***d----------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***

***- 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)***

Text

Description automatically generated

Text

Description automatically generated

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

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

\*\* 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: userid

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

\*\* 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

--------------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 thành phố

mã quận

mã huyện

Address detail

Số điện thoại

\*\* 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

-----------Trang thanh toán-----------

Lấy cho t cái phí ship là đc (Tính toán sao đó tự lo)

Trang này sẽ có 1 api xác nhận đã thanh toán

T chưa nghĩ đc input, chắc bấm nút thanh toán xong thì m tự động tạo đơn hàng tự lưu vô database

-------------Trang quản lý cửa hàng----------------

\*\* All Product: Lấy tất cả các sản phẩm đang bán (Làm đầy đủ CRUD)

Input là userid

Output vẫn là danh sách và cần các thông tin sau

Hình ảnh sp,

Tên sp

Tên loại sp

Tên Hãng

\*\* Quản lý đơn hàng

Cần 3 api

1. Input là user id, lấy ds các đơn hàng có các thông tin sau (id đơn hàng,hình ảnh, tên sp, tên người mua kèm id của ng mua, trạng thái đơn hàng,)
2. Input là id đơn hàng, lấy cho t các thông tin đơn hàng như sau (Tên sp, hình ảnh, id sp, trạng thái đơn hàng, giá thành, mã thành phố, mã quận, mã huyện, address detail, tên người mua, số điện thoại)
3. Input là mã đơn hàng, api này dùng cho người bán xác nhận đã đóng gói hàng và chuyển trạng thái từ đang đóng gói sang đang vận chuyển.

***-----------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)***