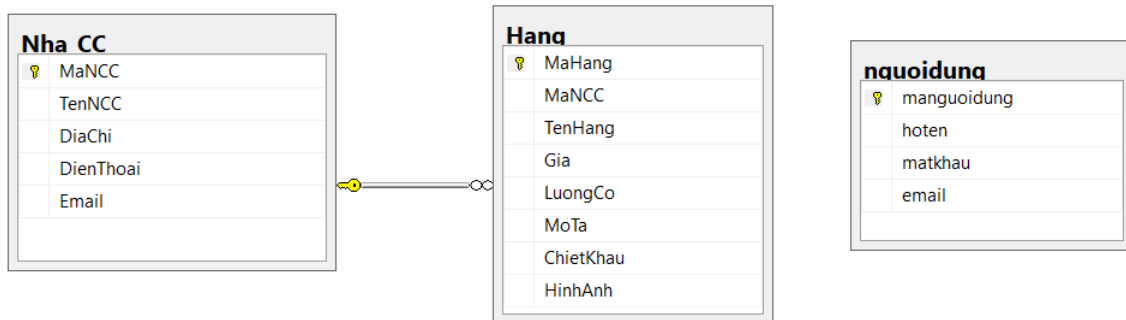


# Bài tập 5 – Phần 1

## Lab 12 – Sắp xếp, lọc và phân trang

- Trong folder HoaQua được cung cấp, chạy file script **fShopDB.sql** trong SQLServer để tạo cơ sở dữ liệu **fShopDB**



- Tạo một project đặt tên là BaiTap12, chọn mẫu **MVC**.
- Cài đặt EntityFramework sử dụng NuGet Package Manager
- Sử dụng EF code first để kết nối với cơ sở dữ liệu **fShopDB** (đặt tên ADO.NET Entity Model là **fShopDB**) chọn tất cả các bảng trong cơ sở dữ liệu.
- Kích **ReBuild Solution** để build lại project (mỗi khi sửa model cần Rebuild lại hệ thống)
- Tạo một controller theo mẫu **MVC 5 Controller with view, using Entity Framework** gắn với model Hang.
- Sửa phần ActionLink trong \_Layout.cshtml thành



```
...
<ul class="nav navbar-nav">
    <li>@Html.ActionLink("Home", "Index", "Home")</li>
    <li>@Html.ActionLink("Xem hàng", "Index", "Hangs")</li>
</ul>
...
```

### 1. Sắp xếp

- Sửa code trang Index.cshtml của controller Hangs:
  - Bỏ các trường MaNCC, LuongCo, MoTa, ChietKhau.
  - Sửa phần tiêu đề cột <th>...</th> như sau:

```
<table class="table">
    <tr>
        <th>
            Mã hàng
        </th>
        <th>
            Tên hàng
        </th>
        <th>
            Giá
        </th>
        <th>
            Hình ảnh
        </th>
    </tr>
    <tr>
        <td>
            ...
        </td>
    </tr>
</table>
```

- Xóa nút Create, sửa phần ActionLink của Edit, Details và Delete để khi chạy hiển thị như sau:

Application name    Home    Xem hàng				
Danh sách hàng				
Mã hàng	Tên hàng	Giá	Hình ảnh	
RYL01	Bơ Sáp Đặc Lắc	75000.00		<a href="#">Chi tiết</a>   <a href="#">Thêm vào giỏ</a>
			bo_sap.jpg	
RYL02	Bưởi da xanh	98000.00		<a href="#">Chi tiết</a>   <a href="#">Thêm vào giỏ</a>
			bui_da_xanh_2.jpg	

- Sửa action method Index() trong HangsController như sau.

```
public ActionResult Index()
{
    //Lấy danh sách hàng
    var hangs = db.Hangs.Select(p=>p);
    return View(hangs.ToList());
}
```

- Ấn phím F5 (hoặc Ctrl+F5) để chạy thử.
- Sửa action method Index() trong HangsController để sắp xếp danh sách hàng theo TenHang và Gia như sau:

```
public ActionResult Index(string sortOrder)
{
    //Các biến sắp xếp
    ViewBag.SapTheoTen = String.IsNullOrEmpty(sortOrder) ? "name_desc" : "";
    ViewBag.SapTheoGia = sortOrder == "Gia" ? "gia_desc" : "Gia";
    //Lấy danh sách hàng
    var hangs = db.Hangs.Select(p=>p);
    //Sắp xếp
    switch (sortOrder)
    {
        case "name_desc":
            hangs = hangs.OrderByDescending(s => s.TenHang);
            break;
        case "Gia":
            hangs = hangs.OrderBy(s => s.Gia);
            break;
        case "gia_desc":
            hangs = hangs.OrderByDescending(s => s.Gia);
            break;
        default:
            hangs = hangs.OrderBy(s => s.TenHang);
            break;
    }
    return View(hangs.ToList());
}
```

- Sửa code trang Index.cshtml của controller Hangs phần tiêu đề cột <th>...</th> như sau:

```
<table class="table">
    <tr>
        <th>
            Mã hàng
        </th>
        <th>
            @Html.ActionLink("Tên hàng", "Index", new { sortOrder = ViewBag.SapTheoTen })
        </th>
        <th>
            @Html.ActionLink("Giá", "Index", new { sortOrder = ViewBag.SapTheoGia })
        </th>
        <th>
            Hình ảnh
        </th>
    </tr>
</table>
...

```

- Ấn phím F5 (hoặc Ctrl+F5) để chạy thử.

## 2. Lọc

- Sửa action method Index() trong HangsController để cho phép tìm kiếm hàng theo tên như sau:

```
public ActionResult Index(string sortOrder, string searchString)
{
    //Các biến sắp xếp
    ViewBag.SapTheoTen = String.IsNullOrEmpty(sortOrder) ? "name_desc" : "";
    ViewBag.SapTheoGia = sortOrder == "Gia" ? "gia_desc" : "Gia";
    //Lấy danh sách hàng
    var hang = db.Hangs.Select(p=>p);

    //Lọc theo tên hàng
    if (!String.IsNullOrEmpty(searchString))
    {
        hang = hang.Where(p => p.TenHang.Contains(searchString));
    }
    //Sắp xếp
    switch (sortOrder)
    {
        case "name_desc":
            hang = hang.OrderByDescending(s => s.TenHang);
            break;
        case "Gia":
            hang = hang.OrderBy(s => s.Gia);
            break;
        case "gia_desc":
            hang = hang.OrderByDescending(s => s.Gia);
            break;
        default:
            hang = hang.OrderBy(s => s.TenHang);
            break;
    }
    return View(hang.ToList());
}
```

- Sửa code trang Index.cshtml của controller Hangs thêm textbox tìm kiếm như sau:

```
<h2>Danh sách hàng</h2>
@using (Html.BeginForm())
{
    <p>
        Tìm mặt hàng: @Html.TextBox("SearchString")
        <input type="submit" value="Tìm" />
    </p>
}
```

- Ấn phím F5 (hoặc Ctrl+F5) để chạy thử.

## 3. Phân trang đơn giản

- Thêm ActionLink sau vào menu trong \_Layout: <li>@Html.ActionLink("Xem nhà cung cấp", "DisplaySupplier", "Home")</li>
- Sử dụng Nuget cài gói PagedList.Mvc
- Tạo một view tên là DisplaySupplier theo Template List của model Nha\_CC.
- Sửa HomeController thêm một action method là DisplaySupplier:

```
public class HomeController : Controller
{
    private FShopDB = new FShopDB();
    public ActionResult DisplaySupplier()
    {
        var supplies = db.Nha_CC.Select(s => s);
        return View(supplies.ToList());
    }
}
```

- Ấn phím F5 (hoặc Ctrl+F5) để chạy thử.

- Trong HomeController:

- Thêm khai báo namespaces

```
using PagedList;
```

- Sửa action method `DisplaySupplier()`

```
public ActionResult DisplaySupplier(int? page)
{
    var supplies = db.Nha_CC.Select(s => s);

    //Cần sắp xếp trước khi phân trang
    supplies = supplies.OrderBy(s => s.MaNCC);

    int pageSize = 3; //Kích thước trang
    int pageNumber = (page ?? 1); //Nếu page bằng null thì trả về 1

    return View(supplies.ToPagedList(pageNumber, pageSize));
}
```

- `DisplaySupplier.cshtml` của controller Home trong folder Home:

- Sửa khai báo

```
@model IEnumerable<BaiTap12.Models.Nha_CC>
```

Thành

```
@model PagedList.IPagedList<BaiTap12.Models.Nha_CC>
@using PagedList.Mvc;
<link href="~/Content/PagedList.css" rel="stylesheet" type="text/css" />
```

- Thêm đoạn code chuyển trang sau vào cuối trang

```
<br />
Trang @(Model.PageCount < Model.PageNumber ? 0 : Model.PageNumber) / @Model.PageCount

@Html.PagedListPager(Model, page => Url.Action("DisplaySupplier", new { page }))
```

- Ấn phím F5 (hoặc Ctrl+F5) để chạy thử.

#### 4. Phân trang có cả sắp xếp và tìm kiếm

- Trong HangsController:

- Thêm khai báo namespaces

```
using PagedList;
```

- Sửa action method `Index()`

```
public ActionResult Index(string sortOrder, string searchString, string currentFilter, int?
page)
{
    //Các biến sắp xếp
    ViewBag.CurrentSort = sortOrder; //Biến lấy yêu cầu sắp xếp hiện tại

    ViewBag.SapTheoTen = String.IsNullOrEmpty(sortOrder) ? "name_desc" : "";
    ViewBag.SapTheoGia = sortOrder == "Gia" ? "gia_desc" : "Gia";

    //Lấy giá trị của bộ lọc dữ liệu hiện tại
    if (searchString != null)
    {
        page = 1; //Trang đầu tiên
    }
    else
    {
        searchString = currentFilter;
    }
    ViewBag.CurrentFilter = searchString;

    //Lấy danh sách hàng
```

```

var hangs = db.Hangs.Select(p => p);
//Lọc theo tên hàng
if (!String.IsNullOrEmpty(searchString))
{
    hangs = hangs.Where(p => p.TenHang.Contains(searchString));
}

//Sắp xếp
switch (sortOrder)
{
    case "name_desc":
        hangs = hangs.OrderByDescending(s => s.TenHang);
        break;
    case "Gia":
        hangs = hangs.OrderBy(s => s.Gia);
        break;
    case "gia_desc":
        hangs = hangs.OrderByDescending(s => s.Gia);
        break;
    default:
        hangs = hangs.OrderBy(s => s.TenHang);
        break;
}

int pageSize = 3; //Kích thước trang
int pageNumber = (page ?? 1);

return View(hangs.ToPagedList(pageNumber, pageSize));
}

```

- Trong trang Index.cshtml của controller Hangs

- Sửa khái báo

```
@model IEnumerable<BaiTap12.Models.Hang>
```

Thành

```

@model PagedList.IPagedList<BaiTap12.Model.Hang>
@using PagedList.Mvc;
<link href="~/Content/PagedList.css" rel="stylesheet" type="text/css" />

```

- Sửa

```

@using (Html.BeginForm())
{
    <p>
        Tìm mặt hàng: @Html.TextBox("SearchString")
        <input type="submit" value="Tìm" />
    </p>
}

```

Thành

```

@using (Html.BeginForm())
{
    <p>
        Tìm mặt hàng: @Html.TextBox("SearchString", ViewBag.CurrentFilter as string)
        <input type="submit" value="Tìm" />
    </p>
}

```

- Sửa

```

<th>
    @Html.ActionLink("Tên hàng", "Index", new { sortOrder = ViewBag.SapTheoTen })
</th>
<th>
    @Html.ActionLink("Giá", "Index", new { sortOrder = ViewBag.SapTheoGia })
</th>

```

Thành

```

<th>
    @Html.ActionLink("Tên hàng", "Index", new { sortOrder = ViewBag.SapTheoTen,
currentFilter = ViewBag.CurrentFilter })
</th>
<th>
    @Html.ActionLink("Giá", "Index", new { sortOrder = ViewBag.SapTheoGia,
currentFilter = ViewBag.CurrentFilter })
</th>

```

- Thêm đoạn code chuyển trang sau vào cuối trang

```

</table>

<br />
Trang @(Model.PageCount < Model.PageNumber ? 0 : Model.PageNumber) / @Model.PageCount

@Html.PagedListPager(Model, page => Url.Action("Index",
    new { page, sortOrder = ViewBag.CurrentSort, currentFilter = ViewBag.CurrentFilter
}))

```

- Ấn phím F5 (hoặc Ctrl+F5) để chạy thử