Hello Dev, So today we are going to learn Laravel 10 User Roles and Permissions Tutorial Example. This tutorial will cover on laravel 10 user roles and permissions tutorial with example.

We are going to use spatie/laravel-permission package for this tutorial eventually. Spatie role and permission composer package provides a way to create ACL in laravel 10. They provide how to assign role to user, how to assign permission to user and how to assign permission assign to roles. I will write step by step creating roles and permissions in laravel 10 application during this tutorial.

**Steps on Laravel 10 User Roles and Permissions Tutorial Example:**

* **Step 1:**Installing fresh new Laravel 10 Application
* **Step 2:**Creating Database and Configuration
* **Step 3:**Installing Composer Packages
* **Step 4:**Creating Database and Santri Migration
* **Step 5:**Creating Models
* **Step 6:**Adding Middleware
* **Step 7:**Creating Authentication
* **Step 8:**Creating Routes
* **Step 9:**Adding Controllers
* **Step 10:**Adding Blade Files
* **Step 11:**Creating Seeder For Permissions and Admin User
* **Step 12:**Testing
* **Step 13:**Conclusion

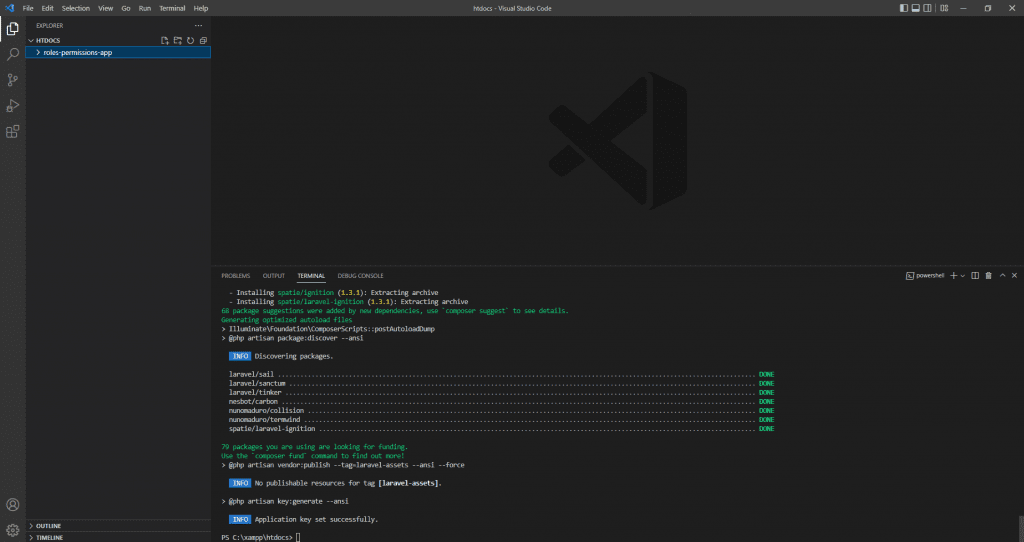
### **Step 1: Installing fresh new Laravel 10 Application**

Firstly we are going to install a fresh new laravel 10 application. To install a laravel 10 application run the following command into terminal.

composer create-project laravel/sisppit

cd sisppit

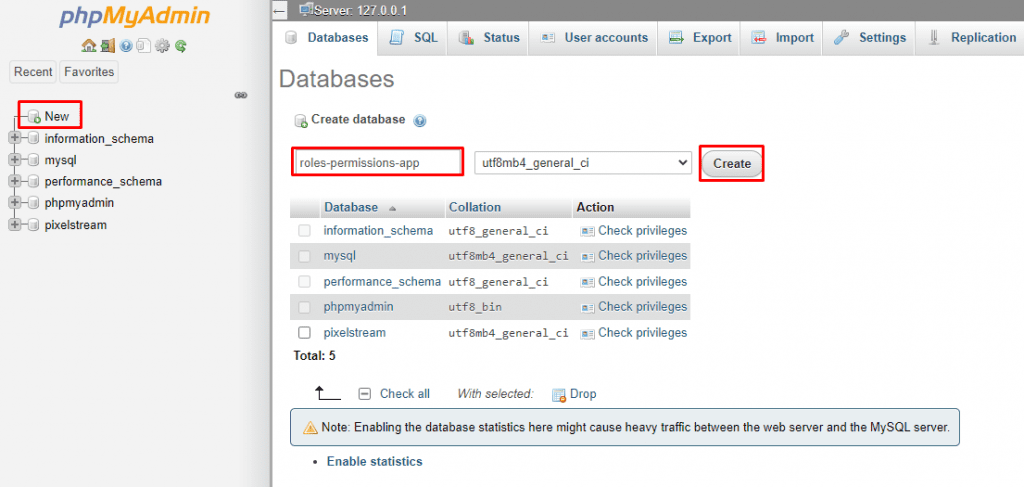
Note: “sisppit” is our laravel application name.



**Installing fresh new Laravel 10 Application**

### **Step 2: Creating Database and Configuration**

Now we are going to create a database. Open phpmyadmin and create a database with a name “sisppit” (you can use whatever name you like).



**Creating Database**

So now, After create a database now we are going to add database details to laravel **.env** file. Open **.env** file and enter the database details.

DB\_CONNECTION=mysql

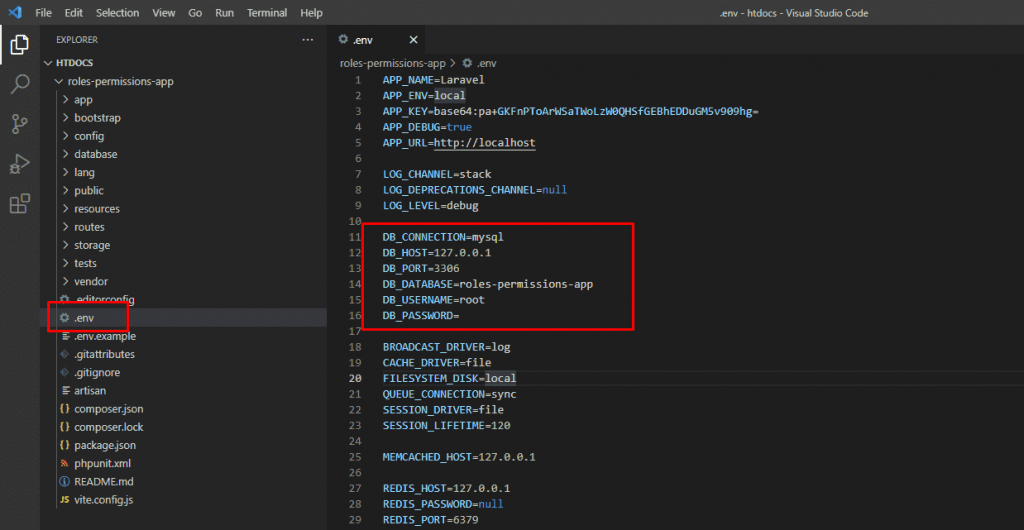
DB\_HOST=127.0.0.1

DB\_PORT=3306

DB\_DATABASE=sisppit

DB\_USERNAME=root

DB\_PASSWORD=



**Database Configuration**

### **Step 3: Installing Composer Packages**

Now we require to install Spatie package for ACL, that way we can use it’s method. Also we will install form collection package. Run the following code in terminal.

composer require spatie/laravel-permission

composer require laravelcollective/html

We can also custom changes on Spatie package, so if you also want to changes then you can fire bellow command and get config file in **config/permission.php** and migration files.

php artisan vendor:publish --provider="Spatie\Permission\PermissionServiceProvider"

Now you can see **permission.php** file and one migrations. so you can run migration using following command:

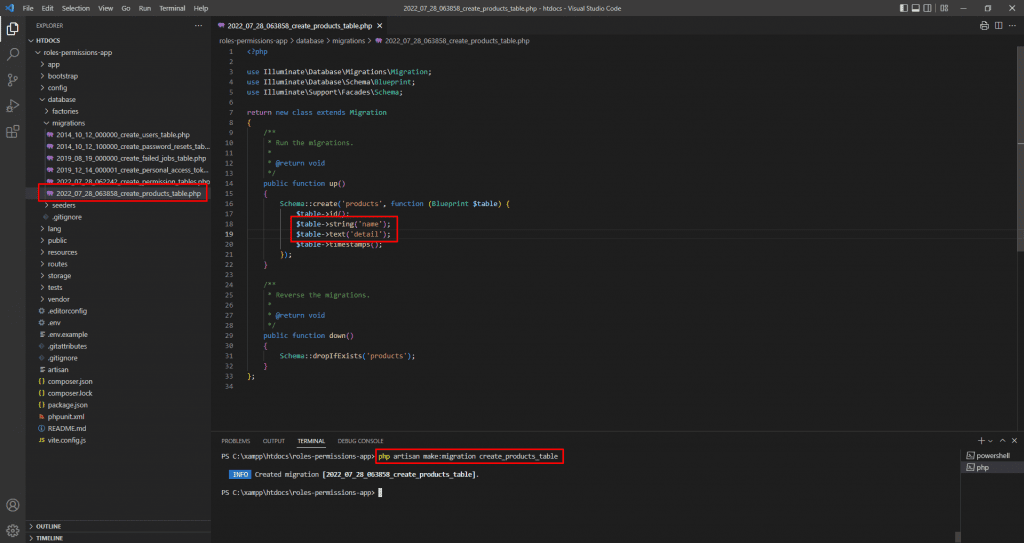
php artisan migrate

### **Step 4: Santri Migration**

Now we are going to create a Santris Migration file. To create a Santris Migration file run the following command in terminal.

php artisan make:migration create\_santris\_table

Now add the following fields to santris migration.



**Santris Migration**

**Buka create\_santris\_table.php isikan dengan di bawah ini**

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

return new *class* extends Migration

{

*/\*\**

*\* Run the migrations.*

*\**

*\* @return void*

*\*/*

    public *function* up()

    {

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

            $table->id();

            $table->bigInteger('nisl')->unique();

            $table->bigInteger('nisn');

            $table->bigInteger('nik');

            $table->string('name');

            $table->string('tempat\_lahir');

            $table->date('tanggal\_lahir');

            $table->string('jenis\_kelamin');

            $table->integer('kelas');

            $table->string('kelas\_paralel'); *//masih dipakai ga nih?*

            $table->string('kelas\_paralel\_kemenag'); *//masih dipakai ga nih?*

            $table->string('nomor\_absen');

            $table->string('status');

            $table->string('asal\_sekolah');

            $table->string('npsn\_asal\_sekolah');

            $table->string('hoby');

            $table->string('cita-cita');

            $table->string('anak\_ke\_berapa');

            $table->string('dari\_berapa\_anak');

            $table->string('jumlah\_saudara');

            $table->string('Jenis\_sekolah');

            $table->string('status\_sekolah');

            $table->string('kota\_kabupaten\_sekolah');

            $table->integer('nomor\_peserta\_ujian');

            $table->integer('nomor\_kartu\_keluarga');

            $table->integer('nik\_bapak');

            $table->string('nama\_bapak');

            $table->string('tempat\_lahir\_bapak');

            $table->date('tanggal\_lahir\_bapak');

            $table->string('pendidikan\_bapak');

            $table->string('pekerjaan\_bapak');

            $table->integer('nomor\_kontak\_bapak');

            $table->string('alamat\_bapak');

            $table->integer('nik\_ibu');

            $table->string('nama\_ibu');

            $table->string('tempat\_lahir\_ibu');

            $table->date('tanggal\_lahir\_ibu');

            $table->string('pendidikan\_ibu');

            $table->string('pekerjaan\_ibu');

            $table->integer('nomor\_kontak\_ibu');

            $table->string('alamat\_ibu');

            $table->integer('rata\_rata\_penghasilan');

            $table->integer('nomor\_skhun');

            $table->integer('nomor\_ijazah');

            $table->integer('nilai\_total');

            $table->string('yatim');

            $table->string('asrama');

            $table->string('agk');

            $table->string('aktif');

            $table->string('nama\_arab');

            $table->string('online');

            $table->timestamps();

        });

    }

*/\*\**

*\* Reverse the migrations.*

*\**

*\* @return void*

*\*/*

    public *function* down()

    {

        Schema::dropIfExists('santris');

    }

};

Now run migration.

php artisan migrate

### **Step 5: Creating Models**

So now, In this step we have to create model for User and Santri table, so if you get fresh project then you have User Model have so just replace code and other you should create.

**app/Models/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;

use Spatie\Permission\Traits\HasRoles;

*class* User extends Authenticatable

{

    use HasApiTokens, HasFactory, Notifiable, HasRoles;

*/\*\**

*\* The attributes that are mass assignable.*

*\**

*\* @var array*

*\*/*

    protected $fillable = [

        'name',

        'email',

        'password',

    ];

*/\*\**

*\* The attributes that should be hidden for serialization.*

*\**

*\* @var array*

*\*/*

    protected $hidden = [

        'password',

        'remember\_token',

    ];

*/\*\**

*\* The attributes that should be cast.*

*\**

*\* @var array*

*\*/*

    protected $casts = [

        'email\_verified\_at' => 'datetime',

    ];

}

**php artisan make:model Santri**

**app/Models/Santri.php**

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

*class* Santri extends Model

{

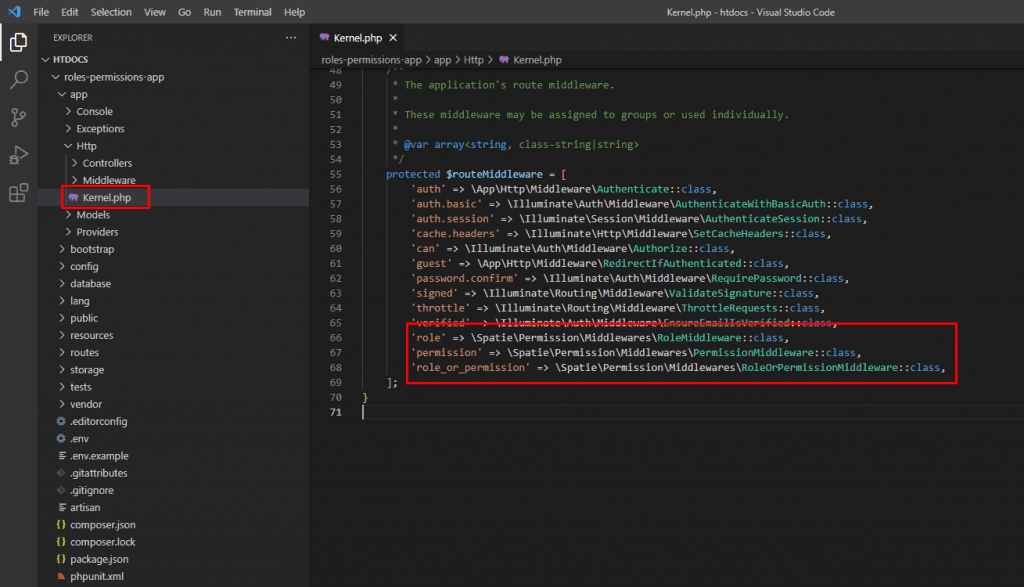
    use HasFactory;

    protected $guarded = ['id'];

}

### **Step 6: Adding Middleware**

Spatie package provide it’s in-built middleware that way we can use it simply and that is display as below. So, we have to add middleware in Kernel.php file this way :



**Adding Middleware**

**app/Http/Kernel.php**

………………………

    protected $routeMiddleware = [

………………………

        'role' => \Spatie\Permission\Middlewares\RoleMiddleware::class,

        'permission' => \Spatie\Permission\Middlewares\PermissionMiddleware::class,

        'role\_or\_permission' => \Spatie\Permission\Middlewares\RoleOrPermissionMiddleware::class,

    ];

### **Step 7: Creating Authentication**

Now we are going to create a authentication for our laravel 10 application. Run the following command to create a authentication for laravel 10 application.

composer require laravel/ui

Now need to create auth scaffolding we are using a bootstrap. Run the following command to do so.

php artisan ui bootstrap --auth

**Update vite.config.js with following code:**

import { defineConfig } from 'vite';

import laravel from 'laravel-vite-plugin';

import path from 'path'

export default defineConfig({

*plugins:* [

        laravel([

            'resources/js/app.js',

        ]),

    ],

*resolve:* {

*alias:* {

*'~bootstrap':* path.resolve(\_\_dirname, 'node\_modules/bootstrap'),

        }

    },

});

**resources/js/bootstrap.js**

import loadash from 'lodash'

window.\_ = loadash

import \* as Popper from '@popperjs/core'

window.Popper = Popper

import 'bootstrap'

*/\*\**

*\* We'll load the axios HTTP library which allows us to easily issue requests*

*\* to our Laravel back-end. This library automatically handles sending the*

*\* CSRF token as a header based on the value of the "XSRF" token cookie.*

*\*/*

import axios from 'axios'

window.axios = axios

window.axios.defaults.headers.common['X-Requested-With'] = 'XMLHttpRequest';

*/\*\**

*\* Echo exposes an expressive API for subscribing to channels and listening*

*\* for events that are broadcast by Laravel. Echo and event broadcasting*

*\* allows your team to easily build robust real-time web applications.*

*\*/*

*// import Echo from 'laravel-echo';*

*// window.Pusher = require('pusher-js');*

*// window.Echo = new Echo({*

*//     broadcaster: 'pusher',*

*//     key: process.env.MIX\_PUSHER\_APP\_KEY,*

*//     cluster: process.env.MIX\_PUSHER\_APP\_CLUSTER,*

*//     forceTLS: true*

*// });*

**resources/js/app.js**

import './bootstrap';

import '../sass/app.scss'

So now, we need to run npm command, otherwise you can not see better layout of login and register page.

**Install Node JS package (pastikan sudah terinstall node.js dengan node -v dan npm -v)**

npm install

Tambahan berdasarkan error yang ditemukan Laravel 10, mengenai lodash (<https://github.com/laravel/framework/issues/46128>)

npm i lodash

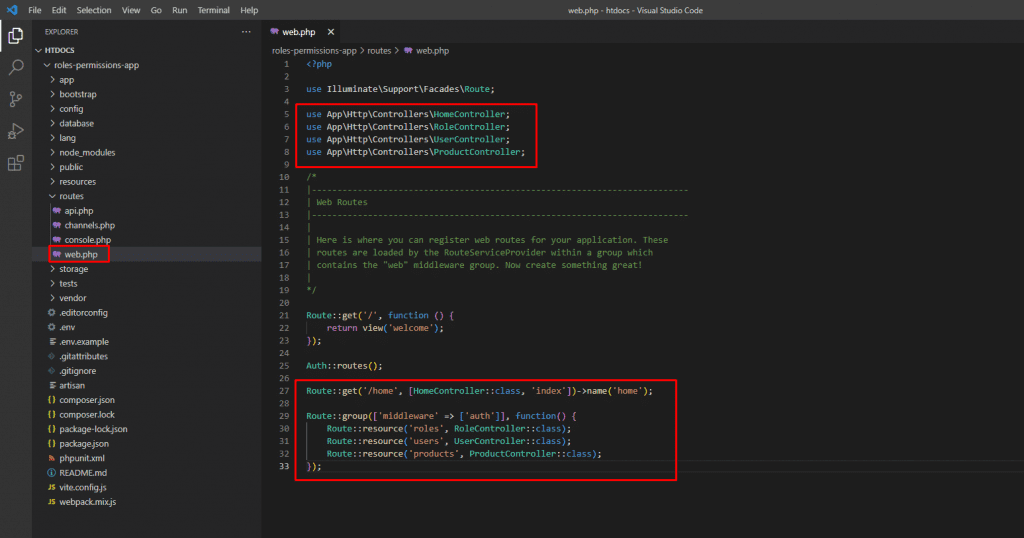
Now run the vite command and make it keep running.

npm run dev

Note: If error occurs then updated node to v16.16.0 and it will work

### **Step 8: Creating Routes**

We are going to create some routes for our application. Basically going to create  users module, santris module and roles module routes. we had also use middleware for our routes. Just add the route as show below.



**Creating Routes**

**web.php**

<?php

use Illuminate\Support\Facades\Auth;

use Illuminate\Support\Facades\Route;

use App\Http\Controllers\HomeController;

use App\Http\Controllers\RoleController;

use App\Http\Controllers\UserController;

use App\Http\Controllers\SantriController;

*/\**

*|--------------------------------------------------------------------------*

*| Web Routes*

*|--------------------------------------------------------------------------*

*|*

*| Here is where you can register web routes for your application. These*

*| routes are loaded by the RouteServiceProvider within a group which*

*| contains the "web" middleware group. Now create something great!*

*|*

*\*/*

Route::get('/', *function* () {

    return view('welcome');

});

Auth::routes();

Route::get('/home', [HomeController::class, 'index'])->name('home');

Route::group(['middleware' => ['auth']], *function*() {

    Route::resource('roles', RoleController::class);

    Route::resource('users', UserController::class);

    Route::resource('santris', SantriController::class);

});

### **Step 9: Adding Controllers**

Now we are going to create a controllers for users module, santris module and roles module. Create a controller inside app/Http/Controllers/ as show below.

**php artisan make:controller UserController**

**app/Http/Controllers/UserController.php**

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Http\Controllers\Controller;

use App\Models\User;

use Spatie\Permission\Models\Role;

use Illuminate\Support\Facades\DB;

use Illuminate\Support\Facades\Hash;

use Illuminate\Support\Arr;

*class* UserController extends Controller

{

*/\*\**

*\* Display a listing of the resource.*

*\**

*\* @return \Illuminate\Http\Response*

*\*/*

    public *function* index(Request $request)

    {

        $data = User::orderBy('id','DESC')->paginate(5);

        return view('users.index',compact('data'))

        ->with('i', ($request->input('page', 1) - 1) \* 5);

    }

*/\*\**

*\* Show the form for creating a new resource.*

*\**

*\* @return \Illuminate\Http\Response*

*\*/*

    public *function* create()

    {

        $roles = Role::pluck('name','name')->all();

        return view('users.create',compact('roles'));

    }

*/\*\**

*\* Store a newly created resource in storage.*

*\**

*\* @param  \Illuminate\Http\Request  $request*

*\* @return \Illuminate\Http\Response*

*\*/*

    public *function* store(Request $request)

    {

*$this*->validate($request, [

            'name' => 'required',

            'email' => 'required|email|unique:users,email',

            'password' => 'required|same:confirm-password',

            'roles' => 'required'

        ]);

        $input = $request->all();

        $input['password'] = Hash::make($input['password']);

        $user = User::create($input);

        $user->assignRole($request->input('roles'));

        return redirect()->route('users.index')

        ->with('success','User created successfully');

    }

*/\*\**

*\* Display the specified resource.*

*\**

*\* @param  int  $id*

*\* @return \Illuminate\Http\Response*

*\*/*

    public *function* show($id)

    {

        $user = User::find($id);

        return view('users.show',compact('user'));

    }

*/\*\**

*\* Show the form for editing the specified resource.*

*\**

*\* @param  int  $id*

*\* @return \Illuminate\Http\Response*

*\*/*

    public *function* edit($id)

    {

        $user = User::find($id);

        $roles = Role::pluck('name','name')->all();

        $userRole = $user->roles->pluck('name','name')->all();

        return view('users.edit',compact('user','roles','userRole'));

    }

*/\*\**

*\* Update the specified resource in storage.*

*\**

*\* @param  \Illuminate\Http\Request  $request*

*\* @param  int  $id*

*\* @return \Illuminate\Http\Response*

*\*/*

    public *function* update(Request $request, $id)

    {

*$this*->validate($request, [

            'name' => 'required',

            'email' => 'required|email|unique:users,email,'.$id,

            'password' => 'same:confirm-password',

            'roles' => 'required'

        ]);

        $input = $request->all();

        if(!empty($input['password'])){

            $input['password'] = Hash::make($input['password']);

        }else{

            $input = Arr::except($input,array('password'));

        }

        $user = User::find($id);

        $user->update($input);

        DB::table('model\_has\_roles')->where('model\_id',$id)->delete();

        $user->assignRole($request->input('roles'));

        return redirect()->route('users.index')

        ->with('success','User updated successfully');

    }

*/\*\**

*\* Remove the specified resource from storage.*

*\**

*\* @param  int  $id*

*\* @return \Illuminate\Http\Response*

*\*/*

    public *function* destroy($id)

    {

        User::find($id)->delete();

        return redirect()->route('users.index')

        ->with('success','User deleted successfully');

    }

}

**php artisan make:controller SantriController**

**app/Http/Controllers/SantriController.php**

<?php

namespace App\Http\Controllers;

use App\Models\Santri;

use Illuminate\Http\Request;

*class* SantriController extends Controller

{

*/\*\**

*\* Display a listing of the resource.*

*\**

*\* @return \Illuminate\Http\Response*

*\*/*

*function* \_\_construct()

    {

*$this*->middleware('permission:santri-list|santri-create|santri-edit|santri-delete', ['only' => ['index','show']]);

*$this*->middleware('permission:santri-create', ['only' => ['create','store']]);

*$this*->middleware('permission:santri-edit', ['only' => ['edit','update']]);

*$this*->middleware('permission:santri-delete', ['only' => ['destroy']]);

    }

*/\*\**

*\* Display a listing of the resource.*

*\**

*\* @return \Illuminate\Http\Response*

*\*/*

    public *function* index()

    {

        $santris = Santri::latest()->paginate(5);

        return view('santris.index',compact('santris'))

        ->with('i', (request()->input('page', 1) - 1) \* 5);

    }

*/\*\**

*\* Show the form for creating a new resource.*

*\**

*\* @return \Illuminate\Http\Response*

*\*/*

    public *function* create()

    {

        return view('santris.create');

    }

*/\*\**

*\* Store a newly created resource in storage.*

*\**

*\* @param  \Illuminate\Http\Request  $request*

*\* @return \Illuminate\Http\Response*

*\*/*

    public *function* store(Request $request)

    {

        request()->validate([

            'name' => 'required',

            'detail' => 'required',

        ]);

        Santri::create($request->all());

        return redirect()->route('santris.index')

        ->with('success','Santri created successfully.');

    }

*/\*\**

*\* Display the specified resource.*

*\**

*\* @param  \App\Santri  $santri*

*\* @return \Illuminate\Http\Response*

*\*/*

    public *function* show(Santri $santri)

    {

        return view('santris.show',compact('santri'));

    }

*/\*\**

*\* Show the form for editing the specified resource.*

*\**

*\* @param  \App\Santri  $santri*

*\* @return \Illuminate\Http\Response*

*\*/*

    public *function* edit(Santri $santri)

    {

        return view('santris.edit',compact('santri'));

    }

*/\*\**

*\* Update the specified resource in storage.*

*\**

*\* @param  \Illuminate\Http\Request  $request*

*\* @param  \App\Santri  $santri*

*\* @return \Illuminate\Http\Response*

*\*/*

    public *function* update(Request $request, Santri $santri)

    {

        request()->validate([

            'name' => 'required',

            'detail' => 'required',

        ]);

        $santri->update($request->all());

        return redirect()->route('santris.index')

        ->with('success','Santri updated successfully');

    }

*/\*\**

*\* Remove the specified resource from storage.*

*\**

*\* @param  \App\Santri  $santri*

*\* @return \Illuminate\Http\Response*

*\*/*

    public *function* destroy(Santri $santri)

    {

        $santri->delete();

        return redirect()->route('santris.index')

        ->with('success','Santri deleted successfully');

    }

}

**php artisan make:controller RoleController**

**app/Http/Controllers/RoleController.php**

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Http\Controllers\Controller;

use Spatie\Permission\Models\Role;

use Spatie\Permission\Models\Permission;

use Illuminate\Support\Facades\DB;

*class* RoleController extends Controller

{

*/\*\**

*\* Display a listing of the resource.*

*\**

*\* @return \Illuminate\Http\Response*

*\*/*

*function* \_\_construct()

    {

*$this*->middleware('permission:role-list|role-create|role-edit|role-delete', ['only' => ['index','store']]);

*$this*->middleware('permission:role-create', ['only' => ['create','store']]);

*$this*->middleware('permission:role-edit', ['only' => ['edit','update']]);

*$this*->middleware('permission:role-delete', ['only' => ['destroy']]);

    }

*/\*\**

*\* Display a listing of the resource.*

*\**

*\* @return \Illuminate\Http\Response*

*\*/*

    public *function* index(Request $request)

    {

        $roles = Role::orderBy('id','DESC')->paginate(5);

        return view('roles.index',compact('roles'))

        ->with('i', ($request->input('page', 1) - 1) \* 5);

    }

*/\*\**

*\* Show the form for creating a new resource.*

*\**

*\* @return \Illuminate\Http\Response*

*\*/*

    public *function* create()

    {

        $permission = Permission::get();

        return view('roles.create',compact('permission'));

    }

*/\*\**

*\* Store a newly created resource in storage.*

*\**

*\* @param  \Illuminate\Http\Request  $request*

*\* @return \Illuminate\Http\Response*

*\*/*

    public *function* store(Request $request)

    {

*$this*->validate($request, [

            'name' => 'required|unique:roles,name',

            'permission' => 'required',

        ]);

        $role = Role::create(['name' => $request->input('name')]);

        $role->syncPermissions($request->input('permission'));

        return redirect()->route('roles.index')

        ->with('success','Role created successfully');

    }

*/\*\**

*\* Display the specified resource.*

*\**

*\* @param  int  $id*

*\* @return \Illuminate\Http\Response*

*\*/*

    public *function* show($id)

    {

        $role = Role::find($id);

        $rolePermissions = Permission::join("role\_has\_permissions","role\_has\_permissions.permission\_id","=","permissions.id")

        ->where("role\_has\_permissions.role\_id",$id)

        ->get();

        return view('roles.show',compact('role','rolePermissions'));

    }

*/\*\**

*\* Show the form for editing the specified resource.*

*\**

*\* @param  int  $id*

*\* @return \Illuminate\Http\Response*

*\*/*

    public *function* edit($id)

    {

        $role = Role::find($id);

        $permission = Permission::get();

        $rolePermissions = DB::table("role\_has\_permissions")->where("role\_has\_permissions.role\_id",$id)

        ->pluck('role\_has\_permissions.permission\_id','role\_has\_permissions.permission\_id')

        ->all();

        return view('roles.edit',compact('role','permission','rolePermissions'));

    }

*/\*\**

*\* Update the specified resource in storage.*

*\**

*\* @param  \Illuminate\Http\Request  $request*

*\* @param  int  $id*

*\* @return \Illuminate\Http\Response*

*\*/*

    public *function* update(Request $request, $id)

    {

*$this*->validate($request, [

            'name' => 'required',

            'permission' => 'required',

        ]);

        $role = Role::find($id);

        $role->name = $request->input('name');

        $role->save();

        $role->syncPermissions($request->input('permission'));

        return redirect()->route('roles.index')

        ->with('success','Role updated successfully');

    }

*/\*\**

*\* Remove the specified resource from storage.*

*\**

*\* @param  int  $id*

*\* @return \Illuminate\Http\Response*

*\*/*

    public *function* destroy($id)

    {

        DB::table("roles")->where('id',$id)->delete();

        return redirect()->route('roles.index')

        ->with('success','Role deleted successfully');

    }

}

### **Step 10: Adding Blade Files**

Now in next step we are going to create a blade files. For display layouts, users module, roles module, santris module.

Lists of file we are going to create.

**Home Layout:**

* resources/views/layouts/app.blade.php

**User Blade Pages:**

* resources/views/users/index.blade.php
* resources/views/users/create.blade.php
* resources/views/users/edit.blade.php
* resources/views/users/show.blade.php

**Roles Blade Pages:**

* resources/views/roles/index.blade.php
* resources/views/roles/create.blade.php
* resources/views/roles/edit.blade.php
* resources/views/roles/show.blade.php

**Santris Blade Pages:**

* resources/views/santris/index.blade.php
* resources/views/santris/create.blade.php
* resources/views/santris/edit.blade.php
* resources/views/santris/show.blade.php

So let get started create blade’s files.

**resources/views/layouts/app.blade.php**

<html *lang*="{{ app()->getLocale() }}">

<head>

    <meta *charset*="utf-8">

    <meta *name*="viewport" *content*="width=device-width, initial-scale=1">

*<!-- CSRF Token -->*

    <meta *name*="csrf-token" *content*="{{ csrf\_token() }}">

    <title>{{ config('app.name', 'Laravel 9 User Roles and Permissions Tutorial Example') }}</title>

    @vite(['resources/js/app.js'])

*<!-- Fonts -->*

    <link *rel*="dns-prefetch" *href*="//fonts.gstatic.com">

    <link *href*="https://fonts.googleapis.com/css?family=Nunito" *rel*="stylesheet">

</head>

<body>

    <div *id*="app">

        <nav *class*="navbar navbar-expand-md navbar-light bg-white shadow-sm">

            <div *class*="container">

                <a *class*="navbar-brand" *href*="{{ url('/') }}">

                    Laravel 9 User Roles and Permissions Tutorial Example - LaravelTuts.com

                </a>

                <button *class*="navbar-toggler" *type*="button" *data-toggle*="collapse" *data-target*="#navbarSupportedContent" *aria-controls*="navbarSupportedContent" *aria-expanded*="false" *aria-label*="Toggle navigation">

                    <span *class*="navbar-toggler-icon"></span>

                </button>

                <div *class*="collapse navbar-collapse" *id*="navbarSupportedContent">

*<!-- Left Side Of Navbar -->*

                    <ul *class*="navbar-nav me-auto"></ul>

*<!-- Right Side Of Navbar -->*

                    <ul *class*="navbar-nav ms-auto">

*<!-- Authentication Links -->*

                        @guest

                        @if (Route::has('login'))

                        <li *class*="nav-item">

                            <a *class*="nav-link" *href*="{{ route('login') }}">{{ \_\_('Login') }}</a>

                        </li>

                        @endif

                        @if (Route::has('register'))

                        <li *class*="nav-item">

                            <a *class*="nav-link" *href*="{{ route('register') }}">{{ \_\_('Register') }}</a>

                        </li>

                        @endif

                        @else

                        <li><a *class*="nav-link" *href*="{{ route('users.index') }}">Manage Users</a></li>

                        <li><a *class*="nav-link" *href*="{{ route('roles.index') }}">Manage Role</a></li>

                        <li><a *class*="nav-link" *href*="{{ route('santris.index') }}">Manage Santri</a></li>

                        <li *class*="nav-item dropdown">

                            <a *id*="navbarDropdown" *class*="nav-link dropdown-toggle" *href*="#" *role*="button" *data-bs-toggle*="dropdown" *aria-haspopup*="true" *aria-expanded*="false" *v-pre*>

                                {{ Auth::user()->name }}

                            </a>

                            <div *class*="dropdown-menu dropdown-menu-end" *aria-labelledby*="navbarDropdown">

                                <a *class*="dropdown-item" *href*="{{ route('logout') }}"

*onclick*="event.preventDefault();

                                document.getElementById('logout-form').submit();">

                                {{ \_\_('Logout') }}

                            </a>

                            <form *id*="logout-form" *action*="{{ route('logout') }}" *method*="POST" *class*="d-none">

                                @csrf

                            </form>

                        </div>

                    </li>

                    @endguest

                </ul>

            </div>

        </div>

    </nav>

    <main *class*="py-4">

        <div *class*="container">

            @yield('content')

        </div>

    </main>

</div>

</body>

</html>

**resources/views/users/index.blade.php**

@extends('layouts.app')

@section('content')

<div *class*="row">

    <div *class*="col-lg-12 margin-tb">

        <div *class*="pull-left">

            <h2>Users Management</h2>

        </div>

        <div *class*="pull-right">

            <a *class*="btn btn-success" *href*="{{ route('users.create') }}"> Create New User</a>

        </div>

    </div>

</div>

@if ($message = Session::get('success'))

<div *class*="alert alert-success">

    <p>{{ $message }}</p>

</div>

@endif

<table *class*="table table-bordered">

    <tr>

        <th>No</th>

        <th>Name</th>

        <th>Email</th>

        <th>Roles</th>

        <th *width*="280px">Action</th>

    </tr>

    @foreach ($data as $key => $user)

    <tr>

        <td>{{ ++$i }}</td>

        <td>{{ $user->name }}</td>

        <td>{{ $user->email }}</td>

        <td>

            @if(!empty($user->getRoleNames()))

            @foreach($user->getRoleNames() as $v)

            <span *class*="badge rounded-pill bg-dark">{{ $v }}</span>

            @endforeach

            @endif

        </td>

        <td>

            <a *class*="btn btn-info" *href*="{{ route('users.show',$user->id) }}">Show</a>

            <a *class*="btn btn-primary" *href*="{{ route('users.edit',$user->id) }}">Edit</a>

            {!! Form::open(['method' => 'DELETE','route' => ['users.destroy', $user->id],'style'=>'display:inline']) !!}

            {!! Form::submit('Delete', ['class' => 'btn btn-danger']) !!}

            {!! Form::close() !!}

        </td>

    </tr>

    @endforeach

</table>

{!! $data->render() !!}

<p *class*="text-center text-primary"><small>Tutorial by LaravelTuts.com</small></p>

@endsection

**resources/views/users/create.blade.php**

@extends('layouts.app')

@section('content')

<div *class*="row">

    <div *class*="col-lg-12 margin-tb">

        <div *class*="pull-left">

            <h2>Create New User</h2>

        </div>

        <div *class*="pull-right">

            <a *class*="btn btn-primary" *href*="{{ route('users.index') }}"> Back</a>

        </div>

    </div>

</div>

@if (count($errors) > 0)

<div *class*="alert alert-danger">

    <strong>Whoops!</strong> There were some problems with your input.<br><br>

    <ul>

        @foreach ($errors->all() as $error)

        <li>{{ $error }}</li>

        @endforeach

    </ul>

</div>

@endif

{!! Form::open(array('route' => 'users.store','method'=>'POST')) !!}

<div *class*="row">

    <div *class*="col-xs-12 col-sm-12 col-md-12">

        <div *class*="form-group">

            <strong>Name:</strong>

            {!! Form::text('name', null, array('placeholder' => 'Name','class' => 'form-control')) !!}

        </div>

    </div>

    <div *class*="col-xs-12 col-sm-12 col-md-12">

        <div *class*="form-group">

            <strong>Email:</strong>

            {!! Form::text('email', null, array('placeholder' => 'Email','class' => 'form-control')) !!}

        </div>

    </div>

    <div *class*="col-xs-12 col-sm-12 col-md-12">

        <div *class*="form-group">

            <strong>Password:</strong>

            {!! Form::password('password', array('placeholder' => 'Password','class' => 'form-control')) !!}

        </div>

    </div>

    <div *class*="col-xs-12 col-sm-12 col-md-12">

        <div *class*="form-group">

            <strong>Confirm Password:</strong>

            {!! Form::password('confirm-password', array('placeholder' => 'Confirm Password','class' => 'form-control')) !!}

        </div>

    </div>

    <div *class*="col-xs-12 col-sm-12 col-md-12">

        <div *class*="form-group">

            <strong>Role:</strong>

            {!! Form::select('roles[]', $roles,[], array('class' => 'form-control','multiple')) !!}

        </div>

    </div>

    <div *class*="col-xs-12 col-sm-12 col-md-12 text-center">

        <button *type*="submit" *class*="btn btn-primary">Submit</button>

    </div>

</div>

{!! Form::close() !!}

<p *class*="text-center text-primary"><small>Tutorial by LaravelTuts.com</small></p>

@endsection

**resources/views/users/edit.blade.php**

@extends('layouts.app')

@section('content')

<div *class*="row">

    <div *class*="col-lg-12 margin-tb">

        <div *class*="pull-left">

            <h2>Edit New User</h2>

        </div>

        <div *class*="pull-right">

            <a *class*="btn btn-primary" *href*="{{ route('users.index') }}"> Back</a>

        </div>

    </div>

</div>

@if (count($errors) > 0)

<div *class*="alert alert-danger">

    <strong>Whoops!</strong> There were some problems with your input.<br><br>

    <ul>

        @foreach ($errors->all() as $error)

        <li>{{ $error }}</li>

        @endforeach

    </ul>

</div>

@endif

{!! Form::model($user, ['method' => 'PATCH','route' => ['users.update', $user->id]]) !!}

<div *class*="row">

    <div *class*="col-xs-12 col-sm-12 col-md-12">

        <div *class*="form-group">

            <strong>Name:</strong>

            {!! Form::text('name', null, array('placeholder' => 'Name','class' => 'form-control')) !!}

        </div>

    </div>

    <div *class*="col-xs-12 col-sm-12 col-md-12">

        <div *class*="form-group">

            <strong>Email:</strong>

            {!! Form::text('email', null, array('placeholder' => 'Email','class' => 'form-control')) !!}

        </div>

    </div>

    <div *class*="col-xs-12 col-sm-12 col-md-12">

        <div *class*="form-group">

            <strong>Password:</strong>

            {!! Form::password('password', array('placeholder' => 'Password','class' => 'form-control')) !!}

        </div>

    </div>

    <div *class*="col-xs-12 col-sm-12 col-md-12">

        <div *class*="form-group">

            <strong>Confirm Password:</strong>

            {!! Form::password('confirm-password', array('placeholder' => 'Confirm Password','class' => 'form-control')) !!}

        </div>

    </div>

    <div *class*="col-xs-12 col-sm-12 col-md-12">

        <div *class*="form-group">

            <strong>Role:</strong>

            {!! Form::select('roles[]', $roles,$userRole, array('class' => 'form-control','multiple')) !!}

        </div>

    </div>

    <div *class*="col-xs-12 col-sm-12 col-md-12 text-center">

        <button *type*="submit" *class*="btn btn-primary">Submit</button>

    </div>

</div>

{!! Form::close() !!}

<p *class*="text-center text-primary"><small>Tutorial by LaravelTuts.com</small></p>

@endsection

**resources/views/users/show.blade.php**

@extends('layouts.app')

@section('content')

<div *class*="row">

    <div *class*="col-lg-12 margin-tb">

        <div *class*="pull-left">

            <h2> Show User</h2>

        </div>

        <div *class*="pull-right">

            <a *class*="btn btn-primary" *href*="{{ route('users.index') }}"> Back</a>

        </div>

    </div>

</div>

<div *class*="row">

    <div *class*="col-xs-12 col-sm-12 col-md-12">

        <div *class*="form-group">

            <strong>Name:</strong>

            {{ $user->name }}

        </div>

    </div>

    <div *class*="col-xs-12 col-sm-12 col-md-12">

        <div *class*="form-group">

            <strong>Email:</strong>

            {{ $user->email }}

        </div>

    </div>

    <div *class*="col-xs-12 col-sm-12 col-md-12">

        <div *class*="form-group">

            <strong>Roles:</strong>

            @if(!empty($user->getRoleNames()))

            @foreach($user->getRoleNames() as $v)

            <span *class*="badge rounded-pill bg-dark">{{ $v }}</span>

            @endforeach

            @endif

        </div>

    </div>

</div>

@endsection

**resources/views/roles/index.blade.php**

@extends('layouts.app')

@section('content')

<div *class*="row">

    <div *class*="col-lg-12 margin-tb">

        <div *class*="pull-left">

            <h2>Role Management</h2>

        </div>

        <div *class*="pull-right">

            @can('role-create')

            <a *class*="btn btn-success" *href*="{{ route('roles.create') }}"> Create New Role</a>

            @endcan

        </div>

    </div>

</div>

@if ($message = Session::get('success'))

<div *class*="alert alert-success">

    <p>{{ $message }}</p>

</div>

@endif

<table *class*="table table-bordered">

    <tr>

        <th>No</th>

        <th>Name</th>

        <th *width*="280px">Action</th>

    </tr>

    @foreach ($roles as $key => $role)

    <tr>

        <td>{{ ++$i }}</td>

        <td>{{ $role->name }}</td>

        <td>

            <a *class*="btn btn-info" *href*="{{ route('roles.show',$role->id) }}">Show</a>

            @can('role-edit')

            <a *class*="btn btn-primary" *href*="{{ route('roles.edit',$role->id) }}">Edit</a>

            @endcan

            @can('role-delete')

            {!! Form::open(['method' => 'DELETE','route' => ['roles.destroy', $role->id],'style'=>'display:inline']) !!}

            {!! Form::submit('Delete', ['class' => 'btn btn-danger']) !!}

            {!! Form::close() !!}

            @endcan

        </td>

    </tr>

    @endforeach

</table>

{!! $roles->render() !!}

<p *class*="text-center text-primary"><small>Tutorial by LaravelTuts.com</small></p>

@endsection

**resources/views/roles/create.blade.php**

@extends('layouts.app')

@section('content')

<div *class*="row">

    <div *class*="col-lg-12 margin-tb">

        <div *class*="pull-left">

            <h2>Create New Role</h2>

        </div>

        <div *class*="pull-right">

            <a *class*="btn btn-primary" *href*="{{ route('roles.index') }}"> Back</a>

        </div>

    </div>

</div>

@if (count($errors) > 0)

<div *class*="alert alert-danger">

    <strong>Whoops!</strong> There were some problems with your input.<br><br>

    <ul>

        @foreach ($errors->all() as $error)

        <li>{{ $error }}</li>

        @endforeach

    </ul>

</div>

@endif

{!! Form::open(array('route' => 'roles.store','method'=>'POST')) !!}

<div *class*="row">

    <div *class*="col-xs-12 col-sm-12 col-md-12">

        <div *class*="form-group">

            <strong>Name:</strong>

            {!! Form::text('name', null, array('placeholder' => 'Name','class' => 'form-control')) !!}

        </div>

    </div>

    <div *class*="col-xs-12 col-sm-12 col-md-12">

        <div *class*="form-group">

            <strong>Permission:</strong>

            <br/>

            @foreach($permission as $value)

            <label>{{ Form::checkbox('permission[]', $value->id, false, array('class' => 'name')) }}

                {{ $value->name }}</label>

                <br/>

                @endforeach

            </div>

        </div>

        <div *class*="col-xs-12 col-sm-12 col-md-12 text-center">

            <button *type*="submit" *class*="btn btn-primary">Submit</button>

        </div>

    </div>

    {!! Form::close() !!}

    <p *class*="text-center text-primary"><small>Tutorial by LaravelTuts.com</small></p>

    @endsection

**resources/views/roles/edit.blade.php**

@extends('layouts.app')

@section('content')

<div *class*="row">

    <div *class*="col-lg-12 margin-tb">

        <div *class*="pull-left">

            <h2>Edit Role</h2>

        </div>

        <div *class*="pull-right">

            <a *class*="btn btn-primary" *href*="{{ route('roles.index') }}"> Back</a>

        </div>

    </div>

</div>

@if (count($errors) > 0)

<div *class*="alert alert-danger">

    <strong>Whoops!</strong> There were some problems with your input.<br><br>

    <ul>

        @foreach ($errors->all() as $error)

        <li>{{ $error }}</li>

        @endforeach

    </ul>

</div>

@endif

{!! Form::model($role, ['method' => 'PATCH','route' => ['roles.update', $role->id]]) !!}

<div *class*="row">

    <div *class*="col-xs-12 col-sm-12 col-md-12">

        <div *class*="form-group">

            <strong>Name:</strong>

            {!! Form::text('name', null, array('placeholder' => 'Name','class' => 'form-control')) !!}

        </div>

    </div>

    <div *class*="col-xs-12 col-sm-12 col-md-12">

        <div *class*="form-group">

            <strong>Permission:</strong>

            <br/>

            @foreach($permission as $value)

            <label>{{ Form::checkbox('permission[]', $value->id, in\_array($value->id, $rolePermissions) ? true : false, array('class' => 'name')) }}

                {{ $value->name }}</label>

                <br/>

                @endforeach

            </div>

        </div>

        <div *class*="col-xs-12 col-sm-12 col-md-12 text-center">

            <button *type*="submit" *class*="btn btn-primary">Submit</button>

        </div>

    </div>

    {!! Form::close() !!}

    @endsection

    <p *class*="text-center text-primary"><small>Tutorial by LaravelTuts.com</small></p>

**resources/views/roles/show.blade.php**

@extends('layouts.app')

@section('content')

<div *class*="row">

    <div *class*="col-lg-12 margin-tb">

        <div *class*="pull-left">

            <h2> Show Role</h2>

        </div>

        <div *class*="pull-right">

            <a *class*="btn btn-primary" *href*="{{ route('roles.index') }}"> Back</a>

        </div>

    </div>

</div>

<div *class*="row">

    <div *class*="col-xs-12 col-sm-12 col-md-12">

        <div *class*="form-group">

            <strong>Name:</strong>

            {{ $role->name }}

        </div>

    </div>

    <div *class*="col-xs-12 col-sm-12 col-md-12">

        <div *class*="form-group">

            <strong>Permissions:</strong>

            @if(!empty($rolePermissions))

            @foreach($rolePermissions as $v)

            <label *class*="label label-success">{{ $v->name }},</label>

            @endforeach

            @endif

        </div>

    </div>

</div>

@endsection

**resources/views/santris/index.blade.php**

@extends('layouts.app')

@section('content')

<div *class*="row">

    <div *class*="col-lg-12 margin-tb">

        <div *class*="pull-left">

            <h2>Santri</h2>

        </div>

        <div *class*="pull-right">

            @can('santri-create')

            <a *class*="btn btn-success" *href*="{{ route('santris.create') }}"> Create New Santri</a>

            @endcan

        </div>

    </div>

</div>

@if ($message = Session::get('success'))

<div *class*="alert alert-success">

    <p>{{ $message }}</p>

</div>

@endif

<table *class*="table table-bordered">

    <tr>

        <th>No</th>

        <th>Name</th>

        <th>Details</th>

        <th *width*="280px">Action</th>

    </tr>

    @foreach ($santris as $santri)

    <tr>

        <td>{{ ++$i }}</td>

        <td>{{ $santri->name }}</td>

        <td>{{ $santri->detail }}</td>

        <td>

            <form *action*="{{ route('santris.destroy',$santri->id) }}" *method*="POST">

                <a *class*="btn btn-info" *href*="{{ route('santris.show',$santri->id) }}">Show</a>

                @can('santri-edit')

                <a *class*="btn btn-primary" *href*="{{ route('santris.edit',$santri->id) }}">Edit</a>

                @endcan

                @csrf

                @method('DELETE')

                @can('santri-delete')

                <button *type*="submit" *class*="btn btn-danger">Delete</button>

                @endcan

            </form>

        </td>

    </tr>

    @endforeach

</table>

{!! $santris->links() !!}

<p *class*="text-center text-primary"><small>Tutorial by LaravelTuts.com</small></p>

@endsection

**resources/views/santris/create.blade.php**

@extends('layouts.app')

@section('content')

<div *class*="row">

    <div *class*="col-lg-12 margin-tb">

        <div *class*="pull-left">

            <h2>Add New Santri</h2>

        </div>

        <div *class*="pull-right">

            <a *class*="btn btn-primary" *href*="{{ route('santris.index') }}"> Back</a>

        </div>

    </div>

</div>

@if ($errors->any())

<div *class*="alert alert-danger">

    <strong>Whoops!</strong> There were some problems with your input.<br><br>

    <ul>

        @foreach ($errors->all() as $error)

        <li>{{ $error }}</li>

        @endforeach

    </ul>

</div>

@endif

<form *action*="{{ route('santris.store') }}" *method*="POST">

    @csrf

    <div *class*="row">

        <div *class*="col-xs-12 col-sm-12 col-md-12">

            <div *class*="form-group">

                <strong>Name:</strong>

                <input *type*="text" *name*="name" *class*="form-control" *placeholder*="Name">

            </div>

        </div>

        <div *class*="col-xs-12 col-sm-12 col-md-12">

            <div *class*="form-group">

                <strong>Detail:</strong>

                <textarea *class*="form-control" *style*="height:150px" *name*="detail" *placeholder*="Detail"></textarea>

            </div>

        </div>

        <div *class*="col-xs-12 col-sm-12 col-md-12 text-center">

            <button *type*="submit" *class*="btn btn-primary">Submit</button>

        </div>

    </div>

</form>

<p *class*="text-center text-primary"><small>Tutorial by LaravelTuts.com</small></p>

@endsection

**resources/views/santris/edit.blade.php**

@extends('layouts.app')

@section('content')

<div *class*="row">

    <div *class*="col-lg-12 margin-tb">

        <div *class*="pull-left">

            <h2>Edit Santri</h2>

        </div>

        <div *class*="pull-right">

            <a *class*="btn btn-primary" *href*="{{ route('santris.index') }}"> Back</a>

        </div>

    </div>

</div>

@if ($errors->any())

<div *class*="alert alert-danger">

    <strong>Whoops!</strong> There were some problems with your input.<br><br>

    <ul>

        @foreach ($errors->all() as $error)

        <li>{{ $error }}</li>

        @endforeach

    </ul>

</div>

@endif

<form *action*="{{ route('santris.update',$santri->id) }}" *method*="POST">

    @csrf

    @method('PUT')

    <div *class*="row">

        <div *class*="col-xs-12 col-sm-12 col-md-12">

            <div *class*="form-group">

                <strong>Name:</strong>

                <input *type*="text" *name*="name" *value*="{{ $santri->name }}" *class*="form-control" *placeholder*="Name">

            </div>

        </div>

        <div *class*="col-xs-12 col-sm-12 col-md-12">

            <div *class*="form-group">

                <strong>Detail:</strong>

                <textarea *class*="form-control" *style*="height:150px" *name*="detail" *placeholder*="Detail">{{ $santri->detail }}</textarea>

            </div>

        </div>

        <div *class*="col-xs-12 col-sm-12 col-md-12 text-center">

            <button *type*="submit" *class*="btn btn-primary">Submit</button>

        </div>

    </div>

</form>

<p *class*="text-center text-primary"><small>Tutorial by LaravelTuts.com</small></p>

@endsection

**resources/views/santris/show.blade.php**

@extends('layouts.app')

@section('content')

<div *class*="row">

    <div *class*="col-lg-12 margin-tb">

        <div *class*="pull-left">

            <h2> Show Santri</h2>

        </div>

        <div *class*="pull-right">

            <a *class*="btn btn-primary" *href*="{{ route('santris.index') }}"> Back</a>

        </div>

    </div>

</div>

<div *class*="row">

    <div *class*="col-xs-12 col-sm-12 col-md-12">

        <div *class*="form-group">

            <strong>Name:</strong>

            {{ $santri->name }}

        </div>

    </div>

    <div *class*="col-xs-12 col-sm-12 col-md-12">

        <div *class*="form-group">

            <strong>Details:</strong>

            {{ $santri->detail }}

        </div>

    </div>

</div>

@endsection

<p *class*="text-center text-primary"><small>Tutorial by LaravelTuts.com</small></p>

### **Step 11: Creating Seeder For Permissions and Admin User**

Finally Now we are create a seeder for permissions.  Right now we have fixed permission so we create using seeder as listed bellow, but if you can add more permission as you want.

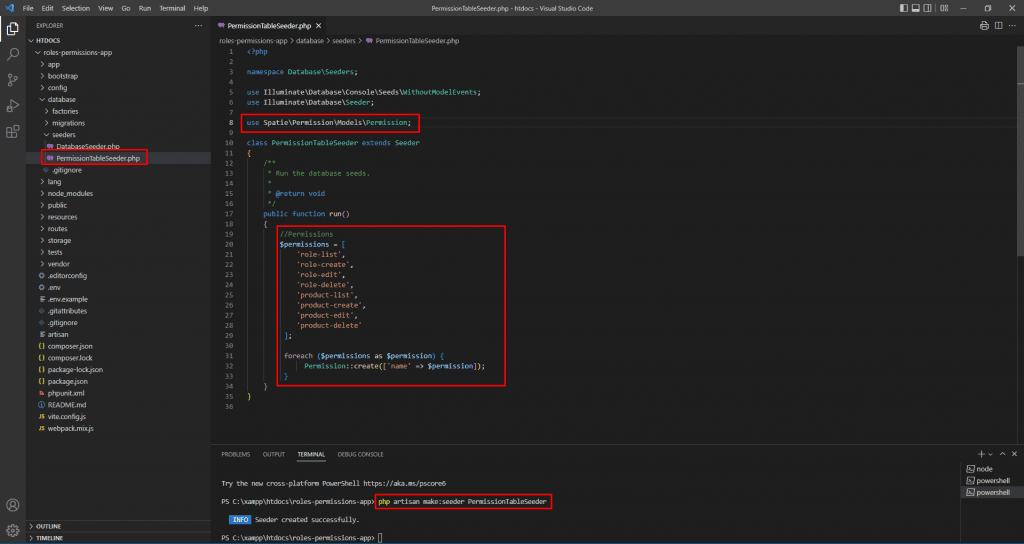
**Permission Lists:**

* role-list
* role-create
* role-edit
* role-delete
* santri-list
* santri-create
* santri-edit
* santri-delete

So now, First create a Permission seeder with below command.

php artisan make:seeder PermissionTableSeeder

And add the following code to the seeder file.



**Permission Seeder**

**database/seeders/PermissionTableSeeder.php**

<?php

namespace Database\Seeders;

use Illuminate\Database\Console\Seeds\WithoutModelEvents;

use Illuminate\Database\Seeder;

use Spatie\Permission\Models\Permission;

*class* PermissionTableSeeder extends Seeder

{

*/\*\**

*\* Run the database seeds.*

*\**

*\* @return void*

*\*/*

    public *function* run()

    {

*//Permissions*

        $permissions = [

            'role-list',

            'role-create',

            'role-edit',

            'role-delete',

            'santri-list',

            'santri-create',

            'santri-edit',

            'santri-delete'

        ];

        foreach ($permissions as $permission) {

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

        }

    }

}

Now, run the following command for Permission seeder.

php artisan db:seed --class=PermissionTableSeeder

So Now are creating a new Seeder for admin. Run the following code to create admin seeder.

php artisan make:seeder CreateAdminUserSeeder

**database/seeders/CreateAdminUserSeeder.php**

<?php

namespace Database\Seeders;

use Illuminate\Database\Console\Seeds\WithoutModelEvents;

use Illuminate\Database\Seeder;

use App\Models\User;

use Spatie\Permission\Models\Role;

use Spatie\Permission\Models\Permission;

*class* CreateAdminUserSeeder extends Seeder

{

*/\*\**

*\* Run the database seeds.*

*\**

*\* @return void*

*\*/*

    public *function* run()

    {

*//Admin Seeder*

        $user = User::create([

            'name' => 'LaravelTuts',

            'email' => 'admin@laraveltuts.com',

            'password' => bcrypt('password')

        ]);

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

        $permissions = Permission::pluck('id','id')->all();

        $role->syncPermissions($permissions);

        $user->assignRole([$role->id]);

    }

}

Run the following command to seed admin seeder.

php artisan db:seed --class=CreateAdminUserSeeder

### **Step 12: Testing**

So now everything is done we can now test our laravel 10 application. Now, Run the following command to start laravel server:

php artisan serve

So now open the following URL in any web browser:

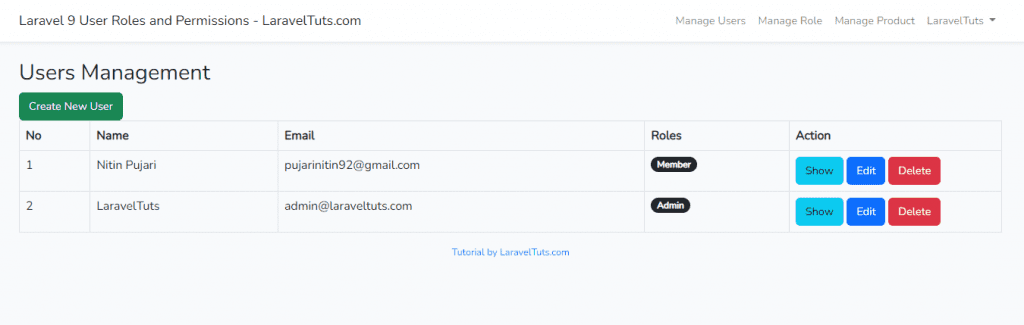
http://127.0.0.1:8000

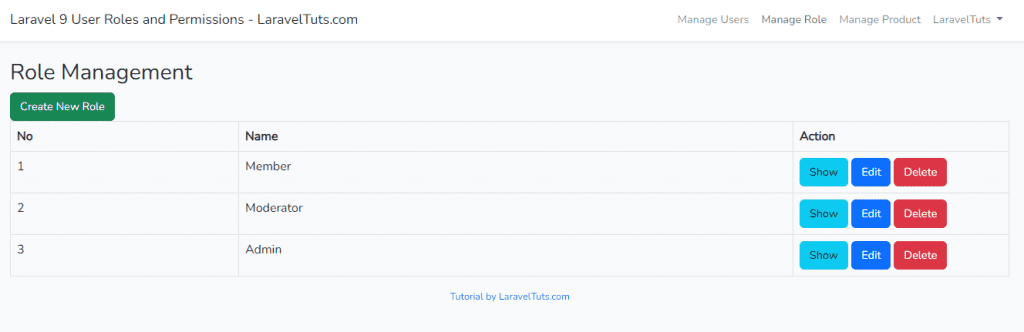
Now you can login with your admin details:

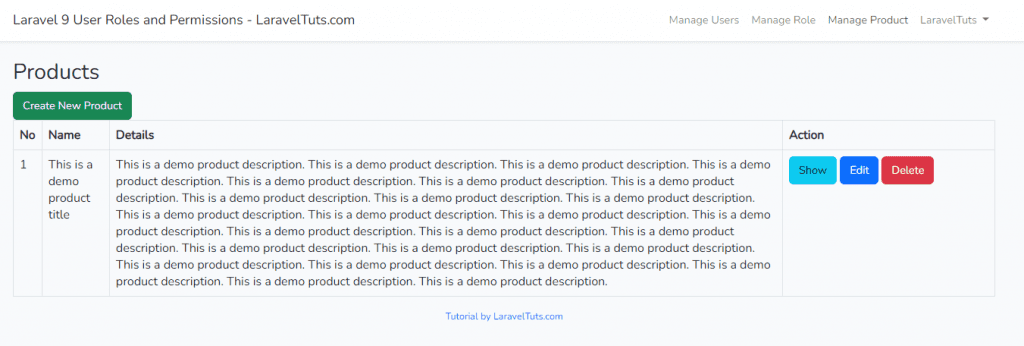
Email: admin@laraveltuts.com

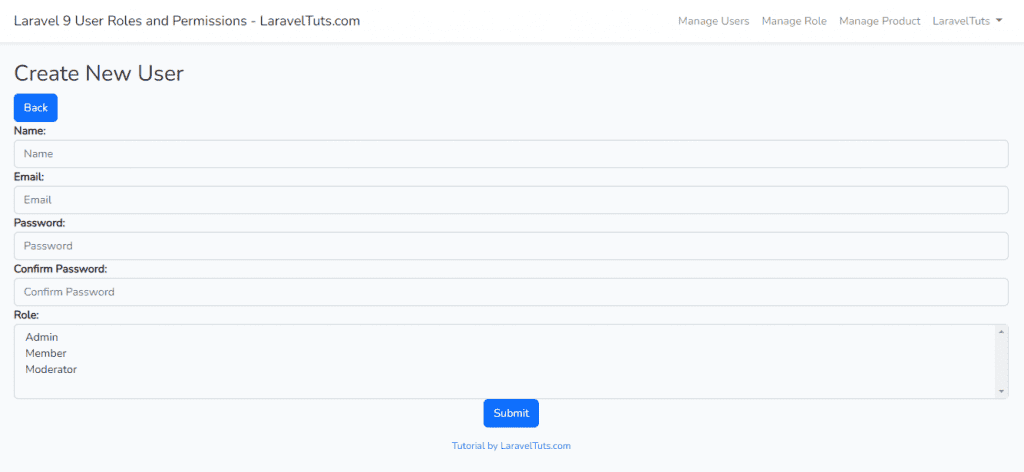
Password: password

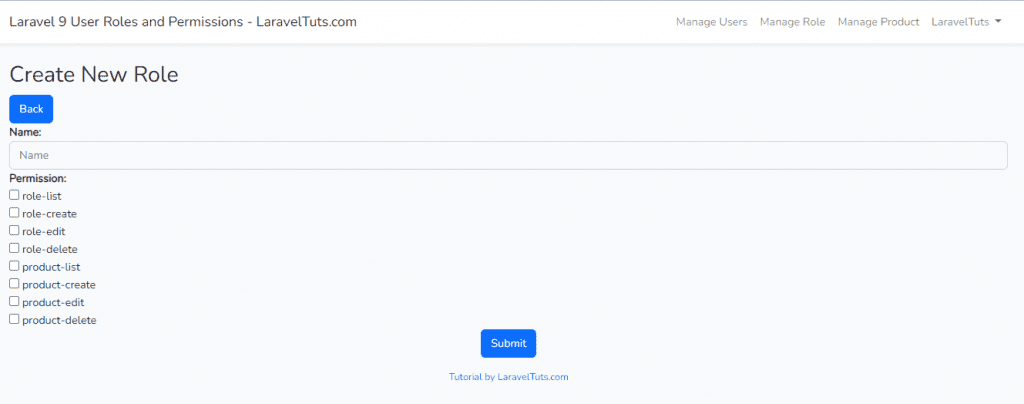
**Previews:**

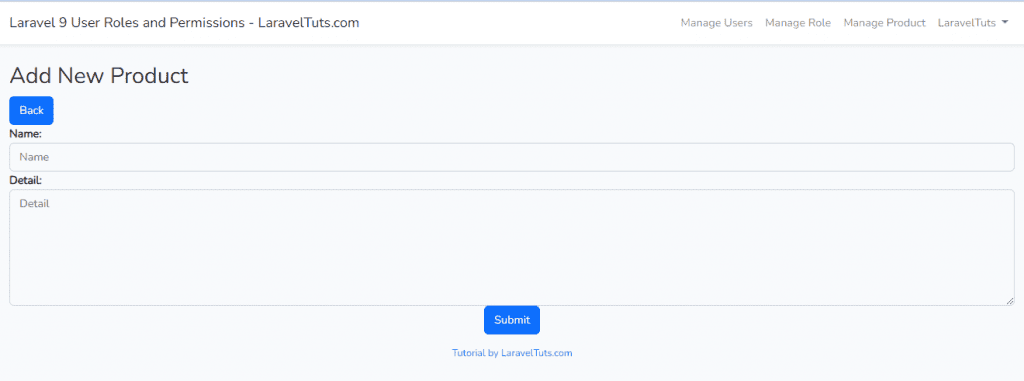
**User management**

**Role management**

**Products management**

**Create User**

**Create Role**

**Create Product**

### **Step 13: Conclusion**

Today, We had learn **Laravel 10 User Roles and Permissions Tutorial Example**. Hope this tutorial helped you with learning Laravel 10. So now, If you have any question you can ask us at comment section below. If you like the tutorial please subscribe our [YouTube Channel](https://www.youtube.com/channel/UCwIbFoGoht3Dom9Q1NkOS3A) and follow us on social network [Facebook](https://www.facebook.com/laraveltutscom) and [Instagram](https://www.instagram.com/laraveltuts/).