

Case Study

MyERP App

Thiago Olivier – UX Designer & Engineer

Introduction and Contextualization

MyERP is a straightforward application designed to showcase my skills in building API-based apps that can scale and be utilized by various types of client applications.

Main Features

- Robust and safe authentication system
- Visually appealing
- Good UX
- CRUD operations

Adopted Technologies



API

Laravel (PHP)



Client

Vue.js App



Database

MySQL

What did i learn?

Authentication and CORS

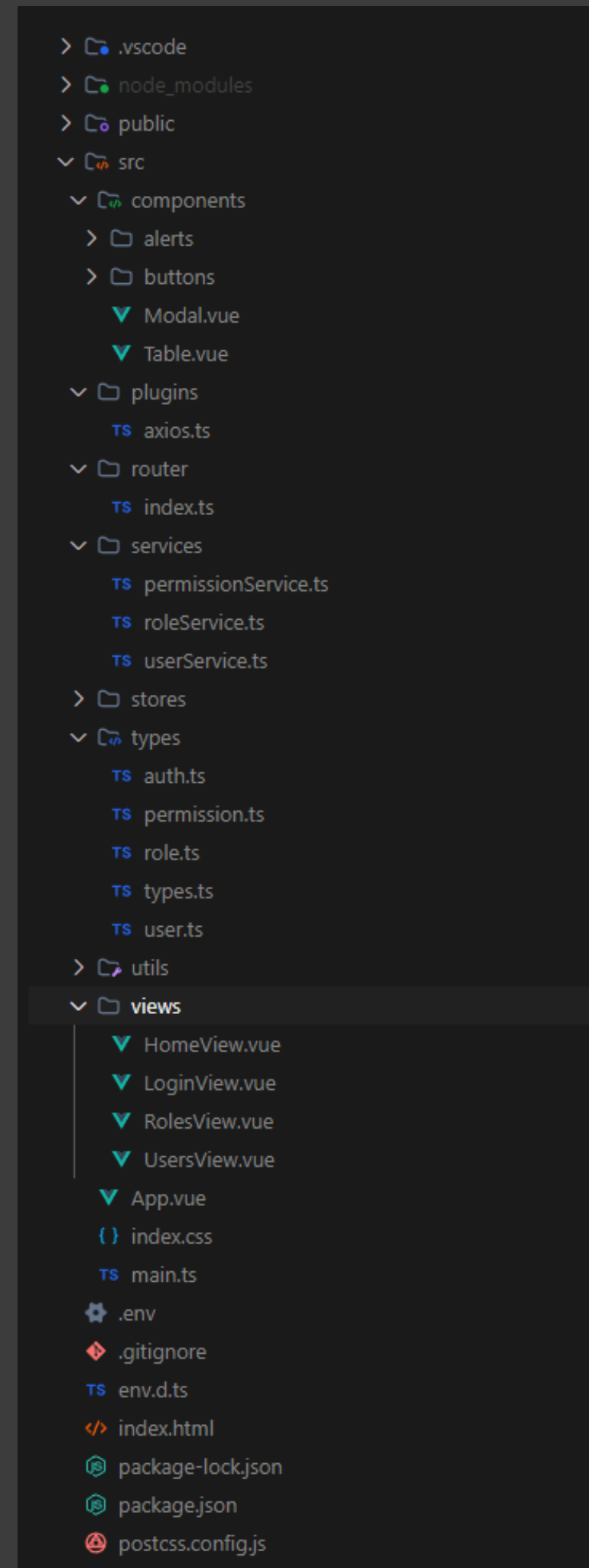
I had the opportunity to enhance my skills on the authentication flow using JWT tokens and Laravel session cookies, as well as to learn about CORS and its importance when dealing with a distributed application or one that is consumed by multiple client applications.

```
1 <?php
2
3 return [
4     /*
5      |-----
6      | Cross-Origin Resource Sharing (CORS) Configuration
7      |-----
8      |
9      | Here you may configure your settings for cross-origin resource sharing
10     | or "CORS". This determines what cross-origin operations may execute
11     | in web browsers. You are free to adjust these settings as needed.
12     |
13     | To learn more: https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS
14     |
15     */
16
17     'paths' => ['*'],
18
19     'allowed_methods' => ['*'],
20
21     'allowed_origins' => ['http://127.0.0.1:*', 'http://localhost:*'],
22
23     'allowed_origins_patterns' => [],
24
25     'allowed_headers' => ['*'],
26
27     'exposed_headers' => [],
28
29     'max_age' => 0,
30
31     'supports_credentials' => true,
32 ];
```

Reinforcing CRUD concepts

Reviewing is beneficial! I was able to create models and controllers following best practices from scratch, starting from database modeling to code construction, translating existing database relationships into code, leveraging the convenience provided by Laravel.

```
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7  use Illuminate\Database\Eloquent\Relations\BelongsToMany;
8
9  class Role extends Model
10 {
11     use HasFactory;
12
13     protected $fillable = [
14         'name',
15         'description',
16     ];
17
18     protected $hidden = [
19         'created_at',
20         'updated_at',
21     ];
22
23     public function permissions(): BelongsToMany
24     {
25         return $this->belongsToMany(Permission::class, 'role_permissions', 'role_id', 'permission_id');
26     }
27
28     public function users(): BelongsToMany
29     {
30         return $this->belongsToMany(User::class, 'user_roles', 'role_id', 'user_id');
31     }
32 }
```



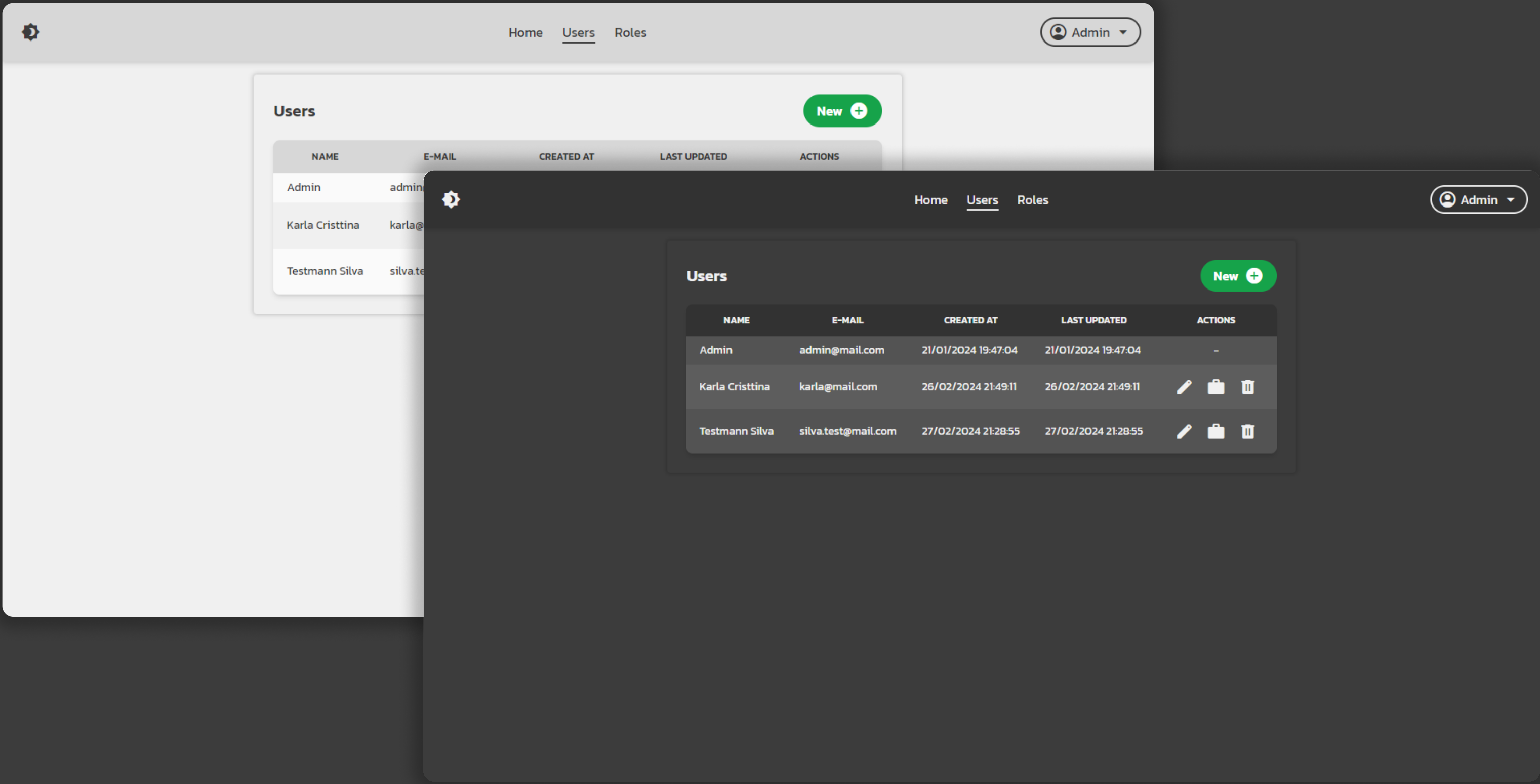
Building Views

I created a Vue.js + TypeScript application to consume the Laravel API, where I implemented:

- Authentication logic
- Code modularization
- Responsibility separation
- UX/UI and dark mode using Tailwind CSS

What did i learn?

Dark Mode



Conclusion

Building an entire application can be challenging, but it yields extremely valuable results.

Regarding MyERP, the idea was not only to demonstrate a basic CRUD but to do it in a scalable manner. The base application can be expanded as needed, adding new modules and components.

See More!

API Laravel

https://github.com/thiagoolivier/erp_api

Vue.js Web Client

https://github.com/thiagoolivier/erp_web

Portfolio

<https://thiagoolivier.github.io/>

LinkedIn

<https://www.linkedin.com/in/thiagoolivier/>