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

# **BÁO CÁO ANGULARJS**

**Giảng viên: Phan Thanh Nhuần**

**Môn: CHUYÊN ĐỀ WEB 2** | HKI – [2019 – 2020]



**NHÓM 4:**

**Thành viên:**

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

Mục lục

[**BÁO CÁO ANGULARJS** 1](#_Toc21767449)

[1. Giới thiệu 3](#_Toc21767450)

[2. Các kỹ thuật của đề tài 4](#_Toc21767451)

[2.1. AngularJS Directives 4](#_Toc21767452)

[2.2. Biểu thức AngularJS 8](#_Toc21767453)

[2.3. Số AngularJS 10](#_Toc21767454)

[2.4. Chuỗi AngularJS 10](#_Toc21767455)

[2.5. Đối tượng AngularJS 12](#_Toc21767456)

[2.6. Mảng AngularJS 12](#_Toc21767457)

[2.7. Modules AngularJS 13](#_Toc21767458)

[2.8. AngularJS Controller 17](#_Toc21767459)

[2.9 Controller Methods 17](#_Toc21767460)

[2.10 AngularJS Scopes 18](#_Toc21767461)

[2.11 AngularJS Filters 19](#_Toc21767462)

[2.12 AngularJS Services 19](#_Toc21767463)

[2.13 AngularJS $http 21](#_Toc21767464)

[2.14 JSON 22](#_Toc21767465)

[2.15 AngularJS Tables 22](#_Toc21767466)

[2.16 AngularJS SQL 23](#_Toc21767467)

[2.17 AngularJS DOM 24](#_Toc21767468)

[2.18 AngularJS Events 25](#_Toc21767469)

[3. Ứng dụng của đề tài 28](#_Toc21767470)

[3.1 Ý tưởng ứng dụng demo 28](#_Toc21767471)

[3.2 Database 28](#_Toc21767472)

[3.3 Giao diện User 30](#_Toc21767473)

[3.4 Login 33](#_Toc21767474)

[3.5 Admin 37](#_Toc21767475)

[3.6 Cài đặt ứng dụng chạy demo. 43](#_Toc21767476)

## 1. Giới thiệu

Đề tài mà nhóm chọn là **AngularJS**.

Nói về **AngularJS** thì đây là một mã nguồn mở, một framework cho các ứng dụng web. Được phát triển từ năm 2009, hiện tại được duy trì bởi **Google** và đã ra mắt phiên bản 7.0

Định nghĩa chính thức được đưa ra như sau:

**AngularJS** là một framework có cấu trúc cho ứng dụng web động. Nó cho phép bạn sử dụng **HTML** như là ngôn ngữ mẫu và cho phép bạn mở rộng cú pháp của **HTML** để diễn đạt các thành phần ứng dụng của bạn mọt cách rõ ràng và súc tích. Hai tính năng cốt lõi: **Data** **binding** và **Dependency** **injection** của **AngularJS** loại bỏ phần lớn code mà bạn thường phải viết. Nó xảy ra trong tất cả các trình duyệt, làm cho nó trở thành đói tác lý tưởng của bất kỳ công nghệ Sever nào.

Để học được **AngularJS** bạn cần phải có những kiến thức cơ bản như **Javascript**, **object**, **string** … Việc bạn cơ hiểu biết chuyên sâu về **Javascript** sẽ giúp cho bạn dể dàng học **AngularJS**. Bản chất của **AngularJS** là hoạt động dạng **Single** **Page**, sử dụng **API** để lấy data, cho nên bạn cần biết các kĩ thuật **DHTML**, **Ajax**.

Lịch sử **AngularJS**

* AngularJS version 1.0 đã được phát hành vào năm 2012
* Miško Hevery, một nhân viên của Google, bắt đầu làm việc với AngularJS vào năm 2009.
* Ý tưởng hóa ra rất tốt và dự án hiện được Google hỗ trợ chính thức.

Đặc trưng của **AngularJS**

* Phát triển dựa trên **Javascript**
* Tạo các ứng dụng client-side theo mô hình **MVC**
* Khả năng tương thích cao, tự động xử lý mã Javascript để phù hợp với mỗi trình duyệt
* Mã nguồn mở miễn phí hoàn toàn và được sử dụng rộng rãi.

Các tính năng cơ bản: **Scope**, **Controller**, **Data**-**binding**, **Service**, **Filter**, **Directive**, **Temple**, **Routing**, **MVC** **&** **MVVM**, **Deep** **link**, **Dependency** **Injection**.

Các **Component** chính:

* **ng**-**app**: định nghĩa và liên kết một ứng dụng **AngularJS** tới **HTML**.
* **ng**-**model**: gắn kết giá trị của dữ liệu ứng dụng **AngularJS** đến các điều khiển đầu vào **HTML**.
* **ng**-**bind**: gắn kết dữ liệu ứng dụng **AngularJS** đến các thẻ **HTML**

Ưu điểm của **AngularJS**

* Cung cấp khả năng tạo ra các Single Page Aplication dể dàng.
* Cung cấp khả năng data binding tới **HTML**, khiến cho người dùng cám giác linh hoạt, thân thiện
* Dể dàng Unit Test
* Dể dàng tái sử dụng **Component**
* Giúp lập trình viên viết code ít hơn với nhiều chức năng hơn
* Chạy được trên các loại trình duyệt, tren cả Pc lẫn Mobile.

Nhược điểm

* Không an toàn: được phát triển từ **Javascript** cho nên ứng dụng dược viết bởi **AngularJS** không an toàn. Nên có sự bảo mật và xác thực phái Server sẽ giúp ứng dụng trở nên an toàn hơn.
* Nếu người sử dụng vô hiệu hóa **Javascript** thì sẽ chỉ nhìn thấy trang cơ bản.

## 2. Các kỹ thuật của đề tài

### 2.1. AngularJS Directives

#### **2.1.1 Thuộc tính HTML mở rộng có tiền tố ng-**

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

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

- Lệnh ng-app xác định thành phần gốc của ứng dụng **AngularJS**.

- Lệnh ng-app này sẽ tự động tữ khởi động (tự động khởi chạy) ứng dụng khi trang web được tải.

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

**-** Lệnh ng-init khởi tạo dữ liệu ứng dụng

**-** Lệnh ng-init xác định các giá trị ban đầu cho ứng dụng **AngularJS**.

**-** Thông thường sẽ không sử dụng ng-init, ta sẽ sử dụng một **Controller** hoặc **Modules** thay thế.

##### **2.1.3.1 ng-model Directive**

**-** Lệnh ng-model này liên kết giá trị của các điều khiển HTML (đầu vào, chọn, textarea) với dữ liệu ứng dụng.

**-** Lệnh ng-model này cũng có thể:

**+** Cung cấp xác nah65n loại cho dữ liệu ứng dụng (số, email, bắt buộc).

**+** Cung cấp trạng thái cho dữ liệu ứng dụng (không hợp lệ, bẩn, chạm, lỗi).

**+** Cung cấp các lớp CSS cho các phần tử HTML.

**+** Liên kết các phần tử HTML với các biểu mẫu HTML

**Ví dụ:**

<div ng-app="" ng-init="firstName='John'">

<p>Name: <input type="text" ng-model="firstName"></p>

<p>You wrote: {{ firstName }}</p>

</div>

* Lệnh ng-app này cũng cho AngularJS biết rằng phần tử <div> là "chủ sở hữu" của ứng dụng AngularJS.

**2.1.2 Data binding**

**-** Các {{ firstName }}biểu hiện, trong ví dụ trên, là một AngularJS dữ liệu ràng buộc biểu.

**-** Liên kết dữ liệu trong AngularJS liên kết các biểu thức AngularJS với dữ liệu AngularJS.

**-** {{ firstName }}bị ràng buộc với ng-model="firstName".

**Ví dụ: Hai trường văn bản được liên kết với nhau bằng hai lệnh ng-model**

<div ng-app="" ng-init="quantity=1;price=5">

Quantity: <input type="number" ng-model="quantity">

Costs:    <input type="number" ng-model="price">

Total in dollar: {{ quantity \* price }}

</div>

**2.1.3 Lặp lại các phần tử HTML**

- Lệnh ng-repeat lập lại một phần tử HTML

**Ví dụ:**

<div ng-app="" ng-init="names=['Jani','Hege','Kai']">

<ul>

<li ng-repeat="x in names">

{{ x }}

</li>

</ul>

</div>

* Lệnh ng-repeat này thực sự nhân bản các phần tử HTML một lần cho mỗi mục trong bộ sưu tập.
* Lệnh ng-repeat được sử dụng trên một mảng các đối tượng

**Ví dụ:**

<div ng-app="" ng-init="names=[  
{name:'Jani',country:'Norway'},  
{name:'Hege',country:'Sweden'},  
{name:'Kai',country:'Denmark'}]">

<ul>

<li ng-repeat="x in names">

{{ x.name + ', ' + x.country }}

</li>

</ul>

</div>

#### **2.1.4 Tạo mới Directives**

**-** Ngoài tất cả các chỉ thị AngularJS tích hợp, ta còn có thể tạo các chỉ thị của riêng mình.

**-** Chỉ thị mới được tạo bằng cách sử dụng .directivechức năng.

**-** Để gọi lệnh mới, tạo một phần tử HTML có cùng tên thẻ với lệnh mới.

**-**Khi đặt tên cho lệnh, ta phải sử dụng tên trường hợp lạc đà w3TestDirective, nhưng khi gọi nó, bạn phải sử dụng -tên riêng , w3-test-directive

**Ví dụ:**

<body ng-app="myApp">

<w3-test-directive></w3-test-directive>

<script>

var app = angular.module("myApp", []);

app.directive("w3TestDirective", function() {

return {

template : "<h1>Made by a directive!</h1>"

};

});

</script>

</body>

* Ta có thể gọi một lệnh bằng cách sử dụng:

+ Tên thành phần. Ví dụ : <w3-test-directive></w3-test-directive>

+ Thuộc tính. Ví dụ : <div w3-test-directive></div>

+ Lớp. Ví dụ: <div class="w3-test-directive"></div>

+ Comment. Ví dụ : <!-- directive: w3-test-directive -->

#### **2.1.5 Restrictions**

- Ta có thể hạn chế các chỉ thị của mình chỉ được gọi bằng một số phương thức.

Ví dụ:

Bằng cách thêm một thuộc tính restrict với giá trị “A”, Directive chỉ có thể được gọi bởi các thuộc tính

var app = angular.module("myApp", []);

app.directive("w3TestDirective", function() {

return {

restrict : "A",

template : "<h1>Made by a directive!</h1>"

};

});

### 2.2. Biểu thức AngularJS

- Biểu thức **AngularJS** có thể được viết bên trong dấu ngoặc kép: {{ expression }}

- Các biểu thức AngularJS cũng có thể được viết bên trong một lệnh: ng-bind=” expression”

- **AngularJS** sẽ giải quyết biểu thức và trả về kết quả chính xác nơi biểu thức được viết.

- **Các biểu thức AngularJS** rất giống các **biểu thức JavaScript:** Chúng có thể chứa các ký tự, toán tử và biến. Ví dụ: {{5 + 5}} hoặc {{FirstName + "" + lastName}}

**Ví dụ:**

<!DOCTYPE html>

<html>

<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>

<body>

<div ng-app="">

<p>My first expression: {{ 5 + 5 }}</p>

</div>

</body>  
</html>

* Nếu loại bỏ lệnh ng-app, HTML sẽ hiển thị biểu thức như hiện tại mà không giải quyết nó.
* Có thể viết biểu thức ở bất cứ nơi nào mình thích, AngularJS sẽ chỉ giải quyết biểu thức và trả về kết quả.

### 2.3. Số AngularJS

* Số AngularJS giống như số JavaScript

**Ví dụ:**

<div ng-app="" ng-init="quantity=1;cost=5">

<p>Total in dollar: {{ quantity \* cost }}</p>

</div>

* Sử dụng ng-bind

<div ng-app="" ng-init="quantity=1;cost=5">

<p>Total in dollar: <span ng-bind="quantity \* cost"></span></p>

</div>

### 2.4. Chuỗi AngularJS

- Các chuỗi AngularJS cũng giống như các chuỗi JavaScript

**Ví dụ:**

<div ng-app="" ng-init="firstName='John';lastName='Doe'">

<p>The name is {{ firstName + " " + lastName }}</p>

</div>

**Ví dụ tương tự bằng các sử dụng ng-bind :**

<div ng-app="" ng-init="firstName='John';lastName='Doe'">

<p>The name is <span ng-bind="firstName + ' ' + lastName"></span></p>

</div>

### 2.5. Đối tượng AngularJS

Các đối tượng AngularJS giống như các đối tượng JavaScript:

**Ví dụ:**

<div ng-app="" ng-init="person={firstName:'John',lastName:'Doe'}">

<p>The name is {{ person.lastName }}</p>

</div>

**Ví dụ tương tự bằng cách sử dụng ng-bind :**

<div ng-app="" ng-init="person={firstName:'John',lastName:'Doe'}">

<p>The name is <span ng-bind="person.lastName"></span></p>

</div>

### 2.6. Mảng AngularJS

Các mảng **AngularJS** giống như các mảng **JavaScript**:

**Ví dụ:**

<div ng-app="" ng-init="points=[1,15,19,2,40]">

<p>The third result is {{ points[2] }}</p>

</div>

Ví dụ tương tự bằng cách sử dụng ng-bind:

<div ng-app="" ng-init="points=[1,15,19,2,40]">

<p>The third result is <span ng-bind="points[2]"></span></p>

</div>

Nếu so sánh biểu thức AngularJS với biểu thức Javascript thì giống như các biểu thức JavaScript, 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.

### 2.7. Modules AngularJS

Một mô-đun **AngularJS** định nghĩa một ứng dụng.

Các mô-đun là một thùng chứa cho các phần khác nhau của một ứng dụng.

Các mô-đun là một container cho các bộ điều khiển ứng dụng.

Bộ điều khiển luôn thuộc về một Modules.

* + 1. **Tạo Modules**
  + Một **Modules** được tạo bằng cách sử dụng chức năng **AngularJS** angualr.modul**e**

**Ví dụ:**

<div ng-app="myApp">...</div>

<script>

var app = angular.module("myApp", []);

</script>

* + Tham số "myApp" đề cập đến một yếu tố **HTML** trong đó ứng dụng sẽ chạy.
    1. **Thêm Controller**
  + Thêm một bộ điều khiển vào ứng dụng của bạn và tham khảo bộ điều khiển với ng-controller chỉ thị.

**Ví dụ:**

<div ng-app="**myApp**" ng-controller=**"myCtrl"**>

{{ firstName + " " + lastName }}

</div>

<script>

var app = angular.module(**"myApp"**, []);

app.controller(**"myCtrl"**, function($scope) {

$scope.firstName = "John";

$scope.lastName = "Doe";

});

</script>

#### **2.7.3 Thêm Directive**

- AngularJS có một bộ các chỉ thị tích hợp mà bạn có thể sử dụng để thêm chức năng cho ứng dụng của mình.

- Ngoài ra, bạn có thể sử dụng **Modules** để thêm các **Directive** của riêng bạn vào các ứng dụng của bạn

**Ví dụ:**

<div ng-app="myApp" w3-test-directive></div>

<script>

var app = angular.module("myApp", []);

app.directive("w3TestDirective", function() {

return {

template : "I was made in a directive constructor!"

};

});

</script>

* + 1. **Modules và Controller trong tập tin**
  + Các ứng dụng AngularJS thường đặt mô-đun và bộ điều khiển trong các tệp JavaScript.
  + "myApp.js" chứa định nghĩa mô-đun ứng dụng, trong khi "myCtrl.js" chứa bộ điều khiển.

**Ví dụ:**

<!DOCTYPE html>

<html>

<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>

<body>

<div ng-app="**myApp**" ng-controller="**myCtrl**">

{{ firstName + " " + lastName }}

</div>

<script src="**myApp.js**"></script>

<script src="**myCtrl.js**"></script>

</body>  
</html>

myApp.js

var app = angular.module(**"myApp"**, []);

Tham số [] trong định nghĩa Modules có thể được sử dụng để xác định các Modules phụ thuộc.

Không có tham số [], thì không tạo Modules mới mà truy xuất Modules hiện có

myCtrl.js

app.controller(**"myCtrl"**, function($scope) {  
  $scope.firstName = "John";  
  $scope.lastName= "Doe";  
});

**Khi nào tải thư viện**

Mặc dù các ứng dụng HTML thường đặt các tập lệnh ở cuối <body>phần tử, nhưng nên tải thư viện AngularJS trong <head>hoặc ở đầu của phần tử <body>.

Điều này là do các lệnh gọi angular.modulechỉ có thể được biên dịch sau khi thư viện đã được tải.

**Ví dụ:**

<!DOCTYPE html>  
<html>  
<body>  
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>

<div ng-app="myApp" ng-controller="myCtrl">

{{ firstName + " " + lastName }}

</div>

<script>  
var app = angular.module("myApp", []);

app.controller("myCtrl", function($scope) {

$scope.firstName = "John";

$ scope.lastName = "Doe";

});  
</script>

</body>  
</html>

### 2.8. AngularJS Controller

- Các ứng dụng AngularJS được điều khiển bởi các bộ điều khiển.

- Lệnh **ng-controller** xác định bộ điều khiển ứng dụng.

- Bộ điều khiển là **Đối tượng JavaScript** , được tạo bởi **trình xây dựng đối tượng** JavaScript tiêu chuẩn .

**Ví dụ:**

<div ng-app="myApp" ng-controller="myCtrl">

First Name: <input type="text" ng-model="firstName"><br>

Last Name: <input type="text" ng-model="lastName"><br>

<br>

Full Name: {{firstName + " " + lastName}}

</div>

<script>

var app = angular.module('myApp', []);

app.controller('myCtrl', function($scope) {

$scope.firstName = "John";

$scope.lastName = "Doe";

});

</script>

Giải thích:

Ứng dụng AngularJS được xác định bởi  **ng-app = "myApp"** . Ứng dụng chạy bên trong <div>.

Thuộc tính **ng-controller = "myCtrl"** là một lệnh AngularJS. Nó định nghĩa một bộ điều khiển.

Hàm **myCtrl** là một hàm JavaScript.

AngularJS sẽ gọi bộ điều khiển với đối tượng **$ scope**

Trong AngularJS, $ scope là đối tượng ứng dụng (chủ sở hữu của các biến và hàm ứng dụng).

Bộ điều khiển tạo hai thuộc tính (biến) trong phạm vi ( **FirstName** và **lastName** ).

Các lệnh **ng-model** liên kết các trường đầu vào với các thuộc tính của bộ điều khiển (FirstName và lastName).

### 2.9 Controller Methods

- Một bộ điều khiển cũng có thể có các phương thức (biến và hàm)

**Ví dụ:**

<div ng-app="myApp" ng-controller="personCtrl">

First Name: <input type="text" ng-model="firstName"><br>

Last Name: <input type="text" ng-model="lastName"><br>

<br>

Full Name: {{fullName()}}

</div>

<script>

var app = angular.module('myApp', []);

app.controller('personCtrl', function($scope) {

$scope.firstName = "John";

$scope.lastName = "Doe";

$scope.fullName = function() {

return $scope.firstName + " " + $scope.lastName;

  };

});

</script>

### 2.10 AngularJS Scopes

- Scope là phần ràng buộc giữa HTML (view) và JavaScript (controller).

- Scope là một đối tượng với các thuộc tính và phương thức có sẵn.

- Scope có sẵn cho cả view và controller.

Khi ta tạo một controller trong AngularJS, ta truyền $scope đối tượng làm đối số

**Ví dụ: Các thuộc tính được thực hiện trong controller, có thể được tham chiếu trong khung nhìn:**

<div ng-app="myApp" ng-controller="myCtrl">

<h1>{{carname}}</h1>

</div>

<script>

var app = angular.module('myApp', []);

app.controller('myCtrl', function($scope) {

$scope.carname = "Volvo";

});

</script>

### 2.11 AngularJS Filters

AngularJS cung cấp các bộ lọc để chuyển đổi dữ liệu:

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

**Ví dụ: Bộ orderBy lọc sắp xếp một mảng**

<div ng-app="myApp" ng-controller="namesCtrl">

<ul>

<li ng-repeat="x in names | orderBy:'country'">

{{ x.name + ', ' + x.country }}

</li>

</ul>

</div>

### 2.12 AngularJS Services

- Trong AngularJS, một dịch vụ là một hàm hoặc đối tượng, có sẵn và giới hạn cho ứng dụng AngularJS.

- AngularJS có khoảng 30 services tích hợp. Một trong số đó là $location services.

- Các $location dịch vụ có phương pháp mà trả về thông tin về vị trí của trang web hiện tại

**Ví dụ:** Sử dụng $location dịch vụ trong bộ Controller

var app = angular.module('myApp', []);

app.controller('customersCtrl', function($scope, $location) {

$scope.myUrl = $location.absUrl();

});

Lưu ý rằng $location service được truyền vào Controller dưới dạng đối số. Để sử dụng dịch vụ trong Controller, nó phải được xác định và phụ thuộc.

$http Service

Các $http service là một trong những services sử dụng phổ biến nhất trong các ứng dụng AngularJS. Services đưa ra yêu cầu đến máy chủ và cho phép ứng dụng của bạn xử lý phản hồi.

**Ví dụ:** Sử dụng $http service để yêu cầu dữ liệu từ máy chủ

var app = angular.module('myApp', []);

app.controller('myCtrl', function($scope, $http) {

$http.get("welcome.htm").then(function (response) {

$scope.myWelcome = response.data;

});

});

$timeout Service

Các $timeout service là phiên bản AngularJS của window.setTimeout function.

**Ví dụ: hiển thị một tin nhắn mới sau 2s**

var app = angular.module('myApp', []);

app.controller('myCtrl', function($scope, $timeout) {

$scope.myHeader = "Hello World!";

$timeout(function () {

$scope.myHeader = "How are you today?";

}, 2000);

});

$interval Service

Các $interval dịch vụ là phiên bản AngularJS của window.setInterval function.

**Ví dụ:** hiển thị thời gian mỗi giây

var app = angular.module('myApp', []);

app.controller('myCtrl', function($scope, $interval) {

$scope.theTime = new Date().toLocaleTimeString();  
  $interval(function () {

$scope.theTime = new Date().toLocaleTimeString();

}, 1000);

});

### 2.13 AngularJS $http

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

**Ví dụ:** thực hiện một yêu cầu đơn giản đến máy chủ và hiển thị kết quả trong một tiêu đề

<div ng-app="myApp" ng-controller="myCtrl">

<p>Today's welcome message is:</p>

<h1>{{myWelcome}}</h1>

</div>

<script>

var app = angular.module('myApp', []);

app.controller('myCtrl', function($scope, $http) {

$http.get("welcome.htm")

.then(function(response) {

$scope.myWelcome = response.data;

});

});

</script>

METHODS

Methods .get là một methods tắt của service $http. Có một số methods phím tắt:

.delete()

.get()

.head()

.jsonp()

.patch()

.post()

.put()

Các phương thức trên là tất cả các phím tắt gọi sevices $http

**Ví dụ:**

var app = angular.module('myApp', []);

app.controller('myCtrl', function($scope, $http) {

 $http({

method : "GET",

url : "welcome.htm"

}).then(function mySuccess(response) {

$scope.myWelcome = response.data;

}, function myError(response) {

$scope.myWelcome = response.statusText;

});

});

### 2.14 JSON

- Dữ liệu ta nhận được từ phản hồi dự kiến sẽ định dạng JSON

- JSON là một cách tuyệt vời để vận chuyển dữ liệu và rất dể sử dụng trong AngularJS hoặc bất kỳ Javascript nào khác.

**Ví dụ:** Lệnh ng-repeat này hoàn hảo để lặp qua một mảng

<div ng-app="myApp" ng-controller="customersCtrl">

<ul>

<li ng-repeat="x in myData">

{{ x.Name + ', ' + x.Country }}

</li>

</ul>

</div>

<script>

var app = angular.module('myApp', []);  
app.controller('customersCtrl', function($scope, $http) {  
  $http.get("customers.php").then(function(response) {  
    $scope.myData = response.data.records;  
  });  
});

</script>

### 2.15 AngularJS Tables

#### **2.15.1 Hiển thị dữ diệu trong bảng**

**Ví dụ:**

<div ng-app="myApp" ng-controller="customersCtrl">

<table>  
   <tr ng-repeat="x in names">  
     <td>{{ x.Name }}</td>  
     <td>{{ x.Country }}</td>  
   </tr>  
</table>  
  
</div>  
  
<script>  
var app = angular.module('myApp', []);  
app.controller('customersCtrl', function($scope, $http) {  
  $http.get("customers.php")  
  .then(function (response) {$scope.names = response.data.records;});  
});  
</script>

#### **2.15.2 Sử dụng $even và $odd**

**Ví dụ:**

<table>  
   <tr ng-repeat="x in names">  
    <td ng-if="$odd" style="background-color:#f1f1f1">{{ x.Name }}</td>  
     <td ng-if="$even">{{ x.Name }}</td>  
     <td ng-if="$odd" style="background-color:#f1f1f1">{{ x.Country }}</td>  
     <td ng-if="$even">{{ x.Country }}</td>  
   </tr>  
</table>

### 2.16 AngularJS SQL

- AngularJS hoàn hảo để hiển thi dữ liệu từ SQL. Chỉ cần đảm bảo dữ liệu ở định dạng JSON

**Ví dụ:** tìm nạp dữ liệu từ máy chủ PHP chạy MySQL

<div ng-app="myApp" ng-controller="customersCtrl">

<table>  
  <tr ng-repeat="x in names">  
    <td>{{ x.Name }}</td>  
    <td>{{ x.Country }}</td>  
  </tr>  
</table>  
  
</div>  
  
<script>  
var app = angular.module('myApp', []);  
app.controller('customersCtrl', function($scope, $http) {  
  $http.get("customers\_mysql.php")  
  .then(function (response) {$scope.names = response.data.records;});  
});  
</script>

**Ví dụ:** tìm nạp dữ liệu từ máy chủ ASP.NET chạy SQL

<div ng-app="myApp" ng-controller="customersCtrl">  
  
<table>  
  <tr ng-repeat="x in names">  
     <td>{{ x.Name }}</td>  
     <td>{{ x.Country }}</td>  
   </tr>  
</table>  
  
</div>  
  
<script>  
var app = angular.module('myApp', []);  
app.controller('customersCtrl', function($scope, $http) {  
  $http.get("customers\_sql.aspx")  
  .then(function (response) {$scope.names = response.data.records;});  
});  
</script>

### 2.17 AngularJS DOM

**2.17.1 ng-disabled Directive**

- Lệnh ng-disabled liên kết dữ liệu ứng dụng AngularJS với thuộc tính bị vô hiệu hóa của các phần tử HTML

**Ví dụ:**

<div ng-app="" ng-init="mySwitch=true">  
  
<p>  
<button ng-disabled="mySwitch">Click Me!</button>  
</p>  
  
<p>  
<input type="checkbox" ng-model="mySwitch">Button  
</p>  
  
<p>  
{{ mySwitch }}  
</p>  
  
</div>

Lệnh ng-disabled liên kết dữ liệu ứng dụng mySwitch với thuộc tính disable của nút HTML

Lệnh ng-model liên kết giá trị của phần tử hộp kiểm HTML với giá trị của mySwitch

Nếu giá trị của mySwitch ước tính là true, nút sẽ bị tắt

<p>  
<button disabled>Click Me!</button>  
</p>

Nếu giá trị của mySwitch ước tính là false, nút sẽ không bị tắt

<p>  
<button>Click Me!</button>  
</p>

#### **2.17.2 ng-show Directive**

- Lệnh ng-show hiển thị hoặc ẩn một phần tử html

Ví dụ:

<div ng-app="">  
  
<p ng-show="true">I am visible.</p>  
  
<p ng-show="false">I am not visible.</p>  
  
</div>

#### **2.17.3 ng-hide Directive**

- Lệnh ng-hide ẩn hoặc hiển thị phần tử HTML.

**Ví dụ:**

<div ng-app="">  
  
<p ng-hide="true">I am not visible.</p>  
  
<p ng-hide="false">I am visible.</p>  
  
</div>

### 2.18 AngularJS Events

- Ta có thể thêm trình lắng nghe AngularJS Events vào các thành phần HTML của mình bằng cách sử dụng một hoặc nhiều Directives

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

**Mouse Events**

**Các sự kiện mouse xảy ra khi con trỏ di chuyển qua 1 phần tử, theo thứ tự này:**

ng-mouseover

ng-mouseenter

ng-mousemove

ng-mouseleave

Hoặc khi nhấp vào nút chuột trên một phần tử, theo thứ tự dưới đây

ng-mousedown

ng-mouseup

ng-click

**Ví dụ:**

<div ng-app="myApp" ng-controller="myCtrl">  
  
<h1 ng-mousemove="count = count + 1">Mouse over me!</h1>  
  
<h2>{{ count }}</h2>  
  
</div>  
<script>  
var app = angular.module('myApp', []);  
app.controller('myCtrl', function($scope) {  
  $scope.count = 0;  
});  
</script>

**ng-click Directive**

**Lệnh ng-click xác định mã AngularJS sẽ được thực thi khi phần tử đang được nhấp**

**Ví dụ:**

<div ng-app="myApp" ng-controller="myCtrl">  
  
<button ng-click="count = count + 1">Click me!</button>  
  
<p>{{ count }}</p>  
  
</div>  
<script>  
var app = angular.module('myApp', []);  
app.controller('myCtrl', function($scope) {  
  $scope.count = 0;  
});  
</script>

**$event Object**

**Ta có thể truyền $event object làm đối số khi gọi hàm.**

**$event object chúa đối tượng sự kiện của trình duyệt**

**Ví dụ:**

<div ng-app="myApp" ng-controller="myCtrl">  
  
<h1 ng-mousemove="myFunc($event)">Mouse Over Me!</h1>  
  
<p>Coordinates: {{x + ', ' + y}}</p>  
  
</div>  
<script>  
var app = angular.module('myApp', []);  
app.controller('myCtrl', function($scope) {  
  $scope.myFunc = function(myE) {  
    $scope.x = myE.clientX;  
    $scope.y = myE.clientY;  
  }  
});  
</script>

## 3. Ứng dụng của đề tài

### 3.1 Ý tưởng ứng dụng demo

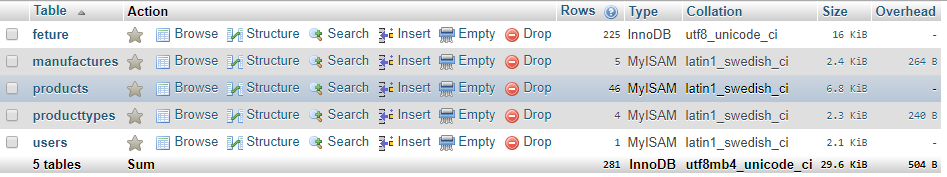
- Sử dụng Framework AngularJS để có thể tạo một ứng dụng website bán hàng với 2 phần giao diện là (user và admin).

- Website này bán những sản phẩm điện tử với 3 thành phần chính là product, product type, manufacture.

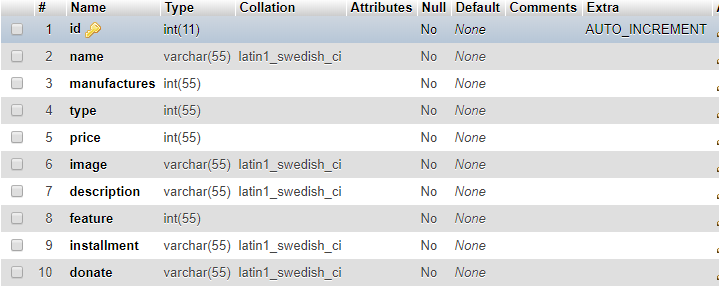
- Sử dụng toàn bộ những kiến thức đã được học trong 2 năm để có thể kết hợp tạo nên 1 ứng dụng hoàn hảo.

### 3.2 Database

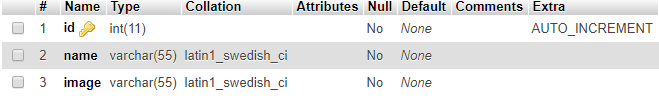
**-** Tạo 1 database tkw2 với 5 table.



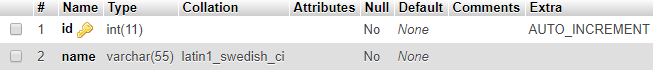
- Table products – lưu giữ những thông tin sản phẩm.



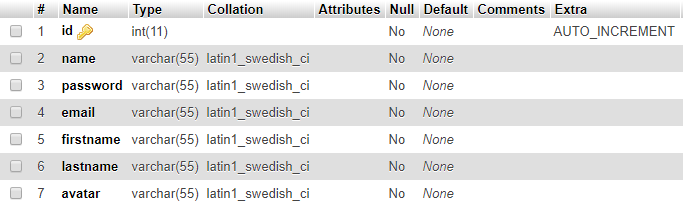
- Table manufactures – lưu trữ thông tin thương hiệu



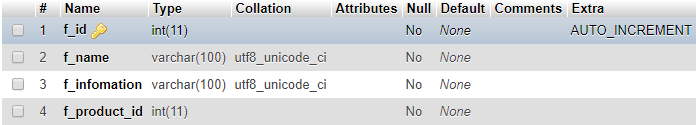
- Table producttypes – lưu trữ thông tin loại sản phẩm



-Table users – lưu trữ thông tin admin



-Table feture – lưu trữ thông tin chi tiết theo sản phẩm



### 3.3 Giao diện User

**-** Giao diện dành cho user có thể tìm kiếm, xem danh sách sản phẩm.

- Các chức năng có AngularJS:

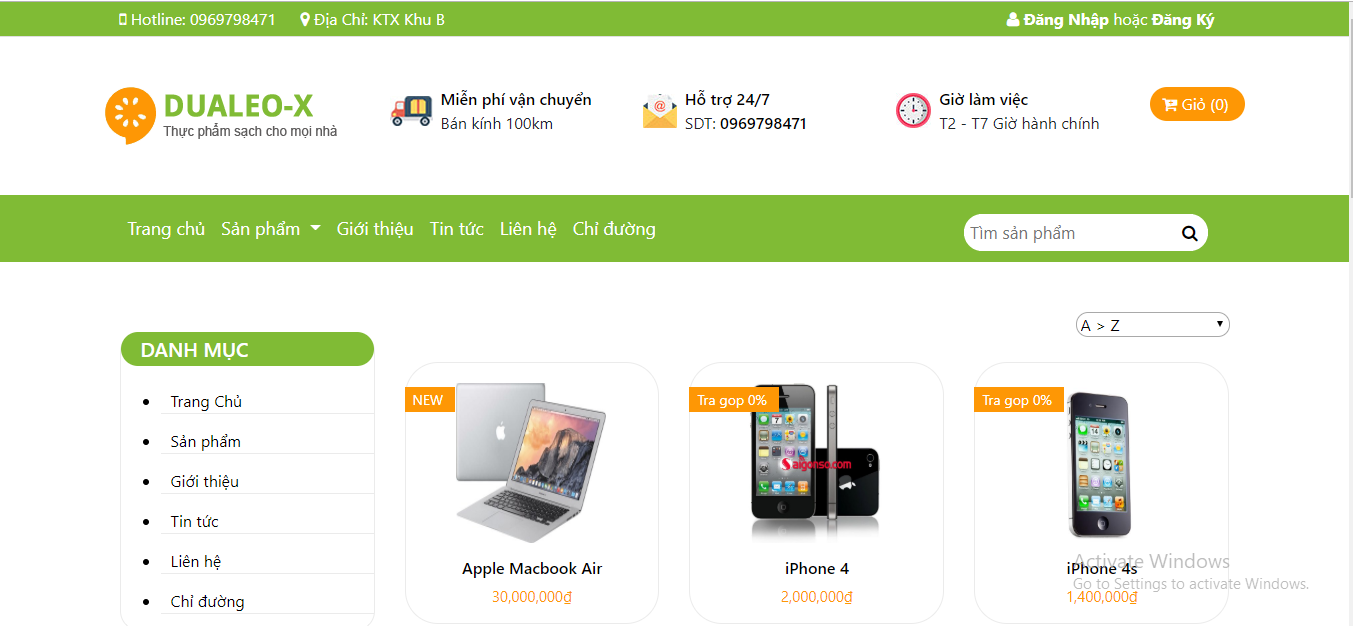
Hiển thị sản phẩm

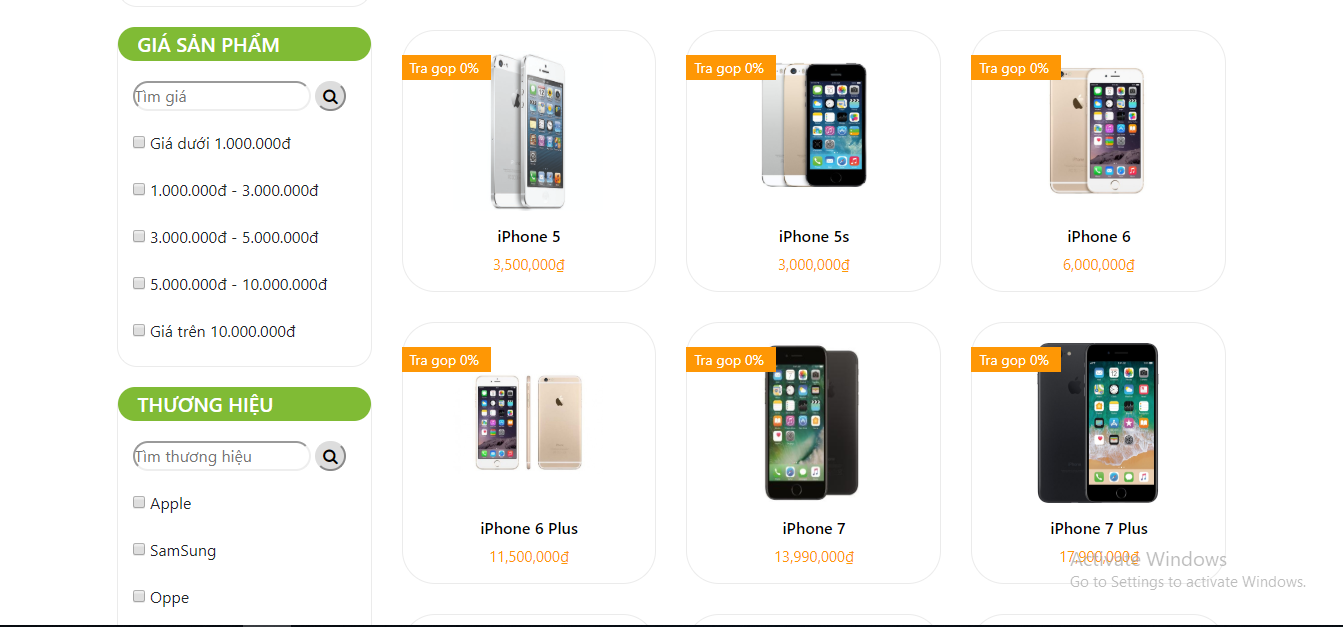
Phân trang – 12 sản phẩm trên 1 trang.

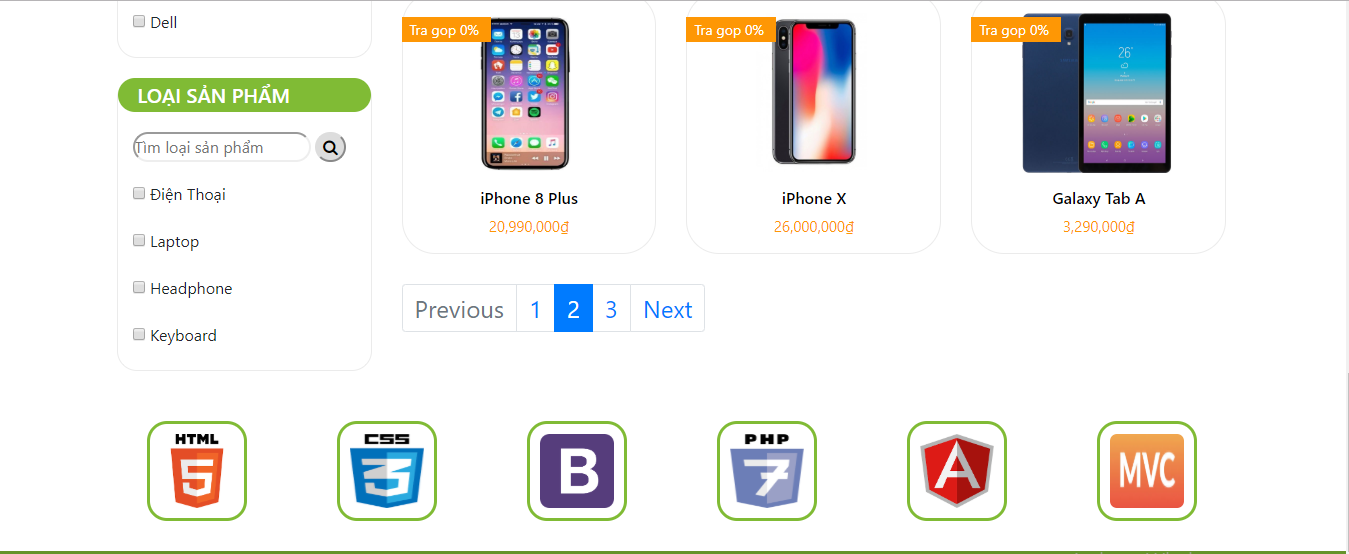
Hiển thị số tiền theo định dạng x,xxx,xxx.

Sắp xếp theo tùy chọn.

Lấy id sản phẩm khi click vào sản phẩm.









#### **3.3.1 Khai báo Modules và Controller**



Index.php

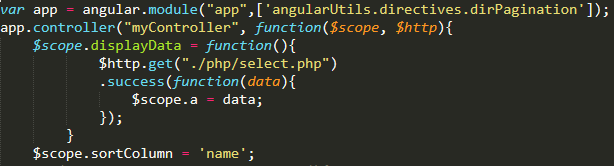


App.js

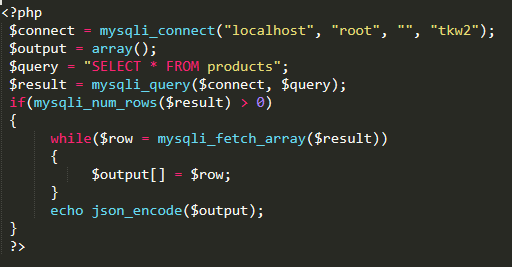
#### **3.3.2 Hiện thị và phân trang từ databse**



**Index.php**



**App.js**



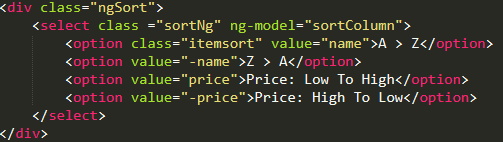
**Select.php**

#### **3.3.3 Hiển thị số tiền với định dạng x,xxx,xxx**



**Index.php**

#### **3.3.4 Sắp xếp theo tùy chọn**





**Index.php**



**App.js**

#### **3.3.5 Lấy id sản phẩm khi ckick vào sản phẩm**



**Index.php**

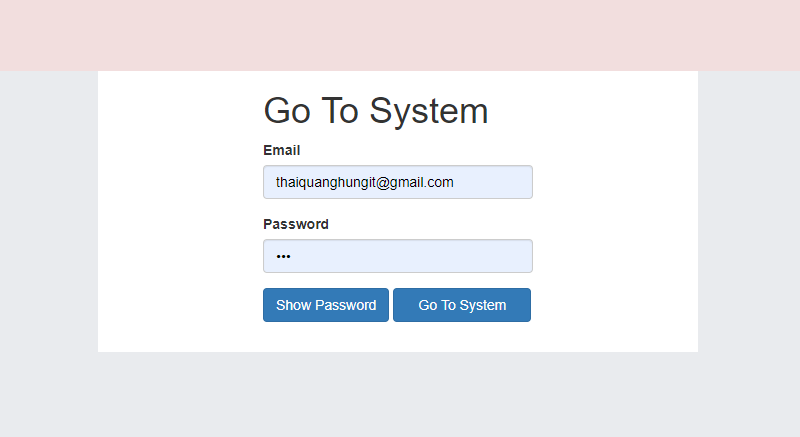
### 3.4 Login

- Màn hình login dùng để kiểm tra xem có phải là admin để có thể truy cập vào trang admin được không.

- Các chức năng có sử dụng AngularJS:

Kiểm tra đăng nhập, thông báo các lỗi.

Hiện/Ẩn mật khẩu.



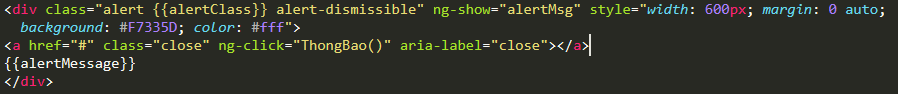
#### **3.4.1 Khai báo Module và Controller**

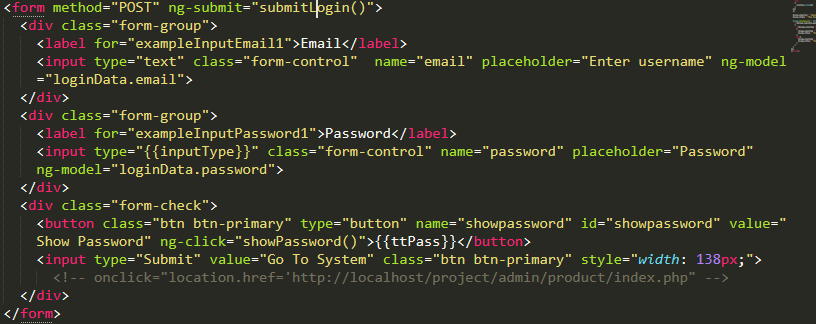




Index.php

#### **3.4.2 Kiểm tra đăng nhập, thông báo các lỗi.**

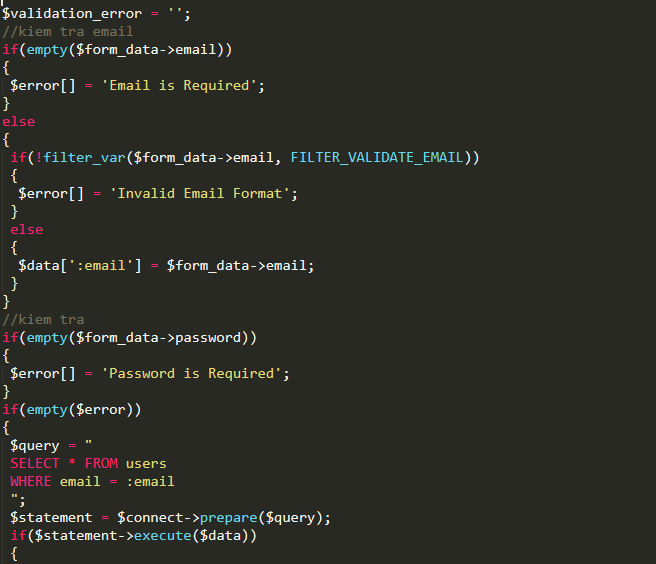


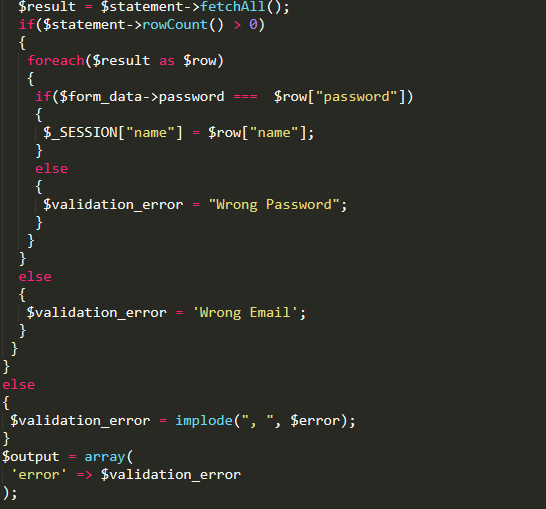


Index.php



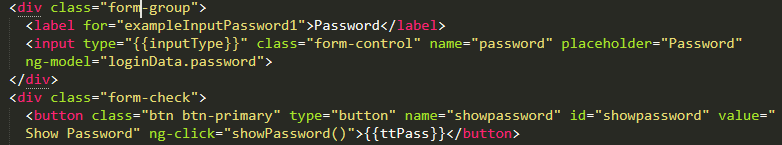
App.js





Login.php

#### **3.4.3 Hiện/Ẩn mật khẩu**



Index.php



App.js

### 3.5 Admin

- Màn hình này giúp cho những người dùng thuộc admin có thể quản lí database của trang web.

- Màn hình quản lí 3 phần với chức năng khá tương đồng nhau: Product, Manufacture, Product type.

- Bảng các chức năng đã thực hiện được.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên Chức năng | Product-type | Manufacture | Product |
| 1 | Hiển thị dữ liệu | X | x | x |
| 2 | Thêm dữ liệu | X | x |  |
| 3 | Sửa dữ liệu | X |  |  |
| 4 | Xóa dữ liệu | X | x | x |
| 5 | Sắp xếp theo title table | X | x | x |
| 6 | Hiển thị theo tùy chọn số dòng | X | x | x |
| 7 | Tìm kiếm trực tiếp | X | x | x |
| 8 | Phân Trang | X | x | x |
| 9 | Upload file |  | x | x |

*(x là những chức năng đã thực hiện được!)*

*-* Các chức năng có sử dụng AngularJS trong **manufacture.**

#### **3.5.1 Khai báo Module và Controller**

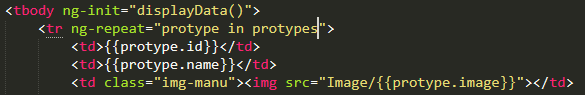


Index.php

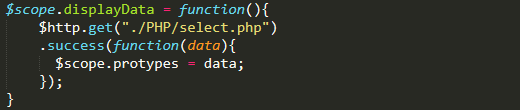


App.js

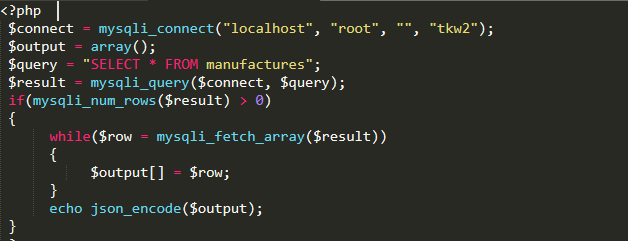
#### **3.5.2 Hiển thị danh sách.**



Index.php

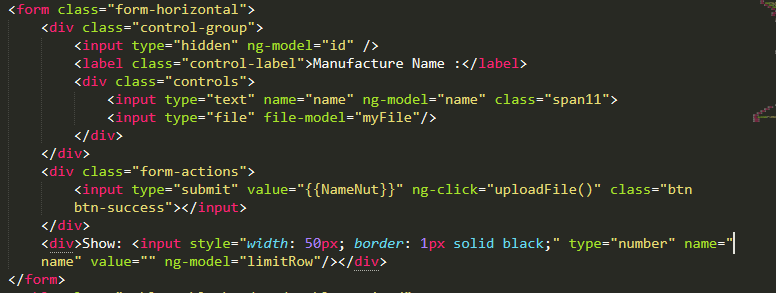


App.js

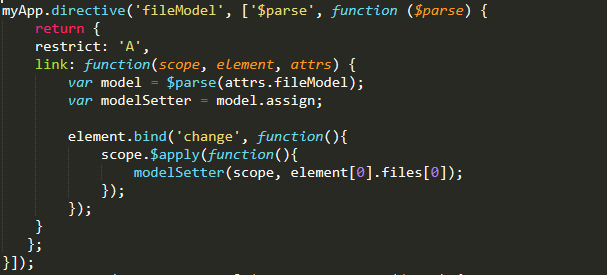


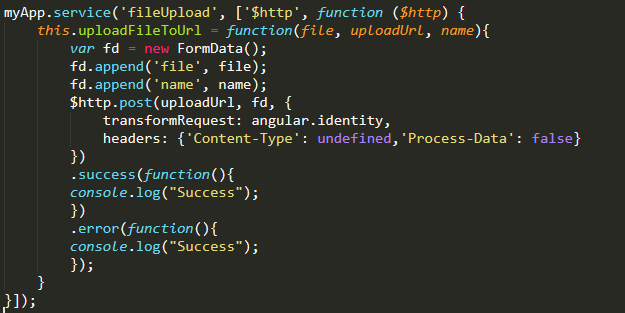
Select.php

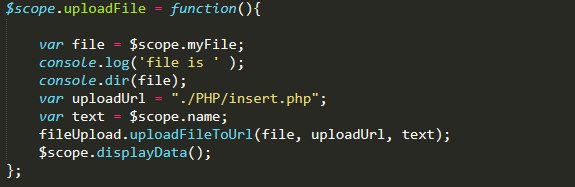
#### **3.5.3 Thêm dữ liệu.**



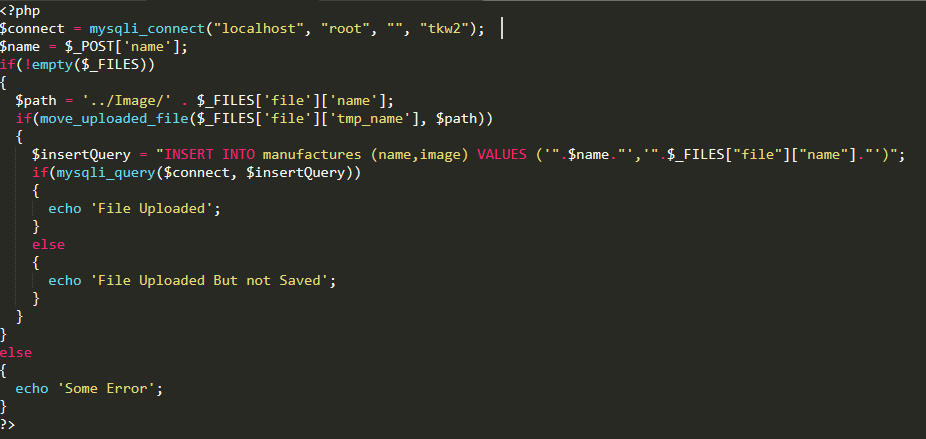
Index.php







App.js

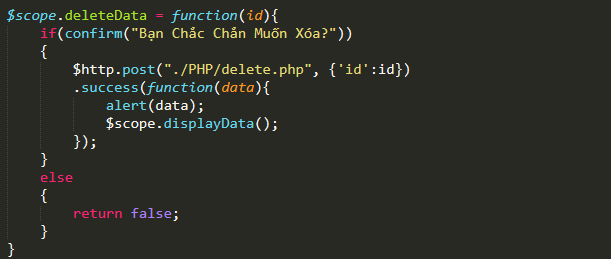


Insert.php

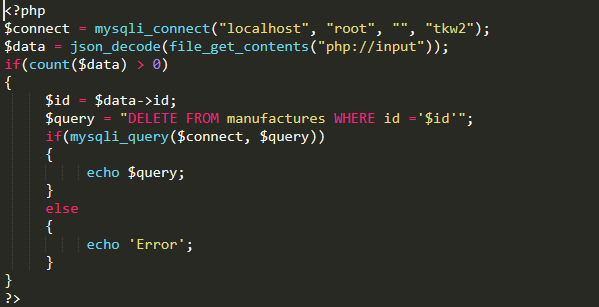
#### **3.5.4 Xóa dữ liệu**



Index.php

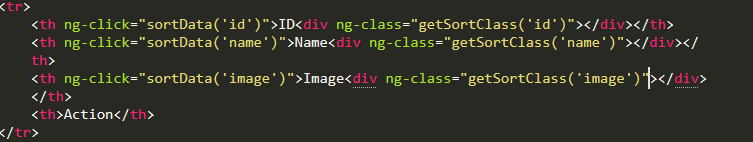


App.js

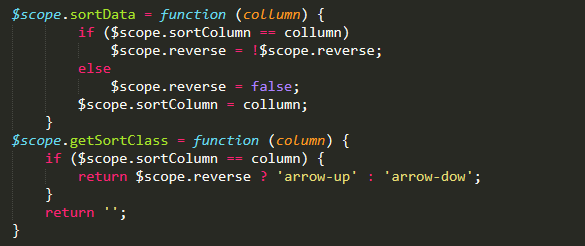


Delete.php

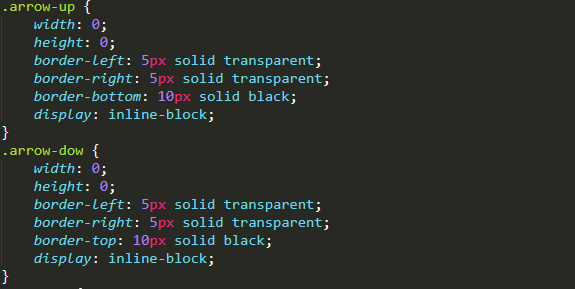
#### **3.5.5 Sắp xếp theo title table**



Index.php



App.js



Styles.css

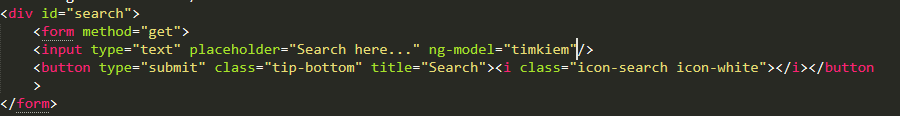
#### **3.5.6 Hiển thị theo tùy chọn số dòng.**





Index.php

#### **3.5.7 Tìm kiếm trực tiếp.**

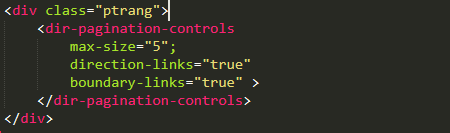




Index.php

#### **3.5.8 Phân trang.**





Index.php

### 3.6 Cài đặt ứng dụng chạy demo.

Chạy ứng dụng xampp, wampp với port Apache 8080.

Đưa project vào htdocs với xampp, www với wampp.

Tạo một database rỗng với tên tkw2, import file tkw2.sql.

Username: [thaiquanghungit@gmail.com](mailto:thaiquanghungit@gmail.com).

Password: 123.