Multi auth, multi guard with multi table

Laravel UI

Basic auth to multi auth feature

Three types of auth

1. Simple auth
2. Multi auth- multiple table used for auth
3. Role based auth – like blog different wordpress users, like admin, editor, author

**Example first Simple Auth and configure in .env**

Create db- custom\_auth\_laravel

C:\xampp\htdocs>**composer global require laravel/installer**

C:\xampp\htdocs>**laravel new authwebsite**

C:\xampp\htdocs\authwebsite>**php artisan serve**

C:\xampp\htdocs\authwebsite>**php artisan make:controller MainController**

Controller created successfully.

class MainController extends Controller

{

    public function login()

    {

        return view("auth.login");

    }

 public function register(Request $req)

    {

        return view("auth.regsiter");

    }

public function save(Request $req)

    {

        $req->validate([

            "name"=>"required",

            "email"=>"required | email",

            "password"=>"required | password",

        ]);

        return $req->input();

        //return view("auth.regsiter");

    }

}

use App\Http\Controllers\MainController;

Route::get('/auth/login',[MainController::class,'login'])->name('login');

Route::get('/auth/regsiter',[MainController::class,'register'])->name('register');

Route::post('/auth/save',[MainController::class,'save'])->name('save');

**Views/auth/login.blade.php and Views/auth/register.blade.php**

**Create form and**

**Create model and migration in one command**

C:\xampp\htdocs\authwebsite>**php artisan make:model Admin -m**

**Model created successfully.**

**Created Migration: 2021\_05\_04\_101250\_create\_admins\_table**

**C:\xampp\htdocs\authwebsite\database\migrations\2021\_05\_04\_101250\_create\_admins\_table.php**

    public function up()

    {

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

            $table->id();

             $table->string('name');

             $table->string('email');

             $table->string('password');

            $table->timestamps();

        });

C:\xampp\htdocs\authwebsite>**php artisan migrate**

Migration table created successfully.

**Now table Admin created but our email id value should be unique for this validation we need to change little bit**

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Models\Admin;

use Illuminate\Support\Facades\Hash;

use Session;

class MainController extends Controller

{

    public function loginnow(Request $req)

    {

        $req->validate([

            "email"=>"required|email",

            "password"=>"required",

        ]);

        $admin=Admin::where("email","=",$req->email)->first();

        if(!$admin)

        {

        return back()->with('fail','no record found');

        }

        else

        {

            if(Hash::check($req->password,$admin->password))

            {

                 Session::put('uid',$admin->id);

                 return redirect('/admin/dashboard');

            }

            else

            {

                return back()->with('fail','wrong password entered');

            }

        }

    }

    public function register()

    {

        return view("auth.regsiter");

    }

    public function save(Request $req)

    {

        $req->validate([

            "name"=>"required",

            "email"=>"required|email| unique:admins",

            "password"=>"required",

        ]);

        $data=new Admin;

        $data->name=$req->name;

        $data->email=$req->email;

        $data->password=Hash::make($req->password);

        $save=$data->save();

        if($save)

        {

            return back()->with('success','successfuly registered');

        }

        else

        {

            return back()->with('fail','unable to register');

        }

    }

}

<form action="{{route('save')}}" method="post">

@if(Session::get('success'))

<div class="alert alert-success">{{Session::get('success')}}</div>

@endif

@if(Session::get('fail'))

<div class="alert alert-fail">{{Session::get('fail')}}</div>

@endif

Now make login enable

<h2>Login Form</h2>

@if(Session::get('fail'))

<div class="alert alert-fail">{{Session::get('fail')}}</div>

@endif

<form action="{{route('loginnow')}}" method="post">

    @csrf

<http://127.0.0.1:8000/admin/dashboard>

C:\xampp\htdocs\authwebsite\resources\views\admin\dashboard.blade.php

**Create dashboard page**

<h1>hello</h1>

Route::view('/admin/dashboard','admin.dashboard')->name('dashboard');

**Admin/dashboard.blade.php**

<div class="jumbotron text-center">

    @if(Session::get('adminuser'))

  <h2>Welcome to Admin

      {{Session::get('adminuser')->name}}

</h2>

  <p><a href="/admin/logout">logout</a></p>

  @endif

</div>

**Logout functionality**

Route::get('/admin/logout',[MainController::class,'logout'])->name('logout');

public function logout()

    {

        Session::pull('adminuser');

        return redirect('/auth/login');

    }

**After logout we can access the dashboard without login too**

**For this we have to create middleware**

C:\xampp\htdocs\authwebsite>**php artisan make:middleware DashboardAccessCheck**

**Middleware created successfully.**

**C:\xampp\htdocs\authwebsite\app\Http\Middleware\DashboardAccessCheck.php**

 public function handle(Request $request, Closure $next)

    {

        echo "hi";

        return $next($request);

    }

**Register this middleware into kernel.php**

**Key should be middleware name and value should be path class**

protected $routeMiddleware = [

'DashboardAccessCheck' => \App\Http\Middleware\DashboardAccessCheck::class,]

**After registration when we browse any pages shows hi on the top, but when define middleware in routemiddleware it will not**

Now question is how we can access this middleware,answer is route with group

Route::group(

['middleware'=>

['DashboardAccessCheck']

], function(){

});

**Now we have pass which route which we wants to check this DashboardAccess middleware**

<?php

use Illuminate\Support\Facades\Route;

Route::get('/', function () {

    return view('welcome');

});

use App\Http\Controllers\MainController;

Route::post('/auth/login', [MainController::class, 'loginnow'])->name('loginnow');

Route::post('/auth/save', [MainController::class, 'save'])->name('save');

Route::get('/admin/logout', [MainController::class, 'logout'])->name('logout');

Route::group(['middleware' => ['DashboardAccessCheck']], function () {

    Route::view('/auth/login', 'auth.login')->name('loginnow');

    Route::get('/auth/regsiter', [MainController::class, 'register'])->name('register');

    Route::view('/admin/dashboard', 'admin.dashboard')->name('dashboard');

});

**NOW** echo "hi"; WILL SHOW HI ONLY Above three ROUTES ONLY

**DashboardCheck.php**

<?php

namespace App\Http\Middleware;

use Closure;

use Illuminate\Http\Request;

class DashboardAccessCheck

{

    /\*\*

     \* Handle an incoming request.

     \*

     \* @param  \Illuminate\Http\Request  $request

     \* @param  \Closure  $next

     \* @return mixed

     \*/

    public function handle(Request $request, Closure $next)

    {

        //means no session

        // no login page

        // no regsiter page

        if(!session('adminuser') && ($request->path()!=="auth/login") && ($request->path()!=="auth/register"))

        {

            return redirect('/auth/login');

        }

        // means we are logged into dashboard page and use login or register page it will redirect to same page on dashboard page

        if(session('adminuser') && ($request->path()=="auth/login") || ($request->path()=="auth/register"))

        {

            return back();

        }

        //Laravel Prevent Browser Back Button After Logout, so that he will not be able to access auth protected routes or pages after logout.

        return $next($request)->header('Cache-Control','nocache, no-store, max-age=0, must-revalidate')

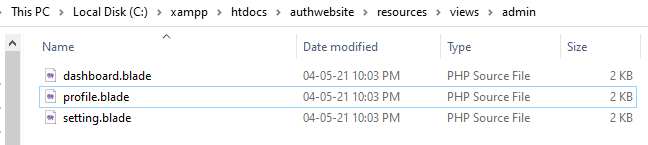
        ->header('Pragma','no-cache')

        ->header('Expires','Sat, 26 Jul 1997 05:00:00 GMT');

    }

}

**Lets add more dashboard like pages with DashboardCheck auth**

****

Route::group(['middleware' => ['DashboardAccessCheck']], function () {

    Route::view('/auth/login', 'auth.login')->name('loginnow');

    Route::get('/auth/regsiter', [MainController::class, 'register'])->name('register');

    Route::view('/admin/dashboard', 'admin.dashboard')->name('dashboard');

    Route::view('/admin/setting', 'admin.setting')->name('setting');

    Route::view('/admin/profile', 'admin.setting')->name('profile');

});