1.**Удалить** таблицы **users** и **products** из БД **laravel-shop** + **запись о них** в таблице **migrations**.

2.**Добавить** в **миграцию** таблицы **users** поле **role**:

**$table->enum('role', array('user', 'redac', 'admin'))->default('user');**

3.В **DatabaseSeeder.php** добавить:

...

**use App\Models\User;**

public function run()

{

**// Users**

**User::create(**

**[**

**'name' => 'admin',**

**'email' => 'admin@gmail.com',**

**'password' => bcrypt('12345678'),**

**'role' => 'admin',**

**'remember\_token' => str\_random(10),**

**]**

**);**

...

//Products – **ЭТО ЕСТЬ И ОСТАЕТСЯ!!!**

Product::create(

[

'name' => 'Modern Chair',

'price' => '180$',

'image' => 'bg-img/1.jpg',

'top9' => 1,

]

);

…

т.е. данные по **users** (из **DatabaseSeeder.php** проекта **laravel-cards**)

+**php artisan migrate --seed**

2.**(ЭТО УЖЕ ДОЛЖНО БЫЛО БЫТЬ СДЕЛАНО РАНЕЕ )** Откорректировать **views\front\layout.blade.php** таким образом, чтобы в **header** появились пункты **Login, Register** (copy from **front\navbar** проекта **laravel-cards**)

**<!-- Authentication Links -->**

@guest

<li class="nav-item">

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

</li>

<li class="nav-item">

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

</li>

@else

<li class="nav-item dropdown">

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

{{ Auth::user()->name }} <span class="caret"></span>

</a>

<div class="dropdown-menu dropdown-menu-right size" aria-labelledby="navbarDropdown">

@admin

<a class="dropdown-item" href="#">Dashboard</a>

@endadmin

<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" style="display: none;">

@csrf

</form>

</div>

</li>

@endguest

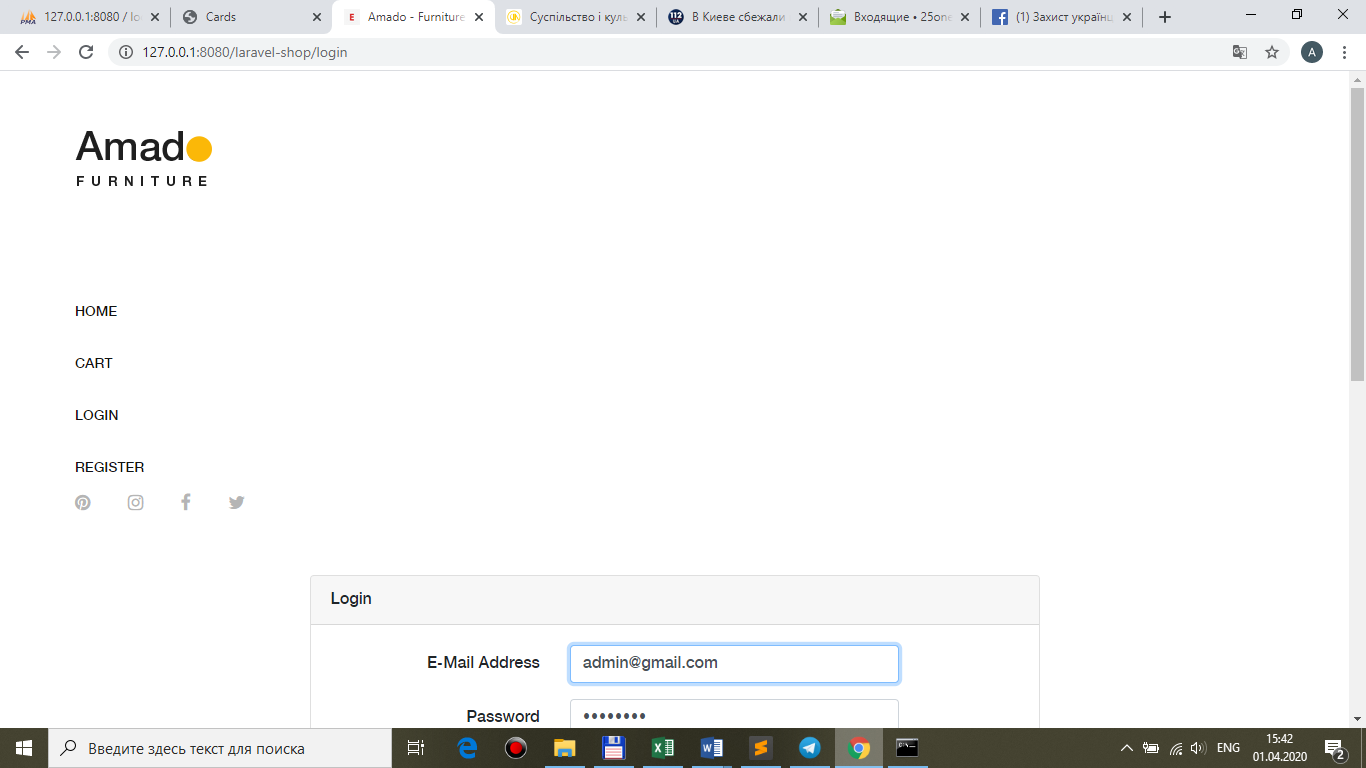
(**роут Dashboard пока отсутствует (href="#")**)

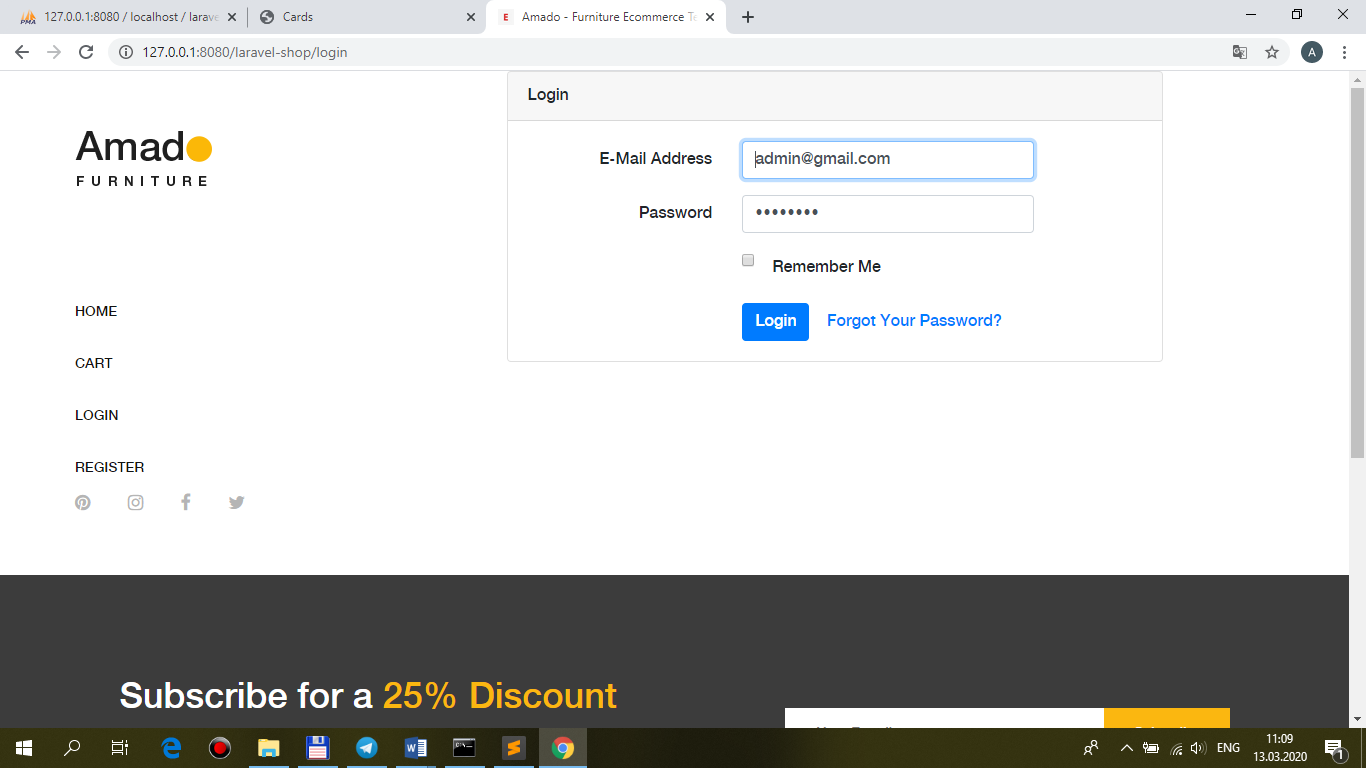
3.**Заменить** стандартные **laravel-views-login (views\auth)**, **register** на **кастомные** (copy from проект **laravel-cards**)

**+ !!!php artisan view:clear**

+**(ЭТО УЖЕ ДОЛЖНО БЫЛО БЫТЬ СДЕЛАНО РАНЕЕ )** **LoginController.php, RegisterController.php**

**protected $redirectTo = '/';**





(при необходимости – **подкорректировать** свою верстку для нормального отображения login/register-формы **(см. выше – начальное состояние верстки + исправленное состояние верстки)**)

+ **залогиниться/разлогиниться**

4.Описать директивы **admin (redac)** в **app\Providers\AppServiceProvider.php + подключить фасад Blade**

**use Illuminate\Support\Facades\Blade;**

public function **boot**()

{

**Blade::if('admin', function () {**

**return auth()->user()->role === 'admin';**

**});**

**Blade::if('redac', function () {**

**return auth()->user()->role === 'redac';**

**});**

}

5.Поставить **роут** для **Dashboard** в **resources\views\front\layout.blade.php**

**@admin**

<a class="dropdown-item" href="**{{ route('dashboard') }}**">Dashboard</a>

**@endadmin**

+описать его в **routes\web.php**

/\*

|--------------------------------------------------------------------------

| Backend

|--------------------------------------------------------------------------|

\*/

Route::prefix('')->**namespace('Back')**->group(function () {

Route::**name('dashboard')->get('/dashboard', 'AdminController@index')**;

});

**+(если не сделано ранее)** применить Route::prefix('')->**namespace('Front')**->group(function () **{…})**; для всех ранее стоящих роутов (**убрав Front\ из них**).

**!!!+php artisan view:clear**

6.Создать начальный **Back\AdminController.php** !c подключением **AdminRepository** вызовом из него **(!!!через trait)** метода **выборки всех товаров из БД** и **выводом их во вью** (в методе **index**):

namespace App\Http\Controllers\**Back**;

use Illuminate\Http\Request;

use App\ {

Http\Controllers\Controller,

Repositories\**Admin**Repository,

Http\Controllers\Traits\Indexable

};

class **Admin**Controller extends Controller

{

**use Indexable;**

/\*\*

\* Create a new controller instance.

\*

\* @return void

\*/

public function \_\_construct(**Admin**Repository $repository)

{

//$this->middleware('auth');

**$this->repository = $repository;**

**$this->namespace = 'back';**

}

}

7.Создать начальный **AdminRepository** c подключением модели **Product** и выборкой **всех товаров из БД**:

<?php

namespace App\Repositories;

use App\Models\ {

**Product**

};

class **Admin**Repository

{

/\*\*

\* The Model instance.

\*

\* @var \Illuminate\Database\Eloquent\Model

\*/

**protected $model\_product;**

/\*\*

\* Create a new AdminRepository instance.

\*

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

\*/

public function **\_\_construct(Product $product)**

**{**

**$this->model\_product = $product;**

}

/\*\*

\* Create a query for Product.

\*

\* @return \Illuminate\Database\Eloquent\Builder

\*/

public function **funcSelect($request, $parameters)**

**{**

**$query = $this->model\_product**

**->select('id', 'name', 'price', 'image')**

//->orderBy('price', 'asc');

**->orderBy($parameters['order'], $parameters['direction']);**

**//**if(isset($request->search)) $query->where('name', 'like', '%' . $request->search . '%');

return $query->get();

}

}

**+config\parameters.php**

…

**'back' => [**

**'order' => 'price',**

**'direction' => 'asc',**

**],**

…

**+php artisan config:cache**

8.Создать начальный **back\index.blade.php**

**@php**

**print\_r($products);**

**@endphp**

