

Министерство науки и высшего образования Российской Федерации  
**Федеральное государственное бюджетное образовательное учреждение  
высшего образования**

«Владимирский государственный университет  
имени Александра Григорьевича и Николая Григорьевича Столетовых»  
(ВлГУ)

Кафедра информационных систем и программной инженерии

**Лабораторная работа №5**  
**по дисциплине**  
**«Основы разработки веб-приложений»**  
**ЗНАКОМСТВО С ФРЕЙМВОРКОМ LARAVEL**

**Выполнил:**  
ст. гр. ПРИ-120  
Д. А. Грачев

**Принял:**  
Старший преподаватель  
кафедры ИСПИ  
Спирин И. В.

Владимир, 2023

## ЦЕЛЬ РАБОТЫ

Познакомиться со структурой приложений Laravel и разработать приложение «Корзина покупателя».

## ВЫПОЛНЕНИЕ РАБОТЫ

1. Задание:
  - 1.1. Воспроизвести шаги из методички и показать работоспособность.
  - 1.2. Реализовать 2 CRUD операции с продуктами на выбор.
  - 1.3. Реализовать логику работы корзины покупателя с использованием объекта session (при перезагрузке страницы данные не теряются).
2. Создадим новый Laravel проект с помощью команды «php composer.phar create-project --prefer-dist laravel/laravel lab5»

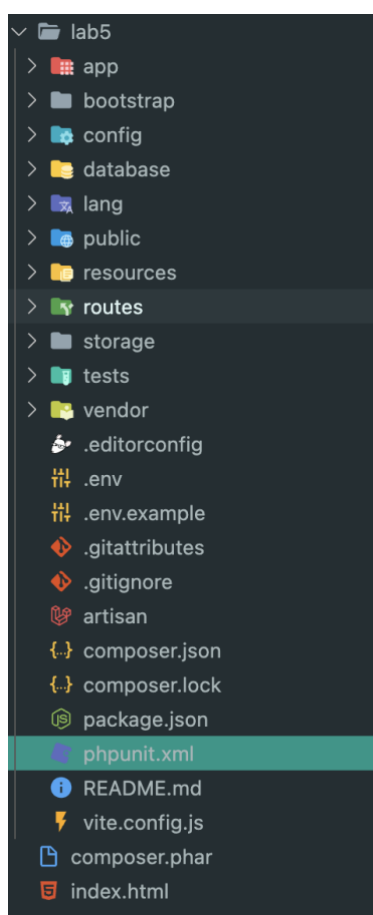


Рисунок 1. Созданный проект

3. Создадим новую модель с помощью команды «php artisan make:model Product -m»

```
Product.php 2023_02_10_091714_create_products_table.php ×
lab5 > database > migrations > 2023_02_10_091714_create_products_table.php
1
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('products', function (Blueprint $table) {
17             $table->id();
18             $table->timestamps();
19         });
20     }
21
22     /**
23      * Reverse the migrations.
24      *
25      * @return void
26      */
27     public function down()
28     {
29         Schema::dropIfExists('products');
30     }
31 };
32
```

Рисунок 2. Созданная миграция

```
14 public function up()
15 {
16     Schema::create('products', function (Blueprint $table) {
17         $table->id();
18         $table->string("name", 255)->nullable();
19         $table->text("description")->nullable();
20         $table->string("photo", 255)->nullable();
21         $table->decimal("price", 6, 2);
22         $table->timestamps();
23     });

```

Рисунок 3. Добавление полей модели

С помощью команды «php artisan migrate» проведем миграцию, чтобы таблица отобразилась в БД

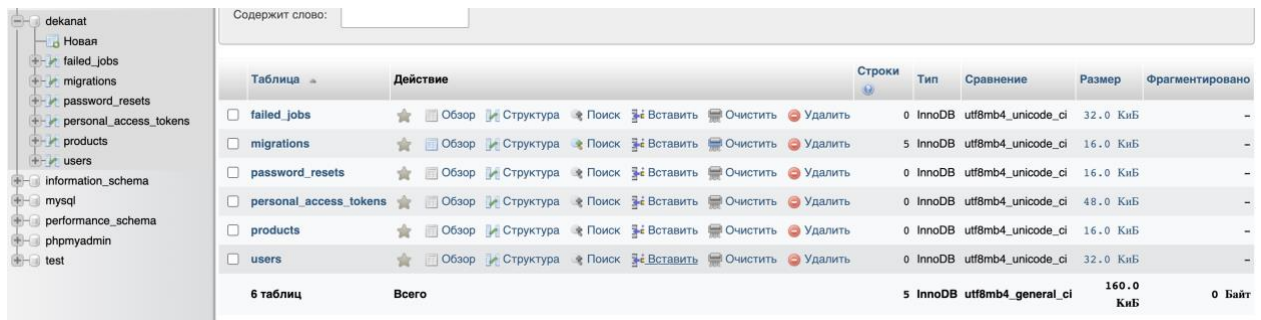


Таблица	Действие	Строки	Тип	Сравнение	Размер	Фрагментировано
<input type="checkbox"/> failed_jobs	★ Обзор Структура Поиск Вставить Очистить Удалить	0	InnoDB	utf8mb4_unicode_ci	32.0 КиБ	-
<input type="checkbox"/> migrations	★ Обзор Структура Поиск Вставить Очистить Удалить	5	InnoDB	utf8mb4_unicode_ci	16.0 КиБ	-
<input type="checkbox"/> password_resets	★ Обзор Структура Поиск Вставить Очистить Удалить	0	InnoDB	utf8mb4_unicode_ci	16.0 КиБ	-
<input type="checkbox"/> personal_access_tokens	★ Обзор Структура Поиск Вставить Очистить Удалить	0	InnoDB	utf8mb4_unicode_ci	48.0 КиБ	-
<input type="checkbox"/> products	★ Обзор Структура Поиск Вставить Очистить Удалить	0	InnoDB	utf8mb4_unicode_ci	16.0 КиБ	-
<input type="checkbox"/> users	★ Обзор Структура Поиск Вставить Очистить Удалить	0	InnoDB	utf8mb4_unicode_ci	32.0 КиБ	-
6 таблиц	Всего		5 InnoDB	utf8mb4_general_ci	160.0 КиБ	0 Байт

Рисунок 4. Созданная база данных

4. С помощью команды «php artisan make:seed ProductsSeeder» создадим скрипт для заполнения БД начальными данными

```
1 <?php
2
3 namespace Database\Seeders;
4
5 use Illuminate\Database\Console\Seeds\WithoutModelEvents;
6 use Illuminate\Database\Seeder;
7
8 class ProductsSeeder extends Seeder
9 {
10     /**
11      * Run the database seeds.
12      *
13      * @return void
14      */
15     public function run()
16     {
17         //
18     }
19 }
20
```

Рисунок 5. Созданный файл

```

1 <?php
2
3 namespace Database\Seeders;
4
5 use Illuminate\Database\Console\Seeds\WithoutModelEvents;
6 use Illuminate\Database\Seeder;
7
8 class ProductsSeeder extends Seeder
9 {
10     /**
11      * Run the database seeds.
12      *
13      * @return void
14      */
15     public function run()
16     {
17         DB::table('products')->insert([
18             'name' => 'LG',
19             'description' => 'Protection Corning Gorilla Glass 4. MISC Colors Space Black, Luxe White, Modern Beige, Oc
20             'photo' => 'https://i.ebayimg.com/00/s/NjQxWDQyNA==/z/VDoAA0Swgk1XF2oo/$_35.JPG?set_id=89040003C1',
21             'price' => 39999.00
22         ]);
23     }
24 }
25
26

```

Рисунок 6. Добавление продукта

Самостоятельно добавил 5 записей в таблицу

```

...
DB::table('products')->insert([
    'name' => 'Samsung',
    'description' => 'S22 Ultra 5G 6.7"HD Smart Phone 6800mAh Face ID Android 4+64+128GB TF Card',
    'photo' => 'https://i.ebayimg.com/images/g/AQgAA0SwDkViWTQe/s-l500.jpg',
    'price' => 7999.00
]);
DB::table('products')->insert([
    'name' => 'Google',
    'description' => 'Google Pixel 6 - 128GB - Sorta Seafoam (Google Ed. and Unlocked)',
    'photo' => 'https://i.ebayimg.com/images/g/6Z0AA0SwCmJjRcc8/s-l1600.jpg',
    'price' => 18263.00
]);
DB::table('products')->insert([
    'name' => 'Samsung Folder',
    'description' => 'Samsung Galaxy Folder 2 SM-G1650 dual-SIM Android Mobile Flip Phone 4G LTE 8MP',
    'photo' => 'https://i.ebayimg.com/images/g/uKIAA0SwfURdIhAq/s-l500.jpg',
    'price' => 10125.00
]);
DB::table('products')->insert([
    'name' => 'Apple',
    'description' => 'Apple iPhone X 64 and 256 - Back and Silver NO FACE ID-',
    'photo' => 'https://i.ebayimg.com/images/g/hDAAA0SwqgpjrV76/s-l1600.jpg',
    'price' => 11714.00
]);
DB::table('products')->insert([
    'name' => 'LG',
    'description' => 'LG Wine Smart D486 4G LTE 4GB ROM 3.5" Android Flip Keyboard Unlocked Smartphone',
    'photo' => 'https://i.ebayimg.com/images/g/73EAA0SwDfFjrRFF/s-l1600.jpg',
    'price' => 6209.00
]);

```

Рисунок 7. Добавленные данные

5. С помощью двух команд создадим записи в базе данных

5.1.«php ../composer.phar dump-autoload»

```
gggchaseggg@MacBook-Pro-Daniil lab5 % php ../composer.phar dump-autoload
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi

INFO Discovering packages.

laravel/sail ..... DONE
laravel/sanctum ..... DONE
laravel/tinker ..... DONE
nesbot/carbon ..... DONE
nunomaduro/collision ..... DONE
nunomaduro/termwind ..... DONE
spatie/laravel-ignition ..... DONE

Generated optimized autoload files containing 5239 classes
```

Рисунок 8. Работа команды

## 5.2. «php artisan db:seed --class=ProductSeeder»

```
gggchaseggg@MacBook-Pro-Daniil lab5 % php artisan db:seed --class=ProductsSeeder

INFO Seeding database.

gggchaseggg@MacBook-Pro-Daniil lab5 %
```

	id	name	description	photo	price	created_at	updated_at
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	1	LG	Protection Corning Gorilla Glass 4. MISC Colors Sp...	https://i.ebayimg.com/00/s/NjQxWDQyNA==/z/VDoAAOSw...	3999.00	NULL	NULL
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	2	Samsung	S22 Ultra 5G 6.7"HD Smart Phone 6800mAh Face ID An...	https://i.ebayimg.com/images/g/AQgAAOSwDkVIWTQe/s-...	7999.00	NULL	NULL
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	3	Google	Google Pixel 6 - 128GB - Sorta Seafoam (Google Ed....	https://i.ebayimg.com/images/g/6Z0AAOSwCmJlRcc8/s-...	1826.00	NULL	NULL
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	4	Samsung Folder	Samsung Galaxy Folder 2 SM-G1650 dual-SIM Android ...	https://i.ebayimg.com/images/g/uKIAAOSwUrdlhAq/s-...	1012.00	NULL	NULL
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	5	Apple	Apple iPhone X 64 and 256 - Back and Silver NO FAC...	https://i.ebayimg.com/images/g/hDAAOSwqgppjv76/s-...	1171.00	NULL	NULL
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	6	LG	LG Wine Smart D486 4G LTE 4GB ROM 3.5" Android Fli...	https://i.ebayimg.com/images/g/73EAAOSwDfJrRFF/s-...	6209.00	NULL	NULL

Рисунок 9. Добавленные данные в БД

6. С помощью команды «php artisan make:controller ProductsController» создадим контроллер продуктов

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class ProductsController extends Controller
8  {
9      public function index() {
10         $products = Product::all();
11
12         return view('products', compact('products'));
13     }
14
15 }
16
```

Рисунок 10. Контроллер продуктов

7. В папке «resources/views» создадим файл «layout.blade.php»

```
lab5 > resources > views > layout.blade.php
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5      <title>@yield('title')</title>
6      <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
7  </head>
8  <body>
9      <div class="container page">
10         @yield('content')
11     </div>
12     @yield('scripts')
13 </body>
14 </html>
15
```

Рисунок 11. Файл layout.blade.php

8. В папке «resources/views» создадим файл «products.blade.php»

```
lab5 > resources > views > products.blade.php
1  extends('layout')
2  @section('title', 'Products')
3  @section('content')
4
5  <div class="container products">
6      <div class="row">
7          @foreach($products as $product)
8              <div class="col-xs-18 col-sm-6 col-md-3">
9                  <div class="thumbnail">
10                     
11                     <div class="caption">
12                         <h4>{{ $product->name }}</h4>
13                         <p>{{ str_limit(strtolower($product->description), 50) }}</p>
14                         <p><strong>Price: </strong> {{ $product->price }}</p>
15                         <p class="btn-holder"><a href="{{ url('add-to-cart/'.$product->id) }}" class="btn btn-warning btn-block text-c
16                     </div>
17                 </div>
18             </div>
19         @endforeach
20     </div>
21 </div>
22
```

Рисунок 12. Файл products.blade.php

9. Поменяем стандартную страницу на страницу товаров

```
lab5 > routes > web.php
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4
5  /*
6  |-----
7  | Web Routes
8  |-----
9  |
10 | Here is where you can register web routes for your application. These
11 | routes are loaded by the RouteServiceProvider within a group which
12 | contains the "web" middleware group. Now create something great!
13 |
14 */
15
16 Route::get('/', 'ProductsController@index' );
17
```

Рисунок 13. Замененный стандартный путь



## 10. Запустим страничку

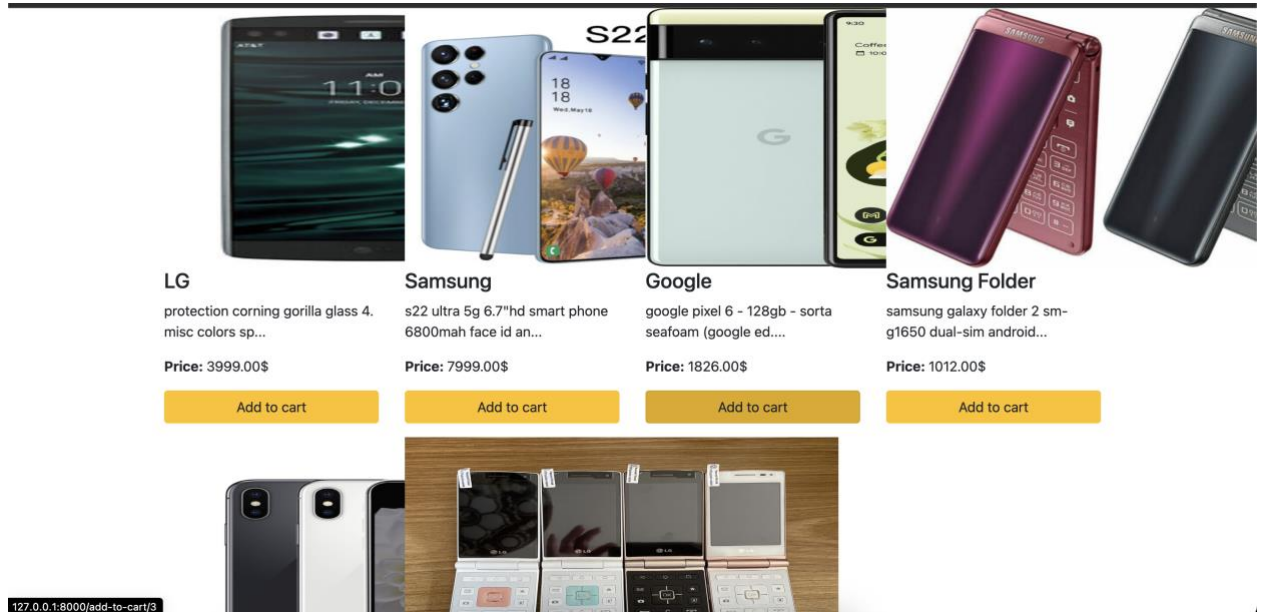


Рисунок 14. Страница с товарами

## 11. Дополним контроллер методами для CRUD операций

```
15 public function create() { return view('product.create'); }
16
17 public function store(Request $request)
18 {
19     $request->validate([
20         'photo' => 'required',
21         'name' => 'required',
22         'description' => 'required',
23         'price' => 'required',
24     ]);
25     Product::create($request->post());
26     return redirect()->route('product.index')->with('success','Product has been created successfully.');
```

```
27 }
28
29 public function edit(Product $product) { return view('update',compact('product')); }
30
31 public function update(Request $request, Product $product)
32 {
33     $request->validate([
34         'photo' => 'required',
35         'name' => 'required',
36         'description' => 'required',
37         'price' => 'required',
38     ]);
39     $product->fill($request->post())->save();
40     return redirect()->route('product.index')->with('success','Product has been updated successfully');
```

```
41 }
42
43 public function destroy(Product $product)
44 {
45     $product->delete();
46     return redirect()->route('product.index')->with('success','Product has been deleted successfully');
```

```
47 }
48
49 public function cart(Product $product)
50 {
51     $products = Product::all();
```

Рисунок 15. Листинг контроллера



## 12. Создадим карточку товара

```
66
67 <div class="container products">
68   <div class="row">
69     @foreach($products as $product)
70       <div class="col-xs-18 col-sm-6 col-md-3">
71         <div class="thumbnail">
72           
73         <div class="caption">
74           <h4>{{ $product->id }}.{{ $product->name }}</h4>
75           <p>{{ \Illuminate\Support\Str::limit(strtolower($product->description), 50) }}</p>
76           <p><strong>Price: </strong> {{ $product->price }}$</p>
77           <p class="btn-holder">
78             <a class="btn btn-primary" href="{{ route('product.edit',$product->id) }}">Редактировать</a>
79             <form action="{{ route('product.destroy',$product->id) }}" method="Post">
80               @csrf
81               @method('DELETE')
82               <button type="submit" class="btn btn-danger">Удалить</button>
83             </form>
84
85             <form action="{{ route('product.cart') }}" method="Post">
86               <input type="hidden" name="id" value="{{ $product->id }}">
87               @csrf
88               <button type="submit" class="btn btn-warning">В корзину</button>
89             </form>
90           </p>
91         </div>
92       </div>
93     </div>
94   @endforeach
```

Рисунок 16. Листинг карточки товара

## 13. Создадим карточку товара на страницу корзины товаров

```
24 <div class="container products">
25   <div class="row">
26     @foreach($sesStr as $cartProductId)
27       <?php
28       if ($cartProductId !== "") {
29         if ($products->has($cartProductId - 1)) {
30           $product = $products["$cartProductId" - 1];
31         }
32       }
33       ?>
34       @if($cartProductId !== "")
35       @if($products->has($cartProductId - 1))
36       <div class="col-xs-9 col-sm-6 col-md-3 border border-warning rounded mr-3 mt-3">
37         <div class="thumbnail">
38           
39         <div class="caption">
40           <h4>{{ $product->id }}.{{ $product->name }}</h4>
41           <p>{{ \Illuminate\Support\Str::limit(strtolower($product->description), 50) }}</p>
42           <p><strong>Price: </strong> {{ $product->price }}$</p>
43         </div>
44       </div>
45     </div>
46   @endif
47 @endif
48 @endforeach
49 </div>
50 </div>
```

Рисунок 17. Листинг карточки товара в корзине

## 14. С помощью сессии сохраним товары в корзине

```

2  <?php
3      $sesStr = session('cart');
4
5      $sesStr[] = $_POST["id"];
6      session(['cart' => $sesStr]);

```

Рисунок 18. Листинг работы с сессией

## 15. Настроим пути в приложении

```

17 Route::resource('product', ProductsController::class );
18 Route::post('cart', [ProductsController::class, "cart"])->name('product.cart');
19 Route::get('cart', [ProductsController::class, "clearsession"])->name('product.cart');

```

Рисунок 19. Пути в приложении

## 16. Скриншоты работы приложения

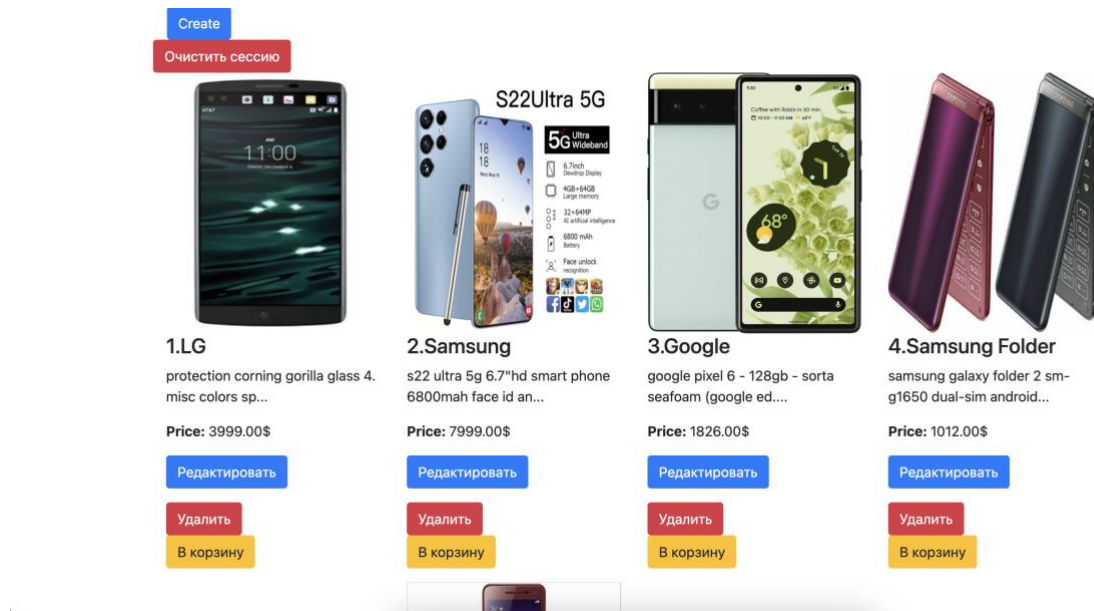


Рисунок 20. Страница списка товаров

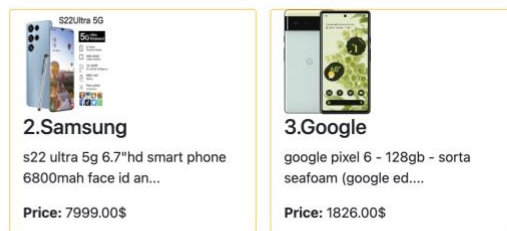


Рисунок 21. Страница корзины

## Edit product

[Back](#)

**Photo:**

<https://i.ebayimg.com/images/g/AQgAAOSwDkVIWTQe/s-l500.jpg>

**Name:**

Samsung

**Description:**

S22 Ultra 5G 6.7"HD Smart Phone 6800mAh Face ID Android 4+64+128GB TF Card

**Price:**

7999.00

[Submit](#)

Рисунок 22. Страница редактирования товара

## Add product

**product Name:**

Name

**Description:**

Description

**Price:**

Price

**Photo Url:**

Photo url

[Create](#)

Рисунок 23. Форма создания товара

## ВЫВОД

В ходе выполнения лабораторной работы ознакомился со структурой приложения Laravel и разработал приложение «Корзина покупателя».