

Trường cao đẳng công nghệ Thủ Đức

Khoa công nghệ thông tin

-----&  &-----



Báo cáo đồ án back end 2

Shop bán hàng “Luxury shop”

Nhóm 7:

1. Lê Trung Hiếu – 19211TT0370 (Nhóm trưởng)
2. Cao Trung Hiếu – 19211TT0067
3. Cao Minh Hiếu – 19211TT2241
4. Trần Hữu Phúc – 19211TT0761

Lớp học phần: CNC10745202

Giảng viên hướng dẫn: Bùi Thị Phương Thảo

TP. HỒ CHÍ MINH – NĂM 2021

Mục lục

Chương 1 - Tổng quan về Framework Laravel	1
1. Laravel là gì? Tổng quan về Laravel php framework.	1
2. Mô hình MVC (Model – View – Controller) của Laravel:	2
3. Cấu trúc các thư mục của Laravel	3
Chương 2: Giới thiệu về website “Luxury shop”	7
1. Giới thiệu:.....	7
.1.1 Mô tả:.....	7
.1.2 Các loại tài khoản và quyền tương ứng:.....	7
.1.3 Header trang chủ.....	7
.1.4 Body trang chủ (Tag Hot Product)	8
.1.5 Body Trang chủ (Tag All)	10
.1.6 Footer Trang chủ	11
.1.7 Chi tiết cách Controller truyền biến xuống View.....	12
2. Sơ đồ use case website	13
.2.1 Sơ đồ use case.....	13
.2.2 Các chức năng.....	13
3. Mô tả cơ sở dữ liệu	15
.3.1 Bảng tbl_product	15
.3.2 Bảng tbl_brand	16
.3.3 Bảng tbl_category_product	16
.3.4 Bảng tbl_customer	16
.3.5 Bảng tbl_order	17
.3.6 Bảng tbl_order_details.....	17
.3.7 Bảng tbl_shipping.....	17
.3.8 Bảng tbl_payment.....	18
.3.9 Bảng tbl_comment.....	18
.3.10 Bảng tbl_admin	18
.3.11 Bảng tbl_author	18
4. Chức năng người dùng khách hàng	19
.4.1 Nhận biết người dùng khách hàng đăng nhập hay chưa.....	19
.4.2 Xem danh sách các sản phẩm nổi bật	20
.4.3 Xem danh sách tất cả sản phẩm.....	20

.4.4	Xem danh sách sản phẩm theo thương hiệu (Brand).....	21
.4.5	Xem danh sách sản phẩm loại sản phẩm (Category).....	23
.4.6	Tìm kiếm sản phẩm theo từ khoá (Keyword).....	25
.4.7	Xem chi tiết sản phẩm	27
.4.8	Comment và đánh giá sản phẩm.....	28
.4.9	Giỏ hàng	31
.4.10	Chức năng Order	34
5.	Chức Năng Đăng nhập/Dăng Ký/Dăng xuất	40
.5.1	Chức năng đăng nhập (Sign in)	40
.5.2	Chức năng đăng ký (Register)	41
.5.3	Chức năng đăng xuất (Sign out)	44
6.	Chức Năng Quản Trị	45
.6.1	Giới thiệu	45
.6.2	Đăng nhập	45
.6.3	Giao diện chính trang quản trị	47
.6.4	Quản trị đơn hàng	47
.6.5	Quản trị Caterogy	54
.6.6	Quản trị Brand	62
.6.7	Quản trị Product.....	69
	CHƯƠNG 3: TỔNG KẾT	81
1.	Phân công thành viên	81
2.	Thuận lợi khó khăn.....	83
.2.1	Khó khăn.....	83
.2.2	Thuận lợi.....	83
3.	Kết quả đạt được	84
4.	Đánh giá thành viên.....	84
	TÀI LIỆU THAM KHẢO	85

Hình 4: Cấu trúc Laravel-----	3
Hình 5: Thư mục app -----	4
Hình 6: Thư mục bootstrap -----	5
Hình 7: Thư mục database -----	5
Hình 8: Thư mục resources -----	5
Hình 9: Thư mục storage-----	6
Hình 10: Thư mục tests -----	6
Controller 2: Index.....	12
Controller 3: Điều kiện người đăng nhập.....	19
Controller 4: Hiển thị thương thiệu	21
Controller 5:Loại sản phẩm	23
Controller 6: Tìm kiếm sản phẩm.....	26
Controller 7: Chi tiết sản phẩm	27
Controller 8: Hàm tính trung bình sao.....	29
Controller 9: Thêm đánh giá sản phẩm	30
Controller 10: Xoá đánh giá sản phẩm.....	30
Controller 11: Lưu giỏ hàng	32
Controller 12: Hiển thị giỏ hàng.....	33
Controller 13: Xoá sản phẩm và cập nhập số lượng giỏ hàng.....	33
Controller 14: Check out	35
Controller 15: Lưu thông tin người dùng	35
Controller 16: Thanh toán	36
Controller 17: order	37
Controller 18: Đăng nhập	41
Controller 19: Đăng kí.....	43
Controller 20: Đăng xuất	44
Controller 21: Đăng nhập admin	45
Controller 22: Quản trị đơn hàng	48
Controller 23: Tìm kiếm đơn hàng	49
Controller 24: Cập nhập đơn hàng	50
Controller 25: Xoá đơn hàng	51

Controller 26: Kiểm tra ảnh.....	55
Controller 27: Thêm loại sản phẩm	56
Controller 28: Lấy loại sản phẩm	57
Controller 29: Tìm kiếm loại sản phẩm.....	58
Controller 30: Xoá loại sản phẩm.....	59
Controller 31: Cập nhập loại sản phẩm	61
Controller 32: Kiểm tra hình ảnh.....	63
Controller 33: Thêm thương hiệu	63
Controller 34: Lấy thương hiệu	64
Controller 35: Tìm kiếm thương hiệu.....	65
Controller 36: Xoá thương hiệu.....	66
Controller 37: Cập nhập thương hiệu	68
Controller 38: Kiểm tra hình ảnh.....	70
Controller 39: Thêm sản phẩm	70
Controller 40: Lấy sản phẩm	71
Controller 41: Tìm kiếm sản phẩm.....	72
Controller 42: Xoá sản phẩm.....	73
Controller 43: Cập nhập sản phẩm	75
Router 1: Index	12
Router 2: Đường dẫn category	22
Router 3: Loại sản phẩm.....	24
Router 4: Tìm kiếm sản phẩm	26
Router 5: Chi tiết sản phẩm	27
Router 6: Thêm đánh giá sản phẩm	30
Router 7: Xoá đánh giá sản phẩm.....	30
Router 8: Giỏ hàng	33
Router 9: Check out.....	35
Router 10: Thanh toán	36
Router 11: Đăng nhập.....	41
Router 12: Đăng ký	43
Router 13: Đăng xuất.....	44

Router 14: Đăng nhập admin.....	46
Router 15: Quản trị đơn hàng	52
Router 16: Thêm loại sản phẩm.....	56
Router 17: Lấy loại sản phẩm.....	58
Router 18: Tìm kiếm loại sản phẩm	58
Router 19: Xoá loại sản phẩm	59
Router 20: Cập nhập loại sản phẩm.....	61
Router 21: Thêm thương hiệu	63
Router 22: Lấy thương hiệu.....	65
Router 23: Tìm kiếm thương hiệu	65
Router 24: Xoá thương hiệu	66
Router 25: Cập nhập thương hiệu.....	68
Router 26: Thêm sản phẩm.....	70
Router 27: Lấy sản phẩm.....	72
Router 28: Tìm kiếm sản phẩm	73
Router 29: Xoá sản phẩm	73
Router 30: Cập nhập sản phẩm.....	75
 View 1: Điều kiện đăng nhập	19
View 2: Điều kiện đăng nhập hoặc đăng xuất.....	20
View 3: category.....	22
View 4: Loại sản phẩm.....	24
View 5: Tìm kiếm sản phẩm	25
View 6: Tìm kiếm sản phẩm	26
View 7: Chi tiết sản phẩm	28
View 8: Check out	38
View 9: Thanh toán	38
View 10: Hoàn thành đặt hàng	39
View 11: Đăng nhập admin	46
View 12: Quản trị đơn hàng	52
View 13: Thông tin khách hàng	53
View 14: Thông tin giao hàng	53

View 15: Chi tiết đơn hàng.....	54
View 16: Thêm loại sản phẩm.....	56
View 17: Kiểm tra trạng thái loại sản phẩm.....	57
View 18: Tìm kiếm loại sản phẩm	59
View 19: Xoá loại sản phẩm	59
View 20: Cập nhập loại sản phẩm	61
View 21: Thêm thương hiệu.....	63
View 22: Trạng thái thương hiệu	64
View 23: Tìm kiếm thương hiệu	66
View 24: Xoá thương hiệu	67
View 25: Cập nhập thương hiệu.....	68
View 26: Trạng thái sản phẩm.....	71
View 27: Xoá sản phẩm	73
View 28: Cập nhập sản phẩm 1	76
View 29: Cập nhập sản phẩm 2	76
Query 1: Lấy sản phẩm và thương hiệu	7
Query 2: Lấy dữ liệu sản phẩm và thương hiệu	7
Query 3: Lấy sản phẩm hot	8
Query 4: Lấy tất cả sản phẩm	10
Query 5: Lấy dữ liệu author	11
Query 6: Lấy 8 sản phẩm.....	20
Query 7: Lấy tất cả sản phẩm	20
Query 8: Đánh giá sản phẩm phân trang	29

Chương 1 - Tổng quan về Framework Laravel

1. Laravel là gì? Tổng quan về Laravel php framework.



Hình 1: Laravel

Laravel là PHP Web Framework miễn phí, mã nguồn mở, được tạo bởi Taylor Otwell và dành cho việc phát triển các ứng dụng web theo mô hình kiến trúc mô hình MVC và dựa trên Symfony PHP Framework. Một số tính năng của Laravel như là sử dụng hệ thống đóng gói module, quản lý package (Composer), hỗ trợ nhiều hệ quản trị CSDL quan hệ (MySQL, MariaDB, SQLite, PostgreSQL,...), các tiện ích hỗ trợ triển khai và bảo trì ứng dụng.

Mã nguồn của Laravel được lưu trữ trên GitHub và được cấp phép theo các điều khoản của Giấy phép MIT:

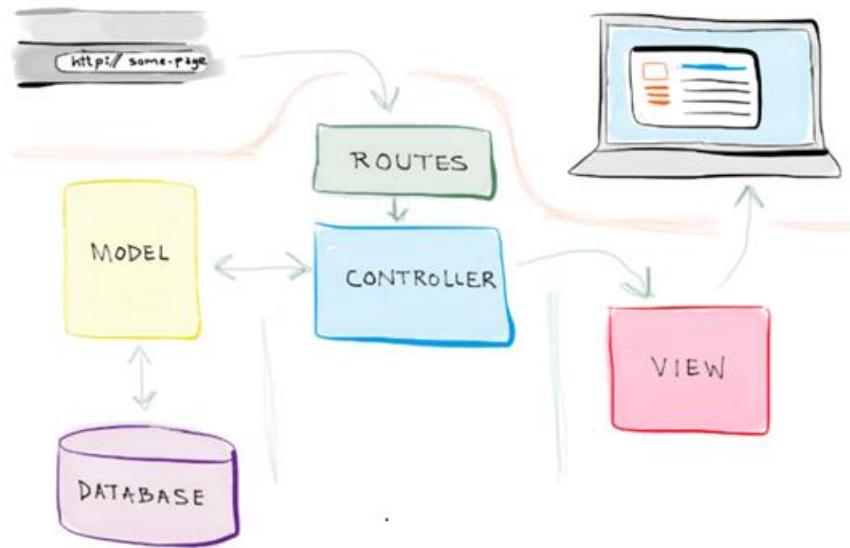
- Developer(s): Taylor Otwell
- Phiên bản hiện tại: 7.12.5
- Repository: Laravel Repository
- Ngôn ngữ lập trình: PHP
- Thể loại: Web framework

❖ **ƯU ĐIỂM:**

- ✓ Sử dụng các tính năng (feature) mới nhất của PHP
- ✓ Sử dụng mô hình MVC (Model – View – Controller)
- ✓ Hệ thống xác thực (Authentication) và ủy quyền (Authorization) tuyệt vời
- ✓ Có hệ thống tài liệu chi tiết, dễ hiểu
- ✓ Cộng đồng hỗ trợ đông đảo
- ✓ Tích hợp công cụ Artisan – Công cụ quản lý dòng lệnh
- ✓ Sử dụng composer để quản lý PHP package
- ✓ Sử dụng npm để quản lý các gói Javascript, giao diện
- ✓ Hỗ trợ Eloquent ORM, Query Builder, Template Engine (Blade)
- ✓ Hỗ trợ routing mềm dẻo

2. Mô hình MVC (Model – View – Controller) của Laravel:

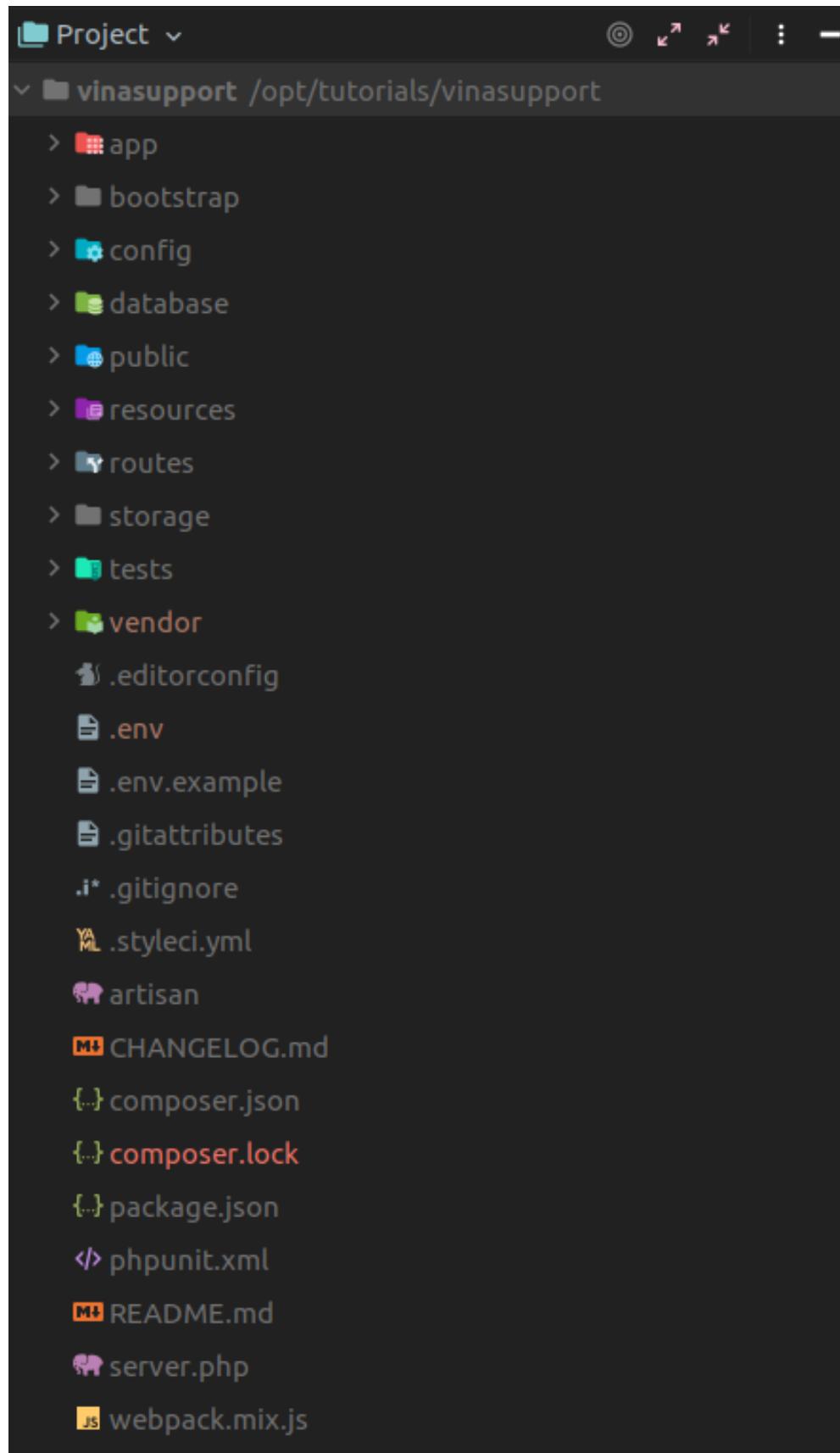
Laravel được phát triển từ Symfony Framework nên nó kế thừa mô hình MVC của Symfony, tương tự như các PHP Framework khác. Dưới đây là mô hình đầy đủ của Laravel:



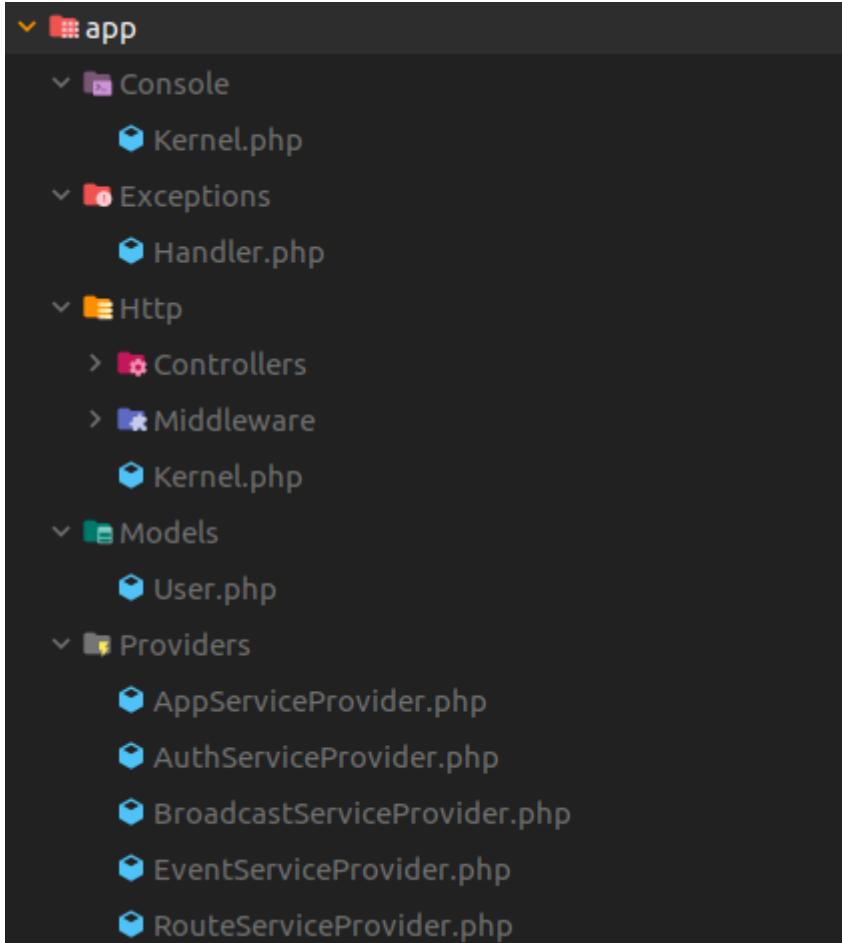
Hình 2: Mô hình MVC

Thành phần	Mô tả
Model	Gồm những Class, tạo ra các table của Database, nơi mà Eloquent ORM thao tác với CSDL.
View	Chứa các template được thiết kế ra và được xử lý để output ra mã html/css cho trang web.
Controller	Chứa các class, function để xử lý các request từ người dùng.

3. Cấu trúc các thư mục của Laravel

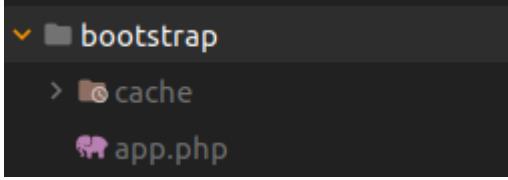
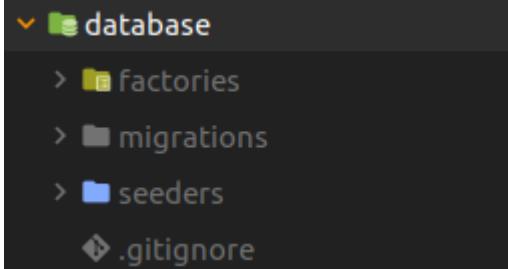
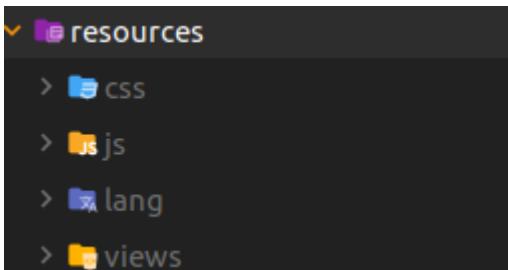


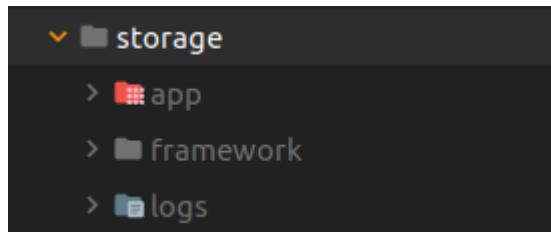
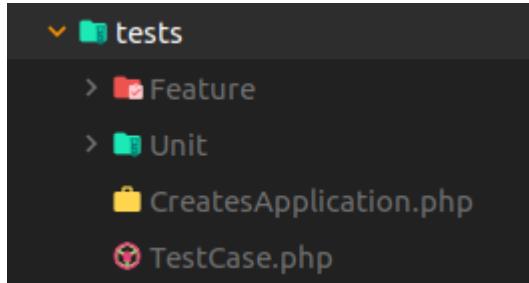
Hình 1: Cấu trúc Laravel

Thành phần	Mô tả
App	<p>Là thư mục ứng dụng và bao gồm toàn bộ mã nguồn của dự án. Nó chứa các sự kiện (Event), ngoại lệ (Exception) và khai báo Middleware.</p> <p>Trong App lại chứa các thành phần con sau:</p>  <pre> app +-- Console +-- Kernel.php +-- Exceptions +-- Handler.php +-- Http +-- Controllers +-- Middleware +-- Kernel.php +-- Models +-- User.php +-- Providers +-- AppServiceProvider.php +-- AuthServiceProvider.php +-- BroadcastServiceProvider.php +-- EventServiceProvider.php +-- RouteServiceProvider.php </pre>

Hình 2: Thư mục app

- **Console:** Nơi bạn định nghĩa ra các lệnh của Laravel.
- **Exceptions:** Thư mục này chứa tất cả các phương thức cần thiết để xử lý các ngoại lệ. Nó cũng chứa tệp handle.php xử lý tất cả các trường hợp ngoại lệ.
- **Http:** Thư mục Http này chứa các thư mục con controllers, middleware. Vì Laravel tuân theo mô hình MVC nên thư mục này chứa các xử lý để xử lý request của người dùng.
- **Models:** Chứa các Class định nghĩa các thành phần để thao tác với CSDL.
- **Providers:** Chứa tất cả các Service Provider. Khởi động ứng dụng một cách bắt buộc, đăng ký sự kiện hoặc thực hiện bất kỳ tác vụ nào khác để chuẩn bị ứng dụng của bạn khi các yêu cầu đến.

Bootstrap	Thư mục bootstrap chứa tệp app.php khởi động khung. Thư mục này cũng chứa một thư mục bộ đệm chứa các tệp được tạo khung để tối ưu hóa hiệu suất, chẳng hạn như tệp bộ đệm của tuyến đường và dịch vụ. Bạn thường không cần phải sửa đổi bất kỳ tệp nào trong thư mục này.
	 <p>Hình 3: Thư mục bootstrap</p>
Config	Thư mục này chứa tất cả cấu hình (config) của hệ thống của bạn. Nó chứa sẵn các thông tin mặc định như cấu hình database, cache, ...
Database	Thư mục này chứa các thông tin về database migration, seeders.
	 <p>Hình 4: Thư mục database</p>
Public	Thư mục public chứa file index.php, là nơi nhận tất cả các request tới ứng dụng. Thư mục này cũng chứa các nội dung của bạn như hình ảnh, JavaScript và CSS.
Resources	Thư mục tài nguyên chứa các view template của bạn cũng như các nội dung thô (raw), chưa được biên dịch của bạn như CSS hoặc JavaScript. Thư mục này cũng chứa tất cả các tệp ngôn ngữ của bạn.
	 <p>Hình 5: Thư mục resources</p>
Routes	Thư mục routes chứa tất cả các định nghĩa routing cho ứng dụng của bạn. Mặc định, một số router được định nghĩa sẵn bao gồm: web.php, api.php, console.php và channel.php.

Storage	Thư mục storage chứa các thông tin về log, các views blade đã biên dịch, các file session, file cache và các file khác được tạo bởi framework.
	 <p>Hình 6: Thư mục storage</p>
Tests	Thư mục tests chứa các kiểm tra tự động (Automated Tests). Các mẫu PHPUnit.
	 <p>Hình 7: Thư mục tests</p>
Vendor	Thư mục vendor chứa các thư viện PHP phụ thuộc được quản lý bởi Composer.

Chương 2: Giới thiệu về website “Luxury shop”

1. Giới thiệu:

.1.1 Mô tả:

Xây dựng một Website bán quần áo, cũng như là quản lý các sản phẩm trong cửa hàng

- Người dùng được chia làm 3 loại người dùng
 - Người dùng chưa đăng nhập
 - Người dùng khách
 - Người dùng quản trị

.1.2 Các loại tài khoản và quyền tương ứng:

- **Tài khoản quản trị:** Người dùng có thể quản lý trang web ví dụ như chỉnh sửa số lượng sản phẩm, thay đổi hoặc thêm các mẫu sản phẩm mới, thêm hãng hoặc loại sản phẩm, v.v....
- **Tài khoản khách hàng:** Người dùng có thể xem các mẫu sản phẩm theo mặc định hoặc theo loại/hãng sản phẩm, thêm sản phẩm vào giỏ hàng, quản trị giỏ hàng, đặt hàng, bình luận, đánh giá sản phẩm
- **Người dùng không đăng nhập:** Người dùng chỉ được xem danh sách sản phẩm, chi tiết sản phẩm

.1.3 Header trang chủ

- Phía trên của trang chủ sẽ là nơi để Logo, Giỏ hàng Mini, Text Field để tìm kiếm sản phẩm
- Nút home sẽ đưa người dùng trở về trang chủ
- Nút Brand khi nhấn vào sẽ xổ ra hàng loạt các nhà sản xuất, người dùng có thể nhấp vào 1 nhà sản xuất để lọc sản phẩm theo nhà sản xuất đó
- Nút Category khi nhấn vào sẽ xổ ra hàng loạt các loại sản phẩm (Áo, Quần, Giày, Dép, vv...), người dùng có thể nhấp vào 1 loại sản phẩm để lọc sản phẩm theo loại sản phẩm đó
- Nút Cart sẽ đưa người dùng qua trang giỏ hàng
- Danh sách Brand và Category sẽ được Controller lấy từ Database và truyền xuống View

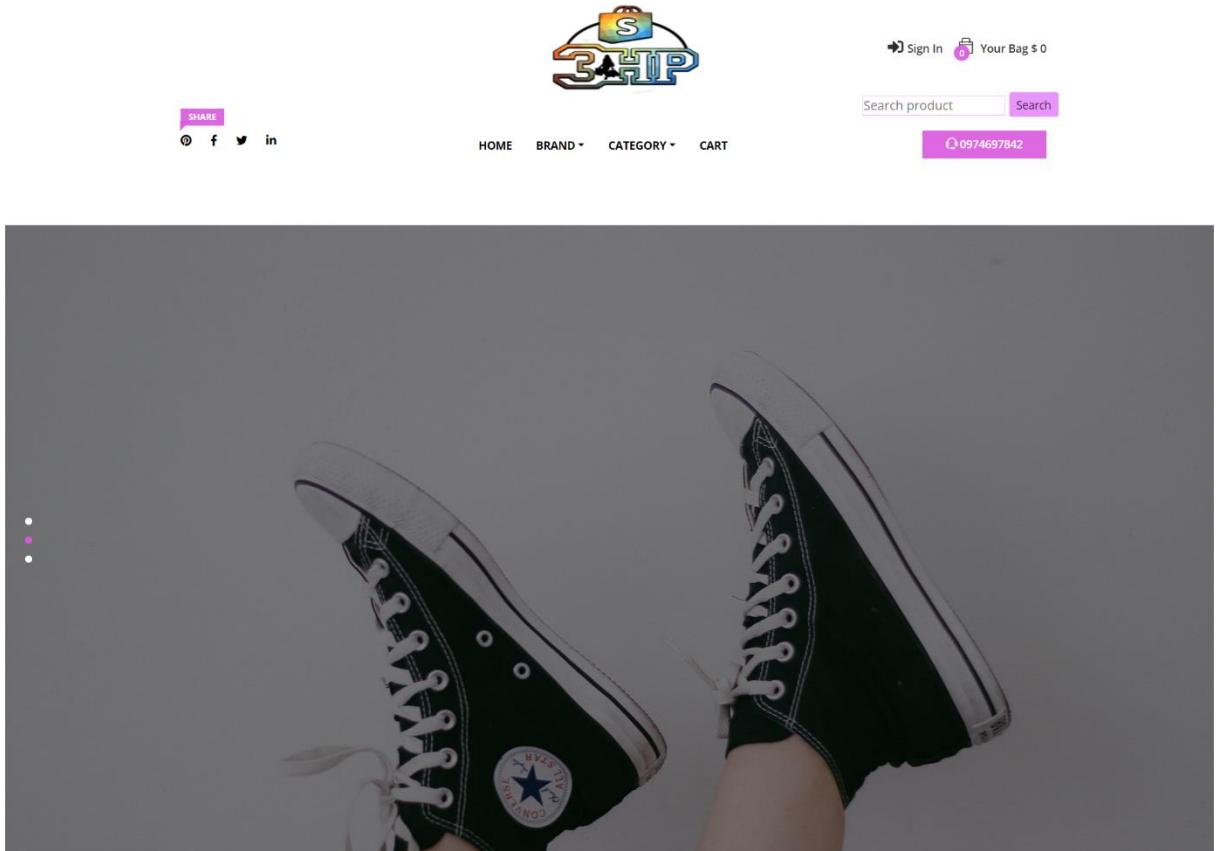
```
$cate_product = Category::where('category_status', '0')->orderBy('category_name', 'asc')->get();  
$brand_product = Brand::where('brand_status', '0')->orderBy('brand_name', 'asc')->get();
```

Query 1: Lấy sản phẩm và thương hiệu

Query 2: Lấy dữ liệu sản phẩm và thương hiệu

=> Header của Website sẽ xuất hiện ở các trang của Website, để người dùng có thể tương tác với Website dễ dàng hơn

- Màn hình trang chủ



Hình 10: Trang chủ

1.4 Body trang chủ (Tag Hot Product)

- Ở Body của trang chủ sẽ có 2 tag là “Hot product” và “All”
- Mặc định khi trang vừa load lên thì tag Hot Product sẽ được Focus
- Ở Tag “Hot product” sẽ hiển thị ra 8 sản phẩm nổi bật nhất, 8 sản phẩm này do Controller đọc từ Database và truyền xuống, và sắp theo id của sản phẩm

```
$hot_products = DB::table('tbl_product')->where('product_status', '0')->orderBy('product_id', 'desc')->limit(8)->get();
```

Query 3: Lấy sản phẩm hot

➤ Màn hình sản phẩm hot

HOT PRODUCT ALL



380,000 VND
MŨ BÓNG CHÀY THÊU NỐI SIÊU
NHE

Add to cart



430,000 VND
MŨ BÓNG CHÀY 3 SỌC AEROREADY

Add to cart



400,000 VND
MŨ DAD CAP

Add to cart



6,500,000 VND
Mũ Gucci Original GG Canvas Black

Add to cart



6,200,000 VND
Mũ Gucci Original GG Canvas
Baseball With Web Beige

Add to cart



7,600,000 VND
Mũ Gucci Original GG Canvas
Baseball With Web Black

Add to cart



1,685,000 VND
Supreme Static Knitted Beanie Hat

Add to cart



2,377,000 VND
Supreme Checkerboard Boucle
Camp Cap SS19

Add to cart

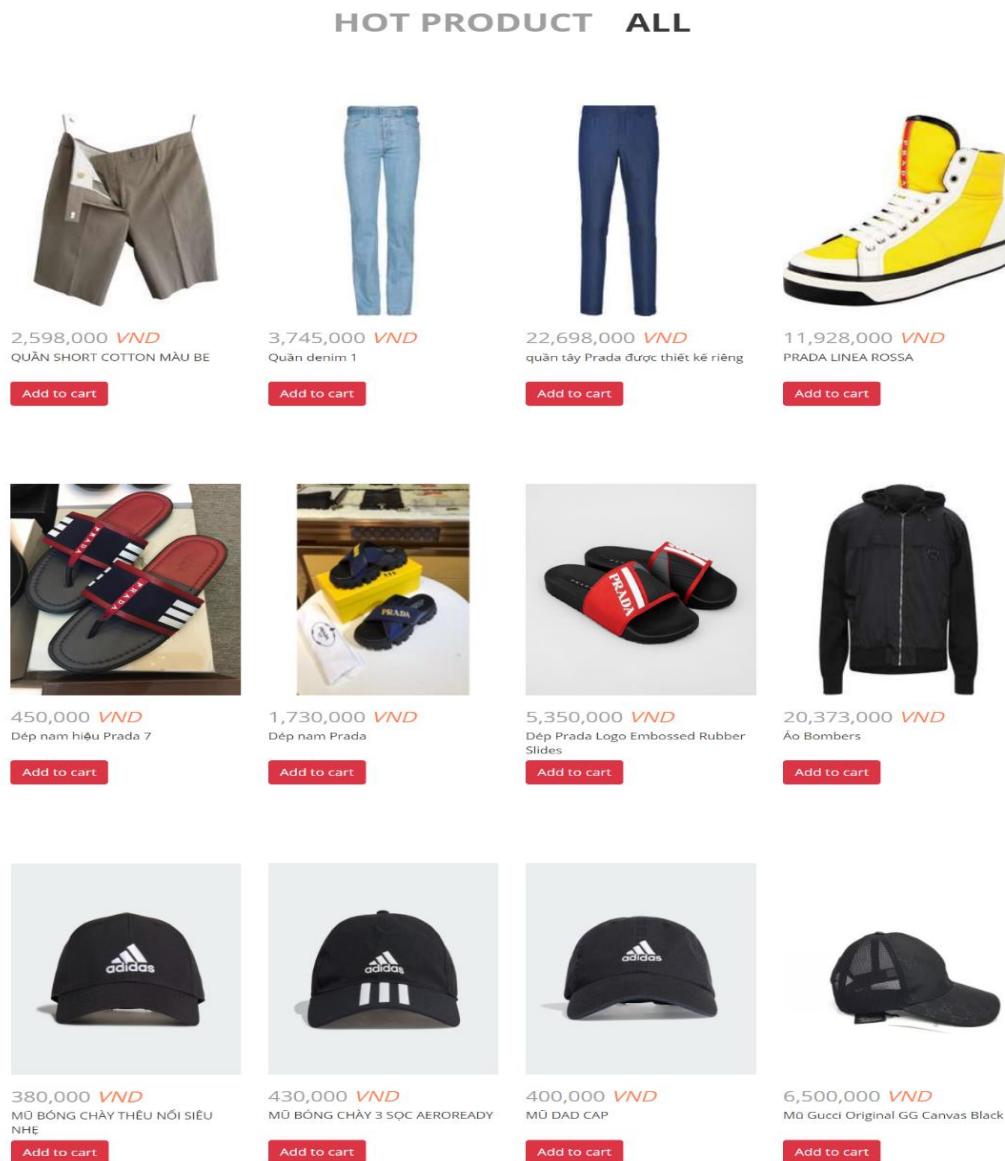
Hình 11: Sản phẩm hot

1.1.5 Body Trang chủ (Tag All)

- Ở Tag “All” sẽ hiển thị ra toàn bộ sản phẩm. Toàn bộ sản phẩm này sẽ được Controller đọc từ Database và truyền xuống, sắp xếp theo id của Category sản phẩm đó

```
$all_product = Product::where('product_status', '0')->orderByDesc('category_id')->get();
```

Query 4: Lấy tất cả sản phẩm



Hình 12: Tất cả sản phẩm

- Thông tin hiển chi tiết 1 sản phẩm gồm có:
- ❖ Hình ảnh sản phẩm
 - ❖ Giá sản phẩm
 - ❖ Tên sản phẩm
- Nút “Add to cart sẽ thêm 1 sản phẩm đó vào giỏ hàng”:

.1.6 Footer Trang chủ

- Footer của trang chủ sẽ show ra thông tin chung của toàn bộ sản phẩm, cũng như về chất lượng, tạo lòng tin cho khách hàng
- Ngoài ra tên của những người xây dựng trang web này cũng xuất hiện ở đây, như 1 lời tri ân và đánh dấu bảng quyền cho nhóm 7, nhóm đã tạo nên Website này
- Thông tin của các tác giả được Controller lấy từ Database, sau đó truyền xuống View

```
$authors = DB::table('tbl_author')->get();
```

Query 5: Lấy dữ liệu author

- Footer của Webside sẽ xuất hiện ở các trang của Web



AUTHOR

“

Tên: Trần Hữu Phúc Sở thích: đọc truyện, anime, đi du lịch dù không có tiền cũng ráng lèch
đi. Yêu thích: thể thao đặc biệt là bóng đá. Thần tượng: Lionel Messi. Sở trường: ăn, ngủ,
phâ. Sở đoản: lười, lười vcl. Ước mơ: dùng mọi cách kiếm thật nhiều tiền

○
○
○
●



Trần Hữu Phúc
Dù thế giới có quay lưng với em Thế giới hướng nào anh quay theo hướng đó

”

Hình 13: Hiển thị author

.1.7 Chi tiết cách Controller truyền biến xuống View

.1.7.1 Controller

```
class HomeController extends Controller
{
    //Truyền thông tin xuống trang chủ
    public function index()
    {
        //Lấy thông tin các Category
        $cate_product = Category::where('category_status', '0')->orderBy('category_name', 'asc')->get();
        //Lấy thông tin các Brand
        $brand_product = Brand::where('brand_status', '0')->orderBy('brand_name', 'asc')->get();
        //Lấy thông tin các Product
        $all_product = Product::where('product_status', '0')->orderByDesc('category_id')->get();
        //Lấy thông tin các sản phẩm đang hot
        $hot_products = DB::table('tbl_product')->where('product_status', '0')->orderBy('product_id', 'desc')->limit(8)->get();
        //Lấy thông tin các tác giả
        $authors = DB::table('tbl_author')->get();
        //Truyền các thông tin xuống View
        return view('pages.home')
            ->with('category', $cate_product)
            ->with('brand', $brand_product)
            ->with('all_product', $all_product)
            ->with('hot_products', $hot_products)
            ->with('authors', $authors);
    }
}
```

Controller 1: Index

.1.7.2 Router:

```
//Trang chủ
Route::get('/', 'HomeController@index');
Route::get('/homePage', 'HomeController@index');
```

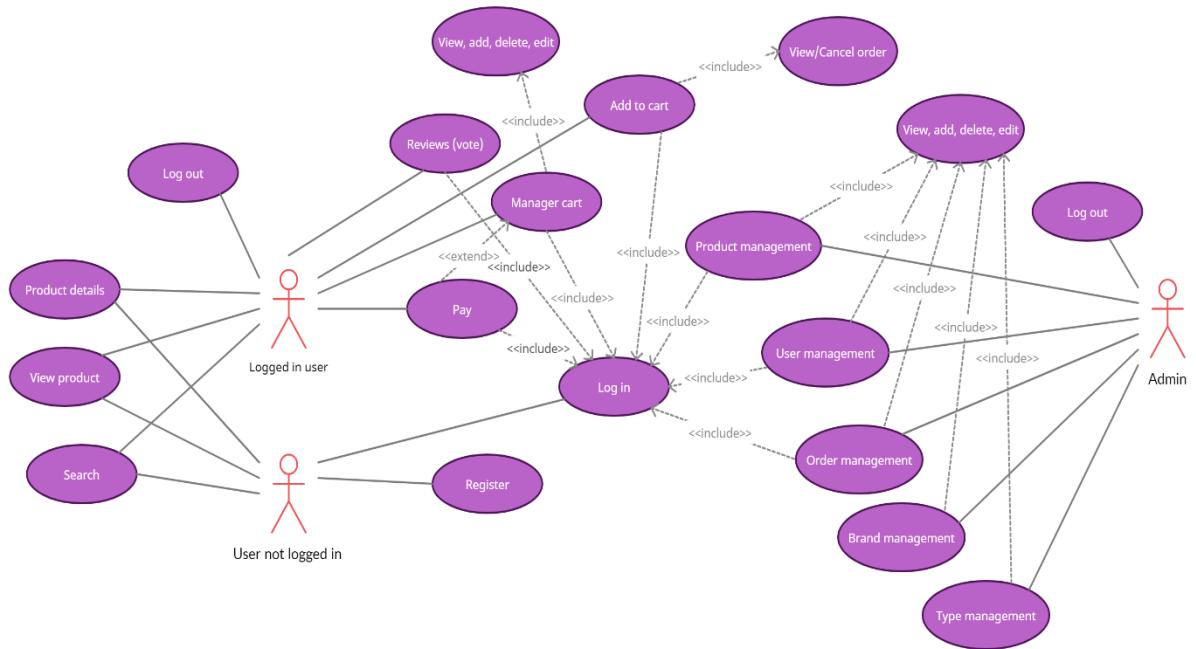
Router 1: Index

.1.8 Link host

<https://nhom7-be2-st3.000webhostapp.com/>

2. Sơ đồ use case website

2.1 Sơ đồ use case



Hình 14: Sơ đồ use case

2.2 Các chức năng

2.2.1 Admin:

STT/ID	Tên chức năng	STT/ID	Tên chức năng	STT/ID	Tên chức năng
1.1	Logout	1.8	Edit Category	1.15	Delete Product
1.2	Search Product	1.9	View Brand	1.16	Edit Product
1.3	Search Category	1.10	Add New Brand	1.17	View Order List
1.4	Search Brand	1.11	Delete Brand	1.18	View Order Detail
1.5	View Category	1.12	Edit Brand	1.19	Processing Order
1.6	Add New Category	1.13	View Product	1.20	Delete Order
1.7	Delete Category	1.14	Add New Product		

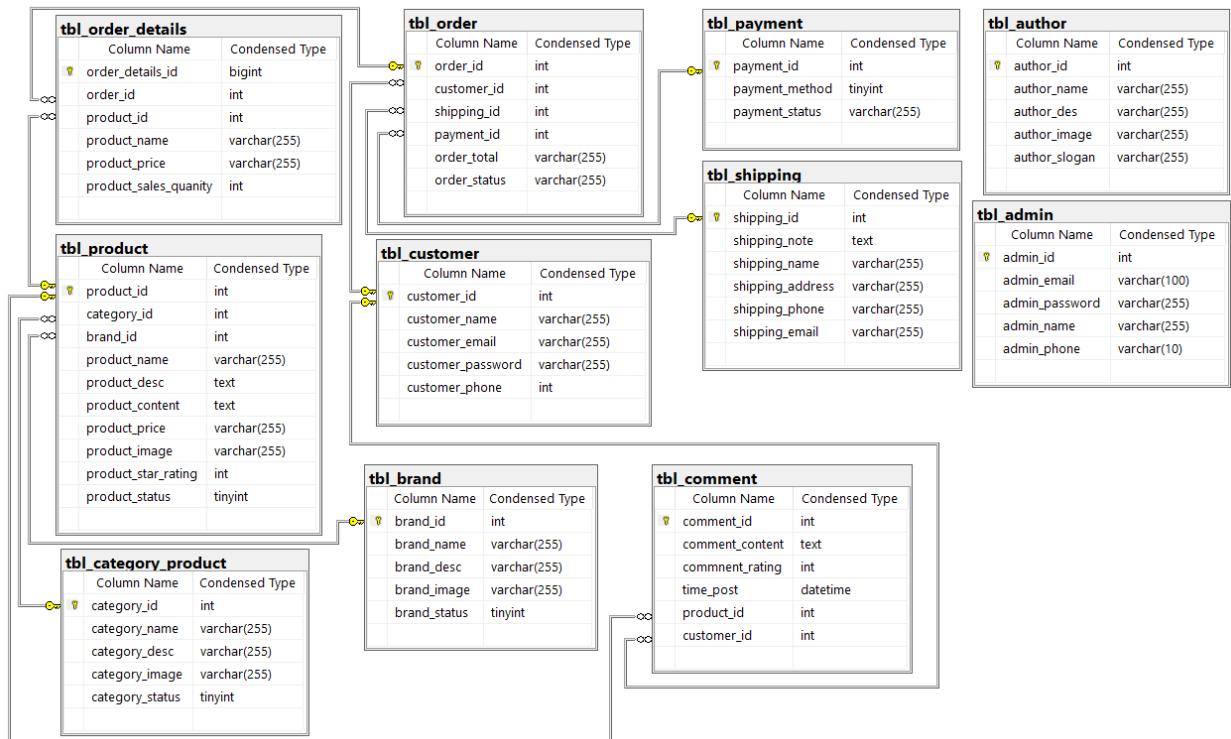
.2.2.2 Logged in user

STT/ID	Tên chức năng	STT/ID	Tên chức năng	STT/ID	Tên chức năng
2.1	Logout	2.7	View Brand	2.13	Clear Cart
2.2	Search Products	2.8	View Cart	2.14	Edit Order
2.3	Get products By Category	2.9	View Order	2.15	Send Order
2.4	Get products By Brand	2.10	View Product Detail	2.16	Cancel Order
2.5	View Products	2.11	Add Products to Cart	2.17	Comment product
2.6	View Category	2.12	Remove Product From Cart		

.2.2.3 User not logged in

STT/ID	Tên chức năng	STT/ID	Tên chức năng	STT/ID	Tên chức năng
3.1	Login	3.5	Get products By Brand	3.9	View Cart
3.2	Register	3.6	View Products	3.10	View Product Detail
3.3	Search Product	3.7	View Category	3.11	Add Products to Cart
3.4	Get products By Category	3.8	View Brand	3.12	Remove Product From Cart

3. Mô tả cơ sở dữ liệu



Hình 15: Diagram

3.1 Bảng tbl_product

STT	Trường	Loại	Giải thích
1	Product_id	Int	Mã id của sản phẩm là duy nhất.
2	Category_id	Int	Mã id của loại sản phẩm.
3	Brand_id	Int	Mã id của thương hiệu.
4	Product_name	Varchar(255)	Tên sản phẩm.
5	Product_desc	Text	Mô tả của sản phẩm.
6	Product_content	Text	Nội dung sản phẩm.
7	Product_price	Varchar(255)	Giá sản phẩm.
8	Product_image	Varchar(255)	Hình ảnh sản phẩm.
9	Product_star_rating	Int	Số sao của sản phẩm
10	Product_status	Tinyint	Trạng thái của sản phẩm (0/1 - Ân/hiện).

.3.2 Bảng tbl_brand

STT	Trường	Loại	Giải thích
1	Brand_id	Int	Mã id của thương hiệu là duy nhất.
2	Brand_name	Varchar(255)	Tên của thương hiệu.
3	Brand_desc	Varchar(255)	Mô tả thương hiệu.
4	Brand_image	Varchar(255)	Hình ảnh thương hiệu.
5	Brand_status	Tinyint	Trạng thái thương hiệu (0/1 - Ân/hiện).

.3.3 Bảng tbl_category_product

STT	Trường	Loại	Giải thích
1	Category_id	Int	Mã id của loại sản phẩm là duy nhất.
2	Category_name	Varchar(255)	Tên của loại sản phẩm.
3	Category_desc	Varchar(255)	Mô tả loại sản phẩm.
4	Category_image	Varchar(255)	Hình ảnh loại sản phẩm.
5	Category_status	Tinyint	Trạng thái loại sản phẩm (0/1 - Ân/hiện).

.3.4 Bảng tbl_customer

STT	Trường	Loại	Giải thích
1	Customer_id	Int	Mã id của khách hàng là duy nhất.
2	Customer_name	Varchar(255)	Tên tài khoản khách hàng.
3	Customer_email	Varchar(255)	Thư điện tử của khách hàng.
4	Customer_password	Varchar(255)	Mật khẩu của tài khoản khách hàng.
5	Customer_phone	Int	Số điện thoại khách hàng.

.3.5 Bảng tbl_order

STT	Trường	Loại	Giải thích
1	Order_id	Int	Mã id đặt hàng là duy nhất.
2	Customer_id	Int	Mã id của khách hàng.
3	Shipping_id	Int	Mã id của đơn hàng.
4	Payment_id	Int	Mã id của trả tiền.
5	Order_total	Varchar(255)	Tổng tiền của đơn hàng.
6	Order_status	Varchar(255)	Trạng thái đơn hàng.

.3.6 Bảng tbl_order_details

STT	Trường	Loại	Giải thích
1	Order_details_id	Bigint	Mã id chi tiết đơn hàng là duy nhất.
2	Order_id	Int	Mã id của đơn hàng.
3	Product_id	Int	Mã id của sản phẩm.
4	Product_name	Varchar(255)	Tên sản phẩm.
5	Product_price	Varchar(255)	Giá sản phẩm.
6	Product_sales_quantity	Int	Số lượng sản phẩm.

.3.7 Bảng tbl_shipping

STT	Trường	Loại	Giải thích
1	Shipping_id	Int	Mã id của đơn hàng là duy nhất.
2	Shipping_name	Varchar(255)	Tên người nhận.
3	Shipping_email	Varchar(255)	Thư điện tử của khách hàng.
4	Shipping_note	Text	Lưu ý giao hàng.
5	Shipping_phone	Varchar(255)	Số điện thoại người nhận.
6	Shipping_address	Varchar(255)	Địa chỉ giao hàng.

.3.8 Bảng tbl_payment

STT	Trường	Loại	Giải thích
1	Payment_id	Int	Mã id trả tiền là duy nhất.
2	Payment_method	Tinyint	Cách thức trả tiền (1/2 – Trả tiền mặt/Trả bằng ví điện tử)
3	Payment_status	Varchar(255)	Trạng thái trả tiền.

.3.9 Bảng tbl_comment

STT	Trường	Loại	Giải thích
1	Comment_id	Int	Mã id của bình luận là duy nhất.
2	Comment_content	Text	Nội dung bình luận.
3	Comment_rating	Int	Số sao bình phẩm.
4	Time_post	Datetime	Thời gian bình luận.
5	Product_id	Int	Mã id của sản phẩm được bình luận.
6	Customer_id	Int	Mã id của khách hàng bình luận.

.3.10 Bảng tbl_admin

STT	Trường	Loại	Giải thích
1	Admin_id	Int	Mã id của tài khoản là duy nhất.
2	Admin_name	Varchar(255)	Tên tài khoản admin.
3	Admin_password	Varchar(255)	Mật khẩu của tài khoản.
4	Admin_email	Varchar(255)	Email của admin.
5	Admin_phone	Varchar(255)	Số điện thoại của admin.

.3.11 Bảng tbl_author

STT	Trường	Loại	Giải thích
1	Author_id	Int	Mã id của tác giả là duy nhất.
2	Author_name	Varchar(255)	Tên tác giả.
3	Author_des	Varchar(255)	Mô tả về tác giả.
4	Author_image	Varchar(255)	Ảnh tác giả.
5	Author_slogan	Varchar(255)	Câu nói tâm đắc của tác giả.

4. Chức năng người dùng khách hàng

4.1 Nhận biết người dùng khách hàng đăng nhập hay chưa

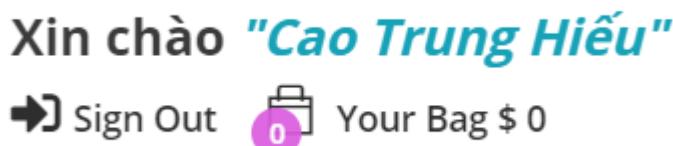
- Ở trên Header, nếu người dùng đã đăng nhập rồi, thì hệ thống sẽ in ra câu “Chào ‘Tên người dùng’”, còn nếu chưa đăng nhập thì hệ thống sẽ không in câu chào nào cả
- Nếu người dùng chưa đăng nhập, hệ thống sẽ để 1 icon “Sign In” trên Header để người đăng nhập
- Nếu người dùng đã đăng nhập, hệ thống sẽ để 1 icon “Sign Out” để người đăng xuất khỏi tài khoản
- Khi người dùng đăng nhập thành công, hệ thống sẽ lưu Session gồm id và tên của người dùng. Id của người dùng dùng để cho phần Order, còn Tên khách hàng sẽ được thực hiện cho câu chào

.4.1.1 Khi khách hàng chưa đăng nhập



Hình 16: Người dùng chưa đăng nhập

.4.1.2 Khi khách hàng đăng nhập thành công



Hình 17: Người dùng đã đăng nhập

.4.1.3 Ở Controller, nếu người dùng đăng nhập thành công thì Lưu Session lại

```
//Nếu người dùng đăng nhập thành công
if ($result) {
    Session::put('customer_id', $result->customer_id);
    Session::put('customer_name', $result->customer_name);

    return redirect('/homePage')->with('all_product', $all_product);
}
```

Controller 2: Điều kiện người đăng nhập

.4.1.4 Ở View, kiểm tra nếu Session “customer_name” tồn tại thì in câu chào ra

```
@if(Session::has("customer_name") != null)
<h5>Xin chào <i class="text-info">"{{ Session::get('customer_name') }}"</i></h5>
@endif
```

View 1: Điều kiện đăng nhập

4.1.5 Kiểm tra Session Id để in Sign In hoặc Sign Out

```
<!-- Nếu biến Session customer_id tồn tại (tức là ng dùng đã đăng nhập) thì in Sign Out ra -->
@if(Session::has("customer_id") != null)
|   <a class="user" href="{{URL::to('/logout_checkout')}}" title="Sign Out" rel="nofollow">
|       <i class="fas fa-sign-in-alt"></i>Sign Out</a>
<!-- Nếu biến Customer_id không có giá trị (tức là ng dùng chưa đăng nhập) thì in Sign In ra -->
@else
|   <a class="user" href="{{URL::to('/login_checkout')}}" title="Sign In" rel="nofollow">
|       <i class="fas fa-sign-in-alt"></i>Sign In</a>
@endif
```

View 2: Điều kiện đăng nhập hoặc đăng xuất

4.2 Xem danh sách các sản phẩm nổi bật

- Ở trang chủ sẽ có 2 tag là “Hot product” và “All”
- Mặc định khi trang vừa load lên thì tag Hot Product sẽ được Focus
- Ở Tag “Hot product” sẽ hiển thị ra 8 sản phẩm nổi bật nhất, 8 sản phẩm này do Controller đọc từ Database và truyền xuống, và sắp theo id của sản phẩm

```
$hot_products = DB::table('tbl_product')->where('product_status', '0')->orderBy('product_id', 'desc')->limit(8)->get();
```

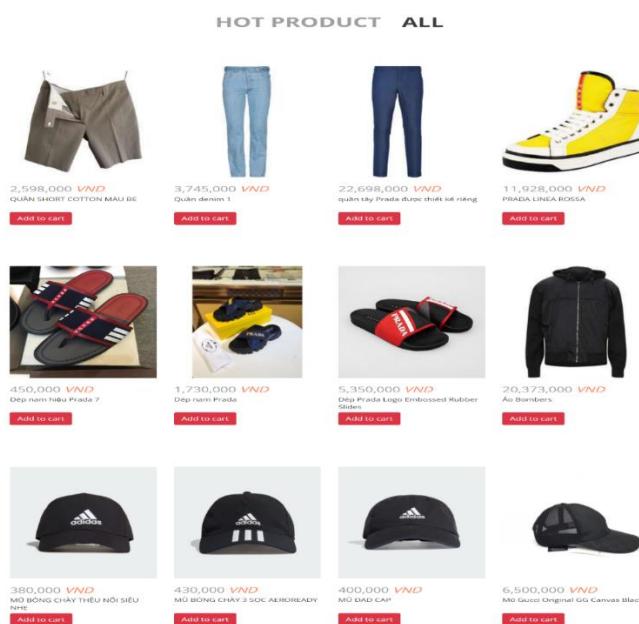
Query 6: Lấy 8 sản phẩm

4.3 Xem danh sách tất cả sản phẩm

- Ở Tag “All” sẽ hiển thị ra toàn bộ sản phẩm. Toàn bộ sản phẩm này sẽ được Controller đọc từ Database và truyền xuống, sắp xếp theo id của Category sản phẩm đó

```
$all_product = Product::where('product_status', '0')->orderByDesc('category_id')->get();
```

Query 7: Lấy tất cả sản phẩm



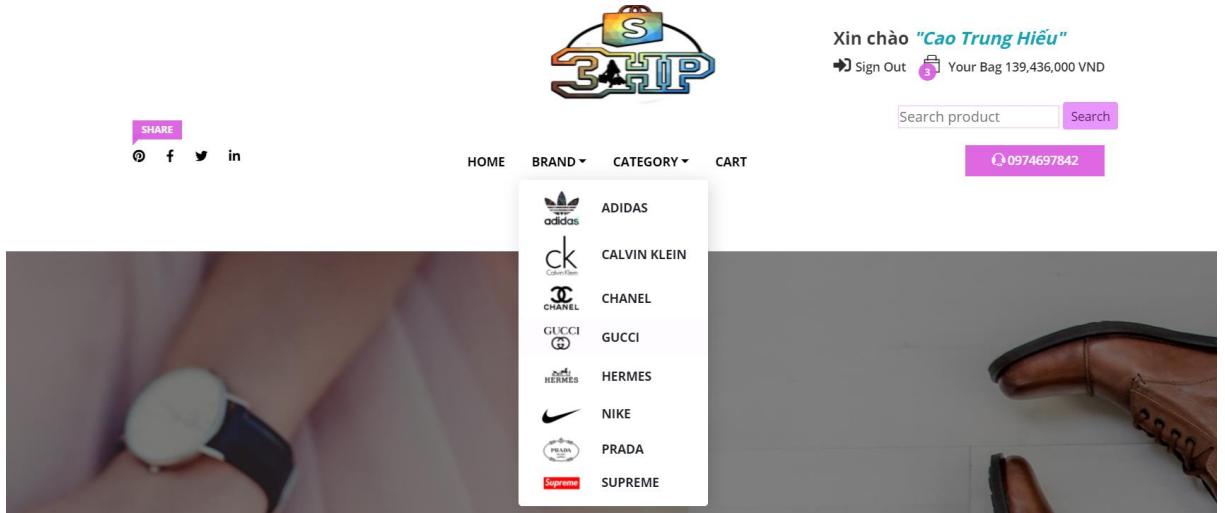
Hình 18: Màn hình tất cả sản phẩm

- Thông tin hiển chi tiết 1 sản phẩm gồm có:

- ❖ Hình ảnh sản phẩm
- ❖ Giá sản phẩm
- ❖ Tên sản phẩm

4.4 Xem danh sách sản phẩm theo thương hiệu (Brand)

- Ở phần Header của Website, có một Combobox “Brand”, khi người dùng nhấn vào sẽ Comobox sẽ xổ xuống tất cả các loại nhãn hiệu mà admin cho phép xuất hiện (Dựa vào product_status, nếu là 0 thì là hiển thị sản phẩm ra, còn 1 thì ẩn sản phẩm đó đi)



Hình 19: Hiển thị thương hiệu

- Danh sách Brand sẽ do Controller đọc từ Database, và truyền xuống View

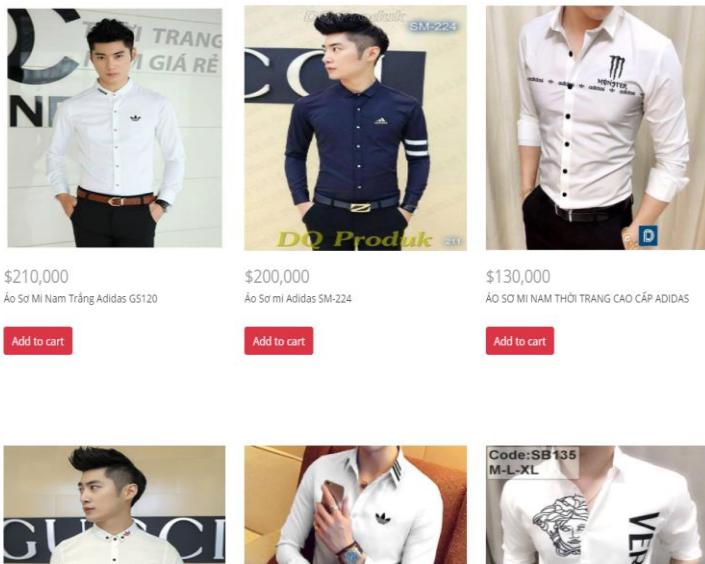
```
public function show_brand_home($brand_id)
{
    //Lấy các sản phẩm thuộc Brand_id view truyền lên
    $brand_by_id = Product::where('brand_id', $brand_id)->get();
    //Các thông số cần cho Header và Footer
    $cate_product = Category::where('category_status', '0')->orderBy('category_name', 'asc')->get();
    $brand = Brand::where('brand_status', 0)->orderBy('brand_name', 'asc')->get();
    $brand_name = Brand::where('tbl_brand.brand_id', $brand_id)->take(1)->get();
    $all_product = Product::where('product_status', '0')->orderByDesc('category_id')->get();
    $authors = DB::table('tbl_author')->get();

    return view('pages.brand.show_brand')
        ->with('category', $cate_product)
        ->with('brand', $brand)
        ->with('brand_id', $brand_by_id)
        ->with('brand_name', $brand_name)
        ->with('all_product', $all_product)
        ->with('authors', $authors);
}
```

Controller 3: Hiển thị thương hiệu

➤ Giao diện khi nhấn vào 1 Brand

WE FIND 48 PRODUCT OF ADIDAS



Hình 20: Sản phẩm của một thương hiệu

➤ Route

```
// Category product home
Route::get('/category_product/{category_id}', 'CategoryProduct@show_category_home');
```

Router 2: Đường dẫn category

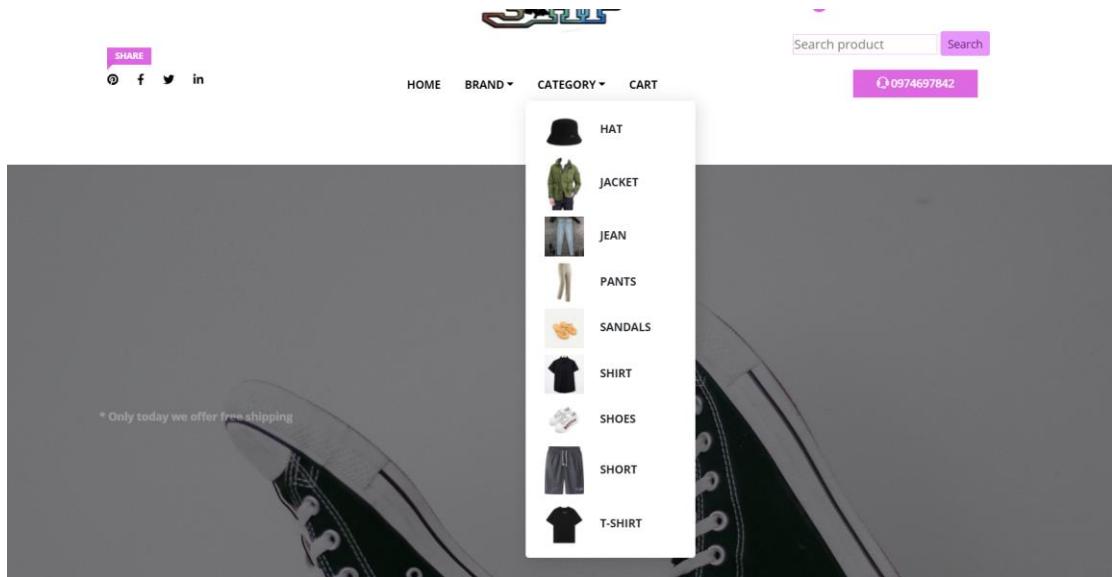
➤ View Code

```
<section class="new_arrivals_area section_padding_100_0 clearfix">
    <div class="container">
        <div class="row">
            <div class="col-12">
                <div class="section_heading text-center">
                    @foreach ($brand_name as $key => $name_brand)
                        <h2>We find <i class="text-info">{{ count($brand_id) }}</i> Product of {{ $name_brand->brand_name }}</h2>
                    @endforeach
                </div>
            </div>
        </div>
        <div class="container">
            <div class="row karl-new-arrivals">
                <!-- Single gallery Item Start -->
                @foreach ($brand_id as $key => $product)
                    <a href="{{ URL::to('/details_product/'.$product->product_id) }}">...
                    </a>
                @endforeach
            </div>
        </div>
    </div>
</section>
```

View 3: category

4.5 Xem danh sách sản phẩm loại sản phẩm (Category)

- Ở phần Header của Website, có một Combobox “Category”, khi người dùng nhấn vào sẽ Comobox sẽ xổ xuống tất cả các loại sản phẩm mà admin cho phép xuất hiện (Dựa vào product_status, nếu là 0 thì là hiển thị sản phẩm ra, còn 1 thì ẩn sản phẩm đó đi)



Hình 21: Loại sản phẩm

- Danh sách Category sẽ do Controller đọc từ Database, và truyền xuống View

```
public function show_category_home($category_id)
{
    //Lấy tất cả sản phẩm dựa vào category_id view gửi lên
    $category_by_id = Product::where('category_id', $category_id)->get();
    //Các thông số cần cho Header và Footer
    $brand_product = Brand::where('brand_status', 0)->orderBy('brand_name', 'asc')->get();
    $cate_product = Category::where('category_status', 0)->orderBy('category_name', 'asc')->get();
    $category_name = Category::where('category_id', $category_id)->take(1)->get();
    $all_product = Product::where('product_status', '0')->orderByDesc('category_id')->get();
    $authors = DB::table('tbl_author')->get();

    return view('pages.category.show_category')
        ->with('category', $cate_product)
        ->with('brand', $brand_product)
        ->with('category_id', $category_by_id)
        ->with('category_name', $category_name)
        ->with('all_product', $all_product)
        ->with('authors', $authors);
}
```

Controller 4:Loại sản phẩm

➤ Giao diện khi nhấn vào 1 Category

Hình 22: Sản phẩm của một loại

➤ Route

```
// Category product home
Route::get('/category_product/{category_id}', 'CategoryProduct@show_category_home');
```

Router 3: Loại sản phẩm

➤ View Code

```
<section class="new_arrivals_area section_padding_100_0 clearfix">
    @foreach ($category_name as $key => $name_category)

        <div class="container">
            <div class="row">
                <div class="col-12">
                    <div class="section_heading text-center">
                        |   <i class="text-info">{{ count($category_id) }}</i> Product of {{ $name_category->category_name }}</h2>
                    </div>
                </div>
            </div>
        @endforeach

        <div class="container">
            <div class="row karl-new-arrivals">

                <!-- Single gallery Item Start -->
                @foreach ($category_id as $key => $product)
                    <a href="{{ URL::to('/details_product/'.$product->product_id) }}">...
                    </a>
                @endforeach
            </div>
        </div>
    </section>
```

View 4: Loại sản phẩm

4.6 Tìm kiếm sản phẩm theo từ khoá (Keyword)

- Ở phần Header của Website, Sẽ có 1 ô input cho người dùng nhập từ khoá vào, và khi nhấn nút Search, Website sẽ chuyển sang trang Search và Show tất cả sản phẩm có từ khoá đó



Hình 23: Tìm kiếm sản phẩm

- Khi ấn vào hoặc ghi gì đó vào ô input, hàng loạt tên sản phẩm gợi ý sẽ được xổ ra để người dùng tìm kiếm tiện lợi hơn.



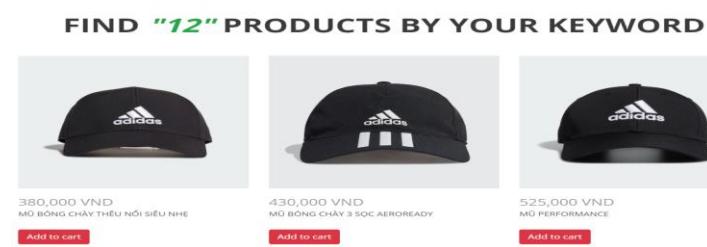
Hình 24: Tìm kiếm sản phẩm

- Để làm được các từ khoá sẵn đó thì chúng ta sẽ sử dụng DataList, với các Option lấy từ tên tất cả sản phẩm Controller gửi qua

```
<!-- Khai báo autocomplete="off" để ngăn các từ khóa mà google tự động nhá -->
<form action="{{URL::to('/search')}}" method="POST" class="font-form" autocomplete="off">
    @csrf
    <div class="search_box pull_right">
        <!-- Phải khai báo list=id của datalist -->
        <input class="help-line input-search" list="browsers" type="text" name="keywords_submit" id="" placeholder="Search product" required>
        <!-- Datalist -->
        <datalist id="browsers">
            <!-- Các Option -->
            @foreach($all_product as $product)
            <option value="{{($product->product_name)}}">
            @endforeach
        </datalist>
        <input type="submit" value="Search" class="btn btn-default btn-sm search-font">
    </div>
</form>
```

View 5: Tìm kiếm sản phẩm

- Giao diện chính khi nhấn vào nút tìm kiếm



Hình 25: Sản phẩm được tìm kiếm

- Danh sách sản phẩm theo từ khoá sẽ do Controller đọc từ Database, và truyền xuống View

```

public function search(Request $request)
{
    //Lấy từ khoá ra
    $keywords = $request->keywords_submit;
    //Lấy danh sách sản phẩm ra theo từ khoá
    $search_product = Product::where('product_name', 'like', '%' . $keywords . '%')->get();
    //các thông số khác cần cho Header và Footer
    $cate_product = Category::where('category_status', '0')->orderBy('category_name', 'asc')->get();
    $brand_product = Brand::where('brand_status', '0')->orderBy('brand_name', 'asc')->get();
    $all_product = Product::where('product_status', '0')->orderByDesc('category_id')->get();
    $authors = DB::table('tbl_author')->get();

    return view('/pages/product/search')
        ->with('category', $cate_product)
        ->with('brand', $brand_product)
        ->with('search_product', $search_product)
        ->with('all_product', $all_product)
        ->with('authors', $authors);
}

```

Controller 5: Tìm kiếm sản phẩm

- Route

```

//Search
Route::post('/search', 'HomeController@search');

```

Router 4: Tìm kiếm sản phẩm

- View Code

```

<section class="new_arrivals_area section_padding_100_0 clearfix">
    <div class="container">
        <div class="row">
            <div class="col-12">
                <div class="section_heading text-center">
                    <h2>We find <i class="text-success">"{{ count($search_product) }}" products by your keyword</h2>
                </div>
            </div>
        </div>
    </div>

    <div class="container">
        <div class="row karl-new-arrivals">
            <!-- Single gallery Item Start -->
            @foreach ($search_product as $key => $product)
                <a href="{{ URL::to('/details_product/'.$product->product_id) }}">...
            </a>
            @endforeach
        </div>
    </div>
</section>

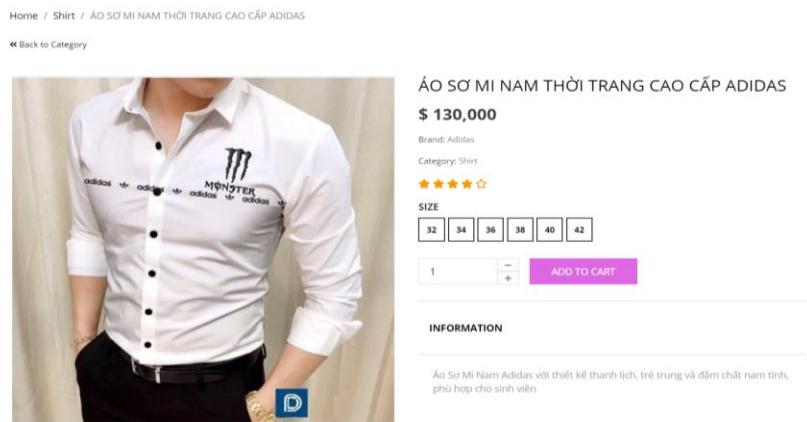
```

View 6: Tìm kiếm sản phẩm

4.7 Xem chi tiết sản phẩm

- Người dùng có thể xem chi tiết sản phẩm bằng cách nhấp vào hình sản phẩm muốn xem. Các chi tiết người dùng có thể xem là hình ảnh, giá, ... về sản phẩm
- Mỗi sản phẩm trong danh sách sản phẩm đều sẽ dẫn đến Controller của chi tiết sản phẩm, và id của sản phẩm đó sẽ được hiển thị lên thanh địa chỉ của Website, Controller chi tiết sản phẩm sẽ lấy id sản phẩm đó xuống, thực hiện truy vấn dựa trên id đó, và trả kết quả trả về ở trang Product Detail

➤ View



Hình 26: Chi tiết sản phẩm

➤ Controller

```
public function product_details($product_id)
{
    //Những thông số cơ bản để chạy Header và Footer
    $cate_product = Category::where('category_status', '0')->orderBy('category_name', 'asc')->get();
    $brand_product = Brand::where('brand_status', '0')->orderBy('brand_name', 'asc')->get();
    $all_product = Product::where('product_status', '0')->orderByDesc('category_id')->get();
    $authors = DB::table('tbl_author')->get();

    //Trong trang Product Detail có hiển thị ra Brand và Category của sản phẩm đó nên phải nối 2 bảng đó
    $details_product = DB::table('tbl_product')
        ->join('tbl_category_product', 'tbl_category_product.category_id', '=', 'tbl_product.category_id')
        ->join('tbl_brand', 'tbl_brand.brand_id', '=', 'tbl_product.brand_id')
        ->where('tbl_product.product_id', $product_id)
        ->get();
    foreach ($details_product as $value) {
        $category_id = $value->category_id;
    }
    $related_product = Product::where('tbl_product.category_id', $category_id)->whereNotIn('tbl_product.product_id', [$product_id])->get();
    return view('pages.product.show_details')
        ->with('category', $cate_product)
        ->with('brand', $brand_product)
        ->with('product_details', $details_product)
        ->with('related_product', $related_product)
        ->with('all_product', $all_product)
        ->with('authors', $authors);
}
```

Controller 6: Chi tiết sản phẩm

➤ Router

```
// Detail product home
Route::get('/details_product/{product_id}', 'ProductController@product_details');
```

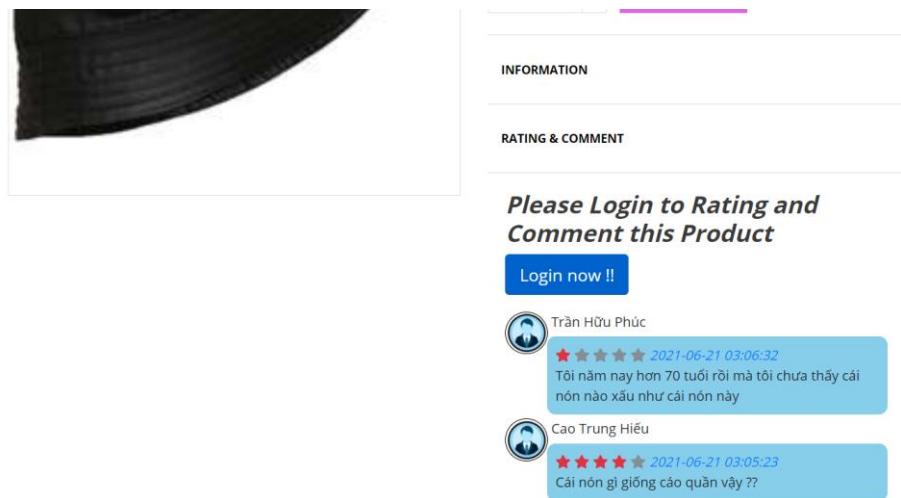
Router 5: Chi tiết sản phẩm

➤ View

View 7: Chi tiết sản phẩm

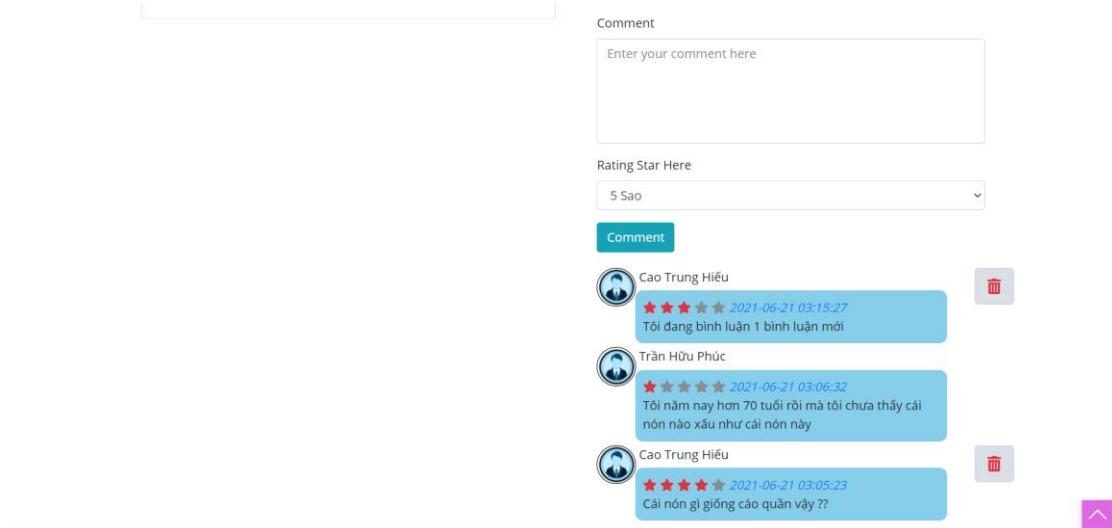
.4.8 Comment và đánh giá sản phẩm

- Ở trang chi tiết sản phẩm, người dùng có thể ấn vào tag “**RATING & COMMENT**” để xem các bình luận và đánh giá sao về sản phẩm đó
 - Đối với người dùng chưa đăng nhập, tag “**RATING & COMMENT**” sẽ hiện thông báo yêu cầu đăng nhập để tham gia comment và rating



Hình 27: Đánh giá sản phẩm chưa đăng nhập

- Đối với người dùng đăng nhập thì form đăng nhập sẽ hiển thị ra, và người dùng có thể tham gia Comment và Rating



Hình 28: Đánh giá sản phẩm đã đăng nhập

- Icon Delete chỉ hiện ra đối với các Comment của người dùng đang đăng nhập, người dùng có thể xoá Comment của chính mình
- Khi người dùng thêm hoặc xoá Comment, số sao của sản phẩm sẽ được Update lại dựa trên số sao trung bình của các Comment về sản phẩm đó
- Lấy danh sách comment của 1 sản phẩm ở controller

```
//Lấy các thông số của Comment (chỉ lấy 5 comment để phân trang)
$comments = DB::table('tbl_comment')
->join('tbl_customer', 'tbl_customer.customer_id', '=', 'tbl_comment.customer_id')
->where('tbl_comment.product_id', $product_id)
->orderBy('comment_id', 'DESC')
->paginate(5);
```

Query 8: Đánh giá sản phẩm phân trang

- Hàm tính sao trung bình

```
//Hàm tính sao trung bình
private function getAverageRating($list_comment){
    if(count($list_comment) == 0){
        return 5;
    }
    $num = 0;
    foreach($list_comment as $comment){
        $num += $comment->comment_rating;
    }
    return round($num/count($list_comment));
}
```

Controller 7: Hàm tính trung bình sao

➤ Controller Thêm một Comment do người dùng gửi lên

```
//Thêm Comment
public function AddNewComment(Request $request)
{
    //Bắt các thông tin
    $comment_content = $request->comment_content;
    $comment_rating = $request->comment_rating;
    $product_id = $request->product_id;
    $custommer_id = $request->customer_id;
    //Thực hiện thêm dữ liệu mới nhận vào database
    DB::table('tbl_comment')->insert(
        [
            'comment_content' => $comment_content,
            'comment_rating' => $comment_rating,
            'product_id' => $product_id,
            'customer_id' => $custommer_id
        ]
    );
    //Bắt tất cả các comment để tính sao trung bình
    $comments = DB::table('tbl_comment')->where('product_id', $product_id)->get();
    $num = $this->getAverageRating($comments);
    //Cập nhật lại sao cho sản phẩm
    DB::table('tbl_product')->where('product_id', $product_id)->update(['product_star_rating' => $num]);
    return redirect('details_product/' . $product_id);
}
```

Controller 8: Thêm đánh giá sản phẩm

➤ Router Thêm một Comment

```
Route::post('/add-comment', 'CommentController@AddNewComment');
```

Router 6: Thêm đánh giá sản phẩm

➤ Controller xoá một Comment khi người dùng ấn vào nút xoá comment

```
//Xoá Comment
public function DeleteComment($comment_id, $product_id)
{
    //Truy vấn comment theo id và xoá comment
    DB::table('tbl_comment')->where('comment_id', $comment_id)->delete();
    //Bắt tất cả comment của sản phẩm đó lại
    $comments = DB::table('tbl_comment')->where('product_id', $product_id)->get();
    $num = $this->getAverageRating($comments);
    DB::table('tbl_product')->where('product_id', $product_id)->update(['product_star_rating' => $num]);
    return redirect('details_product/' . $product_id);
}
//Làm tính sao trung bình
```

Controller 9: Xoá đánh giá sản phẩm

➤ Router Xoá một Comment

```
Route::get('/delete_comment/{comment_id}/{product_id}', 'CommentController@DeleteComment');
```

Router 7: Xoá đánh giá sản phẩm

.4.9 Giỏ hàng

Người dùng có thể thêm hàng vào giỏ hàng, xem danh sách sản phẩm trong giỏ hàng

- ❖ Giỏ hàng có thể được xem ở 2 nơi:
 - Giỏ hàng mini: Luôn nằm trên Header của trang web
 - *Lúc chưa nhấn vào*



Hình 29: Giỏ hàng lúc chưa ấn vào

- *Lúc nhấn vào thì show các sản phẩm trong giỏ hàng mini ra*

The screenshot displays a detailed shopping cart view. At the top, it says 'Xin chào "Cao Trung Hiếu"' and shows a sign-out link and a cart summary: 'Your Bag 139,436,000 VND'. Below this, three items are listed with their details and prices:

- ÁO SƠ MI NAM THỜI TRANG CAO CẤP ADIDAS
5X - 130000 VND
- quần tây Prada được thiết kế riêng
6X - 22698000 VND
- QUẦN SHORT COTTON MÀU BE
1X - 2598000 VND

A 'Cart' button and a 'Checkout' button are at the bottom right. The total amount 'TOTAL: 139,436,000 VND' is also shown.

Hình 30: Giỏ hàng lúc ấn vào

- Chi tiết giỏ hàng: Khi người dùng nhấn vào nút “Cart” trong giỏ hàng Mini hoặc nút “Cart” trên Header của trang web

Product	Price	Quantity	Total	
ÁO SƠ MI NAM THỜI TRANG CAO CẤP ADIDAS	\$ 130,000	5	\$ 650,000	
quần tây Prada được thiết kế riêng	\$ 22,698,000	6	\$ 136,188,000	
QUẦN SHORT COTTON MÀU BE	\$ 2,598,000	1	\$ 2,598,000	

CONTINUE SHOPPING

Hình 31: Chi tiết giỏ hàng

- Khi nhấn “Add to cart” từ danh sách sản phẩm, sẽ thêm 1 sản phẩm đó vào giỏ hàng
- Khi nhấn “Add to cart” trong trang chi tiết sản phẩm, chúng ta có thể chọn số lượng sản phẩm, sau đó thêm vào giỏ hàng
- Bên dưới sẽ là tổng giá tiền của giỏ hàng

CART TOTAL

Subtotal	139,436,000.00 VND
Shipping	Free
Total	139,436,000.00 VND

PROCESS TO CHECKOUT

Hình 32: Tổng tiền giỏ hàng

- Khi nhấn vào “Process to checkout”, người dùng sẽ vào giao diện thanh toán (Hoặc Login nếu người dùng chưa đăng nhập)
 - Giỏ này dùng thư viện giỏ hàng của Laravel
- ```
use Gloudemans\Shoppingcart\Facades\Cart;
```

Use 1: Thư viện giỏ hàng

- Controller khi gửi thông tin 1 sản phẩm tới giỏ hàng

```
public function save_cart(Request $request)
{
 $product_id = $request->product_id_hidden;
 $quantity = $request->quantity;
 $product_info = Product::where('product_id', $product_id)->first();

 // Cart::add('293ad', 'Product 1', 1, 9.99, 550);
 $data['id'] = $product_info->product_id;
 $data['qty'] = $quantity;
 $data['name'] = $product_info->product_name;
 $data['price'] = $product_info->product_price;
 $data['weight'] = $product_info->product_price;
 $data['options']['image'] = $product_info->product_image;
 Cart::add($data);
 return redirect('/show_cart');
}
```

Controller 10: Lưu giỏ hàng

➤ Controller khi người dùng vào giỏ hàng lớn

```
public function show_cart()
{
 $cate_product = Category::where('category_status', '0')->orderBy('category_name', 'asc')->get();
 $brand_product = Brand::where('brand_status', '0')->orderBy('brand_name', 'asc')->get();
 $all_product = Product::where('product_status', '0')->orderByDesc('category_id')->get();
 $authors = DB::table('tbl_author')->get();

 return view('pages.cart.show_cart')
 ->with('category', $cate_product)
 ->with('brand', $brand_product)
 ->with('all_product', $all_product)
 ->with('authors', $authors);
}
```

Controller 11: Hiển thị giỏ hàng

➤ Controller xoá sản phẩm và Update số lượng sản phẩm trong giỏ hàng

```
public function show_cart()
{
 $cate_product = Category::where('category_status', '0')->orderBy('category_name', 'asc')->get();
 $brand_product = Brand::where('brand_status', '0')->orderBy('brand_name', 'asc')->get();
 $all_product = Product::where('product_status', '0')->orderByDesc('category_id')->get();
 $authors = DB::table('tbl_author')->get();

 return view('pages.cart.show_cart')
 ->with('category', $cate_product)
 ->with('brand', $brand_product)
 ->with('all_product', $all_product)
 ->with('authors', $authors);
}

//Xoá 1 sản phẩm khỏi giỏ hàng
public function delete_to_cart($rowId){
 Cart::update($rowId, 0);
 return redirect('/show_cart');
}

//Update số lượng 1 sản phẩm trong giỏ hàng
public function update_cart_quantity(Request $request){
 $rowId = $request->rowId_cart;
 $qty = $request->quantity;
 Cart::update($rowId, $qty);

 return redirect('/show_cart');
}
```

Controller 12: Xoá sản phẩm và cập nhập số lượng giỏ hàng

➤ Router

```
// Cart
Route::post('/update_cart_quantity', 'CartController@update_cart_quantity');
Route::post('/save_cart', 'CartController@save_cart');
Route::get('/show_cart', 'CartController@show_cart');
Route::get('/delete_to_cart/{rowId}', 'CartController@delete_to_cart');
```

Router 8: Giỏ hàng

#### 4.10 Chức năng Order

- Sau khi người dùng chọn process to checkout

#### CART TOTAL

|              |                           |
|--------------|---------------------------|
| Subtotal     | 139,436,000.00 VND        |
| Shipping     | Free                      |
| <b>Total</b> | <b>139,436,000.00 VND</b> |

**PROCESS TO CHECKOUT**

Hình 33: Check out

- Người dùng sẽ được chuyển tới trang thông tin giao hàng

#### BILLING ADDRESS

Enter your cupone code

NAME \*

Lê Trung Hiếu

ADDRESS \*

4 đường 18 phường Linh Chiểu quận Thủ Đức

PHONE \*

0974697842

EMAIL ADDRESS \*

leetrunghieu001@gmail.com

NOTE

Gọi điện thoại trước khi giao nhé shop <3

SEND

Hình 34: Thông tin giao hàng

➤ Router

```
Route::get('/show_checkout', 'CheckoutController@show_checkout');
```

*Router 9: Check out*

➤ Controller

```
public function show_checkout()
{
 $cate_product = Category::where('category_status', '0')->orderBy('category_name', 'asc')->get();
 $brand_product = Brand::where('brand_status', '0')->orderBy('brand_name', 'asc')->get();

 return view('pages.checkout.show_checkout')->with('category', $cate_product)->with('brand', $brand_product);
}
```

*Controller 13: Check out*

➤ Thông tin người nhận hàng sẽ được lưu bằng phương thức save\_checkout\_customer

```
public function save_checkout_customer(Request $request)
{
 $data = $request->validate([
 'shipping_name' => 'required',
 'shipping_address' => 'required',
 'shipping_email' => 'required',
 'shipping_phone' => 'required',
 'shipping_note' => 'required',
]);

 $shipping_id = new Shipping();
 $shipping_id->shipping_name = $data['shipping_name'];
 $shipping_id->shipping_address = $data['shipping_address'];
 $shipping_id->shipping_email = $data['shipping_email'];
 $shipping_id->shipping_phone = $data['shipping_phone'];
 $shipping_id->shipping_note = $data['shipping_note'];
 $shipping_id->save();
 Session::put('shipping_id', $shipping_id->shipping_id);
 return redirect('/payment');
}
```

*Controller 14: Lưu thông tin người dùng*

3. Sau khi điền thông tin và ấn “send” khách hàng sẽ được chuyển tới trang kiểm tra thanh và thanh toán

## Check cart

| Product                                                                                                                     | Price         | Quantity | Total         |
|-----------------------------------------------------------------------------------------------------------------------------|---------------|----------|---------------|
|  Supreme Aerial Tapestry Harrington Jacket | \$ 14,721,000 | 1        | \$ 14,721,000 |

## Select payment

**FORMAL PAYMENT**  
Select the one you want

Receive payment

**CART TOTAL**  
Final info

|              |                         |
|--------------|-------------------------|
| Subtotal     | \$ 14,721,000.00        |
| Shipping     | Free                    |
| <b>Total</b> | <b>\$ 14,721,000.00</b> |

**ORDER**

Hình 35: Kiểm tra và thanh toán

### ➤ Router

```
Route::get('/payment', 'CheckoutController@payment');
```

Router 10: Thanh toán

### ➤ Controller

```
public function payment()
{
 $cate_product = Category::where('category_status', '0')->orderBy('category_name', 'asc')->get();
 $brand_product = Brand::where('brand_status', '0')->orderBy('brand_name', 'asc')->get();
 return view('pages.checkout.payment')->with('category', $cate_product)->with('brand', $brand_product);
}
```

Controller 15: Thanh toán

➤ Phương thức order\_place:

Lấy thông tin bao gồm đơn hàng, chi tiết đơn hàng, phương thức thanh toán và lưu lại trên hệ thống chờ xử lý.

```
public function order_place(Request $request)
{
 // Insert payment
 $payment = new Payment();
 $payment->payment_method = $request->payment_option;
 $payment->payment_status = 'Pending...';
 $payment->save();

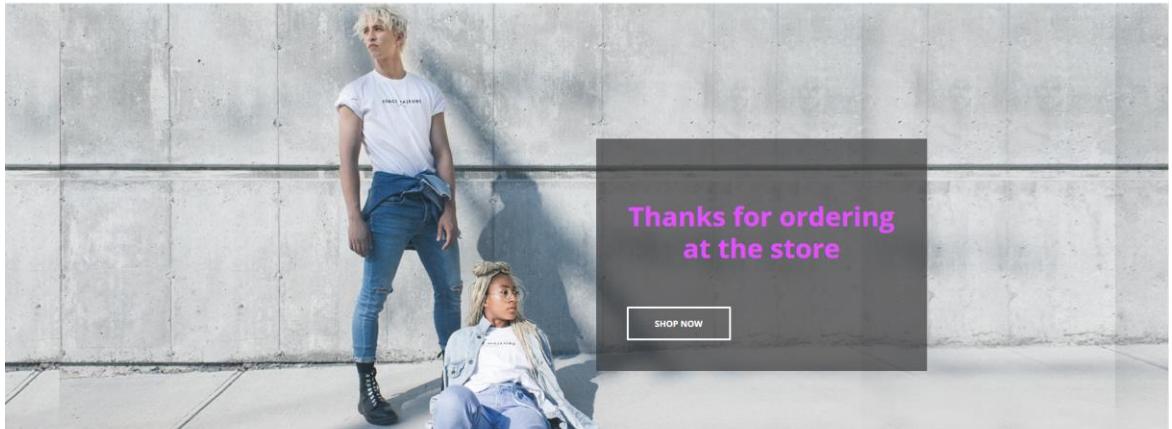
 // Insert order
 $order_id = new Order();
 $order_id->customer_id = Session::get('customer_id');
 $order_id->shipping_id = Session::get('shipping_id');
 $order_id->payment_id = $payment->payment_id;
 $order_id->order_total = Cart::subtotal();
 $order_id->order_status = 'Pending...';
 $order_id->save();

 // Insert order-detail
 $content = Cart::content();
 foreach ($content as $value_content) {
 $order_d_data = new OrderDetails();
 $order_d_data->order_id = $order_id->order_id;
 $order_d_data->product_id = $value_content->id;
 $order_d_data->product_name = $value_content->name;
 $order_d_data->product_price = $value_content->price;
 $order_d_data->product_sales_quantity = $value_content->qty;
 $order_d_data->save();
 }

 if ($payment->payment_method == 2) {
 echo 'Thanh toán thru atm';
 } else {
 Cart::destroy();
 $cate_product = Category::where('category_status', '0')->orderBy('category_name', 'asc')->get();
 $brand_product = Brand::where('brand_status', '0')->orderBy('brand_name', 'asc')->get();
 return view('pages.checkout.handcash')->with('category', $cate_product)->with('brand', $brand_product);
 }
 return redirect('/payment');
}
```

Controller 16: order

4. Sau khi chọn hình thức thanh toán và ấn order



Hình 36: Hoàn thành đặt hàng

## ➤ View Show\_checkout

```
<div class="cart-page-heading">
 <h5>Billing Address</h5>
 <p>Enter your cupone code</p>
</div>

<form action="{{URL::to('/save_checkout_customer')}}" method="post">
 @csrf
 <div class="row">
 <div class="col-md-12 mb-3">
 <label for="first_name">Name *</label>
 <input data-validation="length" data-validation-length="min1" data-validation-error-msg="Please enter your name" type="text" name="shipping_name" class="form-control" id="first_name" value="" required>
 </div>
 <div class="col-12 mb-3">
 <label for="street_address">Address *</label>
 <input data-validation="length" data-validation-length="min1" data-validation-error-msg="Please enter your address" type="text" name="shipping_address" class="form-control mb-3" id="street_address" value="">
 </div>
 <div class="col-12 mb-3">
 <label for="phone_number">Phone *</label>
 <input data-validation="length" data-validation-length="min1" data-validation-error-msg="Please enter your phone" type="number" name="shipping_phone" class="form-control" id="phone_number" min="0" value="">
 </div>
 <div class="col-12 mb-4">
 <label for="email_address">Email Address *</label>
 <input data-validation="length" data-validation-length="min1" data-validation-error-msg="Please enter your email" type="email" name="shipping_email" class="form-control" id="email_address" value="">
 </div>
 <div class="col-12 mb-4">
 <label for="email_address">Note</label>
 <textarea class="form-control font-note" name="shipping_note" id="" cols="30" rows="5"></textarea>
 </div>
 </div>
 <input type="submit" value="Send" name="send_order" class="btn karl-checkout-btn">
</form>
```

View 8: Check out

## ➤ View payment

```
<div class="col-12 col-md-6 col-lg-4">
 <div class="shipping-method-area mt-70">
 <div class="cart-page-heading">
 <h5>Formal payment</h5>
 <p>Select the one you want</p>
 </div>

 <div class="custom-control custom-radio mb-30">
 <input type="radio" id="customRadio1" name="payment_option" class="custom-control-input" value="1" required>
 <label class="custom-control-label d-flex align-items-center justify-content-between" for="customRadio1">
 Receive payment
 </label>
 </div>
 </div>
</div>

<div class="col-12 col-lg-4">
 <div class="cart-total-area mt-70">
 <div class="cart-page-heading">
 <h5>Cart total</h5>
 <p>Final info</p>
 </div>

 <ul class="cart-total-chart">
 Subtotal $ {{Cart::subtotal()}}
 <!-- Eco Tax $ {{Cart::tax()}} -->
 Shipping Free
 Total $ {{Cart::subtotal()}}

 <input type="submit" name="send_order" id="" value="Order" class="btn karl-checkout-btn">
 </div>
</div>
```

View 9: Thanh toán

➤ View handcash (sau khi order xong)

```
<section class="offer_area height-700 section_padding_100 bg-img" style="background-image: url({{url('public')}}/frontend/images/bg-img/bg-5.jpg);">
 <div class="container h-100">
 <div class="row h-100 align-items-end justify-content-end">
 <div class="col-12 col-md-8 col-lg-6">
 <div class="offer-content-area wow fadeInUp" data-wow-delay="1s">
 <h4 class="thanks-style">
 | Thanks for ordering at the store
 </h4>
 Shop Now
 </div>
 </div>
 </div>
 </div>
</section>
```

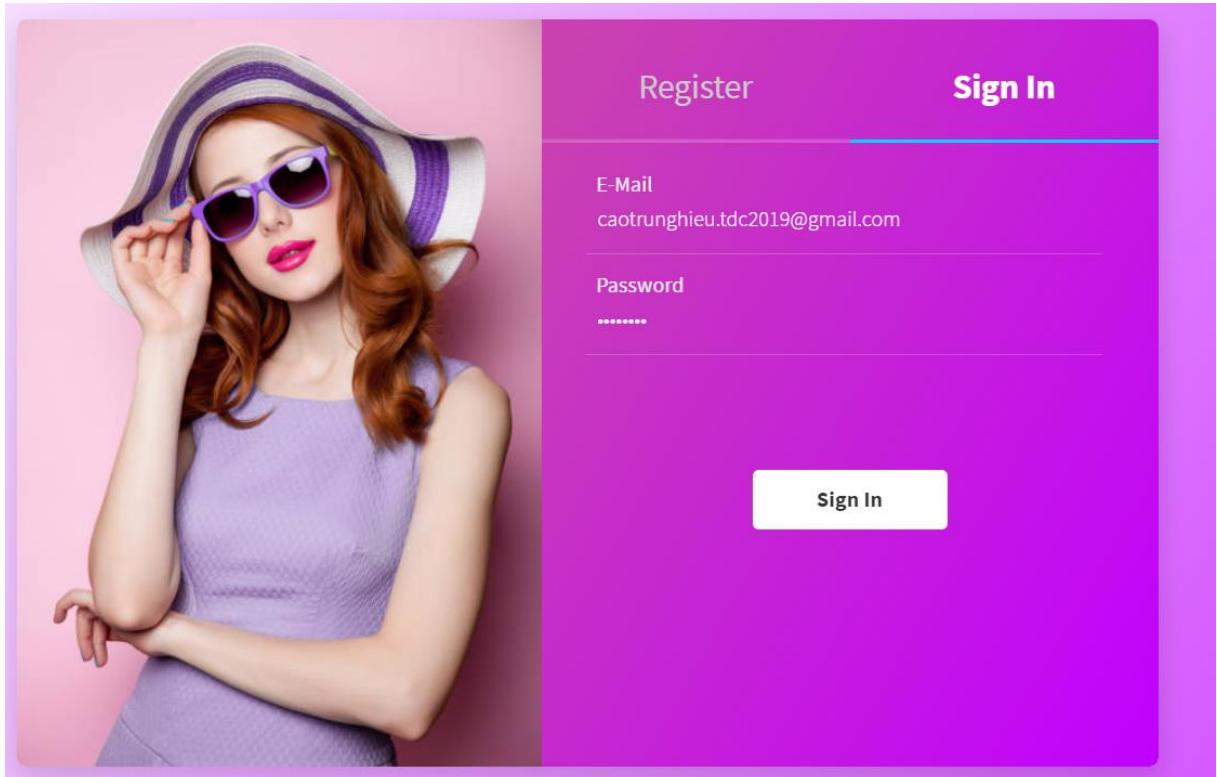
*View 10: Hoàn thành đặt hàng*

## 5. Chức Năng Đăng nhập/Đăng Ký/Đăng xuất

### .5.1 Chức năng đăng nhập (Sign in)

- Người dùng có thể đăng nhập thông qua nút Sign in trên Header Website (Nếu người dùng đăng nhập rồi thì nút đó sẽ trở thành nút Sign out)
- Khi người dùng ấn vào Nút Order ở giỏ hàng mà chưa đăng nhập thì cũng sẽ được chuyển qua trang đăng nhập (Để xác nhận ai đang đặt hàng)

#### .5.1.1 Màn hình đăng nhập



Hình 37: Màn hình đăng nhập

#### .5.1.2 Hướng đi:

- Khi người dùng nhấn Sign in, Website sẽ gửi tên E-Mail và Password lên Controller bằng phương thức Post (Để bảo mật).
- Login Controller sẽ lấy danh sách User trên Database về, và kiểm tra xem Tên E-Mail và Passwork (Đã mã hoá) có tồn tại hay không
- Nếu có thì hệ thống sẽ tạo 2 Session chứa id và name của người dùng, và chuyển người dùng qua trang chủ, cùng câu xin chào
- Nếu không thì Website sẽ trả về trang đăng nhập, và in ra thông báo tài khoản không đúng

### 5.1.3 Login Controller

```
public function login_customer(Request $request)
{
 //Lấy email và password view gửi lên
 $email = $request->email_account;
 $password = md5($request->password_account);
 //Thực hiện truy vấn, so sánh email và password
 $result = Customer::where('customer_email', $email)->where('customer_password', $password)->first();
 //Nếu truy vấn thành công thì lưu 2 Session lại, và trả về trang chủ
 if ($result) {
 Session::put('customer_id', $result->customer_id);
 Session::put('customer_name', $result->customer_name);
 return redirect('/homePage');
 }
 //Nếu truy vấn thất bại thì trả về trang Login và xuất ra câu sai Email hoặc Password
 return redirect('/login_checkout')->with('message', 'Wrong E-Mail or Password !');
}
```

Controller 17: Đăng nhập

### 5.1.4 Login Router

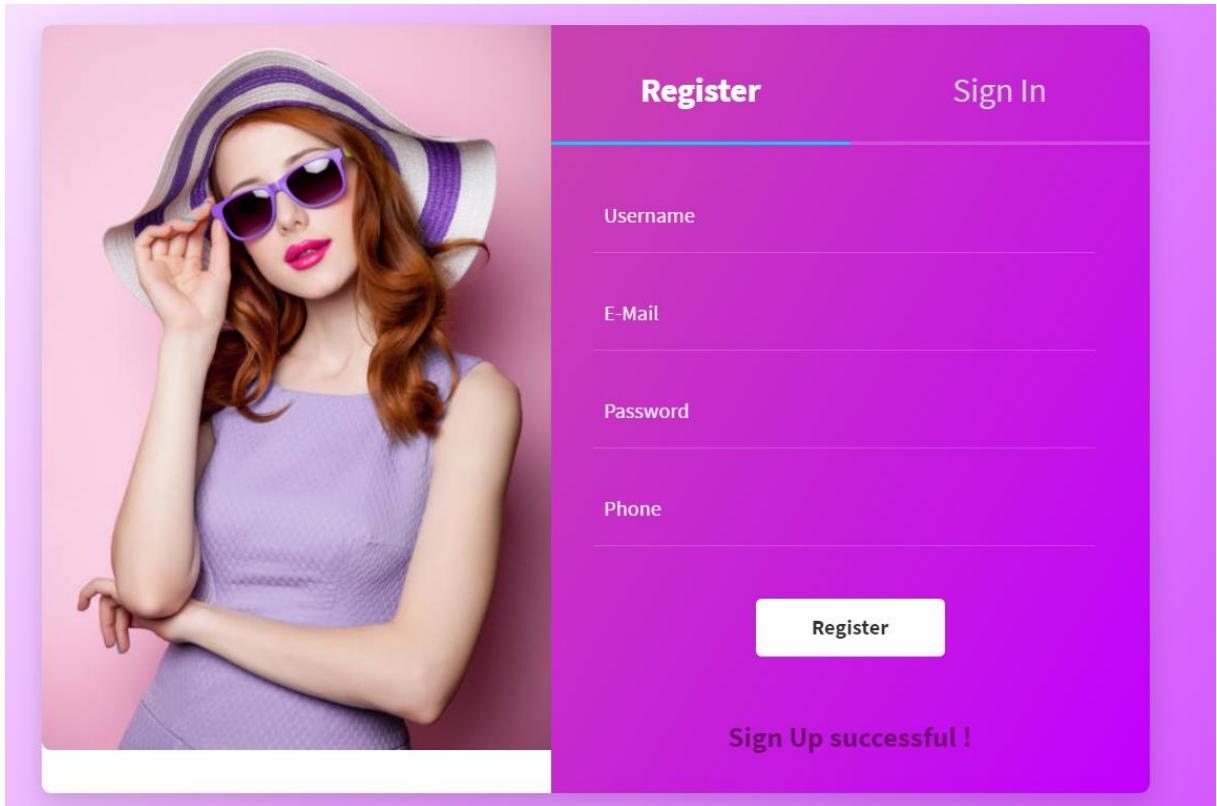
```
Route::post('/login_customer', 'CheckoutController@login_customer');
```

Router 11: Đăng nhập

## 5.2 Chức năng đăng ký (Register)

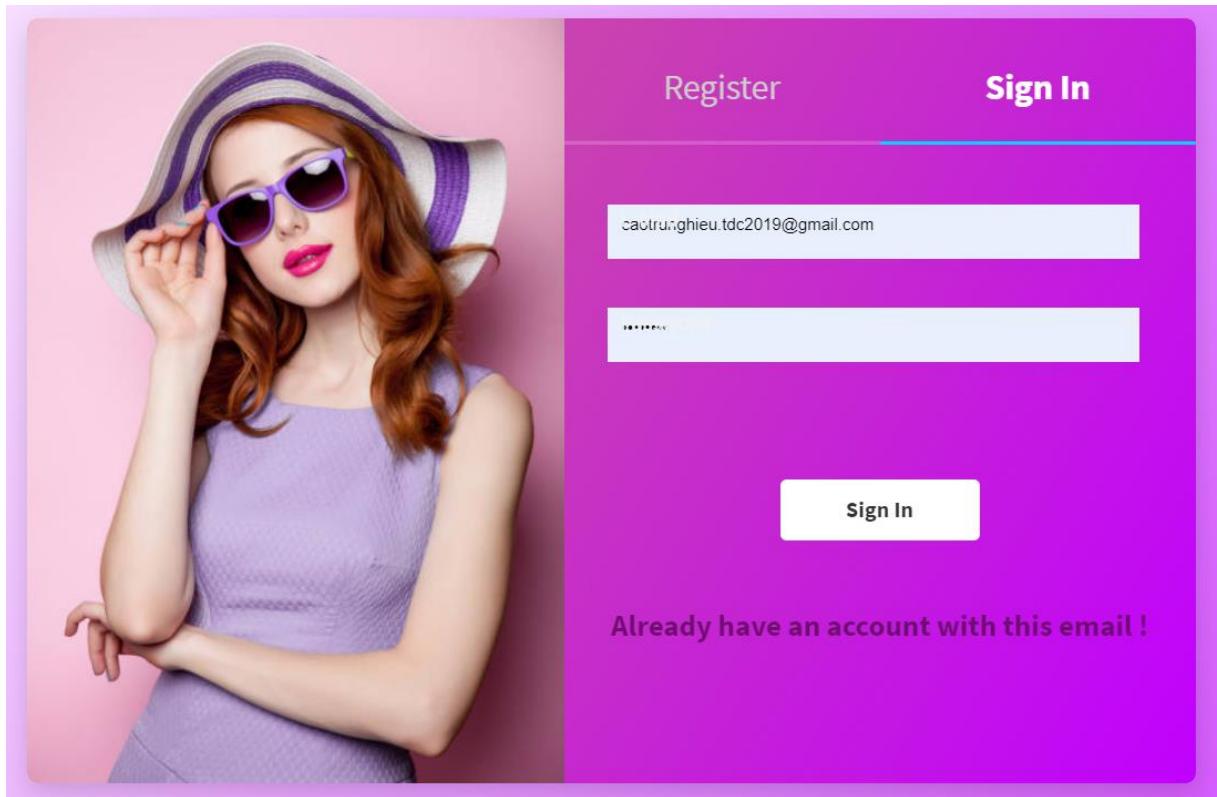
- Ở màn hình đăng nhập, nếu người dùng chưa có tài khoản thì người dùng có thể nhấn qua tag Register để tiến hành đăng ký
- Người dùng sẽ nhập các thông tin như là: Username, E-Mail, Password, Phone
- Khi nhấn vào nút “Register”, Website sẽ gửi toàn bộ các thông tin đó lên Controller
- Controller sẽ tiến hành truy vấn dựa trên E-Mail người dùng gửi lên.
- Nếu truy vấn trả về ít nhất 1 kết quả tức là E-Mail đó đã được người khác sử dụng. Hệ thống sẽ trả về trang Login và xuất thông báo E-Mail đã tồn tại
- Nếu truy vấn không trả về ít nhất 1 kết quả nào tức là E-Mail chưa được sử dụng. Hệ thống sẽ tạo thêm 1 account của khách hàng trên Database dựa trên thông tin khách hàng nhập vào. Đồng thời gửi Mail về cho khách thông báo đăng ký thành công, và trả về trang đăng nhập kèm thông báo đăng ký thành công

### .5.2.1 Giao diện đăng ký



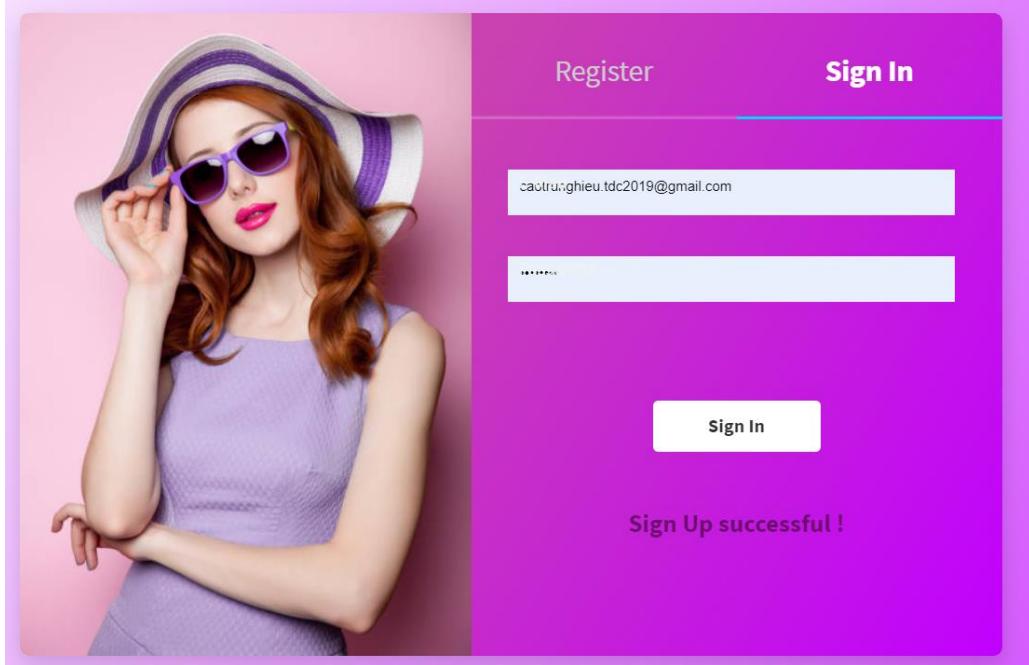
Hình 38: Màn hình đăng ký

### .5.2.2 Giao diện khi đăng ký dùng E-Mail đã có người khác sử dụng



Hình 39: Đăng ký trùng tài khoản

### 5.2.3 Giao diện khi đăng ký thành công



Hình 40: Đăng ký thành công

#### ➤ Register Controller

```
public function add_customer(Request $request)
{
 //Các thông tin về Email Admin
 $to_name = "Hieu Le";
 $to_email = $request->customer_email; //send to this email -- mail ad send to mail customer
 $data = array("name" => $to_email, "body" => "Bạn vừa tạo thành công 1 tài khoản trên Website bán hàng của chúng tôi. Chúc bạn một ngày tốt lành"); //body of mail

 Mail::send('pages.send_mail', $data, function ($message) use ($to_name, $to_email) {
 $message->to($to_email)->subject("Bạn vừa tạo 1 tài khoản trên Website bán hàng của chúng tôi !!"); //send this mail with subject
 $message->from($to_email, $to_name); //send from this mail
 });

 $data = $request->validate([
 'customer_name' => 'required',
 'customer_email' => 'required',
 'customer_password' => 'required',
 'customer_phone' => 'required',
]);
 //Tạo 1 Customer
 $customer = new Customer();
 $customer->customer_name = $data['customer_name'];
 $customer->customer_email = $data['customer_email'];
 $customer->customer_password = md5($data['customer_password']);
 $customer->customer_phone = $data['customer_phone'];
 //Thực hiện truy vấn để kiểm tra E-Mail
 $tbl_customer = Customer::where('customer_email', $request->customer_email)->get();
 //Nếu kết quả trả về khác 0 tức là E-Mail đã tồn tại trong Database
 //Xuất ra thông báo Khách hàng already have an account with this email
 if (count($tbl_customer) != 0) {
 return redirect('/login_checkout')
 ->with('message', 'Already have an account with this email !');
 }
 //Nếu không thì lưu thông tin vào DB và xuất ra thông báo Sign Up successful
 $customer->save();
 return redirect('/login_checkout')->with('message', 'Sign Up successful !');
}
```

Controller 18: Đăng kí

### 5.2.4 Registet Router

```
Route::post('/add_customer', 'CheckoutController@add_customer');
```

Router 12: Đăng kí

### .5.3 Chức năng đăng xuất (Sign out)

- Khi khách hàng đã đăng nhập, và nhấn vào nút “Sign out” trên Header thì Website sẽ gọi đến Sign out Controller
- Sign out Controller chỉ đơn giản là huỷ hết toàn bộ Session và trả về câu thông báo

#### .5.3.1 Nút Sign out



Hình 41: Đăng xuất

#### .5.3.2 Sign out Controller

```
public function logout_checkout()
{
 //Huỷ các Session
 Session::flush();
 //Truyền câu thông báo xuống View
 return redirect('/login_checkout')->with('message', 'Sign out successful!');
}
```

Controller 19: Đăng xuất

#### .5.3.3 Sign out Router

```
Route::get('/logout_checkout', 'CheckoutController@logout_checkout');
```

Router 13: Đăng xuất

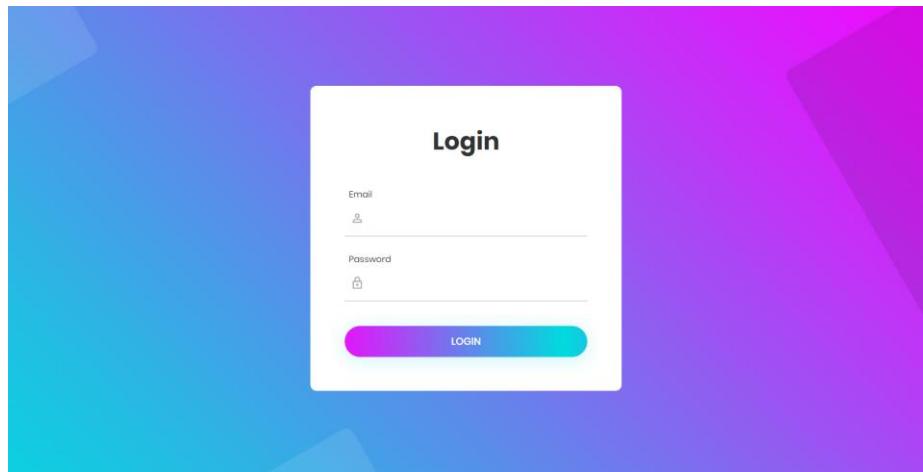
## 6. Chức Năng Quản Trị

### 6.1 Giới thiệu

- Đây là trang Website riêng dành cho người dùng quản trị.
- Người dùng có thể truy cập Website này bằng cách vào đường dẫn "URL/admin"
- Muốn vào được các tính năng quản trị, người dùng sẽ phải qua 1 cổng đăng nhập, và phải nhập đúng tài khoản quản trị thì mới vào giao diện quản trị được
- Người dùng quản trị có thể thực hiện hiện xem, xem, xoá, sửa các Brand, Category, Product. Những thông tin thay đổi này sẽ được thay đổi trên Database
- Người dùng quản trị có thể xem, xoá và duyệt đơn hàng do khách hàng đặt

### 6.2 Đăng nhập

Đây là màn hình khi người dùng chưa đăng nhập



Hình 42: Đăng nhập admin

- Khi nhấn vào nút đăng nhập, Website sẽ truyền tên E-Mail và Password lên Router, Router sẽ truyền lên Controller, Controller sẽ tiến hành truy vấn Database dựa trên E-Mail và chuỗi Password đã mã hoá mà người dùng gửi lên
- Nếu đăng nhập thành công, Controller sẽ lưu 2 Session lưu Admin name và Admin id
- Nếu đăng nhập thất bại thì trả về trang admin đăng nhập và xuất ra thông báo "Wrong email or password!"
- Controller

```
public function dashboard(Request $request)
{
 // Lấy các thông tin Router gửi lên
 $admin_email = $request->admin_email;
 $admin_password = md5($request->admin_password);
 // Truy van du lieu
 $result = Admin::where('admin_email', $admin_email)->where('admin_password', $admin_password)->first(); //First lay duy nhat
 // so sanh ket qua voi db neu co tra ve /dashboard
 if ($result) {
 Session::put('admin_name', $result->admin_name);
 Session::put('admin_id', $result->admin_id);
 return redirect('/dashboard');
 }
 // neu sai xuất ra thông báo và trả về /admin login
 return redirect('/admin')->with('message', 'Wrong email or password !');
}
```

Controller 20: Đăng nhập admin

## ➤ Router

```
Route::post('/admin-dashboard', 'AdminController@dashboard');
```

Router 14: Đăng nhập admin

## ➤ View

```
<form class="login100-form validate-form" action="{{URL::to('/admin-dashboard')}}" method="POST">
 @csrf
 ...

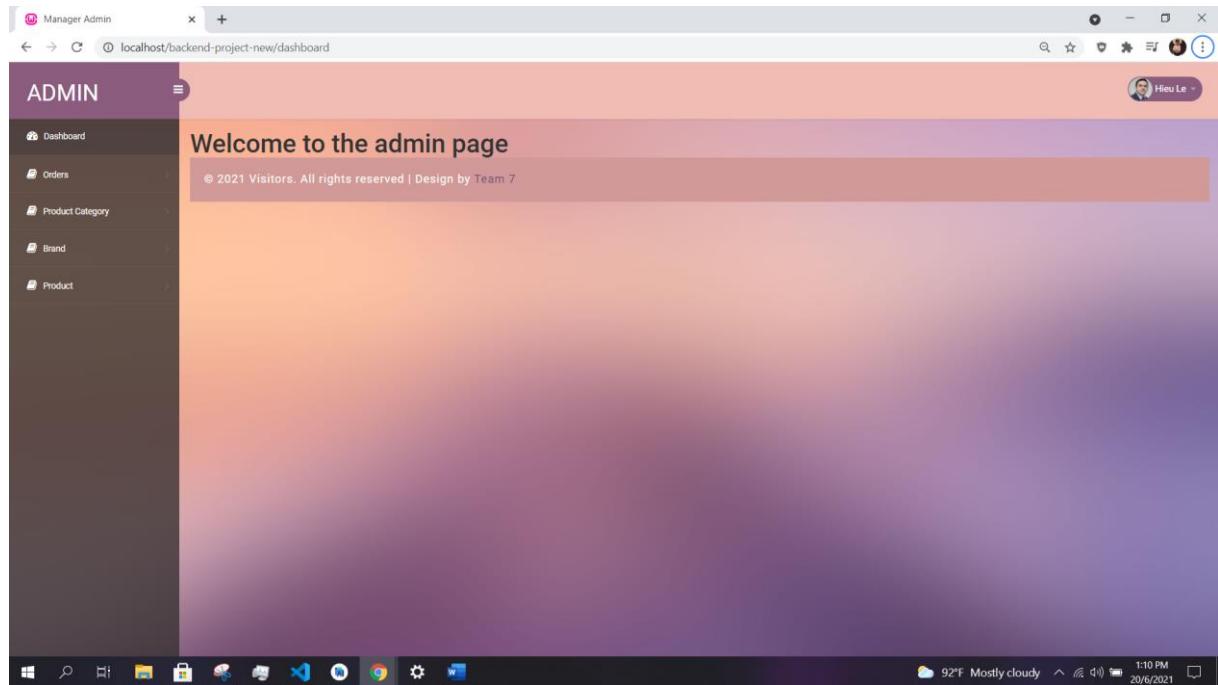
 <?php
 //khai bao $message
 $message = Session::get('message');
 // neu co thi echo va gan $message = null
 if ($message) { ...
 }
 ?>
 <div class="wrap-input100 validate-input m-b-23" data-validate="Username is required">...
 </div>

 <div class="wrap-input100 validate-input" data-validate="Password is required">...
 </div>
 <div class="text-left-right" style="display: flex;">...
 </div>
 <div class="container-login100-form-btn">
 <div class="wrap-login100-form-btn">
 <div class="login100-form-bgbtn"></div>
 <button class="login100-form-btn">
 Login
 </button>
 </div>
 </div>
</form>
```

View 11: Đăng nhập admin

### .6.3 Giao diện chính trang quản trị

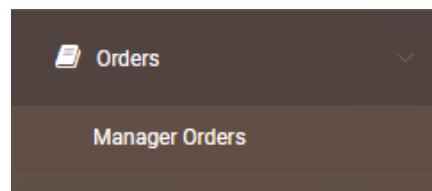
- Khi người dùng đăng nhập thành công, Website sẽ vào giao diện quản trị, và dùng Session Admin nam để diện tên Admin bên góc phải Header
- Các chức năng quản trị như quản trị Order, Category, Brand và Product sẽ làm ở các tag menu trái



Hình 42: Giao diện trang quản trị

### .6.4 Quản trị đơn hàng

- Người dùng chọn Manager Orders để vào mục quản trị đơn hàng



Hình 43: Quản trị đơn hàng

- Màn hình quản trị đơn hàng gồm các thông tin:
  1. Order Name: Tên tài khoản khách hàng đặt hàng
  2. Customer ID: mã khách hàng
  3. Total: tổng giá trị đơn hàng
  4. Status: trạng thái đơn hàng
  5. Vùng tiêm kiểm đơn hàng theo mã khách hàng, kiểm tra và xoá đơn hàng

## ➤ Màn hình quản trị đơn hàng

Hình 44: Màn hình quản trị đơn hàng

## ➤ Hàm lấy đơn hàng và hiển thị

```
public function manager_order()
{
 $this->AuthLogin();

 $all_order = Order::addSelect([
 'customer_name' => Customer::select('customer_name')
 ->whereColumn('customer_id', 'tbl_order.customer_id')
 ->orderByDesc('order_id')
])->paginate(5);

 $manager_order = view('admin.manager_order')->with('all_order', $all_order);
 return view('admin_layout')->with('admin.manager_order', $manager_order);
}
```

Controller 21: Quản trị đơn hàng

## ➤ Màn hình tìm kiếm

Hình 45: Tìm kiếm đơn hàng

- Hàm tìm kiếm đơn hàng theo mã khách hàng và hiển thị

```
// search
public function search_order(Request $request)
{
 $keywords = $request->keywords_submit;

 $cate_product = Category::where('category_status', '0')->orderBy('category_name', 'asc')->get();
 $brand_product = Brand::where('brand_status', '0')->orderBy('brand_name', 'asc')->get();

 $search_order = Order::where('customer_id', 'like', '%' . $keywords . '%')->orderBy('order_id', 'desc')->get();
 $search_customer = Customer::where('customer_id', 'like', '%' . $keywords . '%')->get();

 return view('/admin/search_order')->with('category', $cate_product)
 ->with('brand', $brand_product)
 ->with('search_order', $search_order)
 ->with('search_customer', $search_customer);
}
```

*Controller 22: Tìm kiếm đơn hàng*

- Sau khi người dùng quản trị ấn vào kiểm tra đơn hàng màn hình sẽ hiển thị 3 bảng thông tin: thông tin khách hàng, thông tin giao hàng và thông tin đơn hàng

CUSTOMER INFORMATION	
Customer name	Phone
hieule	974697842

SHIPPING INFORMATION			
Shipper name	Address	Phone	Note
Lê Trung Hiếu	4 đường 18 phường Linh Chiểu quận Thủ Đức	0974697842	Gọi điện thoại trước khi giao nhé shop <3

*Hình 46: Thông tin khách hàng*

ORDERS DETAILS			
Product name	Quantity	Price	Total price
Supreme Aerial Tapestry Harrington Jacket	1	\$ 14,721,000	\$ 14,721,000
Total:	\$ 14,721,000.00		
Note:	<input type="text" value="Được phép kiểm tra hàng trước khi thanh toán !!"/> <span style="border: 1px solid #007bff; padding: 2px;">Done</span>		

*Hình 47: Chi tiết đơn hàng*

- Sau khi kiểm tra giao hàng và giao hàng thì người dùng có thể ghi chú đơn hàng và nhấn “done” để kết thúc kiểm tra
- Màn hình sau khi kết thúc kiểm tra

LIST ORDERS			
Order Name	Customer ID	Total	Status
hieule	22	14,721,000.00	Được phép kiểm tra hàng trước khi thanh toán !!
hieule	22	185,000.00	Pending...
Show 2			

Hình 48: Màn hình sau khi kiểm tra đơn hàng

- Hàm cập nhập đơn hàng

```
public function update_order($order_id, Request $request)
{
 $this->AuthLogin();

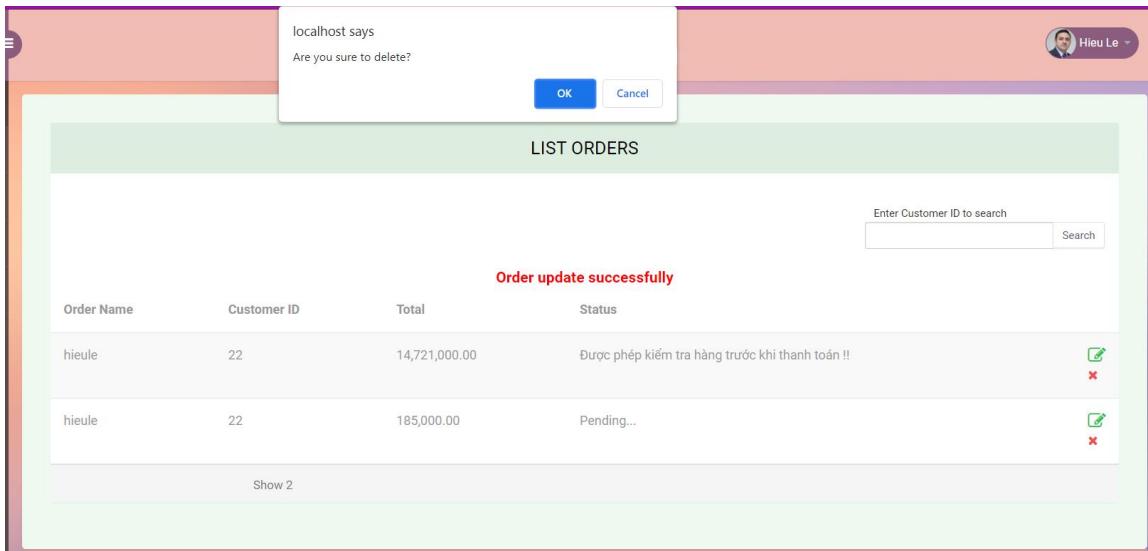
 $data = $request->all();
 $order = Order::find($order_id);
 $order->order_status = $data['order_status'];

 $order->save();

 return redirect('manager_order')->with('message', 'Order update successfully ');
}
```

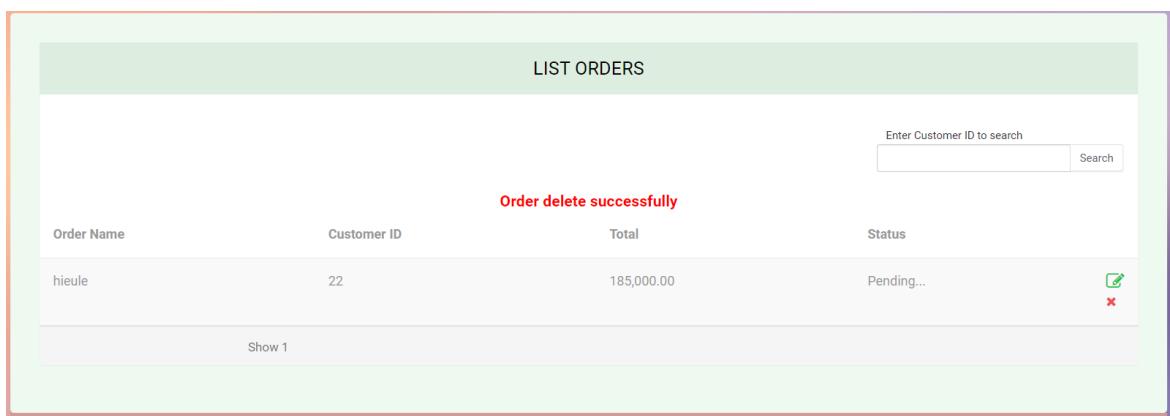
Controller 23: Cập nhập đơn hàng

- Nếu người dùng muốn xoá đơn hàng sẽ có thông báo hiện lên



Hình 49: Màn hình xoá đơn hàng

- Màn hình sau khi xoá



Hình 50: Màn hình sau khi xoá đơn hàng

- Hàm xoá đơn hàng

```
public function delete_order($order_id)
{
 $this->AuthLogin();
 $order = Order::find($order_id);
 $order_details = OrderDetails::where('order_id', $order_id)->delete();
 $shipping = Shipping::find($order->shipping_id);
 $payment = Payment::find($order->payment_id);

 $shipping->delete();
 $payment->delete();
 $order->delete();

 return redirect('manager_order')->with('message', 'Order delete successfully ');
}
```

Controller 24: Xoá đơn hàng

➤ Router:

```
// Orders
Route::get('/manager_order', 'CheckoutController@manager_order');
Route::get('/view_order/{order_id}', 'CheckoutController@view_order');
Route::get('/delete_order/{rowId}', 'CheckoutController@delete_order');
Route::post('/update_order/{order_id}', 'CheckoutController@update_order');
```

Router 15: Quản trị đơn hàng

➤ View quản trị đơn hàng

```
<table class="table table-striped b-t b-light">
 <thead>
 <tr>
 <th>Order Name</th>
 <th>Customer ID</th>
 <th>Total</th>
 <th>Status</th>
 <th style="width:30px;"></th>
 </tr>
 </thead>
 <tbody>
 @foreach ($all_order as $order)
 <tr>
 <td>{{ $order->customer_name }}</td>
 <td>{{ $order->customer_id }}</td>
 <td>{{ $order->order_total }}</td>
 <td>{{ $order->order_status }}</td>
 <td>
 order_id) }}" class="active styling-edit" ui-toggle-class="">
 <i class="fa fa-pencil-square-o text-success text-active"></i>

 order_id) }}"
 class="active styling-edit" ui-toggle-class="">
 <i class="fa fa-times text-danger text"></i>

 </td>
 </tr>
 @endforeach
 </tbody>
</table>
</div>
<footer class="panel-footer">
 <div class="row">
 <div class="col-sm-5 text-center">
 <small class="text-muted inline m-t-sm m-b-sm">Show {{count($all_order)}} </small>
 </div>
 <div class="col-sm-7 text-right text-center-xs">
 <ul class="pagination pagination-sm m-t-none m-b-none">
 {!! $all_order->render() !!}

 </div>
 </div>
</footer>
```

View 12: Quản trị đơn hàng

➤ View customer information

```
<div class="table-agile-info">
 <div class="panel panel-default">
 <div class="panel-heading">
 Customer Information
 </div>
 <div class="table-responsive">
 <table class="table table-striped b-t b-light">
 <thead>
 <tr>
 <th>Customer name</th>
 <th>Phone</th>
 </tr>
 </thead>
 <tbody>
 <tr>
 <td>{$customer->customer_name}</td>
 <td>{$customer->customer_phone}</td>
 </tr>
 </tbody>
 </table>
 </div>
 </div>
</div>
```

View 13: Thông tin khách hàng

➤ View shipping information

```
<div class="table-agile-info">
 <div class="panel panel-default">
 <div class="panel-heading">
 Shipping Information
 </div>
 <div class="table-responsive">
 <table class="table table-striped b-t b-light">
 <thead>
 <tr>
 <th>Shipper name</th>
 <th>Address</th>
 <th>Phone</th>
 <th>Note</th>
 </tr>
 </thead>
 <tbody>
 <tr>
 <td>{$shipping->shipping_name}</td>
 <td>{$shipping->shipping_address}</td>
 <td>{$shipping->shipping_phone}</td>
 <td>{$shipping->shipping_note}</td>
 </tr>
 </tbody>
 </table>
 </div>
 </div>
</div>
```

View 14: Thông tin giao hàng

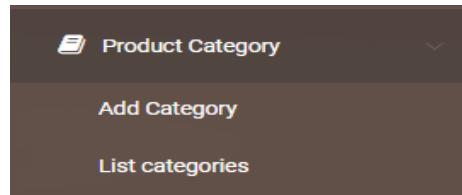
## ➤ View order details

```
<table class="table table-striped b-t b-light">
 <thead>
 <tr>
 <th>Product name</th>
 <th>Quantity</th>
 <th>Price</th>
 <th>Total price</th>
 <th style="width:30px;"></th>
 </tr>
 </thead>
 <tbody>
 @foreach ($order_details as $value)
 <tr>
 <td>{$value->product_name}</td>
 <td>{$value->product_sales_quantity}</td>
 <td>$ {$number_format($value->product_price)}</td>
 <td>$ {$number_format($value->product_price*$value->product_sales_quantity)}</td>
 </tr>
 @endforeach
 <tr>
 <th>Total:</th>
 <td></td>
 <td></td>
 <td>$ {$order->order_total}</td>
 </tr>
 <form action="{{URL::to('/update_order/'.$value->order_id)}}" method="post">
 @csrf
 <tr>
 <th>
 <div class="form-group">
 <label for="exampleInputEmail1">Note:</label>
 <input type="text" class="form-control form-group" value="{{$value->order_status}}" name="order_status">
 </div>
 </th>
 <td>
 <div class="done">
 <button type="submit" name="update_order" class="btn btn-info">Done</button>
 </div>
 </td>
 <td></td>
 <td></td>
 </tr>
 </form>
 </tbody>
</table>
```

View 15: Chi tiết đơn hàng

### .6.5 Quản trị Category

- Người dùng quản trị đã đăng nhập có thể nhấp vào Tag “Product Category” ở Menu bên trái để chọn các tính năng quản trị Category
- Tag “Add Category” dùng để thêm 1 Category mới
- Tag “List Category” dùng để xem danh sách Category đang tồn tại, xoá, sửa các Category



Hình 51: Quản trị loại

#### .6.5.1 Thêm Category

- Khi người dùng nhấp vào Tag “Add Category”, Website sẽ chuyển qua giao diện nhập thông tin cho Category mới
- Text Field “Category Name” là tên của Category mới
- Text Field “Category Description” là mô tả của Category mới
- Choose file “Category Image” là file ảnh do người dùng chọn
- Combobox “Display” sẽ quyết định Category đó có được hiển thị ra hay không
- Tất cả các ô Input đều bắt buộc người dùng nhập

The screenshot shows a web form titled "ADD CATEGORY PRODUCT". It contains four main input fields: "Category Name" (text input with placeholder "Please enter category name"), "Category Description" (text area), "Category Image" (file input with placeholder "Chọn tệp Không có tệp nào được chọn"), and a "Display" dropdown menu currently set to "Show". At the bottom is a blue "Add" button.

Hình 52: Màn hình quản trị loại

- Khi người dùng vào Nút Add, Website sẽ gửi các thông tin đó lên add category controller
- Add category controller sẽ nhận các thông tin người dùng gửi lên
- Controller cũng sẽ kiểm tra xem File người dùng chọn có phải là ảnh hay không (Dựa vào đuôi của file)
- Hàm kiểm tra ảnh

```
//Hàm kiểm tra hình ảnh
function checkImage($fileName)
{
 //Cắt tên file truyền vào dựa vào dấu '.'
 $substr = explode('.', $fileName);
 //Nếu biến $substr trả ra 0 phần tử tức là file chọn không có tên => Không phải ảnh
 //Nếu biến $substr trả ra 1 phần tử tức là file chọn không có đuôi file => Không phải ảnh
 if (count($substr) == 0 || count($substr) == 1) {
 return false;
 //Nếu phần tử cuối cùng của biến $substr trùng với các định dạng đuôi hình ảnh=>File là ảnh
 } else if (
 strtolower($substr[count($substr) - 1]) == 'png' ||
 strtolower($substr[count($substr) - 1]) == 'jpg' ||
 strtolower($substr[count($substr) - 1]) == 'gif' ||
 strtolower($substr[count($substr) - 1]) == 'bmp'){
 return true;
 }
 //Nếu không phải thì đó không phải là ảnh
 return false;
}
```

Controller 25: Kiểm tra ảnh

## ➤ Add Category Controller

```
// Add
public function save_category_product(Request $request)
{
 $this->AuthLogin();
 //Lấy tất cả thông tin View truyền lên
 $data = $request->all();
 //Tạo ra 1 Category mới
 $category = new Category();
 $category->category_name = $data['category_product_name'];
 $category->category_desc = $data['category_product_desc'];
 $category->category_status = $data['category_product_status'];
 //Đặt file mà người dùng chọn
 $get_image = $request->file('category_image');
 //Nếu mà file có tồn tại thì mới tiến hành kiểm tra
 if ($get_image) {
 //Lấy tên file
 $get_name_image = $get_image->getClientOriginalName();
 $name_image = current(explode('.', $get_name_image));
 $new_image = $name_image . '.' . $get_image->getClientOriginalExtension();
 //Nếu như không phải là ảnh thì không cho thêm
 if (!$this->checkImage($new_image)) {
 return redirect('/add_category_product')->with('message', 'Can\'t add Category because The file is not a image !');
 }
 //Nếu file là ảnh thì thêm category mới vào database
 $get_image->move('public/uploads/category', $new_image);
 $category->category_image = $new_image;
 $category->save();
 return redirect('/add_category_product')->with('message', 'Add category success !');
 }
 //Nếu file không tồn tại thì tự động gán 1 tên ảnh và lưu vào database
 $category->category_image = 'NoImage.jpg';
 $category->save();
 return redirect('/add_category_product')->with('message', 'Add category success !');
}
```

*Controller 26: Thêm loại sản phẩm*

## ➤ Add Category Router

```
Route::post('/save_category_product', 'CategoryProduct@save_category_product');
```

*Router 16: Thêm loại sản phẩm*

## ➤ Add Category View

```
<! -- Phải thêm enctype="multipart/form-data" mỗi truyền file lên Controller được -->
<form role="form" enctype="multipart/form-data" method="POST" action="{{URL::to('/save_category_product')}}">
 @csrf
 <div class="form-group">
 <label for="exampleInputEmail1">Category Name</label>
 <input required data-validation="length" data-validation-length="min1" data-validation-error-msg="Please enter category name" type="text" class="form-control" name="category_product_name" placeholder="Enter category name">
 </div>
 <div class="form-group">
 <label for="exampleInputPassword1">Category Description</label>
 <textarea style="resize: none;" required rows="5" class="form-control" name="category_product_desc" id="exampleInputPassword1" placeholder="Enter category description"></textarea>
 </div>
 <div class="form-group">
 <label for="exampleInputEmail1">Category Image</label>
 <input type="file" class="form-control" name="category_image" required>
 </div>
 <div class="form-group">
 <label for="exampleInputPassword1">Display</label>
 <select name="category_product_status" class="form-control input-sm m-bot15">
 <option value="0">Show</option>
 <option value="1">Hide</option>
 </select>
 </div>
 <button type="submit" name="add_category_product" class="btn btn-info">Add</button>
</form>
```

*View 16: Thêm loại sản phẩm*

### .6.5.2 Xem danh sách Category

- Người dùng nhấn vào tag “List Categories” sẽ chuyển qua Giao diện danh sách Category, cùng với các thông tin như tên, hình ảnh, Display, Cùng với 2 Icon Update và Delete Category đó
- Phía trên cùng góc phải là Form tìm kiếm Category theo tên
- Một trang chỉ được có tối đa 5 Category, nếu có quá 5 Category thì Website sẽ phân ra các trang phía sau

Category Name			Category Image	Display	
Hat					 
Short					 
Jean					 
Pants					 
Shoes					 

« 1 2 »

Hình 53: Danh sách loại sản phẩm

- Nếu category\_status là 0 thì Display Category sẽ hiển thị nút like
- Nếu category\_status là 1 thì Display Category sẽ hiển thị nút dislike
- Có thể trực tiếp thay đổi trạng thái Show, Hidden bằng cách nhấp trực tiếp vào Icon Like
- ⇒ Dislike và ngược lại, đồng thời cập nhật category\_status vào Database

```
@if($cate_pro->category_status == 0)
| category_id)}}">
@else
| category_id)}}">
@endif
```

View 17: Kiểm tra trạng thái loại sản phẩm

- Show list Category Controller

```
// Get all Category product
public function all_category_product()
{
 $this->AuthLogin();
 //Lấy 5 category trong 1 trang
 $all_category_product = Category::orderBy('category_id', 'DESC')->paginate(5);
 $manager_category_product = view('admin.all_category_product')->with('all_category_product', $all_category_product);
 return view('admin_layout')->with('admin.all_category_product', $manager_category_product);
}
```

Controller 27: Lấy loại sản phẩm

- Show list Category Router

```
Route::get('/all_category_product', 'CategoryProduct@all_category_product');
```

*Router 17: Lấy loại sản phẩm*

#### .6.5.3 Tìm kiếm Category theo tên

- Người dùng có thể tìm kiếm Category theo tên bằng cách viết từ khoá vào ô Text Field “Enter Name of Category to search” và nhấn vào Button Search
- Khi nhấn vào Button Search thì Website sẽ truyền từ khoá đó lên Controller, Controller sẽ thực hiện truy vấn dựa trên từ khoá website gửi lên, và trả kết quả về View để View duyệt
- Vd Search với từ khoá “SH”

We found 4 Brands have keyword "Sh"			
Category Name	Category Image	Display	
Short			
Shoes			
T-shirt			
Shirt			

*Hình 54: Hiển thị sản phẩm tìm kiếm theo loại*

- Search Category Controller

```
public function search_category(Request $request)
{
 //Nhận keyword người dùng viết vào
 $keywords = $request->keywords_submit;
 //Thực hiện truy vấn dựa trên keyword
 $search_category = Category::where('category_name', 'like', '%' . $keywords . '%')->orderBy('category_id', 'desc')->get();
 //Truyền danh sách đã truy vấn về View
 return view('/admin/search_category')->with('search_category', $search_category);
}
```

*Controller 28: Tìm kiếm loại sản phẩm*

- Search Category Router

```
Route::post('/search_category', 'CheckoutController@search_category');
```

*Router 18: Tìm kiếm loại sản phẩm*

## ➤ Search Category View

```

@foreach ($search_category as $key => $cate_pro)
| | | | |
| --- | --- | --- | --- |
| {{ $cate_pro->category_name }} | Category Image | <?php if ($cate_pro->category_status == 0) { ?> category_id)}}"> <?php } else { ?> category_id)}}"> <?php } ?> | |

@endforeach

```

View 18: Tìm kiếm loại sản phẩm

### .6.5.4 Xoá Category

- Ở List Category, người dùng có thể nhấn vào nút dấu x ở cuối mỗi hàng Catrgory trong danh sách
- Khi nhấn vào, Website sẽ gửi id của Category đó lên Delete Category Controller
- Controller sẽ kiểm tra cách sản phẩm thuộc Category đó, nếu còn sản phẩm thì không xoá Category đó, và đồng in in ra thông báo không xoá Category đó được
- Nếu không còn sản phẩm thì xoá Category đó, và in ra thông báo xoá thành công
- Delete Category Controller

```

// delete
public function delete_category_product($category_product_id)
{
 $this->AuthLogin();
 //Tìm kiếm Category dựa vào id truyền lên
 $category = Category::find($category_product_id);
 //Thực hiện truy vấn sản phẩm dựa vào category_id = id truyền lên
 $product_by_category = DB::table('tbl_product')->where('category_id', $category_product_id)->get();
 //Nếu hết sản phẩm đó rồi thì mới xoá
 if (count($product_by_category) == 0) {
 // delete image
 $destinationPath = 'public/uploads/category/' . $category->category_image;
 if (file_exists($destinationPath)) {
 unlink($destinationPath);
 }
 $category->delete();
 return redirect('all_category_product')->with('message', 'Product category delete successfully');
 }
 return redirect('all_category_product')->with('message', 'Cannot delete Category because this category has products');
}

```

Controller 29: Xoá loại sản phẩm

## ➤ Delete Category Router

```
Route::get('/delete_category_product/{category_product_id}', 'CategoryProduct@delete_category_product');
```

Router 19: Xoá loại sản phẩm

## ➤ Delete Category View

```

<a onclick="return confirm('Are you sure to delete?')"
 href="{{URL::to('/delete_category_product/'.$cate_pro->category_id)}}"
 class="active styling-edit" ui-toggle-class="">
 <i class="fa fa-times text-danger text" style="color: #c00;"></i>


```

View 19: Xoá loại sản phẩm

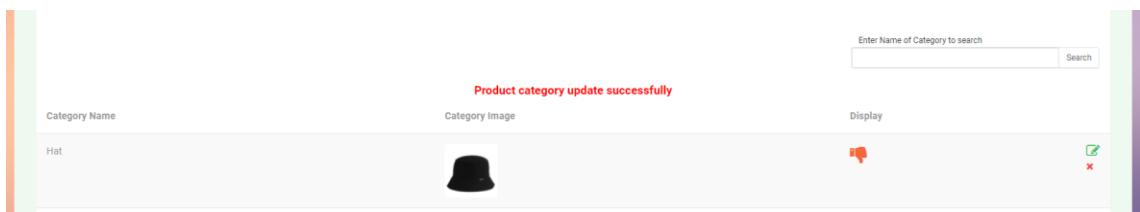
#### .6.5.5 Update Category

- Ở List Category, người dùng có thể nhấn vào Icon Update ở cuối mỗi hàng Catrgory trong danh sách
- Khi nhấn vào, Website sẽ gửi id của Category đó lên Update Category Controller
- Controller sẽ trả về View thông tin của Category đó về view
- View sẽ điền các thông tin của Category vào các ô input
- Khi nhấn Submit, Website sẽ gửi các thông tin về update Controller (Bao gồm cả id)
- Controller cũng sẽ kiểm tra người dùng có chọn file mới hay không
- Nếu người dùng không chọn file mới, Controller sẽ không thay đổi giá trị tên hình
- Nếu người dùng chọn file mới, Controller sẽ kiểm tra file mới có phải là hình hay không, nếu file là hình thì Controller sẽ Upload hình mới, và xoá hình cũ, nếu file không phải là hình, hệ thống sẽ không update hình ảnh
- Giao diện khi nhấn vào Update 1 Category

The screenshot shows a web-based form titled "EDIT CATEGORY PRODUCT". The form has three main sections: "Category Name" with a text input containing "Hat", "Category Image" with a file input labeled "Chọn tệp Không có tệp nào được chọn" and a preview area showing a small black silhouette of a hat, and "Category Description" with a text input containing "Hat". At the bottom is a blue "Update category" button.

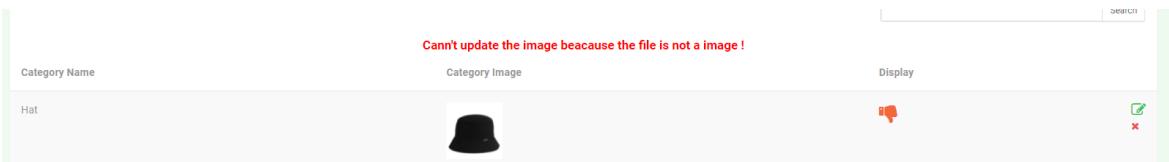
Hình 55: Sửa loại sản phẩm

- Nếu Update thành công



Hình 56: Cập nhập loại sản phẩm thành công

- Nếu Update mà người dùng chọn file không phải là hình



Hình 57: Cập nhập loại sản phẩm không thành công

## ➤ Update Category Controller

```
public function update_category_product($category_product_id, Request $request) > edit_category_product
{
 $this->AuthLogin();
 //Lấy tất cả thông tin Updatr Website gửi lên
 $data = $request->all();
 //Tìm kiếm Category dựa vào Id gửi lên
 $category = Category::find($category_product_id);
 //Update các thông tin (Nhưng chưa lưu)
 $category->category_name = $data['category_product_name'];
 $category->category_desc = $data['category_product_desc'];
 //Lấy thông tin file
 $get_image = $request->file('category_image');
 //Nếu file có tồn tại
 if ($get_image) {
 //Lấy tên file
 $get_name_image = $get_image->getClientOriginalName();
 $name_image = current(explode('.', $get_name_image));
 $new_image = $name_image . '.' . $get_image->getClientOriginalExtension();
 //Kiểm tra file có phải là ảnh hay không
 //Nếu file không phải là hình thì không update và upload hình
 if (!$this->checkImage($new_image)) {
 $category->save();
 return redirect('all_category_product')->with('message', 'Can\'t update the image because the file is not a image !');
 }
 //Nếu file là hình thì update và upload hình, và xoá hình cũ
 $destinationPath = 'public/uploads/category/' . $category->category_image;
 if (file_exists($destinationPath)) {
 unlink($destinationPath);
 }
 $get_image->move('public/uploads/category', $new_image);
 $category->category_image = $new_image;
 $category->save();
 return redirect('all_category_product')->with('message', 'Product category update successfully ');
 }
 //Nếu không chọn file thì không update và upload hình
 $category->save();
 return redirect('all_category_product')->with('message', 'Product category update successfully ');
}
```

Controller 30: Cập nhập loại sản phẩm

## ➤ Update Category Router

```
Route::post('/update_category_product/{category_product_id}', 'CategoryProduct@update_category_product');
```

Router 20: Cập nhập loại sản phẩm

## ➤ Update Category View

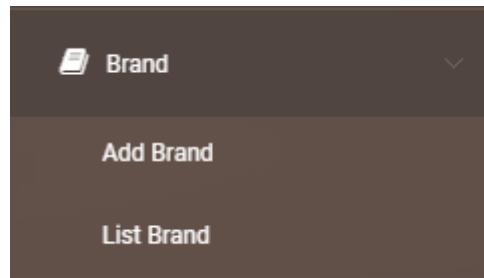
```
<!-- Phải thêm enctype="multipart/form-data" thì mới Upload hình lên Controller được -->
<form enctype="multipart/form-data" role="form" method="POST" action="{{URL::to('/update_category_product/'.$edit_value->category_id)}}>
 @csrf
 <div class="form-group">
 <label for="exampleInputEmail1">Category Name</label>
 <input type="text" class="form-control" value="{{ $edit_value->category_name}}" name="category_product_name" >
 </div>
 <div class="form-group">
 <label for="exampleInputEmail1">Category Image</label>
 <input type="file" class="form-control" name="category_image">

 </div>
 <div class="form-group">
 <label for="exampleInputPassword1">Category Description</label>
 <textarea style="resize: none;" rows="5" class="form-control" name="category_product_desc" id="exampleInputPassword1">{{ $edit_value->category_desc}}</textarea>
 </div>
 <button type="submit" name="update_category_product" class="btn btn-info">Update category</button>
</form>
```

View 20: Cập nhập loại sản phẩm

## .6.6 Quản trị Brand

- Người dùng quản trị đã đăng nhập có thể nhấn vào Tag “Brand” ở Menu bên trái để chọn các tính năng quản trị Brand
- Tag “Add Brand” dùng để thêm 1 Brand mới
- Tag “List Brand” dùng để xem danh sách Brand đang tồn tại, xoá, sửa các Brand



Hình 58: Quản trị thương hiệu

### .6.6.1 Thêm Brand

- Khi người dùng nhấn vào Tag “Add Brand”, Website sẽ chuyển qua giao diện nhập thông tin cho Brand mới
- Text Field “Brand Name” là tên của Brand mới
- Text Field “Brand Description” là mô tả của Brand mới
- Choose file “Brand Image” là file ảnh do người dùng chọn
- Combobox “Display” sẽ quyết định Brand đó có được hiển thị ra hay không
- Tất cả các ô Input đều bắt buộc người dùng nhập

A screenshot of a web-based form titled "ADD BRAND PRODUCT". The form contains four main input fields: "Brand Name" (text input), "Brand Description" (text area with a placeholder "Vui lòng điền vào trường này."), "Brand Image" (file upload input with a message "Chọn tệp | Không có tệp nào được chọn"), and "Display" (dropdown menu with "Show" selected). A blue "Add" button is located at the bottom left of the form area.

Hình 59: Màn hình thêm thương hiệu

- Khi người dùng vào Nút Add, Website sẽ gửi các thông tin đó lên add brand controller
- Add brand controller sẽ nhận các thông tin người dùng gửi lên
- Controller cũng sẽ kiểm tra xem File người dùng chọn có phải là ảnh hay không (Dựa vào đuôi của file)

## ➤ Hàm kiểm tra ảnh

```
//Hàm kiểm tra hình ảnh
function checkImage($fileName)
{
 //Cắt tên file truyền vào dựa vào dấu '.'
 $substr = explode('.', $fileName);
 //Nếu biến $substr trả ra 0 phần tử tức là file chọn không có tên => Không phải ảnh
 //Nếu biến $substr trả ra 1 phần tử tức là file chọn không có đuôi file => Không phải ảnh
 if (count($substr) == 0 || count($substr) == 1) {
 return false;
 }
 //Nếu phần tử cuối cùng của biến $substr trùng với các định dạng đuôi hình ảnh=>File là ảnh
 } else if (
 strtolower($substr[count($substr) - 1]) == 'png' ||
 strtolower($substr[count($substr) - 1]) == 'jpg' ||
 strtolower($substr[count($substr) - 1]) == 'gif' ||
 strtolower($substr[count($substr) - 1]) == 'bmp'){
 return true;
 }
 //Nếu không phải thì đó không phải là ảnh
 return false;
}
```

Controller 31: Kiểm tra hình ảnh

## ➤ Add Brand Controller

```
// Add
public function save_brand_product(Request $request)
{
 $this->AuthLogin();
 $data = $request->all();
 $brand = new Brand();
 $brand->brand_name = $data['brand_product_name'];
 $brand->brand_desc = $data['brand_product_desc'];
 $brand->brand_status = $data['brand_product_status'];
 $get_image = $request->file('brand_product_image');
 if ($get_image) {
 $get_name_image = $get_image->getClientOriginalName();
 $name_image = current(explode('.', $get_name_image));
 $new_image = $name_image . '.' . $get_image->getClientOriginalExtension();
 if (!$this->checkImage($new_image)) {
 return redirect('/add_brand_product')->with('message', 'The file is not a image !');
 }
 // $new_image = $name_image . rand(0, 99) . '.' . $get_image->getClientOriginalExtension(); // -- rand() set cho image link them so de khong bi trùng tên
 $get_image->move('public/uploads/brand', $new_image);
 $brand->brand_image = $new_image;
 $brand->save();
 return redirect('/add_brand_product')->with('message', 'Add product success !');
 }
 $brand->brand_image = 'NoImage.jpg';
 $brand->save();
 return redirect('/add_brand_product')->with('message', 'Add product success !');
}
```

Controller 32: Thêm thương hiệu

## ➤ Add Brand Router

```
Route::post('/save_brand_product', 'BrandProduct@save_brand_product');
```

Router 21: Thêm thương hiệu

## ➤ Add Brand View

```
<form role="form" enctype="multipart/form-data" method="POST" action="{{ URL::to('/save_brand_product') }}>
 @csrf
 <div class="form-group">
 <label for="exampleInputEmail1">Brand Name</label>
 <input required data-validation-length="min1" data-validation-error-msg="Please enter brand name" type="text" class="form-control" name="brand_product_name" placeholder="Enter brand name">
 </div>
 <div class="form-group">
 <label for="exampleInputPassword1">Brand Description</label>
 <textarea style="resize: none;" required rows="5" class="form-control" name="brand_product_desc" id="exampleInputPassword1"></textarea>
 </div>
 <div class="form-group">
 <label for="exampleInputEmail1">Brand Image</label>
 <input type="file" class="form-control" name="brand_product_image" required>
 </div>
 <div class="form-group">
 <label for="exampleInputPassword1">Display</label>
 <select name="brand_product_status" class="form-control input-sm m-bot15">
 <option value="0">Show</option>
 <option value="1">Hide</option>
 </select>
 </div>
 <button type="submit" name="add_brand_product" class="btn btn-info">Add</button>
</form>
```

View 21: Thêm thương hiệu

### .6.6.2 Xem danh sách Brand

- Người dùng nhấn vào tag “List Brand” sẽ chuyển qua Giao diện danh sách Brand, cùng với các thông tin như tên, hình ảnh, Display, Cùng với 2 Icon Update và Delete Brand đó
- Phía trên cùng góc phải là Form tìm kiếm Brand theo tên
- Một trang chỉ được có tối đa 5 Brand, nếu có quá 5 Brand thì Website sẽ phân ra các trang phía sau

LIST BRAND			
Brand Name	Brand Image	Display	
Prada			 
Nike			 
Chanel			 
Calvin Klein			 
Hermes			 

Hình 60: Danh sách thương hiệu

- Nếu brand\_status là 0 thì Display Brand sẽ hiển thị nút like
- Nếu brand\_status là 1 thì Display Brand sẽ hiển thị nút dislike
- Có thể trực tiếp thay đổi trạng thái Show, Hidden bằng cách nhấn trực tiếp vào Icon Like
- ⇒ Dislike và ngược lại, đồng thời cập nhật brand\_status vào Database

```
@if($brand_pro->brand_status == 0)
 brand_id)}}">
@else
 brand_id)}}">
@endif
```

View 22: Trạng thái thương hiệu

- Show list Brand Controller

```
// Get all product
public function all_brand_product()
{
 $this->AuthLogin();
 //Lấy 5 Brand trên 1 trang
 $all_brand_product = Brand::orderBy('brand_id', 'DESC')->paginate(5);
 $manager_brand_product = view('admin.all_brand_product')->with('all_brand_product', $all_brand_product);
 return view('admin.layout')->with('admin.all_brand_product', $manager_brand_product);
}
```

Controller 33: Lấy thương hiệu

- Show list Brand Router

```
Route::get('/all_brand_product', 'BrandProduct@all_brand_product');
```

Router 22: Lấy thương hiệu

#### .6.6.3 Tìm kiếm Brand theo tên

- Người dùng có thể tìm kiếm Brand theo tên bằng cách viết từ khoá vào ô Text Field “Enter Name of Brand to search” và nhấn vào Button Search
- Khi nhấn vào Button Search thì Website sẽ truyền từ khoá đó lên Controller, Controller sẽ thực hiện truy vấn dựa trên từ khoá website gửi lên, và trả kết quả về View để View duyệt
- Vd Search với từ khoá “Ca”

LIST BRAND		
Brand Name	Brand Image	Display
<b>We found 1 Brands have keyword "Ca"</b>		
Calvin Klein		  

Hình 61: Tìm kiếm thương hiệu

- Search Brand Controller

```
public function search_brand(Request $request){
 //Lấy từ khoá Form gửi lên
 $keywords = $request->keywords_submit;
 //Thực hiện truy vấn
 $search_brand = Brand::where('brand_name', 'like', '%' . $keywords . '%')->orderBy('brand_id', 'desc')->get();
 return view('/admin/search_brand')->with('search_brand', $search_brand);
}
```

Controller 34: Tìm kiếm thương hiệu

- Search Brand Router

```
Route::post('/search_brand', 'CheckoutController@search_brand');
```

Router 23: Tìm kiếm thương hiệu

## ➤ Search Brand View

```

@foreach ($search_brand as $key => $brand_pro)
|
<td>{{ $brand_pro->brand_name }}</td>
<td></td>
<td>
 <?php
 if ($brand_pro->brand_status == 0) { ?>
 brand_id)}}>
 <?php } else { ?>
 brand_id)}}>
 <?php } ?>
 </td>
<td>
 brand_id)}}> class="active styling-edit" ui-toggle-class="">
 <i class="fa fa-pencil-square-o text-success text-active"></i>

 brand_id)}}> class="active styling-edit" ui-toggle-class="">
 <i class="fa fa-times text-danger text"></i>

</td>

| |

@endforeach

```

View 23: Tìm kiếm thương hiệu

### .6.6.4 Xoá Brand

- Ở List Brand, người dùng có thể nhấn vào icon delete ở cuối mỗi hàng brand trong danh sách
- Khi nhấn vào, Website sẽ gửi id của Brand đó lên Delete Brand Controller
- Controller sẽ kiểm tra cách sản phẩm thuộc Brand đó, nếu còn sản phẩm thì không xoá Brand đó, và đồng in in ra thông báo không xoá Brand đó được
- Nếu không còn sản phẩm thì xoá Brand đó, và in ra thông báo xoá thành công đồng thời xoá luôn file ảnh của brand đó.
- Delete Brand Controller

```

// delete
public function delete_brand_product($brand_product_id)
{
 $this->AuthLogin();
 $brand = Brand::find($brand_product_id);
 $product_by_brand = DB::table('tbl_product')->where('brand_id', $brand_product_id)->get();
 //Nếu hết sản phẩm rồi mới xoá dc
 if (count($product_by_brand) == 0) {
 // delete image
 $destinationPath = 'public/uploads/brand/' . $brand->brand_image;
 if (file_exists($destinationPath)) {
 unlink($destinationPath);
 }
 $brand->delete();
 return redirect('all_brand_product')->with('message', 'Product brand delete successfully ');
 }
 return redirect('all_brand_product')->with('message', 'Cannot delete Brand because this Brand has products');
}

```

Controller 35: Xoá thương hiệu

- Delete Brand Router

```
Route::get('/delete_brand_product/{brand_product_id}', 'BrandProduct@delete_brand_product');
```

Router 24: Xoá thương hiệu

## ➤ Delete Brand View

```
brand_id)}}"
class="active styling-edit" ui-toggle-class="">
```

*View 24: Xoá thương hiệu*

### .6.6.5 Update Brand

- Ở List Brand, người dùng có thể nhấn vào Icon Update ở cuối mỗi hàng Brand trong danh sách
- Khi nhấn vào, Website sẽ gửi id của Brand đó lên Update Brand Controller
- Controller sẽ trả về View thông tin của Brand đó về view
- View sẽ điền các thông tin của Brand vào các ô input
- Khi nhấn Submit, Website sẽ gửi các thông tin về update Controller (Bao gồm cả id)
- Controller cũng sẽ kiểm tra người dùng có chọn file mới hay không
- Nếu người dùng không chọn file mới, Controller sẽ không thay đổi giá trị tên hình
- Nếu người dùng chọn file mới, Controller sẽ kiểm tra file mới có phải là hình hay không, nếu file là hình thì Controller sẽ Upload hình mới, và xoá hình cũ, nếu file không phải là hình, hệ thống sẽ không update hình ảnh
- Giao diện khi nhấn vào Update 1 Brand

EDIT BRAND PRODUCT

Brand Name  
Gucci

Brand Image  
Chọn tệp Không có tệp nào được chọn  
GUCCI

Brand Description  
The House of Gucci, hay được biết đến ngắn gọn là Gucci, là một biểu tượng thời trang sở hữu bởi Ý và Pháp, một nhãn hiệu đồ da nổi tiếng.  
Nhà sáng lập: Guccio Gucci  
Ngày thành lập: 1921, Florence, Ý  
Công ty mẹ: Kering

Update brand

*Hình 62: Cập nhập thương hiệu*

- Nếu Update thành công

ADD BRAND PRODUCT

Update brand success !

Brand Name

Brand Description

Brand Image  
Chọn tệp Không có tệp nào được chọn

Display  
Show

Add

*Hình 63: Cập nhập thương hiệu thành công*

- Nếu Update mà người dùng chọn file không phải là hình

Hình 64: Cập nhập thương hiệu không thành công

- Update Brand Controller

```
// update
public function update_brand_product($brand_product_id, Request $request)
{
 $this->AuthLogin();
 $data = $request->all();
 $brand = Brand::find($brand_product_id);
 $brand->brand_name = $data['brand_product_name'];
 $brand->brand_desc = $data['brand_product_desc'];
 $get_image = $request->file('brand_product_image');

 if ($get_image) {
 //Lấy tên hình ảnh mới
 $get_name_image = $get_image->getClientOriginalName();
 $name_image = current(explode('.', $get_name_image));
 $new_image = $name_image . '.' . $get_image->getClientOriginalExtension();
 if (!$this->checkImage($new_image)) {
 $brand->save();
 return redirect('/add_brand_product')->with('message', 'Cannot update the image because the file is not a image !');
 }
 // delete image
 $destinationPath = 'public/uploads/brand/' . $brand->brand_image;
 if (file_exists($destinationPath)) {
 unlink($destinationPath);
 }
 //Upload file file và cập nhật ảnh
 $get_image->move('public/uploads/brand', $new_image);
 $brand->brand_image = $new_image;
 $brand->save();
 return redirect('/add_brand_product')->with('message', 'Update brand success !');
 }
 $brand->save();

 return redirect('/add_brand_product')->with('message', 'Update brand success !');
}
```

Controller 36: Cập nhập thương hiệu

- Update Brand Router

```
Route::post('/update_brand_product/{brand_product_id}', 'BrandProduct@update_brand_product');
```

Router 25: Cập nhập thương hiệu

- Update Brand View

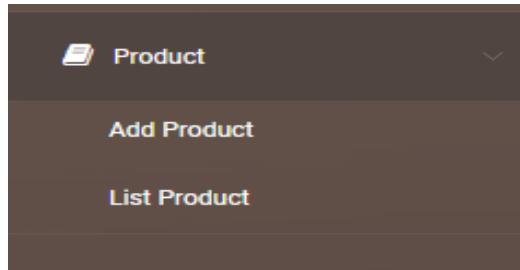
```
@foreach ($edit_brand_product as $key => $edit_value)
<div class="position-center">
 <form enctype="multipart/form-data" role="form" method="POST" action="{{URL::to('/update_brand_product/'.$edit_value->brand_id)}}>
 @csrf
 <div class="form-group">
 <label for="exampleInputEmail1">Brand Name</label>
 <input type="text" class="form-control" value="{{ $edit_value->brand_name}}" name="brand_product_name" >
 </div>
 <div class="form-group">
 <label for="exampleInputEmail1">Brand Image</label>
 <input type="file" class="form-control" name="brand_product_image">

 </div>
 <div class="form-group">
 <label for="exampleInputPassword1">Brand Description</label>
 <textarea style="resize: none;" rows="5" class="form-control" name="brand_product_desc" id="exampleInputPassword1" value="{{ $edit_value->brand_desc}}</textarea>
 </div>
 <button type="submit" name="update_brand_product" class="btn btn-info">Update brand</button>
 </form>
</div>
@endforeach
```

View 25: Cập nhập thương hiệu

## 6.7 Quản trị Product

- Người dùng quản trị đã đăng nhập có thể nhấn vào Tag “Product” ở Menu bên trái để chọn các tính năng quản trị Product
- Tag “Add Product” dùng để thêm 1 Product mới
- Tag “List Product” dùng để xem danh sách Brand đang tồn tại, xoá, sửa các Product



Hình 65: Quản trị sản phẩm

### 6.7.1 Thêm Product

- Khi người dùng nhấn vào Tag “Add Product”, Website sẽ chuyển qua giao diện nhập thông tin cho Brand mới
- Text Field “Product Name” là tên của Product mới
- Text Field “Product Price” là giá tiền của Product mới
- Choose file “Product Image” là file ảnh do người dùng chọn
- Text Field “Product Description” là mô tả của Product mới
- Text Field “Product Content” là nội dung của Product mới
- Combobox “Display” sẽ quyết định Brand đó có được hiển thị ra hay không
- Combobox “Category Product” dùng để xác định Category id cho product đó
- Combobox “Brand Product” dùng để xác định Brand id cho product đó
- Tất cả các ô Input đều bắt buộc người dùng nhập

A screenshot of a web-based form titled "ADD PRODUCT". The form consists of several input fields: "Product Name" (text input), "Product Price" (text input), "Product Image" (file input with placeholder "Chọn tệp | Không có tệp nào được chọn"), "Product Description" (text area), "Product Content" (text area), "Display" (dropdown menu set to "Show"), "Category Product" (dropdown menu set to "Hat"), and "Brand Product" (dropdown menu set to "Prada"). At the bottom right of the form is a blue "Add" button.

Hình 66: Màn hình thêm sản phẩm

- Khi người dùng vào Nút Add, Website sẽ gửi các thông tin đó lên add Product Controller
- Add Product controller sẽ nhận các thông tin người dùng gửi lên
- Controller cũng sẽ kiểm tra xem File người dùng chọn có phải là ảnh hay không (Dựa vào đuôi của file)
- Hàm kiểm tra ảnh

```
//Hàm kiểm tra hình ảnh
function checkImage($fileName)
{
 //Cắt tên file truyền vào dựa vào dấu '.'
 $substr = explode('.', $fileName);
 //Nếu biến $substr trả ra 0 phần tử tức là file chọn không có tên => Không phải ảnh
 //Nếu biến $substr trả ra 1 phần tử tức là file chọn không có đuôi file => Không phải ảnh
 if (count($substr) == 0 || count($substr) == 1) {
 return false;
 }
 //Nếu phần tử cuối cùng của biến $substr trùng với các định dạng đuôi hình ảnh=>File là ảnh
} else if (
 strtolower($substr[count($substr) - 1]) == 'png' ||
 strtolower($substr[count($substr) - 1]) == 'jpg' ||
 strtolower($substr[count($substr) - 1]) == 'gif' ||
 strtolower($substr[count($substr) - 1]) == 'bmp'){
 return true;
}
//Nếu không phải thì đó không phải là ảnh
return false;
}
```

*Controller 37: Kiểm tra hình ảnh*

- Add Product Controller

```
// Add
public function save_product(Request $request)
{
 $this->AuthLogin();
 $data = $request->all();
 $product = new Product();
 $product->category_id = $data['product_cate'];
 $product->brand_id = $data['product_brand'];
 $product->product_name = $data['product_name'];
 $product->product_desc = $data['product_desc'];
 $product->product_content = $data['product_content'];
 $product->product_price = $data['product_price'];
 $product->product_status = $data['product_status'];
 $get_image = $request->file('product_image');
 if ($get_image) {
 $get_name_image = $get_image->getClientOriginalName();
 $name_image = current(explode('.', $get_name_image));
 $new_image = $name_image . '.' . $get_image->getClientOriginalExtension();
 //Kiểm tra file
 if (!$this->checkImage($new_image)) {
 $product->product_image = '';
 $product->save();
 return redirect('/add_product')->with('message', 'Add product success, but cannot update the image because the file is not a image !');
 }
 $get_image->move('public/uploads/product', $new_image);
 $product->product_image = $new_image;
 $product->save();
 return redirect('/add_product')->with('message', 'Add product success !');
 }
 $product->product_image = 'NoImage.jpg';
 $product->save();
 return redirect('/add_product')->with('message', 'Add product success !');
}
```

*Controller 38: Thêm sản phẩm*

- Add Product Router

```
Route::post('/save_product', 'ProductController@save_product');
```

*Router 26: Thêm sản phẩm*

### 6.7.2 Xem danh sách Product

- Người dùng nhấn vào tag “List Product” sẽ chuyển qua Giao diện danh sách Product, cùng với các thông tin như tên, hình ảnh, Display, Cùng với 2 Icon Update và Delete Product đó
- Phía trên cùng góc phải là Form tìm kiếm Product theo tên
- Một trang chỉ được có tối đa 5 Product, nếu có quá 5 Product thì Website sẽ phân ra các trang phía sau

Product Name	Price	Product Image	Category	Brand	Show
Mũ đen Prada	10824000		Hat	Prada	 
QUẦN SHORT COTTON MÀU BE	2598000		Short	Prada	 
Quần dài & Bermuda Prada 2	13047000		Short	Prada	 
Quần dài & Bermuda Prada 1	15794000		Short	Prada	 
Quần denim 3	7966000		Jean	Prada	 

... 1 2 3 4 5 6 7 8 9 10 ... 50 51 >

Hình 67: Hiển thị danh sách sản phẩm

- Nếu product\_status là 0 thì Display product sẽ hiển thị nút like
- Nếu product\_status là 1 thì Display product sẽ hiển thị nút dislike
- Có thể trực tiếp thay đổi trạng thái Show, Hidden bằng cách nhấn trực tiếp vào Icon Like
- ⇒ Dislike và ngược lại, đồng thời cập nhật product\_status vào Database

```
@if($pro->product_status == 0)
 product_id)}}">
@else
 product_id)}}">
@endif
```

View 26: Trạng thái sản phẩm

- Show list Product Controller

```
// Get all product
public function all_product()
{
 $this->AuthLogin();
 //Lấy thông tin 5 sản phẩm trong trang, và nối với 2 bảng còn lại
 $all_product = DB::table('tbl_product')
 ->join('tbl_category_product', 'tbl_category_product.category_id', '=', 'tbl_product.category_id')
 ->join('tbl_brand', 'tbl_brand.brand_id', '=', 'tbl_product.brand_id')
 ->orderBy('tbl_product.product_id', 'desc')->paginate(5);
 $manager_product = view('admin.all_product')->with('all_product', $all_product);
 return view('admin.layout')->with('admin.all_product', $manager_product);
}
```

Controller 39: Lấy sản phẩm

➤ Show list product Router

```
Route::get('/all_product', 'ProductController@all_product');
```

Router 27: Lấy sản phẩm

#### .6.7.3 Tìm kiếm Product theo tên

- Người dùng có thể tìm kiếm Product theo tên bằng cách viết từ khoá vào ô Text Field “Enter Name of Product to search” và nhấn vào Button Search
- Khi nhấn vào Button Search thì Website sẽ truyền từ khoá đó lên Controller, Controller sẽ thực hiện truy vấn dựa trên từ khoá website gửi lên, và trả kết quả về View để View duyệt
- Vd Search Product với từ khoá “Mũ”

The screenshot shows a search results page for products containing the keyword "Mũ". At the top, there is a search bar labeled "Enter Product Name to search" with the placeholder text "Search". Below the search bar, the text "We found 12 products have keyword \"Mũ\"" is displayed. A table lists three products:

Product Name	Price	Product Image	Category	Brand	Show
Mũ đen Prada	10824000		Hat	Prada	
Quần đùi & Bermuda Prada 2	13047000		Short	Prada	
Quần đùi & Bermuda Prada 1	15794000		Short	Prada	

Hình 68: Tìm kiếm sản phẩm theo tên

➤ Search Product Controller

```
public function search_product(Request $request)
{
 //Lấy keyword
 $keywords = $request->keywords_submit;
 //Lấy danh sách product dựa vào keyword
 $search_product = DB::table('tbl_product')
 ->where('product_name', 'like', '%' . $keywords . '%')
 ->join('tbl_category_product', 'tbl_category_product.category_id', '=', 'tbl_product.category_id')
 ->join('tbl_brand', 'tbl_brand.brand_id', '=', 'tbl_product.brand_id')
 ->orderBy('tbl_product.product_id', 'desc')->get();
 $all_product = Product::all();
 return view('/admin/search_product')
 ->with('search_product', $search_product)
 ->with('keywords', $keywords)
 ->with('all_product', $all_product);
}
```

Controller 40: Tìm kiếm sản phẩm

## ➤ Search Product Router

```
Route::get('/search_product', 'CheckoutController@search_product');
```

Router 28: Tìm kiếm sản phẩm

### .6.7.4 Xoá Product

- Ở List Product, người dùng có thể nhấn vào icon delete ở cuối mỗi hàng brand trong danh sách
- Khi nhấn vào, Website sẽ gửi id của Product đó lên Delete Product Controller
- Đồng thời Controller cũng sẽ xoá file hình ảnh của sản phẩm đó
- Delete Product Controller

```
// delete
public function delete_product($product_id)
{
 $this->AuthLogin();
 $product = Product::find($product_id);
 // delete image
 //Nếu sp không có tên hình hoặc tên hình là hình mặc định thì không cần xoá file
 if ($product->product_image == "" || $product->product_image = "NoImage.jpg") {
 $product->delete();
 return redirect('all_product')->with('message', 'Product delete successfully ');
 }
 //Nếu sp có tên thì xoá file
 $destinationPath = 'public/uploads/product/' . $product->product_image;

 if (file_exists($destinationPath)) {
 unlink($destinationPath);
 }
 $product->delete();
 return redirect('all_product')->with('message', 'Product delete successfully ');
}
```

Controller 41: Xoá sản phẩm

## ➤ Delete Product Router

```
Route::get('/delete_product/{product_id}', 'ProductController@delete_product');
```

Router 29: Xoá sản phẩm

## ➤ Delete Product View

```
<a onclick="return confirm('Are you sure to delete?')"
href="{{URL::to('/delete_product/'.$pro->product_id)}}"
class="active styling-edit ui-toggle-class="">
| <i class="fa fa-times text-danger text"></i>

```

View 27: Xoá sản phẩm

#### .6.7.5 Update Product

- Ở List Product, người dùng có thể nhấn vào Icon Update ở cuối mỗi hàng Product trong danh sách
- Khi nhấn vào, Website sẽ gửi id của Product đó lên Update Product Controller
- Controller sẽ trả về View thông tin của Product đó về view
- View sẽ điền các thông tin của Product vào các ô input
- Khi nhấn Submit, Website sẽ gửi các thông tin về update Controller (Bao gồm cả id)
- Controller cũng sẽ kiểm tra người dùng có chọn file mới hay không
- Nếu người dùng không chọn file mới, Controller sẽ không thay đổi giá trị tên hình
- Nếu người dùng chọn file mới, Controller sẽ kiểm tra file mới có phải là hình hay không, nếu file là hình thì Controller sẽ Upload hình mới, và xoá hình cũ, nếu file không phải là hình, hệ thống sẽ không update hình ảnh
- Giao diện khi nhấn vào Update 1 Product

UPDATE PRODUCT

Product Name	Mũ đen Prada
Product Price	10824000
Product Image	<input type="button" value="Chọn tệp"/> Không có tệp nào được chọn
	
Product Description	Logo hình tam giác bằng kim loại trắng men Lớp lót bông.
Product Content	none

Hình 69: Cập nhập sản phẩm

- Nếu Update thành công

LIST PRODUCT

Product Name	Price	Product Image	Category	Brand	Show	
Mũ đen Prada	10824000		Hat	Prada		
QUẦN SHORT COTTON MÀU BE	2598000		Short	Prada		
Quần dài & Bermuda Prada 2	13047000		Short	Prada		

Hình 70: Cập nhập sản phẩm thành công

- Nếu Update mà người dùng chọn file không phải là hình

LIST PRODUCT					
Enter Product Name to search					Search
Product Name	Price	Product Image	Category	Brand	Show
Mũ den Prada	10824000		Hat	Prada	  
QUẦN SHORT COTTON MÀU BE	2598000		Short	Prada	  

Hình 71: Cập nhập sản phẩm không thành công

- Update Product Controller

```
public function update_product(Request $request, $product_id)
{
 $this->authLogin();
 $data = $request->all();
 $product = Product::find($product_id);
 $product->category_id = $data['product_cate'];
 $product->brand_id = $data['product_brand'];
 $product->product_name = $data['product_name'];
 $product->product_desc = $data['product_desc'];
 $product->product_content = $data['product_content'];
 $product->product_price = $data['product_price'];
 $product->product_status = $data['product_status'];
 $get_image = $request->file('product_image');
 if ($get_image) {
 //Lấy tên ảnh
 $get_name_image = $get_image->getClientOriginalName();
 $name_image = current(explode('.', $get_name_image));
 $new_image = $name_image . '.' . $get_image->getClientOriginalExtension();
 if (!$this->checkImage($new_image)) {
 $product->save();
 return redirect('/all_product')->with('message', 'Add product success,But cannot update the image because the file is not a image !');
 }
 // delete image
 if ($product->product_image == "") {
 $get_image->move('public/uploads/product', $new_image);
 $product->product_image = $new_image;
 $product->save();
 return redirect('/all_product')->with("message", "Add product success !");
 }
 $destinationPath = 'public/uploads/product' . $product->product_image;
 if (file_exists($destinationPath)) {
 unlink($destinationPath);
 }
 $get_image->move('public/uploads/product', $new_image);
 $product->product_image = $new_image;
 $product->save();
 return redirect('/all_product')->with('message', 'Add product success !');
 }
}
```

Controller 42: Cập nhập sản phẩm

- Update Product Router

```
Route::post('/save_product', 'ProductController@save_product');
```

Router 30: Cập nhập sản phẩm

## ➤ Update Product View

```
 @foreach ($edit_product as $key => $pro)
 @form role="form" enctype="multipart/form-data" method="POST" action="{{URL::to('/update_product/'.$pro->product_id)}}"
 @csrf
 <div class="form-group">
 <label for="exampleInputEmail1">Product Name</label>
 <input type="text" class="form-control" name="product_name" value="{{$pro->product_name}}>
 </div>
 <div class="form-group">
 <label for="exampleInputEmail1">Product Price</label>
 <input type="text" class="form-control" name="product_price" value="{{$pro->product_price}}>
 </div>
 <div class="form-group">
 <label for="exampleInputEmail1">Product Image</label>
 <input type="file" class="form-control" name="product_image">

 </div>
 <div class="form-group">
 <label for="exampleInputPassword1">Product Description</label>
 <textarea style="resize: none;" rows="5" class="form-control" name="product_desc" id="ckeditor1">{{$pro->product_desc}}></textarea>
 </div>
 <div class="form-group">
 <label for="exampleInputPassword1">Product Content</label>
 <textarea style="resize: none;" rows="5" class="form-control" name="product_content" id="ckeditor">{{$pro->product_content}}></textarea>
 </div>
 <div class="form-group">
 <label for="exampleInputPassword1">Display</label>
 <select name="product_status" class="form-control input-sm m-bot15">
 <option value="0">Hide</option>
 <option value="1">Show</option>
 </select>
 </div>
```

View 28: Cập nhập sản phẩm 1

```
</div>
<div class="form-group">
 <label for="exampleInputPassword1">Display</label>
 <select name="product_status" class="form-control input-sm m-bot15">
 <option value="0">Hide</option>
 <option value="1">Show</option>
 </select>
</div>
<div class="form-group">
 <label for="exampleInputPassword1">Category Product</label>
 <select name="product_cate" class="form-control input-sm m-bot15">
 @foreach($cate_product as $key => $cate)
 @if($cate->category_id == $pro->category_id)
 <option selected value="{{$cate->category_id}}>{{$cate->category_name}}</option>
 @else
 <option value="{{$cate->category_id}}>{{$cate->category_name}}</option>
 @endif
 @endforeach
 </select>
</div>
<div class="form-group">
 <label for="exampleInputPassword1">Brand Product</label>
 <select name="product_brand" class="form-control input-sm m-bot15">
 @foreach($brand_product as $key => $brand)
 @if($brand->brand_id == $pro->brand_id)
 <option selected value="{{$brand->brand_id}}>{{$brand->brand_name}}</option>
 @else
 <option value="{{$brand->brand_id}}>{{$brand->brand_name}}</option>
 @endif
 @endforeach
 </select>
</div>
<button type="submit" name="add_product" class="btn btn-info">Update</button>
</form>
@endforeach
@endforeach
```

View 29: Cập nhập sản phẩm 2

## 7. Test case

### 7.1.1 Mô tả:

-Test case sẽ cảng cứ vào sơ đồ Use case và đặc tả ban đầu

-Use case sẽ kiểm tra các chức năng của Website có hoạt động đúng yêu cầu hay không

### 7.1.2 Test case người dùng quản trị

TC ID	Tiêu Đề	Test Type	Điều kiện tiên quyết	Các bước thực hiện	Kết quả thực tế	Mức độ quan trọng	Danh giá(*)
<b>ADMIN</b>							
							N/A
1	Logout	Kiểm tra chức năng	Người dùng phải vào được trang đăng nhập của trang quản trị	Click vào nút đăng xuất	Reload trang hiện tại, trở về màn hình đăng nhập của admin	Trung bình	PASSED
2	Search Product	Kiểm tra chức năng	Người dùng phải đăng nhập tài khoản admin thành công	1. Vào tag "List Product" 2. Click vào thanh tìm kiếm 3. Nhập từ khóa về sản phẩm bạn muốn tìm kiếm 4. Click tiếp vào biểu tượng tìm kiếm	Website hiển thị ra toàn bộ sản phẩm theo từ khoá	Trung bình	PASSED
3	Search Category	Kiểm tra chức năng	Người dùng phải đăng nhập tài khoản admin thành công	1. Vào tag "List Category" 2. Click vào thanh tìm kiếm 3. Nhập từ khóa về Category bạn muốn tìm kiếm 4. Click tiếp vào biểu tượng tìm kiếm	Website hiển thị ra toàn bộ Category theo từ khoá	Trung bình	PASSED
4	Search Brand	Kiểm tra chức năng	Người dùng phải đăng nhập tài khoản admin thành công	1. Vào tag "List Brand" 2. Click vào thanh tìm kiếm 3. Nhập từ khóa về Brand bạn muốn tìm kiếm 4. Click tiếp vào biểu tượng tìm kiếm	Website hiển thị ra toàn bộ Brand theo từ khoá	Trung bình	PASSED
5	View Category	Kiểm tra chức năng	1.Người dùng phải đăng nhập tài khoản admin thành công 2. Phải có ít nhất 1 Category	1. Vào tag "All Category"	Website hiển thị ra 5 Category trong số toàn bộ Category	Trung bình	PASSED
6	Giao diện phản ứng cho tag "All Category"	Kiểm tra giao diện	1.Người dùng phải đăng nhập tài khoản admin thành công 2.Trang "All Category" phải có hơn 5 Category	1. Vào tag "All Category" 2. Kéo xuống bên dưới List Category (Gắn Footer)	1.Website hiển thị ra 5 Category trong số toàn bộ Category 2.Dưới List Category phải hiện thi ra danh sách các trang tiếp theo	Trung bình	PASSED
7	Chức năng phản ứng cho tag "All Category"	Kiểm tra chức năng	1.Người dùng phải đăng nhập tài khoản admin thành công 2.Trang "All Category" phải có hơn 5 Category	1. Vào tag "All Category" 2. Kéo xuống bên dưới List Category (Gắn Footer) 3. Nhấn qua trang tiếp theo	Website hiển thị ra 5 Category tiếp theo trong số toàn bộ Category	Trung bình	PASSED
8	Add New Category	Kiểm tra chức năng	Người dùng phải đăng nhập tài khoản admin thành công	1. Vào tag "Add Category" 2. Điền tất cả các Field 3.Nhấn nút Add 4.Vào tag "All Category"	Reload trang hiện tại và tồn tại Category mới vừa thêm	Trung bình	PASSED
9	Delete Category	Kiểm tra chức năng	1.Người dùng phải đăng nhập tài khoản admin thành công 2.Phải có ít nhất 1 Category	1. Vào tag "All Category" 2. Nhấn vào Icon Delete ở cuối một Category bất kỳ	Reload trang hiện tại và Category vừa xoá biến mất khỏi danh sách	Trung bình	PASSED

10	Edit Category	Kiểm tra chức năng	1.Người dùng phải đăng nhập tài khoản admin thành công 2.Phải có ít nhất 1 Category	1. Vào tag "All Category" 2. Nhấn vào Icon Update ở cuối một Category bất kỳ	Reload trang hiện tại và các thông tin của Category vừa Update sẽ được Update lên Category hiện tại	Trung bình	PASSED
11	View Brand	Kiểm tra chức năng	1.Người dùng phải đăng nhập tài khoản admin thành công 2. Phải có ít nhất 1 Brand	1. Vào tag "All Brand"	Website hiển thị ra 5 Brand trong số toàn bộ Brand	Trung bình	PASSED
12	Giao diện phản trang cho tag "All Brand"	Kiểm tra giao diện	1.Người dùng phải đăng nhập tài khoản admin thành công 2.Trang "All Brand" phải có hơn 5 Brand	1. Vào tag "All Brand" 2. Kéo xuống bên dưới List Brand (Gần Footer)	1.Website hiển thị ra 5 Brand trong số toàn bộ Brand 2.Dưới List Brand phải hiện thi ra danh sách các trang tiếp theo	Trung bình	PASSED
13	Chức năng phản trang cho tag "All Brand"	Kiểm tra chức năng	1.Người dùng phải đăng nhập tài khoản admin thành công 2.Trang "All Brand" phải có hơn 5 Brand	1. Vào tag "All Brand" 2. Kéo xuống bên dưới List Brand (Gần Footer) 3. Nhấn qua trang tiếp theo	Website hiển thị ra 5 Brand tiếp theo trong số toàn bộ Brand	Trung bình	PASSED
14	Add New Brand	Kiểm tra chức năng	Người dùng phải đăng nhập tài khoản admin thành công	1. Vào tag "Add Brand" 2. Diện tất cả các Field 3.Nhấn nút Add 4.Vào tag "All Brand"	Reload trang hiện tại và tồn tại Brand mới vừa thêm	Trung bình	PASSED
15	Delete Brand	Kiểm tra chức năng	1.Người dùng phải đăng nhập tài khoản admin thành công 2.Phải có ít nhất 1 Brand	1. Vào tag "All Brand" 2. Nhấn vào Icon Delete ở cuối một Brand bất kỳ	Reload trang hiện tại và Brand vừa xoá biến mất khỏi danh sách	Trung bình	PASSED
16	Edit Brand	Kiểm tra chức năng	1.Người dùng phải đăng nhập tài khoản admin thành công 2.Phải có ít nhất 1 Brand	1. Vào tag "All Brand" 2. Nhấn vào Icon Update ở cuối một Brand bất kỳ	Reload trang hiện tại và các thông tin của Brand vừa Update sẽ được Update lên Category hiện tại	Trung bình	PASSED

17	View Product	Kiểm tra chức năng	1.Người dùng phải đăng nhập tài khoản admin thành công 2. Phải có ít nhất 1 Product	1. Vào tag "All Product"	Website hiển thị ra 5 Product trong số toàn bộ Product	Trung bình	PASSED
18	Giao diện phản trang cho tag "All Product"	Kiểm tra giao diện	1.Người dùng phải đăng nhập tài khoản admin thành công 2.Trang "All Product" phải có hơn 5 Product	1. Vào tag "All Product" 2. Kéo xuống bên dưới List Product (Gần Footer)	1.Website hiển thị ra 5 Product trong số toàn bộ Product 2.Dưới List Product phải hiện thi ra danh sách các trang tiếp theo	Trung bình	PASSED
19	Chức năng phản trang cho tag "All Product"	Kiểm tra chức năng	1.Người dùng phải đăng nhập tài khoản admin thành công 2.Trang "All Product" phải có hơn 5 Product	1. Vào tag "All Product" 2. Kéo xuống bên dưới List Product (Gần Footer) 3. Nhấn qua trang tiếp theo	Website hiển thị ra 5 Product tiếp theo trong số toàn bộ Product	Trung bình	PASSED
20	Add New Product	Kiểm tra chức năng	Người dùng phải đăng nhập tài khoản admin thành công	1. Vào tag "Add Product" 2. Diện tất cả các Field 3.Nhấn nút Add 4.Vào tag "All Product"	Reload trang hiện tại và tồn tại Product mới vừa thêm	Trung bình	PASSED
21	Delete Product	Kiểm tra chức năng	1.Người dùng phải đăng nhập tài khoản admin thành công 2.Phải có ít nhất 1 Product	1. Vào tag "All Product" 2. Nhấn vào Icon Delete ở cuối một Product bất kỳ	Reload trang hiện tại và Product vừa xoá biến mất khỏi danh sách	Trung bình	PASSED
22	Edit Product	Kiểm tra chức năng	1.Người dùng phải đăng nhập tài khoản admin thành công 2.Phải có ít nhất 1 Product	1. Vào tag "All Product" 2. Nhấn vào Icon Update ở cuối một Product bất kỳ	Reload trang hiện tại và các thông tin của Product vừa Update sẽ được Update lên Product hiện tại	Trung bình	PASSED

24	View Order Detail	Kiểm tra chức năng	1.Người dùng phải đăng nhập tài khoản admin thành công 2.Phải có ít nhất 1 Order	1.Click vào Manager Orders 2.Click vào Icon Update của 1 Order bất kỳ	Thông tin chi tiết của Order sẽ được show ra như là Thông tin khách hàng, Thông tin các đơn hàng khách đã Order	Trung bình	PASSED
25	Processing Order	Kiểm tra chức năng	1.Người dùng phải đăng nhập tài khoản admin thành công 2.Phải có ít nhất 1 Order	1.Click vào Manager Orders 2.Click vào Icon Update của 1 Order bất kỳ 3.Kéo xuống Form Input "New order Status" 4.Nhập Status Order mới và nhấn Submit	Order Status mới sẽ được cập nhật	Trung bình	PASSED
26	Delete Order	Kiểm tra chức năng	1.Người dùng phải đăng nhập tài khoản admin thành công 2.Phải có ít nhất 1 Order	1.Click vào Manager Orders 2.Click vào Icon Delete của 1 Order bất kỳ	Hiển thị "Bạn có muốn xóa món hàng này không?" Nếu chọn CÓ thì hàng hóa đã chọn được xóa Nếu chọn KHÔNG thì hàng hóa đã chọn vẫn giữ nguyên	Trung bình	PASSED

### 7.1.3 Test case người dùng khách hàng đã đăng nhập

TC ID	Tiêu Đề	Test Type	Điều kiện tiên quyết	Các bước thực hiện	Kết quả thực tế	Mức độ quan trọng	Đánh giá(*)
<b>LOGGED IN USER</b>							
1	Logout	Kiểm tra chức năng	Người dùng phải đăng nhập bằng tài khoản khách hàng	Click vào nút đăng xuất	Hiển thị "Bạn có muốn đăng xuất hay không?" Reload trang hiện tại ở chế độ "User not logged in"	Trung bình	N/A <b>PASSED</b>
2	Search Product	Kiểm tra chức năng	Người dùng phải đăng nhập bằng tài khoản khách hàng	1. Click vào thanh tìm kiếm 2. Nhập từ khóa về sản phẩm bạn muốn tìm kiếm 3. Click tiếp vào biểu tượng tìm kiếm	Hiển thị sản phẩm bạn muốn tìm kiếm với từ khóa đã nhập	Cao	<b>PASSED</b>
3	Search Category	Kiểm tra chức năng	1.Người dùng phải đăng nhập bằng tài khoản khách hàng 2.Phải có ít nhất 1 Category trong danh sách	1. Click vào Combobox Danh sách Category 2. Nhấn vào 1 Category trong danh sách	Website hiển thị toàn bộ sản phẩm thuộc Category đã chọn	Cao	<b>PASSED</b>
4	Search Brand	Kiểm tra chức năng	1.Người dùng phải đăng nhập bằng tài khoản khách hàng 2.Phải có ít nhất 1 Brand trong danh sách	1. Click vào Combobox Danh sách Brand 2. Nhấn vào 1 Brand trong danh sách	Website hiển thị toàn bộ sản phẩm thuộc Brand đã chọn	Cao	<b>PASSED</b>
5	View Product	Kiểm tra giao diện	1.Người dùng phải đăng nhập bằng tài khoản khách hàng 2. Phải có ít nhất 1 sản phẩm có Status là 0	1.Truy cập Website 2.Vào tag Hot Product hoặc tag All Product	Website hiển thị toàn bộ sản phẩm	Cao	<b>PASSED</b>
6	View Category	Kiểm tra giao diện	Người dùng phải đăng nhập bằng tài khoản khách hàng	1. Click vào Combobox Danh sách Category	Combobox hiển thị toàn bộ Category mà Admin cho phép hiển thị	Trung bình	<b>PASSED</b>
7	View Brand	Kiểm tra giao diện	Người dùng phải đăng nhập bằng tài khoản khách hàng	1. Click vào Combobox Danh sách Brand	Combobox hiển thị toàn bộ Brand mà Admin cho phép hiển thị	Trung bình	<b>PASSED</b>
8	View Cart	Kiểm tra chức năng	1.Người dùng phải đăng nhập bằng tài khoản khách hàng 2.Phải có ít nhất 1 sản phẩm trong giỏ hàng	Click vào biểu tượng giỏ hàng	Hiển thị page giỏ hàng	Trung bình	<b>PASSED</b>
9	View Order	Kiểm tra chức năng	1.Người dùng phải đăng nhập bằng tài khoản khách hàng 2.Phải có ít nhất 1 sản phẩm trong giỏ hàng	1.Người dùng truy cập vào giỏ hàng 2. Người dùng nhấn vào nút Order bên dưới trang Cart	Website hiển thị những sản phẩm người dùng đã chọn	Trung bình	<b>PASSED</b>
HƯỚNG DẪN TỰ ĐỘNGIZM							
13	10	View Product Detail	Kiểm tra chức năng	Người dùng phải đăng nhập bằng tài khoản khách hàng	Click sản phẩm muốn xem thông tin	Hiển thị thông tin sản phẩm được chọn	Trung bình <b>PASSED</b>
14	11	Add Products to Cart	Kiểm tra chức năng	Người dùng phải đăng nhập bằng tài khoản khách hàng	Click thêm vào giỏ hàng	Sản phẩm được click được thêm vào giỏ hàng của user	Trung bình <b>PASSED</b>
15	12	Remove Product From Cart	Kiểm tra chức năng	Người dùng phải đăng nhập bằng tài khoản khách hàng	Đang ở page giỏ hàng: Click làm sạch giỏ hàng	Sản phẩm được click được xóa đi	Trung bình <b>PASSED</b>
16	13	Clear Cart	Kiểm tra chức năng	Người dùng phải đăng nhập bằng tài khoản khách hàng	Đang ở page giỏ hàng: Click clear giỏ hàng	Tất cả sản phẩm trong giỏ hàng được xóa đi	Trung bình <b>PASSED</b>
17	14	Edit Order	Kiểm tra chức năng	1.Người dùng phải đăng nhập bằng tài khoản khách hàng 2.Phải có ít nhất 1 sản phẩm trong giỏ hàng	1.Người dùng truy cập vào giỏ hàng 2. Người dùng nhấn vào nút Order bên dưới trang Cart	Ở trang xem lại Order, người dùng có thể tính chỉnh số lượng sản phẩm trong trang này	Trung bình <b>FAILED</b>
18	15	Send Order	Kiểm tra chức năng	1.Người dùng phải đăng nhập bằng tài khoản khách hàng 2.Phải có ít nhất 1 sản phẩm trong giỏ hàng	1.Người dùng truy cập vào giỏ hàng 2. Người dùng nhấn vào nút Order bên dưới trang Cart 3.Chọn phương thức thanh toán và nhấn Order	Website gửi thông tin đơn hàng của khách lên Database, clear giỏ hàng và chuyển qua màn hình cảm ơn khách hàng	Trung bình <b>PASSED</b>
HƯỚNG DẪN TỰ ĐỘNGIZM							
16	Cancel Order	Kiểm tra chức năng	Người dùng phải đăng nhập bằng tài khoản khách hàng	1.Người dùng truy cập vào giỏ hàng 2. Người dùng nhấn vào nút Order bên dưới trang Cart 3.Chọn phương thức thanh toán và không nhấn Order	Order sẽ không gửi lên Database, và giỏ hàng sẽ còn nguyên	Trung bình	<b>PASSED</b>
17	Comment product	Kiểm tra chức năng	Người dùng phải đăng nhập bằng tài khoản khách hàng	Đang ở page thông tin sản phẩm: 1. Click textbox comment 2. Nhập thông tin 3. Click gửi	Thông tin được nhập sẽ hiển thị tại mục comment của sản phẩm	Trung bình	<b>PASSED</b>
18	Rating Product	Kiểm tra chức năng	Người dùng phải đăng nhập bằng tài khoản khách hàng	Đang ở page thông tin sản phẩm: 1. Click rating	Cập nhật rating vừa được thêm	Trung bình	<b>PASSED</b>

#### 7.1.4 Test case người dùng khách hàng chưa đăng nhập

TC ID	Tiêu Đề	Test Type	Điều kiện tiên quyết	Các bước thực hiện	Kết quả thực tế	Mức độ quan trọng	Đánh giá(*)
<b>USER NOT LOGGED IN</b>							
1	Login	Kiểm tra chức năng	Người dùng phải truy cập Website	1. Người dùng vẫn vào nút "Sign in" trên thành Header của Website 2. Nhấn vào tag "Sign in" 3. Nhập đúng E-Mail và password của khách hàng	Website ghi nhận lại thông tin của khách hàng, trả về trang chủ và xuất ra câu chào trên thành Header Website	Trung bình	N/A
2	Register	Kiểm tra chức năng	Người dùng phải truy cập Website	1. Người dùng vẫn vào nút "Sign in" trên thành Header của Website 2. Nhấn vào tag "Register" 3. Nhập các thông tin và nhấn Register	Website trả về tag Login và xuất ra câu Đăng ký thành công Website gửi 1 E-Mail về E-Mail đã đăng ký thông báo đã đăng ký thành công	Trung bình	PASSED
2	Search Product	Kiểm tra chức năng	1.Người dùng phải truy cập Website 2.Phải có ít nhất 1 Product	1. Click vào thanh tìm kiếm 2. Nhập từ khóa về sản phẩm bạn muốn tìm kiếm 3. Click tiếp vào biểu tượng tìm kiếm	Hiển thị sản phẩm bạn muốn tìm kiếm với từ khóa đã nhập	Cao	PASSED
3	Search Category	Kiểm tra chức năng	1.Người dùng phải truy cập Website 2.Phải có ít nhất 1 Product	1. Click vào Combobox Danh sách Category 2. Nhấn vào 1 Category trong danh sách	Website hiển thị toàn bộ sản phẩm thuộc Category đã chọn	Cao	PASSED
4	Search Brand	Kiểm tra chức năng	1.Người dùng phải truy cập Website 2.Phải có ít nhất 1 Product	1. Click vào Combobox Danh sách Brand 2. Nhấn vào 1 Brand trong danh sách	Website hiển thị toàn bộ sản phẩm thuộc Brand đã chọn	Cao	PASSED
5	View Product	Kiểm tra giao diện	1.Người dùng phải truy cập Website 2. Phải có ít nhất 1 sản phẩm có Status là 0	1.Truy cập Website 2.Vào tag Hot Product hoặc tag All Product	Website hiển thị toàn bộ sản phẩm	Cao	PASSED
6	View Category	Kiểm tra giao diện	Người dùng phải truy cập Website	1. Click vào Combobox Danh sách Category	Combobox phải hiện ra toàn bộ Category mà Admin cho phép hiển thị	Trung bình	PASSED
7	View Brand	Kiểm tra giao diện	Người dùng phải truy cập Website	1. Click vào Combobox Danh sách Brand	Combobox phải hiện ra toàn bộ Brand mà Admin cho phép hiển thị	Trung bình	PASSED
8	View Cart	Kiểm tra chức năng	1.Người dùng phải truy cập Website 2.Phải có ít nhất 1 sản phẩm trong giỏ hàng	Click vào biểu tượng giỏ hàng	Hiển thị page giỏ hàng	Trung bình	PASSED
10	View Product Detail	Kiểm tra chức năng	Người dùng phải truy cập Website	Click sản phẩm muốn xem thông tin	Hiển thị thông tin sản phẩm được chọn	Trung bình	PASSED
11	Add Products to Cart	Kiểm tra chức năng	Người dùng phải truy cập Website	Click thêm vào giỏ hàng	Sản phẩm được click được thêm vào giỏ hàng của user	Trung bình	PASSED
12	Remove Product From Cart	Kiểm tra chức năng	Người dùng phải truy cập Website	Đang ở page giỏ hàng: Click làm sạch giỏ hàng	Sản phẩm được click được xóa đi	Trung bình	PASSED

## CHƯƠNG 3: TỔNG KẾT

### 1. Phân công thành viên

Loại người dùng	ID chức năng	Chức năng	Phân công	Hoàn thành
Admin User	<b>1.1</b>	Logout	Lê Trung Hiếu	100%
	<b>1.2</b>	Search Product	Cao Trung Hiếu	100%
	<b>1.3</b>	Search Category	Cao Minh Hiếu	100%
	<b>1.4</b>	Search Brand	Cao Minh Hiếu	100%
	<b>1.5</b>	View Category	Trần Hữu Phúc	100%
	<b>1.6</b>	Add New Category	Trần Hữu Phúc	100%
	<b>1.7</b>	Delete Category	Trần Hữu Phúc	100%
	<b>1.8</b>	Edit Category	Trần Hữu Phúc	100%
	<b>1.9</b>	View Brand	Trần Hữu Phúc	100%
	<b>1.10</b>	Add New Brand	Trần Hữu Phúc	100%
	<b>1.11</b>	Delete Brand	Trần Hữu Phúc	100%
	<b>1.12</b>	Edit Brand	Trần Hữu Phúc	100%
	<b>1.13</b>	View Product	Trần Hữu Phúc	100%
	<b>1.14</b>	Add New Product	Trần Hữu Phúc	100%
	<b>1.15</b>	Delete Product	Trần Hữu Phúc	100%
	<b>1.16</b>	Edit Product	Trần Hữu Phúc	100%
	<b>1.17</b>	View Order List	Lê Trung Hiếu	100%
	<b>1.18</b>	View Order Detail	Lê Trung Hiếu	100%
	<b>1.19</b>	Processing Order	Lê Trung Hiếu	100%
	<b>1.20</b>	Delete Order	Lê Trung Hiếu	100%

<b>Logged in user</b>	<b>2.1</b>	Logout	Lê Trung Hiếu	100%
	<b>2.2</b>	Search Products	Cao Trung Hiếu	100%
	<b>2.3</b>	Get products By Category	Cao Trung Hiếu	100%
	<b>2.4</b>	Get products By Brand	Cao Trung Hiếu	100%
	<b>2.5</b>	View Products	Cao Trung Hiếu	100%
	<b>2.6</b>	View Category	Cao Trung Hiếu	100%
	<b>2.7</b>	View Brand	Cao Trung Hiếu	100%
	<b>2.8</b>	View Cart	Lê Trung Hiếu	100%
	<b>2.9</b>	View Order	Lê Trung Hiếu	100%
	<b>2.10</b>	View Product Detail	Lê Trung Hiếu	100%
	<b>2.11</b>	Add Products to Cart	Lê Trung Hiếu	100%
	<b>2.12</b>	Remove Product From Cart	Lê Trung Hiếu	100%
	<b>2.13</b>	Clear Cart	Lê Trung Hiếu	100%
	<b>2.14</b>	Edit Order	Lê Trung Hiếu	100%
	<b>2.15</b>	Send Order	Lê Trung Hiếu	100%
	<b>2.16</b>	Cancel Order	Lê Trung Hiếu	100%
	<b>2.17</b>	Comment product	Cao Trung Hiếu	100%
	<b>2.18</b>	Rating Product	Cao Turgng Hiếu	100%

User not logged in	<b>3.1</b>	Login	Lê Trung Hiếu	100%
	<b>3.2</b>	Register	Lê Trung Hiếu	100%
	<b>3.3</b>	Search Product	Cao Tùng Hiếu	100%
	<b>3.4</b>	Get products By Category	Cao Tùng Hiếu	100%
	<b>3.5</b>	Get products By Brand	Cao Tùng Hiếu	100%
	<b>3.6</b>	View Products	Cao Tùng Hiếu	100%
	<b>3.7</b>	View Category	Cao Tùng Hiếu	100%
	<b>3.8</b>	View Brand	Cao Tùng Hiếu	100%
	<b>3.9</b>	View Cart	Lê Trung Hiếu	100%
	<b>3.10</b>	View Product Detail	Lê Trung Hiếu	100%
	<b>3.11</b>	Add Products to Cart	Lê Trung Hiếu	100%
	<b>3.12</b>	Remove Product From Cart	Lê Trung Hiếu	100%

## 2. Thuận lợi khó khăn

### 2.1 Khó khăn

- Khó khăn trong bước đầu cài đặt Laravel và tìm kiếm tài liệu
- Khó khăn trong mùa dịch nhất là càng về những tuần cuối. Vì không gặp trực tiếp để trao đổi được cũng như hỏi ý kiến tham khảo từ giảng viên.
- Còn hạn chế trong việc tìm đọc tài liệu bằng tiếng anh.

### 2.2 Thuận lợi

- Thời gian làm project khá thong thả.
- Đội nhóm phối hợp khá ăn ý.
- Các thành viên nắm lượng kiến thức đủ để hoàn thành khối lượng công việc được giao.
- Tài liệu nhiều trên mạng cùng với sự hướng dẫn tận tình của giảng viên.
- Các nhóm bạn cũng rất nhiệt tình khi hỗ trợ sửa lỗi.

### 3. Kết quả đạt được

- Hoàn thành được các chức năng đã đề ra.
- Hiểu rõ hơn về kiến thức đang học cùng với những kiến thức mới.
- Rèn luyện kỹ năng đội nhóm.
- Cải thiện mối quan hệ giữa các thành viên trong nhóm cũng như các thành viên trong lớp.
- Các tính năng cả nhóm cùng nhau làm:
  1. Lên ý tưởng đồ án
  2. Thiết kế Database
  3. Thiết kế Use case
  4. Tìm kiếm tư liệu, giao diện
  5. File báo cáo đồ án
- Các công việc nhóm làm thêm:

Tên công việc	Tên thành viên làm	Tình trạng
Thiết kế Logo cho Website	Cao Minh Hiếu	Hoàn thành
Viết Use case	Cao Minh Hiếu	Hoàn thành
Test Use case	Cao Minh Hiếu	Hoàn thành
Đẩy Website từ Localhost sang Host	Cao Trung Hiếu	Hoàn thành
Gửi E-Mail tới người dùng vừa mới đăng ký tài khoản	Lê Trung Hiếu	Hoàn thành
Làm file báo cáo PowerPoint	Cao Minh Hiếu	Hoàn Thành

### 4. Đánh giá thành viên

Tên thành viên	Chức vụ	Mức độ hoàn thành độ án (Trên tổng 100% )
Lê Trung Hiếu	Nhóm trưởng	30%
Cao Trung Hiếu	Thành viên	30%
Cao Minh Hiếu	Thành viên	20%
Trần Hữu Phúc	Thành viên	20%

## TÀI LIỆU THAM KHẢO

- Laravel.com/docs/7.x
- Tự thiết kế giỏ hàng  
<https://www.youtube.com/playlist?list=PLZusWOyEknKys0Roo4pLVNRXBZZN4QiIE>
- Sử dụng giỏ hàng của Laravel  
[https://packagist.org/packages/bumbummen99/shoppingcart?fbclid=IwAR0Y5BRk2IxBYkJ84oHmyTuu1goljiK040t2wSyekq\\_UEg4fWYcrcu3Mmj0](https://packagist.org/packages/bumbummen99/shoppingcart?fbclid=IwAR0Y5BRk2IxBYkJ84oHmyTuu1goljiK040t2wSyekq_UEg4fWYcrcu3Mmj0)
- Lưu giữ project và trao đổi với nhau: Github, Google Drive, Message, Zalo
- Tìm hiểu về sql của Laravel: <https://laravel.com/docs/7.x/queries>
- Để hiểu rõ hơn về Laravel, nên lên youtube để xem các Youtuber hướng dẫn chi tiết, và có thể tua lại nếu chưa nắm rõ
- Sửa lỗi thông qua: <https://stackoverflow.com/>