# Chương 1: CƠ SỞ LÝ THUYẾT

## GIỚI THIỆU FRAMEWORK LARAVEL

* + 1. **Khái niệm**

Đưa ra một số khái niệm về Framework Laravel trong đó laravel là một nền tảng ứng dụng web mã nguồn mở được viết bằng ngôn ngữ PHP dựa theo mô hình MVC. Nó là framework khá mới mẻ, ra đời vào tháng 4-2011 và cha đẻ của nó Taylor Otwell. Việc tiếp cận và sử dụng Laravel cũng giống như các framework khác không phải là các vấn đề quá khó khăn. Các nhà phát triển framework Laravel đã xây dựng hệ thống hướng dẫn sử dụng khá hoàn chỉnh và đầy đủ. Nó giúp cho các lập trình viên tiết kiệm thời gian, công sức và chi phí để xây dựng một website.

* + 1. **Lịch sử phát triển của Framework Laravel**

Framework Laravel có nhiều phiên bản và phiên bản đầu tiên Laravel 1 được phát hình vào 6/2011 dưới bàn tay của Taylor Otwell**.**Tuy là một PHP Framework sinh sau đẻ muộn nhưng Laravel đã đem đến cộng đồng lập trình PHP một cách toàn diện từ Document và các phiên bản sau này được cập nhật thường xuyên. Nói một cách khác, chính Larvel đã kéo lại một lượng đông đảo các Dev chuyển sang Ruby on Rails ( một Ruby Framework nổi tiếng ). Đây là một Framework khá mới mẻ và dễ tiếp cận, được rất nhiều lập trình viên ưa thích và sử dụng.

* + 1. **Tổng quan cấu trúc ứng dụng**

Nội dung thư mục gốc:

Tại thư mục gốc của Laravel chứa tệp cấu hình của Composer, các thông tin về Framework, thư mục chứa cấu hình môi trường, các tập tin công khai, mã ứng dụng và thư mục với tất cả các gói được cài đặt bao gồm cả gói làm nên Laravel Framework

Bảng 1.1: Mô tả các tệp và thư mục trong thư mục gốc của laravel

|  |  |
| --- | --- |
| Tệp/Thư mục | Mục đích |
| /app | Chứa mã ứng dụng |
| /bootstrap | Chứa đường dẫn thư mục Laravel Framework, các tập tin được biên dịch và cấu hình môi trường ứng dụng. |
| /public | Đây là thư mục công khai. nó sẽ chứa CSS, Javascript và các file và thư mục khác sẽ cho phép truy cập bởi người sử dụng ứng dụng. |
| /vendor | Chứa các gói của bên thứ 3 và các gói làm nên Laravel Framework được cài đặt trong thư mục này. |
| composer.json | Đây là tập tin cấu hình Composer chứa danh sách các gói mà ứng dụng sử dụng, phiên bản cài đặt cho mỗi gói và thiết lập tính ổn định |
| CONTRIBUTING.md | Chứa hướng dẫn đóng góp vào Laravel Framework |
| readme.md | Cung cấp thông tin chung về Laravel và các liên kết tới tài liệu, các yêu cầu và giấy phép của ứng dụng. |
| server.php | Laravel có một lệnh đặc biệt chạy ứng dụng hiện tại trên máy chủ PHP của hệ thống. Tệp này giúp cho chức năng đó. |
| Artisan | Cung cấp tích hợp trong giao diện dòng lệnh của Laravel, giúp nhà phát triển có thể tăng tốc công việc. |
| phpunit.xml | Tệp cấu hình cho PHPUnit người dùng công cụng kiểm tra. |

Nội dung thư mục app

Khi xây dựng một ứng dụng với Laravel, các định tuyến, các bộ lọc định tuyến, cấu hình thiết lập khác và di trú cơ sở dữ liệu cho ứng dụng sẽ nằm trong thư mục này

Bảng 1.2: Mô tả các tệp và thư mục trong thư mục app của laravel

|  |  |
| --- | --- |
| Tệp/Thư mục | Mục đích |
| /models | Chứa các lớp đại diện cho các mô hình dữ liệu |
| /views | Chứa các bản mẫu cho views và bố cục của view |
| /controller | Chứa các controllers của ứng dụng |
| /config | Cài đặt các ứng dụng cụ thể như thông tin cơ sở dữ liệu, thiết đặt trình điều khiển phiên và các bộ đệm. |
| /database | Dữ liệu mẫu và di trú cơ sở dữ liệu |
| /lang | Chuỗi ngôn ngữ cho validation, pagination và các email nhắc nhở |
| /commands | Các lớp tùy biến bao gồm các lệnh Artisan |
| /storage | Bộ đệm, phiên tạm thời và các views được biên dịch |
| /start | Các file hỗ trợ điều chỉnh lỗi và chế độ bảo trì ứng dụng |
| /test | Người dùng kiểm tra của ứng dụngfilter.php Logic và định nghĩa các bộ lọc sử dụng trong ứng dụng |
| routes.php | Danh sách đầu cuối được đăng ký của ứng dụng |

* + 1. **Ưu điểm của Laravel.**

Trước hết, sự tinh tế của Laravel nằm ở chỗ bắt kịp được xu hướng công nghệ mà điểm nhấn ở đây là các tính năng mới trong các phiên bản PHP 5.3 trở lên. Điều đó được thể hiện qua khái niệm namespace, composer, closure và rất nhiều những tiêu chuẩn trong Design Pattern được áp dụng trên nền tảng Framework này. Đồng thời, với cách hướng dẫn đơn giản và dễ tiếp cận đã khiến người dùng thích ngay từ lần đầu làm việc với Framework này.

Laravel có nhiều ưu điểm so với nhiều thể loại cùng họ. Nó được thiết kế theo mô hình Model-View-Controller: Mô hình MVC giúp tách thành phần hiển thị giao diện (presentation) và xử lý (business logic) của một phần mềm thành những thành phần độc lập, từ đó giúp cho việc thiết kế, xử lý và bảo trì mã nguồn dễ dàng, đồng thời tăng khả năng mở rộng của phần mềm. CodeIgniter vận dụng mô hình này trong thiết kế, giúp tách biệt các tập tin giao diện với các tập tin xử lý dữ liệu, nâng cao khả năng quản lý và dễ bảo trì.

Hệ thống tài liệu (document) đã được làm lại một cách cẩn thận và rõ ràng hơn: Mỗi trang tài liệu đã được dành vô số thời gian tỉ mỉ để chăm chút cẩn thận. Điều này cũng giúp cho người sử dụng nhanh chóng nắm được cách sử dụng Laravel, từ đó nhiều người sử dụng nó hơn. Hệ thống tài liệu lần này đi kèm với khả năng tìm kiếm real-time và auto-complete.

Việc quản lý layout thật giản đơn. Dễ dàng tích hợp các thư viện của Zend và các gói bundle (giống như model) vào project. Có rất nhiều gói bundle được cộng đồng mạng build sẵn và chia sẻ trên Internet.

Nguồn tài nguyên của**Laravel**rất thân thiện với các lập trình viên với đa dạng tài liệu khác nhau để tham khảo. Các phiên bản được phát hành đều có nguồn tài liệu phù hợp với ứng dụng.

Lavarel là framework được trang bị API sạch trên thư viện **SwiftMailer**, do đó, có thể gửi thư qua các dịch vụ dựa trên nền tảng đám mây hoặc local.

Laravel hỗ trợ hiệu quả cho việc tạo lập website hay các dự án lớn trong thời gian ngắn.

Laravel sử dụng Blade Template Engine. Rất nhẹ nhàng và hữu dụng. Chỉ cần có 1 trang master template sẽ được sử dụng và kế thừa từ các file template con.

Laravel hỗ trợ nhiều cách viết để ngắn gọn nhất có thể và nhiều cách viết khác nhau như Fluent Query Builder (trình tạo truy vấn) và Eloquent (tài hùng biện). Cực kỳ ngắn gọn – dễ quản lý và hữu dụng. Hỗ trợ sẵn các hàm để tìm dữ liệu, nạp dữ liệu, đếm dòng ,… Laravel hỗ trợ tương tác với cơ sở dữ liệu như tạo cơ sở dữ liệu, tạo bảng, chỉnh sửa bảng, insert dữ liệu… thông qua Schema Builder (trình tạo giản đồ).

Route trong Laravel, có thể nhóm các rule (quy định) lại với nhau, các rule con nằm trong rule mẹ sẽ chịu ảnh hưởng.

Ví dụ

Route::group(array(‘before’=>’auth’),function(){

Route::get(‘admin/contacts’,’AdminController@contacts’);

Route::get(‘admin/deletessh’,’AdminController@deletessh’);

Route::post(‘admin/deletessh’,’AdminController@dodel’);

Route::get(‘admin/viewcontact/{id}’,’AdminController@viewcontact’);

}

Có thể thấy các rule con bên trong trước khi được thực thi sẽ phải chạy qua rule mẹ để kiểm tra, nếu thỏa mãn mới thực thi.

Laravel được phát hành theo giấy phép MIT, với source code được lưu trữ tại Github nên dễ dàng tìm kiếm và lấy code

**Laravel**được đón nhận và trở nên sử dụng phổ biến vì nó rất dễ sử dụng. Thường chỉ mất vài giờ, có thể bắt đầu thực hiện một dự án nhỏ với vốn kiến thức cơ bản nhất về lập trình với PHP.

Laravel cung cấp sẵn các tính năng bảo mật mạnh mẽ để người dùng hoàn toàn tập trung vào việc phát triển sản phẩm của mình.

* + 1. **Nhược điểm của Laravel**

Laravel chưa có nhiều tài liệu tiếng việt nên gặp rất nhiều khó khăn trong việc học hỏi và tìm kiếm code, tìm kiếm cách sửa lỗi khi có lỗi trong quá trình code.

Laravel thực hiện điều hướng hoàn toàn qua route. Tại đây người lập trình có thể làm tất cả các việc mà ngay cả Models và Controller đảm nhận theo mô hình MVC rất tiện. Bên cạnh đó khi đưa vào làm việc nhóm rất có khả năng xảy ra tranh trấp Laravel khi có lỗi ở route thì toàn bộ hệ thống sẽ bị lỗi và ngừng hoạt động.

* + 1. **Lý do lựa chọn Laravel**

Với những tính năng nổi bật của Laravel bao gồm cú pháp dễ hiểu rõ ràng, một hệ thống đóng gói modular (khối kết cấu) và quản lý gói phụ thuộc, nhiều cách khác nhau để truy cập vào các cơ sở dữ liệu quan hệ, nhiều tiện ích khác nhau hỗ trợ việc triển khai vào bảo trì ứng dụng, Laravel đã bắt đầu được nhiều lập trình viên Việt Nam hướng tới với những khả năng mà nó đem lại.

Cũng như các Framework khác, Laravel cho phép xây dựng bộ thư viện riêng và cho phép bất cứ đâu trong hệ thống có thể gọi tới và sử dụng thuận tiện cho việc lập trình. Những đặc điểm quan trọng này là cơ sở để phát triển ứng dụng lựa chọn Laravel làm công cụ chung để xây dựng hệ thống quản lý và tra cứu danh mục các công trình nghiên cứu khoa học

## CÀI ĐẶT

* + 1. **Yêu cầu hệ thống.**

Cũng như các yêu cầu cơ bản của **LARAVEL FRAMEWORK.** Trước tiên khi cài đặt Laravel, cần phải đảm bảo rằng Apache, PHP, MySQL đã được cài đặt. Để có thể cài Laravel, yêu cầu bắt buộc như sau:

* PHP >=5.5.9
* OpenSSL PHP Extension
* PDO PHP Extension
* Mbstring PHP Extension
* Tokenizer PHP Extension
  + 1. **Hướng dẫn cài đặt.**

Để cài Laravel, ta sẽ thực hiện theo các bước tuần tự.

Lưu ý: Máy tính phải được cài sẵn Xampp hoặc Wamp.

* Cách 1: Thông qua Laravel Installer

Đây là một thư viện của **Laravel**viết ra giúp cài đặt **Laravel**bởi 1 câu lệnh duy nhất, cài đặt thông qua Composer. Mở Terminal (CMD hoặc Git Bash) ở bất kỳ vị trí nào và gõ dòng lệnh sau:

composer global require "laravel/installer"

Khi cài đặt xong bạn cần chắc chắn rằng đường dẫn tập tin thực thi của Composerđã được thêm vào**Windows Enviroment Variables Path**

Đối với Windows, đường dẫn đó là "%appdata%\Composer\vendor\bin" và đối với macOS và Linux thì nó ở "~/.composer/vendor/bin".

Sau khi cài đặt xong, di chuyển vào thư mục htdocs của XAMPP, tại đây mở cửa sổ lệnh (đối với windows thì nhấp Shift + chuột phải và chọn **Commad Window Here**hoặc **Git Bash Here**) và gõ dòng lệnh sau:

laravel new blog

Trong đó “blog” chính là tên thư mục**laravel project**.

* Cách 2: Thông qua Composer

Di chuyển thẳng vào thư mục htdocs của XAMPP, tại đây các bạn mở cửa sổ lệnh (như trên) và gõ dòng lệnh sau:

composer create-project --prefer-dist laravel/laravel blog

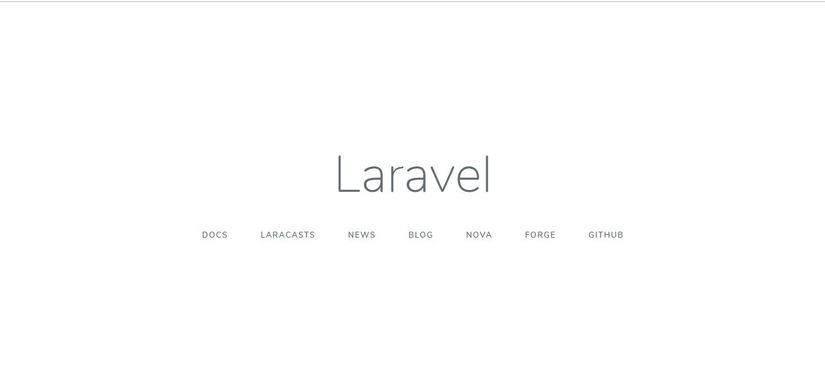
Trong đó blog chính là tên thư mục **laravel project**.

Qua một trong hai cách trên,ta đã có thể khởi tạo một ứng dụng Laravel framework rồi, tiếp theo là cách để khởi động ứng dụng.

Nếu máy tính đã cài đặt PHP thì có thể chạy dòng lệnh sau để khởi động server:

php artisan serve

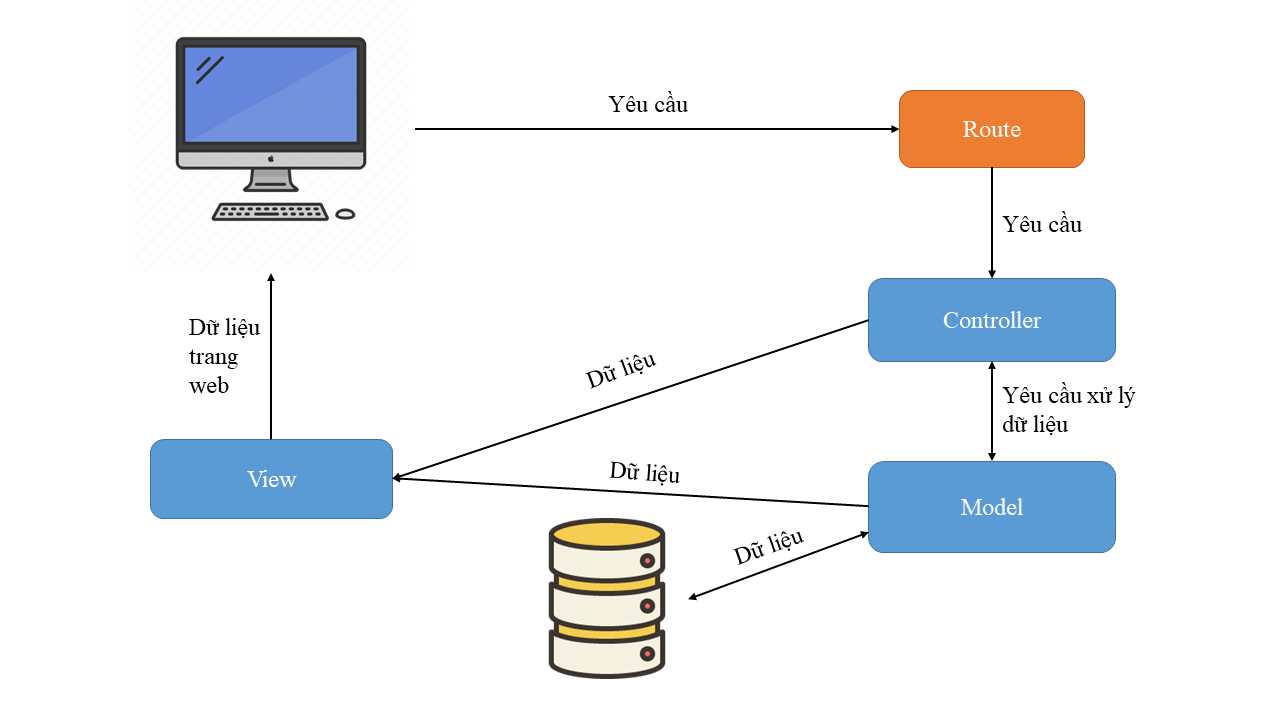
Sau khi chạy dòng lệnh, mở trình duyệt và truy cập địa chỉ [http://localhost:8000](http://localhost:8000/) sẽ thu được kết quả như hình bên dưới:



Mặc định, Laravel khi khởi động chạy ở port 8000, nếu muốn thay đổi thiết lập này có thể thêm tham số port vào lệnh:

php artisan serve --port=8080

## MÔ HÌNH MVC TRONG FRAMEWORK LARAVEL.



MVC là viết tắt của ba từ Model - View - Controller. Trong đó:

Model: cấu trúc dữ liệu theo cách tin cậy và chuẩn bị dữ liệu theo lệnh của controller

View Hiển thị dữ liệu cho người dùng theo cách dễ hiểu dựa trên hành động của người dùng.

Controller Nhận lệnh từ người dùng, gửi lệnh đến cho Model để cập nhập dữ liệu, truyền lệnh đến View để cập nhập giao diện hiển thị

## HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU.

**Hệ quản trị cơ sở dữ liệu** (*Database Management System)*có thể hiểu là hệ thống được thiết kế để quản lí một khối lượng dữ liệu nhất định một cách tự động và có trật tự. Các hành động quản lý này bao gồm chỉnh sửa, xóa, lưu thông tin và tìm kiếm (truy xuất thông tin) trong một nhóm dữ liệu nhất định.

## HƯỚNG DẪN SỬ DỤNG LARAVEL FRAMEWORK.

* + 1. **Cách sử dụng Route.**

Câu lệnh:

Phương thức đường dẫn tham số

Route::get( ‘home’, function(){} );

* + 1. **Sử dụng View.**

Vào thư mục resources/views trong laravel

Tạo một file view có đuôi .blade.php (ví dụ: home.blade.php)

Sau đó dùng câu lệnh sau:

Route::get( ‘home’, function(){

Return view(‘home’);

} );

* + 1. **Sử dụng Controller trong Laravel 5.**

Vào thư mục laravel gõ cmd:

Câu lệnh: php artisan make:controller homeController

Trong homeController:

Public function viewhome(){

Return view(‘home’);

}

Trong route:

Route::get( ‘home’, ‘homeController@viewhome’);

* + 1. **Tạo bảng và khóa ngoại trong database.**

Vào thư mục laravel gõ cmd:

Câu lệnh: php artisan make:migration create\_tablename\_table

Vào database/migration tìm file vừa tạo:

Tạo khóa chính:

$table -> bigIncrements(‘id’);

Tạo khóa phụ:

$table -> unsignedBigInteger(‘tên khóa phụ’);

$table->foreign(‘tên khóa phụ’)->references(‘id’)->on(‘bảng chứa khóa phụ’)->onDelete('cascade');

## KẾT LUẬN CHƯƠNG

Có thêm hiểu biết về Laravel Framework. Hiểu cách thao tác với cơ sở dữ liệu theo phương pháp lập trình hướng đối tượng mà Laravel Framework cung cấp. Áp dụng cơ sở lý thuyết để phân tích bài toán ở chương tiếp theo.