




LARAVEL FRAMEWORK










BÀI 6: AUTHENTICATION & MIDDLEWARE

PHẦN 1: AUTHENTICATION








- Hiểu về triển khai Authentication trong Laravel
- Hiểu và sử dụng Middleware trong Laravel

<p>Email</p> <input type="text" value="admin"/>	
<p>Password</p> <input type="password" value="....."/>	
<p><input type="checkbox"/> Remember me</p> <p>Forgot your password? LOG IN</p>	
<p>Forgot your password? No problem. Just let us know your email address and we will email you a password reset link that will allow you to choose a new one.</p> <p>Email</p> <input type="text"/>	<p>Name</p> <input type="text"/>
	<p>Email</p> <input type="text"/>
	<p>Password</p> <input type="password"/>
	<p>Confirm Password</p> <input type="password"/>
<p>EMAIL PASSWORD RESET LINK</p>	<p>Already registered? REGISTER</p>

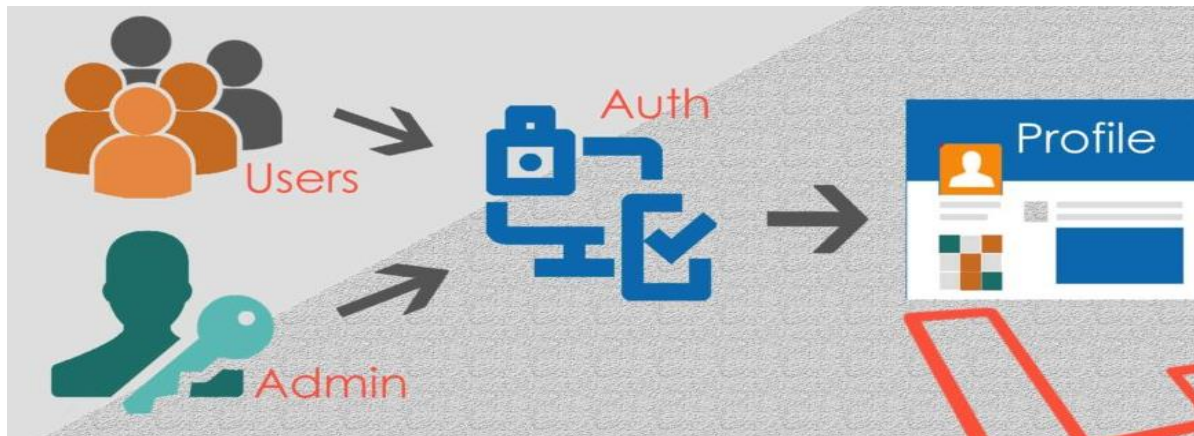
Phần I:

-  Authentication là gì
-  Cấu hình authentication
-  Các starter kit trong laravel
-  Cài Laravel breeze , xem kết quả và test hoạt động
-  Cài Laravel jetstream, xem kết quả và test hoạt động
-  Lấy thông tin của user đã đăng nhập
-  Xác định user đã login hay chưa
-  Route protect – bảo vệ chức năng
-  Chuyển hướng user chưa đăng nhập

Phần II:

-  HTTP Basic Authentication
-  Chức năng Logout
-  Laravel Authentication Events
-  Middleware trong Laravel
-  Tạo mới middleware
-  Middleware & Response
-  Đăng ký middleware với Laravel

- ❑ Authentication là hoạt động xác thực người dùng.
- ❑ Mục đích là để xác minh rõ ai đang truy cập vào, để làm căn cứ cho/cấm sử dụng các chức năng trong hệ thống.
- ❑ Authentication là hoạt động cần phải có trong hầu hết các website.
- ❑ Laravel cung cấp sẵn tiện ích giúp triển khai authentication rất nhanh chóng , dễ dàng.



- ❑ Thực hiện authentication phải tương tác với bảng **users**. Laravel tạo sẵn model **User** cho mục đích này.
- ❑ Các cấu hình cho hoạt động authentication, khai báo trong file config/auth.php. Mặc định guard web (có thể tạo thêm) được dùng để xác thực user qua model Eloquent User, thông tin lưu vào session

```
'defaults' => [
    'guard' => 'web',
    'passwords' => 'users',
],
'guards' => [
    'web' => [
        'driver' => 'session',
        'provider' => 'users',
    ],
],
'providers' => [
    'users' => [
        'driver' => 'eloquent',
        'model' => App\Models\User::class,
    ],
],
```

- ❑ Starter Kit là các gói bổ sung, miễn phí giúp triển khai authentication được nhanh chóng. Các gói này gồm:
 - ❖ **Laravel Breeze** là gói đơn giản, gồm các chức năng register, login,, password reset, email verification và password confirm.
 - ❖ **Laravel Fortify** là gói triển khai authentication cho backend. Gồm authentication bình thường và có thêm two-factor authentication.
 - ❖ **Laravel Jetstream**: có các tính năng như Fortify nhưng UI đẹp hơn đi cùng với Tailwind CSS, Livewire, Inertia.js. Jetstream có two-factor authentication, browser session management, profile management, built-in sẵn Sanctum để API token authentication

Để triển khai Breeze, thực hiện như sau:

- ❑ B1: Tạo 1 project mới và cấu hình kết nối đến database.
- ❑ B2: Tải breeze vào project với lệnh
composer require laravel/breeze --dev
- ❑ B3: Cài breeze (để tạo views, routes, controllers, resources) với các lệnh:

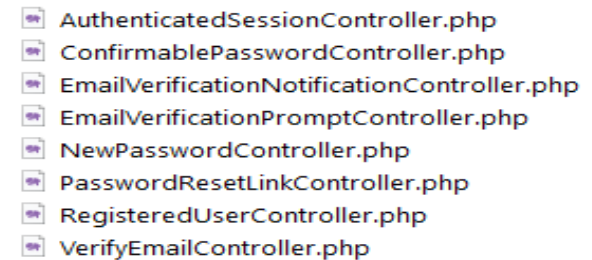
php artisan breeze:install

npm install

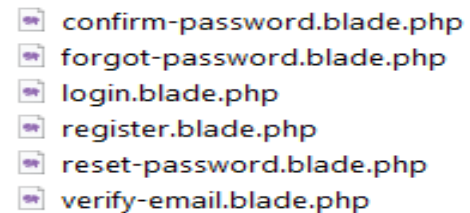
npm run dev

php artisan migrate

- ❑ Khi cài Breeze xong, vào folder **app\Http\Controllers\Auth** sẽ thấy các controller phục vụ authentication
- ❑ Trong **resources\views\auth** cũng sẽ thấy các views được tạo ra
- ❑ Mở **routes\auth.php** sẽ thấy các route được tạo ra



- AuthenticatedSessionController.php
- ConfirmablePasswordController.php
- EmailVerificationNotificationController.php
- EmailVerificationPromptController.php
- NewPasswordController.php
- PasswordResetLinkController.php
- RegisteredUserController.php
- VerifyEmailController.php



- confirm-password.blade.php
- forgot-password.blade.php
- login.blade.php
- register.blade.php
- reset-password.blade.php
- verify-email.blade.php

```
Route::get('/register', [RegisteredUserController::class, 'create'])->middleware('guest');
Route::post('/register', [RegisteredUserController::class, 'store'])->middleware('guest');
Route::get('/login', [AuthenticatedSessionController::class, 'create'])->middleware('guest');
Route::post('/login', [AuthenticatedSessionController::class, 'store'])->middleware('guest');
Route::get('/forgot-password', [PasswordResetLinkController::class, 'create'])->middleware('guest');
Route::post('/forgot-password', [PasswordResetLinkController::class, 'store'])->middleware('guest');
Route::get('/reset-password/{token}', [NewPasswordController::class, 'create'])->middleware('guest');
Route::post('/reset-password', [NewPasswordController::class, 'store'])->middleware('guest');
Route::get('/verify-email', [EmailVerificationPromptController::class, '__invoke'])->middleware('auth');
Route::get('/verify-email/{id}/{hash}', [VerifyEmailController::class, '__invoke'])->middleware('auth');
Route::post('/email/verification-notification', [EmailVerificationNotificationController::class, 'store'])->middleware('auth');
Route::get('/confirm-password', [ConfirmablePasswordController::class, 'show'])->middleware('auth');
Route::post('/confirm-password', [ConfirmablePasswordController::class, 'store'])->middleware('auth');
Route::post('/logout', [AuthenticatedSessionController::class, 'destroy'])->middleware('auth');
```


- ❑ Test chức năng đăng ký, đăng nhập . Mở các địa chỉ
 - ❖ <http://localhost:8000/register> → đăng ký thử sẽ có user trong table users
 - ❖ <http://localhost:8000/login> → đăng nhập thử với user mới đăng ký sẽ vào

The image displays three browser screenshots of a web application's authentication interface. Each screenshot has a red box highlighting the address bar.

- Top Left Screenshot:** The address bar shows `localhost:8000/register`. The page contains four input fields labeled "Name", "Email", "Password", and "Confirm Password". At the bottom right is a dark blue button labeled "REGISTER". At the bottom left is a link labeled "Already registered?".
- Top Right Screenshot:** The address bar shows `localhost:8000/forgot-password`. The page has a heading "Forgot your password? No problem. Just let us know your email address and we will email you a password reset link that will allow you to choose a new one." Below this is an "Email" input field and a dark blue button labeled "EMAIL PASSWORD RESET LINK".
- Bottom Screenshot:** The address bar shows `localhost:8000/login`. The page has "Email" and "Password" input fields. Below the password field is a checkbox labeled "Remember me". At the bottom right is a dark blue button labeled "LOG IN". At the bottom left is a link labeled "Forgot your password?".

❑ Test chức năng forgot-password, cần cấu hình trước mail như sau:

1. Mở file .env, cấu hình các thông số gửi mail như gợi ý

```
MAIL_MAILER=smtp
MAIL_HOST=smtp.gmail.com
MAIL_PORT=465
MAIL_USERNAME=longnv.wordpress@gmail.com
MAIL_PASSWORD=KhongC0SD@u
MAIL_ENCRYPTION=ssl
MAIL_FROM_ADDRESS=longnv.wordpress@gmail.com
MAIL_FROM_NAME="Nguyen Văn Long"
```

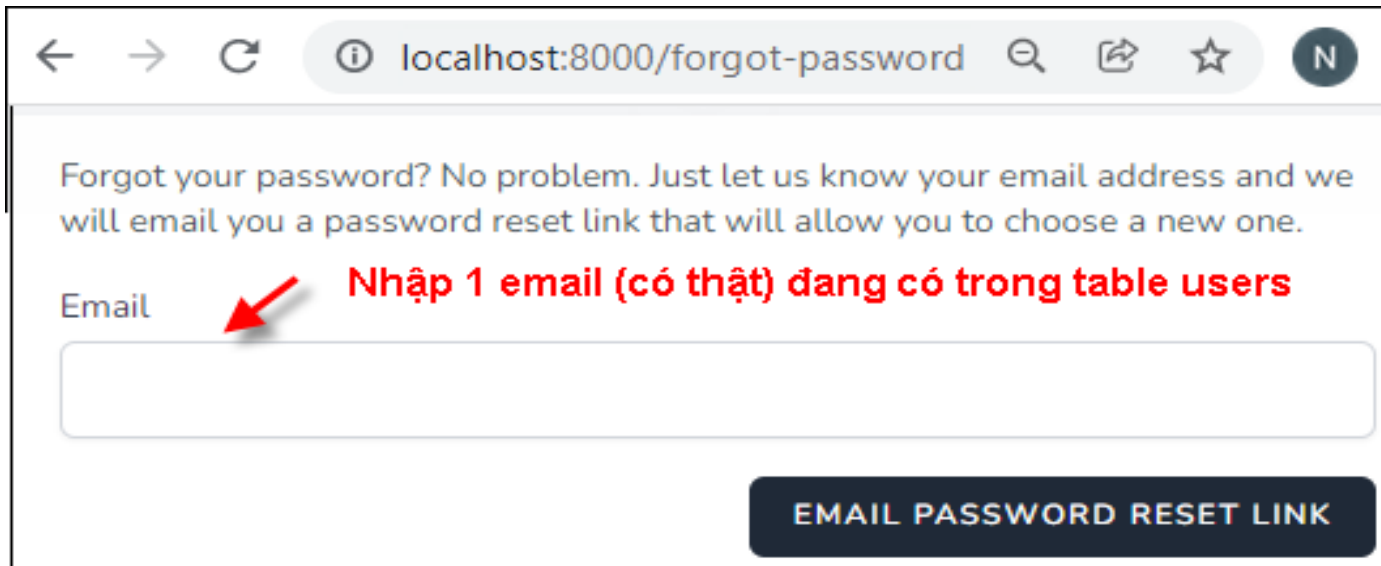
Server mail
Port hoạt động của mail server
Email và password của hộp thư gửi mail
Email người gửi
Tên người gửi

2. Nếu gửi mail từ localhost, thì mở file config/ mail.php, thêm code sau trong mục smtp

```
'stream' => [
    'ssl' => [
        'allow_self_signed' => true,
        'verify_peer' => false,
        'verify_peer_name' => false,
    ],
],
```




- Mở trình duyệt gõ <http://localhost:8000/forgot-password>
- Nhập vào 1 email có thật và đang có trong bảng **users** rồi nhấn nút **Email Password Reset Link**
Một token sẽ được sinh ra và lưu vào table **password_resets** ,
đồng thời 1 email sẽ được gửi tới hộp thư



← → ↻ ⓘ localhost:8000/forgot-password 🔍 📄 ☆ N

Forgot your password? No problem. Just let us know your email address and we will email you a password reset link that will allow you to choose a new one.

Email  **Nhập 1 email (có thật) đang có trong table users**

EMAIL PASSWORD RESET LINK

- ❑ Chú ý: để gửi email thành công, hộp thư gửi mail khai báo trong file .env phải bật tính năng truy cập kém an toàn cho ứng dụng.

Để sử dụng Starter Kit Jetstream, thực hiện như sau:

❑ B1: Tạo project mới và chỉnh cấu hình kết nối đến db

❑ B2: Tải Jetstream: Trong folder project và chạy lệnh:

composer require laravel/Jetstream

❑ B3: Cài jetstream (để tạo views, routes, controllers): chạy các lệnh

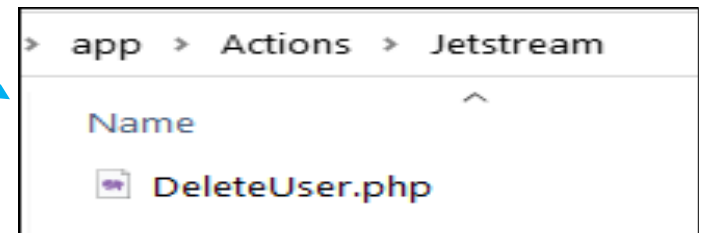
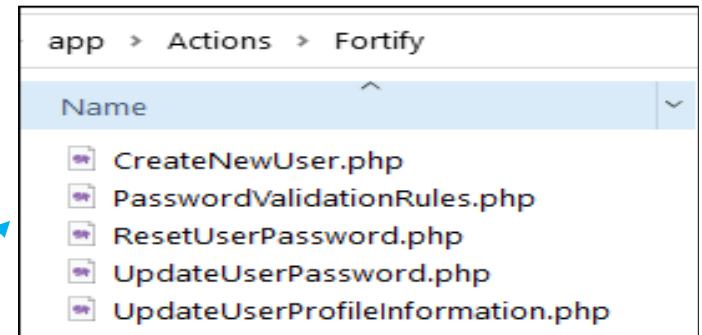
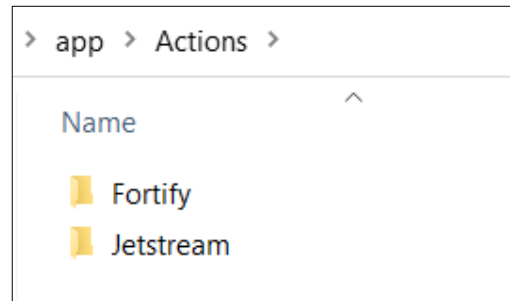
php artisan jetstream:install livewire

npm install

npm run dev

php artisan migrate

- ❑ Cài Jetstream xong, vào folder **app\Actions** sẽ thấy 2 folder con



- ❑ Nếu muốn chèn user thì mở file **database\seeders\DatabaseSeeder.php**, bỏ khóa cho lệnh đang có trong hàm run.

```
public function run()
{
    // \App\Models\User::factory(10)->create();
}
```

Xóa

- ❑ Xong thì chạy **php artisan db:seed** sẽ có 10 user trong bảng users (mỗi user có mật khẩu là password)

- ❑ Gõ lệnh sau để thấy các route được Jetstream hỗ trợ
- php artisan route:list**

Method	URI	Name	Action
GET HEAD	/		Closure
GET HEAD	api/user		Closure
GET HEAD	dashboard	dashboard	Closure
GET HEAD	forgot-password	password.request	Laravel\Fortify\Http\Controllers\Pa
POST	forgot-password	password.email	Laravel\Fortify\Http\Controllers\Pa
GET HEAD	livewire/livewire.js		Livewire\Controllers\LivewireJavaSc
GET HEAD	livewire/livewire.js.map		Livewire\Controllers\LivewireJavaSc
POST	livewire/message/{name}	livewire.message	Livewire\Controllers\HttpConnection
GET HEAD	livewire/preview-file/{filename}	livewire.preview-file	Livewire\Controllers\FilePreviewHan
POST	livewire/upload-file	livewire.upload-file	Livewire\Controllers\FileUploadHand
POST	login		Laravel\Fortify\Http\Controllers\Au
GET HEAD	login	login	Laravel\Fortify\Http\Controllers\Au
POST	logout	logout	Laravel\Fortify\Http\Controllers\Au
GET HEAD	register	register	Laravel\Fortify\Http\Controllers\Re
POST	register		Laravel\Fortify\Http\Controllers\Re
POST	reset-password	password.update	Laravel\Fortify\Http\Controllers\Ne
GET HEAD	reset-password/{token}	password.reset	Laravel\Fortify\Http\Controllers\Ne
GET HEAD	sanctum/csrf-cookie		Laravel\Sanctum\Http\Controllers\Cs

- ❑ Trong **routes/auth.php** sẽ thấy các route được tạo ra

- ❑ Chạy project: **php artisan serve**
- ❑ Xem trang : <http://localhost:8000/login> và đăng nhập thử
- ❑ Mở chức năng profile, có thể cập nhật thông tin, đổi pass, bật xác thực 2 yếu tố...

The screenshot shows a web browser at the URL `localhost:8000/user/profile`. The page title is "Profile". On the right, there is a user menu for "Kane Reynolds" with a dropdown containing "Manage Account" and "Profile". A red arrow points to the "Profile" option. The main content area is divided into two sections. The left section, titled "Update Password", contains three input fields: "Current Password", "New Password", and "Confirm Password". These fields are enclosed in a red rectangular box. The right section, titled "Two Factor Authentication", contains a message stating "You have not enabled two factor authentication." and an "ENABLE" button. A red arrow points to the "ENABLE" button. A "SAVE" button is located at the bottom right of the password update section.

- ❑ Để xác định user đã đăng nhập hay chưa, sử dụng method `check()` của class `Auth` như sau

```
use Illuminate\Support\Facades\Auth;

if (Auth::check()) {
    // The user is logged in...
}
```

- ❑ Hàm `check` thường dùng để xử lý ẩn hiện thông tin trong web. Còn để bảo vệ cho các chức năng của website, bạn có cách hay hơn đó là dùng `middleware`

- ❑ Để lấy thông tin của user đã đăng nhập, bạn dùng method `user()` của class `Auth` như sau:

```
use Illuminate\Support\Facades\Auth;

$user = Auth::user(); //lấy thông tin của user đang đăng nhập
$id = Auth::id(); // chỉ lấy id của user đang đăng nhập
```

- ❑ Cách khác là truy xuất trong đối tượng request.

```
use Illuminate\Http\Request;

class FlightController extends Controller {
    public function update(Request $request) {
        $u = $request->user(); //thông tin user đang Login
    }
}
```

- ❑ Route nào trong website muốn chỉ phục vụ cho những user đã đăng nhập thì sử dụng middleware trong route như sau:

```
Route::get('/download', function () {  
    // Chỉ user đã đăng nhập mới vào được route này  
})->middleware('auth');
```

- ❑ Chú ý: có thể chỉ định nhiều middleware khi cần bảo vệ route

```
Route::get('/download', function () {  
    // Chỉ user đã đăng nhập mới vào được route này  
})->middleware('auth','admin');
```

- ❑ Khi auth middleware xác định chưa đăng nhập, nó sẽ chuyển user đến route có name là **login**.
- ❑ Có thể chỉnh trong **app/Http/Middleware/Authenticate.php**

```
class Authenticate extends Middleware {  
    protected function redirectTo($request) {  
        if (! $request->expectsJson()) {  
            return route('login');  
        }  
    }  
}
```



DEMO

Demo xác thực đăng ký mới user





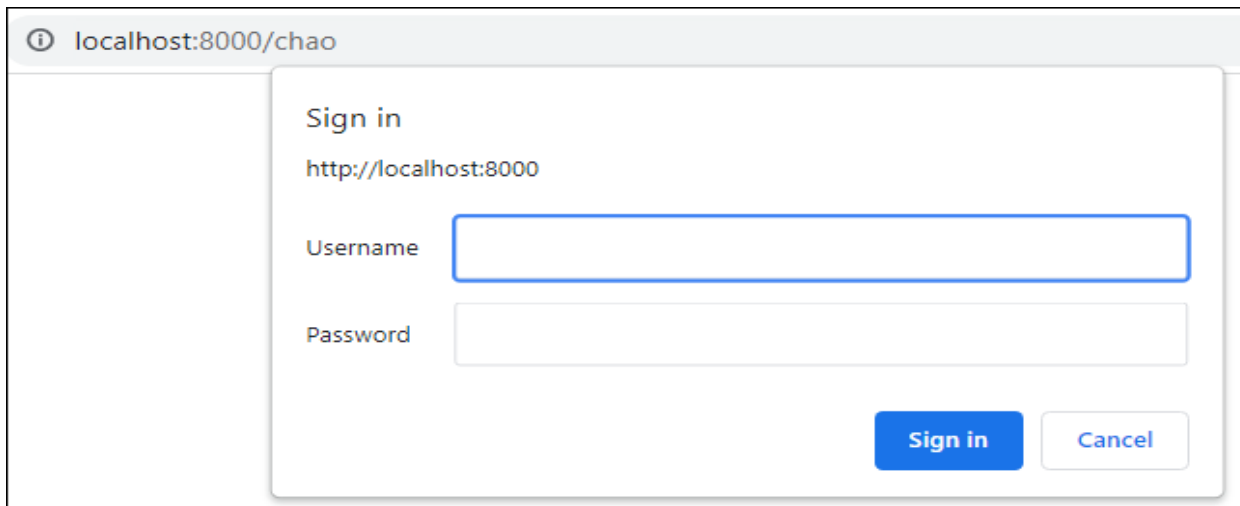
LARAVEL FRAMEWORK

BÀI 6: AUTHENTICATION & MIDDLEWARE

PHẦN 2: MIDDLEWARE

- ❑ Có thể sử dụng HTTP Basic Authentication – phương pháp xác thực đơn giản mà không cần tạo trang login.
- ❑ Mặc định auth.basic middleware sử dụng email để xác thực
- ❑ Khai báo sử dụng trong 1 route như sau

```
Route::get('/chao', function () {  
    echo "Chào bạn";  
})->middleware('auth.basic');
```



The screenshot shows a web browser window with the address bar displaying 'localhost:8000/chao'. A modal dialog box is open, titled 'Sign in' with the URL 'http://localhost:8000'. The dialog contains two input fields: 'Username' and 'Password'. At the bottom right of the dialog are two buttons: 'Sign in' (blue) and 'Cancel' (white with blue border).

- ❑ Để tạo chức năng thoát bằng tay, sử dụng method `logout()` của class `Auth`
- ❑ Khi thoát, có thể redirect user đến route nào tùy ý

```
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

public function logout(Request $request){
    Auth::logout();
    $request->session()->invalidate();
    $request->session()->regenerateToken();
    return redirect('/');
}
```


- ❑ Quá trình xác thực của Laravel phát sinh nhiều sự kiện, có thể lắng nghe các sự kiện trong **app\Providers\EventServiceProvider.php**

```
protected $listen = [  
    'Illuminate\Auth\Events\Registered' => [ 'App\Listeners\LogRegisteredUser', ],  
    'Illuminate\Auth\Events\Attempting' => [ 'App\Listeners\LogAuthenticationAttempt', ],  
    'Illuminate\Auth\Events\Authenticated' => [ 'App\Listeners\LogAuthenticated', ],  
    'Illuminate\Auth\Events>Login' => [ 'App\Listeners\LogSuccessfulLogin', ],  
    'Illuminate\Auth\Events\Failed' => [ 'App\Listeners\LogFailedLogin', ],  
    'Illuminate\Auth\Events\Validated' => [ 'App\Listeners\LogValidated', ],  
    'Illuminate\Auth\Events\Verified' => [ 'App\Listeners\LogVerified', ],  
    'Illuminate\Auth\Events\Logout' => [ 'App\Listeners\LogSuccessfulLogout', ],  
    'Illuminate\Auth\Events\CurrentDeviceLogout' => [ 'App\Listeners\LogCurrentDeviceLogout', ],  
    'Illuminate\Auth\Events\OtherDeviceLogout' => [ 'App\Listeners\LogOtherDeviceLogout', ],  
    'Illuminate\Auth\Events\Lockout' => [ 'App\Listeners\LogLockout', ],  
    'Illuminate\Auth\Events>PasswordReset' => [ 'App\Listeners\LogPasswordReset', ],  
];
```

- ❑ Middleware là những class đặc biệt trong Laravel cung cấp cách thức kiểm soát request của user.
Ví dụ: middleware kiểm tra người dùng đã đăng nhập hay chưa, nếu chưa thì middleware sẽ redirect đến route login. Nếu đã đăng nhập thì cho request xử lý bình thường.
- ❑ Middleware có thể thực hiện xử lý trước và sau luồng xử lý nghiệp vụ chính. Ví dụ: kiểm tra quyền của user, ghi log lỗi, log các hành động của user, thêm http header...
- ❑ Một vài middleware đã có sẵn trong Laravel, như CSRF protection, mã hóa cookie... trong **app/Http/Middleware**

- ❑ Để tạo middleware , thực hiện như sau

```
php artisan make:middleware checkAdmin
```

- ❑ File app\Http\Middleware có hàm handle, là nơi bạn code xử lý để thực hiện tác vụ cần thiết

```
namespace App\Http\Middleware;
use Closure;
use Illuminate\Http\Request;
class checkAdmin {
    public function handle(Request $request, Closure $next) {
        return $next($request);
    }
}
```

- ❑ Middleware chạy trước hay sau response tùy vào cách bạn code:
- ❑ Middleware sau thực hiện tác vụ trước khi request xử lý:

```
class BeforeMiddleware {  
    public function handle($request, Closure $next) {  
        // Code thực hiện task của bạn ←  
        return $next($request);  
    }  
}
```

- ❑ Middleware sau thực hiện tác vụ sau khi request xử lý:

```
class AfterMiddleware {  
    public function handle($request, Closure $next) {  
        $response = $next($request);  
        //Code thực hiện task của bạn ←  
        return $response;  
    }  
}
```

- ❑ **Global middleware:** Là middleware thực thi trong mọi HTTP request, đăng ký trong mục **\$middleware** của **Http/Kernel.php**

```
protected $middleware = [  
    \App\Http\Middleware\TrustProxies::class,  
    \Illuminate\Http\Middleware\HandleCors::class,  
    \App\Http\Middleware\PreventRequestsDuringMaintenance::class,  
    // đăng ký middleware ở đây ←  
];
```

- ❑ **Route middleware :** Là middleware thực thi khi user request vào 1 route cụ thể, đăng ký trong **\$routeMiddleware** của **Http/Kernel.php**

```
protected $routeMiddleware = [  
    'auth' => \App\Http\Middleware\Authenticate::class,  
    'auth.basic' => \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,  
    'cache.headers' => \Illuminate\Http\Middleware\SetCacheHeaders::class,  
    'can' => \Illuminate\Auth\Middleware\Authorize::class,  
    // đăng ký middleware ở đây ←  
];
```

Khai báo middle trong route : dùng method **middleware()** gán cho route.

(1 hoặc nhiều middleware đều được)

```
Route::get('/profile', function () {  
    //  
})->middleware('auth');
```

```
Route::get('/', function () {  
    //  
})->middleware(['first', 'second']);
```












DEMO








-Demo xác thực Change Password



Phần I:

-  Authentication là gì
-  Cấu hình authentication
-  Các starter kit trong laravel
-  Cài Laravel breeze , xem kết quả và test hoạt động
-  Cài Laravel jetstream, xem kết quả và test hoạt động
-  Lấy thông tin của user đã đăng nhập
-  Xác định user đã login hay chưa
-  Route protect – bảo vệ chức năng
-  Chuyển hướng user chưa đăng nhập

Phần II:

-  HTTP Basic Authentication
-  Chức năng Logout
-  Laravel Authentication Events
-  Middleware trong Laravel
-  Tạo mới middleware
-  Middleware & Response
-  Đăng ký middleware với Laravel





Cảm ơn