**Autentikasi Multi-user dengan Laravel UI**

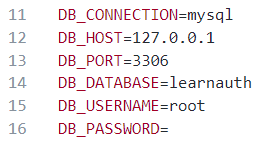
**Membuat Project Laravel**

Gunakan perintah berikut pada CMD / Terminal:

$ **composer create-project --prefer-dist laravel/laravel multiuser-auth**

**Konfigurasi tabel users**

Buka konfigurasi database pada file **.env** sebagai berikut:



Kemudian buka file migration pada **database/migrations/create\_users** dan tuliskan kodenya berikut:

    public function up()

    {

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

            $table->id();

            $table->string('name');

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

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

            $table->string('password');

**// role: 0 ~> user, 1 ~> admin, 2 ~> manager**

**$table->tinyInteger('role')->default(0);**

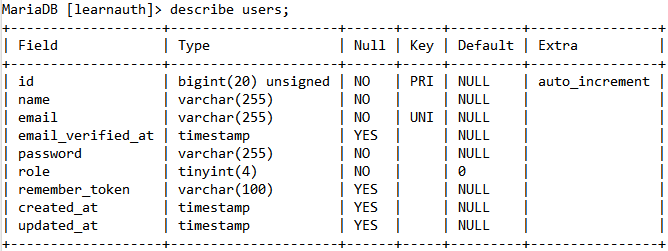
            $table->rememberToken();

            $table->timestamps();

        });

    }

Kemudian jalankan file migration diatas dengan perintah $ **php artisan migrate:fresh**



Maka sebuah tabel **users** muncul dengan struktur tabel seperti gambar diatas.

**Konfigurasi File Migration dan Model**

Buka file model pada direktori **app/Models/Users.php** dan ubah kode didalamnya:

..........

**use Illuminate\Database\Eloquent\Casts\Attribute;**

class User extends Authenticatable {

    use HasApiTokens, HasFactory, Notifiable;

    protected $fillable = [

        'name',

        'email',

        'password',

**'role',**

    ];

**protected function role(): Attribute {**

**return new Attribute(**

**get: fn ($value) => ["user", "admin", "manager"][$value],**

**);**

**}**

..........

**Membuat Authentikasi dengan Scaffolding UI**

Scaffolding UI telah disediakan oleh Laravel, gunakan perintah dibawah untuk menggunakannya:

$ **composer require laravel/ui**

$ **php artisan ui bootstrap --auth**

Kemudian untuk menjalankannya dengan perintah berikut:

$ **npm install**

$ **npm run dev**



**Membuat Middleware Akses Setiap User**

Gunakan perintah dibawah untuk membuat sebuah middleware yang baru:

$ **php artisan make:middleware UserAccess**

Buka file **UserAccess.php** dan tuliskan kode didalamnya:

**class UserAccess {**

**public function handle(Request $request, Closure $next, $userRole) {**

**if (auth()->user()->role == $userRole) {**

**return $next($request);**

**}**

**return response()->json(["You don't have permission access this page."]);**

**// return response()->view('errors.check-permission');**

**}**

**}**

Kemudian daftarkan UserAccess diatas pada **app/Http/Kernel.php**:

    protected $routeMiddleware = [

..........

        'verified' => \Illuminate\Auth\Middleware\EnsureEmailIsVerified::class,

**'user-access' => \App\Http\Middleware\UserAccess::class,**

    ];

**Membuat Perutean**

Buka file **routes/Web.php** dan buat perutean baru sebagai berikut:

<?php

use Illuminate\Support\Facades\Route;

**use App\Http\Controllers\HomeController;**

**use Illuminate\Support\Facades\Auth;**

Route::get('/', function () {

    return view('welcome');

});

Auth::routes();

// Route untuk User

**Route::middleware(['auth', 'user-access:user'])->group(function () {**

**Route::get('/home', [HomeController::class, 'index'])->name('home');**

**});**

// Route untuk Admin

**Route::middleware(['auth', 'user-access:admin'])->group(function () {**

**Route::get('/admin/home', [HomeController::class, 'adminHome'])->name('admin.home');**

**});**

// Route untuk Manager

**Route::middleware(['auth', 'user-access:admin'])->group(function () {**

**Route::get('/manager/home', [HomeController::class, 'managerHome'])->name('manager.home');**

**});**

**Mengkonfigurasi HomeController**

Buka file **app/Http/Controllers/HomeController.php** dan ubah kodenya sebagai berikut:

class HomeController extends Controller {

    public function \_\_construct() {

        $this->middleware('auth');

    }

    public function index() {

        return view('home');

    }

**public function adminHome() {**

**return view('adminHome');**

**}**

**public function managerHome(){**

**return view('managerHome');**

**}**

}

**Membuat View untuk Semua User**

Buat sebuah file blade baru pada **resources/views/adminHome.blade.php**:

@extends('layouts.app')

@section('content')

    <div class="container">

        <div class="row justify-content-center">

            <div class="col-md 8">

                <div class="card">

                    <div class="card-header">{{ \_\_('Dashboard') }}</div>

                    <div class="card-body">

**This is Admin page.**

                    </div>

                </div>

            </div>

        </div>

    </div>

@endsection

Buat sebuah file blade baru pada **resources/views/managerHome.blade.php**:

@extends('layouts.app')

@section('content')

    <div class="container">

        <div class="row justify-content-center">

            <div class="col-md 8">

                <div class="card">

                    <div class="card-header">{{ \_\_('Dashboard') }}</div>

                    <div class="card-body">

**This is Manager page.**

                    </div>

                </div>

            </div>

        </div>

    </div>

@endsection

**Mengkonfigurasi LoginController**

Buka file **app/Http/Controllers/Auth/LoginController.php** dan tambahkan kode berikut:

    public function login(Request $request)

    {

        $input = $request->all();

        $this->validate($request, [

            'email' => 'required|email',

            'password' => 'required',

        ]);

        if (auth()->attempt(array('email' => $input['email'],

'password' => $input['password']))) {

            if (auth()->user()->role == 'admin') {

                return redirect()->route('admin.home');

            } else if (auth()->user()->role == 'manager') {

                return redirect()->route('manager.home');

            } else {

                return redirect()->route('home');

            }

        } else {

            return redirect()->route('login')->

with('error', 'Email or Password is wrong!');

        }

    }

**Membuat Seeder untuk Isian Tabel Users**

Buat sebuah seeder baru dengan perintah dibawah ini:

$ **php artisan make:seeder CreateUsersSeeder**

Buka file **database/seeders/CreateUsersSeeder.php** dan tambahkan kode berikut:

..........

use App\Models\User;

class CreateUsersSeeder extends Seeder {

    public function run() {

        $users = [

            [

                'name' => 'Admin',

                'email' => 'admin@example.com',

                'role' => 1,

                'password' => bcrypt('Pa$$w0rd'),

            ],

            [

                'name' => 'Manager',

                'email' => 'manager@example.com',

                'role' => 2,

                'password' => bcrypt('Pa$$w0rd'),

            ],

            [

                'name' => 'User',

                'email' => 'user@example.com',

                'role' => 0,

                'password' => bcrypt('Pa$$w0rd'),

            ],

        ];

        foreach ($users as $key => $user) {

            User::create($user);

        }

    }

}

Kemudian jalankan seeder diatas agar mengisi tabel users dengan perintah:

$ **php artisan db:seed --class=CreateUsersSeeder**

**Melihat tabel users**

