

# LFTS - Admin Section

## 62.Limit Access to Only Admins

Ahora vamos a separar los usuarios por su rol, y de esta manera crear un usuario administrador con su respectiva pagina administradora

```
23 Route::post('/logout', [SessionsController::class, 'destroy'])->middleware('auth');
24
25 Route::get('/admin/posts/create', [PostController::class, 'create'])->middleware('admin');
```

resources > views > posts > create.blade.php

```
1 <x-layout>
2
3     <section class="px-6 py-8">
4
5         Hello
6
7     </section>
8
9 </x-layout>
```

```
58 'auth' => \App\Http\Middleware\Authenticate::class,
59 'admin' => MustBeAdmin::class,
60 'auth.basic' => \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,
```

```
1 <?php
2
3 namespace App\Http\Middleware;
4
5 use Closure;
6 use Illuminate\Http\Request;
7 use Illuminate\Http\Response;
8
9 class MustBeAdmin
10 {
11     /**
12      * Handle an incoming request.
13      *
14      * @param \Illuminate\Http\Request $request
15      * @param \Closure(\Illuminate\Http\Request): (\Illuminate\Http\Response|\Illuminate\Http\RedirectResponse) $next
16      * @return \Illuminate\Http\Response|\Illuminate\Http\RedirectResponse
17      */
18     public function handle(Request $request, Closure $next)
19     {
20
21         if(auth()->guest() || auth()->user()->username !== 'CrisVM') {
22             abort(Response::HTTP_FORBIDDEN);
23         }
24
25         return $next($request);
26     }
27 }
```

```

    public function create()
    {
        return view('posts.create');
    }
}

```

## 63. Create the Publish Post Form

Ahora vamos a crear una forma para que el administrador pueda crear post

```

24
25 Route::get('/admin/posts/create', [PostController::class, 'create'])->middleware('admin');
26 Route::post('/admin/posts', [PostController::class, 'store'])->middleware('admin');

```

```

4
5 <x-panel class="max-w-sm mx-auto">
6     <form action="/admin/posts" method="POST">
7         @csrf
8         <div class="mb-6">
9             <label for="title" class="block mb-2 uppercase font-bold text-xs text-gray-700">Title</label>
10            <input type="text" name="title" id="title" class="border border-gray-400 p-2 w-full" value="{{old('title')}}" required>
11            @error('title')
12                <p class="text-red-500 text-xs mt-2">{{ $message }}</p>
13            @enderror
14        </div>
15        <div class="mb-6">
16            <label for="slug" class="block mb-2 uppercase font-bold text-xs text-gray-700">Slug</label>
17            <input type="text" name="slug" id="slug" class="border border-gray-400 p-2 w-full" value="{{old('slug')}}" required>
18            @error('slug')
19                <p class="text-red-500 text-xs mt-2">{{ $message }}</p>
20            @enderror
21        </div>
22        <div class="mb-6">
23            <label for="excerpt" class="block mb-2 uppercase font-bold text-xs text-gray-700">Excerpt</label>
24            <textarea name="excerpt" id="excerpt" rows="3" class="border border-gray-400 p-2 w-full" value="{{old('excerpt')}}" placeholder="" required></textarea>
25            @error('excerpt')
26                <span class="text-xs text-red-500 mt-2">{{ $message }}</span>
27            @enderror
28        </div>
29        <div class="mb-6">
30            <label for="body" class="block mb-2 uppercase font-bold text-xs text-gray-700">Body</label>
31            <textarea name="body" id="body" rows="5" class="border border-gray-400 p-2 w-full" value="{{old('body')}}" placeholder="" required></textarea>
32            @error('body')
33                <span class="text-xs text-red-500 mt-2">{{ $message }}</span>
34            @enderror
35        </div>
36        <div class="mb-6">
37            <label for="category_id" class="block mb-2 uppercase font-bold text-xs text-gray-700">Category</label>
38            <select name="category_id" id="category_id">
39                @foreach (\App\Models\Category::all() as $category)
40                    <option value="{{ $category->id }}" {{ old('category_id') === $category->id ? 'selected' : '' }}>{{ ucwords($category->name) }}</option>
41                @endforeach
42            </select>
43            @error('category_id')
44                <span class="text-xs text-red-500 mt-2">{{ $message }}</span>
45            @enderror
46        </div>
47        <x-submit-button>Publish</x-submit-button>
48    </form>
49 </x-panel>
50
51 </section>
52
53 </x-layout>

```

```

32     }
33
34     public function store()
35     {
36         $attributes = request()->validate([
37             'title' => 'required',
38             'slug' => ['required', Rule::unique('posts', 'slug')],
39             'excerpt' => 'required',
40             'body' => 'required',
41             'category_id' => ['required', Rule::exists('categories', 'id')],
42         ]);
43
44         $attributes['user_id'] = auth()->id();
45
46         Post::create($attributes);
47
48         return redirect('/');
49     }
50 }

```

## 64. Validate and Store Post Thumbnails

En este episodio hacemos todos los pasos necesarios para que ahora los post puedan incluir una miniatura personalizada

```

6   <div class="col-span-4 lg:text-center lg:pt-14 mb-10">
7       
8   </div>

```

```

2   <div class="mb-6">
3       <label for="thumbnail" class="block mb-2 uppercase font-bold text-xs text-gray-700">Thumbnail
4       <input type="file" name="thumbnail" id="thumbnail" class="border border-gray-400 p-2 w-full"
5       @error('thumbnail')
6           <p class="text-red-500 text-xs mt-2">{{ $message }}</p>
7       @enderror
8   </div>
9   <div class="mb-6">
10      <label for="excerpt" class="block mb-2 uppercase font-bold text-xs text-gray-700">Excerpt</label>

```

```

5   <div>
6       

```

```

15
16     'default' => env('FILESYSTEM_DRIVER', 'public'),
17
18     /*

```

```

public function up()
{
    Schema::create('posts', function (Blueprint $table) {
        $table->id();
        $table->foreignId('user_id')->constrained()->cascadeOnDelete();
        $table->foreignId('category_id')->constrained()->cascadeOnDelete();
        $table->string('slug')->unique();
        $table->string('title');
        $table->string('thumbnail')->nullable();
        $table->text('excerpt');
        $table->text('body');
        $table->timestamps();
        $table->timestamp('published_at')->nullable();
    });
}

```

```

public function definition()
{
    return [
        'user_id' => User::factory(),
        'category_id' => Category::factory(),
        'title' => $this->faker->sentence,
        'slug' => $this->faker->slug,
        'thumbnail' => 'thumbnails/wallpaperflare.com_wallpaper (9).jpg',
        'excerpt' => '<p>' . implode('</p><p>', $this->faker->paragraphs(2)) . '</p>',
        'body' => '<p>' . implode('</p><p>', $this->faker->paragraphs(6)) . '</p>',
    ];
}

```

```


</div>

```

```

public function store()
{
    $attributes = request()->validate([
        'title' => 'required',
        'slug' => ['required', Rule::unique('posts', 'slug')],
        'thumbnail' => 'required|image',
        'excerpt' => 'required',
        'body' => 'required',
        'category_id' => ['required', Rule::exists('categories', 'id')],
    ]);

    $attributes['user_id'] = auth()->id();
    $attributes['thumbnail'] = request()->file('thumbnail')->store('public/thumbnails');
    $attributes['thumbnail'] = str_replace('public/', '', $attributes['thumbnail']);

    Post::create($attributes);

    return redirect('/');
}

```

## 65. Extract Form-Specific Blade Components

Reorganizamos la vista para hacerla mas modular, por decirlo de alguna manera

```
1  @props(['name'])
2
3  @error('{{ $name }}')
4  |   <p class="text-red-500 text-xs mt-2">{{ $message }}</p>
5  @enderror
```

```
1  @props(['name'])
2
3  @error('{{ $name }}')
4  |   <p class="text-red-500 text-xs mt-2">{{ $message }}</p>
5  @enderror
```

```
resources > views > components > form > label.blade.php
1  @props(['name'])
2  <label for="{{ $name }}" class="block mb-2 uppercase font-bold text-xs text-gray-700">{{ ucwords($name) }}</label>
```

```
resources > views > components > form > section.blade.php
1  <div class="mb-6">
2  |   {{ $slot }}
3  </div>
```

```
1  @props(['name', 'rows' => '3'])
2  <x-form.section>
3  |   <x-form.label name="{{ $name }}" />
4  |   <textarea name="{{ $name }}" id="{{ $name }}" rows="{{ $rows }}" class="border border-gray-400 p-2 w-full" value="{{ $value }}" />
5  |   <x-form.error name="{{ $name }}" />
6  </x-form.section>
```

```
5  <x-panel>
6  |   <form action="/admin/posts" method="POST" enctype="multipart/form-data">
7  |       @csrf
8  |       <x-form.input name="title" />
9  |       <x-form.input name="slug" />
10 |       <x-form.input name="thumbnail" type="file" />
11 |       <x-form.textarea name="excerpt" />
12 |       <x-form.textarea name="body" rows="5" />
13 |       <x-form.section>
14 |           <x-form.label name="category" />
15 |           <select name="category_id" id="category_id">
16 |               @foreach (\App\Models\Category::all() as $category)
17 |                   <option value="{{ $category->id }}" {{ old('category_id') === $category->id ? 'selected' : '' }}>{{ $category->name }}</option>
18 |               @endforeach
19 |           </select>
20 |           <x-form.error name="category_id" />
21 |       </x-form.section>
22 |   </div>
23 |   <x-submit-button>Publish</x-submit-button>
```

## 66. Extend the Admin Layout

Ahora vamos a modificar varias cosas para que la interfaz sea mas util y amigable con el cliente o administrador

```
3 <x-form.label name="{{{$name}}}" />
4 <input type="{{{$type}}}" name="{{{$name}}" id="{{{$name}}" class="border border-gray-200 rounded p-2 w-full" value="{{old($name}}}" required {{$attributes}}|
5 <x-form.error name="{{{$name}}}" />
```

```
25 <div class="mt-8 md:mt-0 flex items-center">
26 |
27 | @auth
28 | <x-dropdown>
29 |   <x-slot name="trigger">
30 |     <button class="text-xs font-bold uppercase">Welcome, {{auth()->user()->name}}!</button>
31 |   </x-slot>
32 |   <x-dropdown-item href="/admin/dashboard" class="{{request()->is('/admin/dashboard') ? 'bg-blue-500 text-white' : ''}}">Dashboard</x-dropdown-item>
33 |   <x-dropdown-item href="/admin/posts/create" class="{{request()->is('/admin/posts/create') ? 'bg-blue-500 text-white' : ''}}">New Post</x-dropdown-item>
34 |   <x-dropdown-item href="#" x-data="{}" @click.prevent="document.querySelector('#logout-form').submit()">Log Out</x-dropdown-item>
35 |   <form id="logout-form" action="/logout" method="POST" class="hidden">
36 |     @csrf
37 |   </form>
38 </x-dropdown>
```

```
1 @props(['heading'])
2 <section class="py-8 max-w-4xl mx-auto">
3   <h1 class="text-lg font-bold mb-8 pb-2 border-b">{{{$heading}}}</h1>
4
5   <div class="flex">
6     <aside class="w-48">
7       <h4 class="font-semibold mb-4">Links</h4>
8       <ul>
9         <li>
10          <a href="/admin/dashboard" class="{{request()->is('/admin/dashboard') ? 'text-blue-500' : ''}}">Dashboard</a>
11        </li>
12        <li>
13          <a href="/admin/posts/create" class="{{request()->is('/admin/posts/create') ? 'text-blue-500' : ''}}">New Post</a>
14        </li>
15      </ul>
16    </aside>
17    <main class="flex-1">
18      <x-panel>
19        {{slot}}
20      </x-panel>
21    </main>
22  </div>
23
24 </section>
```

```
1 @props(['trigger'])
2
3 <div x-data="{show:false}" @click.away="show=false" class="relative">
4
```

```
3 <x-form.label name="{{{$name}}}" />
4 <textarea name="{{{$name}}" id="{{{$name}}" rows="{{{$rows}}" class="border border-gray-200 rounded p-2 w-full" value="{{old($name}}}" required></textarea>
5 <x-form.error name="{{{$name}}}" />
```

```
1 <x-layout>
2   <x-setting heading="Publish New Post">
3     <form action="/admin/posts" method="POST" enctype="multipart/form-data">
```

```
3 <section class="px-6 py-8">
4   <main class="max-w-lg mx-auto mt-10">
5     <x-panel>
6       <h1 class="text-center font-bold text-xl">Register</h1>
7       <form method="POST" action="/register" class="mt-10">
8         @csrf
9         <x-form.input name="name" />
10        <x-form.input name="username" />
11        <x-form.input name="email" type="email" autocomplete="username" />
12        <x-form.input name="password" type="password" autocomplete="new-password" />
13        <x-submit-button>Register</x-submit-button>
14      </form>
15    </x-panel>
16  </main>
```

## 67. Create a Form to Edit and Delete Posts

Ahora vamos a completar el CRUD creando las funciones y vistas para editar y eliminar posts

```

3      <x-form.label name="{{ $name }}" />
4      <input type="{{ $type }}" name="{{ $name }}" id="{{ $name }}" class="border border-gray-200 rounded p-2 w-full"
5      |      [{{ $attributes(['value' => old($name)]) }}]>
6      <x-form.error name="{{ $name }}" />

```

```

25
26 Route::get('/admin/posts', [AdminPostController::class, 'index'])->middleware('admin');
27 Route::get('/admin/posts/create', [AdminPostController::class, 'create'])->middleware('admin');
28 Route::post('/admin/posts', [AdminPostController::class, 'store'])->middleware('admin');
29 Route::get('/admin/posts/{post}/edit', [AdminPostController::class, 'edit'])->middleware('admin');
30 Route::patch('/admin/posts/{post}', [AdminPostController::class, 'update'])->middleware('admin');
31 Route::delete('/admin/posts/{post}', [AdminPostController::class, 'destroy'])->middleware('admin');

```

```

1 <x-layout>
2 CAUsers\cris\USW811_Cristian\VMs\webservice\sites\lfts.isw811.xyz\resources - Contains emphasized items
3 <x-setting heading="Edit Post">
4     <form action="/admin/posts/{{ $post->id }}" method="POST" enctype="multipart/form-data">
5         @csrf
6         @method('PATCH')
7         <x-form.input name="title" :value="old('title', $post->title)" />
8         <x-form.input name="slug" :value="old('slug', $post->slug)" />
9         <div class="flex mt-6">
10             <div class="flex-1">
11                 <x-form.input name="thumbnail" type="file" :value="old('thumbnail', $post->thumbnail)" />
12             </div>
13             
14             </div>
15             <x-form.textarea name="excerpt">{{ old('excerpt', $post->excerpt) }}</x-form.textarea>
16             <x-form.textarea name="body" rows="5">{{ old('body', $post->body) }}</x-form.textarea>
17             <x-form.section>
18                 <x-form.label name="category" />
19                 <select name="category_id" id="category_id">
20                     @foreach (\App\Models\Category::all() as $category)
21                         <option value="{{ $category->id }}" {{ old('category_id', $post->category_id) == $category->id ? 'selected' : '' }}>{{ ucwords($category->name) }}</option>
22                     @endforeach
23                 </select>
24                 <x-form.error name="category_id" />
25             </x-form.section>
26             <x-submit-button>Update</x-submit-button>
27         </form>
28     </x-setting>
29
30 </x-layout>

```

```

1 <x-layout>
2   <x-setting heading="Manage Posts">
3     <div class="flex flex-col">
4       <div class="-my-2 overflow-x-auto sm:-mx-6 lg:-mx-8">
5         <div class="py-2 align-middle inline-block min-w-full sm:px-6 lg:px-8">
6           <div class="shadow overflow-hidden border-b border-gray-200 sm:rounded-lg">
7             <table class="min-w-full divide-y divide-gray-200">
8               <tbody class="bg-white divide-y divide-gray-200">
9                 @foreach ($posts as $post)
10                  <tr>
11                    <td class="px-6 py-4 whitespace-nowrap">
12                      <div class="flex items-center">
13                        <div class="text-sm font-medium text-gray-900">
14                          <a href="/posts/{{ $post->slug }}">
15                            {{ $post->title }}
16                          </a>
17                        </div>
18                      </div>
19                    </td>
20                  </tr>
21                  <tr>
22                    <td class="px-6 py-4 whitespace-nowrap text-right text-sm font-medium">
23                      <a href="/admin/posts/{{ $post->id }}/edit" class="text-blue-500 hover:text-blue-600">Edit</a>
24                    </td>
25                  </tr>
26                  <tr>
27                    <td class="px-6 py-4 whitespace-nowrap text-right text-sm font-medium">
28                      <form method="POST" action="/admin/posts/{{ $post->id }}">
29                        @csrf
30                        @method('DELETE')
31                        <button class="text-xs text-red-400">Delete</button>
32                      </form>
33                    </td>
34                  </tr>
35                @endforeach
36              </tbody>
37            </table>
38          </div>
39        </div>
40      </div>
41    </x-setting>
42  </x-layout>

```

```

30 </x-slot>
31 <x-dropdown-item href="/admin/posts" class="{{(request()-is('admin/posts')) ? 'bg-blue-500 text-white' : ''}}">All Posts</x-dropdown-item>
32 <x-dropdown-item href="/admin/posts/create" class="{{(request()-is('admin/posts/create')) ? 'bg-blue-500 text-white' : ''}}">New Post</x-dropdown-item>
33 <x-dropdown-item href="#" x-data="" @click.prevent="document.querySelector('#logout-form').submit()">Log Out</x-dropdown-item>

```

```

5 <div class="flex">
6   <aside class="w-48 flex-shrink-0">
7     <h4 class="font-semibold mb-4">Links</h4>
8     <ul>
9       <li>
10        <a href="/admin/posts" class="{{(request()-is('/admin/posts')) ? 'text-blue-500' : ''}}">All Posts</a>
11      </li>
12    </ul>

```

```

3 <x-form.label name="{{ $name }}" />
4 <textarea name="{{ $name }}" id="{{ $name }}" rows="{{ $rows }}" class="border border-gray-200 rounded p-2 w-full" required>{{ $slot ?? old($name) }}</textarea>
5 <x-form.error name="{{ $name }}" />

```

```

2 <a href="/?category={{ $category->slug }}"
3   class="px-3 py-1 border border-blue-300 rounded-full text-blue-300
4

```



```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\Post;
6
7  class PostController extends Controller
8  {
9      public function index()
10     {
11
12         return view('posts.index', [
13             'posts' => Post::latest()->filter(request(['search', 'category', 'author']))->paginate(5)->withQueryString(),
14         ]);
15     }
16
17     public function show(Post $post)
18     {
19         return view('posts.show', [
20             'post' => $post,
21         ]);
22     }
23 }
```

```

9  class AdminPostController extends Controller
10 {
11     public function index()
12     {
13         return view('admin.posts.index', [
14             'posts' => Post::paginate(50)
15         ]);
16     }
17
18     public function create()
19     {
20         return view('admin.posts.create');
21     }
22
23     public function store()
24     {
25         $attributes = request()->validate([
26             'title' => 'required',
27             'slug' => ['required', Rule::unique('posts', 'slug')],
28             'thumbnail' => 'required|image',
29             'excerpt' => 'required',
30             'body' => 'required',
31             'category_id' => ['required', Rule::exists('categories', 'id')],
32         ]);
33
34         $attributes['user_id'] = auth()->id();
35         $attributes['thumbnail'] = request()->file('thumbnail')->store('public/thumbnails');
36         $attributes['thumbnail'] = str_replace('public/', '', $attributes['thumbnail']);
37
38         Post::create($attributes);
39
40         return redirect('/');
41     }
42
43     public function edit(Post $post)
44     {
45         return view('admin.posts.edit', [
46             'post' => $post
47         ]);
48     }
49
50     public function update(Post $post)
51     {
52         $attributes = request()->validate([
53             'title' => 'required',
54             'slug' => ['required', Rule::unique('posts', 'slug')->ignore($post->id)],
55             'thumbnail' => 'image',
56             'excerpt' => 'required',
57             'body' => 'required',
58             'category_id' => ['required', Rule::exists('categories', 'id')],
59         ]);
60
61         if(isset($attributes['thumbnail'])) {
62             $attributes['thumbnail'] = request()->file('thumbnail')->store('public/thumbnails');
63             $attributes['thumbnail'] = str_replace('public/', '', $attributes['thumbnail']);
64         }
65
66         $post->update($attributes);
67
68         return back()->with('success', 'Post Updated!');
69     }
70
71     public function destroy(Post $post)
72     {
73         $post->delete();
74
75         return back()->with('success', 'Post Deleted!');
76     }
77 }

```

## 68. Group and Store Validation Logic

Ahora limpiamos un poco el Controlador de AdminPost para evitar duplicacion de codigo

```

17
18     public function create()
19     {
20         return view('admin.posts.create');
21     }
22
23     public function store()
24     {
25         $attributes = request()->validate([
26             $post = new Post();
27             $attributes = $this->validatePost($post);
28         ]);
29
30         $attributes['user_id'] = auth()->id();
31         $attributes['thumbnail'] = request()->file('thumbnail')->store('public/thumbnails');
32         $attributes['thumbnail'] = str_replace('public/', '', $attributes['thumbnail']);
33
34         Post::create($attributes);
35
36         return redirect('/');
37     }
38
39     public function edit(Post $post)
40     {
41         return view('admin.posts.edit', [
42             'post' => $post
43         ]);
44     }
45
46     public function update(Post $post)
47     {
48         $attributes = $this->validatePost($post);
49
50
51         if(isset($attributes['thumbnail'])) {
52             $attributes['thumbnail'] = request()->file('thumbnail')->store('public/thumbnails');
53             $attributes['thumbnail'] = str_replace('public/', '', $attributes['thumbnail']);
54         }
55
56         $post->update($attributes);
57
58         return back()->with('success', 'Post Updated!');
59     }
60
61     public function destroy(Post $post)
62     {
63         $post->delete();
64
65         return back()->with('success', 'Post Deleted!');
66     }
67
68     protected function validatePost(Post $post): array
69     {
70         return request()->validate([
71             'title' => 'required',
72             'slug' => ['required', Rule::unique('posts', 'slug')->ignore($post)],
73             'thumbnail' => $post->exists ? ['image'] : ['required', 'image'],
74             'excerpt' => 'required',
75             'body' => 'required',
76             'category_id' => ['required', Rule::exists('categories', 'id')],
77         ]);
78     }
79 }

```

## 69. All About Authorization

En este capítulo se soluciona un pequeño bug que hacía que todos los usuarios pudieran ver las opciones de administrador, no tenían acceso, pero las veían

```

17
18 ✓ Route::middleware('guest')->group(function () {
19     Route::get('/register', [RegisterController::class, 'create'])->middleware('guest');
20     Route::post('/register', [RegisterController::class, 'store'])->middleware('guest');
21
22     Route::get('/login', [SessionsController::class, 'create'])->middleware('guest');
23     Route::post('/login', [SessionsController::class, 'store'])->middleware('guest');
24 });
25
26 Route::post('/logout', [SessionsController::class, 'destroy'])->middleware('auth');
27
28 ✓ Route::middleware('can:admin')->group(function () {
29     Route::resource('admin/posts', AdminPostController::class)->except('show');
30 });

```

```

31 @admin
32     <x-dropdown-item href="/admin/posts" class="{{request()->is('admin/posts') ? 'bg-blue-500 text-white' : 'bg-white text-blue-500'}}">
33     <x-dropdown-item href="/admin/posts/create" class="{{request()->is('admin/posts/create') ? 'bg-blue-500 text-white' : 'bg-white text-blue-500'}}">
34 @endadmin

```

```

44 //Model::unguard();
45
46 Gate::define('admin', function (User $user) {
47     return $user->username === 'CrisVM';
48 });
49 Blade::if('admin', function () {
50     if(!request()->user())
51         return false;
52     return request()->user()->can('admin');
53 });
54 }
55 }

```