**Authorization**

Tạo project authorization bằng cách copy sourcecode đính kèm (authorization)

Tạo domain authorization.laravel.com và trỏ vào thư mục authorization (dùng virtual host)

Tạo database authorization-laravel-k37 (lưu ý k37 sẽ thay đổi theo khóa học)

Import file authorization\database\database.sql vào database ở trên

Chỉ định database trong file .env

DB\_DATABASE= authorization-laravel-k37

Thêm columns deleted\_at và khóa ngoại user\_id vào các bản students, subjects, registers – thêm vào function up() trong mỗi file tương ứng của thư mục database/migrations

php artisan make:migration add-columns-to-student --table=students

Schema::table('students', function (Blueprint $table) {

            $table->unsignedBigInteger('user\_id')->default(23);

            $table->foreign('user\_id')->references('id')->on('users');

$table->softDeletes();

        });

php artisan make:migration add-columns-to-subject --table=subjects

Schema::table('subjects', function (Blueprint $table) {

            //

            $table->unsignedBigInteger('user\_id')->default(23);

            $table->foreign('user\_id')->references('id')->on('users');

$table->softDeletes();

        });

php artisan make:migration add-columns-to-register --table=registers

Schema::table('registers', function (Blueprint $table) {

            //

            $table->unsignedBigInteger('user\_id')->default(23);

            $table->foreign('user\_id')->references('id')->on('users');

$table->softDeletes();

        });

Tạo các tables cần thiết cho authorization

php artisan make:migration create\_roles\_table --create=roles

Schema::create('roles', function (Blueprint $table) {

            $table->increments('id');

            $table->string('name');

        });

php artisan make:migration create\_permissions\_table --create=permissions

Schema::create('permissions', function (Blueprint $table) {

            $table->increments('id');

            $table->string('name');

        });

php artisan make:migration create\_role\_user\_table --create=role\_user

 Schema::create('role\_user', function (Blueprint $table) {

            $table->unsignedBigInteger('user\_id');

            $table->unsignedInteger('role\_id');

            $table->foreign('user\_id')->references('id')->on('users')->onDelete('cascade');

            $table->foreign('role\_id')->references('id')->on('roles')->onDelete('cascade');

        });

php artisan make:migration create\_permission\_role\_table --create=permission\_role

Schema::create('permission\_role', function (Blueprint $table) {

            $table->unsignedInteger('permission\_id');

            $table->unsignedInteger('role\_id');

            $table->foreign('permission\_id')->references('id')->on('permissions')->onDelete('cascade');

            $table->foreign('role\_id')->references('id')->on('roles')->onDelete('cascade');

        });

Chạy migrate:

php artisan migrate

Tạo seeder để chèn data vào các 4 bảng mới tạo ở trên, cập nhật function run() bằng dòng code bên dưới vào database\seeders\DatabaseSeeder.php

use Illuminate\Support\Facades\DB;

public function run()

    {

        DB::table('permissions')->insert([

            ['id' => 1, 'name' => 'view\_student'],

            ['id' => 2,'name' => 'update\_student'],

            ['id' => 3,'name' => 'create\_student'],

            ['id' => 4,'name' => 'delete\_student'],

            ['id' => 5,'name' => 'restore\_student'],

            ['id' => 6,'name' => 'force\_delete\_student'],

            ['id' => 7,'name' => 'view\_subject'],

            ['id' => 8,'name' => 'update\_subject'],

            ['id' => 9,'name' => 'create\_subject'],

            ['id' => 10,'name' => 'delete\_subject'],

            ['id' => 11,'name' => 'restore\_subject'],

            ['id' => 12,'name' => 'force\_delete\_subject'],

            ['id' => 13,'name' => 'view\_register'],

            ['id' => 14,'name' => 'update\_register'],

            ['id' => 15,'name' => 'create\_register'],

            ['id' => 16,'name' => 'delete\_register'],

            ['id' => 17,'name' => 'restore\_register'],

            ['id' => 18,'name' => 'force\_delete\_register'],

        ]);

        DB::table('roles')->insert([

            ['id' => 1,'name' => 'admin'],

            ['id' => 2,'name' => 'viewer'],

        ]);

        DB::table('role\_user')->insert([

            [

                'role\_id' => 1,

                'user\_id' => 23,

            ],

            [

                'role\_id' => 2,

                'user\_id' => 24,

            ]

        ]

        );

        DB::table('permission\_role')->insert([

            ['permission\_id' => 1, 'role\_id' => 1],

            ['permission\_id' => 2, 'role\_id' => 1],

            ['permission\_id' => 3, 'role\_id' => 1],

            ['permission\_id' => 4, 'role\_id' => 1],

            ['permission\_id' => 5, 'role\_id' => 1],

            ['permission\_id' => 6, 'role\_id' => 1],

            ['permission\_id' => 7, 'role\_id' => 1],

            ['permission\_id' => 8, 'role\_id' => 1],

            ['permission\_id' => 9, 'role\_id' => 1],

            ['permission\_id' => 10, 'role\_id' => 1],

            ['permission\_id' => 11, 'role\_id' => 1],

            ['permission\_id' => 12, 'role\_id' => 1],

            ['permission\_id' => 13, 'role\_id' => 1],

            ['permission\_id' => 14, 'role\_id' => 1],

            ['permission\_id' => 15, 'role\_id' => 1],

            ['permission\_id' => 16, 'role\_id' => 1],

            ['permission\_id' => 17, 'role\_id' => 1],

            ['permission\_id' => 18, 'role\_id' => 1],

        ]);

    }

Run seeder để import data vào các tables tương ứng

php artisan db:seed

Tạo model app\Models\Permission.php

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Permission extends Model

{

    public $timestamps = false;

}

Thêm hàm user() vào model app\Models\Student.php

public function user()

    {

        return $this->belongsTo('App\Models\User');

    }

Tao model app\Models\Role.php

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Role extends Model

{

    public $timestamps = false;

    public function permissions()

    {

        return $this->belongsToMany(Permission::class);

    }

}

?>

Thêm đoạn code bên dưới vào model User.php

protected $permissionList = null;

    public function roles()

    {

        return $this->belongsToMany(Role::class);

    }

    public function hasRole($role)

    {

        if (is\_string($role)) {

            return $this->roles->contains('name', $role);

        }

        return false;

    }

    public function hasPermission($permission = null)

    {

        if (is\_null($permission)) {

            return $this->getPermissions()->count();

        }

        if (is\_string($permission)) {

            return $this->getPermissions()->contains('name', $permission);

        }

        return false;

    }

    private function getPermissions()

    {

        $role = $this->roles->first();

        if ($role) {

            if (! $role->relationLoaded('permissions')) {

                $this->roles->load('permissions');

            }

            $this->permissionList = $this->roles->pluck('permissions')->flatten();

        }

        return $this->permissionList ?? collect();

    }

Tạo policy (chức năng: sau khi đăng nhập thì kiểm tra xem user đó có quyền gì)

php artisan make:policy StudentPolicy --model=Student (--model=Student có nghĩa là add thêm tham số thứ 2 là đối tượng Student cho các function của StudentPolicy – app/Policies/StudentPolicy.php)

Cập nhật code bên dưới vào file StudentPolicy.php

<?php

namespace App\Policies;

use App\Models\Student;

use App\Models\User;

use Illuminate\Auth\Access\HandlesAuthorization;

class StudentPolicy

{

    use HandlesAuthorization;

    /\*\*

     \* Determine whether the user can view any models.

     \*

     \* @param  \App\Models\User  $user

     \* @return mixed

     \*/

    public function viewAny(User $user)

    {

        //

    }

    /\*\*

     \* Determine whether the user can view the model.

     \*

     \* @param  \App\Models\User  $user

     \* @param  \App\Models\Student  $student

     \* @return mixed

     \*/

    public function view(User $user, Student $student)

    {

        //

    }

    /\*\*

     \* Determine whether the user can create models.

     \*

     \* @param  \App\Models\User  $user

     \* @return mixed

     \*/

    public function create(User $user)

    {

        //

        return $user->hasPermission("create\_student");

    }

    /\*\*

     \* Determine whether the user can update the model.

     \*

     \* @param  \App\Models\User  $user

     \* @param  \App\Models\Student  $student

     \* @return mixed

     \*/

    public function update(User $user, Student $student)

    {

        //

        return $user->id == $student->user\_id || $user->hasPermission("update\_student");

    }

    /\*\*

     \* Determine whether the user can delete the model.

     \*

     \* @param  \App\Models\User  $user

     \* @param  \App\Models\Student  $student

     \* @return mixed

     \*/

    public function delete(User $user, Student $student)

    {

        //

        return $user->id == $student->user\_id || $user->hasPermission("update\_student");

    }

    /\*\*

     \* Determine whether the user can restore the model.

     \*

     \* @param  \App\Models\User  $user

     \* @param  \App\Models\Student  $student

     \* @return mixed

     \*/

    public function restore(User $user, Student $student)

    {

        //

        return $user->id == $student->user\_id || $user->hasPermission("restore\_student");

    }

    /\*\*

     \* Determine whether the user can permanently delete the model.

     \*

     \* @param  \App\Models\User  $user

     \* @param  \App\Models\Student  $student

     \* @return mixed

     \*/

    public function forceDelete(User $user, Student $student)

    {

        //

        return $user->hasPermission("force\_delete\_student");

    }

}

Đăng ký policy ở file app\Providers\AuthServiceProvider.php (mặc định của laravel thì model Student sẽ vào StudentPolicy, model Subject sẽ vào SubjectPolicy,… nếu muốn trỏ tới policy khác thì mới cần khai báo trong file này)

use App\Models\Student;

use App\Policies\StudentPolicy;

protected $policies = [

        // 'App\Models\Model' => 'App\Policies\ModelPolicy',

        //Student::class => StudentPolicy::class,

    ];

Khai báo Gate để hỗ trợ authorize, việc khai báo này ở hàm boot() của file app\Providers\AuthServiceProvider.php

public function boot()

    {

        $this->registerPolicies();

        Gate::define('view-student', [StudentPolicy::class, 'view']);

        Gate::define('create-student', [StudentPolicy::class, 'create']);

        Gate::define('update-student', [StudentPolicy::class, 'update']);

        Gate::define('delete-student', [StudentPolicy::class, 'delete']);

        Gate::define('restore-student', [StudentPolicy::class, 'restore']);

        Gate::define('force-delete-student', [StudentPolicy::class, 'forceDelete']);

    }

Nhớ use Gate trong StudentController.php

use Illuminate\Support\Facades\Gate;

Cập nhật hàm create của StudentController.php

//

        if (!Gate::allows("create-student")) {

            abort(403);

        }

        return view('student.create');