

## Single Login Page

## 2 - Multi Authentication System Using Model in LARAVEL

### Single user Authentication –

Commands : -

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

```
php artisan breeze:install
```

```
> blade
```

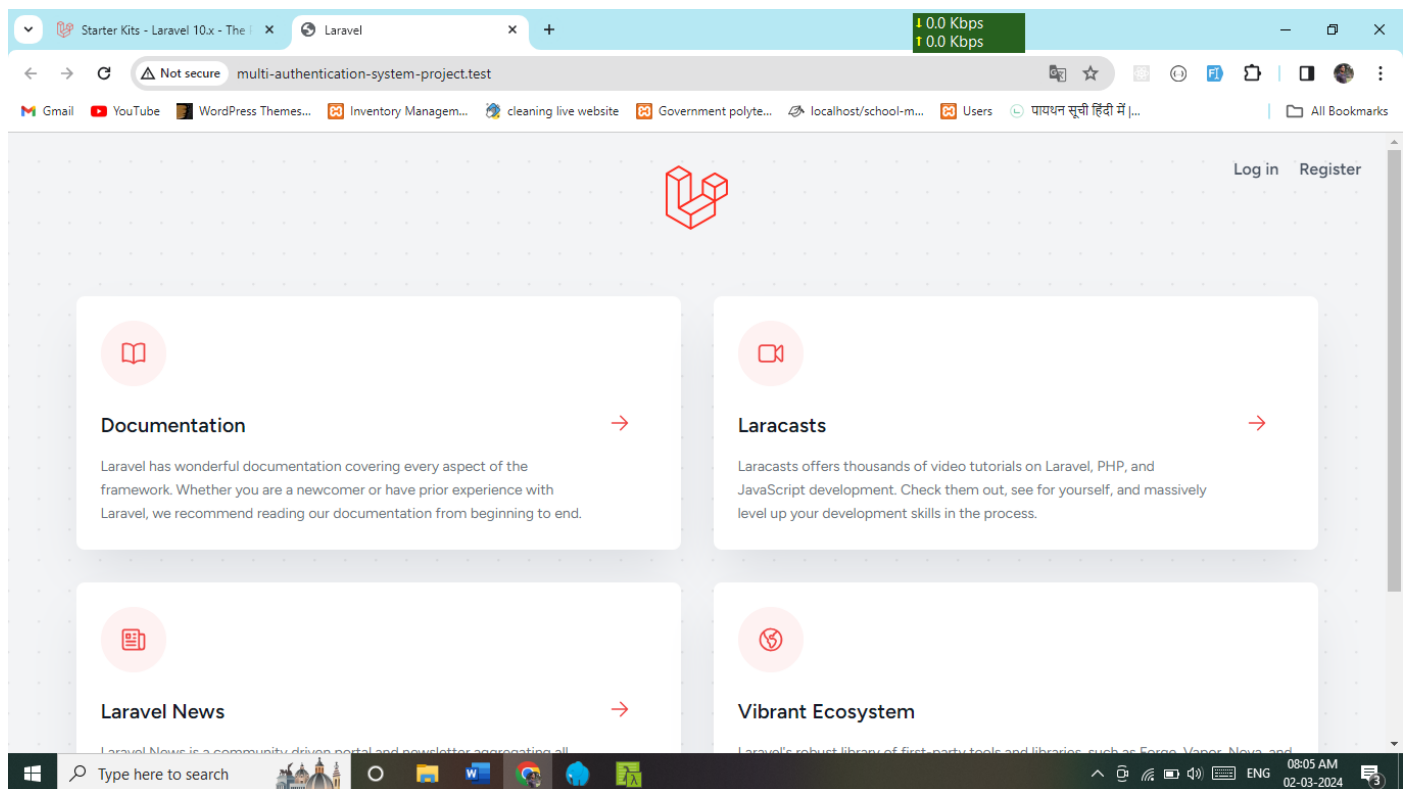
```
> phpUnit
```

```
php artisan migrate
```

Some Optional Command -

```
npm install
```

```
npm run dev
```



## Multi Authentication (Admin ) –

Commands : -

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

```
php artisan breeze:install
```

## Migration

Configure the 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('avatar')->default('/uploads/user.png');
            $table->string('name');
            $table->string('email')->unique();
            $table->enum('role', ['user', 'admin'])->default('user');
            $table->timestamp('email_verified_at')->nullable();
            $table->string('password');
            $table->rememberToken();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
```

```
{
    Schema::dropIfExists('users');
}
};
```

Then migrate

## Seeder code –

### Commands : -

php artisan make:seeder UserSeeder

php artisan db:seed

### UserSeeder.php

```
<?php

namespace Database\Seeders;

use App\Models\User;
use Illuminate\Database\Seeder;
use Illuminate\Support\Facades\Hash;
use Illuminate\Database\Console\Seeds\WithoutModelEvents;

class UserSeeder extends Seeder
{
    protected static ?string $password;
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
        User::insert([
            [
                'name' => 'Admin',
                'email' => 'admin@gmail.com',
                'role' => 'admin',
```

```

                'password' => static::$password ??=
Hash::make('password'),

            ],
            [
                'name' => 'User',
                'email' => 'user@gmail.com',
                'role' => 'user',
                'password' => static::$password ??=
Hash::make('password'),

            ]
        ]);
    }
}

```

**DatabaseSedder.php** call UserSeeder file

```

<?php

namespace Database\Seeders;

// use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;

class DatabaseSeeder extends Seeder
{
    /**
     * Seed the application's database.
     */
    public function run(): void
    {
        // \App\Models\User::factory(10)->create();

        // \App\Models\User::factory()->create([
        //     'name' => 'Test User',
        //     'email' => 'test@example.com',
        // ]);

        $this->call(UserSeeder::class);
    }
}

```

Then seed this file using command

## Admin Controller code –

Create Admin folder then create a controller file – AdminDashboardController.php

**Commands : -**

php artisan make:controller Admin/AdminDashboardController

**AdminDashboardController.php** code-

```
<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;

class AdminDashboardController extends Controller
{
    public function index()
    {
        return View('admin.dashboard.index');
    }
}
```

## Auth Controller code –

Configure this code -

**AuthenticatedSessionController.php** code-

```
public function store(LoginRequest $request): RedirectResponse
{
    $request->authenticate();

    $request->session()->regenerate();
}
```

```

        if($request->user()->role === 'admin')
        {
            return redirect()->intended(RouteServiceProvider::ADMIN);
        }

        return redirect()->intended(RouteServiceProvider::HOME);
    }

```

## Middleware code –

### Commands : -

php artisan make:middleware RoleMiddleware

### RoleMiddleware.php code-

```

<?php

namespace App\Http\Middleware;

use Closure;
use Illuminate\Http\Request;
use Symfony\Component\HttpFoundation\Response;

class RoleMiddleware
{
    /**
     * Handle an incoming request.
     *
     * @param  \Closure(\Illuminate\Http\Request):
(\Symfony\Component\HttpFoundation\Response)  $next
     */
    public function handle(Request $request, Closure $next, $role):
Response
    {
        if($request->user()->role===$role)
        {

            return $next($request);
        }
    }
}

```

```
        return to_route('dashboard');
    }
}
```

## Kernel code –

Define role middleware

### Kernel.php

```
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,

    'role' => \App\Http\Middleware\RoleMiddleware::class,
];
```

## Route code –

Create a admin route then write the code:-

### admin.php

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\Admin\AdminDashboardController;

Route::group(['prefix' => 'admin', 'as' => 'admin.'], function(){
```

```
Route::get('dashboard',[AdminController::class,'index']->name('dashboard '));

});
```

## Providers code –

### RouteServiceProvider.php

```
* @var string
*/
public const HOME = '/dashboard';
public const ADMIN = 'admin/dashboard';
```

```
public function boot(): void
{
    RateLimiter::for('api', function (Request $request) {
        return Limit::perMinute(60)->by($request->user()?->id ?: $request->ip());
    });

    $this->routes(function () {
        Route::middleware('api')
            ->prefix('api')
            ->group(base_path('routes/api.php'));

        Route::middleware('web')
            ->group(base_path('routes/web.php'));

        Route::middleware(['web','auth','role:admin'])
            ->group(base_path('routes/admin.php'));
    });
}
```

## View code –

Create folder admin/dashboard then create a file **index.blade.php**

Copy all code **dashboard.blade.php** in **index.blade.php**

**Index.blade.php** code-



```
<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 - /login

**Created By – Vishal Kumar Maurya** Laravel Developer