



ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS



DESARROLLO DE APLICACIONES WEB (TDSD414)

ASIGNATURA:

Desarrollo de Aplicaciones Web

PROFESOR:

Ing. Ivonne Maldonado MSc.

PERÍODO ACADÉMICO:

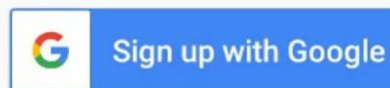
2022-B

Prueba 2

TÍTULO:

Parte Práctica

Laravel Socialite Google Gmail Account Login Example



Integrantes:

- Carlos Jurado
- Omar Guachamin
- Ariel Moposita
- Paulina Males
- Marcos Moreira

PROPOSITO DE LA PRÁCTICA

- Crear un proyecto web con un inicio del sistema por medio de redes sociales.

OBJETIVO GENERAL

- Cumplir con el Sprint 1.

OBJETIVOS ESPECÍFICOS

- Establecer que el proyecto administre notificaciones.
- Implementar que el inicio al sistema web sea por medio de redes sociales.

DESARROLLO

- Credenciales con los cuales nos vamos a poder loguear al inicio d sesión en Google y nos redireccionará a la pagina inicial.

```
39
40     'google' => [
41         'client_id' => env('GOOGLE_CLIENT_ID'),
42         'client_secret' => env('GOOGLE_CLIENT_SECRET'),
43         'redirect' => 'http://127.0.0.1:8000/auth/google/call-back',
44     ],
45 ];
46
```

- Definimos las rutas.

```
Route::get('auth/google', [GoogleAuthController::class,'redirect'])->name('google-auth');
Route::get('auth/google/call-back', [GoogleAuthController::class,'callbackGoogle']);
```

- Configuramos el controlador del inicio de sesión con Google.

```
GoogleAuthController.php X
app > Http > Controllers > GoogleAuthController.php > ...
12 public function redirect(){
13     return Socialite::driver('google')->redirect();
14 }
15
16 public function callbackGoogle(){
17     try{
18         $google_user = Socialite::driver('google')->user();
19
20         $user = User::where('google_id',$google_user->getId())->first();
21
22         if(!$user){
23             $new_user = User::create([
24                 'name' => $google_user->getName(),
25                 'email' => $google_user->getEmail(),
26                 'google_id' => $google_user->getId()
27             ]);
28
29             Auth::login($new_user);
30             return redirect('/home');
31             return redirect()->intended('home');
32         }else{
33             Auth::login($user);
34             // return redirect('/home');
35             return redirect()->intended('home');
36         }
37     }
38
39     }catch(\Throwable $th){
40         dd("Something went wrong!",$th->getMessage());
41     }
42 }
43 }
```

- Configuramos las funciones del CRUD del formulario.

```

app > Http > Controllers > ContactController.php > ...
14  * Create a new controller instance.
15  *
16  * @return void
17  */
18  public function __construct()
19  {
20      $this->middleware('auth');
21  }
22
23  /**
24   * Show the application dashboard.
25   *
26   * @return \Illuminate\Contracts\Support\Renderable
27   */
28  public function index()
29  {
30      $users = User::where('id','!=', auth()->id()->get());
31      return view('contact', compact('users'));
32  }
33
34  public function store(Request $request)
35  {
36      // return $request->all();
37      $message = Message::create([
38          'sender_id' => auth()->id(),
39          'recipient_id' => $request->recipient_id,
40          'body' => $request->body_message,
41      ]);
42
43      // auth()->user()->notify(new MessageNotification($message));
44      $user = User::where('id','!=', $message->recipient_id)->get()
45              ->each(function(User $user) use ($message){
46                  $user->notify(new MessageNotification($message));
47              });
48  }

```

- Creamos los campos de la tabla user en las migraciones.

```

/**
 * Run the migrations.
 *
 * @return void
 */
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')->nullable();
        $table->string('google_id')->nullable();
        $table->rememberToken();
        $table->timestamps();
    });
}

```

- Creamos los campos de la tabla notifications en las migraciones.

```

1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       *
12       * @return void
13       */
14       public function up()
15       {
16           Schema::create('notifications', function (Blueprint $table) {
17               $table->uuid('id')->primary();
18               $table->string('type');
19               $table->morphs('notifiable');
20               $table->text('data');
21               $table->timestamp('read_at')->nullable();
22               $table->timestamps();
23           });
24       }
25
26       /**
27        * Reverse the migrations.
28        *
29        * @return void
30        */
31       public function down()
32       {

```

- Creamos los campos de la tabla messages en las migraciones.

```
<?php

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

return new class extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('messages', function (Blueprint $table) {
            $table->id();
            $table->unsignedInteger('sender_id');
            $table->unsignedInteger('recipient_id');
            $table->text('body');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('messages');
    }
}
```

PRESENTACIÓN

Al finalizar tu tarea deberás subir:

- Un archivo en formato pdf con el nombre (Prueba2PartePráctica_ AWeb_2022B _NApellido). Napellido del estudiante que vaya a subir el documento. En el archivo deben estar todos los nombres del grupo de trabajo. El documento debe contar con la investigación realizada, algo sencillo, recuerde no tiene sentido que coloque un copy/paste. Además debe incluir la explicación de la implementación realizada a modo de informe. No olvide incluir el link de GitHub donde se encuentra su código. Fecha de presentación: 14 de febrero del año en curso.

RECURSOS NECESARIOS

- Visual Studio Code
- Herramienta ofimática
- Internet
- Material de clase

CONCLUSIONES

- Laravel cuenta con una serie de herramientas y paquetes que facilitan la integración de redes sociales, lo que puede reducir el tiempo y esfuerzo necesario para implementar esta funcionalidad.

- Es importante considerar la seguridad y privacidad de los usuarios al implementar la integración de redes sociales, asegurándose de cumplir con las políticas de privacidad y protección de datos.
- La integración de redes sociales puede ofrecer una forma rápida y conveniente para que los usuarios inicien sesión en el sitio web, lo que puede aumentar la tasa de conversión y mejorar la experiencia del usuario.

BIBLIOGRAFIA

N/A