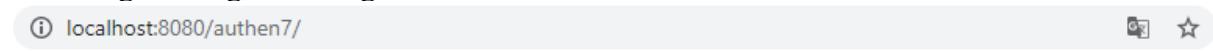
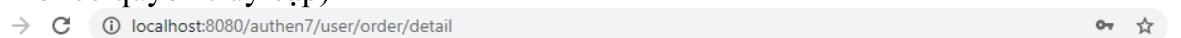


Câu 07. Xây dựng chi tiết demo ứng dụng laravel **xác thực** và **phân role** người dùng trên hệ thống ứng dụng. Hệ thống ứng dụng gồm ba role sau: **admin, user, student**. Đối với role admin toàn quyền trên hệ thống. Một số giao diện ứng dụng như sau:

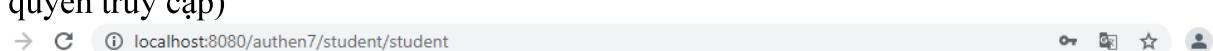
a. Trang chủ người dùng



b. Trang chi tiết đơn hàng của người dùng (chỉ người dùng có role user hoặc admin mới có quyền truy cập)



c. Trang thông tin về sinh viên (chỉ người dùng có role student hoặc admin mới có quyền truy cập)



d. Trang quản trị admin (chỉ người dùng có role admin mới có quyền truy cập)



Gợi ý:

(Lưu ý: trong Laravel 7, mặc định **không** hỗ trợ chức năng **Authentication**. Do đó, muốn sử dụng chức năng này cần phải cài đặt. Các thông số khai báo mặc định về Authentication trong file **config/auth.php**).

- **B1.** Cài đặt gói package laravel's **laravel/ui**

composer require laravel/ui:^2.4

- **B2.** Chạy lệnh sau để cài đặt các thành phần **layout view, registration, login views, routes ...**

php artisan ui vue --auth

Nhắc lý thuyết: => Thực hiện thành công, một vài **authentication controllers** được xây dựng trước với **App\Http\Controllers\Auth** namespace như:

- + **RegisterController:** xử lý người dùng đăng ký mới.
- + **LoginController:** xử lý xác thực.
- + **ForgotPasswordController:** xử lý liên kết gửi email cho đặt lại mật khẩu
- + **ResetPasswordController:** xử lý đặt lại mật khẩu.
- + Nếu ứng dụng **không cần** chức năng **đăng ký thành viên**, bạn có thể vô hiệu hóa chức năng này. Bằng cách **xóa RegisterController** và sửa đổi khai báo route của bạn: **Auth::routes(['register' => false]);**

+ **Đăng ký** các Route xác thực cho một ứng dụng:

Auth::routes();

Route::get('home', 'HomeController@index')->name('home');

+ để **thay đổi** các thông tin **user** khi đăng ký, thực hiện trong **RegisterController**.

- phương thức **validator()**: định nghĩa quy tắc **kiểm tra** tra dữ liệu.

- phương thức **create()**: lưu dữ liệu vào bảng **users** trong csdl.

+ phương thức **Auth::user()**: lấy thông tin user đã login (hoặc sử dụng lớp **Request** ví dụ: **\$request->user()** để lấy thông tin user đã login)

+ phương thức **Auth::id()**: lấy thông tin id của user đã login

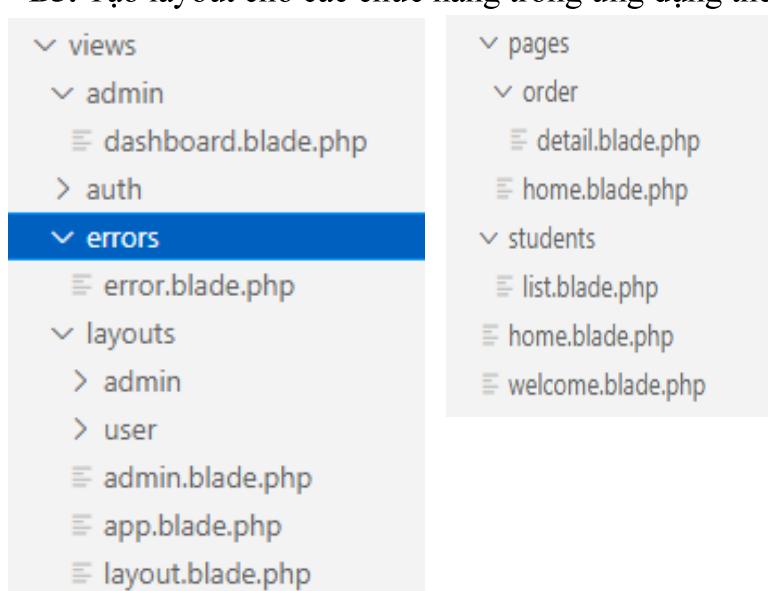
+ phương thức **Auth::check()**: xác định user đã login chưa. Trả về **true** đã login.

+ để bảo vệ các route bởi các user chưa đăng nhập sử dụng phương thức **middleware('auth')**

+ phương thức **redirectTo()** trong file **Http\Middleware/Authenticate.php** sử dụng nếu user chưa đăng nhập, chuyển user tới route có tên login.

+ phương thức **Auth::logout()** đăng xuất ứng dụng.

- **B3.** Tạo layout cho các chức năng trong ứng dụng theo cấu trúc hình sau:



+ Layout master trang chủ người dùng, **layouts\layout.blade.php**

```
<div class="container mt-5 mb-5">
    <div class="row">
        <div class="col-sm-12">
            @include('layouts.user.header')
        </div>
    </div>
    <div class="row">
        <div class="col-sm-4">
            @include('layouts.user.left')
        </div>
        <div class="col-sm-8">
            @yield('main')
        </div>
    </div>
    <div class="row">
        <div class="col-sm-12">
            @include('layouts.user.footer')
        </div>
    </div>
</div>
```

+ Layout master trang quản trị hệ thống, **layouts\admin.blade.php**

```
<div class="container mt-5 mb-5">
    <div class="row">
        <div class="col-sm-12">
            @include('layouts.admin.header')
        </div>
    </div>
    <div class="row">
        <div class="col-sm-4">
            @include('layouts.admin.left')
        </div>
        <div class="col-sm-8">
            @yield('main')
        </div>
    </div>
    <div class="row">
        <div class="col-sm-12">
            @include('layouts.admin.footer')
        </div>
    </div>
</div>
```

+ Layout trang **admin\header.blade.php**

```
<div class="container">
    <div class="row">
        <div class="col-sm-9">
            <header><h4 class="text-primary">Management System</h4></header>
        </div>
        <div class="col-sm-3" style="text-align: right">
            <!-- Authentication Links -->
            @guest
            @else
                Xin chào,
                <a href="">
                    {{ Auth::user()->name }} 
                </a> &ampnbsp&ampnbsp&ampnbsp
            @endguest
        </div>
    </div>
</div>
```

```

<a href="{{ route('logout') }}"
    onclick="event.preventDefault();
    document.getElementById('logout-form').submit();">
    {{__('Logout')}}
</a>

<form id="logout-form" action="{{ route('logout') }}" method="POST">
    @csrf
</form>
@endguest
</div>
</div>
<hr>
+ Layout trang admin\left.blade.php
<h5>Menu</h5>
+ Layout trang admin\footer.blade.php
<hr>
<div class="container">
    <div class="row">
        <div class="col-sm-12">
            <footer>
                <h5 style="text-align:right">
                    &copy; <?php date('Y'); ?> My Laptop Shop
                </h5>
            </footer>
        </div>
    </div>
</div>

```

+ Xây dựng layout tương tự cho người dùng **user\header.blade.php**, **left.blade.php**, **footer.blade.php**.

+ Layout trang **admin\dashboard.blade.php**, **pages\order\detail.blade.php**, **students\list.blade.php**

```

@extends('layouts.admin')
@section('main')
    Trang admin !
@endsection

```

```

@extends('layouts.layout')
@section('main')
    Trang chi tiết đơn hàng người dùng !
@endsection

```

```

@extends('layouts.admin')
@section('main')
    Trang student !
@endsection

```

- **B4.** Mở file **migrations\create_users_table.php**, bổ sung mã lệnh để tạo hai thuộc tính **typeuser**, **status** cho bảng **users** và mở file model **app\User.php** bổ sung mã cho phép truy cập hai thuộc tính vừa tạo thông qua **model** như sau:

```

class CreateUsersTable extends Migration
{
    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->tinyInteger('typeuser')->default(2)->comment('1:admin 2:user 3:student');
        $table->tinyInteger('status')->default(1)->comment('1:active 2:inactive');
        $table->rememberToken();
        $table->timestamps();
    });
}
}

class User extends Authenticatable
{
    use Notifiable;
    protected $fillable = [
        'name', 'email', 'password', 'typeuser', 'status',
    ];

    protected $hidden = [
        'password', 'remember_token',
    ];

    protected $casts = [
        'email_verified_at' => 'datetime',
    ];
}

```

- B5. Mở file **seeds\DatabaseSeeder.php** thêm đoạn mã sau tạo dữ liệu mẫu cho bảng **users** trong cơ sở dữ liệu.

```

class DatabaseSeeder extends Seeder
{
    public function run()
    {
        // $this->call(UserSeeder::class);
        DB::table('users')->insert([
            'name' => 'admin',
            'email' => 'admin@gmail.com',
            'password' => bcrypt('123456'),
            'typeuser' => 1
        ]);
        DB::table('users')->insert([
            'name' => 'user',
            'email' => 'user@gmail.com',
            'password' => bcrypt('123456'),
            'typeuser' => 2
        ]);
        DB::table('users')->insert([
            'name' => 'student',
            'email' => 'student@gmail.com',
            'password' => bcrypt('123456'),
            'typeuser' => 3
        ]);
    }
}

```

- **B6.** Định nghĩa lớp **Middleware\CheckRole middleware** kiểm tra **role** người dùng trong ứng dụng.

```
use Illuminate\Support\Facades\DB;
class CheckRole
{
    public function handle($request, Closure $next,$typeuser = 0)
    {
        $user = DB::table('users')->where('id',auth()->id())->first();
        if(!empty($user->typeuser) && ($user->typeuser === intval($typeuser) || $user->typeuser === 1))
        {
            return $next($request);
        }
        //return abort(401);
        return redirect()->route('homepage');
    }
}
=> Đăng ký sử dụng CheckRole middleware trong file Kernel.php
```

```
class Kernel extends HttpKernel
{
    protected $routeMiddleware = [
        ...
        'checkrole' => \App\Http\Middleware\CheckRole::class,
    ];
}
```

- **B7.** Mở file **routes\web.php** định nghĩa các **route** định tuyến đến các chức năng của ứng dụng.

```
Route::get('/login',function(){
    return view('auth.login');
})->name('login');

Route::get('/register',function(){
    return view('auth.register');
})->name('register');

Auth::routes(['register'=>true]);
Route::get('/home', 'HomeController@index')->name('home');
```

```
//Home page
Route::get('/', function () {
    return view('pages.home');
})->name('homepage');

Route::middleware(['auth'])->group(function () {
    Route::middleware(['checkrole:2'])->group(function () {
        //Role user
        Route::prefix('user')->group(function () {
            Route::get('/order/detail',function(){
                return view('pages.order.detail');
            })->name('userorderdetail');
        });
    });
});

Route::middleware(['checkrole:1'])->group(function () {
    //Role admin
    Route::prefix('admin')->group(function () {
```

```
Route::get('/', function () {
    return view('admin.dashboard');
})->name('adminpage');

});

//Role student
Route::middleware(['checkrole:3'])->group(function () {
    Route::prefix('student')->group(function () {
        Route::get('/student', function () {
            return view('students.list');
        })->name('studentpage');
    });
});
```

- B8. Chạy và kiểm tra các chức năng đã xây dựng cho ứng dụng.

Bài tập làm thêm: em hãy dựa trên các bài tập 5, 6, 7 đã làm, tự xây dựng riêng cơ chế xác thực và phân role cho người dùng trên hệ thống ứng dụng laravel.