**BỘ QUỐC PHÒNG**

**TRƯỜNG ĐẠI HỌC THÔNG TIN LIÊN LẠC**

**ĐỀ CƯƠNG**

**KHÓA LUẬN TỐT NGHIỆP**

***Đề tài:***

**NGHIÊN CỨU CÔNG NGHỆ ASP.NET ỨNG DỤNG XÂY DỰNG WEBSITE BÁN SÁCH THEO MÔ HÌNH MVC**

**GVHD: ThS. Phùng Bá Tâm**

**SINH VIÊN: Lê Thị Phương Thảo**

**LỚP: ĐHCN4B**

**KHOA: Công nghệ thông tin – TCKGM**

**KHÓA HỌC: 2017 - 2021**

**BỘ QUỐC PHÒNG**

**TRƯỜNG ĐẠI HỌC THÔNG TIN LIÊN LẠC**

**ĐỀ CƯƠNG**

**KHÓA LUẬN TỐT NGHIỆP**

***Đề tài:***

**NGHIÊN CỨU CÔNG NGHỆ ASP.NET ỨNG DỤNG XÂY DỰNG WEBSITE BÁN SÁCH THEO MÔ HÌNH MVC**

**GVHD: ThS. Phùng Bá Tâm**

**SINH VIÊN: Lê Thị Phương Thảo**

**LỚP: ĐHCN4B**

**KHOA: Công nghệ thông tin – TCKGM**

**KHÓA HỌC: 2017 - 2021**

# MỞ ĐẦU

## **Cơ sở khoa học và thực tiễn của đề tài**

Sự tiến bộ của khoa học, công nghệ cùng với sự hiện diện của mạng Internet đã tạo điều kiện cho sự ra đời các website tiện ích với người dùng. Trong những năm gần đây, website càng trở nên phổ biến. Cùng với nhu cầu tìm kiếm sách vở của con người ngày càng nhiều. Hình thức trao đổi buôn bán sách cũng ngày càng đa dạng.

Xuất phát từ đặc điểm trên, “Xây dựng website bán sách theo mô hình MVC” là một lợi thế trong thời đại 4.0 hiện nay. Để việc trao đổi mau bán sách một cách thuận tiện, đỡ tốn kém về mặt thời gian. Cũng như khách hàng sẽ nắm rõ thông tin từng loại sách, giá sách, các nhà sách tránh mua phải sách không rõ nguồn gốc. Hỗ trợ khách hàng tìm kiếm loại sách phù hợp với nhu cầu mình cần mà chỉ cần một vài lần click chuột trên máy tính là có thể mua được sách mình yêu thích. Bên cạnh đó việc xây dựng một hệ thống quản lý bán sách cũng như giúp người làm công tác quản lý cập nhật và thống kê hàng một cách nhanh chóng và chính xác.

## **Lý do chọn đề tài**

Bán hàng trực tuyến là một hình thức kinh doanh đang được các trang thương mại điệ tử sử dụng hiện nay, các sản phẩm hay dịch vụ đều được tích hợp trên một trang Web hoặc trên các ứng dụng của các thiết bị di động. Việc tận dụng ưu thế của mạng internet để tạo ra quá trình mua bán, trao đổi hàng hóa một cách thuận lợi và nhanh cóng. Bên cạnh đó đối với người bán hàng chỉ cần có một kênh thông tin bán hàng thông qua một trang web hoặc ứng dụng trên mạng internet là có thể tiếp cận khách hàng trong thời đại công nghệ số 4.0 hiện nay. Trong xu thế đó các Website bán hàng được coi như là một cửa hàng hiện hữu cụ thể trên một không gian số, mà qua đó chúng ta có thể trưng bày các sản phẩm ảo của mình thông qua các thông tin hiển thị, hình ảnh mô tả cụ thể về sản phẩm. Ngoài ra, Website còn tích hợp các chức năng giúp cho việc mua bán diễn ra một cách nhanh chóng và tiện lợi nhất. Xuất phát từ những lý do trên em chọn đề tài “Nghiên cứu công nghệ Asp.NET ứng dụng xây dựng website bán sách theo mô hình MVC” nhằm hỗ trợ cho việc mua bán sách một cách thuận lợi mà không phải tốn quá nhiều thời gian.

## **Phương pháp nghiên cứu**

Nghiên cứu lý thuyết về phân tích và thiết kế hệ thống

Nghiên cứu công nghệ Web, ASP.NET

Nghiên cứu xây dựng Website bán sách theo mô hình MVC

# LỜI MỞ ĐẦU

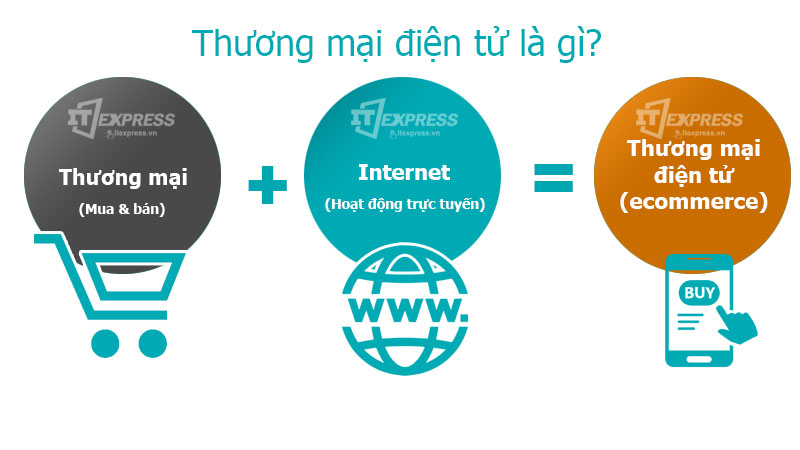
Ngày nay, CNTT đã và đang đóng vai trò quan trọng trong đời sống kinh tế, xã hội của nhiều quốc gia trên thế giới, là một phần không thể thiếu trong xã hội năng động, ngày càng hiện đại hoá. Vì vậy, việc tin học hoá vào một số lĩnh vực là hoàn toàn có thể và phù hợp với xu hướng hiện nay. Xuất phát từ nhu cầu thực tế đó, trong công việc mua và bán sách, việc quản lý sách nhập và bán là một việc không thể thiếu. Nhằm thay thế một số công việc mà trước đó phải thao tác bằng tay trên giấy tờ đạt hiệu quả không cao, mất nhiều thời gian. Vì vậy, em đã thực hiện báo cáo với đề tài “Xây dựng website bán sách theo mô hình MVC” Do trong khuôn khổ thời gian ngắn, trình độ chuyên môn, kinh nghiệm và kiến thức của bản thân còn hạn chế, nên chúng em rất mong được sự góp ý của cô và các bạn trong lớp, để đề tài nghiên cứu của chúng em ngày càng hoàn thiện hơn và được ứng dụng trong thực tế. Xin chần thành cảm ơn!

# Chương 1 CƠ SỞ LÝ THUYẾT

## **Thương mại điện tử**

Thương mại điện tử là hoạt động mua-bán sản phẩm (hoặc dịch vụ) thông qua các hệ thống điện tử như Internet và các mạng máy tính. Thương mại điện tử còn được gọi là E-commerce, nghĩa là Electronic Commerce.

Thế giới ngày càng bùng nổ về mạng máy tính, con người có thể nhanh chóng giao tiếp và kết nối với nhau dễ dàng hơn thông qua nhiều loại dich vụ internet. Đây chính là điều kiện thuận lợi cho lĩnh vực thương mại điện tử ngày càng phát triển hơn. Hiện nay, thương mại điện tử đã trở thành một phương tiện giao dịch quen thuộc của các công ty thương mại lớn trên thế giới. Thương mại điện tử có khả năng giúp ích rất nhiều cho những doanh nghiệp cả lớn lẫn nhỏ và người hưởng lợi nhất thường là khách hàng. Khách hàng sẽ mua được sản phẩm rẻ hơn, nhanh hơn, hiệu quả hơn và thuận lợi hơn, còn doanh nghiệp có thể đưa sản phẩm của mình đến với thị trường một cách nhanh nhất, bán hàng thuận lợi hơn.



Hình 1. 1 Thương mại điện tử

## **Các loại hình thương mại điện tử**

Ngày nay, thương mại điện tử phát triển mạnh mẽ bởi tốc độ sử dụng internet cùng với nhiều các công nghệ hiện đại ra đời. Con người ngày càng ưu thích giao dịch dưới hình thức này bởi những thuận lợi mà nó mang lại. Có rất nhiều hình thức thương mại điện tử khác nhau,dưới đây là một số mô hình thương mại điện tử điển hình.

### **Thương mại điện tử B2B ( Business To Business )**

Là thương mại điện tử giữa các công ty. Đây là loại hình thương mại điện tử gắn với mối quan hệ giữa các công ty với nhau.

* Nó giúp các doanh nghiệp tiết kiệm thời gian, tiền bạc và kinh doanh hiệu quả hơn. Mặt khác, giúp tăng cơ hội hợp tác giữa các doanh nghiệp với nhau.
* Điều chỉnh theo nhu cầu, mục đích sử dụng của khách hàng nhanh chóng hơn.
* Mọi hoạt động giao dịch diễn ra qua Internet tiện lợi, giúp doanh nghiệp giảm chi phí đi lại gặp gỡ đối tác cung ứng.
* Nhờ có B2B, doanh nghiệp có nhiều cơ hội lựa chọn nhà cung ứng tốt cũng như giá thành đầu vào rẻ hơn.



Hình 1. 2 Thương mại điện tử B2B

### **Thương mại điện tử B2C ( Business to Customers)**

Thương mại điện tử B2C là hình thức kinh doanh từ doanh nghiệp đến khách hàng. Chi cần kết nối mạng Internet, các cá nhân sẽ mua hàng phục vụ mục đích tiêu dùng bình thường.

* Khách hàng của mô hình B2C chủ yếu là người tiêu dùng cá nhân. Họ có nhu cầu mua sắm trên Internet cho mục đích của mình, nên không phát sinh thêm giao dịch mua bán.
* Doanh nghiệp tiết kiệm chi phí bán hàng, không mất tiền thuê mặt bằng, người bán hàng,...
* Tạo điều kiện cho doanh nghiệp tiếp xúc với nhiều lượng khách hàng trên khắp cả nước
* Người tiêu dùng có nhiều lựa chọn để mua hàng, không mất thời gian đi lại.



Hình 1. 3 Thương mại điện tử B2C

### **Thương mại điện tử C2C**

Thương mại điện tử khách hàng tới khách hàng C2C đơn giản là thương mại giữa các cá nhân và người tiêu dùng.

Loại hình thương mại điện tử này được phân loại bởi sự tăng trưởng của thị trường điện tử và đấu giá trên mạng, đặc biệt với các ngành theo trục dọc nơi các công ty/ doanh nghiệp có thể đấu thầu cho những cái họ muốn từ các nhà cung cấp khác nhau. Có lẽ đây là tiềm năng lớn nhất cho việc phát triển các thị trường mới.



Hình 1. 4 Thương mại điện tử C2C

### **Thương mại điện tử B2G**

Thương mại điện tử giữa doanh nghiệp với chính phủ (B2G) được định nghĩa chung là thương mại giữa công ty và khối hành chính công. Nó bao hàm việc sử dụng Internet cho mua bán công, thủ tục cấp phép và các hoạt động khác liên quan tới chính phủ.

Các chính sách mua bán trên web tăng cường tính minh bạch của quá trình mua hàng (và giảm rủi ro của việc không đúng quy cách). Tuy nhiên, tới nay, kích cỡ của thị trường thương mại điện tử B2G như là một thành tố của của tổng thương mại điện tử thì không đáng kể, khi mà hệ thống mua bán của chính phủ còn chưa phát triển.



Hình 1. 5 Thương mại điện tử B2G

### **Lợi ích của thương mại điện tử**

* Thương mại điện tử đã đưa lại những lợi ích tiềm năng thể hiện ở một số mặt như sau:
* Giúp người tham gia thu nhập thông tin phong phú
* Giảm chi phí bán hàng và tiếp thị
* Giảm chi phí giao dịch
* Giúp thiết lập và củng cố quan hệ quốc tế
* Tạo điều kiện sớm tiếp cận “Kinh tế số hóa”.

## **Tổng quan về Asp.Net MVC**

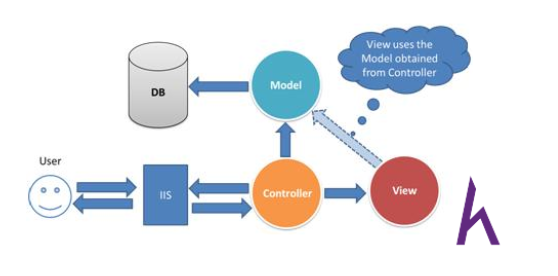
ASP.NET MVC là một framework web được phát triển bởi Microsoft, thực thi mô hình MVC (model–view–controller). ASP.NET MVC là 1 phần mềm mã mở, tách rời với thành phần độc quyền ASP.NET Web Forms. ASP.NET MVC đang nổi lên là phương pháp phát triển web mạnh nhất và phổ biến nhất trên nền ASP.NET hiện nay.

1. Ưu điểm mà ASP.Net MVC:

* Giúp tối ưu ứng dụng và dễ dàng trong việc viết code, giao diện
* Giao diện trong ASP.Net MVCsử dụng công nghệ thiết kế web HTML, CSS nền việc thiết kế giao diện trở nên dễ dàng và giúp cho designer linh hoạt trong việc thiết kế
* ASP.Net MVC không sử dụng view state vì vậy trang web không bị tăng kích thước do đó hiệu năng hoạt động không bị bị giảm.
* 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, dễ dàng tạo ra mã SEO URL để thu hút lượng truy cập ứng dụng bất kỳ.

##### Nhược điểm mà ASP.Net MVC:

* Không thích hợp việc phát ứng dụng nhỏ vì mô hình yêu cầu lưu trữ số lượng lớn các file
* Không thể chạy trên Linux
* Không có Visual Studio thì khó có thể viết code cho Asp.Net



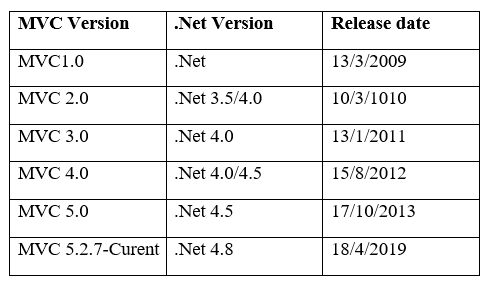
Hình 1. 6 Cơ chế hoạt động của asp.net

Dựa vào hình mô phỏng phía trên ta khái quá sơ qua cơ chế hoạt động của ASP.Net MVC

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

### **sử phát triển Asp.Net MVC**

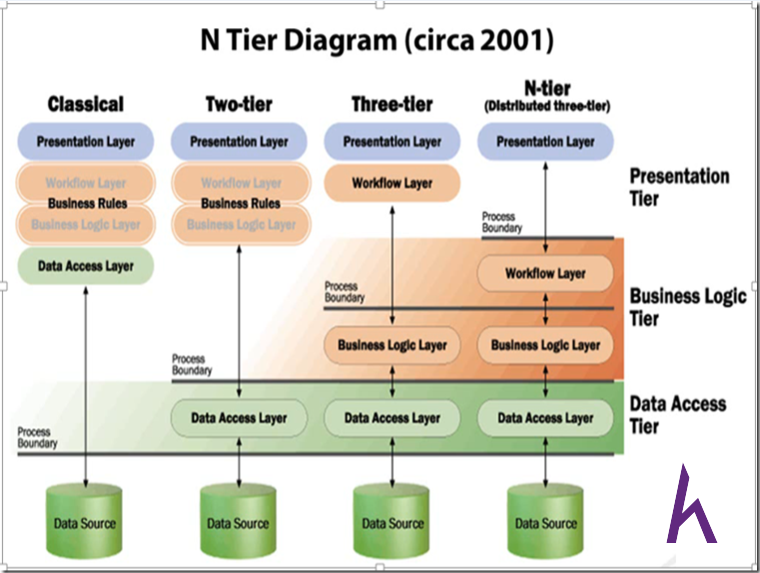
ASP.NET bắt nguồn từ Microsoft và mục đích của nó là nhằm giúp cho các chuyên gia phát triển một công cụ lập trình, tạo nên các website cũng như ứng dụng trên nền tảng Web. Thực chất, sự phát triển của ngôn ngữ này dựa vào thành công của người “tiền nhiệm” ASP. Người lập trình ASP.NET cũng có thể dựa vào bất kỳ ngôn ngữ .NET nào để làm việc. Ngoài ra, việc Microsoft luôn thống trị trong lĩnh vực trình duyệt web với Internet Explorer cũng là một yếu tố thúc đẩy ASP. NET ngày càng vững mạnh hơn. Dưới đây là bảng tóm tắt lại quá trình phát triển của Asp.net:

****

Bảng 1. 1 Lịch sử phát triển ASp.Net

**1.3.2 Tổng quát về Asp.Net MVC**

ASP.NET MVC là một framework phong phú để xây dựng các ứng dụng web và API bằng cách sử dụng mẫu thiết kế Model-View-Controller (MVC).

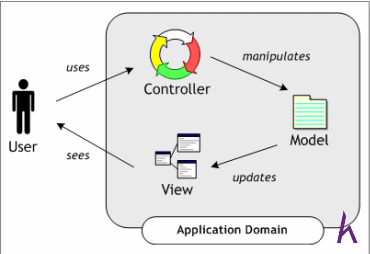


Hình 1. 7 Tổng quát về ASP.net mô hình MVC

Trong lập trình web chia ra 3 tầng như sau:

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

Minh họa mô hình MVC:



Hình 1. 8 Mô hình MVC

* Model: Là Data Access Layer và tầng Business Logic Layer. Hai tầng này là hai tầng tương đương với tầng model trong mô hình MVC.
* View: là tầng giao diện, hiển thị dữ liệu được truy xuất từ tầng model. Tầng này tương đương với tầng Presentation Layer trong cấu trúc Three – Tier.
* Controller: Đây là tầng giúp kết nối giữa tầng model, có nghĩa là nếu phía client yêu cầu hiển thị dữ liệu thì controller gọi giữ liệu từ model và trả về cho view vì view tương tác trực tiếp với client.

**1.3.3 Lợi ích của Asp.Net MVC**

Nền tảng ASP.NET mang lại lợi ích như sau:

* Dễ quản lý sự phức tạp của ứng dụng bằng cách chia ứnng dụng thành ba phần:

Model

View

Controller

* Nókhông sử dụng view state hoặc sever-based form. Điều này không tốt cho lập trình viên muốn quản lý hết các khía cạnh của một ứng dụng.
* Nó sử dụng Font Controller, mẫu này giúp quản lý các request chỉ thông qua một Controller. Nhờ đó có thể thiết kế một hạ tầng quản lý định tuyến.
* Hỗ trợ tốt hơn cho mô hình phát triển ứng dụng.
* Hỗ trợ tốt cho các ứng dụng được xây dựng bởi những đội có nhiều lập trình viên và thiết kế mà vaanc quản lý đụpcư tính năng của ứng dụng.

### **Tính năng của Asp.Net MVC**

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.MVC là một nền tảng khả mở rộng (extensible) & khả nhúng (pluggable). Các thành phần của ASP.NET MVC được thiết kế để chúng có thể được thay thế một cách dễ dàng hoặc dễ dàng tùy chỉnh.

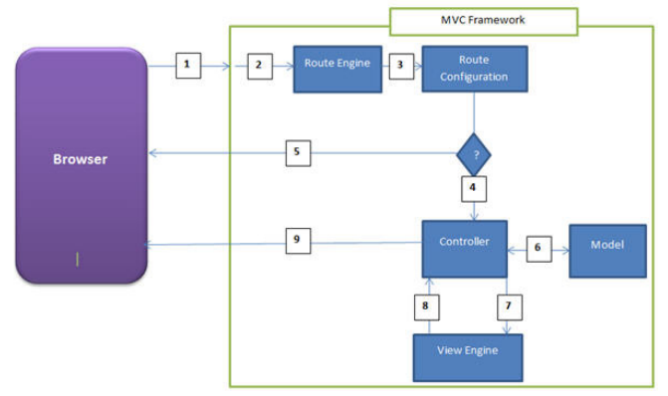
* ASP.NET MVC có thành phần ánh xạ URL mạnh mẽ cho phép bạn xây dựng những ứng dụng có các địa chỉ URL xúc tích và dễ tìm kiếm.
* Hỗ trợ sử dụng đặc tả (các thẻ) của các trang ASP.NET(.aspx), điều khiển người dùng (.ascx) và trang master page (marter).
* Hỗ trợ các tính năng có sẵn của ASP.NET như cơ chế xác thực người dùng, quản lý thành viên, quyền, output caching và data caching,..
* 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.
* ASP.NetMVC5 hỗ trợ chứng thực qua các API khác(facebook,google+..) bootstrap được thêm vào hỗ trợ phần thiết kế giao diện.

**1.3.5 Giao tiếp trong kiến trúc Asp.Net MVC**

Các thành phần cơ bản của ứng dụng Asp.Net MVC bao gồm :

* MVC Framework
* Route engine
* Route configuration
* Control
* Model
* View engine
* View

Các thành phần này giao tiếp với nhau nhằm xử lý các yêu cầu của ứng dụng Asp.Net MVC. Quá trình xử lý yêu cầu thường liên quan tới một chuỗi các xử lý, mỗi xử lý sẽ được một component trong Asp.net đảm nhiệm.



Hình 1. 9 Thành phần giao tiếp của ASp.Net

1. Trình duyệt gửi yêu cầu của ứng dụng Asp.Net MVC
2. MVC Engine chuyển yêu cầu tới cho Routing engine (Bộ điều hướng)
3. Routing engine kiểm tra cấu hình điều hướng (route configuration) của ứng dụng nhằm chuyển đến các controller phù hợp với yêu cầu.
4. Khi Controller được tìm thấy, control này sẽ được thực thi
5. Nếu Controller không được tìm thấy, bộ điều hướng sẽ chỉ ra rằng controller không được tìm thấy và MVC Engine sẽ thông báo lỗi cho trình duyệt.
6. Controller giao tiếp với model, tầng Model đại diện cho các Entity bên trong hệ thống, Controller làm việc với tầng model để lấy thông tin cần cho người dùng
7. Controller yêu cầu View engine hiển thị thông tin dựa trên dữ liệu của tầng model
8. View engine trả kết quả cho controller
9. Controller gửi kết quả cho trình duyệt thông qua giao thức Http.

### **Sự khác nhau giữa ASP.NET MVC với web Forms**

ASP.net WebForm sử dụng ViewState để quản lý, các trang ASP.net đều có lifecycle, postback và dùng các web controls, các event để thực hiện các hành động cho UI (User Interface) khi có sự tương tác với người dùng nên hầu hết ASP.net WebForm xử lý chậm.

-  ASP.net MVC chia ra làm 3 phần: Models, View, Controller. Mọi tương tác của người dùng với Views sẽ được thực hiện hành động trong Controllers, không còn postback, lifecycel và events.

-  Việc kiểm tra (test), gỡ lỗi (debug) với ASP.net WebForm đều phải chạy tất cả các tiến trình của ASP.net, và sự thay đổi ID của bất kỳ Controls nào cũng ảnh hưởng đến ứng dụng. Đối với MVC thì việc đó có thể sử dụng các unit  test có thể thẩm định rất dễ dàng các Controllers thực hiện như thế nào.

|  |  |  |
| --- | --- | --- |
| **Các tính năng** | **ASP.net WebForm** | **ASP.net MVC** |
| Kiến trúc chương trình | Kiến trúc mô hình WebForm –> Bussiness –> Database | Kiến trúc sử dụng việc phân chia chương trình thành: Models, Views, Controllers |
| Cú pháp chương trình | Sử dụng cú pháp của WebForm, tất cả các sự kiện và controls do server quản lý | Các sự kiện được điều khiển bởi controllers, các controls không do server quản lý. |
| Truy cập dữ liệu | Sử dụng hầu hết các công nghệ truy cập dữ liệu trong ứng dụng | Phần lớn dùng LINQ và SQL class để tạo mô hình truy cập đối tượng. |
| Debug | Debug phải thực hiện tất cả bao gồm các lớp truy cập dữ liệu, sự hiển thị, điều khiển các controls. | Debug có thể sử dụng các unit test để kiểm tra các phương thức trong controllers |
| Tốc độ phân tải | Tốc độ phân tải chậm khi trong trang có quá nhiều các controls vì ViewState quá lớn | Phân tải nhanh hơn do không phải quản lý ViewState để quản lý các controls trong trang. |
| Tương tác với JavaScript | Tương tác với JavaScript khó khăn vì các controls được điều khiển bởi server | Tương tác với JavaScript dễ dàng vì các đối tượng không do server quản lý điều khiển không khó |
| URL address | Cấu trúc địa chỉ URL có dạng:  <filename>.aspx?&<các tham số> | Cấu trúc địa chỉ rành mạch theo dạng Controllers/Action/ID |

Bảng 1. 2 So sánh Asp.net và Web Forms

**1.4 Ngôn ngữ Web**

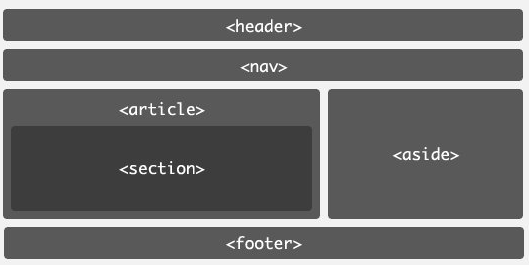
**1.4.1 HTML5**

**HTML (Hypertext Markup Language)** là một nền tảng tương tự như **Microsoft Word** giúp người dùng **thiết kế** thành phần trong **website**, cấu trúc các trang, chuyên mục hoặc các thiết kế các ứng dụng…

HTML5: Tương thích với các trang web đang tồn tại

* Chuẩn hóa các kỹ thuật không chính thức
* Tăng khả năng phục vụ đa phương tiện.

Cấu trúc cơ bản của trang web sử dụng HTML5:



Hình 1. 10 Cấu trúc cơ bản HTML5

Ví dụ:

<html>

<head>ThaoDHCN4</head>

<body>

<header>...</header>

<nav>...</nav>

<article>

<section>

...

</section>

</article>

<aside>...</aside>

<footer>...</footer>

</body>

</html>

**1.4.2 CSS3**

**CSS** là ngôn ngữ tạo phong cách cho trang web – Cascading Style Sheet language. Nó dùng để tạo phong cách và định kiểu cho những yếu tố được viết dưới dạng ngôn ngữ đánh dấu, như là [HTML](https://www.hostinger.vn/huong-dan/html-la-gi/). **CSS3** đóng vai trò quan trọng trong việc định hình **Web Responsive** nhờ có **Bootstrap**.

* CSS hoạt động bằng cách khoanh vùng chọn dựa vào tên một thẻ HTML,ID hay class.
* Có thể hiểu CSS đóng vai trò như một công cụ giúp chúng ta thêm vào những thay đổi về mặt hình thức như đổi bố cục, màu sắc, font chữ,..
* Nếu một website không có CSS thì đó sẽ chỉ đơn thuần là một trang chứa văn bản với 2 màu chủ đạo là trắng và đen.
* Điểm giống và khác nhau giữa CSS và CSS3

|  |  |
| --- | --- |
| **Giống** | **Khác** |
| Đều là phiên bản được thừa hưởng tất cả những gì đã có . | Sự khác biệt lớn nhất của CSS3 là Modun.Modun cho phép các thuộc tính hoàn thnahf và phê duyệt nhanh hơn vì các phân đoạn được hoàn thành và phê duyệt theo từng khối. |

**Ví dụ:**

<div class= “multiple-backgrounds”>

<p></p>

<p>Lập trình asp.net mvc</p>

</div>

P{

Background: green;

Padding: 15px;

Color: black;

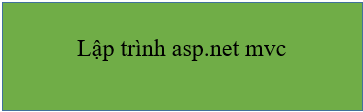
}

P:empty{

Display: none;

}

Kết quả:



**1.4.3 JQUERY**

JQuery là thư viện được viết từ JavaScript, jQuery giúp xây dựng các chức năng bằng Javascript dễ dàng, nhanh và giàu tính năng hơn. Nó thực hiện những nhiệm vụ như HTML document traversal và manipulation, animation, event handling, AJAX một cách rất đơn giản bằng một API dễ sử dụng và có khả năng hoạt động trên nhiều loại trình duyệt khác nhau.

Các module phổ biến của jQuery bao gồm:

* Ajax – xử lý Ajax
* Atributes – Xử lý các thuộc tính của đối tượng HTML
* Effect – xử lý hiệu ứng
* Event – xử lý sự kiện
* Form – xử lý sự kiện liên quan tới form
* DOM – xử lý Data Object Model
* Selector– xử lý luồng lách giữa các đối tượng HTML
* Ưu điểm của jQuery
* jQuery xử lý code rất nhanh và có khả năng mở rộng.
* jQuery tạo điều kiện cho người dùng viết các mã chức năng bằng các dòng tối thiểu.
* jQuery cải thiện hiệu suất lập trình web.
* jQuery phát triển các ứng dụng có tương thích với trình duyệt.
* Hầu hết các tính năng mới của trình duyệt mới đều được jQuery sử dụng.

Ví dụ về Jquery:



**1.4.4 BOOSTRAP**

Bootstrap là một framework bao gồm các HTML, CSS và JavaScript template dùng để phát triển website chuẩn responsive.

Bootstrap cho phép quá trình thiết kế website diễn ra nhanh chóng và dễ dàng hơn dựa trên những thành tố cơ bản sẵn có như typography, forms, buttons, tables, grids, navigation, image carousels…

Bootstrap là một bộ sưu tập miễn phí của các mã nguồn mở và công cụ dùng để tạo ra một mẫu webiste hoàn chỉnh. Với các thuộc tính về giao diện được quy định sẵn như kích thước, màu sắc, độ cao, độ rộng…, các designer có thể sáng tạo nhiều sản phẩm mới mẻ nhưng vẫn tiết kiệm thời gian khi làm việc với framework này trong quá trình thiết kế giao diện website.

Ví dụ:

|  |
| --- |
| <div class="container"> |
| <div class="row"> |

|  |
| --- |
|  |
| <div class="col-md-12"> |

|  |
| --- |
| <h2>@ViewBag.title</h2> |
| <a href="<https://phuongthao123.com>" class="btn btn- success">https://phuongthao123.com</a> |

|  |
| --- |
| </div> |
| </div> |

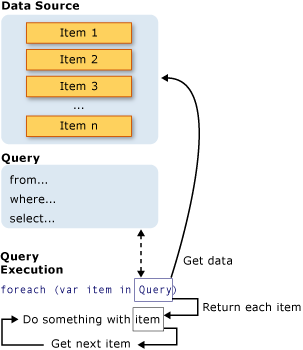
|  |
| --- |
| </div> |

**1.4.5 LINQ**

**LINQ - Language Integrated Query (ngôn ngữ truy vấn tích hợp) - là cách thức truy vấn dữ liệu từ một tập hợp dữ liệu.** Là một tập mở rộng ngôn ngữ cho phép thực hiện các truy vấn trong ngôn ngữ C# 2008 và VisualBasic 2008. LINQ cho phép Select (chọn), Filter (lọc), Sort (phân loại), Group (nhóm) và tranfom data (chuyển dữ liệu) từ các nguồn data source (dữ liệu nguồn) khác nhau theo một cách chung.

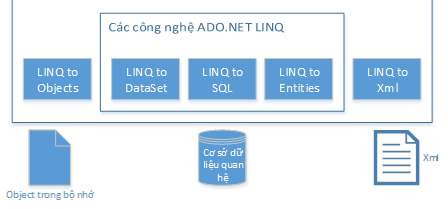
### Thành phần cấu thành LINQ

* Nguồn dữ liệu.
* Tạo câu lệnh truy vấn.
* Thực thi truy vấn để lấy kết quả.



Hình 1. 11 Thành phần LINQ

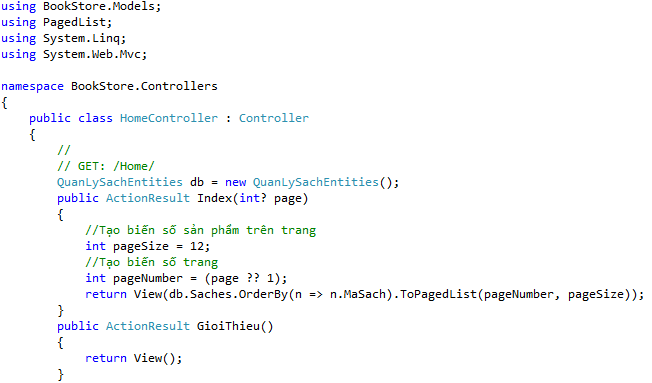
Các nguồn dữ liệu hỗ trợ LINQ:



Hình 1. 12 Nguồn dữ liệu hỗ trợ LINQ

* LINQ to Objects thực hiện truy vấn các đối tượng.
* LINQ to DataSet thực hiện truy vấn DataSet.
* LINQ to SQL thực hiện truy vấn đến cơ sở dữ liệu SqlServer mà không phải viết code.
* LINQ to XML đọc dữ liệu từ XML.

Ví dụ về lệnh LINQ:



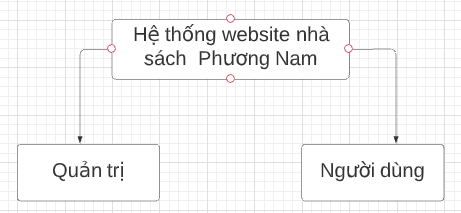
# Chương 2 PHÂN TÍCH THIẾT KẾ

## **Khảo sát hệ thống**

Xã hội ngày càng phát triển, nhu cầu hiểu biết trên tất cả các lĩnh vực của con người ngày càng được nâng cao. Đáp úng nhu cầu này, ở Việt Nam nhiều nhà sách xuát hiện, nhưng đa số các của hàng sách hiện nay đều tổ chức kinh doanh theo phương thức truyền thống :

Nhập sách từ các nhà cung cấp: Lập đơn đặt mua, gửi đơn đặt mua, nhận sách từ nhà cung cấp,thanh toán, trưng bày sách, thanh toán theo phương thức truyền thống… Chính phương thức kinh doanh này đã trực tiếp dẫn đến những khó khăn cho cửa hàng như sau: Tốn diện tích rất lớn cho việc trưng bày sách. Khách hàng họ cũng mất khá nhiều thời gian tìm kiếm sách. Việc quảng cáo cửa hàng gặp khó khăn và tốn kém. Những khó khăn trên trực tiếp làm giảm đáng kể doanh thu của cửa hàng. Giải pháp áp dụng công nghệ vào quản lý, thương mại hiện nay là một xu thế tất yếu. Một hệ thống quản lý bán sách qua mạng sẽ giúp cho cửa hàng giải quyết được các khó khăn trên và điều quan trọng là đáp ứng nhu cầu đông đảo khách hàng có nhu cầu tra cứu sách, đặt mua sách… và đây cũng là phương tiện quảng cáo sách cho cửa hàng một cách hiệu quả nhất. Với hệ thống này, nhà sách cũng tạo thu thập ý kiến từ phía khách hàng để giải quyết các sự cố cũng như cải thiện khả năng phục vụ khách hàng.

* 1. **Phân tích hệ thống về mặt chức năng**
     1. Sơ đồ chức năng tổng quát

****

Hình 2. 1 Sơ đồ tổng quát

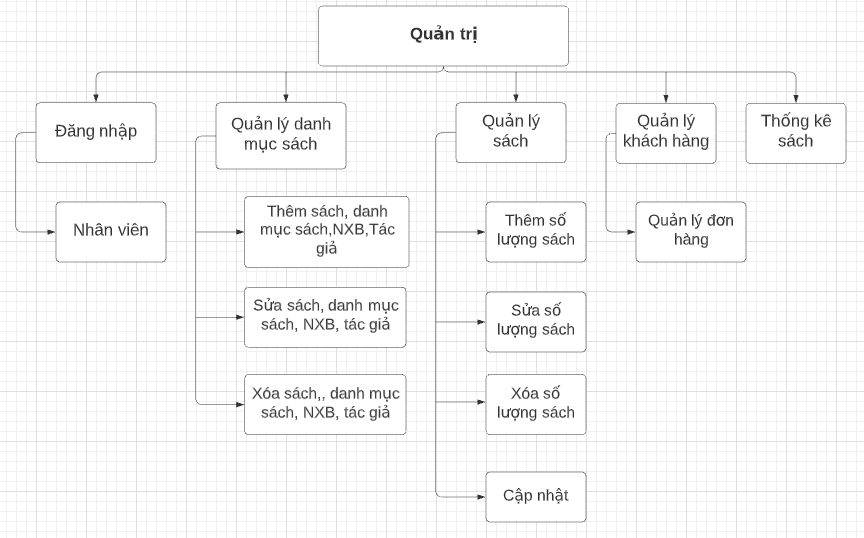
Phần quản trị:

Quản lý danh mục sách: bao gồm các công việc thêm, sửa, xóa các thông tin về loại sách, danh mục sách, nhà xuất bản, tác giả.

Phần người dùng:

Khách hàng có thể đăng ký để tạo tài khoản riêng của mình trên Website. Sau khi đăng kí, khách hàng có thể vào xem một số thông tin của nhà sách như: các thể loại sách, nhà xuất bản,.. cũng như đặt mua các loại sách.

1. **Sơ đồ chức năng của quản trị**

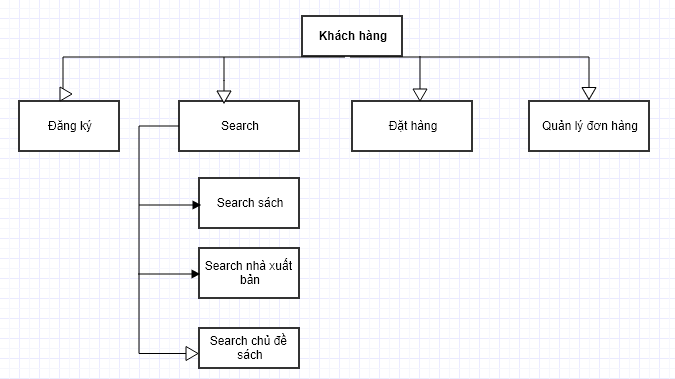


Hình 2. 2 Sơ đồ chức năng quản trị

Khi quản trị viên đăng nhập vào hệ thống: Nhân viên sẽ Sẽ quản lý được danh mục các loại sách,quản lý về số lượng sách được bán ra, quản lý khách hàng và thống kê lại số sách.

* Quản lý danh mục sách: khi quản trị viên xem danh mục sách thì hệ thống sẽ hiển thị ra mục thêm/sửa/xóa sách, danh mục sách, nhà xuất bản, tác giả.
* Quản lý sách: bao gồm các công việc thêm, xóa, sửa thông tin về quyển sách, cập nhật số lượng sách mới
* Hệ thống quản lý khách hàng khi đăng kí vào hệ thống và xem được lịch sử khách hàng đó mua các loại sách nào
* Hệ thống sẽ thống kê lại toàn bộ các sách

1. **Sơ đồ chức năng của khách hàng**



Hình 2. 3 Sơ đồ chức năng của khách hàng

Khi khách hàng đã đăng kí xong tài khoản của mình. Khách hàng muốn xem các loại sách mà nhà sách hiện đang có như sau:

Tìm kiếm: Tìm kiếm các chủ đề sách, tên sách, các nhà xuất bản

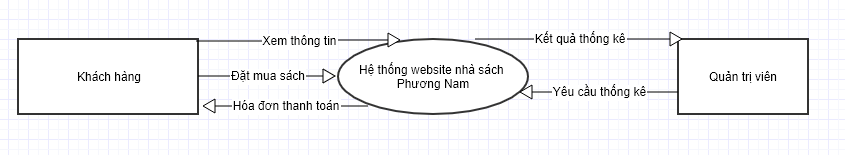
Sau khi tìm kiếm khách hàng có thể xem thông tin chi tiết giới thiệu về quyển sách.

Khách hàng có thể cho quyến sách vào giỏ hàng bằng cách đặt mua và có thể xem thêm thông tin của các quyển sách khác.

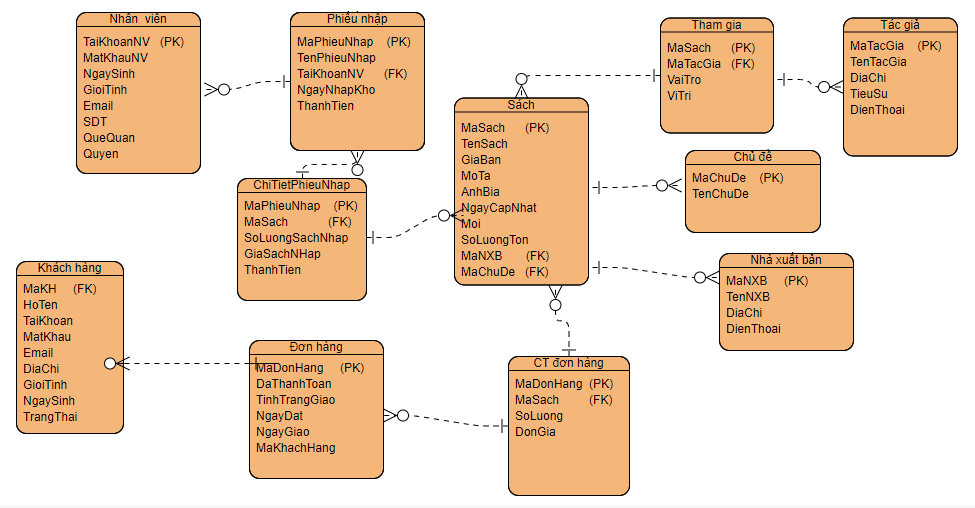
Sau khi lựa chọn được giỏ hàng khách hàng có thể đặt mua qua web site và đăng nhập vào tài khoản để mua hàng.

Khách hàng có thể quản lý đơn hàng mình đã mua.

### **Sơ đồ DFD**

****

Sơ đồ ER

****

* 1. **Phân tích hệ thống về mặt dữ liệu**

Chương trình có 4 chức năng chính:

1. Quản lý khách hàng.
2. Quản lý nhân viên
3. Quản lý sách
4. Cập nhật danh mục.
5. Báo cáo thống kê.
6. **Chức năng khách hàng**

Mô tả hoạt động:

1. **Chức năng Quản lý Sách Bán:**

Mô tả hoạt động:

* Tìm kiếm sách theo chủ đề, nhà xuất bản, tác giả, tên sách theo yêu cầu của khách hàng.
* Cho phép chỉnh sửa, xóa, nhập dữ liệu.
* Hiển thị danh sách các quyển sách bán.

1. **Cập nhật danh mục.**

Mô tả hoạt động:

* Cập nhật danh mục nhà xuất bản
* Cập nhật thể loại
* Cập nhật tác giả
* Cập nhật sách.

1. **Báo cáo thống kê**

Mô tả hoạt động:

* Thống kê số lượng tồn.
* Báo cáo doanh thu.
* Báo cáo sách nhập.
* Báo cáo sách bán
  1. **Bảng thuộc tính dữ liệu**

**Bảng 2. 1 Khách hàng**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **KDL** | **Ràng buộc** | **Ghi chú** |
| MaKhachHang | int | PK | Mã khách hàng |
| HoTen | nvarchar |  | Họ tên |
| TaiKhoan | varchar |  | Tài khoản |
| MatKhau | nvarchar |  | Mật khẩu |
| Email | nvarchar |  | Email |
| DiaChi | nvarchar |  | Địa chỉ |
| DienThoai | varchar |  | Điện thoại |
| GioiTinh | nvarchar |  | Giới tính |
| NgaySinh | datetime |  | Ngày sinh |
| TrangThai | int |  | Trạng thái |

**Bảng 2. 2 Nhân viên**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **KDL** | **Ràng buộc** | **Ghi chú** |
| TaiKhoan | vachar | PK | Tài khoản |
| MatKhauNV | vachar |  | Mật khẩu nhân viên |
| TenNhanVien | nvachar |  | Tên nhân viên |
| NgaySinh | date |  | Ngày sinh |
| GioiTinh | nvachar |  | Giới tính |
| Email | vachar |  | Email |
| SoDienThoai | vachar |  | Số điện thoại |
| QueQuan | nvachar |  | Quê quán |
| Quyen | int |  | Quyền |

**Bảng 2. 3 Phiếu nhập**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **KDL** | **Ràng buộc** | **Ghi chú** |
| MaPhieuNhap | int | PK | Mã phiếu nhập |
| TenNhaCungCap | int |  | Tên nhà cung cấp |
| TaiKhoanNV | vachar | PK | Tài khoản nhân viên |
| NgayNhapKho | date |  | Ngày nhập kho |
| ThanhTien | money |  | Thành tiền |

**Bảng 2. 4 Nhà xuất bản**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **KDL** | **Ràng buộc** | **Ghi chú** |
| MaNXB | int | PK | Mã nhà xuất bản |
| TenNXB | nvachar |  | Tên nhà xuất bản |
| DiaChi | nvachar |  | Địa chỉ |
| DienThoai | vachar |  | Điện thoại |

**Bảng 2. 5 Chủ đề**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **KDL** | **Ràng buộc** | **Ghi chú** |
| MaChuDe | int | PK | Mã chủ đề |
| TenChuDe | nvachar |  | Tên chủ đề |

**Bảng 2. 6 Sách**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **KDL** | **Ràng buộc** | **Ghi chú** |
| MaSach | int | PK | Mã sách |
| TenSach | nvachar |  | Tên sách |
| GiaBan | decimal |  | Giá bán |
| MoTa | nvachar |  | Mô tả |
| AnhBia | nvachar |  | Ảnh bìa |
| NgayCapNhat | datetime |  | Ngày cập nhật |
| SoLuongTon | int |  | Số lượng tồn |
| Moi | int |  | Mới |
| MaNXB | int | PK | Mã nhà xuất bản |
| MaChuDe | int | PK | Mã chủ đề |

**Bảng 2. 7 Tác giả**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **KDL** | **Ràng buộc** | **Ghi chú** |
| MaTacGia | int | PK | Mã tác giả |
| TenTacGia | nvachar |  | Tên tác giả |
| DiaChi | nvachar |  | Địa chỉ |
| TieuSu | nvachar |  | Tiểu sử |
| DienThoai | vachar |  | Điện thoại |

**Bảng 2. 8 Tham gia**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **KDL** | **Ràng buộc** | **Ghi chú** |
| MaSach | PK |  | Mã sách |
| MaTacGia | PK |  | Mã tác giả |
| VaiTro | nvachar |  | Vai trò |
| ViTri | nvachar |  | Vị trí |

**Bảng 2. 9 Chi tiết phiếu nhập**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **KDL** | **Ràng buộc** | **Ghi chú** |
| MaPhieuNhap | int | PK | Mã phiếu nhập |
| MaSach | int | PK | Mã sách |
| SoLuongSachNhap | int |  | Số lượng tồn |
| GiaSachNhap | money |  | Giá sách nhập |
| ThanhTien | money |  | Thành tiền |

**Bảng 2. 10 Chi tiết đơn hàng**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **KDL** | **Ràng buộc** | **Ghi chú** |
| MaDonHang | int | PK | Mã đơn hàng |
| MaSach | int | PK | Mã sách |
| SoLuong | int |  | Số lượng |
| DonGia | decimal |  | Đơn giá |

**Bảng 2. 11 Đơn hàng**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **KDL** | **Ràng buộc** | **Ghi chú** |
| MaDonHang | int | PK | Mã đơn hàng |
| DaThanhToan | int |  | Đã thanh toán |
| TinhTrangGiaoHang | int |  | Tình trạng giao hàng |
| NgayDat | date |  | Ngày đặt |
| NgayGiao | date |  | Ngày giao |
| MaKhachHang | int | FK | Mã khách hàng |

* 1. **Thiết kế Website**

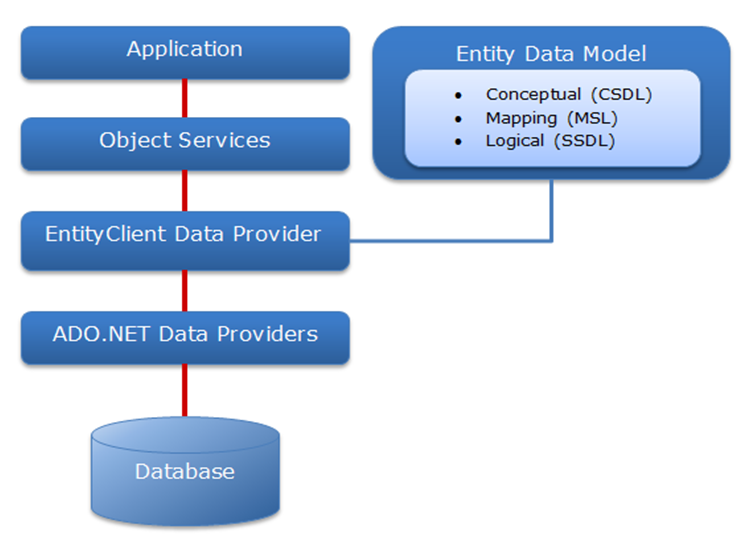
**2.5.1 Entity Framework**

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

**Tính năng của Emtity Framwork**

* Là một công cụ của Microsoft.
* Đang được phát triển như một sản phẩm mã nguồn mở
* Không còn bị ràng buộc hoặc phụ thuộc vài chu kỳ cơ sở dữ liệu quan hệ nào đó có Entity Framwork provider hợp lệ
* Tạo truy vấn SQL từ LINQ to Entities
* Sẽ tạo các truy vấn có tham số
* Theo dõi các thay đổi từ các đối tượng trong bộ nhớ
* Cho phép tạo cá câu lệnh thêm, cập nhật và xóa
* Entity Framwork hỗ trợ stored procedure

### **Kiến trúc Entity Framwork**



-Application: là tầng chứa giao diện trang Web (HTML, CSS, Javascript, hình ảnh,) và các đoạn mã nguồn (C#, VB) để tương tác dữ liệu với các tầng khác trong mô hình thông qua Object Services.

-Object Services: là tầng chứa quá trình tương tác giữa ứng dụng và database. Object Services cung cấp các tiện ích để truy vết các thay đổi và quản lý nhận dạng, đồng thời là các quan hệ và thay đổi ở database.

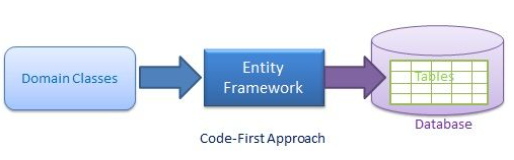
**-EntityClient Data Provider** :Đây là tầng cung cấp các kết nối, diễn dịch các truy vấn thực thể thành truy vấn nguồn dữ liệu (chuyển L2E – [LINQ](https://www.dammio.com/glossary/linq) to Entity hay các truy vấn thực thể SQL thành truy vấn SQL), trả về data reader để [EF](https://www.dammio.com/glossary/ef) dùng chuyển dữ liệu thực thể thành các đối tượng. Phần này kết nối ADO.NET Data Providers để gửi hoặc lấy dữ liệu từ database. Tầng này hoàn toàn khác với EDM (Entity Data Model) khi thực thi các truy vấn tương tự như cách thực hiện ở ADO.NET Provider. EntityClient Data Provider sử dụng ESQL (Entity SQL), một ngôn ngữ truy vấn độc lập dạng văn bản, tương tự như SQL.

**ADO.NET Data Providers** : Đây là tầng thấp nhấp để dịch các truy vấn L2E ([LINQ](https://www.dammio.com/glossary/linq) to Entity) thông qua cây lệnh thành các câu lệnh SQL và thực thi các câu lệnh trong hệ thống DBMS (database management system – hệ quản lý dữ liệu) nào đó. Tầng này kết với database sử dụng ADO.NET.

**EDM (Entity Data Model):** chứa 3 phần chính: mô hình khái niệm (CSDL – Conceptual schema definition language), mô hình ánh xạ (MSL – mapping specification language) và mô hình lưu trữ (SSDL – store schema definition language). EDM khác với EntityClient Data Provider ở chỗ EDM sử dụng [LINQ](https://www.dammio.com/glossary/linq) là ngôn ngữ truy vấn tương tác với database.

**2.5.2 Giới thiệu mô hình Code First**

Quy trình phát triển theo cách tiếp cận Code first sẽ là: Tạo hoặ sửa đổi các lớp miền-> cấu hình các lớp miền này bằng các thuộc tính chú thích dữ liệu hoặc Fluent API ->Tạo hoặ cập nhật lược đồ cơ sở dữ liệu bằng automated migration hoặc code – based migration.

****

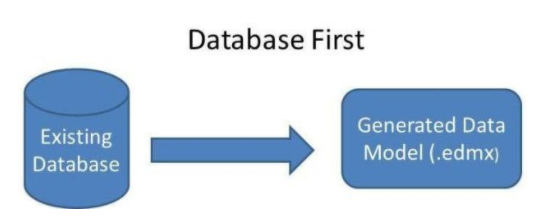
Hình 2. Quy trình code first

**Các quy ước trong Code First:**

1. Quy ước tên bảng: Khi tạo 1 class mô tả các thực thể là User sẽ được lưu trữ trong database thì Entity Framework sẽ tự động tạo ra 1 bảng có tên là Users
2. Quy ước khóa chính: Khi bạn tạo 1 thuộc tính có tên là UserId trong lớp User của model thì thuộc tính này được nhận làm khóa chính. Hơn nữa nó sẽ thiết lập 1 cột khóa auto-increment để lưu trữ giá trị nếu kiểu khóa là số nguyên.
3. Quy ước về mối quan hệ: Entity Framework cung cấp các quy ước khác nhau để nhận biết 1 mối quan hệ giữa 2 model dựa vào tên của thuộc tính và kiểu dữ liệu Để tìm hiểu chi tiết hơn mình sẽ làm 1 ví dụ bên dưới.

**2.5.3 Giới thiệu mô hình Database First**

Database First tức là xây dựng cơ sở dữ liệu trước rồi mới đến việc xây dựng mã nguồn ứng dụng. Database First cho phép chúng ta đảo ngược 1 mô hình từ 1 database đã tồn tại từ trước. Mô hình này được lưu trong tập tin tên là EDMX (.edmx) và có thể xem và chỉnh sửa trong Entity Framework Designer (phần thiết kế Entity Framework).

****

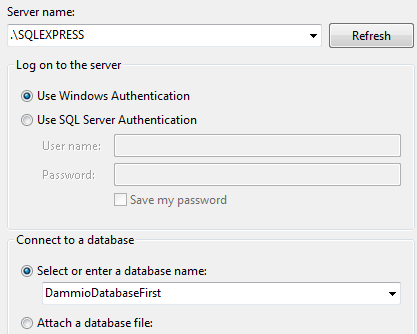
**Ưu, nhược điểm khi tạo database first:**

|  |  |
| --- | --- |
| **Ưu điểm** | **Nhược điểm** |
| * Dễ dàng, quen thuộc với hầu hết tất cả mọi người * Code tự động sinh ra dựa vào các bảng trong database * Phù hợp cho các ứng dụng lớn | * Không quản lý được version của database * Bất tiện khi làm việc theo nhóm, cả nhóm thường phải chung 1 server để database thống nhất cho cả team |

## **Tạo 1 ứng dụng ASP .NET MVC với cách tiếp cận Database First**

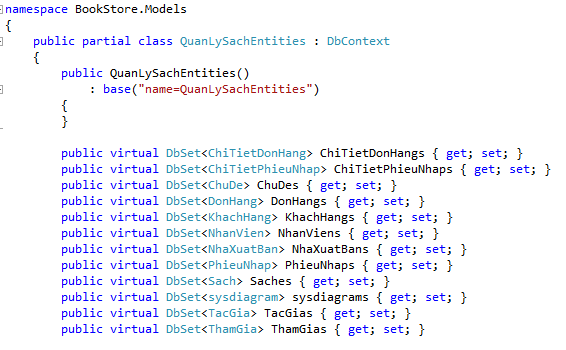
Bước 1: Tạo project

Trước hết tạo database. Để tạo database chọn View -> **Server Explorer**. Chuột phải vào Database Connections -> Add Connections Sau đó điền tên database là **DammioDatabaseFirst**



Bước 2: Thiết lập dữ liệu từ database

Bước 3: Tạo các model từ database

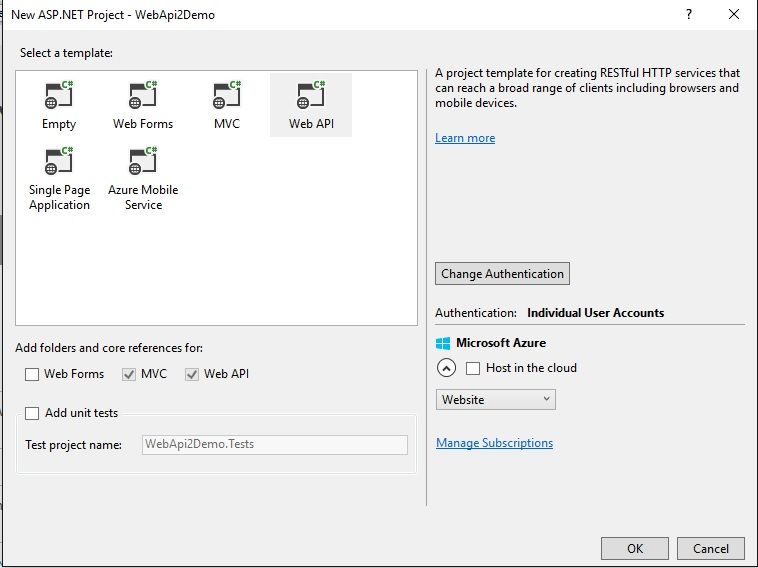


**2.5.4 Entity Framework API**

Entity Framework là một khung ORM mã nguồn mở cho các ứng dụng .NET được Microsoft hỗ trợ. Nó cho phép các nhà phát triển làm việc với dữ liệu bằng cách sử dụng các đối tượng của các lớp cụ thể của miền mà không cần tập trung vào các bảng và cột cơ sở dữ liệu cơ bản nơi dữ liệu này được lưu trữ. Với Entity Framework, các nhà phát triển có thể làm việc ở mức độ trừu tượng cao hơn khi họ xử lý dữ liệu và có thể tạo và duy trì các ứng dụng hướng dữ liệu với ít mã hơn so với các ứng dụng truyền thống.

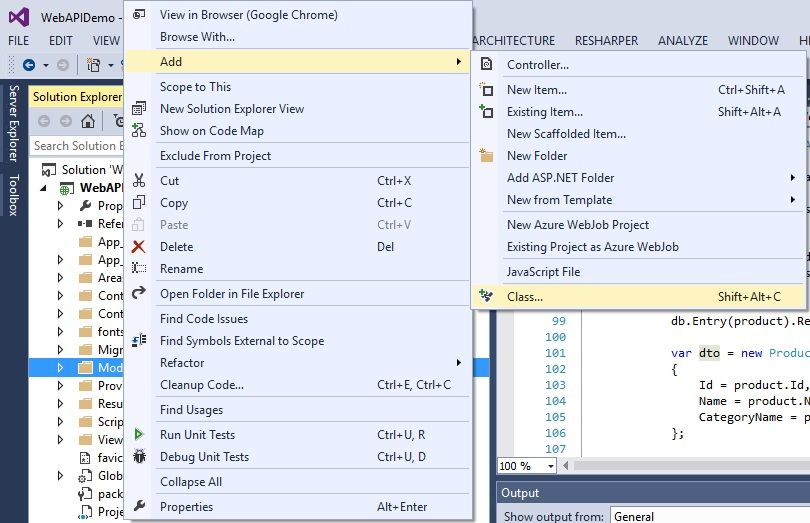
Ví dụ tạo 1 project

Bước 1: Mở visual studio 2013 ->New project ->tại Dialog chọn Asp.net website application -> chọn web API



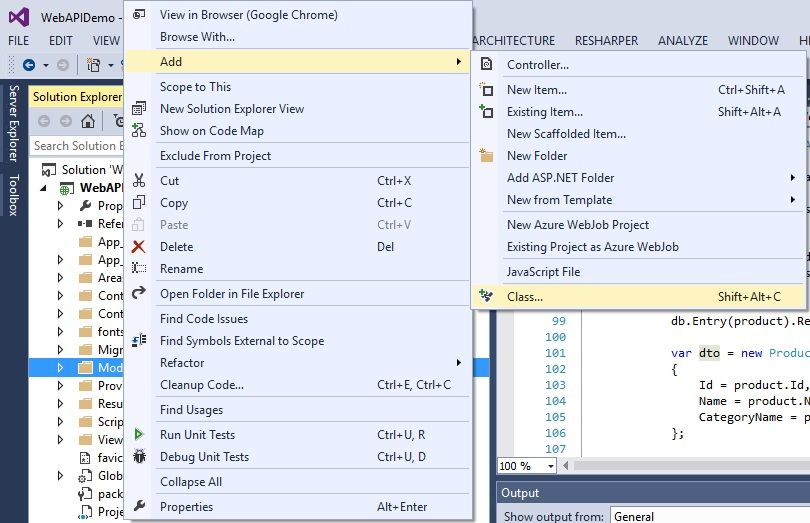
Bước 2: Tạo data models

Click chuột phải tạo Models chọn “New” ->Class

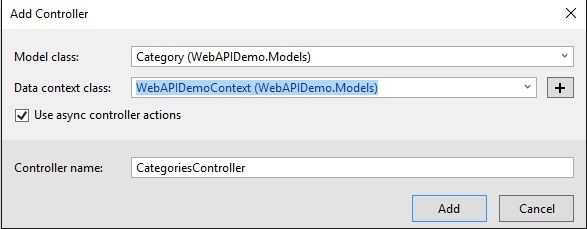


Bước 3: Tạo cotroller

Chột phải vào Controller->Add ->Controller. Tại cửa sổ “Add Scaffold” chọn “Web API 2 Controller with actions,using Entity Framework”



Tại cửa sổ “Add controller” ->Model class dropdownlist ->chọn Category class. Tích chọn”use async controller actions”.Sau đó đặt tên cho Controller.Tiếp đến click(+) button ở mục “Data context class”



**2.5.5 Điều khiển dữ liệu – Controller**

Lớp controller chứa cac phương thức public gọi là các phương thức hành động(Action method).

Controller và phương thức hành động của nó xử lý các yêu cầu đến từ trình duyệt, lấy dữ liệu model cần thiết và trả về các phản hồi thích hợp.

Trong ASP.Net MVC, mọi tên lớp của controller phải kết thúc bằng từ “Controller”.

(chụp hình

**2.5.6 Sinh giao diện và chia sẻ dữ liệu**

# Chương 3 CÀI ĐẶT VÀ ĐÁNH GIÁ THỬ NGHIỆM

* 1. **Cài đặt và triển khai Website**

### **Cài đặt SQL Sever**

* + - 1. *Khái quát*

SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System – RDBMS) sử dụng câu lệnh SQL (Transact-SQL) để trao đổi dữ liệu giữa máy khách và máy cài SQL Server. Một RDBMS bao gồm các

database, database engine và các dứng dụng dùng để quản lý dữ liệu và các bộ phận khác nhau trong RDBMS.

SQL Server được tối ưu để chạy trên môi trường cơ sở dữ liệu rất lớn (Very Large Database Enviroment) lên đến TeraBytes và có thể phục vụ cùng lúc cho hàng nghìn người dùng. SQL Server có thể kết hợp với các server khác như Microsoft Internet Information Server (IIS), E-Commerce Server, Proxy Server,..

SQL Server có một vài phiên bản có thể kể đến như sau:

* Enterprise: Chứa tất cả các đặc điểm nổi bật của SQL Server, bao gồm nhân bộ máy cơ sở dữ liệu và các dịch vụ đi kèm cùng với các công cụ cho tạo và quản lý phân cụm SQL Server. Nó có thể quản lý các CSDL lớn tới 524 PetaBytes và đánh địa chỉ 12 TeraBytes bộ nhớ và hỗ trợ tới 640 bộ vi xử lý.
* Standard: Rất thích hợp cho các công ty vừa và nhỏ vì giá thành rẻ hơn nhiều so với bản Enterprise, nhưng lại bị giới hạn một số chức năng cao cấp khác.
* Developer: Có đầy đủ các tính năng của phiên bản Enterprise nhưng giới hạn nhiều điều kiện như số lượng người kết nối vào server cùng một lúc,… đây là phiên bản sử dụng cho phát triển và kiểm tra ứng dụng. Phiên bản này phù hợp cho tất cả các cá nhân, tổ chức xây dựng và kiểm tra ứng dụng.
* Express: SQL Server Express dễ sử dụng và quản trị CSDL đơn giản. Được tích hợp với Microsoft Visual Studio, nên dễ dàng để phát triển các dứng dụng dữ liệu, an toàn trong lưu trữ và nhanh trong triển khai. SQL Server Express là phiên bản **miễn phí**, không giới hạn về số cơ sở dữ liệu hoặc người sử dụng, nhưng nó chỉ dùng cho 1 bộ vi xử lý với 1 GB bộ nhớ và 10 GB file cơ sở dữ liệu. SQL Server Express là lựa chọn tốt cho những người dùng chỉ cần một phiên bản SQL Server 2005 nhỏ gọn, dùng trên máy chủ có cấu hình thấp, những nhà phát triển ứng dụng không chuyên hay những người yêu thích xây dựng các ứng dụng nhỏ.

Đối tượng của SQL Server là các bảng dữ liệu với các cột và các hàng. Cột được gọi là trường dữ liệu và hàng là bản ghi của bảng. Cột dữ liệu và kiểu dữ liệu xác định tạo nên cấu trúc của bảng.

Khi bảng được tôt chức thành một hệ thống cho một mục đích sử dụng cụ thể vào một công việc nào đó thì sẽ trở thành một cơ sở dữ liệu.

*b Cài đặt*

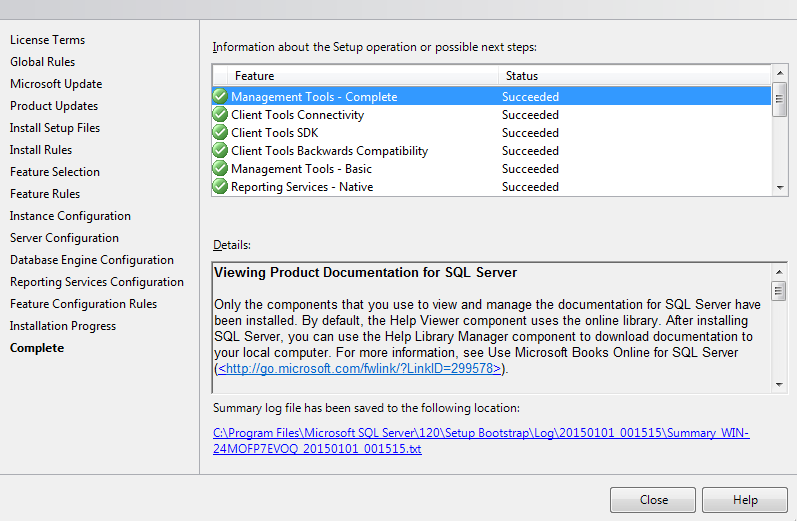
Để có thể sử dụng, và phục vụ cho mục đích phát triển cũng như áp dụng cho mô hình website nhỏ này, nên phiên bản Express của SQL Server là thích hợp nhất vì nó yêu cầu vi xử lý thấp, và cung cấp một phạm vi lưu trữ dữ liệu lên đến 10 GB. Những điều kiện như vậy là hoàn hảo để xây dựng nên một webite với quy mô nhỏ.

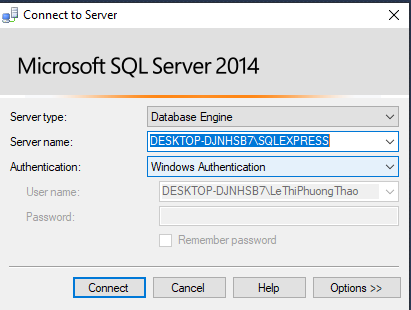
* Tải SQL Server

Để cài đặt SQL Server, chúng ta có thể truy cập trực tiếp vào trang chủ của Microsoft để tải bộ công cụ cho máy tính theo địa chỉ:

[*https://www.microsoft.com/en-in/sql-server/sql-server-downloads*](https://www.microsoft.com/en-in/sql-server/sql-server-downloads)

Sau đó tiến hành cài đặt như SQL Server 2014 đã được học.





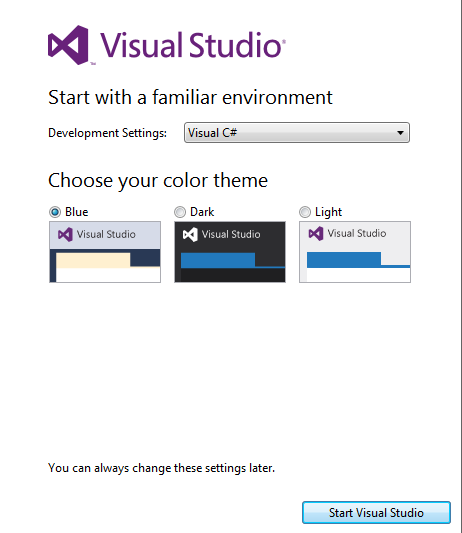
### **Cài đặt Visual studio 2013**

### Tổng quan về Visual Studio 2013

**Visual studio** là một trong những công cụ hỗ trợ [lập trình website](https://monamedia.co/dich-vu/thiet-ke-website/) rất nổi tiếng nhất hiện nay của Microsoft và chưa có một phần mềm nào có thể thay thế được nó. Visual Studio được viết bằng 2 ngôn ngữ đó chính là C# và VB+. Đây là 2 ngôn ngữ lập trình giúp người dùng có thể lập trình được hệ thống một các dễ dàng và nhanh chóng nhất thông qua Visual Studio.

### Cài đặt Visual Studio 2013

Đối với việc cài đặt Visual Studio 2013, thì đầu tiên ta truy cập vào website: <https://code.visualstudio.com>. Để dowload phần mềm về máy và chạy tệp tin với dạng .exe để bắt đầu cài đặt. Chúng ta tiến hành cài đặt như các chương trình khác.

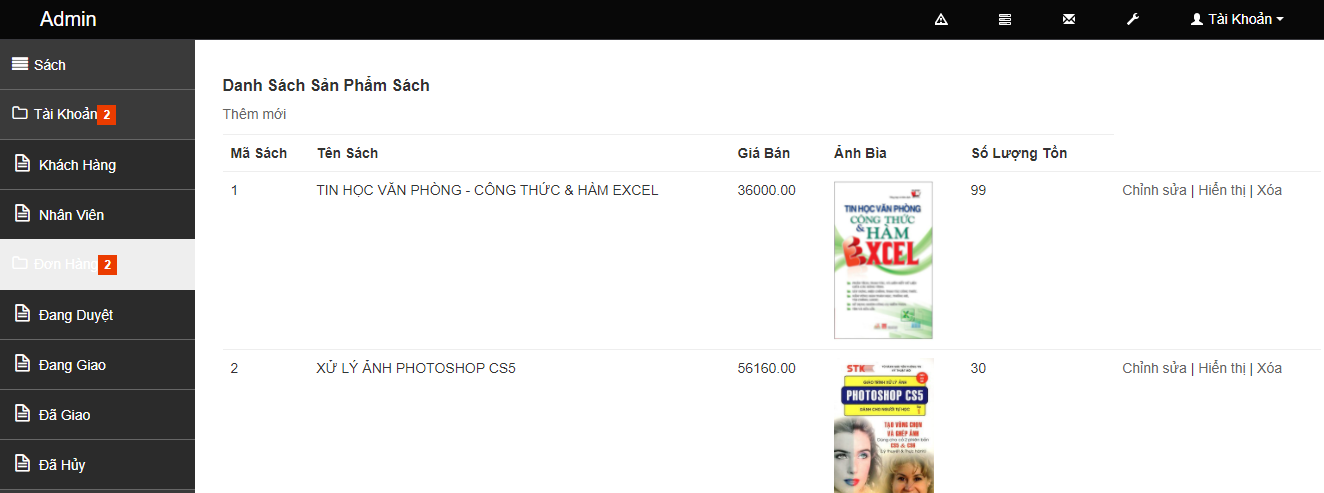


**3.2 Đăng ký Web Hosting miễn phí**

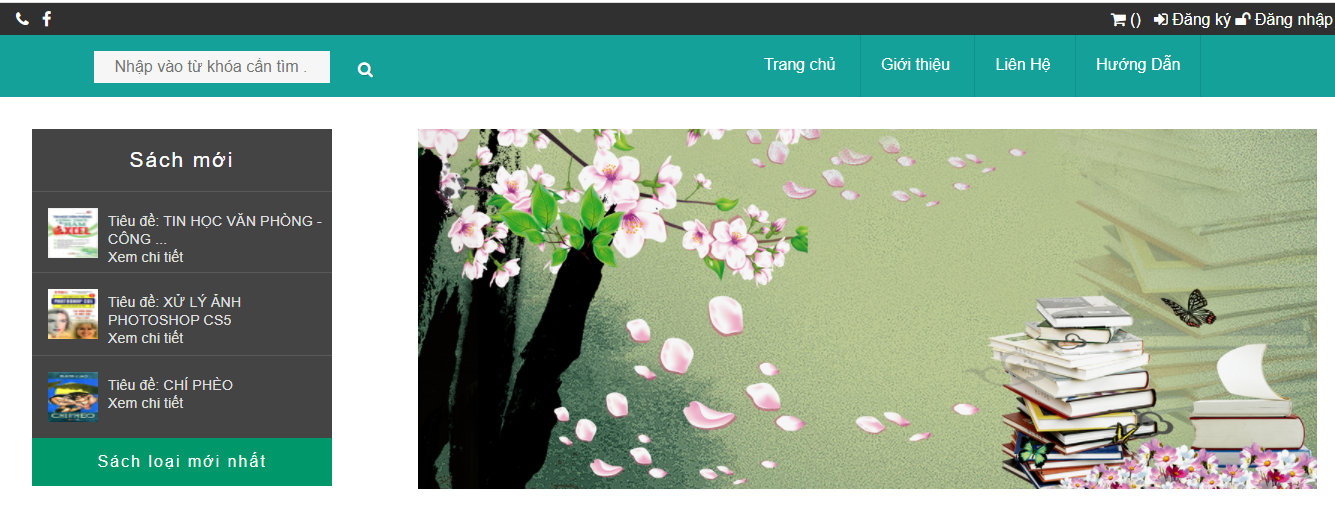
**3.3 Quản lý Hosting Server**

* 1. **Triển khai Website Asp.NET MVC**

1. **Trang chủ Admin**

****

1. **Giao diện trang chủ cho người dùng**

****

1. **Giao diện nhà xuất bản**



1. **Giao diện đặt hàng**

