**Toaster:**

**App.balde.php**

**Js link-**

<script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/toastr.js/latest/js/toastr.min.js"></script>

<script>

@if(Session::has('message'))

var type = "{{ Session::get('alert-type','info')}}"

switch(type){

case 'info':

toastr.info(" {{ Session::get('message')}}");

break;

case 'success':

toastr.success(" {{ Session::get('message')}}");

break;

case 'warning':

toastr.warning(" {{ Session::get('message')}}");

break;

case 'error':

toastr.error(" {{ Session::get('message')}}");

break;

}

@endif

</script>

Css link-

<link rel="stylesheet" type="text/css" href="https://cdnjs.cloudflare.com/ajax/libs/toastr.js/latest/css/toastr.css">

**Use in function demo:**

homeController:-

use Auth;

public function Logout()

{

Auth::logout();

$notification = array(

'message' => 'Successfully Logout',

'alert-type' => 'success'

);

return Redirect()->route('login')->with($notification);

}

**Admin all work**

Resource->view->admin(folder)->home.blade.php->admin\_layouts.blade.php->auth(folder)->login.blade.php

App->controller->AdminController.php->Admin(folder)->LoginController.php

**Create**:Admin model and admins table

**Route->Web.php:**

// Admin Routes

Route::get('admin/home', 'AdminController@index');

Route::get('admin', 'Admin\LoginController@showLoginForm')->name('admin.login');

Route::post('admin', 'Admin\LoginController@login');

Route::get('admin/logout', 'AdminController@Logout')->name('admin.logout');

**Admin table**

public function up()

    {

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

            $table->id('id');

            $table->string('name');

            $table->string('phone')->nullable();

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

            $table->string('password');

            $table->timestamps();

        });

    }

**Admin model code:**

<?php

namespace App;

use Illuminate\Contracts\Auth\MustVerifyEmail;

use Illuminate\Foundation\Auth\User as Authenticatable;

use Illuminate\Notifications\Notifiable;

class Admin extends Authenticatable

{

    use Notifiable;

    /\*\*

     \* The attributes that are mass assignable.

     \*

     \* @var array

     \*/

    protected $fillable = [

        'name', 'email', 'password','phone'

    ];

    /\*\*

     \* The attributes that should be hidden for arrays.

     \*

     \* @var array

     \*/

    protected $hidden = [

        'password', 'remember\_token',

    ];

}

**App->http->Admin(folder)->LoginController code**

<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;

use Illuminate\Foundation\Auth\AuthenticatesUsers;

use Illuminate\Support\Facades\Auth;

class LoginController extends Controller

{

    /\*

    |--------------------------------------------------------------------------

    | Login Controller

    |--------------------------------------------------------------------------

    |

    | This controller handles authenticating users for the application and

    | redirecting them to your home screen. The controller uses a trait

    | to conveniently provide its functionality to your applications.

    |

    \*/

    use AuthenticatesUsers;

    /\*\*

     \* Where to redirect users after login.

     \*

     \* @var string

     \*/

    protected $redirectTo = 'admin/home';

    /\*\*

     \* Create a new controller instance.

     \*

     \* @return void

     \*/

     public function \_\_construct()

    {

        $this->middleware('guest:admin')->except('logout');

    }

    public function showLoginForm()

    {

         if (Auth::id()) {

             return redirect()->back();

         }else{

            return view('admin.auth.login');

         }

    }

    protected function guard()

    {

        return Auth::guard('admin');

    }

}

**App->http->AdminController.php**

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Admin;

use Auth;

class AdminController extends Controller

{

       public function \_\_construct()

    {

        $this->middleware('auth:admin');

    }

  public function index(){

   return view('admin.home');

  }

  public function Logout(){

   Auth::logout();

     $notification = array(

        'message' => 'Admin Successfully Logout',

        'alert-type' => 'success'

     );

     return Redirect()->route('admin.login')->with($notification);

  }

}

**Config->auth.php**

<?php

return [

    /\*

    |--------------------------------------------------------------------------

    | Authentication Defaults

    |--------------------------------------------------------------------------

    |

    | This option controls the default authentication "guard" and password

    | reset options for your application. You may change these defaults

    | as required, but they're a perfect start for most applications.

    |

    \*/

    'defaults' => [

        'guard' => 'web',

        'passwords' => 'users',

    ],

    /\*

    |--------------------------------------------------------------------------

    | Authentication Guards

    |--------------------------------------------------------------------------

    |

    | Next, you may define every authentication guard for your application.

    | Of course, a great default configuration has been defined for you

    | here which uses session storage and the Eloquent user provider.

    |

    | All authentication drivers have a user provider. This defines how the

    | users are actually retrieved out of your database or other storage

    | mechanisms used by this application to persist your user's data.

    |

    | Supported: "session", "token"

    |

    \*/

    'guards' => [

        'web' => [

            'driver' => 'session',

            'provider' => 'users',

        ],

        'admin' => [

            'driver' => 'session',

            'provider' => 'admins',

        ],

        'api' => [

            'driver' => 'token',

            'provider' => 'users',

            'hash' => false,

        ],

    ],

    /\*

    |--------------------------------------------------------------------------

    | User Providers

    |--------------------------------------------------------------------------

    |

    | All authentication drivers have a user provider. This defines how the

    | users are actually retrieved out of your database or other storage

    | mechanisms used by this application to persist your user's data.

    |

    | If you have multiple user tables or models you may configure multiple

    | sources which represent each model / table. These sources may then

    | be assigned to any extra authentication guards you have defined.

    |

    | Supported: "database", "eloquent"

    |

    \*/

    'providers' => [

        'users' => [

            'driver' => 'eloquent',

            'model' => App\User::class,

        ],

        'admins' => [

            'driver' => 'eloquent',

            'model' => App\Admin::class,

        ],

        // 'users' => [

        //     'driver' => 'database',

        //     'table' => 'users',

        // ],

    ],

    /\*

    |--------------------------------------------------------------------------

    | Resetting Passwords

    |--------------------------------------------------------------------------

    |

    | You may specify multiple password reset configurations if you have more

    | than one user table or model in the application and you want to have

    | separate password reset settings based on the specific user types.

    |

    | The expire time is the number of minutes that the reset token should be

    | considered valid. This security feature keeps tokens short-lived so

    | they have less time to be guessed. You may change this as needed.

    |

    \*/

    'passwords' => [

        'users' => [

            'provider' => 'users',

            'table' => 'password\_resets',

            'expire' => 60,

            'throttle' => 60,

        ],

        'admins' => [

            'provider' => 'admins',

            'table' => 'password\_resets',

            'expire' => 60,

            'throttle' => 60,

        ],

    ],

];