Министерство науки и высшего образования Российской Федерации

**Федеральное государственное бюджетное образовательное учреждение**

**высшего профессионального образования**

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых»

**(ВлГУ)**

**Кафедра информационных систем и программной инженерии**

Рейтинг-контроль №5

по дисциплине "Основы разработки веб-приложений"

по теме: **«Знакомство с фреймворком Laravel»**

Выполнил:

студент гр. ПРИ-117

Журавлев Н.М.

Приняла:

Козаев А.Т.

Владимир 2020 г.

ЦЕЛЬ РАБОТЫ

Познакомиться со структурой приложений Laravel и разработать приложение “Корзина покупателя”.

ЗАДАНИЯ

1. Воспроизвести шаги из методички и показать работоспособность.
2. Реализовать 2 CRUD операции с продуктами на выбор.
3. Реализовать логику работы корзины покупателя с использованием объекта session (при перезагрузке страницы данные не теряются).

ВЫПОЛНЕНИЕ ЗАДАНИЙ

1. Воспроизвести шаги из методички и показать работоспособность.



Рисунок 1. Установка и проверка функционирования утилиты Composer.

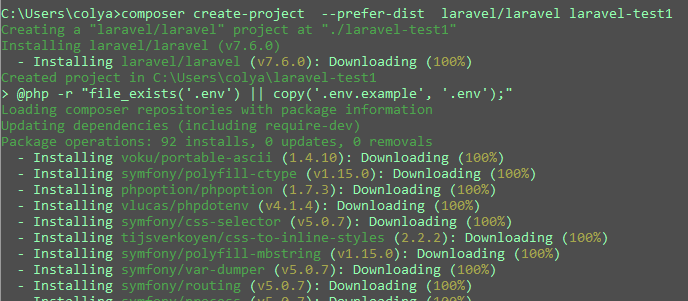


Рисунок 2. Создание Laravel-проекта.

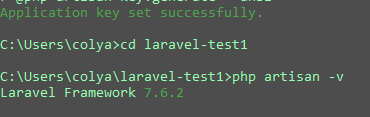


Рисунок 3. Проверка создания проекта.

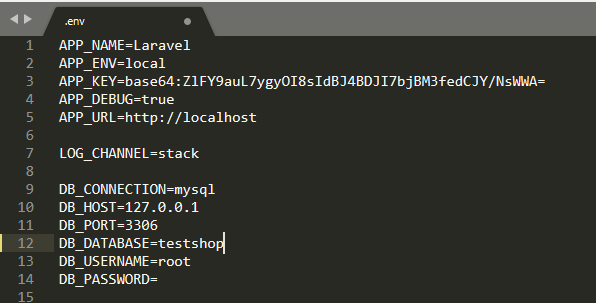


Рисунок 4. Изменение подключения к базе данных внутри проекта.

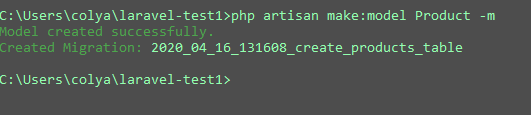


Рисунок 5. Создание модели для сущности Продукт.



Рисунок 6. Создание структуры таблицы базы данных для генерации

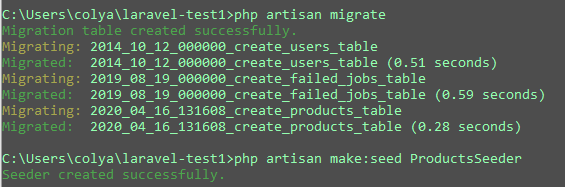


Рисунок 7. Миграция таблицы из проекта в базу данных.

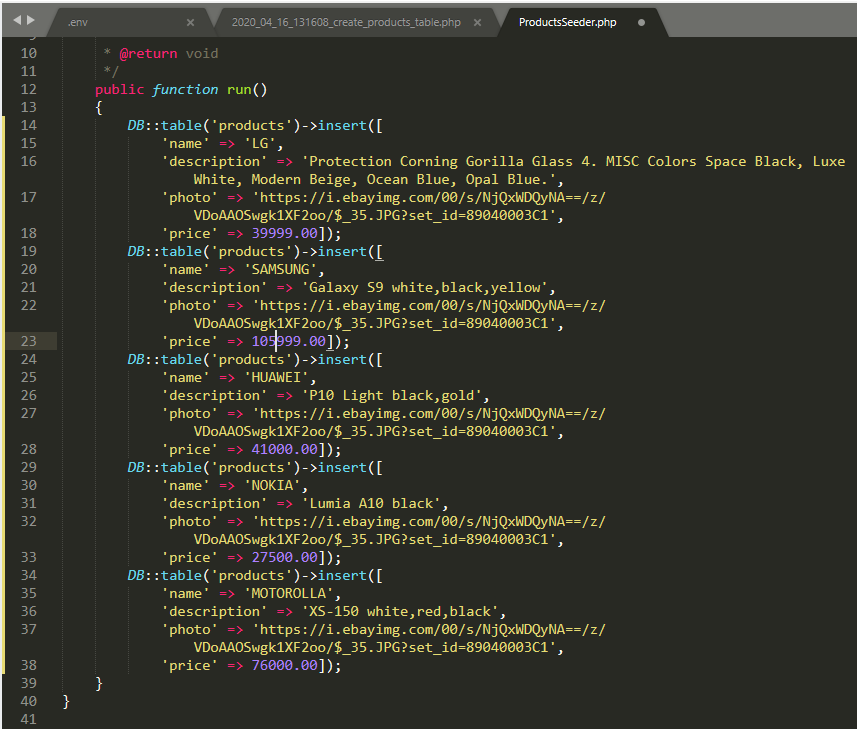


Рисунок 8. Создание метода для заполнения базы данных из приложения.

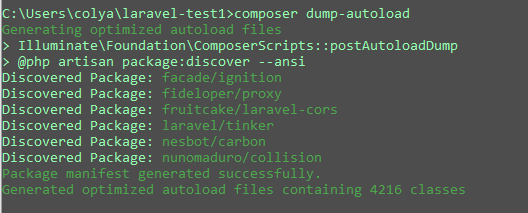


Рисунок 9-10. Заполнение базы данных.

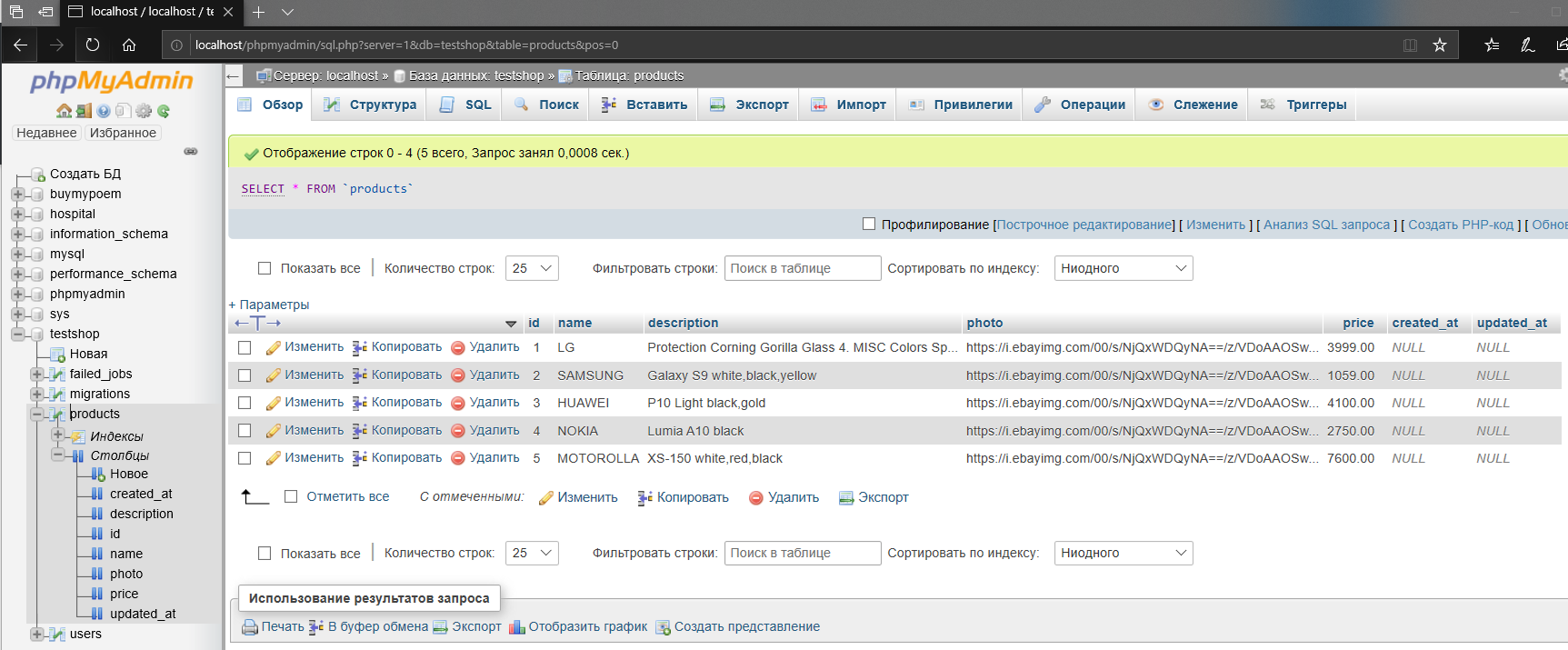


Рисунок 11. Созданная и заполненная база данных.



Рисунок 12. Создание контроллера через командную строку.



Рисунок 13. Редактирование контроллера для вывода всех записей таблицы базы данных.

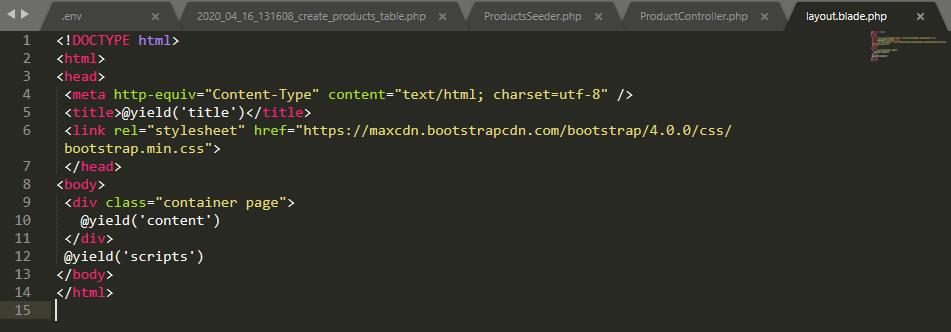


Рисунок 14. Настройка файла Layout



Рисунок 15. Создание связи между контроллером и представлением в файле web.php.

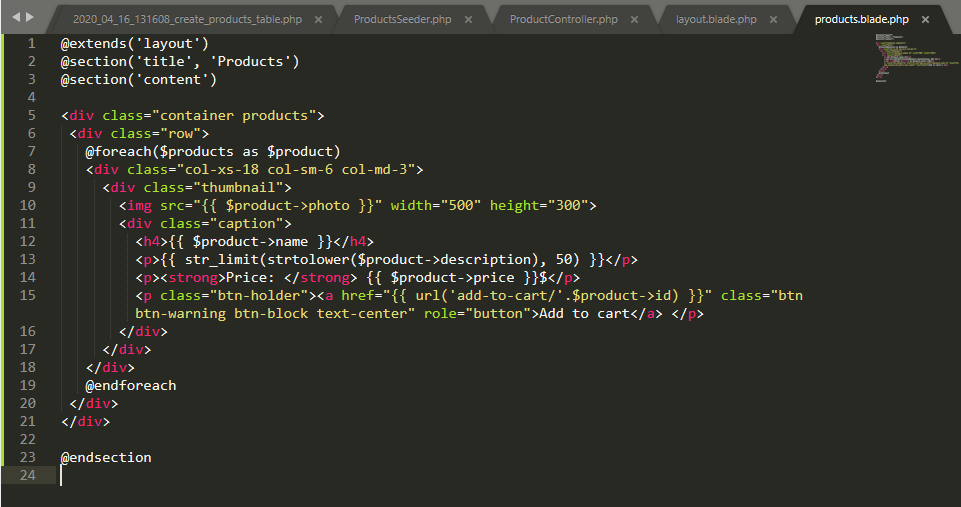
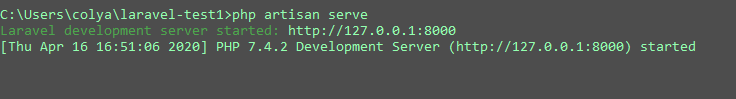


Рисунок 16. Создание представления



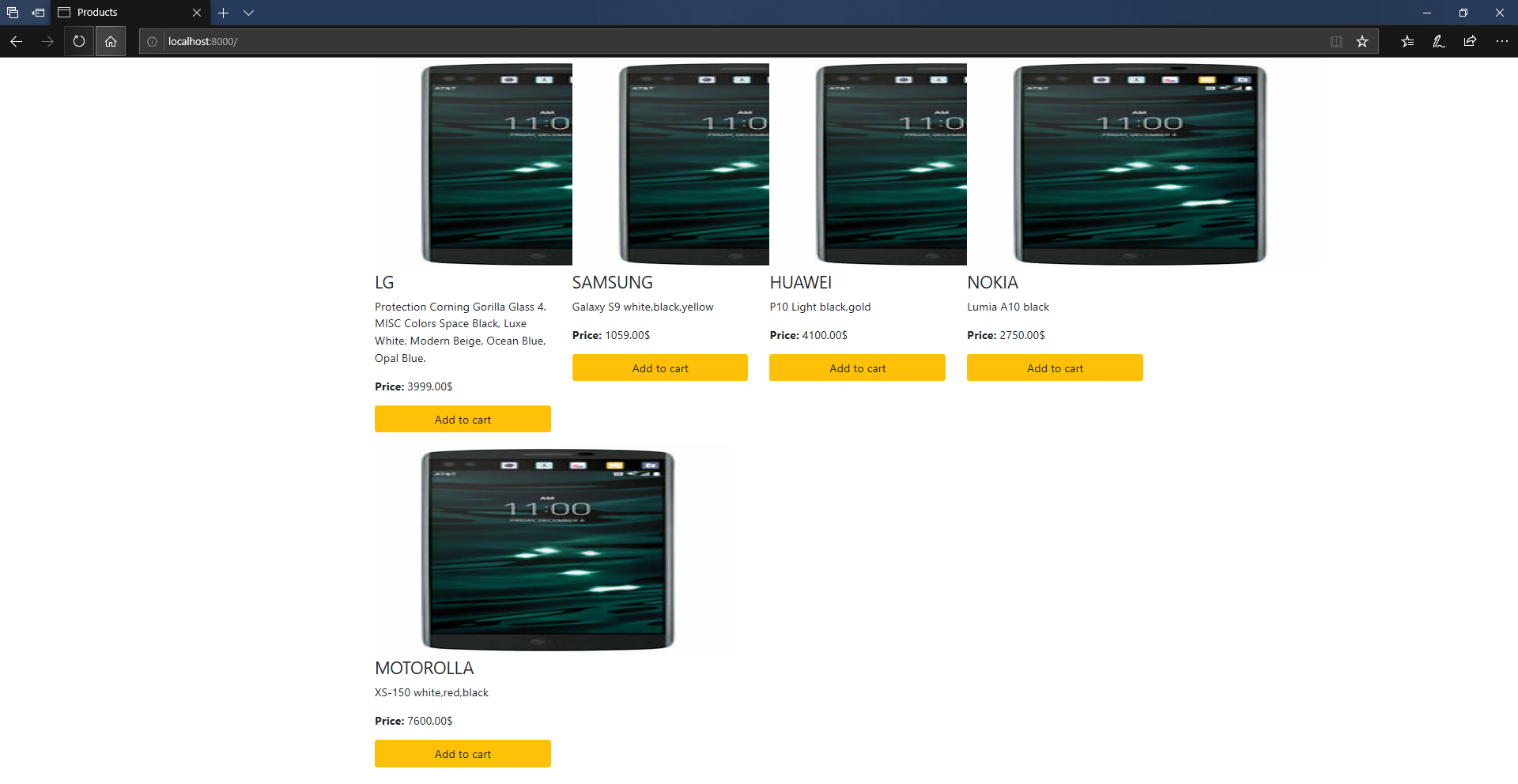


Рисунок 17. Результат запуска приложения.

2. Реализовать 2 CRUD операции с продуктами на выбор.

3. Реализовать логику работы корзины покупателя с использованием объекта session (при перезагрузке страницы данные не теряются).

В процессе выполнения заданий 2 и 3 был создан следующий функционал. Добавление и удаление продукции. Добавление продукта в корзину. Просмотр корзины. Очистка корзины.

Ниже приведен код разработанных контроллеров и представлений:

**Файл web.php:**

<?php

use Illuminate\Support\Facades\Route;

/\*

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

| Web Routes

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

|

| Here is where you can register web routes for your application. These

| routes are loaded by the RouteServiceProvider within a group which

| contains the "web" middleware group. Now create something great!

|

\*/

Route::get('/', 'ProductController@index');

Route::post('/create', 'ProductController@store');

Route::get('/create','ProductController@create');

Route::delete('/delete/product/{id}','ProductController@destroy');

Route::get('/bucket', 'BucketController@showBucket');

Route::post('/productintobucket/{id}', 'BucketController@addIntoBucket');

Route::post('/dropbucket', 'BucketController@dropBucket');

**Файл Product.php:**

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Product extends Model

{

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

}

**Файл ProductController.php:**

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Product;

class ProductController extends Controller

{

public function index()

{

$products = Product::all();

return view('products', compact('products'));

}

public function create()

{

return view('create');

}

public function store(Request $request)

{

$request->validate([

'name' => 'required',

'description' => 'required',

'price' => 'required',

]);

$product = Product::create(array(

'name' => $request->name,

'description' => $request->description,

'photo'=>$request->photo,

'price'=>$request->price));

return redirect('');

}

public function destroy(Request $request, $id)

{

$product = Product::find($id);

$product->delete();

return redirect('');

}

}

**Файл BucketController.php:**

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Product;

use Session;

class BucketController extends Controller

{

public function showBucket(Request $request){

$bucket = $request->session()->get('key');

$price\_sum=0;

if ($bucket == null)

{

return "Корзина пуста";

}

else

{

foreach ($bucket as $product)

{

$price\_sum+=$product->price;

}

return view('showBucket', ['bucket\_products'=>$bucket,'sum'=>$price\_sum]);

}

}

public function addIntoBucket(Request $request, $id){

$product = Product::find($id);

session()->push('key', $product);

return redirect('');

}

public function dropBucket(Request $request){

$request->session()->flush();

return redirect('');

}

}

**Файл products.blade.php:**

@extends('layout')

@section('title', 'Products')

@section('content')

<div class="container products">

<div class="row">

@foreach($products as $product)

<div class="col-xs-18 col-sm-6 col-md-3">

<div class="thumbnail">

<img src="{{ $product->photo }}" width="100" height="150">

<div class="caption">

<h4>{{ $product->name }}</h4>

<p>{{ $product->description }}</p>

<p><strong>Price: </strong> {{ $product->price }}$</p>

<p class="btn-holder">

<!--<a href="{{ url('productintobucket/'.$product->id) }}" class="btn btn-warning btn-block text-center" role="button">Добавить в корзину</a>-->

<form action="{{ url('productintobucket' , $product->id ) }}" method="POST">

<input type="hidden" name="\_method" value="POST">

<input type="hidden" name="\_token" value="{{ csrf\_token() }}">

<input type="submit" class="btn btn-danger" value="Добавить в корзину"/>

</form>

<form action="{{ url('delete/product' , $product->id ) }}" method="POST">

<input type="hidden" name="\_method" value="DELETE">

<input type="hidden" name="\_token" value="{{ csrf\_token() }}">

<input type="submit" class="btn btn-danger" value="Удалить продукт"/>

</form>

</p>

</div>

</div>

</div>

@endforeach

</div>

<p class="btn-holder"><a href="/create" class="btn btn-warning btn-block text-center" role="button">Создать продукт</a> </p>

<p class="btn-holder"><a href="/bucket" class="btn btn-warning btn-block text-center" role="button">Поcмотреть корзину</a> </p>

<p class="btn-holder"><form action="{{ url('dropbucket') }}" method="POST">

<input type="hidden" name="\_method" value="POST">

<input type="hidden" name="\_token" value="{{ csrf\_token() }}">

<input type="submit" class="btn btn-danger" value="Очистить корзину"/>

</form> </p>

</div>

@endsection

**Файл create.blade.php:**

@extends('layout')

@section('content')

<h1>Add New Product</h1>

<hr>

<form action="/create" method="post">

{{ csrf\_field() }}

<div class="form-group">

<label for="title">Product name</label>

<input type="text" class="form-control" id="ProductName" name="name">

</div>

<div class="form-group">

<label for="description">Product description</label>

<input type="text" class="form-control" id="ProductDescription" name="description">

</div>

<div class="form-group">

<label for="description">Product photo link</label>

<input type="text" class="form-control" id="ProductPhoto" name="photo">

</div>

<div class="form-group">

<label for="description">Product price</label>

<input type="text" class="form-control" id="ProductPrice" name="price">

</div>

@if ($errors->any())

<div class="alert alert-danger">

<ul>

@foreach ($errors->all() as $error)

<li>{{ $error }}</li>

@endforeach

</ul>

</div>

@endif

<button type="submit" class="btn btn-primary">Submit</button>

</form>

@endsection

**Файл showBucket.blade.php:**

@extends('layout')

@section('title', 'Bucket')

@section('content')

<h1>КОРЗИНА</h1>

<div class="container products">

<div class="row">

<div class="col-xs-18 col-sm-6 col-md-3">

<div class="thumbnail">

@forelse ($bucket\_products as $p)

<img src="{{ $p->photo }}" width="100" height="150">

<div class="caption">

<h4>{{ $p->name }}</h4>

<p>{{ $p->description }}</p>

<p>{{ $p->price }}</p>

</div>

@empty

<p>Корзина пуста</p>

@endforelse

</div>

<h4>Суммарная стоимость товаров в корзине = {{ $sum }}$</h4>

</div>

</div>

<p class="btn-holder">

<form action="{{ url('dropbucket') }}" method="POST">

<input type="hidden" name="\_method" value="POST">

<input type="hidden" name="\_token" value="{{ csrf\_token() }}">

<input type="submit" class="btn btn-danger" value="Очистить корзину"/>

</form>

</p>

</div>

@endsection

Результаты запуска приложения.

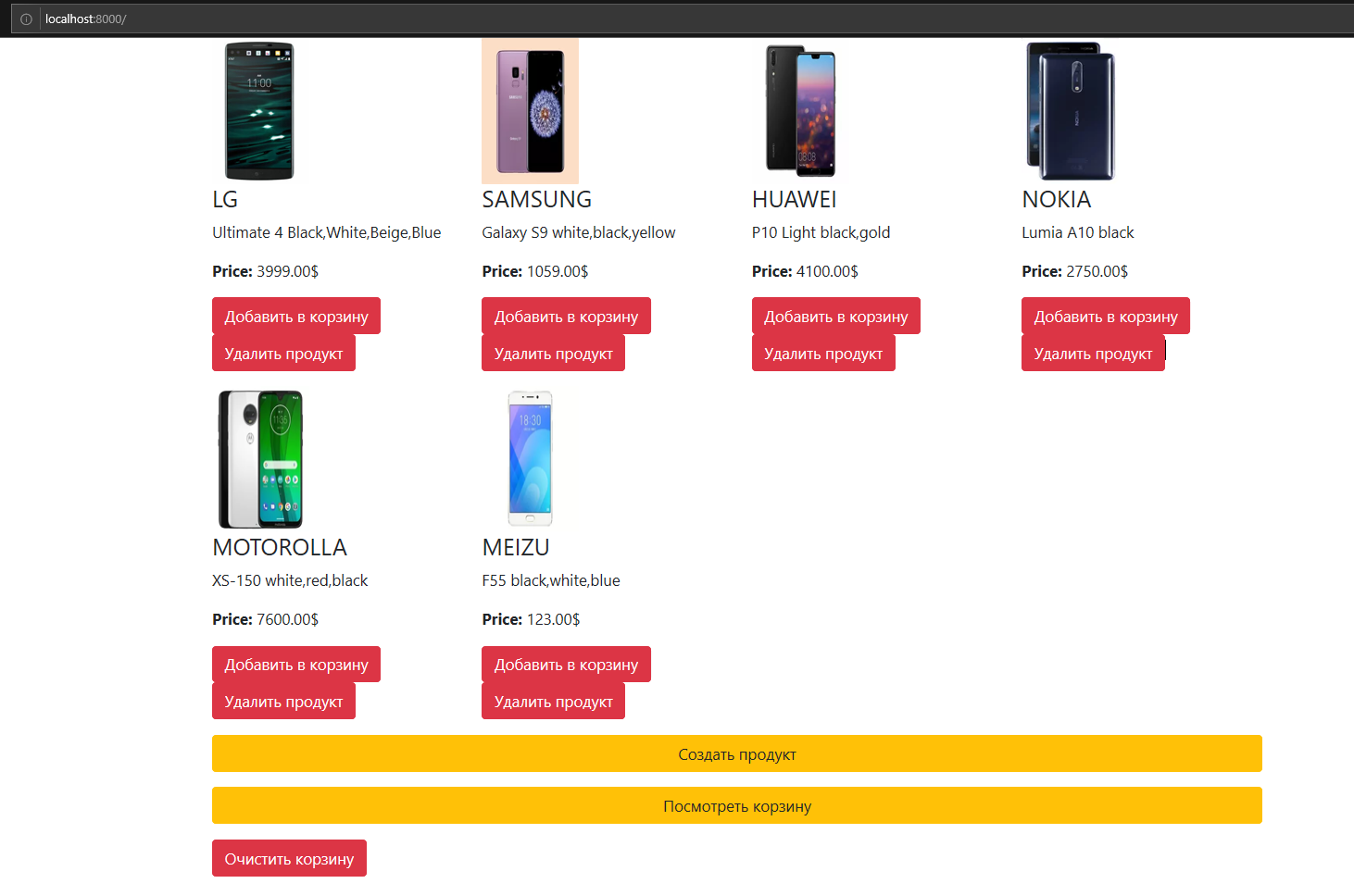


Рисунок 18. Стартовая страница отображения всех записей таблицы Products.

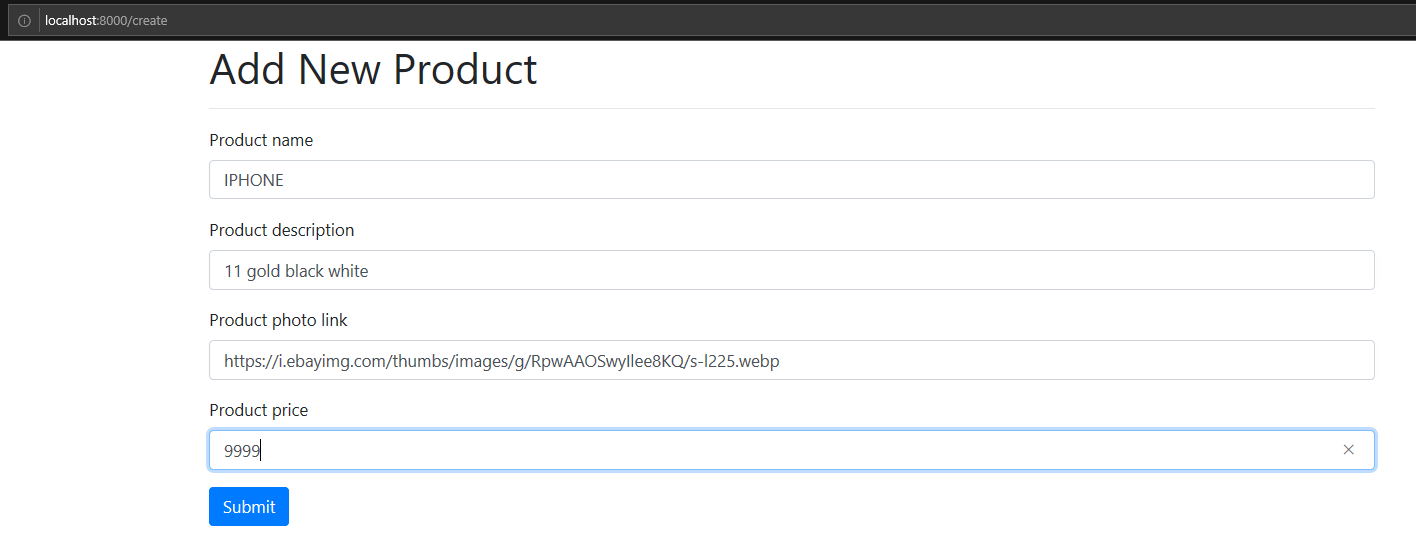


Рисунок 19. Добавление записи в таблицу Products.

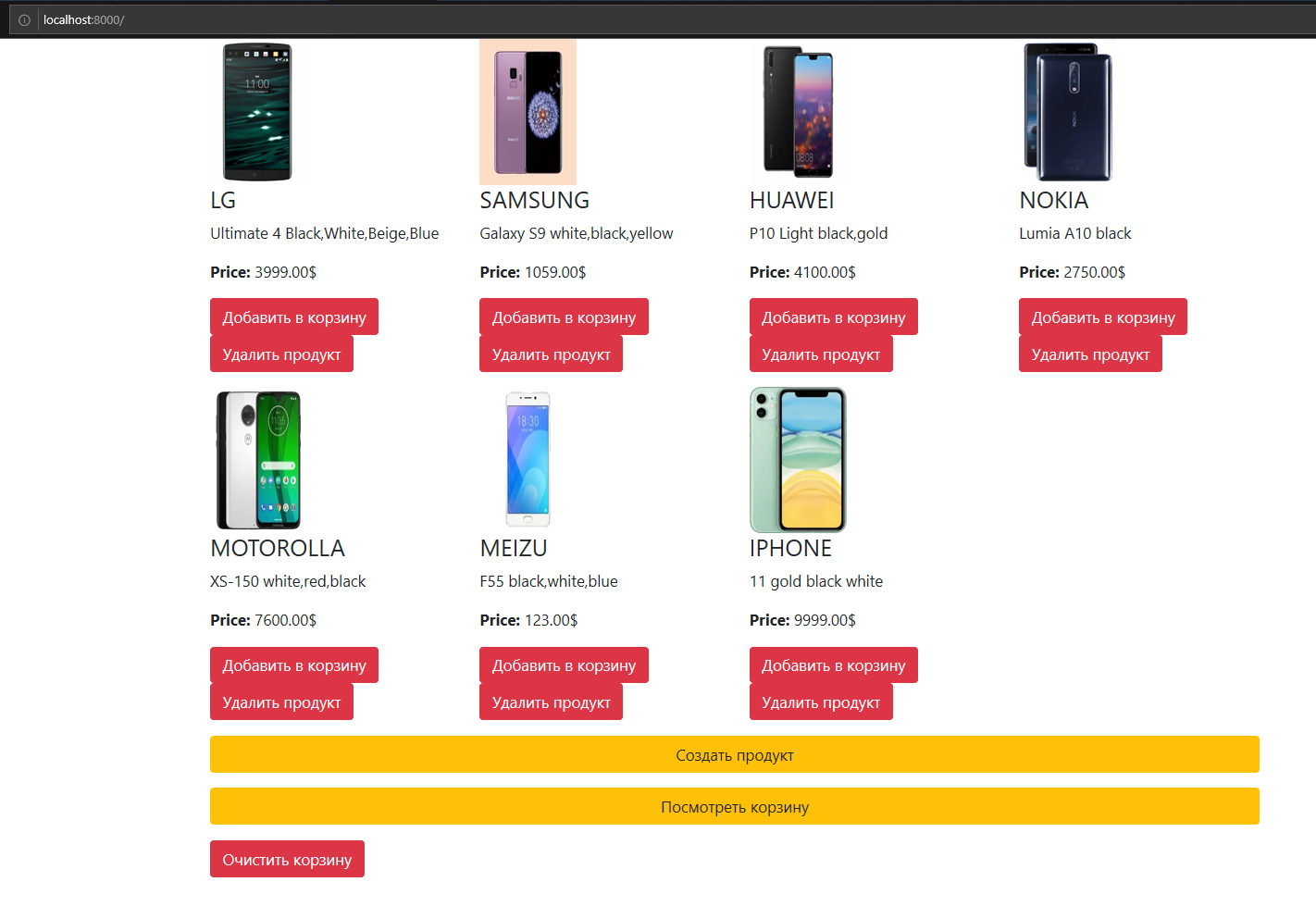


Рисунок 20. Подтверждение добавления записи в таблицу Products.

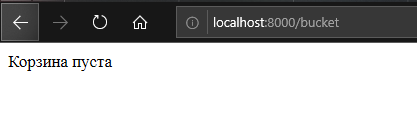


Рисунок 21. Попытка просмотра содержимого корзины до добавления в нее продуктов.

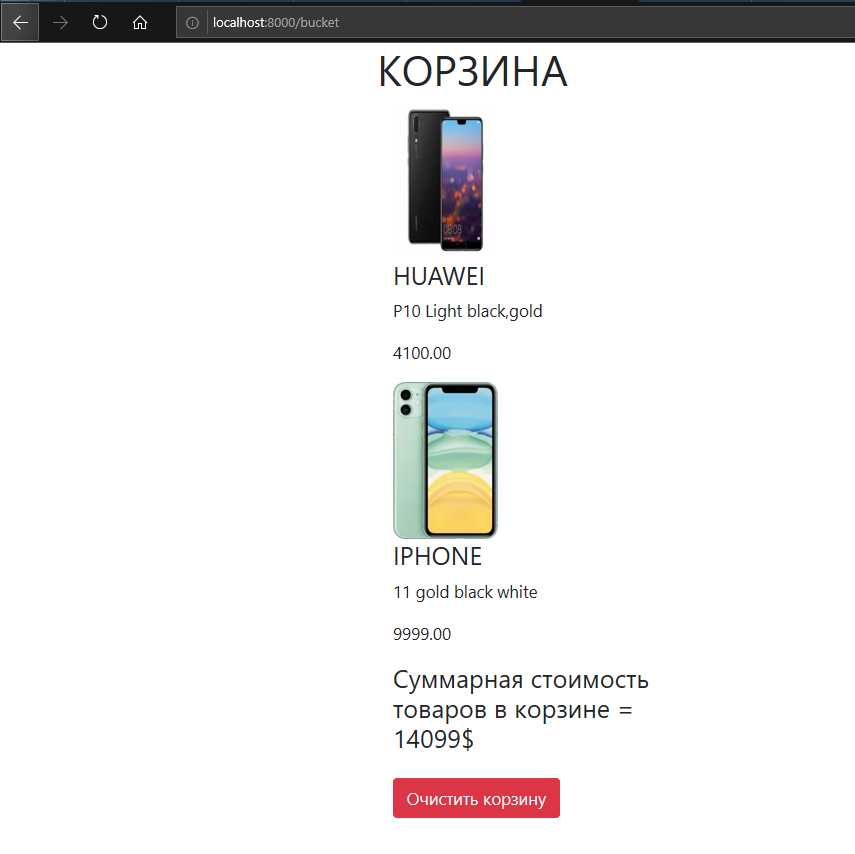


Рисунок 21. Попытка просмотра содержимого корзины после добавления в нее продуктов.

ВЫВОД

В ходе работы было проведено Ознакомление со структурой приложений Laravel и разработано приложение “Корзина покупателя”.