

**TRƯỜNG ĐẠI HỌC YERSIN ĐÀ LẠT
KHOA CÔNG NGHỆ THÔNG TIN**



YERSIN UNIVERSITY

**BÁO CÁO MÔN HỌC
LẬP TRÌNH WEB 1**

**TÊN ĐỀ TÀI: XÂY DỰNG WEBSITE THƯƠNG MẠI ĐIỆN
TỬ KINH DOANH SẢN PHẨM QUẦN ÁO THỜI TRANG**

GVHD	:Nguyễn Đức Tấn
SVTH	:Bùi Quang Hoàng Anh
Mã số SV	:2301010054
Khóa học	:2024 - 2025

Đà Lạt, tháng 6 - 2025

PHẦN NHẬN XÉT CỦA GIẢNG VIÊN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Điểm:

Ngày ... tháng ... năm...

Ký và ghi rõ họ tên

MỤC LỤC

Chương 1 Tìm hiểu về lập trình Web MVC (Cơ sở lý thuyết)	7
1.1 Khái niệm	8
1.2 Đặc điểm	9
1.3 Nguyên lý hoạt động	10
1.4 Công nghệ triển khai ASP.NET Core	11
Chương 2 Xây dựng ứng dụng Web (Triển khai thực tế)	14
2.1 Phát biểu bài toán ứng dụng	14
2.2 Phân tích yêu cầu của ứng dụng	16
2.3 Thiết kế cơ sở dữ liệu	17
2.4 Thiết kế giao diện người dùng:	21
2.5 Thiết kế các thành phần MVC:	30
2.6 Triển khai và cài đặt	30
Chương 3 Kết quả chương trình	35
Kết luận	35
Tài liệu tham khảo	37
PHỤ LỤC	38

DANH MỤC HÌNH

Hình ảnh	Mô tả	Trang
Hình 1	Nguyên tắc hoạt động chung của MVC	8
Hình 2	Use-case Admin và User	16
Hình 3	Mô hình cơ sở dữ liệu	17
Hình 4	Product	20
Hình 5	Order	21
Hình 6	OrderDetail	22
Hình 7	ShoppingCartItem	22
Hình 8	Home	23
Hình 9	Home	23
Hình 10	Product	24
Hình 11	Detail	24
Hình 12	Order	25
Hình 13	CheckoutComplete	25
Hình 14	Invoice	26
Hình 15	HomeController	27
Hình 16	OrderController	28
Hình 17	Trang chủ	31
Hình 18	Đăng kí	31
Hình 19	Đăng nhập	32
Hình 20	Trang chủ shop	32
Hình 21	Thông tin chi tiết sản phẩm	33
Hình 22	Giỏ hàng	33

Hình 23	Mua hàng	34
Hình 24	Thông tin cá nhân mua hàng	34
Hình 25	Kiểm tra hóa đơn	34

BÁO CÁO TIẾN ĐỘ

Nội dung công việc	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	Hoàn thành
Tương ứng chương 1: Lý thuyết Sinh viên tìm hiểu và trình bày nội dung lý thuyết	X	X	X	X														<input checked="" type="checkbox"/>
Tổng hợp chương 1					X	X	X											<input checked="" type="checkbox"/>
Báo cáo lý thuyết								X										
Thiết kế giao diện									X	X	X	X	X	X	X			<input checked="" type="checkbox"/>
Xây dựng code backend ứng dụng Web													X	X				<input checked="" type="checkbox"/>
Thu thập hình ảnh và kết quả ứng dụng Web													X	X				<input checked="" type="checkbox"/>
Chuẩn bị nội dung slide thuyết trình													X			X		<input checked="" type="checkbox"/>
Thông nhất nội dung thuyết trình															X	X		<input checked="" type="checkbox"/>
Thuyết trình																	X	<input checked="" type="checkbox"/>
Tổng hợp và chỉnh sửa báo cáo																X		<input checked="" type="checkbox"/>
Báo cáo + Demo lab																	X	<input checked="" type="checkbox"/>

LỜI NÓI ĐẦU

Trong thời đại mà công nghệ bùng nổ như bây giờ, việc ai cũng có một phương tiện thông minh để quản lý gần như tất cả công việc trong cuộc sống của mình. Bên cạnh đó, sau đợt đại dịch Covid-19 diễn ra trong gần 3 năm liền, thói quen và lối sống của người tiêu dùng đã thay đổi khác trước, nên vì thế mà sự bùng nổ về thương mại điện tử là điều tất yếu xảy ra trong hoạt động kinh doanh hiện đại. Đặc biệt trong lĩnh vực thời trang - một ngành đòi hỏi sự cập nhật xu hướng liên tục và khả năng tiếp cận khách hàng nhanh chóng – việc xây dựng một website thương mại chuyên nghiệp đóng vai trò vô cùng quan trọng. Website không chỉ là kênh bán hàng trực tuyến hiệu quả mà còn là bộ mặt thương hiệu, giúp doanh nghiệp khẳng định vị thế trên thị trường. Xuất phát từ nhu cầu đó, em thực hiện đề tài "Xây dựng website thương mại kinh doanh quần áo thời trang" nhằm tạo ra một nền tảng trực tuyến thân thiện, dễ sử dụng và đáp ứng tốt nhu cầu mua sắm của người tiêu dùng hiện đại.

Website thương mại mang đến cho người dùng một nền tảng bán hàng trực tuyến tiện lợi, dễ sử dụng, giao diện hiện đại và khả năng quản lý linh hoạt. Hệ thống được xây dựng dựa trên mô hình kiến trúc phần mềm MVC (Model – View – Controller). Trong quá trình phát triển hệ thống, em đã sử dụng công cụ **Visual Studio** làm môi trường lập trình chính. Visual Studio không chỉ hỗ trợ tốt việc phát triển ứng dụng trên nền tảng **ASP.NET Core** – một framework mạnh mẽ và tối ưu của Microsoft – mà còn tích hợp nhiều tính năng hỗ trợ debug, quản lý cơ sở dữ liệu và triển khai hệ thống dễ dàng. Bên cạnh đó, để đảm bảo trải nghiệm người dùng được tối ưu về mặt thiết kế giao diện, em sử dụng công cụ **Figma** để thiết kế các bản mockup và wireframe của website trước khi tiến hành lập trình. Figma hỗ trợ làm việc nhóm trực tuyến, giúp em điều chỉnh thiết kế và đảm bảo tính nhất quán trong toàn bộ hệ thống.

Cuối cùng, em xin được bày tỏ lòng biết ơn sâu sắc đến thầy Nguyễn Đức Tấn – người đã tận tình hướng dẫn, hỗ trợ và truyền cảm hứng cho em trong suốt quá trình thực hiện đề tài. Sự đồng hành của thầy là động lực to lớn giúp em hoàn thành tốt đồ án này.

Em xin chân thành cảm ơn!

1. Chương 1 Tìm hiểu về lập trình Web MVC (Cơ sở lý thuyết)

1.1 Khái niệm

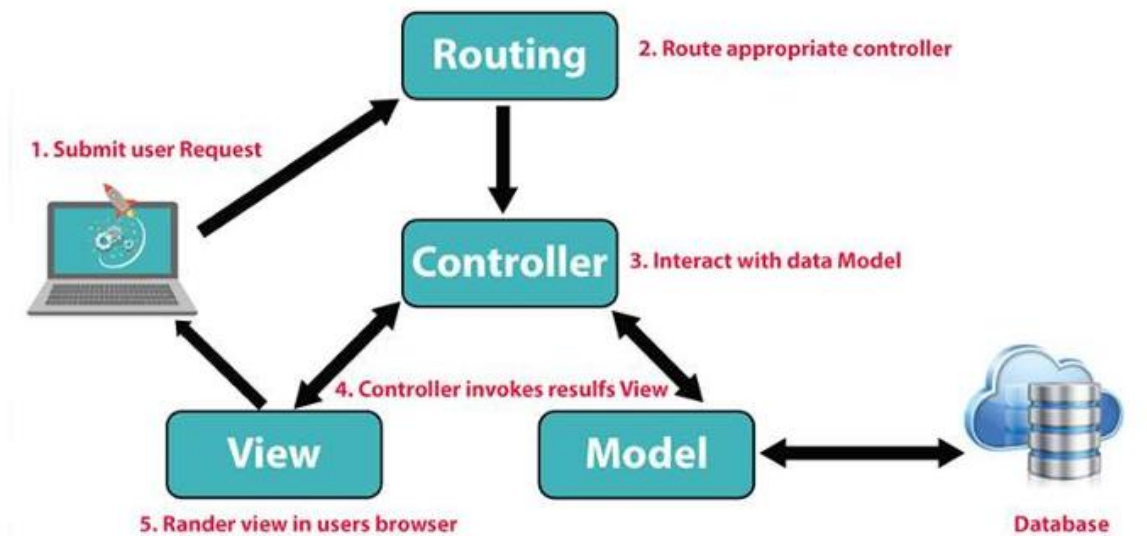
MVC là gì?

MVC là viết tắt của Model-View-Controller. Cấu trúc Model-View-Controller (MVC) là một mẫu kiến trúc/ thiết kế (design pattern) tách ứng dụng thành ba phần thành phần logic chính: Model, View và Controller. Mỗi thành phần kiến trúc được xây dựng để xử lý các khía cạnh phát triển cụ thể của một ứng dụng:

- **Model (Dữ liệu và logic nghiệp vụ):** Là nơi định nghĩa các lớp đối tượng, mối quan hệ và các thao tác liên quan đến dữ liệu (truy vấn, cập nhật, xóa, tính toán...). Model không phụ thuộc vào View hay Controller, giúp dữ liệu có thể được tái sử dụng trong nhiều bối cảnh khác nhau mà không cần viết lại.
- **View (Giao diện người dùng):** Là phần chịu trách nhiệm hiển thị thông tin cho người dùng và tiếp nhận các thao tác từ họ. View không chứa logic nghiệp vụ mà chỉ tập trung vào hiển thị dữ liệu được cung cấp bởi Controller hoặc Model. Điều này giúp nhà thiết kế giao diện có thể làm việc độc lập, cải tiến UI/UX mà không lo ảnh hưởng đến logic hệ thống.
- **Controller (Xử lý nghiệp vụ):** Là cầu nối giữa View và Model. Controller tiếp nhận các yêu cầu từ người dùng thông qua View (ví dụ: bấm nút, điền form...), xử lý logic điều hướng, tương tác với dữ liệu thông qua Model, sau đó cập nhật kết quả hiển thị lên View. Việc xử lý logic được tập trung tại đây giúp code trở nên rõ ràng, dễ kiểm soát và bảo trì.

→ Mô hình này giúp tách biệt rõ ràng giữa dữ liệu, giao diện và điều khiển luồng xử lý, từ đó dễ dàng bảo trì, mở rộng và phát triển ứng dụng.

Tóm lại, MVC là một mẫu kiến trúc tách ứng dụng thành ba thành phần chính: Model, View và Controller. Sự tách biệt này mang lại một số lợi ích, bao gồm cải thiện tổ chức code, khả năng bảo trì và khả năng mở rộng. MVC rất linh hoạt, có thể áp dụng trên các nền tảng web, máy tính để bàn và thiết bị di động.



Hình 1. Nguyên tắc hoạt động chung của MVC

1.2 Đặc điểm

Mô hình lập trình MVC không chỉ là một kiến trúc phổ biến mà còn mang lại nhiều lợi ích thiết thực trong quá trình phát triển và vận hành hệ thống web. Một trong những ưu điểm nổi bật của MVC là **khả năng hỗ trợ phát triển theo nhóm**. Việc chia hệ thống thành ba phần riêng biệt – Model, View và Controller – cho phép các thành viên trong nhóm có thể làm việc song song mà không bị phụ thuộc lẫn nhau. Ví dụ, người thiết kế giao diện có thể làm việc với phần View mà không cần quan tâm đến việc xử lý dữ liệu trong Model, trong khi lập trình viên backend có thể tập trung vào logic trong Controller và Model. Nhờ đó, tiến độ phát triển được cải thiện rõ rệt, đồng thời giảm thiểu xung đột trong quá trình làm việc nhóm.

Một điểm mạnh khác của MVC là **khả năng bảo trì và mở rộng hệ thống một cách dễ dàng**. Trong quá trình phát triển dài hạn, việc thêm mới tính năng hoặc cập nhật chức năng là điều không thể tránh khỏi. Với MVC, mỗi phần được tách riêng và quản lý độc lập, nên khi cần sửa đổi, em chỉ cần tập trung chỉnh sửa đúng nơi cần thiết mà không ảnh hưởng đến các phần còn lại. Điều này không những giúp tiết kiệm thời gian mà còn làm giảm nguy cơ gây lỗi cho toàn bộ hệ thống.

Tính tái sử dụng cao cũng là một lợi ích đáng kể của mô hình này. Các thành phần như Model có thể sử dụng lại cho nhiều View khác nhau. Ví dụ, một chức năng lấy thông tin sản phẩm trong Model có thể được hiển thị ở nhiều trang khác nhau như danh sách sản phẩm, trang chi tiết, hay trang khuyến mãi mà không cần viết lại mã. Việc tái sử dụng như vậy giúp giảm khối lượng công việc và tăng hiệu quả kiểm thử.

Bên cạnh đó, MVC còn giúp **phân tách rõ ràng vai trò giữa các phần trong hệ thống**. Nhờ vậy, lập trình viên dễ dàng nắm bắt chức năng của từng phần, giảm tình trạng viết mã lẫn lộn hoặc khó kiểm soát. Sự rõ ràng này cũng giúp em dễ dàng kiểm soát lỗi, xác định phạm vi xử lý, và cải thiện chất lượng mã nguồn tổng thể.

Cuối cùng, một ưu điểm quan trọng nhưng thường ít được chú ý là **khả năng hỗ trợ SEO (Search Engine Optimization)** tốt của MVC. Do MVC cho phép kiểm soát URL theo cấu trúc rõ ràng và dễ tùy chỉnh, các trang web được xây dựng bằng MVC thường có địa chỉ thân thiện với công cụ tìm kiếm như Google, Bing,... Điều này giúp nâng cao khả năng hiển thị và tiếp cận của website, đặc biệt hữu ích với các dự án thương mại điện tử như website bán quần áo thời trang – nơi việc tiếp cận khách hàng qua tìm kiếm là rất quan trọng.

Tổng thể, với những ưu điểm như dễ mở rộng, dễ bảo trì, hỗ trợ làm việc nhóm tốt, thân thiện với SEO và cấu trúc rõ ràng, mô hình MVC là lựa chọn phù hợp cho cả các dự án nhỏ lẫn những hệ thống lớn cần phát triển lâu dài.

1.3 Nguyên lý hoạt động

- Nguyên tắc hoạt động của mô hình MVC được chia thành các bước rõ ràng nhằm tách biệt các thành phần chức năng, giúp dễ dàng quản lý, mở rộng và bảo trì hệ thống. Quá trình bắt đầu khi người dùng thực hiện một hành động trên trình duyệt, chẳng hạn như nhấp chuột, gửi biểu mẫu hoặc truy cập một liên kết; khi đó, một yêu cầu (request) sẽ được gửi đến máy chủ. Thành phần Routing của hệ thống tiếp nhận yêu cầu này và tiến hành định tuyến đến Controller phù hợp dựa trên URL đã được cấu hình sẵn.
- Sau khi Controller nhận được yêu cầu, nó sẽ xử lý logic liên quan và nếu cần thiết, sẽ tương tác với Model để truy xuất hoặc cập nhật dữ liệu từ cơ sở dữ liệu. Model là nơi xử lý dữ liệu và thực hiện các nghiệp vụ liên quan đến thông tin trong hệ thống. Khi đã có dữ liệu cần thiết, Controller sẽ lựa chọn View tương ứng và truyền dữ liệu này sang View. View sẽ chịu trách nhiệm hiển thị thông tin ra giao diện người dùng một cách trực quan và dễ sử dụng. Cuối cùng, nội dung của View được render (kết xuất) thành mã HTML và trả về trình duyệt để hiển thị cho người dùng.
- Với quy trình hoạt động này, mô hình MVC không chỉ giúp phân chia rõ ràng trách nhiệm giữa các thành phần mà còn tăng khả năng tái sử dụng mã nguồn, cải thiện hiệu suất phát triển và đảm bảo tính nhất quán trong toàn bộ hệ thống. Đây là một trong những lý do mà em lựa chọn mô hình MVC để xây dựng website thương mại kinh doanh quần áo thời trang.

1.4 Công nghệ triển khai – Giới thiệu về ASP.NET Core

Trong quá trình xây dựng website thương mại điện tử kinh doanh quần áo thời trang, em đã lựa chọn **ASP.NET Core** làm công nghệ chính để phát triển hệ thống. ASP.NET Core là một nền tảng lập trình do Microsoft phát triển, được sử dụng để xây dựng các ứng dụng web hiện đại. Đây là phiên bản cải tiến từ công nghệ ASP.NET trước đây, được thiết kế lại để hoạt động nhanh hơn, nhẹ hơn, dễ sử dụng hơn và có thể chạy trên nhiều hệ điều hành khác nhau như Windows, Linux và macOS.

Lý do em chọn ASP.NET Core là vì nó hỗ trợ rất tốt cho mô hình MVC (Model – View – Controller). Với mô hình này, ứng dụng sẽ được chia thành ba phần riêng biệt. Phần **Model** dùng để xử lý dữ liệu và làm việc với cơ sở dữ liệu. Phần **View** hiển thị giao diện người dùng. Còn **Controller** là phần trung gian, nhận yêu cầu từ người dùng, xử lý và trả kết quả về cho View. Việc tách rõ các phần như vậy giúp cho hệ thống dễ phát triển, dễ bảo trì và cũng giúp em quản lý mã nguồn một cách khoa học hơn.

ASP.NET Core có thể chạy trên nhiều hệ điều hành khác nhau như **Windows, Linux và macOS**, cho phép lập trình viên lựa chọn môi trường phát triển và triển khai phù hợp với hệ thống thực tế. Với kiến trúc hiện đại, ASP.NET Core loại bỏ những ràng buộc cũ của ASP.NET, cung cấp hiệu năng vượt trội và khả năng tích hợp mạnh mẽ với các công nghệ mới như **Docker, Kubernetes**, hay các nền tảng điện toán đám mây như **Azure, AWS**.

Em sử dụng ASP.NET Core kết hợp với phần mềm **Visual Studio**, là một môi trường lập trình mạnh mẽ, giúp lập trình, chạy thử, kiểm tra và quản lý toàn bộ dự án một cách thuận tiện. Sự kết hợp giữa ASP.NET Core và Visual Studio giúp em dễ dàng xây dựng được một website hoàn chỉnh, từ giao diện đến chức năng, đáp ứng nhu cầu mua sắm online cho người dùng.

Tóm lại, ASP.NET Core là một công nghệ phù hợp với xu hướng hiện nay, mang lại nhiều lợi ích trong quá trình phát triển website. Nhờ vào khả năng linh hoạt, hiệu suất cao và dễ sử dụng, ASP.NET Core đã giúp em triển khai hiệu quả hệ thống website bán hàng quần áo thời trang.

1.5. Đặc điểm nổi bật, hiệu năng, điểm mạnh – điểm yếu, khả năng hỗ trợ và bảo mật của ASP.NET Core và Visual Studio

Trong quá trình xây dựng website thương mại điện tử bán quần áo thời trang, em sử dụng **ASP.NET Core** kết hợp với **Visual Studio** làm công nghệ và công cụ phát triển chính. Qua quá trình tìm hiểu và thực hiện, em nhận thấy hai công cụ này có nhiều đặc điểm nổi bật, hiệu suất cao, cùng với khả năng hỗ trợ lập trình viên rất tốt.

Đặc điểm nổi bật của ASP.NET Core:

ASP.NET Core là một nền tảng lập trình mã nguồn mở, hỗ trợ đa nền tảng (Windows, Linux, macOS), có thể dùng để xây dựng các ứng dụng web hiện đại, RESTful API, và cả ứng dụng thời gian thực. Nó hỗ trợ mô hình kiến trúc **MVC**, giúp tách biệt rõ ràng giữa dữ liệu, giao diện và điều khiển. ASP.NET Core còn cho phép cấu hình hệ thống linh hoạt, dễ tích hợp các dịch vụ như cơ sở dữ liệu, xác thực người dùng, ghi log, gửi email, v.v.

Khả năng hỗ trợ lập trình viên:

ASP.NET Core đi kèm với **Visual Studio**, một môi trường lập trình rất mạnh mẽ, giúp em dễ dàng viết mã, tự động gợi ý cú pháp, kiểm tra lỗi, debug, quản lý các thư viện cài thêm (NuGet) và kết nối cơ sở dữ liệu. Visual Studio cũng hỗ trợ giao diện kéo thả, quản lý source code và chạy thử ứng dụng ngay trên trình duyệt mà không cần cấu hình phức tạp.

Tóm lại, ASP.NET Core kết hợp với Visual Studio là một giải pháp rất phù hợp để xây dựng các website thương mại điện tử. Hai công cụ này không chỉ có hiệu năng cao, hỗ trợ đầy đủ tính năng bảo mật và lập trình, mà còn giúp em dễ dàng tổ chức mã nguồn, kiểm soát luồng dữ liệu và xây dựng giao diện người dùng một cách khoa học, chuyên nghiệp.

Ví dụ minh họa về ASP.NET Core:

1.5.1. Model

```
2. namespace AnhBui.Models
3. {
4.     public class Product
5.     {
6.         public int Id { get; set; }
7.         public string? Name { get; set; }
8.         public string? Detail { get; set; }
9.         public string? ImageUrl { get; set; }
10.        public decimal Price { get; set; }
11.        public bool IsTrendingProduct { get; set; }
12.    }
13. }
```

1.5.2. View

@model IEnumerable<Product>

```
< <section data-bs-version="5.1" class="header18 cid-tsEQyIZRcx mbr-fullscreen mbr-parallax-background"
    id="header18-1">
    <div class="mbr-overlay"
        style="opacity:0.9; background-image:
url('https://lh5.googleusercontent.com/n4Pb9SE_ilmNDIZxosZGNDHPhBsw-zY9eivqAVb0aEj2NIgfZJuLi-
AAzNrn29rmWNkvc2G7fwIfwd-nMncRRHHKUyFRimzpIf8N7a4oOV2dU7LtpxCXT5C7A-
DcK7UP1WTOjwSeSfnlCmpmcCSmVt6mxN8pa1v70GfPmQbJj4q1DIETMwz8jvzuTmPGtZl');">
    </div>
    <div class="mbr-overlay" style="opacity: 0.7; background-color: rgb(0, 0, 0);"></div>
    <div class="align-center container">
        <div class="row justify-content-center">
```

```

<div class="col-12 col-lg-10">
  <h1 class="mbr-section-title mbr-fonts-style mbr-white mb-3 display-1">
    <strong>
      AnhBui Fashion Shop
    </strong>
  </h1>

  <p class="mbr-text mbr-fonts-style mbr-white display-7">
    Style của bạn, dấu ấn của riêng bạn </p>

```

1.5.3. Controller

namespace AnhBui.Controllers

```

{

  public class HomeController : Controller
  {

    private readonly IProductRepository _productRepository;

    public HomeController(IProductRepository productRepository)
    {
      _productRepository = productRepository;
    }

    public IActionResult Index()
    {
      var trendingProducts = _productRepository.GetTrendingProducts();

      return View(trendingProducts);
    }
  }

```

Chương 2 Xây dựng ứng dụng Web (Triển khai thực tế)

2.1 Phát biểu bài toán ứng dụng

2.1.1. Bài toán đặt ra trước mắt

Trong bối cảnh nhu cầu mua sắm trực tuyến ngày càng tăng, đặc biệt là trong lĩnh vực thời trang, các cửa hàng quần áo truyền thống đang dần chuyển dịch sang mô hình thương mại điện tử để tiếp cận khách hàng nhanh hơn, hiệu quả hơn. Tuy nhiên, việc xây dựng một hệ thống website bán hàng không chỉ dừng lại ở việc hiển thị sản phẩm, mà còn đòi hỏi nhiều yếu tố kỹ thuật và trải nghiệm người dùng để đảm bảo hoạt động ổn định, dễ sử dụng và tiện lợi trong quá trình mua sắm.

Bài toán đặt ra là: **làm thế nào để xây dựng một website thương mại điện tử chuyên nghiệp dành cho cửa hàng quần áo**, đáp ứng đầy đủ các chức năng cơ bản như:

- Hiển thị danh mục sản phẩm theo từng loại (áo, quần, váy, phụ kiện...)
- Xem chi tiết sản phẩm, hình ảnh, giá cả và mô tả
- Cho phép người dùng thêm sản phẩm vào giỏ hàng và tiến hành thanh toán
- Quản lý đơn hàng và trạng thái giao hàng
- Hỗ trợ người quản trị dễ dàng thêm, sửa, xoá sản phẩm
- Hệ thống đăng nhập/đăng ký cho khách hàng và phân quyền cho quản trị viên
- Thiết kế giao diện hiện đại, dễ sử dụng trên cả máy tính và điện thoại

Vì vậy, em cần lựa chọn công nghệ lập trình phù hợp, mô hình kiến trúc hợp lý và công cụ hỗ trợ phát triển hiệu quả để hiện thực hóa một hệ thống website vừa thân thiện với người dùng, vừa dễ quản lý cho người bán hàng.

2.1.2. Mục tiêu mà đề tài của em muốn hướng tới

Mục tiêu của đề tài là xây dựng một website thương mại điện tử bán quần áo thời trang với các chức năng cơ bản, giao diện thân thiện và dễ sử dụng. Cụ thể, em hướng đến:

- Hiển thị sản phẩm theo danh mục, xem chi tiết sản phẩm
- Thêm sản phẩm vào giỏ hàng và tiến hành đặt hàng
- Đăng ký, đăng nhập và quản lý tài khoản người dùng
- Quản trị hệ thống: thêm/sửa/xoá sản phẩm, quản lý đơn hàng
- Áp dụng mô hình lập trình MVC để tổ chức mã rõ ràng, dễ mở rộng
- Đảm bảo bảo mật thông tin người dùng và hoạt động ổn định
- Tối ưu website cho công cụ tìm kiếm (SEO)

Thông qua đề tài, em mong muốn vận dụng kiến thức lập trình web để xây dựng một hệ thống hoàn chỉnh, phục vụ thực tế.

2.1.3. Vai trò và giá trị mang lại

Việc xây dựng website thương mại điện tử bán quần áo thời trang mang lại nhiều vai trò và giá trị thiết thực trong học tập cũng như thực tiễn. Cụ thể:

- **Ứng dụng thực tế cao:** Hệ thống có thể triển khai cho các cửa hàng thời trang thật, hỗ trợ hoạt động bán hàng online hiệu quả.
- **Tiếp cận công nghệ hiện đại:** Giúp em làm quen với các công nghệ phổ biến như ASP.NET Core, MVC, cơ sở dữ liệu SQL Server, giao diện Figma,...
- **Rèn luyện kỹ năng lập trình web:** Từ thiết kế giao diện đến xử lý logic và quản lý dữ liệu.
- **Phát triển tư duy hệ thống:** Biết cách phân tích bài toán, chia nhỏ chức năng và kết hợp nhiều công nghệ trong cùng một dự án.
- **Củng cố kiến thức chuyên ngành:** Giúp em áp dụng lý thuyết vào thực hành, chuẩn bị cho công việc thực tế sau này.
- **Thúc đẩy tư duy tự học và nghiên cứu:** Quá trình triển khai giúp em học thêm nhiều kỹ năng mới, không chỉ dựa vào bài giảng.

Đối tượng sử dụng:

Hệ thống website được xây dựng hướng đến các đối tượng chính sau:

- **Khách hàng mua sắm online:** Có thể xem sản phẩm, đặt hàng, theo dõi đơn hàng và quản lý tài khoản cá nhân.
- **Chủ cửa hàng/quản trị viên:** Quản lý sản phẩm, đơn hàng, khách hàng và theo dõi hoạt động kinh doanh.
- **Sinh viên và lập trình viên:** Có thể tham khảo, học hỏi cách xây dựng hệ thống thương mại điện tử sử dụng mô hình MVC.

Phạm vi áp dụng:

- Áp dụng trong **mô hình cửa hàng quần áo vừa và nhỏ**, hoặc các cửa hàng mới bắt đầu chuyển sang bán hàng online.
- Có thể mở rộng thêm để phù hợp với các lĩnh vực khác như bán giày dép, phụ kiện, mỹ phẩm,...

Hệ thống triển khai thử nghiệm ở môi trường nội bộ hoặc cục bộ

2.2 Phân tích yêu cầu của ứng dụng

2.2.1. Mô tả chức năng ứng dụng

Chức năng dành cho người quản trị (Admin)

Quản lý danh mục tài liệu:

- Thêm, sửa, xóa các danh mục như: Tên hàng hóa (quần áo), thay đổi giá, thay đổi tên/thể loại trang phục
- Phân loại quần áo thời trang: Giày, dép, áo, quần, nón, váy,...
- Thêm mới, chỉnh sửa, cập nhật mô tả, giá bán và trạng thái hiển thị của hàng hóa
- Theo dõi danh sách người dùng, phân quyền, mở/khóa tài khoản

Chức năng dành cho người dùng (User)

Hiển thị giao diện với người dùng:

- Trang chủ: Hiển thị các sản phẩm nổi bật, danh mục và thông tin giới thiệu
- Danh mục sản phẩm: Lọc và hiển thị sản phẩm theo loại (áo, quần, váy, phụ kiện,...)
- Trang chi tiết sản phẩm: Hiển thị ảnh, mô tả, giá, size và nút thêm vào giỏ hàng
- Đăng kí/Đăng nhập: Người dùng tạo tài khoản, đăng nhập để đặt hàng và quản lý thông tin
- Quản lý đơn hàng: Người dùng theo dõi tình trạng đơn hàng

2.2.2 Yêu cầu phi chức năng

Ngoài các yêu cầu chức năng, hệ thống cần đáp ứng một số yêu cầu phi chức năng quan trọng để đảm bảo hiệu năng, độ tin cậy và khả năng mở rộng lâu dài. Những yêu cầu này được chia thành các nhóm cụ thể như sau:

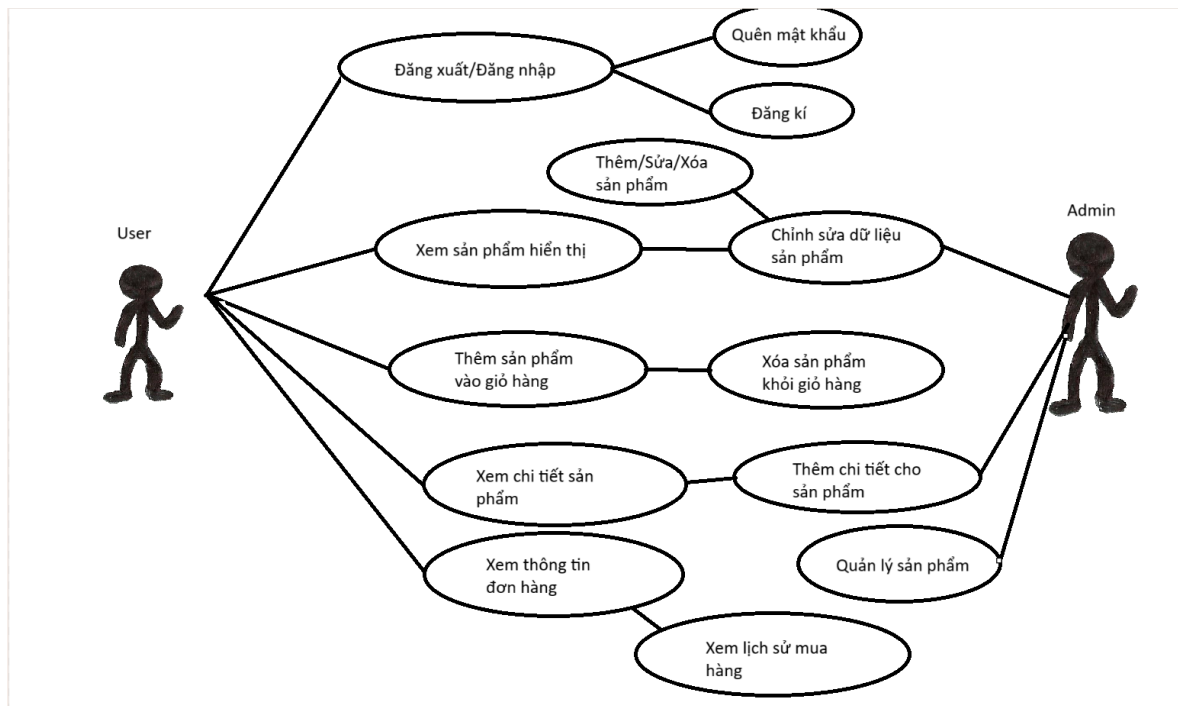
2.2.3. Hiệu suất (Performance) Ứng dụng cần đảm bảo tốc độ phản hồi nhanh, thời gian tải toàn bộ nội dung trên mỗi trang không vượt quá 2 giây

trong điều kiện mạng ổn định. Giao dịch thanh toán phải được xử lý nhanh chóng, toàn bộ quá trình từ khi người dùng nhấn “thanh toán” đến khi nhận được xác nhận không vượt quá 5 giây. Các hình ảnh tài liệu và ảnh xem trước cần được nén và tối ưu hóa để tránh gây chậm trễ trong việc hiển thị và tải trang.

2.2.4. Khả năng sử dụng Giao diện hệ thống cần được thiết kế trực quan, thân thiện, dễ thao tác đối với mọi đối tượng người dùng, kể cả người không rành về công nghệ. Website phải tương thích với các trình duyệt phổ biến hiện nay như Google Chrome, Firefox, Microsoft Edge, Safari,... Hệ thống cần hỗ trợ hiển thị tốt trên nhiều thiết bị như máy tính để bàn, máy tính xách tay, máy tính bảng và điện thoại thông minh.

2.2.5. Bảo mật Hệ thống sử dụng ASP.NET Core Identity để quản lý người dùng, thực hiện xác thực, phân quyền truy cập. Mật khẩu người dùng phải được mã hóa bằng thuật toán Hash hiện đại, không lưu trữ dưới dạng văn bản thuần.

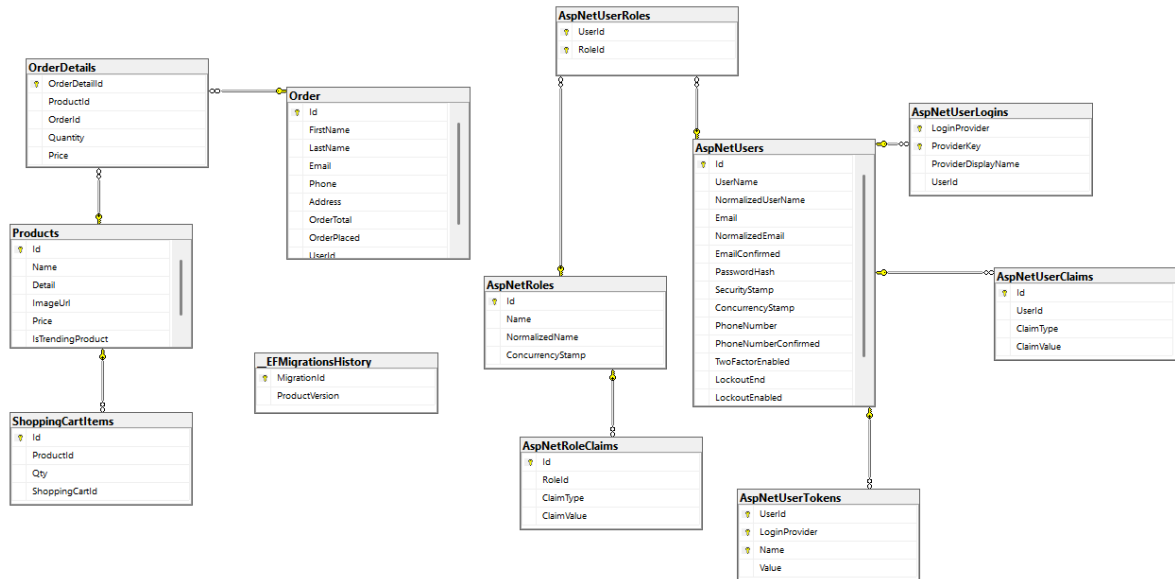
2.2.7. Biểu đồ Use-Case



Hình 2: Use-case Admin và User

2.3 Thiết kế cơ sở dữ liệu

Mô hình vật lý



Hình 3: Mô hình cơ sở dữ liệu

1. AspNetUsers

Đây là bảng trung tâm để lưu trữ thông tin người dùng.

- **Id (PK - Khóa chính):** Kiểu dữ liệu string (thường là GUID), là định danh duy nhất cho mỗi người dùng.
- **UserName:** Tên người dùng mà họ sử dụng để đăng nhập.
- **NormalizedUserName:** Phiên bản chuẩn hóa của UserName (thường là chữ hoa) để tối ưu hóa việc tìm kiếm.
- **Email:** Địa chỉ email của người dùng.
- **NormalizedEmail:** Phiên bản chuẩn hóa của Email (thường là chữ hoa) để tối ưu hóa việc tìm kiếm.
- **EmailConfirmed:** Một giá trị boolean cho biết địa chỉ email của người dùng đã được xác nhận hay chưa.
- **PasswordHash:** Hàm băm của mật khẩu người dùng. Đây là nơi lưu trữ mật khẩu đã được mã hóa, không phải mật khẩu gốc.
- **SecurityStamp:** Một giá trị được cập nhật khi thông tin bảo mật của người dùng thay đổi (ví dụ: đổi mật khẩu, thêm xác thực hai yếu tố), giúp vô hiệu hóa các cookie xác thực cũ.
- **ConcurrencyStamp:** Một giá trị được sử dụng để xử lý đồng thời, giúp ngăn chặn các vấn đề khi nhiều người dùng cố gắng cập nhật cùng một bản ghi.
- **PhoneNumber:** Số điện thoại của người dùng.
- **PhoneNumberConfirmed:** Một giá trị boolean cho biết số điện thoại của người dùng đã được xác nhận hay chưa.
- **TwoFactorEnabled:** Một giá trị boolean cho biết xác thực hai yếu tố có được bật cho người dùng này hay không.

- **LockoutEnd:** Thời điểm người dùng bị khóa đăng nhập (nếu có). Thường là một giá trị DateTimeOffset.
- **LockoutEnabled:** Một giá trị boolean cho biết liệu tài khoản người dùng có thể bị khóa hay không (ví dụ: sau nhiều lần đăng nhập sai).
- **AccessFailedCount:** Số lần đăng nhập sai liên tiếp của người dùng.

2. AspNetRoles

Bảng này lưu trữ các vai trò (roles) trong hệ thống, được sử dụng để phân quyền cho người dùng.

- **Id (PK - Khóa chính):** Kiểu dữ liệu string (thường là GUID), là định danh duy nhất cho mỗi vai trò.
- **Name:** Tên của vai trò (ví dụ: "Admin", "User", "Manager").
- **NormalizedName:** Phiên bản chuẩn hóa của Name (thường là chữ hoa) để tối ưu hóa việc tìm kiếm.
- **ConcurrencyStamp:** Một giá trị được sử dụng để xử lý đồng thời, tương tự như trong AspNetUsers.

3. AspNetUserRoles

Đây là bảng kết nối giữa AspNetUsers và AspNetRoles, tạo ra mối quan hệ nhiều-nhiều (many-to-many). Mỗi bản ghi trong bảng này cho biết một người dùng thuộc vai trò nào.

- **UserId (PK, FK - Khóa chính, Khóa ngoại):** Tham chiếu đến Id của bảng AspNetUsers.
- **RoleId (PK, FK - Khóa chính, Khóa ngoại):** Tham chiếu đến Id của bảng AspNetRoles.

4. AspNetUserLogins

Bảng này lưu trữ thông tin đăng nhập từ các nhà cung cấp bên ngoài (ví dụ: Google, Facebook, Microsoft Account).

- **LoginProvider (PK):** Tên của nhà cung cấp đăng nhập (ví dụ: "Google", "Facebook").
- **ProviderKey (PK):** Định danh duy nhất của người dùng từ nhà cung cấp đăng nhập đó.
- **ProviderDisplayName:** Tên hiển thị của nhà cung cấp đăng nhập (thường là tên thân thiện hơn cho người dùng).
- **UserId (FK - Khóa ngoại):** Tham chiếu đến Id của bảng AspNetUsers, liên kết thông tin đăng nhập bên ngoài với tài khoản người dùng nội bộ.

5. AspNetUserClaims

Bảng này lưu trữ các "claims" (yêu cầu/khẳng định) của người dùng. Claims là các cặp khóa-giá trị đại diện cho một thuộc tính của người dùng (ví dụ: "DateOfBirth", "City", "EmployeeId").

- **Id (PK - Khóa chính):** Kiểu dữ liệu int (hoặc tương tự), là định danh duy nhất cho mỗi claim.
- **UserId (FK - Khóa ngoại):** Tham chiếu đến Id của bảng AspNetUsers.
- **ClaimType:** Loại của claim (ví dụ: "<http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress>", "CustomClaim:EmployeeId").
- **ClaimValue:** Giá trị của claim (ví dụ: "john.doe@example.com", "12345").

6. AspNetRoleClaims

Bảng này tương tự như AspNetUserClaims nhưng dành cho các claim liên quan đến vai trò.

- **Id (PK - Khóa chính):** Kiểu dữ liệu int (hoặc tương tự), là định danh duy nhất cho mỗi claim.
- **RoleId (FK - Khóa ngoại):** Tham chiếu đến Id của bảng AspNetRoles.
- **ClaimType:** Loại của claim.
- **ClaimValue:** Giá trị của claim.

7. AspNetUserTokens

Bảng này lưu trữ các token liên quan đến người dùng, thường được sử dụng cho các mục đích như reset mật khẩu, xác nhận email, hoặc lưu trữ token OAuth/OpenID Connect.

- **UserId (PK, FK - Khóa chính, Khóa ngoại):** Tham chiếu đến Id của bảng AspNetUsers.
- **LoginProvider (PK):** Tên của nhà cung cấp token (ví dụ: "Default", "Google").
- **Name (PK):** Tên của token (ví dụ: "AuthenticatorKey", "RefreshToken").
- **Value:** Giá trị của token.

Tóm tắt mối quan hệ:

- **AspNetUsers** là trung tâm, liên kết với tất cả các bảng Identity khác.
- **AspNetUserRoles** là bảng liên kết nhiều-nhiều giữa **AspNetUsers** và **AspNetRoles**.
- **AspNetUserLogins**, **AspNetUserClaims**, và **AspNetUserTokens** đều có mối quan hệ một-nhiều với **AspNetUsers** (một người dùng có thể có nhiều đăng nhập, nhiều claim, nhiều token).

- **AspNetRoleClaims** có mối quan hệ một-nhiều với **AspNetRoles** (một vai trò có thể có nhiều claim).

2.4 Thiết kế các thành phần MVC:

Model: Định nghĩa các lớp quản lý dữ liệu (ví dụ: lớp Baiviet với các thuộc tính id, ten).

- Tầng Controller đóng vai trò trung gian giữa người dùng và hệ thống. Đây là nơi tiếp nhận các yêu cầu từ phía client (trình duyệt), xử lý logic cần thiết, sau đó gọi tới tầng Model để thao tác dữ liệu, và cuối cùng trả về View tương ứng để hiển thị kết quả.
- Mỗi controller đại diện cho một nhóm chức năng riêng biệt trong hệ thống.
- Các phương thức bên trong controller (gọi là action) sẽ xử lý các thao tác như hiển thị danh sách, tạo mới, cập nhật hoặc xóa dữ liệu. Controller giúp đảm bảo quá trình luân chuyển dữ liệu diễn ra mạch lạc và rõ ràng giữa các tầng.

Class Product:

```

namespace AnhBui.Models
{
    26 references
    public class Product
    {
        18 references
        public int Id { get; set; }
        10 references
        public string? Name { get; set; }
        7 references
        public string? Detail { get; set; }
        9 references
        public string? ImageUrl { get; set; }
        11 references
        public decimal Price { get; set; }
        7 references
        public bool IsTrendingProduct { get; set; }
    }
}

```

Hình 4: Product

Class Order:

```

namespace AnhBui.Models
{
    public class Order
    {
        public int Id { get; set; }

        public string? FirstName { get; set; }

        public string? LastName { get; set; }

        public string? Email { get; set; }

        public string? Phone { get; set; }
        3 references
        public string? Address { get; set; }
        3 references
        public decimal OrderTotal { get; set; }
        3 references
        public DateTime OrderPlaced { get; set; }
        0 references
        public string? UserId { get; set; }
        3 references
        public List<OrderDetail>? OrderDetails { get; set; }
    }
}

```

Hình 5: Order

Class OrderDetail:

```

namespace AnhBui.Models
{
    4 references
    public class OrderDetail
    {
        0 references
        public int OrderDetailId { get; set; }
        1 reference
        public int ProductId { get; set; }
        1 reference
        public Product? Product { get; set; }
        0 references
        public int OrderId { get; set; }
        0 references
        public Order? Order { get; set; }
        3 references
        public int Quantity { get; set; }
        3 references
        public decimal Price { get; set; }
    }
}

```

Hình 6: OrderDetail

Class ShoppingCartItem:

```

using System.ComponentModel.DataAnnotations.Schema;

```

```

namespace AnhBui.Models
{
    public class ShoppingCartItem
    {
        0 references
        public int Id { get; set; }
        11 references
        public Product? Product { get; set; }
        9 references
        public int Qty { get; set; }
        6 references
        public string? ShoppingCartId { get; set; }
    }
}

```

Hình 7: ShoppingCartItem

View: - Tầng View chịu trách nhiệm trình bày dữ liệu ra giao diện người dùng. Nó được xây dựng bằng Razor (.cshtml) – một công cụ kết hợp giữa cú pháp HTML và C#, cho phép hiển thị dữ liệu động một cách linh hoạt. View không xử lý logic nghiệp vụ mà chỉ tập trung vào việc trình bày thông tin. Dữ liệu cần hiển thị sẽ được truyền từ Controller sang View thông qua Model hoặc ViewModel. Tầng View đóng vai trò quan trọng trong việc tạo ra trải nghiệm người dùng thân thiện, dễ sử dụng và phù hợp với chức năng hệ thống.

Home:

```
@model IEnumerable<Product>

<section data-bs-version="5.1" class="header18 cid-tsEQyIzRcx mbr-fullscreen mbr-parallax-background" id="header18-1">
  <div class="mbr-overlay" style="opacity:0.9; background-image: url('https://lh5.googleusercontent.com/nWp9SE_ilmNOL2xos2GNDhpH8se-zY9eivqAV0aEj2Nigf2JuLi-AAzHrn29zmHkvc2G7fa1fad-nMncRRH8UyFRImzpIf8N7a6oQV2du7LtpaCXT5C7A-DCK7'); background-color: #000000;"/>
  <div class="mbr-section-title mbr-fonts-style mbr-white mb-3 display-1">
    <strong>AnhBui Fashion Shop</strong>
  </div>
  <p class="mbr-text mbr-fonts-style mbr-white display-7">
    Style của bạn, dấu ấn của riêng bạn
  </p>
  <div class="mbr-section-btm mt-3">
    <a class="btn btn-primary display-4" asp-controller="Product" asp-action="Shop">
      <span class="mobi-mBri mobi-mBri-shopping-bag mbr-iconfont mbr-iconfont-btm"></span>Shop now</a>
  </div>
</section>
```

- Giao diện trang chủ của web, với màu nền trắng và vàng tạo nên cảm giác sang trọng, thanh lịch khi user ban đầu truy cập trang web

```
<section data-bs-version="5.1" class="content2 cid-tsEQ5Cqggh" id="content2-2">
  <div class="container">
    <div class="mbr-section-head">
      <h2 class="mbr-section-title mbr-fonts-style align-center mb-0 display-2">
        <strong>Top Selling</strong>
      </h2>
      <h3 class="mbr-section-subtitle mbr-fonts-style align-center mb-0 mt-2 display-5">
        Xu Hướng Dẫn Đầu
      </h3>
    </div>
    <div class="row mt-4">
      <div class="item features-image col-12 col-md-6 col-lg-4">
        <div class="item-wrapper">
          <div class="item-img">
            
          </div>
          <div class="item-content">
            <a asp-controller="Product" asp-action="Detail" asp-route-id="@item.Id" target="_self">
              <h4 class="item-subtitle mbr-fonts-style mt-1 display-7">
                <strong>@item.Name</strong>
              </h4>
              <p class="mbr-text mbr-fonts-style mt-3 display-7">@item.Price.ToString("C")</p>
            </a>
            <div class="mbr-section-btm item-footer mt-2">
              <a asp-controller="ShoppingCart" asp-action="AddToShoppingCart" asp-route-pID="@item.Id" class="btn item-btn btn-primary display-4" target="_self">
                <span class="mBriB-cart-add mbr-iconfont mbr-iconfont-btm"></span>Add to cart
              </a>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</section>
```

- Giao diện cho User nhìn thấy các sản phẩm đang vô cùng bán chạy của cửa hàng, nhờ đó thu hút được user, giúp họ nhìn nhận trang web rõ ràng

Product:


```

@model IEnumerable<AnhBui.Models.Product>

<section data-bs-version="5.1" class="info3 cid-tsEY2vwh5a mbr-parallax-background" id="info3-b">
    <div class="mbr-overlay" style="opacity:0.9; background-image: url('https://thing.vn/wp-content/uploads/2022/02/y-tuong-thiet-ke-shop-quan-ao-dep-7.jpg');">
    </div>
    <div class="mbr-overlay" style="opacity: 0.7; background-color: #000000;">
    <div class="container">
        <div class="row justify-content-center">
            <div class="card col-12 col-lg-10">
                <div class="card-wrapper">
                    <div class="card-box align-center">
                        <h4 class="card-title mbr-fonts-style align-center mb-4display-1">
                            <strong>Shop</strong>
                        </h4>
                    </div>
                </div>
            </div>
        </div>
    </div>
</section>

<section data-bs-version="5.1" class="content2 cid-tsEZVFGbrL" id="content2-g">
    <div class="container">
        <div class="mbr-section-head">
            <h5 class="mbr-section-subtitle mbr-fonts-style align-center mb-0 mt-2display-5">
                Không chỉ đẹp - Là khác biệt
            </h5>
            <partial name="_Product" model="@Model" />
        </div>
    </div>
</section>

```

- Đoạn này để admin thiết kế giao diện ban đầu của shop và đưa ra các sản phẩm của shop bên dưới

Detail

```

@model AnhBui.Models.Product
<br />
<div class="container">
    <div class="card-wrapper">
        <div class="row align-items-center">
            <div class="col-12 col-lg-6">
                <div class="image-wrapper">
                    
                </div>
            </div>
            <div class="col-12 col-lg">
                <div class="text-box">
                    <h5 class="mbr-title mbr-fonts-style display-2">
                        <strong>@Model.Name</strong>
                    </h5>
                    <p class="mbr-text mbr-fonts-style display-7">
                        @Model.Detail
                    </p>
                    <div class="cost">
                        <span class="currentcost mbr-fonts-style pr-2 display2">@Model.Price.ToString("c")</span>
                    </div>
                    <div class="mbr-section-btn pt-3">
                        <a asp-controller="ShoppingCart" asp-action="AddToShoppingCart" asp-route-id="@Model.Id" class="btn btn-primary display-4">
                            <span class="mbriconcart-add mbr-iconfont mbr-iconfontbtn"></span>Add to cart&nbsp;
                        </a>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
<br />

```

- Đoạn này cho user biết thông tin chi tiết sản phẩm cụ thể ra sao để user có thể xem xét lựa chọn mua

Order

```

@model Order

<form asp-action="Checkout" method="post" role="form">
  <section data-bs-version="5.1" class="form6 cid-tsFfsp7eqk" id="form6-n">
    <div class="container">
      <div class="mt-5">
        <div class="d-flex justify-content-between align-items-center mb-4 border-bottom pb-4">
          <h3 class="fs-5 fw-bolder m-0 lh-1">Shipping Address</h3>
        </div>
        <div class="row">
          <div class="col-sm-6">
            <div class="form-group">
              <label asp-for="FirstName" class="form-label">First name</label>
              <input asp-for="FirstName" type="text" class="form-control" placeholder="" value="" required="">
            </div>
          </div>
          <div class="col-sm-6">
            <div class="form-group">
              <label asp-for="LastName" class="form-label">Last name</label>
              <input asp-for="LastName" type="text" class="form-control" id="lastNameBilling" placeholder="" value="" required="">
            </div>
          </div>
          <div class="col-sm-6">
            <div class="form-group">
              <label asp-for="Email" class="form-label">Email</label>
              <input type="email" class="form-control" asp-for="Email" placeholder="">
            </div>
          </div>
          <div class="col-sm-6">
            <div class="form-group">
              <label asp-for="Phone" class="form-label">Phone</label>
              <input type="tel" class="form-control" asp-for="Phone" id="phone" placeholder="">
            </div>
          </div>
          <div class="col-12">
            <div class="form-group">
              <label asp-for="Address" class="form-label">Address</label>
              <input type="text" class="form-control" asp-for="Address" placeholder="123 Some Street Somewhere">
            </div>
          </div>
          <div class="col-auto mbr-section-btn align-center">
            <button type="submit" class="btn btn-primary display-4">Place Order</button>
          </div>
        </div>
      </div>
    </div>
  </section>
</form>

```

- Thông tin cá nhân của khách hàng sẽ được lưu ở đây nếu khách hàng có ý định đăng nhập, đăng kí

```

<div class="container">
  <br><br><br><br>
  <div class="text-center">
    
    <br>
    <h1><strong>Your order has been received</strong></h1>
    <br>
    <p>Thank you for your purchase !</p>
    <a class="btn btn-primary w-20" asp-controller="Product" asp-action="Shop">
      Start Shopping Again
    </a>
    <a class="btn btn-primary w-20" asp-controller="Orders" asp-action="List">Xem tất cả đơn hàng của tôi</a>
  </div>
</div>

```

- Khách hàng kiểm tra thông tin địa chỉ khi hoàn thành xác nhận đơn hàng

```

<h3>Chi tiết đơn hàng:</h3>
<table>
  <thead>
    <tr>
      <th>Sản phẩm</th>
      <th>Số lượng</th>
      <th>Giá mỗi sản phẩm</th>
      <th>Thành tiền</th>
    </tr>
  </thead>
  <tbody>
    @foreach (var item in Model.OrderDetails!)
    {
      <tr>
        <td>@item.Product?.Name</td>
        <td>@item.Quantity</td>
        <td>@item.Price.ToString("C")</td>
        <td>@((item.Quantity * item.Price).ToString("C"))</td>
      </tr>
    }
  </tbody>
</table>
</div>

<div class="invoice-total">
  <p>Tổng cộng: @Model.OrderTotal.ToString("C")</p>
</div>

<div class="text-center" style="margin-top: 30px;">
  <p>Cảm ơn bạn đã mua sắm tại AnhBui Shop!</p>
  <a class="btn btn-primary" asp-controller="Products" asp-action="Shop">Tiếp tục mua sắm</a>
</div>
</div>
</body>
</html>

```

- Chi tiết đơn hàng và lưu trong hóa đơn

Controller: Xác định các hành động (ví dụ: themTintuc, xoaTintuc).

```

namespace AnhBui.Controllers
{
    1 reference
    public class HomeController : Controller
    {
        private readonly IProductRepository _productRepository;

        0 references
        public HomeController(IProductRepository productRepository)
        {
            _productRepository = productRepository;
        }

        0 references
        public IActionResult Index()
        {
            var trendingProducts = _productRepository.GetTrendingProducts();
            return View(trendingProducts);
        }

        0 references
        public IActionResult Privacy()
        {
            return View();
        }

        0 references
        public IActionResult Error()
        {
            return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier });
        }
    }
}

```

- Người dùng từ trang chính dùng xem sản phẩm, xem thông tin bảo mật

```

public class OrdersController : Controller
{
    private IOrderRepository orderRepository;
    private IShoppingCartRepository shoppingCartRepository;
    private IEnumerable<Order> order;
    private readonly object dbContext;

    0 references
    public OrdersController(IOrderRepository orderRepository, IShoppingCartRepository shoppingCartRepository)
    {
        this.orderRepository = orderRepository;
        this.shoppingCartRepository = shoppingCartRepository;
    }

    0 references
    public IActionResult Checkout()
    {
        return View();
    }

    [HttpPost]
    0 references
    public IActionResult Checkout(Order order)
    {
        orderRepository.PlaceOrder(order);
        shoppingCartRepository.ClearCart();

        return RedirectToAction("CheckoutComplete");
    }

    0 references
    public IActionResult CheckoutComplete()
    {
        return View();
    }

    0 references
    public IActionResult Invoice(int orderId)
    {
        var order = orderRepository.GetOrderById(orderId);
        if (order == null)
        {
            return NotFound();
        }
        return View(order);
    }
}

```

- Người dùng kiểm tra được thông tin đơn hàng, đặt bởi ai, kiểm tra xem tình trạng đơn hàng

2.5 Triển khai và cài đặt

Cấu hình máy phát triển:

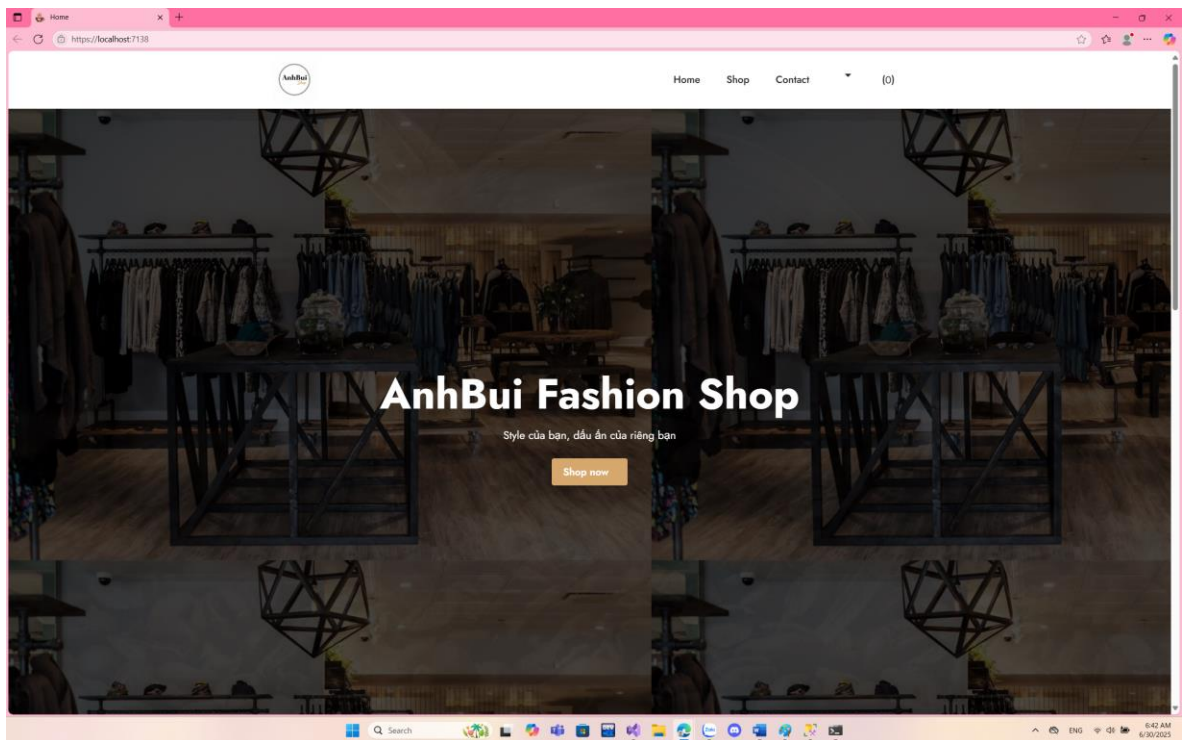
- Hệ điều hành: Windows 11 Pro 64-bit
- Bộ xử lý (CPU): Intel Core i5-12400
- Bộ nhớ RAM: 16 GB DDR4
- Card đồ họa (GPU): NVIDIA GeForce GTX 1650
- Công cụ phát triển: Visual Studio 2022 Community
- Trình duyệt kiểm thử: Google Chrome, Microsoft Edge

Môi trường phát triển:

- Ngôn ngữ lập trình: C#
- Kiến trúc ứng dụng: ASP.NET Core MVC
- Giao diện người dùng: Razor View Engine
- Quản lý cơ sở dữ liệu: Entity Framework Core (Code First)
- Hệ quản trị CSDL: Microsoft SQL Server LocalDB
- Xác thực & phân quyền: ASP.NET Core Identity
- Thiết kế giao diện: Bootstrap, Tailwind CSS
- Biểu tượng giao diện: Font Awesomes
- Hiệu ứng & tương tác: CSS, JavaScript
- Bộ SDK: .NET 8.0 SDK

Chương 3 Kết quả chương trình

3.1. Trang chủ



3.2. Đăng kí/ Đăng Nhập

3.2.1. Đăng kí

Home Shop Contact (0)

Create a new account.

Email

Password

Confirm Password

Register

Use another service to register.

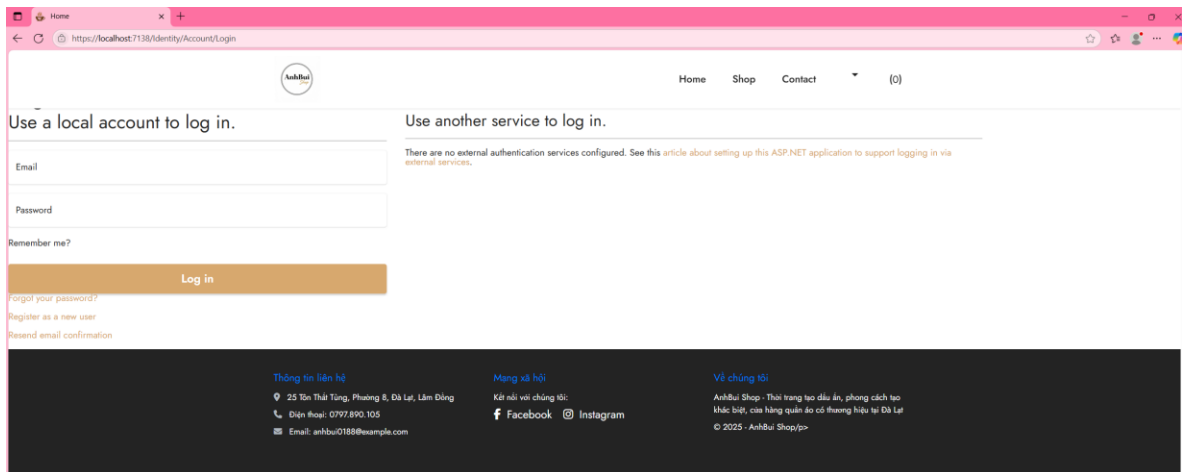
There are no external authentication services configured. See this article about setting up this ASP.NET application to support logging in via external services.

Thông tin liên hệ
 25 Tôn Thất Tùng, Phường 8, Đà Lạt, Lâm Đồng
 Điện thoại: 0797.890.105
 Email: anhbu0188@example.com

Mạng xã hội
 Kết nối với chúng tôi:
 Facebook Instagram

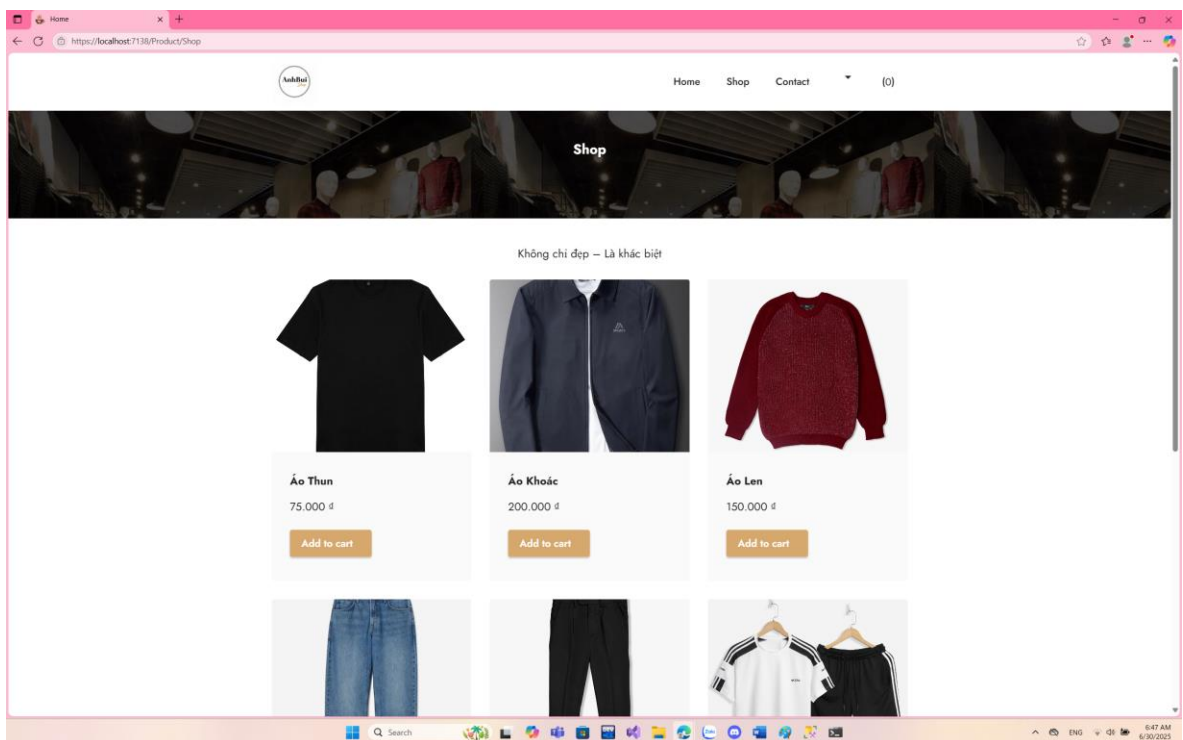
Về chúng tôi
 AnhBui Shop: Thời trang tạo dấu ấn, phong cách tạo khác biệt, các hàng quần áo có thương hiệu tại Đà Lạt
 © 2025 AnhBui Shop.vn

3.2.2. Đăng nhập

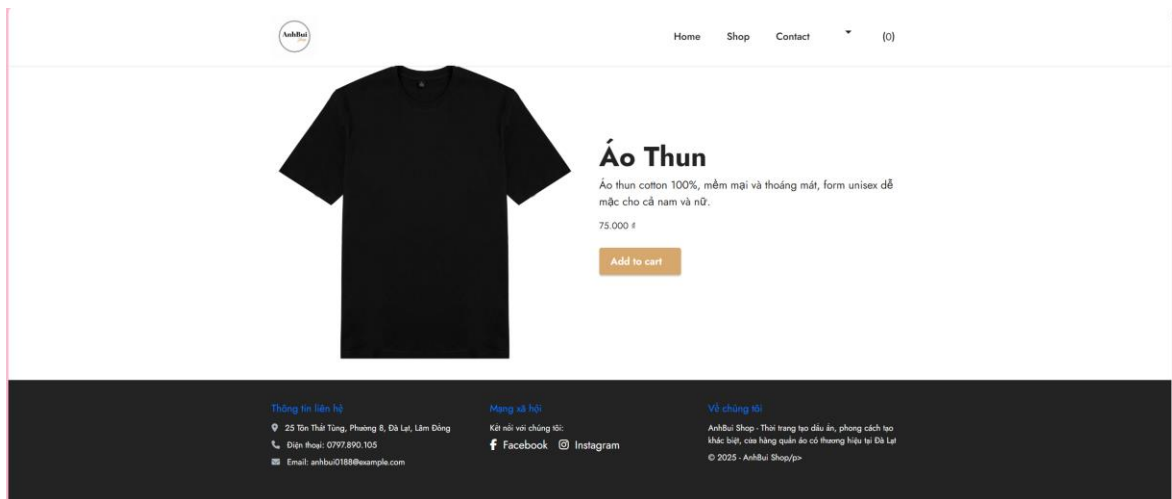


3.3. Shop

3.3.1. Trang chủ chính của Shop

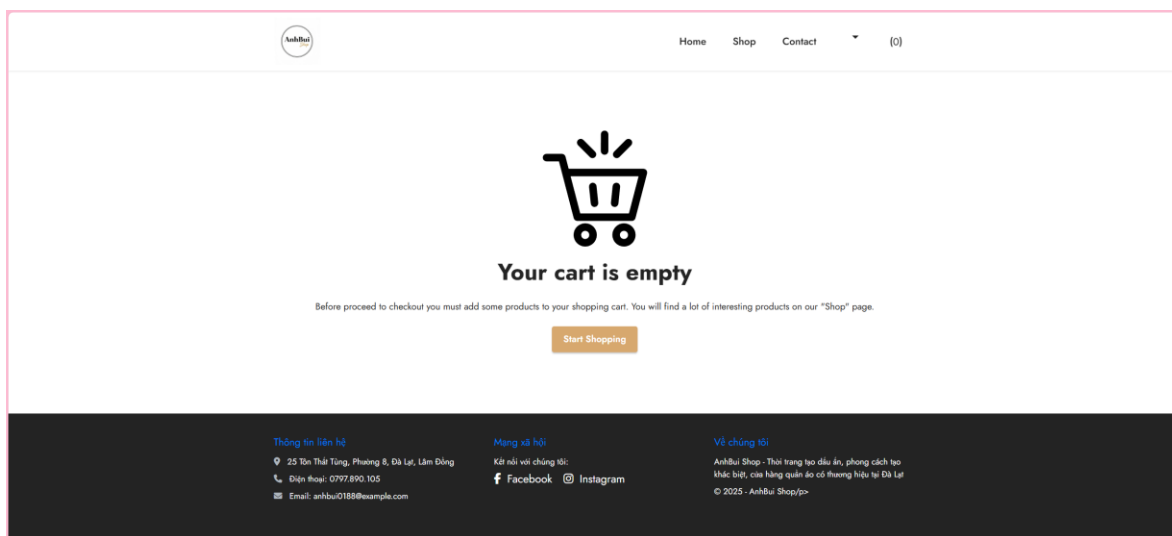


3.3.1. Thông tin chi tiết sản phẩm

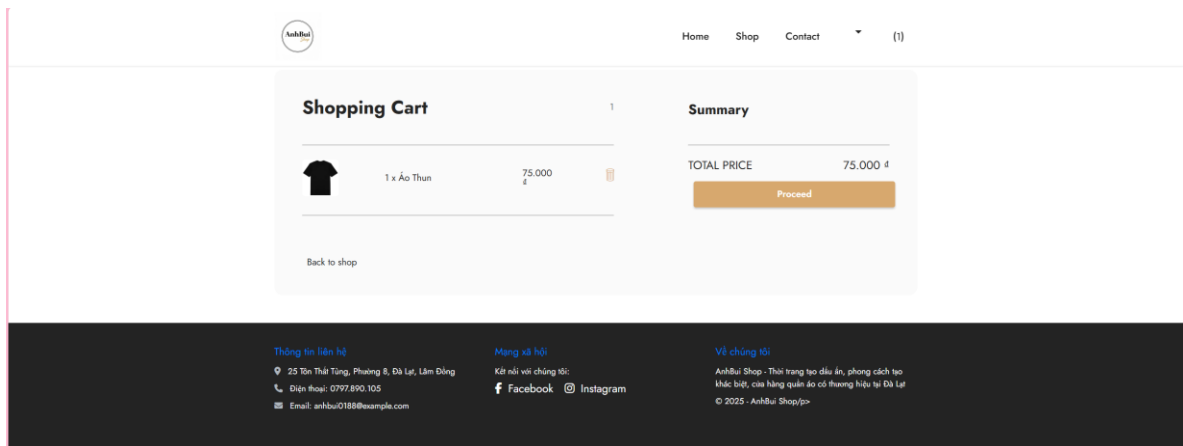


3.4. Giỏ hàng và mua hàng

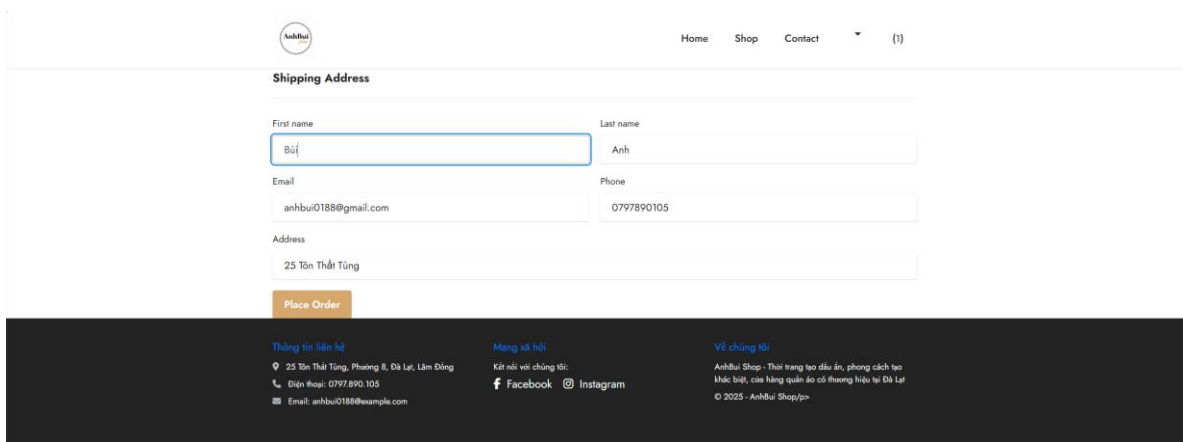
3.4.1. Giỏ hàng trống



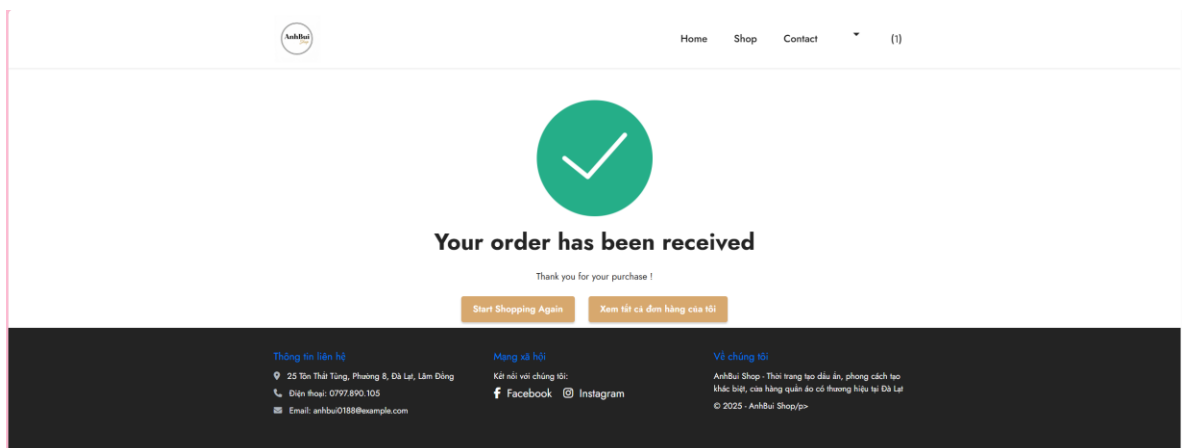
3.4.2. Giỏ hàng có hàng



3.4.3. Điền thông tin cá nhân khi mua hàng



3.4.4. Xác nhận mua hàng, quay về shop hoặc kiểm tra hóa đơn mua hàng



Chương 4. Kết luận

4.1 Tổng kết kiến thức đạt được

Đồ án đã giúp em nắm vững và vận dụng hiệu quả các công nghệ và kỹ năng quan trọng trong phát triển ứng dụng web:

- **Mô hình MVC:** Hiểu và áp dụng thành thạo cấu trúc MVC để tổ chức ứng dụng rõ ràng, dễ bảo trì và mở rộng.
- **ASP.NET Core MVC:** Thành thạo xây dựng website động, bao gồm định tuyến, xử lý logic và hiển thị dữ liệu bằng Razor View.
- **Entity Framework Core (Code First):** Áp dụng để tạo, quản lý và thao tác dữ liệu với cơ sở dữ liệu qua các lớp model hiệu quả.
- **ASP.NET Core Identity:** Tích hợp thành công chức năng đăng ký, đăng nhập và phân quyền người dùng một cách bảo mật.
- **Phát triển giao diện:** Sử dụng Bootstrap, Tailwind CSS, JavaScript và AJAX để xây dựng giao diện đẹp, responsive và tối ưu trải nghiệm người dùng.
- **Công cụ và kỹ năng lập trình:** Làm việc thành thạo với Visual Studio và SQL Server, đồng thời cải thiện kỹ năng debug, tổ chức dự án và viết mã sạch.
- **Tư duy hệ thống:** Nâng cao tư duy thiết kế hệ thống, tổ chức mã nguồn và giải quyết vấn đề theo luồng dữ liệu thực tế.

Ngoài ra, đồ án còn rèn luyện khả năng tự học, tìm kiếm tài liệu và hình thành tư duy giải quyết vấn đề chuyên nghiệp, tạo bước đệm vững chắc cho các dự án thực tế sau này.

4.2. Điểm tồn tại

Mặc dù đồ án đã hoàn thành với nhiều chức năng cơ bản và giao diện thân thiện, nhưng vẫn còn tồn tại một số điểm hạn chế cần cải thiện trong tương lai, cụ thể như sau:

- Chưa hoàn thành được phần quản trị, chưa phân quyền được giữa Admin và người dùng
- Giao diện tuy sáng sủa nhưng vẫn quá nhỏ, chưa phát huy được thế mạnh
- Chưa có thống kê doanh thu, hóa đơn theo tổng số liệu
- Hệ thống email và chức năng Quên mật khẩu chưa được hoàn thành, người dùng vẫn còn bất cập trong truy cập tài khoản và lấy lại mật khẩu cũng như định danh

4.3. Hướng mở rộng đề tài

- Mở rộng tính năng để phục vụ người dùng thuận tiện hơn
- Phân quyền vai trò để Admin quản lý cẩn thận, tránh sai sót hệ thống,
- Quản lý đơn hàng nâng cao, cập nhật để đem lại trải nghiệm tốt cho khách hàng
- Chức năng hóa đơn cho khách hàng sẽ được cập nhật sớm nhất
- Hệ thống thanh toán điện tử kèm theo đó là quản lý vận chuyển

Tài liệu tham khảo

- [1] Microsoft Docs – ASP.NET Core MVC overview – Microsoft – 2024.
<https://learn.microsoft.com/en-us/aspnet/core/mvc/overview>
- [2] **MVC là gì: Tổng quan MVC và Ứng dụng mô hình MVC trong lập trình**
[MVC là gì: Hiểu rõ mô hình MVC A-Z trong lập trình - ITviec Blog](#)
- [3] Microsoft Learn – Entity Framework Core Documentation – Microsoft – 2023.
<https://learn.microsoft.com/en-us/ef/core/>
- [4] Xây dựng website với ASP.NET Core MVC – Ths. Nguyễn Đức Tấn – 2025 Đại học Yersin (Bài giảng nội bộ)

PHỤ LỤC