TRƯỜNG ĐẠI HỌC SƯ PHẠM TPHCM

**KHOA CÔNG NGHỆ THÔNG TIN**

**TRẦN VIỆT TIẾN – K40.104.140**

**XÂY DỰNG HỆ THỐNG WEBSITE BÁN HÀNG ONLINE BẰNG**

**C# ASP.NET MVC**

CHUYÊN NGÀNH: CÔNG NGHỆ PHẦN MỀM

**ĐỒ ÁN TỐT NGHIỆP**

**GIẢNG VIÊN HƯỚNG DẪN: THS. LƯƠNG TRẦN HY HIẾN**

TRƯỜNG ĐẠI HỌC SƯ PHẠM TPHCM

**KHOA CÔNG NGHỆ THÔNG TIN**

**XÂY DỰNG HỆ THỐNG WEBSITE BÁN HÀNG ONLINE BẰNG**

**C# ASP.NET MVC**

**ĐỒ ÁN TỐT NGHIỆP**

**Lời cám ơn**

Em xin chân thành cảm ơn khoa Công Nghệ Thông Tin, trường Đại Học Sư Phạm TP.HCM đã tạo điều kiện tốt cho em thực hiện đề tài này.

Em xin chân thành cảm ơn thầy Lương Trần Hy Hiến, là người đã tận tình, có những trao đổi, hướng dẫn giúp em giải quyết các vấn đề và hoàn thiện đề tài.

Em cũng xin gửi lời cảm ơn sâu sắc đến quý thầy cô trong khoa đã tận tình giảng dạy, trang bị cho em những kiến thức quý báu trong những năm học vừa qua.

Em xin gửi lòng biết ơn sâu sắc đến ba, mẹ, các anh chị và bạn bè đã ủng hộ, giúp đỡ và động viên em trong những lúc khó khăn cũng như trong suốt thời gian học tập và nghiên cứu.

Mặc dù em đã cố gắng hoàn thành tiểu luận trong phạm vi và khả năng cho phép, nhưng chắc chắn sẽ không tránh khỏi những thiếu sót, kính mong sự cảm thông và tận tình chỉ bảo của quý thầy cô.

Người thực hiện

Trần Việt Tiến

**Mục Lục**

[Tóm tắt tiểu luận 5](#_Toc500460939)

[Chương 1: Tổng quan 6](#_Toc500460940)

[1.1 Giới thiệu hình thức bán hàng online 6](#_Toc500460941)

[1.2 Tìm năng từ bán hàng online 6](#_Toc500460942)

[1.3 Ưu điểm 6](#_Toc500460943)

[1.4 Nhược điểm 7](#_Toc500460944)

[1.5 Lý do chọn đề tài 7](#_Toc500460945)

[1.6 Mục tiêu và nội dung của đề tài 8](#_Toc500460946)

[1.7 Công nghệ sử dụng 8](#_Toc500460947)

[Chương 2: Cơ sở lý thuyết 10](#_Toc500460948)

[2.1 ASP.Net là gì? 10](#_Toc500460949)

[2.1.1 ASP.NET không phải là ngôn ngữ lập trình 10](#_Toc500460950)

[2.1.2 Khác biệt giữa ASP.NET và ASP 10](#_Toc500460951)

[2.2 ASP.Net Web Form 12](#_Toc500460952)

[2.3 ASP.Net MVC 13](#_Toc500460953)

[2.3.1 Tổng quan về ASP.NET MVC 13](#_Toc500460954)

[2.3.2 Lựa chọn áp dụng MVC trong xây dựng ứng dụng 15](#_Toc500460955)

[2.3.3 Các tính năng của nền tảng ASP.NET MVC 17](#_Toc500460956)

[2.4 ASP.Net Identity 18](#_Toc500460957)

[2.5 HTML 5 21](#_Toc500460958)

[2.5.1 HTML 5 là gì? 21](#_Toc500460959)

[2.5.2 HTML 5 Web Storage 22](#_Toc500460960)

[2.5.3 HTML 5 Application Cache 22](#_Toc500460961)

[2.5.4 HTML 5 Web Workers 23](#_Toc500460962)

[2.5.5 HTML 5 Geolocation 23](#_Toc500460963)

[2.5.6 HTML 5 Audio tag và Video tag 23](#_Toc500460964)

[2.6 CSS 24](#_Toc500460965)

[2.6.1 Tổng quan Css 24](#_Toc500460966)

[2.6.2 Các đặc tính cơ bản 24](#_Toc500460967)

[2.6.3 Ưu điểm của Css 25](#_Toc500460968)

[2.7 JavaScript 25](#_Toc500460969)

[2.7.1 Tổng quan 25](#_Toc500460970)

[2.7.2 Tại sao phải dùng JavaScript 26](#_Toc500460971)

[Chương 3: Sơ đồ phân tích thiết kế 28](#_Toc500460972)

[3.1 Sơ đồ Class 28](#_Toc500460973)

[3.2 Sơ đồ Usecase 29](#_Toc500460974)

[3.3 Sơ đồ vật lý 30](#_Toc500460975)

[Chương 4: Các tính năng cơ bản 31](#_Toc500460976)

[4.1 Khách hàng 31](#_Toc500460977)

[4.1.1 Đăng nhập 31](#_Toc500460978)

[4.1.2 Đăng ký 31](#_Toc500460979)

[4.1.3 Chỉnh sửa thông tin cá nhân 32](#_Toc500460980)

[4.1.4 Xem sản phẩm 32](#_Toc500460981)

[4.1.5 Giỏ hàng 33](#_Toc500460982)

[4.1.6 Thanh toán 33](#_Toc500460983)

[4.1.7 Quản lý đơn hàng 34](#_Toc500460984)

[4.2 Quản trị viên 34](#_Toc500460985)

[4.2.1 Quản lý danh mục sản phẩm 34](#_Toc500460986)

[4.2.2 Quản lý hãng sản xuất 35](#_Toc500460987)

[4.2.3 Quản lý sản phẩm 35](#_Toc500460988)

[4.2.4 Quản lý khách hàng 36](#_Toc500460989)

[Chương 5: Kết luận 37](#_Toc500460990)

[5.1 Những kết quả đạt được 37](#_Toc500460991)

[5.2 Những điểm cần khắc phục 37](#_Toc500460992)

[Tài liệu tham khảo 38](#_Toc500460993)

**Tóm tắt tiểu luận**

Ngày nay internet đã trở thành một thứ rất phổ biến trong cuộc sống, nó giúp cho mọi người trên thế giới có thể kết nối với nhau, trao đổi thông tin cho nhau. Nhưng mãi đến những năm gần đây, con người mới phát triển ra thêm nhiều tính năng khác từ internet như là mạng xã hôi, game online, không đơn thuần chỉ dùng để trao dổi thông tin, hoặc liên lạc như nhưng năm đầu.

Tiêu biểu nhất trong các tính năng mới, đó là thương mại điện tử. Nếu như trước đây chưa có thương mại điện tử, mọi người muốn mua sắm bất cứ thứ gì thì phải bỏ công ra tới tận nơi bán để mua, tốn rất nhiều thời gian và công sức. Nhưng đối với thương mại điện tử thì mọi chuyện đơn giãn hơn, chỉ cần một thiết bị có thể kết nối internet là chúng ta có thể tự do mua sắm.

Tiểu luận được chia thành chương như sau:

Chương 1 – Tổng quan: Giới thiệu về hình thức bán hàng online. Phần tiếp theo nêu tổng quan tình hình phát triển hình thức bán hàng online tại Việt Nam, ưu nhược điểm cũng như giá trị đầy tiềm năng của hình thức kinh doanh này. Trình bày lý do chọn đề tài, mục tiêu và nội dung thực hiện đề tài và công nghệ sử dụng.

Chương 2 – Cơ sở lý thuyết: Lịch sử ra đời ASP.Net MVC , các khái niệm cơ bản, công nghệ sử dụng.

Chương 3 – Sơ đồ phân tích thiết kế: trình bày sơ đồ phân tích thiết kế theo các mô hình đã học.

Chương 4 – Các tính năng cơ bản của hệ thống

Chương 5 – Kết luận: Trình bày những kết quả đạt được, những điểm cần khắc phục

**Chương 1: Tổng quan**

**1.1 Giới thiệu hình thức bán hàng online**

Bán hàng online, kinh doanh online hay bán hàng trực tuyến là bất kỳ hình thức kinh doanh nào được xảy ra trên internet. Mọi loại hoạt động phát sinh trao đổi, mua bán trên internet đều được gọi là bán hàng online hay bán hàng trên mạng.

Chúng ta hãy liệt kê ra tất cả cụm từ liên quan tới lĩnh vực này nhé: bán hàng online, bán hàng trên mạng, bán hàng trực tuyến, kinh doanh online, kinh doanh trên mạng, kinh doanh trực tuyến v.v… rất rất nhiều cụm từ được dùng để mô tả hành động trao đổi, mua bán qua mạng.

**1.2 Tìm năng từ bán hàng online**

Theo thống kê Google, năm 2016 ở Việt Nam có hơn 50 triệu người sử dụng Internet thường xuyên, việc mua sắm online cũng ngày càng trở nên phổ biến. Vì thế, khi bạn đem sản phẩm của mình bán trên Internet là bạn đang bán hàng trên một thị trường vô cùng rộng lớn, tiềm năng. Vậy đấy, có tận hơn 50 triệu khách hàng đang chờ đón bạn.

Hãy dạo quanh một vòng Internet, giờ đây việc mua hàng qua mạng thực sự đã phát triển rất nhanh và trở nên thông dụng với mỗi người. Hàng ngày, các cửa hàng online được mọc lên như nấm phục vụ đông đảo người dân mua sắm.

**1.3 Ưu điểm**

Khách hàng xuất hiện ở rất nhiều nơi trên Internet, từ các mạng xã hội như Facebook, Zalo, Instagram đến các website, sàn thương mại điện tử. Chính vì vậy, khi bạn tiếp cận khách hàng trên nhiều kênh khác nhau nghĩa là tỉ lệ hàng bán ra sẽ tăng lên, doanh thu và lợi nhuận cũng sẽ tăng trưởng nhanh chóng.

Trong môi trường kinh doanh online bạn sẽ được tham gia vào rất nhiều kênh bán hàng khác như: Mạng xã hội, bán hàng trên website, trên các sàn thương mại điện tử, các chợ rao vặt, các diễn đàn nổi tiếng v.v… Khác với việc bán hàng truyền thống, khách hàng của bạn chỉ là khách qua đường, chỉ là những người ở gần cửa hàng của bạn thì với bán hàng online, khách hàng của bạn sẽ đến từ khắp mọi nơi, khắp mọi vùng miền và trên thế giới.

**1.4 Nhược điểm**

Có một điều mà ai cũng thấy khi gia nhập cộng đồng mạng đó là SỰ BẢO MẬT. Sự bảo mật ở internet không được đảm bảo, chưa có một luật hay chính sách nào quy định rõ về an ninh mạng. Chính vì vậy khó để khách hàng xác minh được họ đang mua hàng của ai, thông tin cá nhân của khách hàng có được bảo mật hay không, thông tin tài khoản dữ liệu thẻ thanh toán thẻ tín dụng của họ liệu có bị lộ ra ngoài không.

Sự bảo mật ở internet không được đảm bảo, chưa có một luật hay chính sách nào quy định rõ về an ninh mạng.

Chưa kể khách hàng lo lắng việc mua hàng có đúng chất lượng như lời PR giới thiệu, hình ảnh đẹp lung linh như các trang bán hàng đăng tải hay không.

**1.5 Lý do chọn đề tài**

Với xu thế hiện nay, bán hàng online hầu như quá phổ biến tại Việt Nam. Internet ngày càng phổ thông, chỉ với một chiếc smart phone là có thể kết nối internet một cách dễ dàng.

Trong một xã hội hiện đại hóa, công nghiệp hóa. Con người dành nhiều thời gian cho công việc hơn, nên sẽ cần những loại hình dịch vụ tiện lớn nhất, và bán hàng online là một dịch vụ như vậy. Mọi người có thể mua sắm thoải mái mà không phải tốn nhiều công sức. Tha hồ lựa chọn một số lượng sản phẩm dồi dào và đa dạng. Sản phẩm sau khi được mua sẽ được vận chuyển tới tận nhà.

**1.6 Mục tiêu và nội dung của đề tài**

Xây dựng một hệ thống cho phép người sử dụng có thể đăng bán mọi sản phẩm mà mình muốn. Giao diện dễ sử dụng, dễ dàng thao tác với những người không rành về tin học.

Xây dựng hệ thống bao gồm 2 thành phần riêng biệt. Phần để bán sản phẩm, và phần để quản lý:

Phần bán sản phẩm: Xây dựng một giao diện dễ sử dụng, với cách thiết kế rõ ràng, kết hợp với những icon phù hợp. Giúp cho người sử dụng website dễ dàng tìm ra được sản phẩm mình mong muốn, xem chi tiết và đặt hàng.

Phần quản lý sản phẩm: Xây dựng một giao diện dễ sử dụng, với cách thiết kế rõ ràng, kết hợp với những icon phù hợp. Giúp cho quản trị viễn dễ dàng quản lý hệ thống, chỉnh sửa sản phẩm, đơn hàng, khách hàng,…

**1.7 Công nghệ sử dụng**

Sử dụng ngôn ngữ lập trình C#, với công nghệ ASP.Net của microsoft, được thiết kế để chuyên làm về web.  
 Sử dụng mô hình MVC (Model – View - Controller).

Sử dụng công nghệ Identity của microsoft.

Sử dụng Css 3, Html 5, javasctip, jqurey, datatable để thiết kế giao diện người dùng. Sử lý logic trên giao diện người dùng.

**Chương 2: Cơ sở lý thuyết**

**2.1 ASP.Net là gì?**

**2.1.1 ASP.NET không phải là ngôn ngữ lập trình**

ASP.NET là một công nghệ dùng để phát triển các ứng dụng về mạng. Chúng ta cần thấy rằng ASP.NET cũng như ASP là một Framework để thiết lập các ứng dụng chứ không phải là một ngôn ngữ lập trình.

Ngôn ngữ lập trình được dùng để diễn đạt ASP là JavaScript và VBScript.

Trong lúc đó, ngôn ngữ lập trình được dùng để diển đạt ASP.NET thông thường là Visual Basic .NET(VB.NET), lưu ý là Visual Basic đầy đủ còn gọi là Visual Basic 7 chứ không phải là VBScript, ngoài ra còn có C# (C sharp), C++, JScript ... Bạn có thể viết ASP.NET bằng VB.NET hay C# chẳng hạn và chúng sẽ được compiled at run time, chứ không phải là Script như trong ASP.NET.

NET gồm có hai phần:

Framework và Integrated Development Environment (IDE).

* Framework cung cấp tất cả những gì cần thiết căn bản.
* Còn IDE cung cấp một môi trường giúp ta triển khai dễ dàng, nhanh chóng hơn. Trong .NET, C# và VB.NET đều dùng cùng một IDE. Riêng với chúng ta, tạm thời lúc này ta sẽ dùng Notepad++ như là IDE chính.

**2.1.2 Khác biệt giữa ASP.NET và ASP**

Hoàn toàn không phải là sản phẩm Upgrade từ ASP, ASP.NET được làm lại từ số không và được thay đổi tận gốc rễ.

Các thay đổi căn bản:

* Tập tin của ASP.NET có phần mở rộng là .ASPX, còn tập tin của ASP là .ASP.
* Tập tin của ASP.NET được phân tích ngữ pháp bởi XSPISAPI.DLL, còn tập tin của ASP được phân tích bởi ASP.DLL.
* ASP.NET là kiểu mẫu lập trình phát động bằng sự kiện (event driven), còn các trang ASP được thi hành theo thứ tự tuần tự từ trên xuống dưới.
* ASP.NET xử dụng compiled code nên rất nhanh, còn ASP dùng interpreted code do đó hiệu suất và tốc độ phát triển cũng thua sút hẳn. ASP.NET hỗ trợ gần 25 ngôn ngữ lập trình mới với .NET và chạy trong môi trường biên dịch (compiled environment), còn ASP chỉ chấp nhận VBScript và JavaScript nên ASP chỉ là một scripted language trong môi trường thông dịch(in the interpreter environment). Không những vậy, ASP.NET còn kết hợp nhuần nhuyễn với XML (Extensible Markup Language) để chuyển vận các thông tin (information) qua mạng.
* ASP.NET hỗ trợ tất cả các browser và quan trọng hơn nữa là hỗ trợ các thiết bị lưu động (mobile devices).

Tuy nhiên, trong lúc này bạn chưa cần quá quan tâm đến những cái "cao siêu" và khó hiểu đó. Tạm thời dù bạn đã qua ASP hay chưa thì chúng ta chỉ cần tạm hiểu đơn giản thế này:

ASP.NET không phải là một ngôn ngữ lập trình, nó là môi trường dùng để tạo và chạy các ứng dụng Web động và có tương tác. Các trang web viết bằng ngôn ngữ này có phần mở rộng là .aspx (ví dụ HelloWorld.aspx) thay vì .htm hay .html. Nội dung file ASPX về cơ bản hao hao giống file Html bình thường, nó bao gồm các cú pháp html cùng với XMK, và ASP.NET code

**2.2 ASP.Net Web Form**

ASP.NET WebForms là một phần của khung ứng dụng ASP.NET. Nó là một trong ba mô hình lập trình khác nhau mà bạn có thể sử dụng để tạo ứng dụng web ASP.NET, phần còn lại là ASP.NET MVC và ASP.NET Web Pages.

Web Forms là trang mà người dùng yêu cầu thông qua trình duyệt của họ và hình thức giao diện người dùng (UI) cung cấp cho các ứng dụng của bạn có cái nhìn và cảm nhận. Những trang này được viết bằng cách sử dụng một sự kết hợp của HTML, Server Control và Server code. Khi người dùng yêu cầu từ một trang, nó được biên dịch và thực thi từ máy chủ và sau đó nó được tạo ra các HTML markup (đánh dấu) mà trình duyệt có thể render (chuyển tải, làm cho).

Sử dụngVisual Studio, bạn có thể tạo ASP.NET Web Forms sử dụng một cách mạnh mẽ IDE. Ví dụ, điều này cho phép bạn dễ dàng kéo và thả Server Controls để tạo thiết lập (tạo ra) giao diện trang Web Forms. Sau đó bạn có thể dễ dàng thiết lập thuộc tính (Properties), phương thức (methods), và các sự kiện (event) cho các điểu khiển (Control) hoặc cho việc xác định hành vi của trang, nhìn và cảm nhận, và cứ thế. Để viết mã máy chủ (server code) để xử lý logic cho các trang, bạn có thể dùng ngôn ngữ .NET như làVisual Basic hoặc C# (CSharp).

ASP.NET Web Form cung cấp (offer):

* + Tách biệt HTML và mã giao diện người dùng đến ứng dụng logoc.
  + Một bộ phong phú Server Control cho common tasks (chức năng phổ biến, nhiệm vụ), bao gồm cả truy cập dữ liệu (data access).
  + Dữ liệu ràng buộc mạnh mẽ, với sự hỗ trợ công cụ tuyệt vời (great)
  + Hỗ trợ cho sử dụngAjax, sự kiện nếu bạn không biết về Javascript

**2.3 ASP.Net MVC**

**2.3.1 Tổng quan về ASP.NET MVC**

Mẫu kiến trúc Model – View – Controller được sử dụng nhằm chi ứng dụng thành ba thành phần chính: model, view và controller. Nền tảng ASP.NET MVC giúp cho chúng ta có thể tạo được các ứng dụng web áp dụng mô hình MVC thay vì tạo ứng dụng theo mẫu ASP.NET Web Forsm. Nền tảng ASP.NET MVC có đặc điểm nổi bật là nhẹ (lighweigt), dễ kiểm thử phần giao diện (so với ứng dụng Web Forms), tích hợp các tính năng có sẵn của ASP.NET. Nền tảng ASP.NET MVC được định nghĩa trong namespace System.Web.Mvc và là một phần của name space System.Web.

MVC là một mẫu thiết kế (design pattern) chuẩn mà nhiều lập trình viên đã quen thuộc. Một số loại ứng dụng web sẽ thích hợp với kiến trúc MVC. Một số khác vẫn thích hợp với ASP.NET Web Forms và cơ chế postbacks. Đôi khi có những ứng dụng kết hợp cả hai kiến trúc trên.

Nền tảng MVC bao gồm các thành phần dưới đây:

* **Models:** Các đối tượng Models là một phần của ứng dụng, các đối tượng này thiết lập logic của phần dữ liệu của ứng dụng. Thông thường, các đối tượng model lấy và lưu trạng thái của model trong CSDL. Ví dụ như, một đối tượng Product (sản phẩm) sẽ lấy dữ liệu từ CSDL, thao tác trên dữ liệu và sẽ cập nhật dữ liệu trở lại vào bảng Products ở SQL Server. Trong các ứng dụng nhỏ, model thường là chỉ là một khái niệm nhằm phân biệt hơn là được cài đặt thực thụ, ví dụ, nếu ứng dụng chỉ đọc dữ liệu từ CSDL và gởi chúng đến view, ứng dụng khong cần phải có tầng model và các lớp lien quan. Trong trường hợp này, dữ liệu được lấy như là một đối tượng model (hơn là tầng model).
* **Views:** Views là các thành phần dùng để hiển thị giao diện người dùng (UI). Thông thường, view được tạo dựa vào thông tin dữ liệu model. Ví dụ như, view dùng để cập nhật bảng Products sẽ hiển thị các hộp văn bản, drop-down list, và các check box dựa trên trạng thái hiện tại của một đối tượng Product.
* **Controllers:** Controller là các thành phần dùng để quản lý tương tác người dùng, làm việc với model và chọn view để hiển thị giao diện người dùng. Trong một ứng dụng MVC, view chỉ được dùng để hiển thị thông tin, controller chịu trách nhiệm quản lý và đáp trả nội dung người dùng nhập và tương tác với người dùng. Ví dụ, controller sẽ quản lý các dữ liệu người dùng gởi lên (query-string values) và gởi các giá trị đó đến model, model sẽ lấy dữ liệu từ CSDL nhờ vào các giá trị này.

Mẫu MVC giúp bạn tạo được các ứng dụng mà chúng phân tách rạch ròi các khía cạnh của ứng dụng (logic về nhập liệu, logic xử lý tác vụ và logic về giao diện). Mẫu MVC chỉ ra mỗi loại logic kể trên nên được thiếp lập ở đâu trên ứng dụng. Logic giao diện (UI logic) thuộc về views. Logic nhập liệu (input logic) thuộc về controller. Và logic tác vụ (Business logic – là logic xử lý thông tin, mục đích chính của ứng dụng) thuộc về model. Sự phân chia này giúp bạn giảm bớt được sự phức tạp của ứng dụng và chỉ tập trung vào mỗi khía cạnh cần được cài đặt ở mỗi thời điểm. Ví dụ như bạn chỉ cần tập trung vào giao diện (views) mà không phải quan tâm đến logic xử lý thông tin của ứng dụng.

Để quản lý sự phức tạp của ứng dụng, mẫu MVC giúp cho chúng ta có thể kiểm thử ứng dụng dễ dàng hơn hẳn so với khi áp dụng mẫu Web Forms. Ví dụ, trong một ứng dụng ASP.NET Web Forms, một lớp thường được sử dụng để hiển thị thông tin xuất ra cho người dùng và đồng thời xử lý thông tin người dùng nhập. Việc xây dựng các bộ test tự động cho ứng dụng Web Forms là rất phức tạp, bởi để kiểm thử mỗi trang web, bạn phải khởi tạo đối tượng trang, khởi tạo tất cả các control được sử dụng trong trang và các lớp phụ thuộc trong ứng dụng. Và bởi vì có quá nhiều lớp cần được khởi tạo để chạy được trang, thật khó để có thể viết các test chỉ tập trung vào một khía cạnh nào đó của ứng dụng. Và vì thế, kiểm thử đối với các ứng dụng dứa trên nền tảng Web Forms sẽ khó khăn hơn nhiều so với khi áp dụng trên ứng dụng MVC. Hơn thế nữa, việc kiểm thử trên nền tảng Web Forms yêu cầu phải sử dụng đến web server. Nền tảng MVC phân tách các thành phần và sử dụng các interface (khái niệm giao diện trong lập trình hướng đối tượng), và nhờ đó có thể kiểm thử các thành phần riêng biệt trong tình trạng phân lập với các yếu tố còn lại của ứng dụng.

Sự phân tách rạch ròi ba thành phần của ứng dụng MVC còn giúp cho việc lập trình diễn ra song song. Ví dụ như một lập trình viên làm việc với view, lập trình viên thứ hai lo cài đặt logic của controller và lập trình viên thứ ba có thể tập trung vào logic tác vụ của model tại cùng một thời điểm.

**2.3.2 Lựa chọn áp dụng MVC trong xây dựng ứng dụng**

Bạn cần phải xem xét kỹ càng việc áp dụng mô hình ASP.NET MVC hay mô hình ASP.NET Web Forms khi xây dựng một ứng dụng. Mô hình MVC không phải là mô hình thay thế cho Web Forms, bạn có thể dùng một trong hai mô hình.

Trước khi quyết định sử dụng MVC hay Web Forms cho một web site cụ thể, bạn cần phải phân tích lợi ích khi chọn một trong hai hướng.

**Lợi ích của ứng dụng web dựa trên mô hình MVC**

* Dễ dàng quản lý sự phức tạp của ứng dụng bằng cách chia ứng dụng thành ba thành phần model, view, controller
* Nó không sử dụng view state hoặc server-based form. Điều này tốt cho những lập trình viên muốn quản lý hết các khía cạnh của một ứng dụng.
* Nó sử dụng mẫu Front Controller, mẫu này giúp quản lý các requests (yêu cầu) chỉ thông qua một Controller. Nhờ đó bạn có thể thiết kế một hạ tầng quản lý định tuyến. Để có nhiều thông tin hơn, bạn nên xem phần Front Controller trên web site MSDN
* Hỗ trợ tốt hơn cho mô hình phát triển ứng dụng hướng kiểm thử (TDD)
* Nó hỗ trợ tốt cho các ứng dụng được xây dựng bởi những đội có nhiều lập trình viên và thiết kế mà vẫn quản lý được tính năng của ứng dụng

**Lợi ích của ứng dụng được xây dựng trên nền tảng Web Forms**

* Nó hỗ trợ cách lập trình hướng sự kiện, quản lý trạng thái trên giao thức HTTP, tiện dụng cho việc phát triển các ứng dụng Web phục vụ kinh doanh. Các ứng dụng trên nền tảng Web Forms cung cấp hàng tá các sự kiện được hỗ trợ bởi hàng trăm các server controls.
* Sử dụng mẫu Page Controller. Xem thêm ở mục Page Controller trên MSDN
* Mô hình này sử dụng view state hoặc server-based form, nhờ đó sẽ giúp cho việc quản lý trạng thái các trang web dễ dàng.
* Nó rất phù hợp với các nhóm lập trình viên quy mô nhỏ và các thiết kế, những người muốn tận dụng các thành phần giúp xây dựng ứng dụng một cách nhanh chóng.
* Nói tóm lại, áp dụng Web Forms giúp giảm bớt sự phức tạp trong xây dựng ứng dụng, bởi vì các thành phần (lớp Page, controls,…) được tích hợp chặc chẽ và thường thì giúp bạn viết ít code hơn là áp dụng theo mô hình MVC.

**2.3.3 Các tính năng của nền tảng ASP.NET MVC**

Tách bạch các tác vụ của ứng dụng (logic nhập liệu, business logic, và logic giao diện), dễ dàng kiểm thử và mặc định áp dụng hướng phát triển TDD. Tất cả các tính năng chính của mô hình MVC được cài đặt dựa trên interface và được kiểm thử bằng cách sử dụng các đối tượng mocks, mock object là các đối tượng mô phỏng các tính năng của những đối tượng thực sự trong ứng dụng. Bạn có thể kiểm thử unit-test cho ứng dụng mà không cần chạy controller trong tiến trình ASP.NET, và điều đó giúp unit test được áp dụng nhanh chóng và tiện dụng. Bạn có thể sử dụng bất kỳ nền tảng unit-testing nào tương thích với nền tảng .NET.

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. Bạn có thể nhúng thêm view engine, cơ chế định tuyến cho URL, cách kết xuất tham số của action-method và các thành phần khác. ASP.NET MVC cũng hỗ trợ việc sử dụng Dependency Injection (DI) và Inversion of Control (IoC). DI cho phép bạn gắn các đối tượng vào một lớp cho lớp đó sử dụng thay vì buộc lớp đó phải tự mình khởi tạo các đối tượng. IoC quy định rằng, nếu một đối tượng yêu cầu một đối tượng khác, đối tượng đầu sẽ lấy đối tượng thứ hai từ một nguồn bên ngoài, ví dụ như từ tập tin cấu hình. Và nhờ vậy, việc sử dụng DI và IoC sẽ giúp kiểm thử dễ dàng hơn.

ASP.NET MVC có thành phần ánh xạ URL mạnh mẽ cho phép bạn xây dựng những ứng dụng có các địa chỉ URL xúc tích và dễ tìm kiếm. Các địa chỉ URL không cần phải có phần mở rộng của tên tập tin và được thiết kế để hỗ trợ các mẫu định dạng tên phù hợp với việc tối ưu hóa tìm kiếm (URL) và phù hợp với lập địa chỉ theo kiểu REST.

Hỗ trợ sử dụng đặc tả (các thẻ) của các trang ASP.NET(.aspx), điều khiển người dùng (.ascx) và trang master page (.marter). Bạn có thể sử dụng các tính năng có sẵn của ASP.NET như là sử dụng lồng các trang master page, sử dụng in-line expression (<%= %>), sử dụng server controls, mẫu, data-binding, địa phương hóa (localization) và hơn thế nữa.

Hỗ trợ các tính năng có sẵn của ASP.NET như cơ chế xác thực người dùng, quản lý membership, quyền, output caching và data caching, seession và profile, quản lý tình trạng ứng dụng, hệ thống cấu hình…

Kể từ ASP.NET MVC 3 còn bổ sung một view engine mới là Razor View Engine cho phép thiết lập các view nhanh chóng, dễ dàng và tốn ít công sức hơn so với việc sử dụng Web Forms view engine.

**2.4 ASP.Net Identity**

Khi câu chuyện membership trong ASP.NET đã phát triển qua nhiều năm, nhóm ASP.NET đã học được rất nhiều từ phản hồi từ khách hàng.

Giả định rằng người dùng đăng nhập bằng cách nhập tên người dùng và mật khẩu mà họ đã đăng ký trong ứng dụng của riêng bạn không còn hợp lệ. Các trang web đã trở nên xã hội. Người dùng tương tác với nhau theo thời gian thực thông qua các kênh xã hội như Facebook, Twitter và các trang web xã hội khác. Các nhà phát triển muốn người dùng có thể đăng nhập với đặc điểm xã hội của họ để họ có thể có một trải nghiệm phong phú trên các trang web của họ. Một hệ thống membership hiện đại phải cho phép các đăng nhập dựa trên chuyển hướng đến các nhà cung cấp xác thực như Facebook, Twitter và những người khác. Khi phát triển web phát triển, các mô hình phát triển web cũng vậy. Đơn vị kiểm tra mã ứng dụng đã trở thành một mối quan tâm cốt lõi cho các nhà phát triển ứng dụng. Trong năm 2008 ASP.NET đã thêm một khung công tác mới dựa trên mô hình Model-View-Controller (MVC), một phần để giúp các nhà phát triển xây dựng các ứng dụng ASP.NET có thể kiểm tra. Các nhà phát triển muốn kiểm tra đơn vị logic ứng dụng của họ cũng muốn có thể làm điều đó với hệ thống membership.

Xem xét những thay đổi này trong phát triển ứng dụng web, ASP.NET Identity đã được phát triển với các mục tiêu sau:

**Một hệ thống Nhận dạng của ASP.NET**

* ASP.NET Identity có thể được sử dụng với tất cả các khuôn khổ ASP.NET, chẳng hạn như ASP.NET MVC, Web Forms, Web Pages, Web API, và SignalR.
* ASP.NET Identity có thể được sử dụng khi bạn đang xây dựng các ứng dụng web, điện thoại, cửa hàng hoặc ứng dụng lai.
* Dễ dàng kết nối dữ liệu tiểu sử về người dùng
* Bạn có quyền kiểm soát giản đồ của người dùng và thông tin tiểu sử. Ví dụ: bạn có thể dễ dàng kích hoạt hệ thống để lưu trữ ngày sinh được nhập bởi người dùng khi họ đăng ký tài khoản trong ứng dụng của bạn.

**Kiểm soát lâu dài**

* Theo mặc định, hệ thống Nhận dạng ASP.NET lưu trữ tất cả thông tin người dùng trong cơ sở dữ liệu. ASP.NET Identity sử dụng Entity Framework Code First để thực hiện tất cả cơ chế persistence của nó.
* Vì bạn kiểm soát lược đồ cơ sở dữ liệu, các tác vụ phổ biến như thay đổi tên bảng hoặc thay đổi kiểu dữ liệu của khóa chính là đơn giản để làm.
* Dễ dàng kết nối các cơ chế lưu trữ khác nhau như SharePoint, Dịch vụ Bảng lưu trữ Azure, Cơ sở dữ liệu NoSQL, v.v ... mà không phải ném các trường hợp ngoại lệ System.NotImplementedExceptions .

**Khả năng kiểm tra đơn vị**

* ASP.NET Identity làm cho ứng dụng web thêm đơn vị kiểm chứng được. Bạn có thể viết bài kiểm tra đơn vị cho các phần của ứng dụng của bạn sử dụng ASP.NET Identity.

**Role provider**

* Role provider cho phép bạn hạn chế quyền truy cập vào các phần của ứng dụng theo vai trò. Bạn có thể dễ dàng tạo ra các vai trò như "Quản trị" và thêm người dùng vào vai trò.

**Claims Based**

* ASP.NET Identity hỗ trợ chứng thực dựa trên xác nhận quyền sở hữu, nơi mà danh tính của người dùng được đại diện như một tập hợp yêu cầu. Các yêu cầu bồi thường cho phép các nhà phát triển diễn đạt nhiều hơn trong việc mô tả danh tính của người dùng hơn là cho phép các vai trò. Trong khi vai trò membership chỉ là một boolean (membership hoặc không phải membership), khiếu nại có thể bao gồm thông tin phong phú về danh tính và membership của người dùng.

**Social Login Providers**

* Bạn có thể dễ dàng thêm các đăng nhập xã hội như Tài khoản Microsoft, Facebook, Twitter, Google và các ứng dụng khác vào ứng dụng của bạn và lưu trữ dữ liệu người dùng cụ thể trong ứng dụng của bạn.

**Azure Active Directory**

* Bạn cũng có thể thêm chức năng đăng nhập bằng Azure Active Directory và lưu trữ dữ liệu người dùng cụ thể trong ứng dụng của bạn. Để biết thêm thông tin, xem Tài nguyên Tổ chức trong Tạo Dự án Web ASP.NET trong Visual Studio 2013

**OWIN**

* Xác thực ASP.NET bây giờ dựa trên middleware OWIN có thể được sử dụng trên bất kỳ máy chủ dựa trên OWIN. ASP.NET Identity không có bất kỳ sự phụ thuộc vào System.Web. Nó là một khuôn khổ hoàn toàn OWIN tuân thủ và có thể được sử dụng trong bất kỳ ứng dụng OWIN tổ chức.
* ASP.NET Identity sử dụng OWIN Authentication để đăng nhập / đăng xuất khỏi người dùng trong trang web. Điều này có nghĩa là thay vì sử dụng FormsAuthentication để tạo ra cookie, ứng dụng sử dụng CookieAuthentication OWIN để làm điều đó.

**NuGet package**

* ASP.NET Identity được phân phối lại dưới dạng một gói NuGet được cài đặt trong các mẫu ASP.NET MVC, Web Forms và Web API kèm theo Visual Studio 2013. Bạn có thể tải gói NuGet này từ thư viện NuGet.
* Việc phát hành ASP.NET Identity như là một gói phần mềm NuGet làm cho nhóm ASP.NET dễ dàng hơn trong việc lặp lại các tính năng và sửa lỗi mới và cung cấp cho các nhà phát triển một cách nhanh nhẹn.

**2.5 HTML 5**

**2.5.1 HTML 5 là gì?**

Về cơ bản, HTML 5 là một phiên bản mới sửa đổi thứ 5 của ngôn ngữ World Wide Web: the Hypertext Markup Language (HTML).HTML 5 sẽ cho phép một lớp ứng dụng web mới ra đời, hỗ trợ nội dung đa phương tiện và các chức năng offline mà không cần đến những công nghệ độc quyền đi kèm.

**2.5.2 HTML 5 Web Storage**

Với HTML5, trang web có thể lưu dữ liệu ở local bằng trình duyệt web của người dung. Trước đây việc lưu trữ local của cookies, tuy nhiên Web Strorage bảo mật và chạy nhanh hơn. Dữ liệu không gửi lên server nên do đó chỉ người dùng có thể truy cập được khi dung. Nó có thể lưu một lượng dự liệu lớn mà không làm ảnh hưởng đến hiệu suất của website. Dữ liệu được lưu thành một cặp khóa / giá trị, và trang web chỉ có thể truy xuất đến dữ liệu chính nó tạo ra nghĩa là trang nào chỉ thấy dữ liệu trang web đó.

Ví dụ một ứng dụng tra cứu sách trực tuyến, các sách đã được tra sẽ được lưu lại trên máy của người dung. Khi cần tra lại, máy người dung sẽ không cần kết nối đến server để tải lại những dữ liệu cũ.

Web Storage được hỗ trợ trên trình duyệt Internet Explorer 8 +, Firefox, Opera, Chrome, Safari.

Đối tượng localStorage: stores data không thời hạn. Dữ liệu sẽ không bị xóa khi tắt trình và nó luôn luôn có sẵn khi ta cần truy xuất.

**2.5.3 HTML 5 Application Cache**

Với HTML 5 bạn dễ dàng tạo ra ứng dụng web offline mà không cần đến kết nối internet.

**Application cache có ba lợi thế sau:**

* Duyệt web offline: người dùng có thể dùng ứng dụng web offline mà không cần kết nối internet.
* Tốc độ: cache có tốc độ nhanh.
* Giảm tải cho server: trình duyệt chỉ cần tải những cập nhật từ server.

Cơ bản về Cache Manifest

**Nếu một ứng dụng đã được cache thì ứng dụng đó vẫn còn cache cho tới khi gặp một trong các việc sau:**

* Người dùng xóa cache
* File manifest bị chỉnh sửa
* Ứng dụng cache là một chương trình cập nhật.

**2.5.4 HTML 5 Web Workers**

Khi thực thi một scripts trong trang HTML thì trang sẽ không thực hiện giao tiếp với server cho đến khi script hoàn thành. Một web workers là một Javascript chạy dưới nền web, độc lập với các script khác và không làm ảnh hưởng đến hiệu suất của trang. Bạn có thể làm mọi thứ như click chuột, quét chọn… trong khi web worker đang chạy.

Web worker được hỗ trợ bởi hầu hết các trình duyệt, ngoại trừ Internet Explorer. Kiểm tra trình duyệt có hỗ trợ Web Worker.

**2.5.5 HTML 5 Geolocation**

HTML 5 Geolocation API được sử dụng để định vị vị trí địa lý người dùng. Việc xác định vị trí người dùng là thỏa thuận giữa hai bên, do đó nếu người đung không chấp thuận cho biết vị trí của mình thì ta không biết được.

**2.5.6 HTML 5 Audio tag và Video tag**

Tag <audio> định nghĩa âm thanh như nhạc hay trường audio khác.

Tag <video> xác định một video, chẳng hạn như một phim hoặc một trường video.

Thường dùng với tag <source> để hiển thị được nhiều nội dung hơn.

Dòng text nằm giữa <audio> </audio> sẽ hiển thị khi trình duyệt không hỗ trợ tag <audio>

Trình duyệt hỗ trợ: Internet Explorer 9+, Firefox, Opera, Chrome, Safari.

**2.6 CSS**

**2.6.1 Tổng quan Css**

CSS là chữ viết tắt của cụm từ tiếng Anh “Cascading Style Sheet”, là kiểu thiết kế sử dụng nhiều lớp định dạng chồng lên nhau. CSS được tổ chức World Wide Web (W3C) giới thiệu vào năm 1996. Cách đơn giản nhất để hiểu CSS là hãy coi nó như một phần mở rộng của HTML để giúp đơn giản hóa và cải tiến việc thiết kế cho các trang web.

Khi bắt đầu một Style Sheets, thì bắt buộc mở bằng <HEAD> và kết thúc bằng

</HEAD> và tiếp theo sau đó là khai báo <STYLE> và kết thúc bằng </STYLE>

**2.6.2 Các đặc tính cơ bản**

CSS quy định cách hiển thị của các thẻ HTML bằng cách quy định các thuộc tính của các thẻ đó (font chữ, màu sắc). Để cho thuận tiện có thể đặt toàn bộ các thuộc tính của thẻ vào trong một file riêng có phần mở rộng là “.css”, thường người ta hay đặt tên nó là stylesheet.css.

CSS phá vỡ giới hạn trong thiết kế Web, bởi chỉ cần một file CSS có thể cho phép quản lí định dạng và layout trên nhiều trang khác nhau. Các nhà phát triển Web có thể định nghĩa sẵn thuộc tính của một số thẻ HTML nào đó và sau đó nó có thể dùng lại trên nhiều trang khác.

- Có thể khai báo CSS bằng nhiều cách khác nhau. Có thể đặt đoạn CSS phía trong thẻ <Head>…</Head>, hoặc ghi nó ra một file riêng với phần mở rộng “.css”, ngoài ra còn có thể đặt chúng trong từng thẻ HTML riêng biệt.

**2.6.3 Ưu điểm của Css**

CSS có thể tách riêng phần định dạng ra khỏi nội dung một trang web, do đó sẽ rất thuận tiện khi muốn thay đổi giao diện của một trang web.

CSS là một sợi chỉ xuyên suốt trong quá trình thiết kế một website bởi vì nó cho phép nhà thiết kế kiểm soát toàn bộ giao diện, kiểu cách và sự sắp đặt của nhiều trang hay nhiều đối trong một lần định nghĩa. Để thay đổi tổng thể hay nhiều đối tượng có cùng Style, chỉ cần thay đổi Style đó và lập tức tất cả các thành phần áp dụng Style đó sẽ thay đổi theo. Nó tiết kiệm công sức rất nhiều.

Do định nghĩa các Style có thể được tách riêng ra khỏi nội dung của trang web, chúng được các trình duyệt load một lần và sử dụng cho nhiều lần, do đó giúp các trang web nhẹ hơn và chạy nhanh hơn.

**2.7 JavaScript**

**2.7.1 Tổng quan**

Javascript là một ngôn ngữ chương trình máy tính động. Javascript được nhúng hoặc tích hợp vào tập tin HTML, dùng để tạo các script ở máy client và máy server. Các script ở client được thực thi tại trình duyệt và các script ở server được thực hiện trên server. JavaScript và Java là hai ngôn ngữ hoàn toàn khác nhau, cả về khái niệm và thiết kế.

JavaScript là ngôn ngữ lập trình HTML và Web. JavaScript là một ngôn ngữ kịch bản dựa trên đối tượng nhằm phát triển các ứng dụng Internet chạy trên phía client và phía server. Javascript rất hữu ích trong việc xây dựng các hệ thống HTML có thể tương tác với người dùng

JavaScript được phát minh bởi Brendan Eich vào năm 1995, và trở thành một tiêu chuẩn ECMA năm 1997.

ECMA-262 là tên chính thức. ECMAScript 2016 (tháng 6 năm 2016) là phiên bản mới nhất của JavaScript.

**2.7.2 Tại sao phải dùng JavaScript**

**JavaScript là một trong 3 ngôn ngữ được tất cả các nhà phát triển web sử dụng:**

* HTML để xác định nội dung của các trang web.
* CSS để xác định bố cục của trang web.
* JavaScript để các chương trình của các trang web hoạt động.

**Ưu điểm của JavaScript:**

* Sự tương tác Server ít hơn.
* Phản hồi thông tin ngay lập tức tới khách truy cập.
* Khả năng tương tác với người dùng tăng lên.
* Giao diện phong phú hơn.

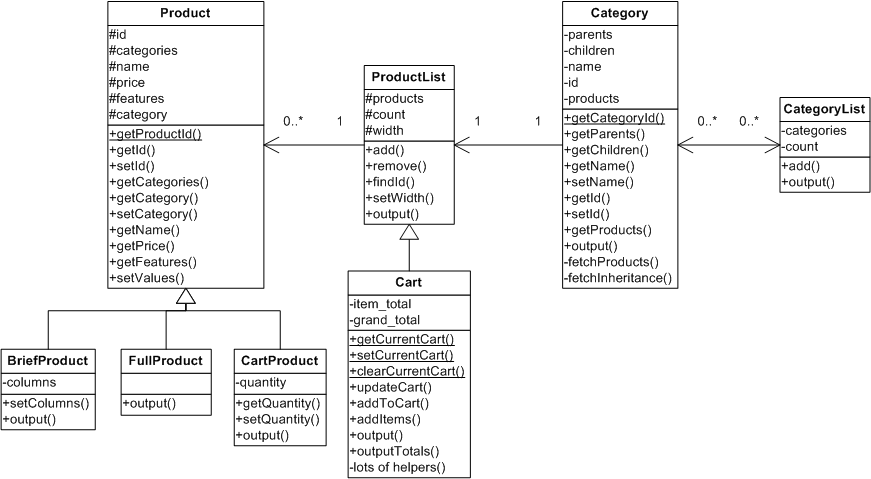
**Hạn chế của JavaScript**

JavaScript là một ngôn ngữ lập trình không đầy đủ. Nó thiếu các tính năng quan trọng sau đây:

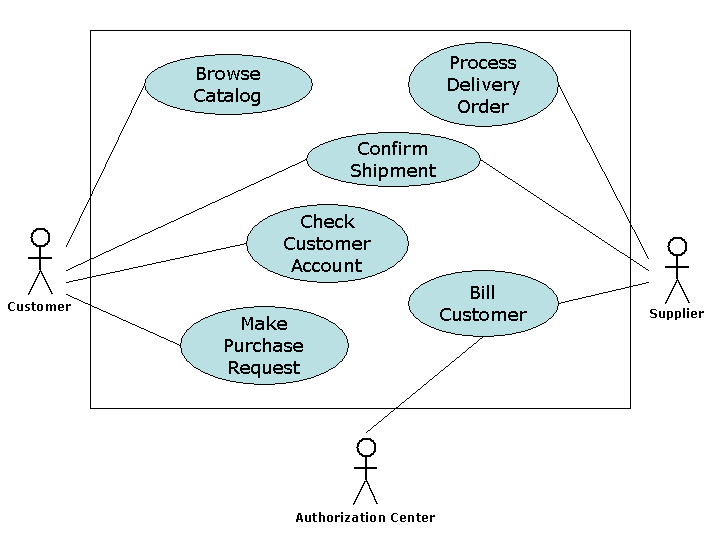
* Client-side JavaScript không cho phép đọc hoặc ghi các tập tin vì lý do bảo mật.
* JavaScript không được sử dụng cho các ứng dụng mạng, vì không hỗ trợ.
* JavaScript không hỗ trợ xử lý đa luồng hay đa xử lý.

**Chương 3: Sơ đồ phân tích thiết kế**

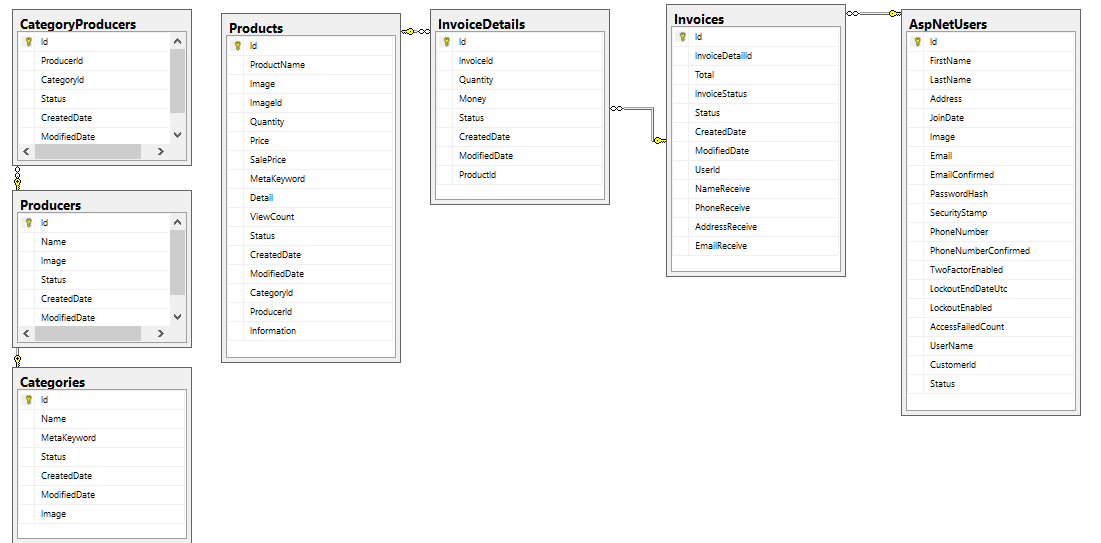
**3.1 Sơ đồ Class**



**3.2 Sơ đồ Usecase**



**3.3 Sơ đồ vật lý**

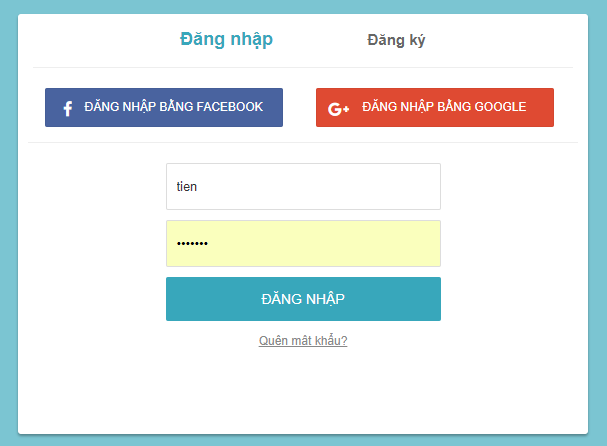


**Chương 4: Các tính năng cơ bản**

**4.1 Khách hàng**

**4.1.1 Đăng nhập**

Khách hàng có thể đăng nhập bằng tài khoản đã được đăng ký của mình.



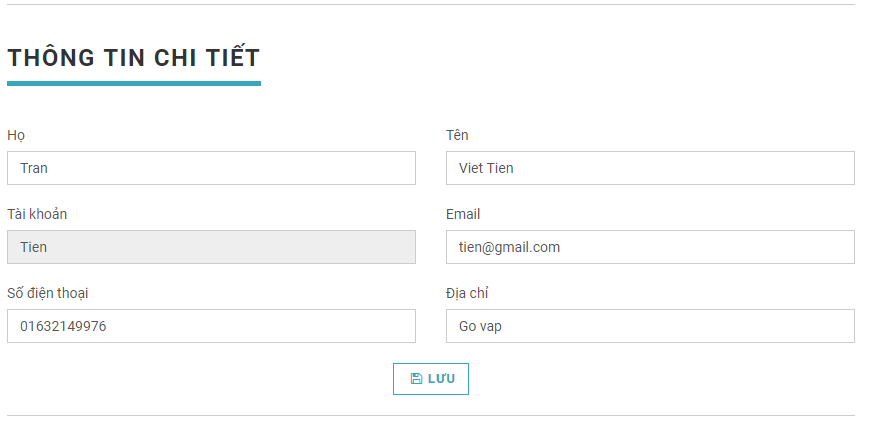
**4.1.2 Đăng ký**

Khách hàng dễ dàng đăng ký bằng một số thông tin cá nhân cơ bản.



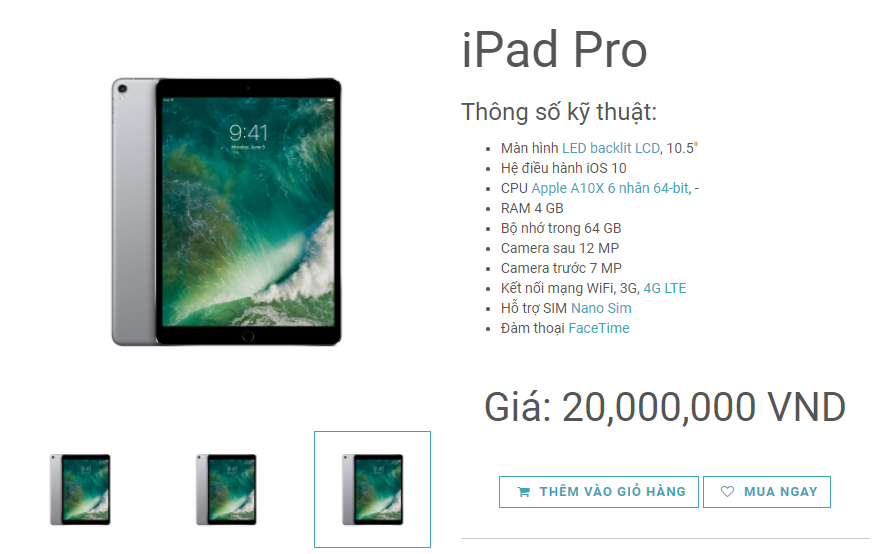
**4.1.3 Chỉnh sửa thông tin cá nhân**

Khách hàng có thể tự chỉnh sửa thông tin cá nhân của mình.



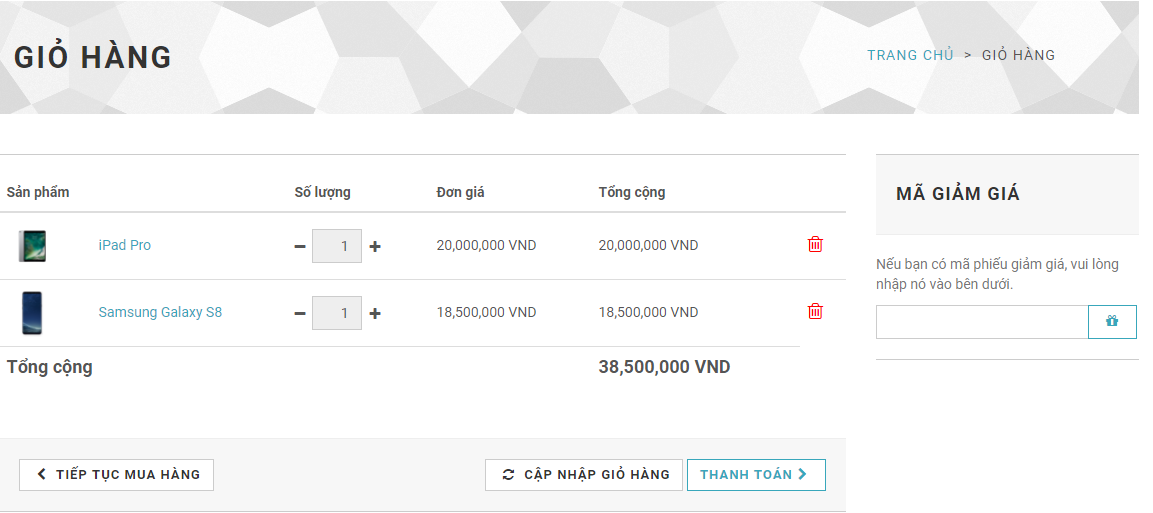
**4.1.4 Xem sản phẩm**

Khách hàng có thể xem thông tin sản phẩm như tên, giá, thông số kỹ thuật, cấu hình,…



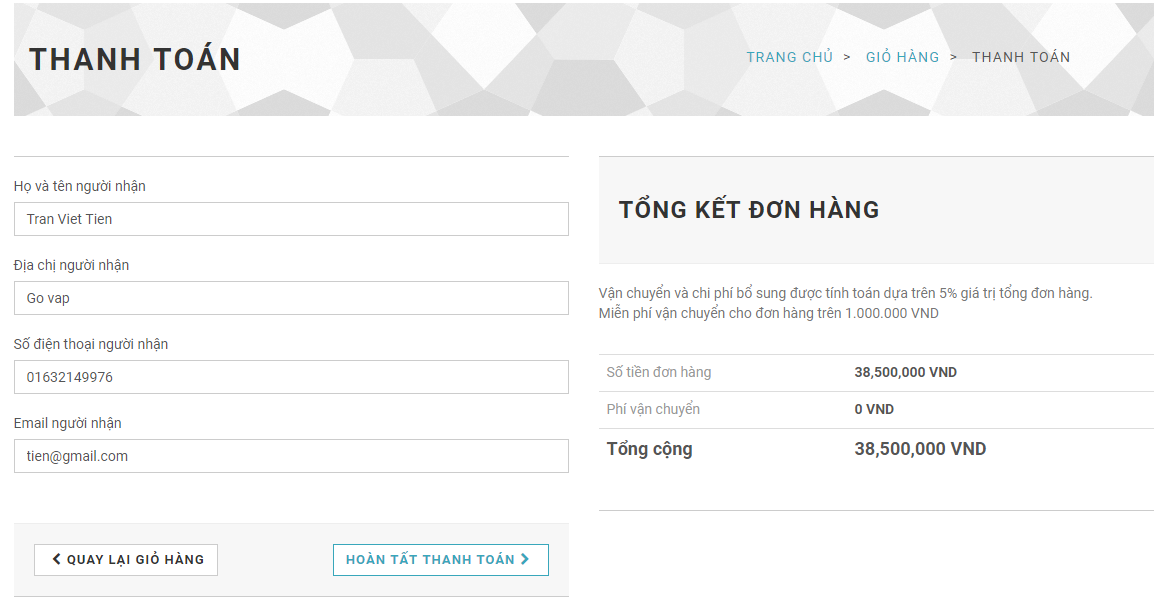
**4.1.5 Giỏ hàng**

**`** Khách hàng có thể kiểm tra lại giỏ hàng trước khi thanh toán.



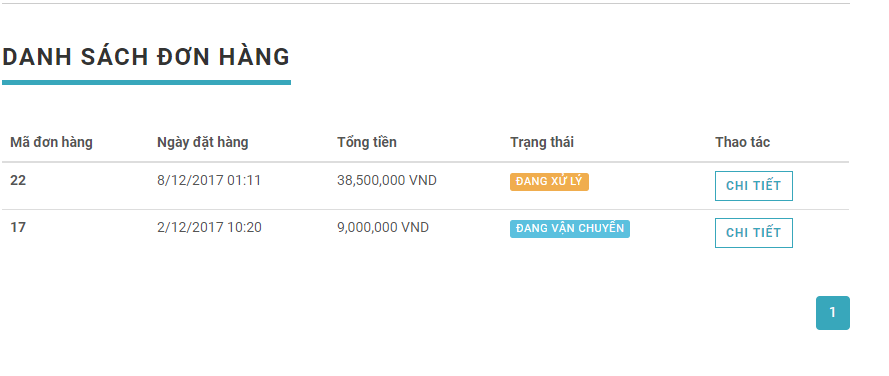
**4.1.6 Thanh toán**

Khách hàng chỉ cần một vài thông tin cơ bản là có thể tiến hàng thanh toán.



**4.1.7 Quản lý đơn hàng**

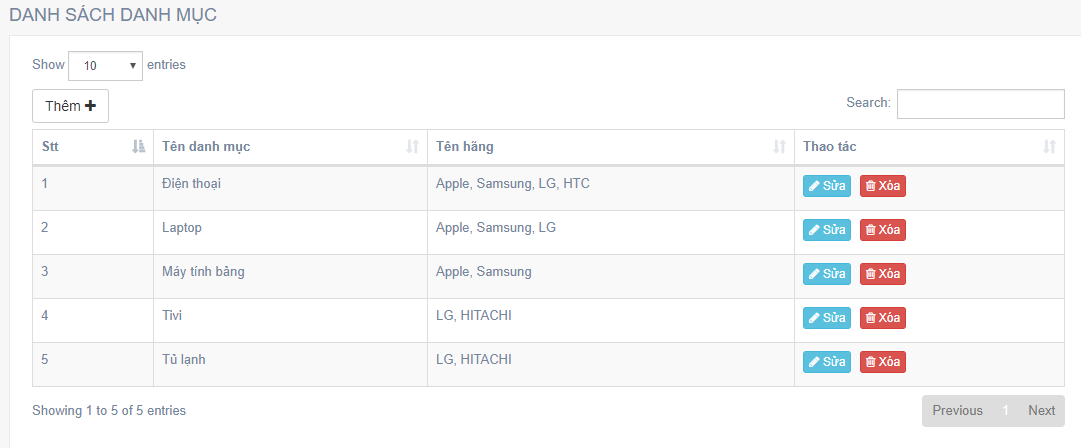
Khách hàng có thể xem lại tình trạng đơn hàng của mình sau khi thanh toán.



**4.2 Quản trị viên**

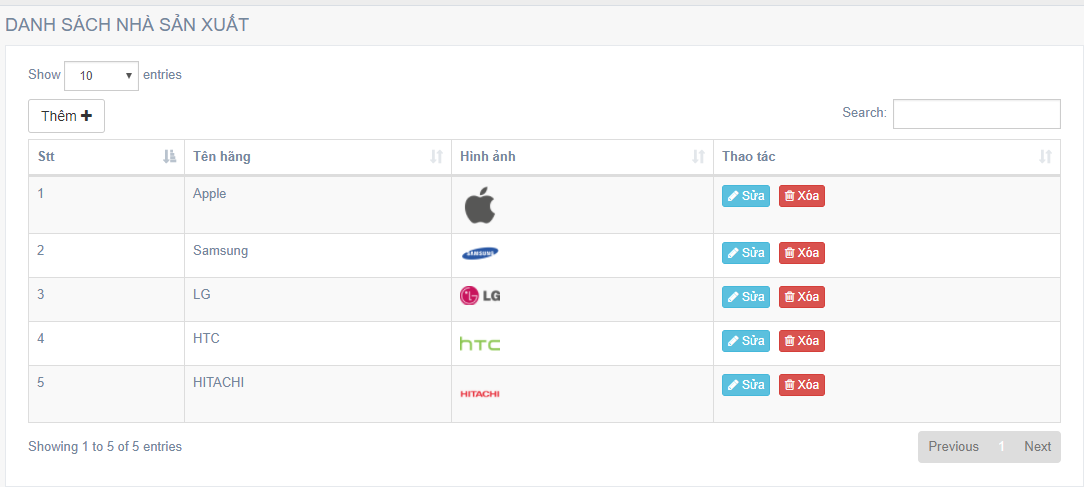
**4.2.1 Quản lý danh mục sản phẩm**

Quản trị viên có thể thêm mới, chỉnh sửa, hoặc xóa những danh mục mà hệ thống đang bán.



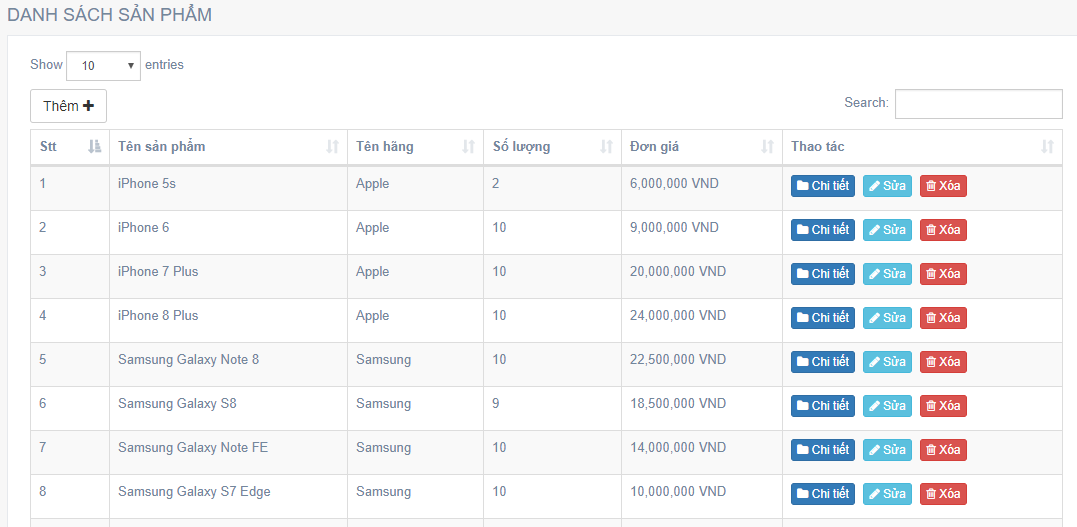
**4.2.2 Quản lý hãng sản xuất**

Quản trị viên có thể thêm mới, chỉnh sửa, hoặc xóa những hãng sản xuất đang có trong hệ thống.



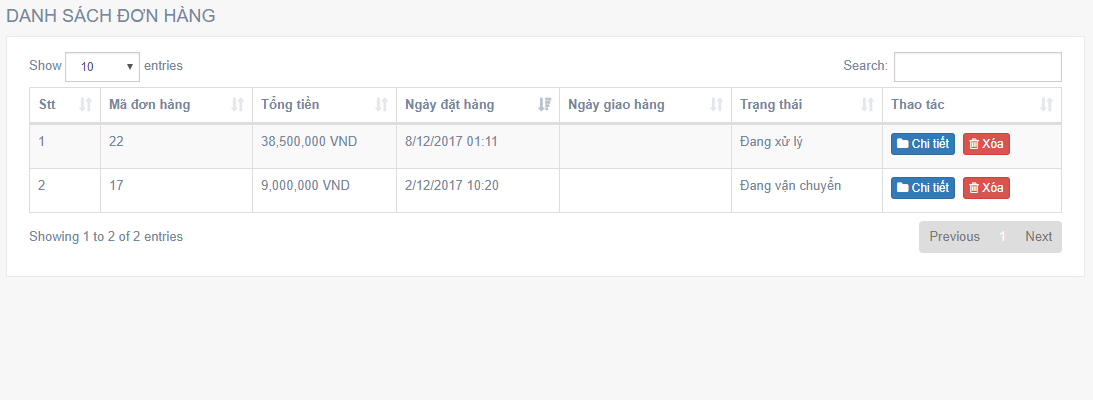
**4.2.3 Quản lý sản phẩm**

Quản trị viên có thể thêm mới, chỉnh sửa, hoặc xóa những sản phẩm đang có trong hệ thống.



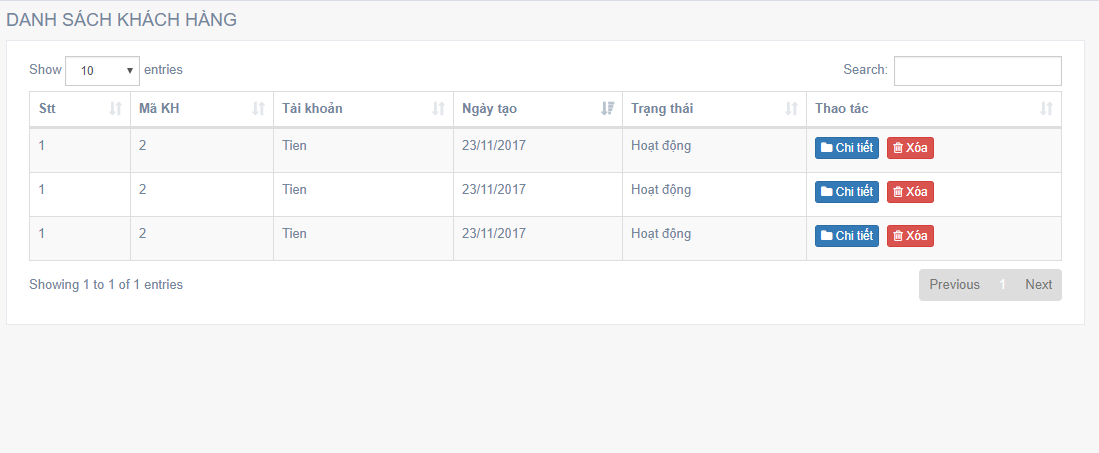
**4.2.4 Quản lý đơn hàng**

Quản trị viên có thể cập nhập, chỉnh sửa trạng thái của đơn hàng, cũng như có thể hủy nếu đơn hàng không hợp lệ.



**4.2.4 Quản lý khách hàng**

Quản trị viên có thể xem thông tin chi tiết khách hàng, khóa tài khoản của khách hàng cũng như là xóa tài khoản của khách hàng.



**Chương 5: Kết luận**

**5.1 Những kết quả đạt được**

Tiểu luận đã phần nào tạo nên một website bán hàng online với những tính năng cơ bản. Đã tích hợp được công nghệ ASP.Net MVC một cách hiệu quả, bên cạnh đó kết hợp ASP.Net Identity giúp website được bảo mật hơn.

Kết hợp được những công nghệ mới như html 5, css 3, jquery để giúp web site linh động hơn.

Giao diện website đơn giản, rõ ràng, giúp cho người dùng dễ dàng thao tác.

**5.2 Những điểm cần khắc phục**

Vì tiểu luận được thực hiện trong 3 tháng nên hệ thống vẫn còn thiếu nhiều chức năng, chỉ áp dụng hệ thống cho những mô hình kinh doanh nhỏ. Tốc độ tải trang còn chậm, cần tối ưu hóa hơn.

Bên cạnh đó cần mở rộng tính năng thanh toán, như tích hợp thanh toán online,… Tích hợp thêm chat box để quản trị viên có thể chăm sóc khách hàng một cách dễ dàng.

**Tài liệu tham khảo**

**Online**

[1] <https://docs.microsoft.com/en-us/aspnet/mvc/overview/getting-started/introduction/getting-started>

[2] <https://msdn.microsoft.com/en-us/library/aa937723(v=vs.113).aspx>

[3] <https://msdn.microsoft.com/en-us/library/mt173608(v=vs.108).aspx>

[4] <https://jquery.com/>

[5] <https://datatables.net/>