**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 CÁM 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 bỡ ngỡ. 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 Thể 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**](#_Toc520991380)

[**CHƯƠNG I: GIỚI THIỆU ĐỀ TÀI 2**](#_Toc520991381)

[1.1. Giới thiệu đề tài 2](#_Toc520991382)

[1.2. Mục đích đề tài 2](#_Toc520991383)

[1.3. Mục tiêu đề tài 2](#_Toc520991383)

[1.4. Công nghệ sử dụng 2](#_Toc520991383)

[1.5. Lĩnh vực 2](#_Toc520991383)

[**CHƯƠNG II: GIỚI THIỆU CÔNG NGHỆ SỬ DỤNG 3**](#_Toc520991384)

[2.1. Giới thiệu về Bootstrap 4. 3](#_Toc520991385)

[2.2. Các tính năng mới trong HTML5 4](#_Toc520991386)

[2.3. Các tính năng mới trong CSS 5](#_Toc520991387)

[2.4. Giới thiệu về Framework Angular 5](#_Toc520991388)

[2.4.1. Sơ lược về Angular 5](#_Toc520991389)

[2.4.2. Angular 4 : 7](#_Toc520991391)

[2.5. Giới thiệu ASP.NET CORE 1.0 8](#_Toc520991392)

[2.5.1. ASP.NET là gì? 8](#_Toc520991393)

[2.5.2. ASP.NET Core là gì ? 9](#_Toc520991394)

[**CHƯƠNG III: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 13**](#_Toc520991395)

[3.1. Mô hình dữ liệu thực thể kết hợp. 13](#_Toc520991396)

[3.1.1. Danh sách các thực thể 13](#_Toc520991397)

[3.1.2. Lượt đồ BFD 16](#_Toc520991398)

[3.2. Mô hình cơ sở dữ liệu. 16](#_Toc520991399)

[**CHƯƠNG IV: GIỚI THIỆU SẢN PHẨM MINH HỌA 17**](#_Toc520991400)

[4.1. Thiết kế giao diện khách hàng và chức năng: 17](#_Toc520991401)

[4.1.1. Giao diện chính: 17](#_Toc520991402)

[4.1.2. Chức năng: 18](#_Toc520991403)

[4.2. Thiết kế giao diện quản lý và chức năng: 20](#_Toc520991404)

[4.2.1. Giao diện chính: 20](#_Toc520991405)

[4.2.2. Chức năng 20](#_Toc520991406)

[**KẾT LUẬN 23**](#_Toc520991407)

[**DANH MỤC TÀI LIỆU THAM KHẢO 24**](#_Toc520991414)

# DANH MỤC HÌNH

[Hình 2.1. Cách tổ chức cấu trúc thư mục cho Component 7](#_Toc520991247)

[Hình 2.2. Mô hình xây dựng Angular 4 cơ bản. 8](#_Toc520991248)

[Hình 2.3. Mô hình xây dựng .Net Core MVC 12](#_Toc520991249)

[Hình 3.1. Lượt đồ phân rã chức năng. 16](#_Toc520991250)

[Hình 3.2. Mô hình cơ sở dữ liệu 16](#_Toc520991251)

# Danh mỤc các ký hiỆu và cHỮ viẾt tẮt

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

# 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 đuợc 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.

1. GIỚI THIỆU ĐỀ TÀI
   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 hiên mua bán mà còn giúp cho người chủ thực hiên 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 thoãi 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.

1.3. Mục tiêu đề tài

* Xây dựng Website thiết kế và đặt mua cây cảnh
  + 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
  + Dịch vụ đặt mua hàng(Khách hàng)
    - Xem sản phẩm ưu thích
    - Đặt mua hàng
    - Hủy đơn hàng mình đã đặt mua

1.4. Công nghệ sử dung

* + 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 sơ dữ liệu SQL Server
* Lĩnh vực liên quan: Bán hàng

1. GIỚI THIỆU CÔNG NGHỆ SỬ DỤNG

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](https://twitter.com/mdo) và [@fat](https://twitter.com/fat). Trước khi chính thức trở thành một framework mã nguồn mở, Bootstrap được biến đến như là bản thiết kế của Twitter. Sau một vài tháng đi vào phát triển, Twitter đã tổ chức [Tuần lễ Hack đầu tiên](https://blog.twitter.com/2010/hack-week) 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](https://dev.twitter.com/blog/bootstrap-twitter) vào [Thứ 6, ngày 19 tháng 8, năm 2011](https://twitter.com/mdo/statuses/104620039650557952) tới nay, chúng tôi đã có tới hơn [20 lần phát hành](https://github.com/twbs/bootstrap/releases), 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](http://freetuts.net/tag/sass-css) 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 lả 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](http://freetuts.net/tag/javascript-toan-tap) của nó sử dụng [jquery](http://freetuts.net/tag/jquery-toan-tap) 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 reponsive 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 CSS

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 hổ 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.



* 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.

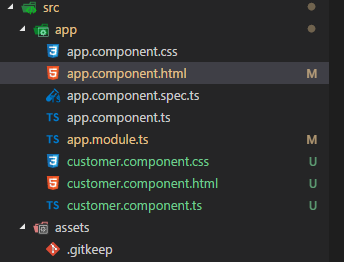
Một component sẽ hình thành từ một view xác định từ một mẫu HTML, Code xử lý sẽ được định nghĩa với một Class và thông tin bổ xung cho Class thì được xác định bằng metadata. Chúng ta sẽ tìm hiểu chi tiết hơn trong những bài sắp tới.

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

* + 1. 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:

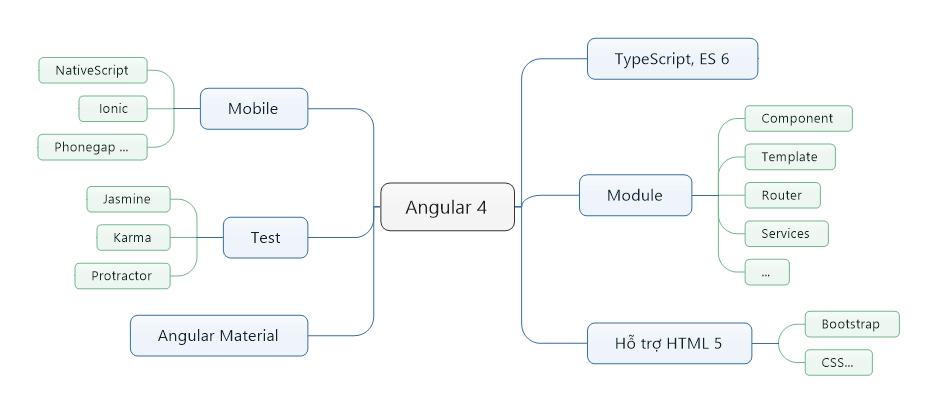
Tạo cấu trúc project với một loạt thư viện tích hợp sẵn: build tool với webpack, unit testing với Karma, biên dịch được SCSS thành CSS.

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 hướng đến toàn bộ ứng dụng web. Do đó, để tránh tình trạng mất kiểm soát các thuocj 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:





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

2.5. Giới thiệu ASP.NET CORE 1.0

* + 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 dụng 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 sao 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.

* + 1. 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, loT 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](https://github.com/aspnet/home)

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 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 chỉ 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 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 thoải 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 điều có chức năng “chỉnh sửa trang này” ở phía trên và bạn có thể chỉnh sửa các tài liệu từ Microsoft

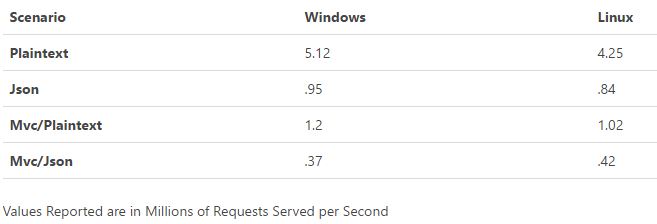
* ***Hỗ trợ đầy đủ cho framework:***

Một thông tin hữu ích là asp.net core dẫu trở thành mã nguồn mở nhưng Microsoft vẫn sẽ cung cấp hỗ trợ trong 3 năm cho mỗi bản phát hành lớn nhỏ của họ.

* ***Hiệu suất:***

Microsoft giới thiệu máy chủ mới web Kestrel chạy trong host IIS của bạn hoặc chạy sau một host process khác. Kestrel hiện tại là máy chủ .NET chạy nhanh nhất hiện nay

Dưới đây là bảng hiệu xuất đánh giá từ Microsoft



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

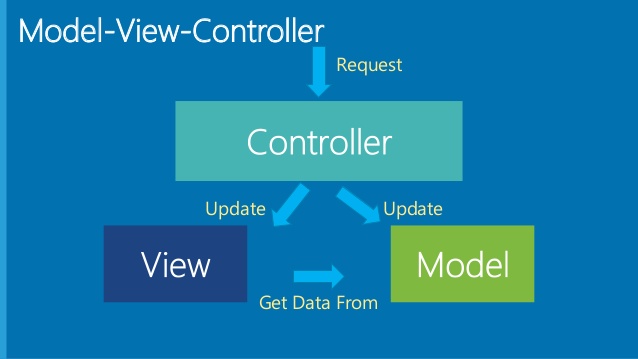
Nếu mà giải thích thì chỉ vài dòng thì sẽ hơi khó hiểu : ASP.NET Core đã được xây dựng trong Dependency Injection.Dependency Injection là một mẫu thiết kế cho phép các phụ thuộc của một class được injected như các đối tượng được yêu cầu trong ứng dụng của bạn. Với ASP.NET Core, Microsoft đã cung cấp 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****

1. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

3.1. Mô hình dữ liệu thực thể kết hợp

3.1.1. Danh sách các thực thể

* Người Dùng(User)

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Notes |
| Id | uniqueidentifier | Primary Key |
| UserName | nvarchar(256) |  |
| Password | nvarchar(MAX) |  |
| FirstName | nvarchar(MAX) |  |
| LastName | nvarchar(MAX) |  |
| Address | nvarchar(MAX) |  |
| PhoneNumber | nvarchar(MAX) |  |
| JobTitle | nvarchar(MAX) |  |
| Email | nvarchar(256) |  |

* Danh mục sản phẩm(ProductCategory)

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Notes |
| Id | uniqueidentifier | Primary Key |
| Code | nvarchar(MAX) |  |
| Name | nvarchar(MAX) |  |

* Sản Phẩm(Product)

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Notes |
| Id | uniqueidentifier | Primary Key |
| Code | nvarchar(MAX) |  |
| CreatedDate | datetime2(7) |  |
| Description | nvarchar(MAX) |  |
| ImageUrlList | nvarchar(MAX) |  |
| IsActive | bit |  |
| Name | nvarchar(MAX) |  |
| SalePrice | float |  |
| ProductCategoryId | uniqueidentifier |  |

* Hóa đơn(Order)

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Notes |
| Id | uniqueidentifier | Primary Key |
| Address | nvarchar(MAX) |  |
| Code | nvarchar(MAX) |  |
| CreatedDate | datetime2(7) |  |
| TotalMoney | float |  |
| UserId | uniqueidentifier |  |
| isDeleted | bit |  |

* Danh sách chi tiết của Hóa đơn(OrderItem)

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Notes |
| Id | uniqueidentifier | Primary key |
| Amount | int |  |
| OrderId | uniqueidentifier |  |
| SalePrice | float |  |
| TotalMoney | float |  |
| ProductId | uniqueidentifier |  |

* Bài Đăng(Post)

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Notes |
| Id | uniqueidentifier | PrimaryKey |
| Code | nvarchar(MAX) |  |
| Content | nvarchar(MAX) |  |
| CreatedDate | datetime2(7) |  |
| Description | nvarchar(MAX) |  |
| Image | nvarchar(MAX) |  |
| Tittle | nvarchar(MAX) |  |

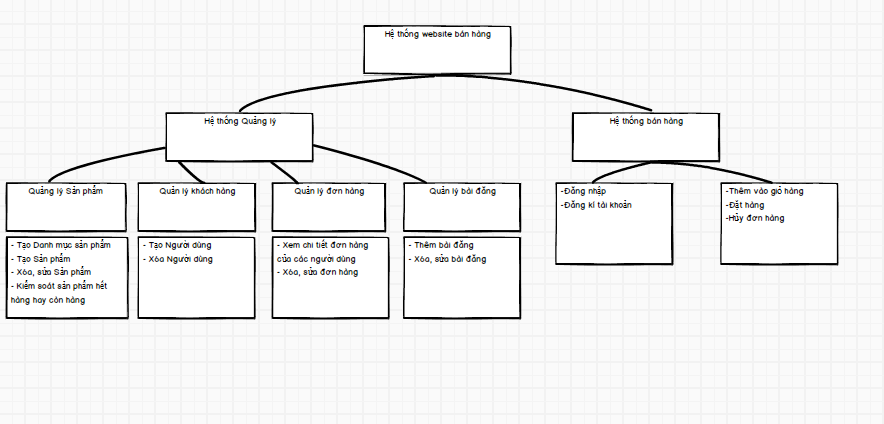
* ASP UserRole :

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Notes |
| UserId | uniqueidentifier | PrimaryKey |
| RoleId | uniqueidentifier | PrimaryKey |

* ASP Role :

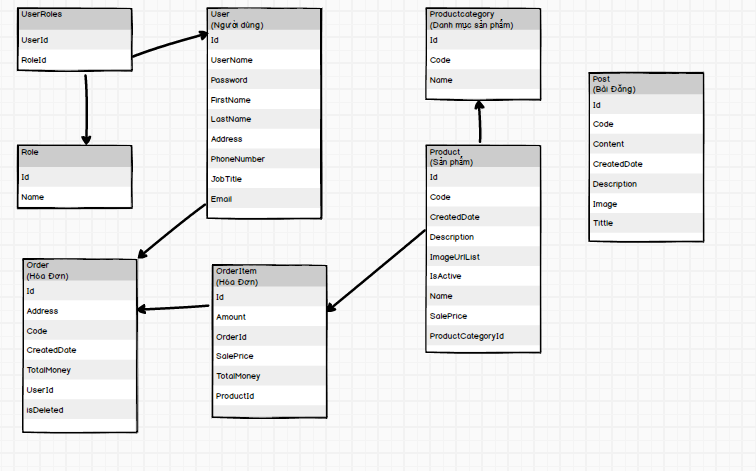
|  |  |  |
| --- | --- | --- |
| Field Name | Type | Notes |
| Id | uniqueidentifier | PrimaryKey |
| Name | nvarchar(256) |  |

* + 1. Lượt đồ BFD



**Hình 3.1. Lượt đồ phân rã chức năng.**

* 1. Mô hình cơ sở dữ liệu.

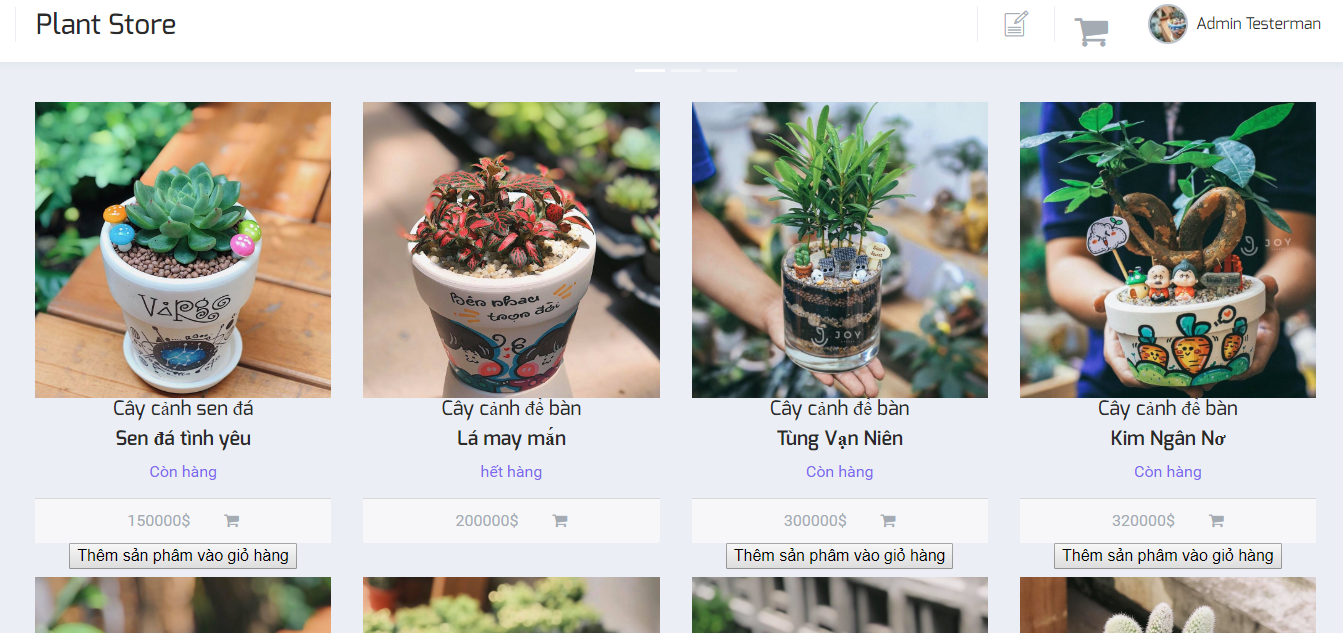


**Hình 3.2. Mô hình cơ sở dữ liệu**

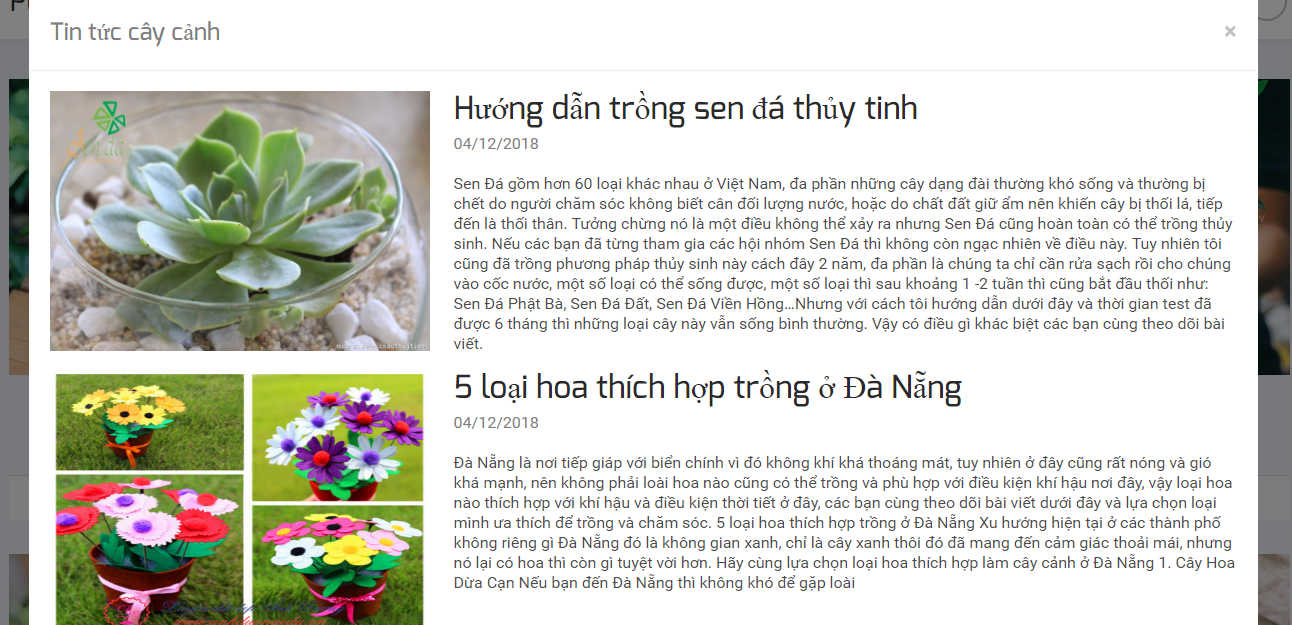
1. DEMO CHƯƠNG TRÌNH
   1. **Thiết kế giao diện khách hàng và chức năng:**

**4.1.1. Giao diện chính:**

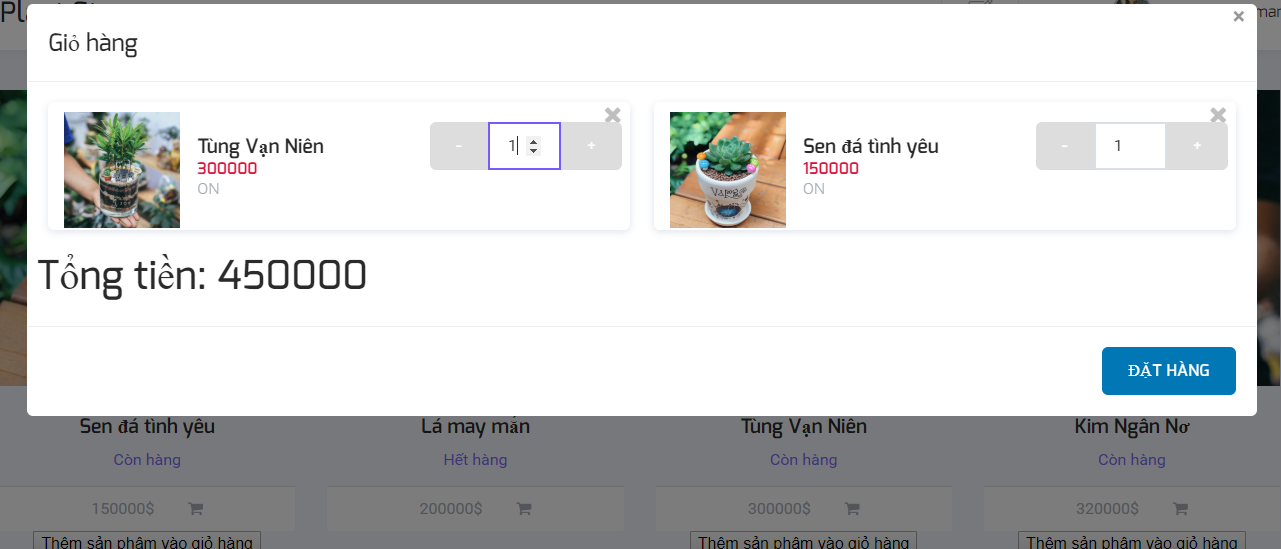
* Giao diện cho người dùng:

****

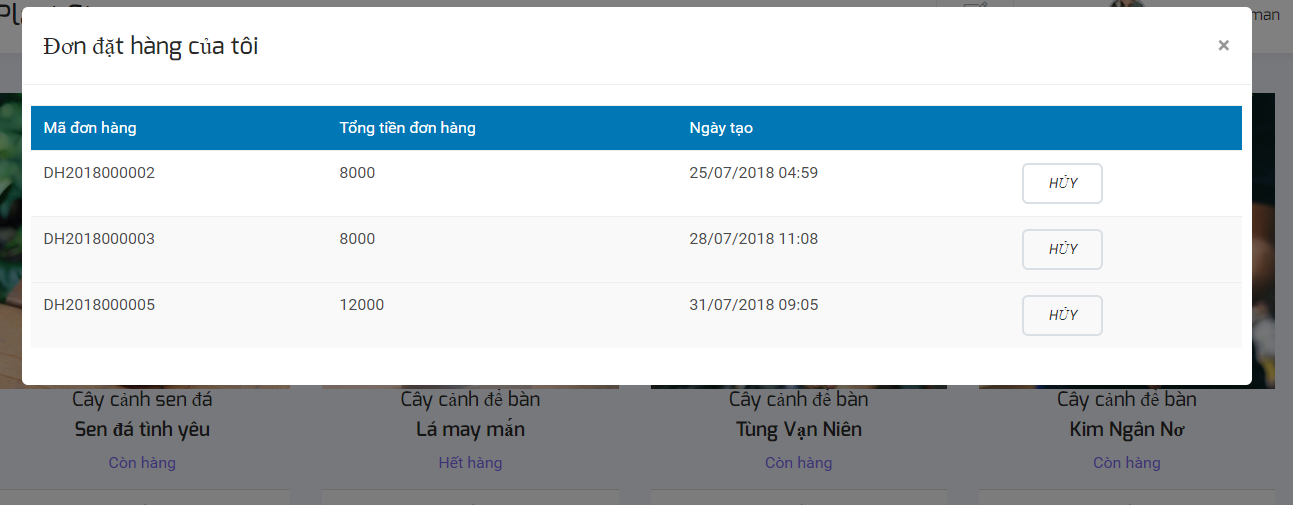
* Giao diện tin tức cho người dùng:

****

* + 1. **Chức năng:**
* Các chức năng chính:
* Người dùng phải đăng nhập để được đặt hàng và thêm những sản phầm 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
* Khi khách hàng thêm sản phẩm vào giở hàng sẽ hiên 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



* 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
* Khi khách hàng thành công nếu muốn hủy đơn hàng thì có thể vào mục đơn hàng của tôi để hủy

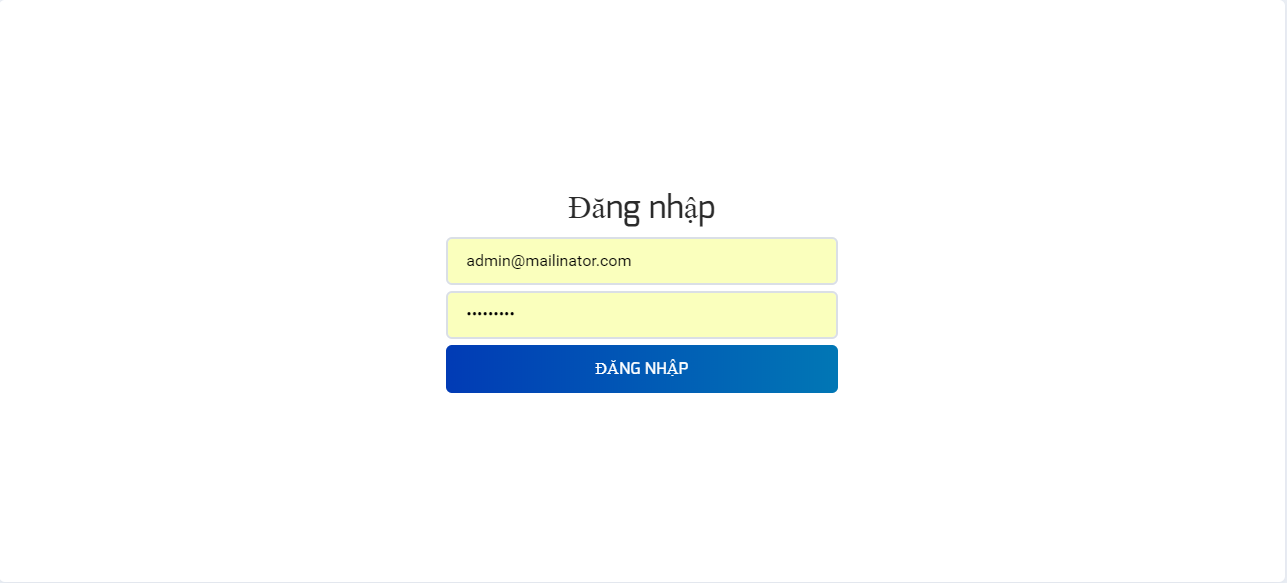


## 4.2. Thiết kế giao diện quản lý và chức năng:

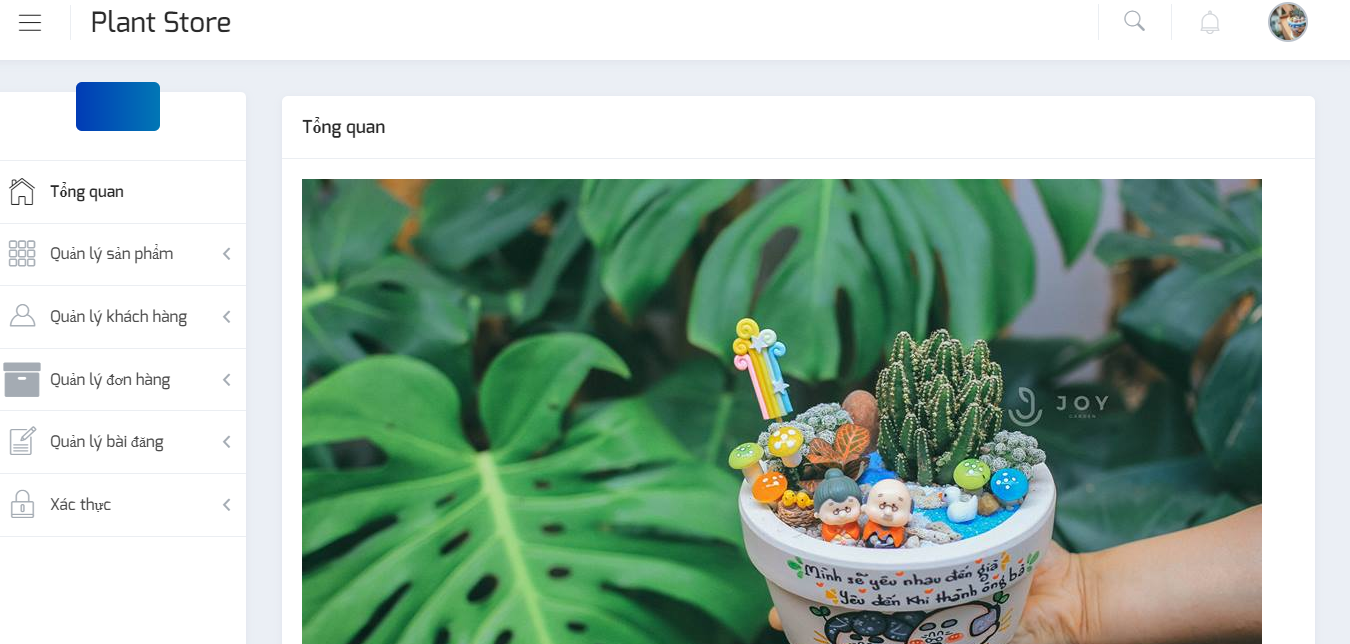
### 4.2.1. Giao diện chính:

* Màn hình đăng nhập cho admin:

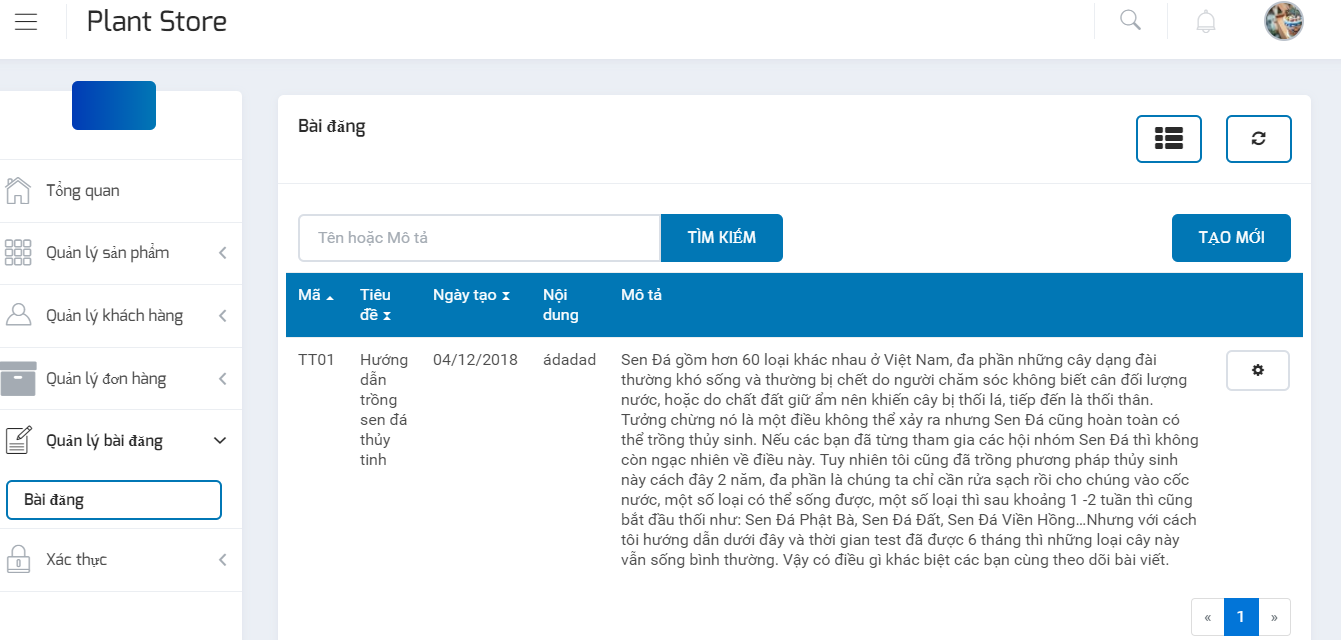
Admin nhập user và password để đăng nhập vào trang quản lý



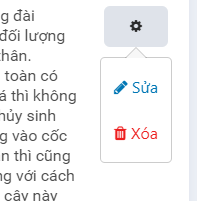
* Màn hình chính sao khi đăng nhập :



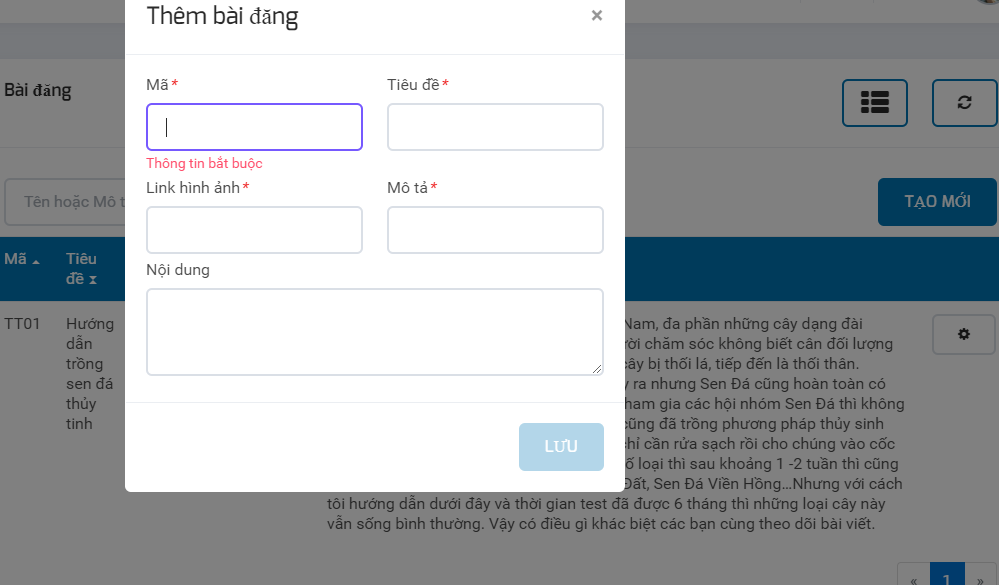
* + 1. **Chức năng**
* Có 4 chức năng:
  + Quản lý sản phẩm
  + Sản phẩm
  + Danh mục sản phẩm
  + Quản lý khách hàng
  + Quản lý đơn hàng
  + Quản lý bài đăng
* Với từng mục Quản lý ta có giao diện và chức năng:

****

* Chức năng :
  + Có thể xem sửa và xóa với từng đối tượng mình quản lý

****

* + Tạo mới đối tượng:

****

* + Tìm kiếm đối tượng trong danh sách

**KẾT LUẬN**

* Hạn chế:
* **Chưa triển khai được hệ thống trên icloud.**
* Tính tiện ích của chương trình chưa được nhiều.
* Về Phần Thiết Kê chưa làm được.
* Chưa quản lý được tồn kho
* Chưa up hình ảnh trực tiếp lên file local
* Kết quả đạt được:
* Giao diện thân thiện.
* Website đảm bảo được các yêu cầu cơ bản như quản lý bài viết, đăng bài, và cho phép mua bán
* Hướng phát triển website:
* Hoàn thiện các tính năng hiện tại và còn thiếu
* Hỗ trợ giao diện đa ngôn ngữ
* Nâng cấp tính bảo mật cho website
* Hỗ trợ cho người dùng có thể thiết kế được chậu cây ưu thích

**DANH MỤC TÀI LIỆU THAM KHẢO**

Danh mục các website tham khảo:

**1**. [**https://github.com/akveo/ngx-admin**](https://github.com/akveo/ngx-admin)

**2.** [**https://www.w3schools.com/**](https://www.w3schools.com/)

**3.** [**https://angular.io/**](https://angular.io/)

**4.** [**https://docs.microsoft.com/en-us/aspnet/core/getting-started/?view=aspnetcore-2.1&tabs=windows**](https://docs.microsoft.com/en-us/aspnet/core/getting-started/?view=aspnetcore-2.1&tabs=windows)