**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

------------------------------



**ĐỒ ÁN THỰC TẬP**

**TỐT NGHIỆP ĐẠI HỌC**

***Đề tài*:**

**“PHÁT TRIỂN NỀN TẢNG PHÂN PHỐI DỊCH VỤ KẾT NỐI**

**GIỮA CÁC CÔNG TY CUNG CẤP DỊCH VỤ VÀ NGƯỜI TIÊU DÙNG”.**

**Người hướng dẫn : ThS. LÊ MINH HÓA**

**Sinh viên thực hiện : VŨ THỊ HẢO**

**Mã số sinh viên : N14DCCN172**

**Lớp : D14CQCP01-N**

**Khoá** **: 2014**

**Hệ** **: CHÍNH QUY**

**TP.HCM, tháng 8/2018**

**CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**Độc lập- Tự do- Hạnh phúc**

TP. Hồ Chí Minh, ngày tháng năm 20…

**NHẬN XÉT CỦA ĐƠN VỊ**

**THỰC TẬP TỐT NGHIỆP ĐẠI HỌC**

1. Tên đề tài**: Phát triển nền tảng phân phối dịch vụ kết nối giữa các công ty cung cấp dịch vụ và người tiêu dùng**
2. Sinh viên : **Vũ Thị Hảo** Lớp: **D14CQCP01-N**
3. Người hướng dẫn :
4. Nơi công tác : SaiGonLab

**NỘI DUNG NHẬN XÉT**

1. Đánh giá chung:

……………………………………………………………………………………………………………………………………………………………………………………

1. Đánh giá chi tiết:

……………………………………………………………………………………………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………………………………………………

1. Nhận xét về tinh thần, thái độ làm việc:

……………………………………………………………………………………………………………………………………………………………………………………

1. Kết luận:

……………………………………………………………………………………………………………………………………………………………………………………

1. Điểm hướng dẫn ():

**ĐƠN VỊ THỰC TẬP**

(Ký, ghi rõ họ tên)

**CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**Độc lập- Tự do- Hạnh phúc**

TP. Hồ Chí Minh, ngày tháng năm 20…

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

**THỰC TẬP TỐT NGHIỆP ĐẠI HỌC**

1. Tên đề tài**: Phát triển nền tảng phân phối dịch vụ kết nối giữa các công ty cung cấp dịch vụ và người tiêu dùng**
2. Sinh viên : **Vũ Thị Hảo** Lớp: **D14CQCP01-N**
3. Giáo viên hướng dẫn : **ThS. Lê Minh Hóa**
4. Nơi công tác : Học Viện Công Nghệ Bưu Chính Viễn Thông

**NỘI DUNG NHẬN XÉT**

1. Đánh giá chung:

……………………………………………………………………………………………………………………………………………………………………………………

1. Đánh giá chi tiết:

……………………………………………………………………………………………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………………………………………………

1. Nhận xét về tinh thần, thái độ làm việc:

……………………………………………………………………………………………………………………………………………………………………………………

1. Kết luận:

……………………………………………………………………………………………………………………………………………………………………………………

1. Điểm hướng dẫn ():

**GIẢNG VIÊN HƯỚNG DẪN**

(Ký, ghi rõ họ tên)

**LỜI CẢM ƠN**

Kính thưa quý thầy cô!

Trong quá trình thực hiện đề tài **“Phát triển nền tảng phân phối dịch vụ kết nối giữa các công ty cung cấp dịch vụ và người tiêu dùng**.**”,** em đã nhận được rất nhiều sự giúp đỡ , tạo điều kiện của tập thể Ban Giám Hiệu, Khoa Công Nghệ Thông Tin, các thầy cô, cán bộ các phòng ban chức năng Học Viện Công Nghệ Bưu Chính Viễn Thông tại TP.HCM. Em xin bày tỏ lòng cảm ơn chân thành về sự giúp đỡ đó.

Đặc biệt, em xin bày tỏ lòng biết ơn sâu sắc tới ThS.Lê Minh Hóa. Thầy đã hướng dẫn tận tình, truyền đạt kiến thức, chỉ bảo em hoàn thành đề tài này.

Sau cùng, em xin kính chúc các quý thầy cô Khoa Công Nghệ Thông Tin và ThS.Lê Minh Hóa thật dồi dào sức khỏe, niềm tin để tiếp tục ươm mầm những thế hệ mai sau.

Trân trọng!

TP. Hồ Chí Minh, ngày 2 tháng 08 năm 2018

Sinh viên thực hiện

**VŨ THỊ HẢO**

**MỤC LỤC**

[MỞ ĐẦU 1](#_Toc521046937)

[CHƯƠNG I: TỔNG QUAN 2](#_Toc521046938)

[1.1 Lý do chọn đề tài. 2](#_Toc521046939)

[1.2 Tổng quan về hoạt động của các công ty kinh doanh dịch vụ tại Việt Nam nói chung và nhu cầu thị yếu của người dùng. 2](#_Toc521046940)

[1.2.1 Giới thiệu chung về các công ty kinh doanh trực tuyến tại thị trường Việt Nam 2](#_Toc521046941)

[1.2.2 Giới thiệu về nhu cầu thị trường Việt Nam 3](#_Toc521046943)

[1.2.3 Mối liên hệ giữa các công ty và khách hàng. 3](#_Toc521046945)

[1.2.4 Giới thiệu về những yêu cầu về dịch vụ mà người tiêu dùng quan tâm 3](#_Toc521046948)

[1.2.5 Giới thiệu về nghiệp vụ tổ chức sự kiện để quảng cáo sản phẩm 4](#_Toc521046949)

[1.3 Mục tiêu và những yêu cầu khi xây dựng hệ thống. 5](#_Toc521046950)

[1.3.1 Mục tiêu xây dựng hệ thống 5](#_Toc521046951)

[1.3.2 Yêu cầu đặt ra cho hệ thống 5](#_Toc521046952)

[1.3.3 Hoạt động của hệ thống 5](#_Toc521046953)

[1.3.4 Hoạt động của quản lý công ty. 5](#_Toc521046954)

[1.3.5 Hoạt động của người tiêu dùng( khách hàng). 5](#_Toc521046955)

[1.4 Phương pháp giải quyết các vấn đề. 6](#_Toc521046956)

[CHƯƠNG II: CƠ SỞ LÝ THUYẾT 5](#_Toc521046957)

[2.1 Một số khái niệm. 5](#_Toc521046958)

[2.1.1 Internet 5](#_Toc521046959)

[2.1.2 Một số khái niệm về web 5](#_Toc521046960)

[2.1.2.1 Website 5](#_Toc521046961)

[2.1.3 Phân loại website 5](#_Toc521046962)

[2.2 HTML, CSS, JAVASCRIPT. 7](#_Toc521046963)

[2.2.1 Tổng quan về HTML 7](#_Toc521046964)

[2.2.2 Tổng quan về JavaScript 8](#_Toc521046965)

[2.3 Tổng quan về CSS. 9](#_Toc521046966)

[2.3.1 Khái niệm về CSS 9](#_Toc521046967)

[2.3.2 Cách sử dụng CSS 10](#_Toc521046968)

[2.4 Tổng quan về Angular 11](#_Toc521046969)

[2.4.1 Angular JS 11](#_Toc521046970)

[2.4.2 Tổng quan về Angular 4 12](#_Toc521046971)

[2.4.3 Mô hình MVC 13](#_Toc521046973)

[2.4.4 Angular 4 được lập trình như thế nào? 14](#_Toc521046975)

[2.4.4.1 Các thành phần của AngularJs 14](#_Toc521046976)

[2.4.4.2 Modules 15](#_Toc521046977)

[2.4.4.3 Scope 16](#_Toc521046978)

[2.4.4.4 Model 16](#_Toc521046979)

[2.4.4.5 Controller 16](#_Toc521046980)

[2.4.4.6 Services 17](#_Toc521046981)

[2.4.4.7 Multiple Views and Routing 18](#_Toc521046982)

[2.5 Tổng quan về TypeScript 20](#_Toc521046983)

[2.5.1 Khái niệm TypeScript 20](#_Toc521046984)

[2.5.2 Ưu điểm TypeScript 21](#_Toc521046985)

[2.6 Tổng quan về Node Js. 21](#_Toc521046986)

[2.6.1 Cấu trúc hoạt động của NodeJs và ExpressJs 22](#_Toc521046987)

[2.6.2 Lập trình không đồng bộ 24](#_Toc521046988)

[2.6.3 Đặc điểm môi trường 24](#_Toc521046989)

[2.6.4 Module 24](#_Toc521046990)

[2.6.5 Ưu và nhược điểm 26](#_Toc521046991)

[2.7 ASP.NET MVC 27](#_Toc521046992)

[2.7.1 Tổng quan về ASP.NET MVC 27](#_Toc521046993)

[2.7.2 Tổng quan về mô hình lập trình web ASP.Net MVC 29](#_Toc521046994)

[2.7.2.1 Tổng quan về ASP.NET MVC 29](#_Toc521046995)

[Nhược điểm khi lập trình với ASP.Net web forms: 30](#_Toc521046996)

[2.7.2.2 Cơ chế hoạt động và những ưu điểm trong lập trình web với ASP.Net MVC 30](#_Toc521046997)

[2.7.2.3 Cơ chế hoạt động của ASP.Net MVC 31](#_Toc521046998)

[2.7.2.4 Ưu điểm mà ASP.Net MVC 31](#_Toc521046999)

[2.8 Hệ quản trị cơ sở dữ liệu SQL SERVER. 31](#_Toc521047000)

[2.8.1 Cơ sở dữ liệu 31](#_Toc521047001)

[2.8.2 Hệ thống quản lý cơ sở dữ liệu quan hệ (RDBMS) 32](#_Toc521047002)

[2.8.3 SQL Server 33](#_Toc521047003)

[CHƯƠNG III: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 30](#_Toc521047004)

[3.1 Tổng quan ứng dụng. 30](#_Toc521047005)

[3.1.1 Đối tượng sử dụng ứng dụng 30](#_Toc521047006)

[3.1.2 Các chức năng chính của hệ thống 30](#_Toc521047007)

[3.1.3 Mô tả các thực thể 30](#_Toc521047008)

[3.1.4 Mối quan hệ giữa các thực thể 31](#_Toc521047009)

[3.2 Thiết kế cơ sở dữ liệu. 32](#_Toc521047010)

[3.2.1 Tổng quan cơ sở dữ liệu 32](#_Toc521047011)

[3.2.2 Chi tiết cơ sở dữ liệu 33](#_Toc521047012)

[3.3 Thiết kế giao diện 33](#_Toc521047013)

[3.3.1 Thiết kế giao diện hệ thống công ty 33](#_Toc521047014)

[3.3.2 Thiết kế giao diện người tiêu dùng 36](#_Toc521047015)

[CHƯƠNG IV: XÂY DỰNG WEBSITE 40](#_Toc521047016)

[4.1 Giới thiệu các công nghệ sử dụng 40](#_Toc521047017)

[4.1.1 Các công nghệ sử dụng trong việc xây dựng website 40](#_Toc521047018)

[4.1.2 Các bước thực hiện xây dựng website 40](#_Toc521047019)

[CHƯƠNG V: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 41](#_Toc521047020)

[5.1 Kết quả đạt được. 41](#_Toc521047021)

[5.2 Hạn chế. 41](#_Toc521047022)

[5.3 Hướng phát triển 41](#_Toc521047023)

[TÀI LIỆU THAM KHẢO 42](#_Toc521047024)

**DANH MỤC CÁC BẢNG, SƠ ĐỒ, HÌNH**

[Hình 2.1.Mô hình MVC 13](#_Toc521047025)

[Hình 2.2. NodeJS 21](#_Toc521047026)

[Hình 2.3. Cấu trúc miêu tả một số thành phần của NodeJs 22](#_Toc521047027)

[Hình 2.4. Cơ chế hoạt động của NodeJs 23](#_Toc521047028)

[Hình 2.5. Mô hình cấu trúc lập trình web 27](#_Toc521047029)

[Hình 2.6. Mô hình tương tác giữa User với MVC 28](#_Toc521047030)

[Hình 2.7. Mô hình lập trình Web ASP.Net MVC 29](#_Toc521047031)

[Hình 2.8. Mô hình Model của ASP.NET 29](#_Toc521047032)

[Hình 2.9. Lịch sử ra đời từ 2009 đến 2013 của ASP.NET MVC 30](#_Toc521047033)

[Hình 2.10. Cơ chế hoạt động của mô hình ASP.Net MVC 31](#_Toc521047034)

[Hình 2.11. SQL Server 33](#_Toc521047035)

[Hình 3.1. Tổng quan cơ sở dữ liệu 32](#_Toc521047036)

[Hình 3.2. Trang quản lý hệ thống công ty 33](#_Toc521047037)

[Hình 3.3. Trang quản lý công ty 34](#_Toc521047038)

[Hình 3.5. Trang quản lý dịch vụ 35](#_Toc521047039)

[Hình 3.6. Trang xem chi tiết đơn đặt hàng 36](#_Toc521047040)

[Hình 3.7. Trang chủ 36](#_Toc521047041)

[Hình 3.8.Trang hiển thị chi tiết giỏ hàng của bạn 37](#_Toc521047042)

[Hình 3.9. Trang xác nhận thông tin để đặt hàng 37](#_Toc521047043)

[Hình 3.10. Trang hiển danh sách đơn đặt hàng 37](#_Toc521047044)

**KÍ HIỆU CÁC CỤM TỪ VIẾT TẮT**

|  |  |
| --- | --- |
| **Từ viết tắt** | **Ý nghĩa** |
| **CSDL** | **Cơ sở dữ liệu** |
| **API** | **[Application Programming Interface](https://vi.wiktionary.org/w/index.php?title=application_programming_interface&action=edit&redlink=1" \o "application programming interface (trang chưa được viết))** |
| **HTTP** | **Hypertext Transfer Protocol** |
| **HTML** | **Hypertext Markup Language** |
| **CSS** | **Cascading Style Sheet** |
| **DOM** | **Document Object Model** |
| **JSON** | **JavaScript Object Notation** |
| **MVC** | **Model-View-Controller** |
| **SPA** | **Single Page Application** |
| **DN** | **Doanh Nghiệp** |

# MỞ ĐẦU

Công nghệ thông tin là một ngành khoa học đang phát triển mạnh ứng dụng rộng rãi trên nhiều lĩnh vực khác nhau. Cùng với xu hướng phát triển của các phương tiện truyền thông như báo chí, truyền hình, radio… thì việc sử dụng Internet ngày càng phổ biến. Truy cập Internet, chúng ta có được một kho thông tin khổng lồ phục vụ cho mọi nhu cầu, mục đích của con người.

Nhận thức được nhu cầu tìm hiểu thông tin, mua bán trao đổi, giải trí của xã hội, là sự ra đời của hàng loạt các website cho các mục đích thương mại, giải trí, tin tức… Để đáp ứng được việc cập nhật thông tin hàng ngày, tình hình xã hội, chính trị, thời sự, thể thao… thì website tin tức ra đời là một nhu cầu tất yếu.

Để tạo ra môi trường cho người cung cấp các dịch vụ đến với khách hàng có nơi trao đổi thông tin, tìm hiểu rõ hơn về dịch vụ, nhu cầu mình yêu thích thì website dịch vụ thương mại giới thiệu về những loại dịch vụ được cung ứng từ các công ty tại Việt Nam là điều cần thiết. Và đó cũng là kênh thông tin cho khách hàng biết đến các hoạt động của công ty cung ứng các dịch vụ đó.

Bên cạnh việc thương mại của các công ty đến gần với khách hàng hơn thì website này còn kết hợp đưa tin tức về nhu cầu thị trường , thị yếu của người dùng cho đến cho các khách hàng được biết. Thông qua đó là muốn đưa các loại dịch vụ tốt nhất đến gần với người có nhu cầu mà lại không có thông tin một cách đáng tin cậy và đầy đủ minh chứng thực tiễn nhất.

Chính vì vậy, đề tài: **“Phát triển nền tảng phân phối dịch vụ kết nối giữa các công ty cung cấp dịch vụ và người tiêu dùng”** là một website được thiết kế và thực hiện nhằm bắt kịp các công nghệ mới và đáp ứng nhu cầu thực tế đối với người dùng.

Tuy nhiên, vì thời gian và kiến thức còn hạn chế em đã cố gắng rất nhiều để có thể hoàn thành đề tài, nhưng trong quá trình làm bài em không tránh khỏi những thiếu sót, kính mong được lời nhận xét và góp ý của thây cô.

# CHƯƠNG I: TỔNG QUAN

* 1. **Lý do chọn đề tài.**

Xã hội ngày càng phát triển, mức sống của con người càng cao và nhịp sống cũng tăng nhanh hơn. Việc nhu cầu sử dụng các loại dịch vụ thông qua trao đổi thương mại bằng Internet đã không còn xa lạ với chúng ta như những thập niên trước kia nữa. Nó đã trở thành nhu cầu , thói quen mỗi khi chúng ta cần mua bán , tìm hiểu thông tin đều nhớ đến Website Internet.

Là một fan mua bán online cùng với vốn kiến thức em được trau dồi trong quá trình ngồi ghế giảng đường Học Viện Công Nghệ Bưu Chính Viễn Thông tại Thành phố Hồ Chí Minh. Mong muốn có một website có thể đáp ứng được những nhu cầu của người tiêu dùng và với cả công ty cung ứng. Là nơi môi giới , nền tảng để những công ty đưa dịch vụ cuả công ty đến gần hơn với khách hàng , và chính khách hàng cũng có một nơi đáng tin cậy để thỏa mãn nhu cầu của họ một cahs thật tuyệt vời.

* 1. **Tổng quan về hoạt động của các công ty kinh doanh dịch vụ tại Việt Nam nói chung và nhu cầu thị yếu của người dùng.**
     1. **Giới thiệu chung về các công ty kinh doanh trực tuyến tại thị trường Việt Nam**

Theo dự báo của Hiệp hội [Thương mại điện tử](http://tapchitaichinh.vn/tags/IHRoxrDGoW5nIG3huqFpIMSRaeG7h24gdOG7rQ==/thuong-mai-dien-tu.html" \t "http://tapchitaichinh.vn/tai-chinh-kinh-doanh/_blank) Việt Nam (VECOM), đến năm 2020, thị trường bán lẻ trực tuyến của Việt Nam có thể đạt con số 10 tỷ USD, chiếm 5% tổng giá trị thị trường bán lẻ. Dự báo này cho thấy, tiềm năng [kinh doanh trực tuyến](http://tapchitaichinh.vn/tags/a2luaCBkb2FuaCB0cuG7sWMgdHV54bq_bg==/kinh-doanh-truc-tuyen.html" \t "http://tapchitaichinh.vn/tai-chinh-kinh-doanh/_blank) tại Việt Nam là rất lớn. Bài viết nhìn lại bức tranh kinh doanh trực tuyến trong năm 2016, trên cơ sở đó, đưa ra một số đề xuất cho thời gian tới**.**

Tại Việt Nam, thị trường thương mại điện tử còn non trẻ nhưng có tốc độ phát triển nhanh chóng. Theo khảo sát mới được CPA Australia công bố hôm 28/2/2017, các doanh nghiệp (DN) nhỏ Việt Nam có tốc độ tăng trưởng nhanh nhất khu vực và đặc biệt, thể hiện tốt trên phương diện nắm bắt các cơ hội trong nền kinh tế kỹ thuật số với 86% người được hỏi cho biết họ kiếm doanh thu từ bán hàng trực tuyến và 92% sử dụng truyền thông xã hội vì mục đích kinh doanh.

Kết quả khảo sát mới đây của Bizweb với trên 5.000 chủ [website](http://tapchitaichinh.vn/tags/d2Vic2l0ZQ==/website.html" \t "http://tapchitaichinh.vn/tai-chinh-kinh-doanh/_blank) vừa công bố cho thấy, so với năm 2015, năm 2016 kết quả kinh doanh online đã thực sự có bước tiến rõ rệt. Theo khảo sát của Bizweb, năm 2016, 82,1% cửa hàng kinh doanh online có sự tăng trưởng, trong đó 36% có mức tăng trưởng trên 10%, cao hơn hẳn so với tỷ lệ 24% trong năm 2015.

Trong khi đó, theo kết quả khảo sát năm 2016 của Cục Thương mại điện tử và [Công nghệ](http://tapchitaichinh.vn/tags/Y8O0bmcgbmdo4buH/cong-nghe.html" \t "http://tapchitaichinh.vn/tai-chinh-kinh-doanh/_blank) thông tin tại các DN xuất nhập khẩu, có tới 32% DN đã thiết lập quan hệ kinh doanh với đối tác nước ngoài qua kênh trực tuyến, 11% tham gia các sàn thương mại điện tử và 49% có website.

Trong khi đó, theo Công ty cổ phần Đầu tư và Công nghệ OSB (Đại lý uỷ quyền chính thức của Tập đoàn Alibaba tại Việt Nam), trong năm 2016,  nhóm DN đang có nhu cầu tham gia xuất khẩu trực tuyến đạt gần 2.400 lượt, tăng 34% so với năm 2015. Không chỉ các DN nhỏ và vừa chú trọng tới ứng dụng thương mại điện tử trong hoạt động xuất nhập khẩu mà các DN lớn cũng đang chuyển dịch sang xu hướng này, coi đây là một công cụ quan trọng trong chiến lược phát triển mở rộng thị trường.

* + 1. **Giới thiệu về nhu cầu thị trường Việt Nam**

Theo thống kê năm 2016, xu hướng bán hàng đa kênh (omni channel) bắt đầu hiện diện ở Việt Nam. Điều này được thể hiện qua sự kiện thương mại điện tử lớn nhất năm là Ngày mua sắm trực tuyến (Online Friday) được tổ chức vào thứ Sáu đầu tiên của tháng 12/2016. Phương thức bán hàng đa kênh với lợi thế kết hợp thực và ảo nhằm gia tăng sự tiện lợi cho khách hàng có ý nghĩa then chốt trong tiếp cận và gắn kết khách hàng trong thời đại kinh tế số. Năm 2016 cũng đánh dấu bước đầu sự phối hợp của một loạt các nhà cung cấp dịch vụ hỗ trợ DN xuất khẩu trực tuyến thông qua sàn giao dịch thương mại điện tử.

* + 1. **Mối liên hệ giữa các công ty và khách hàng.**

Các công ty lo lắng không có một sàn giao dịch , hay môi trường nhằm giúp đỡ các công ty đưa những dịch vụ, thông tin của họ đến với khách hàng.

Chính vì vậy, dựa theo thông tin thống kê trên ta có thể thấy nhu cầu về phát triển công ty theo hướng Website thương mại là một chiến lược rất khả dụng mở mở rộng thị trường, giúp sản phẩm , dịch vụ của công ty có một sàn giao dịch , môi trường để đến gần hơn với những người tiêu dùng đang có nhu cầu.

* + 1. **Giới thiệu về những yêu cầu về dịch vụ mà người tiêu dùng quan tâm**

Hiện tại, ba nhóm hàng được mua trực tuyến nhiều nhất tại Việt Nam là thời trang, sản phẩm chăm sóc sức khỏe – sắc đẹp và đồ điện – điện tử. Tuy nhiên, việc thanh toán trực tuyến qua điện thoại, dùng mobile banking chưa cao. Tỷ lệ này tại TP HCM và Hà Nội chỉ mới 13%. So với năm ngoái tăng được 5 điểm phần trăm và tiếp tục có xu hướng cải thiện. 16% người tiêu dùng được hỏi khẳng định sẽ dùng mobile banking trong thời gian tới.

Vậy những điều mà khách hàng quan tâm về dịch vụ mà họ cần gầm có 4 chỉ tiêu chính :

* Giá cả: Đây là vấn đề hàng đầu mà **khách hàng quan tâm** khi đưa ra các quyết định khi **mua sắm trực tuyến**, bởi nó phụ thuộc vào khả năng chi trả của khách hàng. Và việc khách hàng có sự so sánh về giá giữa các website bán hàng là điều không thể tránh khỏi.  
  Bên cạnh đó, sự minh bạch về giá cả cũng tạo nên sự tin cậy ở khách hàng. Tuy vậy, rất nhiều website bán hàng hiện nay không nêu giá cụ thể của sản phẩm mà chỉ đưa ra giá cả tượng trưng để “câu” khách hàng. Đây là sai lầm nhiều doanh nghiệp đang mắc phải, thay vào đó, hãy đưa ra mức giá trước và sau khi giảm giá để tạo kích thích việc mua sắm đối với khách hàng.
* Chất lượng : Khách hàng thường lo lắng sản phẩm , dịch vụ cung ứng đó có đúng như công ty cung ứng giới thiệu không. Chính vì vậy các công ty cần tạo dựng nên uy tín cho họ, tạo nên môi trường đáng tin cậy đối với khách hàng.
* Mô tả sản phẩm: Một điều nữa mà **khách hàng**cũng khá**quan tâm khi mua sắm trực tuyến** đó là vấn đề hàng tồn trong kho. Hãy thử đặt mình vào vị trí của khách hàng để xem xét tình huống sau. Thử nghĩ xem, bạn sẽ thấy thế nào nếu như bạn đã rất khó khăn để tìm thấy một sản phẩm ưng ý và quyết định mua nó nhưng sản phẩm đó lại hết hàng. Chắc chắn lúc đó bạn đang rất thất vọng và có ấn tượng không tốt về website bán hàng của công ty. Vì vậy mới thấy được sự cần thiết của việc hàng còn tồn trong kho để có thể đáp ứng nhu cầu của khách hàng bất cứ khi nào. Hiện tại, việc trưng bày hàng tồn trong kho đang là một trong những chiến lược mà nhà đầu tư lựa chọn để kích thích việc tăng doanh số. Chẳng hạn với thông tin về loại hàng hóa nào đó chỉ còn 1 sản phẩm trong kho mà thôi, điều này sẽ giúp khách hàng chú ý hơn và bị kích thích mua sắm hơn.
  + 1. Giới thiệu về nghiệp vụ tổ chức sự kiện để quảng cáo sản phẩm

Dựa trên nhu cầu của khách hàng và quyền lợi của các nhà cung cấp có được mà xây dựng trang web tổ chức giới thiệu các dịch vụ sản phẩm ưu đãi đến với người tiêu dùng, tạo sự kiện để thu hút mọi người có thể tìm hiểu mua bán và hưởng khuyến mại lớn.

* Sản phẩm: Đó là các dịch vụ được cung cấp bởi các nhà tài trợ.
* Sự kiện: Là những đợt khuyến mãi, hay những thông tin về những loại dịch vụ mới hấp dẫn mà người tiêu dùng đang quan tâm đến.
* Tất cả các công ty, khách hàng khi đã có tài khoản tại website đều có quyền tham gia các sự kiện và ưu đãi lớn (nếu đạt yêu cầu trúng thưởng, hưởng khuyến mại).
  1. **Mục tiêu và những yêu cầu khi xây dựng hệ thống.** 
     1. **Mục tiêu xây dựng hệ thống**
* Đáp ứng nhu cầu tạo môi trường môi giới cho các công ty để họ có một phương thức đưa thông tin , dịch vụ đến với người tiêu dùng.
* Đáp ứng môi trường đáng tin cậy , đầy đủ để có thể đáp ứng nhu cầu cho khách hàng.
* Môi trường này tập hợp nhiều công ty với nhiều dịch vụ , giúp khách hàng có một cái nhìn tổng quát nhất đối với dịch vụ họ cần, qua đó khách hàng có thể chọn được dịch vụ cần với giá rẻ nhất, chất lượng tốt nhất, phù hợp với đúng đủ kiện lựa chọn của bản thân.
  + 1. **Yêu cầu đặt ra cho hệ thống**
* Tạo sự thân thiện, tiện lợi, dễ sử dụng cho người dùng.
* Hỗ trợ cho nhiều công ty liên kết với nhau.
* Hỗ trợ các công ty quản lý các thành viên, quản lý dịch vụ tung ra thị trường, đơn hàng, giao dịch với khách hàng, các chiến lược khuyến mãi, tung thông tin mới nhằm thu hút người tiêu dùng.
* Hỗ trợ người tiêu dùng tra cứu mọi nhu cầu mà có thông tin đầy đủ nhất từ nhiều công ty, có thể chọn lựa được dịch vụ phù hợp nhất( về giá cả , chất lượng, hay địa điểm...) sau khi có kết quả tìm kiếm đầy đủ từ các công ty.
  + 1. **Hoạt động của hệ thống**

Trong hoạt động của hệ thống có 2 đối tượng và các hoạt động chính.

* Hoạt động của người tiêu dùng dịch vụ.
* Hoạt động của người quản lý của các công ty thông qua website quản lý.
  + 1. **Hoạt động của quản lý công ty.**
* Quản lý các chi nhánh của công ty mình.
* Quản lý nhân viên của công ty.
* Quản lý Tài khoản khách hàng của công ty.
* Quản lý giá cả, dịch vụ mà công ty cung ứng.
* Quản lý đơn hàng của khách hàng.
  + 1. **Hoạt động của người tiêu dùng( khách hàng).**
* Người dùng là đối tượng sử dụng hệ thống nhằm đáp ứng nhu cầu tìm kiếm thông tin của họ.
* Người dùng có thể truy cập vào website để đặt các dịch vụ theo nhu cầu của bản thân, quản lý những đơn hàng này.
* Người dùng có thể tham gia chỉnh sửa thông tin tài khoản cảu bản thân.
* Người dùng có thể đưa ra đánh giá, nhận xét về những dịch vụ mà bản thân đã sử dụng để chia sẻ với mọi người, nhằm có cái nhìn khách quan từ phía khách hàng đối với sản phẩm để công ty cung ứng dịch vụ đó có thể cái thiện duy trì những ưu điểm , khuyết điểm của nó để đưa dịch vụ càng hoàn thiện nhất, chất lượng, uy tín đến với khách hàng.
  1. **Phương pháp giải quyết các vấn đề.**

Từ các yêu cầu, phân tích hoạt động đặt ra ở đề tài, tiến hành phân tích và đưa ra phương hướng giải quyết theo các bước:

* Tìm hiểu các khái niệm liên quan đến tin tức, các yêu cầu cần thiết của một website thương mại từ đó phân tích các chức năng sẽ đưa vào trong website.
* Tìm hiểu về ngôn ngữ lập trình, môi trường, các dạng cơ sở dữ liệu từ đó đưa ra giải pháp phù hợp để tiến hành xây dựng website.

# CHƯƠNG II: CƠ SỞ LÝ THUYẾT

* 1. **Một số khái niệm.**
     1. **Internet**
        1. Lịch sử phát triển

Mạng Internet là một mạng máy tính toàn cầu, bao gồm hàng chục triệu người sử dụng. Nó được hình thành từ cuối thập kỷ 60 từ một thí nghiệm của bộ quốc phòng Mỹ. Tại thời điểm ban đầu đó là mạng ARPAnet (Advance Reseach Project Agency Network) của ban quản lý dự án nghiên cứu Quốc phòng.

* + - 1. Khái niệm

Với sự phát triển nhanh chóng của Internet như hiện nay, nhu cầu tìm kiếm thông tin của con người đã trở nên rất cần thiết. Internet và các dịch vụ internet đã mở ra cho chúng ta một cánh cổng mới, xóa đi khoảng cách địa lý và sự chênh lệch trình độ. Bạn có thể nhìn ra một thế giới mới với một máy tính cá nhân kết nối internet.

Như vậy, Internet là một mạng diện rộng (WAN) là mạng lưới của các mạng vi tính nó tập hợp các mạng máy tính trên thế giới được kết nối với nhau.

* + 1. **Một số khái niệm về web**
       1. **Website**

Website là tập hợp tất cả các file được quản lý bởi một webserver và tất cả các trang liên kết được truy tìm bằng một trang hay URL mặc định (bao gồm các file HTML, file văn bản, file đồ họa và file đa phương tiện khác).

* Web Browser

Để truy cập vào www, bạn cần một chương trình gọi là trình duyệt web (Web Browser).

Web Browser là trình ứng dụng cho phép đọc và xem các trang web. Trình duyệt là một chương trình hiển thị các tệp tin dưới dạng HTML (Hypertext Markup Language – ngôn ngữ đánh dấu siêu văn bản). Trình duyệt cũng sẽ mở trang tiếp khi nháy vào một diêu liên kết trên trang Web.

Khi vào một địa chỉ Web hay nháy vào một siêu liên kết sẽ làm phát sinh ra một chuỗi sự kiện. Trình duyệt có nhiệm vụ sao và truyền dữ liệu trang web từ máy tính lưu trữ trang web này vào máy tính của bạn. Tiếp đó trình duyệt sẽ diễn giải dữ liệu và hiển thị trang web trên màn hình cho bạn.

* + 1. **Phân loại website**

Website được chia làm 2 loại: website tĩnh và website động

* Web tĩnh:
* Trang web tĩnh thường được xây dựng bằng các ngôn ngữ HTML, DHTML…
* Trang web tĩnh thường được dùng để thiết kế các trang web có nội dung ít cần thay đổi và cập nhật.
* Website tĩnh là website chỉ bao gồm các trang web tĩnh mà không có cơ sở dữ liệu đi kèm.
* Trang web tĩnh và webstie tĩnh có ưu và nhược điểm cơ bản sau đây:

***Ưu diểm***

*Thiết kế đồ họa đẹp:* Trang web tĩnh thường được trình bày ấn tượng và cuốn hút hơn trnag web động về phần mỹ thuật đồ họa vì chúng ta có thể hoàn toàn tự do trình bày các ý tưởng về đồ họa trên toàn diện tích từng trang web tĩnh.

*Tốc độ truy cập nhanh:* Tốc độ truy cập của người dùng vào các trang web tĩnh nhanh hơn các trang web động vì không mất thời gian trong việc truy vẫn cơ sở dữ liệu như các trang web động.

*Thân thiện hơn với các máy tìm kiếm (search engine):* Bởi các địa chỉ URL của các .html, .htm… trong trang web tĩnh không chứa dấu chấm hỏi (?) như trong web động.

*Chi phí đầu tư thấp:* Chi phí xây dựng website tĩnh thấp hơn nhiều so với website động vi không phải xây dựng các cơ sở dữ liệu, lạp trình phần mềm cho website và các chi phí cho việc thuê chỗ cho cơ sở dữ liệu, chi phí yêu cầu hệ điều hành tương thích (nếu có) và nó không cần nhiều kinh phí cho mỗi lần nâng cấp.

***Nhược điểm***

*Khó khăn trong việc thay đổi và cập nhật thông tin:* Muốn thay đổi và cập nhật nội dung thông tin của trang website tĩnh, bạn cần phải biết về ngôn ngữ html, sử dụng được các chương trình thiết kế đồ họa và thiết kế web cũng như các chương trình cập nhật file lên server.

*Thông tin không có tính linh hoạt, không thân thiện với người dùng:* Do nội dung trên trang web tĩnh được thiết kế cố định nên khi yêu cầu về thông tin của người truy cập tang cao thì thông tin trên website tĩnh sẽ không đáp ứng được yêu cầu đó.

*Khó tích hợp, nâng cấp, mở rộng:* Khi muốn mở rộng, nâng cấp một website tĩnh hầu như phải làm mới lại website.

* Web động
  + - * Web động là thuật ngữ được dùng để chỉ những website có cở dữ liệu và được hỗ trợ bởi các phần mềm phát triển web.
        + Với web động, thông tin hiển thị được gọi ra từ một cơ sở dữ liệu khi người dùng truy vấn tới một trang web. Trang web được gửi tới trình duyệt gồm những câu chữ, hình ảnh, âm thanh hay những dữ liệu số hoặc ở dạng bảng hoặc ở nhiều hình thức khác nữa.
        + Web động thường được phát triển bằng các ngôn ngữ lập trình tiên tiến như PHO, ASP, ASP.NET, Java và sử dụng các cơ sở dữ liệu quan hệ mạnh như Access, My SQL, SQL Server, Oracle.
        + Thông tin trên web động luôn luôn mới vì nó dễ dàng được bạn thường xuyên cập nhật thông qua việc bạn sử dụng các công cụ cập nhật của các phần mềm quản trị web. Thông tin luôn được cập nhật trong một cơ sở dữ liệu và nguời dùng Internet có thể xem những chỉnh sửa đó ngay lập tức. Vì vậy website được hỗ trợ bởi cơ sở dữ liệu là phương tiện trao đổi thông tin nhanh nhất với người dùng Internet. Điều dễ nhận thấy là những website thường xuyên được cập nhật sẽ thu hút nhiều bạn đọc hơn.
        + Web động có tính tương tác với người dùng cao. Với web động, bạn hoàn toàn có thể dễ dàng quạn trị nội dung và điều hành website của mình thông qua các phần mềm hỗ trợ.
        + Song web động có nhược điểm là với người quản trị không những phải biết về HTML mà còn phải hiểu một vài ngôn ngữ khác để truy cập được cở sở dữ liệu của trang web như ASP, PHP…
  1. **HTML, CSS, JAVASCRIPT.**
     1. **Tổng quan về HTML**

**Giới thiệu về HTML**

* Căn bản về HTML:

HyperText Markup Language. Đây là một ngôn ngữ đơn giản được sử dụng trong các tài liệu siêu văn bản. Nó cũng chỉ là các tài liệu văn bản bình thường nhưng có chức các thành phần đặc biệt gọi là các thẻ (hoặc các cặp thẻ) đánh dấu. Dựa theo các cặp thẻ này mà trình duyệt có thể biết được nó phải thực hiện cái gì. Bản chất HTML không phải là một ngôn ngữ lập trình nên các cũng chẳng cần lo lắng đến những thuật toán lằng nhằng. Nó chỉ là một “ngôn ngữ” để đánh dấu văn bản thôi.

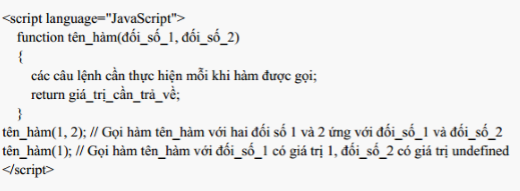
* Cách sử dụng mẫu biểu trong HTML
* Một biểu mẫu (form) trong HTML bao gồm nhiều thành phần khác nhau. Các thành phần có thể là ô văn bản, ô kéo thả, ô danh sách, nút, nút bấm, hay các ô check. Mẫu biểu được bắt đầu bằng thẻ <form> và kết thúc bởi thẻ </form>. Giữa 2 cặp thẻ này, các bạn có thể sử dụng các cặp thẻ HTML khác.
* Thẻ form có một số thuộc tính sau: Thuộc tính này có 2 giá trị POST hoặc GET, để xác định dữ liệu gửi lên theo kiểu POST hay GET. Kiểu GET chính là kiểu mà khi nhập dữ liệu lên máy chủ, các dữ liệu này sẽ được hiển thị trên ô Address dưới dạng các cặp tên giá trị. Nhược điểm của kiểu này là toàn bộ URL và xâu tên giá trị kia sẽ bị giới hạn dưới 255 kí tự. Vì vậy để có thể gửi nhiều dữ liệu hơn, người ta sinh ra kiểu POST. Với kiểu này, dữ liệu sẽ không bị giới hạn chiều dài 255 ký tự của chuỗi địa chỉ do không bị gộp vào chuỗi địa chỉ. Kiểu POST cũng thường dùng để truyền các dữ liệu nhạy cảm mà người dùng không muốn hiển thị ở trên ô Address.
  + 1. **Tổng quan về JavaScript**

Cùng thời điểm Netscape bắt đầu sử dụng công nghệ Java trên trình duyệt Netscape. LiveScript đã được đổi tên thành JavaScipt để được chú ý hơn bởi ngôn ngữ lập trình Java lúc đó đang được coi là một hiện tượng. JavaScript được bổ sung vào trình duyệt Netscape bắt đầu vào tháng 10 năm 1995. Trên thực tế, hai ngôn ngữ lập trình Java và JavaScript không có liên quan đến nhau, ngoại trừ cú pháp của cả hai ngôn ngữ cùng được phát triển dựa trên cú phấp của C. JavaScript gồm 2 mảng là client-server thực hiện lệnh trên máy của end-user và web-server.

Sau thành công của JavaScript, Microsoft bắt đầu phát triển Jscript, một ngôn ngữ có cùng ứng dụng và tương thích với JavaScript. Jscript được bổ sung vào trình duyệt Internet Explorer bắt đầu từ Internet Explorer phiên bản 3.0 được phát hành tháng 8 năm 1996.

DOM (Document Object Model), một khái niệm thường được nhắc đến với JavaScript trên thực tế không phải một phần của chuẩn ECMAScript, DOM là một chuẩn riêng biệt có liên quan chặt chẽ với XML.

Hàm là một khối các lệnh với một danh sách một hoặc nhiều đối số (có thể không có đối số) và thường có tên (mặc dù trong JavaScript hàm không nhất thiết phải có tên). Hàm có thể trả lại một giá trị. Cú pháp của hàm như sau:



Trong JavaScript, khi gọi hàm không nhất thiết phải gọi hàm với cùng đối số như khi định nghĩa hàm, nếu đối số ít hơn khi định nghĩa hàm, nhưng đối số không được chuyển cho hàm sẽ mang giá trị underfined. Các kiểu cơ bản được chuyển vào hàm theo giá trị, đối tượng sẽ được chuyển vào hàm theo tham chiếu.

Hàm là đối tượng hạng nhất trong JavaScript. Tất cả các hàm là đối tượng của nguyên mẫu Function. Hàm có thể được tạo ra và dùng trong phép toán gán như bất kỳ một đối tượng nào khác, và cũng có thể được dùng làm đối số cho các hàm khác. Do đó, JavaScript hỗ trợ hàm cấp độ cao.

* 1. **Tổng quan về CSS.**
     1. **Khái niệm về CSS**
* *CSS là gì?:* CSS: Cascading Style Sheet: Đây là những mẫu để quy định cách thức thể hiện các thẻ HTML.
* *CSS có 3 cách sử dụng:*
* Sử dụng trực tiếp kèm với các thẻ HTML (Inline Style Sheet)
* Định nghĩa trong 1 trang web (Internal Style Sheet)
* Định nghĩa thành 1 file CSS riêng (External Style Sheet)

Style được đưa vào HTML 4.0 để giải quyết một số vấn đề. CSS giúp bạn tiết kiệm được rất nhiều thời gian và công sức cho việc thiết kế web. Bạn có thể định nghĩa nhiều style vào một thẻ HTML Style (mẫu định dạng) giải quyết một số vấn đề chung: Ta biết rằng các thẻ HTML chuẩn được thiết kế để định nghĩa nội dung của một văn bản. Đầu tiên các thẻ HTML hỗ trợ cách viết "Đây là tiêu đề", "Đây là đoạn", "Đây là bảng".... bằng cách sử dụng các thẻ <H1>, <P>, <TABLE>... Cách bố trí văn bản này được qui định bởi trình duyệt web và không có bất cứ một thẻ nào để định dạng văn bản.

Đến các trình duyệt thế hệ sau đặc biệt là Nescape và Internet Explorer tiếp tục đưa thêm vào các thẻ HTML mới cùng các thuộc tính định dạng riêng của mình (như các thẻ <FONT> và thuộc tính Color...). Do đó ngày càng khó để tạo ra được một web site khi mà nội dung của nó bị tách rời khỏi cách bố trí.

Để giải quyết vấn đề này World Wide Web Consortium (W3C) đã tạo ra STYLE để đưa thêm vào trong HTML 4.0. Cả hai trình duyệt lớn là Nescape và Internet Explorer đều hỗ trợ CSS. CSS giúp bạn tiết kiệm được rất nhiều thời gian và công sức cho việc thiết kế web.

Style trong phiên bản HTML 4.0 trở lên qui định cách thức thể hiện các thẻ HTML tương tự như thẻ <FONT> hay thuộc tính COLOR trong HTML 3.2. Style thường được lưu trong các file nằm ngoài trang web. Chúng giúp bạn có thể thay đổi cách thức định dạng và cách bố trí các trang web chỉ bằng cách thay đổi riêng file CSS. Chỉ khi bạn muốn thay đổi lại toàn bộ màu sắc, cách định dạng của các tiêu đề, nội dung bạn mới hiểu rõ tác dụng thực sự của CSS.

CSS cho phép chúng ta điều khiển cách định dạng và cách bố trí của cùng lúc nhiều trang web với chỉ duy nhất 1 lần thay đổi tại một vị trí. Là một người thiết kế web, bạn có thể định nghĩa 1 file CSS cho các thẻ HTML và áp dụng nó vào nhiều trang web mà bạn muốn. Để thay đổi tổng thể các trang web này bạn chỉ đơn giản là thay đổi file CSS và tất cả các trang đã áp dụng sẽ được thay đổi một cách tự động. Bạn có thể định nghĩa nhiều style vào một thẻ HTML.

CSS cho phép bạn đưa các thông tin định nghĩa thẻ thông qua nhiều con đường khác nhau. Style có thể được qui định ở trong chỉ một thẻ HTML, được qui định trong một trang web hoặc ở trong một file CSS bên ngoài. Thứ tự áp dụng các định dạng. Như trên đã nói, ta có thể sử dụng nhiều cách khác nhau để làm CSS. Điều gì sẽ xảy ra nếu bạn áp dụng nhiều cách định dạng cho 1 thẻ HTML? Theo một cách chung nhất ra có thể nói các style của bạn sẽ được "xếp tầng" (cascade). Việc xếp tầng này tuân theo thứ tự sau: (Số 1 là ưu tiên nhất, số 4 là kém ưu tiên nhất).  
 1. Inline Style (Style được qui định trong 1 thẻ HTML cụ thể).  
 2. Internal Style (Style được qui định trong phần <HEAD> của 1

trang HTML ).

3. External Style (style được qui định trong file CSS ngoài).  
 4. Browser Default (thiết lập mặc định của trình duyệt). Như vậy ta thấy các thiết lập trong 1 thẻ HTML có mức ưu tiên cao nhất, Những gì được định nghĩa ở đây sẽ bị bỏ qua tất cả các định nghĩa khác (như trong thẻ <HEAD>, File CSS ngoài,...).

* + 1. **Cách sử dụng CSS**

Khi trình duyệt web đọc trang web của bạn nó sẽ định dạng trang web theo cách CSS đã qui định cho nó.

Chúng ta có tất cả 3 loại CSS dó đó chúng ta cũng sẽ có 3 cách để chèn CSS vào trang web của mình.

* Với External Style ( Sử dụng file CSS được định nghĩa thành 1 file riêng). Sử dụng External Style là một cách lý tưởng khi ta cần phải định dạng nhiều trang web theo một mẫu thống nhất.
* Với External Style chúng ta có thể thay đổi dáng vẻ của một trang web chỉ với việc thay đổi duy nhất 1 file. Mỗi trang web sử dụng file CSS ngoài này đều phải sử dụng thẻ <LINK>. Thẻ <LINK> được đặt bên trong thẻ  
  <HEAD>, với thuộc tính href sẽ trỏ đến file .CSS bên ngoài.
  1. **Tổng quan về Angular** 
     1. **Angular JS**

AngularJS là một framework ứng dụng web mã nguồn mở. Nó được phát triển lần đầu năm 2009 bởi Misko Hevery và Adam Abrons. Hiện tại nó được duy trì bởi Google.

AngularJS là một framework có cấu trúc cho các ứng dụng web động. Nó cho phép bạn sử dụng HTML như là ngôn ngữ mẫu và cho phép bạn mở rộng cú pháp của HTML để diễn đạt các thành phần ứng dụng của bạn một cách rõ ràng và súc tích. Hai tính năng cốt lõi: Data binding và Dependency injection của AngularJS loại bỏ phần lớn code mà bạn thường phải viết. Nó chạy trên tất cả các trình duyệt, làm cho nó trở thành đối tác lý tưởng của bất kỳ công nghệ Server nào.

Các tính năng chung của AngularJS:

* AngularJS là một Framework phát triển mạnh mẽ dựa trên JavaScript để tạo các ứng dụng RICH Internet Application (RIA).
* AngularJS cung cấp cho lập trình viên những tùy chọn để viết các ứng dụng client-side trong mô hình MVC (Model View Controller) một cách rõ ràng.
* Các ứng dụng được viết bởi AngularJS tương thích với nhiều phiên bản trình duyệt web. AngularJS tự động xử lý mã JavaScript để phù hợp với mỗi trình duyệt.
* AngularJS có mã nguồn mở, miễn phí hoàn toàn, được sử dụng bởi hàng ngàn lập trình viên trên thế giới. Nó hoạt động dưới giấy phép Apache License version 2.0.

Ưu điểm của AngularJS:

* AngularJS cung cấp khả năng tạo ra các Single Page Application một cách rất rõ ràng và dễ dàng để duy trì.
* AngularJS cung cấp khả năng Data binding tới HTML do đó giúp người dùng cảm giác linh hoạt, thân thiện.
* AngularJS code dễ dàng khi unit test.
* AngularJS sử dụng dependency injection.
* AngularJS cung cấp khả năng tái sử dụng các component (thành phần).
* Với AngularJS, lập trình viên sẽ viết ít code hơn, với nhiều chức năng hơn.
* Với AngularJS, view là thành phần trong trang HTML thuần, trong khi controller được viết bởi JavaScript với quá trình xử lý nghiệp vụ.

Mặc dù AngularJS có thể kể đến rất nhiều các ưu điểm, nhưng đến thời điểm này, nó vẫn có một số điểm yếu sau:

* Không an toàn : Là một JavaScript framework, ứng dụng được viết bởi AngularJS không an toàn. Phải có các tính năng bảo mật và xác thực phía server sẽ giúp ứng dụng trở nên an toàn hơn.
* Nếu người sử dụng ứng dụng của bạn vô hiệu hóa JavaScript thì họ chỉ nhìn được trang cơ bản, không thấy gì thêm.
  + 1. **Tổng quan về Angular 4**

Để bắt đầu hành trình này, bạn sẽ cần có một cái nhìn toàn cảnh về Angular. Đơn giản mà nói, Angular là một Javascript framework để xây dựng những ứng dụng phía client sử dụng HTML, CSS và một ngôn ngữ lập trình như Javascript. Có thể chia thành 2 đối tượng như sau:

* Đối tượng là những bạn chưa biết gì về Angular
* Đối tượng là những bạn đã biết qua về Angular 1

Ưu điểm của ứng dụng viết bằng Angular mà không sử dụng Javascript framework khác:

* Angular làm cho HTML của chúng ta trở nên linh hoạt hơn.
* Angular làm cho code HTML của chúng ta trở nên mạnh mẽ hơn với những đặc trưng như điều kiện “if” , vòng lặp “for” và những biến địa phương “local variables”.
* Angular có cơ chế binding data mạnh mẽ, chúng ta có thể dễ dàng hiển thị các field từ data model của chúng ta, theo dõi những thay đổi và cập nhật lại từ người dùng.
* Angular hoạt động theo thiết kế module (mô dun). Ứng dụng của chúng ta sẽ được xây dựng từ những khối module độc lập, làm cho việc xây dựng sẽ dễ dàng và có thể tái sử dụng được nội dung.
* Angular hỗ trợ việc giao tiếp với những back- end service. Điều này sẽ dễ dàng cho việc tích hợp những back-end service để việc giải quyết các bài toán logic nhanh chóng
* Angular là cực kỳ phổ biến với hàng triệu nhà phát triển đang sử dụng nó.

Những ưu điểm của Angular 4 hơn những verson cũ 1, 2 của nó là:

* Angular 4 nhanh hơn Angular 1 và liên tục được nâng cao hiệu năng.
* Angular 4 hiện đại, nó sử dụng những tính năng của Javascript mới nhất (ES6, ES7) và hơn thế nữa các Class, Module, Decorators hỗ trợ nhiều trình duyệt hiện tại Edge, Chrome, Firefox và cả IE thần thánh.
* Angular 4 đơn giản hoá các API, nó có ít directive hơn nên dễ nhớ dễ học hơn, binding data đơn giản, khái niệm tổng thể cũng đơn giản hơn.

Ưu diểm của ứng dụng viết bằng Angular 4:

Angular – ứng dụng là một tập hợp những component (thành phần) và một vài services (dịch vụ) cung cấp chức năng trên những component đó.

|  |
| --- |
| **Angular Application = Component + Component + Component … + services** |

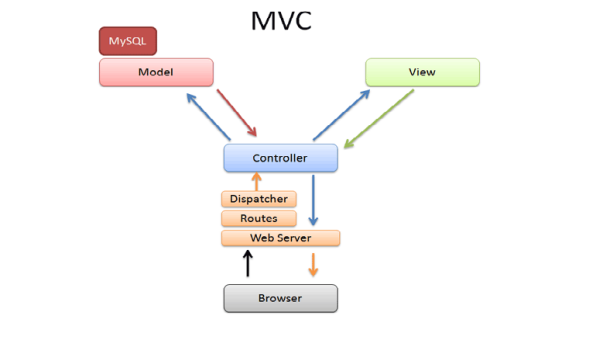
|  |
| --- |
| **Component = Template + Class + Metadata** |

* Mỗi component bao gồm một mẫu HTML cái mà sẽ render ra một “frame” giao diện người dùng.
* Một Class để code những gì liên kết với view. Class chứa những thuộc tính, những phần tử dữ liệu có sẵn để phục vụ cho các view và các phương thức thực hiện những hành động cho view, chẳng hạn như sự phản hồi của một nút bấm.
* Một component cũng có metadata cung cấp thêm những thông tin của component cho Angular. Metadata này để xác định Class là một Angular component.

Vậy tóm lại, một component sẽ hình thành từ một view xác định từ một mẫu HTML, Code xử lý sẽ được định nghĩa với một Class và thông tin bổ xung cho Class thì được xác định bằng metadata.

Chúng ta đã biết một component là gì, làm thế nào để ghép chúng lại với nhau để trở thành một ứng dụng? Chúng ta cần Angular modules (mô-dun). **Angular modules** hỗ trợ tổ chức ứng dụng Angular thành một khối gắn kết các chức năng. Mỗi ứng dụng Angular có ít nhất một module, module này gọi là Root Angular Module (mô-dun gốc). Một ứng dụng thực tế thì có thể có nhiều hơn một module.

* + 1. **Mô hình MVC**



Hình 2.1.Mô hình MVC

Mô hình MVC (Model- View- Controller) là một kiến trúc phần mềm hay mô hình thiết kế được sử dụng trong kỹ thuật phần mềm. Nó giúp cho các developer tách ứng dụng của họ ra thành 3 phần khác nhau Model, View và Controller. Mỗi thành phần có một nhiệm vụ riêng biệt và độc lập với các thành phần khác.

Mô hình MVC được giới thiệu từ những năm 70 như một phần của Smalltalk, nhưng đối với nền tảng web, thì nó mới được thịnh hành gần đây.

Ý tưởng đằng sau MVC là để chia rõ 3 thành phần chính là model (xử lý, truy xuất database), view (giao diện) và controller (điều hướng yêu cầu người dùng).

MVC thể hiện tính chuyên nghiệp trong lập trình, phân tích thiết kế. Do được chia thành các thành phần độc lập nên giúp phát triển ứng dụng nhanh, dơn giản, dễ nâng cấp, bảo trì. Đối với Angular, View sẽ là DOM, Controller là các lớp JavaScript, còn Model sẽ là dữ liệu được lưu thuộc tính của các đối tượng trong JS.

Sau khi chứng kiến nhiều tranh luận về MV\*, cho biết Angular JS là một MVW framework (Model – View – Whatever, trong đó whatever là viết tắt của whatever works for you).

* + 1. **Angular 4 được lập trình như thế nào?**
       1. **Các thành phần của AngularJs**

*Angular Template*

* Đôi lúc trong quá trình xây dựng hệ thống, file HTML trở nên phức tạp, để giải quyết vấn đề này ta cần chia thành nhiều phần khác nhau, AngularJs cung cấp cho chúng ta một giải pháp hữu ích đó là template.
* Angular template là một đặc tả dạng declarative (khai báo), cùng với thông tin từ Model & Controller, trở thành rendered View và User thấy trên trình duyệt. Nó là static DOM, chứa HTML, CSS và các thành phần, thuộc tính riêng của Angular. Các thành phần Angular và các thuộc tính giúp Angular thêm các hành vi và biến đổi template DOM thành dynamic view DOM.

*Trong Angular, chúng ta có 2 cách để tạo ra một template.*

* Dùng file ngoài: Chúng ta có thể dùng thêm một file html bên ngoài để làm template cho file chính
* Dùng script: Chúng ta có thể tích hợp thẳng template vào file hiện hành thông qua thẻ script với type là text/ng-template.

*Cách sử dụng template*

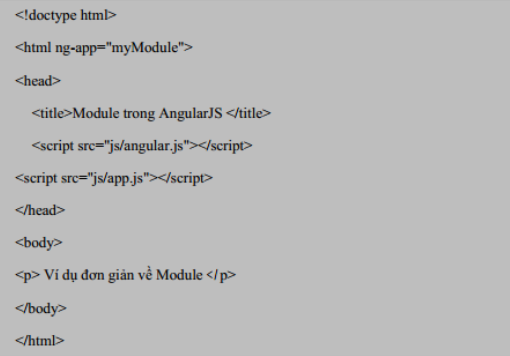
Có nhiều cách sử dụng template, tuy nhiên trong AngularJs có 2 cách thông dụng nhất để dùng đó là ng-include và ngRoute.

* + - 1. **Modules**

Trong các ứng dụng thực tế, việc phân chia ứng dụng thành các thành phần khác nhau là điều cần thiết. Dưới đây là lợi ích của việc chia nhỏ ứng dụng.

* Dễ dàng cho việc quản lý và khai báo ứng dụng cũng như kiểm tra lại.
* Khả năng tái sử dụng cũng như tận dụng cao.
* Nạp từng ứng dụng sẽ nhanh hơn là phải nạp toàn bộ ứng dụng rồi mới chạy.

Trong Angular, module được hỗ trợ trong khai báo ng-app bên cạnh khai báo nó trong mã nguồn script.



Trong ví dụ này module được khai báo ngay sau thẻ <html> với tên là myModule.

var app = angular.module(‘myModule’,[]);

* ng-app = “myModule”: Khai bào một angular app là myModule sử dụng myModule để khai báo trong script.
* Trong script, hàm angular.module() là hàm khai báo cho module.
* Cặp dấu ngoặc [] chính là mảng các module sẽ được nạp chung vào để chạy chung với ứng dụng.
* Controller trong module
* Method.controller của module sẽ đóng vai trò khai báo them một controller cho module.
* Function đại diện cho controller được khai báo bình thường giống như controller khai báo bên ngoài module.
* AngularJs sử dụng Dependence Injection để tách biệt giữa các modules.
* Các dependency được đưa vào tự động bởi framework.
* Tổ chức của một ứng dụng thực tế

**app**: thư mục chứa toàn bộ code chính của chương trình phía client.

**brower\_components**: thư mục chứa thư viện của angularJs

**image**: thư mục chứa hình ảnh

**node\_modules**: thư mục chưa các module thư viện

**index.html**: file HTML, tự động export các script mới tạo.

* + - 1. **Scope**

“$scope” là đối tượng tham chiếu tới model được sử dụng trong controller. Là thành phần gắn kết giữa views và controller. Scope cung cấp các APIs để theo dõi các thay đổi của Models: $watch

Scope cung cấp APIs để truyền bất kỳ thay đổi nào của model tới views: $apply

* + - 1. **Model**

<input type = “text” ng-model = “greeting”>

{{greeting}}

ng-model cho biết đối tượng nhận giá trị của input. {{greeting}}

sẽ in ra giá trị được nhập vào input.

* + - 1. **Controller**



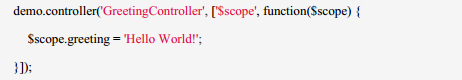
Tạo một module cho ứng dụng.



Đặt biến “demo” chứa module(‘myApp’) của ứng dụng.



Tạo một controller mới cho module này.



‘GreetingController’ là tên của controller, cách đặt tên chứa tên và đuôi “Controller” để dễ nhận biết.



Viết hàm sử dụng với biến $scope.



Gán giá trị cho biến greeting. Để in được giá trị của greeting ra trình duyệt



ng-controller cho biết controller đang hoạt động. Đoạn mã này sẽ in ra trình duyệ chuỗi.

“Hello Words!” được khai báo trong “GreetingController”.

* + - 1. **Services**

Angular service là một object hoặc một function phụ trách một tác vụ nào đó. Việc viết service chỉ là việc gom các đoạn xử lý logic vào object hoặc function để dễ quản lý hơn.

Trong AngularJs Service sẽ chứa tất cả các phần xử lý, Controller nhận yêu cầu và gọi các service cần thiết để xử lý, model handle dữ liệu từ các control, view hiển thị dữ liệu, route điều hướng request tới controller.

Bản thân AngularJs chứa những service như: $http, $scope, $window, $RouterProvider…

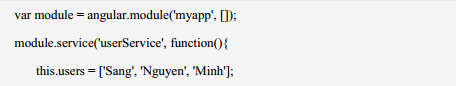
Có 2 cách tạo service trong AngularJs

**Cách 1:**

Cú pháp:

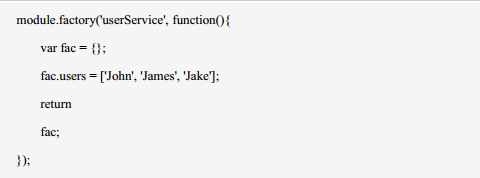


Ví dụ:



**Cách 2:**

Cú pháp:



Cả 2 cách đều tạo ra được service.

* Với cách 1 khi đăng ký tên service xong bạn sẽ được cung cấp 1 thể hiện của function mà bạn truyền vào module.service.
* Với cách 2 sau khi đăng ký tên factory, trong function truyền vào trong module.factory cần return thể hiện của đối tượng tạo bên trong nó.
  + - 1. **Multiple Views and Routing**

Đôi khi trong một trang, nhiều khi chúng ta chỉ muốn hiển thị một phần HTML ứng với mỗi chức năng cụ thể mà không cần chuyển đổi trang, Angular là một full- stack framework hiệu quả giúp chúng ta có thể làm được việc này nhanh chóng và dễ dàng. Router là bộ điều hướng các yêu cầu từ phía người dùng tới các controller tương ứng để xử lý dựa theo các đối số truyền trên thanh url. Router trong AngularJs là thành phần quan trọng giúp AngularJs tạo được ứng dụng SPA, chuyển trang mà không cần tải lại trình duyệt.

Trong AngularJs chúng ta sẽ sử dụng $routerProvider để bắt các yêu cầu.

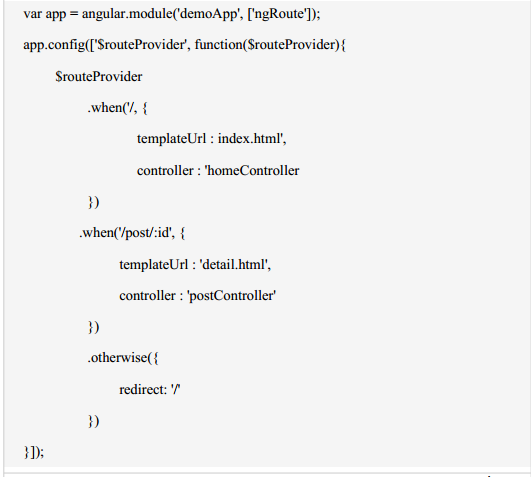


Từ phiên bản AngularJs 1.0.7 thì Router đã tách thành một file js riêng biệt, và để sử dụng được nó ta phải nhúng file angular-router.js vào ứng dụng.

Directive ng-view được sử dụng để hiển thị dữ liệu.



Đăng ký biến app thành một module trong AngularJs và thiết lập router cơ bản. Để làm việc được với Router thì cần gọi và sử dụng một extends module của angular là ngRouter.



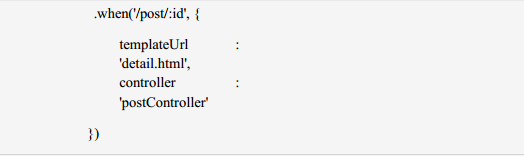
app.config là method cho phép khái báo cacs Controller, Views tương ứng với url.

Với router thế này thì khi truy cập vào đường dẫn chính cảu trang web thì trình duyệt sẽ load file index.html và blind nó vào <div ng-view></div>.



Trong router tới chi tiết bài đăng có sử dụng $routerProvider service dùng để nhận các đối số là id của bài viết, giúp xác định được chính xác bài viết cần hiển thị. Đường dẫn truy cập tới chi tiết bài đăng sẽ có dạng #post/id. Trong đó id là mã của bài đăng. Khi cấu hình router không cần phải ghi dấu “#”,

Khi gặp đường dẫn có dạng như vậy thì router sẽ gọi tới controller là postController và sử dụng template từ file detail.html



$routerProvider.otherwise xử lý cho router mặc định. Khi đường dẫn không khớp với những router đã thiết lập thì sẽ được tái điều hướng về trang chủ.



* 1. **Tổng quan về TypeScript**
     1. **Khái niệm TypeScript**

TypeScript là một dự án mã nguồn mở được phát triển bởi Microsoft, nó có thể được coi là một phiên bản nâng cao của Javascript bởi việc bổ sung tùy chọn kiểu tĩnh và lớp hướng đối tượng mà điều này không có ở Javascript. TypeScript có thể sử dụng để phát triển các ứng dụng chạy ở client-side (Angular2) và server-side (NodeJS).

TypeScript sử dụng tất cả các tính năng của của ECMAScript 2015 (ES6) như classes, modules. Không dừng lại ở đó nếu như ECMAScript 2017 ra đời thì mình tin chắc rằng TypeScript cũng sẽ nâng cấp phiên bản của mình lên để sử dụng mọi kỹ thuật mới nhất từ ECMAScript. Thực ra TypeScript không phải ra đời đầu tiên mà trước đây cũng có một số thư viện như CoffeScript và Dart được phát triển bởi Google, tuy nhiên điểm yếu là hai thư viện này sư dụng cú pháp mới hoàn toàn, điều này khác hoàn toàn với TypeScript, vì vậy tuy ra đời sau nhưng TypeScript vẫn đang nhận được sự đón nhận từ các lập trình viên.

* + 1. **Ưu điểm TypeScript**
* **Dễ phát triển dự án lớn**: Với việc sử dụng các kỹ thuật mới nhất và lập trình hướng đối tượng nên TypeScript giúp chúng ta phát triển các dự án lớn một cách dễ dàng.
* **Nhiều Framework lựa chọn**: Hiện nay các Javascript Framework đã dần khuyến khích nên sử dụng TypeScript để phát triển, ví dụ như AngularJS 2.0 và Ionic 2.0.
* **Hô trợ các tính năng của Javascript phiên bản mới nhất**: TypeScript luôn đảm bảo việc sử dụng đầy đủ các kỹ thuật mới nhất của Javascript, ví dụ như version hiện tại là ECMAScript 2015 (ES6).
* **Là mã nguồn mở**: TypeScript là một mã nguồn mở nên bạn hoàn toàn có thể sử dụng mà không mất phí, bên cạnh đó còn được cộng đồng hỗ trợ.
* **TypeScript là Javscript**: Bản chất của TypeScript là biên dịch tạo ra các đoạn mã javascript nên ban có thê chạy bất kì ở đâu miễn ở đó có hỗ trợ biên dịch Javascript. Ngoài ra bạn có thể sử dụng trộn lẫn cú pháp của Javascript vào bên trong TypeScript, điều này giúp các lập trình viên tiếp cận TypeScript dễ dàng hơn.
  1. **Tổng quan về Node Js.**

****

Hình 2.2. NodeJS

**Node.js** là một hệ thống phần mềm được thiết kế để viết các ứng dụng internet có khả năng mở rộng, đặc biệt là máy chủ web. Chương trình được viết bằng JavaScript, sử dụng kỹ thật điều khển theo sự kiện, nhập/xuất không đồng bộ để tối tiểu tổng chi phí và tối đại khả năng mở rộng. Node.js bao gồm có V8 JavaScript engine của Google, libUV, và vài thư viện khác.  
 Node.js được tạo bởi Ryan Dahl từ năm 2009, và phát triển dưới sự bảo trợ của Joyent.

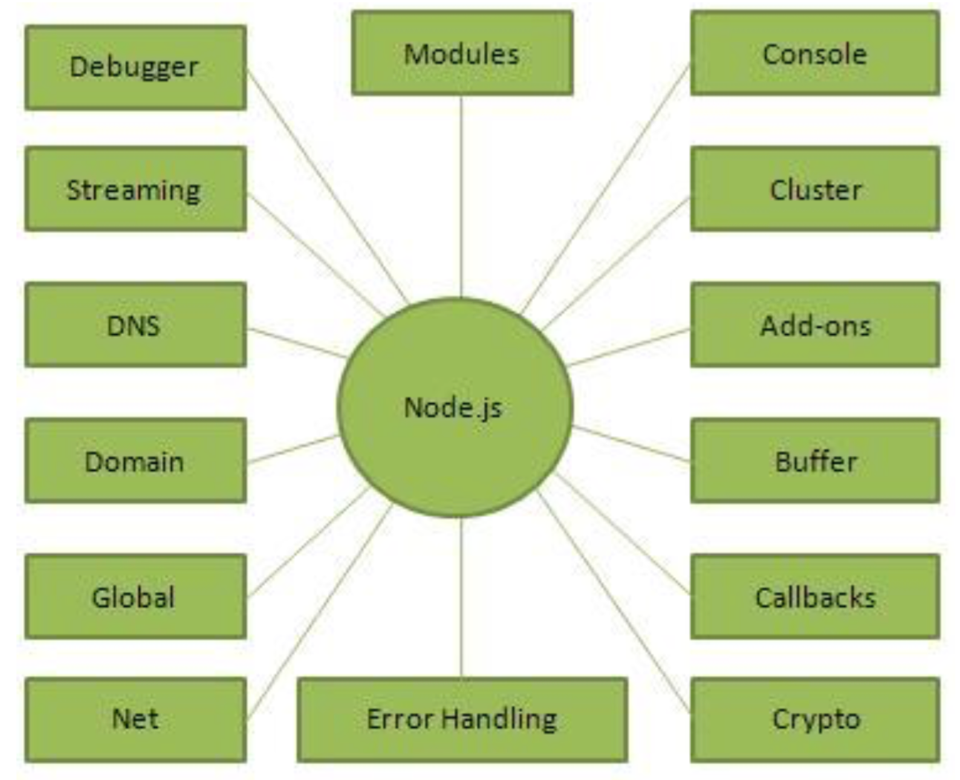
Mục tiêu ban đầu của Dahl là làm cho trang web có khả năng push như trong một số ứng dụng web như Gmail. Sau khi thử với vài ngôn ngữ Dahl chọn Javascript vì một API Nhập/Xuất không đầy đủ. Điều này cho phép anh có thể định nghĩa một quy ước Nhập/Xuất điểu khiển theo sự kiện, non-blocking.

Vài môi trường tương tự được viết trong các ngôn ngữ khác bao gồm Twisted cho Python, Perl Object Environment cho Perl, libevent cho C và EventMachine cho Ruby. Khác với hầu hết các chương trình Javascript, Nodejs không chạy trên một trình duyệt mà chạy trên Server. Node.js sử dụng nhiều chi tiết kỹ thuật của CommonJS. Nó cung cấp một môi trường REPL cho kiểm thử tương tác.

Node.js được *InfoWorld* bình chọn là “Công nghệ của năm” năm 2012.

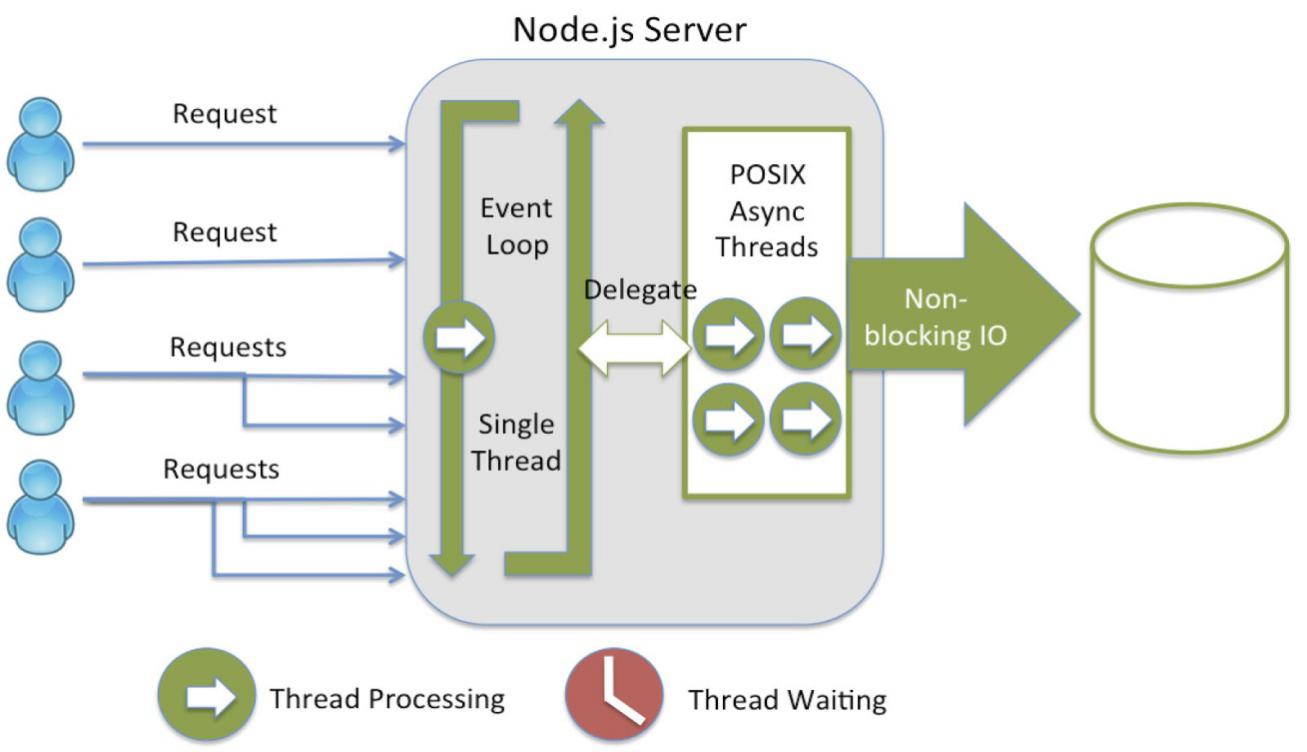
Để bắt đầu dùng Node.js, bạn phải hiểu sự khác nhau giữa Node.js với các môi trường truyền thống chạy trên server (server side) phổ biến như PHP, Python, Ruby, etc.

* + 1. **Cấu trúc hoạt động của NodeJs và ExpressJs**



Hình 2.3. Cấu trúc miêu tả một số thành phần của NodeJs

* **Debugger:** hộ trợ lập trình viên debug chương trình.
* **Streaming:** hổ trợ xử lý luồng dữ liệu.
* **DNS:** hỗ trợ cho việc xử lý liên quan đến tra cứu DNS.
* **Domain:** hỗ trợ cho việc xử lý nhiều hoạt động I/O như một nhóm duy nhất.
* **Global:** hỗ trợ việc khai báo một đối tượng toàn cục. Biến này có thể được sử dụng trực tiếp trong mọi module mà không cần phải include.
* **Net:** hỗ trợ cho cả server lẫn client dưới dạng luồng và hoạt động như một nút mạng.
* **Error handling:** kiểm soát và cho biết được những lỗi trong khi thực thi.
* **Crypto:** hỗ trợ cho việc giải mã và mã hóa.
* **Callbacks:** hỗ trợ thực thi những phương thức một cách bất đồng bộ.
* **Buffer:**  hỗ trợ việc quan lý và phân bổ các vùng nhớ bên ngoài môi trường V8 engine.
* **Add-ons:** hỗ trợ việc tích hợp thêm những module ngoài hệ thống.
* **Cluster:** NodeJS hoạt động đơn luồng nên để tận dụng được lợi thế của hệ thống đa lõi đôi lúc ta cần phải chạy một cụm các luồng NodeJS để xử lý. Cluster sẽ hổ trợ làm công việc này.
* **Console:** cung cấp giao diện cơ bản để in thông tin lỗi, hoặc dữ liệu nào đó giống như cơ chể JavaScript ở các trình duyệt.

**Cơ chế hoạt động của NodeJs**

Hình 2.4. Cơ chế hoạt động của NodeJs

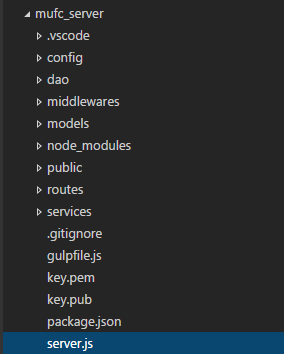
* Tiếp nhận request (yêu cầu)
* Vòng lặp tiếp nhận sự kiện và phản ứng lại theo cơ chế bất đồng bộ. Do đó sẽ non-blocking I/O (không chiếm quyền lấy, truyền và xử lý dữ liệu).
* Do cơ chế hoạt động non-blocking I/O nên chương trình sẽ không có thời gian chờ, mà tất cả tác vụ sẽ thực hiện bất đồng bộ ở chế độ background processing.
  + 1. **Lập trình không đồng bộ**

Là lợi thế nếu bạn đã quen thuộc với các phương pháp lập trình không đồng bộ. Tất cả các hàm trong Nodejs là không đồng bộ. Do đó, tất cả chạy như các block thread thông thường thay vì chạy nền. Đây là điều quan trọng nhất để nhớ về Nodejs. Ví dụ, nếu bạn đang đọc một tập tin trên hệ thống tập tin, bạn phải chỉ định một chức năng gọi lại, đó là thực hiện khi đã hoàn thành các hoặt động.

* + 1. **Đặc điểm môi trường**

Nodejs chỉ là môi trường – điều đó có nghĩa là bạn phải tự làm tất cả. Đó phải là một server http ngầm định hoặc là bất cứ server nào khác. Điều này có thể là khó hiểu với người mới, nhưng thành công thật sự của nó là đưa lại một hiệu năng đáng kể. Một script có thể điều phối mọi kết nối với các client. Điều này làm sử dụng ít tài nguyên đưa đến một hiệu quả rất cao.

* + 1. **Module**



* + - **config:** thư mục chứa file config server và file kết nối đến hệ quản trị cơ sở dữ liệu MySQL.
    - **dao:** thư mục chứa các module xử lý request từ client.
    - **middlewares:** thư mục chứa error- handler, giúp người dùng biết mình bị lỗi gì khi thao tác
    - **models:** thư mục chứa các module lấy cơ sở dữ liệu được yêu cầu.
    - **node\_modules:** thư mục chứa cácmodule thư viện.
    - **public:** thư mục chứa hình ảnh
    - **routes:** thư mục chứa các module router (định tuyến luồng xử lý).
    - **package.json:** cấu hình project phía server, các module sử dụng.
    - **sever.js:** cấu hình khởi tạo và sử dụng các module.

Nodejs sử dụng một kiến trúc mô-đun để đơn giản hóa việc tạo ra các ứng dụng phức tạp. Mô-đun giống như các thư viện trong C, hoặc các đơn vị trong Pascal. Mỗi module có chứa một tập hợp các chức năng liên quan đến đối tượng của các mô-đun. Ví dụ, các mô-đun chưa http chứa các chức năng cụ thể cho HTTP. Nodejs cung cấp một vài mô-đun cơ bản để giúp bạn truy cập các tập tin trên hệ thống tập tin, tạo ra trình điều khiển server HTTP bà TCP/UDP và thực hiện các chức năng hữu ích.

Để gọi hàm module, chỉ cần gọi hàm require() như sau

Var http = require(‘http’);

Hàm require() trả về tham chiếu đến các module quy định. Trong trường hợp của mã này, một tham chiếu đến các module http được lưu trữ trong biến http. Trong đoạn code trên, ta đã truyền tên của module vảo trong hàm require(). Việc này chi định cho Node.js tìm trong thư mục node\_modules module tương ứng để thực hiện. Nếu Node không thấy module tương ứng trong thư mục thì nó sẽ tim trên global module cache. Bạn cũng có thể chỉ định một module qua một file vật lý qua đường dẫn tương đối hay tuyệt đối như sau:

Var myModule = require(‘./myModule.js’);

Module được đóng gói từng phần mã. Đoạn mà nằm trong một mô-đun chủ yếu là private - có nghĩa là các chức năng và biến được định nghĩa trong họ chỉ có thể truy cập từ bên trong của các mô-đun. Tuy nhiên, bạn có thể tiếp xúc với chức năng và / hoặc các biến được sử dụng từ bên ngoài của mô-đun. Để làm như vậy. phải sử dụng các đối tượng export với các thuộc tính và phương thức cũa nó với từng phần mã bạn muốn gọi từ bên ngoài. Hãy xem xét các module vi dụ sau đây:

var PI = Math.PI;

exports.area = function (r) {

return PI \* r \* r;

};

Exports.circumference = function (r) {

return 2 \* PI \* r;

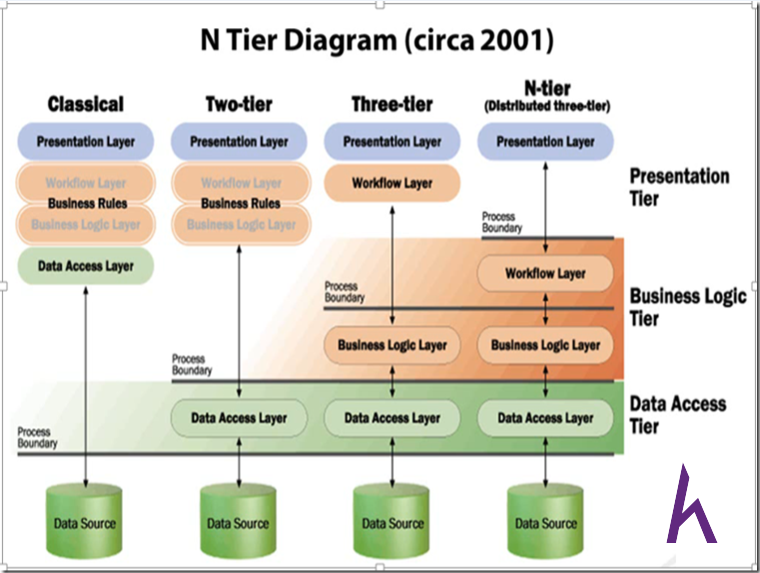
};

Trong ví dụ này PI là biến private và chỉ được sử dụng bên trong đoạn mã, trong đó có hai hàm area() và circumference() được từ khóa exports chỉ định thì sẽ có thể truy cập được từ bên ngoài.

* + 1. **Ưu và nhược điểm**

**Tại sao lại sử dụng NodeJS?**

* + **Ưu điểm:**
    - Điều đầu tiên nhắc đến là ưu điểm về tốc độ thực thi và khả năng mở rộng. NodeJS nhận và xử lý nhiều kết nối với chỉ một single-thread. Điều này giúp hệ thống tốn ít RAM nhất và chạy nhanh nhất khi không phải tạo thread mới cho mỗi truy vấn giống PHP. Ngoài ra, tận dụng ưu điểm non-blocking I/O của JavaScript, NodeJS có thể tận dụng tối đa tài nguyên của server mà không tạo ra độ trễ nhưu PHP.
    - JSON API với cơ chế hướng sự kiện (event-driven), non-blocking I/O (Input/Output) và mô hình kết hợp với JavaScript là sự lựa chọn tuyệt vời cho các dịch vụ Webs sử dụng kiểu dữ liệu JSON.
    - Khả năng xứ lý nhiều Request/s đồng thời phản hồi nhanh.
    - Streamming Data (luồng dữ liệu). Các Web thông thường gửi HTTP request và nhận phản hồi lại. Nếu như sẽ cần xử lý một luồng dữ liệu cực lớn, NodeJS sẽ xây dựng các Proxy phân vùng các luồng dữ liệu để đảm bảo tối đa hoạt động cho các luồng dữ liệu khác.
  + **Nhược điểm:**
    - Một ứng dụng nặng tốn tài nguyên như encoding video, convert file, decoding encryption…NodeJS hay bị rơi vào trường hợi thắt cổ chai khi làm việc những file dung lượng lớn.
    - Giống như hầu hết các công nghệ mới việc triển khai NodeJS trên host không phải là điều dễ dàng. Nếu sử dụng hosting xài chung, thì không hề đơn gian là tải lên một ứng dụng NodeJS và mong chờ nó sẽ hoạt động. Do đó, VPS hay dedicated server là một sự lựa chọn tốt hơn.
    - Một nhược điểm lớn của NodeJS đó là nó vẫn đang trong giai đoạn phát triển, có nghĩa là một số đặc trưng, API có thể sẽ thay đổi một cách tương thích ngược trong quá trình phát triền tiếp theo. Do đó, cần phải cẩn trọng và hiểu rõ những API cần sử dụng với những dự án lớn và lâu dài.
  1. **ASP.NET MVC**
     1. **Tổng quan về ASP.NET MVC**



Hình 2.5. Mô hình cấu trúc lập trình web

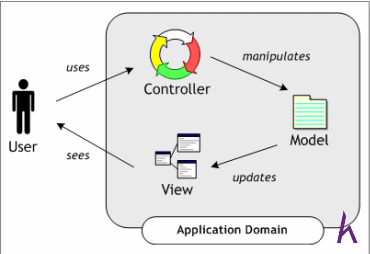
Trong lập trình web chúng ta chia ra 3 tầng như sau:

1. **Presentation Layer (**tầng 3): tầng này giúp hiển thị giao diện
2. **Business Logic Layer** (tầng 2): tầng này chứa các hàm thực thi, giúp xử lý sự kiện xảy ra trên tầng Presentation Layer.
3. **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 …

* Theo như hình mô tả, cấu trúc cổ điển là 3 tầng này trộn vào nhau hay còn gọi là cấu trúc One – Tier. Khi làm việc với cấu trúc One – Tier chúng ta gặp khó khăn trong việc chỉnh sửa code hay chỉnh sửa giao diện vì các tầng này bi trộn vào nhau.
* Tiếp đó, cấu trúc Two – Tier ra đời, tuy nhiên vẫn chưa giải quyết vấn đề của cấu trúc One – Tier gặp phải đó là việc chỉnh sửa code hay giao diện gặp khó khăn vì tầng 2 và tầng 3 vẫn chưa tách biệt khỏi nhau.
* Vì vậy câu trúc Three – Tier đã ra đời và giải quyết tất cả các vần đề mà One – Tier, Two – Tier gặp phải. Lúc này 3 tầng đã tách biệt nhau ra do đó việc chỉnh sửa code hay giao diện sẽ trở nên dễ dàng hơn giúp cho ứng dụng tối ưu hóa nhất

Sau khi hiểu và nhớ lại cấu trúc của một mô hình lập trình web cơ bản thì chúng ta sẽ đi qua tìm hiểu về mô hình MVC

* Minh họa mô hình MVC:



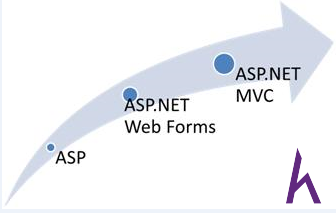
Hình 2.6. Mô hình tương tác giữa User với MVC

**MVC** (Model – View - Controller) là một design partern đã tồn tại rất lâu trong ngành công nghệ phần mềm. Một ứng dụng viết theo mô hình MVC sẽ bao gồm 3 thành phần tách biệt nhau đó là Model, View, Controller. Giống như trong cấu trúc Three – Tier, mô hình MVC giúp tách biệt 3 tầng trong mô hình lập trình web, vì vậy giúp tối ưu ứng dụng, dễ dàng thêm mới và chỉnh sửa code hoặc giao diện

* Model: ở phần trước mình đã nhắc lại cho các bạn về 3 tầng trong mô hình
* Three – Tier thì trong đó gồm có 2 tầng 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 và tầng view trong mô hình MVC, 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

**Ví dụ: User yêu cầu hiển thị thông tin cá nhân của user**

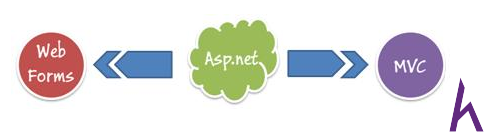
1. User gửi một yêu cầu tới controller.
2. Controller nhận yêu cầu, xử lý yêu cầu, nếu yêu cầu cần truy xuất dữ liệu thì controller sẽ gửi yêu đó xuống tầng model để truy xuất dữ liệu
3. Tầng model sẽ lấy dữ liệu từ database sau đó truyền dữ liệu qua tầng view thông qua tầng controller để tầng view hiển thị dữ liệu cho User
4. User sẽ thấy thông tin hiển thị ở giao diện và cụ thể ở đây là tầng view
   * 1. **Tổng quan về mô hình lập trình web ASP.Net MVC**
        1. **Tổng quan về ASP.NET MVC**



Hình 2.7. Mô hình lập trình Web ASP.Net MVC

Hình mô tả phía trên cho các bạn thấy trước khi **ASP.Net MVC** ra đời thì ASP.Net webform là 1 framework lập trình web khá quen thuộc đối với những tín đồ của **ASP.Net**. Tuy nhiên, khi **ASP.Net MVC** xuất hiện thì dường như ASP.Net webform bị lãng quên. Để hiểu được lý do tạo sao lại như vậy thì chúng ta sẽ đi từng bước tìm hiểu nhé!

ASP.Net là gì



Hình 2.8. Mô hình Model của ASP.NET

**ASP.NET**là 1 framwork lập trình web được cung cấp bởi Microsoft và nó sử dụng [ngôn ngữ C#](https://www.howkteam.vn/redirect?Id=Y/HKZv6fbvNJnzXtx4Ic0v/nhq7uuXcoIaJELghl1NaOB+LvlFbFuWsdaEQDP3dhhkITUiNcue1z6QBgn59UQw==" \t "https://www.howkteam.vn/Course/Lap-trinh-website-voi-ASPNet-MVC-co-ban/_blank) làm ngôn ngữ phát triển. ASP.NETbao gồm 2 model đó là Web Forms và MVC.

### Nhược điểm khi lập trình với ASP.Net web forms:

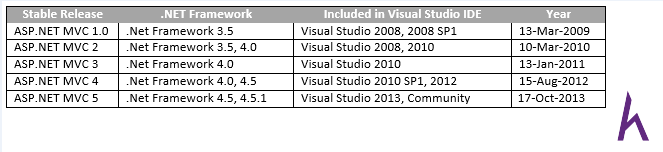
Trong nội dung bài ngày hôm nay, mình tập trung giới thiệu **ASP.Net MVC** nên mình sẽ không đi vào cụ thể về Web Forms mà mình chỉ đưa ra 1 vài nhược điểm của Web Forms.

* Web Forms không có sự tách biệt rõ ràng 3 tầng trong mô hình lập trình web.
* Trong Web Forms có sử dụng 1 đối tượng View State mà đối tượng này làm cho trang web bị tăng kích thước vì vậy làm giảm hiệu năng của ứng dụng.
* Giao diện của Web Forms được thiết kế dựa trên những toolbox có sẵn mà web forms cung cấp mà hiện nay đa phần designer sử dụng html, css để thiết kế website vì nó gây khó khăn trong việc thiết kế web hiện nay, việc thiết kế trở nên mất linh hoạt.

Với những khó khăn như vậy thì Microsoft đã cho ra đời ASP.Net MVC.

* + - 1. **Cơ chế hoạt động và những ưu điểm trong lập trình web với ASP.Net MVC**

**ASP.Net MVC**là 1 framework lập trình web mới của microsoft, công nghệ này ứng dụng mô hình MVC vào trong ASP.Net.



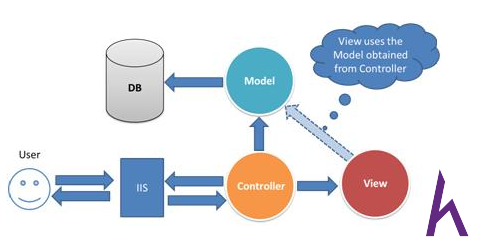
Hình 2.9. Lịch sử ra đời từ 2009 đến 2013 của ASP.NET MVC

Qua hình mô tả phía trên cho các bạn thấy tuy ASP.Net MVC ra đời năm 2009 nhưng mà tính tới năm 2013 thì nó đã được update lên tới phiên bản **ASP.Net MVC 5**, đây cũng là 1 trong những ưu điểm của ông lớn Microsoft cập nhật công nghệ liên tục.

Và mới đây thì ASP.Net cũng đã cho ra đời 1 công nghệ mới đó là ASP.Net core 1.0

Với sự cập nhật thường xuyên như vậy đã giúp cho ASP.Net MVC ngày nay phát triển mạnh mẽ không thua kém gì các công nghệ lập trình web sử dụng Java và Php.

* + - 1. **Cơ chế hoạt động của ASP.Net MVC**



Hình 2. 10. Cơ chế hoạt động của mô hình ASP.Net MVC

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.
  + - 1. **Ưu điểm mà ASP.Net MVC**

Do sử dụng mô hình MVC nên trong ASP.Net MVC đã tách biệt được các tầng trong mô hình lập trình web vì vậy 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 giảm.

* 1. **Hệ quản trị cơ sở dữ liệu SQL SERVER.**
     1. **Cơ sở dữ liệu**

Một Database (Cơ sở dữ liệu) là một ứng dụng riêng rẽ mà lưu trữ một tập hợp dữ liệu. Mỗi cơ sở dữ liệu có một hoặc nhiều API riêng biệt để tạo, truy cập, quản lý, tìm kiếm và tái tạo dữ liệu nó đang giữ.

Một số loại kho lưu dữ liệu khác có thể được sử dụng, chẳng hạn như file trên hệ thống file hoặc các Hash Table lớn, nhưng việc lấy và ghi dữ liệu không thể nhanh và dễ dàng với các loại kho lưu dữ liệu này của các hệ thống.

Vì thế, ngày nay, chúng ta sử dụng các Hệ thống quản lý cơ sở dữ liệu quan hệ (RDBMS) lưu giữ và quản lý khối lượng lớn dữ liệu. Nó được gọi là cơ sở dữ liệu quan hệ, bởi vì tất cả dữ liệu được lưu giữ trong các bảng dữ liệu khác nhau và các mối quan hệ được thành lập bởi sử dụng các Primary Key (khóa chính) và một số khóa khác được biết đến như là Foreign Key.

* + 1. **Hệ thống quản lý cơ sở dữ liệu quan hệ (RDBMS)**
* Cho bạn khả năng triển khai một Database với các bảng dữ liệu, cột (column), và các chỉ mục (Index).
* Bảo đảm Referential Integrity (có thể dịch là toàn vẹn quan hệ) giữa các hàng và các bảng đa dạng.
* Cập nhật tự động các chỉ mục.
* Thông dịch một truy vấn SQL và tổ hợp thông tin từ các bảng khác nhau.

Thuật ngữ RDBMS

* **Database**: Một cơ sở dữ liệu là một tập hợp các bảng dữ liệu, với dữ liệu có liên quan.
* **Bảng dữ liệu**: Một bảng là một ma trận dữ liệu. Một bảng trong một cơ sở dữ liệu trông giống như một bảng tính đơn giản.
* **Cột**: Một cột chứa cùng một kiểu dữ liệu, ví dụ như tên khách hàng.
* **Hàng**: Một hàng (row, entry, record) là một nhóm dữ liệu có liên quan.
* **Redundancy**: (có thể hiểu là dữ liệu dự phòng) Dữ liệu được lưu giữ hai lần, để làm cho hệ thống nhanh hơn.
* **Primary Key**: Một Primary Key (Khóa chính) là duy nhất. Một giá trị key không thể xuất hiện hai lần trong một bảng. Với một key, bạn có thể tìm thấy phần lớn trên một hàng.
* **Foreign Key**: Bạn tưởng tượng về Foreign Key như là cái ghim liên kết giữa hai bảng.
* **Compound Key**: Một Compound Key (hay composite key) là một key mà gồm nhiều cột, bởi vì một cột là không duy nhất.
* **Index:** Một chỉ mục trong một cơ sở dữ liệu tương tự như chỉ mục trong một cuốn sách.
* **Referential Integrity**: Đảm bảo rằng một giá trị Foreign Key luôn luôn trỏ tới một hàng đang tồn tại.
  + 1. **SQL Server**



Hình 2.11. SQL Server

SQL Server là một RDBMS nhanh và dễ dàng để sử dụng. SQL Server đang được sử dụng cho nhiều công việc kinh doanh từ lớn tới nhỏ. SQL Server được tối ưu để có thể chạy trên môi trường cơ sở dữ liệu rất lớn (Very Large Database Environment) lên đến Tera-Byte và có thể phục vụ cùng lúc cho hàng ngàn user. SQL Server có thể kết hợp “ăn ý” với các server khác như Microsoft Internet Information Server (IIS), E-Commerce Server, Proxy Server…

Một vài ấn bản của SQL Server như : Enterprise, Standard , Developer, Workgroup, Express.

SQL Server trở thành khá phổ biến vì nhiều lý do:

* **SQL** là một loại ngôn ngữ máy tính phổ biến để chỉnh sửa, tạo, truy vấn và xóa dữ liệu hay SQL dùng để quản lý dữ liệu, được sử dụng trong hầu hết các ứng dụng cần lưu dữ liệu của người dùng.Máy chủ cơ sở dữ liệu MySQL cung cấp đặc tính linh hoạt, có sức chứa để xử lý các ứng dụng được nhúng sâu với dung lượng chỉ 1MB để chạy các kho dữ liệu đồ sộ lên đến hàng terabytes thông tin. Sự linh hoạt về flatform là một đặc tính lớn của MySQL với tất cả các phiên bản của Linux, Unix, và Windows đang được hỗ trợ. Và dĩ nhiên, tính chất mã nguồn mở của MySQL cho phép sự tùy biến hoàn toàn theo ý muốn để thêm vào các yêu cầu thích hợp cho database server.
* Bằng cách đưa các thông tin này về một dạng cấu trúc rõ ràng, thống nhất và dễ hiểu SQL hỗ trợ việc lưu các thông tin phức tạp. Tuy nhiên SQL còn rất nhiều tính năng khác không dừng lại ở việc hỗ trợ lưu các thông tin phức tạp điều đó làm cho **[học SQL Server nâng cao](http://http//stanford.com.vn/dao-tao/khoa-hoc/id/12/khoa-hoc-co-so-du-lieu-sql-server)**trở thành ngôn ngữ đáng học cho bất cứ ai làm trong ngành lập trình.
* **Dữ liệu có ở mọi nơi:** Trên internet dữ liệu tồn tại ở hầu như mọi ứng dụng bạn tìm thấy. Bạn cần phải lưu dữ liệu ngay cả các ứng dụng laptop, desktop và mobile. Khi bạn đang cung cấp thông tin cho ứng dụng, bạn đăng ký tài khoản trên ứng dụng. Dù bạn nhận ra hay không khi chơi một game mobile bạn đang tương tác với thông tin lưu của game đó. Ví dụ trong game Flappy Bird bạn cần phải theo dõi xem trạng thái của chú chim mà bạn đang chơi và đưa ra những thử thách dựa trên trạng thái này.
* **Sửa, thêm, đọc và xóa dữ liệu dễ dàng.**
* **SQL được sử dụng và hỗ trợ bởi nhiều công ty lớn:** Việc nắm vững SQL là một kỹ năng là một điều vô cùng cần thiết, mặc dù SQL là ngôn ngữ khá đơn giản. Tất cả các công ty như IBM, Microsoft, Oracle về công nghệ trên thế giới hiện nay  đều hỗ trợ việc phát triển ngôn ngữ này và SQL được rất nhiều công ty lớn sử dụng.
* **SQL giúp công việc lập trình trở lên dễ dàng:** Trong ứng dụng của bạn, bạn có rất nhiều loại thông tin cần lưu và bạn muốn sử dụng nhiều file để lưu mỗi loại dữ liệu. Nhưng việc này lại khiến cho quy trình maintain không dễ dàng chút nào việc đọc dữ liệu trở lên phân tán. Ngoài ra việc quản lý dữ liệu lại càng trở lên khó khăn nếu bạn có nhiều ứng dụng khác nhau cần lưu thông tin. Trong nhiều ứng dụng khác nhau bạn có thể lưu nhiều dữ liệu cho trên cũng một cơ sở dữ liệu SQL và việc truy cập các CSDL này trở lên đơn giản hơn nhờ một cách thức giống nhau đối với nhiều CSDL.Với những lợi ích và khả năng trên có thể thấy rằng SQL là một trong những hệ quản trị CSDL bạn nên chọn.

# CHƯƠNG III: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

* 1. **Tổng quan ứng dụng.**
     1. **Đối tượng sử dụng ứng dụng**
* Người tiêu dùng
* Người tiêu dùng là đối tượng sử dụng website để thực hiện cho các nhu cầu tìm kiếm thông tin , đặt hàng các dịch vụ theo nhu cầu của bản thân.
* Là người yêu thích những dịch vụ , sản phẩm của các công ty kinh doanh liên quan và có nhu cầu sở hữu nó.
* Công ty

Là đối tượng quản lý tất cả các hoạt động của công ty mình , kể cả đối với các vấn đề về khách hàng.

* + 1. **Các chức năng chính của hệ thống**
* Đối với người tiêu dùng

Hệ thống có các chức năng hỗ trợ và giải quyết nhu cầu của khách hàng:

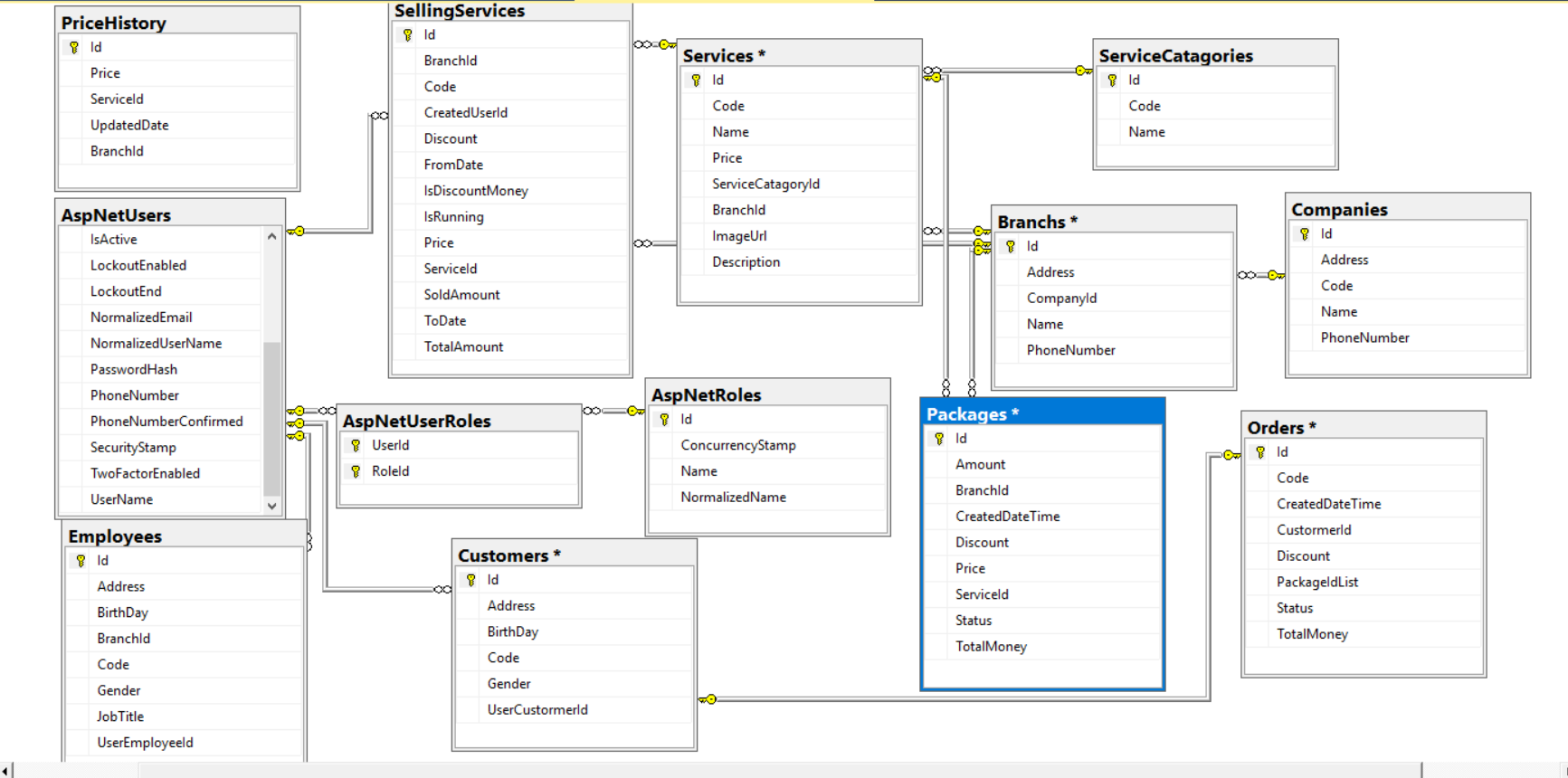
* Xem thông tin các dịch vụ.
* Đăng ký tài khoản của hệ thống để có thể đăng chọn dịch vụ, đặt hàng, hưởng các dịp khuyến mại.
* Thay đổi thông tin cá nhân.
* Đối với Công ty

Hệ thống cần có các chức năng để hỗ trợ việc , đăng sản phẩm, dịch vụ:

* Tạo , chỉnh sửa sản phẩm, dịch vụ cho công ty mình.
* Toàn quyền trong Công ty của mình.
* Quản lý dịch vụ của công ty mình.
* Quản lý các tài khoản của cả nhân viên, khách hàng.
* Quản lý đơn hàng của khách hàng.
* Quản lý các chi nhánh của công ty.
  + 1. **Mô tả các thực thể**
* **Role** (Id, Name)
* **User** (Id, UserName, Password, FirstName, LastName, PhoneNumber, RoleId)
* **Employee** (Id, Code, BranchId, JobTitle, gender, BirthDay, Address, UserId)
* **Customer** (Id, UserId, Code, gender, BirthDay, Address,)
* **ServiceCatagory** (Id, Code, Name)
* **Service** (Id,Code, Name, ServiceCatagoryId , Price ).
* **SellingService** (Id,Code, ServiceId, BranchId , isRunning, TotalAmount, SoldAmount, FromDate, ToDate, Price, isDiscountMoney, Discount )
* **Company** (Id, code, Name, Address, Phone)
* **Branch** (Id, CompanyId, Name, Addess , Phone)
* **Package**(Id,ServiceId, BranchId, Amount, Price, Discount, TotalMoney, CreatedDateTime, Status)
* **Order** (Id, Code, ServiceList { ServiceId, Code, Name, Amount, Price, SumPrice }, Discount, TotalMoney, EmployeeId, CustormerId, Status ) . trong đó Status có 3 trạng thái: Booked, Paid, Canceled) -đã đặt, đã thanh toán, bị hủy .
* **PriceHistory** (Id, ServiceId, Price, UpdatedDate )
  + 1. **Mối quan hệ giữa các thực thể**
* Giữa bảng Servicecategories và bảng Services có mối quan hệ một – nhiều. Một Servicecategories có nhiều Services nhưng một Services chỉ thuộc một Servicecategories .
* Giữa bảng users và bảng Roles có mối quan hệ nhiều– nhiều. Một users có nhiều Roles và một Roles thuộc nhiều users.
* Giữa bảng users và bảng Customers có mối quan hệ một – nhiều. Một users có nhiều Customers nhưng một Customers chỉ thuộc một users.
* Giữa bảng users và bảng Employees có mối quan hệ một – nhiều. Một users có nhiều Employees nhưng một Employees chỉ thuộc một users.
* Giữa bảng Comanies và bảng Branchs có mối quan hệ một – nhiều. Một Comanies có nhiều Branchs nhưng một Branchs chỉ thuộc một Comanies .
* Giữa bảng Services và bảng Branchs có mối quan hệ nhiều– nhiều. Một Branchs có nhiều Services và một Services thuộc nhiều chi nhánh của cùng một công ty.
* Giữa bảng PriceHistory và bảng Services có mối quan hệ nhiều– nhiều. Một PriceHistory có nhiều Services và một Services có thuộc nhiều PriceHistory .
* Giữa bảng PriceHistory và bảng Branchs có mối quan hệ nhiều– nhiều. Một PriceHistory có nhiều Branchs và một Branchs có thuộc nhiều PriceHistory .
* Giữa bảng SellingService và bảng Branchs có mối quan hệ nhiều – nhiều. Một SellingService có nhiều Branchs và một Branchs thuộc nhiều SellingService.
* Giữa bảng SellingService và bảng Services có mối quan hệ nhiều – nhiều. Một SellingService có nhiều Services và một Services thuộc nhiều SellingService.
* Giữa bảng Packages và bảng Services có mối quan hệ nhiều – nhiều. Một Packages có nhiều Services và một Services thuộc nhiều Packages.
* Giữa bảng Packages và bảng Orders có mối quan hệ một– nhiều. Một Packages chỉ thuộc một Orders nhưng một Orders có nhiều Packages.
* Giữa bảng Orders và bảng Users có mối quan hệ nhiều – một. Một Users có nhiều Orders nhưng một Orders chỉ thuộc một Users.
  1. **Thiết kế cơ sở dữ liệu.**
     1. **Tổng quan cơ sở dữ liệu**

Từ những phân tích tổng quát của hệ thống, đưa đến việc phát triển cơ sở dữ liệu nhằm lưu trữ và truy xuất dữ liệu.

Hệ quản trị cơ sở dữ liệu sử dụng: SQL Server



Hình 3.1. Tổng quan cơ sở dữ liệu

* + 1. **Chi tiết cơ sở dữ liệu**

Danh sách các thực thể có trong cơ sở dữ liệu

* Users: Thông tin Công ty, khách hàng.
* ServiceCategories: Thông tin thể loại dịch vụ.
* Services: Thông tin chi tiết về dịch vụ.
* PriceHistory: Thông tin lịch sử giá của dịch vụ.
* SellingService: Thông tin các dịch vụ đang được cung ứng.
* Companies: Thông tin các công ty trong hệ thống.
* Branchs: Thông tin chi nhánh của các công ty tham gia trong hệ thống .
* Roles: Thông tin về Quyền sử dụng trong hệ thống.
* Users: Thông tn tài khoản truy cập hệ thống.
* Customer: Thông tin khách hàng.
* Employees: Thông tin nhân viên trong công ty.
* Packages: Thông tin các gói dịch vụ khi khách hàng đặt hàng.
* Orders: Thông tin các đơn đặt hàng cuả khách hàng.

## 3.3 Thiết kế giao diện

* + 1. **Thiết kế giao diện hệ thống công ty**

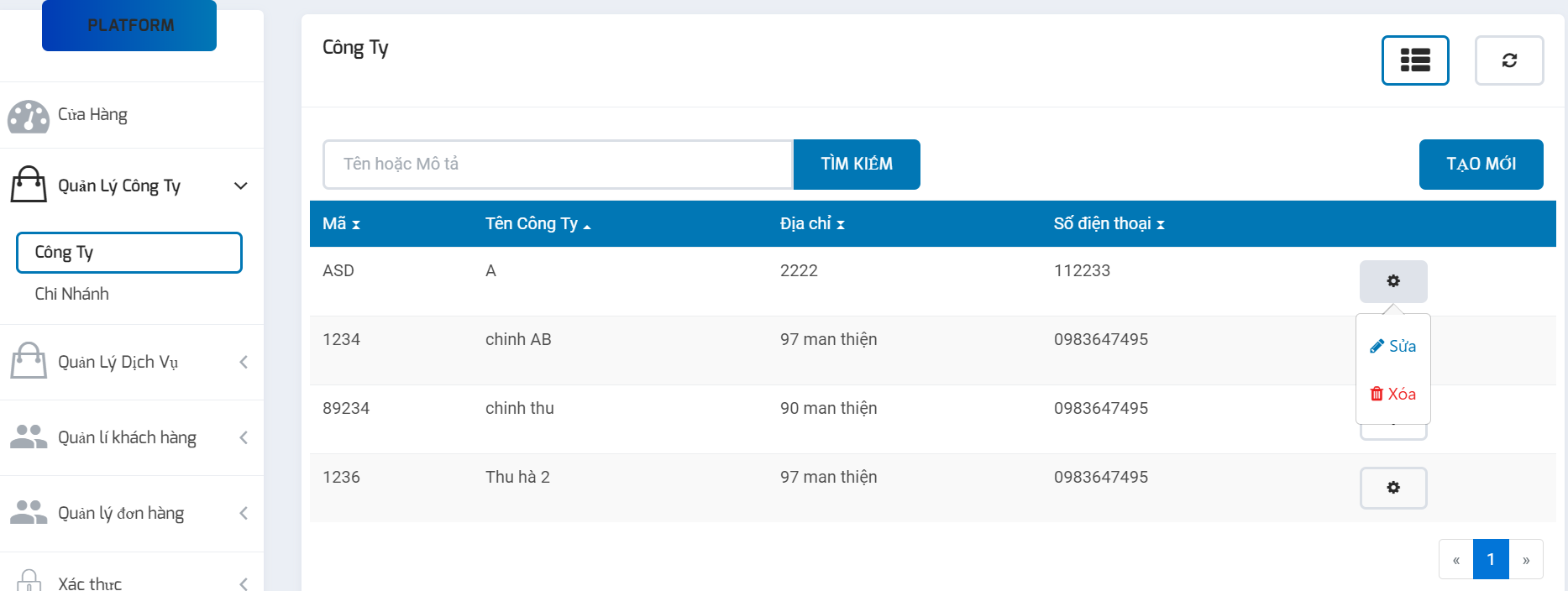
Giao diện hệ thống này giành cho người quản trị viên – chính là admin của công ty, hoặc Admin của cả hệ thống được thao tác trên giao diện hiển thị này.

Gồm có các chức năng chính là:



Hình 3.2. Trang quản lý hệ thống công ty

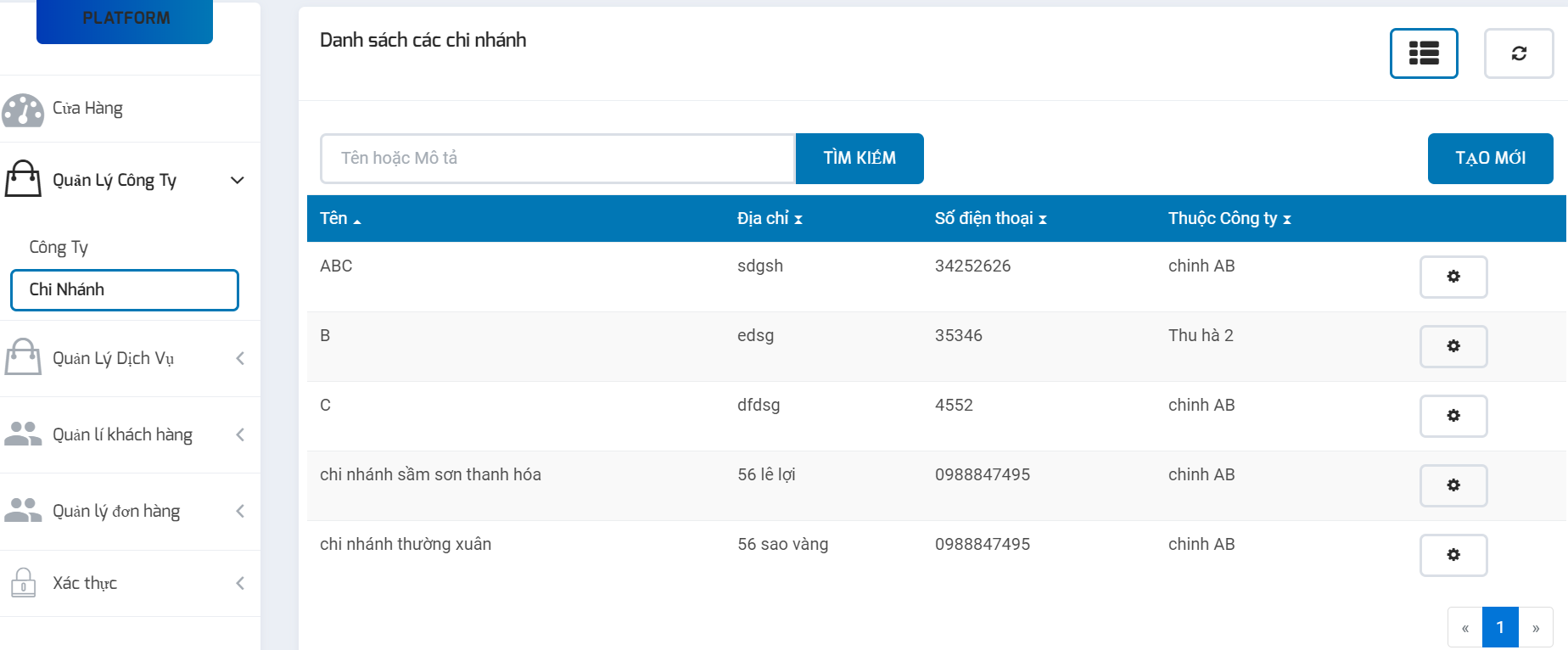
* Quản lý công ty, chi nhánh.
* Quản lý dịch vụ.
* Quản lý khách hàng.
* Quản lý đơn hàng của khách hàng.
* Quản lý thể loại dịch vụ, .
* Công ty:



Hình 3.3. Trang quản lý công ty

Thiết kế này đảm bảo cho việc thao tác với Công ty. Hỗ trợ quản lý trong giao diện này là các chức năng: Xem danh sách công ty, thêm, xóa, sửa và cập nhật thông tin các công ty.

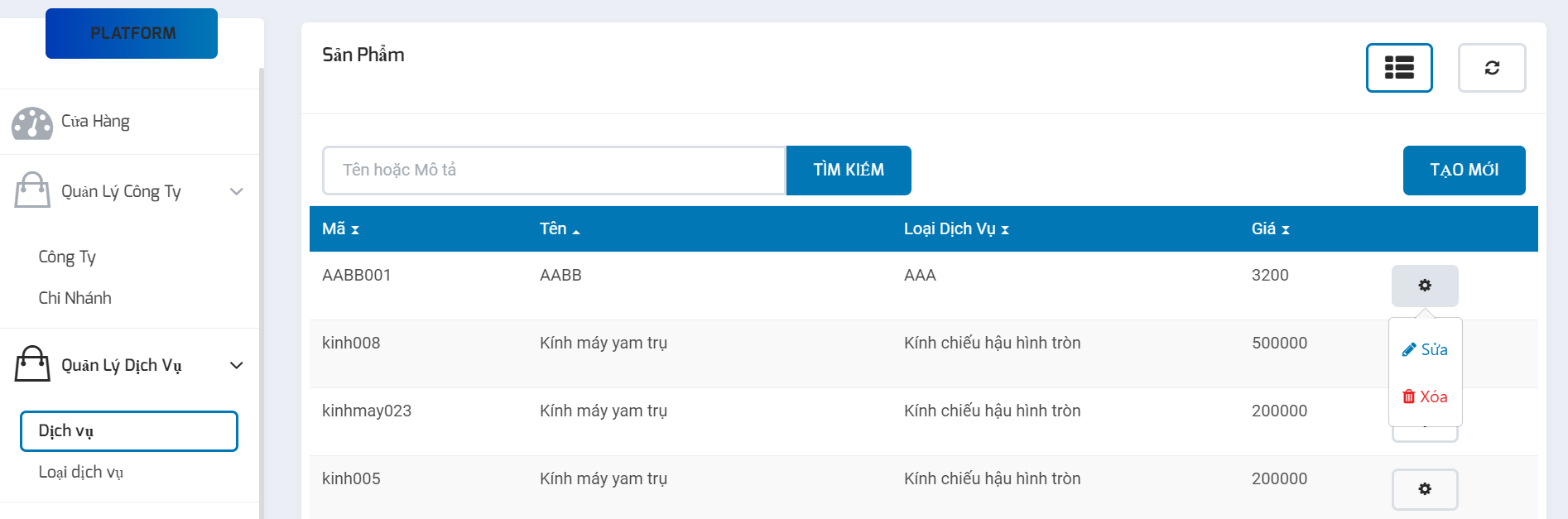
* *Chi nhánh* : Xem danh sách các chi nhánh của công ty, thêm, xóa, sửa các chi nhánh.



*Hình 3.4. Trang quản lý Chi nhánh*

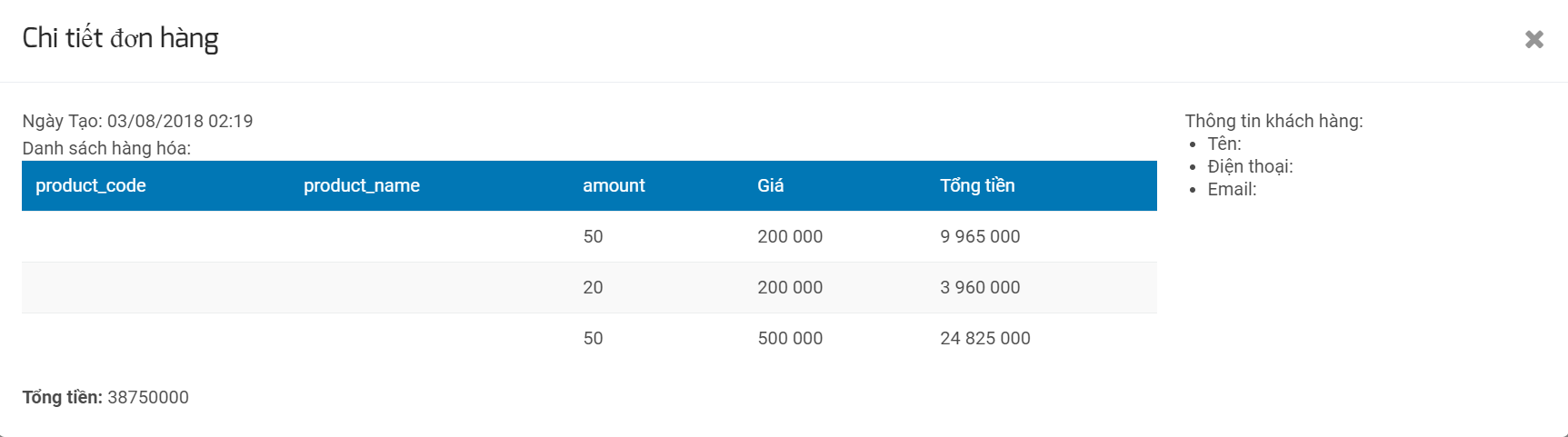
Hỗ trợ quản lý trong giao diện này là xem danh sách chi nhánh, xóa, sửa chi nhánh trong danh sách. Có thể thêm chi nhánh mới.

* *Dịch vụ*: Xem danh sách các dịch vụ của công ty, thêm, xóa, sửa các dịch vụ.



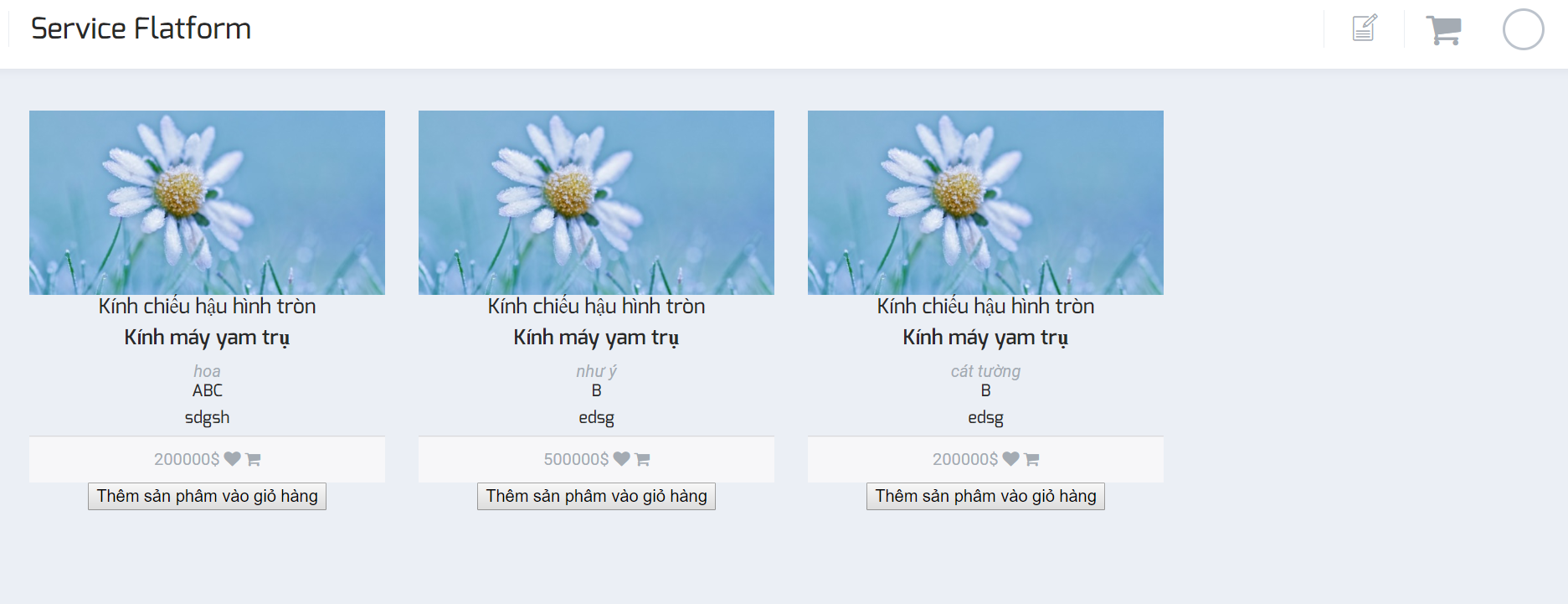
Hình 3.5. Trang quản lý dịch vụ

* Khách hàng: Xem danh sách khách hàng.
* Đơn hàng: Xem danh sách đơn đặt hàng của khách hàng.



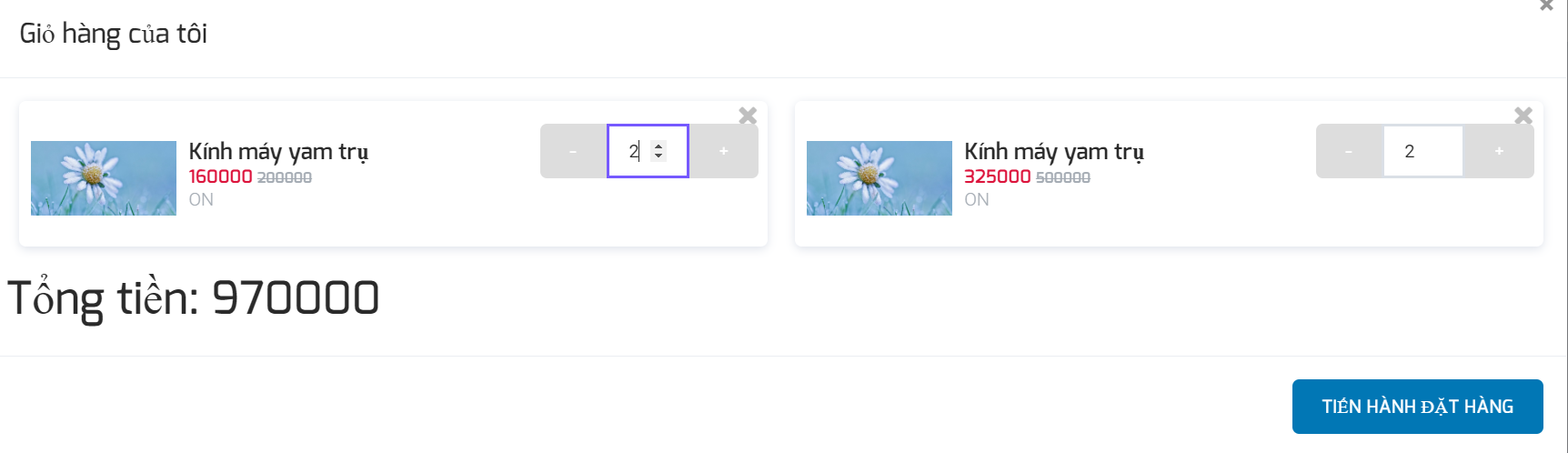
Hình 3.6. Trang xem chi tiết đơn đặt hàng

* + 1. **Thiết kế giao diện người tiêu dùng**
* Trang chủ: Hiện thị toàn bộ các dịch vụ của các công ty cho người dùng được xem không cần đăng nhập.



Hình 3.7. Trang chủ

* Hiển thị list dịch vụ theo từng công ty.
* Hiển thị chi tiết dịch vụ: Người dùng được phép đặt hàng khi đã đăng nhập vào hệ thống



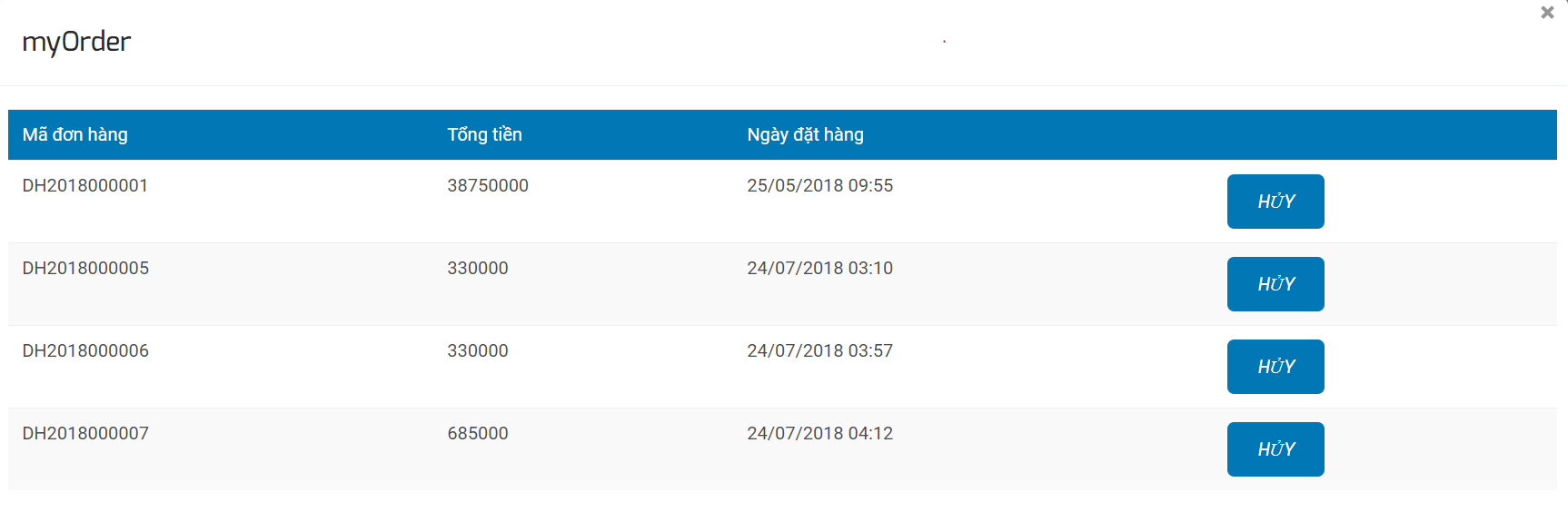
Hình 3.8.Trang hiển thị chi tiết giỏ hàng của bạn

* Hiển thị tiến hành đặt hàng: Trang chứa nội dung của User để bạn xác nhận các thông tin đã đúng để bên công ty cung ứng dịch vụ đó sẽ liên hệ với khách hàng.



Hình 3.9. Trang xác nhận thông tin để đặt hàng

- HIển thị những đơn đặt hàng của khách hàng: xem danh sách đơn đặt hàng, có thể hủy nếu còn trong thời gian cho phép.



Hình 3.10. Trang hiển danh sách đơn đặt hàng

# CHƯƠNG IV: XÂY DỰNG WEBSITE

* 1. **Giới thiệu các công nghệ sử dụng**
     1. **Các công nghệ sử dụng trong việc xây dựng website**
* Môi trường: NodeJs, ASP.NET MVC C#
* Cơ sở dữ liệu: SQL Server
* Server: Express JS
* Client: Angular 4
  + 1. Các bước thực hiện xây dựng website

*Bước 1*: Phân tích, thiết kế, mô tả các chức năng cần thiết của website.

*Bước 2*: Phân tích, thiết kế cơ sở dữ liệu để phù hợp với chức năng và thực hiện hiệu quả việc truy vấn dữ liệu.

*Bước 3*: Viết API cần thiết khi thiết kế xong cơ sở dữ liệu.

*Buốc 4*: Thiết kế MockUp giao diện người dùng sơ bộ theo mô tả chức năng cần có của website.

*Bươc 5*: Thiết kế hoàn chỉnh bằng phần mềm hỗ trợ đồ họa để có bản thiết kế hoàn chỉnh cho website

*Bước 6*: Sử dụng AngularJs, HTML, CSS, Bootstrap…để tiến hành việc lập trình theo thiết kế hoàn chỉnh.

*Bước 7*: Chạy website

# CHƯƠNG V: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

* 1. **Kết quả đạt được.**

Website tin tức cung cấp thông tin đến đến bạn đọc, giới thiệu về những vấn đề liên quan đến nhu cầu thị trường mà người tiêu dùng quan tâm.

Giới thiệu được những sản phẩm của công ty đến người tiêu dùng, giúp họ tìm kiếm nhanh hơn những sản phẩm mà có nhu cầu mua, họ thể tham gia mua sản phẩm đó.

Trong quá trình thực tập, bản thân em đã học hỏi và tìm hiểu được rất nhiều các công nghệ mới .

* 1. **Hạn chế.**

Tuy nhiên, do thời gian ngắn và khả năng lập trình của em không tốt lắm cộng với số lượng công việc cũng như tìm hiểu các công nghệ mới cho nên đề tài của em chưa được hoàn thiện về mặt thực hành. Các chức năng chưa được đầy đủ như dự kiến. Mong nhận được nhiều ý kiến từ thầy cô và giúp em hoàn thiện sản phẩm hơn.

* 1. **Hướng phát triển**

Về mặt đối tượng sử dụng, có thể mở rộng hơn cho nhiều đối tượng công ty và khách hàng khác nhau nữa, kết hợp việc tìm hiểu nghiệp vụ trao đổi thông tin dịch vụ thông qua công ty.

Về mặt kỹ thuật, có thể tối ưu code hơn và xây dựng web đấu giá có thanh toán trực tuyến.

# TÀI LIỆU THAM KHẢO

**Tiếng Anh:**

1. AngularJs – Bard Green and Shyam
2. HTML & CSS by Jon Duckett
3. JavaScript: The Good Parts by Douglas Crockford

**Danh mục các Website tham khảo:**

1. <http://www.w3schools.com/w3css/default.asp>
2. <http://www.nodejs.org/en/docs>
3. <http://www.tutorialspoint.com/nodejs/>
4. <http://getbootstrap.com>
5. <http://www.tutorialspoint.com/mysql/>
6. <https://docs.angularjs.org>