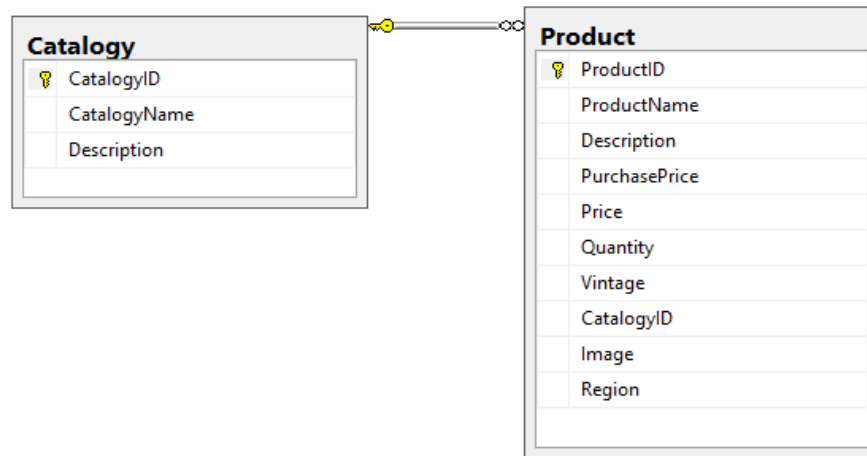


BÀI 11. SỬ DỤNG LINQ ĐỂ HIỂN THỊ DỮ LIỆU CÓ SẮP XẾP, LỌC VÀ PHÂN TRANG

- Tạo cơ sở dữ liệu **WineStore**



```

--Tạo cơ sở dữ liệu
CREATE DATABASE [WineStore]
GO
USE [WineStore]
GO
--Tạo bảng Catalogy
CREATE TABLE [dbo].[Catalogy](
    [CatalogyID] [nchar](10) NOT NULL PRIMARY KEY,
    [CatalogyName] [nvarchar](50) NOT NULL,
    [Description] [nvarchar](100) NULL
)
GO
--Tạo bảng Product
CREATE TABLE [dbo].[Product](
    [ProductID] [int] NOT NULL PRIMARY KEY,
    [ProductName] [nvarchar](50) NOT NULL,
    [Description] [text] NULL,
    [PurchasePrice] [numeric](8, 2) NOT NULL,
    [Price] [numeric](8, 2) NOT NULL,
    [Quantity] [int] NOT NULL,
    [Vintage] [nchar](20) NULL,
    [CatalogyID] [nchar](10) NOT NULL,
    [Image] [text] NULL,
    [Region] [nvarchar](100) NOT NULL
)
GO
--Tạo các khóa ngoài
ALTER TABLE [dbo].[Product] WITH CHECK ADD CONSTRAINT [FK_Products_Catalogy] FOREIGN
KEY([CatalogyID])
REFERENCES [dbo].[Catalogy] ([CatalogyID])
GO
--Chèn dữ liệu cho bảng Catalogy
Insert into Catalogy(CatalogyID,CatalogyName,Description) values('01','Rượu vang','Đây
là rượu vang' )
Insert into Catalogy(CatalogyID,CatalogyName,Description) values('02','Rượu VODKA','Đây
là rượu VODKA' )
  
```

--Chèn dữ liệu cho bảng Product

Insert into

Product(ProductID,ProductName,Description,PurchasePrice,Price,Quantity,Vintage,CatalogyID,Image,Region)

values(1,'RƯỢU VANG PETRUS',N'Rượu vang đỏ Chateau Petrus Pomerol, huyền thoại của của thương hiệu vang hàng đầu thế giới. Vườn nho nhà Petrus tọa lạc trên một cao nguyên ở phía đông Pomerol, tiểu vùng sản xuất rượu vang đỏ thượng hạng của xứ Bordeaux, Pháp. Đặc điểm của đất nơi đây là đất sét xanh, chỉ thích với giống nho Merlot (chiếm 95%) và Cabernet Franc.',600,800,200,1992,'01','h1.png',N'Pháp')

Insert into

Product(ProductID,ProductName,Description,PurchasePrice,Price,Quantity,Vintage,CatalogyID,Image,Region)

values(2,'RƯỢU VODKA PUTINKA LIMITED',N'Rượu Vodka Putinka limited Edition Chai Vuông (Vodka Putinka Limited Edition) thuộc Moscow Distillery Cristall, Nga. Rượu Putinka ra đời năm 2003 nhằm tôn vinh Tổng thống Putin.',800,900,140,1990,'02','h2.png',N'Nga')

...

1.1. Hiển thị dữ liệu

- Tạo một project đặt tên là BaiTap10, chọn mẫu **MVC**.
- Cài đặt EntityFramework sử dụng NuGet Package Manager
- Sử dụng EF code first để kết nối với cơ sở dữ liệu **WineStore** (đặt tên ADO.NET Entity Model là **WineStoreDB**) 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 Product.
- Sửa phần ActionLink trong _Layout.cshtml thành

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

F5 chạy thử

- Sửa code trang **Index.cshtml** của controller Products:
 - Bỏ các trường CatalogyID, Description, PurchasePrice, Quantity, Vintage, Region.
 - Sửa phần tiêu đề cột <th>...</th>
 - Xóa nút Create, sửa phần ActionLink của Edit, Details và Delete

```

<h2>Danh sách hàng</h2>
<table class="table">
  <tr>
    <th>
      Mã hàng
    </th>
    <th>
      Tên hàng
    </th>
    <th>
      Giá
    </th>
    <th>
      Hình ảnh
    </th>
  </tr>

```

```

@foreach (var item in Model)
{
  <tr>
    <td>
      @Html.DisplayFor(modelItem => item.ProductID)
    </td>
    <td>
      @Html.DisplayFor(modelItem => item.ProductName)
    </td>
    <td>
      @Html.DisplayFor(modelItem => item.Price)
    </td>
    <td>
      @{ //Lấy đường dẫn file ảnh
        string ImagePath = "~/wwwroot/WineImages/" + item.Image;
      }
      
      <br />
      @Html.DisplayFor(modelItem => item.Image)
    </td>
    <td>
      @Html.ActionLink("Chi tiết", "Details", new { id = item.ProductID }) |
      @Html.ActionLink("Thêm giỏ hàng", "", new { id = item.ProductID })
    </td>
  </tr>
}
</table>

```



- Sửa action method **Index()** trong ProductsController như sau.

```

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

```

F5 chạy thử, hiển thị như sau:

Application name Home Xem sản phẩm				
Danh sách hàng				
Mã hàng	Tên hàng	Giá	Hình ảnh	
1	RƯỢU VANG BIN 555	320.00	 h9.png	Chi tiết Thêm giỏ hàng
2	RƯỢU VANG LOUIS PINEL MERLOT IGP	360.00	 h2.png	Chi tiết Thêm giỏ hàng

1.2. Sắp xếp

- Sửa action method **Index()** trong **ProductsController** để sắp xếp danh sách hàng theo **ProductName** và **Price** như sau:

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

- Sửa code trang **Index.cshtml** trong View/Products phần tiêu đề cột `<th>...</th>` như sau:




```

<h2>Danh sách hàng</h2>
<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>
  <tr>
    <td>
      1
    </td>
    <td>
      RƯỢU VANG BIN 555
    </td>
    <td>
      320.00
    </td>
    <td>
      <img alt="Bottle of Bin 555 wine" data-bbox="648 458 718 508"/>
      <a href="#">Chi tiết</a> | <a href="#">Thêm giỏ hàng</a>
    </td>
  </tr>
  <tr>
    <td>
      2
    </td>
    <td>
      RƯỢU VANG LOUIS PINEL MERLOT IGP
    </td>
    <td>
      360.00
    </td>
    <td>
      <img alt="Bottle of Louis Pinel Merlot wine" data-bbox="648 528 718 578"/>
      <a href="#">Chi tiết</a> | <a href="#">Thêm giỏ hàng</a>
    </td>
  </tr>
  <tr>
    <td>
      3
    </td>
    <td>
      RƯỢU VANG MAISON CASTEL CHARDONNAY
    </td>
    <td>
      380.00
    </td>
    <td>
      <img alt="Bottle of Maison Castel Chardonnay wine" data-bbox="648 598 718 648"/>
      <a href="#">Chi tiết</a> | <a href="#">Thêm giỏ hàng</a>
    </td>
  </tr>
</table>

```

- Ấn phím F5 (hoặc Ctrl+F5) để chạy thử. Click chuột vào liên kết tiêu đề cột Tên hàng, Giá

Application name	Home	Xem sản phẩm
------------------	------	--------------

Danh sách hàng				
Mã hàng	Tên hàng	Giá	Hình ảnh	
1	RƯỢU VANG BIN 555	320.00	 h4.png	Chi tiết Thêm giỏ hàng
2	RƯỢU VANG LOUIS PINEL MERLOT IGP	360.00	 h2.png	Chi tiết Thêm giỏ hàng
3	RƯỢU VANG MAISON CASTEL CHARDONNAY	380.00	 h3.png	Chi tiết Thêm giỏ hàng

1.3. Lọc

- Sửa action method **Index()** trong ProductsController 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) ? "ten_desc" : "";
    ViewBag.SapTheoGia = sortOrder == "gia" ? "gia_desc" : "gia";
    //Lấy danh sách hàng
    var products = db.Products.Select(p => p);



    //Lọc theo tên hàng
    if (!String.IsNullOrEmpty(searchString))
    {
        products = products.Where(p => p.ProductName.Contains(searchString));
    }

    //Sắp xếp
    switch (sortOrder)
    {
        case "ten_desc":
            products = products.OrderByDescending(s => s.ProductName);
            break;
        case "gia":
            products = products.OrderBy(s => s.Price);
            break;
        case "gia_desc":
            products = products.OrderByDescending(s => s.Price);
            break;
        default:
            products = products.OrderBy(s => s.ProductName);
            break;
    }
    return View(products.ToList());
}
```

- Sửa code trang **Index.cshtml** trong View/Products, 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>
}
<table class="table">
....
```

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

Application name Home Xem sản phẩm				
Danh sách hàng <div> Tìm mặt hàng: <input type="text" value="vodka"/> <input type="button" value="Tìm"/> </div>				
Mã hàng	Tên hàng	Giá	Hình ảnh	
7	RƯỢU VODKA CỎ ZUBROWKA	600.00	 h9.png	Chi tiết Thêm giỏ hàng
6	RƯỢU VODKA PUTINKA LIMITED	750.00	 h6.png	Chi tiết Thêm giỏ hàng

1.4. Phân trang đơn giản

- Thêm `NavLink` sau vào menu trong `_Layout`:

```
<li>@Html.ActionLink("Xem danh mục", "Display", "Catalogies")</li>
```
- Sử dụng Nuget cài gói **PagedList.Mvc**: *Kích chuột phải vào tên project và chọn Manage NuGet Packages..., chọn PagedList.Mvc => Install*
- Sửa **CatalogiesController** thêm một action method là **Display**:

```
using Bai10.Models;
namespace Bai10.Controllers
{
    public class CatalogiesController : Controller
    {
        private WineStoreDB db = new WineStoreDB();
        public ActionResult Index()
        {
            return View(db.Catalogies.ToList());
        }
        public ActionResult Display()
        {
            var danh muc = db.Catalogies.Select(s => s);
            return View(danh muc.ToList());
        }
    }
}
```

- Tạo một view tương ứng tên là **Display** theo Template List của model **Catalogy**.

Add View
 ✕

View name:

Template:

Model class:

Data context class:

Options:

☐ Create as a partial view
 ☒ Reference script libraries
 ☒ Use a layout page:

...

(Leave empty if it is set in a Razor _viewstart file)

Add

Cancel

- Ấn phím F5 (hoặc Ctrl+F5) để chạy thử.
- Trong **CatalogiesController**:
 - Thêm khai báo namespaces

```
using PagedList;
```

- Sửa action method **Display()**

```
public ActionResult Display(int? page)
{
    var danhmuc = db.Catalogies.Select(s => s);

    
        //Cần sắp xếp trước khi phân trang
        danhmuc = danhmuc.OrderBy(s => s.CatalogyID);
        int pageSize = 3; //Kích thước trang
        int pageNumber = (page ?? 1); //Nếu page bằng null thì trả về 1
        return View(danhmuc.ToPagedList(pageNumber, pageSize));
    
}
```

- **Display.cshtml** trong View/Catalogies:

- Sửa khai báo

```
@model IEnumerable<BaiTap10.Models.Nha_CC>
```

Thành

```
@model PagedList.IPagedList<Bai10.Models.Catalogy>
@using PagedList.Mvc;
<link href="~/Content/PagedList.css" rel="stylesheet" type="text/css" />
```


- Sửa tiêu đề bảng

```
<table class="table">
  <tr>
    <th>
      Tên danh mục
    </th>
    <th>
      Mô tả
    </th>
    <th></th>
  </tr>
  . . .
```

- 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("Display", new { page }))
```

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

Tên danh mục	Mô tả	
Rượu vang	Đây là rượu vang	Edit Details Delete
Rượu Vodka	Đây là rượu Vodka	Edit Details Delete
Rượu mơ	Đây là rượu mơ Nhật	Edit Details Delete

Trang 1 / 2

1 2 »

1.5. Phân trang có cả sắp xếp và tìm kiếm

- Trong ProductsController:
 - 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) ? "ten_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 products = db.Products.Select(p => p);

    //Lọc theo tên hàng
    if (!String.IsNullOrEmpty(searchString))
    {
        products = products.Where(p => p.ProductName.Contains(searchString));
    }

    //Sắp xếp
    switch (sortOrder)
    {
        case "ten_desc":
            products = products.OrderByDescending(s => s.ProductName);
            break;
        case "gia":
            products = products.OrderBy(s => s.Price);
            break;
        case "gia_desc":
            products = products.OrderByDescending(s => s.Price);
            break;
        default:
            products = products.OrderBy(s => s.ProductName);
            break;
    }

    int pageSize = 3; //Kích thước trang
    int pageNumber = (page ?? 1); //Nếu page bằng null thì trả về 1
    return View(products.ToPagedList(pageNumber, pageSize));
}
```

- Trang **Index.cshtml** trong View/Products

- Sửa khái báo

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

Thành

```
@model PagedList.IPagedList<Bai10.Models.Product>
@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

