**Bài 22: Authentication Trong Laravel**

Authentication  là một hành động xác  thực một cái gì đó có được hoạt động tiếp hay không hay dừng lại.Và hôm nay chúng ta sẽ cùng tìm hiểu về authentication trong Laravel nhé!

## 1, Authentication là gì?

-Authentication theo Wikipedia:

là một hành động nhằm thiết lập hoặc chứng thực một cái gì đó (hoặc một người nào đó) đáng tin cậy, có nghĩa là, những lời khai báo do người đó đưa ra hoặc về vật đó là sự thật. Xác thực một đối tượng còn có nghĩa là công nhận nguồn gốc (provenance) của đối tượng, trong khi, xác thực một người thường bao gồm việc thẩm tra nhận dạng của họ. Việc xác thực thường phụ thuộc vào một hoặc nhiều nhân tố xác thực (authentication factors) để minh chứng cụ thể.

-Theo như wikipedia định nghĩa thì các bạn cũng đã hiểu sơ qua về nó rồi đúng không? Thì ở trong lập trình khái niện nó cũng như thế, và ứng dụng thực tiễn là chức năng đăng nhập vào hệ thống.

## 2, Authentication trong Laravel.

-Ở phần đầu chúng ta đã hiểu về khái niệm rồi và phần này chúng ta sẽ xem xem Authentication trong Laravel hoạt động như thế nào nhé!

-Đầu tiên mọi người hãy mở thư mục config trong project Laravel và mở tệp tin **auth.php** ra và xem:

<?php

return [

/\*

|--------------------------------------------------------------------------

| Authentication Defaults

|--------------------------------------------------------------------------

|

| This option controls the default authentication "guard" and password

| reset options for your application. You may change these defaults

| as required, but they're a perfect start for most applications.

|

\*/

'defaults' => [

'guard' => 'web',

'passwords' => 'users',

],

/\*

|--------------------------------------------------------------------------

| Authentication Guards

|--------------------------------------------------------------------------

|

| Next, you may define every authentication guard for your application.

| Of course, a great default configuration has been defined for you

| here which uses session storage and the Eloquent user provider.

|

| All authentication drivers have a user provider. This defines how the

| users are actually retrieved out of your database or other storage

| mechanisms used by this application to persist your user's data.

|

| Supported: "session", "token"

|

\*/

'guards' => [

'web' => [

'driver' => 'session',

'provider' => 'users',

],

'api' => [

'driver' => 'token',

'provider' => 'users',

],

],

/\*

|--------------------------------------------------------------------------

| User Providers

|--------------------------------------------------------------------------

|

| All authentication drivers have a user provider. This defines how the

| users are actually retrieved out of your database or other storage

| mechanisms used by this application to persist your user's data.

|

| If you have multiple user tables or models you may configure multiple

| sources which represent each model / table. These sources may then

| be assigned to any extra authentication guards you have defined.

|

| Supported: "database", "eloquent"

|

\*/

'providers' => [

'users' => [

'driver' => 'eloquent',

'model' => App\User::class,

],

/\* 'users' => [

'driver' => 'database',

'table' => 'users',

], \*/

],

/\*

|--------------------------------------------------------------------------

| Resetting Passwords

|--------------------------------------------------------------------------

|

| You may specify multiple password reset configurations if you have more

| than one user table or model in the application and you want to have

| separate password reset settings based on the specific user types.

|

| The expire time is the number of minutes that the reset token should be

| considered valid. This security feature keeps tokens short-lived so

| they have less time to be guessed. You may change this as needed.

|

\*/

'passwords' => [

'users' => [

'provider' => 'users',

'table' => 'password\_resets',

'expire' => 60,

],

],

];

-Nó sẽ có đoạn như trên (chú ý: Mình đang viết bài trên phiên bản mới nhất -5.3 Sau này khi có bản mới hơn có thể nó sẽ khác đi một chút).

-Đoạn code trên khá là dài đúng không các bạn?(toàn comment) Chức năng của đoạn code đó là xác định phương thức xác thực cũng như lưu trữ mặc định, và tùy biến.

-Vì nó khá là loàng ngoàng lên ở series basic này mình sẽ không trình  bày nhiều về cả đoạn này vì nói nhiều các bạn sẽ loạn. Các bạn chỉ cần quan tâm đoạn code này là được.

'providers' => [

'users' => [

'driver' => 'eloquent',

'model' => App\User::class,

],

-Đây là đoạn xác lập phương thức, và nguồn dữ liệu được lấy ra để xác thực.

**Chú thích**:

1. **Driver**: Đây là thiết lập xác định phương thức lấy thông tin người dùng để xác thực, và như các bạn thấy thì  mặc định nó sẽ sử dụng [***eloquent***](http://toidicode.com/Bai-12-Eloquent-Orm-Trong-Laravel-13)***,*** tuy nhiên còn một sự lựa chọn khác là database và thậm trí bạn cũng có thể tạo thêm một driver của riêng bạn( sẽ nói ở phần nâng cao). Đối với series cơ bản này thì mình khuyên các bạn nên dữ nguyên.
2. **Model**: Đây là thiết lập nguồn dữ liệu lấy ra từ đâu. Ở đây mặc định Laravel chọn là moder User (local: App\User.php). Thông số này bạn có thể chỉnh thành name model của bạn.

## 3, Tạo Auth Login Trong Laravel.

-Để tạo Auth trong Laravel thì cũng hết sức đơn giản. Các bạn chỉ cần dùng lệnh:

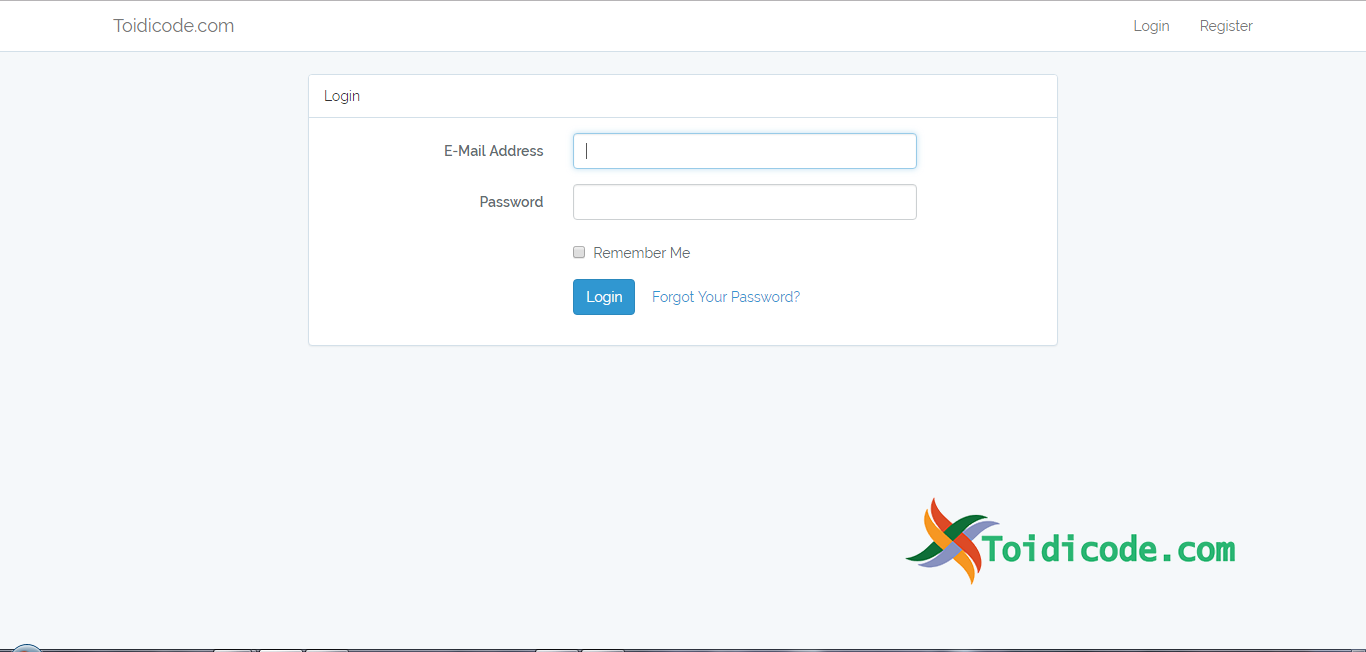
php artisan make:auth

Sau khi chạy lệnh này lên thì Laravel sẽ thêm cho chúng ta một homeController,2 route và rất nhiều view mới.

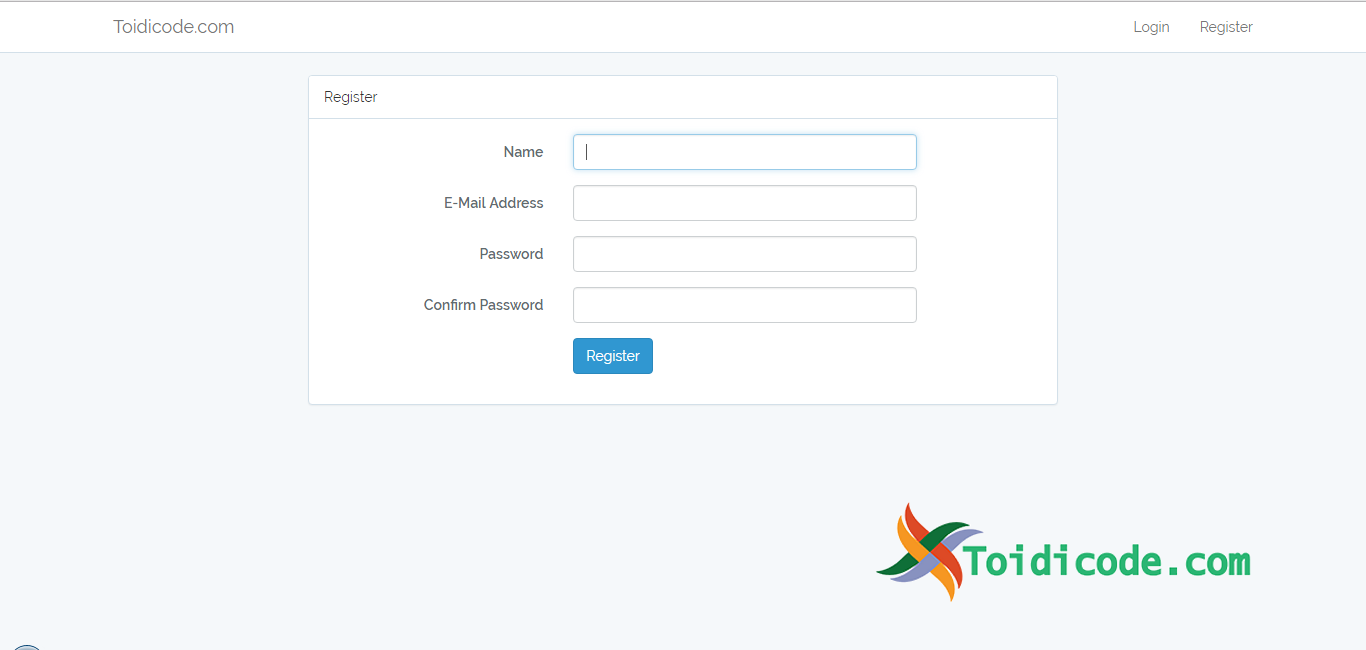
-Ngay lúc này bạn run project lên và sẽ thấy một số thay đổi về giao diện welcome mặc định của Laravel.



-Trên màn hình welcome của laravel đã xuất hiện thêm login và register, Tiếp tục click vào từng action ta thấy:



Login Form



Register form

-Sau đó bạn cứ thử đăng ký một users, đăng nhập vào hệ thống rồi vào database xem có gì ở bảng users nhé!