DB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

DB\_PORT=3306

DB\_DATABASE=laraposv1

DB\_USERNAME=root

DB\_PASSWORD=

php artisan make:controller Dashboard/WelcomeController

Controller created successfully.

WelcomeController.php

<?php

namespace App\Http\Controllers\Dashboard;

use App\Models\Category;

use App\Models\Client;

use App\Models\Order;

use App\Models\Product;

use App\Models\User;

use App\Http\Controllers\Controller;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\DB;

class WelcomeController extends Controller

{

public function index()

{

$categories\_count = Category::count();

$products\_count = Product::count();

$clients\_count = Client::count();

$users\_count = User::whereRoleIs('admin')->count();

$sales\_data = Order::select(

DB::raw('YEAR(created\_at) as year'),

DB::raw('MONTH(created\_at) as month'),

DB::raw('SUM(total\_price) as sum')

)->groupBy('month')->get();

return view('dashboard.welcome', compact('categories\_count', 'products\_count', 'clients\_count', 'users\_count', 'sales\_data'));

}//end of index

}

<https://github.com/Astrotomic/laravel-translatable>

composer require astrotomic/laravel-translatable

php artisan make:model Category -m

Model created successfully.

Created Migration: 2022\_09\_28\_152514\_create\_categories\_table

C:\xampp\htdocs\laraposv1>php artisan make:model CategoryTranslation -m

Model created successfully.

Created Migration: 2022\_09\_28\_152546\_create\_category\_translations\_table

C:\xampp\htdocs\laraposv1>php artisan make:model Client -m

Model created successfully.

Created Migration: 2022\_09\_28\_152603\_create\_clients\_table

C:\xampp\htdocs\laraposv1>php artisan make:model Order -m

Model created successfully.

Created Migration: 2022\_09\_28\_152615\_create\_orders\_table

C:\xampp\htdocs\laraposv1>php artisan make:model Product -m

Model created successfully.

Created Migration: 2022\_09\_28\_152627\_create\_products\_table

C:\xampp\htdocs\laraposv1>php artisan make:model ProductTranslation -m

Model created successfully.

Created Migration: 2022\_09\_28\_152639\_create\_product\_translations\_table

public function up()

{

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

$table->increments('id');

$table->string('first\_name');

$table->string('last\_name');

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

$table->string('image')->default('default.png');

$table->timestamp('email\_verified\_at')->nullable();

$table->string('password');

$table->rememberToken();

$table->timestamps();

});

}

public function up()

{

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

$table->increments('id');

$table->timestamps();

});

}

public function up()

{

Schema::create('category\_translations', function (Blueprint $table) {

$table->increments('id');

$table->integer('category\_id')->unsigned();

$table->string('name');

$table->string('locale')->index();

$table->unique(['category\_id', 'locale']);

$table->foreign('category\_id')->references('id')->on('categories')->onDelete('cascade');

});

}

public function up()

{

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

$table->increments('id');

$table->string('name');

$table->string('phone');

$table->text('address');

$table->timestamps();

});

}

public function up()

{

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

$table->increments('id');

$table->integer('category\_id')->unsigned();

$table->string('image')->default('default.png');

$table->double('purchase\_price', 8, 2);

$table->double('sale\_price', 8, 2);

$table->integer('stock');

$table->timestamps();

$table->foreign('category\_id')->references('id')->on('categories')->onDelete('cascade');

});

}

public function up()

{

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

$table->increments('id');

$table->integer('client\_id')->unsigned();

$table->double('total\_price', 8, 2)->nullable();

$table->timestamps();

$table->foreign('client\_id')->references('id')->on('clients')->onDelete('cascade');

});

}

public function up()

{

Schema::create('product\_order', function (Blueprint $table) {

$table->increments('id');

$table->integer('product\_id')->unsigned();

$table->integer('order\_id')->unsigned();

$table->integer('quantity')->default(1);

$table->foreign('product\_id')->references('id')->on('products')->onDelete('cascade');

$table->foreign('order\_id')->references('id')->on('orders')->onDelete('cascade');

});

}

<https://github.com/mcamara/laravel-localization>

composer require mcamara/laravel-localization

php artisan vendor:publish --provider="Mcamara\LaravelLocalization\LaravelLocalizationServiceProvider"

config

laravellocalization.php

'ar'

Config

app.php

'locale' => 'ar',

'fallback\_locale' => 'ar',

Kernel.php

'localize' => \Mcamara\LaravelLocalization\Middleware\LaravelLocalizationRoutes::class,

'localizationRedirect' => \Mcamara\LaravelLocalization\Middleware\LaravelLocalizationRedirectFilter::class,

'localeSessionRedirect' => \Mcamara\LaravelLocalization\Middleware\LocaleSessionRedirect::class,

'localeViewPath' => \Mcamara\LaravelLocalization\Middleware\LaravelLocalizationViewPath::class,

<https://laratrust.santigarcor.me/docs/7.x/installation.html>

composer require santigarcor/laratrust

php artisan vendor:publish --tag="laratrust"

php artisan config:clear

>php artisan laratrust:setup

Creating migration

Laratrust Migration Creation.

A migration that creates permission\_role, permission\_user, permissions, role\_user, roles tables will be created in database/migrations directory

Proceed with the migration creation? (yes/no) [yes]:

> yes

Creating migration

Migration created successfully.

Creating Role model

Role model created successfully.

Creating Permission model

Permission model created successfully.

Adding LaratrustUserTrait to User model

LaratrustUserTrait added successfully to App\Models\User

composer dump-autoload

php artisan migrate

Migrating: 2022\_09\_28\_141629\_laratrust\_setup\_tables

Migrated: 2022\_09\_28\_141629\_laratrust\_setup\_tables (8,994.36ms)

Config

laratrust\_seeder.php

<?php

return [

'role\_structure' => [

'super\_admin' => [

'categories' => 'c,r,u,d',

'products' => 'c,r,u,d',

'clients' => 'c,r,u,d',

'orders' => 'c,r,u,d',

'users' => 'c,r,u,d',

],

'admin' => []

],

'permissions\_map' => [

'c' => 'create',

'r' => 'read',

'u' => 'update',

'd' => 'delete'

]

];

php artisan make:seeder UsersTableSeeder

Seeder created successfully.

C:\xampp\htdocs\laraposv1>php artisan make:seeder CategoriesTableSeeder

Seeder created successfully.

C:\xampp\htdocs\laraposv1>php artisan make:seeder ClientsTableSeeder

Seeder created successfully.

C:\xampp\htdocs\laraposv1>php artisan make:seeder ProductsTableSeeder

Seeder created successfully.

<?php

namespace Database\Seeders;

use Illuminate\Database\Seeder;

class CategoriesTableSeeder extends Seeder

{

/\*\*

\* Run the database seeds.

\*

\* @return void

\*/

public function run()

{

$categories = ['cat one', 'cat two', 'cat three'];

foreach ($categories as $category) {

\App\Category::create([

'ar' => ['name' => $category],

'en' => ['name' => $category],

]);

}//end of foreach

}

}

<?php

namespace Database\Seeders;

use Illuminate\Database\Seeder;

class ProductsTableSeeder extends Seeder

{

/\*\*

\* Run the database seeds.

\*

\* @return void

\*/

public function run()

{

$products = ['pro one', 'pro two'];

foreach ($products as $product) {

\App\Product::create([

'category\_id' => 1,

'ar' => ['name' => $product, 'description' => $product . ' desc'],

'en' => ['name' => $product, 'description' => $product . ' desc'],

'purchase\_price' => 100,

'sale\_price' => 150,

'stock' => 100,

]);

}//end of foreach

}

}

<?php

namespace Database\Seeders;

use Illuminate\Database\Seeder;

class ClientsTableSeeder extends Seeder

{

/\*\*

\* Run the database seeds.

\*

\* @return void

\*/

public function run()

{

$clients = ['ahmed', 'mohamed'];

foreach ($clients as $client) {

\App\Client::create([

'name' => $client,

'phone' => '011111112',

'address' => 'haram',

]);

}//end of foreach

}

}

class Category extends Model

{

use HasFactory;

use \Dimsav\Translatable\Translatable;

protected $guarded = [];

public $translatedAttributes = ['name'];

public function products()

{

return $this->hasMany(Product::class);

}//end of products

}

class CategoryTranslation extends Model

{

use HasFactory;

public $timestamps = false;

protected $fillable = ['name'];

}

class Client extends Model

{

use HasFactory;

protected $guarded = [];

protected $casts = [

'phone' => 'array'

];

public function getNameAttribute($value)

{

return ucfirst($value);

}//end of get name attribute

public function orders()

{

return $this->hasMany(Order::class);

}//end of orders

}

class Order extends Model

{

use HasFactory;

protected $guarded = [];

public function client()

{

return $this->belongsTo(Client::class);

}//end of user

public function products()

{

return $this->belongsToMany(Product::class, 'product\_order')->withPivot('quantity');

}//end of products

}

class Product extends Model

{

use HasFactory;

use \Dimsav\Translatable\Translatable;

protected $guarded = ['id'];

public $translatedAttributes = ['name', 'description'];

protected $appends = ['image\_path', 'profit\_percent'];

public function getImagePathAttribute()

{

return asset('uploads/product\_images/' . $this->image);

}//end of image path attribute

public function getProfitPercentAttribute()

{

$profit = $this->sale\_price - $this->purchase\_price;

$profit\_percent = $profit \* 100 / $this->purchase\_price;

return number\_format($profit\_percent, 2);

}//end of get profit attribute

public function category()

{

return $this->belongsTo(Category::class);

}//end fo category

public function orders()

{

return $this->belongsToMany(Order::class, 'product\_order');

}//end of orders

}

class ProductTranslation extends Model

{

use HasFactory;

public $timestamps = false;

protected $fillable = ['name', 'description'];

}

php artisan vendor:publish --tag=translatable

composer require dimsav/laravel-translatable

config

translatable.php

'locales' => [

'ar',

'en',

],

<?php

namespace Database\Seeders;

use Illuminate\Database\Seeder;

class UsersTableSeeder extends Seeder

{

/\*\*

\* Run the database seeds.

\*

\* @return void

\*/

public function run()

{

$user = \App\User::create([

'first\_name' => 'super',

'last\_name' => 'admin',

'email' => 'super\_admin@app.com',

'password' => bcrypt('123456'),

]);

$user->attachRole('super\_admin');

}

}

<https://image.intervention.io/v2/introduction/installation>

composer require intervention/image

php artisan vendor:publish --provider="Intervention\Image\ImageServiceProviderLaravelRecent"

Database

Seeders

LaratrustSeeder.php

php artisan make:controller Dashboard/CategoryController

Controller created successfully.

C:\xampp\htdocs\laraposv1>php artisan make:controller Dashboard/ClientController

Controller created successfully.

C:\xampp\htdocs\laraposv1>php artisan make:controller Dashboard/OrderController

Controller created successfully.

C:\xampp\htdocs\laraposv1>php artisan make:controller Dashboard/ProductController

Controller created successfully.

C:\xampp\htdocs\laraposv1>php artisan make:controller Dashboard/UserController

Controller created successfully.