

**TRƯỜNG ĐẠI HỌC BÁCH KHOA
ĐẠI HỌC ĐÀ NẴNG
KHOA: KHOA HỌC VÀ CÔNG NGHỆ TIỀN TIẾN**

~~~~~\*~~~~~



**BÁO CÁO ĐỒ ÁN  
CHUYÊN ĐỀ MẠNG**

**Giảng viên hướng dẫn : PGS.TS Nguyễn Tấn Khôi**

**Sinh viên thực hiện : Quý Anh Khoa**

**Nguyễn Cảnh Long**

**Lớp : 18PFIEV2**

**Nhóm : 1**

**Đà Nẵng, 07/2022**

## Contents

|                                                                |    |
|----------------------------------------------------------------|----|
| <b>TÓM TẮT</b> .....                                           | 4  |
| <b>MỞ ĐẦU</b> .....                                            | 6  |
| <b>Chương 1: CƠ SỞ LÝ THUYẾT</b> .....                         | 8  |
| <b>1.1. Kiến thức cơ bản</b> .....                             | 8  |
| <b>1.1.1. HTML</b> .....                                       | 8  |
| <b>1.1.2. CSS</b> .....                                        | 8  |
| <b>1.1.3. Javascript</b> .....                                 | 8  |
| <b>1.1.4. SQL</b> .....                                        | 8  |
| <b>1.2. NodeJS và Express Framework</b> .....                  | 9  |
| <b>1.2.1. NodeJS</b> .....                                     | 9  |
| <b>1.2.2. Express Framework</b> .....                          | 9  |
| <b>1.3. React</b> .....                                        | 10 |
| <b>1.3.1. Giới thiệu</b> .....                                 | 10 |
| <b>1.3.2. Các khái niệm trong React</b> .....                  | 10 |
| <b>1.3.3. Vòng đời trong React</b> .....                       | 11 |
| <b>1.3.4. Lợi ích của React</b> .....                          | 11 |
| <b>1.4. MySQL</b> .....                                        | 12 |
| <b>1.5. API</b> .....                                          | 12 |
| <b>1.5.1. API là gì?</b> .....                                 | 12 |
| <b>1.5.2. Web API là gì?</b> .....                             | 12 |
| <b>1.5.3. RESTful API</b> .....                                | 13 |
| <b>1.5.4. RESTful API hoạt động như thế nào?</b> .....         | 13 |
| <b>1.6. Kết chương</b> .....                                   | 14 |
| <b>Chương 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG</b> .....          | 15 |
| <b>2.1 Phân tích hệ thống</b> .....                            | 15 |
| <b>2.2 Thiết kế hệ thống:</b> .....                            | 15 |
| <b>2.2.1 Sơ đồ cây phân rã chức năng</b> .....                 | 15 |
| <b>2.2.2 Thiết kế cơ sở dữ liệu</b> .....                      | 15 |
| <b>2.2.3 Tổ chức chương trình</b> .....                        | 16 |
| <b>Chương 3: TRIỂN KHAI HỆ THỐNG VÀ ĐÁNH GIÁ KẾT QUẢ</b> ..... | 19 |
| <b>3.1 Triển khai hệ thống</b> .....                           | 19 |
| <b>3.2 Kết quả</b> .....                                       | 19 |

|                                                          |    |
|----------------------------------------------------------|----|
| 3.2.1. Giao diện một số chức năng của chủ cửa hàng ..... | 19 |
| 3.2.1. Giao diện một số chức năng của người mua .....    | 21 |
| 3.3 Kết chương .....                                     | 22 |
| KẾT LUẬN .....                                           | 23 |
| TÀI LIỆU THAM KHẢO .....                                 | 24 |

## TÓM TẮT

Tên đề tài: Xây dựng website mua bán vật tư thiết bị

Tóm tắt đề tài:

Website mua bán vật tư thiết bị được xây dựng nhằm mục đích giúp tạo ra một trang thông tin điện tử được thiết lập để phục vụ một phần hoặc toàn bộ quy trình của hoạt động mua bán hàng hóa hay cung ứng dịch vụ, từ trưng bày giới thiệu hàng hóa, dịch vụ đến giao kết hợp đồng, cung ứng dịch vụ, thanh toán và dịch vụ sau bán hàng giúp cho việc tìm kiếm, đặt mua hàng của người dùng trở nên nhanh chóng, thuận tiện hơn. Bên cạnh đó, người bán hàng có thể giới thiệu sản phẩm ở thị trường rộng hơn và tiếp cận được nhiều khách hàng thông qua internet, từ đó thúc đẩy việc mua bán giữa 2 bên diễn ra dễ dàng và nhanh chóng. Website phục vụ đối tượng người dùng có nhu cầu mua những sản phẩm điện tử phục vụ các project học tập và các dự án vừa và nhỏ... Website hỗ trợ đối tượng người bán hàng quản lý danh mục và sản phẩm của cửa hàng, thống kê doanh số bán hàng,...

## DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT

### CHỮ VIẾT TẮT:

| Chữ viết tắt | Diễn giải                            |
|--------------|--------------------------------------|
| SQL          | Structured Query Language            |
| HTTP         | Hypertext Transfer Protocol          |
| API          | Application Programming Interface.   |
| HTML         | Hyper Text Markup Language           |
| SGML         | Standard Generalized Markup Language |
| CSS          | Cascading Style Sheets               |
| DOM          | Document Object Model                |
| XML          | Extensible Markup Language           |
| JSX          | JavaScript XML                       |
| JSON         | JavaScript Object Notation           |
| UI           | User Interface                       |
| URL          | Uniform Resource Locator             |

## MỞ ĐẦU

### 1. Tổng quan về đề tài

Cùng với sự tăng trưởng kinh tế trên toàn cầu là sự phát triển của các ngành khoa học kỹ thuật, đặc biệt là các ngành công nghệ thông tin. Việc ứng dụng công nghệ thông tin và các nghiên cứu của công nghệ thông tin vào đời sống trên toàn thế giới ngày càng nhiều và càng phát triển. Sự phát triển đó thể hiện bằng hàng loạt các phần mềm, thiết bị điện tử ra đời, tham gia vào hầu hết các lĩnh vực từ học tập, kinh doanh đến vui chơi giải trí. Xuất phát từ nhu cầu đặt ra việc ứng dụng công nghệ thông tin vào lĩnh vực kinh doanh, mua bán sản phẩm đã và đang ngày càng phát triển một cách mạnh mẽ. Đặc biệt đối với nhu cầu cuộc sống hiện nay, con người ngày càng có ý thức và đầu tư vào phát triển bản thân và phát triển đời sống tinh thần, vì vậy để một phần cải thiện chất lượng sống, việc mua sắm quần áo cá nhân là một điều tất yếu để nâng cao giá trị bản thân, giúp cá nhân tự tin hơn, bên cạnh đó, việc thế giới đang ngày càng xã hội hóa, toàn cầu hóa nên các nền văn hóa ngày càng đa dạng dẫn đến các dịp lễ, sự kiện trên thế giới đang ngày càng xã hội hóa, toàn cầu hóa nên các nền văn hóa ngày càng đa dạng dẫn đến các dịp lễ, sự kiện trên thế giới ngày càng phát triển và hòa nhập, việc tổ chức những sự kiện lễ hội ngày càng phát triển, và việc chọn các trang phục phù hợp với từng dịp lễ là một phần không thể thiếu. Đáp ứng những nhu cầu đó, trong đề án này em đã xây dựng website mua bán đa dạng các loại quần áo.

### 2. Mục đích đề tài

Xây dựng website mua bán các sản phẩm vật tư thiết bị điện tử phục vụ chủ yếu cho hai đối tượng khách hàng: Khách mua hàng, Chủ cửa hàng. Do đó hệ thống được tạo ra với 3 mục đích chính:

- Giúp cho người mua hàng tiết kiệm thời gian mua sắm, tìm kiếm các sản phẩm mong muốn một cách nhanh và hiệu quả nhất, quản lý các đơn hàng mình đã đặt, quản lý giỏ hàng và thanh toán một cách nhanh chóng và tiện lợi.
- Giúp cho các cửa hàng có thể giới thiệu các sản phẩm của mình đến người dùng một cách nhanh và kinh tế nhất. Giúp giảm nhiều công sức trong việc quản lý cửa hàng, thống kê bán hàng, quản lý các thông tin liên quan đến sản phẩm, quản lý đơn hàng,
- Giúp cho quản trị viên của website quản lý người dùng, quản lý tất cả các thông tin của các cửa hàng đã đăng ký bán hàng và quản lý các giao dịch dịch vụ chặt chẽ và hiệu quả.

### 3. Phạm vi đối tượng

Xây dựng được hệ thống có thể sử dụng cho phạm vi, đối tượng là người dùng trên khắp cả nước, người dùng có thể đồng thời là khách hàng và chủ cửa hàng. Chức năng bán hàng phù hợp cho các cửa hàng có quy mô vừa và nhỏ, cần tiết kiệm chi phí quảng cáo, marketing tiếp cận khách hàng, số lượng sản phẩm không quá nhiều, công việc quản lý cửa hàng không quá phức tạp. Chức năng mua hàng phù hợp cho tất cả mọi người, đặc biệt là người có ít thời gian mua sắm.

#### **4. Các bước triển khai**

Bước 1: Xác định yêu cầu của dự án

Bước 2: Tìm hiểu cơ sở lý thuyết, phân tích hệ thống

Bước 3: Thiết kế hệ thống, dữ liệu, giao diện

Bước 4: Thực hiện xây dựng các chức năng và giao diện theo thiết kế đã định

Bước 5: Kiểm thử và hoàn thiện website

#### **5. Bố cục đồ án**

Đồ án bao gồm các nội dung sau:

- Mở đầu
- Chương I: Cơ sở lý thuyết
- Chương II: Phân tích và thiết kế hệ thống
- Chương III: Triển khai hệ thống và đánh giá
- Kết luận

## Chương 1: CƠ SỞ LÝ THUYẾT

### 1.1. Kiến thức cơ bản

#### 1.1.1. HTML

HTML [1] là viết tắt cho HyperText Markup Language, hay là "Ngôn ngữ đánh dấu siêu văn bản", được thiết kế ra để tạo nên các trang web với các mẫu thông tin được trình bày trên World Wide Web. Cùng với CSS và JavaScript, HTML tạo ra bộ ba nền tảng kỹ thuật cho World Wide Web. HTML được định nghĩa như là một ứng dụng đơn giản của SGML và được sử dụng trong các tổ chức cần đến các yêu cầu xuất bản phức tạp. HTML đã trở thành một chuẩn Internet do tổ chức World Wide Web Consortium (W3C) duy trì. Phiên bản chính thức mới nhất của HTML là HTML 4.01 (1999). Sau đó, các nhà phát triển đã thay thế nó bằng XHTML. Hiện nay, HTML đang được phát triển tiếp với phiên bản HTML5 hứa hẹn mang lại diện mạo mới cho Web.

#### 1.1.2. CSS

CSS [2] là chữ viết tắt của Cascading Style Sheets, nó là một ngôn ngữ được sử dụng để tìm và định dạng lại các phần tử được tạo ra bởi các ngôn ngữ đánh dấu (ví dụ như HTML). Bạn có thể hiểu đơn giản rằng, nếu HTML đóng vai trò định dạng các phần tử trên website như việc tạo ra các đoạn văn bản, các tiêu đề, bảng, ... thì CSS sẽ giúp chúng ta có thể thêm một chút "phong cách" vào các phần tử HTML đó như đổi màu sắc trang, đổi màu chữ, thay đổi cấu trúc, ... rất nhiều.

#### 1.1.3. Javascript

Javascript [3] là một ngôn ngữ lập trình kịch bản dựa trên đối tượng được phát triển từ các ý niệm nguyên mẫu. Ngôn ngữ này được dùng rộng rãi cho các trang web, nhưng cũng được dùng để tạo khả năng viết script sử dụng các đối tượng nằm sẵn trong các ứng dụng. Nó vốn được phát triển bởi Brendan Eich tại Hãng truyền thông Netscape với cái tên đầu tiên Mocha, rồi sau đó đổi tên thành LiveScript, và cuối cùng thành JavaScript. Giống Java, JavaScript có cú pháp tương tự C, nhưng nó gần với Self hơn Java. .js là phần mở rộng thường được dùng cho tập tin mã nguồn JavaScript.

#### 1.1.4. SQL

SQL [4] là viết tắt của Structured Query Language có nghĩa là ngôn ngữ truy vấn có cấu trúc, là một ngôn ngữ máy tính để lưu trữ, thao tác và truy xuất dữ liệu được lưu trữ trong một cơ sở dữ liệu quan hệ. SQL là ngôn ngữ chuẩn cho hệ cơ sở dữ liệu quan hệ. Tất cả các hệ thống quản lý cơ sở dữ liệu quan hệ (RDMS) như MySQL, MS Access, Oracle, Sybase, Informix, Postgres và SQL Server đều sử dụng SQL làm ngôn ngữ cơ sở dữ liệu chuẩn.



## 1.2. NodeJS và Express Framework

### 1.2.1. NodeJS

NodeJS [5] là một mã nguồn được xây dựng dựa trên nền tảng Javascript V8 Engine, nó được sử dụng để xây dựng các ứng dụng web như các trang video clip, các forum và đặc biệt là trang mạng xã hội phạm vi hẹp. NodeJS là một mã nguồn mở được sử dụng rộng bởi hàng ngàn lập trình viên trên toàn thế giới.

NodeJS có thể chạy trên nhiều nền tảng hệ điều hành khác nhau từ Window cho tới Linux, OS X nên đó cũng là một lợi thế. NodeJS cung cấp các thư viện phong phú ở dạng Javascript Module khác nhau giúp đơn giản hóa việc lập trình và giảm thời gian ở mức thấp nhất.

Node.js sử dụng kiến trúc hướng sự kiện event-driven, mô hình non-blocking I/O làm cho nó nhẹ và hiệu quả hơn. Hệ thống nén của Node.js, npm, là hệ thống thư viện nguồn mở lớn nhất thế giới.

Đặc điểm:

- Không đồng bộ: Tất cả các API của NodeJS đều không đồng bộ (non-blocking), nó chủ yếu dựa trên nền của NodeJS Server và chờ đợi Server trả dữ liệu về. Việc di chuyển máy chủ đến các API tiếp theo sau khi gọi và cơ chế thông báo các sự kiện của Node.js giúp máy chủ để có được một phản ứng từ các cuộc gọi API trước.

- Chạy rất nhanh: NodeJS được xây dựng dựa vào nền tảng V8 Javascript Engine nên việc thực thi chương trình rất nhanh.

- Đơn luồng nhưng khả năng mở rộng cao: Node.js sử dụng một mô hình luồng duy nhất với sự kiện lập. cơ chế tổ chức sự kiện giúp các máy chủ để đáp ứng một cách không ngăn chặn và làm cho máy chủ cao khả năng mở rộng như trái ngược với các máy chủ truyền thống mà tạo đề hạn chế để xử lý yêu cầu. Node.js sử dụng một chương trình đơn luồng và các chương trình tương tự có thể cung cấp dịch vụ cho một số lượng lớn hơn nhiều so với yêu cầu máy chủ truyền thống như Apache HTTP Server.

- Không đệm: NodeJS không đệm bất kỳ một dữ liệu nào và các ứng dụng này chủ yếu là đầu ra dữ liệu.

### 1.2.2. Express Framework

Express [6] là một trong những framework phổ biến dùng để xây dựng API và Website phổ biến nhất của NodeJS. Nó được sử dụng rộng rãi đến mức hầu như mọi dự án Web nào đều bắt đầu bằng việc tích hợp Express. Có rất nhiều lý do để chọn Express:

- Có nhiều tính năng hỗ trợ tất cả những gì bạn cần trong việc xây dựng Web và API

- Quản lý các route dễ dàng
- Cung cấp một nền tảng phát triển cho các API
- Hỗ trợ nhiều thư viện và plugin
- Bảo mật và an toàn hơn so với việc code thuần
- Hỗ trợ cộng đồng tuyệt vời

### 1.3. React

#### 1.3.1. Giới thiệu

React (React.js hay ReactJS) [7] là một thư viện JavaScript hiệu quả và linh hoạt để xây dựng giao diện người dùng. React là một thư viện giao diện người dùng dựa trên thành phần open-source chỉ chịu trách nhiệm cho lớp xem (View) của ứng dụng. Nó được duy trì bởi Facebook.

React được xem là cơ sở cho sự phát triển của các ứng dụng một trang (singlepage-app) hướng client-side-render, tức cập nhập ở phía máy khách. React sử dụng cú pháp JSX để xây dựng các thành phần của giao diện. Một ứng dụng được tạo thành từ nhiều thành phần (Component), mỗi thành phần chịu trách nhiệm hiển thị một đoạn HTML nhỏ, có thể tái sử dụng. Các thành phần có thể được lồng trong các thành phần khác để cho phép các ứng dụng phức tạp được xây dựng từ các khối xây dựng đơn giản. Một thành phần cũng có thể duy trì trạng thái bên trong nó. React sẽ cập nhật và hiển thị đúng thành phần khi trạng thái dữ liệu thay đổi.

#### 1.3.2. Các khái niệm trong React

- JSX: JSX là 1 cú pháp mở rộng cho JavaScript, là kết hợp của JavaScript và XML. Nó chuyển đổi cú pháp dạng gần như XML về thành JavaScript, giúp chúng ta có thể code React bằng cú pháp của XML thay vì phải dùng JavaScript.

- Component: Component là nơi render ra các mã html thuần, 1 điều quan trọng khác đó là nó cho phép ta chia nhỏ đoạn code của UI thành những phần độc lập với nhau, với mục đích để tiện cho việc quản lý và tái sử dụng.

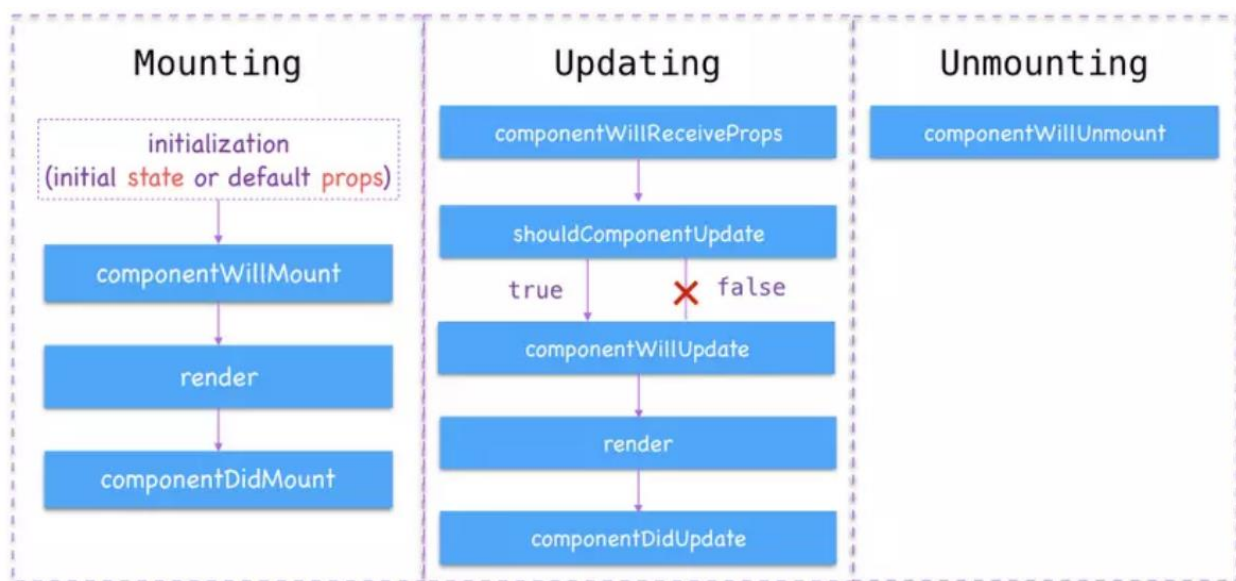
- Props: Props là viết tắt của Properties, ta có thể hiểu nó như là những thuộc tính của 1 Component. Chúng ta có thể thay đổi props của component bằng cách truyền dữ liệu từ bên ngoài vào. Props có thể là 1 object, function, string, number..... Để định nghĩa Props thì ta khai báo chúng ở bên trong dấu {} của Component.

- State: State được dùng để biểu diễn trạng thái của Component. Tức là giá trị của State có thể thay đổi được, trái ngược hoàn toàn với Props là 1 giá trị bất biến. Mỗi khi state

của 1 Component thay đổi thì nó sẽ tạo ra các sự thay đổi trong cấu trúc của DOM và nhiệm vụ của React là render lại tất cả những sự thay đổi đó.

### 1.3.3. Vòng đời trong React

Một thành phần trong React có thể có các trạng thái, thuộc tính riêng của nó. Mỗi khi trạng thái của nó thay đổi, React sẽ cập nhật và hiển thị đúng sự thay đổi đó. Cụ thể, vòng đời của một thành phần trong React được thể hiện qua hình sau:



### 1.3.4. Lợi ích của React

- Dễ sử dụng: React là một thư viện GUI nguồn mở JavaScript tập trung vào một điều cụ thể; hoàn thành nhiệm vụ UI hiệu quả. Nó được phân loại thành kiểu “V” trong mô hình MVC (Model-View-Controller).

- Hỗ trợ các thành phần tái sử dụng (reusable component): React cho phép sử dụng lại components đã được phát triển thành các ứng dụng khác có cùng chức năng. Tính năng tái sử dụng component là một lợi thế khác biệt cho các lập trình viên.

- Viết component dễ dàng hơn: React component dễ viết hơn vì nó sử dụng JSX, mở rộng cú pháp tùy chọn cho JavaScript cho phép kết hợp HTML với JavaScript.

- Hiệu suất tốt hơn với Virtual DOM: React sẽ cập nhật hiệu quả quá trình DOM (Document Object Model – Mô hình đối tượng tài liệu). React cho phép xây dựng các virtual DOM và host chúng trong bộ nhớ. Nhờ vậy, mỗi khi có sự thay đổi trong DOM thực tế, thì virtual sẽ thay đổi ngay lập tức. Hệ thống này sẽ ngăn DOM thực tế để buộc các bản cập nhật được liên tục. Do đó, tốc độ của ứng dụng sẽ không bị gián đoạn.

- Thân thiện với SEO: React cho phép bạn tạo giao diện người dùng có thể được truy cập trên các công cụ tìm kiếm khác nhau.

## 1.4. MySQL

MySQL [8] là hệ quản trị cơ sở dữ liệu mã nguồn mở phổ biến hàng đầu trên thế giới và đặc biệt được ưa chuộng trong quá trình xây dựng, phát triển ứng dụng. Đây là hệ quản trị cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có khả năng thay đổi mô hình sử dụng phù hợp với điều kiện công việc khả chuyển. MySQL hoạt động trên nhiều hệ điều hành, cung cấp một hệ thống lớn các hàm tiện ích rất mạnh.

Ưu điểm:

- Dễ sử dụng: MySQL là cơ sở dữ liệu tốc độ cao, ổn định, dễ sử dụng và hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh.
- Độ bảo mật cao: MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên Internet khi sở hữu nhiều nhiều tính năng bảo mật thậm chí là ở cấp cao.
- Đa tính năng: MySQL hỗ trợ rất nhiều chức năng SQL được mong chờ từ một hệ quản trị cơ sở dữ liệu quan hệ cả trực tiếp lẫn gián tiếp.
- Khả năng mở rộng và mạnh mẽ: MySQL có thể xử lý rất nhiều dữ liệu và hơn thế nữa nó có thể được mở rộng nếu cần thiết.
- Nhanh chóng: Việc đưa ra một số tiêu chuẩn cho phép MySQL để làm việc rất hiệu quả và tiết kiệm chi phí, do đó nó làm tăng tốc độ thực thi.

Nhược điểm:

- Dung lượng hạn chế: Nếu số bản ghi của dự án lớn thì việc truy xuất dữ liệu là khá khó khăn

## 1.5. API

### 1.5.1. API là gì?

API [9] là các phương thức, giao thức kết nối với các thư viện và ứng dụng khác. Nó là viết tắt của Application Programming Interface – giao diện lập trình ứng dụng. API cung cấp khả năng cung cấp khả năng truy xuất đến một tập các hàm hay dùng. Và từ đó có thể trao đổi dữ liệu giữa các ứng dụng.

### 1.5.2. Web API là gì?

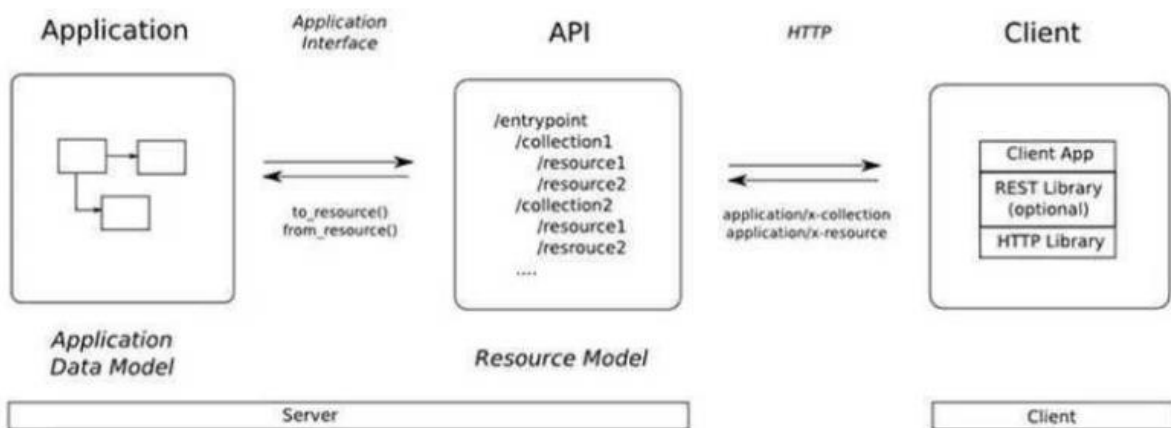
Web API [9] là một phương thức dùng để cho phép các ứng dụng khác nhau có thể giao tiếp, trao đổi dữ liệu qua lại. Dữ liệu được Web API trả lại thường ở dạng JSON hoặc XML thông qua giao thức HTTP hoặc HTTPS. Web API hỗ trợ restful đầy đủ các phương thức:

Get/Post/put/delete dữ liệu. Nó giúp bạn xây dựng các HTTP service một cách rất đơn giản và nhanh chóng. Nó cũng có khả năng hỗ trợ đầy đủ các thành phần HTTP: URI, request/response headers, caching, versioning, content forma.

### 1.5.3. RESTful API

RESTful API [10] là một tiêu chuẩn dùng trong việc thiết kế các API cho các ứng dụng web để quản lý các resource. RESTful là một trong những kiểu thiết kế API được sử dụng phổ biến ngày nay để cho các ứng dụng (web, mobile...) khác nhau giao tiếp với nhau. Chức năng quan trọng nhất của REST là quy định cách sử dụng các HTTP method (như GET, POST, PUT, DELETE...) và cách định dạng các URL cho ứng dụng web để quản các resource. RESTful không quy định logic code ứng dụng và không giới hạn bởi ngôn ngữ lập trình ứng dụng, bất kỳ ngôn ngữ hoặc framework nào cũng có thể sử dụng để thiết kế một RESTful API.

### 1.5.4. RESTful API hoạt động như thế nào?



REST [10] hoạt động chủ yếu dựa vào giao thức HTTP.

Các hoạt động cơ bản nêu trên sẽ sử dụng những phương thức HTTP riêng.

- GET (SELECT): Trả về một Resource hoặc một danh sách Resource.
- POST (CREATE): Tạo mới một Resource.
- PUT (UPDATE): Cập nhật thông tin cho Resource.
- DELETE (DELETE): Xóa một Resource. Những phương thức hay hoạt động này thường được gọi là CRUD tương ứng với Create, Read, Update, Delete – Tạo, Đọc, Sửa, Xóa

## 1.6. Kết chương

Chương này cung cấp cơ sở lý thuyết về các công nghệ, ngôn ngữ lập trình được sử dụng trong hệ thống, đồng thời nêu ra những lý do để áp dụng các công nghệ, ngôn ngữ lập trình nói trên vào hệ thống hiện tại.

## Chương 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

### 2.1 Phân tích hệ thống

Website được chia thành 2 đối tượng chính: Chủ cửa hàng đăng ký bán hàng trên trang web và khách hàng mua hàng.

- Khách hàng mua hàng sẽ có một số chức năng chính như sau:

+ Xem thông tin về tất cả các loại sản phẩm mà website hiển thị.

- Đối với chủ cửa hàng, ngoài việc có đầy đủ các chức năng của khách hàng còn có thêm các nhóm chức năng sau:

+ Quản lý sản phẩm: Chủ cửa hàng quản lý sản phẩm bao gồm danh mục sản phẩm, danh sách các sản phẩm thuộc danh mục, chi tiết sản phẩm. Chủ cửa hàng có đầy đủ quyền để thêm mới, chỉnh sửa hay xóa sản phẩm của cửa hàng.

### 2.2 Thiết kế hệ thống:

#### 2.2.1 Sơ đồ cây phân rã chức năng



#### 2.2.2 Thiết kế cơ sở dữ liệu

- Bảng cơ sở dữ liệu của chủ cửa hàng:

| STT | Name         | Avatar | Email           | Password | Role  | Created at | Updated at |
|-----|--------------|--------|-----------------|----------|-------|------------|------------|
| 1   | Quí Anh Khoa |        | qkhoa@gmail.com | 123456   | Admin |            |            |
| 2   |              |        |                 |          |       |            |            |

- Bảng cơ sở dữ liệu phân loại sản phẩm category:

| STT | Name |
|-----|------|
| 1   |      |
| 2   |      |

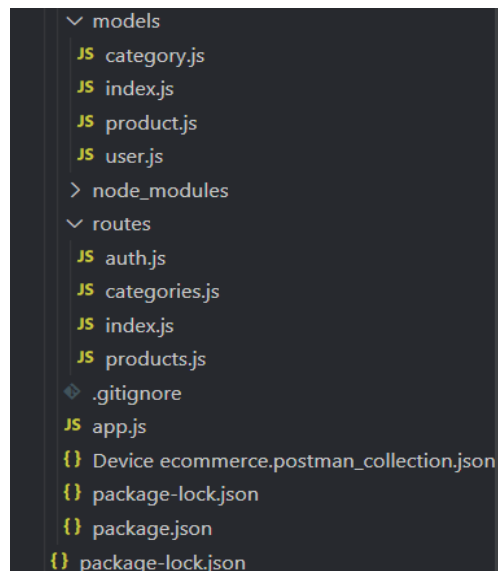
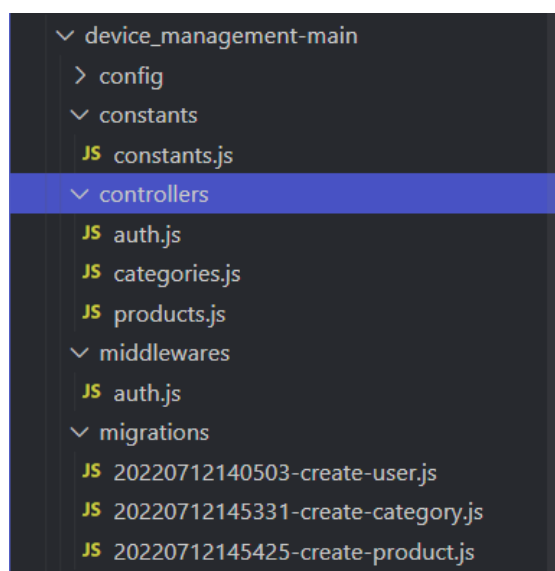
- Bảng cơ sở dữ liệu của sản phẩm:

| STT | Name | Description | Price | Stock | Image | Sold quantity | CategoryId |
|-----|------|-------------|-------|-------|-------|---------------|------------|
| 1   |      |             |       |       |       |               |            |
| 2   |      |             |       |       |       |               |            |

## 2.2.3 Tổ chức chương trình

### 2.2.3.1 Chương trình Back-end NodeJs

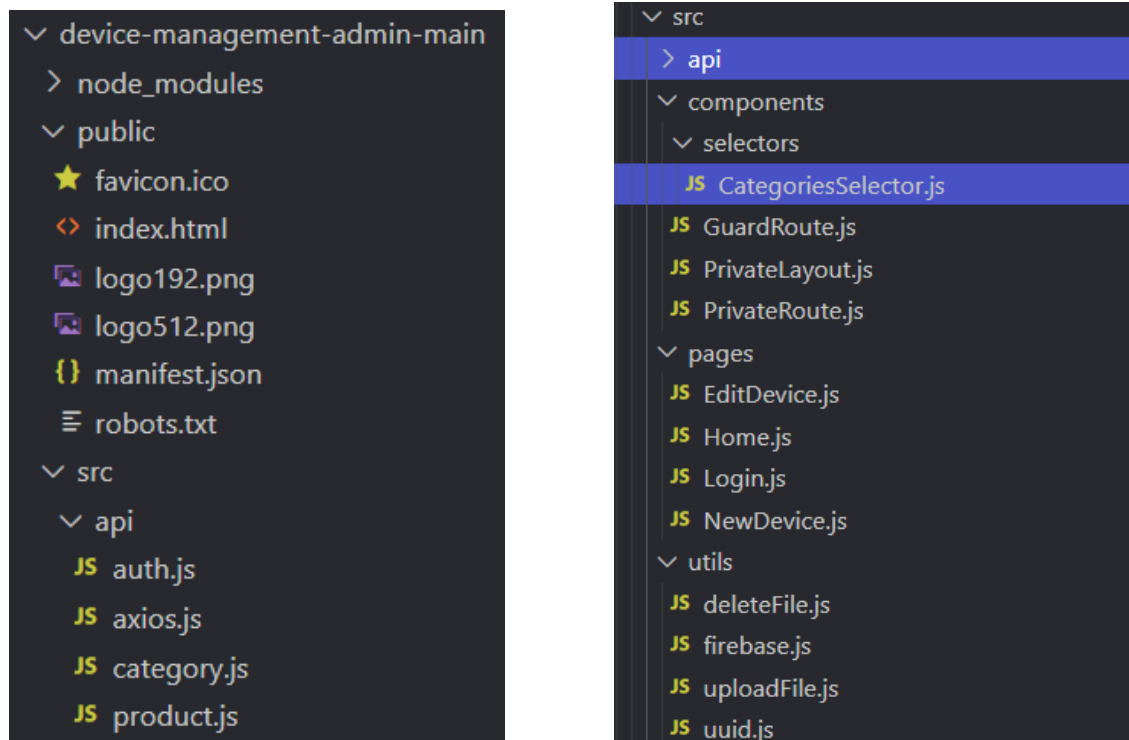
- Tổ chức thư mục:





- Folder models: tạo các file để tương tác với database.
- Folder migrations: chạy các file tạo những table ở database.
- Folder controllers: các đoạn code xử lý chính của project sẽ nằm ở đây.
- Folder routes: các file trong thư mục router sẽ import các file từ controller vào; thư mục chứa những đường link mong muốn cho api với file index chứa tất cả đường link ở trong.
- Middle ware: login logout đã giúp project lấy được token rồi lưu lại và những request còn lại đều được xác thực thông qua token và để xử lý được token cần một bộ chặn trước; xác định token để xác định được người gửi request lên hệ thống; xử lý trước khi gọi đến các routes còn lại trong server.
- Folder config: Lấy những thông tin từ cloud database để đưa vào file config như host, database name, user, password.
- Folder constant: đặt các role trong website (admin và user).

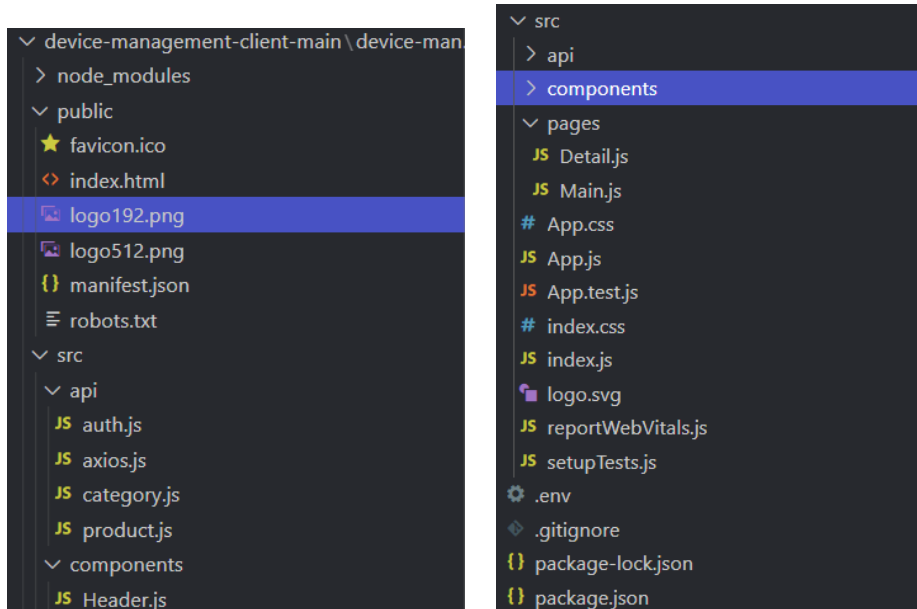
### 2.2.3.2 Chương trình Front-end cho chủ cửa hàng:



- Folder api: chứa các đường link và file axios (cài sẵn thư viện axios) thực hiện giao tiếp giữa front-end và back-end.
- Folder component: chứa các component của trang web.

- Folder pages: là trang chính: thực hiện chức năng như đăng nhập, tạo thêm sản phẩm, sửa sản phẩm trên giao diện trang web.
- Folder utils: giúp project thực hiện các thao tác trao đổi với firebase.

### 2.2.3.3 Chương trình Front-end cho khách hàng:



## Chương 3: TRIỂN KHAI HỆ THỐNG VÀ ĐÁNH GIÁ KẾT QUẢ

### 3.1 Triển khai hệ thống

- Ngôn ngữ lập trình: JavaScript
- Back-end: Express Framework
- Front-end: React
- Hệ quản trị cơ sở dữ liệu: MySQL
- Thao tác khởi chạy hệ thống:

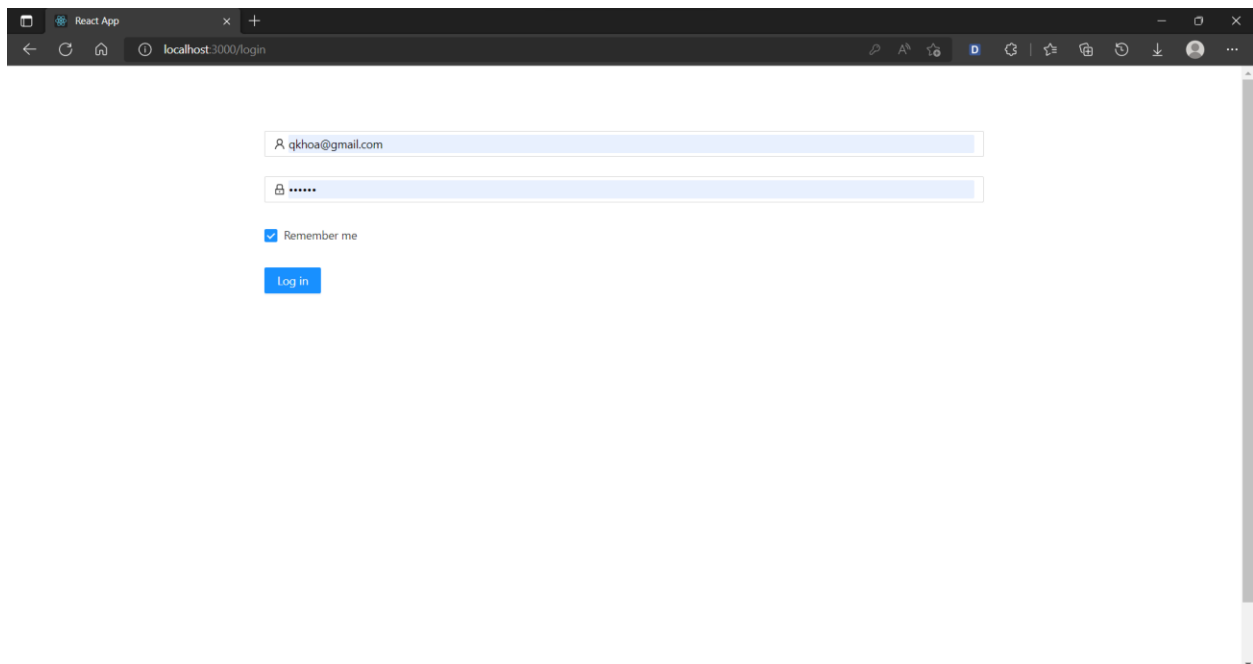
+ npm install

+ npm start

### 3.2 Kết quả

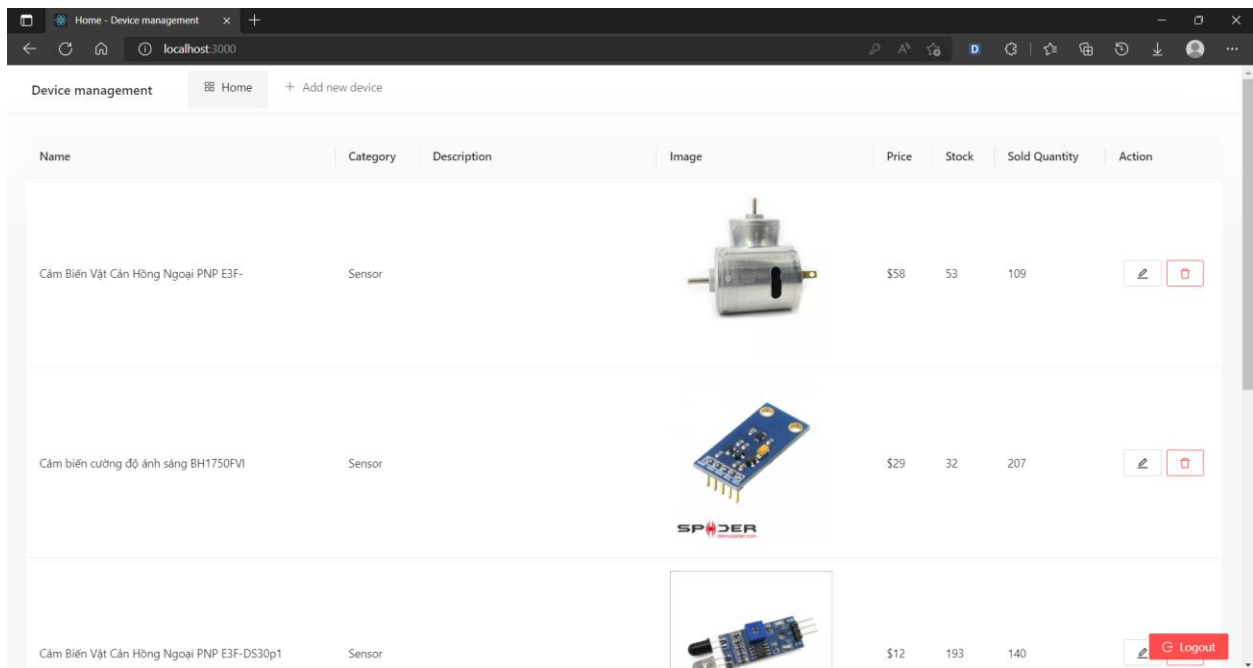
Dưới đây là một số hình ảnh giao diện của hệ thống đã được xây dựng:










#### 3.2.1. Giao diện một số chức năng của chủ cửa hàng



Màn hình đăng nhập của chủ cửa hàng

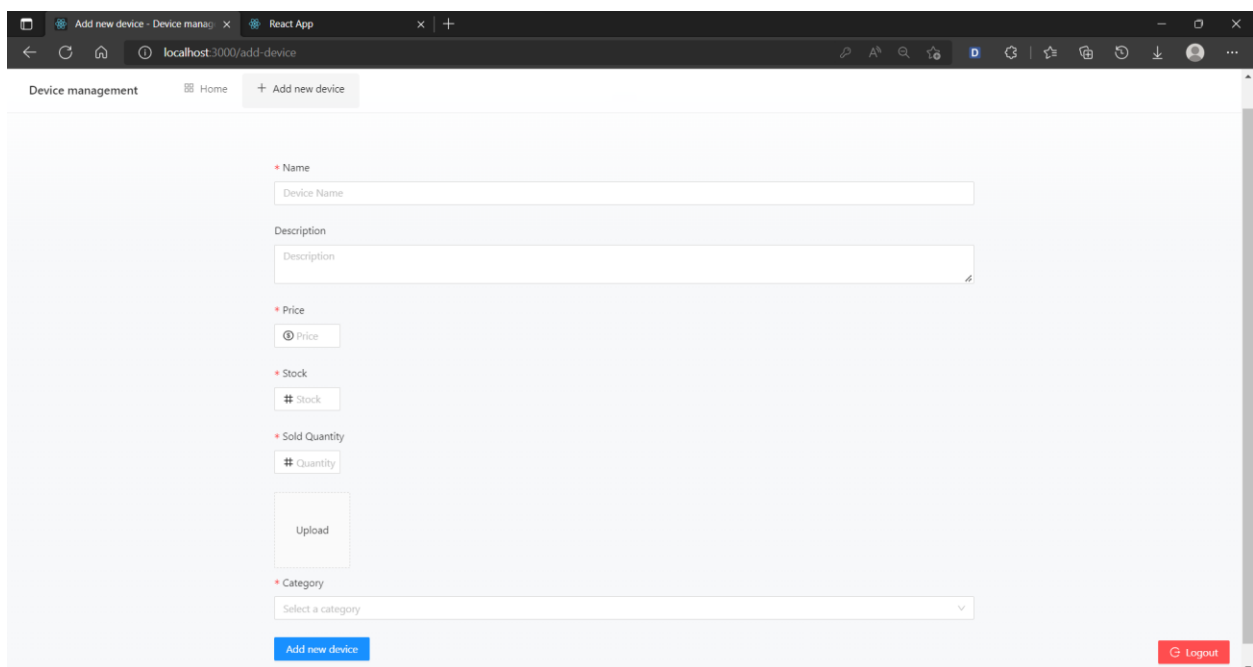
Màn hình trên là giao diện đăng nhập khi chủ cửa hàng muốn đăng nhập vào hệ thống. Chủ cửa hàng đăng nhập vào hệ thống bằng cách nhập email và mật khẩu hợp lệ và nhấn login.



| Name                                       | Category | Description | Image                                                                              | Price | Stock | Sold Quantity | Action                                                                                                                                                                  |
|--------------------------------------------|----------|-------------|------------------------------------------------------------------------------------|-------|-------|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cảm Biến Vật Cận Hồng Ngoại PNP E3F-       | Sensor   |             |   | \$58  | 53    | 109           |   |
| Cảm biến cường độ ánh sáng BH1750FVI       | Sensor   |             |   | \$29  | 32    | 207           |   |
| Cảm Biến Vật Cận Hồng Ngoại PNP E3F-DS30p1 | Sensor   |             |  | \$12  | 193   | 140           |   |

Giao diện xem các sản phẩm và chi tiết sản phẩm của cửa hàng

Tại màn hình xem chính chủ của hàng có thể thực hiện các thao tác xem, sửa, xóa, add thêm sản phẩm vào trang web.



Device management

Home + Add new device

\* Name  
Device Name

Description  
Description

\* Price  
Price

\* Stock  
# Stock

\* Sold Quantity  
# Quantity

Upload

\* Category  
Select a category

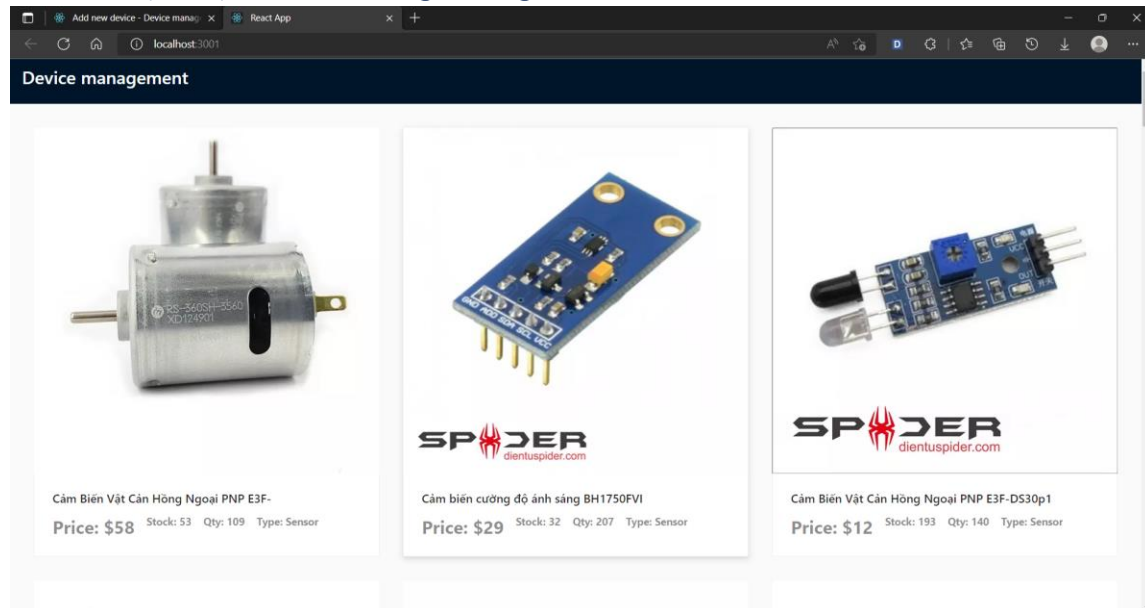
Add new device Logout

Màn hình add thêm sản phẩm vào trang web

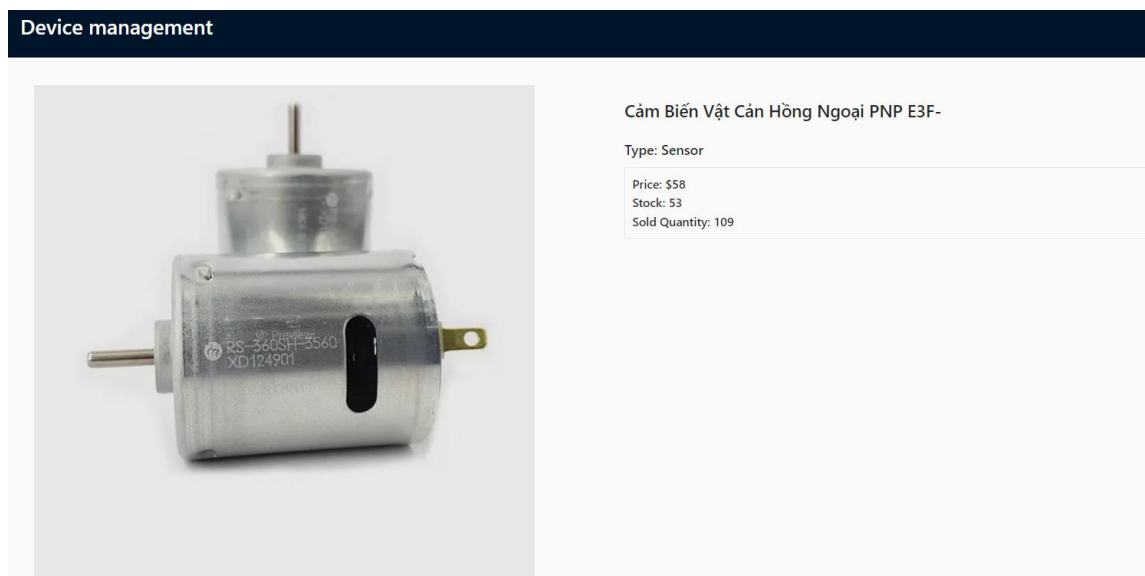
✓ Added device successfully

Thông báo khi đã add thêm sản phẩm thành công

### 3.2.1. Giao diện một số chức năng của người mua



Giao diện xem các sản phẩm của người mua



Giao diện chi tiết của từng sản phẩm

### 3.3 Kết chương

Chương này trình bày một số yêu cầu để triển khai hệ thống. Một số hình ảnh về việc triển khai website, giao diện website của một số chức năng cơ bản, giúp người đọc có cái nhìn tổng quan và cụ thể hơn về website.

## KẾT LUẬN

### 1. Kết quả

Trong thời gian tìm hiểu, nghiên cứu cơ sở lý thuyết và triển khai ứng dụng công nghệ, đề án đã đạt được những kết quả sau:

#### 1.1 Kết quả đạt được

Về mặt lý thuyết:

- Nắm được quy trình xây dựng phần mềm.
- Vận dụng được các kiến thức đã học để phân tích và thiết kế hệ thống.
- Có thêm kiến thức, thực hành về ngôn ngữ Javascript.
- Có thêm kiến thức, thực hành về Node.js, Express Framework và React.
- Nắm bắt được các kinh nghiệm về thiết kế dữ liệu, thiết kế giao diện.
- Xử lý sự kiện thời gian thực với tính năng nhận thông báo.

Về mặt thực tiễn ứng dụng, đã xây dựng website đáp ứng được một số tính năng cơ bản của một website thương mại điện tử.

#### 1.2 Kết quả chưa đạt được

- Website chưa thực sự bắt mắt người dùng.
- Website người mua hàng chưa có chức năng tìm kiếm sản phẩm, chức năng tạo giỏ hàng để thực hiện mua hàng và thanh toán onl.
- Chưa hỗ trợ việc tư vấn trực tiếp giữa người bán và người mua trên nền tảng.
- Chưa có áp dụng được thuật toán để gợi ý sản phẩm cho người mua, hay thuật toán giúp cho người dùng tìm kiếm tối ưu.

### 2. Hướng phát triển

- Thêm chức năng mua hàng, tạo giỏ hàng và thanh toán onl cho khách hàng.
- Thêm các chức năng tương tác giữa người bán và người mua.
- Xây dựng ứng dụng trên thiết bị di động.
- Áp dụng các thuật toán nâng cao như hệ thống gợi ý sản phẩm.

## TÀI LIỆU THAM KHẢO

- [1] HTML, <https://developer.mozilla.org/en-US/docs/Web/HTML>
- [2] CSS, <https://developer.mozilla.org/en-US/docs/Web/HTML>
- [3] JavaScript, <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- [4] SQL, <https://vi.wikipedia.org/wiki/SQL>
- [5] NodeJS, Express, [https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express\\_Nodejs/Introduction](https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Introduction)
- [6] React, <https://reactjs.org>
- [7] MySQL, Sequelize ORM, <https://sequelize.org/api/v6>