BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**

**NIÊN LUẬN**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**Đề tài**

**Tìm hiểu và ứng dụng công nghệ RFID**

**Sinh viên thực hiện:**

**Trần Lý Văn**

**Mã số sinh viên: B1305056**

**Khóa: 39**

**Cần thơ, 03/2017**

BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**

**NIÊN LUẬN**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**Đề tài**

**Tìm hiểu và ứng dụng công nghệ RFID**

**Giảng viên hướng dẫn: Sinh viên thực hiện:**

**PGS.TS TRẦN CAO ĐỆ TRẦN LÝ VĂN**

**Mã số sinh viên: B1305056**

**Khóa: 39**

**Cần thơ, 03/2017**

***Nhận Xét Của Giáo Viên Hướng Dẫn***

Cần Thơ, ngày … tháng … năm 2017

Giáo viên hướng dẫn

PGS.TS TRẦN CAO ĐỆ

# **LỜI CẢM ƠN**

**…………….🙣🕮🙡…………….**

Lời đầu tiên em xin chân thành cảm ơn quý thầy cô, gia đình và bạn bè đã tận tình giúp đỡ em trong suốt thời gian hoàn thành Niên luận vừa qua.

Em cũng xin chân thành gửi cảm ơn đến thầy Trần Cao Đệ, người đã hướng dẫn rất tận tình cũng như cho em nhiều góp ý bổ ích, giúp em hoàn thiện đề tài của mình. Nhân đây em cũng xin chúc thầy và gia đình luôn dồi dào sức khỏe và thành công hơn trong công việc. Xin cảm ơn thầy rất nhiều!

Trong thời gian thực hiện đề tài này, em đã cố gắn tìm hiểu, học hỏi để hoành thành niên luận. Mặc dù vậy, có lẽ không tránh khỏi những thiếu sót và hạn chế, rất mong các thầy cô và các bạn chỉ dẫn thêm.

Một lần nữa em xin cảm ơn gia đình, thầy cô và bạn bè và chúc cho mọi người sẽ gặt hái được nhiều thành công. Em xin chân thành cảm ơn!

Trần Lý Văn

**MỤC LỤC**

[LỜI CẢM ƠN 3](#_Toc478745603)

[MỤC LỤC 4](#_Toc478745604)

[CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI 6](#_Toc478745605)

[1.1 ĐẶT VẤN ĐỀ: 6](#_Toc478745606)

[1.2 PHẠM VI ĐỀ TÀI: 7](#_Toc478745607)

[1.3 PHƯƠNG PHÁP NGHIÊN CỨU: 7](#_Toc478745608)

[1.4 NỘI DUNG CHÍNH CỦA NIÊN LUẬN: 8](#_Toc478745609)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 9](#_Toc478745610)

[2.1 CÔNG NGHỆ RFID: 9](#_Toc478745611)

[CẤU TẠO: 9](#_Toc478745612)

[NGUYÊN LÝ HOẠT ĐỘNG: 9](#_Toc478745613)

[ĐỘ BẢO MẬT VÀ TIN CẬY: 10](#_Toc478745614)

[ỨNG DỤNG: 11](#_Toc478745615)

[2.2 LARAVEL FRAMEWORK: 12](#_Toc478745616)

[LARAVEL LÀ GÌ? 12](#_Toc478745617)

[LỊCH SỬ PHÁT TRIỂN CỦA LARAVEL 12](#_Toc478745618)

[CÁC ƯU ĐIỂM NỔI BẬC 14](#_Toc478745619)

[CẤU TRÚC THƯ MỤC CỦA LARAVEL 5.4 16](#_Toc478745620)

[NGUYÊN LÝ HOẠT ĐỘNG CỦA LARAVEL 18](#_Toc478745621)

[2.3 RESPONSIVEVOICE API: 19](#_Toc478745622)

[Hình 1. Các cổng an ninh siêu thị sử dụng RFID 6](file:///C:\Users\lyvam\Desktop\RFID\doc\report\RFID_Report_v1.0.docx#_Toc478678560)

[Hình 2. Minh họa hoạt động RFID 9](file:///C:\Users\lyvam\Desktop\RFID\doc\report\RFID_Report_v1.0.docx#_Toc478678561)

[Hình 3. Một số dạng thẻ RFID 10](file:///C:\Users\lyvam\Desktop\RFID\doc\report\RFID_Report_v1.0.docx#_Toc478678562)

[Hình 4. Đầu đọc RFID USB 125 Khz 10](file:///C:\Users\lyvam\Desktop\RFID\doc\report\RFID_Report_v1.0.docx#_Toc478678563)

1. **TỔNG QUAN VỀ ĐỀ TÀI**
   1. **ĐẶT VẤN ĐỀ:**

Trong cuộc sống hằng ngày, chúng ta vẩn thường xuyên tiếp xúc và sử dụng rất nhiều loại thiết bị nhận dạng tự động như: máy quét mã vạch, dấu vân tai, nhận dạng khuôn mặt, giọng nói…và một trong số đó là công nghệ mà chúng ta rất thường tiếp xúc. Trong các trạm giữ xe các bạn sẽ thấy một máy quét nhận dạng bằng bằng một thẻ cứng (người dùng giữ làm phiếu giữ xe), và đó chính là công nghệ RFID.

RFID được dùng để nhận dạng hàng hóa nhầm quản lý, thanh toán cho khách hàng và kiểm tra tình trạng hàng hóa đã được thanh toán hay chưa. (chức năng trong các cổng an ninh ở lối ra vào siêu thị)

**Hình 1. Các cổng an ninh siêu thị sử dụng RFID**

Mặc dù không còn quá mới, và rất thông dụng nhưng ít có người dùng nào biết được RFID chính là tên công nghệ mà họ đã sử dụng hằng ngày.

Để giúp mọi người hiểu hơn về công nghệ này. Đề tài niên luận này sẽ trình bày về công nghệ RFID và xây dưng một website đơn giản với các chức năng dựa trên công nghệ quen thuộc này.

* 1. **PHẠM VI ĐỀ TÀI:**

Xây dựng một website thực hiện việc đăng ký thông tin sinh viên với mã thẻ RFID và gọi tên sinh viên đó sau khi đã đăng kí. Website gồm các chức năng như sau:

* Đối với người dùng quản trị:
  + Quản lý một danh sách sinh viên cho trước.
  + Đăng ký thẻ cho sinh viên.
  + Chỉnh sửa kết quả đăng ký.
  + Tìm kiếm sinh viên.
* Đối với người dùng bình thường:
  + Gọi tên người dùng đã đăng ký khi quét thẻ.
  1. **PHƯƠNG PHÁP NGHIÊN CỨU:**

Về lý thuyết:

* Kỹ thuật phân tích thiết kế hệ thống.
* Phương pháp phân tích và thiết kế cơ sở dữ liệu.
* Năm vững ngôn ngữ PHP và hệ quản trị Mysql.
* Các kiến thức lập trình cơ bản.

Về kỹ thuật:

* Sử dụng PHP (5.6 trở lên) và HTML 5.0 cho việc xây dựng web.
* Sử dụng hệ quản trị cơ sở dữ liệu MySQL để quản lý cơ sở dữ liệu.
* Sử dụng BootStrap để thiết kế giao diện.
* Gói phần mềm XAMPP để tạo server php để thực thi và kiểm thử, gồm có: MySQL, PHP, phpMyAdmin.
* Tìm hiểu và sử dụng jquery để validate dữ liệu.
* Hiểu và sử dụng framework Laravel.

Về chức năng:

* Cho phép Admin cập nhật danh sách sinh viên một cách dễ dàng.
* Việc đăng thẻ được thực hiện tuần tự và đơn giản.
* Hổ trợ tìm kiếm thông tin sinh viên ở nhiều tiều chí (họ tên, mã số, số điện thoại, ngày sinh).
* Sử dụng chức năng gọi tên dựa trên một api text to speach đã nghiên cứu, không xây dừng lại từ điển âm thanh để đọc tiếng việt. (vd: responsivevoice api)
  1. **NỘI DUNG CHÍNH CỦA NIÊN LUẬN:**

Nhằm cung cấp thông tin một cách có hệ thống, niên luận sẽ được trình bày thành nhiều chương chương, mỗi chương chia thành nhiều đề mục nhằm giúp người đọc dễ hiểu.

Các nội dung bao gồm:

1. Giới thiệu tổng quan về đề tài, lý do chọn đề tài, phạm vi của đề tài và phương pháp thực hiện của đề tài.
2. Những cơ sở lý thuyết và công nghệ để thực hiện đề tài. Như RFID, Laravel, ResponsiveVoice api.
3. Phân tích, đặc tả yêu cầu đã liệt kê ở phần tổng quan, hoàn thành các mô hình hệ thống về chức năng và cơ sở dữ liệu.
4. Thiết kế giao diện, kiến trúc hệ thống.
5. Cài đặc và thử nghiệm hệ thống.
6. **CƠ SỞ LÝ THUYẾT**

Phần này trình bày tổng quát về công nghệ RFID và các kĩ thuật sử dụng để xây dựng website. Do đó, các nội dung có thể thiếu một số chí tiết như hướng dẫn sử dụng và đi sâu vào kĩ thuật. Các nội dung này các bạn có thể tìm đọc từ phần tham khảo (xem tại mục lục) trong quyển niên luận này.

1. **CÔNG NGHỆ RFID:**

RFID là viết tắt của cụm từ Radio Frequency Identification (Nhận dạng tần số sóng vô tuyến). là công nghệ nhận dạng đối tượng bằng sóng vô tuyến. Hai thiết bị này hoạt động thu phát sóng điện từ cùng tần số với nhau. Các tần số thường được sử dụng trong hệ thống RFID là 125Khz hoặc 900Mhz.

### **CẤU TẠO:**

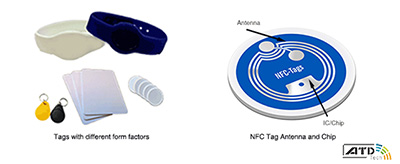
Một thiết bị hay một hệ thống RFID được cấu tạo bởi hai thành phần chính là thiết bị đọc ( reader) và thiết bị phát mã RFID có gắn chip hay còn gọi là tag. Thiết bị đọc được gắn antenna để thu-phát sóng điện từ, thiết bị phát mã RFID tag được gắn với vật cần nhận dạng, mỗi thiết bi RFID tag chứa một mã số nhất định và không trùng lặp nhau.

### **NGUYÊN LÝ HOẠT ĐỘNG:**

**Hình 2. Minh họa hoạt động RFID**

**Để hoạt động được** công nghệ RFID **cần có một bộ thu (reader) và một bộ phát tín hiệu (tag hay transponder)**, hai thiết bị này hoạt động cùng tần số với nhau, tại bộ phát một mã số nhận dạng được đính kèm trong quá trình phát tín hiệu.

Cấu tạo tag bao gồm 1 chip (chứa mã số nhận dạng + dữ liệu) và 1 antenna, chip được chế tạo không cần năng lượng để duy trì hoạt động. Khi tag được đưa lại gần đầu đọc nó sẽ nhận được 1 lượng năng lượng thông qua antenna (cơ chế hoạt động giống như sơ cấp và thứ cấp biến thế) và chuyển hóa năng lượng thành điện để kích hoạt chế độ hoạt động của chip, khi đó chip sẽ truyền mã số nhận dạng và dữ liệu ngược ra antenna và đầu đọc sẽ thu được mã số và dữ liệu của tag.



**Hình 3. Một số dạng thẻ RFID**

Hiện này bộ thu RFID đã hổ trợ cỗng USB, có thể nhận kết quả nhận dạng rồi trực tiếp truyền đến máy tính xử lý ở dạng text là id thu được thông qua cổng USB. Dưới đây ảnh của đầu đọc thẻ RFID 125 khz USB, cũng là thiết bị được sử dụng để xây dựng website của niên luận này.

**Hình 4. Đầu đọc RFID USB 125 Khz**

### **ĐỘ BẢO MẬT VÀ TIN CẬY:**

Thẻ chip (tag) RFID chứ rất nhiều mã nhận dạng khác nhau, thông thường là 32bit tương ứng với hơn 4 tỷ mã số khác nhau. Ngoài ra khi xuất xưởng mỗi thẻ chip RFID được gán một mã số khác nhau . Do vậy khi một vật được gắn chip RFID thì khả năng nhận dạng nhầm vật đó với 1 thẻ chip RFID khác là rất thấp, xác suất là 1 phần 4 tỷ.

Với ưu điểm về mặt công nghệ như vậy nên sự bảo mật và độ an toàn của các thiết bị ứng công nghệ RFID là rất cao.

### **ỨNG DỤNG:**

Ứng dụng tiêu biểu nhất của công nghệ này là chống mất trộm hàng hóa trong siêu thị. Thiết bị chip RFID (tag) được gắn với các hàng hóa trong đó. Thiết bị Reader và antenna được gắn bên ngoài cổng an ninh. Nếu một đồ vật chưa được tháo chip đi qua cổng an nình thì thiết bị Reader dễ dàng nhận dạng thấy và phát cảnh bảo.

Dùng RFID để sản xuất khóa thẻ từ. Khóa thẻ từ gồm 2 phần chính là ổ khóa và thẻ từ. Ổ khóa được lắp đặt tại cửa ra vào với nhiều tính năng thông minh như: tự động khóa khi đóng cửa, cảnh báo mọi người khi việc đóng cửa chưa hoàn tất, có âm thanh cảnh báo khi việc sử dụng thẻ chưa đúng…. Trong khi đó thẻ từ có vai trò tương tự như chiếc chìa khóa của ổ khóa truyền thống. Tuy nhiên thẻ để mở khóa từ đã được tích hợp con chip thông minh lưu giữ ID riêng của mã khóa đó, thiết bị đọc thẻ trên ổ khóa nhận dạng tự động những thẻ có ID đã được đăng ký từ trước với nó.

Một trong nhưng công dụng tuyệt vời nhất của công nghệ RFID là sử dụng trong  quản lý lưu thông hàng hóa. Dựa trên cơ sở kết hợp với Internet + GPRS + Cloud cho phép chúng ta theo dõi được món hàng được vận chuyển đã đi đến đâu trong suốt lộ trình vận chuyển. Đặc biệt với những hàng hóa quan trọng, vận chuyển xuyên biển, đường dài… giúp việc theo dõi, kiểm tra, giám sát trở nên cực kỳ đơn giản.

RFID cũng đặc biệt hữu ích cho những kho hàng với loại hàng nặng, cồng kềnh… Việc ứng dụng công nghệ RFID cho các kho hàng loại này, cuối mỗi ngày bạn chỉ cần bật thiết bị quét RFID lên nó sẽ giúp bạn thu thập dữ liệu tất cả các hàng hóa có dán nhãn RFID. Việc còn lại thật đơn giản, bạn chỉ cần đổ dữ liệu vào máy tính và phần mềm kiểm kho sẽ giúp bạn tất cả. Việc này giúp giảm rất nhiều chi phí quản lý kho hàng, kiểm kê kho hàng… tránh được nhiều thất thoát.

Một ứng dụng khác cũng được áp dụng công nghệ này mang đến lợi ích rất lớn là ứng dụng trong việc sản xuất khóa chống trộm xe máy. Việc áp dụng công nghệ RFID vào khóa chống trộm này sẽ làm cho những tên trộm gian manh nhất cũng phải bó tay.

1. **LARAVEL FRAMEWORK:**

### **LARAVEL LÀ GÌ?**

Laravel là một PHP framework  mã nguồn mở và miễn phí, được phát triển bởi Taylor Otwell và nhắm vào mục tiêu hỗ trợ phát triển các ứng dụng web theo kiếm trúc model-view-controller (MVC). Những tính năng nổi bật của Laravel bao gồm cú pháp dễ hiểu – rõ ràng , một hệ thống đóng gói modular 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.

Vào khoảng Tháng 3 năm 2015, các lập trình viên đã có một cuộc bình chọn PHP framework phổ biến nhất, Laravel đã giành vị trí quán quân cho PHP framework phổ biến nhất năm 2015, theo sau lần lượt là Symfony2, Nette, CodeIgniter, Yii2 vào một số khác. Trước đó, Tháng 8 2014, Laravel đã trở thành project PHP phổ biến nhất và được theo dõi nhiều nhất trên Github.

Laravel được phát hành theo giấy phép MIT, với source code được lưu trữ tại [Github](https://github.com/laravel/laravel).

### **LỊCH SỬ PHÁT TRIỂN CỦA LARAVEL**

Laravel được Taylor Otwell tạo ra như một giải pháp thay thế cho CodeIgniter, cung cấp nhiều tính năng quan trọng hơn  như xác thực và phân quyền. Tôi cũng không chắc về điều này, nhưng có thể Taylor vốn là một .NET developer khi bắt đầu có nhu cầu làm việc với PHP khoảng vào những năm 2010-2011, đã chọn CodeIgniter khi đó đang là một ngôi sao mới nổi, thậm chí lấn át cả Symfony gạo cội. Và Taylor nhanh chóng nhận ra những điểm khiếm khuyết ở CodeIgniter, với tài năng và kiến thức xuất sắc về design-pattern của mình, Taylor quyết định tự mình tạo ra một framework sao cho thật đơn giản, dễ hiểu, hỗ trợ lập trình viên hiện thực ý tưởng một cách nhanh nhất bằng nhiều tính năng hỗ trợ như Eloquent ORM mạnh mẽ, xác thực đơn giản, phân trang hiệu quả, và hơn thế nữa.

Bản Laravel beta đầu tiên được phát hành vào ngày 9/6/2011, tiếp đó là **Laravel 1** phát hành trong cùng tháng. Laravel 1 bao gồm các tính năng như xác thực, bản địa hóa, model, view, session, định tuyến và các cơ cấu khác, nhưng vẫn còn thiếu controller, điều này làm nó chưa thật sự là một MVC framework đúng nghĩa.

**Laravel 2** được phát hành vào tháng 9 năm 2011, mang đến nhiều cài tiến từ tác giả và cộng đồng. Tính năng đáng kể bao gồm hỗ trợ controller, điều này thực sự biến Laravel 2 thành một MVC framework hoàn chỉnh, hỗ trợ Inversion of Control ([IoC](https://en.wikipedia.org/wiki/Inversion_of_control)), hệ thống template Blade. Bên cạnh đó, có một nhược điểm là hỗ trợ cho các gói của nhà phát triển bên thứ 3 bị gỡ bỏ.

**Laravel 3** được phát hành vào tháng 2 năm 2012, với một tấn tính năng mới bao gồm giao diện dòng lệnh (CLI) tên “Artisan”, hỗ trợ nhiều hơn cho hệ thống quản trị cơ sở dữ liệu, chức năng ánh xạ cơ sở dữ liệu Migration, hỗ trợ “bắt sự kiện” trong ứng dụng,  và hệ thống quản lý gói gọi là “Bundles”. Lượng người dùng và sự phổ biến tăng trưởng mạnh kể từ phiên bản Laravel 3.

**Laravel 4**, tên mã “Illuminate”, được phát hành vào tháng 5 năm 2013. Lần này thực sự là sự lột xác của Laravel framework, di chuyển và tái cấu trúc các gói hỗ trợ vào một tập được phân phối thông qua Composer, một chương trình quản lý gói thư viện phụ thuộc độc lập của PHP. Bố trí mới như vậy giúp khả năng mở rộng của Laravel 4 tốt hơn nhiều so với các phiên bản trước. Ra mắt lịch phát hành chính thức mỗi sáu tháng một phiên bản nâng cấp nhỏ. các tính năng khác trong Laravel 4 bao gồm tạo và thêm dữ liệu mẫu (database seeding), hỗ trợ hàng đợi, các kiểu gởi mail, và hỗ trợ “xóa mềm”  (soft-delete: record bị lọc khỏi các truy vấn từ Eloquent mà không thực sự xóa hẳn khỏi DB).

**Laravel 5** được phát hành trong tháng 2 năm 2015, như một kết quả thay đổi đáng kể cho việc kết thúc vòng đời nâng cấp Laravel lên 4.3. Bên cạnh một loạt tính năng mới và các cải tiến như hiện tại, Laravel 5 cũng giới thiệu cấu trúc cây thư mục nội bộ cho phát triển ứng dụng mới. Những tính năng mới của Laravel 5 bao gồm hộ trợ lập lịch định kỳ thực hiện nhiệm vụ thông qua một gói tên là “Scheduler”, một lớp trừu tượng gọi là “Flysystem” cho phép điều khiển việc lưu trữ từ xa đơn giản như lưu trữ trên máy local – dễ thấy nhất là mặc định hỗ trợ dịch vụ Amazone S3, cải tiến quản lý assets thông qua “Elixir”, cũng như đơn giản hóa quản lý xác thực với các dịch vụ bên ngoài bằng gói “Socialite”.

**Laravel 5.1** phát hành vào tháng 6 năm 2015, là bản phát hành đầu tiên nhận được hỗ trợ dài hạng (LTS) với một kết hoạch fix bug lên tới 2 năm vào hỗ trợ vá lỗi bảo mật lên tới 3 năm. Các bản phát hành LTS của Laravel được lên kế hoạch theo mỗi 2 năm.

**Laravel 5.2** đã được phát hành tháng 12 năm 2015, nó có nhiều điểm mới so với phiên bản 5.1 như hỗ trợ nhiều trình điều khiển chứng thực, ràng buộc mô hình dữ liệu, cải tiến xác thực, hạn chế tốc độ xử lý trung gian.

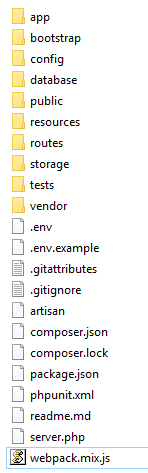
**Laravel 5.3** được phát hành vào ngày 23 tháng 8 năm 2016, Các tính năng mới trong 5,3 được tập trung vào việc cải thiện tốc độ phát triển bằng cách bổ sung thêm những cải tiến cho các tác vụ thường dùng.

**Laravel 5.4** phiên bản mới nhất tại thời điểm này được phát hành vào ngày 24 tháng 1 năm 2017, cũng phiên bản được dùng cho niên luận này. Bản phát hành này có nhiều tính năng mới, như Laravel Dusk, Laravel Mix, Markdown Email, Facades tự động, cải tiến Route và nhiều thứ khác.

### **CÁC ƯU ĐIỂM NỔI BẬC**

1. **Quick Emergence**: Sự ra đời của Laravel như một luồng gió mới thổi vào cộng đồng framework PHP. Nó có thể giải quyết tốt những web project lớn. Rất nhiều công ty đã sử dụng nó để phát triển các sản phẩm cho công ty của họ. Như hình bên trên bạn có thể nhìn thấy nó chiếm phần lớn nhất trong "miếng bánh" framework PHP.
2. **Open Source**: Nó mở và hoàn toàn free. Bạn chẳng phải quan tâm đến việc trả phí khi ứng dụng web của bạn to lên. Điều bạn cần làm là hiểu nó và code những tính năng tuyệt với mà thôi.
3. **Templete engine**: Nó sử dụng Blade template và các công cụ như Elixir giúp bạn dễ dàng quản lý các assets( biên dịch, ghép nối, nén ...), cũng như chia nhỏ code HTML ra thành nhiều phần. Giúp cho việc thiết kế cũng như quản lý các assets trở nên vô cùng đơn giản.
4. **Modular**: Laravel được xây dựng dựa trên hơn 20 thư viện khác nhau. Hiểu được cách thiết kế framework khiến các dev hoàn toàn có thể đóng góp cho framework cũng như mở rộng chúng một cách dễ dàng. Dễ hiểu khi Laravel luôn có cộng đồng support đông đảo và "trâu" như vậy.
5. **MVC Architecture Support**:  Laravel sử dụng MVC pattern để xây dựng framework. Nó đảm bảo rằng các code được trình bày logic và hợp lý. Dễ dàng để có thể hiểu được nó. Vì vậy nó làm cho web project của bạn được trở nên dễ quản lý cũng như phát triển rất rất nhiều.
6. **Libraries and configuration**: Laravel tạo điều kiện cho các developer có thể thay đổi tùy chọn và cấu hình để phù hợp với nhiều môi trường. Các developer cũng có thể viết các gói ( package ) có thể là một tính năng hay một đoạn code gì đó. Nó dễ dàng được khai báo trong Laravel thông qua các provider. Chính điều này  + với số lượng các developer đông đảo thì số lượng package mà Laravel có ngày càng tăng lên. Bạn có thể dễ dàng tìm được một vài đoạn code cũng như một vài tính năng mà bạn mong muốn trong project của bạn chỉ trong vài phút. Nó tiết kiệm hơn so với việc phải ngồi viết code cho tính năng đó.

### **CẤU TRÚC THƯ MỤC CỦA LARAVEL 5.4**

1. **Thư mục gốc (root directory)**

**app**: chứa code chính (core code) của ứng dụng. Hầu hết các Class đều nằm trong thư mục này.

**bootstrap**: chứa các file bootstrap của framework và cấu hình autoload. Ngoài ra thư mục này còn chứa thư mục cache, thư mục cache chứa các file được tạo ra nhằm tăng hiệu năng của laravel như route và các file services cache.

**config**: "cái tên nói lên tất cả". Thư mục này chứa tất cả cái file cấu hình ứng dụng.

**database**: chứa các file migration và seed. Ngoài ra các bạn có thể đặt SQLite database vào trong thư mục này.

**public**: chứa file index.php. Tất cả mọi request tới ứng dụng đều phải qua file này xử lý. Ngoài ra thư mục public còn có thể chứa các file như: images, JavaScript, and CSS.

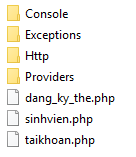
**resources**: chứa các file view và các file chưa được biên dịch như LESS, SASS, or JavaScript. Thư mục này còn chứa các file ngôn ngữ.

**routes**: chứa các định nghĩa về route của ứng dụng. Laravel mặc định có 3 file là web.php, api.php, và console.php.

**storage**: chứa các file đã được biên dịch như các file template, session, cache...

**tests**: chứa các case test.

**vendor**: chứa Composer dependencies(các gói của các nhà cung cấp sẽ được tải về thư mục này)

1. **Thư mục ứng dụng (app directory)**

**Console**: chứa toàn bộ các câu lệnh Artisan, lệnh này có thể được tạo bởi lệnh make:command.

**Events**: mặc định thì không có thư mục này, nhưng thư mục events sẽ được tạo ra khi chạy lệnh event:generate và make:event.

**Exceptions**: chứa các xử lý về exception.

**Http**: chứa controllers, middleware, và form requests. Hầu hết các xử lý logic về request đều đặt trong thư mục này.

**Jobs**: được tạo mặc định, thư mục Jobs sẽ được tạo khi chạy lệnh make:job. Thư mục này chứa các queueable jobs.

**Listeners**: được tạo khi chạy lệnh event:generate hoặc make:listener. Thư mục này chứa các Class xử lý các event.

**Mail**: được tạo khi chạy lệnh make:mail. Thư mục chứa các Class xử lý tác vụ liên quan đến việc gửi mail.

**Notification**: được tạo khi chạy lệnh make:notification. Nó chứa các "transactional" liên quan đến thông báo. Laravel cung cấp sẵn một số driver về việc gửi thông báo như email, Slack, SMS, hoặc lưu lại database.

**Policies**: được tạo khi chạy lệnh make:policy. Thư mục chứa Class về chính sách cấp phép (authorization policy). Xem thêm chi tiết tại authorization documentation.

**Providers**: chứa toàn bộ service providers của ứng dụng.

### **NGUYÊN LÝ HOẠT ĐỘNG CỦA LARAVEL**

Khi tương tác với ứng dụng Laravel, từ trình duyệt người dùng sẽ gởi một yêu cầu đến web server, web server nhận yêu cầu và chuyển cho bộ định tuyến (Routing) của Laravel. Bộ định tuyến của Laravel nhận được yêu cầu và chuyển hướng đến phương thức của lớp Controller thích hợp dựa trên khuôn mẫu định tuyến URL.

Lúc này  Controller sẽ thực hiện việc  điều hành của mình nhằm có phản hồi phù hợp với tương tác từ phía người dùng. Trong một số trường hợp, Controller ngay lập tức render một View và gởi trả kết quả cho trình duyệt.Thông thường, đối với các trang web động, Controller tương tác với Model, tương tác với cơ sở dữ liệu,lấy các thông tin cần thiết, cuối cùng mới thực hiện render một View và gởi kết quả về cho trình duyệt.

1. **RESPONSIVEVOICE API:**