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

# Menu EDIT

## thiết lập tiêu đề

@model MyClass.Model.Categories

@{

ViewBag.Title = "Chỉnh sửa loại sản phẩm";

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

}

## 

## Thiết lập menu

## Thiết lập nội dung

#### Sử dụng nội dung trang Create

### Phần nội dung nhập bằng tay

<div class="card-body">

@\*Phan NOI DUNG\*@

<div class="row">

@\* Phần nhập tay \*@

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

<div class="form-group">

<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 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 = "3"})

@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="3"})

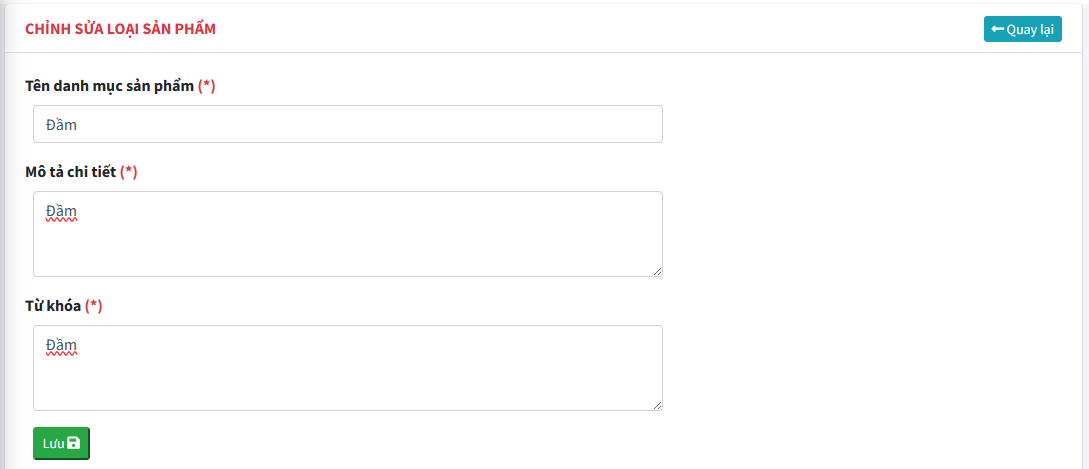
@Html.ValidationMessageFor(model => model.MetaKey, "", new{@class = "text-danger" })

</div>

</div>

</div>

…



### Phần nội dung lựa chọn

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

// GET: Admin/Category/Edit/5: Cap nhat mau tin

public ActionResult Edit(int? id)

{

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

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

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Categories categories = categoryDAO.getRow(id);

if (categories == null)

{

return HttpNotFound();

}

return View(categories);

}

Trong CategoryController cần trả về 2 danh sách để tiến hành lựa chọn, để trong Edit.cshtml tiến hành lựa chọn trên giao diện

@\* Phần lựa chọn \*@

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

<div class="form-group">

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

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

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

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

</div>

</div>

<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" })

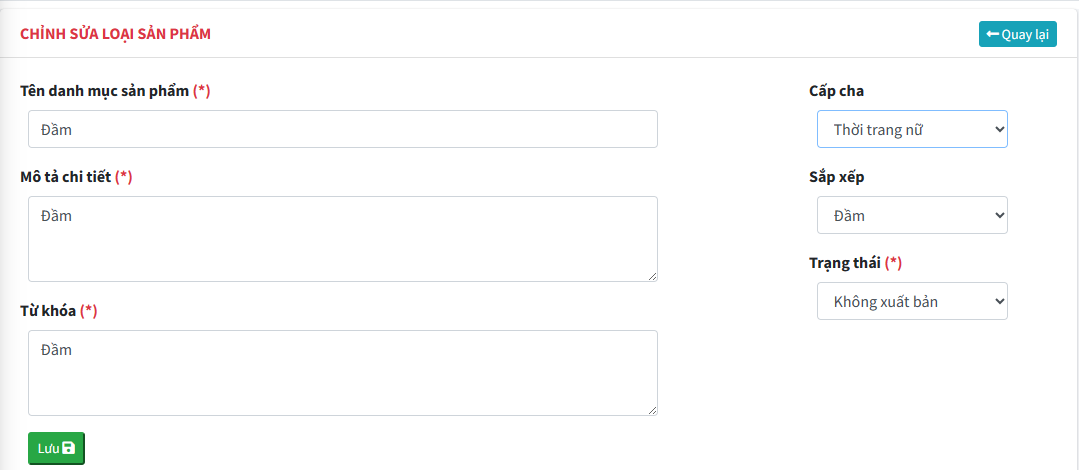
@Html.ValidationMessageFor(model => model.Order, "", new { @class = "text-danger" })

</div>

…

</div>

</div>



### Khi nhấn Lưu:



#### Nguyên nhân chưa trả về ListCat và List Order trở khi load trang Index

// POST: Admin/Category/Edit/5: Cap nhat mau tin

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Edit(Categories categories)

{

if (ModelState.IsValid)

{

categoryDAO.Update(categories);

return RedirectToAction("Index");

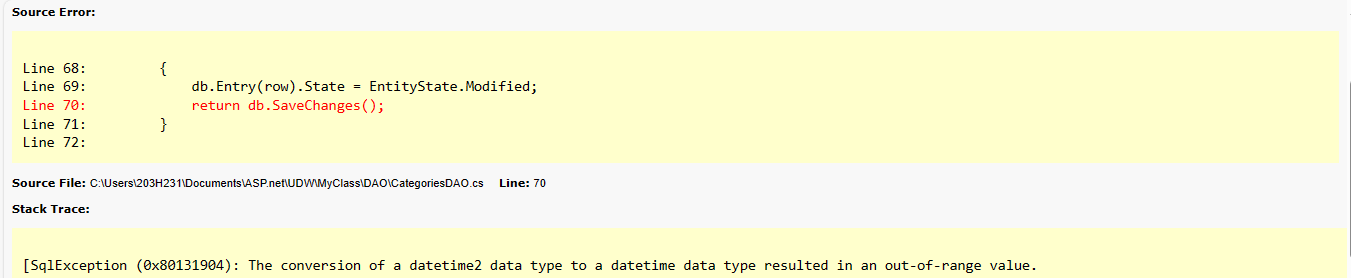
}

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

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

return View(categories);

}



#### Nguyên nhân một số trường tự động chưa tiến hành cập nhật: CreateAt/By, UpdateAt/By, Slug

### Cập nhật một số trường tự động

// POST: Admin/Category/Edit/5: Cap nhat mau tin

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Edit(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 -

//Xu ly cho muc ParentId

if (categories.ParentID == null)

{

categories.ParentID = 0;

}

//Xu ly cho muc Order

if (categories.Order == null)

{

categories.Order = 1;

}

else

{

categories.Order = categories.Order + 1;

}

//Xu ly cho muc CreateAt

categories.CreateAt = DateTime.Now;

//Xu ly cho muc CreateBy

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

//Xy ly cho muc UpdateAt

categories.UpdateAt = DateTime.Now;

//Xy ly cho muc UpdateBy

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

//Thong bao thanh cong

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

//Cap nhat du lieu[[1]](#footnote-1)

categoryDAO.Update(categories);

return RedirectToAction("Index");

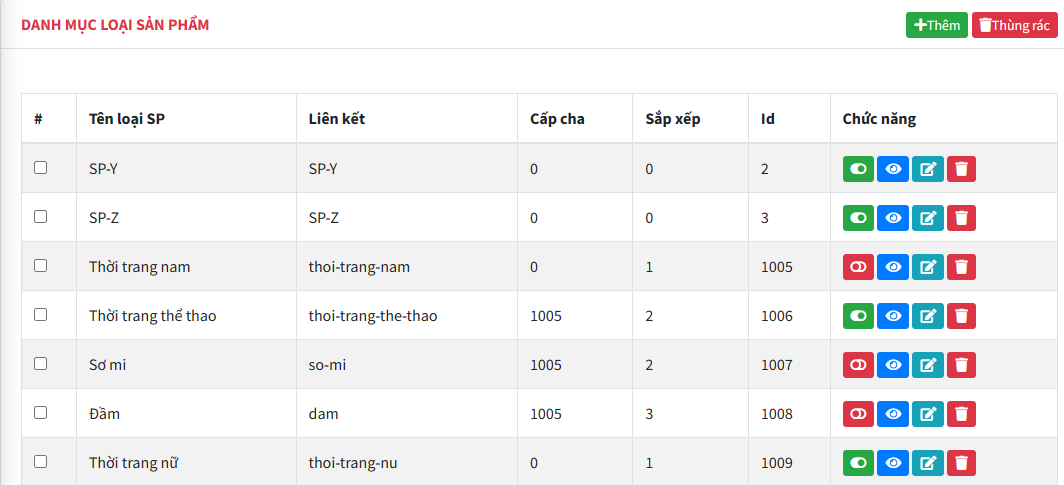
}

return View(categories);

}

# Kết quả

## Trước khi Edit



Danh mục Đầm có **cấp cha** là **1005** thuộc về **thời trang thể thao**

## Sau khi Edit

Danh mục Đầm có **cấp cha** là **1009** thuộc về **thời trang thể nữ**



# Vấn đề phát sinh



Ngày tạo (CreateAt) và ngày cập nhật (UpdateAt) sẽ giống nhau.

Nguyên nhân là vì:

Ngay từ đầu ta thiết lập các trường: CreateAt/By và UpdateAt/By không cho phép NULL. Vì thế mà khi Edit ta không thể bỏ trống các trường này (Hệ thống sẽ báo lỗi DateTime).

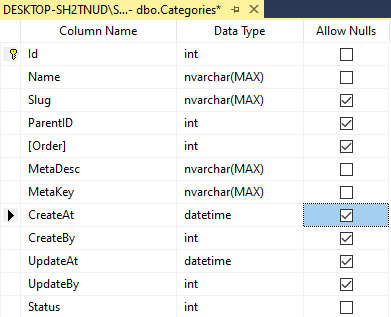
Nên là: Trong đoạn chương trình EDIT (CategoryController) ta không thể xóa các dòng CreateAt/By

Tương tự như vậy trong đoạn chương trình CREATE (CategoryController) ta cũng không thể xóa dòng UpdateAt/By.

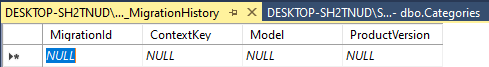
# Giải pháp:

Sửa lại cơ sở dữ liệu, thay thế các thuộc tính thành AllowNull cho cả 2 phần:

## SQL



### Xóa dữ liệu ở \_MigrationHistory (nếu có)



## Categories.cs

[Display(Name = "Ngày tạo")]

public DateTime? CreateAt { get; set; }

[Display(Name = "Người tạo")]

public int? CreateBy { get; set; }

[Display(Name = "Ngày cập nhật")]

public DateTime? UpdateAt { get; set; }

[Display(Name = "Người cập nhật")]

public int? UpdateBy { get; set; }

## Chỉnh sửa lại mục Create

//Xy ly cho muc UpdateAt

~~categories.UpdateAt = DateTime.Now;~~

//Xy ly cho muc UpdateBy

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

## chỉnh sửa lại mục Edit

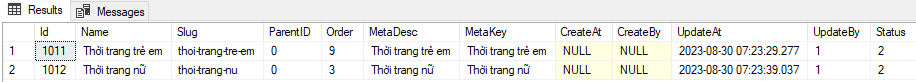
//Xu ly cho muc CreateAt

~~categories.CreateAt = DateTime.Now;~~

//Xu ly cho muc CreateBy

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

# kết quả



### Xem xét đoạn đầu tiên

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

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

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

Ta thấy rằng @Html.HiddenFor(model => model.Id); nhằm ngăn chặn việc cập nhật giá trị cho ID

Tương tự ta sẽ ngăn chặn giá trị cho CreatAt và CreatBy

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

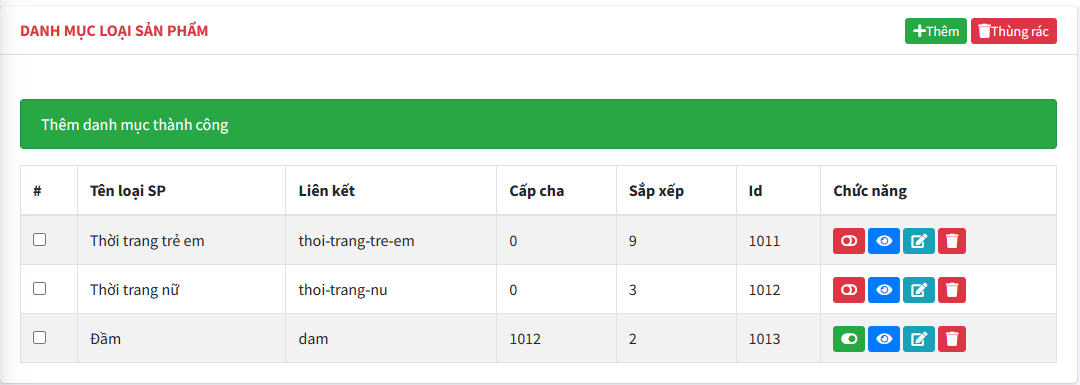
@Html.ValidationSummary(true, "", new { @class = "text-danger" })

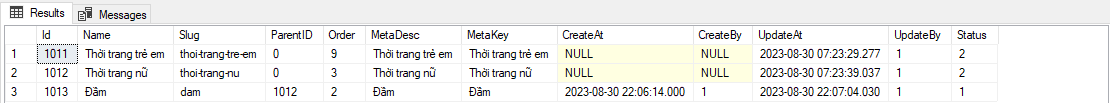
@Html.HiddenFor(model => model.Id);

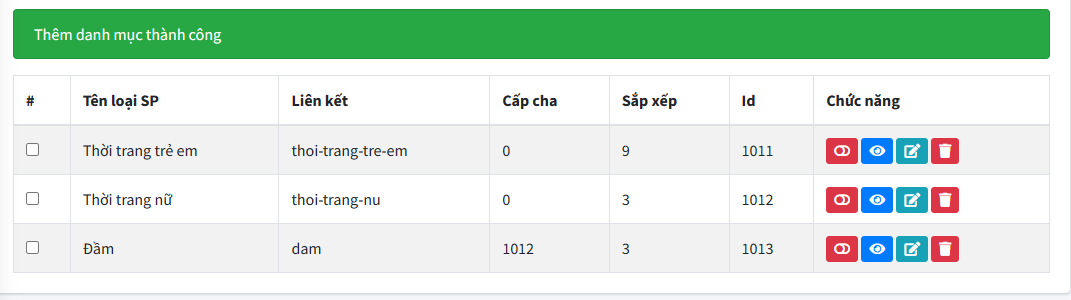
@Html.HiddenFor(model => model.CreateAt);

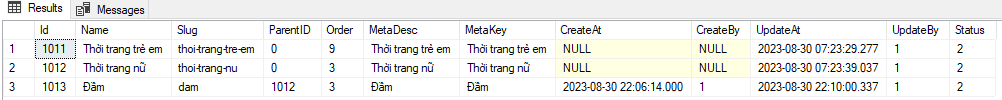
@Html.HiddenFor(model => model.CreateBy);

### Kết quả









1. Rất quan trọng [↑](#footnote-ref-1)