

BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH  
KHOA ĐÀO TẠO CHẤT LƯỢNG CAO



**BÁO CÁO ĐỒ ÁN CÔNG NGHỆ PHẦN MỀM MỚI**

***Đề tài:***

**XÂY DỰNG WEBSITE BÁN ĐỒ THỂ THAO**

***Giảng viên hướng dẫn: Thầy Nguyễn Minh Đạo***

***Lớp: Công nghệ phần mềm mới***

**Nhóm: 09**

***Sinh viên:***

**Nguyễn Lâm Gia Khang 18110132**

**Nguyễn Thành Công 18110086**

**Võ Anh Huy 18110124**

*TP. Hồ Chí Minh, tháng 12, năm 2021*

## Mục lục

<b>MỤC LỤC HÌNH ẢNH.....</b>	<b>3</b>
<b>NỘI DUNG .....</b>	<b>5</b>
I. GIỚI THIỆU ĐỀ TÀI .....	5
1. Lý do chọn đề tài .....	5
2. Mục tiêu đề tài.....	5
3. Phương pháp thực hiện .....	5
4. Khảo sát thực trạng .....	5
4.1. Sporter.vn .....	5
4.2. Sport1.vn .....	7
4.3. Các vấn đề cần rút kết và dự định mới .....	8
II. QUÁ TRÌNH THỰC HIỆN.....	9
1. Tài liệu đặc tả yêu cầu .....	9
2. Kiến trúc hệ thống .....	12
2.1. Mô hình kiến trúc .....	12
2.2. Ứng dụng mô hình mvc.....	14
2.3. Phân tích hệ thống back-end .....	16
2.4. Phân tích hệ thống front-end .....	18
3. Sơ đồ usecase.....	21
4. Thiết kế cơ sở dữ liệu .....	23
III. THỬ NGHIỆM HỆ THỐNG.....	31
1. Thử nghiệm tính năng khách vãng lai .....	31
1.1. Xem sản phẩm .....	31
1.2. Xem chi tiết sản phẩm .....	32
1.3. Đăng ký .....	33
2. Tính năng khách hàng .....	34
1. Đăng nhập.....	34
2. Xem thông tin tài khoản .....	36
3. Thêm vào giỏ hàng .....	38
4. Đặt hàng .....	40
5. Thanh toán bằng tiền mặt .....	41
6. Thanh toán bằng paypal .....	42
7. Xem lịch sử đặt hàng .....	43
8. Xem chi tiết đơn hàng .....	44
9. Bình luận .....	45
10. Chính sửa thông tin tài khoản.....	46
3. Tính năng người bán hàng .....	47
1. Tạo cửa hàng .....	47
2. Chính sửa cửa hàng .....	48
3. Thêm sản phẩm quần áo.....	50
4. Chính sửa thông tin sản phẩm .....	52

5. Thêm sản phẩm giày dép.....	53
6. Thêm sản phẩm phụ kiện .....	55
7. Xử lý đơn hàng .....	57
8. Thống kê .....	59
<b>4. Tính năng người giao hàng .....</b>	<b>60</b>
1. Lấy thông tin đơn hàng cần giao .....	60
2. Đơn hàng đã nhận.....	61
3. Đơn hàng đã giao.....	62
5. Tính năng admin.....	64
1. Quản lý người dùng.....	64
2. Quản lý người bán .....	65
3. Duyệt người giao hàng .....	66
4. Quản lý cửa hàng .....	67
5. Quản lý đơn hàng .....	68
6. Quản lý sản phẩm .....	69
<b>IV. SỬ DỤNG GITHUB .....</b>	<b>70</b>
<b>V. CÁC THƯ VIỆN ĐÃ SỬ DỤNG .....</b>	<b>71</b>
1. <i>Backend</i> .....	71
2. Front-end .....	71
<b>VI. KẾT LUẬN .....</b>	<b>72</b>
4.1. Đánh giá mức độ hoàn thành.....	72
4.2. Khó khăn .....	72
4.3. Hướng phát triển.....	73
<b>VII. TÀI LIỆU THAM KHẢO .....</b>	<b>73</b>

## Mục lục hình ảnh

Hình 1. 1: sporter.vn .....	6
Hình 1. 2: sport1.vn .....	7
Hình 2. 1: Mô hình mvc .....	12
Hình 2. 2: Front-end .....	13
Hình 2. 3: backend.....	13
Hình 2. 4: database .....	14
Hình 2. 5: DAO .....	14
Hình 2. 6: Service .....	15
Hình 2. 7: model .....	15
Hình 2. 8: controller .....	15
Hình 2. 9: view .....	16
Hình 2. 10: folder bin .....	16
Hình 2. 11: folder config .....	16
Hình 2. 12: folder middleware.....	17
Hình 2. 13: folder model .....	17
Hình 2. 14: folder routes.....	17
Hình 2. 15: folder schemas .....	17
Hình 2. 16: folder services.....	18
Hình 2. 17: folder admin .....	18
Hình 2. 18: folder assets .....	19
Hình 2. 19: folder body .....	19
Hình 2. 20: folder footer .....	19
Hình 2. 21: folder header.....	19
Hình 2. 22: folder rest.....	20
Hình 2. 23: folder seller .....	20
Hình 2. 24: folder shipper.....	21
Hình 2. 25: folder utils.....	21
Hình 2. 26: folder user .....	21
Hình 2. 27: sơ đồ usecase .....	22
Hình 2. 28: Cơ sở dữ liệu .....	24
Hình 2. 29: schema user .....	25
Hình 2. 30: schema seller .....	25
Hình 2. 31: Schema store.....	26
Hình 2. 32: Schema product .....	26
Hình 2. 33: Schema comment .....	27
Hình 2. 34: Schema category clothes .....	27
Hình 2. 35: Schema category shoes.....	28
Hình 2. 36: Schema category accessories.....	28
Hình 2. 37: Schema order .....	29
Hình 2. 38: Schema category order detail .....	30
Hình 2. 39: shipper schema .....	30

Hình 3. 1: Xem sản phẩm .....	32
Hình 3. 2: xem chi tiết sản phẩm .....	33
Hình 3. 3: Đăng ký .....	34
Hình 3. 4: Đăng nhập.....	36
Hình 3. 5: xem thông tin tài khoản .....	37
Hình 3. 6: thêm vào giỏ hàng .....	38
Hình 3. 7: Đặt hàng .....	41
Hình 3. 8: Thanh toán bằng tiền mặt .....	42
Hình 3. 9: Thanh toán bằng paypal .....	43
Hình 3. 10: xem lịch sử đặt hàng.....	44
Hình 3. 11: xem chi tiết đơn hàng .....	45
Hình 3. 12: Bình luận .....	46
Hình 3. 13: Chính sửa thông tin khách hàng .....	47
Hình 3. 14: Tạo cửa hàng .....	48
Hình 3. 15: Chính sửa cửa hàng .....	50
Hình 3. 16: thêm sản phẩm quần áo .....	51
Hình 3. 17: thêm sản phẩm quần áo .....	52
Hình 3. 18: chỉnh sửa thông tin sản phẩm .....	53
Hình 3. 19: thêm sản phẩm giày dép .....	55
Hình 3. 20: thêm sản phẩm giày dép .....	55
Hình 3. 21: thêm sản phẩm phụ kiện .....	57
Hình 3. 22: Thêm sản phẩm phụ kiện.....	57
Hình 3. 23: xử lý đơn hàng.....	59
Hình 3. 24: thống kê .....	60
Hình 3. 25: Đơn hàng cần giao .....	61
Hình 3. 26: đơn hàng đã nhận.....	62
Hình 3. 27: Đơn hàng đã giao.....	64
Hình 3. 28: Quản lý người dùng .....	65
Hình 3. 29: Quản lý người bán .....	66
Hình 3. 30: duyệt người giao hàng .....	67
Hình 3. 31: Quản lý cửa hàng.....	68
Hình 3. 32: Quản lý đơn hàng .....	69
Hình 3. 33: Quản lý sản phẩm .....	70

## NỘI DUNG

### I. Giới thiệu về tài

#### 1. Lý do chọn đề tài

Chuyển đổi số đang là xu hướng của thế giới và đất nước chúng ta cũng không ngoại lệ. Việc chuyển đổi số giúp các công ty cũng như các cửa hàng có thể thích ứng trong mùa dịch covid-19. Tuy nhiên, có rất ít cửa hàng có thể chuyển đổi số thành công do các vấn đề về kinh phí cũng như nguồn nhân lực. Chính vì điều này nhóm chúng em xin phép chọn đề xây dựng website bán đồ thể thao nhằm mục đích cung cấp ứng dụng có thể tích hợp và phát triển hệ thống cửa hàng bán đồ thể thao cũng như hỗ trợ một phần nhỏ trong việc chuyển đổi số của các cửa hàng.

#### 2. Mục tiêu đề tài

Chương trình cung cấp một ứng dụng đơn giản, dễ sử dụng, có thể dễ dàng tạo ra các cửa hàng để tham gia vào việc kinh doanh và quản trị trên hệ thống. Các tính năng hỗ trợ sẽ liên tục được cập nhật trong các phiên bản sắp tới.

#### 3. Phương pháp thực hiện

Ứng dụng sử dụng công nghệ mới MERN. Ngoài ra nhóm cũng đã tham khảo một số website bán đồ thể thao như shopee, sporter, sporter1... để giúp chương trình dễ dàng sử dụng hơn. Chương trình cung cấp một website đơn giản dễ dàng thao tác và sử dụng. Ngoài ra hệ thống cung cấp thêm các chức năng cho việc quản lý cửa hàng dễ dàng. Thiết kế và thực hiện theo mô hình MVC và 3-tier để có thể nâng cấp và sửa chữa trong quá trình sử dụng.

#### 4. Khảo sát thực trạng

##### 4.1. Sporter.vn



Hình 1. 1: sporter.vn

### ✚ Giới thiệu:

Hệ thống bán lẻ Sporter.vn đồng hành cùng bạn giúp bạn nâng cao khả năng, trình độ thể thao.

Các sản phẩm, dịch vụ của sporter.vn:

- Nhận đặt áo bóng đá đội tuyển, câu lạc bộ.
- Bán lẻ giày thể thao các loại.
- Bán lẻ quần short thể thao Nike chính hãng.
- Bán lẻ áo thun thể thao chính hãng: Nike, Adidas, Puma,...
- Các phụ kiện thể thao cho các môn: Bóng đá, Cầu lông, Bóng chuyền, Tennis,...

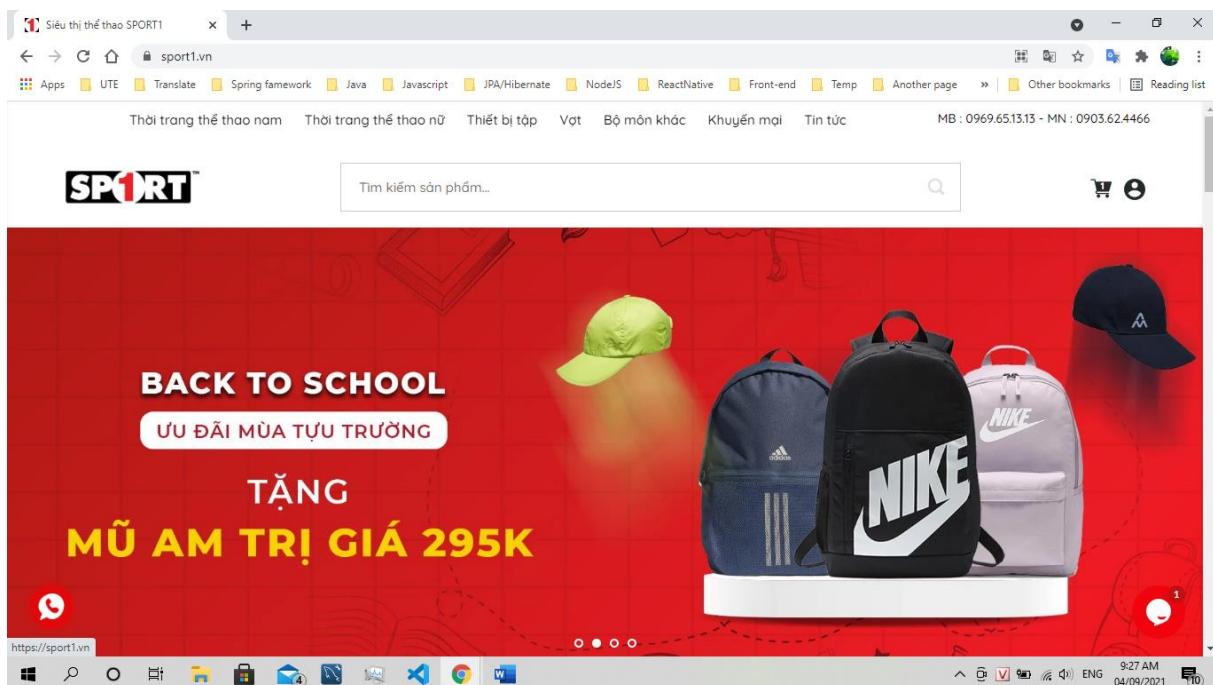
### ✚ Ưu điểm

- Bộ cục rõ ràng phù hợp với mục tiêu bán đồ thể thao.
- Có thể nhanh chóng đặt mua mà không cần đăng ký tài khoản.
- Tốc độ tái trang nhanh.
- Thông tin cho các sản phẩm được thể hiện đầy đủ.

### ✚ Nhược điểm

- Đối với các sản phẩm chưa được bán vẫn có thể đặt hàng được thay vì chỉ xem qua thông tin hoặc đặt hàng trước.
- Chức năng thanh toán chưa tích hợp các ví điện tử, khi muốn thanh toán online phải liên hệ trực tiếp với shop.
- Không lựa chọn được kích cỡ.
- Các sản phẩm đã hết hàng và các sản phẩm chưa được bán nằm xen kẽ với nhau và nằm chung với các sản phẩm đang bán gây cảm giác rối mắt.
- Đối với một số danh mục không thể hiện rõ sản phẩm dành cho nữ và cho nam.
- Chưa cho phép đặt sản phẩm số lượng lớn.

## 4.2. Sport1.vn



Hình 1. 2: sport1.vn

### Giới thiệu

CÔNG TY TNHH THỂ THAO ĐỨC TRUNG (SPORT1) – trực thuộc Tập Đoàn ĐỨC TRUNG được thành lập và chính thức đi vào hoạt động từ năm 2007. Sự ra đời của siêu thị thể thao mang thương hiệu SPORT1 được đánh giá là một bước tiến quan trọng trong phương thức phân phối các dụng cụ thể thao tới người tiêu dùng.

### Ưu điểm

- Bố cục rõ ràng.
- Thể hiện đầy đủ và chi tiết danh mục sản phẩm.
- Được lựa chọn kích thước sản phẩm.
- Tốc độ tải trang nhanh.
- Thiết kế chức năng tìm kiếm tốt.

#### Nhược điểm

- Chưa tích hợp ví điện tử trong thanh toán.
- Chưa chọn lựa màu sắc sản phẩm.
- Vẫn còn các sản phẩm không có thông tin chi tiết.

### 4.3. Các vấn đề cần rút kinh nghiệm mới

#### Những vấn đề cần lưu ý:

- Thiết kế giao diện phù hợp với người sử dụng.
- Đánh giá tác vụ nặng như load ảnh, ... để tăng tốc độ trang web từ đó tăng trải nghiệm khách hàng.
- Bố cục, màu sắc phù hợp với người sử dụng.
- Thiết kế chức năng đơn giản và phù hợp với các khách hàng.

#### Các ưu điểm đã đúc kết được:

- Xây dựng website đơn giản, bố cục rõ ràng, dễ sử dụng và thích hợp cho nhiều lứa tuổi.
- Dễ dàng thay đổi kích thước, màu sắc sản phẩm, ....
- Đa dạng hóa các loại sản phẩm.
- Tối ưu hóa tốc độ tải trang.
- Tránh các thủ tục rườm rà, phức tạp.
- Dễ dàng phân loại các loại sản phẩm dành cho phái nam và phái nữ.

#### Các nhược điểm cần rút kinh nghiệm:

- Tránh xây dựng trang web rườm rà, phức tạp.
- Không xây dựng các chức năng dư thừa gây lãng phí tài nguyên.

- Không hiển thị các thông tin không liên quan.
  - Hạn chế tối đa việc chèn quá nhiều quảng cáo và các khuyến mãi.
-  **Đúc kết thêm những dự định mới**
- Xây dựng thêm chức năng cho phép các cửa hàng tham gia buôn bán trên website.
  - Tích hợp thêm hình thức thanh toán trực tuyến.
  - Thêm các chức năng tùy chọn kích thước, màu sắc.

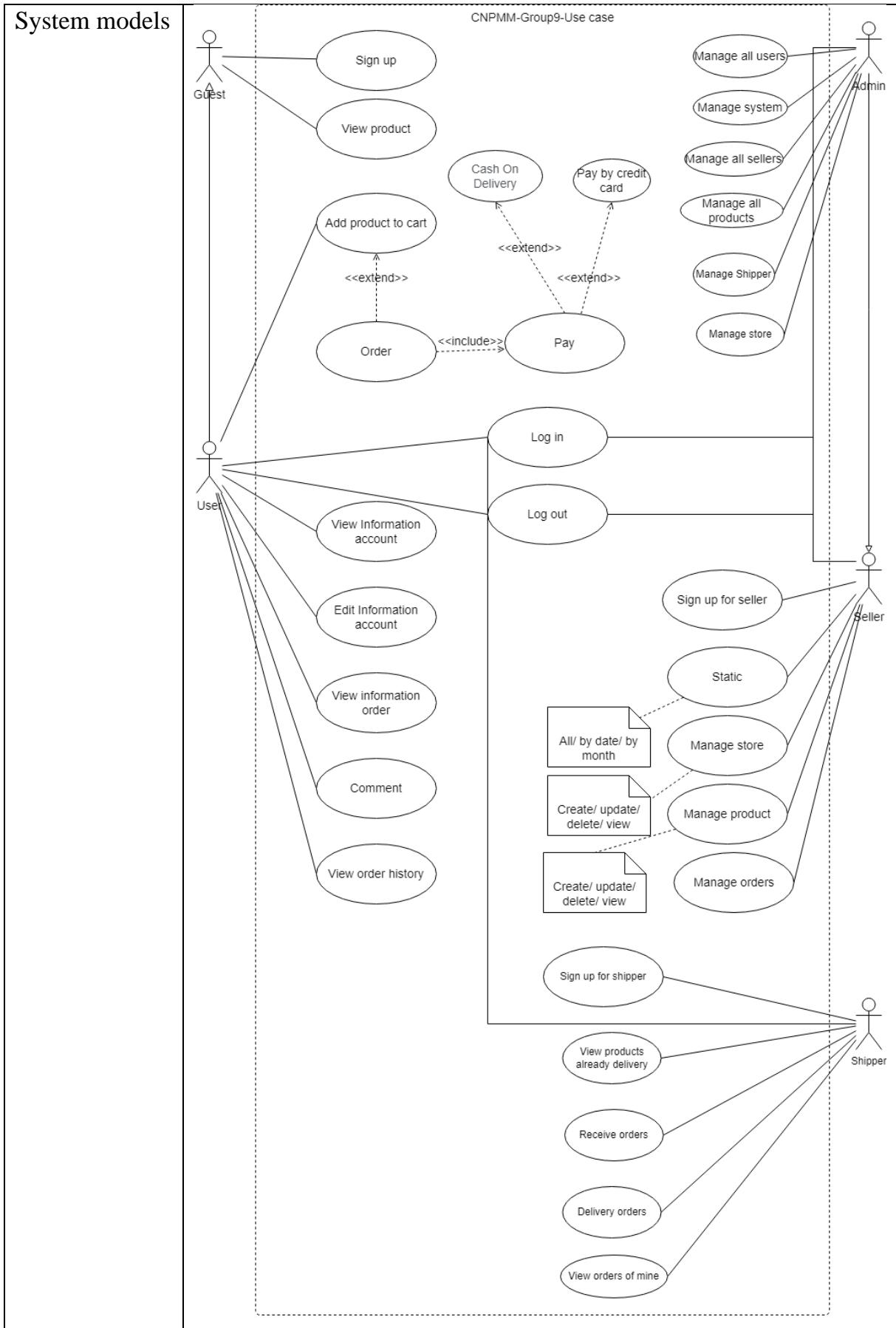
## **II. Quá trình thực hiện**

### **1. Tài liệu đặc tả yêu cầu**

Chapter	Description
Preface	Đây là tài liệu đặc tả dành cho ban lãnh đạo, các bên liên quan nhằm giúp cho mọi người hiểu hơn về dự án đang phát triển. Dự án sẽ ra mắt với những chức năng cơ bản và các chức năng mới sẽ liên tục được cập nhật và phát triển nhằm giúp hệ thống ổn định và phát triển lâu dài.
Introduction	Ứng dụng cung cấp 1 website giúp các cửa hàng bán đồ thể thao có thể thực hiện việc bán đồ online nhằm giúp các cửa hàng có thể chuyển đổi số. Ngoài ra hệ thống đã tích hợp hệ thống giao hàng giúp các cửa hàng có thể giao đơn trực tiếp tới khách hàng.
Glossary	V1.0: phiên bản đầu tiên
User requirements definition	Phiên bản 1.0 sẽ mang đến cho khách hàng các chức năng sau:  <b>Yêu cầu chức năng:</b> Guest: Đăng ký, xem sản phẩm. User: Thêm sản phẩm vào giỏ hàng, mua hàng, thanh toán, đăng nhập, đăng xuất, xem thông tin tài khoản, chỉnh sửa thông tin tài khoản, chỉnh sửa giỏ hàng, bình luận, xem lịch sử mua hàng. Người bán hàng: đăng ký, đăng nhập, đăng xuất, quản lý cửa hàng, quản lý sản phẩm, quản lý đơn hàng, thống kê. Người giao hàng: đăng ký, đăng nhập, đăng xuất, xem các đơn hàng đã chuẩn bị xong, nhận hàng, giao hàng, xem đơn hàng đã giao thành công.

	<p>Người quản trị hệ thống: quản lý tất cả người dùng, quản lý tất cả người bán hàng, quản lý tất cả người giao hàng, quản lý tất cả sản phẩm, quản lý tất cả cửa hàng.</p> <p>➡ Yêu cầu phi chức năng:</p> <ol style="list-style-type: none"> <li>1. Sức chứa khoảng 100 người dùng truy cập cùng lúc.</li> <li>2. Chức năng linh hoạt cho việc đăng nhập giữa các actor khác nhau.</li> <li>3. Khả năng bị tấn công.</li> <li>4. Có khả năng khôi phục dữ liệu khi bị mất.</li> <li>5. Toàn vẹn dữ liệu.</li> </ol>
System architecture	Hệ thống sử dụng các công nghệ mới: Mongodb, nodejs, expressjs, reactjs..
System requirements specification	<p>➡ Yêu cầu chức năng:</p> <ol style="list-style-type: none"> <li>1. Sử dụng mô hình mvc + 3 tier để thiết kế.</li> <li>2. Phân loại người sử dụng (user, seller, shipper và admin).</li> </ol> <p>➡ Yêu cầu phi chức năng</p> <ol style="list-style-type: none"> <li>1. Khả năng bị tấn công (100 session thì có 1 session tấn công).</li> <li>2. Các dữ liệu đã được backup nên dễ dàng phục hồi.</li> <li>3. Giao diện hiện đại phù hợp với tất cả lứa tuổi.</li> </ol>

## System models

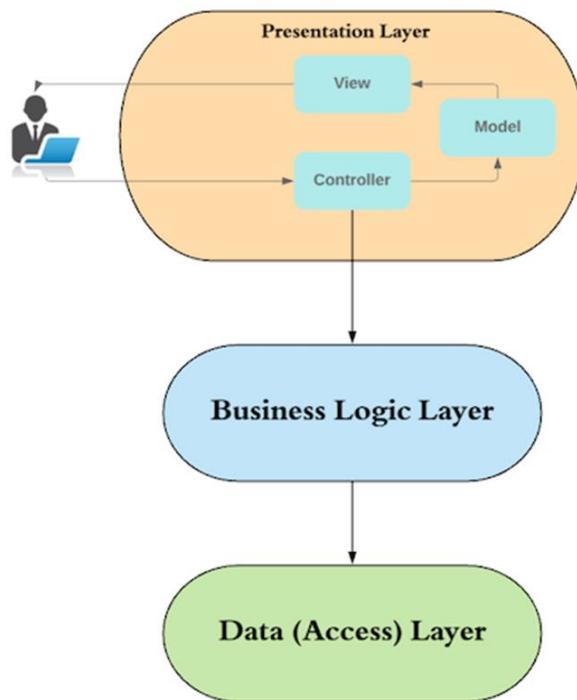


System evolution	1. Tích hợp thêm các chức năng thanh toán bằng paypal... 2. Tích hợp thêm các chức năng thanh toán bằng...
Appendices	Phiên bản phần mềm: v1.0. Hệ quản trị cơ sở dữ liệu: mongodb. Công nghệ sử dụng: nodejs, expressjs, reactjs. Host: Heroku.
Index	

## 2. Kiến trúc hệ thống

### 2.1. Mô hình kiến trúc

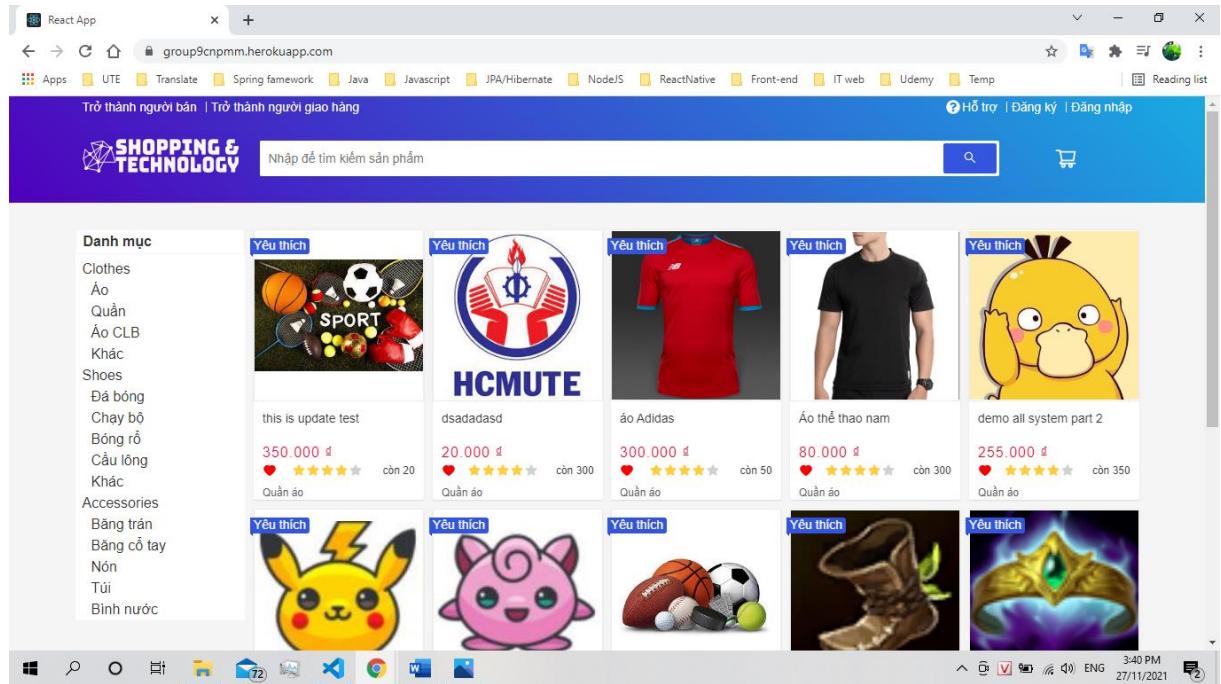
Website sử dụng mô hình kiến trúc mvc + 3 tier thuận lợi cho việc sửa lỗi, phát triển và nâng cấp các chức năng hệ thống.



Hình 2. 1: Mô hình mvc

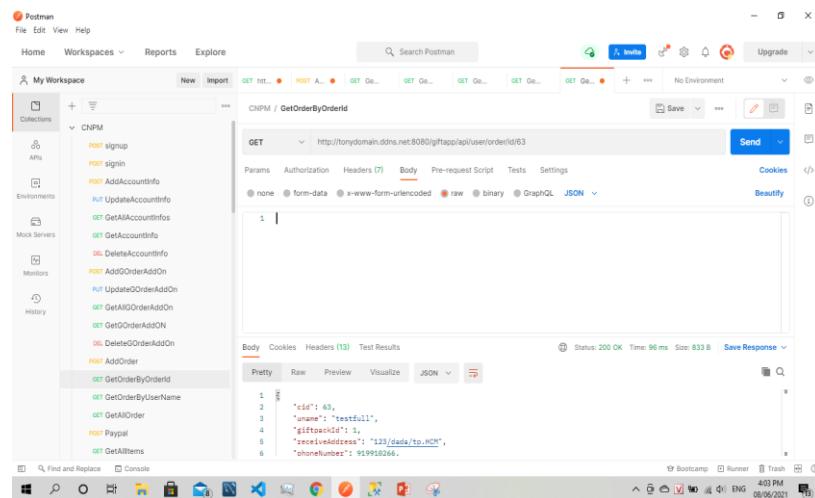
Ứng dụng được chia làm 3 phần: Front-end, backend và database giúp tách biệt và dễ dàng nâng cấp, sửa chữa khi cần thiết.

- Front-end: sử dụng html, css, javascript và reactjs để phát triển.



Hình 2. 2: Front-end

- Backend: Sử dụng ngôn ngữ javascript, nodejs, expressjs và kiểm thử bằng postman.



Hình 2. 3: backend

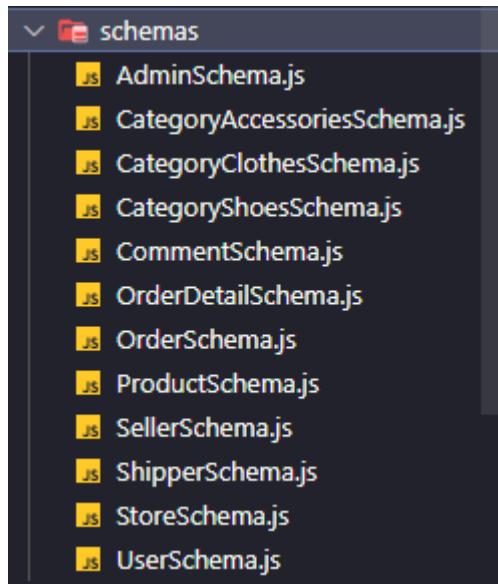
Database: sử dụng hệ cơ sở dữ liệu mongodb và host: mongo atlas.

Collection Name	Documents	Avg. Document Size	Total Document Size	Num. Indexes	Total Index Size	Properties
adminschemas	3	138.0 B	414.0 B	2	41.0 KB	
categoryaccessoriesschemas	2	178.0 B	356.0 B	2	73.7 KB	
categoryclotheschemas	7	238.1 B	1.7 KB	2	73.7 KB	
categoryshoesschemas	3	296.3 B	889.0 B	2	73.7 KB	
commentschemas	8	121.6 B	973.0 B	2	73.7 KB	
identitycounters	18	83.6 B	1.5 KB	2	73.7 KB	
orderdetailschemas	8	208.5 B	1.7 KB	2	73.7 KB	
orderschemas	4	167.0 B	668.0 B	2	73.7 KB	
productschemas	12	257.9 B	3.1 KB	2	73.7 KB	

Hình 2. 4: database

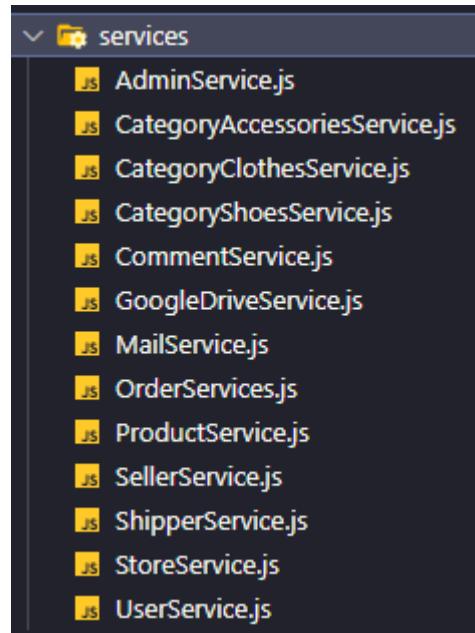
## 2.2. Ứng dụng mô hình mvc

- Data access layer: Tương tác với cơ sở dữ liệu.



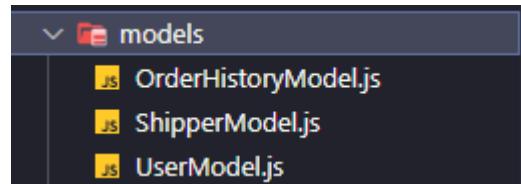
Hình 2. 5: DAO

- Business logic layer: xử lý các nghiệp vụ bài toán.



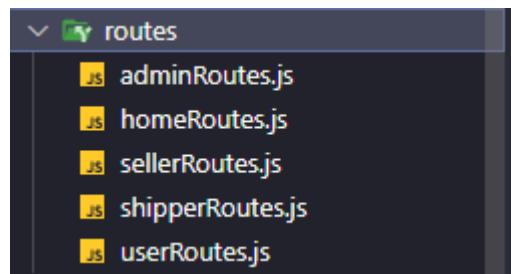
Hình 2. 6: Service

- Model: Khuôn mẫu cho việc hiển thị dữ liệu.



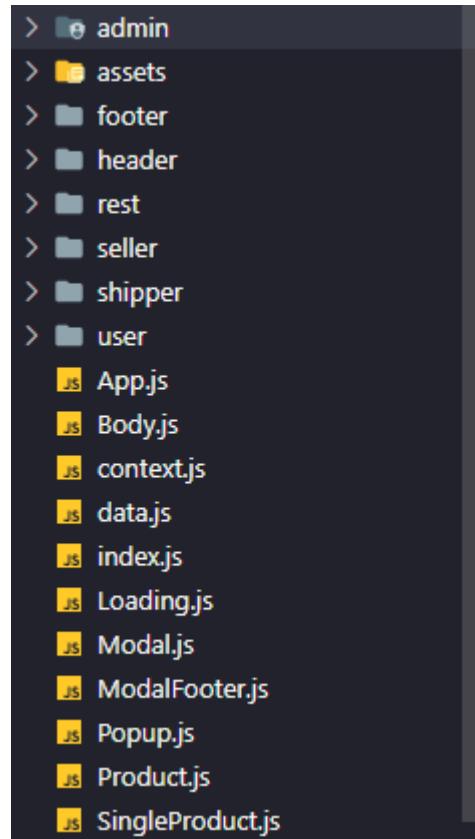
Hình 2. 7: model

- Controller: là cầu nối để thực hiện các phương thức khi nhận một yêu cầu từ client.



Hình 2. 8: controller

- View: Front-end



Hình 2. 9: view

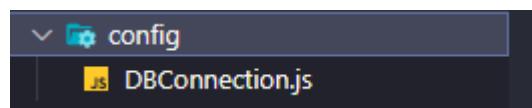
### 2.3. Phân tích hệ thống back-end

- Folder bin: Chứa cấu hình chạy được cả localhost và Heroku.



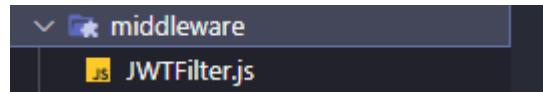
Hình 2. 10: folder bin

- Folder config: Chứa cấu hình kết nối với cơ sở dữ liệu.



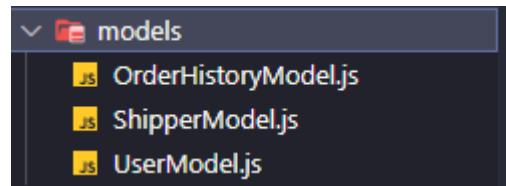
Hình 2. 11: folder config

- Folder middleware: Chứa filter jwt.



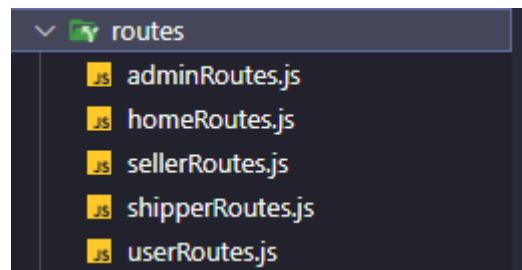
Hình 2. 12: folder middleware

- Folder models: Chứa khuôn mẫu hiển thị dữ liệu.



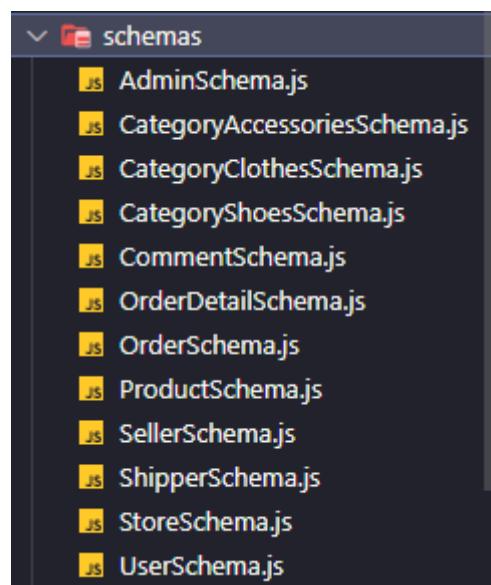
Hình 2. 13: folder model

- Folder routes: Chứa đường dẫn tới các API.



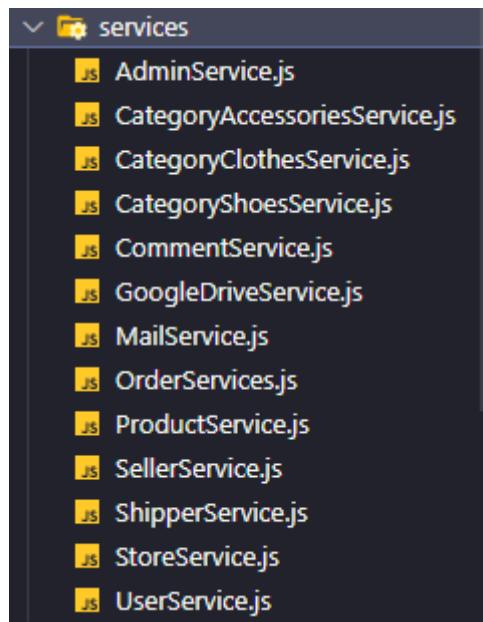
Hình 2. 14: folder routes

- Folder schemas: Thao tác với cơ sở dữ liệu.



Hình 2. 15: folder schemas

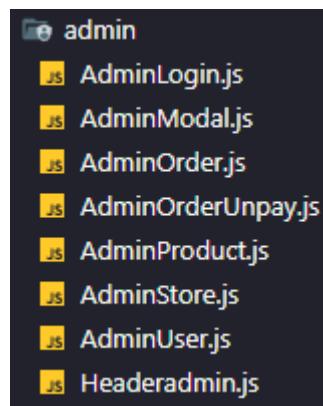
- Folder services: xử lý các logic của hệ thống.



*Hình 2. 16: folder services*

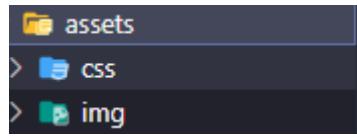
#### 2.4. Phân tích hệ thống front-end

- Admin: Folder admin chứa các file javascript phục vụ cho phần xử lý hiển thị của chức năng admin gồm: đăng nhập, điều hướng, xem tài khóa người dùng, xem cửa hàng, xem đơn hàng, xem sản phẩm...



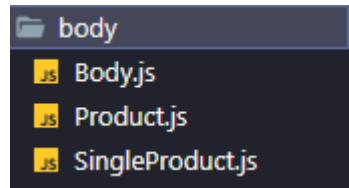
*Hình 2. 17: folder admin*

- Assets: Chứa các file hình ảnh và file css để format cho trang web đẹp hơn.



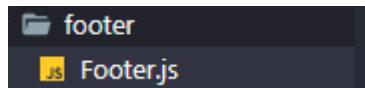
Hình 2. 18: folder assets

- Body: Đây là nơi chứa file javascript phục vụ cho mục đích hiển thị sản phẩm ở trang chủ và hiển thị trang thông tin sản phẩm chi tiết.



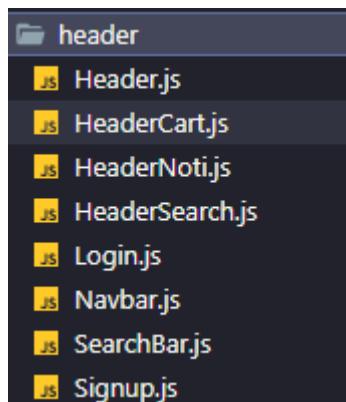
Hình 2. 19: folder body

- Footer: Chứa file phục vụ cho mục đích hiển thị chân trang trong tất cả các trang.



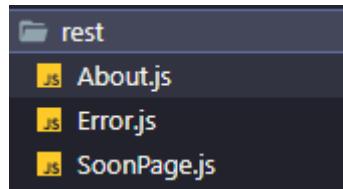
Hình 2. 20: folder footer

- Header: Thư mục chứa các file thuộc phần đầu của trang web. Ở đây có chứa phần hiển thị thanh navbar, thanh search, giỏ hàng... Trang login và signup cũng được để ở đây.



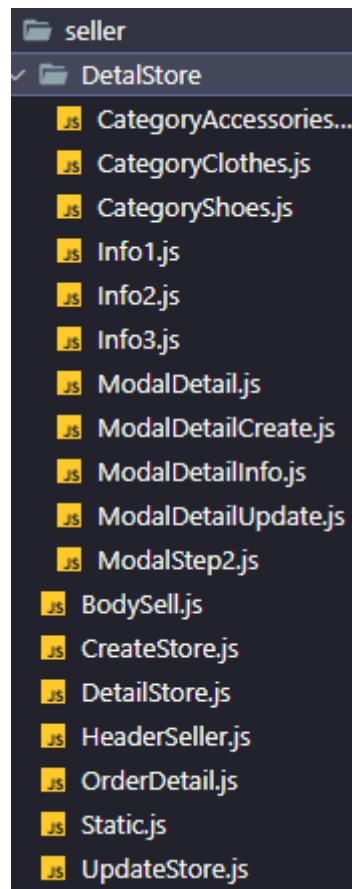
Hình 2. 21: folder header

- Rest: Các trang báo lỗi, trang cung cấp thông tin về web và trang thông báo về chức năng đang phát triển.



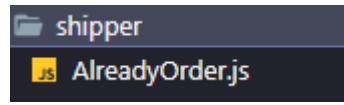
Hình 2. 22: folder rest

- Seller: Ở đây chứa các thành phần liên quan đến việc xử lý hiển thị các chức năng của người bán hàng như: tạo cửa hàng, xóa sửa cửa hàng, xem cửa hàng, tạo sản phẩm...



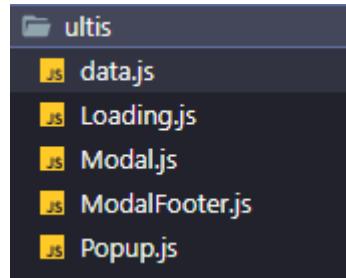
Hình 2. 23: folder seller

- Shipper: Chứa file javascript phục vụ cho mục đích hiển thị, xử lý cho trang của người giao hàng.



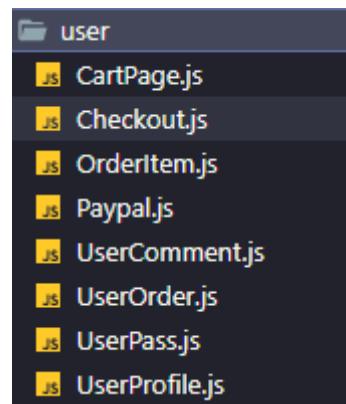
Hình 2. 24: folder shipper

- Utils: Đây là các phần được sử dụng chung cho nhiều trang gồm: hiển thị loading, hiện thông báo, ...



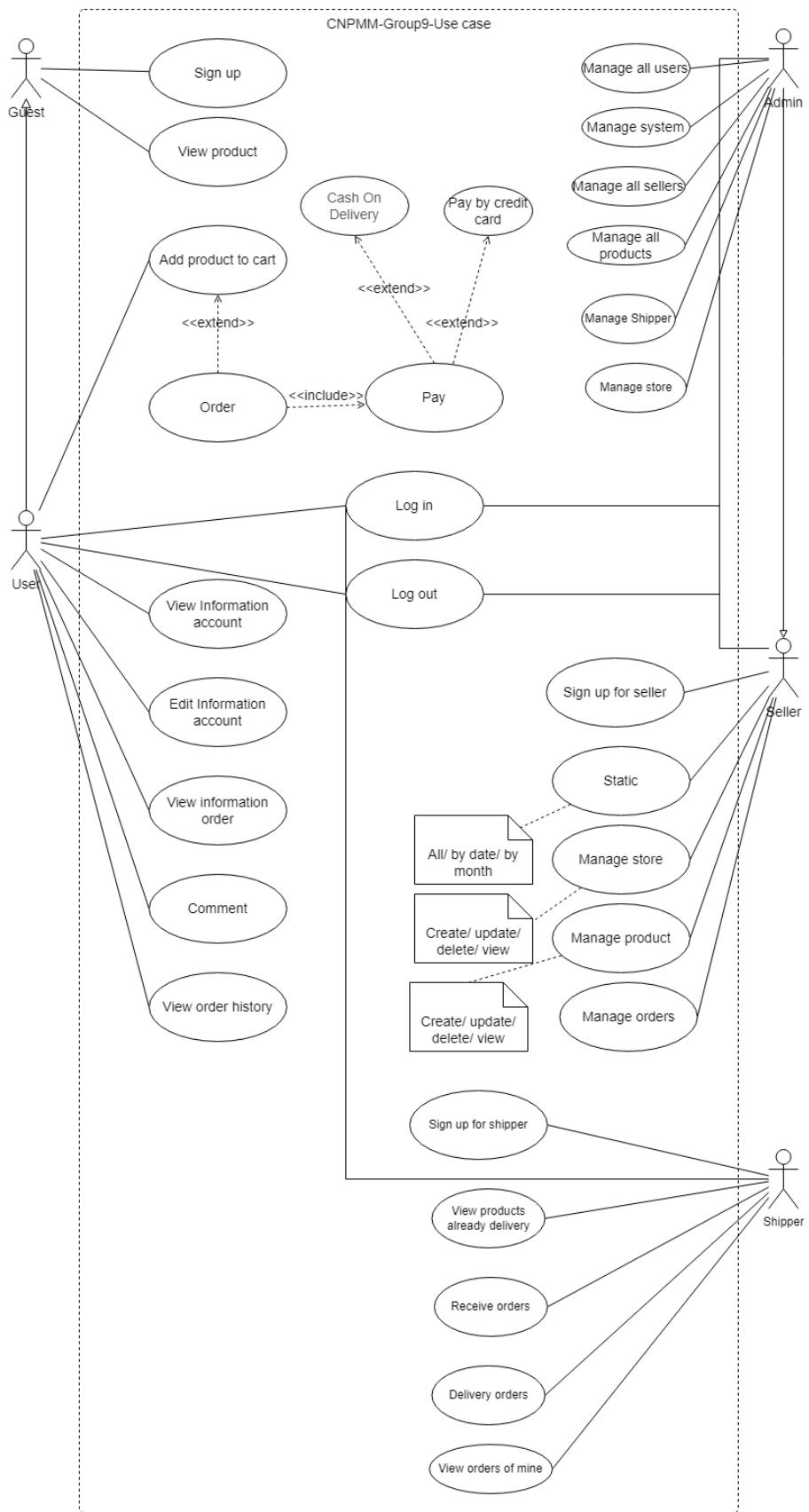
Hình 2. 25: folder utils

- User: Gồm những thành phần liên quan đến người dùng như: xem giỏ hàng, thanh toán, bình luận, thay đổi thông tin cá nhân...



Hình 2. 26: folder user

### 3. Sơ đồ usecase



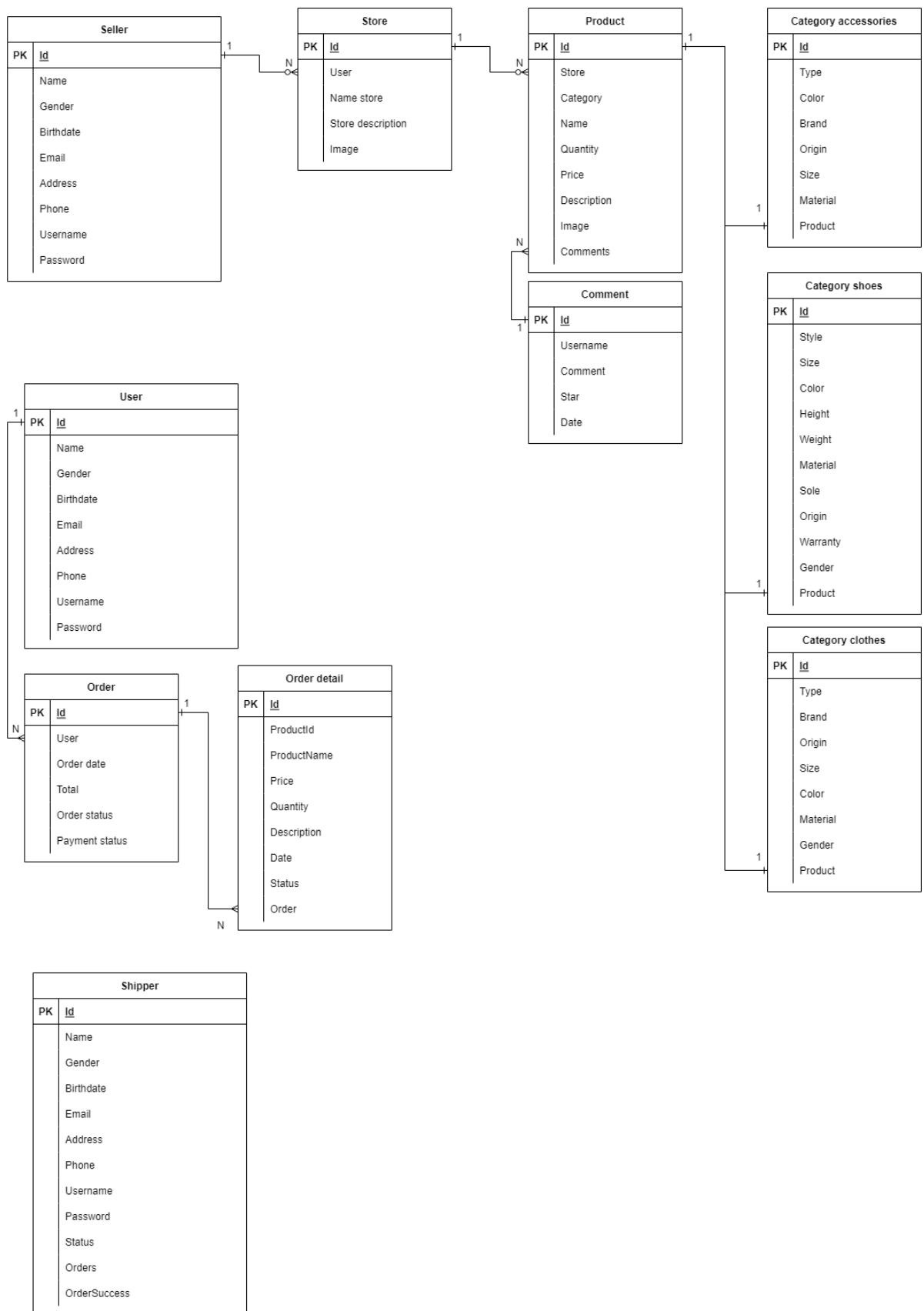
Hình 2. 27: sơ đồ usecase

- Khách hàng chưa có tài khoản có thể thực hiện các chức năng: đăng ký tài khoản và xem các sản phẩm.
- Khách hàng đã có tài khoản có thể thực hiện các chức năng của khách hàng chưa có tài khoản và các chức năng mới như: đăng nhập, đăng xuất, thêm sản phẩm vào giỏ hàng, thực hiện đặt hàng kèm lựa chọn một trong hai phương thức thanh toán (than toán trực tiếp khi nhận hàng hoặc thanh toán thẻ), xem thông tin tài khoản, chỉnh sửa thông tin tài khoản, bình luận sản phẩm và xem lịch sử mua hàng.
- Đối với người bán hàng có thể thực hiện các chức năng như: quản lý cửa hàng của chính mình (thêm, xóa, chỉnh sửa cửa hàng), quản lý sản phẩm của cửa hàng (xem, thêm, xóa, chỉnh sửa), quản lý đơn hàng, thống kê.
- Đối với người giao hàng có thể thực hiện các chức năng như: xem các đơn hàng đã chuẩn bị xong, nhận hàng, giao hàng, xem các đơn hàng đã giao.
- Người quản trị hệ thống có thể thực hiện các chức năng của người bán hàng và các chức năng mới như: quản trị hệ thống, quản lý người bán hàng, quản lý người giao hàng, quản lý sản phẩm, quản lý cửa hàng.

➡ Ảnh thiết kế usecase của hệ thống:

<https://github.com/giakhangnguyenlam/CNPMM/blob/main/System%20design/Use%20case.drawio.png>

#### 4. Thiết kế cơ sở dữ liệu



Hình 2. 28: Cơ sở dữ liệu

- Một khách hàng có thể đăng ký tài khoản như một khách hàng bình thường hoặc trở thành chủ cửa hàng.

```
var UserSchema = new Schema({
  id: Number,
  name: String,
  dateofbirth: String,
  email: String,
  address: String,
  gender: String,
  username: String,
  password: String,
  phone: String
})
```

*Hình 2. 29: schema user*

```
var SellerSchema = new Schema({
  id: Number,
  name: String,
  dateofbirth: String,
  email: string,
  address: String,
  gender: String,
  username: String,
  password: String,
  phone: String
})
```

*Hình 2. 30: schema seller*

- Một chủ cửa hàng có thể tạo ra một hoặc nhiều cửa hàng. Để xác định cửa hàng thuộc quyền sở hữu của chủ cửa hàng nào em sử dụng thêm 1 thuộc tính user.

```
var StoreSchema = new Schema({
  id: Number,
  userId: {
    type: Schema.Types.Number,
    ref: 'UserSchema'
  },
  nameStore: String,
  storeDescription: String,
  image: String
})
```

Hình 2. 31: Schema store

- Một cửa hàng có thể có một hoặc nhiều sản phẩm. Để xác định sản phẩm thuộc sở hữu của cửa hàng nào em sử dụng một thuộc tính store.

```
var ProductSchema = new Schema ({
  id: Number,
  storeId: {
    type: Schema.Types.Number,
    ref: 'StoreSchema'
  },
  category: Number,
  name: String,
  quantity: Number,
  price: Number,
  description: String,
  image: String,
  comments:[{
    type:Schema.Types.Number,
    ref:'CommentsSchema'
  }]
})
```

Hình 2. 32: Schema product

- Một sản phẩm có thể có nhiều bình luận. Để xác định các bình luận thuộc sản phẩm nào em sử dụng embedded vào sản phẩm.

```
var CommentSchema = new Schema ({
  id: Number,
  username: String,
  comment: String,
  star: Number,
  date: String
})
```

Hình 2. 33: Schema comment

- Tùy vào loại sản phẩm có thông tin chi tiết khác nhau. Nhóm em đã phân loại 1 sản phẩm thuộc 1 trong 3 nhóm:
  - + Quần áo: Đôi với quần áo sẽ có thêm các thuộc tính đặc biệt đặc trưng cho nó. Để xác định thuộc tính này thuộc về sản phẩm nào em có sử dụng thêm thuộc tính productId.

```
var CategoryClothesSchema = new Schema({
  id: Number,
  type: String,
  brand: String,
  origin: String,
  size: [String],
  color: [String],
  material: String,
  gender: String,
  productId: [
    type: Schema.Types.Number,
    ref: 'ProductSchema'
  ]
})
```

Hình 2. 34: Schema category clothes

- + Giày dép: Đôi với giày dép sẽ có thêm các thuộc tính đặc biệt đặc trưng cho nó.

```
var CategoryShoesschema = new Schema({
  id: Number,
  style: String,
  size: [Number],
  color: [String],
  height: Number,
  weight: Number,
  material: String,
  sole: String,
  origin: String,
  warranty: Number,
  gender: String,
  productId: {
    type: Schema.Types.Number,
    ref: 'ProductSchema' You,
  }
})
```

Hình 2. 35: Schema category shoes

+ Phụ kiện khác: Đối với phụ kiện sẽ có thêm các thuộc tính đặc biệt đặc trưng cho nó.

```
var CategoryAccessoriesSchema = new Schema({
  id: Number,
  type: String,
  color: [String],
  brand: String,
  origin: String,
  material: String,
  productId: {
    type: Schema.Types.Number,
    ref: 'ProductSchema' You, 24 mi
  }
})
```

Hình 2. 36: Schema category accessories

- Một khách hàng có thể đặt một hoặc nhiều đơn hàng. Để xác định đơn hàng thuộc về khách hàng nào em sử dụng thuộc tính user.

```
var OrderSchema = new Schema({
  id: Number,
  userId: {
    type: Schema.Types.Number,
    ref: 'UserSchema'
  },
  orderDate: String,
  total: Number,
  orderStatus: String,
  paymentStatus: String
})
```

Hình 2. 37: Schema order

- Một đơn hàng có thể có một hoặc nhiều sản phẩm khác nhau. Để xác định sản phẩm thuộc về đơn hàng nào và thuộc sản phẩm nào em sử dụng thêm 2 thuộc tính orderId. Việc tách ra như thế sẽ dễ dàng phân chia lại cho các cửa hàng để chuẩn bị.

```
var OrderDetailsSchema = new Schema({
  id: Number,
  orderId: {
    type: Schema.Types.Number,
    ref: 'OrdersSchema'
  },
  productId: Number,
  productName: String,
  price: Number,
  quantity: Number,
  description: String,
  date: String,
  status: String
})
```

Hình 2. 38: Schema category order detail

- Một shipper có thể nhận và giao đơn hàng.

```
var ShipperSchema = new Schema({
  id: Number,
  name: String,
  dateofbirth: string,
  email: String,
  address: String,
  gender: String,
  username: String,
  password: String,
  phone: String,
  status: String,
  orders: [Number],
  ordersSuccess: [Number]
})
```

Hình 2. 39: shipper schema

- Database của hệ thống đã được triển khai trên atlas. Link MongoDBCompass:

mongodb+srv://khangnguyen:123@cluster0.ddgvr.mongodb.net/myFirstDatabase?retryWrites=true&w=majority

⊕ Source github: <https://github.com/giakhangnguyenlam/CNPMM>

⊕ Hình ảnh cơ sở dữ liệu của hệ thống:

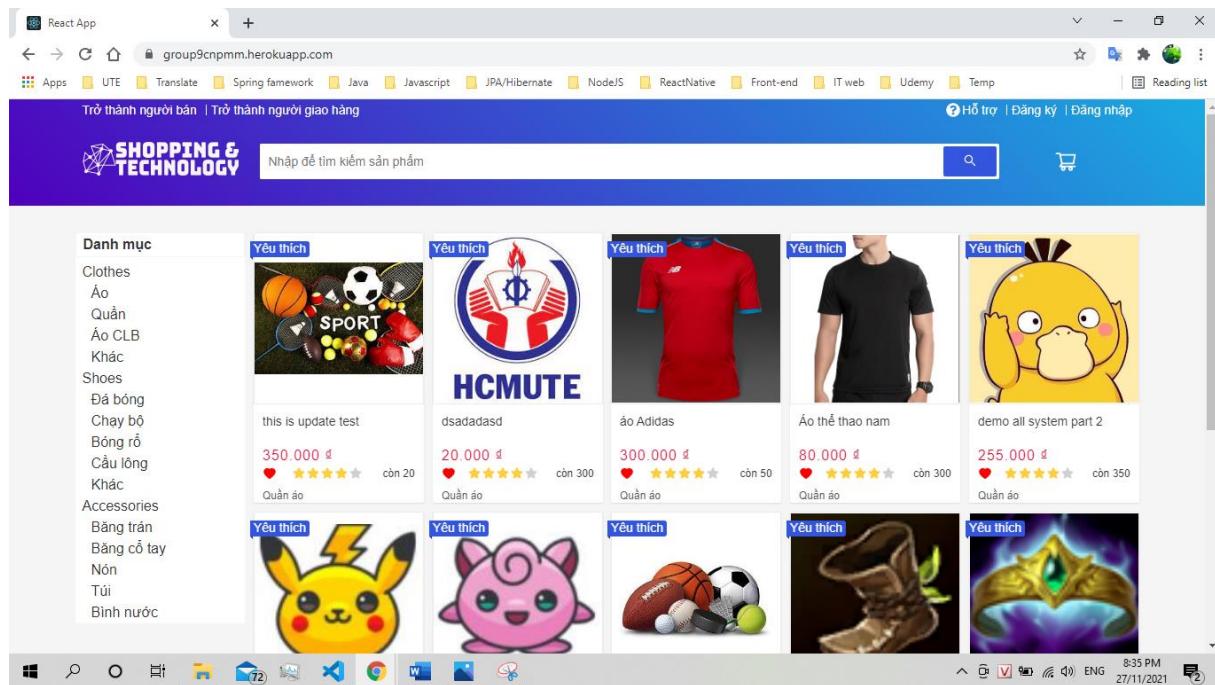
<https://github.com/giakhangnguyenlam/CNPMM/blob/main/System%20design/BaseCNPMM.drawio.png>

### **III. Thủ nghiệm hệ thống**

#### **1. Thủ nghiệm tính năng khách vãng lai**

##### **1.1. Xem sản phẩm**

Name	Xem thông tin sản phẩm
<b>Brief description</b>	Khách hàng truy cập vào trang chủ hệ thống sẽ hiển thị thông tin sản phẩm.
<b>Actor(s)</b>	Guest
<b>Pre-conditions</b>	
<b>Post-conditions</b>	Sản phẩm sẽ được hiển thị trên trang chủ
<b>Flow of events</b>	
Basic flow (Thành công)	Người dùng thực hiện các bước sau: <ol style="list-style-type: none"><li>Truy cập vào trang chủ.</li><li>Hệ thống gọi đến API <a href="https://cnpmbe.herokuapp.com/products">https://cnpmbe.herokuapp.com/products</a> lấy tất cả các sản phẩm trong hệ thống.</li><li>BE trả về tất cả các sản phẩm.</li><li>FE hiển thị tất cả sản phẩm lên trang chủ.</li></ol>
Alternative flow (Thất bại)	1. Nếu xảy ra các trường hợp khác hệ thống sẽ thông báo lỗi.
<b>Extension point</b>	Không có

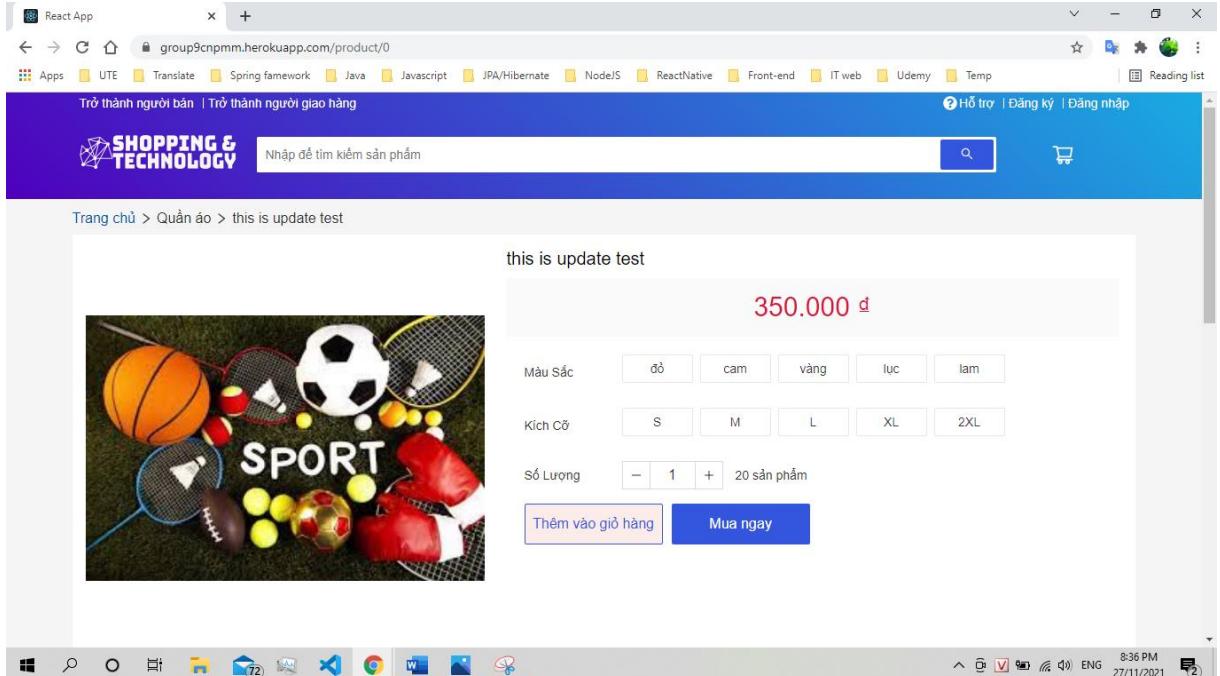


Hình 3. 1: Xem sản phẩm

## 1.2. Xem chi tiết sản phẩm

Name	Xem thông tin sản phẩm
<b>Brief description</b>	Khách hàng truy cập vào một sản phẩm hệ thống sẽ hiển thị thông tin chi tiết của sản phẩm.
<b>Actor(s)</b>	Guest
<b>Pre-conditions</b>	
<b>Post-conditions</b>	Sản phẩm sẽ được hiển thị trên trang chi tiết sản phẩm.
<b>Flow of events</b>	
Basic flow (Thành công)	<p>Người dùng thực hiện các bước sau:</p> <ol style="list-style-type: none"> <li>Truy cập vào trang chủ.</li> <li>Hệ thống gọi đến API <a href="https://cnpmmbbe.herokuapp.com/product/category">https://cnpmmbbe.herokuapp.com/product/category</a> lấy tất cả các chi tiết sản phẩm trong hệ thống.</li> <li>BE trả về tất cả các sản phẩm.</li> <li>FE hiển thị tất cả sản phẩm lên trang chủ.</li> </ol>
Alternative flow	<ol style="list-style-type: none"> <li>Nếu xảy ra các trường hợp khác hệ thống sẽ thông báo lỗi.</li> </ol>

(Thất bại)	
<b>Extension point</b>	Không có

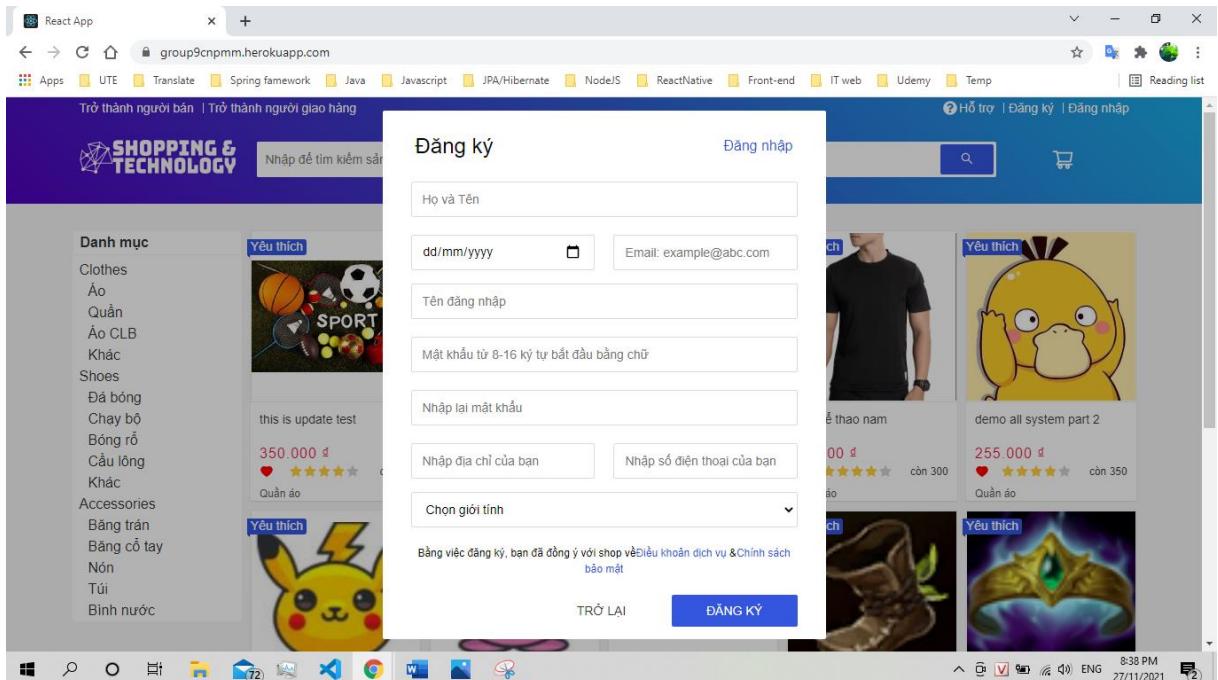


Hình 3. 2: xem chi tiết sản phẩm

### 1.3. Đăng ký

Name	Đăng ký
<b>Brief description</b>	Khi một khách hàng chưa có tài khoản trong hệ thống có thể chọn mục đăng ký.
<b>Actor(s)</b>	Guest
<b>Pre-conditions</b>	Chưa có tài khoản
<b>Post-conditions</b>	Có tài khoản trong hệ thống

Flow of events	
Basic flow (Thành công)	<p>Người dùng chưa có tài khoản thực hiện các bước sau:</p> <ol style="list-style-type: none"> <li>Ấn vào nút đăng ký.</li> <li>Điền tất cả thông tin vào form đăng ký.</li> <li>Ấn đăng ký.</li> <li>Hệ thống gọi đến API <a href="https://cnpmmbbe.herokuapp.com/signup">https://cnpmmbbe.herokuapp.com/signup</a> để tạo ra một user mới.</li> <li>Hệ thống chuyển về trang chủ kèm cùng với tài khoản đã đăng ký.</li> </ol>
Alternative flow (Thất bại)	<ol style="list-style-type: none"> <li>Nếu xảy ra các trường hợp khác hệ thống sẽ thông báo lỗi.</li> </ol>
Extension point	Không có

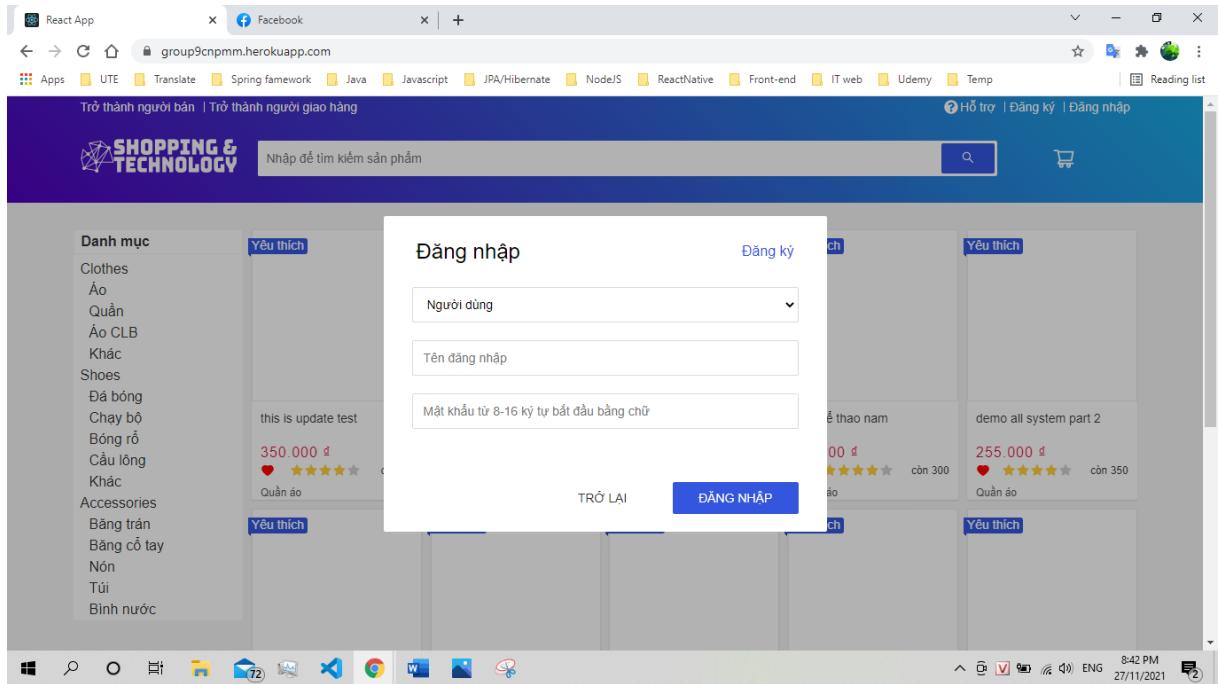


Hình 3. 3: Đăng ký

## 2. Tính năng khách hàng

### 1. Đăng nhập

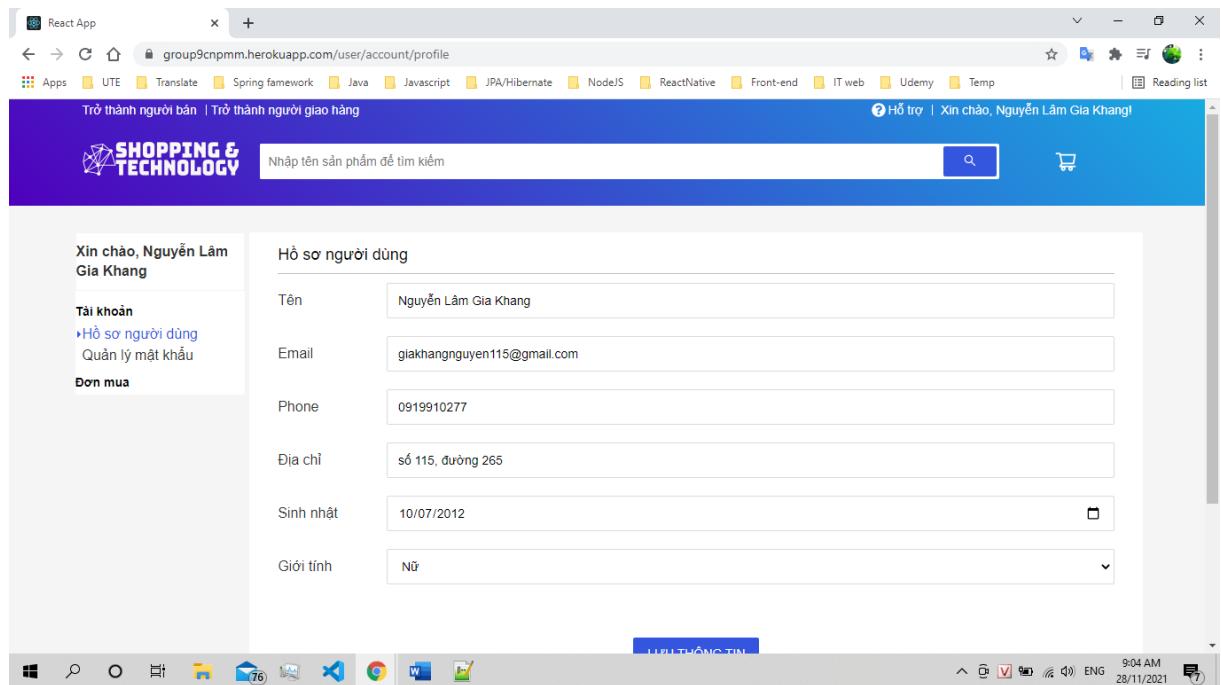
<b>Name</b>	<b>Đăng ký</b>
<b>Brief description</b>	Đăng nhập vào tài khoản khách hàng.
<b>Actor(s)</b>	User
<b>Pre-conditions</b>	Đã có tài khoản
<b>Post-conditions</b>	Đăng nhập thành công và có thể thực hiện các thao tác tương ứng.
<b>Flow of events</b>	
Basic flow (Thành công)	<p>Người dùng thực hiện các bước sau:</p> <ol style="list-style-type: none"> <li>1. Nhấn vào nút đăng nhập.</li> <li>2. Điền tất cả thông tin vào form đăng nhập.</li> <li>3. Nhấn đăng nhập.</li> <li>4. Hệ thống gọi đến API  <a href="https://cnpmmbbe.herokuapp.com/login">https://cnpmmbbe.herokuapp.com/login</a>.</li> <li>5. Hệ thống chuyển về trang chủ kèm cùng với tài khoản đã đăng nhập.</li> </ol>
Alternative flow (Thất bại)	<ol style="list-style-type: none"> <li>1. Nếu xảy ra các trường hợp khác hệ thống sẽ thông báo lỗi.</li> </ol>
<b>Extension point</b>	Không có



Hình 3. 4: Đăng nhập

## 2. Xem thông tin tài khoản

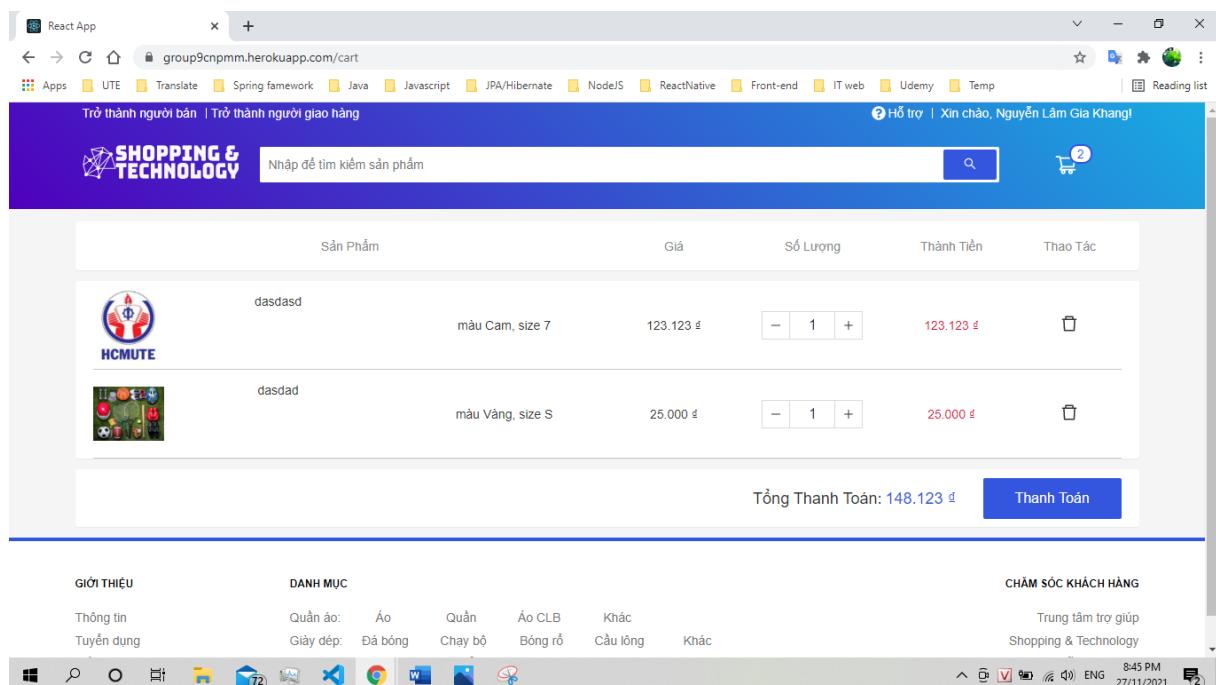
Name	Xem thông tin tài khoản
<b>Brief description</b>	Xem thông tin tài khoản khách hàng.
<b>Actor(s)</b>	User
<b>Pre-conditions</b>	Đã có tài khoản
<b>Post-conditions</b>	Hiển thị thông tin tài khoản
<b>Flow of events</b>	
Basic flow (Thành công)	<p>Người dùng thực hiện các bước sau:</p> <ol style="list-style-type: none"> <li>Ấn vào tài khoản của tôi.</li> <li>Hệ thống gọi đến API <a href="https://cnpmmbbe.herokuapp.com/user/:id">https://cnpmmbbe.herokuapp.com/user/:id</a>.</li> <li>Hệ thống hiển thị thông tin tài khoản.</li> </ol>
Alternative flow (Thất bại)	<ol style="list-style-type: none"> <li>Nếu xảy ra các trường hợp khác hệ thống sẽ thông báo lỗi.</li> </ol>
<b>Extension point</b>	Không có



Hình 3. 5: xem thông tin tài khoản

### 3. Thêm vào giỏ hàng

Name	Thêm vào giỎ hàng
Brief description	Thêm sản phẩm vào giỎ hàng.
Actor(s)	User
Pre-conditions	Đã có tài khoản
Post-conditions	Sản phẩm thêm vào giỎ hàng thành công.
Flow of events	
Basic flow (Thành công)	<p>Người dùng thực hiện các bước sau:</p> <ol style="list-style-type: none"> <li>1. Ăn vào sản phẩm muốn thêm.</li> <li>2. Chọn thêm vào giỎ hàng.</li> <li>3. Hệ thống tự động thêm sản phẩm vào giỎ hàng.</li> <li>4. Hiển thị thông báo thêm vào giỎ hàng thành công.</li> </ol>
Alternative flow (Thất bại)	<ol style="list-style-type: none"> <li>1. Nếu xảy ra các trường hợp khác hệ thống sẽ thông báo lỗi.</li> </ol>
Extension point	Không có

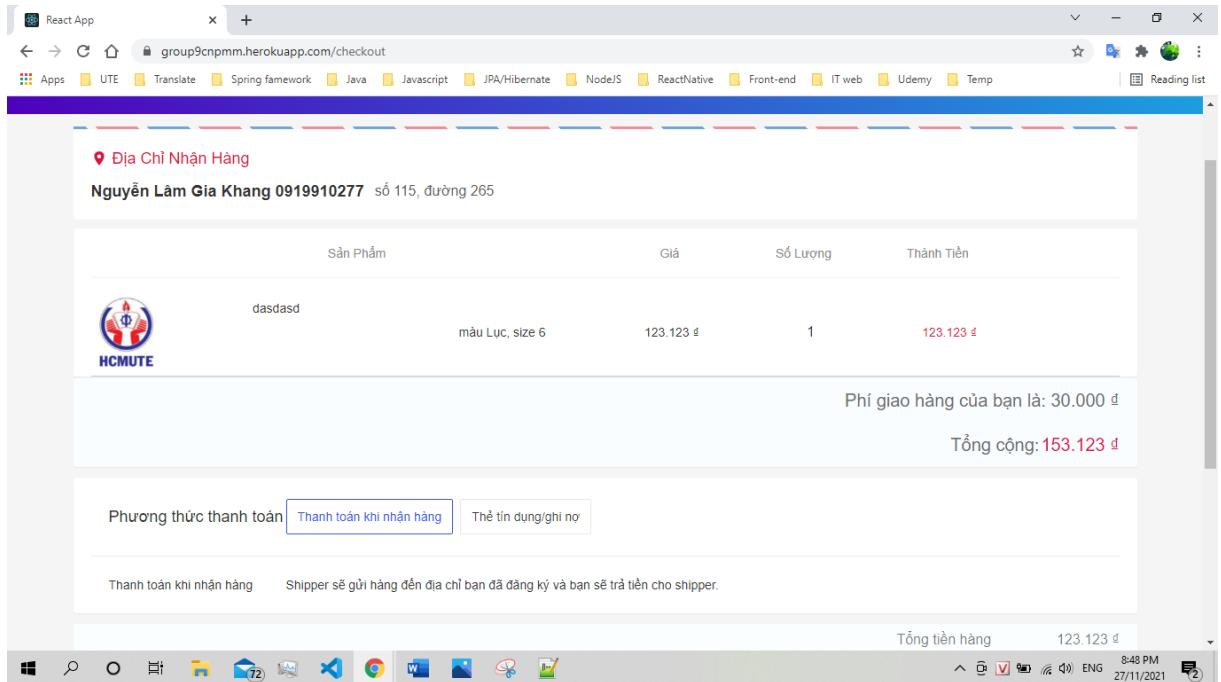


Hình 3. 6: thêm vào giỎ hàng



#### 4. Đặt hàng

<b>Name</b>	Đặt hàng
<b>Brief description</b>	Thao tác đặt hàng.
<b>Actor(s)</b>	User
<b>Pre-conditions</b>	Đã có tài khoản
<b>Post-conditions</b>	Đặt hàng thành công.
<b>Flow of events</b>	
Basic flow (Thành công)	<p>Người dùng thực hiện các bước sau:</p> <ol style="list-style-type: none"> <li>1. Án vào giỏ hàng.</li> <li>2. Chọn thanh toán.</li> <li>3. Kiểm tra thông tin sản phẩm và lựa chọn hình thức thanh toán.</li> <li>4. Chọn đặt hàng.</li> <li>5. Hệ thống gửi đến API  <a href="https://cnpmmbbe.herokuapp.com/user/order">https://cnpmmbbe.herokuapp.com/user/order</a>.         </li> <li>6. BE gửi thông báo thành công.</li> <li>7. Hệ thống chuyển về trang chủ.</li> </ol>
Alternative flow (Thất bại)	<ol style="list-style-type: none"> <li>1. Nếu xảy ra các trường hợp khác hệ thống sẽ thông báo lỗi.</li> </ol>
<b>Extension point</b>	Không có

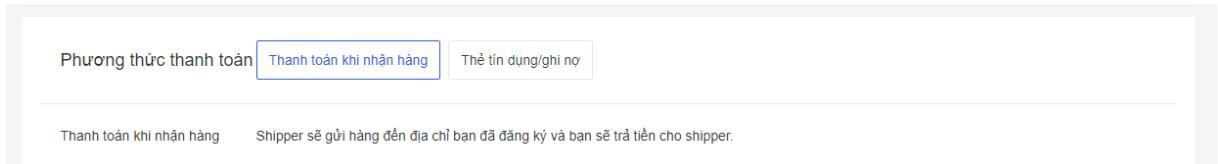


*Hình 3. 7: Đặt hàng*

## 5. Thanh toán bằng tiền mặt

Name	Thanh toán bằng tiền mặt
<b>Brief description</b>	Thanh toán bằng tiền mặt.
<b>Actor(s)</b>	User
<b>Pre-conditions</b>	Đã có tài khoản
<b>Post-conditions</b>	Đặt hàng thành công
<b>Flow of events</b>	
Basic flow (Thành công)	<p>Người dùng thực hiện các bước sau:</p> <ol style="list-style-type: none"> <li>1. Án vào giỏ hàng.</li> <li>2. Chọn thanh toán.</li> <li>3. Kiểm tra thông tin sản phẩm và lựa chọn hình thức thanh toán bằng tiền mặt.</li> <li>4. Chọn đặt hàng.</li> <li>5. Hệ thống gửi đến API</li> </ol> <p><a href="https://cnpmbe.herokuapp.com/user/order">https://cnpmbe.herokuapp.com/user/order</a>.</p>

	<p>6. BE gửi thông báo thành công.</p> <p>7. Hệ thống chuyển về trang chủ.</p>
Alternative flow (Thất bại)	<p>1. Nếu xảy ra các trường hợp khác hệ thống sẽ thông báo lỗi.</p>
<b>Extension point</b>	Không có



Hình 3. 8: Thanh toán bằng tiền mặt

## 6. Thanh toán bằng paypal

<b>Name</b>	<b>Thanh toán bằng paypal</b>
<b>Brief description</b>	Thanh toán bằng paypal.
<b>Actor(s)</b>	User
<b>Pre-conditions</b>	Đã có tài khoản
<b>Post-conditions</b>	Đặt hàng thành công
<b>Flow of events</b>	
Basic flow (Thành công)	<p>Người dùng thực hiện các bước sau:</p> <ol style="list-style-type: none"> <li>1. Án vào giỏ hàng.</li> <li>2. Chọn thanh toán.</li> <li>3. Kiểm tra thông tin sản phẩm và lựa chọn hình thức thanh toán bằng paypal.</li> <li>4. Chọn đặt hàng.</li> <li>5. Hệ thống gửi đến API  <a href="https://cnpmmbbe.herokuapp.com/user/order">https://cnpmmbbe.herokuapp.com/user/order</a>.         </li> <li>6. BE gửi thông báo thành công.</li> <li>7. Hệ thống chuyển về trang chủ.</li> </ol>

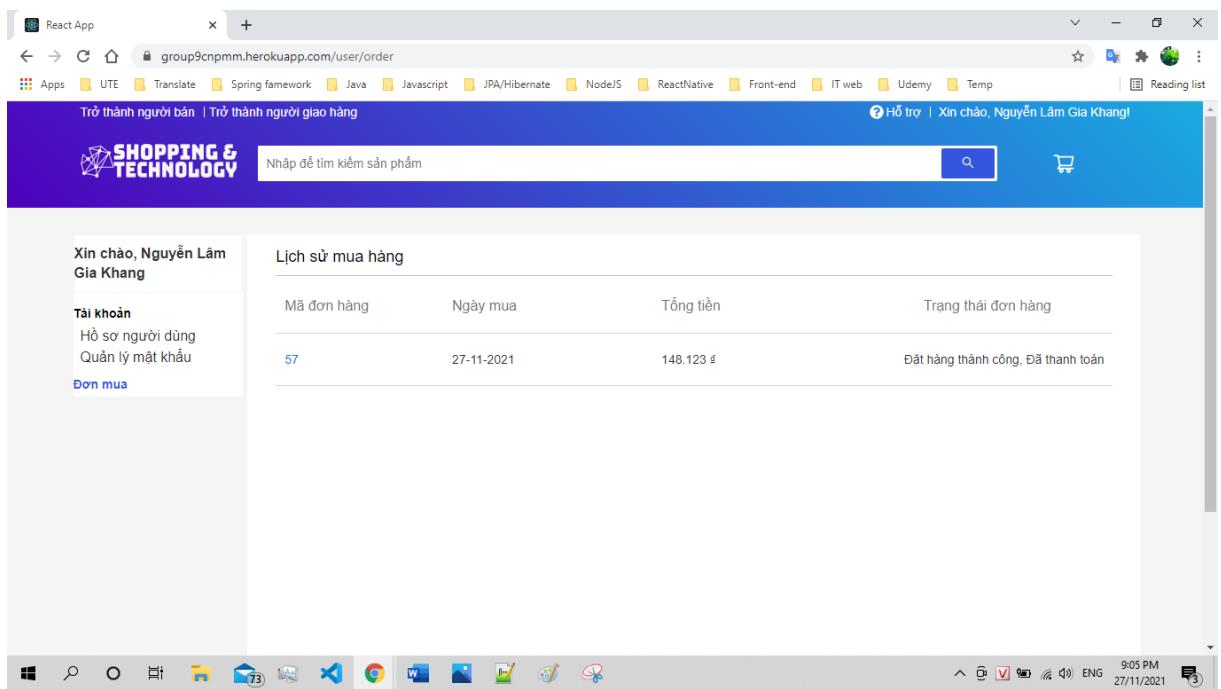
Alternative flow (Thất bại)	1. Nếu xảy ra các trường hợp khác hệ thống sẽ thông báo lỗi.
<b>Extension point</b>	Không có



Hình 3. 9: Thanh toán bằng paypal

## 7. Xem lịch sử đặt hàng

Name	Xem lịch sử đặt hàng
<b>Brief description</b>	Khách hàng có thể xem lịch sử đặt hàng.
<b>Actor(s)</b>	User
<b>Pre-conditions</b>	Đã có tài khoản
<b>Post-conditions</b>	Hệ thống hiển thị lịch sử đặt hàng.
<b>Flow of events</b>	
Basic flow (Thành công)	<p>Người dùng thực hiện các bước sau:</p> <ol style="list-style-type: none"> <li>1. Ấn vào đơn hàng.</li> <li>2. Hệ thống gửi đến API <a href="https://cnpmmbbe.herokuapp.com/user/orderhistory/:id">https://cnpmmbbe.herokuapp.com/user/orderhistory/:id</a>.</li> <li>3. BE gửi danh sách đơn hàng.</li> <li>4. Hệ thống hiển thị lên trang xem lịch sử.</li> </ol>
Alternative flow (Thất bại)	1. Nếu xảy ra các trường hợp khác hệ thống sẽ thông báo lỗi.
<b>Extension point</b>	Không có

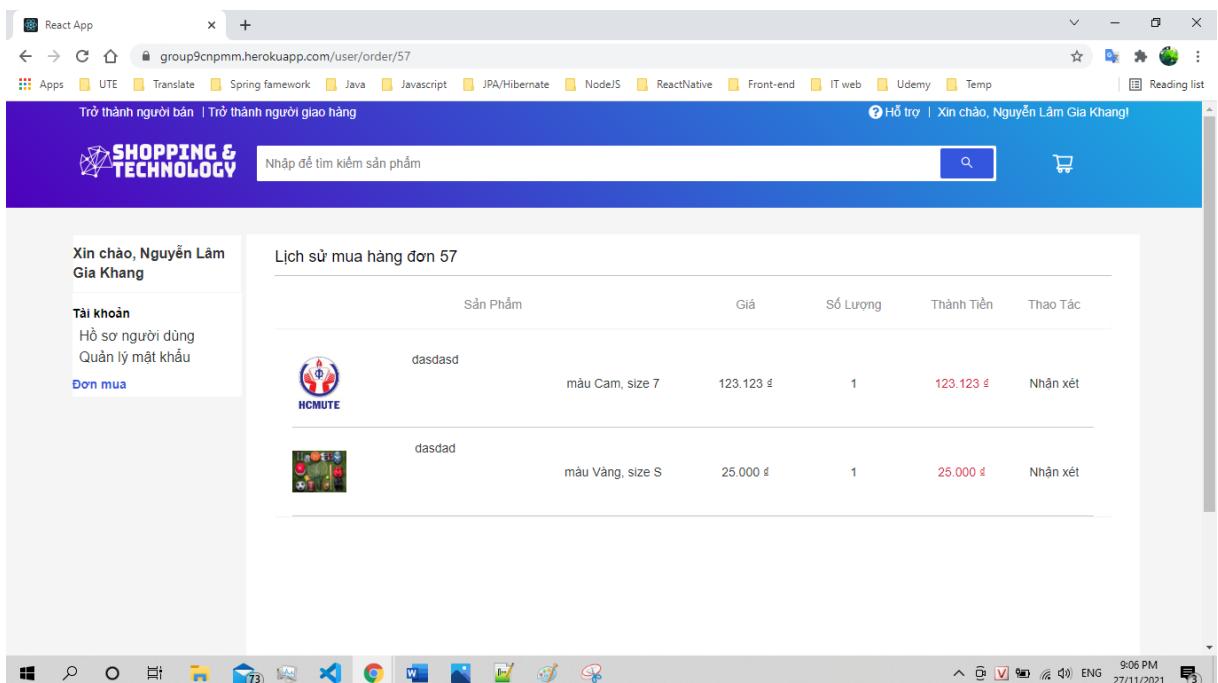


Hình 3. 10: xem lịch sử đặt hàng

## 8. Xem chi tiết đơn hàng

Name	Xem lịch sử chi tiết đơn hàng
<b>Brief description</b>	Khách hàng có thể xem lịch sử chi tiết đơn hàng.
<b>Actor(s)</b>	User
<b>Pre-conditions</b>	Đã có tài khoản
<b>Post-conditions</b>	Hệ thống hiển thị lịch sử chi tiết đơn hàng.
<b>Flow of events</b>	
Basic flow (Thành công)	<p>Người dùng thực hiện các bước sau:</p> <ol style="list-style-type: none"> <li>1. Án vào đơn hàng.</li> <li>2. Án vào một đơn hàng bất kỳ.</li> <li>3. Hệ thống gửi đến API</li> </ol> <p><a href="https://cnppmbe.herokuapp.com/user/orderdetailhistory/:id">https://cnppmbe.herokuapp.com/user/orderdetailhistory/:id</a></p>

	<p>4. BE gửi danh sách chi tiết sản phẩm trong đơn hàng.</p> <p>5. Hệ thống hiển thị lên trang xem lịch sử.</p>
Alternative flow (Thất bại)	<p>1. Nếu xảy ra các trường hợp khác hệ thống sẽ thông báo lỗi.</p>
<b>Extension point</b>	Không có

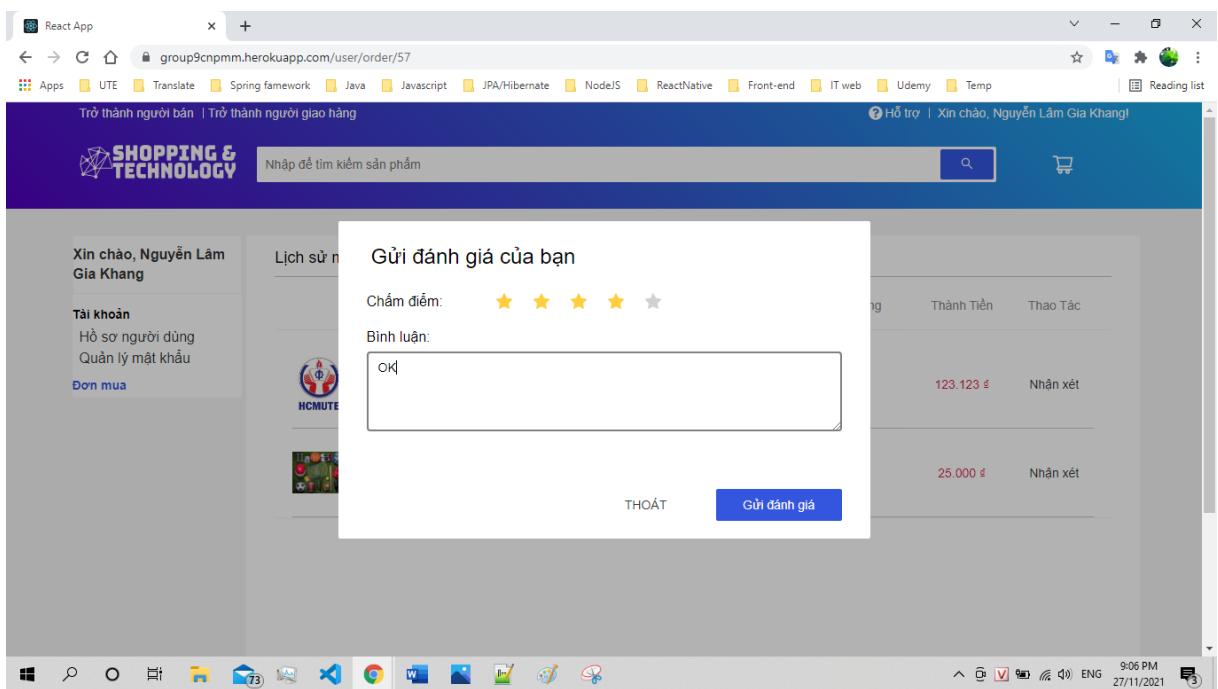


Hình 3. 11: xem chi tiết đơn hàng

## 9. Bình luận

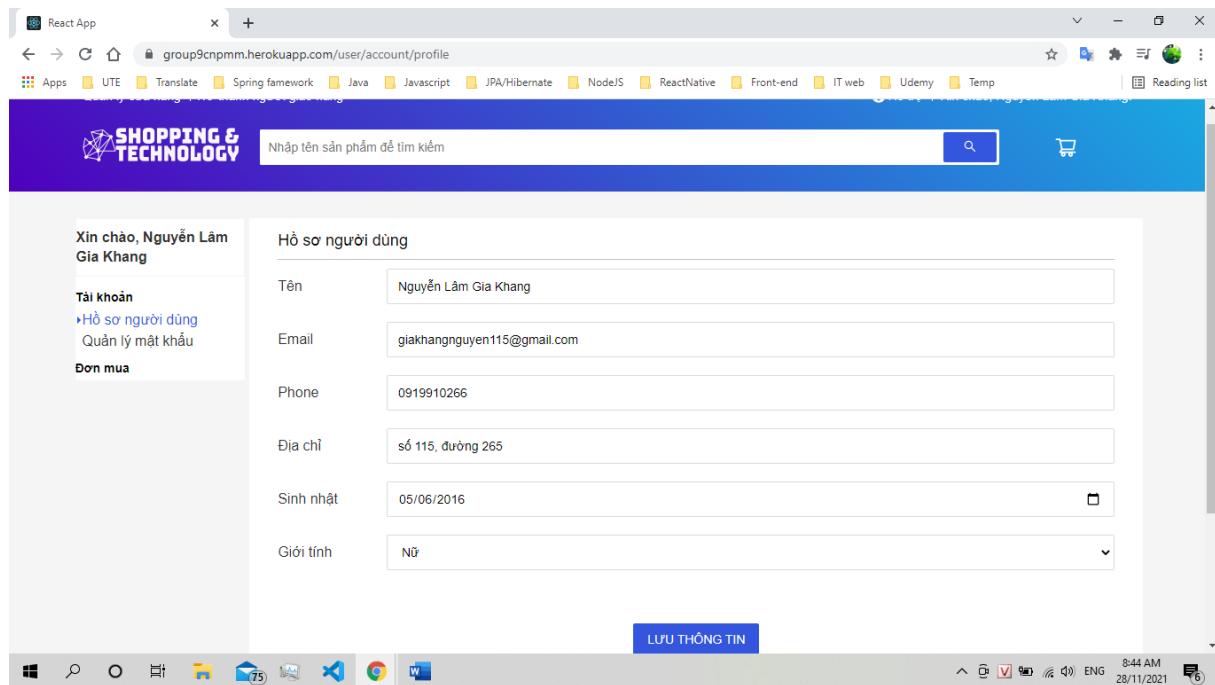
Name	Bình luận
<b>Brief description</b>	Khách hàng có thể bình luận sản phẩm.
<b>Actor(s)</b>	User
<b>Pre-conditions</b>	Đã có tài khoản, đã mua sản phẩm.
<b>Post-conditions</b>	Bình luận được thêm vào sản phẩm.
<b>Flow of events</b>	

Basic flow (Thành công)	<p>Người dùng thực hiện các bước sau:</p> <ol style="list-style-type: none"> <li>1. Án vào đơn hàng.</li> <li>2. Án vào một đơn hàng bất kỳ.</li> <li>3. Án vào nhận xét</li> <li>4. Hệ thống gửi đến API <a href="https://cnpmmbe.herokuapp.com/user/comment">https://cnpmmbe.herokuapp.com/user/comment</a>.</li> <li>5. BE lưu bình luận vào sản phẩm.</li> <li>6. Hệ thống hiển thị thông báo bình luận thành công.</li> </ol>
Alternative flow (Thất bại)	<ol style="list-style-type: none"> <li>1. Nếu xảy ra các trường hợp khác hệ thống sẽ thông báo lỗi.</li> </ol>
Extension point	Không có



Hình 3. 12: Bình luận

## 10. Chỉnh sửa thông tin tài khoản



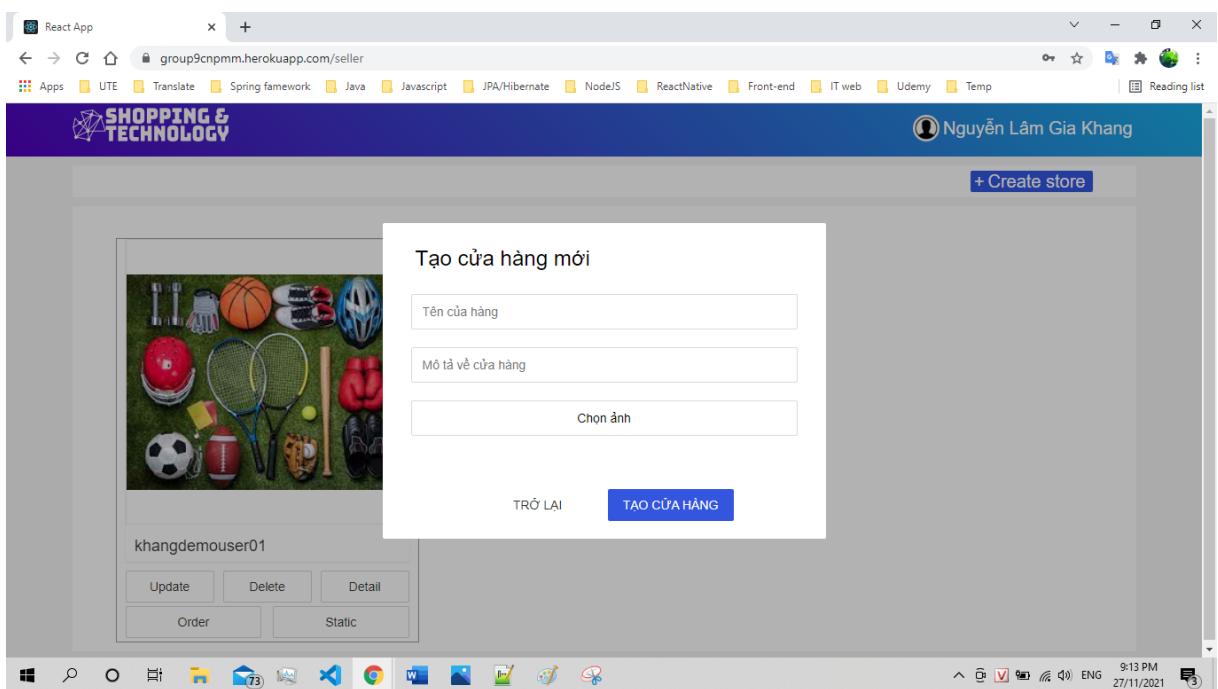
Hình 3. 13: *Chỉnh sửa thông tin khách hàng*

### 3. Tính năng người bán hàng

#### 1. Tạo cửa hàng

Name	Tạo ra một cửa hàng mới
<b>Brief description</b>	Người bán hàng có thể tạo ra một cửa hàng mới.
<b>Actor(s)</b>	Seller
<b>Pre-conditions</b>	Đăng ký tài khoản bán hàng.
<b>Post-conditions</b>	Tạo thành công một cửa hàng.
<b>Flow of events</b>	
Basic flow (Thành công)	<p>Người dùng thực hiện các bước sau:</p> <ol style="list-style-type: none"> <li>1. Truy cập vào trang chủ.</li> <li>2. Đăng nhập bằng tài khoản người bán hàng.</li> <li>3. Truy cập vào trang chủ của người bán hàng.</li> <li>4. Án tạo cửa hàng.</li> <li>5. Nhập thông tin cần thiết.</li> <li>6. Án nút tạo cửa hàng.</li> </ol>

	<p>7. Hệ thống gọi đến API tạo cửa hàng mới.</p> <p>8. BE tạo ra một cửa hàng và lưu trữ vào trong database.</p> <p>9. BE trả về thông tin cửa hàng.</p> <p>10. Hệ thống gọi đến API lấy tất cả các cửa hàng thuộc sở hữu của chủ cửa hàng đó.</p> <p>11. BE lấy tất cả các cửa hàng thuộc sở hữu của chủ cửa hàng đó.</p> <p>12. BE trả về các cửa hàng.</p> <p>13. FE hiển thị cửa hàng lên trang quản lý.</p>
Alternative flow (Thất bại)	<p>1. Nếu xảy ra các trường hợp khác hệ thống sẽ thông báo lỗi.</p>
<b>Extension point</b>	Không có

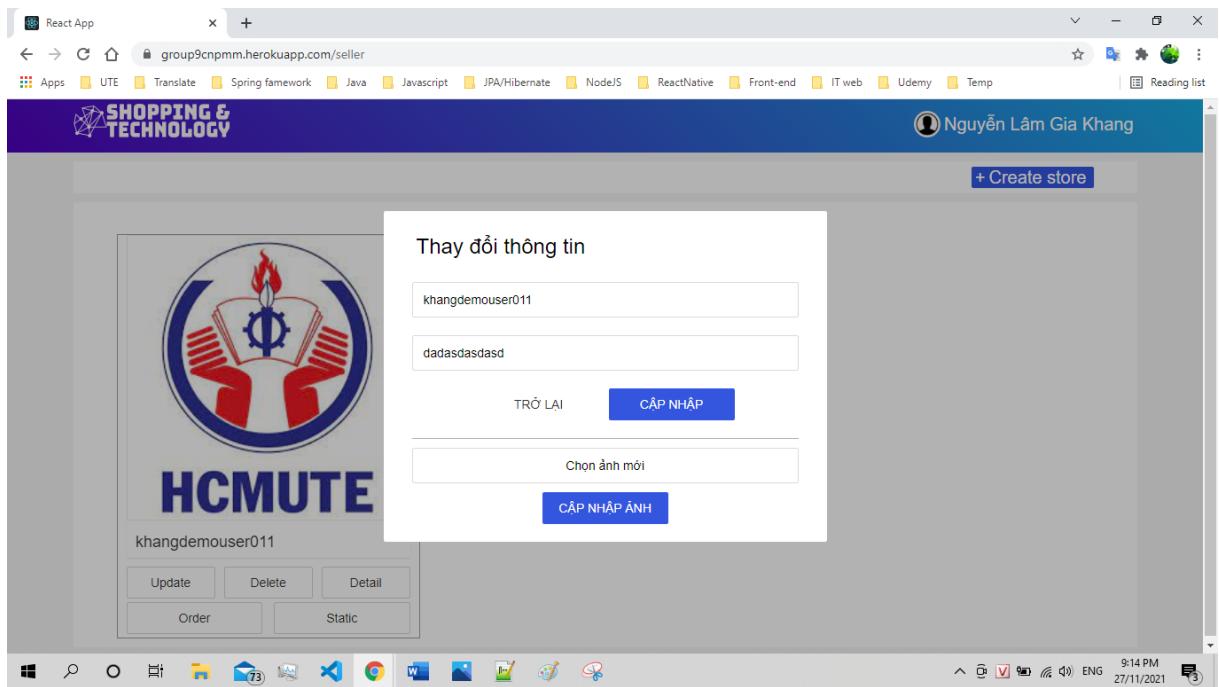


Hình 3. 14: Tạo cửa hàng

## 2. Chính sửa cửa hàng

Name	Chỉnh sửa thông một cửa hàng mới
------	----------------------------------

<b>Brief description</b>	Người bán hàng có thể tạo ra một cửa hàng mới.
<b>Actor(s)</b>	Seller
<b>Pre-conditions</b>	Đăng ký tài khoản bán hàng. Cửa hàng phải tồn tại.
<b>Post-conditions</b>	Chỉnh sửa thành công thông tin một cửa hàng.
<b>Flow of events</b>	
Basic flow (Thành công)	<p>Người dùng thực hiện các bước sau:</p> <ol style="list-style-type: none"> <li>1. Người bán hàng ấn vào nút chỉnh sửa của cửa hàng.</li> <li>2. FE hiển thị thông tin của cửa hàng.</li> <li>3. Người bán hàng nhập các thông tin cần chỉnh sửa và ấn chỉnh sửa.</li> <li>4. Hệ thống gọi đến API chỉnh sửa thông tin cửa hàng.</li> <li>5. BE cập nhật thông tin vào trong cơ sở dữ liệu.</li> <li>6. BE trả về thông báo cập nhật thành công.</li> <li>7. FE hiển thị thông báo cập nhật thành công và chuyển về trang quản lý.</li> </ol>
Alternative flow (Thất bại)	<ol style="list-style-type: none"> <li>1. Nếu xảy ra các trường hợp khác hệ thống sẽ thông báo lỗi.</li> </ol>
<b>Extension point</b>	Không có

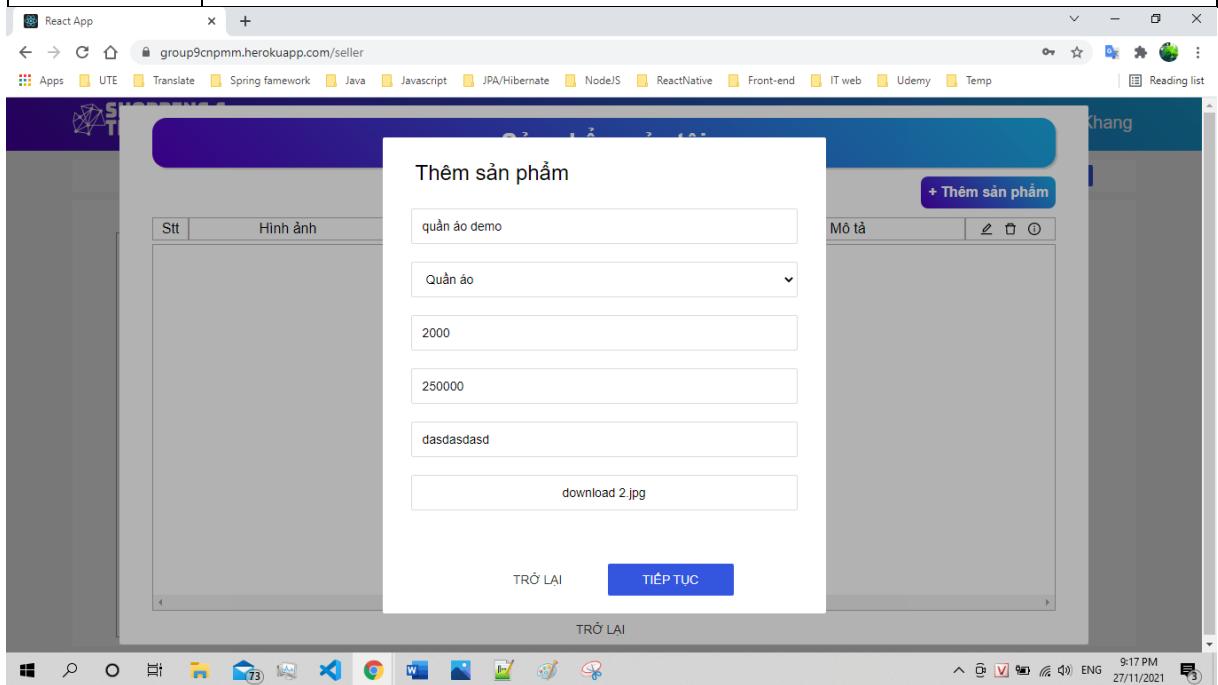


Hình 3. 15: *Chỉnh sửa cửa hàng*

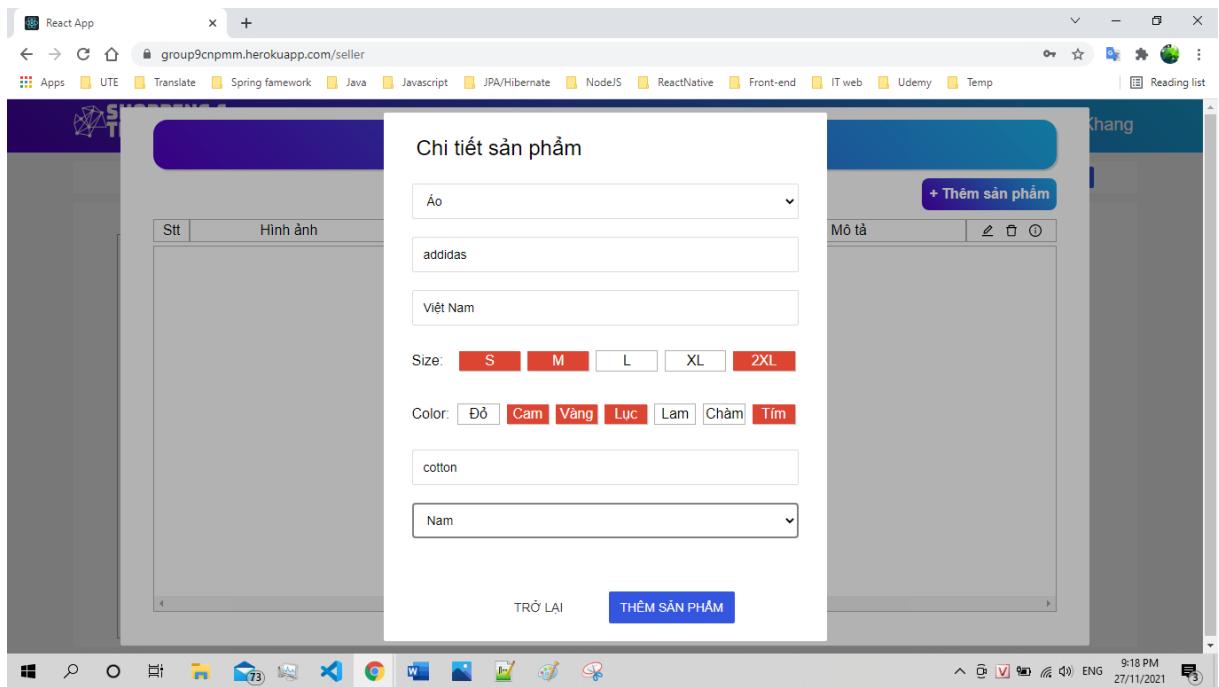
### 3. Thêm sản phẩm quần áo

Name	Thêm sản phẩm quần áo
<b>Brief description</b>	Thêm sản phẩm loại quần áo.
<b>Actor(s)</b>	Seller
<b>Pre-conditions</b>	Đăng ký tài khoản bán hàng. Cửa hàng phải tồn tại.
<b>Post-conditions</b>	Thêm sản phẩm quần áo thành công.
<b>Flow of events</b>	
Basic flow (Thành công)	<p>Người dùng thực hiện các bước sau:</p> <ol style="list-style-type: none"> <li>1. Án vào chi tiết của một cửa hàng.</li> <li>2. Chọn thêm sản phẩm.</li> <li>3. Nhập thông tin chung.</li> </ol>

	<p>4. Án tạo sản phẩm.</p> <p>5. Hệ thống gọi đến API</p> <p><a href="https://cnpmmbe.herokuapp.com/seller/product">https://cnpmmbe.herokuapp.com/seller/product</a>.</p> <p>6. Nhập thông tin chi tiết của loại quần áo.</p> <p>7. Hệ thống gọi đến API</p> <p><a href="https://cnpmmbe.herokuapp.com/seller/product/categoryclothes">https://cnpmmbe.herokuapp.com/seller/product/categoryclothes</a></p> <p>.</p> <p>8. Hệ thống thông báo thêm sản phẩm thành công.</p>
Alternative flow (Thất bại)	2. Nếu xảy ra các trường hợp khác hệ thống sẽ thông báo lỗi.
<b>Extension point</b>	Không có



Hình 3. 16: thêm sản phẩm quần áo

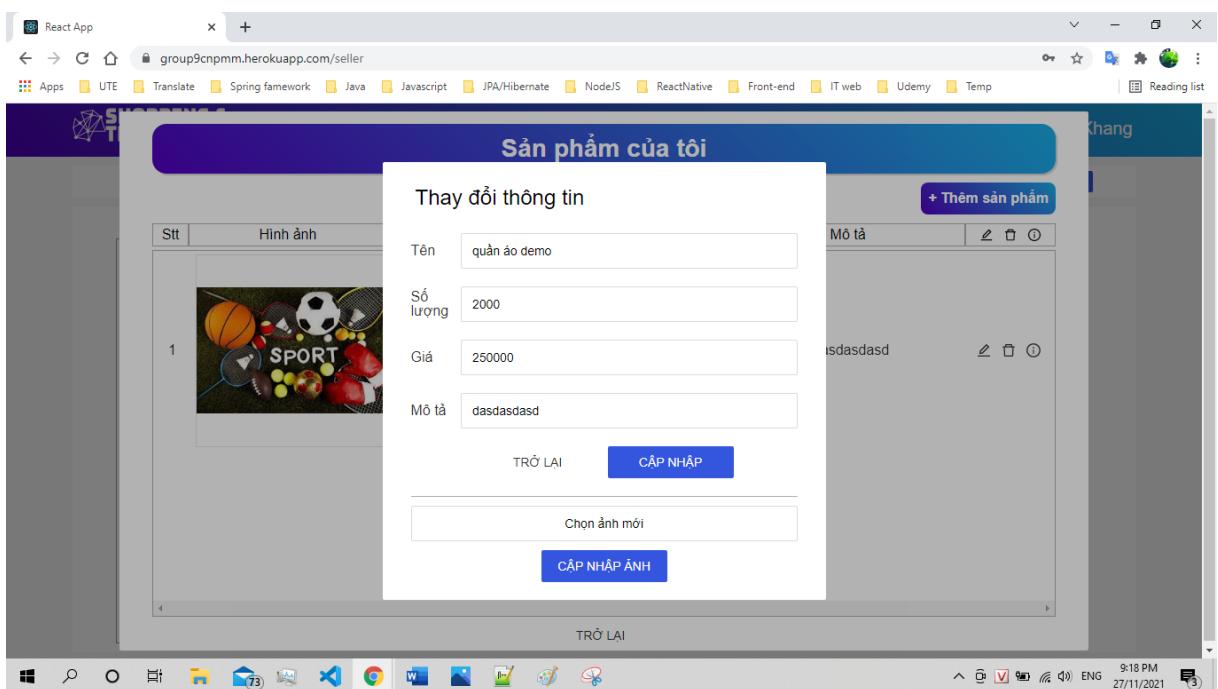


Hình 3. 17: thêm sản phẩm quần áo

#### 4. Chính sửa thông tin sản phẩm

Name	Chỉnh sửa một sản phẩm
<b>Brief description</b>	Người bán hàng có thể chỉnh sửa một sản phẩm.
<b>Actor(s)</b>	Seller
<b>Pre-conditions</b>	Đăng ký tài khoản bán hàng. Có ít nhất một cửa hàng. Có ít nhất một sản phẩm.
<b>Post-conditions</b>	Chỉnh sửa thành công một sản phẩm.
<b>Flow of events</b>	
Basic flow (Thành công)	Người dùng thực hiện các bước sau: 1. Đăng nhập bằng tài khoản người bán hàng. 2. Truy cập vào trang quản lý cửa hàng. 3. Bấm vào nút chi tiết của cửa hàng cần tạo sản phẩm. 4. Nhấn vào nút thông tin một sản phẩm. 5. FE hiển thị thông tin sản phẩm.

	<p>6. Người bán hàng ấn nút chỉnh sửa.</p> <p>7. Chỉnh sửa các thông tin cần thiết và ấn nút lưu.</p> <p>8. Hệ thống gọi đến API chỉnh sửa thông tin sản phẩm.</p> <p>9. BE lưu thông tin sản phẩm chỉnh sửa vào cơ sở dữ liệu.</p> <p>10. BE trả về thông tin đã chỉnh sửa.</p> <p>11. FE hiển thị thông tin cho người bán hàng.</p>
Alternative flow (Thất bại)	<p>1. Nếu xảy ra các trường hợp khác hệ thống sẽ thông báo lỗi.</p>
<b>Extension point</b>	Không có

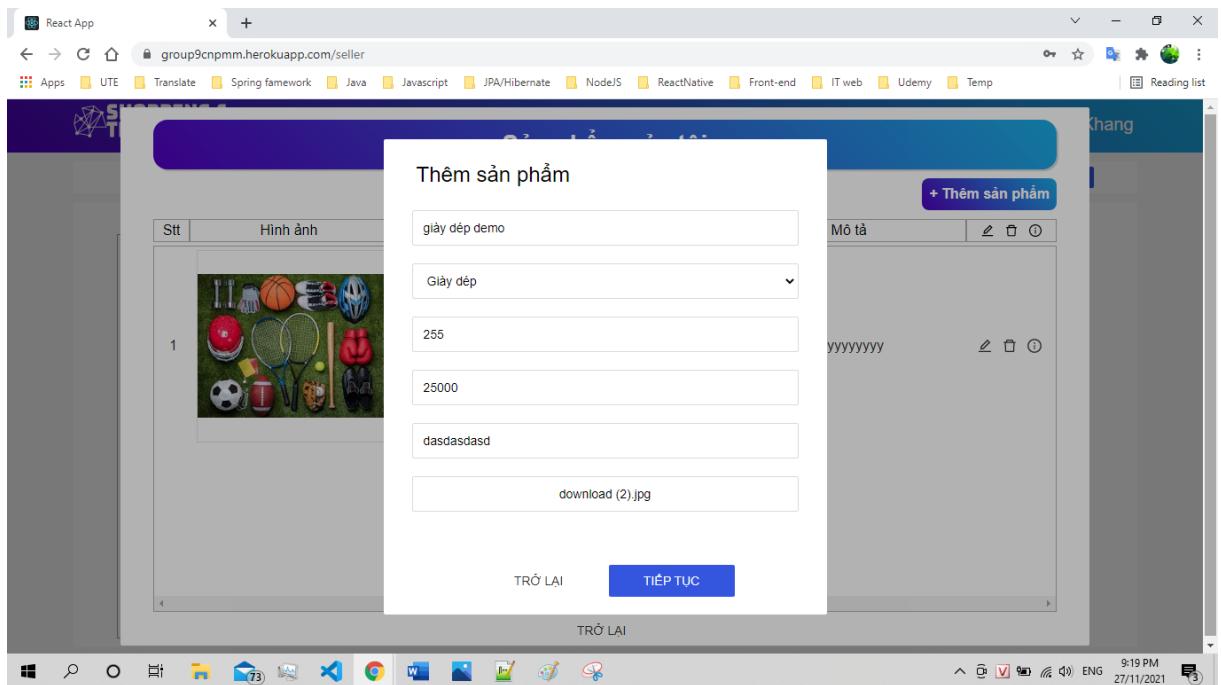


Hình 3. 18: chỉnh sửa thông tin sản phẩm

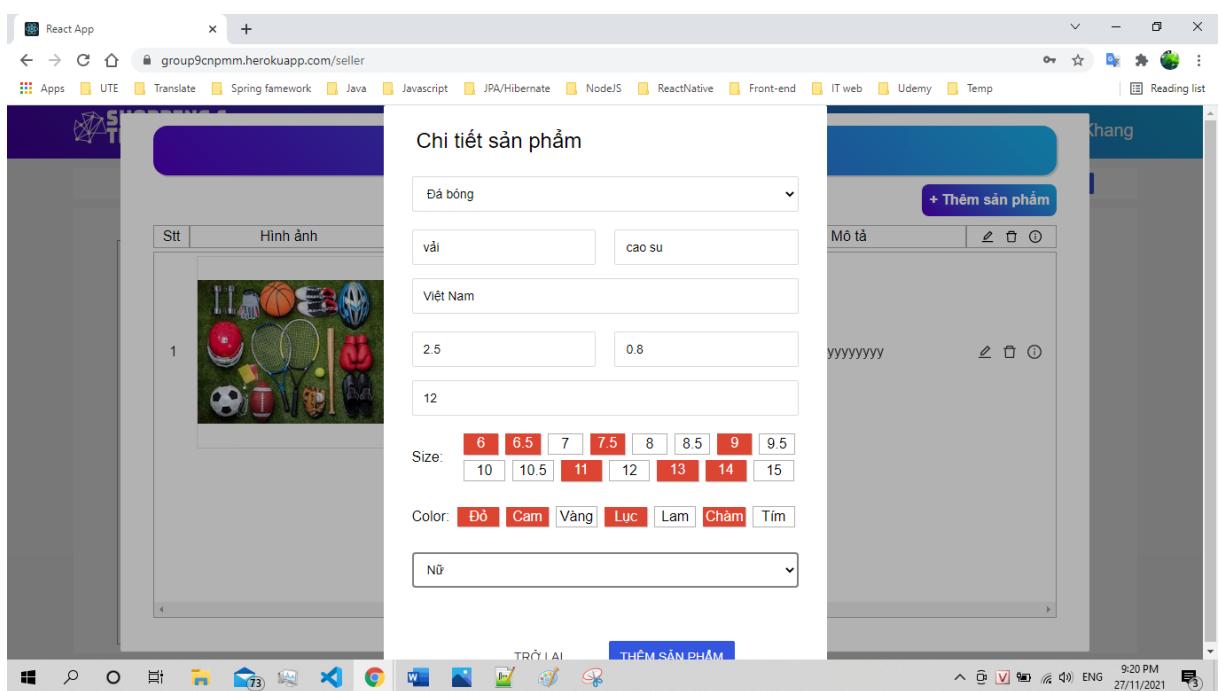
## 5. Thêm sản phẩm giày dép

Name	<b>Thêm sản phẩm giày dép</b>
Brief description	Thêm sản phẩm loại giày dép.
Actor(s)	Seller

<b>Pre-conditions</b>	Đăng ký tài khoản bán hàng. Cửa hàng phải tồn tại.
<b>Post-conditions</b>	Thêm sản phẩm giày dép thành công.
<b>Flow of events</b>	
Basic flow (Thành công)	<p>Người dùng thực hiện các bước sau:</p> <ol style="list-style-type: none"> <li>1. Án vào chi tiết của một cửa hàng.</li> <li>2. Chọn thêm sản phẩm.</li> <li>3. Nhập thông tin chung.</li> <li>4. Án tạo sản phẩm.</li> <li>5. Hệ thống gọi đến API <a href="https://cnpmmbe.herokuapp.com/seller/product">https://cnpmmbe.herokuapp.com/seller/product</a>.</li> <li>6. Nhập thông tin chi tiết của loại giày dép.</li> <li>7. Hệ thống gọi đến API <a href="https://cnpmmbe.herokuapp.com/seller/product/categorieshoes">https://cnpmmbe.herokuapp.com/seller/product/categorieshoes</a>.</li> <li>8. Hệ thống thông báo thêm sản phẩm thành công.</li> </ol>
Alternative flow (Thất bại)	<ol style="list-style-type: none"> <li>1. Nếu xảy ra các trường hợp khác hệ thống sẽ thông báo lỗi.</li> </ol>
<b>Extension point</b>	Không có



Hình 3. 19: thêm sản phẩm giày dép

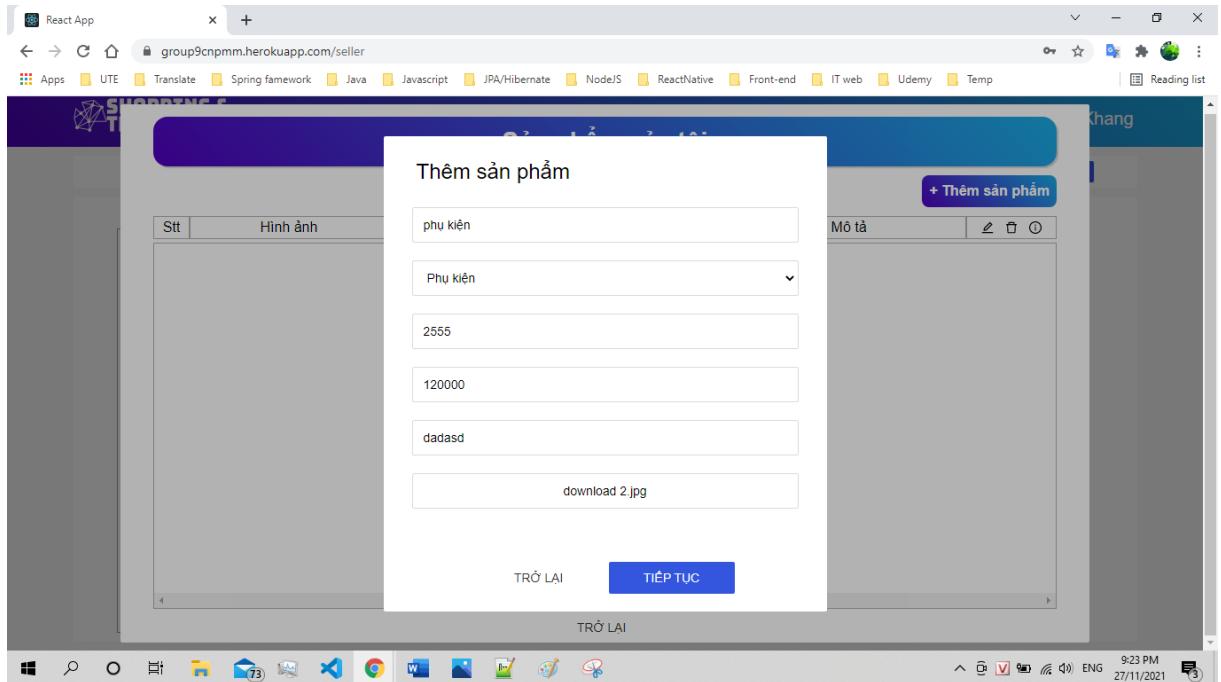


Hình 3. 20: thêm sản phẩm giày dép

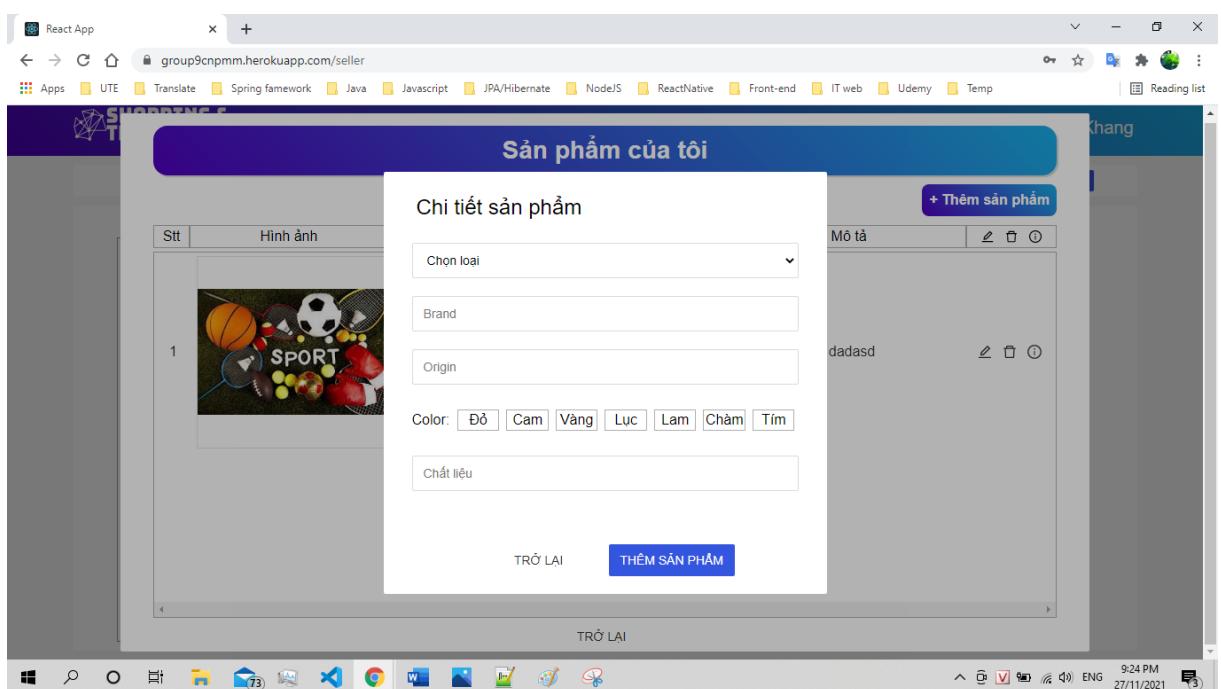
## 6. Thêm sản phẩm phụ kiện

Name	Thêm sản phẩm phụ kiện
------	------------------------

<b>Brief description</b>	Thêm sản phẩm loại phụ kiện.
<b>Actor(s)</b>	Seller
<b>Pre-conditions</b>	Đăng ký tài khoản bán hàng. Cửa hàng phải tồn tại.
<b>Post-conditions</b>	Thêm sản phẩm phụ kiện thành công.
<b>Flow of events</b>	
Basic flow (Thành công)	<p>Người dùng thực hiện các bước sau:</p> <ol style="list-style-type: none"> <li>1. Án vào chi tiết của một cửa hàng.</li> <li>2. Chọn thêm sản phẩm.</li> <li>3. Nhập thông tin chung.</li> <li>4. Án tạo sản phẩm.</li> <li>5. Hệ thống gọi đến API <a href="https://cnpmmbbe.herokuapp.com/seller/product">https://cnpmmbbe.herokuapp.com/seller/product</a>.</li> <li>6. Nhập thông tin chi tiết của loại phụ kiện.</li> <li>7. Hệ thống gọi đến API <a href="https://cnpmmbbe.herokuapp.com/seller/product/categoryaccessories">https://cnpmmbbe.herokuapp.com/seller/product/categoryaccessories</a>.</li> <li>8. Hệ thống thông báo thêm sản phẩm thành công.</li> </ol>
Alternative flow (Thất bại)	2. Nếu xảy ra các trường hợp khác hệ thống sẽ thông báo lỗi.
<b>Extension point</b>	Không có



Hình 3. 21: thêm sản phẩm phụ kiện

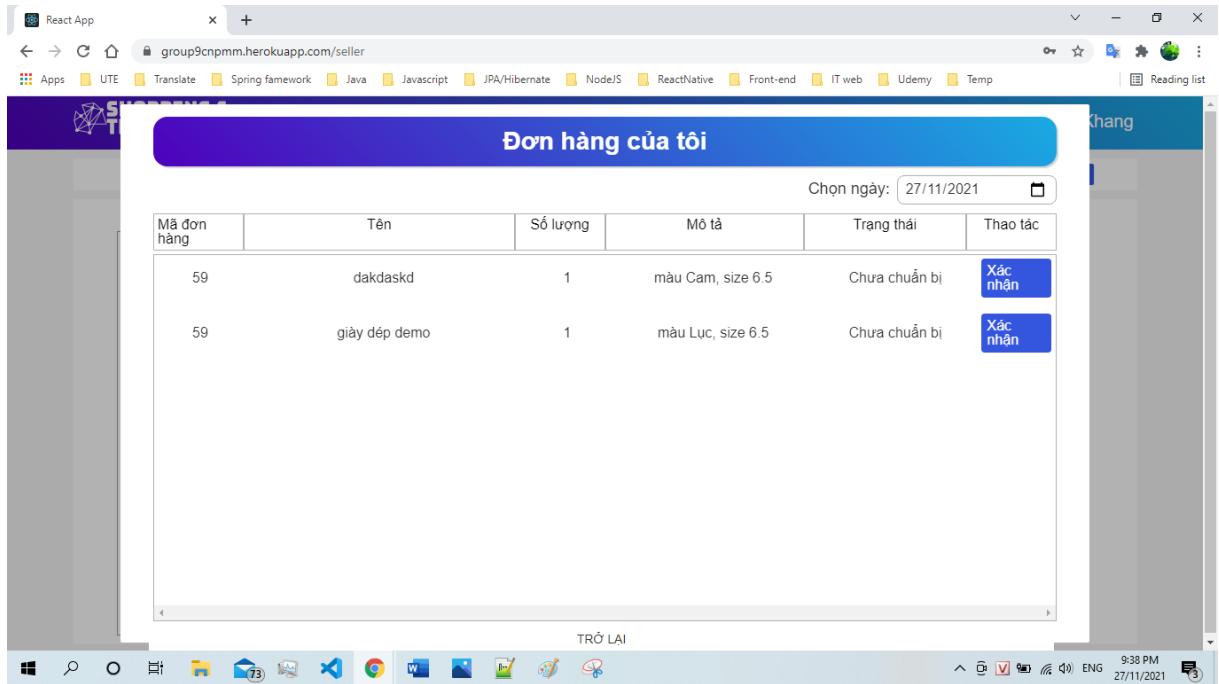


Hình 3. 22: Thêm sản phẩm phụ kiện

## 7. Xử lý đơn hàng

Name	Xử lý đơn hàng
------	----------------

<b>Brief description</b>	Xử lý đơn hàng.
<b>Actor(s)</b>	Seller
<b>Pre-conditions</b>	Đăng ký tài khoản bán hàng. Cửa hàng phải tồn tại. Đã có đơn hàng.
<b>Post-conditions</b>	Xử lý đơn hàng thành công.
<b>Flow of events</b>	
Basic flow (Thành công)	<p>Người dùng thực hiện các bước sau:</p> <ol style="list-style-type: none"> <li>1. Vào trang quản lý cửa hàng.</li> <li>2. Chọn order.</li> <li>3. Hệ thống gọi đến API <a href="https://cnpmmbbe.herokuapp.com/seller/order/4">https://cnpmmbbe.herokuapp.com/seller/order/4</a>.</li> <li>4. BE gửi lại thông tin đặt hàng.</li> <li>5. Seller xác nhận đơn hàng.</li> <li>6. Hệ thống gọi đến API <a href="https://cnpmmbbe.herokuapp.com/seller/orderdetail/status/2">https://cnpmmbbe.herokuapp.com/seller/orderdetail/status/2</a></li> <li>7. Hệ thống thông báo xác nhận thành công.</li> </ol>
Alternative flow (Thất bại)	<ol style="list-style-type: none"> <li>1. Nếu xảy ra các trường hợp khác hệ thống sẽ thông báo lỗi.</li> </ol>
<b>Extension point</b>	Không có

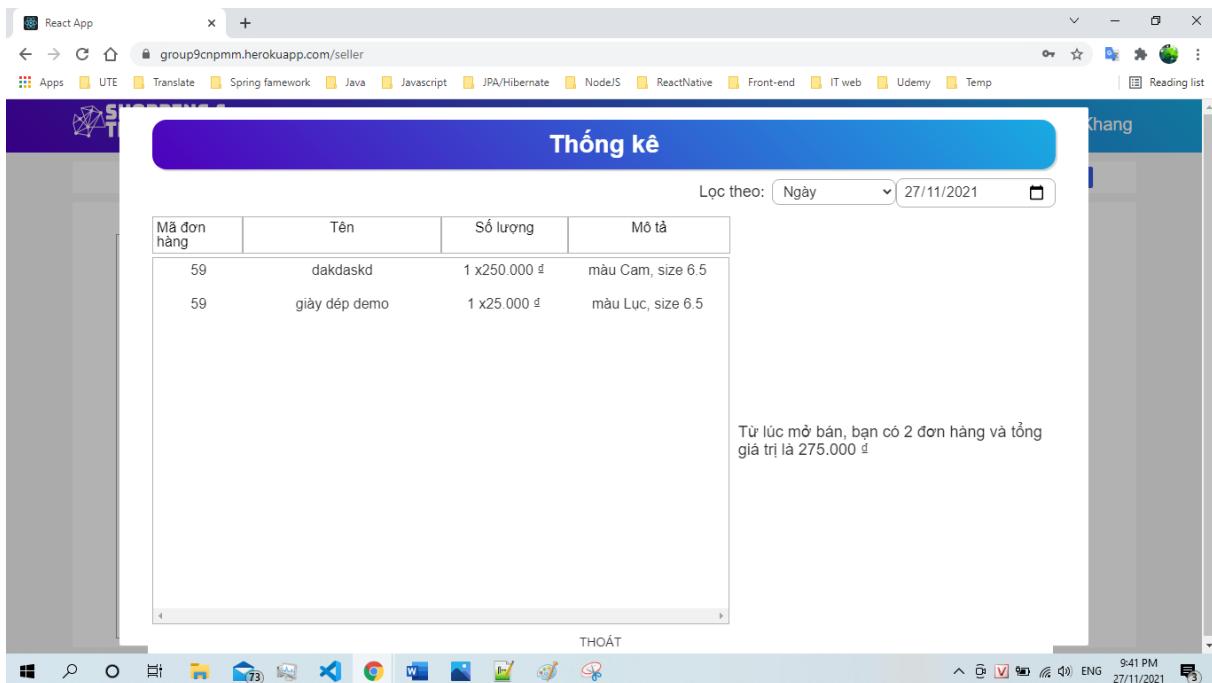


Hình 3. 23: xử lý đơn hàng

## 8. Thông kê

<b>Name</b>	Xử lý đơn hàng
<b>Brief description</b>	Xử lý đơn hàng.
<b>Actor(s)</b>	Seller
<b>Pre-conditions</b>	Đăng ký tài khoản bán hàng. Cửa hàng phải tồn tại.
<b>Post-conditions</b>	Hiển thị thống kê
<b>Flow of events</b>	
Basic flow (Thành công)	<p>Người dùng thực hiện các bước sau:</p> <ol style="list-style-type: none"> <li>1. Vào trang quản lý cửa hàng.</li> <li>2. Chọn static.</li> <li>3. Hệ thống gọi đến API</li> </ol> <p><a href="https://cnpmmbe.herokuapp.com/seller/order/2/statusfinished">https://cnpmmbe.herokuapp.com/seller/order/2/statusfinished</a></p>

	4. BE gửi lại thông tin thống kê. 5. Hệ thống hiển thị thống kê.
Alternative flow (Thất bại)	1. Nếu xảy ra các trường hợp khác hệ thống sẽ thông báo lỗi.
<b>Extension point</b>	Không có



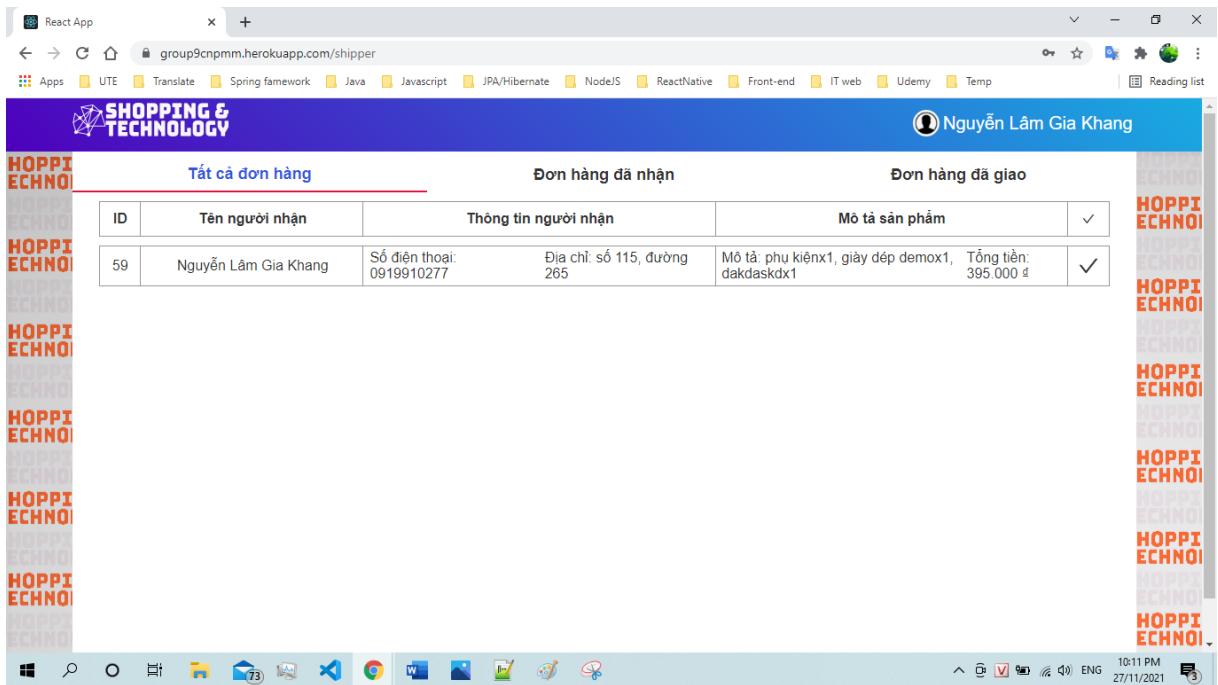
Hình 3. 24: thống kê

#### 4. Tính năng người giao hàng

##### 1. Lấy thông tin đơn hàng cần giao

<b>Name</b>	Lấy thông tin đơn hàng cần giao
<b>Brief description</b>	Lấy thông tin toàn bộ đơn hàng cần giao.
<b>Actor(s)</b>	Shipper
<b>Pre-conditions</b>	Đăng ký tài khoản người giao hàng.
<b>Post-conditions</b>	Hiển thị danh sách đơn hàng cần giao.

Flow of events	
Basic flow (Thành công)	<p>Người dùng thực hiện các bước sau:</p> <ol style="list-style-type: none"> <li>Vào trang người giao hàng.</li> <li>Hệ thống gọi đến API <a href="https://cnpmmbbe.herokuapp.com/shipper">https://cnpmmbbe.herokuapp.com/shipper</a>.</li> <li>BE gửi lại thông tin đơn hàng.</li> <li>Hệ thống hiển thị đơn hàng.</li> </ol>
Alternative flow (Thất bại)	<ol style="list-style-type: none"> <li>Nếu xảy ra các trường hợp khác hệ thống sẽ thông báo lỗi.</li> </ol>
Extension point	Không có

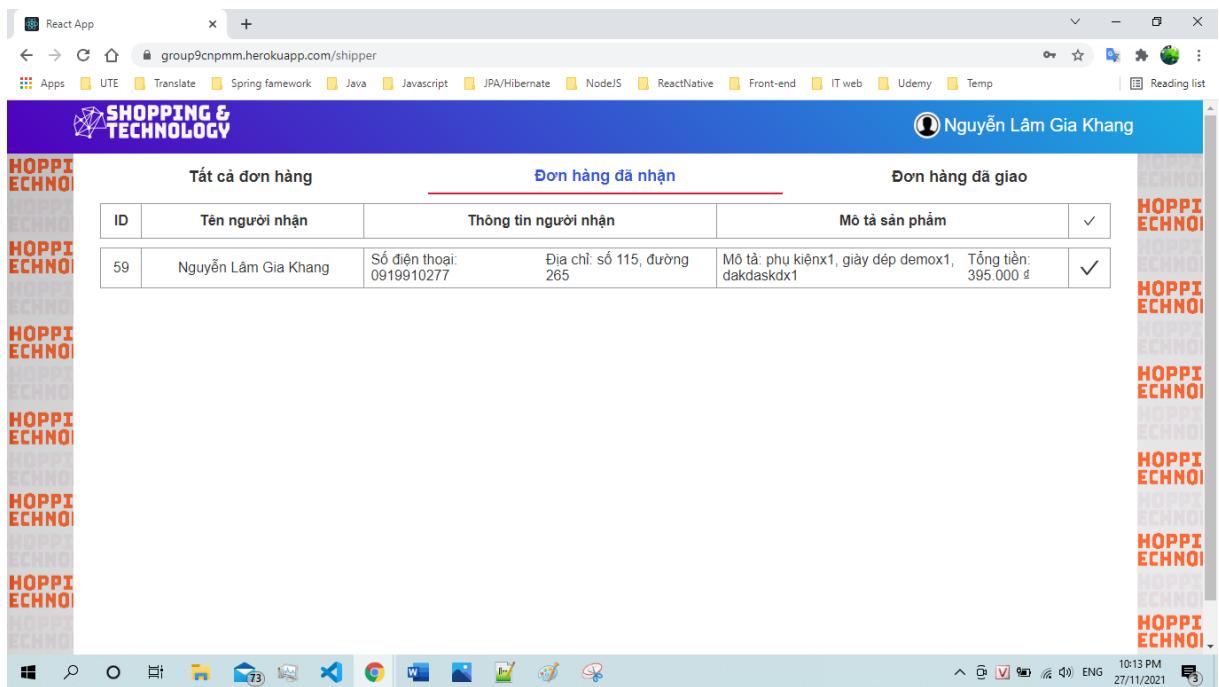


Hình 3. 25: Đơn hàng cần giao

## 2. Đơn hàng đã nhận

Name	Lấy thông tin đơn hàng đã nhận.
Brief description	Lấy thông tin toàn bộ đơn hàng đã nhận.
Actor(s)	Shipper

<b>Pre-conditions</b>	Đăng ký tài khoản người giao hàng.
<b>Post-conditions</b>	Hiển thị danh sách đơn hàng cần giao.
<b>Flow of events</b>	
Basic flow (Thành công)	<p>Người dùng thực hiện các bước sau:</p> <ol style="list-style-type: none"> <li>1. Vào trang người giao hàng.</li> <li>2. Vào mục đơn hàng đã nhận.</li> <li>3. Hệ thống gọi đến API <a href="https://cnpmmbbe.herokuapp.com/shipper/order/0">https://cnpmmbbe.herokuapp.com/shipper/order/0</a>.</li> <li>4. BE gửi lại thông tin đơn hàng.</li> <li>5. Hệ thống hiển thị đơn hàng.</li> </ol>
Alternative flow (Thất bại)	<ol style="list-style-type: none"> <li>1. Nếu xảy ra các trường hợp khác hệ thống sẽ thông báo lỗi.</li> </ol>
<b>Extension point</b>	Không có

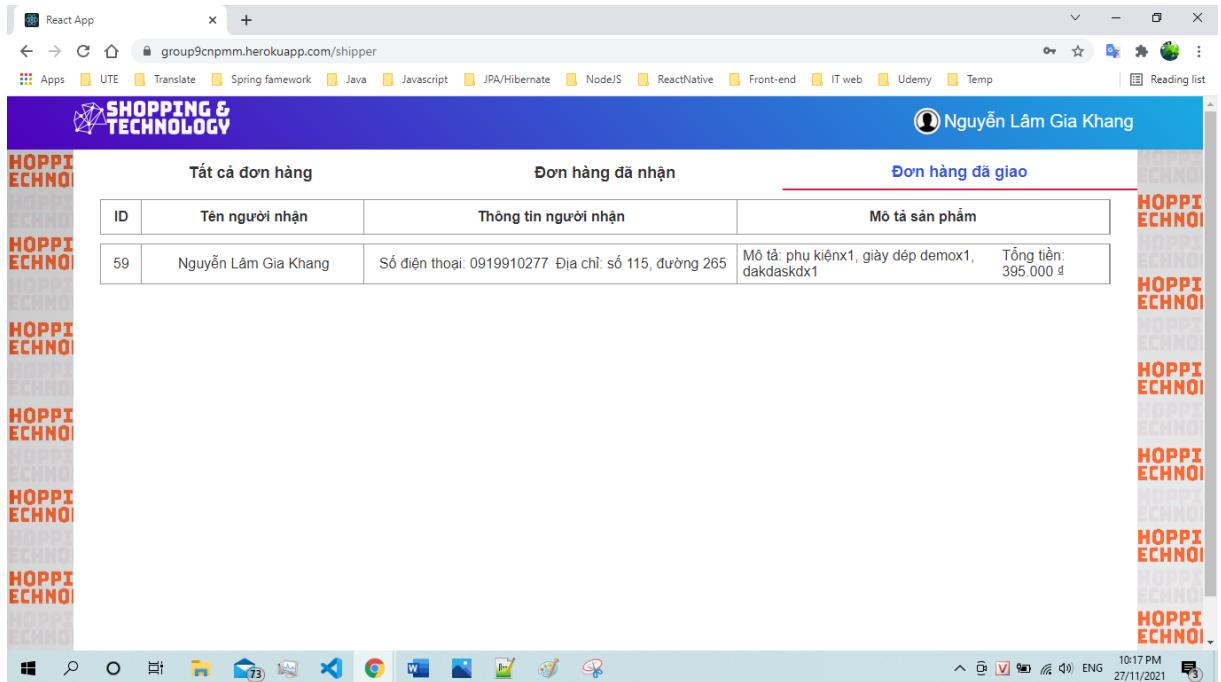


Hình 3. 26: đơn hàng đã nhận

### 3. Đơn hàng đã giao

Name	Lấy thông tin đơn hàng đã nhận.
------	---------------------------------

<b>Brief description</b>	Lấy thông tin toàn bộ đơn hàng đã nhận.
<b>Actor(s)</b>	Shipper
<b>Pre-conditions</b>	Đăng ký tài khoản người giao hàng.
<b>Post-conditions</b>	Hiển thị danh sách đơn hàng cần giao.
<b>Flow of events</b>	
Basic flow (Thành công)	<p>Người dùng thực hiện các bước sau:</p> <ol style="list-style-type: none"> <li>6. Vào trang người giao hàng.</li> <li>7. Vào mục đơn hàng đã giao.</li> <li>8. Hệ thống gọi đến API <a href="https://cnpmmbe.herokuapp.com/shipper/ordersuccess/0">https://cnpmmbe.herokuapp.com/shipper/ordersuccess/0</a>.</li> <li>9. BE gửi lại thông tin đơn hàng.</li> <li>10. Hệ thống hiển thị đơn hàng.</li> </ol>
Alternative flow (Thất bại)	<p>1. Nếu xảy ra các trường hợp khác hệ thống sẽ thông báo lỗi.</p>
<b>Extension point</b>	Không có



Hình 3. 27: Đơn hàng đã giao

## 5. Tính năng admin

### 1. Quản lý người dùng

Name	Quản lý người dùng
<b>Brief description</b>	Lấy thông tin toàn bộ người dùng.
<b>Actor(s)</b>	Admin
<b>Pre-conditions</b>	Đăng nhập bằng tài khoản admin.
<b>Post-conditions</b>	Hiển thị danh sách toàn bộ người dùng.
<b>Flow of events</b>	
Basic flow (Thành công)	<p>Người dùng thực hiện các bước sau:</p> <ol style="list-style-type: none"> <li>Vào trang admin.</li> <li>Hệ thống gọi đến API <a href="https://cnpmbe.herokuapp.com/admin/users">https://cnpmbe.herokuapp.com/admin/users</a>.</li> <li>BE gửi lại thông tin toàn bộ khách hàng.</li> <li>Hệ thống hiển thị thông tin khách hàng.</li> </ol>
Alternative flow	1. Nếu xảy ra các trường hợp khác hệ thống sẽ thông báo lỗi.

(Thất bại)	
<b>Extension point</b>	Không có

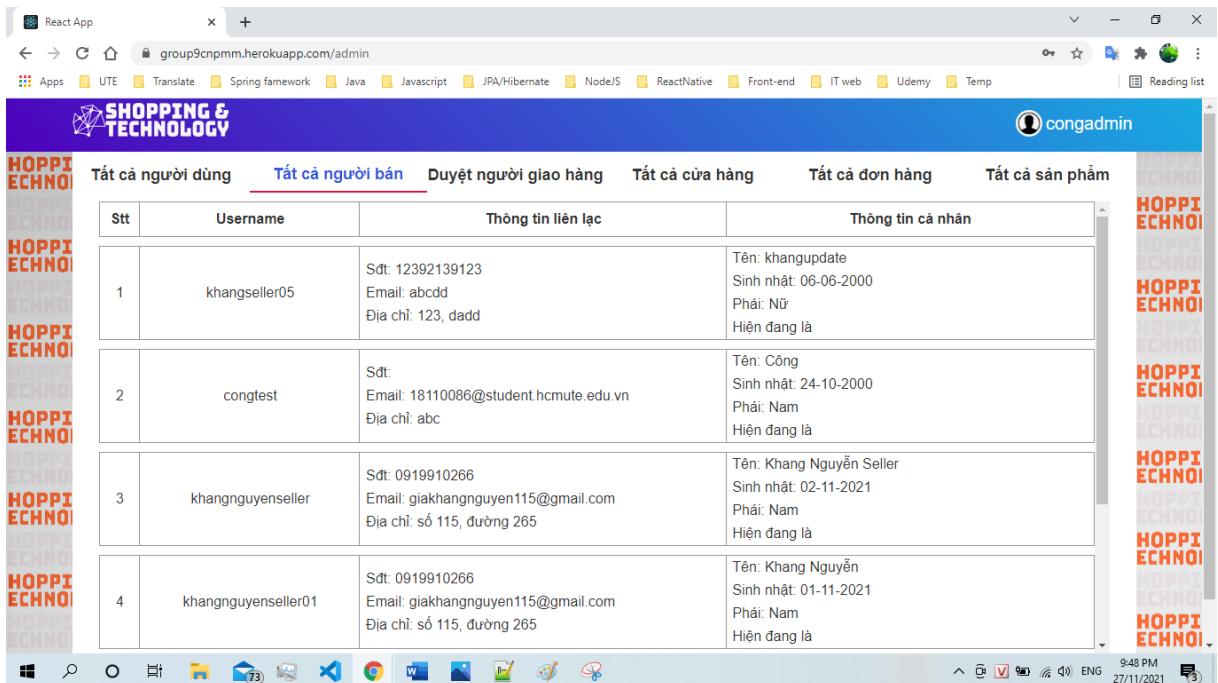
Stt	Username	Thông tin liên lạc	Thông tin cá nhân
1	khang	Sđt: Email: giakhangnguyen115@gmail.com Địa chỉ: 123, dadd	Tên: khang Sinh nhật: 06-06-2000 Phái: Nam Hiện đang là người dùng
2	khangseller	Sđt: Email: abc Địa chỉ: 123, dadd	Tên: khang Sinh nhật: 06-06-2000 Phái: Nam Hiện đang là người bán hàng
3	khangadmin	Sđt: Email: abc Địa chỉ: 123, dadd	Tên: khang Sinh nhật: 06-06-2000 Phái: Nam Hiện đang là người quản trị
4	congadmin	Sđt: Email: cong@gmail.com Địa chỉ: 123, dadd	Tên: cong Sinh nhật: 18-02-2000 Phái: Nam Hiện đang là người quản trị

Hình 3. 28: Quản lý người dùng

## 2. Quản lý người bán

Name	Quản lý người bán hàng
<b>Brief description</b>	Lấy thông tin toàn bộ người bán hàng.
<b>Actor(s)</b>	Admin
<b>Pre-conditions</b>	Đăng nhập bằng tài khoản admin.
<b>Post-conditions</b>	Hiển thị danh sách toàn bộ người bán hàng.
<b>Flow of events</b>	
Basic flow (Thành công)	<p>Người dùng thực hiện các bước sau:</p> <ol style="list-style-type: none"> <li>1. Vào trang admin.</li> <li>2. Vào mục tất cả người bán.</li> <li>3. Hệ thống gọi đến API <a href="https://cnpmbe.herokuapp.com/admin/sellers">https://cnpmbe.herokuapp.com/admin/sellers</a>.</li> </ol>

	4. BE gửi lại thông tin toàn bộ người bán hàng. 5. Hệ thống hiển thị thông tin người bán hàng.
Alternative flow (Thất bại)	1. Nếu xảy ra các trường hợp khác hệ thống sẽ thông báo lỗi.
<b>Extension point</b>	Không có

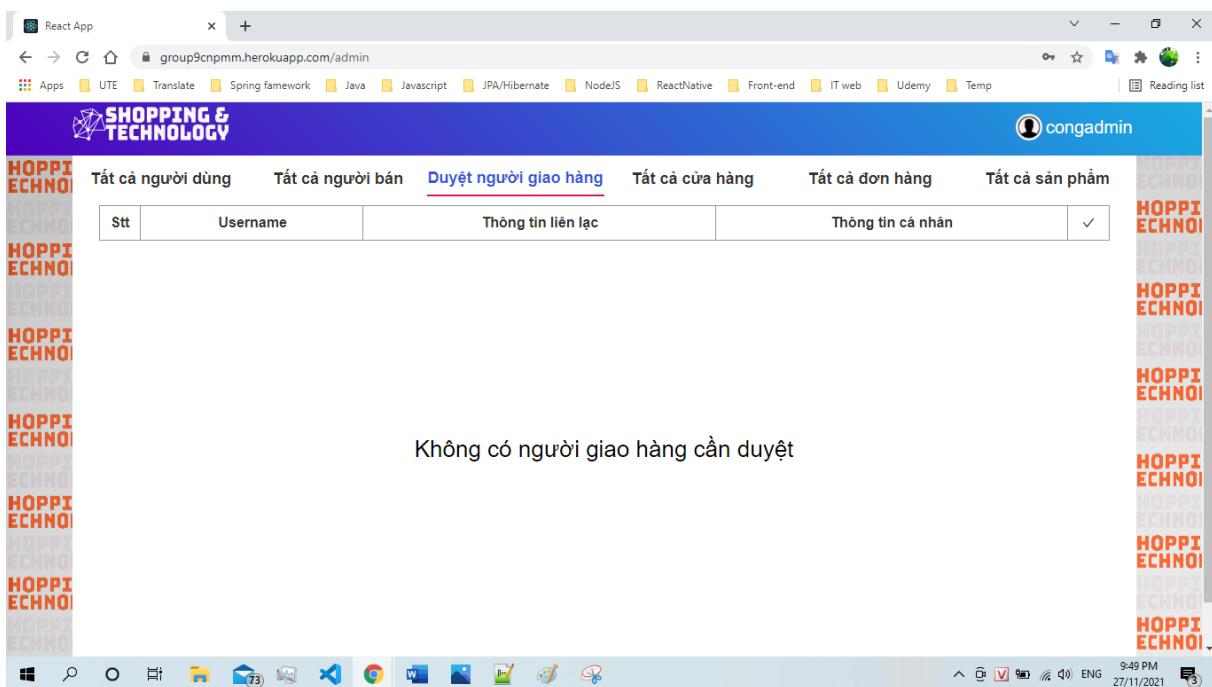


Hình 3. 29: Quản lý người bán

### 3. Duyệt người giao hàng

Name	Quản lý người giao hàng
Brief description	Lấy thông tin toàn bộ người giao hàng.
Actor(s)	Admin
Pre-conditions	Đăng nhập bằng tài khoản admin.
Post-conditions	Hiển thị danh sách toàn bộ người giao hàng.
Flow of events	
Basic flow (Thành công)	Người dùng thực hiện các bước sau: 1. Vào trang admin.

	<p>2. Vào mục tất cả người bán.</p> <p>3. Hệ thống gọi đến API</p> <p><a href="https://cnpmmbbe.herokuapp.com/admin/shippers">https://cnpmmbbe.herokuapp.com/admin/shippers</a>.</p> <p>4. BE gửi lại thông tin toàn bộ người giao hàng.</p> <p>5. Hệ thống hiển thị thông tin người giao hàng.</p>
Alternative flow (Thất bại)	1. Nếu xảy ra các trường hợp khác hệ thống sẽ thông báo lỗi.
<b>Extension point</b>	Không có

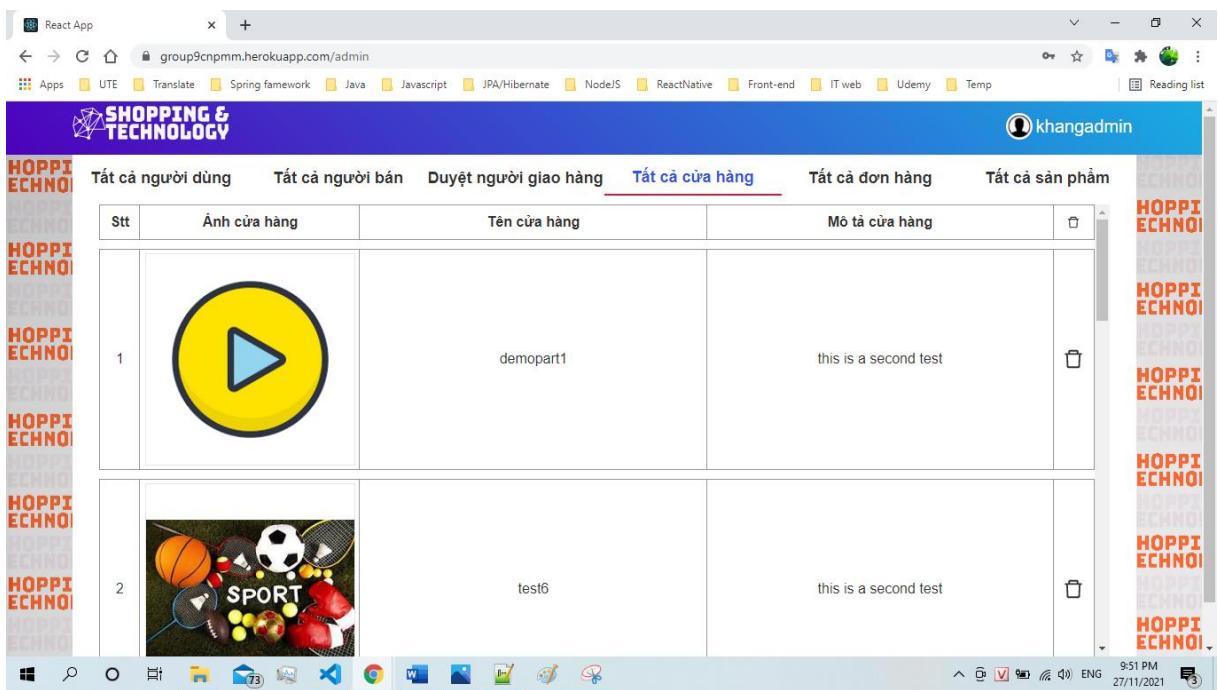


Hình 3. 30: duyệt người giao hàng

#### 4. Quản lý cửa hàng

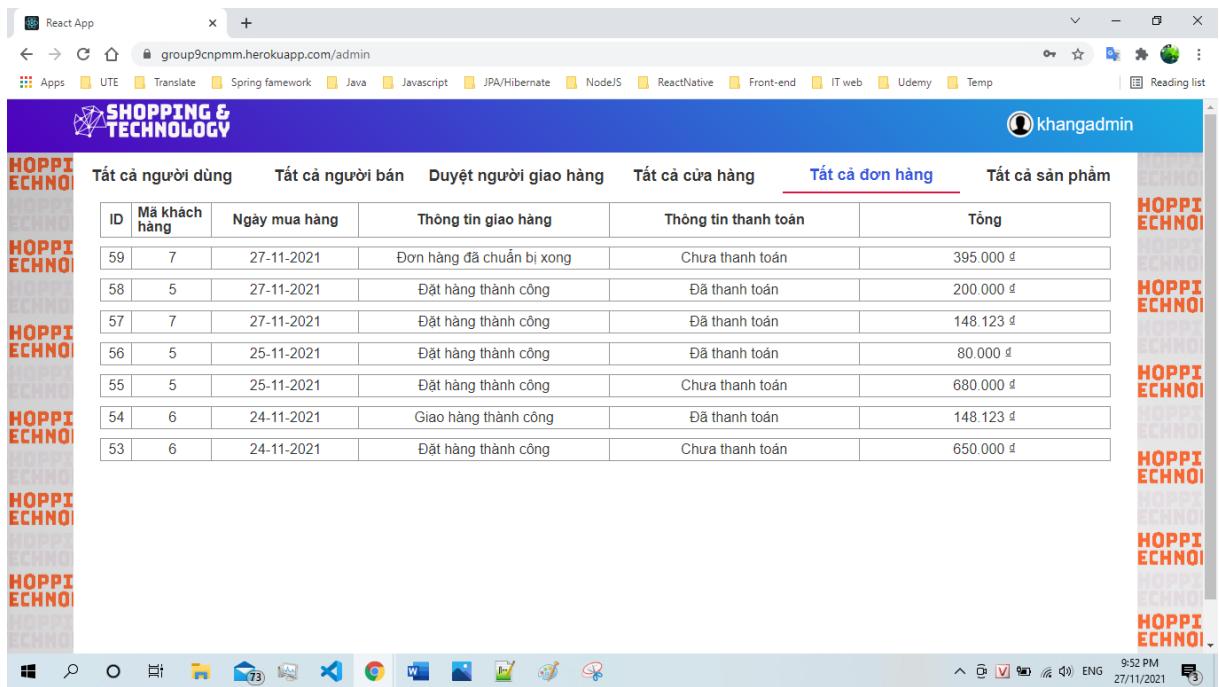
Name	Quản lý cửa hàng
<b>Brief description</b>	Lấy thông tin toàn bộ cửa hàng.
<b>Actor(s)</b>	Admin
<b>Pre-conditions</b>	Đăng nhập bằng tài khoản admin.
<b>Post-conditions</b>	Hiển thị danh sách toàn bộ người giao hàng.

Flow of events	
Basic flow (Thành công)	<p>Người dùng thực hiện các bước sau:</p> <ol style="list-style-type: none"> <li>1. Vào trang admin.</li> <li>2. Vào mục tất cả cửa hàng.</li> <li>3. Hệ thống gọi đến API <a href="https://cnpmbe.herokuapp.com/admin/stores">https://cnpmbe.herokuapp.com/admin/stores</a>.</li> <li>4. BE gửi lại thông tin toàn bộ cửa hàng.</li> <li>5. Hệ thống hiển thị thông tin cửa hàng.</li> </ol>
Alternative flow (Thất bại)	<ol style="list-style-type: none"> <li>1. Nếu xảy ra các trường hợp khác hệ thống sẽ thông báo lỗi.</li> </ol>
Extension point	Không có



Hình 3.31: Quản lý cửa hàng

## 5. Quản lý đơn hàng

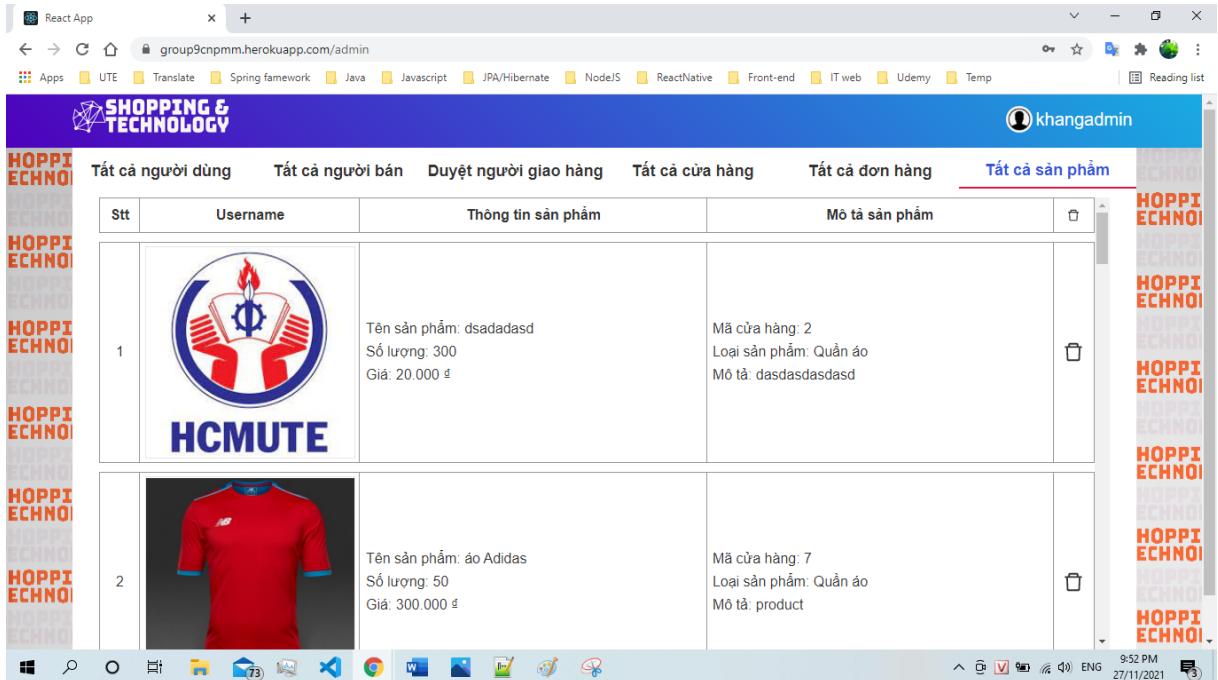


Hình 3. 32: Quản lý đơn hàng

## 6. Quản lý sản phẩm

Name	Quản lý sản phẩm
<b>Brief description</b>	Lấy thông tin toàn bộ sản phẩm.
<b>Actor(s)</b>	Admin
<b>Pre-conditions</b>	Đăng nhập bằng tài khoản admin.
<b>Post-conditions</b>	Hiển thị danh sách toàn bộ sản phẩm.
<b>Flow of events</b>	
Basic flow (Thành công)	<p>Người dùng thực hiện các bước sau:</p> <ol style="list-style-type: none"> <li>Vào trang admin.</li> <li>Vào mục tất cả cửa hàng.</li> <li>Hệ thống gọi đến API <a href="https://cnpmbe.herokuapp.com/admin/products">https://cnpmbe.herokuapp.com/admin/products</a>.</li> <li>BE gửi lại thông tin toàn bộ sản phẩm.</li> <li>Hệ thống hiển thị thông tin sản phẩm.</li> </ol>
Alternative flow	1. Nếu xảy ra các trường hợp khác hệ thống sẽ thông báo lỗi.

(Thất bại)	
<b>Extension point</b>	Không có



Hình 3. 33: Quản lý sản phẩm

Link kiểm thử website: <https://group9cnpm.herokuapp.com/>

#### IV. Sử dụng github

Link: <https://github.com/giakhangnguyenlam/CNPMM>

Backend: <https://github.com/giakhangnguyenlam/CNPMM/tree/main/Back-end>

Front-end: <https://github.com/giakhangnguyenlam/CNPMM/tree/main/Front-end>

Thiết kế:

<https://github.com/giakhangnguyenlam/CNPMM/tree/main/System%20design>

README:

<https://github.com/giakhangnguyenlam/CNPMM/blob/main/Back-end/README.md>

## **V. Các thư viện đã sử dụng**

### **1. Backend**

- Bcrypt: bcrypt là một hàm băm mật khẩu được sử dụng để hash password trong đồ án.
- Cors: một cơ chế cho phép nhiều tài nguyên khác của một trang web có thể được truy vấn từ Front-end với Backend của trang.
- Dotenv: Dùng để load các biến môi trường từ file .env.
- Express: Là một framework được xây dựng trên nền tảng của Nodejs. Nó cung cấp các tính năng mạnh mẽ để phát triển web hoặc mobile. Một số chức năng chính của expressjs: thiết lập các lớp trung gian để trả về các http request, định nghĩa router cho phép sử dụng với các hành động khác nhau dựa trên phương thức http và url, trả về các trang html dựa vào tham số.
- Googleapis: API của google cho phép thực hiện các thao tác trên các ứng dụng mà google cho phép. Trong dự án được sử dụng để upload hình ảnh lên google drive.
- Jsonwebtoken: Dùng phân quyền cho người sử dụng.
- Mongoose: thư viện hỗ trợ các thao tác trên cơ sở dữ liệu mongodb.
- Mongoose-auto-increment: Hỗ trợ tăng giá trị id trong mongodb.
- Nodemailer: dùng gửi mail xác nhận đặt hàng.

### **2. Front-end**

- React: Được biết đến là một thư viện của javascript rất phổ biến để xây dựng giao diện người dùng. Một điều rât tuyệt vời của React là cho khả năng phản hồi nhanh bằng việc sử dụng phương pháp mới để render trang web. Đây là một thư viện xây dựng giao diện người dùng tập trung vào việc xây dựng giao diện React cho phép bạn tạo giao diện người dùng có thể được truy cập trên các công cụ tìm kiếm khác nhau. Tính năng này là một lợi thế rất lớn vì không phải tất cả các khung JavaScript đều thân thiện với SEO. Ngoài ra, vì React có thể tăng tốc quá trình của ứng dụng nên có thể cải thiện kết quả SEO. Cuối cùng tốc độ web đóng một vai trò quan trọng trong tối ưu hóa SEO.

- React-dom: Là một thư viện hỗ trợ cho React làm việc với DOM (mô hình đối tượng tài liệu).
- Axios: Axios là một HTTP client được viết dựa trên Promises được dùng để hỗ trợ cho việc xây dựng các ứng dụng API từ đơn giản đến phức tạp và có thể được sử dụng cả ở trình duyệt hay Node.js. Để sử dụng axios thì ta chỉ cần gọi axios (), truyền vào một object có chứa các thông tin về method, data, url, headers...
- React-paginate: Một React component hỗ trợ cho việc phân trang. Điều ta cần làm là css cho nó là ta đã có thể có phần phân trang cho trang web đẹp và đáp ứng được yêu cầu về phân trang. Như ảnh ở trên ta thấy là cần cung cấp cho nó tên css class ta muốn áp dụng vào các phần của việc phân trang (css của số trang, css của nút tiếp tục, css nút quay lại), nội dung của nút quay lại và tiếp tục...
- React-icon: Là một thư viện cung cấp cho chúng ta một bộ các icon thường dùng trên web giúp cho web trông đẹp hơn. Đây là thư viện có rất nhiều icon.
- React-router-dom: React không được tích hợp sẵn router tuy nhiên ta có thể cài đặt thư viện react-router-dom. Ở đồ án này nhóm em sử dụng BrowserRouter để thực hiện việc chuyển trang.

## **VI. Kết luận**

### **4.1. Đánh giá mức độ hoàn thành**

Nhóm thực hiện tự đánh giá rằng nhóm đã cơ bản hoàn thành các yêu cầu của đồ án. Tất cả các tài liệu về api cũng như các hình ảnh kiểm thử đã được bổ sung đầy đủ. Qua đồ án nhóm cũng đã học thêm được nhiều kiến thức mới như MERN, phương pháp thiết kế một ứng dụng cũng như các kiến thức liên quan đến việc làm việc nhóm, xử lý các tình huống xung đột trên github cũng như có cái nhìn khái quát một ứng dụng được hình thành như thế nào. Tuy nhiên trong quá trình làm đồ án cũng không thể tránh những thiếu sót, nhóm sẽ tiếp tục hoàn thiện và phát triển trong những dự án tiếp theo.

### **4.2. Khó khăn**

Trong quá trình làm việc nhóm chưa thể giải quyết vấn đề bảo mật ứng dụng. Việc sử dụng github cũng gặp đôi chút khó khăn khi xảy ra xung đột. Kiến thức MERN của nhóm cũng chưa hoàn thiện dẫn đến những khó khăn nhất trong việc giải quyết các tình huống phát sinh khi xây dựng ứng dụng. Đồng thời tác động của dịch bệnh nên nhóm chỉ có thể giao tiếp qua google meet nên việc triển khai ứng dụng trở nên khó khăn hơn rất nhiều.

### **4.3. Hướng phát triển**

Trong thời gian tiếp theo, nhóm thực hiện sẽ cố gắng cải thiện web với giao diện cũng như những tính năng mới hơn, đáp ứng đầy đủ yêu cầu của người dùng hơn.

Các vấn đề còn lại của trang web, nhóm sẽ tiến hành tìm kiếm tài liệu để học hỏi nhiều hơn để tích luỹ kiến thức về việc tạo lập một trang web hoàn thiện hơn. Song song với việc trên, nhóm thực hiện sẽ tiến hành kiểm tra lại trang web và tìm cách nâng cấp để làm giảm đi những vấn đề còn hạn chế của phần mềm.

Nhóm sẽ tiếp tục phát triển web theo hướng viết api và sẽ cố gắng cung cấp những phiên bản mới hơn trong thời gian tới đồng thời cũng có kiến thức để ứng dụng việc bảo mật vào web để hoàn thiện ứng dụng.

## **VII. Tài liệu tham khảo**

1. Expressjs là gì? Tất tần tật về Express.js, Bài viết gốc tại: medium.freecodecamp.com, topdev.vn, link: <https://topdev.vn/blog/express-js-la-gi/>