Hệ thống xác thực tài khoản bằng Email

Phân tích

### **Bước 1: hiệu chỉnh cấu trúc database**

#### **Hiệu chỉnh table users**

Mình cần thêm cột active kiểu bool để nhận biết tài khoản nào đã được kích hoạt hay chưa?

* Nếu giá trị = 1: user này đã kích hoạt tài khoản rồi.
* Nếu giá trị = 0: user này chưa kích hoạt tài khoản.

php artisan make:migration alter\_users\_table --table=users

File: alter\_users\_table.php

**<?php**

use Illuminate\Support\Facades\Schema;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Database\Migrations\Migration;

class AlterUsersTable extends Migration

{

/\*\*

\* Run the migrations.

\*

\* @return void

\*/

public function up()

{

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

$table->boolean('active')->default(false);

});

}

/\*\*

\* Reverse the migrations.

\*

\* @return void

\*/

public function down()

{

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

$table->dropColumn('active');

});

}

}

#### **Thêm table user\_activations**

Thêm table user\_activations để lưu trữ tài khoản và mã kích hoạt tài khoản

php artisan make:migration create\_user\_activations\_table --create=user\_activations

File: create\_user\_activations\_table.php

**<?php**

use Illuminate\Support\Facades\Schema;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Database\Migrations\Migration;

class CreateUserActivationsTable extends Migration

{

/\*\*

\* Run the migrations.

\*

\* @return void

\*/

public function up()

{

Schema::create('user\_activations', function (Blueprint $table) {

$table->integer('user\_id')->unsigned();

$table->string('activation\_code')->index();

$table->timestamps();

});

}

/\*\*

\* Reverse the migrations.

\*

\* @return void

\*/

public function down()

{

Schema::dropIfExists('user\_activations');

}

}

#### **Migrate database**

Yêu cầu Laravel áp dụng các thay đổi cấu trúc database

php artisan migrate

### **Bước 2: tạo hệ thống xác thực mặc định của Laravel**

Trong Laravel có hệ thống xác thực mặc định, bao gồm chức năng login, logout, register. Mặc định khi cài đặt Laravel sẽ tắt chức năng này. Kích hoạt tính năng Xác thực mặc định bằng câu lệnh:

php artisan make:auth

### **Bước 3: tạo Model để quản lý dữ liệu cho table user\_activations**

Tạo Model cho table user\_activations bằng câu lệnh:

php artisan make:model UserActivation

File UserActivation.php

**<?php**

namespace App;

use Illuminate\Database\Eloquent\Model;

use Carbon\Carbon;

class UserActivation extends Model

{

protected $table = 'user\_activations';

// Tạo mã xác thực

protected function getToken()

{

return hash\_hmac('sha256', str\_random(40), config('app.key'));

}

// Tạo xác thực người dùng

public function createActivation($user)

{

$activation = $this->getActivation($user);

if (!$activation) {

return $this->createToken($user);

}

return $this->regenerateToken($user);

}

// Tạo lại dòng dữ liệu có chứa tài khoản và mã xác thực

private function regenerateToken($user)

{

$token = $this->getToken();

UserActivation::where('user\_id', $user->id)->update([

'activation\_code' => $token,

'created\_at' => new Carbon()

]);

return $token;

}

// Tạo dòng dữ liệu có chứa tài khoản và mã xác thực

private function createToken($user)

{

$token = $this->getToken();

UserActivation::insert([

'user\_id' => $user->id,

'activation\_code' => $token,

'created\_at' => new Carbon()

]);

return $token;

}

// Tìm user cần xác thực theo "user\_id"

public function getActivation($user)

{

return UserActivation::where('user\_id', $user->id)->first();

}

// Tìm user cần xác thực theo "activation\_code"

public function getActivationByToken($token)

{

return UserActivation::where('activation\_code', $token)->first();

}

// Xóa tài khoản cần xác thực

public function deleteActivation($token)

{

UserActivation::where('activation\_code', $token)->delete();

}

}

### **Bước 4: tạo Route để quản lý việc xác thực tài khoản**

File routes/web.php

Auth::routes();

Route::get('user/activation/{token}', 'Auth\RegisterController@activateUser')->name('user.activate');

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

### **Bước 5: tạo class Mailable để quản lý việc gởi mail có link kích hoạt tài khoản**

#### **Sử dụng câu lệnh sau để tạo class để quản lý gởi mail kích hoạt:**

php artisan make:mail UserActivationEmail

File app/Mail/UserActivationEmail.php

**<?php**

namespace App\Mail;

use Illuminate\Bus\Queueable;

use Illuminate\Mail\Mailable;

use Illuminate\Queue\SerializesModels;

use Illuminate\Contracts\Queue\ShouldQueue;

class UserActivationEmail extends Mailable

{

use Queueable, SerializesModels;

// Lưu trữ thông tin user đăng nhập

protected $user;

/\*\*

\* Create a new message instance.

\*

\* @return void

\*/

public function \_\_construct($user)

{

$this->user = $user;

}

/\*\*

\* Build the message.

\*

\* @return $this

\*/

public function build()

{

return $this->view('emails.user-activation')->with('user', $this->user);

}

}

#### **Tạo mẫu giao diện (view) khi gởi mail kích hoạt tài khoản**

Tạo file resources/emails/user-activation.blade.php

<p>

Chào mừng {{ $user->name }} đã đăng ký thành viên tại sunshine.com. Bạn hãy click vào đường link sau đây để hoàn tất việc đăng ký.

</br>

<a href="{{ $user->activation\_link }}">{{ $user->activation\_link }}</a>

</p>

### **Bước 6: tạo class dịch vụ quản lý việc gởi mail kèm code kích hoạt**

File app/Classes/ActivationService.php

**<?php**

namespace App\Classes;

use Mail;

use App\UserActivation;

use App\Mail\UserActivationEmail;

use App\User;

class ActivationService

{

protected $resendAfter = 24; // Gởi lại mã xác nhận sau 24h

protected $userActivation;

public function \_\_construct(UserActivation $userActivation)

{

$this->userActivation = $userActivation;

}

public function sendActivationMail($user)

{

if ($user->activated || !$this->shouldSend($user)) return;

$token = $this->userActivation->createActivation($user);

$user->activation\_link = route('user.activate', $token);

$mailable = new UserActivationEmail($user);

Mail::to($user->email)->send($mailable);

}

public function activateUser($token)

{

$activation = $this->userActivation->getActivationByToken($token);

if ($activation === null) return null;

$user = User::find($activation->user\_id);

$user->active = true;

$user->save();

$this->userActivation->deleteActivation($token);

return $user;

}

private function shouldSend($user)

{

$activation = $this->userActivation->getActivation($user);

return $activation === null || strtotime($activation->created\_at) + 60 \* 60 \* $this->resendAfter < time();

}

}

### **Bước 7: thực hiện override các hàm Đăng ký, Đăng nhập mặc định của Laravel**

file app/Http/Controllers/Auth/RegisterController.php

**<?php**

namespace App\Http\Controllers\Auth;

use App\User;

use App\Http\Controllers\Controller;

use Illuminate\Support\Facades\Validator;

use Illuminate\Foundation\Auth\RegistersUsers;

use App\Classes\ActivationService;

use Illuminate\Http\Request;

use Illuminate\Auth\Events\Registered;

class RegisterController extends Controller

{

/\*

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

| Register Controller

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

|

| This controller handles the registration of new users as well as their

| validation and creation. By default this controller uses a trait to

| provide this functionality without requiring any additional code.

|

\*/

use RegistersUsers;

/\*\*

\* Where to redirect users after registration.

\*

\* @var string

\*/

protected $redirectTo = '/home';

protected $activationService;

/\*\*

\* Create a new controller instance.

\*

\* @return void

\*/

public function \_\_construct(ActivationService $activationService)

{

$this->middleware('guest');

$this->activationService = $activationService;

}

/\*\*

\* Get a validator for an incoming registration request.

\*

\* @param array $data

\* @return \Illuminate\Contracts\Validation\Validator

\*/

protected function validator(array $data)

{

return Validator::make($data, [

'name' => 'required|string|max:255',

'email' => 'required|string|email|max:255|unique:users',

'password' => 'required|string|min:6|confirmed',

]);

}

/\*\*

\* Create a new user instance after a valid registration.

\*

\* @param array $data

\* @return \App\User

\*/

protected function create(array $data)

{

return User::create([

'name' => $data['name'],

'email' => $data['email'],

'password' => bcrypt($data['password']),

]);

}

/\*\*

\* Handle a registration request for the application.

\*

\* @param \Illuminate\Http\Request $request

\* @return \Illuminate\Http\Response

\*/

public function register(Request $request)

{

$this->validator($request->all())->validate();

$user = $this->create($request->all());

event(new Registered($user));

$this->activationService->sendActivationMail($user);

return redirect('/login')->with('status', 'Bạn hãy kiểm tra email và thực hiện xác thực theo hướng dẫn.');

}

public function activateUser($token)

{

if ($user = $this->activationService->activateUser($token)) {

auth()->login($user);

return redirect('/login');

}

abort(404);

}

}

file app/Http/Controllers/Auth/LoginController.php

**<?php**

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;

use Illuminate\Foundation\Auth\AuthenticatesUsers;

use App\Classes\ActivationService;

use Illuminate\Http\Request;

class LoginController extends Controller

{

/\*

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

| Login Controller

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

|

| This controller handles authenticating users for the application and

| redirecting them to your home screen. The controller uses a trait

| to conveniently provide its functionality to your applications.

|

\*/

use AuthenticatesUsers;

/\*\*

\* Where to redirect users after login.

\*

\* @var string

\*/

protected $redirectTo = '/home';

protected $activationService;

/\*\*

\* Create a new controller instance.

\*

\* @return void

\*/

public function \_\_construct(ActivationService $activationService)

{

$this->middleware('guest')->except('logout');

$this->activationService = $activationService;

}

/\*\*

\* The user has been authenticated.

\*

\* @param \Illuminate\Http\Request $request

\* @param mixed $user

\* @return mixed

\*/

protected function authenticated(Request $request, $user)

{

if (!$user->active) {

$this->activationService->sendActivationMail($user);

auth()->logout();

return back()->with('warning', 'Bạn cần xác thực tài khoản, chúng tôi đã gửi mã xác thực vào email của bạn, hãy kiểm tra và làm theo hướng dẫn.');

}

return redirect()->intended($this->redirectPath());

}

}

### **Bước 8: hiệu chỉnh giao diện View mặc định để hiển thị thông báo chưa kích hoạt tài khoản**

file resources/views/auth/login.blade.php

@extends('layouts.app')

@section('content')

<div class="container">

<div class="row">

<div class="col-md-8 col-md-offset-2">

<div class="panel panel-default">

<div class="panel-heading">Login</div>

<div class="panel-body">

@if (session('warning'))

<span class="alert alert-warning help-block">

<strong>{{ session('warning') }}</strong>

</span>

@endif

<form class="form-horizontal" method="POST" action="{{ route('login') }}">

{{ csrf\_field() }}

<div class="form-group{{ $errors->has('email') ? ' has-error' : '' }}">

<label for="email" class="col-md-4 control-label">E-Mail Address</label>

<div class="col-md-6">

<input id="email" type="email" class="form-control" name="email" value="{{ old('email') }}" required autofocus>

@if ($errors->has('email'))

<span class="help-block">

<strong>{{ $errors->first('email') }}</strong>

</span>

@endif

</div>

</div>

<div class="form-group{{ $errors->has('password') ? ' has-error' : '' }}">

<label for="password" class="col-md-4 control-label">Password</label>

<div class="col-md-6">

<input id="password" type="password" class="form-control" name="password" required>

@if ($errors->has('password'))

<span class="help-block">

<strong>{{ $errors->first('password') }}</strong>

</span>

@endif

</div>

</div>

<div class="form-group">

<div class="col-md-6 col-md-offset-4">

<div class="checkbox">

<label>

<input type="checkbox" name="remember" {{ old('remember') ? 'checked' : '' }}> Remember Me

</label>

</div>

</div>

</div>

<div class="form-group">

<div class="col-md-8 col-md-offset-4">

<button type="submit" class="btn btn-primary">

Login

</button>

<a class="btn btn-link" href="{{ route('password.request') }}">

Forgot Your Password?

</a>

</div>

</div>

</form>

</div>

</div>

</div>

</div>

</div>

@endsection