**KIỂM TRA**

**I. PHẦN LÝ THUYẾT:**

**Câu 1:** Trình bày mô hình MVC và cấu trúc thư mục của Laravel

**Câu 2:** Hãy trình bày cấu trúc thư mục MVC thực sự cần thiết cho dự án của mình trong môn học này

**II. PHẦN THỰC HÀNH**

**Câu 1:**Tạo trang hiển thị nội dung bài viết theo 2 dạng:

Danh sách bài viết theo nguyên tắc, bài viết mới post sẽ xuất hiện trước

Hiển thị danh sách bài viết theo dạng bảng, mỗi dòng 3-4 cột

**Câu 2:**Tạo Project có tên: kiemtra\_PHP

Tạo file kết nối CSDL theo PDO

Lấy dữ liệu từ Website\_Demo: chia ra thành 4 file: header.php; footer.php; menu.php; index.php

Sửa các code các file đăng nhập, đăng ký sang PDO

**Làm:**

**Câu 1:**

***Mô hình MVC là gì?***

**MVC** là từ viết tắt bởi 3 từ **Model – View – Controller**. Đây là mô hình thiết kế sử dụng trong kỹ thuật phần mềm. Mô hình source code thành 3 phần, tương ứng mỗi từ. Mỗi từ tương ứng với một hoạt động tách biệt trong một mô hình.

Để hiểu rõ hơn, sau đây chúng ta sẽ cùng đi phân tích từng thành phần:



Mô hình MVC và các thành phần bên trong của MVC.

**Model (M):**

Là bộ phận có chức năng lưu trữ toàn bộ dữ liệu của ứng dụng. Bộ phận này là một cầu nối giữa 2 thành phần bên dưới là View và Controller. Model thể hiện dưới hình thức là một cơ sở dữ liệu hoặc có khi chỉ đơn giản là một [file XML](https://monamedia.co/file-xml-la-gi/) bình thường. Model thể hiện rõ các thao tác với cơ sở dữ liệu như cho phép xem, truy xuất, xử lý dữ liệu,…

**View (V):**

Đây là phần giao diện (theme) dành cho người sử dụng. Nơi mà người dùng có thể lấy được thông tin dữ liệu của MVC thông qua các thao tác truy vấn như tìm kiếm hoặc sử dụng thông qua các website.

Thông thường, các ứng dụng web sử dụng MVC View như một phần của  hệ thống, nơi các thành phần [HTML](https://monamedia.co/html-va-css-la-gi/) được tạo ra. Bên cạnh đó, View cũng có chức năng ghi nhận hoạt động của  người dùng để tương tác với Controller. Tuy nhiên, View không có mối quan hệ trực tiếp với Controller, cũng không được lấy dữ liệu từ Controller mà chỉ hiển thị yêu cầu chuyển cho Controller mà thôi.

Ví dụ: Nút “delete” được tạo bởi View khi người dùng nhấn vào nút đó sẽ có một hành động trong Controller.

**Controller (C):**

Bộ phận có nhiệm vụ xử lý các yêu cầu người dùng đưa đến thông qua view. Từ đó, C đưa ra dữ liệu phù hợp với người dùng. Bên cạnh đó, Controller còn có chức năng kết nối với model.



Luồng tương tác giữa các thành phần trong MVC.

Ví dụ: Chức năng thêm bài viết mới trong trang quản trị website. Nơi đây có 2 trường nhập về tiêu đề và nội dung bài viết, thì trong đó:

* View sẽ hiển thị ra phần nhập form tiêu đề và nội dung.
* Controller lấy dữ liệu từ 2 trường và gửi tới Model.
* Model lúc này sẽ nhận dữ liệu từ Controller để lưu trữ trong cơ sở dữ liệu.

Mỗi bộ phận thực hiện chức năng nhất định, nhưng chúng có sự thống nhất, liên kết với nhau tạo nên **mô hình MVC**. Mô hình này tương đối nhẹ. Nó có thể tích hợp được nhiều tính năng có trong ASP.NET hiện giờ. Ví dụ như authentication (quá trình xác thực).

## Luồng đi trong mô hình MVC như thế nào?

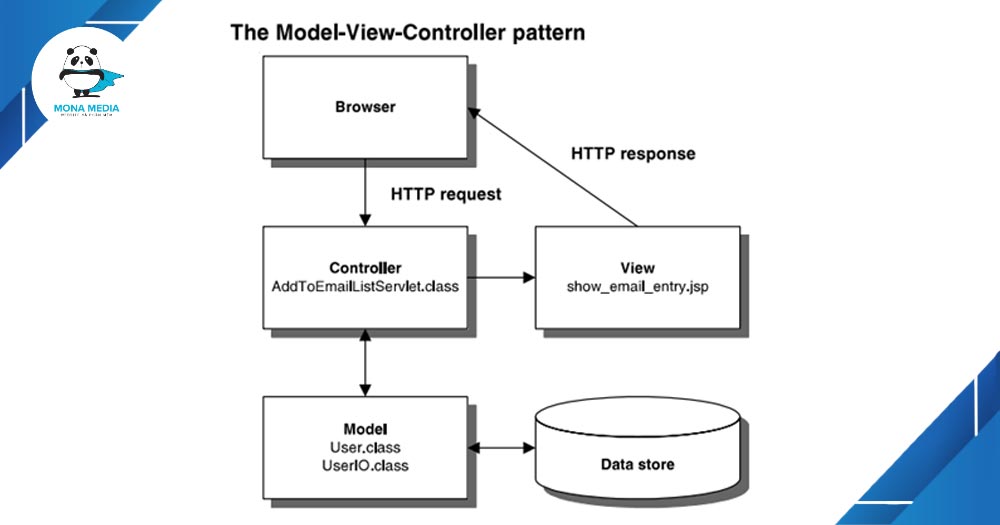
Bạn có thể hình dung, khi  một yêu cầu từ máy client gửi tới server, Controller sẽ thực hiện nhiệm vụ của mình đó là tiếp nhận và xử lý yêu cầu. Trong trường hợp cần thiết, nó có thể liên hệ Model – bộ phận làm việc với database để hỗ trợ.

Khi xử lý xong yêu cầu, kết quả sẽ được trả về View. Tại View sẽ tạo mã HTML thành giao diện và trả về hiển thị trên trình duyệt.

## Ưu & nhược điểm của MVC?

### Ưu điểm của mô hình MVC

Nhẹ, tiết kiệm băng thông: MVC không sử dụng viewstate nên khá tiết kiệm diện tích băng thông. Khi sử dụng, người dùng có thể sử dụng ứng dụng trên web cần tương tác gửi và nhận dữ liệu một cách liên tục. Do đó, việc giảm băng thông giúp cho website hoạt động tốt và ổn định hơn.



Ưu – nhược điểm của mô hình MVC.

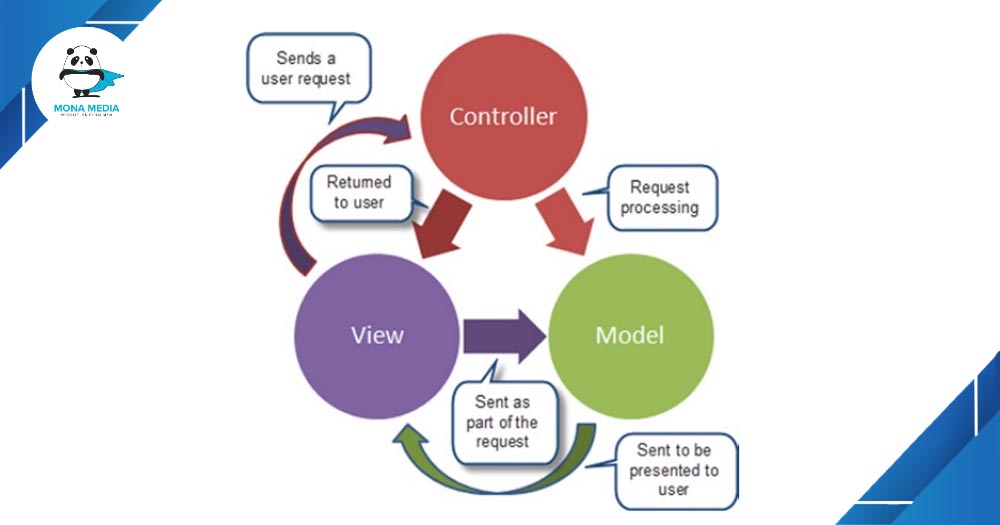
* Kiểm tra dễ dàng: Với **MVC**, bạn có thể dễ dàng kiểm tra, rà soát lỗi phần mềm trước khi tới tay người tiêu dùng, đảm bảo chất lượng và độ uy tín cao hơn.
* Chức năng control: Trên các nền website thì ngôn ngữ lập trình như CSS, HTML, [Javascript](https://monamedia.co/javascript-la-gi-gioi-thieu-ve-ngon-ngu-js-cho-nguoi-moi-hoc/) có một vai trò vô cùng quan trọng. Việc sử dụng mô hình MVC sẽ giúp bạn có một bộ control ưu việt trên nền tảng các ngôn ngữ hiện đại với nhiều hình thức khác nhau.
* View và size: View sẽ là nơi lưu trữ các dữ liệu. Càng nhiều yêu cầu được thực hiện thì kích thước càng tệp càng lớn. Khi  đó, đường truyền mạng cũng giảm tốc độ load. Việc sử dụng mô hình MVC sẽ giúp bạn tiết kiệm được diện tích băng thông một cách tối ưu.
* Chức năng Soc (Separation of Concern): Chức năng này cho phép bạn phân tách rõ ràng các phần như Model, giao diện, data, nghiệp vụ.
* Tính kết hợp: Việc tích hợp ở **mô hình MVC** cho phép bạn thoải mái viết code trên nền tảng website. Khi đó, server của bạn sẽ được giảm tải khá nhiều.
* Đơn giản: Đây là một mô hình với kết cấu tương đối đơn giản. Dù bạn không có quá nhiều chuyên môn cũng có thể sử dụng được.

### Nhược điểm của mô hình MVC

MVC thường được sử dụng vào những  dự án lớn. Do đó, với các dự án nhỏ, mô hình MVC có thể gây cồng kềnh, tốn thời gian trong quá trình phát triển cũng như thời gian trung chuyển dữ liệu.

## Ứng dụng mô hình MVC vào lập trình

Mô hình MVC được ứng dụng trong nhiều [ngôn ngữ lập trình](https://monamedia.co/top-10-ngon-ngu-lap-trinh-phan-mem-tot-nhat-hien-nay/) khác nhau, nhưng phổ biến nhất là ứng dụng ASP.NET MVC hay PHP MVC.



MVC đang là mô hình được ứng dụng rất nhiều trong lập trình.

Hệ thống MVC phát triển tốt sẽ cho phép phát triển front – end, back – end cùng trên hệ thống mà không có sự can thiệp, chia sẻ, chỉnh sửa các tập tin trong khi một hoặc hai bên vẫn đang làm việc.

Việc sử dụng mô hình tương đối đơn giản. Chỉ cần hiểu rõ quy trình vận hành, nắm được các chức năng của từng bộ phận thì việc triển khai **mô hình MVC** tương đối dễ dàng.

Trên đây là một số kiến thức cơ bản về mô hình MVC. Hi vọng bài viết trên đây của [Mona Media](https://monamedia.co/) sẽ giúp bạn hiểu rõ hơn về MVC và ứng dụng một cách hiệu quả.

***Cấu trúc thư mục Laravel:***

# Cấu trúc thư mục Laravel (Dicrectory Structure Laravel)

Chúng ta lại gặp nhau nữa rồi, và tập ngày hôm nay mình xin giới thiệu với các bạn về cấu trúc thư mục trong Laravel và chức năng của từng thành phần.

Laravel cung cấp một cấu trúc thư mục có thể bắt đầu với một project dù lớn hay nhỏ. Chúng ta có thể tự do tổ chức ứng dụng chúng ta theo một cách riêng nếu muốn.

## 1. Thư mục app (The app directory)

Thư mục này chứa những code cốt lõi (core code) của ứng dụng. Hầu như tất cả các lớp (class) bạn tạo cho project sẽ nằm ở đây.

## 2. Thư mục bootstrap (The bootstrap directory)

Thư mục bootstrap chứa file app.php làm việc như một bootstrap của ứng dụng. Ngoài ra còn có thư mục cache dùng để chứa các file bộ nhớ config, route, services... cho việc tối ưu hiệu năng.

## 3. Thư mục config (The config directory)

Thư mục này chứa tất cả file config ứng dụng, rất thuận tiện cho việc thay đổi các thiết lập.

## 4. Thư mục database (The database directory)

Như tên gọi của nó, thư mục này sẽ chứa các file làm việc với cơ sở dữ liệu (database) của ứng dụng. Trong này gồm 3 phần: factories, migrations và seeds.

Về phần factories, nói dễ hiểu nó sẽ có chức năng tạo dữ liệu ảo database, phối hợp cho việc testing.

Thư mục migrations sẽ chứa các file dùng để khởi tạo các bảng (table) trong database. Mình sẽ tìm hiểu sâu hơn trong những tập kế tiếp.

Còn về seeds, chẳng hạn khi ứng dụng của bạn bị xóa hết các table trong database, thì với seeds cùng với các file seeder sẽ giúp chúng ta khôi phục lại database theo những gì đã thiết lập sẵn trước đó nhưng có thể sẽ mất đi dữ liệu đã có hoặc thay thế bằng dữ liệu mặc định được khai báo trong các file seeder.

## 5. Thư mục public (The public directory)

Thư mục public chứa file index.php, file này đảm nhận vai trò như một đích đến của các request và autoload các lớp. Ngoài ra nó còn chứa các tài nguyên mà trình duyệt (browser) có thể truy cập như JS, CSS, hình ảnh...

## 6. Thư mục resources (The resources directory)

Thư mục resources chứa các tài nguyên thô chưa được biên dịch như view, LESS, SASS hoặc Javascript...

## 7. Thư mục routes (The routes directory)

Thư mục routes chứa các tuyến đường (route) đã định nghĩa của ứng dụng. Mặc định các file: api.php, web.php, channels.php và console.php được kết nối với Laravel.

Mình sẽ giải thích đơn giản công dụng của từng file:

File web.php: đơn giản nó sẽ chứa những route chứa request từ trình duyệt, chịu ảnh hưởng từ session, cookie, CSRF (tính năng bảo mật trong Laravel). Nếu ứng dụng của bạn không có các RESTful API thì hầu như các route sẽ nằm trong file này.

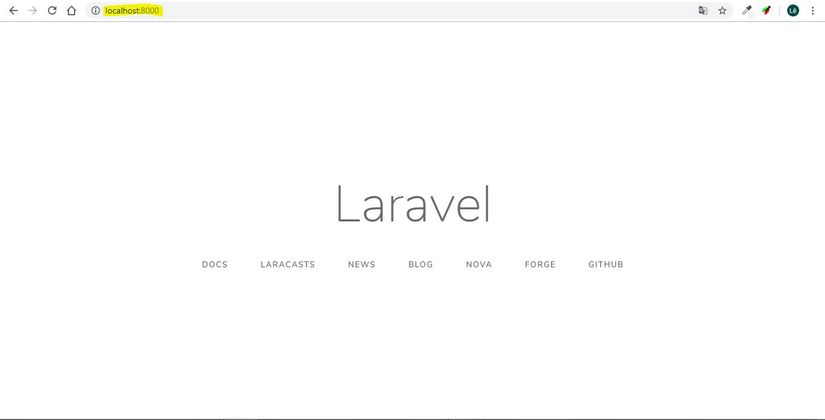
Chẳng hạn đoạn code bên dưới ở file routes/web.php:

Route::get('/', function () {

return view('welcome');

});

Đây chính là code khai báo một route với phương thức GET, khi truy cập đường dẫn [http://localhost:8000](http://localhost:8000/) thì trình duyệt sẽ trả về một view với tên là *"welcome"*.



File api.php: như ở trên đã nói, file này sẽ chứa các route có chức năng như là RESTful API hoặc cái gì đó tương tự, chịu ảnh hưởng qua kiểm duyệt token, authenticated... và có thể giới hạn thời gian tồn tại.

File console.php: nơi đây bạn có thể định nghĩa các Clouser bằng các lệnh console, dù nằm trong nhóm route nhưng đây không phải định nghĩa một route theo phương thức HTTP.

Để test xem nó làm việc thế nào, bạn có thể mở file routes/console.php sẽ thấy một đoạn code ví dụ (example code) bên dưới:

Artisan::command('inspire', function () {

$this->comment(Inspiring::quote());

})->describe('Display an inspiring quote');

Bây giờ các bạn mở commander lên và chạy thử lệnh:

php artisan inspire

# Cấu trúc thư mục Laravel (Dicrectory Structure Laravel)

Laravel cung cấp một cấu trúc thư mục có thể bắt đầu với một project dù lớn hay nhỏ. Chúng ta có thể tự do tổ chức ứng dụng chúng ta theo một cách riêng nếu muốn.

## 1. Thư mục app (The app directory)

Thư mục này chứa những code cốt lõi (core code) của ứng dụng. Hầu như tất cả các lớp (class) bạn tạo cho project sẽ nằm ở đây.

## 2. Thư mục bootstrap (The bootstrap directory)

Thư mục bootstrap chứa file app.php làm việc như một bootstrap của ứng dụng. Ngoài ra còn có thư mục cache dùng để chứa các file bộ nhớ config, route, services... cho việc tối ưu hiệu năng.

## 3. Thư mục config (The config directory)

Thư mục này chứa tất cả file config ứng dụng, rất thuận tiện cho việc thay đổi các thiết lập.

## 4. Thư mục database (The database directory)

Như tên gọi của nó, thư mục này sẽ chứa các file làm việc với cơ sở dữ liệu (database) của ứng dụng. Trong này gồm 3 phần: factories, migrations và seeds.

Về phần factories, nói dễ hiểu nó sẽ có chức năng tạo dữ liệu ảo database, phối hợp cho việc testing.

Thư mục migrations sẽ chứa các file dùng để khởi tạo các bảng (table) trong database. Mình sẽ tìm hiểu sâu hơn trong những tập kế tiếp.

Còn về seeds, chẳng hạn khi ứng dụng của bạn bị xóa hết các table trong database, thì với seeds cùng với các file seeder sẽ giúp chúng ta khôi phục lại database theo những gì đã thiết lập sẵn trước đó nhưng có thể sẽ mất đi dữ liệu đã có hoặc thay thế bằng dữ liệu mặc định được khai báo trong các file seeder.

## 5. Thư mục public (The public directory)

Thư mục public chứa file index.php, file này đảm nhận vai trò như một đích đến của các request và autoload các lớp. Ngoài ra nó còn chứa các tài nguyên mà trình duyệt (browser) có thể truy cập như JS, CSS, hình ảnh...

## 6. Thư mục resources (The resources directory)

Thư mục resources chứa các tài nguyên thô chưa được biên dịch như view, LESS, SASS hoặc Javascript...

## 7. Thư mục routes (The routes directory)

Thư mục routes chứa các tuyến đường (route) đã định nghĩa của ứng dụng. Mặc định các file: api.php, web.php, channels.php và console.php được kết nối với Laravel.

Mình sẽ giải thích đơn giản công dụng của từng file:

File web.php: đơn giản nó sẽ chứa những route chứa request từ trình duyệt, chịu ảnh hưởng từ session, cookie, CSRF (tính năng bảo mật trong Laravel). Nếu ứng dụng của bạn không có các RESTful API thì hầu như các route sẽ nằm trong file này.

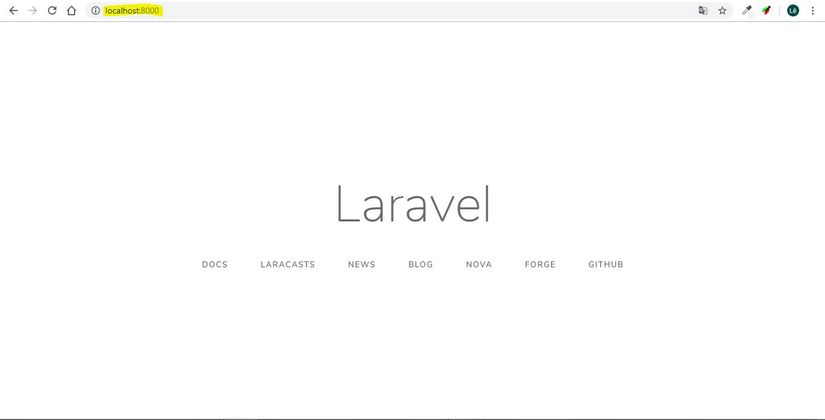
Chẳng hạn đoạn code bên dưới ở file routes/web.php:

Route::get('/', function () {

return view('welcome');

});

Đây chính là code khai báo một route với phương thức GET, khi truy cập đường dẫn [http://localhost:8000](http://localhost:8000/) thì trình duyệt sẽ trả về một view với tên là "welcome".



File api.php: như ở trên đã nói, file này sẽ chứa các route có chức năng như là RESTful API hoặc cái gì đó tương tự, chịu ảnh hưởng qua kiểm duyệt token, authenticated... và có thể giới hạn thời gian tồn tại.

File console.php: nơi đây bạn có thể định nghĩa các Clouser bằng các lệnh console, dù nằm trong nhóm route nhưng đây không phải định nghĩa một route theo phương thức HTTP.

Để test xem nó làm việc thế nào, bạn có thể mở file routes/console.php sẽ thấy một đoạn code ví dụ (example code) bên dưới:

Artisan::command('inspire', function () {

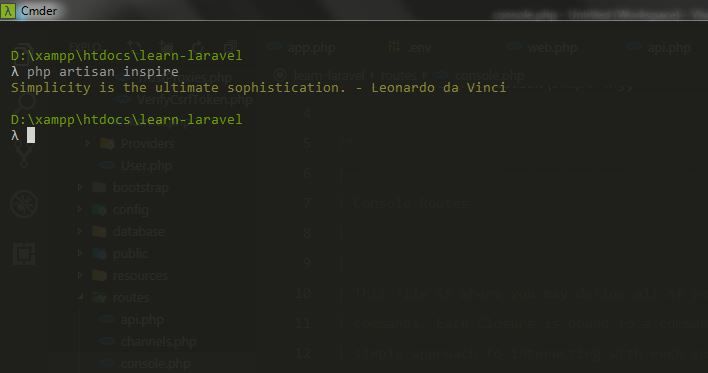
$this->comment(Inspiring::quote());

})->describe('Display an inspiring quote');

Bây giờ các bạn mở commander lên và chạy thử lệnh:

php artisan inspire

Một kết quả khá thú vị sẽ được trả về:



Nó trả về cho chúng ta một câu nói hoặc danh ngôn của người nổi tiếng, nếu bạn chạy lệnh một lần nữa sẽ ra một câu khác. Đó chính là cách "làm việc" của console route.

File channels.php: bạn tìm đến file này khi ứng dụng của bạn sử dụng thời gian thực (real-time), file này hỗ trợ cho ứng dụng của bạn có thể tương tác các sự kiện giữa phía người dùng (client-side) và phía hệ thống (server-side).

## 8. Thư mục storage (The storage directory)

Thư mục storage chứa các file blade template (chúng ta sẽ tìm hiểu nó ở tập sau) đã được phiên dịch (complied), các file session, file cache và một số file khác được tạo bởi framework. Thư mục này gồm app, framework và logs.

Thư mục app dùng để lưu trữ bất kỳ file nào do ứng dụng của mình tạo ra. Thư mục storages/app/public có thể dùng để lưu trữ các file do người dùng (user) đăng tải, chẳng hạn như ảnh đại diện (avatar) có thể truy cập công khai.

Về framework như đã nói ở trên, nó dùng để lưu trữ các file mà framework tạo ra để hỗ trợ trong việc chạy ứng dụng.

Cuối cùng, thư mục logs sẽ chứa các file log gồm có các lỗi trong quá trình code (errror log).

## 9. Thư mục tests (The tests directory)

Thư mục này bạn chỉ hiểu đơn giản dùng để test các class trong quá trình thử nghiệm trên commander.

## 10. Thư mục vendor (The vendor directory)

Thư mục vendor chứa các thư việc tích hợp và mã nguồn của Laravel, nếu bạn nào thích tìm tòi, khám phá và học hỏi cách code của nó thì nên tìm hiểu sâu thư mục này.

**Câu 2:** Hãy trình bày cấu trúc thư mục MVC thực sự cần thiết cho dự án của mình trong môn học này :

Bất kì một Framework nào cũng đưa ra một mô hình các cấu trúc thư mục chuẩn để giúp các lập trình viên sử dụng có thẻ đọc code của nhau và hiểu được, với cấu trúc này giúp cho lập trình viên dễ dàng tiếp xúc hơn thay vì bắt họ tự nghĩ ra.

Cấu trúc thư mục cơ bản trong Laravel FW

pathroot/

/app/

/commands/

/config/

/controllers/

/database/

/lang/

/models/

/start/

/storage/

/tests/

/views/

/bootstrap/

/public/

/index.php

/assets/

/packages/

/uploads/

/vendor/

/workbench/

/[Composer](https://freetuts.net/composer-la-gi-695.html).json

/Artisan

/server.php

Trong đó:

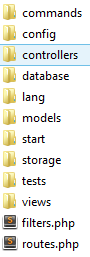
* Pathroot là đường dẫn tới thư mục laravel và chứa tất cả các file của laravel, bao gồm các file cần phải có composer.json, artisan, server.php, và các thư mục con chính app, bootstrap, public, vendor, workbench(*có thể không có*).​

Composer.json là file để cấu hình việc thao tác với composer như install hay update Laravel, thêm các file hỗ trợ ...

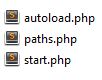
Artisan là file mà laravel tạo ra để hỗ trợ chạy lệnh: php artisan

server.php cần có để chạy lệnh: php artisan serve

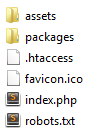
* app là thư mục chứa các file cấu hình, lưu trữ, tập lệnh của laravel, trong đó gồm có:
  + commands: các command sử dụng trong laravel (hiện tại chưa cần để ý đến nó).
  + config: nơi chứa các file cấu hình laravel như database, mail, url, ...
  + models, views, controllers: nơi chứa file của mô hình MVC
  + database: nơi chứa các file xây dựng và khởi tạo cơ sở dữ liệu
  + lang: nơi chứa các file ngôn ngữ
  + start: các file xử lý khi laravel hoạt động
  + storage: nơi chứa các file lưu trữ của laravel như log, cache, ...
  + tests: chứa test file (cũng chưa cần để ý đến nó).
  + File routes.php: nơi chứa các định tuyến (route) của laravel
  + file filters.php: nơi chứa các bộ lọc định tuyến.



* bootstrap: thư mục chứa file cài đặt các biến cơ bản của laravel (paths.php), nơi cài đặt môi trường làm việc (start.php) đồng thời cũng là nơi các filekhác được include vào laravel (autoload.php).

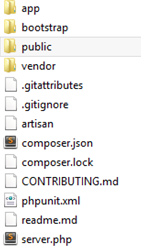


* public: chứa file index.php, .htaccess, assets (thường dùng để chứa các file js, css, image của giao diện) . File khi khởi chạy ứng dụng, file .htaccess sẽ chuyển hướng mọi yêu cầu (request) tới file index.php, index.php sẽ gọi đến các thành phần tương ứng của laravel (model, view, controller, ...) để thực thi và trả về kết quả (response).



* vendor: chứa bộ mã nguồn của laravel và các thành phần đi kèm laravel, cũng như các gói (packages) sau này sẽ thêm vào laravel
* workbench: thư mục dành cho các lập trình viên tự tạo ra các gói (package). Mặc định thư mục này sẽ không tồn tại

Hình ảnh bao quát:



Lưu ý: Một việc rất quan trọng trong học laravel đó là tìm lỗi của laravel, mặc định thì laravel đã ẩn lỗi đi, nếu muốn hiển thị lỗi bạn vào file app/config/app.php tìm và sửa như sau:

*//tìm dòng*

*'debug' => flase,*

*và sửa thành*

*'debug' => true,*