|  |  |  |
| --- | --- | --- |
| 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](#_Toc23923304)

[2. Các kỹ thuật của đề tài 2](#_Toc23923305)

[2.1. AngularJS Directives 2](#_Toc23923306)

[2.2. Biểu thức AngularJS 5](#_Toc23923307)

[2.3. Số AngularJS 6](#_Toc23923308)

[2.4. Chuỗi AngularJS 7](#_Toc23923309)

[2.5. Đối tượng AngularJS 7](#_Toc23923310)

[2.6. Mảng AngularJS 8](#_Toc23923311)

[2.7. Modules AngularJS 8](#_Toc23923312)

[2.8. AngularJS Controller 11](#_Toc23923313)

[2.9 Controller Methods 12](#_Toc23923314)

[2.10 AngularJS Scopes 13](#_Toc23923315)

[2.11 AngularJS Filters 13](#_Toc23923316)

[2.12 AngularJS Services 14](#_Toc23923317)

[2.13 AngularJS $http 15](#_Toc23923318)

[2.14 JSON 17](#_Toc23923319)

[2.15 AngularJS Tables 17](#_Toc23923320)

[2.16 AngularJS SQL 18](#_Toc23923321)

[2.17 AngularJS DOM 19](#_Toc23923322)

[2.18 AngularJS Events 20](#_Toc23923323)

[3. Ứng dụng của đề tài 23](#_Toc23923324)

[3.1 Ý tưởng ứng dụng demo 23](#_Toc23923325)

[3.2 Database 23](#_Toc23923326)

[3.3 Giao diện User 25](#_Toc23923327)

[3.4 Login 28](#_Toc23923328)

[3.5 Admin 31](#_Toc23923329)

[3.6 Cài đặt ứng dụng chạy demo. 37](#_Toc23923330)

**DANH MỤC VÍ DỤ.**

[Ví dụ 1. ng-model Directive 3](#_Toc23923225)

[Ví dụ 2. Data binding 3](#_Toc23923226)

[Ví dụ 3. Ng-repeat 4](#_Toc23923227)

[Ví dụ 4. ng-repeat 2 4](#_Toc23923228)

[Ví dụ 5. Tạo mới Directives 5](#_Toc23923229)

[Ví dụ 6. Biểu thức AngularJS 6](#_Toc23923230)

[Ví dụ 7. Số Angular JS 6](#_Toc23923231)

[Ví dụ 8. Số Angular JS sử dụng ng-bind 6](#_Toc23923232)

[Ví dụ 9. Chuỗi AngularJS 7](#_Toc23923233)

[Ví dụ 10. Chuỗi AngularJS sử dụng ng-bind 7](#_Toc23923234)

[Ví dụ 11. Đối tượng Angular JS 7](#_Toc23923235)

[Ví dụ 12. Đối tượng Angular JS sử dụng ng-bind 7](#_Toc23923236)

[Ví dụ 13. Mảng Angular JS 8](#_Toc23923237)

[Ví dụ 14. Mảng Angular JS sử dụng ng-bind 8](#_Toc23923238)

[Ví dụ 15. Tạo modules 9](#_Toc23923239)

[Ví dụ 16. Thêm Controller 9](#_Toc23923240)

[Ví dụ 17. Thêm Directive 10](#_Toc23923241)

[Ví dụ 18. Modules và Controller trong tập tin 10](#_Toc23923242)

[Ví dụ 19. Khi nào tải thư viện 11](#_Toc23923243)

[Ví dụ 20. AngularJS Controller 12](#_Toc23923244)

[Ví dụ 21. Controller Methods 13](#_Toc23923245)

[Ví dụ 22. AngularJS Scopes 13](#_Toc23923246)

[Ví dụ 23. AngularJS Filters 14](#_Toc23923247)

[Ví dụ 24. Sử dụng $location dịch vụ trong bộ Controller 14](#_Toc23923248)

[Ví dụ 25. Sử dụng $http service để yêu cầu dữ liệu từ máy chủ 15](#_Toc23923249)

[Ví dụ 26. hiển thị một tin nhắn mới sau 2s 15](#_Toc23923250)

[Ví dụ 27. Hiển thị thời gian mỗi giây 15](#_Toc23923251)

[Ví dụ 28. Thực hiện một yêu cầu đơn giản đến máy chủ và hiển thị kết quả trong một tiêu đề 16](#_Toc23923252)

[Ví dụ 29. Ví dụ methods GET 16](#_Toc23923253)

[Ví dụ 30. Lệnh ng-repeat hoàn hảo để lặp qua một mảng 17](#_Toc23923254)

[Ví dụ 31. Hiển thị dữ diệu trong bảng 18](#_Toc23923255)

[Ví dụ 32. Sử dụng $even và $odd 18](#_Toc23923256)

[Ví dụ 33. tìm nạp dữ liệu từ máy chủ PHP chạy MySQL 18](#_Toc23923257)

[Ví dụ 34. Tìm nạp dữ liệu từ máy chủ ASP.NET chạy SQL 19](#_Toc23923258)

[Ví dụ 35. ng-disabled Directive 19](#_Toc23923259)

[Ví dụ 36. ng-show Directive 20](#_Toc23923260)

[Ví dụ 37. ng-hide Directive 20](#_Toc23923261)

[Ví dụ 38. Mouse Events 22](#_Toc23923262)

[Ví dụ 39. ng-click Directive 22](#_Toc23923263)

[Ví dụ 40. $event Object 23](#_Toc23923264)

**DANH MỤC HÌNH ẢNH**

No table of figures entries found.

## 1. Giới thiệu

**AngularJS** là một framework được viết bởi ngôn ngữ JavaScript vào năm 2009.

Được tạo bởi 2 nhân viên cũ của Google và phát triển bởi Google.

Trước khi làm việc với AngularJS bạn phải có kiến thức cơ bản đến chuyên sâu HTML, CSS, JavaScript, AJAX. 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**.

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

## 2. Các kỹ thuật của đề tài

### 2.1. AngularJS Directives

#### **2.1.1 Thuộc tính HTML mở rộng có phân đầu là ng-**

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

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

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

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

**-** Lệnh ng-model liên kết giá trị giữa phần HTML và phần AngularJS.

**-** Lệnh ng-model này cũng có thể:

**+** Cung cấp xác nhận 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.

**Ví dụ:**

<div ng-app="" ng-init="firstName=Hung">

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

<p>You wrote: {{ firstName }}</p>

</div>

Ví dụ 1. ng-model Directive

**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>

Ví dụ 2. Data binding

**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>

Ví dụ 3. Ng-repeat

* 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>

Ví dụ 4. ng-repeat 2

#### **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 .directive chức 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>

Ví dụ 5. Tạo mới Directives

* 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>

Ví dụ 6. Biểu thức AngularJS

* 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>

Ví dụ 7. Số Angular JS

* 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>

Ví dụ 8. Số Angular JS sử dụng ng-bind

### 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 9. Chuỗi AngularJS

**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>

Ví dụ 10. Chuỗi AngularJS sử dụng ng-bind

### 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 11. Đối tượng Angular JS

**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>

Ví dụ 12. Đối tượng Angular JS sử dụng ng-bind

### 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 13. Mảng Angular JS

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>

Ví dụ 14. Mảng Angular JS sử dụng ng-bind

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** angular.modul**e**

**Ví dụ:**

<div ng-app="myApp">...</div>

<script>

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

</script>

Ví dụ 15. Tạo modules

* + 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>

Ví dụ 16. Thêm Controller

#### **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>

Ví dụ 17. Thêm Directive

* + 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"**, []);

Ví dụ 18. Modules và Controller trong tập tin

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>

Ví dụ 19. Khi nào tải thư viện

### 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>

Ví dụ 20. AngularJS Controller

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>

Ví dụ 21. Controller Methods

### 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>

Ví dụ 22. AngularJS Scopes

### 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>

Ví dụ 23. AngularJS Filters

### 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();

});

Ví dụ 24. Sử dụng $location dịch vụ trong bộ Controller

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

Ví dụ 25. Sử dụng $http service để yêu cầu dữ liệu từ máy chủ

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

Ví dụ 26. hiển thị một tin nhắn mới sau 2s

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);

});

Ví dụ 27. Hiển thị thời gian mỗi giây

### 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>

Ví dụ 28. Thực hiện một yêu cầu đơn giản đến máy chủ và hiển thị kết quả trong một tiêu đề

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;

});

});

Ví dụ 29. Ví dụ methods GET

### 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>

Ví dụ 30. Lệnh ng-repeat hoàn hảo để lặp qua một mảng

### 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>

Ví dụ 31. Hiển thị dữ diệu trong bảng

#### **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>

Ví dụ 32. Sử dụng $even và $odd

### 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 33. tìm nạp dữ liệu từ máy chủ PHP chạy MySQL

**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>

Ví dụ 34. Tìm nạp dữ liệu từ máy chủ ASP.NET chạy SQL

### 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>

Ví dụ 35. ng-disabled Directive

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>

Ví dụ 36. ng-show Directive

#### **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>

Ví dụ 37. ng-hide Directive

### 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>

Ví dụ 38. Mouse Events

**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>

Ví dụ 39. ng-click Directive

**$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>

Ví dụ 40. $event Object

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

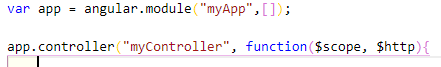
- Sử dụng Framework AngularJS và toàn bộ kiến thức đã học được trong 2 năm để có thể tạo những module hoàn hảo cho dự án phát triển web 2.

- 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**.



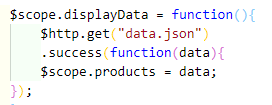
Hình 1. Khởi tạo module và controller.



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

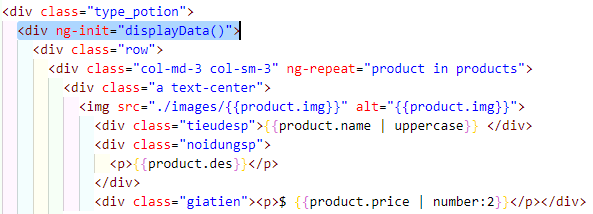
1. Module 04.

* 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 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.

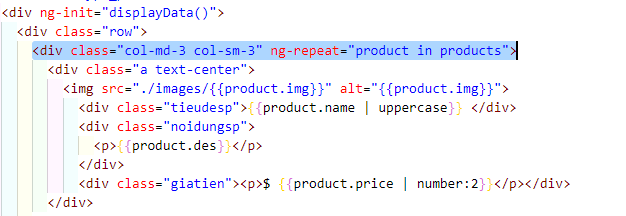


Hình 3.Function trả về dữ liệu là products.

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.



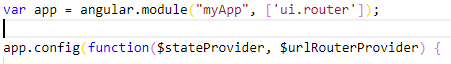
Hình 8. Định dạng kiển hiển thị của giá tiền.



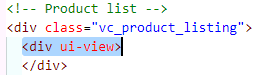
Hình 9. Kết quả của module 04.

1. Module 06

* Sử dụng 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.



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



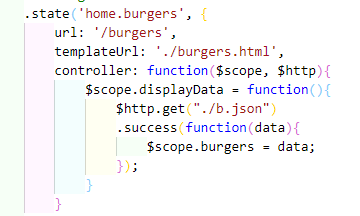
Hình 11. Sử dụng ui-view để khai báo chổ thay đổi template.



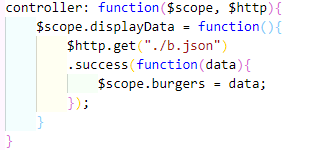
Hình 12. Sử dụng ui-sref để biết sử dụng template nào để thay thế.



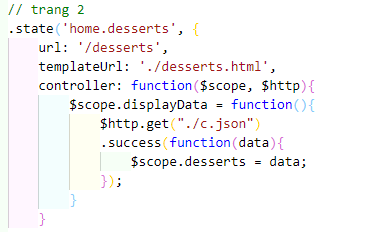
Hình 13.Khai báo template nào được sử dụng mặc định.



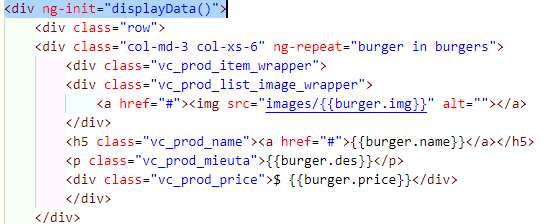
Hình 14.Cài đặt cho từng template.



Hình 15. Function displayData trả về dữ liệu burgers.



Hình 16. Function displayData trả về dữ liệu desserts



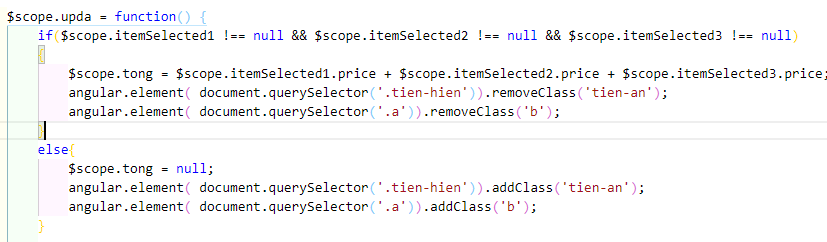
Hình 17. Khai báo ng-init="displayData()" để khởi tạo dữ liệu.



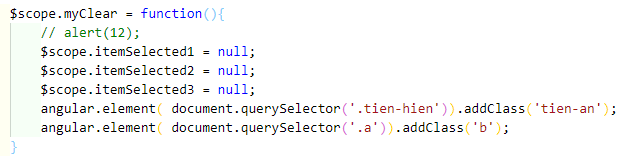
Hình 18. ng-repeat để lặp hiển thị sản phẩm.

1. Module 08
2. Module 11

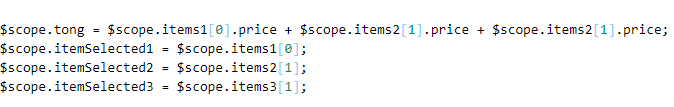
* Sử dụng function upda() và ng-change để cập nhật giá tiền sau khi chọn option của sản phẩm.
* Sử dụng function myClear() để xóa các option đã chọn.
* Sử dụng ng-model và data-ng-option để đổ dữ liệu từ file json ra option.
* Sử dụng addClass và removeClass để ẩn hiện giá tiền.



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

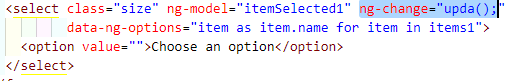


Hình 20. Function myClear - Xóa tất cả các option đã chọn và ẩn đi giá tiền.

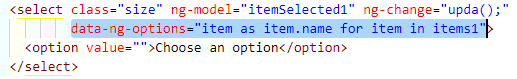


Hình 21. Sét các option mặc định.  


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