**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC DUY TÂN**



**KHÓA LUẬN CỬ NHÂN CÔNG NGHỆ THÔNG TIN**

Tên đề tài:

**XÂY DỰNG WEBSITE THÔNG TIN NGÀNH CÔNG NGHỆ THÔNG TIN SỬ DỤNG MEAN**

**Giảng viên hướng dẫn: ThS. Phạm An Bình**

**Danh sách sinh viên:**

1. **Trương Thành Nam**

**Mã SV: 1921140806**

1. **Trương Phương Tường Vy**

**Mã SV: 1920129099**

1. **Nguyễn Lưu Vũ**

**Mã SV: 1921123170**

1. **Huỳnh Ngọc Minh Trí**

**Mã SV: 1921123209**

**5**

**Đà Nẵng, 5/2017**

**MỤC LỤC**

[1. LÝ DO CHỌN ĐỀ TÀI 10](#_Toc476844893)

[2. Ý TƯỞNG CỦA ĐỂ TÀI 10](#_Toc476844894)

[3. MỤC TIÊU VÀ PHẠM VI ĐỀ TÀI 10](#_Toc476844895)

[4. ĐỐI TƯỢNG VÀ PHƯƠNG PHÁP NGHIÊN CỨU 11](#_Toc476844896)

[5. BỐ CỤC ĐỀ TÀI 11](#_Toc476844897)

[CHƯƠNG 1: GIỚI THIỆU VỀ MEAN 12](#_Toc476844898)

[1.1. GIỚI THIỆU VỀ MEAN 12](#_Toc476844899)

[1.2. GIỚI THIỆU VỀ MONGODB 12](#_Toc476844900)

[1.2.1. Giới thiệu về NoSQL 12](#_Toc476844901)

[1.2.2. Giới thiệu về MongoDB 14](#_Toc476844902)

[1.2.3. Cách cài đặt mongoDB 17](#_Toc476844903)

[1.2.4. Các thao tác cơ bản trên mongoDB 18](#_Toc476844904)

[1.3. GIỚI THIỆU VỀ ANGULARJS 21](#_Toc476844905)

[1.3.1. AngularJS là gì? 21](#_Toc476844906)

[1.3.2. Các đặc tính của AngularJS 21](#_Toc476844907)

[1.3.3. Các nội dung khi nghiên cứu AngularJS 22](#_Toc476844908)

[1.3.4. Cách cài đặt 23](#_Toc476844909)

[1.3.5. Giới thiệu một vài nội dung trong AngularJS 23](#_Toc476844910)

[1.3.6. Ưu điểm và nhược điểm của AngularJS 29](#_Toc476844911)

[1.3.7. Two-way Binding trong AngularJs 29](#_Toc476844912)

[1.4. GIỚI THIỆU VỀ NODEJS 30](#_Toc476844913)

[1.4.1. Giới thiệu về nodeJS 30](#_Toc476844914)

[1.4.2. Các đặc điểm của NodeJS 31](#_Toc476844915)

[1.4.3. Cách cài đặt 31](#_Toc476844916)

[1.4.4. Cách cài đặt các module nodeJS và Cách khai báo module trong nodejs 31](#_Toc476844917)

[1.4.5. Các module hay dùng trong nodeJS 32](#_Toc476844918)

[1.5. GIỚI THIỆU VỀ EXPRESSJS 33](#_Toc476844919)

[1.5.1. Giới thiệu về express 33](#_Toc476844920)

[1.5.2. Các đặc điểm của express 33](#_Toc476844921)

[1.5.3. Cách cài đặt express và chương trình hello world 33](#_Toc476844922)

[1.6. GIỚI THIỆU VỀ FRAMEWORK MEAN.IO VÀ CÁCH CÀI ĐẶT 34](#_Toc476844923)

[1.6.1. Giới thiệu MEAN.IO 34](#_Toc476844924)

[1.6.2. Cách cài đặt mean.io 34](#_Toc476844925)

[1.6.3. Các lệnh cơ bản với mean.io 35](#_Toc476844926)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 67](#_Toc476844973)

[TÀI LIỆU THAM KHẢO 68](#_Toc476844974)

**DANH MỤC VIẾT TẮT**

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Từ viết tắt |  | Ý nghĩa |
| 1 | CSDL | Cơ sở dữ liệu | Cơ sở dữ liệu |
| 2 | SPA | Single Page Application | Ứng dụng web mà thao tác chỉ trên 1 trang duy nhất |
| 3 | RDBMS | Relational Database Management System | Hệ thống quản lý cơ sở dữ liệu quan hệ |
| 4 | JSON | JavaScript Object Notation | Định dạng hoán vị đơn vị dữ liệu nhanh |

**DANH MỤC BẢNG BIỂU**

[Bảng 1.1. Các thao tác trên mongoDB 18](#_Toc476815069)

[Bảng 1.2. Các thành phần của AngularJS 22](#_Toc476815070)

[Bảng 1.3. Các Directive của AngularJS 24](#_Toc476815071)

[Bảng 1.4. Các Filter của AngularJS 28](#_Toc476815072)

[Bảng 1.5. Các module hay dung trong nodeJS 32](#_Toc476815073)

[Bảng 1.6. Cấu trúc thư mục project với mean.io 35](#_Toc476815074)

[Bảng 1.7. Các lệnh thao tác cơ bản với mean.io 36](#_Toc476815075)

**LỜI MỞ ĐẦU**

## 1. LÝ DO CHỌN ĐỀ TÀI

Theo nhu cầu học tập và tìm hiểu của các bạn đam mê công nghệ, một website đem lại các bài viết về nhiều chủ đề công nghệ mới hay một loạt bài viết về một chủ đề đang nổi trội của làng công nghệ với các hướng dẫn chi tiết về các bước thực hiện, xây dựng, giải thích,.... sẽ rất thú vị. Vì vậy nhóm chúng em quyết định xây dựng một website để mọi người cùng học tập và chia sẻ kiến thức mới về công nghệ.

## 2. Ý TƯỞNG CỦA ĐỂ TÀI

Tuy website chia sẻ bài viết tuy không còn mới nhưng công nghệ thì không bao giờ cũ. Tận dụng những công nghệ mới của NodeJS, Angular, MongoDB, ... để xây dựng một website chia sẻ những công nghệ mới này và trong tương lai xa hơn nữa.

## 3. MỤC TIÊU VÀ PHẠM VI ĐỀ TÀI

* Mục tiêu của đề tài:
* Nghiên cứu MEAN.
* Xây dựng website sử dụng MEAN.
* Phạm vi của đề tài: Ý tưởng, Phân tích thiết kế hệ thống, và cài đặt website trên MEAN với các chức năng:
* Đăng nhập (có thể dùng tài khoản mạng xã hội)
* Đăng ký (có thể dùng tài khoản mạng xã hội)
* Xem bài viết mới, được xem nhiều, đánh giá cao, theo danh mục, seri
* Tìm kiếm bài viết theo tên, tác giả, danh mục, lược xem, đánh giá.
* Bình luận
* Các chức năng quản lý cho admin:
* Quản lý Người dùng
* Quản lý Quyền
* Quản lý Bài viết
* Quản lý Danh mục

## 4. ĐỐI TƯỢNG VÀ PHƯƠNG PHÁP NGHIÊN CỨU

* Đối tượng nghiên cứu: MEAN.
* Phương pháp nghiên cứu:
* Phân tích, tổng hợp tài liệu.
* Tham khảo các website trực tuyến.
* Thực hiện xây dựng website.

## 5. BỐ CỤC ĐỀ TÀI

* Lời mở đầu: Lý do chọn đề tài, ý tưởng, mục đích đề tài, đối tượng và phương pháp nghiên cứu.
* Chương 1: Giới thiệu về MEAN và cách cài đặt.
* Chương 2: Phân tích và thiết kế hệ thống website.
* Chương 3: Cài đặt hệ thống.
* Kết luận và hướng phát triển.

# CHƯƠNG 1: GIỚI THIỆU VỀ MEAN

## GIỚI THIỆU VỀ MEAN

MEAN hay MEAN Stack là sự kết hợp hài hòa giữa cơ sở dữ liệu NoSQL mongoDB, Framework phía client AngularJS, server nodeJS và web application expressJS.

MEAN Stack là giải pháp hoàn hảo để xây dựng một mô hình SPA hoàn chỉnh nhất.

## GIỚI THIỆU VỀ MONGODB

### Giới thiệu về NoSQL

#### Khái niệm

NoSQL viết tắt của None-Relational SQL hay Not-Only SQL.

NoSQL là một khái niệm chỉ về một lớp các hệ cơ sở dữ liệu không sử dụng mô hình quan hệ (RDBMS).

NoSQL được phát triển trên Javascript Framework với kiểu dữ liệu là JSON và dạng dữ liệu theo kiểu key và value (1 đặc trưng về dữ liệu trong JSON). NoSQL ra đời như là 1 mảnh vá cho những khuyết điểm và thiếu sót cũng như hạn chế của mô hình dữ liệu quan hệ RDBMS về tốc độ, tính năng, khả năng mở rộng, memory cache,…

NoSQL ra đời năm 1998 bởi Carlo Strozzi khi ông lập mới một hệ cơ sở dữ liệu quan hệ mã nguồn mở nhanh và nhẹ không liên quan đến SQL. Vào năm 2009, Eric Evans, nhân viên của Rackspace giới thiệu lại thuật ngữ NoSQL khi Johan Oskarsson của last.fm muốn tổ chức một hội thảo về cơ sở dữ liệu nguồn mở phân tán. Thuật ngữ NoSQL đánh dấu bước phát triển của thế hệ CSDL mới: phân tán (distributed) và không ràng buộc (non-relational).

#### Đặc điểm

* NoSQL lưu trữ dữ liệu của mình theo dạng cặp giá trị “key – value”. Sử dụng số lượng lớn các node để lưu trữ thông tin.
* Chấp nhận dữ liệu bị trùng lặp do một số node sẽ lưu cùng thông tin giống nhau
* Một truy vấn sẽ được gửi tới nhiều máy cùng lúc, do đó khi một máy nào đó không phục vụ được sẽ không ảnh hưởng lắm đến chất lượng trả về kết quả
* Phi quan hệ – không có ràng buộc nào cho việc nhất quán dữ liệu
* Tính nhất quán không theo thời gian thực: Sau mỗi thay đổi CSDL, không cần tác động ngay đến tất cả các CSDL liên quan mà được lan truyền theo thời gian.

#### Các dạng NoSQL cơ bản

1. **Key – value data stores**

Dữ liệu lưu dưới dạng cặp key– value. Giá trị được truy xuất thông qua key.

* **Ví dụ:** Redis, Dynomite, Project Voldemort.
* **Thường cho: Các ứng dụng đưa nội dung vào bộ nhớ đệm**
* **Ưu điểm:** Tìm kiếm rất nhanh
* **Nhược điểm:** Lưu dữ liệu không theo khuôn dạng nhất định

1. **Column-based – Tabular**

Cơ sở dữ liệu tổ chức dưới dạng các bảng. Gần giống với mô hình RDBMS. Tuy nhiên, Chúng lưu dữ liệu bởi các cột chứ không phải bằng các dòng. Nó khá thích hợp với để hiển thị bằng các phần mềm quản lý kho dữ liệu

* **Ví dụ:**Apache Hbase, Apache Cassandra, Hypertable
* **Thường cho:**các hệ phân tán file
* **Ưu điểm:**Tìm kiếm nhanh, Phân tán dữ liệu tốt
* **Nhược điểm:**Hỗ trợ được với rất ít phần mềm

1. **Document-based**

Dữ liệu( bán cấu trúc hay semi-structured) được lưu trữ và tổ chức dưới dạng một tập hợp các document. Các document này linh hoạt, mỗi document có một tập nhiều trường.

* **Ví dụ :**Apache CouchDB và MongoDB
* **Thường cho:** Các ứng dụng chạy trên nền tảng web
* **Ưu điểm:**Dùng khi dữ liệu nguồn không được mô tả đầy đủ
* **Nhược điểm:**Hiệu năng truy vấn, Không có cú pháp chuẩn cho câu truy vấn dữ liệu

1. **Graph-based data-stores**

Những CSDL này áp dụng lý thuyết đồ thị trong khoa học máy tính để lưu trữ và truy xuất dữ liệu. Chúng tập trung vào tính rời rạc giữa các phần dữ liệu. Các phần tử đơn vị dữ liệu được biểu thị như một nút và liên kết với các thành phần khác bằng các cạnh.

* **Ví dụ :** Neo4j, InfiniteGraph, DEX
* **Thường cho:**Mạng xã hội, Hệ trợ giúp
* **Ưu điểm:** Ứng dụng các thuật toán trên đồ thị như Đường đi ngắn nhất, liên thông,…
* **Nhược điểm:**Phải duyệt nội bộ đồ thị, để trả lời lại các truy vấn. Không dễ để phân tán.

### Giới thiệu về MongoDB

#### Khái niệm

MongoDB (bắt nguồn từ “humongous”) là một hệ cơ sở dữ liệu NoSQL mã nguồn mở.

Thay cho việc lưu trữ dữ liệu vào các bảng có quan hệ với nhau như truyền thống, MongoDB lưu các dữ liệu cấu trúc dưới dạng giống với JSON và gọi tên là BSON. Dự án được bắt đầu triển khai vào tháng 10 năm 2007 bởi 10gen trong khi công ty này đang xây dựng một nền tảng như là dịch vụ (Platform as a Service) giống như Google App Engine. Phải đến năm 2009, dự án này được tách độc lập. Hệ thống có thể chạy trên Windows, Linux, OS X và Solaris.

#### Ưu điểm của MongoDB

* Dễ học, có một số nét khá giống với CSDL quan hệ – Quản lý bằng command line hoặc bằng GUI như RockMongo hoặc phpMoAdmin
* Linh động, không cần phải định nghĩa cấu trúc dữ liệu trước khi tiến hành lưu trữ nó => rất tốt khi ta cần làm việc với các dạng dữ liệu không có cấu trúc.
* Khả năng mở rộng tốt (distributed horizontally), khả năng cân bằng tải cao, tích hợp các công nghệ quản lý dữ liệu vẫn tốt khi kích thước và thông lượng trao đổi dữ liệu tăng.
* Miễn phí

#### Kiến trúc tổng quát

Một MongoDB Server sẽ chứa nhiều database. Mỗi database lại chứa một hoặc nhiều colection. Đây là một tập các documents, về mặt logic thì chúng gần tương tự như các table trong CSDL quan hệ. Tuy nhiên, điểm hay ở đây là ta không cần phải định nghĩa trước cấu trúc của dữ liệu trước khi thao tác thêm, sửa dữ liệu… Một document là một đơn vị dữ liệu – một bản ghi (không lớn hơn 16MB). Mỗi chúng lại chứa một tập các trước hoặc các cặp key – value. Key là một chuỗi ký tự, dùng để truy xuất giá trị dạng : string, integer, double,…

Ví dụ về mongoDB document

{

"\_id" : ObjectId("58400390a313032a34831238"),

"name" : "Java",

"slug" : "java",

"description" : "đây là java",

"keywords" : "java",

"parent\_id" : null,

"status" : "show",

"created" : ISODate("2016-12-01T11:03:44.030Z"),

"\_\_v" : 0

}

{

"\_id" : ObjectId("584004e8a313032a34831239"),

"name" : "C#",

"slug" : "C",

"description" : "<p>đ&acirc;y l&agrave; c#</p>\n",

"keywords" : "c#",

"parent\_id" : null,

"status" : "hide",

"created" : ISODate("2016-12-01T11:09:28.109Z"),

"\_\_v" : 0

}

Cấu trúc có vẻ khá giống JSON, tuy nhiên, khi lưu trữ document này ra database, MongoDB sẽ serialize dữ liệu thành một dạng mã hóa nhị phân đặc biệt – BSON. Ưu điểm của BSON là hiệu quả hơn các dạng format trung gian như XML hay JSON cả hệ tiêu thụ bộ nhớ lẫn hiệu năng xử lý. BSON hỗ trợ toàn bộ dạng dữ liệu mà JSON hỗ trợ (string, integer, double, Boolean, array, object, null) và thêm một số dạng dữ liệu đặc biệt như regular expression, object ID, date, binary, code

#### So sánh giữa RDBMS và MongoDB

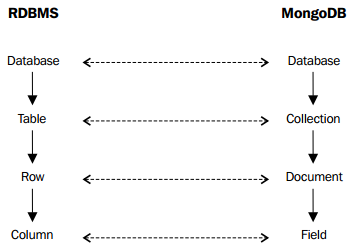


Table: users

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **id** | **username** | **password** | **email** | **age** |
| 123 | Thanhnam | ifqSrcUzvsNxMNgGuaD | [thanhnam@gmail.com](mailto:thanhnam@gmail.com) | 21 |

Collection: users

{

"\_id" : ObjectId("583fa7e4114b9128944751ce"),

"username" : "thanhnam",

"password" : "ifqSrcUzvsNxMNgGuaD",

"email" : "thanhnam@gmail.com",

"age" : 21

}

### Cách cài đặt mongoDB

Vào link: <https://www.mongodb.com/download-center> chọn phiên bản phù hợp rồi tải về và cài đặt bình thường.

Cấu hình mongoDB

* Mở cmd.exe ở chế độ Administrator
* Tạo các thư mục như sau: `C:\data\db` và `C:\data\log`
* Tạo file cấu hình:

Tạo file `mongod.cfg` có nội dung như sau:

*systemLog:*

*destination: file*

*path: c:\data\log\mongod.log*

*storage:*

*dbPath: c:\data\db*

Lưu vào: `C:\Program Files\MongoDB\Server\3.2\mongod.cfg`.

Mở cmd.exe ở chế độ Administrator và gõ và chạy lệnh sau:

*"C:\Program Files\MongoDB\Server\3.2\bin\mongod.exe" --config "C:\Program Files\MongoDB\Server\3.2\mongod.cfg" --install*

* Khởi động mongoDB service

Mở cmd.exe ở chế độ Administrator và gõ và chạy lệnh sau:

*net start MongoDB*

Khi thông báo là đã cài đặt và cấu hình thành công:

*The MongoDB service is starting.*

*The MongoDB service was started successfully.*

Vào giao diện làm việc với mongoDB, gõ lệnh sau:

*mongo*

### Các thao tác cơ bản trên mongoDB

Bảng 1.1. Các thao tác trên mongoDB

|  |  |
| --- | --- |
| Lệnh | Ý nghĩa |
| show dbs | Hiển thị toàn bộ database |
| use <database\_name> | Truy cập đến database ứng với database\_name; nếu database\_name chưa có thì sẽ khởi tạo database mới có tên là database\_name |
| db | Xem database đang truy cập hiện thời |
| db.dropDatabase() | Xóa database hiện thời đang truy cập |
| Show collections | Hiển thị toàn bộ danh sách collections có trong database |
| db.<cellection\_name>.find() | Select toàn bộ document của collection có trong collection\_name |
| db.<collection\_name>.find().pretty() | Select toàn bộ document trong collection\_name nhưng hiển thị dễ nhìn hơn. |
| db.<collection\_name>.find(  {<key>:<value>}  ) | Select theo điều kiện của key = value |
| db.<collection\_name>.find(  {<key>:{$lt:<value>}}  ) | Select theo điều kiện key < value |
| db.<collection\_name>.find(  {<key>:{$lte:<value>}}  ) | Select theo điều kiện key <= value |
| db.<collection\_name>.find(  {<key>:{$gt:<value>}}  ) | Select theo điều kiện key > value |
| db.<collection\_name>.find(  {<key>:{$gte:<value>}}  ) | Select theo điều kiện key >= value |
| db.<collection\_name>.find(  {<key>:{$ne:<value>}}) | Select theo điều kiện key != value |
| db.<collection\_name>.find(  {  $and: [  {key1: value1}, {key2:value2}  ]  }  ) | Select theo 2 điều kiện: key1= value1 and key2=value2 |
| db.<collection\_name>.find(  {  $or: [  {key1: value1}, {key2:value2}  ]  }  ) | Select theo 2 điều kiện: key1= value1 or key2=value2 |
| db.createCollection(<collection\_name>, options) | Khởi tạo collection trong database hiện thời |
| db.<collection\_name>.drop() | Xóa collection ứng với collection\_name |
| db.<collection\_name>.insert(document)  hoặc  db.<collection\_name>.save(document) | Insert document vào collection ứng với collection \_name |
| db.<collection\_name>.update(selection\_criteria, updated\_data) | Update document của collection\_name với updated\_data theo điều kiện selection\_criteria |
| db.<collection\_name>.remove(selection\_criteria) | Delete document của collection\_name với điều kiện selection\_criteria |

## GIỚI THIỆU VỀ ANGULARJS

Việc phát triển 1 website sử dụng AJAX thì không khó, bạn có thể dùng jQuery để làm việc này với $.ajax tuy nhiên làm thế nào để xây dựng một phần mềm có thể mở rộng, dễ test, nâng cấp và bảo trì thì không hề đơn giản vì bản thân JavaScript không được thiết kế ngay từ đầu để làm những việc này. Do đó sự ra đời của những framework hỗ trợ lập trình viên xây dựng ứng dụng web 1 cách có hệ thống đã ra đời như AngularJS.

### AngularJS là gì?

AngularJS là một framework mã nguồn mở mạnh mẽ, dễ dàng để học và ứng dụng. Nó cho phép bạn sử dụng HTML là ngôn ngữ mẫu và cho phép bạn mở rộng cú pháp HTML để diễn tả các thành phần ứng dụng của bạn một cách rõ ràng và ngắn gọn. AngularJS  framwork đã được xây dựng sẵn rất nhiều tính năng nên việc sử dụng nó sẽ giúp bạn tiết kiệm thời gian và code ít hơn.

AngularJS được phát triển bởi Google. Được phát triển từ năm 2009, phiên bản 1.0 ra đời tháng 6 năm 2012. Phiên bản mới nhất là 1.5.8.

### Các đặc tính của AngularJS

* AngularJS là một Framework phát triển dựa trên Javascript để tạo các ứng dụng web phong phú
* AngularJS thường dùng để phát triển frontend (giao diện khách hàng) thông qua các API để gọi data, sử dụng mô hình MVC rất mạnh mẽ
* Mã nguồn AngularJS tự động fix với các trình duyệt khác nhau nên bạn không cần phải lo vấn đề tương thích trình duyệt
* Angular là mã nguồn mở, hoàn toàn miễn phí và được phát triển bởi hàng ngàn các lập trình viên trên thế giới.

Chung quy lại có thể hiểu khi làm việc với AngularJS giống như là đang làm việc với Ajax, sử dụng cơ chế bind data, hoạt động theo mô hình MVC và sử dụng service để tương tác với dữ liệu từ server. Để rõ hơn thì chúng ta tìm hiểu các tính năng cố lõi của nó nhé.

### Các nội dung khi nghiên cứu AngularJS

Bảng 1.2. Các thành phần của AngularJS

|  |  |
| --- | --- |
| **Nội dung** | **Mô tả** |
| Template | Sử dụng ngôn ngữ HTML |
| Directives | Cho phép mở rộng HTML và bạn có thể custom lại các thuộc tính, phần tử (elements) |
| Model | Lưu trữ dữ liệu và hiển thị ra ngoài view của người dùng khi có 1 tương tác nào đó |
| Scope | Điểm trung gian, làm nhiệm vụ giao tiếp giữa model, controller, service, view,.. |
| Expressions | Truy cập vào các biến và function từ scope |
| Compiler | parses template and khởi tạo directives và expressions |
| Filter | Bộ lọc, format lại dữ liệu hiển thị ra ngoài view người dùng |
| View | Những gì người dùng có thể thấy được |
| Data Binding | chuyển đổi dữ liệu giữa model và view |
| Controller | Lớp business logic phía sau views |
| Dependency Injection | Liên kết, thống nhất các đối tượng và chức năng |
| Injector | Giống như thùng chứa các biển thể độc lập (hơi khó hiểu, nhưng mà sẽ rõ hơn khi đụng tới :p) |
| Module | Cũng giống như các framework khác, module như thùng chứa các thứ như là controller, service, directives,… |
| Service | Xây dựng service, mục đích là tái sử dụng lại các phương thức chung có thể sử dụng ở nhiều view hay controller khác nhau |

### Cách cài đặt

Tải thư viện angulajs.min.js phiên bản 1.5.8 từ link sau: <http://code.angularjs.org/1.5.8/angular-1.5.8.zip>, sau đó include và phần header của trang web.

*<script src=”/js/angularjs.min.js”></script>*

Hoặc include trực tiếp link CDN từ trang chủ.

*<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.5.6/angular.min.js"> </script>*

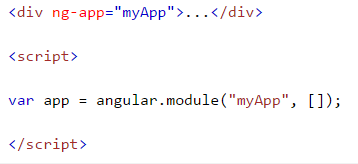
### Giới thiệu một vài nội dung trong AngularJS

#### Module

Module là một container cho các thành phần khác nhau của một ứng dụng như filters, controller, services, directives,…Bạn có thể hiểu đơn giản như sau, Module là hàm main của các loại ứng dụng khác.

Cách tạo Module như sau: **angular.module**

Ví dụ:



“myApp” là tên của của tham số dùng để chỉ phân tử HTML trong đó ứng dụng sẽ chạy.

#### Controller

Trong AngularJS thì controller là 1 hàm Javascript, có tác dụng là xây dựng 1 Model để cho View hiển thị.

Cách tạo Controller: **app.controller(“<tên controller>”,<tên hàm trong javascript>);**

Ví dụ:



#### Directives

Directive là một thành phần tính năng của AngularJS, nó giúp xấy dựng các thẻ mở rộng cho HTML nhằm thực hiện một mục đích nào đó.

Cấu trục của Directive: **ng-<tên directive>**

Các Directives Built-in phổ biến:

Bảng 1.3. Các Directive của AngularJS

|  |  |
| --- | --- |
| **Directive** | **Description** |
| ng-app | Xác định các yếu tố gốc của một ứng dụng. |
| ng-bind | Liên kết với các nội dung của một phần tử HTML để dữ liệu ứng dụng. |
| ng-change | Chỉ định một biểu thức để đánh giá khi nội dung được thay đổi bởi người sử dụng. |
| ng-checked | Chỉ định nếu một phần tử được kiểm tra hay không. |
| ng-class | Chỉ định các lớp CSS trên các phần tử HTML. |
| ng-click | Chỉ định một biểu thức để đánh giá khi một phần tử đang được nhấp vào. |
| ng-cloak | Ngăn chặn sự rung khi ứng dụng của bạn đang được nạp. |
| ng-controller | Xác định đối tượng điều khiển cho một ứng dụng. |
| ng-copy | Chỉ định một hành vi trên các sự kiện sao chép. |
| ng-csp | Thay đổi chính sách bảo mật nội dung. |
| ng-cut | Chỉ định một hành vi trên các sự kiện cắt. |
| ng-dblclick | Chỉ định một hành vi trên các sự kiện nhấp đúp. |
| ng-disabled | Chỉ định nếu một phần tử bị vô hiệu hóa hay không. |
| ng-focus | Chỉ định một hành vi trên các sự kiện tập trung. |
| ng-form | Chỉ định dạng HTML để thừa kế điều khiển từ. |
| ng-hide | Ẩn hoặc hiện các phần tử HTML. |
| ng-href | Chỉ định một địa chỉ cho các yếu tố <a>. |
| ng-if | Loại bỏ các yếu tố HTML nếu một điều kiện là sai. |
| ng-include | Bao gồm HTML trong một ứng dụng. |
| ng-init | Xác định giá trị ban đầu cho một ứng dụng. |
| ng-jq | Chỉ định rằng các ứng dụng phải sử dụng một thư viện, như jQuery. |
| ng-keydown | Chỉ định một hành vi trên các sự kiện KeyDown. |
| ng-keypress | Chỉ định một hành vi trên các sự kiện phím nhấn. |
| ng-keyup | Chỉ định một hành vi trên các sự kiện KeyUp. |
| ng-list | Chuyển đổi văn bản thành một danh sách (mảng). |
| ng-maxlength | Xác định số lượng ký tự tối đa cho phép trong các lĩnh vực đầu vào. |
| ng-minlength | Xác định số lượng tối thiểu của các nhân vật được cho phép trong các lĩnh vực đầu vào. |
| ng-model | Liên kết với các giá trị của các điều khiển HTML để dữ liệu ứng dụng. |
| ng-model-options | Chỉ định cách cập nhật trong mô hình được thực hiện. |
| ng-mousedown | Chỉ định một hành vi trên các sự kiện mousedown. |
| ng-mouseenter | Chỉ định một hành vi trên các sự kiện mouseenter. |
| ng-mouseleave | Chỉ định một hành vi trên các sự kiện MouseLeave. |
| ng-mousemove | Chỉ định một hành vi trên các sự kiện MouseMove. |
| ng-mouseover | Chỉ định một hành vi trên các sự kiện mouseover. |
| ng-mouseup | Chỉ định một hành vi trên các sự kiện mouseup. |
| ng-non-bindable | Chỉ định rằng không có ràng buộc dữ liệu có thể xảy ra trong phần tử này, hoặc con của nó. |
| ng-open | Chỉ định thuộc tính mở của một phần tử. |
| ng-options | Chỉ định <options> trong <select> danh sách. |
| ng-paste | Chỉ định một hành vi trên các sự kiện dán. |
| ng-pluralize | Chỉ định một tin nhắn để hiển thị theo quy định nội địa hóa en-us. |
| ng-readonly | Chỉ định thuộc tính readonly của một phần tử. |
| ng-repeat | Định nghĩa một mẫu cho mỗi dữ liệu trong bộ sưu tập. |
| ng-required | Chỉ định các thuộc tính cần thiết của một phần tử. |
| ng-selected | Chỉ định thuộc tính lựa chọn của một phần tử. |
| ng-show | Hiển thị hoặc ẩn các phần tử HTML. |
| ng-src | Chỉ định thuộc tính src của thẻ <img>. |
| ng-srcset | Chỉ định thuộc tính srcset cho các phần tử <img>. |
| ng-style | Chỉ định thuộc tính phong cách cho một phần tử. |
| ng-submit | Chỉ định biểu thức để chạy trên các sự kiện onsubmit. |
| ng-switch | Chỉ định một điều kiện sẽ được sử dụng để hiển thị các yếu tố con / hide. |
| ng-transclude | Chỉ định một điểm để chèn các phần tử nhúng. |
| ng-value | Xác định giá trị của một yếu tố đầu vào. |

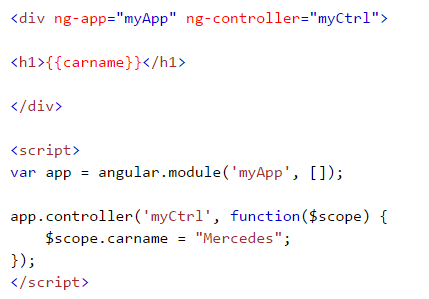
Trong AngularJS, có 3 Directive chính, đó là: ng-app, ng-model, ng-init.

#### Biến $scope

Biến $scope là một đối tượng có nhiệm vụ giao tiếp dữ liệu giữa controller và view. $scope chứa thông tin là các dữ liệu model, bao gồm các thuộc tính và phương thức. Trong controller, dữ liệu modle có thể được truy cập qua đối tượng $scope.

Cách dùng biến $scope: **$scope.<tên biến>**

Ví dụ:



#### Filters

Filter là định dạng giá trị của một biểu thức để hiển thị cho người dùng. Chúng có thể được sử dụng trong các view templates, controller hoặc services.

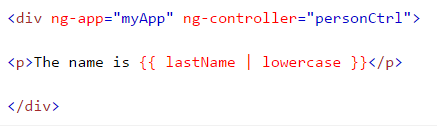
Các filter mà AngularJS cung cấp:

Bảng 1.4. Các Filter của AngularJS

|  |  |
| --- | --- |
| currency | Định dạng số sang 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 vào một chuỗi JSON. |
| limitTo | Giới hạn một mảng/ chuỗi, vào một số quy định của các yếu tố/ ký tự. |
| lowercase | Định dạng chuỗi sang chữ thường. |
| number | Định dạng một số thành một chuỗi. |
| orderBy | Đơn đặt hàng một mảng bằng một biểu thức. |
| uppercase | Định dạng chuỗi sang chữ hoa. |

Cách sử dụng Filter: {{ expression | filter }}

Ví dụ:



#### Services

Trong AngularJS, Service là một chức năng, hoặc đối tượng, nó có sẵn và giới hạn ứng dụng AngularJS của bạn. AngularJS có khoảng 30 Services được xây dựng sẵn.

Ví dụ: service $http, $location

#### $http Service

$http là một Service trong AngularJS, dùng để đọc dữ liệu từ các máy chủ từ xa.

Ví dụ:



### Ưu điểm và nhược điểm của AngularJS

Ưu điểm:

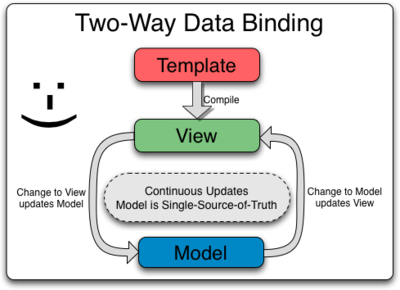
* Angular cho phép tạo ra các ứng dụng một cách đơn giản, code sạch
* Angular sử dụng data bind giống .NET với tính năng liên kết với HTML nên giúp người dùng cảm thấy dễ chịu.
* Angular đang ở giai đoạn thử nghiệm
* Angular có thể chạy trên hầu hết các trình duyệt điện thoại thông minh.

Nhược điểm:

* Mặc dù angular có nhiều lợi thế ưu điểm nhưng nó cũng có mặt trái ngược lại:
* **Không an toàn**: Được phát triển từ javascript nên nó không an toàn, phía máy chủ phải thường xuyên xác nhận quyền để hệ thống chạy trơn tru.
* **Phụ thuộc**: Nếu người dùng vô hiệu hóa javascript thì coi như ứng dụng vô dụng.

### Two-way Binding trong AngularJs

Data Binding giúp thể hiện giá trị của DOM, mà khi data được cập nhật thì kết quả sẽ hiển thị ngay chứ không cần phải reload page tức là cách thức tự đồng bộ hóa (synchronization) dữ liệu giữa model và view trong angularJS, nó cho phép tạo sự liên kết ràng buộc giữa model và template thông qua AngularJS.



Hình 1.1: Two-way Binding trong AngularJs

Quá trình đồng bộ luôn xảy ra giữa model và view. Mọi sự thay đổi trong lớp này sẽ được cập nhật vào trong lớp kia. Scope trong controller sẽ được truyền tới View và ngược lại.

Scope Data: sẽ được truyền tới View và nếu data thay đổi thì nó sẽ lập tức truyền tới View và Data View sẽ thay đổi theo.

View Data: sẽ được truyền tới Scope/Controller và nếu data thay đổi thì nó sẽ lập tức truyền tới Scope và Scope Data sẽ thay đổi theo.

## GIỚI THIỆU VỀ NODEJS

### Giới thiệu về nodeJS

NodeJS là một nền tảng được xây dựng trên “V8 Javascript engine” được viết bằng c++ và Javasccript. Nền tảng này được phát triển bởi Ryan Lienhart Dahl vào năm 2009.

NodeJS là một mã nguồn mở được sử dụng rộng bởi hàng ngàn lập trình viên trên toàn thế giới. NodeJS có thể chạy trên nhiều nền tảng hệ điều hành khác nhau từ WIndow cho tới Linux, OS X nên đó cũng là một lợi thế. NodeJS cung cấp các thư viện phong phú ở dạng Javascript Module khác nhau giúp đơn giản hóa việc lập trình và giảm thời gian ở mức thấp nhất.

Khi nói đến NodeJS thì phải nghĩ tới vấn đề Realtime. Realtime ở đây chính là xử lý giao tiếp từ client tới máy chủ theo thời gian thực.

### Các đặc điểm của NodeJS

**- Không đồng bộ**: Tất cả các API của NodeJS đều không đồng bộ (*none-blocking*), nó chủ yếu dựa trên nền của NodeJS Server và chờ đợi Server trả dữ liệu về. Việc di chuyển máy chủ đến các API tiếp theo sau khi gọi và cơ chế thông báo các sự kiện của Node.js giúp máy chủ để có được một phản ứng từ các cuộc gọi API trước (Realtime).

- **Chạy rất nhanh**: NodeJ được xây dựng dựa vào nền tảng V8 Javascript Engine nên việc thực thi chương trình rất nhanh.

- **Đơn luồng nhưng khả năng mở rộng cao**: Node.js sử dụng một mô hình luồng duy nhất với sự kiện lặp. cơ chế tổ chức sự kiện giúp các máy chủ để đáp ứng một cách không ngăn chặn và làm cho máy chủ cao khả năng mở rộng như trái ngược với các máy chủ truyền thống mà tạo đề hạn chế để xử lý yêu cầu. Node.js sử dụng một chương trình đơn luồng và các chương trình tương tự có thể cung cấp dịch vụ cho một số lượng lớn hơn nhiều so với yêu cầu máy chủ truyền thống như Apache HTTP Server.

- **Không đệm**: NodeJS không đệm bất kì một dữ liệu nào và các ứng dụng này chủ yếu là đầu ra dữ liệu.

- **Có giấy phép**: NodeJS đã được cấp giấy phép bởi [MIT License](https://raw.githubusercontent.com/joyent/node/v0.12.0/LICENSE).

### Cách cài đặt

- Truy cập <https://nodejs.org/en/download/> và tải phiên bản phù hợp với máy cần cài

- Tiến hành cài đặt bình thường

- mở cmd.exe gõ lệnh `node --version` và `npm --version`để kiểm tra, nếu màn hình hiển thị phiên bản (như *v7.0.0*) thì việc cài đặt đã thành công.

### Cách cài đặt các module nodeJS và Cách khai báo module trong nodejs

Các module nodeJS có thể được tìm thấy trên trang <https://www.npmjs.com/>

Để cài module, sử dụng câu lệnh như sau:

*`npm install <module\_name> --save`*

Cách khai báo module cần dùng, sử dụng cú pháp sau:

*`var <tenbien> = require(‘module\_name’);*

### Các module hay dùng trong nodeJS

Bảng 1.5. Các module hay dung trong nodeJS

|  |  |
| --- | --- |
| Tên module | Chức năng |
| **express** | đây được là là web framework của nodejs,cái này chúng ta sẽ tìm hiểu kỹ ở phần sau |
| **mongoose** | quản lý mongdb |
| **jade** | 1 dạng template mới dùng trong nodejs view,nhằm rút ngắn gọn view |
| **connect-mongodb** | giúp kết nối với mongodb |
| **nodemailer** | module gửi mail |
| **socket.io** | Sử dụng trong những ứng dụng cần realtime như chat, game realtime |
| **request** | quản lý yêu cầu và dữ liệu được truyền giữa client và server nodejs |
| **async** | module giúp bạn khử bất đồng bộ, chạy các hàm 1 cách tuần tự. |
| **lodash** | các hàm mở rộng để xử lý chuỗi, mảng, object |

## GIỚI THIỆU VỀ EXPRESSJS

### Giới thiệu về express

Express là một framework nhỏ và tiện ích để xây dựng các ứng dụng web, cung cấp một lượng lớn của tính năng mạnh mẽ để phát triển các ứng dụng web và mobile. Nó rất dễ dàng để phát triển các ứng dụng nhanh dựa trên Node.js cho các ứng dụng Web.

### Các đặc điểm của express

* Cho phép thiết lập các lớp trung gian để trả về các HTTP request.
* Định nghĩ bảng routing có thể được sử dụng với các hành động khác nhau dựa trên phương thức HTTP và URL.
* Cho phép trả về các trang HTML dựa vào các tham số truyền vào đến template.

### Cách cài đặt express và chương trình hello world

Để cài express vào nodeJS, dùng lệnh sau:

*`npm install express --save`*

Chương trình: Hello World

Tạo 1 thư mục chứa code có tên là: `firstapp`

Trong thư mục firstapp, tạo file app.js có nội dung như sau:

*var express = require('express')*

*var app = express()*

*app.get('/', function (req, res) {*

*res.send('Hello World!')*

*})*

*app.listen(3000, function () {*

*console.log('app listening on port 3000!')})*

Trong thư mục firstapp, ấn Shift + Right Click => open command windown here

Sau đó gõ lệnh: *`npm install express --save`*

Tiếp theo gõ lệnh: *`node app.js`*

Ra trình duyệt, vào địa chỉ: *`http://localhost:3000/`* và xem kết quả.

## GIỚI THIỆU VỀ FRAMEWORK MEAN.IO VÀ CÁCH CÀI ĐẶT

### Giới thiệu MEAN.IO

MEAN.IO là 1 framework miễn phí về mean stack. MEAN.IO do nhiều lập trình viên hàng đầu thế giới phát triển và có một cộng đồng lập trình viên đông đảo cùng sử dụng và phát triển nó. Hiện tại thì trang chủ chính thức của mean.io là <http://mean.io/>, và phiên bản mới nhất là mean-0.7

### Cách cài đặt mean.io

Để cài bộ mean.io, trước hết phải cài các thành phần sau:

* mongoDB
* nodeJS
* Git: truy cập trang web: <https://git-scm.com/downloads> để tải và cài đặt.

Sau khi cài các thành phần trên, cần cài thêm các gói tiên quyết sau:

* Gulp: *$ npm install -g gulp*
* Bower: *$* *npm install -g bower*
* Mean-cli: *$* *npm install -g mean-cli*

Khởi tạo ứng dụng mean stack với mean.io

* *$ mean init* <name\_app>
* *$ cd* <name\_app> *&& npm install*

Chạy ứng dụng mean vừa khởi tạo:

* *$ gulp hoặc $ node server*
* Vào trình duyệt và đi đến:[*http://localhost:3000*](http://localhost:3000)

Cấu trúc project với mean.io

Bảng 1.6. Cấu trúc thư mục project với mean.io

|  |  |  |
| --- | --- | --- |
| Server | config | File cấu hình |
| controllers | File controller |
| models | File Model |
| routes | File route |
| views | Các file html để render |
| Client | assets | JavaScript/CSS/Images |
| controllers | Angular controllers |
| directives | Angular directive |
| routes | Angular route |
| services | Angular service |
| views | Angular view |

### Các lệnh cơ bản với mean.io

|  |  |
| --- | --- |
| Lệnh | Ý nghĩa |
| $ mean --help | Mở trợ giúp |
| $ mean list | Danh sách package đã cài |
| $ mean package <package\_name> | Khởi tạo custom package |
| $ mean install <module\_name> | Cài module từ mean.io |
| $ mean uninstall <module\_name> hoặc  $ mean uninstall <package\_name> | Gỡ module hoặc custom package |

Bảng 1.7. Các lệnh thao tác cơ bản với mean.io

# CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## GIỚI THIỆU TỔNG QUAN HỆ THỐNG

Theo nhu cầu học tập và tìm hiểu của các bạn đam mê công nghệ, website đem lại các bài viết về nhiều chủ đề công nghệ mới hay một loạt bài viết về một chủ đề đang nổi trội của làng công nghệ với các hướng dẫn chi tiết về các bước thực hiện, xây dựng, giải thích, .... Các thành viên có thể đóng góp bài viết hoặc bình luận, góp ý cho bài viết thêm hoàn thiện và chính xác. Chia sẻ các bài viết thú vị cho bạn bè và những người quan tâm. Đó chính là mục tiêu mà hệ thống của chúng em nhắm tới.

## PHÂN TÍCH YÊU CẦU

### Yêu cầu chức năng

Website có các chức năng chính sau:

* Tác nhân Quản trị
* Quản lý Thành viên
* Quản lý Quyền
* Quản lý Danh mục

Và các chức năng của tác nhân Thành viên

* Tác nhân Thành viên
* Quản lý Bài viết
* Quản lý loạt bài viết

Và các chức năng của tác nhân Khách

* Tác nhân Khách
* Tìm kiếm
* Xem bài viết
* Xem loạt bài viết
* Đăng ký
* Đăng nhập

### Yêu cầu phi chức năng

* Giao diện dễ nhìn
* Thao tác dễ dàng, không phức tạp rườm rà
* Website tải nhanh

### Yêu cầu nền tảng hệ thống

* Máy tính có kết nối Internet

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. **CÁC VẤN ĐỀ ĐÃ LÀM ĐƯỢC**

Đã hoàn thiện được yêu cầu cơ bản của website.

Ưu điểm :

* Giao diện đẹp, đơn giản, dễ sử dụng.
* Thực hiện đa dạng các chức năng cơ bản, theo yêu chức năng của hệ thống.

1. **Các vẤn đỀ chưa làm đưỢc**

Website vẫn còn thiếu sót:

* Một số chức năng chưa được cài đặt: Đăng nhập qua mạng xã hội, Chia sẻ bài viết,....
* Chưa tối ưu với người dùng.

1. **HƯỚNG MỞ RỘNG VÀ PHÁT TRIỂN**

Tiếp tục nghiên cứu đa dạng các chức năng để đưa chương trình vào ứng dụng thực tế. Cải thiện, bổ sung các tính năng và chỉnh sửa giao diện thân thiện với người dùng hơn.

Nhận được sự chỉ dẫn của giảng viên hướng dẫn, chúng em sẽ cố gắng khắc phục những nhược điểm và phát triển chương trình.

# TÀI LIỆU THAM KHẢO

**Địa chỉ web**

[1]. http://www.uml-diagrams.org

[2]. https://docs.angularjs.org/guide/concepts

[3]. http://www.w3schools.com/angular/angular\_intro.asp

[4]. https://docs.mongodb.com

[5]. http://mean.io

[6]. https://nodejs.org/en/docs