

**BỘ THÔNG TIN VÀ TRUYỀN THÔNG
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**



BÁO CÁO

THỰC TẬP TỐT NGHIỆP

Đề tài :

**XÂY DỰNG WEBSITE THIẾT KẾ VÀ ĐẶT MUA CÂY
CẢNH CHO CỬA HÀNG BÁN CÂY CẢNH**

Người hướng dẫn : TS.GVC. VÕ XUÂN THỂ
Sinh viên thực hiện : Nguyễn Thành Định
Mã số sinh viên : N14DCCN190
Lớp : D14CQCPN01

TP.HCM tháng 08/ 2018

LỜI CẢM ƠN

Trên thực tế không có sự thành công nào mà không gắn liền với những sự hỗ trợ, giúp đỡ dù ít hay nhiều, dù trực tiếp hay gián tiếp của người khác. Trong suốt thời gian từ khi bắt đầu học tập tại trường đến nay, em đã nhận được rất nhiều sự quan tâm, giúp đỡ của quý thầy cô, gia đình và bạn bè. Với lòng biết ơn sâu sắc nhất, em xin gửi đến quý thầy cô ở khoa Công Nghệ Thông Tin 2 – Học viện Công Nghệ Bưu Chính Viễn Thông cơ sở tại TP.HCM đã tận tâm chỉ bảo, truyền đạt vốn kiến thức quý báu cho chúng em trong suốt thời gian học tập tại trường. Và đặc biệt trong học kỳ này, nếu không có những lời hướng dẫn, dạy bảo của các thầy cô thì em nghĩ bài báo cáo này của em rất khó có thể hoàn thiện được. Một lần nữa, em xin chân thành cảm ơn thầy cô. Bước đầu đi vào thực tế của em còn hạn chế và còn nhiều ngỡ ngàng. Do vậy, không tránh khỏi những thiếu sót là điều chắc chắn, em rất mong nhận được những ý kiến đóng góp quý báu của quý Thầy Cô và các bạn học cùng lớp để kiến thức của em trong lĩnh vực này được hoàn thiện hơn.

Em xin gửi lời cảm ơn chân thành và sự tri ân sâu sắc đối với các thầy cô ở khoa Công Nghệ Thông Tin 2 – Học viện Công Nghệ Bưu Chính Viễn Thông cơ sở tại TP.HCM đã tạo điều kiện cho em để em có thể hoàn thành tốt bài báo cáo thực tập này. Đặc biệt em xin gửi lời cảm ơn đến thầy TS.GVC. Võ Xuân Thử với những buổi vừa học kết hợp với thực hành trên lớp cùng những buổi nói chuyện rất hữu ích đã giúp em định hướng và hoàn thành tốt đề tài này.

Trong quá trình làm bài báo cáo, cũng như là trong quá trình làm đề tài khó tránh khỏi sai sót, rất mong các Thầy, Cô bỏ qua. Đồng thời do trình độ lý luận cũng như kinh nghiệm thực tiễn còn hạn chế nên bài báo cáo không thể tránh khỏi những thiếu sót, em rất mong nhận được ý kiến đóng góp Thầy, Cô để em học thêm được nhiều kinh nghiệm làm hành trang vững chắc để em tự tin theo đuổi sự nghiệp của mình.

Em xin chân thành cảm ơn!

Thành phố Hồ Chí Minh, 08/2018

Nguyễn Thành Định

MỤC LỤC

MỞ ĐẦU.....	1
CHƯƠNG I: GIỚI THIỆU ĐỀ TÀI	2
1.1. Giới thiệu đề tài.....	2
1.2. Mục đích đề tài	2
1.3. Giới thiệu các chức năng chính của sản phẩm đề tài	2
1.4. Các yếu tố kỹ thuật có sử dụng trong đề tài	2
1.5. Lĩnh Vực.....	3
1.6. Bố cục báo cáo của đề tài	3
CHƯƠNG II: CƠ SỞ LÝ THUYẾT CỦA ĐỀ TÀI.....	4
2.1. Giới thiệu về Bootstrap 4.....	4
2.2. Các tính năng mới trong HTML5.....	5
2.3. Các tính năng mới trong CSS3	6
2.4. Giới thiệu về Framework Angular.....	6
2.4.1. Sơ lược về Angular.....	6
2.4.2. Angular 4 :	8
2.5. Giới thiệu ASP.NET CORE 1.0	9
2.5.1. ASP.NET là gì?.....	9
2.5.2. ASP.NET Core là gì ?.....	10
CHƯƠNG III: PHÂN TÍCH ,THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG.....	13
3.1. Yêu cầu cơ bản của sản phẩm đề tài.....	13
3.1.1. Đối tượng sử dụng phần mềm:	13
3.1.2. Giới hạn nội dung của đề tài:.....	13
3.2. Cơ sở dữ liệu:.....	14
3.3. Phân tích các chức năng xử lý :	17
3.4. Phân tích thiết kế giao diện chính cho đề tài :	18
CHƯƠNG IV: GIỚI THIỆU SẢN PHẨM CỦA ĐỀ TÀI	21
4.1. Giới thiệu sản phẩm và chức năng:	21
4.1.1. Giới thiệu sản phẩm:.....	21
4.1.2. Yêu cầu kỹ thuật:	21
4.2. Giới thiệu màn hình và hệ thống các chức năng:	21
4.2.1 Hệ thống cho khách hàng:	21

4.2.1.1	Màn hình đăng nhập:	21
4.2.1.2	Màn hình đăng kí	22
4.2.1.3	Màn hình trang chủ	23
4.2.1.4	Màn hình thêm vào giỏ hàng	24
4.2.1.5	Màn hình đặt hàng	25
4.2.1.6	Màn hình hủy đơn hàng	25
4.2.2	Hệ thống cho Quản lý(Admin):	26
4.2.2.1	Màn hình trang chủ	26
4.2.2.2	Màn hình quản lý sản phẩm.....	27
4.3.	Giới thiệu các mã lệnh cơ bản dùng trong đề tài:.....	28
4.3.1	Mã lệnh gọi API đăng nhập.....	28
4.3.2	Mã lệnh gọi API lấy các đối tượng trong danh sách	29
4.3.3	Mã lệnh gọi API xóa một đối tượng trong danh sách	29
4.3.4	Mã lệnh thay đổi trạng thái của một đối tượng	29
CHƯƠNG V: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....		31
5.1.	Kết Luận:	31
5.1.1	Những kết quả đạt được.....	31
5.1.2.	Hạn chế	31
5.1.3	Hướng khắc phục.....	31
5.2	Hướng phát triển:.....	32
TÀI LIỆU THAM KHẢO.....		33

DANH MỤC HÌNH

Hình 2.1. Cách tổ chức cấu trúc thư mục cho Component.....	8
Hình 2.2. Mô hình xây dựng Angular 4 cơ bản.....	9
Hình 2.3. Mô hình xây dựng .Net Core MVC.....	12
Hình 3.1. Lược đồ cơ sở dữ liệu Diagram.....	14
Hình 3.2. Lược đồ BFD.....	18
Hình 4.1. Màn hình đăng nhập	22
Hình 4.2. Màn hình đăng ký	23
Hình 4.3. Màn hình trang chủ.....	24
Hình 4.4. Màn hình giỏ hàng	24
Hình 4.5. Màn hình đặt hàng	25
Hình 4.6. Màn hình hủy đơn hàng.....	26
Hình 4.7. Màn hình trang chủ quản lý.....	27
Hình 4.8. Màn hình quản lý sản phẩm.....	27

DANH MỤC CÁC KÝ HIỆU VÀ CHỮ VIẾT TẮT

TT:		Thông tin
AJAX	Asyschoronous JavaScript And XML	
ASP	Active Server Pages	
BFD	Bussiness Function Diagram	Mô hình chức năng nghiệp vụ
CSS	Cascading Style Sheets	
HTTP	HyperText Transfer Protocol	
SQL	Structure Query Language	

MỞ ĐẦU

Trong những năm gần đây, khi kinh tế Việt Nam đang hội nhập vào nền kinh tế thế giới với một sự phát triển tương đối ổn định. Thu nhập trung bình của người dân Việt Nam đang dần được cải thiện. Với những điều kiện thuận lợi như vậy thì nhu cầu trong tiêu dùng của người dân đang ngày càng tăng, những đòi hỏi về hình thức phục vụ, sự tiện lợi trong mua bán cũng như thanh toán cũng được người dân Việt Nam chú ý đến.

Đi cùng xu thế đó, bán hàng qua mạng là một giải pháp hoàn toàn phù hợp với những đòi hỏi mới của người tiêu dùng. Với những thế mạnh của mình, bán hàng qua mạng đã trở thành một hình thức bán hàng phổ biến trên thế giới. Trong khoảng hai đến ba năm trở lại đây, bán hàng qua mạng ở Việt Nam đã có những sự phát triển vượt bậc, thể hiện ở chất lượng bán hàng được cải thiện, sự quan tâm đến hình thức mua bán này của người tiêu dùng ngày càng một lớn hơn.

Đề tài “Xây dựng website bán cây cảnh” hỗ trợ thuận tiện nhất cho người dùng mua bán cây cảnh trực tiếp trên website để giúp người dùng có những sản phẩm tốt nhất.

CHƯƠNG I: GIỚI THIỆU ĐỀ TÀI

1.1. Giới thiệu đề tài

- Website bán cây cảnh là một trang web dùng để giúp cho những nhà kinh doanh về cây cảnh có thể kinh doanh sản phẩm của mình trên mạng(Internet), giúp người bán và người mua có thể thực hiện những giao dịch trực tiếp trên website.

- Website cây cảnh không chỉ giúp người mua thực hiện mua bán mà còn giúp cho người chủ thực hiện được quản lý cửa hàng của mình một cách tốt hơn.

- Website cây cảnh cung cấp những thông tin hữu ích cho khách hàng về những loại cây, cách chăm sóc cây cảnh, trang trí nội thất của mình một cách đẹp nhất và thoải mái nhất

1.2. Mục đích đề tài

- Xây dựng website thiết kế và đặt mua cây cảnh giúp cho khách hàng với những người ưu thích cây cảnh sẽ dễ dàng chọn cây và mua cây.

- Giúp khách hàng tiết kiệm được thời gian, thuận tiện trong việc mua sản phẩm online mà không trực tiếp đến cửa hàng

1.3. Giới thiệu các chức năng chính của sản phẩm đề tài

- Xây dựng Website thiết kế và đặt mua cây cảnh
 - o Hỗ trợ người quản trị (admin)
 - Quản trị các User (Khách hàng)
 - Quản lý các sản phẩm và danh mục sản phẩm
 - Quản lý đơn hàng
 - Quản lý các đơn hàng
 - o Dịch vụ đặt mua hàng(Khách hàng)
 - Xem sản phẩm ưu thích
 - Đặt mua hàng
 - Hủy đơn hàng mình đã đặt mua

1.4. Các yếu tố kỹ thuật có sử dụng trong đề tài

- Bootstrap 4 & HTML5
- Angular 4
- ASP.Net Core

- SQL Server 2014

1.5. Lĩnh Vực

- Chuyên ngành: Công Nghệ thông tin
- Lập trình website thiết kế và đặt mua cây cảnh. Sử dụng Angular 4 và Web API viết bằng .NET Core kết nối dữ liệu từ hệ cơ sở dữ liệu SQL Server.
- Lĩnh vực liên quan: Bán hàng.

1.6. Bộ cục báo cáo của đề tài

- Chương 1: Giới thiệu đề tài, những mục đích, lĩnh vực mà đề tài hướng tới
- Chương 2: Cơ sở lý thuyết về Bootstrap 4, HTML5 , CSS3, Angular và ASP. NET Core trong đề tài.
- Chương 3: Phân tích những yêu cầu , các chức năng xử lý và dữ liệu của sản phẩm đề tài, thiết kế giao diện của sản phẩm đề tài.
- Chương 4: Giới thiệu sản phẩm và hệ thống các chức năng cơ bản đã làm được của sản phẩm đề tài
- Chương 5: Kết quả đạt được, những hạn chế, hướng khắc phục và hướng phát triển của đề tài

CHƯƠNG II: CƠ SỞ LÝ THUYẾT CỦA ĐỀ TÀI

2.1. Giới thiệu về Bootstrap 4.

Khởi nguồn Bootstrap được tạo ra bởi một nhà thiết kế và một nhà phát triển làm việc tại Twitter. Kể từ đó đến nay, nó đã trở thành một trong số những framework front-end và dự án mã nguồn mở phổ biến nhất trên thế giới.

Bootstrap được tạo ra tại Twitter vào giữa năm 2010 bởi @mdo và @fat. Trước khi chính thức trở thành một framework mã nguồn mở, Bootstrap được biến đổi như là *bản thiết kế của Twitter*. Sau một vài tháng đi vào phát triển, Twitter đã tổ chức Tuần lễ Hack đầu tiên và hàng loạt các nhà phát triển ở mọi trình độ đã tham gia vào dự án. Và trong hơn một năm sau đó, Bootstrap giống như một bản hướng dẫn style dành riêng cho việc phát triển các công cụ nội bộ phục vụ cho công ty trước khi nó được phát hành chính thức, và nó vẫn đóng một vai trò như vậy cho đến ngày nay.

Kể từ lần đầu tiên được phát hành vào Thứ 6, ngày 19 tháng 8, năm 2011 tới nay, chúng tôi đã có tới hơn 20 lần phát hành, bao gồm cả 2 bản viết lại chính của phiên bản 2 và phiên bản 3. Với Bootstrap 2, chúng tôi đã thêm tính năng responsive như một stylesheet tùy ý. Nhưng đến khi xây dựng Bootstrap 4, chúng tôi đã viết lại thư viện một lần nữa để tích hợp tính năng responsive một cách mặc định.

Bootstrap là một Font-end Framework được viết bằng SASS và biên dịch thành CSS, nó là một bộ thư viện mạnh mẽ được tích hợp nhiều CSS có sẵn giúp cho việc lập trình HTML & CSS trở nên đơn giản và dễ dàng hơn. Thông thường khi chúng ta lập trình bằng CSS thuần thì công việc kiểm tra tính tương thích trên các trình duyệt khác nhau và trên các thiết bị di động khác nhau rất là khó khăn, nhưng với Bootstrap thì lập trình viên không cần phải tốn nhiều công sức nữa vì mọi thứ đã có người tạo nên Bootstrap lo.

Bootstrap giúp các nhà thiết kế website có thể giảm bớt được thời gian trong việc thiết kế website. Khi sử dụng Bootstrap, bạn có thể hạn chế được thời gian bỏ ra để viết code cho giao diện vì trong thư viện của nó đã có sẵn những đoạn mã có thể chèn vào trong website.

Bootstrap chia layout của một trang web ra thành 960 Grid và gồm 12 cột, mỗi cột 80Grid và đây chính là chuẩn thiết kế HTML & CSS của Bootstrap. Nó có hỗ trợ hầu hết các module của một trang web như menu, tabs, tooltip, popup, ... Ngoài ra nó còn sử dụng thêm Javascript để xử lý các hiệu ứng cấp cao, code javascript của nó sử dụng jquery nên để sử dụng được bắt buộc bạn phải bổ sung thêm thư viện jQuery nữa.

Ngày nay, khi mà xu hướng phát triển website tương thích với mọi thiết bị thì Bootstrap nổi lên như 1 Framework cung cấp sẵn responsive css phù hợp với mọi thiết bị như SmartPhone, tablets, dextop.....

2.2. Các tính năng mới trong HTML5

HTML5 cung cấp các công cụ quản lý dữ liệu, đồ họa, video, và âm thanh có hiệu quả. Nó tạo điều kiện cho sự phát triển của các ứng dụng giữa các trình duyệt với nhau cho trang web cũng như cho các thiết bị di động. HTML5 là một trong những công nghệ thúc đẩy những cải tiến trong các dịch vụ điện toán đám mây di động, vì nó tính đến tính linh hoạt rộng hơn, cho phép phát triển các trang web thú vị và có khả năng tương tác. Nó cũng đưa vào thẻ và các cải tiến mới, bao gồm cấu trúc thu nhỏ, các nút điều khiển của biểu mẫu, các API, đa phương tiện, hỗ trợ cơ sở dữ liệu, và tốc độ xử lý nhanh hơn.

HTML5 cung cấp:

- Các thẻ mô tả chính xác những gì chúng được thiết kế để chứa đựng.
- Tăng cường khả năng truyền thông trên mạng.
- Cải thiện khả năng lưu trữ chung.
- Các trình làm việc trên nền Web (Web Workers) để chạy các quá trình nền.
- Giao diện WebSocket để tạo kết nối liên tục giữa các ứng dụng cư trú và máy chủ.
- Truy vấn dữ liệu đã được lưu trữ tốt hơn.
- Cải thiện tốc độ nạp và lưu trang.
- Hỗ trợ cho CSS3 để quản lý giao diện người dùng đồ họa (GUI), có nghĩa là HTML5 có thể được định hướng nội dung.
- Cải thiện xử lý biểu mẫu trình duyệt.

- Một API cơ sở dữ liệu dựa trên-SQL cho phép lưu trữ cục bộ, phía máy khách..
- Canvas và video, để thêm đồ họa và video mà không cần cài đặt thêm các plug-in
- Đặc tả Geolocation API (API định vị toàn cầu), sử dụng khả năng định vị của máy điện thoại thông minh để kết hợp các dịch vụ và các ứng dụng đám mây di động.
- Các biểu mẫu cải tiến làm giảm nhu cầu tải về mã JavaScript, cho phép truyền thông hiệu quả hơn giữa các thiết bị di động và các máy chủ điện toán đám mây.

2.3. Các tính năng mới trong CSS3

Phần này giới thiệu cho bạn các tính năng mới có thể tìm thấy trong các đặc tả CSS 3 bao gồm: các bộ chọn CSS mới ví dụ như các lớp có cấu trúc, dựa vào trạng thái, và giả phủ định; cũng như các kiểu bộ chọn mới khác.

Nó cung cấp các hiệu ứng mà trước đây phải dùng đến các chương trình khác hỗ trợ: các hình bóng rơi trên văn bản và các hộp, các góc được làm tròn trên các đường viền, và sử dụng tính mờ đục để tạo ra một sự xuất hiện mờ. Nhiều tính năng trong số đó (chẳng hạn như độ mờ đục và các góc được làm tròn) là tương đối phổ biến trong sử dụng do thực tế là chúng làm giảm nhiều khả năng tùy chỉnh trong các trình duyệt Web di sản.

Tiếp theo, bạn sẽ tìm hiểu về các cách bố trí nhiều cột mới mà bạn có thể tạo ra chúng khi sử dụng CSS3. Các cách bố trí này là một sự tạo mẫu cho bố cục tờ báo ở nơi văn bản sẽ trải dài trên một số các cột đã đặt hoặc theo chiều rộng cột cụ thể khi cần. Một tính năng khác sẽ được thảo luận là vấn đề bao gồm các font chữ Web phi tiêu chuẩn nhờ sử dụng thẻ @font-face.

2.4. Giới thiệu về Framework Angular

2.4.1. Sơ lược về Angular

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. 2.5.2 Lịch sử phát triển của ngôn ngữ PHP

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 phân 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.

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 sẽ tìm hiểu chi tiết hơn trong những bài sắp tới.

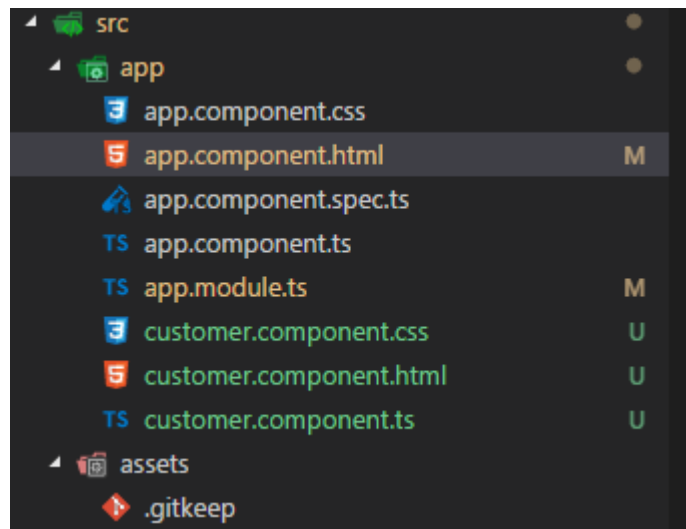
Chúng ta đã biết một component là gì, làm thế nào để ghép chúng lại với nhau để trở thành một ứng dụng? Chúng ta cần Angular modules (mô-dun). **Angular modules** hỗ trợ tổ chức ứng dụng Angular thành một khối gắn kết các chức năng. Mỗi ứng dụng Angular có ít nhất một module, module này gọi là Root Angular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

- **Component trong Angular**

Component là thành phần cơ bản và cốt lõi của angular 2,4, tôi sẽ tóm lược các yếu tố chính của component như sau

- Component có thể là 1 phần của màn hình (còn gọi là View), ví dụ như Danh Sách Khách hàng là 1 component, chi tiết khách hàng lại là 1 component khác...
- Component được tạo lên bởi HTML/CSS/Javascript..... chung quy là tốn khá nhiều code để tạo ra 1 component. Nhưng khi sử dụng thì lại cực kỳ đơn giản vì mỗi một component được thể hiện qua những tag của riêng nó, ví dụ như `<customers></customers>`, `<task></task>`
- Component có thể nhận tham số đầu vào, ví dụ như component chi tiết khách hàng thì tham số đầu vào có thể là Id khách hàng hoặc Json Object chứa các thông tin của khách hàng

- Component có thể chứa lẫn nhau, điều này giúp việc tạo ra các view đơn giản hơn & tăng tính tái sử dụng



Hình 2.1. Cách tổ chức cấu trúc thư mục cho Component

2.4.2. Angular 4 :

- Về Component:

Ý tưởng chủ đạo của Angular phiên bản mới là component. Component là khái niệm không lạ với ReactJS, nhưng component là một cải tiến của Angular so với phiên bản 1. Component được Angular 4 áp dụng để tạo cấu trúc code rõ ràng và có khả năng tái sử dụng lại cái component đã viết.

- Cli tool mạnh mẽ:

Tạo cấu trúc project với một loạt thư viện tích hợp sẵn: build tool với webpack, unit testing với Karma, biên dịch được SCSS thành CSS.

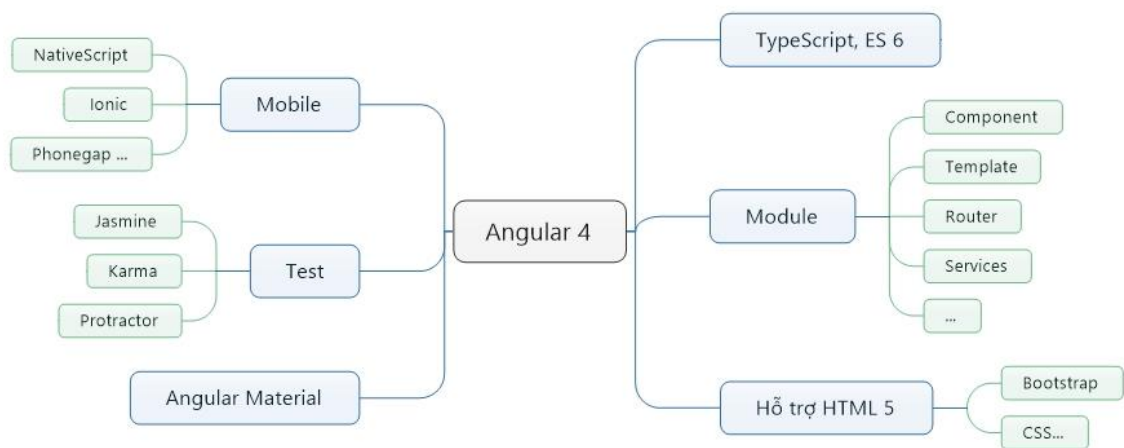
Generator mạnh mẽ, có thể dùng để tạo ra code mẫu cho component, directive, service

...

- Quản lý CSS tốt hơn:

Đặc tính cơ bản của CSS đó là cascade: một thuộc tính của CSS luôn ảnh hưởng đến toàn bộ ứng dụng web. Do đó, để tránh tình trạng mất kiểm soát các thuộc tính CSS ảnh hưởng lung tung lẫn nhau, ta luôn phải tìm cách để quy định file CSS nào sẽ áp dụng cho trang nào. Với Angular 4, ta có thể chỉ định component nào sẽ sử dụng file CSS nào với thuộc tính styleUrls:

```
@Component({
  selector: 'todo-app',
  templateUrl: './app/app.component.html',
  styleUrls: ['./app/app.component.css'],
  directives: [TodosComponent],
  providers: [TodoService]
})
```



Hình 2.2. Mô hình xây dựng Angular 4 cơ bản.

2.5. Giới thiệu ASP.NET CORE 1.0

2.5.1. ASP.NET là gì?

ASP.NET là nền tảng phát triển web (web application frameword), cung cấp một mô hình lập trình, cơ sở hạ tầng phần mềm toàn diện và các dịch vụ cần thiết để xây dựng các ứng dụng web động mạnh mẽ cho máy tính cũng như trên các thiết bị di động.

ASP.NET là một phần của nền tảng Microsoft.NET. Ứng dụng ASP.NET được biên dịch mã, được viết bằng cách sử dụng mở rộng và tái sử dụng các thành phần hoặc đối tượng trong nền tảng NET. Các mã này được sử dụng cho toàn bộ hệ thống phân cấp của các class trong .NET

Các ứng dụng ASP.NET có thể được viết bằng bất kỳ ngôn ngữ nào sau đây:

- C#
- Visual Basic.Net
- Jscript
- J#

ASP.NET được sử dụng để tạo ra các tương tác, dữ liệu điều khiển các ứng dụng web trên internet. ASP.NET bao gồm một số lượng lớn các controls như là các text box, button và labels cho assembling, và các thao tác mã để tạo ra các trang HTML.

2.5.2. ASP.NET Core là gì ?

ASP.NET core là một mã nguồn mở và là nền tảng mới cho xây dựng cloud trên internet kết nối các ứng dụng web, IoT và mobile backends. ASP.NET Core có thể chạy trên .NET Core hoặc chạy đầy đủ trên .NET Framework. ASP.NET Core được kiến trúc để cung cấp một nền tảng phát triển tối ưu cho các ứng dụng được triển khai tới cloud hoặc chạy on-premises. ASP.NET Core bao gồm các thành phần mô-đun cần thiết tối thiểu, do đó bạn giữ lại được tính linh hoạt trong khi xây dựng các solution của bạn. Bạn có thể phát triển và chạy ASP.NET Core trên Windows, MAC và LINUX. ASP.NET Core là mã nguồn mở tại [GitHub](#)

Các phiên bản đầu tiên của ASP.NET đã được ra đời gần 15 năm trước đây như là một phần của .NET Framework. Kể từ đó có hàng triệu nhà phát triển đã sử dụng ASP.NET để xây dựng và chạy các ứng dụng web tuyệt vời, và trong những năm qua, đã được Microsoft liên tục phát triển.

ASP.NET Core có một số thay đổi kiến trúc làm cho nó gọn nhẹ hơn. ASP.NET Core không còn dựa trên System.Web.dll mà dựa trên tập hợp các granular và các NuGet. Nhờ thế cho phép bạn tối ưu hóa các ứng dụng của bạn nhờ chỉ cần sử dụng các gói NuGet mà bạn cần. Ngoài ra nó bảo mật hơn, cải thiện hiệu suất và giảm chi phí nhờ bạn chỉ cần chi trả cho những gì bạn sử dụng mà thôi.

Đặc điểm riêng biệt của ASP.NET Core:

- Hỗ trợ đa nền tảng:

Ngày nay bạn có thể phát triển và chạy ASP.NET trên cả Windows, Mac, và Linux. Và nếu trên Windows bạn có thể sử dụng công cụ tốt nhất Visual Studio 2015 để tạo, quản lý và gỡ lỗi các ứng dụng ASP.NET Core của bạn, thì nay trên bất

kỳ nền tảng nào bạn có thể sử dụng Visual Studio Code. Visual Studio Code là một trình soạn thảo với các plugin có hỗ trợ để chỉnh sửa các ứng dụng ASP.NET Core của bạn.

- ***Nguồn mở:***

Ngày nay mã nguồn và tài liệu ASP.NET đã được Microsoft mở tất cả. Các mã nguồn giờ đã có sẵn trong Github bạn giờ có thể tải hay thay đổi bất kỳ mã nào mà bạn thích. Nếu bạn có bất kỳ một góp ý cải tiến gì đó, bạn có thể gửi một yêu cầu đến cho microsoft để xem xét và kết hợp.

Tương tự như vậy, tất cả các tài liệu cũng là mã nguồn mở và có sẵn trong bài viết docs.asp.net. Mỗi trang trên đó đều có chức năng “chỉnh sửa trang này” ở phía trên và bạn có thể chỉnh sửa các tài liệu từ Microsoft

- ***Entity Framework:***

Entity Framework là một Object Relational Mapper(ORM) một công cụ mapping giữa các object trong ứng dụng với các tables và columns của cơ sở dữ liệu quan hệ.

- Entity Framework là một Open source ORM Framework.
- Một ORM có thể tạo kết nối đến một Relational database, thực thi câu lệnh và trả về kết quả là các **objects** trong ứng dụng.
- Một ORM sẽ theo dõi sự thay đổi dữ liệu, trạng thái của các objects trả về ở trên, mỗi khi có sự thay đổi dữ liệu từ những objects này ORM sẽ gửi những thay đổi đó đến Relational database.

- ***Hỗ trợ xây dựng bằng Dependency Injection***

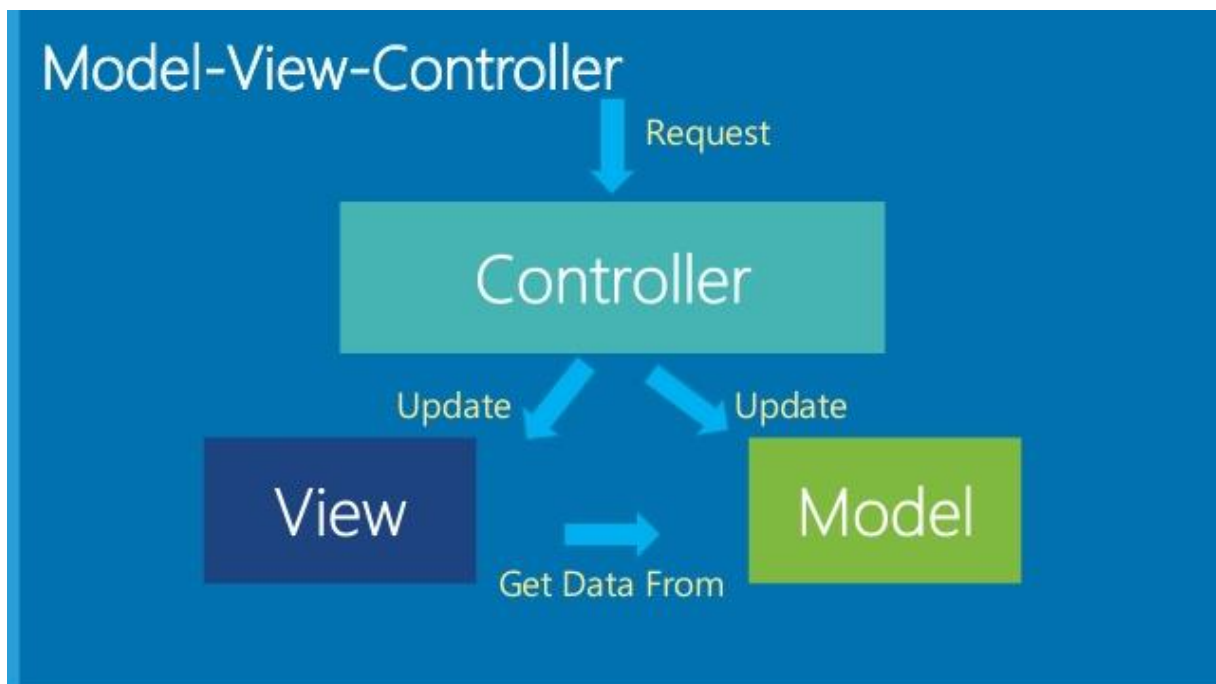
Nếu mà giải thích thì chỉ vài dòng thì sẽ hơi khó hiểu : ASP.NET Core đã được xây dựng trong Dependency Injection. Dependency Injection là một mẫu thiết kế cho phép các phụ thuộc của một class được injected như các đối tượng được yêu cầu trong ứng dụng của bạn. Với ASP.NET Core, Microsoft đã cung cấp một Dependency Injection mà bạn có thể sử dụng để xác định sự phụ thuộc được đưa vào Controller, View của bạn, hoặc bất kỳ lớp học khác mà framework sẽ tạo ra cho bạn. Bạn có thể bắt đầu việc cấu hình thông qua phương thức ConfigureServices trong tập tin Startup.cs.

- ***Một Framework duy nhất***

Trong ASP.NET Core, Microsoft đã đưa tất cả các framework vào một framework duy nhất vừa nhẹ hơn và vừa có những tính năng của MVC và WebAPI. Với việc sáp nhập này của MVC và Web API, mọi thứ đơn giản hơn nhiều khi bạn không cần phải cân nhắc định tuyến khác nhau, an toàn, hoặc các bộ lọc cho một ApiController so với MVC Controller. Tất cả các lớp Controller hiện giờ có thể xử lý các yêu cầu sử dụng API Web hoặc cách tiếp cận MVC.

- ***MVC Helpers Tag:***

Với ASP.NET Core, Microsoft đã giới thiệu tag helpers để tạo ra mã phía client từ .NET và làm cho nó dễ dàng hơn để tái sử dụng trong Razor markup. Nó được tham chiếu trong đánh dấu phía máy chủ của bạn như thể họ là một tag HTML mà bạn được sử dụng. Công cụ Razor sẽ nhận ra thẻ và thực thi các mã .NET có liên quan tương ứng với nó.



Hình 2.3. Mô hình xây dựng .Net Core MVC

CHƯƠNG III: PHÂN TÍCH ,THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG**3.1. Yêu cầu cơ bản của sản phẩm đề tài****3.1.1. Đối tượng sử dụng phần mềm:****❖ Người dùng (User)**

Người dùng là đối tượng sử dụng phần mềm thực hiện các nhu cầu như xem sản phẩm bán , đặt mua hàng và hủy các hóa đơn hàng. Ngoài ra người dùng muốn thực hiện việc mua bán thì có thể đăng kí thành viên.

❖ Người quản lý (Admin)

Người quản lý là người chịu trách nhiệm với toàn bộ hệ thống. Toàn quyền quản lý người dùng , quản lý đơn hàng của khách hàng, việc đăng bài tin tức , tạo các danh mục sản phẩm và sản phẩm bán hàng để phục vụ người dùng.

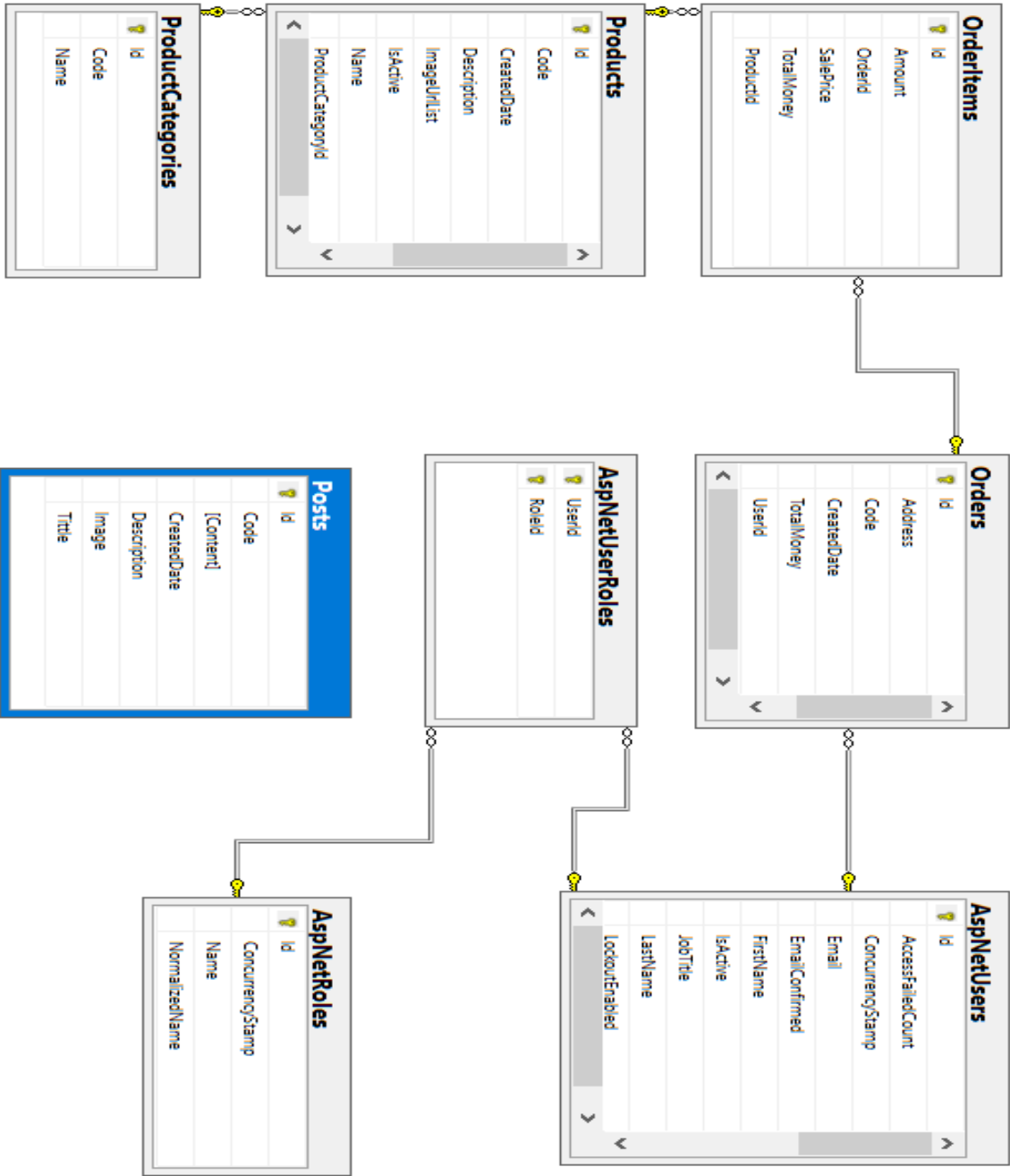
3.1.2. Giới hạn nội dung của đề tài:**❖ Lý Thuyết**

- Hiểu được quy trình bán hàng trực tuyến
- Biết được các quản lý bán hàng online về các mục người dùng, sản phẩm và hóa đơn

❖ Thực hành

- Với admin thực hiện các chức năng quản lý sản phẩm , hóa đơn ,người dùng.
- Với người dùng có thực hiện việc thêm vào giỏ hàng , đặt mua hàng và hủy đơn hàng đã mua.
- Người dùng chỉ có thể xem các sản phẩm được admin tạo ra và chưa có thể xem top các mặt hàng được bán nhiều.
- Việc các file hình ảnh upload lên được xử lý bởi link hình có trực tiếp trên mạng chưa thể upload về local.
- Thanh toán các sản phẩm bằng thẻ tín dụng chưa được xử lý.
- Thời gian hủy đơn hàng chưa được thiết lập.
- Doanh thu mỗi tháng chưa được thực hiện.

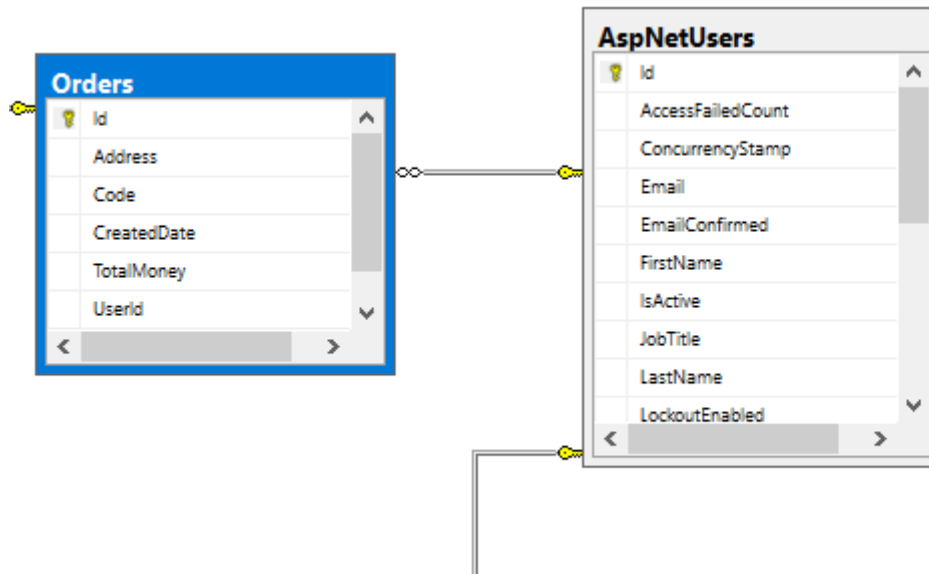
3.2. Cơ sở dữ liệu:



Hình 3.1. Lược đồ cơ sở dữ liệu Diagram

- Phân tích cơ sở dữ liệu:

❖ Về phía Người dùng

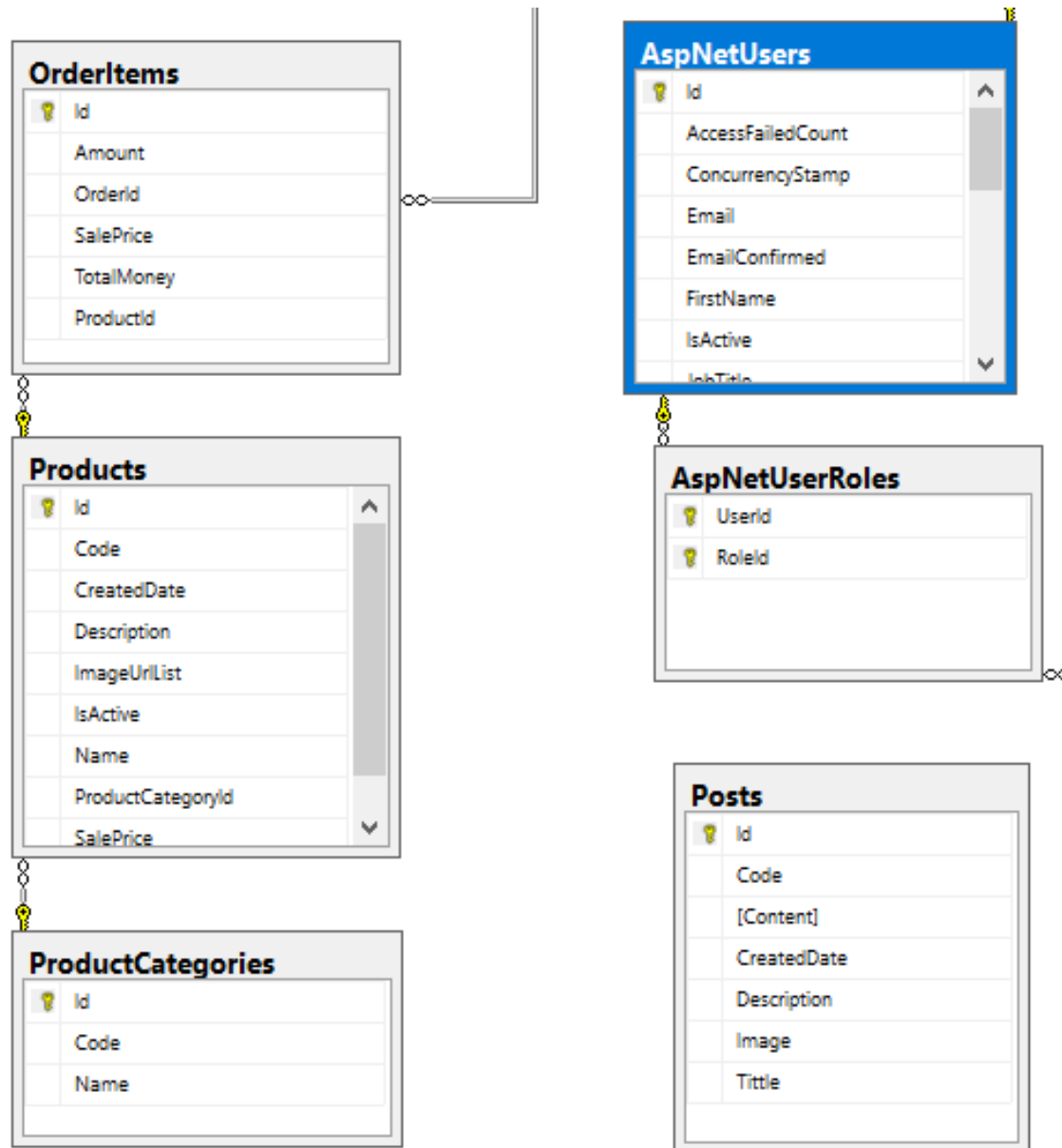


- Id để kiểu dữ liệu **uniqueidentifier** là cơ chế tự sinh id của **Class Guid** trong Entity framework.

- Bảng User(Người dùng): username , id(primary) , password , email, first name, last name, Address, Phone Number, role chứa những thông tin cơ bản của người dùng. Trong đó Username và Password là thông tin cần thiết để người dùng đăng nhập vào phần mềm. Role là Field để phân biệt đó người dùng hay admin.

- Bảng Order(Hóa đơn): id(primary), Code, AddressTotalMoney, CreateDate, UserId là những field hiển thị sẽ giúp cho người dùng thấy những hóa đơn của mình đã đặt hàng với tổng giá tiền với những hóa đơn đó.

❖ Về phía Quản trị



- Bảng ProductCategories(Danh mục sản phẩm): id(primary), code , name.Chứa những thông tin mã và tên danh mục sản phẩm.
- Bảng Product(Sản phẩm): Id(primary) , Code , Name, CreatedDate, IsActive, ImageUrlList, ProductCategory,SalePrice,chứa những thông tin cho một sản phẩm. Với ProductCategoryId ta có thể xác định được sản phẩm đó thuộc danh mục nào. IsActive thể hiện còn hàng hay hết hàng của một loại sản phẩm.
- Bảng Post(Bài đăng): id(primary), tittle ,code, Image, chứa những thông tin cơ bản của bài đăng giúp quản trị có thể quản lý từng bài đăng.

- Ngoài ra 2 bảng Role và UserRole:
 - Role: id(primary) , name, là 2 field dùng để phân quyền cho một người dùng.
 - UserRole: idUser(primary) ,idRole(primary), có chức năng lưu những idUser ứng với từng idRole(Người dùng nào có vai trò nào) thông qua id.

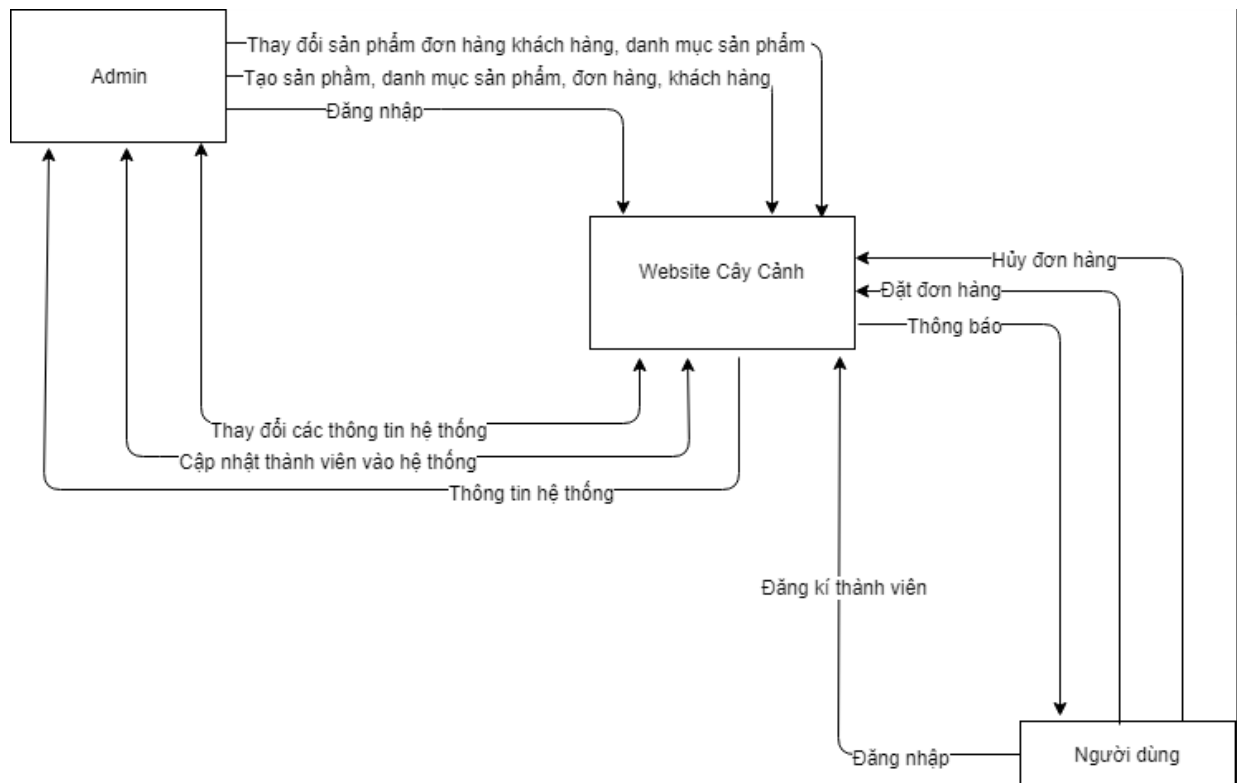
3.3. Phân tích các chức năng xử lý :

❖ Về phía Server:

- Cơ sở dữ liệu được lưu vào SQL Server
- Tạo sơ sở dữ liệu với CodeFirst qua Entity Framework

❖ Về phía Client:

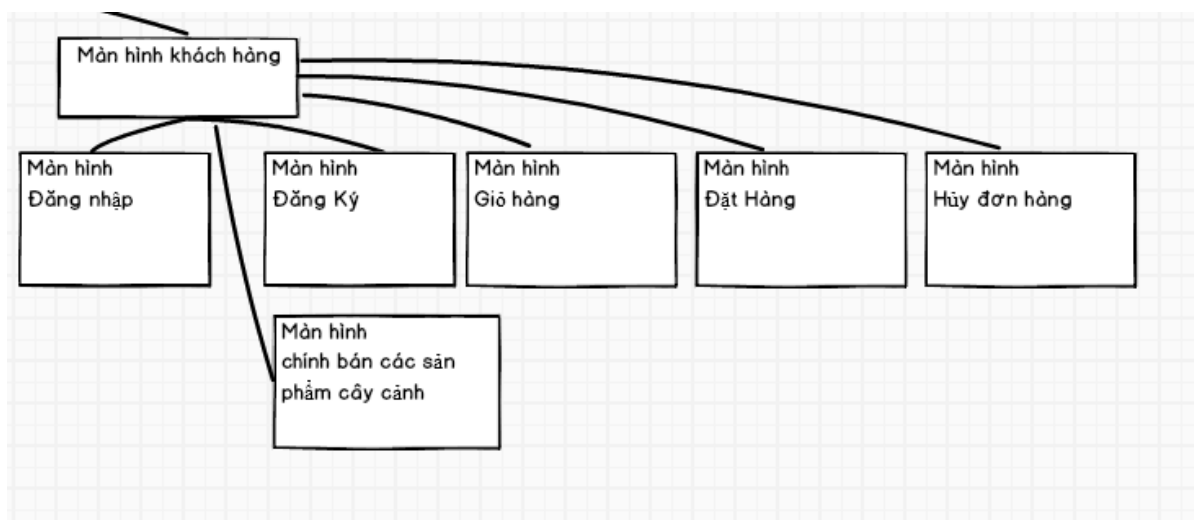
- Khách hàng:
 - Đăng kí thành viên,
 - Đăng nhập
 - Thêm sản phẩm vào giỏ hàng
 - Đặt mua hàng và hủy các đơn hàng.
- Admin(Quản trị): Quản lý các hạng mục của hệ thống.
 - Quản lý Người dùng
 - Quản lý Sản phẩm
 - Quản lý Hóa đơn
 - Quản lý Bài đăng
- Các quản lý hạng mục ta có thể thực hiện các thao tác thêm xóa sửa và tìm kiếm.
- Với lượt đồ BFD ta sẽ rõ từng chức năng của hệ thống với 2 đối tượng Admin và người dùng:



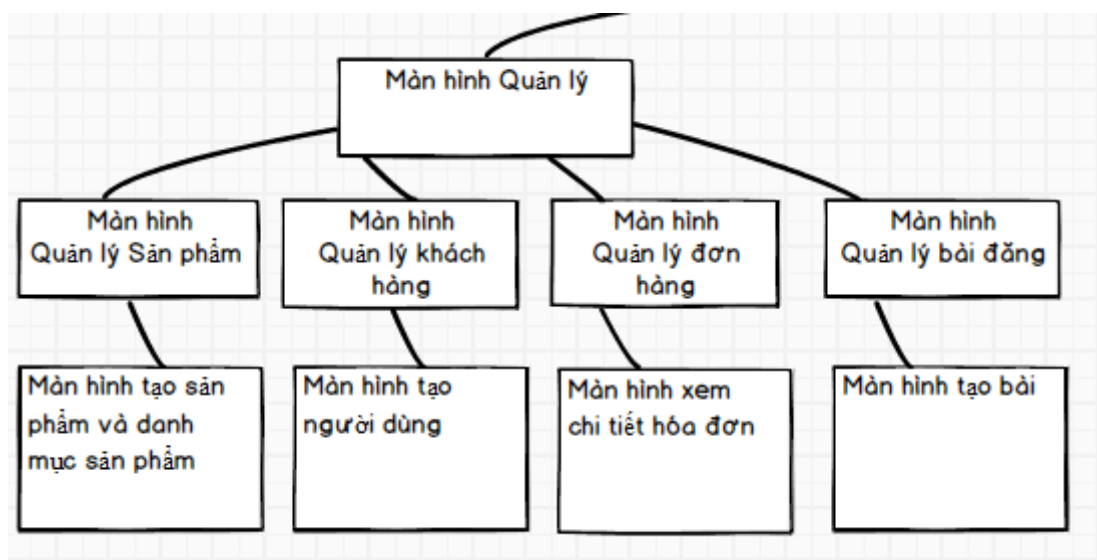
Hình 3.2. Lược đồ BFD.

3.4. Phân tích thiết kế giao diện chính cho đề tài :

- **Người dùng(User):** Trang giao diện khách hàng với:



- Trang chủ: hiển thị các sản phẩm cần bán, mỗi sản phẩm có tên sản phẩm, tên danh mục sản phẩm và button thêm vào giỏ hàng. Các button như giỏ hàng, menu đăng nhập đăng xuất và button xem tin tức.
- Trang đăng kí thành viên: cho phép người dùng đăng kí tài khoản, gồm các input điền thông tin cho khách hàng và button đăng kí.
- Trang đăng nhập: cho phép người dùng đăng nhập tài khoản của mình, có username và password, một button đăng nhập.
- Trang tin tức: hiển thị các tin tức mới về sản phẩm, cách dùng. Gồm có các thẻ image, các label thể hiện tiêu đề của bài đăng và content thể hiện nội dung của tiêu đề đó..
- Trang về giỏ hàng: hiện các sản phẩm được đặt hàng với các nút thêm giảm số lượng và thành tiền cho tất cả các sản phẩm được thêm vào giỏ hàng.
- Trang đặt mua hàng: hiển ra thông tin của người dùng với username, địa chỉ sẽ được gửi đến và số điện thoại. Button đặt đơn hàng chức năng đặt đơn hàng.
- **Quản lý (Admin):** Trang giao diện cho quản lý với:



- Màn hình đăng nhập: admin đăng nhập với vai trò là Owner sẽ gồm một menu bên trái với các mục quản lý, bên góc phải là menu đăng nhập.
- Một menu với các hạng mục quản lý: bao gồm các hạng mục quản lý được thiết kế với template của ngx

- Quản lý sản phẩm: hiển thị các sản phẩm được quản lý và một màn hình tạo sản phẩm để được thêm vào để bán
- Quản lý đơn hàng: hiển thị các đơn hàng được quản lý và một màn hình xem chi tiết đơn hàng của người dùng, đơn hàng của người dùng đặt mua có những sản phẩm nào.
- Quản lý người dùng: hiển thị các user được quản lý và một màn hình tạo user đăng nhập với compopox vai trò lựa chọn user hay quản trị
- Quản lý các bài đăng: hiển thị các bài đăng được quản lý và một màn hình tạo bài đăng mới cho website.

CHƯƠNG IV: GIỚI THIỆU SẢN PHẨM CỦA ĐỀ TÀI**4.1. Giới thiệu sản phẩm và chức năng:****4.1.1. Giới thiệu sản phẩm:**

- ❖ Tên sản phẩm:
 - Website thiết kế và đặt mua cây cảnh cho cửa hàng bán cây cảnh
- ❖ Chức năng:
 - Đặt mua cây cảnh online trực tuyến cho khách hàng
 - Quản lý hệ thống cửa hàng cây cảnh
- ❖ Đối tượng sử dụng:
 - Người dùng: khách của cửa hàng
 - Quản trị cửa hàng: chủ của cửa hàng.

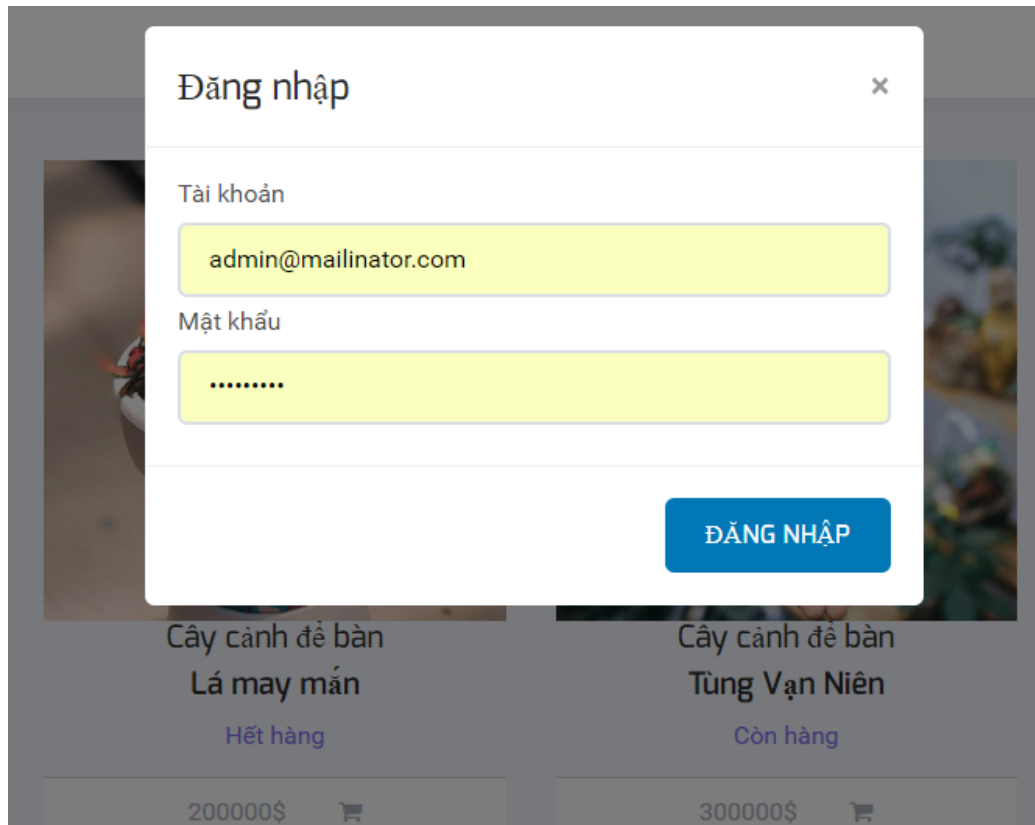
4.1.2. Yêu cầu kỹ thuật:

- Host Web: Angular 4
- Service API : ASP.NET Core
- Database : SQL Server 2014

4.2. Giới thiệu màn hình và hệ thống các chức năng:**4.2.1 Hệ thống cho khách hàng:****4.2.1.1 Màn hình đăng nhập:**

Thực hiện chức năng đăng nhập cho người dùng:

- Button close: thoát để trở về màn hình chính
- Input tài khoản: Người dùng sẽ nhập tên tài khoản để đăng nhập vào.
- Input mật khẩu: với type :”password” nên sẽ mã hóa an toàn khi người dùng nhập vào thì sẽ không hiển thị nội dung của mật khẩu
- Button đăng nhập: Khi nhấn vào sẽ thực hiện kiểm tra nội dung trong 2 input, nếu một trong hai nội dung chưa có sẽ hiển thị thông báo để người dùng nhập lại. Sau đó sẽ gọi đến api đăng nhập nếu đăng nhập thành công sẽ quay lại màn hình chính và thông báo đăng nhập thành công.



Hình 4.1. Màn hình đăng nhập

4.2.1.2 Màn hình đăng kí

Thực hiện chức năng đăng ký tài khoản để truy cập

- Label hiển thị nội dung: “Đăng kí”.
- Các thẻ input Tài khoản(username), mật khẩu, Họ, tên, Email, Số điện thoại, Địa chỉ để người dùng nhập thông tin đăng kí tài khoản.
- Button đăng kí: kiểm tra người dùng đã nhập hết nội dung trong Đăng kí hay chưa, nếu chưa sẽ thông báo cho khách đang bị trống. Sau khi kiểm tra xong sẽ thực hiện việc api đăng kí tài khoản.
- Với việc đăng kí tài khoản này thì password sẽ được mã hóa.
- Việc mã hóa này do sử dụng thư viện: **namespace**

Microsoft.AspNetCore.Identity do ASP.NET core cung cấp với việc sử dụng hàm

+ **public virtual Task<IdentityResult> CreateAsync(TUser user, string password);**

Đăng kí ✕

Tài khoản * Mật khẩu *

Tên tài khoản không thể thay đổi khi đã tạo.

Thông tin bắt buộc

Tên * Họ *

Email * Điện thoại

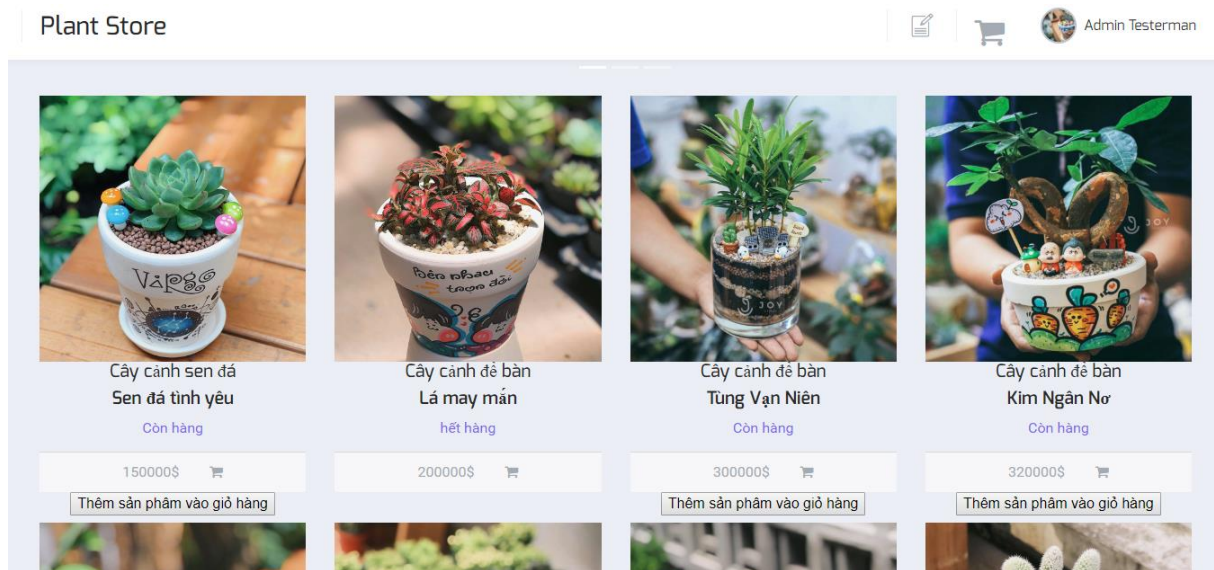
Địa chỉ

ĐĂNG KÍ

Hình 4.2. Màn hình đăng ký

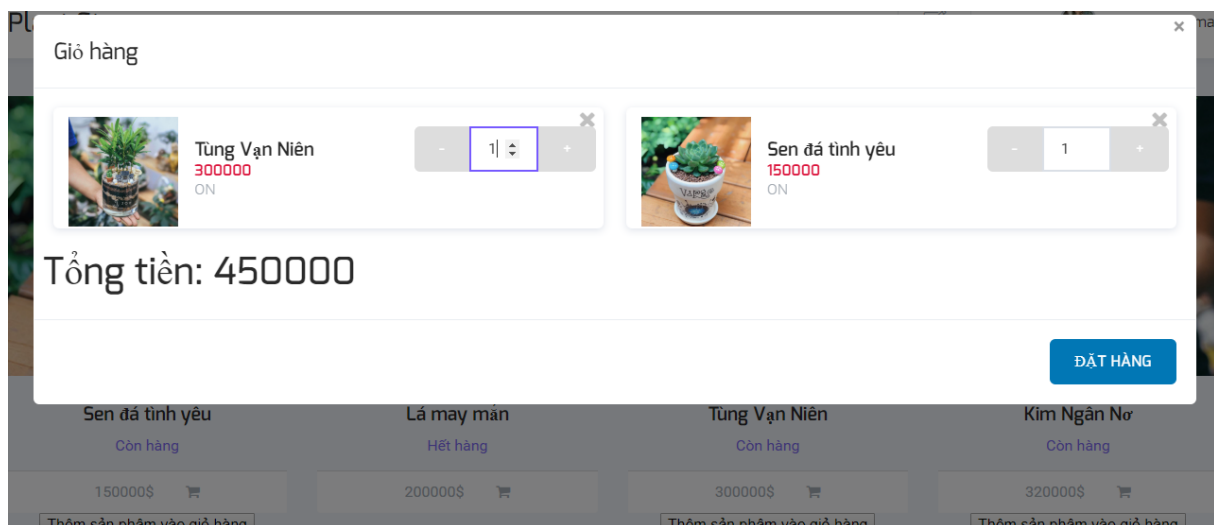
4.2.1.3 Màn hình trang chủ

- Màn hình chính cho khách hàng để hiện ra các sản phẩm.
- Mỗi sản phẩm gồm có hình ảnh , tên sản phẩm, giá sản phẩm và giá sản phẩm.
- Mỗi sản phẩm gồm có hình ảnh , tên sản phẩm, giá sản phẩm và giá sản phẩm.
- Với mỗi sản phẩm hết hàng bạn sẽ không thể thêm vào giỏ hàng được.
- Bên phải một menu nếu đã đăng nhập vào thì nó sẽ hiện ra Đơn đặt hàng của tôi và đăng xuất. Còn nếu chưa đăng nhập thì sẽ hiện một menu : Đăng nhập và Đăng kí
- Người dùng phải đăng nhập để được đặt hàng và thêm những sản phẩm vào giỏ hàng. Nếu khách hàng chưa có tài khoản thì phải đăng kí tài khoản.
- Sau khi đăng ký với quyền là user thì bạn sẽ đăng nhập và xem tất cả sản phẩm của cửa hàng thêm giỏ hàng và đặt hàng.



Hình 4.3. Màn hình trang chủ

4.2.1.4 Màn hình thêm vào giỏ hàng



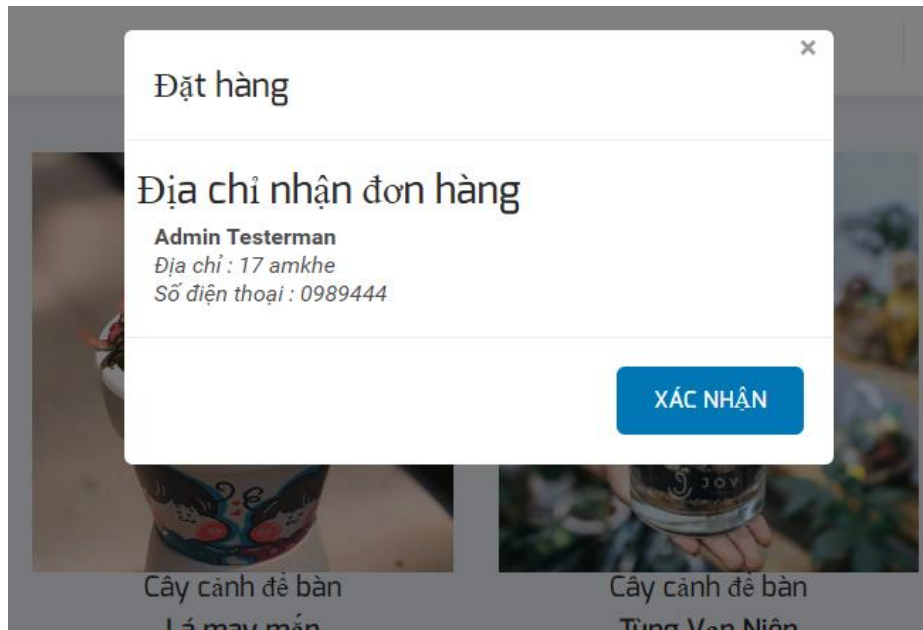
Hình 4.4. Màn hình giỏ hàng

- Với mỗi lần bạn thêm vào giỏ hàng thì dữ liệu sẽ tạm thời lưu vào Local trong trình duyệt.
- Khi khách hàng thêm sản phẩm vào giỏ hàng sẽ hiện thông báo thêm vào giỏ hàng thành công.
- Khách hàng điều chỉnh số lượng với sản phẩm. Số tiền mà bạn phải trả cho đơn hàng sẽ tự động điều chỉnh với số lượng và giá tiền.

- Với một nút closer của mỗi sản phẩm thì bạn có thể xóa sản phẩm đó ra khỏi giỏ hàng

4.2.1.5 Màn hình đặt hàng

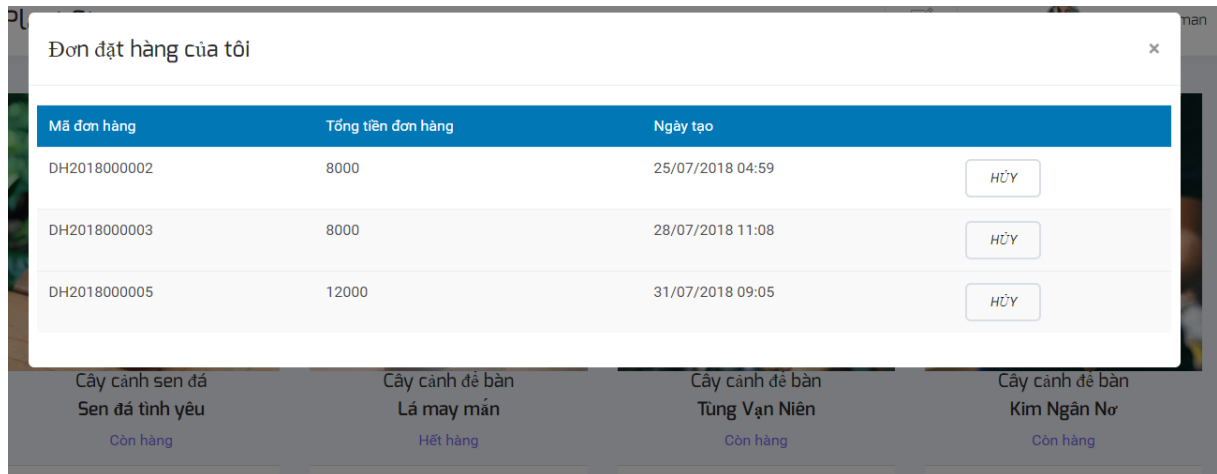
- Đặt hàng(sau khi đã thêm sản phẩm vào giỏ hàng)
- Tiến hành đặt với những thông tin mà bạn đã tạo khi bạn đăng kí thành viên gồm có địa chỉ số điện thoại và tên người mua.



Hình 4.5. Màn hình đặt hàng

4.2.1.6 Màn hình hủy đơn hàng

- Sau khi đặt hàng với tài khoản của mình thì đơn hàng sẽ được lưu trong Đơn hàng của tôi của khách hàng.
- Màn hình đơn hàng của tôi sẽ hiển thị các thông tin đơn hàng của khách hàng với Mã đơn hàng, tổng tiền đơn hàng ngày tạo.
- Button hủy đơn hàng cho phép người hủy đơn hàng của mình đã đặt hàng.

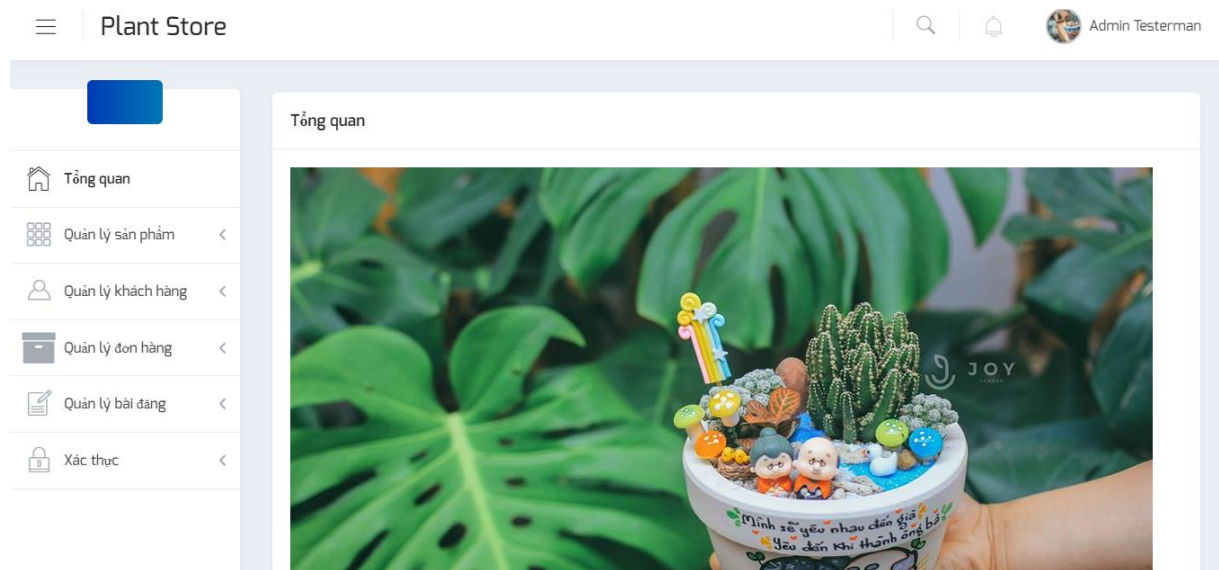


Hình 4.6. Màn hình hủy đơn hàng

4.2.2 Hệ thống cho Quản lý(Admin):

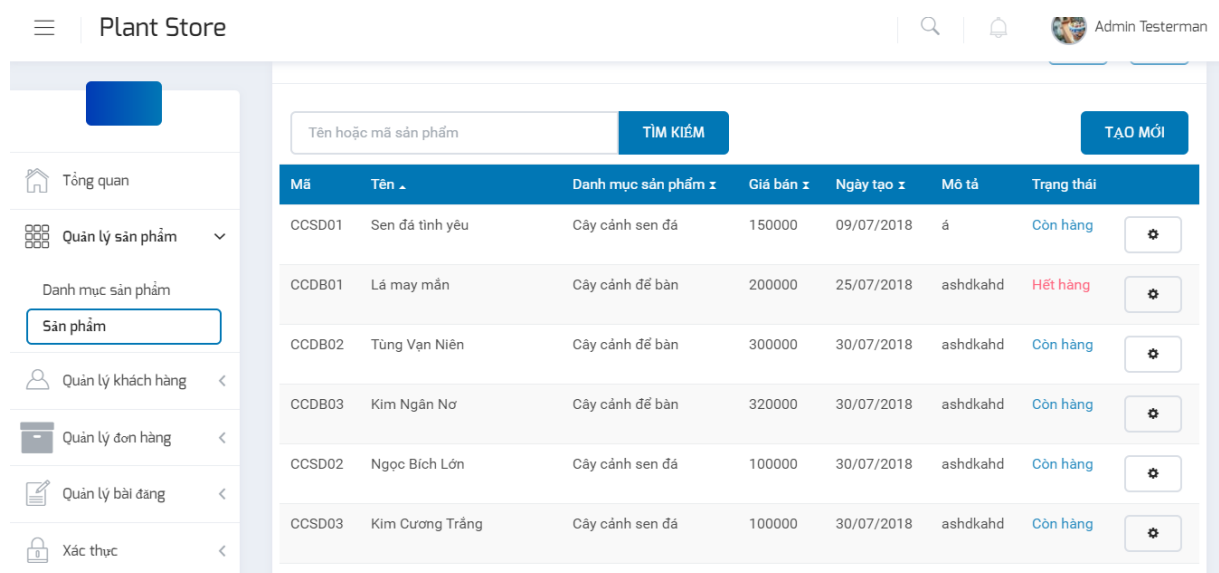
4.2.2.1 Màn hình trang chủ

- Màn hình chính sau khi đăng nhập :
 - Bên trái màn hình sẽ là danh sách menu list với các mục quản lý
 - Bên trên góc phải màn hình ta sẽ có menu đăng nhập và đăng xuất tài khoản.
 - Tổng quan sẽ hiện lên một thẻ image hình của cửa hàng
- Quản lý có 4 mục :
 - Quản lý sản phẩm
 - Sản phẩm
 - Danh mục sản phẩm
 - Quản lý khách hàng
 - Quản lý đơn hàng
 - Quản lý bài đăng



Hình 4.7. Màn hình trang chủ quản lý

4.2.2.2 Màn hình quản lý sản phẩm



Hình 4.8. Màn hình quản lý sản phẩm

- Mục quản lý sản phẩm sẽ chia 2 mục: Sản phẩm và Danh mục sản phẩm
- Mục sản phẩm: chứa một table hiển thị các sản phẩm thể hiện đầy đủ các thuộc tính của sản phẩm đó.
- Mỗi sản phẩm sẽ có một button bên phải ngoài cùng. Button này có 2 chức năng:
 - Xóa: Xóa sản phẩm đó ra khỏi danh sách sản phẩm
 - Sửa: Update lại sản phẩm đó trong danh sản phẩm đó
- Button tạo mới có chức năng tạo mới một sản phẩm hay bất kì đối tượng nào như người dùng, danh mục sản phẩm và bài đăng
- Thanh mục tìm kiếm sản phẩm giúp quản lý search theo bất mã sản phẩm nào.

4.3. Giới thiệu các mã lệnh cơ bản dùng trong đề tài:

4.3.1 Mã lệnh gọi API đăng nhập

```
async login() {  
  try {  
    // lấy user và password vào data  
    let data = await this.authService.login(this.user.username,  
      this.user.password);  
    // lấy user và role hiện tại  
    var userProfile = await this.userService.getCurrentUser();  
    this.helperService.setLocalStorage(CONSTANT.USER_PROFILE, userProfile);  
    this.helperService.setLocalStorage(CONSTANT.CURRENT_ROLE,  
      userProfile.roleNames[0]);  
    // kiểm tra quyền đăng nhập và đặc tính của nó là isActive  
    if(userProfile.roleNames[0] == CONSTANT.ROLES.USER || userProfile.isActive  
      === false){  
      this.authService.logout();  
      this.toastrService.error("Tài khoản không có quyền đăng nhập", "error");  
    }  
    else{  
      // Nếu role là Owner thì cho phép đăng nhập  
      this.router.navigateByUrl('pages');  
    }  
  }  
}
```

4.3.2 Mã lệnh gọi API lấy các đối tượng trong danh sách

- Về phía Client: một api có cấu tạo sau:

```
async getAll(): Promise<any> {  
  try {  
    // header đây chứa token và kiểu json  
    let headers = this.helperService.getHeadersRequest();  
    let options = new RequestOptions({ headers: headers });  
    // hàm gọi api từ controller bên phía server  
    const response = await this.http.get(this.domain + "/all", options)  
      .toPromise();  
    return response.json();  
  } catch (error) {  
    this.helperService.handleError(error);  
  }  
}
```

4.3.3 Mã lệnh gọi API xóa một đối tượng trong danh sách

```
async delete(id: string): Promise<any> {  
  try {  
    // header đây chứa token và kiểu json  
    let headers = this.helperService.getHeadersRequest();  
    let options = new RequestOptions({ headers: headers });  
    let url = this.domain + "/" + id;  
    // hàm gọi api từ controller bên phía server  
    const response = await this.http.delete(url, options)  
      .toPromise();  
    return response.json();  
  } catch (error) {  
    this.helperService.handleError(error);  
  }  
}
```

4.3.4 Mã lệnh thay đổi trạng thái của một đối tượng

```
public async Task<Guid> ChangeActiveStatusAsync(Guid id, bool isActive)  
{  
  //Lấy ra product có mã là id  
  var product = await _entity.SingleOrDefaultAsync(p => p.Id == id);  
  if (product == null)  
  {
```

```
throw new Exception("Can not find product with id=" + id);
}
//kiểm tra biến trạng thái đó có trùng với trạng thái hiện tại
if (product.IsActive == isActive)
{
throw new Exception("Can not change active status!");
}
product.IsActive = isActive;
_entity.Update(product);
//Lưu trạng thái đã thay đổi
var updated = await _context.SaveChangesAsync();
if (updated < 1)
{
throw new InvalidOperationException("Database context could not updated
product.");
}

return product.Id;
}
```

CHƯƠNG V: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết Luận:

5.1.1 Những kết quả đạt được

❖ Lý thuyết

- Hiểu và sử dụng được Angular 4 cho thiết kế giao diện.
- Hiểu và nắm được quy trình của ASP.NET Core.

❖ Thực hành

- Chương trình đã thực hiện được các chức năng:
 - Với người dùng (khách hàng):
 - Thực hiện được giao dịch mua bán như đặt hàng mình muốn mua.
 - Có thể đăng kí được tài khoản và đăng nhập.
 - Xem đơn hàng của mình và có thể hủy đơn hàng của mình đã đặt.
 - Với người quản trị cửa hàng:
 - Quản lý được các mục như đơn hàng, hóa đơn , người dùng, sản phẩm
 - Quản lý các bài đăng của mình.
 - Đối với từng mục quản lý có thể xem được chi tiết, thực hiện các chức năng thêm xóa sửa.

5.1.2. Hạn chế

- Nhiều phần chức năng chưa tối ưu cho người dùng.
- Chưa thể triển khai trên hệ thống trên icloud.
- Tồn kho và doanh thu chưa thể xử lý.
- Thời gian đặt đơn hàng chưa có

5.1.3 Hướng khắc phục

- Cải tiến các chức năng đã thực hiện được, tối ưu nhất cho người dùng và người quản trị.
- Triển khai được hệ thống trên icloud.
- Với tồn kho của sản phẩm cập nhật đúng sản phẩm với số lượng tồn và số lượng còn trong mỗi sản phẩm. Doanh thu hoàn chỉnh với các hạng mục của việc bán hàng.

- Thời gian đặt đơn hàng sẽ có thời hạn nhất định để việc hủy đơn hàng chính xác hơn

5.2 Hướng phát triển:

- Để có thể tạo cho người dùng một cách thuật tiện trong việc lựa chọn cây thì thêm phần thiết kế cho cây và chậu cây.
- Tạo thêm các file báo cáo cho người quản lý để họ có thể quản trị một cách tối ưu nhất.
- Hỗ trợ giao diện đa ngôn ngữ.
- Nâng cao tính bảo mật cho website.

TÀI LIỆU THAM KHẢO

[1] Akveo team, Admin template , 2017

<https://github.com/akveo/ngx-admin>

[2] W3school, 2018

<https://www.w3schools.com/>

[3] Document about Angular,2018

<https://angular.io/docs>

[4] Microsoft for developer,doc ASP.NET core,2017

<https://docs.microsoft.com/en-us/aspnet/core/getting-started/?view=aspnetcore-2.1&tabs=windows>