Ejercicio 17

Creamos nuevo proyecto laravel e instalamos ui con bootstrap

```
laravel new ejercicio17
```

Creamos users_verify

```
php artisan make:migration create_users_verify_table
```

la migracion

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class UsersVerify extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('users_verify', function (Blueprint $table) {
            $table->integer('user_id');
            $table->string('token');
            $table->timestamps();
        });

        Schema::table('users', function (Blueprint $table) {
            $table->boolean('is_email_verified')->default(0);
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
    }
}
```

Modelo user.php

```
<?php

namespace App\Models;

use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;

class User extends Authenticatable
{
    use HasFactory, Notifiable;

    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $fillable = [
        'name',
        'email',
        'password',
        'is_email_verified'
    ];

    /**
     * The attributes that should be hidden for arrays.
     *
     * @var array
     */
    protected $hidden = [
        'password',
        'remember_token',
    ];

    /**
     * The attributes that should be cast to native types.
     *
     * @var array
     */
    protected $casts = [
        'email_verified_at' => 'datetime',
    ];
}
```

Creamos UserVerify.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class UserVerify extends Model
{
    use HasFactory;

    public $table = "users_verify";

    /**
     * Write code on Method
     *
     * @return response()
     */
    protected $fillable = [
        'user_id',
        'token',
    ];

    /**
     * Write code on Method
     *
     * @return response()
     */
    public function user()
    {
        return $this->belongsTo(User::class);
    }
}
```

Añadimos la ruta en web.php

```
<?php

use Illuminate\Support\Facades\Route;

use App\Http\Controllers\Auth\AuthController;

/*
|--------------------------------------------------------------------------
| Web Routes
|--------------------------------------------------------------------------
```

```
    |
    | Here is where you can register web routes for your application. These
    | routes are loaded by the RouteServiceProvider within a group which
    | contains the "web" middleware group. Now create something great!
    |
    */

    Route::get('login', [AuthController::class, 'index'])->name('login');
    Route::post('post-login', [AuthController::class, 'postLogin'])-
>name('login.post');
    Route::get('registration', [AuthController::class, 'registration'])-
>name('register');
    Route::post('post-registration', [AuthController::class, 'postRegistration'])-
>name('register.post');
    Route::get('logout', [AuthController::class, 'logout'])->name('logout');

    /* New Added Routes */
    Route::get('dashboard', [AuthController::class, 'dashboard'])-
>middleware(['auth', 'is_verify_email']);
    Route::get('account/verify/{token}', [AuthController::class,
'verifyAccount'])->name('user.verify');
```

modificamos authController

```
    <?php

    namespace App\Http\Controllers\Auth;

    use App\Http\Controllers\Controller;
    use Illuminate\Http\Request;
    use Illuminate\Support\Facades\Auth;
    use Session;
    use App\Models\User;
    use App\Models\UserVerify;
    use Hash;
    use Illuminate\Support\Str;
    use Mail;

    class AuthController extends Controller
    {
        /**
         * Write code on Method
         *
         * @return response()
         */
        public function index()
        {
            return view('auth.login');
        }

        /**
```

```
             * Write code on Method
             *
             * @return response()
             */
            public function registration()
            {
                return view('auth.registration');
            }

            /**
             * Write code on Method
             *
             * @return response()
             */
            public function postLogin(Request $request)
            {
                $request->validate([
                    'email' => 'required',
                    'password' => 'required',
                ]);

                $credentials = $request->only('email', 'password');
                if (Auth::attempt($credentials)) {
                    return redirect()->intended('dashboard')
                                ->withSuccess('You have Successfully loggedin');
                }

                return redirect("login")->withSuccess('Oppes! You have entered invalid
    credentials');
            }

            /**
             * Write code on Method
             *
             * @return response()
             */
            public function postRegistration(Request $request)
            {
                $request->validate([
                    'name' => 'required',
                    'email' => 'required|email|unique:users',
                    'password' => 'required|min:6',
                ]);

                $data = $request->all();
                $createUser = $this->create($data);

                $token = Str::random(64);

                UserVerify::create([
                    'user_id' => $createUser->id,
                    'token' => $token
                    ]);
                Mail::send('email.emailVerificationEmail', ['token' => $token],
    function($message) use($request){
                    $message->to($request->email);
                    $message->subject('Email Verification Mail');
                });

                return redirect("dashboard")->withSuccess('Great! You have
    Successfully loggedin');
            }

            /**
             * Write code on Method
             *
             * @return response()
             */
            public function dashboard()
            {
                if(Auth::check()){
                    return view('dashboard');
                }

                return redirect("login")->withSuccess('Opps! You do not have access');
            }

            /**
             * Write code on Method
             *
             * @return response()
             */
            public function create(array $data)
            {
            return User::create([
                'name' => $data['name'],
                'email' => $data['email'],
                'password' => Hash::make($data['password'])
            ]);
            }

            /**
             * Write code on Method
             *
             * @return response()
             */
            public function logout() {
                Session::flush();
                Auth::logout();

                return Redirect('login');
            }
            /**
             * Write code on Method
             *
             * @return response()
             */
            public function verifyAccount($token)
            {
                $verifyUser = UserVerify::where('token', $token)->first();

                $message = 'Sorry your email cannot be identified.';

                if(!is_null($verifyUser) ){
                    $user = $verifyUser->user;

                    if(!$user->is_email_verified) {
                        $verifyUser->user->is_email_verified = 1;
                        $verifyUser->user->save();
                        $message = "Your e-mail is verified. You can now login.";
                    } else {
                        $message = "Your e-mail is already verified. You can now 
    login.";
                    }
                }

            return redirect()->route('login')->with('message', $message);
            }
        }
```

Creamos emails/emailVerificationEmail.blade.php

```
    <h1>Email Verification Mail</h1>

    Please verify your email with bellow link:
    <a href="{{ route('user.verify', $token) }}">Verify Email</a>
```

creamos middleware

```
    php artisan make:middleware IsVerifyEmail
```

```
    <?php

    namespace App\Http\Middleware;

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

    class IsVerifyEmail
    {
        /**
         * Handle an incoming request.
         *
         * @param  \Illuminate\Http\Request  $request
         * @param  \Closure  $next
         * @return mixed
         */
        public function handle(Request $request, Closure $next)
        {
            if (!Auth::user()->is_email_verified) {
                auth()->logout();
                return redirect()->route('login')
                        ->with('message', 'You need to confirm your account. We
have sent you an activation code, please check your email.');
            }

            return $next($request);
        }
    }
```

Kernel.php

```
    protected $routeMiddleware = [
        ....
        'is_verify_email' => \App\Http\Middleware\IsVerifyEmail::class,
    ];
```

.env

```
    MAIL_DRIVER=smtp
    MAIL_HOST=smtp-mail.outlook.com
    MAIL_PORT=587
    MAIL_USERNAME=example@hotmail.com
    MAIL_PASSWORD=123456789
    MAIL_ENCRYPTION=tls
    MAIL_FROM_ADDRESS=example@hotmail.com
```