**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

**--------------------------------------------**



**BÁO CÁO TIẾN ĐỘ MÔN HỌC**

**MÔN: ĐỒ ÁN III – IT 3914**

**Đề tài: Xây dựng Website bán sách online**

Sinh viên**: Bùi Thị Mến**

MSSV **: 20166461**

Giảng viên hướng dẫn: **Lê Thị Hoa**

***Hà Nội, tháng 3 năm 2019***

***Mục lục***

[**I.** **PHP căn bản** 2](#_Toc2787423)

[1. Cài đặt môi trường phát triển và công cụ lập trình PHP 2](#_Toc2787424)

[2. Lập trình PHP căn bản 3](#_Toc2787425)

[3. Xử lý form 6](#_Toc2787426)

[**II.** **LARAVEL FrameWork** 6](#_Toc2787427)

[1. Cài đặt công cụ (Sublime Text, Xamp, Composer, Laravel) 6](#_Toc2787428)

[2. Tìm hiểu về Route 6](#_Toc2787429)

[3. Cơ bản về Controller 6](#_Toc2787430)

[4. Tìm hiểu về View 6](#_Toc2787431)

[5. Xuất dữ liệu dạng Json, hiển thị dữ liệu trên giao diện 6](#_Toc2787432)

[6. Quản lý cơ sở dữ liệu với Migrate, sử dụng Migrate để rollback dữ liệu 6](#_Toc2787433)

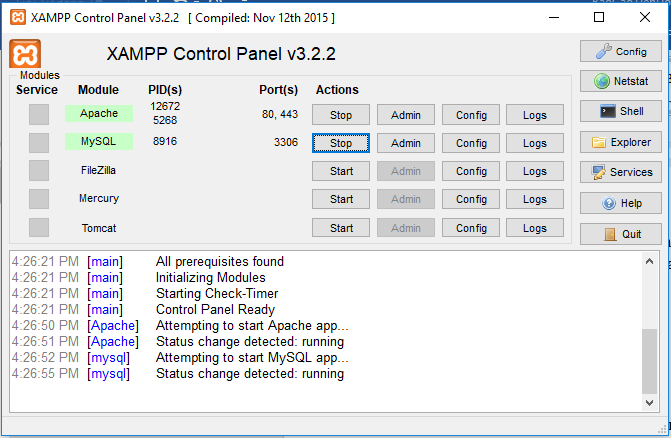
[7. Tạo dữ liệu mẫu với Seed 6](#_Toc2787434)

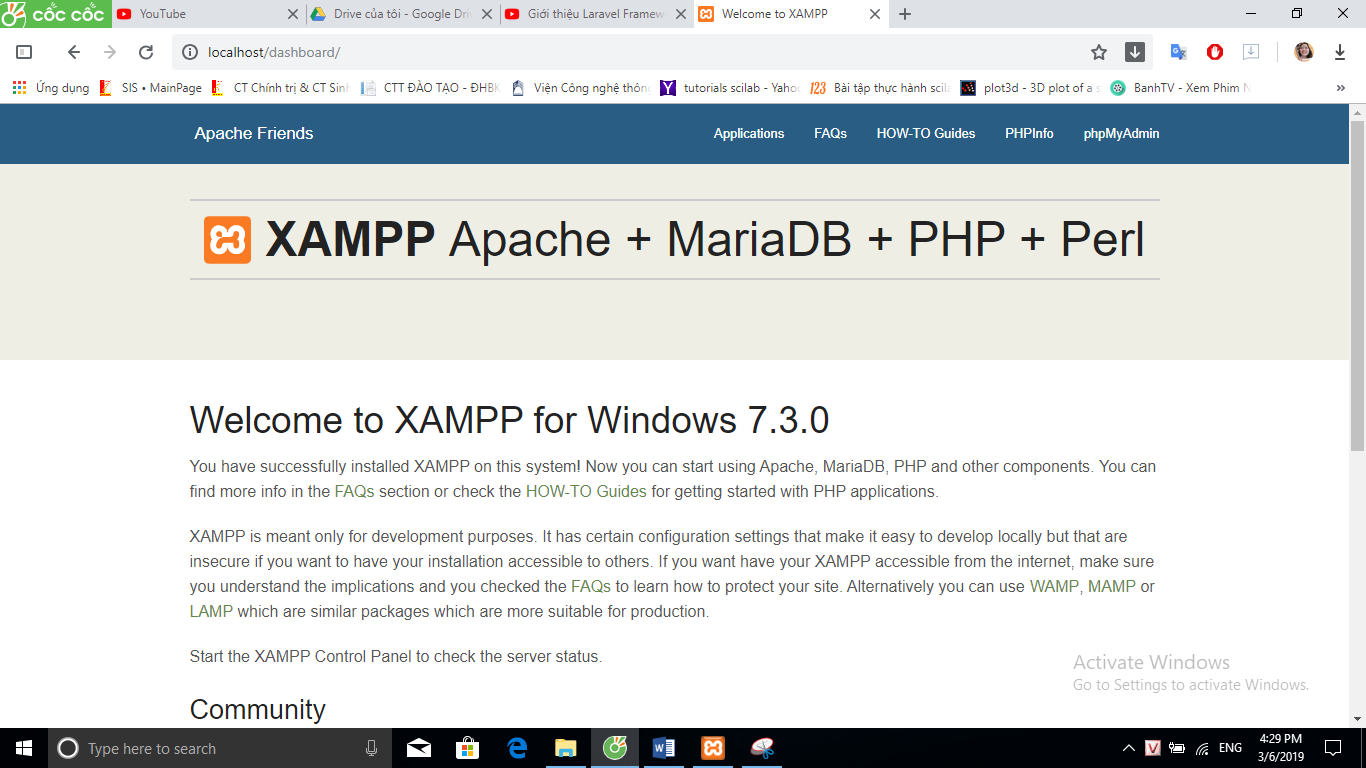
[8. Truy vẫn dữ liệu với QueryBuilder 6](#_Toc2787435)

# **PHP căn bản**

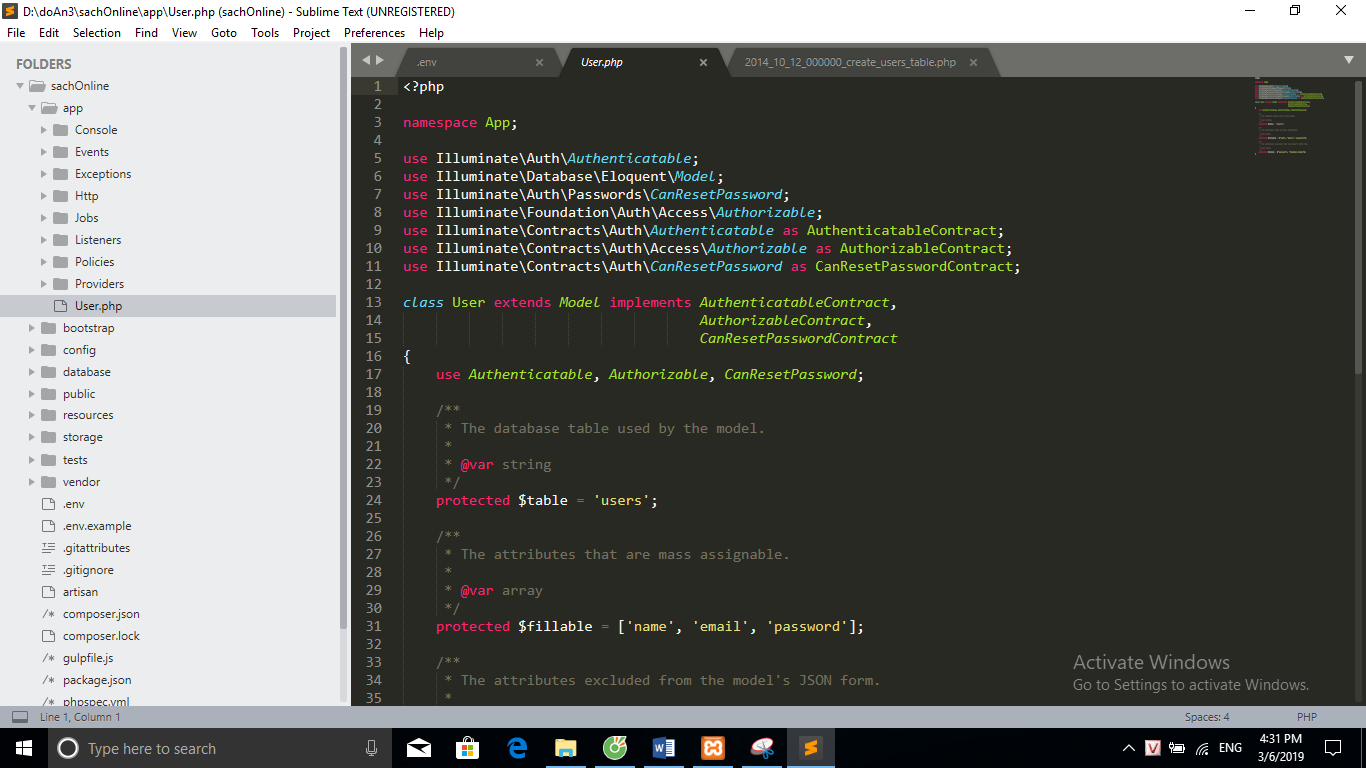
## **Cài đặt môi trường phát triển và công cụ lập trình PHP**

Đã tải và cài đặt thành công Xamp





Dùng công cụ lập trình Sublime Text



## **Lập trình PHP căn bản**

1. Biến

* Khai báo biến
* Quy định khi đặt tên biến trong PHP
* Xuất một biến

1. Kiểu dữ liệu

* Số nguyên – interger
* Số thực – float (hoặc double)
* Chuỗi – string
* Logic – boolean
* Mảng – array
* Đối tượng – Object
* Null
* Resource – thực chất là tham chiếu đến hàm hoặc tài nguyên bên ngoài PHP

1. Hằng số

* Có giá trị không thay đổi, bắt đầu bằng ký tự hoặc dấu gạch dưới không có $ phía trước tên hằng số
* Khai báo hằng số: definde(<Tên hằng số>, <Giá trị>, <Không phân biệt hoa thường>

<Tên hằng số>: Được đặt theo nguyên tắc ở trên

<Giá trị>: Giá trị của hằng số giống gán giá trị cho một biến

<Không phân biệt hoa thường>: Mặc định là fail, true hằng số sẽ không phân biệt hoa và thường

1. Toán tử số học

* Cộng: +
* Trừ: -
* Nhân: \*
* Chia: /
* Chia lấy phần dư: %

1. Toán tử gán

* Toán tử so sánh

== : bằng nhau

=== : đồng nhất (cùng giá trị và cùng kiểu dữ liệu)

!= : không bằng nhau

<> : không bằng nhau

!== : không đồng nhất (khác giá trị hoặc khác kiểu dữ liệu)

> : lớn hơn

< : nhỏ hơn

>= : lớn hơn hoặc bằng

<= : nhỏ hơn hoặc bằng

* Toán tử tăng giảm:

++$x : tăng giá trị của biến $x 1 đơn vị trước khi tính biểu thức chứa $x.

$x++ : tính biểu thức chứa $x trước sau đó tăng giá trị của biến $x 1 đơn

vị.

--$x : giảm giá trị của biến $x 1 đơn vị trước khi tính biểu thức chứa $x.

$x-- : tính biểu thức chứa $x trước sau đó giảm giá trị của biến $x 1 đơn vị.

* Phép gán tương đương:

++$x ⬄ $x = $x + 1

$x++ ⬄ $x = $x + 1

--$x ⬄ $x = $x – 1

$x-- ⬄ $x = $x – 1

1. Toán tử Logic

* and : và, $x and $y, trả về true nếu $x và $y là true.
* or : hoặc, $x or $y, trả về true nếu $x hoặc $y là true.
* xor : hoặc, $x xor $y, trả về true nếu $x hoặc $y là true và cả hai không đồng thời là true.
* && : và, tương tự and.
* || : hoặc, tương tự or
* ! : không, !$x, trả về true nếu $x là false

1. Toán tử nỗi chuỗi

* **.** : $a . $b, nối hai chuỗi $a và $bvới nhau
* **.**= : $a .= $b  $a = $a . $b

1. Câu lệnh điều kiện

* Câu lệnh if
* Câu lệnh Switch case

1. Câu lệnh lặp

* Lệnh lặp for
* Vòng lặp While
* Vòng lặp do … while
* Vòng lặp foreach
* Lệnh break và continue

1. Hàm trong PHP

* Hàm là một đoạn chương trình nhỏ được sử dụng trong chương trình lớn. Mỗi hàm có một chức năng riêng nhằm xử lý một phần nhỏ trong bài toán lớn. PHP hỗ trợ sẵn hơn 1000 hàm với nhiều nhóm chức năng như mảng, chuỗi, v.v...

1. Hàm do người dùng tự định nghĩa

* Tạo hàm
* Tham số
* Tham số có giá trị mặc định
* Hàm trả về giá trị

1. Mảng

* Khái niệm
* Mảng trong PHP
* Tạo mảng
* Mảng chỉ số
* Kích thước của mảng chỉ số và duyệt mảng
* Mảng kết hợp
* Duyệt mảng kết hợp bằng foreach
* Mảng đa chiều
* Duyệt mảng đa chiều bằng câu lệnh for

1. Biến toàn cục – Superglobals

* $\_GLOBALS
* $\_SERVER
* $\_REQUEST
* $\_POST
* $\_GET
* $\_FILES
* $\_ENV
* $\_COOKIE
* $\_SESSION

1. Biến $\_GLOBALS

* Biến $GLOBALS là một biến toàn cục chứa tất cả những biến khác được khai báo trong chương trình. Nó cung cấp cho lập trình viên một cách thức truy cập đến tất cả các biến trong chương trình một cách toàn cục.

1. Biến $\_SERVER

* Biến $\_SERVER trong PHP cung cấp những thông tin liên quan đến server, host, đường dẫn, v.v...

1. Biến $\_GET

* Biến $\_GET dùng để lấy dữ liệu do người dùng nhập được truyền từ form và dữ liệu được truyền trực tiếp thông qua URL.

1. Biến $\_POST

* Biến $\_POST tương tự như biến $\_GET cũng được sử dụng trong form. Tuy nhiên form phải có phương thức method="post". Và $\_POST không lấy được dữ liệu truyền vào URL.

## **Xử lý form**

1. Xử lý form

* Xây dựng form bằng HTML
* Tách phần HTML và PHP xử lý
* GET và POST

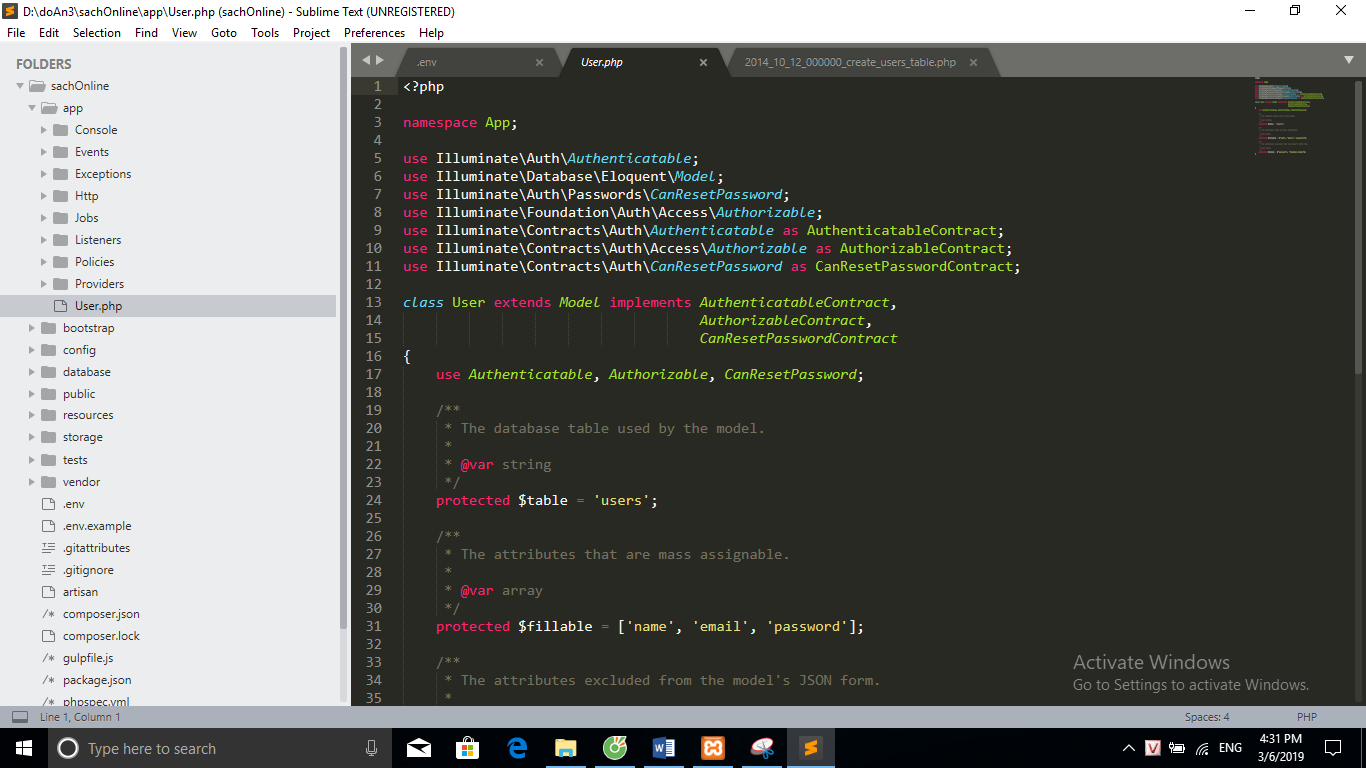
1. Xác thực dữ liệu trong form

* Tại sao phải xác thực dữ liệu trong form
* Xác thực dữ liệu
* Bỏ các ký tự không cần thiết

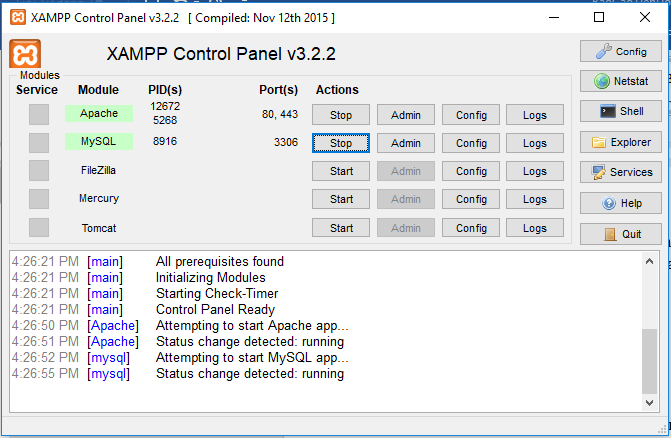
# **LARAVEL FrameWork**

## **Cài đặt công cụ (Sublime Text, Xamp, Composer, Laravel)**

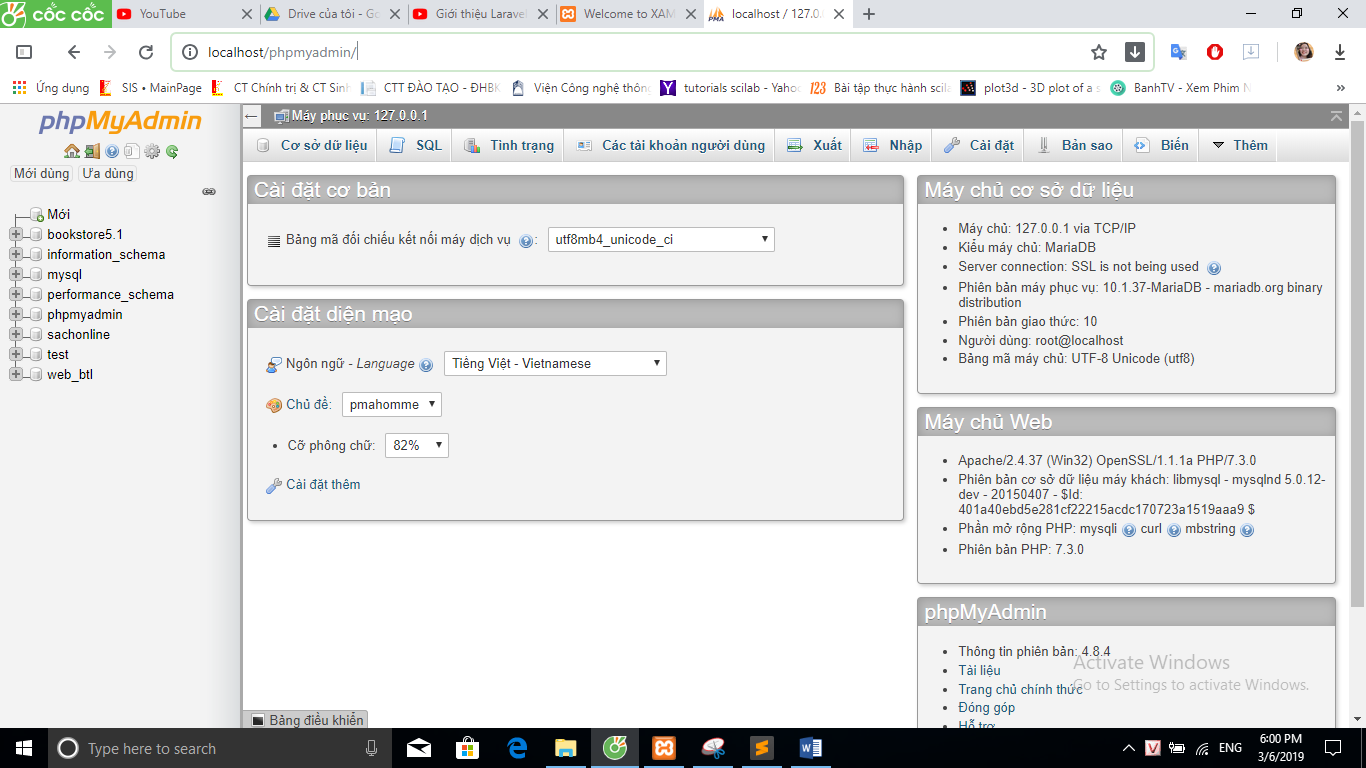
* Sublime Text:

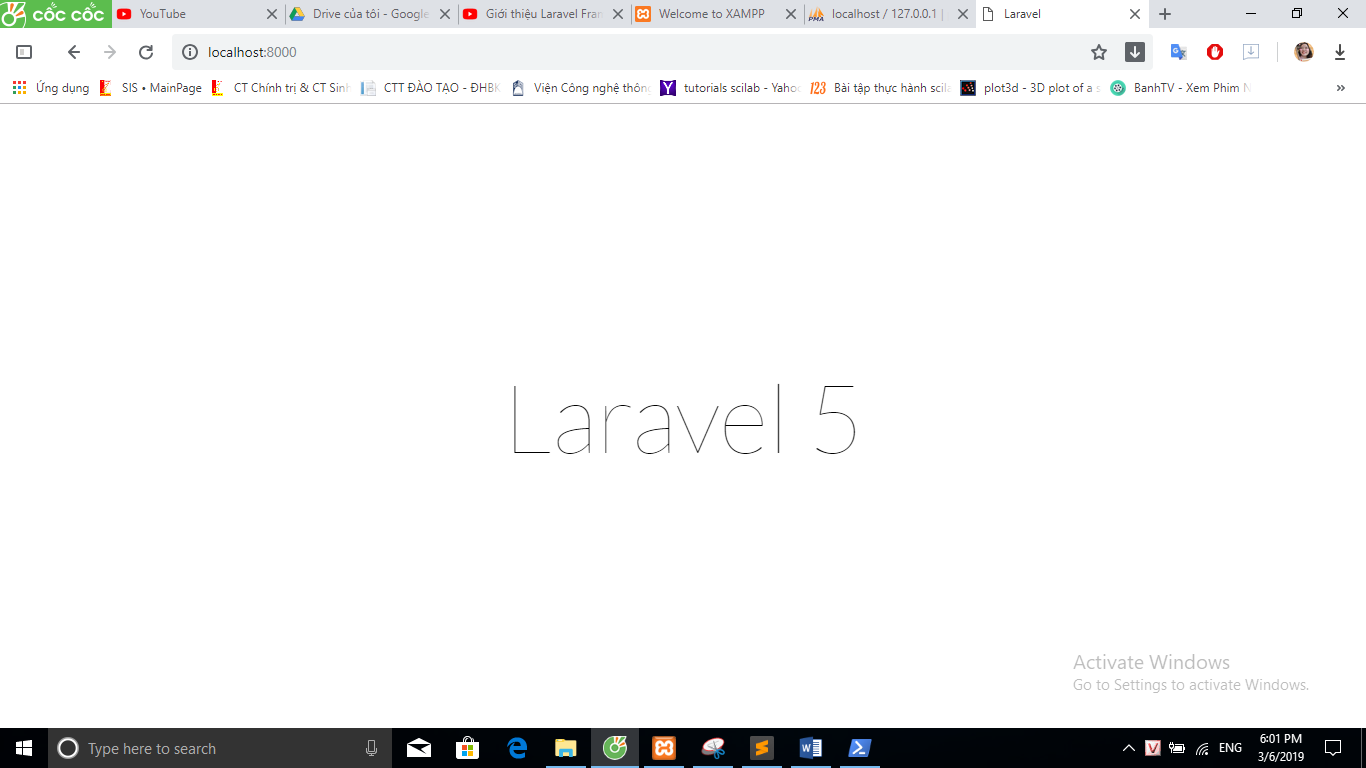


* Xamp:



* Laravel:





## **Tìm hiểu về Route**

* Các phương thức trong Route

Route::get($url, $callback);

Route::post($url, $callback);

Route::put($url, $callback);

Route::patch($url, $callback);

Route::delete($url, $callback);

Route::options($url, $callback);

Trong đó : $url là đường dẫn Route

$callback Là một hàm gọi đến sẽ được thực hiện

* Mọi Request từ phía người dùng đều phải qua Route

## **Cơ bản về Controller**

* Tạo Controller bằng lệnh:

Php artisan make:controller controllername --resource

Controllername: Tên Controller

--resource: Để laravel sẽ tự sinh một file controller có sẵn một số funtion()

* Controller sẽ lấy dữ liệu người dùng gửi dùng qua Route để xử lý, cần dữ liệu sẽ lấy từ Model lên hoặc cập nhật xuống Model

## **Tìm hiểu về View**

* View có nhiệm vụ lấy dữ liệu từ Controller và sau đó đưa vào layout của giao diện nó sẽ hiển thị ra cho người dùng

## **Xuất dữ liệu dạng Json, hiển thị dữ liệu trên giao diện**

* Xuất dữ liệu dạng Json
* Hiển thị dữ liệu trên giao diện

## **Quản lý cơ sở dữ liệu với Migrate, sử dụng Migrate để rollback dữ liệu**

* Quản lý cơ sở dữ liệu với Migrate
* Sử dụng Migrate để Rollback dữ liệu: Lệnh này nhằm rollback toàn bộ CSDL đồng thời chạy lại luôn toàn bộ các file migrate

Php artisan migrate:rollback

## **Tạo dữ liệu mẫu với Seeder**

* Sau khi tạo CSDL, cần một số lượng dữ liệu khá lớn để hiển thị lên Website của chúng ta, thay vì nhập thủ công vừa tốn thời gian vừa tốn công thì Laravel hỗ trợ Seeder để nhanh chóng tạo dữ liệu, nhanh chóng phát triển các tính năng
* Tạo dữ liệu mẫu bằng lệnh trong Laravel:

Php artisan make:seeder news

* Đẩy dữ liệu mẫu lên mySQL:

Php artisan db:seed

## **Truy vẫn dữ liệu với QueryBuilder**

* Laravel QueryBuilder có thể xây dựng bất kỳ câu truy vấn nào dù nó phức tạp đến đâu
* Laravel Query Builder sử dụng PDO (PHP Data Object, hệ thống API có sẵn của PHP để kết nối đến các CSDL thông dụng), bản thân API PDO đã bảo vệ bạn trước các tấn công SQL Injection, do đó khi xử dụng Query Builder bạn không cần lo lắng xử lý dữ liệu trước khi chèn vào database.
* Truy xuất dữ liệu từ database

*Lấy toàn bộ các bản ghi từ một bảng*

*Lấy một bản ghi hoặc một cột trong bảng*

*Lấy nhiều cột trong bảng*

*Lấy giá trị tổng hợp từ kết quả truy vấn*

* Các phương thức trong QueryBuilder

*Select*

*Raw expression*

*Mệnh đề Where*

*Thao tác với kết quả trả về*

*Insert một hoặc nhiều bản ghi vào CSDL*

*Update dữ liệu trong CSDL*

*Delete dữ liệu trong CSDL*

## **Tìm hiểu về Model**

* Model đóng vai trò giao tiếp với database
* Tạo Model bằng lệnh Laravel cung cấp sẵn

Php artisan make:model News

News: Là tên model tạo mới

* Truy vẫn dữ liệu với Model