**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**

Tôi xin trân trọng cảm ơn quý thầy cô Trường Đại học Trà Vinh đã tạo điều kiện thuận lợi cho tôi trong suốt quá trình học tập và thực hiện báo cáo. Tôi đặc biệt biết ơn thầy cô Khoa Kỹ thuật và Công nghệ đã tận tâm truyền đạt kiến thức và kinh nghiệm quý báu, góp phần giúp tôi phát triển năng lực chuyên môn.

Đặc biệt, tôi xin gửi lời cảm ơn sâu sắc đến thầy Đoàn Phước Miền vì sự hướng dẫn tận tình và những định hướng chuyên môn quan trọng, giúp tôi hoàn thành tốt đồ án cơ sở ngành.

Tôi cũng xin cảm ơn các anh, chị và bạn bè đã hỗ trợ tôi trong quá trình hoàn thành báo cáo.

Mặc dù đã nỗ lực, bài báo cáo không tránh khỏi thiếu sót. Tôi kính mong nhận được ý kiến đóng góp của quý thầy cô để hoàn thiện hơn trong thời gian tới.

Xin trân trọng cảm ơn!

Sinh viên ký và ghi rõ họ và tên

**MỤC LỤC**

[MỞ ĐẦU ix](#_Toc215216095)

[CHƯƠNG 1: TỔNG QUAN 1](#_Toc215216096)

[1.1 Lý do chọn đề tài 1](#_Toc215216097)

[1.2 Mục tiêu nghiên cứu 1](#_Toc215216098)

[1.3 Phương pháp nghiên cứu 2](#_Toc215216099)

[1.4 Đối tượng nghiên cứu 2](#_Toc215216100)

[1.5 Phạm vi nghiên cứu 3](#_Toc215216101)

[1.6 Ý nghĩa khoa học và thực tiễn của đề tài 4](#_Toc215216102)

[1.6.1 Ý nghĩa khoa học 4](#_Toc215216103)

[1.6.2 Ý nghĩa thực tiễn 5](#_Toc215216104)

[1.7 Tính cấp thiết của đề tài 5](#_Toc215216105)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 6](#_Toc215216106)

[2.1 Tổng quan về mô hình MVC 6](#_Toc215216107)

[2.1.1 Khái niệm 6](#_Toc215216108)

[2.1.2 Luồng xử lý trong MVC 6](#_Toc215216109)

[2.1.3 Vai trò MVC trong phát triển website 7](#_Toc215216110)

[2.2 Công nghệ ASP.NET MVC 7](#_Toc215216111)

[2.2.1 Tổng quan 7](#_Toc215216112)

[2.2.2 Lịch sử phát triển 7](#_Toc215216113)

[2.2.3 Tính năng của ASP.NET MVC 8](#_Toc215216114)

[2.3 Giới thiệu ngôn ngữ lập trình C# (C Sharp) 8](#_Toc215216115)

[2.3.1 Tổng quan 8](#_Toc215216116)

[2.3.2 Đặc điểm nổi bật 9](#_Toc215216117)

[2.3.3 Ứng dụng trong phát triển website 9](#_Toc215216118)

[2.4 Giới thiệu các Controls của ASP.NET 9](#_Toc215216119)

[2.4.1 Tổng quan 9](#_Toc215216120)

[2.4.2 HTML Controls 9](#_Toc215216121)

[2.4.3 Web Server Controls 10](#_Toc215216122)

[2.4.4 Validation Controls 10](#_Toc215216123)

[2.5 Giới thiệu ADO.NET và các đối tượng truy cập CSDL 11](#_Toc215216124)

[2.5.1 Khái niệm về ADO.NET 11](#_Toc215216125)

[2.5.2 Mô hình hoạt động của ADO.NET 11](#_Toc215216126)

[2.5.3 Các đối tượng chính trong ADO.NET 11](#_Toc215216127)

[2.5.4 Quy trình truy cập dữ liệu với ADO.NET 12](#_Toc215216128)

[2.6 Giới thiệu về SQL Server 12](#_Toc215216129)

[2.6.1 Tổng quan 12](#_Toc215216130)

[2.6.2 Lịch sử phát triển 13](#_Toc215216131)

[2.6.3 Cấu trúc của SQL Server 13](#_Toc215216132)

[2.6.4 Kết luận 13](#_Toc215216133)

[2.7 Giới thiệu về Bootstrap 13](#_Toc215216134)

[2.7.1 Khái niệm về Bootstrap 13](#_Toc215216135)

[2.7.2 Mục tiêu và ưu điểm của Bootstrap 14](#_Toc215216136)

[2.7.2.1. Mục tiêu phát triển 14](#_Toc215216137)

[2.7.2.2. Ưu điểm nổi bật 14](#_Toc215216138)

[2.7.3 Cấu trúc và thành phần của Bootstrap 14](#_Toc215216139)

[2.7.4 Ứng dụng của Bootstrap 15](#_Toc215216140)

[2.8 Giới thiệu về Entity Framework 15](#_Toc215216141)

[2.8.1 Giới thiệu về Entity Framework 15](#_Toc215216142)

[2.8.2 Mục tiêu và vai trò của Entity Framework 16](#_Toc215216143)

[CHƯƠNG 3: CÀI ĐẶT VÀ THỬ NGHIỆM 17](#_Toc215216144)

[3.1 Cài đặt 17](#_Toc215216145)

[3.2 Thử nghiệm 17](#_Toc215216146)

[3.3 Ứng dụng minh họa 18](#_Toc215216147)

[3.3.1 Mô tả bài toán 18](#_Toc215216148)

[3.3.2 Thiết kế hệ thống 20](#_Toc215216149)

[3.3.3 Yêu cầu chức năng 20](#_Toc215216150)

[3.3.3.1 Chức năng nghiệp vụ 20](#_Toc215216151)

[3.3.3.2 Chức năng hệ thống 21](#_Toc215216152)

[CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU 23](#_Toc215216153)

[4.1 Các công nghệ sử dụng trong đề tài 23](#_Toc215216154)

[4.1.1 ASP.NET MVC 23](#_Toc215216155)

[4.1.1.1 Tạo dự án với ASP.NET MVC 23](#_Toc215216156)

[4.1.1.2 Cấu trúc cây thư mục của ASP.NET MVC 25](#_Toc215216157)

[4.1.2 Entity Framework 25](#_Toc215216158)

[4.1.2.1 Cài đặt Entity Framework cho dự án 25](#_Toc215216159)

[4.2 Thiết kế giao diện 26](#_Toc215216160)

[4.2.1 Giao diện đăng nhập và đăng ký 26](#_Toc215216161)

[4.2.2 Giao diện trang quản trị 27](#_Toc215216162)

[4.2.3 Giao diện trang chủ 28](#_Toc215216163)

[4.2.4 Giao diện trang sản phẩm 28](#_Toc215216164)

[4.2.5 Giao diện trang chi tiết sản phẩm 29](#_Toc215216165)

[4.2.6 Giao diện giỏ hàng 29](#_Toc215216166)

[4.2.7 Giao diện thông tin người dùng 30](#_Toc215216167)

[4.2.8 Giao diện lịch sử mua hàng 30](#_Toc215216168)

[4.2.9 Giao diện danh mục sản phẩm 31](#_Toc215216169)

[4.2.10 Giao diện lọc sản phẩm 31](#_Toc215216170)

[4.2.11 Giao diện trang quản lý đơn hàng 32](#_Toc215216171)

[4.2.12 Giao diện quản lý khách hàng 32](#_Toc215216172)

[4.2.13 Giao diện sửa thông tin khách hàng 33](#_Toc215216173)

[4.2.14 Giao diện thêm mới danh mục 33](#_Toc215216174)

[4.2.15 Giao diện thêm mới sản phẩm 34](#_Toc215216175)

[CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 35](#_Toc215216176)

[5.1 Các kết quả đạt được 35](#_Toc215216177)

[5.2 Hạn chế, hướng phát triển 35](#_Toc215216178)

[5.2.1 Hạn chế: 35](#_Toc215216179)

[5.2.2 Hướng phát triển: 35](#_Toc215216180)

[DANH MỤC TÀI LIỆU THAM KHẢO 36](#_Toc215216181)

**DANH MỤC HÌNH ẢNH**

[Hình 1 Bảng dữ liệu tài khoản người dùng 17](#_Toc215294372)

[Hình 2 Bảng dữ liệu sản phẩm linh kiện 18](#_Toc215294373)

[Hình 3 Mô hình ERD 20](#_Toc215294374)

[Hình 4 Sơ đồ User Case tổng quan 21](#_Toc215294375)

[Hình 5 Bước 1 - Tạo Project mới trên Visual Studio 23](#_Toc215294376)

[Hình 6 Bước 2 – Chọn ASP.NET Core Web App (Model-View-Controller) 23](#_Toc215294377)

[Hình 7 Bước 3 – Đặt tên cho Project và đường dẫn lưu 24](#_Toc215294378)

[Hình 8 Bước 4 – nhấn chọn Create 24](#_Toc215294379)

[Hình 9 Giao diện làm việc và cấu trúc cây thư mục của ASP.NET MVC 25](#_Toc215294380)

[Hình 10 Package Manager Console sử dụng dòng lệnh 25](#_Toc215294381)

[Hình 11 Giao diện đăng ký tài khoản 26](#_Toc215294382)

[Hình 12 Giao diện đăng nhập tài khoản 27](#_Toc215294383)

[Hình 13 Giao diện trang quản trị 27](#_Toc215294384)

[Hình 14 Giao diện trang chủ 28](#_Toc215294385)

[Hình 15 Giao diện trang sản phẩm 28](#_Toc215294386)

[Hình 16 Giao diện trang chi tiết sản phẩm 29](#_Toc215294387)

[Hình 17 Giao diện giỏ hàng 29](#_Toc215294388)

[Hình 18 Giao diện thông tin người dùng 30](#_Toc215294389)

[Hình 19 Giao diện lịch sử mua hàng 30](#_Toc215294390)

[Hình 20 Giao diện danh mục sản phẩm 31](#_Toc215294391)

[Hình 21 Giao diện lọc sản phẩm 31](#_Toc215294392)

[Hình 22 Giao diện trang quản lý đơn hàng 32](#_Toc215294393)

[Hình 23 Giao diện quản lý khách hàng 32](#_Toc215294394)

[Hình 24 Giao diện sửa thông tin khách hàng 33](#_Toc215294395)

[Hình 25 Giao diện thêm mới danh mục 33](#_Toc215294396)

[Hình 26 Giao diện thêm mới sản phẩm 34](#_Toc215294397)

**DANH MỤC CỤM TỪ VIẾT TẮT**

|  |  |
| --- | --- |
| **Từ viết tắt** | **Diễn giải** |
| MVC | Model-View-Controller |
| SQL | Structured Query Language |
| HTML | HyperText Markup Language |
| UX/UI | User Experience/ User Interface |
| C# | C Sharp |
| CSDL | Cơ sở dữ liệu |
| ADO.NET | ActiveX Data Objects for .NET |
| SSMS | SQL Server Management Studio |
| EF | Entity Framework |
| ORM | Object-Relational Mapping Framework |
| ERD | Entity-Relationship Diagram |
| DFD | Data Flow Diagram |
| GUI | Graphical User Interface |
| RDBMS | Relational Database Management System |

# MỞ ĐẦU

Trong bối cảnh cuộc cách mạng công nghiệp 4.0 và xu hướng chuyển đổi số đang diễn ra mạnh mẽ, công nghệ thông tin ngày càng đóng vai trò then chốt trong mọi lĩnh vực, đặc biệt là thương mại điện tử. Việc ứng dụng các giải pháp công nghệ vào hoạt động kinh doanh không chỉ giúp doanh nghiệp nâng cao hiệu quả quản lý, giảm chi phí vận hành mà còn mở rộng khả năng tiếp cận khách hàng và tăng tính cạnh tranh trên thị trường.

Linh kiện điện tử là một lĩnh vực có nhu cầu tiêu thụ lớn, sản phẩm đa dạng, nhiều thông số kỹ thuật và tốc độ đổi mới nhanh. Việc xây dựng một website bán linh kiện điện tử chuyên nghiệp sẽ giúp người dùng dễ dàng tra cứu thông tin, so sánh, lựa chọn sản phẩm phù hợp, đồng thời hỗ trợ doanh nghiệp quản lý danh mục sản phẩm, đơn hàng và khách hàng một cách hiệu quả.

Xuất phát từ nhu cầu thực tiễn và mong muốn áp dụng kiến thức đã học về lập trình web, cơ sở dữ liệu, giao diện người dùng và triển khai ứng dụng thực tế, đề tài “Xây dựng website bán linh kiện điện tử” được thực hiện. Đề tài nhằm mục tiêu nghiên cứu, thiết kế và xây dựng một hệ thống thương mại điện tử hoàn chỉnh, vừa đảm bảo tính trực quan, tiện dụng cho người dùng, vừa đáp ứng yêu cầu quản trị và vận hành hiệu quả, góp phần nâng cao năng lực ứng dụng thực tế của người thực hiện và phục vụ mục đích học thuật.

Báo cáo chuyên đề là kết quả của quá trình học tập, nghiên cứu và thực hành nghiêm túc. Mặc dù đã cố gắng hoàn thiện, nhưng chắc chắn không thể tránh khỏi những thiếu sót. Kính mong quý thầy cô góp ý để em có thể rút kinh nghiệm và nâng cao kiến thức trong thời gian tới.

# CHƯƠNG 1: TỔNG QUAN

## 1.1 Lý do chọn đề tài

Trong bối cảnh công nghệ thông tin phát triển mạnh mẽ, hoạt động mua sắm trực tuyến ngày càng trở nên phổ biến, đặc biệt trong lĩnh vực linh kiện điện tử. Người dùng có xu hướng tìm kiếm, so sánh thông số kỹ thuật, giá cả và đánh giá sản phẩm trên Internet trước khi đưa ra quyết định mua hàng. Tuy nhiên, nhiều cửa hàng kinh doanh linh kiện điện tử vẫn quản lý theo phương thức truyền thống, chưa có hệ thống trực tuyến hỗ trợ tiếp cận khách hàng hiệu quả.

Việc xây dựng website bán linh kiện điện tử không chỉ đáp ứng nhu cầu mua sắm tiện lợi của người tiêu dùng, mà còn hỗ trợ chủ cửa hàng trong việc quản lý sản phẩm, tồn kho, đơn hàng và thông tin khách hàng một cách khoa học và chính xác. Đồng thời, một hệ thống thương mại điện tử còn góp phần mở rộng phạm vi kinh doanh, nâng cao khả năng cạnh tranh và hỗ trợ quá trình chuyển đổi số.

Từ yêu cầu thực tiễn này, cùng với định hướng từ Giảng viên Đoàn Phước Miền và mong muốn vận dụng kiến thức lập trình web vào bài toán ứng dụng cụ thể, em lựa chọn thực hiện đề tài “Xây dựng website bán linh kiện điện tử”. Đề tài có ý nghĩa thiết thực khi góp phần tin học hóa công tác quản lý kinh doanh, đồng thời mang giá trị học thuật qua việc áp dụng kiến thức về phân tích hệ thống, thiết kế cơ sở dữ liệu, lập trình web và triển khai ứng dụng vào môi trường thực tế.

## 1.2 Mục tiêu nghiên cứu

Mục tiêu chính của đề tài là xây dựng một hệ thống website thương mại điện tử bán linh kiện điện tử hoàn chỉnh, đáp ứng yêu cầu thực tiễn và áp dụng hiệu quả kiến thức chuyên môn đã học. Cụ thể:

- Vận dụng kiến thức chuyên môn: Ứng dụng các lý thuyết về lập trình web, đặc biệt là ASP.NET, mô hình MVC, ngôn ngữ C#, cơ sở dữ liệu SQL Server, HTML/CSS, Bootstrap để phát triển hệ thống thực tế.

- Nâng cao kỹ năng thực hành: Tìm hiểu quy trình phát triển ứng dụng web bằng ASP.NET, cách kết nối và truy vấn dữ liệu với SQL Server, cũng như tổ chức xử lý nghiệp vụ trong môi trường triển khai thực tế.

- Thiết kế giao diện thân thiện: Xây dựng giao diện trực quan, dễ thao tác cho cả khách hàng và quản trị viên. Giao diện hỗ trợ người dùng tìm kiếm, so sánh và xem chi tiết linh kiện, đồng thời giúp quản trị viên quản lý sản phẩm, đơn hàng và dữ liệu hiệu quả.

- Tối ưu trải nghiệm người dùng (UX/UI): Đảm bảo tính thẩm mỹ, tiện dụng và nâng cao trải nghiệm mua sắm trực tuyến, góp phần xây dựng website hoàn chỉnh, dễ sử dụng và vận hành ổn định. xây dựng một website thương mại điện tử hoàn chỉnh và dễ sử dụng.

## 1.3 Phương pháp nghiên cứu

- Phương pháp nghiên cứu tài liệu: Thu thập, tìm hiểu và tổng hợp kiến thức liên quan đến C#, ASP.NET, MVC, HTML/CSS, JavaScript và SQL Server. Nghiên cứu giáo trình, tài liệu trực tuyến, diễn đàn lập trình và các website bán linh kiện điện tử hiện có để tham khảo cách tổ chức giao diện, bố cục chức năng và quy trình xử lý nghiệp vụ.

- Phương pháp phân tích hệ thống: Phân tích yêu cầu từ góc độ người dùng và quản trị viên để xác định danh sách chức năng cần thiết. Xây dựng sơ đồ chức năng, sơ đồ luồng dữ liệu (DFD) và mô hình thực thể – quan hệ (ERD) nhằm thiết kế hệ thống logic, khoa học và dễ triển khai.

- Phương pháp thiết kế và lập trình: Áp dụng quy trình phát triển phần mềm để thiết kế giao diện, xây dựng cơ sở dữ liệu và lập trình các chức năng theo mô hình MVC hoặc phân lớp. Sử dụng Visual Studio và SQL Server để triển khai các chức năng tìm kiếm sản phẩm, xem chi tiết, giỏ hàng, đặt hàng và quản lý danh mục, sản phẩm, đơn hàng.

- Phương pháp thực nghiệm: Chạy thử hệ thống để kiểm tra độ chính xác và ổn định của các chức năng. Ghi nhận lỗi, điều chỉnh giao diện và tối ưu hiệu năng nhằm hoàn thiện website.

## 1.4 Đối tượng nghiên cứu

- Quản trị viên (Admin): Người quản lý toàn bộ nội dung và dữ liệu trên website, bao gồm sản phẩm, danh mục linh kiện, đơn hàng và tài khoản người dùng. Nghiên cứu yêu cầu nghiệp vụ và các tính năng cần thiết để xây dựng hệ thống quản trị hiệu quả và dễ thao tác.

- Người dùng: Truy cập website để tìm kiếm, xem thông tin linh kiện, so sánh thông số kỹ thuật, thêm sản phẩm vào giỏ hàng, đặt mua và theo dõi đơn hàng. Nghiên cứu hành vi và trải nghiệm người dùng nhằm xây dựng giao diện trực quan, dễ sử dụng.

- Hệ thống: Nghiên cứu cấu trúc và cách thức hoạt động của hệ thống, bao gồm các chức năng chính, kiến trúc phần mềm, tổ chức mã nguồn, giao diện người dùng và tương tác dữ liệu.

- Công nghệ ASP.NET và các công cụ hỗ trợ: Nghiên cứu cách sử dụng ASP.NET (Web Forms hoặc MVC), C#, HTML/CSS, Bootstrap, JavaScript và SQL Server để phát triển website. Khai thác các framework và thư viện nhằm nâng cao hiệu quả lập trình, bảo mật, tối ưu hiệu suất và trải nghiệm người dùng.

## 1.5 Phạm vi nghiên cứu

- Về chức năng hệ thống: Website tập trung vào các chức năng cơ bản của một hệ thống thương mại điện tử bán linh kiện điện tử, bao gồm:

+ Đăng ký, đăng nhập và quản lý thông tin người dùng: Cho phép khách hàng tạo tài khoản, đăng nhập và quản lý thông tin cá nhân.

+ Hiển thị danh mục linh kiện và thông tin chi tiết sản phẩm: Người dùng có thể tra cứu danh sách sản phẩm, xem chi tiết thông số kỹ thuật, hình ảnh và giá cả.

+ Tìm kiếm và lọc linh kiện: Hỗ trợ người dùng tìm kiếm sản phẩm theo tên, loại, hãng sản xuất hoặc mức giá.

+ Quản lý giỏ hàng và thực hiện đặt hàng: Cho phép khách hàng thêm sản phẩm vào giỏ hàng, chỉnh sửa số lượng và đặt hàng trực tuyến.

+ Quản lý từ phía quản trị viên (Admin): Bao gồm quản lý sản phẩm, danh mục linh kiện, đơn hàng và thông tin người dùng để duy trì hoạt động website ổn định và hiệu quả.

- Về công nghệ sử dụng

Hệ thống phát triển trên nền tảng ASP.NET (Web Forms hoặc MVC), C#, SQL Server, kết hợp HTML/CSS, Bootstrap và JavaScript. Áp dụng mô hình MVC hoặc 3-tier giúp tách biệt Model, View và Controller, tạo cấu trúc mã nguồn khoa học, dễ bảo trì và mở rộng.

- Về quy mô hệ thống

+ Hệ thống được xây dựng ở mức độ mô phỏng, tập trung vào các chức năng cơ bản của một website thương mại điện tử như quản lý sản phẩm, giỏ hàng, đặt hàng và quản lý người dùng. Hiện tại, hệ thống chưa tích hợp chức năng cao hơn hay các giải pháp bảo mật nâng cao, nhằm đảm bảo phạm vi nghiên cứu phù hợp với thời gian và năng lực của người thực hiện. Việc này giúp tập trung vào việc phân tích, thiết kế và triển khai các tính năng cốt lõi trước khi mở rộng sang các chức năng nâng cao trong tương lai.

+ Dữ liệu về sản phẩm và người dùng mang tính thử nghiệm, chủ yếu phục vụ mục đích học tập, kiểm thử và trình bày chuyên đề. Quy mô này cho phép đánh giá hiệu quả các chức năng cơ bản, kiểm tra tương tác giữa người dùng và hệ thống, đồng thời tạo tiền đề để nghiên cứu, phát triển và mở rộng website bán linh kiện điện tử trong các dự án thực tế sau này.

- Về thời gian thực hiện

Thực hiện trong phạm vi học phần chuyên đề, do đó phạm vi nghiên cứu và phát triển được giới hạn ở mức cơ bản, tập trung vào việc triển khai các chức năng cốt lõi, thiết kế giao diện, cơ sở dữ liệu và kiểm thử hệ thống, đồng thời tạo nền tảng cho việc mở rộng, nâng cấp trong các dự án sau này.

## 1.6 Ý nghĩa khoa học và thực tiễn của đề tài

### 1.6.1 Ý nghĩa khoa học

Làm rõ quy trình xây dựng hệ thống web theo mô hình MVC.

Vận dụng kiến thức về phân tích – thiết kế hệ thống, lập trình web và quản trị cơ sở dữ liệu.

Góp phần cung cấp tài liệu tham khảo cho những nghiên cứu, phát triển liên quan đến thương mại điện tử.

### 1.6.2 Ý nghĩa thực tiễn

Giúp các cửa hàng linh kiện điện tử dễ dàng tiếp cận khách hàng qua nền tảng trực tuyến.

Hỗ trợ chủ cửa hàng quản lý sản phẩm, đơn hàng và khách hàng một cách hiệu quả.

Mang đến cho người dùng trải nghiệm mua sắm thuận tiện, nhanh chóng và tin cậy.

Tạo môi trường thực hành giúp sinh viên nâng cao kỹ năng lập trình web và vận dụng kiến thức vào thực tế.

## 1.7 Tính cấp thiết của đề tài

Trong xu thế chuyển đổi số diễn ra mạnh mẽ, việc áp dụng thương mại điện tử vào hoạt động kinh doanh trở thành yêu cầu cấp bách. Lĩnh vực linh kiện điện tử có số lượng sản phẩm đa dạng, thông số kỹ thuật phức tạp, nên việc quản lý thủ công dễ gây sai sót, khó đáp ứng yêu cầu tra cứu nhanh của khách hàng. Website bán linh kiện điện tử sẽ giúp:

- Hỗ trợ doanh nghiệp tối ưu quy trình quản lý và mở rộng thị trường.

- Mang đến nguồn thông tin rõ ràng, minh bạch và chính xác cho khách hàng.

- Giảm thời gian và chi phí giao dịch.

- Tạo ra một công cụ kinh doanh hiện đại, phù hợp với nhu cầu phát triển của thị trường công nghệ.

Do đó, việc nghiên cứu và xây dựng hệ thống này là hoàn toàn cần thiết, phù hợp với yêu cầu thực tiễn và góp phần thúc đẩy ứng dụng công nghệ vào đời sống.

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## 2.1 Tổng quan về mô hình MVC

### 2.1.1 Khái niệm

MVC là viết tắt của Model – View – Controller, là một mô hình kiến trúc phần mềm phổ biến, giúp phân tách rõ ràng các thành phần của ứng dụng nhằm tăng tính tổ chức, dễ bảo trì và dễ mở rộng. Mô hình này bao gồm ba thành phần chính:

- Model (M) – Tầng dữ liệu và xử lý nghiệp vụ:

+ Đại diện cho dữ liệu và logic nghiệp vụ của ứng dụng (ví dụ: tính toán giá, kiểm tra tồn kho).

+ Là nơi xử lý các thao tác với cơ sở dữ liệu (CSDL) như truy vấn, thêm, sửa, xóa dữ liệu (CRUD).

+ Model không phụ thuộc vào giao diện hiển thị.

- View (V) – Tầng giao diện người dùng

+ Là phần hiển thị thông tin ra cho người dùng dưới dạng HTML, CSS và JavaScript.

+ View nhận dữ liệu từ Controller và trình bày, không xử lý logic nghiệp vụ.

- Controller (C) – Tầng điều khiển

+ Là trung gian giữa Model và View.

+ Nhận yêu cầu từ người dùng (Request), xử lý yêu cầu đó, tương tác với Model để xử lý nghiệp vụ, sau đó chọn View thích hợp để hiển thị kết quả.

### 2.1.2 Luồng xử lý trong MVC

Luồng xử lý diễn ra theo các bước tuần tự khi người dùng tương tác với hệ thống:

- Người dùng gửi Yêu cầu (Request): Người dùng tương tác trên giao diện (View), gửi yêu cầu (ví dụ: nhấn nút "Đăng nhập") đến Controller.

- Controller xử lý và gọi Model: Controller tiếp nhận yêu cầu, xử lý logic ban đầu, sau đó gọi đến các phương thức tương ứng của Model để xử lý dữ liệu và logic nghiệp vụ.

- Model thao tác dữ liệu: Model tương tác với CSDL (thực hiện CRUD) và trả dữ liệu/trạng thái cho Controller.

- Controller chọn View: Dựa trên kết quả xử lý từ Model, Controller quyết định View nào sẽ được hiển thị và truyền dữ liệu cần thiết cho View.

- View hiển thị: View nhận dữ liệu và hiển thị ra giao diện cho người dùng, hoàn thành một vòng lặp xử lý.

### 2.1.3 Vai trò MVC trong phát triển website

- Phân tách rõ ràng dữ liệu, giao diện và logic nghiệp vụ, giúp quản lý mã nguồn khoa học.

- Dễ bảo trì, mở rộng và tích hợp các tính năng mới.

- Tăng khả năng kiểm thử và ứng dụng tốt cho các hệ thống web thương mại điện tử, nơi có nhiều chức năng quản lý và hiển thị dữ liệu động.

## 2.2 Công nghệ ASP.NET MVC

### 2.2.1 Tổng quan

ASP.NET MVC là một framework phát triển ứng dụng web của Microsoft, được xây dựng trên nền tảng ASP.NET và áp dụng kiến trúc MVC.

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

- Xây dựng ứng dụng web linh hoạt, dễ mở rộng và dễ kiểm thử (Testable).

- Tách biệt rõ ràng các mối quan tâm (Separation of Concerns) giữa logic dữ liệu, điều khiển và giao diện.

### 2.2.2 Lịch sử phát triển

2009: Microsoft công bố ASP.NET MVC tại hội nghị MIX, phiên bản 1.0 ra mắt giữa năm.

2010: ASP.NET MVC 2 cải tiến khả năng validation, hỗ trợ khuôn mẫu dữ liệu (Data Annotations) và area.

2011: ASP.NET MVC 3 giới thiệu Razor View Engine, hỗ trợ dependency injection và filters nâng cao.

2012: ASP.NET MVC 4 bổ sung Web API, hỗ trợ phát triển RESTful services và cải thiện mobile templates.

2013: ASP.NET MVC 5 tích hợp tính năng ASP.NET Identity, attribute routing và nâng cấp Bootstrap mặc định.

2016: ASP.NET Core MVC trong .NET Core 1.0 ra mắt, phát triển đa nền tảng, hiệu năng cao, modular hóa gói NuGet.

Hiện tại: ASP.NET Core MVC tiếp tục được cập nhật song song với .NET 5/6/7/8, tối ưu hiệu năng và hỗ trợ minimal APIs.

### 2.2.3 Tính năng của ASP.NET MVC

ASP.NET MVC được ưu tiên sử dụng nhờ các tính năng cốt lõi sau:

- Hệ thống định tuyến (Routing): Cho phép ánh xạ các URL thân thiện với người dùng (URL Friendly) đến các hành động (Action) cụ thể trong Controller, giúp cải thiện khả năng SEO và tổ chức ứng dụng.

- Engine Razor View: Cho phép chèn trực tiếp mã C# vào trong mã HTML bằng cú pháp đơn giản (@), ngắn gọn và dễ đọc, nâng cao hiệu quả hiển thị dữ liệu động.

- Model Binding & Validation: Giúp tự động chuyển đổi dữ liệu từ các biểu mẫu (form) hoặc tham số URL sang các đối tượng Model trong C#, đồng thời hỗ trợ kiểm tra tính hợp lệ của dữ liệu đầu vào thông qua các thuộc tính (Data Annotation).

- Khả năng kiểm thử (Testability): Cấu trúc tách biệt giúp việc viết và chạy các bài kiểm thử đơn vị (Unit Test) cho các thành phần (Controller, Model) trở nên dễ dàng.

## 2.3 Giới thiệu ngôn ngữ lập trình C# (C Sharp)

### 2.3.1 Tổng quan

C# (C Sharp) là ngôn ngữ lập trình hướng đối tượng, do Microsoft phát triển vào năm 2000. Là một phần của nền tảng .NET, C# được thiết kế hiện đại, an toàn và dễ sử dụng, đồng thời hỗ trợ đầy đủ các nguyên lý lập trình hướng đối tượng như đóng gói, kế thừa và đa hình.

### 2.3.2 Đặc điểm nổi bật

- Hướng đối tượng: Quản lý dữ liệu và logic nghiệp vụ thông qua các lớp và đối tượng.

- Tích hợp tốt với .NET: Hỗ trợ phát triển đa dạng ứng dụng từ web, desktop, mobile đến game và cloud.

- Cú pháp hiện đại: Dễ đọc, dễ học với lập trình viên đã quen C, C++ hoặc Java.

- Quản lý bộ nhớ: Tích hợp Garbage Collection, giảm rủi ro rò rỉ bộ nhớ.

- LINQ: Thao tác dữ liệu trực tiếp từ ngôn ngữ lập trình mà không cần viết SQL thủ công.

### 2.3.3 Ứng dụng trong phát triển website

- Kết hợp với ASP.NET MVC để xây dựng các ứng dụng web động, quản lý logic nghiệp vụ và dữ liệu.

- Tạo các lớp Model cho cơ sở dữ liệu, xử lý nghiệp vụ và liên kết với Controller.

- Phối hợp với các thư viện .NET khác để nâng cao bảo mật, hiệu năng và khả năng mở rộng của hệ thống.

## 2.4 Giới thiệu các Controls của ASP.NET

### 2.4.1 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.

### 2.4.2 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.

### 2.4.3 Web Server Controls

- Là nhóm control do ASP.NET cung cấp, tự sinh HTML và tương tác mạnh mẽ hơn HTML Controls.

- Nhóm nhập liệu: TextBox, Button, CheckBox, RadioButton, DropDownList, ListBox.

- Nhóm hiển thị: Label, Image, HyperLink, Literal.

- Nhóm container: Panel, PlaceHolder, MultiView.

- Nhóm dữ liệu: GridView, Repeater, DataList, FormView, DetailsView.

- Nhóm điều hướng: Menu, TreeView, SiteMapPath.

- Nhóm đăng nhập/đăng ký: Login, LoginView, LoginStatus, CreateUserWizard.

### 2.4.4 Validation Controls

- Dùng để kiểm tra tính hợp lệ dữ liệu trước khi gửi lên server, nâng cao bảo mật và giảm lỗi nhập liệu.

- Các control phổ biến: RequiredField Validator, Range Validator, Compare Validator, RegularExpression Validator, Custom Validator, Validation Summary.

- Vai trò trong phát triển website bán linh kiện điện tử.

- Giúp xây dựng giao diện thân thiện, dễ thao tác cho khách hàng và quản trị viên.

- Quản lý và hiển thị dữ liệu sản phẩm từ cơ sở dữ liệu một cách linh hoạt.

- Đảm bảo tính hợp lệ dữ liệu, giảm sai sót khi khách hàng nhập thông tin đặt hàng hoặc quản trị viên cập nhật sản phẩm.

## 2.5 Giới thiệu ADO.NET và các đối tượng truy cập CSDL

### 2.5.1 Khái niệm về ADO.NET

- ADO.NET (ActiveX Data Objects .NET) là một công nghệ truy cập dữ liệu trong nền tảng .NET Framework, được Microsoft phát triển nhằm giúp các ứng dụng có thể kết nối, truy vấn, thao tác và quản lý dữ liệu từ nhiều nguồn khác nhau như: SQL Server, Oracle, MySQL, tệp XML, hoặc các nguồn dữ liệu khác.

- ADO.NET đóng vai trò là cầu nối giữa ứng dụng .NET và cơ sở dữ liệu, cho phép thực hiện các thao tác như:

+ Kết nối cơ sở dữ liệu

+ Gửi câu lệnh SQL

+ Nhận dữ liệu trả về

+ Thêm, sửa, xóa dữ liệu

### 2.5.2 Mô hình hoạt động của ADO.NET

ADO.NET hoạt động theo 2 mô hình chính: Kết nối (Connected) và Ngắt kết nối (Disconnected).

- Mô hình Kết nối (Connected): Ứng dụng mở kết nối đến CSDL, thực thi lệnh và đọc dữ liệu trực tiếp rồi đóng kết nối.

+ Đối tượng sử dụng (Connected): Connection, Command, DataReader.

+ Ưu điểm (Connected): Nhanh, xử lý dữ liệu trực tiếp.

- Mô hình Ngắt kết nối (Disconnected): Dữ liệu được lấy về bộ nhớ và xử lý offline, không cần duy trì kết nối liên tục.

+ Đối tượng sử dụng (Disconnected): DataAdapter, DataSet, DataTable.

+ Ưu điểm (Disconnected): Tiết kiệm tài nguyên, dễ xử lý nhiều bảng dữ liệu.

### 2.5.3 Các đối tượng chính trong ADO.NET

- Connection: Tạo và quản lý kết nối đến cơ sở dữ liệu.

- Command: Thực thi câu lệnh SQL hoặc Stored Procedure.

- DataReader: Đọc dữ liệu nhanh theo dạng chỉ tiến (forward-only), yêu cầu kết nối mở.

- DataAdapter: Cầu nối giữa DataSet và cơ sở dữ liệu, dùng để lấy hoặc cập nhật dữ liệu.

- DataSet: Tập hợp dữ liệu dạng ngắt kết nối, chứa nhiều bảng.

- DataTable: Đại diện cho một bảng trong DataSet.

- DataRow: Một dòng dữ liệu trong DataTable.

- DataColumn: Một cột dữ liệu trong DataTable.

- Parameter: Tham số truy vấn dùng trong Command, giúp tránh SQL Injection.

- DataRelation: Thiết lập quan hệ giữa các bảng trong DataSet.

### 2.5.4 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.

## 2.6 Giới thiệu về SQL Server

### 2.6.1 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, ...

### 2.6.2 Lịch sử phát triển

- SQL Server bắt đầu từ năm 1989.

- Từ 1995, Microsoft tự phát triển độc lập.

- Các bước nhảy lớn: SQL Server 7.0, 2005, 2012, 2017, 2019.

- Phiên bản mới nhất (hiện nay): SQL Server 2022, tích hợp sâu với điện toán đám mây.

### 2.6.3 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.

### 2.6.4 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.

## 2.7 Giới thiệu về Bootstrap

### 2.7.1 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.

### 2.7.2 Mục tiêu và ưu điểm của Bootstrap

#### 2.7.2.1. 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.

#### 2.7.2.2. 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.

### 2.7.3 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.

### 2.7.4 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.

## 2.8 Giới thiệu về Entity Framework

### 2.8.1 Giới thiệu về Entity Framework

- Entity Framework (EF) là một Object-Relational Mapping (ORM) framework của Microsoft dành cho .NET, giúp lập trình viên làm việc với cơ sở dữ liệu bằng cách thao tác trực tiếp với các đối tượng C# hoặc VB.NET, thay vì viết câu lệnh SQL thủ công.

- 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.

### 2.8.2 Mục tiêu và vai trò của Entity Framework

Mục tiêu Entity Framework được Microsoft phát triển nhằm:

- Đơn giản hóa việc thao tác dữ liệu: Cho phép lập trình viên làm việc với cơ sở dữ liệu thông qua các đối tượng trong C# hoặc VB.NET thay vì viết SQL thủ công.

- Tăng năng suất lập trình: Tự động sinh truy vấn, quản lý kết nối và ánh xạ dữ liệu giúp tiết kiệm thời gian phát triển.

- Hỗ trợ các mô hình phát triển linh hoạt: Bao gồm Database First, Code First và Model First.

- Đảm bảo tính nhất quán và an toàn dữ liệu: Quản lý giao dịch, ràng buộc và dữ liệu phức tạp một cách dễ dàng.

Vai trò của Entity Framework:

- ORM Framework: Chuyển đổi dữ liệu giữa cơ sở dữ liệu quan hệ và các lớp đối tượng.

- Trung gian giữa ứng dụng và cơ sở dữ liệu: Giúp lập trình viên tập trung vào logic ứng dụng thay vì SQL.

- Hỗ trợ CRUD và truy vấn dữ liệu: Thực hiện thao tác tạo, đọc, cập nhật, xóa dữ liệu dễ dàng thông qua LINQ hoặc phương thức sẵn có.

- Tăng khả năng bảo trì và mở rộng ứng dụng: Giảm sự phụ thuộc vào cấu trúc cơ sở dữ liệu và SQL thủ công.

# CHƯƠNG 3: CÀI ĐẶT VÀ THỬ NGHIỆM

## 3.1 Cài đặt

**- Visual Studio 2022:** Môi trường phát triển tích hợp (IDE) mạnh mẽ của Microsoft, dùng để xây dựng ứng dụng Windows, web, di động và đám mây.

**- SQL Server:** Hệ quản trị cơ sở dữ liệu dùng để lưu trữ, cập nhật và truy xuất dữ liệu.

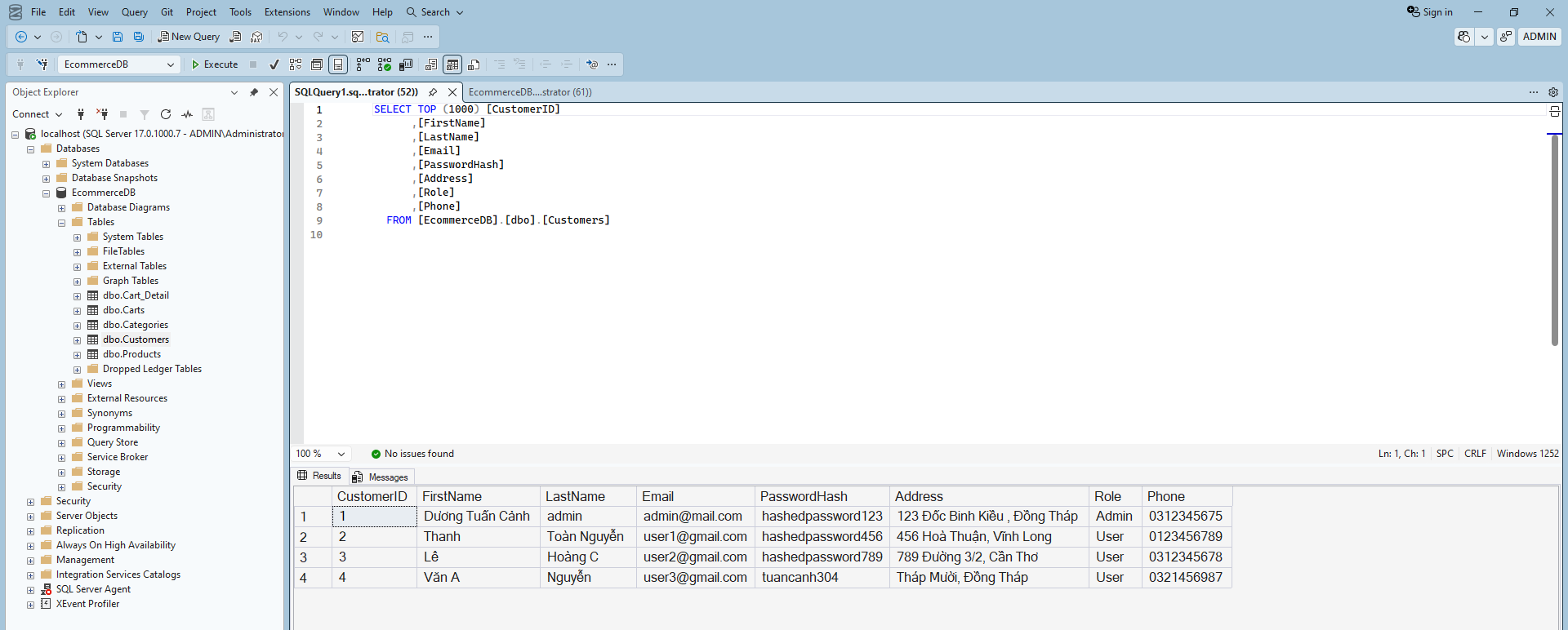
**- SQL Server Management Studio (SSMS):** Công cụ giao diện đồ họa (IDE) hỗ trợ quản lý, cấu hình và phát triển SQL Server cũng như các cơ sở dữ liệu liên quan.

## 3.2 Thử nghiệm

Dữ liệu thực nghiệm là thành phần quan trọng trong quá trình xây dựng website, giúp mô phỏng môi trường hoạt động thực tế. Việc bổ sung dữ liệu này không chỉ hỗ trợ kiểm tra giao diện và chức năng, mà còn giúp phát hiện kịp thời các lỗi liên quan đến hiển thị và xử lý dữ liệu, góp phần nâng cao độ chính xác và tính hoàn thiện của hệ thống.

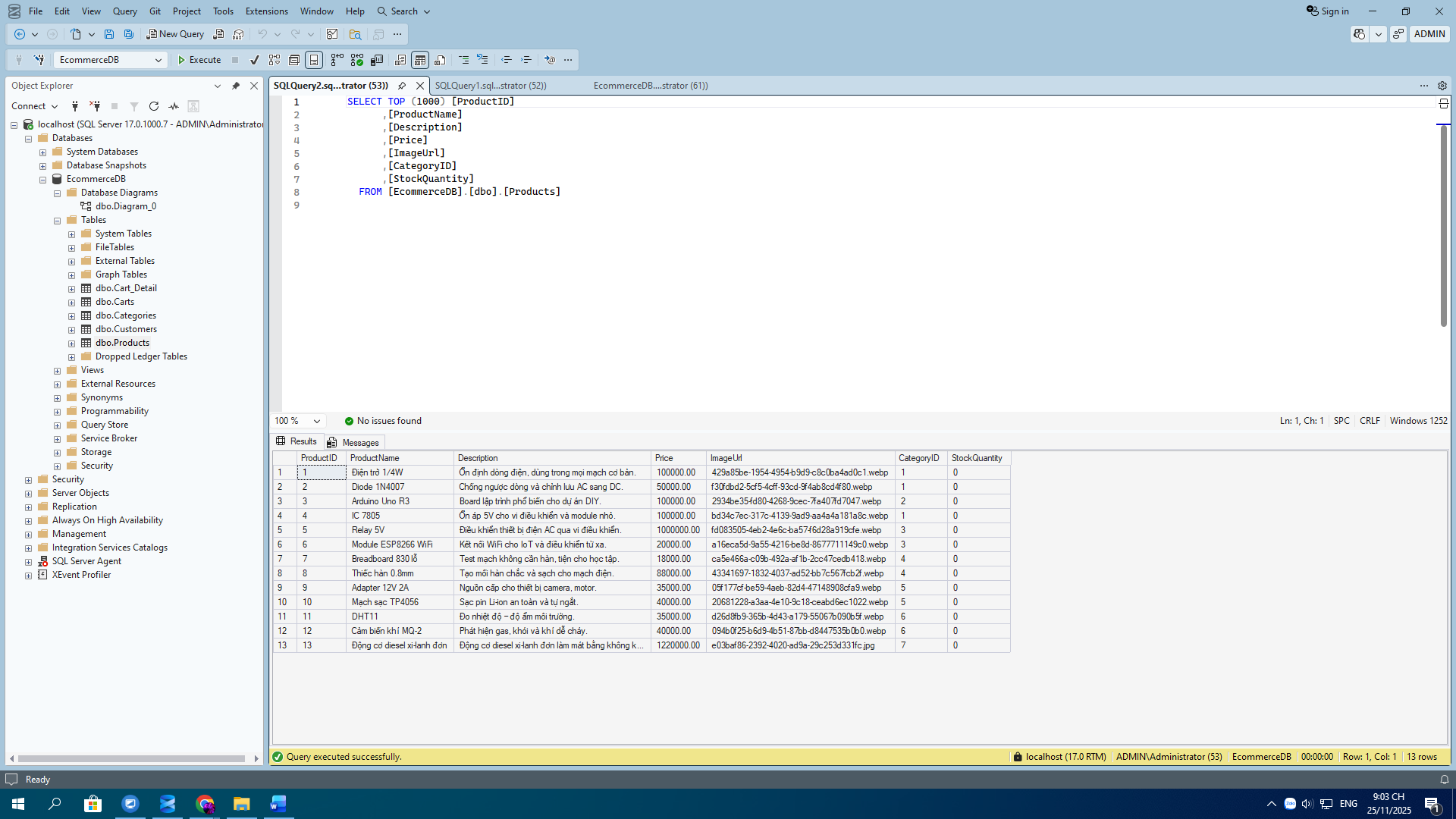
Dưới đây là một số bảng dữ liệu thực nghiệm chính:

Hình 1 Bảng dữ liệu tài khoản người dùng



- Bảng dữ liệu người dùng cho thấy các thông tin tài khoản admin, khách hàng, người dùng đăng ký được lưu vào CSDL của hệ thống

Hình 2 Bảng dữ liệu sản phẩm linh kiện



- Bảng dữ liệu sản phẩm là nơi lưu trữ toàn bộ thông tin liên quan đến các mặt hàng được kinh doanh trên hệ thống, bao gồm: tên sản phẩm, giá bán, mô tả, hình ảnh, số lượng và các thuộc tính khác. Bảng này do quản trị viên (Admin) quản lý thông qua chức năng thêm, sửa hoặc xóa sản phẩm.

- Bảng dữ liệu sản phẩm là nơi lưu trữ toàn bộ thông tin liên quan đến các mặt hàng được kinh doanh trên hệ thống, bao gồm: tên sản phẩm, giá bán, mô tả, hình ảnh, số lượng và các thuộc tính khác. Bảng này do quản trị viên (Admin) quản lý thông qua chức năng thêm, sửa hoặc xóa sản phẩm.

## 3.3 Ứng dụng minh họa

### 3.3.1 Mô tả bài toán

Quản lý khách hàng: Mỗi khách hàng được lưu trữ và quản lý với các thông tin cơ bản: Họ tên, địa chỉ, điện thoại, email, tên đăng nhập, mật khẩu.

Những thông tin này phục vụ cho việc xác thực tài khoản và hỗ trợ giao hàng.

Quản lý linh kiện điện tử (mặt hàng): Mỗi linh kiện điện tử trên website được quản lý với đầy đủ thông tin chi tiết, bao gồm: Tên mặt hàng, đơn giá, số lượng, hình ảnh, mô tả, kích thước, ngày thêm, ngày cập nhật, trạng thái.

Quá trình đặt hàng:

- Khách hàng truy cập website để xem và lựa chọn linh kiện điện tử cần mua.

- Khi chọn một sản phẩm, hệ thống hiển thị đầy đủ thông tin như hình ảnh, đơn giá, mô tả và thông số kỹ thuật.

- Khách hàng có thể thêm sản phẩm vào giỏ hàng với số lượng mong muốn.

- Khách hàng tiến hành đăng nhập.

- Sau khi đăng nhập thành công, khách hàng nhập thông tin nhận hàng.

- Hình thức thanh toán có thể là thanh toán khi nhận hàng (COD).

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

User: User là những người có nhu cầu tìm kiếm và mua sắm các loại linh kiện điện tử. Vì vậy, hệ thống phải đáp ứng các chức năng sau:

- Hiển thị danh sách linh kiện điện tử theo từng loại (vi điều khiển, cảm biến, điện trở – tụ điện, module, phụ kiện…).

- Xem thông tin chi tiết sản phẩm, gồm hình ảnh, giá, mô tả, thông số kỹ thuật, số lượng còn lại.

- Tìm kiếm sản phẩm theo tên, theo loại linh kiện, hoặc theo thông số.

- Phân trang danh sách sản phẩm giúp khách hàng dễ dàng xem các sản phẩm khác.

- Thêm sản phẩm vào giỏ hàng, cập nhật số lượng trong giỏ hàng.

- Thanh toán đơn hàng với hình thức thanh toán khi nhận hàng (COD).

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

Admin: Admin là người sở hữu quyền kiểm soát toàn bộ hệ thống website bán linh kiện điện tử. Admin được cấp tài khoản đặc biệt để đăng nhập vào khu vực quản trị. Nếu như quá trình đăng nhập thành công thì người quản trị có những chức năng sau:

Quản lý sản phẩm: Thêm mới linh kiện điện tử, sửa thông tin linh kiện (giá, mô tả, thông số kỹ thuật, hình ảnh…), xóa sản phẩm không còn kinh doanh

Quản lý danh mục: Thêm danh mục mới, chỉnh sửa danh mục, xóa danh mục

Quản lý đơn hàng: Tiếp nhận và kiểm tra các đơn đặt hàng, xác nhận đơn hàng và giao hàng, cập nhật trạng thái đơn hàng, hủy đơn hàng không hợp lệ.

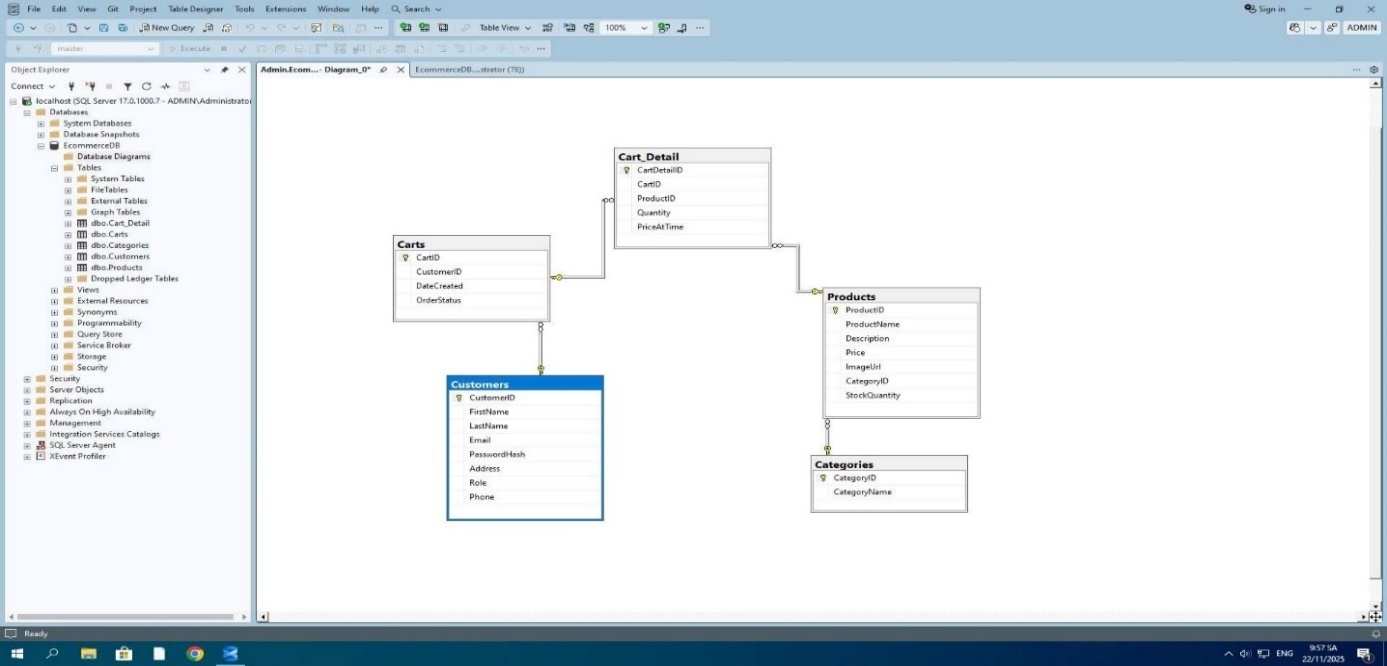
Quản lý tài khoản: Xem danh sách toàn bộ tài khoản người dùng, xem chi tiết thông tin từng tài khoản, đổi mật khẩu, cập nhật thông tin tài khoản người dùng trong trường hợp cần thiết, chỉnh sửa hoặc xóa tài khoản Admin (phân quyền, thay đổi thông tin…), quản lý phân quyền truy cập giữa User và Admin

Thống kê dữ liệu hệ thống, bao gồm: Thống kê số lượng sản phẩm, thống kê đơn hàng, thống kê doanh thu.

### 3.3.2 Thiết kế hệ thống

Mô hình thực thể quan hệ ERD : Entity-Relationship Diagram

Hình 3 Mô hình ERD



### **3.3.3 Yêu cầu chức năng**

#### 3.3.3.1 Chức năng nghiệp vụ

Lưu trữ: Thông tin khách hàng, thông tin sản phẩm, thông tin danh mục sản phẩm, thông tin đơn đặt hàng.

Tra cứu: Tra cứu thông tin sản phẩm, tra cứu thông tin khách hàng, tra cứu thông tin danh mục sản phẩm, tra cứu thông tin đơn đặt hàng.

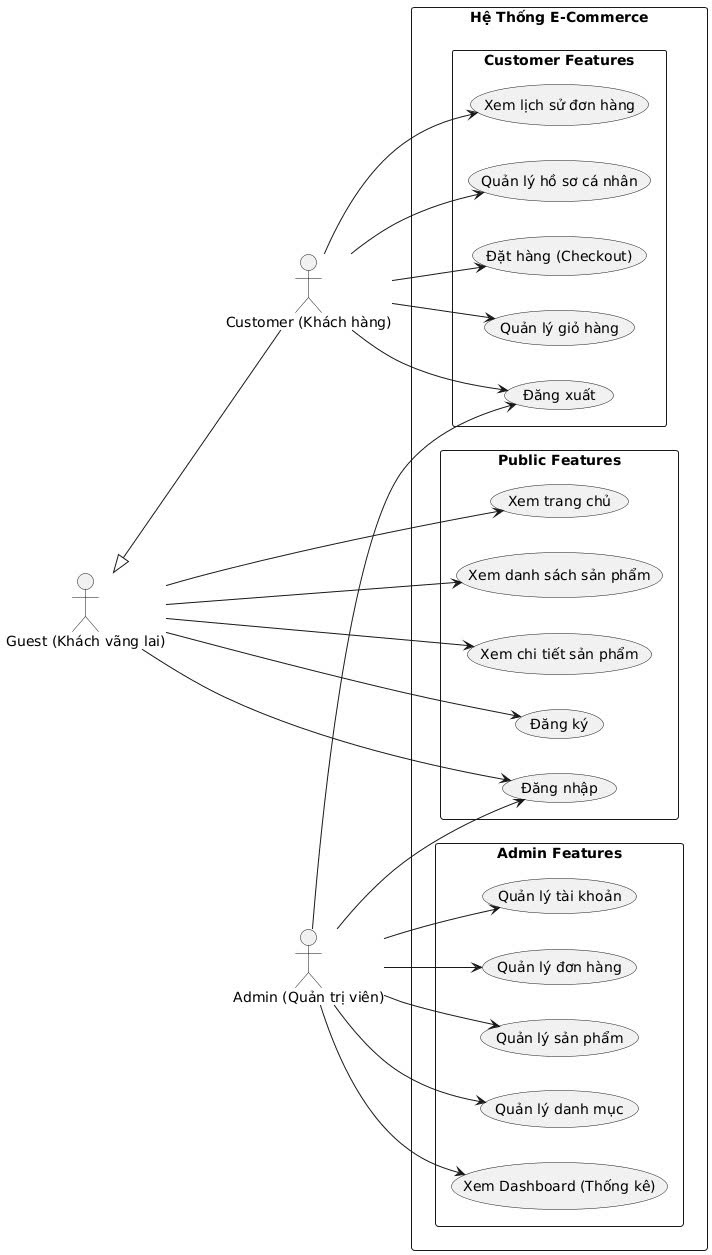
#### 3.3.3.2 Chức năng hệ thống

Đăng nhập: Người quản trị (admin) có toàn quyền trên hệ thống: truy vấn dữ liệu, cập nhật dữ liệu.

Cập nhật thông tin: Người dùng có thể đăng nhập và đổi mật khẩu và cập nhật các thông tin.

Sơ đồ User Case:

Hình 4 Sơ đồ User Case tổng quan



Hệ thống cung cấp ba nhóm chức năng chính tương ứng với ba loại người dùng: **Khách vãng lai (Guest)**, **Khách hàng (Customer)** và **Quản trị viên (Admin)**. Mỗi nhóm có các chức năng riêng nhằm đáp ứng nhu cầu sử dụng và quản lý hệ thống.

**- Chức năng dành cho Khách vãng lai (Guest):** Khách vãng lai là những người dùng chưa có tài khoản hoặc chưa đăng nhập. Họ được phép sử dụng các chức năng công khai của hệ thống: Xem trang chủ, xem danh sách sản phẩm, xem chi tiết sản phẩm, đăng ký tài khoản, đăng nhập hệ thống.

**- Chức năng dành cho Khách hàng (Customer):** Sau khi đăng nhập, khách hàng có thể sử dụng đầy đủ các chức năng phục vụ quá trình mua sắm: Quản lý giỏ hàng, đặt hàng (Checkout), quản lý hồ sơ cá nhân, xem lịch sử đơn hàng, đăng xuất.

**- Chức năng dành cho Quản trị viên (Admin):** Quản trị viên có toàn quyền quản lý dữ liệu và vận hành hệ thống: Quản lý tài khoản người dùng, quản lý đơn hàng, quản lý sản phẩm, quản lý danh mục, xem Dashboard (Thống kê).

# **CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU**

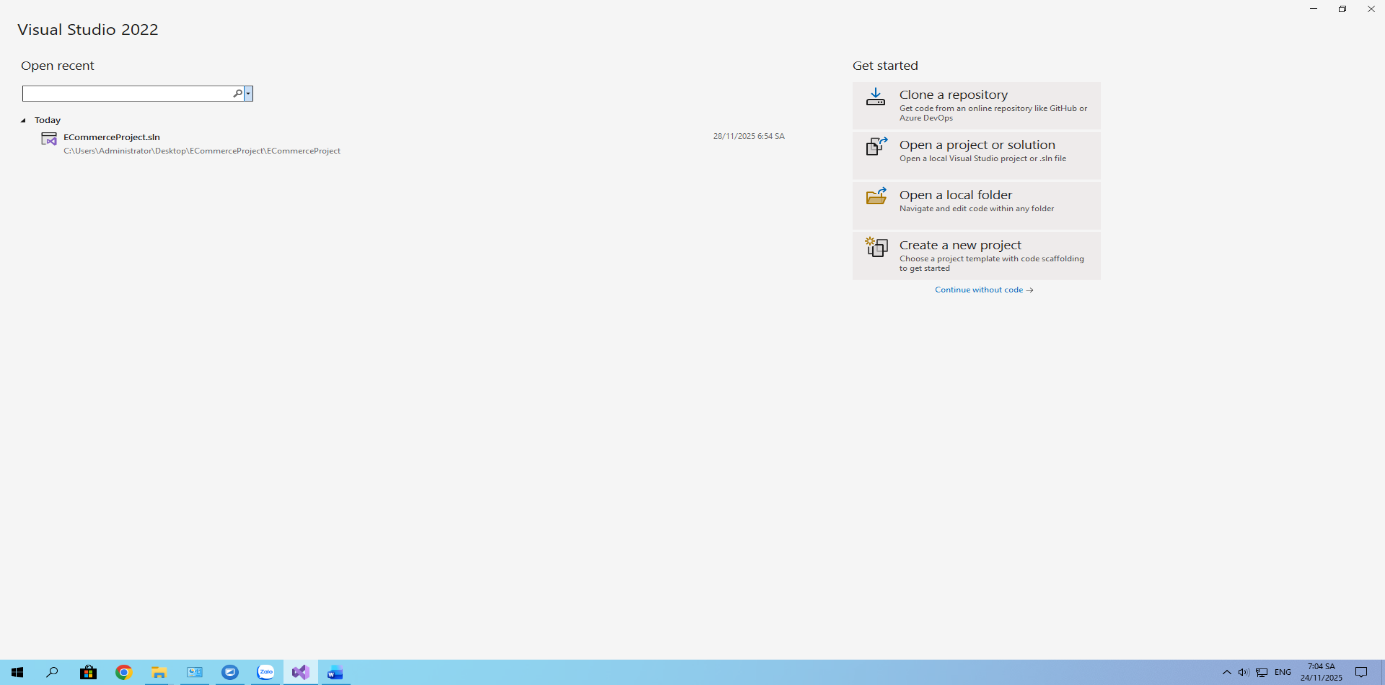
## 4.1 Các công nghệ sử dụng trong đề tài

### 4.1.1 ASP.NET MVC

#### 4.1.1.1 Tạo dự án với ASP.NET MVC

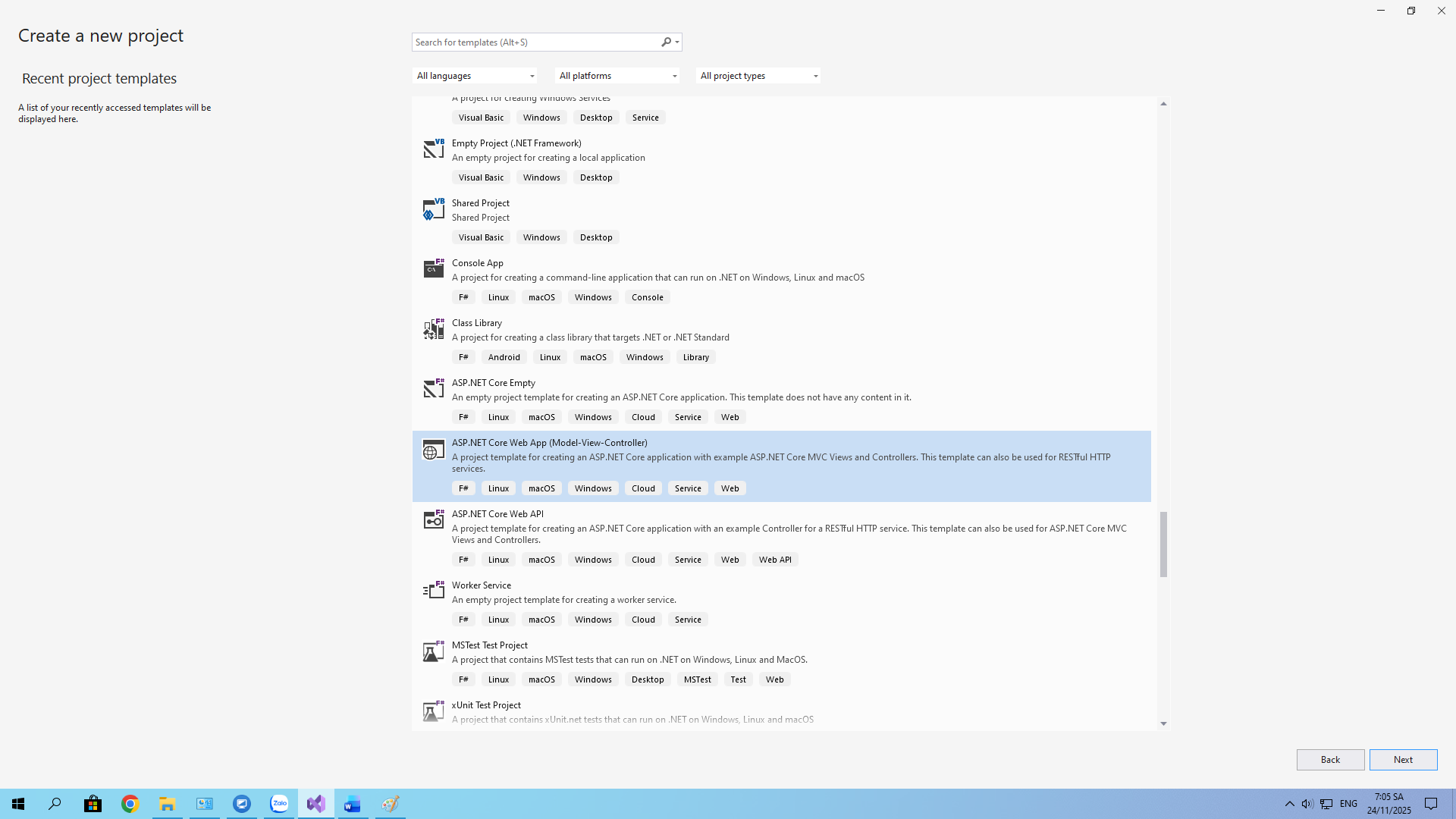
Bước 1: Mở Visual Studio Create a New Project

Hình 5 Bước 1 - Tạo Project mới trên Visual Studio



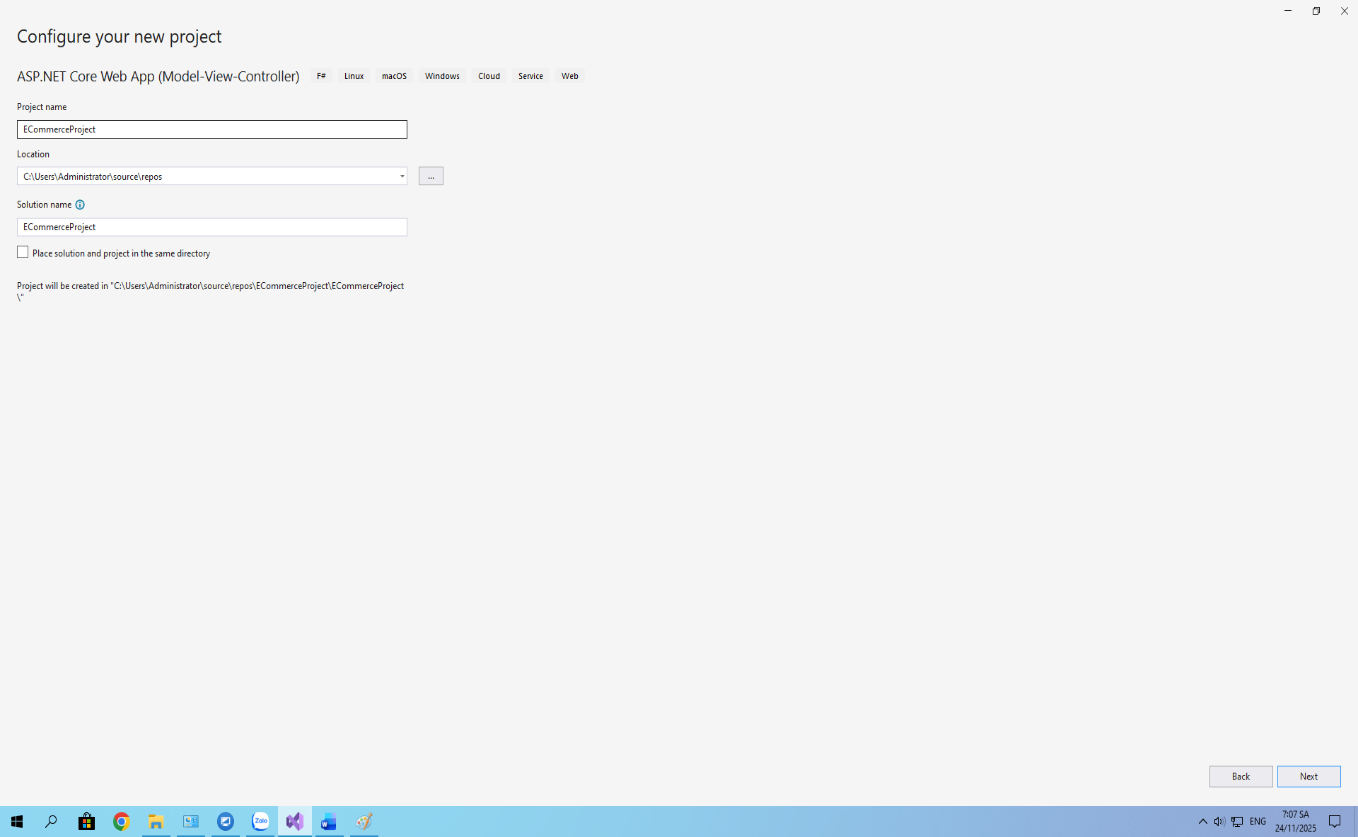
Bước 2: Chọn ASP.NET Core Web App (Model-View-Controller)

Hình 6 Bước 2 – Chọn ASP.NET Core Web App (Model-View-Controller)



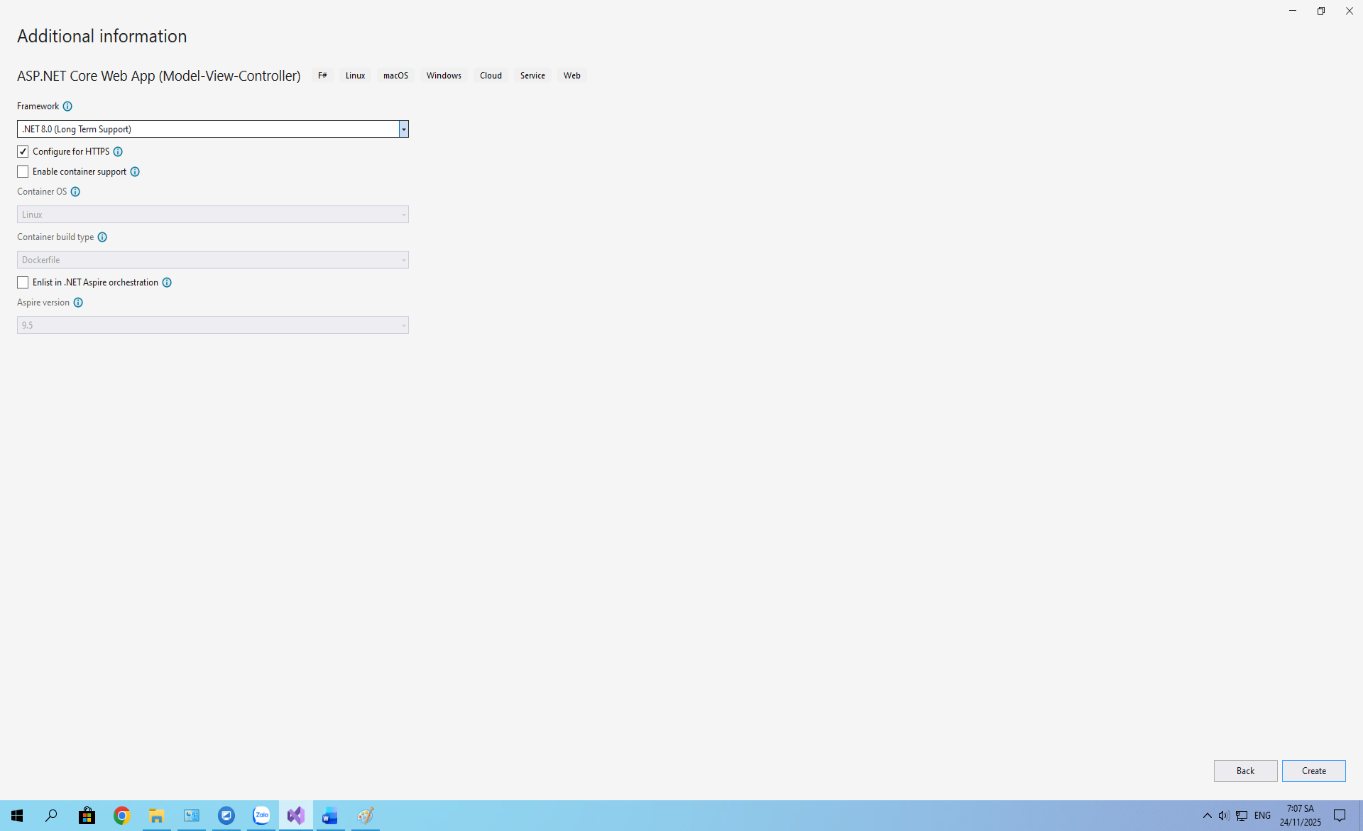
Bước 3: Đặt tên Project, đường dẫn thư mục lưu và nhấn “Next”

Hình 7 Bước 3 – Đặt tên cho Project và đường dẫn lưu



Bước 4: Nhấn “Create” để tiếp tục

Hình 8 Bước 4 – nhấn chọn Create

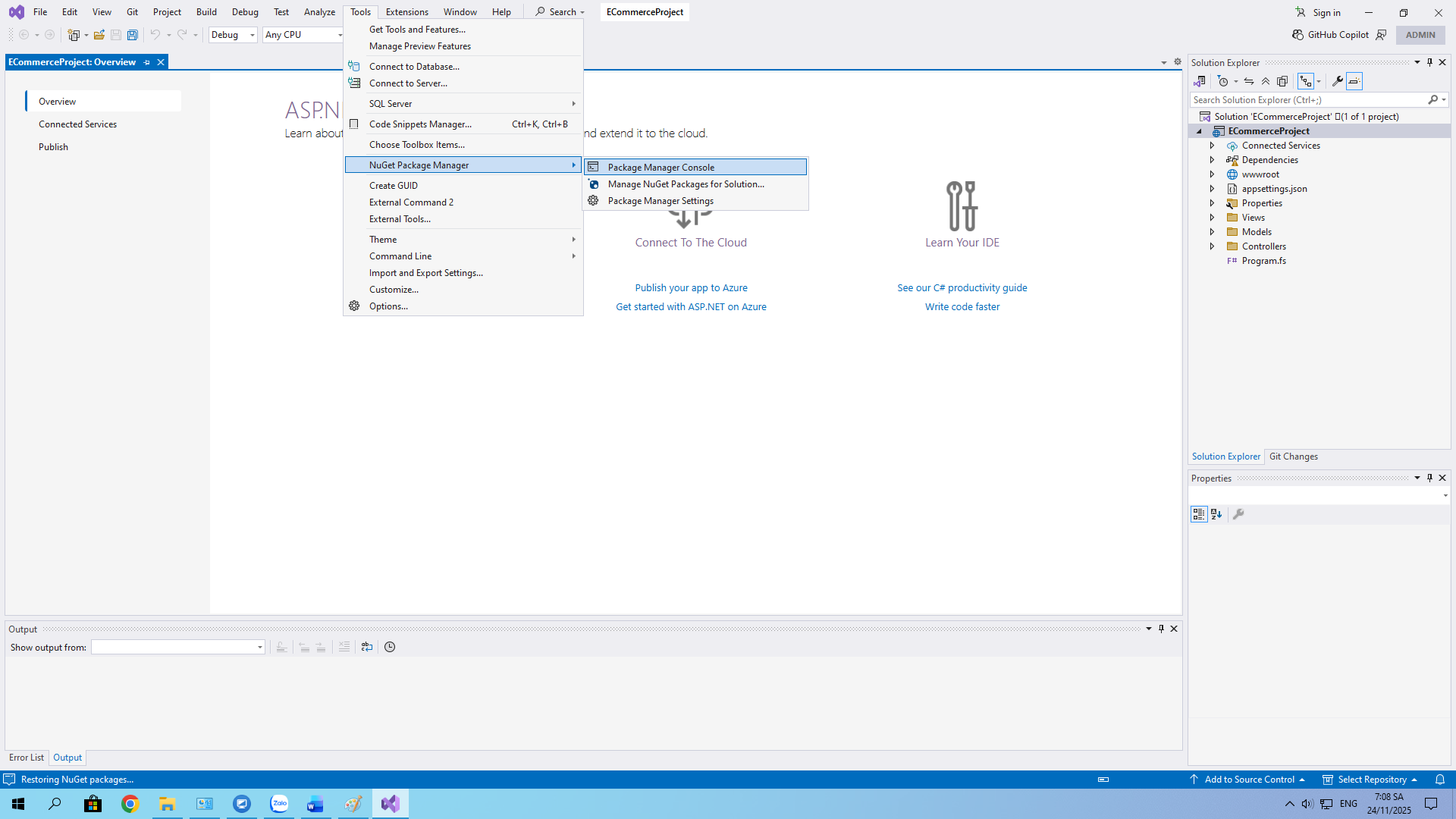


#### 4.1.1.2 Cấu trúc cây thư mục của ASP.NET MVC

Hình 9 Giao diện làm việc và cấu trúc cây thư mục của ASP.NET MVC

### 4.1.2 Entity Framework

#### 4.1.2.1 Cài đặt Entity Framework cho dự án

Vào Tools NuGet Package Manager Package Manager Console để có thể sử dụng dòng lệnh để cài đặt EF Core và provider cho SQL Server

Hình 10 Package Manager Console sử dụng dòng lệnh

### 4.2 Thiết kế giao diện

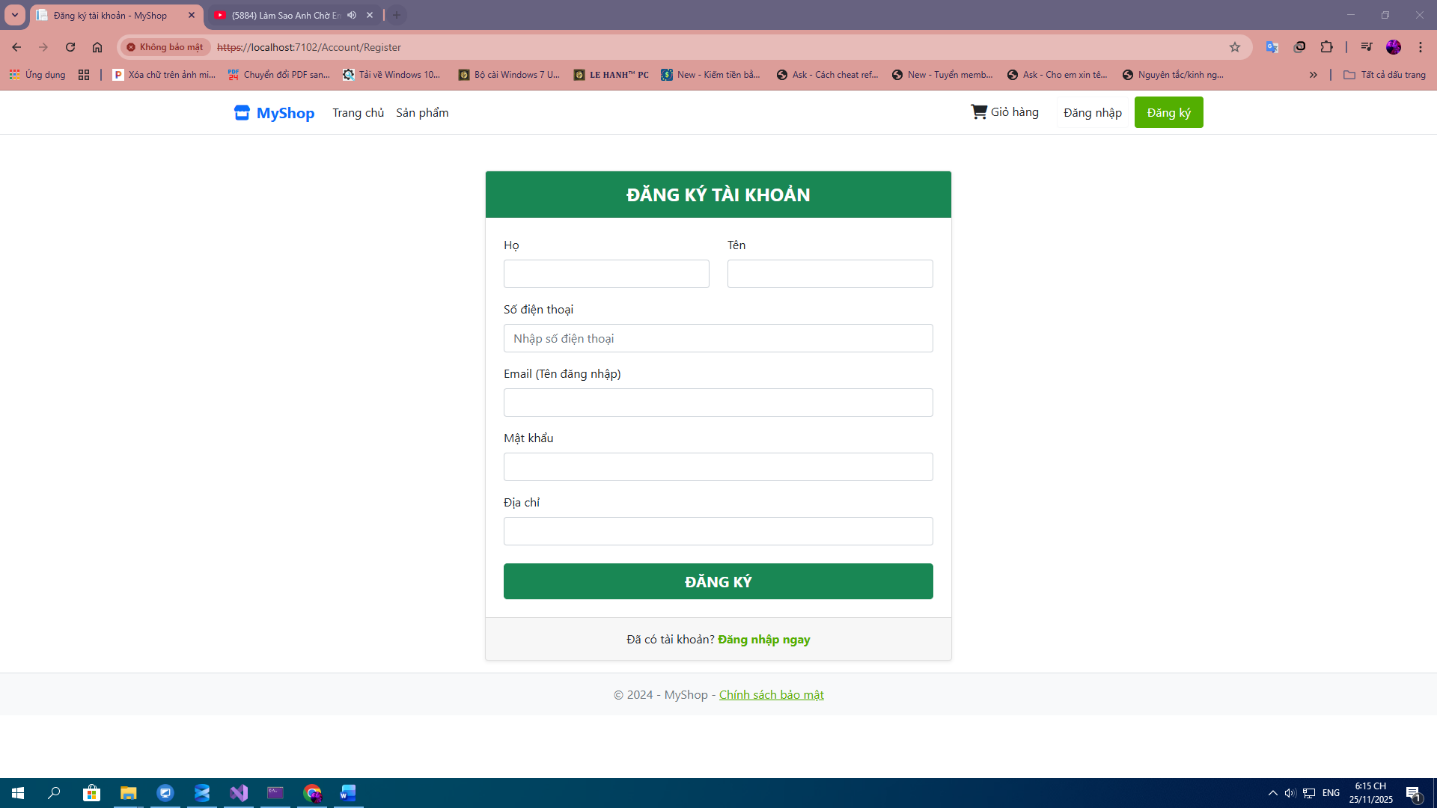
#### 4.2.1 Giao diện đăng nhập và đăng ký

Giao diện đăng nhập: là giao diện dùng để đăng nhập tài khoản người dùng vào hệ thống. Mỗi tài khoản được cấp một quyền riêng trên website. Khách hàng người dùng có thể đăng ký tài khoản để đăng nhập vào hệ thống và thực hiện mua sắm giỏ hàng.

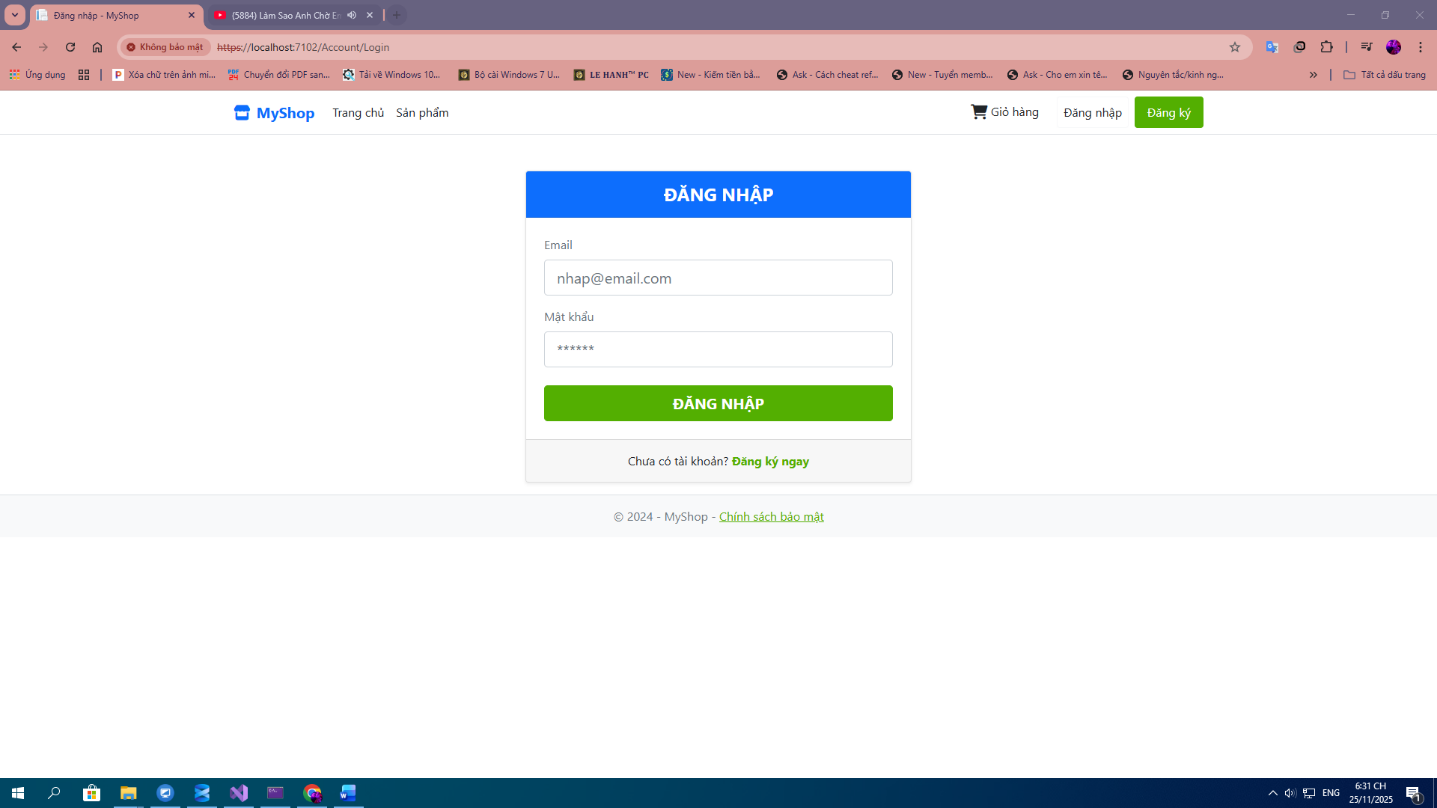
Giao diện trang chủ dành cho khách hàng: dùng để giới thiệu các mặt hàng hiện có trên website. Tại đây, khách hàng có thể xem các sản phẩm nổi bật và truy cập nhanh đến các danh mục sản phẩm khác.

Giao diện trang quản trị (Admin Dashboard): hiển thị đầy đủ các chức năng quản lý dành cho quản trị viên, bao gồm: Quản lý tài khoản người dùng, quản lý sản phẩm, và quản lý đơn hàng.

Hình 11 Giao diện đăng ký tài khoản

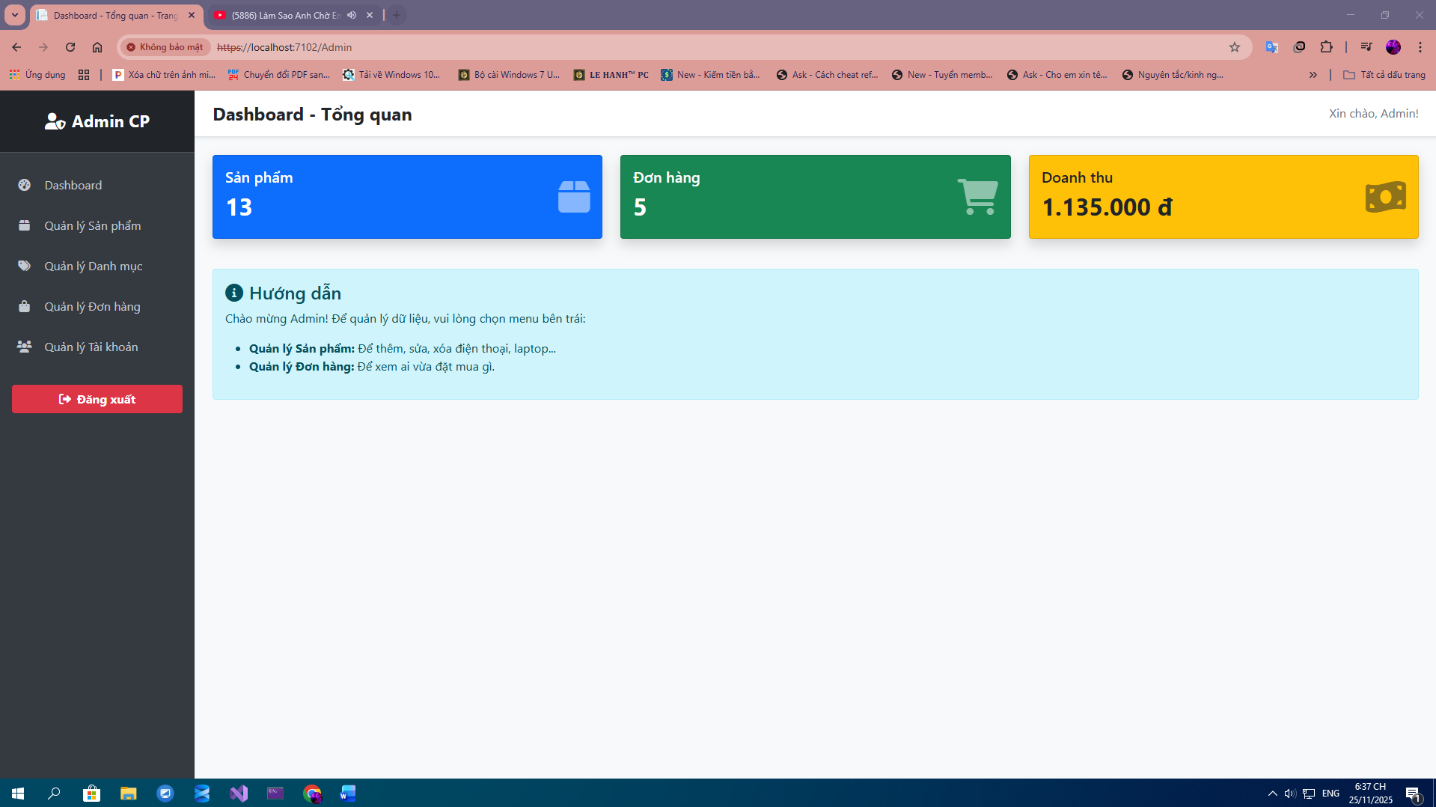


Hình 12 Giao diện đăng nhập tài khoản



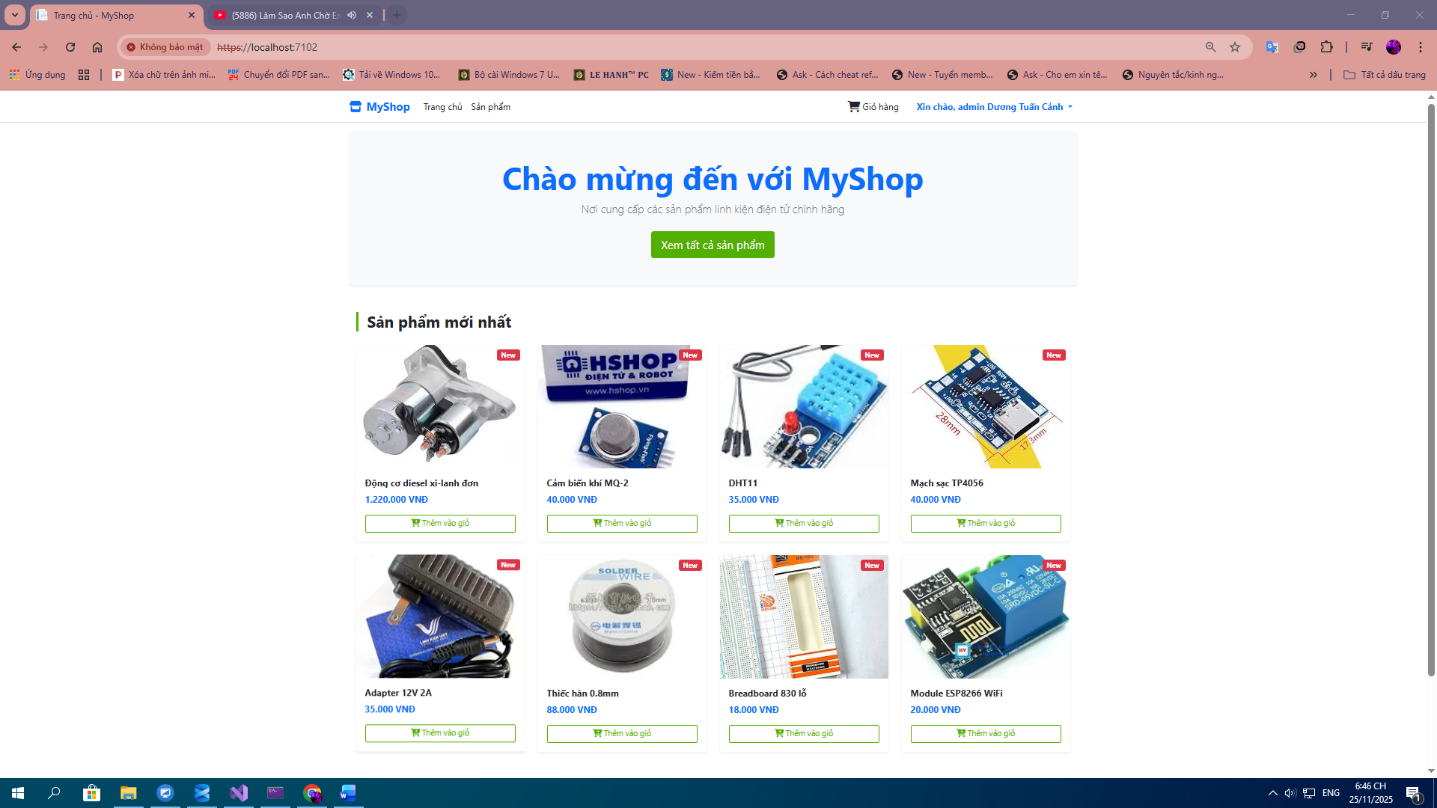
#### 4.2.2 Giao diện trang quản trị

Hình 13 Giao diện trang quản trị



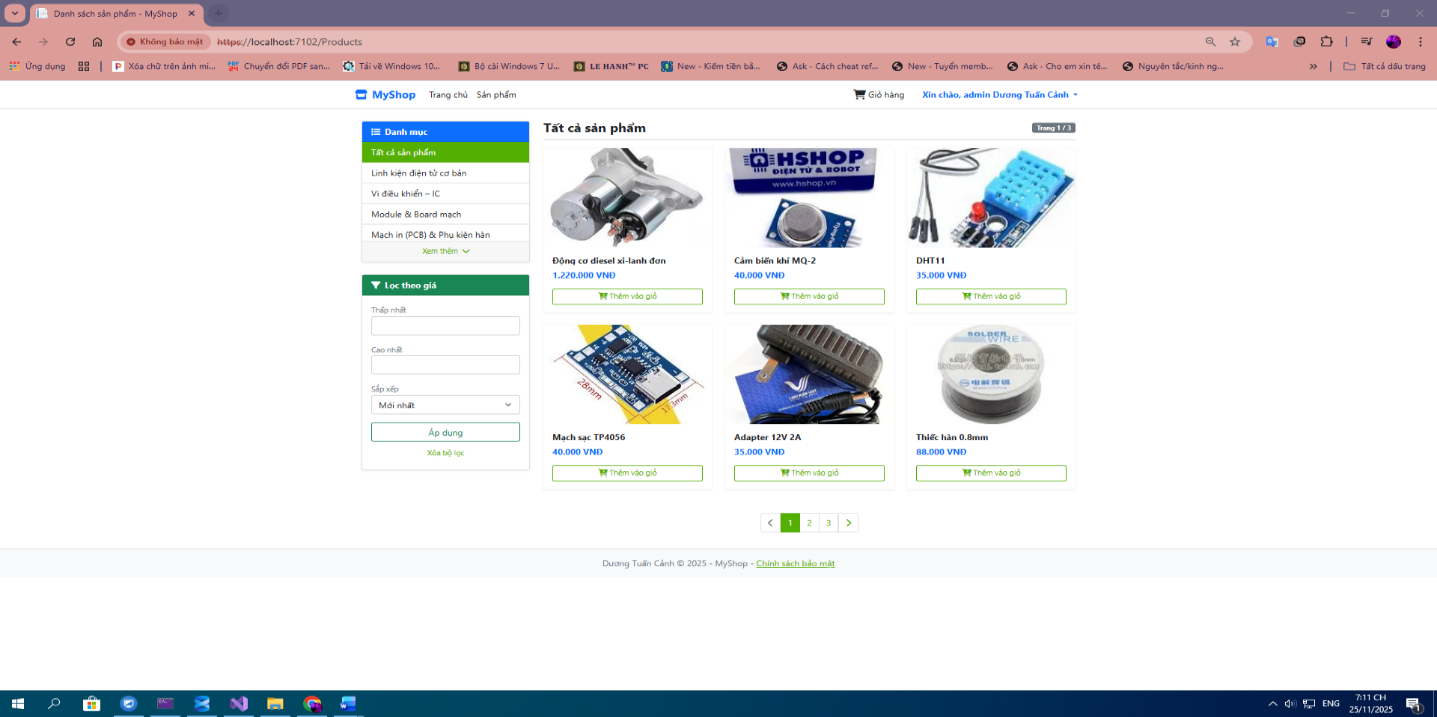
#### 4.2.3 Giao diện trang chủ

Hình 14 Giao diện trang chủ



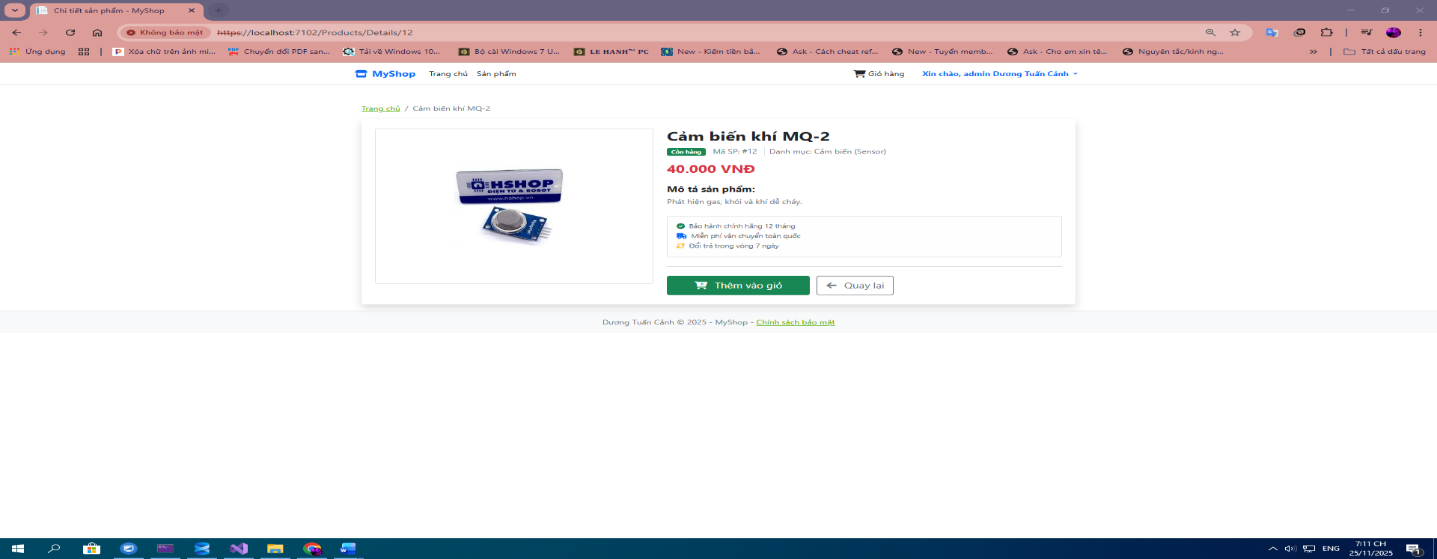
#### 4.2.4 Giao diện trang sản phẩm

Hình 15 Giao diện trang sản phẩm



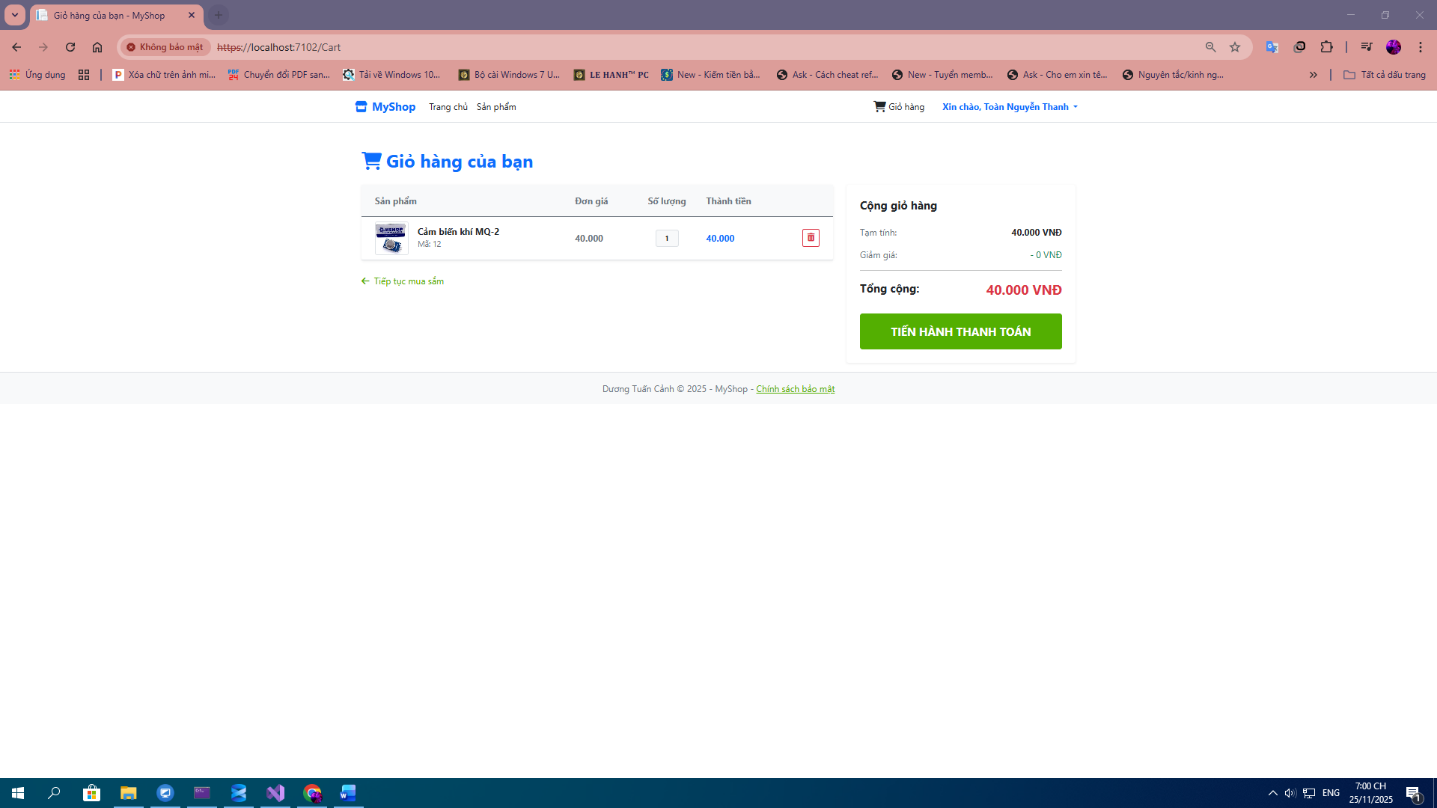
#### 4.2.5 Giao diện trang chi tiết sản phẩm

Hình 16 Giao diện trang chi tiết sản phẩm



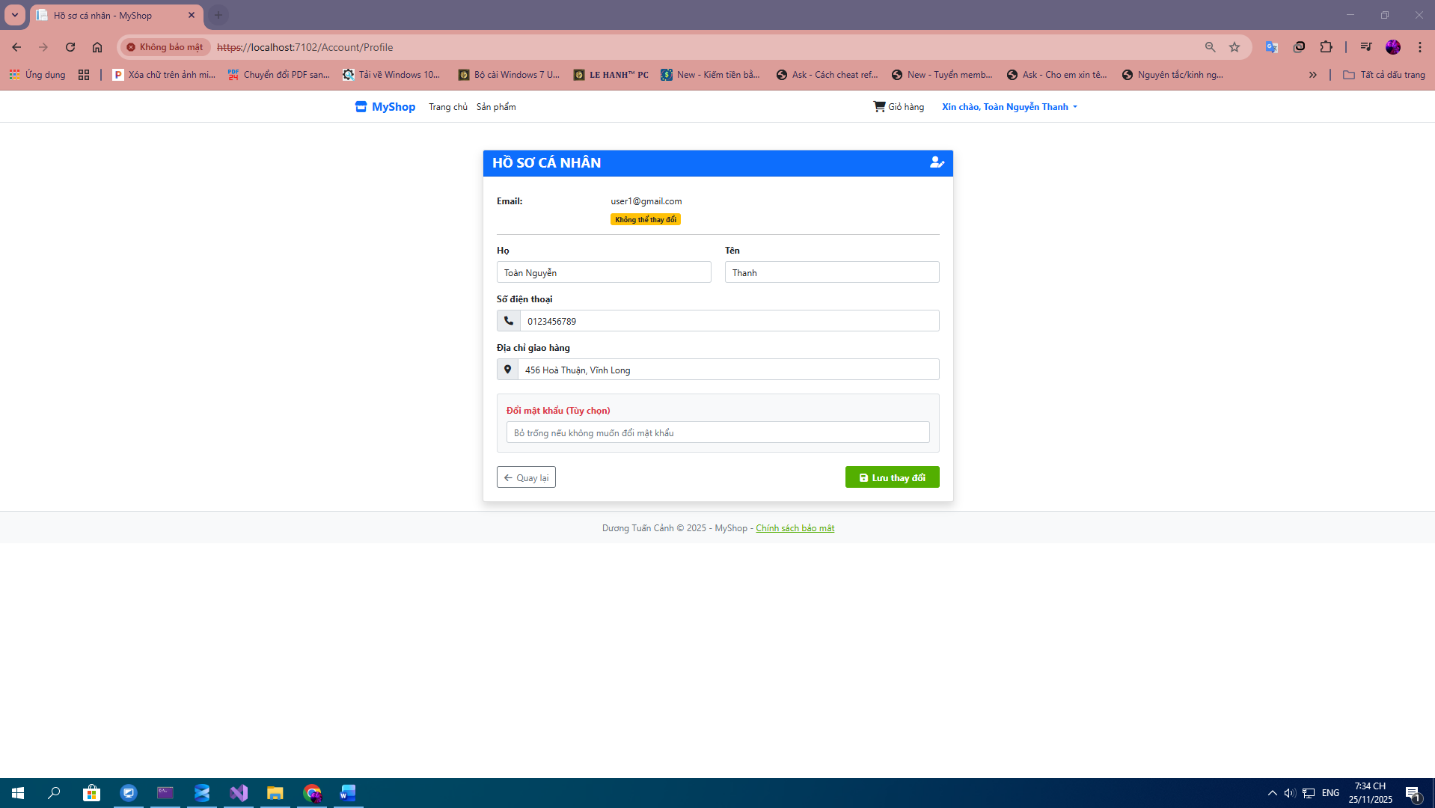
#### 4.2.6 Giao diện giỏ hàng

Hình 17 Giao diện giỏ hàng



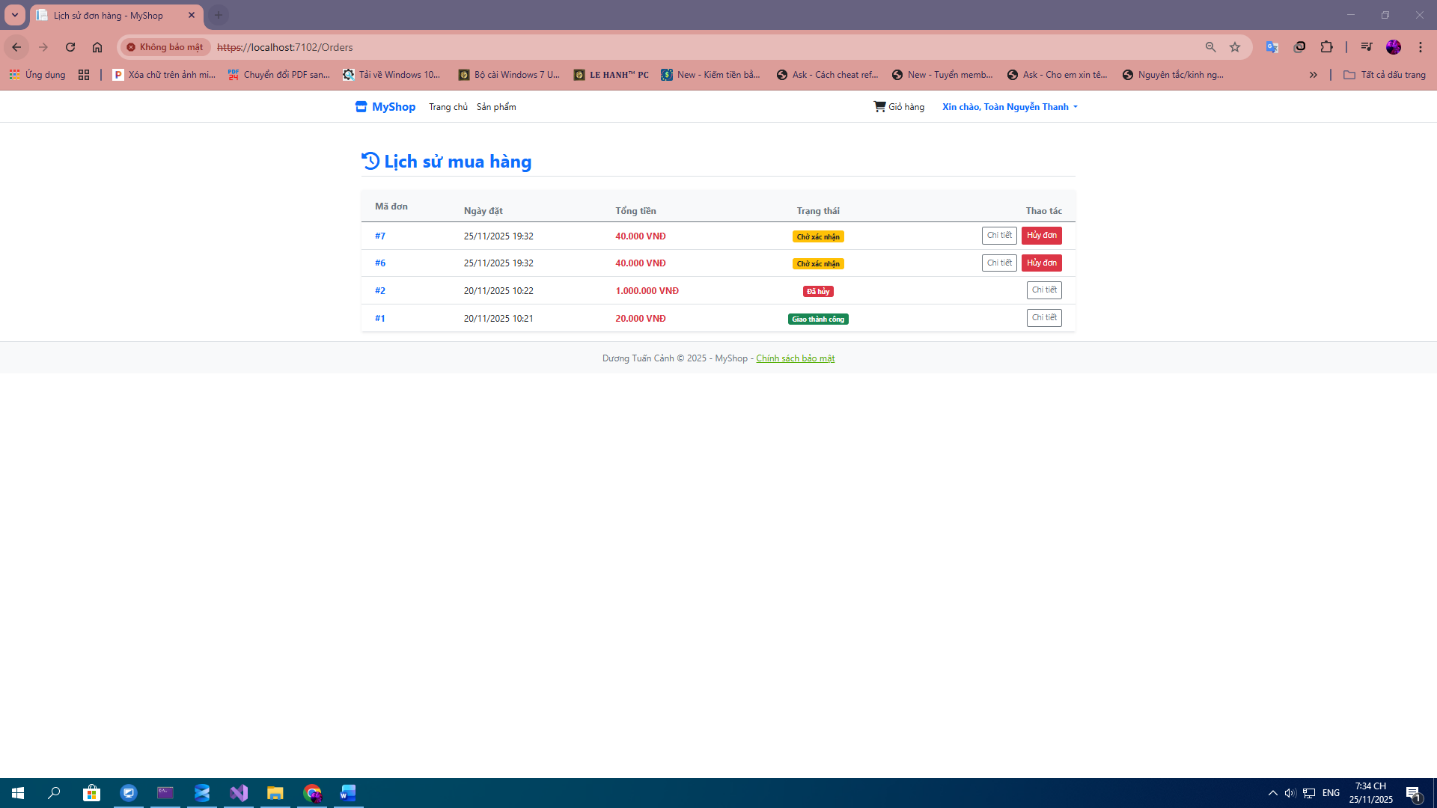
#### 4.2.7 Giao diện thông tin người dùng

Hình 18 Giao diện thông tin người dùng



#### 4.2.8 Giao diện lịch sử mua hàng

Hình 19 Giao diện lịch sử mua hàng

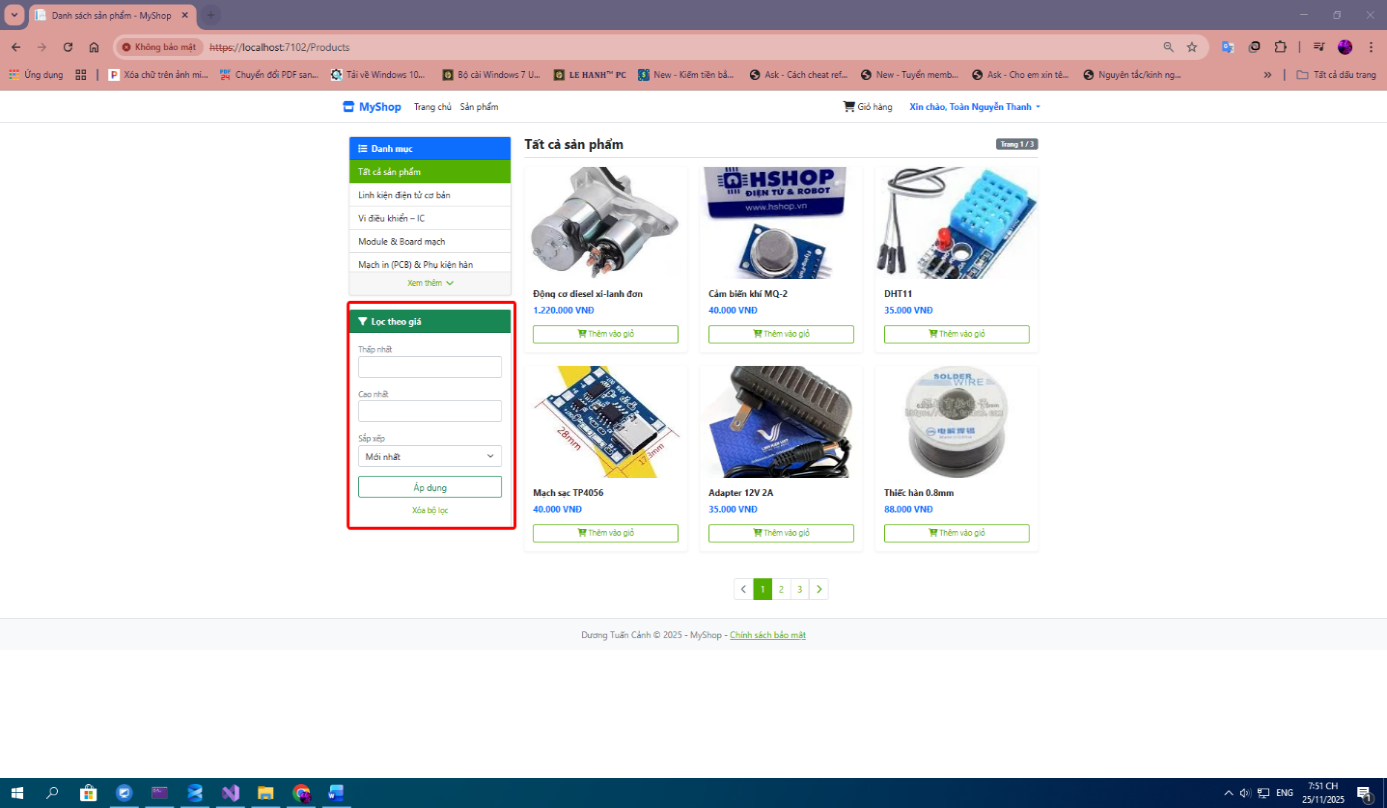


#### 4.2.9 Giao diện danh mục sản phẩm

Hình 20 Giao diện danh mục sản phẩm

#### 4.2.10 Giao diện lọc sản phẩm

Hình 21 Giao diện lọc sản phẩm

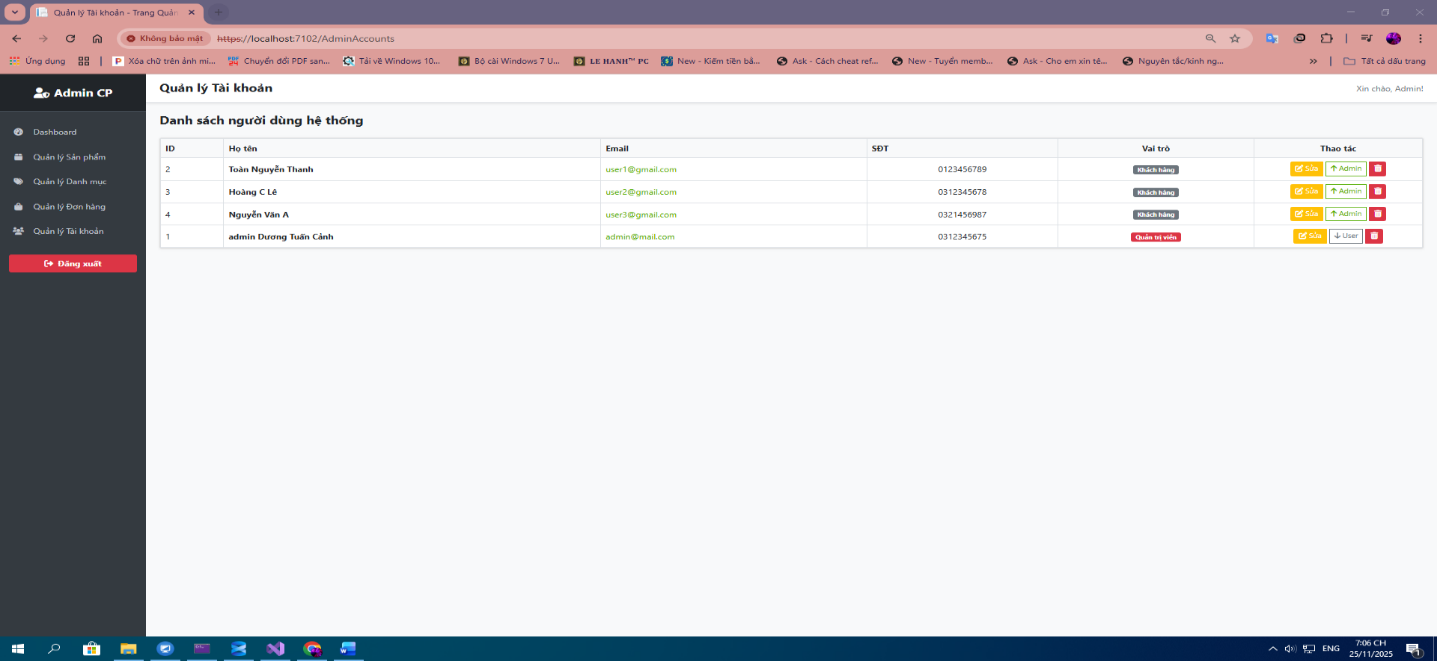


#### 4.2.11 Giao diện trang quản lý đơn hàng

Hình 22 Giao diện trang quản lý đơn hàng



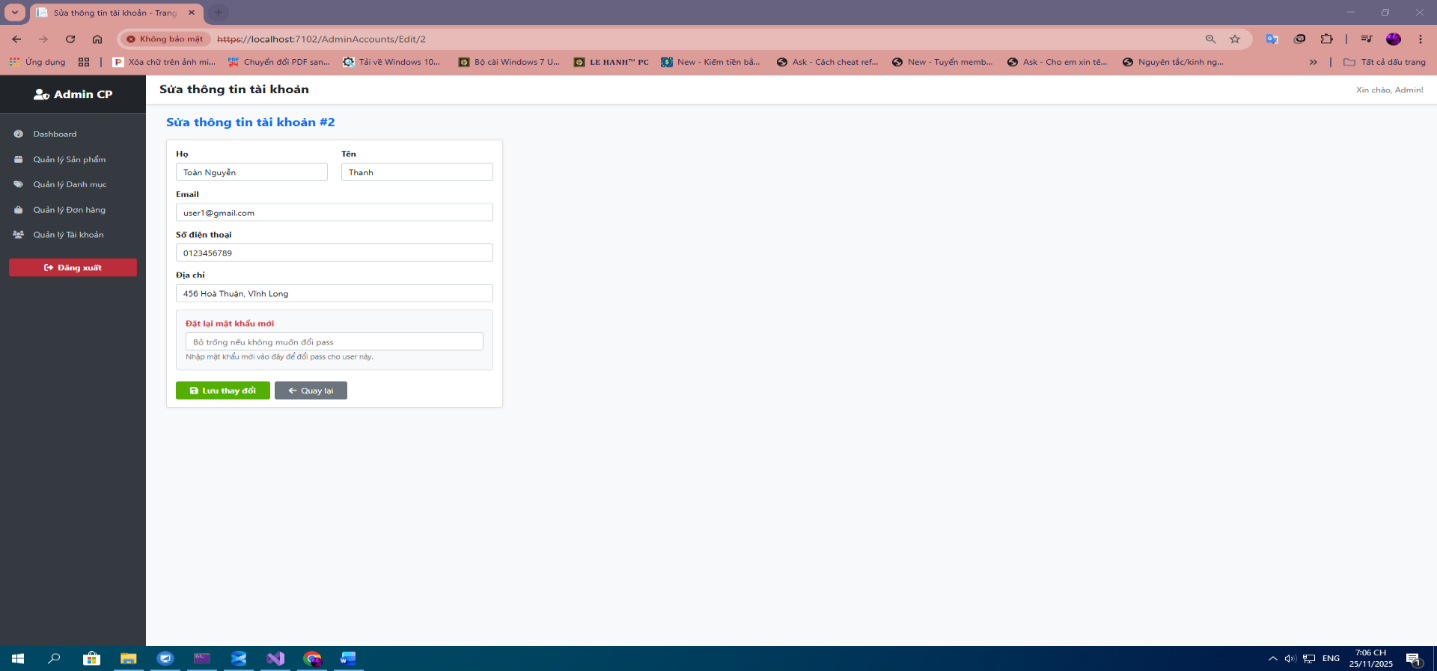
#### 4.2.12 Giao diện quản lý khách hàng



Hình 23 Giao diện quản lý khách hàng

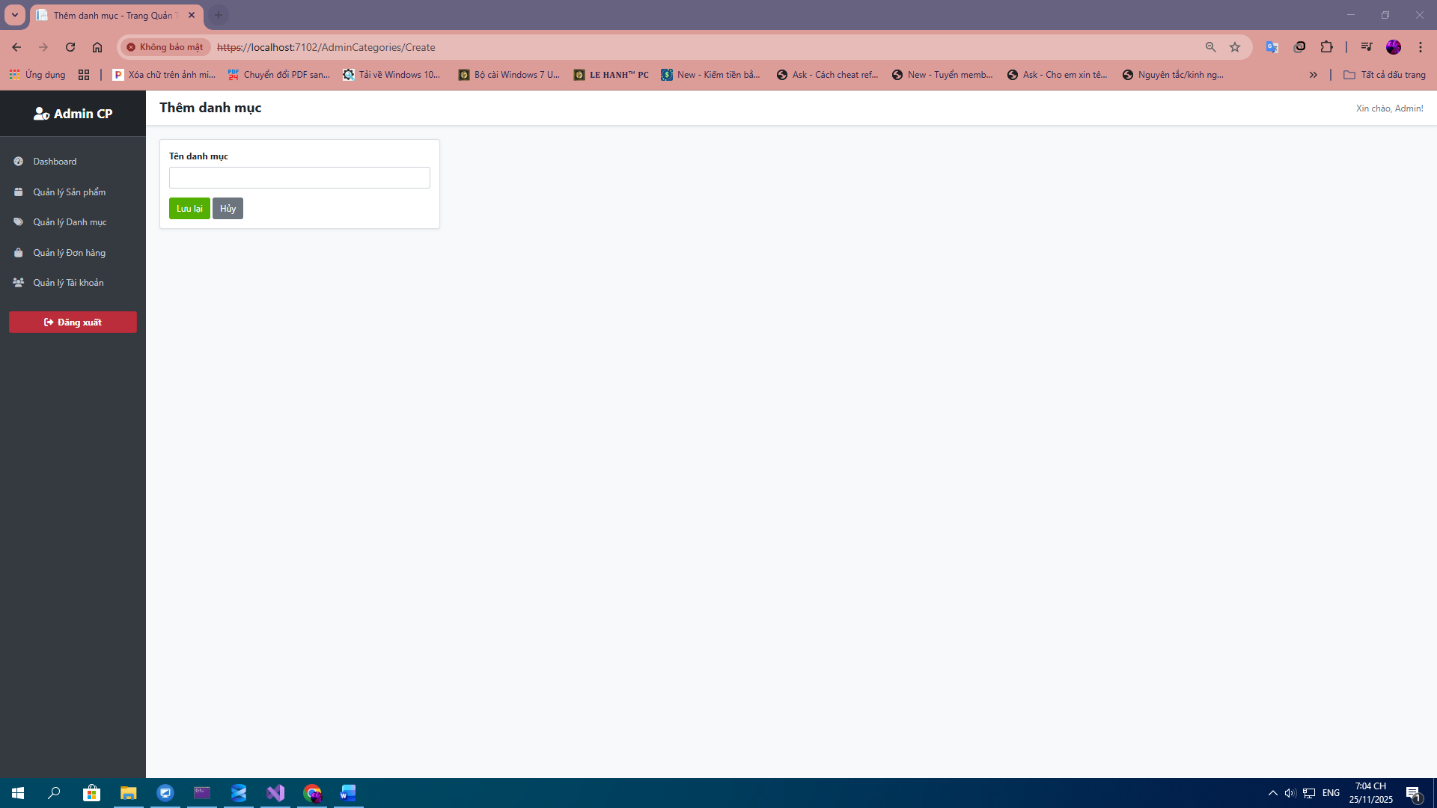
#### 4.2.13 Giao diện sửa thông tin khách hàng

Hình 24 Giao diện sửa thông tin khách hàng

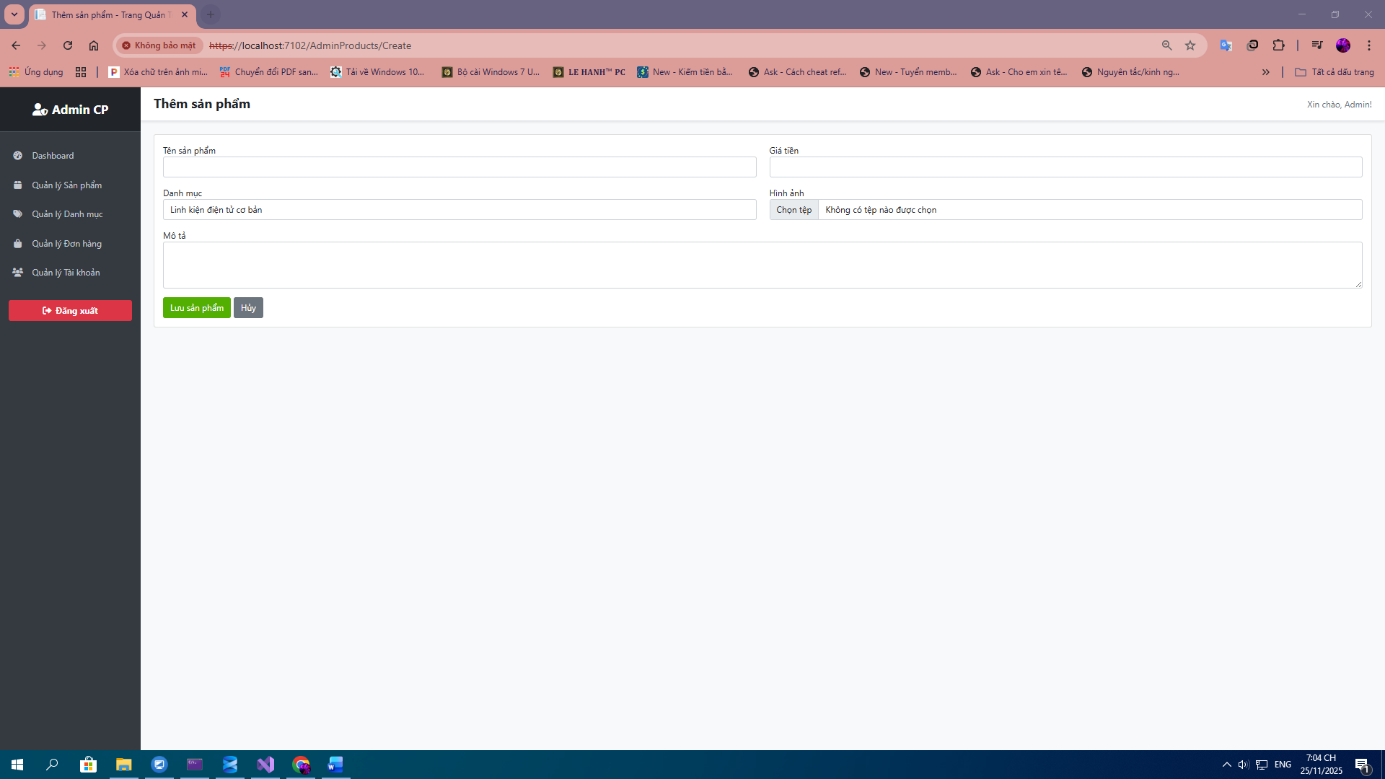


#### 4.2.14 Giao diện thêm mới danh mục

Hình 25 Giao diện thêm mới danh mục



#### 4.2.15 Giao diện thêm mới sản phẩm



Hình 26 Giao diện thêm mới sản phẩm

# CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 5.1 Các kết quả đạt được

- Hoàn thành xây dựng cơ sở dữ liệu cho hệ thống bán linh kiện điện tử.

- Xây dựng website với giao diện trực quan và các chức năng cơ bản: thêm, xóa, sửa, tra cứu, xuất dữ liệu.

- Áp dụng ràng buộc dữ liệu nhằm đảm bảo tính chính xác và đồng nhất.

- Đóng góp quy trình thiết kế và xây dựng hệ thống web thương mại điện tử ở mức cơ bản.

## 5.2 Hạn chế, hướng phát triển

### 5.2.1 Hạn chế:

- Một số tính năng bảo mật chưa được triển khai đầy đủ.

- Giao diện và trải nghiệm người dùng mới ở mức cơ bản.

### 5.2.2 Hướng phát triển:

- Tăng cường bảo mật cho hệ thống.

- Tối ưu và mở rộng cơ sở dữ liệu theo quy mô lớn.

- Tích hợp Web API để hỗ trợ kết nối đa nền tảng.

- Bổ sung tính năng nâng cao: Đánh giá sản phẩm, phản hồi khách hàng, thanh toán trực tuyến.

- Phát triển các nghiệp vụ hoàn chỉnh: Xử lý đơn hàng, vận chuyển và giao hàng.

# 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] Microsoft Docs. ASP.NET Core documentation:** <https://learn.microsoft.com/en-us/aspnet/core>

**[4] Microsoft Docs. Entity Framework Core documentation:** <https://learn.microsoft.com/en-us/ef/core/>

**[5] W3Schools. ASP.NET Tutorial:** <https://www.w3schools.com/aspnet/>

**[6] TutorialsPoint. ASP.NET Tutorial:** <https://www.tutorialspoint.com/asp.net/index.htm>

**[7] C# Corner. ASP.NET Core Tutorials:** <https://www.c-sharpcorner.com/technologies/asp-net-core>

**[8] DotNetCurry. ASP.NET Core Articles & Tutorials:**

<https://www.dotnetcurry.com/aspnet-core>

**[9] CodeProject. ASP.NET Example Projects:** <https://www.codeproject.com/>