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

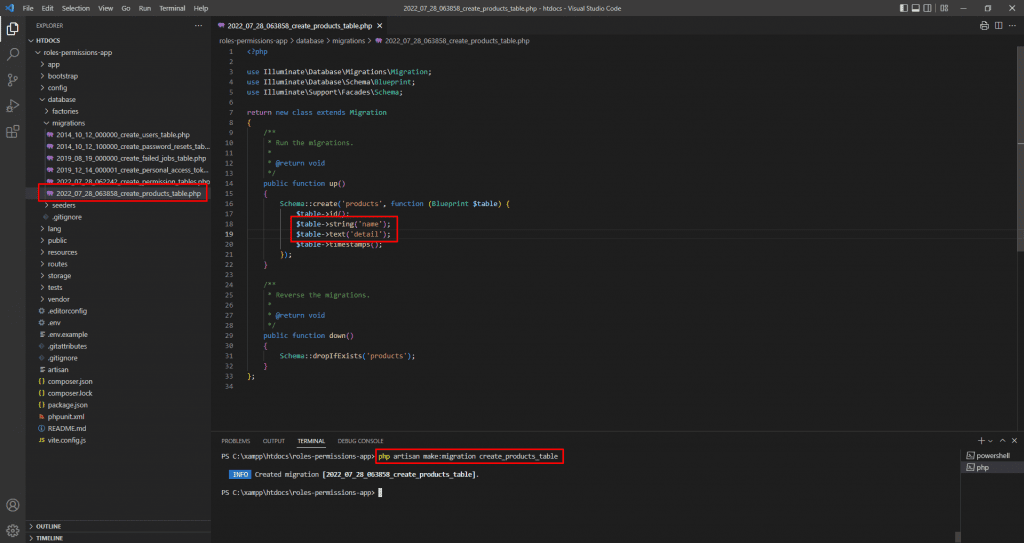
* **Step 4:**Creating Database and Product Migration
* **Step 5:**Creating Models
* **Step 8:**Creating Routes
* **Step 9:**Adding Controllers
* **Step 10:**Adding Blade Files
* **Step 11:**Creating Seeder For Permissions
* **Step 12:**Testing
* **Step 13:**Conclusion

### **Step 4: Product Migration**

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

php artisan make:migration create\_products\_table

Now add the following fields to products migration.



**Products Migration**

**create\_products\_table.php**

<?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('products', *function* (*Blueprint* $table) {

            $table->id();

            $table->string('name');

            $table->text('detail');

            $table->timestamps();

        });

    }

    /\*\*

     \* Reverse the migrations.

     \*

     \* @return void

     \*/

    public *function* down()

    {

*Schema*::dropIfExists('products');

    }

};

Now run migration.

php artisan migrate

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

So now, In this step we have to create model for Product table.

**php artisan make:model Product**

**app/Models/Product.php**

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\*HasFactory*;

use Illuminate\Database\Eloquent\*Model*;

*class* Product extends *Model*

{

    use *HasFactory*;

    /\*\*

     \* The attributes that are mass assignable.

     \*

     \* @var array

     \*/

    protected $fillable = [

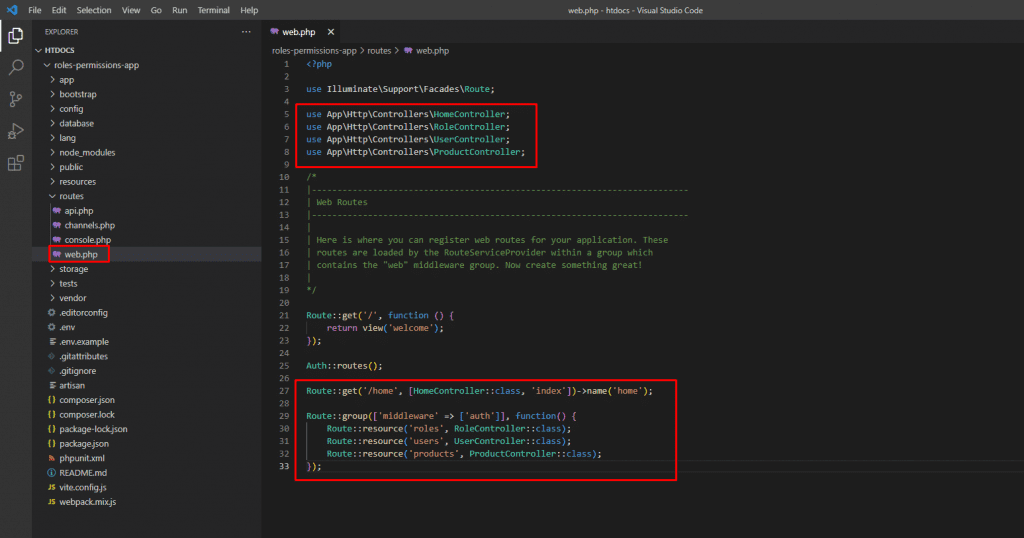
        'name', 'detail'

    ];

}

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

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



**Creating Routes**

**web.php**

---

use App\Http\Controllers\*ProductController*;

---

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

---

*Route*::resource('products', *ProductController*::class);

});

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

Now we are going to create a controllers for products module. Create a controller inside app/Http/Controllers/ as show below.

**php artisan make:controller ProductController app/Http/Controllers/ProductController.php**

<?php

namespace App\Http\Controllers;

use App\Models\*Product*;

use Illuminate\Http\*Request*;

*class* ProductController extends *Controller*

{

    /\*\*

     \* Display a listing of the resource.

     \*

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

     \*/

*function* \_\_construct()

    {

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

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

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

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

    }

    /\*\*

     \* Display a listing of the resource.

     \*

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

     \*/

    public *function* index()

    {

        $products = *Product*::latest()->paginate(5);

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

            ->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('products.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',

        ]);

*Product*::create($request->all());

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

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

    }

    /\*\*

     \* Display the specified resource.

     \*

     \* @param  \App\*Product*  $product

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

     \*/

    public *function* show(*Product* $product)

    {

        return view('products.show',compact('product'));

    }

    /\*\*

     \* Show the form for editing the specified resource.

     \*

     \* @param  \App\*Product*  $product

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

     \*/

    public *function* edit(*Product* $product)

    {

        return view('products.edit',compact('product'));

    }

    /\*\*

     \* Update the specified resource in storage.

     \*

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

     \* @param  \App\*Product*  $product

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

     \*/

    public *function* update(*Request* $request, *Product* $product)

    {

         request()->validate([

            'name' => 'required',

            'detail' => 'required',

        ]);

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

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

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

    }

    /\*\*

     \* Remove the specified resource from storage.

     \*

     \* @param  \App\*Product*  $product

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

     \*/

    public *function* destroy(*Product* $product)

    {

        $product->delete();

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

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

    }

}

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

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

Lists of file we are going to create.

**Home Layout:**

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

**Products Blade Pages:**

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

So let get started create blade’s files.

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

---

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

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

---

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

@extends('layouts.app')

@section('content')

    <div class="row">

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

            <div class="pull-left">

                <h2>Products</h2>

            </div>

            <div class="pull-right">

                @can('product-create')

                <a class="btn btn-success" href="{{ route('products.create') }}"> Create New Product</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 ($products as $product)

        <tr>

            <td>{{ ++$i }}</td>

            <td>{{ $product->name }}</td>

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

            <td>

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

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

                    @can('product-edit')

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

                    @endcan

                    @csrf

                    @method('DELETE')

                    @can('product-delete')

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

                    @endcan

                </form>

            </td>

        </tr>

        @endforeach

    </table>

    {!! $products->links() !!}

    <p class="text-center text-primary"><small>SISPPIT</small></p>

@endsection

**resources/views/products/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 Product</h2>

            </div>

            <div class="pull-right">

                <a class="btn btn-primary" href="{{ route('products.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('products.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>SISPPIT</small></p>

@endsection

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

@extends('layouts.app')

@section('content')

    <div class="row">

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

            <div class="pull-left">

                <h2>Edit Product</h2>

            </div>

            <div class="pull-right">

                <a class="btn btn-primary" href="{{ route('products.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('products.update',$product->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="{{ $product->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">{{ $product->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>SISPPIT</small></p>

@endsection

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

@extends('layouts.app')

@section('content')

    <div class="row">

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

            <div class="pull-left">

                <h2> Show Product</h2>

            </div>

            <div class="pull-right">

                <a class="btn btn-primary" href="{{ route('products.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>

                {{ $product->name }}

            </div>

        </div>

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

            <div class="form-group">

                <strong>Details:</strong>

                {{ $product->detail }}

            </div>

        </div>

    </div>

@endsection

<p class="text-center text-primary"><small>SISPPIT</small></p>

### **Step 11: Creating Seeder For Permissions**

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.

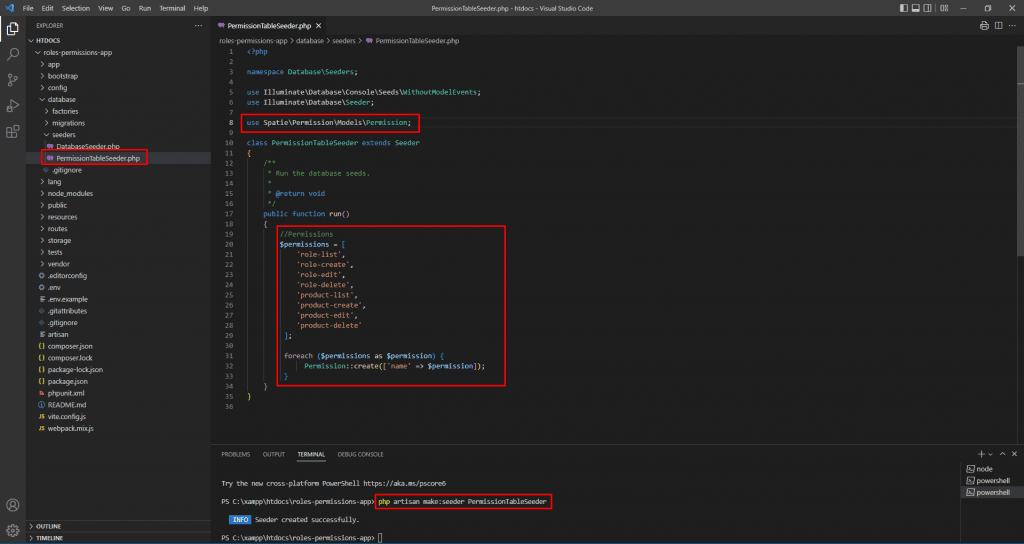
**Permission Lists:**

* product-list
* product -create
* product -edit
* product -delete

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

**php artisan make:seeder PermissionTableSeeder2**

And add the following code to the seeder file.



**Permission Seeder**

**database/seeders/PermissionTableSeeder2.php**

<?php

namespace Database\Seeders;

use Illuminate\Database\Console\Seeds\*WithoutModelEvents*;

use Illuminate\Database\*Seeder*;

use Spatie\Permission\Models\*Permission*;

*class* PermissionTableSeeder2 extends *Seeder*

{

    /\*\*

    \* Run the database seeds.

    \*

    \* @return void

    \*/

    public *function* run()

    {

        //Permissions

        $permissions = [

            'product-list',

            'product-create',

            'product-edit',

            'product-delete'

        ];

        foreach ($permissions as $permission) {

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

        }

    }

}

Now, run the following command for Permission seeder.

**php artisan db:seed --class=PermissionTableSeeder2**

### **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:

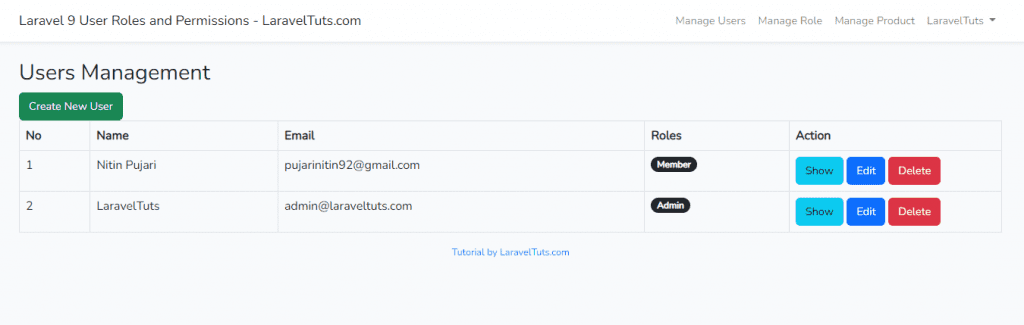
http://127.0.0.1:8000

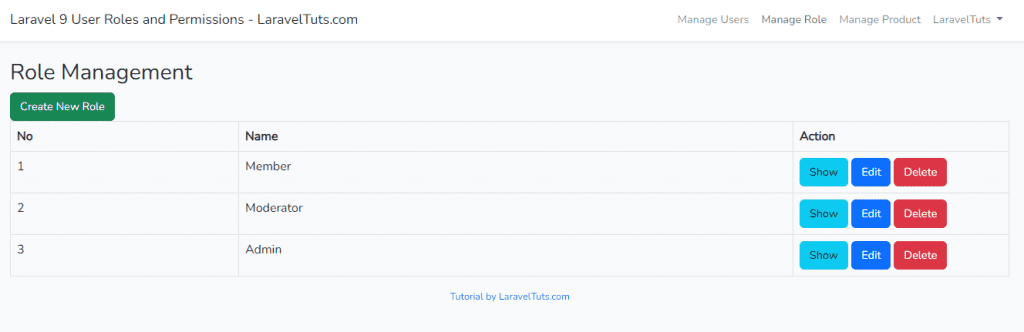
Now you can login with your admin details:

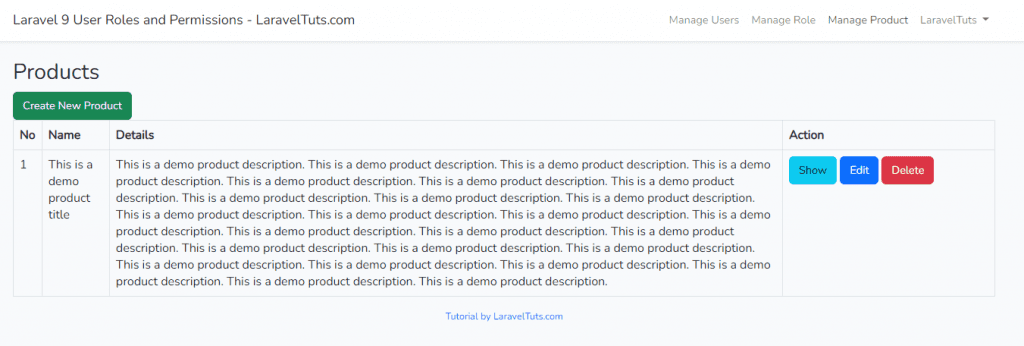
Email: admin@laraveltuts.com

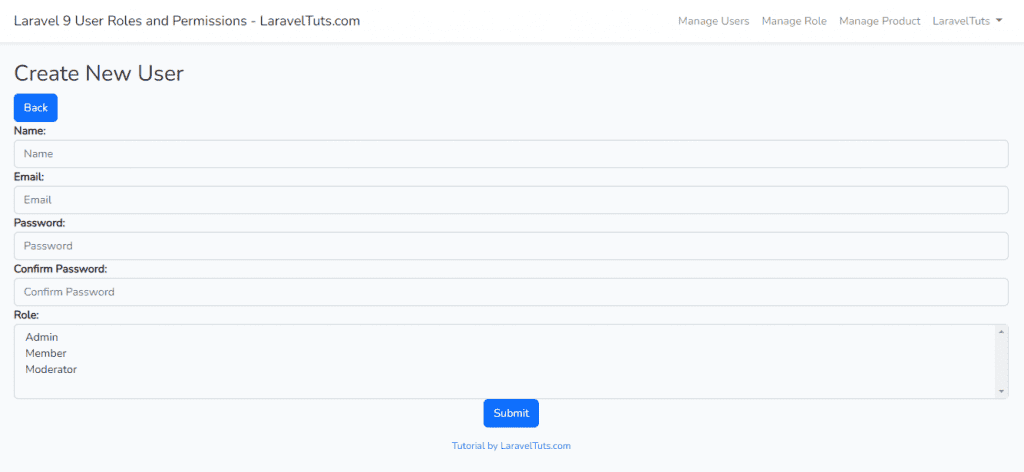
Password: password

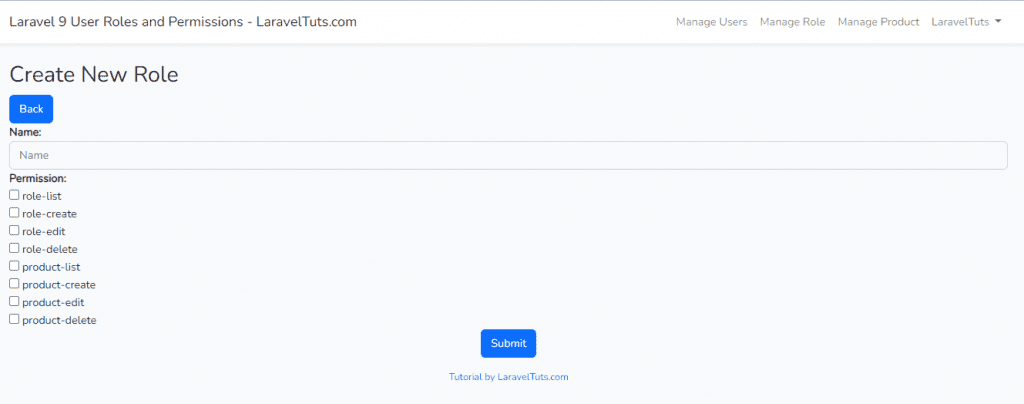
**Previews:**

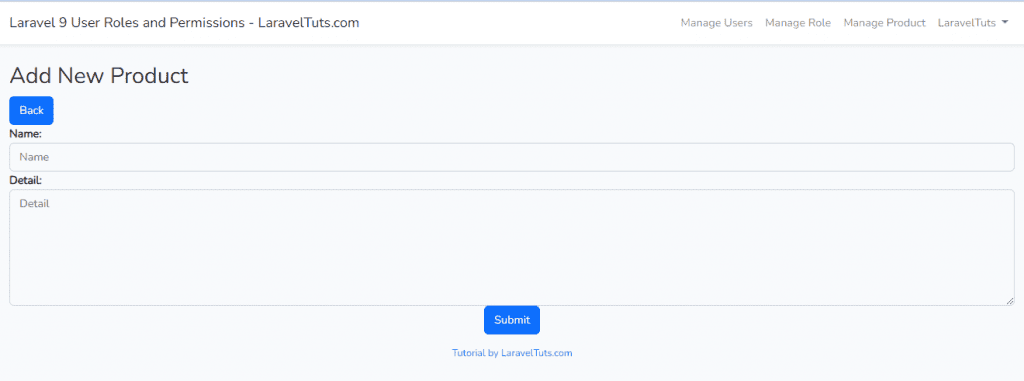
**User management**

**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/).