**ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**

**CÔNG NGHỆ PHẦN MỀM MỚI**

**Đề tài:**

**XÂY DỰNG WEBSITE BÁN**

**PHỤ KIỆN ĐIỆN THOẠI**

|  |
| --- |
| **SVTH: PHẠM QUỐC VIỆT** |
| **MSSV: 17110254** |
| **SVTH: DƯƠNG TẤN VỸ** |
| **MSSV: 17145255** |
| **Khoá: 2017** |
| **Ngành: CÔNG NGHỆ THÔNG TIN** |
| **GVHD:**  **NGUYỄN THANH PHƯỚC** |

Thành phố Hồ Chí Minh, tháng 01 năm 2021

**LỜI CẢM ƠN**

Lời đầu tiên, nhóm thực hiện xin được phép gửi lời cảm ơn chân thành đến khoa Đào tạo Chất Lượng Cao – Trường Đại học Sư phạm Kỹ thuật Thành phố Hồ Chí Minh đã tạo mọi điều kiện thuận lợi nhất cho nhóm thực hiện có cơ hội được tự do tiếp cận, tham khảo, mở rộng thêm kiến thức trong lĩnh vực Công nghệ thông tin nói chung và môn Công nghệ phần mềm mới nói riêng!

Lời cảm ơn trân trọng nhất nhóm thực hiện xin chân thành gửi đến Thầy **Nguyễn Thanh Phước** cùng đồng hành và trực tiếp giảng dạy, hướng dẫn và tạo mọi điều kiện thuận lợi giúp đỡ cho nhóm phát huy hết khả năng cũng như nâng cao kiến thức trong suốt quá trình học tập, đặc biệt là trong quá trình chuẩn bị và thực hiện môn học Công nghệ phần mềm mới. Cảm ơn sự nhiệt tình của Thầy, là động lực vô cùng to lớn giúp nhóm thực hiện, kiên trì trong suốt quá trình thực hiện đề tài và khám phá ra những kiến thức mới đầy thú vị và bổ ích liên quan đến đề tài mà cụ thể là đề tài **Xây dựng website bán phụ kiện điện thoại**

**Nhóm sinh viên thực hiện**

**Phạm Quốc Việt**

**Dương Tấn Vỹ**

**LỜI MỞ ĐẦU**

Ngày nay công nghệ thông tin đã và đang phát triển đạt đến mức độ cao, được ứng dụng vào mọi ngành, mọi lĩnh vực của nền kinh tế đất nước góp phần quan trọng làm cho các ngành này phát triển nhanh hơn, hiệu quả hơn. Việc thiết kế và xây dựng các website trong thời đại công nghệ 4.0 này nhằm đáp ứng nhu cầu ngày càng cao của người dùng trên nhiều lĩnh vực.

Bên cạnh đó nhu cầu mua phụ kiện điện thoại cũng như mua hàng online cũng được nhiều người quan tâm. Theo tìm hiểu thực tế, có nhiều cửa hàng đã tích hợp cả bán hàng tại của hàng và bán hàng thông qua website, tuy nhiên có nhiều cửa hàng vẫn chưa áp dụng việc bán hàng trên website nên việc quản lý trở nên phức tạp và khó kiểm soát hàng hóa. Vì vậy, việc phát triển và xây dựng một website với đầy đủ các chức năng để người dùng mua hàng trực tuyến và cho phép người quản lý cửa hàng có thể quản lý nghiệp vụ trực tiếp trên website tránh tình trạng không theo dõi được chính xác doanh thu và sản phẩm của cửa hàng.

**MỤC LỤC**

[LỜI CẢM ƠN i](#_Toc60516702)

[LỜI MỞ ĐẦU ii](#_Toc60516703)

[MỤC LỤC iii](#_Toc60516704)

[DANH MỤC CÁC HÌNH ẢNH, BIỂU ĐỒ vi](#_Toc60516705)

[DANH MỤC CÁC BẢNG BIỂU vii](#_Toc60516706)

[Chương 1: TỔNG QUAN 1](#_Toc60516707)

[1.1 Lý do chọn đề tài 1](#_Toc60516708)

[1.2. Mục tiêu nghiên cứu đề tài 1](#_Toc60516711)

[1.3. Công nghệ sử dụng 1](#_Toc60516712)

[1.4. Danh sách màn hình 1](#_Toc60516713)

[1.5. Danh sách service, endpoint 2](#_Toc60516714)

[Chương 2: NỘI DUNG THIẾT KẾ 4](#_Toc60516715)

[2.1. React 4](#_Toc60516718)

[2.2. NodeJS 4](#_Toc60516719)

[2.3. MongoDB 5](#_Toc60516720)

[2.4. High level design 6](#_Toc60516721)

[2.4.1. Database 6](#_Toc60516722)

[2.4.1.1. Thiết kế Database 6](#_Toc60516723)

[2.4.1.2. Các chức năng trong bảng 6](#_Toc60516724)

[2.4.2. Backend 7](#_Toc60516725)

[2.4.2.1. Express 7](#_Toc60516726)

[2.4.2.2. Express-async-handler 7](#_Toc60516727)

[2.4.2.3. Mongoose 7](#_Toc60516728)

[2.4.2.4. Dotenv 8](#_Toc60516729)

[2.4.2.5. Bcryptjs 8](#_Toc60516730)

[2.4.2.6. Jsonwebtoken 9](#_Toc60516731)

[2.4.2.7. Multer 9](#_Toc60516732)

[2.4.3. Front-end 9](#_Toc60516733)

[2.4.3.1. Redux 9](#_Toc60516734)

[2.4.3.2. Axios 11](#_Toc60516735)

[2.4.3.3. React Hook 11](#_Toc60516736)

[2.5. Detail design 12](#_Toc60516737)

[2.5.1. Front-end 12](#_Toc60516738)

[2.5.1.1. Màn hình trang chi tiết sản phẩm 12](#_Toc60516739)

[2.5.1.2. Màn hình trang giỏ hàng 15](#_Toc60516740)

[2.5.1.3. Màn hình trang đặt hàng 17](#_Toc60516741)

[2.5.2. Backend 20](#_Toc60516742)

[2.5.2.1. Định tuyến API 20](#_Toc60516743)

[2.5.2.2. Tạo dữ liệu sản phẩm mẫu 20](#_Toc60516744)

[2.5.2.3. Tạo sản phẩm mới 21](#_Toc60516745)

[2.5.2.4. Thêm sản phẩm vào giỏ hàng 22](#_Toc60516746)

[2.5.2.5. Đăng ký người dùng 22](#_Toc60516747)

[2.5.2.6. Cập nhật thông tin cá nhân của người dùng 23](#_Toc60516748)

[Chương 3: CÀI ĐẶT SẢN PHẨM 25](#_Toc60516749)

[3.1. Hướng dẫn cài đặt 25](#_Toc60516751)

[Chương 4: Kiểm thử 27](#_Toc60516752)

[4.1. Test plan 27](#_Toc60516754)

[4.1.1. Môi trường kiểm thử 27](#_Toc60516755)

[4.1.2. Nhân sự 27](#_Toc60516756)

[4.1.3. Phạm vi kiểm thử 27](#_Toc60516757)

[4.1.3.1. Những chức năng được kiểm thử 27](#_Toc60516758)

[4.1.3.2. Những chức năng không được kiểm thử 28](#_Toc60516759)

[4.1.4. Chiến lược kiểm thử 29](#_Toc60516760)

[4.1.5. Điều kiện chấp nhận 29](#_Toc60516761)

[4.1.6. Defect Tracking 29](#_Toc60516762)

[4.2. Test case 30](#_Toc60516763)

[Chương 5: TỔNG KẾT 37](#_Toc60516764)

[5.1. Kết luận 37](#_Toc60516768)

[5.2. Ưu điểm 37](#_Toc60516769)

[5.3. Nhược điểm 37](#_Toc60516770)

[5.4. Hướng phát triển 38](#_Toc60516771)

[TÀI LIỆU THAM KHẢO 39](#_Toc60516772)

# DANH MỤC CÁC HÌNH ẢNH, BIỂU ĐỒ

[Hình 1: Database 6](#_Toc60589977)

[Hình 2: Redux 10](#_Toc60589978)

[Hình 3: Trang chủ 12](#_Toc60589979)

[Hình 4: LoadingBox 12](#_Toc60589980)

[Hình 5: MessageBox 13](#_Toc60589981)

[Hình 6: Redux và hooks 13](#_Toc60589982)

[Hình 7: Hàm detailsProduct 14](#_Toc60589983)

[Hình 8: Hàm productDetailReducer 14](#_Toc60589984)

[Hình 9: Reducer 14](#_Toc60589985)

[Hình 10: Trang giỏ hàng 15](#_Toc60589986)

[Hình 11: Quản lý state 15](#_Toc60589987)

[Hình 12: Hàm useSelector 16](#_Toc60589988)

[Hình 13 : Hàm addToCart 16](#_Toc60589989)

[Hình 14: Hàm cartReducer 17](#_Toc60589990)

[Hình 15: Trang đặt hàng 17](#_Toc60589991)

[Hình 16: Component CheckoutSteps 18](#_Toc60589992)

[Hình 17: Hàm useSelector 18](#_Toc60589993)

[Hình 18: Hàm useDispatch 18](#_Toc60589994)

[Hình 19: Hàm createOrder 19](#_Toc60589995)

[Hình 20: Hàm orderCreateReducer 19](#_Toc60589996)

[Hình 21: Định tuyến API 20](#_Toc60589997)

[Hình 22: Phần productRoute 20](#_Toc60589998)

[Hình 23: Create product 21](#_Toc60589999)

[Hình 24: Thêm sản phẩm vào giỏ hàng 22](#_Toc60590000)

[Hình 25: Đăng ký người dùng 22](#_Toc60590001)

[Hình 26: Cập nhật thông tin cá nhân 23](#_Toc60590002)

# DANH MỤC CÁC BẢNG BIỂU

[Bảng 1: Mô tả chức năng 7](#_Toc60516940)

[Bảng 2: Nhân sự 27](#_Toc60516941)

[Bảng 3: Defect Tracking 30](#_Toc60516942)

[Bảng 4: Test case 36](#_Toc60516943)

# Chương 1: TỔNG QUAN

## Lý do chọn đề tài

Hiện nay do sự phát triển của công nghệ cũng như nhu cầu mua sắm ngày càng cao của người dùng nên dẫn đến sự phát triển của các trang web bán hàng online.

Nhu cầu sử dụng điện thoại và phụ kiện điện thoại của người dùng đang được gia tăng, để thuận tiện trong việc mua bán hàng trực tuyến nên nhóm chúng em chọn đề tài này



## Mục tiêu nghiên cứu đề tài

Thiết kế và xây dựng được một website bán phụ kiện điện thoại với đầy đủ các tính năng như: đăng tin, giới thiệu sản phẩm, giúp khách hàng có thể lựa chọn và đặt mua sản phẩm phù hợp. Ngoài ra hỗ trợ người quản lý của hàng dễ dàng quản lý nhân viên, sản phẩm, có thể dễ dàng theo dõi doanh thu và đơn hàng. Bên cạnh đó nhân viên bán hàng có thể kết hợp quản lý sản phẩm và bán hàng.

Xây dựng giao diện người dùng gần gũi thu hút khách hàng đặc biệt là dễ sử dụng và đặt hàng, dễ tiếp cận với mọi lứa tuổi. Với giao diện người quản lý có đầy đủ chức năng để quản lý nhân viên, sản phẩm.

## Công nghệ sử dụng

Nhóm sử dụng ngôn ngữ JavaScript

Back\_End : Sử dụng NodeJS với framework Express và MongoDB

Front\_End : Sử dụng Angular

## Danh sách màn hình

Customer role:

Trang home: cho phép người dùng xem các sản phẩm hiện có, đăng nhập, đăng ký tài khoản và mua hàng.

Trang đăng ký: cho phép người dùng có thể đăng ký tài khoản thành viên với tên, địa chỉ gmail và mật khẩu.

Trang đăng nhập: người dùng sau khi đăng ký tài khoản có thể dùng tên và mật khẩu để đăng nhập vào hệ thống.

Trang chi tiết sản phẩm: cho phép người dùng xem các thông tin của sản phẩm như: hình ảnh, mô tả sản phẩm và thực hiện thêm sản phẩm vào giỏ hàng nếu còn sản phẩm.

Trang giỏ hàng: hiển thị danh sách các sản phẩm đã được thêm vào giỏ hàng, người dùng có thể thêm, sửa và xóa sản phẩm khỏi giỏ hàng, hay tiến hành mua hàng.

Admin role:

Trang danh sách sản phẩm: cho phép người quản trị thêm, sửa và xóa sản phẩm.

Trang danh sách người dùng cho phép người quản trị thêm admin moi

Trang cho phép chỉnh sửa thông tin admin

## Danh sách service endpoint

Các endpoint được sử dụng trong chương trình:

/api/uploads: được sử dụng cho request POST tải ảnh lên phía frontend.

/api/users: được sử dụng cho request CRUD người dùng.

/api/products: được sử dụng cho request CRUD sản phẩm.

/api/orders: được sử dụng cho request CRUD đơn hàng.

/api/config/paypal: được sử dụng để gọi API của Sanbox Paypal.

**Chương 2: NỘI DUNG THIẾT KẾ**







## Angular

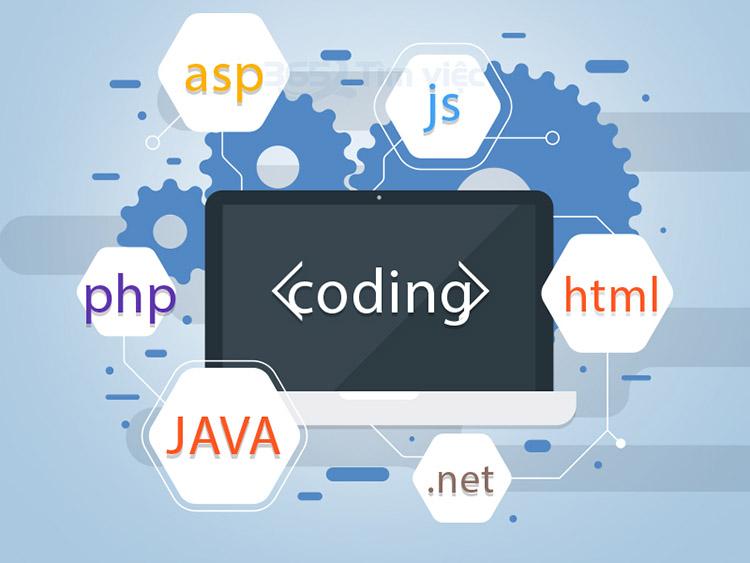
Angular là một Javacript framework do Google phát triển để xây dựng các Single Page Application (SPA) bằng JavaScript , HTML và TypeScript . Angular cung cấp các tính năng tích hợp cho animation

6, http service và có các tính năng như auto-complete , navigation , toolbar , menus ,… Code được viết bằng TypeScript , biên dịch thành JavaScript và hiển thị tương tự trong trình duyệt.

## 2.1.1 Sử dụng Angular sẽ làm được gì?

Sau khi hiểu rõ định nghĩa **Angular là gì** thì dưới đây là những điều bạn có thể áp dụng Angular để có thể dần làm quen với chúng.

* CRUD Web Apps: đây là một trong những tính năng mà chính người sử dụng Angular ưu tiên sử dụng nhiều nhất.
* Mobile Apps: Lập trình viên có thể sử dụng Angular kết hợp cùng Phonegap để tạo ra một loại Mobile App  bạn có thể tạo một Mobile Web App.



*Angular và các ngôn ngữ có thể sử dụng*

* CSS3 Animations.
* Chrome Extensions: Đây là cách đơn giản nhất giúp lập trình viên tạo ra Chrome Extension lả việc sử dụng Yeoman- một Chrome Extension Generator.
* Testable JS Apps.
* Firebase Powered Apps.

## 2.1.2 Các đặc trưng cơ bản của Angular là gì?

* Được sử dụng để có thể phát triển dựa trên JavaScript.
* Angular có khả năng tạo các ứng dụng client-side dựa trên mô hình MVC.
* Angular sở hữu khả năng tương thích cao có thể tự động xử lý dễ dàng các mã Javascript sao cho phù hợp với các trình duyệt nhất.
* Khi có mã nguồn mở và miễn phí thì nó sẽ được sử dụng rộng rãi hơn.

Thông thường, kiến trúc của một ứng dụng Angular là dựa trên những ý tưởng liên quan đến Components. Mỗi một ứng dụng Angular thường bắt đầu với những  level trên cùng tên gọi là Root Component.

## NodeJS

NodeJS là một nền tảng được xây dựng trên V8 JavaScript Engine – trình thông dịch thực thi mã JavaScript, giúp xây dựng các ứng dụng web một cách đơn giản và dễ dàng mở rộng.

NodeJS được phát triển bởi Ryan Dahl vào năm 2009 và có thể chạy trên nhiều hệ điều hành khác nhau: OS X, Microsoft Windows, Linux.

Ưu điểm khi sử dụng NodeJS:

* NodeJS được viết bằng JavaScript với cộng đồng người dùng lớn mạnh.
* Tốc độ xử lý nhanh. Nhờ cơ chế xử lý bất đồng độ (non-blocking), NodeJS có thể xử lý hàng ngàn kết nối cùng lúc mà không gặp bất cứ khó khăn nào.
* Dễ dàng mở rộng.

## MongoDB

MongoDB là một database hướng tài liệu (document), một dạng NoSQL database. Vì thế, MongoDB sẽ tránh cấu trúc table-based của relational database để thích ứng với các tài liệu như JSON có một schema rất linh hoạt gọi là BSON. MongoDB sử dụng lưu trữ dữ liệu dưới dạng Document JSON nên mỗi một collection sẽ các các kích cỡ và các document khác nhau. Các dữ liệu được lưu trữ trong document kiểu JSON nên truy vấn sẽ rất nhanh.

Ưu điểm của MongoDB:

* Dữ liệu lưu trữ phi cấu trúc, không có tính ràng buộc, toàn vẹn nên tính sẵn sàng cao, hiệu suất lớn và dễ dàng mở rộng lưu trữ.
* Dữ liệu được caching (ghi đệm) lên RAM, hạn chế truy cập vào ổ cứng nên tốc độ đọc và ghi cao.

Nhược điểm của MongoDB:

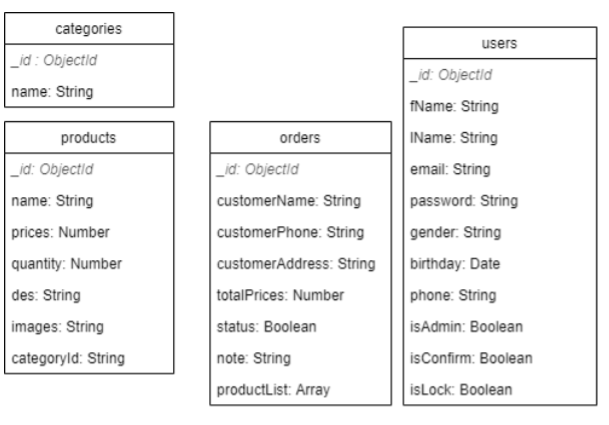
* Không ứng dụng được cho các mô hình giao dịch nào có yêu cầu độ chính xác cao do không có ràng buộc.
* Không có cơ chế transaction (giao dịch) để phục vụ các ứng dụng ngân hàng.
* Dữ liệu lấy RAM làm trọng tâm hoạt động vì vậy khi hoạt động yêu cầu một bộ nhớ RAM lớn.
* Mọi thay đổi về dữ liệu mặc định đều chưa được ghi xuống ổ cứng ngay lập tức vì vậy khả năng bị mất dữ liệu từ nguyên nhân mất điện đột xuất là rất cao.

## High level design

### Database

#### Thiết kế Database

Hệ cơ sở dữ liệu gồm 4 bảng chính: Order, User, Product, Categories. Database được thiết kế trên MongoDB nên sẽ không có mối quan hệ giữa các bảng.



Hình 1: Database

#### Các chức năng trong bảng

|  |  |  |
| --- | --- | --- |
| STT | Tên bảng | Chức năng |
| 1 | users | Lưu thông tin của người dùng, bao gồm các thông tin tài khoản, thông tin cá nhân, vai trò của người dùng. |
| 2 | products | Lưu thông tin các sản phẩm bao gồm tên sản phẩm, đường dẫn hình ảnh, giá sản phẩm, phân loại sản phẩm, số lượng tồn tại trong kho, mô tả. |
| 3 | orders | Lưu thông tin danh sách các đơn đặt hàng của chương trình, bao gồm các thông tin chi tiết của đơn hàng, tổng giá tiền, trạng thái, ghi chú, một list danh sách và số lượng mua, thông tin người mua. |
| 4 | categories | Lưu tên của danh mục sản phẩm.q |

Bảng 1: Mô tả chức năng

### Backend

#### Express

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. Expressjs hỗ trợ các method HTTP và midleware tạo ra API vô cùng mạnh mẽ và dễ sử dụng.

Các chức năng chính của Express:

* Thiết lập các lớp trung gian để trả về các HTTP request.
* Define router cho phép sử dụng với các hành động khác nhau dựa trên phương thức HTTP và URL.
* Cho phép trả về các trang HTML dựa vào các tham số.

#### Mongoose

Mongoose là một framework JavaScript thường được sử dụng trong ứng dụng Node.js với database là MongoDB.Mongoose là một Object Document Mapper (ODM). Điều này có nghĩa là Mongoose cho phépđịnh nghĩa các object với một schema được định nghĩa rõ ràng, được ánh xạ tới một MongoDB document.

Mongoose cung cấp một số lượng lớn các chức năng cho việc tạo ra và làm việc với các schema. Mongoose hiện có 8 SchemaTypes:

1. String
2. Number
3. Date
4. Buffer
5. Boolean
6. Mixed
7. ObjectId
8. Array

Mỗi loại dữ liệu cho phép ta xác định:

* Giá trị mặc định.
* Hàm xác thực.
* Cho biết một trường bắt buộc.
* Hàm get cho phép thao tác dữ liệu trước khi nó được trả về như một object.
* Hàm set cho phép thao tác dữ liệu trước khi nó được lưu vào cơ sở dữ liệu.
* Tạo indexes cho phép dữ liệu được nạp nhanh hơn.

#### Bcryptjs

Bcrypt là một chức năng mã hóa mật khẩu dựa trên các thuật toán mã hóa Blowfish. Bcrypt là một nền tảng tập tin tiện ích mã hóa chéo. Passphrase phải từ 8 đến 56 ký tự và được mã hóa trong nội bộ một khóa 448 bit. Bcrypt là một thuật toán mã hóa một chiều. Ta không thể lấy lại mật khẩu khi đã biết chuỗi mật khẩu trong dữ liệu databse trước đó mà bất kỳ46 ai tấn công vào để đánh cắp. Bcrypt nó tự động tạo ra các chuỗi mã hóa ngẫu nhiên sau mỗi lần tạo ra. Điều này làm cho nó khó khăn hơn cho việc mã hóa và tìm ra mật khẩu cho những cuộc tấn công. Nó cũng có thể được thực hiện để đi chậm hơn, giúp chống lại các cuộc tấn công.

#### Jsonwebtoken

JWT là một phương tiện đại diện cho các yêu cầu chuyển giao giữa hai bên Client – Server, các thông tin trong chuỗi JWT được định dạng bằng JSON. Trong đó chuỗi Token phải có 3 phần là header, phần payload và phần signature được ngăn bằng dấu “.”. Ta nên dùng JSON Web Token trong các trường hợp:

* Authentication: Đây là trường hợp phổ biến nhất thường sử dụng JWT. Khi người dùng đã đăng nhập vào hệ thống thì những request tiếp theo từ phía người dùng sẽ chứa thêm mã JWT. Điều này cho phép người dùng được cấp quyền truy cập vào các url, service, và resource mà mã Token đó cho phép.
* Trao đổi thông tin: JSON Web Token là 1 cách thức khá hay để truyền thông tin an toàn giữa các thành viên với nhau, nhờ vào phần signature của nó. Phía người nhận có thể biết được người gửi là ai thông qua phần signature. Và chữ ký được tạo ra bằng việc kết hợp cả phần header, payload lại nên thông qua đó ta có thể xác nhận được chữ ký có bị giả mạo hay không.

#### Multer

Multer là một middleware được sử dụng để xử lý multipart/form-data, chủ yếu được sử dụng để upload file.

#### Body-Parser

BodyParser trả về một function hoạt động như một middleware. Chức năng lắng nghe trên req.on (\'data\') và xây dựng req.body từ các đoạn dữ liệu mà nó nhận được.

### Front-end

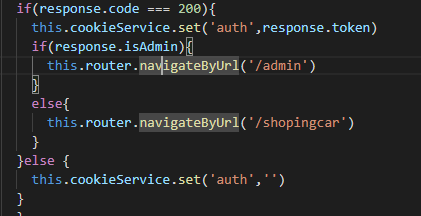
#### Angular routing

## Angular routing

Angular  cho phép bạn chia ứng dụng thành các view nhỏ và bạn có thể điều hướng giữa các view đó. Routing cho phép người dùng gọi tới các component khác nhau dựa trên URL mà họ gõ trên trình duyệt. Nếu mở file package.json bạn sẽ thấy nó được khai báo trong dependencies. cho phép chúng ta điều hướng các URL tới các hàm/phương thức/lớp/controller nào đó trong ứng dụng, đây là tính năng có trong hầu hết các web framework phổ biến ngày nay.

Trong Angular thì một URL sẽ được điều hướng tới một lớp Component, tức là khi người dùng trỏ URL nào vào trong trình duyệt thì Angular sẽ hiển thị template của lớp Component được điều hướng tương ứng.

Ví dụ



#### CookieService

Đây là 1 thư viện giúp lấy địa chỉ cookie của người dùng

npm install ngx-cookie-service --save

Thêm vào mô-đun của bạn: import { CookieService } from 'ngx-cookie-service';

Thêm CookieService cho các nhà cung cấp của mô-đun.

Tiêm nó vào nhà xây dựng của bạn.

Sử dụng cookie.get(nameOfCookie) để nhận cookie cụ thể, sử dụng cookie.set(nameOfCookie,cookieValue) để thêm cookie mới



#### Dependency Injection

Bao gồm 3 phần thực hiện:

Injector, gồm các API dùng để inject các dependency và làm chúng hoạt động.

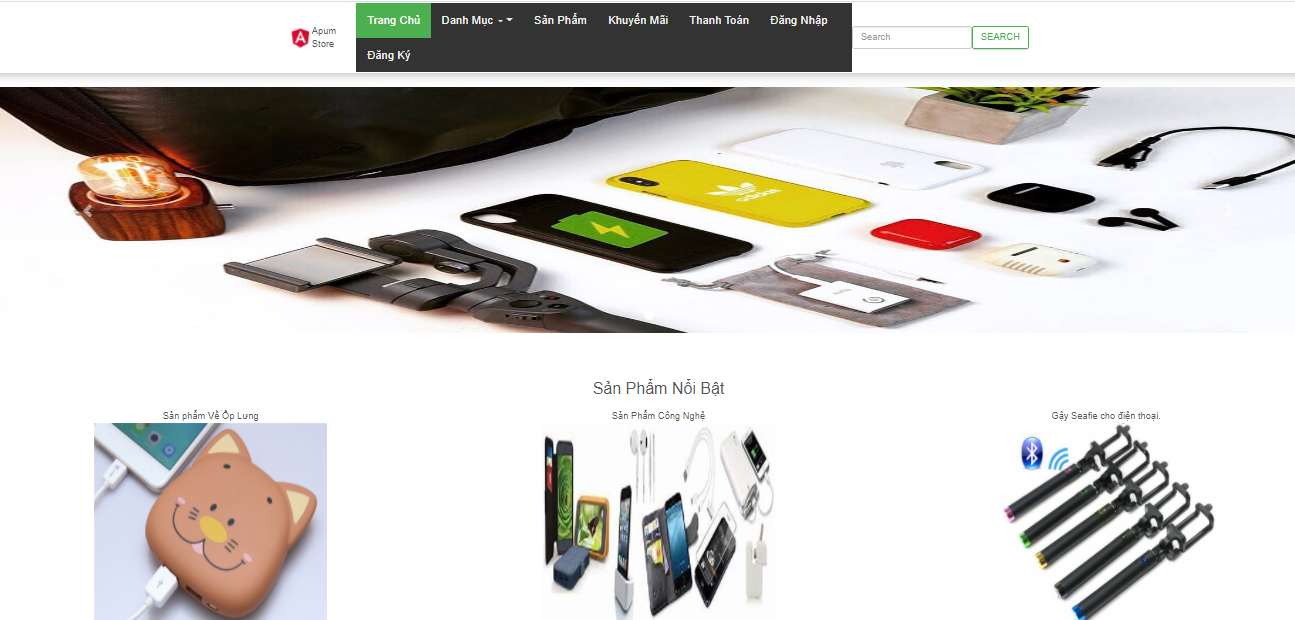
Bindings, giúp định nghĩa các dependency dưới một tên biến để sử dụng.

Cuối cùng, các dependency thực sẽ được khởi tạo và có thể được inject. Một object có thể được truyền thông qua constructor nên ta chỉ cần sử dụng injector view để thực hiện việc inject đơn giản.

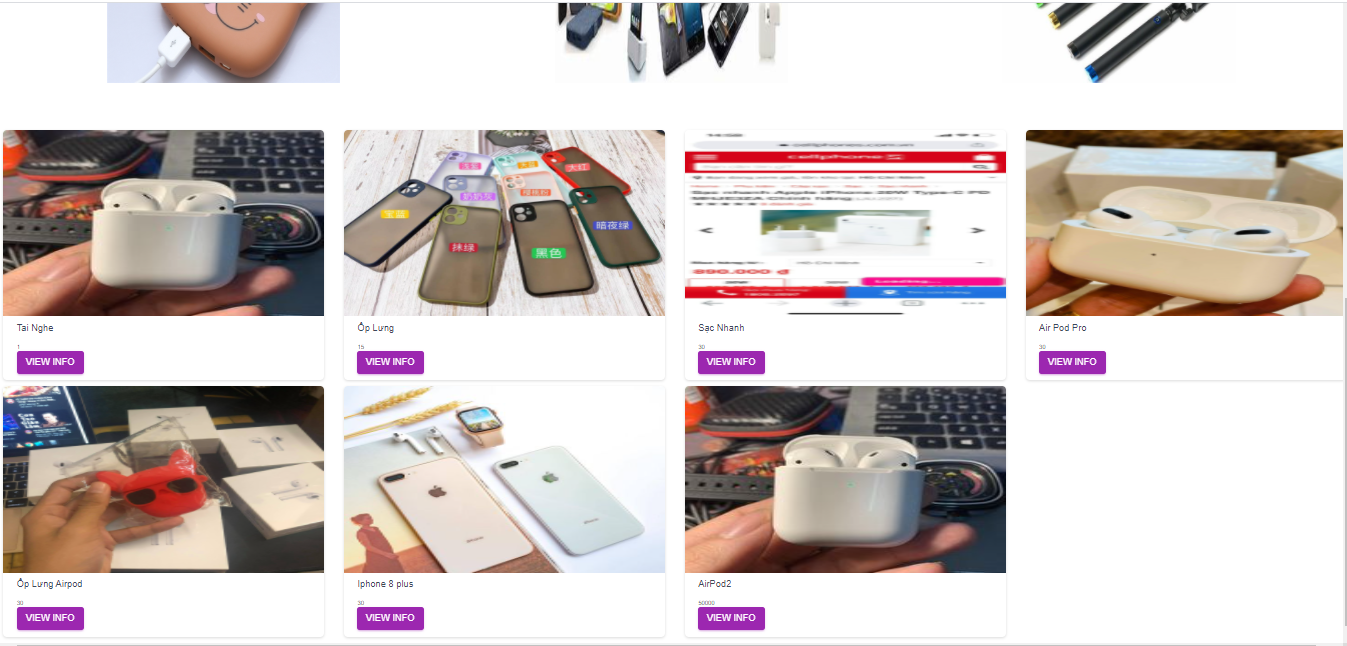
## Detail design

### Front-end

#### Màn hình Trang Home



Hình 4: Trang chủ

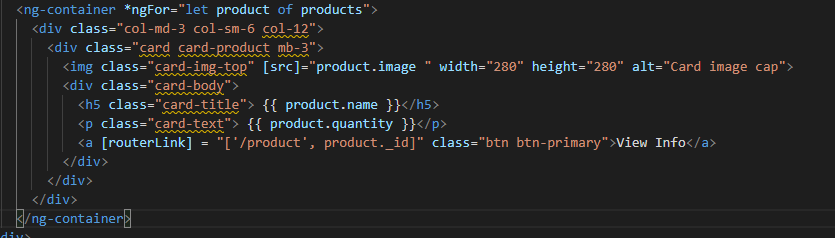


Hình 5: Trang Chủ http://localhost:4200/#/home

Màn hình này load danh mục sản phẩm,đăng nhập,đăng ký

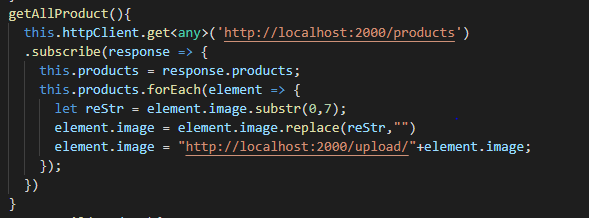
Đồng thời hiển thị hình ảnh nổi bật của sản phẩm, tên,giá cả từ Api backend

load dữ liệu lên giao diện từ trang html home.component.html



Hình 6: load dữ liệu

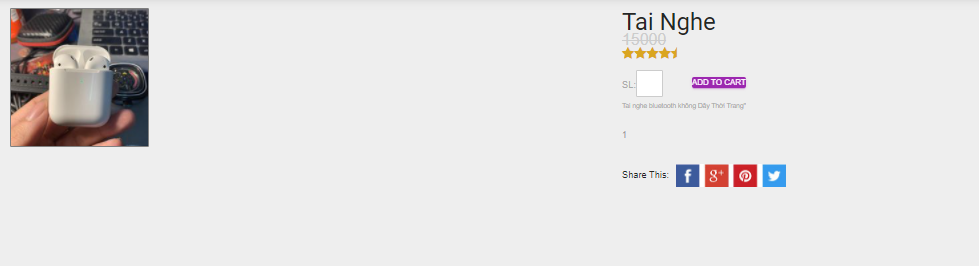
Trong phần home.component sẽ được thực thi, gọi API tới backend và nhận dữ liệu, để thay đổi các state, các reducer này được lưu trữ trong store để phần view có thể truy xuất tới.



Hình 7:getAllProduct

Hàm productDetailReducer thực hiện thay đổi các state:

#### Màn hình chi tiết sản phẩm

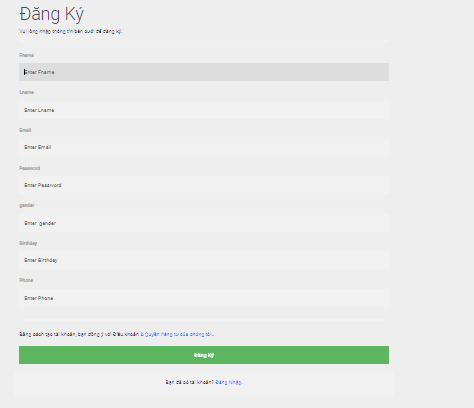


Hình 8:giao diện chi tiết sản phẩm

http://localhost:4200/#/product

Màn hình này hiển thị thông tin chi tiết của một sản phẩm như hình ảnh, tên, giá cả, mô tả, số lượng trong kho, cho phép người dùng thay đổi số lượng sản phẩm và lựa chọn thêm vào giỏ hàng.

#### Giao Diện Đăng Ký Tài Khoản



Hình 9:giao diện đăng ký user

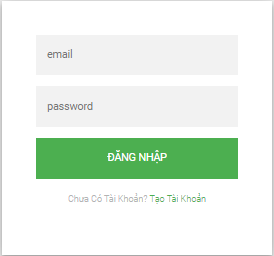
http://localhost:4200/#/register

Nhập thông tin user bắt Post dữ liệu từ api backend





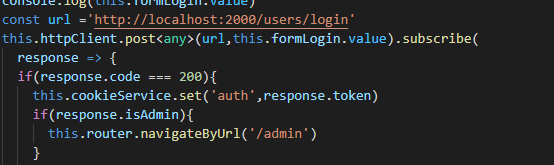
#### Giao Diện Đăng Nhập



Hình 9: Đăng Nhập

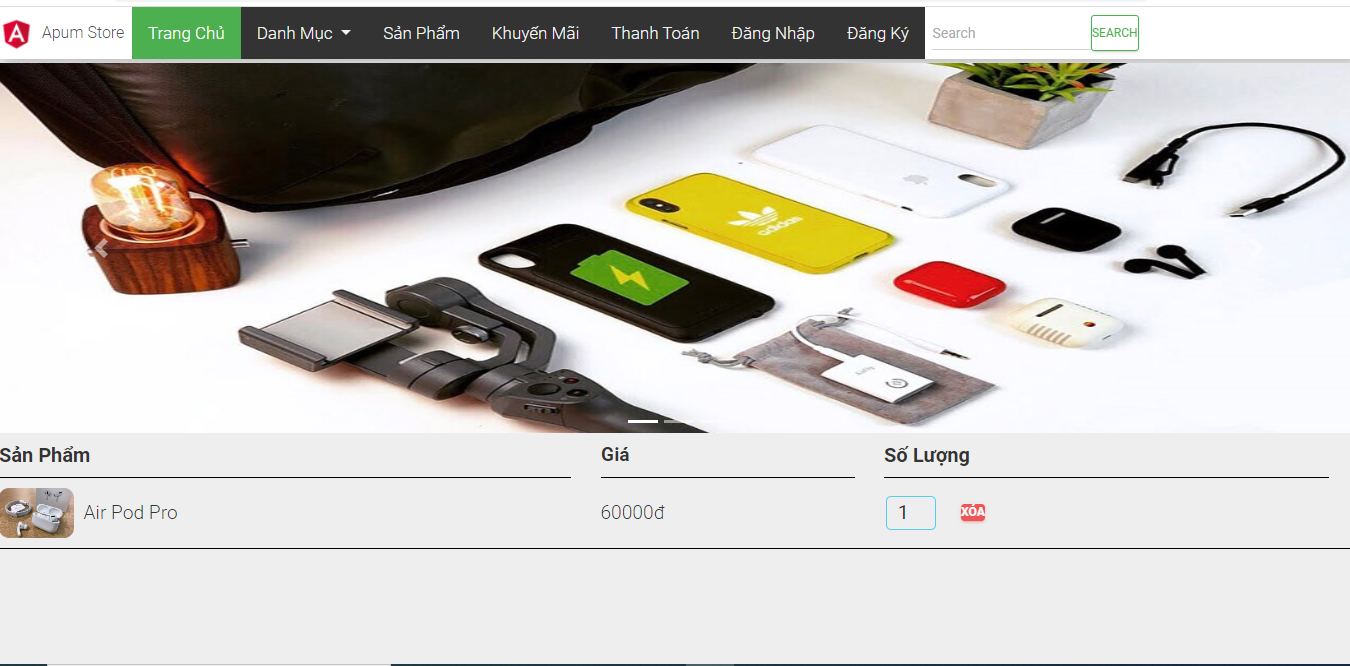
http://localhost:4200/#/login

Nhập thông tin đăng nhập xử lý lấy token từ api trả ra trang



Sử dụng cookieservice để xét token đăng nhập từ api

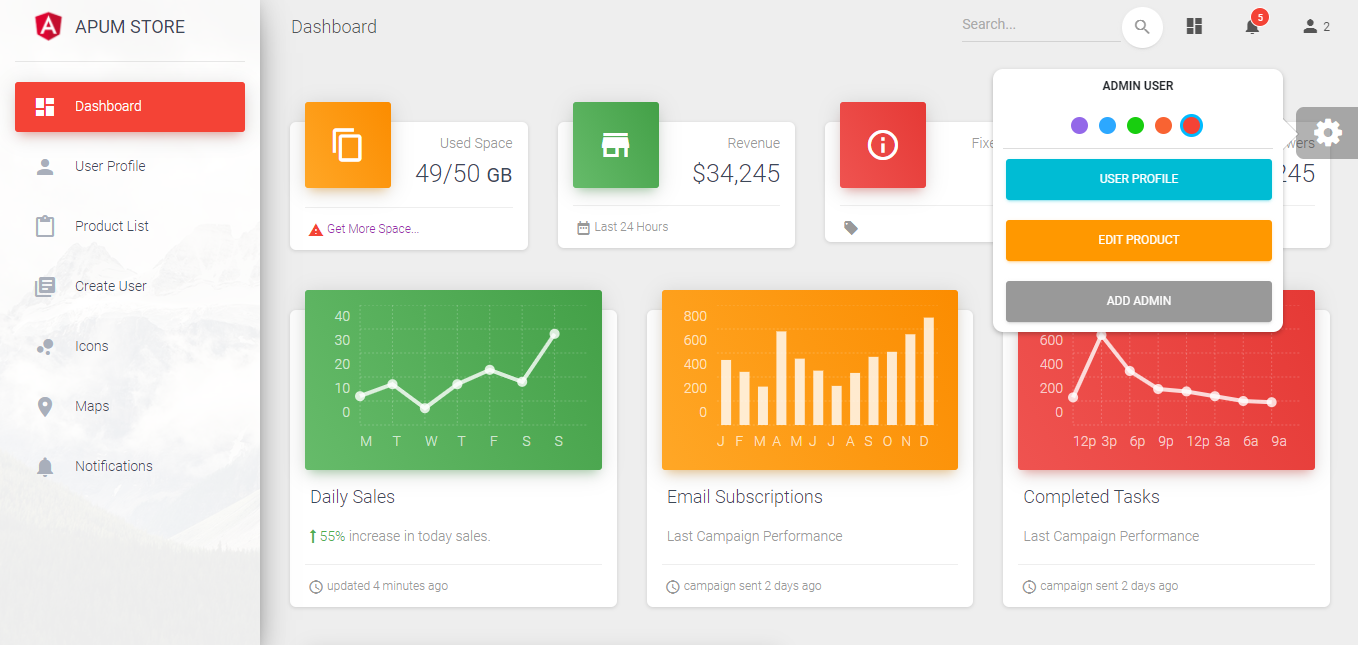
#### Trang giỏ hàng



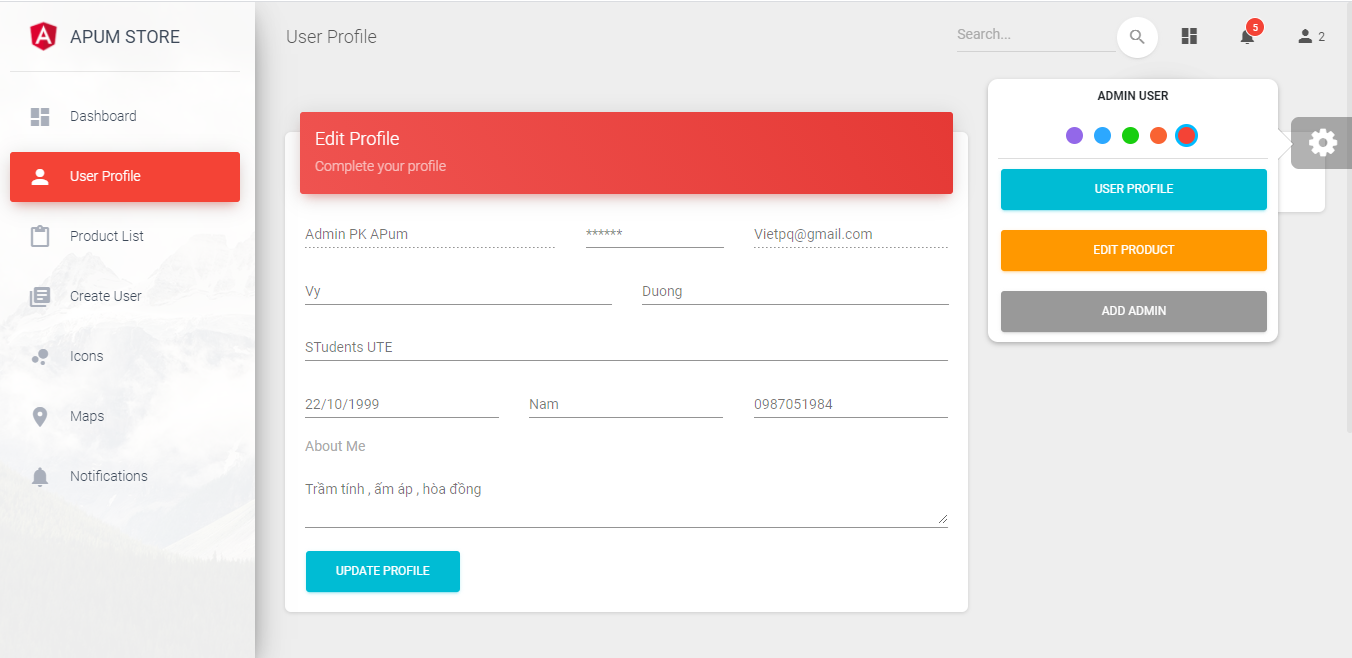
Hình 9: Đăng Nhập

Load thông tin sản phẩm đã được thêm vào giỏ

#### Màn hình trang admin

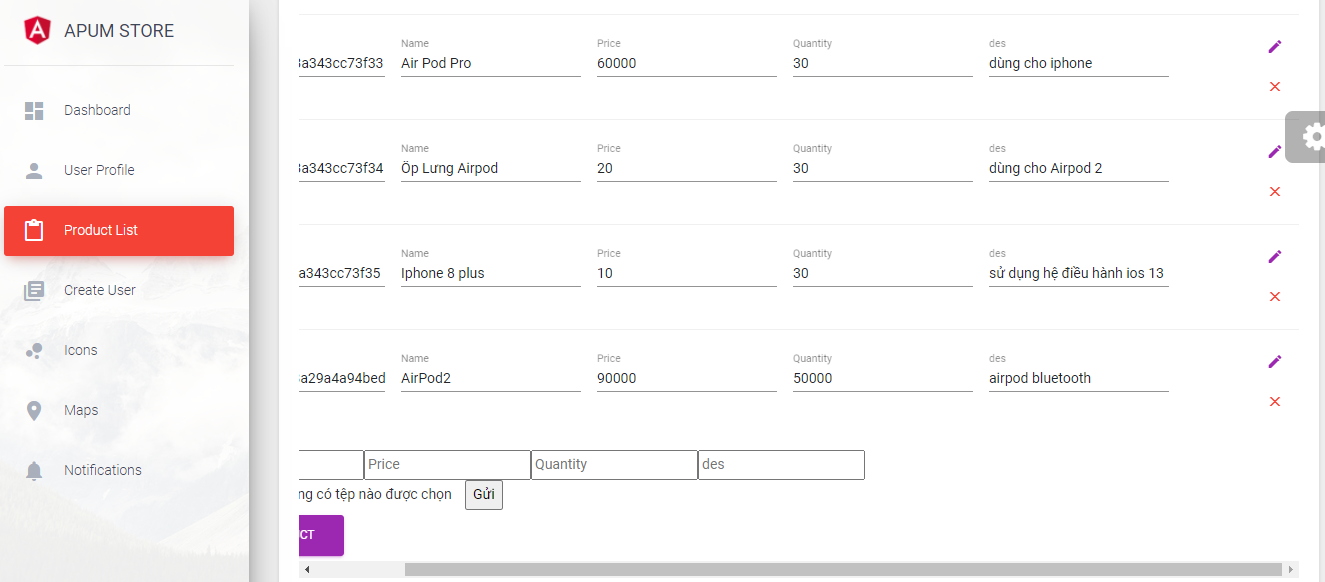


Hình 10: Trang admin http://localhost:4200/#/admin



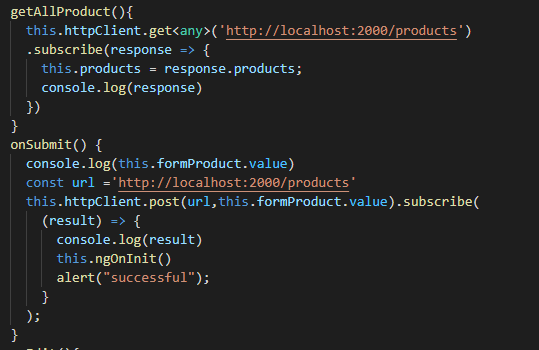
Hình 11: Trang admin http://localhost:4200/#/admin

Load thông tin admin đăng nhập và chỉnh sửa thông tin

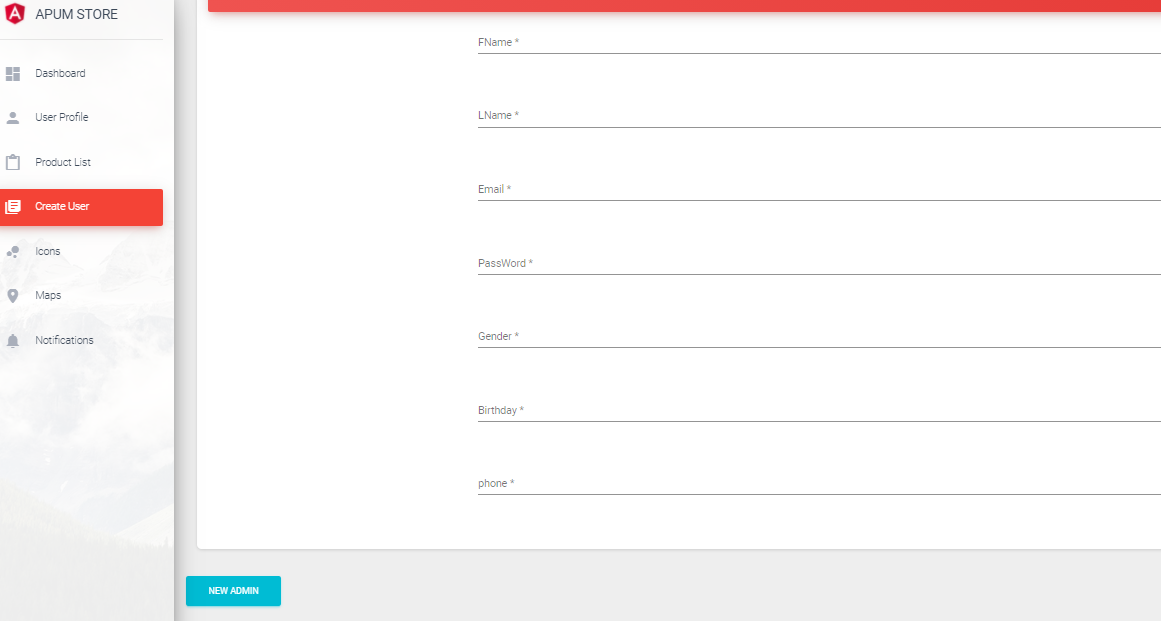


Hình 12: Trang admin http://localhost:4200/#/admin

Trang load thông tin sản phẩm từ trang, thêm sửa xóa thông tin sản phẩm







Hình 13: Trang admin

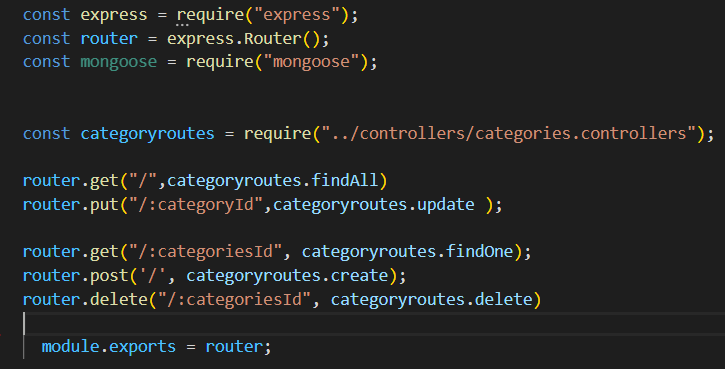
Chức Năng tạo admin

#### Định tuyến API

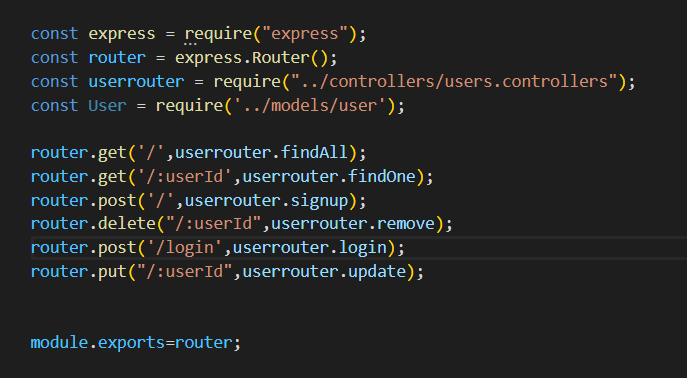
* Product



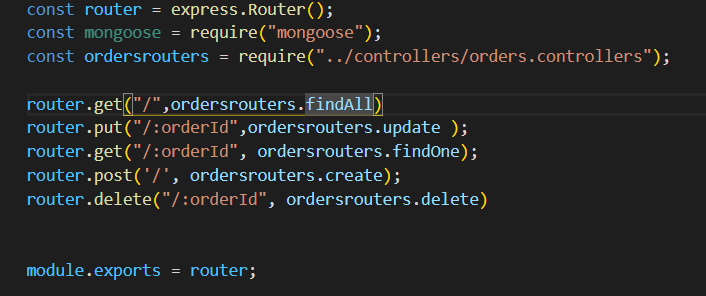
* Categories



* User



* Order



/users: được sử dụng cho request CRUD người dùng.

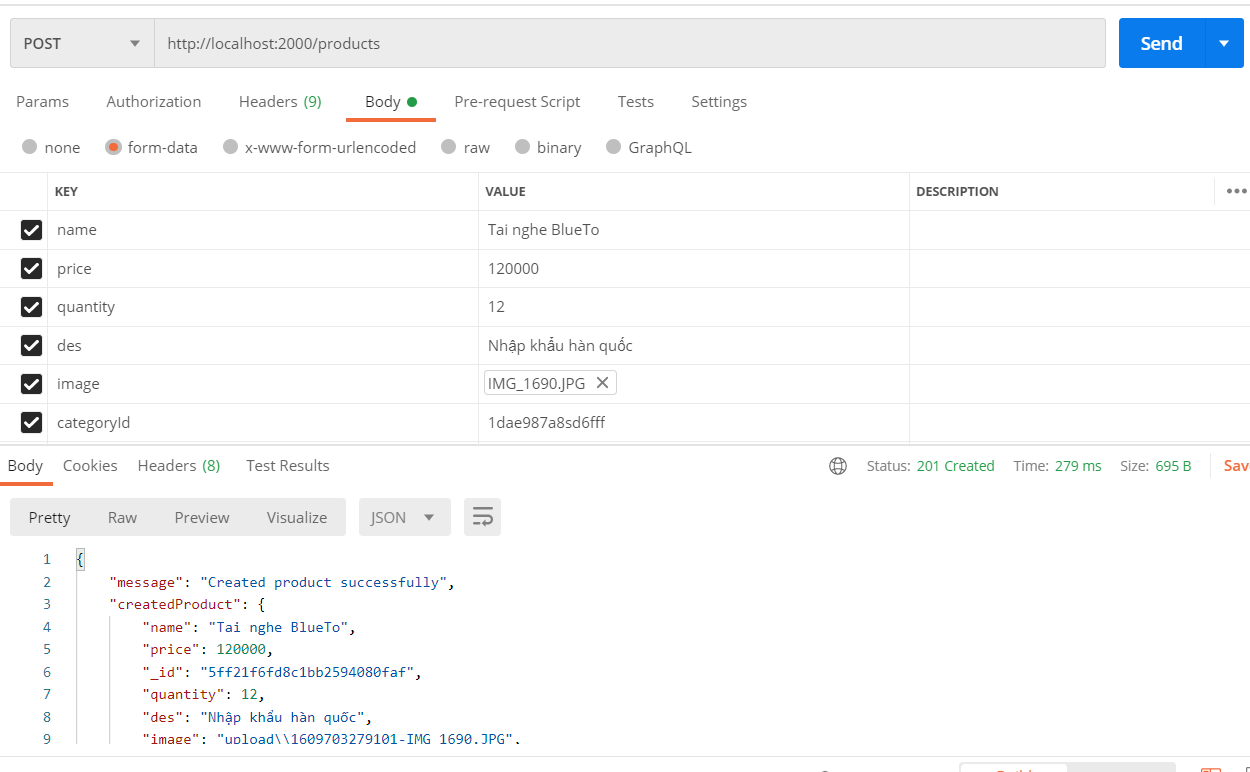
/products: được sử dụng cho request CRUD sản phẩm.

/categories: được sử dụng cho request CRUD danh mục.

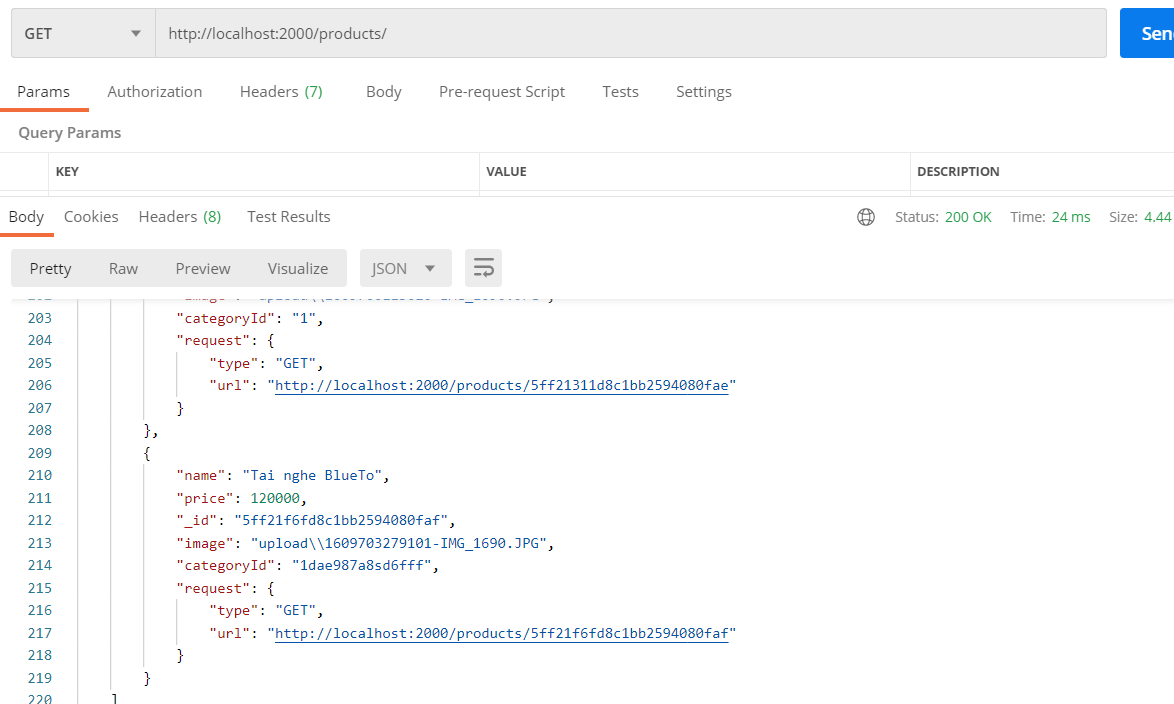
/orders: được sử dụng cho request CRUD hóa đơn.

#### Product

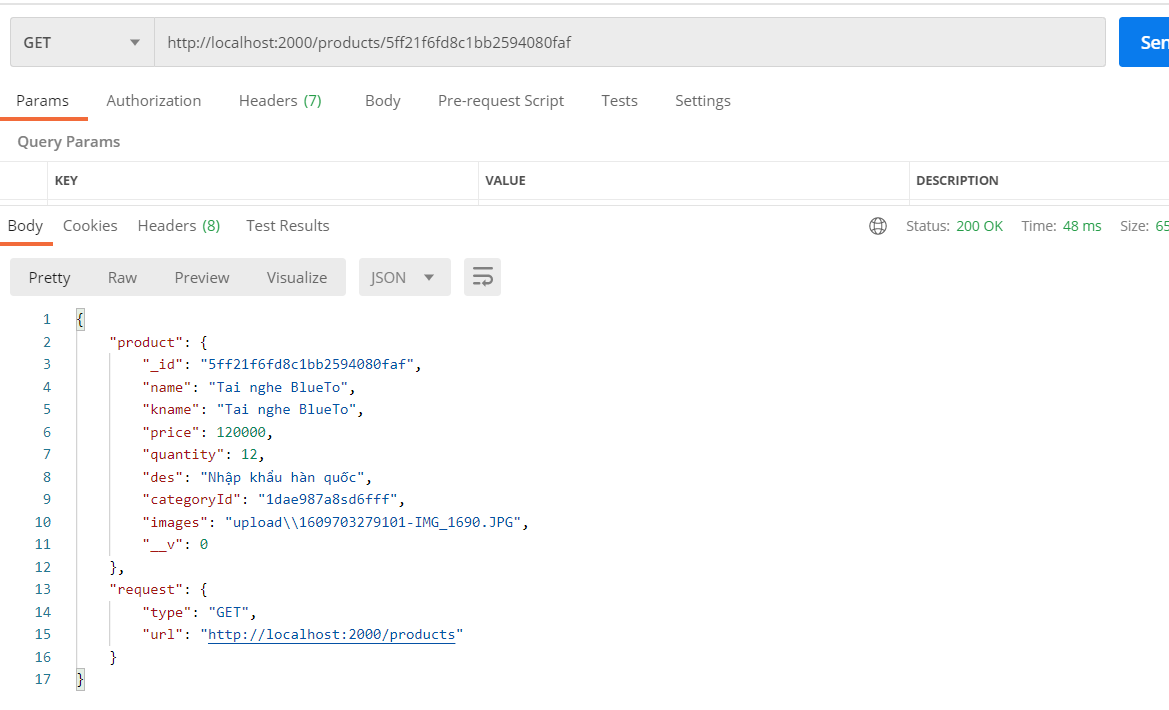
* Tạo sản phẩm mới



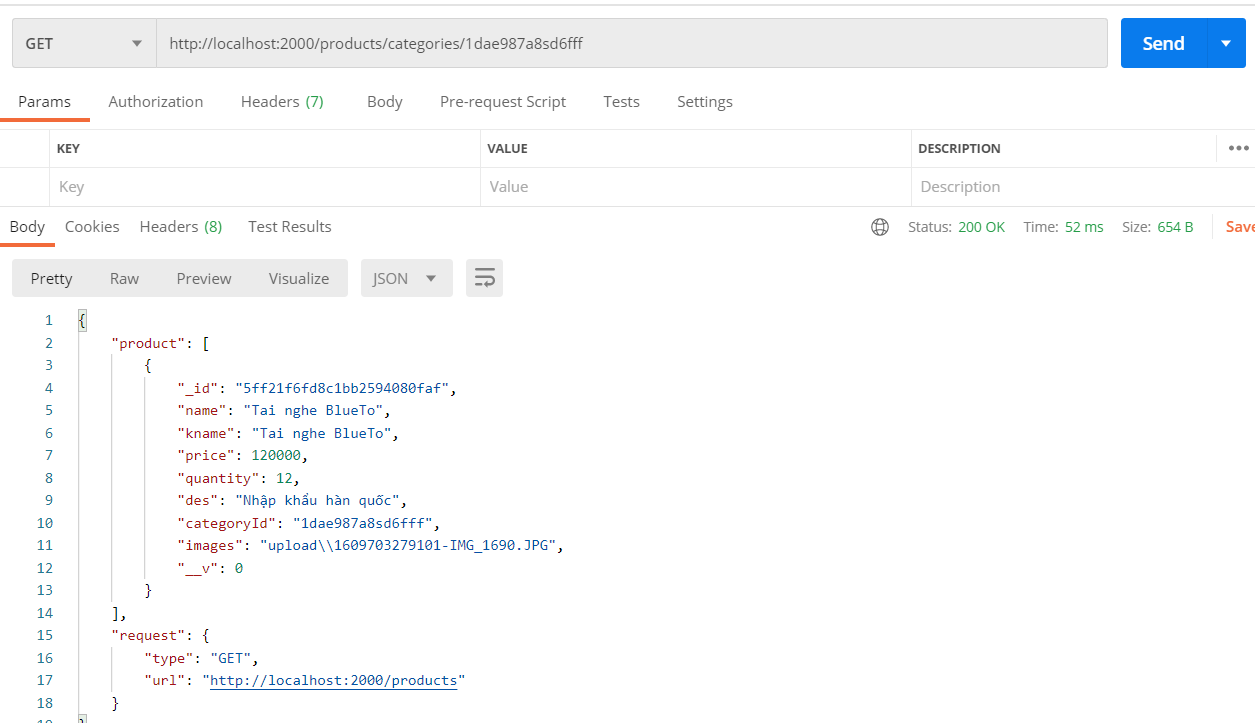
* Lấy hết sản phẩm



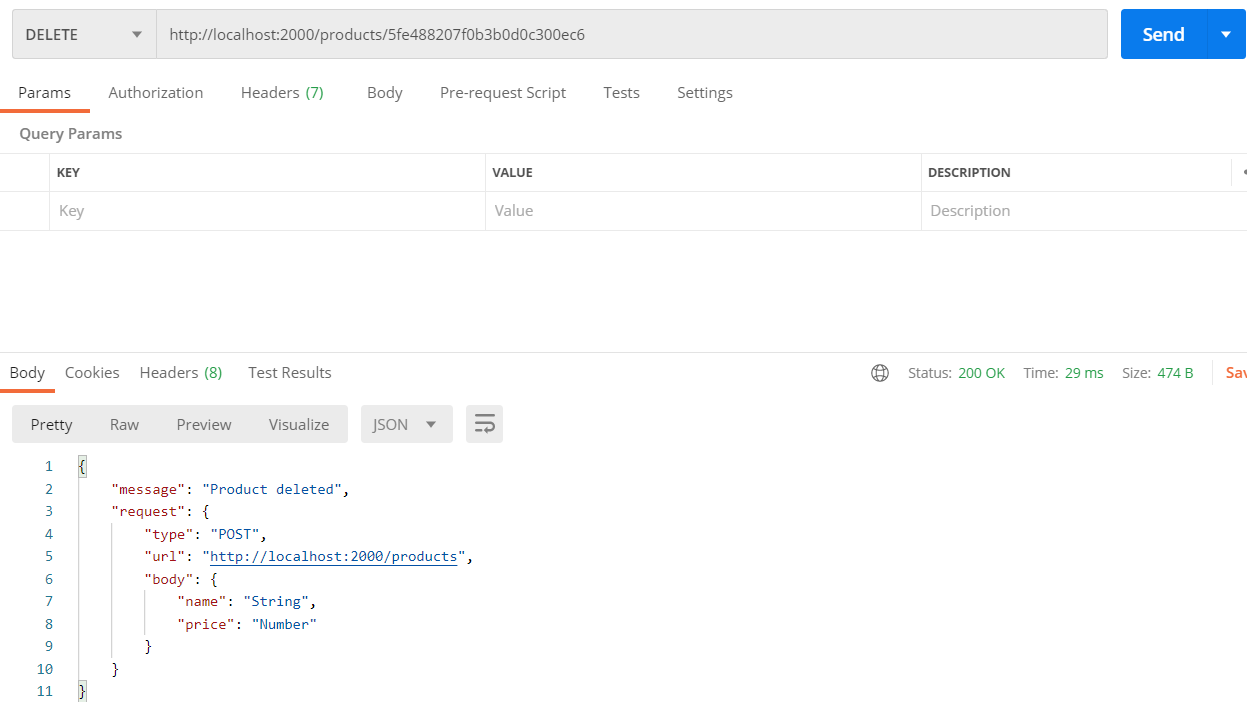
* Lấy sản phẩm theo ID



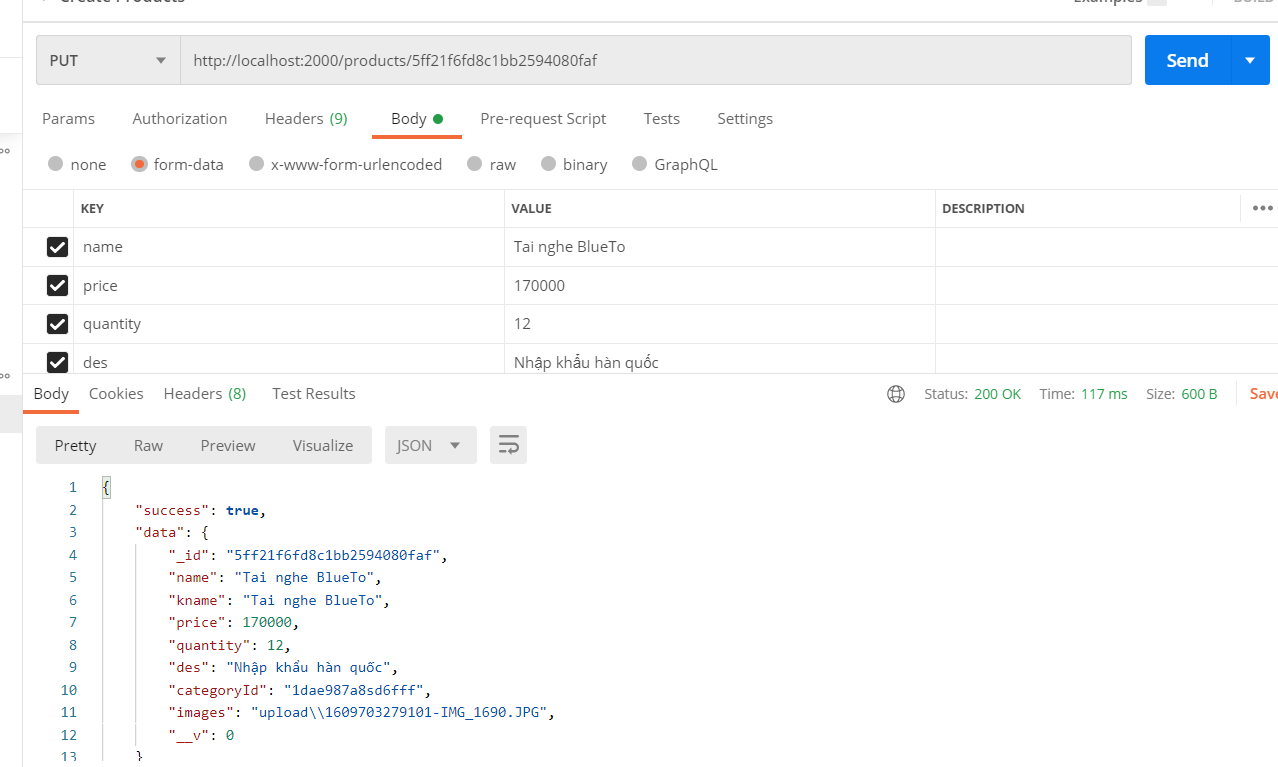
* Lấy sản phẩm theo Idcategory



* Xóa sản phẩm

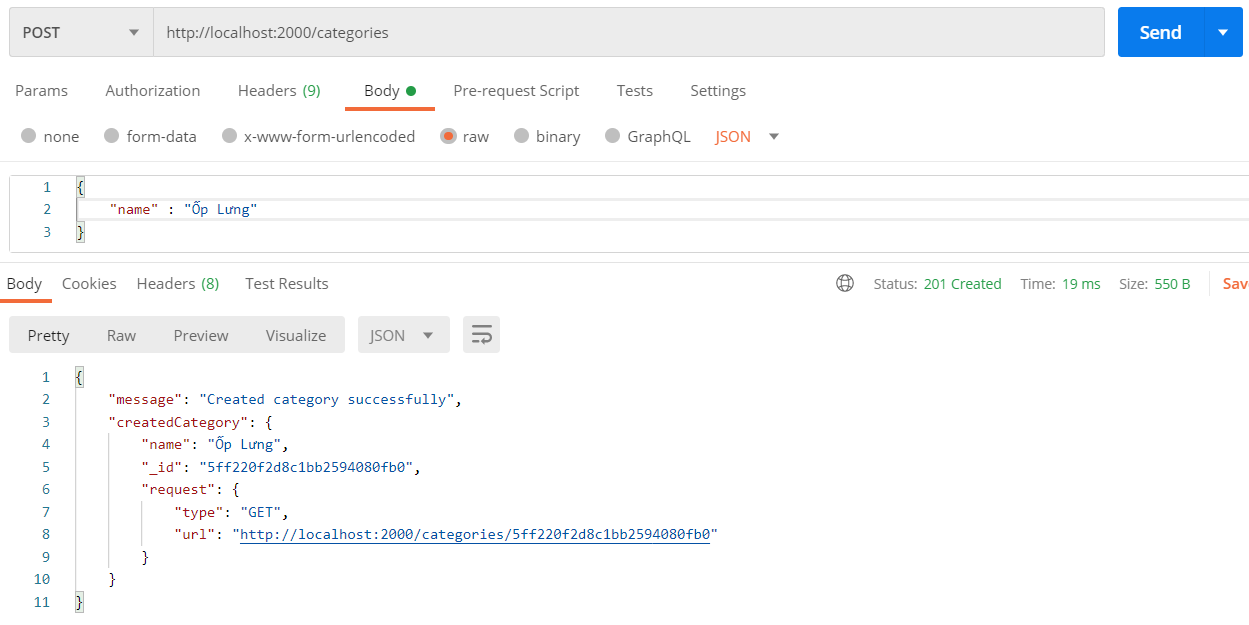


* Cập nhật sản phẩm

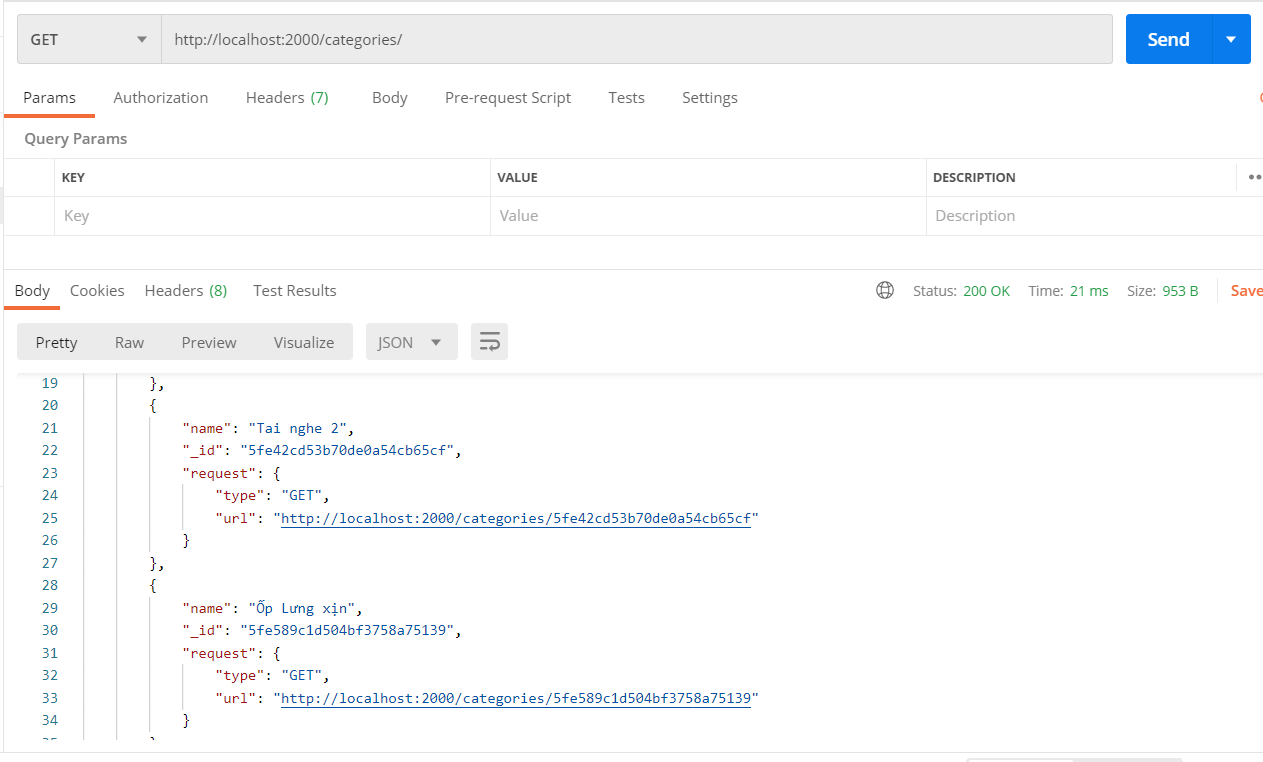


#### Category

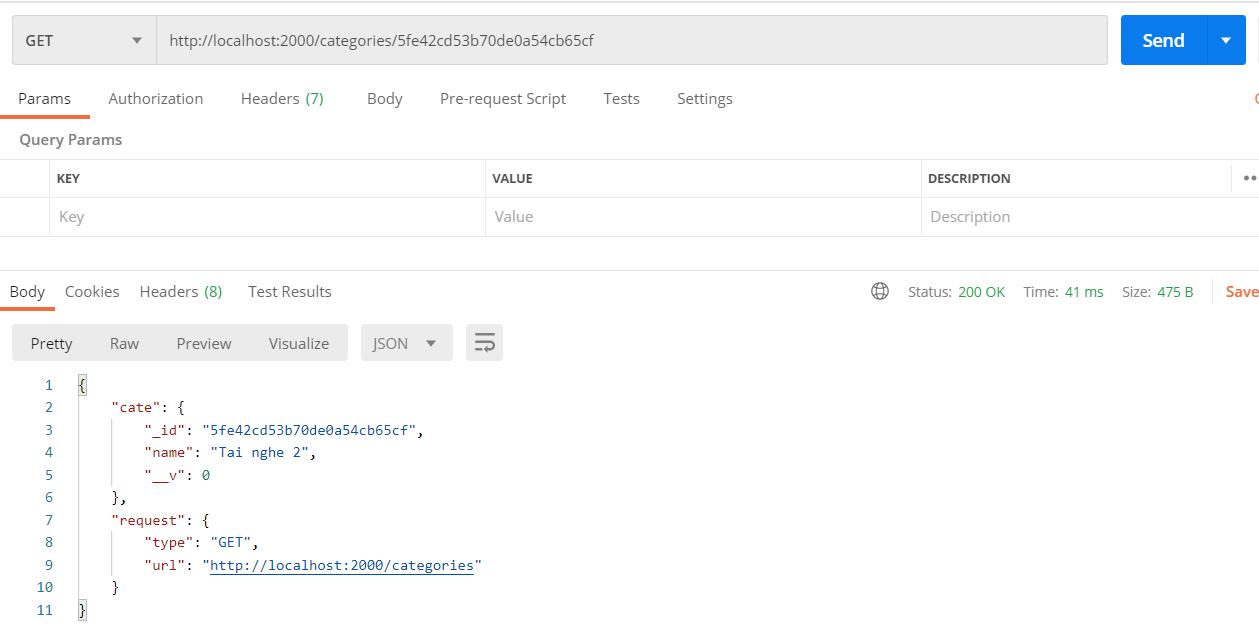
* Tạo danh mục mới



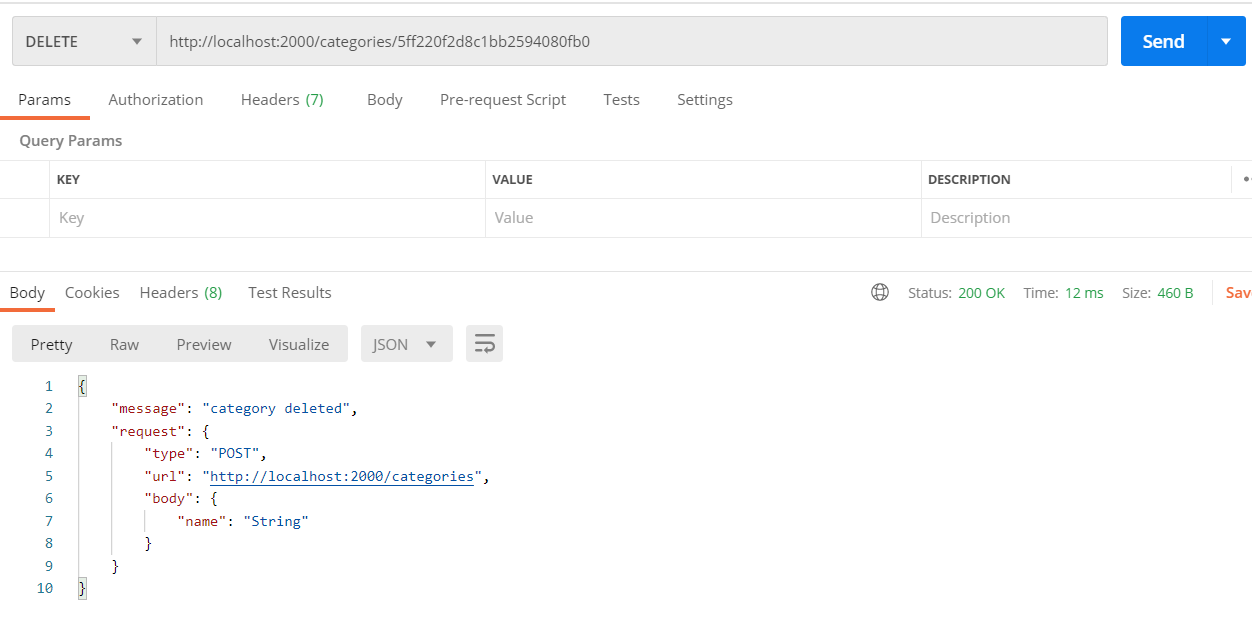
* Lấy hết danh mục



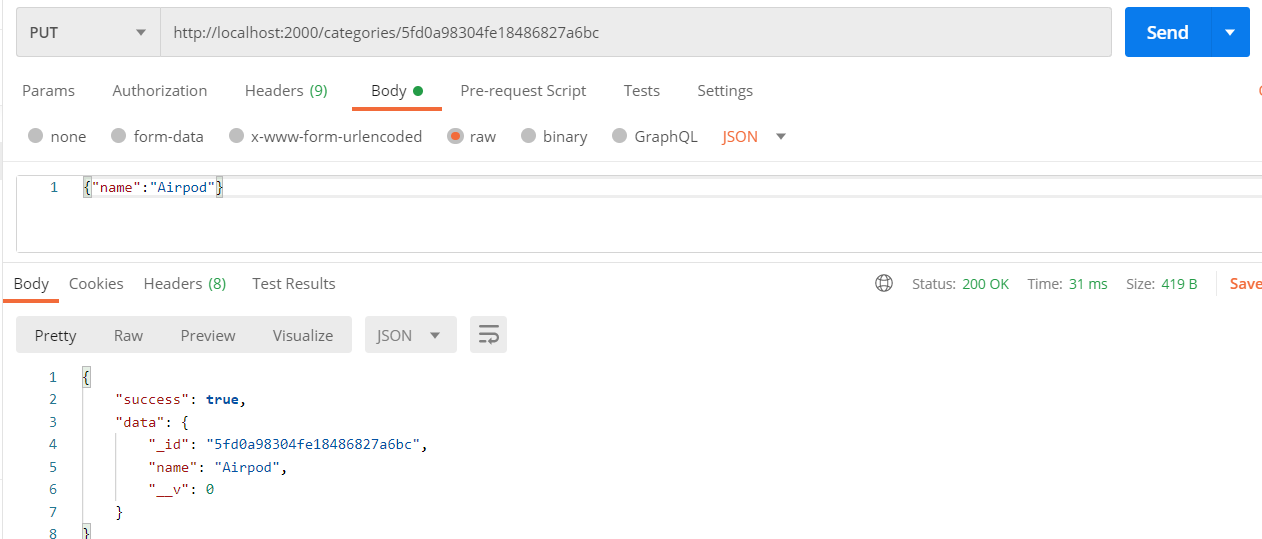
* Lấy sản phẩm ID



* Xóa danh mục

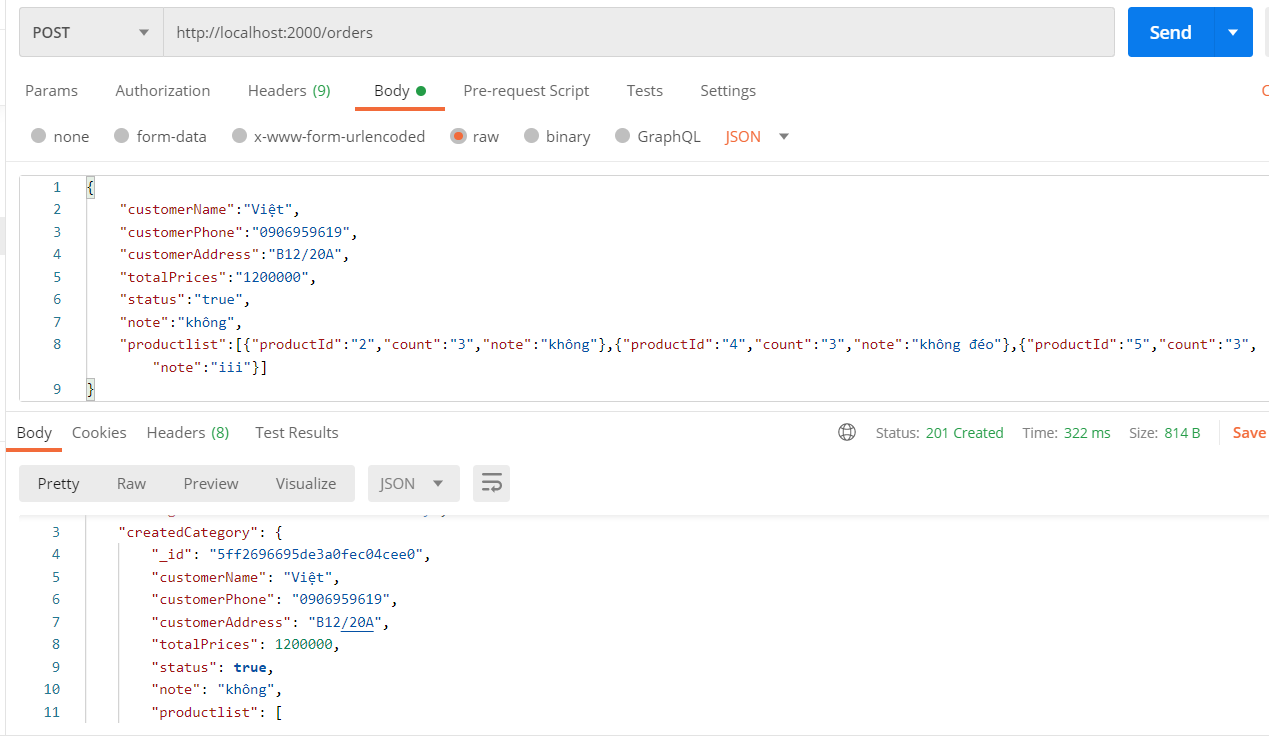


* Cập nhật danh mục

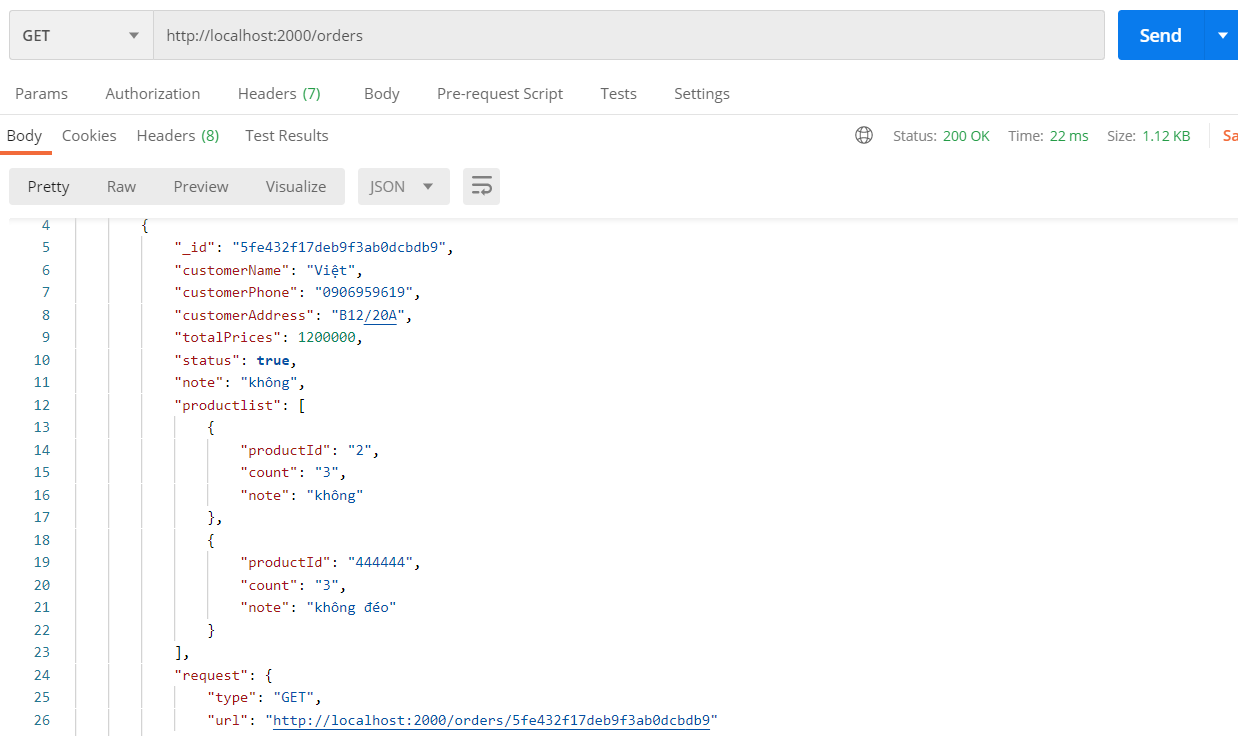


#### Order

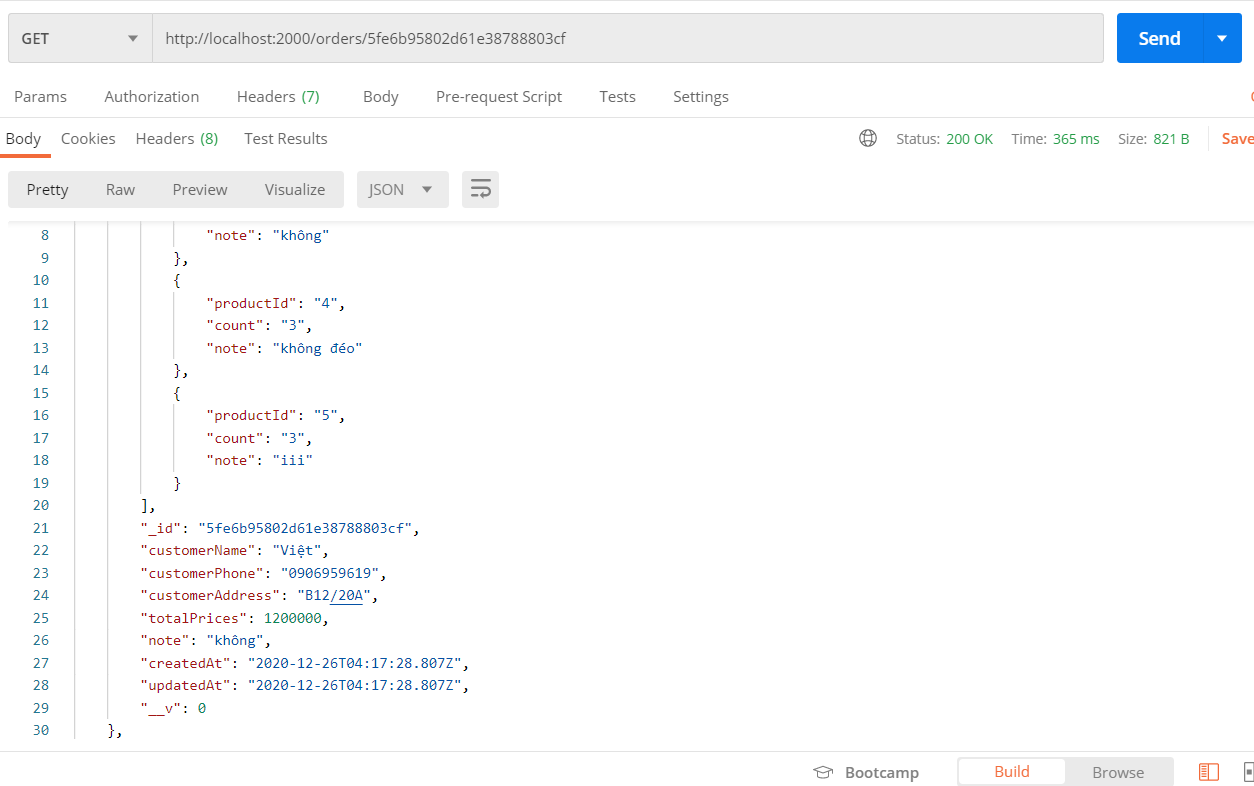
* Tạo hóa đơn mới



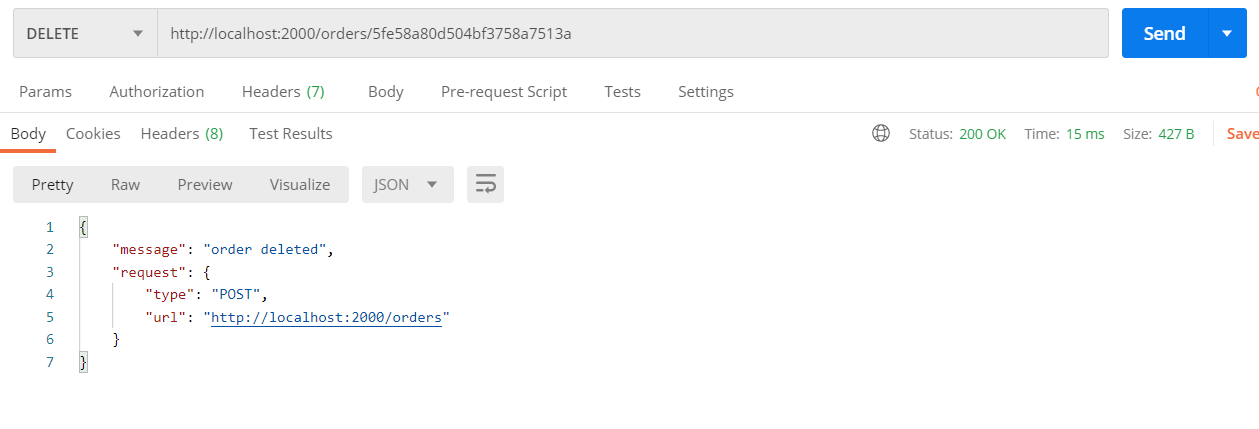
* Lấy hết hóa đơn



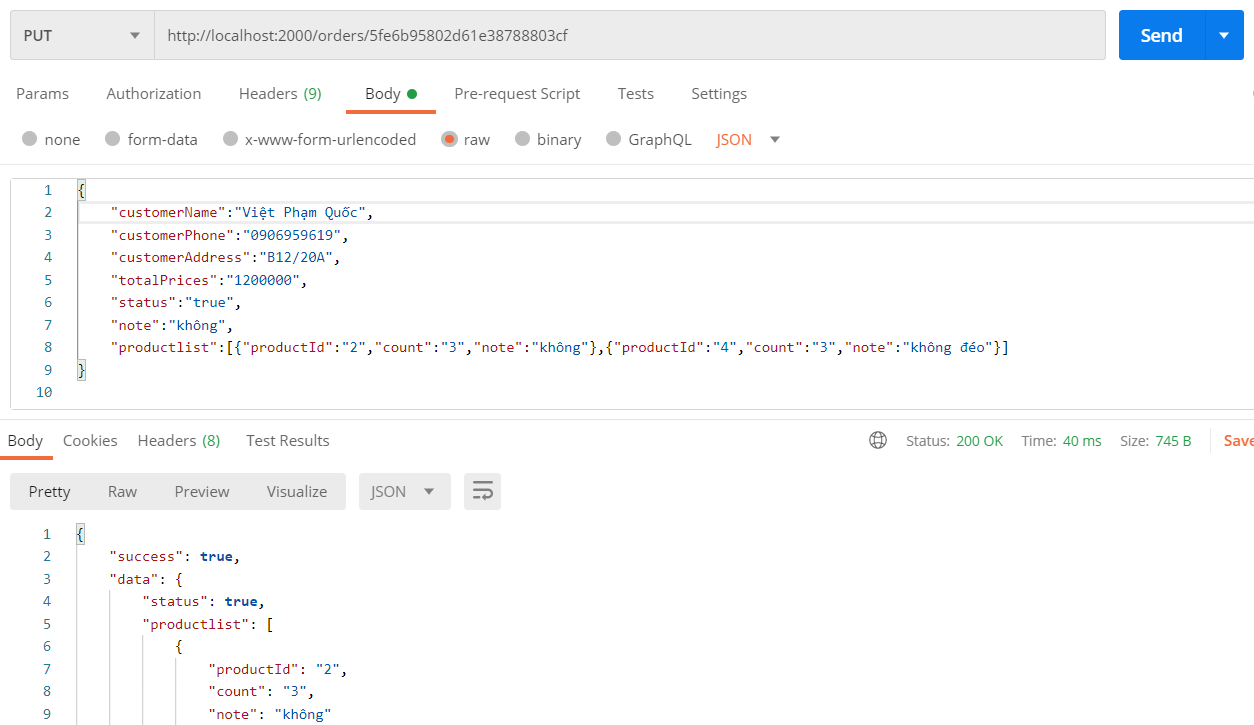
* Lấy hóa đơn theo ID



* Xóa hóa đơn

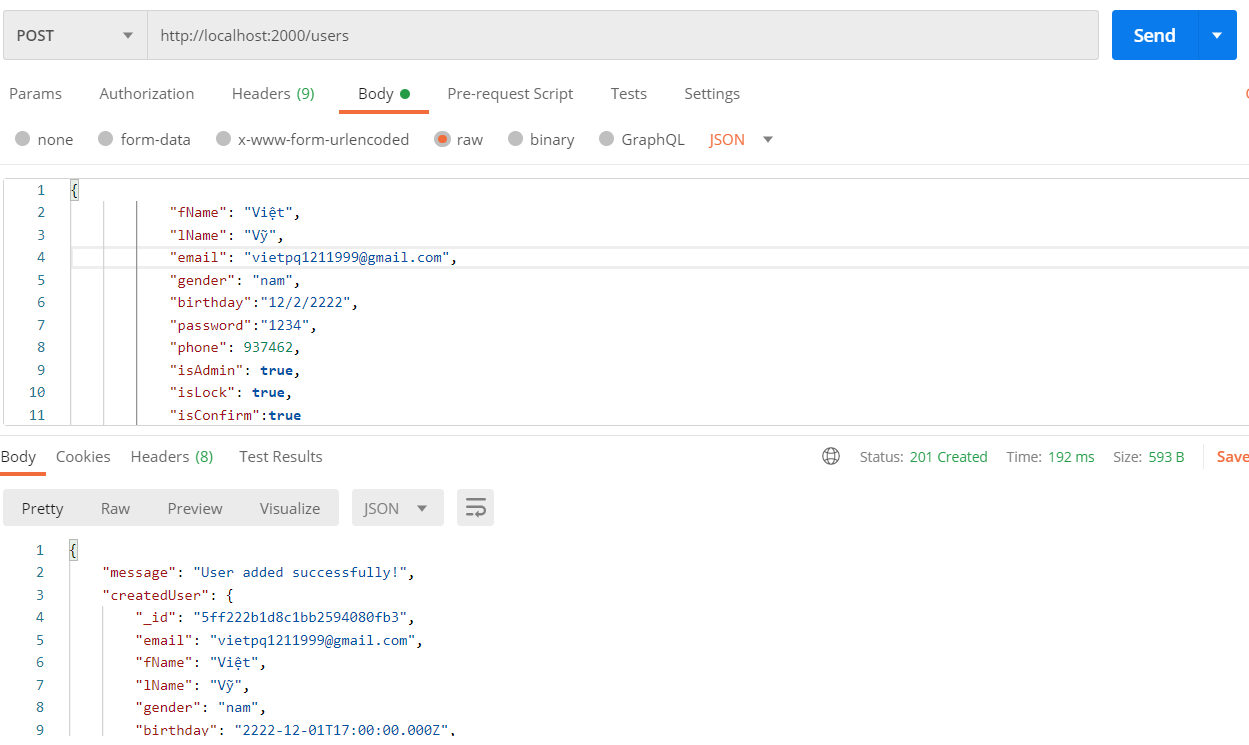


* Cập nhật hóa đơn

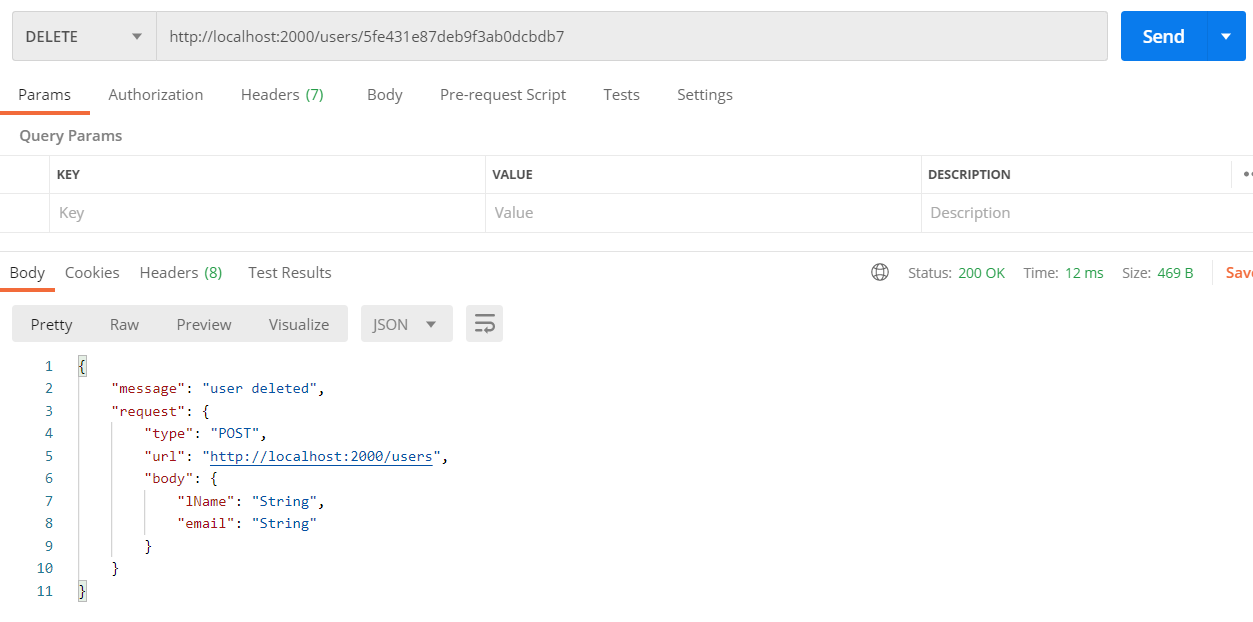


#### User

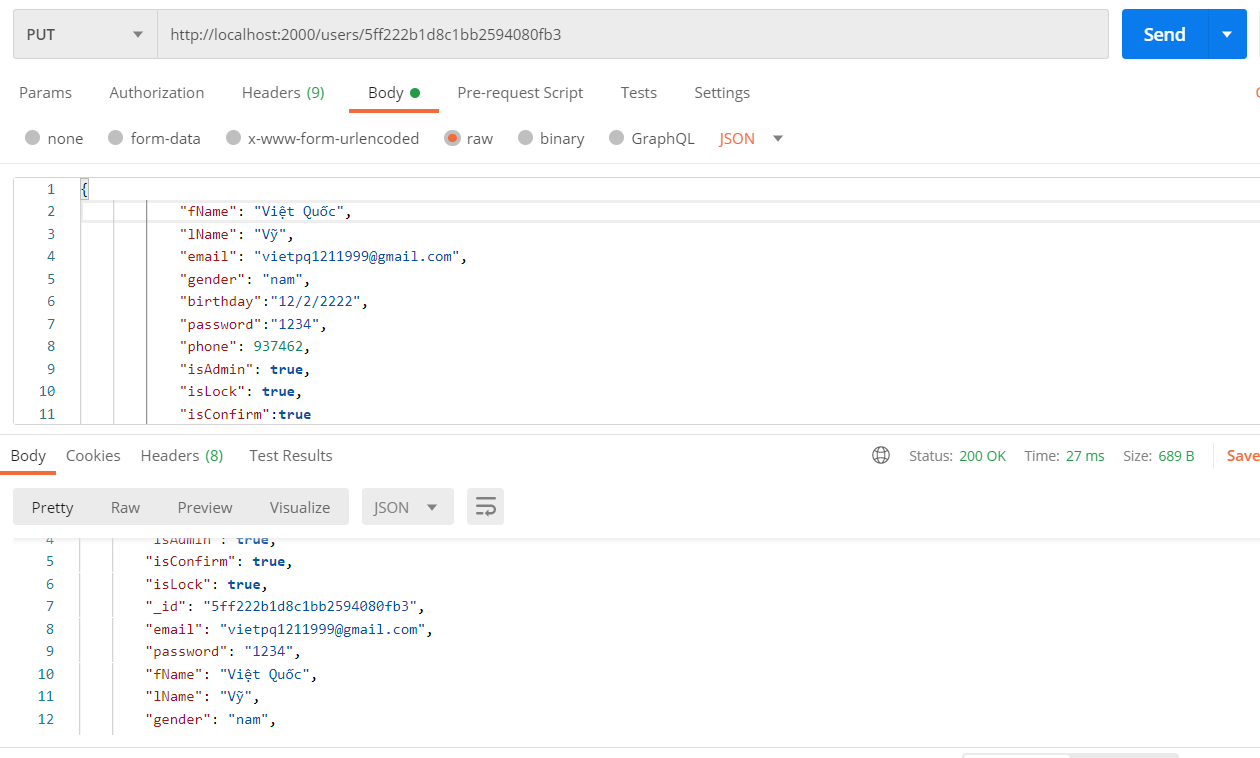
* Tạo người dùng mới



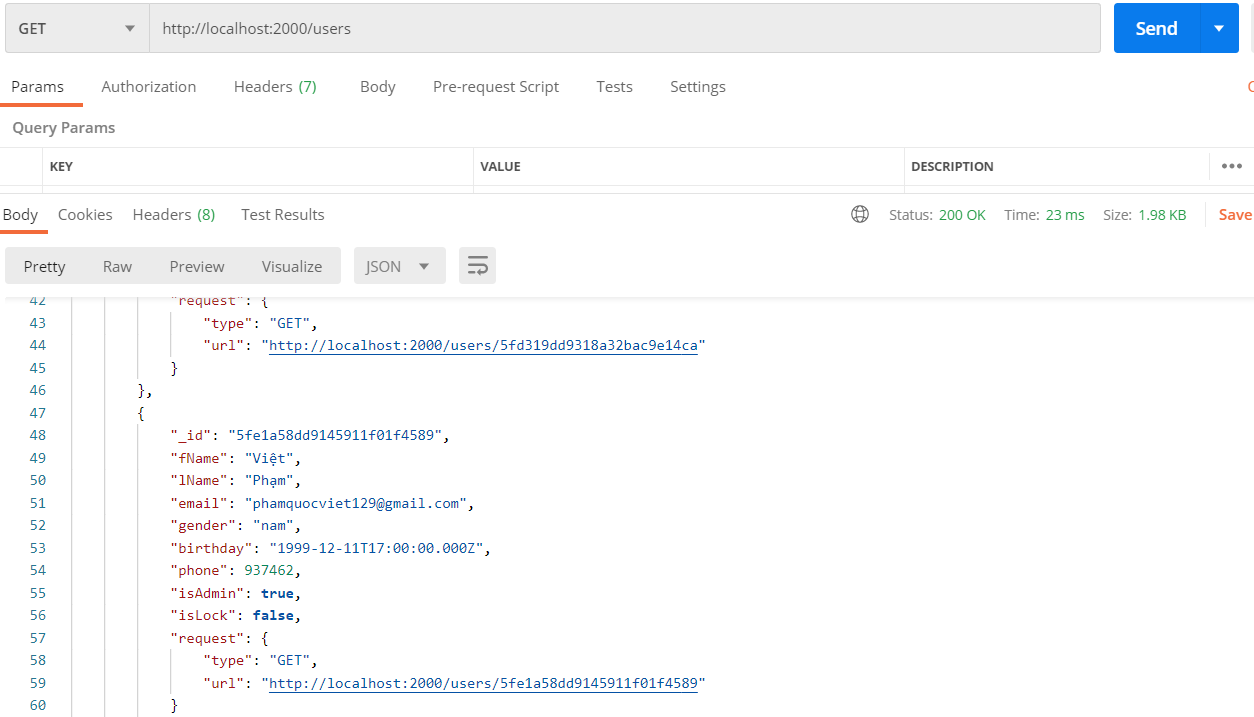
* Xóa người dùng



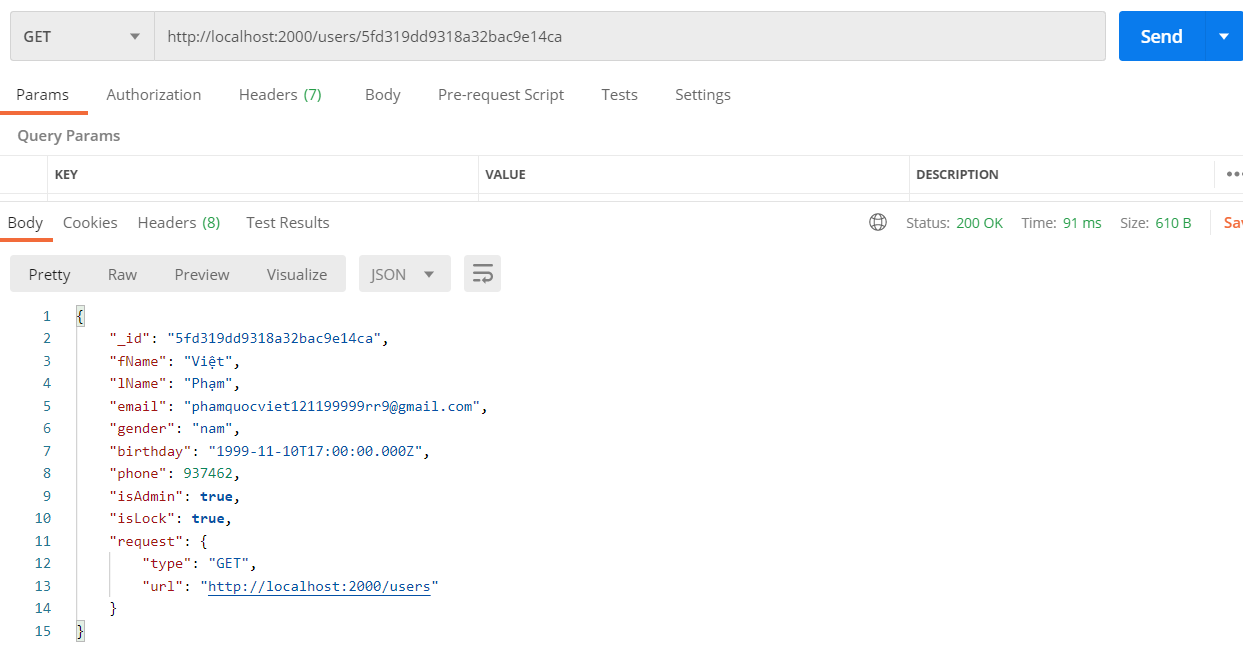
* Cập nhật người dùng



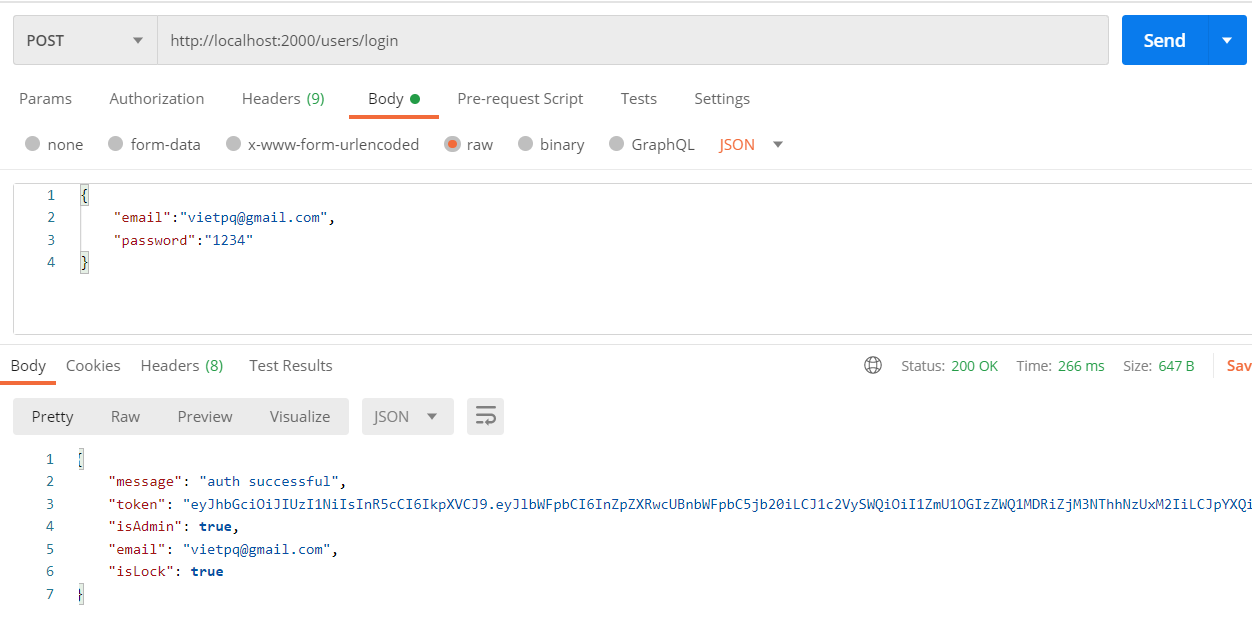
* Lấy hết danh sách người dùng



* Lấy danh sách người dùng theo ID



* Đăng nhập



# Chương 3: CÀI ĐẶT SẢN PHẨM



## Hướng dẫn cài đặt

Chạy chương trình ở local:

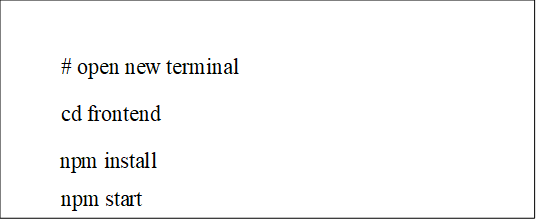
* 1. Clone respon

**G**

* 1. Cấu hình MongoDB
* Tạo database tại https://cloud.mongodb.com
* Tạo file .env ở root folder
* Set MONGODB\_URL=mongodb+srv://your-db-connection
  1. Chạy Backend

npm install  
 npm start

* 1. Chạy frontend



* 1. Tạo người dùng và sản phẩm mẫu
* Chạy url: http://localhost:2000/api/products/seed trên trình duyệt, trình duyệt sẽ trả về thông người dùng mà mật khẩu.
* Chạy url: <http://localhost:2000/api/products/seed> trên trình duyệt, trình duyệt sẽ tạo ra một vài sản phẩm.
  1. Đăng nhập bằng Admin
* Chạy url: http://localhost:4200/#/login
* Nhập thông tin email của người và password để đăng nhập vào chương trình.
* Chạy url: http://localhost:4200/#/ product-list
* Nhập thông tin sản phẩm:tên, giá,số lượng sản phẩm có và tiến hành add product

# Chương 4: TỔNG KẾT







## Kết luận

Chương trình đã hoàn thành được khoảng 70% so với yêu cầu đề ra ban đầu. Chương trình có các chức năng cơ bản của một trang web bán hàng như:

* Về phía người dùng:
* Đăng nhập, đăng ký tài khoản.
* Xem, thêm, xóa sản phẩm trong giỏ hàng.
* Đặt hàng, thanh toán đơn hàng.
* Cập nhật thông tin cá nhân.
* Tìm kiếm sản phẩm theo tên, theo danh mục.
* Về phía người quản trị:
* Thêm, sửa, xóa tài khoản người dùng.
* Thêm, sửa, xóa các sản phẩm.
* Sửa, xóa thông tin các đơn hàng.

## Ưu điểm

* Chương trình sử dụng các công nghệ mới trong việc phát triển web hiện nay như Angular, Nodejs, MongoDB, Express.
* Chương trình có các chức năng cơ bản của một trang web bán hàng, tạo nền tảng để phát triển thêm nhiều chức năng nâng cao khác.
* Các tổ chức cấu trúc code rõ ràng.

## Nhược điểm

* Chương trình vẫn còn chưa hoàn thiện.
* Một vài chỗ xử lý logic chưa được chính xác.
* Giao diện chưa được bắt mắt.
* Chưa sử dụng nhiều các ràng buộc trong cơ sở dữ liệu

## Hướng phát triển

* Tiếp tục hoàn thiện các chức năng đăng nhập, bình luận cho sản phẩm
* Thêm chức năng sắp xếp sản phẩm.
* Thêm chức năng phân trang.
* Deploy chương trình lên google cloud
* Sử dụng bootstrap để cải thiện độ thẩm mỹ của chương trình.

**TÀI LIỆU THAM KHẢO**



[1] <https://www.hostinger.vn/huong-dan/react-la-gi-va-no-hoat-dong-nhu-the-nao/>

[2] <https://itviec.com/blog/hoc-nodejs/>

[3] <https://topdev.vn/blog/mongodb-la-gi/>

[4] https://viblo.asia/p/angular-4-tu-a-den-z-Qbq5QLpwlD8