

Projecte GLOBAL: VideosApp

4t SPRINT



Professor: Jordi Vega

Assignatures: M7, M8 i M9

Nom: Marc

Cognoms: Pallares Pino



INDEX

Link github:
4t Sprint

2
3

Link github:

<https://github.com/Mpallares1/laravelmarc>


INDEX

4t Sprint

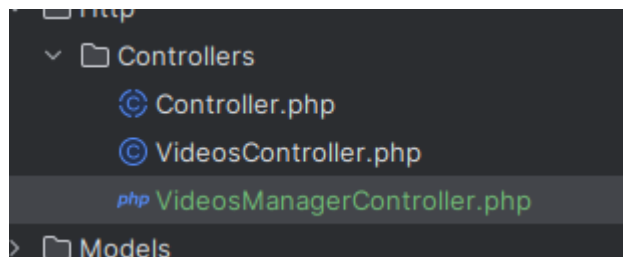
- Corregir els errors del 3r sprint. En cas que als tests no s'hagi comprovat que els usuaris en permisos puguin accedir a la ruta /videosmanage, modificar-ho.

L'error indica que el mètode createFirstVideo torna null en lloc d'una instància d'App\Models\Video. Això pot passar si el mètode create falla en inserir el registre a la base de dades.

```
© Video.php  © DatabaseSeeder.php x  © DefaultVideos.php
1  <?php
2  namespace Database\Seeders;
3  use <...>
10
11 no usages ± «Marc» +1 *
12 class DatabaseSeeder extends Seeder
13 {
14     /**
15      * Seed the application's database.
16      */
17     ± «Marc» +1 *
18     public function run(): void
19     {
20         User::truncate();
21         Video::truncate();
22         Role::truncate();
23         Permission::truncate();
24
25         // Crear permisos
26         CreateUsers::create_role_and_permissions();
27
28         // Crear el rol 'Regular User' si no existe
29         if (!Role::where('name', 'Regular User')->exists()) {
30             Role::create(['name' => 'Regular User', 'guard_name' => 'web']);
31         }
32
33         // Crear usuarios por defecto
34         $regularUser = CreateUsers::createRegularUser();
35         $videoManagerUser = CreateUsers::createVideoManagerUser();
36         $superAdminUser = CreateUsers::createSuperAdminUser();
37
38         // Asignar roles a los usuarios
39         $regularUser->assignRole('Regular User');
40         $videoManagerUser->assignRole('video_manager');
41         $superAdminUser->assignRole('super_admin');
42
43         // Crear un video por defecto
44         DefaultVideos::createFirstVideo();
45     }
46 }
```

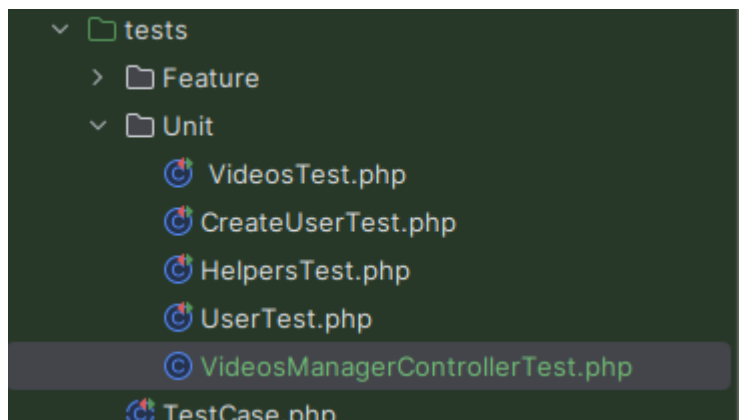
 INSTITUT DE L'EBRE TORTOSA	GLOBAL M7, M8 i M9	DAM 2n
--	---------------------------	-------------------------

- Crear videosManageController en les funcions testedby, index, store, show, edit, update, delete i destroy.

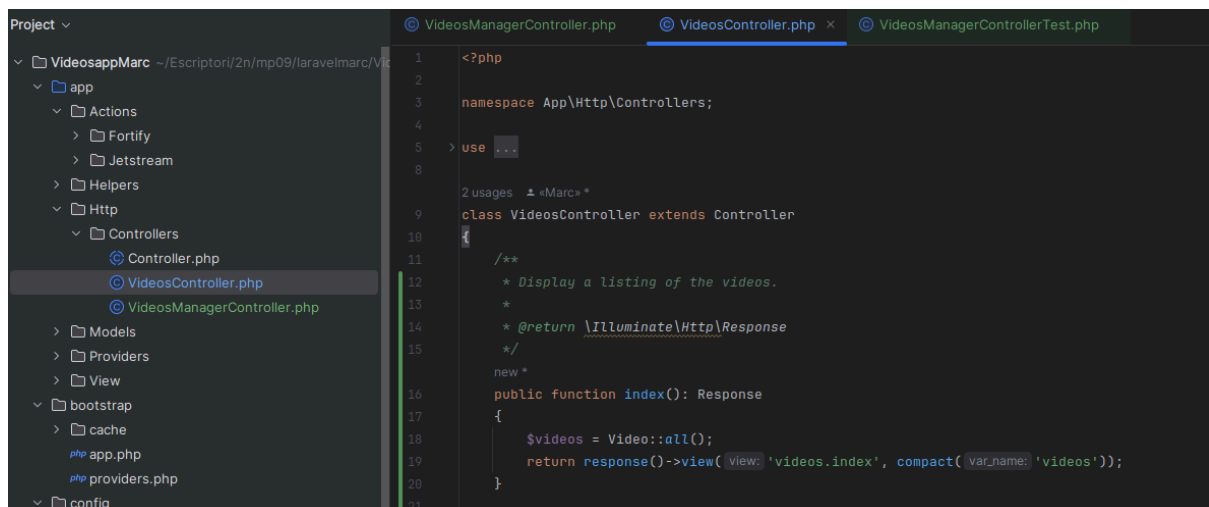


php VideosManagerController.php x

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\Video;
6  use Illuminate\Http\Request;
7
8  no usages new *
9  class VideosManageController extends Controller
10 {
11     /**
12      * Retorna la classe del test.
13      *
14      * @return string
15      */
16     nousages new *
17     public function testedBy()
18     {
19         return \Tests\Unit\VideosManageControllerTest::class;
20     }
21
22     /**
23      * Mostra una llista de vídeos.
24      *
25      * @return \Illuminate\Http\Response
26      */
27     new *
28     public function index()
29     {
30         $videos = Video::all();
31         return view('videos.index', compact('videos'));
32     }
33
34     /**
35      * Emmagatzema un vídeo nou.
36      *
37      * @param \Illuminate\Http\Request $request
38      * @return \Illuminate\Http\Response
39      */
40     new *
41     public function store(Request $request)
```



- Crear la funció index a videosController.



- Revisar que tinguessiu 3 videos creats a helpers i afegits al databaseSeeder

helpers:



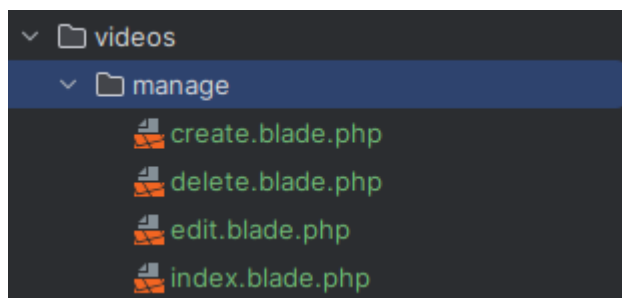
```
© DefaultVideos.php × © VideoSeeder.php
8      class DefaultVideos
39      public static function createSecondDefaultVideo(array $attributes = [])
40      {
41          $defaultAttributes = [
42              'title' => 'Florian',
43              'description' => 'Second Default Description',
44              'url' => 'https://www.youtube.com/embed/k8BMwR704jE',
45              'published_at' => Carbon::now(),
46              'previous' => null,
47              'next' => null,
48              'series_id' => 2,
49          ];
50
51          $attributes = array_merge($defaultAttributes, $attributes);
52
53          return Video::create($attributes);
54      }
55
56      /**
57       * Crea el tercer video por defecto.
58       *
59       * @param array $attributes
60       * @return \App\Models\Video
61       */
62      public static function createThirdDefaultVideo(array $attributes = [])
63      {
64          $defaultAttributes = [
65              'title' => 'Videos Random',
66              'description' => 'Third Default Description',
67              'url' => 'https://www.youtube.com/embed/EXeN6v-7aT8',
68              'published_at' => Carbon::now(),
69              'previous' => null,
70              'next' => null,
71              'series_id' => 3,
72          ];
73
74          $attributes = array_merge($defaultAttributes, $attributes);
75
76          return Video::create($attributes);
77      }
78  }
```

\App\Helpers > DefaultVideos > createThirdDefaultVideo()

database

```
© DefaultVideos.php  © VideoSeeder.php x
1  <?php
2
3  namespace Database\Seeders;
4
5  > use ...
6
7
8  no usages  «Marc» *
9  class VideoSeeder extends Seeder
10 {
11     «Marc» *
12     public function run()
13     {
14         DefaultVideos::createDefaultVideo();
15         DefaultVideos::createSecondDefaultVideo();
16         DefaultVideos::createThirdDefaultVideo();
17     }
18 }
```

- Crear les vistes per al CRUD que només poden veure-ho els que tinguin els permisos adients: `resources/views/videos/manage/index.blade.php`, `resources/views/videos/manage/create.blade.php`, `resources/views/videos/manage/edit.blade.php`, `resources/views/videos/manage/delete.blade.php`.



create

```
© DefaultVideos.php create.blade.php x
1 @extends('layouts.app')
2
3 @section('content')
4     <div class="container">
5         @can('create videos')
6             <h1>Create Video</h1>
7             <!-- Form to create a video -->
8         @else
9             <p>You do not have permission to create videos.</p>
10        @endcan
11    </div>
12 @endsection
```

delete

```
© DefaultVideos.php delete.blade.php x
1 @extends('layouts.app')
2
3 @section('content')
4     <div class="container">
5         @can('delete videos')
6             <h1>Delete Video</h1>
7             <!-- Form to delete a video -->
8         @else
9             <p>You do not have permission to delete videos.</p>
10        @endcan
11    </div>
12 @endsection
```

edit

```
© DefaultVideos.php  edit.blade.php x
1  @extends('layouts.app')
2
3  @section('content')
4      <div class="container">
5          @can('edit videos')
6              <h1>Edit Video</h1>
7              <!-- Form to edit a video -->
8          @else
9              <p>You do not have permission to edit videos.</p>
10         @endcan
11     </div>
12 @endsection
```

index

```
© DefaultVideos.php  index.blade.php x
1  @extends('layouts.app')
2
3  @section('content')
4      <div class="container">
5          @can('view videos')
6              <h1>Manage Videos</h1>
7              <!-- List videos here -->
8          @else
9              <p>You do not have permission to view videos.</p>
10         @endcan
11     </div>
12 @endsection
```

- A la vista index.blade.php, afegir la taula del CRUD de vídeos

```
index.blade.php x
1  @extends('layouts.app')
2
3  @section('content')
4      <div class="container">
5          @can('view videos')
6              <h1>Manage Videos</h1>
7              <table class="table">
8                  <thead>
9                      <tr>
10                         <th>Title</th>
11                         <th>Description</th>
12                         <th>URL</th>
13                         <th>Published At</th>
14                         <th>Actions</th>
15                     </tr>
16                 </thead>
17                 <tbody>
18                     @foreach($videos as $video)
19                         <tr>
20                             <td>{{ $video->title }}</td>
21                             <td>{{ $video->description }}</td>
22                             <td><a href="{{ $video->url }}" target="_blank">View</a></td>
23                             <td>{{ $video->published_at }}</td>
24                             <td>
25                                 @can('edit videos')
26                                     <a href="{{ route('videos.edit', $video->id) }}" class="btn btn-primary">Edit</a>
27                                 @endcan
28                                 @can('delete videos')
29                                     <form action="{{ route('videos.destroy', $video->id) }}" method="POST" style="display:inline;">
30                                         @csrf
31                                         @method('DELETE')
32                                         <button type="submit" class="btn btn-danger">Delete</button>
33                                     </form>
34                                 @endcan
35                             </td>
36                         </tr>
37                     @endforeach
38                 </tbody>
39             </table>
40         @else
41             <p>You do not have permission to view videos.</p>
42         @endcan
43     </div>
44 @endsection
```

- A la vista create.blade.php, afegir el formulari per posar els vídeos, s'ha d'utilitzar l'atribut data-qa per a que sigui més fàcil identificar per als tests.

```
index.blade.php create.blade.php x
1 @extends('layouts.app')
2
3 @section('content')
4     <div class="container">
5         @can('create videos')
6             <h1>Create Video</h1>
7             <form action="{{ route('videos.store') }}" method="POST">
8                 @csrf
9                 <div class="form-group">
10                     <label for="title">Title</label>
11                     <input type="text" class="form-control" id="title" name="title" data-qa="video-title" required>
12                 </div>
13                 <div class="form-group">
14                     <label for="description">Description</label>
15                     <textarea class="form-control" id="description" name="description" data-qa="video-description">
16                 </div>
17                 <div class="form-group">
18                     <label for="url">URL</label>
19                     <input type="url" class="form-control" id="url" name="url" data-qa="video-url" required>
20                 </div>
21                 <div class="form-group">
22                     <label for="published_at">Published At</label>
23                     <input type="datetime-local" class="form-control" id="published_at" name="published_at" data-c
24                 </div>
25                 <button type="submit" class="btn btn-primary" data-qa="video-submit">Submit</button>
26             </form>
27         @else
28             <p>You do not have permission to create videos.</p>
29         @endcan
30     </div>
31 @endsection
```

- A la vista edit.blade.php, afegir la taula del CRUD de vídeos.

```
edit.blade.php x
1 @extends('layouts.app')
2
3 @section('content')
4     <div class="container">
5         @can('edit videos')
6             <h1>Edit Video</h1>
7             <form action="{{ route('videos.update', $video->id) }}" method="POST">
8                 @csrf
9                 @method('PUT')
10                <div class="form-group">
11                    <label for="title">Title</label>
12                    <input type="text" class="form-control" id="title" name="title" value="{{ $video->title }}" />
13                </div>
14                <div class="form-group">
15                    <label for="description">Description</label>
16                    <textarea class="form-control" id="description" name="description" data-qa="video-description">{{ $video->description }}</textarea>
17                </div>
18                <div class="form-group">
19                    <label for="url">URL</label>
20                    <input type="url" class="form-control" id="url" name="url" value="{{ $video->url }}" data-qa="video-url">
21                </div>
22                <div class="form-group">
23                    <label for="published_at">Published At</label>
24                    <input type="datetime-local" class="form-control" id="published_at" name="published_at" value="{{ $video->published_at }}" />
25                </div>
26                <button type="submit" class="btn btn-primary" data-qa="video-submit">Update</button>
27            </form>
28
29            <h2 class="mt-5">Manage Videos</h2>
30            <table class="table">
31                <thead>
32                    <tr>
33                        <th>Title</th>
34                        <th>Description</th>
35                        <th>URL</th>
36                        <th>Published At</th>
37                        <th>Actions</th>
38                    </tr>
39                </thead>
40            </table>
41        @endcan
42    </div>
43 @endsection
```

- A la vista delete.blade.php, afegir la confirmació de l'eliminació del vídeo.

```
edit.blade.php delete.blade.php x
1 @extends('layouts.app')
2
3 @section('content')
4     <div class="container">
5         @can('delete videos')
6             <h1>Delete Video</h1>
7             <form action="{{ route('videos.destroy', $video->id) }}" method="POST" onsubmit="return confirm('Are
8                 @csrf
9                 @method('DELETE')
10                <button type="submit" class="btn btn-danger">Delete</button>
11            </form>
12        @else
13            <p>You do not have permission to delete videos.</p>
14        @endcan
15    </div>
16 @endsection
```

- Crear la vista de resources/views/videos/index.blade.php on es vegin tots els vídeos tipus la pàgina principal de youtube i al clicar que porti al detall del vídeo (el show del sprint anterior).

```
index.blade.php x
1 @extends('layouts.app')
2
3 @section('content')
4     <div class="container">
5         @can('view videos')
6             <h1>Manage Videos</h1>
7             <table class="table">
8                 <thead>
9                     <tr>
10                        <th>Title</th>
11                        <th>Description</th>
12                        <th>URL</th>
13                        <th>Published At</th>
14                        <th>Actions</th>
15                    </tr>
16                </thead>
17                <tbody>
18                    @foreach($videos as $video)
19                        <tr>
20                            <td>{{ $video->title }}</td>
21                            <td>{{ $video->description }}</td>
22                            <td><a href="{{ $video->url }}" target="_blank">View</a></td>
23                            <td>{{ $video->published_at }}</td>
24                            <td>
25                                @can('edit videos')
26                                    <a href="{{ route('videos.edit', $video->id) }}" class="btn btn-primary">Edit</a>
27                                @endcan
28                                @can('delete videos')
29                                    <form action="{{ route('videos.destroy', $video->id) }}" method="POST" style="display: inline-block;">
30                                        @csrf
31                                        @method('DELETE')
32                                        <button type="submit" class="btn btn-danger">Delete</button>
33                                    </form>
34                                @endcan
35                            </td>
36                        </tr>
37                    @endforeach
38                </tbody>
39            </table>
40
41            <h1 class="mt-5">All Videos</h1>
42        </div>
43    @endsection
```

- Modificar el test `user_with_permissions_can_manage_videos()` per a que hi hagi 3 videos.

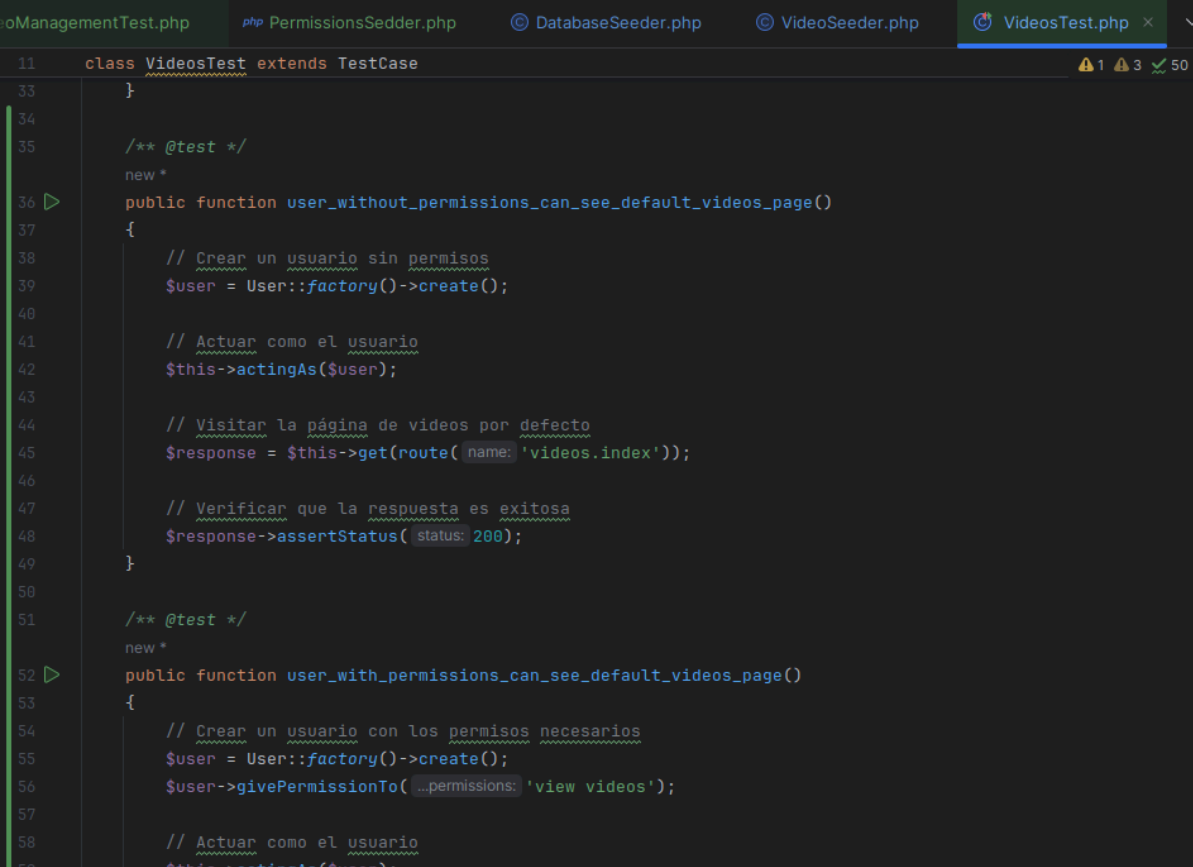
```
php VideoManagementTest.php x
1 public function test_user_with_permissions_can_manage_videos()
2 {
3     $user = User::factory()->create();
4     $user->givePermissionTo('view videos', 'create videos', 'edit videos', 'delete videos');
5
6     $videos = Video::factory()->count(3)->create();
7
8     $this->actingAs($user);
9
10    // Visit the manage videos page
11    $response = $this->get(route('videos.index'));
12
13    // Assert the response is successful
14    $response->assertStatus(200);
15
16    // Assert the presence of the 3 videos
17    foreach ($videos as $video) {
18        $response->assertSee($video->title);
19        $response->assertSee($video->description);
20    }
21 }
```

- A helpers crear els permisos de videos per al crud i assignar-los als usuaris corresponents.


```
php VideoManagementTest.php  php PermissionsSeeder.php x  DatabaseSeeder.php
1  <?php
2
3  use Illuminate\Database\Seeder;
4  use Spatie\Permission\Models\Permission;
5
6  class PermissionsSeeder extends Seeder
7  {
8      public function run()
9      {
10         // Definir permisos
11         $permissions = [
12             'view videos',
13             'create videos',
14             'edit videos',
15             'delete videos',
16         ];
17
18         // Crear permisos
19         foreach ($permissions as $permission) {
20             Permission::firstOrCreate(['name' => $permission]);
21         }
22     }
23 }
24
```

```
php VideoManagementTest.php  php PermissionsSeeder.php  © DatabaseSeeder.php x
12  class DatabaseSeeder extends Seeder
16      */
    ⤴ «Marc» +1 *
17  public function run(): void
18  {
19      User::truncate();
20      Video::truncate();
21
22      // Crear permisos
23      CreateUsers::createPermissions();
24
25      // Crear usuarios por defecto
26      $regularUser = CreateUsers::createRegularUser();
27      $videoManagerUser = CreateUsers::createVideoManagerUser();
28      $superAdminUser = CreateUsers::createSuperAdminUser();
29
30      // Asignar roles a los usuarios
31      $regularUser->assignRole('Regular User');
32      $videoManagerUser->assignRole('Video Manager');
33      $superAdminUser->assignRole('Super Admin');
34
35      $this->call(class: PermissionsSeeder::class);
36
37      // Crear un video por defecto
38      DefaultVideos::createDefaultVideo();
39  }
```

- A VideoTest crear les funcions:
user_without_permissions_can_see_default_videos_page,
user_with_permissions_can_see_default_videos_page,
not_logged_users_can_see_default_videos_page



```
11 class VideosTest extends TestCase
33     }
34
35     /** @test */
36     new *
37     public function user_without_permissions_can_see_default_videos_page()
38     {
39         // Crear un usuario sin permisos
40         $user = User::factory()->create();
41
42         // Actuar como el usuario
43         $this->actingAs($user);
44
45         // Visitar la página de videos por defecto
46         $response = $this->get(route('name: 'videos.index'));
47
48         // Verificar que la respuesta es exitosa
49         $response->assertStatus(status: 200);
50     }
51
52     /** @test */
53     new *
54     public function user_with_permissions_can_see_default_videos_page()
55     {
56         // Crear un usuario con los permisos necesarios
57         $user = User::factory()->create();
58         $user->givePermissionTo(permissions: 'view videos');
59
60         // Actuar como el usuario
61         $this->actingAs($user);
```

- A VideosManageControllerTest crear les funcions:
loginAsVideoManager,
loginAsSuperAdmin,
loginAsRegularUser,
user_with_permissions_can_see_add_videos,
user_without_videos_manage_create_cannot_see_add_videos,
user_with_permissions_can_store_videos,
user_without_permissions_cannot_store_videos,
user_with_permissions_can_destroy_videos,
user_without_permissions_cannot_destroy_videos,
user_with_permissions_can_see_edit_videos,
user_without_permissions_cannot_see_edit_videos,
user_with_permissions_can_update_videos,
user_without_permissions_cannot_update_videos
user_with_permissions_can_manage_videos,
regular_users_cannot_manage_videos,
guest_users_cannot_manage_videos,
superadmins_can_manage_videos

```

php VideoManagementTest.php  VideosManageControllerTest.php  php PermissionsSedder.php  DatabaseSeeder.php
10  class VideosManageControllerTest extends TestCase
140  {
141
142      /** @test */
143      new *
144      public function user_with_permissions_can_update_videos()
145      {
146          $this->loginAsVideoManager();
147          $video = Video::factory()->create();
148          $response = $this->put(route('name: videos.update', $video), [
149              'title' => 'Updated Title',
150              'description' => 'Updated Description',
151          ]);
152          $response->assertStatus(status: 302);
153      }
154
155      /** @test */
156      new *
157      public function user_without_permissions_cannot_update_videos()
158      {
159          $this->loginAsRegularUser();
160          $video = Video::factory()->create();
161          $response = $this->put(route('name: videos.update', $video), [
162              'title' => 'Updated Title',
163              'description' => 'Updated Description',
164          ]);
165          $response->assertStatus(status: 403);
166      }
167  }

```

- Crear les rutes de videos/manage per al CRUD de videos amb el seu middleware correspondent i la ruta del index de videos. Les rutes del CRUD han d'aparèixer només quan estàs logejat i la d'index tant si estas logejat com si no.

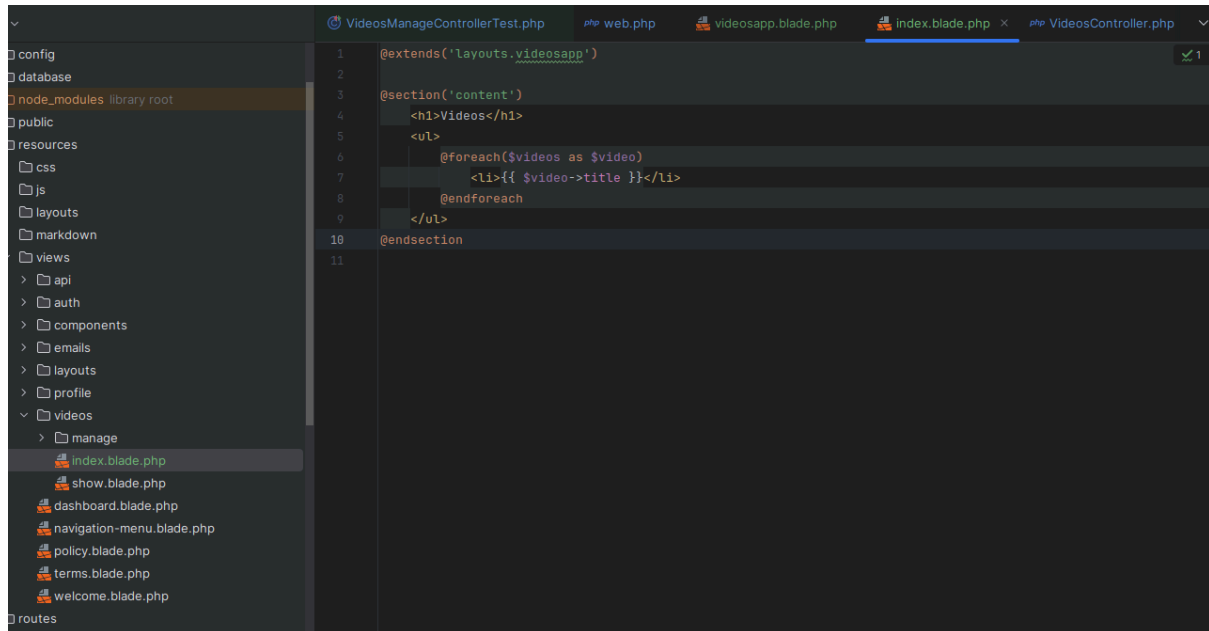
```
VideosManageControllerTest.php  php web.php  php VideosController.php x  php PermissionsSedder.php  DatabaseSeed v
9  class VideoController extends Controller
10
11  new *
12  public function index()
13  {
14      $videos = Video::all();
15      return view('videos.index', compact('videos'));
16  }
17
18  new *
19  public function create()
20  {
21      return view('videos.create');
22  }
23
24  new *
25  public function store(Request $request)
26  {
27      $video = Video::create($request->all());
28      return redirect()->route('videos.index');
29  }
30
31  new *
32  public function edit(Video $video)
33  {
34      return view('videos.edit', compact('video'));
35  }
36
37  new *
38  public function update(Request $request, Video $video)
```

```

1  <?php
2
3  > use ...
4
5
6  // Ruta de l'index de videos (accessible tant si estàs logejat com si no)
7  Route::get(uri: '/videos', [VideosController::class, 'index'])->name(name: 'videos.index');
8
9  // Ruta per mostrar un video especific
10 Route::get(uri: '/videos/{id}', [VideosController::class, 'show'])->name(name: 'videos.show');
11
12 // Ruta de benvinguda
13 Route::get(uri: '/', function () {
14     return view(view: 'welcome');
15 });
16
17 // Rutes del CRUD de videos (només accessibles quan estàs logejat)
18 Route::middleware(['auth:sanctum', config(key: 'jetstream.auth_session'), 'verified'])->group(function () {
19     Route::get(uri: '/dashboard', function () {
20         return view(view: 'dashboard');
21     })->name(name: 'dashboard');
22
23     Route::get(uri: '/videos/create', [VideosController::class, 'create'])->name(name: 'videos.create');
24     Route::post(uri: '/videos', [VideosController::class, 'store'])->name(name: 'videos.store');
25     Route::get(uri: '/videos/{video}/edit', [VideosController::class, 'edit'])->name(name: 'videos.edit');
26     Route::put(uri: '/videos/{video}', [VideosController::class, 'update'])->name(name: 'videos.update');
27     Route::delete(uri: '/videos/{video}', [VideosController::class, 'destroy'])->name(name: 'videos.destroy');
28 });
29

```

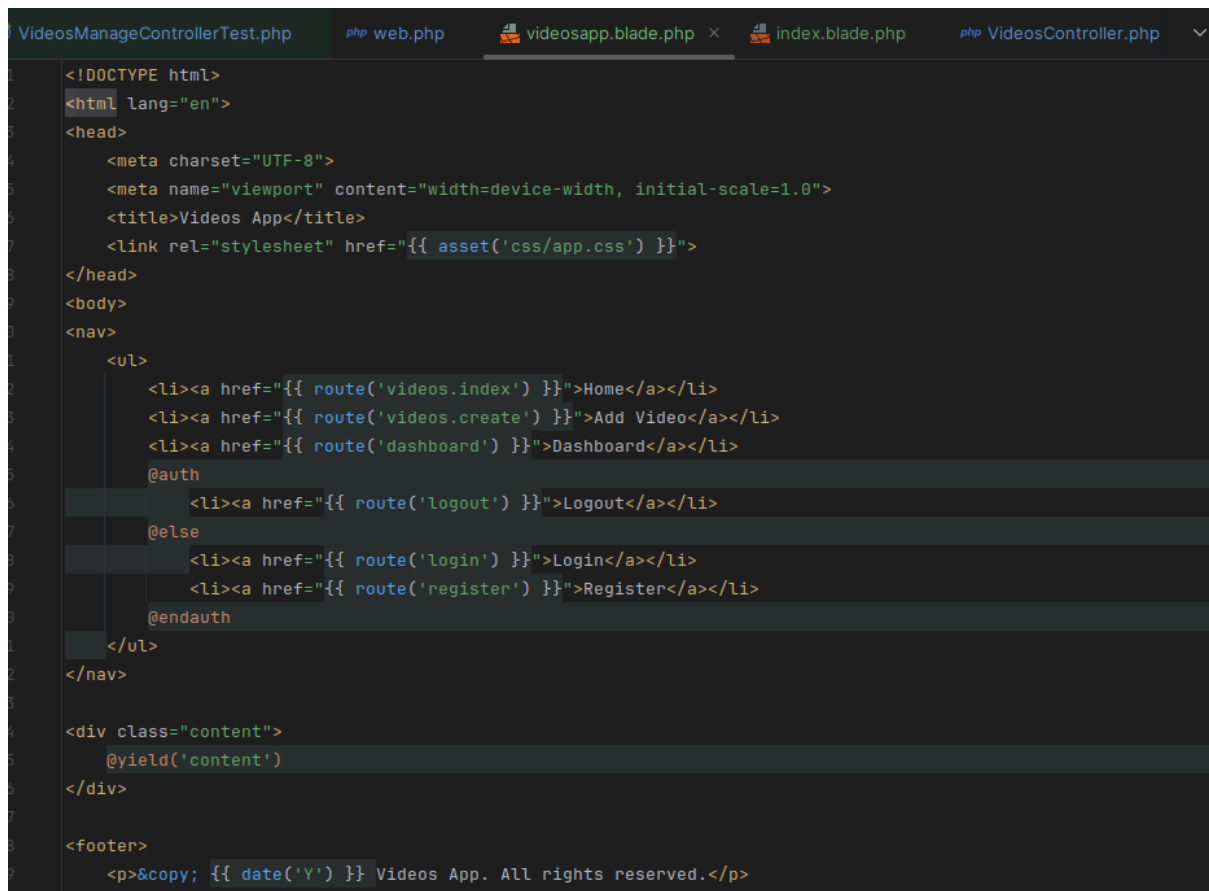
- Afegir navbar i footer a la plantilla resources/layouts/videosapp. S'ha de poder navegar entre pàgines.



```

1 @extends('layouts.videosapp')
2
3 @section('content')
4     <h1>Videos</h1>
5     <ul>
6         @foreach($videos as $video)
7             <li>{{ $video->title }}</li>
8         @endforeach
9     </ul>
10 @endsection
11

```

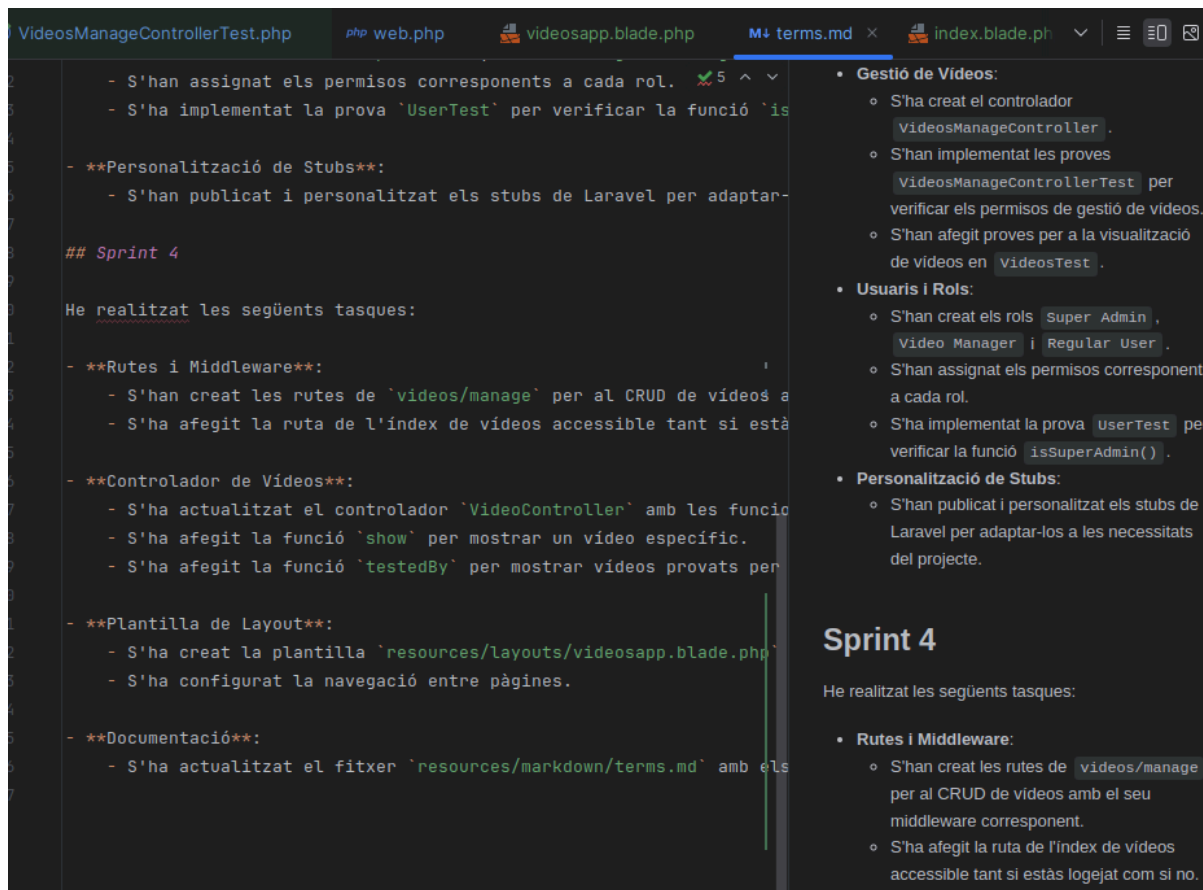


```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Videos App</title>
7     <link rel="stylesheet" href="{{ asset('css/app.css') }}">
8 </head>
9 <body>
10 <nav>
11     <ul>
12         <li><a href="{{ route('videos.index') }}">Home</a></li>
13         <li><a href="{{ route('videos.create') }}">Add Video</a></li>
14         <li><a href="{{ route('dashboard') }}">Dashboard</a></li>
15         @auth
16             <li><a href="{{ route('logout') }}">Logout</a></li>
17         @else
18             <li><a href="{{ route('login') }}">Login</a></li>
19             <li><a href="{{ route('register') }}">Register</a></li>
20         @endauth
21     </ul>
22 </nav>
23
24 <div class="content">
25     @yield('content')
26 </div>
27
28 <footer>
29     <p>&copy; {{ date('Y') }} Videos App. All rights reserved.</p>
30

```

- Afegir a resources/markdown/terms el que heu fet al sprint.



Sprint 4

He realitzat les següents tasques:

- S'han assignat els permisos corresponents a cada rol.
- S'ha implementat la prova `UserTest` per verificar la funció `isSuperAdmin()`.

****Personalització de Stubs**:**

- S'han publicat i personalitzat els stubs de Laravel per adaptar-los a les necessitats del projecte.

****Rutes i Middleware**:**

- S'han creat les rutes de `videos/manage` per al CRUD de vídeos amb el seu middleware corresponent.
- S'ha afegit la ruta de l'índex de vídeos accessible tant si estàs logejat com si no.

****Controlador de Vídeos**:**

- S'ha actualitzat el controlador `VideoController` amb les funcions `show` i `testedBy`.

****Plantilla de Layout**:**

- S'ha creat la plantilla `resources/layouts/videosapp.blade.php`.
- S'ha configurat la navegació entre pàgines.

****Documentació**:**

- S'ha actualitzat el fitxer `resources/markdown/terms.md` amb els canvis realitzats.

Gestió de Vídeos:

- S'ha creat el controlador `VideosManageController`.
- S'han implementat les proves `VideosManageControllerTest` per verificar els permisos de gestió de vídeos.
- S'han afegit proves per a la visualització de vídeos en `VideosTest`.

Usuaris i Rols:


- S'han creat els rols `Super Admin`, `Video Manager` i `Regular User`.
- S'han assignat els permisos corresponents a cada rol.
- S'ha implementat la prova `UserTest` per verificar la funció `isSuperAdmin()`.

Personalització de Stubs:

- S'han publicat i personalitzat els stubs de Laravel per adaptar-los a les necessitats del projecte.

- Comprovar en Larastan tots els fitxers que heu creat.



 INSTITUT DE L'EBRE TORTOSA	GLOBAL M7, M8 i M9	DAM 2n
--	---------------------------	-------------------------