|  |  |
| --- | --- |
| LOGO | **BỘ GIÁO DỤC VÀ ĐÀO TẠO**  **TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH**  **KHOA CÔNG NGHỆ THÔNG TIN** |

BÁO CÁO ĐỒ ÁN

LẬP TRÌNH WEB

1. **Tên đề tài:**

**Môn học : Lập trình web**

**Lớp : 20DTMDT1A**

**Giảng viên : BÙI DUY TÂN**

**SV thực hiện 1 : Nguyễn lê Quốc Bảo [2100004053]**

**SV thực hiện 2 : Lưu Hoàng bảo Duy [210000]**

***Học kỳ 3 Năm 2021 - 2022***

**LỜI MỞ ĐẦU**

Trong những năm gần đâu với sự chuyển giao của công nghệ 4.0 và đang lên 5.0 thi ngành công nghệ thông tin và mạng máy tính được ứng dụng rộng rãi với một trong những ứng dụng phổ biến trên mạng máy tính là Internet và các dịch vụ mạng trở nên không thể thiếu trong cuộc sống hiện đại. Để phục vụ học tập, nghiên cứu và đánh giá của sinh viên khoa CNTT, em xin phép biên soạn tài liệu đồ án bộ môn **lập trình Web**. Cuống tài liệu được biên soạn với phương châm đảm bảo tính logic, khoa học, thiết thực, dễ hiểu nhằm trang bị cho sinh viên những kiến thức cơ bản phục vụ cho việc nghiên cứ thực hành để lập trình ra một trang web hoàn chỉnh.

Tuy có rất nhiều cố gắng trong công tác biên soạn, nhưng vẫn không tránh khỏi thiếu sót. Trong phạm vi kiến thức hẹp hòi của cuốn tài liệu này, không thể đề cập hết các vấn đề nóng nổi trong lĩnh vực thiết kế và lập trình web vì thế rất mong sự đóng góp phê bình của đọc giả để tài liệu này được hoàn chỉnh hơn.

**LỜI CẢM ƠN**

Trong thời gian làm đồ án lập trình Web, em đã nhận được nhiều sự giúp đỡ, đóng góp ý kiến và chỉ bảo nhiệt tình của thầy cô, gia đình và bạn bè.

Em xin gửi lời cảm ơn chân thành đến thầy Bùi Duy Tân, giảng viên Bộ môn lập trình Web - trường ĐH Nguyễn Tất Thành người đã tận tình hướng dẫn, chỉ bảo em trong suốt quá trình làm đồ án.

Em cũng xin chân thành cảm ơn các thầy cô giáo trong trường ĐH Nguyễn Tất Thành nói chung, các thầy cô trong khoa Công Nghệ Thông Tin nói riêng đã dạy dỗ cho em kiến thức về các môn đại cương cũng như các môn chuyên ngành, giúp em có được cơ sở lý thuyết vững vàng và tạo điều kiện giúp đỡ em trong suốt quá trình học tập.

Cuối cùng, em xin chân thành cảm ơn gia đình và bạn bè, đã luôn tạo điều kiện, quan tâm, giúp đỡ, động viên em trong suốt quá trình học tập và hoàn đồ án môn lập trình Web.

**MỤC LỤC**

**DANH MỤC HÌNH ẢNH**

Hình 1.1 : Truyền dữ liệu từ view sang controller bằng Action Arguments

Hình 1.2 : Mô tả layout

Hình 1.3 : Mô tả layout

Hình 1.4 : Kiến trúc của Entity Framework

Hình 1.5 : Ảnh ánh xạ tương đương sau giữa các đối tượng trong database và EF:

Hình 1.6 :Entity Data Model

Hình 1.7 : Kiểu truy vấn

Hình 1.8 : Kiểu truy vấn

Hình 1.9 : Một vài phương thức truy vấn cơ bản

Hình 1.10 : Một vài phương thức truy vấn cơ bản

Hình 1.11 : Một vài phương thức truy vấn cơ bản

Hình 1.12 : Một vài phương thức truy vấn cơ bản

Hình 2.1: Diagram database

**Nội dung báo cáo Đồ án môn học Lập trình Web**

Thông tin chung

Tên đồ án: Lập trình website thương mại điện tử bán xe

Nhóm :

* Nguyễn Lê Quốc Bảo - 2100004053
* Lưu Hoàng Bảo Duy - 210000

Phân công thực hiện :

* Nguyễn Lê Quốc Bảo :
* Thực hiện đổ layout vào asp.net và viết các chức năng của web
* Viết báo cáo
* Lưu Hoàng Bảo Duy :
* Thiết kế layout
* Làm database
* Viết báo cáo

# **Chương 1. Tổng quan**

## 1.1.Tóm tắc về website

* Website thuộc loại thương mại điện tử chủ đề chính của website là bán xe máy, ngoài ra tên của cửa hàng là Thảo My tọa lạc tại Đà lạt
* Tổng quan các chức năng sẽ thực hiện
* Có 2 phân vùng chính là phân vùng người dùng và phân vùng quản trị. Phân vùng người dụng sẽ tương tác được các chức năng như xem, đặt hàng, đăng nhập, đăng kí. Phân vùng quản trị sẽ tương tác với các chức năng như thêm sản phẩm, thêm loại, thêm hãng, chỉnh sửa sản phẩm, xóa sản phẩm.
* Chức năng đăng nhập : sẽ có hai phần đăng nhập phía người dùng và phía quản trị.
* Chức năng hiển thị : hiển thị được giao diện các sản phẩm, và giao diện từng sản phẩm phía quản trị nhập vào.
* Giao diện sẽ được hoàn thiện tương thích vói các loại thiết bị khác nhau.
* Chức năng tìm kiếm cơ bản : nhập tên sản phẩm và hệ thống sẽ trả về các sản phẩm có tồn tại tên đã nhập.
* Chức năng giỏ hàng : khi nhấn vào thêm sản phẩm thì sản phẩm sẽ được cập nhật vào trong gio hàng của khách hàng, muốn thêm vào giỏ hàng thì khách hàng phải đăng nhập nếu chưa có tài khoản,
* Chức năng đặc hàng : khi nhấn vào đặc hàng thì sẽ được gửi mail để thông báo và xác nhận.
* Chức năng gửi mail tự độnh : khi nhấn vào đặc hàng mail sẽ được tự động gửi đén khách hàng để thông báo về chi tiết đơn hàng và xác nhận đặc hàng.
* Chức năng đánh giá sản phẩm : mỗi khách hàng sẽ được đánh giá và chấm điểm cho sản phẩm, phần điểm sẽ được tính trung bình và hiển thị tại phần giao diện các sản phẩm.
* Các chức năng không có trên web :
* Chức năng phân trang sản phẩm
* Chức năng nhập xuất hàng
* Chức năng tính toán hàng đã bán được và hàng còn tồn
* Chức năng thanh toán online
* Chức năng tích hợp chat realtime, và đăng nhập bằng nhiều nền tảng khác như google, facebook
* Nguồn thông tin và dữ liệu trang web
* Layout trang web được tìm trên trang css-free, layout đã được tinh chỉnh và bổ xung thêm khá nhiều để phù hợp vói trang web
* Phần Database thì được thành viên Bảo Duy làm hoàn toàn mới
* Phần thông tin tĩnh trên trang web cũng được thành viên Bảo Duy viết

## 1.2.Cơ sở lý thuyết

### Phần thiết kế layout :

* đây là kiến thức yêu cầu sẵn có và không nằm trong học phần này nên chỉ nêu yêu cầu và không nêu chi tiết
* Nắm vững kiến thức thiết kế web bao gồm css, html, javaScript
* Hiểu biết về framework bootstrap

### Phần dữ liệu database :

* đây là kiến thức yêu cầu sẵn có và không nằm trong học phần này nên chỉ nêu yêu cầu và không nêu chi tiết
* Nắm vững các kiến thức cơ bản về cơ sỡ dự liệu và kĩ năng sử dụng sql sever

### Phần lập trình chức năng cho website :

Phần này sẽ được nêu chi tiết

* ASP.NET là gì ?

ASP.NET – tình trạng hỗ trợ phát triển web, được nghiên cứu và phát hành bởi hãng Microsoft. Nền tảng ngày cho phép nhà lập trình tạo ra nhiều website động, ứng dụng dịch vụ hack.

* Cấu trúc cơ bản của ngôn ngữ ASP.NET

Lớp ngôn ngữ: Bao gồm nhiều ngôn ngữ lập trình cùng tồn tại trong mô hình .NET Framework. Chúng đóng vai trò hỗ trợ phát triển ứng dụng web.

Lớp thư viện: Trong .NET Framework tích hợp một lớp thư viện tiêu chuẩn. Thông thường thư viện Library sử dụng phổ biến nhất trong ứng dụng web .NET là Web Library. Trong Web Library tập hợp mọi thành phần cần thiết cho quá trình xây dựng ứng dụng web-based.

Lớp chạy ngôn ngữ lập trình: Common Language Infrastructure giữ vai trò như cơ sở hạ tầng ngôn ngữ, hỗ trợ chạy chương trình .NET. CLR có thể thực thi tác vụ chính như xử lý, thu gom.

* Ưu và nhược điểm của ASP.NET

Ưu :

Lập trình đơn giản với C#, VB.NET

Trang ASP.Net được biên dịch trước. Thay vì phải đọc và thông dịch mỗi khi trang web được yêu cầu, ASP.Net biên dịch những trang web động thành những tập tin DLL mà Server có thể thi hành nhanh chóng và hiệu quả.

ASP.Net hỗ trợ mạnh mẽ bộ thư viện phong phú và đa dạng của .Net Framework, làm việc với XML, Web Service, truy cập cơ sở dữ liệu qua ADO.Net,…

APX và ASP có thể cùng hoạt động trong 1 ứng dụng.

ASP.Net sử dụng phong cách lập trình mới: Code behide. Tách code riêng, giao diện riêng. Dễ đọc, dễ quản lý và bảo trì.

Kiến trúc lập trình giống ứng dụng trên Windows.

Hỗ trợ quản lý trạng thái của các control

Tự động phát sinh mã HTML cho các Server control tương ứng với từng loại Browser

Hỗ trợ nhiều cơ chế cache.

Không cần lock, không cần đăng ký DLL

Cho phép nhiều hình thức cấu hình ứng dụng

Hỗ trợ quản lý ứng dụng ở mức toàn cục

Global.asax có nhiều sự kiện hơn

Quản lý session trên nhiều Server, không cần Cookies

Nhược :

Không thể chạy trên Linux

Không có Visual Studio thì khó có thể viết code cho ASP.net

* MVC là gì?

MVC là gì? Chắc hẳn đây là câu hỏi thường gặp của những người đang tìm hiểu về ngành công nghệ thông tin. MVC là mô hình thiết kế trong ngôn ngữ lập trình Smalltalk-76 được tiến sĩ Trygve Reenskaug trình bày vào năm 1970 tại trung tâm nghiên cứu Xerox Palo Alto (PARC).

MVC là từ viết tắt được ghép từ 3 thành phần của mô hình là Model – View – Controller. Mỗi thành phần sẽ có một hoạt động riêng biệt và khi kết hợp sẽ tạo thành mô hình thiết kế hoàn chỉnh.

* Thành phần trong mô hình MVC

Model (M) Model chứa một cấu trúc dữ liệu có chức năng lưu trữ toàn bộ các thông tin dữ liệu của một ứng dụng. Trong mô hình MVC thì Model đóng vai trò kết nối cho 2 thành phần View và Controller. Đối với Model được thiết lập như một cơ sở dữ liệu hoặc đơn giản hóa như một file XML thông thường. Khi thiết lập thành phần model thì lập trình viên cần đảm bảo các thao tác với cơ sở dữ liệu như các hoạt động xem, truy xuất hoặc xử lý dữ liệu trong ứng dụng.

View (V) View là thành phần liên quan đến giao diện của ứng dụng khi người dùng trải nghiệm. Thông qua dữ liệu của MVC , người dùng sẽ thực hiện các thao tác tìm kiếm, sử dụng thông tin website, ứng dụng. Thành phần View được ứng dụng nhiều trong quá trình lập trình website và đây cũng là nơi mà các thành HTML tạo ra. Chức năng khác của thành phần View này chính là khả năng ghi nhận hành vi của người dùng để tương tác được với Controller. Mặc dù không có mối liên hệ trực tiếp với Controller nhưng View sẽ có nhiệm vụ hiển thị yêu cầu chuyển đến cho Controller xử lý thông tin. Dễ hình dung hơn thì bạn có thể tham khảo ví dụ khi người dùng nhấn vào nút “Back” hoặc “Trở về” là thành phần view thì người dùng đang tạo ra 1 hành động trên ứng dụng mà controller cần xử lý.

Controller (C) Controller là bộ phận sẽ xử lý các yêu cầu khi người dùng thao tác trên ứng dụng thông qua thành phần view. Lúc này, Controller sẽ thực hiện truy vấn và xuất dữ liệu phù hợp với yêu cầu của người dùng. Và để làm được điều đó controller còn có cần phải nối được với model để lấy dữ liệu.

* Ưu và nhược điểm mô hình MVC

Mô hình MVC phù hợp với các dự án với các đội ngũ làm việc độc lập với nhau, như vậy các công việc đều có thể được thực hiện một cách linh hoạt. Như vậy, mô hình này nổi bật nhờ các ưu điểm sau:

Hỗ trợ quá trình phát triển nhanh chóng: Với đặc điểm hoạt động độc lập của từng thành phần, các lập trình viên có thể làm việc đồng thời trên từng bộ phận khác nhau của mô hình này. MVC giúp bạn tiết kiệm rất nhiều thời gian.

Khả năng cung cấp đồng thời nhiều khung View: Với mô hình MVC, bạn có thể tạo ra đồng thời nhiều khung View cho Model.

Hỗ trợ các kỹ thuật không đồng bộ: MVC có thể hoạt động trên nền tảng JavaScript. Điều này có nghĩa là các ứng dụng MVC có thể hoạt động với các file PDF, các trình duyệt web cụ thể, và cả các widget máy tính.

Dễ dàng thao tác chỉnh sửa: Bộ phận Model hoạt động tách biệt với View đồng nghĩa với việc bạn có thể đưa ra các thay đổi, chỉnh sửa hoặc cập nhật dễ dàng ở từng bộ phận.

Giữ nguyên trạng thái data: Mô hình MVC truyền lại dữ liệu nhưng không định dạng lại dữ liệu. Do đó, các dữ liệu này có thể được dùng lại cho các thay đổi sau này.

Hỗ trợ các nền tảng phát triển SEO: Với mô hình MVC, bạn có thể dễ dàng tạo ra các mã SEO URL để thu hút lượng truy cập đối với ứng dụng bất kỳ.

Khó khăn trong quá trình điều hướng code: Điều hướng khung có thể phức tạp vì mô hình này bao gồm nhiều lớp và yêu cầu người dùng thích ứng với các tiêu chí phân tách của MVC.

Không thích hợp việc phát triển các ứng dụng nhỏ vì mô hình này yêu cầu bạn lưu trữ một số lượng lớn các file.

Nhiều khung hoạt động đồng thời: Việc phân tách một tính năng thành ba bộ phận khác nhau dễ dẫn đến hiện tượng phân tán. Do đó, đòi hỏi các nhà phát triển phải duy trì tính nhất quán của nhiều bộ phận cùng một lúc.

* Truyền và nhận tham số

Bạn có thể truyền dữ liệu từ controller lên view theo 3 cách :  
Cách 1 : Bạn dùng ViewBag(MVC3),ViewData(MVC2) để truyền dữ liệu từ controller lên view, ViewBag là 1 Object Dynamic chính vì vậy bạn có thể gán cho ViewBag = 1 Object,1 String, hay 1 Integer đều được.

Cách 2 : Có thể truyền dữ liệu từ controller lên view thông qua Các Action

public ActionResult Index() {

List<string> listColors= new List<string>(); listColors.Add("red"); listColors.Add("green"); listColors.Add("blue"); return View(listColors); }

Trên view bắt buộc bạn phải khai báo Code:

@model IEnumerable<string> : // gán dữ liệu từ controller vào model của view để view có thể sử dụng đựoc

Cách 3 : bạn có thể truyền và lấy dữ liệu từ view thông qua Ajax - Bạn muốn lấy dữ liệu từ view thì bạn có thể tham khảo FormCollection,Các HTTP POST,GET, AJAX..

Có thể truyền dữ liệu từ views lên controller thông qua

4 cách truyền bao gồm : Request, FormCollection**,** ActionArguments**,** Modelbạn có thể dụng bất kỳ cách nào cũng được, tùy vào dữ liệu nhiều ít và quen thì sử dụng, thường dùng 3 cách Request**,** ActionArguments**,** Model.

Cách 1 : Request dữ liệu

Ví dụ ta có đường dẫn:

h<ttp://localhost:58408/TruyenThamSoAction/ViDu1?ho=Ph%E1%BA%A1>m&ten=S%C6%A1n&congViec=IT

Cách lấy: có 3 cách

String value=Request[“ParameterName”];

String value=Request.QueryString[“ParameterName”];

String value=Request.Params[“ParameterName”];

VD:

String value=Request[“ho”]; // ho=Phạm

String value=Request.Params[“ten”]; // ten=Sơn

String value=Request.Params[“congviec”]; // congviec=ID

Lưu ý: 3 cách trên đều giống nhau, bạn chọn cách nào cũng được

Cách 2 : Get&Post trong Controller với Form Collection

VD: ta có form dữ liệu tại view như sau

<form action=”/Student/Register” method=”post”>

Id <input name=”Id” />

Id <input name=”Name” />

Id <input name=”Marks” />

<input type=”submit” value=”Register” />

</form>

Cách dùng Request.Form

Cấu trúc: String Value=Request.Form[“<tham số>”];

Public ActionResult Register()

{

String Id= Request.Form[“Id”];

String Name= Request.Form[“Name”];

double Marks= Convert.ToDouble(Request.Form[“Marks”]);

Return View();

}

Các dùng FormCollection

Public ActionResult Register(FormCollection Fields)

{

String Id= Fields[“Id”];

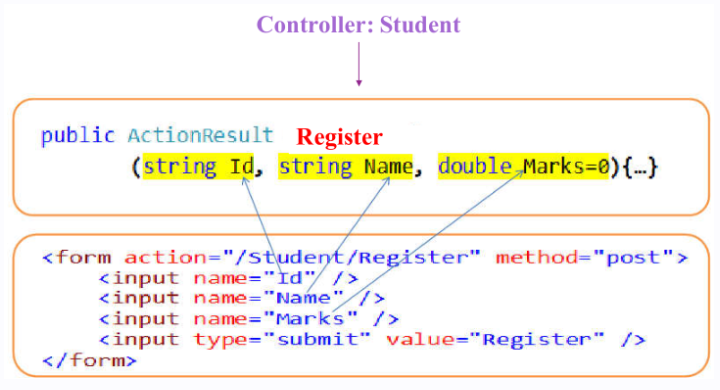
String Name= Fields[“Name”];

double Marks= Convert.ToDouble(Fields[“Marks”]);

Return View();

}

Cách 3 : Sử dụng Action Arguments



Hỉnh 1.1

Cách 4 : Sử dụng Model

VD: ta có Form

<form action=”/Student/Register” method=”post”>

Id <input name=”Id” />

Id <input name=”Name” />

Id <input name=”Marks” />

<input type=”submit” value=”Register” />

</form>

Cách lấy dữ liệu bằng Model:  
  
B1: tạo 1 Class trong Models để lấy dữ liệu truyền từ Form Action truyền đi, Click chuột phải vào folder Models chọn tạo mới 1 class.

Public string StudentInfo

{

Public string Id {get; set;}

Public string Name {get; set;}

Public double Marks {get; set;}

}

B2: tạo 1 ActionResult trong controller để lấy dữ liệu ra

Public ActionResult UserModel(StudentInfo model)

{

String Id= model.Id;

String Name= model.Name;

double Marks= Convert.ToDouble(model.Marks);

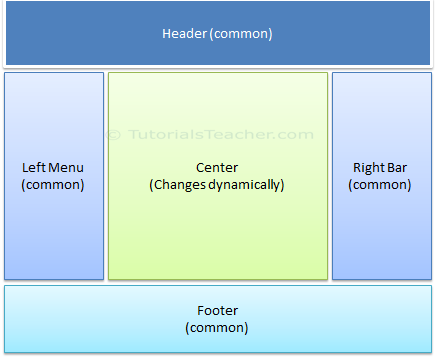
Return View();

}

* Layout trong ASP.NET MVC

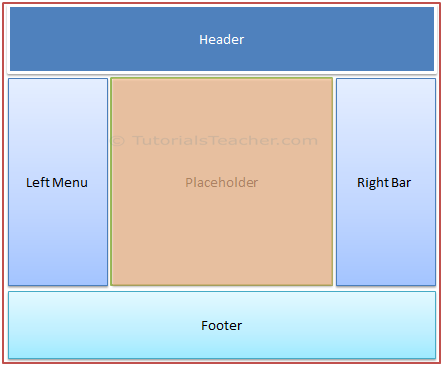
Một ứng dụng có thể chứa các phần chung trong giao diện người như logo, tiêu đề, menu trái , menu phải hoặc footer. ASP.NET MVC đã giới thiệu một giao diện Bố cục chứa các phần UI chung này, do đó chúng ta không phải viết cùng một mã trong mỗi trang. Giao diện bố cục giống như trang chính của ứng dụng webform ASP.NET.

Ví dụ: Giao diện người dùng ứng dụng có thể chứa Header, Thanh menu bên trái, thanh menu bên phải và phần footer vẫn giữ nguyên trong mỗi trang và chỉ phần Center thay đổi.



Hình 1.2

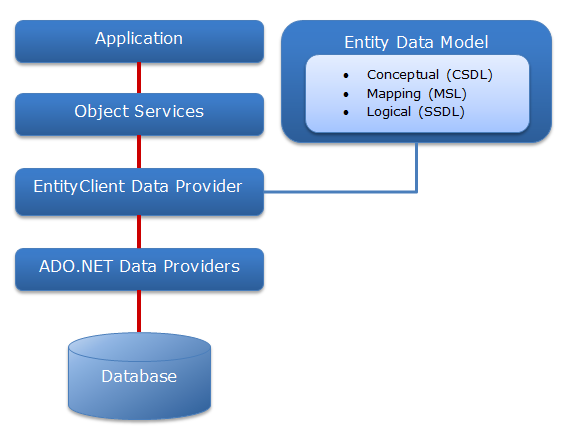
Layout cho phép bạn xác định một mẫu trang web chung, có thể được kế thừa trong nhiều view để cung cấp giao diện nhất quán  nhiều trang trong một ứng dụng. Layout bố trí giúp loại bỏ mã hóa trùng lặp và tăng cường tốc độ phát triển và bảo trì dễ dàng. Giao diện bố trí cho giao diện người dùng ở trên sẽ chứa phần Header, Menu bên trái, Thanh bên Right và Phần Footer. Nó chứa một placeholdernằm ở Center để dễ dàng thay đổi nội dung.



Hình 1.3

* Entity framework

Kiến trúc của Entity Framework được minh họa như sau:



Hình 1.4

**Object Services**

Đây là các class tự động sinh ra tương ứng với mô hình dữ liệu. Các class này bao gồm:

ObjectContext đại diện cho một database. ObjectContext có chức năng quản lý các kết nối, định nghĩa mô hình dữ liệu với metadata và thao tác với database. Lớp này cũng có thể thêm vào các phương thức đại diện cho các stored procedure trong database.

ObjectSet<TEntity> là một  một tập hợp các entity. Mỗi đối tượng này tương ứng với một table. Có thể lấy được các đối tượng này thông qua các property tương ứng của ObjectContext.

EntityObject, ComplexObject là các lớp tương ứng cho một dòng dữ liệu của table trong database. Khác biệt chính giữa hai loại này là ComplexObject không chứa primary key.

EntityCollection<TEntity> và EntityReference<TEntity>: là các đối tượng thể hiện mối quan hệ (relationship) giữa hai entity class. Mỗi đối tượng này có thể được truy xuất thông qua các property của entity class.

Ta có bảng ánh xạ tương đương sau giữa các đối tượng trong database và EF:

|  |  |
| --- | --- |
| Database Object | Entity Framework Object |
| Database | ObjectContext |
| Table, View | EntityObject, ComplexObject |
| Column | Property |
| Relationship | EntityCollection<TEntity>, EntityReference<TEntity> |

Hình 1.5

Ví dụ sau cho thấy cách tạo một đối tượng ObjectContext từ mô hình dữ liệu NorthwindEntities (tên của connection string được lưu trong app.config), sau đó sử dụng LINQ to Entities để lấy ra các sản phẩm (trong bảng Products) có tên bắt đầu bằng “G” và in ra màn:

ObjectContext context = new ObjectContext("name=NorthwindEntities");

ObjectSet<Product> products = context.CreateObjectSet<Product>();

var query = from p in products

where p.ProductName.StartsWith("g")

select p;

foreach (var item in query)

{

Console.WriteLine("{0,-4}{1}",item.ProductID, item.ProductName);

}

Output:

6   Grandma's Boysenberry Spread

15  Genen Shouyu

22  Gustaf's Knäckebröd

24  Guaraná Fantástica

26  Gumbär Gummibärchen

31  Gorgonzola Telino

33  Geitost

37  Gravad lax

44  Gula Malacca

56  Gnocchi di nonna Alice

69  Gudbrandsdalsost

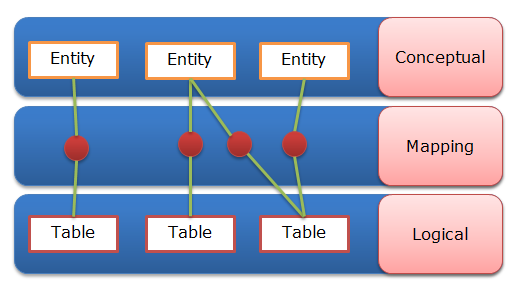
**Entity Data Model**

Entity Data Model (EDM) là mô hình dữ liệu được mô tả thông qua các ngôn ngữ theo chuẩn XML. EDM được chia làm 3 lớp là: Conceptual, Mapping và Logical. Mỗi lớp này được định nghĩa bởi ngôn ngữ riêng theo định dạng XML:

Conceptual – Conceptual Schema Definition Language (CSDL): là ngôn ngữ định nghĩa các entity, relationship, hàm trong tập tin với phần mở rộng .csdl. Có thể tạo được các entity class (object layer).

Mapping – Mapping specification language (MSL): định nghĩa các ánh xạ giữa lớp conceptual và logical, nội dung này được lưu trong tập tin .msl.

Logical – Store Schema Definition Language (SSDL): định nghĩa mô hình lưu trữ của dữ liệu, lưu trữ trong tập tin .ssdl.



Hình 1.6

Các nội dung của ba tập tin .csdl, .msl và .ssdl được lưu trữ trong cùng tập tin .edmx trong Visual Studio. Tập tin này có thể được tạo ra tự động từ database và công cụ Entity Framework Model Wizard của Visual Studio. Trong quá trình biên dịch, các tập tin .csdl, .msl và .ssdl sẽ được tạo ra dựa vào tập tin .edmx này.

Note: Khi mở tập tin .edmx này, VS tự động hiển thị giao diện trực quan của nó bằng công cụ mặc định là ADO.NET Entity Data Model Designer. Để xem nội dung của tập tin này, bạn nhấn chuột phải, nhấn Open with… và chọn mục XML Editor hoặc một trình xem dưới dạng văn bản bất kì.

**EntityClient Data Provider**

EntityClient là một data provider mới của ADO.NET dùng để truy xuất đến database. Được xây dựng bên trên các ADO.NET data provider cơ bản, EntityClient không truy xuất trực tiếp dữ liệu mà thông qua các data provider khác dựa vào các thông tin dữ liệu từ Entity Data Model.

EntityClient cũng bao gồm các lớp giống như các ADO.NET data provider khác và tên lớp được đặt với tiền tố Entity. Ví dụ bạn có thể tạo kết nối bằng EntityConnection, tạo các câu truy vấn bằng EntityCommand và đọc kết quả bằng EntityDataReader.

Một điểm khác biệt với các data provider khác là EntityClient sử dụng Entity SQL để truy vấn dữ liệu. Các lệnh Entity SQL sẽ được chuyển thành một cấu trúc lệnh dạng cây (command tree) và chuyển xuống cho các data provider khác.

Ví dụ sau sử dụng EntityClient để tạo kết nối, tạo đối tượng EntityCommand và sử dụng tham số để lấy các sản phẩm có CategoryID là 1:

int categoryID = 1;

using (EntityConnection con = new EntityConnection("Name=NorthwindEntities"))

{

con.Open();

EntityCommand cmd = con.CreateCommand();

cmd.CommandText = "SELECT VALUE p FROM NorthwindEntities.Products AS p WHERE p.CategoryID = @catId";

cmd.Parameters.AddWithValue("catId", categoryID);

EntityDataReader reader = cmd.ExecuteReader(CommandBehavior.SequentialAccess);

while (reader.Read())

{

var name = reader["ProductName"];

var catId=reader["CategoryID"];

Console.WriteLine("{0,-3}{1}",catId,name);

}

reader.Close();

}

Output:

1 CategoryID ProductName

1 Chai

1 Chang

1 Guaraná Fantástica

1 Sasquatch Ale

1 Steeleye Stout

1 Côte de Blaye

1 hartreuse verte

1 Ipoh Coffee

1 Laughing Lumberjack Lager

1 Outback Lager

1 Rhönbräu Klosterbier

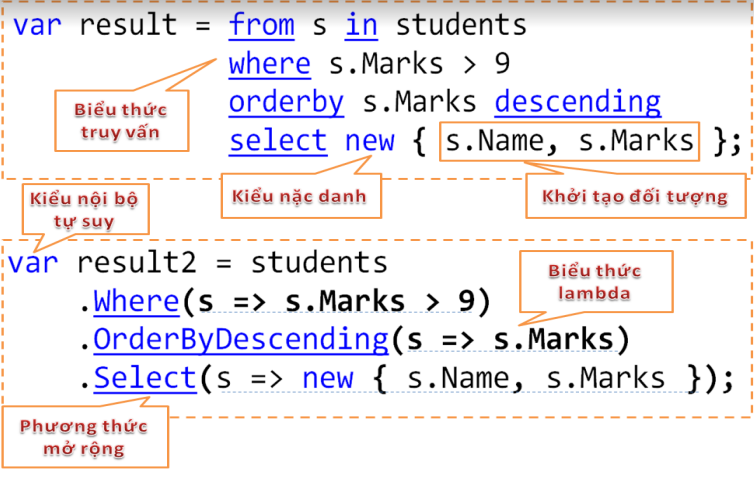
1 Lakkalikööri

* LinQ

LinQ có tên gọi là Language Integrated Query, một thư viện cung cấp cho ngôn ngữ lập trình C# và Visual Basic. Thư viện này cung cấp khả năng truy vấn database ngay trên ngôn ngữ lập trình. Khi sử dụng, LinQ sẽ mô phỏng lại cơ sở dữ liệu và lưu trong trương trình, sau đó mỗi entity sẽ được xây dựng thành mỗi class như mô hình MVC, từ đó có thể truy xuất database thông qua những class đó.

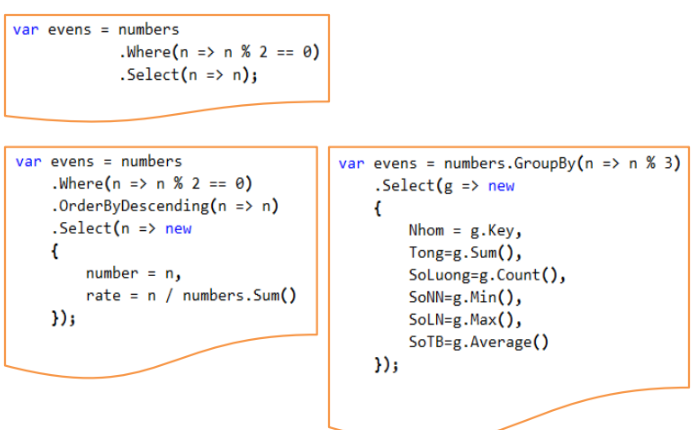
Có 2 cách truy vấn chính :

Kiểu ……..



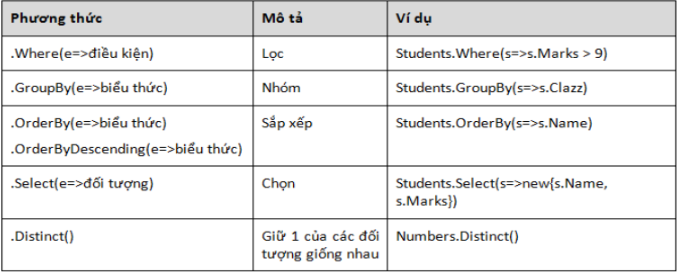
Hình 1.7

Và kiểu



Hình 1.8

Một vài truy vấn cơ bản



Hình 1.9

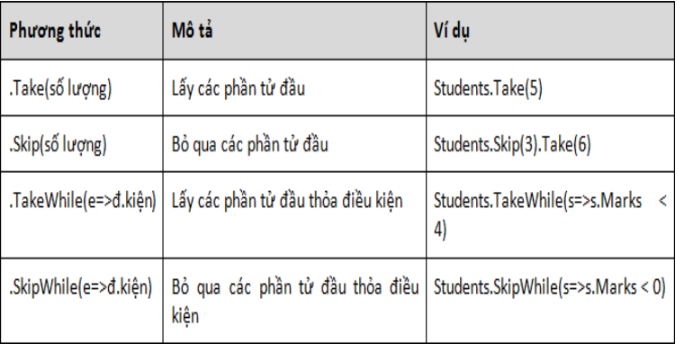
Vd code:

var studs = Students

.Where(s=>s.Marks > 9)

.OrderBy(s=>s.Marks)

.Select(s=>s);



Hình 1.10

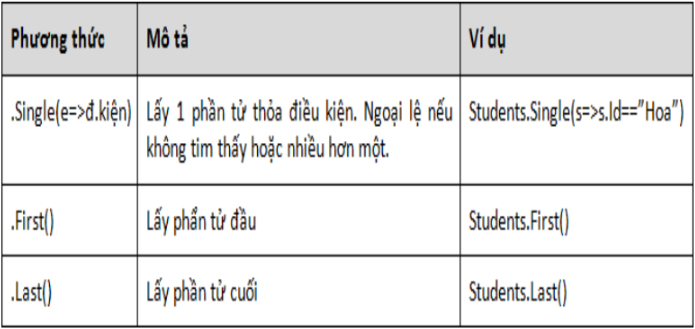
Vd code:

var result = db.Products

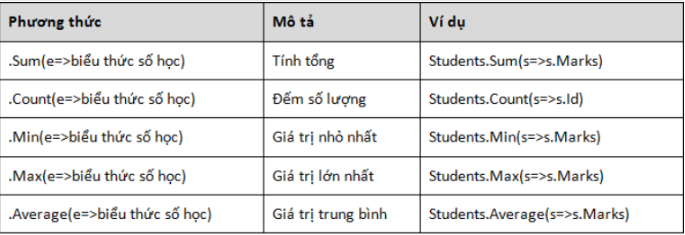
.Skip(10).Take(20)

var result = db.Customers

.Single(c=>c.Id==“A” && c.Password=“B”)



Hình 1.11



Hình 1.12

Var result = db.Products

GroupBy(p=>p.Category)

Select(g=>new{g.Key.Name, g.Count})

# **Chương 2 : Phân tích và hoạch định**

## 2.1.Sơ đồ chức năng

Đăng nhập

Quản trị

Người dùng

Hiển thị sản phẩm

Thêm sản phẩm

Thêm sản phẩm vào giỏ hàng

Sửa sản phẩm

Đánh giá, chấm điểm

Gửi mail tự động

Đặc hàng

Xóa sản phẩm

Thêm loại

Thêm hãng

Xóa loại

Xóa hãng

Sữa hãng

Sữa loại

* Mô tả :
* Chức năng đăng nhập : Có nhiệm vụ là chia người truy cập trang web ra hai loại một là người dùng có nhu cầu mua xem và đặc hàng sản phẩm hai là người quản trị truy cập vào để quản lý, thêm, xóa sản phẩm.
* Chức năng thêm, xóa, sửa sản phẩm, hãng, loại : dùng để quản trị viên thêm xóa sửa sản phẩm, loại sản phẩm, hãng sản phẩm trên trang web.
* Chức năng hiển thị sản phẩm : Sản phẩm đưowcj hiển thị đầy đủ tại giao diện.
* Chức năng đánh giá, chấm điểm sản phẩm: mỗi khách hàng có thể đánh giá mỗi sản phẩm một lần.
* Chức năng thêm sản phẩm vào giỏ hàng: người dùng muốn mua hàng có thể thêm sản phẩm vào giỏ hàng và tiến hành đặc hàng sau khi tinh chỉnh số lượng.
* Chức năng đặc hàng: sau khi tinh chỉnh tại giỏ hàng người dùng có thể tiến hành đặc hàng.
* Chức năng gửi mail tự động: sau khi nhấn đặc hàng người dùng sẽ được gửi mail để thông báo và trao đổi chi tiết cũng nhue xác nhận đơn hàng.

## 2.2.Sơ đồ đường dẫn trang web

Danh sách loại

Xóa loại

Thêm loại

Sữa loại

Thêm hãng

Thêm sản phẩm

Sửa sản phẩm

Xóa hãng

Quản lý hãng

Quản lý loại

Quản lý xe

Sữa hãng

Xóa sản phẩm

Danh sách hãng

Danh sách sản phẩm

Trang chủ

Đăng nhập

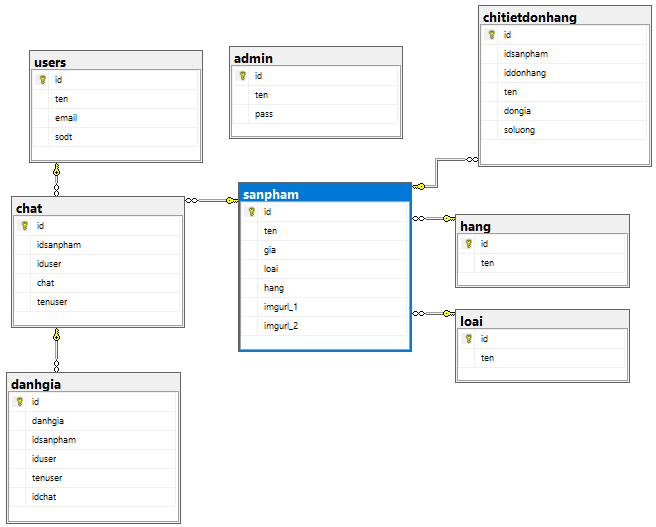
Hiển thị sản phẩm

Chi tiết sản phẩm

Giỏ hàng

* Mô tả :
* Trang chủ : là giao diện chính để kết nối với tất cả những trang khác, là trang giới thiệu về chức năng, dịch vụ chủ yếu của trang web .
* Trang đăng nhập: đây là trang dùng để chia quyền cũng như phân loại người dùng trang web.
* Trang hiển thị sản phẩm: trang hiển thị và giới thiệu được các sản phẩm của trang web.
* Trang giỏ hàng: hiển thị được các sản phẩm mà người dùng đã thêm vào giỏ hàng.
* Trang chi tiết sản phẩm: trang giới thiệu chi tiết về một sản phẩm nhất định.
* Trang quản lý xe, loại, hãng: trang để quản lý các sản phẩm của quản trị viên có các chức năng như thêm xóa sửa sản phẩm.

## 2.3.Sơ đồ Database



Hình 2.1

* Mô tả :
* Bảng admin : chứa các dữ liệu về quản trị viên
* Bảng user : chứa các thông tin về người dùng liên kết với bảng chat để quản lý đánh giá của người dùng
* Bảng chat : chứa đánh giá của người dùng và có khóa ngoại là id của bảng user để biết được thông tin của người đánh giá, liên kết với bảng danhgia để quản lý việc chấm điểm của người dùng, liên kết với bảng bảng sản phẩm để biết được đánh giá cho sản phẩm nào.
* Bảng danhgia : quản lý việc chấm diểm của người dùng liên kết với bảng chat để có được thông tin của người chấm điểm.
* Bảng sản phẩm : chứa tất cả các thông tin của sản phẩm và cả link của hình ảnh sản phẩm, liên kết với các bảng hãng và loại để biết được hãng và loại của sản phẩm, liên kết với bảng chitietdonhang để quản lý trong việc thêm sản phẩm vào giỏ hàng.

# **Chương 3 : Triển khai ứng dụng web**

## 3.1.Thiết kế giao diện : thành viên nhóm trình bài

## 3.2.Thiết kế Database : thành viên nhóm trình bài

## 3.3.Thiết kế xử lý:

### Web thiết kế dựa trên mô hình mvc

* Model bao gồm : các class admin,chat,chitietdonhang,Database,hang,loai,phan quyen,quyen,sanpham,user liên kết với các bảng cùng tên trong database để lấy dữ liệu, và còn các class như kiemtradangnhap, mail, chuyendoi, kiemtramatkhau, kiemtrasesion, maping\_admin, maping\_hang, maping\_loai, mapsanpham
* Controller bao gồm : sáu controller để xử lý các chức năng trên web ở hướng người dùng lần lược là chitietController, DangnhapController, DanhsachxeController, GiohangController, MailController, TrangchuController và năm controller xử lý hướng quản trị viên DangnhapadminController, QuanlyadminController, QuanlyhangController, QuanlyloaiController, QuanlyxeController.
* View bao gồm : tám view để hiển thị giao diện phía người dùng lần lược là chitiet, dangnhap, danhsachxe, timkiem, giohang, layout, mail, trangchu và 15 view phía quản trị viên dangnhap, layout, them,xoa,sua của admin, xe, hang, loai và view thông báo quyền.

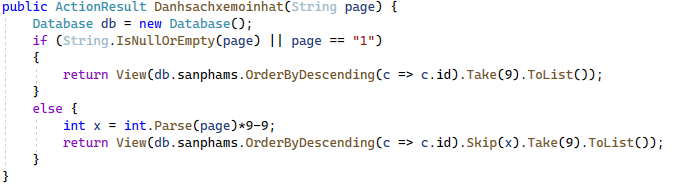
### Chi tiết các chức năng của web

##### Hiển thị thông tin và tìm kiếm :

Chức năng là hiện thị dược đầy đủ thông tin của sản phẩm, thông tin giỏ hảng ở phía người dùng, và thông tin sản phẩm, hảng, loại, admin phía quản trị viên, đồng thời phải hiển thị được những thông tin mà người dùng và quản trị viên tìm kiếm.

Phía người dùng có controller danhsachxe xử lý phần hiển thị thông tin, và 3 action để xử lý là Danhsachxemoinhat, timkiem, Danhsachxeloai

Controller Danhsachxemoinhat :



Hình 3.1

Controller này giữ nhiệm vụ hiển thị được các sản phẩm mới nhất, và phân trang sau cho một trang có tối đa 9 sản phẩm.

Db là biến tham chiếu đến class database trong model là class chủ kết nối tới database để xử lý dử liệu.

Biến page là biến nhận vào từ view để hiển thị trang mà người dùng muốn xem, câu lệnh if kiểm tra biến page có null hoặc bằng q thì trả ra trang san phẩm đầu tiên ngược lại thì trả ra những trang theo thao tác phía người dùng. Tại dòng return ta sẽ sử dụng câu lệnh của linq để sắp xếp những sản phẩm được quản trị viên thêm vào gần nhất lên đầu và Tack để lấy ra 9 phần tử cho mỗi trang còn skip có chức năng bỏ qua những trang phía trước biến x dùng để tính toán ra số phần tử phải bỏ.

Xử lý bên view :

View sẽ nhận vào danh sách các sản phẩm đã được controller truyền qua và hiển thị lên giao diện

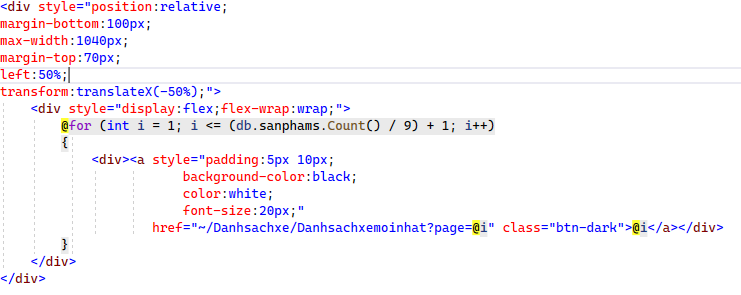


Hình 3.2



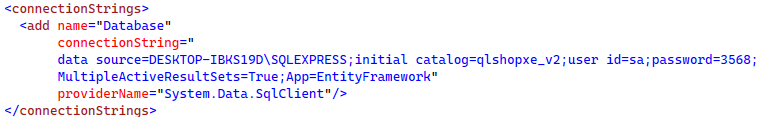
Hình 3.3

Xử lý phân trang :



Hình 3.4

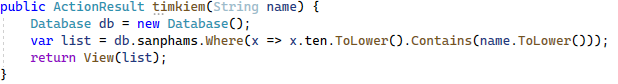
Có chức năng in ra các nút dựa trên tổng số sản phẩn chia cho 9, vì mõi trang có 9 sản phẩm, tag a có link dẫn đến action và truyền qua biến page để xửa lý



Hình 3.5

Controller timkiem :

Giữ nhiệm vụ khi người dùng nhập vào tên sản phẩm ở khung tìm kiếm thì phải đưa ra những sản phẩm có tên giống với khung tìm kiếm.

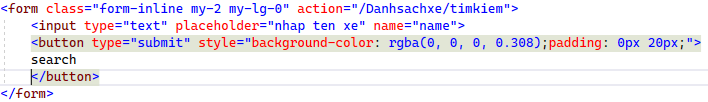


Hình 3.6

Controller nhận vào một biến name ở khung nhập và sửa dụng cú pháp where của linq tìm ra những sản phẩm trong tên có chưa kí tự mà người dùng nhập vào tức là biến name. hàm toflower để chuyển các kí tự thành chữ thường.

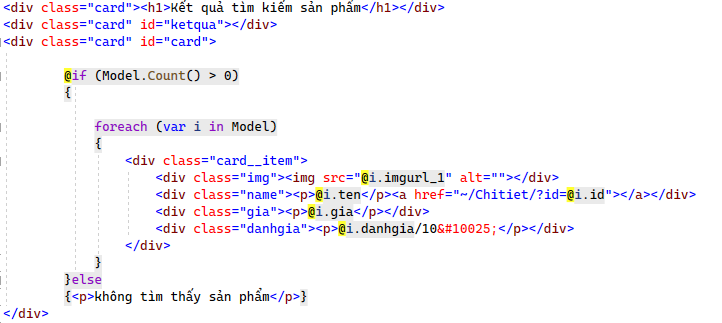
Xử lý bên view :

Khung tìm kiếm :



Hình 3.7

Khung hiển thị danh sách sản phẩm tìm kiếm:

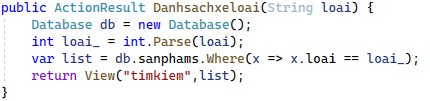


Hình 3.8

Controller Danhsachxeloai :

Giữ nhiệm vụ khi người dùng lựa chọn loại xe thì giao diện sẽ hiển thị những sản phẩm có đúng loại đấy ra trang web.

Phần xử lý phía controller :



Hình 3.9

Controller nhận vào một biến loai ở khung nhập và sửa dụng cú pháp where của linq tìm ra những sản phẩm có loại giống với phần loại mà người dùng lựa chọn tức là biến name. hàm toflower để chuyển các kí tự thành chữ thường.

Xử lý bên view :

Khung loại ở view :

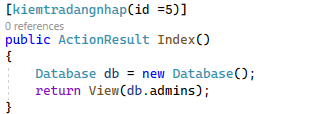


Hình 3.10

Khung hiển thị của action này được dùng chung với action tìm kiếm.

Phía quản trị viên có Controller Quanlyadmin, Controller Quanlyxe, Controller Quanlyloai, Controller Quanlyhang, và mỗi controller có một action là index xửa lý hiển thị dữ liệu, ngoài ra controller Quanlyxe có khung tìm kiếm để tìm kiếm sản phẩm.

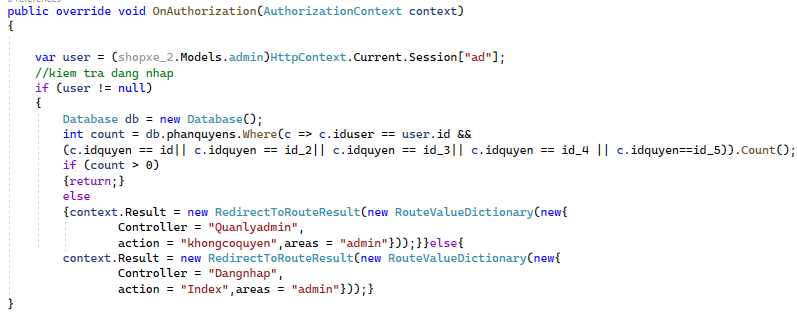
Controller Quanlyadmin :



Hình 3.11

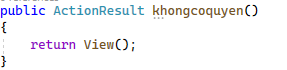
Action chỉ đơn giản trả ra list các admin trong bảng admin ở database.

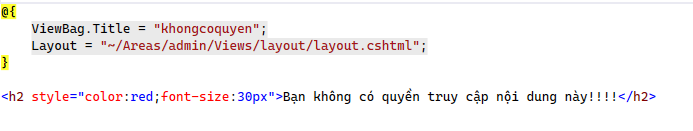
[Kiemtradangnhap(id = 5)] là một class được overide lại và được sử dụng để kiểm tra đăng nhập, đồng thời kiểm tra quyền của quản trị viên.



Hình 3.12

Class giữ nhiệm vụ là kiểm tra đăng nhập (user!=null) nếu phát hiện chưa đăng nhập thì chuyển về trang đăng nhập. Nếu đã được đăng nhập thì tiếp tục kiểm tra quyền trong bảng phân quyền ở database quyền sẽ c truyền từ ngoài vào tức là id, ví dụ như ở action này quyền là 5([Kiemtradangnhap(id = 5)]), và 5 là quyền cao nhất(quyền admin), trường hợp sai quyền thì sẽ đưa về trang thông báo không có quyền truy cập.

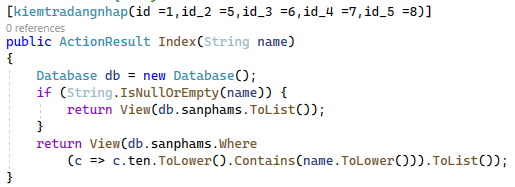




Hình 3.13

Controller Quanlyxe :

Có chức năng hiển thị danh sách xe và tìm kiếm xe

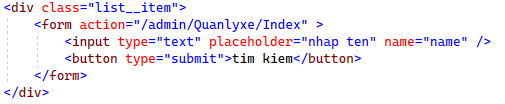


Hình 3.14

Đầu tiên action nhận vào một biến name và kiểm tra name đó nếu như name null hoặc rỗng thì action sẽ truyền một danh sách gồm tất cả các sản phẩm sang view còn nếu ngược lại thì sẽ truyền một danh sách gồm tất cả các sản phẩm nào trong tên có tồn tại biến name sang view.

Xử lý bên view:

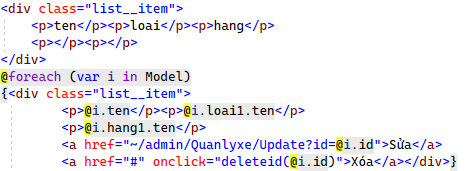
Khung tìm kiếm :



Hình 3.15

Khung có nhiệm vụ truyền biến name sang cho action xử lý

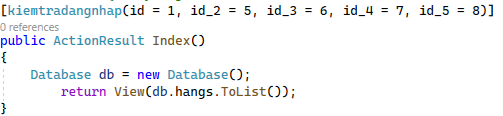
Khung hiển thị thông tin :



Hình 3.16

Có tác dụng hiện thị ra các thông tin nhận từ action ra dưới dạng table và 2 tag a để xửa lý phần sửa, xóa dữ liệu.

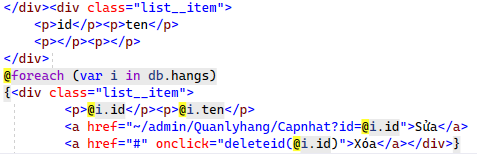
Controller Quanlyloai - hang :



Hình 3.17

Phần thông tin hiển thị của hảng và loại sẽ như nhau, action sẽ truyền qua view tất cả các thông tin của hảng và loại.

Xử lý bên view :



Hình 3.18

Như phần view của admin view của hảng và loại cũng sữ hiển thị các thông tin nhận được từ action dưới dạng table.

##### Thêm sửa xóa thông tin

Đây là chức năng dành cho quản trị viên,bao gồm Controller Quanlyadmin, Controller Quanlyxe, Controller Quanlyloai, Controller Quanlyhang và mỗi controller có 3 action là them, sua, xoa để thực hiện chức năng xửa lý dữ liệu.

Các hàm cần dùng trong chức năng này : saveChanges() dùng để lưu lại dữ liệu xuống database, Server.MapPath("/imgxe/") có chức năng ánh xạ tới thư mục img để lưu ảnh, hàm chuyendoi dùng để chuyển chữ in hoa thành chữ thường, Add dùng để thêm 1 dữ liệu vào danh sách nào đó, saveAs() dùng để lưu tệp vào trong thư mục.

Chức năng thêm, sửa, xóa sản phẩm(xe) :

Thêm :



Hình 3.19

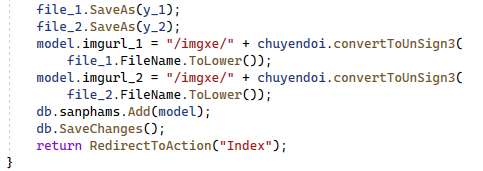
Action này có nhiệm vụ hiển thị ra view dùng để điền thông tin của sản phẩm, và dưới đây là chi tiết view



Hình 3.20

Form có chức năng nhận dữ liệu mà quản trị nhập vào sau đó truyền qua action them để tiến hành xử lý và dòng enctype="multipart/form-data" trên form để có thể truyền ảnh sang phía action, 2 vòng lặp foreach để hiển thị ra các hãng, loại của sản phẩm để lựa chọn.

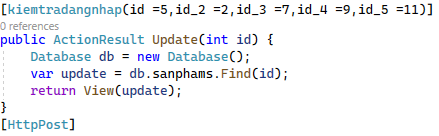




Hình 3.21

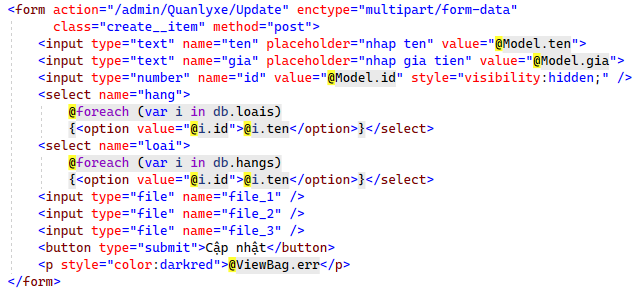
Đầu tiên action này nhận vào một biến model có kiểu dữ liệu là class sanpham, và hai file có kiểu dữ liệu là hình ảnh, sau đó lần lược kiểm tra với 3 if, kiểm tra xong thì tạo biến x để ánh xạ đến thư mục imgxe, lần lược y\_1 và y\_2 là đường dẫn và tên của ảnh sau khi chuyển thành chữ thường, ảnh được lưu lại tại thư mục imgxe và sau đó sẽ lưu lại đường đãn đó trong cột imgurl\_1, imgurl\_2 của biến model tiếp đến là add phần modle vào trong class sanpham và lưu về database.

Sửa :



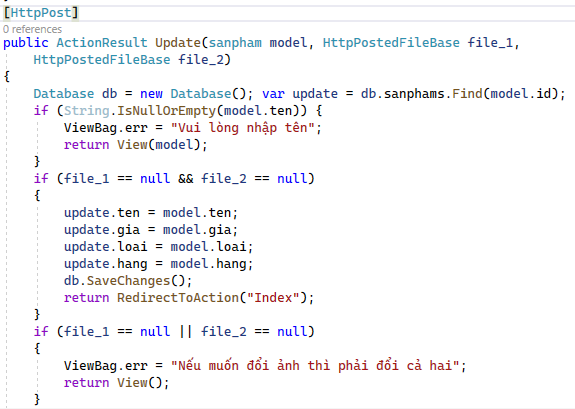
Hình 3.22

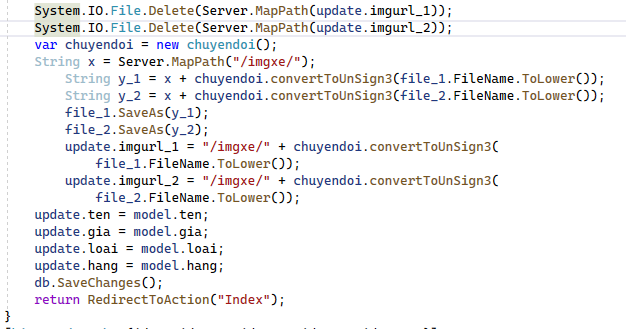
Tại action này sẽ nhận vào một biến id và action sẽ truyền qua view một sản phẩm có id bằng với biến id nhận vào dưới cú pháp của linq find.



Hình 3.23

Form này sẽ hiện thông tin của sản phần cần sữa và sẽ truyền thông tin sau khi sửa qua controller, 2 vòng lặp foreach để hiển thị ra các hãng, loại của sản phẩm để lựa chọn

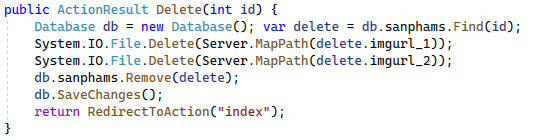




Hình 3.24

Về cơ bảng thì phần update sẽ giống hệt phần thêm chỉ khác ở chổ ta phải tìm ra sản phẩm và cập nhật lại nó thay vì thêm một sản phẩm mới. Đầu tiên thì action update sẽ có một số ràng buộc như nếu không chọn ảnh nào thì ảnh sẽ giữ nguyên, không được bỏ trống dữ liệu và nếu đổi ảnh sẽ phải đổi cả hai ảnh và tất cả các ràng buộc kể trên đều được code về dạng if và else. Trước tiên sẽ có biến update được tạo nên dựa vào cú pháp linq find, biến này chính là sản phẩm cần sửa, sau đó qua một vài ràng buột tiến hành update biến này và lưu lại vào database.

Xóa

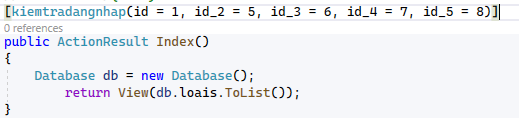


Hình 3.25

Action xóa này sẽ không có view, action sẽ nhận vào một id, tạo biến delete dưới cú pháp linq tìm ra sản phẩm cần xóa dựa trên id và gán cho delete, tiếp đó dùng hàm remove xóa đi sản phẩm đó và cập nhật lại database.

Chức năng thêm, sửa, xóa loại :

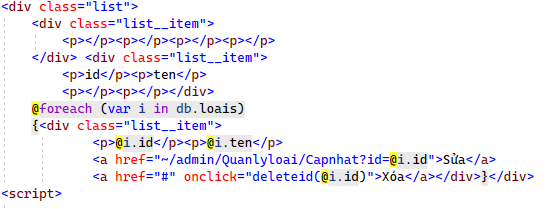
Thêm :



Hình 3.26

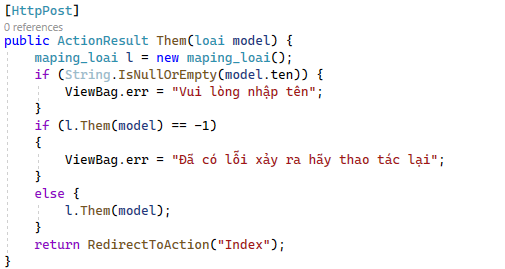
Action trả về view danh sách các loại của sản phẩm

Xử lý bên view :



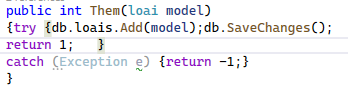
Hình 3.27

Phần view có nhiệm vụ hiển thị các thông tin nhận được từ action dưới dạng table.



Hình 3.28

Action này nhận vào một biến model có kiểu dữ liệu là loai sau đó là hai if để kiểm tra rỗng và bắt lỗi khi savechange() bị lỗi, ở đây có sử dụng hàm maping\_loai().them.



Hình 3.29

Hàm này sẽ nhận vào một biến model có kiểu dữ liệu là loai và sẽ sử dụng try cath để bắt lỗi khi lưu dữ liệu xuống database.

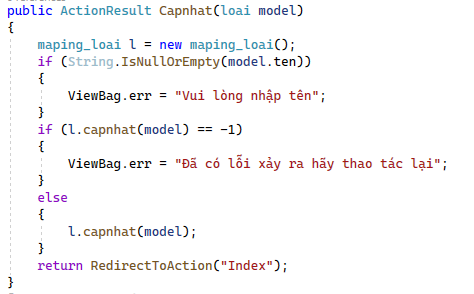
Sửa :



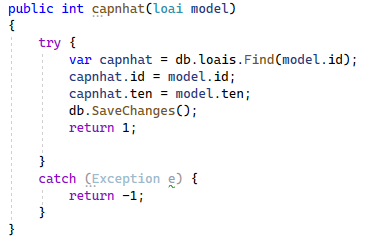


Hình 3.30

Hàm sẽ nhận vào một id sau đó gọi hàm maping\_loai.chitiet(id) để tìm và xuất ra loại cần sửa, hàm chitiet sửa dụng cú pháp linq và trả về một loại có id bằng với id truyền vào.



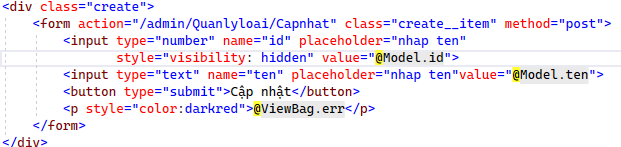
Hình 3.31



Hình 3.32

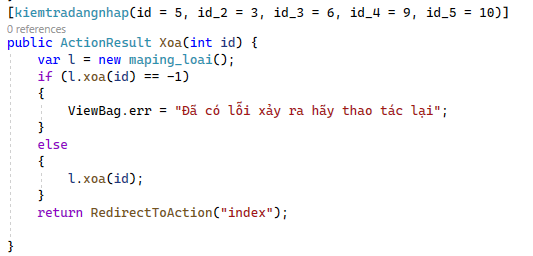
Action này nhận vào một biến model có kiểu dữ liệu là loai 2 if kiểm tra rổng và lỗi khi lưu xuống database bằng hàm maping\_loai().capnhat, sau khi kiểm tra thì dùng hàm capnhat để cập nhật lại loại cần cập nhật, hàm capnhat sẽ nhận vào một biến kiểu loai và dùng cú pháp lin1 tìm ra loại cần sửa và tiến hành sửa.

Xử lý bên view:

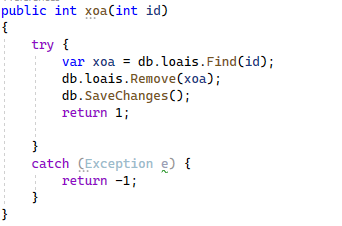


Phần value sẽ được truyền từ action qua để hiển thị loại cần sửa.

Xóa :



Hình 3.33

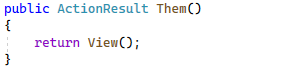


Hình 3.34

Action này sẽ nhận vào một id sau đó gọi hàm xoa trong maping\_loai để tiến hành xóa loại, hàm xóa sẽ nhận vào một id dùng cú pháp linq tìm và xoó loại có id dó, và action này sẽ không có view.

Chức năng thêm, sửa, xóa loại :

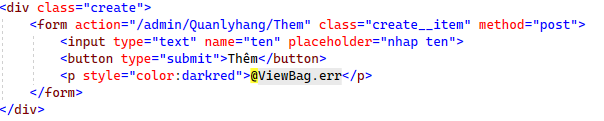
Thêm :



Hình 3.35

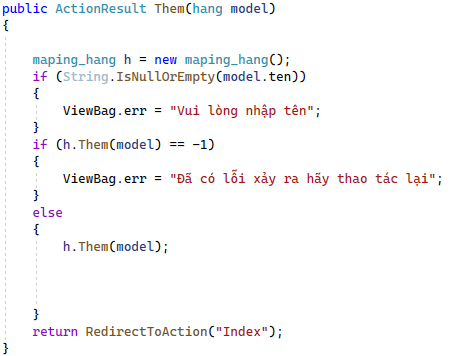
Action trả về view danh sách các loại của sản phẩm

Xử lý bên view :



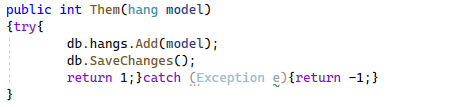
Hình 3.36

Phần view có nhiệm vụ hiển thị các thông tin nhận được từ action dưới dạng table.



Hình 3.37

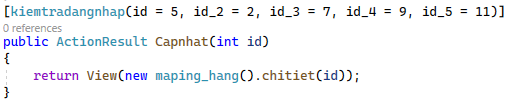
Action này nhận vào một biến model có kiểu dữ liệu là hang sau đó là 2 if để kiểm tra rỗng và bắt lỗi khi savechange() bị lỗi, ở đây có sử dụng hàm maping\_hang().them.

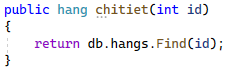


Hình 3.38

Hàm này sẽ nhận vào một biến model có kiểu dữ liệu là hang và sẽ sử dụng try cath để bắt lỗi khi lưu dữ liệu xuống database.

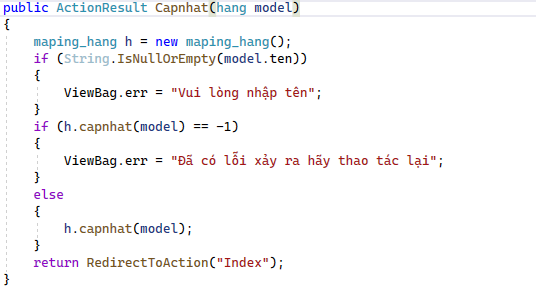
Sửa :



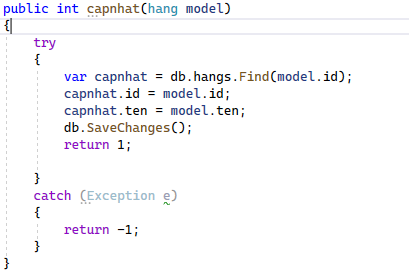


Hình 3.39

Action sẽ nhận vào một id sau đó gọi hàm maping\_hang.chitiet(id) để tìm và xuất ra hãng cần sửa, hàm chitiet sửa dụng cú pháp linq và trả về một hãng có id bằng với id truyền vào.



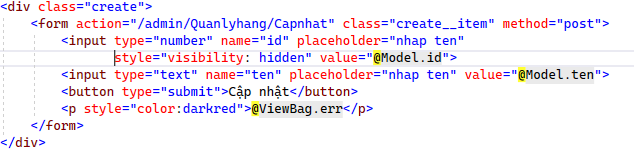
Hình 3.40



Hình 3.41

Action này nhận vào một biến model có kiểu dữ liệu là hang 2 if kiểm tra rổng và lỗi khi lưu xuống database bằng hàm maping\_hang().capnhat, sau khi kiểm tra thì dùng hàm capnhat để cập nhật lại hãng cần cập nhật, hàm capnhat sẽ nhận vào một biến kiểu hang và dùng cú pháp linq tìm ra hãng cần sửa và tiến hành sửa.

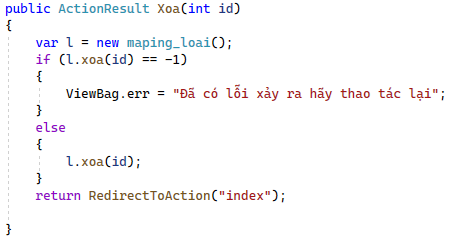
Xử lý bên view:



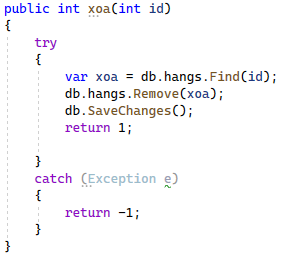
Hình 3.42

Phần value sẽ được truyền từ action qua để hiển thị hãng cần sửa.

Xóa :



Hình 3.43

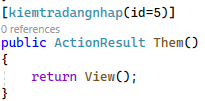


Hình 3.44

Action này sẽ nhận vào một id sau đó gọi hàm xoa trong maping\_hang để tiến hành xóa hãng, hàm xóa sẽ nhận vào một id dùng cú pháp linq tìm và xóa loại có id dó, và action này sẽ không có view.

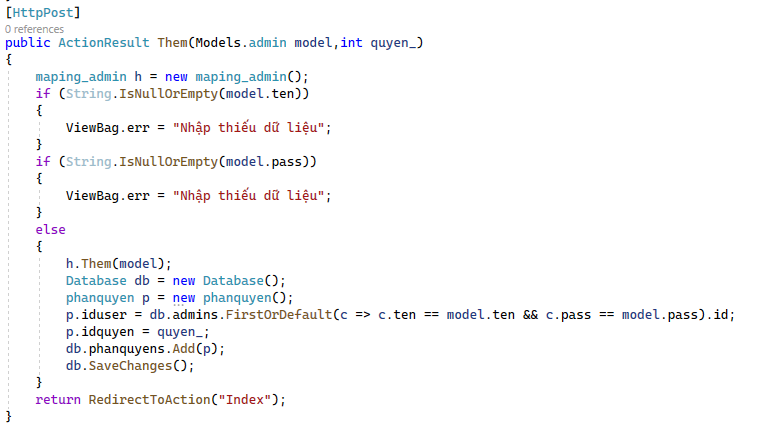
Chức năng thêm, sửa, xóa admin:

Thêm :

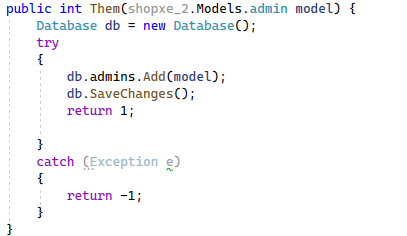


Hình 3.45

Action trả về view danh sách các admin



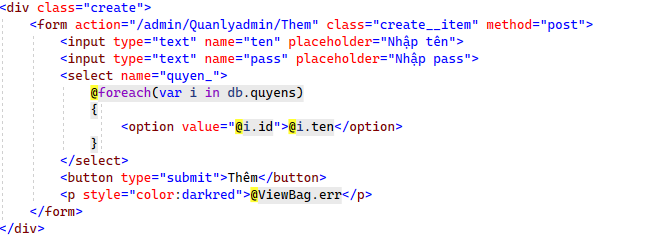
Hình 3.47



Hình 3.48

Action này sẽ nhận vào một model kiểu admin và một quyen kiểu int sau đó sẽ gọi class maping\_admin, và kiểm tra null hoặc rỗng, sau đó dùng hàm them trong maping\_admin để thêm, tạo một biến p có kiểu phanquyen và thêm iduser, idquyen, vào biến p sau đó lưu tất cả xuống database.

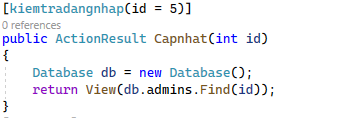
Xử lý bên view :



Hình 3.49

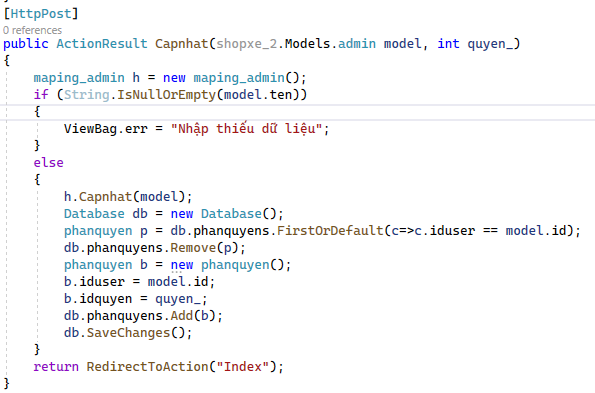
Form sẽ nhận giá trị và gửi về action them để xử lý

Sửa :

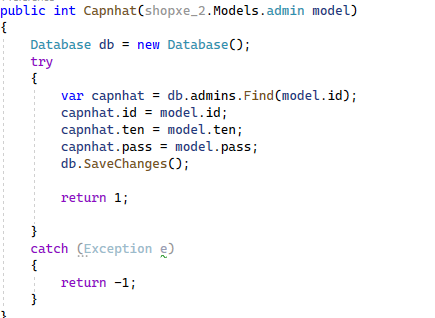


Hình 3.50

Action sẽ nhận vào một id sau đó sửa dụng cú pháp linq và trả về một hãng có id bằng với id truyền vào.



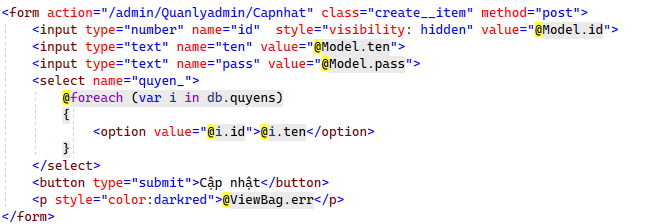
Hình 3.51



Hình 3.52

Action sẽ nhận vào một model kiểu admin và một quyen kiểu int, câu if kiểm tra rổng, tạo biến h có kiểu là class maping\_admin, sau đó dùng hàm capnhat trong maping\_admin đê cập nhật lại admin, sau đó tạo một biến phân quyền p dùng cú pháp linq tìm ra quyền cũ và xóa nó đi sau đó tạo biến b kiểu quyen và cập nhật iduser, idquyen cho nó và lưu xuống database.

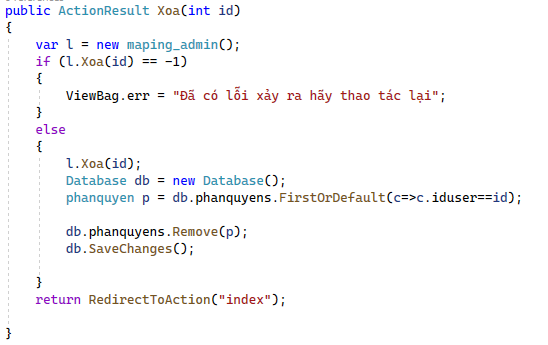
Xử lý bên view :



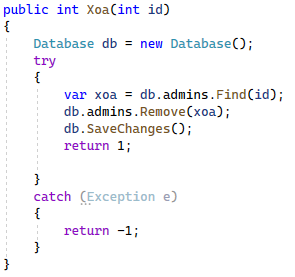
Hình 3.53

Form sẽ nhận giá trị và gửi về action them để xử lý

Xóa :



Hình 3.54

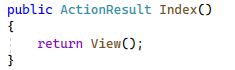


Hình 3.55

Action sẽ không có view, đầu tiên nhận vào một id và gọi hàm xoa trong maping\_dmin, hàm sẽ dùng cú pháp linq tìm và xóa đi admin có id đã nhận, sau đó tạo một biến p kiểu phanquen tìm tới hàng có iduser bằng id trong database và xóa nó sau đó lưu database lại.

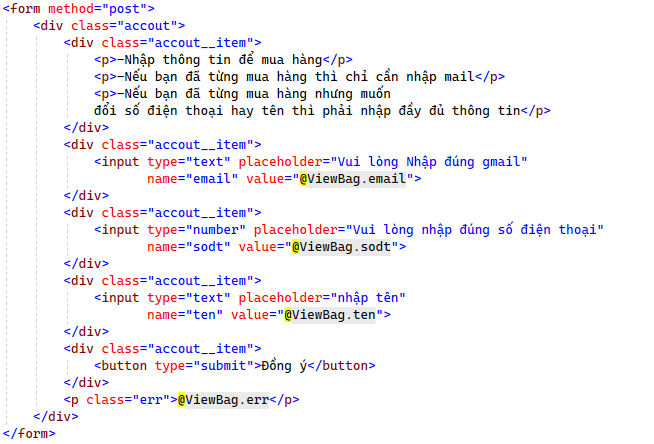
##### Đăng nhập :

Đăng nhập người dùng, quản trị sẽ có một controller xử lý là Dangnhap và một action là index.



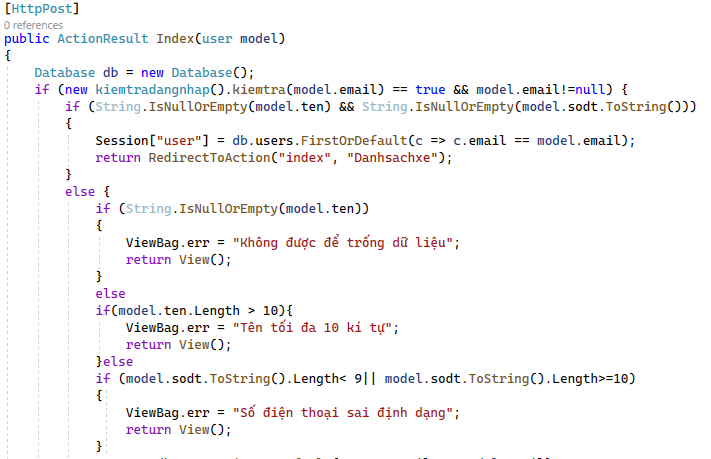
Hình 3.56

Action này sẽ trả ra view của trang đăng nhập.

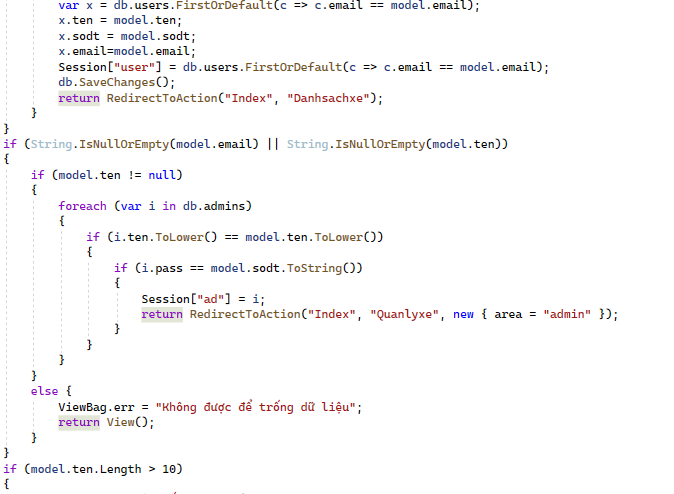


Hình 3.57

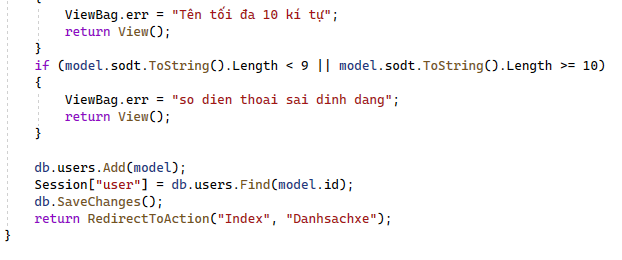
View sẽ có nhiệm vụ là nhận giá trị từ các tag input được người dùng nhập vào và truyền sang action index để xử lý đăng nhập hay đăng ký.



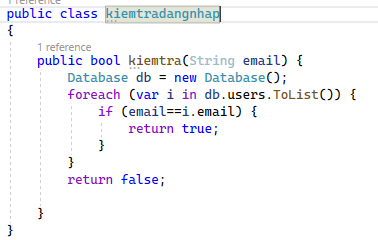
Hình 3.58



Hình 3.59



Hình 3.60



Hình 3.61

Action này sẽ nhận vào một biến model kiểu user và tiếp đến sẽ có một số ràng buộc để tùy chỉnh đăng nhập và đăng ký, đầu tiên dùng if và hàm kiemtra(hình 3.61) để kiểm tra xem email này có tồn tại trong database không, và kiểm tra rổng, null của email. Lúc này sẽ chia ra các trường hợp

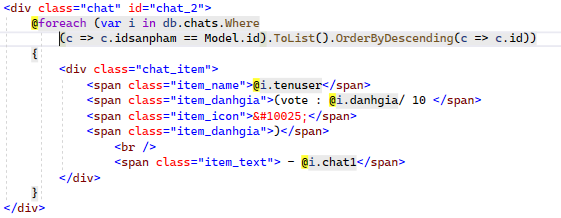
Đầu tiên là email có tồn tại, trường hợp này tiếp tục chia ra thêm hai trường hợp là người dùng muốn dùng thông tin cũ trước đó hay muốn cập nhật lại thông tin cụ thể là tên và số diện thoại, tại trang view người dùng sẽ được thông báo về việc cập nhật thông tin, tiếp tục dùng if để kiểm tra từng trường hợp nếu thấy cả cột tên và số điện thoại người dùng điều bỏ trống thì tạo ra sesion user và tiến hành đăng nhập, giữ lại thông tin cũ cho khách hàng, sau đó dùng else để bắt trường hợp còn lại tức trường hợp người dùng muốn cập nhật thông tin và trong trường hợp này sẽ có tiếp ba if kiểm tra một vài ràng buộc như không được để trống dữ liệu, sai định dạng số điện thoại hay tên quá dài, nếu không sai ràng buộc thì tiến hành dùng cú pháp linq tìm ra email của khách hàng và cập nhật lại các thông tin sau đó tạo sestion để đăng nhập cho người dùng.

Thứ hai là email không tồn tại, tại đây sẽ chia ra hai trường hợp là đăng nhập người dùng hay đăng nhập quản trị if đầu tiên sẽ kiểm tra null, rổng email hoặc tên sau đó sẽ có if kiểm tra null của tên nếu null thì sẽ phản hồi lại người dùng nếu không null sẽ dùng vòng for chạy bảng admin nếu tên bằng với tên admin và số điện thoại bằng với pass thì sẽ tạo session cho admin và chuyển qua trang quản trị. Nếu sai thì lại thông báo ra giao diện, nếu trường hợp email không null thì sẽ có các if tiếp tục kiểm tra ràng buộc số điện thoại, tên, sau cùng là tiến hành add người dùng mới xuống database và tạo session cho người dùng.

##### Đánh giá chấm điểm sản phẩm :

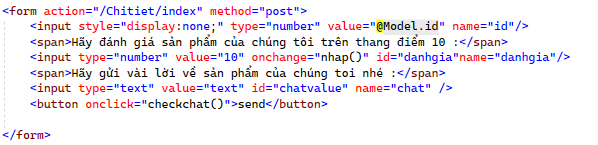
Xủ lý bên view :

Phần đánh giá sẽ nằm trong phần chi tiết sản phẩm



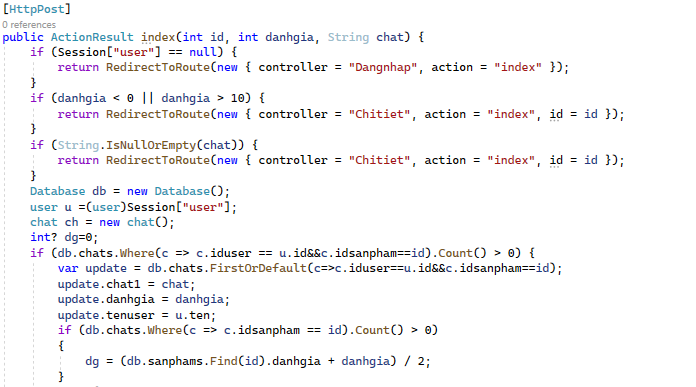
Hình 3.62

View sẽ dùng cú pháp của linq để hiển thị ra tất cả những đánh giá của sản phẩm đang được hiển thị dựa trên id. Dưới đây là khung để nhập đánh giá.

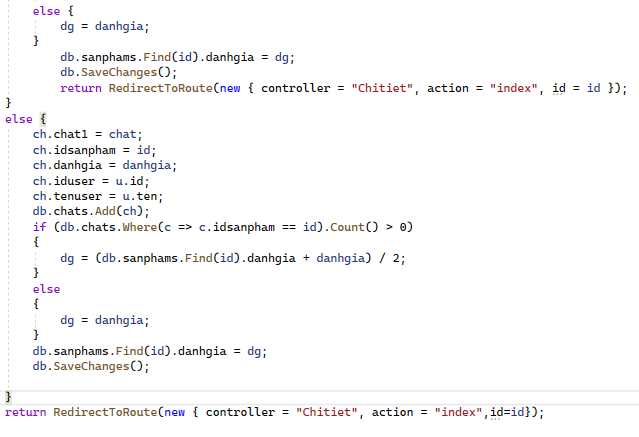


Hình 3.63

Form này sẽ truyền ba giá trị gồm id, danhgia, chat qua action để tiến hành xử lý



Hình 3.64



Hình 3.65

Action sẽ nhận ba tham số từ forrm truyền vào theo methoth post, đầu tiên if kiểm tra đăng nhập nếu phát hiện chưa đăng nhập tức session null sẽ chuyển về trang đăng nhập, if thứ 2 kiểm tra phần vote của người dùng, if 3 kiểm tra null, rổng phần chat, tạo ra biến u là session tức là người dụng hiện tại, tạo ra biến ch kiểu chat, tới đây sẽ chia ra hai trường hợp một là người dùng từng đánh giá và muốn update hai là người dùng đánh giá mới, sau cùng trả về trang chi tiết sản phẩm.

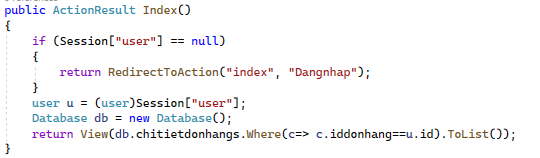
Trường hợp 1 : tạo biến update kiểu chat, tìm ra phần chat của người dùng trong database bằng FistOrDfault và gán cho update sau đó gán lại các giá trị mới cho update, sau đó kiểm tra xem san phẩm đã được đánh giá hay chưa bằng if với điều kiện là đém bảng chat có id của sản phẩm hiện tại, nếu đã có đánh giá thì tính lại trung bình của các đánh giá nếu chưa thì gán cho đánh giá trong bản sản phẩm bằng với đánh giá mà người dùng vừa truyền vào, sau đó lưu lại database.

Trường hợp hai : cập nhật lại các phần tử của biến ch, add biến ch vào bảng chat của database, sau đó tiếp tục tính toán đánh giá như trên trường hợp q và lưu lại database, sau cùng trả về trang chi tiết sản phẩm.

##### Giỏ hàng – đặc hàng và gửi mail tự động :

Giỏ hàng – đặc hàng :

Đây là chức năng giúp người dùng có thể lưu lại những sản phẩm ưng ý và tiến hành đặc hàng, chức năng này cần một controller và 4 action.



3.66

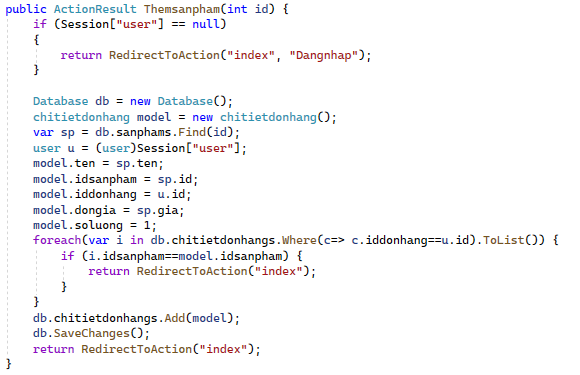
Action này sẽ trả về một danh sách các sản phẩm mà người dùng đã bấm đặc hàng, đầu tiên sẽ có một if kiểm tra session của người dùng để xác nhận đăng nhập, nếu sai điều kiện sẽ trả về view của chức năng đăng nhập, sau đó action sẽ truyền một list các chitietdonhang ra ngoài view.

Xử lý bên view :



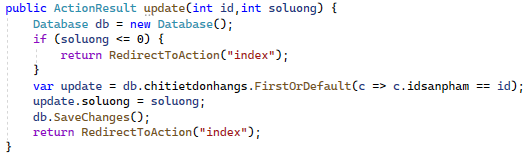
Hình 3.67

View sẽ xuất ra các dữ liệu nhận bên controller cụ thể là các sản ohaamr đã thêm vào giỏ hàng, đồng thời tất cả các sản phẩm đều nằm trong một form, form này sau sẽ truyền qua cho action index của controller mail để tiến hành gửi mail.



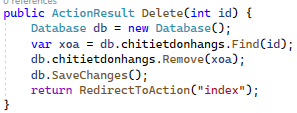
Hình 3.68

Action thêm này sẽ hoạt động khi người dùng bấm thêm sản phẩm vào giỏ hàng, action này sẽ nhận vào id của sản phẩm, sau đó kiểm tra đăng nhập người dùng, tiếp đén sẽ tạo biến model kiểu chitietdonhang cập nhật các biến đó dựa trên biến ap có kiểu là sanpham, biến sp có giá trị chính bằng sản phẩn có id đã nhận vào trước đó, vòng lập foreach để kiểm tra xem người dùng đã có sản phẩm này trong giỏ hàng chưa nếu chưa sẽ thêm mới còn ngược lại sẽ chuyển trang về trang giỏ hàng, cuối cùng là lưu lại tất cả xuống database.



Hình 3.69

Action này sẽ chạy khi khung số lượng của các sản phẩm trong giỏ hàng thay đổi nó sẽ cập nhật lại số lượng sản phẩm cho người dùng, if kiểm tra số lượng có bé hơn không nếu có sẽ trả về số lượng là một, action này nhận vào một biến id và một biến soluong, id sẽ được dùng để tìm ra chitietdonhang và sửa lại, còn soluong là số lượng của người dùng cần đổi, sau đó tạo biến update là chitietsanpham cần sửa, gán lại số lượng và lưu vào database.



Hình 3.70

Action sex nhận vào một id, đây là id của sản phẩm, tao biến xoa, dùng cú pháp linq tìm ra đúng chitietdonhang có id bằng với id đã nhận, gán nó cho xoa, remove đi chitietdonhang đó và lưu lại dattabase, action này sẽ chạy khi người dùng bấm nút xóa.

Gửi mail tự động :

Đây là chức năng khi người dùng bấm đặc hàng thì sẽ được gửi một mail chứa các thông tin về sản phẩm, thông tin liên hệ với cửa hàng.