

LAB 6 MÔN LẬP TRÌNH PHP3: AUTHENTICATION VÀ MIDDLEWARE

MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- ✓ Triển khai authentication trong website
- ✓ Tạo và sử dụng middleware trong Laravel

NỘI DUNG

Chuẩn bị:

1. Cài Laravel vào folder **la6**
2. Tạo database có tên **la6** và file .env cấu hình kết nối đến database vừa tạo.

Bài 1: Cài Starter Kit Laravel Breeze

1. Cài starter kit Laravel Breeze

Trong folder project, lần lượt chạy các lệnh:

composer require laravel/breeze --dev

php artisan breeze:install

npm install

npm run dev

2. Chuẩn bị users

- Vào folder database/migrations, và mở file là **xxx_create_users_table.php**
- Trong hàm up, bổ sung thêm field mới **diachi**, **idgroup** như sau:

```
public function up() {
    Schema::create('users', function (Blueprint $table) {
        $table->id();
        $table->string('name');
        $table->string('email')->unique();
        $table->timestamp('email_verified_at')->nullable();
        $table->string('password');
        $table->rememberToken();
        $table->timestamps();
        $table->string('diachi',100) -> nullable();
        $table->integer('idgroup')->default(0);
    });
}
```

- Chạy **php artisan migrate** , xong vào db sẽ phải thấy table users với các field:

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra |
|----|---|--------------|--------------------|------------|------|---------|----------|----------------|
| 1 | id  | bigint(20) | | UNSIGNED | No | None | | AUTO_INCREMENT |
| 2 | name | varchar(255) | utf8mb4_unicode_ci | | No | None | | |
| 3 | email  | varchar(255) | utf8mb4_unicode_ci | | No | None | | |
| 4 | email_verified_at | timestamp | | | Yes | NULL | | |
| 5 | password | varchar(255) | utf8mb4_unicode_ci | | No | None | | |
| 6 | remember_token | varchar(100) | utf8mb4_unicode_ci | | Yes | NULL | | |
| 7 | created_at | timestamp | | | Yes | NULL | | |
| 8 | updated_at | timestamp | | | Yes | NULL | | |
| 9 | diachi | varchar(100) | utf8mb4_unicode_ci | | Yes | NULL | | |
| 10 | idgroup | int(11) | | | No | 0 | | |

3. Chèn dữ liệu vào bảng users

- Mở seeds/DatabaseSeeder.php, code trong hàm run để chèn users như sau

```
public function run() {
    \App\Models\User::factory(10)->create();
    \DB::table('users')->insert([
        'name' => 'Vui Từng Phút Giây', 'email' => 'vuiqua@gmail.com',
        'password' => bcrypt('hehe'), 'idgroup' => 1, 'diachi'=>'TPHCM'
    ]);
    \DB::table('users')->insert([
        'name' => 'Buồn Từng Phút Giây', 'email' => 'buonqua@gmail.com',
        'password' => bcrypt('huhu'), 'idgroup' => 1, 'diachi'=>'TPHCM'
    ]);
    \DB::table('users')->insert([
        'name' => 'Nguyen Thi Gia Hu', 'email' => 'giahu@gmail.com',
        'password' => bcrypt('hihi'), 'idgroup' => 0, 'diachi'=>'HN'
    ]);
}
```

- Chạy **php artisan db:seed** sẽ phải thấy các user được chèn trong bảng users
- Xem các route authentication được support:

Mở **routes/web.php**, sẽ thấy lệnh `require file auth.php`. Mở file `auth.php` sẽ thấy các route được hỗ trợ. Cũng có thể xem các route bằng lệnh **php artisan route:list**

| | | | |
|----------|--------------------------|---------------------|---------------------|
| GET HEAD | forgot-password | password.request | Auth\Passwo |
| POST | forgot-password | password.email | Auth\Passwo |
| GET HEAD | login | login | Auth\Authentic |
| POST | login | | Auth\Authenti |
| POST | logout | logout | Auth\Authentic |
| GET HEAD | register | register | Auth\Reg |
| POST | register | | Auth\Re |
| POST | reset-password | password.update | Auth |
| GET HEAD | reset-password/{token} | password.reset | Auth\ |
| GET HEAD | sanctum/csrf-cookie | | Laravel\Sanctum |
| GET HEAD | verify-email | verification.notice | Auth\EmailVerificat |
| GET HEAD | verify-email/{id}/{hash} | verification.verify | Auth\Ve |

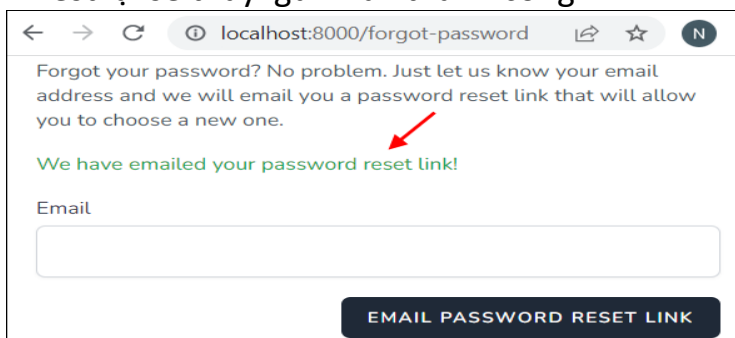
4. Test hoạt động các chức năng authentication

- Test chức năng đăng nhập <http://localhost:8000/login> → sẽ phải đăng nhập được với các user đã tạo
- Test chức năng đăng ký <http://localhost:8000/register> → sẽ phải đăng ký được và đăng nhập được với các user mới
- Test chức năng forgot-password <http://localhost:8000/forgot-password> → sẽ thấy báo lỗi không gửi mail được .
+ Mở file **.env** và cấu hình email cho đúng (email, mật khẩu...)

```
MAIL_MAILER=smtp
MAIL_HOST=smtp.gmail.com
MAIL_PORT=465
MAIL_USERNAME=HopThuCuaBan@gmail.com
MAIL_PASSWORD=MậtKhẩuCủaBạn
MAIL_ENCRYPTION=ssl
MAIL_FROM_ADDRESS="HopThuCuaBan@gmail.com"
MAIL_FROM_NAME="Họ tên người gửi"
```

+ Vào hộp thư vừa khai báo và bật tính năng truy cập kém an toàn cho ử dụng

+ Test lại sẽ thấy gửi mail thành công



- Chú ý 1: Nếu gửi mail không được do lỗi gửi từ localhost thì mở file **config/mail.php** và cấu hình bổ sung như sau, sẽ gửi được

```
'mailers' => [
    'smtp' => [
        'transport' => 'smtp',
        'host' => env('MAIL_HOST', 'smtp.mailgun.org'),
        'port' => env('MAIL_PORT', 587),
        'encryption' => env('MAIL_ENCRYPTION', 'tls'),
        'username' => env('MAIL_USERNAME'),
        'password' => env('MAIL_PASSWORD'),
        'timeout' => null,
        'stream' => [
            'ssl' => [
                'allow_self_signed' => true,
                'verify_peer' => false,
                'verify_peer_name' => false,
            ],
        ],
    ],
],
```

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

- e. Chú ý 2: Nếu gặp lỗi như bên dưới thì phần nhiều do chương trình antivirus chặn port, tạm tắt antivirus là xong.

Symfony \ Component \ Mailer \ Exception \ TransportException PHP 8.1.2 9.3

Connection could not be established with host "ssl://smtp.gmail.com:465": stream_socket_client(): SSL operation failed with code 1. OpenSSL Error messages: error:1416F086:SSL routines:tls_process_server_certificate:certificate...

Bài 2: Hạn chế truy cập và hiện thông tin user login

Vớ các chức năng trong website chỉ cho phép thành viên đã đăng nhập được sử dụng, có 2 cách hiện thực: hạn chế truy cập qua route và code trong controller.

1. Hạn chế truy cập qua route

Trong các route cần hạn chế truy cập , thêm middleware auth để

- Tạo route

```
Route::get('download', function(){
    return view("download");
});
```

- Tạo views/download.blade.php

<p>Chào bạn !</p>
 Đây là trang download software, chỉ dành cho thành viên đã đăng nhập

- Test: Nếu đã login thì hãy logout. Xong test bằng cách vào <http://localhost:8000/download> sẽ thấy view download
- Hạn chế truy cập bằng cách thêm middleware vào route

```
Route::get('download', function(){
    return view("download");
})->middleware('auth');
```


- Vào lại địa chỉ download như trên sẽ không còn thấy view download nữa mà sẽ thấy form login => Đăng nhập vào thành công mới thấy view download hiện ra.

2. Hạn chế truy cập trong controller

Trong hàm __construct của controller, thêm lệnh **\$this->middleware('auth');** sẽ hạn chế đăng nhập cho mọi action trong controller.

- Tạo QuanTriTinController và code như sau:

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
class QuanTriTinController extends Controller {
    function __construct(){
        $this->middleware('auth');
    }
    function index(){
        echo "<h1>Danh sách tin</h1>";
    }
}
```



- Routing

```
use App\Http\Controllers\QuanTriTinController;
Route::get('/quantritin',[QuanTriTinController::class,'index']);
```

- Test: Nếu đã login thì hãy logout. Xong test bằng cách vào <http://localhost:8000/quantritin> sẽ thấy bị buộc phải đăng nhập

3. Hiện thông tin user login

Trong view download, chỉnh view download để được như sau:

```
<p>Chào bạn <?=Auth::user()->name?> </p>
```

Đây là trang download software, chỉ dành cho thành viên đăng nhập

```
<p><a href="thoat">Thoát</a></p>
```

Xem lại thử sẽ thấy tên user login hiện ra

Bài 3: Sử dụng middleware

1. Tạo middleware Quantri

- Chạy lệnh

```
php artisan make:middleware Quantri
```

- Mở `app/Http/ Middleware/Quantri.php` và code trong hàm handle

```
class Quantri {
    public function handle(Request $request, Closure $next) {
        if(auth()->user()->idgroup == 1){
            return $next($request);
        }
        else return redirect('/thongbao');
    }
}
```

2. Đăng ký middleware với Laravel

Mở `app/Http/Kernel.php` rồi code thêm:

```
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,
    'guest' => \App\Http\Middleware\RedirectIfAuthenticated::class,
    'password.confirm' => \Illuminate\Auth\Middleware\RequirePassword::class,
    'signed' => \Illuminate\Routing\Middleware\ValidateSignature::class,
    'throttle' => \Illuminate\Routing\Middleware\ThrottleRequests::class,
    'verified' => \Illuminate\Auth\Middleware\EnsureEmailIsVerified::class,
    'Quantri' => \App\Http\Middleware\Quantri::class,
];
```

3. Bảo vệ route - chỉ dành cho admin

- Bảo vệ 1 route với middleware đã tạo:

```
Route::get('/quantri', function () {
    return view("quantri");
})->middleware('auth', 'Quantri');
```

- Tạo views/quantri.blade.php và code:

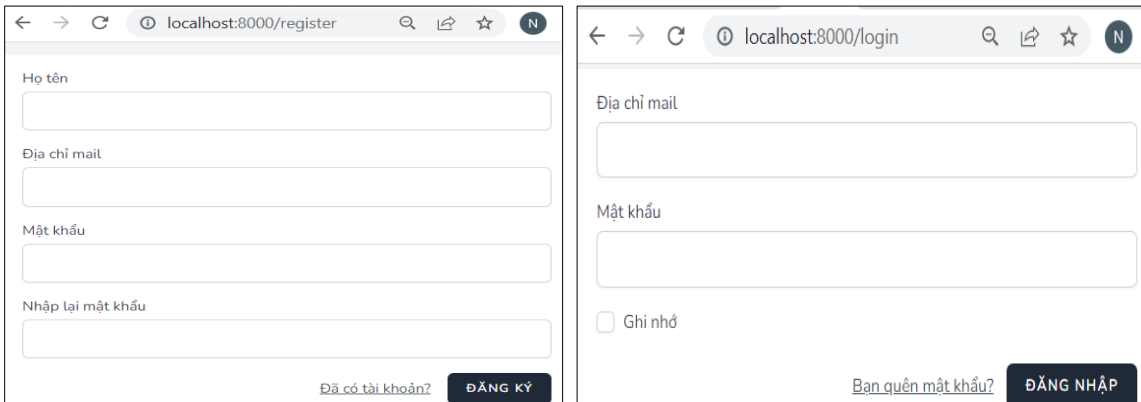
```
<p>Chào bạn <?=Auth::user()->name?> </p>
Đây là trang quản trị chỉ dành cho admin
```

- Test: <http://localhost:8000/quantri> → chỉ có user buongqua và vuiqua vào được (vì sao?)

Bài 4: Thực hiện các yêu cầu theo gợi ý

1. Viết hóa các views đăng ký, đăng nhập

Gợi ý : Mở auth/register.blade.php và auth/login.blade.php + định dạng.



2. Bổ sung field nghenghiiep, phai vào form đăng ký.

Gợi ý:

- Mở views/auth/register.blade.php bổ sung field nghenghiiep, phai
- Mở Controllers/Auth/RegisterController.php bổ sung field nghenghiiep, phai
- Mở App\User.php bổ sung field nghenghiiep, phai

3. Hạn chế truy cập bằng http basic authentication

Tạo 1 route và hạn chế truy cập bằng http basic

```
Route::get('/dl', function(){  
    return view("download");  
})->middleware('auth.basic');
```

*** Yêu cầu nộp bài:

SV nén file (*hoặc share thư mục google drive*) bao gồm các yêu cầu đã thực hiện trên, nộp LMS đúng thời gian quy định của giảng viên. KHÔNG NỘP BÀI COI NHƯ KHÔNG CÓ ĐIỂM.

--- Hết ---