**Laravel Permissions**

Step-1:

First of all install Spatie permission in your project.

Installation Code: composer require spatie/laravel-permission

Step-2:

Then open config/app.php…. and add to providersthis code below…

'providers' => [

// ...

Spatie\Permission\PermissionServiceProvider::class,

];

Then you need to assign these roles to the user. At the ‘user.php’ Model you need to add this line of codes.

use Illuminate\Foundation\Auth\User as Authenticatable;

use Spatie\Permission\Traits\HasRoles;

class User extends Authenticatable

{

use HasRoles;

// ...

}

Step-3:

The migrate the database with :

Php artisan migrate

Php artisan optimize

Step-4:

Then create 2 seeder….. Write code in CMD….

1. Php artisan make:seeder RolePermissionSeeder
2. Php artisan make:seeder DatabaseSeeder
3. Php artisan make:seeder UserSeeder

You can find this seeders in database/seeders folder.

Step-5:

At UserSeeder write below codes inside run() function:

 public function run()

    {

        $user=User::where('email','admin@gmail.com')->first();

        if(is\_null($user)){

            $user=new User();

            $user->name="Super Admin";

            $user->email="admin@gmail.com";

            $user->password=Hash::make('12345678');

            $user->save();

        }else{

            return 0;

        }

    }

Run this seeder with CMD code: Php artisan db:seed

Step-6:

At RolePermissionSeeder you will write this codes below:

  public function run()

    {

        //create role

        $roleSuperAdmin = Role::create(['name' => 'Super Admin']);

        $roleAdmin = Role::create(['name' => 'Admin']);

        $roleManager = Role::create(['name' => 'Manager']);

        $roleExecutive = Role::create(['name' => 'Executive']);

        $roleGeneralUser = Role::create(['name' => 'General user']);

        //create permissions

        $permission = Permission::create(['name' => 'edit articles']);

        $permissions=[

            'dashboard.view',

            'banner.create',

            'banner.store',

            'banner.edit',

            'banner.delete',

            'banner.view',

            'user.create',

            'user.view',

            'user.edit',

            'user.store',

            'user.delete',

            'service.create',

            'service.view',

            'service.edit',

            'service.store',

            'service.delete',

            'product.create',

            'product.view',

            'product.edit',

            'product.store',

            'product.delete',

            'content.create',

            'content.view',

            'content.edit',

            'content.store',

            'content.delete',

            'blog.create',

            'blog.view',

            'blog.edit',

            'blog.store',

            'blog.delete',

            'setting.view',

            'setting.edit',

            'setting.update',

            'setting.delete',

        ];

        //create permissions

        for($i=0;$i < count($permissions);$i++){

            $permission = Permission::create(['name' =>  $permissions[$i]]);

            $roleSuperAdmin->givePermissionTo($permission);

            $permission->assignRole($roleSuperAdmin);

        }

    }

Run this seeder with CMD code: Php artisan db:seed

Step-7:

At DatabaseSeeder you will write this codes below:

 public function run()

    {

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

        $this->call(RolePermissionSeeder::class);

    }

Run this seeder with CMD code: Php artisan db:seed

Step-8:

Check the roles,permissions,roles\_has\_permissions table in database . You will see results.

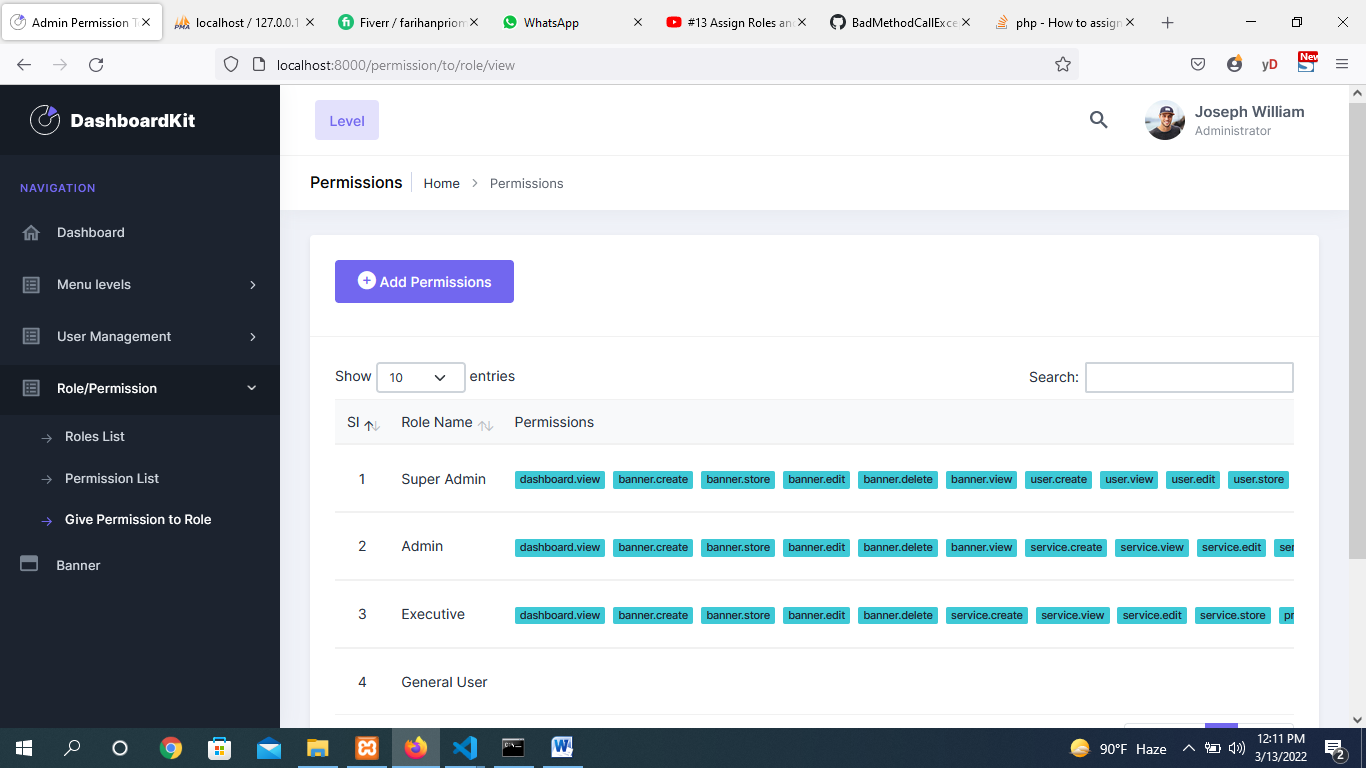
Now create crud to show roles and permissions data in your site.

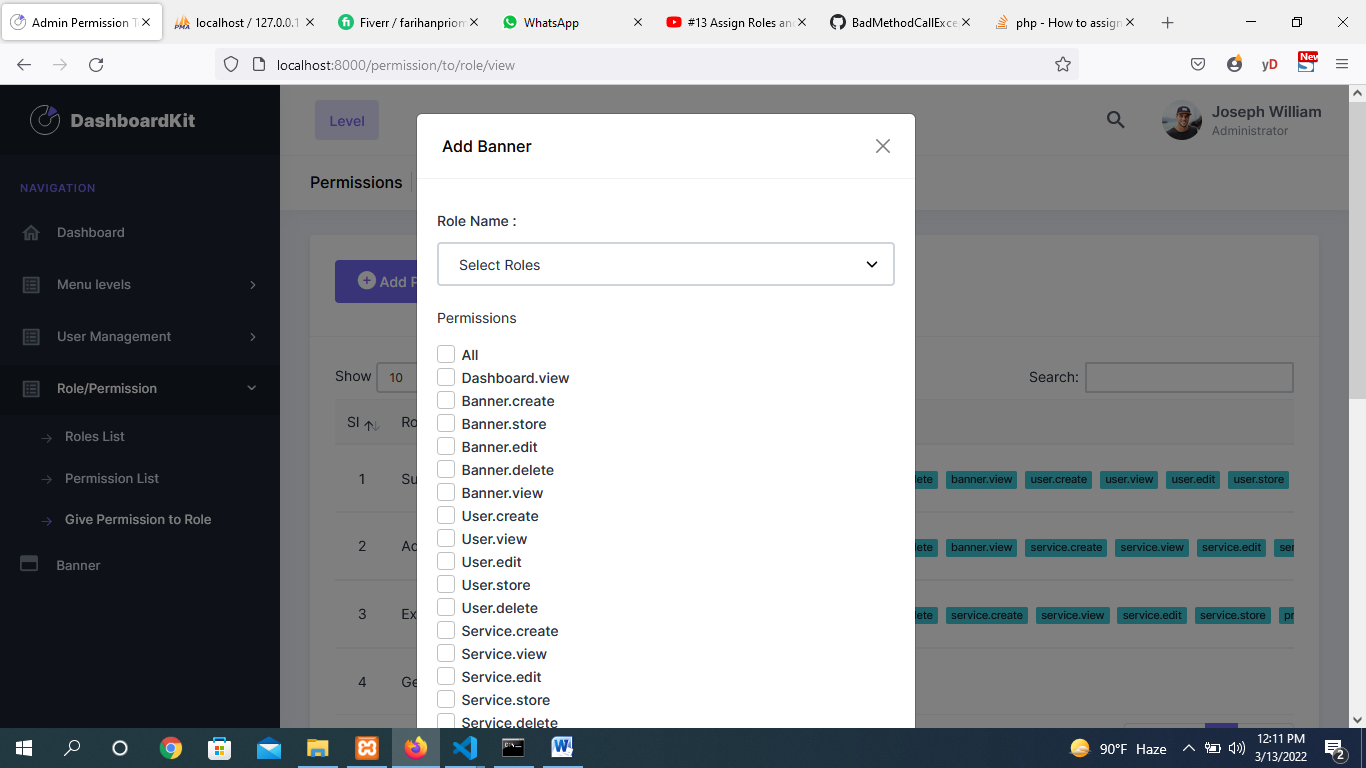
And to create more permissions and roles you can assign permissions to role by creating more cruds. Example tutorial has been given below:

<https://www.youtube.com/watch?v=tIk8OmT_0dg&list=PL_ftyCsXJUO4aDoMz7P8xnMf6xsUWN57x&index=6>

Step-9:

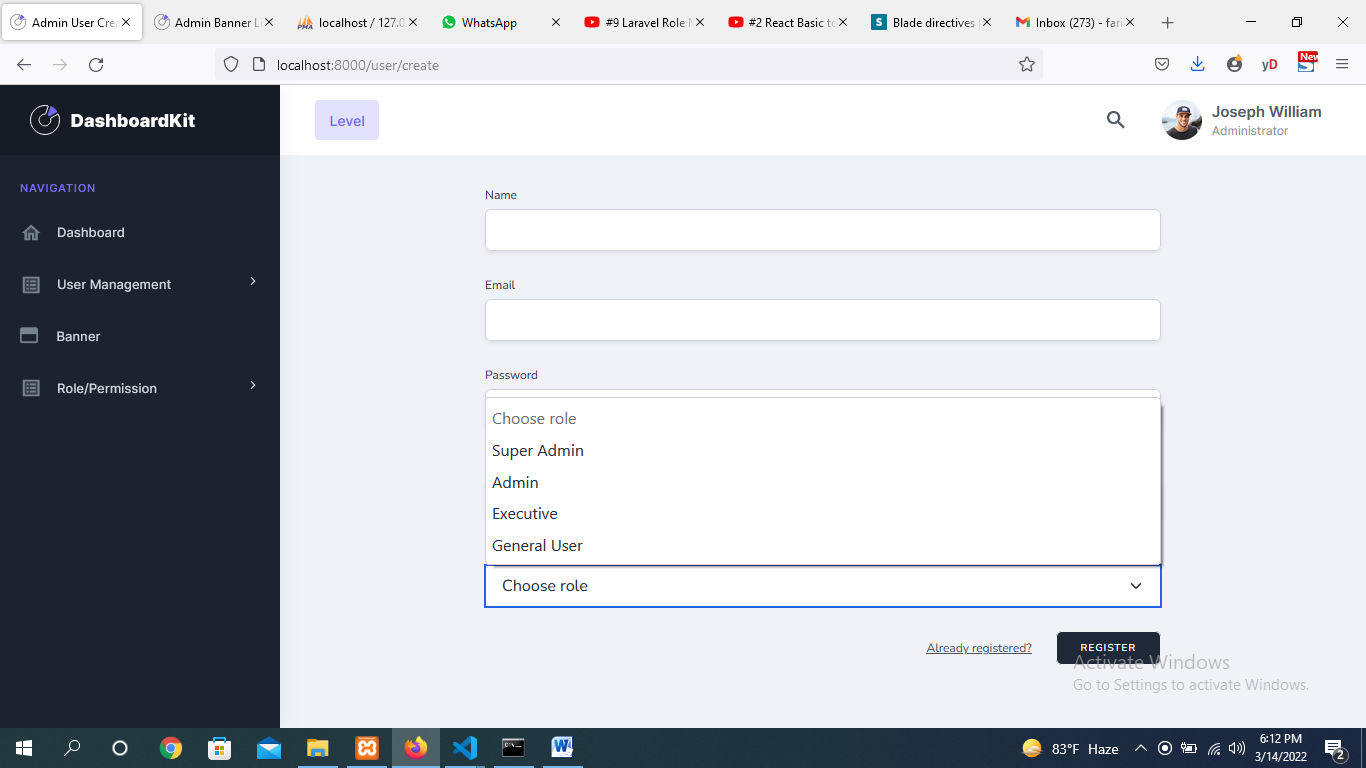
After that your Role view will be like that:





Then you need to check the database ‘role\_has\_permissions’ table… If the roles and permissions id will be generated then that will be okay.

Step-10: After assigning permissions to roles you can assign users the roles. You just need to do some changes in ‘registration.blade.php’ .



Generate all the roles in a dropdown to choose.

Then in ‘userController.php’ you need to write a simple code in ‘store’ function just like this.

**$user->assignRole($request->role);**

Example:

            $validated = $request->validate([

                'name' => 'required|max:255',

                'email' => 'required|email',

                'password'=> 'min:8'

            ]);

            $confirm=$request->password\_confirmation;

            $user=new User();

            $user->name= $request->name;

            $user->email=$request->email;

            $user->password=Hash::make($request->password);

            $user->status ='Active';

            $user->created\_by = auth()->user()->id;

            $user->deleted ='No';

            if($confirm==$request->password){

                $user->save();

                $user->assignRole($request->role);

                return redirect('user/view')

->with('message','User saved sucessfully');

            }else{

                return back()->with('message','User not saved');

            }

You can check this at ‘model\_has\_roles’ table. The ‘model\_id’ will be similar to your user id which you have just saved.

Step-11:

After that the user got a role and role has some permissions. Now you need to permit the permissions in your controller or blade page just like that.

For controller:

public function index()

    {

       $user=Auth::user();

        if($user->can('user.view')){

            $users=User::where('deleted','=','No')->get();

            return view('admin.User.userList',['users'=>$users,'roles'=>$roles]);

        }else{

            abort(403,'You cannot see user list');

        }

    }

And for blade page :

 @if(Auth::guard('web')->user()->can('user.create'))

         <a class="btn btn-primary"

href="{{route('userCreate')}}" style="color:#fff;">

<i class="fa-solid fa-circle-plus"></i>

Add User

</a>

  @endif

Where you use this if/else condition you can get entry if your user id has this permission. Otherwise this will be invincible.