Estudo de Caso

MyERP App

Thiago Olivier – UX Designer & Engineer

Introdução e Contexto

MyERP é um aplicativo simples, cujo objetivo é demonstrar minhas habilidades na construção de aplicativos baseados em API, que podem ser escalados e utilizados por diferentes tipos de aplicação cliente.

Principais Funcionalidades

- Sistema de autenticação robusto e seguro
- Visual moderno
- Boa experiência do usuário
- Operações CRUD

Tecnologias Utilizadas

API

Laravel (PHP)

Client

Vue.js App

Banco de dados

MySQL

O que eu aprendi?

Autenticação e CORS

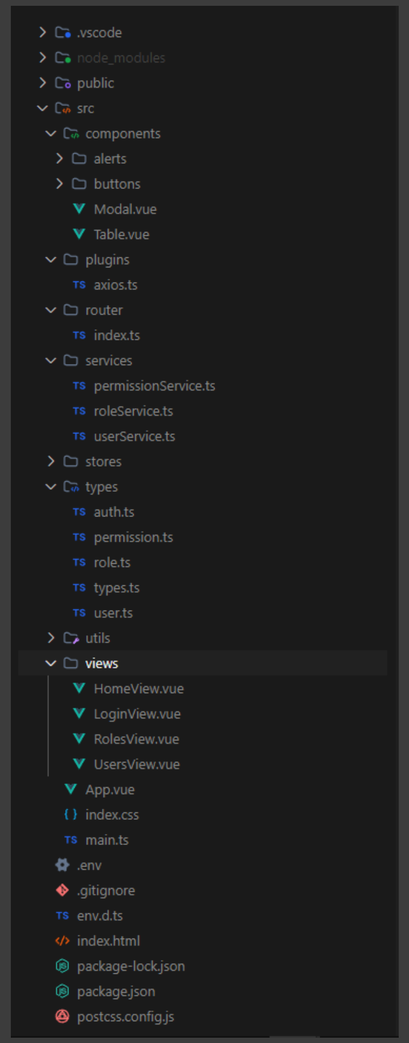
Tive a oportunidade de melhorar minhas habilidades sobre como é o fluxo de autenticação utilizando tokens JWT e cookies de sessão do Laravel, além de aprender o que é CORS e qual sua importância quando temos uma aplicação distribuida, ou que é consumida por diversas aplicações-cliente.

```php
<?php

return [
    /*
    |--------------------------------------------------------------------------
    | Cross-Origin Resource Sharing (CORS) Configuration
    |--------------------------------------------------------------------------
    |
    | Here you may configure your settings for cross-origin resource sharing
    | or "CORS". This determines what cross-origin operations may execute
    | in web browsers. You are free to adjust these settings as needed.
    |
    | To learn more: https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS
    |
    */

    'paths' => ['*'],

    'allowed_methods' => ['*'],

    'allowed_origins' => ['http://127.0.0.1:*', 'http://localhost:*'],

    'allowed_origins_patterns' => [],

    'allowed_headers' => ['*'],

    'exposed_headers' => [],

    'max_age' => 0,

    'supports_credentials' => true,
];
```

Reforçando os Conceitos CRUD

Revisar faz bem! Pude criar models e controllers seguindo as boas práticas do zero, desde a modelagem do banco de dados à construção do código, traduzindo os relacionamentos existentes no banco de dados para código, aproveitando a praticidade que o Laravel proporciona.

```php
<?php

namespace App\Models;

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

class Role extends Model
{
    use HasFactory;

    protected $fillable = [
        'name',
        'description',
    ];

    protected $hidden = [
        'created_at',
        'updated_at',
    ];

    public function permissions(): BelongsToMany
    {
        return $this->belongsToMany(Permission::class, 'role_permissions', 'role_id', 'permission_id');
    }

    public function users(): BelongsToMany
    {
        return $this->belongsToMany(User::class, 'user_roles', 'role_id', 'user_id');
    }
}
```

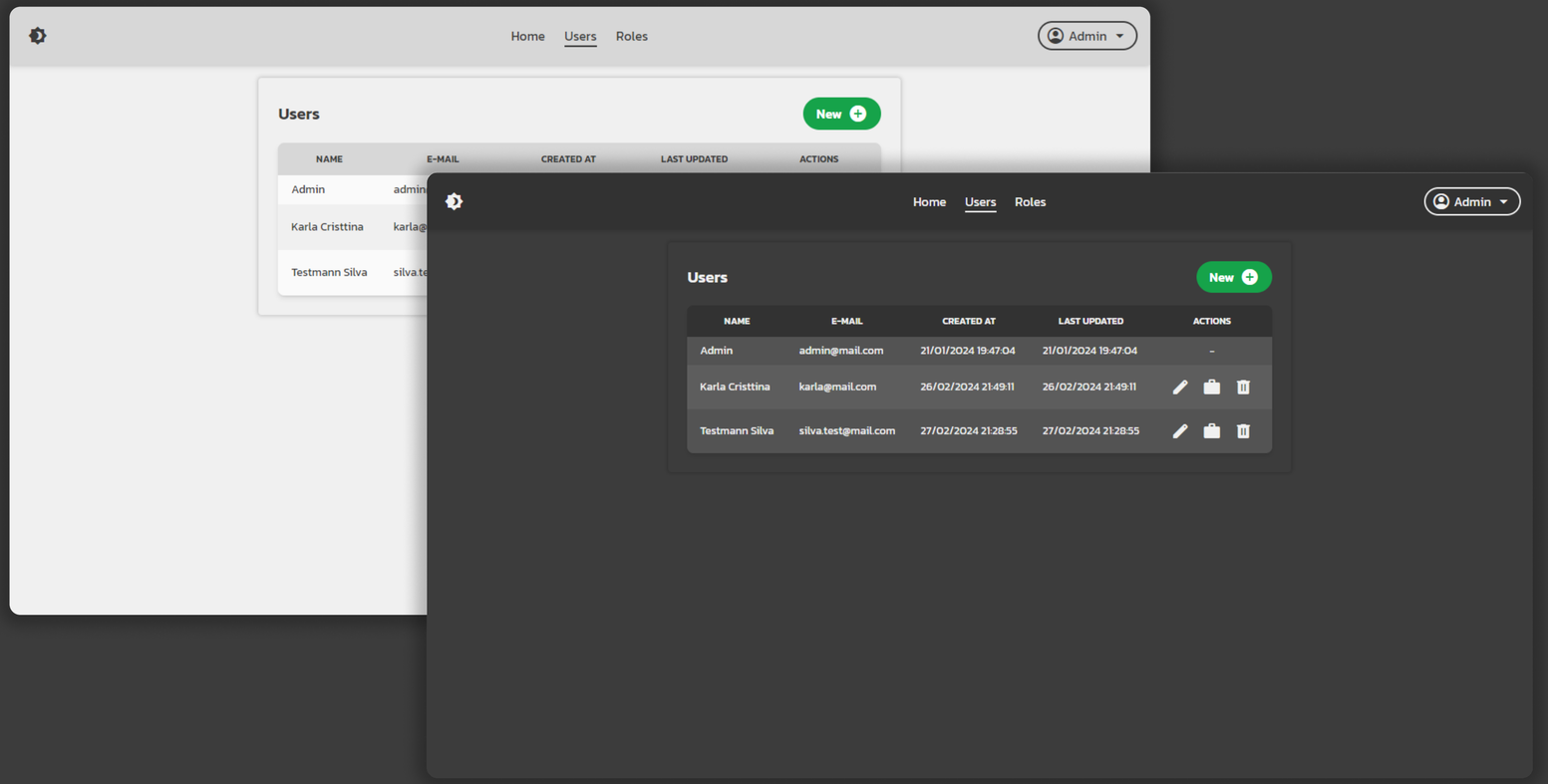


Criando Telas

Criei uma aplicação Vue.js + TypeScript para consumir a API Laravel, onde implementei:

- Lógica de autenticação
- Modularização de código
- Separação de responsabilidades
- UX/UI e darkmode com Tailwind CSS

Modo Dark


Home
Users
Roles
Admin
Users
New
NAME
E-MAIL
CREATED AT
LAST UPDATED
ACTIONS
Admin
admin@mail.com
21/01/2024 19:47:04
21/01/2024 19:47:04
-
Karla Cristtina
karla@mail.com
26/02/2024 21:49:11
26/02/2024 21:49:11
Testmann Silva
silva.test@mail.com
27/02/2024 21:28:55
27/02/2024 21:28:55

Conclusão

Criar uma aplicação inteira pode ser um desafio, mas que tem resultados extremamente valiosos.

Sobre o MyERP, a idéia foi não apenas demonstrar um CRUD básico, mas fazer de uma forma escalável. A aplicação base pode ser expandida conforme necessidade, adicionando novos módulos e componentes.

Veja mais!

API Laravel	https://github.com/thiagoolivier/erp_api
Cliente web Vue.js	https://github.com/thiagoolivier/erp_web
Portfólio	https://thiagoolivier.github.io/
Linkedin	https://www.linkedin.com/in/thiagoolivier/