

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO CUỐI KÌ
CHUYÊN ĐỀ CÔNG NGHỆ PHẦN MỀM**

**XÂY DỰNG WEBSITE BÁN HÀNG LINH
KIỆN MÁY TÍNH ASP.NET MVC WEB
API VÀ REACTJS**

Người hướng dẫn: **GV DOÃN XUÂN THANH**

Người thực hiện: **NGÔ MINH HIẾU – 51702017**

Lớp : 17050201

Khoá : 21

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2021

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO CUỐI KÌ
CHUYÊN ĐỀ CÔNG NGHỆ PHẦN MỀM**

**XÂY DỰNG WEBSITE BÁN HÀNG LINH
KIỆN MÁY TÍNH ASP.NET MVC WEB
API VÀ REACTJS**

Người hướng dẫn: GV DOÃN XUÂN THANH

Người thực hiện: NGÔ MINH HIẾU

Lớp : 17050201

Khoá : 21

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2021

ĐỒ ÁN ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Tôi xin cam đoan đây là sản phẩm đồ án của riêng tôi và được sự hướng dẫn của GV Doãn Xuân Thanh ;. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình. Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày 16 tháng 11 năm 2021

Tác giả

(ký tên và ghi rõ họ tên)

Hieu

Ngô Minh Hiếu

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

Phần xác nhận của GV hướng dẫn

Tp. Hồ Chí Minh, ngày tháng năm
(kí và ghi họ tên)

Phần đánh giá của GV chấm bài

Tp. Hồ Chí Minh, ngày tháng năm
(kí và ghi họ tên)

MỤC LỤC

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN	2
MỤC LỤC.....	3
CHƯƠNG 1 – GIỚI THIỆU	6
1. GIỚI THIỆU ĐỀ TÀI	6
2. CỔ SỞ LÝ THUYẾT.....	7
a) WEB API	7
b) REACTJS.....	7
c) SQL SERVER.....	8
3. PHÂN TÍCH THIẾT KẾ	8
CHƯƠNG 2 – HIỆN THỰC.....	10
1. XÂY DỰNG WEB API VÀ CÁCH SỬ DỤNG API	10
2. MỘT SỐ ĐOẠN CODE CẦN GIẢI THÍCH.....	25
a) WEB API	25
b) REACT JS.....	27
3. XÂY DỰNG VỚI REACJS.....	28
4. KẾT LUẬN.....	44

Hình 1 : Sơ Đồ Use Case	8
Hình 2 : Sơ đồ cơ sở dữ liệu.....	9
Hình 6 : API Lấy tất cả sản phẩm	10
Hình 7 : API Lấy chi tiết sản phẩm.....	11
Hình 8 : API Lấy chi tiết sản phẩm không tồn tại.....	11
Hình 9 : API dùng để đăng nhập	12
Hình 10 : API dùng để đăng kí.....	13
Hình 11 : API dùng để thêm sản phẩm	14
Hình 12 : API trả kết quả thêm sản phẩm thất bại	15
Hình 13 : API dùng để xóa sản phẩm	15
Hình 14 : API trả kết quả việc xóa sản phẩm.....	16
Hình 15 : API để sửa đổi thông tin sản phẩm	17
Hình 16 : API dùng thay đổi thông tin tài khoản	18
Hình 17 : API thay đổi mật khẩu của tài khoản	19
Hình 18 : API lấy sản phẩm theo ID đơn hàng	20
Hình 19 : API lấy tất cả đơn hàng theo ID của user	21
Hình 20 : API dùng để thêm sản phẩm vào đơn hàng	22
Hình 21 : API dùng để xóa đơn hàng.....	23
Hình 22 : API dùng để xóa sản phẩm trong đơn hàng.....	24
Hình 23: Hình ảnh đoạn code cho phép web api sử dụng jwt	25
Hình 24 : Hình ảnh đoạn code mở CORS cho web api	26
Hình 25 : Hình ảnh đoạn code tạo ra kết quả trả token.....	26
Hình 26 : Hình ảnh đoạn code giúp lưu ảnh ở 1 side online.....	27
Hình 27 : Giao diện trang chủ phần thanh thông báo	28
Hình 28 : Giao diện trang chủ khi kéo xuống.....	28
Hình 29 : Hình ảnh tượng trưng cho một số nhãn hàng.....	29

Hình 30 : Hình ảnh cuối trang web	29
Hình 31 : Hình ảnh trang web sử dụng ở màn hình điện thoại	30
Hình 32 : Hình ảnh chi tiết sản phẩm.....	31
Hình 33 : Hình ảnh sản phẩm cùng loại trong chi tiết sản phẩm	31
Hình 34 : Hình ảnh trang chi tiết sản phẩm sử dụng ở điện thoại.....	32
Hình 35 : Hình ảnh các sản phẩm khác cùng loại khi hiện thị ở thiết bị điện thoại	33
Hình 36 : Hình ảnh chi tiết hóa đơn hiện tại	34
Hình 37 : Hình ảnh việc in hóa đơn của khách hàng	35
Hình 38 : Hình ảnh hóa đơn khi sử dụng với thiết bị điện thoại.....	36
Hình 39 : Hình ảnh tất cả hóa đơn mà khách từng đặt hàng.....	37
Hình 40 : Hình ảnh tất cả hóa đơn khi sử dụng ở thiết bị điện thoại	38
Hình 41 : Hình ảnh trang thông tin của tài khoản.....	39
Hình 42 : Hình ảnh trang giao diện của người quản lý	39
Hình 43 : Hình ảnh quản lý sản phẩm và người dùng	40
Hình 44 : Hình ảnh trang giao diện quản trị khi sử dụng ở thiết bị điện thoại	41
Hình 45 : Hình ảnh trang giao diện quản trị khi sử dụng ở thiết bị điện thoại	42
Hình 46 : Hình ảnh khi người quản trị muốn thêm sản phẩm.....	43
Hình 47 : Hình ảnh người quản trị muốn thay đổi thông tin sản phẩm	43

CHƯƠNG 1 – GIỚI THIỆU

1. GIỚI THIỆU ĐỀ TÀI

- Đề tài sẽ xây dựng một website bán hàng các sản phẩm linh kiện máy tính cũng cấp các sản phẩm như là bàn phím , chuột , tai nghe , lót chuột có chất lượng vừa dùng để sử dụng vừa có thể trang trí bàn máy tính của bản thân
- Vì cảm thấy đề tài xây dựng website với ASP.NET MVC WEB API (Back-End) + REACTJS (Front-End).
- Có thể bản thân làm được và với sở thích các sản phẩm công nghệ nên chọn đề tài này để áp dụng các kiến thức học được về ASP.NET MVC WEB API + REACTJS để hiện thực đề tài
- Một số yêu cầu của đề tài cần thực hiện
 - o Xây dựng giao diện của trang web với ReactJS các chức năng:
 - Thêm, sửa, xóa, lấy sản phẩm
 - Sửa, lấy thông tin tài khoản
 - Thêm, sửa, xóa, lấy thông tin hóa đơn
 - Thay đổi mật khẩu tài khoản
 - Đăng nhập , đăng kí tài khoản người dùng
 - Phân quyền admin , người dùng khi đăng nhập với website
 - Hiện thị website ở điện thoại , máy tính
 - Tự động hiện thị danh sách sản phẩm
 - Khi xem sản phẩm sẽ hiện thị các sản phẩm cùng loại
 - Thực hiện xuất hóa đơn thành file pdf và có thể in hóa đơn
 - o Xây dựng các chức năng với ASP.NET MCV WEB API:
 - Thêm, sửa, xóa, lấy sản phẩm
 - Sửa, lấy thông tin tài khoản
 - Thêm, sửa, xóa, lấy thông tin hóa đơn
 - Thay đổi mật khẩu tài khoản

- Đăng nhập , đăng kí tài khoản người dùng
- Phân quyền admin , người dùng khi đăng nhập với website

2. CỞ SỞ LÝ THUYẾT

a) *WEB API*

- Web API được sử dụng trong các hệ thống website. Có một số website đều được thiết kế theo dạng web API cho phép người lập trình viên hiểu và có thể sử dụng để gửi yêu cầu đến web API đó để có thể lấy dữ liệu phù hợp với yêu cầu request mà web API đó đặt ra. Ví dụ như ứng dụng của bạn được viết ra với các yêu cầu từ khách hàng là có thể sử dụng tài khoản có sẵn thuộc một số trang mạng xã hội, trang công nghệ lớn như facebook , google , gmail, ... yêu cầu người lập trình viên phải đọc cách giao tiếp với các API trên để cấp quyền cũng như đảm bảo API của bạn gọi đến API của họ một cách hợp lệ.
- Hệ thống API được thiết kế tuân theo tiêu chuẩn REST và HTTP như các phương thức sau
 - GET : được dùng để lấy dữ liệu từ API
 - POST : được dùng để đưa dữ liệu đến API
 - PUT : được dùng để đưa dữ liệu đến API và thường được dùng cập nhật lại dữ liệu đã (quy ước chung theo tiêu chuẩn REST)
 - DELETE: như tên của phương thức thì phương thức này sử dụng để gửi yêu cầu đến API xóa một thông tin nào đó có trong API khi bạn có quyền được thực hiện việc xóa dữ liệu trên API đó

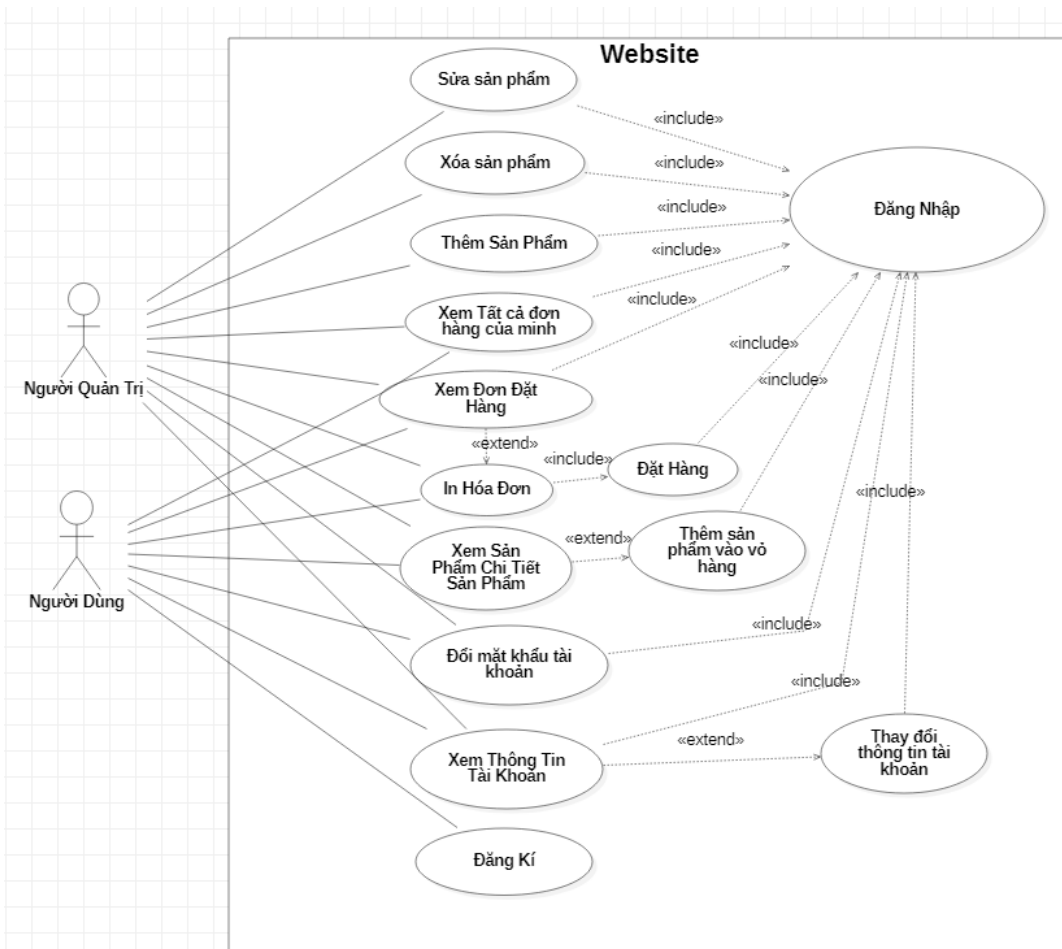
b) *REACTJS*

- React là thư viện được phát triển bởi facebook dùng để xây dựng giao diện người dùng được viết bằng javascript. React được sử dụng phổ biến và có lượng thư viện hỗ trợ trên NPM rất lớn để các lập trình viên front-end thỏa sức sáng tạo để tạo ra các trang web có cảm giác người dùng , giao diện người dùng thân thiện và dễ sử dụng cho người dùng

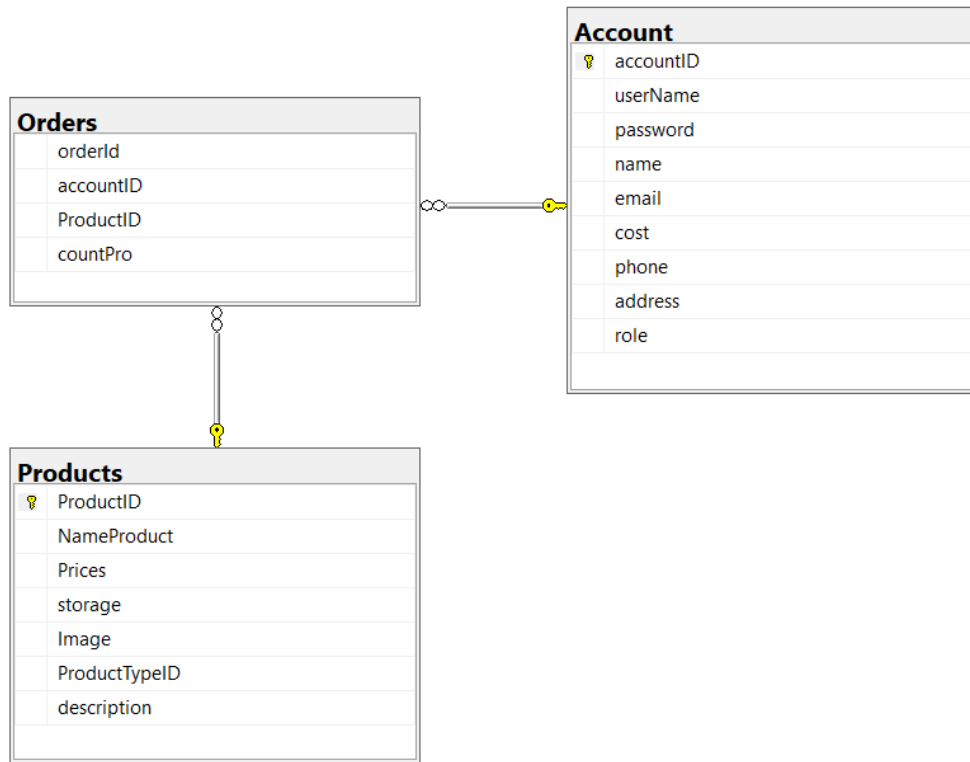
c) *SQL SERVER*

- SQL server hay còn gọi mà Microsoft SQL hay MSSQL là sản phẩm được phát triển bởi Microsoft và được sử dụng dễ dàng để lưu trữ cho dữ liệu được quy định ở dạng RDBMS là hệ quản trị cơ sở dữ liệu theo dạng quan hệ đối tượng
- SQL server là một phần mềm giúp ta giao tiếp với cơ sở dữ liệu thông qua ngôn ngữ SQL

3. PHÂN TÍCH THIẾT KẾ



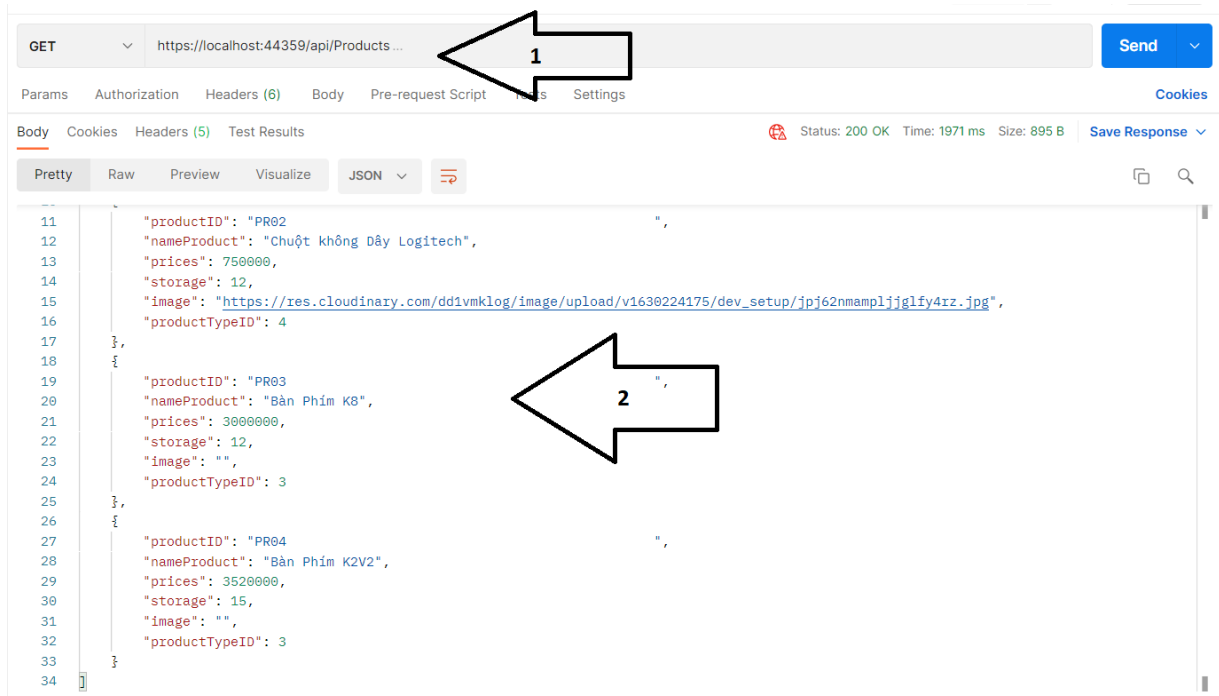
Hình 1 : Sơ Đồ Use Case



Hình 2 : Sơ đồ cơ sở dữ liệu

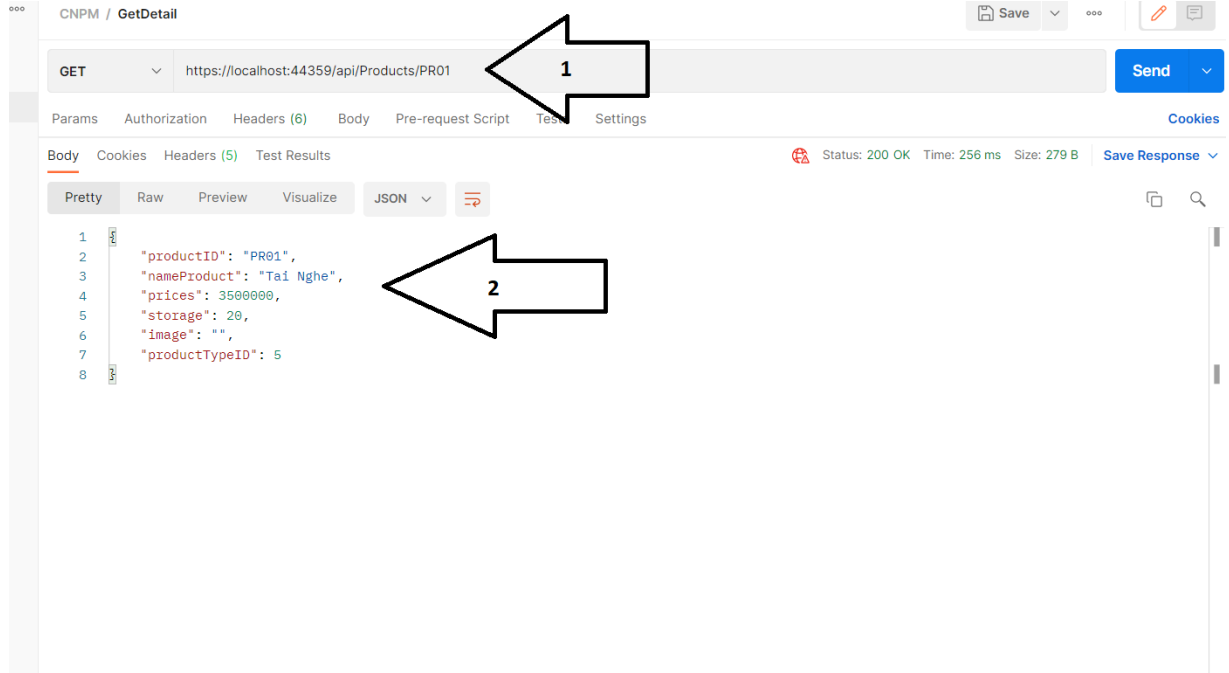
CHƯƠNG 2 – HIỆN THỰC

1. XÂY DỰNG WEB API VÀ CÁCH SỬ DỤNG API



Hình 3 : API Lấy tất cả sản phẩm

- 1 : Đường dẫn tới api : <https://localhost:44359/api/Products>
- 1 : Phương thức được sử dụng là GET
- 2 : Chức năng được dùng để lấy tất cả sản phẩm có trong cơ sở dữ liệu



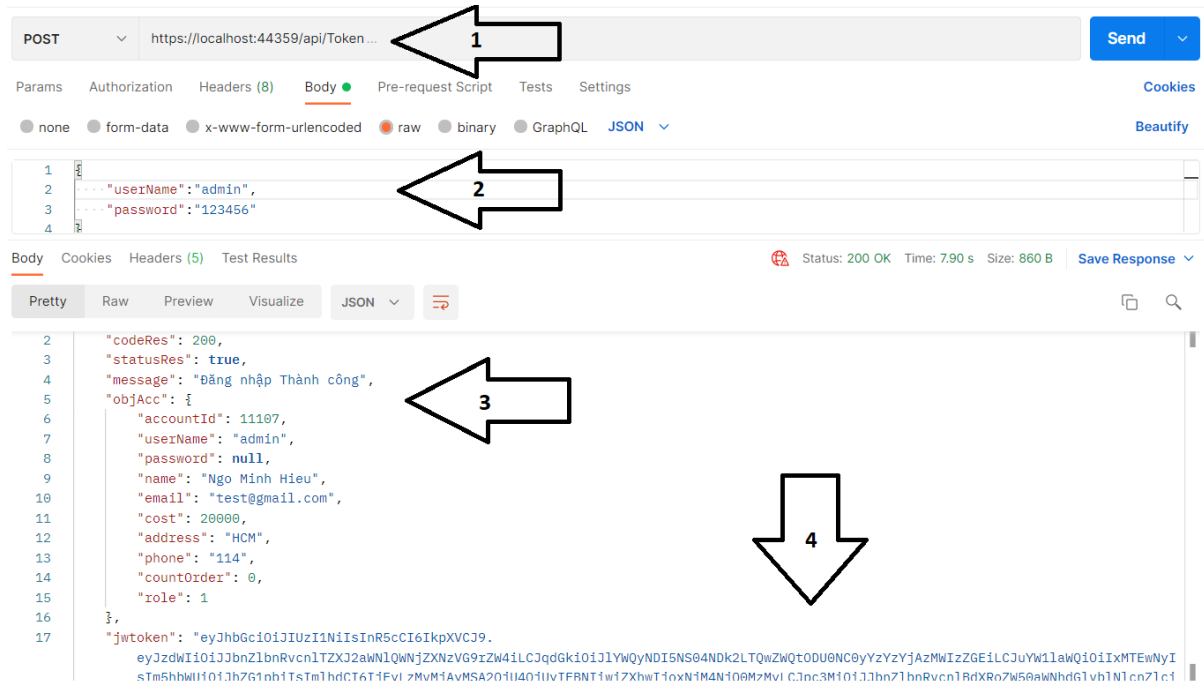
Hình 4 : API Lấy chi tiết sản phẩm

- 1 : Đường dẫn tới api : <https://localhost:44359/api/Products/{idproduct}>
- 1 : Phương thức được sử dụng là phương thức GET, dùng để lấy chi tiết sản phẩm
- 2 : Là dữ liệu được API trả về.



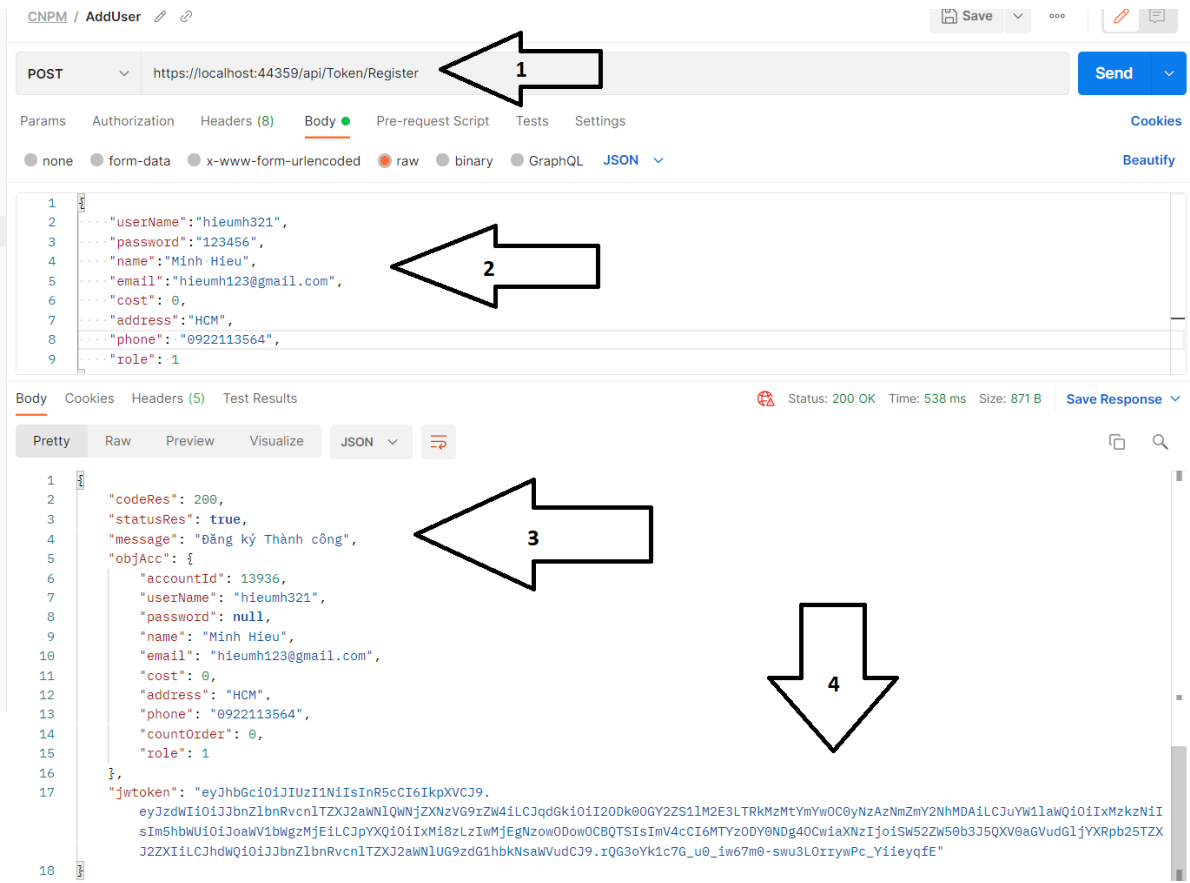
Hình 5 : API Lấy chi tiết sản phẩm không tồn tại

- Nếu id product (1) không nằm trong cơ sở dữ liệu thì phía web API sẽ trả về dữ liệu có dạng ProductId là dạng rỗng (2).



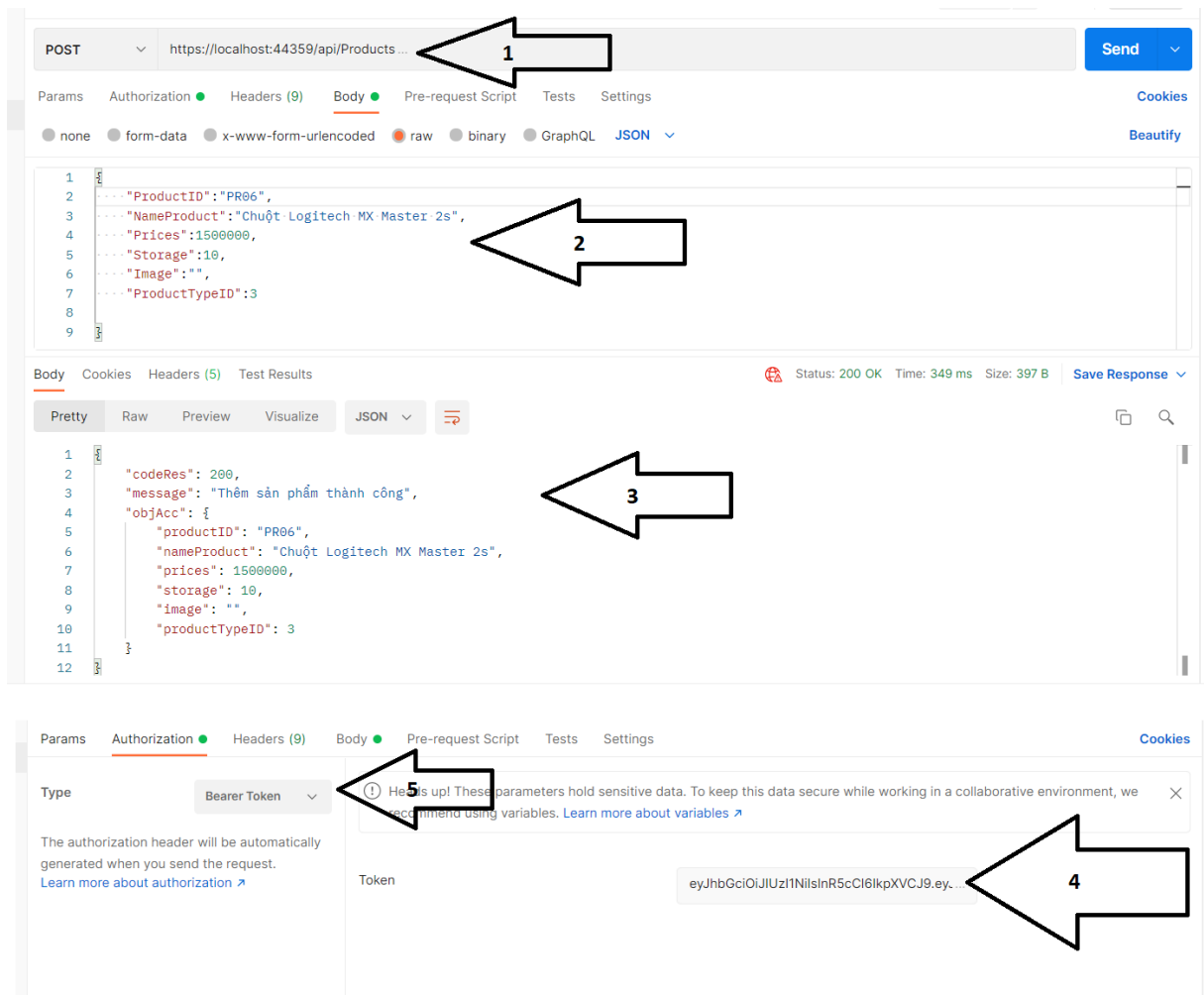
Hình 6 : API dùng để đăng nhập

- 1 : Đường dẫn tới api : <https://localhost:44359/api/Token>
- 1 : Phương thức được sử dụng gọi đến đường dẫn này là POST
- 2 : Thông tin truyền theo gồm userName , password
- 3 : Kết quả API trả về là thông tin tài khoản
- 4: Là JWT được đính kèm dùng để thao tác với các tác vụ khác
- Được dùng để đăng nhập.



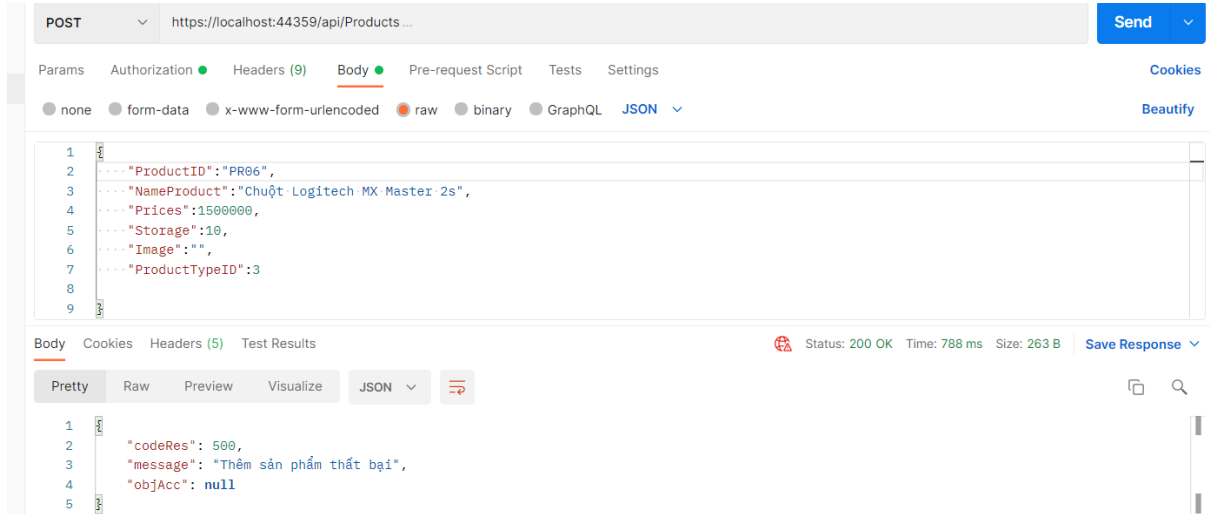
Hình 7 : API dùng để đăng kí

- 1 : Đường dẫn tới api : <https://localhost:44359/api/Token/Register>
- 1 : Phương thức được sử dụng gọi đến đường dẫn này là POST
- 2 : Thông tin truyền theo gồm userName, password,name,email,address,phone và role sẽ được định nghĩa ở front-end gắn cứng ở form để phân biệt quyền admin
- 3 : Kết quả API trả về là thông tin tài khoản đã đăng kí thành công
- 4: Là JWT được đính kèm dùng để thao tác với các tác vụ khác
- Được dùng để đăng nhập.



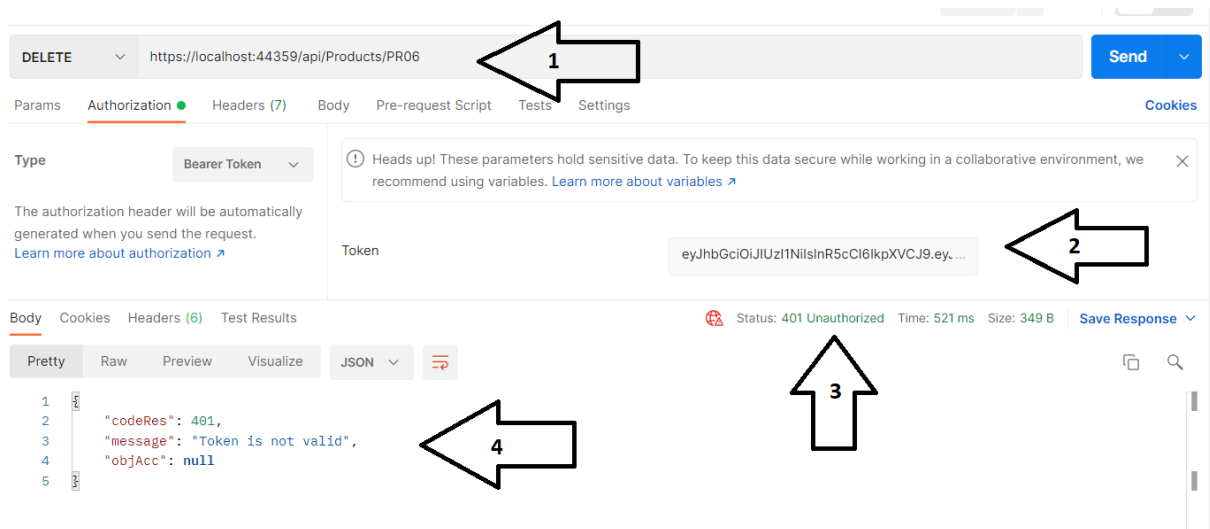
Hình 8 : API dùng để thêm sản phẩm

- 1 : Đường dẫn tới api : <https://localhost:44359/api/Products>
- 1 : Phương thức sử dụng ở đây là Post, được dùng để thêm sản phẩm.
- 2 : Thông tin sản phẩm productid, nameProduct, prices, storage, image, productid.
- 3 : Thông tin được trả về từ web API
- 4,5 : Sau khi đăng nhập ta có jwt từ đó ta gán cùng theo request với bearer token gửi đến server web api.



Hình 9 : API trả kết quả thêm sản phẩm thất bại

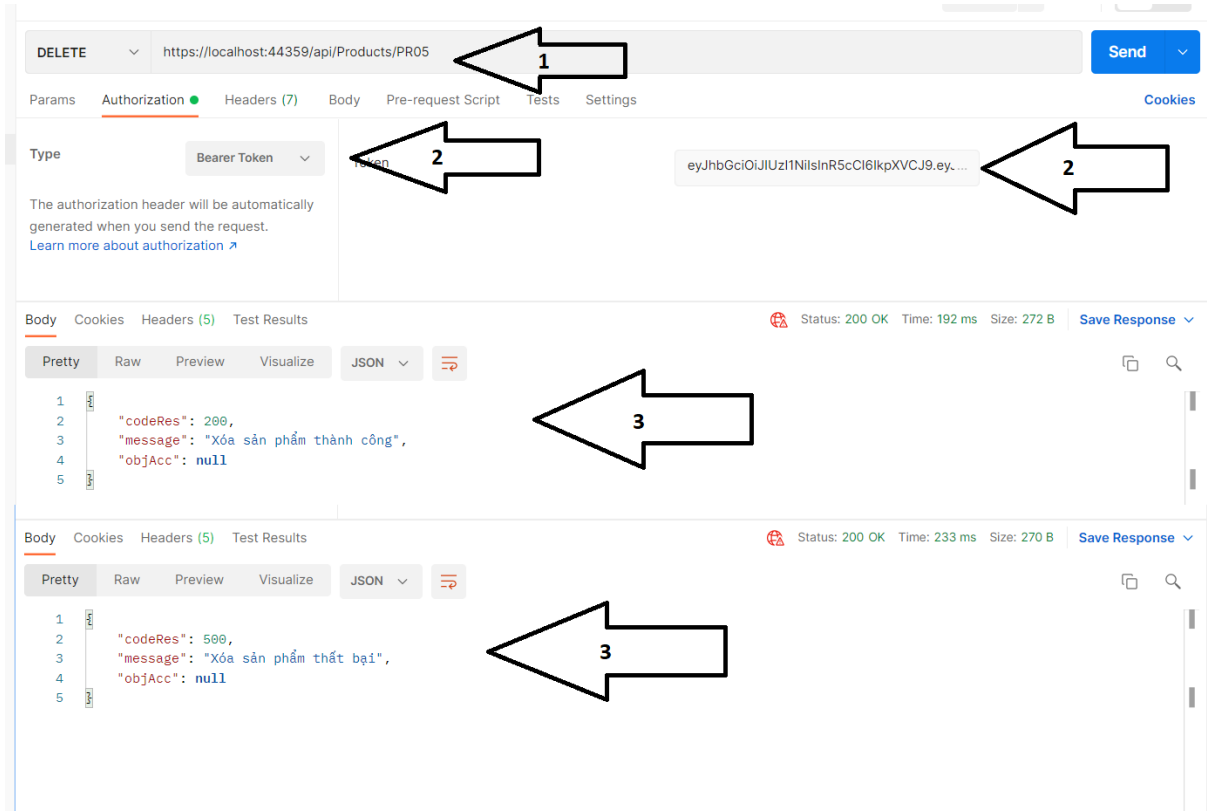
- Nếu ta tiến hành thêm 1 sản phẩm y chang lần trước thì server trả về thêm sản phẩm thất bại vì ProductId là duy nhất và không được trùng.



Hình 10 : API dùng để xóa sản phẩm

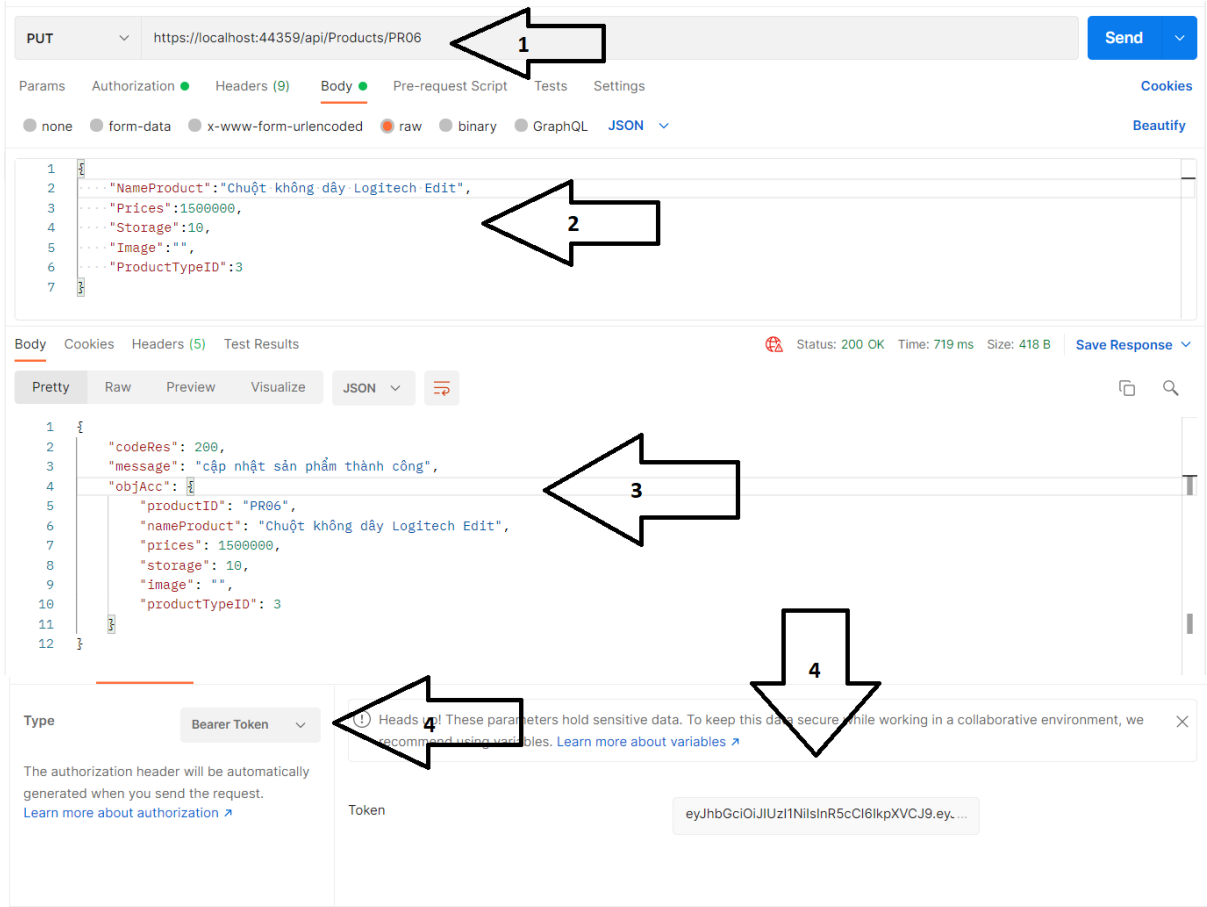
- 1 : Đường dẫn tới api : <https://localhost:44359/api/Products/{idproduct}>
- 1 : Phương thức được sử dụng là phương thức Delete được dùng để xóa sản phẩm
- 2 : Đính kèm jwt sau khi đăng nhập vào bearer Token
- 3 : Code mặc định của web API trả về qua là 401 unauthorzied

- 4 : dữ liệu được web API trả về dưới dạng json để tiện xử lý ở phía client với các thông tin là codeRes, message , objAcc(tùy biến theo một số phương thức)



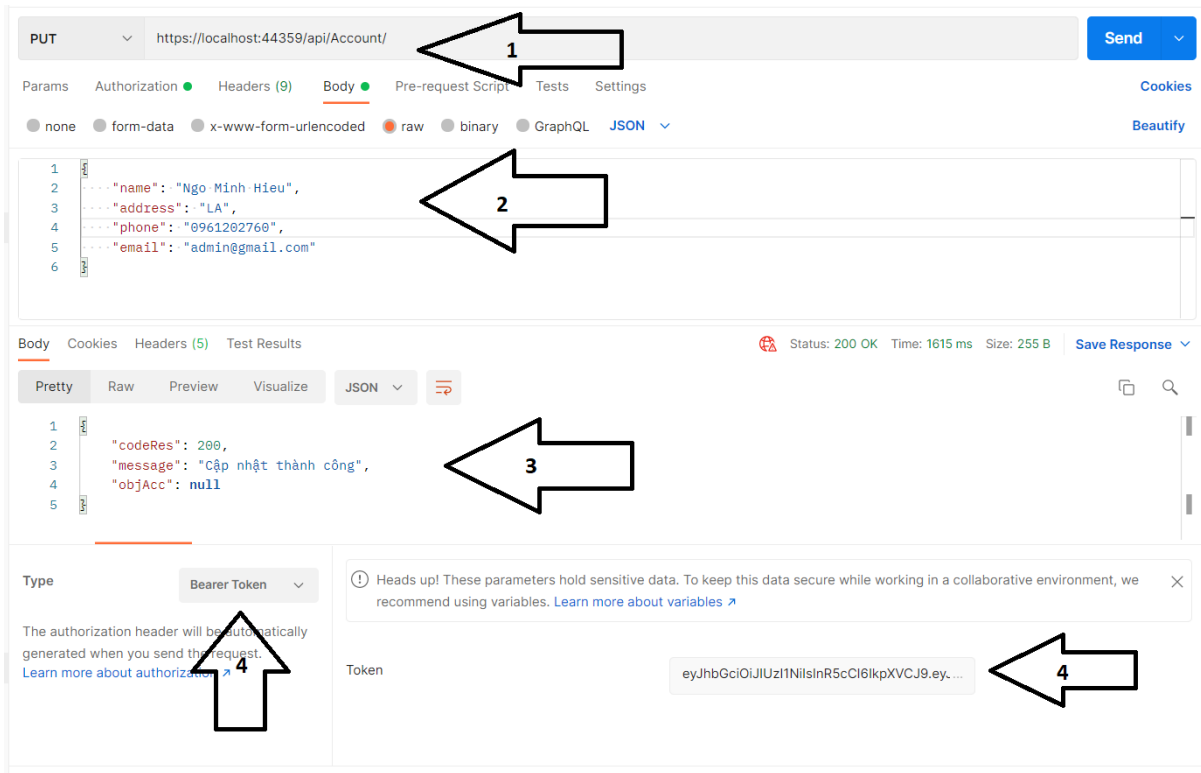
Hình 11 : API trả kết quả việc xóa sản phẩm

- Nếu kèm theo token hợp lệ thì thực hiện yêu cầu xóa thành công API sẽ trả về dữ liệu như hình và kết quả là thực hiện xóa sản phẩm thành công .
Nếu thực hiện việc xóa với cùng một mã sản phẩm thì sẽ báo lỗi là xóa thất bại vì đã xóa mất sản phẩm có mã như trước



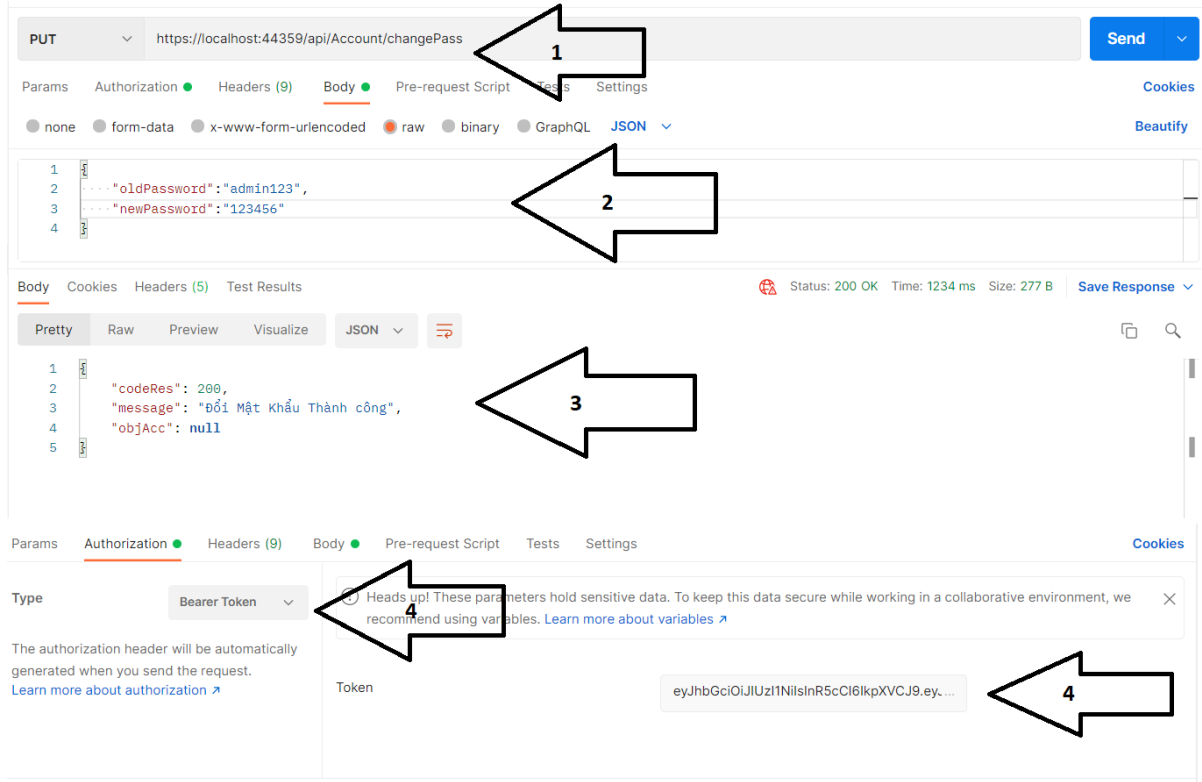
Hình 12 : API để sửa đổi thông tin sản phẩm

- 1 : Đường dẫn tới api : <https://localhost:44359/api/Products/{idproduct}>
- 1 : Phương thức được sử dụng là phương thức PUT
- 2 : Thông tin được gửi đi kèm gồm nameProduct, prices, storage, image, productid.
- 3 : Thông tin được trả về nếu cập nhật thành công
- 4 : Đính kèm jwt vào bearer token để request gửi lên web api có thể thực hiện được thao tác cập nhật



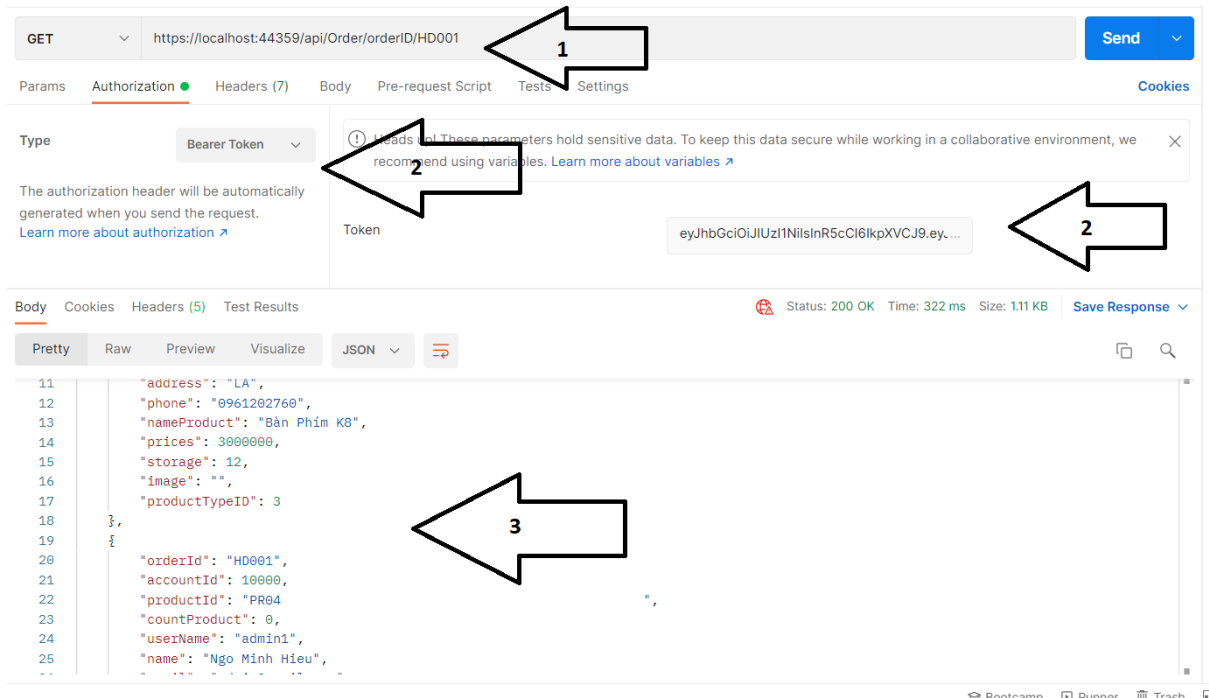
Hình 13 : API dùng thay đổi thông tin tài khoản

- 1 : Đường dẫn tới api : <https://localhost:44359/api/Account>
- 1 : Phương thức được sử dụng là phương thức PUT
- 2 : Thông tin được gửi đi kèm gồm name, address , phone, email.
- 3 : Thông tin được trả về nếu cập nhật thành công
- 4 : đính jwt vào bearer token để lấy được userid trong jwt.



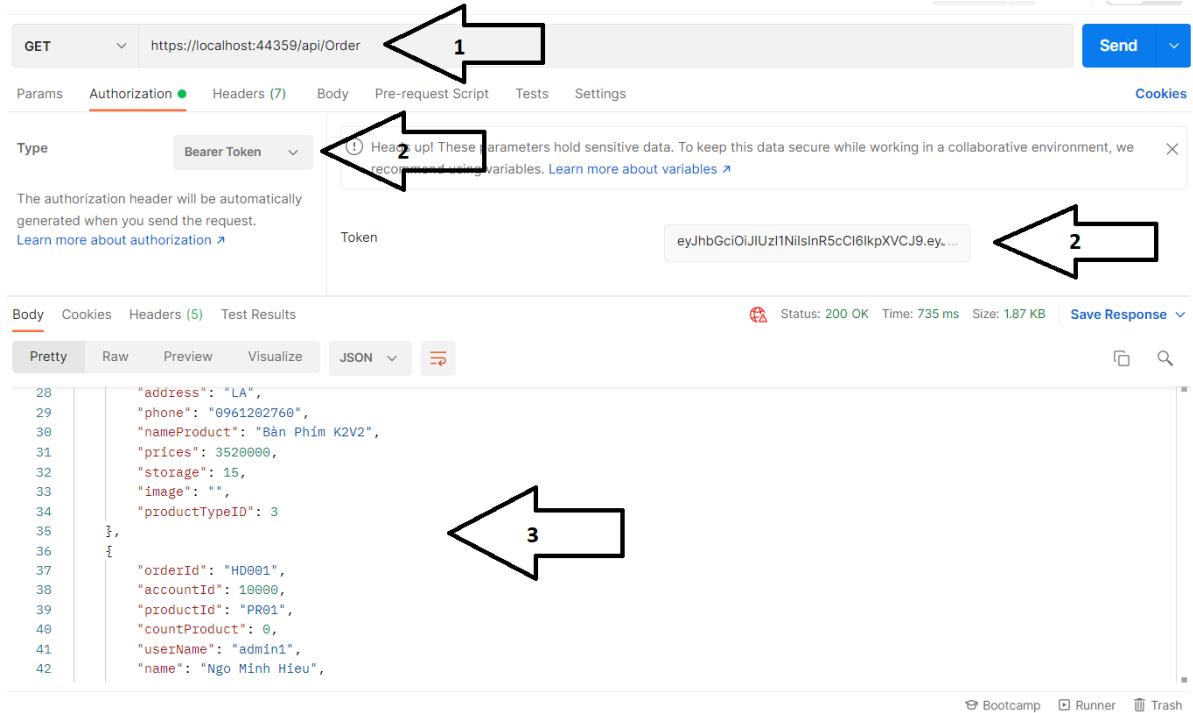
Hình 14 : API thay đổi mật khẩu của tài khoản

- 1 : Đường dẫn tới api : <https://localhost:44359/api/Account/changePass>
- 1 : Phương thức được sử dụng là phương thức PUT
- 2 : Thông tin được gửi đi kèm gồm oldPassword,newPassword
- 3 : Thông tin được trả về nếu thay đổi mật khẩu thành công
- 4 : đính jwt vào bearer token để lấy được userid trong jwt.



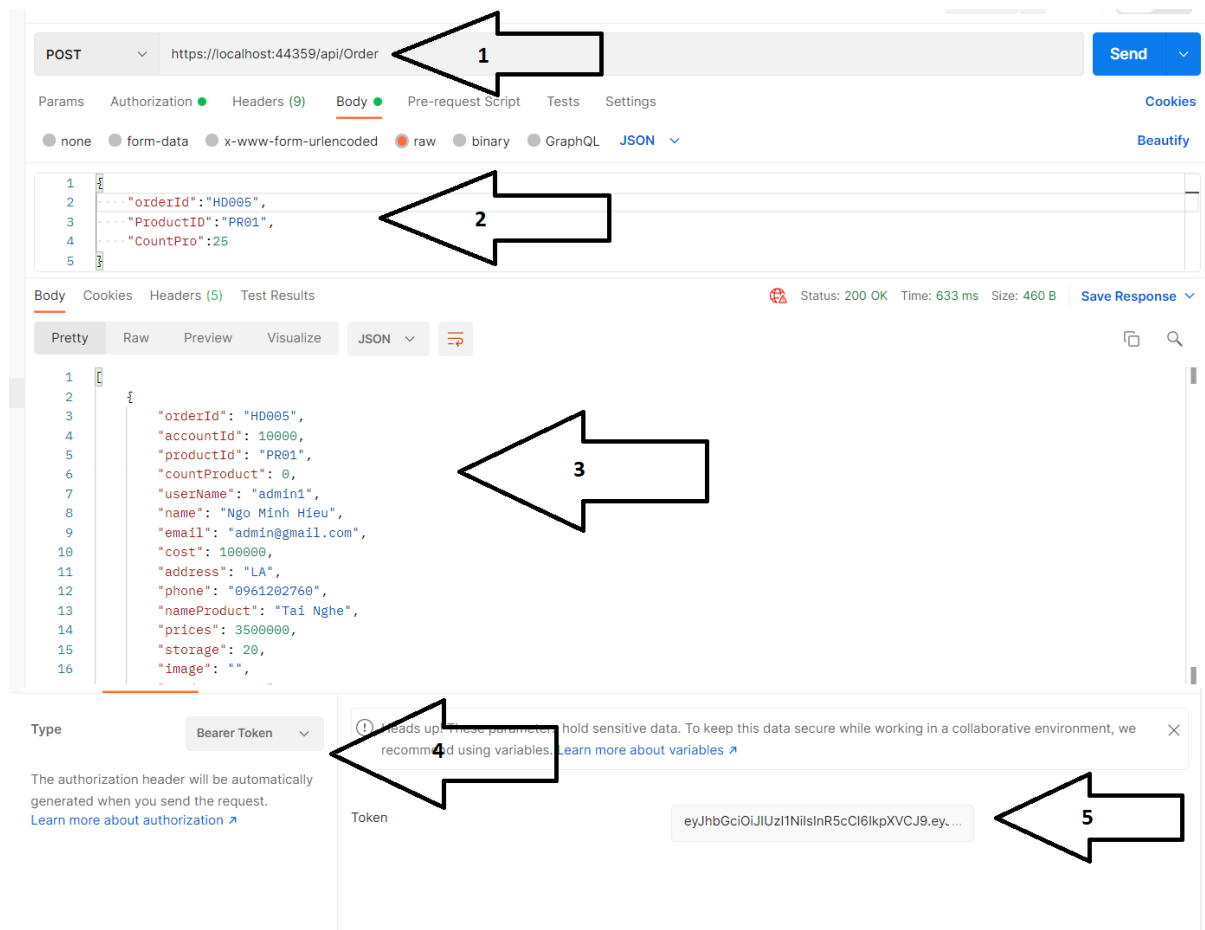
Hình 15 : API lấy sản phẩm theo ID đơn hàng

- 1 : Đường dẫn tới api : <https://localhost:44359/api/Order/orderID/{idorder}>
- 1 : Phương thức được sử dụng là phương thức GET, dùng để lấy tất cả sản phẩm trong cùng một order
- 2 : Đính jwt vào bearer token để lấy được userid trong jwt.
- 3 : Thông tin được trả về nếu cập nhật thành công.



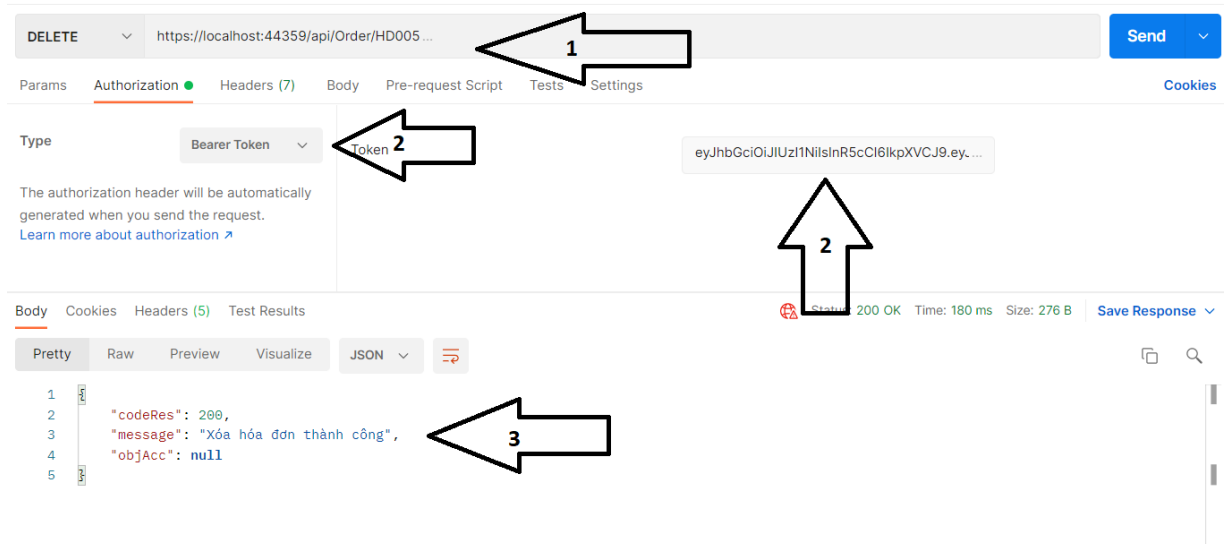
Hình 16 : API lấy tất cả đơn hàng theo ID của user

- 1 : Đường dẫn tới api : <https://localhost:44359/api/Order>
- 1 : Phương thức được sử dụng là phương thức GET, dùng để lấy tất cả order của user
- 2 : Đính jwt vào bearer token để lấy được userid trong jwt.
- 3 : Thông tin được trả về nếu lấy hóa đơn thành công



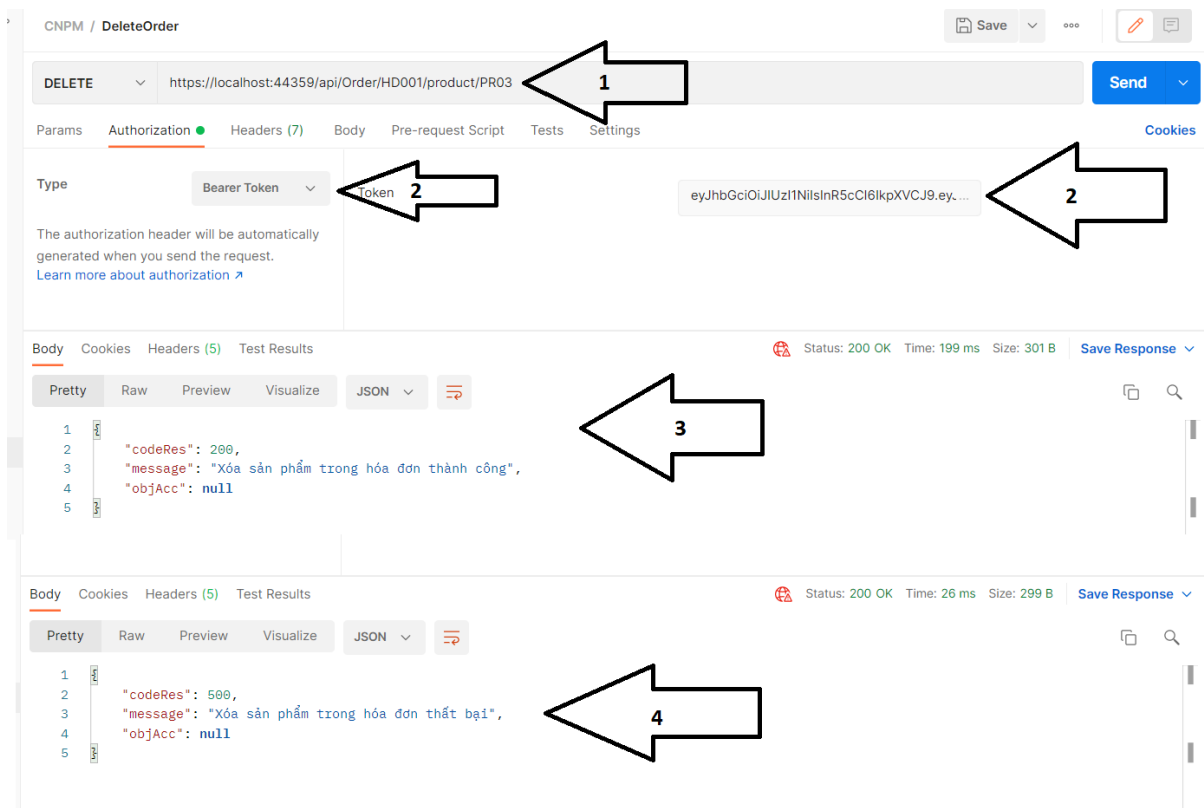
Hình 17 : API dùng để thêm sản phẩm vào đơn hàng

- 1 : Đường dẫn tới api : <https://localhost:44359/api/Order>
- 1 : Phương thức được sử dụng là phương thức Post, dùng thêm hóa đơn với sản phẩm
- 2 : Thông tin đính kèm gồm orderId, ProductID, countPro
- 3 : Thông tin được trả về nếu thêm hóa đơn thành công
- 4 : Đính jwt vào bearer token để lấy được userid trong jwt.



Hình 18 : API dùng để xóa đơn hàng

- 1 : Đường dẫn tới api : <https://localhost:44359/api/Order/{idOrder}>
- 1 : Phương thức được sử dụng là phương thức Delete, dùng xóa hóa đơn
- 2 : Thông tin được trả về nếu thêm hóa đơn thành công
- 3 : Đính jwt vào bearer token để lấy được userid trong jwt.



Hình 19 : API dùng để xóa sản phẩm trong đơn hàng

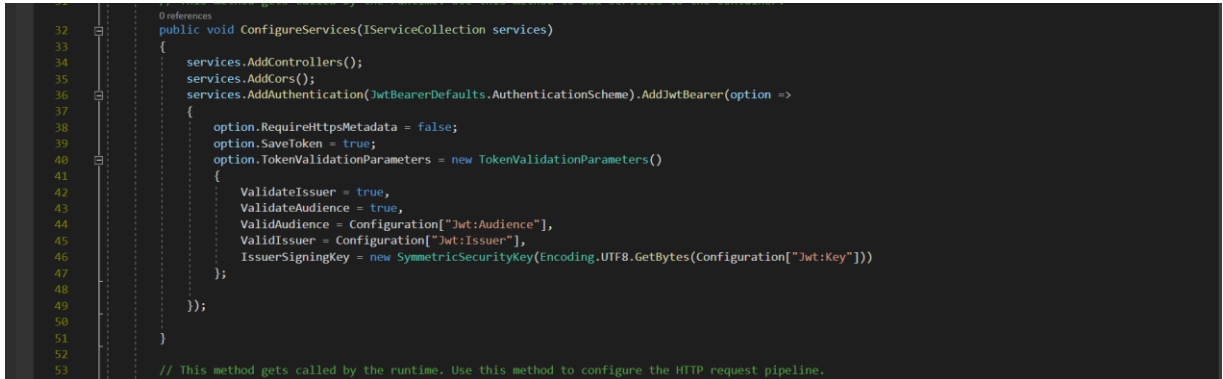
- 1 : Đường dẫn tới api :

<https://localhost:44359/api/Order/{idOrder}/product/{idProduct}>

- 1 : Phương thức được sử dụng là phương thức Delete, dùng xóa sản phẩm trong hóa đơn
- 2 : Đính jwt vào bearer token để lấy được userid trong jwt.
- 3 : Thông tin được trả về nếu thêm hóa đơn thành công
- 4 : Nếu thực hiện thêm 1 lần xóa sản phẩm thì sẽ báo lỗi vì sản phẩm trong order đã được xóa trước đó nên trả về thông báo lỗi

2. MỘT SỐ ĐOẠN CODE CẦN GIẢI THÍCH

a) WEB API



```

32 public void ConfigureServices(IServiceCollection services)
33 {
34     services.AddControllers();
35     services.AddCors();
36     services.AddAuthentication(JwtBearerDefaults.AuthenticationScheme).AddJwtBearer(option =>
37     {
38         option.RequireHttpsMetadata = false;
39         option.SaveToken = true;
40         option.TokenValidationParameters = new TokenValidationParameters()
41         {
42             ValidateIssuer = true,
43             ValidateAudience = true,
44             ValidAudience = Configuration["Jwt:Audience"],
45             ValidIssuer = Configuration["Jwt:Issuer"],
46             IssuerSigningKey = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(Configuration["Jwt:Key"]))
47         };
48     });
49 }
50
51 // This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
52
53

```

Hình 20: Hình ảnh đoạn code cho phép web api sử dụng jwt

- Option.RequireHttpsMetadata : Cho phép yêu cầu địa chỉ hoặc một địa chỉ siêu dữ liệu và mặc định là true và set là false khi trong môi trường phát triển
- SaveToken : Được sử dụng để phía server lưu lại token và xác thực chúng thì set bằng true
- TokenValidationParameters : Dùng để chứa 1 tập hợp các câu lệnh để thiết lập cho vấn đề xác thực qua token
 - ValidateIssuer : xác thực là người dùng hay không thiết lập true
 - ValidateAudience : kiểm soát đối tượng đã được xác thực hay chưa
 - ValidAudience : set bằng 1 chuỗi nào đó mà ta đặt sẵn dùng để kiểm soát mặc định là null
 - ValidIssuer: set bằng 1 chuỗi nào đó để xác thực người dùng mặc định là null
 - IssuerSigningKey: Được dùng để tạo ra khóa bí mật để ký và xác thực người dùng

```

app.UseCors(x => x
    .AllowAnyMethod()
    .AllowAnyHeader()
    .SetIsOriginAllowed(origin => true) // allow any origin
    .AllowCredentials()); // allow credentials

```

Hình 21 : Hình ảnh đoạn code mở CORS cho web api

- Ta sử dụng mở CORS cho web api để tiến hành kết nối với phía front-end để 2 side là web api(.NET) và React (JS) giao tiếp với nhau trao đổi và thay đổi dữ liệu

```

app.Use(async (context, next) =>
{
    await next();

    if (context.Response.StatusCode == (int)HttpStatusCode.Unauthorized) // 401
    {
        context.Response.ContentType = "application/json";
        await context.Response.WriteAsync(new messageSend
        {
            codeRes = 401,
            message = "Token is not valid",
        }.showMess());
    }
});

```

Hình 22 : Hình ảnh đoạn code tạo ra kết quả trả token

- Khi đăng nhập với jwt không đúng thì kết quả chỉ trả về no Authorzie việc này khó thể bắt sự kiện ở phía front-end để xử lý
- Nên tạo ra 1 kết quả trả về có code mà message để khi bất kì side nào gọi đến API và không có token hợp lệ nó sẽ trả message cho người gọi đến API biết

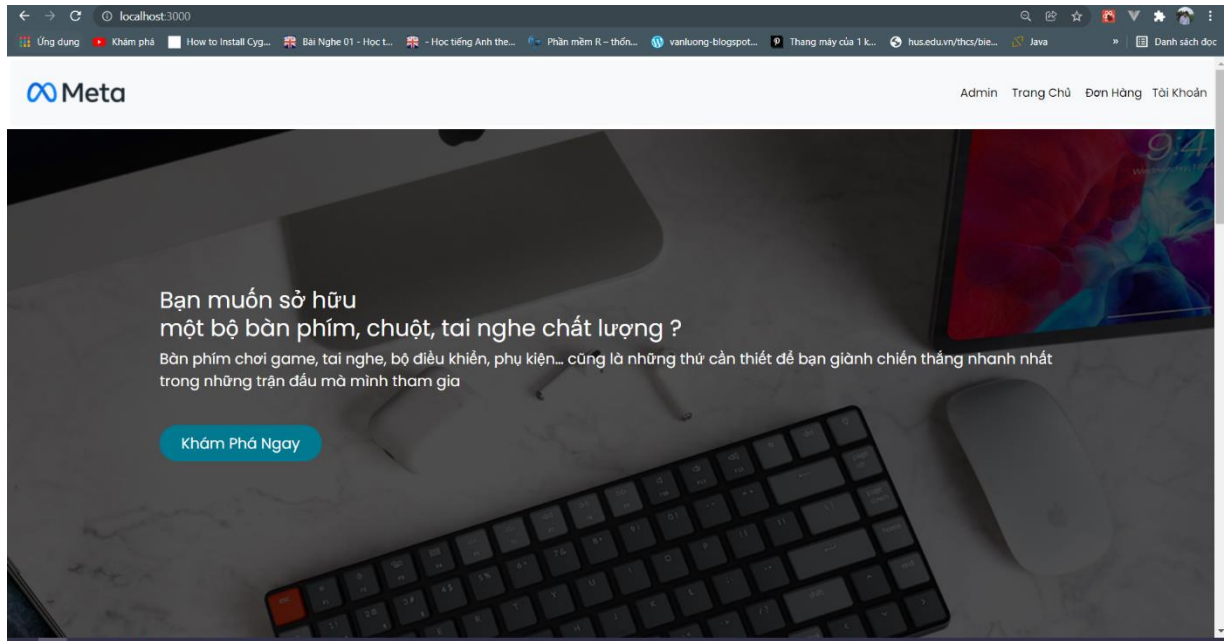
b) REACT JS

```
const UploadImage = (file) => {
  const formData = new FormData();
  formData.append("file", file);
  formData.append("upload_preset", "dev_setup");
  setDisable(true);
  PostUploadImg(formData).then((response) => {
    if (response) {
      setProduct({ ...product, Image: response.url });
      setDisable(false);
    }
  });
};
```

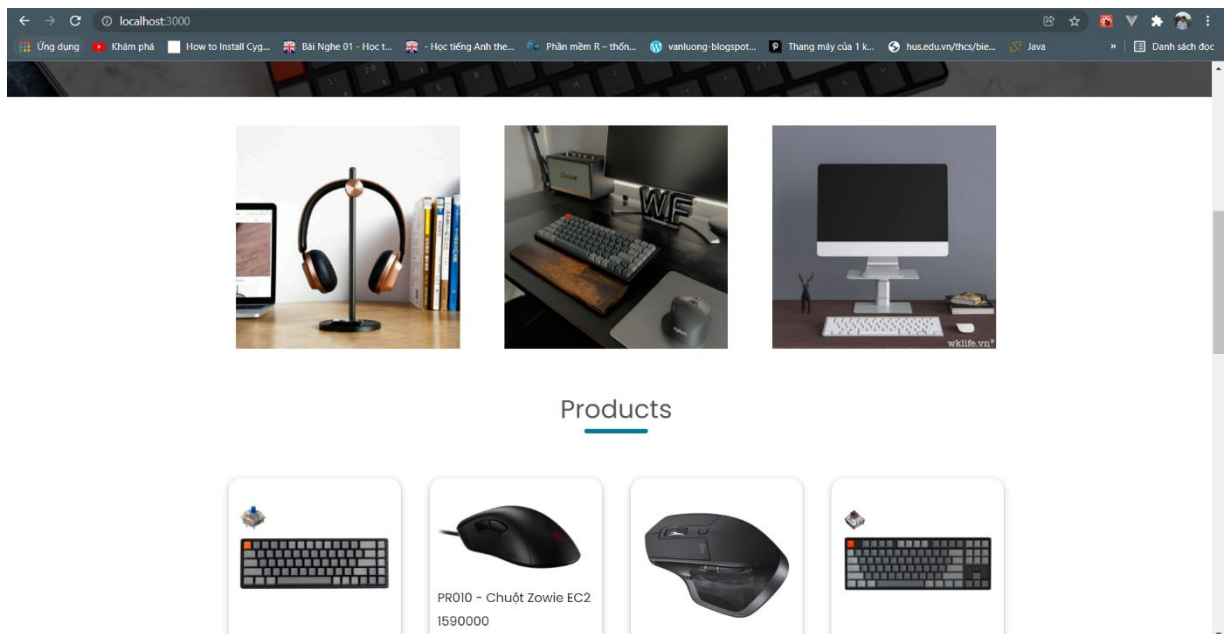
Hình 23 : Hình ảnh đoạn code giúp lưu ảnh ở 1 side online

- Để lưu trữ 1 tấm ảnh ở tập tin của dự án mỗi lần lưu hình ảnh thì dung lượng của tập tin sẽ ngày càng lớn
- Vì thế ở dự án này sẽ sử dụng 1 side lưu ảnh có tên là : <https://cloudinary.com/>
- Việc đơn giản là ta đăng kí 1 tài khoản và tạo 1 tập tin online trên cloudinary và lưu nó với key api của mỗi người cùng với tên của tập tin online đó
- Để tránh việc nhấn quá nhiều thì khi lưu ảnh sẽ setdisable để khóa thao tác thêm sản phẩm đến khi đường dẫn tấm ảnh online trả về thì sẽ thiết lập disable cho phép người quản trị tiến hành thêm sản phẩm

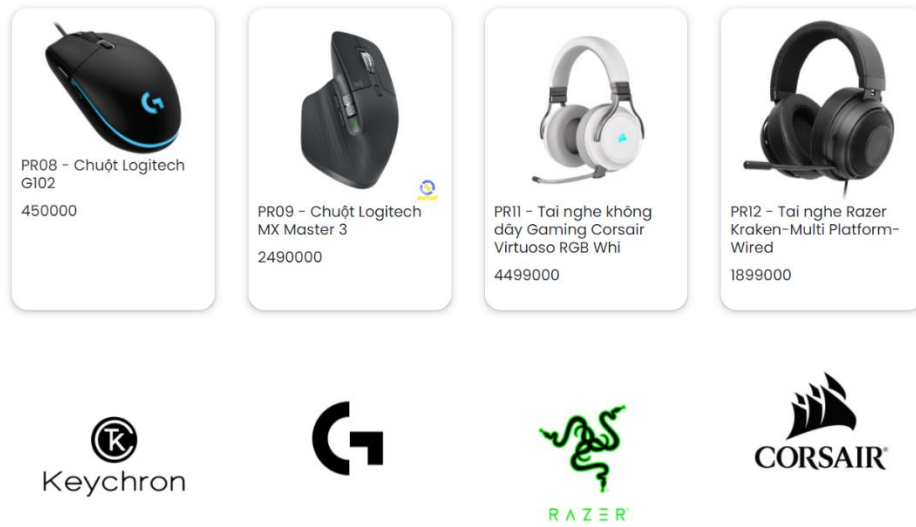
3. XÂY DỰNG VỚI REACJS



Hình 24 : Giao diện trang chủ phần thanh thông báo



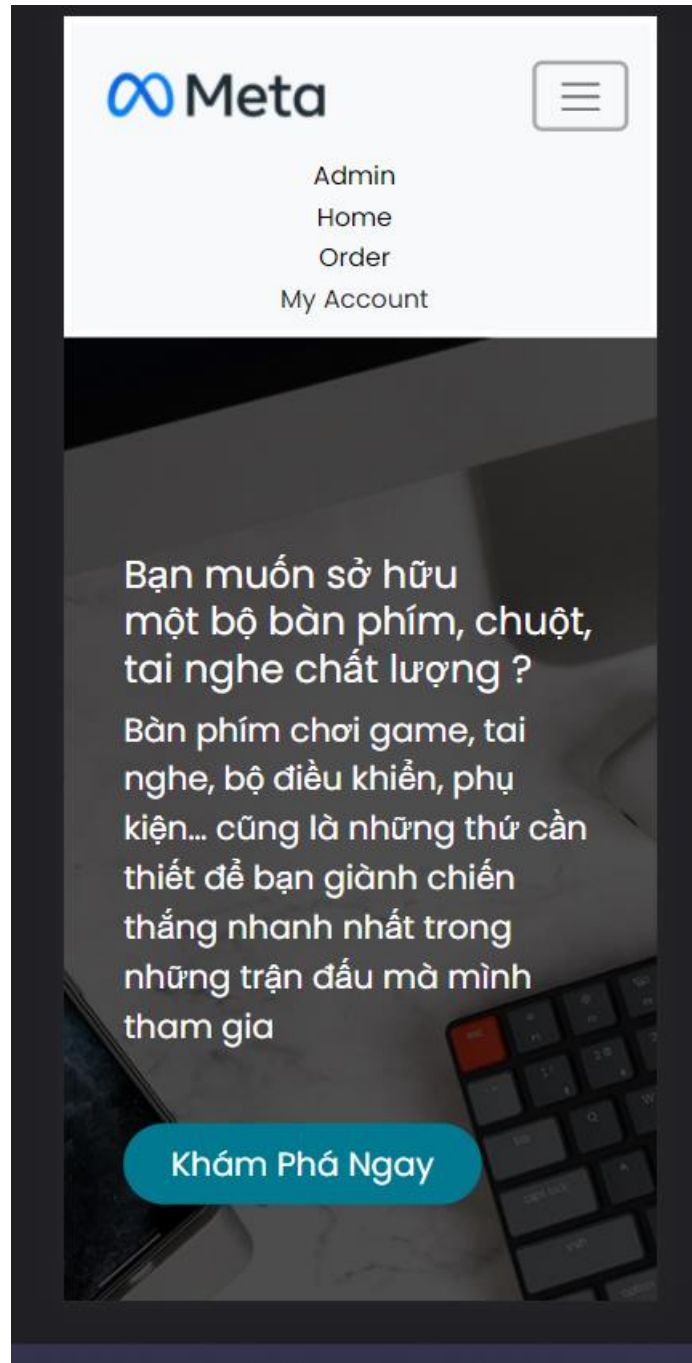
Hình 25 : Giao diện trang chủ khi kéo xuống



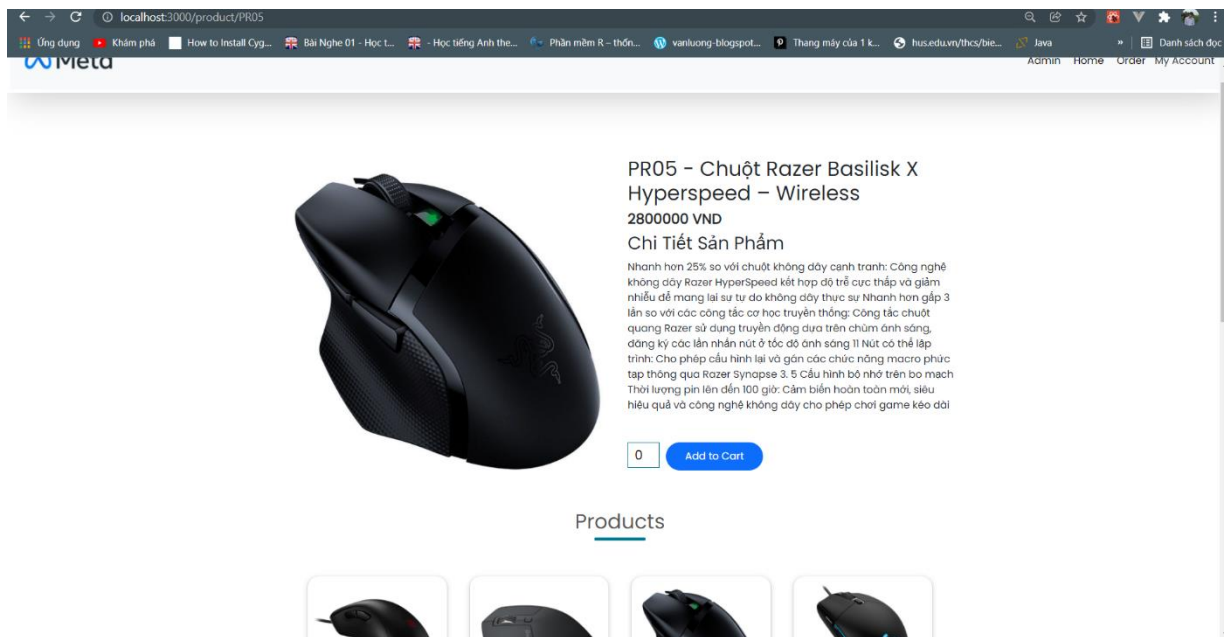
Hình 26 : Hình ảnh trưng bày cho một số nhãn hàng



Hình 27 : Hình ảnh cuối trang web

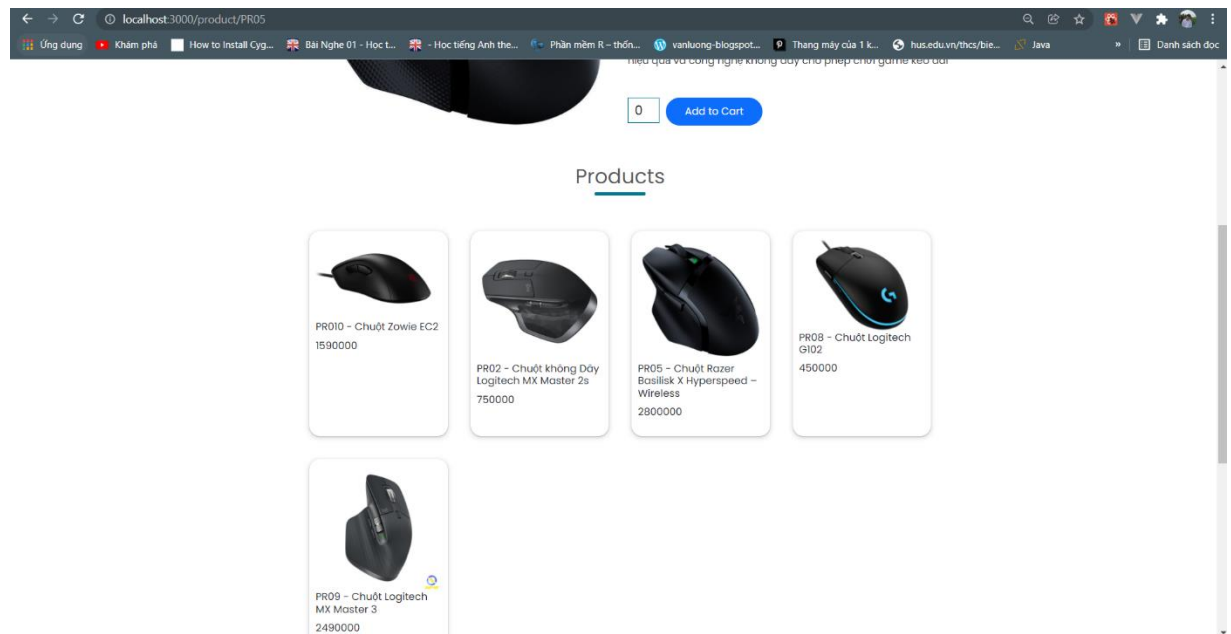


Hình 28 : Hình ảnh trang web sử dụng ở màn hình điện thoại



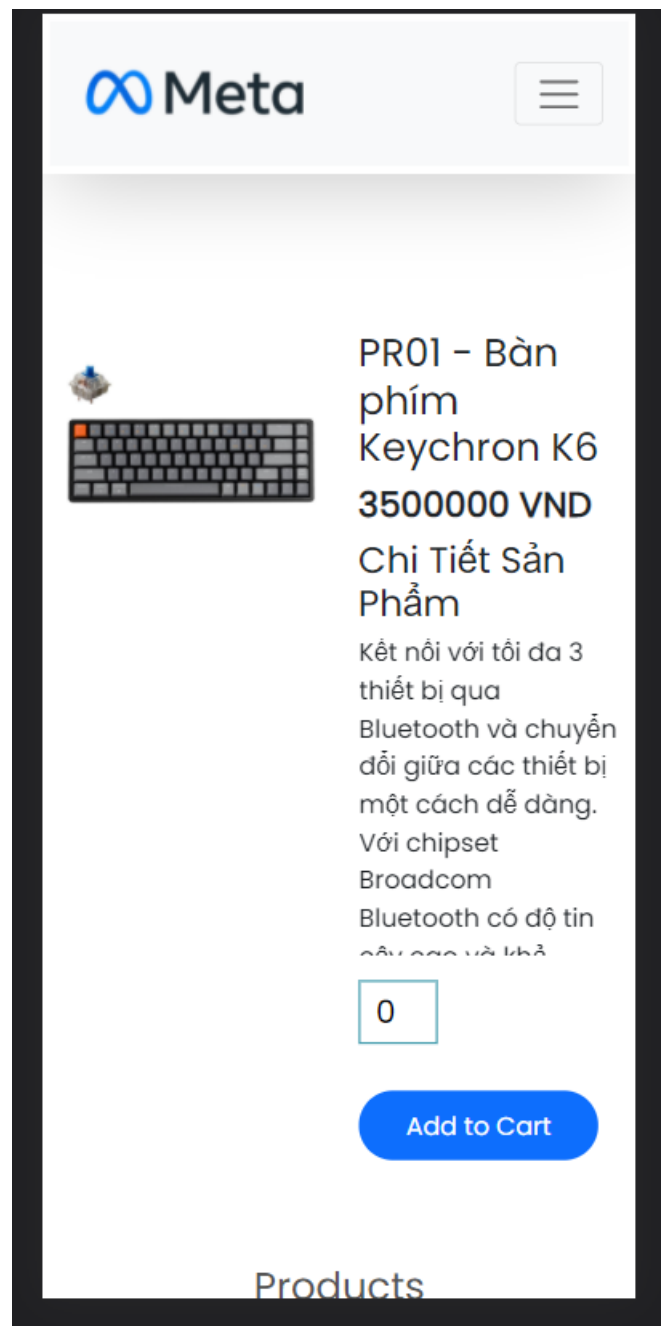
Hình 29 : Hình ảnh chi tiết sản phẩm

- Khi người dùng nhấn vào sản phẩm ở trang chủ người dùng sẽ được dẫn đến trang chi tiết sản phẩm ở đây người dùng có thêm sản phẩm vào giỏ hàng

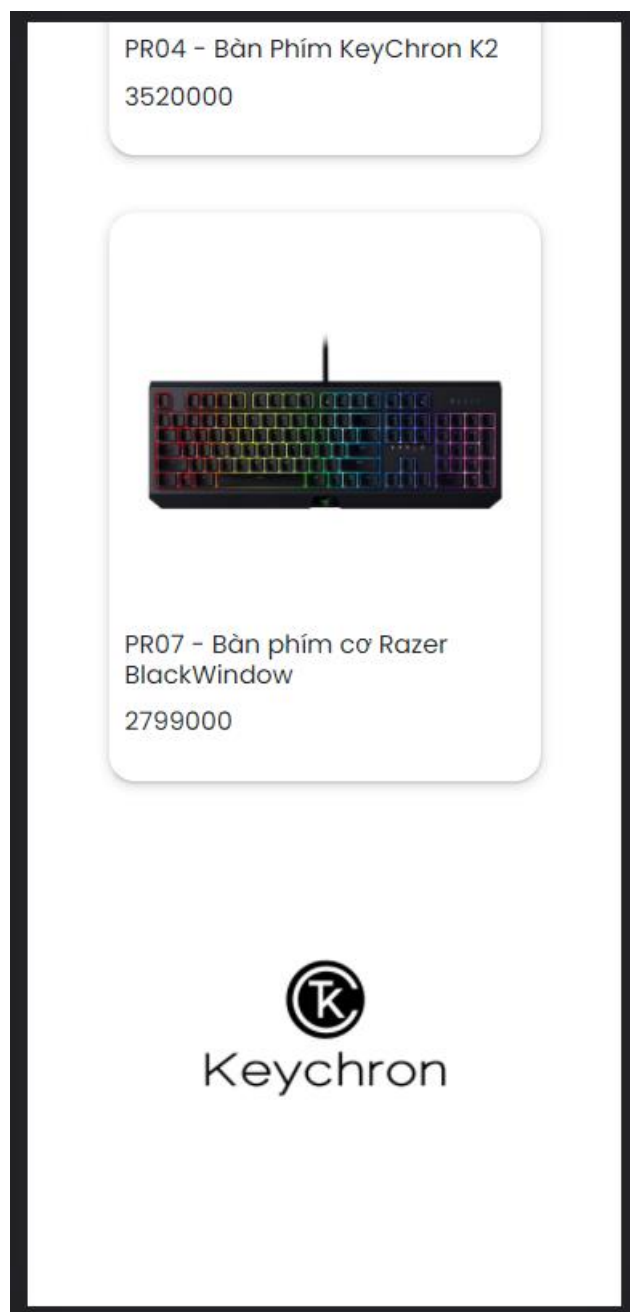


Hình 30 : Hình ảnh sản phẩm cùng loại trong chi tiết sản phẩm

- Hoặc có thể xem các sản phẩm cùng loại ngay bên dưới sản phẩm đó



Hình 31 : Hình ảnh trang chi tiết sản phẩm sử dụng ở điện thoại



Hình 32 : Hình ảnh các sản phẩm khác cùng loại khi hiển thị ở thiết bị điện thoại


Hóa Đơn Của Bạn: HD2412202103649

Họ Và Tên: Ngô Minh Hiếu

Phone: 0961202760

Address: Hồ Chí Minh

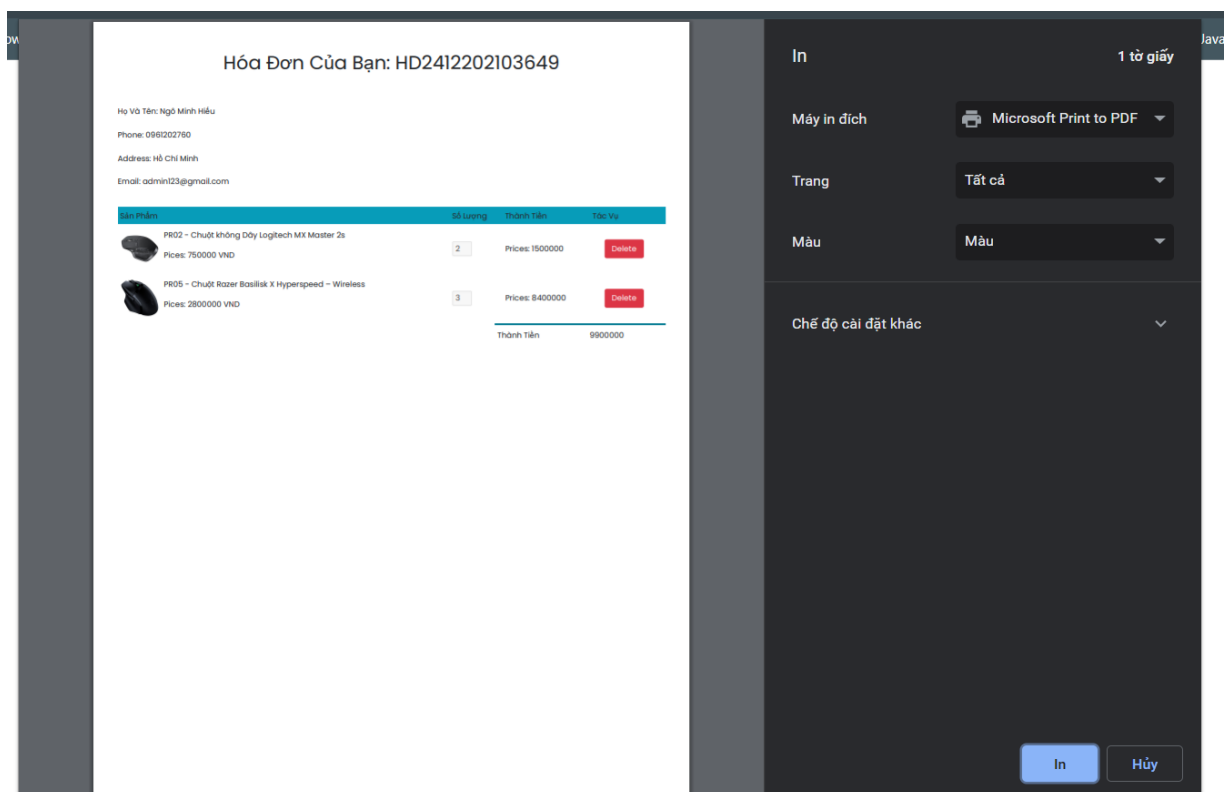
Email: admin123@gmail.com

Sản Phẩm	Số Lượng	Thành Tiền	Tác Vụ
 PR05 - Chuột Razer Basilisk X Hyperspeed – Wireless Pices: 2800000 VND	<input type="text" value="3"/>	Prices: 8400000	Delete
		Thành Tiền	8400000
Thanh Toán			



Hình 33 : Hình ảnh chi tiết hóa đơn hiện tại

- Khi người dùng thêm sản phẩm vào đơn hàng người dùng có thể kiểm tra lại đơn hàng
- Có thể xóa vật phẩm và tiến hành thêm vật phẩm mới vào trong đơn hàng
- Khi cảm thấy đủ sản phẩm như mong muốn người dùng có thể nhấn vào nút thanh toán và sẽ tiến hành in hóa đơn



Hình 34 : Hình ảnh việc in hóa đơn của khách hàng

Hóa Đơn Của Bạn:


HD2412202103649

Họ Và Tên: Ngô Minh Hiếu

Phone: 0961202760

Address: Hồ Chí Minh

Email: admin123@gmail.com

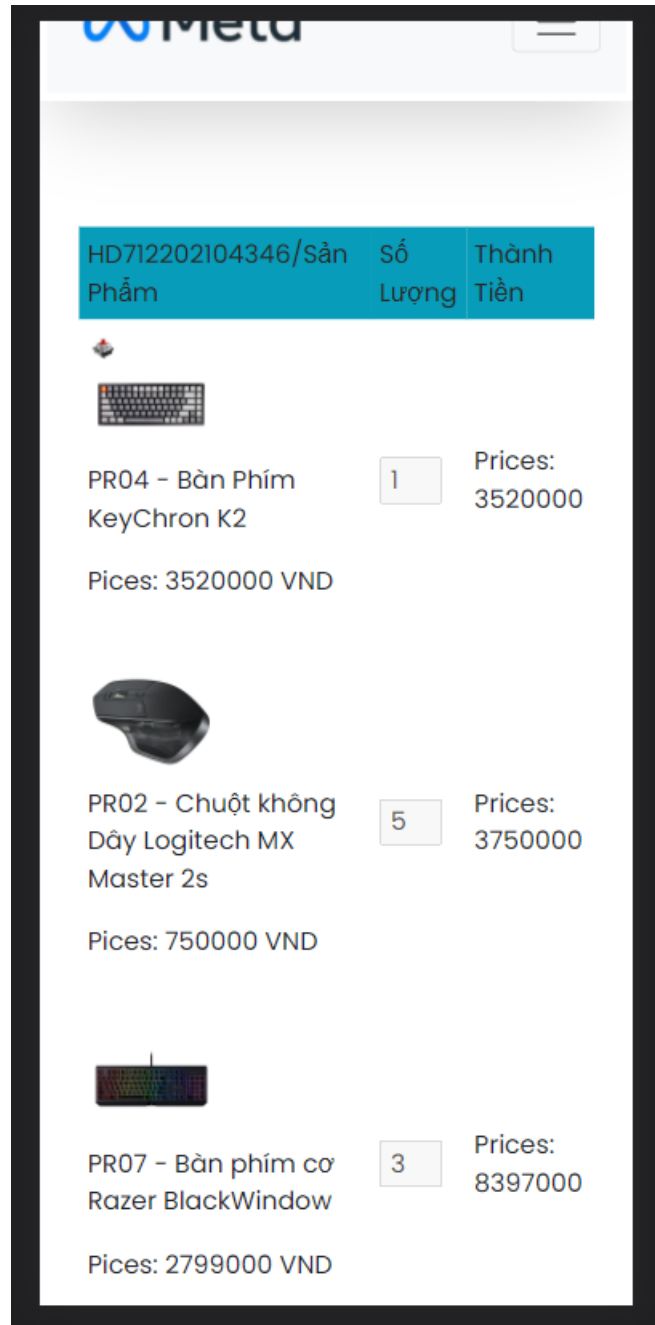
Sản Phẩm	Số Lượng	Thành Tiền	Tác Vụ
 <div> PR05 - Chuột Razer Basilisk X Hyperspeed - Wireless </div>	<input type="text" value="3"/>	Prices: 8400000	<div>Delete</div>
Pices: 2800000 VND			
Thành Tiền		8400000	
<div>Thanh Toán</div>			

Hình 35 : Hình ảnh hóa đơn khi sử dụng với thiết bị điện thoại

HD712202104346/Sản Phẩm	Số Lượng	Thành Tiền
PR04 - Bàn Phím KeyChron K2 Pices: 3520000 VND	1	Prices: 3520000
PR02 - Chuột không Dây Logitech MX Master 2s Pices: 750000 VND	5	Prices: 3750000
PR07 - Bàn phím cơ Razer BlackWindow Pices: 2799000 VND	3	Prices: 8397000
HD712202103020/Sản Phẩm	Số Lượng	Thành Tiền
PR04 - Bàn Phím KeyChron K2 Pices: 3520000 VND	2	Prices: 7040000
HD2412202103649/Sản Phẩm	Số Lượng	Thành Tiền
PR05 - Chuột Razer Basilisk X Hyperspeed - Wireless Pices: 2800000 VND	3	Prices: 8400000

Hình 36 : Hình ảnh tất cả hóa đơn mà khách từng đặt hàng

- Khách hàng có thể kiểm tra tất cả các hóa đơn mình đã từng đặt khi chọn vào hóa đơn -> tất cả hóa đơn



Hình 37 : Hình ảnh tất cả hóa đơn khi sử dụng ở thiết bị điện thoại

Thông Tin của bạn

Address
Hô Chí Minh

Your Name
Ngô Minh Hiếu

Your Phone
0961202760

Thay Đổi Thông Tin

Bạn có muốn chỉnh sửa ?

Hình 38 : Hình ảnh trang thông tin của tài khoản

- Khi người dùng nhấn vào tên của mình ở trang chủ trên thanh công cụ, người dùng có thể xem được thông tin của mình đã đăng kí và có thể thay đổi thông tin ấy nếu cần

Meta Trang Chủ

Người Dùng 3

Sản Phẩm 12

Lấy Người Dùng

Thêm Sản Phẩm
















Bảng Sản Phẩm

Sản Phẩm	Số Lượng	Thành Tiền	Tác vụ
PR01 - Bàn phím Keychron K6	7	Tiền: 3500000 VND	
PR010 - Chuột Zowie EC2	5	Tiền: 1590000 VND	
PR02 - Chuột không Dây Logitech MX Master 2s	12	Tiền: 750000 VND	

Hình 39 : Hình ảnh trang giao diện của người quản lý

Thêm Sản Phẩm

Bảng Sản Phẩm

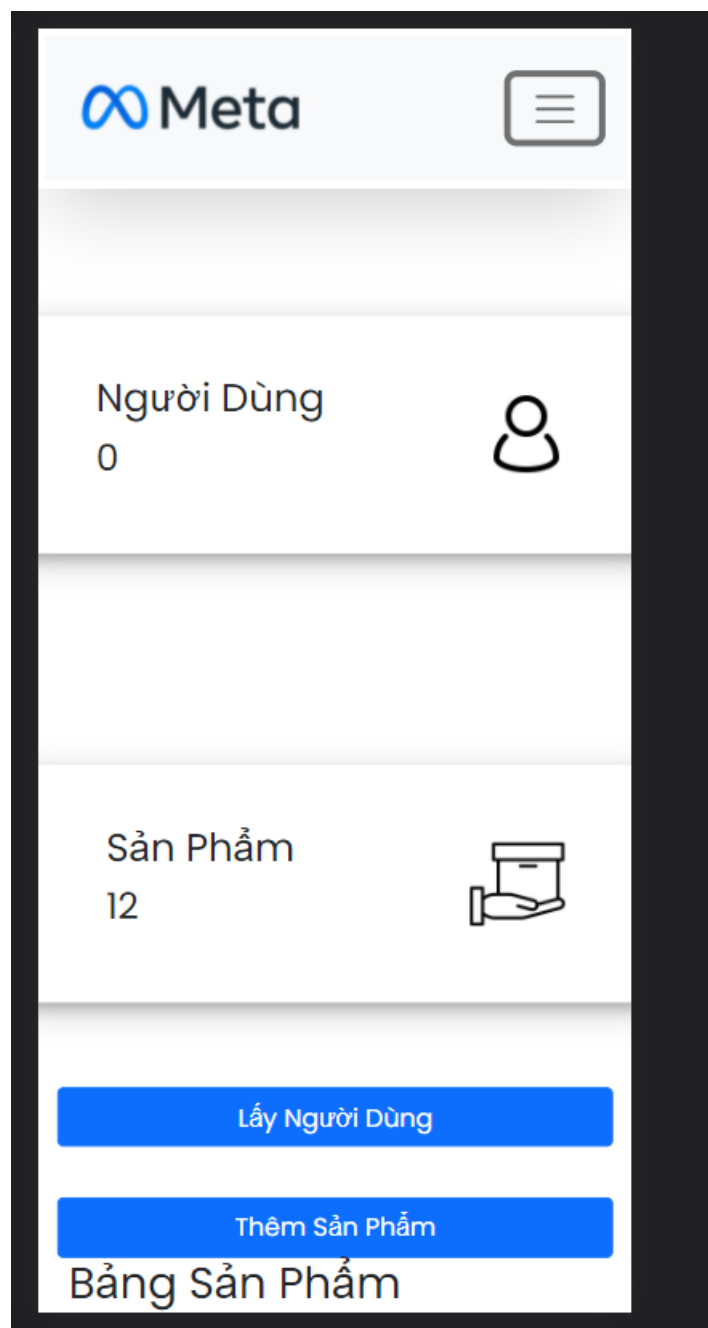
Sản Phẩm	Số Lượng	Thành Tiền	Tác vụ
 PR01 - Bàn phím Keychron K6	<input type="text" value="7"/>	Tiền: 3500000 VND	 
 PR010 - Chuột Zowie EC2	<input type="text" value="5"/>	Tiền: 1590000 VND	 
 PR02 - Chuột không Dây Logitech MX Master 2s	<input type="text" value="12"/>	Tiền: 750000 VND	 
 PR03 - Bàn Phím KeyChron K8	<input type="text" value="12"/>	Tiền: 3000000 VND	 
 PR04 - Bàn Phím KeyChron K2	<input type="text" value="15"/>	Tiền: 3520000 VND	 

Bảng Người Dùng

Email - Họ và Tên	Phone - Mã Tài Khoản	Địa Chỉ
admin@gmail.com Ngo Minh Hieu	0961202760 10000	LA
admin123@gmail.com Ngo Minh Hieu	0961202760 11107	Hồ Chí Minh

Hình 40 : Hình ảnh quản lý sản phẩm và người dùng

- Ở 2 hình trên ta có thể thấy được số người trong hệ thống cũng như là số lượng sản phẩm mà trang web đang có



Hình 41 : Hình ảnh trang giao diện quản trị khi sử dụng ở thiết bị điện thoại

PR010 - Chuột Zowie EC2

5

1590000 VND

PR02 - Chuột không

12

Tiền: 750000 VND

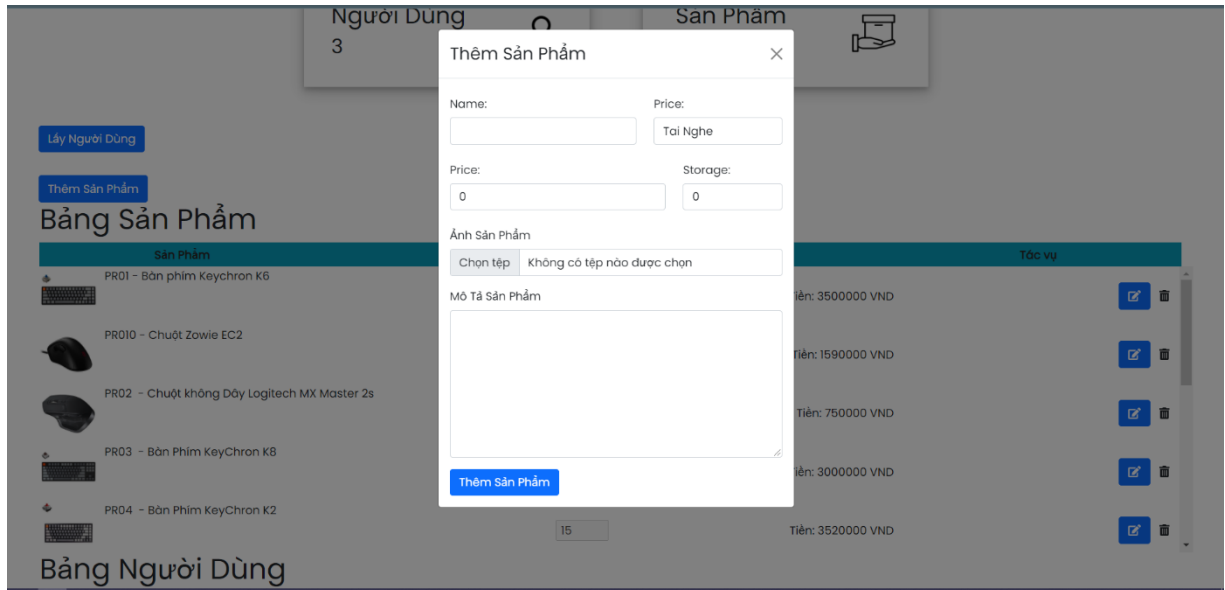
12

Tiền: 3000000

Bảng Người Dùng

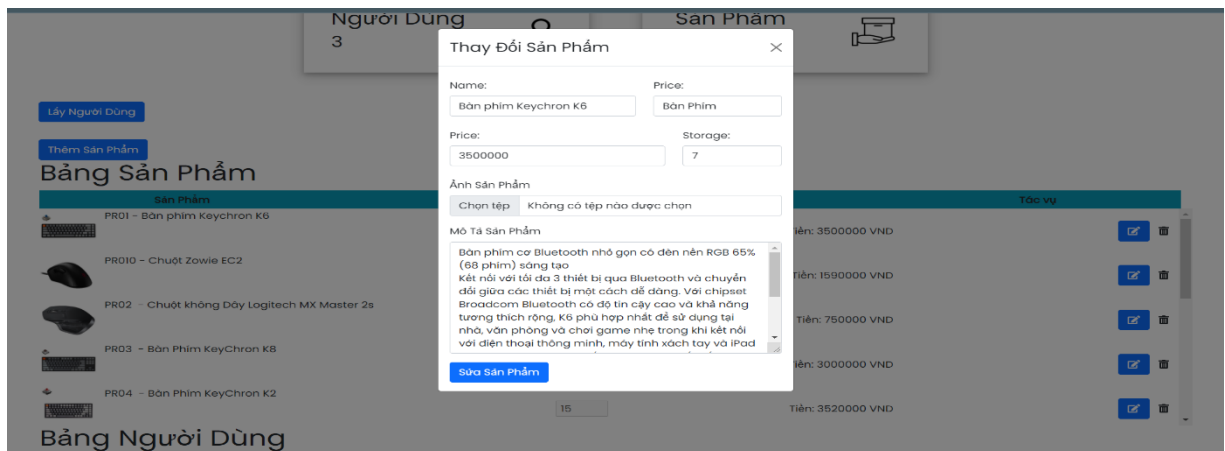
Email - Họ và Tên	Phone - Mã Tài Khoản	Địa Chỉ
admin@gmail.com Ngo Minh Hieu	0961202760 10000	LA
admin123@gmail.com Ngô Minh Hiếu	0961202760 11107	Hồ Chí Minh
hieule@gmail.com Le Minh Hieu	0999991222 14059	Hồ Chí Minh

Hình 42 : Hình ảnh trang giao diện quản trị khi sử dụng ở thiết bị điện thoại



Hình 43 : Hình ảnh khi người quản trị muốn thêm sản phẩm

- Người quản trị sẽ nhấn vào nút thêm sản phẩm khi đó một modal sẽ hiện ra là form để người quản trị nhập thông tin sản phẩm vào để thêm vào hệ thống



Hình 44 : Hình ảnh người quản trị muốn thay đổi thông tin sản phẩm

- Người quản trị muốn thay đổi giá , tên sản phẩm , hay số lượng , mô tả , hình ảnh của sản phẩm thì chỉ cần nhấn vào biểu tượng edit ở phần tác vụ để mở thông báo chính sửa và ở đây người quản trị sẽ tiến hành thay đổi thông tin cho phù hợp

4. KẾT LUẬN

- Khuyết Điểm :

- Còn sử dụng query gọi nhiều lần đến database để lấy , thêm , sửa , xóa quá nhiều lần
- Chưa làm được tìm kiếm sản phẩm , hóa đơn, tài khoản ở phía người quản trị , và tìm kiếm sản phẩm ở phía người dùng
- Trang web chưa được đẹp mắt , UI/UX ở mức chấp nhận được và có một số điểm gây cho người dùng sự khó chịu
- Chưa có bộ lọc theo giá tiền , theo bản chữ cái
- Ở phía người quản trị phải thực hiện nút bấm lấy người dùng thì mới loading được danh sách người dùng (Bán tự động)
- Nếu loading 2 danh sách sản phẩm và người dùng tự động thì trang admin sẽ bị giật lag

- Ưu điểm :

- Dễ dàng sử dụng, sửa chữa
- Có thể tìm được lỗi và sửa ngay

- Tương lai:

- Phát triển tính năng tìm kiếm sản phẩm theo mã sản phẩm , theo tên sản phẩm
- Lọc dữ liệu theo giá tiền , theo chữ cái
- Liên kết đăng nhập với google mail
- Cải thiện giao diện khi hiển thị ở thiết bị điện thoại

TÀI LIỆU THAM KHẢO

1) Các bài LAB trên lớp thực hành

- [1] L. Jackson, ".NET Core 3.1 MVC REST API - Full Course," 22 02 2020. [Online]. Available: <https://www.youtube.com/watch?v=fmvcAzHpsk8&t=1965s>.
- [2] K. L. Rick Anderson, "Microsoft," [Online]. Available: <https://docs.microsoft.com/en-us/aspnet/core/tutorials/first-web-api?view=aspnetcore-6.0&tabs=visual-studio>.
- [3] CodingNepal, "Animated Login Form using HTML & CSS only | No JavaScript or jQuery," 24 08 2019. [Online]. Available: <https://www.youtube.com/watch?v=eeHqZeJ9Vqc>.
- [4] E. Tutorials, "How To Make Ecommerce Website Using HTML And CSS Step By Step | Create e-Commerce Website," 08 07 2020. [Online]. Available: <https://www.youtube.com/watch?v=yQimoqo0-7g&t=1503s>.
- [5] E. Tutorials, "Part 3 Ecommerce Website Design In HTML & CSS | Create eCommerce Site Product Details Page," 18 07 2020. [Online]. Available: https://www.youtube.com/watch?v=ENyk_W-Eleo&t=1047s.
- [6] E. Tutorials, "How To Make Shopping Cart Page On eCommerce Website | E-commerce Web Design Using HTML CSS," 21 07 2020. [Online]. Available: <https://www.youtube.com/watch?v=oXrlgOEiy6o>.
- [7] i. creative, "Responsive Admin Dashboard using HTML and CSS," 12 05 2021. [Online]. Available: <https://www.youtube.com/watch?v=kAt27KFNZNY>.