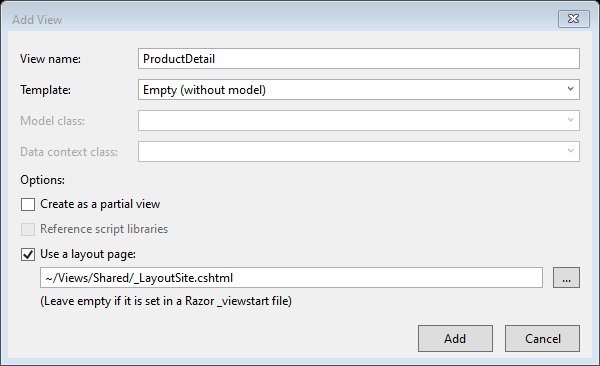
# Xây dựng trang chi tiết sản phẩm

### Tạo mới ProductDetail.cshtml



Sao chép nội dung trang: single-product.html vào đoạn sau của ProductDetail.cshtml

@{

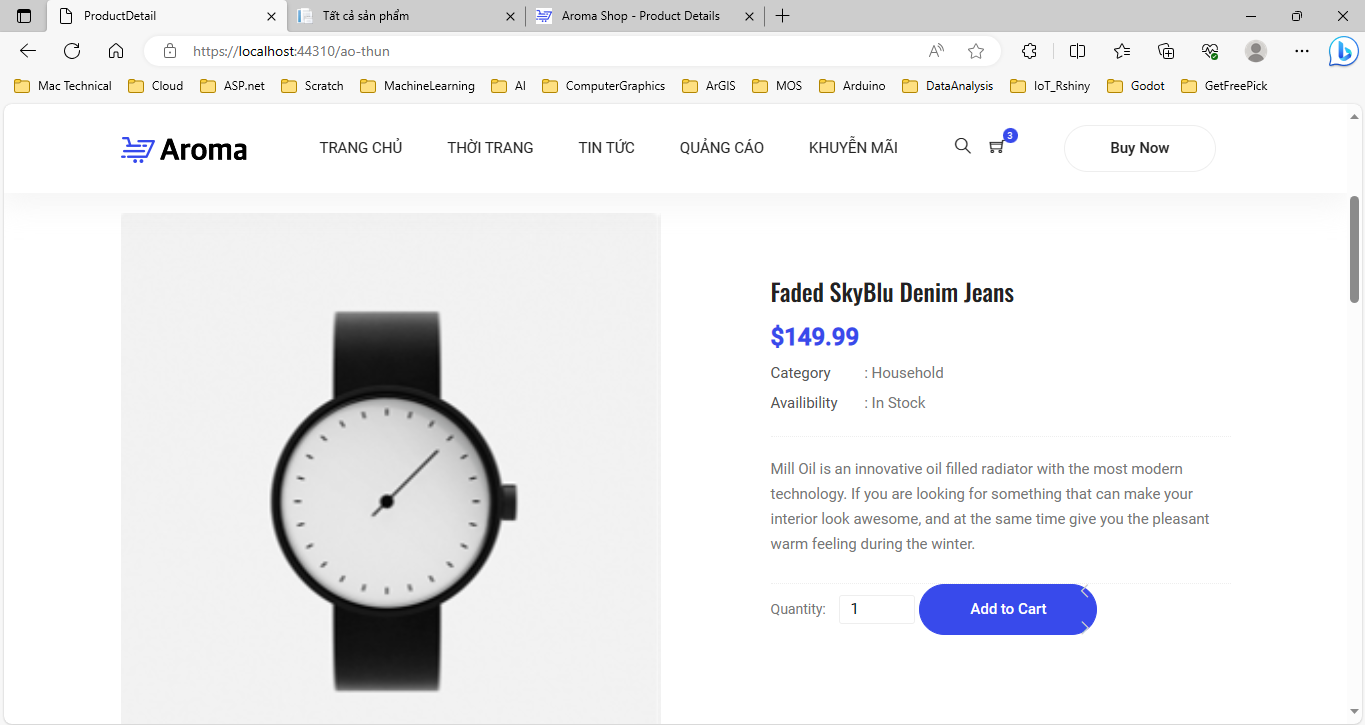
ViewBag.Title = "ProductDetail";

Layout = "~/Views/Shared/\_LayoutSite.cshtml";

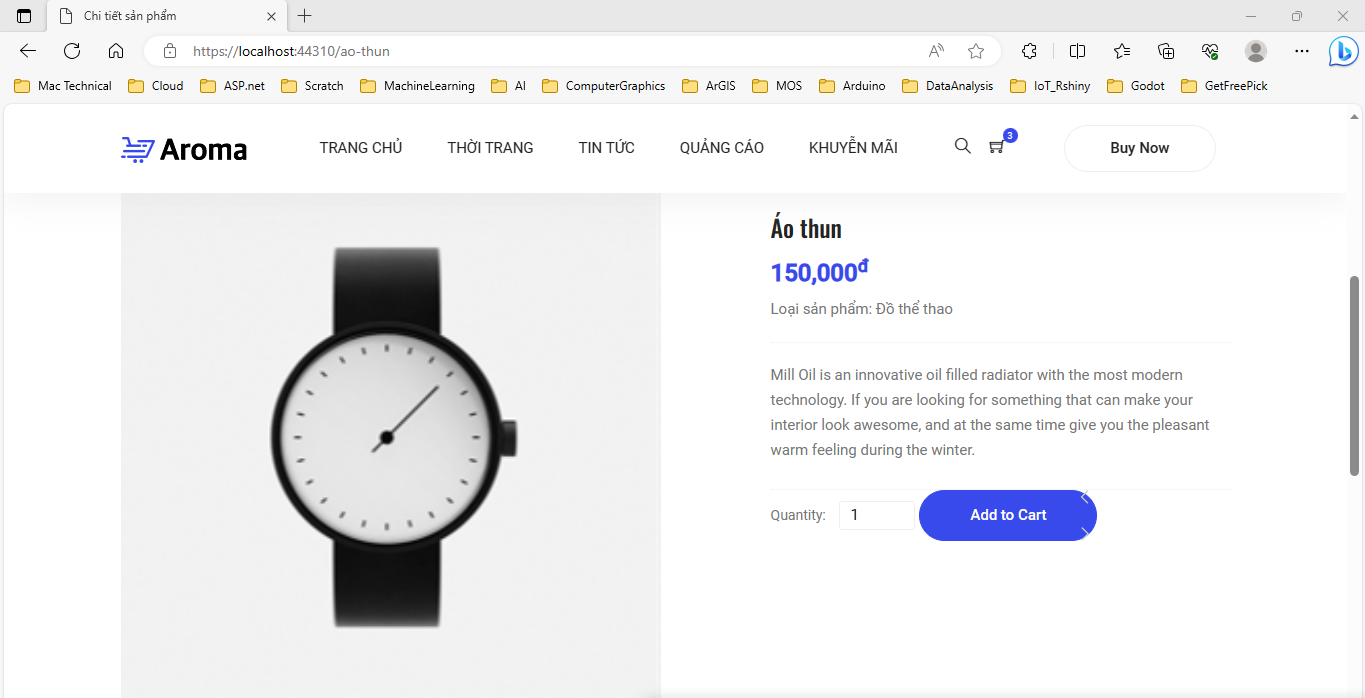
}

#### Copy code here

Tiến hành chỉnh sửa (bỏ qua tiêu đề, footer)



# Thiết lập dữ liệu động



Trên trang này thể hiện Tên sản phẩm (Name), Loại sản phẩm (CatName), Giá bán (Price)… được truy xuất từ bảng dữ liệu ProductInfo. Trong đó có nhiều hơn 1 trường dữ liệu (List) nên trong Controller chúng ta cần trả về 1 List danh sách các trường.

## SiteController

/////////////////////////////////////////////////////////////////////////////

//Product/Details

public ActionResult ProductDetail(string slug[[1]](#footnote-1))

{

// Hiển thị nội dung của sản phẩm

ProductsDAO productsDAO = new ProductsDAO();

List[[2]](#footnote-2)<ProductInfo> list[[3]](#footnote-3) = productsDAO.GetProductDetailBySlug[[4]](#footnote-4)(slug);

return View("ProductDetail", list);

}

## ProductDAO

/////////////////////////////////////////////////////////////////////////////////////

///Chi tiet san pham

///Trang giao dien chi tiet san pham

public List<ProductInfo> GetProductDetailBySlug(string slug)

{

List<ProductInfo> list = null;

list = db.Products

.Where(p => p.Slug == slug && p.Status == 1)

.Join(

db.Categories,

p => p.CatID,

c => c.Id,

(p, c) => new { Product = p, Category = c }

)

.Join(

db.Suppliers,

pc => pc.Product.SupplierId,

s => s.Id,

(pc, s) => new ProductInfo

{

Id = pc.Product.Id,

CatID = pc.Product.CatID,

Name = pc.Product.Name,

CatName = pc.Category.Name,

SupplierId = pc.Product.SupplierId,

SupplierName = s.Name,

Slug = pc.Product.Slug,

Detail = pc.Product.Detail,

Image = pc.Product.Image,

Price = pc.Product.Price,

SalePrice = pc.Product.SalePrice,

Amount = pc.Product.Amount,

MetaDesc = pc.Product.MetaDesc,

MetaKey = pc.Product.MetaKey,

CreateBy = pc.Product.CreateBy,

CreateAt = pc.Product.CreateAt,

UpdateBy = pc.Product.UpdateBy,

UpdateAt = pc.Product.UpdateAt,

Status = pc.Product.Status

}

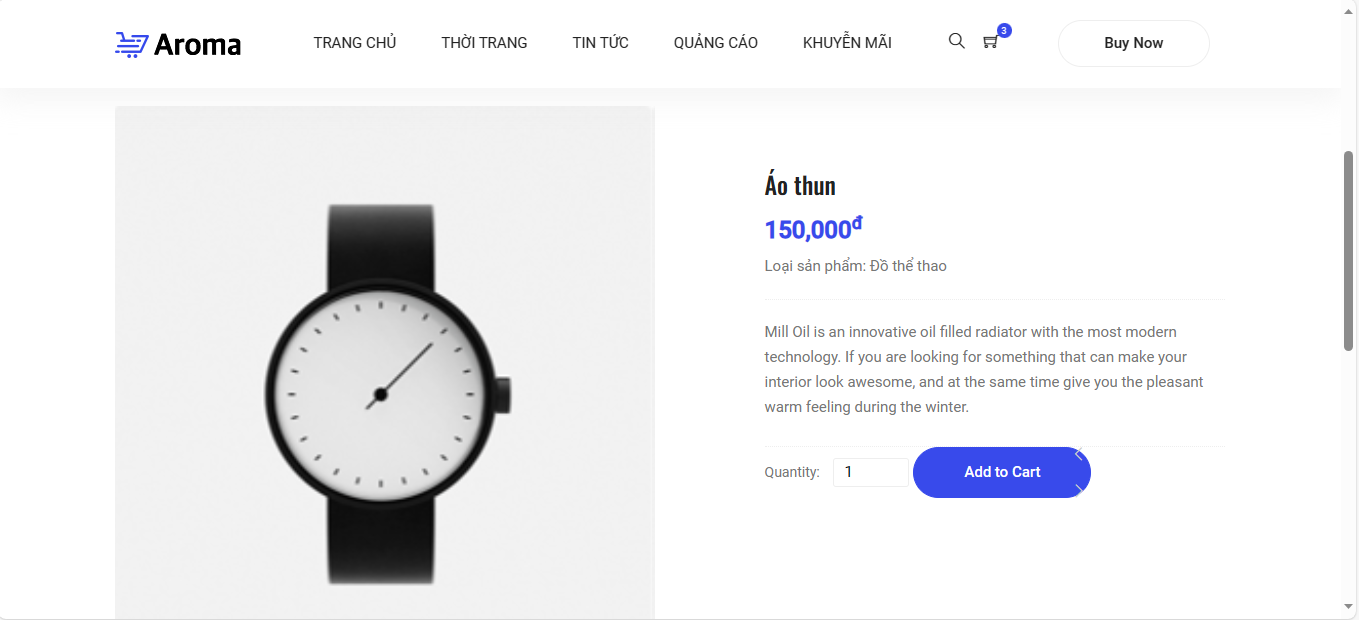
)

.ToList();

return list;

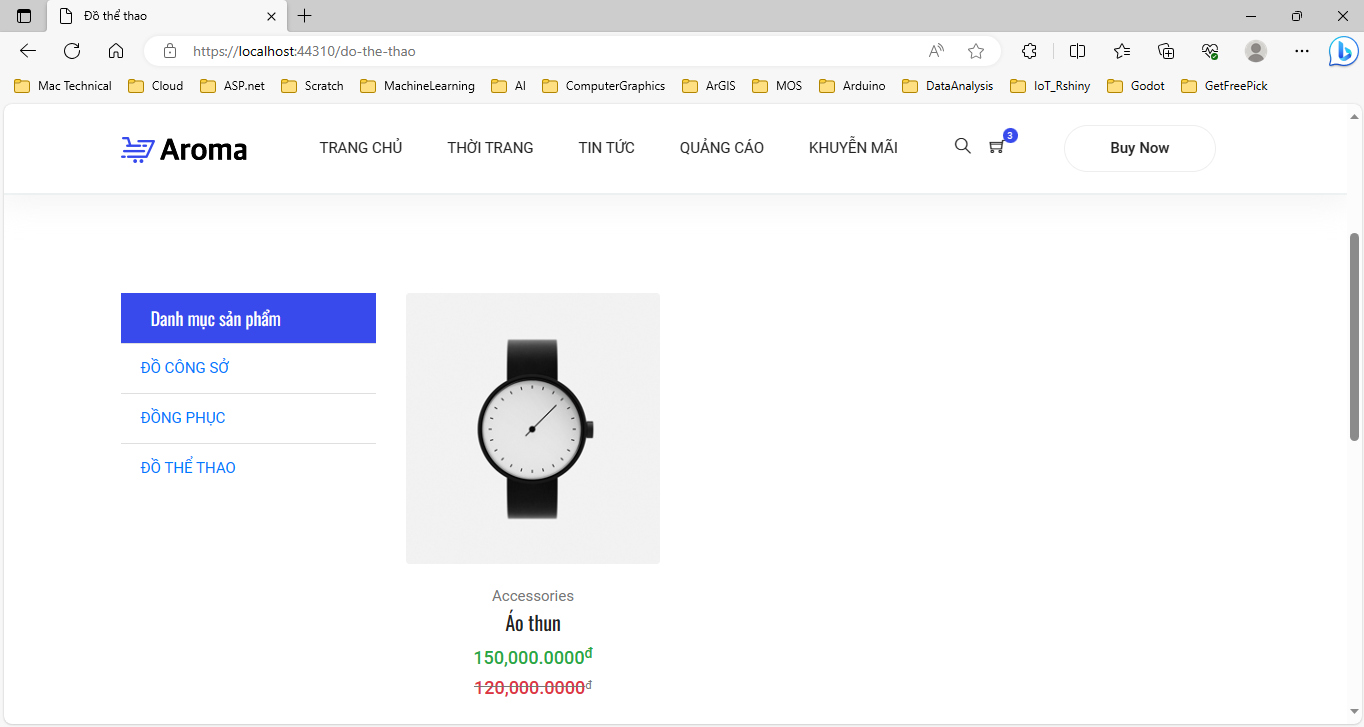
}

# Kết quả



Vấn đề phát sinh tiếp theo, khi Click lên Loại sản phẩm thì trả về link (trang của Categories) tương ứng. Để giải quyết bài toán này chúng ta cần 1 trường CategoriesSlug được trả về từ Model ProductInfo.

### Khi chuyển trang:



# Thực hiện

## ProductInfo.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace MyClass.Model

{

public class ProductInfo

{

public int Id { get; set; }

public int CatID { get; set; }

public string Name { get; set; }

public string CatName { get; set; }

//bo sung them truong Slug cua Categories: detail product

public string CategorySlug { get; set; }

public int SupplierId { get; set; }

public string SupplierName { get; set; }

public string Slug { get; set; }

public string Detail { get; set; }

public string Image { get; set; }

public decimal Price { get; set; }

public decimal SalePrice { get; set; }

public int Amount { get; set; }

public string MetaDesc { get; set; }

public string MetaKey { get; set; }

public int CreateBy { get; set; }

public DateTime CreateAt { get; set; }

public int? UpdateBy { get; set; }

public DateTime? UpdateAt { get; set; }

public int Status { get; set; }

}

}

Từ đó, ta bổ sung trường CategorySlug cho hàm ProductDAO

/////////////////////////////////////////////////////////////////////////////////////

///Chi tiet san pham

///Trang giao dien chi tiet san pham

public List<ProductInfo> GetProductDetailBySlug(string slug)

{

List<ProductInfo> list = null;

list = db.Products

.Where(p => p.Slug == slug && p.Status == 1)

.Join(

db.Categories,

p => p.CatID,

c => c.Id,

(p, c) => new { Product = p, Category = c }

)

.Join(

db.Suppliers,

pc => pc.Product.SupplierId,

s => s.Id,

(pc, s) => new ProductInfo

{

Id = pc.Product.Id,

CatID = pc.Product.CatID,

Name = pc.Product.Name,

CatName = pc.Category.Name,

SupplierId = pc.Product.SupplierId,

SupplierName = s.Name,

Slug = pc.Product.Slug,

Detail = pc.Product.Detail,

Image = pc.Product.Image,

Price = pc.Product.Price,

SalePrice = pc.Product.SalePrice,

Amount = pc.Product.Amount,

MetaDesc = pc.Product.MetaDesc,

MetaKey = pc.Product.MetaKey,

CreateBy = pc.Product.CreateBy,

CreateAt = pc.Product.CreateAt,

UpdateBy = pc.Product.UpdateBy,

UpdateAt = pc.Product.UpdateAt,

Status = pc.Product.Status,

//bo sung truong Slug cua Categories

CategorySlug = pc.Category.Slug

}

)

.ToList();

return list;

}

Gán link liên kết cho ProductDetail.cs

@model IEnumerable<MyClass.Model.ProductInfo>

@{

ViewBag.Title = "Chi tiết sản phẩm";

Layout = "~/Views/Shared/\_LayoutSite.cshtml";

}

<body>

@foreach (var product in Model)

{

<!-- ================ start banner area ================= -->

<section class="blog-banner-area" id="blog">

<div class="container h-100">

<div class="blog-banner">

<div class="text-center">

<h1>Chi tiết sản phẩm</h1>

<nav aria-label="breadcrumb" class="banner-breadcrumb">

<ol class="breadcrumb">

<li class="breadcrumb-item"><a href="@product.CategorySlug[[5]](#footnote-5)">@product.CatName</a></li>

</ol>

</nav>

</div>

</div>

</div>

</section>

<!--================ end banner area =================-->

<!--================Single Product Area =================-->

<div class="product\_image\_area">

<div class="container">

<div class="row s\_product\_inner">

<div class="col-lg-6">

<div class="owl-carousel owl-theme s\_Product\_carousel">

<div class="single-prd-item">

<img class="img-fluid" src="~/Public/img/product/@product.Image" alt="">

</div>

</div>

</div>

<div class="col-lg-5 offset-lg-1">

<div class="s\_product\_text">

<h3>@product.Name</h3>

<h2>@string.Format("{0:0,00}", @product.Price)<sup>đ</sup></h2>

<ul class="list">

<li><a class="active" href="@product.CategorySlug">Loại sản phẩm: @product.CatName</a></li>

</ul>

<p> Mill Oil is an innovative oil filled radiator with the most modern technology. If you are looking for something that can make your interior look awesome, and at the same time give you the pleasant warm feeling during the winter.

</p>

<div class="product\_count">

<label for="qty">Quantity:</label>

<button onclick="var result = document.getElementById('sst'); var sst = result.value; if( !isNaN( sst )) result.value++;return false;"

class="increase items-count" type="button">

<i class="ti-angle-left"></i>

</button>

<input type="text" name="qty" id="sst" size="2" maxlength="12" value="1" title="Quantity:" class="input-text qty">

<button onclick="var result = document.getElementById('sst'); var sst = result.value; if( !isNaN( sst ) &amp;&amp; sst > 0 ) result.value--;return false;" class="reduced items-count" type="button">

<i class="ti-angle-right"></i>

</button>

<a class="button primary-btn" href="#">Add to Cart</a>

</div>

</div>

</div>

</div>

</div>

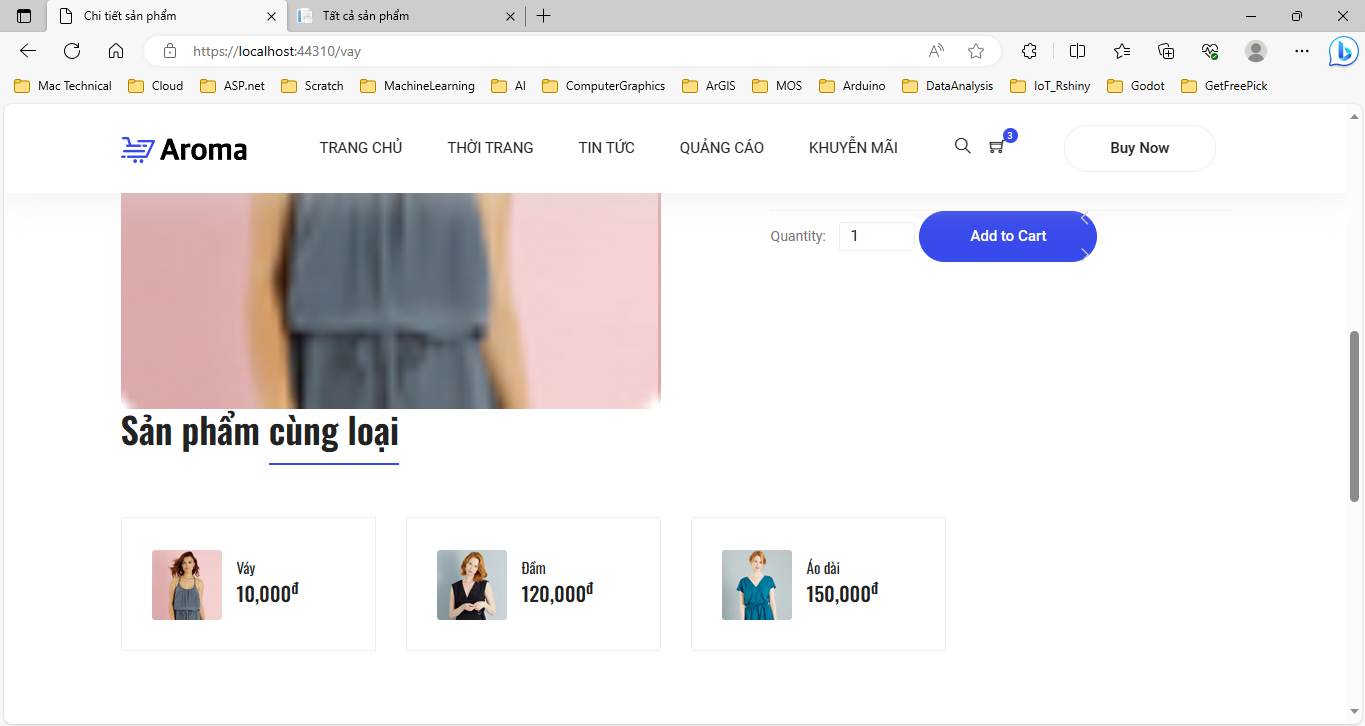
</div>

<!--================End Single Product Area =================-->

}

</body>

# Bổ sung danh sách các sản phẩm cùng loại trong danh mục chi tiết sản phẩm



Để thêm mục sản phẩm cùng loại, trước tiên trong hàm ProductDetail trong SiteController, sau khi bạn đã lấy thông tin sản phẩm cần hiển thị, hãy thực hiện một truy vấn để lấy danh sách sản phẩm cùng loại dựa trên “catid”.

#### SiteController

/////////////////////////////////////////////////////////////////////////////

//Product/Details

public ActionResult ProductDetail(string slug)

{

// Hien thi noi dung cua san pham

ProductsDAO productsDAO = new ProductsDAO();

List<ProductInfo> list = productsDAO.GetProductDetailBySlug(slug);

// Lay CatID cua san pham hien tai

CategoriesDAO categoriesDAO = new CategoriesDAO();

int catid = list.First().CatID;

// Lay danh sach san pham cung loai (related products)

List<ProductInfo> relatedProducts = productsDAO.GetProductDetailByCategoryId(catid);

// Truyen danh sach san pham cung loai cho View

ViewBag.RelatedProducts = relatedProducts;

//tra ve danh sach cho tiet san pham cho list

return View("ProductDetail", list);

}

Hàm GetProductDetailByCategoryId được xây dựng trong lớp ProductsDAO lấy ra danh sách các sản phẩm có cùng "CategoryId" và trả về nó dưới dạng danh sách (List) ProductInfo.

Truyền danh sách sản phẩm cùng loại vào View bằng cách sử dụng ViewBag.

#### ProductDAO

/////////////////////////////////////////////////////////////////////////////////////

///Hien thi danh sach cac san pham cung loai

///Trang giao dien chi tiet san pham

public List<ProductInfo> GetProductDetailByCategoryId(int catid)

{

var relatedProducts = db.Products

.Where(p => p.CatID == catid && p.Status == 1)

.Join(

db.Categories,

p => p.CatID,

c => c.Id,

(p, c) => new { Product = p, Category = c }

)

.Join(

db.Suppliers,

pc => pc.Product.SupplierId,

s => s.Id,

(pc, s) => new ProductInfo

{

Id = pc.Product.Id,

CatID = pc.Product.CatID,

Name = pc.Product.Name,

CatName = pc.Category.Name,

SupplierId = pc.Product.SupplierId,

SupplierName = s.Name,

Slug = pc.Product.Slug,

Detail = pc.Product.Detail,

Image = pc.Product.Image,

Price = pc.Product.Price,

SalePrice = pc.Product.SalePrice,

Amount = pc.Product.Amount,

MetaDesc = pc.Product.MetaDesc,

MetaKey = pc.Product.MetaKey,

CreateBy = pc.Product.CreateBy,

CreateAt = pc.Product.CreateAt,

UpdateBy = pc.Product.UpdateBy,

UpdateAt = pc.Product.UpdateAt,

Status = pc.Product.Status

}

)

.ToList();

return relatedProducts;

}

Trong View (ProductDetails), sử dụng vòng lặp foreach để hiển thị các sản phẩm cùng loại trong thẻ section "related-product-area".

#### ProductDetail.cshtml

@model IEnumerable<MyClass.Model.ProductInfo>

@{

ViewBag.Title = "Chi tiết sản phẩm";

Layout = "~/Views/Shared/\_LayoutSite.cshtml";

}

<body>

@foreach (var product in Model)

{

<!-- ================ start banner area ================= -->

Nội dung chi tiết ở đây bị ẩn

<!--================ end banner area =================-->

<!--================Single Product Area =================-->

Nội dung chi tiết ở đây bị ẩn

<!--================End Single Product Area =================-->

}

<!-- ================Start related Product area =================-->

<section class="related-product-area section-margin--small mt-0">

<div class="container">

<div class="section-intro pb-60px">

<h2>Sản phẩm <span class="section-intro\_\_style">cùng loại</span></h2>

</div>

<div class="row mt-30">

@foreach (var item in ViewBag.RelatedProducts)

{

<div class="col-sm-6 col-xl-3 mb-4 mb-xl-0">

<div class="single-search-product-wrapper">

<div class="single-search-product d-flex">

<a href="#"><img src="~/Public/img/product/@item.Image" alt=""></a>

<div class="desc">

<a href="#" class="title">@item.Name</a>

<h5>@string.Format("{0:0,00}", @item.Price)<sup>đ</sup></h5>

</div>

</div>

</div>

</div>

}

</div>

</div>

</section>

<!--================ end related Product area =================-->

</body>

1. Trong các bài tập trước chỗ này ta trả về Posts posts không thể gắn liên kết nên ta cần chuyển nó thành string [↑](#footnote-ref-1)
2. Kiểu dữ liệu [↑](#footnote-ref-2)
3. Biến: với 2 và 3 thì chúng ta mới có thể thực hiện lời gọi ở trang giao diện: @model IEnumerable<MyClass.Model.ProductInfo> [↑](#footnote-ref-3)
4. Hàm trong ProductsDAO [↑](#footnote-ref-4)
5. CagorySlug: trỏ đến trang Category [↑](#footnote-ref-5)