Install Laravel

composer create-project laravel/laravel:^10.0 example-app

## \*\*Install Tailwind CSS in vscode terminal

Install tailwindcss and its peer dependencies via npm, and then run the init command to generate both tailwind.config.js and postcss.config.js.

npm install -D tailwindcss postcss autoprefixer

The command `npm install -D tailwindcss postcss autoprefixer` installs three development dependencies for a project using npm (Node Package Manager). Let's break it down word by word:

- `npm`: Node Package Manager, a tool used to install packages and manage dependencies for JavaScript projects.

- `install`: The npm command to install one or more packages.

- `-D`: A flag that stands for `--save-dev`, indicating that the packages should be installed as development dependencies. These are packages needed only during the development process, not in the production environment.

- `tailwindcss`: A utility-first CSS framework for rapidly building custom user interfaces.

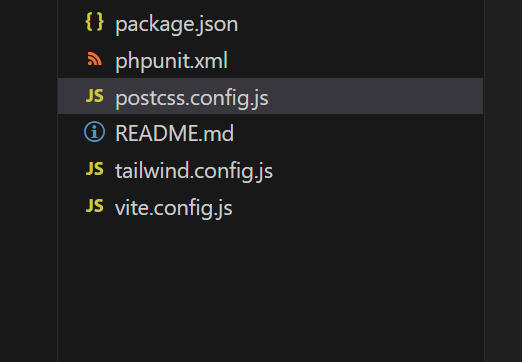
- `postcss`: A tool for transforming CSS with JavaScript plugins, used to process and optimize CSS files.

- `autoprefixer`: A PostCSS plugin that adds vendor prefixes to CSS rules using data from Can I Use, ensuring compatibility with different browsers.

So, the command installs `tailwindcss`, `postcss`, and `autoprefixer` as development dependencies in the project.

npx tailwindcss init -p

\*\*when that line is ran, tailwind.config.js file will appear. And also the post css file



## Configure your template paths

Add the paths to all of your template files in your tailwind.config.js file.

In this file copy this code :

import defaultTheme from 'tailwindcss/defaultTheme';

import forms from '@tailwindcss/forms';

/\*\* @type {import('tailwindcss').Config} \*/

export default {

    content: [

        './vendor/laravel/framework/src/Illuminate/Pagination/resources/views/\*.blade.php',

        './storage/framework/views/\*.php',

        './resources/views/\*\*/\*.blade.php',

    ],

    theme: {

        extend: {

            fontFamily: {

                sans: ['Figtree', ...defaultTheme.fontFamily.sans],

            },

        },

    },

    plugins: [forms],

};

So, we have customized the font that’s why we extend it

But this is the original code of that file

/\*

/\*\* @type {import('tailwindcss').Config} \*/

/\*module.exports = {

  content: [],

  theme: {

    extend: {},

  },

  plugins: [],

}

## \*\*Add the Tailwind directives to your CSS

Add the @tailwind directives for each of Tailwind’s layers to your ./resources/css/app.css file.

@tailwind base;

@tailwind components;

@tailwind utilities;

### \*\*To install the plugin, run the following command in your project directory:

npm install @tailwindcss/forms

### \*\*Build Your CSS

Run the build process to generate your Tailwind CSS file with the updated configuration:

This will process your CSS with the extended theme settings, and you should now see your custom font applied to the elements using the font-sans class.

npx tailwindcss build -o dist/output.css

npm run dev

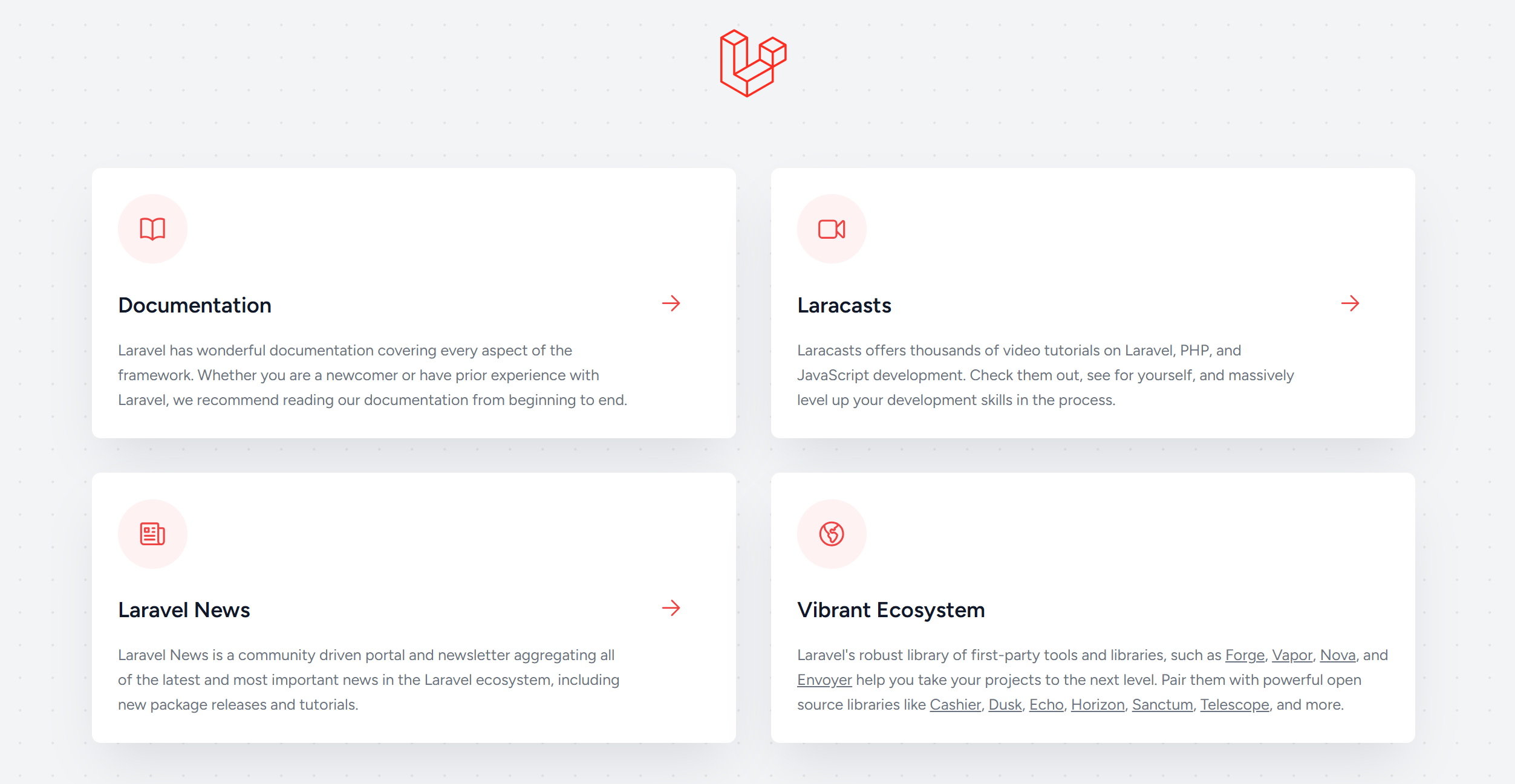
\*\*change the env in ‘filament\_tutorial’

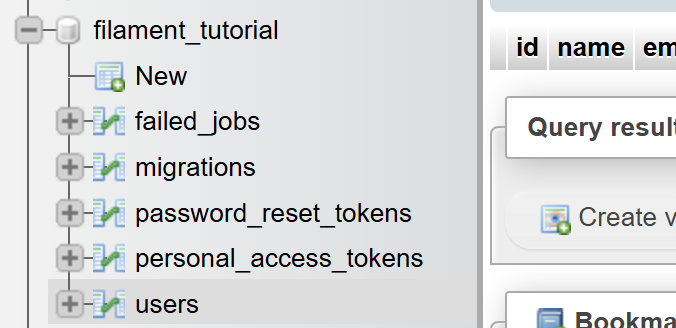
php artisan serve

npm run dev

php artisan migrate

\*\*this must be the result





Make a filament

\*\*visit this yt: [Laravel Install FilamentPHP - Filament V3 Tutorial (youtube.com)](https://www.youtube.com/watch?v=rN9XI9KCz0c&list=PL6tf8fRbavl3jfL67gVOE9rF0jG5bNTMi&index=1)

\*\*visit site of filament: [Installation - Panel Builder - Filament (filamentphp.com)](https://filamentphp.com/docs/3.x/panels/installation)

\*\*run this

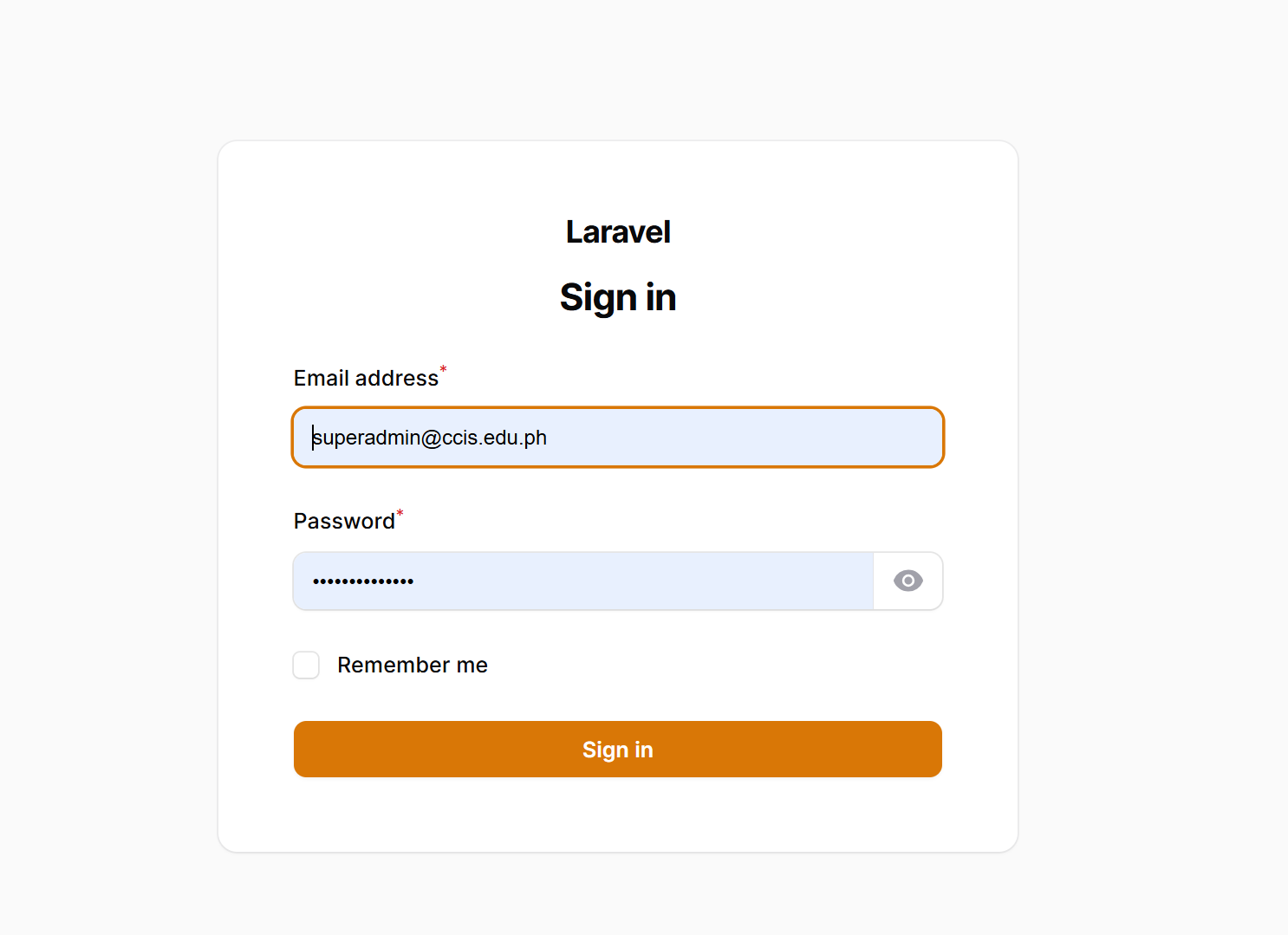
composer require filament/filament:"^3.2" -W

php artisan filament:install –panels

\*\*type id as superadmin

\*\*type no

\*\*by choosing superadmin as id, we can access the project as superadmin/login



\*\*now we don’t have user , in db , so we have to run this to create user

php artisan make:filament-user

name:Vicente Pitogo

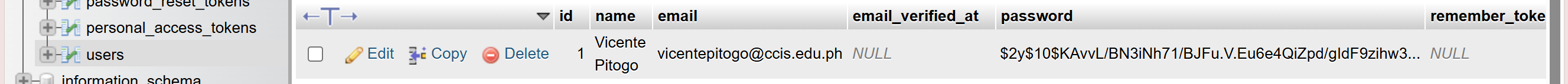
email:vicentepito@ccis.edu.ph

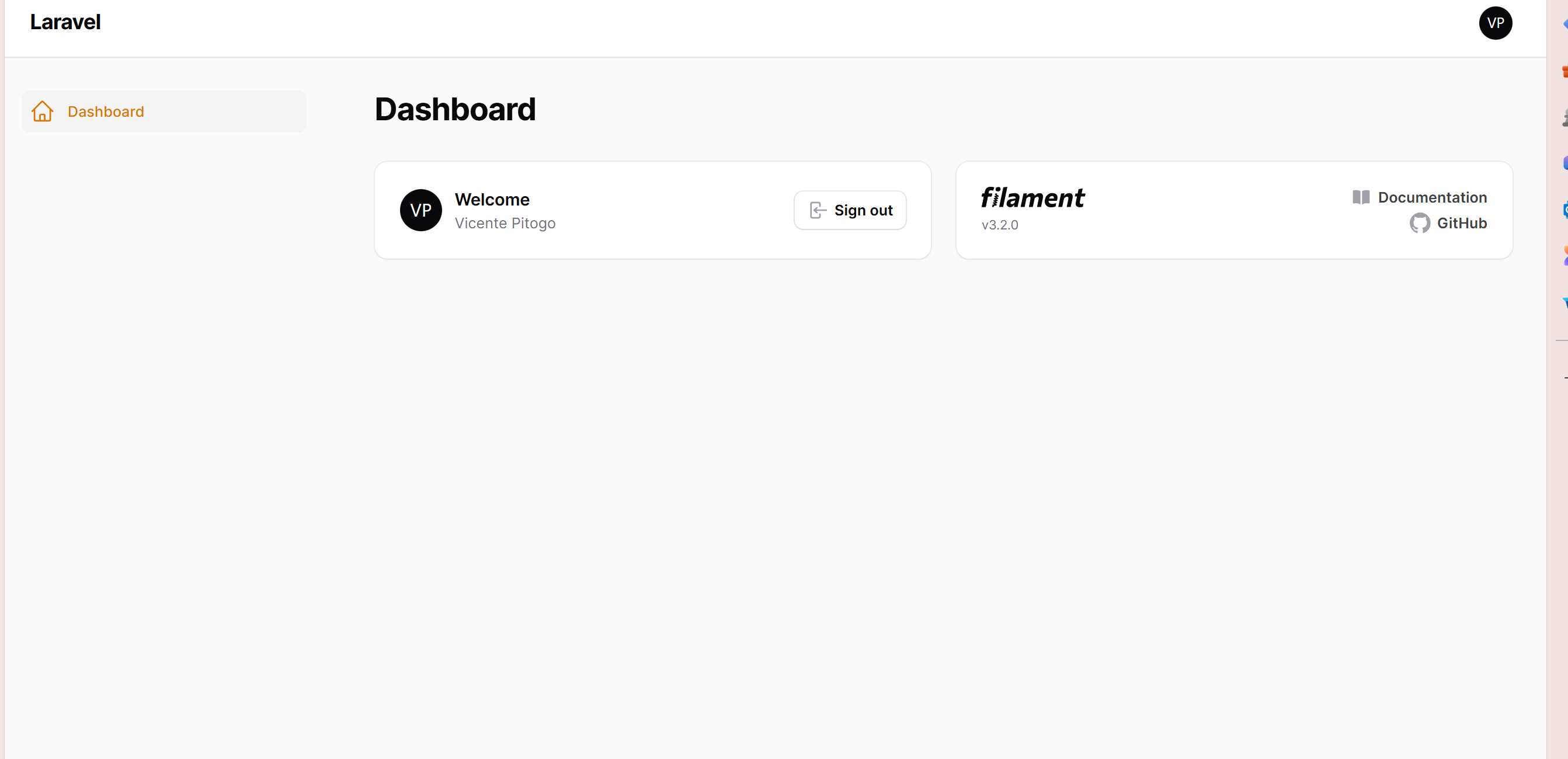
password:vicentepitogoccis

php artisan serve

superadmin/login

\*\*and login that account





**MAKING MODELS**

\*\*in models/user.php, paste this

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

return new class extends Migration

{

    /\*\*

     \* Run the migrations.

     \*/

    public function up(): void

    {

        Schema::create('users', function (Blueprint $table) {

            $table->id();

            $table->string('name');

            $table->string('email')->unique();

            $table->timestamp('email\_verified\_at')->nullable();

            $table->string('password');

            $table->rememberToken();

            $table->timestamps();

        });

    }

    /\*\*

     \* Reverse the migrations.

     \*/

    public function down(): void

    {

        Schema::dropIfExists('users');

    }

};

\*\*create migrations for facilities

php artisan make:migration create\_facilities\_table

\*\*in database/migrations/ create\_facilities, copy this code

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

return new class extends Migration

{

    /\*\*

     \* Run the migrations.

     \*/

    public function up(): void

    {

        Schema::create('facilities', function (Blueprint $table) {

            $table->id();

            $table->string('name');

            $table->string('connection\_type');

            $table->string('facility\_type');

            $table->string('cooling\_tools');

            $table->string('floor\_level');

            $table->string('building')->default('HIRAYA');

            $table->string('remarks');

            $table->string('facility\_img');

            $table->timestamps();

        });

    }

    /\*\*

     \* Reverse the migrations.

     \*/

    public function down(): void

    {

        Schema::dropIfExists('facilities');

    }

};

php artisan migrate

**RESOURCES IN FILAMENT-MIGRATION-RESOURCE-MODEL**

**Resource-dri makita ang mga panel sa side menu**

php artisan make:filament-resource Facility

php artisan make:filament-resource User

\*\*then go to app/filament/ , you can see UserResource and FacilityResource .php and UserResource and FacilityResource folder that contain pages :create, edit, list.

\*\*open the FacilityResource.php, u notice that the table and schema are empty

\*\*to fill the schema and colums customized , add –generatet in last

php artisan make:filament-resource Facility --generate

php artisan make:filament-resource User --generate

\*\*open FacilityResource.php , and copy this

<?php

namespace App\Filament\Resources;

use App\Filament\Resources\FacilityResource\Pages;

use App\Filament\Resources\FacilityResource\RelationManagers;

use App\Models\Facility;

use Filament\Forms;

use Filament\Forms\Form;

use Filament\Resources\Resource;

use Filament\Tables;

use Filament\Tables\Table;

use Illuminate\Database\Eloquent\Builder;

use Illuminate\Database\Eloquent\SoftDeletingScope;

use Filament\Forms\Components\Section;

use Filament\Forms\Components\Grid;

class FacilityResource extends Resource

{

    protected static ?string $model = Facility::class;

    protected static ?string $navigationIcon = 'heroicon-o-rectangle-stack';

    public static function form(Form $form): Form

    {

        return $form

            ->schema([

                Section::make('Facility Information')

                    ->schema([

                        Grid::make(2)

                            ->schema([

                                Forms\Components\TextInput::make('name')

                                    ->required()

                                    ->maxLength(255),

                                Forms\Components\TextInput::make('connection\_type')

                                    ->required()

                                    ->maxLength(255),

                                Forms\Components\TextInput::make('facility\_type')

                                    ->required()

                                    ->maxLength(255),

                                Forms\Components\TextInput::make('cooling\_tools')

                                    ->required()

                                    ->maxLength(255),

                                Forms\Components\TextInput::make('floor\_level')

                                    ->required()

                                    ->maxLength(255),

                                Forms\Components\TextInput::make('building')

                                    ->required()

                                    ->maxLength(255)

                                    ->default('HIRAYA'),

                            ]),

                    ]),

                Section::make('Facility Image')

                    ->schema([

                        Forms\Components\FileUpload::make('facility\_img')

                            ->image()

                            ->label('Facility Image')

                            ->imageEditor()

                            ->disk('public')

                            ->directory('facility'),

                    ]),

                Section::make('Remarks')

                    ->schema([

                        Forms\Components\RichEditor::make('remarks')

                            ->required()

                            ->disableToolbarButtons([

                                'attachFiles'

                            ])

                    ]),

            ]);

    }

    public static function table(Table $table): Table

    {

        return $table

            ->columns([

                Tables\Columns\TextColumn::make('name')

                    ->searchable(),

                Tables\Columns\TextColumn::make('connection\_type')

                    ->searchable(),

                Tables\Columns\TextColumn::make('facility\_type')

                    ->searchable(),

                Tables\Columns\TextColumn::make('cooling\_tools')

                    ->searchable(),

                Tables\Columns\TextColumn::make('floor\_level')

                    ->searchable(),

                Tables\Columns\TextColumn::make('building')

                    ->searchable(),

                Tables\Columns\TextColumn::make('remarks')

                    ->searchable(),

                Tables\Columns\TextColumn::make('facility\_img')

                    ->searchable(),

                Tables\Columns\TextColumn::make('created\_at')

                    ->dateTime()

                    ->sortable()

                    ->toggleable(isToggledHiddenByDefault: true),

                Tables\Columns\TextColumn::make('updated\_at')

                    ->dateTime()

                    ->sortable()

                    ->toggleable(isToggledHiddenByDefault: true),

            ])

            ->filters([

                //

            ])

            ->actions([

                Tables\Actions\EditAction::make(),

            ])

            ->bulkActions([

                Tables\Actions\BulkActionGroup::make([

                    Tables\Actions\DeleteBulkAction::make(),

                ]),

            ]);

    }

    public static function getRelations(): array

    {

        return [

            //

        ];

    }

    public static function getPages(): array

    {

        return [

            'index' => Pages\ListFacilities::route('/'),

            'create' => Pages\CreateFacility::route('/create'),

            'edit' => Pages\EditFacility::route('/{record}/edit'),

        ];

    }

}

\*\*open UserResource.php , and copy this

<?php

namespace App\Filament\Resources;

use App\Filament\Resources\UserResource\Pages;

use App\Filament\Resources\UserResource\RelationManagers;

use App\Models\User;

use Filament\Forms;

use Filament\Forms\Form;

use Filament\Resources\Resource;

use Filament\Tables;

use Filament\Tables\Table;

use Illuminate\Support\Str;

use Illuminate\Support\Facades\Hash;

use Illuminate\Database\Eloquent\Builder;

use Illuminate\Database\Eloquent\SoftDeletingScope;

use Filament\Forms\Components\Section;

class UserResource extends Resource

{

    protected static ?string $model = User::class;

    protected static ?string $navigationGroup = 'User Management';

    protected static ?string $activeNavigationIcon = 'heroicon-o-user-group';

    protected static ?string $navigationIcon = 'heroicon-o-rectangle-stack';

    public static function form(Form $form): Form

    {

        return $form

            ->schema([

                Section::make()

                    ->schema([

                        Forms\Components\TextInput::make('name')

                            ->required()

                            ->maxLength(255),

                        Forms\Components\TextInput::make('email')

                            ->email()

                            ->unique(ignoreRecord: true),

                        Forms\Components\Select::make('roles')

                            ->relationship('roles', 'name')

                            ->preload()

                            ->searchable(),

                        Forms\Components\TextInput::make('password')

                            ->password()

                            ->required()

                            ->revealable()

                            ->dehydrateStateUsing(fn ($state) => Hash::make($state))

                            ->hiddenOn('edit')

                    ])

            ]);

    }

    public static function table(Table $table): Table

    {

        return $table

            ->columns([

                Tables\Columns\TextColumn::make('name')

                    ->searchable(),

                Tables\Columns\TextColumn::make('email')

                    ->searchable(),

                Tables\Columns\TextColumn::make('roles.name')->label('Role')

                    ->formatStateUsing(fn ($state): string => Str::headline($state))

                    ->colors(['info'])

                    ->badge(),

            ])

            ->filters([

                //

            ])

            ->actions([

                Tables\Actions\EditAction::make(),

            ])

            ->bulkActions([

                Tables\Actions\BulkActionGroup::make([

                    Tables\Actions\DeleteBulkAction::make(),

                ]),

            ]);

    }

    public static function getRelations(): array

    {

        return [

            //

        ];

    }

    public static function getPages(): array

    {

        return [

            'index' => Pages\ListUsers::route('/'),

            'create' => Pages\CreateUser::route('/create'),

            'edit' => Pages\EditUser::route('/{record}/edit'),

        ];

    }

}

\*\*make a model for facility

php artisan make:model Facility

\*\*then open Models/Facility.php, paste this

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

class Facility extends Model

{

    use HasFactory;

    protected $fillable = [

        'name',

        'connection\_type',

        'facility\_type',

        'cooling\_tools',

        'floor\_level',

        'building',

        'remarks',

        'facility\_img',

    ];

}

\*\*the result must be

