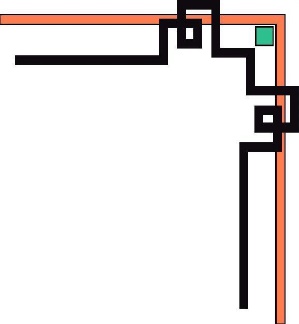
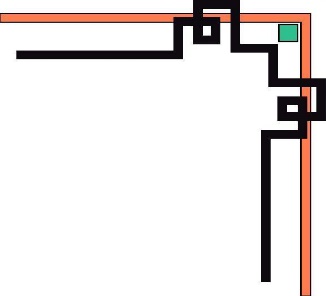
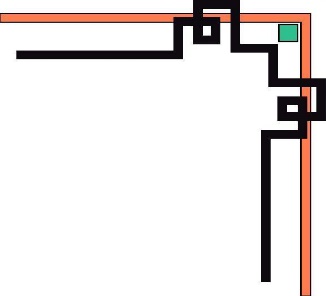
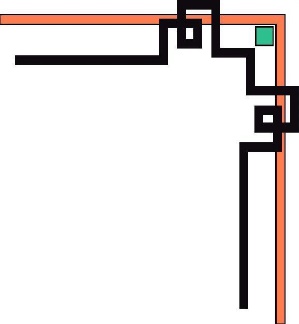
**Bộ giáo dục và đào tạo**



**Trường Đại học Ngoại ngữ - Tin học TP.HCM**



**BÁO CÁO ĐỒ ÁN**

**Các chủ đề nâng cao trong công nghệ phần mềm**

**Đề tài: Xây dựng website bán giày**

Giảng viên hướng dẫn:  Trần Phương Tuấn

Thành viên : Trần Minh Triết -17DH110049

Lê Ngọc Bảo Toàn -17DH110050

Nguyễn Bá Duy Việt-17DH110057

**Tháng 06 Năm 2021**

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

**Mục lục**

[Mở đầu 2](#_Toc75366864)

[Chương 1 :Giới thiệu về ngôn ngữ lập trình 2](#_Toc75366865)

[1. Tổng quan về Angular 2](#_Toc75366866)

[2. Ưu điểm -Nhược điểm 3](#_Toc75366867)

[3. Kiến trúc của Angular 3](#_Toc75366868)

[3.1 Component: 3](#_Toc75366869)

[3.2 Template 4](#_Toc75366870)

[3.3 Metadata 4](#_Toc75366871)

[3.4 Data Binding 4](#_Toc75366872)

[3.5 Directive 4](#_Toc75366873)

[3.6 Service 5](#_Toc75366874)

[3.7 Dependency Injection 5](#_Toc75366875)

[4. Các phiên bản Angular 5](#_Toc75366876)

[5. Các tính năng cơ bản 5](#_Toc75366877)

[6. Các đặc trưng 6](#_Toc75366878)

[7. Ứng dụng 7](#_Toc75366879)

[8. Tại sao nên sử dụng Angular ? 7](#_Toc75366880)

[8.1 Angular giúp nâng cao năng suất của các lập trình viên 7](#_Toc75366881)

[8.2 Cấu trúc phát triển rõ ràng 7](#_Toc75366882)

[8.3 Extensive Biding 8](#_Toc75366883)

[8.4 Hỗ trợ đầy đủ tính năng điều hướng 9](#_Toc75366884)

[8.5 Giảm thiểu tối đa kích thước và tăng tối đa hiệu suất của ứng dụng 9](#_Toc75366885)

[8.6 Document và Community 10](#_Toc75366886)

[Chương 2 - Khảo sát và phân tích hệ thống 10](#_Toc75366887)

[1. Phân tích yêu cầu đề 10](#_Toc75366888)

[1.1 Tên đề tài: 10](#_Toc75366889)

[1.2 Chức năng 11](#_Toc75366890)

[1.3 Yêu cầu đặt ra 11](#_Toc75366891)

[2. Phân tích hệ thống 12](#_Toc75366892)

[3. Đặc tả usecase 12](#_Toc75366893)

[3.1 Đặc tả usecase xem thông tin 12](#_Toc75366894)

[3.2 Đặc tả usecase thêm xóa sửa sản phẩm 12](#_Toc75366895)

[Chương 3 Thiết kế giao diện và cơ sở dữ liệu 14](#_Toc75366896)

[1. Một số hình ảnh của website 14](#_Toc75366897)

[1.1 Giao diện trang chủ 14](#_Toc75366898)

[1.2 Giao diện trang sản phẩm 15](#_Toc75366899)

[1.3 Giao diện trang chi tiết sản phẩm 16](#_Toc75366900)

[1.4 Giao diện trang quản lý 17](#_Toc75366901)

[2. Cơ sở dữ liệu 21](#_Toc75366902)

[Chương 4: Chức năng 22](#_Toc75366903)

[1. Cấu trúc thư mục và các tập tin cần thiết 22](#_Toc75366904)

[1.1 Layout 22](#_Toc75366905)

[1.2 Shared 23](#_Toc75366906)

[1.3 Models 24](#_Toc75366907)

[1.4 API 25](#_Toc75366908)

[1.5 Component 28](#_Toc75366909)

[2. Chức năng 29](#_Toc75366910)

[2.1 Người dùng 29](#_Toc75366911)

[2.2 Admin 32](#_Toc75366914)

[CHƯƠNG III: TỔNG KẾT 47](#_Toc75366915)

[1. Những việc đã làm được 47](#_Toc75366916)

[2. Những việc chưa làm được 47](#_Toc75366917)

[3. Nhận xét quá trình triển khai 47](#_Toc75366918)

[4. Phân công công việc 47](#_Toc75366919)

[5. Tài liệu tham khảo 47](#_Toc75366920)

Danh mục hình ảnh

[Hình 1:Useacse 12](#_Toc75366804)

[Hình 2: Giao diện trang chủ 1.1 14](#_Toc75366805)

[Hình 3:Giao diện trang chủ 1.2 14](#_Toc75366806)

[Hình 4:Giao diện trang chủ 1.3 15](#_Toc75366807)

[Hình 5: Giao diện trang sản phẩm 1.1 15](#_Toc75366808)

[Hình 6:Giao diện trang sản phẩm 1.2 16](#_Toc75366809)

[Hình 7:Giao diện chi tiết sản phẩm 16](#_Toc75366810)

[Hình 8:Giao diện Dashboard 17](#_Toc75366811)

[Hình 9:Giao diện quản lý sản phẩm 17](#_Toc75366812)

[Hình 10:Giao diện thêm sản phẩm 18](#_Toc75366813)

[Hình 11:Giao diện chỉnh sửa sản phẩm 18](#_Toc75366814)

[Hình 12:Giao diện xóa sản phẩm 19](#_Toc75366815)

[Hình 13:Giao diện quản lý loại sản phẩm 19](#_Toc75366816)

[Hình 14:Giao diện thêm loại sản phẩm 20](#_Toc75366817)

[Hình 15:Giao diện chỉnh sửa loại sản phẩm 20](#_Toc75366818)

[Hình 16:Giao diện xóa loại sản phẩm 21](#_Toc75366819)

[Hình 17:Cơ sở dự liệu 21](#_Toc75366820)

[Hình 18:Layout Folder 1.1 22](#_Toc75366821)

[Hình 19:Layout Folder 1.2 22](#_Toc75366822)

[Hình 20:Layout Folder 1.3 23](#_Toc75366823)

[Hình 21:Shared Folder 23](#_Toc75366824)

[Hình 22:Models Folder 1.1 24](#_Toc75366825)

[Hình 23:Models Folder 1.2 24](#_Toc75366826)

[Hình 24:Models Folder 1.3 24](#_Toc75366827)

[Hình 25:API Folder 25](#_Toc75366828)

[Hình 26: Category Service 25](#_Toc75366829)

[Hình 27: Product Service 27](#_Toc75366830)

[Hình 28:Component Folder 28](#_Toc75366831)

[Hình 29:Store-management component 28](#_Toc75366832)

[Hình 30: Product-component 28](#_Toc75366833)

[Hình 31:Chức năng xem sản phẩm 1 29](#_Toc75366834)

[Hình 32:Data Json 30](#_Toc75366835)

[Hình 33:Chức năng xrm sản phẩm 1.1 30](#_Toc75366836)

[Hình 34:Chức năng xem chi tiết sản phẩm 1 31](#_Toc75366837)

[Hình 35: Chức năng xem chi tiết sản phẩm 1.1 32](#_Toc75366838)

[Hình 36: Chức năng xem chi tiết sản phẩm 1.2 32](#_Toc75366839)

[Hình 37:Modal Component (Typescript) 33](#_Toc75366840)

[Hình 38:Modal Component (HTML) 34](#_Toc75366841)

[Hình 39:Chức năng quản lý sản phẩm 1.1 35](#_Toc75366842)

[Hình 40: Chức năng quản lý sản phẩm 1.2 35](#_Toc75366843)

[Hình 41: Chức năng quản lý sản phẩm 1.3 36](#_Toc75366844)

[Hình 42: Chức năng quản lý sản phẩm 1.4 36](#_Toc75366845)

[Hình 43:Chức năng thêm mới sản phẩm 37](#_Toc75366846)

[Hình 44:Chức năng chỉnh sửa sản phẩm 38](#_Toc75366847)

[Hình 45:Chức năng update hình ảnh sản phẩm 38](#_Toc75366848)

[Hình 46: Chức năng xóa sản phẩm 1.1 39](#_Toc75366849)

[Hình 47: Chức năng xóa sản phẩm 1.2 39](#_Toc75366850)

[Hình 48:Chức năng chỉnh sửa 1.3 39](#_Toc75366851)

[Hình 49:Hiển thị dữ liệu dưới dạng table 40](#_Toc75366852)

[Hình 50:Chức năng quản lý loại sản phẩm 1.1 41](#_Toc75366853)

[Hình 51: Chức năng quản lý loại sản phẩm 1.2 41](#_Toc75366854)

[Hình 52: Chức năng quản lý loại sản phẩm 1.3 42](#_Toc75366855)

[Hình 53:Reactive form 42](#_Toc75366856)

[Hình 54:Chức năng thêm mới loại sản phẩm 43](#_Toc75366857)

[Hình 55:Chức năng chỉnh sửa loại sản phẩm 1.1 43](#_Toc75366858)

[Hình 56:Chức năng chỉnh sửa loại sản phẩm 1.2 44](#_Toc75366859)

[Hình 57:Chức năng xóa loại sản phẩm 1.1 44](#_Toc75366860)

[Hình 58:Chức năng xóa loại sản phẩm 1.2 44](#_Toc75366861)

[Hình 59:Chức năng xóa loại sản phẩm 1.2 45](#_Toc75366862)

[Hình 60:Hiển thị dữ liệu dưới dạng table 46](#_Toc75366863)

# Mở đầu

Với sự phát triển nhảy vọt của công nghệ thông tin hiện nay,Internet ngày càng giữ 1 vai trò quan trọng trong các lĩnh vực khoa học kĩ thuật và đời sống. Dĩ nhiên các bạn cũng đã được nghe nhiều về internet.Nói 1 cách đơn giản thì Internet là 1 tập hợp máy tính kết nối với nhau, là 1 mạng máy tính toàn cầu mà bất kì ai cũng có thể kết nối bằng pc của họ.Với mạng Internet, tin học thật sự đã tạo nênmột cuộc cách mạng trao đổi thông tin trong mọi lĩnh vực văn hóa, xã hội, chính trị, kinh tế,. . .

Và với tình hình dịch bệnh trên thế giới phức tạp như hiện nay thì thương mai điện tử đang trở thành 1 lựa chọn tốt nhất đối với mọi doanh nghiệp lớn vừa và nhỏ để có thể buôn bán, trao đổi, quảng bá hình ảnh, sản phẩm của công ty mình đến với người dân.Có thể thấy thương mại điện tử quan trọng tới nỗi nếu doanh nghiệp muốn phát triển mạnh hơn thì phải đầu tư cho thương mại điện tử, nếu ko thì rất khó để phát triển.

Với lí do đó đưới sự hướng dẫn của giảng viên Tuấn Trần nhóm em đã chọn đề tài “**Xây dựng website bán** giày“ làm đề tài cho đồ án của tụi em.

Tuy nhiê mặc dù đã cố gắng hết sức nhưng sẽ không tránh được những sai sót, em rất mong nhận được sự thông cảm, những lời góp ý và chỉ bảo tận tình của các thầy cô và các bạn.

# Chương 1 :Giới thiệu về ngôn ngữ lập trình

## Tổng quan về Angular

Angular là một javascript framework do google phát triển để xây dựng các Single Page Application (SPA) bằng JavaScript , HTML và TypeScript . Angular cung cấp các tính năng tích hợp cho animation , http service và có các tính năng như auto-complete , navigation , toolbar , menus ,… Code được viết bằng TypeScript , biên dịch thành JavaScript và hiển thị tương tự trong trình duyệt.

Angular sở hữu rất nhiều ưu điểm. Đầu tiên phải kể đến là tính mô-đun. Mỗi chức năng cốt lõi khác nhau sẽ được chuyển sang các mô-đun khác nhau. Điều đó khiến việc thực hiện viết giao diện web sẽ nhanh chóng hơn, nhẹ nhàng hơn. Cũng chính điều này đã khiến Angular được sử dụng rất phổ biến với nhiều developer hiện nay.

## Ưu điểm -Nhược điểm

Ưu điểm:

* Khả năng ràng buộc dữ liệu ở cả hai chiều: Code ở cả JavaScript và HTML đều được đồng bộ hóa, giúp tiết kiệm thời gian lập trình.
* Mở rộng, phát triển tính năng của các file HTML nhờ sự hiện diện của các chỉ thị. Người dùng chỉ cần thêm tiền tố ng- trước thuộc tính HTML để có thể kích hoạt chỉ thị.
* Là công cụ của tương lai bởi hệ thống chức năng tuyệt vời. Cơ sở của người dùng được phát triển liên tục, đi kèm với đó là sự cập nhật thường xuyên tài liệu chuyên sâu.
* Angualr được phát triển từ Google nên sẽ được cộng đồng hỗ trợ đầy đủ khi gặp khó khăn trong việc code .
* Hỗ trợ phát triển các ứng dụng cho thiết bị di động, tương thích với nhiều thiết bị
* Tối ưu hoá tốt hơn SEO.
* Có thể sử dụng angular ci để tự động tạo các component.
* Angular giúp giảm tối đa kích thước và tăng tối đa hiệu suất của ứng dụng.

Nhược điểm:

* Phải học thêm Typescript.
* Không an toàn : bản chất của Angular là front end, mà front end thì vốn không được bảo mật bằng back end. Cho nên khi sử dụng API bạn nên xây dựng một hệ thống kiểm tra dữ liệu trả về tốt nhất.

## Kiến trúc của Angular

### Component:

* + Angular Component là một class nó được bổ trợ bởi @Component decorator. Component điều khiển một phần của UI.
  + Component có 4 phần như sau:
    - Import statement
    - Class
    - Template
    - Metadata

### Template

* + Component cần một view để hiển thị. Template định nghĩa view. Template chỉ là một tập con của HTML, nó chỉ cho Angular biết làm sao để hiển thị view. Nó là một trang HTML chuẩn sử dụng các thẻ H1, H2...Nó cũng sử dụng các lệnh của Angular như {}, []...

### Metadata

* + Metadata chỉ cho Angular làm sao để xử lý class. Chúng ta thêm Metadata vào class sử dụng class decorator. Khi chúng ta thêm @Component decorator vào class nó sẽ trở thành component.
  + Class decorator sử dụng đối tượng cấu hình, cung cấp thông tin Angular cần để tạo component. Ví dụ @Component directive đi với cấu hình như selector, templateURL, directive...

### Data Binding

* + Angular sử dụng Data Binding để lấy dữ liệu từ Component đến View. Nó được hoàn thành sử dụng cú phpas HTML đặc tả được biết đến là Template Syntax
  + Angular hỗ trợ 4 kiểu của Data binding:
  + Interpolation: Data được bind ra từ component sang view
  + Property Binding: Data bind ra từ component sang thuộc tính HTML trong view.
  + Event Binding: Các sự kiện DOM được bind ra từ view ra method của Component
  + Two-way binding/Model binding: Luồng dữ liệu cả hai chiều từ view sang component và ngược lại.

### Directive

* + Directive giúp chúng ta thao tác với View. Một directive là một class, chúng ta tạo ra sử dụng @Directive nó chứa metadata và logic thao tác trên DOM.
  + View được tạo ra bởi Angular sử dụng template metadata được định nghĩa trong Component. Các template là động và được chuyển đổi sử dụng directive.
  + Angular hỗ trợ 2 loại directive là Structural directive sẽ thay đổi cấu trúc của view. Còn loại kia là attribute directive sẽ chỉnh sửa style của view.

### Service

* + Service cung cấp dịch vụ cho các Components hoặc các service khác. Angular không có bất cứ đặc tả nào cho Service nó chỉ là các class có export method và chứa một số task.

### Dependency Injection

* + Dependency Injection là một phương thức khi tự động tạo một instance của một service được inject vào một component hoặc một service khác khi nó được yêu cầu.
  + Denpendency injection được dùng hầu hết khi inject service vào component hoặc service khác.

## Các phiên bản Angular

* 14/09/2016: Angular 2
* 23/03/2017: Angular 4 – Version 4
* 11/11/2017: Angular 5 – Version 5…
* 03/05/2018: Angular 6 – Version 6
* 18/10/2018: Angular 7 – Version 7
* 25/08/2019: Angular 8 – Version 8
* 06/02/2020: Angular 9.0
* 25/03/2020: Angular 9.1
* 08/04/2020: Angular 10

## Các tính năng cơ bản

* Controller : xử lý dữ liệu cho đối tượng $scope, từ đây bên views sẽ sử dụng các dữ liệu trong scope để hiển thị ra tương ứng.
* Data-binding : tự động đồng bộ dữ liệu giữa model và view
* Service : Nó được xem là singleton object có khả năng khởi tạo 1 lần duy nhất dành riêng cho mỗi ứng dụng và nó cung cấp các phương án dữ liệu có sẵn như: ($http,  $sce,  $rootElement, $controller, $document,  $httpBackend, $compile, $parse, $rootScope …..)
* Scope : là một trong những đối tượng có nhiệm vụ giao tiếp giữa controller và view của các ứng dụng.
* Filter : Việc lọc các tập hợp con từ bên trong  item ở các mảng và trả nhanh về các mảng mới.
* Directive : được sử dụng để tạo ra các thẻ HTML riêng nhằm mục đích phục vụ một số mục đích  riêng. AngularJS thường có những directive  sở hữu sẵn như ngBind, ngModel…
* Temple : Là một thành phần của view có khả năng hiển thị thông tin từ controller.
* Routing : là sự chuyển đổi giữa các action trong controller, qua lại ngay giữa các view.
* MVC & MVVM : Là mô hình thiết kế nhằm mục đích phân chia các ứng dụng có nhiều thành nhiều phần khác nhau (nó được gọi là Model, View và Controller) và mỗi phần thường sẽ sở hữu một nhiệm vụ nhất định. AngularJS thường không triển khai MVC dựa theo cách truyền thống mà chủ yếu gắn liền với Model-View-ViewModel hơn.
* Deep link : Liên kết sâu này cho phép lập trình viên mã hóa các trạng thái của ứng dụng bên trong URL để có thể bookmark với nhiều công cụ tìm kiếm khác. Hầu hết, các ứng dụng này đều có thể được phục hồi lại từ những địa chỉ URL với cùng trạng thái.
* Dependency Injection: AngularJS có sẵn dependency injection hỗ trợ  bạn tạo ra các ứng dụng có tiềm năng phát triển, dễ hiểu và kiểm tra.

## Các đặc trưng

* Được sử dụng để có thể phát triển dựa trên JavaScript.
* Angular có khả năng tạo các ứng dụng client-side dựa trên mô hình MVC.
* Angular sở hữu khả năng tương thích cao có thể tự động xử lý dễ dàng các mã Javascript sao cho phù hợp với các trình duyệt nhất.
* Khi có mã nguồn mở và miễn phí thì nó sẽ được sử dụng rộng rãi hơn.

## Ứng dụng

* CRUD Web Apps: đây là một trong những tính năng mà chính người sử dụng Angular ưu tiên sử dụng nhiều nhất.
* Mobile Apps: Lập trình viên có thể sử dụng Angular kết hợp cùng Phonegap để tạo ra một loại Mobile App  bạn có thể tạo một Mobile Web App.
* CSS3 Animations.
* Chrome Extensions: Đây là cách đơn giản nhất giúp lập trình viên tạo ra Chrome Extension lả việc sử dụng Yeoman- một Chrome Extension Generator.
* Testable JS Apps.
* Firebase Powered Apps.

## Tại sao nên sử dụng Angular ?

### Angular giúp nâng cao năng suất của các lập trình viên

* Việc phát triển Web đã có bước thay đổi đáng kể trong vài năm qua. Với phiên bản ECMAScript (ES) 2015 - chúng ta quen thuộc với cái tên ES6, với những class hay arrow function. Angular 2+ ứng dụng những tính năng mới này giúp việc code với Angular trở nên rõ ràng và dễ học hơn rất nhiều
* Thêm vào đó, với việc ứng dụng Typescript - một ngôn ngữ - hay là một bản nâng cấp đáng giá của Javascript, Angular kết hợp với Typescript, chúng ta có một công cụ tuyệt vời giúp xử lý các vấn đề hạn chế của JS như kiểm tra kiểu dữ liệu, refactor code an toàn hơn,... từ đó cũng hỗ trợ tốt hơn cho việc Debug cũng như giúp các Dev thực sự hiểu rõ mã nguồn của họ hơn.

### Cấu trúc phát triển rõ ràng

* Angular sử dụng class trong ES6 với một loạt các thuộc tính để xây dựng toàn bộ các cấu trúc chủ chốt, giả sử bạn muốn tạo một Angular component. Tạo một class và thêm vào các thuộc tính cần thiết. Hay bạn muốn tạo một Angular module -Hãy tạo một class và thêm vào đó các thuộc tính cần thiết. Về cơ bản sẽ là như vậy, Angular cung cấp một cấu trúc rõ ràng để xây dựng từng tính năng cho ứng dụng của bạn
* Các dependency mạnh mẽ được sử dụng trong ứng dụng khi cần thiết, và khi cần tích hợp bất kì dependency nào, như HTTP hay Router, chúng ta chỉ cần thêm nó vào bên trong constructor của class
* Mô hình MVVM cũng giúp Angular chiếm lợi thế trong xây dựng ứng dụng client-side, thường ta sẽ có 3 điều cần quan tâm chính: đó là giao diện người dùng, mã nguồn điều khiển giao diện và mô hình dữ liệu (data) cho giao diện. Angular với MVVM phân biệt hoàn toàn rõ ràng các yếu tố trên nhờ mô hình MVVM

### Extensive Biding

* Rất nhiều ứng dụng Web làm việc với dữ liệu (data). App sẽ truy xuất dữ liệu từ Server và hiển thị dữ liệu đó tới người dùng trên view, sử dụng template. Và các tương tác của người dùng sẽ được khiến dữ liệu thay đổi, được view ghi nhận và lưu lại trên server. Data Binding trong Angular giúp bạn thực thi tiến trình trên rất dễ dàng. Đơn thuần từ việc ràng buộc thành phần HTML trong template với các thuộc tính trong class và dữ liệu sẽ tự động xuất hiện trên màn hình. Với các tương tác của người dùng đòi hỏi thay đổi dữ liệu, Angular sử dụng phương pháp two-way binding. Bất kỳ thay đổi dữ liệu đến từ view sẽ tự động cập nhật thuộc tính "model" bên trong class
* Thêm vào đó, Angular cũng hỗ trợ property binding - cho phép chúng ta điều khiển DOM bằng cách ràng buộc thuộc tính HTML với thuộc tính của component class, data sẽ tự động xuất hiện bên trong view. Ví dụ, chúng ta ràng buộc thuộc tính hidden đối với một thẻ img với thuộc tính hide Img bên trong class. Khi thuộc tính hide Img nhận giá trị true, image sẽ tự động hidden và ngược lại khi hide Img nhận giá trị false, thẻ img sẽ tự động hiển thị tới người dùng
* Cuối cùng, Angular hỗ trợ event binding, có nghĩa là chúng ta có thể xử lí bất kì event nào từ phía view, như HTML event. Về cơ bản chúng ta sẽ gắt event với một method bên trong class. Mỗi khi event xuất hiện, method tương ứng sẽ được thực thi. Extensive binding giúp quá trình hiển thị dữ liệu, điều khiển DOM, thực thi các event một cách trơn tru và dễ dàng

### Hỗ trợ đầy đủ tính năng điều hướng

* Angular cung cấp đầy đủ tính năng cho việc này, chúng ta định nghĩa các đường dẫn (route) cho mỗi page view của ứng dụng. Và chúng ta sẽ kích hoạt route dựa trên tương tác của người dùng (user). Chúng ta có thể truyền thêm dữ liệu vào các route, giúp view hiển thị nội dung một cách dynamic, có thể bảo vệ route để người dùng chỉ có thể truy cập sau khi đã đăng nhập hoặc có quyền truy cập, có thể ngăn chặn việc người dùng ngay lập tức rời một trang khi các thao tác còn dang dở cho đến khi họ thực sự xác nhận việc rời đi hoặc lưu lại tiến trình sử dụng,...Angular đồng thời cũng hỗ trợ child-route cho việc điều hướng bên trong một route. Việc điều hướng giữa các view bên trong ứng dụng Angular thực sự rất linh hoạt và mạnh mẽ

### Giảm thiểu tối đa kích thước và tăng tối đa hiệu suất của ứng dụng

* Kích thước và hiệu năng có mối liên quan mật thiết khi chúng ta làm việc trên nền tảng Web. Một component nhỏ hơn sẽ giúp nâng cao hiệu suất khởi động - giảm cả thời gian download cũng như thời gian cũng như thời gian compile trên trình duyệt. Giảm kích thước component và giúp tăng hiệu suất là một ưu điểm cũng như mục tiêu mà Angular mong muốn mang đến cho các lập trình viên
* Giảm kích thước ứng dụng có thể thực hiện bằng nhiều cách. Đầu tiên chúng ta có thể giảm tối đa kích thước của từng component tới mức tối thiểu có thể. Tiếp theo các component sẽ được sắp xếp bên trong Angular Module bằng 1 cách đề cho các nhóm logic có liên quan đến nhau sẽ được download cùng với nhau. Và bước thứ ba, lazy loading bên trong các route sẽ chỉ download những module cần thiết cho việc hiển thị nội dung cần thiết tới người dùng, và sẽ không bao giờ download những nội dung không cần thiết

### Document và Community

* Document cho Angular 2+ - angular.io - rất đầy đủ và chi tiết, bao hàm giới thiệu cơ bản giúp bạn làm quen nhanh chóng với Angular, giới thiệu chi tiết, từ cơ bản đến nâng cao các API của Angular, cũng như có hẳn một Tutorial Basic được xây dựng nên bởi Angular team, cung cấp cho bạn nhanh chóng nắm bắt các thuộc tính cơ bản của Framework

# Chương 2 - Khảo sát và phân tích hệ thống

## Phân tích yêu cầu đề

### Tên đề tài:

Xây dựng website bán giày

### Chức năng

Đây là 1 website giới thiệu sản phẩm của cửa hàng, công ty đến người tiêu dùng với các chi tiết mặt hang với giá cả chính xác. Có các chức năng như sau:

* Cho phép cập nhật hàng vào cơ sở dữ liệu
* Hiển thị hàng hóa để khách chọn mua
* Hiên thị thông tin chi tiết sản phẩm
* Quản lí đơn hang
* Cập nhật hàng hóa, loại hàng, tin tức, nhà sản xuất, …

### Yêu cầu đặt ra

Thứ nhất: Phần khách hàng

* Hiển thị các mặt hàng tiêu biểu để khách hàng chọn mua
* Khách hàng xem được thông tin mới, khuyến mãi trên web
* Sau khi khách hàng chọn và đặt hàng thì phải hiện lên đơn hàng để khách hàng có thể nhập thông tin mua hàng và xem hóa đơn mua hàng

Thứ hai: Phần admin

* Chức năng cập nhật, thêm, xóa, sửa mọi mặt hàng, thông tin, nhà sản xuất, tin tức.
* Tiếp nhận đơn đặt hàng của khách hàng, hiển thị đơn đặt hàng.

## Phân tích hệ thống

Text

Description automatically generated

Hình :Useacse

## Đặc tả usecase

### Đặc tả usecase xem thông tin

* Người sử dụng: Admin, khách hàng
* Mô tả: cho phép admin và khách hàng xem thông tin về các sản phẩm có trong cửa hàng.
* Luông sự kiện chính:

1. Admin và khách hàng chọn sản phẩm cần xem
2. Form xem thông tin chi tiết sản phẩm sẽ xuất hiện, hệ thống sẽ hiển thị thông tin sản phẩm mà admin hoặc khách hàng đã chọn
3. Admin và khách hàng xem thông tin chi tiết về sản phẩm đã được hiển thị
4. Usecase kết thúc

### Đặc tả usecase thêm xóa sửa sản phẩm

* Người sử dụng:admin
* Mô tả: usecase cho phép admin xem danh sách sản phẩm và thêm xóa sửa sản phẩm
* Luồng sự kiện chính:

1. Admin bấm vào xem danh sách đơn hàng
2. Chọn sản phẩm muốn thêm xóa hoặc sửa
   1. Muốn thêm hoặc sửa thì cần nhập đầy đủ thông tin muốn thêm hoặc sửa sản phẩm
   2. Muốn xóa thì xác nhận muốn xóa sản phẩm
3. Usecase kết thúc

# Chương 3 Thiết kế giao diện và cơ sở dữ liệu

## Một số hình ảnh của website

### Giao diện trang chủ

Graphical user interface, website

Description automatically generated

Hình : Giao diện trang chủ 1.1

Website

Description automatically generated with low confidence

Hình :Giao diện trang chủ 1.2

A picture containing text, cat, wall, screenshot

Description automatically generated

Hình :Giao diện trang chủ 1.3

### Giao diện trang sản phẩm

Graphical user interface, application

Description automatically generated

Hình : Giao diện trang sản phẩm 1.1

Graphical user interface, website

Description automatically generated

Hình :Giao diện trang sản phẩm 1.2

### Giao diện trang chi tiết sản phẩm

Graphical user interface, application

Description automatically generated

Hình :Giao diện chi tiết sản phẩm

### Giao diện trang quản lý

#### Giao diện dashboard

Graphical user interface, chart, application

Description automatically generated

Hình :Giao diện Dashboard

#### Giao diện quản lý sản phẩm

Graphical user interface, application

Description automatically generated

Hình :Giao diện quản lý sản phẩm

Graphical user interface, application

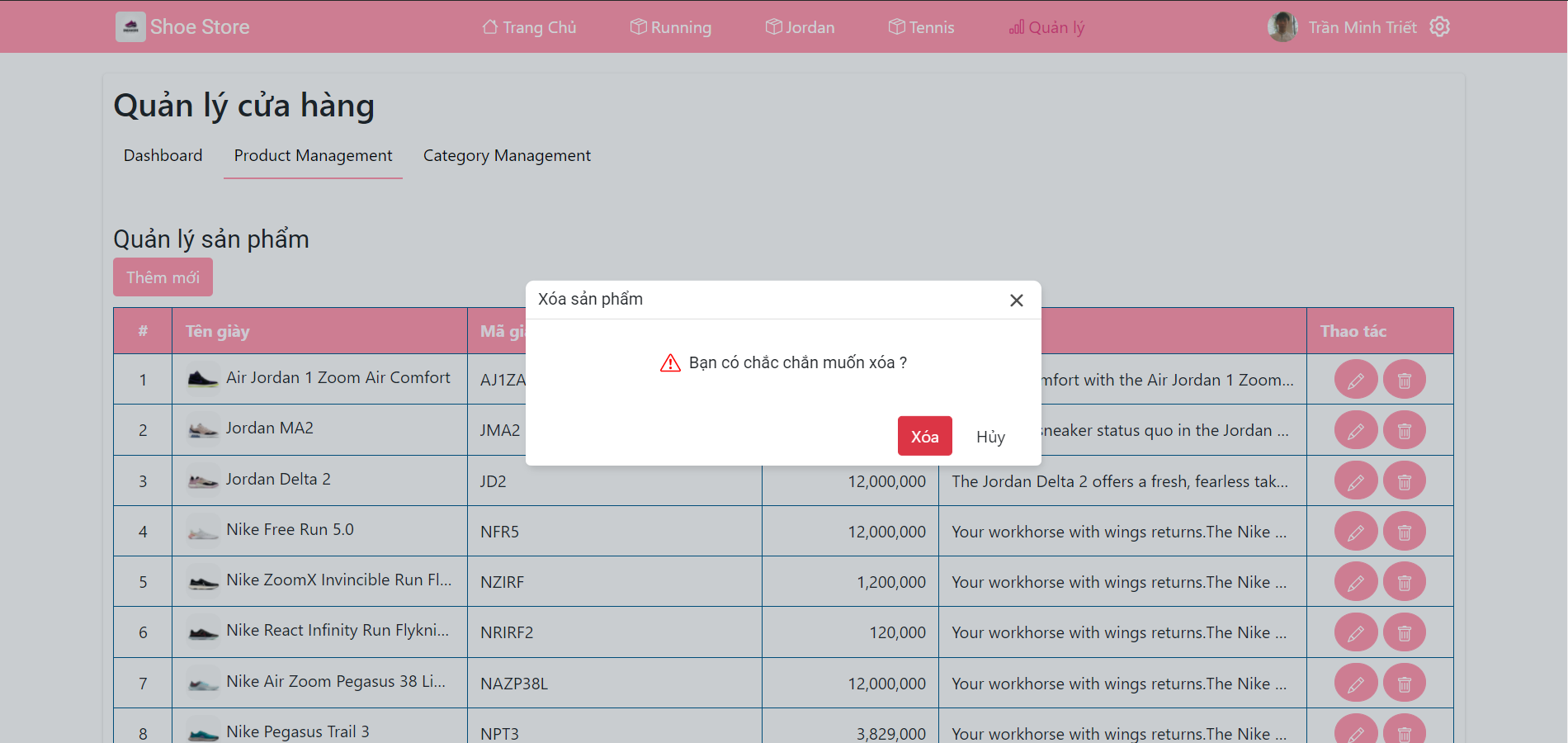
Description automatically generated

Hình :Giao diện thêm sản phẩm

Graphical user interface, email, website

Description automatically generated

Hình :Giao diện chỉnh sửa sản phẩm



Hình :Giao diện xóa sản phẩm

#### Giao diện quản lý loại sản phẩm

Graphical user interface, application

Description automatically generated

Hình :Giao diện quản lý loại sản phẩm

Graphical user interface, application

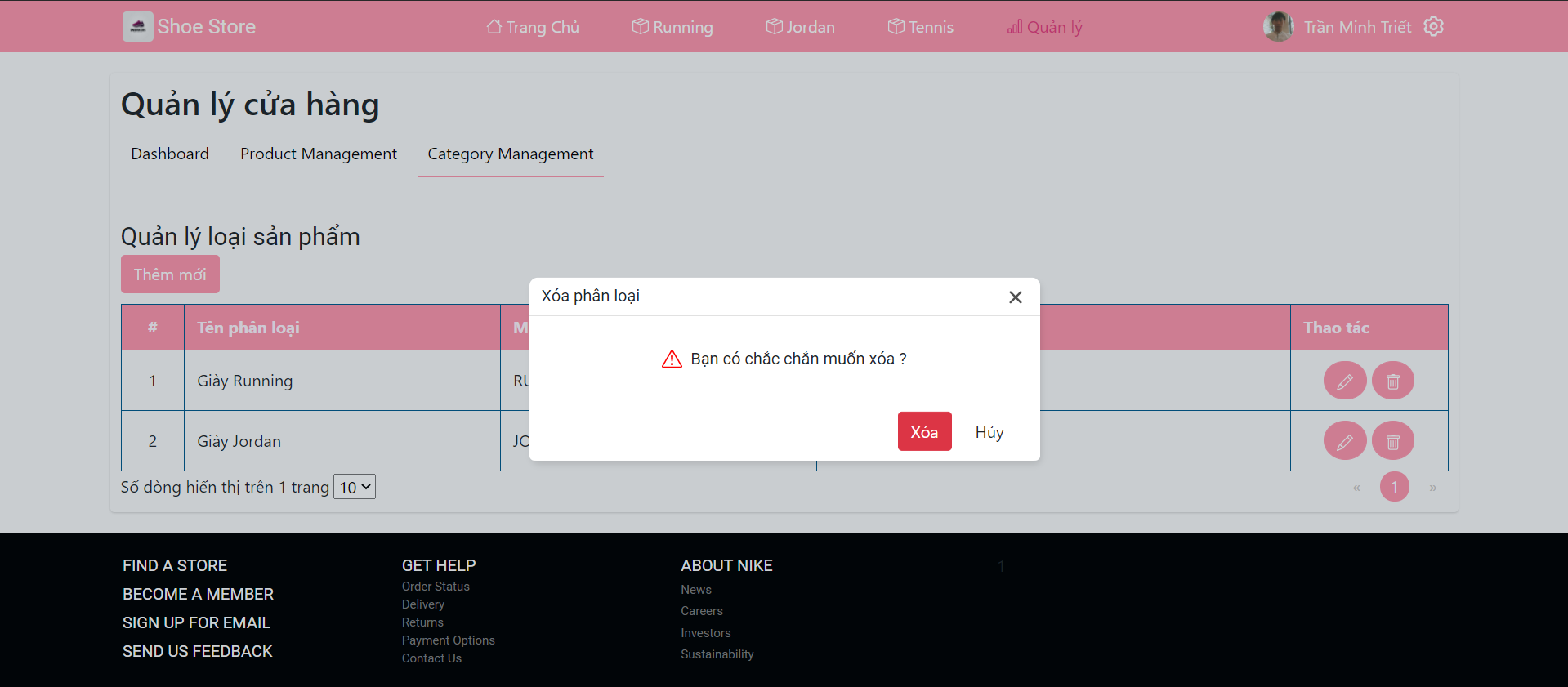
Description automatically generated

Hình :Giao diện thêm loại sản phẩm

Graphical user interface, application

Description automatically generated

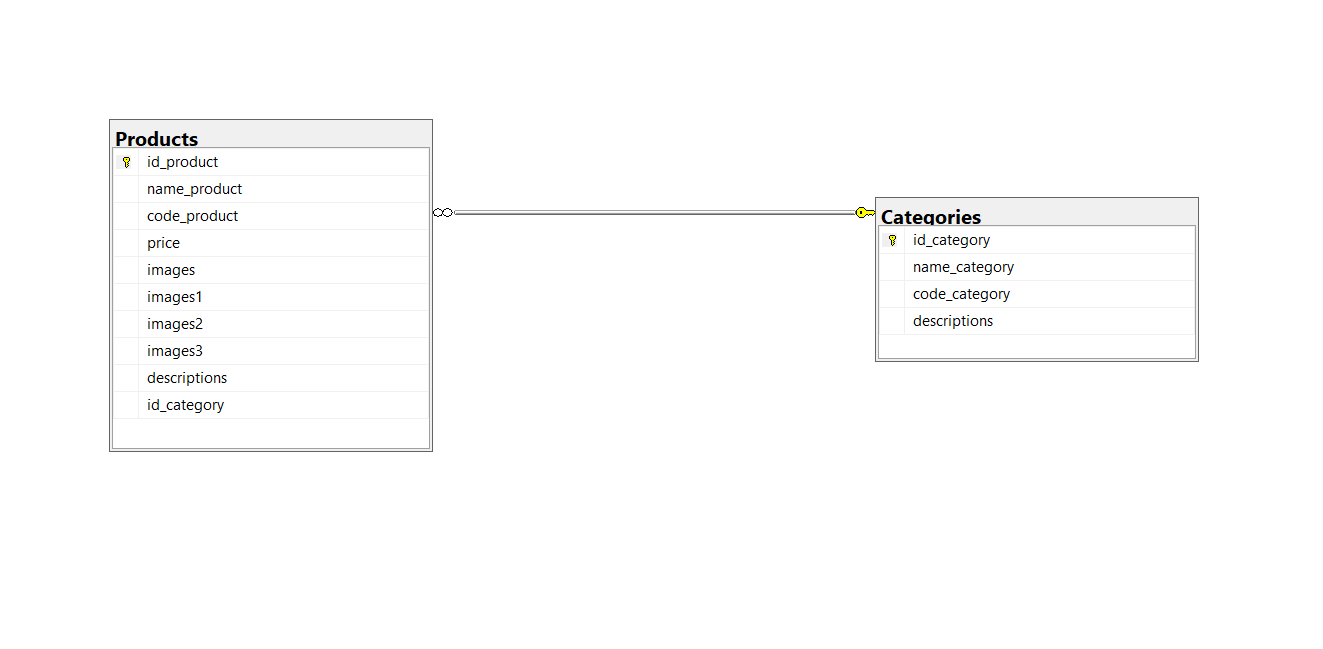
Hình :Giao diện chỉnh sửa loại sản phẩm



Hình :Giao diện xóa loại sản phẩm

## Cơ sở dữ liệu

* Products: id\_product,name\_product,code\_product,price,images,images1, images2, images3,descriptions,id\_category
* Categories: id\_category,name\_category,code\_category,descriptions



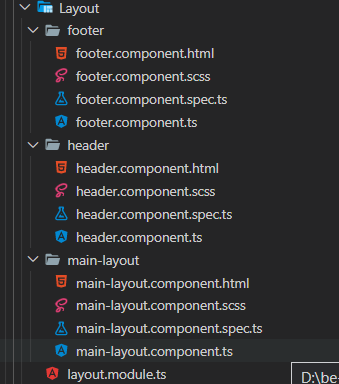
Hình :Cơ sở dự liệu

# Chương 4: Chức năng

## Cấu trúc thư mục và các tập tin cần thiết

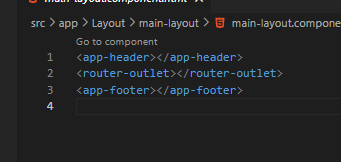
### Layout

* Chứa các component như footer, header và main-layout(chứa cả 2 footer và header).
* Layout sẽ có một module riêng để import vào app.modules.ts.



Hình :Layout Folder 1.1

* Chỉ cần khai báo router cái component là child của Main-layout component tự động những component đó sẽ tự động nhận header và footer.



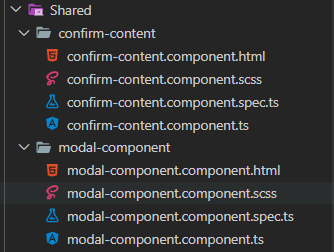
Hình :Layout Folder 1.2



Hình :Layout Folder 1.3

### Shared

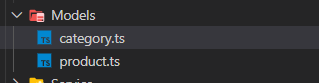
* Chứa các component dùng chung như modal component và confirm content



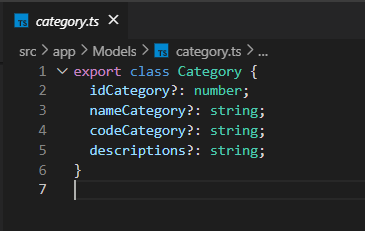
Hình :Shared Folder

### Models

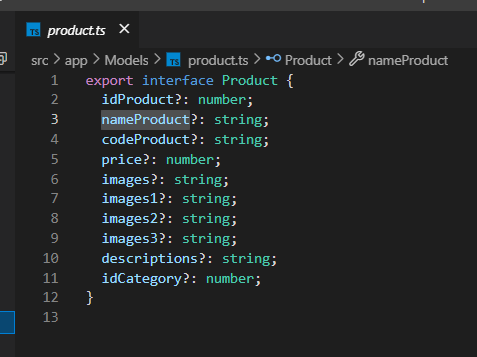
* Chứa class cần thiết cho project.



Hình :Models Folder 1.1



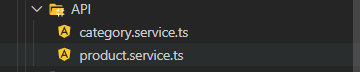
Hình :Models Folder 1.2



Hình :Models Folder 1.3

### API

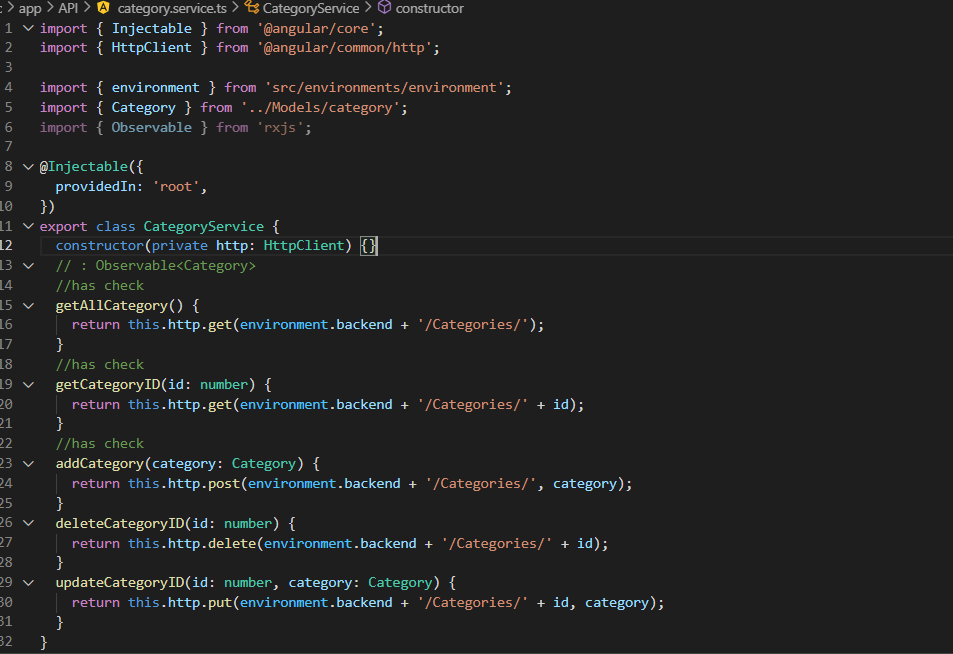
* Chứa service, http request.



Hình :API Folder

#### Category service

* Khởi tạo các http request từ phía back-end:
* getAllCategory(): Lấy tất cả dữ liệu về category (http get).
* getCategory(): Lấy dữ liệu về category dựa theo ID (http get).
* addCategory(): Thêm mới category (http post).
* deleteCategory(): Xóa category (http delete).
* updateCategoryID(): Chỉnh sửa category dựa theo ID (http put).



Hình : Category Service

#### Product Service

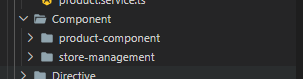
* Khởi tạo các http request từ phía back-end:
* getAllProduct(): Lấy tất cả dữ liệu về product (http get).
* getAllProductOnCate(): Lấy tất cả dữ liệu về product dựa theo category (http get).
* getProductID() : Lấy dữ liệu về product dựa theo ID (http get).
* addProduct(): Thêm mới product (http post).
* deleteProductID(): Xóa product (http delete).
* updateProductID(): Chỉnh sửa product dựa theo ID (http put).
* updateImageProduct(): Thêm hình ảnh đại diện cho sản phẩm dựa theo ID (http put).
* updateImageProduct1(): Thêm hình ảnh phụ cho sản phẩm dựa theo ID (http put).
* updateImageProduct2(): Thêm hình ảnh phụ cho sản phẩm dựa theo ID (http put).
* updateImageProduct3(): Thêm hình ảnh phụ cho sản phẩm dựa theo ID (http put).



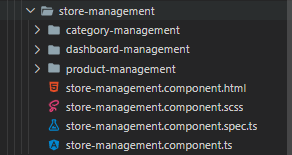
Hình : Product Service

### Component

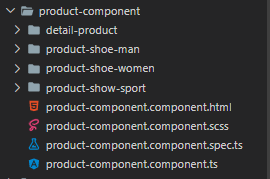
* Gồm 2 component chính là product-component (giao diện sản phẩm) và store-management (giao diện quản lý).
* Product-component chứa các component con như : detail-component, product-shoe,…
* Store-mangement chứa các component con như: dashboard-management, product-management, category-management, modal-create-product, modal-create-category.



Hình :Component Folder



Hình :Store-management component



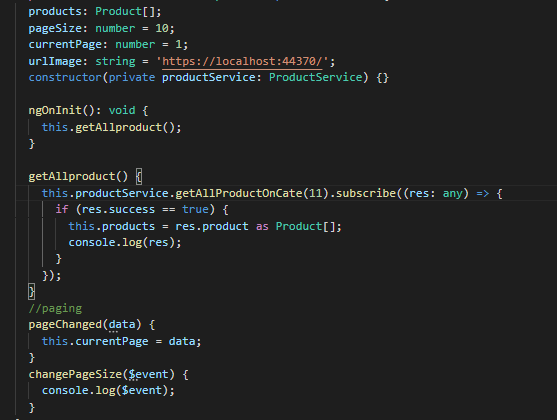
Hình : Product-component

## Chức năng

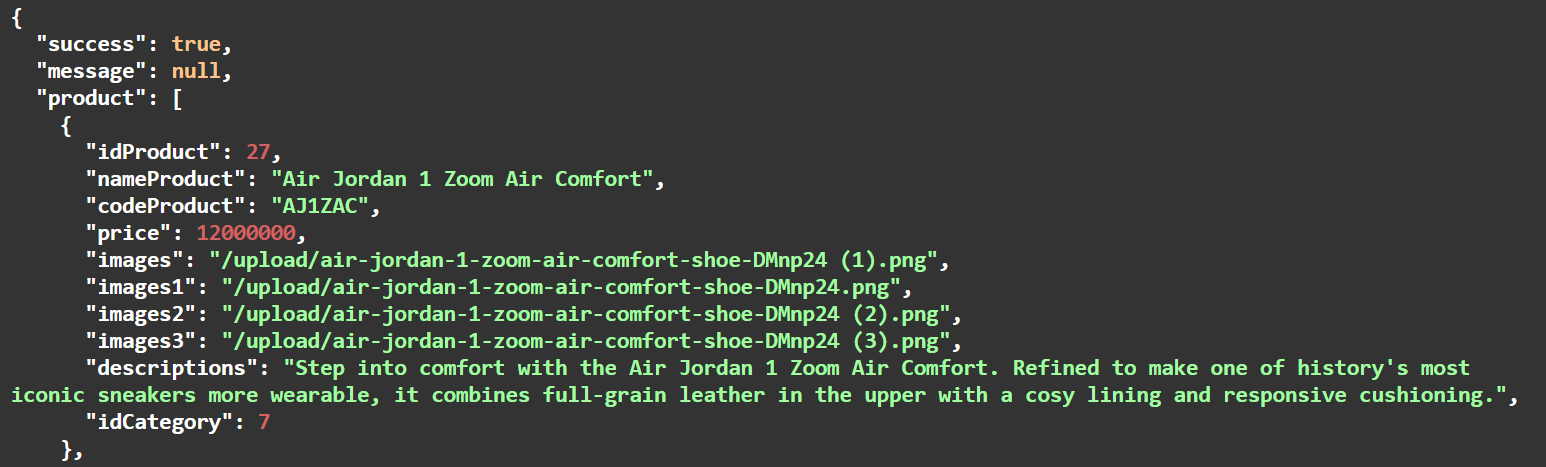
### Người dùng

#### Xem sản phẩm

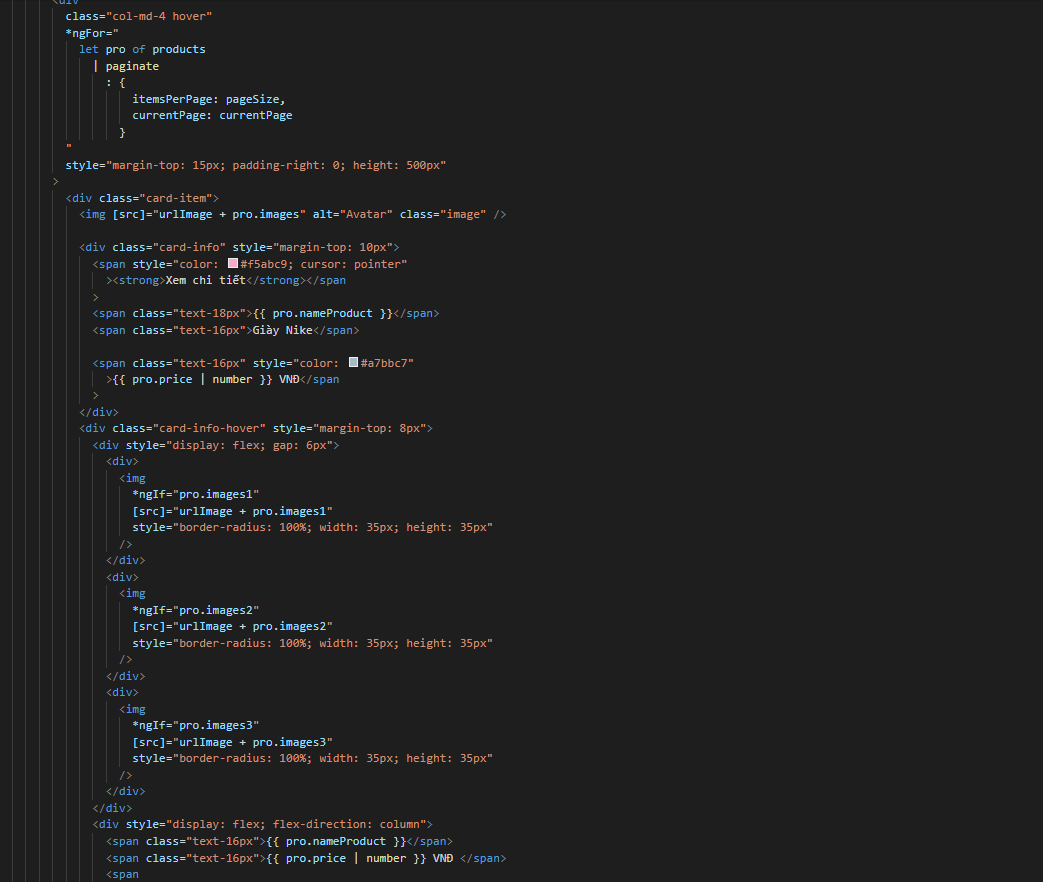
* Khởi tạo array dựa theo Product
* Gọi tới hàm gettAllProductOnCate() trong service để lấy dữ liệu.
* Khi subscribe , API sẽ trả về một response data theo dạng json.
* Nếu “success” là true , thì sẽ gán dữ liệu.
* Sử dụng structural directives \*ngFor để hiển thị dữ liệu từ array.



Hình :Chức năng xem sản phẩm 1



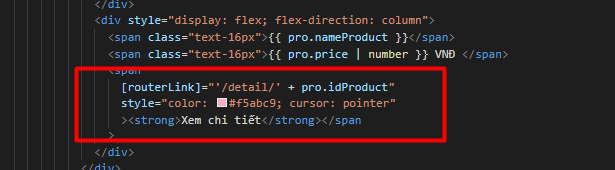
Hình :Data Json



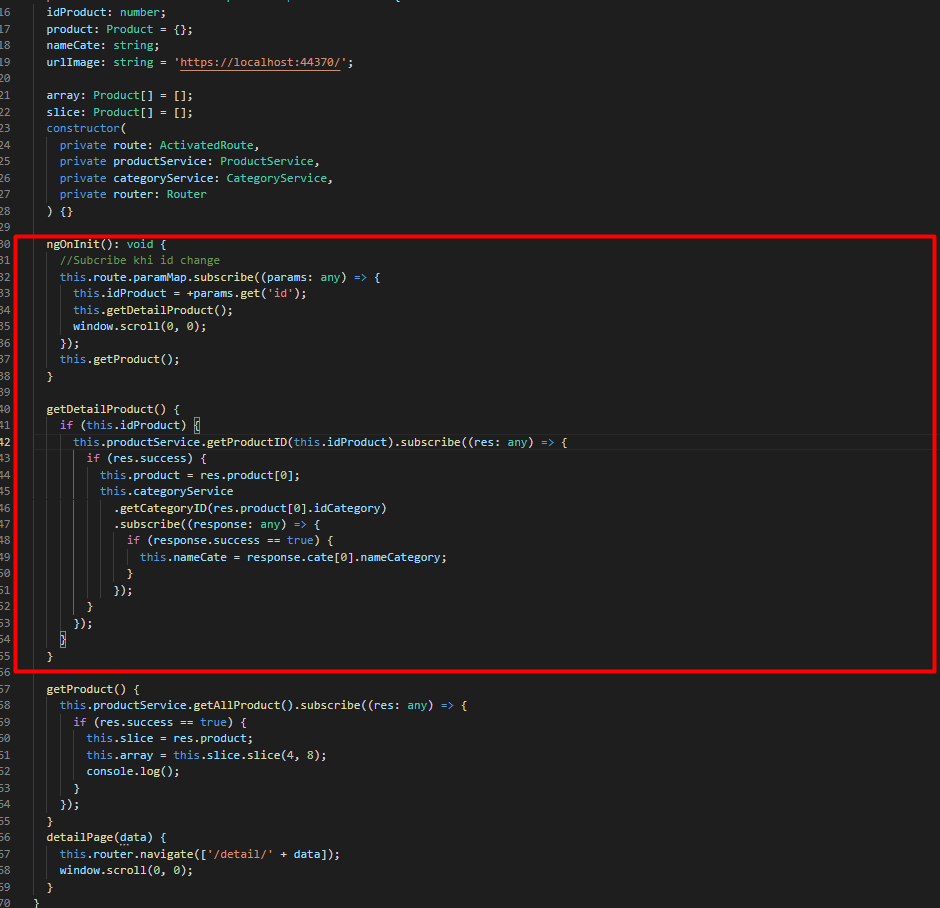
Hình :Chức năng xrm sản phẩm 1.1

#### Xem thông tin chi tiết sản sản phẩm

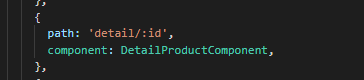
* Tạo component detail, cài đặt router với id
* Tọa routerlink chuyển hướng đến detail-component kèm với ID product.
* Subscribe router để nhận biết khi nào ID product thay đổi.
* Từ đó , gọi tới hàm getDetailProduct().
* Hàm getDetailProduct() sẽ gọi tới hàm geProductID() trong Product Service, từ đó chúng ta sẽ truyền id product vào và subscribe để lấy dữ liệu từ đó truyền dữ liệu qua html.



Hình :Chức năng xem chi tiết sản phẩm 1



Hình : Chức năng xem chi tiết sản phẩm 1.1

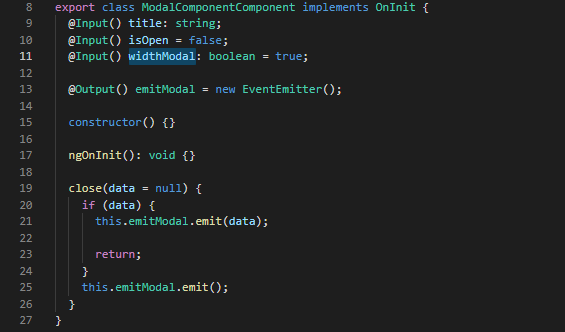


Hình : Chức năng xem chi tiết sản phẩm 1.2

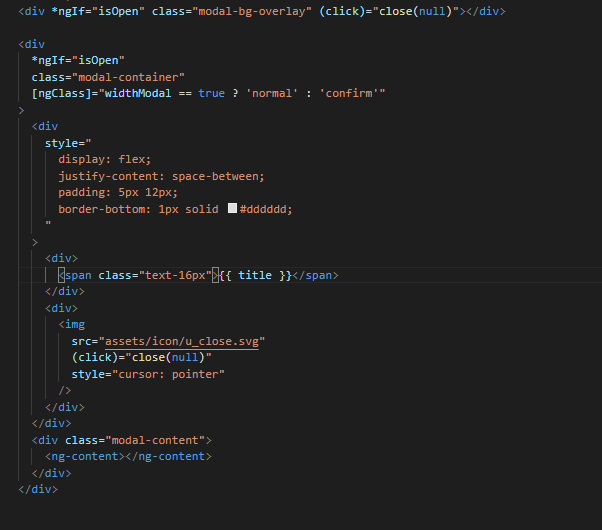


### Admin

* Tạo modal dùng chung cho việc thêm, xóa, sửa.
* Modal sẽ được truyền vào title, isOpen, widthModal
* Modal sẽ được bật khi isOpen bằng true.
* Với widthModal chúng ta dùng [ngClass] có thể thay đổi width dựa theo true và false.
* Trong html, sử dụng tag <ng-content> để thêm vào nội dụng mong muốn.
* Sử dụng Custom EventEmitter để có thể đóng mở modal.



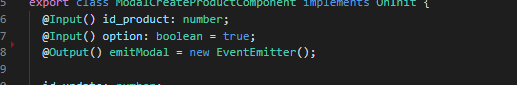
Hình :Modal Component (Typescript)



Hình :Modal Component (HTML)

#### Quản lý sản phẩm

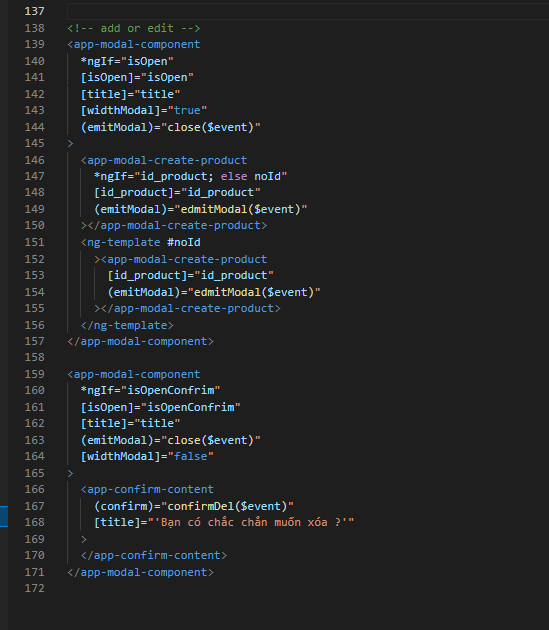
* Tạo một modal-create-component dùng chung cho 2 trường hợp thêm và sửa.
* Gọi modal
* Truyền vào id product và option.
* Ouput dung để emit data về để có thể đóng modal.
* Sử dụng reactive form để thêm và chỉnh sửa sản phẩm.
* Dựa vào ID product , chúng ta có thể set title cho button và cả chức năng cho button.



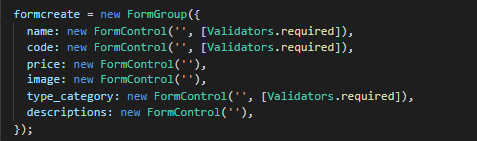
Hình :Chức năng quản lý sản phẩm 1.1



Hình : Chức năng quản lý sản phẩm 1.2

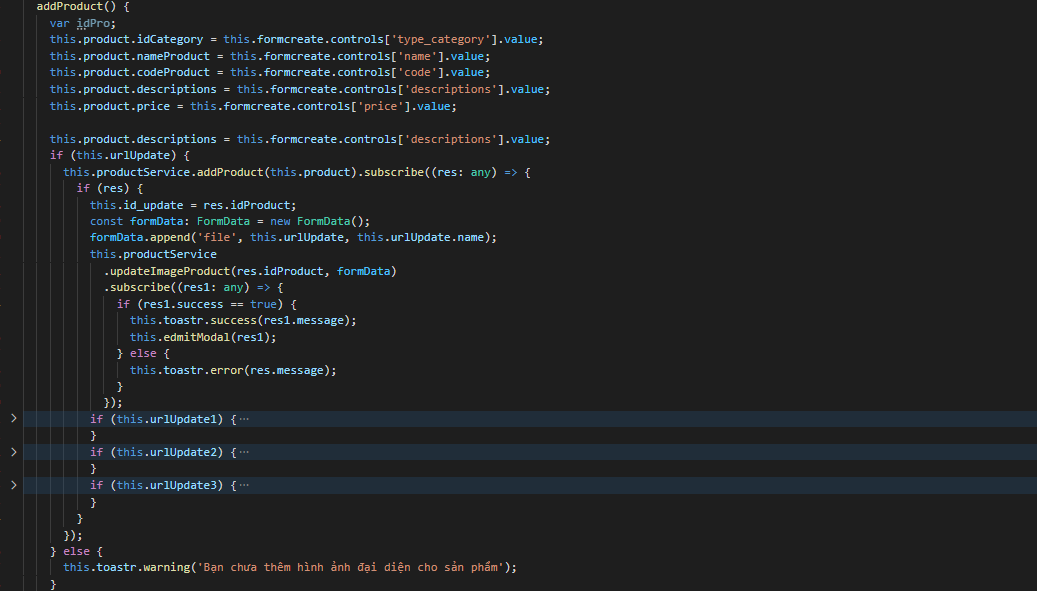


Hình : Chức năng quản lý sản phẩm 1.3



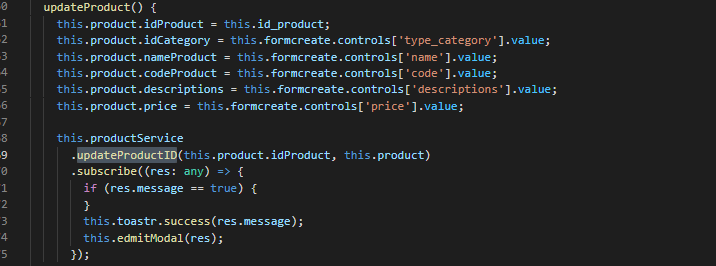
Hình : Chức năng quản lý sản phẩm 1.4

* Thêm
* Bật modal-component bên trong có thẻ <ng-content> để có thể thêm modal-create-component.
* Nhập dữ liệu cho forms.
* Sử dụng các validator cho reactive form.
* Sử dụng [ngClass] để chỉnh sửa màu input label khi người dùng nhập sai hoặc thiếu.
* Gán dữ liệu từ form vào object product
* Sử dụng các hàm addProduct(), updateImageProduct() trong service để thêm thông tin và ảnh dựa theo ID product.

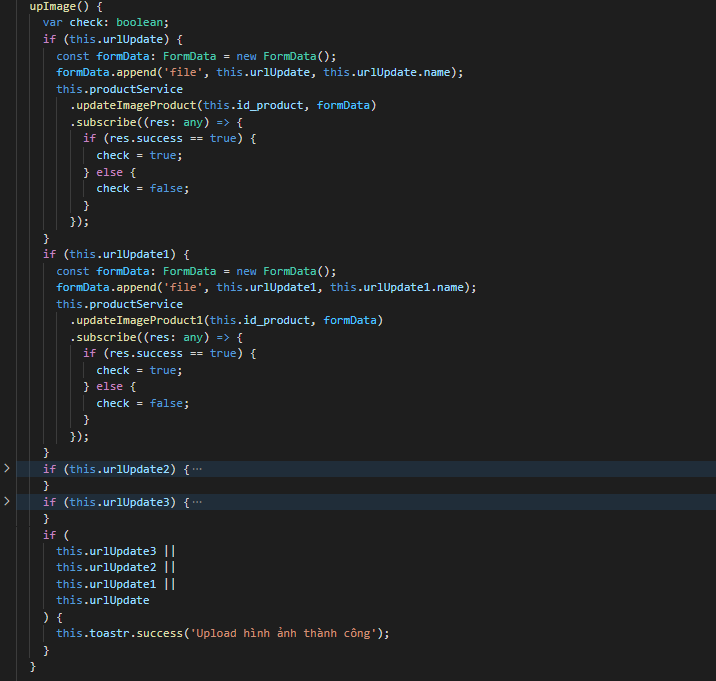


Hình :Chức năng thêm mới sản phẩm

* Sửa
* Gọi tới hàm getProductID() để get dữ liệu khi có id truyền vào.
* Gán dữ liệu vào reactive forms.
* Sử dụng các hàm updateImageProduct() trong service thể thêm ảnh dựa theo ID product.
* Gán dữ liệu ngược trở lại từ form vào object product .
* Sử dụng các hàm updateProductID(), updateImageProduct() trong service để chỉnh sửa thông tin và ảnh dựa theo ID product.

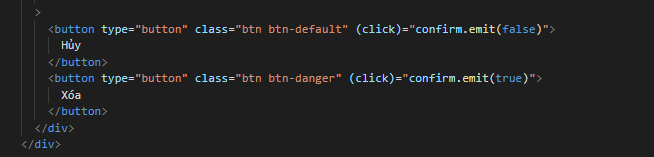


Hình :Chức năng chỉnh sửa sản phẩm



Hình :Chức năng update hình ảnh sản phẩm

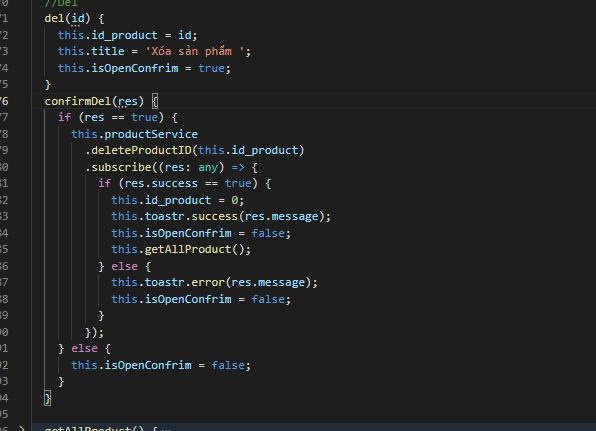
* Xóa
* Xử dụng confirm-content.
* Output confirm-content sẽ trả về true false từ đó sẽ xóa hoặc hủy.



Hình : Chức năng xóa sản phẩm 1.1



Hình : Chức năng xóa sản phẩm 1.2



Hình :Chức năng chỉnh sửa 1.3

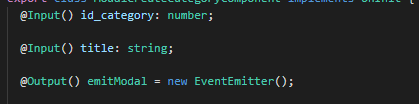
* Hiển thị theo dạng table



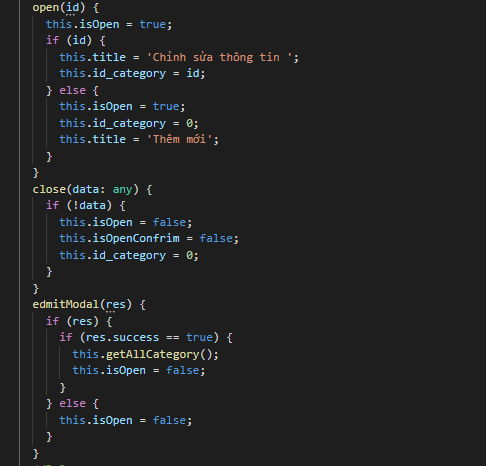
Hình :Hiển thị dữ liệu dưới dạng table

#### Quản lý loại sản phẩm

* Tạo một modal-create-component dùng chung cho 2 trường hợp thêm và sửa.
* Gọi modal
* Truyền vào id product và option.
* Ouput dung để emit data về để có thể đóng modal.
* Sử dụng reactive form để thêm và chỉnh sửa loại sản phẩm.
* Dựa vào ID product , chúng ta có thể set title cho button và cả chức năng cho button.



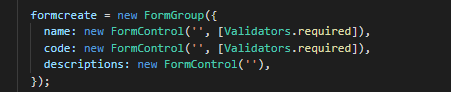
Hình :Chức năng quản lý loại sản phẩm 1.1



Hình : Chức năng quản lý loại sản phẩm 1.2

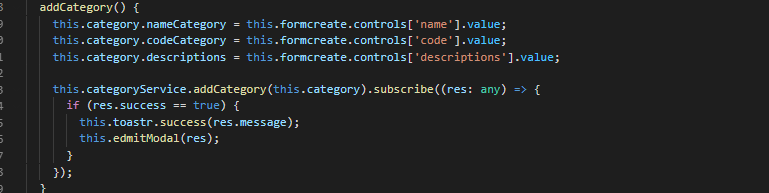


Hình : Chức năng quản lý loại sản phẩm 1.3



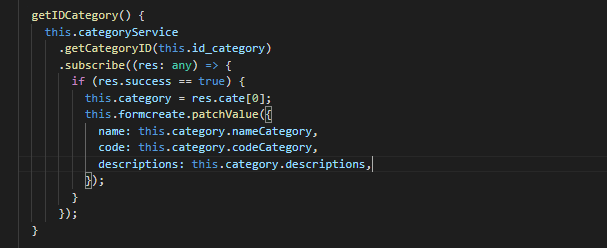
Hình :Reactive form

* Thêm
* Bật modal-component bên trong có thẻ <ng-content> để có thể thêm modal-create-component.
* Nhập dữ liệu cho forms.
* Sử dụng các validator cho reactive form.
* Sử dụng [ngClass] để chỉnh sửa màu input label khi người dùng nhập sai hoặc thiếu.
* Gán dữ liệu từ form vào object product
* Sử dụng hàm addCategory() để thêm loại sản phẩm.

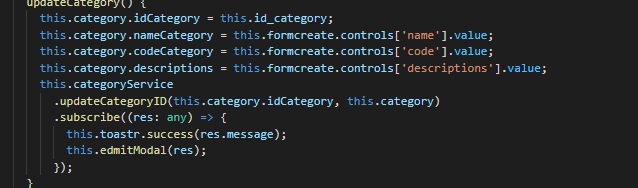


Hình :Chức năng thêm mới loại sản phẩm

* Sửa
* Gọi tới hàm getIDCategory () để get dữ liệu khi có id truyền vào.
* Gán dữ liệu vào reactive forms.
* Gán dữ liệu ngược trở lại từ form vào object product .
* Sử dụng các hàm updateCategory(), updateImageProduct() để chỉnh sửa thông tin.

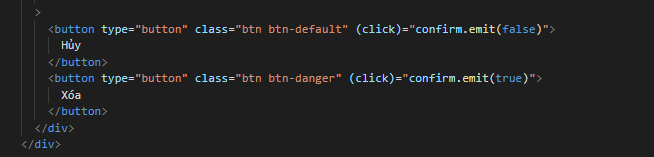


Hình :Chức năng chỉnh sửa loại sản phẩm 1.1



Hình :Chức năng chỉnh sửa loại sản phẩm 1.2

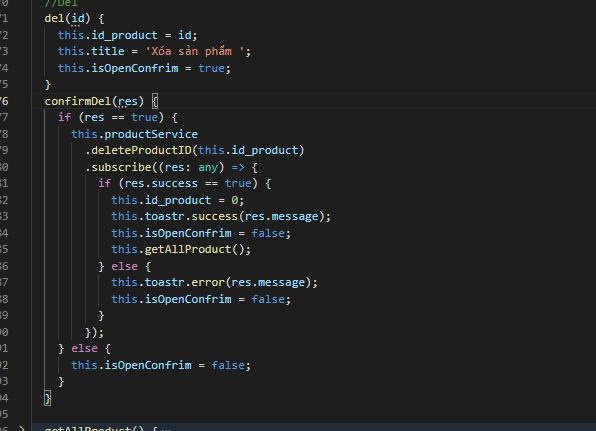
* Xóa
* Xử dụng confirm-content.
* Output confirm-content sẽ trả về true false từ đó sẽ xóa hoặc hủy.



Hình :Chức năng xóa loại sản phẩm 1.1

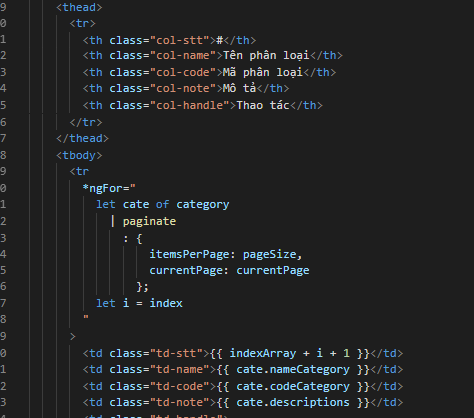


Hình :Chức năng xóa loại sản phẩm 1.2



Hình :Chức năng xóa loại sản phẩm 1.2

* Hiển thị theo dạng table



Hình :Hiển thị dữ liệu dưới dạng table

# CHƯƠNG III: TỔNG KẾT

## Những việc đã làm được

* Áp dụng các kiến thức Angular vào dự án
* Thiết kế được giao diện website
* Thực hiện Thêm/ Xóa/ Sửa bằng Angular
* Xem thông tin và hình ảnh sản phẩm

## Những việc chưa làm được

* + Chưa tạo được phân quyền giữa Admin và Khách hàng
  + Chưa thêm được sản phẩm vào giỏ hàng
  + Chưa có chức năng giỏ hàng và thanh toán
  + Chưa có tính năng tìm kiếm sản phẩm

## Nhận xét quá trình triển khai

* Dự án hoàn thành đúng tiến độ
* Chưa hoàn hành được mục tiêu đề ra

## Phân công công việc

|  |  |  |
| --- | --- | --- |
| STT | Tên | Công việc thực hiện |
| 1 | Trần Minh Triết | * Viết API , Service * Xử lý front-end và Back-end |
| 2 | Lê Ngọc Bảo Toàn | * Xử lý front-end * Tạo các file , component cần thiết |
| 3 | Nguyễn Bá Duy Việt | * Xử lý front-end |

## Tài liệu tham khảo

* Angular: <https://angular.io/start>
* .NET CORE Web API: <https://docs.microsoft.com/en-us/aspnet/core/tutorials/first-web-api?view=aspnetcore-5.0&tabs=visual-studio>