**LAB 5 – FRONTEND FRAMEWORK ( ANGULARJS)**

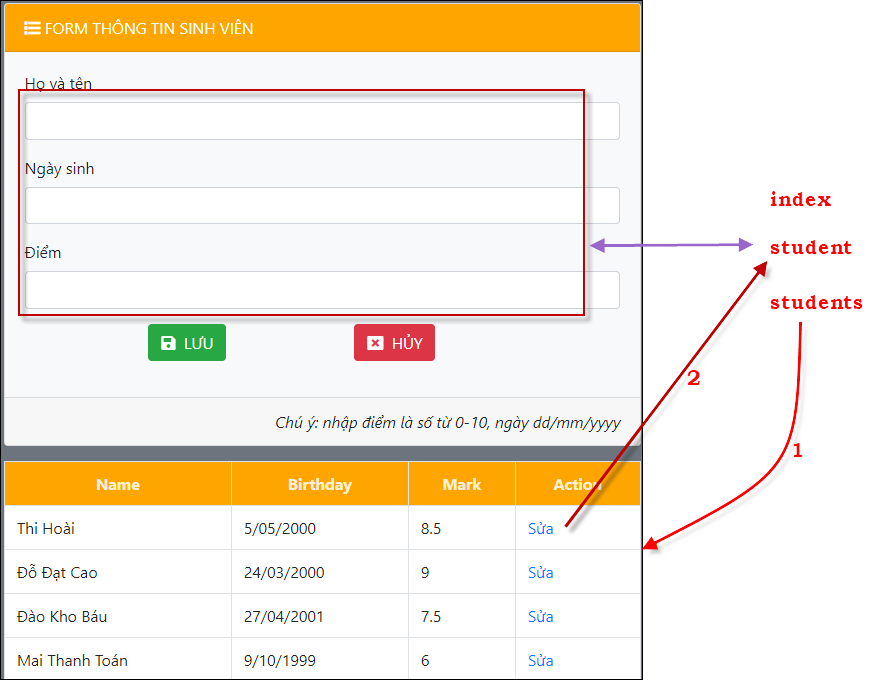
Chuẩn bị: tạo folder **<MaSV\_HoTen>\_Lab5** và làm lab trong folder vừa tạo.

### Bài 1: Chia sẻ dữ liệu giữa các $scope

Xây dựng trang web với yêu cầu sau:

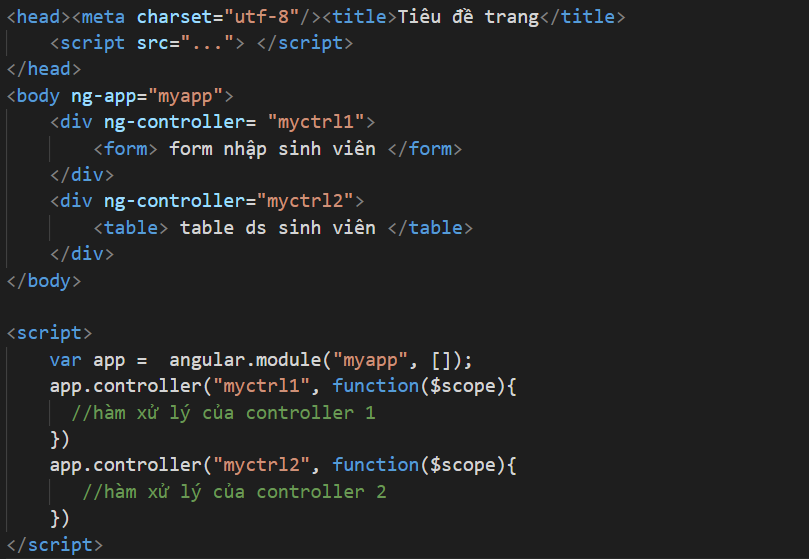
1. Giao diện sơ bộ gồm 2 phần: phần trên là form nhập liệu, đặt trong controller1, phần dưới là danh sách sinh viên đặt trong một controller2.
2. Chương trình dùng 3 biến: **index**, **student**, **students** đặt ở rootscope để dùng chung cho 2 controller

* Biến **index**: chỉ số của đối tượng sinh viên hiện tại đang chỉnh sửa trong form.
* Mảng **students** là danh sách các sinh viên.
* Đối tượng **student** là đối tượng chứa thông sinh viên hiện hành đang chỉnh sửa trong form



#### Tạo file

Tạo file **bai1.html** và code html để tạo app, nhúng thư viện angularjs, khai báo 2 controller



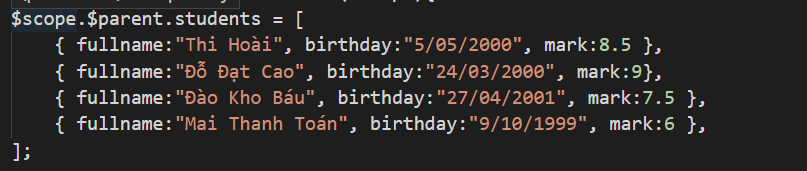
#### Thiết kế form trong controller1

Chèn thư viện bootstrap vào và tạo form có 3 textfield 2 button như mẫu + định dạng cho đẹp

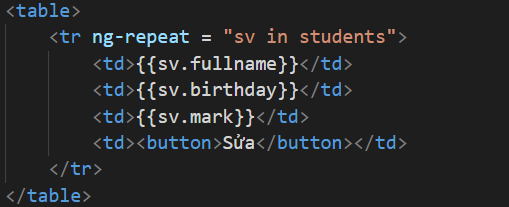


#### Hiện danh sách sinh viên trong controller2

1. Code trong hàm xử lý của controller2 để khai báo mảng **students** và đặt trong $scope cha



1. Lặp để hiện mảng students vừa định nghĩa trong controller2



1. Định đạng thêm cho đẹp:



#### Hiện thông tin 1 student vào form để sửa

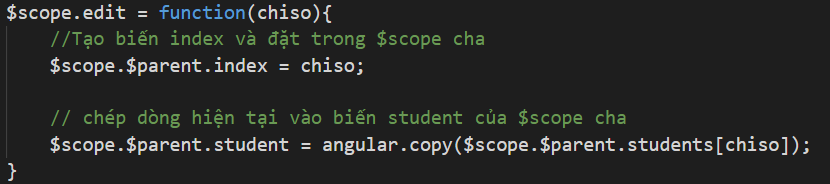
Yêu cầu: Nhắp nút **Sửa** thì thông tin của sinh viên hiện tại sẽ được hiện ra trong form

1. Gọi hàm sửa 1 student

Trong sự kiện click của button **Sửa**, gọi hàm edit và truyền tham số

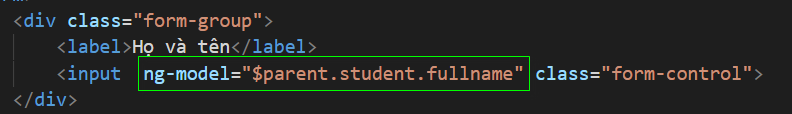


1. Định nghĩa hàm **edit** trong hàm xử lý của controller2

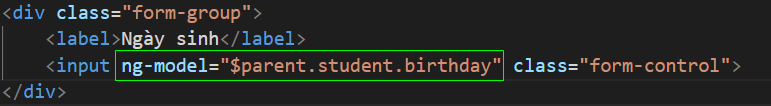


1. Hiện thông tin của biến **student** vào form

* Bind giá trị fullname của đối tượng student vào textfield họ tên:

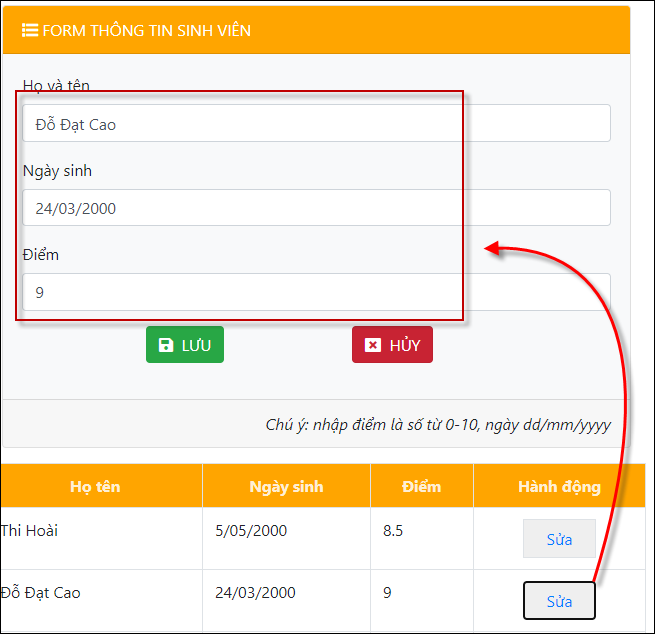


* Bind giá trị birthday của đối tượng student vào textfield Ngày sinh:



* Bind giá trị mark của đối tượng student vào textfield Điểm: Thực hiện tương tự

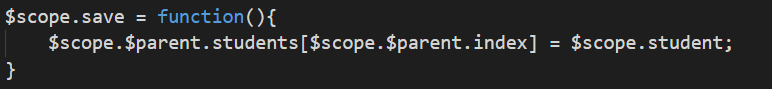
1. Test bằng cách nhắp nút **Sửa** tại 1 Sudent, sẽ thấy thông tin hiện trong form



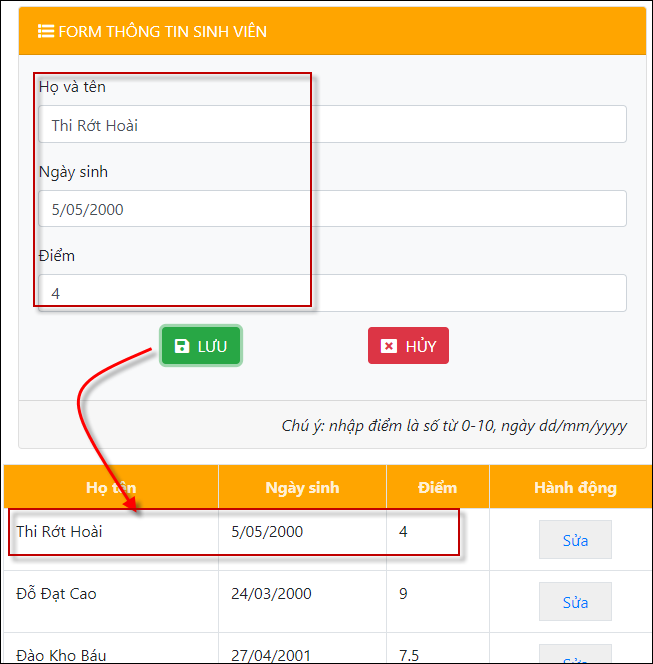
#### Lưu thông tin sinh viên từ form vào mảng students

Yêu cầu: khi nhắp nút **Save**, những gì đã chỉnh sửa trong form phải được lưu vào mảng **students**

1. Viết hàm **save** trong hàm xử lý của controller1 để lưu **student** vào mảng **students**



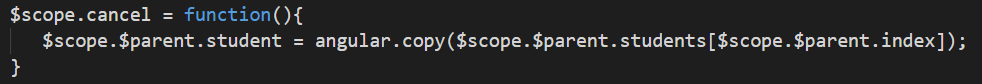
1. Gọi hàm **save** trong sự kiện click của button Save: sinh viên tự thực hiện
2. Test : nhắp nút sửa 🡺 sẽ hiện thông tin sinh viên trong form. Sau đó thực hiện chỉnh các giá trị trong form, rồi nhắp nút Save phải thấy các thay đổi trong danh sách bên dưới



#### Chức năng Hủy

Yêu cầu: Button Hủy khi nhắp vào sẽ hủy những gì mà user đã sửa trong form, trả lại nguyên gốc giá trị của **student.**

1. Định nghĩa hàm **cancel** trong hàm xử lý của controller1



1. Gọi hàm **cancel** khi nhắp nút Hủy: sinh viên tự thực hiện
2. Test: test xem có đúng yêu cầu không.

#### Sử dụng từ khóa rootScope

* Save as file bai1.html thành bai1a.html
* Thay $scope.$parent bằng $rootScope và truyền $rootScope vào các controller
* Test: sẽ phải thấy hoạt động không thay đổi.

**Bài 2 ở dưới**

### Bài 2: Xây dựng trang CRUD với AngularJS

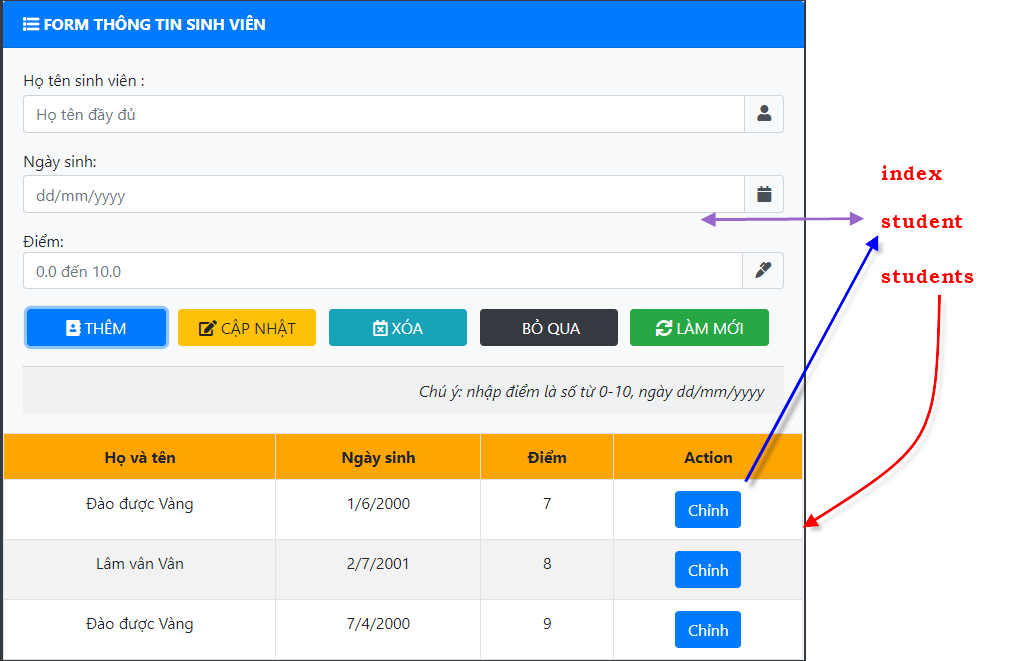
Xây dựng trang web với yêu cầu sau:

1. Giao diện gồm 2 phần: phần trên là form nhập liệu, phần dưới là danh sách sinh viên.
2. Chương trình dùng 3 biến: **index**, **student**, **students**

* Mảng **students** là danh sách các sinh viên
* Biến **index**: chỉ số của đối tượng **student** (trong **students**) đang chỉnh sửa trong form.
* Đối tượng **student** là đối tượng chứa thông sinh viên hiện hành đang chỉnh sửa trong form

1. Các nút chức năng

* Nhắp nút **Chỉnh** sẽ hiển thị thông tin của sinh viên lên form.
* Nhắp nút **CẬP NHẬT** sẽ cập nhật dữ liệu từ form vào trên dưới.
* Nhắp nút **XÓA** sẽ xóa student đang xem khỏi danh sách và xóa trắng form
* Nhắp nút **BỎ QUA** sẽ phục hồi lại dữ liệu gốc (lúc chưa sửa)
* Nhấp nút **LÀM MỚI** sẽ xóa trắng form



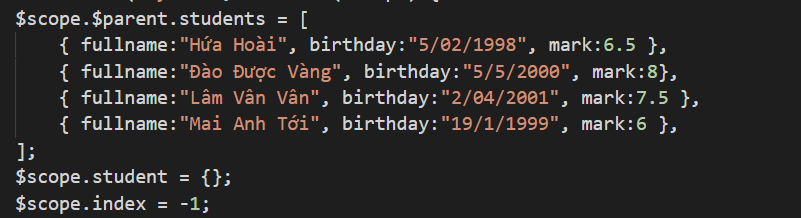
#### Tạo file

Tạo file **bai2.html** và code tạo app, nhúng thư viện angularjs

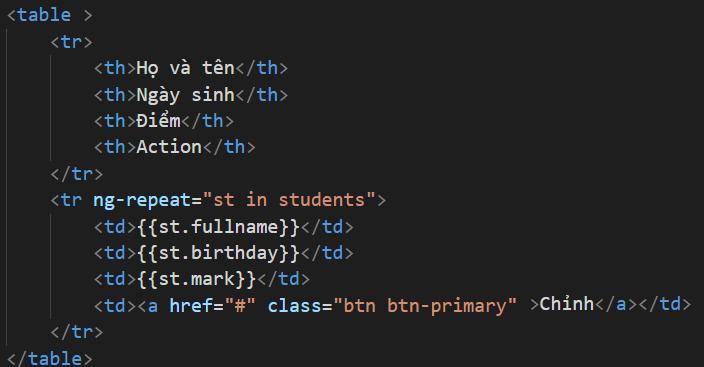


#### Hiện danh sách sinh viên

1. Code trong hàm xử lý của controller: khai báo biến **students**,**student**,**index**

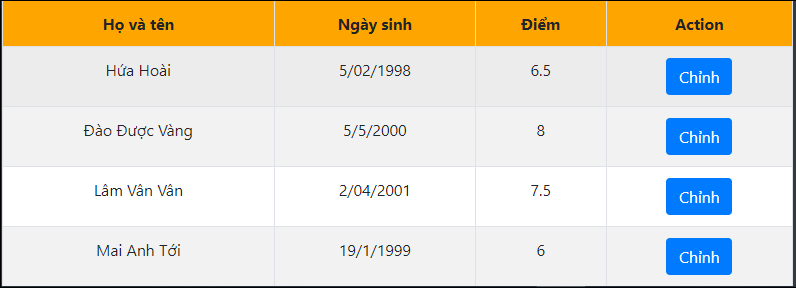


1. Tạo table và lặp qua mảng **students** vừa định nghĩa trong controller2



1. Định đạng thêm cho đẹp:

Chèn thư viện bootstrap vào và định dạng cho đẹp



#### Thiết kế form trong controller

Tạo form như sau ở trên table danh sách sinh viên. Form có 3 textfield và 5 button như hình + định dạng cho đẹp.



#### Hiện thông tin của student cần chỉnh vào form:

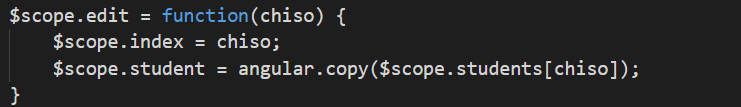
* Trong sự kiện ng-click của link Chỉnh, gọi hàm edit và truyền tham số là chỉ số của dòng hiện tại



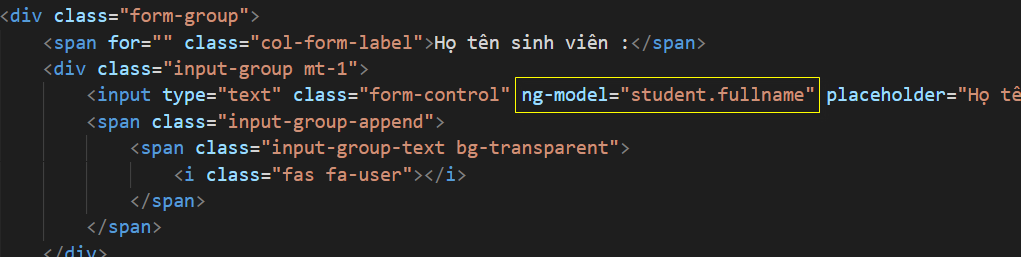
* Định nghĩa hàm **edit** trong hàm xử lý của controller :

+ Ghi nhận chỉ số dòng hiện tại vào $scope

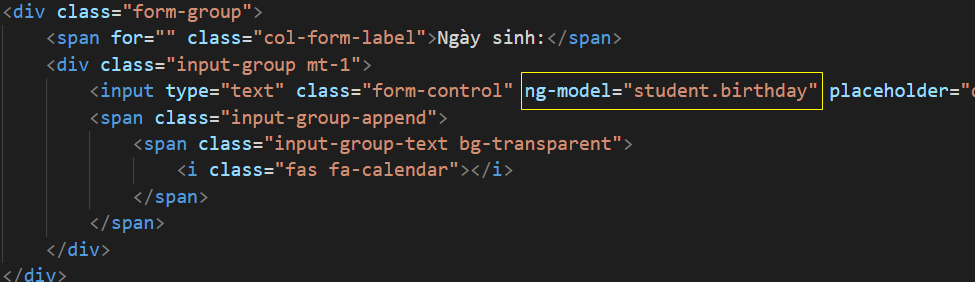
+ Lấy dòng hiện tại của mảng **students** đưa vào biến **student**



* Bind giá trị fullname của đối tượng **student** vào textfield họ tên:



* Bind giá trị birthday của đối tượng student vào textfield Ngày sinh:



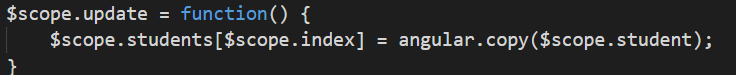
* Bind giá trị mark của đối tượng student vào textfield Điểm: Thực hiện tương tự
* Test: Nhắp nút Chỉnh, phải thấy thông tin sinh viên hiện trong form (nếu không thì mua cái máy mới rồi làm lại ☺ )

#### Cập nhật thông tin từ form vào mảng students:

* Gọi hàm **update()**: Trong sự kiện ng-click của button **Cập nhật**, gọi hàm update()



* Định nghĩa hàm **update()** trong hàm xử lý của controller

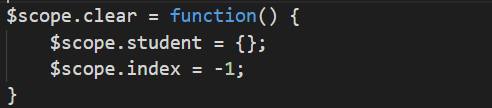


* Test: Sửa giá trị trong form và nhắp nút Cập nhật 🡺 phải thấy giá trị được lưu trong dãy

#### Nút Làm mới

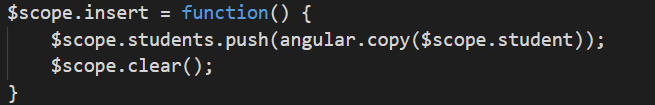
Dùng để bỏ qua các thông tin student hiện tại, chuẩn bị thêm 1 student mới

* Gọi hàm **clear()** trong sự kiện ng-click của button **Làm mới** : Sinh viên tự thực hiện
* Định nghĩa hàm clear() trong $scope: (Reset giá trị student và index)



#### Thêm 1 sinh viên

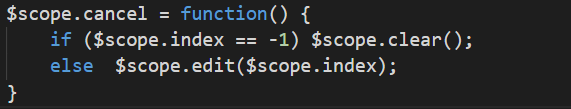
* Gọi hàm **insert()** trong sự kiện ng-click của button **Thêm**: Sinh viên tự thực hiện.
* Định nghĩa hàm insert() trong $scope: (Bổ sung thông tin trong form vào dãy **students** và clear form)



* Test

#### Nút bỏ qua

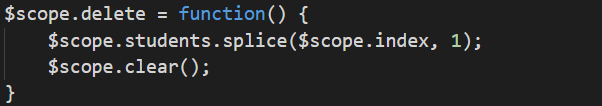
* Gọi hàm **cancel()** trong sự kiện ng-click của button **Bỏ qua**: Sinh viên tự thực hiện.
* Định nghĩa hàm cancel() trong $scope: (Bỏ qua những gì đã gõ trong form)



* Test

#### Xóa student

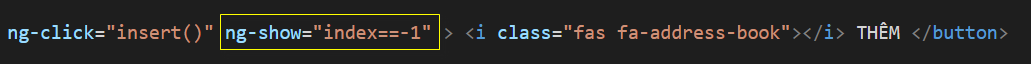
* Gọi hàm **delete()** trong sự kiện ng-click của button **Xóa**: Sinh viên tự thực hiện.
* Định nghĩa hàm delete() trong $scope: (Xóa sinh viên đang hiện trong form)



#### Hiển thị các nút phù hợp với hành động người dùng

1. Save as **bai2.html** thành **bai2a.html**
2. Viết điều kiện để nút **THÊM** chỉ hiển thị khi user nhắp nút **Làm mới** ($scope.index là -1)

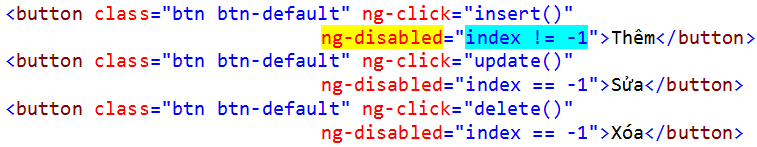
Code trong button **THÊM**



1. Viết điều kiện để nút **CẬP NHẬT** và nút **XÓA** chỉ hiển thị khi $scope.index != -1

Thực hiện tương tự

Nói thêm chi biết (khỏi làm): có thể không ẩn các button mà chỉ khóa button với lệnh **@ng-disabled**



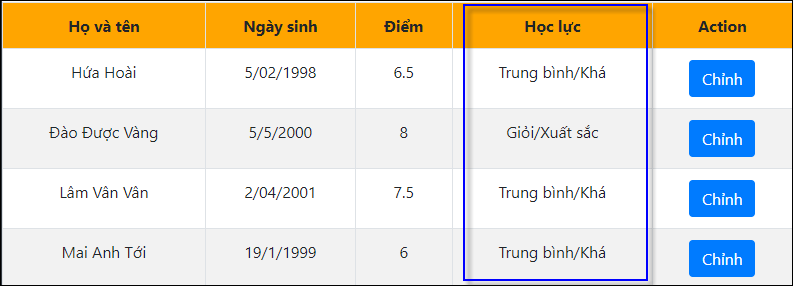
#### Bổ sung cột loại học lực vào bảng danh sách sinh viên

Công thức xếp học lực :

* Điểm < 5: Yếu/Kém
* 5<=Điểm<8: Trung bình/Khá
* Điểm>=8: Giỏi/xuất sắc

Hướng dẫn: Dùng 3 tag span chứa 3 hạng học lực và dùng ng-if trong từng tag span

Kết quả như sau:



### Bài 3 : Xây dựng trang hiện sản phẩm

Cho mảng chứa danh sách sản phẩm như sau:

$scope.products = [

 { name: "Samsung Galaxy Tab A", image: "1.png", price:  1000 },

{ name: "Máy tính bảng Mobell", image: "2.png", price: 1500 },

{ name: "Điện thoại Iphone 8", image: "3.png", price: 2000 },

{ name: "Điện thoại BlackBarry", image: "4.png", price: 3000 },

{ name: "Điện thoại Realme 5", image: "5.png", price: 2000 },

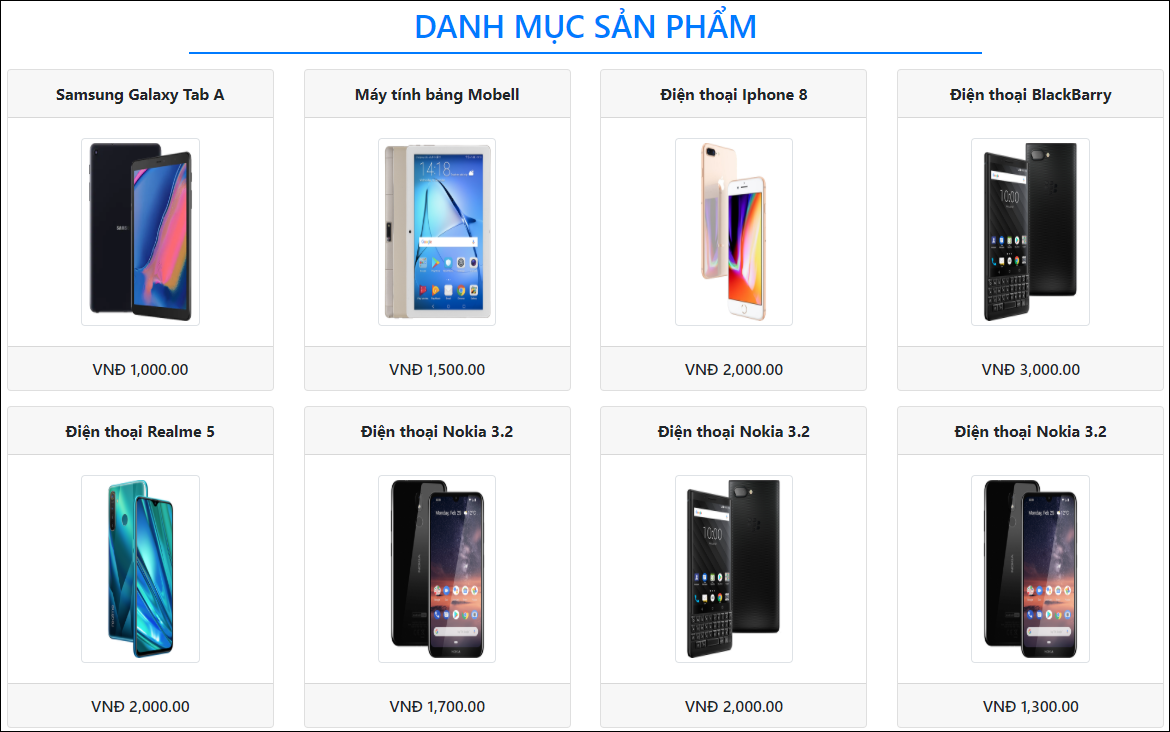
{ name: "Điện thoại Nokia 3.2", image: "6.png", price: 1700 },

{ name: "Điện thoại Nokia 3.2", image: "7.png", price: 2000 },

{ name: "Điện thoại Nokia 3.2", image: "8.png", price: 1300 }

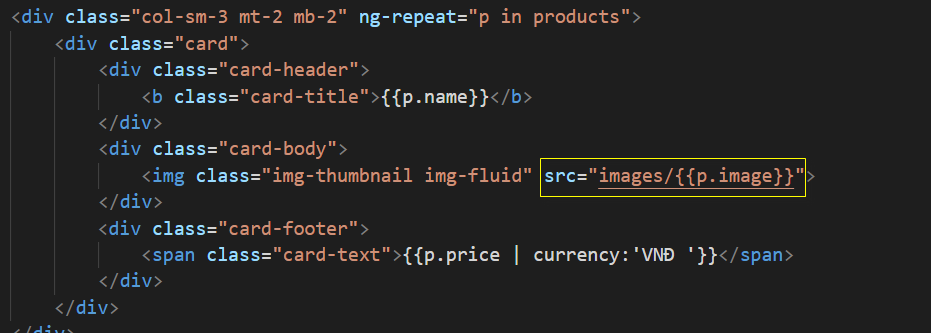
];

Hãy tạo trang web hiển thị các sản phẩm như hình:



Hướng dẫn :

* Tìm 8 hình sản phẩm lưu vào folder **images** với tên file lần lượt là **1.png** đến **8.png**
* Dùng đối tượng card của bootstrap để lặp
* Bind thuộc tính image của mỗi sản phẩm vào src của tag image



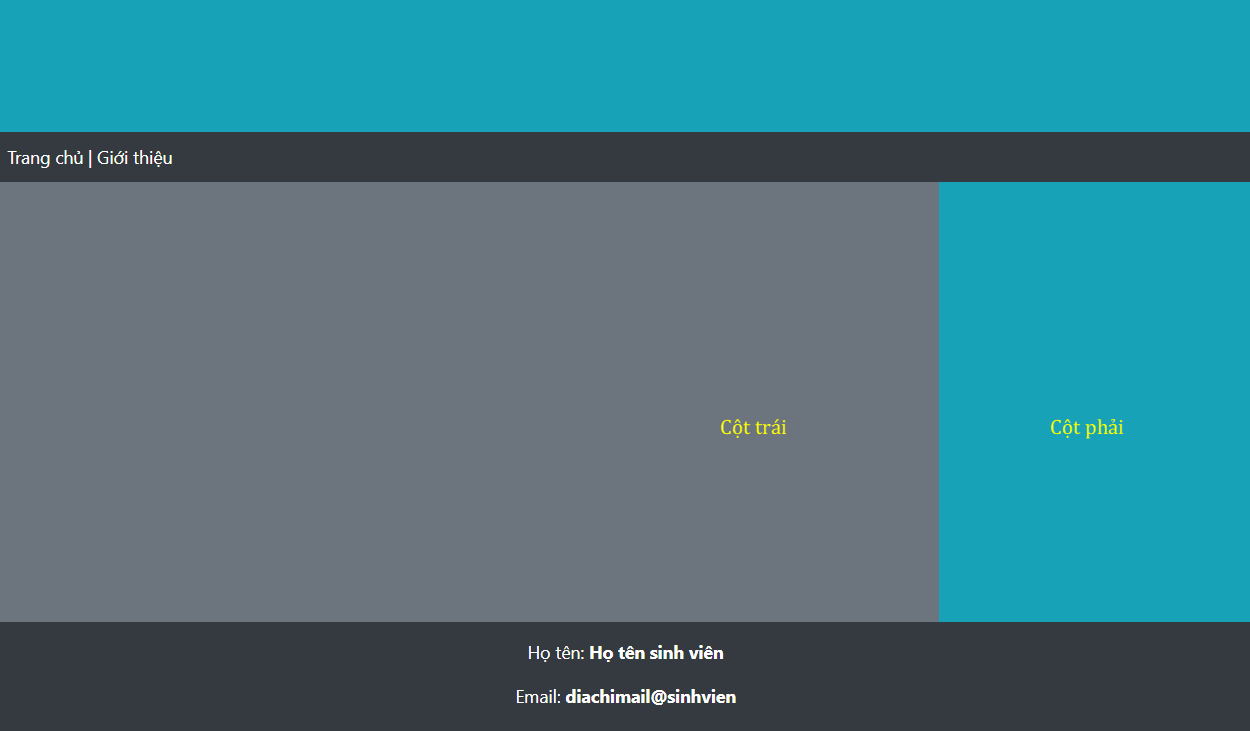
* Định dạng cho đẹp

Bài 4 phía dưới

### Bài 4 : Hiện thể loại , loại tin, tin tức

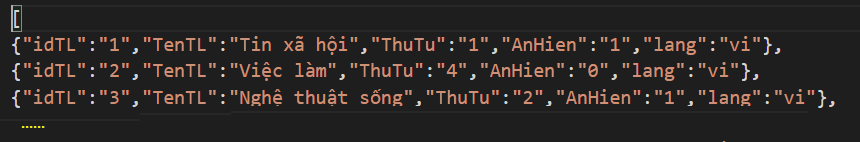
#### Tạo file

Tạo file **bai4.html** và dùng bootstrap để tạo layout như sau

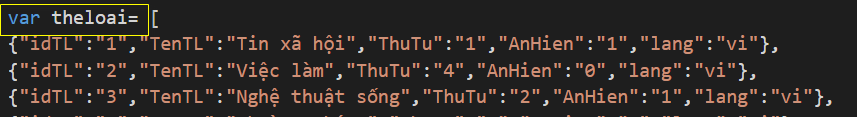


#### Hiện thể loại vào thanh menu

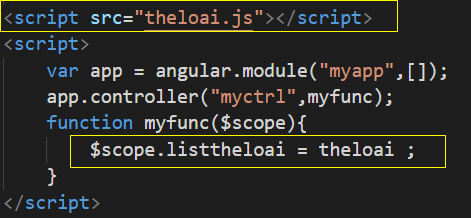
1. Mở file **theloai.js** bạn sẽ thấy mảng các thể loại như sau:



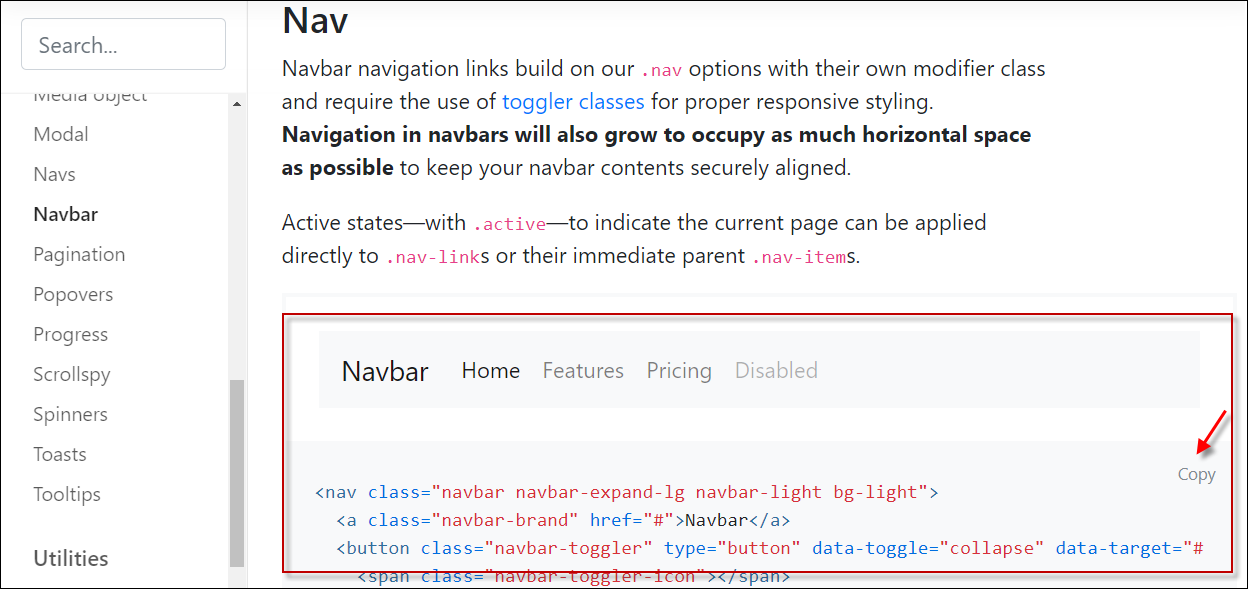
1. Khai báo biến **theloai** như sau:



1. Nhúng file theloai.js vào trang **bai4.html** và đưa giá trị biến theloai vào $scope:



1. Trong thanh menu: xóa chữ Trang chủ | Liên hệ và chèn thanh navbar của bootstrap vào

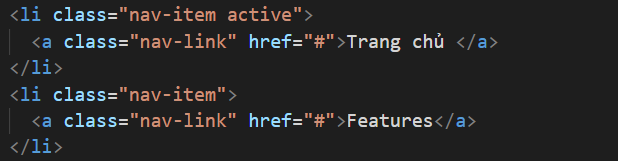


Xem thử:

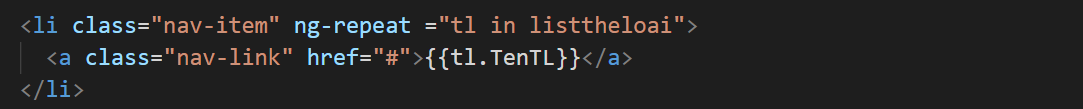


1. Xóa logo Navbar và thay chữ **Home** thành **Trang chủ**
2. Xóa mục **Pricing** và **Disabled** .

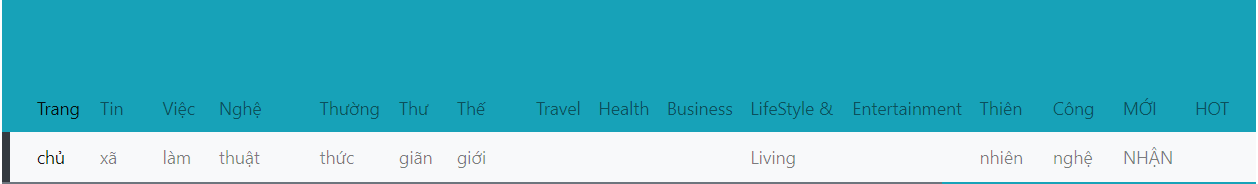
Kết quả còn lại gồm 2 mục như sau:



1. Xử lý với mục Features: thêm **ng-repeat** để lặp qua dãy listtheloai và hiện tên thể loại thay cho chữ Featured



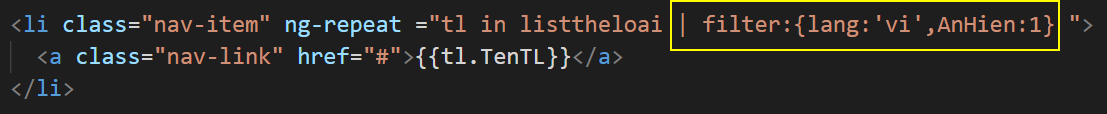
Xem thử sẽ thấy rất nhiều item



#### Lọc bớt các mục theo điều kiện khi lặp

Khi lặp, có thể lọc bớt các item cần lặp, tức là chỉ lặp các mục thỏa 1 điều kiện nào đó.

Thêm code sau vào trong chỉ thị ng-repeat ở trên

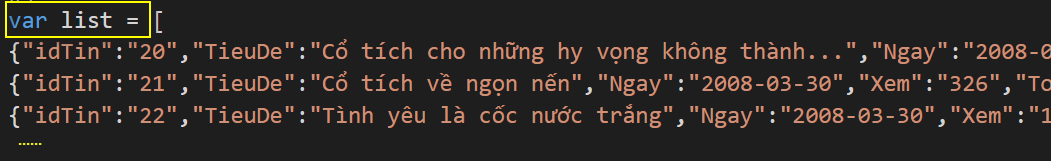


Xem thử sẽ chỉ thấy các thể loại thỏa điều kiện được lọc

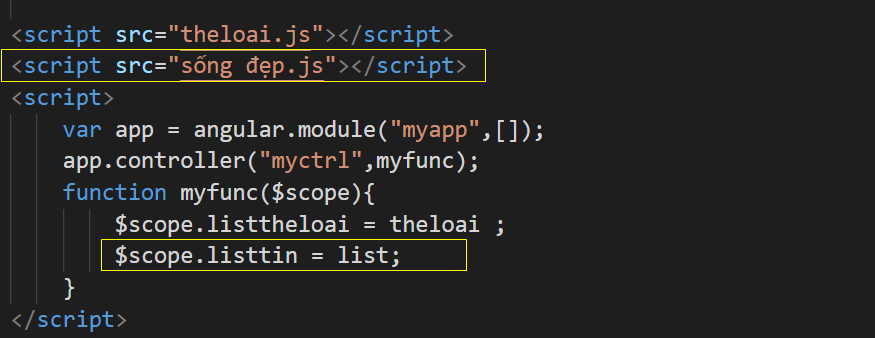


#### Hiện danh sách tin vào cột trái:

1. Mở file **sống đẹp.js** và khai báo biến **list** như sau:



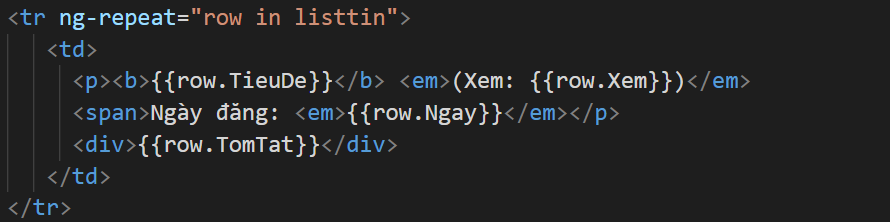
1. Nhúng file **sống đẹp.js** vào trang **bai4.html** và đưa giá trị biến **list** vào $scope:



1. Trong cột trái, chèn 1 table và thực hiện lặp để được như sau



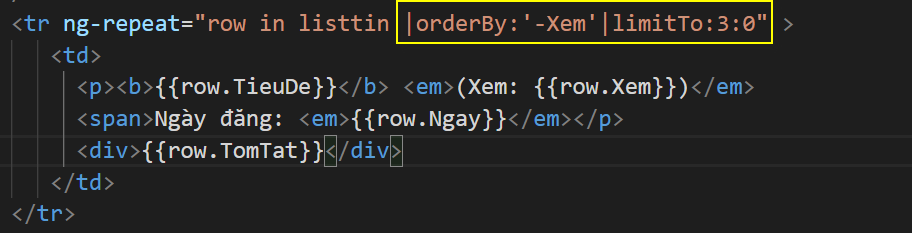
Gợi ý : cách lặp tag tr:



#### Sắp xếp và giới hạn số lần lặp

Khi lặp, có thể sắp xếp item cần lặp và giới hạn số lần hiện các mục.

Thêm code sau vào trong chỉ thị ng-repeat ở trên để sắp xếp theo số lần xem tin giảm dần, và chỉ hiện 3 tin tính từ tin thứ 0



Kết quả:

