ASP.NET Core Web API 62](#_Toc167095776)

[3.6.2 Cài đặt Vue.js 64](#_Toc167095777)

[3.7 Giao diện chương trình 65](#_Toc167095778)

[3.7.1 Giao diện người dùng 65](#_Toc167095779)

[3.7.2 Giao diện người quản trị 68](#_Toc167095780)

[3.8 Kiểm thử chương trình 70](#_Toc167095781)

[3.8.1 Chức năng đăng nhập 70](#_Toc167095782)

[3.8 2 Chức năng bảo trì sản phẩm 71](#_Toc167095783)

[3.8.3 Chức năng tìm kiếm sản phẩm theo tên 73](#_Toc167095784)

[3.8.4 Chức năng xem sản phẩm theo danh mục 74](#_Toc167095785)

[3.8.5 Chức năng quản lý giỏ hàng 75](#_Toc167095786)

[KẾT LUẬN 78](#_Toc167095787)

[TÀI LIỆU THAM KHẢO 78](#_Toc167095788)

[PHỤ LỤC 79](#_Toc167095789)

# Danh mục các ký hiệu chữ viết tắt

|  |  |
| --- | --- |
| API | Application Programming Interface |
| B2B | Business To Business |
| B2C | Business To Consumer |
| B2G | Business To Government |
| CORS | Cross-Origin Resource Sharing |
| C2C | Consumer To Consumer |
| G2C | Government To Consumer |
| TMĐT | Thương Mại Điện Tử |

# Danh mục các bảng

[Bảng 2. 1Kịch bản cho chatbot 17](#_Toc167050733)

[Bảng 3. 1 Kiểm thử chức năng đăng nhập 69](#_Toc167050618)

[Bảng 3. 2 Kiểm thử chức năng bảo trì sản phẩm 70](#_Toc167050619)

[Bảng 3. 3 Kiểm thử chức năng tìm kiếm sản phẩm theo tên 73](#_Toc167050620)

[Bảng 3. 4 Kiểm thử chức năng xem sản phẩm theo danh mục 73](#_Toc167050621)

[Bảng 3. 5 Kiểm thử chức năng quản lý giỏ hàng 74](#_Toc167050622)

# Danh mục các hình vẽ ,đồ thị

[Hình 2. 1 Tương tác giữa Vue js và Asp.net core web API 13](#_Toc166922069)

[Hình 2. 2Dữ liệu lấy được qua phương thức GET từ bảng PRODUCT 16](#_Toc166922070)

[Hình 2. 3 Màn hình tạo Agent 21](#_Toc166922071)

[Hình 2. 4 Màn hình tạo Intents và Entites 22](#_Toc166922072)

[Hình 2. 5 Màn hình lấy mã sinh ra để tích hợp vào Website 23](#_Toc166922073)

[Hình 3.1 Tạo dự án ASP.net core Web API 24](#_Toc166922017)

[Hình 3. 2 Thiết lập CROS 24](#_Toc166922018)

[Hình 3. 3 Sử dụng CROS sau khi thiết lập 25](#_Toc166922019)

[Hình 3. 4 Tạo dự án Vue js 25](#_Toc166922020)

[Hình 3. 5 Cài đặt các thư viện sử dụng trong Vue js 25](#_Toc166922021)

[Hình 3. 6 Các gói cài đặt trong dữ án Vue js 25](#_Toc166922022)

[Hình 3. 7 Mô hình ERD 27](#_Toc166922023)

[Hình 3. 8 Biểu đồ usecase phía người dùng 28](#_Toc166922024)

[Hình 3. 9 Biểu đồ usecase phía quản trị viên 29](#_Toc166922025)

[Hình 3. 10 Biểu đồ trình tự usecase đăng nhập 50](#_Toc166922026)

[Hình 3. 11 Biểu đồ trình tự usecase xem sản phẩm theo thương hiệu 50](#_Toc166922027)

[Hình 3. 12 Biểu đồ trình tự usecase xem giỏ hàng 51](#_Toc166922028)

[Hình 3. 13 Biểu đồ trình tự usecase đăng kí 52](#_Toc166922029)

[Hình 3. 14 Biểu đồ trình tự usecase đặt hàng 52](#_Toc166922030)

[Hình 3. 15 Biểu đồ trình tự usecase quên mật khẩu 53](#_Toc166922031)

[Hình 3. 16 Biểu đồ trình tự usecase bảo trì sản phẩm 55](#_Toc166922032)

[Hình 3. 17 Biểu đồ trình tự usecase xem thống kê 55](#_Toc166922033)

[Hình 3. 18 Biểu đồ trình tự usecase xem chi tiết sản phẩm 55](#_Toc166922034)

[Hình 3. 19 Biểu đồ trình tự usecase thêm sản phẩm vào giỏ hàng 56](#_Toc166922035)

[Hình 3. 20 Biểu đồ trình tự usecase quản lý giỏ hàng 57](#_Toc166922036)

[Hình 3. 21 Biểu đồ trình tự usecase tìm kiếm sản phẩm theo tên 58](#_Toc166922037)

[Hình 3. 22 Biểu đồ trình tự usecase bảo trì thương hiệu 60](#_Toc166922038)

[Hình 3. 23 Biểu đồ trình tự usecase quản lý thông tin tài khoản 60](#_Toc166922039)

[Hình 3. 24 Biểu đồ trình tự usecase quản lý tài khoản người dùng 61](#_Toc166922040)

[Hình 3. 25 Biểu đồ trình tự usecase quản lý đơn hàng 62](#_Toc166922041)

[Hình 3. 26 Giao diện trang chủ 63](#_Toc166922042)

[Hình 3. 27 Giao diện trang cửa hàng 63](#_Toc166922043)

[Hình 3. 28 Giao diện trang đăng nhập 64](#_Toc166922044)

[Hình 3. 29 Giao diện trang đăng ký 64](#_Toc166922045)

[Hình 3. 30 Giao diện trang giỏ hàng 65](#_Toc166922046)

[Hình 3. 31 Giao diện trang kiểm tra trước khi đặt hàng 65](#_Toc166922047)

[Hình 3. 32 Giao diện trang quản lý sản phẩm 66](#_Toc166922048)

[Hình 3. 33 Giao diện quản lý tài khoản người dùng 66](#_Toc166922049)

[Hình 3.34 Giao diện quản lý voucher 67](#_Toc166922050)

[Hình 3.35 Giao diện thống kê doanh thu 67](#_Toc166922051)

# MỞ ĐẦU

1. Lý do chọn đề tài

Ngày nay song song với sự phát triển không ngừng của công nghệ thông tin, tin học đã thâm nhập xâu vào hầu hết tất cả các lĩnh vực của xã hội và đặc biệt với nhiều ứng dụng to lớn trong hệ thống quản lý của các công ty. Và hình thức dịch vụ trực tuyến cũng ngày càng phổ biến, giữ vị trí quan trọng trong cuộc sống. Các website bán hàng ra đời với mục đích đáp ứng được nhu cầu về tính tiện dụng và tiết kiệm khi mua sắm của khách hàng. Ngoài ra việc xây dựng các website còn là cách giúp các doanh nghiệp tiếp cận khách hàng tốt nhất để gia tăng doanh số bán hàng, tăng độ nhận diện của thương hiệu trên thị trường.

Một doanh nghiệp muốn thực hiện tin học hóa việc mua sắm thông qua website. Đây là trang website cung cấp tất cả những sản phẩm về nước hoa giúp việc mua sắm trở nên dễ dàng hơn bao giờ hết. Có thể cung cấp một loạt các sản phẩm từ khắp nơi, người mua có thể lựa chọn và đặt mua từ bất kỳ đâu và vào bất kỳ thời điểm nào. Người bán không cần phải thuê hoặc duy trì một cửa hàng vật lý, giúp họ tiết kiệm được các chi phí như thuê mặt bằng, tiện ích, và nhân viên. Họ cũng có khả năng tiếp cận một số lượng lớn khách hàng tiềm năng qua mô hình kinh doanh trực tuyến.

Vì vậy, em đã thực hiện đề tài “ Xây dựng website bán nước hoa sử dụng ASP.net Core Web API và Vue JS”.

1. Mục tiêu của đề tài

* Tìm hiểu dự án thực tế để có điều kiện trải nghiệm
* Nắm bắt được quy trình xây dựng hệ thống bằng ASP.NET core.
* Xác định được đúng mục đích, mục tiêu của dự án.
* Lập kế hoạch quản lý dự án.
* Xây dựng tài liệu phác thảo dự án một cách tường minh, cụ thể.
* Lập được bản kế hoạch.
* Biết cách sử dụng và áp dụng các phương pháp, công cụ để lập tiến độ thực hiện và thực thi dự án.
* Dự đoán được các rủi ro có thể xảy ra; phân tích, lập kế hoạch phòng ngừa, kiểm soát và xử lý được các rủi ro.
* Hoàn thiện dự án thành công, biết cách kết thúc dự án.

1. Đối tượng và phạm vi nghiên cứu
   1. Đối tượng

* Công nghệ ASP.NET Core Web API: Tập trung nghiên cứu các kỹ thuật, phương pháp và ứng dụng của ASP.NET Core Web API trong việc xây dựng các dịch vụ backend, quản lý dữ liệu sản phẩm, đơn hàng, khách hàng, và tích hợp với hệ thống cơ sở dữ liệu.
* Framework Vue.js: Nghiên cứu các khả năng và lợi ích của Vue.js trong việc phát triển giao diện người dùng, tạo ra các trang web tương tác, linh hoạt, và dễ sử dụng.
* Các thành phần và mô-đun của website bán nước hoa:
* Quản lý sản phẩm: Các chức năng thêm, sửa, xóa, hiển thị thông tin sản phẩm (nước hoa) như tên, nhãn hiệu, giá cả, mô tả, và hình ảnh.
* Quản lý người dùng: Đăng ký, đăng nhập, và quản lý thông tin tài khoản người dùng.
* Quản lý đơn hàng: Tạo đơn hàng, theo dõi trạng thái đơn hàng, và xử lý thanh toán.
* Tích hợp thanh toán trực tuyến: Nghiên cứu và triển khai các phương thức thanh toán trực tuyến an toàn và tiện lợi.
* Trải nghiệm người dùng : Nghiên cứu các yếu tố và phương pháp thiết kế giao diện người dùng để đảm bảo tính thẩm mỹ, dễ sử dụng, và tối ưu hóa trải nghiệm mua sắm trực tuyến.
  1. Phạm vi nghiên cứu
  + Khảo sát và phân tích yêu cầu quy trình quản lý và mua bán hàng hoá.
  + Phân tích thiết kế hệ thống.
  + Thiết kế cơ sở dữ liệu.
  + Lập trình backend API.
  + Lập trình giao diện và kết nối tới API.
  + Có kế hoạch và đánh giá kiểm thử hệ thống.

1. Ý nghĩa khoa học và thực tiễn của đề tài

4.1 Ý nghĩa khoa học

* Nghiên cứu và ứng dụng công nghệ hiện đại: Đề tài này liên quan đến việc sử dụng hai công nghệ tiên tiến là ASP.NET Core Web API và Vue.js, đóng góp vào việc nghiên cứu và ứng dụng các công nghệ mới trong lĩnh vực phát triển web. ASP.NET Core là một framework mạnh mẽ và hiệu quả cho việc phát triển API, trong khi Vue.js là một framework front-end phổ biến cho việc xây dựng giao diện người dùng linh hoạt và tương tác.
* Kết hợp giữa backend và frontend: Việc tích hợp ASP.NET Core Web API với Vue.js thể hiện sự kết hợp giữa backend và frontend, giúp nghiên cứu và phát triển các mô hình kiến trúc phần mềm hiện đại, đặc biệt là các kiến trúc tách biệt giữa server và client.
* Phát triển kỹ năng lập trình: Đề tài này giúp người thực hiện nâng cao kỹ năng lập trình và hiểu biết sâu sắc về các ngôn ngữ và framework liên quan, như C# (cho ASP.NET Core) và JavaScript (cho Vue.js).

4.2 Ý nghĩa thực tiễn

* Ứng dụng thực tế: Xây dựng một website bán nước hoa có thể áp dụng trực tiếp vào kinh doanh, giúp các doanh nghiệp bán lẻ nước hoa mở rộng thị trường, tiếp cận khách hàng tiềm năng thông qua nền tảng trực tuyến.
* Tăng cường trải nghiệm người dùng: Sử dụng Vue.js để phát triển giao diện người dùng giúp tạo ra các trang web mượt mà, tương tác tốt, giúp cải thiện trải nghiệm người dùng và tăng khả năng giữ chân khách hàng.
* Hiệu quả trong quản lý và vận hành: ASP.NET Core Web API cung cấp các dịch vụ backend mạnh mẽ, hỗ trợ tốt cho việc quản lý dữ liệu sản phẩm, đơn hàng, khách hàng một cách hiệu quả và an toàn.
* Tối ưu hóa SEO và marketing: Một website bán hàng chuyên nghiệp với giao diện và chức năng tốt sẽ giúp cải thiện SEO (tối ưu hóa công cụ tìm kiếm), từ đó tăng cường khả năng tiếp cận và thu hút khách hàng thông qua các chiến lược marketing trực tuyến.
* Đáp ứng nhu cầu thị trường: Với sự phát triển của thương mại điện tử, nhu cầu mua sắm trực tuyến ngày càng tăng. Đề tài này giúp đáp ứng nhu cầu đó, cung cấp một giải pháp bán hàng hiệu quả, linh hoạt và dễ dàng mở rộng.

# Chương 1. TỔNG QUAN VỀ WEBSITE THƯƠNG MẠI ĐIỆN TỬ

## Giới thiệu chung về thương mại điện tử

* + 1. Khái niệm về thương mại điện tử

Thương mại điện tử là việc tiến hành một phần hay toàn bộ hoạt động thương mại bằng những phương tiện điện tử. Thương mại điện tử vẫn mang bản chất như các hoạt động thương mại truyền thống. Tuy nhiên, thông qua các phương tiện điện tử mới, các hoạt động thương mại được thực hiện nhanh hơn, hiệu quả hơn, giúp tiết kiệm chi phí và mở rộng không gian kinhdoanh.

Internet tạo ra thương mại điện tử nên có thể hiểu một cách đơn giản hơn thương mại điện tử là việc thực hiện các giao dịch thương mại thông qua Internet.

Thương mại điện tử đang ngày càng phát triển và dần trở thành xu hướng tất yếu của thị trường. Nhiều doanh nghiệp đã dần chuyển sang lĩnh vực thương mại điện tử bởi những lợi ích mà mô hình này đem lại. Công nghệ tiên tiến hiện nay giúp doanh nghiệp biến Website của mình thành những siêu thị hàng hóa trên Internet, biến người mua thực sự trở thành những người chủ với toàn quyền trong việc chọn lựa sản phẩm, tìm kiếm thông tin, so sánh giá cả, đặt mua hàng, ký kết hợp đồng với hệ thống tính toán tiền tự động, rõ ràng, trung thực.

* + 1. Điều kiện để phát triển thương mại điện tử
* Đòi hỏi trình độ của người tham gia môi trường kinh doanh:

- Phía người bán

- Phía người mua

* Đòi hỏi cơ sở vật chất công nghệ

- Cơ sở của cá nhân

- Hệ thống quốc gia, vùng miền

* + 1. Thuận lợi của thương mại điện tử

- Toàn cầu hóa

- Tăng tính cạnh tranh và chất lượng dịch vụ

- Đáp ứng nhanh nhu cầu người tiêu dùng

- Giảm chi phí giao dịch , chi phí tồn kho, hạ giá thành sản phẩm

- Tiện lợi và giảm chi phí cho khách hàng

- Thuận lợi cho xã hội

* + 1. Khó khăn của thương mại điện tử

Một số dạng hiểm họa chủ yếu thường gặp trong các giao dịch điện tử nói chung và đặc biệt đối với các giao dịch TMĐT nói riêng

1. Virus máy tính và phần mềm độc hại - *Malwares:* Worm, Spyware, Trojan horses, Logic bomb…
2. Các loại tội phạm điện tử: SPAM, DoS, Phising, Pharming
3. Hacker và Cracker
   * 1. Mô hình thương mại điện tử phổ biến

Dựa vào chủ thể của thương mại điện tử, có thể phân chia thương mại điện tử ra các loại hình phổ biến như sau:

* Giao dịch giữa doanh nghiệp với doanh nghiệp - B2B
* Giao dịch giữa doanh nghiệp với khách hàng - B2C
* Giao dịch giữa doanh nghiệp với cơ quan nhà nước - B2G
* Giao dịch trực tiếp giữa các cá nhân với nhau - C2C
* Giao dịch giữa cơ quan nhà nước với cá nhân - G2C

Kết luận chương 1

Trong chương 1 của báo cáo đã giới thiệu về thương mại điện tử, một lĩnh vực đang phát triển mạnh mẽ và có ảnh hưởng sâu rộng đến cách thức kinh doanh và tiêu dùng trên toàn cầu. Chương này đã trình bày khái niệm cơ bản, lịch sử phát triển, các mô hình thương mại điện tử chủ yếu, cùng với lợi ích và thách thức mà nó mang lại.

# Chương 2. CÔNG NGHỆ SỬ DỤNG ASP.NET CORE WEB API VÀ VUE JS

## Giới thiệu về ASP.NET core Web API và Vue.js

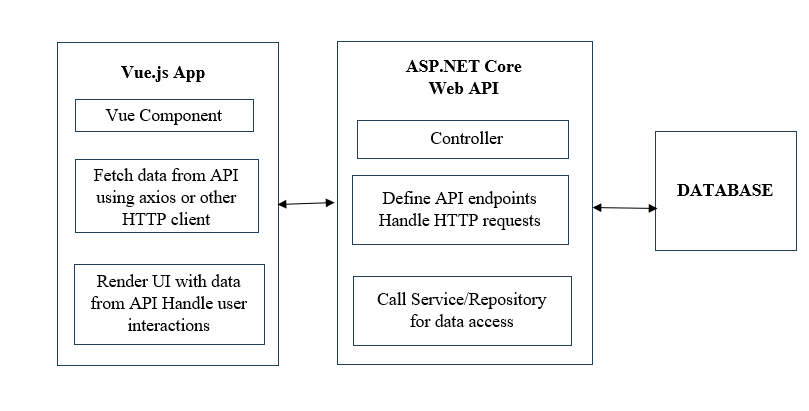
2.1.1 ASP.NET Core Web API

ASP.NET Core Web API là một framework phát triển dành cho xây dựng các dịch vụ web HTTP dựa trên .NET Core. Dưới đây là một số điểm nổi bật:

* Hiệu suất cao: ASP.NET Core được tối ưu hóa để đạt hiệu suất cao và khả năng mở rộng, giúp ứng dụng xử lý lượng truy cập lớn một cách hiệu quả.
* Hỗ trợ đa nền tảng: .NET Core là một framework đa nền tảng, cho phép triển khai ứng dụng trên nhiều hệ điều hành như Windows, Linux và macOS.
* Bảo mật mạnh mẽ: ASP.NET Core cung cấp các tính năng bảo mật tích hợp như xác thực và phân quyền, giúp bảo vệ dữ liệu và người dùng của ứng dụng.
* Tiêu chuẩn RESTful: ASP.NET Core Web API thúc đẩy việc thiết kế các API theo kiến trúc RESTful, giúp tạo ra các dịch vụ linh hoạt và dễ dàng tích hợp với các ứng dụng khác.
  + 1. Vue.js:

Vue.js là một framework JavaScript phổ biến được sử dụng để xây dựng giao diện người dùng phía client. Dưới đây là những lý do chính cho việc lựa chọn Vue.js:

* Dễ dàng học và sử dụng: Vue.js có cú pháp đơn giản và dễ hiểu, cho phép các nhà phát triển mới có thể nhanh chóng bắt đầu và phát triển ứng dụng.
* Hiệu suất cao: Vue.js được thiết kế để có hiệu suất cao và khả năng tương tác tốt trên các ứng dụng web động, giúp cải thiện trải nghiệm người dùng.
* Cộng đồng lớn: Vue.js có một cộng đồng lớn và tích cực, cung cấp nhiều tài liệu, thư viện và công cụ hỗ trợ cho việc phát triển.
* Tích hợp dễ dàng: Vue.js có thể dễ dàng tích hợp vào các dự án có sẵn mà không cần phải thay đổi cơ sở hạ tầng hiện có.
  + 1. Lý do lựa chọn:
* Tính linh hoạt: Kết hợp ASP.NET Core Web API và Vue.js cung cấp một nền tảng linh hoạt cho việc phát triển ứng dụng web, cho phép tách biệt phát triển backend và frontend.
* Hiệu suất và tương tác: Sự kết hợp của ASP.NET Core và Vue.js đảm bảo hiệu suất cao và trải nghiệm người dùng tốt trên các ứng dụng web động.
* Cộng đồng và tài liệu: Cả hai công nghệ đều có cộng đồng lớn và tài liệu phong phú, giúp dễ dàng tìm kiếm hỗ trợ và giải quyết vấn đề.
  + 1. **Kiến trúc hệ thống**
       1. Backend (ASP.NET Core Web API):
* Sử dụng ASP.NET Core Web API để xây dựng backend cho ứng dụng.
* Backend xử lý các yêu cầu từ client, thực hiện các thao tác đối với cơ sở dữ liệu và trả về dữ liệu cho client thông qua các API endpoints.
* ASP.NET Core Web API được sử dụng vì hiệu suất cao, khả năng mở rộng và tính bảo mật.
  + - 1. Frontend (Vue.js):
* Sử dụng Vue.js để phát triển giao diện người dùng phía client của ứng dụng.
* Frontend là nơi người dùng tương tác trực tiếp với ứng dụng, hiển thị dữ liệu từ backend và gửi yêu cầu đến backend để cập nhật dữ liệu hoặc thực hiện các thao tác khác.



Hình 2. 1 Tương tác giữa Vue js và Asp.net core web API

Trong hình 2.1 trên:

* Frontend (Vue.js) là phần giao diện người dùng mà người dùng trực tiếp tương tác thông qua trình duyệt web.
* Backend (ASP.NET Core Web API) là phần xử lý logic của ứng dụng, nhận yêu cầu từ frontend, thực hiện các thao tác đối với cơ sở dữ liệu và trả về dữ liệu cho frontend thông qua các API endpoints.
* Cơ sở dữ liệu lưu trữ dữ liệu của ứng dụng, được truy cập và cập nhật bởi backend thông qua các thao tác đối với cơ sở dữ liệu.
  + 1. Xây dựng Backend với ASP.NET Core Web API:
       1. Tạo dự án ASP.NET Core Web API:
* Bắt đầu bằng việc tạo một dự án mới trong Visual Studio sử dụng mẫu "ASP.NET Core Web API".
* Dự án này sẽ chứa tất cả các logic backend của ứng dụng, bao gồm các controller để xử lý các yêu cầu từ client.
  + - 1. Xác định Model và Controller:
* Định nghĩa các model để đại diện cho dữ liệu trong ứng dụng, ví dụ như Model cho sản phẩm nước hoa.
* Tạo các controller để xử lý các yêu cầu HTTP như GET, POST, PUT và DELETE liên quan đến các model này.
  + - 1. Triển khai các Endpoint API:

Xác định và triển khai các endpoint API cho các tác vụ như tạo, đọc, cập nhật và xóa dữ liệu.

Ví dụ:

GET /api/products: Lấy danh sách tất cả các sản phẩm nước hoa.

GET /api/products/{id}: Lấy thông tin chi tiết của một sản phẩm cụ thể.

POST /api/products: Tạo mới một sản phẩm nước hoa.

PUT /api/products/{id}: Cập nhật thông tin của một sản phẩm cụ thể.

DELETE /api/products/{id}: Xóa một sản phẩm nước hoa.

* + - 1. Xử lý Logic Nghiệp vụ:

Trong các phương thức của controller, thực hiện logic nghiệp vụ như truy xuất dữ liệu từ cơ sở dữ liệu, xử lý yêu cầu và trả về kết quả tương ứng.

Sử dụng các service và repository để tách biệt logic nghiệp vụ và truy cập dữ liệu.

### Phát triển giao diện ggười dùng với Vue.js:

Cấu trúc thư mục dự án Vue.js:

* src/: Thư mục chứa mã nguồn của ứng dụng Vue.js.
* assets/: Chứa các tài nguyên như hình ảnh, biểu tượng.
* components/: Chứa các thành phần Vue.js tái sử dụng như header, footer, sản phẩm item.
* views/: Chứa các trang (components lớn hơn) như trang chủ, trang chi tiết sản phẩm.
* store/: Chứa các file liên quan đến Vuex store, bao gồm các modules, getters, mutations, actions.
* router/: Chứa các tệp liên quan đến Vue Router, như tệp cấu hình router và các route của ứng dụng.
* plugins/: Chứa các plugins bổ sung cho ứng dụng Vue.js, như Vue-chart.js.

Các thành phần chính trong dự án Vue.js:

* Header.vue: Thành phần hiển thị header của trang web.
* Footer.vue: Thành phần hiển thị footer của trang web.
* ProductList.vue: Thành phần hiển thị danh sách các sản phẩm nước hoa.
* ProductDetail.vue: Thành phần hiển thị thông tin chi tiết của một sản phẩm nước hoa.
* Cart.vue: Thành phần hiển thị giỏ hàng và các chức năng liên quan đến giỏ hàng.
* Checkout.vue: Thành phần hiển thị trang thanh toán với các thông tin đơn hàng và hình thức thanh toán.
* Vuex Store: Sử dụng Vuex để quản lý trạng thái toàn cầu của ứng dụng, bao gồm thông tin sản phẩm, giỏ hàng, người dùng đăng nhập.
* Vue Router: Sử dụng Vue Router để quản lý định tuyến trong ứng dụng, xác định các route cho trang chủ, trang chi tiết sản phẩm, giỏ hàng, thanh toán và các trang khác.
* Vue-chart.js: Sử dụng Vue-chart.js để hiển thị biểu đồ thống kê, ví dụ như biểu đồ doanh số bán hàng.

### Tích hợp giữa ASP.NET Core Web API và Vue.js:

Gửi Yêu cầu từ Frontend đến Backend:

Trong frontend Vue.js, khi người dùng thực hiện một hành động như tải trang, click vào một nút, hoặc điền vào một biểu mẫu, ứng dụng sẽ tạo ra một yêu cầu HTTP đến backend.

Ví dụ, khi người dùng truy cập vào trang danh sách sản phẩm, frontend sẽ gửi một yêu cầu GET /api/products đến backend để lấy danh sách sản phẩm.

Xử lý Yêu cầu trong Backend:

Backend sử dụng ASP.NET Core Web API để xử lý yêu cầu từ frontend.

Một controller được kích hoạt để xử lý yêu cầu tương ứng, ví dụ: ProductsController xử lý yêu cầu liên quan đến sản phẩm.

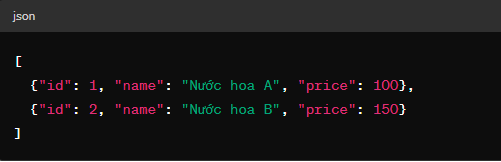
Trong controller, các logic nghiệp vụ được thực thi, chẳng hạn như truy xuất dữ liệu từ cơ sở dữ liệu, xử lý yêu cầu và trả về kết quả.

Trả về Dữ liệu từ Backend đến Frontend:

Sau khi xử lý yêu cầu, backend trả về dữ liệu cho frontend thông qua các API endpoint.

Dữ liệu thường được trả về dưới dạng JSON.

Ví dụ, sau khi xử lý yêu cầu GET /api/products, backend sẽ trả về danh sách sản phẩm dưới dạng JSON như sau:



Hình 2. 2Dữ liệu lấy được qua phương thức GET từ bảng PRODUCT

Xử lý Dữ liệu trong Frontend:

Frontend nhận dữ liệu từ backend và sử dụng nó để cập nhật giao diện người dùng.

Thông thường, frontend sử dụng Vuex để quản lý trạng thái ứng dụng và lưu trữ dữ liệu trả về từ backend.

## Giới thiệu về chatbot AI Dialogflow

* + 1. Chatbot AI Dialogflow

Dialogflow là một nền tảng tích hợp giúp phát triển các chatbot, voicebot và các hệ thống hội thoại thông minh có khả năng hiểu và phản hồi ngôn ngữ tự nhiên của người dùng.Được phát triển bởi Google và tích hợp sâu với các dịch vụ khác của Google Cloud.

Các thành phần cơ bản của Dialogflow:

* Agent: Vai trò và cách tạo một agent trong Dialogflow.
* Intents: Cách hoạt động và tầm quan trọng của intents trong việc xác định ý định của người dùng.
* Entities: Định nghĩa và ví dụ về entities.
* Contexts: Cách sử dụng contexts để duy trì trạng thái hội thoại.
* Fulfillment: Cách sử dụng fulfillment để kết nối với các dịch vụ bên ngoài và xử lý logic tùy chỉnh.
  + 1. Kịch bản xây dựng cho chatbot

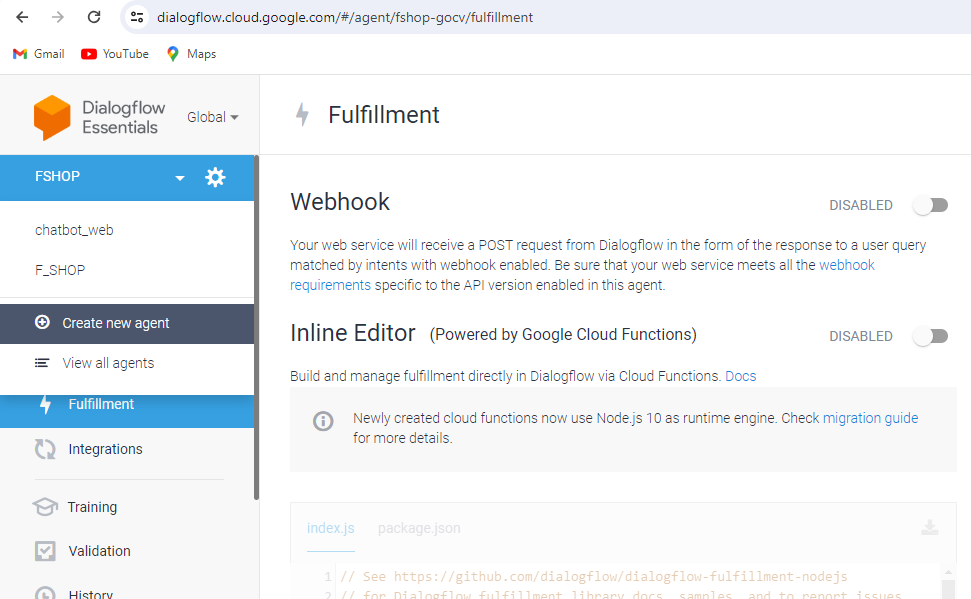
Bảng 2. 1Kịch bản cho chatbot

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Enitities |  | Intents | Câu hỏi | Câu trả lời |
| eChaoHoi | Xin chào;hello;hi;chào shop | iChaoHoi iChao  Hoi | Xin chào,  Hello,hi , chào shop, | Rất vui được gặp bạn |
| eCuaHang | Cửa hàng;trực tiếp; | iCuaHang | Ngoài bán hàng online trên website thì bên shop có cửa hàng bán hàng trực tiếp không  Bên bạn có bán trực tiếp tại của hàng không.  Shop có cửa hàng bán trực tiếp không | Hiện tại bên mình có 1 cửa hàng tại Tầng 7, 19A Cộng Hòa, Tòa Nhà Scetpa, Phường 12, Quận Tân Bình, TP. Hồ Chí Minh |
| eThuongHieu | Thương hiệu;hãng sản xuất;hãng, | iThuongHieu | Bên shop bán nước hoa của những hãng sản xuất nào.  Bên bạn có những thương hiệu nước hoa nào.  Cửa hàng có những thương hiệu nước hoa nào. | Bên cửa hàng bán nước hoa của những hàng nhữ là :   * GUCCI * VALENTINO * NARCISO RODRIGUEZ * BURBERRY * VERSACE * JO MALONE |
| eTheLoai | Thể loại; danh mục;loại nước hoa | iTheLoai | Bên shop có những thể loại nước hoa như nào .  Bên cửa hàng có những danh mục nước hoa nào . | Bên cửa hàng có những thể loại nước hoa là :  -Nước hoa nam  -Nước hoa nữ.  -Nước hoa unisex(dành cho cả nam và nữ) |
| eGia | Giá;số tiền; | iGia | Giá của những sản phẩm nước hoa bên shop giao động ở mức nào | Bên cửa hàng có bán rất nhiều loại nước hoa và giá cả cũng rất đa dạng và hợp lý .  Quý khách có thể truy cập vào trang cửa hàng của trang web để lọc những sản phẩm về giá phù hợp với bản thân. |
| eGiamGia | Giảm giá;khuyến mãi;ưu đãi | iGiamGia | * Bên cửa hàng có thường xuyên có chương trình khuyến mãi cho các sản phẩm không. * Cửa hàng có chương trình khuyến mãi không.   Bên bạn có những chương trình giảm giá không. | Tất nhiên , vào các dịp lễ và cũng để tri ân khách hàng bên cửa hàng có cũng nhiều chương trình giảm giá cho các sản phẩm bên cửa hàng. |
| eDoiTra | Đổi trả;hoàn trả; | iDoiTra | Khi đặt mua sản phẩm của bên shop mà lúc nhận hàng không ưng ý thì có thể hoàn trả không.  Chính sách đổi trả của bên shop như thế nào | Tất nhiên , bên cửa hàng có hỗ trợ khách hàng hoàn trả nếu sản phẩm không vừa ý với điều kiện là hoàn trả sau 5 ngày nhận hàng và sản phẩm vẫn chưa bị mở lắp. |
| eCuaHang\_ThoiGian(cấp 2) | Thời gian;mở cửa | iCuaHang\_ThoiGian | Cửa hàng trực tiếp của shop mở cửa vào thời gian nào .  Thời gian mở cửa tại cửa hàng là mấy giờ | Bên cửa hàng mở cửa vào lúc 7 h sáng và đóng cửa lúc 20 h tối tất cả các ngày trong tuần kể cả ngày nghỉ lễ. |
| ePhuHop | Sản phẩm phù hợp;loại nước hoa phù hợp;phù hợp; | iPhuHop | Có cách nào để tôi biết liệu hương thơm của sản phẩm có phù hợp với da của tôi không? | Chúng tôi khuyến khích bạn thử sản phẩm trên một phần nhỏ của da trước khi sử dụng nó toàn bộ, để đảm bảo không gây kích ứng da. |
| eHuongDan | Hướng dẫn;sử dung;cách dùng; | iHuongDan | Sản phẩm này có hướng dẫn sử dụng kèm theo không?  Khi mua sản phẩm có kèm hướng dẫn sử dụng không | Có, mỗi sản phẩm đều có hướng dẫn sử dụng và bảo quản khi bạn truy cập vào chi tiết sẩn phẩm đó. Nếu bạn cần thêm thông tin, vui lòng liên hệ với chúng tôi. |
| eMauThu | Mẫu thử; test thử;sử dụng thử;thử | iMauThu | Tôi có thể nhận được mẫu thử trước khi quyết định mua không?  Shop có thể cho tôi sử dụng thử trước khi mua hàng không | Rất tiếc, hiện tại chúng tôi không cung cấp mẫu thử. Tuy nhiên, chúng tôi cam đoan chất lượng của sản phẩm và chính sách đổi trả nếu bạn không hài lòng. |
| eChinhHang | Chính hãng; | iChinhHang | Có cách nào để kiểm tra tính chính hãng của sản phẩm không?  Làm sao tôi biết được sản phẩm của shop có chính hãng không | Chúng tôi cam kết cung cấp sản phẩm chính hãng. Bạn có thể kiểm tra bằng cách kiểm tra mã vạch trên sản phẩm hoặc kiểm tra chứng nhận của nhà sản xuất. |
| eQuaTang | Quà tặng;món quà; | iQuaTang | Nước hoa này có thể làm quà tặng được không? | Vâng , các sản phẩm nước hoa đều rất thích hợp để bạn tặng người thân hay bạn bè. |

* + 1. Hướng dẫn tạo và tích hợp chatbot AI Dialogflow vào website

Bước 1: Tạo Agent trên Dialogflow

Đăng nhập vào Dialogflow Console:Truy cập Dialogflow Console và đăng nhập bằng tài khoản Google. Nhấp vào nút "Create Agent".Điền tên Agent, chọn ngôn ngữ và múi giờ phù hợp.Nhấn "Create" để tạo Agent.



Hình 2. 3 Màn hình tạo Agent

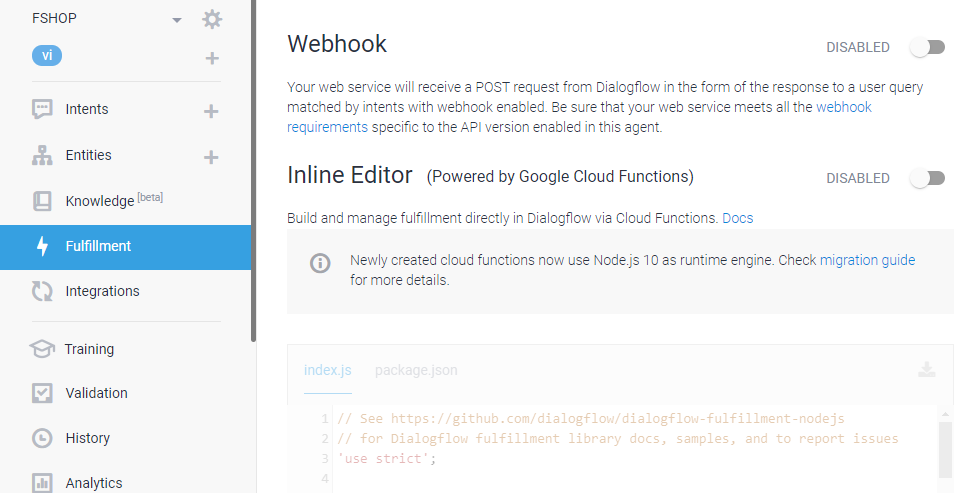
Bước 2: Tạo Intents và Entities

* Tạo Intents:

Intents xác định mục đích của người dùng. Nhấp vào “Intents” trong menu bên trái và nhấn “Create Intent”.Đặt tên cho Intent và thêm các câu mẫu (Training Phrases) mà người dùng có thể nói hoặc nhập.Định nghĩa phản hồi (Responses) mà chatbot sẽ trả lời khi Intent này được kích hoạt.

* Tạo Entities:

Entities giúp trích xuất thông tin cụ thể từ câu nói của người dùng. Nhấp vào “Entities” trong menu bên trái và nhấn “Create Entity”.Đặt tên cho Entity và định nghĩa các giá trị mà Entity này có thể nhận.



Hình 2. 4 Màn hình tạo Intents và Entites

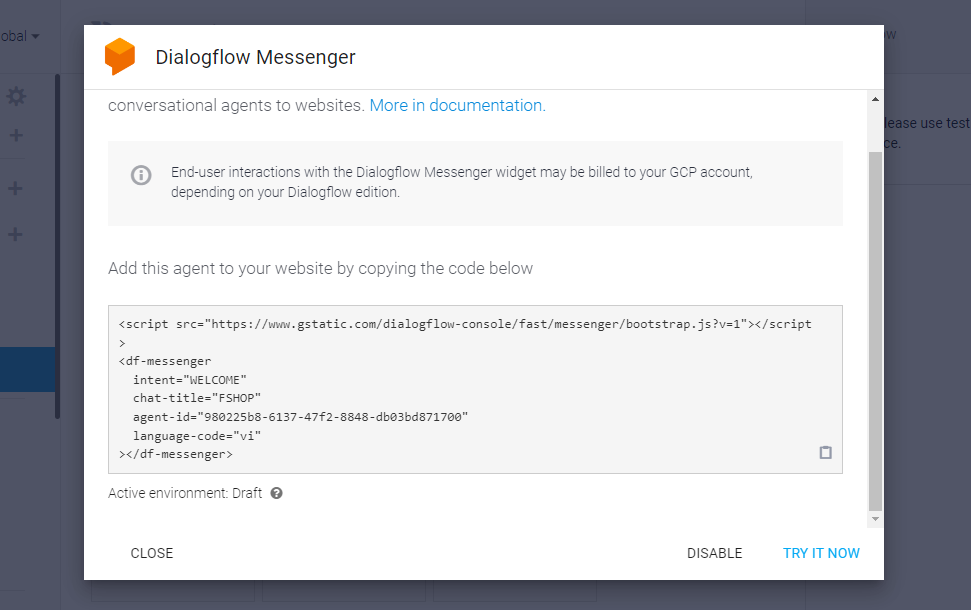
Bước 3: Tích hợp với Website

Lấy mã tích hợp từ Dialogflow:

Trong Dialogflow Console, đi đến phần “Integrations”.Chọn “Dialogflow Messenger” và bật nó lên.Sao chép đoạn mã nhúng được cung cấp.

Thêm mã vào Website:

Mở mã nguồn của website của bạn.Dán đoạn mã nhúng từ Dialogflow vào vị trí mong muốn trong mã HTML của trang web, thường là trong phần <body>.



Hình 2. 5 Màn hình lấy mã sinh ra để tích hợp vào Website

Kết luận chương 2

Trong chương 2 của báo cáo đã đi sâu vào giới thiệu ba công nghệ chính được sử dụng trong việc phát triển website hiện đại: ASP.NET Core Web API, Vue.js và Chatbot Dialogflow.

Đầu tiên là tìm hiểu về ASP.NET Core Web API, một framework mạnh mẽ giúp xây dựng các dịch vụ web linh hoạt và hiệu quả. ASP.NET Core cung cấp một nền tảng vững chắc cho việc phát triển các API mạnh mẽ, có khả năng mở rộng và bảo mật cao.

Tiếp theo, Vue.js được giới thiệu như một framework JavaScript tiến bộ, nổi bật với tính linh hoạt và dễ dàng tích hợp. Vue.js cho phép xây dựng giao diện người dùng tương tác và năng động, hỗ trợ mạnh mẽ cho việc phát triển ứng dụng một cách nhanh chóng và hiệu quả.

Cuối cùng là về Chatbot Dialogflow, một công cụ phát triển chatbot của Google, giúp tạo ra các chatbot thông minh và có khả năng xử lý ngôn ngữ tự nhiên. Dialogflow hỗ trợ việc tích hợp chatbot vào nhiều nền tảng khác nhau, mang lại trải nghiệm tương tác người dùng tự nhiên và hiệu quả.

# Chương 3.XÂY DỰNG WEBSITE BÁN NƯỚC HOA

## Phân tích đề tài

* + 1. Yêu cầu hệ thống
* Đáp ứng được các nhu cầu của việc quản lí một cửa hàng bán nước hoa.
* Tìm kiếm thông tin nhanh và lọc sản phẩm.
* Giao diện đơn giản, dễ sử dụng.
* Lưu trữ thông tin hàng hóa, người dùng.
* Chạy ổn định trên các trình duyệt web: Chrome, Cốc Cốc, …
* Trang web cần phải được bảo mật thông tin khách hàng để đảm bảo thông tin cá nhân của khách hàng được bảo vệ an toàn.
  + 1. Yêu cầu chức năng

Chức năng của quản trị

* Dễ dàng quản lý thông tin loại sản phẩm, sản phẩm, nhãn hàng, khách hàng, hóa đơn bán hàng.
* Dễ dàng thêm xóa sửa sản phẩm, nhãn hàng.
* Đối với tài khoản khách hàng thì quản trị viên có thể khóa tài khoản nếu khác hàng có vấn đề.

Chức năng của người dùng

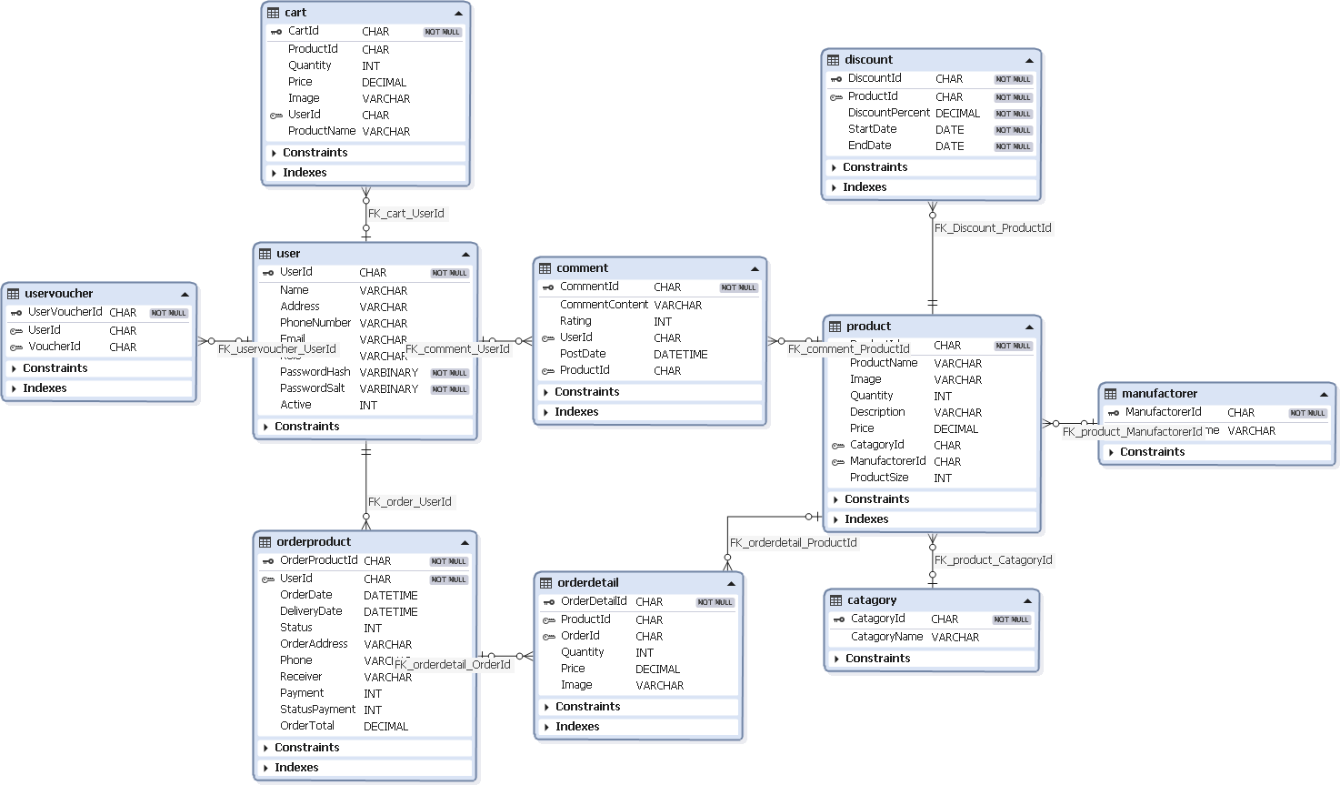
* Có thể tìm kiếm thông tin sản phẩm một cách dễ dàng.
* Xem thông tin chi tiết cho 1 sản phẩm.
* Có thể thêm sản phẩm vào giỏ hàng và đặt hàng.
* Thanh toán bằng hình thức trả tiền lúc nhận hàng hoặc online.

Yêu cầu phi chức năng

* Giao diện đẹp dễ dùng
* Ổn định, xử lý nhanh.
* Hỗ trợ nhiều người dùng.
* Bố cục rõ ràng.

## Thiết kế mô hình cơ sở dữ liệu

* + 1. Các thực thể cơ sở dữ liệu
* Thực thể 1:Hãng sản xuất (Manufactorer): Lưu thông tin nhà sản xuất sản phẩm.
* Thực thể 2: Loại sản phẩm (Categories): Lưu thông tin loại sản phẩm.
* Thực thể 3: Sản phẩm (Products): Lưu thôg tin sản phẩm.
* Thực thể 4:Đánh giá sản phẩm (Comment): Lưu thông tin đánh giá sản phẩm của khách hàng.
* Thực thể 5: Tài khoản (Users): Lưu thông tin tài khoản đăng ký.
* Thực thể 6:Đơn hàng (Order): Lưu thông tin đơn hàng mà người dùng đặt mua.
* Thực thể 7: Giỏ hàng (ShoppingCarts): Lưu thông tin sản phẩm khi khách hàng thêm hàng hóa vào giỏ hàng
* Thực thể 8: Hóa đơn (OrderDetail):Lưu thông tin chi tiết của một đơn hàng.
* Thực thể 9 :Giảm giá(Discount) :Lưu thông tin của các sản phẩm giảm giá.
* Thưc thể 10 :Voucher giảm giá (UserVoucher):Lưu thông tin của các voucher giảm giá.
  + 1. Mô hình ERD



Hình 3. 1Mô hình ERD

## Biểu đồ usecase

* + 1. Biểu đồ usecase người dùng



Hình 3. 2 Biểu đồ usecase phía người dùng

* + 1. Biểu đồ usecase người quản trị hệ thống



Hình 3.3 Biểu đồ usecase phía quản trị viên

## Mô tả chi tiết usecase

* + 1. Mô tả use case “Đăng nhập “

1. Tên use case: đăng nhập

2. Mô tả vắn tắt: use case này cho phép người dùng đăng nhập vào trang web

3. Luồng sự kiện:

a. Luồng cơ bản:

i. Use case này bắt đầu khi người dùng kích vào “Login”. Hệ thống sẽ hiển thị lên màn hình giao diện đăng nhập bao gồm:Tên đăng nhập, mật khẩu.

ii. Khi khách hàng nhập xong thông tin. Rồi nhấn vào nút “Đăng nhập”, hệ thống sẽ kiểm tra thông tin từ bảng USER và hiển thị giao diện tài khoản. Use case kết thúc.

b. Luồng rẽ nhánh:

i. Tại bước 2 trong luồng cơ bản nếu nhập sai tên đăng nhập hoặc mật khẩu, hệ thống sẽ báo “Thông tin đăng nhập không hợp lệ.” và use case kết thúc.

ii. Tại bất kì thời điểm nào trong quá trình thực hiện ca sủ dụng nếu không kết nối được với CSDL thì hệ thống sẽ hiển thị thông báo lỗi và ca sử dụng kết thúc.

4. Các yêu cầu đặc biệt: không.

5. Tiền điều kiện: không.

6. Hậu điều kiện: không.

7. Điều kiện mở rộng: không.

* + 1. Mô tả usecase “Xem sản phẩm theo thương hiệu”

1.Tên use case : Xem sản phẩm theo thương hiệu

2. Mô tả vắn tắt : Use case này cho phép khách hàng xem được các sản phẩm theo tên các thương hiệu .

3 . Luồng sự kiện :

3.1 : Luồng cơ bản :

*1*. Use case bắt đầu khi khách hàng kích vào mục thương hiệu . Hệ thống sẽ lấy tên các thương hiệu trong bảng MANUFACTORER hiển thị lên màn hình .

*2*.Khi khách hàng kích chọn vào một thương hiệu .Hệ thống sẽ lấy thông tin của các sản phẩm gồm tên mã sản phẩm , tên sản phẩm,ảnh minh hoa, mô tả , giá,thông tin,hướng dẫn sử dụng sản phẩm trong bảng PRODUCT tương ứng với tên thương hiệu đó hiển thị lên màn hình và use case kết thúc.

3.2 : luồng rẽ nhánh:

*1.* Tại bất kỳ thời điểm nào nếu use case không kết nối được với cơ sở dữ liệu thì hệ thống sẽ báo lỗi và use case kết thúc.

4.Các yêu cầu đặc biệt : không có

5. Tiền điều kiện : không có

6. Hậu điều kiện : không có

7. Điểm mở rộng : Không có

* + 1. Mô tả usecase “Xem giỏ hàng”

1. Tên use case: Xem giỏ hàng

2. Mô tả vắn tắt: use case cho phép khách hàng xem, xóa và thay đổi số lượng của sản phẩm trong giỏ hàng.

3. Mô tả chi tiết:

a. Luồng cơ bản:

i. Use case bắt đầu khi khách hàng kích vào biểu tượng giỏ hàng, hệ thống sẽ truy cập vào csdl và lấy ra thông tin về các sản phẩm từ bảng CART bao gồm: ảnh minh hoạ, tên sản phẩm, , số lượng, giá tiền và hiển thị lên màn hình. Use case kết thúc.

b. Luồng rẽ nhánh:

i. Tại bước 1 trong luồng cơ bản, nếu khách hàng chưa thêm sản phẩm nào vào giỏ hàng, hệ thống sẽ hiển thị "Hiện chưa có sản phẩm". Use case kết thúc.

ii. Tại bước 2 trong luồng cơ bản, nếu chưa thêm sản phẩm nào vào giỏ hàng, hệ thống sẽ hiển thị "Giỏ hàng của bạn đang trống" và 1 ô "Tiếp tục mua hàng". Use case kết thúc.

iii. Tại bất kì thời điểm nào trong quá trình thực hiện ca sủ dụng nếu không kết nối được với CSDL thì hệ thống sẽ hiển thị thông báo lỗi và ca sử dụng kết thúc.

4. Điểm mở rộng:không

5. Tiển điều kiện: không

6. Hậu điều kiện: không

7. Các yêu cầu đặc biệt: không

* + 1. Mô tả usecase “Đăng kí”

1. Tên use case: Đăng kí

2. Mô tả vắn tắt: use case này cho phép khách hàng đăng kí tài khoản.

3. Luồng sự kiện:

a. Luồng cơ bản:

i. Use case này bắt đầu khi khách hàng kích vào “Đăng k”". Hệ thống sẽ hiển thị giao diện đăng ký.

ii. Hệ thống sẽ hiển thị giao diện đăng kí bao gồm: họ, tên, giới tính, ngày sinh , email, mật khẩu

iii. Khách hàng điền đầy đủ thông tin theo yêu cầu, rồi kích nút “Đăng kí”, hệ thống sẽ tạo một mã tài khoản mới, lưu thông tin vào bảng USER. Use case kết thúc.

b. Luồng rẽ nhánh:

i. Tại bước 3 trong luồng cơ bản nếu khách hàng nhập thiếu thông tin, hệ thống sẽ yêu cầu khách hàng nhập đầy đủ thông tin. Nếu khách hàng nhập ko đúng định dạng email hoặc mật khẩu dưới 5 kí tự, hệ thống sẽ hiển thị lên 1 bảng thông báo nhỏ và yêu cầu nhập lại. Use case kết thúc.

ii. Tại bất kì thời điểm nào trong quá trình thực hiện ca sủ dụng nếu không kết nối được với CSDL thì hệ thống sẽ hiển thị thông báo lỗi và ca sử dụng kết thúc.

4. Các yêu cầu đặc biệt: không.

5. Tiền điều kiện: không.

6. Hậu điều kiện: không.

7. Điều kiện mở rộng: không.

* + 1. Mô tả usecase “Đặt hàng”

1. Tên use case: Đặt hàng

2. Mô tả vắn tắt: use case này cho phép khách hàng đặt hàng .

3. Luồng sự kiện:

a. Luồng cơ bản:

i. Use case này bắt đầu khi khách hàng kích vào nút “Đặt hàng” . Hệ thống sẽ hiển thị giao diện cho khách hàng đặt hàng .

ii. Khi khách hàng nhập vào các thông tin cho việc đặt hàng như , tên người nhận, số điện thoại và địa chỉ nhận hàng và ấn vào nút “Đặt hàng”. Hệ thống sẽ tạo ra 1 đơn hàng mới trong bảng ORDERPRODUCT và tạo ra các chi tiết đơn hàng trong bảng ORDERDETAIL rồi cập nhật lại số lượng trong bảng PRODUCT .Usecase kết thúc.

b. Luồng rẽ nhánh:

i. Tại bước 3 trong luồng cơ bản nếu khách hàng nhập thiếu thông tin, hệ thống sẽ yêu cầu khách hàng nhập đầy đủ thông tin. Nếu khách hàng nhập ko đúng định dạng email hoặc mật khẩu dưới 5 kí tự, hệ thống sẽ hiển thị lên 1 bảng thông báo nhỏ và yêu cầu nhập lại. Use case kết thúc.

ii. Tại bất kì thời điểm nào trong quá trình thực hiện ca sủ dụng nếu không kết nối được với CSDL thì hệ thống sẽ hiển thị thông báo lỗi và ca sử dụng kết thúc.

4. Các yêu cầu đặc biệt: không.

5. Tiền điều kiện: không.

6. Hậu điều kiện: không.

7. Điều kiện mở rộng: không.

* + 1. Mô tả usecase “Quên mật khẩu “

1. Tên use case: Quên mật khẩu

2. Mô tả vắn tắt: use case này cho phép khách hàng được cấp mật khẩu mới .

3. Luồng sự kiện:

a. Luồng cơ bản:

i. Use case này bắt đầu khi khách hàng kích vào biểu tượng tài khoản. Hệ thống sẽ hiển thị giao diện đăng nhập

ii. Khi khách hàng kích vào dòng quên mật khẩu, hệ thống sẽ hiển thị giao diện quên mật khẩu bao gồm: email

iii. Khách hàng nhập email vào, rồi kích nút “Gửi”. Hệ thống sẽ tìm thông tin từ bảng UESR rồi gửi mật khẩu mới vào hòm thư của email. Use case kết thúc.

b. Luồng rẽ nhánh:

i. Tại bước 3 trong luồng cơ bản nếu khách hàng nhập sai email. Hệ thống sẽ thông báo “Không tìm thấy tài khoản nào với email này”. Use case kết thúc.

ii. Tại bất kì thời điểm nào trong quá trình thực hiện ca sủ dụng nếu không kết nối được với CSDL thì hệ thống sẽ hiển thị thông báo lỗi và ca sử dụng kết thúc.

4. Các yêu cầu đặc biệt: không.

5. Tiền điều kiện: khách hàng phải có tài khoản.

6. Hậu điều kiện: không.

7. Điều kiện mở rộng: không.

* + 1. Mô tả usecase “Bảo trì sản phẩm”

1. tên use case : Bảo trì sản phẩm

2.Mô tả vắn tắt :Use case này cho phép người quản trị thêm, sửa ,xóa các sản phẩm trong bảng PRODUCT.

3. các luồng sự kiện

3.1 luồng cơ bản

1) use case này bắt đầu khi người quản trị kích vào nút mục “quản lý sản phẩm” trên menu quản trị .Hệ thống lấy thông tin chi tiết từ bảng PRODUCT trong cơ sở dữ liệu và hiển thị danh sách các sản phẩm lên màn hình.

2) Thêm sản phẩm:

a. Người quản trị kích vào nút” thêm mới “ trên màn hinh. Hệ thống hiển thị lên màn hình yêu cầu nhập thông tin chi tiết cho sản phẩm cần thêm mới gồm tên sản phẩm, giá bán, số lượng, hình ảnh, tên thương hiệu, tên danh mục.

b.Người quản trị nhập thông tin vào kích vào nút thêm mới . Hệ thống sẽ sinh ra một cửa hàng mới trong bảng PRODUCT và hiển thị danh sách các sản phẩm lên màn hình .

3) Sửa sản phẩm :

a.Người quản trị kích vào nút sửa trên một dòng sản phẩm. Hệ thống sẽ lấy ra thông tin của sản phẩm người quản trị muốn sửa và hiển thị lên màn hình .

b. Người quản trị nhập thông tin mới cho sản phẩm và kích vào nút cập nhật. Hệ thống sẽ sửa thông tin của sản phẩm người quản trị vừa sửa và hiển thị danh sách sản phẩm đã cập nhật lên màn hình.

4)Xóa sản phẩm :

a.Người quản trị kích vào nút xóa trên một sản phẩm .Hệ thống sẽ hiển thị lên màn hình yêu cầu xác nhận xóa.

b. Người quản trị kích vào nút “Đồng ý”. Hệ thống sẽ xóa sản phẩm được chọn khỏi bảng PRODUCT và hiển thị danh sách các sản phẩm đã cập nhật.

Use case kết thúc.

3.2) Các luồng rẽ nhánh.

1)Tại bước 2b,3b trong luồng cơ bản nếu người quản trị nhập thông tin thương hiệu không hợp lệ thì hệ thống sẽ hiển thị thông báo lỗi yêu cầu nhập lại. Người quản trị có thể nhập lại để tiếp tục hoặc kích vào nút “Hủy bỏ” để kết thúc.

2) Tại bước 2b hoặc 3b trong luồng cơ bản nếu người quản trị kích vào nút “Hủy bỏ” hệ thống sẽ bỏ qua thao tác thêm mới hoặc sửa chữa tương ứng và hiển thị danh sách các cửa hàng trong bảng PRODUCT.

3) Tại bước 4b trong luồng cơ bản nếu người quản trị kích vào nút “Không đồng ý” hệ thống sẽ bỏ qua thao tác xóa và hiển thị danh sách các cửa hàng trong bảng PRODUCT.

4) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sử dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

4. Các yêu cầu đặc biệt

Use case này chỉ cho phép một số vai trò như người quản trị, người chủ hệ thống thực hiện.

5. Tiền điều kiện

Người quản trị cần đăng nhập với vai trò quản trị hệ thống trước khi có thể thực hiện use case.

6. Hậu điều kiện

Nếu use case kết thúc thành công thì thông tin về thương hiệu sẽ được cập nhập trong cơ sở dữ liệu.

7. Điểm mở rộng

Không có .

* + 1. Mô tả usecase “Xem thống kê”

1.Tên use case : Xem thống kê

2. Mô tả vắn tắt : Use case này cho phép người quản trị xem được thống kế doanh thu bán hàng

3 . Luồng sự kiện :

3.1 : Luồng cơ bản :

1. Use case bắt đầu khi người quản trị kích vào “xem thống kê” trên trang chủ hệ thống. Hệ thống sẽ lấy các thông tin thống kê được từ bảng ORDERPRODUCT VÀ ORDERDETAIL hiển thị lên mà hình.

2. Khi người quản trị kích chọn vào 1 năm và 1 tháng trong danh sách. Hệ thống sẽ lọc thống kê thu được trong bảng ORDERPRODUCT VÀ ORDERDETAIL hiển thị lên màn hình và usecase kết thúc.

3.2 : luồng rẽ nhánh:

1. Tại bước 1 trong luồng cơ bản nếu khách hàng chưa tạo một tài khoản để đăng nhập vào hệ thống thì hệ thống sẽ chuyển hướng người dùng đến usecase đăng nhập và đăng ký và use case kết thúc.

2. Tại bất kỳ thời điểm nào nếu use case không kết nối được với cơ sở dữ liệu thì hệ thống sẽ báo lỗi và use case kết thúc.

4.Các yêu cầu đặc biệt : không có

5. Tiền điều kiện : Quản trị viên phải đăng nhập vào hệ thống

6. Hậu điều kiện : không có

7. Điểm mở rộng : Không có

* + 1. Mô tả usecase “Xem chi tiết sản phẩm”

*1.*Tên use case : Xem chi tiết sản phẩm

2. Mô tả vắn tắt : Use case này cho phép khách hàng xem được chi tiết sản phẩm của shop

3 . Luồng sự kiện :

3.1 : Luồng cơ bản :

1. Use case bắt đầu khi khách hàng kích vào một sản phẩm bất kì trên trang chủ hệ thống. Hệ thống sẽ lấy các thông tin trong bảng PRODUCT gồm tên sản phẩm,ảnh minh họa , mô tả , thông tin, giá ,hướng dẫn sử dụng sản phẩm hiển thị lên màn hình và use case kết thúc.

3.2 : luồng rẽ nhánh:

1. Tại bất kỳ thời điểm nào nếu use case không kết nối được với cơ sở dữ liệu thì hệ thống sẽ báo lỗi và use case kết thúc.

4.Các yêu cầu đặc biệt : không có

5. Tiền điều kiện : không có

6. Hậu điều kiện : không có

7. Điểm mở rộng : Không có

* + 1. Mô tả usecase “Thêm sản phẩm vào giỏ hàng”

1.Tên use case : Thêm sản phẩm vào giỏ hàng

2. Mô tả vắn tắt : Use case này cho phép khách hàng thêm sản phẩm vào trong bảng giỏ hàng

3 . Luồng sự kiện :

3.1 : Luồng cơ bản :

1. Use case bắt đầu khi khách hàng kích vào”Add to cart” một sản phẩm bất kì trên trang chủ hệ thống. Hệ thống sẽ lấy các thông tin trong bảng PRODUCT gồm tên sản phẩm,ảnh minh họa , đơn giá tạo ra 1 sản phẩm trong bảng CART và use case kết thúc.

3.2 : Luồng rẽ nhánh:

1. Tại bất kỳ thời điểm nào nếu use case không kết nối được với cơ sở dữ liệu thì hệ thống sẽ báo lỗi và use case kết thúc.

4.Các yêu cầu đặc biệt : không có

5. Tiền điều kiện : không có

6. Hậu điều kiện : không có

7. Điểm mở rộng : Không có

* + 1. Mô tả usecase “Quản lý giỏ hàng”

1. Tên use case : Quản lý giỏ hàng

2.Mô tả vắn tắt :Use case này cho phép người dùng sửa ,xóa các sản phẩm trong bảng giỏ hàng.

3. Các luồng sự kiện

3.1 Luồng cơ bản

1) Use case này bắt đầu khi người quản trị kích vào nút mục biểu tượng giỏ hàng trên trang chủ .Hệ thống lấy thông tin chi tiết từ bảng CART trong cơ sở dữ liệu và hiển thị danh sách các sản phẩm trong bảng giỏ hàng lên màn hình.

2) Sửa sản phẩm trong giỏ hàng:

a.Người dùng kích vào nút cộng và trừ trên một dòng sản phẩm. Hệ thống sẽ lấy ra thông tin của sản phẩm người quản trị muốn sửa cập nhật lại số lượng hiển thị lên màn hình .

3)Xóa sản phẩm trong giỏ hàng :

a.Người quản trị kích vào biểu tượng xóa trên một sản phẩm .Hệ thống sẽ xóa sản phẩm mà người dùng chọn và hiển thị lại danh sách lên màn hình.

Use case kết thúc.

3.2) Các luồng rẽ nhánh.

*1.* Tại bất kỳ thời điểm nào nếu use case không kết nối được với cơ sở dữ liệu thì hệ thống sẽ báo lỗi và use case kết thúc.

4. Các yêu cầu đặc biệt

Use case này chỉ cho phép người dùng đã đăng nhập vào hệ thống thực hiện.

5. Tiền điều kiện

Người dùng cần đăng nhập vào hệ thống.

6. Hậu điều kiện

Không có.

7. Điểm mở rộng

Không có .

* + 1. Mô tả usecase “Tìm kiếm sản phẩm theo tên”

1. Tên use case: Tìm kiếm sản phẩm theo tên

2. Mô tả vắn tắt: use case này cho phép khách hàng có thể tìm kiếm được sản phẩm theo từ khóa tìm kiếm.

3. Luồng sự kiện:

a. Luồng cơ bản:

i. Use case này bắt đầu khi khách hàng nhập từ khóa tìm kiếm vào ô tìm kiếm trên giao diện . Hệ thống sẽ lấy dữ liệu của bảng PRODUCT mà có tên sản phẩm có chứa từ khóa tìm kiếm và hiển thị lên màn hình . Use case kết thúc.

b. Luồng rẽ nhánh:

i. Tại bất kì thời điểm nào trong quá trình thực hiện ca sủ dụng nếu không kết nối được với CSDL thì hệ thống sẽ hiển thị thông báo lỗi và ca sử dụng kết thúc.

4. Các yêu cầu đặc biệt: không.

5. Tiền điều kiện: Không.

6. Hậu điều kiện: không.

7. Điều kiện mở rộng: không.

* + 1. Mô tả usecase “Bảo trì thương hiệu”

1. Tên use case : Bảo trì thương hiệu

2.Mô tả vắn tắt :Use case này cho phép người quản trị thêm, sửa ,xóa các thương hiệu trong bảng MANUFACTORER.

3. Các luồng sự kiện

3.1 Luồng cơ bản

1) Use case này bắt đầu khi người quản trị kích vào nút mục “quản lý thương hiệu” trên menu quản trị .Hệ thống lấy thông tin từ bảng MANUFACTORER trong cơ sở dữ liệu và hiển thị danh sách các thương hiệu lên màn hình.

2) Thêm mới thương hiệu:

a. Người quản trị kích vào nút” thêm mới “ trên màn hình. Hệ thống hiển thị lên màn hình yêu cầu nhập thông tin chi tiết cho thương hiệu.

b.Người quản trị nhập thông tin vào kích vào nút thêm mới . Hệ thống sẽ sinh ra một thương hiệu mới trong bảng MANUFACTORER và hiển thị danh sách các thương hiệu lên màn hình .

3) Sửa thương hiệu :

a.Người quản trị kích vào nút sửa trên một dòng thương hiệu. Hệ thống sẽ lấy ra thông tin của thương hiệu người quản trị muốn sửa và hiển thị lên màn hình .

b. Người quản trị nhập thông tin mới cho thương hiệu và kích vào nút cập nhật. Hệ thống sẽ sửa thông tin của thương hiệu người quản trị vừa sửa và hiển thị danh sách thương hiệu đã cập nhật lên màn hình.

4)Xóa thương hiệu:

a.Người quản trị kích vào nút xóa trên một thương hiệu.Hệ thống sẽ hiển thị lên màn hình yêu cầu xác nhận xóa.

b. Người quản trị kích vào nút “Đồng ý”. Hệ thống sẽ xóa thương hiệu được chọn khỏi bảng MANUFACTORER và hiển thị danh sách các thương hiệu đã cập nhật.

Use case kết thúc.

3.2) Các luồng rẽ nhánh.

1)Tại bước 2b,3b trong luồng cơ bản nếu người quản trị nhập thông tin thương hiệu không hợp lệ thì hệ thống sẽ hiển thị thông báo lỗi yêu cầu nhập lại. Người quản trị có thể nhập lại để tiếp tục hoặc kích vào nút “Hủy bỏ” để kết thúc.

2) Tại bước 2b hoặc 3b trong luồng cơ bản nếu người quản trị kích vào nút “Hủy bỏ” hệ thống sẽ bỏ qua thao tác thêm mới hoặc sửa chữa tương ứng và hiển thị danh sách các thương hiệu trong bảng MANUFACTORER.

3) Tại bước 4b trong luồng cơ bản nếu người quản trị kích vào nút “Không đồng ý” hệ thống sẽ bỏ qua thao tác xóa và hiển thị danh sách các thương hiệu trong bảng MANUFACTORER.

4) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sử dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

4. Các yêu cầu đặc biệt

Use case này chỉ cho phép một số vai trò như người quản trị, người chủ hệ thống thực hiện.

5. Tiền điều kiện

Người quản trị cần đăng nhập với vai trò quản trị hệ thống trước khi có thể thực hiện use case.

6. Hậu điều kiện

Nếu use case kết thúc thành công thì thông tin về thương hiệu sẽ được cập nhập trong cơ sở dữ liệu.

7. Điểm mở rộng

Không có .

* + 1. Mô tả usecase “Quản lý thông tin tài khoản”

1. Tên use case : Quản lý thông tin tài khoản

2.Mô tả vắn tắt :Use case này cho phép người dùng sửa các thông tin của chính họ trong bảng USER.

3. Các luồng sự kiện

3.1 Luồng cơ bản

1) Use case này bắt đầu khi người quản trị kích vào tên của người dùng trên menu trang chủ .Hệ thống lấy thông tin chi tiết từ bảng USER trong cơ sở dữ liệu và hiển thị thông tin của người dùng đó trong bảng USER lên màn hình.

2) Sửa thông tin người dùng:

a.Người dùng có thể sửa các thông tin của họ trên giao diện màn hình và ấn vào nút cập nhật. Hệ thống sẽ cập nhật lại thông tin mà người dùng sửa và hiển thị lên màn hình .

Use case kết thúc.

3.2) Các luồng rẽ nhánh.

*1.* Tại bất kỳ thời điểm nào nếu use case không kết nối được với cơ sở dữ liệu thì hệ thống sẽ báo lỗi và use case kết thúc.

4. Các yêu cầu đặc biệt

Use case này chỉ cho phép người dùng đã đăng nhập vào hệ thống thực hiện.

5. Tiền điều kiện

Người dùng cần đăng nhập vào hệ thống.

6. Hậu điều kiện

Không có.

7. Điểm mở rộng

Không có .

* + 1. Mô tả usecase “Quản lý tài khoản”

1. Tên use case : Quản lý các tài khoản của người dùng

2.Mô tả vắn tắt :Use case này cho người quản trị khóa các tài khoản của người dùng trong bảng USER.

3. Các luồng sự kiện

3.1 Luồng cơ bản

1) Use case này bắt đầu khi người quản trị kích vào mục” quản lý tài khoản” trên trang chủ hệ thống .Hệ thống lấy thông tin chi tiết từ bảng USER trong cơ sở dữ liệu và hiển thị danh sách các người dùng lên màn hình.

2) Khóa tài khoản người dùng:

a.Người quản trị kích vào nút khóa trên một dòng người dùng.Hệ thống sẽ cập nhật lại trạng thái của người dùng đó trong bảng USER và hiển thị lên màn hình .

Use case kết thúc.

3.2) Các luồng rẽ nhánh.

*1.* Tại bất kỳ thời điểm nào nếu use case không kết nối được với cơ sở dữ liệu thì hệ thống sẽ báo lỗi và use case kết thúc.

4. Các yêu cầu đặc biệt

Không

5. Tiền điều kiện

Người quản trị cần đăng nhập vào hệ thống.

6. Hậu điều kiện

Không có.

7. Điểm mở rộng

Không có .

* + 1. Mô tả usecase “Quản lý đơn hàng”

1. Tên use case : Quản lý đơn hàng

2.Mô tả vắn tắt :Use case này cho phép người quản trị sửa ,xóa các đơn hàng trong bảng ORDERPRODUCT.

3. Các luồng sự kiện

3.1 Luồng cơ bản

1) Use case này bắt đầu khi người quản trị kích vào nút mục “quản lý đơn hàng” trên menu quản trị .Hệ thống lấy thông tin chi tiết từ bảng ORDERPRODUCT trong cơ sở dữ liệu và hiển thị danh sách các đơn hàng lên màn hình.

2) Sửa đơn hàng:

a.Người quản trị kích vào nút “duyệt” trên một dòng đơn hàng. Hệ thống sẽ cập nhật lại trạng thái của đơn hàng và hiển thị lên màn hình.

4)Xóa sản phẩm :

a.Người quản trị kích vào nút xóa trên một đơn hàng .Hệ thống sẽ hiển thị lên màn hình yêu cầu xác nhận xóa.

b. Người quản trị kích vào nút “Đồng ý”. Hệ thống sẽ đơn hàng đã chọn khỏi bảng ORDERPRODUCT và hiển thị danh sách các đơn hàng đã cập nhật.

Use case kết thúc.

3.2) Các luồng rẽ nhánh.

1)Tại bước 2b,3b trong luồng cơ bản nếu người quản trị nhập thông tin thương hiệu không hợp lệ thì hệ thống sẽ hiển thị thông báo lỗi yêu cầu nhập lại. Người quản trị có thể nhập lại để tiếp tục hoặc kích vào nút “Hủy bỏ” để kết thúc.

2) Tại bước 2b hoặc 3b trong luồng cơ bản nếu người quản trị kích vào nút “Hủy bỏ” hệ thống sẽ bỏ qua thao tác thêm mới hoặc sửa chữa tương ứng và hiển thị danh sách các cửa hàng trong bảng ORDERPRODUCT.

3) Tại bước 4b trong luồng cơ bản nếu người quản trị kích vào nút “Không đồng ý” hệ thống sẽ bỏ qua thao tác xóa và hiển thị danh sách các đơn hàng trong bảng ORDERPRODUCT.

4) Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sử dữ liệu thì hệ thống sẽ hiển thị một thông báo lỗi và use case kết thúc.

4. Các yêu cầu đặc biệt

Use case này chỉ cho phép một số vai trò như người quản trị, người chủ hệ thống thực hiện.

5. Tiền điều kiện

Người quản trị cần đăng nhập với vai trò quản trị hệ thống trước khi có thể thực hiện use case.

6. Hậu điều kiện

Nếu use case kết thúc thành công thì thông tin về thương hiệu sẽ được cập nhập trong cơ sở dữ liệu.

7. Điểm mở rộng

Không có .

## Biểu đồ trình tự

3.5.1 Biểu đồ trình tự usecase “Đăng nhập“



Hình 3. 4 Biểu đồ trình tự usecase đăng nhập

* + 1. Biểu đồ trình tự usecase “Xem sản phẩm theo thương hiệu“



Hình 3.5 Biểu đồ trình tự usecase xem sản phẩm theo thương hiệu

* + 1. Biểu đồ trình tự usecase “Xem giỏ hàng“



Hình 3.6 Biểu đồ trình tự usecase xem giỏ hàng

* + 1. Biểu đồ trình tự usecase “Đăng kí“



Hình 3.7 Biểu đồ trình tự usecase đăng kí

* + 1. Biểu đồ trình tự usecase Đặt hàng



Hình 3.8 Biểu đồ trình tự usecase đặt hàng

* + 1. Biểu đồ trình tự usecase ‘Quên mật khẩu’



Hình 3.9 Biểu đồ trình tự usecase quên mật khẩu

* + 1. Biểu đồ trình tự usecase ‘Bảo trì sản phẩm”



Hình 3.10 Biểu đồ trình tự usecase bảo trì sản phẩm

* + 1. Biểu đồ trình tự usecase “Xem thống kê”



Hình 3.11 Biểu đồ trình tự usecase xem thống kê

* + 1. Biểu đồ trình tự “Xem chi tiết sản phẩm“



Hình 3.12 Biểu đồ trình tự usecase xem chi tiết sản phẩm

* + 1. Biểu đồ trình tự “ Thêm sản phẩm vào giỏ hàng“



Hình 3.13 Biểu đồ trình tự usecase thêm sản phẩm vào giỏ hàng

* + 1. Biểu đồ trình tự “Quản lý giỏ hàng“



Hình 3.14 Biểu đồ trình tự usecase quản lý giỏ hàng

* + 1. Biểu đồ trình tự “Tìm kiếm sản phẩm theo tên”



Hình 3.15 Biểu đồ trình tự usecase tìm kiếm sản phẩm theo tên

* + 1. Biểu đồ trình tự usecase “bảo trị thương hiệu”



Hình 3.16 Biểu đồ trình tự usecase bảo trì thương hiệu

* + 1. Biểu đồ trình tự “Quản lý thông tin tài khoản”



Hình 3.17 Biểu đồ trình tự usecase quản lý thông tin tài khoản

* + 1. Biểu đồ trình tự “Quản lý tài khoản”



Hình 3.18 Biểu đồ trình tự usecase quản lý tài khoản người dùng

3.5.16 Biểu đồ trình tự “Quản lý đơn hàng“

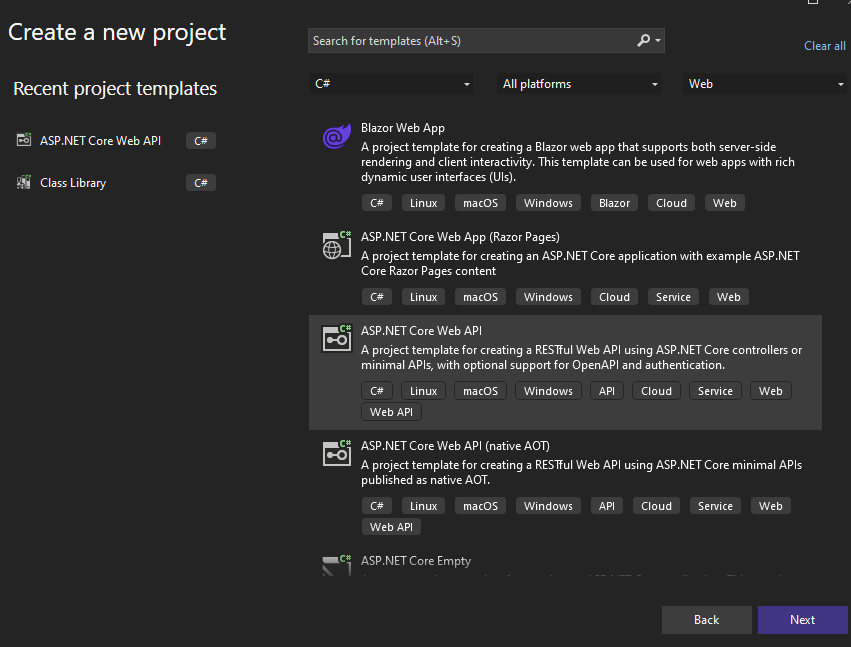


Hình 3.19 Biểu đồ trình tự usecase quản lý đơn hàng

## Cài đặt chương trình

* + 1. Cài đặt ASP.NET Core Web API

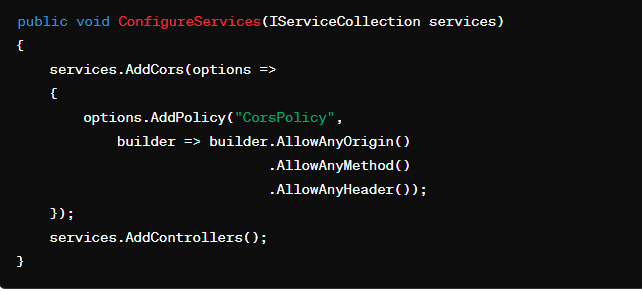
Tạo dự án ASP.NET Core Web API



Hình 3.20 Tạo dự án ASP.net core Web API

Thiết lập CORS

Mở Program.cs và thêm cấu hình CORS trong phương thức ConfigureServices:



Hình 3.21 Thiết lập CROS

Trong phương thức Configure, thêm dòng sau:



Hình 3.22 Sử dụng CROS sau khi thiết lập

* + 1. Cài đặt Vue.js

Cài đặt Node.js và npm :Tải và cài đặt Node.js từ trang web chính thức.

Tạo dự án Vue.js

Mở terminal và chạy các lệnh sau:



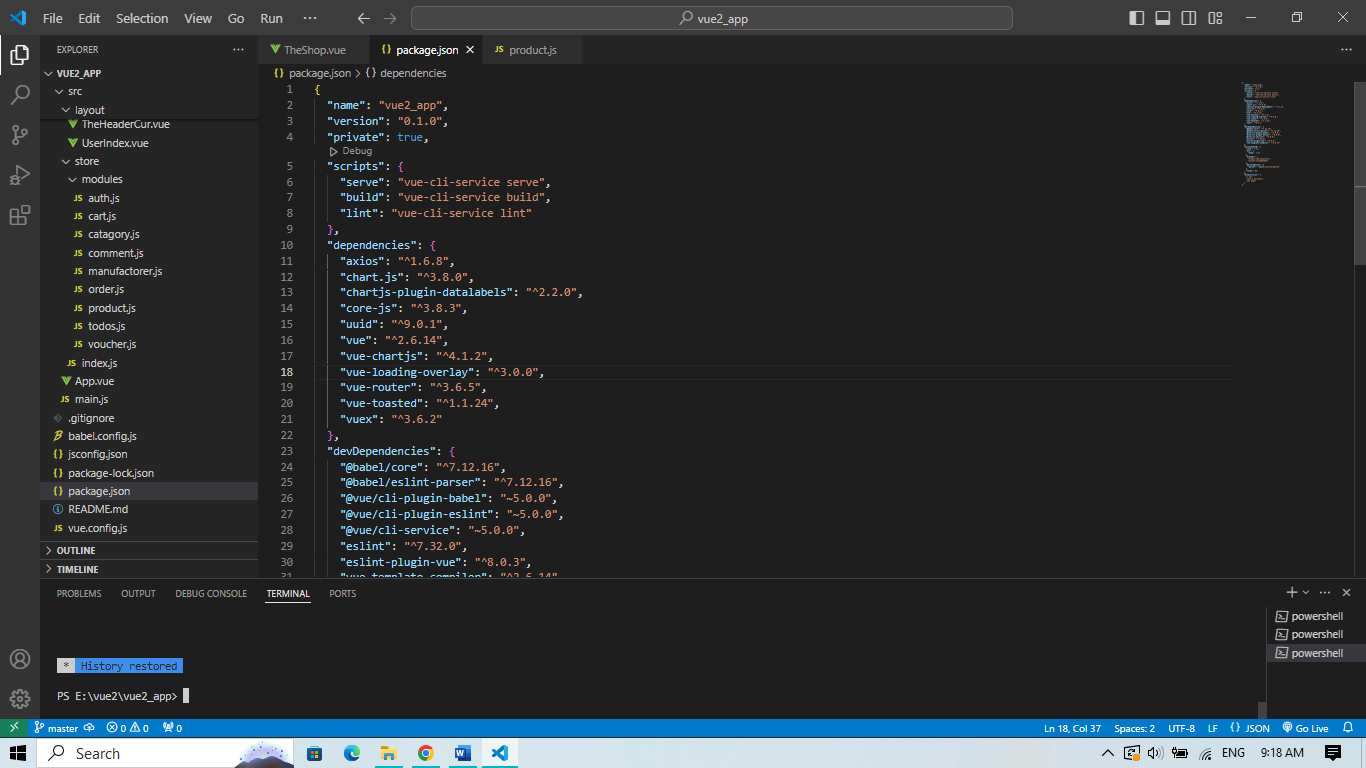
Hình 3.23 Tạo dự án Vue js

Cài đặt các thư viện cùng lúc(Nếu muốn cài đặt đúng phiên bản của từng thư viện chỉ cần ghi rõ phiên bản cài đặt )



Hình 3.24 Cài đặt các thư viện sử dụng trong Vue js

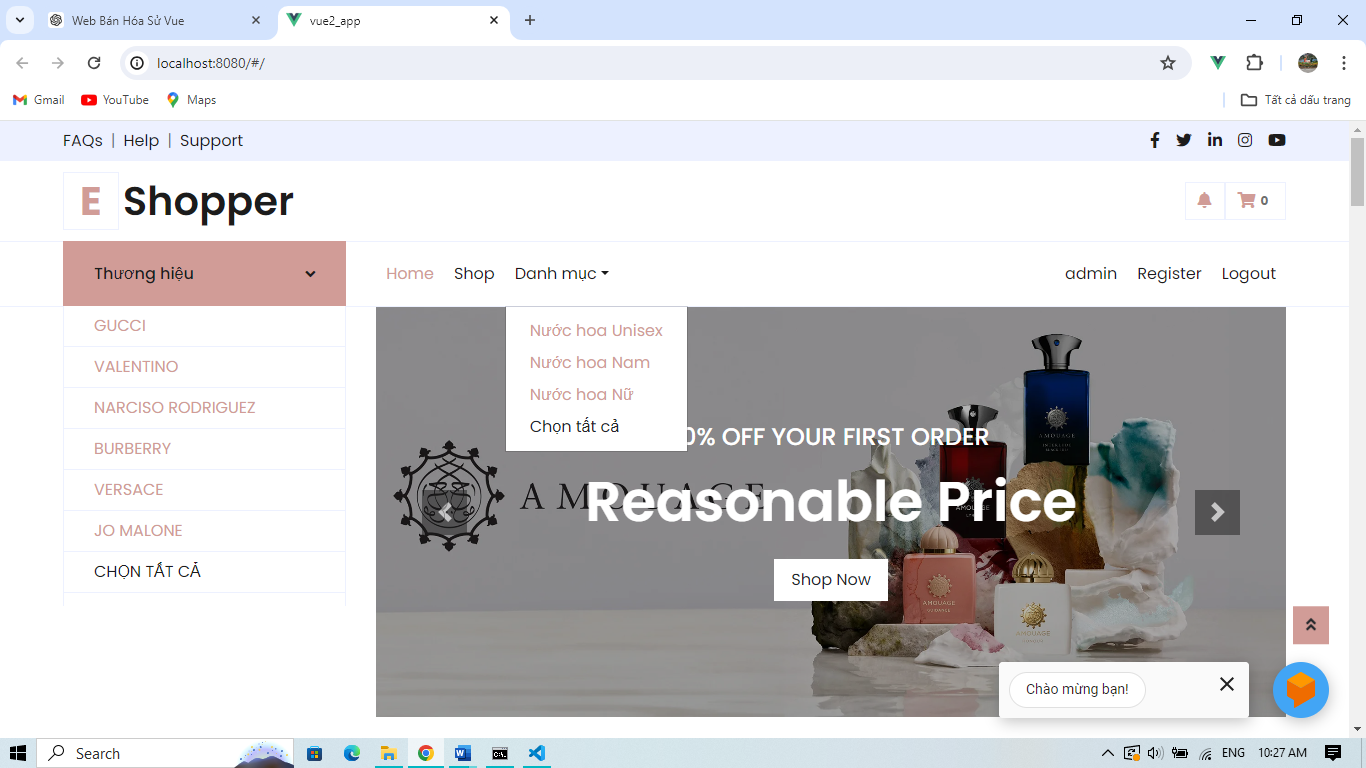
Để xem các phiên bản đã cài đặt truy cập vào file package.json



Hình 3.25 Các gói cài đặt trong dữ án Vue js

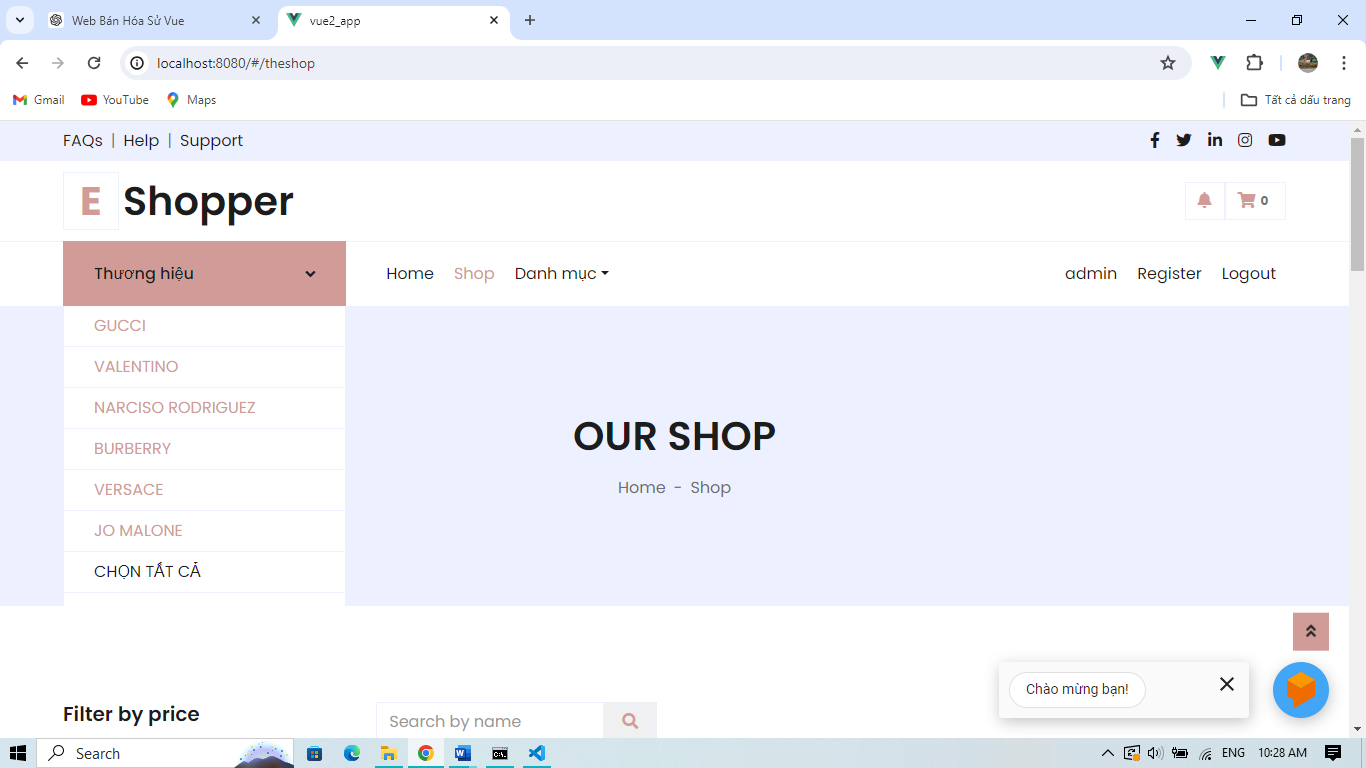
## Giao diện chương trình

* + 1. Giao diện người dùng
       1. Giao diện trang chủ



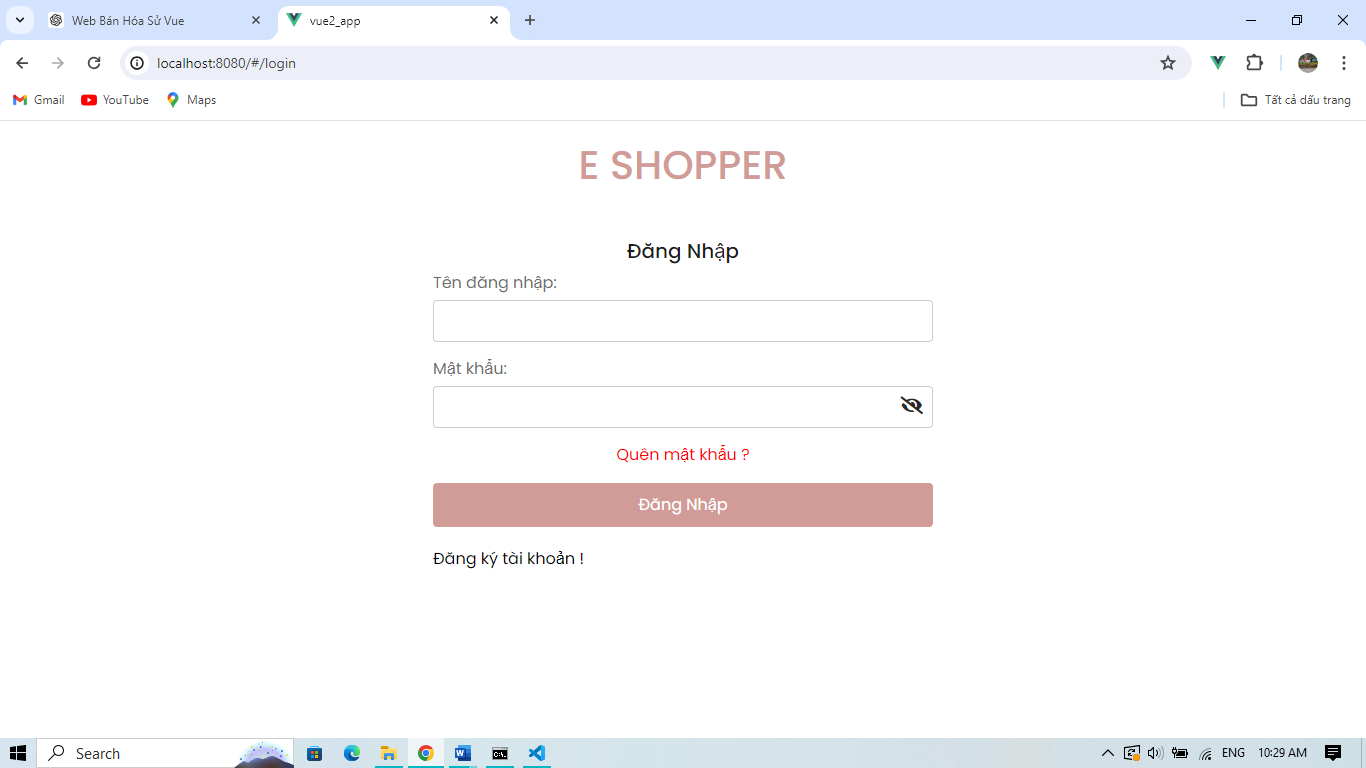
Hình 3.26 Giao diện trang chủ

* + - 1. Giao diện trang cửa hàng



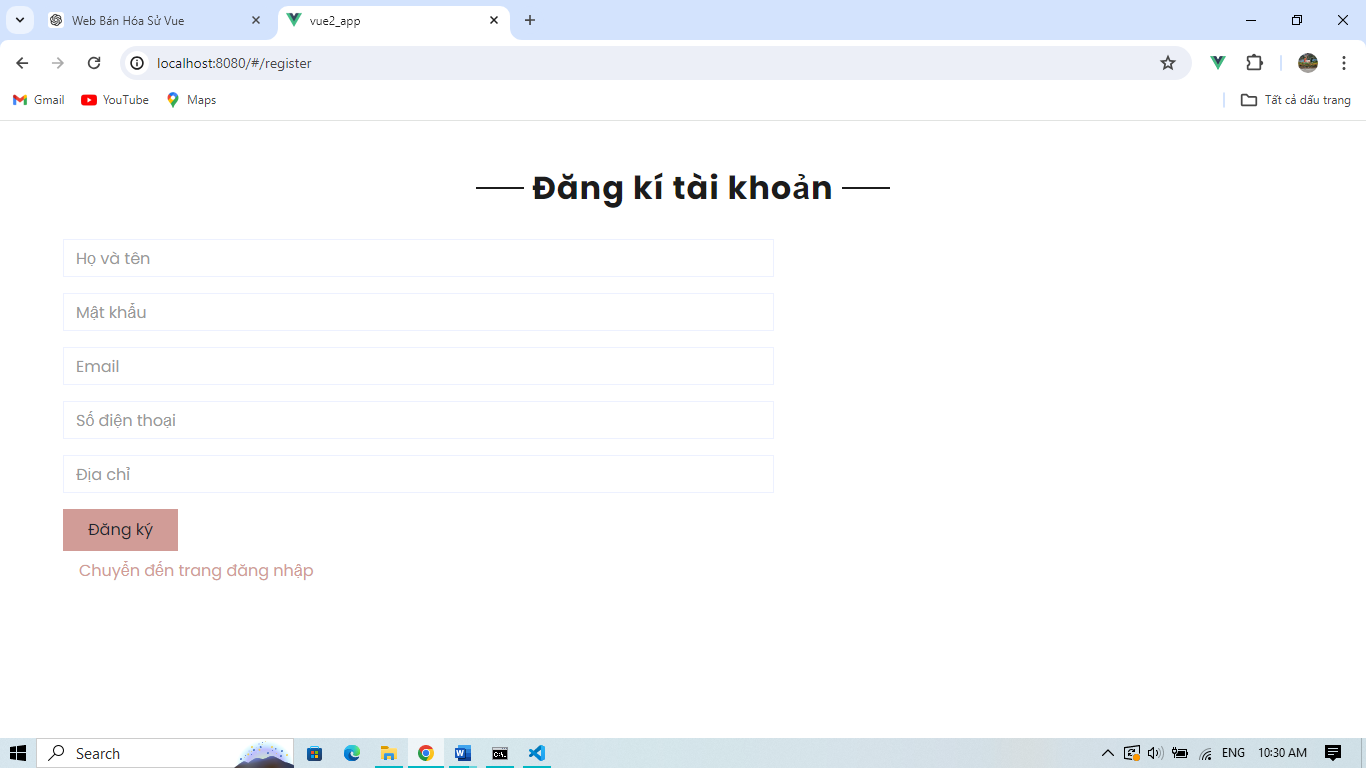
Hình 3.27 Giao diện trang cửa hàng

* + - 1. Giao diện trang đăng nhập



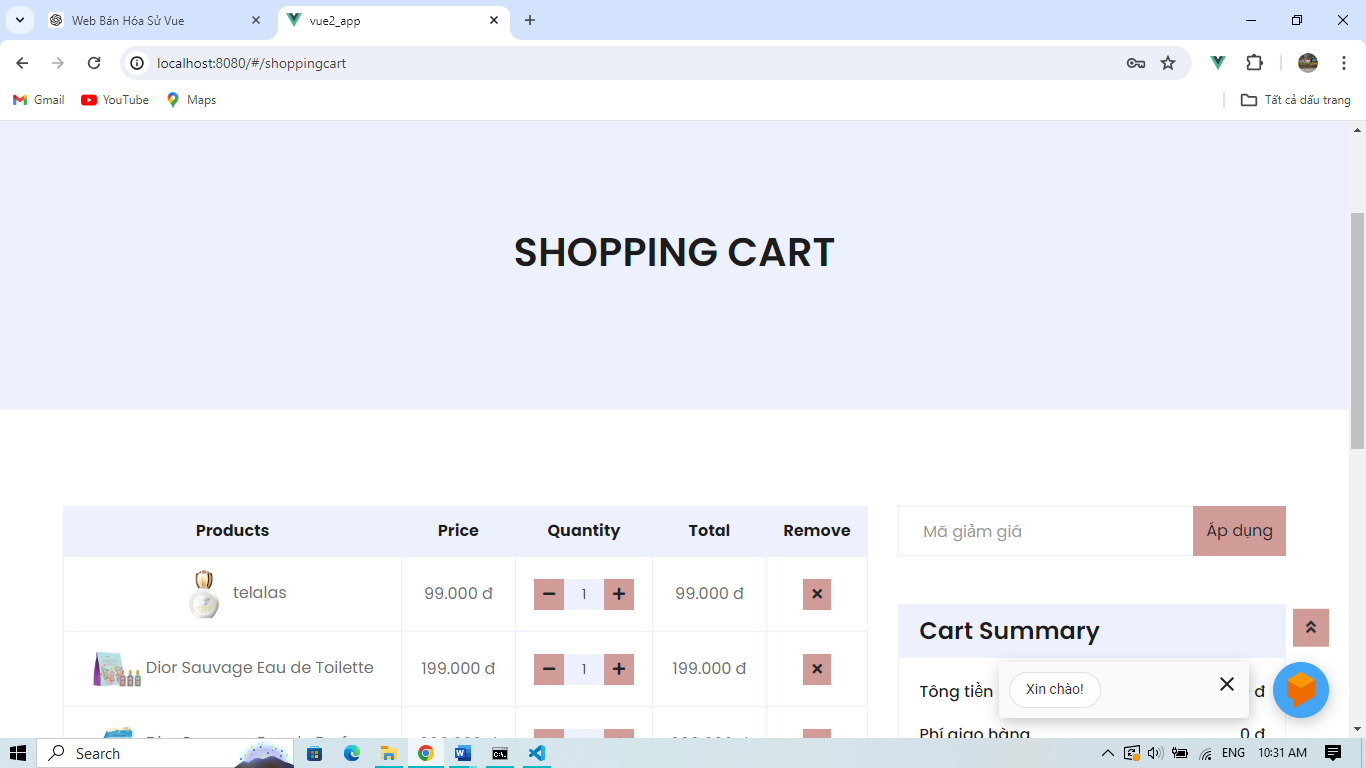
Hình 3.28 Giao diện trang đăng nhập

* + - 1. Giao diện trang đăng ký



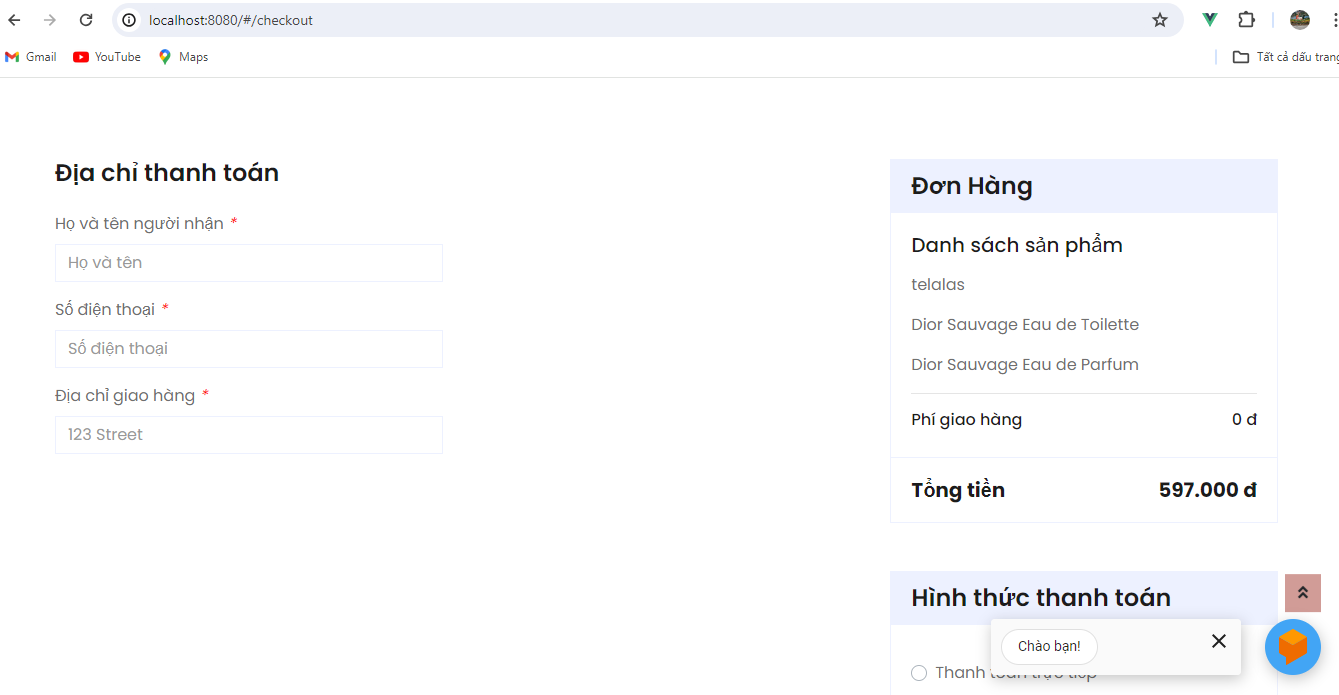
Hình 3.29 Giao diện trang đăng ký

* + - 1. Giao diện trang giỏ hàng



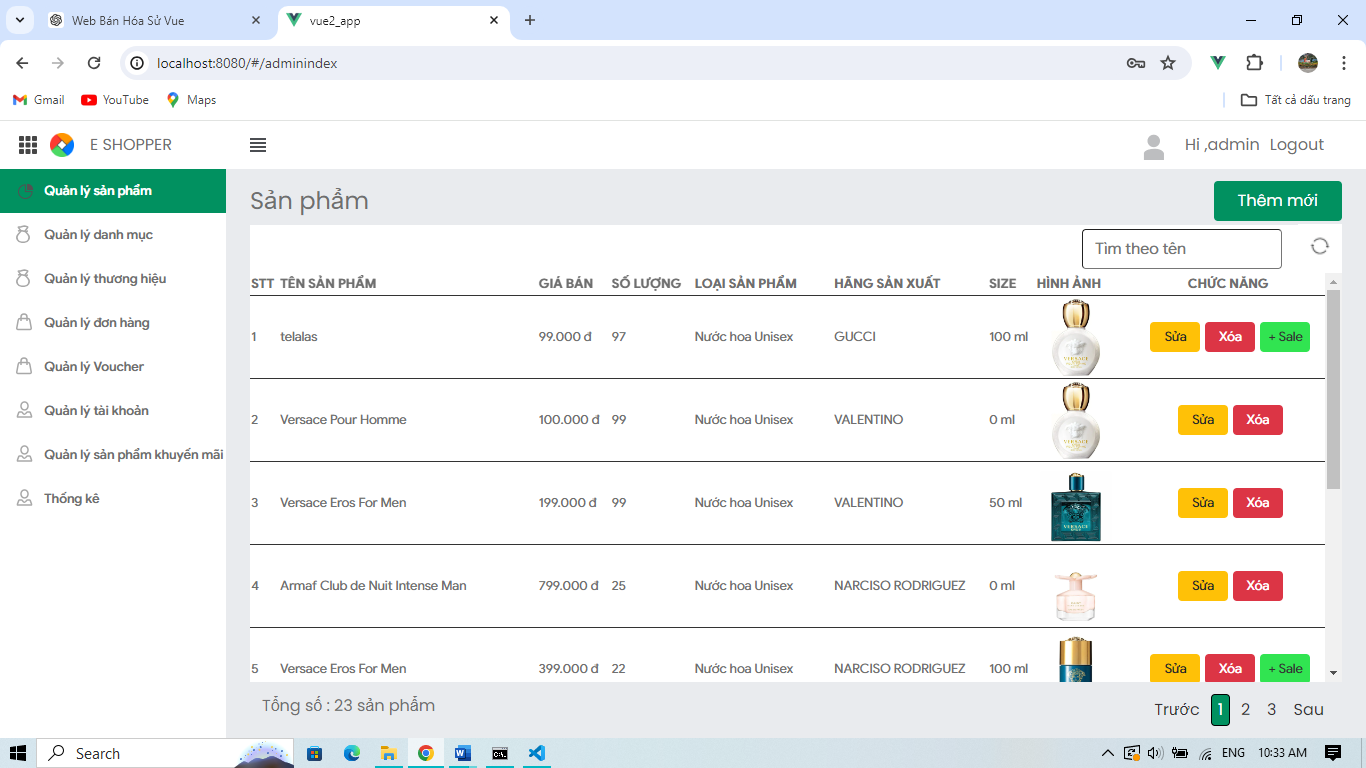
Hình 3.30 Giao diện trang giỏ hàng

* + - 1. Giao diện trang kiểm tra trước khi đặt hàng



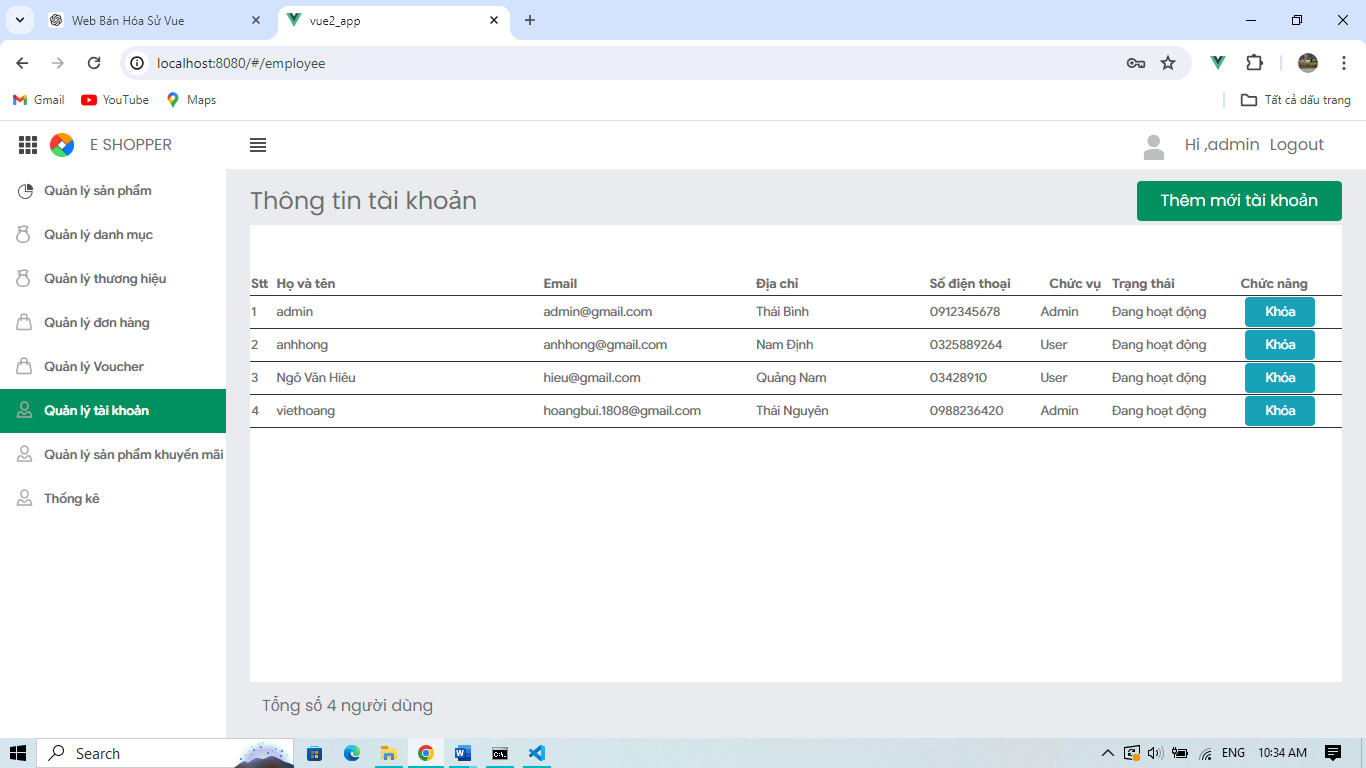
Hình 3.31 Giao diện trang kiểm tra trước khi đặt hàng

* + 1. Giao diện người quản trị
       1. Giao diện trang quản lý sản phẩm



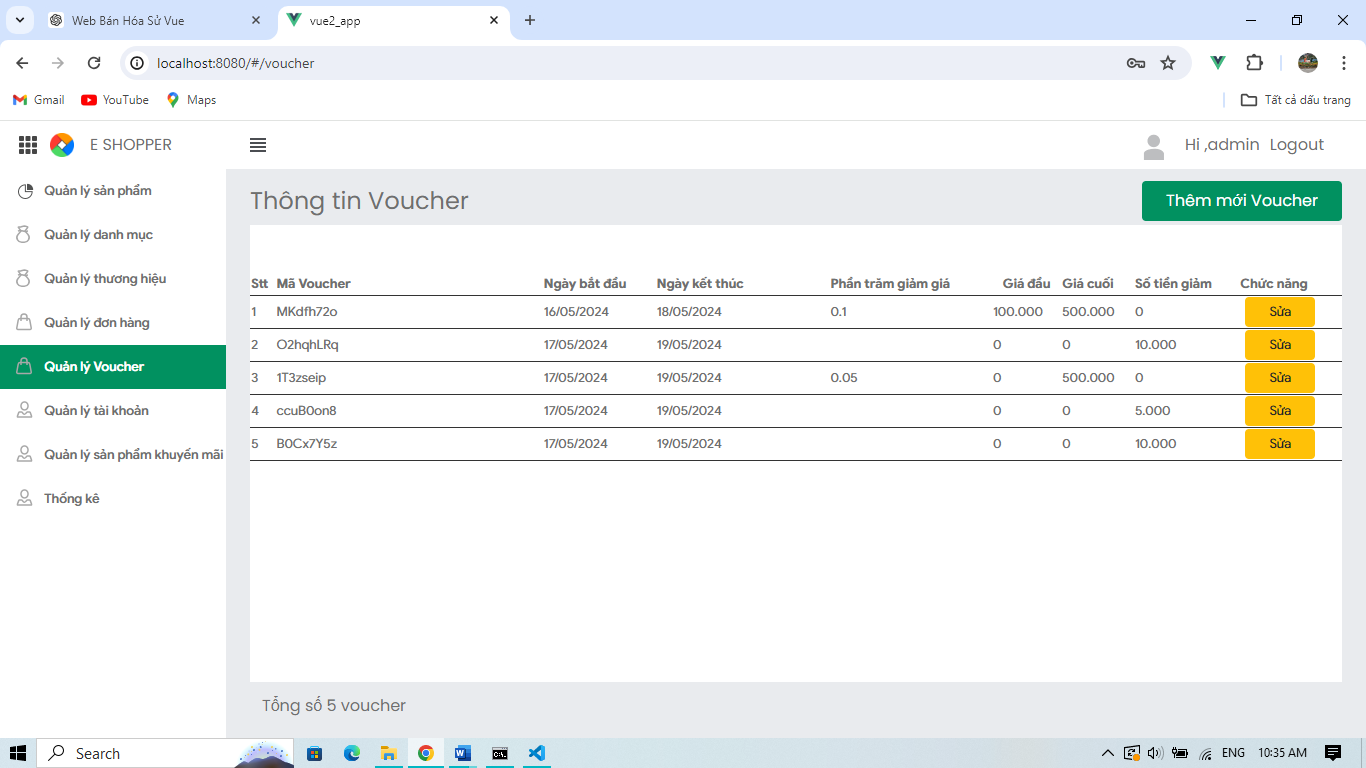
Hình 3.32 Giao diện trang quản lý sản phẩm

* + - 1. Giao diện trang quản lý tài khoản



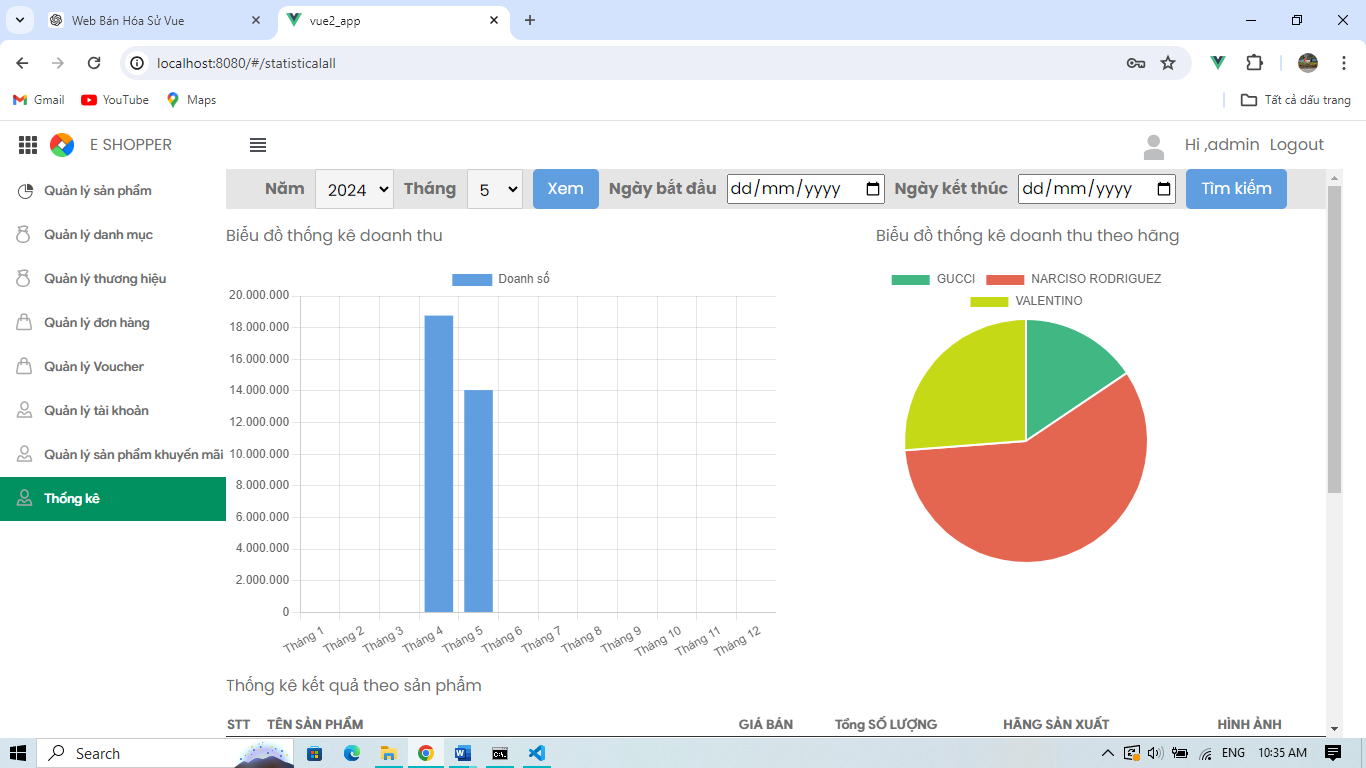
Hình 3.33 Giao diện quản lý tài khoản người dùng

* + - 1. Giao diện trang quản lý voucher



Hình 3.34 Giao diện quản lý voucher

* + - 1. Giao diện trang thống kê doanh thu



Hình 3.35 Giao diện thống kê doanh thu

## Kiểm thử chương trình

* + 1. Chức năng đăng nhập

Bảng 3. 1 Kiểm thử chức năng đăng nhập

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Mô tả trường hợp kiểm tra | Các bước thực hiện | Kết quả mong đợi | Kết quả |
| 1 | Kiểm tra Tài khoản thành công | 1. Nhập tài khoản có trong database 2. Nhấn vào đăng nhập | Đăng nhập thành công | Đạt |
| 2 | Kiểm tra Tên đăng nhập sai | 1. Nhập tên đăng nhập ko có trong DB 2. Nhập mật khẩu đúng 3. Nhấn vào đăng nhập | - Thông báo Tên đăng nhập sai - Set focus và highlight vào trường lỗi | Đạt |
| 3 | Kiểm tra Mật khẩu nhập sai | 1. Nhập tên đăng nhập có trong DB 2. Nhập mật khẩu = rỗng 3. Nhấn vào đăng nhập | - Thông báo Mật khẩu nhập sai - Set focus và highlight vào trường lỗi | Đạt |
| 4 | Kiểm tra Rỗng hết | 1. Nhập tên đăng nhập = Rỗng 2. Nhập mật khẩu= Rỗng 3. Nhấn vào đăng nhập | - Thông báo Tên đăng nhập, Mật khẩu nhập chưa đc nhập - Set focus và highlight vào trường lỗi | Đạt |
| 5 | Kiểm tra Tên đăng nhập Rỗng | 1. Nhập tên đăng nhập = Rỗng 2. Nhập mật khẩu= có trong DB 3. Nhấn vào đăng nhập | - Thông báo Tên đăng nhập chưa đc nhập - Set focus và highlight vào trường lỗi | Đạt |
| 6 | Kiểm tra Mật khẩu Rỗng | 1. Nhập tên đăng nhập = có trong DB 2. Nhập mật khẩu= Rỗng 3. Nhấn vào đăng nhập | - Thông báo Mật khẩu chưa đc nhập - Set focus và highlight vào trường lỗi | Đạt |
| 7 | Kiểm tra mã hóa textbox mật khẩu | 1. Nhập tên đăng nhập tồn tại trong DB 2. Nhập mật khẩu hợp lệ | - Textbox mật khẩu hiển thị dưới dạng mã hóa \*\*\*\*\*\*\*\* | Đạt |

3.8 2 Chức năng bảo trì sản phẩm

Bảng 3. 2 Kiểm thử chức năng bảo trì sản phẩm

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Mô tả trường hợp kiểm tra** | **Các bước thực hiện** | **Kết quả mong đợi** | **Kết quả** |
|
|
|  | **Tên sản phẩm** |  |  |  |
| 1 | Test nhập tên hợp lệ | 1. Nhập tên sản phẩm = ký tự 2. Các thông tin khác được nhập hợp lệ.  3. Click vào "Thêm" button | - Tạo tên sản phẩm thành công | Đạt |
| 2 | Test không nhập tên sản phẩm | 1. Tại ''Tên sản phẩm " = Rỗng 2. Các thông tin khác được nhập hợp lệ.  3. Click vào "Thêm" button | - Message thông báo "Tên sản phẩm không được để trống" - Hightlight màu đỏ và chuột focus vào ''Họ tên" textbox | Message thông báo "Tên sản phẩm không được để trống" |
|  | **Giá bán** |  |  |  |
| 3 | Test nhập giá bán hợp lệ | 1.Nhập giá bán hợp lệ = số dương 2. Các thông tin khác được nhập hợp lệ.  3. Click vào "Thêm" button | - Số lượng mà bạn nhập đã lưu trong CSDL | Đạt |
| 4 | Test nhập giá bán không hợp lệ | 1.Nhập giá bán hợp lệ = số dương 2. Các thông tin khác được nhập không hợp lệ.  3. Click vào "Thêm" button | - Message thông báo "Giá bán không hợp lệ " - Hightlight màu đỏ và chuột focus vào ''giá bán" | Đạt |
|  | **Ảnh Sản Phẩm** |  |  |  |
| 5 | Test tải ảnh lên hợp lệ | 1. Tải file đúng định dạng ảnh  2. Click vào "Thêm" button | - Thêm ảnh bìa thành công | Đạt |
| 6 | Test không tải ảnh lên | 1. Không tải file ảnh 2. Click vào "Thêm" button | - Message thông báo " Ảnh bìa không được để trống" - Hightlight màu đỏ | Message thông báo "Ảnh bìa không được để trống" |
|  | **Số lượng** |  |  |  |
| 7 | Test Số lượng hợp lệ | 1. Nhập Số lượng = ký tự 2. Các thông tin khác được nhập hợp lệ.  3. Click vào "Thêm" button | - Tạo số lượng thành công | Đạt |
| 8 | Test nhập số lượng không hợp lệ | 1. Tại “Số lượng tồn” = ký tự chữ 2. Các thông tin khác được nhập hợp lệ.  3. Click vào "Thêm" button | - Message thông báo "Sai định dạng số" - Hightlight màu đỏ và chuột focus vào “Số lượng” textbox | Đạt |
| 9 | Test không Số lượng | 1. Tại “Số lượng” = Rỗng 2. Các thông tin khác được nhập hợp lệ.  3. Click vào "Thêm" button | - Message thông báo “Số lượng tồn không được để trống" - Hightlight màu đỏ và chuột focus vào “Số lượng” textbox | Đạt |

* + 1. Chức năng tìm kiếm sản phẩm theo tên

Bảng 3. 3 Kiểm thử chức năng tìm kiếm sản phẩm theo tên

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Mô tả trường hợp kiểm tra** | **Các bước thực hiện** | **Kết quả mong đợi** | **Kết quả thu được** |
| 1 | Tìm kiếm sản phẩm theo tên chính xác | 1. Truy cập website  2. Nhập tên sản phẩm " Dior" 3. Nhấn nút tìm kiếm 4. Kiểm tra kết quả tìm kiếm | Các sản phẩm có tên chính xác là " Dior " | Các sản phẩm có tên chính xác là " Dior " |
| 2 | Tìm kiếm sản phẩm với từ khóa phụ | 1. Truy cập website  2. Nhập tên danh mục " Dior " 3. Nhấn nút tìm kiếm 4. Kiểm tra kết quả tìm kiếm | Danh sách các sản phẩm có tên chứa từ khóa "Dior" | Danh sách các sản phẩm có tên chứa từ khóa "Dior" |
| 3 | Tìm kiếm sản phẩm không tồn tại | 1. Truy cập website  2. Nhập tên sản phẩm không tồn tại vào ô tìm kiếm "XYZ123". 3. Nhấn nút tìm kiếm. 4. Kiểm tra kết quả tìm kiếm | Không có sản phẩm nào phù hợp | Không tìm thấy sản phẩm nào |

* + 1. Chức năng xem sản phẩm theo danh mục

Bảng 3. 4 Kiểm thử chức năng xem sản phẩm theo danh mục

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Tên test case** | **Các bước thực hiện** | **Kết quả mong đợi** | **Kết quả thu được** |
| 1 | Hiển thị danh mục sản phẩm trống | 1. Kích vào danh mục sản phẩm 2. Kiểm tra xem danh mục sản phẩm có rỗng không | Hiển thị thông báo "danh mục sản phẩm trống" | Không có danh mục nào rỗng |
| 2 | Sắp xếp danh mục theo thứ tự tăng dần | 1. Kích vào danh mục sản phẩm 2. Kiểm tra xem danh mục có được sắp xếp tăng dần theo tên sản phẩm không | Hiện thị danh sản phẩm theo thứ tự tăng dần tên sách | Hiện thị danh sách theo thứ tự tăng dần tên sản phẩm |
| 3 | Tìm kiếm sản phẩm trong danh mục | 1. Nhập tên sản phẩm cần tìm 2.Kiểm tra kết quả tìm kiếm có chứa tên cuốn sản phẩm đó | Hiển thị danh sách chỉ chứa tên sản phẩm đó | Hiển thị danh sách chỉ chứa tên sản phẩm đó |
| 4 | Tìm kiếm sản phẩm trong danh mục không có kết quả | 1. Nhập từ khóa tìm kiếm không có trong danh mục sách 2. Kiểm tra xem kết quả có rỗng không | Hiển thị thông báo "Không tìm thấy kết quả nào" | Hiển thị thông báo"Không tìm thấy sản phẩm nào" |
| 5 | Xem danh mục sách khi có nhiều sản phẩm | 1. Vào danh mục sách 2. Kiểm tra khả năng cuộn trang hoặc phân trang của danh mục sản phẩm | Hiển thị danh sách sản phẩm 1 cách trực quan, có khả năng cuộn trang hoặc phân trang | Hiển thị danh sách sản phẩm rõ ràng, có thể cuộn trang hoặc phân trang |
| 6 | Xem danh mục sách với các phân loại sản phẩm | 1. Vào danh mục sách 2. Kiểm tra xem danh sách sản phẩm có hiển thị các phân loại hoặc mục lục không | Hiển thị danh sách sản phẩm với các phân loại hoặc mục lục | Hiển thị danh sách sản phẩm với các phân loại và mục lục |

* + 1. Chức năng quản lý giỏ hàng

Bảng 3. 5 Kiểm thử chức năng quản lý giỏ hàng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Mô tả trường hợp kiểm tra** | **Các bước thực hiện** | **Kết quả mong đợi** | **Kết quả thực tế** |
| 1 | Kiểm tra thêm sản phẩm vào giỏ hàng | 1. Chọn một sản phẩm từ danh sách sản phẩm.  2. Nhấp vào nút "Thêm vào giỏ hàng". | Sản phẩm được thêm vào giỏ hàng với số lượng là 1 và hiển thị thông báo thành công. | Trang giỏ hàng hiển thị sản phẩm được thêm mới vào |
| 2 | Kiểm tra xóa sản phẩm khỏi giỏ hang | 1. Chọn một sản phẩm đã có trong giỏ hàng.  2. Nhấp vào nút "Xóa khỏi giỏ hàng" | Sản phẩm được xóa khỏi giỏ hàng và giỏ hàng được cập nhật. | Trang giỏ hàng đã được cập nhật với sản phẩm đã được xóa |
| 3 | Kiểm tra cập nhật số lượng sản phẩm trong giỏ hang | 1. Chọn một sản phẩm đã có trong giỏ hàng.  2. Nhập số lượng mới cho sản phẩm trong trường số lượng. | Số lượng sản phẩm trong giỏ hàng được cập nhật theo giá trị mới. | Trang giỏ hàng đã được cập nhật với số lượng sản phẩm hiện tại. |
| 4 | Kiểm tra tính năng tính tổng giá trị giỏ hàng | 1. Thêm một số sản phẩm vào giỏ hàng với các giá trị khác nhau.  2. Kiểm tra tổng giá trị của giỏ hàng. | Tổng giá trị của giỏ hàng được tính chính xác | Trang giỏ hàng hiển thị chính xác tổng giá trị của giỏ hàng. |
| 5 | Kiểm tra tính năng kiểm tra giỏ hàng trống | 1. Xóa tất cả các sản phẩm khỏi giỏ hàng.  2. Kiểm tra giỏ hàng có rỗng hay không | Giỏ hàng được xác định là trống và hiển thị thông báo tương ứng | Trang giỏ hàng hiển thị thông báo” Chưa có sản phẩm nào trong giỏ hàng” |
| 6 | Kiểm tra tính năng chuyển đến trang kiểm tra | 1. Thêm một số sản phẩm vào giỏ hàng 2. Nhấp vào nút "Kiểm tra" | Người dùng được chuyển đến trang kiểm tra với các sản phẩm trong giỏ hàng được chuyển thành đơn hàng | Người dùng được chuyển đến trang kiểm tra với các sản phẩm trong giỏ hàng được chuyển thành đơn hàng |

Kết luận chương 3

Trong chương 3 của báo cáo đã tiến hành xây dựng một website bán nước, bao gồm nhiều bước và phương pháp quan trọng để đảm bảo tính hiệu quả và thành công của dự án. Các nội dung chính đã được trình bày gồm có:

Phân tích đề tài:

Chương 3 bắt đầu bằng việc phân tích yêu cầu và mục tiêu của website bán nước, xác định các chức năng chính và nhu cầu của người dùng, từ đó đưa ra các yêu cầu cụ thể cho hệ thống.

Thiết kế cơ sở dữ liệu:

Tiếp theo là thiết kế cơ sở dữ liệu để quản lý thông tin sản phẩm, đơn hàng và khách hàng một cách hiệu quả. Các bảng dữ liệu và mối quan hệ giữa chúng đã được mô tả chi tiết để đảm bảo tính nhất quán và toàn vẹn dữ liệu.

Mô tả use case:

Chương 3 đã mô tả các trường hợp sử dụng (use case) của hệ thống, xác định cách thức tương tác giữa người dùng và hệ thống. Các use case này giúp làm rõ các chức năng cần thiết và quy trình thực hiện.

Biểu đồ trình tự :

Chương 3 đã xây dựng biểu đồ trình tự để mô tả luồng thông tin và tương tác giữa các thành phần trong hệ thống. Điều này giúp hiểu rõ hơn về các quy trình xử lý và cách các phần tử hệ thống phối hợp với nhau.

Cài đặt chương trình:

Về phần cài đặt chương trình đã cài đặt các thành phần của website, từ backend với ASP.NET Core Web API đến frontend với Vue.js. Việc cài đặt này đảm bảo hệ thống hoạt động ổn định và đáp ứng các yêu cầu đã đề ra.

Giao diện chương trình:

Về giao diện đã thiết kế và phát triển giao diện người dùng, đảm bảo tính thân thiện, dễ sử dụng và đáp ứng tốt các nhu cầu của người dùng khi tương tác với website.

Kiểm thử chương trình:

Cuối cùng là tiến hành kiểm thử toàn bộ hệ thống để đảm bảo không có lỗi và các chức năng hoạt động đúng như mong đợi. Các phương pháp kiểm thử đã được áp dụng để đảm bảo chất lượng và hiệu quả của website.

# KẾT LUẬN

Đề tài “Xây dựng website nước hoa” cũng xuất phát từ thực tế hiện nay, nhằm giúp người dùng có thể xem và mua được những sản phẩm nước hoa tốt nhất phù hợp với nhu cầu sử dụng và tài chính của mọi người.

Mặc dù đã có nhiều những cố gắng tìm hiểu các kiến thức đã học, kết hợp tra cứu các tài liệu chuyên ngành nhưng do hạn chế về thời gian cũng như khả năng và kinh nghiệm của bản thân, em đã hoàn thành sản phẩm của mình và đạt được một số kết quả như sau:

* Hoàn thành phân tích thiết kế hệ thống, thiết kế cơ sở dữ liệu, thiết kế giao diện.
* Hệ thống hiện tại đã xây dựng được các chức năng theo yêu cầu ban đầu về phía người dùng và người quản trị. Thiết kế rõ ràng, mô tả chi tiết và dễ hình dung.
* Hoạt động người quản trị: Quản lý tài khoản người dùng,các danh mục sản phẩm, tạo voucher giảm giá và theo dõi thống kê doanh số bán hàng .

# TÀI LIỆU THAM KHẢO

[1]. Hoàng Quang Huy (2016), “*Giáo trình cơ sở dữ liệu*”, H. Giáo dục Việt Nam 2011.

[2] W3Schools. (2023). W3Schools Online Web Tutorials. Truy cập từ <https://www.w3schools.com/>

[3]. Vũ Thị Dương, Phùng Đức Hòa, Nguyễn Thị Hương Lan (2015), “Giáo trình Phân tích thiết kế hướng đối tượng”, NXB Khoa học và Kỹ thuật.

# PHỤ LỤC