**TRƯỜNG ĐẠI HỌC NHA TRANG   
KHOA CÔNG NGHỆ THÔNG TIN**



BÁO CÁO ĐỒ ÁN

PHÁT TRIỂN ỨNG DỤNG WEB

Đề Tài : Quản lý cửa hàng bán kẹo ngọt

Giảng viên hướng dẫn : Đoàn Vũ Thịnh, Phạm Thị Kim Ngoan

Sinh viên thực hiện : Trần Khải

Lớp : 63.CNTT 2

MSSV : 63130554

Khánh Hoà 12 tháng 3 năm 2023

**Phụ lục**

[CHHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI 4](#_Toc152537587)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 5](#_Toc152537588)

[CHƯƠNG 3: WEBSITE/ KẾT QUẢ PHÁT TRIỂN WEBSITE 8](#_Toc152537589)

[3.1 Cơ sở dữ liệu: 8](#_Toc152537590)

[3.2 Giao diện UI 9](#_Toc152537591)

[3.3 Danh mục loại sản phẩm: 9](#_Toc152537592)

[3.3.1 Xem chi tiết loại sản phẩm: 9](#_Toc152537593)

[3.3.2 Thay đổi trạng thái loại sản phẩm: 11](#_Toc152537594)

[3.3.3 Tạo mới loại sản phẩm: 12](#_Toc152537595)

[3.3.4 Chỉnh sửa loại sản phẩm: 16](#_Toc152537596)

[3.3.5 Xóa và khôi phục loại sản phẩm: 20](#_Toc152537597)

[3.4 Chức năng danh mục sản phẩm 24](#_Toc152537598)

[3.4.1 Tạo mới sản phẩm 24](#_Toc152537599)

[3.4.2 Xem chi tiết sản phẩm 25](#_Toc152537600)

[3.4.3 Xóa và khôi phục sản phẩm 25](#_Toc152537601)

[3.4.4 Thay đổi trạng thái sản phẩm 27](#_Toc152537602)

[3.4.5 Chỉnh sửa sản phẩm 28](#_Toc152537603)

[3.5 Chức năng nhà cung cấp 30](#_Toc152537604)

[3.5.1 Tạo mới nhà cung cấp 30](#_Toc152537605)

[3.5.2 Xem chi tiết nhà cung cấp 31](#_Toc152537606)

[3.5.3 Xóa và khôi phục nhà cung cấp 32](#_Toc152537607)

[3.5.4 Thay đổi trạng thái nhà cung cấp 34](#_Toc152537608)

[3.5.5 Chỉnh sửa nhà cung cấp 35](#_Toc152537609)

[3.6 Chức năng Menus 37](#_Toc152537610)

[3.6.1 Tạo mới Menus 43](#_Toc152537611)

[3.6.2 Xem chi tiết Menus 43](#_Toc152537612)

[3.6.3 Xóa Menus 44](#_Toc152537613)

[3.6.4 Thay đổi trạng thái Menus 44](#_Toc152537614)

[3.6.5 Chỉnh sửa Menus 45](#_Toc152537615)

[Phần 4: KẾT LUẬN 46](#_Toc152537616)

[1. \*\*Kiến thức về lý thuyết:\*\* 46](#_Toc152537617)

[2. \*\*Kiến thức về thực hành:\*\* 46](#_Toc152537618)

[3. \*\*Xây dựng ứng dụng với các chức năng:\*\* 46](#_Toc152537619)

[4. \*\*Ưu điểm:\*\* 47](#_Toc152537620)

[5. \*\*Hạn chế:\*\* 47](#_Toc152537621)

[6. \*\*Kiến nghị/giải pháp phù hợp:\*\* 47](#_Toc152537622)

[Phần 5 Link github 47](#_Toc152537623)

# CHHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI

* 1. Giới thiệu đề tài :

Trên thị trường hiện nay, đồ nội thất luôn là vật dụng cần thiết đối với mỗi gia đình. Xu hướng thị trường, nhu cầu về nội thất luôn tồn tại và ngày càng tăng cao. Sự kết hợp giữa thiết kế nội thất và công nghệ thông tin có thể mang lại trải nghiệm mua sắm mới và thuận tiện cho người tiêu dùng. Tính hiện đại và tiện ích, sự tích hợp công nghệ vào lĩnh vực nội thất giúp tạo ra các sản phẩm hiện đại, thông minh và tiện ích. Tiếp cận thị trường toàn cầu: Với sự phổ biến của internet, việc sử dụng trang web cho nội thất kết hợp với công nghệ thông tin giúp mở rộng thị trường và tiếp cận khách hàng trên toàn cầu.

* 1. Mục tiêu :

Mục tiêu chính của đề tài là tạo ra một hệ thống quản lý hiệu quả cho cửa hàng bán đồ nội thất. Điều này bao gồm việc áp dụng framework Asp.NET MVC và sử dụng SQL Server để xây dựng cơ sở dữ liệu. Hệ thống sẽ tập trung vào công việc quản lý hàng tồn tại, giao dịch bán hàng và tương tác với khách hàng. Mục tiêu là cải thiện hiệu suất kinh doanh, quản lý quy trình tối ưu hóa và tăng trải nghiệm mua sắm cho khách hàng. Đồng thời, đề tài cũng đặt ra mục tiêu nghiên cứu và áp dụng các phương pháp tiên tiến trong quản lý cửa hàng và phát triển trang web ứng dụng, mang lại giá trị thực tế và bền vững cho doanh nghiệp.

* 1. Giới hạn và phạm vi của đề tài :

Giới Hạn:

- Nghiên cứu sẽ tập trung chủ yếu vào Nghiên cứu cửa hàng bán đồng đồ nội thất và không mở rộng sang các lĩnh vực kinh doanh khác.

- Chúng tôi sẽ không xem xét các yếu tố bên ngoài tác động trực tiếp đến quản lý cửa hàng, giữ cho phạm vi nghiên cứu một cách chặt chẽ.

Phạm Vi:

- Đề tài sẽ xây dựng một hệ thống quản lý trên nền tảng Asp.NET MVC, tập trung vào quản lý hàng tồn tại, giao dịch bán hàng và tương tác với khách hàng.

- Sử dụng cơ sở dữ liệu SQL Server để lưu trữ thông tin cửa hàng.

- Mục tiêu là tạo ra một hệ thống linh hoạt, dễ dàng mở rộng để có thể tích hợp các cải tiến và tính năng mới trong tương lai.

* 1. Yêu cầu đối với website :

Cài đặt chương trình, thể hiện được các chức năng sau:

- Hiện thị danh sách, thêm, sửa và xoá được các bảng dữ liệu trong hệ thống;

- Quản lý được các nghiệp vụ, các chức năng chính của hệ thống;

- Các báo cáo, thống kê, tìm kiếm;

- Phân quyền người dùng;

- Khuyến khích: bảo mật web, deploy lên host, …

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Tổng quan về lập trình web.

Lập trình web là quá trình tạo ra các ứng dụng và trang web trên Internet. Nó liên quan đến việc sử dụng nhiều ngôn ngữ lập trình và công nghệ để phát triển các trang web chức năng, hiệu quả và tương tác. Gồm có:

Ngôn ngữ lập trình:

* HTML (HyperText Markup Language): Ngôn ngữ đánh dấu chính để tạo cấu trúc của trang web.
* CSS (Cascading Style Sheets): Dùng để định dạng và trang trí cho giao diện của trang web.
* JavaScript: Ngôn ngữ lập trình dùng để thêm tính năng tương tác và động cho trang web.

Front-end và Back-end:

* Front-end (Phía trước): Giao diện người dùng và trải nghiệm người dùng. Các công nghệ như React, Angular, hoặc Vue.js thường được sử dụng cho phía front-end.
* Back-end (Phía sau): Xử lý dữ liệu, logic kinh doanh và tương tác với cơ sở dữ liệu. Các ngôn ngữ như Node.js, Python (Django, Flask), Ruby (Ruby on Rails), PHP, Java, và .NET thường được sử dụng cho phía back-end.

Cơ sở dữ liệu:

* SQL (Structured Query Language): Ngôn ngữ truy vấn được sử dụng để tương tác với cơ sở dữ liệu quan hệ như MySQL, PostgreSQL, SQL Server.

Công cụ và Framework:

* Công cụ Phát triển: Visual Studio Code, Sublime Text, Atom là những công cụ phổ biến cho việc viết mã.
* Framework: Đối với front-end, có các framework như React, Angular, Vue.js. Đối với back-end, có Django, Flask (Python), Ruby on Rails (Ruby), Express.js (Node.js), Laravel (PHP).

Responsive Design:

* kế web để phản ánh tốt trên nhiều loại thiết bị và kích thước màn hình khác nhau.

Tích hợp API:

* Sử dụng các API (Application Programming Interface) để kết nối và tương tác với dịch vụ và tài nguyên khác trên Internet.

Bảo mật:

* Quan trọng trong lập trình web để ngăn chặn các loại tấn công như SQL injection, cross-site scripting (XSS), cross-site request forgery (CSRF).

Sử dụng các dịch vụ như Github để đưa ứng dụng web lên Internet.

2.2. Mô hình MVC (Model – View – Controller)

a. Model (Mô hình):

Đại diện cho dữ liệu và logic xử lý dữ liệu.

Thực hiện các thao tác cơ sở dữ liệu và cung cấp dữ liệu cho Controller.

b. View (Giao diện):

Hiển thị dữ liệu từ Model cho người dùng.

Thường chỉ chịu trách nhiệm hiển thị và không thay đổi dữ liệu.

c. Controller (Điều khiển):

Chịu trách nhiệm xử lý các yêu cầu từ người dùng, thay đổi dữ liệu trong Model và cập nhật Giao diện.

Tương tác với Model để nhận và gửi dữ liệu.

2.3. ASP.NET MVC

Là một framework của Microsoft dựa trên mô hình MVC.

Hỗ trợ phát triển ứng dụng web linh hoạt và dễ bảo trì.

Các thành phần chính của ASP.NET MVC:

* Model: Các lớp chứa dữ liệu và logic xử lý dữ liệu.
* View: Các file chứa giao diện người dùng, thường là các file Razor (.cshtml).
* Controller: Các lớp chứa logic điều khiển, xử lý yêu cầu từ người dùng.
* Routing: ASP.NET MVC sử dụng hệ thống định tuyến (Routing) để ánh xạ các yêu cầu từ người dùng với các phương thức trong Controller.

Các kỹ thuật hiển thị dữ liệu: Sử dụng Razor View Engine để tạo giao diện người dùng dễ đọc và dễ hiểu.

Tích hợp với Entity Framework: Cung cấp tích hợp tốt với Entity Framework, một ORM (Object-Relational Mapping) giúp thao tác với cơ sở dữ liệu.

2.4. Hệ quản trị cơ sở dữ liệu SQL sever.

SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ do Microsoft phát triển.

Hỗ trợ lưu trữ và truy vấn dữ liệu theo mô hình quan hệ.

Tính năng của SQL Server:

* Ngôn ngữ SQL: Sử dụng ngôn ngữ truy vấn SQL để tương tác với cơ sở dữ liệu.
* StoredProcedure và Trigger: Cho phép xử lý logic phức tạp trực tiếp trên cơ sở dữ liệu.

Transact-SQL (T-SQL): Một phần mở rộng của SQL dành cho SQL Server, hỗ trợ các thủ tục lưu trữ và nhiều tính năng khác.

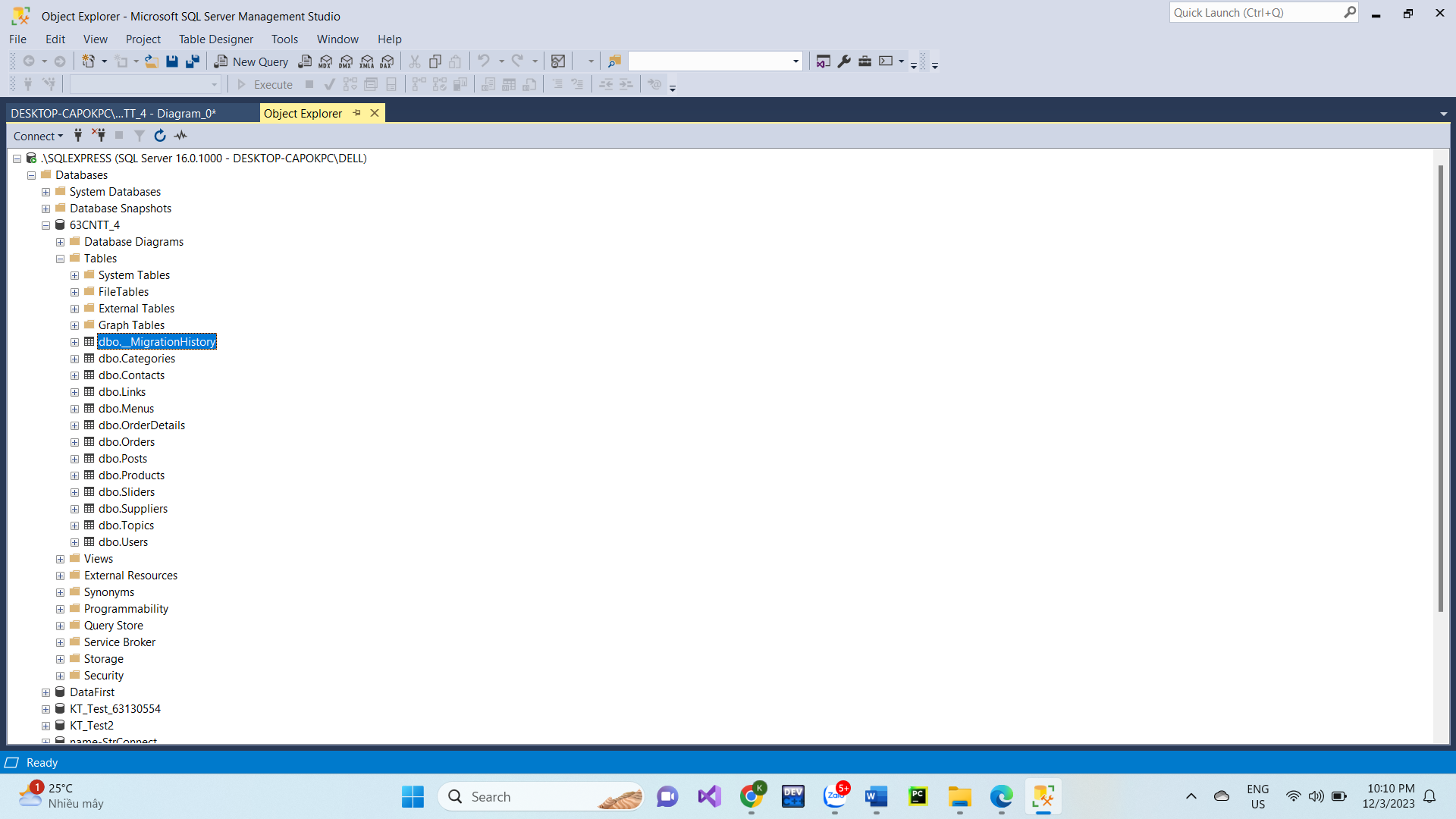
Entity Framework và SQL Server:

* Entity Framework là một ORM cho phép thao tác với cơ sở dữ liệu thông qua các đối tượng .NET, giảm độ phức tạp của việc làm việc trực tiếp với SQL.

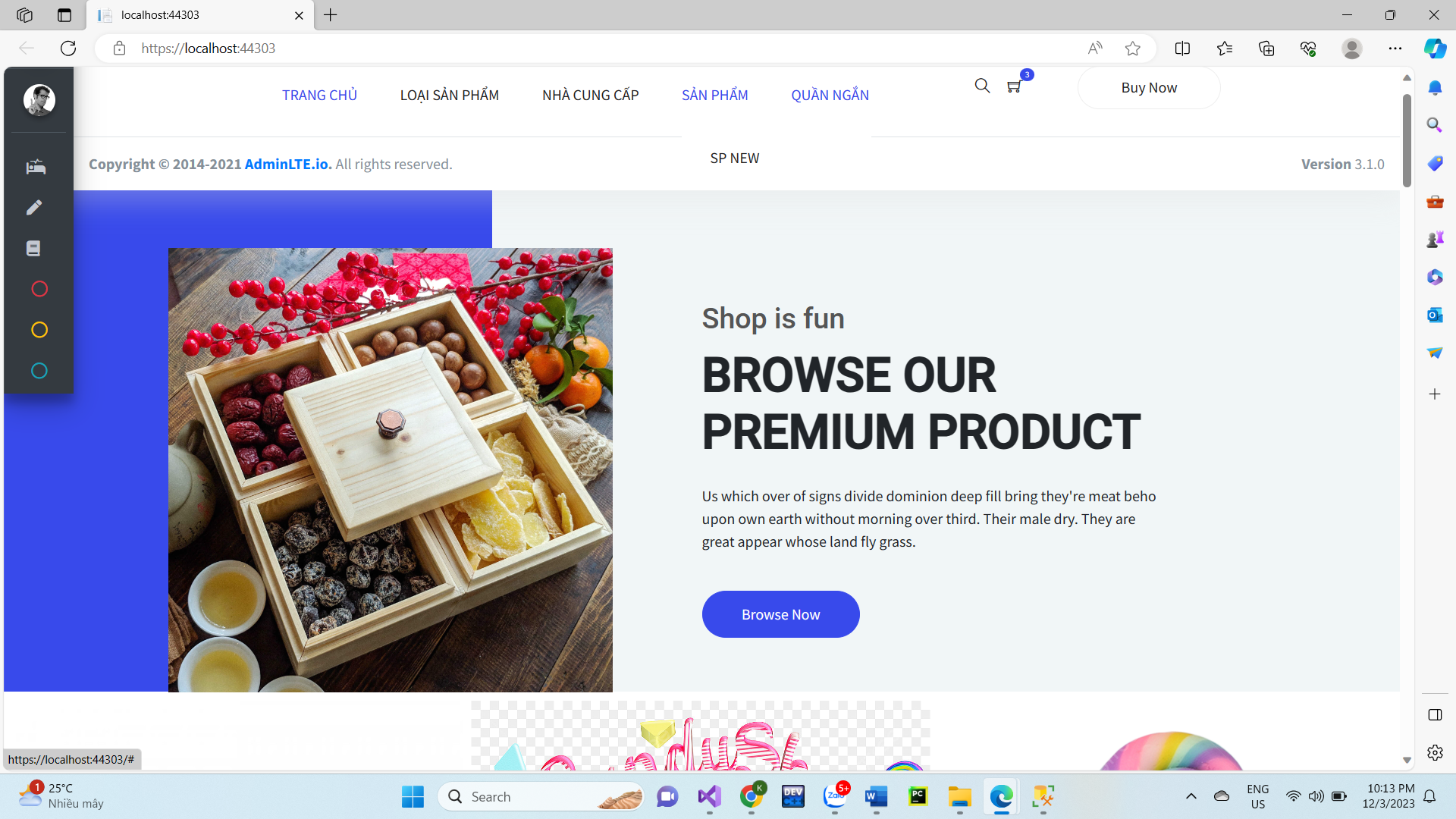
ASP.NET MVC thường sử dụng Entity Framework để tương tác với cơ sở dữ liệu SQL Server.

# CHƯƠNG 3: WEBSITE/ KẾT QUẢ PHÁT TRIỂN WEBSITE

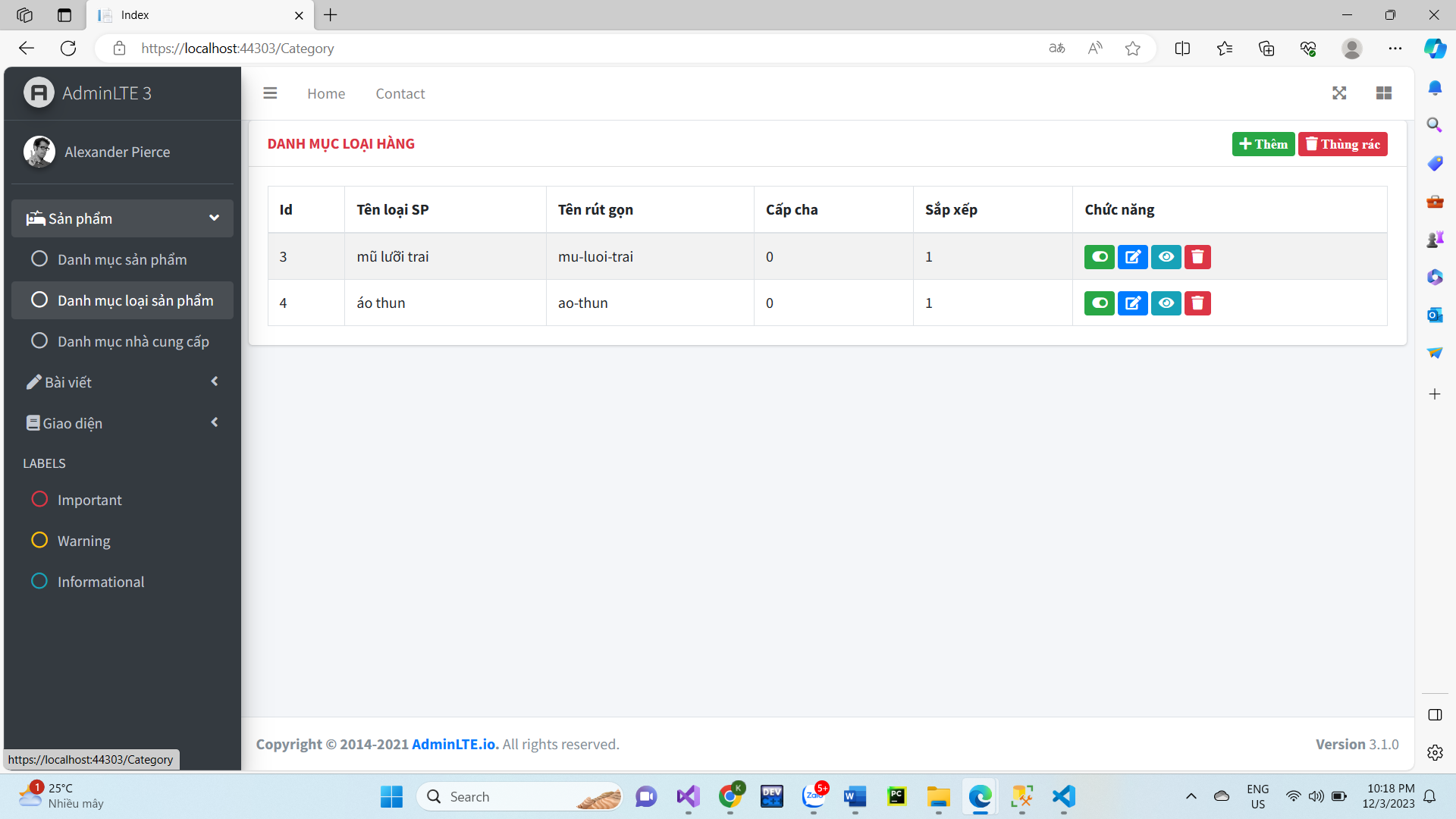
## 3.1 Cơ sở dữ liệu:



## 3.2 Giao diện UI



## 3.3 Danh mục loại sản phẩm:



### 3.3.1 Xem chi tiết loại sản phẩm:

Controller

public ActionResult Details(int? id)

        {

            if (id == null)

            {

                //thong bao that bai

                TempData["message"] = new XMessage("danger", "Không tồn tại loại sản phẩm");

                return RedirectToAction("Index");

            }

            Categories categories = categoriesDAO.getRow(id);

            if (categories == null)

            {

                TempData["message"] = new XMessage("danger", "Không tồn tại loại sản phẩm");

                return RedirectToAction("Index");

            }

            return View(categories);

        }

View

@model MyClass.Model.Categories

@{

    ViewBag.Title = "Chi tiết loại sản phẩm";

    Layout = "~/Areas/Admin/Views/Shared/\_LayoutAdmin.cshtml";

}

<div class="wrapper">

    <!-- Main Sidebar Container -->

    <!-- Content Wrapper. Contains page content -->

    <div class="content-wrapper">

        <!-- Main content -->

        <section class="content">

            <!-- Default box -->

            <div class="card">

                <div class="card-header">

                    <div class="row">

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

                            <strong class="text-danger">DANH MỤC LOẠI HÀNG</strong>

                        </div>

                        <div class="col-md-6 text-right">

                            <a href="@Url.Action("Edit","Category", new { id = Model.Id })" class="btn-sm btn-success"><i class="fa fa-edit"> Sửa</i></a>

                            <a href="@Url.Action("Index","Category")" class="btn-sm btn-info"><i class="fa fa-long-arrow-alt-left"> Quay lại</i></a>

                        </div>

                    </div>

                </div>

                <div class="card-body">

                    <table class="table table-bordered table-striped">

                        <thead>

                            <tr>

                                <th>@Html.DisplayNameFor(model => model.Id)</th>

                                <th>@Html.DisplayNameFor(model => model.Name)</th>

                                <th>@Html.DisplayNameFor(model => model.Slug)</th>

                                <th>@Html.DisplayNameFor(model => model.ParentId)</th>

                                <th>@Html.DisplayNameFor(model => model.Order)</th>

                                <th>@Html.DisplayNameFor(model => model.CreateAt)</th>

                                <th>@Html.DisplayNameFor(model => model.UpdateAt)</th>

                            </tr>

                        </thead>

                        <tbody>

                            <tr>

                                <td>@Html.DisplayFor(model => model.Id)</td>

                                <td>@Html.DisplayFor(model => model.Name)</td>

                                <td>@Html.DisplayFor(model => model.Slug)</td>

                                <td>@Html.DisplayFor(model => model.ParentId)</td>

                                <td>@Html.DisplayFor(model => model.Order)</td>

                                <td>@Html.DisplayFor(model => model.CreateAt)</td>

                                <td>@Html.DisplayFor(model => model.UpdateAt)</td>

                            </tr>

                        </tbody>

                    </table>

                </div> <!-- /.card-footer-->

            </div><!-- /.card -->

        </section><!-- /.content -->

    </div> <!-- /.content-wrapper -->

</div><!-- ./wrapper -->

### 3.3.2 Thay đổi trạng thái loại sản phẩm:

Controller

 public ActionResult Status(int? id)

        {

            if (id == null)

            {

                //thong bao that bai

                TempData["message"] = new XMessage("danger", "Cập nhật trạng thái thất bại");

                return RedirectToAction("Index");

            }

            //truy van dong co id = id yeu cau

            Categories categories = categoriesDAO.getRow(id);

            if (categories == null)

            {

                //thong bao that bai

                TempData["message"] = new XMessage("danger", "Cập nhật trạng thái thất bại");

                return RedirectToAction("Index");

            }

            else

            {

                //chuyen doi trang thai cua Satus tu 1<->2

                categories.Status = (categories.Status == 1) ? 2 : 1;

                //cap nhat gia tri UpdateAt

                categories.UpdateAt = DateTime.Now;

                //cap nhat lai DB

                categoriesDAO.Update(categories);

                //thong bao cap nhat trang thai thanh cong

                TempData["message"] = TempData["message"] = new XMessage("success", "Cập nhật trạng thái thành công");

                return RedirectToAction("Index");

            }

        }

View

### 3.3.3 Tạo mới loại sản phẩm:

Controller

 public ActionResult Create()

        {

            ViewBag.ListCat = new SelectList(categoriesDAO.getList("Index"), "Id", "Name");

            ViewBag.ListOrder = new SelectList(categoriesDAO.getList("Index"), "Order", "Name");

            return View();

        }

        [HttpPost]

        [ValidateAntiForgeryToken]

        public ActionResult Create(Categories categories)

        {

            if (ModelState.IsValid)

            {

                //Xu ly tu dong: CreateAt

                categories.CreateAt = DateTime.Now;

                //Xu ly tu dong: UpdateAt

                categories.UpdateAt = DateTime.Now;

                //Xu ly tu dong: ParentId

                if (categories.ParentId == null)

                {

                    categories.ParentId = 0;

                }

                //Xu ly tu dong: Order

                if (categories.Order == null)

                {

                    categories.Order = 1;

                }

                else

                {

                    categories.Order += 1;

                }

                //Xu ly tu dong: Slug

                categories.Slug = XString.Str\_Slug(categories.Name);

                //Chen them dong cho DB

                categoriesDAO.Insert(categories);

                //thong bao cap nhat trang thai thanh cong

                TempData["message"] = TempData["message"] = new XMessage("success", "Tạo mới loại sản phẩm thành công");

                //tro ve trang index

                return RedirectToAction("Index");

            }

            ViewBag.ListCat = new SelectList(categoriesDAO.getList("Index"), "Id", "Name");

            ViewBag.ListOrder = new SelectList(categoriesDAO.getList("Index"), "Order", "Name");

            return View(categories);

        }

View

@model MyClass.Model.Categories

@{

    ViewBag.Title = "Tạo mới loại sản phẩm";

    Layout = "~/Areas/Admin/Views/Shared/\_LayoutAdmin.cshtml";

}

@using (Html.BeginForm())

{

    @Html.AntiForgeryToken()

    <div class="wrapper">

        <!-- Main Sidebar Container -->

        <!-- Content Wrapper. Contains page content -->

        <div class="content-wrapper">

            <!-- Main content -->

            <section class="content">

                <!-- Default box -->

                <div class="card">

                    <div class="card-header">

                        <div class="row">

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

                                <strong class="text-danger">TẠO MỚI LOẠI HÀNG</strong>

                            </div>

                            <div class="col-md-6 text-right">

                                <a href="@Url.Action("Index","Category")" class="btn-sm btn-info"><i class="fa fa-long-arrow-alt-left"> Quay lại</i></a>

                            </div>

                        </div>

                    </div>

                    <div class="card-body">

                        <div class="row">

                            @\*Phần nhập dữ liệu bằng tay\*@

                            <div class="col-md-9">

                                @Html.ValidationSummary(true, "", new { @class = "text-danger" })

                                <div class="form-group">

                                    @Html.LabelFor(model => model.Name, htmlAttributes: new { @class = "control-label col-md-2" })

                                    <div class="col-md-10">

                                        @Html.EditorFor(model => model.Name, new { htmlAttributes = new { @class = "form-control" } })

                                        @Html.ValidationMessageFor(model => model.Name, "", new { @class = "text-danger" })

                                    </div>

                                </div>

                                <div class="form-group">

                                    @Html.LabelFor(model => model.MetaDesc, htmlAttributes: new { @class = "control-label col-md-2" })

                                    <div class="col-md-10">

                                        @Html.TextAreaFor(model => model.MetaDesc, new { @class = "form-control", @row = "5" })

                                        @Html.ValidationMessageFor(model => model.MetaDesc, "", new { @class = "text-danger" })

                                    </div>

                                </div>

                                <div class="form-group">

                                    @Html.LabelFor(model => model.MetaKey, htmlAttributes: new { @class = "control-label col-md-2" })

                                    <div class="col-md-10">

                                        @Html.TextAreaFor(model => model.MetaKey, new { @class = "form-control", @row = "5" })

                                        @Html.ValidationMessageFor(model => model.MetaKey, "", new { @class = "text-danger" })

                                    </div>

                                </div>

                            </div>

                            @\*Phần lựa chọn dữ liệu\*@

                            <div class="col-md-3">

                                <div class="form-group">

                                    @Html.LabelFor(model => model.ParentId, htmlAttributes: new { @class = "control-label col-md-12" })

                                    <div class="col-md-10">

                                        @Html.DropDownListFor(model => model.ParentId, (SelectList)ViewBag.CatList, "Cấp 0", new { @class = "form-control" })

                                        @Html.ValidationMessageFor(model => model.ParentId, "", new { @class = "text-danger" })

                                    </div>

                                </div>

                                <div class="form-group">

                                    @Html.LabelFor(model => model.Order, htmlAttributes: new { @class = "control-label col-md-12" })

                                    <div class="col-md-10">

                                        @Html.DropDownListFor(model => model.Order, (SelectList)ViewBag.OrderList, "Sắp xếp", new { @class = "form-control" })

                                        @Html.ValidationMessageFor(model => model.Order, "", new { @class = "text-danger" })

                                    </div>

                                </div>

                                <div class="form-group">

                                    @Html.LabelFor(model => model.Status, htmlAttributes: new { @class = "control-label col-md-12" })

                                    <div class="col-md-10">

                                        @Html.DropDownListFor(model => model.Status, new[] { new SelectListItem { Text = "Không xuất bản", Value = "2" }, new SelectListItem { Text = "Xuất bản", Value = "1" } }, new { @class = "form-control" })

                                        @Html.ValidationMessageFor(model => model.Status, "", new { @class = "text-danger" })

                                    </div>

                                </div>

                            </div>

                        </div>

                        @\*Sua nut Create\*@

                        <button class="btn-sm btn-success" type="submit" name="create">[Lưu] Thêm <i class="fa fa-save"></i></button>

                    </div> <!-- /.card-footer-->

                </div><!-- /.card -->

            </section><!-- /.content -->

        </div> <!-- /.content-wrapper -->

    </div><!-- ./wrapper -->

}

### 3.3.4 Chỉnh sửa loại sản phẩm:

Controller

 public ActionResult Edit(int? id)

        {

            if (id == null)

            {

                //thong bao that bai

                TempData["message"] = TempData["message"] = new XMessage("danger", "Không tìm thấy mẩu tin");

                return RedirectToAction("Index");

            }

            Categories categories = categoriesDAO.getRow(id);

            if (categories == null)

            {

                //thong bao that bai

                TempData["message"] = TempData["message"] = new XMessage("danger", "Không tìm thấy mẩu tin");

                return RedirectToAction("Index");

            }

            ViewBag.ListCat = new SelectList(categoriesDAO.getList("Index"), "Id", "Name");

            ViewBag.ListOrder = new SelectList(categoriesDAO.getList("Index"), "Order", "Name");

            return View(categories);

        }

        [HttpPost]

        [ValidateAntiForgeryToken]

        public ActionResult Edit(Categories categories)

        {

            if (ModelState.IsValid)

            {

                //xu ly tu dong: Slug

                categories.Slug = XString.Str\_Slug(categories.Name);

                //xu ly tu dong: ParentID

                if (categories.ParentId == null)

                {

                    categories.ParentId = 0;

                }

                //xu ly tu dong: Order

                if (categories.Order == null)

                {

                    categories.Order = 1;

                }

                else

                {

                    categories.Order += 1;

                }

                //xu ly tu dong: UpdateAt

                categories.UpdateAt = DateTime.Now;

                //cap nhat mau tin

                categoriesDAO.Update(categories);

                //thong bao thanh cong

                TempData["message"] = TempData["message"] = new XMessage("success", "Cập nhật mẩu tin thành công");

                return RedirectToAction("Index");

            }

            ViewBag.ListCat = new SelectList(categoriesDAO.getList("Index"), "Id", "Name");

            ViewBag.ListOrder = new SelectList(categoriesDAO.getList("Index"), "Order", "Name");

            return View(categories);

        }

View

@model MyClass.Model.Categories

@{

    ViewBag.Title = "Thay đổi thông tin loại hàng";

    Layout = "~/Areas/Admin/Views/Shared/\_LayoutAdmin.cshtml";

}

@using (Html.BeginForm())

{

    @Html.AntiForgeryToken()

    @Html.ValidationSummary(true, "", new { @class = "text-danger" })

    @Html.HiddenFor(model => model.Id)

    @Html.HiddenFor(model => model.CreateAt)

    @Html.HiddenFor(model => model.CreateBy)

    <div class="wrapper">

        <!-- Main Sidebar Container -->

        <!-- Content Wrapper. Contains page content -->

        <div class="content-wrapper">

            <!-- Main content -->

            <section class="content">

                <!-- Default box -->

                <div class="card">

                    <div class="card-header">

                        <div class="row">

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

                                <strong class="text-danger">TẠO MỚI LOẠI HÀNG</strong>

                            </div>

                            <div class="col-md-6 text-right">

                                <a href="@Url.Action("Index","Category")" class="btn-sm btn-info"><i class="fa fa-long-arrow-alt-left"> Quay lại</i></a>

                            </div>

                        </div>

                    </div>

                    <div class="card-body">

                        <div class="row">

                            @\*Phần nhập dữ liệu bằng tay\*@

                            <div class="col-md-9">

                                @Html.ValidationSummary(true, "", new { @class = "text-danger" })

                                <div class="form-group">

                                    @Html.LabelFor(model => model.Name, htmlAttributes: new { @class = "control-label col-md-2" })

                                    <div class="col-md-10">

                                        @Html.EditorFor(model => model.Name, new { htmlAttributes = new { @class = "form-control" } })

                                        @Html.ValidationMessageFor(model => model.Name, "", new { @class = "text-danger" })

                                    </div>

                                </div>

                                <div class="form-group">

                                    @Html.LabelFor(model => model.MetaDesc, htmlAttributes: new { @class = "control-label col-md-2" })

                                    <div class="col-md-10">

                                        @Html.TextAreaFor(model => model.MetaDesc, new { @class = "form-control", @row = "5" })

                                        @Html.ValidationMessageFor(model => model.MetaDesc, "", new { @class = "text-danger" })

                                    </div>

                                </div>

                                <div class="form-group">

                                    @Html.LabelFor(model => model.MetaKey, htmlAttributes: new { @class = "control-label col-md-2" })

                                    <div class="col-md-10">

                                        @Html.TextAreaFor(model => model.MetaKey, new { @class = "form-control", @row = "5" })

                                        @Html.ValidationMessageFor(model => model.MetaKey, "", new { @class = "text-danger" })

                                    </div>

                                </div>

                            </div>

                            @\*Phần lựa chọn dữ liệu\*@

                            <div class="col-md-3">

                                <div class="form-group">

                                    @Html.LabelFor(model => model.ParentId, htmlAttributes: new { @class = "control-label col-md-12" })

                                    <div class="col-md-10">

                                        @Html.DropDownListFor(model => model.ParentId, (SelectList)ViewBag.CatList, "Cấp 0", new { @class = "form-control" })

                                        @Html.ValidationMessageFor(model => model.ParentId, "", new { @class = "text-danger" })

                                    </div>

                                </div>

                                <div class="form-group">

                                    @Html.LabelFor(model => model.Order, htmlAttributes: new { @class = "control-label col-md-12" })

                                    <div class="col-md-10">

                                        @Html.DropDownListFor(model => model.Order, (SelectList)ViewBag.OrderList, "Sắp xếp", new { @class = "form-control" })

                                        @Html.ValidationMessageFor(model => model.Order, "", new { @class = "text-danger" })

                                    </div>

                                </div>

                                <div class="form-group">

                                    @Html.LabelFor(model => model.Status, htmlAttributes: new { @class = "control-label col-md-12" })

                                    <div class="col-md-10">

                                        @Html.DropDownListFor(model => model.Status, new[] { new SelectListItem { Text = "Không xuất bản", Value = "2" }, new SelectListItem { Text = "Xuất bản", Value = "1" } }, new { @class = "form-control" })

                                        @Html.ValidationMessageFor(model => model.Status, "", new { @class = "text-danger" })

                                    </div>

                                </div>

                            </div>

                        </div>

                        @\*Sua nut Edit\*@

                        <button class="btn-sm btn-success" type="submit" name="create">[Lưu] Thêm <i class="fa fa-save"></i></button>

                    </div> <!-- /.card-footer-->

                </div><!-- /.card -->

            </section><!-- /.content -->

        </div> <!-- /.content-wrapper -->

    </div><!-- ./wrapper -->

}

### 3.3.5 Xóa và khôi phục loại sản phẩm:

Xóa:

Controller

 public ActionResult Delete(int? id)

        {

            if (id == null)

            {

                //thong bao that bai

                TempData["message"] = new XMessage("danger", "Xóa mẩu tin thất bại");

                return RedirectToAction("Index");

            }

            Categories categories = categoriesDAO.getRow(id);

            if (categories == null)

            {

                //thong bao that bai

                TempData["message"] = new XMessage("danger", "Xóa mẩu tin thất bại");

                return RedirectToAction("Index");

            }

            return View(categories);

        }

        // POST: Admin/Category/Delete/5

        [HttpPost, ActionName("Delete")]

        [ValidateAntiForgeryToken]

        public ActionResult DeleteConfirmed(int id)

        {

            Categories categories = categoriesDAO.getRow(id);

            categoriesDAO.Delete(categories);

            //thong bao thanh cong

            TempData["message"] = new XMessage("success", "Xóa mẩu tin thành công");

            return RedirectToAction("Trash");

        }

View

@model MyClass.Model.Categories

@{

    ViewBag.Title = "Xác nhận xóa mẩu tin";

    Layout = "~/Areas/Admin/Views/Shared/\_LayoutAdmin.cshtml";

}

@using (Html.BeginForm())

{

    @Html.AntiForgeryToken()

    <div class="wrapper">

        <!-- Main Sidebar Container -->

        <!-- Content Wrapper. Contains page content -->

        <div class="content-wrapper">

            <!-- Main content -->

            <section class="content">

                <!-- Default box -->

                <div class="card">

                    <div class="card-header">

                        <div class="row">

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

                                <strong class="text-danger">XÁC NHẬN XÓA MẨU TIN</strong>

                            </div>

                            <div class="col-md-6 text-right">

                                <a href="@Url.Action("Index","Category")" class="btn-sm btn-info"><i class="fa fa-long-arrow-alt-left"> Quay lại</i></a>

                            </div>

                        </div>

                    </div>

                    <div class="card-body">

                        <table class="table table-bordered table-striped">

                            <thead>

                                <tr>

                                    <th>@Html.DisplayNameFor(model => model.Id)</th>

                                    <th>@Html.DisplayNameFor(model => model.Name)</th>

                                    <th>@Html.DisplayNameFor(model => model.Slug)</th>

                                    <th>@Html.DisplayNameFor(model => model.ParentId)</th>

                                    <th>@Html.DisplayNameFor(model => model.Order)</th>

                                    <th>@Html.DisplayNameFor(model => model.CreateAt)</th>

                                    <th>@Html.DisplayNameFor(model => model.UpdateAt)</th>

                                </tr>

                            </thead>

                            <tbody>

                                <tr>

                                    <td>@Html.DisplayFor(model => model.Id)</td>

                                    <td>@Html.DisplayFor(model => model.Name)</td>

                                    <td>@Html.DisplayFor(model => model.Slug)</td>

                                    <td>@Html.DisplayFor(model => model.ParentId)</td>

                                    <td>@Html.DisplayFor(model => model.Order)</td>

                                    <td>@Html.DisplayFor(model => model.CreateAt)</td>

                                    <td>@Html.DisplayFor(model => model.UpdateAt)</td>

                                </tr>

                            </tbody>

                        </table>

                    // NÚT

                        <div class="form-actions no-color">

                            <button class="btn-sm btn-danger" type="submit" name="save">  <i class="fa fa-trash"> Xóa </i></button>

                        </div>

                    </div> <!-- /.card-footer-->

                </div><!-- /.card -->

            </section><!-- /.content -->

        </div> <!-- /.content-wrapper -->

    </div><!-- ./wrapper -->

}

Khôi phục:

 public ActionResult DelTrash(int? id)

        {

            if (id == null)

            {

                //thong bao that bai

                TempData["message"] = new XMessage("danger", "Không tìm thấy mẩu tin");

                return RedirectToAction("Index");

            }

            //truy van dong co id = id yeu cau

            Categories categories = categoriesDAO.getRow(id);

            if (categories == null)

            {

                //thong bao that bai

                TempData["message"] = new XMessage("danger", "Không tìm thấy mẩu tin");

                return RedirectToAction("Index");

            }

            else

            {

                //chuyen doi trang thai cua Satus tu 1,2 -> 0: Không hiển thị ở Index

                categories.Status = 0;

                //cap nhat gia tri UpdateAt

                categories.UpdateAt = DateTime.Now;

                //cap nhat lai DB

                categoriesDAO.Update(categories);

                //thong bao cap nhat trang thai thanh cong

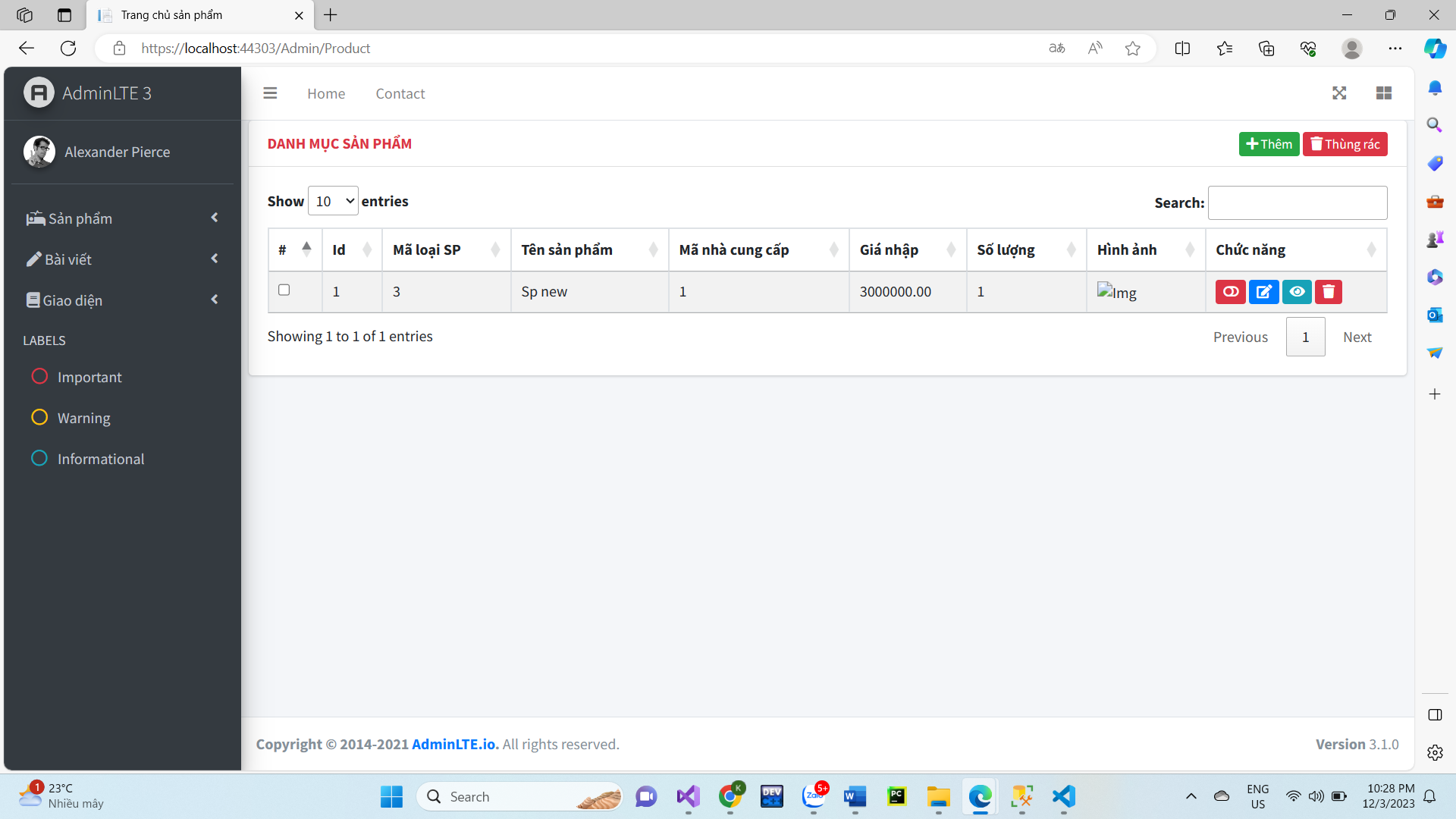
                TempData["message"] = TempData["message"] = new XMessage("success", "Xóa mẩu tin thành công");

                return RedirectToAction("Index");

            }

        }

## 3.4 Chức năng danh mục sản phẩm



### 3.4.1 Tạo mới sản phẩm

public ActionResult Create()

        {

            ViewBag.ListCatID = new SelectList(categoriesDAO.getList("Index"), "Id", "Name");//sai CatId - truy van tu bang Categories

            ViewBag.ListSupID = new SelectList(suppliersDAO.getList("Index"), "Id", "Name");//sai SupplierID - truy van bang Suppliers

            //dung de lua chon tu danh sach droplist nhu bang Categories: ParentID va Supplier: ParentID

            return View();

        }

        [HttpPost]

        [ValidateAntiForgeryToken]

        public ActionResult Create(Products products)

        {

            if (ModelState.IsValid)

            {

                //xu ly thong tin tu dong cho mot so truong

                //Xu ly tu dong: CreateAt

                products.CreateAt = DateTime.Now;

                //Xu ly tu dong: UpdateAt

                products.UpdateAt = DateTime.Now;

                //Xu ly tu dong: CreateBy

                products.CreateBy = Convert.ToInt32(Session["UserId"]);

                //Xu ly tu dong: UpdateBy

                products.UpdateBy = Convert.ToInt32(Session["UserId"]);

                //Xu ly tu dong: Slug

                products.Slug = XString.Str\_Slug(products.Name);

                //xu ly cho phan upload hinh anh

                var img = Request.Files["img"];//lay thong tin file

                if (img.ContentLength != 0)

                {

                    string[] FileExtentions = new string[] { ".jpg", ".jpeg", ".png", ".gif" };

                    //kiem tra tap tin co hay khong

                    if (FileExtentions.Contains(img.FileName.Substring(img.FileName.LastIndexOf("."))))//lay phan mo rong cua tap tin

                    {

                        string slug = products.Slug;

                        //ten file = Slug + phan mo rong cua tap tin

                        string imgName = slug + img.FileName.Substring(img.FileName.LastIndexOf("."));

                        products.Image = imgName;

                        //upload hinh

                        string PathDir = "~/Public/img/product";

                        string PathFile = Path.Combine(Server.MapPath(PathDir), imgName);

                        img.SaveAs(PathFile);

                    }

                }//ket thuc phan upload hinh anh

                //luu vao DB

                productsDAO.Insert(products);

                //thong bao tao mau tin thanh cong

                TempData["message"] = new XMessage("success", "Tạo mới sản phẩm thành công");

                return RedirectToAction("Index");

            }

            ViewBag.ListCatID = new SelectList(categoriesDAO.getList("Index"), "Id", "Name");//sai CatId - truy van tu bang Categories

            ViewBag.ListSupID = new SelectList(suppliersDAO.getList("Index"), "Id", "Name");//sai SupplierID - truy van bang Suppliers

            return View(products);

        }

### 3.4.2 Xem chi tiết sản phẩm

### 3.4.3 Xóa và khôi phục sản phẩm

Xóa:

 public ActionResult Delete(int? id)

        {

            if (id == null)

            {

                //thong bao that bai

                TempData["message"] = new XMessage("danger", "Không tồn tại sản phẩm");

                return RedirectToAction("Index");

            }

            Products products = productsDAO.getRow(id);

            if (products == null)

            {

                //thong bao that bai

                TempData["message"] = new XMessage("danger", "Không tồn tại sản phẩm");

                return RedirectToAction("Index");

            }

            return View(products);

        }

        // POST: Admin/Product/Delete/5

        [HttpPost, ActionName("Delete")]

        [ValidateAntiForgeryToken]

        public ActionResult DeleteConfirmed(int id)

        {

            Products products = productsDAO.getRow(id);

            productsDAO.Delete(products);

            //thong bao xoa mau tin thanh cong

            TempData["message"] = new XMessage("success", "Xóa sản phẩm thành công");

            return RedirectToAction("Trash");

        }

Khôi phục:  
public ActionResult DelTrash(int? id)

        {

            if (id == null)

            {

                //thong bao that bai

                TempData["message"] = new XMessage("danger", "Không tìm thấy mẩu tin");

                return RedirectToAction("Index");

            }

            //truy van dong co id = id yeu cau

            Products products = productsDAO.getRow(id);

            if (products == null)

            {

                //thong bao that bai

                TempData["message"] = new XMessage("danger", "Không tìm thấy mẩu tin");

                return RedirectToAction("Index");

            }

            else

            {

                //chuyen doi trang thai cua Satus tu 1,2 -> 0: Không hiển thị ở Index

                products.Status = 0;

                //cap nhat gia tri UpdateAt

                products.UpdateAt = DateTime.Now;

                //cap nhat lai DB

                productsDAO.Update(products);

                //thong bao cap nhat trang thai thanh cong

                TempData["message"] = TempData["message"] = new XMessage("success", "Xóa mẩu tin vào thùng rác thành công");

                return RedirectToAction("Index");

            }

        }

### 3.4.4 Thay đổi trạng thái sản phẩm

 public ActionResult Status(int? id)

        {

            if (id == null)

            {

                //thong bao that bai

                TempData["message"] = new XMessage("danger", "Cập nhật trạng thái thất bại");

                return RedirectToAction("Index");

            }

            //truy van dong co id = id yeu cau

            Products products = productsDAO.getRow(id);

            if (products == null)

            {

                //thong bao that bai

                TempData["message"] = new XMessage("danger", "Cập nhật trạng thái thất bại");

                return RedirectToAction("Index");

            }

            else

            {

                //chuyen doi trang thai cua Satus tu 1<->2

                products.Status = (products.Status == 1) ? 2 : 1;

                //cap nhat gia tri UpdateAt

                products.UpdateAt = DateTime.Now;

                //cap nhat lai DB

                productsDAO.Update(products);

                //thong bao cap nhat trang thai thanh cong

                TempData["message"] = TempData["message"] = new XMessage("success", "Cập nhật trạng thái thành công");

                return RedirectToAction("Index");

            }

        }

### 3.4.5 Chỉnh sửa sản phẩm

 public ActionResult Edit(int? id)

        {

            if (id == null)

            {

                //thong bao that bai

                TempData["message"] = new XMessage("danger", "Không tồn tại sản phẩm");

                return RedirectToAction("Index");

            }

            Products products = productsDAO.getRow(id);

            if (products == null)

            {

                //thong bao that bai

                TempData["message"] = new XMessage("danger", "Không tồn tại sản phẩm");

                return RedirectToAction("Index");

            }

            return View(products);

        }

        [HttpPost]

        [ValidateAntiForgeryToken]

        public ActionResult Edit(Products products)

        {

            if (ModelState.IsValid)

            {

                //xu ly tu dong cho cac truong: Slug, CreateAt/By, UpdateAt/By, Oder

                //Xu ly tu dong: UpdateAt

                products.UpdateAt = DateTime.Now;

                //Xu ly tu dong: Slug

                products.Slug = XString.Str\_Slug(products.Name);

                //xu ly cho phan upload hinh anh

                var img = Request.Files["img"];//lay thong tin file

                string PathDir = "~/Public/img/product";

                if (img.ContentLength != 0)

                {

                    //Xu ly cho muc xoa hinh anh

                    if (products.Image != null)

                    {

                        string DelPath = Path.Combine(Server.MapPath(PathDir), products.Image);

                        System.IO.File.Delete(DelPath);

                    }

                    string[] FileExtentions = new string[] { ".jpg", ".jpeg", ".png", ".gif" };

                    //kiem tra tap tin co hay khong

                    if (FileExtentions.Contains(img.FileName.Substring(img.FileName.LastIndexOf("."))))//lay phan mo rong cua tap tin

                    {

                        string slug = products.Slug;

                        //ten file = Slug + phan mo rong cua tap tin

                        string imgName = slug + img.FileName.Substring(img.FileName.LastIndexOf("."));

                        products.Image = imgName;

                        //upload hinh

                        string PathFile = Path.Combine(Server.MapPath(PathDir), imgName);

                        img.SaveAs(PathFile);

                    }

                }//ket thuc phan upload hinh anh

                //cap nhat mau tin vao DB

                productsDAO.Update(products);

                //thong bao tao mau tin thanh cong

                TempData["message"] = new XMessage("success", "Cập nhật sản phẩm thành công");

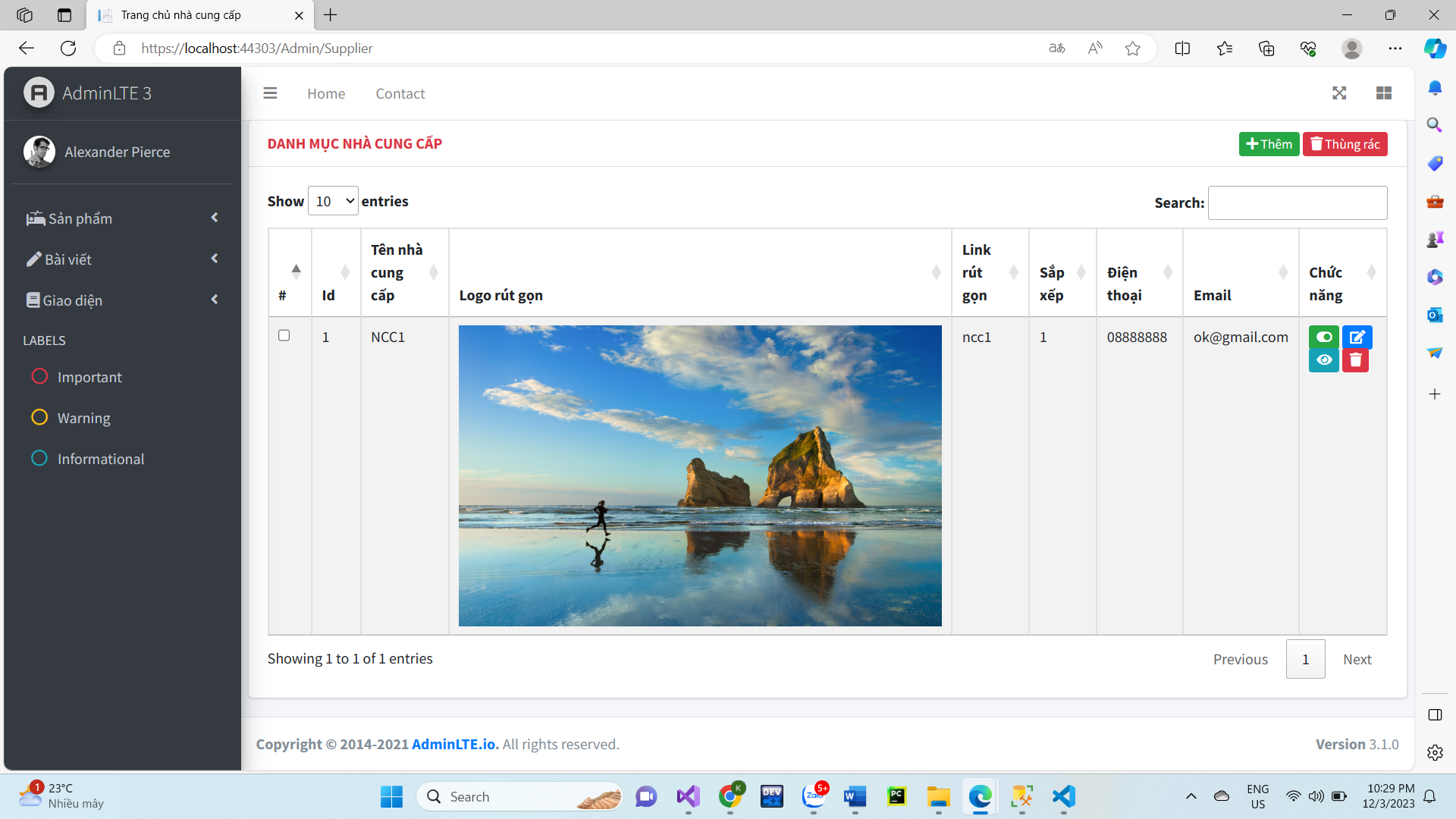
                return RedirectToAction("Index");

            }

            return View(products);

        }

## 3.5 Chức năng nhà cung cấp



### 3.5.1 Tạo mới nhà cung cấp

 public ActionResult Create()

        {

            ViewBag.ListOrder = new SelectList(suppliersDAO.getList("Index"), "Order", "Name");

            return View();

        }

        [HttpPost]

        [ValidateAntiForgeryToken]

        public ActionResult Create(Suppliers suppliers)

        {

            if (ModelState.IsValid)

            {

                //xu ly tu dong cho cac truong: Slug, CreateAt/By, UpdateAt/By, Oder

                //Xu ly tu dong: CreateAt

                suppliers.CreateAt = DateTime.Now;

                //Xu ly tu dong: UpdateAt

                suppliers.UpdateAt = DateTime.Now;

                //Xu ly tu dong: CreateBy

                suppliers.CreateBy = Convert.ToInt32(Session["UserId"]);

                //Xu ly tu dong: UpdateBy

                suppliers.UpdateBy = Convert.ToInt32(Session["UserId"]);

                //Xu ly tu dong: Order

                if (suppliers.Order == null)

                {

                    suppliers.Order = 1;

                }

                else

                {

                    suppliers.Order += 1;

                }

                //Xu ly tu dong: Slug

                suppliers.Slug = XString.Str\_Slug(suppliers.Name);

                //xu ly cho phan upload hinh anh

                var img = Request.Files["img"];//lay thong tin file

                if (img.ContentLength != 0)

                {

                    string[] FileExtentions = new string[] { ".jpg", ".jpeg", ".png", ".gif" };

                    //kiem tra tap tin co hay khong

                    if (FileExtentions.Contains(img.FileName.Substring(img.FileName.LastIndexOf("."))))//lay phan mo rong cua tap tin

                    {

                        string slug = suppliers.Slug;

                        //ten file = Slug + phan mo rong cua tap tin

                        string imgName = slug + img.FileName.Substring(img.FileName.LastIndexOf("."));

                        suppliers.Image = imgName;

                        //upload hinh

                        string PathDir = "~/Public/img/supplier";

                        string PathFile = Path.Combine(Server.MapPath(PathDir), imgName);

                        img.SaveAs(PathFile);

                    }

                }//ket thuc phan upload hinh anh

                //chen mau tin vao DB

                suppliersDAO.Insert(suppliers);

                //thong bao tao mau tin thanh cong

                TempData["message"] = new XMessage("success", "Tạo mới nhà cung cấp thành công");

                return RedirectToAction("Index");

            }

            ViewBag.ListOrder = new SelectList(suppliersDAO.getList("Index"), "Order", "Name");

            return View(suppliers);

        }

### 3.5.2 Xem chi tiết nhà cung cấp

 public ActionResult Details(int? id)

        {

            if (id == null)

            {

                //thong bao that bai

                TempData["message"] = new XMessage("danger", "Không tồn tại nhà cung cấp");

                return RedirectToAction("Index");

            }

            Suppliers suppliers = suppliersDAO.getRow(id);

            if (suppliers == null)

            {

                //thong bao that bai

                TempData["message"] = new XMessage("danger", "Không tồn tại nhà cung cấp");

                return RedirectToAction("Index");

            }

            return View(suppliers);

        }

### 3.5.3 Xóa và khôi phục nhà cung cấp

 public ActionResult Delete(int? id)

        {

            if (id == null)

            {

                //thong bao that bai

                TempData["message"] = new XMessage("danger", "Không tồn tại nhà cung cấp");

            }

            Suppliers suppliers = suppliersDAO.getRow(id);

            if (suppliers == null)

            {

                //thong bao that bai

                TempData["message"] = new XMessage("danger", "Không tồn tại nhà cung cấp");

            }

            return View(suppliers);

        }

        // POST: Admin/Supplier/Delete/5

        [HttpPost, ActionName("Delete")]

        [ValidateAntiForgeryToken]

        public ActionResult DeleteConfirmed(int id)

        {

            Suppliers suppliers = suppliersDAO.getRow(id);

            //xu ly cho phan upload hinh anh

            var img = Request.Files["img"];//lay thong tin file

            string PathDir = "~/Public/img/supplier";

            //xoa mau tin ra khoi DB

            if (suppliersDAO.Delete(suppliers) == 1)

            {

                //Xu ly cho muc xoa hinh anh

                if (suppliers.Image != null)

                {

                    string DelPath = Path.Combine(Server.MapPath(PathDir), suppliers.Image);

                    System.IO.File.Delete(DelPath);

                }

            }

            //thong bao xoa mau tin thanh cong

            TempData["message"] = new XMessage("success", "Xóa nhà cung cấp thành công");

            return RedirectToAction("Trash");

        }

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

        //DELTRAH

        // GET: Admin/Supplier/DelTrash/5

        public ActionResult DelTrash(int? id)

        {

            if (id == null)

            {

                //thong bao that bai

                TempData["message"] = new XMessage("danger", "Không tìm thấy mẩu tin");

                return RedirectToAction("Index");

            }

            //truy van dong co id = id yeu cau

            Suppliers suppliers = suppliersDAO.getRow(id);

            if (suppliers == null)

            {

                //thong bao that bai

                TempData["message"] = new XMessage("danger", "Không tìm thấy mẩu tin");

                return RedirectToAction("Index");

            }

            else

            {

                //chuyen doi trang thai cua Satus tu 1,2 -> 0: Không hiển thị ở Index

                suppliers.Status = 0;

                //cap nhat gia tri UpdateAt

                suppliers.UpdateAt = DateTime.Now;

                //cap nhat lai DB

                suppliersDAO.Update(suppliers);

                //thong bao cap nhat trang thai thanh cong

                TempData["message"] = TempData["message"] = new XMessage("success", "Xóa mẩu tin thành công");

                return RedirectToAction("Index");

            }

        }

### 3.5.4 Thay đổi trạng thái nhà cung cấp

//phat sinh them 1 so action: Status, Trash, DelTrash, Undo: Copy tu Category Controller

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

        //STATUS

        // GET: Admin/Supplier/Status/5

        public ActionResult Status(int? id)

        {

            if (id == null)

            {

                //thong bao that bai

                TempData["message"] = new XMessage("danger", "Cập nhật trạng thái thất bại");

                return RedirectToAction("Index");

            }

            //truy van dong co id = id yeu cau

            Suppliers suppliers = suppliersDAO.getRow(id);

            if (suppliers == null)

            {

                //thong bao that bai

                TempData["message"] = new XMessage("danger", "Cập nhật trạng thái thất bại");

                return RedirectToAction("Index");

            }

            else

            {

                //chuyen doi trang thai cua Satus tu 1<->2

                suppliers.Status = (suppliers.Status == 1) ? 2 : 1;

                //cap nhat gia tri UpdateAt

                suppliers.UpdateAt = DateTime.Now;

                //cap nhat lai DB

                suppliersDAO.Update(suppliers);

                //thong bao cap nhat trang thai thanh cong

                TempData["message"] = TempData["message"] = new XMessage("success", "Cập nhật trạng thái thành công");

                return RedirectToAction("Index");

            }

        }

### 3.5.5 Chỉnh sửa nhà cung cấp

 public ActionResult Edit(int? id)

        {

            if (id == null)

            {

                //thong bao that bai

                TempData["message"] = new XMessage("danger", "Không tồn tại nhà cung cấp");

                return RedirectToAction("Index");

            }

            Suppliers suppliers = suppliersDAO.getRow(id);

            if (suppliers == null)

            {

                //thong bao that bai

                TempData["message"] = new XMessage("danger", "Không tồn tại nhà cung cấp");

                return RedirectToAction("Index");

            }

            ViewBag.ListOrder = new SelectList(suppliersDAO.getList("Index"), "Order", "Name");

            return View(suppliers);

        }

        [HttpPost]

        [ValidateAntiForgeryToken]

        public ActionResult Edit(Suppliers suppliers)

        {

            if (ModelState.IsValid)

            {

                //xu ly tu dong cho cac truong: Slug, CreateAt/By, UpdateAt/By, Oder

                //Xu ly tu dong: UpdateAt

                suppliers.UpdateAt = DateTime.Now;

                //Xu ly tu dong: Order

                if (suppliers.Order == null)

                {

                    suppliers.Order = 1;

                }

                else

                {

                    suppliers.Order += 1;

                }

                //Xu ly tu dong: Slug

                suppliers.Slug = XString.Str\_Slug(suppliers.Name);

                //xu ly cho phan upload hinh anh

                var img = Request.Files["img"];//lay thong tin file

                string PathDir = "~/Public/img/supplier";

                if (img.ContentLength != 0)

                {

                    //Xu ly cho muc xoa hinh anh

                    if (suppliers.Image != null)

                    {

                        string DelPath = Path.Combine(Server.MapPath(PathDir), suppliers.Image);

                        System.IO.File.Delete(DelPath);

                    }

                    string[] FileExtentions = new string[] { ".jpg", ".jpeg", ".png", ".gif" };

                    //kiem tra tap tin co hay khong

                    if (FileExtentions.Contains(img.FileName.Substring(img.FileName.LastIndexOf("."))))//lay phan mo rong cua tap tin

                    {

                        string slug = suppliers.Slug;

                        //ten file = Slug + phan mo rong cua tap tin

                        string imgName = slug + img.FileName.Substring(img.FileName.LastIndexOf("."));

                        suppliers.Image = imgName;

                        //upload hinh

                        string PathFile = Path.Combine(Server.MapPath(PathDir), imgName);

                        img.SaveAs(PathFile);

                    }

                }//ket thuc phan upload hinh anh

                //cap nhat mau tin vao DB

                suppliersDAO.Update(suppliers);

                //thong bao tao mau tin thanh cong

                TempData["message"] = new XMessage("success", "Cập nhật nhà cung cấp thành công");

                return RedirectToAction("Index");

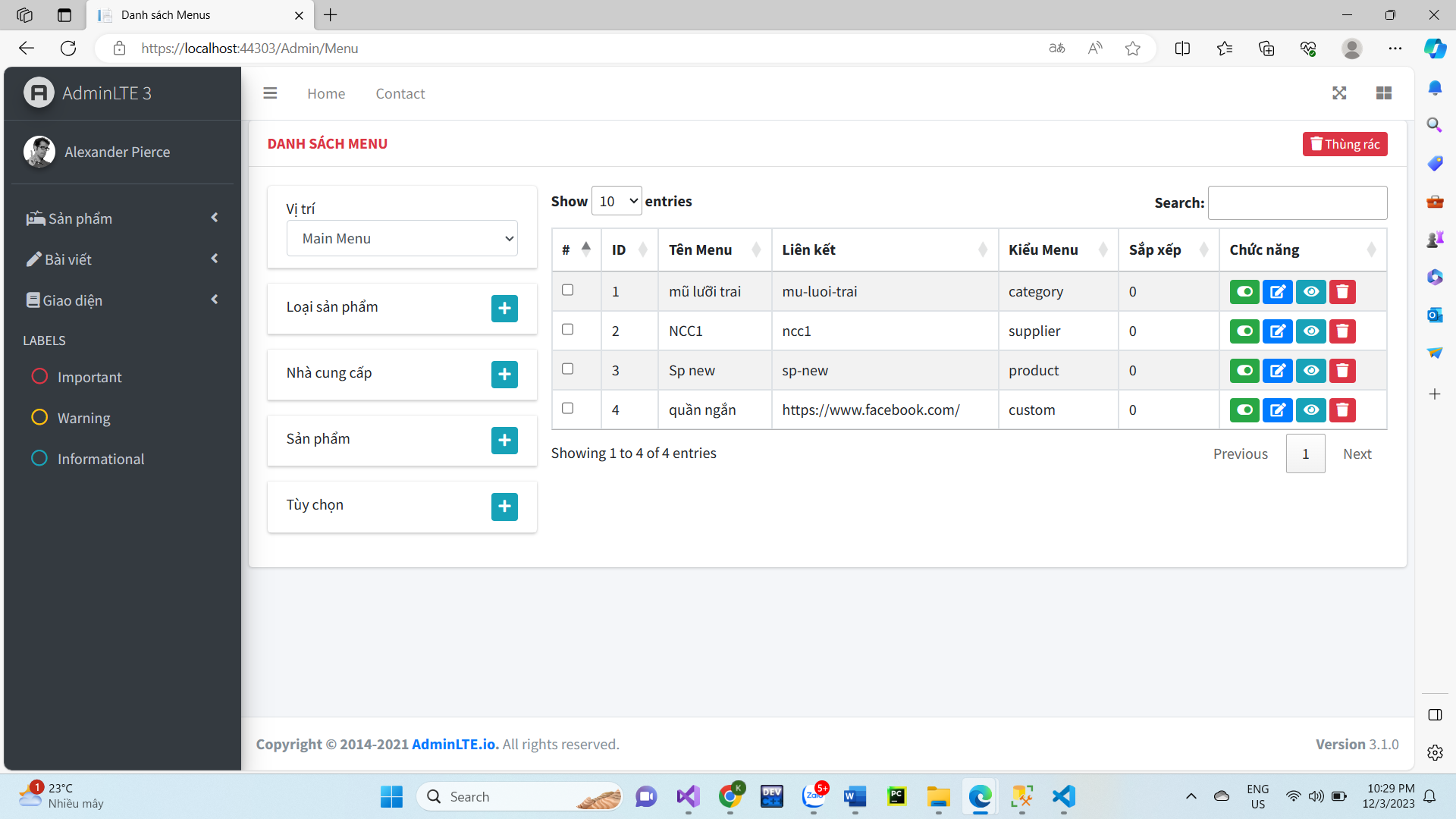
            }

            ViewBag.ListOrder = new SelectList(suppliersDAO.getList("Index"), "Order", "Name");

            return View(suppliers);

        }

## 3.6 Chức năng Menus



public class MenuController : Controller

    {

        CategoriesDAO categoriesDAO = new CategoriesDAO();

        SuppliersDAO suppliersDAO = new SuppliersDAO();

        ProductsDAO productsDAO = new ProductsDAO();

        MenusDAO menusDAO = new MenusDAO();

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

        // GET: Admin/Menu/Index

        public ActionResult Index()

        {

            ViewBag.CatList = categoriesDAO.getList("Index");

            ViewBag.SupList = suppliersDAO.getList("Index");

            ViewBag.ProList = productsDAO.getList("Index");

            return View(menusDAO.getList("Index"));

        }

        [HttpPost]

        public ActionResult Index(FormCollection form)

        {

            //Them Loai san pham

            if (!string.IsNullOrEmpty(form["ThemCategory"]))

            {

                //kiem tra dau check cua muc con

                if (!string.IsNullOrEmpty(form["nameCategory"]))

                {

                    var listitem = form["nameCategory"];

                    //chuyen danh sach thanh dang mang: 1,2,3,4...

                    var listarr = listitem.Split(',');//ngat mang thanh tung phan tu cach nhau boi dau ,

                    foreach (var row in listarr)

                    {

                        int id = int.Parse(row);//ep kieu int

                        //lay 1 ban ghi

                        Categories categories = categoriesDAO.getRow(id);

                        //ta ra menu

                        Menus menu = new Menus();

                        menu.Name = categories.Name;

                        menu.Link = categories.Slug;

                        menu.TypeMenu = "category";

                        menu.Position = form["Position"];

                        menu.ParentId = 0;

                        menu.Order = 0;

                        menu.CreateAt = DateTime.Now;

                        // Trước khi chuyển đổi giá trị từ Session["UserID"], hãy kiểm tra xem nó có null hay không

                        if (Session["UserID"] != null)

                        {

                            // Chuyển đổi giá trị từ Session["UserID"] thành một số nguyên và gán cho menu.CreateBy

                            menu.CreateBy = Convert.ToInt32(Session["UserID"].ToString());

                        }

                        else

                        {

                            // Xử lý trường hợp khi Session["UserID"] là null

                            // Ví dụ: Hiển thị một thông báo hoặc thực hiện hành động khác theo ý bạn

                            TempData["message"] = new XMessage("danger", "Lỗi: UserID không khả dụng.");

                            // Có thể chuyển hướng hoặc xử lý ngay tại đây tùy thuộc vào yêu cầu của bạn

                        }

                        menu.UpdateAt = DateTime.Now;

                        // Trước khi chuyển đổi giá trị từ Session["UserID"], hãy kiểm tra xem nó có null hay không

                        if (Session["UserID"] != null)

                        {

                            // Chuyển đổi giá trị từ Session["UserID"] thành một số nguyên và gán cho menu.CreateBy

                            menu.UpdateBy = Convert.ToInt32(Session["UserID"].ToString());

                        }

                        else

                        {

                            // Xử lý trường hợp khi Session["UserID"] là null

                            // Ví dụ: Hiển thị một thông báo hoặc thực hiện hành động khác theo ý bạn

                            TempData["message"] = new XMessage("danger", "Lỗi: UserID không khả dụng.");

                            // Có thể chuyển hướng hoặc xử lý ngay tại đây tùy thuộc vào yêu cầu của bạn

                        }

                        menu.Status = 2; //tam thoi chua xuat ban

                        //Them vao DB

                        menusDAO.Insert(menu);

                    }

                    TempData["message"] = new XMessage("success", "Thêm vào menu thành công");

                }

                else

                {

                    TempData["message"] = new XMessage("danger", "Chưa chọn danh mục loại sản phẩm");

                }

            }

            //Them Nha cung cap

            if (!string.IsNullOrEmpty(form["ThemSupplier"]))

            {

                //kiem tra dau check cua muc con

                if (!string.IsNullOrEmpty(form["nameSupplier"]))

                {

                    var listitem = form["nameSupplier"];

                    //chuyen danh sach thanh dang mang: 1,2,3,4...

                    var listarr = listitem.Split(',');//ngat mang thanh tung phan tu cach nhau boi dau ,

                    foreach (var row in listarr)

                    {

                        int id = int.Parse(row);//ep kieu int

                        //lay 1 ban ghi

                        Suppliers suppliers = suppliersDAO.getRow(id);

                        //ta ra menu

                        Menus menu = new Menus();

                        menu.Name = suppliers.Name;

                        menu.Link = suppliers.Slug;

                        menu.TypeMenu = "supplier";

                        menu.Position = form["Position"];

                        menu.ParentId = 0;

                        menu.Order = 0;

                        menu.CreateAt = DateTime.Now;

                        // Trước khi chuyển đổi giá trị từ Session["UserID"], hãy kiểm tra xem nó có null hay không

                        if (Session["UserID"] != null)

                        {

                            // Chuyển đổi giá trị từ Session["UserID"] thành một số nguyên và gán cho menu.CreateBy

                            menu.CreateBy = Convert.ToInt32(Session["UserID"].ToString());

                        }

                        else

                        {

                            // Xử lý trường hợp khi Session["UserID"] là null

                            // Ví dụ: Hiển thị một thông báo hoặc thực hiện hành động khác theo ý bạn

                            TempData["message"] = new XMessage("danger", "Lỗi: UserID không khả dụng.");

                            // Có thể chuyển hướng hoặc xử lý ngay tại đây tùy thuộc vào yêu cầu của bạn

                        }

                        menu.UpdateAt = DateTime.Now;

                        // Trước khi chuyển đổi giá trị từ Session["UserID"], hãy kiểm tra xem nó có null hay không

                        if (Session["UserID"] != null)

                        {

                            // Chuyển đổi giá trị từ Session["UserID"] thành một số nguyên và gán cho menu.CreateBy

                            menu.UpdateBy = Convert.ToInt32(Session["UserID"].ToString());

                        }

                        else

                        {

                            // Xử lý trường hợp khi Session["UserID"] là null

                            // Ví dụ: Hiển thị một thông báo hoặc thực hiện hành động khác theo ý bạn

                            TempData["message"] = new XMessage("danger", "Lỗi: UserID không khả dụng.");

                            // Có thể chuyển hướng hoặc xử lý ngay tại đây tùy thuộc vào yêu cầu của bạn

                        }

                        menu.Status = 2; //tam thoi chua xuat ban

                        //Them vao DB

                        menusDAO.Insert(menu);

                    }

                    TempData["message"] = new XMessage("success", "Thêm vào menu thành công");

                }

                else

                {

                    TempData["message"] = new XMessage("danger", "Chưa chọn nhà cung cấp");

                }

            }

            //Them San pham

            if (!string.IsNullOrEmpty(form["ThemProduct"]))

            {

                //kiem tra dau check cua muc con

                if (!string.IsNullOrEmpty(form["nameProduct"]))

                {

                    var listitem = form["nameProduct"];

                    //chuyen danh sach thanh dang mang: 1,2,3,4...

                    var listarr = listitem.Split(',');//ngat mang thanh tung phan tu cach nhau boi dau ,

                    foreach (var row in listarr)

                    {

                        int id = int.Parse(row);//ep kieu int

                        //lay 1 ban ghi

                        Products products = productsDAO.getRow(id);

                        //ta ra menu

                        Menus menu = new Menus();

                        menu.Name = products.Name;

                        menu.Link = products.Slug;

                        menu.TypeMenu = "product";

                        menu.Position = form["Position"];

                        menu.ParentId = 0;

                        menu.Order = 0;

                        menu.CreateAt = DateTime.Now;

                        // Trước khi chuyển đổi giá trị từ Session["UserID"], hãy kiểm tra xem nó có null hay không

                        if (Session["UserID"] != null)

                        {

                            // Chuyển đổi giá trị từ Session["UserID"] thành một số nguyên và gán cho menu.CreateBy

                            menu.CreateBy = Convert.ToInt32(Session["UserID"].ToString());

                        }

                        else

                        {

                            // Xử lý trường hợp khi Session["UserID"] là null

                            // Ví dụ: Hiển thị một thông báo hoặc thực hiện hành động khác theo ý bạn

                            TempData["message"] = new XMessage("danger", "Lỗi: UserID không khả dụng.");

                            // Có thể chuyển hướng hoặc xử lý ngay tại đây tùy thuộc vào yêu cầu của bạn

                        }

                        menu.UpdateAt = DateTime.Now;

                        // Trước khi chuyển đổi giá trị từ Session["UserID"], hãy kiểm tra xem nó có null hay không

                        if (Session["UserID"] != null)

                        {

                            // Chuyển đổi giá trị từ Session["UserID"] thành một số nguyên và gán cho menu.CreateBy

                            menu.UpdateBy = Convert.ToInt32(Session["UserID"].ToString());

                        }

                        else

                        {

                            // Xử lý trường hợp khi Session["UserID"] là null

                            // Ví dụ: Hiển thị một thông báo hoặc thực hiện hành động khác theo ý bạn

                            TempData["message"] = new XMessage("danger", "Lỗi: UserID không khả dụng.");

                            // Có thể chuyển hướng hoặc xử lý ngay tại đây tùy thuộc vào yêu cầu của bạn

                        }

                        menu.Status = 2; //tam thoi chua xuat ban

                        //Them vao DB

                        menusDAO.Insert(menu);

                    }

                    TempData["message"] = new XMessage("success", "Thêm vào menu thành công");

                }

                else

                {

                    TempData["message"] = new XMessage("danger", "Chưa chọn sản phẩm");

                }

            }

            //Them Custom

            if (!string.IsNullOrEmpty(form["ThemCustom"]))

            {

                //kiem tra dau check cua muc con

                if (!string.IsNullOrEmpty(form["nameCustom"]) && !string.IsNullOrEmpty(form["linkCustom"]))

                {

                    //tao ra menu custom

                    Menus menu = new Menus();

                    menu.Name = form["nameCustom"];

                    menu.Link = form["linkCustom"];

                    menu.TypeMenu = "custom";

                    menu.Position = form["Position"];

                    menu.ParentId = 0;

                    menu.Order = 0;

                    menu.CreateAt = DateTime.Now;

                    // Trước khi chuyển đổi giá trị từ Session["UserID"], hãy kiểm tra xem nó có null hay không

                    if (Session["UserID"] != null)

                    {

                        // Chuyển đổi giá trị từ Session["UserID"] thành một số nguyên và gán cho menu.CreateBy

                        menu.CreateBy = Convert.ToInt32(Session["UserID"].ToString());

                    }

                    else

                    {

                        // Xử lý trường hợp khi Session["UserID"] là null

                        // Ví dụ: Hiển thị một thông báo hoặc thực hiện hành động khác theo ý bạn

                        TempData["message"] = new XMessage("danger", "Lỗi: UserID không khả dụng.");

                        // Có thể chuyển hướng hoặc xử lý ngay tại đây tùy thuộc vào yêu cầu của bạn

                    }

                    menu.UpdateAt = DateTime.Now;

                    // Trước khi chuyển đổi giá trị từ Session["UserID"], hãy kiểm tra xem nó có null hay không

                    if (Session["UserID"] != null)

                    {

                        // Chuyển đổi giá trị từ Session["UserID"] thành một số nguyên và gán cho menu.CreateBy

                        menu.UpdateBy = Convert.ToInt32(Session["UserID"].ToString());

                    }

                    else

                    {

                        // Xử lý trường hợp khi Session["UserID"] là null

                        // Ví dụ: Hiển thị một thông báo hoặc thực hiện hành động khác theo ý bạn

                        TempData["message"] = new XMessage("danger", "Lỗi: UserID không khả dụng.");

                        // Có thể chuyển hướng hoặc xử lý ngay tại đây tùy thuộc vào yêu cầu của bạn

                    }

                    menu.Status = 2; //tam thoi chua xuat ban

                    //Them vao DB

                    menusDAO.Insert(menu);

                    TempData["message"] = new XMessage("success", "Thêm vào menu thành công");

                }

                else

                {

                    TempData["message"] = new XMessage("danger", "Chưa đầy đủ thông tin cho menu");

                }

            }

            //tra ve trang Index

            return RedirectToAction("Index", "Menu");

        }

### 3.6.1 Tạo mới Menus

 public ActionResult Create()

        {

            return View();

        }

        // POST: Admin/Menu/Create

        // To protect from overposting attacks, enable the specific properties you want to bind to, for

        // more details see https://go.microsoft.com/fwlink/?LinkId=317598.

        [HttpPost]

        [ValidateAntiForgeryToken]

        public ActionResult Create([Bind(Include = "Id,Name,TableID,TypeMenu,Position,Link,ParentId,Order,CreateAt,CreateBy,UpdateAt,UpdateBy,Status")] Menus menus)

        {

            if (ModelState.IsValid)

            {

                menusDAO.Insert(menus);

                return RedirectToAction("Index");

            }

            return View(menus);

        }

### 3.6.2 Xem chi tiết Menus

public ActionResult Details(int? id)

        {

            if (id == null)

            {

                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

            }

            Menus menus = menusDAO.getRow(id);

            if (menus == null)

            {

                return HttpNotFound();

            }

            return View(menus);

        }

### 3.6.3 Xóa Menus

public ActionResult Delete(int? id)

        {

            if (id == null)

            {

                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

            }

            Menus menus = menusDAO.getRow(id);

            if (menus == null)

            {

                return HttpNotFound();

            }

            return View(menus);

        }

        // POST: Admin/Menu/Delete/5

        [HttpPost, ActionName("Delete")]

        [ValidateAntiForgeryToken]

        public ActionResult DeleteConfirmed(int id)

        {

            Menus menus = menusDAO.getRow(id);

            menusDAO.Delete(menus);

            return RedirectToAction("Index");

        }

### 3.6.4 Thay đổi trạng thái Menus

 public ActionResult Status(int? id)

        {

            if (id == null)

            {

                //thong bao that bai

                TempData["message"] = new XMessage("danger", "Cập nhật trạng thái thất bại");

                return RedirectToAction("Index");

            }

            //truy van dong co id = id yeu cau

            Menus menus = menusDAO.getRow(id);

            if (menus == null)

            {

                //thong bao that bai

                TempData["message"] = new XMessage("danger", "Cập nhật trạng thái thất bại");

                return RedirectToAction("Index");

            }

            else

            {

                //chuyen doi trang thai cua Satus tu 1<->2

                menus.Status = (menus.Status == 1) ? 2 : 1;

                //cap nhat gia tri UpdateAt

                menus.UpdateAt = DateTime.Now;

                //cap nhat lai DB

                menusDAO.Update(menus);

                //thong bao cap nhat trang thai thanh cong

                TempData["message"] = TempData["message"] = new XMessage("success", "Cập nhật trạng thái thành công");

                return RedirectToAction("Index");

            }

        }

### 3.6.5 Chỉnh sửa Menus

public ActionResult Edit(int? id)

        {

            if (id == null)

            {

                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

            }

            Menus menus = menusDAO.getRow(id);

            if (menus == null)

            {

                return HttpNotFound();

            }

            return View(menus);

        }

        // POST: Admin/Menu/Edit/5

        // To protect from overposting attacks, enable the specific properties you want to bind to, for

        // more details see https://go.microsoft.com/fwlink/?LinkId=317598.

        [HttpPost]

        [ValidateAntiForgeryToken]

        public ActionResult Edit([Bind(Include = "Id,Name,TableID,TypeMenu,Position,Link,ParentId,Order,CreateAt,CreateBy,UpdateAt,UpdateBy,Status")] Menus menus)

        {

            if (ModelState.IsValid)

            {

                menusDAO.Update(menus);

                return RedirectToAction("Index");

            }

            return View(menus);

        }

# Phần 4: KẾT LUẬN

Dựa trên việc học học phần ASP.NET MVC, bạn có thể kết luận như sau:

## 1. \*\*Kiến thức về lý thuyết:\*\*

- Đã hiểu rõ về kiến trúc MVC (Model-View-Controller) và cách nó hoạt động trong môi trường web.

- Hiểu về cách xử lý yêu cầu HTTP, quản lý trạng thái và luồng điều khiển trong ứng dụng web.

- Nắm vững các khái niệm về routing, model binding, và validation trong ASP.NET MVC.

- Có kiến thức vững về Razor View Engine và cách tạo giao diện người dùng.

## 2. \*\*Kiến thức về thực hành:\*\*

- Có khả năng xây dựng ứng dụng web sử dụng ASP.NET MVC.

- Biết cách tương tác với cơ sở dữ liệu thông qua Entity Framework hoặc các công nghệ truy cập dữ liệu khác.

- Hiểu về các công cụ và kỹ thuật kiểm thử ứng dụng web.

## 3. \*\*Xây dựng ứng dụng với các chức năng:\*\*

- Tính năng quản lý người dùng và phân quyền.

- Quản lý dữ liệu thông qua CRUD operations (Create, Read, Update, Delete).

- Tích hợp các công nghệ và thư viện bên thứ ba (ví dụ: Bootstrap, jQuery) để tối ưu hóa trải nghiệm người dùng.

- Xử lý và validate dữ liệu đầu vào từ người dùng.

## 4. \*\*Ưu điểm:\*\*

- Dễ dàng phát triển và duy trì do sự tách biệt giữa các thành phần của kiến trúc MVC.

- Hỗ trợ phát triển ứng dụng linh hoạt và có thể mở rộng.

- Tích hợp tốt với công cụ và ngôn ngữ lập trình .NET.

## 5. \*\*Hạn chế:\*\*

- Có thể có độ phức tạp cao đối với người mới học lập trình web.

- Một số tình huống yêu cầu kiến thức sâu rộng về framework và .NET ecosystem.

## 6. \*\*Kiến nghị/giải pháp phù hợp:\*\*

- Tiếp tục nâng cao kỹ năng lập trình và hiểu rõ về các tính năng mới trong ASP.NET MVC.

- Thực hành thường xuyên để củng cố kiến thức và kỹ năng thực hành.

- Tìm hiểu thêm về các công nghệ mới và xu hướng trong lĩnh vực phát triển web.

- Tham gia cộng đồng và dự án mã nguồn mở để có cơ hội học hỏi và chia sẻ kinh nghiệm.

# Phần 5 Link github

https://github.com/KhaiTran2003/PTUDW