***----------Home Page---------***

***>>> Category***

***Output: lấy ra các loại hàng***

***API: http://localhost:8080/api/categories (method get)***

***Param: không có***

***>>> Auction is happening and is waiting***

***Output: các phiên đấu giá đang diễn ra(trả hết thông tin) khoảng 5 phiên và các phiên đấu giá đang chờ (khoảng 5 phiên)***

***API: http://localhost:8080/api/auctions/auctionings?amount=5 (method get)***

***Param: amount=5 (Số sản phẩm hiển thị ra)***

***>>> Maybe you will want***

***Output: lấy hình + tên sản phẩm, 12 sản phẩm***

***API:http://localhost:8080/api/products?pagenumber=0&amount=12 (method get)***

***Param: pagenumber=0 (lấy trang đầu tiên - index = 0)***

***amount=12 (lấy 12 sản phẩm)***

***>>> Search***

***Input: text***

***Output: tên sp đang đấu + giá phù hợp với text input***

***API: http://localhost:8080/api/products/search/?keyValue=Xiaomi***

***Param: keyValue=Xiaomi (key word nhập vào để tìm kiếm)***

***>>>> API Lấy hãng SX theo tên sản phẩm***

***Input: nhận vào tên loại sản phẩm***

***Output: trả ra các hãng bán sản phẩm dó***

***API: http://localhost:8080/api/categories/manufacturers?name=Điện thoại (method get)***

***Param: name=Laptop (Lấy các hãng bán sản phẩm Laptop)***

***>>>> API Filter***

***Input:***

***- KeyValue để Search***

***- Tên loại sản phẩm***

***- Tên hãng***

***- Các sản phẩm đang trong phiên đấu giá/Các sản phẩm đang được bán trên trang***

***- Tăng hay giảm***

***- Gía cao nhất***

***- Gía thấp nhất***

***Graphical user interface, text

Description automatically generated***

***Dữ liệu nhập vào***

***Output: Map chứa các danh sách theo trường hợp***

***API:*** [***http://localhost:8080/api/products/filter***](http://localhost:8080/api/products/filter) ***(method post)***

***Param: Không có***

***Mô tả:***

* ***nameCategory là tên loại hàng***
* ***nameManufacturer là tên hãng sx***
* ***statusProduct là trạng thái hiển thị, có 3 trường hợp là “all”, “Đang bán trên trang”, “Đang đấu giá”***
* ***increase là tăng hay giảm, tăng thì true, giảm thì false***
* ***maxPrice, minPrice là giá cao nhất, thấp nhất***
* ***keyValue là từ khóa nhập vào để search***

***Note: (Yêu cầu ban đầu)***

***Lưu ý:***

***1. Các dữ liệu số 1,2 đều có thể được trả về với một kiểu dữ liệu là "all“ tức là lấy tất cả (Ví dụ nếu tên loại sp t trả về là "all“ thì m lấy cho t tất cả sản phẩm của tất cả các loại sản phẩm)***

***2. Dữ liệu số 3 cx có thể có giá trị all nhưng nếu nó rơi vào giá trị all bắt buộc m phải chia cho t 2 danh sách, 1 là danh sách các sp đang trong phiên đấu giá, 2 là ds các sp đang được mở bán, còn nếu ko rơi vào trường hợp all thì chỉ cần đưa cho t 1 danh sách tùy theo nội dung dữ liệu ở mục 3***

***3. Các số 4 5 6 có thể sẽ ko có dữ liệu nếu dữ liệu ở mục 3 được chọn là lấy các sản phẩm đang được mở bán (bởi vì nó đang được mở bán nên ko có giá cụ thể để sort hay tìm trong khoảng giữa giá cao nhất và giá thấp nhất)***

***4. Nếu số 4 có dữ liệu thì phải sort theo giá trị hiện tại của sp của các phiên đấu giá đang được mở, nếu số 5, 6 có giá trị thì phải tìm các phiên đấu giá có giá sản phẩm nằm trong khoảng 5 và 6***

***--------Product detail-------------***

***- Input: product\_id***

***- Output:***

***Thông tin của sản phẩm (Tên, hình ảnh)***

***Tên người bán + id người bán***

***Tên thương hiệu (Kèm id nếu có)***

***Tên loại sản phẩm (Kèm id nếu có)***

***Số lượng còn lại trong kho (Nếu ko có thì cứ mặc định là 1)***

***Danh sách các comment của những người từng mua hàng (bao gồm tên ng mua vs comment)***

******

***API:*** [***http://localhost:8080/api/products/displayDetail?id=4***](http://localhost:8080/api/products/displayDetail?id=4)

***Param: id = {Gía trị Product ID cần tìm}***

--------Auction detail------------- (Socket)

(Ở đây có 2 api)

***\*\* Api để lấy các thông tin ban đầu***

***- Input: auction\_id, accountName***

***- Output:***

***Thông tin của sản phẩm (Tên, hình ảnh)***

***Giá hiện tại***

***Danh sách các lần đấu giá (Bao gồm commnet, tên người đưa giá, giá đưa ra, id của người đưa ra giá) <-> Dữ liệu trong Table Auction Detail***

***Các thông tin cơ bản của phiên đấu giá như (Thời gian bắt đầu, thời gian kết thúc***

***, giá khởi điểm , số người tham gia, mã phiên đấu giá, tên người mở phiên)***

***Trạng thái của Người dùng hiện tại đối với phiên đấu giá***

***+ 1: Chủ phiên***

***+ 2: Đang tham gia***

***+ 3: Chưa tham gia***

***+ 4: Chưa đăng nhập***

A screenshot of a computer screen

Description automatically generated with medium confidence

Text

Description automatically generated

***API:*** [***http://localhost:8080/api/auctions/infoauctioning?id=5***](http://localhost:8080/api/auctions/infoauctioning?id=5)***?accountName=***

***Param: id = {auction\_id}***

***accountName= {tên người mở phiên} hoặc accountName=***

\*\* Api để đưa ra giá đấu

Input: id người đưa ra giá, Giá đưa ra, comment (Sử dụng socket ở đây)

***-----------------Check ng đó có bán sp ko--------------***

***Nhận vào id user và id sp, nếu nó có bán sp đó thì trả về true, còn ko bán thì trả về false***

***Input:***

***ID User, ID Product***

***Output:***

***Nếu User có bán sản phẩm đó -> true***

***Nếu User không bán -> false***

***API: http://localhost:8080/api/auth/user/users/checkproductofuser?userName=vanbao&productId=5***

***Param:***

***userName={Tên người dùng cần kiểm tra}***

***productId={Id sản phẩm cần kiểm tra}***

***-----------------Giỏ hàng------------------------***

***Lấy ra ds các sp trong giỏ hàng của user***

***Input: accountName***

***Output: Ds các sp trong giỏ hàng, mỗi sp cần các thông tin sau:***

***Tên sp***

***Hình ảnh***

***Tên người bán (Kèm id người bán)***

***Giá***

***A screenshot of a computer

Description automatically generated with medium confidence***

***API:*** [***http://localhost:8080/api/auth/user/cart?accountName=vanbao***](http://localhost:8080/api/auth/user/cart?accountName=vanbao)

***Param: accountName = {tên tài khoản khách hàng}***

***Lưu ý: Cần đăng nhập lấy token thì mới được xem giỏ hàng***

***\*\* Trang này sẽ có thêm 1 api cho phép xóa sản phẩm đã đấu giá thành công nhận vào input là mã sp đã đấu giá thành công (Chỗ này không xóa được nha, xóa là mấy thông tin liên quan xóa theo)***

***Input: id Auction cần xóa***

***Output:***

***True -> xóa thành công***

***False -> Xóa thất bại***

***API:*** [***http://localhost:8080/api/auth/user/auctions/delete?id=9***](http://localhost:8080/api/auth/user/auctions/delete?id=9)

***Param: id = {Auction Id của phiên cần xóa}***

***-----------Register---------------***

***Input:***

***- Username***

***- Pass***

***Output:***

******

***Nếu tài khoản đã có trong db sẽ không đăng kí được và trả về thông báo không đăng ký thành công***

***API:*** [***http://localhost:8080/register***](http://localhost:8080/register) ***(Method post)***

***-----------Login Page---------------***

***Input:***

***- Username***

***- Pass***

***Output:***



***API:*** [***http://localhost:8080/login***](http://localhost:8080/login) ***(Method post)***

***--------------Địa chỉ giao hàng------------------***

***Lấy danh sách các địa chỉ giao hàng***

***input: userid***

***Output: Ds các địa chỉ giao hàng mà ngta đã thêm, mỗi địa chỉ cần các thông tin sau***

***id của địa chỉ giao hàng***

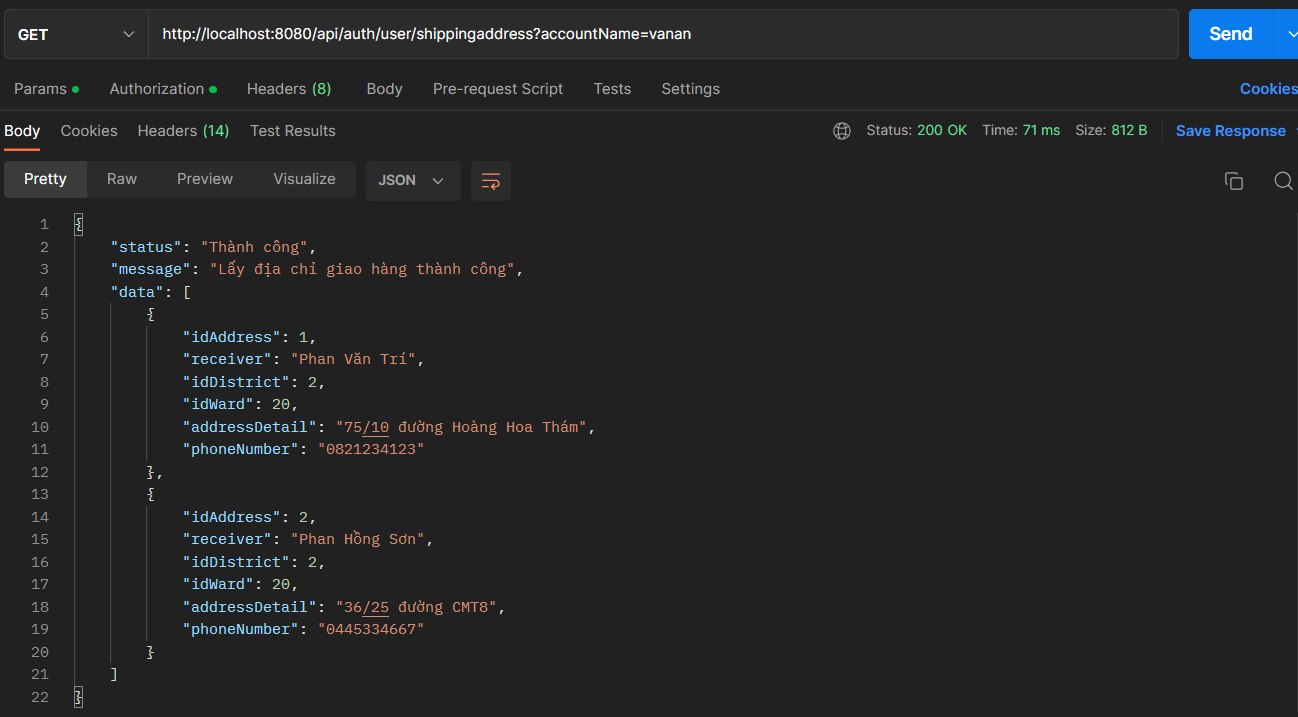
***name (cái này ko phải tên của user mà là tên của ng nhận mà ngta đã ghi cho địa chỉ đó)***

***mã quận***

***mã huyện***

***Address detail***

***Số điện thoại***

******

***API: http://localhost:8080/api/auth/user/shippingaddress?accountName=vanan***

***Param: accountName = {accountName của User}***

***Lưu ý: Cần đăng nhập trước khi gọi API***

***\*\* Trang này có thêm 2 api để thêm địa chỉ và xóa địa chỉ, xóa thì input là id của địa chỉ giao hàng đó***

* ***API Insert:*** [***http://localhost:8080/api/auth/user/shippingaddress***](http://localhost:8080/api/auth/user/shippingaddress) ***(Method Post)***

***Input:***

***{***

***"acccountName": "vanbao",***

***"wardId": 20,***

***"phone": "0441556778",***

***"fullName": "Trần Chí Kiên",***

***"addressDetails": "B44/76 Lý Chính Thắng",***

***"status": true***

***}***

***Output: Thông báo thành công và thông tin sản phẩm vừa thêm***

***Param: không có***

* ***API delete:*** [***http://localhost:8080/api/auth/user/shippingaddress?id=4***](http://localhost:8080/api/auth/user/shippingaddress?id=4) ***(Method delete)***

***Input: Không***

***Output: Dữ liệu địa chỉ vừa xóa và thông báo xóa thành công***

***Param: id={Id của địa chỉ cần xóa}***

***Lưu ý: cần đăng nhập***

* ***API update:http://localhost:8080/api/auth/user/shippingaddress?id=5 (Method Put)***

***Input:***

***{***

***"acccountName": "vanbao",***

***"wardId": 20,***

***"phone": "0441556778",***

***"fullName": "Nguyễn Văn Bình",***

***"addressDetails": "B44/76 Lý Chính Nghía",***

***"status": true***

***}***

***Output: Dữ liệu địa chỉ mới cập nhật và thông báo đã cập nhật thành công***

***Param: id = {địa chỉ address cần cập nhật}***

***\*\* Ngoài ra m làm thêm cho t api lấy danh sách các quận (District) và 1 cái api t nhập id quận vô là lấy đc danh sách các huyện (Ward)***

* ***API lấy danh sách Quận: http://localhost:8080/api/auth/user/shippingaddress/district***

***Input: Không có***

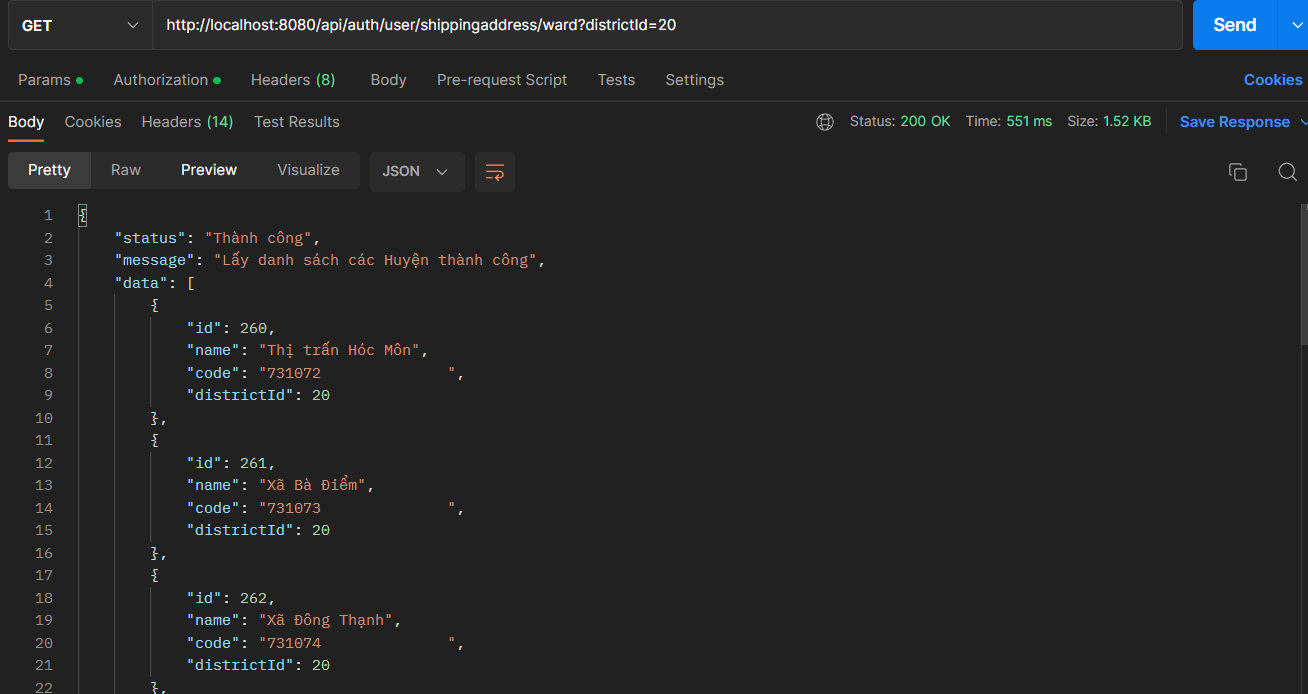
***Output: Danh sách các quận***

***Param: Không có***

* ***API lấy danh sách Huyện:*** [***http://localhost:8080/api/auth/user/shippingaddress/ward?districtId=20***](http://localhost:8080/api/auth/user/shippingaddress/ward?districtId=20) ***(method Get)***

***Input: Id của quận***

***Output: Danh sách các huyện của quận***

******

***Param: id ={id của quận}***

-----------Thanh toán--------------

Ở đây cần các api sau

1/ Api tính tiền ship, dữ liệu nhận vào sẽ là id địa chỉ giao hàng của thằng ng mua và id của ng bán (Dùng id người bán để tìm địa chỉ nơi ở hiện tại của nó để xác định địa chỉ hiện tại của thằng bán rồi tính tiền ship), m chỉ cần trả cho t số tiền ship lấy đc là đc (Ok vậy t sẽ gửi cho m một cái mảng to mỗi cái mảng có 3 giá trị là id ng bán vs id địa chỉ giao hàng vs auctionid nha)

Giống như này này: [

{

sellerID: ….,

addressID: ….,

auctionid: …..

}

]

2/ Api tạo đơn hàng: sau khi thanh toán hoàn tất thì t sẽ gọi cái này để m thêm dữ liệu vào order (t sẽ gửi m một mảng các auctionid)

3/ Làm cho t cái api đổi từ tiền việt sang tiền dola, cái này ko lq tới db t chỉ cần gửi m số tiền Việt rồi m trả t số tiền dola là đc

***-------------------Trang User----------------***

***Api 1: Api lấy tất cả các thông tin của user (Ng đang đăng nhập)***

[***http://localhost:8080/api/auth/user/users/info***](http://localhost:8080/api/auth/user/users/info)

***Input: Token sau khi login thành công***

***Output: Thông tin User***

***Param: không có***

***Api 2: Api dùng để thêm, sửa thông tin user***

[***http://localhost:8080/api/auth/user/users/info***](http://localhost:8080/api/auth/user/users/info)

***Input:***

***{***

***"accountName": "vanan",***

***"pass": 123,***

***"imageUser": "urlll",***

***"statusUser": true,***

***"roles": "USER",***

***"firstName": "Nguyễn Văn",***

***"lastName": "An",***

***"birthday": "2022-02-23",***

***"email": "vanan@gmail.com",***

***"phone": "0984449556",***

***"wardId": 30,***

***"wardName": "Phường 10",***

***"addressDetail": "79/80 Nguyễn Thị Minh Khaiiii"***

***}***

***Output:***

***Text

Description automatically generated***

***Param: Không có***

***Lưu ý: cần login và gửi token***

***Api 3: api dùng để lấy danh sách các thông báo của user (Lấy tất cả những dữ liệu có thể lấy được trong bảng notification ngoài ra lấy thêm cho t tên sản phẩm, hình ảnh sản phẩm và chi tiết phiên đấu giá nếu có auctionid, nếu auction id bằng null thì khỏi lấy chi tiết phiên đấu giá) + cộng với api để xóa thông báo***

* ***API Lấy danh sách thông báo:*** [***http://localhost:8080/api/auth/user/users/notification***](http://localhost:8080/api/auth/user/users/notification) ***(Get method)***

***Input: Gửi Token sau khi login***

***Output: Lấy tất cả dữ liệu tương ứng trong Notification***

***Chi tiết phiên đấu gía***

* ***API xóa thông báo:*** [***http://localhost:8080/api/auth/user/users/notification/delete?id=1***](http://localhost:8080/api/auth/user/users/notification/delete?id=1) ***(Put method)***

***Input: id cần xóa, Token***

***Output: Boolean true/false thông báo xóa thành công hay thất bại***

***Param: id={id thông báo cần xóa}***

***Api 4: api dùng để lấy danh sách các đơn hàng của user ( lấy cho t hình ảnh sp, tên sp, tên người bán, tên người mua, id mỗi cái và trạng thái của đơn hàng)***

***API:*** [***http://localhost:8080/api/auth/user/users/orders***](http://localhost:8080/api/auth/user/users/orders) ***(Method get)***

***Param: Không có***

***Input: Token sau khi login***

***Output: Danh sách các hóa đơn của Username đang đăng nhập***

***\*\* trong api 4 có thêm 2 api nhỏ nữa***

* ***1 là api lấy các chi tiết như: địa chỉ giao hàng, thông tin người mua (Tên, sdt, email)***

***API:*** [***http://localhost:8080/api/auth/user/users/detailorders?id=1***](http://localhost:8080/api/auth/user/users/detailorders?id=1) ***(method Get)***

***Param: id = {Id hóa đơn cần lấy chi tiết}***

***Input: id hóa đơn cần lấy, token sau khi login***

***Output: Chi tiết hóa đơn***

* ***2 là api để chuyển trạng thái đơn hàng từ đang giao sang đã nhận hàng***

***API:*** [***http://localhost:8080/api/auth/user/users/orders/changestatus?id=1***](http://localhost:8080/api/auth/user/users/orders/changestatus?id=1) ***(METHOD PUT)***

***Param: id = {Id hóa đơn cần thay đổi trạng thái}***

***Input: Id hóa đơn cần thay đổi trạng thái, token sau khi login***

***Output: Thông báo hiển thị xóa thành công/thất bại***

----------------Trang User store-----------------

Yêu cầu 1: tạo crud để xem xóa sửa các sản phẩm mà user đang bán

Yêu cầu 2: Phần này cx là phần quản lý đơn hàng nhưng dùng cho phần ng bán, tuy vậy có thể dùng chugn api4 ở phía trên trong trang user chỉ cần thêm một api thay cho yêu cầu số 2 trong api4 là thay vì chuyển trạng thái từ đang giao hàng sang dã nhận hàng thì chuyển trạng thái từ đang đóng gói sang đang giao hàng (Có thể dùng chung bằng cách ngoài id user t gửi theo thì t sẽ gửi cho m thêm 1 param để check xem nó đang trong trang ng mua hay trong trang người bán, tên param m có thể tự đặt)

Yêu cầu 3: tạo 1 api lấy ra danh sách các phiên đấu giá mà user này đang tham gia bao gồm các phiên đã đấu giá thành công và các phiên đang trong quá trình đấu giá

Yêu cầu 4: Tạo 1 api cho phép lấy danh sách các sản phẩm mà user này đang bán nhưng có thêm 1 bộ lọc nhỏ là t sẽ gửi m thêm dữ liệu là loại sản phẩm (VD: laptop, điện thoại,...), cũng sẽ có trường hợp t sẽ gửi dữ liệu là all. M có thể làm luôn ở yêu cầu 1 cx đc tùy m hoặc tách ra 2 api

--------------Admin page-------------------

Yêu cầu 1: Api tính tổng doanh thu trong 1 tháng hiện tại

http://localhost:8080/api/ auth/admin/current-month-stactistics

Phương thức GET

output:

Text

Description automatically generated

Yêu cầu 2: Api tính tổng doanh thu trong 7 tháng gần nhất, từng tháng á (Trả về cho t 2 dữ liệu thôi là tên tháng và số tiền doanh thu)

<http://localhost:8080/api/auth/admin/a-month-stactistics?month=12&year=2022>

Phương thức GET

intput: truyền vào parameter month và year

output: tương tự yêu cầu 1

Yêu cầu 3: Api tính tổng số đơn hàng đã được đặt trong tháng hiện tại

<http://localhost:8080/api/auth/admin/count-curent-month-orders>

Phương thức GET

Input: không có

Output:

Text

Description automatically generated

Yêu cầu 4: Api tính tổng số đơn hàng trong 7 tháng gần nhất, từng tháng (Cứ trả về tên tháng và dữ liệu)

<http://localhost:8080/api/auth/admin/count-last-7months-orders?month=12&year=2022>

input: parameter month và year, là thời gian tháng gần nhất

output:

Text

Description automatically generated

Yêu cầu 5: Tính tổng các user hiện tại hệ thống đang có (trừ admin)

<http://localhost:8080/api/auth/admin/count-user>

Phương thức GET

Output: Graphical user interface

Description automatically generated

Yêu cầu 6: Api lấy danh sách các phí ship giữa các quận (Dữ liệu trả về phải có tên quận gửi, tên quận nhận, giá ship) + api sửa xóa và thêm (Phần thêm này cần check xem dữ liệu tên quận nhận và tên quận gửi được nhập vào đã tồn tại trên database chưa, nếu chưa thì thêm vào nếu rồi thì thông báo không thêm đc)

Yêu cầu 7: API thêm sửa xóa cho loại sản phẩm, và api thêm sửa xóa cho nhãn hàng

API thêm, sửa cho loại sản phẩm

http://localhost:8080/api/auth/admin/add-or-update-category

Phương thức POST

Input: id = null nếu là thêm mới, id = số có sẵn là sửa

Text

Description automatically generated

Output:

Text

Description automatically generated

API xóa loại sp

http://localhost:8080/api/auth/admin/delete-category?id=9

Phương thức DELETE

Input: parameter id (truyền vào id để xóa loại hàng)

Output:

Yêu cầu 8: API lấy danh sách các user trừ admin, tất cả thông tin user + API de khóa hoặc mở khóa cho user (Giống như trạng thái của user, là active hay blocked, nếu chưa có trạng thái này thì m thêm dữ liệu vô ha)

API lấy user

<http://localhost:8080/api/auth/admin/get-all-user>

Phương thức GET

Output: Text

Description automatically generated

API thay đổi trạng thái user

<http://localhost:8080/api/auth/admin/change-user-status>

Phương thức POST

Input: Graphical user interface

Description automatically generated with medium confidence

Output:

Text

Description automatically generated