



**TRƯỜNG ĐẠI HỌC QUỐC TẾ SÀI GÒN**  
THE SAIGON INTERNATIONAL UNIVERSITY

**KHOA CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO NGHIÊN CỨU KHOA HỌC**

**XÂY DỰNG WEBSITE ĐÁNH GIÁ CÔNG  
NGHỆ TRÊN NỀN TẢNG ASP.NET MVC 5**

Giảng viên hướng dẫn: KS. Nguyễn Hoàng Duy

Sinh viên thực hiện:

- Lê Hoàng Long – 91011701336
- Ngô Minh Thành – 91011701337

TP. Hồ Chí Minh, 2020

# MỤC LỤC

CHƯƠNG 1: TỔNG QUAN .....	1
1.1    Giới thiệu đề tài.....	1
1.2    Nhiệm vụ đồ án .....	1
1.3    Sơ đồ cấu trúc đề tài.....	2
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT.....	3
2.1.    Tổng quan về C# .....	3
2.1.1.    Giới thiệu về C# .....	3
2.1.2.    Hình thành.....	4
2.1.3.    Các phiên bản .....	4
2.1.4.    Tính năng của C#.....	5
2.1.5.    Các tính năng mới.....	6
2.2.    Tổng quan về .NET FRAMEWORK.....	8
2.2.1.    Giới thiệu về .NET FRAMEWORK .....	8
2.2.2.    Lịch sử phát triển .....	9
2.2.3.    Các tính năng mới.....	9
2.3.    Tổng quan về ASP.NET MVC 5 .....	12
2.3.1.    Giới thiệu về ASP.NET MVC 5 .....	12
2.3.2.    Ưu điểm của mô hình MVC.....	13
2.3.3.    Tính năng của nền tảng MVC .....	13
2.4.    Tổng quan về HTML .....	14

2.4.1. Khái niệm .....	14
2.4.2. Cách HTML được xử lý.....	15
2.4.3. Vai trò của HTML trong website .....	15
2.5. Tổng quan về CSS.....	16
2.5.1. Khái niệm .....	16
2.5.2. Phương thức hoạt động.....	16
2.6. Tổng quan về Javascript.....	16
2.6.1. Khái niệm .....	16
2.6.2. Sự phổ biến của Javascript .....	17
2.6.3. Ứng dụng của Javascript .....	17
2.7. Tổng quan về Bootstrap .....	18
2.7.1. Khái niệm.....	18
2.7.2. Ưu điểm .....	18
<b>CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ.....</b>	<b>20</b>
3.1. Phân tích.....	20
3.1.1. Mô tả chương trình .....	20
3.1.2. Ý tưởng .....	20
3.1.3. Đối tượng người dùng .....	20
3.1.4. Ràng buộc tổng thể.....	20
3.1.5. Mô hình thực thể kết hợp.....	21
3.2. Cơ sở dữ liệu .....	22
<b>CHƯƠNG 4: KẾT QUẢ THỰC NGHIỆM .....</b>	<b>24</b>
4.1. Giao diện trang chủ .....	24
4.2. Giao diện chọn bài viết .....	25

4.3.	Giao diện bài viết .....	26
4.4.	Giao diện bình luận .....	26
<b>CHƯƠNG 5: KẾT LUẬN .....</b>		<b>28</b>
5.1	Kết quả đạt được .....	28
5.2	Đánh giá Web.....	29
5.2.1	<i>Ưu điểm</i> .....	29
5.2.2	<i>Nhược điểm</i> .....	29
5.3	Hướng phát triển và mở rộng đề tài .....	29
5.4	Tài liệu tham khảo.....	29

## LỜI CẢM ƠN

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

Em xin gửi lời cảm ơn chân thành đến thầy Nguyễn Hoàng Duy, giảng viên khoa Công Nghệ Thông Tin - Trường Đại Học Quốc Tế Sài Gòn, người đã tận tình hướng dẫn, chỉ bảo em trong suốt quá trình thực hiện đề tài.

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

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

TP.HCM, Ngày....Tháng.... Năm....

Lê Hoàng Long

Ngô Minh Thành

# CHƯƠNG 1: TỔNG QUAN

## 1.1 Giới thiệu đề tài

Trong thời đại ngày nay công nghệ thông tin đang phát triển rất vượt trội, nó làm thay đổi cuộc sống chúng ta một cách đáng kinh ngạc điển hình như cách chúng ta đi lại bằng xe ôm công nghệ, đặt phòng khách sạn hay thậm chí là nhà thông minh. Song song đó nhu cầu về tìm hiểu về công nghệ cũng từ đó mà dần dần hình thành.

Từ nhu cầu thực tế và xu hướng của thời đại đề tài này ra đời nhằm mục đích xây dựng một ASP.NET MVC mà tại đó có thể cho phép người dùng đánh giá, tìm hiểu về những công nghệ tiên tiến, món ăn, địa danh, phim, hình ảnh, thời trang,.....cũng như nhận được những thông tin hữu ích hoặc đưa ra những thông tin thiết yếu để giúp những người dùng khác biết được những đề tài mới. Thông qua đó giảm thiểu tối đa thời gian cho việc tìm tòi và ngồi chờ đợi cũng như chi phí đi lại.

Các chương tiếp theo sẽ giới thiệu về ASP.NET MVC bao gồm các đặc điểm, tính năng, công cụ và thành phần trong framework này. Sau những nội dung về thiết kế, tài liệu sẽ trình bày về việc lập trình trên ASP.NET MVC khi giới thiệu về ngôn ngữ lập trình C#. Cuối cùng là giới thiệu về giao diện, chức năng, tài liệu hướng dẫn sử dụng của phần mềm cùng chương tổng kết về các khó khăn và những thành quả trong suốt quá trình phát triển ASP.NET MVC.

## 1.2 Nhiệm vụ đồ án

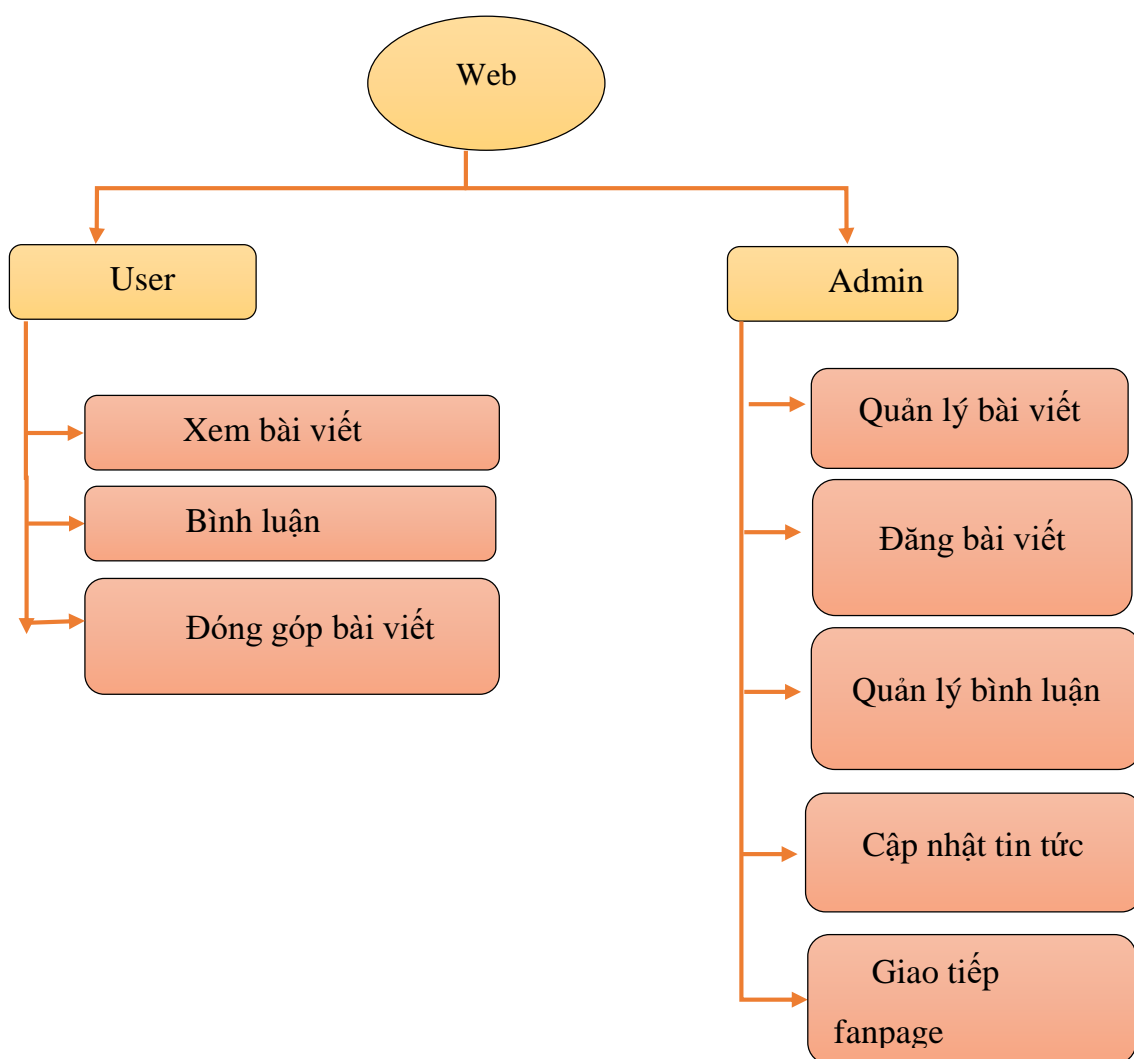
Web được tạo ra trên nền tảng ASP.NET MVC nhằm mục đích phục vụ cho nhu cầu tìm hiểu về những công nghệ mới hoặc cũ. Web có thể là nơi vừa đánh giá, giúp mọi người trải nghiệm những công nghệ tiên tiến dễ dàng hơn .

Người sử dụng sẽ có thể truy cập ASP.NET MVC trên tất cả các thiết bị bao gồm smartphone, máy tính bảng, laptop với các kích thước màn hình khác nhau. Họ có thể nhanh chóng trò chuyện với những người khác và chia sẻ bất kỳ thông tin về công nghệ trên nền tảng mới đảm bảo tính bảo mật thông tin về người dùng.

Chức năng đánh giá cho phép người dùng tìm và chọn các thông tin muốn tìm hiểu, tham khảo mức giá trung bình và sau khi chọn sẽ đến phần đánh giá để đưa ra những bình luận của riêng mình.

Chức năng đăng ký giúp người dùng tạo tài khoản của mình ngay trên ASP.NET MVC mà không cần có thêm bất cứ giấy tờ vật lý nào khác, chỉ cần nhập số điện thoại/gmail là họ có thể tạo một tài khoản của riêng mình.

### 1.3 Sơ đồ cấu trúc đề tài



Hình 1.1 Sơ đồ cấu trúc trang web.

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

### 2.1. Tổng quan về C#

#### 2.1.1. Giới thiệu về C#

Như ta đã biết thì ngôn ngữ C# dựng trên nền tảng C++ và Java nên ngôn ngữ C# khá đơn giản. Nếu chúng ta thân thiện với C và C++ hoặc thậm chí là Java, chúng ta sẽ thấy C# khá giống về diện mạo, cú pháp, biểu thức, toán tử và những chức năng khác được lấy trực tiếp từ ngôn ngữ C và C++, nhưng nó đã được cải tiến để làm cho ngôn ngữ đơn giản hơn. Một vài trong các sự cải tiến là loại bỏ các dư thừa, hay là thêm vào những cú pháp thay đổi.

C# là một ngôn ngữ lập trình đơn giản, hiện đại, mục đích tổng quát, hướng đối tượng được phát triển bởi Microsoft và được phê chuẩn bởi European Computer Manufacturers Association (ECMA) và International Standards Organization (ISO).

C# được phát triển bởi Anders Hejlsberg và team của ông trong khi phát triển .Net Framework. C# được thiết kế cho Common Language Infrastructure (CLI), mà gồm Executable Code và Runtime Environment, cho phép chúng ta sử dụng các ngôn ngữ high-level đa dạng trên các nền tảng và cấu trúc máy tính khác nhau.

Dưới đây là các lý do làm C# là ngôn ngữ lập trình chuyên nghiệp được sử dụng rộng rãi:

- Nó là một ngôn ngữ lập trình hiện đại, mục đích tổng quát.
- Nó là hướng đối tượng.
- Nó dễ dàng để học.
- Nó là một ngôn ngữ được cấu trúc.
- Nó tạo các chương trình hiệu quả.
- Nó có thể được biên dịch trên nhiều nền tảng máy tính khác nhau.
- Nó là một phần của .Net Framework.



Cấu trúc C# khá gần với các ngôn ngữ high-level truyền thống, C và C++, và là một ngôn ngữ lập trình hướng đối tượng. Nó có sự giống nhau mạnh mẽ với Java, nó có nhiều đặc điểm lập trình mạnh mẽ mà làm cho nó trở nên ưa thích với các lập trình viên trên toàn thế giới.

### 2.1.2. Hình thành

Cái tên C# được lấy cảm hứng từ ký hiệu âm nhạc, trong đó có một dấu thăng sau nốt nhạc “#” nghĩa là một nốt chơi cao hơn nửa cung. Điều này tương tự như trường hợp đặt tên của C++. Trong đó “++” chỉ ra rằng giá trị của một biến được tăng lên 1. Biểu tượng # cũng giống với 4 ký tự ++ (trong một lưới 2x2), ngụ ý rằng ngôn ngữ là một phiên bản tăng cường của C++ (gấp đôi C++). Bởi vì giới hạn kỹ thuật của việc hiển thị (các font chuẩn, trình duyệt,...) và sự thật là ký tự thăng (U+266F **MUSIC SHARP SIGN** (HTML **&#9839;**)) không hiện diện trong đa số các bàn phím, ký tự (U+0023 **# NUM SIGN** (HTML **&#35;**)) đã được chọn để diễn đạt một cách tương đương trong cách viết tên ngôn ngữ.

Hậu tố “#” đã được sử dụng bởi một số ngôn ngữ khác của .NET là các biến thể của các ngôn ngữ hiện tại, bao gồm J# (một ngôn ngữ .NET cũng được thiết kế bởi Microsoft có nguồn gốc từ Java 1.1), A# (từ Ada) và ngôn ngữ lập trình chức năng F#. Việc triển khai ban đầu của Eiffel for .NET được gọi là Eiffel#. Hậu tố cũng đã được sử dụng cho các thư viện, chẳng hạn như Gtk# (một wrapper NET cho GTK+ và các thư viện GNOME khác) và Cocoa#.

### 2.1.3. Các phiên bản

➤ C# đã trải qua các phiên bản sau:

Phiên bản	Đặc tả ngôn ngữ			Ngày phát hành	.NET Framework	Visual Studio
	ECMA	ISO/IEC	Microsoft			
C# 1.0	Tháng 12 năm 2002	Tháng 4 năm 2003	Tháng 1 năm 2002	Tháng 1 năm 2002	.NET Framework 1.0	Visual Studio .NET 2002
C# 1.1			Tháng 10 năm 2003	Tháng 4 năm 2003	.NET Framework 1.1	Visual Studio .NET 2003
C# 1.2			September 2005	Tháng 11 năm 2005	.NET Framework 2.0	Visual Studio 2005
C# 2.0	Tháng 6 năm 2006	Tháng 9 năm 2006	Tháng 8 năm 2007	Tháng 11 năm 2007	.NET Framework 2.0 (Except LINQ) <sup>[17]</sup> .NET Framework 3.0 (Except LINQ) <sup>[17]</sup> .NET Framework 3.5	Visual Studio 2008 Visual Studio 2010
C# 3.0	Không <sup>[c]</sup>		Tháng 4 năm 2010	Tháng 4 năm 2010	.NET Framework 4	Visual Studio 2010
C# 4.0	Không <sup>[c]</sup>		Tháng 6 năm 2013	Tháng 8 năm 2012	.NET Framework 4.5	Visual Studio 2012 Visual Studio 2013
C# 5.0	Trong quá trình <sup>[18]</sup>	Không <sup>[c]</sup>	Bản nhập	Tháng 7 năm 2015	.NET Framework 4.6	Visual Studio 2015
C# 6.0	Không <sup>[c]</sup>		Không	Tháng 3 năm 2017	.NET Framework 4.6.2	Visual Studio 2017
C# 7.0	Không	Không	Không	Tháng 8 năm 2017	.NET Framework 4.6.2	Visual Studio 2017 phiên bản 15.3 <sup>[19]</sup>
C# 7.1	Không	Không	Không	Tháng 11 năm 2017	.NET Framework 4.7.1	Visual Studio 2017 phiên bản 15.5 <sup>[20]</sup>
C# 7.2	Không	Không	Không			

Hình 2.1 Các phiên bản của C#.

#### 2.1.4. Tính năng của C#

Với tuổi đời còn rất trẻ của mình, C# là một trong những ngôn ngữ lập trình mới nhất. Tuy nhiên không vì thế mà nó ít được biết tới hay bị coi nhẹ trong lập trình. Lý do rất dễ hiểu – cha đẻ của nó là Tập đoàn khổng lồ Microsoft thêm vào đó là sự cam kết của .NET. Nói thì dễ dàng vậy nhưng nó không hề đơn giản như chúng ta tưởng tượng. Trước C#, Microsoft đã rất vất vả để gây dựng Microsoft Bob nhưng thất bại. Ngôn ngữ đó không thể phổ biến như C# dù Microsoft là gã khổng lồ trong lĩnh vực của mình. Vấn đề là Bob chưa đáp ứng được yêu cầu của nhiều lập trình viên, nó còn khá rối rắm và thô kệch. Ngay sau đó, C# được lên kế hoạch để thay thế Bob và nhiều sản phẩm của Microsoft được viết lại bằng C#. C# đáp ứng được nhiều tiêu chí của người lập trình và nhanh chóng trở nên phổ biến và hiện đang là ngôn ngữ lập trình nổi bật, được ưa chuộng trên toàn thế giới.

Hiện nay, có tới hơn 100 nghìn ứng dụng quản lý được viết bằng C# và chạy trên 2,7 tỉ thiết bị khác nhau như ứng dụng quản lý nhân sự, quản lý nhà hàng, quản lý quán cà phê, ... Từ đó cho thấy C# đang được sử dụng rất rộng rãi và dần len lỏi vào nhiều lĩnh vực cũ và mới.

C# có thể viết hầu hết tất cả các chương trình ứng dụng trong đó tiêu biểu như:

- Winform Application
- Web Application

- WPF
- Ứng dụng cho thiết bị di động
- Game
- ...

#### 2.1.5. Các tính năng mới

##### **C# 2.0**

- Generics
- Partial types
- Hàm Anonymous
- Iterators
- Các kiểu Nullable
- Khả năng tiếp cận getter/setter riêng biệt
- Phương pháp nhóm chuyển đổi (delegate)
- Co- và Contra-variance cho các delegates (delegate)
- Các lớp static
- Delegate inference

##### **C# 3.0**

- LINQ[23]
- Cải tiến trong việc khởi tạo đối tượng
- Các biểu thức lambda: `listOfFoo.Where(delegate(Foo x) { return x.size>10;})` trở thành `listOfFoo.Where(x => x.size>10);`
- Mặc định gõ các biến cục bộ, ví dụ `var x = "hello";` có thể hoán đổi với `string x = "hello";`
- Các property được tự động hiện thực hóa
- Các kiểu anonymous
- Các hàm extension
- Cây biểu thức

- Các hàm dạng partial

#### **C# 4.0**

- Dynamic binding
- Đối số được đặt tên và tùy chọn
- Co- và contravariance dạng generic
- Các kiểu interop nhúng ("NoPIA")

#### **C# 5.0**

- Các hàm asynchronous
- Tìm thông tin thành phần gọi hàm

#### **C# 6.0**

- Compiler-as-a-service (Roslyn)
- Nhập các thành viên kiểu static vào không gian tên.
- Exception filters
- Await trong các khối catch/finally
- Tự động cài đặt property
- Các thành viên trong thân biểu thức
- Toán tử kiểm tra null
- Chuỗi nội suy
- Toán tử nameof

#### **C# 7.0**

- Các biến out
- Pattern matching
- Tuple
- Deconstruction
- Các hàm cục bộ
- Kiểu ValueTask
- Constructor and finalizer trong thân biểu thức

- Getter và setter trong thân biểu thức
- Throw cũng có thể được dùng làm biểu thức

### **C# 7.1**

- Async main
- Tên các phần tử tuple được nội suy

### **C# 7.2**

- Reference ngữ nghĩa với các loại giá trị
- Các đối số được đặt tên không có đuôi
- Giới hạn truy cập private protected cho các field

## **2.2. Tổng quan về .NET FRAMEWORK**

### *2.2.1. Giới thiệu về .NET FRAMEWORK*

.NET Framework là một nền tảng lập trình và cũng là một nền tảng thực thi ứng dụng chủ yếu trên hệ điều hành Microsoft Windows được phát triển bởi Microsoft. Các chương trình được viết trên nền .NET Framework sẽ được triển khai trong môi trường phần mềm (ngược lại với môi trường phần cứng) được biết đến với tên Common Language Runtime (CLR). Môi trường phần mềm này là một máy ảo trong đó cung cấp các dịch vụ như an ninh phần mềm (security), quản lý bộ nhớ (memory management), và các xử lý lỗi ngoại lệ (exception handling). Nhiều thời gian và kinh phí vào việc tái sử dụng và tái thích ứng một Engine để tạo nhiều Web khác nhau.

.NET framework bao gồm tập các thư viện lập trình lớn, và những thư viện này hỗ trợ việc xây dựng các chương trình phần mềm như lập trình giao diện; truy cập, kết nối cơ sở dữ liệu; ứng dụng web; các giải thuật, cấu trúc dữ liệu; giao tiếp mạng... CLR cùng với bộ thư viện này là 2 thành phần chính của .NET framework.

.NET framework đơn giản hóa việc viết ứng dụng bằng cách cung cấp nhiều thành phần được thiết kế sẵn, người lập trình chỉ cần học cách sử dụng và tùy theo sự sáng tạo mà gắn kết các thành phần đó lại với nhau. Nhiều công cụ được tạo ra để hỗ trợ xây dựng ứng

dụng.NET, và IDE (Integrated Development Environment) được phát triển và hỗ trợ bởi chính Microsoft là Visual Studio.

### 2.2.2. Lịch sử phát triển

Phiên bản	Số hiệu phiên bản	Ngày phát hành	Visual Studio	Được phát hành kèm theo
1.0	1.0.3705.0	13 tháng 2 năm 2002	Visual Studio.NET	Windows XP Tablet and Media Center Editions <sup>[1]</sup>
1.1	1.1.4322.573	24 tháng 4 năm 2003	Visual Studio.NET 2003	Windows Server 2003
2.0	2.0.50727.42	7 tháng 11 năm 2005	Visual Studio 2005	Windows Server 2003 R2
3.0	3.0.4506.30	6 tháng 11 năm 2006		Windows Vista, Windows Server 2008
3.5	3.5.21022.8	19 tháng 11 năm 2007	Visual Studio 2008	Windows 7, Windows Server 2008 R2
4.0	4.0.30319.1	12 tháng 4 năm 2010	Visual Studio 2010	
4.5	4.5.50709	15 tháng 8 năm 2012	Visual Studio 2012	Windows 8, Windows Server 2012
4.6		20 tháng 7 năm 2015	Visual Studio 2015	Windows 10

Hình 2.2 Các phiên bản của .NET Framework.

### 2.2.3. Các tính năng mới

#### .NET Framework 1.0

Đây là phiên bản đầu tiên của .NET framework, nó được phát hành vào năm 2002 cho các hệ điều hành Windows 98, NT 4.0, 2000 và XP. Việc hỗ trợ chính thức từ Microsoft cho phiên bản này kết thúc vào 10/7/2007, tuy nhiên thời gian hỗ trợ mở rộng được kéo dài đến 14/7/2009.

#### .NET Framework 1.1

Phiên bản nâng cấp đầu tiên được phát hành vào 4/2003. Sự hỗ trợ của Microsoft kết thúc vào 14/10/2008, và hỗ trợ mở rộng được định đến 8/10/2013.

Những thay đổi so với phiên bản 1.0:

- Tích hợp hỗ trợ mobile ASP.NET (trước đây chỉ là phần mở rộng tùy chọn)
- Thay đổi về kiến trúc an ninh - sử dụng sandbox khi thực thi các ứng dụng từ Internet.
- Tích hợp hỗ trợ ODBC và cơ sở dữ liệu Oracle
- .NET Compact Framework
- Hỗ trợ IPv6 (Internet Protocol version 6)
- Vài thay đổi khác trong API

## .NET Framework 2.0

Kể từ phiên bản này, .NET framework hỗ trợ đầy đủ nền tảng 64-bit. Ngoài ra, cũng có một số thay đổi trong API; hỗ trợ các kiểu "generic"; bổ sung sự hỗ trợ cho ASP.NET; .NET Micro Framework - một phiên bản .NET framework có quan hệ với Smart Personal Objects Technology.

## .NET Framework 3.0

Đây không phải là một phiên bản mới hoàn toàn, thực tế chỉ là một bản nâng cấp của .NET 2.0. Phiên bản 3.0 này còn có tên gọi khác là WinFX, nó bao gồm nhiều sự thay đổi nhằm hỗ trợ việc phát triển và chuyển đổi (porting) các ứng dụng trên Windows Vista. Tuy nhiên, không có sự xuất hiện của .NET Compact Framework 3.0 trong lần phát hành này.

Bốn thành phần chính trong phiên bản 3.0:

- 1) *Windows Presentation Foundation* (WPF - tên mã là Avalon): Đây là một công nghệ mới, và là một nỗ lực của Microsoft nhằm thay đổi phương pháp hay cách tiếp cận việc lập trình một ứng dụng sử dụng giao diện đồ họa trên Windows với sự hỗ trợ của ngôn ngữ XAML.
- 2) *Windows Communication Foundation* (WCF - tên mã là Indigo): Một nền tảng mới cho phép xây dựng các ứng dụng hướng dịch vụ (service-oriented).
- 3) *Windows Workflow Foundation* (WF): Một kiến trúc hỗ trợ xây dựng các ứng dụng workflow (luồng công việc) một cách dễ dàng hơn. WF cho phép định nghĩa, thực thi và quản lý các workflow từ cả cách nhìn theo hướng kỹ thuật và hướng thương mại.
- 4) *Windows CardSpace* (tên mã là InfoCard): một kiến trúc để quản lý định danh (identity management) cho các ứng dụng được phân phối.

## .NET Framework 3.5

Được phát hành vào 11/2007, phiên bản này sử dụng CLR 2.0. Đây có thể được xem là tương đương với phiên bản .NET Framework 2.0 SP1 và .NET Framework 3.0 SP1 cộng lại. .NET Compact Framework 3.5 được ra đời cùng với phiên bản .NET framework này.

Các thay đổi kể từ phiên bản 3.0:

- Các tính năng mới cho ngôn ngữ C# 3.0 và VB.NET 9.0
- Hỗ trợ Expression Tree và Lambda
- Các phương thức mở rộng (Extension methods)
- Các kiểu ẩn danh (Anonymous types)
- LINQ
- Phân trang (paging) cho ADO.NET
- API cho nhập xuất mạng không đồng bộ (asynchronous network I/O)
- Peer Name Resolution Protocol resolver
- Cải thiện WCF và WF
- Tích hợp ASP.NET AJAX
- Namespace mới System.CodeDom
- Microsoft ADO.NET Entity Framework 1.0

Cũng như phiên bản 3.0, có thể minh họa sự thay đổi của .NET 3.5 bằng công thức:

*.NET 3.5 = .NET 3.0 + LINQ + ASP.NET 3.5 + REST*

#### **.NET Framework 4.0**

Phiên bản beta đầu tiên của .NET 4 xuất hiện vào 5/2009 và phiên bản RC (*Release Candidate*) được ra mắt vào 2/2010. Bản chính thức của .NET 4 được công bố và phát hành cùng với Visual Studio 2010 vào 12/4/2010.

Các tính năng mới được Microsoft bổ sung trong .NET 4:

- Dynamic Language Runtime
- Code Contracts
- Managed Extensibility Framework
- Hỗ trợ các tập tin ánh xạ bộ nhớ (memory-mapped files)
- Mô hình lập trình mới cho các ứng dụng đa luồng (multithreaded) và bất đồng bộ (asynchronous)
- Cải thiện hiệu năng, các mô hình workflow.



## .NET Framework 4.5

Những thông tin đầu tiên của .NET 4.5 được Microsoft công bố vào 14/9/2011 tại *BUILD Windows Conference*, và nó chính thức được ra mắt vào 15/8/2012.

Kể từ phiên bản này, Microsoft bắt đầu cung cấp 2 gói cài đặt riêng biệt, gói đầy đủ và gói giảm chức năng *client profiles*.

### 2.3. Tổng quan về ASP.NET MVC 5

#### 2.3.1. Giới thiệu về ASP.NET MVC 5

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 hiện nay.

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.
- *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 gửi lên (query-string values) và gửi các giá trị đó đến model, model sẽ lấy dữ liệu từ CSDL nhờ vào các giá trị này.

### 2.3.2. Ưu điểm của mô hình MVC

Nền tảng ASP.NET MVC mang lại những lợi ích sau:

- 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 có thể thiết kế một hạ tầng quản lý định tuyến.
- 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.

### 2.3.3. Tính năng của nền tảng 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 kế 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 (.master). 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 thành viên, quyền, output caching và data caching, session và profile, quản lý tình trạng ứng dụng, hệ thống cấu hình...

ASP.NET MVC 5 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. Tổng quan về HTML**

### *2.4.1. Khái niệm*

HTML là chữ viết tắt của cụm từ Hyper Text Markup Language (dịch là Ngôn ngữ đánh dấu siêu văn bản) được sử dụng để tạo một trang web, trên một website có thể sẽ chứa nhiều trang và mỗi trang được quy ra là một tài liệu HTML (thi thoảng mình sẽ ghi là một tập tin HTML). Cha đẻ của HTML là Tim Berners-Lee, cũng là người khai sinh ra World Wide Web và chủ tịch của World Wide Web Consortium (W3C – tổ chức thiết lập ra các chuẩn trên môi trường Internet).

Một tài liệu HTML được hình thành bởi các phần tử HTML (HTML Elements) được quy định bằng các cặp thẻ (tag), các cặp thẻ này được bao bọc bởi một dấu ngoặc nhọn (ví dụ <html>) và thường là sẽ được khai báo thành một cặp, bao gồm thẻ mở và thẻ đóng (ví dụ <strong> dụ </strong> và ). Các văn bản muốn được đánh dấu bằng HTML sẽ được khai báo bên trong cặp thẻ (ví dụ <strong>Đây là chữ in đậm</strong>). Nhưng một số thẻ đặc biệt lại không có thẻ đóng và dữ liệu được khai báo sẽ nằm trong các thuộc tính (ví dụ như thẻ <img>).

Một tập tin HTML sẽ bao gồm các phần tử HTML và được lưu lại dưới đuôi mở rộng là .html hoặc .htm.

#### *2.4.2. Cách HTML được xử lý*

Khi một tập tin HTML được hình thành, việc xử lý nó sẽ do trình duyệt web đảm nhận. Trình duyệt sẽ đóng vai trò đọc hiểu nội dung HTML từ các thẻ bên trong và sẽ chuyển sang dạng văn bản đã được đánh dấu để đọc, nghe hoặc hiểu (do các bot máy tính hiểu).

#### *2.4.3. Vai trò của HTML trong website*

HTML là một ngôn ngữ đánh dấu siêu văn bản nên nó sẽ có vai trò xây dựng cấu trúc siêu văn bản trên một website, hoặc khai báo các tập tin kỹ thuật số (media) như hình ảnh, video, nhạc. Vai trò của từng ngôn ngữ trong website

Điều đó không có nghĩa là chỉ sử dụng HTML để tạo ra một website mà HTML chỉ đóng một vai trò hình thành trên website. Ví dụ một website thường sẽ được hình thành bởi:

- HTML – Xây dựng cấu trúc và định dạng các siêu văn bản.
- CSS – Định dạng các siêu văn bản dạng thô tạo ra từ HTML thành một bố cục website, có màu sắc, ảnh nền,....
- Javascript – Tạo ra các sự kiện tương tác với hành vi của người dùng (ví dụ nhấp vào ảnh trên nó sẽ có hiệu ứng phóng to).
- PHP – Ngôn ngữ lập trình để xử lý và trao đổi dữ liệu giữa máy chủ đến trình duyệt (ví dụ như các bài viết sẽ được lưu trong máy chủ).

- MySQL – Hệ quản trị cơ sở dữ liệu truy vấn có cấu trúc (SQL – ví dụ như các bài viết sẽ được lưu lại với dạng dữ liệu SQL).

## **2.5. Tổng quan về CSS**

### *2.5.1. Khái niệm*

CSS là chữ viết tắt của Cascading Style Sheets, nó là một ngôn ngữ được sử dụng để tìm và định dạng lại các phần tử được tạo ra bởi các ngôn ngữ đánh dấu (ví dụ như HTML). Ta có thể hiểu đơn giản rằng, nếu HTML đóng vai trò định dạng các phần tử trên website như việc tạo ra các đoạn văn bản, các tiêu đề, bảng,...thì CSS sẽ giúp chúng ta có thể thêm một chút “phong cách” vào các phần tử HTML đó như đổi màu sắc trang, đổi màu chữ, thay đổi cấu trúc,...rất nhiều.

### *2.5.2. Phương thức hoạt động*

Phương thức hoạt động của CSS là nó sẽ tìm dựa vào các vùng chọn, vùng chọn có thể là tên một thẻ HTML, tên một ID, class hay nhiều kiểu khác. Sau đó là nó sẽ áp dụng các thuộc tính cần thay đổi lên vùng chọn đó.

## **2.6. Tổng quan về Javascript**

### *2.6.1. Khái niệm*

JavaScript là ngôn ngữ lập trình dùng để tạo ra các trang web có tính tương tác. Các slideshow, pop-up quảng cáo và tính năng autocomplete của Google là những thứ được viết bằng JavaScript. Nó được tạo ra lần đầu tiên bởi Brendan Eich tại công ty Netscape vào năm 1995 - được đặt tên là Mocha trong quá trình phát triển, phát hành phiên bản beta với tên là LiveScript và cuối cùng được đặt tên là JavaScript để lợi dụng sự phổ biến của Java (một ngôn ngữ lập trình khác) vì lý do marketing.

Lúc đầu, các lập trình viên đã không coi JavaScript là quan trọng, bởi vì nó không được xem như là một ngôn ngữ phát triển nghiêm túc như các ngôn ngữ phía máy chủ là Java, Ruby hay Python. JavaScript giống như phần kem phết trên bề mặt chiếc bánh vậy, nó chỉ chịu trách nhiệm về phần trải nghiệm người dùng. Nhưng ngôn ngữ này tiếp tục phát

triển và ngày càng trở nên phổ biến, nó thường được đề xuất là ngôn ngữ lập trình nên học đầu tiên dành cho người mới bắt đầu.

### 2.6.2. Sự phổ biến của Javascript

JavaScript là ngôn ngữ kịch bản phía máy khách (client-side) duy nhất. Do đó, nó được sử dụng trên hầu hết các trang web mà bạn nhìn thấy trên internet. Đây là một trong những lý do tại sao Codecademy cung cấp JavaScript là ngôn ngữ đầu tiên cho các học viên tìm hiểu. Không giống như các ngôn ngữ cổ điển đang được giảng dạy ở môn khoa học máy tính tại các trường đại học, JavaScript chắc chắn sẽ được sử dụng trong công việc nếu bạn là một nhà phát triển web.

Ngoài ra, kể từ khi JavaScript chạy trong một trình duyệt, bạn không cần phải tải một phần mềm nào đó về để sử dụng nó. Tất cả bạn cần chỉ là một trình soạn thảo văn bản (như TextWrangler hay TextEdit) và một trình duyệt web. Đây là một lợi ích rõ ràng cho các lập trình viên mới vào nghề.

Mặc cho những tiện ích của nó mang lại, các lập trình viên không coi JavaScript là ngôn ngữ tinh tế nhất - cú pháp của nó có vẻ lộn xộn. CoffeeScript là một nỗ lực để thay đổi điều này - nó làm cho JavaScript trông giống như Ruby hay Python bằng cách loại bỏ các dấu ngoặc xoắn, dấu chấm phẩy, bổ sung thêm khoảng trắng và căn lề thật đầu dòng.

Những thay đổi này làm cho code dễ đọc hơn, cho dù chúng ta đang xem lại phần code cũ của mình và cố gắng nhớ ra nó dùng để làm gì, hay đang làm việc chung trên một dự án với các lập trình viên khác.

### 2.6.3. Ứng dụng của Javascript

JavaScript có thể được sử dụng để xây dựng toàn bộ một trang web nếu sử dụng với Node.js và MongoDB (một cơ sở dữ liệu). Node.js là bất đồng bộ, điều đó làm cho nó có tính dễ mở rộng và rất phù hợp cho tất cả các dữ liệu lớn ở ngoài kia. Ví dụ, một nhà phát triển đã nhận thấy việc gửi 600 email với Node.js chỉ mất 3 giây, trong khi đó phải mất 30 giây với PHP. MongoDB là một cơ sở dữ liệu hướng tài liệu (document-oriented) được xây dựng phục vụ cho khả năng mở rộng nhưng các chức năng nó cung cấp vẫn tốt nhất cho cả hai thế giới.

JavaScript hiện nay là một trong những cách ưa thích để phát triển ứng dụng web. Đó là ngôn ngữ được sử dụng nhiều nhất trên Github. Rất nhiều các trang web được xây dựng với Node, bao gồm cả các trang web lớn như Klout, Storify và Yammer. Việc có thể sử dụng cùng một ngôn ngữ trên cả phần front-end và back-end làm cho cuộc sống của các nhà phát triển phần mềm trở nên dễ dàng hơn.

Riêng bản thân nó, JavaScript chịu trách nhiệm về các hiệu ứng động như các cửa sổ pop-up và với AJAX, những trang web có thể hiển thị dữ liệu mới mà không cần phải refresh lại trang. Bạn có thể xem một ví dụ của AJAX trong thực tế ở đây - bạn thấy thế nào khi nhấp chuột vào một hình ảnh trên trang đó, thì địa chỉ URL thay đổi và nó hiển thị dữ liệu mới tương ứng mà không cần phải refresh lại toàn bộ trang. AJAX là cụm từ viết tắt của Asynchronous JavaScript And Xml, vì vậy nó không thể được sử dụng ngoài JavaScript.

## **2.7. Tổng quan về Bootstrap**

### *2.7.1. Khái niệm*

Bootstrap là framework HTML, CSS và JavaScript phổ biến nhất để phát triển các trang web có tính phản hồi, tích hợp di động. Nếu bạn mong muốn sở hữu một Website Responsive, tương thích với mọi trình duyệt và thiết bị di động, chắc chắn sẽ cần dùng đến công nghệ này.

Bootstrap bao gồm các mã CSS + HTML cơ bản cho typography, forms, buttons, tables, grids, navigation, và nhiều thành phần khác trong một website. Điều này giúp cho các designer tránh việc phải lặp đi lặp lại trong quá trình tạo ra các class CSS và các đoạn mã HTML giống nhau trong khi thiết kế web. Style của các phần tử HTML trong Bootstrap khá đơn giản và thanh lịch. Ví dụ như phần đổ bóng trong input, highlight của bảng biểu, các mã CSS hiển thị cảnh báo, tabs, phân trang...

### *2.7.2. Ưu điểm*

Lý do đầu tiên khiến Bootstrap dần trở nên phổ biến và xứng đáng được lựa chọn đó là gần đây, các nhà lập trình đã bổ sung thêm tính năng Customize (Tùy chỉnh). Chức năng này giúp cho các nhà thiết kế linh hoạt hơn trong việc lựa chọn những thuộc tính, phần tử

phù hợp với dự án họ đang theo đuổi. Chức năng này cũng cho phép bạn không cần phải tải toàn bộ mã nguồn về máy.

Bootstrap không đòi hỏi kiến thức quá phức tạp. Chỉ cần biết sơ qua HTML, CSS, Javascript, JQuery là bạn có thể sử dụng nó để tạo nên một website chuyên nghiệp, website theo yêu cầu đúng như mong muốn.

Bootstrap được viết bởi những bộ não công nghệ tài năng trên khắp thế giới. Sự tương thích của trình duyệt với thiết bị đã được kiểm tra nhiều lần trước khi đưa vào sử dụng, nên khi lựa chọn Bootstrap, bạn có thể tin rằng mình sẽ tạo nên những sản phẩm hoàn hảo.

Do sử dụng Grid System nên Bootstrap mặc định hỗ trợ Responsive. Bootstrap được viết theo xu hướng Mobile First tức là ưu tiên cho việc tương thích với các giao diện trên thiết bị di động. Sử dụng công nghệ này cho website của bạn sẽ giúp website trở nên tương thích với tất cả kích thước màn hình – một điều cực kì quan trọng và cần thiết trong trải nghiệm người dùng.

Bootstrap hoạt động theo xu hướng mã nguồn mở nên bạn có thể vào mã nguồn của nó để thay đổi và chỉnh sửa tùy ý.



## CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ

### 3.1. Phân tích

#### 3.1.1. Mô tả chương trình

Website đánh giá công nghệ được xây dựng lên nhằm mục đích giúp cho người dùng có thể tìm hiểu về món đồ công nghệ mọi lúc mọi nơi mà không phải trực tiếp đến chỗ mua bán qua đó cắt giảm được các chi phí liên quan.

Khi tiến hành đánh giá người dùng sẽ cung cấp thông tin cá nhân để hệ thống lưu vào cơ sở dữ liệu để thuận tiện cho việc quản lý cũng như trả kết quả sau này.

Sản phẩm bao gồm các chức năng chính sau:

- Login/user.
- Coi bài viết.
- Đăng bài.

#### 3.1.1. Ý tưởng

Sau khi được giáo viên hướng dẫn giao đề tài. Chúng em đã tiến hành nghiên cứu và tham khảo một số ý tưởng, đề tài trên mạng. Do chúng em thiết kế và tùy biến toàn bộ giao diện của Web, và thêm vào đó một số chức năng để giúp người chơi trải nghiệm Web tốt hơn.

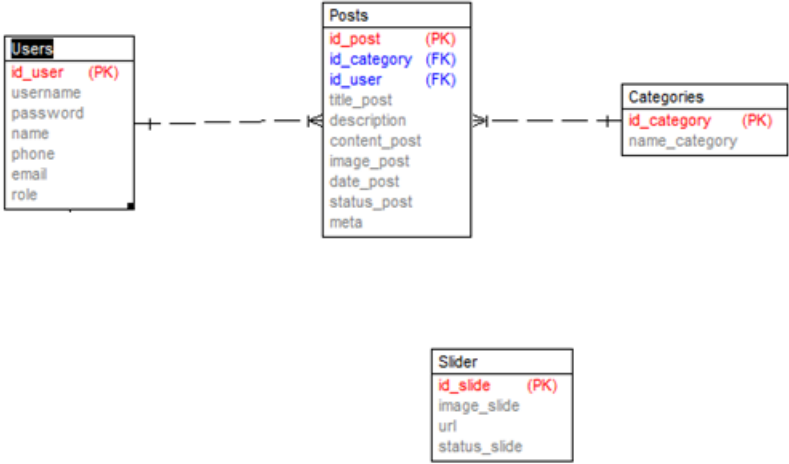
#### 3.1.3. Đối tượng người dùng

- User : là người trực tiếp sử dụng phần mềm ở đây cụ thể là người muốn đánh giá hoặc tìm hiểu về món đồ công nghệ
- Admin: là quản trị viên, nhân viên, có khả năng đăng bài viết mới và các chức năng bên trong của hệ thống như : thêm, xoá, sửa dữ liệu

#### 3.1.4. Ràng buộc tổng thể

- Tương thích với tất cả các thiết bị di động, máy tính.
- Chạy được trên mọi trình duyệt, hệ điều hành.

3.1.5. Mô hình thực thể kết hợp



Hình 3.2 Mô hình thực thể kết hợp

### 3.2. Cơ sở dữ liệu

**USERS**(id\_user, username, password, name, phonenumber, email, role)

STT	Tên Trường	Kiểu dữ liệu	Khoá
1	id_user	int	Khoá chính
2	username	varchar(100)	
3	password	varchar(255)	
4	name	nvarchar100	
5	phonenumber	varchar(11)	
6	email	varchar(100)	
7	Role	int	

*Bảng 3.1 Bảng dữ liệu Users.*

**CATEGORIES**(id\_category, namecategory)

STT	Tên Trường	Kiểu dữ liệu	Khoá
1	id_category	Int	Khoá chính
2	namecategory	nvarchar(100)	

*Bảng 3.2 Bảng dữ liệu Categories.*

**POSTS**(id\_posts, id\_category, id\_users, title\_post, description, content\_post, image\_post, date\_post, status\_post, meta)

STT	Tên Trường	Kiểu dữ liệu	Khoá
1	Id_post	int	Khoá chính

2	Id_category	int	Khóa ngoại
3	Id_user	int	Khóa ngoại
4	Title_post	Nvarchar(255)	
5	description	nvarchar(500)	
6	Content_post	Ntext	
7	Image_post	Varchar(100)	
8	Date_post	Date time	
9	Status_post	int	
10	Meta	Varchar(255)	

*Bảng 3.3 Bảng dữ liệu Posts.*

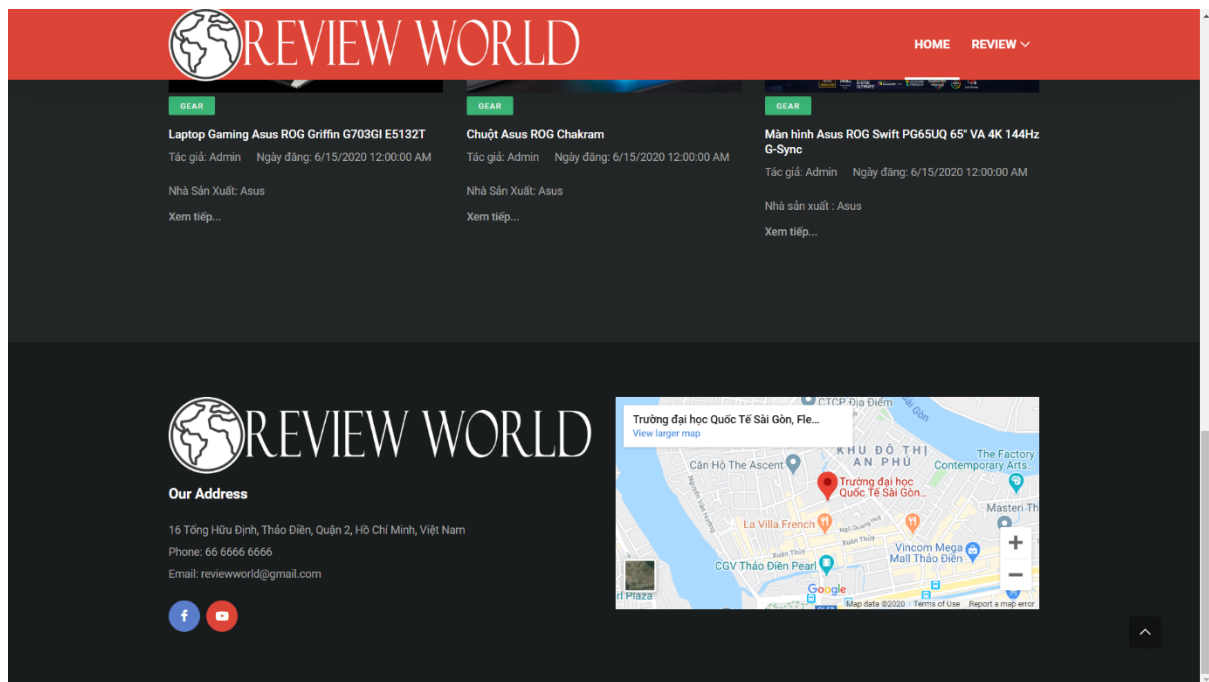
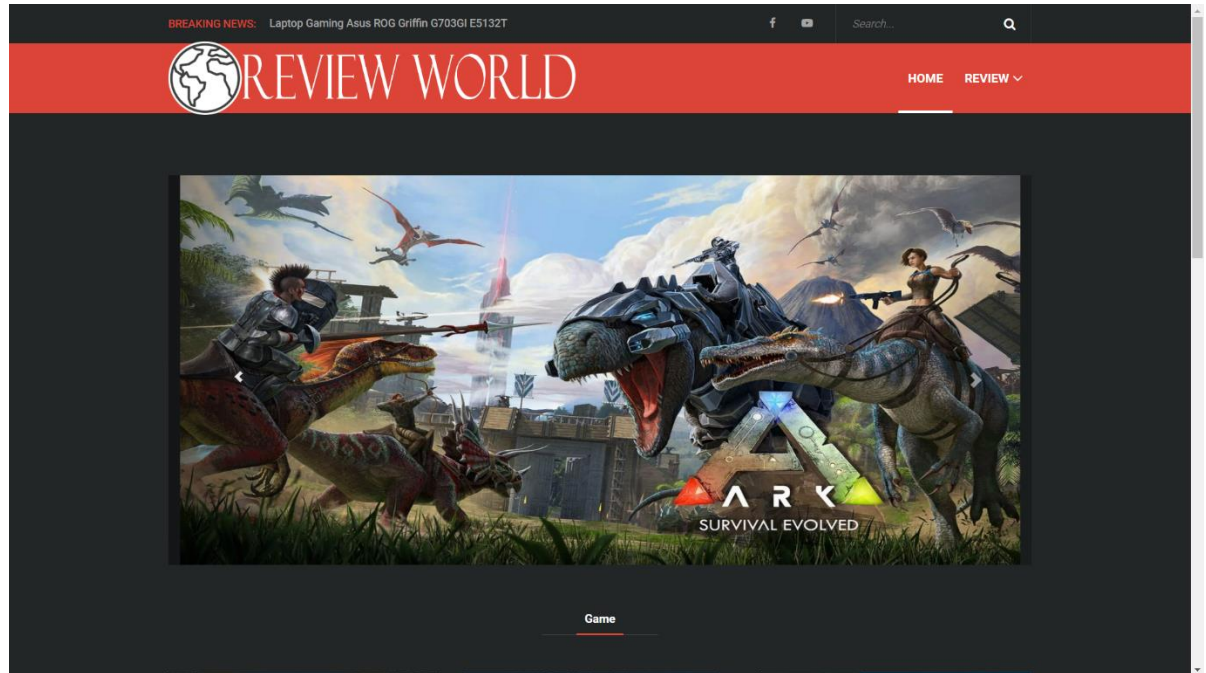
**SLIDER**(ID\_Slide, image\_slide, URL, status\_slide)

STT	Tên Trường	Kiểu dữ liệu	Khoá
1	ID_Slide	Int	Khoá chính
2	Image_slide	varchar(100)	
3	URL	varchar(255)	
4	Status_sldie	int	

*Bảng 3.6 Bảng dữ liệu Slider.*

# CHƯƠNG 4: KẾT QUẢ THỰC NGHIỆM

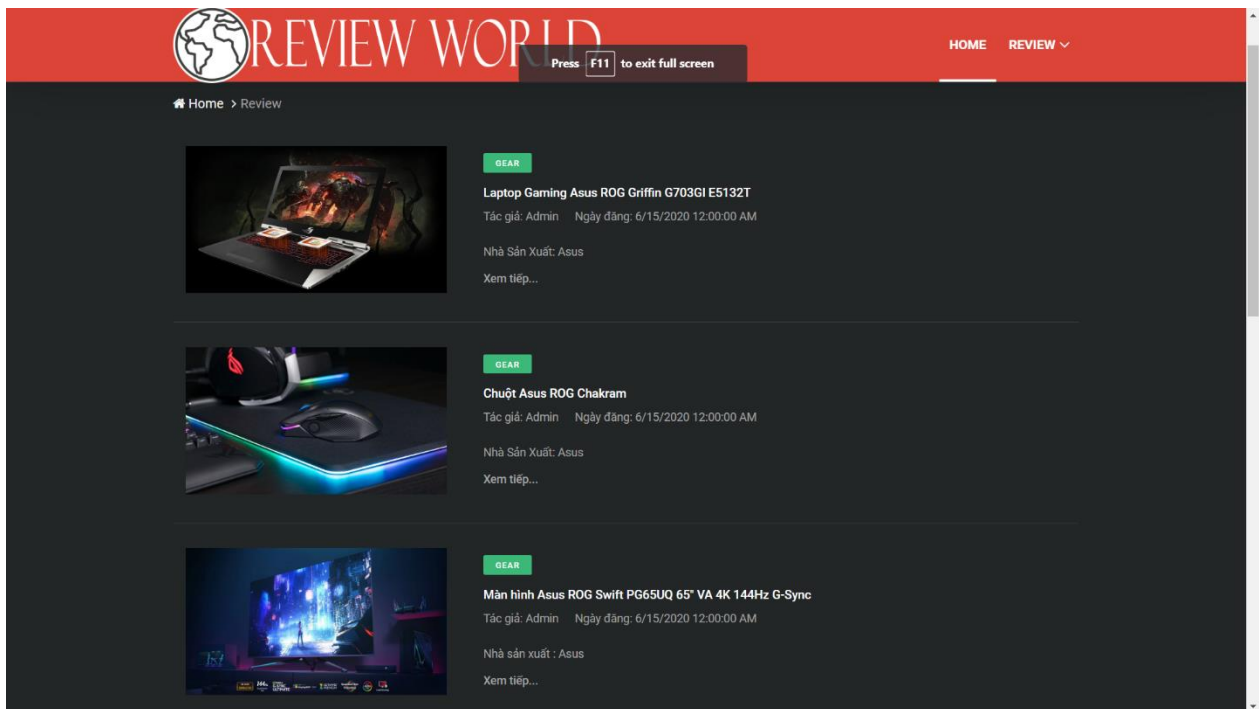
## 4.1. Giao diện trang chủ



Hình 4.1 Giao diện trang chủ.

- Giao diện này bao gồm nút home, nút review, logo, nút chức năng mở rộng ở góc review phải trên cùng sẽ được bung ra khi người dùng đưa chuột vào.
- Tên của trang web là Review World.
- Giao diện chính Web thân thiện, chức năng đơn giản. Ai cũng có thể sử dụng và trên mọi thiết bị.

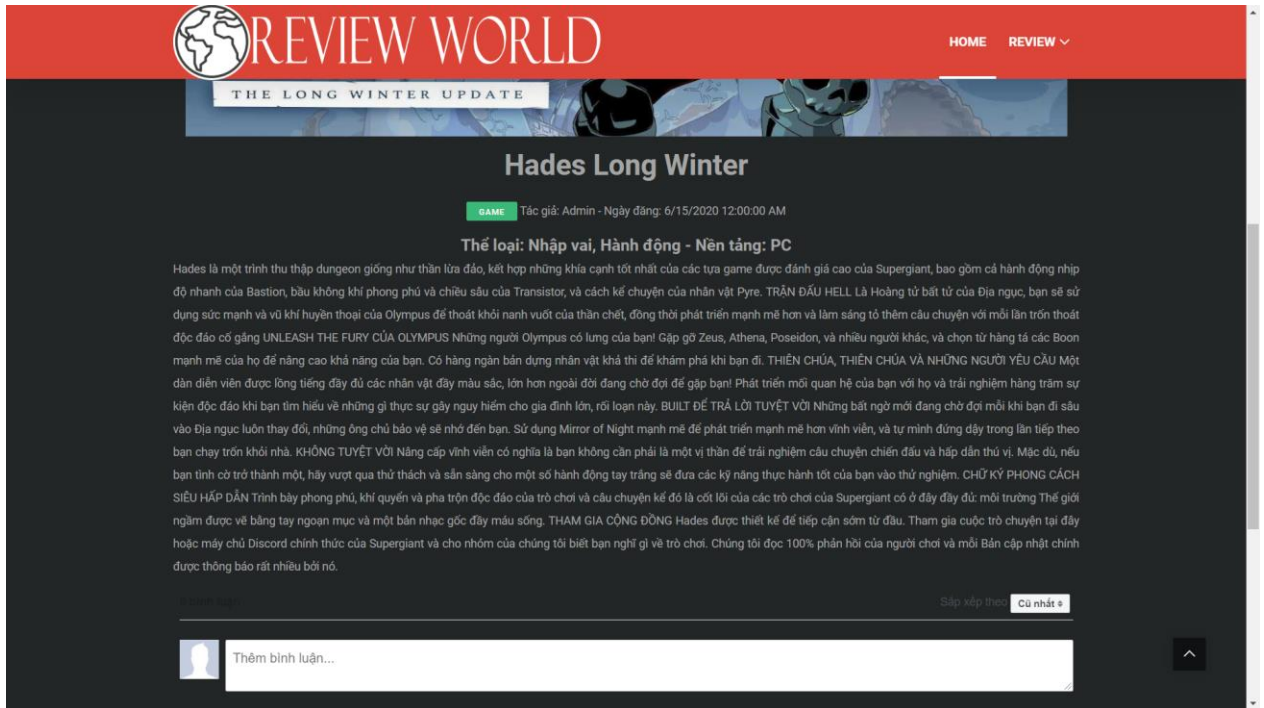
## 4.2. Giao diện chọn bài viết



Hình 4.2 Giao diện chọn bài viết.

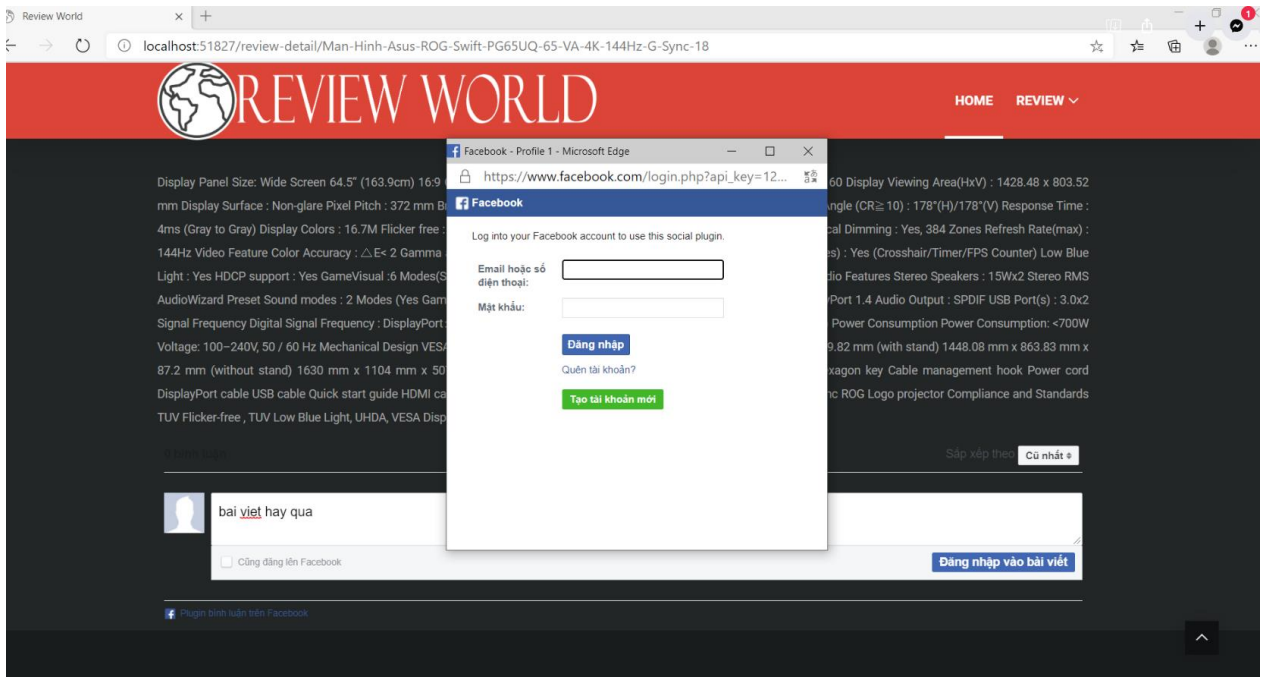
- Thanh chọn bài viết: Người dùng sẽ chọn bài viết mình muốn tìm hiểu hệ thống sẽ đưa ra những tin tức của bài viết đó và thông tin của bài viết.

### 4.3. Giao diện bài viết



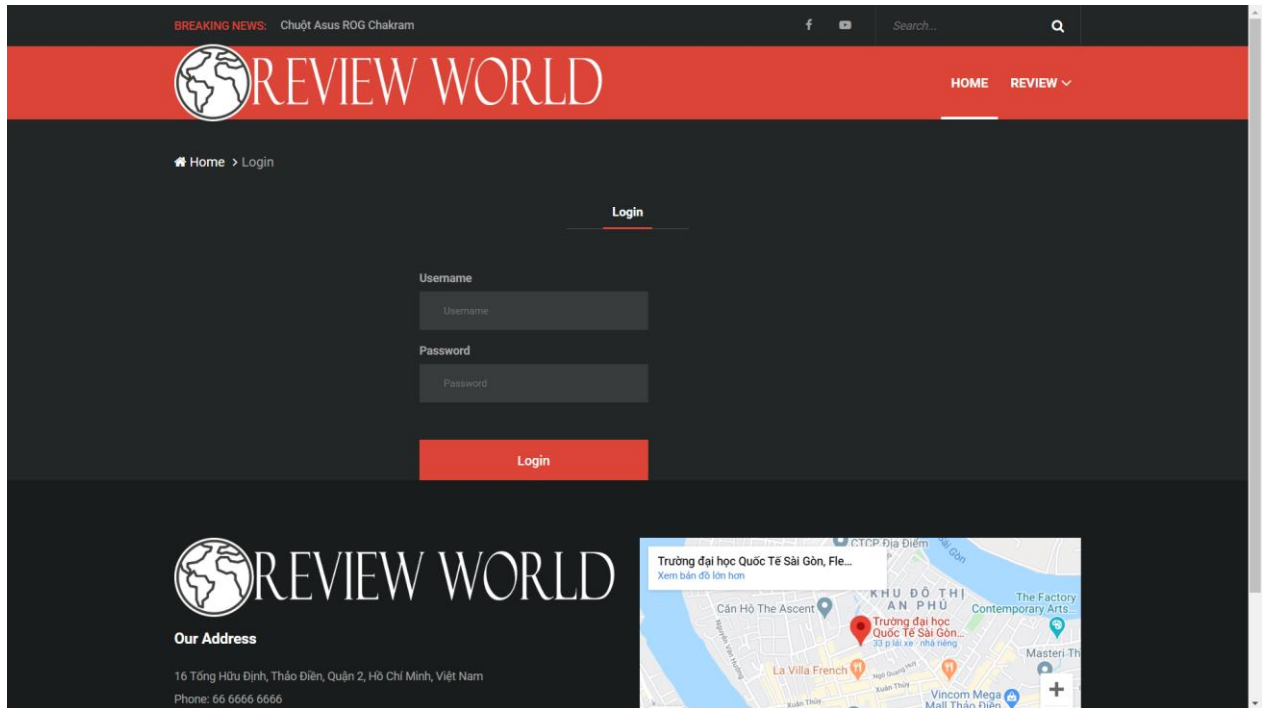
Hình 4.3 Giao diện bài viết.

### 4.4. Giao diện bình luận

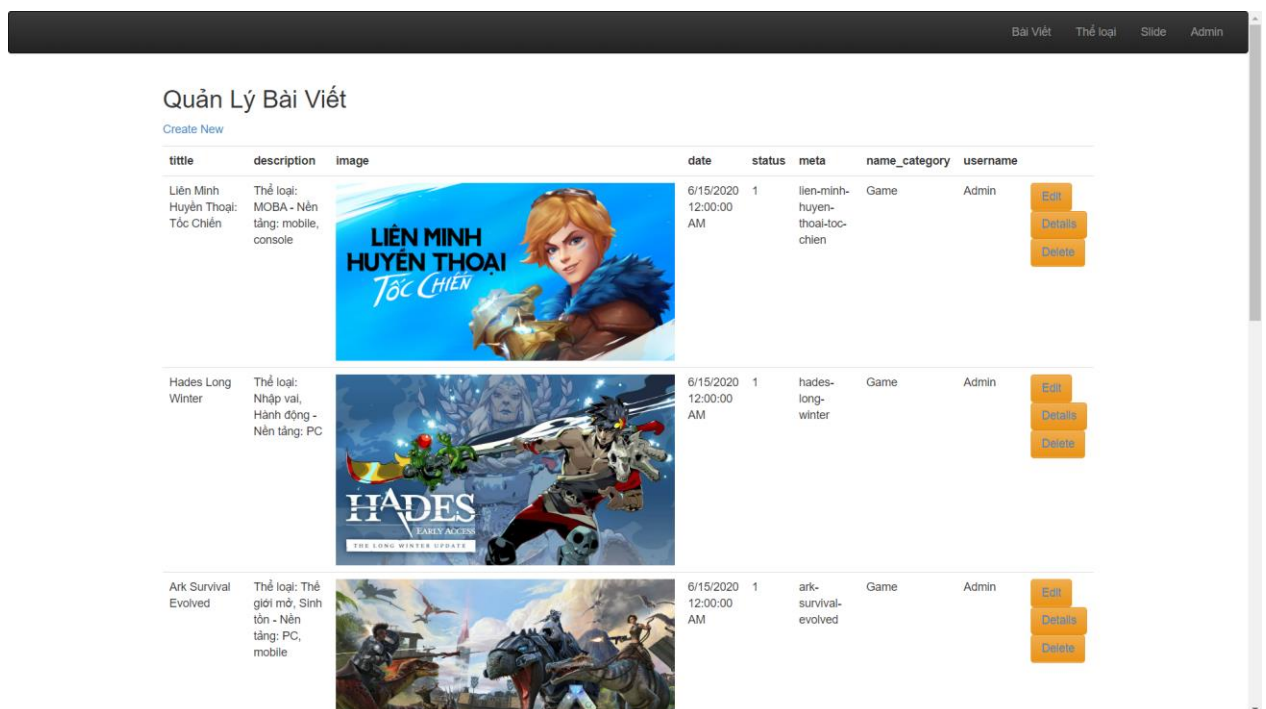


Hình 4.4 Giao diện thông báo đăng nhập để bình luận

## 4.5. Giao diện đăng nhập admin



Hình 4.5 Giao diện thông báo đăng nhập admin



Hình 4.6 Giao diện quản lý bài viết



## Quản lý thể loại

[Create New](#)

name\_category

Game

Edit

Details

Delete

Gear

Edit

Details

Delete

Hình 4.6 Giao diện quản lý thể loại

## CHƯƠNG 5: KẾT LUẬN

### 5.1 Kết quả đạt được

Web được xây dựng trên nền tảng ASP.NET MVC 5. Trước tiên muốn xây dựng một tựa Web thì đòi hỏi trước đó phải tiến hành khảo sát tình huống, mô tả và phân tích thiết kế, lên ý tưởng giao diện cho Web.

Với nhu cầu phát triển của xã hội ngày nay đòi hỏi phần mềm làm ra phải có tính thực tiễn cao, có thể ứng dụng tốt vào thực tế. Qua việc nghiên cứu đề tài này giúp nhóm chúng em củng cố kiến thức về lập trình cũng như là tìm hiểu được công nghệ làm Web trên đa nền tảng của ASP.NET MVC, hiểu được cách thức làm Web trên công nghệ mới này, một công cụ hỗ trợ làm Web cực kỳ mạnh mẽ với nhiều tính năng mới, giúp người lập trình nói chung và nhóm em nói riêng. Bootstrap giúp nhóm em thiết kế giao diện một cách dễ

dàng, tương thích với mọi thiết bị với mọi kích thước màn hình mang đến trải nghiệm người dùng tốt.

## 5.2 Đánh giá Web

### 5.2.1 Ưu điểm

Người dùng có thể truy cập trên tất cả các thiết bị có kết nối internet

- Giao diện đẹp mắt, trong sạch tạo cảm giác sang trọng
- Tiết kiệm thời gian cho người có nhu cầu đọc tin tức
- Nhanh gọn lẹ

### 5.2.2 Nhược điểm

- Công tác quản lý còn gặp nhiều khó khăn do có nhiều vấn đề liên quan
- Web mới xây dựng nên còn thiếu nhiều chức năng, còn hạn chế quyền kiểm soát

## 5.3 Hướng phát triển và mở rộng đề tài

- Khắc phục những khuyết điểm trên.
- Cập nhật chức năng quản lý hoàn chỉnh.
- Triển khai tên miền cho web

## TÀI LIỆU THAM KHẢO:

- <https://tedu.com.vn/video/alias-120.html>
- <https://handsomeroot.blogspot.com/2015/12/ace-responsive-admin-template-free.html>
- <https://www.laptrinh123.com/Hoc-Lap-Trinh/Web?fbclid=IwAR06EPFwrCq44PZB1CNGXY7zg1-fF5AxKEGqhZPGd-xn7sDcKWsTzTFWZuo>
- <https://owlcarousel2.github.io/OwlCarousel2/>
- <https://getbootstrap.com/docs/4.0/components/carousel/?fbclid=IwAR08gr7TPqcV2NlmqLMKfgY4rAxfmsnS0yFNlvsH8iwJvKVGmY6C0O1prMc>