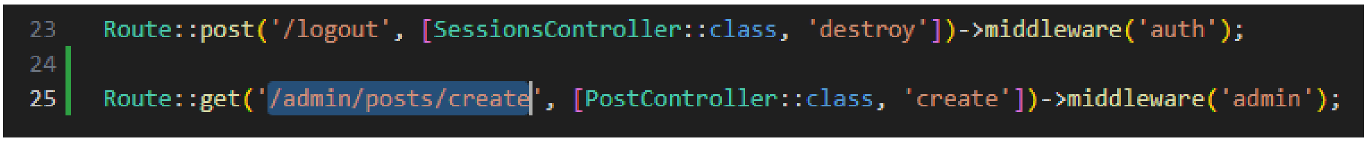
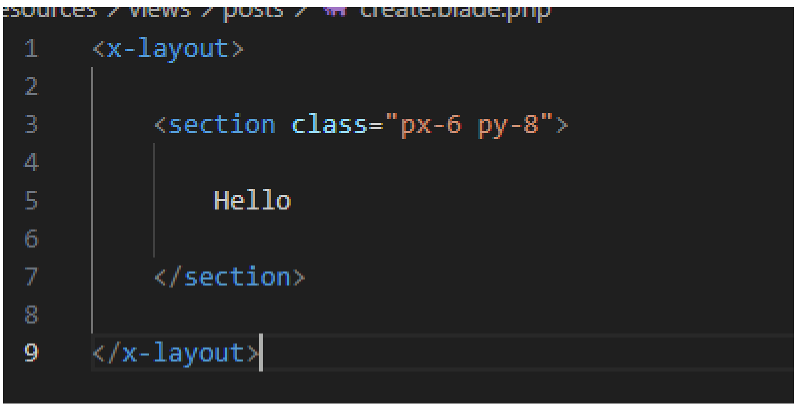
LFTS - Admin Section

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
```

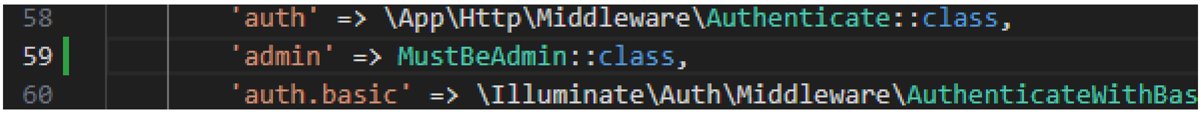


```php
Route::post('/logout', [SessionsController::class, 'destroy'])->middleware('auth');

Route::get('/admin/posts/create', [PostController::class, 'create'])->middleware('admin');
```

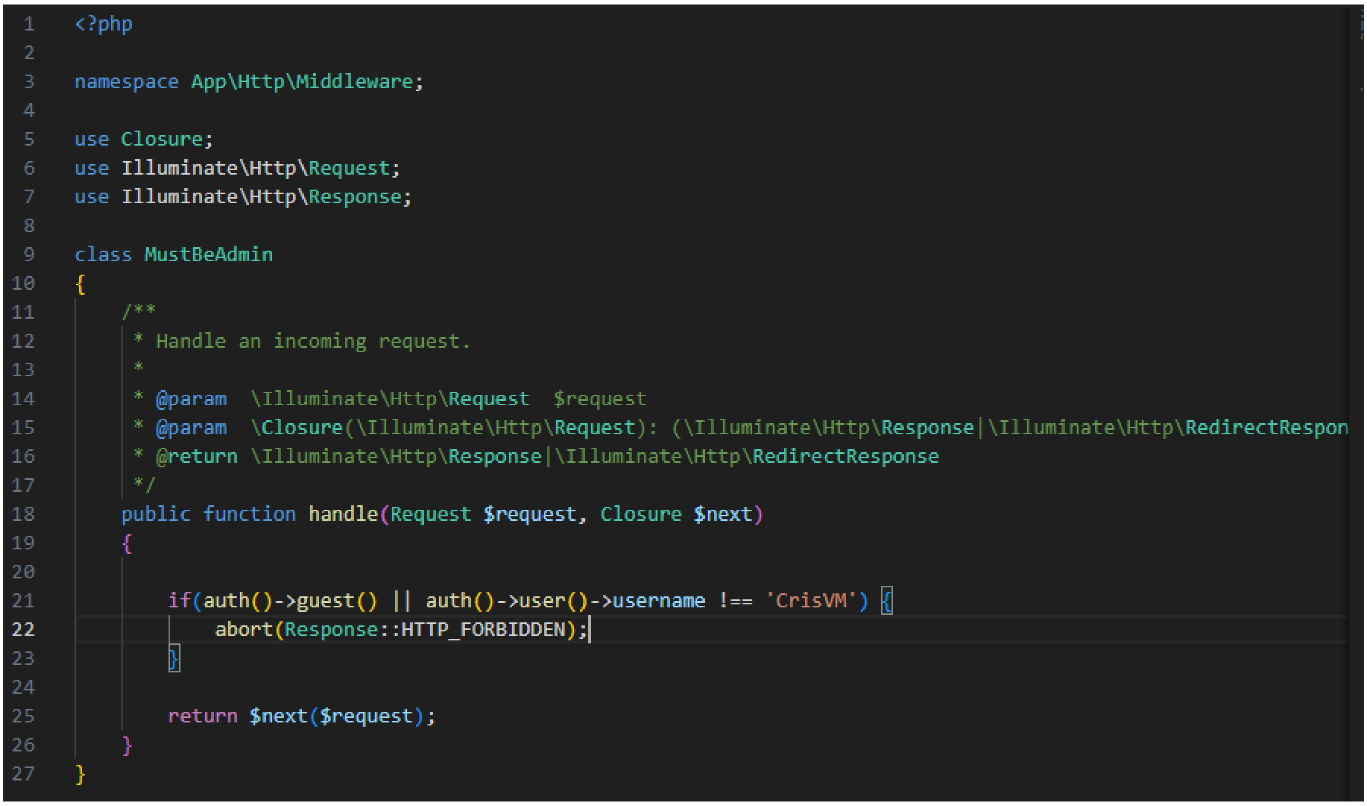


esources > views > posts > create.blade.php

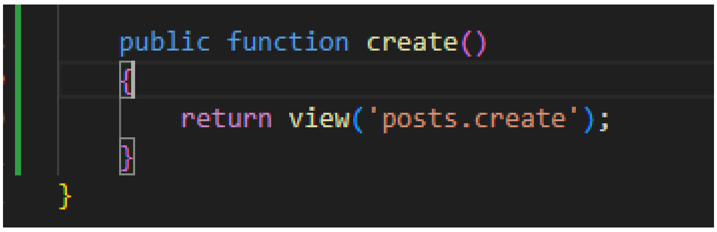
```html
<x-layout>

    <section class="px-6 py-8">

        Hello

    </section>

</x-layout>
```



```php
            'auth' => \App\Http\Middleware\Authenticate::class,
            'admin' => MustBeAdmin::class,
            'auth.basic' => \Illuminate\Auth\Middleware\AuthenticateWithBas
```



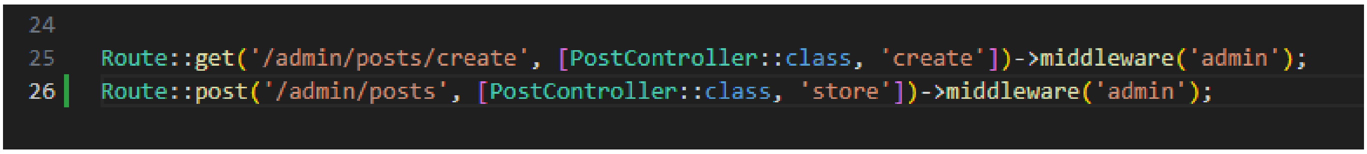
```php
<?php

namespace App\Http\Middleware;

use Closure;
use Illuminate\Http\Request;
use Illuminate\Http\Response;

class MustBeAdmin
{
    /**
     * Handle an incoming request.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  \Closure(\Illuminate\Http\Request): (\Illuminate\Http\Response|\Illuminate\Http\RedirectRespon
     * @return \Illuminate\Http\Response|\Illuminate\Http\RedirectResponse
     */
    public function handle(Request $request, Closure $next)
    {

        if(auth()->guest() || auth()->user()->username !== 'CrisVM') {
            abort(Response::HTTP_FORBIDDEN);
        }

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



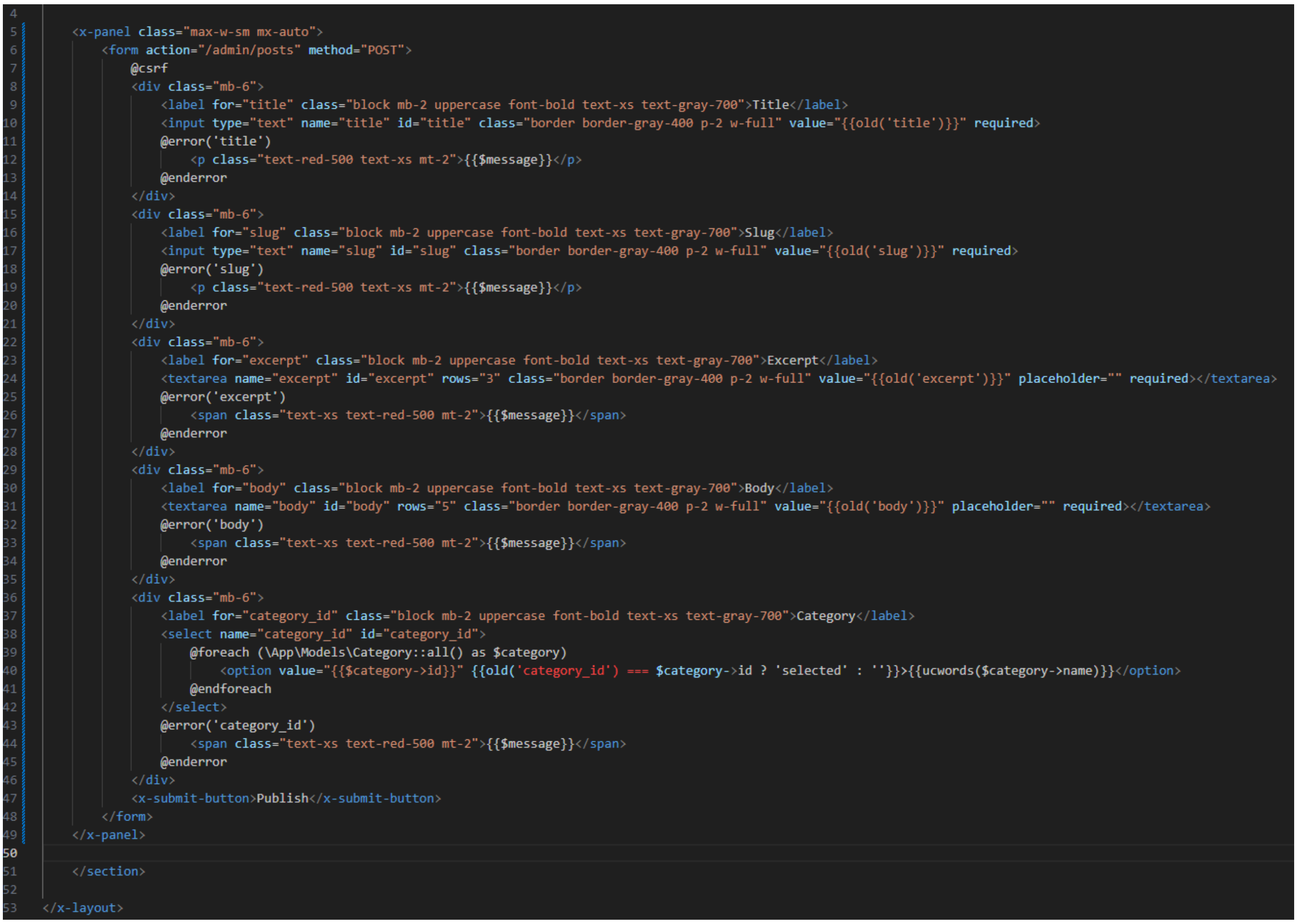
```
    public function create()
    {
        return view('posts.create');
    }
}
```

63. Create the Publish Post Form

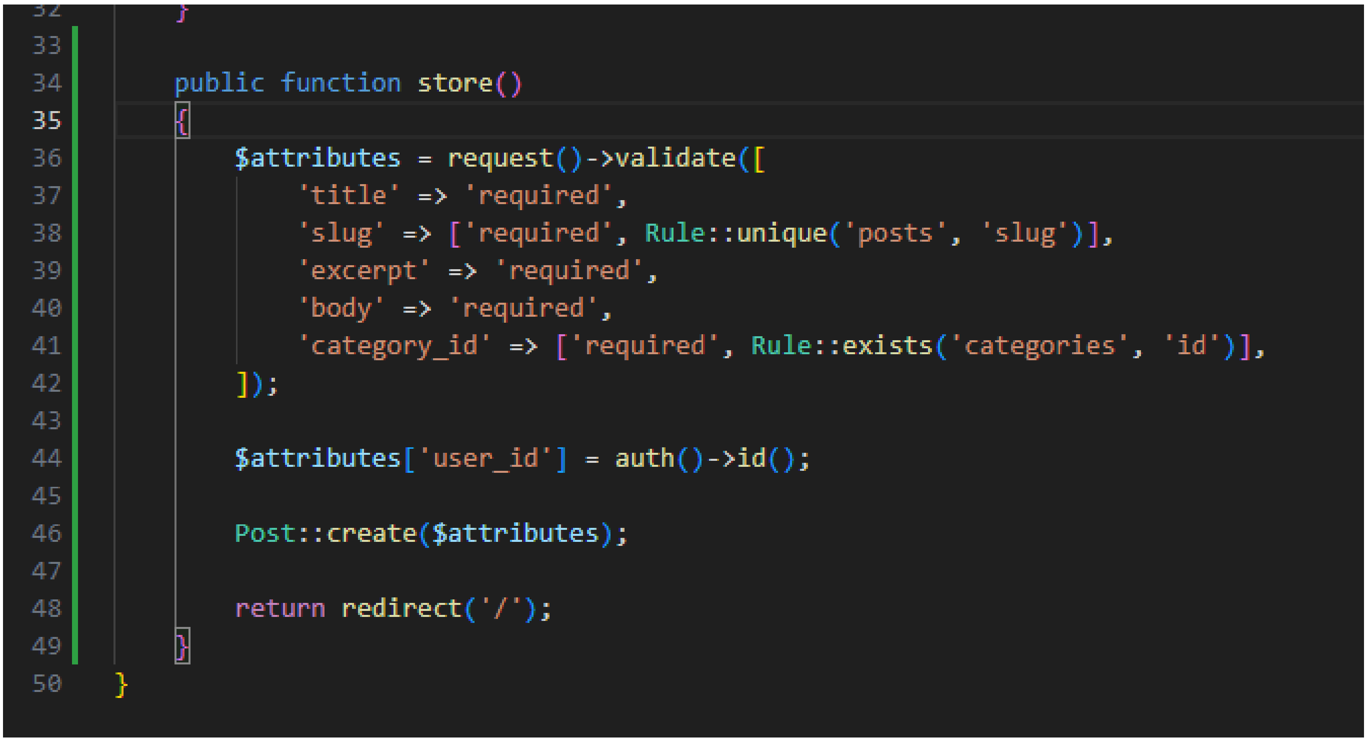
```
Ahora vamos a crear una forma para que el administrador pueda crear post
```



```

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



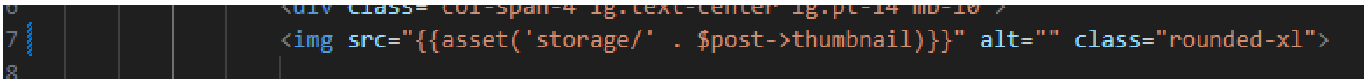
```

    <x-panel class="max-w-sm mx-auto">
        <form action="/admin/posts" method="POST">
            @csrf
            <div class="mb-6">
                <label for="title" class="block mb-2 uppercase font-bold text-xs text-gray-700">Title</label>
                <input type="text" name="title" id="title" class="border border-gray-400 p-2 w-full" value="{{old('title')}}" required>
                @error('title')
                    <p class="text-red-500 text-xs mt-2">{{$message}}</p>
                @enderror
            </div>
            <div class="mb-6">
                <label for="slug" class="block mb-2 uppercase font-bold text-xs text-gray-700">Slug</label>
                <input type="text" name="slug" id="slug" class="border border-gray-400 p-2 w-full" value="{{old('slug')}}" required>
                @error('slug')
                    <p class="text-red-500 text-xs mt-2">{{$message}}</p>
                @enderror
            </div>
            <div class="mb-6">
                <label for="excerpt" class="block mb-2 uppercase font-bold text-xs text-gray-700">Excerpt</label>
                <textarea name="excerpt" id="excerpt" rows="3" class="border border-gray-400 p-2 w-full" value="{{old('excerpt')}}" placeholder="" required></textarea>
                @error('excerpt')
                    <span class="text-xs text-red-500 mt-2">{{$message}}</span>
                @enderror
            </div>
            <div class="mb-6">
                <label for="body" class="block mb-2 uppercase font-bold text-xs text-gray-700">Body</label>
                <textarea name="body" id="body" rows="5" class="border border-gray-400 p-2 w-full" value="{{old('body')}}" placeholder="" required></textarea>
                @error('body')
                    <span class="text-xs text-red-500 mt-2">{{$message}}</span>
                @enderror
            </div>
            <div class="mb-6">
                <label for="category_id" class="block mb-2 uppercase font-bold text-xs text-gray-700">Category</label>
                <select name="category_id" id="category_id">
                    @foreach (\App\Models\Category::all() as $category)
                        <option value="{{$category->id}}" {{old('category_id') === $category->id ? 'selected' : ''}}>{{ucwords($category->name)}}</option>
                    @endforeach
                </select>
                @error('category_id')
                    <span class="text-xs text-red-500 mt-2">{{$message}}</span>
                @enderror
            </div>
            <x-submit-button>Publish</x-submit-button>
        </form>
    </x-panel>

    </section>

</x-layout>
```



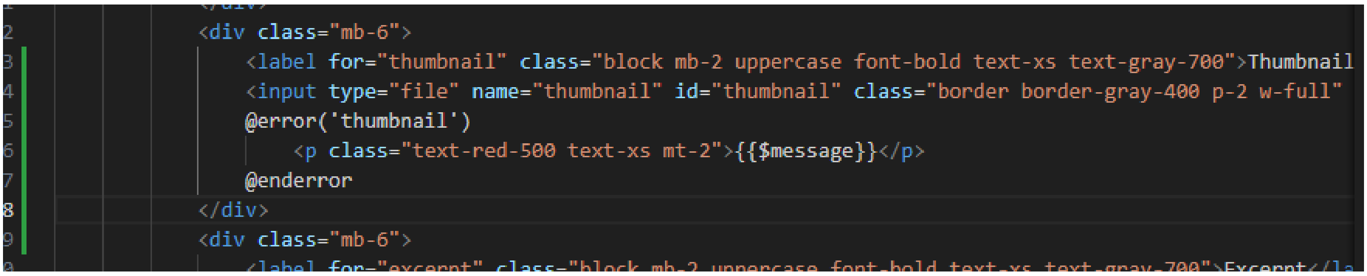
```php
    }

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

        $attributes['user_id'] = auth()->id();

        Post::create($attributes);

        return redirect('/');
    }
}
```

64. Validate and Store Post Thumbnails

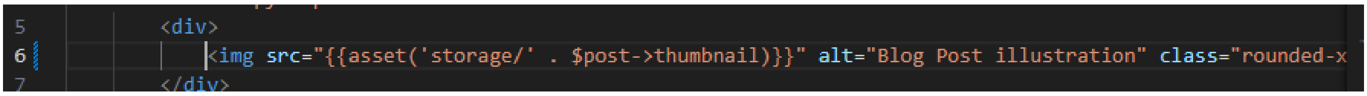
```
En este episodio hacemos todos los pasos necesarios para que ahora los post puedan
incluir una miniatura personalizada
```



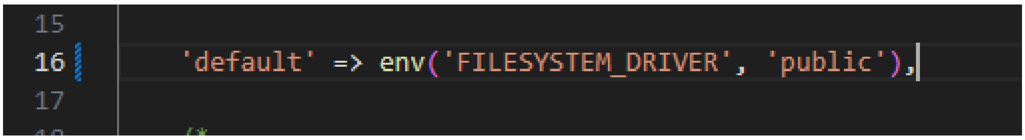
```html
<div class="col-span-4 lg:text-center lg:pt-14 mb-10">
    <img src="{{asset('storage/' . $post->thumbnail)}}" alt="" class="rounded-xl">
```



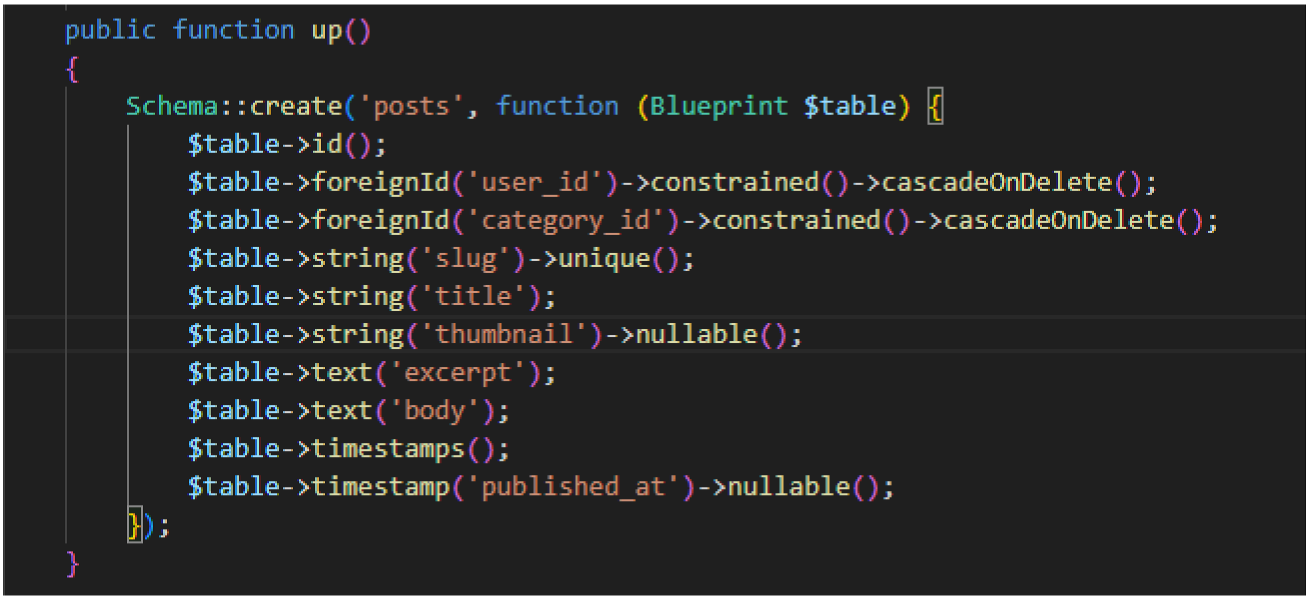
```html
</div>
<div class="mb-6">
    <label for="thumbnail" class="block mb-2 uppercase font-bold text-xs text-gray-700">Thumbnail
    <input type="file" name="thumbnail" id="thumbnail" class="border border-gray-400 p-2 w-full"
    @error('thumbnail')
        <p class="text-red-500 text-xs mt-2">{{$message}}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



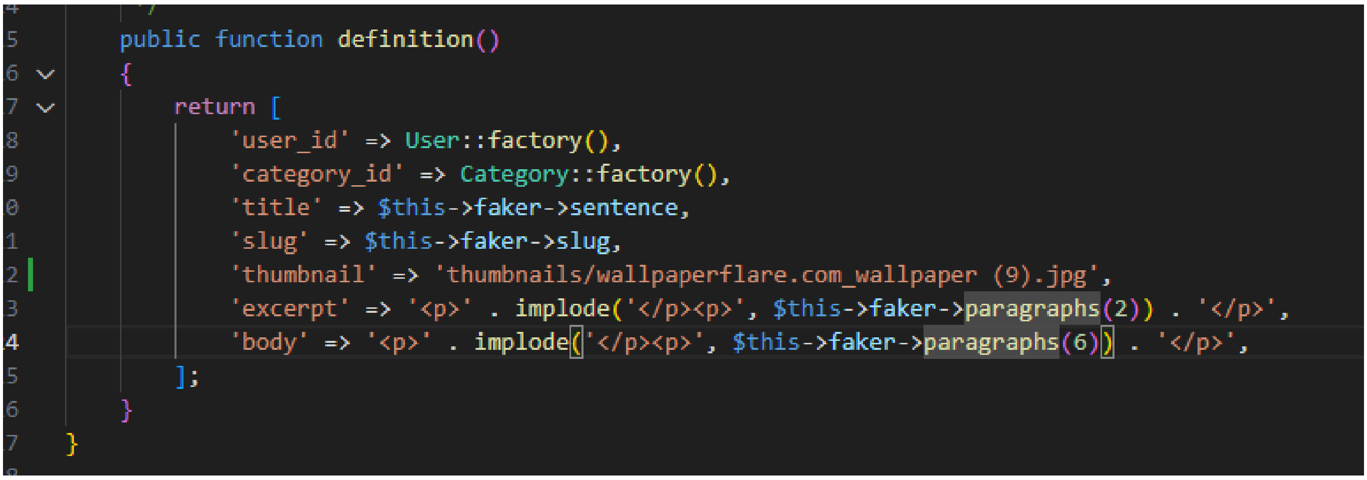
```html
<div>
    <img src="{{asset('storage/' . $post->thumbnail)}}" alt="Blog Post illustration" class="rounded-x
</div>
```



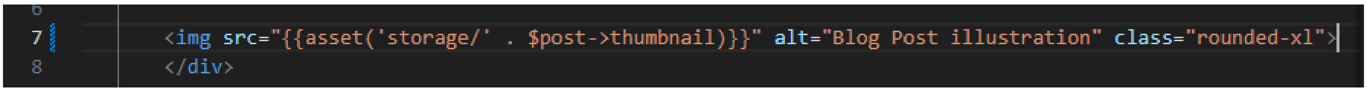
```php
'default' => env('FILESYSTEM_DRIVER', 'public'),

/*
```



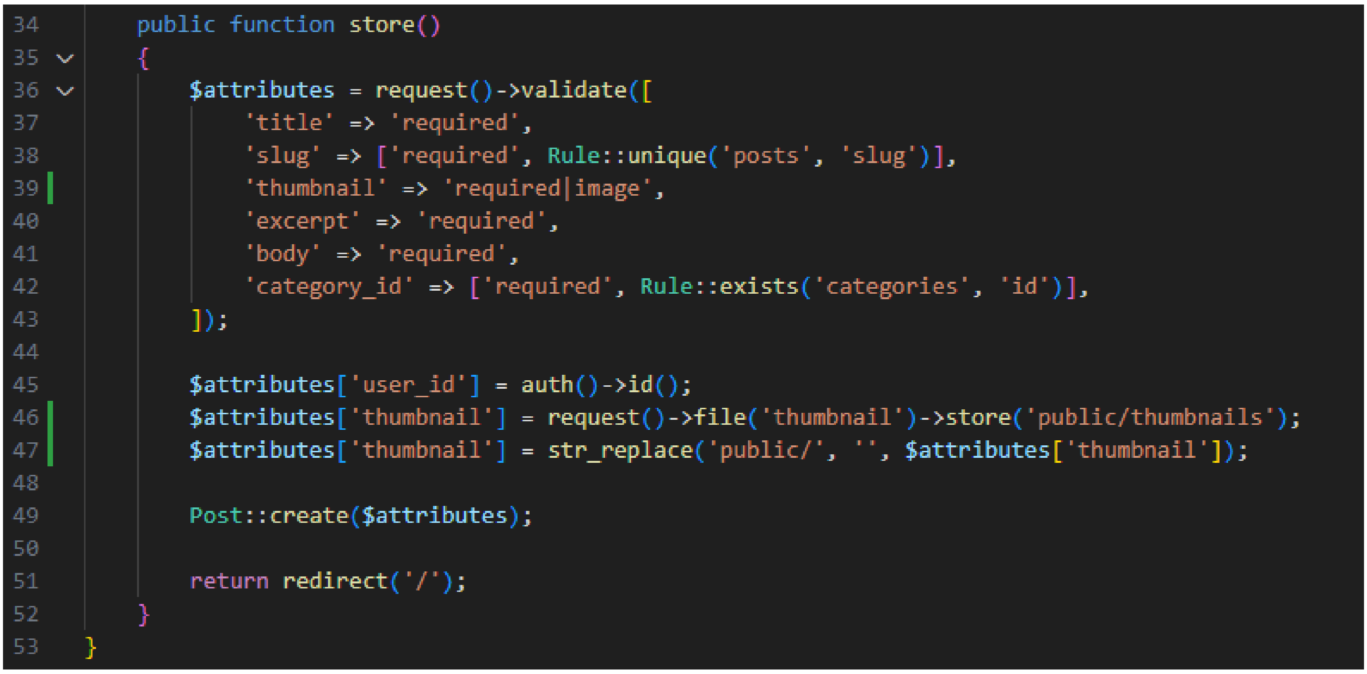
```php
public function up()
{
    Schema::create('posts', function (Blueprint $table) {
        $table->id();
        $table->foreignId('user_id')->constrained()->cascadeOnDelete();
        $table->foreignId('category_id')->constrained()->cascadeOnDelete();
        $table->string('slug')->unique();
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



```php
    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>',
        ];
    }
}
```



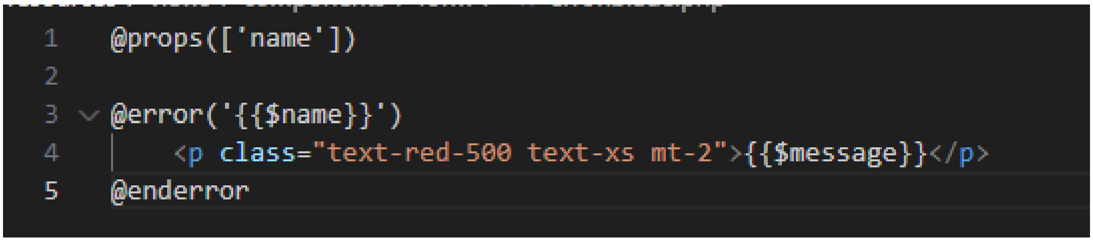
```html
    <img src="{{asset('storage/' . $post->thumbnail)}}" alt="Blog Post illustration" class="rounded-xl">
    </div>
```



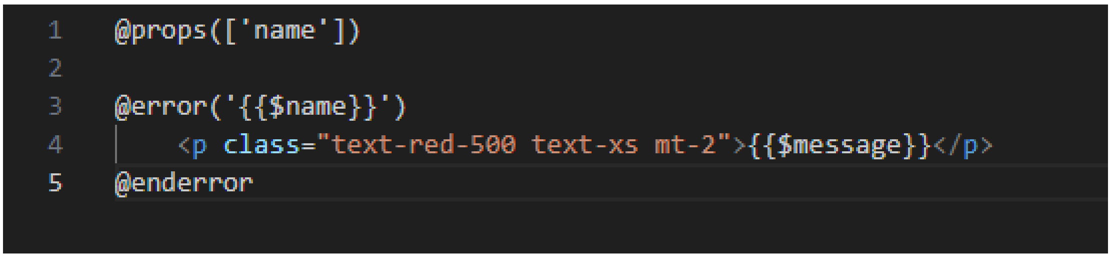
```php
    public function store()
    {
        $attributes = request()->validate([
            'title' => 'required',
            'slug' => ['required', Rule::unique('posts', 'slug')],
            'thumbnail' => 'required|image',
            'excerpt' => 'required',
            'body' => 'required',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

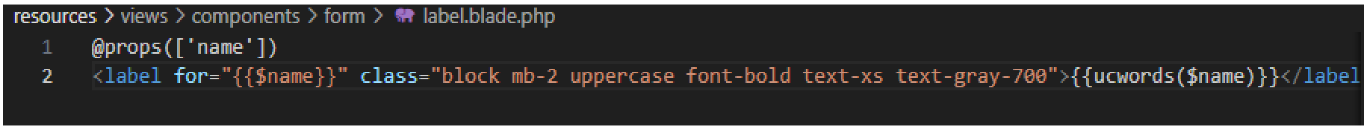
```
Reorganizamos la vista para hacerla mas modular, por decirlo de alguna manera
```



```
@props(['name'])

@error('{{$name}}')
    <p class="text-red-500 text-xs mt-2">{{$message}}</p>
@enderror
```

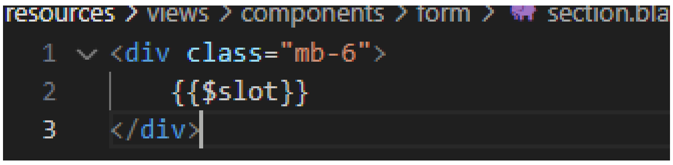


```
@props(['name'])

@error('{{$name}}')
    <p class="text-red-500 text-xs mt-2">{{$message}}</p>
@enderror
```



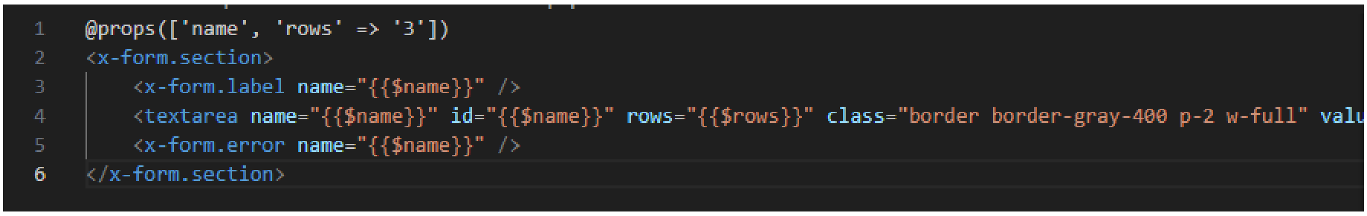
resources > views > components > form > label.blade.php

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

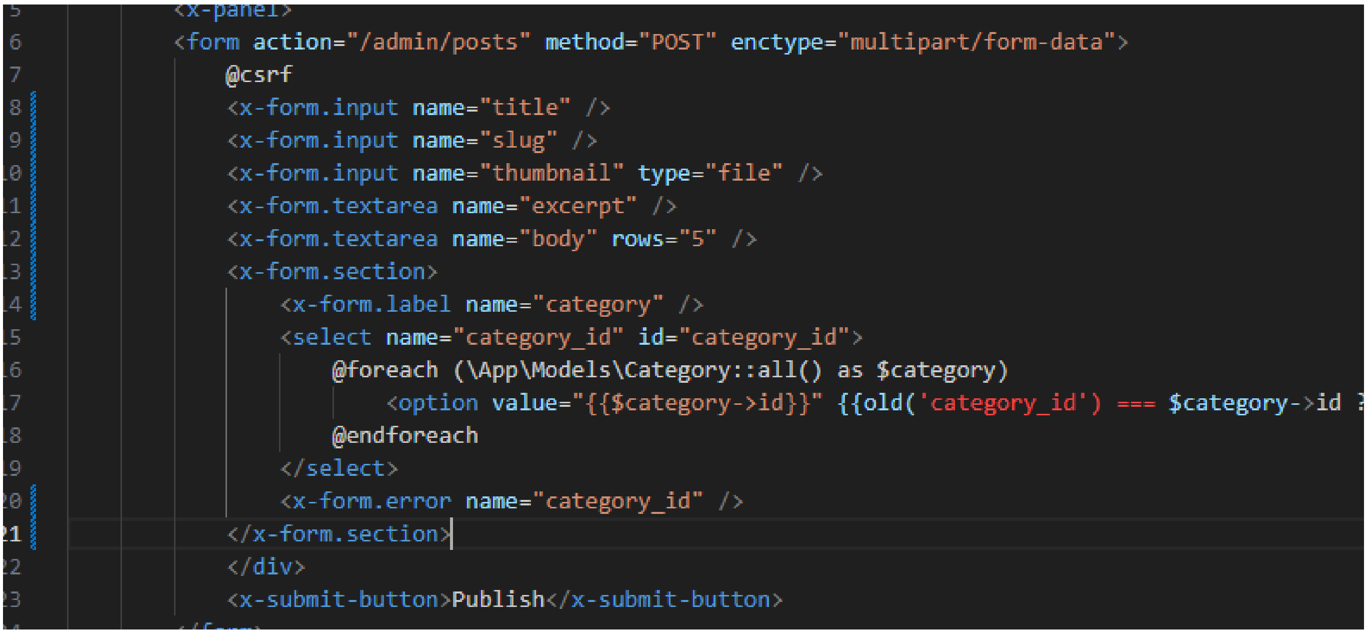


resources > views > components > form > section.bla

```
<div class="mb-6">
    {{$slot}}
</div>
```



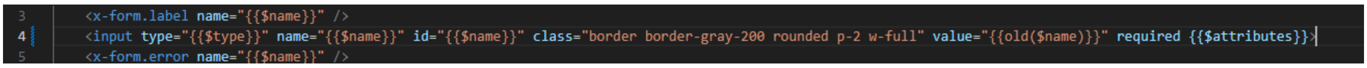
```
@props(['name', 'rows' => '3'])
<x-form.section>
    <x-form.label name="{{$name}}" />
    <textarea name="{{$name}}" id="{{$name}}" rows="{{$rows}}" class="border border-gray-400 p-2 w-full" valu
    <x-form.error name="{{$name}}" />
</x-form.section>
```



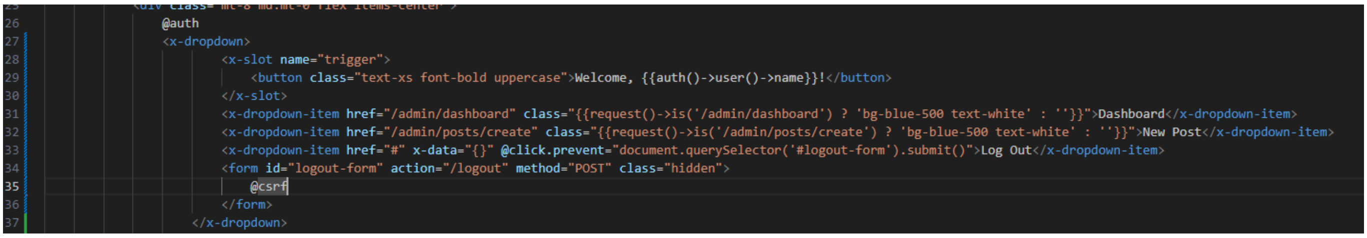
```
        <x-panel>
        <form action="/admin/posts" method="POST" enctype="multipart/form-data">
            @csrf
            <x-form.input name="title" />
            <x-form.input name="slug" />
            <x-form.input name="thumbnail" type="file" />
            <x-form.textarea name="excerpt" />
            <x-form.textarea name="body" rows="5" />
            <x-form.section>
                <x-form.label name="category" />
                <select name="category_id" id="category_id">
                    @foreach (\App\Models\Category::all() as $category)
                        <option value="{{$category->id}}" {{old('category_id') === $category->id ?
                    @endforeach
                </select>
                <x-form.error name="category_id" />
            </x-form.section>
            </div>
            <x-submit-button>Publish</x-submit-button>
```

66. Extend the Admin Layout

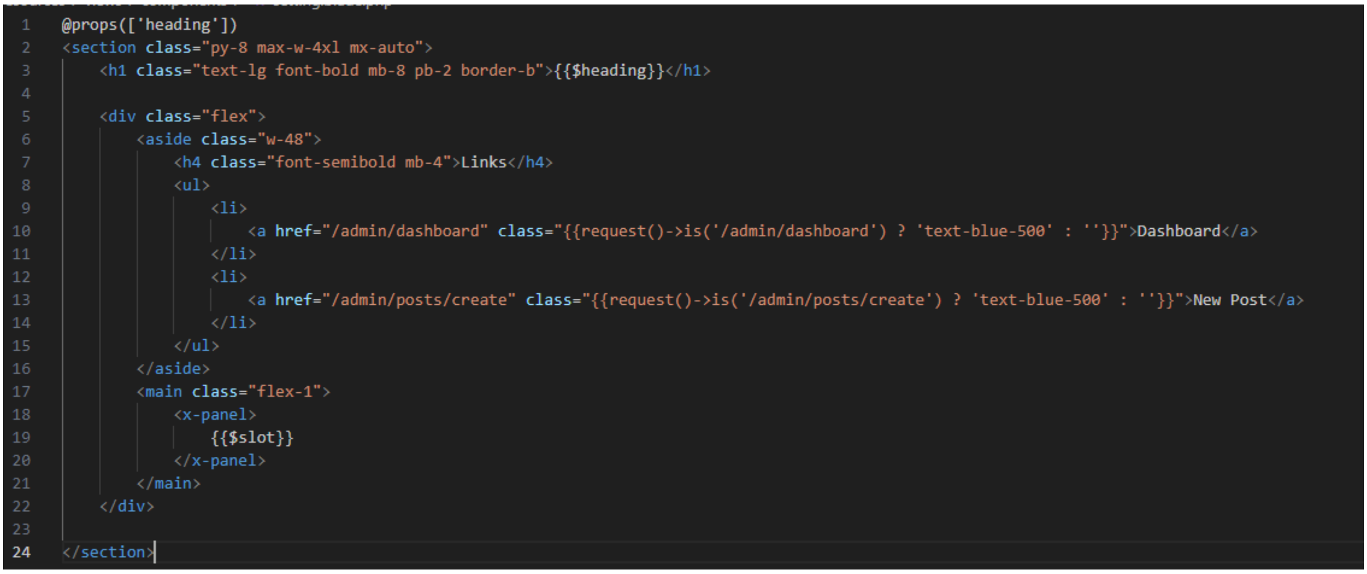
```
Ahora vamos a modificar varias cosas para que la interfaz sea mas util y amigable
con el cliente o administrador
```



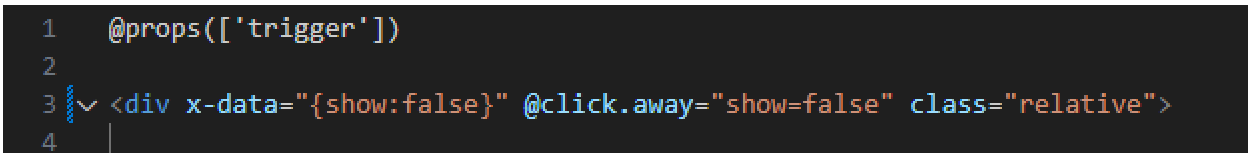
```
<x-form.label name="{{$name}}" />
<input type="{{$type}}" name="{{$name}}" id="{{$name}}" class="border border-gray-200 rounded p-2 w-full" value="{{old($name)}}" required {{$attributes}}>
<x-form.error name="{{$name}}" />
```



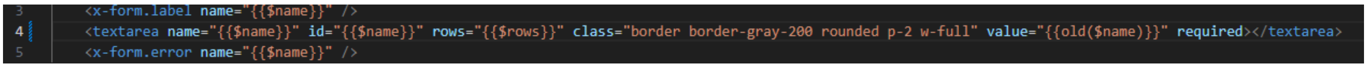
```
            <div class="mt-8 md:mt-0 flex items-center">
                @auth
                <x-dropdown>
                    <x-slot name="trigger">
                        <button class="text-xs font-bold uppercase">Welcome, {{auth()->user()->name}}!</button>
                    </x-slot>
  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



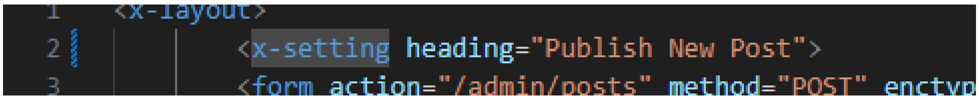
```
@props(['heading'])
<section class="py-8 max-w-4xl mx-auto">
    <h1 class="text-lg font-bold mb-8 pb-2 border-b">{{$heading}}</h1>

    <div class="flex">
        <aside class="w-48">
            <h4 class="font-semibold mb-4">Links</h4>
            <ul>
                <li>
                    <a href="/admin/dashboard" class="{{request()->is('/admin/dashboard') ? 'text-blue-500' : ''}}">Dashboard</a>
                </li>
                <li>
                    <a href="/admin/posts/create" class="{{request()->is('/admin/posts/create') ? 'text-blue-500' : ''}}">New Post</a>
                </li>
            </ul>
        </aside>
        <main class="flex-1">
            <x-panel>
                {{$slot}}
            </x-panel>
        </main>
    </div>

</section>
```



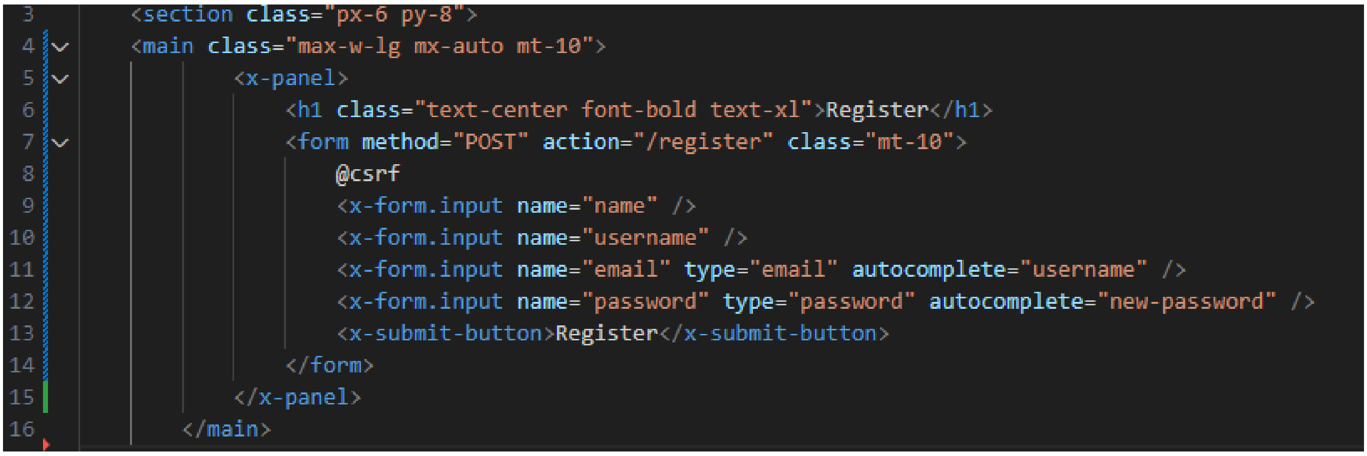
```
@props(['trigger'])

<div x-data="{show:false}" @click.away="show=false" class="relative">

```



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



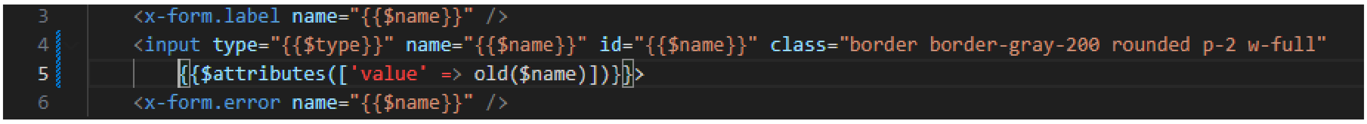
```
<x-layout>
        <x-setting heading="Publish New Post">
        <form action="/admin/posts" method="POST" enctyp
```



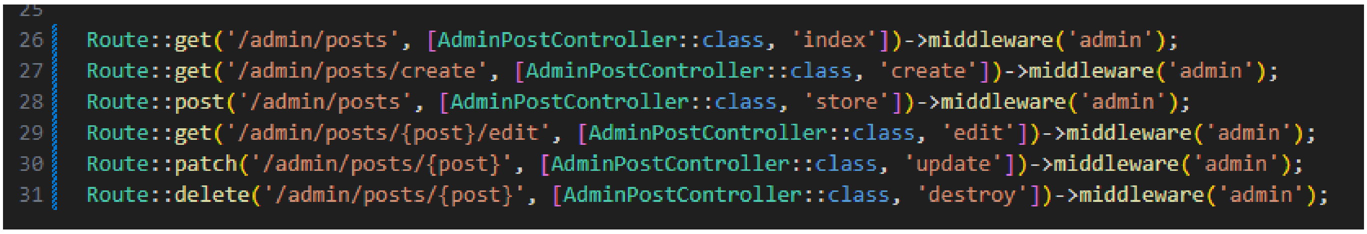
```
    <section class="px-6 py-8">
    <main class="max-w-lg mx-auto mt-10">
        <x-panel>
            <h1 class="text-center font-bold text-xl">Register</h1>
            <form method="POST" action="/register" class="mt-10">
                @csrf
                <x-form.input name="name" />
   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

67.Create a Form to Edit and Delete Posts

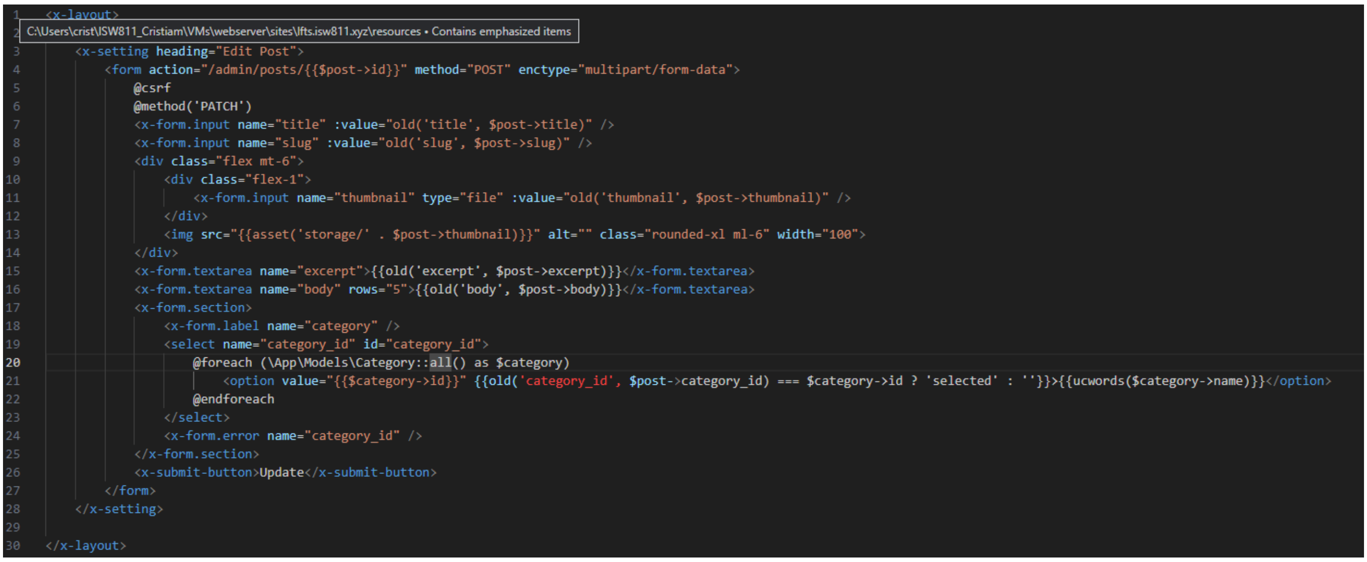
```
Ahora vamos a completar el CRUD creando las funciones y vistas para editar y eliminar posts
```



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

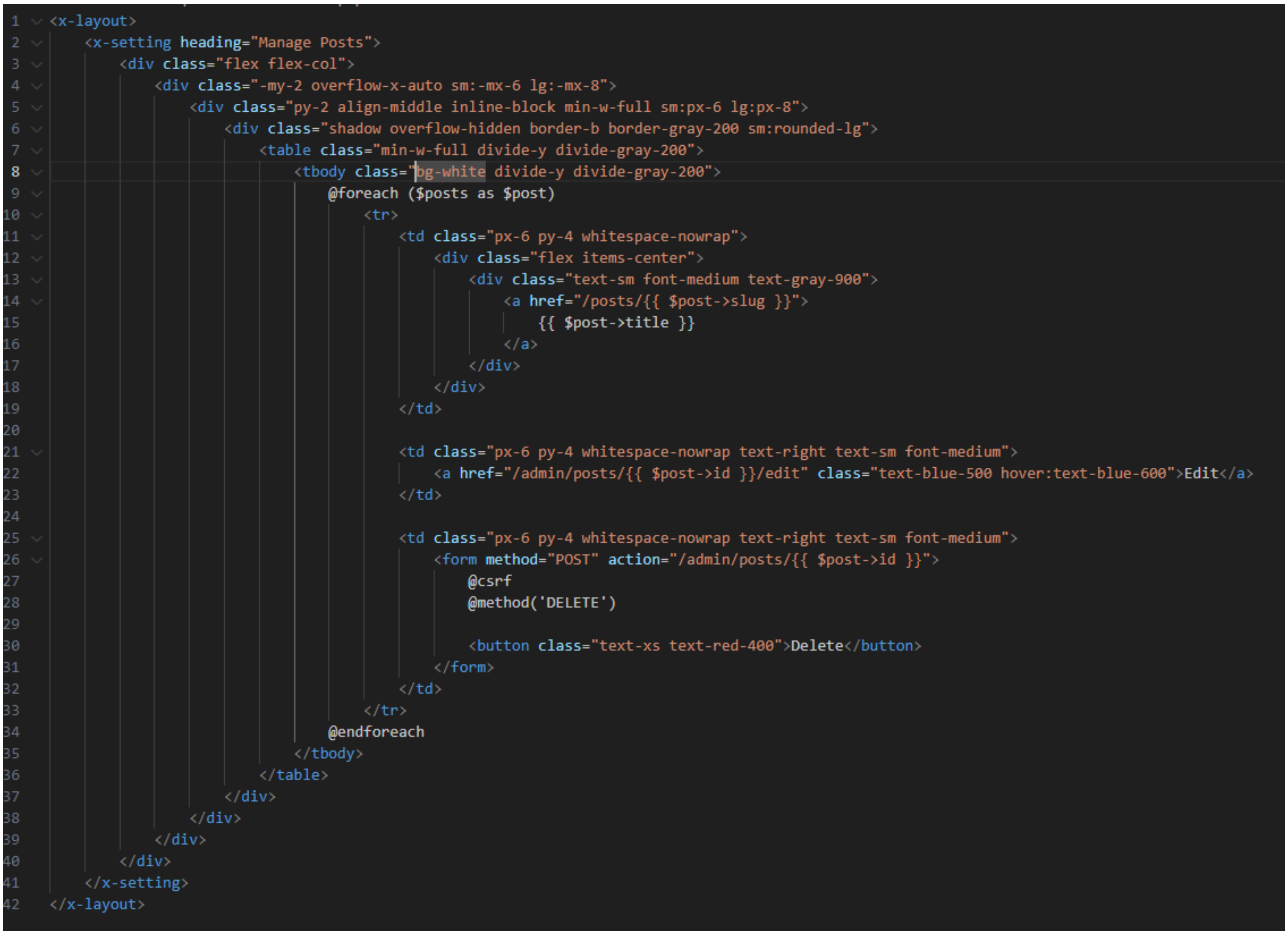


```
Route::get('/admin/posts', [AdminPostController::class, 'index'])->middleware('admin');
Route::get('/admin/posts/create', [AdminPostController::class, 'create'])->middleware('admin');
Route::post('/admin/posts', [AdminPostController::class, 'store'])->middleware('admin');
Route::get('/admin/posts/{post}/edit', [AdminPostController::class, 'edit'])->middleware('admin');
Route::patch('/admin/posts/{post}', [AdminPostController::class, 'update'])->middleware('admin');
Route::delete('/admin/posts/{post}', [AdminPostController::class, 'destroy'])->middleware('admin');
```

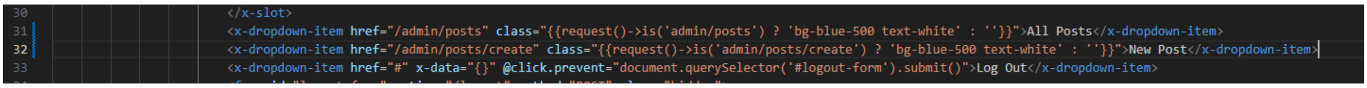


C:\Users\crist\ISW811_Cristiam\VMs\webserver\sites\lfts.isw811.xyz\resources • Contains emphasized items

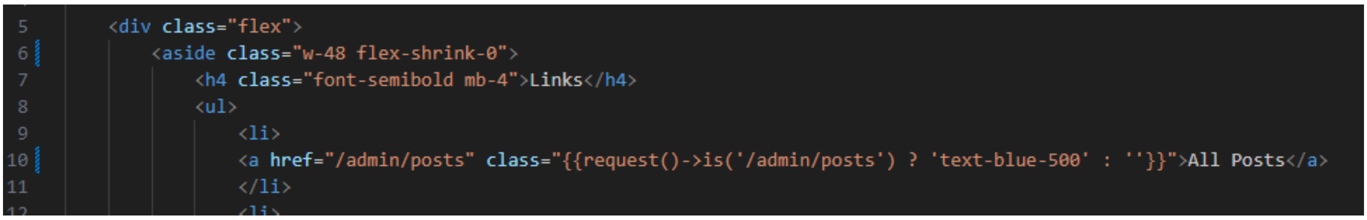
```
<x-layout>

    <x-setting heading="Edit Post">
        <form action="/admin/posts/{{$post->id}}" method="POST" enctype="multipart/form-data">
            @csrf
            @method('PATCH')
            <x-form.input name="title" :value="old('title', $post->title)" />
            <x-form.input name="slug" :value="old('slug', $post->slug)" />
            <div class="flex mt-6">
                <div class="flex-1">
                    <x-form.input name="thumbnail" type="file" :value="old('thumbnail', $post->thumbnail)" />
                </div>
                <img src="{{asset('storage/' . $post->thumbnail)}}" alt="" class="rounded-xl ml-6" width="100">
            </div>
            <x-form.textarea name="excerpt">{{old('excerpt', $post->excerpt)}}</x-form.textarea>
            <x-form.textarea name="body" rows="5">{{old('body', $post->body)}}</x-form.textarea>
            <x-form.section>
                <x-form.label name="category" />
                <select name="category_id" id="category_id">
                    @foreach (\App\Models\Category::all() as $category)
                        <option value="{{$category->id}}" {{old('category_id', $post->category_id) === $category->id ? 'selected' : ''}}>{{ucwords($category->name)}}</option>
                    @endforeach
                </select>
                <x-form.error name="category_id" />
            </x-form.section>
            <x-submit-button>Update</x-submit-button>
        </form>
    </x-setting>

</x-layout>
```



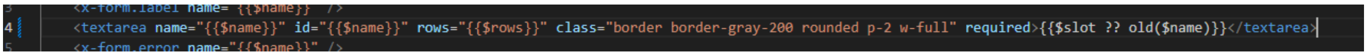
```
<x-layout>
    <x-setting heading="Manage Posts">
        <div class="flex flex-col">
            <div class="-my-2 overflow-x-auto sm:-mx-6 lg:-mx-8">
                <div class="py-2 align-middle inline-block min-w-full sm:px-6 lg:px-8">
                    <div class="shadow overflow-hidden border-b border-gray-200 sm:rounded-lg">
                        <table class="min-w-full divide-y divide-gray-200">
                            <tbody class="bg-white divide-y divide-gray-200">
                                @foreach ($posts as $post)
                                    <tr>
                                        <td class="px-6 py-4 whitespace-nowrap">
                                            <div class="flex items-center">
                                                <div class="text-sm font-medium text-gray-900">
                                                    <a href="/posts/{{ $post->slug }}">
                                                        {{ $post->title }}
                                                    </a>
                                                </div>
                                            </div>
                                        </td>

                                        <td class="px-6 py-4 whitespace-nowrap text-right text-sm font-medium">
                                            <a href="/admin/posts/{{ $post->id }}/edit" class="text-blue-500 hover:text-blue-600">Edit</a>
                                        </td>

                                        <td class="px-6 py-4 whitespace-nowrap text-right text-sm font-medium">
                                            <form method="POST" action="/admin/posts/{{ $post->id }}">
                                                @csrf
                                                @method('DELETE')

                                                <button class="text-xs text-red-400">Delete</button>
                                            </form>
                                        </td>
                                    </tr>
                                @endforeach
                            </tbody>
                        </table>
                    </div>
                </div>
            </div>
        </div>
    </x-setting>
</x-layout>
```



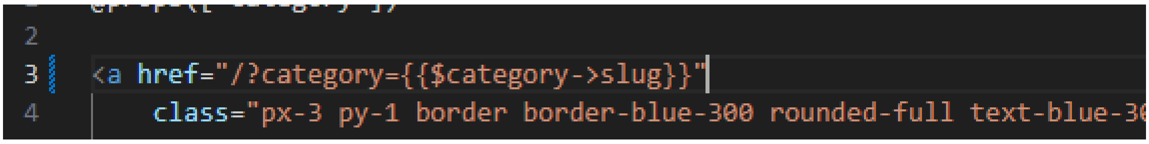
```
</x-slot>
<x-dropdown-item href="/admin/posts" class="{{request()->is('admin/posts') ? 'bg-blue-500 text-white' : ''}}">All Posts</x-dropdown-item>
<x-dropdown-item href="/admin/posts/create" class="{{request()->is('admin/posts/create') ? 'bg-blue-500 text-white' : ''}}">New Post</x-dropdown-item>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



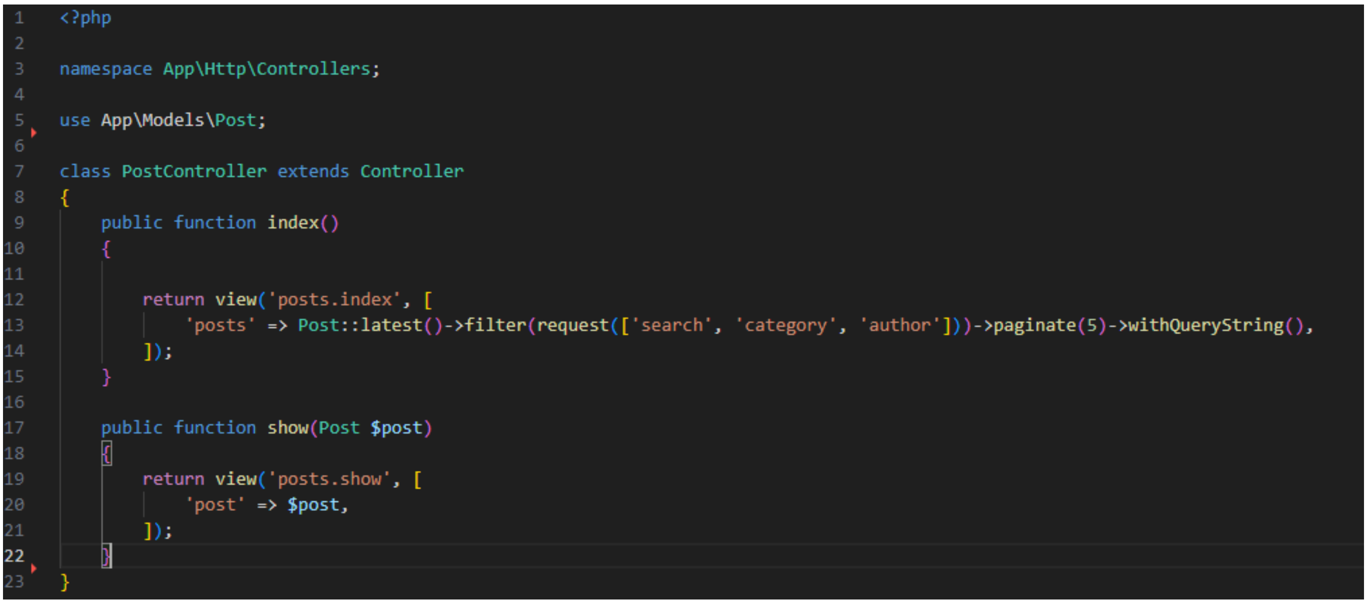
```
<div class="flex">
    <aside class="w-48 flex-shrink-0">
        <h4 class="font-semibold mb-4">Links</h4>
        <ul>
            <li>
            <a href="/admin/posts" class="{{request()->is('/admin/posts') ? 'text-blue-500' : ''}}">All Posts</a>
            </li>
            <li>
```



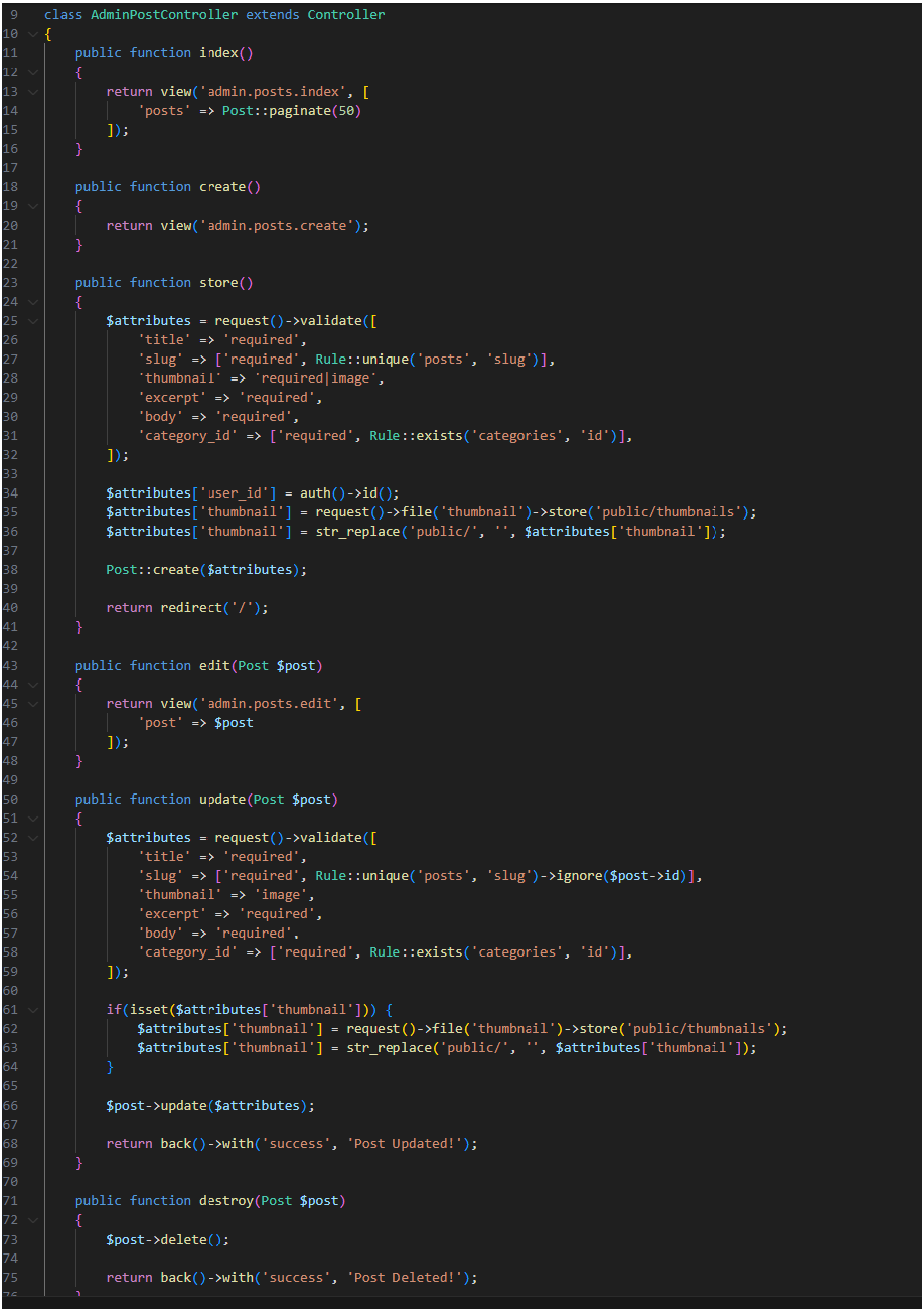
```
<x-form.label name="{{$name}}" />
<textarea name="{{$name}}" id="{{$name}}" rows="{{$rows}}" class="border border-gray-200 rounded p-2 w-full" required>{{$slot ?? old($name)}}</textarea>
<x-form.error name="{{$name}}" />
```



```

<a href="/?category={{$category->slug}}"
    class="px-3 py-1 border border-blue-300 rounded-full text-blue-30
```



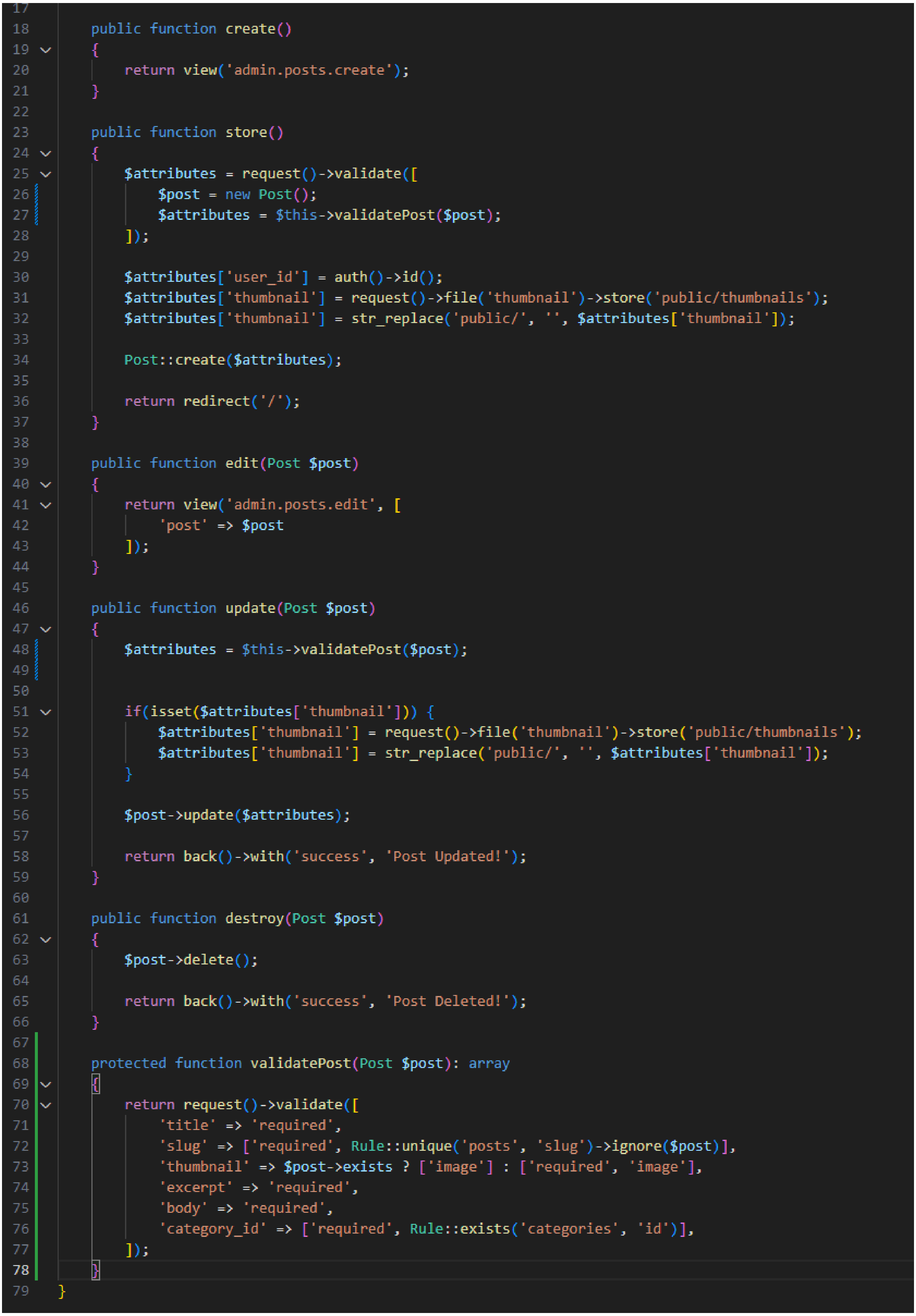
```php
<?php

namespace App\Http\Controllers;

use App\Models\Post;

class PostController extends Controller
{
    public function index()
    {

        return view('posts.index', [
            'posts' => Post::latest()->filter(request(['search', 'category', 'author']))->paginate(5)->withQueryString(),
        ]);
    }

    public function show(Post $post)
    {
        return view('posts.show', [
            'post' => $post,
        ]);
    }
}
```



```php
class AdminPostController extends Controller
{
    public function index()
    {
        return view('admin.posts.index', [
            'posts' => Post::paginate(50)
        ]);
    }

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

    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('/');
    }

    public function edit(Post $post)
    {
        return view('admin.posts.edit', [
            'post' => $post
        ]);
    }

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

        if(isset($attributes['thumbnail'])) {
            $attributes['thumbnail'] = request()->file('thumbnail')->store('public/thumbnails');
            $attributes['thumbnail'] = str_replace('public/', '', $attributes['thumbnail']);
        }

        $post->update($attributes);

        return back()->with('success', 'Post Updated!');
    }

    public function destroy(Post $post)
    {
        $post->delete();

        return back()->with('success', 'Post Deleted!');
    }
```

68. Group and Store Validation Logic

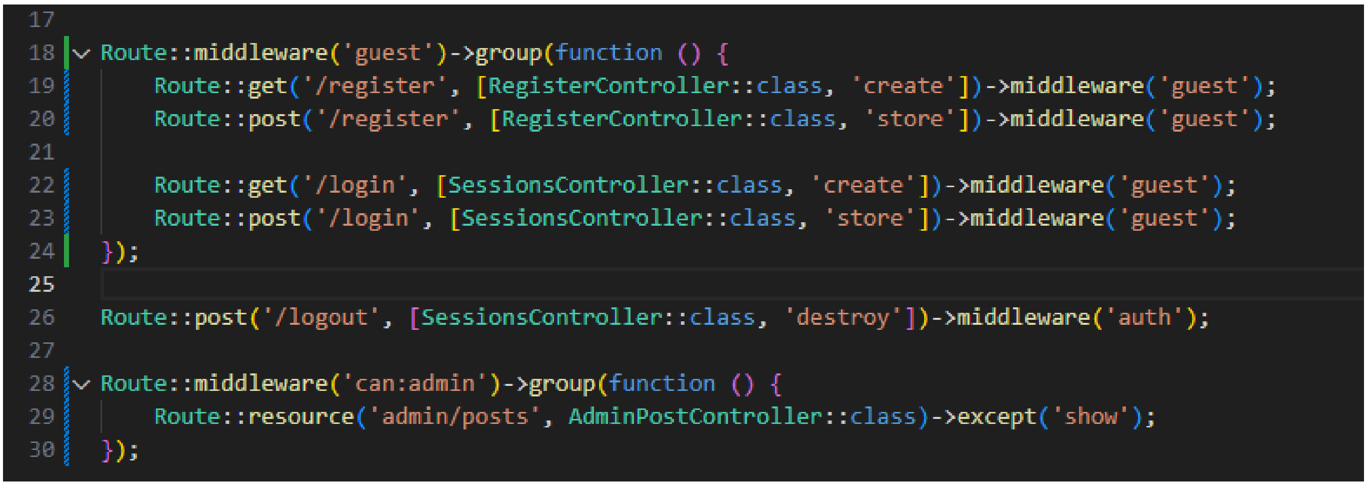
```
Ahora limpiamos un poco el Controlador de AdminPost para evitar duplicacion de
codigo
```



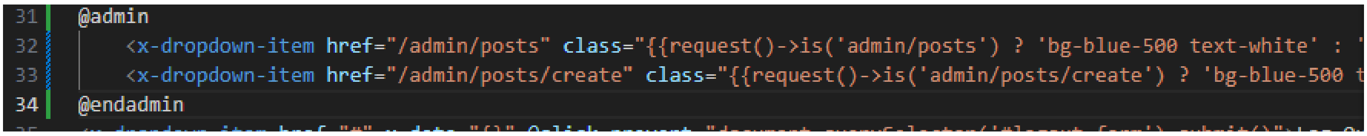
```php

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

    public function store()
    {
        $attributes = request()->validate([
            $post = new Post();
            $attributes = $this->validatePost($post);
        ]);

        $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('/');
    }

    public function edit(Post $post)
    {
        return view('admin.posts.edit', [
            'post' => $post
        ]);
    }

    public function update(Post $post)
    {
        $attributes = $this->validatePost($post);


        if(isset($attributes['thumbnail'])) {
            $attributes['thumbnail'] = request()->file('thumbnail')->store('public/thumbnails');
            $attributes['thumbnail'] = str_replace('public/', '', $attributes['thumbnail']);
        }

        $post->update($attributes);

        return back()->with('success', 'Post Updated!');
    }

    public function destroy(Post $post)
    {
        $post->delete();

        return back()->with('success', 'Post Deleted!');
    }

    protected function validatePost(Post $post): array
    {
        return request()->validate([
            'title' => 'required',
            'slug' => ['required', Rule::unique('posts', 'slug')->ignore($post)],
            'thumbnail' => $post->exists ? ['image'] : ['required', 'image'],
            'excerpt' => 'required',
            'body' => 'required',
            'category_id' => ['required', Rule::exists('categories', 'id')],
        ]);
    }
}
```

69. All About Authorization

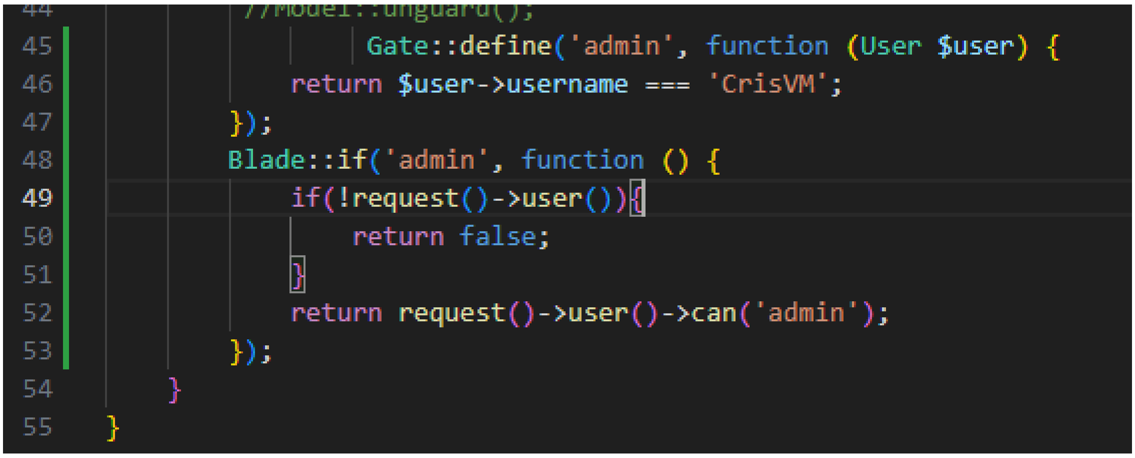
```
En este capitulo se soluciona un pequeño bug que hacia que todos los usuarios
pudieran ver las opciones de admnistrador, no tenian acceso, pero las veian
```



```php

Route::middleware('guest')->group(function () {
    Route::get('/register', [RegisterController::class, 'create'])->middleware('guest');
    Route::post('/register', [RegisterController::class, 'store'])->middleware('guest');

    Route::get('/login', [SessionsController::class, 'create'])->middleware('guest');
    Route::post('/login', [SessionsController::class, 'store'])->middleware('guest');
});

Route::post('/logout', [SessionsController::class, 'destroy'])->middleware('auth');

Route::middleware('can:admin')->group(function () {
    Route::resource('admin/posts', AdminPostController::class)->except('show');
});
```



```
@admin
    <x-dropdown-item href="/admin/posts" class="{{request()->is('admin/posts') ? 'bg-blue-500 text-white' : '
    <x-dropdown-item href="/admin/posts/create" class="{{request()->is('admin/posts/create') ? 'bg-blue-500 t
@endadmin
```



```php
            //Model::unguard();
                Gate::define('admin', function (User $user) {
            return $user->username === 'CrisVM';
        });
        Blade::if('admin', function () {
            if(!request()->user()){
                return false;
            }
            return request()->user()->can('admin');
        });
    }
}
```