TRƯỜNG CAO ĐẲNG CÔNG THƯƠNG TP. HCM

Khoa Công Nghệ Thông Tin

**THỰC TẬP TỐT NGHIỆP**

****

**ĐỀ TÀI:**

**WEBSITE BÁN HÀNG ĐỒ THỦ CÔNG MỸ NGHỆ**

GVHD: Huỳnh Trọng Đức

SVTH: Nguyễn Công Thức

MSSV: 2117110127

TPHCM, tháng 10 năm 20

Mục Lục

DANH MỤC CÁC BẢNG 5

DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ 6

Chương 1. Tổng Quan 7

1.1.Giới thiệu về website 7

1.1.1.Khái niệm. 7

1.1.2.Lịch sử phát triển của website. 7

1.1.3.Ưu điểm và nhược điểm của website. 8

1.1.4.Các loại website. 10

1.2.Quy trình xây dựng website. 10

1.3.Môi trường phát triển ứng dụng web. 12

Chương 2.GIỚI THIỆU VỀ ASP.NET MVC 13

2.1.Giới thiệu về ASP.NET ASP.NET MVC 13

2.2.Mô hình MVC và cách thức hoạt dộng 13

2.2.1.Khái niệm 14

2.2.2.Sự tương tác giữa cáp lớp trong mô hình MVC. 15

2.2.3.Kiến trúc ASP.NET MVC 17

2.3.EntityFreamwork 18

2.3.1.Khái niệm. 18

2.3.2.Lịch sử phát triển EntityFreamwork. 19

2.3.3.Ưu điểm nhược điểm của EntityFreamwork 20

2.4.Ngôn ngữ thiết kế giao diện 21

2.4.1 HTML 21

2.4.2.CSS 21

2.4.3.Bootstrap Freamwork 23

2.5.Microsoft SQL Server 23

2.5.1.Khái niệm về Microsoft SQL server 23

2.5.2.Chức năng 23

2.5.3.Ưu điểm 24

2.5.4.Nhược điểm 24

Chương 3.Phân tích thiết kế cơ sở dữ liệu và giao diện webiste 24

3.1 Đặt tả hệ thống. 24

3.2.Danh sách cá tác nhân. 25

3.3.Các bảng dữ liệu 27

3.3.1.Bảng Category 27

3.3.2.Bảng Contact 27

3.3.3.Bảng Link 28

3.3.4.Bảng Menu 28

3.3.5.Bảng Order 29

3.3.6.Bảng OrderDetail 30

3.3.7.Bảng Product 30

3.3.8 Bảng Slider 31

3.3.9 Bảng User 31

3.4.Database Diagram 32

3.5.Giao diện người dùng 33

3.6.Giao diện trang quản trị 35

Đánh giá kết quả và hướng phát triển 35

3.7.Kết quả đạt được 35

3.8.Hạn chế 35

3.9.Hướng phát triển 35

TÀI LIỆU THAM KHẢO 35

LỜI NÓI ĐẦU

Trong thời đại công nghệ thông tin đang bùng nổ và phát triển rộng rãi như hiện nay có thể nói rằng ngành công nghệ thông tin là một trong những ngành kinh tế mũi nhọn của nhiều quốc gia trên thế giới. Bởi nó có rất nhiều ứng dụng và đem lại cho con người nhiều lợi ích trong mọi lĩnh vực của cuộc sống như: khoa học, kinh tế, chính trị và xã hội…Internet trở nên quen thuộc với mỗi chúng ta hơn. Thông qua mạng Internet, các doanh nghiệp, công ty, cửa hàng đã quảng bá các dịch vụ của mình để tất cả mọi người cùng biết tên thương hiệu đó.

Như chúng ta đã biết, hiện nay các dịch vụ xuất hiện nhằm nâng cao cuộc sống con người. Các dịch vụ chăm sóc sức khỏe ngày càng phát triển, chúng ta cũng chú trọng hơn trong việc thẩm mỹ làm đẹp. Trong đó Nha khoa luôn là vấn đề được nhiều người quan tâm, dịch vụ nha khoa rất cần thiết và gần gũi với con người. Xuất phát từ thực tế đó, nhóm em đã chọn đề tài XÂY DỰNG WEBSITE BÁN HÀNG THỦ CÔNG MỸ NGHỆ làm đề tài.

Mặc dù đã rất cố gắng thực hiện đề tài nhưng vì năng lực cũng như thời gian còn hạn chế nên chương trình khó tránh khỏi những thiếu xót, rất mong thầy cô thông cảm. Những góp ý của thầy cô là bài học và là hành trang để em vững bước vào cuộc sống sau này. Qua đây, em xin gửi lời cám ơn đến Thầy Huỳnh Trọng Đức, người đã nhiệt tình hướng dẫn chỉ bảo em trong quá trình thực hiện hoàn thành đề tài.

# DANH MỤC CÁC BẢNG

# DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ

[Hình 1: Quy trình xây dựng website 8](#_Toc53821576)

# Tổng Quan

## Giới thiệu về website

### Khái niệm.

Một website (cũng viết như trang web ) là một tập hợp các [trang web](https://en.wikipedia.org/wiki/Web_page) và nội dung liên quan được xác định bởi một chung [tên miền](https://en.wikipedia.org/wiki/Domain_name) và được công bố trên ít nhất một [máy chủ web](https://en.wikipedia.org/wiki/Web_server) . Các ví dụ đáng chú ý là [wikipedia.org](https://en.wikipedia.org/wiki/Wikipedia) , [google.com](https://en.wikipedia.org/wiki/Google) và [amazon.com](https://en.wikipedia.org/wiki/Amazon_(company)).

Tất cả các trang web có thể truy cập công khai đều tạo thành [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web) . Cũng có những trang web riêng tư chỉ có thể được truy cập trên [mạng riêng](https://en.wikipedia.org/wiki/Intranet) , chẳng hạn như trang web nội bộ của công ty dành cho nhân viên.

Các trang web thường dành riêng cho một chủ đề hoặc mục đích cụ thể, chẳng hạn như tin tức, giáo dục, thương mại, giải trí hoặc [mạng xã hội](https://en.wikipedia.org/wiki/Social_networking) . [Siêu liên kết](https://en.wikipedia.org/wiki/Hyperlink) giữa các trang web hướng dẫn điều hướng của trang web, thường bắt đầu bằng [trang chủ](https://en.wikipedia.org/wiki/Home_page).

[Người dùng](https://en.wikipedia.org/wiki/User_(computing)) có thể truy cập các trang web trên nhiều loại thiết bị, bao gồm [máy tính để bàn](https://en.wikipedia.org/wiki/Desktop_computer) , [máy tính xách tay](https://en.wikipedia.org/wiki/Laptop) , [máy tính bảng](https://en.wikipedia.org/wiki/Tablet_computer) và [điện thoại thông minh](https://en.wikipedia.org/wiki/Smartphone) . Các [ứng dụng phần mềm](https://en.wikipedia.org/wiki/Application_software) được sử dụng trên những thiết bị này được gọi là một [trình duyệt web](https://en.wikipedia.org/wiki/Web_browser).

### Lịch sử phát triển của website.

Đó là ngày 21/12/1990, tại cơ sở của CERN nằm ở dãy núi Alps của Thụy Sĩ, nhà vật lý học, chuyên gia phần mềm người Anh Tim Berners-Lee đã khởi chạy thành công trang web đầu tiên trên thế giới, với tên miền info.cern.ch, chạy trên 1 máy chủ NeXT của CERN. Hiện trang web này vẫn còn tồn tại cho đến ngày nay.

Dĩ nhiên, vào thời điểm đó, chỉ có Berners-Lee và các đồng nghiệp của mình tại CERN mới có thể truy cập vào trang web này bởi 1 lý do đơn giản: chỉ có máy tính của họ mới có trình duyệt web. Phải đến tận năm 1993, khi trình duyệt Mosaic dành cho nền tảng Unix và Windows ra đời, lúc này website mới bắt đầu dần trở nên phổ biến hơn.

Nội dung của trang web cung cấp các thông tin về world wide web, nền tảng cơ bản của Internet, nơi các văn bản và các trang trên Internet có thể truy cập bởi các URL (Uniform Resource Locator – Định vị Tài nguyên thống nhất) và kết nối với nhau thông qua các siêu liên kết.

Khi Berners-Lee sáng tạo ra trang web đầu tiên, mạng “Internet” là một nhóm các tài liệu tĩnh, sử dụng gần như độc quyền bởi các tổ chức quốc phòng và các tổ chức nghiên cứu mang tính học thuật. Ý tưởng khởi tạo website của Berners-Lee nhằm giúp cho phép các tài liệu điện tử trên Internet dẽ dàng được tìm kiếm và chia sẻ.

“Khi chúng ta liên kết thông tin trên web, chúng ta có thể khám phá sự thật, tạo ra những ý tưởng, mua bán mọi thứ và tạo nên những mối quan hệ với tốc độ và quy mô không thể tưởng tượng được vào thời đại đó”, Berners-Lee, “cha đẻ” của trang web đầu tiên chia sẻ.

Mặc dù trang web đầu tiên được ra đời vào ngày 21/12/1990, tuy nhiên đến ngày 6/8/1991, Berners-Lee mới đăng tải công khai bản tóm tắt về dự án xây dựng trang web của mình trên một số nhóm hoạt động trên Internet. Đây là động thái đánh dấu sự ra mắt công khai của trang web đầu tiên trên thế giới với cộng đồng Internet.

Năm 1994, Berners-Lee thành lập Tổ chức World Wide Web (thường biết đến với tên gọi W3C) tại học viện công nghệ MIT (Bang Massachusetts, Mỹ). Tổ chức W3C sẽ chịu trách nhiệm để tạo ra những tiêu chuẩn cho các trang web để đảm bảo rằng các trang web khác nhau sẽ hoạt động theo cách tương tự nhau.

Hiện tại Berners-Lee vẫn tiếp tục giữ vai trò lãnh đạo W3C. Mặc dù hiện nay vẫn có một vài sự khác biệt giữa các trang web và trình duyệt, nhưng nếu không có sự định hướng của W3C, chắc hẳn thế giới web sẽ trở thành một “mớ hỗn độn” và sẽ không như chúng ta biết ngày hôm nay.

### Ưu điểm và nhược điểm của website.

#### Ưu điểm.

* Tiếp cận người dùng trực tuyến.  
  Nếu doanh nghiệp không có trang web riêng, khách hàng sẽ chỉ có thể mua sản phẩm, nhận dịch vụ và tương tác với chủ doanh nghiệp trong giờ hành chính ngoài. Điều này khiến các dịch vụ cũng như hoạt động của doanh nghiệp bị giới hạn. Tuy nhiên, mọi thứ sẽ thay đổi nếu doanh nghiệp xây dựng một trang web riêng. Không bị giới hạn về thời gian, không gian, tăng khả năng tiếp cận khách hàng và thực thi những đãi ngộ một cách kịp thời.
* Tăng phạm vi người dùng.  
  Một cửa hàng địa phương có thể thu hút được khách địa phương nhưng lại là hạn chế đối với những khách hàng ở khu vực khác. Tuy nhiên, khi doanh nghiệp xây dựng một trang web riêng thì phạm vị khách hàng sẽ không bị giới hạn. Cơ hội nhận được đơn đặt hàng của khách từ khắp mọi nơi trên đất nước sẽ ngày càng tăng cao.
* Tăng tính tương tác.  
  Khi có một trang web riêng, doanh nghiệp sẽ giúp khách hàng tìm kiếm một cách nhanh chóng các sản phẩm và hiển thị những dịch vụ mong muốn. Điều này làm giảm bớt thời gian tìm kiếm cho khách hàng và đưa doanh nghiệp lên một thứ hạng tốt hơn.
* Nền tảng xây dựng thương hiệu.  
  Xây dựng thương hiệu là điều đáng quan tâm của một doanh nghiệp và trang web sẽ giúp doanh nghiệp làm điều này. Nó phản ánh một hình ảnh chuyên nghiệp thông qua giao diện và các hoạt động của doanh nghiệp. Địa chỉ email, tên miền và cách giao tiếp với khách hàng qua web giúp doanh nghiệp xây dựng uy tín, tạo dựng thương hiệu và truyền bá hình ảnh trên toàn thế giới.
* Tăng tính cạnh tranh.  
  Một doanh nghiệp nhỏ có trong tay một trang web sẽ có lợi thế cạnh tranh với các đối thủ “nặng ký” khác. Trên thực tế, các đối thủ cạnh tranh không có trang web thì các khách hàng sẽ có xu hướng chuyển sang những doanh nghiệp có trang web vì ở đó họ có thể thỏa mãn được nhu cầu mua bán của bản thân.
* Cập nhật thông tin một cách nhanh chóng.  
  Thông tin mới nên được chia sẻ với khách hàng và điều này trang web có thể hoàn thành một cách xuất sắc. Không chỉ cập nhật thông tin nhanh chóng, tải lên trang dễ dàng mà còn tốn ít thời gian cũng như công sức. Trước đây, khi doanh nghiệp có thông tin mới muốn được chia sẻ đến khách hàng thường sử dụng phương thức phát tờ rới quảng cáo. Tuy nhiên, với cách thức này doanh nghiệp phải chi trả một số lượng không nhỏ tiền in ấn, phát hành cũng như thời gian quảng bá. Nếu sử dụng trang web, doanh nghiệp chỉ cần thu thập thông tin, tải lên trang và chờ đón phản hồi.
* Dễ dàng lấy ý kiến phản hồi từ khách hàng.  
  Lấy ý kiến của khách hàng là việc làm quan trọng giúp doanh nghiệp nắm bắt tâm lý khách hàng và có chiến lược thay đổi phù hợp. Thông tin phản hồi từ khách hàng có thể dễ dàng thu thập được thông qua trang web. Bởi, khách hàng có thể tự do cung cấp thông tin riêng tư, không bị miễn cưỡng và đặc biệt là không mất quá nhiều thời gian.

#### Nhươc điểm.

### Các loại website.

Website tĩnh không phải chứa những nội dung, hình ảnh tĩnh, mà là những nội dung không được hoặc hiếm khi được chỉnh sửa. Nội dung của các webpage được lưu trực tiếp trên máy chủ và truyền tải nguyên vẹn tới người sử dụng. Thông thường, nội dung website tĩnh được chứa trên một file HTML trên máy chủ, chỉ có thể tùy chỉnh tương đối thông qua các các ứng dụng client-side.

Ngược lại, nội dung của website động thường xuyên và dễ dàng được thay đổi. Những nội dung này sẽ thay đổi dựa trên tương tác của người dùng, của một điều kiện cụ thể nào đó hoặc theo ý muốn của chủ trang web.

## Quy trình xây dựng website.

Các giai đoạn phát triển của website.

1. Thu thập thông tin

2. Kế hoạch

3. Thiết kế

4. Phát triển

5. Kiểm tra và giao hàng

6. Bảo trì

Chi tiết các giai đoạn phát triển:

1. Thu thập thông tin.

Bước đầu tiên trong việc thiết kế một trang web thành công là để thu thập thông tin. Nhiều điều cần được xem xét khi các giao diện của trang web của bạn được tạo ra.

Một số điều cần xem xét là:

**Mục đích :** mục đích của trang web là gì? Bạn có muốn cung cấp thông tin, quảng bá dịch vụ, bán sản phẩm ...?

**Mục tiêu :** Những gì bạn hy vọng để thực hiện bằng cách xây dựng trang web này? Hai trong số các mục tiêu phổ biến hơn hoặc là để kiếm tiền hoặc chia sẻ thông tin.

**Đối tượng :** Có một nhóm người nào đó sẽ giúp bạn đạt được mục tiêu của bạn? Đó là hữu ích để hình "lý tưởng" người mà bạn muốn ghé thăm trang web của bạn. Hãy xem xét độ tuổi quan hệ tình dục, hoặc lợi ích của mình - điều này sau này sẽ giúp xác định phong cách thiết kế tốt nhất cho trang web của bạn.

**Nội dung :** Các thông tin các đối tượng mục tiêu sẽ được tìm kiếm trên trang web của bạn? Họ tìm kiếm thông tin cụ thể, một sản phẩm cụ thể hay dịch vụ, đặt hàng trực tuyến ...?

1. Kế hoạch.

Sủ dụng thông tin thu thập được từ bước một, đó là thời gian để đưa ra kế hoạch cho website của bản

Trong giai đoạn lập kế hoạch, thiết kế web của bạn cũng sẽ giúp bạn quyết định những công nghệ cần được thực hiện. Các yếu tố như hình thức tương tác, thương mại điện tử, đèn flash, vv được thảo luận khi lập kế hoạch trang web của bạn.

1. Thiết kế.  
   Từ các thông tin đã thu thập được đến giai đoạn này, đó là thời gian để xác định các giao diện của website.  
   Thiết kế web của bạn sẽ tạo ra một hoặc nhiều nguyên mẫu thiết kế cho trang web của bạn. Điều này thường là một hình ảnh jpg. Về những gì các thiết kế cuối cùng sẽ như thế. Thường thì bạn sẽ nhận được một email với các-up giả cho trang web của bạn, trong khi các nhà thiết kế khác đi một bước xa hơn bằng cách cho bạn truy cập vào một khu vực an toàn của trang web của họ có nghĩa là cho khách hàng để xem công việc đang tiến.  
   Dù bằng cách nào, nhà thiết kế của bạn nên cho phép bạn xem các dự án của bạn trong suốt các giai đoạn thiết kế và phát triển. Lý do quan trọng nhất cho việc này là nó cho bạn cơ hội để bày tỏ thích và không thích của bạn trên thiết kế trang web.
2. Phát triển.  
   Thời điểm này là thời điểm mà website được tạo ra. Tại thời điểm này website đã có tất cả những yếu tố về về đồ họa, tác nhân tác động đến website, các chức năng chính của website.  
   về mặt kĩ thuật một trang web thành công đòi hỏi sự hiểu biết về phát triển web đầu cuối. Điều này bao gồm văn bản hợp lệ HTML/CSS phải tuân thủ các tiêu chí chuẩn của website hiện tại, tối ưu hóa chức năng cũng như khả năng tiếp cận với lượng người dùng lớn.
3. Kiểm tra và bàn giao website.  
   Tại thời điểm này, người phát triển web sẽ kiểm tra những chức năng , kịch bản cũng như thực nghiệm cuối cùng cho vấn đề tương thích với các trình duyệt web khác nhau.  
   Một khi đã chấp nhập các thiết và các chức năng của website thì website của bạn sẽ được triển khai cho khách hàng. Hầu hết các đơn vị phát triển website sẽ cung cấp đăng ký tên miền và các dịch vụ hosting cho bạn.
4. Bảo trì.  
   Mặt dù website được thiết kế phát triển kiểm tra chỉnh sửa kỹ càng như thế nào thì vẫn luôn có lỗ hổng. Không những thế, trong suốt quá trình vận hành thì chắc chắn bạn sẽ nhận được những phản hồi về website của bạn và bạn cần chỉnh sửa website của mình theo những phản hồi đó. Vì vậy việc lập kế hoạch cho việc bảo trì và cập nhật website là việc cần thiết.

## Môi trường phát triển ứng dụng web.

* Ngôn ngữ lập trình web:JAVA SPRING, ASP.NET, PHP, PYTHON DJANGO,RUBY ON RAILS….
* Ngôn ngữ thiết kế giao diện: HTML, CSS
* Framework: Bootstrap 4, Entity FrameWork
* Công cụ xây dựng cơ sở dữ liệu: Microsoft SQL Server, Oracle Database, MySQL, MongoDb
* Phần mềm phát triển: Visual Studio, Visual Studio Code, Eclipse, Adobe Dreamweaver

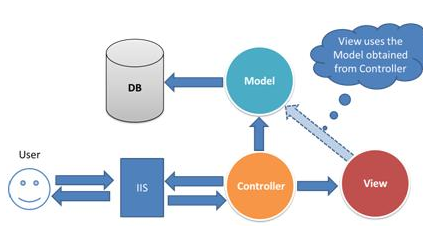
# GIỚI THIỆU VỀ ASP.NET MVC

## Giới thiệu về ASP.NET ASP.NET MVC

ASP.NET là một nền tảng phát triển ứng dụng web (web application framework) được phát triển và cung cấp bởi Microsoft, cho phép người lập trình tạo ra những trang web động, những ứng dụng web và những dịch vụ web. Lần đầu tiên được đưa ra thị trường vào tháng 2 năm 2002 cùng với phiên bản 1.0 của .NET framework, là công nghệ tiếp nối của Microsoft Active Server Page(ASP). ASP.NET được biên dịch dưới dạng Common Language Runtime(CLR), cho phép người lập trình viết mã ASP.NET với bất kỳ ngôn ngữ nào được hỗ trợ bởi .NET language.

Asp.Net MVC là một Framework sử dụng .Net Framework cho việc phát triển ứng dụng web động, trước khi Asp.Net MVC ra đời, lập trình viên sử dụng công nghệ Asp.Net Web Form trên nền tảng .Net Framework để phát triển ứng dụng Web động.  
Asp.Net MVC phát triển trên mẫu thiết kế chuẩn MVC, cho phép người sử dụng phát triển các ứng dụng phần mềm. **MVC** là tên một mẫu phát triển ứng dụng, phương pháp này chia nhỏ một ứng dụng thành ba thành phần để cài đặt, mỗi thành phần đóng một vai trò khác nhau và ảnh hưởng lẫn nhau, đó là models, views, và controllers.

## Mô hình MVC và cách thức hoạt dộng



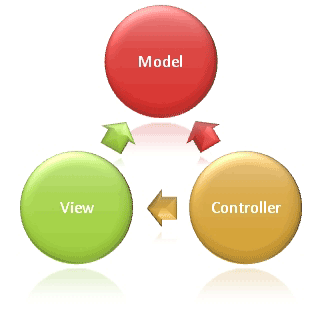
Hinh 1 Mô hình MVC và cách thức hoạt động

Dựa vào hình mô phỏng phía trên này thì mình sẽ 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.

### Khái niệm

Mô hình MVC là một chuẩn mô hình và đóng vai trò quan trọng trong quá trình xây dựng – phát triển – vận hành và bảo trì một hệ thống hay một ứng dụng – phần mềm. Nó tạo ra một mô hình 3 lớp Model – View – Controller tách biệt và tương tác nhau, giúp các chuyên gia có thể dễ dàng dựa vào mô hình để trao đổi và xử lý những nghiệp vụ một cách nhanh chóng. Đây là một mô hình đã xuất hiện từ những năm 70 của thế kỷ 20 tại phòng thí nghiệm Xerox PARC ở Palo Alto, nó không phụ thuộc vào môi trường, nền tảng xây dựng hay ngôn ngữ phát triển. Chúng ta có thể áp dụng mô hình MVC vào các dự án trong môi trường Windows, Linux… và sử dụng bất kỳ ngôn ngữ nào như PHP, ASP, JSP…



**Mô hình MVC**được chia làm 3 lớp xử lý gồm**Model – View – Controller :**

* **Model**: là nơi chứa những nghiệp vụ tương tác với dữ liệu hoặc hệ quản trị cơ sở dữ liệu (mysql, mssql… ); nó sẽ bao gồm các class/function xử lý nhiều nghiệp vụ như kết nối database, truy vấn dữ liệu, thêm – xóa – sửa dữ liệu…
* **View**: là nới chứa những giao diện như một nút bấm, khung nhập, menu, hình ảnh… nó đảm nhiệm nhiệm vụ hiển thị dữ liệu và giúp người dùng tương tác với hệ thống.
* **Controller** : là nới tiếp nhận những yêu cầu xử lý được gửi từ người dùng, nó sẽ gồm những class/ function xử lý nhiều nghiệp vụ logic giúp lấy đúng dữ liệu thông tin cần thiết nhờ các nghiệp vụ lớp **Model** cung cấp và hiển thị dữ liệu đó ra cho người dùng nhờ lớp **View**

### Sự tương tác giữa cáp lớp trong mô hình MVC.

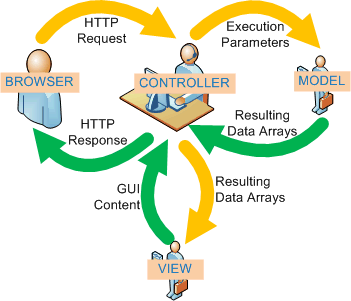
**Controller – View**sẽ lấy những hình ảnh, nút bấm…hoặc hiển thị dữ liệu được trả ra từ Controller để người dùng có thể quan sát và thao tác. Trong sự tương tác này cũng có thể không có dữ liệu được lấy từ Model và khi đó nó chỉ chịu trách nhiệm hiển thị đơn thuần như hình ảnh, nút bấm…

**Controller – Model** là luồng xử lý khi controller tiếp nhận yêu cầu và các tham số đầu vào từ người dùng, controller sẽ sử dụng các lớp/hàm trong Model cần thiết để lấy ra những dữ liệu chính xác.

**View – Model** có thể tương tác với nhau mà không qua Controller, nó chỉ đảm nhận hiển thị dữ liệu chứ không phải qua bất kỳ xử lý nghiệp vụ logics nào. Nó giống như các vùng dữ liệu hiển thị tĩnh trên các website như block slidebar…

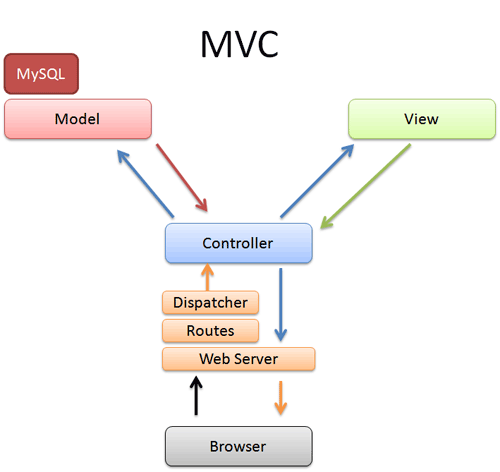
Mô hình MVC có thể áp dụng cho nhiều loại dự án khác nhau : dự án website, dự án ứng dụng – phần mềm,… nhưng trong phạm vi bài viết chúng ta sẽ chỉ đi tìm hiểu mô hình này được áp dụng như nào trong các dự án xây dựng và phát triển website.

**Quy trình hoạt động mô hình MVC trong dự án website**



  Mô hình trên là thể hiện mô hình MVC trong các dự án website và sẽ hoạt động theo quy trình sau:

1. Người dùng sử dụng một BROWSER trình duyệt web bất kỳ (Firefox, Chrome, IE,…) để có thể gửi những yêu cầu (HTTP Request) có thể kèm theo những dữ liệu nhập tới những CONTROLLER xử lý tương ứng. Việc xác định Controllerr xử lý sẽ dựa vào một bộ Routing điều hướng.
2. Khi CONTROLLER nhận được yêu cầu gửi tới, nó sẽ chịu trách nhiệm kiểm tra yêu cầu đó có cần dữ liệu từ MODEL hay không? Nếu có, nó sẽ sử dụng các class/function cần thiết trong MODEL và nó sẽ trả ra kết quả( Resulting Arrays), khi đó CONTROLLER sẽ xử lý giá trị đó và trả ra VIEW để hiển thị. CONTROLLER sẽ xác định các VIEW tương ứng để hiển thị đúng với yêu cầu.
3. Khi nhận được dữ liệu từ CONTROLLER, VIEW sẽ chịu trách nhiệm xây dựng các thành phẩn hiển thị như hình ảnh, thông tin dữ liệu… và trả về GUI Content để CONTROLLER  đưa ra kết quả lên màn hình BROWSER.
4. BROWSER sẽ nhận giá trị trả về( HTTP Response) và sẽ hiển thị với người dùng. Kết thúc một quy trình hoạt động.



### 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.

* Trình duyệt gửi yêu cầu của ứng dụng Asp.Net MVC
* MVC Engine chuyển yêu cầu tới cho Routing engine (Bộ điều hướng).
* Routing engine kiểm tra cấu hình điều hướng (route configuration) của ứng dụng nhằm chuyển đến các controller phù hợp với yêu cầu.
* Khi Controller được tìm thấy, control này sẽ được thực thi
* 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.
* 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.
* Controller yêu cầu View engine hiển thị thông tin dựa trên dữ liệu của tầng model.
* View engine trả kết quả cho controller
* Controller gửi kết quả cho trình duyệt thông qua giao thức Http.

## EntityFreamwork

### Khái niệm.

Entity Framework là một khuôn khổ ánh xạ quan hệ đối tượng (ORM) nguồn mở cho ADO.NET. nó là một phần của .NET Framework, nhưng kể từ phiên bản EntityFramework 6 nó được tách ra khỏi .NET framework.

EntityFramework là một tập hợp các công nghệ trong ADO.NET hỗ trợ phát triển các ứng dụng phần mềm hướng dữ liệu. Lập trình viên phát triển ứng dụng hướng dữ liệu thường phải vật lộn với nhu cầu đạt được hai mục tiêu rất khác nhau. Họ phải mô hình hóa các thực thể, mối quan hệ và logic của các vấn đề kinh doanh mà họ đang giải quyết, đồng thời họ cũng phải làm việc với các công cụ dữ liệu được sử dụng để lưu trữ và truy xuất dữ liệu. Dữ liệu có thể trải dài trên nhiều hệ thống lưu trữ, mỗi hệ thống có giao thức riêng; ngay cả những ứng dụng hoạt động với một hệ thống lưu trữ duy nhất cũng phải cân bằng các yêu cầu của hệ thống lưu trữ so với các yêu cầu viết mã ứng dụng hiệu quả và có thể bảo trì.

Khung thực thể cho phép các nhà phát triển làm việc với dữ liệu dưới dạng các đối tượng và thuộc tính miền cụ thể, chẳng hạn như khách hàng và địa chỉ khách hàng, mà không cần quan tâm đến 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, đồng thời có thể tạo và duy trì các ứng dụng hướng dữ liệu với ít mã hơn so với các ứng dụng truyền thống.

### Lịch sử phát triển EntityFreamwork.

Phiên bản đầu tiên của Entity Framework (EFv1) được bao gồm trong .NET Framework 3.5 Service Pack 1 và Visual Studio 2008 Service Pack 1, được phát hành vào ngày 11 tháng 8 năm 2008. Phiên bản này đã bị chỉ trích rộng rãi, thậm chí thu hút một 'phiếu bất tín nhiệm' được ký bởi khoảng một nghìn nhà phát triển.

Phiên bản thứ hai của Entity Framework, có tên Entity Framework 4.0 (EFv4), được phát hành như một phần của .NET 4.0 vào ngày 12 tháng 4 năm 2010 và giải quyết nhiều lời chỉ trích về phiên bản 1. Phiên bản thứ ba của Entity Framework, phiên bản 4.1, được phát hành vào ngày 12 tháng 4 năm 2011, với sự hỗ trợ của Code First.

Bản làm mới của phiên bản 4.1, có tên Entity Framework 4.1 Update 1, được phát hành vào ngày 25 tháng 7 năm 2011. Nó bao gồm các bản sửa lỗi và các loại được hỗ trợ mới.

Phiên bản 4.3.1 được phát hành vào ngày 29 tháng 2 năm 2012. Có một vài bản cập nhật, như hỗ trợ di chuyển.

Phiên bản 5.0.0 được phát hành vào ngày 11 tháng 8 năm 2012 và được nhắm mục tiêu vào .NET framework 4.5. Ngoài ra, phiên bản này có sẵn cho .Net framework 4, nhưng không có bất kỳ lợi thế thời gian chạy nào so với phiên bản 4.

Phiên bản 6.0 được phát hành vào ngày 17 tháng 10 năm 2013 và hiện là một dự án mã nguồn mở được cấp phép theo Giấy phép Apache v2. Giống như ASP.NET MVC , mã nguồn của nó được lưu trữ tại GitHub bằng Git. Phiên bản này có một số cải tiến để hỗ trợ mã ưu tiên .

Sau đó, Microsoft đã quyết định hiện đại hóa, cấu phần hóa và đưa .NET đa nền tảng lên Linux, OSX và các nơi khác, có nghĩa là phiên bản tiếp theo của Entity Framework sẽ là một bản viết lại hoàn chỉnh. [12] Vào ngày 27 tháng 6 năm 2016, bản này được phát hành dưới dạng Entity Framework Core 1.0, cùng với ASP.NET Core 1.0 và .NET Core 1.0. Ban đầu nó được đặt tên là Entity Framework 7, nhưng đã được đổi tên để làm nổi bật rằng nó là một bản viết lại hoàn chỉnh chứ không phải là một bản nâng cấp gia tăng và nó không thay thế EF6.

Entity Framework Core 1.0 được cấp phép theo Giấy phép Apache v2 và đang được xây dựng hoàn toàn mở trên GitHub . Mặc dù Entity Framework Core 1.0 chia sẻ một số điểm tương đồng về khái niệm với các phiên bản trước của Entity Framework, nhưng nó là một cơ sở mã hoàn toàn mới được thiết kế để hiệu quả hơn, mạnh mẽ, linh hoạt và có thể mở rộng, sẽ chạy trên Windows, Linux và OSX và sẽ hỗ trợ một phạm vi mới kho dữ liệu quan hệ và NOSQL. Entity Framework Core 2.0 được phát hành vào ngày 14 tháng 8 năm 2017 cùng với Visual Studio 2017 15.3 và ASP.NET Core 2.0 Entity Framework Core 3.0 được phát hành vào ngày 23 tháng 9 năm 2019 cùng với Visual Studio 2019 16.3 và ASP.NET Core 3.0, Cách sủ dụng EntityFreamwork.

### Ưu điểm nhược điểm của EntityFreamwork

* Ưu điểm
  + Tăng năng suất thông qua việc giảm lượng code.
  + Giảm sự phức tạp khi bảo trì phần mềm.
  + Entity Framework được đặt trong .NET framework và được tích hợp sẵn trong Visual Studio.
  + Không cần phải phân tích cú pháp đầu vào của dữ liệu trước khi xây dựng câu lệnh SQL nội tuyến.
* Nhược điểm
  + Không thể mở rộng nhiều cơ sở dữ liệu, ít nhất là không dễ dàng.
  + Tất cả các thực thể (bảng, view,…) cần một khóa chính.
  + Đôi khi các truy vấn SQL được viết kém và thực thi chậm.

## Ngôn ngữ thiết kế giao diện

### HTML

HTML (HyperText Markup Language – Ngôn ngữ đánh dấu siêu văn bản) là một ngôn ngữ đánh dấu được thiết kế ra để tạo nên các trang web với các mẩu thông tin được trình bày trên World Wide Web. Cùng với CSS và JavaScript, HTML tạo ra bộ ba nền tảng kỹ thuật cho các website.

HTML là cốt lõi của mọi trang web. Bất kể sự phức tạp của một trang web hoặc số lượng công nghệ liên quan. Đó là một kỹ năng thiết yếu cho bất kỳ chuyên gia web. Đó là điểm khởi đầu cho bất cứ ai học cách tạo nội dung cho web. Và thật may mắn cho những bạn mới bắt đầu là HTML rất dễ học.

Mỗi trang web được tạo thành từ một loạt các thẻ HTML này biểu thị từng loại nội dung trên trang. Mỗi loại nội dung trên trang được “bọc”, tức là được bao quanh bởi các thẻ HTML.

Sử dụng HTML, bạn có thể thêm tiêu đề, định dạng đoạn văn, ngắt dòng điều khiển. Tạo danh sách, nhấn mạnh văn bản, tạo ký tự đặc biệt, chèn hình ảnh, tạo liên kết. Hoặc xây dựng bảng, điều khiển một số kiểu dáng và nhiều hơn thế nữa.

HTML không phải là ngôn ngữ lập trình, đồng nghĩa với việc nó không thể tạo ra các chức năng “động” được. Nó chỉ giống như Microsoft Word, dùng để bố cục và định dạng trang web.

### CSS

CSS là viết tắt của Cascading Style Sheets. Ngôn ngữ lập trình này chỉ ra cách các yếu tố HTML của trang web thực sự sẽ xuất hiện trên giao diện của trang. Nếu HTML cung cấp các công cụ thô cần thiết để cấu trúc nội dung trên một trang web thì CSS sẽ giúp định hình kiểu nội dung này để trang web xuất hiện trước người dùng theo một cách đẹp hơn. Bạn có thể hiểu là nếu HTML là tường gạch thô thì CSS là sơn để trang trí cho tường gạch đó. Các ngôn ngữ này được giữ riêng biệt để đảm bảo các trang web được xây dựng chính xác trước khi chúng được định dạng lại.

Trong khi HTML là cấu trúc cơ bản của trang web của bạn. CSS mang lại cho trang web của bạn phong cách mà bạn muốn.  Những màu sắc đặc trưng, font chữ phù hợp, và hình ảnh nền của website? Tất cả là nhờ CSS. CSS gần như tạo nên bộ mặt của một website. Và nếu trang web của bạn ưa nhìn thì nó sẽ hấp dẫn và lôi cuốn được người dùng phải không.

Để có một trang web đẹp bạn không chỉ dựa vào CSS. Mà các ngôn ngữ phải được thực hiện dựa trên sự sáng tạo và các bản thiết kế bạn tạo ra trước đó. Ngoài ra, bạn cần phải kết hợp nó với nhiều hơn các ngôn ngữ khác nữa.

### Bootstrap Freamwork

Bootstrap là một framework cho phép thiết kế website reponsive nhanh hơn và dễ dàng hơn

Bootstrap là bao gồm các HTML templates, CSS templates và Javascript tạo ra những cái cơ bản có sẵn như: typography, forms, buttons, tables, navigation, modals, image carousels và nhiều thứ khác. Trong bootstrap có thêm các plugin Javascript trong nó. Giúp cho việc thiết kế reponsive của bạn

## Microsoft SQL Server

### Khái niệm về Microsoft SQL server

**SQL** là viết tắt của **Structured Query Language** là ngôn ngữ truy vấn có cấu trúc. Nó là một ngôn ngữ, là tập hợp các lệnh để tương tác với cơ sở dữ liệu. Dùng để lưu trữ, thao tác và truy xuất dữ liệu được lưu trữ trong một cơ sở dữ liệu quan hệ. Trong thực tế, SQL là ngôn ngữ chuẩn được sử dụng hầu hết cho hệ cơ sở dữ liệu quan hệ. Tất cả các hệ thống quản lý cơ sở dữ liệu quan hệ (RDMS) như MySQL, MS Access, Oracle, Postgres và SQL Server… đều sử dụng SQL làm ngôn ngữ cơ sở dữ liệu chuẩn.

### Chức năng

* Cho phép truy cập dữ liệu trong các hệ thống quản lý cơ sở dữ liệu quan hệ.
* Cho phép mô tả dữ liệu.
* Cho phép xác định dữ liệu trong cơ sở dữ liệu và thao tác dữ liệu đó.
* Cho phép nhúng trong các ngôn ngữ khác sử dụng mô-đun SQL, thư viện và trình biên dịch trước.
* Cho phép tạo và thả các cơ sở dữ liệu và bảng.
* Cho phép tạo chế độ view, thủ tục lưu trữ, chức năng trong cơ sở dữ liệu.
* Cho phép thiết lập quyền trên các bảng, thủ tục và view.

### Ưu điểm

* Không cần code: Rất dễ dàng để quản lý các hệ thống cơ sở dữ liệu bằng việc sử dụng SQL chuẩn mà không cần phải viết bất cứ dòng code nào.
* Tính di động: SQL có thể được sử dụng trong chương trình trong PCs, servers, laptops, và thậm chí cả mobile phones.
* Thao tác dễ dàng: Bạn chỉ cần thực hiện một số thao tác với dữ liệu đơn giản trên SQL thay vì phải dùng nhiều câu lệnh phức tạp trên các loại ngôn ngữ khác.

Ngôn ngữ tương tác: Language này có thể được sử dụng để giao tiếp với cơ sở dữ liệu và nhận câu trả lời cho các câu hỏi phức tạp trong vài giây.

### Nhược điểm

* Giao diện khó dùng: SQL có giao diện phức tạp khiến một số người dùng khó truy cập.
* Tính thực thi: Hầu hết các chương trình cơ sở dữ liệu SQL đều có phần mở rộng độc quyền riêng của nhà cung cấp bên cạnh các tiêu chuẩn SQL.
* Giá cả: Chi phí vận hành của một số phiên bản SQL khiến một số lập trình viên gặp khó khăn khi tiếp cận.
* Không được toàn quyền kiểm soát: Các lập trình viên sử dụng SQL không có toàn quyền kiểm soát cơ sở dữ liệu do các quy tắc nghiệp vụ ẩn.

# Phân tích thiết kế cơ sở dữ liệu và giao diện webiste

## Đặt tả hệ thống.

Sau khi khảo sát thực tế và xác định được các yêu cầu của hệ thống, có thể phân tích để đưa ra các quy trình nghiệp vụ chung của hệ thống bao gồm các phần:

Các nghiệp vụ chung được phân tích chi tiết

* Về phía trang người dùng

+Thời gian thực hiện: Khi người dùng vào trang web, thông tin về website sẽ được hiển thị tại trang chủ của hệ thống. Hiển thị những tính năng để cho người dùng thao tác với website như xem tin tức, xem sản phẩm, tạo liên hệ, đặt hàng…

+Tác nhân thực hiện: Người dùng, khách hàng.

* Về phía trang quản trị

+Thời gian thực hiện: Khi người quản trị truy cập vào hệ thống dành cho người quản trị thực hiện

+Tác nhân thực hiện: người quản trị hệ thống

+Nhiệm vụ: Quản lý tin tức, sản phẩm, bài viết, liên hệ, đơn hàng, ……

## Danh sách cá tác nhân.

|  |  |  |
| --- | --- | --- |
| STT | Tên tác nhân | Ý nghĩa |
| 1 | admin | L là người quản lí chung mọi hoạt động của website và tương tác với hệ thống: sản phẩm, quản lí liên hệ, thống kê báo cáo, tra cứu, thay đổi các quy định. |
| 2 | Khách hàng, người dùng | Là người sẽ tương tác với các nghiệp vụ như: xem bài viết, xem sản phẩm, gửi liên hệ… |

Tác nhân khách hàng.

|  |  |  |
| --- | --- | --- |
| STT | Tên use case | Ghi chú |
| 1 | Xem sản phẩm | Use case mô tả người sử dụng có nhu cầu muốn xem các sản phẩm có trên website. |
| 2 | Tạo liên hệ | Use case mô tả người dùng muốn gửi liên hệ đến website. |
| 3 | Mua hàng | Use case mô tả người dùng mua hàng tại website. |
| 4 | Xem tin tức | Use case mô tả người dùng xem tin tức tại website |

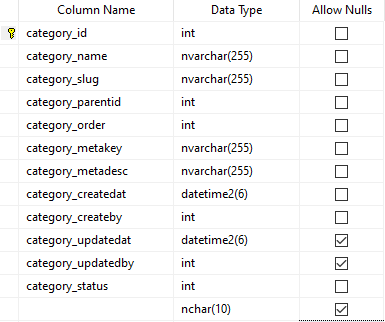
Tác nhân Admin.

|  |  |  |
| --- | --- | --- |
| STT | Tên use case | Ghi chú |
| 1 | Đăng nhập | Use case này giúp cho người quản trị sư dụng các chức năng của hệ thống cần đến quyền truy cập |
| 2 | Quản lý sản phẩm | Use case này mô tả chức năng cập nhật thông tin sản phẩm vào hệ thống của Admin. Khi thông tin của một sản phẩm thay đổi thì Admin là người sẽ cập nhật những thông tin đó vào hệ thống. |
| 3 | Quản lý loại sản phẩm | Use case mô tả chức năng xem, thêm xóa sửa các loại sản phẩm. |
| 4 | Quản lý tin | Use case mô tả chức năng quản lý bài viết (thêm, xóa, sữa,cập nhật) của admin |
| 5 | Quản lý chủ đề bài viết | Use case mô tả chức năng quản lý chủ đề bài viết (thêm, xóa, sữa,cập nhật) của admin |
| 6 | Quản lý slider | Use case mô tả chức năng quản lý slide hình ảnh (thêm, xóa, sữa,cập nhật) của admin |
| 7 | Quản lý user | Use case mô tả chức năng quản lý user thành viên (thêm, xóa, sữa,cập nhật) của admin |
| 8 | Quản lý đơn hàng | Use case mô tả chức năng quản lí các đơn đặt hàng (xem, xử lý, xóa) của admin. |
| 9 | Quản lý menu | Use case mô tả chức năng quản lí các menu (thêm, xóa, sửa) của admin. |

## Các bảng dữ liệu

### Bảng Category

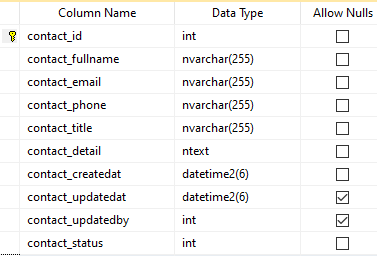
Bảng category chứa các thông tin về loại sản phẩm có trong website.



Hình 1: Bảng Category

### Bảng Contact

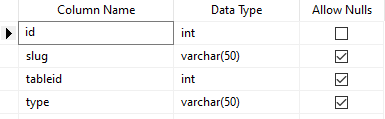
Bảng Contact là bảng chứa thông tin mà khác hàng gửi tới website



Hình 2: Bảng contact

### Bảng Link

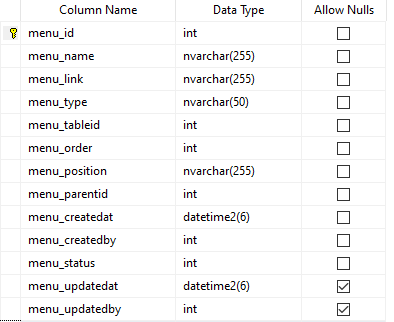
Bảng chứa các thông tin về đường dẫn URL có trong website.



Hình 3: Bảng link

### Bảng Menu

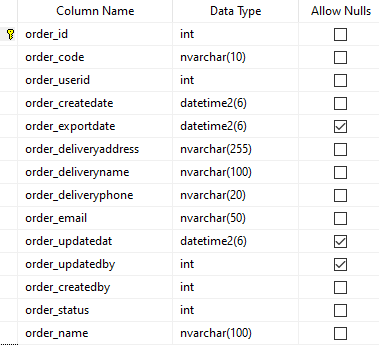
Bảng menu chứa các thông tin về menu có trong website.



Hình 4: Bảng menu

### Bảng Order

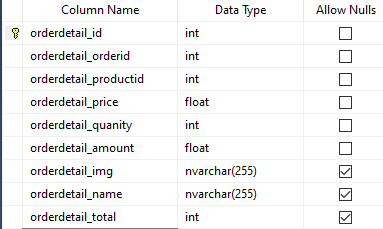
Chứa các thông tin về đơn đặt hàng của website.



Hình 5: Bảng Order

### Bảng OrderDetail

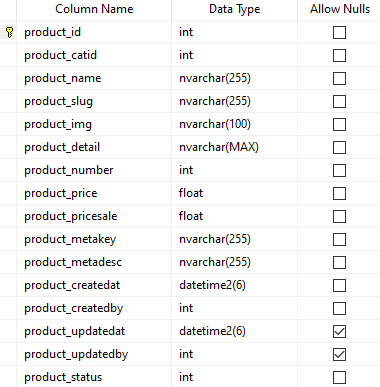
Chứa các thông tin chi tiết về các đơn đặt hàng của website.



Hình 6: Bảng OrderDetail

### Bảng Product

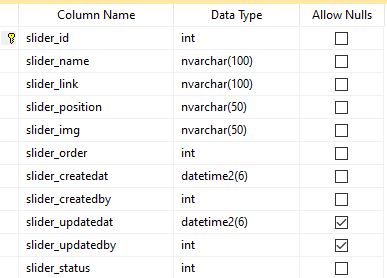
Chứa các thông tin liên quan đến sản phẩm có trong website.



Hình 7: Bảng Product

### Bảng Slider

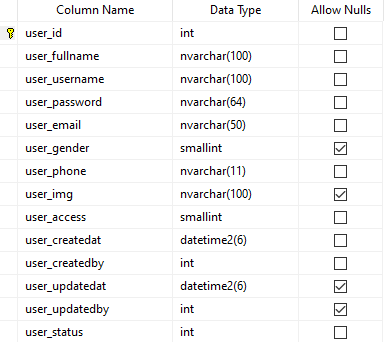
Bảng chứa các thông tin về các slider hình ảnh có trong website.



Hình 8: Bảng Slider

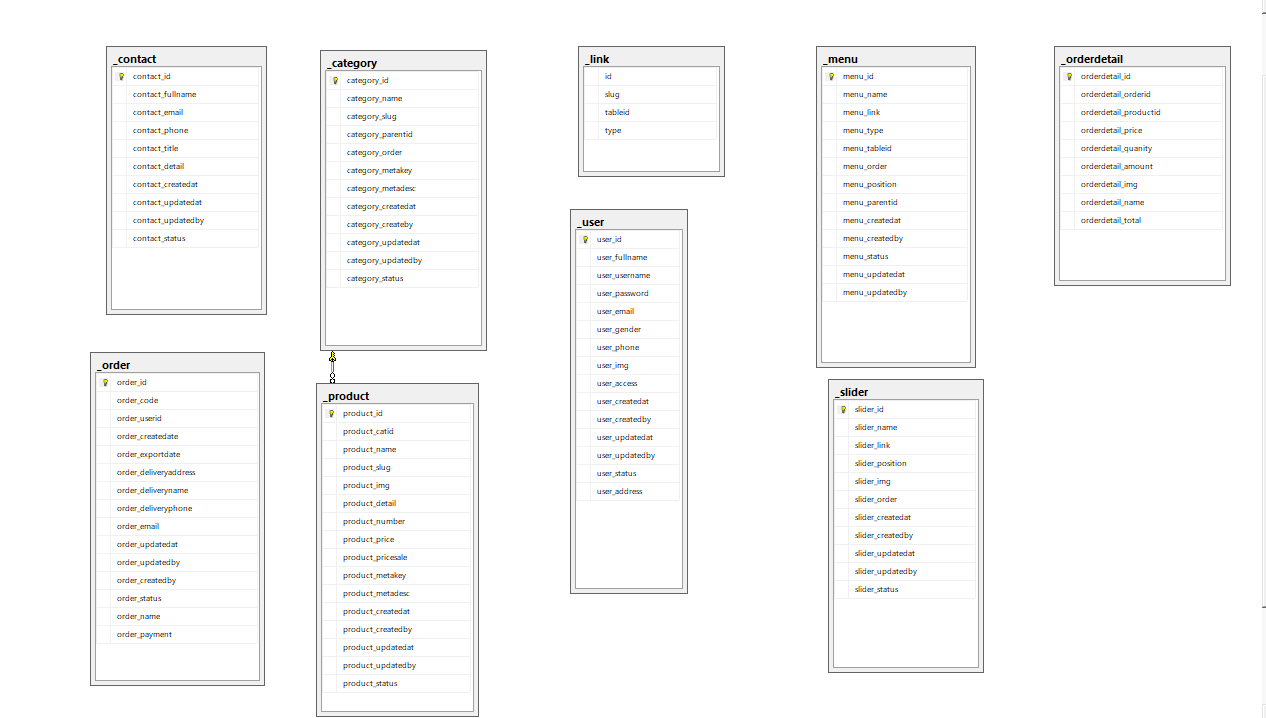
### Bảng User

Chứa các thông tin về người dùng và quản trị viên website.



Hình 9: Bảng User

## Database Diagram



Hình 10: Database Diagram

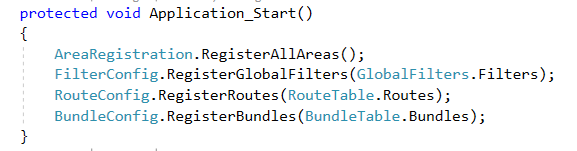
## Giao diện người dùng

Giao diện trang chủ

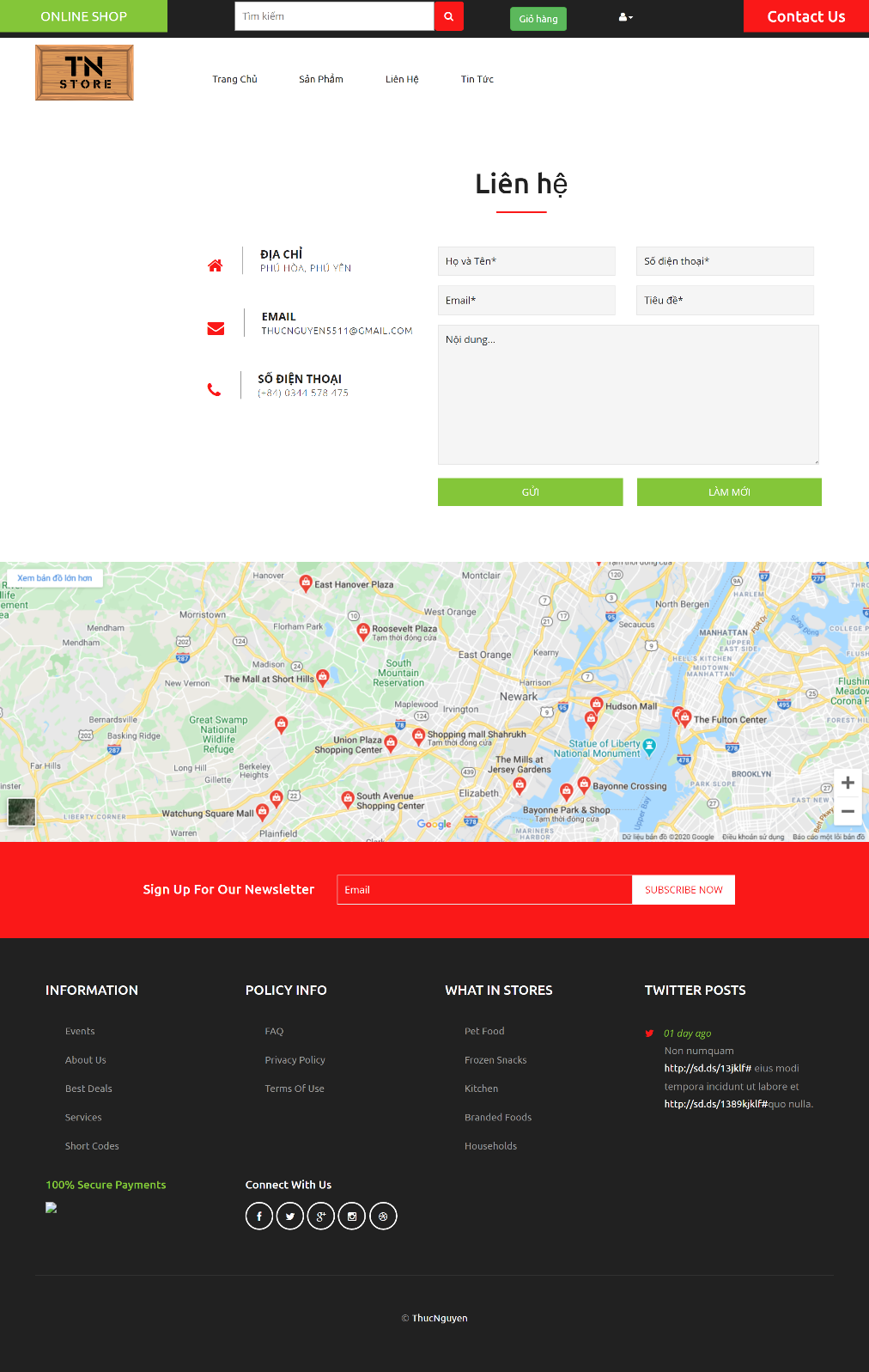


Khi người dùng truy cập website và thực hiện yêu cầu truy cập đến đường một URL, URL routing sẽ được chỉ định yêu cầu tới Controller Action.URL routing sử dụng một bảng định tuyến để điều khiển các yêu cầu. Bảng định tuyến được tạo khi ứng dụng được chạy lần đầu tiên. Bảng định tuyến được thiết lập trong file Global.aspx.

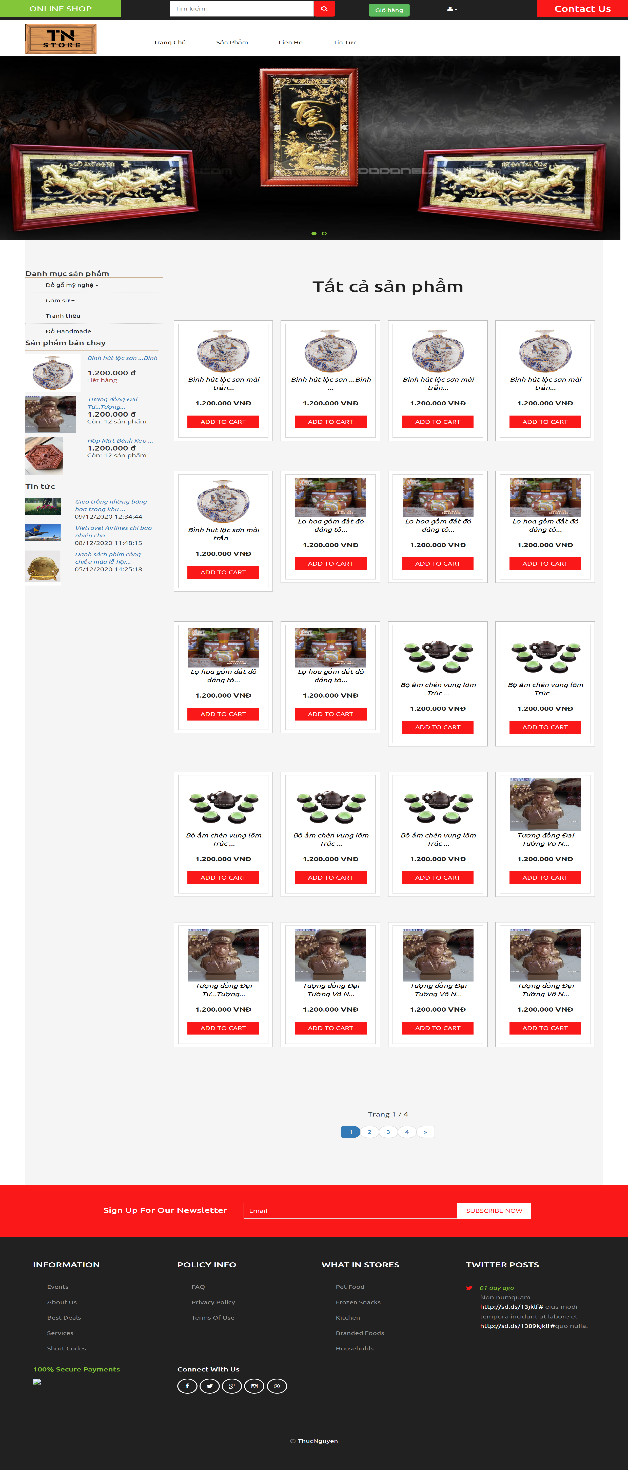
Khi ứng dụng chạy lần đầu tiên, phương thức Application\_ Start () được gọi. Phương thức này gọi một phương thức khác RegisterRouter (RouteTable.Router) để tạo bảng định tuyến.



Giao diện trang liên hệ



Giao diện trang sản phẩm



Giao diện trang tin tức

Giao diện trang sản phẩm theo loại

Giao diện trang sản phẩm khuyến mãi

Giao diện trang tìm kiếm sản phẩm

Giao diện trang chi tiết sản phẩm

Giao diện trang chi tiết bài viết

Giao diện trang chủ đề bài viết

Giao diện trang giỏ hàng

Giao diện trang đăng nhập

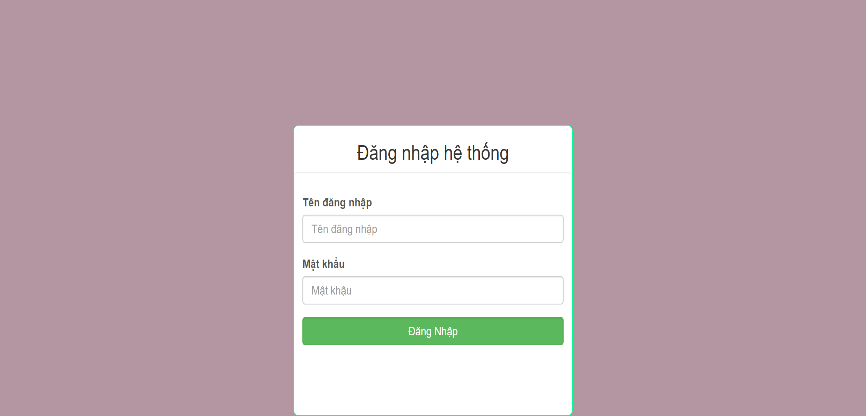
Giao diện trang đăng kí

Giao diện trang profile

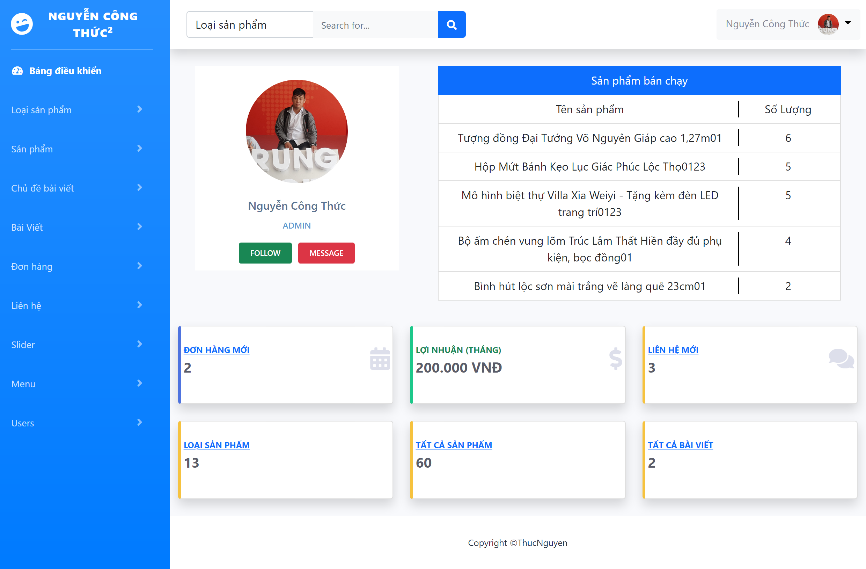
Giao diện trang lịch sử mua hàng

## Giao diện trang quản trị

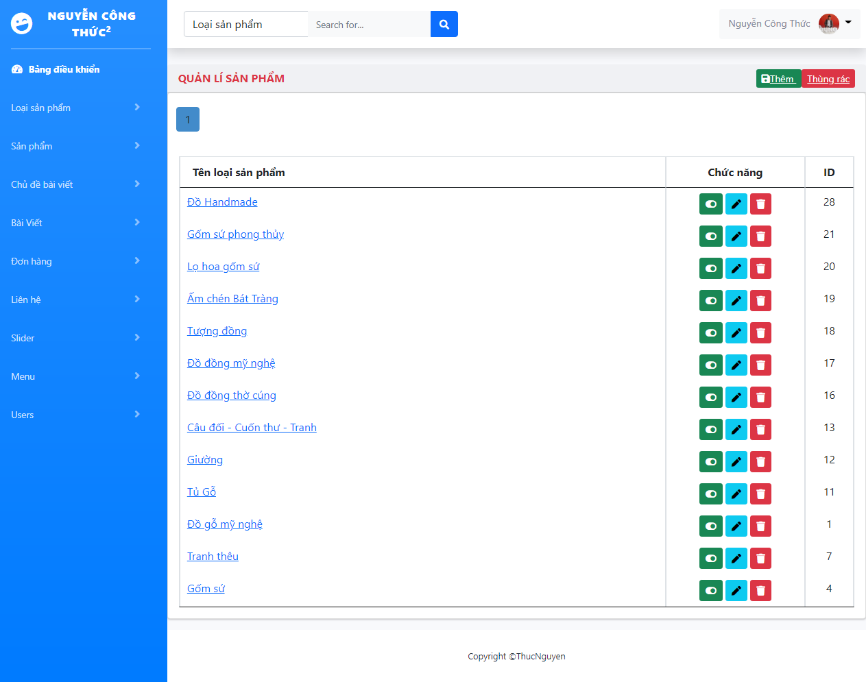
Giao diện trang đăng nhập



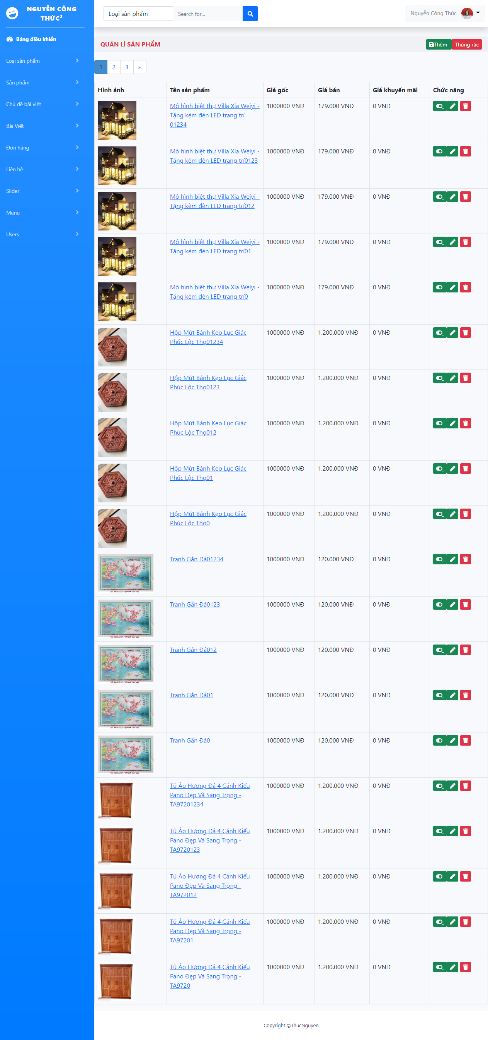
Giao diện trang bảng điều khiển



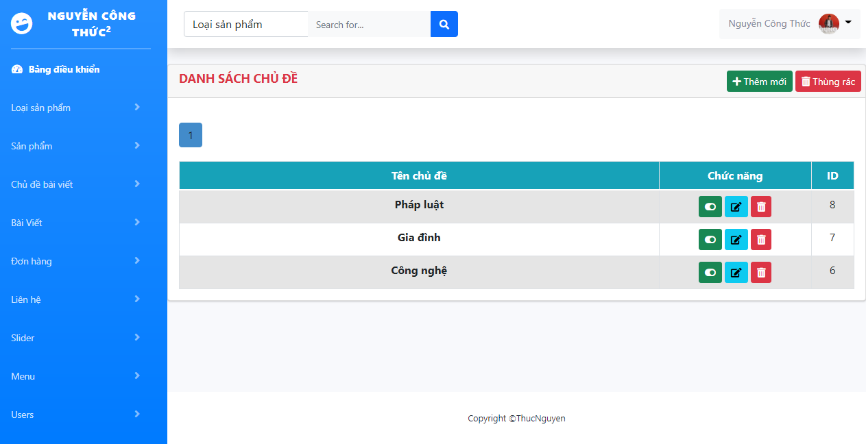
Giao diện trang quản lí sản phẩm



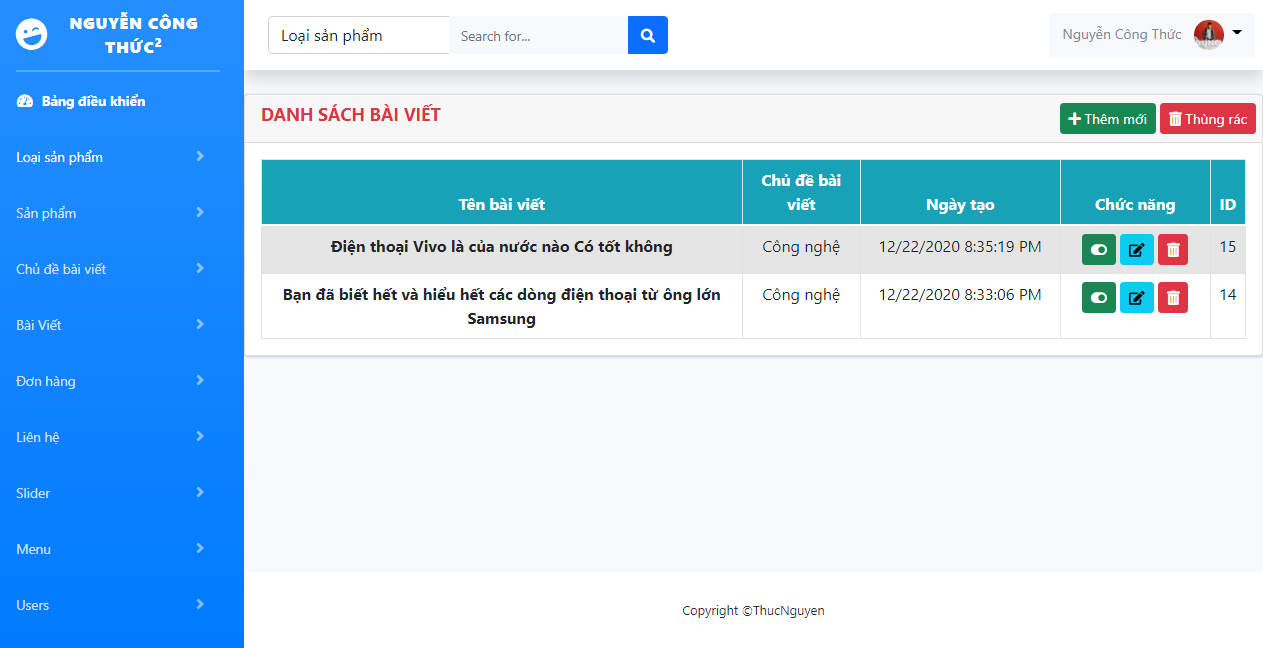
Giao diện trang quản lí loại sản phẩm



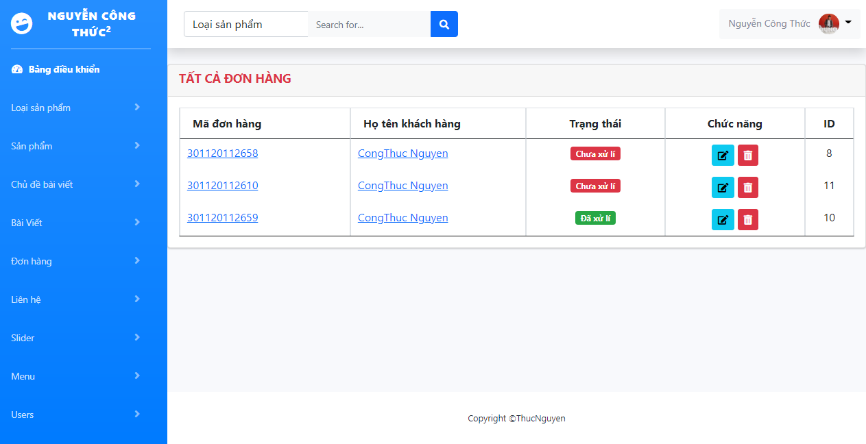
Giao diện trang quản lí chủ đề bài viết



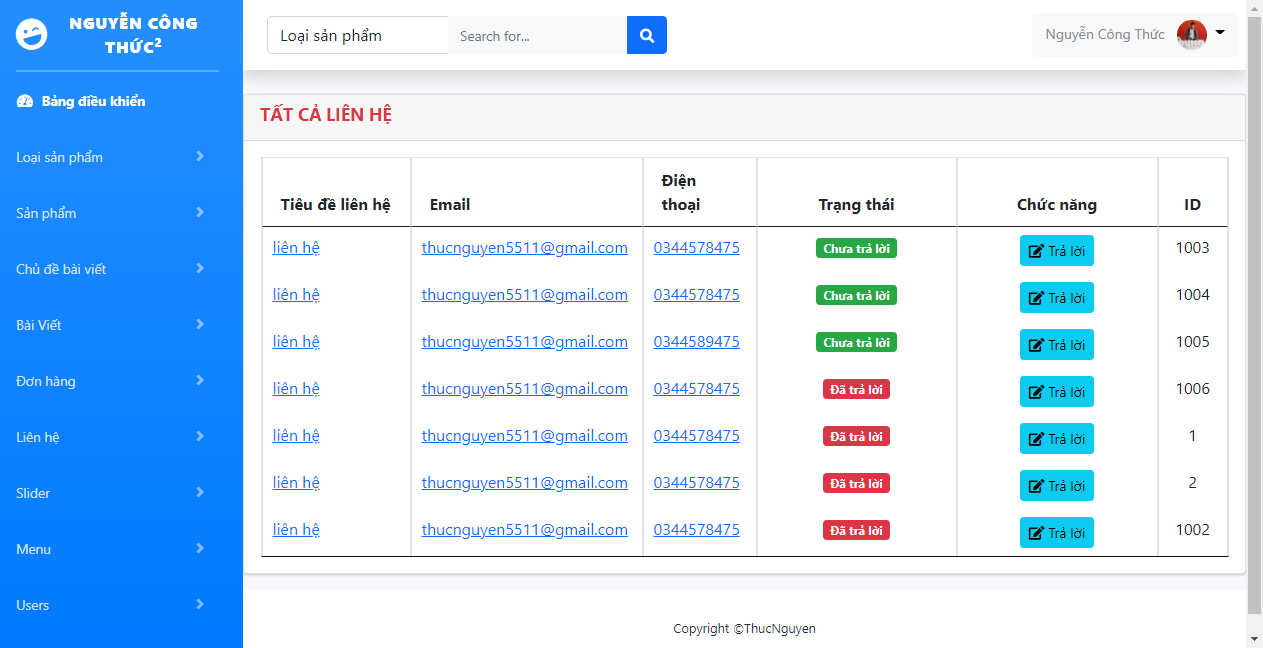
Giao diện trang quản lí bài viết



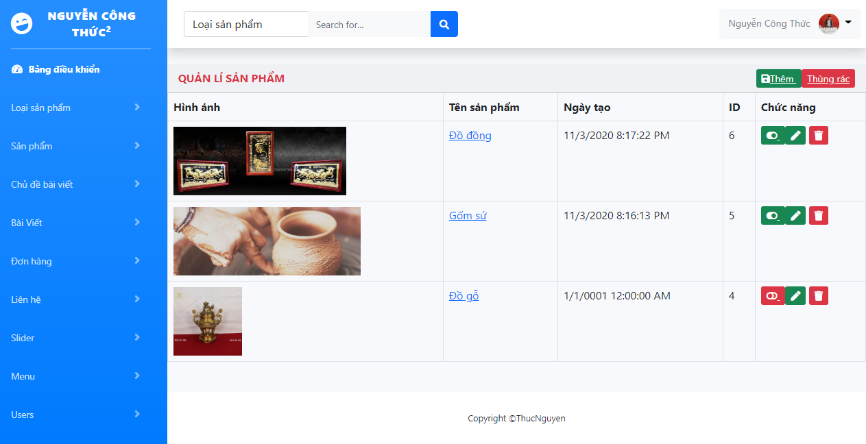
Giao diện trang quản lí đơn hàng



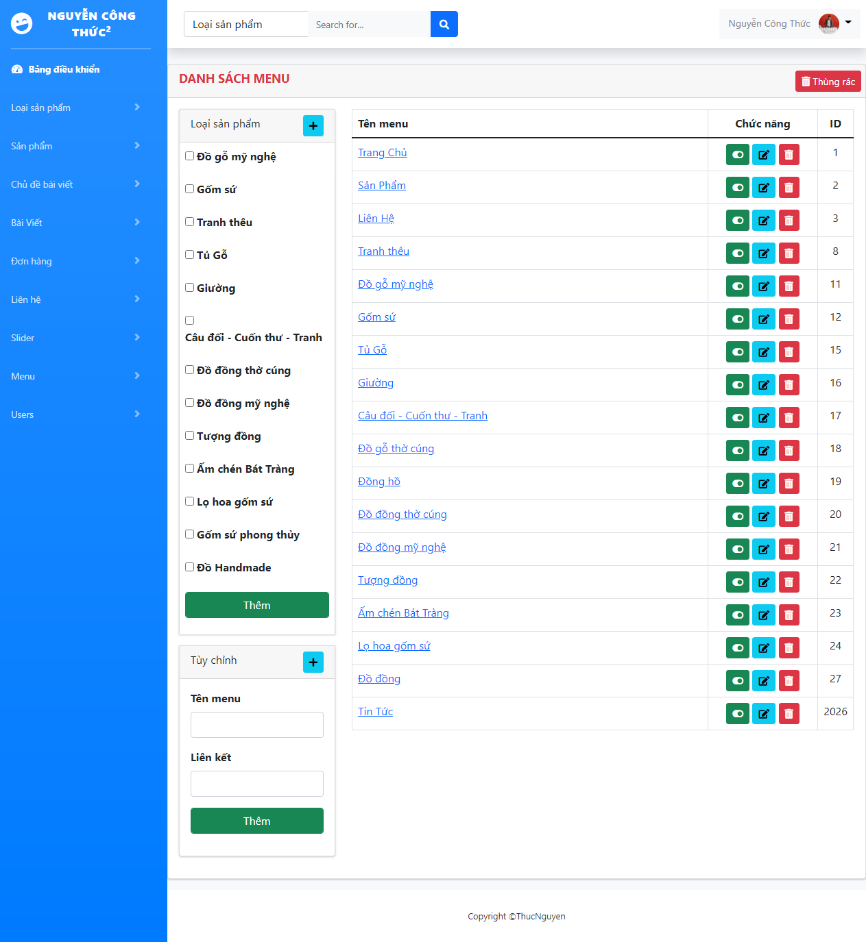
Giao diện trang quản lí liên hệ



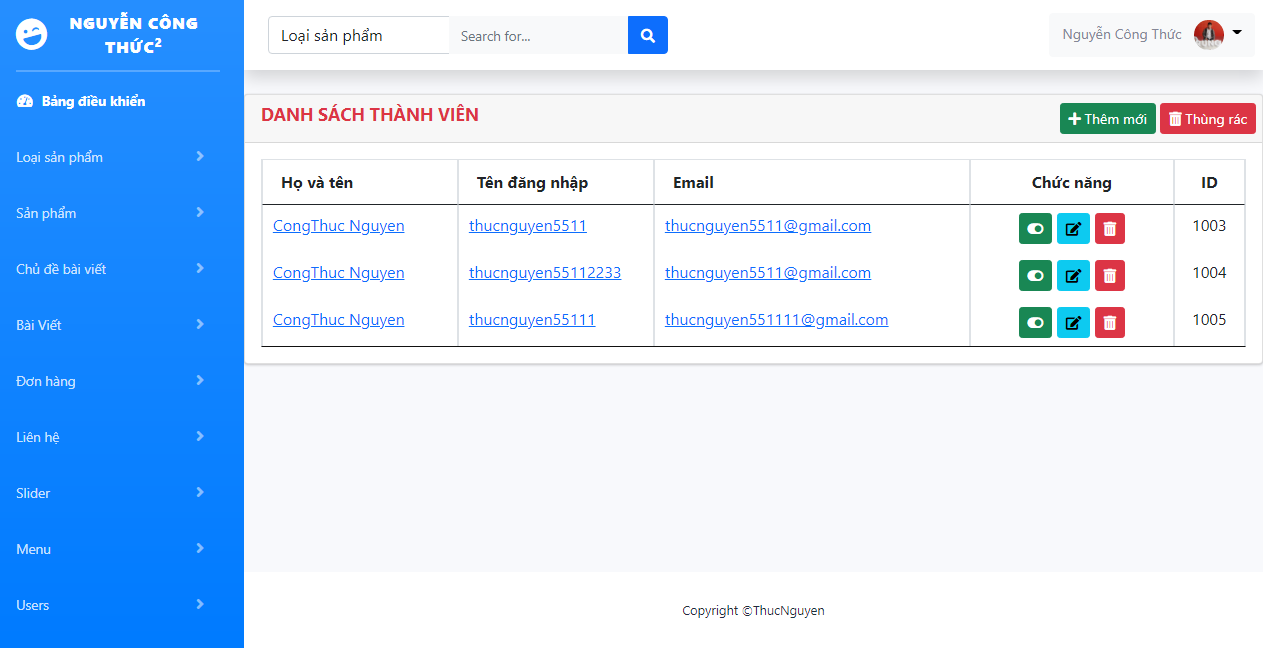
Giao diện trang quản lí slider



Giao diện trang quản lí menu



Giao diện trang quản lí user



# Đánh giá kết quả và hướng phát triển

## Kết quả đạt được

* được hoàn thành các chức năng và yêu cầu đặt ra.
* Hiểu biết thêm về ngôn ngữ lập trình ASP.NET MVC
* Nâng cao được kĩ năng của bản thân trong quá trình phát triển web.

## Hạn chế

* Các chức năng còn đơn giản.
* Wesbite không được đẹp mắt.

## Hướng phát triển

* Phát triển đề tài với quy mô rộng lớn hơn
* Tối ưu hóa các chức năng để đáp ứng nhu cầu của người sử dụng.

# TÀI LIỆU THAM KHẢO

* <https://docs.microsoft.com/vi-vn/aspnet/mvc/overview/getting-started/introduction>
* <http://stackoverflow.com>