|  |  |  |
| --- | --- | --- |
| 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

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

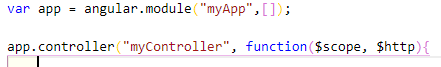
- Sử dụng Framework AngularJS và toàn bộ kiến thức đã học được trong 2 năm để có thể tạo những module hoàn hảo cho dự án phát triển web 2.

- 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**.



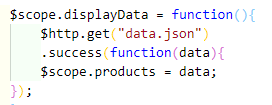
Hình 1. Khởi tạo module và controller.



Hình 2. Gọi ng-app và ng-controller bên thẻ body trong file html.

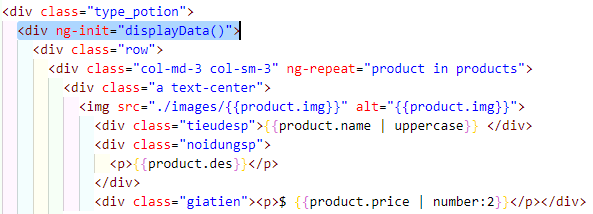
1. Module 04.

* 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.

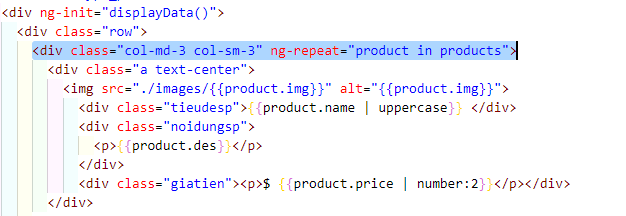


Hình 3.Function trả về dữ liệu là products.

Hình 4. Function trả về dữ liệu.



Hình 5. Khởi tạo giá trị truyển qua HTML.



Hình 6. Gọi vòng lập hiển thị sản phẩm bằng ng-repeat.



Hình 7. Định dạng viết hoa cho Tên Sản Phẩm.



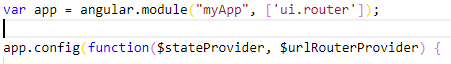
Hình 8. Định dạng kiển hiển thị của giá tiền.



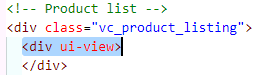
Hình 9. Kết quả của module 04.

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 10. Khai báo thư viện ui.router và pakage url RouterProvider.



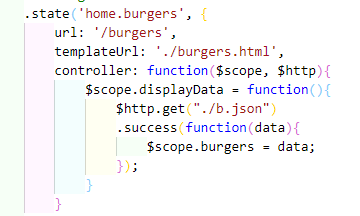
Hình 11. Sử dụng ui-view để khai báo chổ thay đổi template.



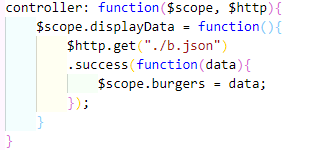
Hình 12. Sử dụng ui-sref để biết sử dụng template nào để thay thế.



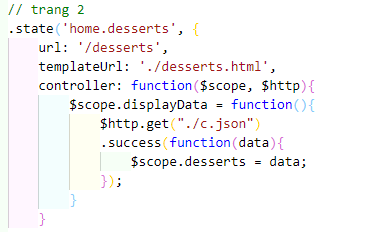
Hình 13.Khai báo template nào được sử dụng mặc định.



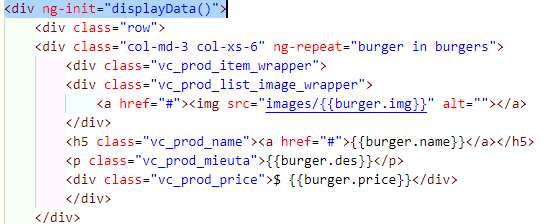
Hình 14.Cài đặt cho từng template.



Hình 15. Function displayData trả về dữ liệu burgers.



Hình 16. Function displayData trả về dữ liệu desserts



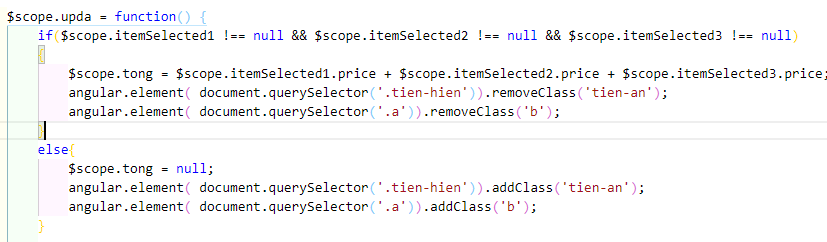
Hình 17. Khai báo ng-init="displayData()" để khởi tạo dữ liệu.



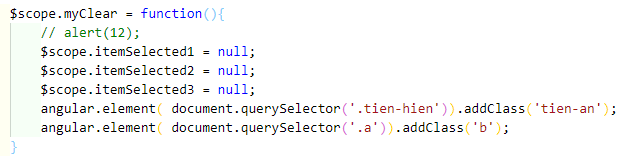
Hình 18. ng-repeat để lặp hiển thị sản phẩm.

1. Module 08
2. Module 11

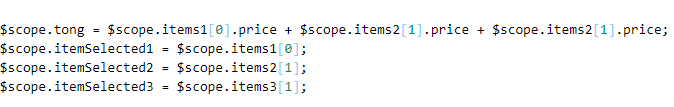
* Khởi tạo function upda() và ng-change để cập nhật giá tiền sau khi chọn option của sản phẩm.
* Khởi tạo function myClear() để xóa các option đã chọn.
* Sử dụng ng-model và data-ng-option để đổ dữ liệu từ file json ra option.
* Sử dụng addClass và removeClass để ẩn hiện giá tiền.



Hình 19. Function upda - cập nhật giá tiền khi có thay đổi option của sản phẩm và hiện, ẩn giá tiền theo điều kiện.

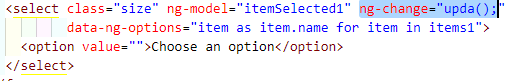


Hình 20. Function myClear - Xóa tất cả các option đã chọn và ẩn đi giá tiền.

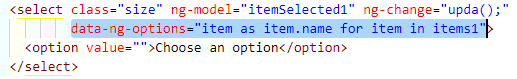


Hình 21. Sét các option mặc định.  


Hình 22. Sử dụng ng-model để lấy giá trị của option rồi truyền qua file JS.



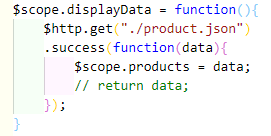
Hình 23. Khai báo sự kiện onChange.



Hình 24. 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 25. Khởi tạo function displayData để lấy dữ liệu.



Hình 26. Sét giá trị được sắp xếp theo mặc định.



Hình 27. Sử dụng thư viện anglarUtils.directives.dirPagination.



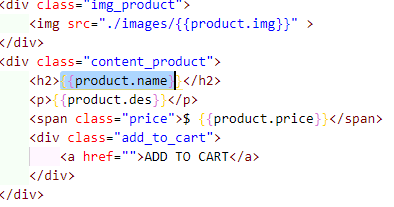
Hình 28. Khởi tạo ng-model.



Hình 29. Khởi tạo ng-init.



Hình 30. Sét các thuộc tính cho dir-paginate - Lặp product từ products - sắp sếp theo sortColumn - Mỗi trang chỉ hiển thị 4 sản phẩm.



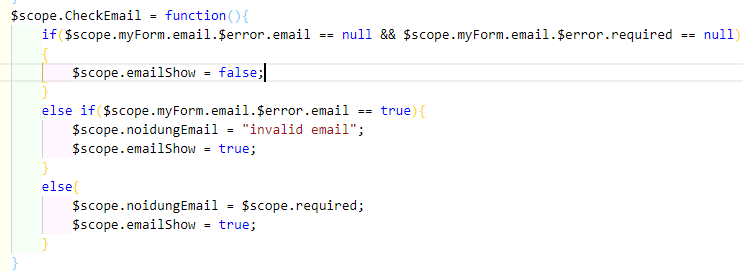
Hình 31. Sử dụng Data Binding để hiển thị thông tin sản phẩm.



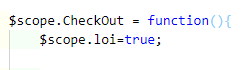
Hình 32. Gọi thẻ dir-pagination-controls để hiển thị thanh phân trnag.

1. Module 19
2. Module Checkout (Module 21, 22, 23)

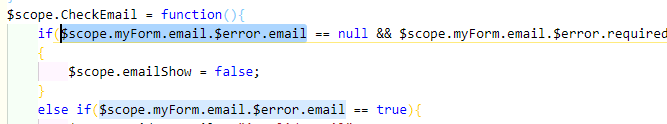
* Khởi tạo function CheckEmail để kiểm tra ô nhập email có nhập hay chưa và nhập có đúng theo định dạng email.
* Sử dụng các giá trị của các biến sau để kiểm tra email:
  + myForm.email.$error.email – Lỗi định dạng email.
  + myForm.email.$error.required – Lỗi chưa nhập email.
* Khởi tạo name cho form.
* 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 33. Khởi tạo function CheckEmail để kiểm tra trường email.



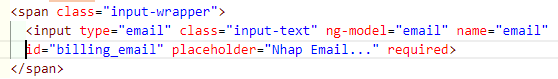
Hình 34. Khởi tạo function CheckOut để gọi các hàm check ra.



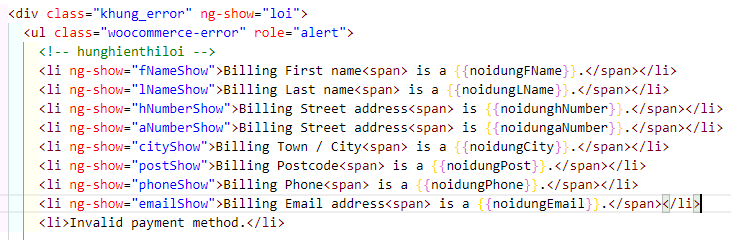
Hình 35. Sử dụng các biến để kiểm tra.



Hình 36. Khởi tạo name cho form.



Hình 37. Khởi tạo các giá trị ng-model, name và thuộc tính required.



Hình 38. Khởi tạo ng-show khung báo lỗi và Data Binding hiển thị nội dụng lỗi.