|  |  |  |
| --- | --- | --- |
| Macintosh HD:Users:hoangnguyen:Documents:TDC:fitlog_blue.png | TRƯỜNG CAO ĐẲNG CÔNG NGHỆ THỦ ĐỨC  **Khoa Công Nghệ Thông Tin** | Macintosh HD:Users:hoangnguyen:Documents:TDC:logoTDC_blue.png |
|  |  |  |

**Chuyên đề phát triển web 2** | HKI – [2019 – 2020]

PRJ – REPORT

**BÁO CÁO TÌM HIỂU CÔNG NGHỆ ANGULAR JS**

**Nhóm J:**

* **Thái Quang Hưng –** 17211TT3105 (Nhóm trưởng)
* **Hà Xuân Phi –** 17211TT0607 (Nhóm phó)
* **Trần Quốc Bữu –** 17211TT3375
* **Võ Đình Chinh –** 17211TT3587
* **Trần Bình Văn –** 17211TT0053

**MỤC LỤC**

[1. Giới thiệu 1](#_Toc24156157)

[2. Các kỹ thuật của AngularJS 1](#_Toc24156158)

[2.1. AngularJS Directives 1](#_Toc24156159)

[2.1.1 Thuộc tính AngularJS có bắt đầu đều là ng- 1](#_Toc24156160)

[2.1.2 Data binding1](#_Toc24156161)

[2.1.4 Tạo mới Directives 2](#_Toc24156162)

[2.2. Biểu thức của AngularJS 3](#_Toc24156163)

[2.3. Biều thức số trong AngularJS 3](#_Toc24156164)

[2.4. Biểu thức chuỗi AngularJS 3](#_Toc24156165)

[2.5. Đối tượng trong AngularJS 4](#_Toc24156166)

[2.6. Mảng trong AngularJS 4](#_Toc24156167)

[2.7. Modules AngularJS 4](#_Toc24156168)

[2.8. Modules AngularJS 5](#_Toc24156169)

[2.9 AngularJS Scopes 5](#_Toc24156170)

[2.10 AngularJS Filters 6](#_Toc24156171)

[2.11 AngularJS $http 7](#_Toc24156172)

[2.12 Angular JSON 7](#_Toc24156173)

[2.13 AngularJS DOM 7](#_Toc24156174)

[2.14 AngularJS Events 8](#_Toc24156175)

[3. Báo cáo tích hợp Angular JS vào các module. 9](#_Toc24156176)

[**3.1.** **Khởi tạo ứng dụng AngularJS.** 9](#_Toc24156177)

[**3.2.** **Module 04.** 10](#_Toc24156178)

[**3.3.** **Module 06** 11](#_Toc24156179)

[**3.4.** **Module 08** 13](#_Toc24156180)

[**3.5.** **Module 11** 14](#_Toc24156181)

[**3.6.** **Module 18** 16](#_Toc24156182)

[**3.7.** **Module 19** 17](#_Toc24156183)

[**3.8.** **Module Checkout (Module 21, 22, 23)** 18](#_Toc24156184)

**DANH MỤC HÌNH ẢNH**

[Hình 1. ng-app, ng-init, ng-model. 1](#_Toc24155298)

[Hình 2. Sử dụng ng-init và ng-model. 2](#_Toc24155299)

[Hình 3. Sử dụng ng-repeat để in tất cả tên trong mảng Tên. 2](#_Toc24155300)

[Hình 4. Tạo ra 1 Directive mới với tên là demoAngular 2](#_Toc24155301)

[Hình 5. Gọi directive bên HTML. 3](#_Toc24155302)

[Hình 6. Sử dụng biểu thức. 3](#_Toc24155303)

[Hình 7. Tương tự như ví dụ trên. 3](#_Toc24155304)

[Hình 8. Biểu thức số trong {{}}. 3](#_Toc24155305)

[Hình 9. Biểu thức số trong ng-bind 3](#_Toc24155306)

[Hình 10. Xử lý chuổi và Data Binding bằng "" 4](#_Toc24155307)

[Hình 11. Xử lý đối tượng. 5](#_Toc24155308)

[Hình 12. Xử lý mảng. 5](#_Toc24155309)

[Hình 13. Tạo 1 modules với tên là myApp. 5](#_Toc24155310)

[Hình 14. Gọi modules myApp vào thẻ body. 6](#_Toc24155311)

[Hình 15. Tao một controller với tên là myController. 6](#_Toc24155312)

[Hình 16. Gọi controller myController trong thẻ body. 6](#_Toc24155313)

[Hình 17. Sử dụng đối tượng $scope. 7](#_Toc24155314)

[Hình 18. Sử dụng filters orderBy vào ng-repeat. 8](#_Toc24155315)

[Hình 19. Sử dụng $http.get để gọi file json. 9](#_Toc24155316)

[Hình 20. Sử dụng $http.get để gọi file home.html. 9](#_Toc24155317)

[Hình 21. Chuyển đổi dữ liệu từ file products.php thành dạng json. 9](#_Toc24155318)

[Hình 22. Sử dụng ng-disbled để vô hiệu hóa button khi HienAn=true. 10](#_Toc24155319)

[Hình 23. Sử dụng ng-disbled để không vô hiệu hóa button khi HienAn=false. 10](#_Toc24155320)

[Hình 24. Khởi tạo các ng-show cho các thành phần HTML. 10](#_Toc24155321)

[Hình 25. Sét các giá trị có các ng-show. 10](#_Toc24155322)

[Hình 26.Khai báo sự kiện ng-click với tên là SuKienDemo. 11](#_Toc24155323)

[Hình 27. Khởi tạo function SuKienDemo để hiện thông báo. 11](#_Toc24155324)

[Hình 28. Khởi tạo module và controller. 12](#_Toc24155325)

[Hình 29. Gọi ng-app và ng-controller bên thẻ body trong file html. 12](#_Toc24155326)

[Hình 30.Function trả về dữ liệu là products. 12](#_Toc24155327)

[Hình 31. Khởi tạo giá trị truyển qua HTML. 12](#_Toc24155328)

[Hình 32. Gọi vòng lập hiển thị sản phẩm bằng ng-repeat. 13](#_Toc24155329)

[Hình 33. Định dạng viết hoa cho Tên Sản Phẩm. 13](#_Toc24155330)

[Hình 34. Định dạng kiển hiển thị của giá tiền. 13](#_Toc24155331)

[Hình 35. Kết quả của module 04. 13](#_Toc24155332)

[Hình 36. Khai báo thư viện ui.router và pakage url RouterProvider. 14](#_Toc24155333)

[Hình 37. Sử dụng ui-view để khai báo chổ thay đổi template. 14](#_Toc24155334)

[Hình 38. Sử dụng ui-sref để biết sử dụng template nào để thay thế. 14](#_Toc24155335)

[Hình 39.Khai báo template nào được sử dụng mặc định. 14](#_Toc24155336)

[Hình 40.Cài đặt cho từng template. 14](#_Toc24155337)

[Hình 41. Function displayData trả về dữ liệu burgers. 14](#_Toc24155338)

[Hình 42. Function displayData trả về dữ liệu desserts 15](#_Toc24155339)

[Hình 43. Khai báo ng-init="displayData()" để khởi tạo dữ liệu. 15](#_Toc24155340)

[Hình 44. ng-repeat để lặp hiển thị sản phẩm. 15](#_Toc24155341)

[Hình 45. Function CheckEmail kiểm tra email. 16](#_Toc24155342)

[Hình 46. Thêm thuộc tính required. 16](#_Toc24155343)

[Hình 47. Khai báo sự kiện ng-click dangKy. 16](#_Toc24155344)

[Hình 48. Khai báo ng-show trong các lỗi có thể xuất hiện. 16](#_Toc24155345)

[Hình 49. Function upda - cập nhật giá tiền khi có thay đổi option của sản phẩm và hiện, ẩn giá tiền theo điều kiện. 17](#_Toc24155346)

[Hình 50. Function myClear - Xóa tất cả các option đã chọn và ẩn đi giá tiền. 17](#_Toc24155347)

[Hình 51. Sét các option mặc định. 17](#_Toc24155348)

[Hình 52. Sử dụng ng-model để lấy giá trị của option rồi truyền qua file JS. 17](#_Toc24155349)

[Hình 53. Khai báo sự kiện onChange. 17](#_Toc24155350)

[Hình 54. Đổ dữ liệu vào select và chỉ hiện thuộc tính tên của items1. 17](#_Toc24155351)

[Hình 55. Khởi tạo function displayData để lấy dữ liệu. 18](#_Toc24155352)

[Hình 56. Sét giá trị được sắp xếp theo mặc định. 18](#_Toc24155353)

[Hình 57. Sử dụng thư viện anglarUtils.directives.dirPagination. 18](#_Toc24155354)

[Hình 58. Khởi tạo ng-model. 18](#_Toc24155355)

[Hình 59. Khởi tạo ng-init. 18](#_Toc24155356)

[Hình 60. Sét các thuộc tính cho dir-paginate - Lặp product từ products - sắp sếp theo sortColumn - Mỗi trang chỉ hiển thị 4 sản phẩm. 18](#_Toc24155357)

[Hình 61. Sử dụng Data Binding để hiển thị thông tin sản phẩm. 18](#_Toc24155358)

[Hình 62. Gọi thẻ dir-pagination-controls để hiển thị thanh phân trnag. 19](#_Toc24155359)

[Hình 63. Khởi tạo function displayData để trả về dữ liệu. 19](#_Toc24155360)

[Hình 64. Khởi tạo dislayData 19](#_Toc24155361)

[Hình 65. Khởi tạo ng-init displayData để tạo dữ liệu. 19](#_Toc24155362)

[Hình 66. Khởi tạo ng-repeat để lặp sản phẩm product từ products. 19](#_Toc24155363)

[Hình 67. Khởi tạo function CheckEmail để kiểm tra trường email. 20](#_Toc24155364)

[Hình 68. Khởi tạo function CheckOut để gọi các hàm check ra. 20](#_Toc24155365)

[Hình 69. Sử dụng các biến để kiểm tra. 20](#_Toc24155366)

[Hình 70. Khởi tạo name cho form. 20](#_Toc24155367)

[Hình 71. Khởi tạo các giá trị ng-model, name và thuộc tính required. 20](#_Toc24155368)

[Hình 72. Khởi tạo ng-show khung báo lỗi và Data Binding hiển thị nội dụng lỗi. 21](#_Toc24155369)

# 1. Giới thiệu

* AngularJS là một Framework được viết bởi 2 người nhân viên củ của Google và sau này được Google mua lại và phát triển đến hiện tại.
* Hiện nay AngularJS rất được thịnh hành và có rất nhiều công ty tuyển lập trình cũng như thực tập sinh về AngularJS.
* AngularJS có tốc độ code và chạy rất mạnh, dễ dạng phát triển các ứng dụng, trang web Single Page Application.
* AngularJS hỗ trợ nhiều Directive tiện lợi nhanh chóng, áp dụng mô hình MVC.
* Phiên bản hiện mới nhất hiện nay của AngularJS là 1.7.8.

# 2. Các kỹ thuật của AngularJS

## 2.1. AngularJS Directives

### 2.1.1 Thuộc tính AngularJS có bắt đầu đều là ng-

##### **2.1.1.1 ng-app Directive**

**-** Lệnh ng-app khởi tạo một ứng dụng có sử dụng **AngularJS**.

**-** Lệnh này là lệnh quan trọng nhất vì nếu không có lệnh ng-app hoặc lệnh ng-app không bao gồm các phần tử bên trong thì các phần tử đó sẽ không được sử dụng Angularjs.

##### **2.1.2.1 ng-init Directive**

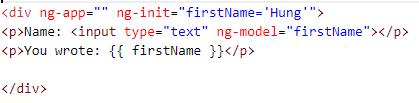
**-** Lệnh ng-init khởi tạo dữ liệu.

**-** Có thể tạo dữ liệu thẳng vào ng-init hoặc truyền data từ file JS.

**2.1.3.1 ng-model Directive**

**-** Lệnh ng-model liên kết để có thể lấy dữ liệu từ bên HTML sang bên JS.

**Ví dụ:**



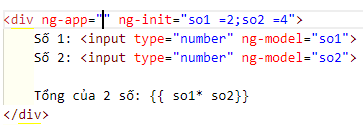
Hình 1. ng-app, ng-init, ng-model.

* Khai báo ng-app ở đây để biết tất cả những gì nằm trong thẻ div đều bị ảnh hưởng của AnglarJS.

**2.1.2 Data binding**

**-** {{ firstName }} trong ví dụ trên được hiểu giống như tên biến và nó mang một giá trị từ ng-model truyền tới.

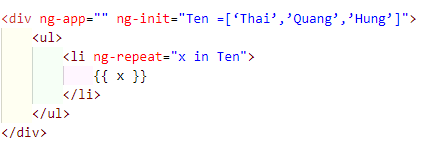
- 2 phần này phải đồng nhất về tên.



Hình 2. Sử dụng ng-init và ng-model.

**2.1.3 Lặp lại các phần tử trong một mảng dữ liệu**

- Lệnh ng-repeat được dùng khá giống for trong c và foreach trong php, là để lặp các phần tử trong một mảng.



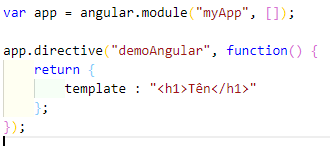
Hình 3. Sử dụng ng-repeat để in tất cả tên trong mảng Tên.

### 2.1.4 Tạo mới Directives

**-** AngularJS có rất nhiều derective mà team cũng chưa có thể sử dụng được hết nhưng vẫn có thể tạo ra nhiều directive theo nhu cầu của mỗi người.

**-** Để tao ra một directive mới chúng ta chỉ cần làm theo cú pháp app.directive(“Tên Directive”, function(){}).

- Cách gọi nó là <tên Directive> nhưng phải có “–“ trước các từ viết hoa.



Hình 4. Tạo ra 1 Directive mới với tên là demoAngular



Hình 5. Gọi directive bên HTML.

## 2.2. Biểu thức của AngularJS

- Tương tự với các ngôn ngử lập trình khác Angularjs cho phép thực hiện các biểu thức.

- Các biểu thức sẽ được thực hiện trong dấu {{Biểu thức}} hoặc ng-bind=”Biểu thức”.

- Nếu so sánh biểu thức AngularJS với biểu thức trong các ngôn ngử lập trình khác thì giống như các biểu thức ngôn ngử lập trình khác, các biểu thức AngularJS có thể chứa các ký tự, toán tử và biến.

- Không giống như các biểu thức JavaScript, các biểu thức AngularJS có thể được viết bên trong HTML.

Các biểu thức AngularJS không hỗ trợ các điều kiện, vòng lặp và ngoại lệ, trong khi các biểu thức JavaScript thì có.

Các biểu thức AngularJS hỗ trợ các bộ lọc, trong khi các biểu thức JavaScript thì không.



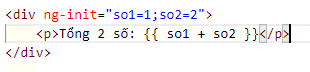
Hình 6. Sử dụng biểu thức.



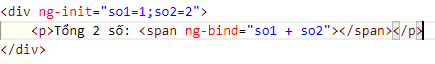
Hình 7. Tương tự như ví dụ trên.

## 2.3. Biều thức số trong AngularJS

* Số trong AngularJS giống như số trong các ngôn ngử lập trình khác.



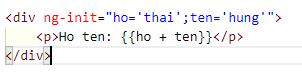
Hình 8. Biểu thức số trong {{}}.



Hình 9. Biểu thức số trong ng-bind

## 2.4. Biểu thức chuỗi AngularJS

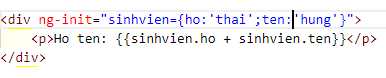
- Biểu thức chuỗi AngularJS cũng giống như biểu thức chuỗi của các ngôn ngử lập trình khác.



Hình 10. Xử lý chuổi và Data Binding bằng ""

## 2.5. Đối tượng trong AngularJS

Đối tượng AngularJS giống như các đối tượng trong các ngôn ngử lập trình khác.



Hình 11. Xử lý đối tượng.

## 2.6. Mảng trong AngularJS

Mảng **AngularJS** giống như các mảngcác ngôn ngử lập trình khác:



Hình 12. Xử lý mảng.

## 2.7. Modules AngularJS

* Trong báo cáo ở phần **ng-app Directive** mình có nói về ng-app nhưng chưa nói sâu về ng-app lắm. Và mình cũng chưa truyền tham số vào trong 2 dâu “”, vậy trong 2 dâu “” chứa gì.
* Trong dấu “” sẽ chứa tên một modules.
* Một module được hiểu như là 1 container trong html hoặc một class trong java nó chứa các function, biến nội bộ.
* Những function, biến đó chỉ được sử dụng khi nó nằm trong khai báo ng-app=”Tên modules”.
  + 1. **Tạo Modules**
* Để tạo một modules thì sử dụng câu lệnh angular.module(“Tên Modules”,[]).



Hình 13. Tạo 1 modules với tên là myApp.



Hình 14. Gọi modules myApp vào thẻ body.

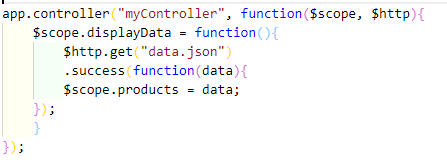
* Ở ví dụ trên thì tất cả các đoạn code nằm trong thẻ body đều có thể chạy AngularJS và modules myApp.

## 2.8. Modules AngularJS

* Controller là một thành phần con của Modules, theo như mình hiểu thì nó như một bộ điều khiển hoặc một function lớn và nó này chứa các function con và biến.
* Chính vì Controller là một thành phần con của Modules nên khi gọi controller thì phải gọi bên trong hoặc ngay sau gọi Modules.

**Tạo Controller**

* Để tao một Controller thì dùng cú pháp .controller(“Tên Controller”, function($scope))



Hình 15. Tao một controller với tên là myController.

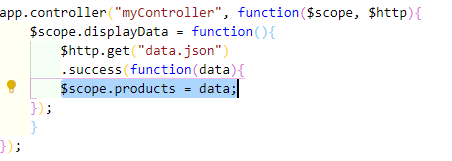


Hình 16. Gọi controller myController trong thẻ body.

* Trong ví dụ trên thì Controller myController sẽ có hiệu lực với tất cả phần tử trong thẻ body.

## 2.9 AngularJS Scopes

* $scope là một thành phần cực kỳ quan trong trong angularjs.
* $scope là một đối tượng có sẳn giúp liên kết dữ liệu, biến từ file JS sang bên HTML.



Hình 17. Sử dụng đối tượng $scope.

* Ngoài $scope thì còn có $rootScope các bạn có thể tim hiểu sâu hơn về nó.

## 2.10 AngularJS Filters

AngularJS cung cấp rất nhiều bộ lọc để chuyển đổi dữ liệu, đây là một tiện ích rất lớn và rất dễ sử dụng thay cho BackEnd mà JavaScript hoặc Jquery không có:

* Curency: Định dạng một số thành một định dạng tiền tệ.
* Date: Định dạng một ngày đến một định dạng cụ thể.
* Filter: Chọn một tập hợp con của các mục từ một mảng.
* Json: Định dạng một đối tượng thành một chuỗi JSON.
* limitTo: Giới hạn một mảng / chuỗi, vào một số phần tử / ký tự được chỉ định.
* Lowercase: Định dạng một chuỗi để viết thường.
* Number: Định dạng một số thành một chuỗi.
* orderBy: Đặt một mảng bằng một biểu thức.
* uppercase: Định dạng một chuỗi để viết hoa.

Thêm filters vào Directives

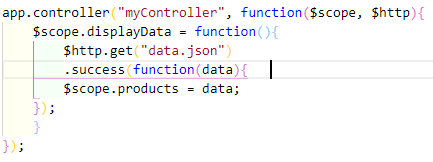
Các bộ lọc được thêm vào các chỉ thị, như ng-repeat, bằng cách sử dụng ký tự ống |, theo sau là bộ lọc



Hình 18. Sử dụng filters orderBy vào ng-repeat.

## 2.11 AngularJS $http

* $http là một service AngularJS đưa ra yêu cầu đến máu chủ và trả về phản hồi



Hình 19. Sử dụng $http.get để gọi file json.



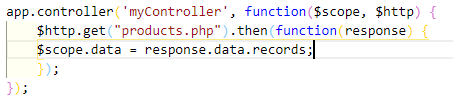
Hình 20. Sử dụng $http.get để gọi file home.html.

* Ngoài $http.get để lấy các file khác về thì còn có rất nhiều cái khác với mục đích khác.
  + .delete()
  + .head()
  + .jsonp()
  + .patch()
  + .post()
  + .put()

## 2.12 Angular JSON

- Ép kiều dữ liệu trả về sẽ chuyển thành định dạng JSON

- Tương tự lệnh json\_encode trong PHP.

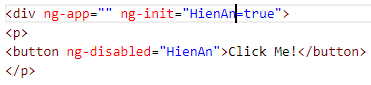


Hình 21. Chuyển đổi dữ liệu từ file products.php thành dạng json.

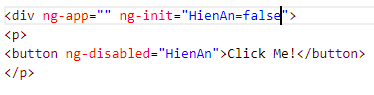
## 2.13 AngularJS DOM

**2.13.1 ng-disabled Directive**

* Lệnh ng-disabled có 2 giá trị là true hoặc false giúp vô hiệu hóa phần tử HTML nếu là true và không vô hiệu hóa nếu là false.



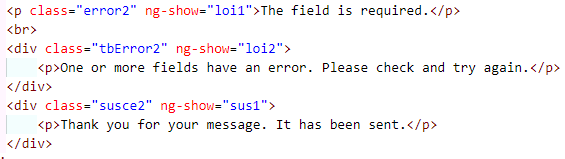
Hình 22. Sử dụng ng-disbled để vô hiệu hóa button khi HienAn=true.



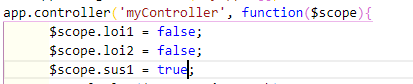
Hình 23. Sử dụng ng-disbled để không vô hiệu hóa button khi HienAn=false.

#### **2.13.2 ng-show Directive**

* Lệnh ng-show hiển thị nếu mang giá trị là true hoặc ẩn nếu mang giá trị là false một phần tử HTML.



Hình 24. Khởi tạo các ng-show cho các thành phần HTML.



Hình 25. Sét các giá trị có các ng-show.

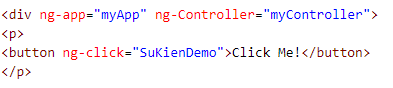
* Trong ví dụ trên thì kết quả sẽ là hiện 1 class là susce2 vì ng-show nó mang giá trị true. Ẩn 2 class là error2 và tbError2 vì mang giá trị false.

#### **2.13.3 ng-hide Directive**

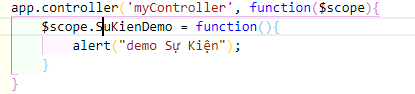
- Lênh ng-hide tương tự như lệnh ng-show nhưng mang giá trị ngược lại.

## 2.14 AngularJS Events

* Tương tự như JavaScript thì AngularJS hỗ trợ rất nhiều sự kiện, còn nhiều hơn cả JavaScript và Jquery.
* Cách gọi sự kiện khá giống với Jquery.
* Cú pháp gọi sự kiện là: ng-SuKien=”Tên Function()”.
* Các sự kiện trong AngularJS:
  + ng-blur
  + ng-change
  + ng-click
  + ng-copy
  + ng-cut
  + ng-dblclick
  + ng-focus
  + ng-keydown
  + ng-keypress
  + ng-keyup
  + ng-mousedown
  + ng-mouseenter
  + ng-mouseleave
  + ng-mousemove
  + ng-mouseover
  + ng-mouseup
  + ng-paste
  + ng-mouseover
  + ng-mouseenter
  + ng-mousemove
  + ng-mouseleave
  + ng-mousedown
  + ng-mouseup
  + ng-click



Hình 26.Khai báo sự kiện ng-click với tên là SuKienDemo.



Hình 27. Khởi tạo function SuKienDemo để hiện thông báo.

# 3. Báo cáo tích hợp Angular JS vào các module.

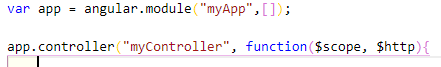
- Sử dụng Framework AngularJS và toàn bộ kiến thức đã học được trong 2 năm để có thể tạo những module hoàn hảo cho dự án phát triển web 2.

- Phiển bản sử dụng là Angular JS 1.4.8.

- Các module áp dụng Angular JS: 04, 06, 08, 11, 18, 19, checkout.

1. **Khởi tạo ứng dụng AngularJS.**

* Tất cả các module áp dụng AngularJS đều cần khởi tạo module và controller bên file JS và gọi ng-app và ng-controller ra tại thẻ **body**.



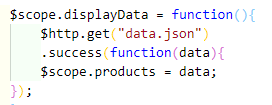
Hình 28. Khởi tạo module và controller.



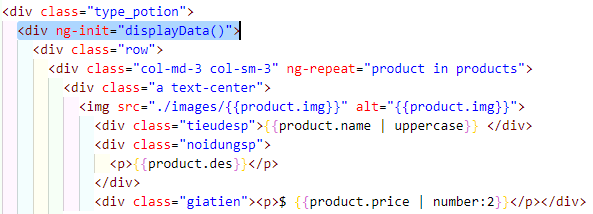
Hình 29. Gọi ng-app và ng-controller bên thẻ body trong file html.

1. **Module 04.**

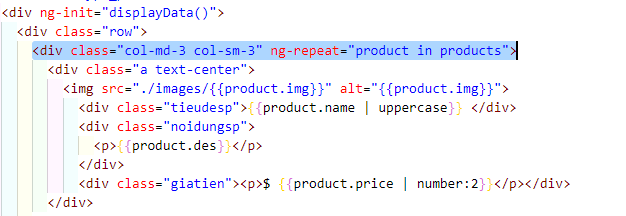
* Khởi tạo function displayData để trả về dữ liệu “data”.
* Sử dụng ng-init để gọi dữ liệu truyền vào từ file JS.
* Sử dụng ng-repeat để hiển thị sản phẩm theo dạng vòng lặp.
* Sử dụng Data Binding để hiển thị sản phẩm theo HTML.
* Sử dụng filter uppercase để hiển thị tên sản phẩm dạng viết hoa toàn bộ.
* Sử dụng filter number để định dạng kiểu hiển thị giá tiền.



Hình 30.Function trả về dữ liệu là products.



Hình 31. Khởi tạo giá trị truyển qua HTML.



Hình 32. Gọi vòng lập hiển thị sản phẩm bằng ng-repeat.



Hình 33. Định dạng viết hoa cho Tên Sản Phẩm.



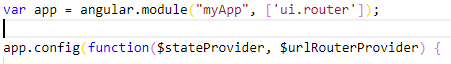
Hình 34. Định dạng kiển hiển thị của giá tiền.



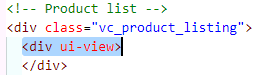
Hình 35. Kết quả của module 04.

1. **Module 06**

* Khởi tạo function config, thư viện ui.router và package urlRouterProvider để có thể thay đổi template mà không cần load lại trang.
* Sử dụng ui-view bên HTML để khai báo đoạn sẽ có thay đổi template.
* Sử dụng ui-sref để biết lấy template nào.
* Sử dụng fuction displayData để trả về dữ liệu “data”.
* Sử dụng ng-init để gọi dữ liệu truyền vào từ file JS.
* Sử dụng ng-repeat để hiển thị sản phẩm theo dạng vòng lặp.
* Sử dụng Data Binding để hiển thị sản phẩm theo HTML.



Hình 36. Khai báo thư viện ui.router và pakage url RouterProvider.



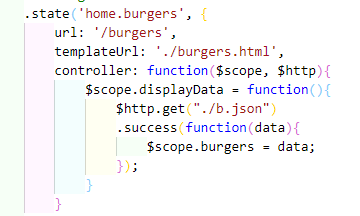
Hình 37. Sử dụng ui-view để khai báo chổ thay đổi template.



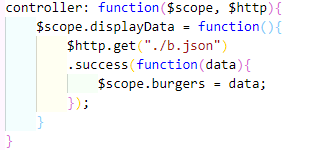
Hình 38. Sử dụng ui-sref để biết sử dụng template nào để thay thế.



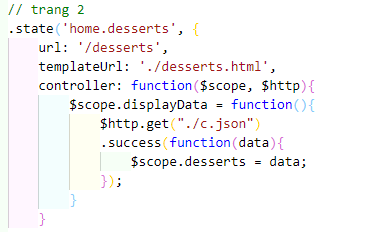
Hình 39.Khai báo template nào được sử dụng mặc định.



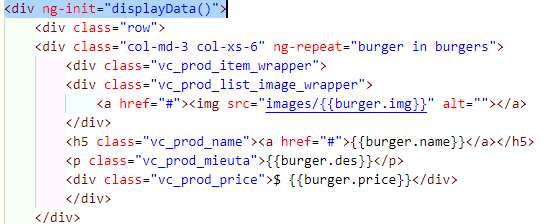
Hình 40.Cài đặt cho từng template.



Hình 41. Function displayData trả về dữ liệu burgers.



Hình 42. Function displayData trả về dữ liệu desserts



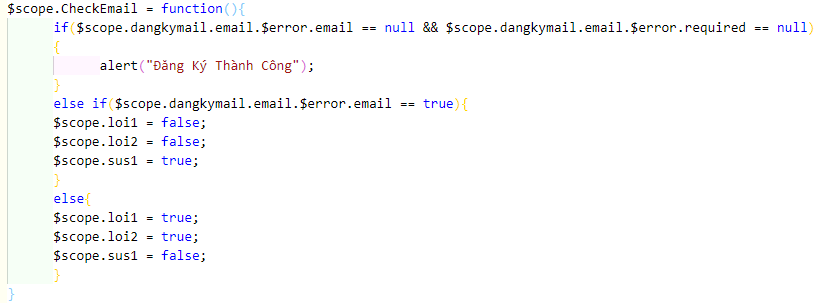
Hình 43. Khai báo ng-init="displayData()" để khởi tạo dữ liệu.



Hình 44. ng-repeat để lặp hiển thị sản phẩm.

1. **Module 08**

* Khởi tạo function CheckEmail để kiểm tra ô nhập email.
* Sử dụng thuộc tính required để bắt buộc phải nhập.
* Sử dụng các giá trị dangkymail.email.$error.email để kiểm tra có đúng định dạng của email và dangkymail.email.$error.required để kiểm tra đã nhập email chưa.
* Sử dụng ng-show để hiện thông báo lỗi.



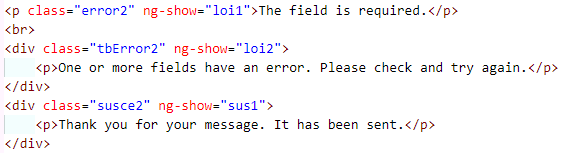
Hình 45. Function CheckEmail kiểm tra email.



Hình 46. Thêm thuộc tính required.



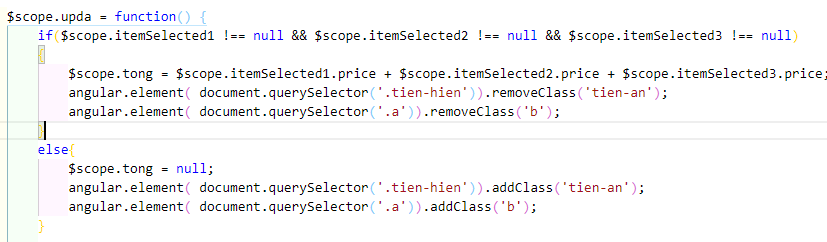
Hình 47. Khai báo sự kiện ng-click dangKy.



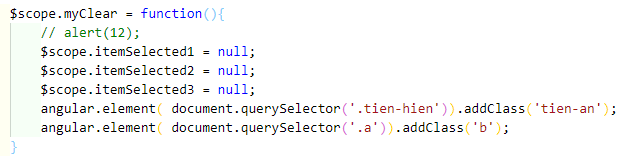
Hình 48. Khai báo ng-show trong các lỗi có thể xuất hiện.

1. **Module 11**

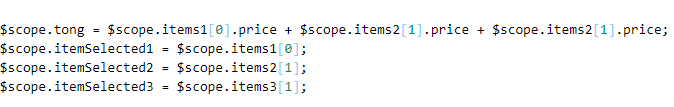
* Khởi tạo function upda() và ng-change để cập nhật giá tiền sau khi chọn option của sản phẩm.
* Khởi tạo function myClear() để xóa các option đã chọn.
* Sử dụng ng-model và data-ng-option để đổ dữ liệu từ file json ra option.
* Sử dụng addClass và removeClass để ẩn hiện giá tiền.



Hình 49. Function upda - cập nhật giá tiền khi có thay đổi option của sản phẩm và hiện, ẩn giá tiền theo điều kiện.



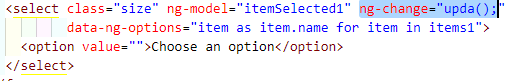
Hình 50. Function myClear - Xóa tất cả các option đã chọn và ẩn đi giá tiền.



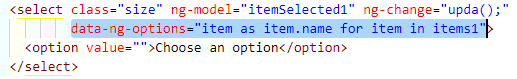
Hình 51. Sét các option mặc định.



Hình 52. Sử dụng ng-model để lấy giá trị của option rồi truyền qua file JS.



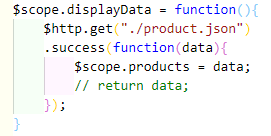
Hình 53. Khai báo sự kiện onChange.



Hình 54. Đổ dữ liệu vào select và chỉ hiện thuộc tính tên của items1.

1. **Module 18**

* Sử dụng fuction displayData để trả về dữ liệu “products”.
* Set sắp xếp mặc định là “menu\_order”.
* Sử dụng thư viện angularUtils.directives.dirPagination để có thể phân trang.
* Khởi tạo ng-init displayData để đổ dữ liệu qua.
* Khởi tạo ng-model sortColumn để liên kết dữ liệu.
* Sét các giá trị dir-paginate.
* Sử dụng Data Binding để hiển thị dữ liệu.
* Sử dụng thẻ dir-pagination-controls để hiển thị thanh phân trang.



Hình 55. Khởi tạo function displayData để lấy dữ liệu.



Hình 56. Sét giá trị được sắp xếp theo mặc định.



Hình 57. Sử dụng thư viện anglarUtils.directives.dirPagination.



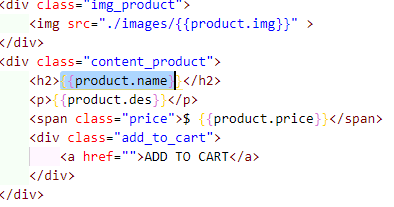
Hình 58. Khởi tạo ng-model.



Hình 59. Khởi tạo ng-init.



Hình 60. Sét các thuộc tính cho dir-paginate - Lặp product từ products - sắp sếp theo sortColumn - Mỗi trang chỉ hiển thị 4 sản phẩm.



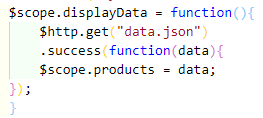
Hình 61. Sử dụng Data Binding để hiển thị thông tin sản phẩm.



Hình 62. Gọi thẻ dir-pagination-controls để hiển thị thanh phân trnag.

1. **Module 19**

* Sử dụng function displayData trả lại dữ liệu.
* Sử dụng ng-init để khởi tạo dữ liệu.
* Sử dụng ng-repeat để lặp sản phẩm ra HTML.
* Sử dụng Data Binding để hiện dữ liệu ra dạng HTML.



Hình 63. Khởi tạo function displayData để trả về dữ liệu.



Hình 64. Khởi tạo dislayData



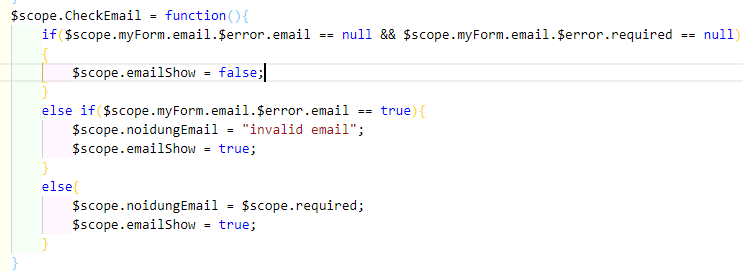
Hình 65. Khởi tạo ng-init displayData để tạo dữ liệu.



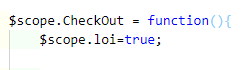
Hình 66. Khởi tạo ng-repeat để lặp sản phẩm product từ products.

1. **Module Checkout (Module 21, 22, 23)**

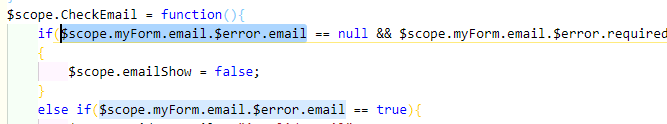
* Khởi tạo function CheckEmail để kiểm tra ô nhập email có nhập hay chưa và nhập có đúng theo định dạng email.
* Sử dụng các giá trị của các biến sau để kiểm tra email:
  + myForm.email.$error.email – Lỗi định dạng email.
  + myForm.email.$error.required – Lỗi chưa nhập email.
* Khởi tạo name cho form.
* Khởi tạo sự kiện ng-click cho function CheckOut().
* Khởi tạo ng-model email để liên kết dữ liệu.
* Sử dụng name để lấy giá trị dữ liệu.
* Sử dụng thuộc tính required để khai báo đây là trường bắt buộc nhập thông tin.
* Khởi tạo ng-hide để hiện lỗi nếu có sau khi checkout.
* Sử dụng Data Binding để hiển thị nội dụng lỗi.



Hình 67. Khởi tạo function CheckEmail để kiểm tra trường email.



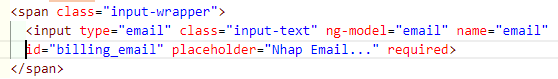
Hình 68. Khởi tạo function CheckOut để gọi các hàm check ra.



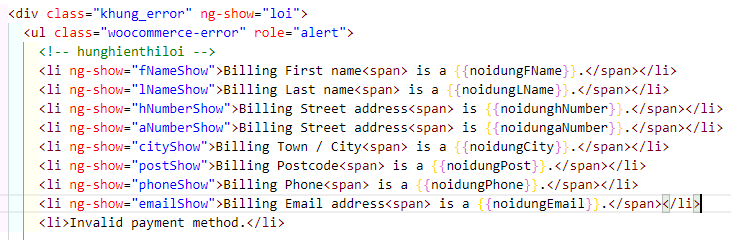
Hình 69. Sử dụng các biến để kiểm tra.



Hình 70. Khởi tạo name cho form.



Hình 71. Khởi tạo các giá trị ng-model, name và thuộc tính required.



Hình 72. Khởi tạo ng-show khung báo lỗi và Data Binding hiển thị nội dụng lỗi.

**TÀI LIỆU THAM KHẢO**

W3SCHOOL…

ANGULAR JS DOCUMENTATION,..