Вебинар 2

Создаем наши первые контроллеры, которые будут выполнять (CRUD) наших товаров

Но для начала надо создать модель и миграцию для создания нашей таблицы products

В терминале пишем

* Php artisan make:model Product -m

Переходим в нашу миграцию /database/migrations/create\_products\_table

Добавляем необходимые поля

{  
 Schema::create('products', function (Blueprint $table) {  
 $table->id();

$table->string('is\_published')->default('0');  
 $table->string('photo\_url');  
 $table->string('title');  
 $table->string('cost');  
 $table->string('country');  
 $table->string('release\_year');  
 $table->string('model');  
 $table->timestamps();  
 });  
}

В терминале пишем

* Php artisan migrate

После переходим в нашу модель app/models/Product

Добавляем переменную fillable

protected $fillable = [  
 'count',  
 'is\_published',  
 'photo\_url',  
 'title',  
 'cost',  
 'country',  
 'release\_year',  
 'model',  
];

После создаем контроллер с главной страницей товаров для Администратора и создания товара

В терминале пишем

* Php artisan make:controller Admin/Product/IndexController –invokable
* Php artisan make:controller Admin/Product/CreateController --invokable

Удаляем $request Request в параметрах метода \_\_invoke

И возвращаем наше представление в нем

return view('admin.product.index');

Добавим наш новый роут и создадим для него группу

Route::group(['namespace' => 'App\Http\Controllers\Admin', 'prefix' => 'admin', 'middleware' => 'admin'], function () {  
 Route::group(['namespace' => 'Product', 'prefix' => 'products'], function () {  
 Route::get('/', 'IndexController')->name('admin.product.index');

Route::get('/create', 'CreateController')->name('admin.product.create);  
 });  
});

Создадим наше представление во views/admin/product/index.blade.php

@extends('layouts.app')  
  
@section('content')  
 <div class="mt-2 ms-5">  
 <a href="{{route('admin.product.create')}}" class="btn btn-primary">  
 Добавить товар  
 </a>  
 </div>

И еще одно представление на создание товара во views/admin/product/create.blade.php

@extends('layouts.app')  
  
@section('content')  
 <div class="w-25 p-5">  
 <form action="{{ route('admin.product.store') }}" method="post" enctype="multipart/form-data">  
 @csrf  
 <div class="input-group mb-3">  
 <span class="input-group-text" id="basic-addon1">Наименование</span>  
 <input type="text" name="title" class="form-control">  
 </div>  
 @error('title')  
 <div class="text-danger">{{ $message }}</div>  
 @enderror  
 <div class="input-group mb-3">  
 <label class="input-group-text" for="inputGroupFile01">Фото</label>  
 <input type="file" name="photo\_url" class="form-control" id="inputGroupFile01">  
 </div>  
 @error('photo\_url')  
 <div class="text-danger">{{ $message }}</div>  
 @enderror  
 <div class="input-group mb-3">  
 <span class="input-group-text">₽</span>  
 <input type="number" name="cost" class="form-control" aria-label="Цена в рублях">  
 <span class="input-group-text">.00</span>  
 </div>  
 @error('cost')  
 <div class="text-danger">{{ $message }}</div>  
 @enderror  
 <div class="input-group mb-3">  
 <span class="input-group-text" id="basic-addon2">Страна</span>  
 <input type="text" name="country" class="form-control">  
 </div>  
 @error('country')  
 <div class="text-danger">{{ $message }}</div>  
 @enderror  
  
 <div class="input-group mb-3">  
 <span class="input-group-text">Год выпуска</span>  
 <input type="text" name="release\_year" class="form-control">  
 </div>  
 @error('release\_year')  
 <div class="text-danger">{{ $message }}</div>  
 @enderror  
  
 <div class="input-group mb-3">  
 <span class="input-group-text">Модель</span>  
 <input type="text" name="model" class="form-control">  
 </div>  
 @error('model')  
 <div class="text-danger">{{ $message }}</div>  
 @enderror  
 <select name="category\_id" class="form-select" aria-label="Default select example">  
 <option selected>Выберите категорию</option>  
 @foreach($categories as $category)  
 <option value="{{ $category->id }}">{{ $category->title }}</option>  
 @endforeach  
 </select>  
 @error('category\_id')  
 <div class="text-danger">{{ $message }}</div>  
 @enderror  
 <button type="submit" class="mt-3 btn btn-primary">Добавить</button>  
 </form>  
 </div>  
@endsection

Создадим Request для валидации наших данных и контроллер для добавления товара в бд

Пишем в терминале

* Php artisan make:request Admin/ProductRequest
* Php arisan make:controller Admin/Product/StoreController –invokable

Изменим наш созданный Request

App/http/requests/admin/ProductRequest

public function authorize(): bool  
{  
 return true;  
}  
  
*/\*\*  
 \* Get the validation rules that apply to the request.  
 \*  
 \** ***@return*** *array<string, \Illuminate\Contracts\Validation\Rule|array|string>  
 \*/*public function rules(): array  
{  
 return [  
 'category\_id' => 'nullable',  
 'count' => 'nullable',  
 'is\_published' => 'nullable',  
 'photo\_url' => 'required|file|mimes:jpg,jpeg,png',  
 'title' => 'required',  
 'cost' => 'required',  
 'country' => 'required',  
 'release\_year' => 'required',  
 'model' => 'required',  
 ];  
}

Добавим роут на storecontroller

Route::group(['namespace' => 'App\Http\Controllers\Admin', 'prefix' => 'admin', 'middleware' => 'admin'], function () {  
 Route::group(['namespace' => 'Product', 'prefix' => 'products'], function () {  
 Route::get('/', 'IndexController')->name('admin.product.index');

Route::get('/create', 'CreateController')->name('admin.product.create);

Route::post('/', 'StoreController')->name('admin.product.store');

});  
  
});

Изменим наш app/http/controllers/admin/product/StoreController

public function \_\_invoke(ProductRequest $request)  
{  
 $photo\_url = Storage::putFile('public/photos', $request->photo\_url);  
 $photo\_url = substr($photo\_url, 7);  
 Product::*create*([  
 'category\_id' => $request->category\_id,  
 'photo\_url' => $photo\_url,  
 'title' => $request->title,  
 'cost' => $request->cost,  
 'country' => $request->country,  
 'release\_year' => $request->release\_year,  
 'model' => $request->model,  
 ]);  
 return redirect()->route('admin.product.index');  
}

Создадим линк нашего storage в папку public

В терминале пишем

* Php artisan storage:link

Создадим новый класс middleware для проверки на администратора

В терминале пишем

* Php artisan make:middleware AdminMiddleware

Переходим в app/http/kernel.php и добавляем наш middleware в переменную middlewareAliases

'admin' => AdminMiddleware::class

Импортируя его в этот файл (сверху файла kernel)

use App\Http\Middleware\AdminMiddleware;

И изменяем наш метод handle

if(!auth()->user()){  
 return redirect()->route('main');  
}  
if(auth()->user()->role !== 'admin'){  
 return redirect()->route('main');  
}  
return $next($request);

Мы проработали одни из CRUD операций, а именно CREATE и часть READ (создали главную страницу с продуктами), а также узнали как создавать свои классы middleware и request