Prevent access each other page (user ,admin)

To achieve the desired behavior where accessing unauthorized routes redirects users to a “Not Found” page , you can use middleware to check the user’s role and redirect them accordingly. Here’s how you can do it .

1.Create a new middleware to check the user’s role.

2.Apply this middle to the routes you want to protected.

3.Define the behavior for unauthorized access in the middleware .

Here is the guide:

1.Create Middleware:

Create a middleware to check the user’s role. If the user is trying to access a route they are not authorized for , redirect them to a “Not Found” page .

```
php artisan make:middleware CheckUserRole
```

Edit Middleware

Open the ‘CheckUserRole’ middleware and implement the logic to check the user’s role. If the role is not appropriate for the accessed route, Redirect to the “Not Found” page .

```
<?php
namespace App\Http\Middleware ;

use Closure;
use Illuminate\Http\Request;
use Illuminate\Support\Faceades\Auth;
```

```
class CheckUserRole
{
    public function handle(Request $request , Closure $next , .
{
    if(Auth::check()){
         $userRole = Auth::user()->role;
            foreach($roles as $role){
                if($userRole==$role){
                    return $next($request);
                }
            }
        }
        return abort(404);
    }
}
```

Register Middle:

Register the middleware in the ‘$routeMiddleware’ array in ‘app/Http/Kernel.php’.

```
protected $routeMiddleware=[
'check.role' =>\App\Htppt\Middleware\CheckUserRole::class,
];
```

Apply Middlewware to Routes:

```
// Routes

//Route Admin
Route::prefix('admin')->middleware(['auth', 'check.role:1'])->g
    // Admin routes...
    Route::get('/dashboard', [DashboardController::class, 'dash
    Route::get('/category/index', [CategoryController::class, 'i
    Route::get('/category/create', [CategoryController::class,
    Route::post('/category/store', [CategoryController::class,
    Route::get('/category/{category}/edit', [CategoryController
    Route::put('/category/{category}',[CategoryController::clas

    Route::get('/laptop/index', [LaptopController::class, 'inde

    Route::get('/accessory/index', [AccessoryController::class,

    Route::get('/setting/introduction', [SettingController::clas
    Route::get('/setting/user', [SettingController::class, 'use
    Route::get('/setting/laptop', [SettingController::class, 'la
    Route::get('/setting/accessory', [SettingController::class,
    Route::get('/setting/general', [SettingController::class, 'g
});

//Route User
Route::prefix('user')->middleware(['auth', 'check.role:0'])->gro
    // User routes...
    Route::get('/home', [HomeController::class, 'home'])->name(
    Route::get('/category/{category}', [AllCategoryController::c
});
```

Prevent from access “/login” “/register” after authentication

```
Route::get('/login',function(){
    if(Auth::check()){
        if(Auth::user()->role==1){
            return redirect()->route('dashboard');
        }else{
                return redirect()->route('home')
        }else{
            return view('auth.login');
}
})->name('login');
Route::get('/register' , function() {
    if(Auth::check()){
        if(Auth::user()->role==1){
            return redirect()->route('dashboard');
        }else{
            return redirect()->route('home');
        }else{
            return view('auth.register');
    }
})->name('register')
```