Silk Road Middle East Store

Contents

Figures & Codes

Abstract

After the development in programming and applications that help and reduce human burdens. And after entering many applications and software that facilitate work and reduce the effort in many works, including arithmetic, administrative, and others. In this research, we will create an application for electronic shopping, which solves the problem of dealing with customers and facilitates the task for them in terms of searching for the product to be purchased and purchasing information to deliver the purchased goods. Where the programming language PHP was used as the main language in building the application based on the Laravel framework, as well as HTML, CSS, and Javascript were used to build the destinations. Where the application was built with easy-to-handle and understand interfaces that provide speed in reaching the needs of the customer who visits the application.

Introduction

The development of the web and social networking sites led to the development of trade and the exchange of goods. As programming enters such a field, which is very important in daily life because most people need to shop and acquire the necessary goods they need. And as we can see that the process of buying and choosing products from among many companies and the characteristics of each commodity, puts the customer at a loss because of the large number of products by international companies and the many features that differ from one product to another. In previous years, many applications were launched in this field, including Amazon, Alibaba, and many others. These applications have many products that most customers need. From this idea, the idea was invented, as the application of electronic shopping is very useful for most companies or merchants who have shops or selling companies. This online shopping application provides many features that make it easier for the customer to search for the products to be purchased or the company he wants to buy from. It also shows him all the specifications he needs for that product to understand the product or compare other products that compete with it in terms of specifications or price. As most of the interfaces are clear, understandable, and easy to deal with, to help the customer navigate the application easily and effortlessly.

Programming Languages

The programming languages used to build and develop this application are divided into two parts: the first section is back-end development and front-end development

1- Backend Development

These include programming languages such as the framework used by Laravel for PHP, as well as MySQL databases

1-1- PHP – Laravel

I- PHP

PHP (PHP) is a server-side scripting language that was originally developed as a family of Perl-based applications. It is primarily intended for the development and programming of online applications, but it may also be used for more general purposes, such as the creation of stand-alone programs that are not web-based.

The most widely accepted meaning of PHP is (PHP: Hypertext Preprocessor).

It's strong enough to power the world's most popular blogging site (WordPress).

It's powerful enough to run the world's most popular social network (Facebook).

It is comprehensive and diverse enough to constitute the world's largest online encyclopedia (Wikipedia).

PHP Features

- Compatibility

- Ease of learning and use

- the speed

- Use and interact with other languages Protection and Powers

- Open source, scalability, and support

- Free

- stability Strong support for databases

II- Laravel

Laravel is one of the most widely used frameworks today. It is an open-source web application framework for the programming language PHP, and it was created by (Taylor Otwell) in 2011. It provides an integrated, smooth, and easy work environment, and it is based on the famous (MVC) method that separates the (Model) from the (View) (Controller).

Laravel is the most popular PHP framework, and it has a plethora of learning materials.

This framework makes it easy to build authentication and authentication, as well as arrange logical switching and regulate resource access. The Laravel framework is known for its extremely secure Internet applications since it uses password methods (Passwords) to prevent the password from being recorded in plain text in the database, and it employs the (Bcrypt Hashing Technique) algorithm to produce an encrypted password.

1-2- MySQL

MySQL is the most widely used relational database management system in the world, thanks to its open-source GNU GPL license.

MySQL is based on three primary principles: speed, stability, and usability.

- MySQL Advantages

The most significant characteristics of MySQL database systems are speed and stability, which explains their widespread usage by developers, managers, and users all around the globe, and we'll go over what makes this rule unique in more depth today.

- MySQL's performance

One of the most important features of the MySQL relational database management system is the time it takes to execute a query and return the results to the query. This is because MySQL uses a multitasking structure, such as indexing, nodes, and cached queries in memory, which resulted in high performance without the need for any custom programming from the user.

- MySQL Features
    - Speed
    - Reliability
    - Security
    - Expandability and portability
    - Ease of use
    - Compliance with existing standards
    - Wide application support
    - Easy License Policy

2- Front-end Development

2-1- HTML

(Hypertext Markup Language) Sometimes known as HTML, and also known as Hypertext Markup Language, it is a unique markup language used for designing and constructing web pages. It is the core structure and infrastructure of web pages, offering a thorough description of how they will be displayed. It features a technique for presenting the contents of a website by separating it into headings and paragraphs, all of which are done using tags.

The purpose of this language, which was created in 1990 by a scientist named Tim Berners-Lee, is to make it easier for scientists at other universities to access the research they publish. HTML is a programming language for presenting data (for example, scientific research) on the Internet. The browser's translation of HTML directives is what you see when you visit any page on the network.

2-2- CSS

CSS, also known as Cascading Style Sheets, is one of the most used approaches for creating web pages, along with HTML. It is distinguished by the addition of elements and characteristics to the structure of web pages, such as colors, fonts, and patterns. Haakon Film Lai, a programmer, initially invented CSS on October 10, 1994. The CSS language has grown in importance in tandem with the development of HTML to complement each other, and its integration has improved the clarity, convenience, and importance of presenting sites across the web for users.

Language properties in CSS

- The ability to mix many HTML and CSS choices and codes to create integrated themes.
- Flexibility, as well as the simplicity with which it may be used and applied.
- sensitivity to the case.
- Allows the user to utilize a variety of commands and tags to create an integrated web page.
- First, learn the fundamentals of HTML.
- Background styles, size, elements, alignment, font colors, and more may all be used to control page proportions and look.
- No matter how big the site is, there should be consistency in the size of the web pages.
- Increase the download speed while lowering the site's hosting expenses.
- Easily achieving compatibility between the site's web pages and browsers, and so adjusting between them, resulting in enticing more people to enjoy visiting the site.
- Allowing the user to put display options in his hands based on the device he is using.

2-3- Javascript

JavaScript is a popular high-level programming language for web development. It was created by Netscape as a means to include dynamic and interactive components into websites.

JavaScript is largely influenced by the Java programming language, has a structure similar to C, and is based on ECMAScript, a Sun Microsystems computer language. It has gradually supplanted several other programming languages to become the industry standard for browser programming.

Brendan Eich, while working for Netscape in 1995, invented Java under the moniker Mocha, borrowing influence from Java, Scheme, and Self.

characteristics of javascript

- It is implemented on the client-side; for example, before sending the request to the server, you can validate any entries.
- It is simple to learn a language that is similar to English.
- It is a stand-alone programming language, not connected to Java as some people believe.
- More browser control services are available.
- Fast and interactive.
- It offers sophisticated interfaces, and you may drag and drop components to add the necessary pieces to your interface.
- A functional programming language.

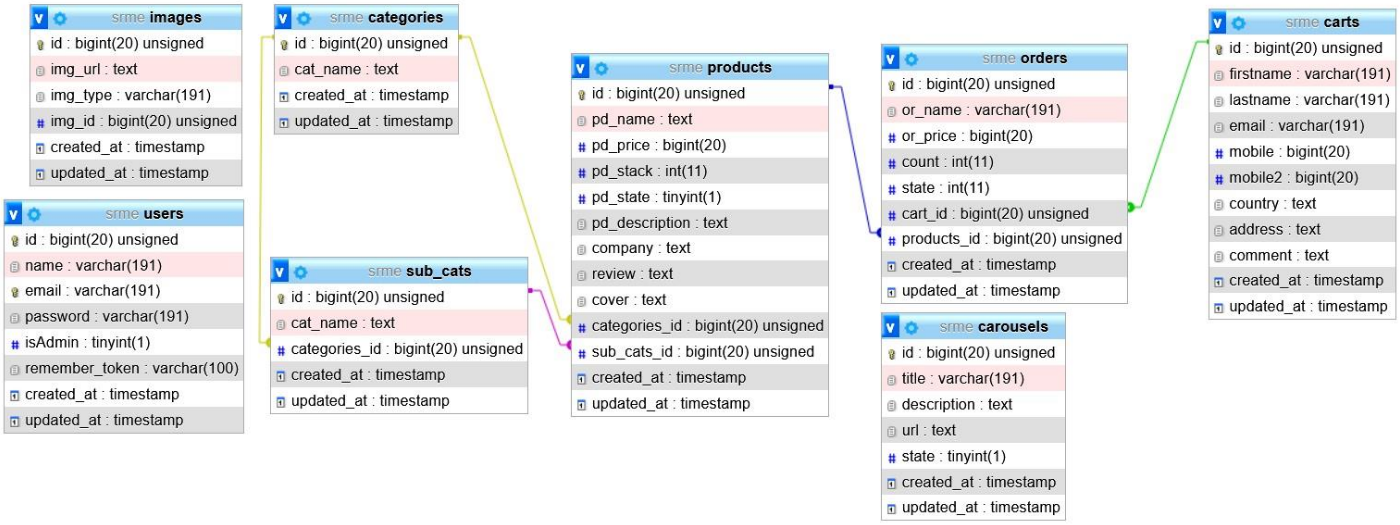
3- InertiaJS

Inertia is a toolkit that allows developers to design SPAs utilizing server-side routing and controllers, combining the benefits of both server-side rendering (SSR) and client-side rendering (CSR).

Developing web apps may be a difficult task. Before choosing from the various frameworks and libraries, consider if it will be a standard server-side rendered program (SSR) or a single page application (SPA). While both server-side and client-side rendering has advantages and disadvantages, Inertia brings the best of both worlds together in one package.

Database

MySQL databases were used, as we mentioned above, where they were created and the relationships between the tables were made, as shown in Figure (1).



srme images
- id : bigint(20) unsigned
- img_url : text
- img_type : varchar(191)
- img_id : bigint(20) unsigned
- created_at : timestamp
- updated_at : timestamp

srme categories
- id : bigint(20) unsigned
- cat_name : text
- created_at : timestamp
- updated_at : timestamp

srme products
- id : bigint(20) unsigned
- pd_name : text
- pd_price : bigint(20)
- pd_stack : int(11)
- pd_state : tinyint(1)
- pd_description : text
- company : text
- review : text
- cover : text
- categories_id : bigint(20) unsigned
- sub_cats_id : bigint(20) unsigned
- created_at : timestamp
- updated_at : timestamp

srme orders
- id : bigint(20) unsigned
- or_name : varchar(191)
- or_price : bigint(20)
- count : int(11)
- state : int(11)
- cart_id : bigint(20) unsigned
- products_id : bigint(20) unsigned
- created_at : timestamp
- updated_at : timestamp

srme carts
- id : bigint(20) unsigned
- firstname : varchar(191)
- lastname : varchar(191)
- email : varchar(191)
- mobile : bigint(20)
- mobile2 : bigint(20)
- country : text
- address : text
- comment : text
- created_at : timestamp
- updated_at : timestamp

srme users
- id : bigint(20) unsigned
- name : varchar(191)
- email : varchar(191)
- password : varchar(191)
- isAdmin : tinyint(1)
- remember_token : varchar(100)
- created_at : timestamp
- updated_at : timestamp

srme sub_cats
- id : bigint(20) unsigned
- cat_name : text
- categories_id : bigint(20) unsigned
- created_at : timestamp
- updated_at : timestamp

srme carousels
- id : bigint(20) unsigned
- title : varchar(191)
- description : text
- url : text
- state : tinyint(1)
- created_at : timestamp
- updated_at : timestamp

Figure (1) – Silk Road Middle East Database

Where a table has been created for the products, which contains columns, for example, the name of the commodity, its price, its description, product images, and the manufacturer.

Also, a table was made for the sub-categories and a table for the main categories to divide and arrange the products according to main and sub-categories to facilitate access to the products.

Also, a table has been created for the purchase basket in which all the goods and orders that have been booked in the application are stored, and which contains all the information of the buyer's customer to communicate with him to deliver the products he requested.

As well as a table for advertisements on the site on the main page, as well as a table for images and another table for the managers who manage the upload on the application.

Application interfaces and Code

The application mainly consists of two sets of pages: First, the user interfaces. Second, the control panel interfaces.

1- User Interface

It is the main interface of the application, in which all products are displayed divided, for example (new products, featured products).

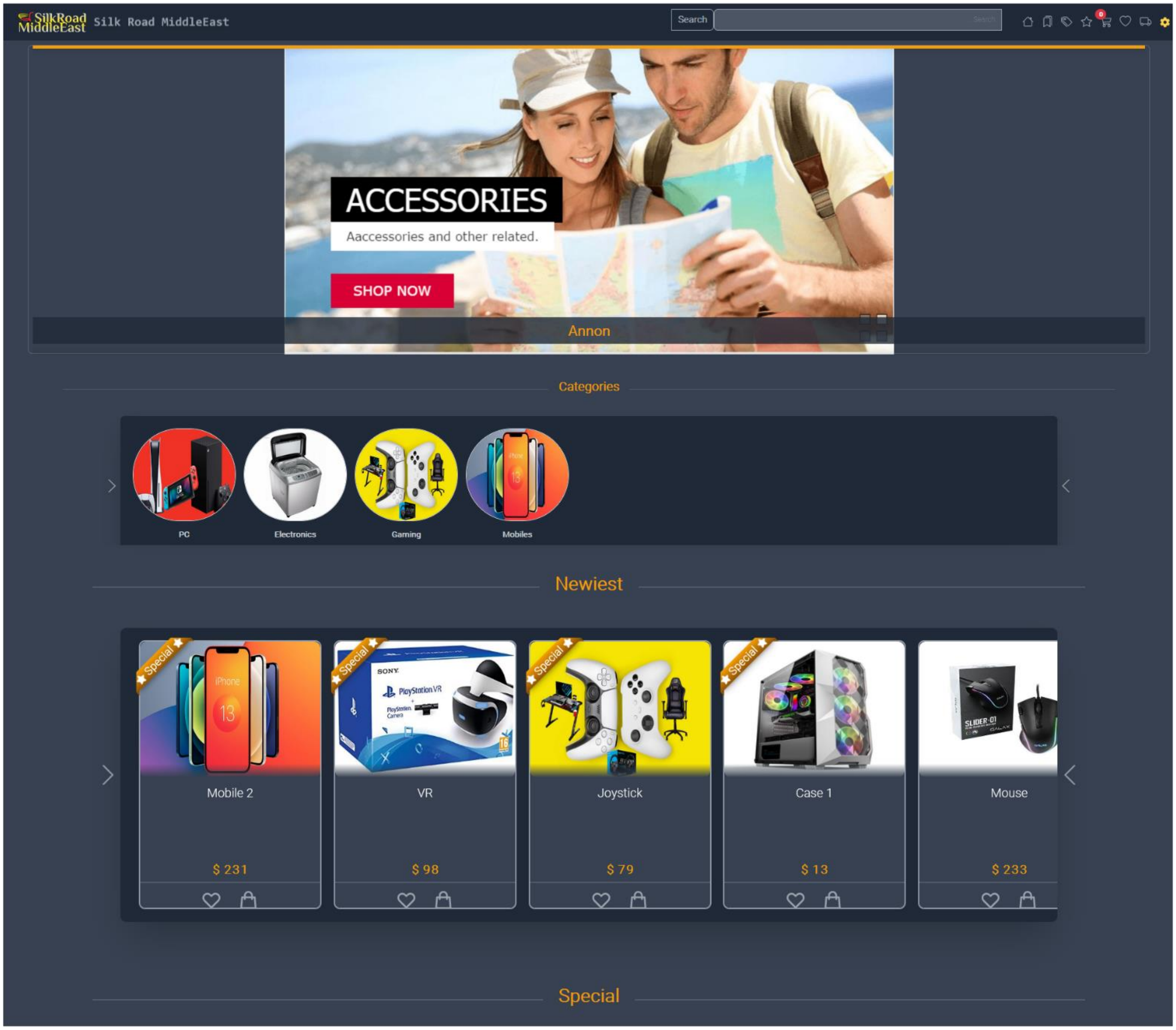
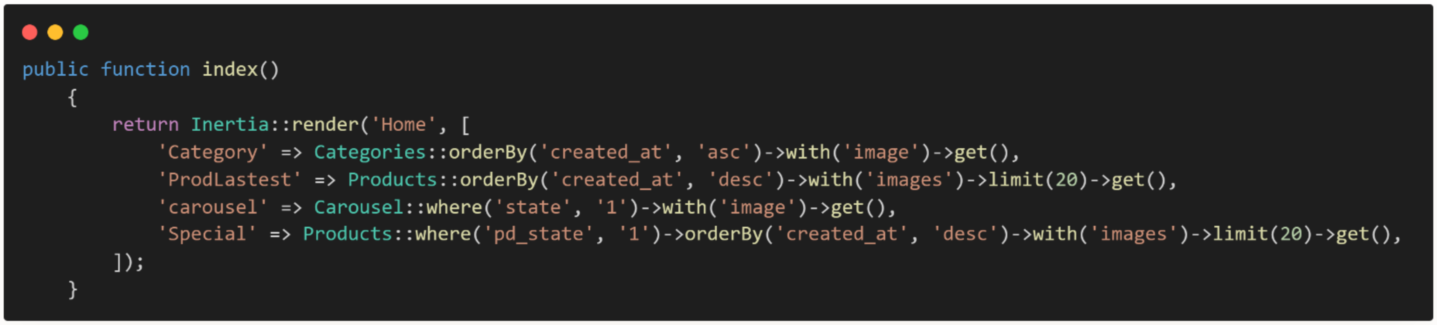


Figure (2) – User Interface (Home)

# Code User interface

By InertiaJS all the required parameters are sent to the main page with the name of the page specified within the application files for the interface



```php
public function index()
    {
        return Inertia::render('Home', [
            'Category' => Categories::orderBy('created_at', 'asc')->with('image')->get(),
            'ProdLastest' => Products::orderBy('created_at', 'desc')->with('images')->limit(20)->get(),
            'carousel' => Carousel::where('state', '1')->with('image')->get(),
            'Special' => Products::where('pd_state', '1')->orderBy('created_at', 'desc')->with('images')->limit(20)->get(),
        ]);
    }
```

Code (1) – User interface

1-1- Products Info Page

The product detail page contains all the information about the product in terms of product description or details, as well as price and images, as shown in Figure (3).

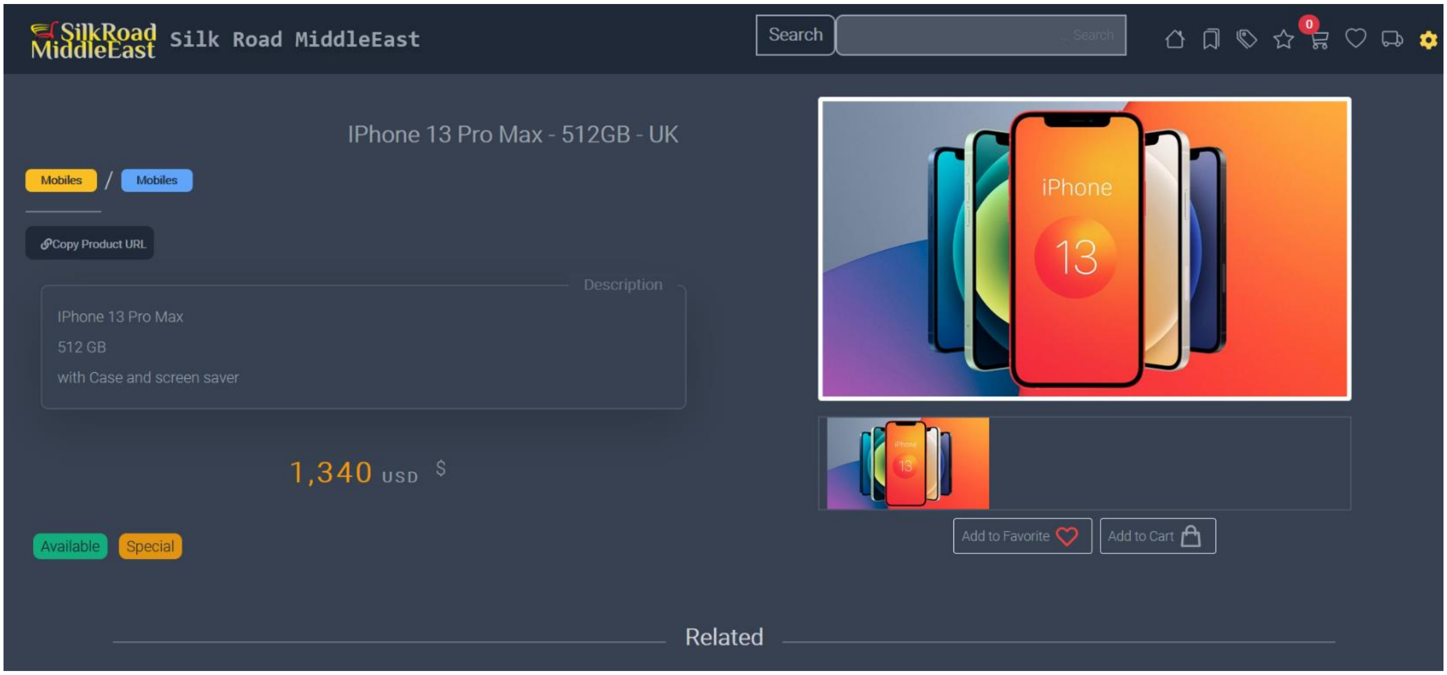
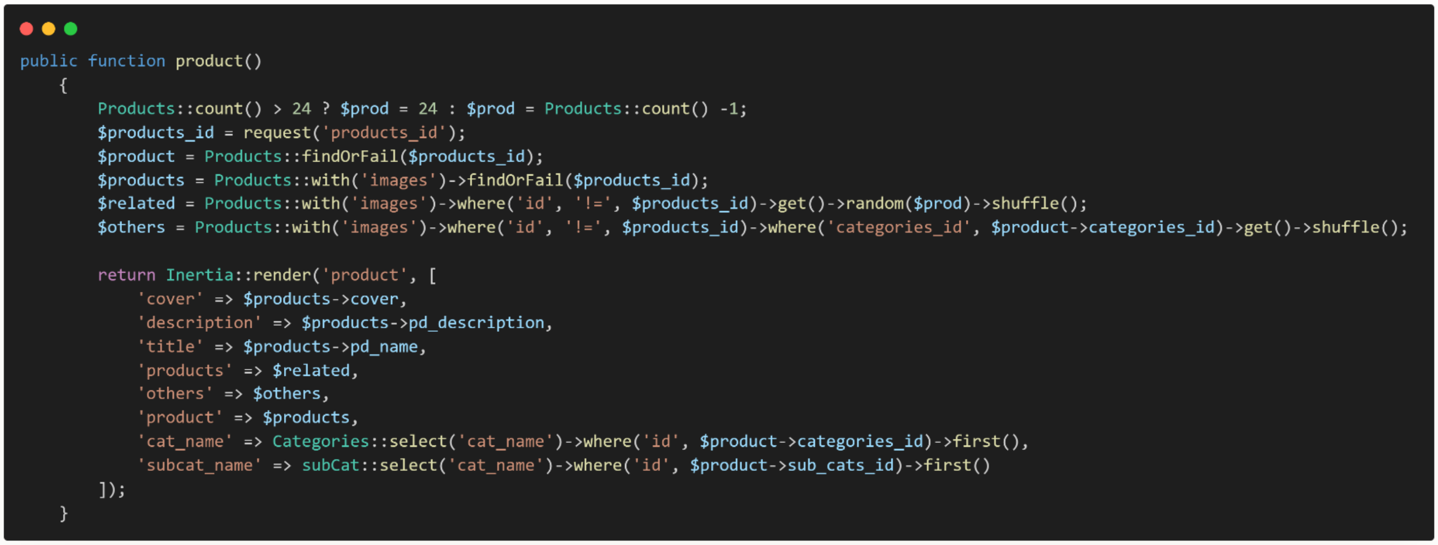


Figure (3) – Products info page

# Code Products info page

Also here the parameters to be displayed in the product information interface are sent using InertiaJS.



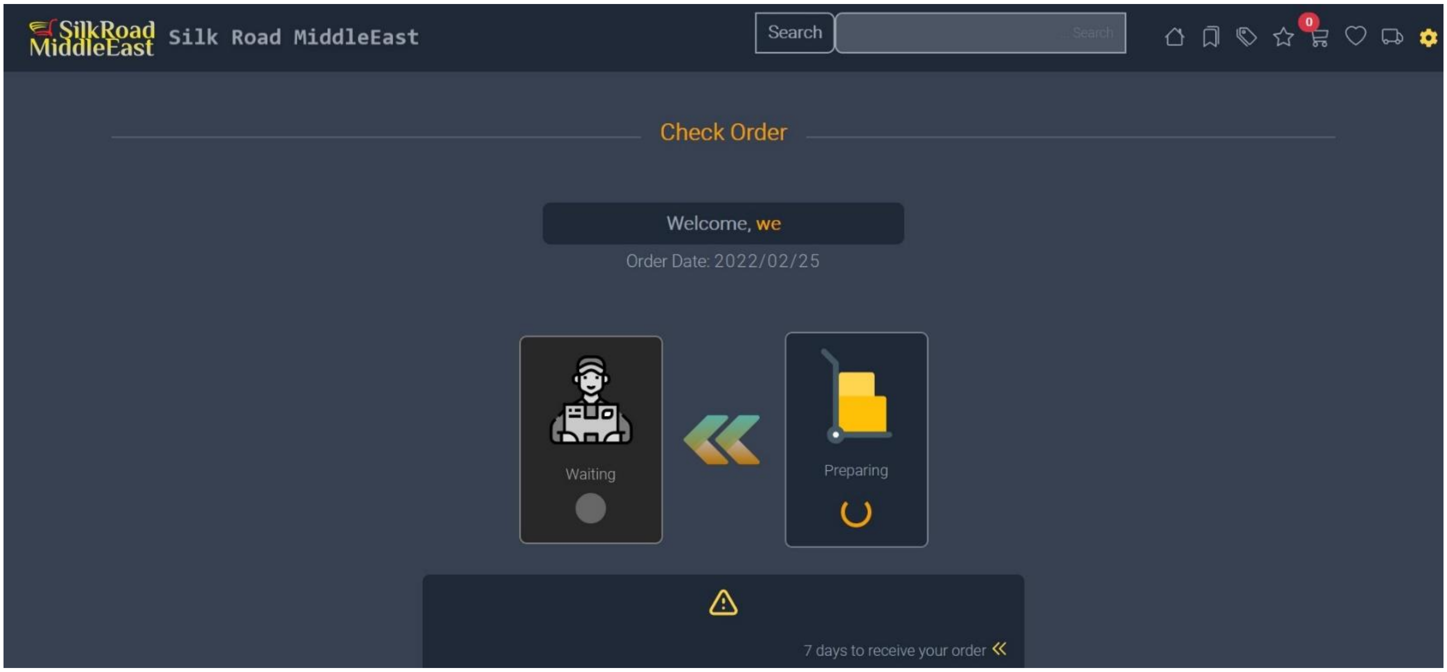
```
public function product()
    {
        Products::count() > 24 ? $prod = 24 : $prod = Products::count() -1;
        $products_id = request('products_id');
        $product = Products::findOrFail($products_id);
        $products = Products::with('images')->findOrFail($products_id);
        $related = Products::with('images')->where('id', '!=', $products_id)->get()->random($prod)->shuffle();
        $others = Products::with('images')->where('id', '!=', $products_id)->where('categories_id', $product->categories_id)->get()->shuffle();

        return Inertia::render('product', [
            'cover' => $products->cover,
            'description' => $products->pd_description,
            'title' => $products->pd_name,
            'products' => $related,
            'others' => $others,
            'product' => $products,
            'cat_name' => Categories::select('cat_name')->where('id', $product->categories_id)->first(),
            'subcat_name' => subCat::select('cat_name')->where('id', $product->sub_cats_id)->first()
        ]);
    }
```

Code (2) – products info page

1-2- Check order status

The page for entering the phone number that was entered within the order information is available to follow up on the status of the application at any stage as shown in Figure (4).



Check Order

Welcome, we

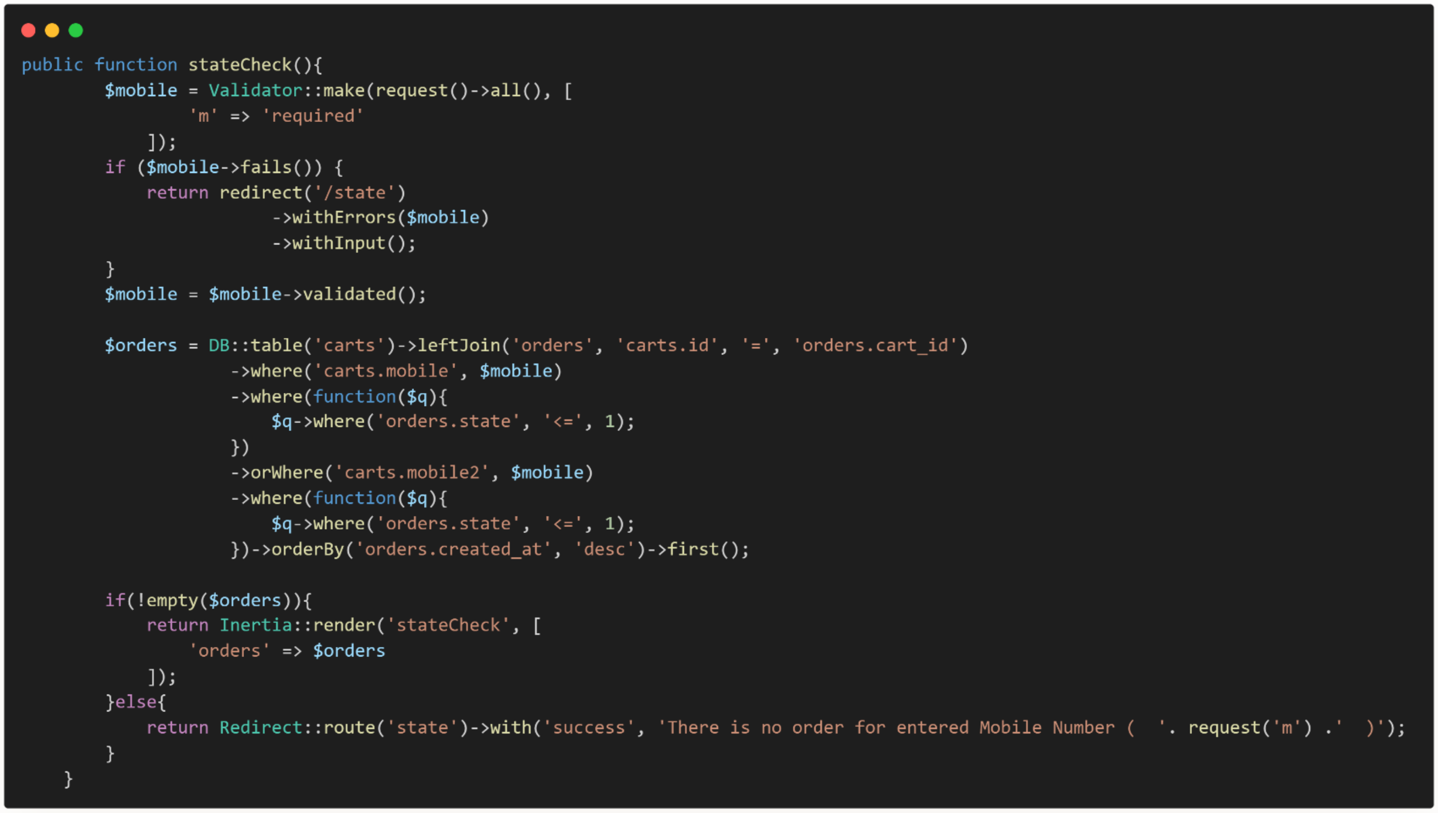
Order Date: 2022/02/25

7 days to receive your order

Figure (4) – Check order status

# Code Check order status

Here, the phone number is received from the interface page that the customer entered to check the status of his request, where the entered number will be searched for whether it is actually in the database, where the status of the request will be returned to him and at what stage it is. Otherwise, a message will be returned that the number does not exist in the database.



```
public function stateCheck(){
        $mobile = Validator::make(request()->all(), [
                'm' => 'required'
            ]);
        if ($mobile->fails()) {
            return redirect('/state')
                        ->withErrors($mobile)
                        ->withInput();
        }
        $mobile = $mobile->validated();

        $orders = DB::table('carts')->leftJoin('orders', 'carts.id', '=', 'orders.cart_id')
                    ->where('carts.mobile', $mobile)
                    ->where(function($q){
                        $q->where('orders.state', '<=', 1);
                    })
                    ->orWhere('carts.mobile2', $mobile)
                    ->where(function($q){
                        $q->where('orders.state', '<=', 1);
                    })->orderBy('orders.created_at', 'desc')->first();

        if(!empty($orders)){
            return Inertia::render('stateCheck', [
                'orders' => $orders
            ]);
        }else{
            return Redirect::route('state')->with('success', 'There is no order for entered Mobile Number (  '. request('m') .'  )');
        }
    }
```

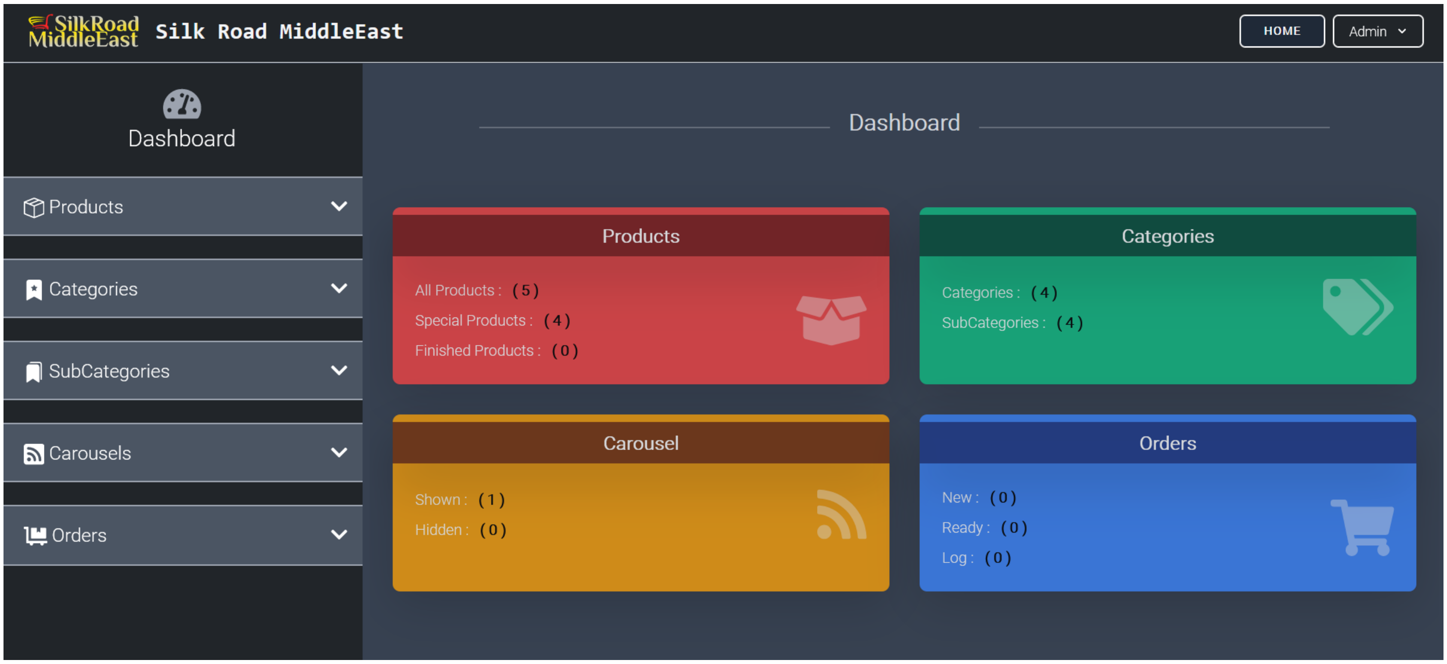
Code (3) – Check order status

2- Admin Interface

This page is for managers who manage the application in terms of adding, modifying, and deleting products, as well as advertisements and categories, as well as following up on the status of orders.

2-1- Dashboard

This page displays all the details for the preparation of products, main and sub categories, as well as the number of orders and the number of advertisements that have been added in the application, as well as access to each one of them, as shown in Figure (5).



Products

All Products : (5)
Special Products : (4)
Finished Products : (0)

Categories

Categories : (4)
SubCategories : (4)

Carousel

Shown : (1)
Hidden : (0)

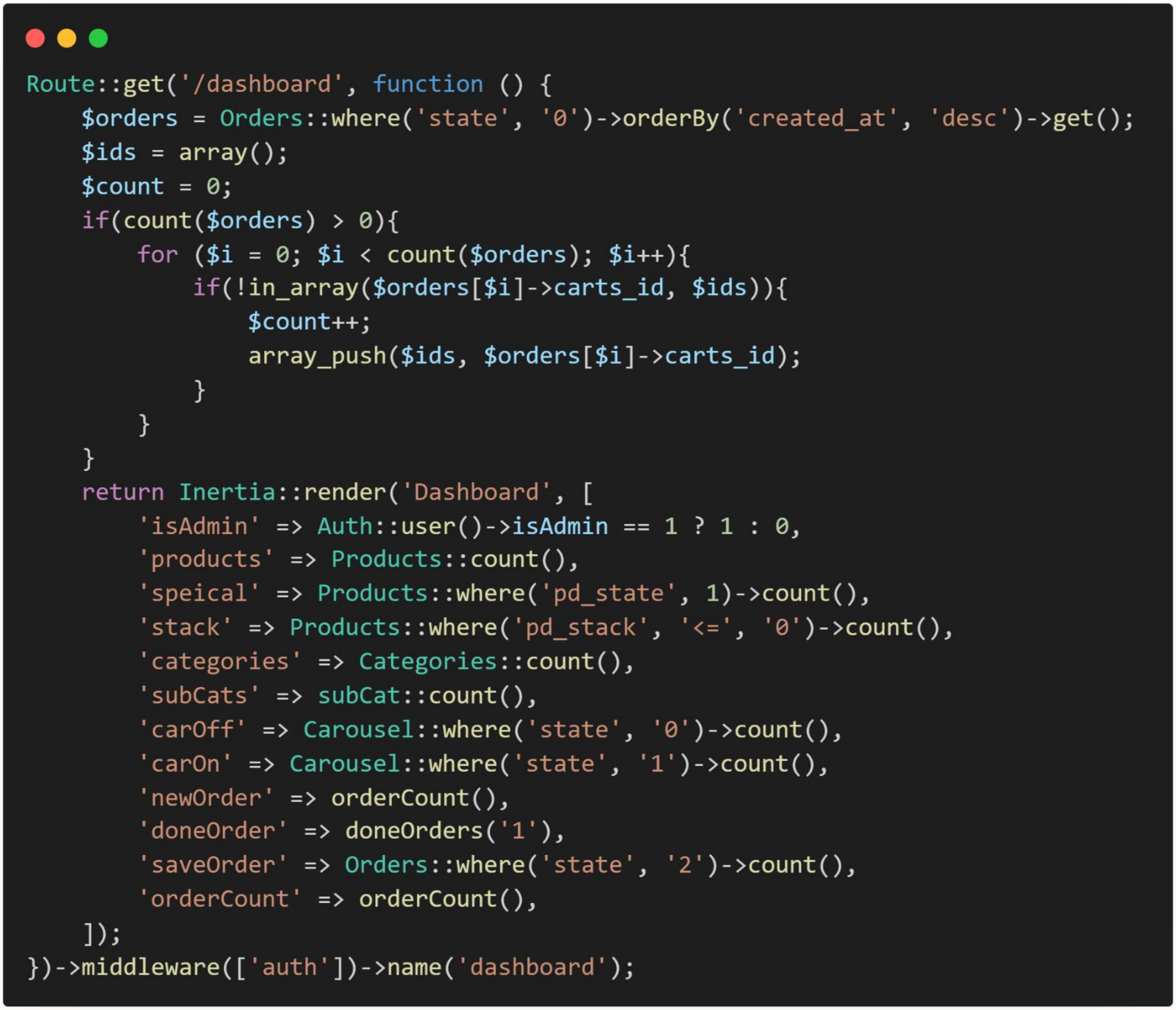
Orders

New : (0)
Ready : (0)
Log : (0)

Figure (5) – Dashboard page

# Code Dashboard page

On this page, the Dashboard link is generated, and the interface page is generated with all parameters being sent to the settings that appear on the page.

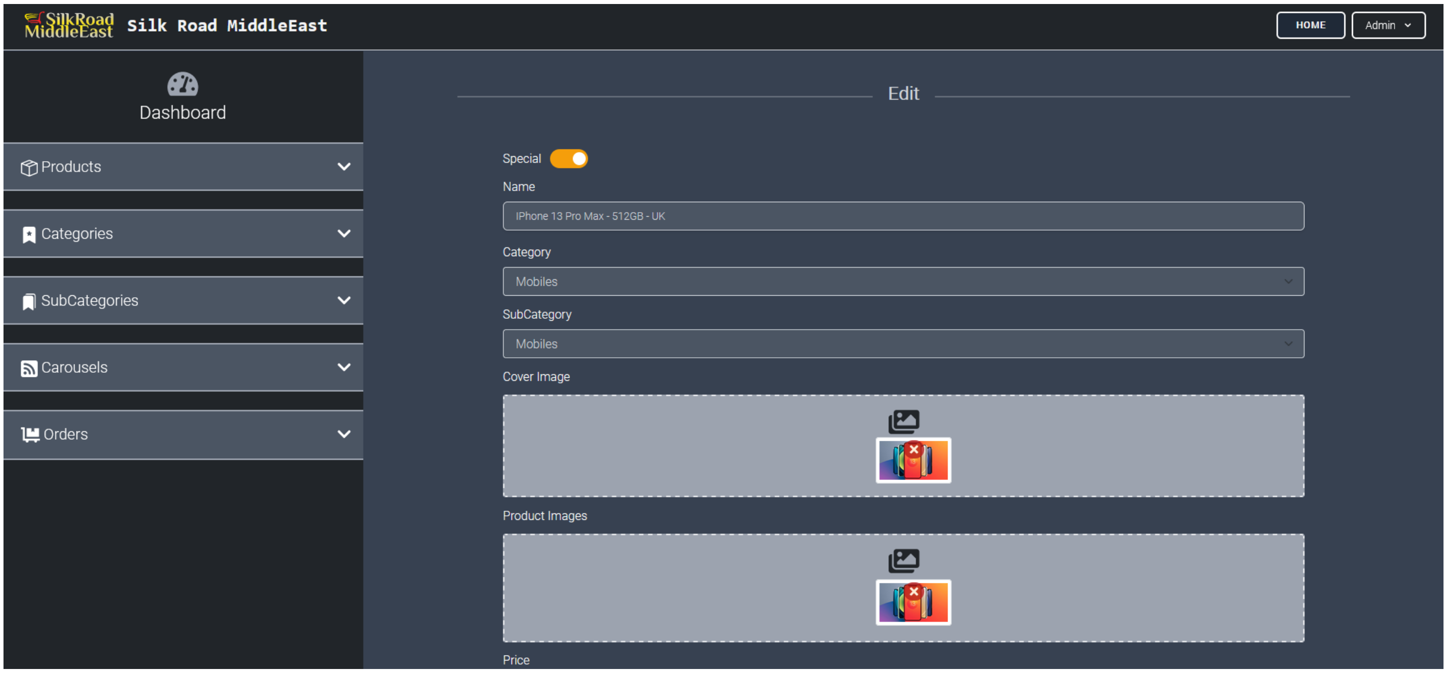


```php
Route::get('/dashboard', function () {
    $orders = Orders::where('state', '0')->orderBy('created_at', 'desc')->get();
    $ids = array();
    $count = 0;
    if(count($orders) > 0){
        for ($i = 0; $i < count($orders); $i++){
            if(!in_array($orders[$i]->carts_id, $ids)){
                $count++;
                array_push($ids, $orders[$i]->carts_id);
            }
        }
    }
    return Inertia::render('Dashboard', [
        'isAdmin' => Auth::user()->isAdmin == 1 ? 1 : 0,
        'products' => Products::count(),
        'speical' => Products::where('pd_state', 1)->count(),
        'stack' => Products::where('pd_stack', '<=', '0')->count(),
        'categories' => Categories::count(),
        'subCats' => subCat::count(),
        'carOff' => Carousel::where('state', '0')->count(),
        'carOn' => Carousel::where('state', '1')->count(),
        'newOrder' => orderCount(),
        'doneOrder' => doneOrders('1'),
        'saveOrder' => Orders::where('state', '2')->count(),
        'orderCount' => orderCount(),
    ]);
})->middleware(['auth'])->name('dashboard');
```

Code (4) – Dashboard page

2-2- Add new products page

On the Add a New Product page, the same is the case with the rest of the additions, such as categories and advertisements, where product information is entered in terms of name, price, pictures and the producing company. There is also a page similar to it to modify the product data entered on the add page, as shown in Figure (6).



Special ●

Name: iPhone 13 Pro Max - 512GB - UK

Category: Mobiles

SubCategory: Mobiles

Cover Image

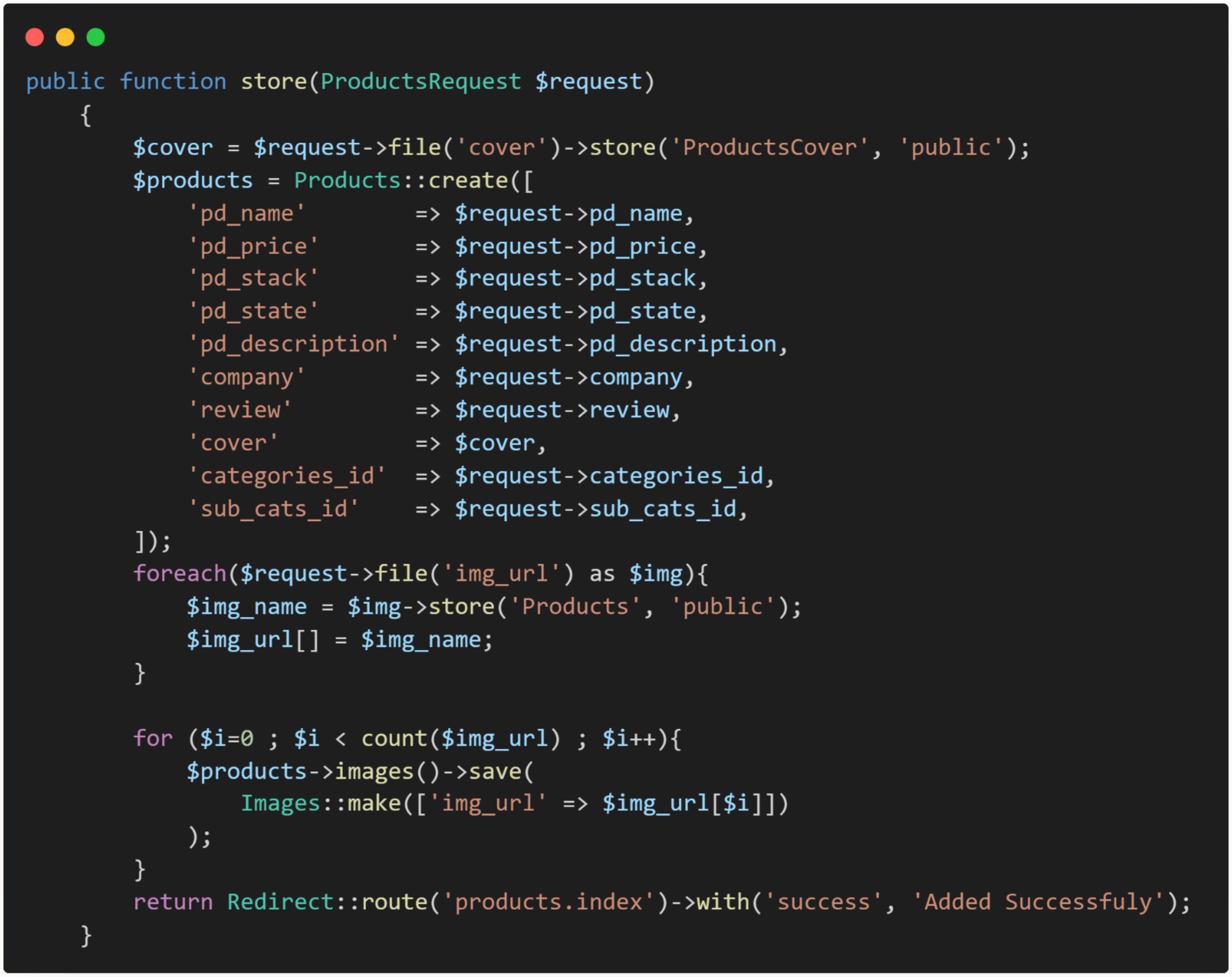
Product Images

Price

Figure (6) – Add new products page

# Code Add new products page

On this page, you receive from the interface designated to enter all the information about the product and then check and check if it is correct to be stored in the database.



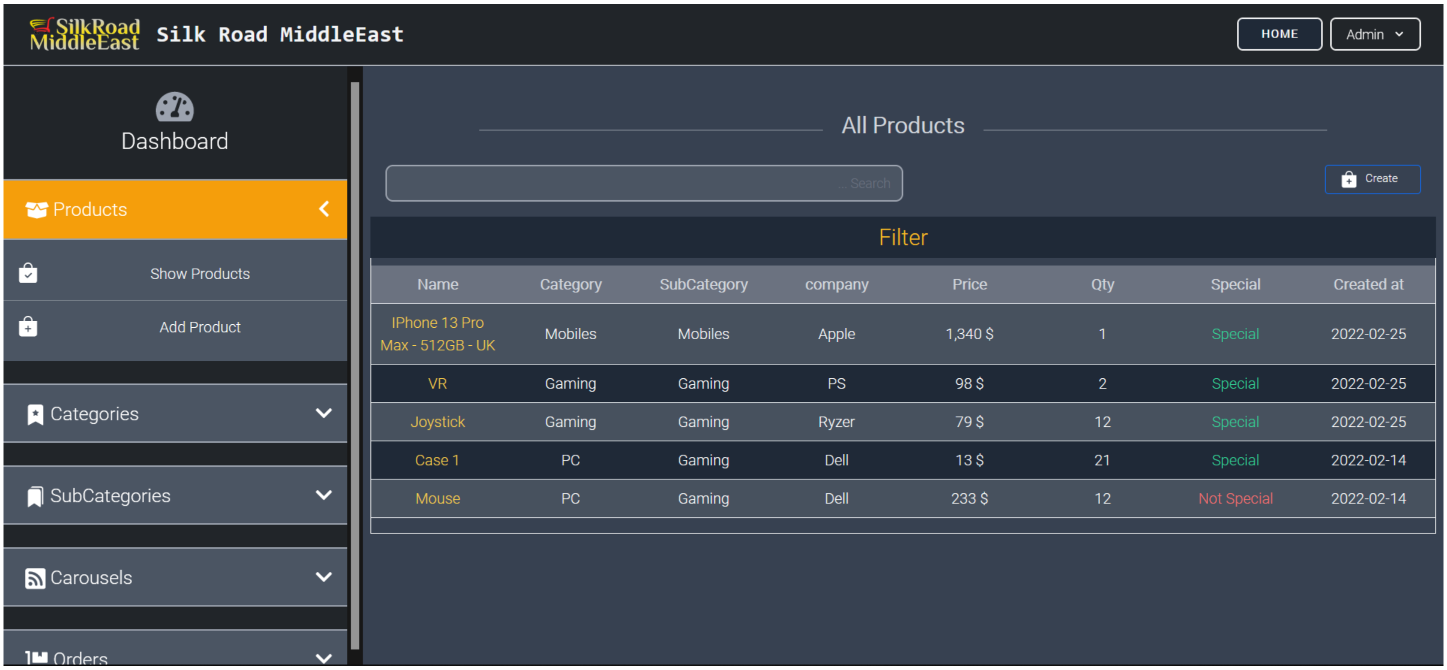
```
public function store(ProductsRequest $request)
    {
        $cover = $request->file('cover')->store('ProductsCover', 'public');
        $products = Products::create([
            'pd_name'         => $request->pd_name,
            'pd_price'        => $request->pd_price,
            'pd_stack'        => $request->pd_stack,
            'pd_state'        => $request->pd_state,
            'pd_description'  => $request->pd_description,
            'company'         => $request->company,
            'review'          => $request->review,
            'cover'           => $cover,
            'categories_id'   => $request->categories_id,
            'sub_cats_id'     => $request->sub_cats_id,
        ]);
        foreach($request->file('img_url') as $img){
            $img_name = $img->store('Products', 'public');
            $img_url[] = $img_name;
        }

        for ($i=0 ; $i < count($img_url) ; $i++){
            $products->images()->save(
                Images::make(['img_url' => $img_url[$i]])
            );
        }
        return Redirect::route('products.index')->with('success', 'Added Successfuly');
    }
```

Code (5) – Add new products page

2-3- Items page

On this page, all the goods that have been added, as well as some details for each product are displayed in a table. You can click on any product and enter a page to modify the product data, as shown in Figure (7).

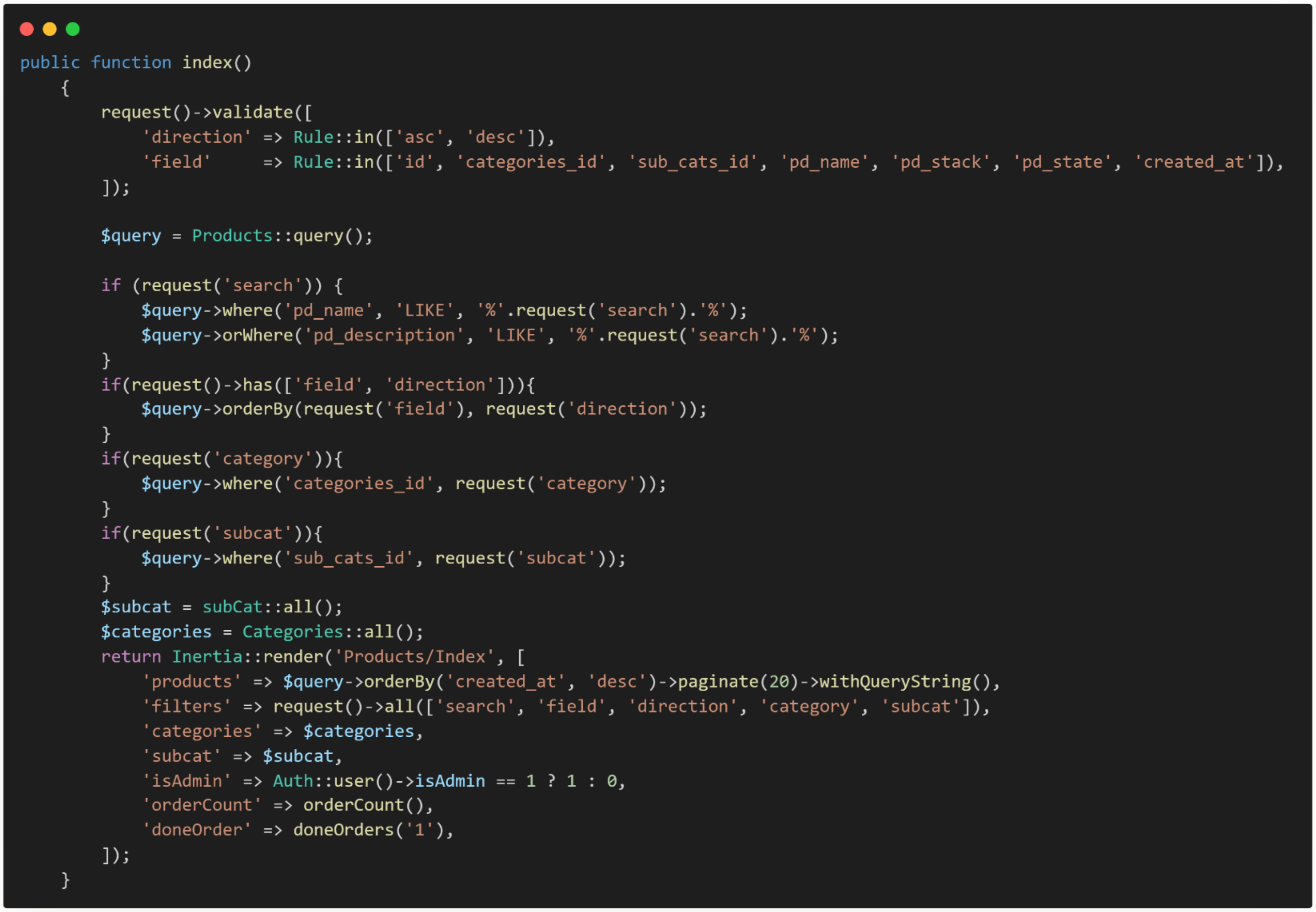


Name	Category	SubCategory	company	Price	Qty	Special	Created at
IPhone 13 Pro Max - 512GB - UK	Mobiles	Mobiles	Apple	1,340 $	1	Special	2022-02-25
VR	Gaming	Gaming	PS	98 $	2	Special	2022-02-25
Joystick	Gaming	Gaming	Ryzer	79 $	12	Special	2022-02-25
Case 1	PC	Gaming	Dell	13 $	21	Special	2022-02-14
Mouse	PC	Gaming	Dell	233 $	12	Not Special	2022-02-14

Figure (7) – items page

# Code Items page

On this page the interface page is generated with the table information to be displayed. Also here on this page contains a filter for the data in the table to arrange the data in the table or search for a specific value in the table to return it also in this code.



```php
public function index()
    {
        request()->validate([
            'direction' => Rule::in(['asc', 'desc']),
            'field'     => Rule::in(['id', 'categories_id', 'sub_cats_id', 'pd_name', 'pd_stack', 'pd_state', 'created_at']),
        ]);

        $query = Products::query();

        if (request('search')) {
            $query->where('pd_name', 'LIKE', '%'.request('search').'%');
            $query->orWhere('pd_description', 'LIKE', '%'.request('search').'%');
        }
        if(request()->has(['field', 'direction'])){
            $query->orderBy(request('field'), request('direction'));
        }
        if(request('category')){
            $query->where('categories_id', request('category'));
        }
        if(request('subcat')){
            $query->where('sub_cats_id', request('subcat'));
        }
        $subcat = subCat::all();
        $categories = Categories::all();
        return Inertia::render('Products/Index', [
            'products' => $query->orderBy('created_at', 'desc')->paginate(20)->withQueryString(),
            'filters' => request()->all(['search', 'field', 'direction', 'category', 'subcat']),
            'categories' => $categories,
            'subcat' => $subcat,
            'isAdmin' => Auth::user()->isAdmin == 1 ? 1 : 0,
            'orderCount' => orderCount(),
            'doneOrder' => doneOrders('1'),
        ]);
    }
```

Code (6) – Items page