TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH

**KHOA CÔNG NGHỆ THÔNG TIN**

-----🙡 🕮 🙣 -----



**BÀI TẬP LỚN MÔN HỌC**

**LẬP TRÌNH CƠ SỞ DỮ LIỆU**

**MÔ TẢ KỸ THUẬT VÀ CÔNG NGHỆ**

***Đề tài:* Xây dựng website bán mỹ phẩm**

**Giảng viên hướng dẫn:**  **Lê Ngọc Hiếu**

**Nhóm sinh viên thực hiện: Nhóm 18**

|  |  |  |
| --- | --- | --- |
|  | **Đinh Quốc Việt** | **1954052122** |
|  | **Lê Thị Hiền Linh** | **1854030187** |
|  | **Nguyễn Thị Như Quỳnh** | **1954052081** |
|  | **Võ Minh Hiếu** | **1954052028** |
|  | **Phạm Thụy** | **1954052097** |

**Lớp:** **TH109-IM1901**

**Tháng 1 năm 2022**

**GIỚI THIỆU CÔNG NGHỆ ASP.NET MVC**

Những ai yêu thích lập trình web sử dụng công nghệ ASP.NET của Microsoft thì chắc hẳn một điều rằng đã nghe qua Webform - một công nghệ khá cũ của Microsoft giúp lập trình web. Tuy nhiên khi sử dụng công nghệ ASP.Net webform thì không thể tránh khỏi những nhược điểm và một trong nhược điểm đáng chú ý là giao diện webform phải sử dụng toolbox sẵn có, do đó chúng ta sẽ mất đi tính linh hoạt trong việc điều khiển giao diện.

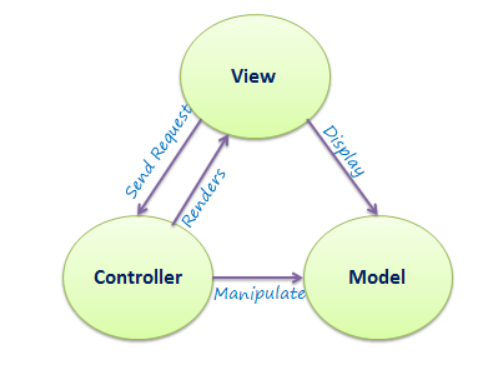
Vì vậy, Microsoft đã cho ra đời một công nghệ lập trình web mới đó là ASP.Net MVC với rất nhiều ưu điểm.

Trong lập trình web chúng ta chia ra 3 tầng như sau:

* Presentation Layer (tầng 3): tầng này giúp hiển thị giao diện
* Business Logic Layer (tầng 2): tầng này chứa các hàm thực thi, giúp xử lý sự kiện xảy ra trên tầng Presentation Layer.
* Data Access Layer (tầng 1): tầng này chứa hàm kết nối với các hệ quản trị cở sở dữ liệu như SQL Server, MySql, Oracle …

Và bây giờ chúng ta sẽ cùng tìm hiểu về mô hình MVC:

MVC (Model – View - Controller) là một design partern đã tồn tại rất lâu trong ngành công nghệ phần mềm. Một ứng dụng viết theo mô hình MVC sẽ bao gồm 3 thành phần tách biệt nhau đó là Model, View, Controller. Giống như trong cấu trúc Three – Tier, mô hình MVC giúp tách biệt 3 tầng trong mô hình lập trình web, vì vậy giúp tối ưu ứng dụng, dễ dàng thêm mới và chỉnh sửa code hoặc giao diện.



**Model:** Trong mô hình Three – Tier thì trong đó gồm có 2 tầng Data Access Layer và tầng Business Logic Layer. Hai tầng này là hai tầng tương đương với tầng model trong mô hình MVC.

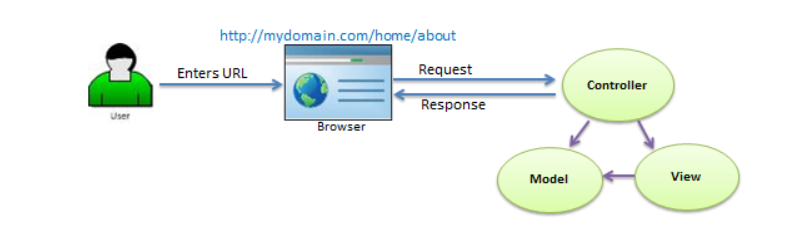
**View:** là tầng giao diện, hiển thị dữ liệu được truy xuất từ tầng model. Tầng này tương đương với tầng Presentation Layer trong cấu trúc Three – Tier.

**Controller:** đây là tầng giúp kết nối giữa tầng model và tầng view trong mô hình MVC, có nghĩa là nếu phía client yêu cầu hiển thị dữ liệu thì controller gọi giữ liệu từ model và trả về cho view vì view tương tác trực tiếp với client.

Tóm lại, hiểu đơn giản thì Model đại diện cho dữ liệu, View là giao diện người dùng, Controller xử lý yêu cầu của người dùng.

Những ưu điểm mà ASP.Net MVC mang lại cho chúng ta:

* Do sử dụng mô hình MVC nên trong ASP.Net MVC đã tách biệt được các tầng trong mô hình lập trình web vì vậy giúp tối ưu ứng dụng và dễ dàng trong việc viết code, giao diện.
* Giao diện trong ASP.Net MVCsử dụng công nghệ thiết kế web HTML, CSS nền việc thiết kế giao diện trở nên dễ dàng và giúp cho designer linh hoạt trong việc thiết kế.
* ASP.Net MVC không sử dụng view state vì vậy trang web không bị tăng kích thước do đó hiệu năng hoạt động không bị giảm.

**CƠ CHẾ HOẠT ĐỘNG CỦA ASP .NET MVC**

Dựa vào hình mô phỏng phía trên này thì chúng ta sẽ khái quá sơ qua cơ chế hoạt động của **ASP.Net MVC:**

1. User gửi 1 yêu cầu tới server bằng cách truyền vào 1 URL trong browser.
2. Yêu cầu đó được gửi tới controller đầu tiên, controller sẽ xử lý yêu cầu, nếu yêu cầu cần truy xuất dữ liệu thì controller sẽ chuyển qua tầng model.
3. Tại tầng model, dữ liệu được truy xuất từ database và sau đó truyền qua view thông qua controller.
4. Controller sẽ giúp dữ liệu được chuyển từ model qua view.
5. View là tầng cuối cùng giao tiếp với User, mọi dữ liệu sẽ được hiển thị cho User thông qua tầng View.

Các tính năng của nền tảng ASP.NET MVC

* Thừa hưởng tính minh bạch và cấu trúc rõ ràng của mô hình MVC.
* MVC là một nền tảng khả mở rộng (extensible) & khả nhúng (pluggable). Các thành phần của ASP.NET MVC được thiết kể để chúng có thể được thay thế một cách dễ dàng hoặc dễ dàng tùy chỉnh.
* ASP.NET MVC có cấu trúc URL tốt cho phép bạn xây dựng những ứng dụng có các địa chỉ URL súc tích và dễ tìm kiếm. Các địa chỉ URL không cần phải có phần mở rộng của tên tập tin.
* Hỗ trợ sử dụng các thẻ của các trang ASP.NET(.aspx) Usercontrol (.ascx) và trang master page (.marter).
* Hỗ trợ khá đầy đủ các tính năng của ASP.NET như data caching, seession và profile…
* ASP.NET MVC 3 bổ sung một view engine mới là Razor View Engine cho phép thiết lập các view nhanh chóng, dễ dàng.
* ASP.NET MVC 4 hỗ trợ tốt hơn cho các thiết bị di động bổ sung ASP.NET Web API.
* ASP.NET MVC 5 hỗ trợ chứng thực qua các API khác(facebook,google+..), Bootstrap được thêm vào hổ trợ phần thiết kế giao diện.

**GIỚI THIỆU ENTITY FRAMEWORK**

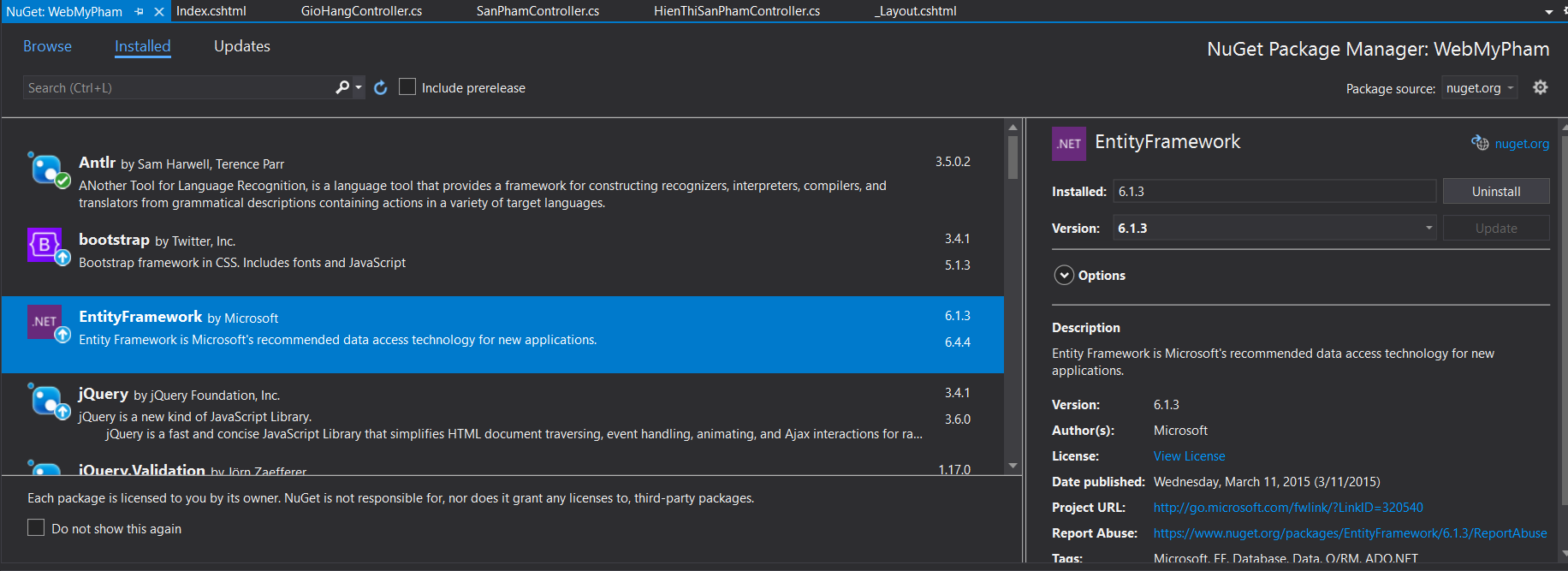
Entity Framework ra đời nhằm hỗ trợ sự tương tác giữa các ứng dụng trên nền tảng .NET với các cơ sở dữ liệu quan hệ. Hay Entity Frmework chính là công cụ giúp ánh xạ giữa các đối tượng trong ứng dụng, phần mềm của bạn với các bảng của một cơ sở dữ liệu quan hệ.

Emntity Framework giúp các nhà phát triển Web tương tác với dữ liệu quan hệ theo phương pháp hướng đối tượng với ít mã hơn so với các ứng dụng truyền thống. Lợi ích lớn nhất của nó là giúp lập trình viên giảm thiểu việc lập trình mã nguồn để thực hiện truy cập và tương tác với cơ sở dữ liệu.

Sử dụng Entity Framework sẽ đem lại những lợi ích sau:

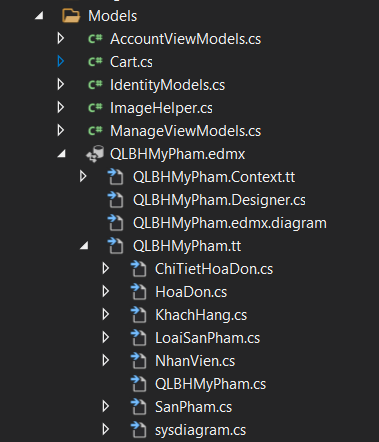
* Sử dụng Linq to Entities để thao tác với objects được sinh ra từ Entity Framework, nghĩa là bạn không phải viết code sql.
* Việc update các classes,commands dễ dàng mỗi khi cơ sở dữ liệu có sự thay đổi, điều này giúp tiết kiệm thời gian đáng kể.
* Entity Framework sẽ tự động tạo ra các classes, commands tương ứng cho việc select, insert, update,delete dữ liệu từ cơ sở dữ liệu quan hệ.
* Entity Framework tự động tạo ra các classes cho việc truy suất cơ sở dữ liệu giúp lập trình viên giảm được thời gian viết code thao tác với database. Hỗ trợ bạn không phải mất quá nhiều thời gian cho việc viết code để thao tác với database.

**ÁP DỤNG KỸ THUẬT VÀO TRONG ĐỀ TÀI CỦA NHÓM**

Trong bài tập lớn nhóm chúng em đã sử dụng Entity Framework để hỗ trợ thao tác dữ liệu một cách dễ dàng và nhanh chóng hơn.

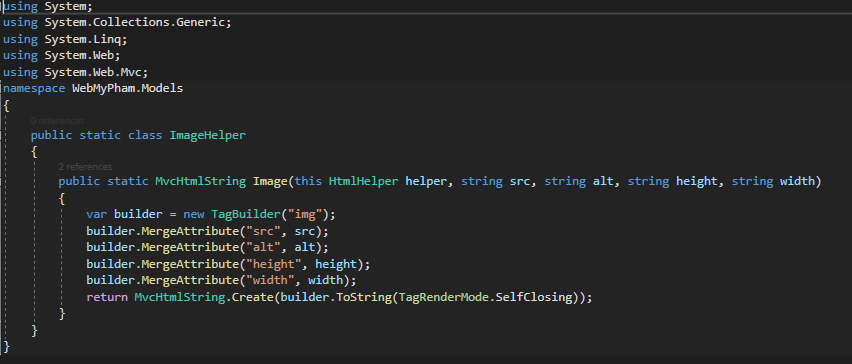
**Model:**

Sử dụng Entity Framework để tạo các model theo các trường dữ liệu các bảng ở database. Ngoài ra nhóm còn viết thêm các model để tương tác với dữ liệu khác.

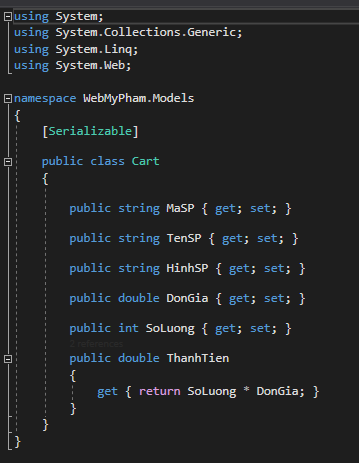


Ở đây có 2 model là AccountViewModels, IndentityModels và ManageViewModels được tạo tự động do project có sử dụng Identity Authentication được hỗ trợ sẵn.

ImageHelper được viết thêm để hiển thị ảnh lên View.

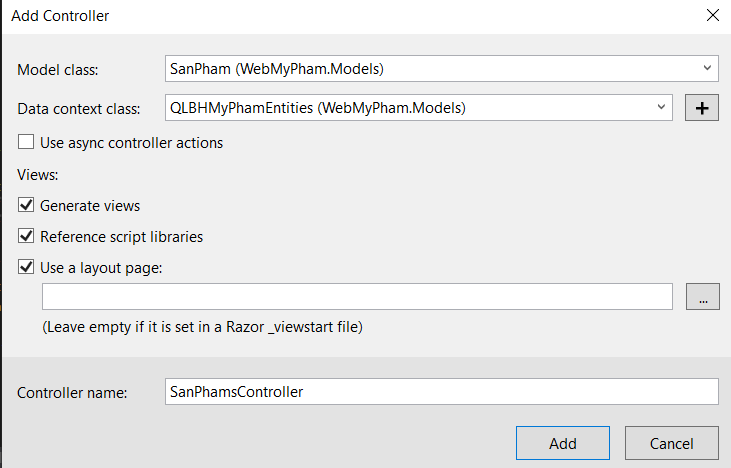


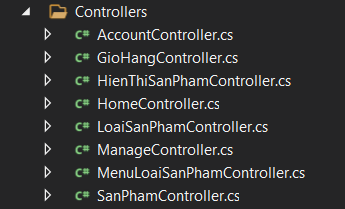
Và một đối tượng có tên là Cart sẽ chứa các thuộc tính của một giỏ hàng.



**Controller:**

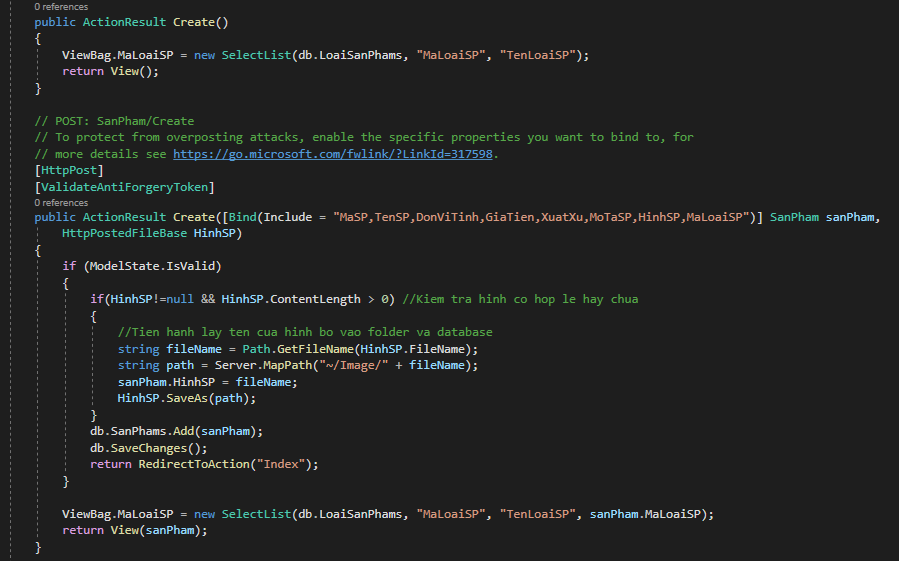
Tạo controller sử dụng Entity Frameswork với các model đã được tạo sử dụng data context cũng là 1 model.





Với các controller được tạo sử dụng Entity Framework sẽ tự động sinh các Action “Create”, “Details”, “Delete”.

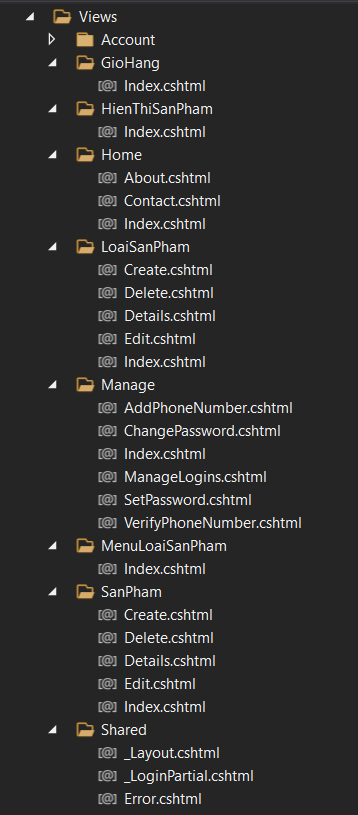
Và dựa trên code của các Action tự sinh, chúng em sẽ điều chỉnh lại code theo các chức năng của đề tài như hình bên dưới.





**View:**

Mỗi một controller chúng ta sẽ tạo view cho nó, mỗi một Action sẽ có một trang view tương ứng.

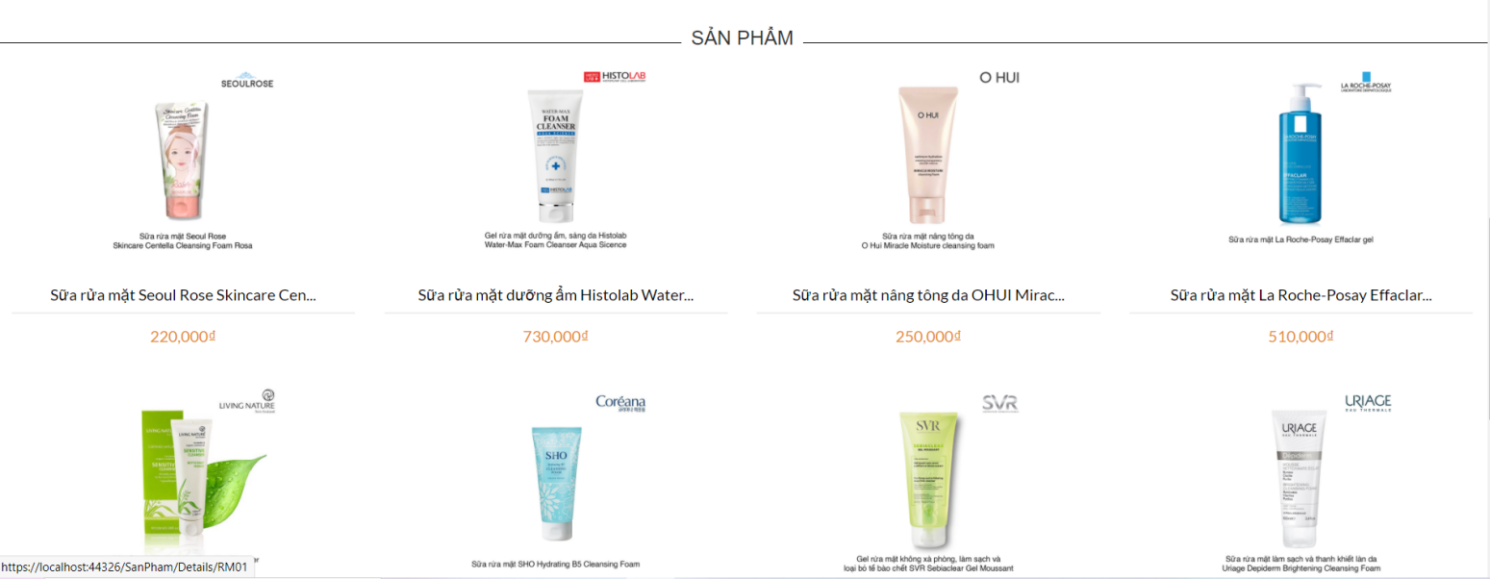


Với các controller dùng Entity Framework sẽ có các view tương ứng với các action cơ bản như “Index”, “Create”, “Detaills”, “Edit”, “Delete”.

Mỗi một chức năng của trang web thì chúng em sẽ viết lại view để phù hợp với đề tài. Và ASP .net đã hỗ trợ các thư viện Boostrap, jQuery để hỗ trợ viết nhanh hơn. Ngoài ra nhóm còn tự viết HTML, CSS, Javascript để trang web thêm phần đẹp mắt, giao diện hiện đại hơn.



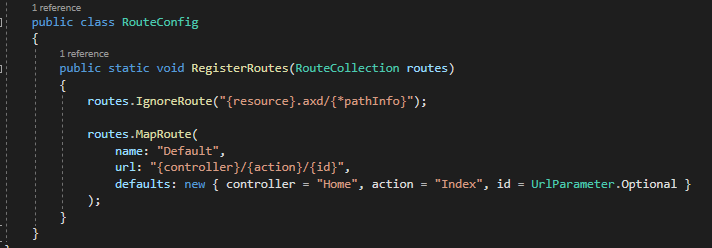
***Một đoạn code ở phần View trang Index của controller SanPham, thể hiện việc hiển thị dữ liệu được lấy từ Model***



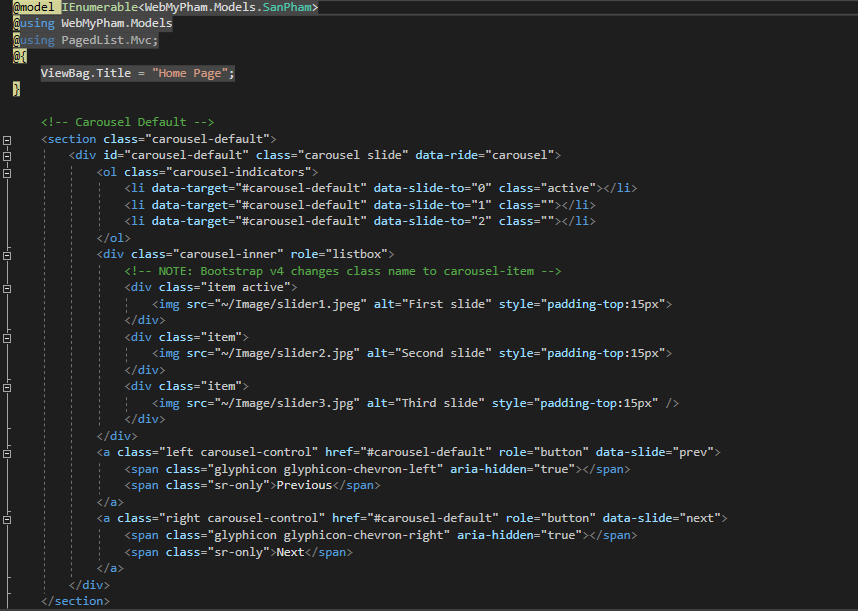
Với mỗi trang View sẽ có một kiểu trả về Model khác nhau tùy theo chức năng.



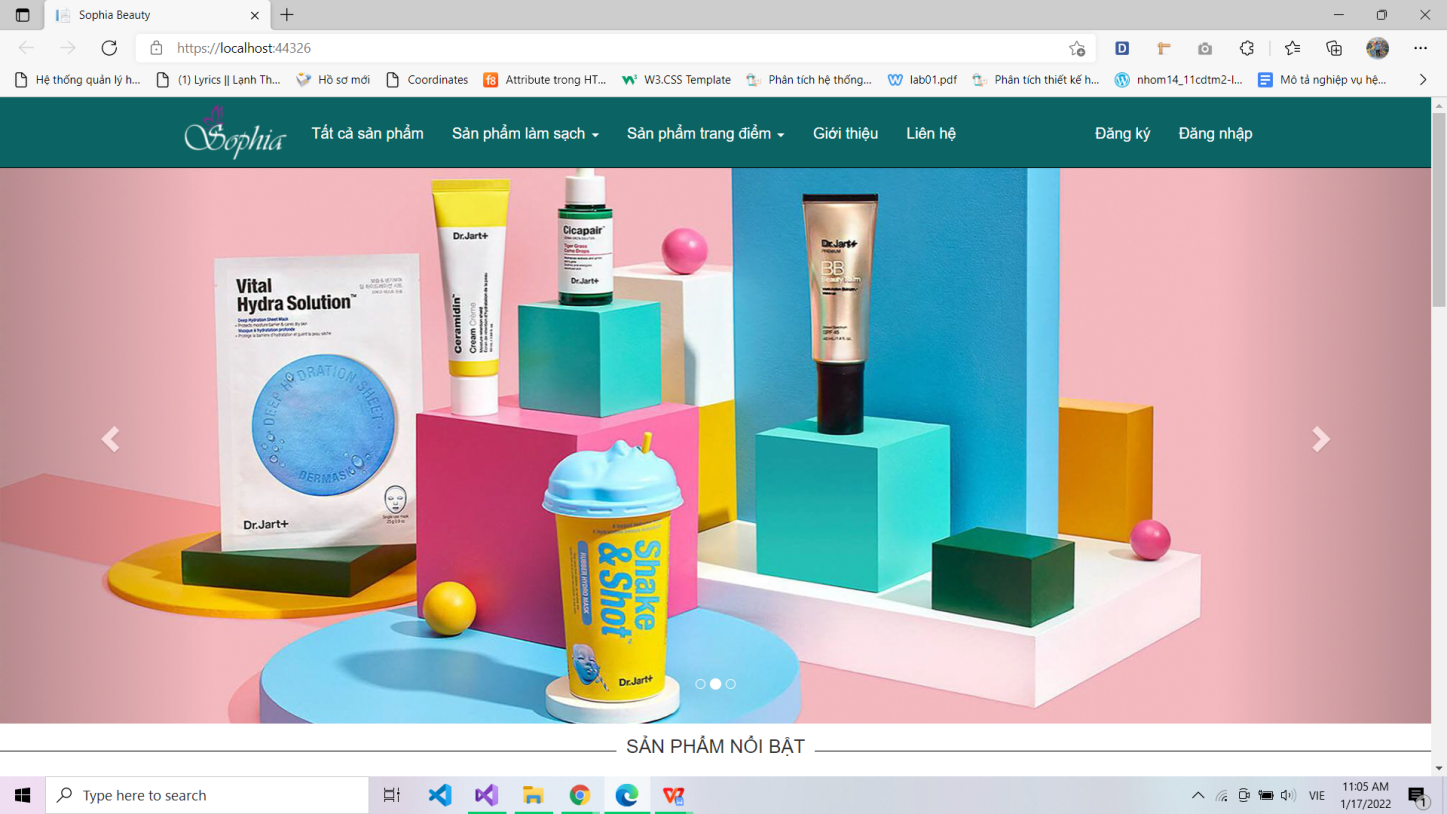
Trang Home mặc định sẽ là Trang chủ của website mỹ phẩm khi khách hàng vào web nên trong phần App\_Start/RouteConfig.cs chúng em sẽ truyền các tham số vào MapRoute như bên dưới.



Một đoạn code phần đầu trang Index của HomeController .



Và đây là kết quả của đoạn code bên trên:



**TÀI LIỆU THAM KHẢO**

|  |  |
| --- | --- |
| [1] | Series lập trình web MVC của cô Nguyễn Thị Mai Trang, Cunghoclaptrinh được truy cập tại link: <https://www.youtube.com/channel/UC-z7tRN3Ql02jyAx2yW5x1Q> |