routes/web.php

<?php

use App\CountryVisit;

use Illuminate\Support\Facades\Auth;

use Illuminate\Support\Facades\Route;

use TCG\Voyager\Facades\Voyager;

Route::get('logs', '\Rap2hpoutre\LaravelLogViewer\LogViewerController@index')->middeware('admin');

// Shop and welcome

Route::get('/', 'WelcomePageController@index')->name('welcome');

Route::get('/shop', 'ShopController@index')->name('shop.index');

Route::get('/shop/{product}', 'ShopController@show')->name('shop.show');

Route::get('/shop/search/{query}', 'ShopController@search')->name('shop.search');

// Cart

Route::get('/cart', 'CartController@index')->name('cart.index');

Route::post('/cart', 'CartController@store')->name('cart.store');

Route::delete('/cart/{product}/{cart}', 'CartController@destroy')->name('cart.destroy');

Route::post('/cart/save-later/{product}', 'CartController@saveLater')->name('cart.save-later');

Route::post('/cart/add-to-cart/{product}', 'CartController@addToCart')->name('cart.add-to-cart');

Route::patch('/cart/{product}', 'CartController@update')->name('cart.update');

// checkout

Route::get('/checkout', 'CheckoutController@index')->name('checkout.index');

Route::post('/checkout', 'CheckoutController@store')->name('checkout.store');

Route::get('/guest-checkout', 'CheckoutController@index')->name('checkout.guest');

// coupon

Route::post('/coupon', 'CouponsController@store')->name('coupon.store');

Route::delete('/coupon/', 'CouponsController@destroy')->name('coupon.destroy');

// auth routes

Auth::routes();

Route::get('/login/{provider}', 'Auth\LoginController@redirectToProvider');

Route::get('/login/{provider}/callback', 'Auth\LoginController@handleProviderCallback');

Route::get('/home', 'HomeController@index')->name('home');

Route::group(['prefix' => 'admin'], function () {

Voyager::routes();

Route::get('/country\_visits', 'VisitsController@index')->name('voyager.visits');

});

database/migrations/2014\_10\_12\_000000\_create\_users\_table.php

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class CreateUsersTable extends Migration

{

/\*\*

\* Run the migrations.

\*

\* @return void

\*/

public function up()

{

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

$table->bigIncrements('id');

$table->string('name');

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

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

$table->string('github\_id')->nullable();

$table->string('facebook\_id')->nullable();

$table->string('google\_id')->nullable();

$table->string('password')->nullable();

$table->rememberToken();

$table->timestamps();

});

}

/\*\*

\* Reverse the migrations.

\*

\* @return void

\*/

public function down()

{

Schema::dropIfExists('users');

}

}

database/migrations/2014\_10\_12\_100000\_create\_password\_resets\_table.php

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class CreatePasswordResetsTable extends Migration

{

/\*\*

\* Run the migrations.

\*

\* @return void

\*/

public function up()

{

Schema::create('password\_resets', function (Blueprint $table) {

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

$table->string('token');

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

});

}

/\*\*

\* Reverse the migrations.

\*

\* @return void

\*/

public function down()

{

Schema::dropIfExists('password\_resets');

}

}

database/migrations/2019\_08\_19\_000000\_create\_failed\_jobs\_table.php

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class CreateFailedJobsTable extends Migration

{

/\*\*

\* Run the migrations.

\*

\* @return void

\*/

public function up()

{

Schema::create('failed\_jobs', function (Blueprint $table) {

$table->bigIncrements('id');

$table->text('connection');

$table->text('queue');

$table->longText('payload');

$table->longText('exception');

$table->timestamp('failed\_at')->useCurrent();

});

}

/\*\*

\* Reverse the migrations.

\*

\* @return void

\*/

public function down()

{

Schema::dropIfExists('failed\_jobs');

}

}

database/migrations/2019\_10\_16\_185424\_create\_products\_table.php

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class CreateProductsTable extends Migration

{

/\*\*

\* Run the migrations.

\*

\* @return void

\*/

public function up()

{

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

$table->bigIncrements('id');

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

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

$table->string('details')->nullable();

$table->integer('price');

$table->text('description');

$table->timestamps();

});

}

/\*\*

\* Reverse the migrations.

\*

\* @return void

\*/

public function down()

{

Schema::dropIfExists('products');

}

}

Database/migrations/2019\_10\_17\_195112\_create\_categories\_table.php

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class CreateCategoriesTable extends Migration

{

/\*\*

\* Run the migrations.

\*

\* @return void

\*/

public function up()

{

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

$table->bigIncrements('id');

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

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

$table->timestamps();

});

}

/\*\*

\* Reverse the migrations.

\*

\* @return void

\*/

public function down()

{

Schema::dropIfExists('category');

}

}

Database/migrations/2019\_10\_17\_195227\_add\_category\_id\_to\_products\_table.php

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class AddCategoryIdToProductsTable extends Migration

{

/\*\*

\* Run the migrations.

\*

\* @return void

\*/

public function up()

{

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

$table->string('image')->after('price');

$table->boolean('featured')->default(false)->after('image');

$table->unsignedBigInteger('category\_id')->after('featured');

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

});

}

/\*\*

\* Reverse the migrations.

\*

\* @return void

\*/

public function down()

{

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

$table->dropForeign('products\_category\_id\_foreign');

$table->dropColumn('image');

$table->dropColumn('category\_id');

});

}

}

database/migrations/2019\_10\_18\_002421\_create\_coupons\_table.php

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class CreateCouponsTable extends Migration

{

/\*\*

\* Run the migrations.

\*

\* @return void

\*/

public function up()

{

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

$table->bigIncrements('id');

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

$table->string('type');

$table->integer('value')->nullable();

$table->integer('percent\_off')->nullable();

$table->timestamps();

});

}

/\*\*

\* Reverse the migrations.

\*

\* @return void

\*/

public function down()

{

Schema::dropIfExists('coupons');

}

}

database/migrations/2019\_10\_19\_145706\_add\_images\_column\_to\_products.php

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class AddImagesColumnToProducts extends Migration

{

/\*\*

\* Run the migrations.

\*

\* @return void

\*/

public function up()

{

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

$table->text('images')->nullable()->after('image');

});

}

/\*\*

\* Reverse the migrations.

\*

\* @return void

\*/

public function down()

{

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

$table->dropColumn('images');

});

}

}

database/migrations/2019\_10\_20\_114921\_create\_tags\_table.php

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class CreateTagsTable extends Migration

{

/\*\*

\* Run the migrations.

\*

\* @return void

\*/

public function up()

{

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

$table->bigIncrements('id');

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

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

$table->timestamps();

});

}

/\*\*

\* Reverse the migrations.

\*

\* @return void

\*/

public function down()

{

Schema::dropIfExists('tags');

}

}

database/migrations/2019\_10\_20\_115306\_create\_products\_tags\_table.php

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class CreateProductsTagsTable extends Migration

{

/\*\*

\* Run the migrations.

\*

\* @return void

\*/

public function up()

{

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

$table->bigIncrements('id');

$table->unsignedBigInteger('product\_id');

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

$table->unsignedBigInteger('tag\_id');

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

$table->timestamps();

});

}

/\*\*

\* Reverse the migrations.

\*

\* @return void

\*/

public function down()

{

Schema::dropIfExists('product\_tag');

}

}

database/migrations/2019\_10\_22\_185425\_create\_orders\_table.php

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class CreateOrdersTable extends Migration

{

/\*\*

\* Run the migrations.

\*

\* @return void

\*/

public function up()

{

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

$table->bigIncrements('id');

$table->unsignedBigInteger('user\_id')->nullable();

$table->foreign('user\_id')->references('id')->on('users')

->onUpdate('cascade')->onDelete('set null');

$table->string('billing\_email');

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

$table->string('billing\_address');

$table->string('billing\_city');

$table->string('billing\_province');

$table->string('billing\_postalcode');

$table->string('billing\_phone');

$table->string('billing\_name\_on\_card');

$table->integer('billing\_discount')->default(0);

$table->string('billing\_discount\_code')->nullable();

$table->integer('billing\_subtotal');

$table->integer('billing\_tax');

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

$table->string('payment\_gateway')->default('stripe');

$table->boolean('shipped')->default(false);

$table->string('error')->nullable();

$table->timestamps();

});

}

/\*\*

\* Reverse the migrations.

\*

\* @return void

\*/

public function down()

{

Schema::dropIfExists('orders');

}

}

database/migrations/2019\_10\_22\_190613\_create\_order\_product\_table.php

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class CreateOrderProductTable extends Migration

{

/\*\*

\* Run the migrations.

\*

\* @return void

\*/

public function up()

{

Schema::create('order\_product', function (Blueprint $table) {

$table->bigIncrements('id');

$table->unsignedBigInteger('product\_id')->nullable();

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

->onUpdate('cascade')->onDelete('set null');

$table->unsignedBigInteger('order\_id')->nullable();

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

->onUpdate('cascade')->onDelete('set null');

$table->unsignedInteger('quantity');

$table->timestamps();

});

}

/\*\*

\* Reverse the migrations.

\*

\* @return void

\*/

public function down()

{

Schema::dropIfExists('order\_product');

}

}

database/migrations/2019\_11\_09\_002942\_add\_quantity\_to\_products\_table.php

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class AddQuantityToProductsTable extends Migration

{

/\*\*

\* Run the migrations.

\*

\* @return void

\*/

public function up()

{

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

$table->unsignedInteger('quantity')->after('featured')->default(10);

});

}

/\*\*

\* Reverse the migrations.

\*

\* @return void

\*/

public function down()

{

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

$table->dropColumn('quantity');

});

}

}

database/migrations/2020\_06\_27\_012504\_create\_visits\_table.php

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class CreateVisitsTable extends Migration

{

/\*\*

\* Run the migrations.

\*

\* @return void

\*/

public function up()

{

Schema::create('country\_visits', function (Blueprint $table) {

$table->bigIncrements('id');

$table->string('country');

$table->unsignedBigInteger('visits')->default(0);

$table->timestamps();

});

}

/\*\*

\* Reverse the migrations.

\*

\* @return void

\*/

public function down()

{

Schema::dropIfExists('visits');

}

}

app/Http/Controllers/CartController.php

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Product;

use Cart;

class CartController extends Controller

{

public function index() {

$mightLike = Product::mightAlsoLike()->get();

return view('cart')->with('mightLike', $mightLike);

}

public function store() {

session()->forget('coupon');

$duplicates = Cart::instance('default')->search(function($cartItem, $rowId) {

return $cartItem->id == request()->id;

});

$duplicatesLater = Cart::instance('saveForLater')->search(function($cartItem, $rowId) {

return $cartItem->id == request()->id;

});

if($duplicates->isNotEmpty()) {

session()->flash('success', 'Item is already in your cart');

return redirect()->route('cart.index');

} else if($duplicatesLater->isNotEmpty()) {

Cart::instance('saveForLater')->remove($duplicatesLater->first()->rowId);

}

Cart::instance('default')->add(request()->id, request()->name, 1, request()->price)->associate('App\Product');

session()->flash('success', 'product added to the cart');

return redirect()->route('cart.index');

}

public function update($id) {

session()->forget('coupon');

// dd(request()->all());

if(request()->productQuantity >= request()->quantity) {

Cart::instance('default')->update($id, request()->quantity);

session()->flash('success', 'quantity updated successfully!');

return response()->json(['success' => ''], 200);

}

session()->flash('error', 'not enough products');

return response()->json(['error' => ''], 405);

}

public function destroy($id, $cart) {

if($cart == 'default')

Cart::instance('default')->remove($id);

else if($cart = 'saveForLater')

Cart::instance('saveForLater')->remove($id);

session()->flash('success', 'item has been removed');

return back();

}

public function saveLater($id) {

session()->forget('coupon');

$item = Cart::get($id);

Cart::remove($id);

$duplicates = Cart::instance('saveForLater')->search(function($cartItem, $rowId) use ($id) {

return $rowId == $id;

});

if($duplicates->isNotEmpty()) {

session()->flash('success', 'Item is already saved for later');

return redirect()->route('cart.index');

}

Cart::instance('saveForLater')->add($item->id, $item->name, 1, $item->price)->associate('App\Product');

session()->flash('success', 'Item has been saved for later');

return redirect()->route('cart.index');

}

public function addToCart($id) {

session()->forget('coupon');

$item = Cart::instance('saveForLater')->get($id);

Cart::instance('saveForLater')->remove($id);

$exist = Cart::instance('default')->search(function($cartItem, $rowId) use($item) {

return $cartItem->id == $item->id;

});

if($exist->isNotEmpty()) {

session()->flash('success', 'Item is already in the cart');

return redirect()->route('cart.index');

}

Cart::instance('default')->add($item->id, $item->name, 1, $item->price)->associate('App\Product');

session()->flash('success', 'item added to the cart');

return redirect()->route('cart.index');

}

}

app/Http/Controllers/CheckoutController.php

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Http\Requests\CheckoutRequest;

use Cart;

use App\Order;

use App\OrderProduct;

use Cartalyst\Stripe\Stripe;

use Mail;

use App\Mail\OrderPlaced;

use App\Product;

class CheckoutController extends Controller

{

public function index()

{

if (Cart::instance('default')->count() > 0) {

$subtotal = Cart::instance('default')->subtotal() ?? 0;

$discount = session('coupon')['discount'] ?? 0;

$newSubtotal = $subtotal - $discount > 0 ? $subtotal - $discount : 0;

$tax = $newSubtotal \* (config('cart.tax') / 100);

$total = $tax + $newSubtotal;

return view('checkout')->with([

'subtotal' => $subtotal,

'discount' => $discount,

'newSubtotal' => $newSubtotal,

'total' => $total,

'tax' => $tax

]);

}

return redirect()->route('cart.index')->withError('You have nothing in your cart , please add some products first');

}

public function store(CheckoutRequest $request)

{

if ($this->productsAreNoLongerAvailable()) {

return back()->withError('Sorry, one of the items on your cart is no longer available');

}

$contents = Cart::instance('default')->content()->map(function ($item) {

return $item->model->slug . ', ' . $item->qty;

})->values()->toJson();

try {

// $stripe = Stripe::make('api\_key');

// $charge = $stripe->charges()->create([

// 'amount' => Cart::instance('default')->total(),

// 'currency' => 'USD',

// 'source' => $request->stripeToken,

// 'description' => 'Order',

// 'receipt\_email' => $request->email,

// 'metadata' => [

// 'contents' => $contents,

// 'quantity' => Cart::instance('default')->count(),

// 'discount' => session()->has('coupon') ? collect(session('coupon')->toJson) : null,

// ],

// ]);

$order = $this->insertIntoOrdersTable($request, null);

// SUCCESSFUL

$this->decreaseQuantities();

Mail::to('me@me.com')->send(new OrderPlaced($order));

Cart::instance('default')->destroy();

session()->forget('coupon');

return redirect()->route('welcome')->with('success', 'Your order is completed successfully!');

} catch (Exception $e) {

$this->insertIntoOrdersTable($request, $e->getMessage());

return back()->withError('Error ' . $e->getMessage());

}

}

private function getNumbers()

{

$tax = config('cart.tax') / 100;

$discount = session()->get('coupon')['discount'] ?? 0;

$code = session()->get('coupon')['code'] ?? null;

$newSubtotal = Cart::instance('default')->subtotal() - $discount;

if ($newSubtotal < 0) {

$newSubtotal = 0;

}

$newTax = $newSubtotal \* $tax;

$newTotal = $newSubtotal + $newTax;

return collect([

'tax' => $tax,

'discount' => $discount,

'code' => $code,

'newSubtotal' => $newSubtotal,

'newTax' => $newTax,

'newTotal' => $newTotal

]);

}

private function insertIntoOrdersTable($request, $error)

{

$order = Order::create([

'user\_id' => auth()->user() ? auth()->user()->id : null,

'billing\_email' => $request->email,

'billing\_name' => $request->name,

'billing\_address' => $request->address,

'billing\_city' => $request->city,

'billing\_province' => $request->province,

'billing\_postalcode' => $request->postal\_code,

'billing\_phone' => $request->phone,

'billing\_name\_on\_card' => $request->name\_on\_card,

'billing\_discount' => $this->getNumbers()->get('discount'),

'billing\_discount\_code' => $this->getNumbers()->get('code'),

'billing\_subtotal' => $this->getNumbers()->get('newSubtotal'),

'billing\_tax' => $this->getNumbers()->get('newTax'),

'billing\_total' => $this->getNumbers()->get('newTotal'),

'error' => $error

]);

foreach (Cart::instance('default')->content() as $item) {

OrderProduct::create([

'product\_id' => $item->model->id,

'order\_id' => $order->id,

'quantity' => $item->qty

]);

}

return $order;

}

private function decreaseQuantities()

{

foreach (Cart::instance('default')->content() as $item) {

$product = Product::find($item->model->id);

$product->update(['quantity' => $product->quantity - $item->qty]);

}

}

private function productsAreNoLongerAvailable()

{

foreach (Cart::instance('default')->content() as $item) {

$product = Product::find($item->model->id);

if ($product->quantity < $item->qty) {

return true;

}

}

return false;

}

}

app/Http/Controllers/CouponsController.php

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Coupon;

use Cart;

class CouponsController extends Controller

{

public function store(Request $request)

{

$this->validate($request, [

'coupon\_code' => 'required'

]);

$coupon = Coupon::where('code', $request->coupon\_code)->first();

if(!$coupon) return redirect()->route('checkout.index')->with('error', 'Invalid Coupon Code');

session()->put('coupon', [

'code' => $coupon->code,

'discount' => $coupon->discount(Cart::instance('default')->subtotal())

]);

return redirect()->route('checkout.index')->withSuccess('Coupon applied successfully!');

}

public function destroy()

{

session()->forget('coupon');

return redirect()->route('checkout.index')->with('success', 'Coupon has been removed');

}

}

app/Http/Controllers/ShopController.php

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Product;

use App\Category;

use App\Tag;

class ShopController extends Controller

{

public function index() {

$pagination = 12;

if(request()->category) {

$category = Category::where('slug', request()->category)->get()->first();

$products = Product::where('category\_id', $category->id);

$categoryName = $category->name;

} else if(request()->tag) {

$tag = Tag::where('slug', request()->tag)->get()->first();

$products = $tag->products();

$tagName = $tag->name;

} else {

$products = Product::where('featured', true);

$categoryName = 'Featured';

}

if(request()->sort == 'low\_high') {

$products = $products->orderBy('price')->paginate($pagination);

} else if(request()->sort == 'high\_low') {

$products = $products->orderBy('price', 'desc')->paginate($pagination);

} else {

$products = $products->inRandomOrder()->paginate($pagination);

}

$categories = Category::all();

$tags = Tag::all();

return view('shop')->with([

'products' => $products,

'categories'=> $categories,

'tags'=> $tags,

'categoryName' => $categoryName ?? null,

'tagName' => $tagName ?? null

]);

}

public function show($slug) {

$product = Product::where('slug', $slug)->firstOrFail();

$images = json\_decode($product->images);

$mightLike = Product::where('slug', '!=', $product->slug)->mightAlsoLike()->get();

$stockLevel = getStockLevel($product->quantity);

return view('product')->with([

'product' => $product,

'mightLike' => $mightLike,

'images' => $images,

'stockLevel' => $stockLevel

]);

}

public function search($query) {

if(strlen($query) < 3) return back()->with('error', 'minimum query length is 3');

$products = Product::search($query)->paginate(10);

return view('search')->with(['products' => $products, 'query' => $query]);

}

}

app/Http/Controllers/VisitsController.php

<?php

namespace App\Http\Controllers;

use App\CountryVisit;

use Illuminate\Http\Request;

class VisitsController extends Controller

{

public function index()

{

$topCountries = CountryVisit::orderBy('visits', 'desc')->limit(10)

->select('country', 'visits')->get();

$CountryVisit = CountryVisit::orderBy('visits', 'desc')->paginate(10);

return view('admin.visits')->with([

'CountryVisit' => $CountryVisit,

'topCountries' => $topCountries,

]);

}

}

app/Http/Controllers/WelcomePageController.php

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Product;

class WelcomePageController extends Controller

{

public function index()

{

$products = Product::inRandomOrder()->take(6)->get();

$hotProducts = Product::inRandomOrder()->take(3)->get();

return view('welcome')->with([

'products'=> $products,

'hotProducts' => $hotProducts

]);

}

}

app/Http/Controllers/Voyager/OrdersController.php

<?php

namespace App\Http\Controllers\Voyager;

use TCG\Voyager\Http\Controllers\VoyagerBaseController;

use App\Order;

use Validator;

use App\Product;

use Illuminate\Http\Request;

use TCG\Voyager\Facades\Voyager;

use TCG\Voyager\Events\BreadDataAdded;

use TCG\Voyager\Events\BreadDataUpdated;

class OrdersController extends VoyagerBaseController {

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// \_\_\_\_\_

// | \_\_ \

// | |\_\_) |

// | \_ /

// | | \ \

// |\_| \\_\

//

// Read an item of our Data Type B(R)EAD

//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

public function show(Request $request, $id)

{

$slug = $this->getSlug($request);

$dataType = Voyager::model('DataType')->where('slug', '=', $slug)->first();

$isSoftDeleted = false;

if (strlen($dataType->model\_name) != 0) {

$model = app($dataType->model\_name);

// Use withTrashed() if model uses SoftDeletes and if toggle is selected

if ($model && in\_array(SoftDeletes::class, class\_uses($model))) {

$model = $model->withTrashed();

}

if ($dataType->scope && $dataType->scope != '' && method\_exists($model, 'scope'.ucfirst($dataType->scope))) {

$model = $model->{$dataType->scope}();

}

$dataTypeContent = call\_user\_func([$model, 'findOrFail'], $id);

if ($dataTypeContent->deleted\_at) {

$isSoftDeleted = true;

}

} else {

// If Model doest exist, get data from table name

$dataTypeContent = DB::table($dataType->name)->where('id', $id)->first();

}

// Replace relationships' keys for labels and create READ links if a slug is provided.

$dataTypeContent = $this->resolveRelations($dataTypeContent, $dataType, true);

// If a column has a relationship associated with it, we do not want to show that field

$this->removeRelationshipField($dataType, 'read');

// Check permission

$this->authorize('read', $dataTypeContent);

// Check if BREAD is Translatable

$isModelTranslatable = is\_bread\_translatable($dataTypeContent);

$view = 'voyager::bread.read';

if (view()->exists("voyager::$slug.read")) {

$view = "voyager::$slug.read";

}

$products = Order::find($id)->products;

return Voyager::view($view, compact('dataType', 'dataTypeContent', 'isModelTranslatable', 'isSoftDeleted', 'products'));

}

}