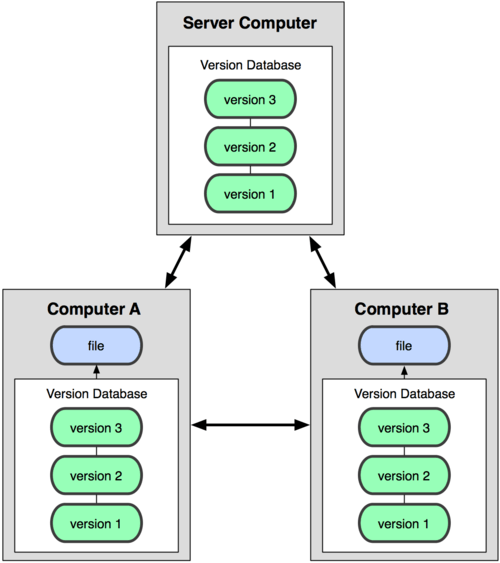
**Nội dung :**

**Git ->**

**CHƯƠNG 2: NỘI DUNG CÔNG VIỆC THỰC TẬP TẠI** **CÔNG TY**

* 1. **Tiến độ thực hiện công việc**
  2. **Quá trình thực hiện công việc.**
     1. **Tìm hiểu về git và một số lệnh cơ bản.**
* **Git là gì?**

Git là tên gọi của một Hệ thống quản lý phiên bản phân tán (Distributed Version Control System – DVCS) là một trong những hệ thống quản lý phiên bản phân tán phổ biến nhất hiện nay. DVCS nghĩa là hệ thống giúp mỗi máy tính có thể lưu trữ nhiều phiên bản khác nhau của một mã nguồn được nhân bản (clone) từ một kho chứa mã nguồn (repository), mỗi thay đổi vào mã nguồn trên máy tính sẽ có thể ủy thác (commit) rồi đưa lên máy chủ nơi đặt kho chứa chính. Và một máy tính khác (nếu họ có quyền truy cập) cũng có thể clone lại mã nguồn từ kho chứa hoặc clone lại một tập hợp các thay đổi mới nhất trên máy tính kia. Trong Git, thư mục làm việc trên máy tính gọi là Working Tree.

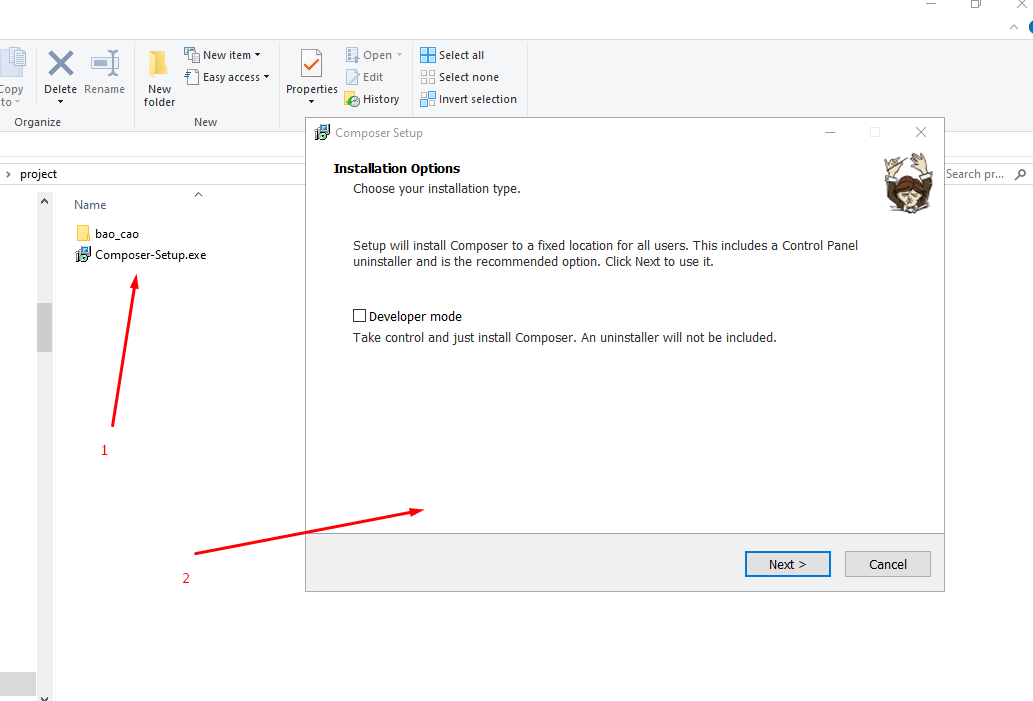


1. Hình 1: Tổng quan về git.

Ngoài ra, có một cách hiểu khác về Git đơn giản hơn đó là nó sẽ giúp bạn lưu lại các phiên bản của những lần thay đổi vào mã nguồn và có thể dễ dàng khôi phục lại dễ dàng mà không cần copy lại mã nguồn rồi cất vào đâu đó. Và một người khác có thể xem các thay đổi của bạn ở từng phiên bản, họ cũng có thể đối chiếu các thay đổi của bạn rồi gộp phiên bản của bạn vào phiên bản của họ. Cuối cùng là tất cả có thể đưa các thay đổi vào mã nguồn của mình lên một kho chứa mã nguồn.

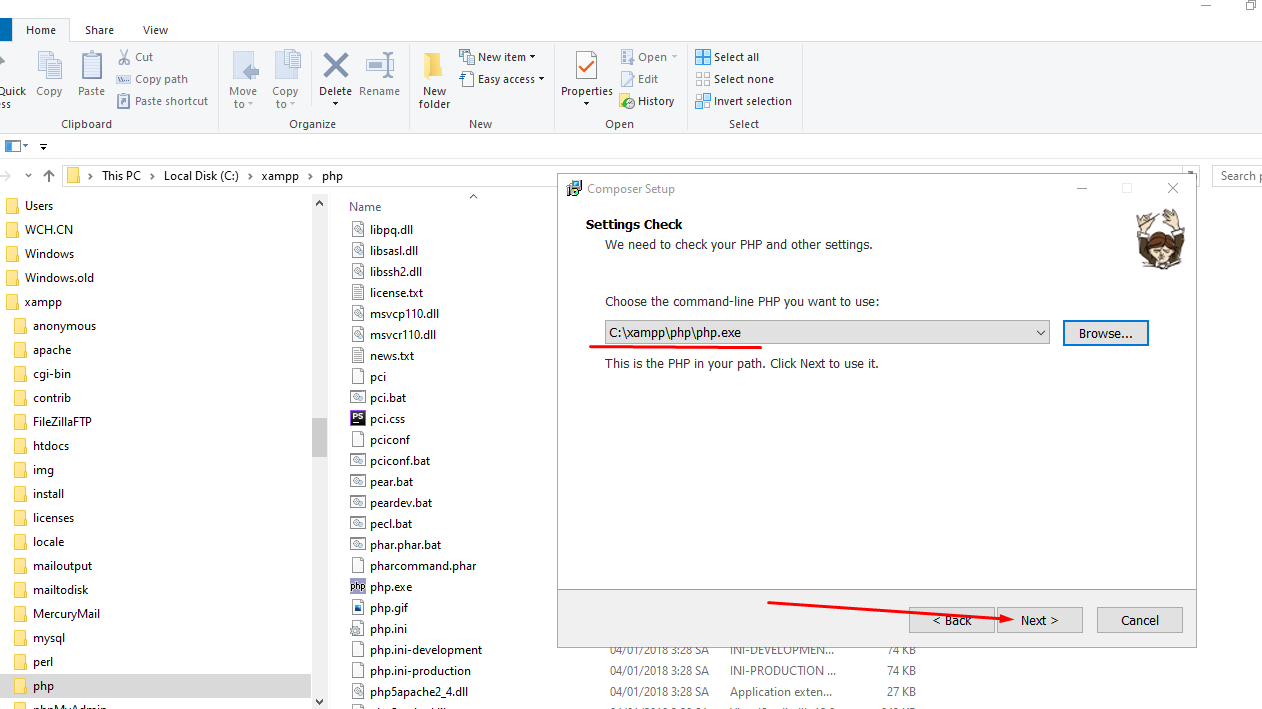
* + 1. **Cấu hình project Laravel trên local.**
* **Cài composer cho windows:**

Để cài Composer thì tải composer trên <https://getcomposer.org/>

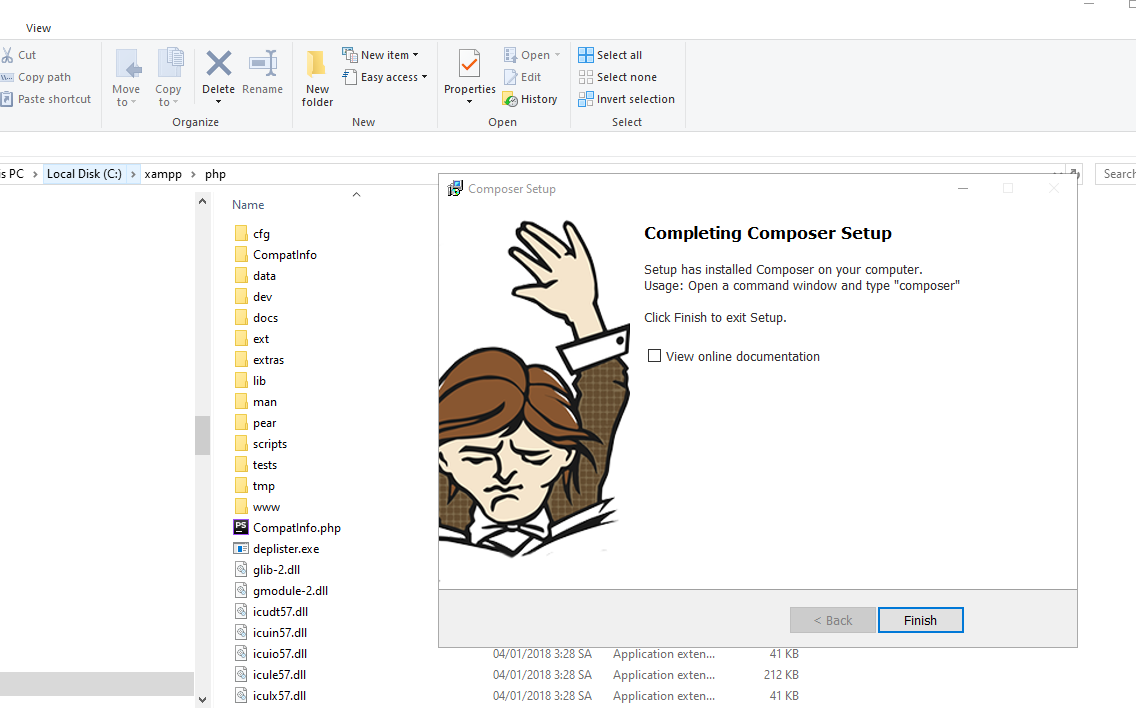


**Hình 2.2.2.1 file cài đặt đã tải về và bắt đầu cài cài vào máy**

Để cài composer trên window thì phải cài xampp hoặc wampserver, bây giờ em cài composer theo xampp



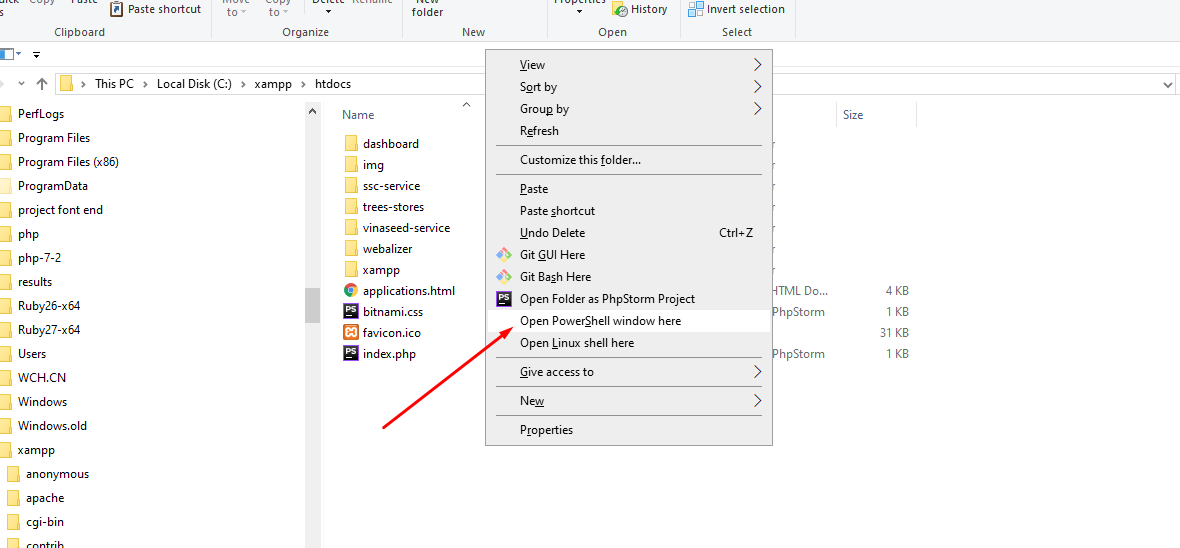
**Hình 2.2.2.2: Cài Composer**



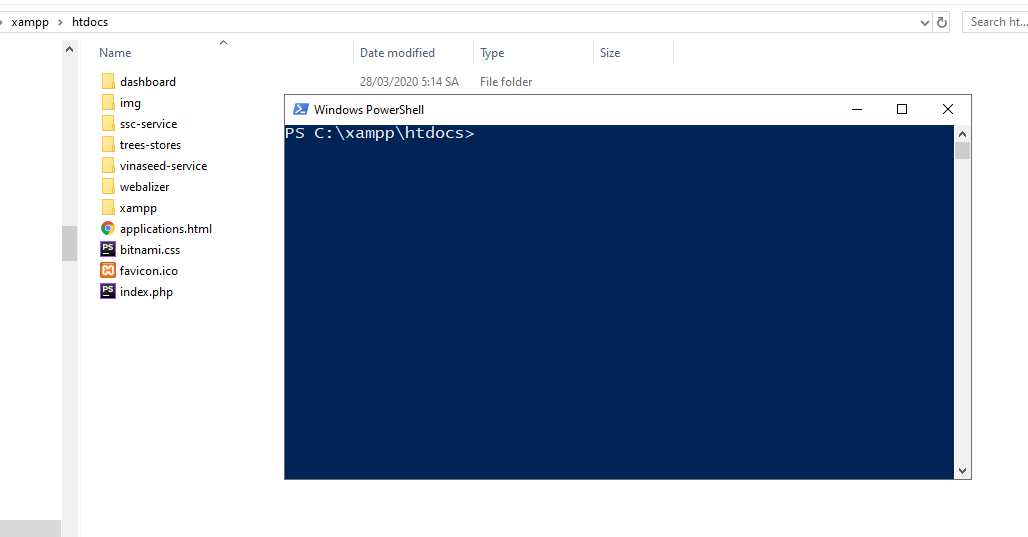
**Hình 2.2.2.3: Cài xong Composer**

* **Cài đặt Framework laravel:**

Vào thư mục htdocs trong xampp và bắt đầu cài laravel:

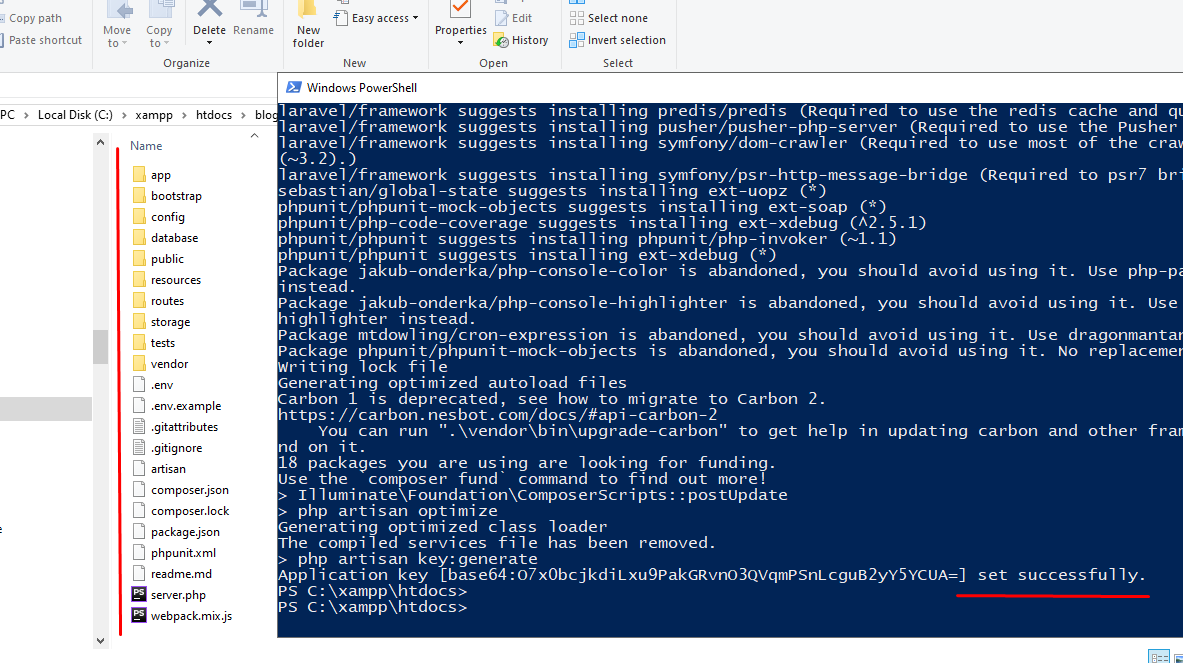


**Hình 2.2.2.4: Chay windows PowerShell**



**Hình 2.2.2.5: Giao die6nj windows PowerShell**

Để tạo project bằng Framework laravel dùng câu lệnh: composer create-project --prefer-dist laravel/laravel blog và Chữ blog là tên project mà em muốn tạo.

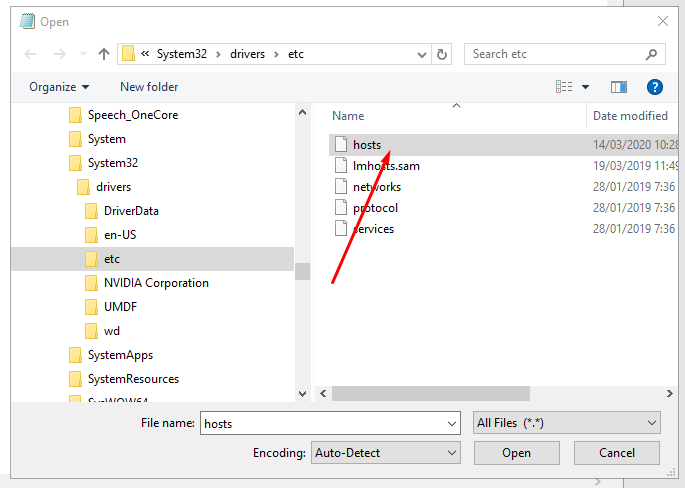


**Hình 2.2.2.6: Tạo xong project blog bằng laravel**

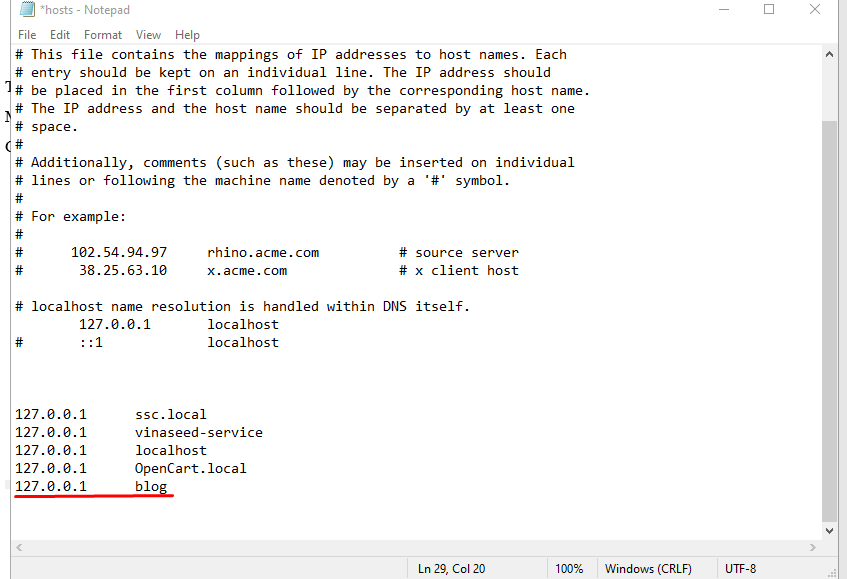
* **Tạo đường dẫn để chạy project blog:**

Mở Notepad bằng run as administrator -> chọn file hosts trong thư mục :

C:\Windows\System32\drivers\etc



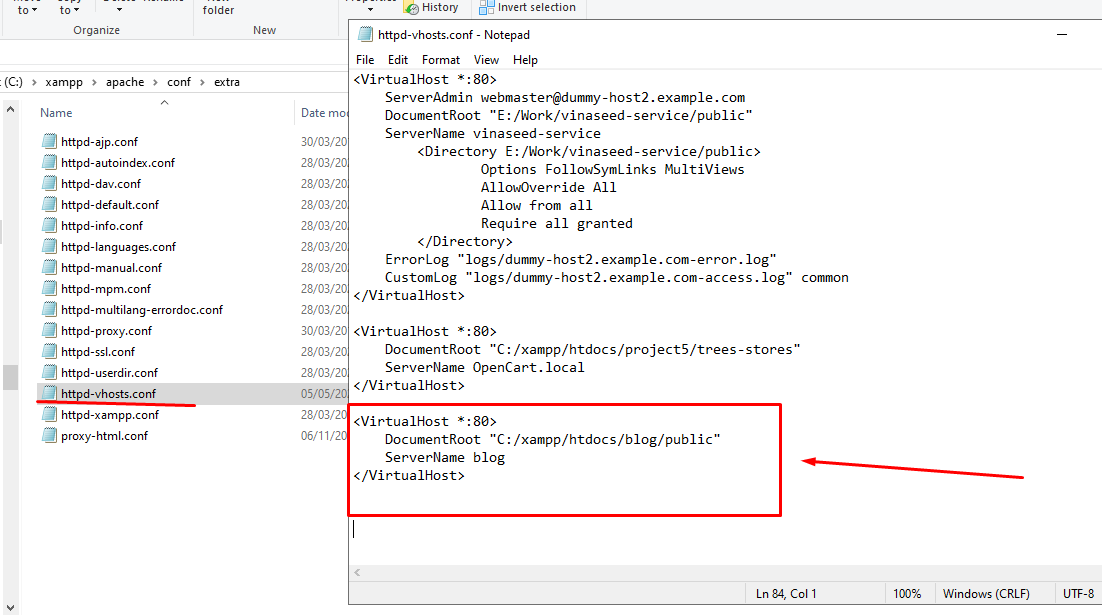
**Hình 2.2.2.7: Mở file hosts**



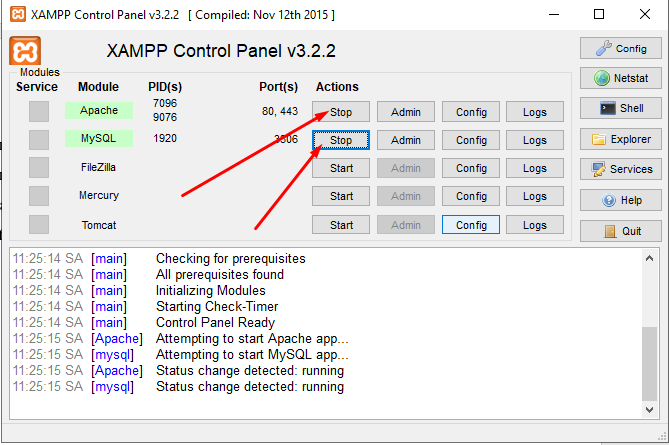
**Hình 2.2.2.8: Thêm dòng 123.0.0.1 blog vao cuối file và lưu lại**

Mở file httpd-vhosts.conf trong thư mục của xam: C:\xampp\apache\conf\extra

Thêm đoạn mã như trong **Hình 2.2.2.9** và lưu lại

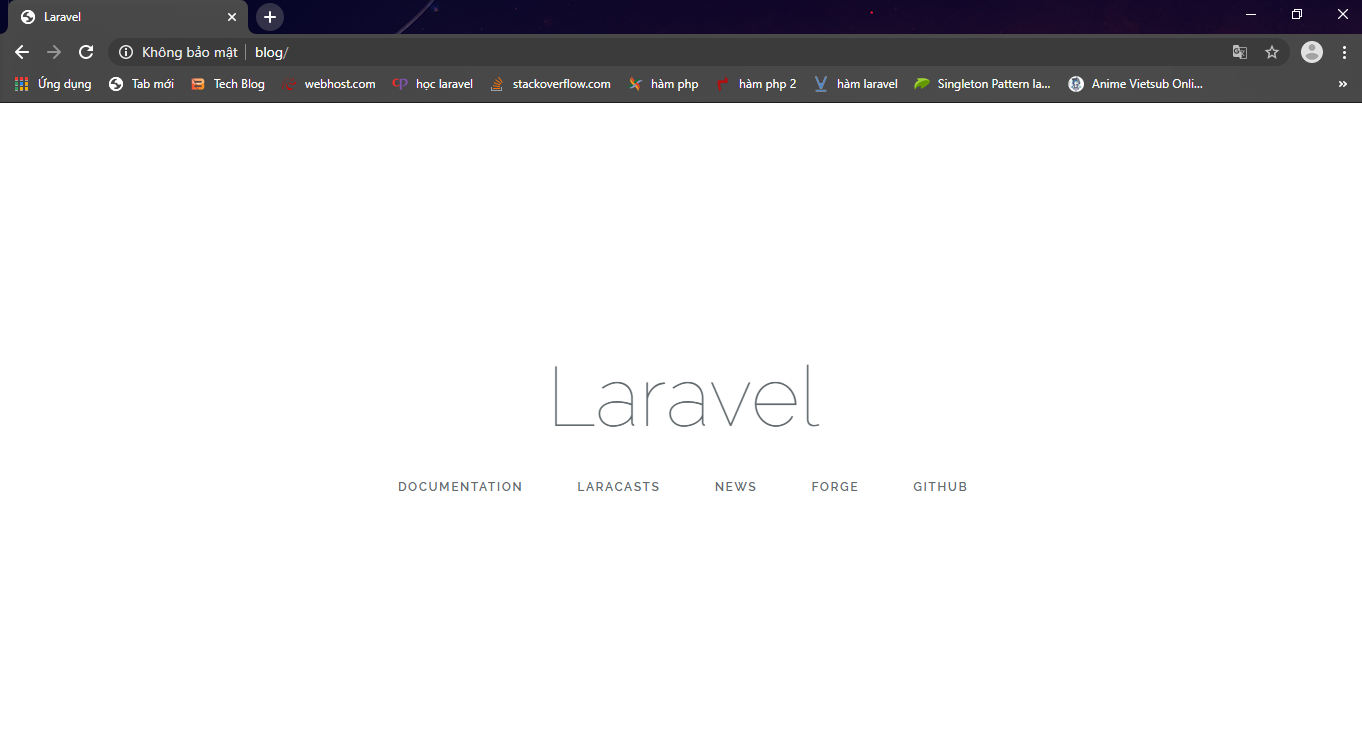


**Hình 2.2.2.9: Chọn ServerName để chạy project blog**



**Hình 2.2.2.10: Khởi động xampp**

Vào trình duyệt chạy : <http://blog/>



**Hình 2.2.2.11: Hoàn tất việc cấu hình laravel chạy trên local**

* + 1. **Tìm hiểu laravel cơ bản và bài test.**
    2. **Tìm hiểu về Laravel Lumen và tham gia dự án thực tế.**
       1. **Framework Lumen**

****

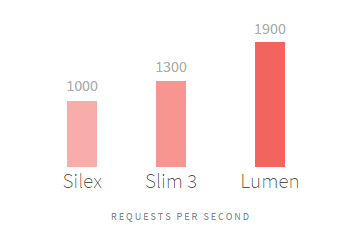
**Hình 2.2.4.1.1 Framework Lumen**

Laravel là một framework hiện đại của PHP, mang rất nhiều những ưu điểm các các framework hiện đại từ các ngôn ngữ khác như Rails / Spring, có thể coi là framework dễ dùng và thích dùng nhất đối với PHP trong thời điểm hiện tại.

Tuy vậy Laravel vẫn có một nhược điểm và số lượng module dùng tương đối lớn, dẫn đến tốc độ không được tối ưu khi xây những service nhỏ. Vì thế Taylor Otwell đã release một phiên bản microframework là Lumen(Lumen được tối ưu hóa cho việc viết API và mô hình hướng dịch vụ — SOA).

* Điểm mạnh của Lumen :

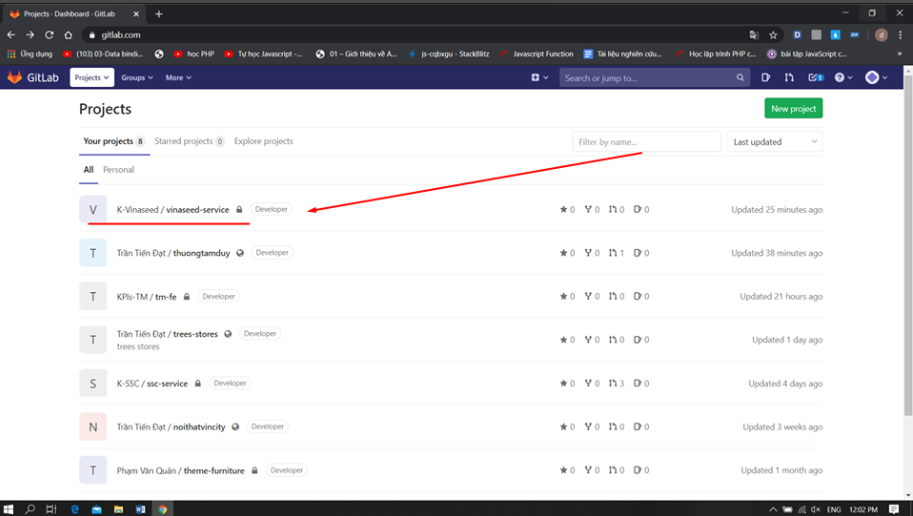
Theo như giới thiệu ở trang chủ thì Lumen gần như đã đạt tốc độ vượt trội so với 2 micro framework khác là Silex (micro framework của Symfony) và Slim. Số lượng request trên 1s gần như gấp đôi so với Silex và gấp rưỡi so với Slim. Tuy nhiên so sánh với Phalcon (framework dùng C để compile) thì không được nhắc đến.



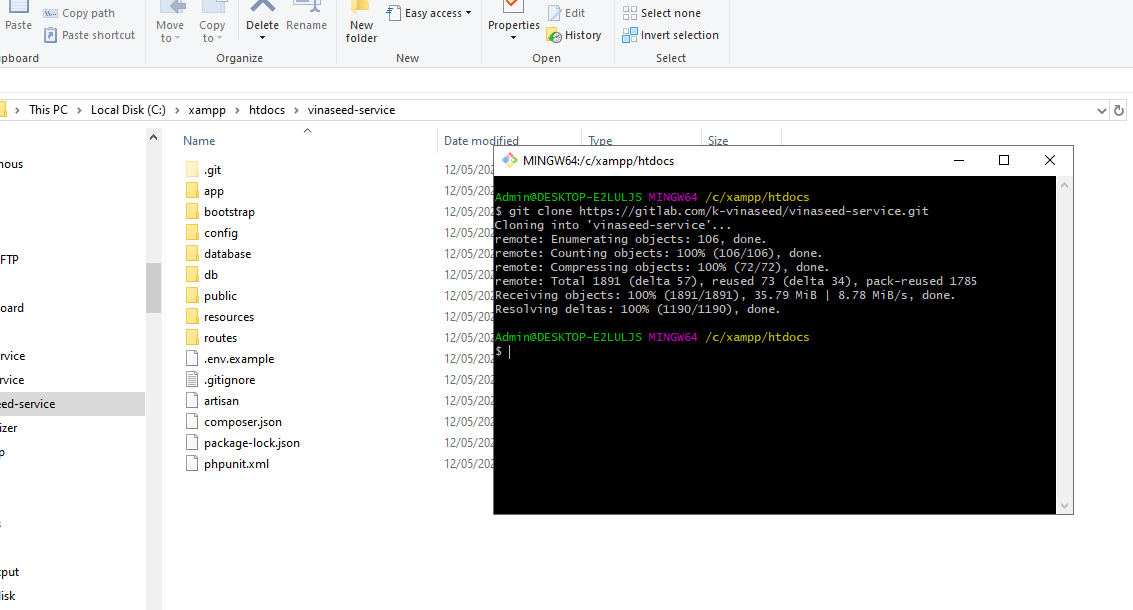
**Hình 2.2.4.1.2 Biểu đồ so sánh**

Và ưu điểm còn lại là mọi module cơ bản đều của Laravel, vì vậy nếu bạn đã quen với Laravel thì bắt đầu với Lumen gần như không tốn nhiều thời gian. Hơn thế nữa khi cần nâng cấp thành một framwork mạnh mẽ hơn thì việc intergrate vào Laravel sẽ vô cùng đơn giản.

* + - 1. **Clone code git**

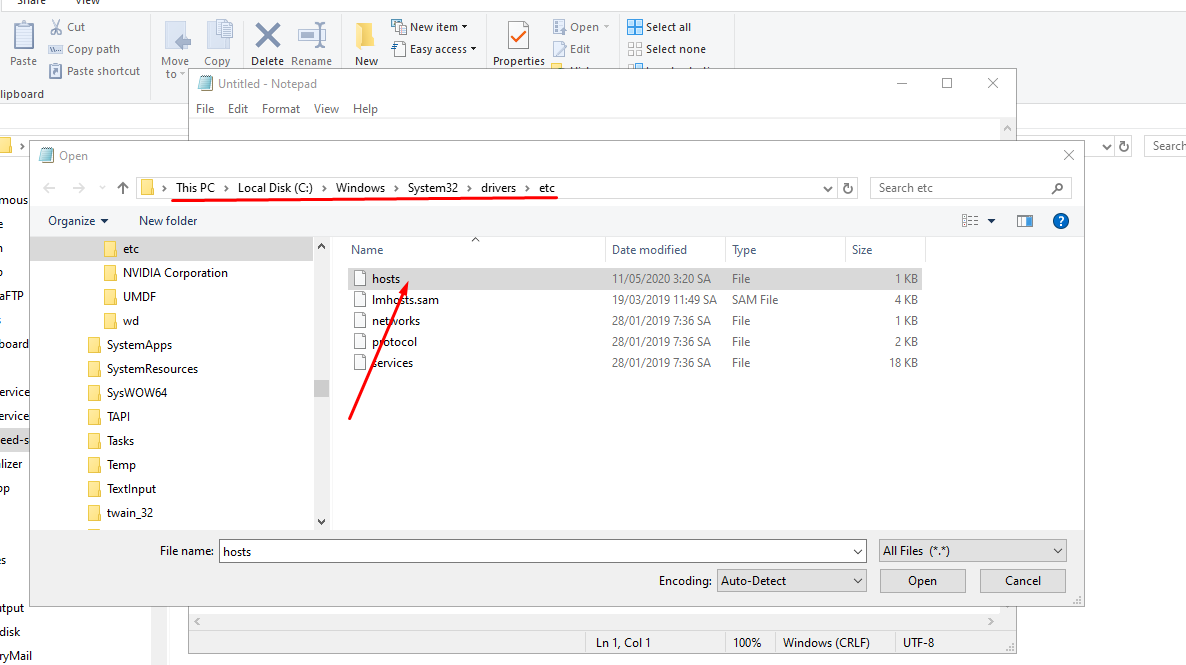


**Hình 2.2.4.2.1 Clone project vinaseed-service từ gitlab.com.**

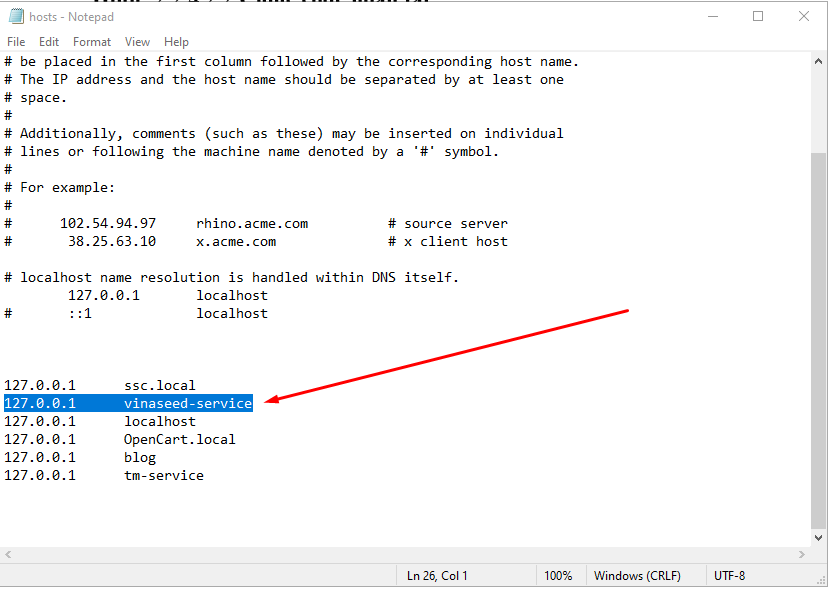


**Hình 2.2.4.2.2 Clone code hoàn tất.**

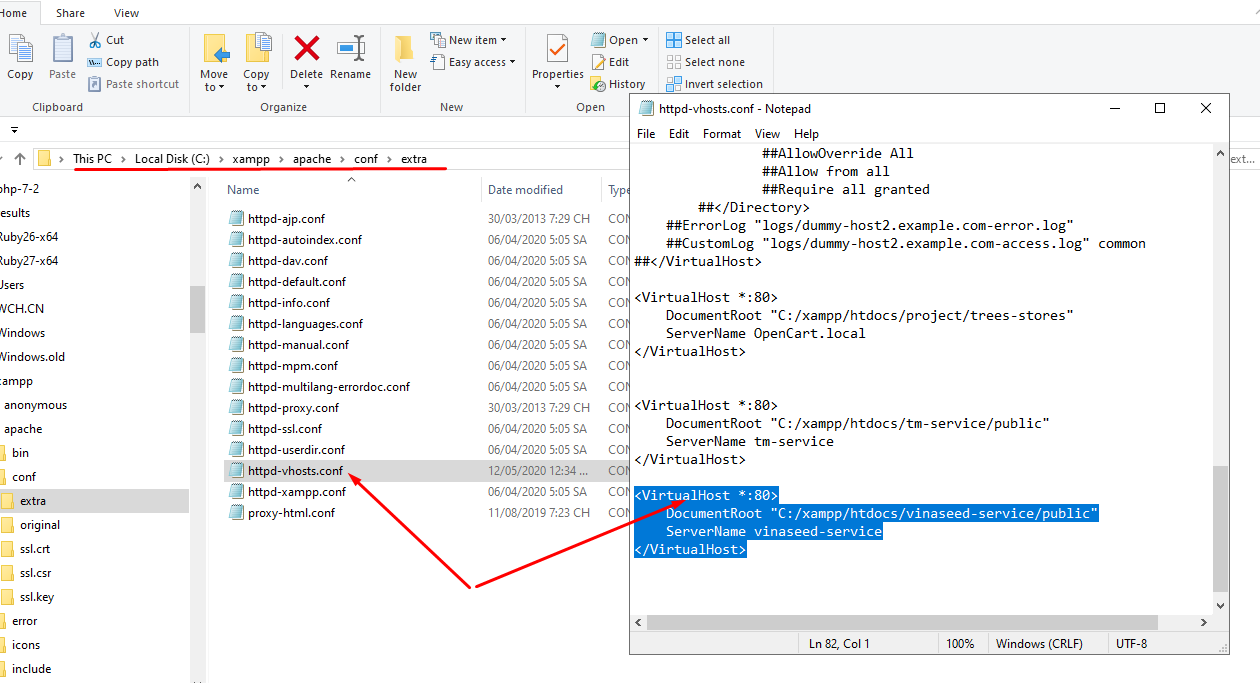
* + - 1. **Cấu hình project trên local**
* **Cấu hình đường dẫn chạy project**



**Hình 2.2.4.3.1 Mở file hosts bằng notepad với run administrator.**



**Hình 2.2.4.3.2 Chỉnh sửa file hosts.**

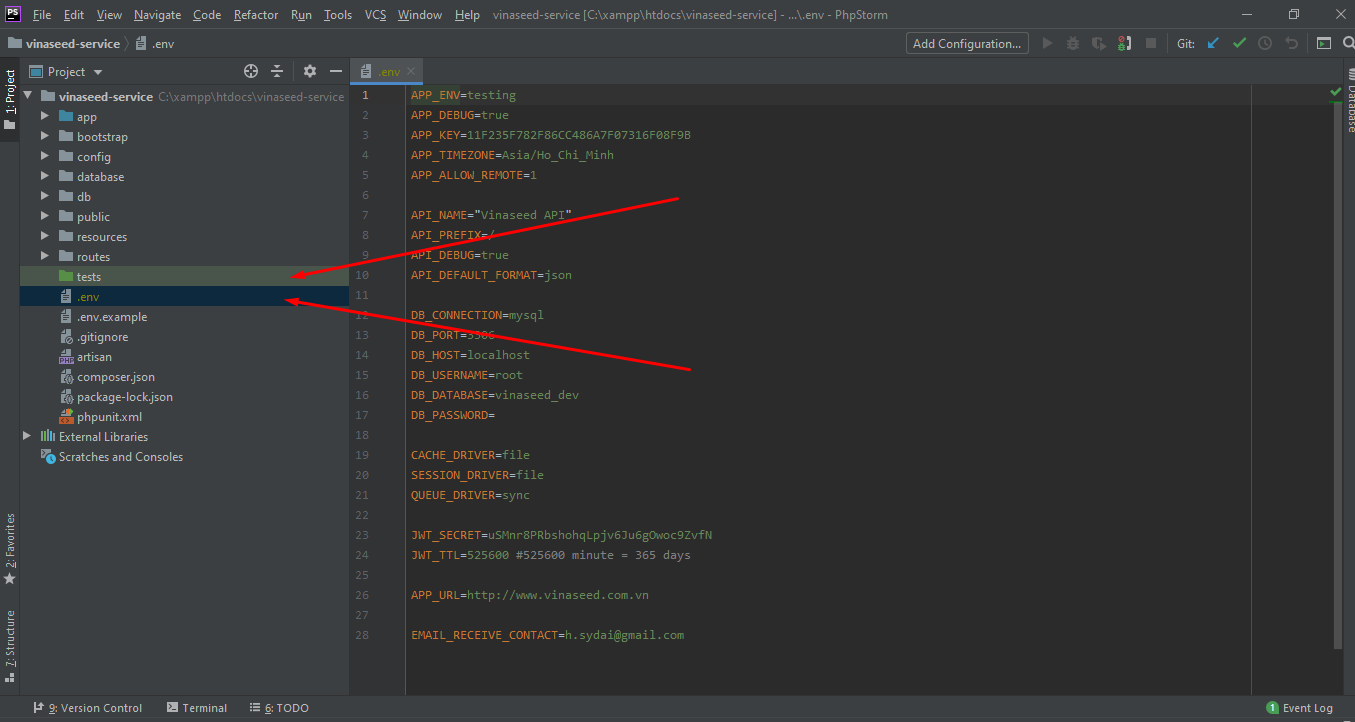


**Hình 2.2.4.3.3 Chỉnh sửa file httpd-vhosts.conf trong xampp.**

* **Cấu hình Chạy project**

**Coppy file .env.example và đổi tên thành .env đồng cấp với nhau.**

**Tạo folder tests**



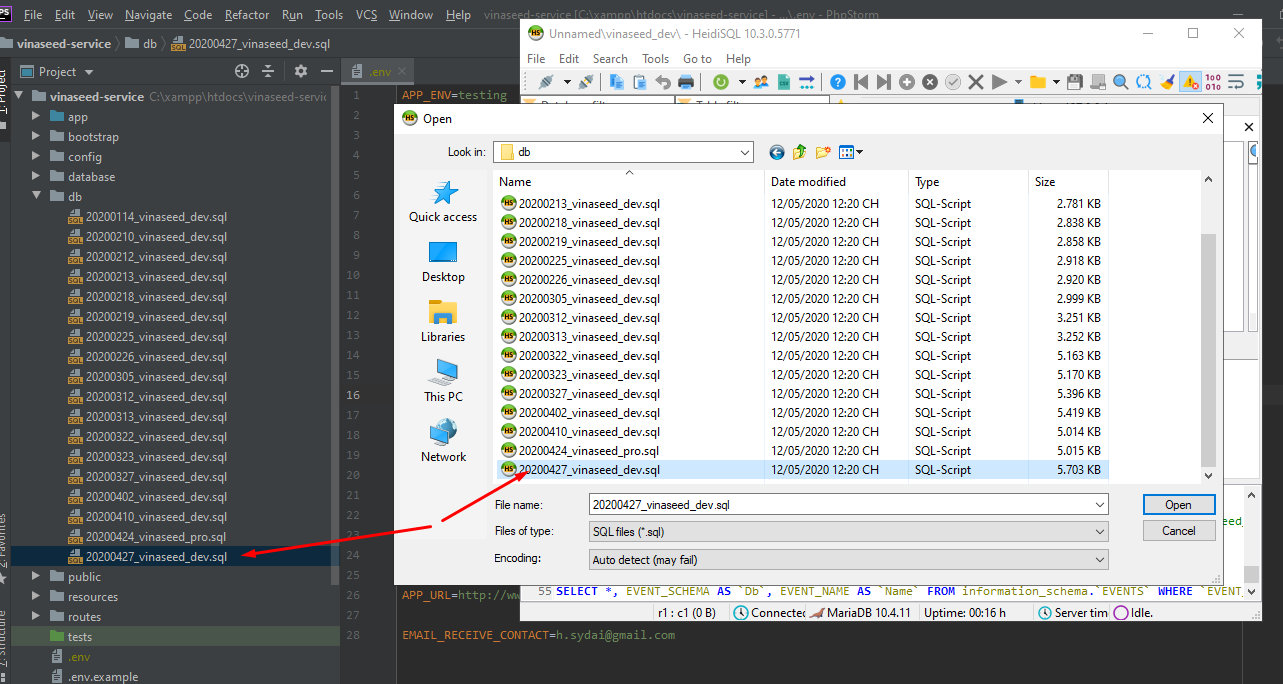
**Hình 2.2.4.3.4 Cấu hình chạy project**

**Tạo database cho project**

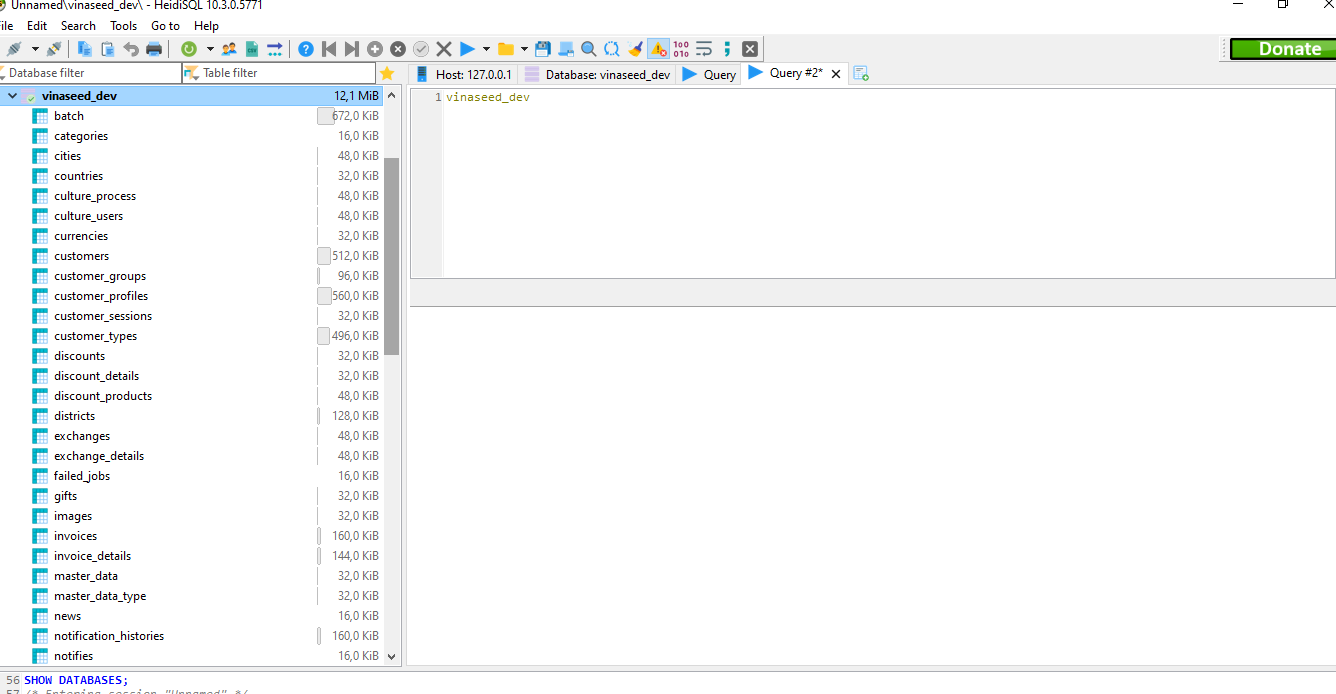


**Hình 2.2.4.3.5 Tạo database cho project.**

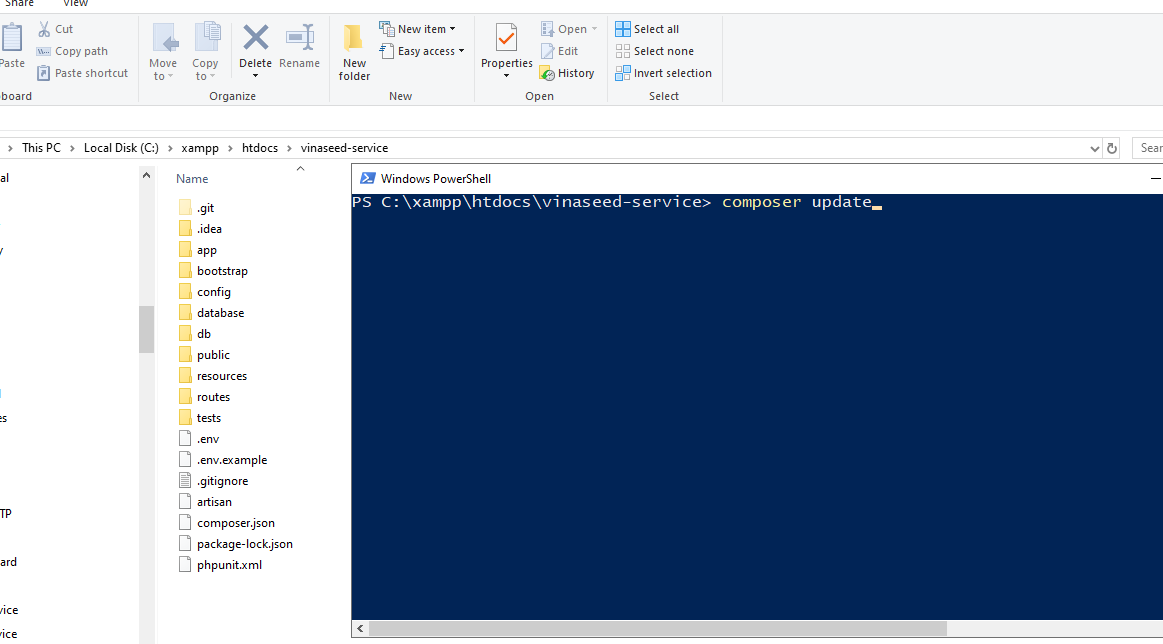
**Chạy database sql mới nhất**



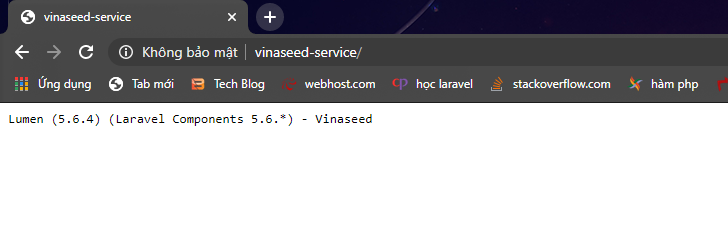
**Hình 2.2.4.3.6 Chạy file sql**



**Hình 2.2.4.3.7 Chạy file sql hoàn tất**

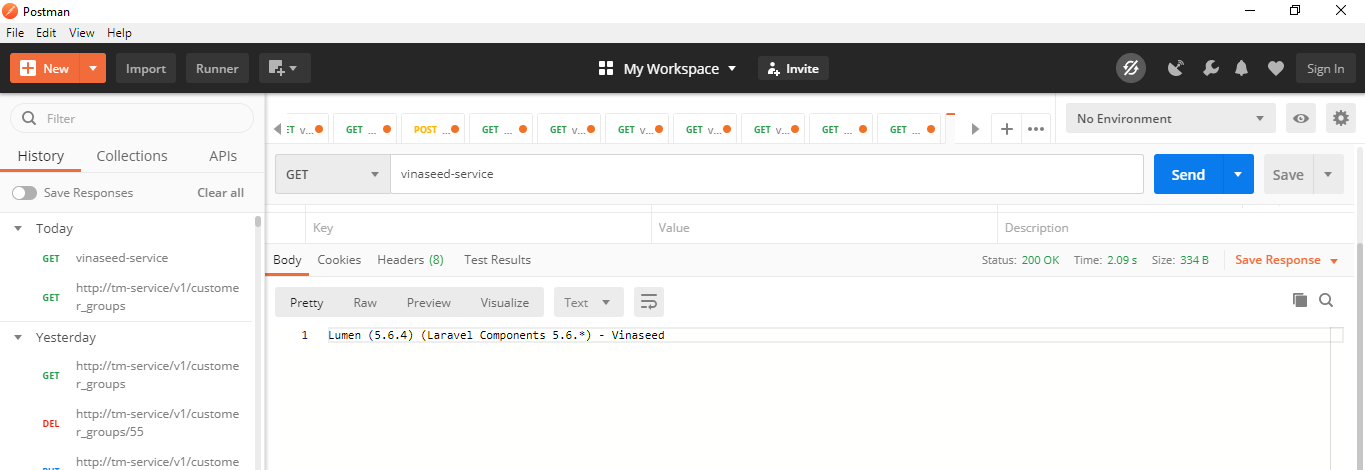


**Hình 2.2.4.3.8 Chạy composer update cho project**



**Hình 2.2.4.3.9 Hoàn thành chạy project thành công**

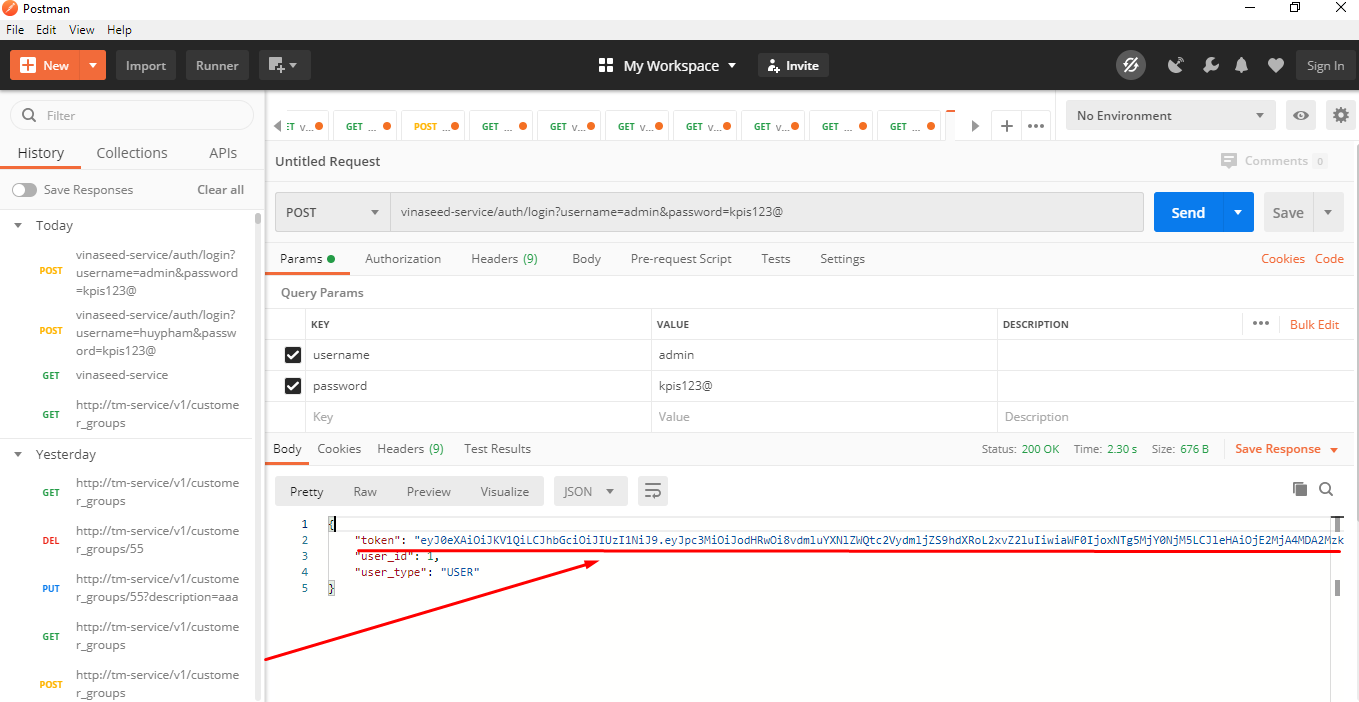
* **Test chạy project trên postman**



**Hình 2.2.4.3.10 Chạy project trên postman**

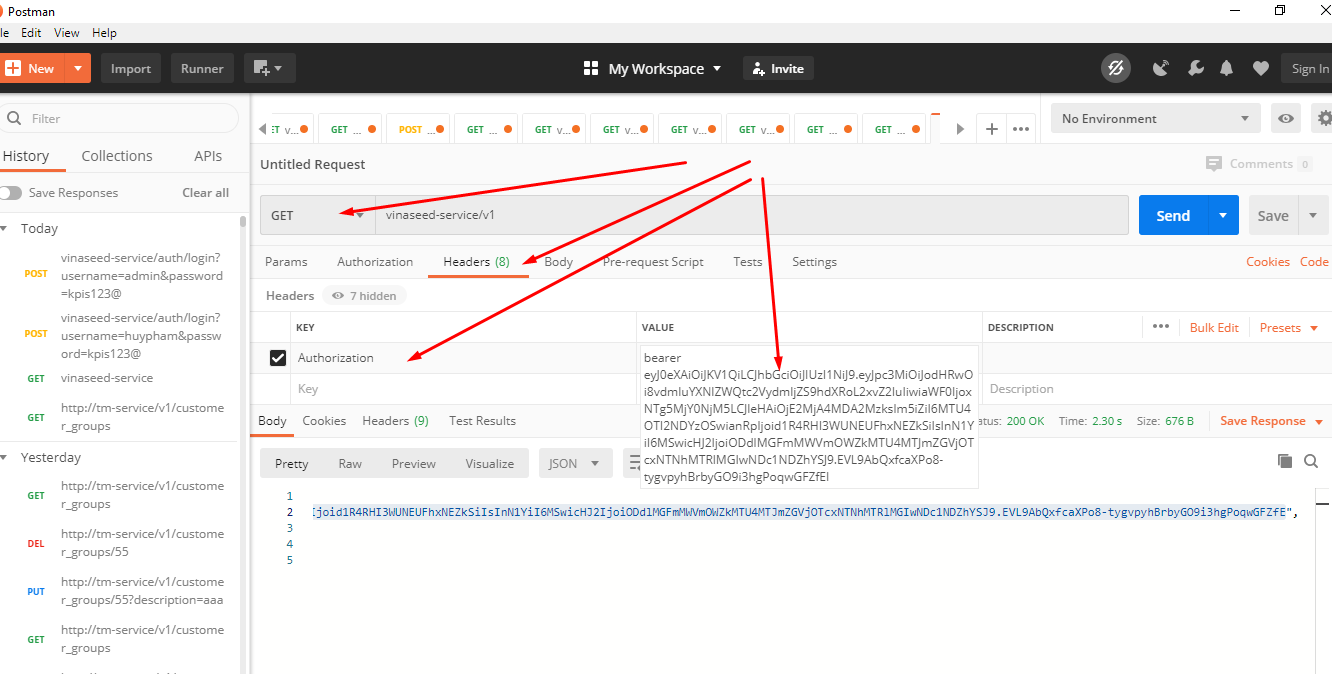
* **Chạy thử một API của project**

**Lấy token của project dùng phương thức post -> với link vinaseed-service/auth/login -> prams truyền vào gồm có username: admin, password: kpis123@ -> send**

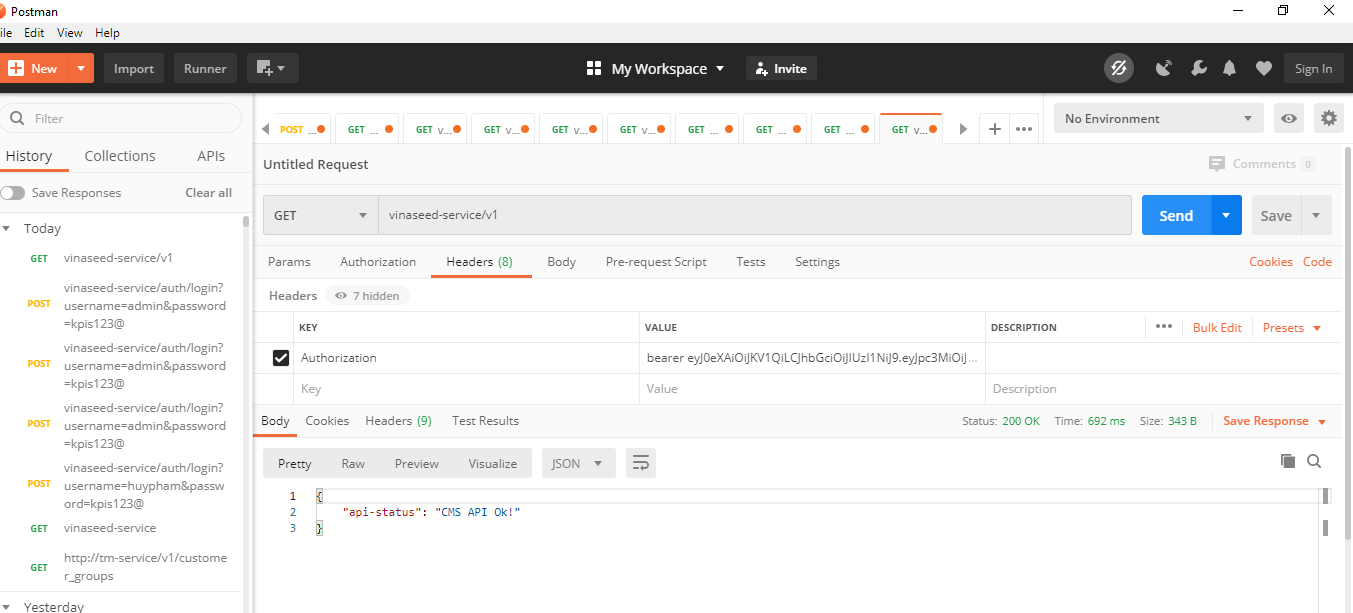


**Hình 2.2.4.3.11 Lấy token**

**Dùng phương thức get -> chọn Headers -> chọn Authorization -> nhập bearer cách ra nhập đoạn token lấy được ở Hình 2.2.4.3.11 -> send**



**Hình 2.2.4.3.12 Thực hiện token**



**Hình 2.2.4.3.13 Hoàn tất token**

**Chạy test thử một API show các users**

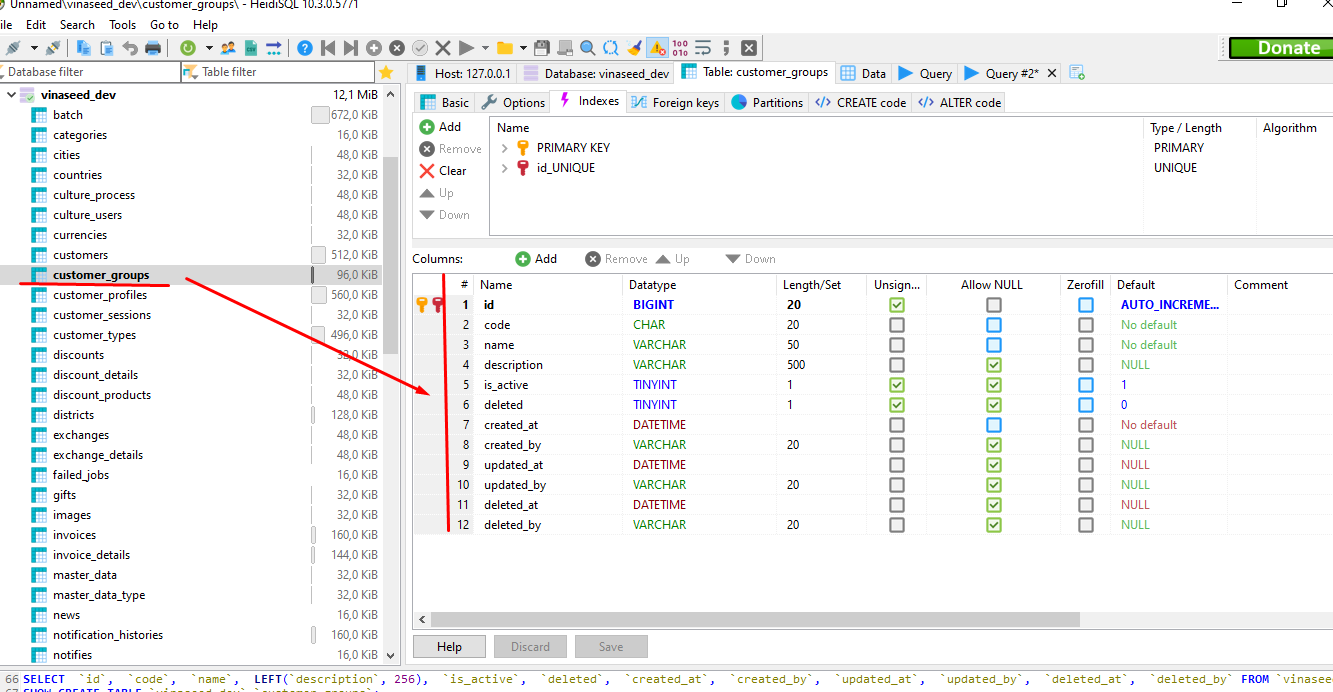


**Hình 2.2.4.3.14 Hoàn thành test 1 API trên postman**

* + - 1. **Nội dung trong quá trình thực tập mà đơn vị cho chúng em làm:**
* **Viết API mới với 4 chức năng CRUD (Create, Read, Update,Delete)**

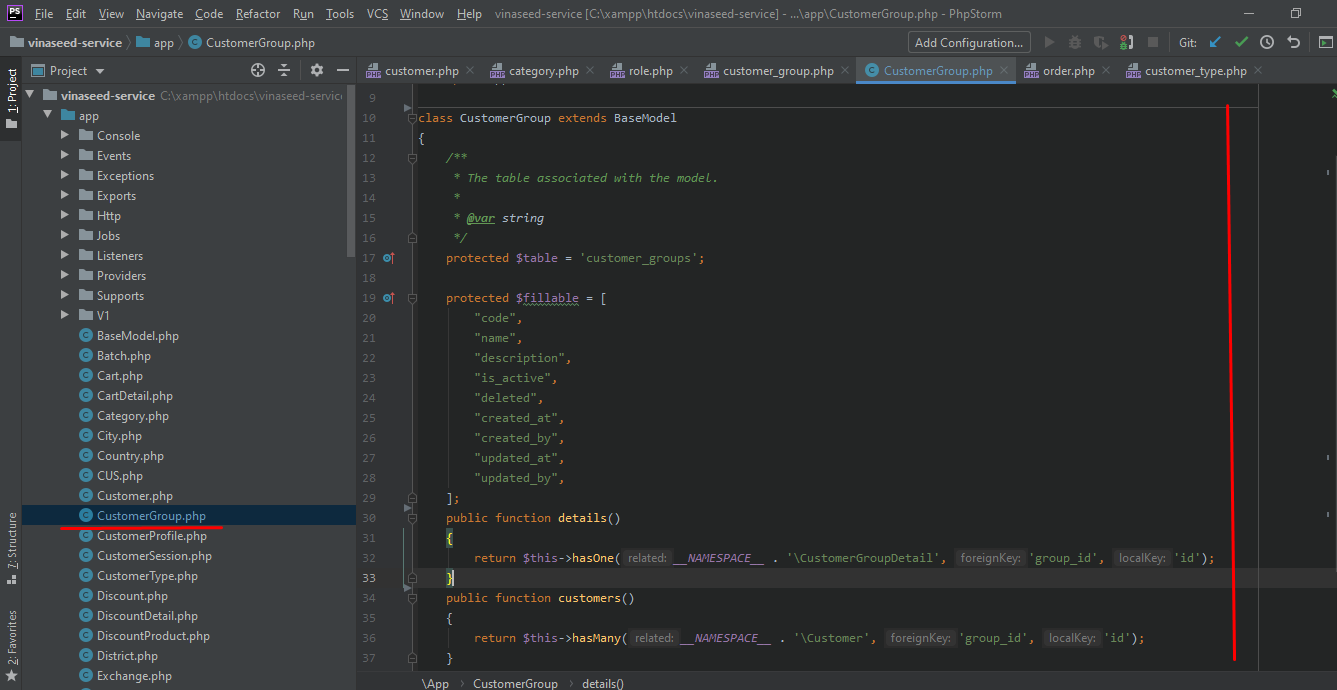
**READ:** Viết một API: <http://vinaseed-service/v1/customer_groups> phương thức Get để lấy tất cả cả dữ liệu trọng table customer\_groups:

* **Tạo table customer\_groups:**



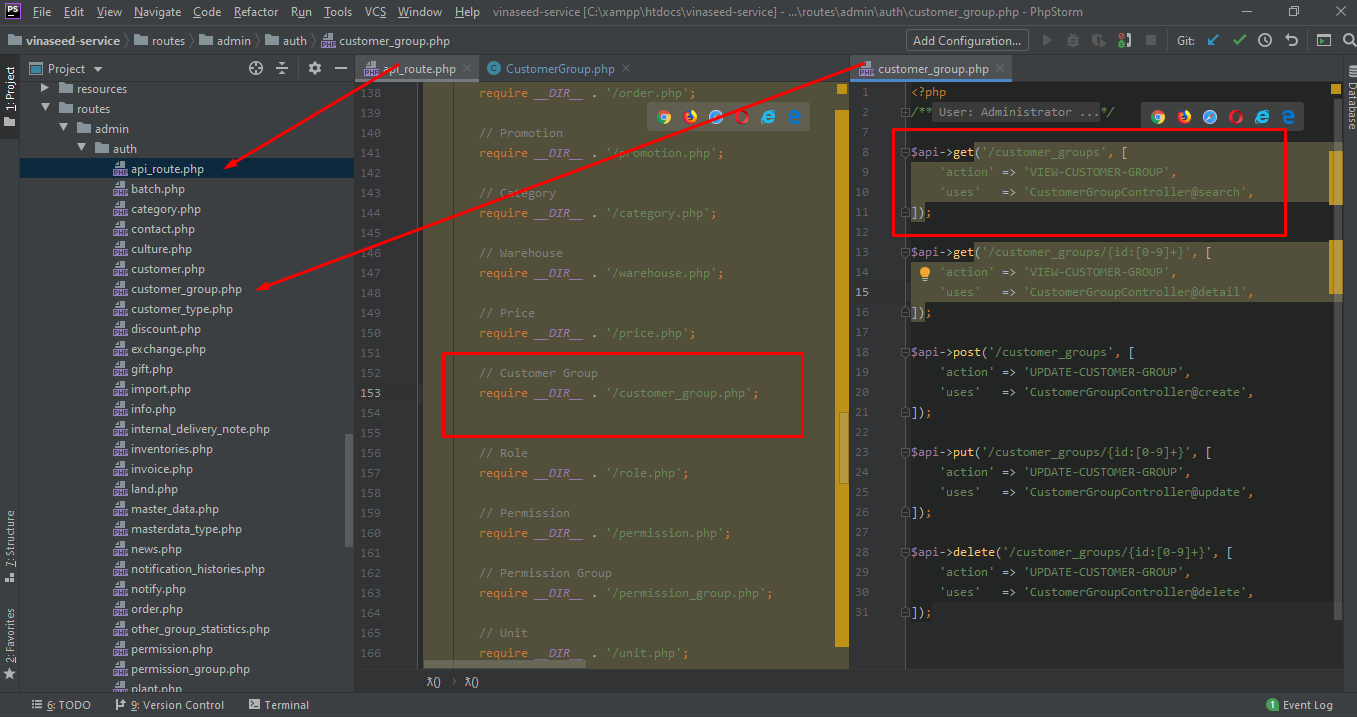
**Hình 2.2.4.3.15 Tạo table customer\_groups**

* **Tạo CustomerGroup.php trong folder app/ của project:**

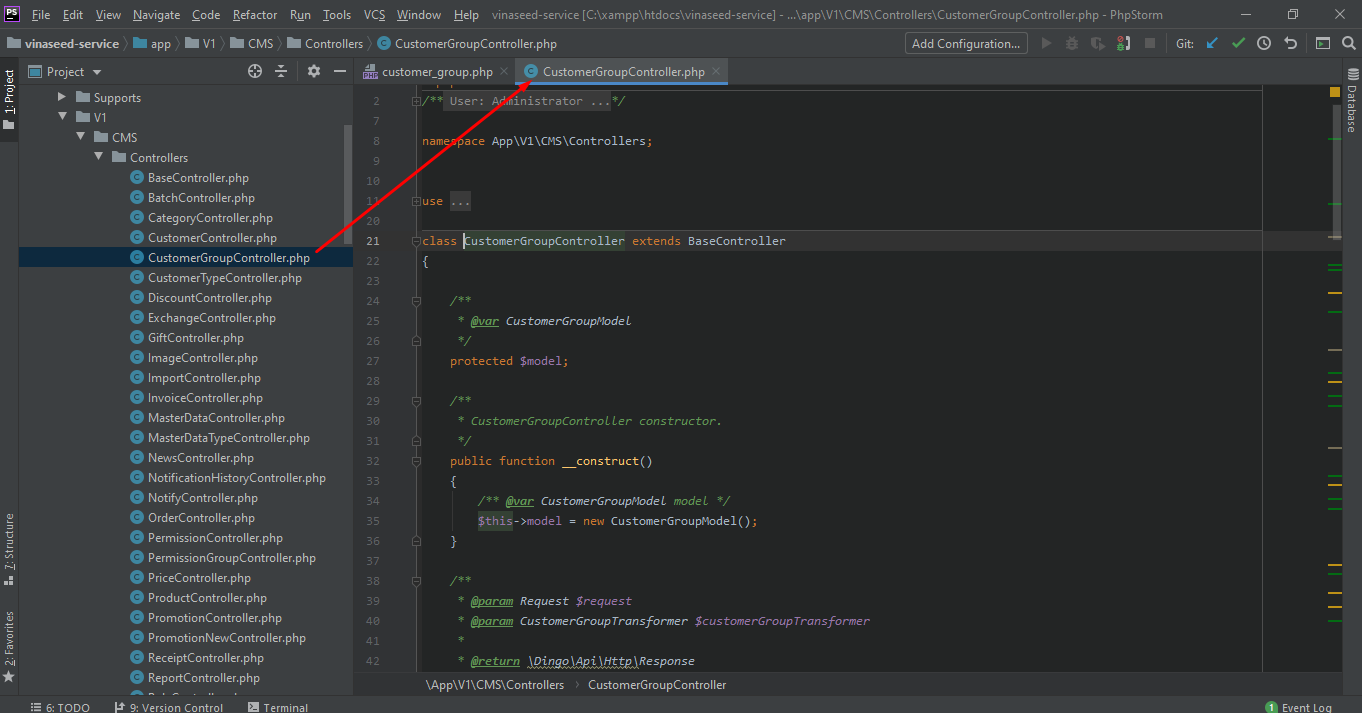


**Hình 2.2.4.3.16 Tạo app/customerGroup.php**

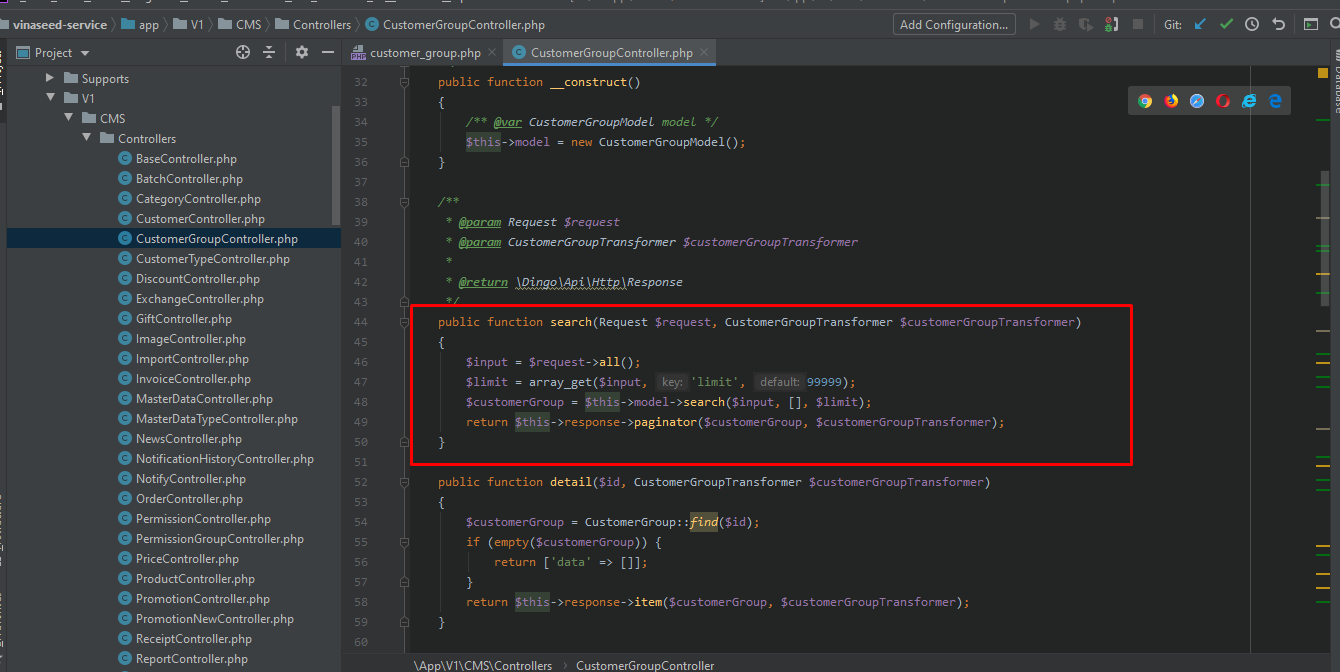
* **Tạo routes cho API**  <http://vinaseed-service/v1/customer_groups> **Phương thức GET**



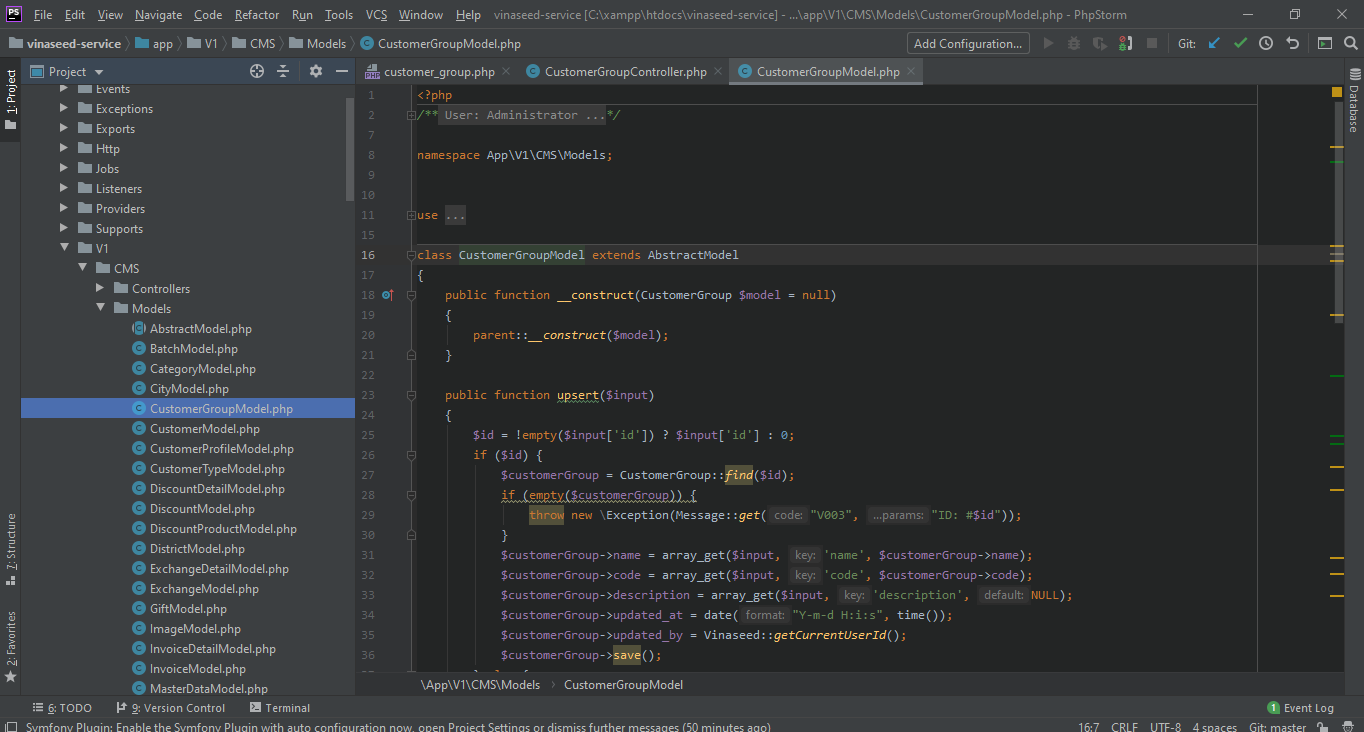
**Hình 2.2.4.3.17 Tạo route cho customer\_group**



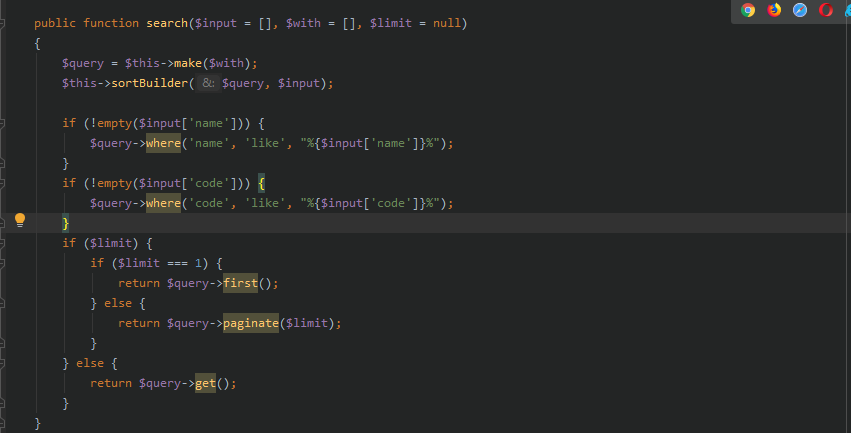
**Hình 2.2.4.3.18 Tạo controller CustomerGroupController.php**



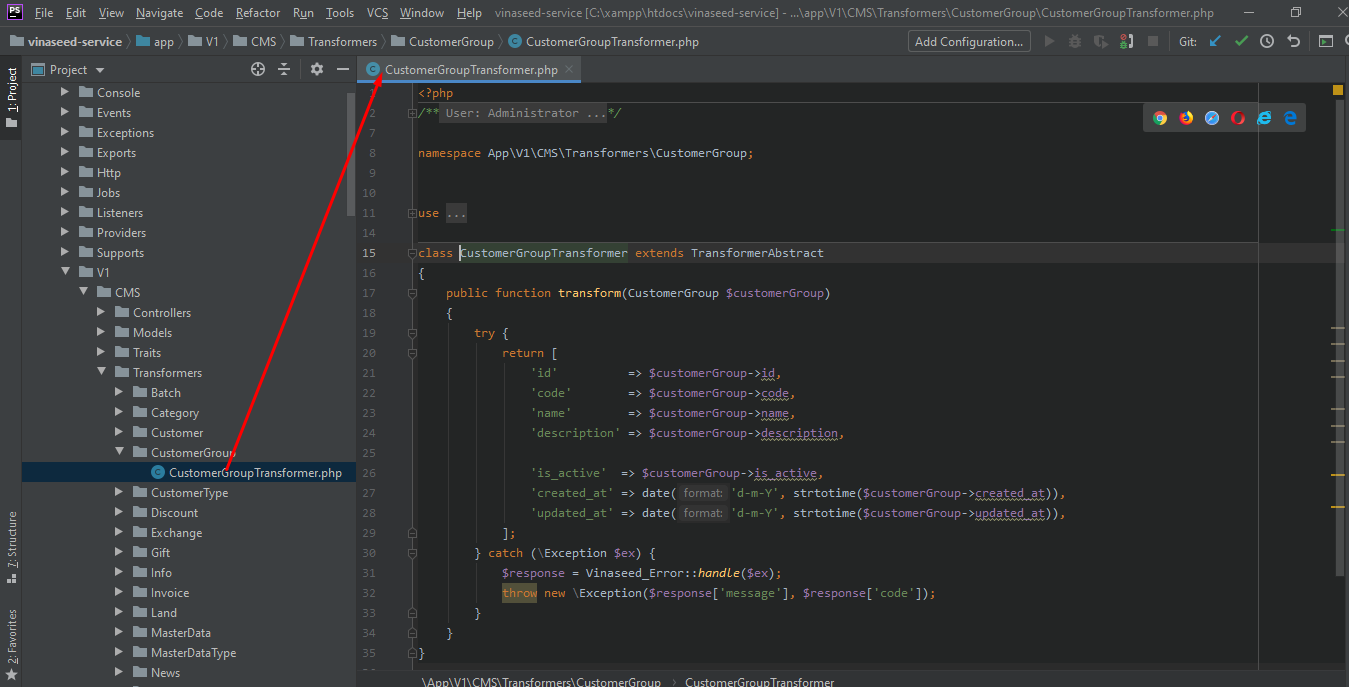
**Hình 2.2.4.3.19 Tạo function search() trong CustomerGroupController**



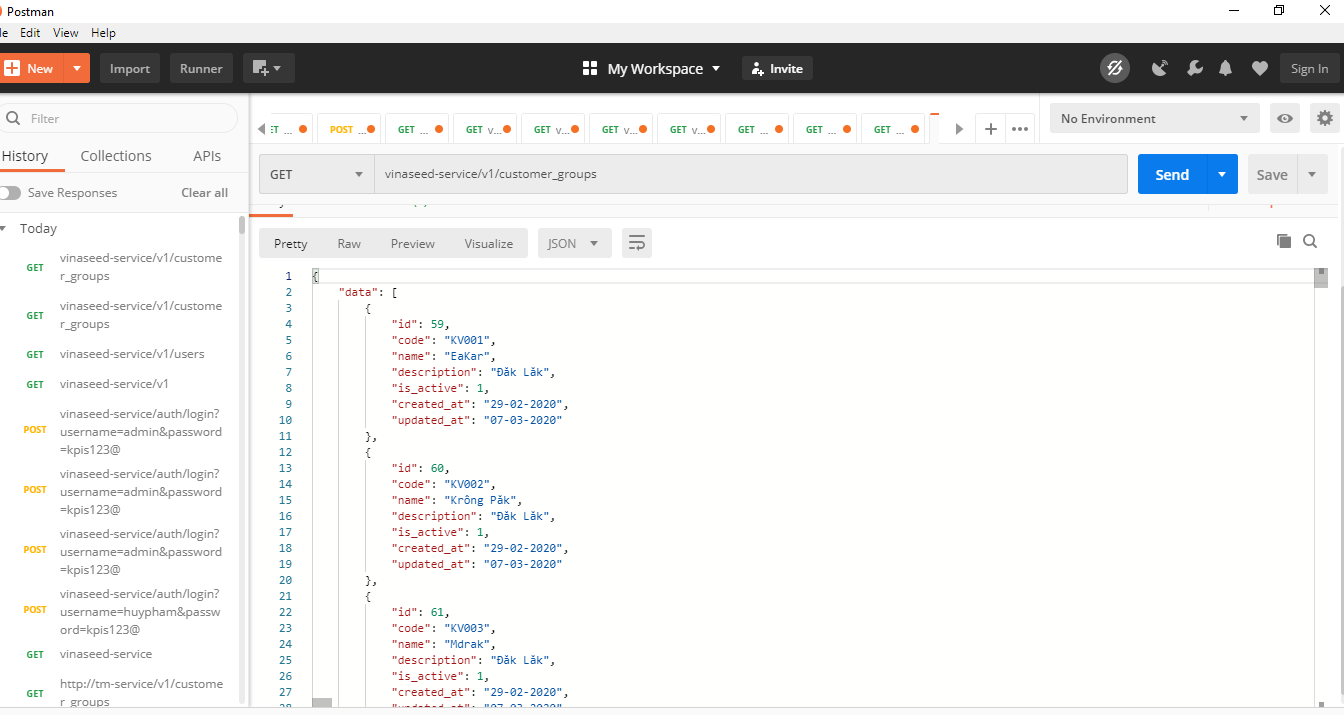
**Hình 2.2.4.3.20 Tạo Model CustomerGroupModel.php**



**Hình 2.2.4.3.21 Tạo function search trong CustomerGroupModel.php**

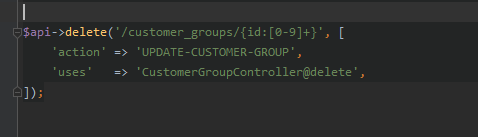


**Hình 2.2.4.3.22 Tạo Transformer CustomerGroupTransformer.php**



**Hình 2.2.4.3.23 Kết quả của API** [**http://vinaseed-service/v1/customer\_groups**](http://vinaseed-service/v1/customer_groups) **phưng thức get**

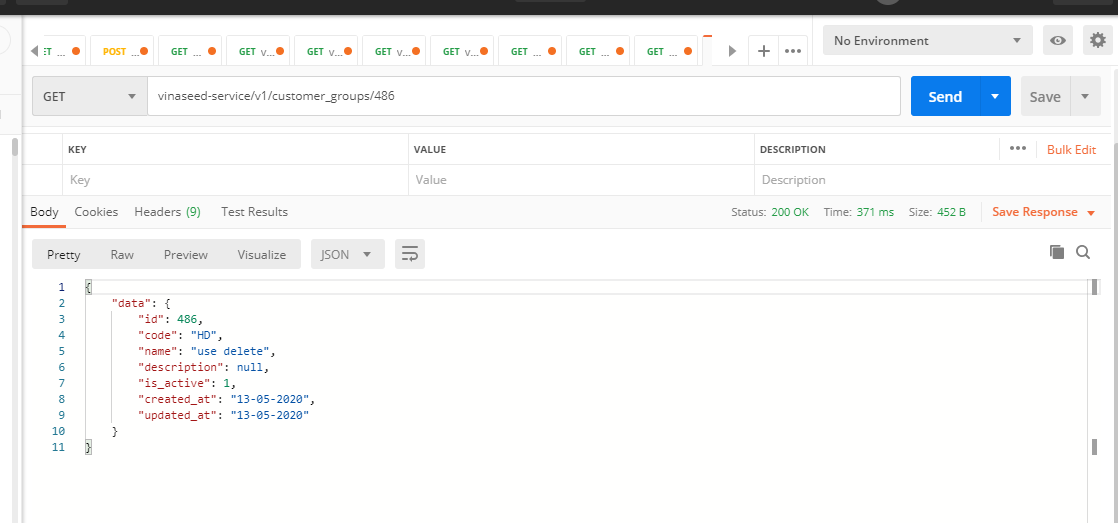
**DELETE:** Viết một API: <http://vinaseed-service/v1/customer_groups>/id phương thức delete để xóa dữ liệu theo id.



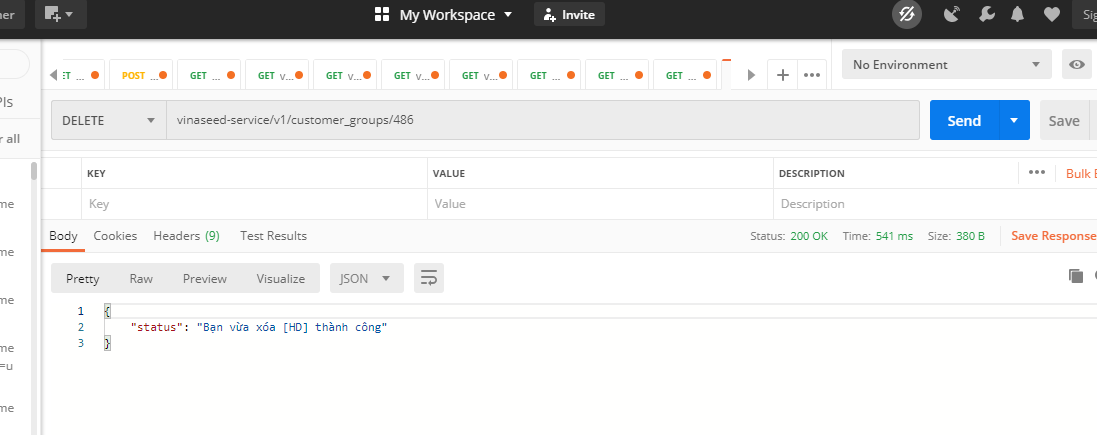
**Hình : Thên route phương thức delete**



**Hình : Tạo function delete cho CustomerGroupController.php**

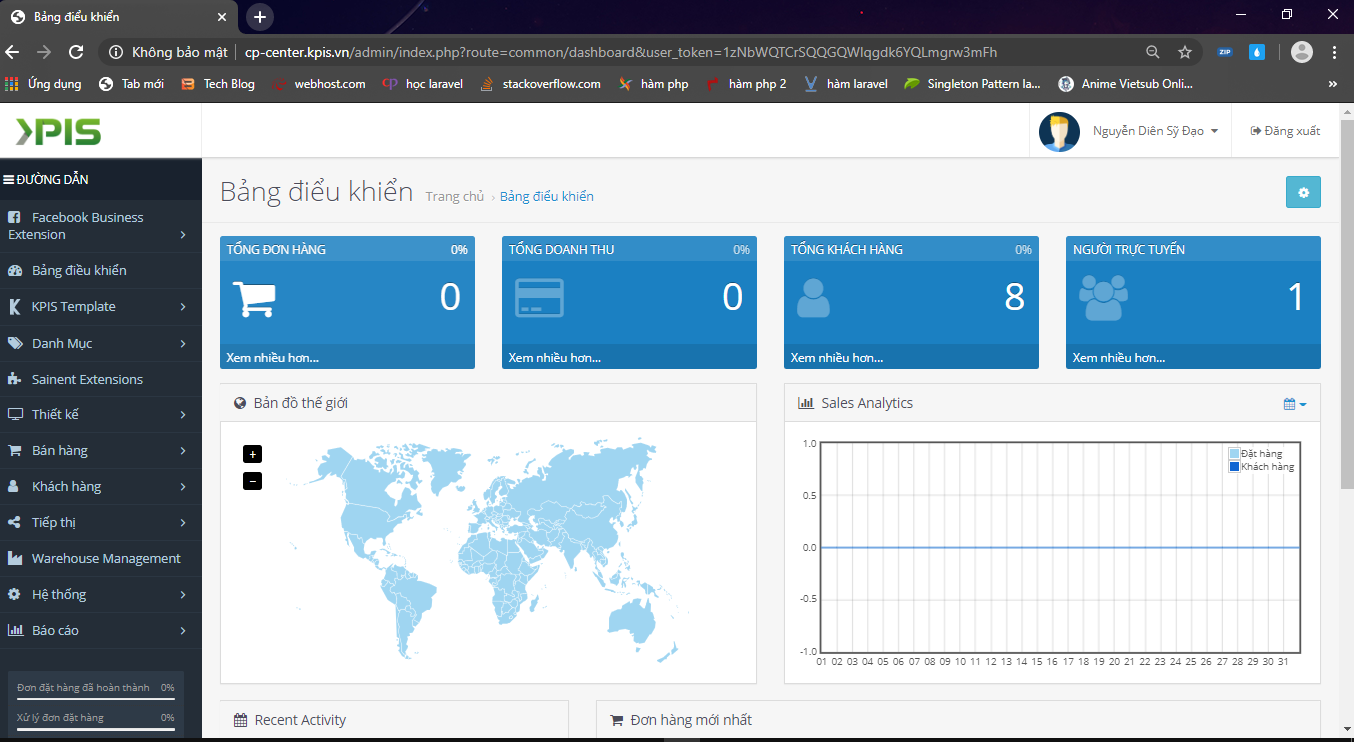


**Hình : dữ liệu để test delete**



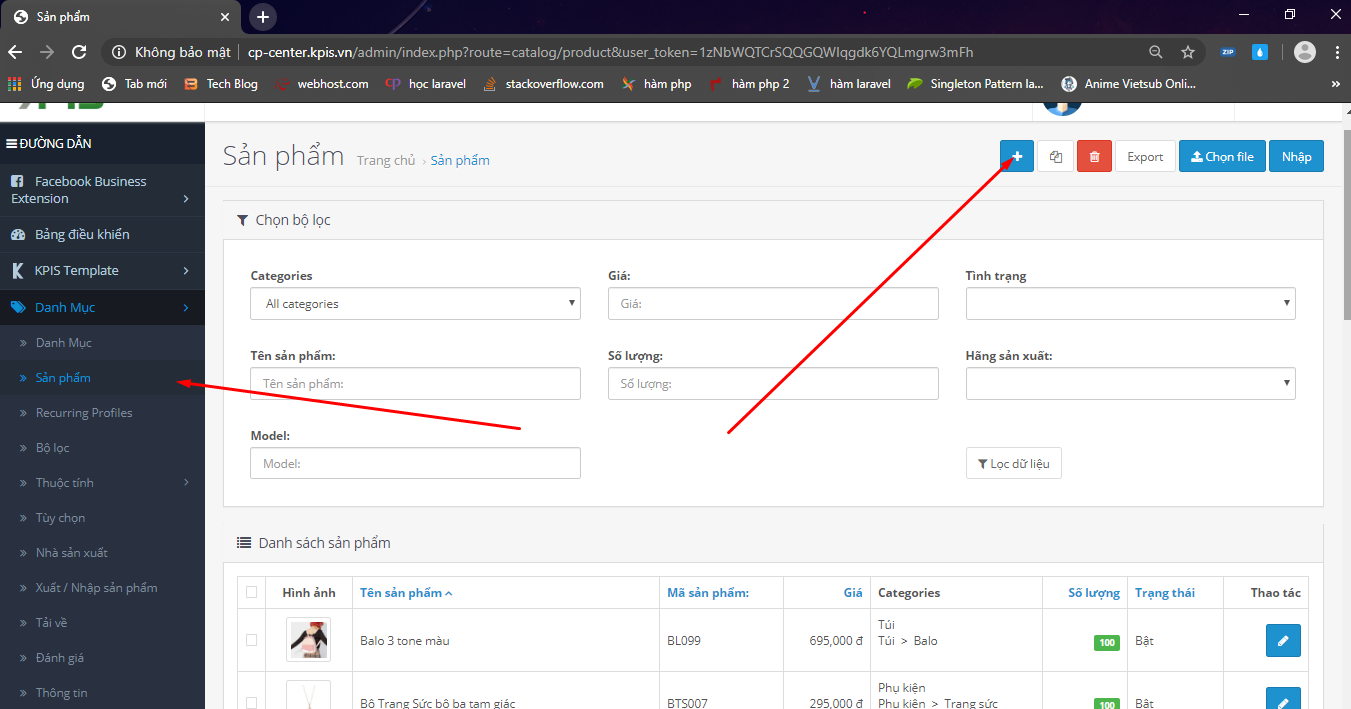
**Hình : Dùng phương thức delete xóa thành công dữ liệu có id là 486**

* + 1. **Tham gia thực tế ( dự án update giao diện đơn giản).**
    2. **Nhập liệu sản phẩm.**
       1. **Trang admin để nhập liệu:**



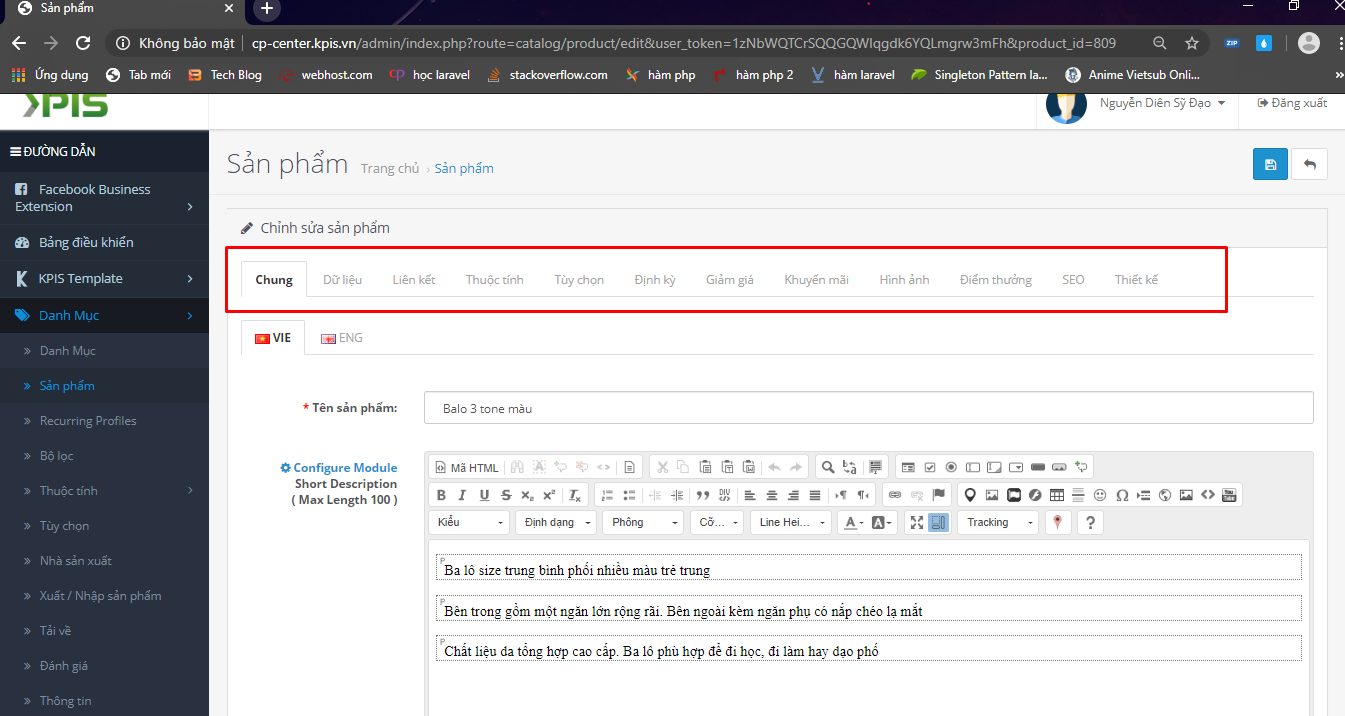
**Hình 2.2.6.1: Trang admin**

* + - 1. **Tạo sản phẩm:**



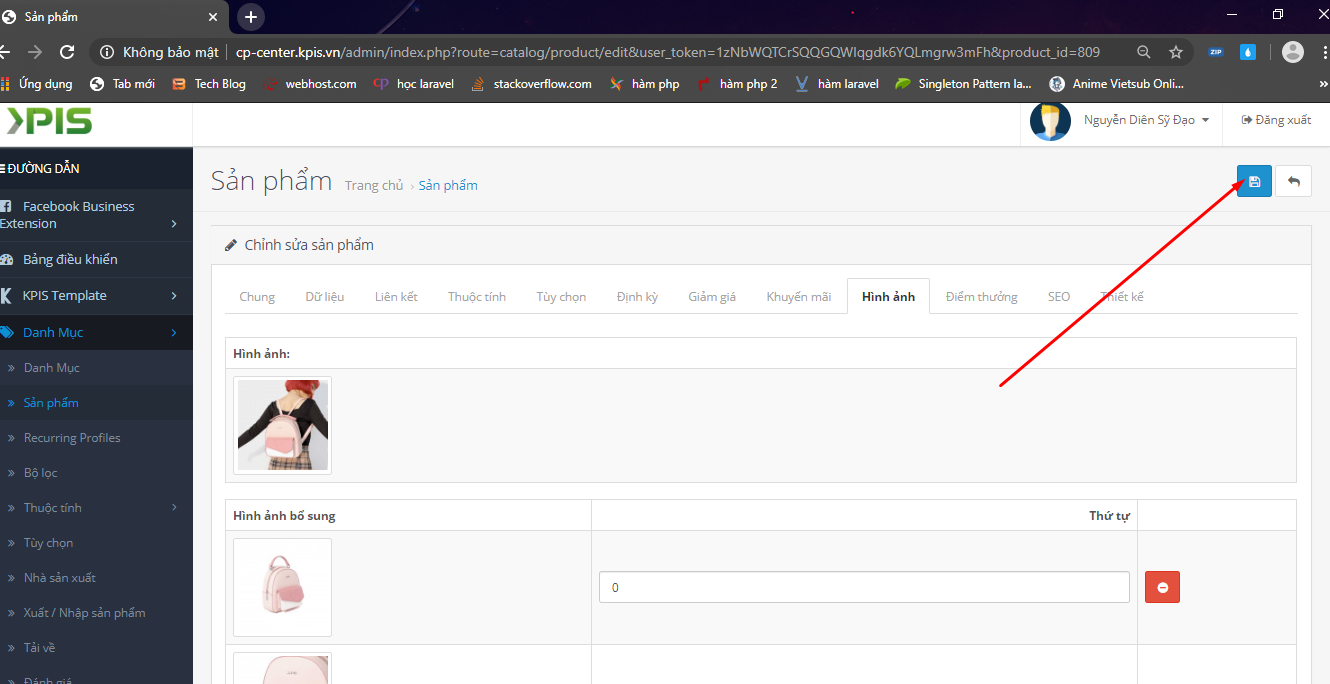
**Hình 2.2.6.2: Tạo sản phẩm**

* + - 1. **Nhập thông tin cho sản phẩm:**



**Hình 2.2.6.3: Nhập thông tin sản phẩm**

* + - 1. **Nhậm xong thì lưu lại:**



**Hình 2.2.6.4: Hoàn tất tạo một sản phẩm**

* 1. **Công tác bảo trì sản phẩm . . .**

**Tự đánh giá quá trình thực tập:**

* **Sinh viên 2: Phạm Công Huy**

Qua thời gian tham gia thực tập tại KPIS, em đã tiếp thu thêm được nhiều kiến thức, cũng như những kỹ năng làm việc thực tế, để lại trong em rất nhiều điều giá trị quý báu cho hành trang sau này. Không những trong công việc mà trong quá trình giao tiếp, trao đổi với các anh trong công ty cũng góp phần giúp đỡ em hoàn thành vấn đề thực tập. Cũng từ công việc này đã giúp em nhận ra được đâu là điểm mạnh đâu là điểm yếu của bản thân để có thể bổ sung thêm hoàn thiện bản thân tốt hơn.

Áp dụng các kiến thức cơ bản đã học ở trường trong quá trình thực tập Hiểu và học kinh nghiệm làm việc nhóm để nâng cao hiệu quả trong công việc. Đáp ứng và thực hiện đúng yêu cầu công việc được giao. Cũng cố cơ sở lý thuyết chuyên môn cũng như thực tế sau khi thực tập tại đơn vị. Tạo sự nhanh nhẹn, chủ động hơn trong công việc đối với bản thân em.

Để có kết quả làm việc tốt nhất thì chắc chắn phải có mục tiêu làm việc rõ ràng, từ đó bản thân nên xây dựng một kế hoạch, một lịch trình cụ thể, xem xét điểm mạnh yếu của bản thân, tích cực học hỏi, nâng cáo kiến thức và kĩ năng.