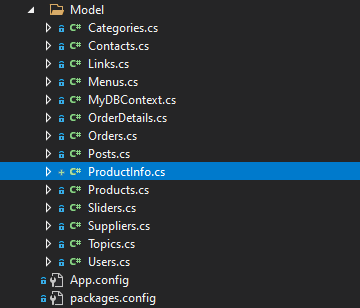


Cột Mã loại SP thay vì hiển thị các số nguyên 1027 1028 1030 thì chúng ta sẽ hiển thị Tên mã loại tương ứng: Đồ công sở, Đồ thể thao, Đồng phục.

Để thực hiện được điều này, chúng ta sẽ kết nối (JOIN) 2 bảng Categories và Products

## Tạo mới ProductInfo.cs

Trong Model - tạo một Class = ProductInfo.cs



using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace MyClass.Model

{

public class ProductInfo

{

Copy các trường của Products.cs vào đây

Lưu ý: không có: KEY, [Required], [Display]

Bổ sung thêm 1 trường: CatName

}

}

**Cụ thể**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace MyClass.Model

{

public class ProductInfo

{

public int Id { get; set; }

public int CatID { get; set; }

public string Name { get; set; }

public string CatName { get; set; }

public string SupplierId { get; set; }

public string Slug { get; set; }

public string Detail { get; set; }

public string Image { get; set; }

public decimal Price { get; set; }

public decimal SalePrice { get; set; }

public int Amount { get; set; }

public string MetaDesc { get; set; }

public string MetaKey { get; set; }

public int CreateBy { get; set; }

public DateTime CreateAt { get; set; }

public int? UpdateBy { get; set; }

public DateTime? UpdateAt { get; set; }

public int Status { get; set; }

}

}

## Join 2 bảng: ProductsDAO.cs

Bổ sung thêm 2 hàm hiển thị dữ liệu của sản phẩm cho trang Quản trị và HOME

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

//Hien thi danh sach toan bo san pham theo tung Categories (catid)

//danh cho trang HOME

public List<ProductInfo> getListByListCatId(List<int> listcatid, int limit)

{

List<ProductInfo> list = db.Products

.Join(

db.Categories,

p => p.CatID,//cua Product

c => c.Id,//join voi Categories thong qua Id cua Categroies

(p, c) => new ProductInfo

{

Id = p.Id,

CatID = p.CatID,

Name = p.Name,

CatName = c.Name,

SupplierId = p.SupplierId,

Slug = p.Slug,

Detail = p.Detail,

Image = p.Image,

Price = p.Price,

SalePrice = p.SalePrice,

Amount = p.Amount,

MetaDesc = p.MetaDesc,

MetaKey = p.MetaKey,

CreateBy = p.CreateBy,

CreateAt = p.CreateAt,

UpdateBy = p.UpdateBy,

UpdateAt = p.UpdateAt,

Status = p.Status

}

)

.Where(m => m.Status == 1 && listcatid.Contains(m.CatID))

.Take(limit)

.OrderByDescending(m => m.CreateBy)

.ToList();

return list;

}

## Site CONTROLLER

public ActionResult HomeProduct(int id)

{

CategoriesDAO categoriesDAO = new CategoriesDAO();

Categories categories = categoriesDAO.getRow(id);

ViewBag.Categories = categories;

//hien thi toan bo cac shan pham ung voi tung loai

//hien thi theo 3 cap: cha - con - con cua con

List<int> listcatid = new List<int>();

//cap 1

listcatid.Add(id);

//cap 2

List<Categories> listcategories2 = categoriesDAO.getListByPareantId(id);

if (listcategories2.Count()!=0)

{

foreach (var categories2 in listcategories2)

{

listcatid.Add(categories2.Id);

//cap 3

List<Categories> listcategories3 = categoriesDAO.getListByPareantId(categories2.Id);

if (listcategories3.Count() != 0)

{

foreach (var categories3 in listcategories3)

{

listcatid.Add(categories3.Id);

}

}

}

}

ProductsDAO productsDAO = new ProductsDAO();

List<ProductInfo[[1]](#footnote-1)> list = productsDAO.getListByListCatId(listcatid, 10);

return View("HomeProduct", list);

}

## HOMEPRODUCT.cshtml

@model IEnumerable<MyClass.Model.ProductInfo>

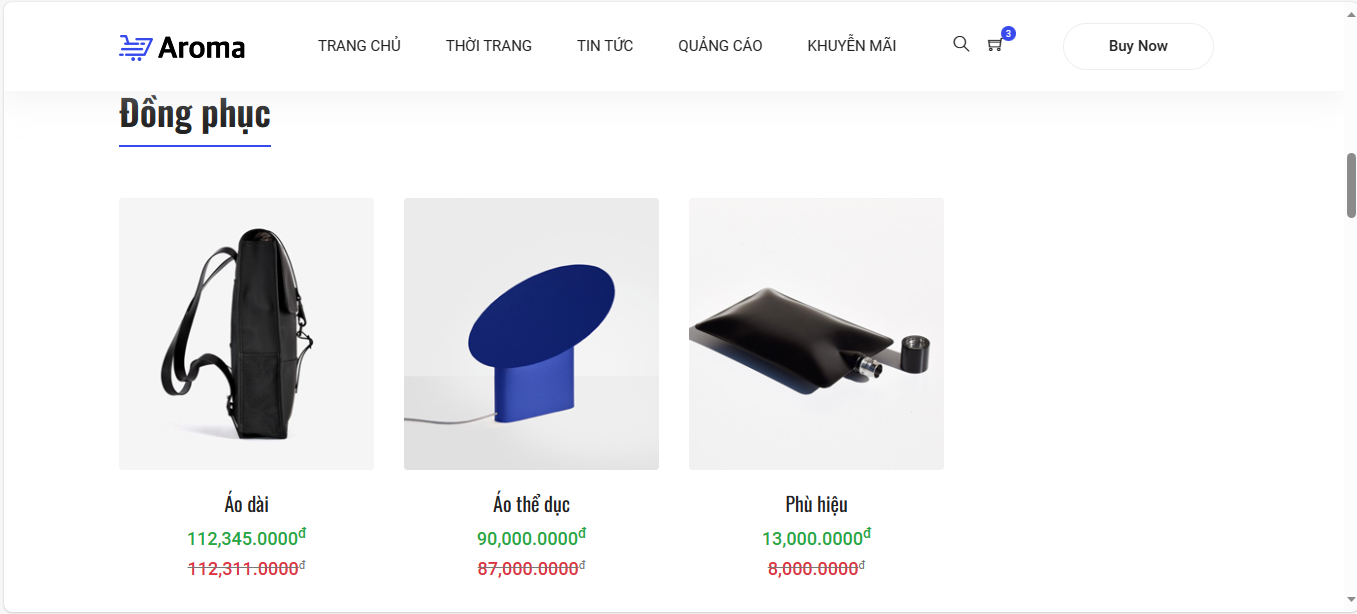
@{

var item = ViewBag.Categories;

}

…

# Kết quả



# Trang quản trị

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

//Hien thi danh sach toan bo Loai san pham: SELCT \* FROM

//danh cho trang Quan tri

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

{

List<ProductInfo> list = null;

switch (status)

{

case "Index":

{

list = db.Products

.Join(

db.Categories,

p => p.CatID,//cua Product

c => c.Id,//join voi Categories thong qua Id cua Categroies

(p, c) => new ProductInfo

{

Id = p.Id,

CatID = p.CatID,

Name = p.Name,

CatName = c.Name,

SupplierId = p.SupplierId,

Slug = p.Slug,

Detail = p.Detail,

Image = p.Image,

Price = p.Price,

SalePrice = p.SalePrice,

Amount = p.Amount,

MetaDesc = p.MetaDesc,

MetaKey = p.MetaKey,

CreateBy = p.CreateBy,

CreateAt = p.CreateAt,

UpdateBy = p.UpdateBy,

UpdateAt = p.UpdateAt,

Status = p.Status

}

)

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

.ToList();

break;

}

case "Trash":

{

list = db.Products

.Join(

db.Categories,

p => p.CatID,//cua Product

c => c.Id,//join voi Categories thong qua Id cua Categroies

(p, c) => new ProductInfo

{

Id = p.Id,

CatID = p.CatID,

Name = p.Name,

CatName = c.Name,

SupplierId = p.SupplierId,

Slug = p.Slug,

Detail = p.Detail,

Image = p.Image,

Price = p.Price,

SalePrice = p.SalePrice,

Amount = p.Amount,

MetaDesc = p.MetaDesc,

MetaKey = p.MetaKey,

CreateBy = p.CreateBy,

CreateAt = p.CreateAt,

UpdateBy = p.UpdateBy,

UpdateAt = p.UpdateAt,

Status = p.Status

}

)

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

.ToList();

break;

}

default:

{

list = db.Products

.Join(

db.Categories,

p => p.CatID,//cua Product

c => c.Id,//join voi Categories thong qua Id cua Categroies

(p, c) => new ProductInfo

{

Id = p.Id,

CatID = p.CatID,

Name = p.Name,

CatName = c.Name,

SupplierId = p.SupplierId,

Slug = p.Slug,

Detail = p.Detail,

Image = p.Image,

Price = p.Price,

SalePrice = p.SalePrice,

Amount = p.Amount,

MetaDesc = p.MetaDesc,

MetaKey = p.MetaKey,

CreateBy = p.CreateBy,

CreateAt = p.CreateAt,

UpdateBy = p.UpdateBy,

UpdateAt = p.UpdateAt,

Status = p.Status

}

)

.ToList();

break;

}

}

return list;

}

## Cập nhật nội dung trang Index.cshtml

### View\Products\Index.cshtml

@model IEnumerable<MyClass.Model.ProductInfo>[[2]](#footnote-2)

@{

ViewBag.Title = "Tất cả sản phẩm";

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

}

@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>

}

<!-- 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">DANH MỤC TỪNG SẢN PHẨM</strong>

</div>

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

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

<a href="@Url.Action("Trash","Product")" class="btn-sm btn-danger"> <i class="fa fa-trash"></i>Thùng rác</a>

</div>

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

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

<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.CatID)</th>

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

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

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

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

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

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

<th>Chức năng</th>

</tr>

</thead>

<tbody>

@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.CatName)</td>

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

<td><imgsrc="~/Public/img/product/@item.Image"alt="Img"class="img-fluid"/></td>

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

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

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

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

<td>

@if (item.Status == 1)

{

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

}

else

{

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

}

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

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

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

</td>

</tr>

}

</tbody>

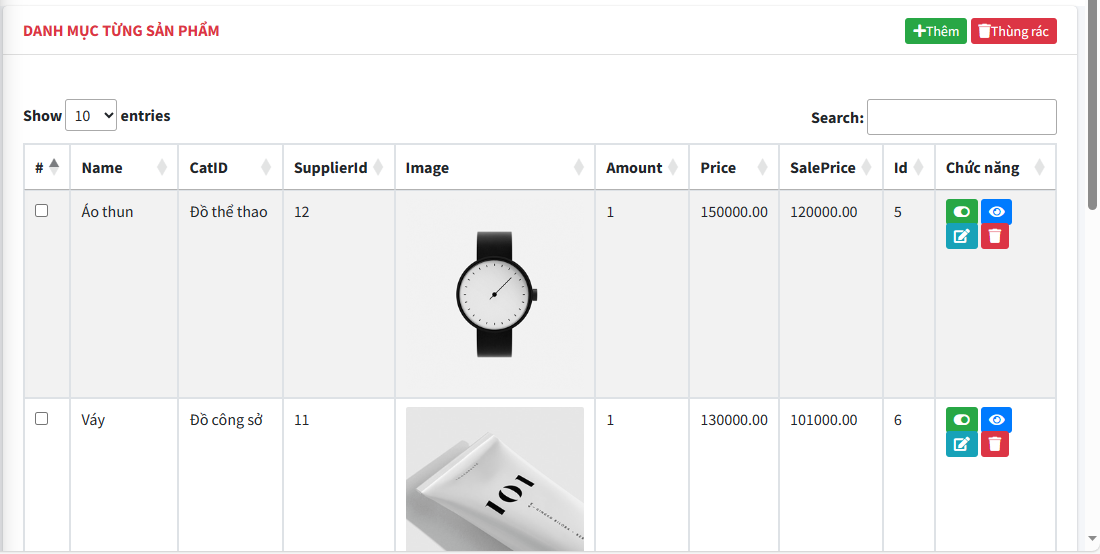
</table>

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

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

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

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



# Tương tự, xây dựng cho nhà cung cấp

## PRODUCTDAO.cs

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

//Hien thi danh sach toan bo san pham theo tung Categories (catid)

//danh cho trang HOME

public List<ProductInfo> getListByListCatId(List<int> listcatid, int limit)

{

List<ProductInfo> list = db.Products

.Join(

db.Categories, // Bảng Categories

p => p.CatID, // Khóa ngoại của Products liên kết với Categories

c => c.Id, // Khóa chính của Categories

(p, c) => new { Product = p, Category = c } // Kết hợp Products và Categories

)

.Join(

db.Suppliers, // Bảng Suppliers

pc => pc.Product.SupplierId, // Khóa ngoại của Product/Category liên kết với Suppliers

s => s.Id, // Khóa chính của Suppliers

(pc, s) => new ProductInfo

{

Id = pc.Product.Id,

CatID = pc.Product.CatID,

Name = pc.Product.Name,

CatName = pc.Category.Name, // Lấy tên danh mục từ bảng Categories

SupplierId = pc.Product.SupplierId,

SupplierName = s.Name, // Lấy tên nhà cung cấp từ bảng Suppliers

Slug = pc.Product.Slug,

Detail = pc.Product.Detail,

Image = pc.Product.Image,

Price = pc.Product.Price,

SalePrice = pc.Product.SalePrice,

Amount = pc.Product.Amount,

MetaDesc = pc.Product.MetaDesc,

MetaKey = pc.Product.MetaKey,

CreateBy = pc.Product.CreateBy,

CreateAt = pc.Product.CreateAt,

UpdateBy = pc.Product.UpdateBy,

UpdateAt = pc.Product.UpdateAt,

Status = pc.Product.Status

}

)

.Where(m => m.Status == 1 && listcatid.Contains(m.CatID))

.Take(limit)

.OrderByDescending(m => m.CreateBy)

.ToList();

return list;

}

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

//Hien thi danh sach toan bo Loai san pham: SELCT \* FROM

//danh cho trang Quan tri

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

{

List<ProductInfo> list = null;

switch (status)

{

case "Index":

{

list = db.Products

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

.Join(

db.Categories, // Bảng Categories

p => p.CatID, // Khóa ngoại của Products liên kết với Categories

c => c.Id, // Khóa chính của Categories

(p, c) => new { Product = p, Category = c }//Kết hợp Products - Categories

)

.Join(

db.Suppliers, // Bảng Suppliers

// Khóa ngoại của Product/Category liên kết với Suppliers

pc => pc.Product.SupplierId,

s => s.Id, // Khóa chính của Suppliers

(pc, s) => new ProductInfo

{

Id = pc.Product.Id,

CatID = pc.Product.CatID,

Name = pc.Product.Name,

CatName = pc.Category.Name, // Lấy tên danh mục từ bảng Categories

SupplierId = pc.Product.SupplierId,

SupplierName = s.Name, // Lấy tên nhà cung cấp từ bảng Suppliers

Slug = pc.Product.Slug,

Detail = pc.Product.Detail,

Image = pc.Product.Image,

Price = pc.Product.Price,

SalePrice = pc.Product.SalePrice,

Amount = pc.Product.Amount,

MetaDesc = pc.Product.MetaDesc,

MetaKey = pc.Product.MetaKey,

CreateBy = pc.Product.CreateBy,

CreateAt = pc.Product.CreateAt,

UpdateBy = pc.Product.UpdateBy,

UpdateAt = pc.Product.UpdateAt,

Status = pc.Product.Status

}

)

.ToList();

break;

}

case "Trash":

{

list = db.Products

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

.Join(

db.Categories, // Bảng Categories

p => p.CatID, // Khóa ngoại của Products liên kết với Categories

c => c.Id, // Khóa chính của Categories

(p, c) => new { Product = p, Category = c }//Kết hợp Products - Categories

)

.Join(

db.Suppliers, // Bảng Suppliers

// Khóa ngoại của Product/Category liên kết với Suppliers

pc => pc.Product.SupplierId,

s => s.Id, // Khóa chính của Suppliers

(pc, s) => new ProductInfo

{

Id = pc.Product.Id,

CatID = pc.Product.CatID,

Name = pc.Product.Name,

CatName = pc.Category.Name, // Lấy tên danh mục từ bảng Categories

SupplierId = pc.Product.SupplierId,

SupplierName = s.Name, // Lấy tên nhà cung cấp từ bảng Suppliers

Slug = pc.Product.Slug,

Detail = pc.Product.Detail,

Image = pc.Product.Image,

Price = pc.Product.Price,

SalePrice = pc.Product.SalePrice,

Amount = pc.Product.Amount,

MetaDesc = pc.Product.MetaDesc,

MetaKey = pc.Product.MetaKey,

CreateBy = pc.Product.CreateBy,

CreateAt = pc.Product.CreateAt,

UpdateBy = pc.Product.UpdateBy,

UpdateAt = pc.Product.UpdateAt,

Status = pc.Product.Status

}

)

.ToList();

break;

}

// Các trường hợp khác xử lý tương tự

default:

{

list = db.Products

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

.Join(

db.Categories, // Bảng Categories

p => p.CatID, // Khóa ngoại của Products liên kết với Categories

c => c.Id, // Khóa chính của Categories

(p, c) => new { Product = p, Category = c }//Kết hợp Products - Categories

)

.Join(

db.Suppliers, // Bảng Suppliers

// Khóa ngoại của Product/Category liên kết với Suppliers

pc => pc.Product.SupplierId,

s => s.Id, // Khóa chính của Suppliers

(pc, s) => new ProductInfo

{

Id = pc.Product.Id,

CatID = pc.Product.CatID,

Name = pc.Product.Name,

CatName = pc.Category.Name, // Lấy tên danh mục từ bảng Categories

SupplierId = pc.Product.SupplierId,

SupplierName = s.Name, // Lấy tên nhà cung cấp từ bảng Suppliers

Slug = pc.Product.Slug,

Detail = pc.Product.Detail,

Image = pc.Product.Image,

Price = pc.Product.Price,

SalePrice = pc.Product.SalePrice,

Amount = pc.Product.Amount,

MetaDesc = pc.Product.MetaDesc,

MetaKey = pc.Product.MetaKey,

CreateBy = pc.Product.CreateBy,

CreateAt = pc.Product.CreateAt,

UpdateBy = pc.Product.UpdateBy,

UpdateAt = pc.Product.UpdateAt,

Status = pc.Product.Status

}

)

.ToList();

break;

}

}

return list;

}

## CHỈNH sỬA TRANG index.CSHTML

<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.CatID)</th>

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

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

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

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

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

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

<th>Chức năng</th>

</tr>

</thead>

<tbody>

@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.CatName)</td>

<td>@Html.DisplayFor(modelItem => item.SupplierName[[3]](#footnote-3))</td>

<td><imgsrc="~/Public/img/product/@item.Image"alt="Img"class="img-fluid"/></td>

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

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

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

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

<td>

@if (item.Status == 1)

{

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

}

else

{

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

}

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

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

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

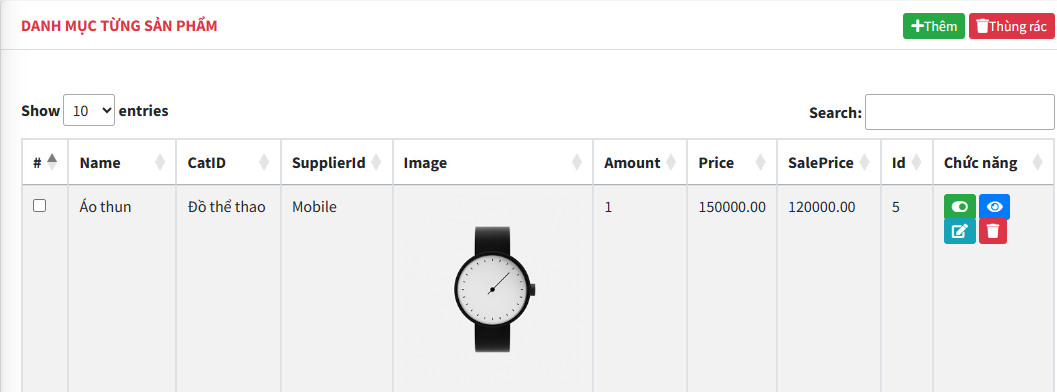
</td>

</tr>

}

</tbody>

</table>



## Trang Trash.cshtml

@model IEnumerable<MyClass.Model.ProductInfo>

<tbody>

@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.CatName)</td>

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

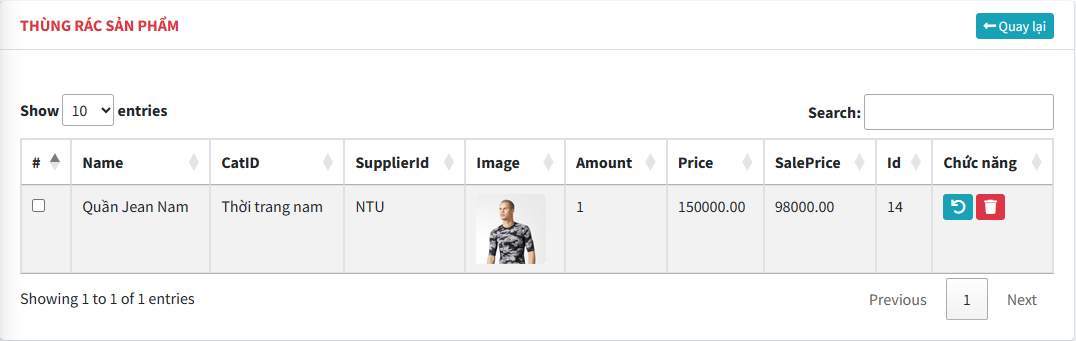
…

</td>

</tr>

}

</tbody>



#### ĐỐI VỚI TRANG HOMEP/HOME PRODUCT THÌ KHÔNG CẦN CHỈNH SỬA NỮA

1. Chuyển thành ProductInfo [↑](#footnote-ref-1)
2. Thay thế bằng ProductInfo [↑](#footnote-ref-2)
3. Hiển thị tên của nhà cung cấp [↑](#footnote-ref-3)