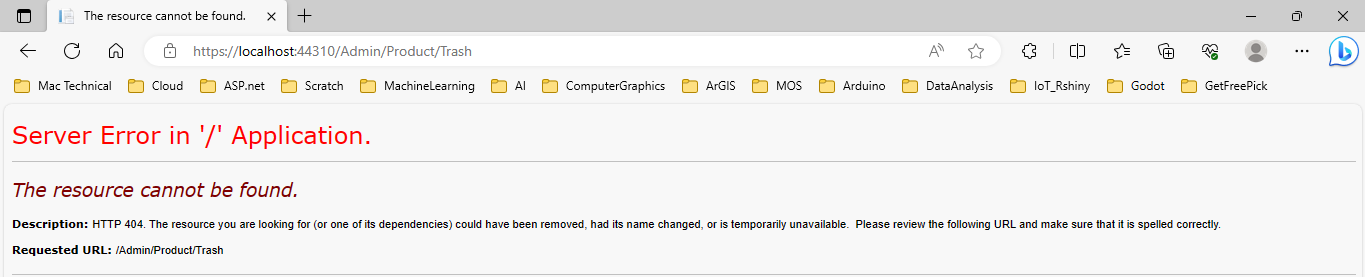
# Bài thực hành 6: xây dựng trang quản trị cho products

# Menu trash



### Thiết kế chức năng:



Khi nhấn vào biểu tượng thùng rác sẽ hiển thị các mẩu tin đã xóa (Status = 0).

Và có 2 chức năng: Phục hồi (chuyển trạng thái về trạng thái =2), xóa hẳn (xóa ra khỏi cơ sở dữ liệu)

## Tạo controller cho trash

Để hiển thị thùng rác nghĩa là xem xét các mẩu tin có Status = 0. Hàm Getlist trong CategoryDAO

public List<Products> getList(string status = "All")

{

List<Products> list = null;

switch (status)

{

case "Index":

{

list = db.Products

.Where(m => m.Status != 0)

.ToList();

break;

}

case "Trash":

{

list = db.Products

.Where(m => m.Status == 0)

.ToList();

break;

}

default:

{

list = db.Products.ToList();

break;

}

}

return list;

}

#### Để ý thầy răng Status mặc định là All, ta sẽ truyền status = “Trash”

Sao chép nội dung của Index

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

// GET: Admin/Product/Trash/5:Hien thi cac mau tin có gia tri la 0

public ActionResult Trash(int? id)

{

return View(productDAO.getList("Trash"));

}

## Tạo giao diện trang thùng rác

R\_Click Trash - Add View

Empty (Empty Model)

Name = Trash

### Chỉnh sửa tiêu đề

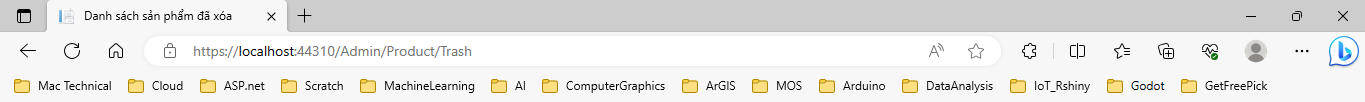
@model IEnumerable<MyClass.Model.Suppliers>

@{

ViewBag.Title = "Danh sách sản phẩm đã xóa";

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

}



### Chỉnh sửa giao diện

Danh sách hiển thị giống với trang Index (Copy tòa bộ trang Index bỏ vô)

Lưu ý: Khai báo sử dụng Model Category

@model IEnumerable<MyClass.Model.Products>

Và hiển thị dưới dạng bảng biểu

@section header {

<link href="~/Public/css/cdn.datatables.net\_1.13.6\_css\_jquery.dataTables.min.css" rel="stylesheet" />

}

@section footer {

<script src="~/Public/js/cdn.datatables.net\_1.13.6\_js\_jquery.dataTables.min.js"></script>

<script>let table = new DataTable('#myTable');</script>

}



### Menu tiêu đề

Chỉ có nút quay trở về trang Index (tham khảo Details)

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

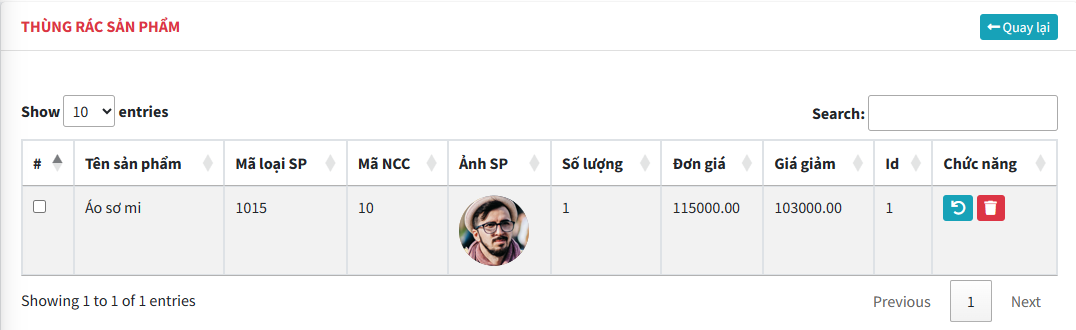


### Menu chức năng

Chỉ có nút Recover và Delete (Xóa khỏi cơ sở dữ liệu)

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

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



# phục hồi dữ liệu

## Thiết lập hàm recover

Hàm Recover chính là việc chuyển trạng thái Status = 0 sang = 2 (Tham khảo chức năng DelTrash).

Lưu ý: sau khi phục hồi tập tin thì vẫn ở trong thùng rác.

#### (yêu cầu hơn: Chuyển về trạng thái trước khi xóa?)

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

// GET: Admin/Product/Recover/5:Thay doi trang thai cua mau tin

public ActionResult Recover(int? id)

{

if (id == null)

{

//Thong bao that bai

TempData["message"] = new XMessage("danger", "Phục hồi sản phẩm thất bại");

//chuyen huong trang

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

}

//khi nhap nut thay doi Status cho mot mau tin

Products products = productDAO.getRow(id);

//kiem tra id cua categories co ton tai?

if (products == null)

{

//Thong bao that bai

TempData["message"] = new XMessage("danger", "Phục hồi sản phẩm thất bại");

//chuyen huong trang

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

}

//thay doi trang thai Status = 2

products.Status = 2;

//cap nhat gia tri cho UpdateAt/By

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

products.UpdateAt = DateTime.Now;

//Goi ham Update trong ProductDAO

productDAO.Update(products);

//Thong bao thanh cong

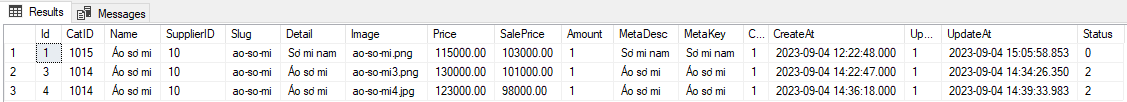
TempData["message"] = new XMessage("success", "Phục hồi sản phẩm thành công");

//khi cap nhat xong thi chuyen ve Trash de phuc hoi tiep

return RedirectToAction("Trash", "Product");

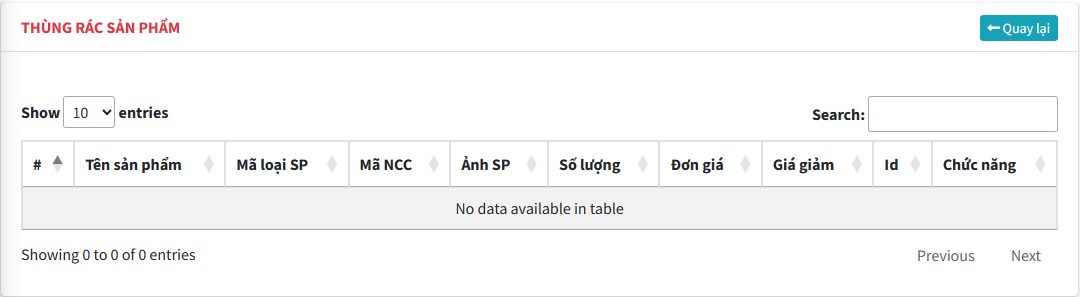
}

### Trước khi phục hồi

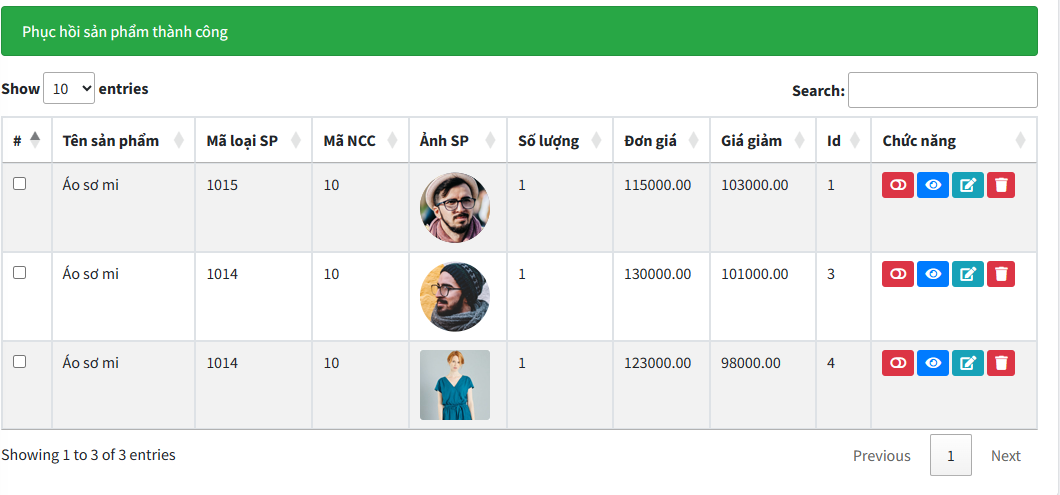


### 

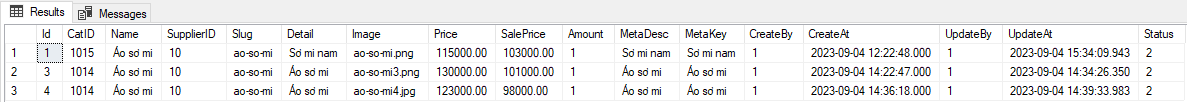
### Sau khi phục hồi



### Kiểm tra trang Index



#### Status = 2 chứ không phải là 1



# xóa ra khỏi cơ sở dữ liệu

Hàm Delete được xây dựng ngay từ đầu bởi EntityFrameWork

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

// GET: Admin/Product/Delete/5

public ActionResult Delete(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Products products = productDAO.getRow(id);

if (products == null)

{

return HttpNotFound();

}

return View(products);

}

#### Vấn đề đặt ra, mỗi sản phẩm sẽ đi kèm với đó là 1 hình. Do vậy khi xóa Sản phẩm thì đồng thời cũng xóa luôn hình ảnh trong folder Public/img/product

// POST: Admin/Supplier/Delete/5

[HttpPost, ActionName("Delete")]

[ValidateAntiForgeryToken]

public ActionResult DeleteConfirmed(int id)

{

//Truy van mau tin theo Id

Suppliers suppliers = suppliersDAO.getRow(id);

//duong dan den anh can xoa

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

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

{

//cap nhat thi phai xoa file cu

if (suppliers.Image != null)

{

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

System.IO.File.Delete(DelPath);

}

}

return RedirectToAction("Trash");

}

### Chỉnh sửa tiêu đề trang xác nhận xóa dữ liệu (Delete)

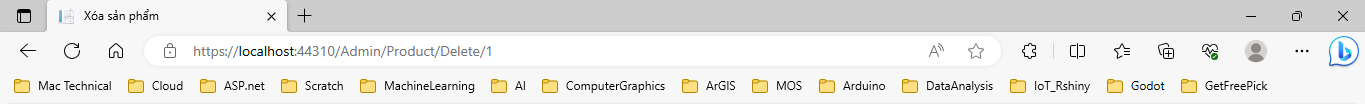
@model MyClass.Model.Products

@{

ViewBag.Title = "Xác nhận xóa danh mục sản phẩm";

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

}



### Chỉnh sửa menu chức năng (như Trash)

Lưu ý không thay thế nội dug vì trang này không sử dụng model mà giữ nguyên như phần tự động tạo ra như sau: @model MyClass.Model.Products

<div class="card-header">

<div class="row">

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

<strong class="text-danger">BẠN CÓ CHẮC CHẮN XÓA MỤC NÀY?</strong>

</div>

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

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

</div>

</div><!-- /.class-row -->

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

### Chỉnh sửa nội dung

Giữ nguyên phần Delete tự tạo bởi Entity, lư ý có form xác nhận

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

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

…

}

#### Phần nội dung

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

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

<div class="content-wrapper">

<!-- Main content -->

<section class="content my-2">

<!-- Default box -->

<div class="card">

<div class="card-header">

<div class="row">

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

<strong class="text-danger">BẠN CÓ CHẮC CHẮN XÓA MỤC NÀY?</strong>

</div>

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

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

</div>

</div>

</div>

<div class="card-body">

@\*Hien thi noi dung thong bao ngay ben tren bang du lieu\*@

@Html.Partial("Message")

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

<thead>

<tr>

<th>#</th>

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

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

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

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

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

<th>Chức năng</th>

</tr>

</thead>

<tbody>

<tr>

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

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

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

<td><img src="~/Public/img/supplier/@Model.Image" alt="Img" class="img-fluid" />

</td>

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

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

<td>

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

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

</div>

</td>

</tr>

</tbody>

</table><!-- /.card-body -->

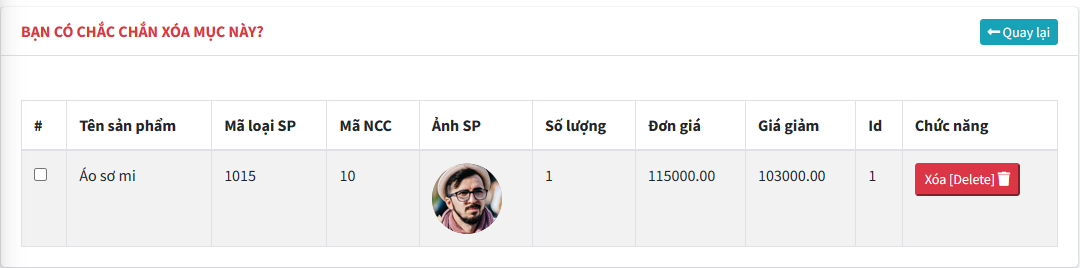
</div>

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

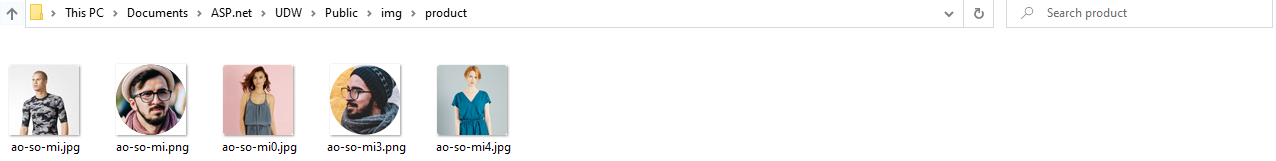
</section> <!-- /.content -->

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

}



## Trước khi xóa



## Sau khi xóa

