# DANH SÁCH API VÀ PHÂN CÔNG VIỆC

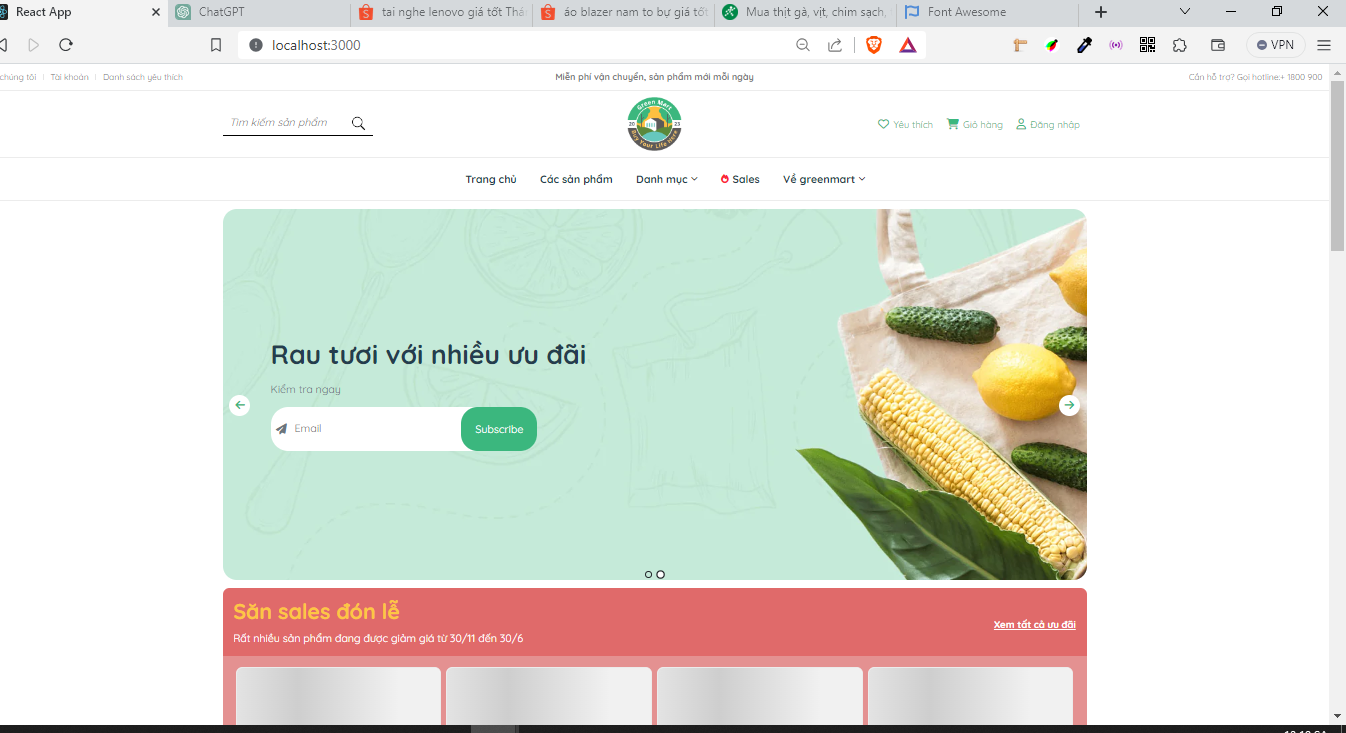
Lưu ý:

Các api sẽ làm tiếng việt

Các API từ mục 1 đến 10 là ĐÃ CHỐT (trong môn web cũ tụi t), các API mới của môn mobile thêm vào sẽ bắt đầu từ 11, hoặc thêm vào các mục từ 1 tới 10 thì sẽ đánh dấu \*\* phía trước, ví dụ 9.5. \*\*Lấy thông tin đơn hàng cũ.

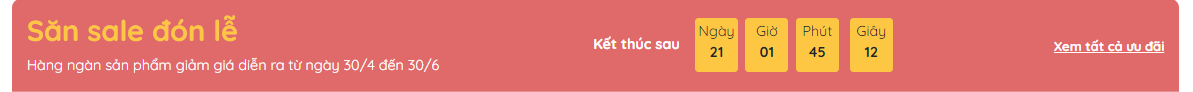
Tham chiếu hết ở trong link db nhe, có j sai sót ae chỉnh lại trực tiếp trong file word này. Còn này là link so đồ data: <https://dbdiagram.io/d/dbMini_market-6558ee393be14957874291fb>

# Trang chủ



## Sales

### Sales event



- \*\*Endpoint:\*\* `GET /api/sales`

- \*\*Mô tả:\*\* Lấy thông tin event, theo t nghĩ thì bên bảng sales event nên thêm vô thuộc tính isOpen để check nó đang được sales

Trả về:

= {

        id: 1,

        name: "Săn sales đón lễ",

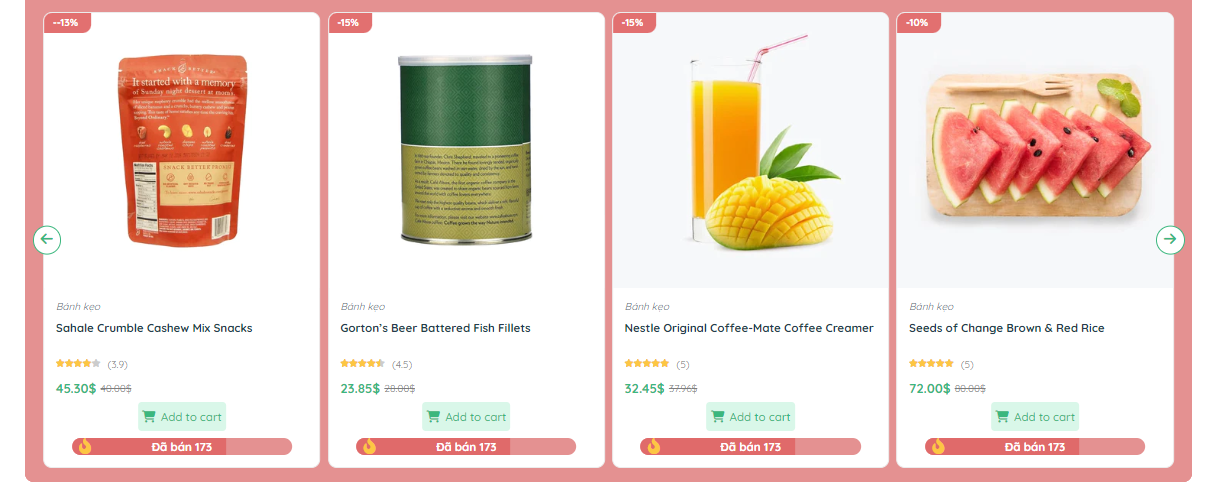
description: ‘Hàng ngàn sản phẩm giảm giá đang diễn ra từ 30/4 đến 30/6

        start\_time: "Nov 11, 2023 00:00:00", // string theo format này

        end\_time: "Dec 30, 2023 00:00:00",

      };

### Sales item



- \*\*Endpoint:\*\* `**GET /api/sales/sanpham**`

- \*\*Mô tả:\*\* Lấy danh sách sản phẩm của sales event hiện tại (nên thêm thuọc tính isOpen cho bảng sales event ?). Sales item sẽ được hiển thị ở 2 chỗ là hình phía trên và trong url /sales

Trả về:

[ một mảng danh sách sản phảm đang sales

{

          id: 2, //id san pham

          name: "Sữa bò Long Thành 100% thiên nhiên, ít béo, phù hợp cho người lớn tuổi và con nít",

          thumbnail:

            "https://bizweb.dktcdn.net/100/144/367/articles/sua-tuoi-edd60355-d601-4684-b461-0ffa984eb6f4.jpg?v=1611374579213",

          reg\_price: 10000,

          discount\_price: 5000,

          discount\_percent: 10,

          rating: 4.3,

          category\_name: "Sữa bò",

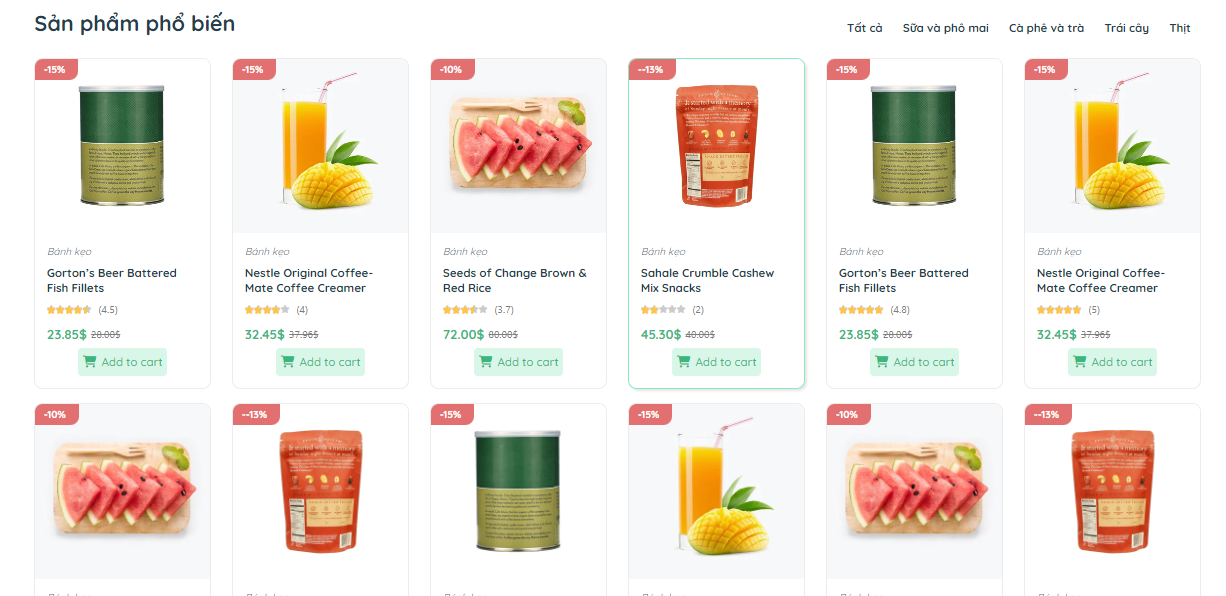
          quantity: 300, // sản phẩm tổng cộng

remaining: 233, // sản phẩm còn lại

        },

]

## Lấy Danh sách Sản phẩm Phổ biến:



- \*\*Endpoint:\*\* `**GET /api/sanpham/phobien** `

- \*\*Mô tả:\*\* Lấy danh sách các sản phẩm phổ biến. (Các sản phẩm phổ biến là các sản phẩm có thuộc tính featured = true)

Phản hồi: cấu trúc như bên dưới, là một mảng gồm nhiều object category, với mỗi category thì có danh sách sản phẩm , **yeu cầu lấy khoảng 5 category và mỗi category lấy ít nhất 12 sản phẩm**

state.data = [

        {

          categoryID: 1,

          name: "Sữa tươi",

          products: [

            {

              id: 1, //id san pham

              name: "Sữa bò Long Thành",

              thumbnail: "url",

              reg\_price: 10000,

              discount\_price: 5000,

              discount\_percent: 10,

              rating: 4.3,

              category\_name: 'Sữa bò'

            },

          ],

        },

        {

          categoryID: 2,

          name: "Bánh quy",

          products: [

            {

              id: 1, //id san pham

              name: "Bánh quy Oreo",

              thumbnail:

                "https://cdn.tgdd.vn/Products/Images/3357/79861/bhx/banh-quy-nhan-kem-vani-oreo-goi-665g-202209120921387006.jpg",

              reg\_price: 10000,

              discount\_price: 5000,

              discount\_percent: 10,

              rating: 4.8,

              category\_name: 'Bánh quy sữa'

            },

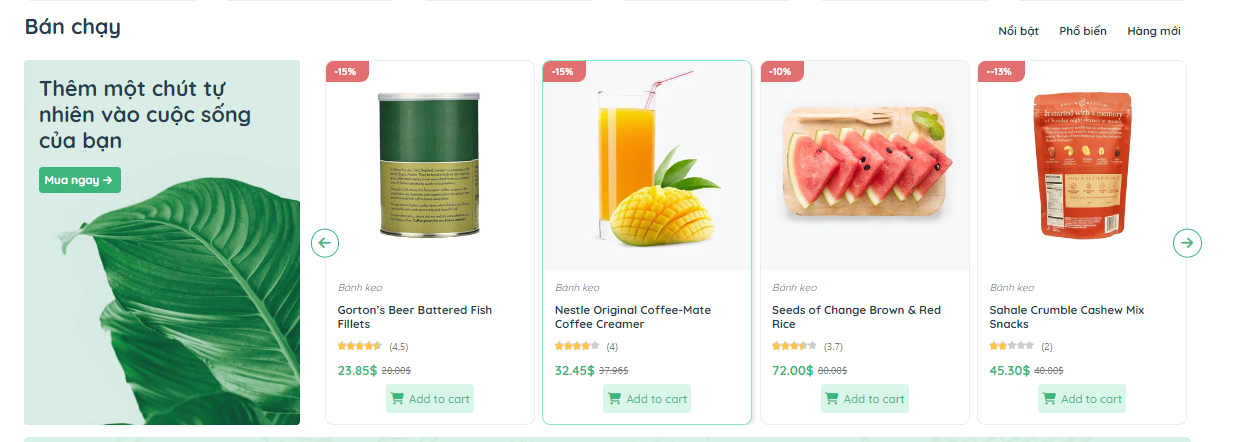
          ],

        },

        // .....

      ];

## dLấy Danh sách Sản phẩm Bán chạy: productsBestSellSlice



- \*\*Endpoint:\*\* `**GET /api/sanpham/banchay** `

- \*\*Mô tả:\*\* Với mỗi type (nổi bật, …) lấy 6 sản phẩm

- \*\*Phản hồi:\*\*

state.data = [

        {

          type: "Nổi bật",

query: 'noibat',

          products: [

//Danh sách sản phẩm, mỗi sản phẩm có dạng

{

id: number;

  name: string;

  thumbnail: string;

  reg\_price: number;

  discount\_price: number;

  discount\_percent: number,

  rating: number,

  category\_name: string

}

],

        },

        {

          type: "Phổ biến",

query: ‘phobien’

          products: [],

        },

        {

          type: "Hàng mới",

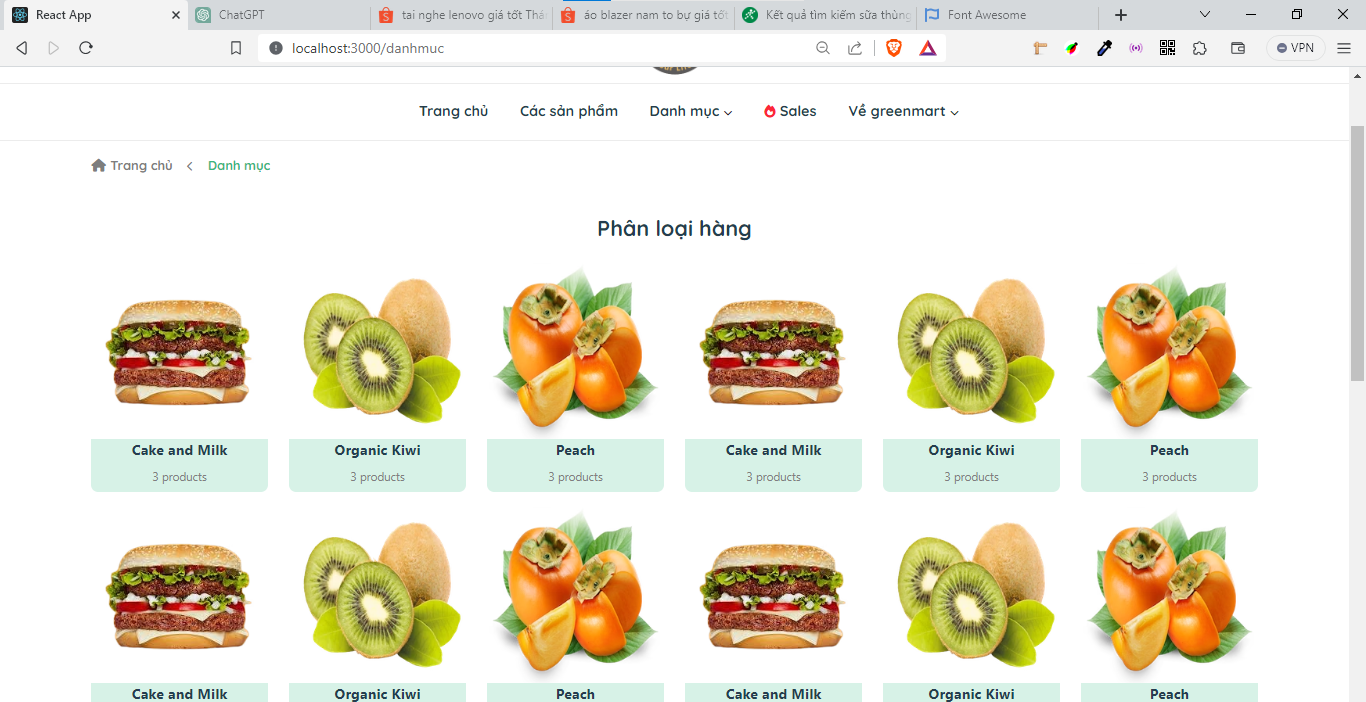
query: ‘hangmoi’

          products: [],

        },

      ];

# Trang Danh mục (Phân loại hàng)



## Lấy danh sách các danh mục lớn: categoryGroupSlice.ts

* + Đường dẫn: **GET /api/danhmuc**
  + Mô tả: Lấy tất cả categoryGroup, dùng để hiện thị danh sách các danh mục thực phẩm (tat ca danh muc, lấy danh mục lớn nha categoryGroup, không phải category)
  + Phản hồi: Mảng các đối tượng danh mục.

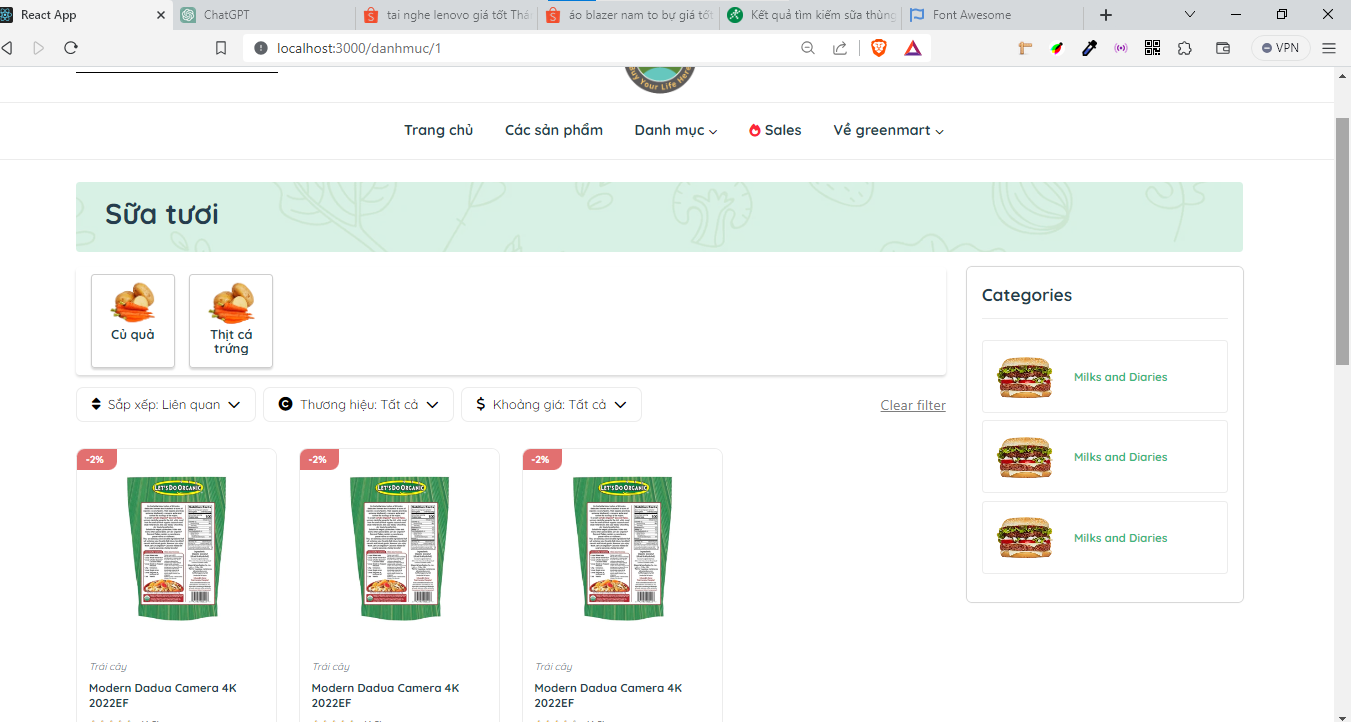
jsonCopy code

// Phản hồi với danh sách danh mục sản phẩm [

{ "id": "1", "name": "Danh mục 1", "thumbnail": "url" }

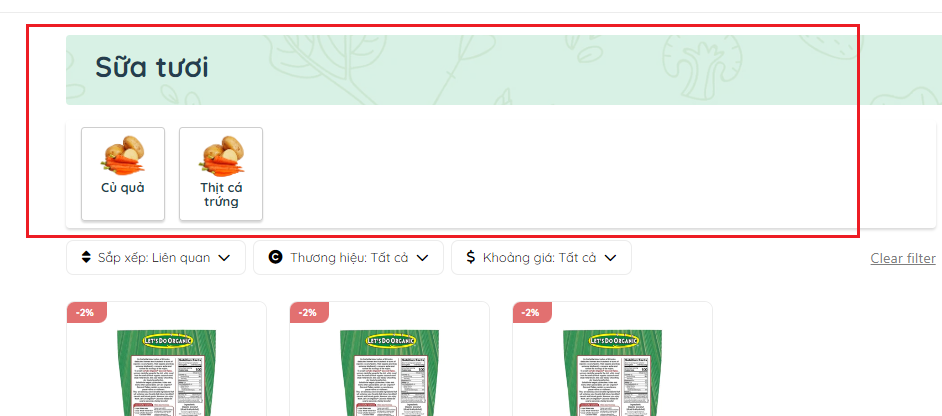
]

# Trang danh sách sản phẩm của danh mục



## Lấy danh sách các danh mục nhỏ thuộc danh mục lớn

* + Đường dẫn: **GET /api/danhmuc/:id/danhmucnho**
  + Mô tả: Lấy tất cả danh mục nhỏ bao gồm tên, thumbnail, id phục vụ cho việc filter các sản phẩm thuọc trang sản phẩm chỗ này. Khi người dùng mới chọn một danh mục lớn (ví dụ thịt, cá, trứng sữa) và bay đến trang này, nó sẽ hiển thị toàn bộ sản phẩm thuộc danh mục lơnd đó, sau đó người dùng chọn mục thịt trên thanh (bên dưới sữa tươi) thì sẽ filter ra các sản phẩm thuộc danh mục nhỏ “Thịt”.



* + Phản hồi: Chi tiết danh mục và danh sách sản phẩm trong danh mục.

```json

{

**categoryGroupName: "Sữa tươi",**

  list: [

    {

      id: 1,

      name: "Thịt",

      thumbnail: "thit.png",

    },

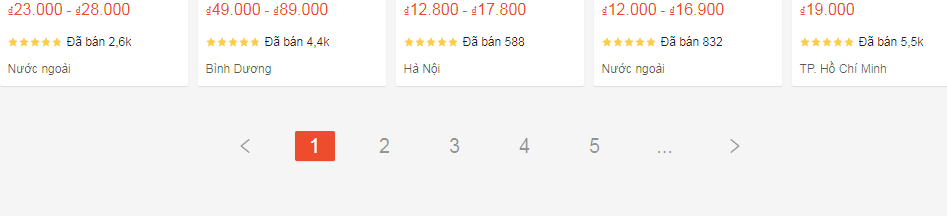
  ],

};

```

## Lấy tất cả sản phẩm thuộc danh mục lớn theo filter và phân trang

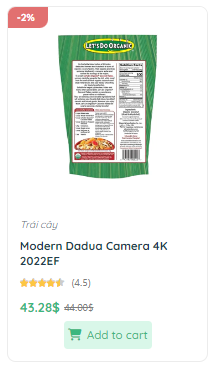
* Đường dẫn: GET **/api/danhmuc/:id?page=0&categoryId=1&sort=az&brand=23&range=0-50**
  + Page: Giá trị 0 -> vô cùng. Dùng để phân trang, ví dụ = 0 thì trả về 16 sản phẩm đầu tiên, = 1 thì (1+1) trả về 16 sản phẩm tiếp theo (giống cơ chế phân trang của Shopee)



* + categoryID: giá trị là id của category dùng để filter thịt cá hay trứng.
  + sort: có các giá trị “sort=banchay” là Bán chạy, “sort=tenaz” tên từ a-z, “tenza”, “giathap” Giá từ thấp tới cao, “giacao” là ngược lại
  + brand: thương hiệu, giá trị là id của thương hiệu, ví dụ Chinsu có id là 23 thì brand=23
  + Range: 0-50 là từ 0 đến dưới 50k, 50-100 từ 50 đến dưới 100k và 100-00 là trên 100k
* Mô tả: Trả về tên để hiển thị dòng sữa tươi chỗ này

jsonCopy code

Trả về danh sách sản phẩm, mỗi sản phẩm hiển thị cho t:



{

data: [

{

id: number;

  name: string;

  thumbnail: string;

  reg\_price: number;

  discount\_price: number;

  discount\_percent?: number,

  rating: number,

  category\_name: string

},

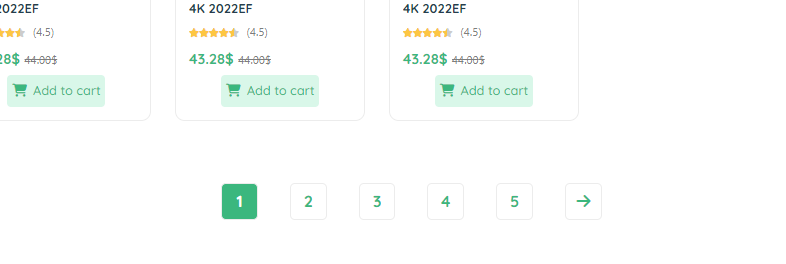
…

]

}

## Trả về tổng số trang pagination cho danh sách sản phẩm

* + Đường dẫn: **GET /api/danhmuc/:id/sotrang?page=0&categoryId=1&sort=az&brand=23&range=0-50**
  + Mô tả: Trả về tổng số trang có thể cho thanh pagination



* + Phản hồi: Chi tiết danh mục và danh sách sản phẩm trong danh mục.

```json

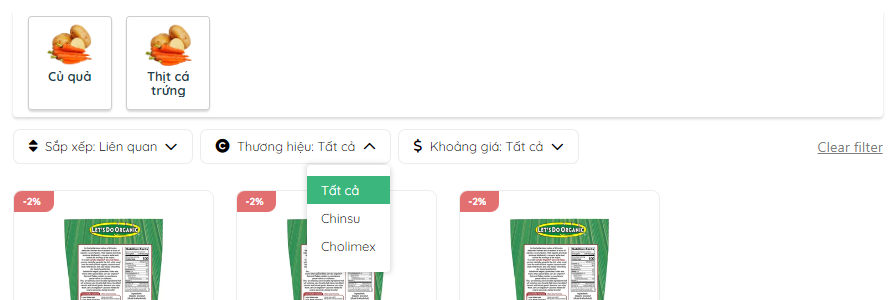
{

n=3;

}

```

## Lấy danh sách thương hiệu có trong danh mục lớn



* + Đường dẫn: **GET /api/danhmuc/:id/thuonghieu**
  + Mô tả: Lấy danh sách thương hiệu có thể có trong 1 danh mục lớn (categoryGroup) để bỏ vào filter (lấy chừng <15 cái)
  + Phản hồi:

[

{

brandName: “Hương việt”,

id: 1,

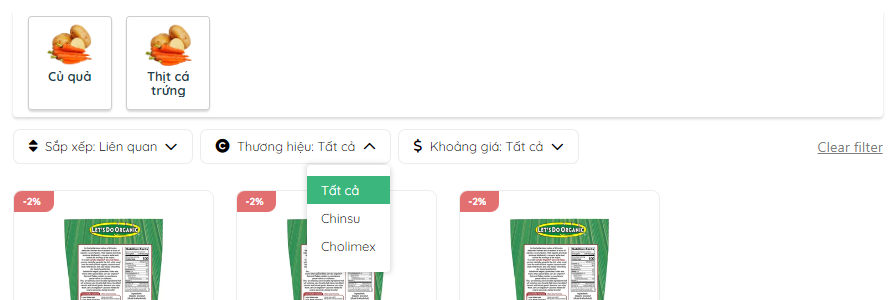
thumbnail: …

},

…

]

## Lấy danh sách thương hiệu có trong danh mục nhỏ



* + Đường dẫn: **GET /api/danhmucnho/:id/thuonghieu**
  + Mô tả: Lấy danh sách thương hiệu có thể có trong 1 danh mục nhỏ để bỏ vào filter. Khác với 4.3 ở chỗ là ban đầu khi người dùng chưa chọn vào category nào thì nó sẽ lấy tất cả thương hiệu của danh mục lớn, nhưng khi chọn vào rồi thì nó sẽ lấy tất cả thương heiue danh mục nhỏ
  + Phản hồi:

[

{

brandName: “Hương việt”,

id: 1,

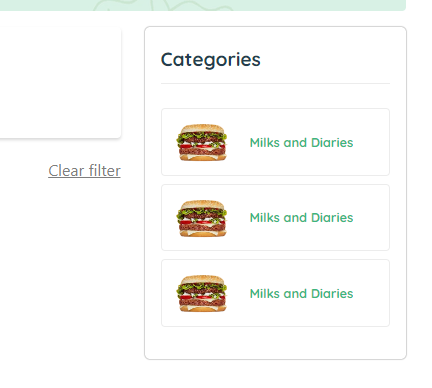
thumbnail: …

},

…

]

## Lấy random danh sách danh mục lớn (khác với danh mục lớn hiện tại)



* + Đường dẫn: **GET /api/danhmuc/:id/random**
  + Mô tả: Lấy danh sách random danh mục lớn hiển thị bên thanh bên phải (lấy random 4 mục khác với mục hiện tại :id)
  + Phản hồi:

[

{

    id: 1, // id category Group để bỏ vào url truy cập đến

    name: "Banh keo, trai cay",

    thumbnail: "url"

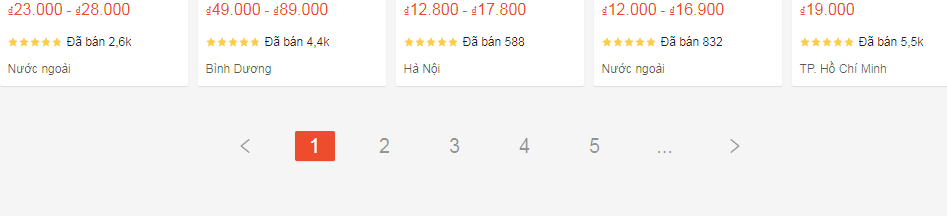
}

…

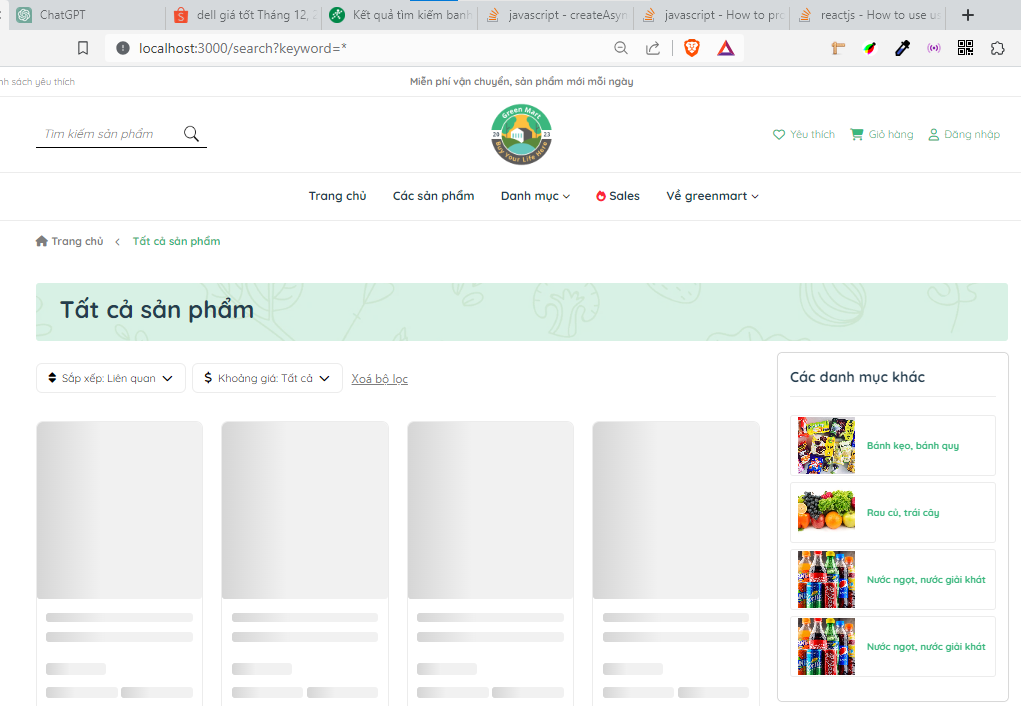
]

1. **Trang danh sách sản phẩm: tất cả / sales / tìm kiếm**

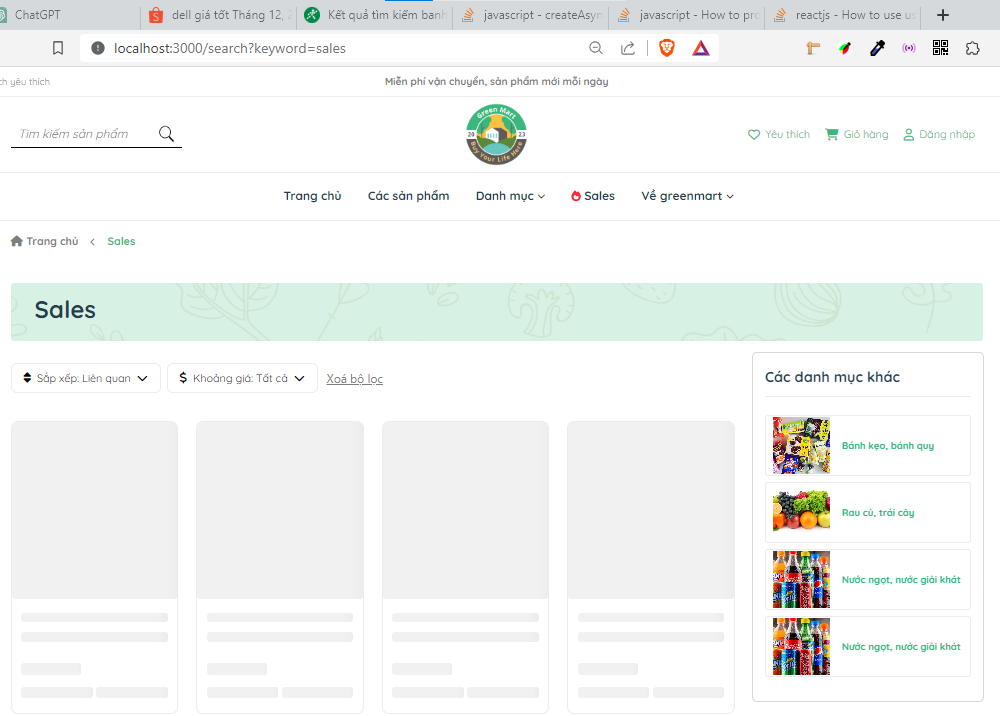
* - Đường dẫn: GET **/api/danhmuc/:id?keyword=\*&page=0&sort=az&range=0-50**
  + Page: Giá trị 1 -> vô cùng. Dùng để phân trang, ví dụ = 1 thì trả về 16 sản phẩm đầu tiên, = 2 thì (1+1) trả về 16 sản phẩm tiếp theo (giống cơ chế phân trang của Shopee) (mặc định khi URL ko có thuộc tính page thì mặc định nó là 1)



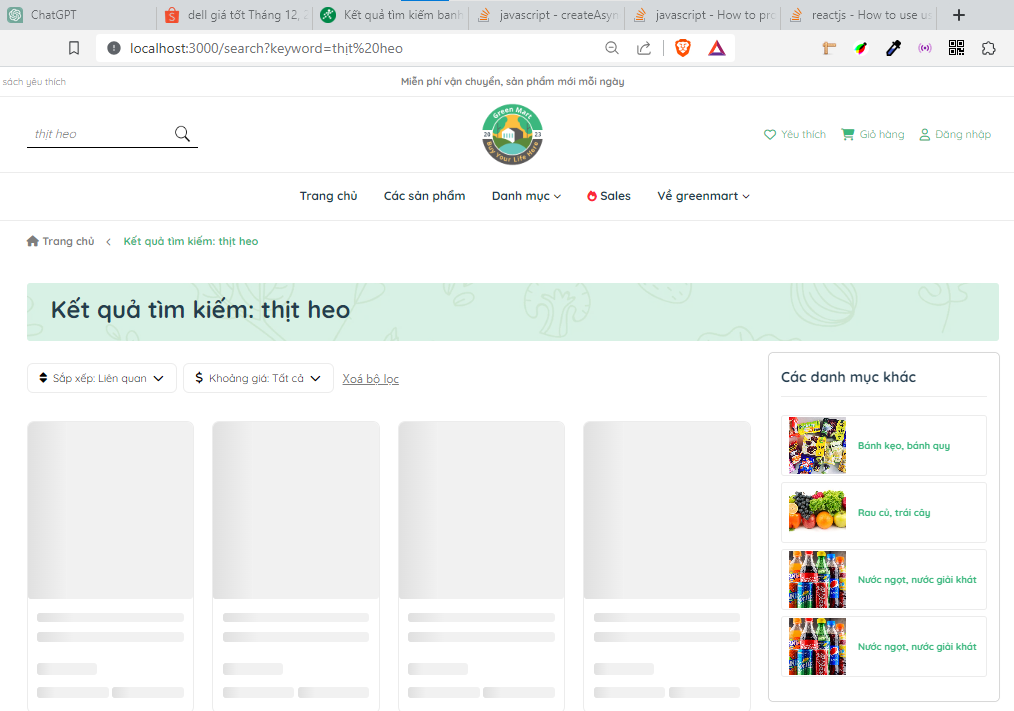
* + sort: có các giá trị “sort=banchay” là Bán chạy, “sort=tenaz” tên từ a-z, “tenza”, “giathap” Giá từ thấp tới cao, “giacao” là ngược lại
  + range: 0-50 là từ 0 đến dưới 50k, 50-100 từ 50 đến dưới 100k và 100-00 là trên 100k
  + **keyword: tuỳ vào giá trị keyword mà ta có ba trang khác nhau**
    - **keyword=\*: trang danh sách tất cả sản phẩm**



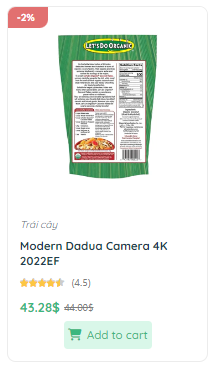
* + - keyword=sales: trang các sản phẩm sales



* + - **keyword=bất kỳ (ví dụ sẽ có dạng ?keyword=thịt%20heo ngoại trừ 2 cái phía trên- > là trang tìm kiếm**



Trả về danh sách sản phẩm, mỗi sản phẩm hiển thị cho t:



[

{

id: number;

  name: string;

  thumbnail: string;

  reg\_price: number;

  discount\_price: number;

  discount\_percent?: number,

  rating: number,

  category\_name: string

},

…

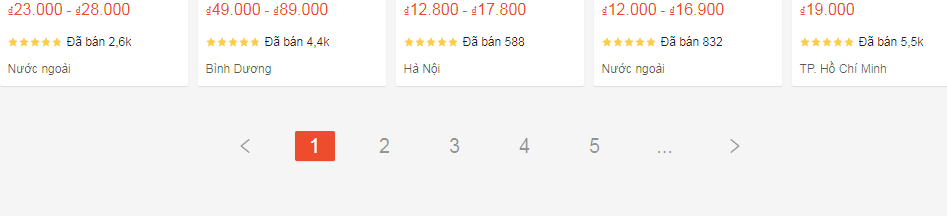
]

# Trang danh sách sản phẩm: tất cả / sales / tìm kiếm

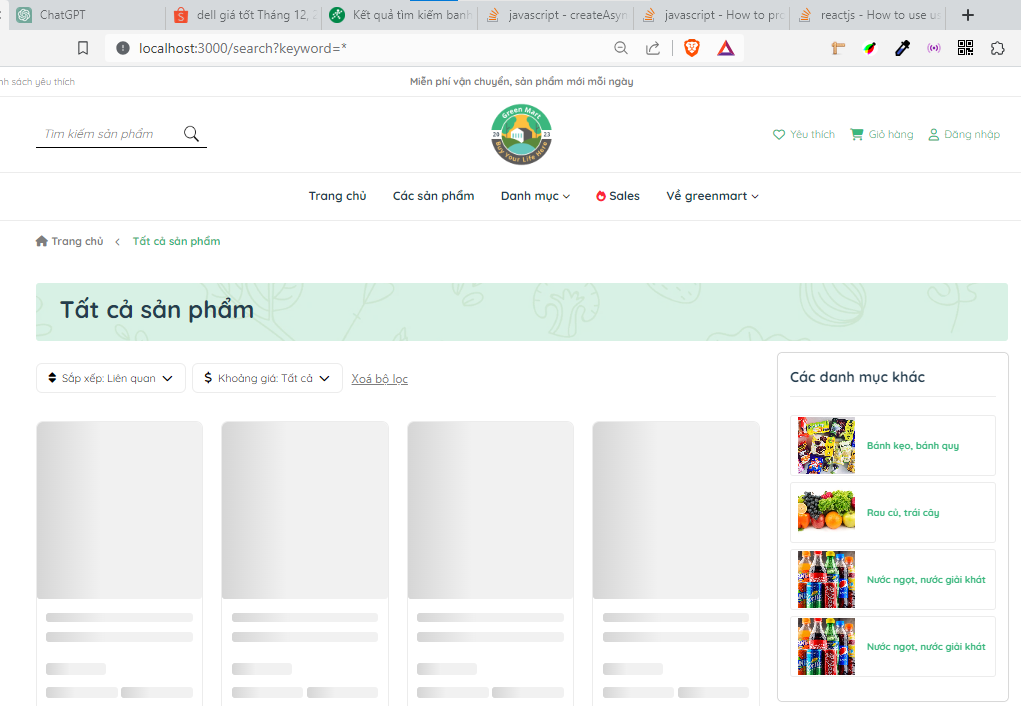
* Đường dẫn: GET **/api/search?keyword=\*&page=0&sort=az&range=0-50**

(ví dụ /search?keyword=\* hay /search?keyword=thịt+heo hay /search?keyword=sales)

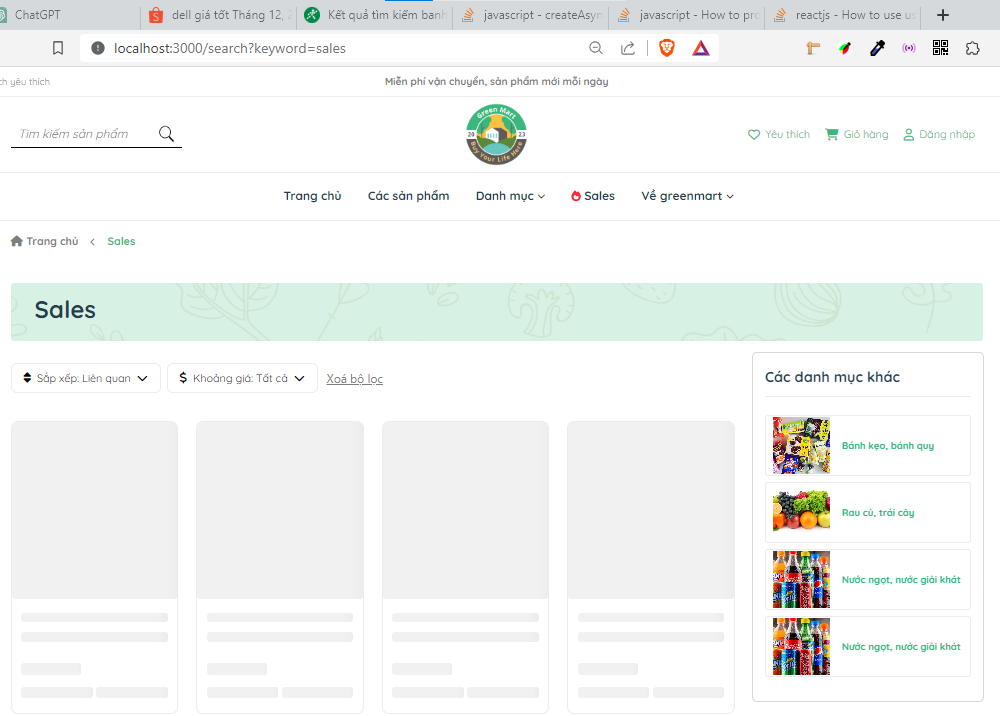
* + Page: Giá trị 1 -> vô cùng. Dùng để phân trang, ví dụ = 1 thì trả về 16 sản phẩm đầu tiên, = 2 thì (1+1) trả về 16 sản phẩm tiếp theo (giống cơ chế phân trang của Shopee) (mặc định khi URL ko có thuộc tính page thì mặc định nó là 1)



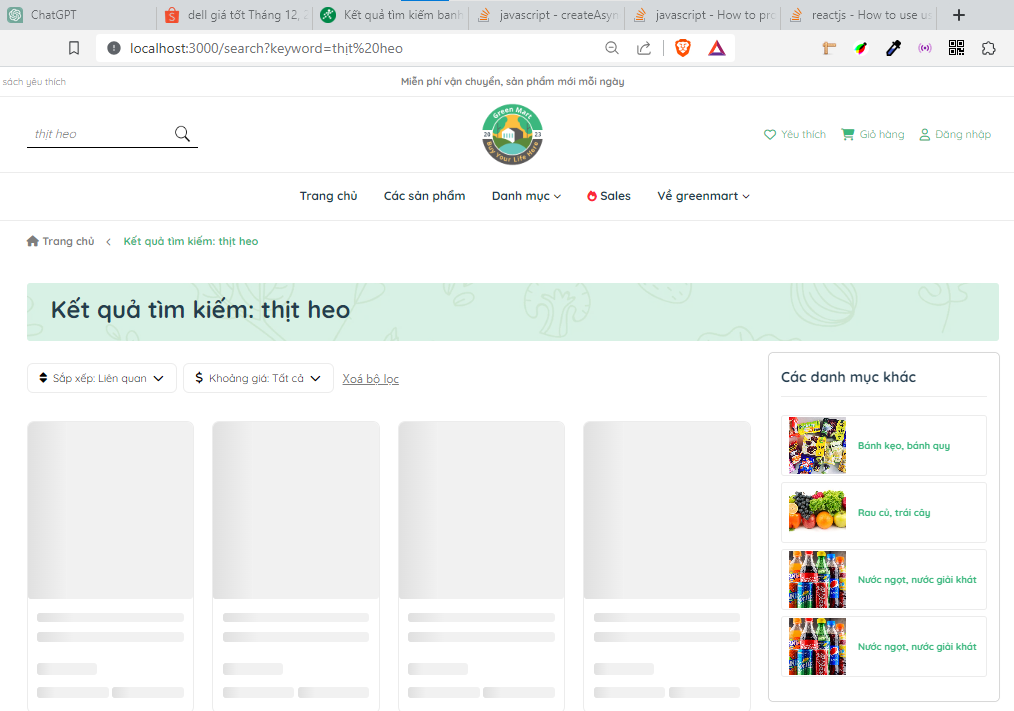
* + sort: có các giá trị “sort=banchay” là Bán chạy, “sort=tenaz” tên từ a-z, “tenza”, “giathap” Giá từ thấp tới cao, “giacao” là ngược lại
  + range: 0-50 là từ 0 đến dưới 50k, 50-100 từ 50 đến dưới 100k và 100-00 là trên 100k
  + **keyword: tuỳ vào giá trị keyword mà ta có ba trang khác nhau**
    - **keyword=\*: trang danh sách tất cả sản phẩm**



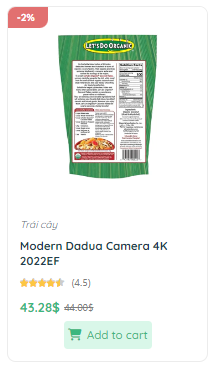
* + - keyword=sales: trang các sản phẩm sales



* + - **keyword=bất kỳ (ví dụ sẽ có dạng ?keyword=thịt%20heo ngoại trừ 2 cái phía trên- > là trang tìm kiếm**



Trả về danh sách sản phẩm, mỗi sản phẩm hiển thị cho t:



[

{

id: number;

  name: string;

  thumbnail: string;

  reg\_price: number;

  discount\_price: number;

  discount\_percent?: number,

  rating: number,

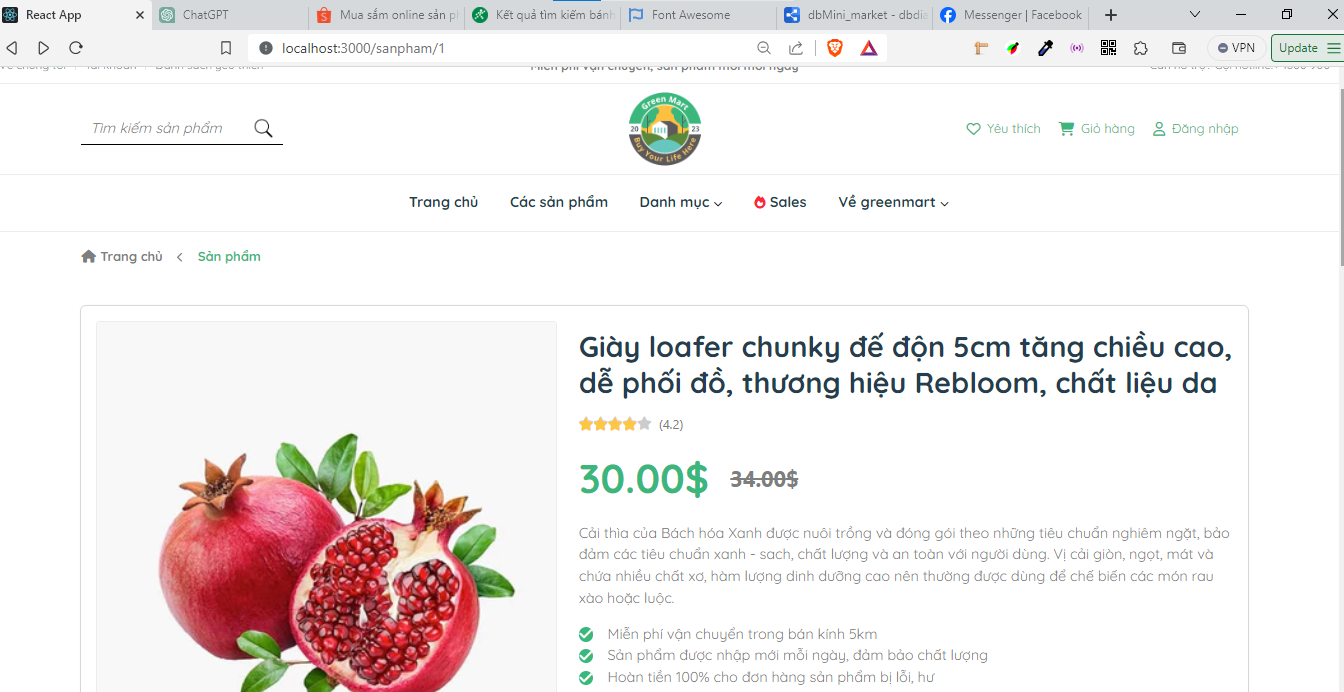
  category\_name: string

},

…

]

# Trang thông tin sản phẩm

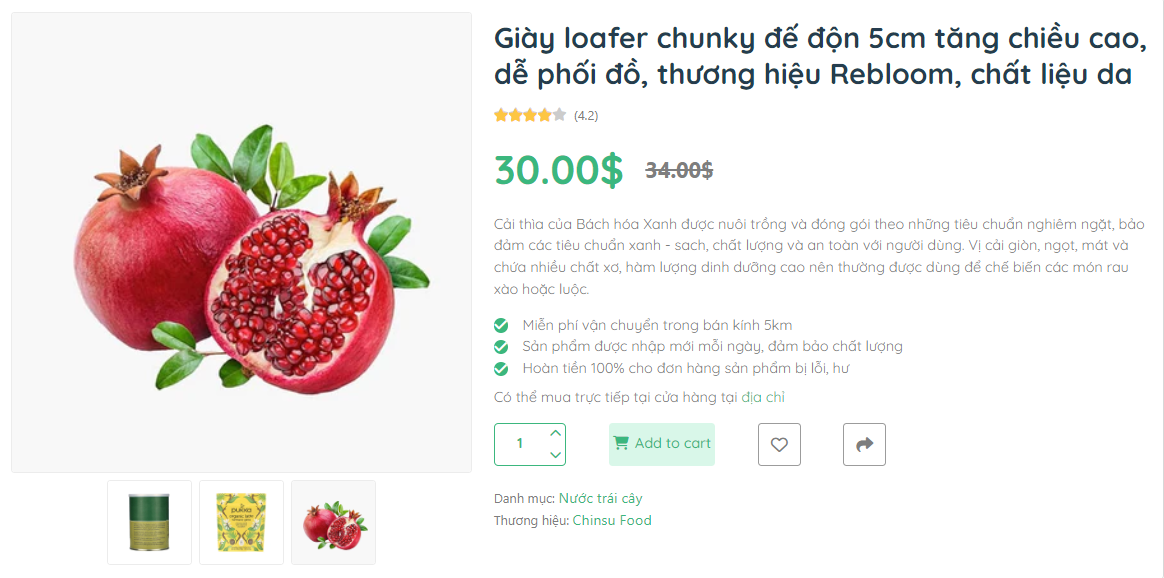


## Lấy Chi tiết Sản phẩm FF

- Đường dẫn: `**GET /api/sanpham/:idsanpham `**

- Mô tả: Lấy chi tiết của một sản phẩm cụ thể.

- Phản hồi: Chi tiết sản phẩm.



state.data = {

        id: 1000,

        thumbnail:

          "https://cdn.tgdd.vn/Products/Images/7618/286781/bhx/sellingpoint.jpg",

        name: "Xúc xích hồ lô C.P gói 500g",

        reg\_price: 106000.00,

        discount\_percent: 0,

        discount\_price: 106000.00,

        canonical: "500g",

        quantity: 28,

        rating: 4.4,

        description:

          "Xúc xích C.P với nguồn nguyên liệu này được chọn lọc kĩ càng, chế biến trên dây chuyền hiện đại. Xúc xích hồ lô C.P gói 500g là sản phẩm xúc xích có vị ngon tự nhiên, tiết kiệm tối đa thời gian nấu nướng mà vẫn có món ăn nhanh thơm ngon cho bữa ăn của mọi gia đình.",

        article: "",

        galleries: [

          {

            thumbnail:

              "https://cdn.tgdd.vn/Products/Images/7618/286781/bhx/sellingpoint.jpg",

            sort: 7488,

            product\_id: 1000,

          },

          {

            thumbnail:

              "https://cdn.tgdd.vn/Products/Images/7618/286781/bhx/xuc-xich-ho-lo-cp-goi-500g-202208190931272126.jpg",

            sort: 7489,

            product\_id: 1000,

          },

          {

            thumbnail:

              "https://cdn.tgdd.vn/Products/Images/7618/286781/bhx/xuc-xich-ho-lo-cp-goi-500g-202208190931275720.jpg",

            sort: 7490,

            product\_id: 1000,

          },

          {

            thumbnail:

              "https://cdn.tgdd.vn/Products/Images/7618/286781/bhx/xuc-xich-ho-lo-cp-goi-500g-202208190931278635.jpg",

            sort: 7491,

            product\_id: 1000,

          },

          {

            thumbnail:

              "https://cdn.tgdd.vn/Products/Images/7618/286781/bhx/xuc-xich-ho-lo-cp-goi-500g-202208190931281448.jpg",

            sort: 7492,

            product\_id: 1000,

          },

          {

            thumbnail:

              "https://cdn.tgdd.vn/Products/Images/7618/286781/bhx/xuc-xich-ho-lo-cp-goi-500g-202208190931386634.jpg",

            sort: 7493,

            product\_id: 1000,

          },

          {

            thumbnail:

              "https://cdn.tgdd.vn/Products/Images/7618/286781/bhx/xuc-xich-ho-lo-cp-goi-500g-202208190931389944.jpg",

            sort: 7494,

            product\_id: 1000,

          },

          {

            thumbnail:

              "https://cdn.tgdd.vn/Products/Images/7618/286781/bhx/xuc-xich-ho-lo-cp-goi-500g-202208190931392621.jpg",

            sort: 7495,

            product\_id: 1000,

          },

          {

            thumbnail: "https://i.ytimg.com/vi/LOFaFLSaBm8/sddefault.jpg",

            sort: 7496,

            product\_id: 1000,

          },

        ],

        brand: {

          id: 2,

          name: "C.P",

          thumbnail:

            "https://img.tgdd.vn/imgt/f\_webp,fit\_outside,quality\_100,s\_104x59/https://cdn.tgdd.vn/Brand/11/cp-14032021212030.jpg",

        },

        category: {

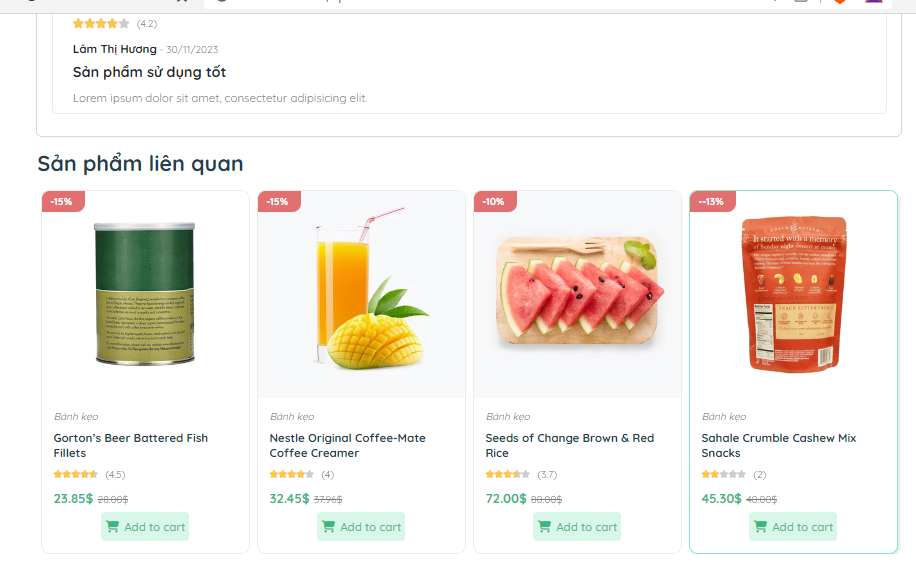
          id: 15,

          name: "Giày thể thao, sandals"

        }

      };

## Lấy danh sách sản phẩm liên quan FF



Đường dẫn: `**GET /api/sanpham/:id/lienquan**`

- Mô tả:lấy khaongr 4-5 cái

[

{

id: number;

  name: string;

  thumbnail: string;

  reg\_price: number;

  discount\_price: number;

  discount\_percent?: number,

  rating: number,

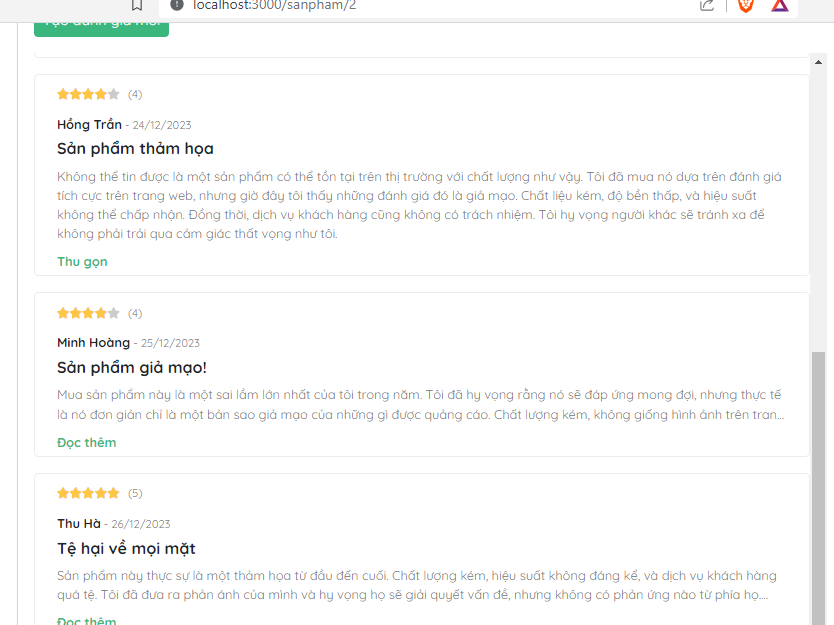
  category\_name: string

},

…

]

## Lấy Đánh giá của Sản phẩm



- Đường dẫn: `**GET /api/sanpham/:idsanpham/danhgi**a`

- Mô tả: Lấy danh sách đánh giá của một sản phẩm.

- Phản hồi: Mảng các đánh giá

state.data = [

        {

          id: 1,

          fullname: "Dũng CT",

          created\_at: "21/12/2023",

          rating: 4,

          title: "Chê, sản phẩm như \*\*\*",

          comment:

            "Tôi thật sự không thể giấu được sự thất vọng và tức giận của mình khi trải nghiệm sản phẩm này. Nếu có thể, tôi muốn đánh giá âm hơn một sao, vì sản phẩm này không đáng đồng tiền bát gạo. Đầu tiên, chất lượng của nó thật sự kém, từ vật liệu đến thiết kế, tất cả đều đánh mất sự chắc chắn và tinh tế. Sự giả mạo từ hình ảnh quảng cáo và thực tế là rõ ràng, khiến tôi cảm thấy như mình đã bị lừa dối.",

        },

        {

          id: 2,

          fullname: "Ngọc Anh",

          created\_at: "22/12/2023",

          rating: 3,

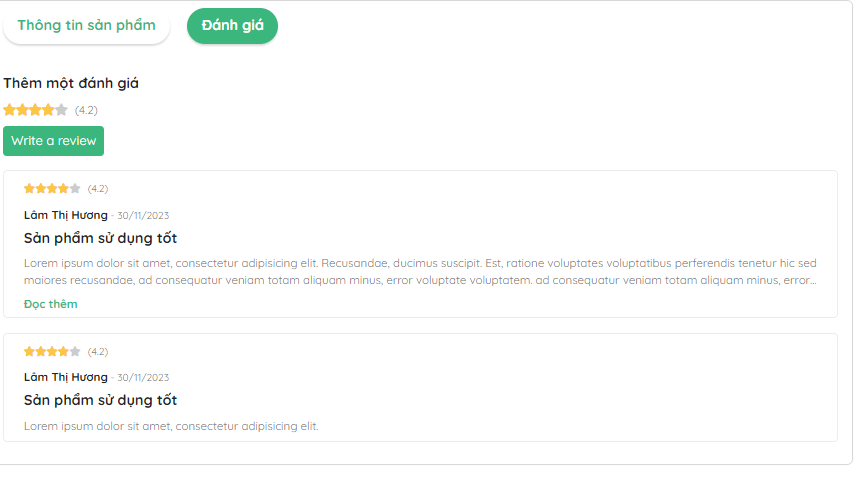
          title: "Gian lận về chất lượng",

          comment:

            "Sản phẩm này là một ví dụ điển hình về sự gian lận về chất lượng. Khi nhận được, tôi đã ngạc nhiên vì khác biệt lớn giữa hình ảnh trên trang web và sản phẩm thực tế. Vật liệu kém chất lượng và họa tiết in độc đáo đã biến thành một mớ lộn xộn không hài hòa. Ngoài ra, sản phẩm còn gặp vấn đề về hiệu suất, làm cho việc sử dụng trở nên khó khăn. Một sự mất lòng tin lớn đối với nhãn hiệu này!",

        },

## Thêm Đánh giá



- Đường dẫn: `**POST** /api/sanpham/themdanhgia` (thân json bên dưới)

- Mô tả: Thêm đánh giá cho một sản phẩm cụ thể.

- Phản hồi: Tất cả đánh giá của sản phẩm vừa thêm đánh giá mới.

// Thân yêu cầu thêm đánh giá

userId: string;

productId: string;

  rating: number;

  title: string;

  comment: string;

// Json trả về khi thành công, trả về danh sách đánh giá mới

state.data = [

        {

          id: 1,

          fullname: "Dũng CT",

          created\_at: "21/12/2023",

          rating: 4,

          title: "Chê, sản phẩm như \*\*\*",

          comment:

            "Tôi thật sự không thể giấu được sự thất vọng và tức giận của mình khi trải nghiệm sản phẩm này. Nếu có thể, tôi muốn đánh giá âm hơn một sao, vì sản phẩm này không đáng đồng tiền bát gạo. Đầu tiên, chất lượng của nó thật sự kém, từ vật liệu đến thiết kế, tất cả đều đánh mất sự chắc chắn và tinh tế. Sự giả mạo từ hình ảnh quảng cáo và thực tế là rõ ràng, khiến tôi cảm thấy như mình đã bị lừa dối.",

        },

        {

          id: 2,

          fullname: "Ngọc Anh",

          created\_at: "22/12/2023",

          rating: 3,

          title: "Gian lận về chất lượng",

          comment:

            "Sản phẩm này là một ví dụ điển hình về sự gian lận về chất lượng. Khi nhận được, tôi đã ngạc nhiên vì khác biệt lớn giữa hình ảnh trên trang web và sản phẩm thực tế. Vật liệu kém chất lượng và họa tiết in độc đáo đã biến thành một mớ lộn xộn không hài hòa. Ngoài ra, sản phẩm còn gặp vấn đề về hiệu suất, làm cho việc sử dụng trở nên khó khăn. Một sự mất lòng tin lớn đối với nhãn hiệu này!",

        },

// Json trả về khi thất bại

{

"status": "error",

"message": "Failed to add review",

"error": {

// details about the error, e.g., error message, code, etc.

}

}

# dTrang giỏ hàng / Pop up giỏ hàng

## Lấy thông tin Giỏ hàng

- Đường dẫn: `**GET** /api/giohang/:cartId`

- Mô tả: Lấy danh sách các mục trong giỏ hàng của người dùng.

- Phản hồi: Mảng các mục trong giỏ hàng.

```json

// Phản hồi với giỏ hàng

{

Quantity, total, savings,

list: [ // danh sach san pham

{

"cartItemId": 1, // cartItem id

productId: 3

thumbnail:

"name": "Sản phẩm 1",

"quantity": 2,

"reg\_price": 29.99,

discount\_price:

}

// hoặc rỗng

]

}

## Thêm vào Giỏ hàng

- Đường dẫn: `**POST** /api/giohang/them`

{ json ddi kem

productId:2

cartId: 3

quantity: 3,

}

- Mô tả: Thêm số lượng vào giỏ hàng (cộng dồn sl sản phẩm)

- Thân yêu cầu: ID sản phẩm, số lượng.

- Phản hồi: Giỏ hàng được cập nhật.

// Trả về thành công (nguyên cái giỏ hàng)

{

Quantity, total, savings,

list: [ // danh sach san pham

{

"cartItemId": 1, // cartItem id

productId: 3

thumbnail:

"name": "Sản phẩm 1",

"quantity": 2,

"reg\_price": 29.99,

discount\_price:

}

// hoặc rỗng

]

}

// Thất bại

{}

## Cập nhật Số lượng Mục Giỏ hàng

- Đường dẫn: `PUT /api/giohang/capnhat /{cartItemId}`

{

quantity: 3

}

- Mô tả: Cập nhật lại số lượng giỏ hàng, nếu cập nhật số lượng 0 thì tự động xoá cart\_item đó khỏi giỏ hàng

- Thân yêu cầu: Số lượng mới.

- Phản hồi: Giỏ hàng được cập nhật.

// Trả về thành công (nguyên cái giỏ hàng)

{

Quantity, total, savings,

list: [ // danh sach san pham

{

"cartItemId": 1, // cartItem id

productId: 3

thumbnail:

"name": "Sản phẩm 1",

"quantity": 2,

"reg\_price": 29.99,

discount\_price:

}

// hoặc rỗng

]

}

// Thất bại

{}

## Xoá toàn bộ giỏ hàng

- Đường dẫn: `DELETE /api/giohang/{id gio hang}/xoa `

- Mô tả: Xóa toàn bộ giỏ hàng (sau khi nhấn đặt hàng thì sẽ xoá toàn bộ sản phẩm)

```json

**Truyền vào id**

// Phản hồi với giỏ hàng được cập nhật sau khi xóa mục

{

Quantity, total, savings,

list: [ ] // rỗng

}

```

# Đăng Nhập / Đăng ký

## Đăng nhập

- Đường dẫn: `**POST /api/dangnhap**`

// Thân yêu cầu đăng nhập người dùng

{

"email": "nguoidungmoi@example.com",

"password": "matkhau123"

}

* Sau khi đăng nhập thành công, mọi request tiếp theo của user sẽ bao gồm **JWT token để xác thực danh tính**. Backend sẽ kiểm tra token này trước khi xử lý các request cần đăng nhập.

**// Phản hồi với token JWT thành công**

{

id: number,

    cartId:number,

    email:”@gmail.com”,

    fullname:”Nguyen van a”,

    phone:”03823352”,

    address: ”113 Tran Hung Dao”,

    isLoggedIn: true // chỗ này luôn luôn trả về true

}

// Fail

{

"error": "Sai tên đăng nhập hoặc mật khẩu"

}

## Đăng ký

- Đường dẫn: `**POST /api /dangky** `

{

"email": "nguoidungmoi@example.com",

"password": "matkhau123"

fullname:”Lý Quí Trung”

}

- Phản hồi: Đối tượng người dùng với định danh duy nhất.

{

id: number,

    cartId:number,

    email:”@gmail.com”,

    fullname:”Nguyen van a”,

    phone:””, // các thông tin rỗng là khi user mới đăng ký, sau này cập nhật sau

    address: ””,

    isLoggedIn: true // chỗ này luôn luôn trả về true

}

// Lỗi

{

"error": "Email đã tồn tại" // cái này bắt buộc trả về cho t nhe

}

## Đăng xuất

- Đường dẫn: **Hiện thì không cần API đăng xuất (do không sử dụng cơ chế token)**

- Mô tả: Đăng xuất người dùng, hủy token JWT.

- Thân yêu cầu: Không cần thân yêu cầu.

- Phản hồi: Thông báo xác nhận đăng xuất.

## Check cookie / Lấy thông tin người dùng

- Đường dẫn: `**GET /api/taikhoan/:userId/thongtin**`

- Mô tả: Cơ chế hiện tại của đăng nhập và đăng ký là sau khi login, thay vì lưu token (không có thời gian chơi cái trò jwt này) nên sẽ lưu userId thay thế vào cookie, khi người dùng thoát ra và vào lại trang web thì sẽ gửi 1 api chứa userId này để check xem là có người dùng đó không, nếu có thì trả về thông tin người dùng đồng thời không cần đăng nhập lại.

```json

// Phản hồi với chi tiết hồ sơ người dùng

{

id: number,

    cartId:number,

    email:”@gmail.com”,

    fullname:”Nguyen van a”,

    phone:”03823352”,

    address: ”113 Tran Hung Dao”,

    isLoggedIn: true // chỗ này luôn luôn trả về true

}

```

# Trang thông tin người dùng sau khi đăng nhập (/taikhoan)

## Cập nhật Hồ sơ Người dùng

- Đường dẫn: `**PUT /api/taikhoan/:userId/capnhat** `

{

    email:”@gmail.com”,

    fullname:”Nguyen van a”,

    phone:”03823352”,

    address: ”113 Tran Hung Dao”,

}

- Mô tả: Cập nhật thông tin hồ sơ của người dùng.

Trả về toàn bộ thông tin như phần dang nhpa ,dang ky

{

id: number,

    cartId:number,

    email:”@gmail.com”,

    fullname:”Nguyen van a”,

    phone:”03823352”,

    address: ”113 Tran Hung Dao”,

    isLoggedIn: true // chỗ này luôn luôn trả về true

}

```

## Đặt Đơn hàng

**Phần này nằm trong phần xác nhận đặt hàng => Thêm một hoá đơn vào db**

**API NÀY CALL KHI USER NHẤN XÁC NHẬN TẠO HOÁ ĐƠN, KHI GỌI ĐẾN API THÌ TẠO HOÁ ĐƠN VÀ XOÁ LUN CÁC CART\_ITEM (backend xử lý)**

- Đường dẫn: `**POST /api/donhang/them** `

{

userId: 3,

address: “So 113 Tran Duy Hung, Long Thanh, Ha Noi”

payment\_method: “truc tiep”,

note: “Gui cho t size 38”

}

- Mô tả: Thêm một đơn hàng sau khi khách hàng chọn đặt hàng

- Thân yêu cầu: userId, address, payment\_method, note.

- Phản hồi: Trả về id đơn hàng vừa tạo

```json

// Thành công thì trả về mã hoá đơn để t send user tới trang /taikhoan/hoadon/id

{

id: number

}

```

// Thất bại

{

Message: “Thất bại khi đặt hàng”

}

## Lấy danh sách hoá đơn ~ lịch sử mua hàng

- Đường dẫn: `**GET /api/taikhoan/:userId/donhang** `

- Phản hồi: trả về một mảng thông tin hoá đơn

Lưu ý là trả về danh sách theo thứ tự thời gian giảm dần ví dụ arr[0] là đơn hàng 28/12, arr[1] là đơn hàng 27/12

```json

// Thành công

[

{

id: number,

  status: string, // Đang giao / Đã giao

  total: number,

date: string, // thời gian đặt hàng, trả về string kiểu “3:42pm 28/12/2023”

  thumbnail: string, // thumbnail của sản phẩm bất kì hoặc sản phầm đầu tiên đều đc

},

…

]

```

// Thất bại

{

Error: “Ghi gì cũng đc”

}

## Lấy thông tin đơn hàng

Đường dẫn: `**GET /api/donhang/:donhangId**`

- Phản hồi: trả về một mảng thông tin hoá đơn

// Thành công

{

id: number, // mã don hang

  address: string,

  total: number,

  note: string,

  status: string, // Đang giao / Đã giao

  payment\_method: string, // hiện tại mặc định luôn là COD

  date: string, // ngày hoá đơn, string có dạng "3:42pm 27/12/2023"

  list: { // trả về danh sách sản phẩm thuojc đơn hàng trong đó mỗi sản phẩm:

    itemId: number, // mã item của đơn hàng

    productId: number;

    name: string,

    thumbnail: string,

    quantity: number,

    total: number,

unit\_price: number //triệu: tao có thêm này trong response, này là giá của một sản phẩm tại thời điểm đơn hàng được tạo, tại vì trong tương lai giá sp đó có thể thay đổi.

  }[]

}

// Thất bại

{

Error: “Ghi gì cũng đc”

}

# Danh sách yêu thích

## Lấy danh sách yêu thích

Đường dẫn: `**GET /api/taikhoan/:userid/yeuthich** `

// Phản hồi với danh sách sản phẩm yêu thích

 [

    {

        id: 1, //id này là id của product để khi nhấn vào thì có thể vào trang chi tiết sản phẩm

        thumbnail: "url",

        name: "ten san pham",

        discount\_price: 23000,

    },

    ...

]

```

## Thêm sản phẩm vào danh sách yêu thích

Đường dẫn: `**POST /api/sanpham/yeuthich**`

{ json đi kèm

productId: 1,

userId: 2

}

Phản hồi:

// Phản hồi với danh sách sản phẩm yêu thích mới

 [

    {

        id: 1,

        thumbnail: "url",

        name: "ten san pham",

        discount\_price: 23000,

    },

    ...

]

```

## Xoá khỏi wishlist

Đường dẫn: `**POST /api/sanpham/yeuthich/xoa**

{ json đi kèm

productId,

userId,

}

Phản hồi:

// Phản hồi với danh sách sản phẩm yêu thích mới

 [

    {

        id: 1,

        thumbnail: "url",

        name: "ten san pham",

        discount\_price: 23000,

    },

    ...

]

```

# Các mục bổ sung (nếu còn thời gian sẽ thêm vào)

## Thanh toán:

### Xác nhận Thanh toán

- Đường dẫn: `POST /api/orders/{orderId}/confirm-payment`

- Mô tả: Xác nhận thanh toán cho một đơn hàng đã đặt.

- Thân yêu cầu: Thông tin thanh toán bổ sung (nếu cần).

- Phản hồi: Thông báo xác nhận thanh toán.

```json

// Thân yêu cầu xác nhận thanh toán

{

"paymentMethod": "credit\_card",

"cardNumber": "\*\*\*\* \*\*\*\* \*\*\*\* 5678",

"expiryDate": "12/25",

"cvv": "456"

}

// Phản hồi với thông báo xác nhận thanh toán

{

"message": "Thanh toán cho đơn hàng #789012 đã được xác nhận."

}

```

### 17. \*\*Hủy Đơn hàng:\*\*

- Đường dẫn: `DELETE /api/orders/{orderId}/cancel`

- Mô tả: Hủy một đơn hàng chưa được xác nhận.

- Thân yêu cầu: Không cần thân yêu cầu.

- Phản hồi: Thông báo xác nhận hủy đơn hàng.

```json

// Phản hồi với thông báo xác nhận hủy đơn hàng

{

"message": "Đơn hàng #789012 đã được hủy thành công."

}

```