# Bài thực hành 4: XÂY DỰNG MENU CRUD CHO CATEGORIES

## trang Index

@{

ViewBag.Title = "Tất cả loại sản phẩm";

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

}



## Giao diện hiển thị

### Danh sách loại sản phẩm

<thead>

<tr>

<th>#</th>

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

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

...

</tr>

</thead>

@foreach (var item in Model)

{

<tr>

<td><input type="checkbox" name="checkid" value=""/></td>

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

...

<th>Chức năng</th>

</tr>

}



### Chỉnh sửa nút Create New (Thêm mới)

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

~~@Html.ActionLink("Create New", "Create")~~

<a href="@Url.Action("Create","Category")" class="btn-sm btn-success"> <i class="fa fa-plus"></i>Thêm</a>

<a href="@Url.Action("Trash[[1]](#footnote-1)","Category")" class="btn-sm btn-danger"> <i class="fa fa-trash"></i>Thùng rác</a>

</div>



### Chỉnh sửa các nút chức năng

<td>

~~@Html.ActionLink("Edit", "Edit", new { id = item.Id }) |~~

~~@Html.ActionLink("Details", "Details", new { id = item.Id }) |~~

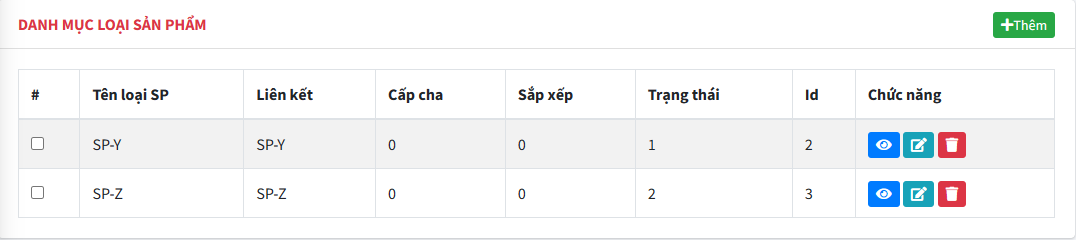
~~@Html.ActionLink("Delete", "Delete", new { id = item.Id })~~

<a href="@Url.Action("Details","Category", new { id = item.Id })" class="btn-sm btn-primary"> <i class="fa fa-eye"></i></a>

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

<a href="@Url.Action("DelTrash","Category", new { id = item.Id })" class="btn-sm btn-danger"> <i class="fa fa-trash"></i></a>

</td>



### Thêm mục Status (Trạng thái) vào mục chức năng

<thead>

<tr>

<th>#</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.Status)</th>~~

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

<th>Chức năng</th>

</tr>

</thead>

@foreach (var item in Model)

{

<tr>

<td><input type="checkbox" name="checkid" value=""/></td>

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

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

<td>@Html.DisplayFor(modelItem => item.ParentID)</td>

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

~~<td>@Html.DisplayFor(modelItem => item.Status)</td>~~

<td>@item.Id</td>

<td>

@if (item.Status == 1)

{

<a href="@Url.Action("Status","Category", new { id = item.Id })" class="btn-sm btn-success"> <i class="fa fa-toggle-on"></i></a>

}

else

{

<a href="@Url.Action("Status","Category", new { id = item.Id })" class="btn-sm btn-danger"> <i class="fa fa-toggle-off"></i></a>

}

<a href="@Url.Action("Details","Category", new { id = item.Id })" class="btn-sm btn-primary"> <i class="fa fa-eye"></i></a>

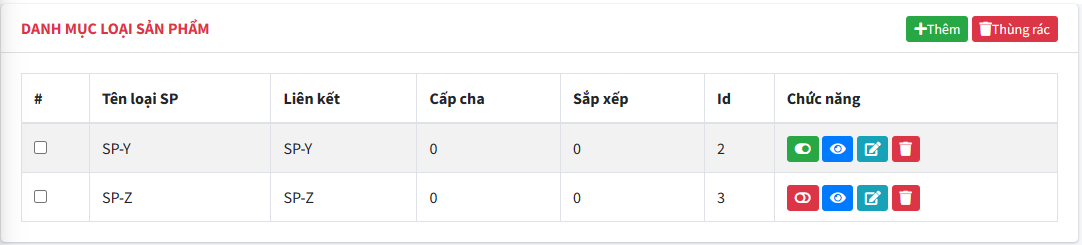
<a href="@Url.Action("Edit","Category", new { id = item.Id })" class="btn-sm btn-info"> <i class="fa fa-edit"></i></a>

<a href="@Url.Action("DelTrash","Category", new { id = item.Id })" class="btn-sm btn-danger"> <i class="fa fa-trash"></i></a>

</td>

</tr>

}



# XÂY DỰNG TRANG THÊM MỚI

## TIÊU ĐỀ TRANG, tiêu để bài viết

@{

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

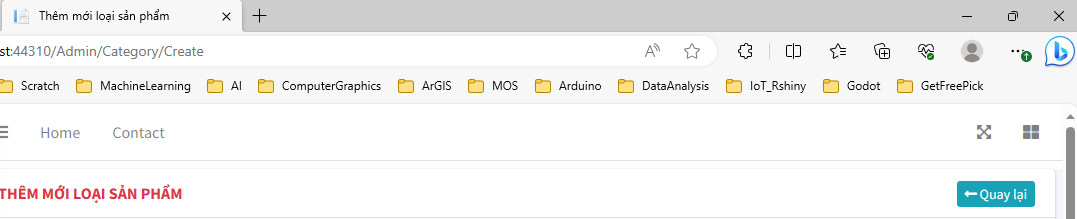
Layout = "~/Areas/Admin/Views/Shared/\_LayoutAdmin.cshtml";

}

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

<strong class="text-danger">THÊM MỚI LOẠI SẢN PHẨM</strong>

</div>



## Thiết kế lại trang nhập thông tin

### Tạo ra 2 vùng nhập dữ liệu

<div class="card-body">

@\*Phan NOI DUNG\*@

<div class="row">

@\* Phần nhập tay \*@

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

</div>

@\* Phần lựa chọn \*@

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

</div>

</div>

</div>

## Khung nhập dữ liệu

#### Nhắc lại bài trước, ta chỉ nhập một số trường thông tin, còn 1 số trường thì lựa chọn hoặc tự sinh

### Tên danh mục loại sản phẩm

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

<div class="form-group">

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

<label>Tên danh mục sản phẩm <span class="text-danger"> (\*)</span></label>

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

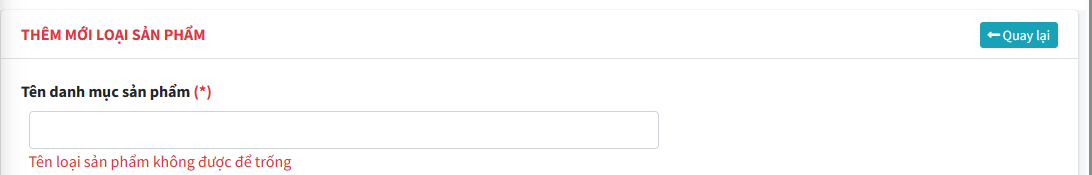
@Html.EditorFor(model=>model.Name,new{htmlAttributes=new @class="form-control"}})

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

</div>

</div>

</div>



Thiết kế những vùng dữ liệu bắt buộc sẽ thêm dấu \*

### Mô tả loại sản phẩm

<div class="form-group">

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

<label>Mô tả chi tiết <span class="text-danger"> (\*)</span></label>

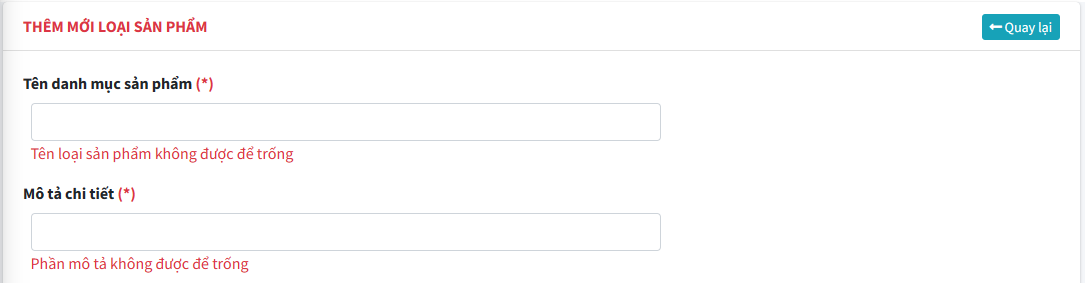
<div class="col-md-10">

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

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

</div>

</div>



### Tử khóa loại sản phẩm

<div class="form-group">

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

<label>Từ khóa <span class="text-danger"> (\*)</span></label>

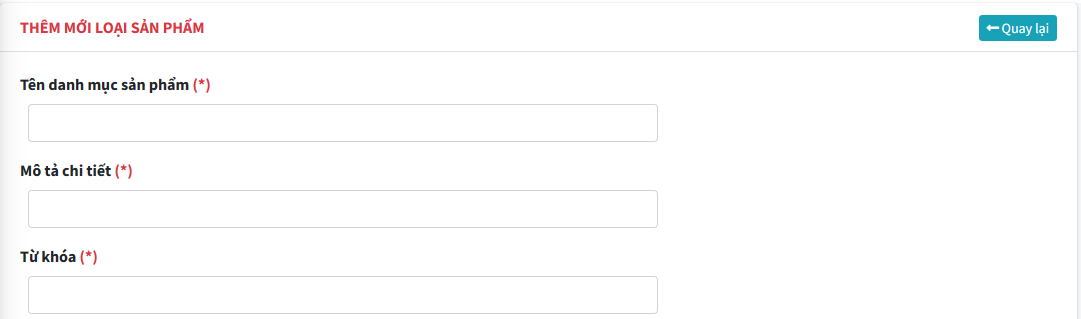
<div class="col-md-10">

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

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

</div>

</div>



### Chỉnh sửa khung nhập liệu

Có thể nhận thấy ở đây mục mô tả chi tiết và Từ khóa cho loại sản phẩm chỉ có 1 dòng, chúng ta có thể mở rộng ra với số hàng tùy ý.

<div class="form-group">

<label>Mô tả chi tiết <span class="text-danger"> (\*)</span></label>

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

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

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

</div>

</div>

<div class="form-group">

<label>Từ khóa <span class="text-danger"> (\*)</span></label>

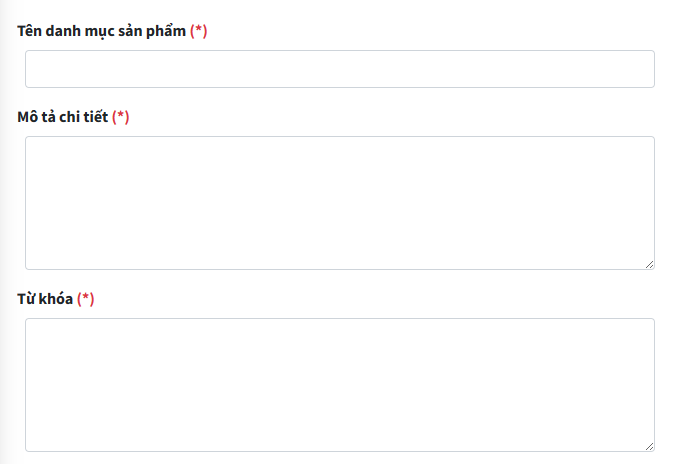
<div class="col-md-10">

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

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

</div>

</div>



## Khung chọn dữ liệu

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

<div class="form-group">

<label>Cấp cha <span class="text-danger"></span></label>

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

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

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

</div>

</div>

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

<div class="form-group">

<label>Sắp xếp<span class="text-danger"></span></label>

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

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

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

</div>

</div>

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

<div class="form-group">

<label>Trạng thái <span class="text-danger"> (\*) </span></label>

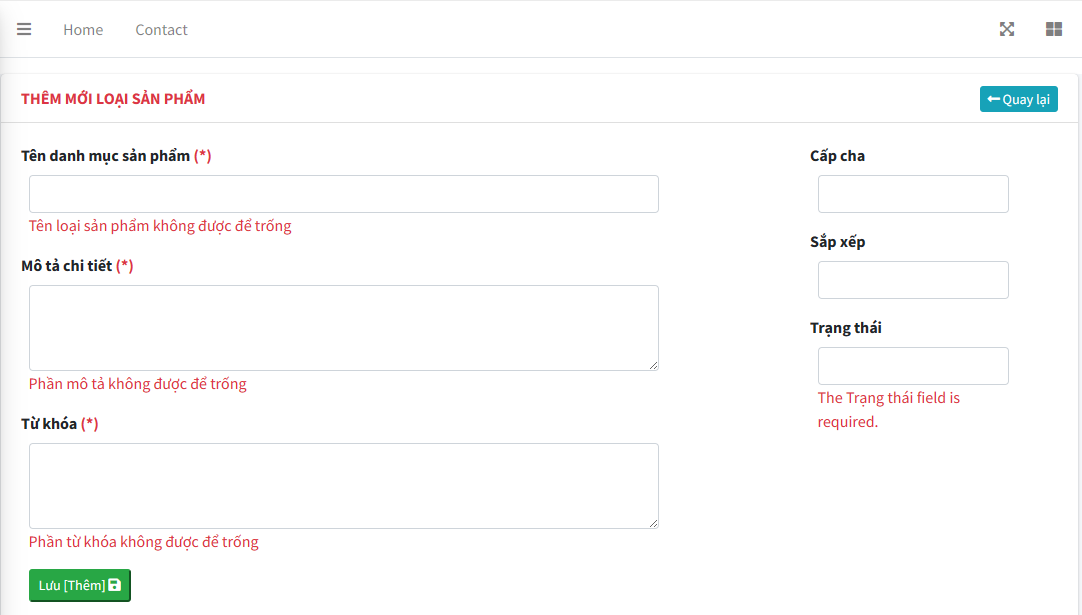
<div class="col-md-10">

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

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

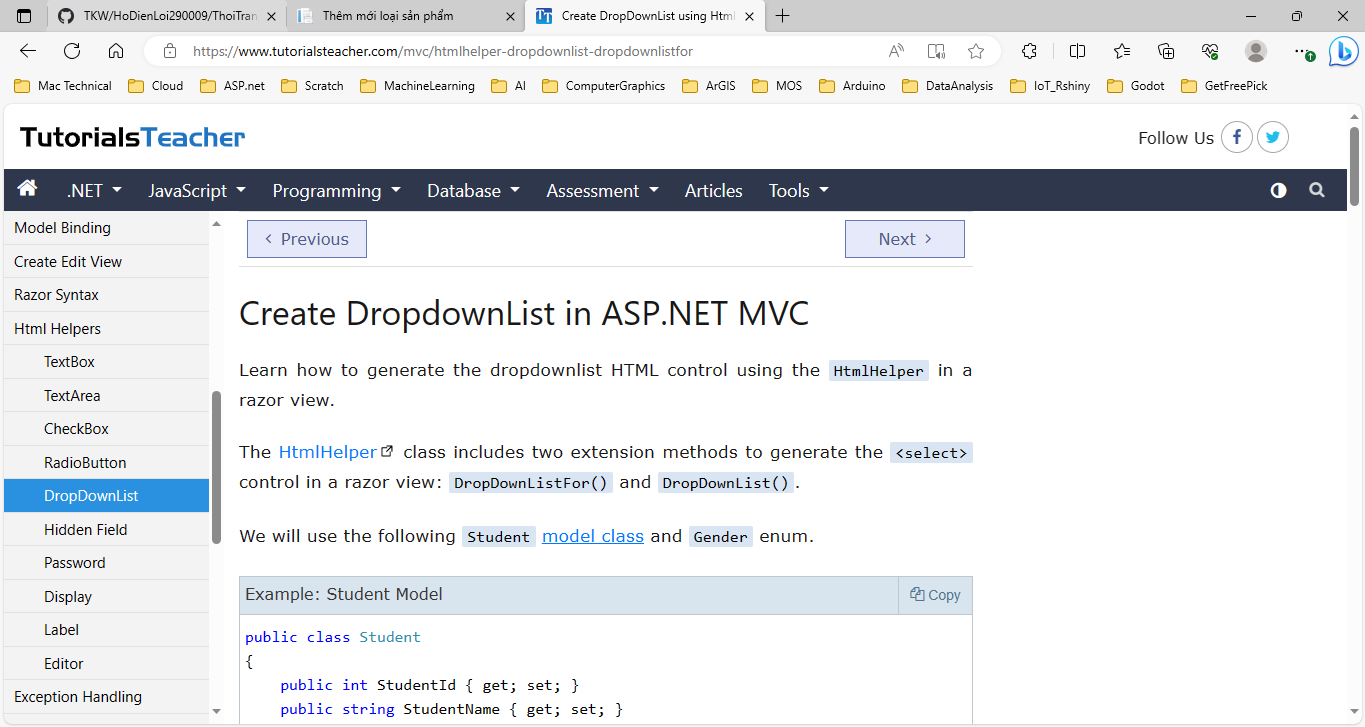
</div>

</div>



## Thiết kế mục chọn cho trạng thái

### Tìm đến tài liệu tham khảo để biết cách thiết lập cho Dropdown



<div class="form-group">

<label>Trạng thái <span class="text-danger"> (\*)</span></label>

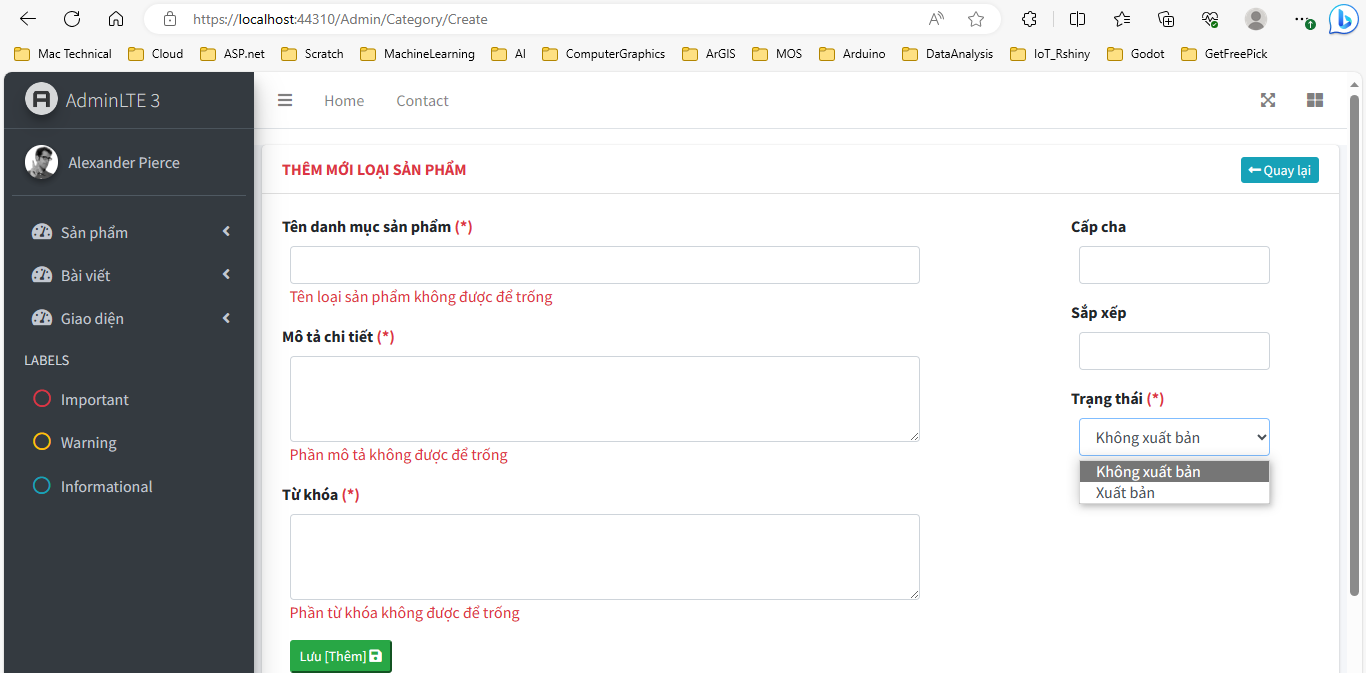
<div class="col-md-10">

@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>



#### Lưu ý: Đối với trạng thái là một danh sách mảng tĩnh (SelectList)

## Thiết kế mục chọn cho MỤC CẤP CHA

Đối với danh mục cấp cha được truy vấn dữ liệu tử cơ sở dữ liệu.

### Tạo ra ListCat trong hàm Create (CategoriesController)

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

// GET: Admin/Category/Create: Them moi mot mau tin

public ActionResult Create()

{

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

return View();

}

**Mục đích:** Tạo ra một danh sách SelectList được trả về từ hàm getList đã viết trong CategoriesDAO và chỉ nhận 2 trường Id và Name

Kết quả trả về ở ViewBag.ListCat[[2]](#footnote-2)

### Thiết kế file Create.cshtml

<div class="form-group">

<label>Cấp cha <span class="text-danger"></span></label>

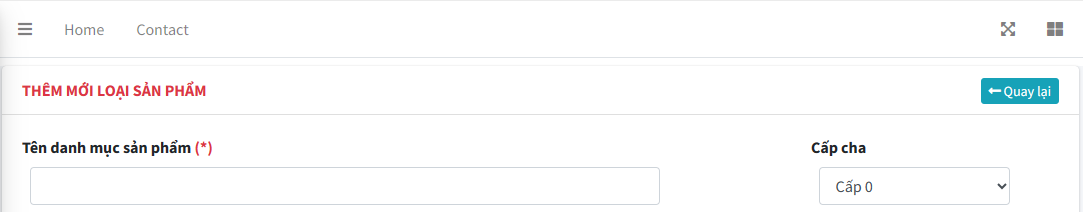
<div class="col-md-10">

@Html.DropDownListFor(model => model.ParentID, (SelectList)[[3]](#footnote-3)@ViewBag.ListCat, "Cấp 0"[[4]](#footnote-4), new { @class="form-control"[[5]](#footnote-5)})

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

</div>

</div>



## Thiết kế mục chọn cho MỤC sắp xếp

### Tạo ra OrderList trong hàm Create (CategoriesController)

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

// GET: Admin/Category/Create: Them moi mot mau tin

public ActionResult Create()

{

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

ViewBag.OrderList = new SelectList(categoryDAO.getList("Index"), "Order", "Name");

return View();

}

**Mục đích:** Tạo ra một danh sách SelectList được trả về từ hàm getList đã viết trong CategoriesDAO và chỉ nhận 2 trường Order và Name

Kết quả trả về ở ViewBag.OrderList[[6]](#footnote-6)

### Thiết kế file Create.cshtml

<div class="form-group">

<label>Sắp xếp <span class="text-danger"></span></label>

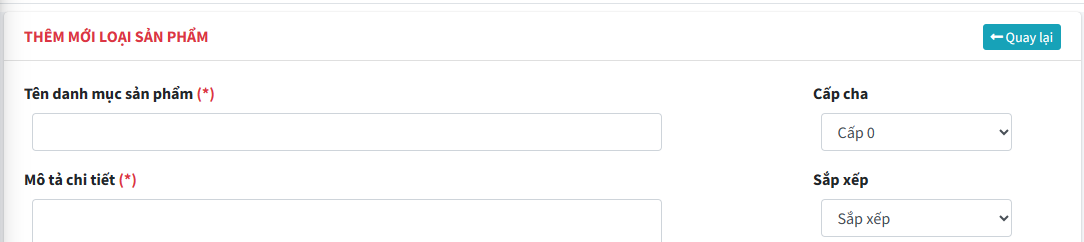
<div class="col-md-10">

@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>



# Lưu ý:

Khi bấm nút thêm mà các trường Tên danh mục, Mô tả chi tiết, từ khóa mà để trống thì thao tác được gọi đến hàm:

// POST: Admin/Category/Create: Them moi mot mau tin

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Create(Categories categories)

{

if (ModelState.IsValid)

{

categoryDAO.Insert(categories);

return RedirectToAction("Index");

}

return View(categories);

}

Mà trong hàm này sẽ trả về View(categories) nghĩa là các trường khác sẽ không tồn tại.



Do vật ta cần thêm phần truyền dữ liệu trả về cho các mục trong ViewBag.

// POST: Admin/Category/Create: Them moi mot mau tin

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Create(Categories categories)

{

if (ModelState.IsValid)

{

categoryDAO.Insert(categories);

return RedirectToAction("Index");

}

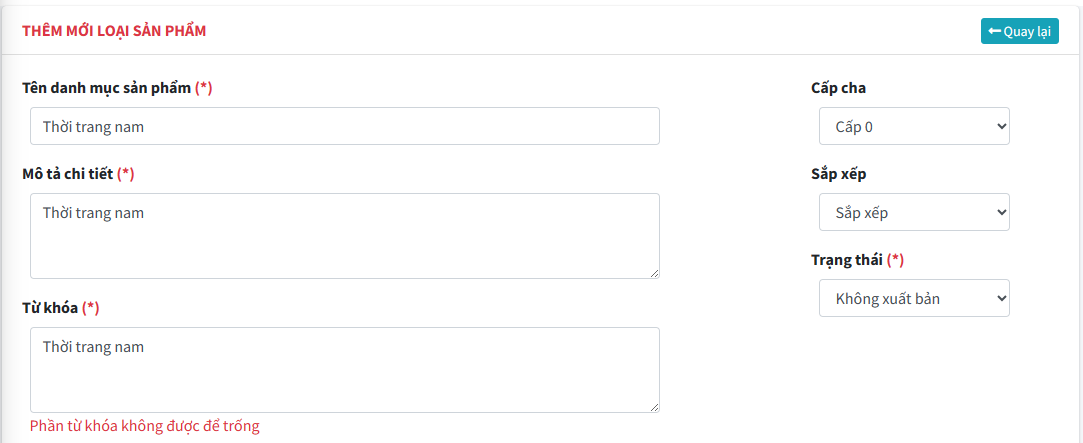
ViewBag.ListCat = new SelectList(categoryDAO.getList("Index"), "Id", "Name");

ViewBag.OrderList = new SelectList(categoryDAO.getList("Index"), "Order", "Name");

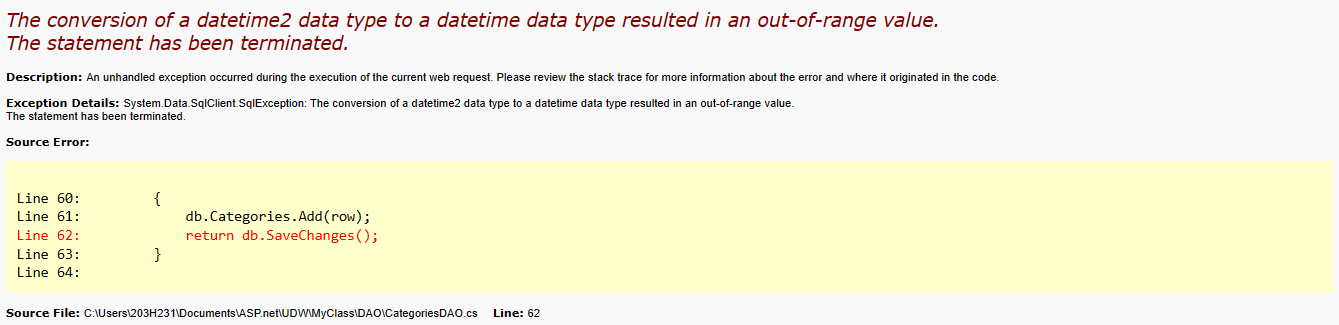
return View(categories);

}

Tuy vậy, khi nhập đủ dữ liệu như sau:



Kết quả sau khi nhấn nút thêm



Lỗi là do các trường bắt buộc: CreateAt, CreateBy, UpdateAt, UpdateBy còn thiếu dữ liệu

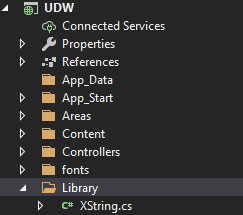
# xử lý dữ liệu cho slug, createat/by, updateat/by

# sluG

## Tạo ra thư viện dùng để chuyển đổi chữ Tiếng Việt

R\_Click UDW - Add - Folder = Library

R\_Click Library - Add Class - Xstring.cs



using System;

using System.Collections.Generic;

using System.Linq;

using System.Text.RegularExpressions;

using System.Web;

namespace UDW.Library

{

public class XString

{

public static string Str\_Slug(string s)

{

string[][] symbols ={

new String[]{"[áàảãạâấầẩẫậăắằẳẵặ]","a"},

new String[]{"[đ]","d"},

new String[]{"[éèẻẽẹêếềểễệ]","e"},

new String[]{"[íìỉĩị]","i"},

new String[]{"[óòỏõọôốồổỗộơớờởỡợ]","o"},

new String[]{"[úùủũụưứừửữự]","u"},

new String[]{"[ýỳỷỹỵ]","y"},

new String[]{"[\\s'\";,]","-"}

};

s = s.ToLower();

foreach (var ss in symbols)

{

s = Regex.Replace(s, ss[0], ss[1]);

}

return s;

}

}

## }

## Thao tác với hàm Create.cshml

### Thêm thư viện vừa tạo

using UDW.Library;

### Xử lý thông tin cho mục Slug

// POST: Admin/Category/Create: Them moi mot mau tin

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Create(Categories categories)

{

if (ModelState.IsValid)

{

//Xu ly cho muc Slug

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

//chuyen doi dua vao truong Name de loai bo dau, khoang cach = dau -

categoryDAO.Insert(categories);

return RedirectToAction("Index");

}

### Xử lý cho mục ParentId

public ActionResult Create(Categories categories)

{

if (ModelState.IsValid)

{

//Xu ly cho muc Slug

…

//Xu ly cho muc ParentId

if (categories.ParentID == null)

{

categories.ParentID = 0;

}

…

}

### Xử lý cho mục Order

if (ModelState.IsValid)

{

//Xu ly cho muc Slug

…

//Xu ly cho muc ParentId

…

//Xu ly cho muc Order

if (categories.Order == null)

{

categories.Order = 1;

}

else

{

categories.Order = categories.Order + 1;

}

…

}

### Xử lý cho mục CreateAt

if (ModelState.IsValid)

{

//Xu ly cho muc Slug

…

//Xu ly cho muc ParentId

…

//Xu ly cho muc Order

…

//Xu ly cho muc CreateAt

categories.CreateAt = DateTime.Now;

…

### }

### Xử lý cho mục CreateBy

Trước tiên, ta sẽ tạo Session khi thực hiện tải trang Web:

#### Open file Global.asax.cs

namespace UDW

{

public class MvcApplication : System.Web.HttpApplication

{

protected void Application\_Start()

{

…

}

//Tao moi Session

protected void Session\_Start()

{

Session["UserID"] = "1";

}

}

}

#### Open file Create.cshtml

if (ModelState.IsValid)

{

//Xu ly cho muc Slug

…

//Xu ly cho muc ParentId

…

//Xu ly cho muc Order

…

//Xu ly cho muc CreateAt

…

//Xu ly cho muc CreateBy

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

…

}

### Xử lý cho mục UpdateAt

if (ModelState.IsValid)

{

//Xu ly cho muc Slug

…

//Xu ly cho muc ParentId

…

//Xu ly cho muc Order

…

//Xu ly cho muc CreateAt

…

//Xu ly cho muc CreateBy

…

//Xy ly cho muc UpdateAt

categories.UpdateAt = DateTime.Now;

…

}

### Xử lý cho mục UpdateBy

if (ModelState.IsValid)

{

//Xu ly cho muc Slug

…

//Xu ly cho muc ParentId

…

//Xu ly cho muc Order

…

//Xu ly cho muc CreateAt

…

//Xu ly cho muc CreateBy

…

//Xy ly cho muc UpdateAt

…

//Xy ly cho muc UpdateBy

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

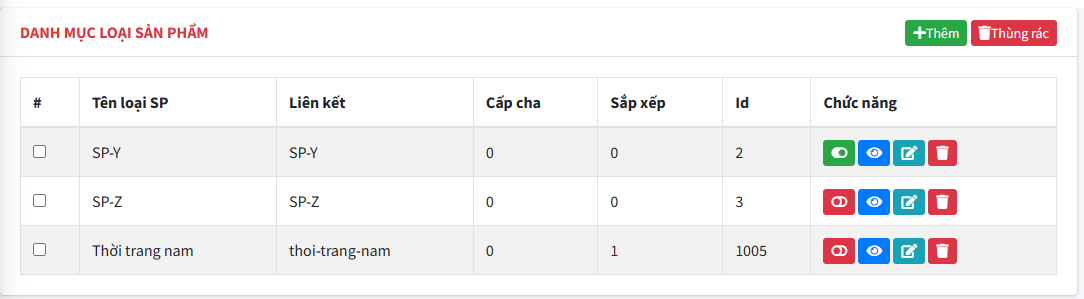
…

}

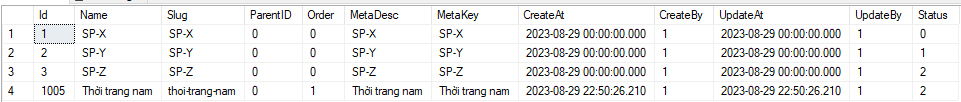
# Thử nghiệm lần 1



## Kết quả trang index

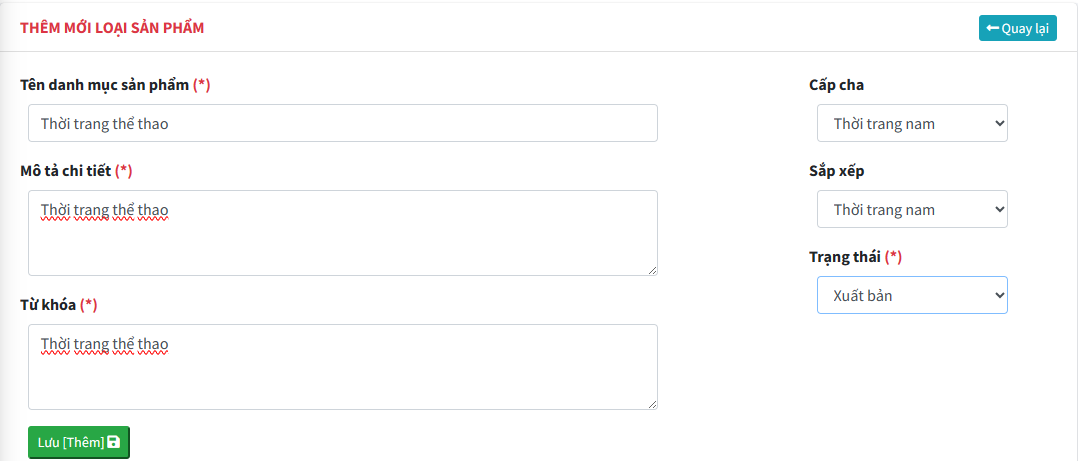


## Kết quả SQL

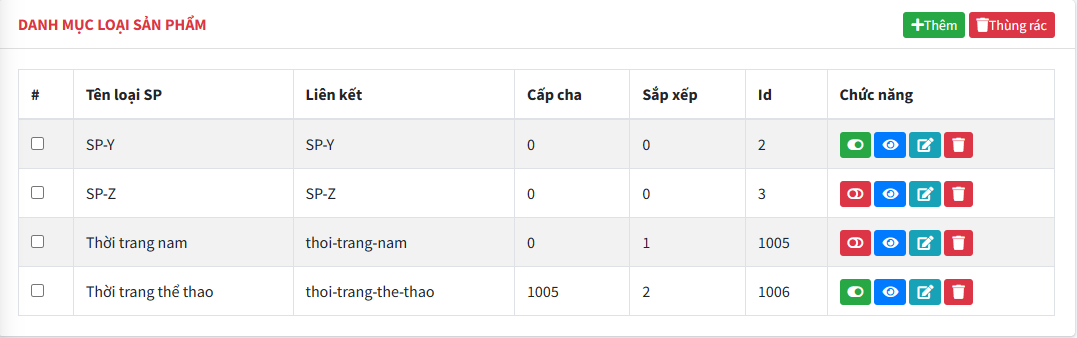


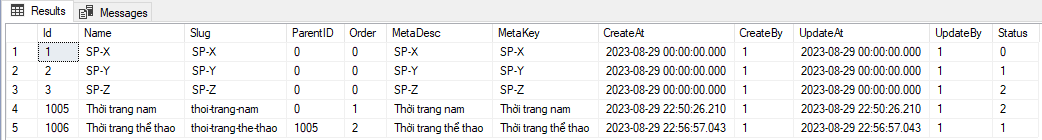
Cấp cha: 0, Order = NULL (default =1), Status = 2 (Không xuất bản) (0: Trash, 1: Index), **Slug**

# Thử nghiệm lần 2



## Kết quả





Tên loại: Thời trang thể thao là Con của Thời trang nam nên nhận ParentID = 1005

Order: Chọn sắp xếp sau Thời trang nam nên giá trị = 2

Status: Xuất bản = 1

Slug: thoi-trang-the-thao

1. Ta sẽ thực hiện sau, tạo chức năng vào thùng rác để xóa hẳn dữ liệu, phục hồi… [↑](#footnote-ref-1)
2. Tên hàm truyền [↑](#footnote-ref-2)
3. Ép kiểu trả về dạng danh sách [↑](#footnote-ref-3)
4. Nếu không chọn cấp thì nó là mục gốc [↑](#footnote-ref-4)
5. CSS cho đẹp [↑](#footnote-ref-5)
6. Tên hàm truyền [↑](#footnote-ref-6)