

Multi Authentication System Using Model in LARAVEL

Single user Authentication –

Commands :-

```
composer require laravel/breeze --dev
```

```
php artisan breeze:install
```

```
> blade
```

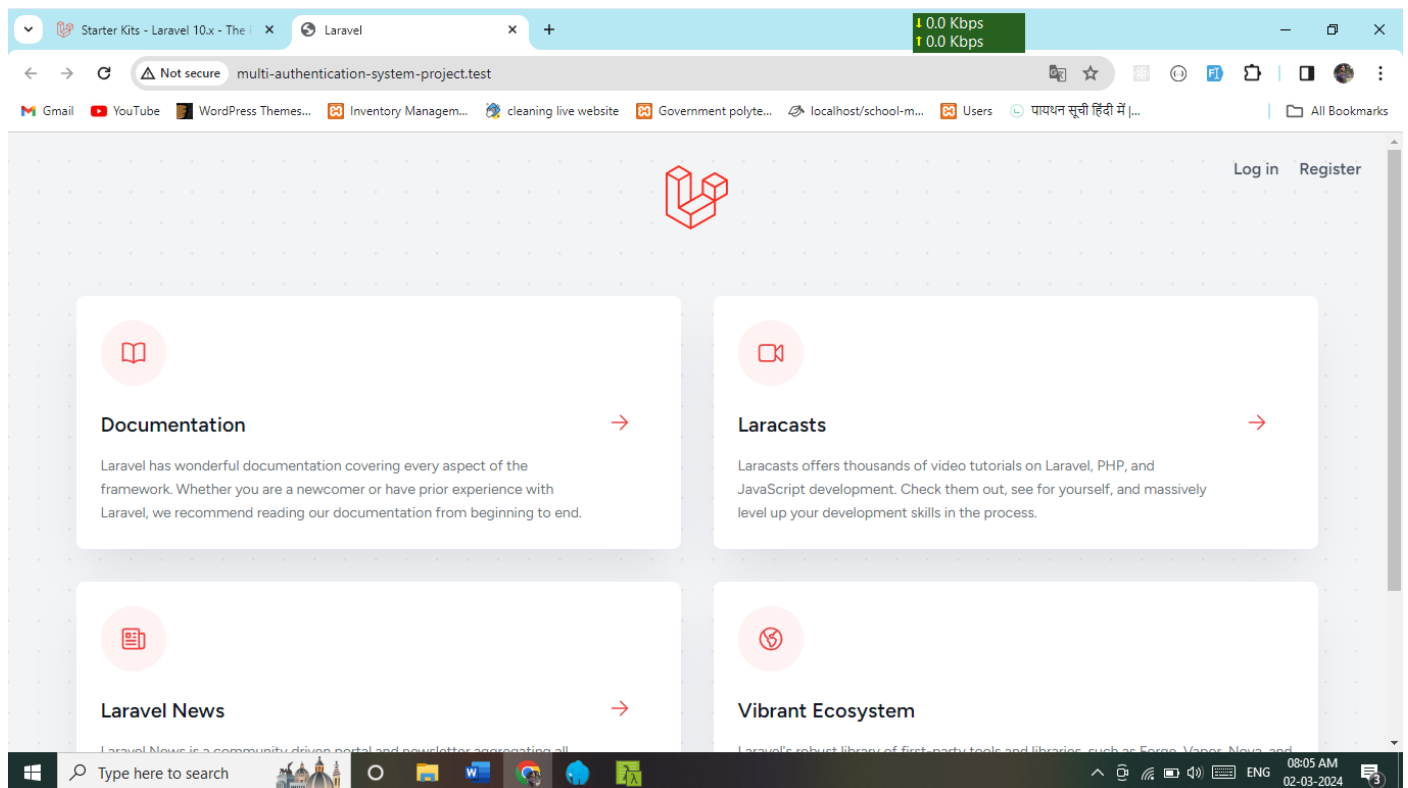
```
> phpUnit
```

```
php artisan migrate
```

Some Optional Command -

```
npm install
```

```
npm run dev
```



Multi Authentication (Admin) –

Commands : -

```
composer require laravel/breeze --dev
```

```
php artisan breeze:install
```

```
php artisan make:model Admin -m
```

 (-m means also create migration file)

Migration

Then copy all migration tables code users migration file to admins migrations file

Users migration file 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('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');
    }
};
```

After copy in admins migration file –

```
<?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('admins', 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('admins');
    }
};
```

Admin Model code –

Copy all codes in the user Model file – **User.php** to **Admin.php** Model file

User.php

```
<?php

namespace App\Models;

// use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;
use Laravel\Sanctum\HasApiTokens;

class User extends Authenticatable
{
    use HasApiTokens, HasFactory, Notifiable;

    /**
     * The attributes that are mass assignable.
     *
     * @var array<int, string>
     */
    protected $fillable = [
        'name',
        'email',
        'password',
    ];

    /**
     * The attributes that should be hidden for serialization.
     *
     * @var array<int, string>
     */
    protected $hidden = [
        'password',
        'remember_token',
    ];

    /**
     * The attributes that should be cast.
     */
}
```

```

    *
    * @var array<string, string>
    */
    protected $casts = [
        'email_verified_at' => 'datetime',
        'password' => 'hashed',
    ];
}

```

Admin.php

```

<?php

namespace App\Models;

// use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;
use Laravel\Sanctum\HasApiTokens;

class Admin extends Authenticatable
{
    use HasApiTokens, HasFactory, Notifiable;

    protected $guard = 'admin';

    /**
     * The attributes that are mass assignable.
     *
     * @var array<int, string>
     */
    protected $fillable = [
        'name',
        'email',
        'password',
    ];

    /**
     * The attributes that should be hidden for serialization.
     *
     * @var array<int, string>
     */
}

```

```

    */

    protected $hidden = [

        'password',

        'remember_token',

    ];

    /**
     * The attributes that should be cast.
     *
     * @var array<string, string>
     */
    protected $casts = [

        'email_verified_at' => 'datetime',

        'password' => 'hashed',

    ];
}

```

Config code –

auth.php

```

<?php

return [

    'defaults' => [

        'guard' => 'web',

        'passwords' => 'users',

    ],

    'guards' => [

        'web' => [

            'driver' => 'session',

            'provider' => 'users',

        ],

        'admin' => [

            'driver' => 'session',

```

```
        'provider' => 'admins',
    ],
],

'providers' => [
    'users' => [
        'driver' => 'eloquent',
        'model' => App\Models\User::class,
    ],
    'admins' => [
        'driver' => 'eloquent',
        'model' => App\Models\Admin::class,
    ],

    // 'users' => [
    //     'driver' => 'database',
    //     'table' => 'users',
    // ],
],
```

```
'passwords' => [
    'users' => [
        'provider' => 'users',
        'table' => 'password_reset_tokens',
        'expire' => 60,
        'throttle' => 60,
    ],
    'admins' => [
        'provider' => 'admins',
        'table' => 'password_reset_tokens',
        'expire' => 60,
        'throttle' => 60,
    ],
],
```

```
'password_timeout' => 10800,
```

```
];
```

Admin Controller code –

Commands : -

php artisan make:controller AdminController

AdminController.php code-

```
<?php

namespace App\Http\Controllers;

use App\Models\Admin;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class AdminController extends Controller
{
    public function index()
    {
        return view('auth.admin.login');
    }

    public function login(Request $request)
    {
        if(Auth::guard('admin')->attempt(['email'=>$request->email, 'password'=>$request->password]))
        {
            return redirect()->route('admin.dashboard')->with('success', 'Login Successfull');
        }
        else
        {
            return redirect()->back()->with('error', 'Invalid Email Or Password');
        }
    }

    public function dashboard()
    {
        return view('admin-dashboard');
    }

    public function logout()
```



```

{
    Auth::guard('admin')->logout();

    // $request->session()->invalidate();

    // $request->session()->regenerateToken();

    return redirect()->route('admin.login.index');
}

public function registerIndex()
{
    return view('auth.admin.register');
}

public function register(Request $request)
{
    $request->validate([
        'name' =>['required', 'string', 'max:255'],
        'email' =>['required', 'email', 'max:255', 'unique:admins'],
        'password' =>['required', 'string', 'max:8', 'confirmed']
    ]);

    $admin = new Admin();

    $admin->name = $request->name;
    $admin->email = $request->email;
    $admin->password = bcrypt($request->password);
    $admin->save();

    Auth::guard('admin')->login($admin);
    return redirect()->route('admin.dashboard');
}
}

```

Middleware code –

Commands : -

php artisan make:middleware Admin

Admin.php code-

```
<?php

namespace App\Http\Middleware;

use Closure;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Symfony\Component\HttpFoundation\Response;

class Admin
{
    /**
     * Handle an incoming request.
     *
     * @param  \Closure(\Illuminate\Http\Request): (\Symfony\Component\HttpFoundation\Response)  $next
     */
    public function handle(Request $request, Closure $next): Response
    {
        if(!Auth::guard('admin')->check())
        {
            return redirect()->route('admin.login.index')->with('error', 'Please login first');
        }
        return $next($request);
    }
}
```

Kernel code –

Kernel.php

Define \$routeMiddleware and write this code given below in same as and same place:-

```
<?php

namespace App\Http;

use Illuminate\Foundation\Http\Kernel as HttpKernel;

class Kernel extends HttpKernel
{
    /**
```

```

* The application's global HTTP middleware stack.
*
* These middleware are run during every request to your application.
*
* @var array<int, class-string|string>
*/
protected $middleware = [
    // \App\Http\Middleware\TrustHosts::class,
    \App\Http\Middleware\TrustProxies::class,
    \Illuminate\Http\Middleware\HandleCors::class,
    \App\Http\Middleware\PreventRequestsDuringMaintenance::class,
    \Illuminate\Foundation\Http\Middleware\ValidatePostSize::class,
    \App\Http\Middleware\TrimStrings::class,
    \Illuminate\Foundation\Http\Middleware\ConvertEmptyStringsToNull::class,
];

protected $routeMiddleware = [
    'admin' => \App\Http\Middleware\Admin::class,
];

/**
 * The application's route middleware groups.
 *
 * @var array<string, array<int, class-string|string>>
 */
protected $middlewareGroups = [
    'web' => [
        \App\Http\Middleware\EncryptCookies::class,
        \Illuminate\Cookie\Middleware\AddQueuedCookiesToResponse::class,
        \Illuminate\Session\Middleware\StartSession::class,
        \Illuminate\View\Middleware\ShareErrorsFromSession::class,
        \App\Http\Middleware\VerifyCsrfToken::class,
        \Illuminate\Routing\Middleware\SubstituteBindings::class,
    ],

    'api' => [
        // \Laravel\Sanctum\Http\Middleware\EnsureFrontendRequestsAreStateful::class,
        \Illuminate\Routing\Middleware\ThrottleRequests::class.'api',
        \Illuminate\Routing\Middleware\SubstituteBindings::class,
    ],
];

/**
 * The application's middleware aliases.

```

```

*
* Aliases may be used instead of class names to conveniently assign middleware to routes and groups.
*
* @var array<string, class-string|string>
*/
protected $middlewareAliases = [
    'auth' => \App\Http\Middleware\Authenticate::class,
    'auth.basic' => \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,
    'auth.session' => \Illuminate\Session\Middleware\AuthenticateSession::class,
    'cache.headers' => \Illuminate\Http\Middleware\SetCacheHeaders::class,
    'can' => \Illuminate\Auth\Middleware\Authorize::class,
    'guest' => \App\Http\Middleware\RedirectIfAuthenticated::class,
    'password.confirm' => \Illuminate\Auth\Middleware\RequirePassword::class,
    'precognitive' => \Illuminate\Foundation\Http\Middleware\HandlePrecognitiveRequests::class,
    'signed' => \App\Http\Middleware\ValidateSignature::class,
    'throttle' => \Illuminate\Routing\Middleware\ThrottleRequests::class,
    'verified' => \Illuminate\Auth\Middleware\EnsureEmailIsVerified::class,
];
}

```

Route code –

auth.php

Write route group of admin code in same as and same place :-

```

<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\AdminController;
use App\Http\Controllers\Auth\PasswordController;
use App\Http\Controllers\Auth\NewPasswordController;
use App\Http\Controllers\Auth\VerifyEmailController;
use App\Http\Controllers\Auth\RegisteredUserController;
use App\Http\Controllers\Auth\PasswordResetLinkController;
use App\Http\Controllers\Auth\ConfirmablePasswordController;
use App\Http\Controllers\Auth\AuthenticatedSessionController;
use App\Http\Controllers\Auth\EmailVerificationPromptController;
use App\Http\Controllers\Auth\EmailVerificationNotificationController;

```

```
Route::middleware('guest')->group(function () {
    Route::get('register', [RegisteredUserController::class, 'create'])
        ->name('register');

    Route::post('register', [RegisteredUserController::class, 'store']);

    Route::get('login', [AuthenticatedSessionController::class, 'create'])
        ->name('login');

    Route::post('login', [AuthenticatedSessionController::class, 'store']);

    Route::get('forgot-password', [PasswordResetLinkController::class, 'create'])
        ->name('password.request');

    Route::post('forgot-password', [PasswordResetLinkController::class, 'store'])
        ->name('password.email');

    Route::get('reset-password/{token}', [NewPasswordController::class, 'create'])
        ->name('password.reset');

    Route::post('reset-password', [NewPasswordController::class, 'store'])
        ->name('password.store');
});

Route::middleware('auth')->group(function () {
    Route::get('verify-email', EmailVerificationPromptController::class)
        ->name('verification.notice');

    Route::get('verify-email/{id}/{hash}', VerifyEmailController::class)
        ->middleware(['signed', 'throttle:6,1'])
        ->name('verification.verify');

    Route::post('email/verification-notification', [EmailVerificationNotificationController::class, 'store'])
        ->middleware('throttle:6,1')
        ->name('verification.send');

    Route::get('confirm-password', [ConfirmablePasswordController::class, 'show'])
        ->name('password.confirm');

    Route::post('confirm-password', [ConfirmablePasswordController::class, 'store']);

    Route::put('password', [PasswordController::class, 'update'])->name('password.update');

    Route::post('logout', [AuthenticatedSessionController::class, 'destroy']);
```

```

        ->name('logout');
    });

Route::group(['prefix'=>'admin','as'=>'admin.'], function(){
    Route::get('login', [AdminController::class, 'index']->name('login.index')); // admin.index
    Route::post('login', [AdminController::class, 'login']->name('login')); // admin.login
    Route::get('dashboard', [AdminController::class, 'dashboard']->name('dashboard')->middleware('admin')); //
admin.dashboard
    Route::post('logout', [AdminController::class, 'logout']->name('logout')->middleware('admin'));

    Route::get('register', [AdminController::class, 'registerIndex']->name('register.index'));
    Route::post('register', [AdminController::class, 'register']->name('register'));
});

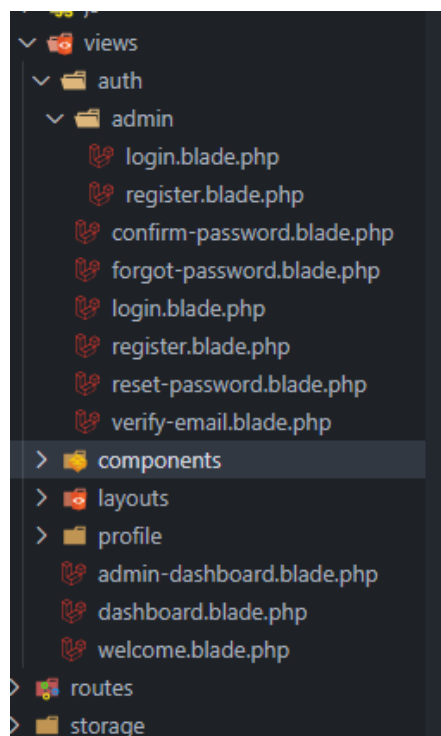
```

View code –

Create folder admin inside auth folder

Then copy and paste **login.blade.php** and **register.blade.php** file

In this structure-



And configure both file –

Login.blade.php

```
<x-guest-layout>
```

```

<!-- Session Status -->
<x-auth-session-status class="mb-4" :status="session('status')" />

<h1><b>Admin Login</b></h1><br>
@if(session()->has('error'))
<span class="text-sm text-red-600 space-y-1">{{ session()->get('error') }}</span>

@endif

<form method="POST" action="{{ route('admin.login') }}">
    @csrf

    <!-- Email Address -->
    <div>
        <x-input-label for="email" :value="__('Email')"/>
        <x-text-input id="email" class="block mt-1 w-full" type="email" name="email" :value="old('email')"
required autofocus autocomplete="username" />
        <x-input-error :messages="$errors->get('email')"/>
    </div>

    <!-- Password -->
    <div class="mt-4">
        <x-input-label for="password" :value="__('Password')"/>
        <x-text-input id="password" class="block mt-1 w-full"
            type="password"
            name="password"
            required autocomplete="current-password" />
        <x-input-error :messages="$errors->get('password')"/>
    </div>

    <!-- Remember Me -->
    <div class="block mt-4">
        <label for="remember_me" class="inline-flex items-center">
            <input id="remember_me" type="checkbox" class="rounded border-gray-300 text-indigo-600 shadow-sm
focus:ring-indigo-500" name="remember">
            <span class="ms-2 text-sm text-gray-600">{{ __('Remember me') }}</span>
        </label>
    </div>

    <div class="flex items-center justify-end mt-4">
        @if (Route::has('password.request'))
            <a class="underline text-sm text-gray-600 hover:text-gray-900 rounded-md focus:outline-none
focus:ring-2 focus:ring-offset-2 focus:ring-indigo-500" href="{{ route('password.request') }}">

```

```

        {{ __('Forgot your password?') }}
    </a>

@endif

    <x-primary-button class="ms-3">
        {{ __('Log in') }}
    </x-primary-button>

</div>

</form>

</x-guest-layout>

```

Register.blade.php

```

<x-guest-layout>

    <h1><b>Admin Register</b></h1><br>

    <form method="POST" action="{{ route('admin.register') }}">

        @csrf

        <!-- Name -->

        <div>

            <x-input-label for="name" :value="__('Name')" />

            <x-text-input id="name" class="block mt-1 w-full" type="text" name="name" :value="old('name')" required
autofocus autocomplete="name" />

            <x-input-error :messages="$errors->get('name')" class="mt-2" />

        </div>

        <!-- Email Address -->

        <div class="mt-4">

            <x-input-label for="email" :value="__('Email')" />

            <x-text-input id="email" class="block mt-1 w-full" type="email" name="email" :value="old('email')"
required autocomplete="username" />

            <x-input-error :messages="$errors->get('email')" class="mt-2" />

        </div>

        <!-- Password -->

        <div class="mt-4">

            <x-input-label for="password" :value="__('Password')" />

            <x-text-input id="password" class="block mt-1 w-full"

                type="password"

                name="password"

                required autocomplete="new-password" />

```



```

        <x-input-error :messages="$errors->get('password')" class="mt-2" />
    </div>

    <!-- Confirm Password -->
    <div class="mt-4">

        <x-input-label for="password_confirmation" :value="__('Confirm Password')"/>

        <x-text-input id="password_confirmation" class="block mt-1 w-full"
            type="password"
            name="password_confirmation" required autocomplete="new-password" />

        <x-input-error :messages="$errors->get('password_confirmation')" class="mt-2" />
    </div>

    <div class="flex items-center justify-end mt-4">

        <a class="underline text-sm text-gray-600 hover:text-gray-900 rounded-md focus:outline-none focus:ring-2
focus:ring-offset-2 focus:ring-indigo-500" href="{{ route('login') }}">
            {{ __('Already registered?') }}
        </a>

        <x-primary-button class="ms-4">
            {{ __('Register') }}
        </x-primary-button>
    </div>
</form>
</x-guest-layout>

```

Also configure **layouts/navigation.blade.php** file-

Same as given code –

```

<nav x-data="{ open: false }" class="bg-white border-b border-gray-100">
    <!-- Primary Navigation Menu -->
    <div class="max-w-7xl mx-auto px-4 sm:px-6 lg:px-8">
        <div class="flex justify-between h-16">
            <div class="flex">
                <!-- Logo -->
                <div class="shrink-0 flex items-center">
                    <a href="{{ route('dashboard') }}">
                        <x-application-logo class="block h-9 w-auto fill-current text-gray-800" />
                    </a>
                </div>
            </div>

            <!-- Navigation Links -->
            <div class="hidden space-x-8 sm:-my-px sm:ms-10 sm:flex">
                <x-nav-link :href="route('dashboard')" :active="request()->routeIs('dashboard')">
                    {{ __('Dashboard') }}
                </x-nav-link>
            </div>
        </div>

        <!-- Settings Dropdown -->
        <div class="hidden sm:flex sm:items-center sm:ms-6">
            <x-dropdown align="right" width="48">
                <x-slot name="trigger">
                    <button class="inline-flex items-center px-3 py-2 border border-transparent text-sm leading-4 font-medium rounded-md text-gray-500 bg-white hover:text-gray-700 focus:outline-none transition ease-in-out duration-150">
                        @if(Auth::guard('admin')->check())
                            <div>{{ Auth::guard('admin')->user()->name }}</div>
                        @endif
                    </button>
                </x-slot>
            </div>
        </div>
    </div>
</nav>

```

```

        @elseif(Auth::guard('web')->check())
            <div>{{ Auth::user()->name }}</div>
        @endif

        <div class="ms-1">
            <svg class="fill-current h-4 w-4" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 20 20">
                <path fill-rule="evenodd" d="M5.293 7.293a1 1 0 011.414 0L10 10.586l3.293-3.293a1 1 0 011.414 1.414l-4 4a1 1 0 01-1.414 0L1-4a1 1 0 010 1.414z" clip-rule="evenodd" />
            </svg>
        </div>
    </button>
</x-slot>

<x-slot name="content">

    <!-- Authentication -->
    @if(Auth::guard('admin')->check())
        <form method="POST" action="{{ route('admin.logout') }}">
            @csrf

            <x-responsive-nav-link :href="route('admin.logout')"
                onclick="event.preventDefault();
                this.closest('form').submit();">
                {{ __('Log Out') }}
            </x-responsive-nav-link>
        </form>
    @else
        <x-dropdown-link :href="route('profile.edit')">
            {{ __('Profile') }}
        </x-dropdown-link>
        <form method="POST" action="{{ route('logout') }}">
            @csrf

            <x-responsive-nav-link :href="route('logout')"
                onclick="event.preventDefault();
                this.closest('form').submit();">
                {{ __('Log Out') }}
            </x-responsive-nav-link>
        </form>
    @endif

</x-slot>
</x-dropdown>
</div>

<!-- Hamburger -->
<div class="me-2 flex items-center sm:hidden">
    <button @click="open = ! open" class="inline-flex items-center justify-center p-2 rounded-md text-gray-400 hover:text-gray-500
    hover:bg-gray-100 focus:outline-none focus:bg-gray-100 focus:text-gray-500 transition duration-150 ease-in-out">
        <svg class="h-6 w-6" stroke="currentColor" fill="none" viewBox="0 0 24 24">
            <path :class="{ 'hidden': open, 'inline-flex': ! open }" class="inline-flex" stroke-linecap="round" stroke-linejoin="round"
            stroke-width="2" d="M4 6h16M4 12h16M4 18h16" />
            <path :class="{ 'hidden': ! open, 'inline-flex': open }" class="hidden" stroke-linecap="round" stroke-linejoin="round"
            stroke-width="2" d="M6 18l18 6l12 12" />
        </svg>
    </button>
</div>
</div>
</div>

<!-- Responsive Navigation Menu -->
<div :class="{ 'block': open, 'hidden': ! open }" class="hidden sm:hidden">
    <div class="pt-2 pb-3 space-y-1">
        <x-responsive-nav-link :href="route('dashboard')" :active="request()->routeIs('dashboard')">
            {{ __('Dashboard') }}
        </x-responsive-nav-link>
    </div>
</div>

<!-- Responsive Settings Options -->
<div class="pt-4 pb-1 border-t border-gray-200">
    <div class="px-4">
        @if(Auth::guard('admin')->check())
            <div class="font-medium text-base text-gray-800">{{ Auth::guard('admin')->user()->name }}</div>
            <div class="font-medium text-sm text-gray-500">{{ Auth::guard('admin')->user()->email }}</div>
        @elseif(Auth::guard('web')->check())
            <div class="font-medium text-base text-gray-800">{{ Auth::user()->name }}</div>
            <div class="font-medium text-sm text-gray-500">{{ Auth::user()->email }}</div>
        @endif
    </div>

    <div class="mt-3 space-y-1">
        <!-- <x-responsive-nav-link :href="route('profile.edit')">
            {{ __('Profile') }}
        </x-responsive-nav-link -->

        <!-- Authentication -->
        @if(Auth::guard('admin')->check())
            <form method="POST" action="{{ route('admin.logout') }}">
                @csrf

```

```

        <x-responsive-nav-link :href="route('admin.logout')"
            onclick="event.preventDefault();
                this.closest('form').submit();">
            {{ __('Log Out') }}
        </x-responsive-nav-link>
    </form>
@else
    <form method="POST" action="{{ route('logout') }}">
        @csrf

        <x-responsive-nav-link :href="route('logout')"
            onclick="event.preventDefault();
                this.closest('form').submit();">
            {{ __('Log Out') }}
        </x-responsive-nav-link>
    </form>
@endif
</div>
</div>
</div>
</nav>

```

Create file **view/admin-dashboard.blade.php** and copy all codes in **view/dashboard.blade.php**

admin-dashboard.blade.php codes-

```

<x-app-layout>
    <x-slot name="header">
        <h2 class="font-semibold text-xl text-gray-800 leading-tight">
            {{ __('Admin Dashboard') }}
        </h2>
    </x-slot>

    <div class="py-12">
        <div class="max-w-7xl mx-auto sm:px-6 lg:px-8">
            <div class="bg-white overflow-hidden shadow-sm sm:rounded-
lg">
                <div class="p-6 text-gray-900">
                    {{ __("You're logged in!") }}
                </div>
            </div>
        </div>
    </div>
</x-app-layout>

```

Admin login page url-

Base url - /admin/login

Admin register url –

Base url - /admin/register

Created By – Vishal Kumar Maurya Laravel Developer