TRƯỜNG ĐẠI HỌC TRÀ VINH

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**ĐỒ ÁN MÔN HỌC: CHUYÊN ĐỀ ASP.NET**

**HỌC KỲ V, NĂM HỌC 2024-2026**

**Xây dựng website bán sản phẩm cà phê giải khát**

*Sinh viên thực hiện:*

Họ tên: Trương Hoàng Hải

MSSV: 170124247

Lớp: DK24TTC2

*Giảng viên hướng dẫn:*

TS. Đoàn Phước Miền

***TP.HCM, tháng 11 năm 2025***

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

*…………, ngày ….. tháng …… năm ……* **Giáo viên hướng dẫn**

*(Ký tên và ghi rõ họ tên)*

**NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG**

*…………, ngày ….. tháng …… năm ……* **Thành viên hội đồng**

*(Ký tên và ghi rõ họ tên)*

**LỜI CẢM ƠN**

Trong thời gian thực hiện báo cáo với sự giúp đỡ và tạo điều kiện của các thầy cô trường Đại Học Trà Vinh, sự góp ý của các bạn và đặc biệt là sự dạy dỗ, hướng dẫn và chỉ bảo tận tình của giảng viên bộ môn **TS. Đoàn Phước Miền** em đã hoàn thành đề tài cùng với bản báo cáo đúng thời gian quy định.

Với khả năng và thời gian có hạn nên không tránh khỏi những thiếu sót, em rất mong nhận được sự quan tâm, giúp đỡ, tạo điều kiện của Thầy để em hoàn thiện hơn đề tài nghiên cứu trong thời gian tới.

Một lần nữa, em xin gửi lời cảm ơn sâu sắc tới các thầy cô trường Đại Học Trà Vinh, đặc biệt là **TS. Đoàn Phước Miền** đã hướng dẫn em trsuốt quá trình làm báo cáo.

Em xin chân thành cám ơn!

Trương Hoàng Hải

**MỤC LỤC**

[1. Lý do chọn đề tài 1](#_Toc215067974)

[2. Mục tiêu nghiên cứu 1](#_Toc215067975)

[3. Đối tượng và phạm vi nghiên cứu 1](#_Toc215067976)

[4. Phương pháp nghiên cứu 2](#_Toc215067977)

[CHƯƠNG 1: NGHIÊN CỨU LÝ THUYẾT 4](#_Toc215067978)

[1.1. Giới thiệu về đề tài 4](#_Toc215067979)

[1.1.1. Tên đề tài 4](#_Toc215067980)

[1.1.2. Mục tiêu của Đề tài 4](#_Toc215067981)

[1.1.3. Một số trang chính của Website 4](#_Toc215067982)

[1.2. Giới thiệu về framework ASP .Net Core MVC 4](#_Toc215067983)

[1.2.1. Khái niệm về MVC: 4](#_Toc215067984)

[1.2.2. Các thành phần của MVC: 5](#_Toc215067985)

[1.2.3. Lịch sử phát triển của MVC: 5](#_Toc215067986)

[1.2.4. Luồng xử lý trong MVC: 6](#_Toc215067987)

[1.3. Công nghệ ASP.NET MVC 7](#_Toc215067988)

[1.3.1. Tổng quan 7](#_Toc215067989)

[1.3.2. Lịch sử phát triển của ASP.NET MVC 7](#_Toc215067990)

[1.3.3. Tính năng của ASP.NET MVC 8](#_Toc215067991)

[1.4. Giới thiệu ngôn ngữ lập trình C# (C Sharp) 10](#_Toc215067992)

[1.4.1. Tổng quan 10](#_Toc215067993)

[1.4.2. Giới thiệu các Controls của ASP.NET 11](#_Toc215067994)

[1.5. Giới thiệu ADO.NET và các đối tượng truy cập CSDL 12](#_Toc215067995)

[1.5.1. Khái niệm về ADO.NET 12](#_Toc215067996)

[1.5.2. Mô hình hoạt động của ADO.NET 13](#_Toc215067997)

[1.5.3. Các đối tượng chính trong ADO.NET 13](#_Toc215067998)

[1.5.4. Quy trình truy cập dữ liệu với ADO.NET 14](#_Toc215067999)

[1.6. Giới thiệu về SQL Server 15](#_Toc215068000)

[1.6.1. Tổng quan 15](#_Toc215068001)

[1.6.2. Lịch sử phát triển 15](#_Toc215068002)

[1.6.3. Cấu trúc của SQL Server 15](#_Toc215068003)

[1.6.4. Kết luận 16](#_Toc215068004)

[1.7. Giới thiệu về Bootstrap 16](#_Toc215068005)

[1.7.1. Khái niệm về Bootstrap 16](#_Toc215068006)

[1.7.2. Mục tiêu và ưu điểm của Bootstrap 16](#_Toc215068007)

[1.7.3. Cấu trúc và thành phần của Bootstrap 17](#_Toc215068008)

[1.7.4. Ứng dụng của Bootstrap 17](#_Toc215068009)

[1.8. Giới thiệu về Entity Framework 17](#_Toc215068010)

[1.8.1. Giới thiệu về Entity Framework 17](#_Toc215068011)

[1.8.2. Mục tiêu và vai trò của Entity Framework 18](#_Toc215068012)

[CHƯƠNG 2: HIỆN THỰC HÓA NGHIÊN CỨU 19](#_Toc215068013)

[2.1. Mô tả bài toán 19](#_Toc215068014)

[2.2. Yêu cầu chức năng 19](#_Toc215068015)

[2.3. Cài đặt 20](#_Toc215068016)

[2.4. Thử Nghiệm 20](#_Toc215068017)

[2.5. Mô hình cơ sở dữ liệu 22](#_Toc215068018)

[2.6. Lược Đồ USE CASE 28](#_Toc215068019)

[2.6.1. Lược Đồ USE CASE – Website 28](#_Toc215068020)

[2.6.2. Lược Đồ USE CASE – Người Dùng (USER) 29](#_Toc215068021)

[2.6.3. Lược Đồ USE CASE – QUẢN TRỊ (Administration) 30](#_Toc215068022)

[2.7. Kiến Trúc Hệ Thống 31](#_Toc215068023)

[2.8. Các bước triển khai hệ thống được thực hiện theo trình tự sau: 32](#_Toc215068024)

[2.8.1. Phân tích yêu cầu và xây dựng mô hình hệ thống 32](#_Toc215068025)

[2.8.2. Thiết kế kỹ thuật và mô hình dữ liệu 32](#_Toc215068026)

[2.8.3. Phát triển giao diện và chức năng ứng dụng 32](#_Toc215068027)

[2.8.4. Kiểm thử, xử lý lỗi và tối ưu hóa hệ thống 32](#_Toc215068028)

[2.8.5. Triển khai và chạy thử trên môi trường thực tế 33](#_Toc215068029)

[CHƯƠNG 3: KẾT QUẢ NGHIÊN CỨU 34](#_Toc215068030)

[3.1.1. Giao diện đăng nhập và đăng ký 34](#_Toc215068031)

[3.1.2. Giao diện Trang Quản Trị 35](#_Toc215068032)

[3.1.3. Giao diện Trang Chủ 36](#_Toc215068033)

[3.1.4. Giao diện Trang Chỉnh Sửa Thông Tin Người Dùng 36](#_Toc215068034)

[3.1.5. Giao diện Trang Quản Lý Thông Tin Sản Phẩm 37](#_Toc215068035)

[3.1.6. Giao diện Trang Chỉnh Thông Tin Sản Phẩm 37](#_Toc215068036)

[3.1.7. Giao diện Trang Thêm Mới Sản Phẩm 38](#_Toc215068037)

[3.1.8. Giao diện Trang Chi Tiết Sản Phẩm 38](#_Toc215068038)

[3.1.9. Giao diện Trang Quản Lý Đơn Hàng 39](#_Toc215068039)

[3.1.10. Giao diện Trang Giỏ Hàng 40](#_Toc215068040)

[3.1.11. Giao diện Trang Profile Người Dùng 40](#_Toc215068041)

[CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 41](#_Toc215068042)

[4.1. Các kết quả đạt được 41](#_Toc215068043)

[4.2. Hạn chế, hướng phát triển 41](#_Toc215068044)

[4.2.1. Hạn chế: 41](#_Toc215068045)

[4.2.2. Hướng phát triển: 41](#_Toc215068046)

[DANH MỤC TÀI LIỆU THAM KHẢO 42](#_Toc215068047)

**DANH MỤC HÌNH ẢNH**

[Hình 1. Bảng dữ liệu tài khoản người dùng 21](#_Toc215068071)

[Hình 2. Bảng dữ liệu đơn hàng 21](#_Toc215068072)

[Hình 3. Bảng dữ liệu sản phẩm 21](#_Toc215068073)

[Hình 4. Mô hình cơ sở dữ liệu 22](#_Toc215068074)

[Hình 5. Lược Đồ USE CASE - Website 28](#_Toc215068075)

[Hình 6. Lược Đồ USE CASE - Người Dùng (USER) 29](#_Toc215068076)

[Hình 7. Lược Đồ USE CASE - QUẢN TRỊ (Administration) 30](#_Toc215068077)

[Hình 8. Mô Hình Kiến Trúc Tổng Quan Hệ Thống 31](#_Toc215068078)

[Hình 9. Giao Diện Đăng Ký Tài Khoản 34](#_Toc215068079)

[Hình 10. Giao Diện Đăng Nhập 35](#_Toc215068080)

[Hình 11. Giao Diện Trang Quản Trị 35](#_Toc215068081)

[Hình 12. Giao Diện Trang Chủ 36](#_Toc215068082)

[Hình 13. Giao Diện Trang Quản Lý Thông Tin Người Dùng 36](#_Toc215068083)

[Hình 14. Giao diện Trang Quản Lý Thông Tin Sản Phẩm 37](#_Toc215068084)

[Hình 15. Giao diện Trang Chỉnh Sửa Thông Tin Sản Phẩm 37](#_Toc215068085)

[Hình 16. Giao diện Trang Thêm Mới Sản Phẩm 38](#_Toc215068086)

[Hình 17. Giao diện Trang Chi Tiết Sản Phẩm 38](#_Toc215068087)

[Hình 18. Giao diện Trang Quản Lý Đơn Hàng 39](#_Toc215068088)

[Hình 19. Giao diện Trang Quản Lý Đơn Hàng Chi Tiết 39](#_Toc215068089)

[Hình 20. Giao diện Trang Giỏ Hàng 40](#_Toc215068090)

[Hình 21. Giao diện Trang Profile Người Dùng 40](#_Toc215068091)

**DANH MỤC BẢNG**

[Table 1. Bảng Web Server Control 12](#_Toc215068098)

[Table 2. Bảng Validation Control 12](#_Toc215068099)

[Table 3. Bảng các đối tượng chính trong ADO.NET 14](#_Toc215068100)

**MỞ ĐẦU**

1. Lý do chọn đề tài

Với những thao tác đơn giản trên máy tính, điện thoại thông minh có kết nối mạng Internet. Chúng ta sẽ tận tay mua những gì mình cần mà không phải mất nhiều thời gian. Chỉ cần vào các trang dịch vụ thương mại điện tử, làm theo hướng dẫn và click vào những gì chúng ta cần. Các nhà dịch vụ sẽ giao sản phẩm đến tận nhà cho chúng ta.

Để tiếp cận và góp phần đẩy mạnh sự phát triển, phổ biến của thương mại điện tử ở Việt Nam, em đã tìm hiểu và xây dựng đề tài “Xây dựng website bán sản phẩm cà phê giải khát” với mặt hàng là các loại cà phê, nước giải khát.

1. Mục tiêu nghiên cứu

Xây dựng một website bán hàng hoàn chỉnh bằng cách vận dụng kiến thức đã học về lập trình web, cơ sở dữ liệu SQL Server, mô hình MVC, HTML/CSS, Bootstrap, ngôn ngữ C# và nền tảng ASP.NET. Thông qua đó, sinh viên có cơ hội tiếp cận quy trình phát triển một hệ thống thương mại điện tử thực tế.

Về giao diện, website cần được thiết kế thân thiện và trực quan, đảm bảo người dùng dễ dàng thao tác. Song song đó, trang quản trị cũng phải rõ ràng, thuận tiện cho việc quản lý. Điều này giúp nâng cao trải nghiệm người dùng và tạo ấn tượng chuyên nghiệp cho hệ thống.

Về chức năng, đề tài tập trung xây dựng các tính năng cốt lõi của một website bán hàng, bao gồm: tìm kiếm và xem chi tiết sản phẩm, quản lý giỏ hàng, đặt mua hàng, quản lý đơn hàng và quản lý thông tin tài khoản người dùng.

Trong quá trình thực hiện, sinh viên tìm hiểu và áp dụng công nghệ ASP.NET, nắm được cách triển khai ứng dụng web theo mô hình MVC, thực hiện kết nối và xử lý dữ liệu với SQL Server, đồng thời hiểu rõ hơn cách tổ chức nghiệp vụ và quy trình vận hành của một ứng dụng web thương mại điện tử.

1. Đối tượng và phạm vi nghiên cứu

Đề tài tập trung nghiên cứu toàn bộ quy trình thiết kế và phát triển một website thương mại điện tử chuyên bán cà phê và nước giải khát, được xây dựng bằng công nghệ ASP.NET. Các đối tượng nghiên cứu chủ yếu bao gồm:

• *Người dùng cuối (khách hàng)*

Đây là nhóm người trực tiếp truy cập và sử dụng website để tìm kiếm sản phẩm, xem thông tin chi tiết, đặt mua hàng và theo dõi đơn hàng. Việc nghiên cứu nhu cầu và thói quen sử dụng của họ giúp xây dựng giao diện và tính năng thân thiện, mang đến trải nghiệm tối ưu.

• *Quản trị viên (admin)*

Admin là người vận hành hệ thống, chịu trách nhiệm quản lý sản phẩm, danh mục, đơn hàng và thông tin người dùng. Việc tìm hiểu yêu cầu và công việc của quản trị viên giúp tạo ra một trang quản trị dễ dùng, hoạt động hiệu quả và hỗ trợ tốt cho công tác quản lý.

• *Hệ thống website bán hàng*

Bao gồm quá trình tìm hiểu và xây dựng các chức năng cốt lõi của một website thương mại điện tử: hiển thị danh sách sản phẩm, giỏ hàng, xử lý đặt hàng, đăng ký – đăng nhập, quản lý dữ liệu, kết nối cơ sở dữ liệu… Đồng thời, nghiên cứu việc tổ chức kiến trúc phần mềm, cấu trúc mã nguồn và cách xử lý tương tác giữa giao diện với dữ liệu.

• *Công nghệ ASP.NET và các công cụ liên quan*

Tập trung nghiên cứu cách vận dụng ASP.NET (Web Forms hoặc MVC), ngôn ngữ C#, HTML/CSS, Bootstrap, JavaScript cùng SQL Server trong quá trình phát triển ứng dụng. Ngoài ra, khai thác các framework, thư viện và công cụ hỗ trợ nhằm nâng cao tốc độ lập trình, đảm bảo bảo mật và tối ưu hiệu suất của hệ thống.

Đối tượng nghiên cứu chủ yếu là các sản phẩm công nghệ, điện thoại. Nắm bắt tìm hiểu về Framework ASP .Net core.

1. Phương pháp nghiên cứu

• *Phương pháp nghiên cứu tài liệu*

Tiến hành tìm hiểu, thu thập và tổng hợp kiến thức từ các nguồn liên quan đến C#, ASP.NET, mô hình MVC, HTML/CSS, JavaScript và SQL Server.

Tham khảo giáo trình, tài liệu trực tuyến, diễn đàn lập trình và các mẫu website thương mại điện tử để học hỏi cách xây dựng, triển khai và tổ chức một hệ thống bán hàng hoàn chỉnh.

• *Phương pháp phân tích hệ thống*

Phân tích nhu cầu của người dùng cuối và quản trị viên nhằm xác định đầy đủ các chức năng mà hệ thống cần có.

Sử dụng các mô hình phân tích như sơ đồ chức năng, sơ đồ luồng dữ liệu (DFD) và mô hình ERD để thiết kế hệ thống một cách khoa học, đảm bảo dễ dàng cho quá trình hiện thực hóa và mở rộng về sau.

• *Phương pháp thiết kế và lập trình*

Vận dụng quy trình phát triển phần mềm để thiết kế giao diện, tạo cấu trúc cơ sở dữ liệu và lập trình các chức năng chính dựa trên mô hình phân lớp (3-tier) hoặc MVC.

Sử dụng Visual Studio làm môi trường phát triển và SQL Server để quản lý dữ liệu, đồng thời kiểm thử ứng dụng trong quá trình xây dựng.

• *Phương pháp thực nghiệm*

Thực hiện các bài kiểm thử trực tiếp trên hệ thống để đánh giá hoạt động của những chức năng như tìm kiếm sản phẩm, đặt hàng, quản lý sản phẩm và xử lý đơn hàng.

Ghi lại lỗi phát sinh trong khi thử nghiệm, tiến hành sửa chữa và tối ưu hệ thống nhằm nâng cao hiệu suất và cải thiện giao diện người dùng.

1. NGHIÊN CỨU LÝ THUYẾT
   1. Giới thiệu về đề tài
      1. Tên đề tài

Xây dựng website bán sản phẩm cà phê giải khát.

* + 1. Mục tiêu của Đề tài

Xây dựng và thiết kế website bán các sản phẩm cà phê giải khát.

* + 1. Một số trang chính của Website
* Trang chủ: Đây là trang giới thiệu sơ lược về các sản phẩm nổi bật, các sản phẩm bán chạy.
* Trang Sản phẩm: Đây là trang mà khách hàng có thể lựa chọn các sản phẩm như: cà phê, nước giải khát, …
* Trang Tin tức: Hiển thị các tin tức liên quan đến các loại nước giải khát, chức năng dinh dưỡng….
* Trang Giỏ hàng: Quản lý các sản phẩm mà người dụng đã chọn.
* Trang Đăng nhập: Người dùng đăng nhập tài khoản để đặt mua sản phẩm của trang hoặc đăng nhập vào trang quản trị đối với Người quản trị.
  1. Giới thiệu về framework ASP .Net Core MVC
     1. Khái niệm về MVC:

**MVC** (Model-View-Controller) là một mô hình thiết kế phần mềm được sử dụng để phát triển ứng dụng web và phần mềm, giúp tổ chức mã nguồn theo cách có cấu trúc, dễ bảo trì và mở rộng. Mô hình này chia ứng dụng thành ba thành phần chính: **Model**, **View**, và **Controller**. Mỗi thành phần có trách nhiệm riêng biệt, giúp tách biệt logic xử lý, giao diện và dữ liệu, từ đó cải thiện khả năng mở rộng và dễ dàng bảo trì phần mềm.

* **Model**: Quản lý dữ liệu và logic nghiệp vụ của ứng dụng.
* **View**: Chịu trách nhiệm hiển thị giao diện người dùng.
* **Controller**: Quản lý các yêu cầu từ người dùng và điều phối giữa Model và View.
  + 1. Các thành phần của MVC:
* **Model (Mô hình)**:
* **Chức năng**: Model đại diện cho dữ liệu và logic nghiệp vụ của ứng dụng. Nó xử lý mọi thao tác với dữ liệu, bao gồm lấy dữ liệu từ cơ sở dữ liệu, thực hiện tính toán và logic nghiệp vụ.
* **Ví dụ**: Một lớp Product trong một ứng dụng thương mại điện tử sẽ chứa thông tin về sản phẩm (tên, giá, mô tả) và các phương thức để truy xuất dữ liệu từ cơ sở dữ liệu.
* **View (Giao diện người dùng)**:
* **Chức năng**: View chịu trách nhiệm hiển thị dữ liệu cho người dùng và nhận đầu vào từ người dùng. Nó không bao giờ trực tiếp xử lý dữ liệu, mà chỉ hiển thị dữ liệu được truyền từ Controller.
* **Ví dụ**: Một trang HTML có danh sách sản phẩm, hoặc các mẫu Razor trong ASP.NET MVC, nơi bạn có thể kết hợp HTML và C# để tạo giao diện động.
* **Controller (Điều khiển)**:
* **Chức năng**: Controller nhận các yêu cầu từ người dùng (thông qua các hành động như nhấn nút, gửi biểu mẫu), xử lý các yêu cầu này (thường là tương tác với Model để lấy dữ liệu), và trả về một View để hiển thị kết quả cho người dùng.
* **Ví dụ**: Một Controller như ProductController có thể xử lý các yêu cầu như "hiển thị danh sách sản phẩm" hoặc "thêm một sản phẩm mới".
  + 1. Lịch sử phát triển của MVC:
* **1979**: Mô hình MVC được giới thiệu lần đầu tiên bởi Trygve Reenskaug tại Xerox PARC (Palo Alto Research Center). Ban đầu, MVC được áp dụng trong môi trường phát triển ứng dụng GUI (giao diện người dùng đồ họa) để tổ chức mã nguồn và tách biệt các phần của ứng dụng.
* **1990s**: MVC bắt đầu được sử dụng rộng rãi trong các ứng dụng web với sự phát triển của các framework đầu tiên như **Ruby on Rails** và **ASP.NET MVC**. Các framework này áp dụng nguyên lý MVC để giúp các lập trình viên dễ dàng xây dựng các ứng dụng web có cấu trúc rõ ràng và dễ duy trì.
* **Hiện nay**: MVC đã trở thành mô hình phổ biến không chỉ trong phát triển ứng dụng web, mà còn trong các ứng dụng di động và phần mềm máy tính để bàn. Các framework hiện đại như Angular, React, Vue.js, ASP.NET MVC, Spring MVC đều áp dụng nguyên lý MVC hoặc các biến thể của nó để xây dựng các ứng dụng dễ mở rộng và bảo trì.
  + 1. Luồng xử lý trong MVC:

Luồng xử lý trong MVC mô tả cách các yêu cầu từ người dùng được xử lý và chuyển đổi thành kết quả cuối cùng. Dưới đây là mô tả quy trình xử lý yêu cầu trong mô hình MVC:

* + - 1. Người dùng gửi yêu cầu:

Khi người dùng truy cập một URL hoặc tương tác với giao diện (ví dụ: nhấn nút, gửi biểu mẫu), yêu cầu này sẽ được gửi đến ứng dụng web.

* + - 1. Router định tuyến yêu cầu:

Hệ thống **routing** của ứng dụng web (thường được cấu hình trước) sẽ quyết định yêu cầu đó cần được chuyển đến Controller nào. Cấu trúc URL sẽ được ánh xạ đến một phương thức trong Controller cụ thể (action method).

* + - 1. Controller xử lý yêu cầu:

Controller nhận yêu cầu từ người dùng, xử lý các logic cần thiết, có thể truy xuất dữ liệu từ Model hoặc thực hiện các tác vụ khác như tính toán hoặc xác thực dữ liệu.

* + - 1. Model tương tác với dữ liệu:

Nếu Controller cần truy xuất hoặc thay đổi dữ liệu, nó sẽ gọi các phương thức trong Model. Model sẽ kết nối với cơ sở dữ liệu để lấy hoặc lưu trữ thông tin. Sau khi dữ liệu được xử lý, nó sẽ được trả về Controller.

* + - 1. Controller gửi dữ liệu đến View:

Sau khi nhận được dữ liệu từ Model, Controller sẽ quyết định View nào cần được hiển thị cho người dùng. Controller sẽ chuyển dữ liệu (thường là trong một đối tượng hoặc tập hợp) đến View.

* + - 1. View hiển thị kết quả:

View nhận dữ liệu từ Controller và sử dụng chúng để tạo giao diện người dùng động. View sẽ trả lại HTML (hoặc mã giao diện phù hợp) cho trình duyệt của người dùng để hiển thị.

* + - 1. Kết quả được trả về cho người dùng:

Trình duyệt của người dùng nhận được HTML từ View và hiển thị nó lên giao diện. Người dùng có thể tiếp tục tương tác với ứng dụng, tạo ra các yêu cầu mới, bắt đầu chu trình lại từ bước 1.

* 1. Công nghệ ASP.NET MVC
     1. Tổng quan

ASP.NET MVC (Model–View–Controller) là một framework phát triển ứng dụng web được Microsoft giới thiệu, dựa trên nền tảng ASP.NET.

ASP.NET MVC áp dụng mô hình kiến trúc MVC, trong đó:

Model: Chịu trách nhiệm quản lý dữ liệu và xử lý các logic nghiệp vụ của ứng dụng.

View: Đảm nhiệm việc hiển thị giao diện và thông tin cho người dùng.

Controller: Đóng vai trò trung gian, tiếp nhận yêu cầu từ người dùng, xử lý và điều phối dữ liệu giữa Model và View.

Mục tiêu của ASP.NET MVC là hỗ trợ lập trình viên:

Xây dựng các ứng dụng web có tính linh hoạt, dễ dàng mở rộng và thuận tiện cho việc kiểm thử.

Khai thác tối đa sức mạnh của nền tảng .NET Framework, và sau này là .NET Core, để tạo ra các ứng dụng hiệu năng cao và hiện đại.

* + 1. Lịch sử phát triển của ASP.NET MVC
       - 2009: Microsoft chính thức giới thiệu ASP.NET MVC tại hội nghị MIX, với phiên bản 1.0 được phát hành vào giữa năm. Đây là bước chuyển quan trọng từ Web Forms sang mô hình MVC hiện đại.
       - 2010: Phiên bản ASP.NET MVC 2 được ra mắt với nhiều cải tiến về kiểm tra dữ liệu (validation), hỗ trợ Data Annotations và tính năng Area nhằm tổ chức dự án tốt hơn.
       - 2011: ASP.NET MVC 3 đánh dấu sự xuất hiện của Razor View Engine, mang đến cú pháp gọn nhẹ hơn. Phiên bản này cũng hỗ trợ Dependency Injection và các bộ lọc (filters) nâng cao.
       - 2012: Với ASP.NET MVC 4, Microsoft bổ sung Web API, cho phép xây dựng các dịch vụ RESTful một cách chuyên nghiệp. Đồng thời, phiên bản này hỗ trợ template dành cho thiết bị di động.
       - 2013: ASP.NET MVC 5 tiếp tục cải tiến với ASP.NET Identity (hệ thống quản lý người dùng mới), hỗ trợ Attribute Routing và cập nhật giao diện mặc định dựa trên Bootstrap.
       - 2016: Sự ra đời của ASP.NET Core MVC trong .NET Core 1.0 mở ra kỷ nguyên mới: phát triển đa nền tảng, hiệu năng cao, và cấu trúc modular dựa trên NuGet, hợp nhất MVC và Web API.
       - Hiện nay: ASP.NET Core MVC liên tục được nâng cấp cùng các phiên bản .NET 5/6/7/8, tập trung vào tối ưu hiệu năng, đơn giản hóa lập trình và hỗ trợ Minimal APIs cho ứng dụng nhẹ, tốc độ cao.
    2. Tính năng của ASP.NET MVC

Công nghệ ASP.NET MVC cung cấp một loạt các tính năng mạnh mẽ giúp phát triển ứng dụng web trở nên linh hoạt, hiệu quả và dễ bảo trì hơn. Dưới đây là những tính năng nổi bật của ASP.NET MVC:

* + - 1. Hệ thống định tuyến (Routing) linh hoạt

Routing trong ASP.NET MVC cho phép ánh xạ các URL thân thiện với người dùng đến các Action trong Controller. Điều này không chỉ giúp tổ chức ứng dụng rõ ràng mà còn cải thiện khả năng SEO (tối ưu hóa công cụ tìm kiếm), giúp website dễ dàng được tìm thấy trên các công cụ tìm kiếm.

* + - 1. Model Binding và Validation

Model Binding giúp tự động chuyển đổi dữ liệu từ biểu mẫu (form) hoặc tham số URL thành các đối tượng Model trong ứng dụng, giảm thiểu việc phải viết code thủ công.

Validation được hỗ trợ thông qua Data Annotations, giúp dễ dàng kiểm tra và xác thực dữ liệu đầu vào ngay khi nhận được từ người dùng.

* + - 1. Razor View Engine

Razor View Engine là một công cụ mạnh mẽ trong ASP.NET MVC, cho phép chèn trực tiếp mã C# vào HTML thông qua cú pháp ngắn gọn và dễ đọc. Điều này nâng cao hiệu quả trong việc hiển thị dữ liệu động và giúp tạo giao diện người dùng trực quan và dễ dàng duy trì.

* + - 1. Controller và Action

Controller và Action trong ASP.NET MVC đóng vai trò điều phối luồng xử lý của ứng dụng. Chúng nhận yêu cầu từ người dùng, thực hiện các thao tác nghiệp vụ cần thiết và trả về View hoặc dữ liệu JSON cho người dùng.

* + - 1. Bộ lọc (Filters)

ASP.NET MVC cung cấp cơ chế Filter cho phép lập trình viên can thiệp vào các giai đoạn trước hoặc sau khi Action thực thi. Bộ lọc thường được sử dụng để xử lý các tác vụ như xác thực người dùng, ghi log, hoặc bộ nhớ đệm (cache), giúp tối ưu hiệu suất và bảo mật của ứng dụng.

* + - 1. Bảo mật và xác thực (Security & Authentication)

ASP.NET MVC chú trọng đến bảo mật ứng dụng với các hệ thống xác thực người dùng và mã hóa dữ liệu hiện đại như ASP.NET Identity và OAuth. Điều này giúp bảo vệ ứng dụng khỏi các mối đe dọa và đảm bảo tính toàn vẹn của dữ liệu người dùng.

* + - 1. Kiểm thử đơn vị (Unit Testing)

Nhờ vào kiến trúc phân tách rõ ràng giữa Model, View, và Controller, ASP.NET MVC hỗ trợ Testability cao. Các thành phần có thể được kiểm thử độc lập, giúp dễ dàng thực hiện Unit Test, từ đó cải thiện chất lượng và độ tin cậy của ứng dụng.

* + - 1. Tích hợp với công nghệ hiện đại

ASP.NET MVC hỗ trợ tích hợp linh hoạt với các công nghệ JavaScript, AJAX, jQuery, và SignalR, giúp xây dựng các ứng dụng web động và tương tác cao. Ngoài ra, framework này cũng dễ dàng tích hợp với các frontend frameworks như Angular, React, và Vue.js, hỗ trợ xây dựng ứng dụng web hiện đại và đáp ứng nhanh với yêu cầu người dùng.

* 1. Giới thiệu ngôn ngữ lập trình C# (C Sharp)
     1. Tổng quan

C# (đọc là “C Sharp”) là một ngôn ngữ lập trình hướng đối tượng được Microsoft phát triển, do Anders Hejlsberg (người từng thiết kế Turbo Pascal và Delphi) khởi xướng vào năm 2000. C# là một phần trong nền tảng .NET Framework, ra đời với mục tiêu tạo ra một ngôn ngữ hiện đại, mạnh mẽ, an toàn và dễ sử dụng, đồng thời khắc phục những hạn chế của các ngôn ngữ lập trình trước đó như C++ hay Visual Basic.

C# được thiết kế dựa trên cú pháp của ngôn ngữ C và C++, đồng thời chịu ảnh hưởng từ Java, nên lập trình viên có thể dễ dàng tiếp cận và học tập. Ngôn ngữ này hỗ trợ đầy đủ các nguyên lý của lập trình hướng đối tượng (OOP) như đóng gói (Encapsulation), kế thừa (Inheritance) và đa hình (Polymorphism). Ngoài ra, C# còn cung cấp nhiều tính năng tiên tiến như thu gom rác tự động (Garbage Collection), quản lý bộ nhớ an toàn, xử lý sự kiện và ngoại lệ, cùng LINQ (Language Integrated Query) giúp thao tác dữ liệu dễ dàng hơn.

Ban đầu, C# chủ yếu được sử dụng để phát triển ứng dụng Windows thông qua .NET Framework. Tuy nhiên, với sự ra đời của .NET Core và hiện nay là .NET (từ phiên bản .NET 5 trở đi), C# đã trở thành ngôn ngữ lập trình đa nền tảng, cho phép xây dựng các ứng dụng Web, Desktop, Mobile, Game và Cloud chạy trên Windows, Linux và macOS.

Một ưu điểm nổi bật của C# là khả năng tích hợp chặt chẽ với hệ sinh thái Microsoft, bao gồm các công nghệ như ASP.NET (phát triển web), Xamarin (ứng dụng di động), Unity (phát triển game), và Azure (dịch vụ điện toán đám mây). Với cú pháp rõ ràng, hiệu năng cao, cộng đồng lớn mạnh và sự hỗ trợ liên tục từ Microsoft, C# ngày nay là một trong những ngôn ngữ lập trình phổ biến và được ưa chuộng nhất thế giới, đặc biệt trong lĩnh vực phát triển phần mềm doanh nghiệp và ứng dụng hiện đại.

* + 1. Giới thiệu các Controls của ASP.NET
       1. Tổng quan

Trong công nghệ ASP.NET, Control (điều khiển) là các thành phần giao diện được sử dụng để tạo, hiển thị và tương tác với người dùng trên trang web. Các control giúp lập trình viên dễ dàng xây dựng giao diện động mà không cần viết quá nhiều mã HTML hoặc JavaScript thủ công. ASP.NET cung cấp một tập hợp phong phú các server controls, HTML controls và validation controls nhằm đáp ứng đa dạng nhu cầu phát triển web.

* + - 1. 1.4.1.2 HTML Controls

HTML Controls là các thẻ HTML thông thường được mở rộng để hoạt động như server controls bằng cách thêm thuộc tính runat="server". Khi đó, ASP.NET có thể truy cập và xử lý các phần tử này trên máy chủ.

Giúp lập trình viên điều khiển nội dung và hành vi của thẻ HTML trực tiếp từ mã C# trên server.

* + - 1. 1.4.1.3 Web Server Controls

Đây là nhóm control được ASP.NET cung cấp sẵn, có khả năng tự động sinh HTML và tương tác với người dùng mạnh mẽ hơn HTML Controls.

Một số Web Server Controls thông dụng gồm:

| **Nhóm** | **Tên Control** | **Chức năng** |
| --- | --- | --- |
| **Control nhập liệu** | TextBox, Button, CheckBox, RadioButton, DropDownList, ListBox | Cho phép người dùng nhập hoặc chọn dữ liệu. |
| **Control hiển thị** | Label, Image, HyperLink, Literal | Hiển thị nội dung văn bản, hình ảnh, liên kết. |
| **Container Controls** | Panel, PlaceHolder, MultiView | Dùng để nhóm và bố trí các control khác. |
| **Control dữ liệu (Data Controls)** | GridView, DataList, Repeater, FormView, DetailsView | Hiển thị và thao tác với dữ liệu từ cơ sở dữ liệu. |
| **Control điều hướng** | Menu, TreeView, SiteMapPath | Giúp người dùng di chuyển giữa các trang. |
| **Login Controls** | Login, LoginView, LoginStatus, CreateUserWizard | Dùng để xác thực người dùng và quản lý phiên đăng nhập. |

Table 1. Bảng Web Server Control

* + - 1. Validation Controls

Nhóm Validation Controls dùng để kiểm tra tính hợp lệ của dữ liệu nhập từ người dùng trước khi gửi lên server, giúp giảm lỗi và tăng tính bảo mật.

Một số control thường dùng gồm:

| **Control** | **Mục đích** |
| --- | --- |
| **RequiredFieldValidator** | Bắt buộc người dùng phải nhập dữ liệu. |
| **RangeValidator** | Kiểm tra giá trị trong một khoảng cho phép. |
| **CompareValidator** | So sánh giá trị của 2 control. |
| **RegularExpressionValidator** | Kiểm tra dữ liệu theo biểu thức chính quy (regex). |
| **CustomValidator** | Cho phép tự định nghĩa quy tắc kiểm tra riêng. |
| **ValidationSummary** | Hiển thị danh sách các lỗi kiểm tra trên trang. |

Table 2. Bảng Validation Control

* 1. Giới thiệu ADO.NET và các đối tượng truy cập CSDL
     1. Khái niệm về ADO.NET

ADO.NET (ActiveX Data Objects .NET) là một công nghệ truy cập và xử lý dữ liệu trong nền tảng .NET Framework do Microsoft phát triển.

Nó cho phép các ứng dụng giao tiếp, kết nối và thao tác dữ liệu với nhiều hệ quản trị cơ sở dữ liệu khác nhau như SQL Server, Oracle, MySQL, Access,…

ADO.NET được xây dựng dựa trên mô hình hướng đối tượng, cung cấp một tập hợp các lớp (classes) trong namespace System.Data và các namespace mở rộng như System.Data.SqlClient, System.Data.OleDb,... giúp lập trình viên dễ dàng thực hiện các thao tác như:

- Kết nối đến CSDL.

- Thực thi truy vấn SQL (SELECT, INSERT, UPDATE, DELETE).

- Lấy dữ liệu và hiển thị lên giao diện.

- Cập nhật dữ liệu ngược trở lại CSDL.

Khác với ADO (phiên bản cũ trong VB6), ADO.NET hỗ trợ mô hình ngắt kết nối (disconnected model), giúp tối ưu hiệu năng, giảm tải cho máy chủ cơ sở dữ liệu và phù hợp với các ứng dụng web.

* + 1. Mô hình hoạt động của ADO.NET

ADO.NET hoạt động dựa trên hai mô hình chính:

- Connected Model (mô hình kết nối):

Ứng dụng kết nối trực tiếp đến CSDL, thực thi câu lệnh SQL, lấy kết quả và ngắt kết nối ngay sau đó.

→ Phù hợp cho thao tác nhanh, dữ liệu nhỏ.

- Disconnected Model (mô hình ngắt kết nối):

Dữ liệu được tải về bộ nhớ tạm (DataSet hoặc DataTable), cho phép thao tác, xử lý offline và đồng bộ lại khi cần.

→ Phù hợp cho ứng dụng web và hệ thống nhiều người dùng.

* + 1. Các đối tượng chính trong ADO.NET

|  |  |  |
| --- | --- | --- |
| **Đối tượng** | **Namespace** | **Chức năng** |
| **Connection** (SqlConnection, OleDbConnection) | System.Data.SqlClient / System.Data.OleDb | Tạo và quản lý kết nối giữa ứng dụng và CSDL. |
| **Command** (SqlCommand, OleDbCommand) | Cùng namespace với Connection | Thực thi các câu lệnh SQL hoặc stored procedure (lệnh lưu trữ). |
| **DataReader** (SqlDataReader, OleDbDataReader) | Cùng namespace | Đọc dữ liệu theo dòng (forward-only), tốc độ nhanh, ít tốn bộ nhớ (dùng cho mô hình Connected). |
| **DataAdapter** (SqlDataAdapter, OleDbDataAdapter) | Cùng namespace | Là cầu nối giữa DataSet và CSDL, giúp lấy dữ liệu vào DataSet và cập nhật ngược lại. |
| **DataSet** | System.Data | Tập hợp dữ liệu trong bộ nhớ (bao gồm nhiều DataTable), làm việc ngắt kết nối với CSDL. |
| **DataTable** | System.Data | Đại diện cho một bảng dữ liệu (giống như bảng trong SQL). |
| **DataView** | System.Data | Cho phép lọc, sắp xếp dữ liệu trong DataTable mà không thay đổi dữ liệu gốc. |
| **Parameter** | System.Data | Dùng để truyền tham số vào câu lệnh SQL hoặc stored procedure. |

Table 3. Bảng các đối tượng chính trong ADO.NET

* + 1. Quy trình truy cập dữ liệu với ADO.NET

Quy trình cơ bản gồm 5 bước:

- Tạo chuỗi kết nối (Connection String) đến CSDL.

- Mở kết nối (Open Connection) bằng đối tượng SqlConnection.

- Khởi tạo và thực thi câu lệnh (Command) thông qua SqlCommand.

- Đọc hoặc lấy dữ liệu bằng SqlDataReader hoặc DataAdapter + DataSet.

- Đóng kết nối (Close Connection) sau khi hoàn tất.

* 1. Giới thiệu về SQL Server
     1. Tổng quan

SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System – RDBMS) do Microsoft phát triển.

Hệ thống này cho phép lưu trữ, quản lý và truy xuất dữ liệu một cách an toàn, hiệu quả và linh hoạt thông qua ngôn ngữ SQL (Structured Query Language).

SQL Server được thiết kế để phục vụ cho cả ứng dụng nhỏ lẫn hệ thống doanh nghiệp quy mô lớn, đồng thời tích hợp chặt chẽ với các công nghệ của Microsoft như .NET, Visual Studio, Azure,...

* + 1. Lịch sử phát triển

Năm 1989: Phiên bản đầu tiên của SQL Server được phát hành, phát triển chung giữa Microsoft, Sybase và Ashton-Tate, chạy trên hệ điều hành OS/2.

Năm 1993: Microsoft tách riêng, phát triển SQL Server cho nền tảng Windows NT.

Các phiên bản quan trọng:

- SQL Server 2000 – hỗ trợ XML và giao diện quản lý cải tiến.

- SQL Server 2005 – ra đời nền tảng .NET Integration, hỗ trợ CLR.

- SQL Server 2012 – thêm chức năng AlwaysOn và Data Tools.

- SQL Server 2016 – cải tiến bảo mật, hỗ trợ JSON, In-Memory OLTP.

- SQL Server 2019 – hỗ trợ Big Data Cluster và tích hợp với Linux.

- SQL Server 2022 – tăng cường khả năng kết nối với Azure và tối ưu hiệu năng truy vấn.

* + 1. Cấu trúc của SQL Server

SQL Server gồm hai lớp chính: Database Engine và SQL Server Agent

- Database Engine: Thành phần cốt lõi, thực hiện lưu trữ, xử lý và bảo mật dữ liệu.

- SQL Server Agent: Quản lý và tự động hóa các tác vụ (backup, bảo trì, lịch trình công việc).

SQL Server Management Studio (SSMS): Giao diện đồ họa cho phép người dùng quản trị, viết truy vấn SQL, và theo dõi hoạt động của hệ thống.

* + 1. Kết luận

SQL Server là một nền tảng quản lý dữ liệu toàn diện, đáng tin cậy và mạnh mẽ, phù hợp cho cả ứng dụng doanh nghiệp, website động, và các hệ thống phân tích dữ liệu lớn.

Nhờ khả năng tích hợp sâu với hệ sinh thái Microsoft và bảo mật cao, SQL Server hiện là một trong những hệ quản trị CSDL phổ biến nhất thế giới.

* 1. Giới thiệu về Bootstrap
     1. Khái niệm về Bootstrap

Bootstrap là một framework mã nguồn mở (open-source framework) được phát triển nhằm hỗ trợ thiết kế giao diện người dùng (UI – User Interface) cho các ứng dụng web một cách nhanh chóng và nhất quán. Framework này được xây dựng chủ yếu dựa trên HTML, CSS và JavaScript, giúp lập trình viên dễ dàng tạo ra các website có giao diện đẹp, tương thích với nhiều kích thước màn hình và thiết bị khác nhau.

Bootstrap ban đầu được phát triển bởi Mark Otto và Jacob Thornton tại Twitter vào năm 2011, với mục tiêu chuẩn hóa các công cụ thiết kế nội bộ. Sau khi được phát hành công khai, Bootstrap nhanh chóng trở thành một trong những framework front-end phổ biến nhất trên thế giới.

* + 1. Mục tiêu và ưu điểm của Bootstrap
       1. Mục tiêu phát triển

Bootstrap được xây dựng với các mục tiêu chính sau:

Tăng tốc quá trình phát triển giao diện web.

Đảm bảo tính thống nhất về thiết kế trên nhiều trình duyệt và thiết bị.

Hỗ trợ thiết kế web responsive (giao diện thích ứng với mọi kích thước màn hình).

Cung cấp một bộ công cụ và thành phần UI phong phú, dễ sử dụng.

* + - 1. Ưu điểm nổi bật

Dễ sử dụng: chỉ cần nắm vững HTML và CSS cơ bản là có thể áp dụng.

Responsive Design: tự động điều chỉnh bố cục phù hợp với các kích thước màn hình (máy tính, máy tính bảng, điện thoại).

Tính nhất quán cao: các thành phần giao diện được định nghĩa sẵn theo cùng một chuẩn thiết kế.

Tích hợp dễ dàng: tương thích với hầu hết các framework front-end và thư viện JavaScript khác.

Cộng đồng lớn: tài liệu phong phú, nhiều ví dụ và hỗ trợ từ cộng đồng lập trình viên toàn cầu.

* + 1. Cấu trúc và thành phần của Bootstrap

Bootstrap bao gồm các thành phần chính sau:

CSS: Cung cấp các lớp định dạng sẵn cho bố cục, kiểu chữ, bảng, biểu mẫu, nút, hình ảnh, v.v.

Grid System: Hệ thống lưới 12 cột giúp xây dựng bố cục web linh hoạt và responsive.

Components: Tập hợp các thành phần giao diện có sẵn như Navbar, Modal, Dropdown, Alert, Carousel, v.v.

Utilities: Các lớp tiện ích (utility classes) hỗ trợ điều chỉnh nhanh các thuộc tính như margin, padding, màu sắc, kích thước.

JavaScript Plugins: Cung cấp các hiệu ứng động như Tooltip, Collapse, Modal, và các thành phần tương tác khác.

* + 1. Ứng dụng của Bootstrap

Bootstrap được sử dụng rộng rãi trong:

- Thiết kế website cá nhân và doanh nghiệp.

- Phát triển giao diện ứng dụng web (web app).

- Tạo prototype nhanh cho dự án.

- Xây dựng hệ thống quản trị (Admin Dashboard).

Nhờ cấu trúc rõ ràng và tài nguyên phong phú, Bootstrap giúp rút ngắn thời gian phát triển mà vẫn đảm bảo tính thẩm mỹ và hiệu năng của sản phẩm.

* 1. Giới thiệu về Entity Framework
     1. Giới thiệu về Entity Framework

Entity Framework (EF) là một ORM (Object-Relational Mapping Framework) do Microsoft phát triển, được sử dụng để hỗ trợ các lập trình viên .NET trong việc tương tác với cơ sở dữ liệu. Thay vì phải viết thủ công các câu lệnh SQL, lập trình viên có thể thao tác trực tiếp với các đối tượng (objects) trong ngôn ngữ lập trình C# hoặc VB.NET.

Entity Framework giúp chuyển đổi dữ liệu giữa cơ sở dữ liệu quan hệ (Relational Database) và các đối tượng hướng đối tượng (Objects) trong ứng dụng, qua đó giảm thiểu sự phức tạp trong quản lý và truy xuất dữ liệu.

* + 1. Mục tiêu và vai trò của Entity Framework

Entity Framework được Microsoft phát triển nhằm:

- Giảm khối lượng mã SQL mà lập trình viên phải viết thủ công.

- Tự động hóa quá trình ánh xạ (mapping) giữa cơ sở dữ liệu và các lớp trong ứng dụng.

- Tăng tính linh hoạt và khả năng mở rộng trong phát triển phần mềm.

- Đảm bảo tính nhất quán dữ liệu thông qua mô hình hướng đối tượng.

Entity Framework đóng vai trò trung gian giữa lớp ứng dụng (Application Layer) và lớp cơ sở dữ liệu (Database Layer), giúp việc truy xuất, thêm, sửa, xóa dữ liệu trở nên dễ dàng và an toàn hơn.

Framework này đặc biệt hữu ích trong các dự án sử dụng kiến trúc 3 lớp (Three-Tier Architecture) hoặc MVC (Model – View – Controller), giúp tăng khả năng tái sử dụng và bảo trì mã nguồn.

1. HIỆN THỰC HÓA NGHIÊN CỨU
   1. Mô tả bài toán

Cửa hàng bán cà phê giải khát muốn triển khai website bán hàng online và quản lý bán hàng. Sau đây là phần mô tả theo các nghiệp vụ hằng ngày ở cửa hàng:

Quản lý khách hàng: Mỗi khách hàng được quản lý bằng các thông tin sau: họ tên, ngày sinh, giới tính, email, mật khẩu, số điện thoại, địa chỉ.

Quản lý sản phẩm: Mỗi sản phẩm được quản lý các thông tin sau: tên sản phẩm, danh mục, giá sản phẩm, hình ảnh, số lượng, mô tả, trạng thái.

Quá trình đặt hàng: Khách hàng khi mua hàng tại website sẽ được gợi ý về những sản phẩm đang sale, sản phẩm mới, và sản phẩm bán chạy của cửa hàng tại trang chủ. Hoặc khách hàng có thể thực hiện tìm kiếm chi tiết tại trang shop. Sau khi tìm được sản phẩm ưng ý khách hàng tiến hành chọn số lượng sản phẩm và nhấn thêm vào giỏ hàng. Khách hàng có thể kiểm tra lại những sản phẩm đã chọn tại trang giỏ hàng. Sau khi kiểm tra xong khách hàng sẽ nhấn thanh toán để tiến hành thanh toán. Website cung cấp cho khách hàng thanh toán bằng phương thức thanh toán khi nhận hàng.

Tất cả khách hàng có thể tra cứu thông tin đơn hàng bằng mã đơn hàng tại trang theo dõi đơn hàng

Đối với những khách hàng đã đăng kí tài khoản tại website thì có thể kiểm tra toàn bộ lịch sử mua hàng tại website của mình ở trang lịch sử mua hàng. Khách hàng có thể thay đổi thông tin cá nhân, thay đổi địa chỉ nhận hàng, thay đổi mật khẩu ở trang cài đặt cá nhân.

Quản lý sẽ quản lý các đơn hàng tại trang quản lý đơn hàng. Dựa trên tình trạng thực tế của đơn hàng Quản lý có thể update tình trạng đơn hàng liên tục để báo cho khách hàng.

* 1. Yêu cầu chức năng

**Yêu cầu trang web:** truy cập hệ thống gồm có hai quyền

**User:** Là những người có nhu cầu mua sắm sản phẩm. Họ sẽ tìm kiếm các sản phẩm cần thiết từ hệ thống và đặt mua các sản phẩm này. Vì thế trang web phải thỏa mãn các chức năng sau:

Hiển thị danh sách các sản phẩm của cửa hàng để khách hàng có thể xem và lựa chọn.

Cung cấp chức năng tìm kiếm sản phẩm, phân trang.

Thêm sản phẩm vào giỏ hàng và thanh toán đơn hàng.

Ngoài ra còn có một số chức năng như: Đăng kí, đăng nhập. Khách hàng có thể thay đổi mật khẩu.

**Admin**: Là người quản trị ứng dụng, có quyền kiểm soát mọi hoạt động của hệ thống. Người này được cấp một tài khoản để đăng nhập vào hệ thống thực hiện các chức năng của mình. Nếu như quá trình đăng nhập thành công thì người quản trị có những chức năng sau:

Chức năng quản lý cập nhật (thêm, xóa, sửa) các sản phẩm trên trang web.

Tiếp nhận và kiểm tra đơn đặt hàng của khách hàng. Hiển thị đơn đặt hàng hay xóa bỏ đơn đặt hàng.

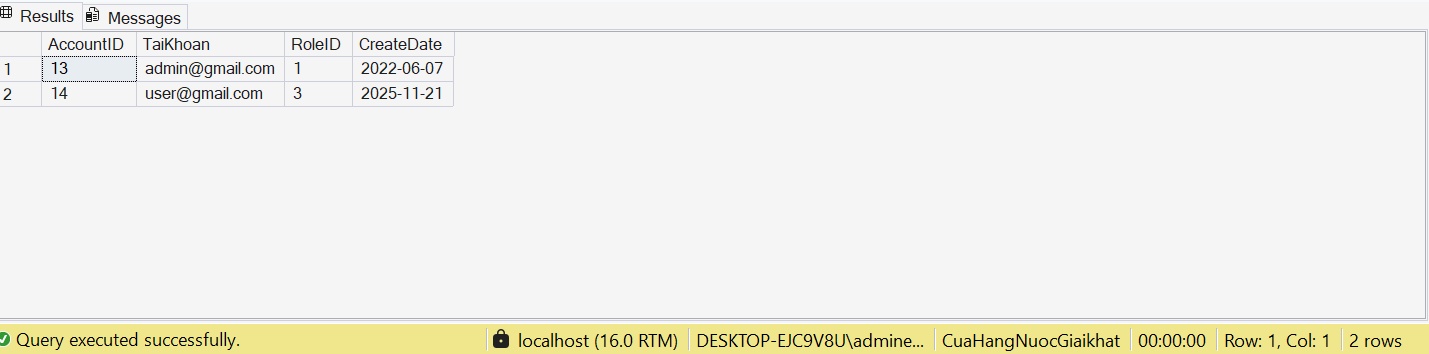
* 1. Cài đặt

Các phần mềm để thực hiện:

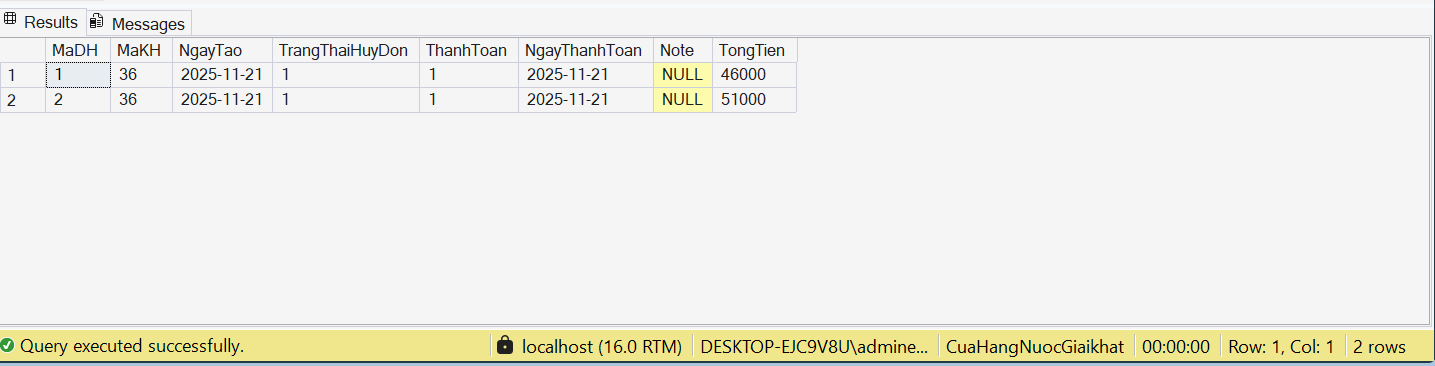
* + Visual Studio 2022 là công cụ Môi trường Phát triển Tích hợp (IDE) hàng đầu của Microsoft, được thiết kế để xây dựng các ứng dụng trên nhiều nền tảng, bao gồm: Windows, Web, di động và Cloud…
  + SQL Server dùng lưu trữ, cập nhật và truy xuất dữ liệu.
  + SQL Server Management Studio cung cấp một môi trường tích hợp (IDE) giao diện đồ họa để truy cập, cấu hình, quản lý và phát triển tất cả các thành phần của SQL Server và các cơ sở dữ liệu có liên quan…
  1. Thử Nghiệm

Dữ liệu thực nghiệm là một phần không thể thiếu trong việc xây dựng trang web. Nó giúp ta trải nghiệm website gần như thực tế. Việc thêm dữ liệu thực nghiệm còn giúp ta phát hiện một số lỗi về hiển thị và xử lý dữ liệu trong việc phát triển website.

***Dưới đây là một số bảng dữ liệu thực nghiệm chính***



Hình 1. Bảng dữ liệu tài khoản người dùng

****

Hình 2. Bảng dữ liệu đơn hàng



Hình 3. Bảng dữ liệu sản phẩm

* 1. Mô hình cơ sở dữ liệu

A screenshot of a computer

AI-generated content may be incorrect.

Hình 4. Mô hình cơ sở dữ liệu

Mô hình cơ sở dữ liệu gồm các bảng chính như sau:

**KhachHang:** chứa thông tin khách hàng (MaKH, TenKH, GioiTinh, AvatarKH, Diachi, Ngaysinh, Phone, Email, Password…).

**SanPham:** MaSP, MaDM, TenSP, AnhSP, GiaSP, TrangThai, SoLuong, MotaSP….

**DanhMucSP:** danh mục sản phẩm (MaDM, TenDM, AnhDM, MoTaDM, TrangThai).

**DonHang:** chứa thông tin đơn hàng (MaDH, MaKH, NgayTao, TrangThai, ThanhToan,…..).

**ChiTietDonHang:** chi tiết đơn hàng (MaCTDH, MaDH, MaSP, TongTien, Ngaygiao, SoLuong).

**TrangThaiDH:** mô tả trạng thái đơn hàng (MaTTDH, MaDH, TrangThai, Mota).

**RoleAccount:** mô tả chức năng của tài khoản (RoleID, RoleName).

**Account:** mô tả tài khoản của hệ thống (AccountID, TaiKhoan, RoleID, CreateDate).

**TinTuc:** mô tả chi tiết về tin tức (MaTT, TenTT, AnhTT, Motangan, Motadai, Tacgia, CreateDate, LoaiTin).

Các bảng có các thuộc tính và mối quan hệ chi tiết bao gồm:

* Bảng **KhachHang**

***Chức năng:***

Lưu trữ toàn bộ thông tin cá nhân của khách hàng sử dụng hệ thống, là thực thể đại diện cho người đặt hàng.

***Thuộc tính:***

MaKH: Khóa chính, định danh duy nhất cho mỗi khách hàng.

TenKH: Họ tên khách hàng.

GioiTinh: Thông tin giới tính.

AvatarKH: Đường dẫn ảnh đại diện.

DiaChi: Địa chỉ liên hệ.

NgaySinh: Ngày sinh khách hàng.

Phone: Số điện thoại.

Email: Địa chỉ email.

Password: Mật khẩu của tài khoản khách hàng.

CreateDate: Ngày tạo tài khoản.

***Vai trò:***

Là bảng gốc phục vụ cho các chức năng đặt hàng, quản lý lịch sử mua hàng và hỗ trợ xử lý thông tin khách trong nghiệp vụ.

* Bảng **DanhMucSP**

***Chức năng:***

Quản lý danh mục sản phẩm, hỗ trợ phân loại các nhóm sản phẩm trong hệ thống

***Thuộc tính:***

MaDM: Khóa chính.

TenDM: Tên danh mục.

AnhDM: Hình ảnh đại diện.

MoTaDM: Mô tả chi tiết danh mục.

TrangThai: Trạng thái hoạt động (kích hoạt/ẩn).

***Vai trò:***

Giúp tổ chức, nhóm các sản phẩm và hỗ trợ tìm kiếm, hiển thị theo danh mục.

* Bảng **SanPham**

***Chức năng:***

Lưu trữ thông tin chi tiết của các sản phẩm được bán trên hệ thống.

***Thuộc tính:***

MaSP: Khóa chính.

MaDM: Khóa ngoại tham chiếu đến DanhMucSP.

TenSP: Tên sản phẩm.

AnhSP: Hình ảnh minh họa sản phẩm.

GiaSP: Giá bán của sản phẩm.

SoLuong: Số lượng tồn kho.

TrangThai: Tình trạng (còn hàng, hết hàng).

MoTaSP: Mô tả chi tiết.

***Vai trò:***

Là dữ liệu trung tâm phục vụ quản lý hàng hóa và được sử dụng trong ChiTietDonHang để xác định mặt hàng khách đã đặt.

* Bảng **DonHang**

***Chức năng:***

Lưu thông tin về các đơn hàng được tạo bởi khách hàng.

***Thuộc tính:***

MaDH: Khóa chính.

MaKH: Khóa ngoại đến KhachHang.

NgayTao: Ngày khởi tạo đơn hàng.

TrangThai: Trạng thái hiện tại của đơn hàng.

ThanhToan: Hình thức hoặc trạng thái thanh toán.

NgayThanhToan: Thời điểm hoàn tất thanh toán.

Note: Ghi chú thêm.

TongTien: Tổng giá trị đơn.

***Vai trò:***

Đóng vai trò trung tâm trong nghiệp vụ bán hàng, là gốc của các chi tiết đơn hàng và lịch sử trạng thái.

* Bảng **ChiTietDonHang**

***Chức năng:***

Mô tả từng sản phẩm có trong một đơn hàng, là bảng trung gian giữa DonHang và SanPham.

***Thuộc tính:***

MaCTDH: Khóa chính.

MaDH: Khóa ngoại tham chiếu đến DonHang.

MaSP: Khóa ngoại tham chiếu đến SanPham.

SoLuong: Số lượng sản phẩm trong đơn.

TongTien: Thành tiền theo số lượng.

NgayGiao: Ngày dự kiến giao hàng.

***Vai trò:***

Giúp xác định cụ thể đơn hàng bao gồm những sản phẩm nào, số lượng ra sao và tổng giá trị từng mục hàng.

* Bảng **TrangThaiDH**

***Chức năng:***

Theo dõi lịch sử thay đổi trạng thái của từng đơn hàng.

***Thuộc tính:***

MaTTDH: Khóa chính.

MaDH: Khóa ngoại liên kết đến DonHang.

TrangThai: Giá trị trạng thái tại thời điểm ghi nhận.

MoTa: Mô tả chi tiết về trạng thái.

***Vai trò:***

Giúp hệ thống lưu lại tiến trình xử lý đơn hàng theo thứ tự thời gian, phục vụ tra cứu và minh bạch quy trình xử lý.

* Bảng **RoleAccount**

***Chức năng:***

Mô tả các vai trò tài khoản nhằm phục vụ phân quyền sử dụng hệ thống.

***Thuộc tính:***

RoleID: Khóa chính.

RoleName: Tên vai trò (Admin, User,…).

***Vai trò:***

Là bảng nền để gán quyền hạn cho từng tài khoản trong bảng Account.

* Bảng **Account**

***Chức năng:***

Quản lý thông tin đăng nhập và quyền hạn của các tài khoản trong hệ thống.

***Thuộc tính:***

AccountID: Khóa chính.

TaiKhoan: Tên đăng nhập.

RoleID: Khóa ngoại đến RoleAccount.

CreateDate: Ngày tạo tài khoản.

***Vai trò:***

Cung cấp cơ chế đăng nhập và kiểm soát phân quyền cho người vận hành và quản trị hệ thống.

* Bảng **TinTuc**

***Chức năng:***

Lưu trữ các bài viết, tin tức quảng cáo hoặc thông báo hệ thống.

***Thuộc tính:***

MaTT: Khóa chính.

TenTT: Tiêu đề tin tức.

AnhTT: Ảnh đại diện bài viết.

MoTaNgan: Tóm tắt nội dung.

MoTaDai: Nội dung chi tiết.

TacGia: Tác giả bài viết.

CreateDate: Ngày đăng.

LoaiTin: Phân loại bài viết.

***Vai trò:***

Phục vụ chức năng truyền thông, giúp hệ thống hiển thị các bài viết và thông tin liên quan.

***Mối quan hệ chi tiết bao gồm:***

Các bảng trong cơ sở dữ liệu được liên kết với nhau thông qua các khóa ngoại nhằm đảm bảo tính toàn vẹn và phản ánh đúng nghiệp vụ. Bảng **KhachHang** liên kết với **DonHang** theo quan hệ một – nhiều, mỗi khách hàng có thể tạo nhiều đơn hàng. Bảng **DanhMucSP** liên kết với **SanPham** theo quan hệ một – nhiều, mỗi danh mục chứa nhiều sản phẩm. **DonHang** và **ChiTietDonHang** cũng có quan hệ một – nhiều, phản ánh rằng một đơn hàng có thể gồm nhiều sản phẩm, trong khi **SanPham** liên kết với **ChiTietDonHang** cho phép mỗi sản phẩm xuất hiện trong nhiều đơn hàng khác nhau. Ngoài ra, bảng **DonHang** có quan hệ một – nhiều với **TrangThaiDH**, giúp lưu lịch sử trạng thái của đơn hàng. Ở cấp quản trị, **RoleAccount** liên kết với **Account** theo quan hệ một – nhiều, phục vụ phân quyền người dùng. Các mối quan hệ này bảo đảm dữ liệu liên kết chặt chẽ, minh bạch và hỗ trợ đầy đủ các nghiệp vụ của hệ thống.

* 1. Lược Đồ USE CASE
     1. Lược Đồ USE CASE – Website

A diagram of a diagram

AI-generated content may be incorrect.

Hình 5. Lược Đồ USE CASE - Website

* + 1. Lược Đồ USE CASE – Người Dùng (USER)

A diagram of a diagram

AI-generated content may be incorrect.

Hình 6. Lược Đồ USE CASE - Người Dùng (USER)

* + 1. Lược Đồ USE CASE – QUẢN TRỊ (Administration)

A diagram of a diagram

AI-generated content may be incorrect.

Hình 7. Lược Đồ USE CASE - QUẢN TRỊ (Administration)

* 1. Kiến Trúc Hệ Thống

Kiến trúc hệ thống áp dụng mô hình phân tầng, dựa trên ASP.NET Core MVC:

Presentation Layer: Giao diện người dùng (Views) sử dụng Bootstrap, HTML5, CSS3, JavaScript.

Business Logic Layer: Các Controller xử lý yêu cầu, coordination logic nghiệp vụ.

Data Access Layer: Sử dụng Entity Framework Core tương tác với cơ sở dữ liệu.

Database: SQL Server hoặc SQLite.

Authentication: ASP.NET Identity để quản lý người dùng và bảo mật.

Thanh toán: Dự kiến tích hợp Payment Gateway (Stripe, VNPay).

A diagram of a software process

AI-generated content may be incorrect.

Hình 8. Mô Hình Kiến Trúc Tổng Quan Hệ Thống

* 1. Các bước triển khai hệ thống được thực hiện theo trình tự sau:
     1. Phân tích yêu cầu và xây dựng mô hình hệ thống

Ở giai đoạn đầu, nhóm phát triển tiến hành thu thập và tổng hợp nhu cầu từ ba nhóm đối tượng sử dụng chính: khách hàng, thợ sửa chữa và quản trị viên. Các yêu cầu chức năng và phi chức năng được phân loại thành các nhóm như: đăng ký, đăng nhập, đặt lịch sửa chữa, quản lý đơn hàng, xử lý thanh toán, đánh giá chất lượng dịch vụ và báo cáo thống kê. Kết quả của bước này là tài liệu đặc tả yêu cầu chi tiết, đóng vai trò định hướng cho quá trình thiết kế và hiện thực hệ thống.

* + 1. Thiết kế kỹ thuật và mô hình dữ liệu

Dựa trên nền tảng kiến trúc ASP.NET Core MVC, hệ thống được tổ chức thành ba tầng chính: Presentation (View, Controller), Business Logic (Service) và Data Access (Repository). Song song với đó, mô hình dữ liệu được xây dựng với các bảng trọng tâm như Users, Services, Orders, Payments và Reviews. Các quan hệ khóa chính – khóa ngoại được thiết lập đầy đủ nhằm bảo đảm tính nhất quán dữ liệu, đồng thời tối ưu khả năng truy xuất trong quá trình vận hành.

* + 1. Phát triển giao diện và chức năng ứng dụng

Giao diện được xây dựng theo hướng hiện đại, trực quan và đáp ứng đa nền tảng thông qua việc sử dụng Bootstrap, HTML5, CSS3 và JavaScript. Các tính năng được lập trình bám sát tài liệu đặc tả, bao gồm: đăng ký/đăng nhập người dùng, tìm kiếm và lựa chọn dịch vụ, đặt lịch hẹn, thực hiện thanh toán, quản lý đơn hàng, đánh giá dịch vụ và các chức năng quản trị. Trong quá trình phát triển, từng module đều được kiểm thử độc lập nhằm bảo đảm sự ổn định trước khi tích hợp với toàn hệ thống.

* + 1. Kiểm thử, xử lý lỗi và tối ưu hóa hệ thống

Hệ thống trải qua nhiều vòng kiểm thử, bao gồm kiểm thử đơn vị, kiểm thử tích hợp và kiểm thử bảo mật, nhằm phát hiện sớm các sai sót và khắc phục kịp thời. Song song với hoạt động kiểm thử, nhóm phát triển cũng tiến hành tối ưu mã nguồn, cải thiện cấu trúc dữ liệu và xem xét hiệu năng nhằm nâng cao tốc độ xử lý và trải nghiệm người dùng cuối.

* + 1. Triển khai và chạy thử trên môi trường thực tế

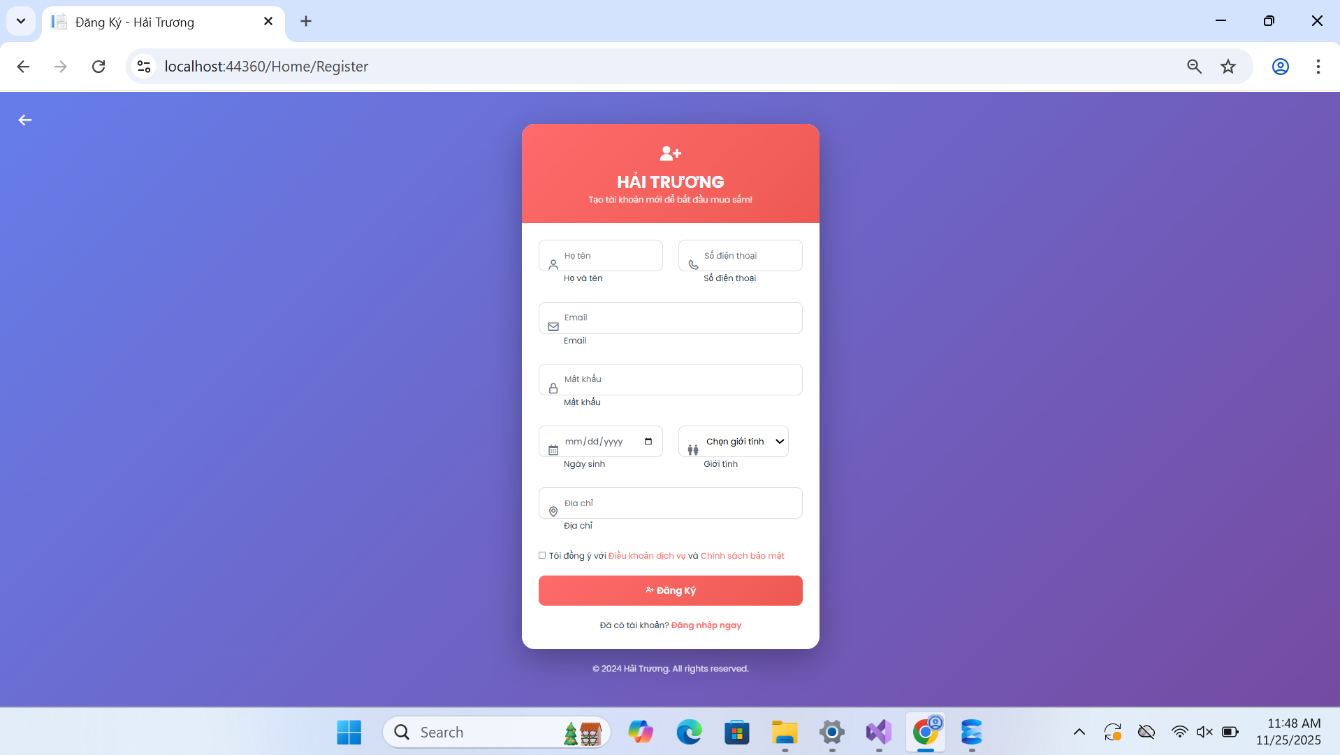
Sau khi hoàn thiện các chức năng cốt lõi, hệ thống được triển khai lên môi trường máy chủ phù hợp và cấu hình HTTPS để bảo vệ dữ liệu. Giai đoạn chạy thử được tiến hành nhằm đánh giá khả năng chịu tải, mức độ ổn định và tính an toàn bảo mật của ứng dụng. Các phản hồi thu thập được trong giai đoạn này được sử dụng để điều chỉnh, hoàn thiện hệ thống trước khi đưa vào vận hành chính thức.

1. KẾT QUẢ NGHIÊN CỨU
   * 1. Giao diện đăng nhập và đăng ký

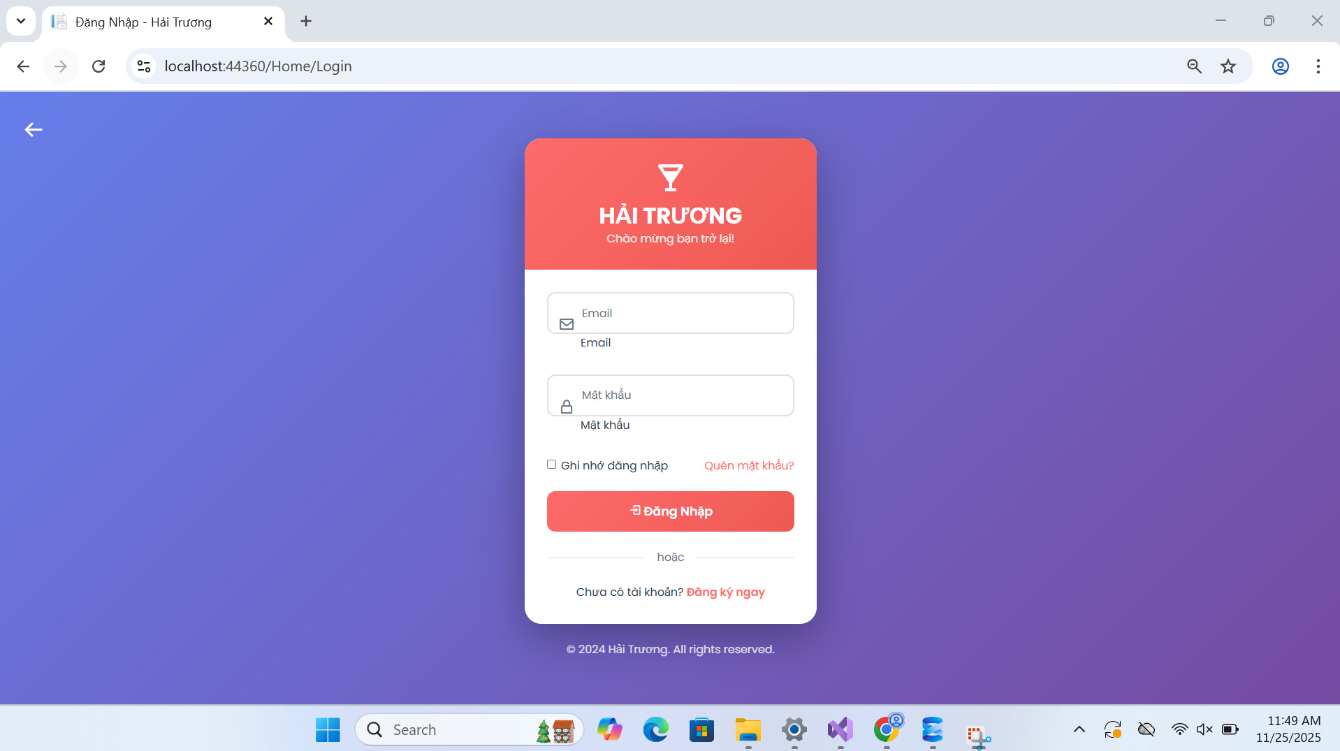
Giao diện đăng nhập là giao diện để tài khoản người dùng đăng nhập vào hệ thống. Các tài khoản sẽ được phân quyền khác nhau trên Website. Khách hàng muốn thực hiện chức năng mua sắm giỏ hàng bắt buộc phải đăng ký tài khoản trên hệ thống thì mới thực hiện được chức năng này.

Khách hàng: Giao diện trang chủ dùng để trình bày các sản phẩm có trên website. Khách hàng có thể xem các sản phẩm nổi bật trên trang chủ.

Trang quản trị: Bao gồm các chức năng quản lý của Admin về tài khoản người dùng, sản phẩm, đơn hàng.

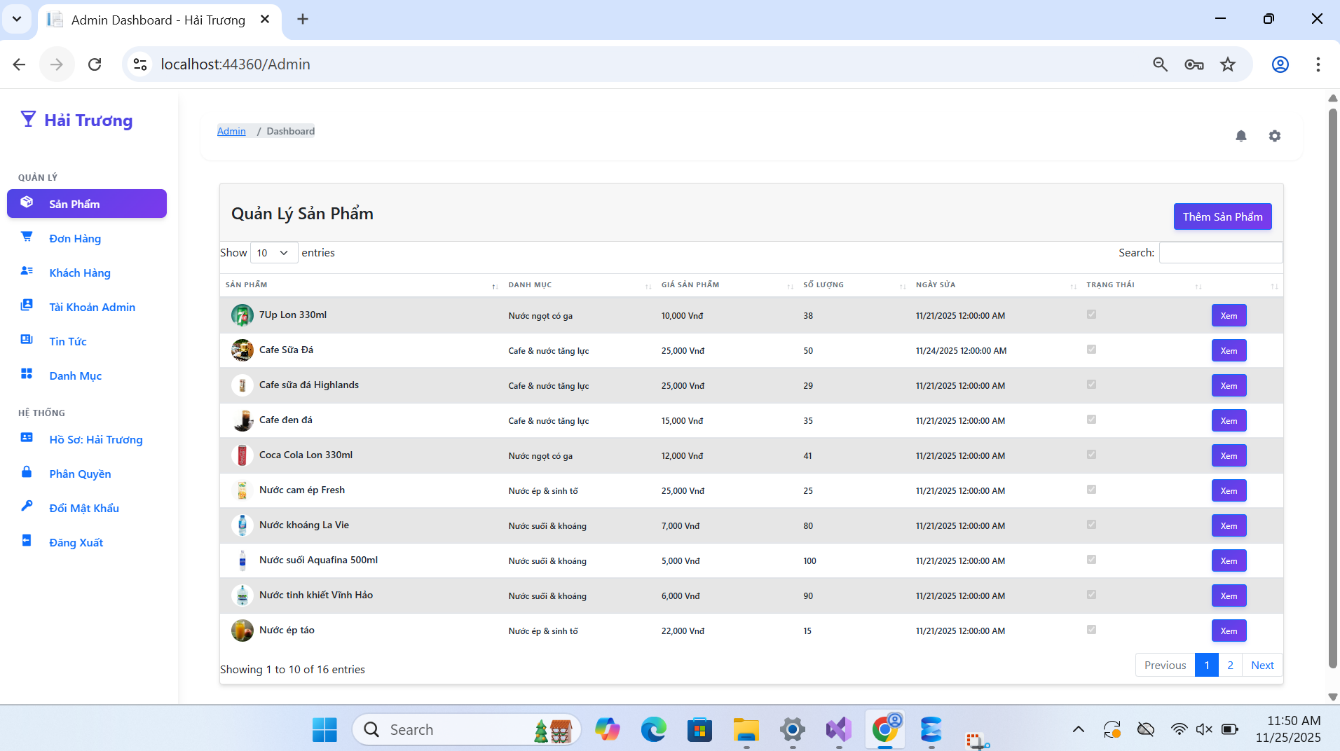


Hình 9. Giao Diện Đăng Ký Tài Khoản



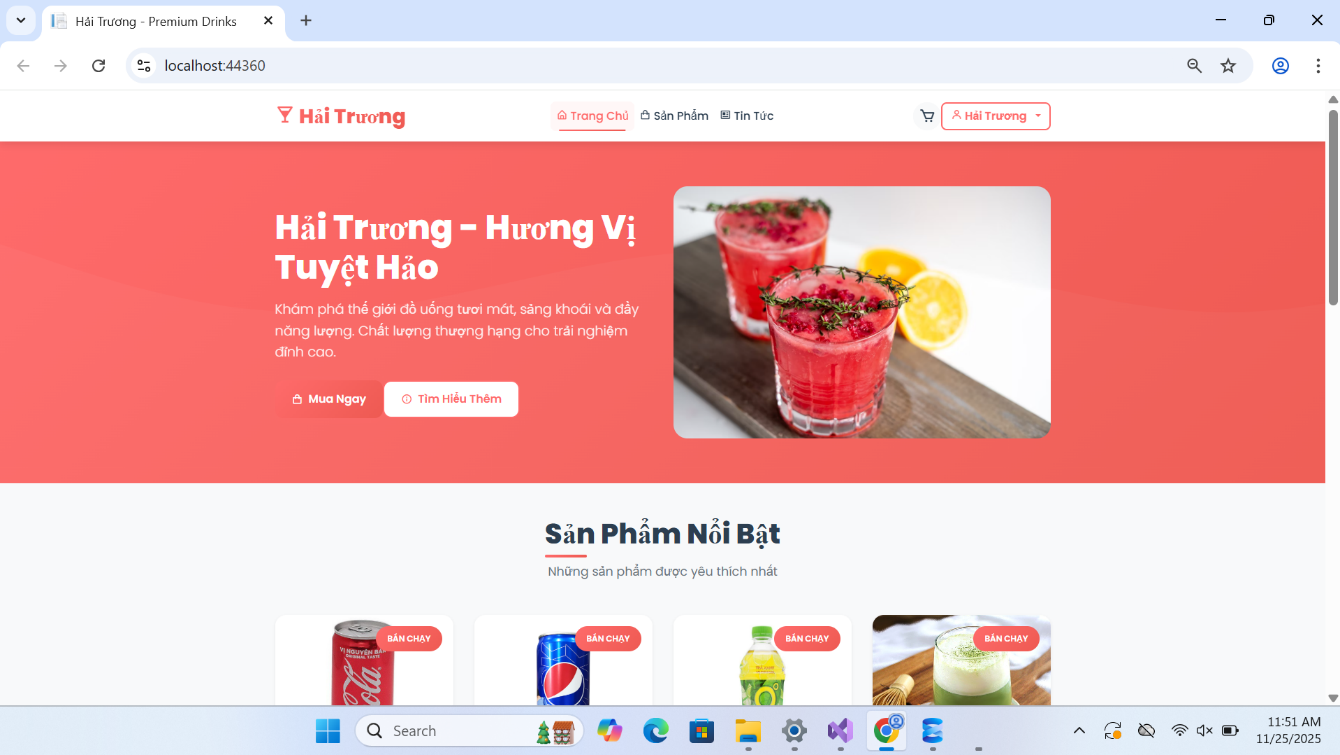
Hình 10. Giao Diện Đăng Nhập

* + 1. Giao diện Trang Quản Trị



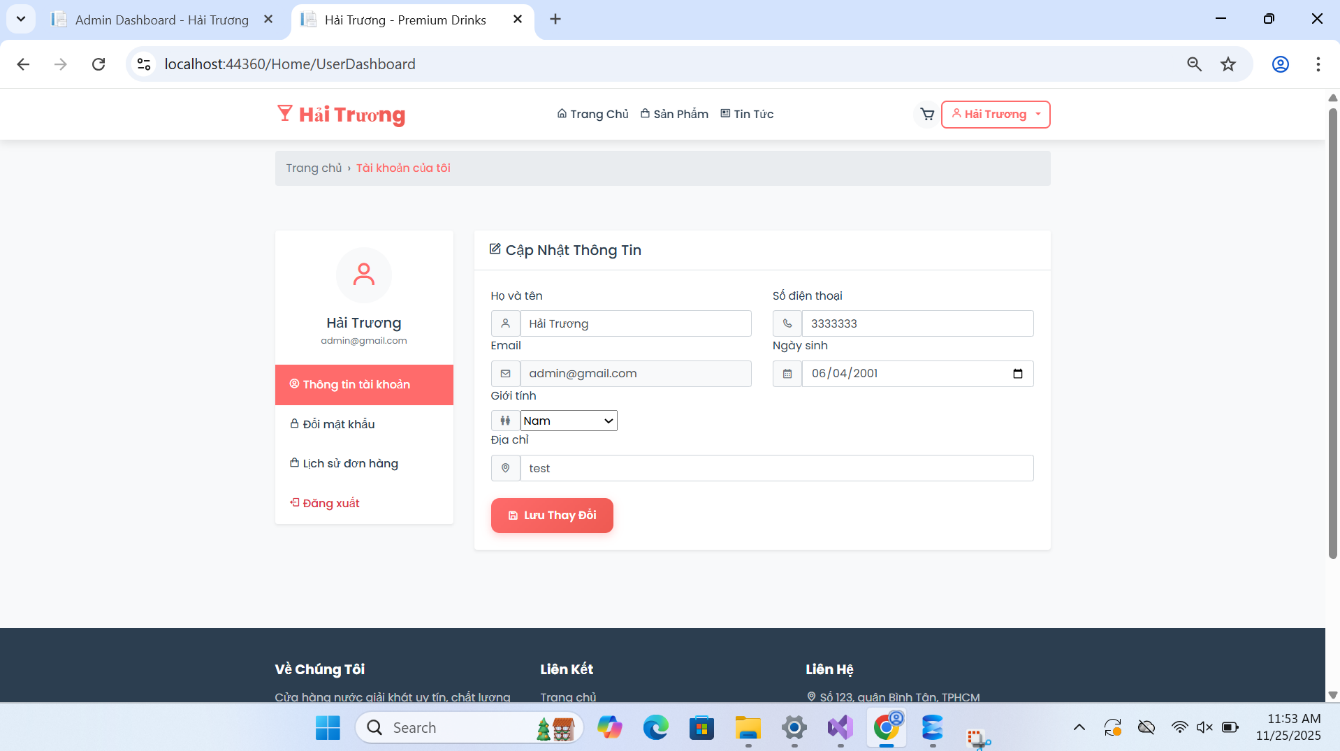
Hình 11. Giao Diện Trang Quản Trị

* + 1. Giao diện Trang Chủ



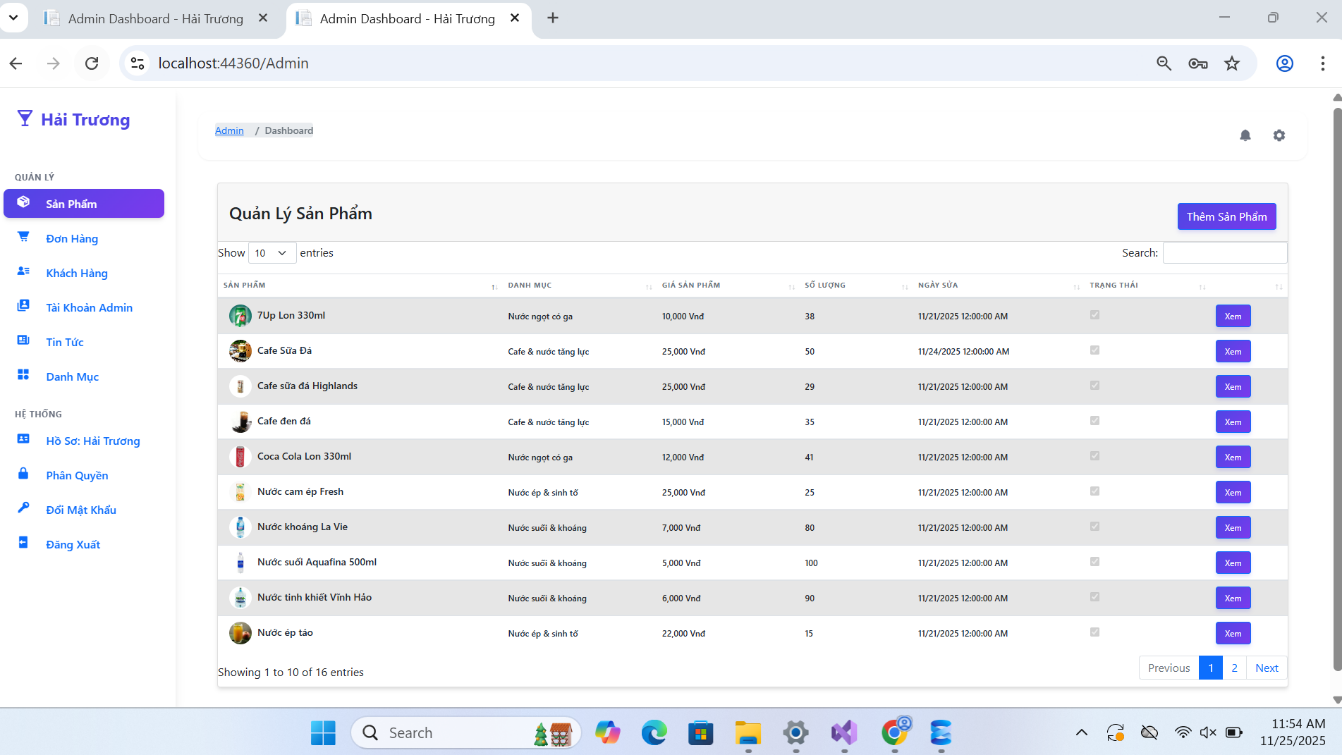
Hình 12. Giao Diện Trang Chủ

* + 1. Giao diện Trang Chỉnh Sửa Thông Tin Người Dùng



Hình 13. Giao Diện Trang Quản Lý Thông Tin Người Dùng

* + 1. Giao diện Trang Quản Lý Thông Tin Sản Phẩm



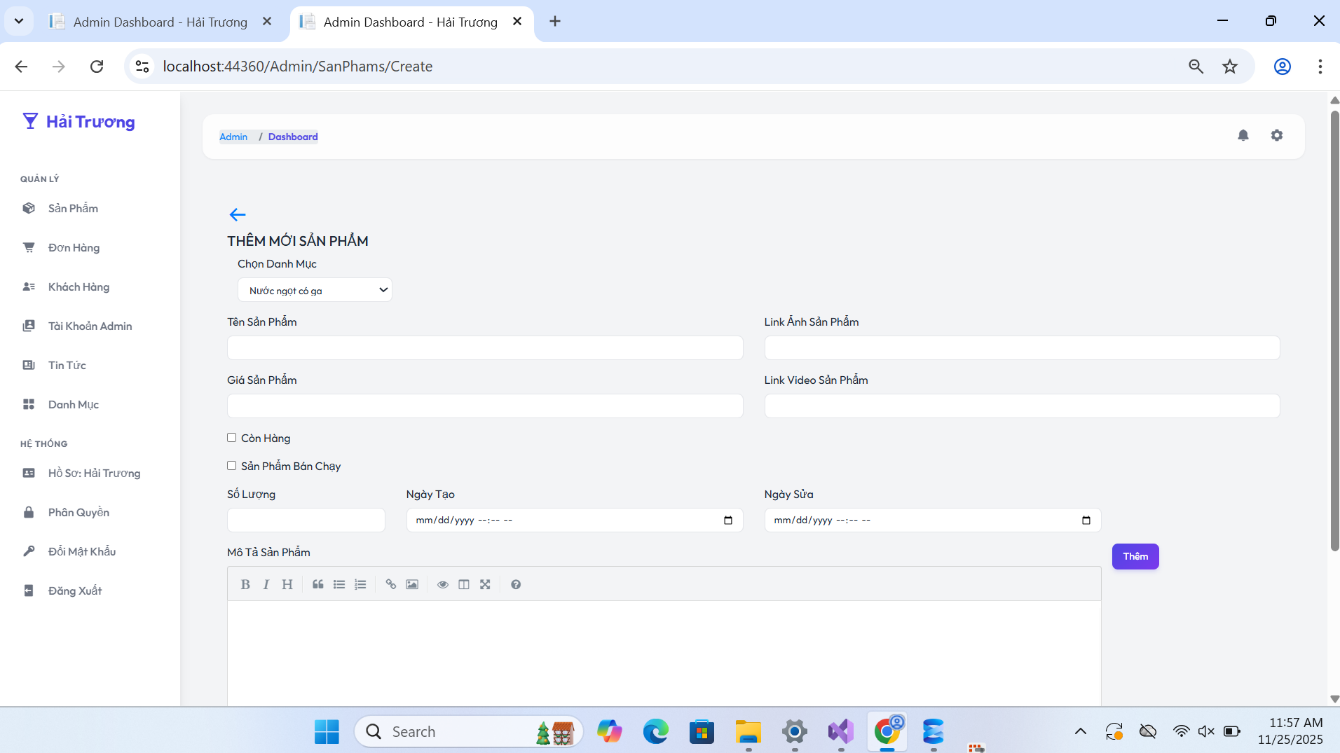
Hình 14. Giao diện Trang Quản Lý Thông Tin Sản Phẩm

* + 1. Giao diện Trang Chỉnh Thông Tin Sản Phẩm



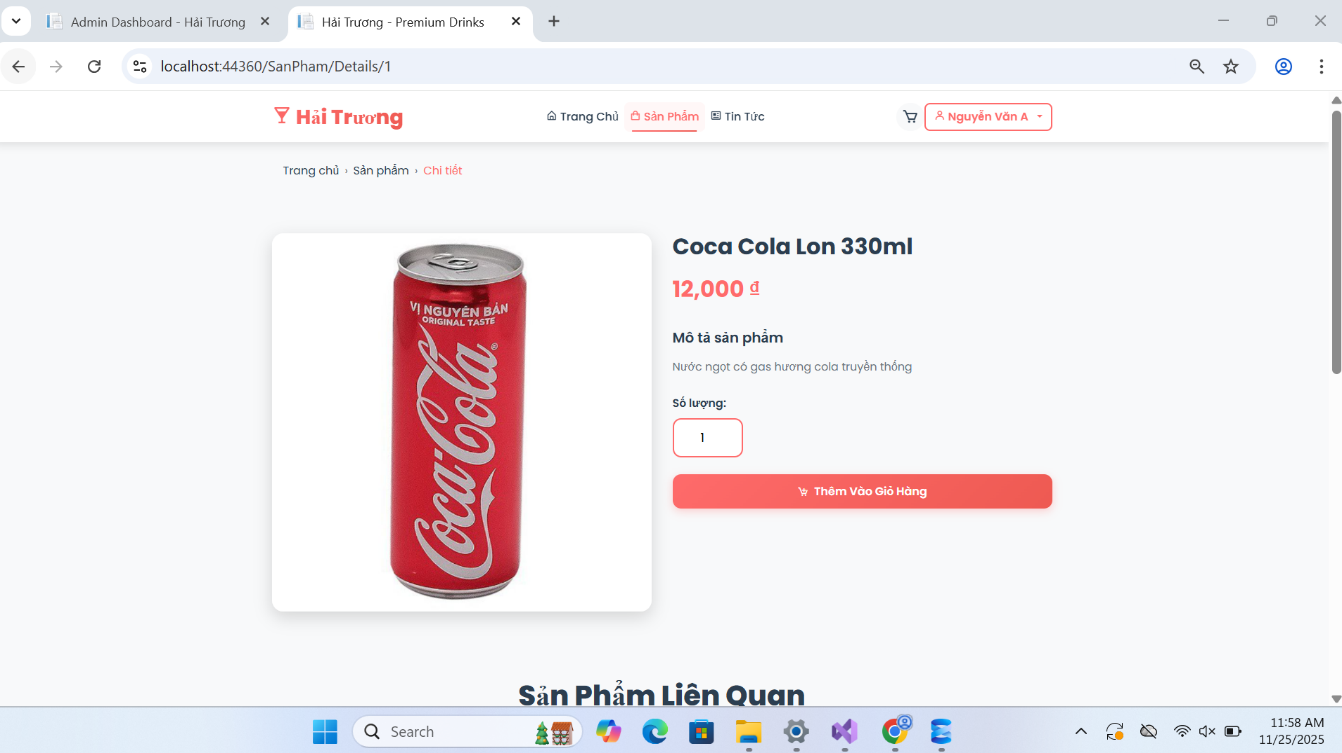
Hình 15. Giao diện Trang Chỉnh Sửa Thông Tin Sản Phẩm

* + 1. Giao diện Trang Thêm Mới Sản Phẩm



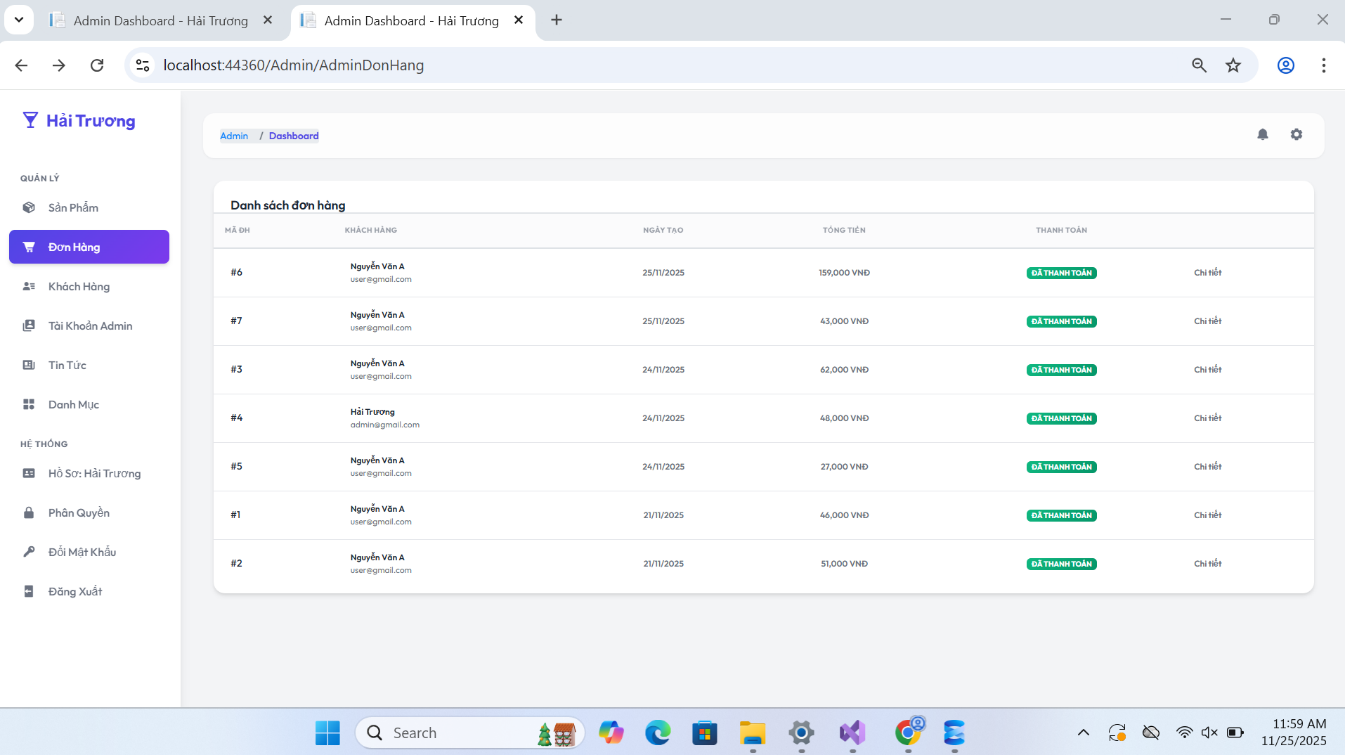
Hình 16. Giao diện Trang Thêm Mới Sản Phẩm

* + 1. Giao diện Trang Chi Tiết Sản Phẩm

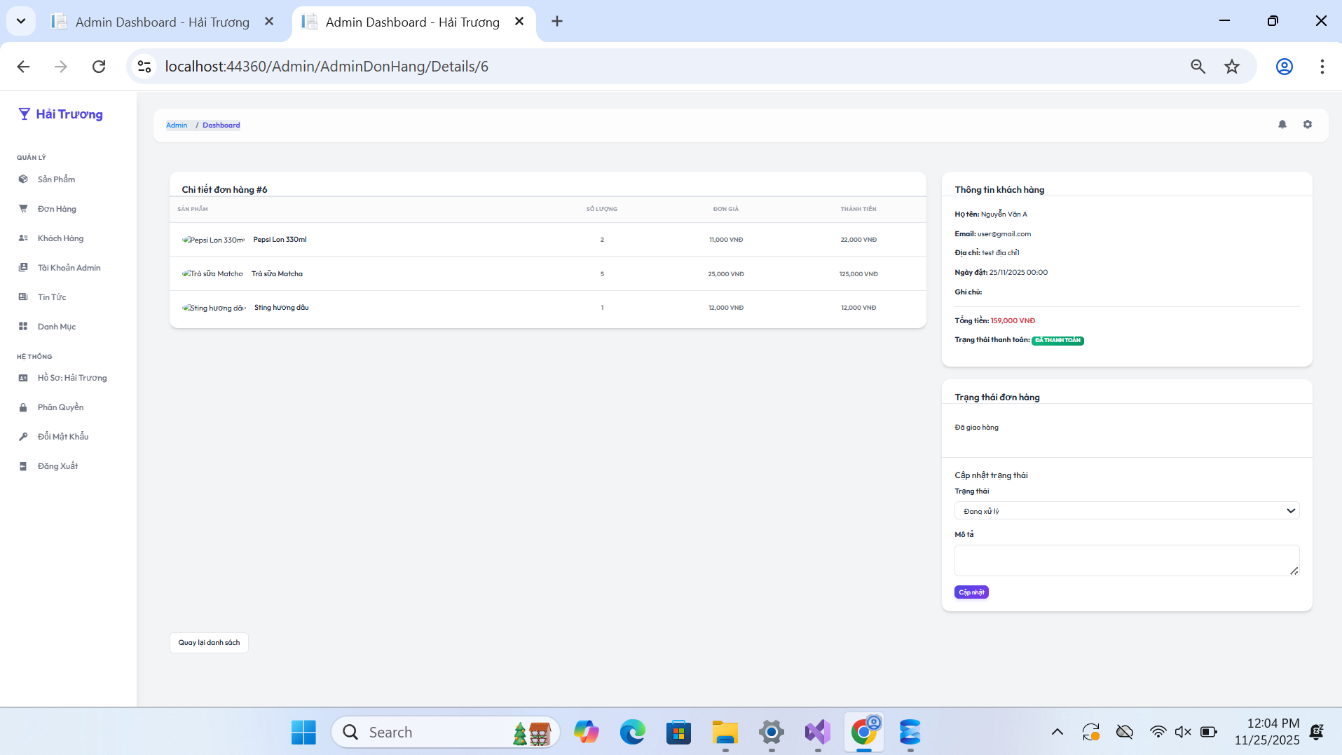


Hình 17. Giao diện Trang Chi Tiết Sản Phẩm

* + 1. Giao diện Trang Quản Lý Đơn Hàng

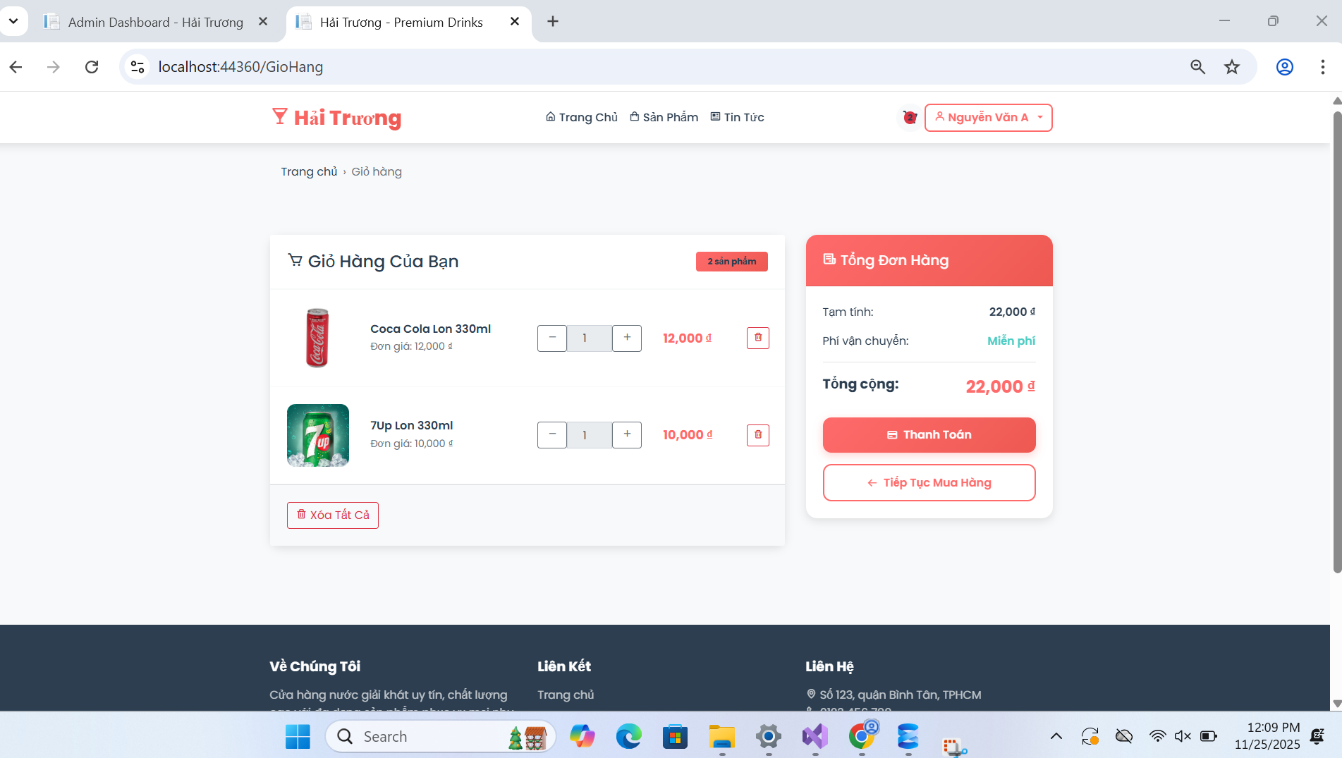


Hình 18. Giao diện Trang Quản Lý Đơn Hàng



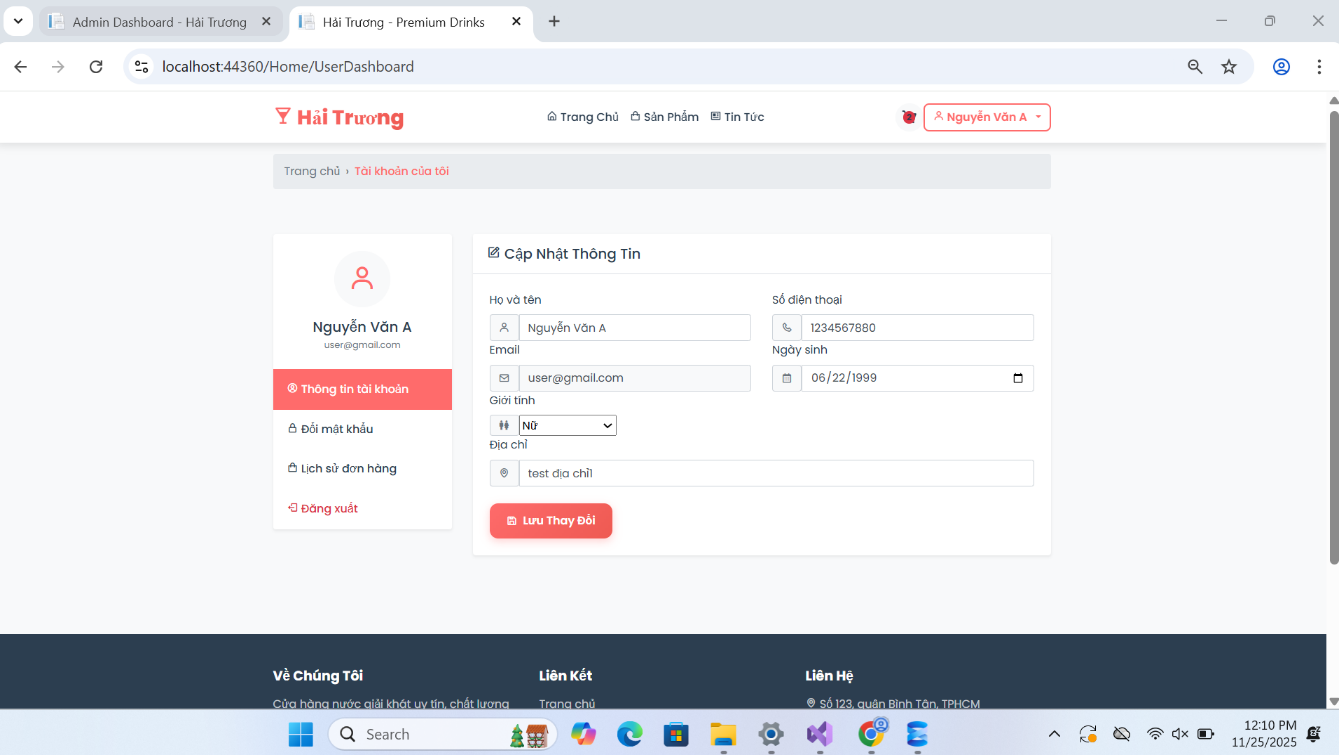
Hình 19. Giao diện Trang Quản Lý Đơn Hàng Chi Tiết

* + 1. Giao diện Trang Giỏ Hàng



Hình 20. Giao diện Trang Giỏ Hàng

* + 1. Giao diện Trang Profile Người Dùng



Hình 21. Giao diện Trang Profile Người Dùng

1. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN
   1. Các kết quả đạt được

* Hoàn thành việc thiết kế Cơ sở dữ liệu về bán sản phẩm Cà phê giải khát.
* Xây dựng thành công website bán sản phẩm cà phê giải khát.
* Hoàn thành các chức năng cơ bản: thêm, xóa, sửa dữ liệu.
* Hoàn thành chức năng tìm kiếm tra cứu thông tin.
* Ràng buộc một số dữ liệu do người dùng nhập.
  1. Hạn chế, hướng phát triển
     1. Hạn chế:
* Về mặt thiết kế cơ sở dữ liệu chưa được hoàn chỉnh, đầy đủ.
* Chưa chuyên nghiệp như các website thương mại điện tử có cùng nội dung.
* Chưa có tính bảo mật cho trang web.
  + 1. Hướng phát triển:
* Tăng tính bảo mật cho trang web.
* Hoàn thiện các tính năng nâng cao: shipper giao hàng, thanh toán bằng nhiều phương thức khác, đơn đặt hàng.
* Khách hàng có thể tự hủy đơn hàng.
* Thiết kế thêm tính năng tự động gửi mail đến email khách hàng khi khách hàng thanh toán xong đơn hàng.
* Tính năng nhận xét sản phẩm.
* Sử dụng công nghệ Web API để tiếp tục phát triển website. Web API giúp người dùng tương tác hiệu quả với website.

# DANH MỤC TÀI LIỆU THAM KHẢO

[1] Get started with ASP.NET Core MVC : <https://learn.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/start-mvc?view=aspnetcore-8.0&tabs=visual-studio>

[2] Get started with Bootstrap :<https://getbootstrap.com/docs/5.3/getting-started/introduction/>

[3] TutorialsTeacher – ASP.NET MVC: <https://www.tutorialsteacher.com/mvc>

[4] Tổng quan công nghệ ASP.NET, từ Web Forms đến MVC và Core: <https://dotnet.microsoft.com/apps/aspnet>

[5] Youtube channel: <https://www.youtube.com/watch?v=HUKMLsVWMP4>