**BỘ GIÁO DỤC & ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SÀI GÒN**

**KHOA CÔNG NGHỆ THÔNG TIN**

--🙢🕮🙠--

**A picture containing text, device, gauge

Description automatically generated**

**BÁO CÁO**

**MÔN HỌC: Xây dựng phần mềm theo**

**mô hình phân lớp**

***Đề tài: Website bán nhạc cụ***

***Giảng viên: Mai Vân Phương Vũ***

***Lớp: DCT1204 Thực hiện: Nhóm 2***

***TP.HCM, ngày 30 tháng 4 năm 2023***

Mục lục

[CHƯƠNG 1: MỞ ĐẦU 3](#_Toc133272707)

[I. Tổng quan 3](#_Toc133272708)

[CHƯƠNG 2: PHÂN TÍCH HỆ THỐNG 5](#_Toc133272709)

[I. ASP.NET & MVC 5](#_Toc133272710)

[1. Tại sao sử dụng ASP.NET và MVC 5](#_Toc133272711)

[2. Ứng dụng vào trong đồ án 5](#_Toc133272712)

[II. Giao diện hệ thống (Template) 5](#_Toc133272713)

[III. Cơ sở dữ liệu SQL Server 6](#_Toc133272714)

[CHƯƠNG 3. THIẾT KẾ 7](#_Toc133272715)

[I. Phân tích thiết kế hệ thống 7](#_Toc133272716)

[1. Use case tổng quát 7](#_Toc133272717)

[2. Use case Admin 7](#_Toc133272718)

[3. Use-case User 8](#_Toc133272719)

[II. Giao diện website 16](#_Toc133272720)

[CHƯƠNG 4: CODE CÁC CHỨC NĂNG ỨNG DỤNG 27](#_Toc133272721)

[I. Hiển thị danh sách 10 sản phẩm trên 1 trang 27](#_Toc133272722)

[CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN TRONG TƯƠNG LAI 58](#_Toc133272723)

[I. Kết luận 58](#_Toc133272724)

[II. Hướng phát triển 58](#_Toc133272725)

[III. Phân công công việc 58](#_Toc133272726)

[CHƯƠNG 6. TÀI LIỆU THAM KHẢO 59](#_Toc133272727)

# CHƯƠNG 1: MỞ ĐẦU

### I. Tổng quan

1. Giới thiệu đề tài

Cùng với sự phát triển như vũ bão của công nghệ thông tin hiện nay thì việc xây dựng website bán hàng hàng online kết hợp với kinh doanh truyền thống là phương pháp phổ biến và không còn gì xa lạ với tất cả các doanh nghiệp. Chúng thay thế quá trình quản lý và tính toán thủ công trên sổ sách, giấy tờ truyền thống nhờ khả năng quản lý thông minh hơn, tự động tính toán giá tiền, hơn nữa việc tra cứu, xử lí thông tin dễ dàng, chính xác và nhanh chóng hơn nhiều. Việc ứng dụng tin học vào các công ty hoạt động chính phủ, tổ chức cũng như các công ty, cửa hàng đóng vai trò hết sức quan trọng, có thể tạo nên bước đột phá, cuộc cách mạng mạnh mẽ.

1. Động lực để làm đề tài

Trong hoạt động sản xuất, kinh doanh, thương mại điện tử đã khẳng định được xúc tiến và thúc đẩy sự phát triển của doanh nghiệp. Đối với một cửa hàng, việc quảng bá và giới thiệu sản phẩm đến khách hàng đáp ứng nhu cầu mua sắm ngày càng cao của khách hàng sẽ là cần thiết. Và hơn hết là trong thời kì dịch bệnh Covid-19 như hiện nay, việc tiếp xúc trực tiếp ở nơi công cộng và đông người sẽ có nguy cơ lây nhiễm rất cao. Ngoài ra Website còn là nơi cửa hàng quảng bá tốt nhất tất cả các sản phẩm mình bán ra, giảm chí phí thuê các dịch vụ quảng cáo truyền thống tốn kém như tờ rơi.

Chính vì thế, nhóm chúng em đã thực hiện đề tài “Xây dựng Website bán nhạc cụ âm nhạc”. Cửa hàng có thể đưa các sản phẩm lên Website của mình và quản lý Website đó, khách hàng có thể đặt mua, mua hàng của cửa hàng mà không cần đến cửa hàng, cửa hàng sẽ gửi sản phẩm đến tận tay khách hàng.

1. Mục đích đề tài

Mục đích đề tài là đáp ứng nhu cầu mua bán, xem thông tin sản phẩm mới một cách nhanh chóng tiện lợi, an toàn, cụ thể như sau:

|  |  |
| --- | --- |
| Khách hàng | Quản trị |
| + Xem thông tin sản phẩm, giá cả, kiểu dáng  + Tìm kiếm sản phẩm theo yêu cầu  + Mua sản phẩm bằng cách đặt hàng qua mạng an toàn nhanh chóng.  + Xem tin tức về sản phẩm mới trên thị trường, thông tin Khuyến mại | + Quản lý các loại sản phẩm  + Quản lý mặt hàng  + Quản lý khách hàng  + Quản lý đơn hàng  + Quản lý tin tức  + Quản lý khuyến mại |

# CHƯƠNG 2: PHÂN TÍCH HỆ THỐNG

Trong đồ án Website quản lý nhạc cụ âm nhạc, công nghệ mà chính mà nhóm em sử dụng đó chính là ASP.NET, được viết bằng C# dựa trên mô hình MVC, ngoài ra còn có các Template Layout hỗ trợ sẵn để rút ngắn quá trình thiết kế giao diện. Cuối cùng là cơ sở dữ liệu SQL Server để thực hiện lưu tất cả thông tin cần thiết cho đồ án.

### I. ASP.NET & MVC

### Tại sao sử dụng ASP.NET và MVC

Lý do, cho việc ta sử dụng Asp.net mà không phải các ngôn ngữ khác như html, css, php,… Bởi vì đây là một Framework được phát triển và hỗ trợ lâu dài bởi công ty lớn là Microsoft. Bản thân chúng cho phép các nhà phát triển tạo ra các ứng dụng web động phức tạp, cung cấp nhiều thư viện, gói phần mềm hỗ trợ cho việc phát triển ứng dụng web, giúp tăng tốc độ phát triển và giảm độ phức tạp của mã nguồn, có tính năng bảo mật cao, với các công cụ bảo mật như mã hóa dữ liệu, xác thực người dùng, kiểm soát truy cập, giúp bảo vệ ứng dụng khỏi các cuộc tấn công mạng, có khả năng tương thích với nhiều nền tảng và hệ điều hành, cho phép nhà phát triển triển khai ứng dụng trên nhiều môi trường khác nhau

Nguyên nhân việc ta chọn sử dụng mô hình MVC so với mô hình ba lớp bởi vì tách biệt các thành phần của ứng dụng, giúp dễ dàng bảo trì, mở rộng và phát triển ứng dụng, có khả năng tái sử dụng cao hơn, giảm thời gian phát triển và tăng tính đáng tin cậy, kiểm thử từng phần của ứng dụng một cách độc lập, được phân chia cho nhiều nhóm nhỏ, tăng tính hiệu quả và tiết kiệm thời gian.

### Ứng dụng vào trong đồ án

Chương trình được thiết kế một cách logic, phân chia nhiệm vụ chức năng của từng thành phần một cách hợp lí. Lớp Model chứa các phương thức để thực hiện thao tác với cơ sở dữ liệu SQL Server như việc lấy thông tin của các loại nhạc cụ. Tiếp đến lớp Controller sẽ tính toán, xử lý các sự kiện theo yêu cầu cho trước như sắp xếp, sửa, xóa, tính toán, truy xuất. Cuối cùng lớp View sẽ hiển thị những giao diện, các layout nút nhấn thao tác, để người dùng có thể dễ dàng tương tác với Controller.

### II. Giao diện hệ thống (Template)

**Giới thiệu giao diện hệ thống**

Là các bản thiết kế giao diện website được xây dựng sẵn với các phần tử và bố cục đã được định sẵn. Ta có thể sử dụng chúng để thiết kế trang web của mình một cách nhanh chóng và dễ dàng hơn so với việc tự viết HTML và CSS thủ công. Nó giúp chúng ta tiết kiệm thời gian, do đó không cần phải tạo ra từng phần tử và tính năng của trang web một cách độc lập. Chúng dễ sử dụng bởi không cần có kiến thức chuyên sâu về HTML, CSS hoặc JavaScript để sử dụng chúng. Cuối cùng là vô cùng tiết kiệm chi phí, bởi nếu bạn không muốn thuê một nhà thiết kế web chuyên nghiệp, sử dụng template web dựng sẵn là một giải pháp tiết kiệm chi phí cho doanh nghiệp của mình.

### III. Cơ sở dữ liệu SQL Server

1. Tại sao lại sử dụng SQL Server

SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ được phát triển bởi Microsoft. Nó cho phép bạn lưu trữ và quản lý dữ liệu của mình trong một cách dễ dàng và có thể truy xuất thông tin từ nhiều nguồn dữ liệu khác nhau. Ưu điểm của SQL Server so với MySQL là nó có khả năng xử lý tốt hơn với các ứng dụng doanh nghiệp lớn, với khả năng quản lý và xử lý dữ liệu có kích thước lớn hơn, tối ưu hóa hiệu suất và bảo mật dữ liệu tốt hơn. Nó cũng hỗ trợ các tính năng quản lý dữ liệu như sao lưu, khôi phục và phân tích dữ liệu. So với các cách lưu trữ dữ liệu thông thường như file Excel hay CSV, SQL Server cho phép bạn lưu trữ dữ liệu trong một cơ sở dữ liệu quan hệ, giúp bạn quản lý và truy xuất dữ liệu một cách dễ dàng và có tính khả năng mở rộng. Nó cũng cung cấp tính năng đa người dùng, cho phép nhiều người dùng cùng truy cập và làm việc trên cùng một cơ sở dữ liệu cùng lúc.

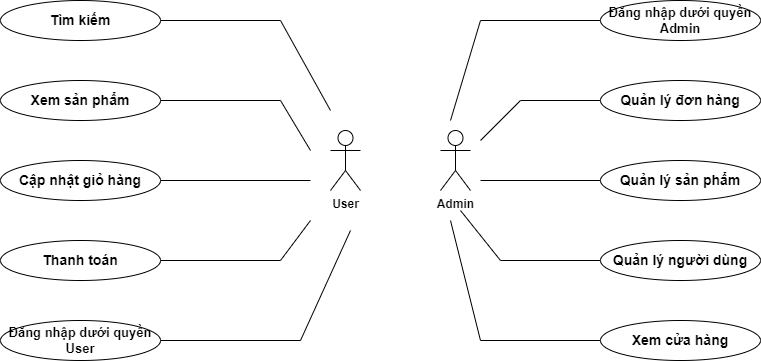
2. Cơ sở dữ liệu SQL Server trong đồ án

Trong đồ án website quản lý nhạc cụ âm nhạc, nhóm em đã dùng SQL Server làm cơ sở dữ liệu trong đồ án. Là nơi lưu trữ mọi dữ liệu của website, từ thông tin khách hàng, thông tin của những nhạc cụ khác nhau, kích cỡ, từng loại nhạc cụ riêng biệt, thông tin đơn đặt, chi tiết người dùng, và các thông tin liên quan. CSDL khi được liên kết với Visual Studio bằng cách ADO.NET Entity Data Model với chức năng tự động cập nhật các model trong ASP.NET MVC của Visual Studio thì sẽ cho kết quả là các class ứng với các bảng trong CSDL.

# CHƯƠNG 3. THIẾT KẾ

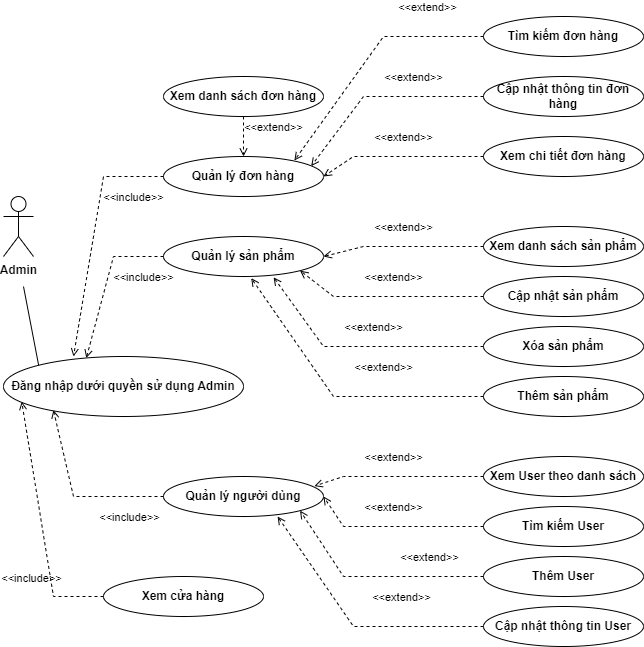
### I. Phân tích thiết kế hệ thống

### Use case tổng quát

****

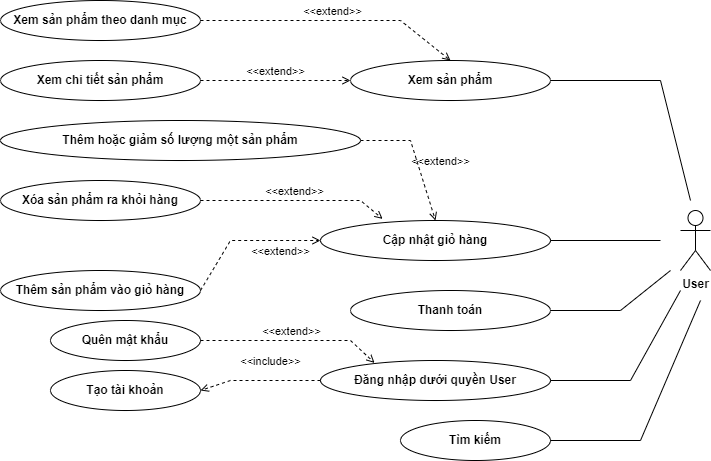
Hình 1. User tổng quát

### Use case Admin

****

Hình 2. Use case Admin

### Use-case User

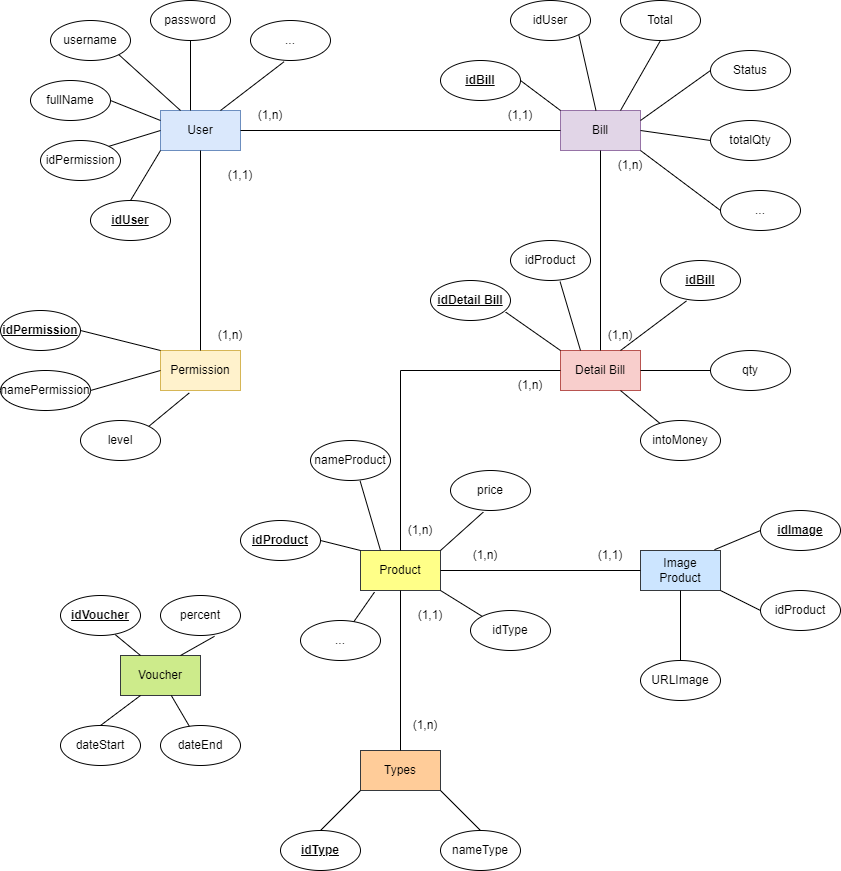


Hinh 3. Use case User

4. Database

**a. Sơ đồ ERD**

- Thiết kế sơ đồ ERD (chen):



– Sơ đồ ERD (dạng bảng) trong CSDL SQL Server:

Graphical user interface, application

Description automatically generated

**b. Các thực thể**

– Thực thể bill (hóa đơn):

Graphical user interface, application

Description automatically generated

Hình 5. Thực thể bill

Tên thực thể: Hóa đơn

Khoá thực thể: idBill

Diễn giải: thực thể Bill có các thuộc tính

|  |  |
| --- | --- |
| **+ idBill** | Mã nhân viên |
| **+ idUser** | Mã người dùng |
| **+ Shipping** | Phí vận chuyển |
| **+ Total** | Tổng tiền |
| **+ PTTT** | Phương thức thanh toán |
| **+ status** | Trạng thái đơn hàng |
| **+ createdAt** | Ngày tạo hóa đơn |
| **+ totalQty** | Số lượng |
| **+ nameBook** | Tên người dùng |
| **+ email** | Email người dùng |
| **+ phone** | Số điện thoại người dùng |
| **+ address** | Địa chỉ giao hàng |

* **Thực thể Detail Bill (chi tiết hóa đơn)**

Graphical user interface, application

Description automatically generated

Hình 6. Thực thể detail bill

Tên thực thể: Chi tiết hóa đơn

Khoá thực thể: idDetailBill

Diễn giải: thực thể DetailBill có các thuộc tính

|  |  |
| --- | --- |
| **+ idDetailBill** | Mã chi tiết hóa đơn |
| **+ idProduct** | Mã sản phẩm |
| **+ idBill** | Mã hóa đơn |
| **+ qty** | Số lượng |
| **+ intoMoney** | Tổng tiền |

* **Thực thể Product (sản phẩm):**

Graphical user interface, application

Description automatically generated

Hình 7. Thực thể product

Tên thực thể: Sản phẩm

Khoá của thực thể: idProduct

Diễn giải: thực thể Product có các thuộc tính:

|  |  |
| --- | --- |
| **+ idProduct** | Mã sản phẩm |
| **+ nameProduct** | Tên sản phẩm |
| **+ sizeM** | Số lượng sản phẩm size M |
| **+ sizeL** | Số lượng sản phẩm size L |
| **+ sizeXL** | Số lượng sản phẩm size XL |
| **+ price** | Giá sản phẩm |
| **+ description** | Mô tả sản phẩm |
| **+ idType** | Mã danh mục |

* Thực thể User (người dùng):

Graphical user interface, application

Description automatically generated

Hình 8. Thực thể User

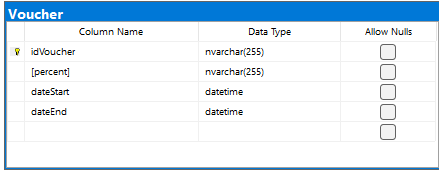
Tên thực thể: Người dùng

Khoá thực thể: idUser

Diễn giải: thực thể User có các thuộc tính:

|  |  |
| --- | --- |
| **+ idUser** | Mã người dùng |
| **+ idPermission** | Mã phân quyền |
| **+ username** | Tên đăng nhập |
| **+ password** | Mật khẩu |
| **+ gender** | Giới tính |
| **+ CMND** | Số CMND |
| **+ address** | Địa chỉ |
| **+ phone** | Số điện thoại |

Thực thể voucher (phiếu giảm giá)



Tên thực thể: Voucher

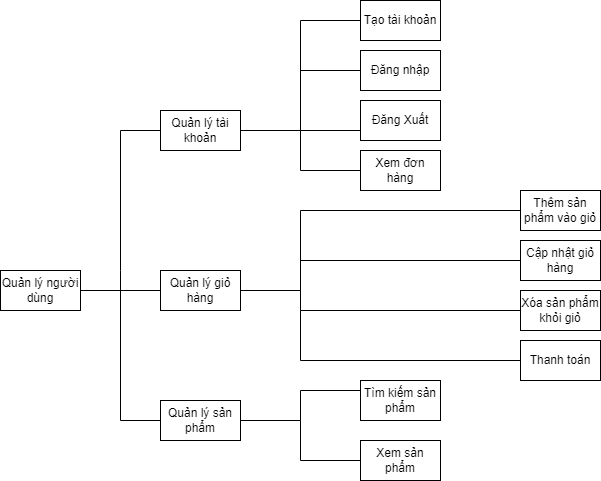
Khoá thực thể: idVoucher

Diễn giải: thực thể Voucher có các thuộc tính:

|  |  |
| --- | --- |
| **+ idVoucher** | Mã phiếu giảm giá |
| **+ percent** | Phần trăm giảm giá |
| **+ dateStart** | Thời gian bắt đầu giảm |
| **+ dateEnd** | Thời gian kết thúc giảm |

5. Cây chức năng (Function Tree)

**a. Cây chức năng User**



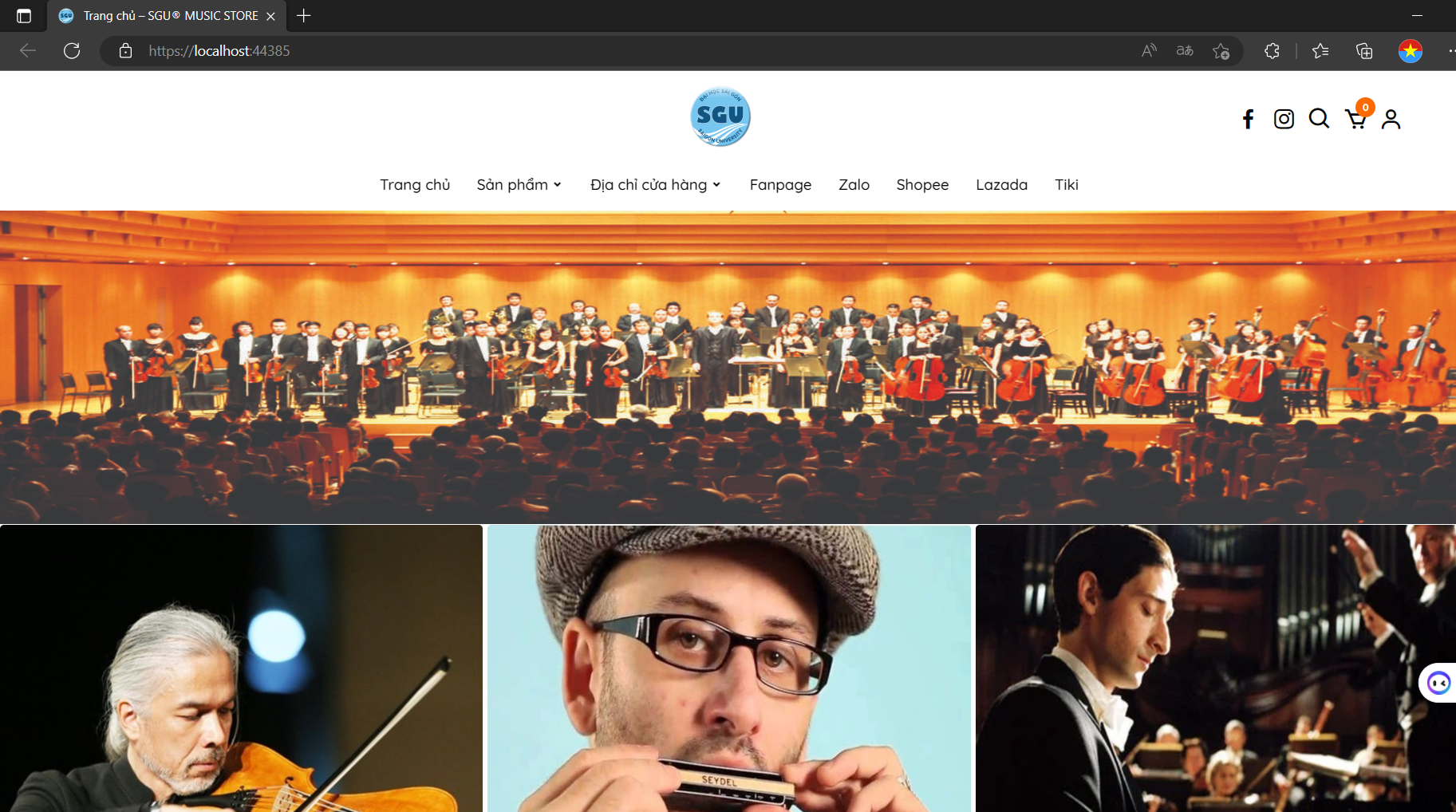
**b. Cây chức năng Admin**

A picture containing graphical user interface

Description automatically generated

### II. Giao diện website

1. Giao diện trang chủ:



Hình 10. Giao diện trang chủ

Graphical user interface, website

Description automatically generated

Hình 11. Giao diện trang chủ

**Giao diện trang chủ bao gồm:**

**Header:**

– Logo shop, thẻ liên kết tới các trang sản phẩm, địa chỉ cửa hàng, các trang mạng xã hội và các icon (facebook, instagram, tìm kiếm, giỏ hàng, thông tin người dùng)

**Body:**

– Banner

– Danh sách sản phẩm mới (tên, hình ảnh, giá)

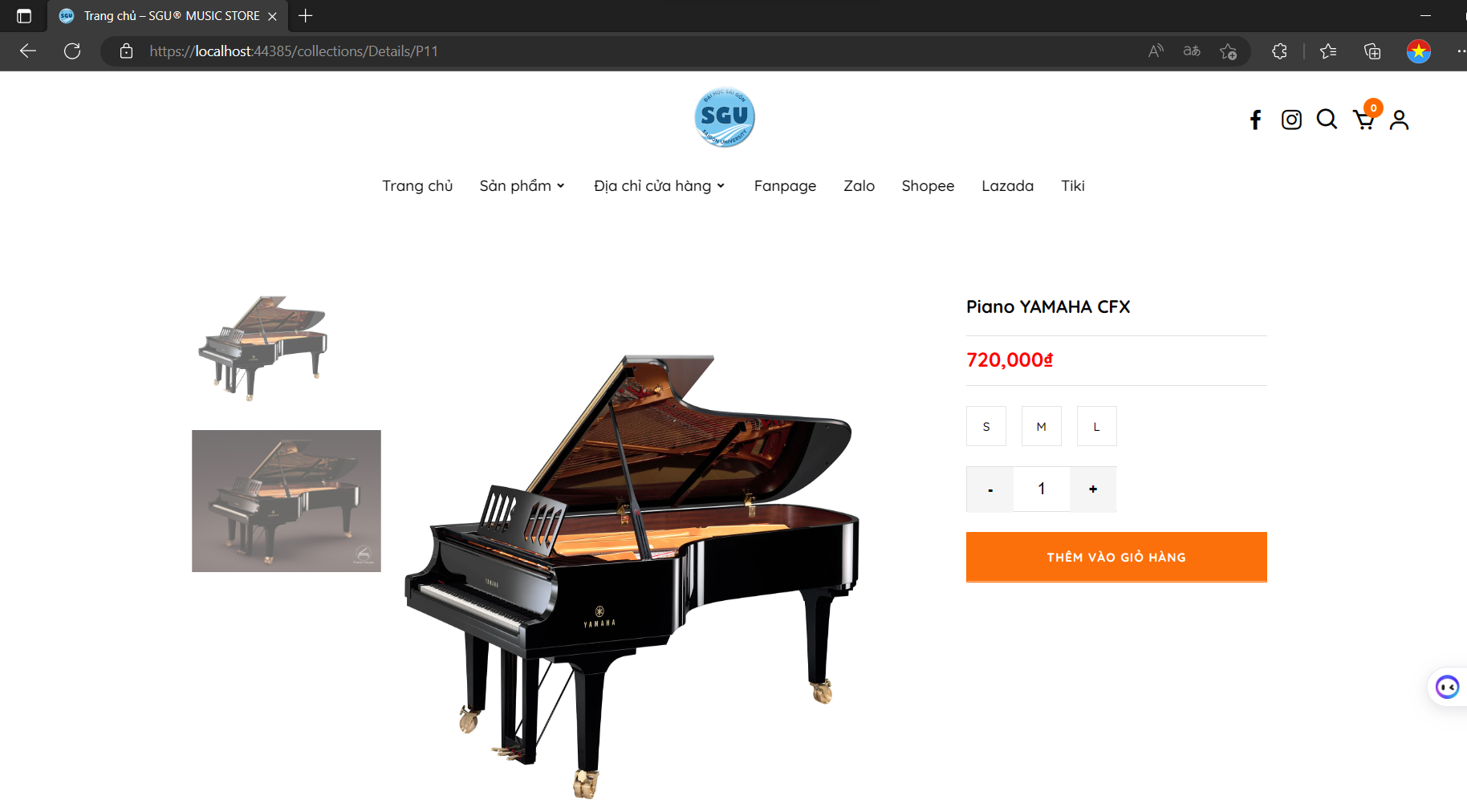
– Đăng kí nhận tin

– Hotline chăm sóc khách hàng.

**Footer:**

– Hiển thị thông tin của shop

2. Trang chi tiết sản phẩm



Hình 12. Trang chi tiết sản phẩm

**– Trang chi tiết sản phẩm bao gồm:**

+ Tên sản phẩm

+ Giá sản phẩm

+ Size

+ Số lượng cần thêm vào giỏ hàng

+ Button thêm giỏ hàng

+ Mô tả sản phẩm

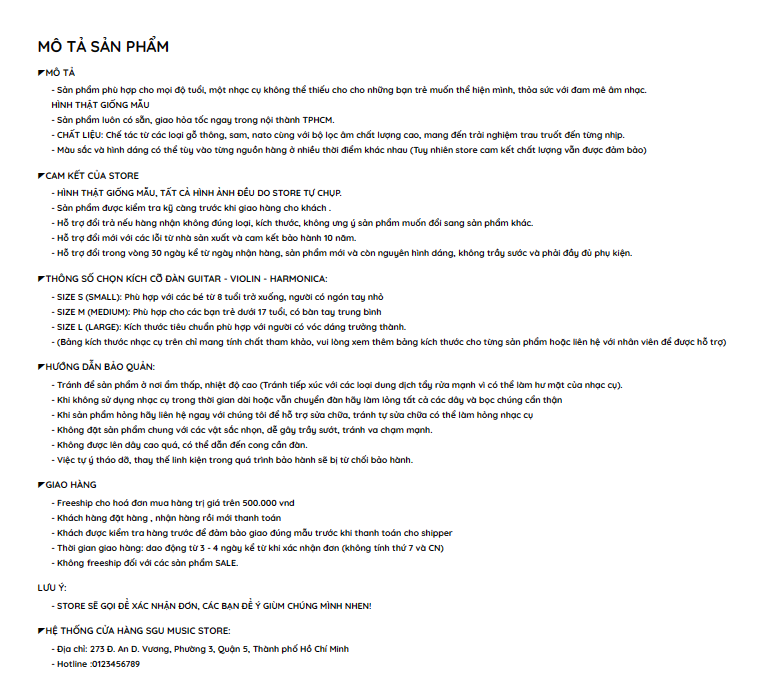
+ Cam kết của shop

+ Thông số chọn size

+ Hướng dẫn bảo quản

+ Hình thức giao hàng

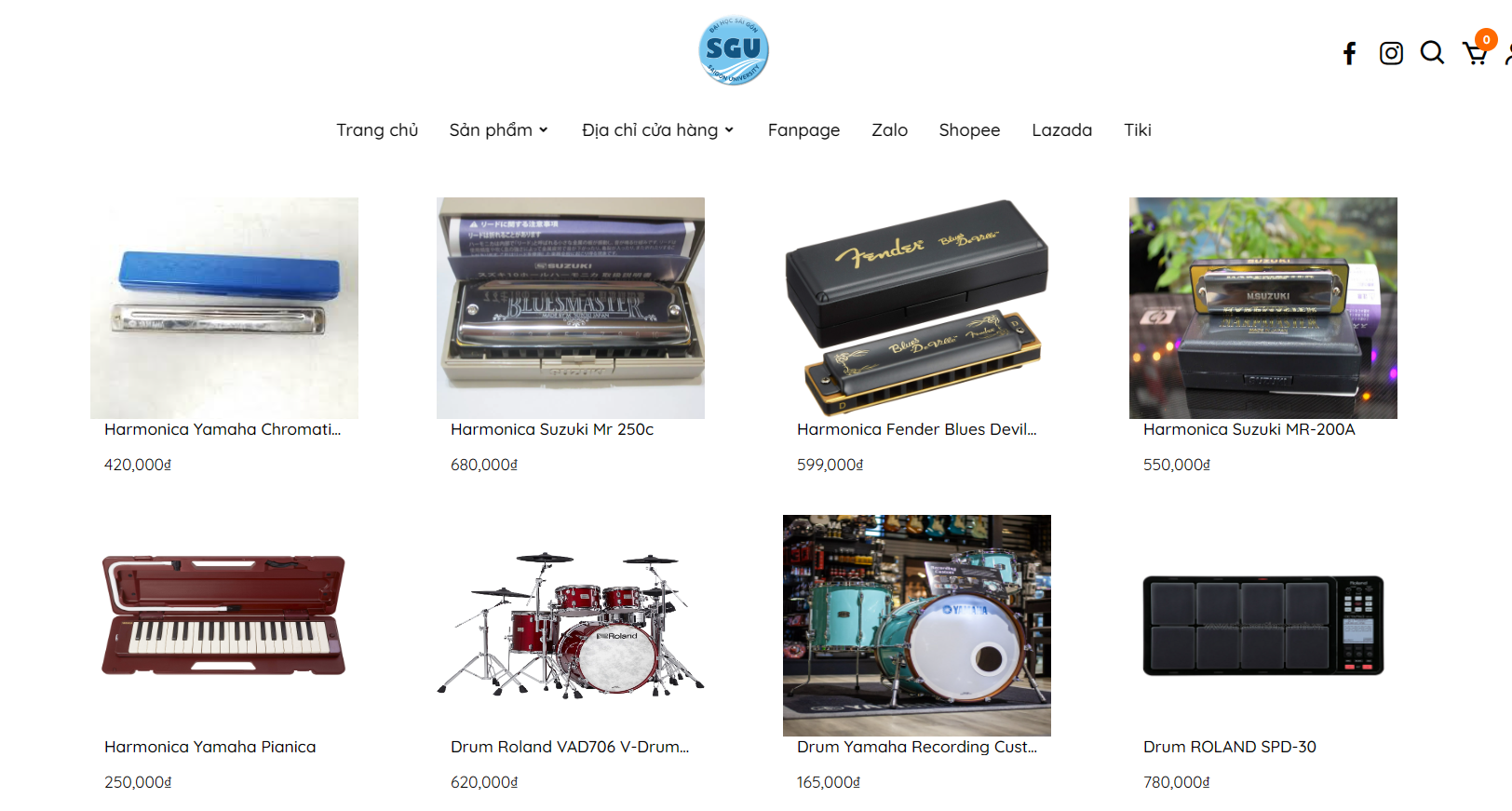
+ Một số lưu ý





Hình 13. Trang chi tiết sản phẩm

3. Trang danh sách sản phẩm



Hình 14. Trang danh sách sản phẩm

Graphical user interface

Description automatically generated

Hình 15. Trang danh sách sản phẩm

**Trang danh sách sản phẩm bao gồm:**

**Header:**

– Logo shop, thẻ liên kết tới các trang sản phẩm, địa chỉ cửa hàng, các trang mạng xã hội và các icon (facebook, instagram, tìm kiếm, giỏ hàng, thông tin người dùng)

**Body:**

– Select danh mục: Chọn xem danh sách sản phẩm theo mục.

– Span sắp xếp theo giá, tên, tình trạng sản phẩm.

– Thông tin của sản phẩm: Tên sản phẩm, giá sản phẩm, ảnh sản phẩm, hiện hết hàng khi sản phẩm sold off.

– Page Navigation: Hiển thị 16 sản phẩm trên 1 trang

+ Khi click vào button next: sẽ chuyển sang trang kế tiếp

+ Chọn số trang: sẽ chuyển qua trang mong muố

+ Khi click vào button previous: sẽ chuyển về trang trước đó

**Footer:**

– Hiển thị thông tin của shop

4. Trang tổng quan quản lý bởi Admin

Graphical user interface, application

Description automatically generated

Hình 16. Trang tổng quan quản lý bởi Admin

**Trang tổng quan quản lý bởi Admin bao gồm:**

– Logo và tên shop

– Danh mục bên trái liên kết đến các trang

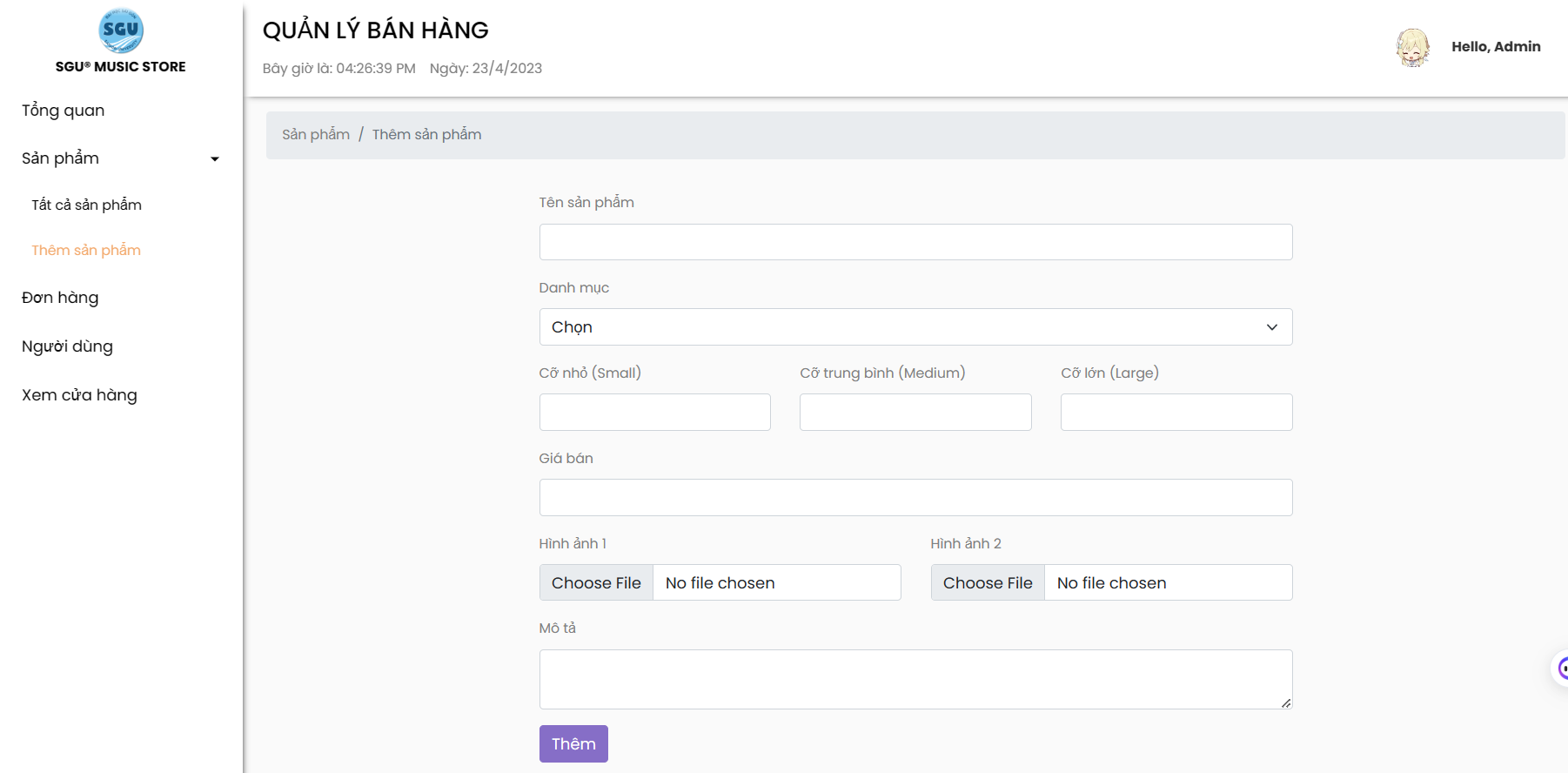
– Heading tên trang và ngày giờ đang sử dụng.

– Ba module hiển thị đếm tổng số sản phẩm, đơn hàng và khách hàng.

– Bảng thống kê số đơn hàng, số lượng, doanh thu gần đây.

– Biểu đồ thống kê số liệu các đơn trong tuần.

5. Trang quản lý thêm mới sản phẩm



Hình 17. Trang thêm sản phẩm

**Trang thêm sản phẩm bao gồm:**

– Logo và tên shop

– Danh mục bên trái liên kết đến các trang

– Heading tên trang và ngày giờ đang sử dụng.

– Textbox nhập thông tin sản phẩm (tên sản phẩm, danh mục, size, giá, số lượng, link ảnh, mô tả)

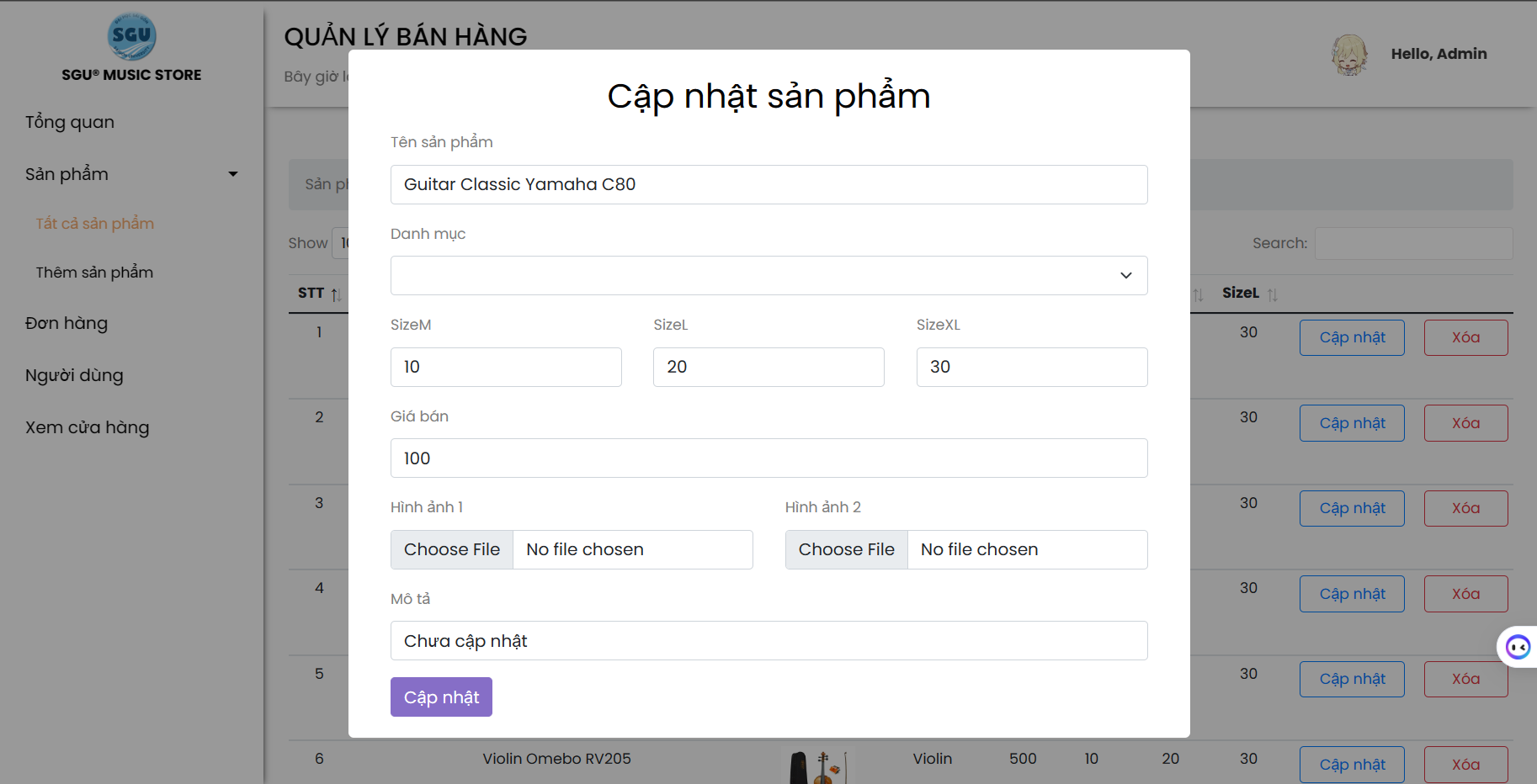
– Choose file (ảnh đính kèm)

– Button:

+ Button hủy: đóng form thêm mới sản phẩm

+ Button lưu: lưu sản phẩm mới thành công và hiển thị sản phẩm đó trên table sản phẩm

6. Trang quản lý cập nhật sản phẩm



Hình 18. Trang cập nhật sản phẩm

**Trang cập nhật sản phẩm bao gồm:**

– Bảng cập nhật sản phẩm

+ Textbox nhập thông tin sản phẩm (tên sản phẩm, danh mục, số lượng size, giá bán, mô tả sản phẩm)

+ Button cập nhật

7. Trang quản lý đơn hàng

**Graphical user interface, application

Description automatically generated**

Hình 19. Đơn hàng

**Trang đơn hàng bao gồm:**

**Header:**

– Logo shop, thẻ liên kết tới các trang sản phẩm, địa chỉ cửa hàng, các trang mạng xã hội và các icon (facebook, instagram, tìm kiếm, giỏ hàng, thông tin người dùng)

**Body:**

– Button cập nhật: hiển thị form thêm mới sản phẩm khi click vào nó.

– Table chứa các thông tin của sản phẩm: ID, tên khách hàng, số điện thoại, tổng tiền, thanh toán, trang thái.

– Button chi tiết: hiên thị form chi tiết sản phẩm khi kích vào nó.

– Page Navigation:

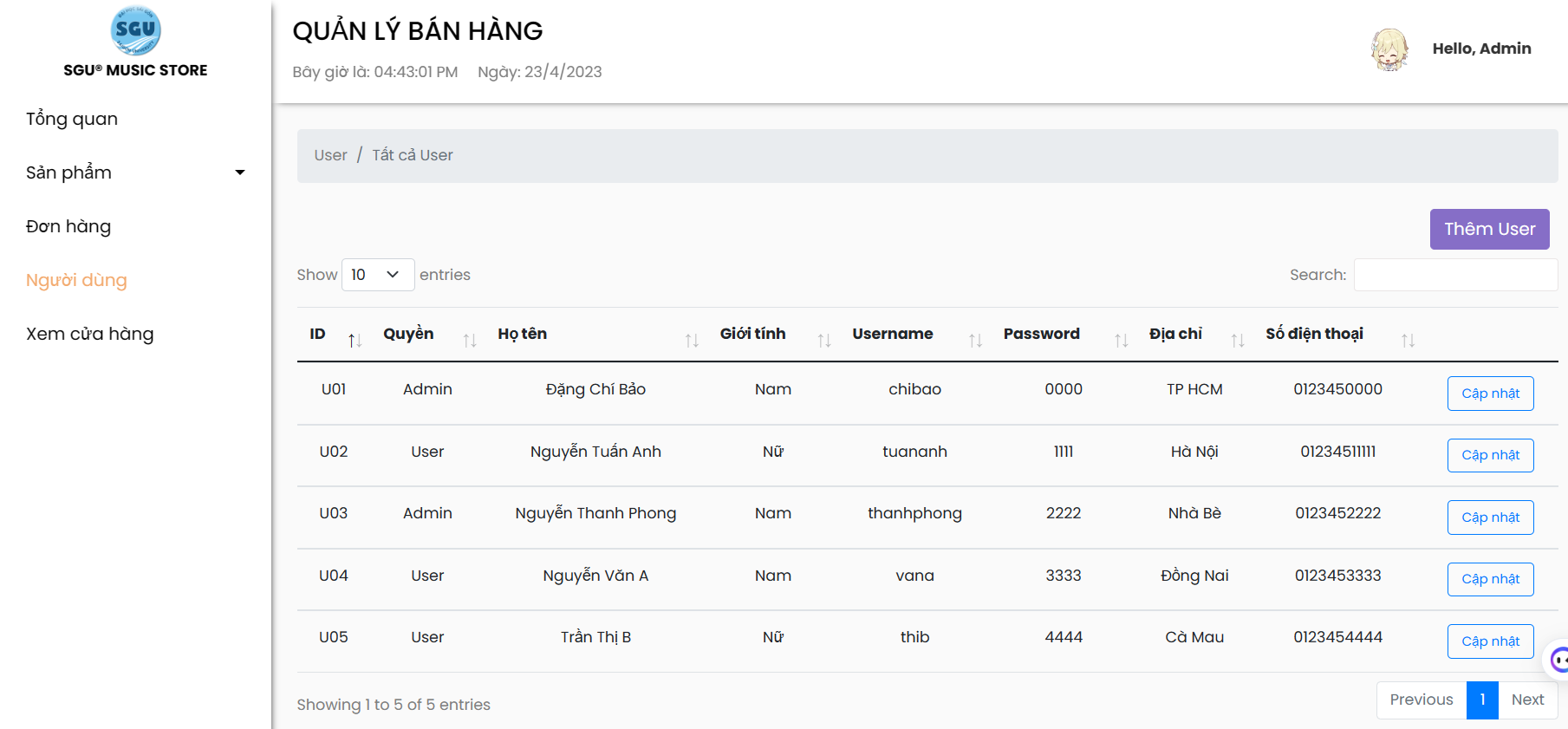
– Khi click vào button next: sẽ chuyển sang trang kế tiếp

– Chọn số trang: sẽ chuyển qua trang mong muốn

**Footer:**

– Hiển thị thông tin của shop

8. Trang quản lý người dùng

****

Hình 20. Trang quản lý người dùng

**Trang quản lý người dùng bao gồm:**

**Header:**

– Logo shop, thẻ liên kết tới các trang sản phẩm, địa chỉ cửa hàng, các trang mạng xã hội và các icon (facebook, instagram, tìm kiếm, giỏ hàng, thông tin người dùng)

**Body:**

– Button cập nhật: hiển thị form thêm mới sản phẩm khi click vào nó.

– Table chứa các thông tin của sản phẩm: ID, họ tên, số điện thoại, giới tính, tài khoản, mật khẩu, phân quyền.

– Button chi tiết: hiên thị form chi tiết sản phẩm khi kích vào nó.

– Page Navigation:

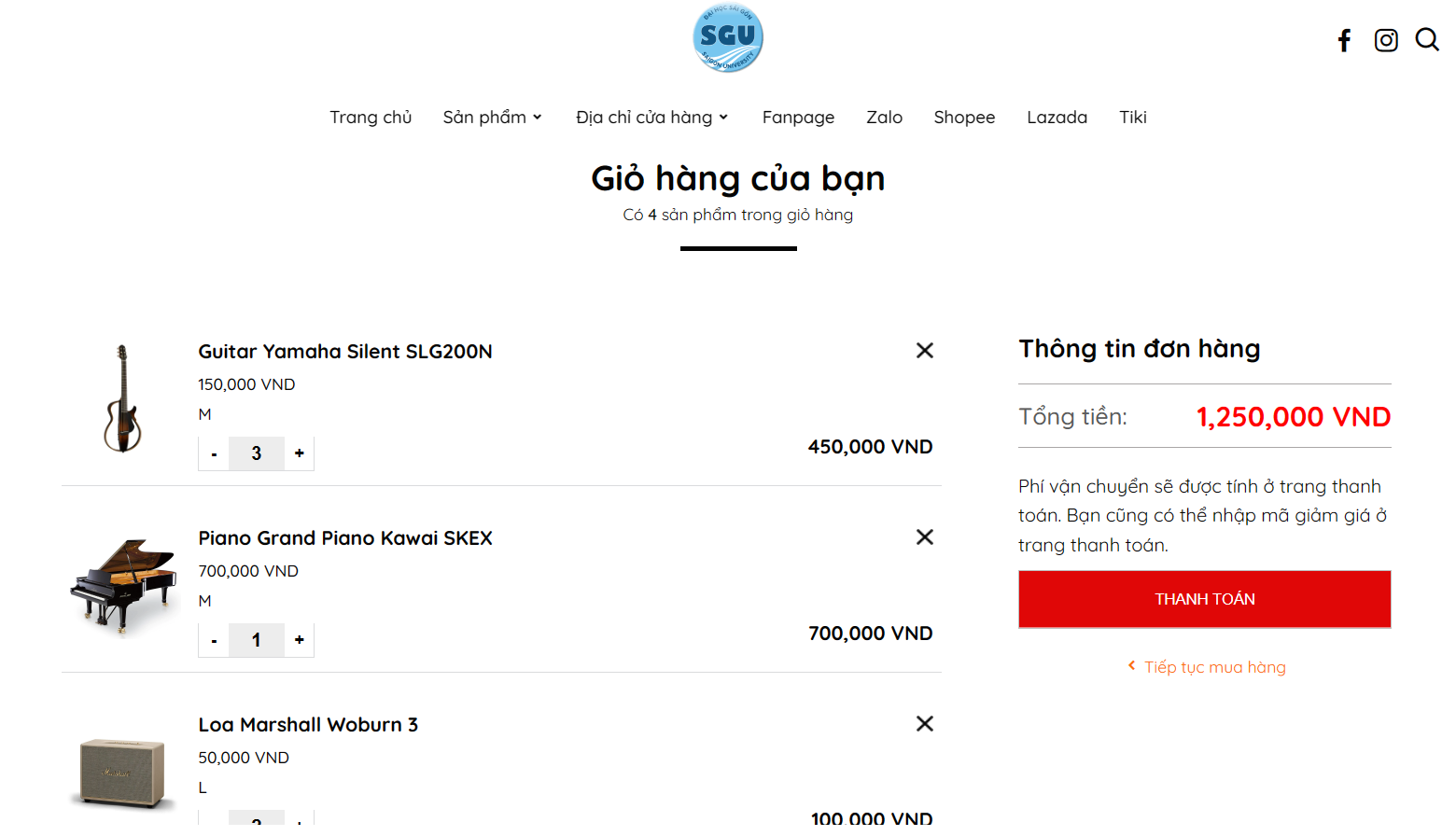
– Khi click vào button next: sẽ chuyển sang trang kế tiếp

– Chọn số trang: sẽ chuyển qua trang mong muốn

**Footer:**

– Hiển thị thông tin của sho

9. Trang giỏ hàng



Hình 21. Trang giỏ hàng

Trang giỏ hàng bao gồm:

Header:

– Logo shop, thẻ liên kết tới các trang sản phẩm, địa chỉ cửa hàng, các trang mạng xã hội và các icon (facebook, instagram, tìm kiếm, giỏ hàng, thông tin người dùng)

Body:

– Text: hiển thị số loại sản phẩm có trong giỏ hàng

– Danh sách các sản phẩm đã chọn

+ Hình ảnh

+ Tên sản phẩm

+ Đơn giá

+ Số lượng

+ Số tiền: hiển thị tổng số tiền mua sản phẩm

+ Thao tác:

\* Button xóa: xóa sản phẩm khỏi giỏ hàng

\* Button thanh toán: thanh toán các sản phẩm đã chọn

**Footer:**

– Hiển thị thông tin của shop

# 

# CHƯƠNG 4: CODE CÁC CHỨC NĂNG ỨNG DỤNG

### I. Hiển thị danh sách 10 sản phẩm trên 1 trang

1. Trang home

- Back End:

Truy vấn LINQ loại sản phẩm mới nhất được thiết kế trong database sau đó Query những dữ liệu image liên quan đến dữ liệu đã query trong linQ, cuối cùng return dữ liệu.

**HomeController.cs**

|  |
| --- |
| namespace SGUMusicStore.Controllers  {  public class HomeController : Controller  {  private SGUMusicStoreEntities db = new SGUMusicStoreEntities();  // GET: Home  public ActionResult Index()  {  // Giới thiệu sản phẩm  string id = "T07";  var productList = (from s in db.Products where s.idType == id select s);  var query = productList.Include(p => p.ImageProducts);  ViewBag.list = query.ToList();  return View(query.ToList());  }  } |

Hình 21. Code trang home back-end

**– Front End:**

Đây là phần hiển thị dữ liệu mới trong phần trong trang chủ dùng vòng lặp for Each duyệt dữ liệu và hiển thị dữ liệu với 8 sản phẩm thôi và chọn ra những sản phẩm đẹp. Chúng ta đã sử dụng câu lệnh điều kiện cho biến đếm chạy tới index thứ 8 và xuất ra tới 15 thì dừng lại.

**Home/ Index.cshtml**

|  |
| --- |
| @foreach (var item in Model)  {  cbot++;  if (cbot > 7)  {  <div class="grid\_\_col-3">  <div class="product">  <h1 style="display:none" id="idProduct">@item.idProduct</h1>  <img src="@item.ImageProducts.First().URLImage"  alt=""  class="product-img product-show" />  <img src="@item.ImageProducts.Last().URLImage"  alt=""  class="product-img product-img-hover" />  </div>  <div class="product-info">  <div class="product-title">  @item.nameProduct  </div>  <div class="product-price">@item.price,000₫</div>  </div>  </div>  }  if (cbot == 15)  { break; }  } |

Hình 22. Code trang chủ Font-End

2. Login và ghi nhớ tài khoản

**– Back End:** function với 3 tham số (username, password, ghinho)

+ Kiểm tra điều kiện cookies người dùng khác null nếu trước đó người dùng đã lưu ghi nhớ tài khoản để thêm dữ liệu người dùng vào cookies khuyết điểm này là chưa mã hóa password cookie nếu áp dụng thực tế sẽ có thể gây xung đột hệ thống.

+ Gọi function checkpassword người dùng nếu đúng sẽ return true và tiếp tục quá trình biên dịch.

+ Gọi function getInfoUser để lấy dữ liệu dữ liệu của user với tham số là username của người dùng.

+ Vì dữ liệu trả về là mảng nên ta đã dùng vòng lặp lấy giá trị ra.

+ Khai báo đối tượng trong C# ta đã khai báo bên trong model và gán giá trị cho nó.

+ Function getListGroupId () lấy quyền của người dùng với tham số username.

+ Kiểm tra câu lệnh điều kiện nếu on thì sử dụng function ghinhotaikhoan lưu vào cookie.

+ Dùng vòng lặp for để lấy ra dữ liệu.

+ Kiểm tra nều quyền admin thì lưu vào cookie với tên session admin và sử dụng function saveInfoCookies lưu dữ liệu vào cookie.

+ Nếu là admin thì di chuyển thẳng tới trang admin ngược lại thì trang home.

+ Nếu câu lệnh điều kiện trả false thì return lại trang login.

**LoginController.cs**

|  |
| --- |
| [HttpPost]  [ValidateAntiForgeryToken]  public ActionResult kiemtradangnhap(string username, string password, string ghinho)  {  var idUser = "";  if (Request.Cookies["username"] != null && Request.Cookies["password"] != null)  {  username = Request.Cookies["username"].Value;  password = Request.Cookies["password"].Value;  }    if (checkpassword(username, password))  {  var infoUser = getInfoUser(username);  var email= "";  var phone = 1;  var fullName = "";    for(var i = 0; i < infoUser.Count; i++)  {  email = infoUser[i].email;  phone = infoUser[i].phone;  fullName = infoUser[i].fullName;  idUser = infoUser[i].userId;  }    var userSession = new UserLogin();  userSession.UserName = username;  userSession.fullName = fullName;  userSession.phone = phone;  userSession.email = email;  userSession.idUser = idUser;  var group = "";  // Có thể viết dòng lệnh lấy các GroupID từ CSDL, ví dụ gán ="ADMIN", dùng List<string>  var listGroups = GetListGroupID(username);    if (ghinho == "on")// Ghi nhớ  ghinhotaikhoan(username, password);  for (var i = 0; i < listGroups.Count; i++)  {  group = listGroups[i];  }  if (group == "Admin")  {  // namePermission  Session.Add("SESSION\_GROUP\_ADMIN", listGroups);  Session.Add("USER\_SESSION", userSession);    SaveInfoInCookies(idUser,username);  return Redirect("~/Admin/Dashboard/Index");  }  else  {  Session.Add("SESSION\_GROUP", listGroups);  Session.Add("USER\_SESSION", userSession);    SaveInfoInCookies(idUser,username);  return Redirect("~/Home");  }  }  return Redirect("~/Login");  } |

Hình 23. Code login và ghi nhớ tài khoản

– Function sử dụng trong functionkiemtradangnhap

|  |
| --- |
| public bool checkpassword(string username, string password)  {  if (db.Users.Where(x => x.username == username && x.password == password).Count() > 0)  return true;  else  return false;  } |

|  |
| --- |
| public List<getUserDTO> getInfoUser (string userName)  {  getUserDTO getUserDTO = new getUserDTO();  var data = (from s in db.Users  where s.username == userName  select new getUserDTO  {  userId = s.idUser,  fullName = s.fullName ,  email = s.email ,  phone = s.phone  });  return data.ToList();  } |

|  |
| --- |
| public List<string> GetListGroupID(string userName)  {  var data = (from a in db.Permissions  join b in db.Users on a.idPermission equals b.idPermission  where b.username == userName  select new  {  UserGroupID = b.idPermission,  UserGroupName = a.namePermission  });  return data.Select(x => x.UserGroupName).ToList();  } |

|  |
| --- |
| public void ghinhotaikhoan(string username, string password)  {  HttpCookie us = new HttpCookie("username");  HttpCookie pas = new HttpCookie("password");  us.Value = username;  pas.Value = password;  us.Expires = DateTime.Now.AddDays(1);  pas.Expires = DateTime.Now.AddDays(1);  Response.Cookies.Add(us);  Response.Cookies.Add(pas);  } |

|  |
| --- |
| // Thêm hàm lưu hết thông tin trong cookies  public void SaveInfoInCookies(string idUser,string username)  {  HttpCookie us = new HttpCookie("username");    HttpCookie user = new HttpCookie("user");  us.Value = username;  user.Value = idUser;  us.Expires = DateTime.Now.AddDays(1);    user.Expires = DateTime.Now.AddDays(1);  Response.Cookies.Add(us);    Response.Cookies.Add(user);  } |

**– Front-End:**

Phần form action: gửi request đến controller login và action result là kiemtradangnhap

**Login/Index.cshtml**

|  |
| --- |
| <form action="~/Login/kiemtradangnhap" method="post" class="form-login form-login-cus">  @Html.AntiForgeryToken()  <div class="login-box-cus">  <p class="heading">đăng nhập tài khoản</p>  <div class="heading-login-with">Nhập tên đăng nhập và mật khẩu của bạn</div>  <div class="header-box login-box">  <div class="form-control err">  <input type="text"  value=""  name="username"  id="email-cus"  placeholder="Username"  class="modal-form-input" />  <small></small>  @Html.ValidationMessageFor(model => model.username, "", new { @class = "text-danger" })  </div>  <div class="form-control err">  <input type="password"  id="password-cus"  placeholder="Password"  name="password"  value=""  class="modal-form-input" />  <small></small>  @Html.ValidationMessageFor(model => model.password, "", new { @class = "text-danger" })  </div> |

3. Đăng xuất

– **Back-End:**

+ Vì đảm bảo toàn vẹn dữ liệu đăng xuất với cookies\_asp chúng ta sẽ xóa nó.

+ Với hàm deleteCookies () với ý nghĩa xóa hết thông tin dược lưu trong cookies.

**LoginController.cs**

|  |
| --- |
| // Thêm hàm xóa hết thông tin trong cookies  private void DeleteCookies()  {  Session.Clear();  HttpCookie us = Request.Cookies["username"];    HttpCookie user = Request.Cookies["user"];    us.Expires = DateTime.Now.AddDays(-1);    user.Expires = DateTime.Now.AddDays(-1);  Response.Cookies.Add(us);      Response.Cookies.Add(user);  } |

|  |
| --- |
| public ActionResult SignOut()  {  Session["USER\_SESSION"] = null;  Session["SESSION\_GROUP"] = null;  if(Request.Cookies["ASP.NET\_SessionId"]!=null)  {  HttpCookie asp = Request.Cookies["ASP.NET\_SessionId"];  asp.Expires = DateTime.Now.AddDays(-1);  Response.Cookies.Add(asp);  DeleteCookies();  }  if (Request.Cookies["username"] != null && Request.Cookies["password"] != null )  {  DeleteCookies();  HttpCookie pass = Request.Cookies["password"];  pass.Expires = DateTime.Now.AddDays(-1);  Response.Cookies.Add(pass);  }  return Redirect("/Home");  } |

**– Front-End:**

+ Gửi request đến controller login và action result signout

**Shared/ Layout.cshtml**

|  |
| --- |
| <!-- đã đăng nhập -->  <div class="header-top-modal header-logged ">      <span class="box-triangle">          <svg viewBox="0 0 20 9" role="presentation">              <path d="M.47108938 9c.2694725-.26871321.57077721-.56867841.90388257-              .89986354C3.12384116 6.36134886 5.74788116 3.76338565 9.2467995.30653888c.              4145057-.4095171 1.0844277-.40860098 1.4977971.00205122L19.4935156 9H              .47108938z"fill="#ffffff"></path>                  </svg>                      </span>                          <p class="heading">tài khoản</p>                              <div class="heading-login-with"></div>                                  <div class="info-box">                                          <p class="info-name"></p>          <p class="info-logout info-detail"><a href="/home/MyInfo"> Thông tin tài khoản</a></p>          <p class=" info-logout info-detail"><a href="/home/MyBill"> Đơn hàng của tôi</a></p>          @if (Session["SESSION\_GROUP\_ADMIN"] != null)          {              <p class="info-logout info-detail"><a href="/Admin/Dashboard/Index"> Quản lý cửa hàng</a></p>          }          <p class="info-logout"><a href="/login/SignOut"> Đăng xuất</a></p>          </div>              </div>                  }                  else                  {                      <span class="header-top-link">                          <a style="color: #050505" href="/login"><i class='bx bx-user'></i></a></span>                  } |

4. Collection (Guitar, Piano, phụ kiện, …)

– Back End:

+ Id ở đây là id của loại đàn piano chúng ta sẽ set cứng nó tuy nhiên nó chưa thật sự tối ưu cho lắm

+ Productlist là query dữ liệu với cái typeId của dòng.

+ Query ra thư mục hình ảnh của sản phẩm loại id đó trong sản phẩm đó có hình .

+ Return dữ liệu.

**CollectionsController.cs**

|  |
| --- |
| // collections/piano  public ActionResult piano(string id)  {  id = "T03";  var productList = (from s in db.Products  where s.idType == id  select s);  var query = productList.Include(p => p.ImageProducts);  ViewBag.list = query.ToList();  return View(query.ToList());  } |

**– Front End:**

+ Ta dùng foreach để duyệt mảng dữ liệu đã trả về cho client.

**Collections/ Piano.cshtml**

|  |
| --- |
| <div class="grid\_\_row">  @foreach (var item in Model)  {  <div class="grid\_\_col-3">  <div class="product">  <h1 style="display:none" id="idProduct">@item.idProduct</h1>  <img src="@item.ImageProducts.First().URLImage"  alt=""  class="product-img product-show" />  <img src="@item.ImageProducts.Last().URLImage"  alt=""  class="product-img product-img-hover" />  </div>  <div class="product-info">  <div class="product-title">  @item.nameProduct  </div>  <div class="product-price">@item.price,000₫</div>  </div>  </div>  }  </div> |

5. Search

**– Backend:** Người dùng sẽ gửi query paramater cho backend và chúng ta sử dụng thêm số q để bắt query paramter.

+ Khai báo đối tượng

+ Chuyển dữ liệu trở thành chữ thường

+ Query tên dữ liệu chưa tên người dùng query đến

+ Return dữ liệu

**SearchController.cs**

|  |
| --- |
| public class SearchController : Controller  {  private SGUMusicStoreEntities db = new SGUMusicStoreEntities();  // GET: http://localhost:46418/search/indexq?=tin  public ActionResult Index(string q)  {  // queryparamater = "instrument";  ProductDTODetail productDTO = new ProductDTODetail();  q = q.ToLower();    var productList = (from s in db.Products  where s.nameProduct.ToLower().Contains(q)  select s);  var query = productList.Include(p => p.ImageProducts);  ViewBag.list = query.ToList();  return View(query.ToList());  } |

**– Front-End:**

+ Chúng ta dùng for each để hiển thị dữ liệu.

**Search / Index.cshtml**

|  |
| --- |
| <div class="grid\_\_row">  @foreach (var item in Model)  {  <div class="grid\_\_col-3">  <div class="product">  <h1 style="display:none" id="idProduct">@item.idProduct</h1>  <img src="@item.ImageProducts.First().URLImage"  alt=""  class="product-img product-show" />  <img src="@item.ImageProducts.Last().URLImage"  alt=""  class="product-img product-img-hover" />  </div>  <div class="product-info">  <div class="product-title">  @item.nameProduct  </div>  <div class="product-price">@item.price,000₫</div>  </div>  </div>  }  </div> |

6. Hiển thị detail sản phẩm

+ Với tham số là params trên url với id của sản phẩm.

+ Ta kiểu tra id nó có NULL không nếu không NULL tiếp tục biên dịch ngược lại thì lỗi.

+ Query sản phẩm có id giống với id params.

+ Nếu NULL báo lỗi.

+ Query hình ảnh của sản phẩm .

+ Return dữ liệu.

**SearchController.cs**

|  |
| --- |
| public ActionResult Details(string id)  {  ProductDTODetail productDTO = new ProductDTODetail();  if (id == null)  {  return new HttpStatusCodeResult(HttpStatusCode.BadRequest);  }  var product = from el in db.Products  where el.idProduct == id  select el;  if (product == null)  {  return HttpNotFound();  }  else  {  var listProduct = from p in product  join image in db.ImageProducts on p.idProduct equals image.idProduct  join type in db.Types on p.idType equals type.idType  select new ProductDTODetail()  {  idProduct = p.idProduct,  type = type.nameType,  nameProduct = p.nameProduct,  price = p.price,  URLImage = image.URLImage,  sizeM = p.sizeM,  sizeS = p.sizeS,  sizeL = p.sizeL,  };  ViewBag.List = listProduct;  return View(listProduct.ToList());  }  } |

**Search/ Index.cshtml**

|  |
| --- |
| <div class="grid\_\_row">  @foreach (var item in Model)  {  <div class="grid\_\_col-3">  <div class="product">  <h1 style="display:none" id="idProduct">@item.idProduct</h1>  <img src="@item.ImageProducts.First().URLImage"  alt=""  class="product-img product-show" />  <img src="@item.ImageProducts.Last().URLImage"  alt=""  class="product-img product-img-hover" />  </div>  <div class="product-info">  <div class="product-title">  @item.nameProduct  </div>  <div class="product-price">@item.price,000₫</div>  </div>  </div>  }  </div> |

7. Thêm sản Phẩm Vào giỏ hàng

Với việc thêm dữ liệu vào giỏ hàng chúng em sử dụng jquery trên frontEnd.

Cách thức hoạt động là đầu tiên set dữ liệu cho thêm vào các biến kiểm tra cart (giở hàng) trong localstorage đã có chưa nếu chưa có add thẳng sản phẩm vào không cần phải so sánh và setLocal cart vào còn nếu ngược lại nghĩa là đã có item (cart) trong local thì việc đầu tiên là duyệt mảng so sánh tên (name) của sản phẩm trong item (cart) so sánh tên sản phẩm thì nếu sản phẩm có cùng tên cùng size thì cộng số lượng lên bên trong local storage thuôc item(cart) ngược lại nếu khác size thì phải add dữ liệu mới vào với số lượng 1 tuy cùng tên nhưng khác size nên nó là 1 dữ liệu. Ngược lại nếu dữ liệu khác tên bên trong mảng item ('cart') không có tên nào cùng thì ta tạo ra dữ liệu add vào item.

**Script/ detail.js**

|  |
| --- |
| let ttttMoney = document.querySelector('.payment-total');  btnAddCart.addEventListener("click", (e) => {  let size = getSize();  if (size != null) {  let img = document.querySelector("#imagebox").src;  let priceFake = document.querySelector(".total-money").textContent;  let price = Number(priceFake.slice(0, 3).trim());  let title = document.querySelector(".detail-title").textContent;  let amount = document.querySelector(".input-amout").value;  let idFood = document.querySelector('.idFood').textContent;  let id = 0  let data = { id, title, price, size, img, amount, idFood};  var cart = window.localStorage.getItem("cart");  if (cart === null) {  window.localStorage.setItem("cart", JSON.stringify([data]));  } else {  const listCart = JSON.parse(window.localStorage.getItem("cart"));  listCart.forEach((el) => {    if (el["title"] === data["title"]) {  if (el["size"] === data["size"]) {  const listCartAmount = Number(el["amount"]);  const dataAmount = Number(data["amount"]);  el["amount"] = listCartAmount + dataAmount;  } else {  }  } else {  const cartFake = listCart.find((el2) => {  return el["title"] === el2["title"];  });  if (!cartFake) {  data['id'] = Number(listCart.length);  listCart.push(data);  }  }  });  let dataSize = listCart.find((el) => {  return el["title"] === data["title"] && el["size"] === data["size"];  });  if (!dataSize) {  data['id'] = Number(listCart.length);  listCart.push(data);  }  window.localStorage.setItem("cart", JSON.stringify(listCart));  }  let cartList = JSON.parse(window.localStorage.getItem('cart'));  renderIconCart(cartList)  $('.cart-count').html(cartList.length)  $('.header-top-cart-box').addClass('header-top-modal-show')  /\* ShowSucces();\*/  /\* location.reload();\*/          } else ShowErr("Bạn chưa chọn size")  }); |

Sau đó hiển thị thêm sản phẩm thành công. Nếu add sản phẩm vào mà chưa ấn size sẽ hiển thị thông báo bạn chưa chọn size.

8. Trang giỏ hàng (thêm, giảm, xóa) <JQUERY>

Chúng ta tạo ra function tạo bảng lấy dữ liệu từ localstorage lên

**Script/ cart.js**

|  |
| --- |
| //render giỏ hàng  function renderTable(dataList) {  $(document).ready(() => {  let str = ''  let total = 0  let count = 0;  dataList.forEach((el, index) => {  count++  let intoMoney = Number(el['price']) \* Number(el['amount'])  let bodyCart = `  <div class="cart-product-item">  <img src="${el['img']}" alt="" class="item-img">  <div class="item-info">  <div class="info-heading">  <div class="heading-title">${el['title']}</div>  <div class="heading-close" onclick="remove('${el['title']}','${el['size']}',event)"><img src="//theme.hstatic.net/200000321771/1000699946/14/ic\_close.png?v=305"></div>  </div>  <div class="info-price">${el['price']},000 VND</div>  <div class="info-price">${el['size']}</div>  <div class="info-total">  <div class="total-amount">  <button class="button-amount button-minus" onclick="clickMinus('${el['title']}','${el['size']}',event)"/>-</button>  <input type="number" class="input-amout" readonly min="1" value="${el['amount']}" />  <button class="button-amount button-add" onclick="clickAdd('${el['title']}','${el['size']}',event)">+</button>  </div>  <div class="total-money">  ${intoMoney.toLocaleString()},000 VND  </div>  </div>  </div>  </div>  `  total += intoMoney  str += bodyCart  })  qty.innerHTML = count  if (count > 0) {  totalMoney.innerHTML = total.toLocaleString() + ",000 VND";  $('#renderBody').html(str)  }  else {  totalMoney.innerHTML = "0 VND";  let emp = <div class="empty-cart">Ch?a có s?n ph?m nào tron gi? hàng</div>  $('#renderBody').html(emp);  }  })  }  renderTable(listCart); |

Giảm sản phẩm: Khi người dùng click vào item trừ thì đồng thời sẽ gọi function giảm sản phẩm đầu tiên chúng ta gọi item (cart) lên và so sánh title và size dựa vào tham số truyền vào nếu sản phẩm lớn hơn 1 thì mới được trừ còn không sản phẩm sẽ không được trừ và cập nhật lại tổng tiền của tất cả sản phẩm cập nhật dữ liệu lại vào localstorage.

**Script/ detail.js**

|  |
| --- |
| el["amount"] = listCartAmount + dataAmount;  } else {  }  } else {  const cartFake = listCart.find((el2) => {  return el["title"] === el2["title"];  });  if (!cartFake) {  data['id'] = Number(listCart.length);  listCart.push(data);  }  }  });  let dataSize = listCart.find((el) => {  return el["title"] === data["title"] && el["size"] === data["size"];  });  if (!dataSize) {  data['id'] = Number(listCart.length);  listCart.push(data);  }  window.localStorage.setItem("cart", JSON.stringify(listCart));  }  let cartList = JSON.parse(window.localStorage.getItem('cart'));  renderIconCart(cartList)  $('.cart-count').html(cartList.length)  $('.header-top-cart-box').addClass('header-top-modal-show')  } else ShowErr("Bạn chưa chọn size")  }); |

Thêm sản phẩm: Khi người dùng click vào dấu cộng và truyền vào tham số chúng em sẽ dựa vào vào dữ liệu tên sản phẩm và size để cộng số sản phẩm lên đồng thời cập nhập lại cart.

**Script/ cart.js**

|  |
| --- |
| function clickAdd(title, size, e) {  let cart = JSON.parse(window.localStorage.getItem('cart'));  cart.forEach((el) => {  if (el['title'] === title && el['size'] === size) {  el['amount'] += 1;  }  })  window.localStorage.setItem('cart', JSON.stringify(cart));  renderTable(cart);  } |

Xóa sản phẩm: Khi người dùng click vào dấu x và truyền vào tham số chúng em sẽ dựa vào vào dữ liệu tên sản phẩm và size xóa sản phẩm dùng hàm filter js lọc ra các sản phẩm khác tên sản phẩm sau đó cập nhật lại giỏ hàng.

|  |
| --- |
| function clickMinus(title, size, e) {  let cart = JSON.parse(window.localStorage.getItem('cart'));  cart.forEach((el) => {  if (el['title'] === title && el['size'] === size) {  if (el['amount'] > 1) {  el['amount'] -= 1;  }  else {  alert("Số lượng giảm trên 1")  }  }  })  window.localStorage.setItem('cart', JSON.stringify(cart));  let total = 0  listCart.forEach((el) => {  let intoMoney = Number(el['price']) \* Number(el['amount'])  total += intoMoney  })  totalMoney.innerHTML = total.toLocaleString() + ",000 VND";  renderTable(cart);  } |

9. Thanh toán hóa đơn bill

Front-End: khi người dùng nhập vào thông tin đặt hàng chúng ta sẽ validate của người dùng khi người dùng để trống hay sao thông tin sẽ ràng buộc lại sau đó định dạng sửa lại data phù hợp để post dữ liệu lên backend với controller là bill action result postBill. Nếu dữ liệu backend trả về thành công ta sẽ xóa dữ liệu localstorage và quay trở lại trang home cho người dùng.

**Script/ Checkout.js**

|  |
| --- |
| $('#save\_btn').click(() => {  let checkEmpty = validation.checkRequired([fullname, email, phone, address]);  let checkEmailInvalid = validation.checkEmail(email);  let checkPhoneInvalid = validation.checkNumberPhone(phone);  let checkAddressInvalid = validation.checkAddress([city, district, ward]);  if (  checkEmpty &&  checkEmailInvalid &&  checkPhoneInvalid &&  !checkAddressInvalid  ) {  var dsChiTietDH = []  const idBill = "B" +GenerateId();  let total = 0;  let totalQty = 0;  listCart.forEach((el) => {  let idDetail = uuidv4();  let converNumberAmount = Number(el.amount);  let intoMoney = el.price \* converNumberAmount;  totalQty += Number(el.amount);  total += intoMoney  var ctdh = {  idDetailBill: idDetail, idProduct: el.idFood, idBill: idBill, qty: converNumberAmount, intoMoney: intoMoney  }  dsChiTietDH.push(ctdh);  })  const idUser = null;  const customData = {  idBill: idBill,  idUser: idUser !== null ? idUser : null,  Shipping: 50,  Total: total,  PTTT: "Tien Mat",  status: 0,  detailBill: dsChiTietDH  }  let thanhpho = $("#province option:selected").text()  let quan = $("#village option:selected").text()  let phuong = $("#district option:selected").text()  let diachi = address.value + ', ' + quan + ', ' + phuong + ', ' + thanhpho  let dienthoai = Number(phone.value)  /\* $.post('/Bill/PostBill',"hello", function (res) {  alert(res);  })\*/  $.ajax('/Bill/PostBill', {  data: {  idBill: idBill,  idUser: idUser !== null ? idUser : null,  Shipping: 50,  Total: total,  totalQty: totalQty,  nameBook: fullname.value,  email: email.value,  phone: dienthoai,  address: diachi,  PTTT: "Tien Mat",  detailBill: dsChiTietDH,  status: false,  },  dataType: 'json',  method: 'Post',  success: function (res) {  alert(res);  window.localStorage.removeItem('cart');  window.location.replace('/home')  }  })  }  }) |

– **Back-End:**

+ Function postBill với các tham số dữ liệu quy định để lưu vào db khi frontend gửi tới dữ liệu và đúng với dữ liệu function đã set sẽ tiến hành quá trình biên dịnh luu vào database đầu tiền kiểm tra người dùng có đăng nhập không nếu đăng nhập thì gán giá idUser để lưu vào db cho người dùng.

+ Khởi tạo đối tượng bill và lưu dữ liệu vào db return dữ liệu thông báo thành công.

**BillController.cs**

|  |
| --- |
| [HttpPost]  public ActionResult PostBill(string idBill , string idUser,int Shipping , int Total , int totalQty,string nameBook ,string email , int phone,  string address , string PTTT , DetailBIll [] detailBill ,bool status )  {  var idUserReal = "";  if(Request.Cookies["user"]!=null)  {  idUserReal = Request.Cookies["user"].Value;    }  var bill = new Bill()  {  idBill = idBill,  idUser = idUserReal != "" ? idUserReal : null,  createdAt = DateTime.Now,  Shipping = Shipping,  Total = Total,  totalQty = totalQty,  nameBook = nameBook,  email = email,  phone = Convert.ToString(phone),  address = address,  PTTT = PTTT,  status = status,  DetailBIlls = detailBill ,  };  db.Bills.Add(bill);    db.SaveChanges();  return Json("Đặt hàng thành công");  } |

# CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN TRONG TƯƠNG LAI

### I. Kết luận

Tương đối đầy đủ các chức năng hiển thị sản phẩm theo trang, thêm xóa sửa sản phẩm mới.

* Người dùng dễ dàng xem chi tiết các sản phẩm cần mua.
* Website luôn cập nhật những sản phẩm mới nhất tùy theo nhu cầu người sử dụng.
* Giao diện đơn giản, dễ sử dụng.
* Admin dễ dàng quản lý các sản phẩm.

### II. Hướng phát triển

* Thêm chức năng khuyến mãi khi khách hàng là thành viên VIP, khi đến dịp sinh nhật.
* Thêm chức năng tính phí ship theo khoảng cách.
* Nâng cao thêm chức năng đánh giá và bình luận khi khách hàng mua sản phẩm.
* Thêm chức năng thống kê đơn hàng.
* Thêm chat hỗ trợ khách hàng.
* Thêm thống kê danh thu theo ngày tháng năm.

### III. Phân công công việc

|  |  |  |  |
| --- | --- | --- | --- |
| **Phân công công việc** | | | |
| **Tên thành viên** | **Đặng Chí Bảo** | **Nguyễn Thanh Phong** | **Nguyễn Tuấn Anh** |
| **Back-end** |  |  |  |
| **Front-end** |  |  |  |
| **Database** |  |  |  |
| **Viết báo cáo** |  |  |  |

# CHƯƠNG 6. TÀI LIỆU THAM KHẢO

[1] Setup TailwindCSS in Angular the easy way,

[**https://dev.to/angular/setup-tailwindcss-in-angular-the-easy-way-1i5l**](https://dev.to/angular/setup-tailwindcss-in-angular-the-easy-way-1i5l)

[2] How to use Tailwind CSS in Angular,

**https://www.angularjswiki.com/angular/angular-tailwind-css/**

[3]Angular 12 pagination with NGX Pagination tutorial,

[**https://www.positronx.io/angular-server-side-pagination-with-ngx-pagination-example/**](https://www.positronx.io/angular-server-side-pagination-with-ngx-pagination-example/)

[4]Pagination for angular,

[**http://michaelbromley.github.io/ngx-pagination/#/**](http://michaelbromley.github.io/ngx-pagination/#/)

[5]Component documentation,

**https://www.npmjs.com/package/ngx-dropzone**

**[**6] Bootstrap 4 modal **-** How to Create a Modal,

[**https://www.w3schools.com/bootstrap4/bootstrap\_modal.asp**](https://www.w3schools.com/bootstrap4/bootstrap_modal.asp)

[7] Modal,

[**https://ng-bootstrap.github.io/#/components/modal/examples**](https://ng-bootstrap.github.io/#/components/modal/examples)

[8] What is bootstrap,

[**https://www.w3schools.com/bootstrap4/bootstrap\_get\_started.asp#:~:text=Bootstrap%204%20is%20the%20newest,9%20support%2C%20use%20Bootstrap%203**](https://www.w3schools.com/bootstrap4/bootstrap_get_started.asp#:~:text=Bootstrap%204%20is%20the%20newest,9%20support%2C%20use%20Bootstrap%203)**.**

**----- HẾT -----**