**TRƯỜNG ĐẠI HỌC ĐÀ LẠT**

KHOA CÔNG NGHỆ THÔNG TIN



**BÁO CÁO BÀI TẬP NHÓM**

**HỌC PHẦN PHÁT TRIỂN ỨNG DỤNG WEB NÂNG CAO**

**ĐỀ TÀI: XÂY DỰNG ỨNG DỤNG WEB**

**QUẢN LÝ CỬA HÀNG SƠN NƯỚC**

Sinh viên thực hiện: 2014508 - Nguyễn Trường Vũ

2011389 - Nguyễn Thị Huế

2011428 - Nghiêm Xuân Anh Quân

**MỤC LỤC**

[I. Tổng quan lý thuyết 1](#_Toc12182)

[1. API 1](#_Toc4487)

[2. React 1](#_Toc12555)

[II. Phân tích và xây dựng website 2](#_Toc3813)

[1. Phân tích: 2](#_Toc23250)

[1.1 Yêu cầu chung của website: 2](#_Toc13963)

[1.2 Sơ đồ use case: 3](#_Toc21780)

[1.3 Danh sách chức năng của website: 3](#_Toc21450)

[2. Xây dựng hệ thống API 4](#_Toc14952)

[1. Account API: 4](#_Toc18755)

[2. Category API: 4](#_Toc21439)

[3. Product API: 5](#_Toc20882)

[4. Product Unit API: 7](#_Toc30966)

[3. Xây dựng ứng dụng 8](#_Toc30242)

[1. Giao diện trang chủ ứng dụng 8](#_Toc25281)

[2. Giao diện và chức năng trang sản phẩm 8](#_Toc30190)

[3. Giao diện xem chi tiết sản phẩm 9](#_Toc24830)

[4. Giao diện và chức năng trang liên hệ 9](#_Toc25334)

[5. Giao diện và chức năng trang đăng nhập và đăng ký: 11](#_Toc16579)

[6. Giao diện Admin và các chức năng: 11](#_Toc13488)

[III. Hướng dẫn cài đặt và chạy chương trình 13](#_Toc18328)

[1. Yêu cầu phần mềm 13](#_Toc16071)

[2. Các bước thực hiện 13](#_Toc5561)

[2.1 Chỉnh sửa tên server cơ sở dữ liệu 13](#_Toc14036)

[2.2 Chọn dự án khởi chạy mặt định 13](#_Toc3939)

[2.3 Chạy chương trình và thay đổi địa chỉ API Endpoint 14](#_Toc2306)

[2.4 Cài đặt các gói tin 16](#_Toc11631)

[2.5 Biên dịch chương trình 17](#_Toc31421)

[2.6 Chạy chương trình 18](#_Toc22465)

[IV. Kết luận và hướng phát triển 18](#_Toc21463)

[1. Kết luận: 18](#_Toc16484)

[2. Hướng phát triển: 19](#_Toc18882)

[V. Bảng phân công công việc 19](#_Toc11332)

2. **Tổng quan lý thuyết**
3. **API**

- Minimal API là một tính năng mới trong ASP.NET Core, một framework phát triển ứng dụng web phía máy chủ của Microsoft. Nó được giới thiệu từ phiên bản ASP.NET Core 6.0.

- Minimal API cung cấp một cách tiếp cận đơn giản và tối giản hóa việc xây dựng các API và ứng dụng web. Với Minimal API, bạn không cần phải cấu hình nhiều điều khiển (controllers), routes và middleware như trước đây. Thay vào đó, bạn chỉ cần viết mã để định nghĩa các endpoint và xử lý các yêu cầu HTTP.

- Cách sử dụng Minimal API khá đơn giản. Bạn chỉ cần khởi tạo một đối tượng WebApplication và sử dụng các phương thức như MapGet, MapPost, MapPut, MapDelete để xác định các endpoint và xử lý yêu cầu tương ứng.

1. **React**

- React là một thư viện JavaScript phổ biến được sử dụng để xây dựng giao diện người dùng cho các ứng dụng web. Nó được phát triển bởi Facebook và được cung cấp miễn phí cho cộng đồng.

- React được thiết kế để tập trung vào việc quản lý trạng thái (state) và hiển thị giao diện. Với React, bạn có thể tạo ra các thành phần (components) tái sử dụng và cấu trúc lại chúng để tạo thành giao diện phức tạp.

- React sử dụng một cú pháp đặc biệt gọi là JSX để tạo ra các thành phần. JSX cho phép bạn viết HTML và JavaScript trong cùng một tệp, làm cho mã nguồn của ứng dụng dễ đọc và dễ bảo trì hơn.

- React cũng có một hệ thống sự kiện (event system) mạnh mẽ để xử lý các tương tác người dùng trên giao diện. Bạn có thể gắn các hàm xử lý sự kiện vào các thành phần và định nghĩa trạng thái của ứng dụng để điều khiển hành vi của giao diện.

- Một tính năng nổi bật của React là virtual DOM. Virtual DOM là một bản sao của DOM, một cây phân cấp của các phần tử HTML trên trang web. Virtual DOM được sử dụng để tối ưu hóa hiệu suất của ứng dụng bằng cách giảm số lần truy cập thực tế vào DOM, giúp tăng tốc độ hiển thị và đáp ứng của ứng dụng.

- React là một công nghệ phát triển phổ biến cho các ứng dụng web hiện đại và có nhiều hỗ trợ từ cộng đồng. Nó có thể được sử dụng độc lập hoặc kết hợp với các thư viện và framework khác để xây dựng ứng dụng web phức tạp.

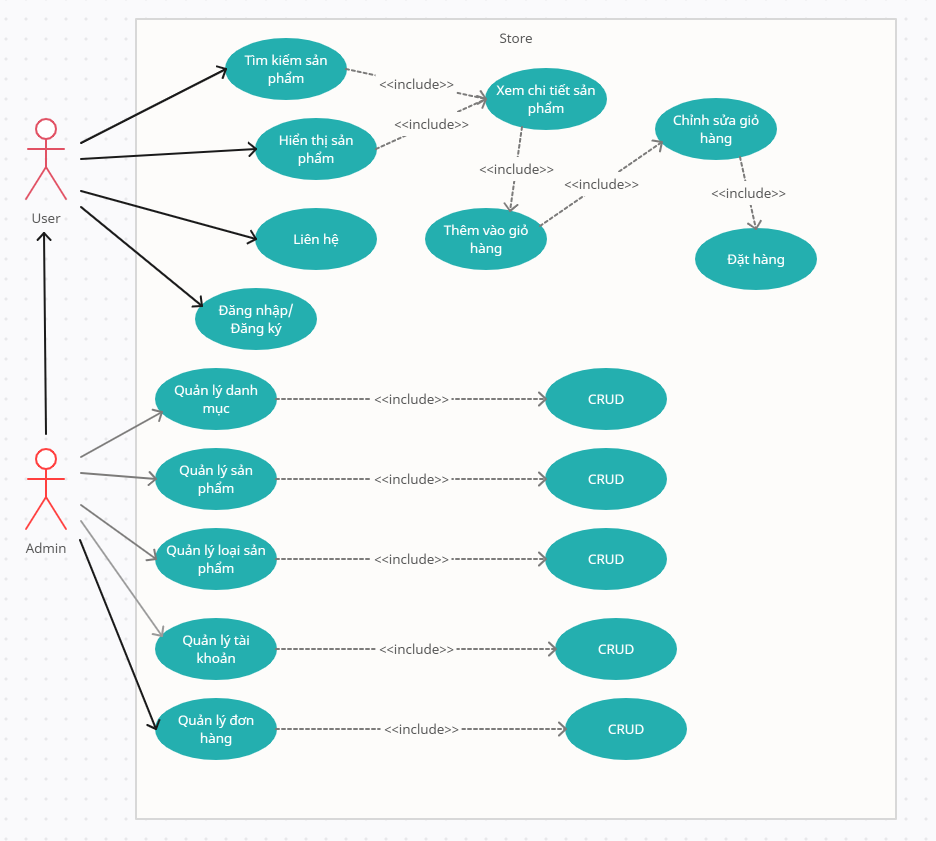
1. **Phân tích và xây dựng website**
2. **Phân tích:**
   1. **Yêu cầu chung của website:**

Xây dựng website cho cửa hàng bán sơn nước.

Yêu cầu chung cho website gồm có:

* Trang chủ sẽ giới thiệu một số thông tin về cửa hàng như banner quảng cáo, giới thiệu, các sản phẩm mới nhất.
* Trang danh mục sản phẩm để hiện thị các sản phẩm thuộc danh mục đó hoặc tất cả sản phẩm. Phần tìm kiếm sẽ giúp người dùng tìm được sản phẩm mong muốn.
* Trang sản phẩm chi tiết để hiện thị toàn bộ thông tin của một sản phẩm kèm các loại sản phảm ứng với mỗi sản phẩm.
* Giỏ hàng để xem các sản phẩm mà khách hàng lựa chọn.
* Trang đặt hàng để người dùng gửi đơn hàng.
* Trang liên hệ để người dùng gửi thắc mắc đến chủ cửa hàng và hiển thị vị trí cửa hàng trên google map.
* Trang tin tức để cập thông tin của cửa hàng cho khác hàng.
* Trang đăng nhập, đăng ký.
* Trang quản thông tin cho khách hàng.
* Trang quản lý danh mục để hiển thị danh sách các danh mục và các chức năng như thêm sửa xóa.
* Trang quản lý sản phẩm để hiển thị danh sách các sản phẩm và các chức năng như thêm sửa xóa. Đồng thời hiện thị danh sách các loại sản phẩm mà sản phẩm, có thể thêm xóa hoặc sửa.
* Trang dashboard quản lý thông tin chung như số hàng, số loại hàng, tiền,…
* Trang quản lý người dùng để quản lý thông tin khách hàng (không hiển thị phần mật khẩu).

**1.2 Sơ đồ use case:**

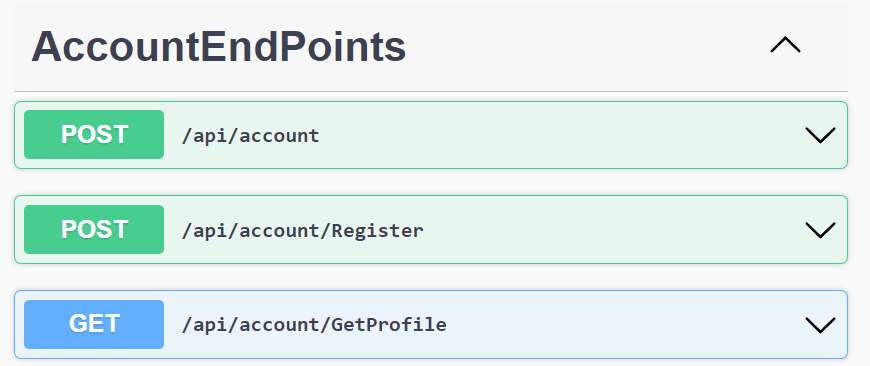


Ghi chú: CRUD - Create (Thêm); Read (Đọc); Update (Cập nhập); Delete (Xóa)

**1.3 Danh sách chức năng của website:**

|  |  |  |
| --- | --- | --- |
| **STT** | **Loại** | **Chức năng** |
| **1** | Quản lý danh mục | * Hiển thị danh sách danh mục * Tạo danh mục * Chỉnh sửa danh mục * Xóa danh mục |
| **2** | Quản lý sản phẩm | * Hiển thị danh sách sản phẩm * Tạo sản phẩm * Chỉnh sửa sản phẩm * Xóa sản phẩm * Thêm hình cho sản phẩm |
| **3** | Quản lý loại sản phẩm | * Hiển thị danh sách loại sản phẩm * Tạo loại sản phẩm * Chỉnh sửa loại sản phẩm * Xóa loại sản phẩm |
| **4** | Xem chi tiết sản phẩm | * Hiển thị chi tiết sản phẩm * Hiển thị các loại sản phẩm và giá cả ứng với loại sản phẩm đó. |
| **5** | Tìm kiếm (admin page) | * Tìm kiếm sản phẩm theo từ khóa * Tìm kiếm danh mục theo từ khóa |
| **6** | Tìm kiếm | * Tìm kiếm sản phẩm theo từ khóa |
| **7** | Đăng nhập | * Đăng nhập |
| **8** | Đăng ký | * Đăng ký |
| **9** | Liên hệ | * Gửi thông tin tới gmail của shop |

1. **Xây dựng hệ thống API**
2. **Account API:**

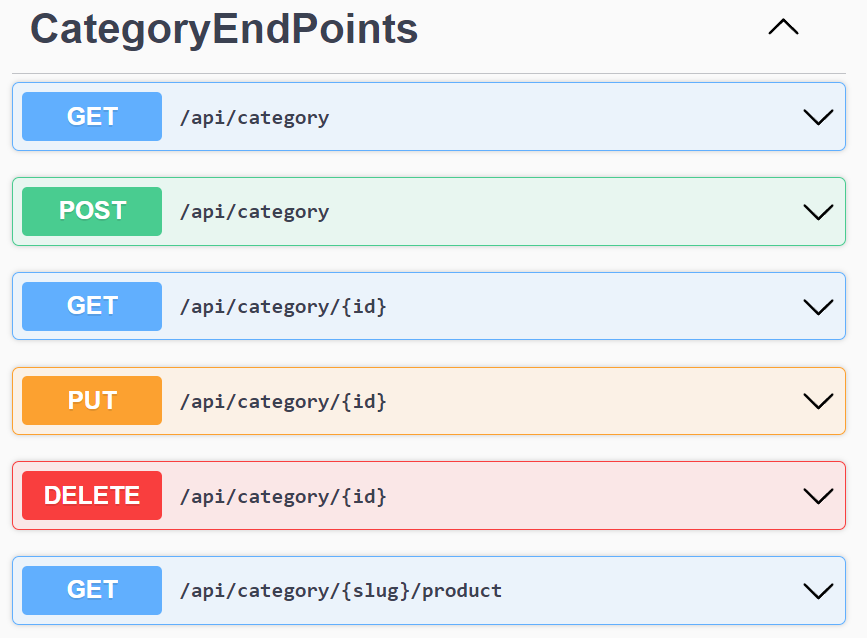


- POST /api/account: Endpoints kiểm tra đăng nhập để trả về Token và thông tin người dùng.

- POST /api/account/Register: Endpoints kiểm tra đăng ký.

- GET /api/account/GetProfile: Lấy thông tin người dùng bằng Token

1. **Category API:**



- GET /api/category: Endpoint sẽ trả về danh sách các danh mục.

- POST /api/category: Endpoint dùng để thêm danh mục mới.

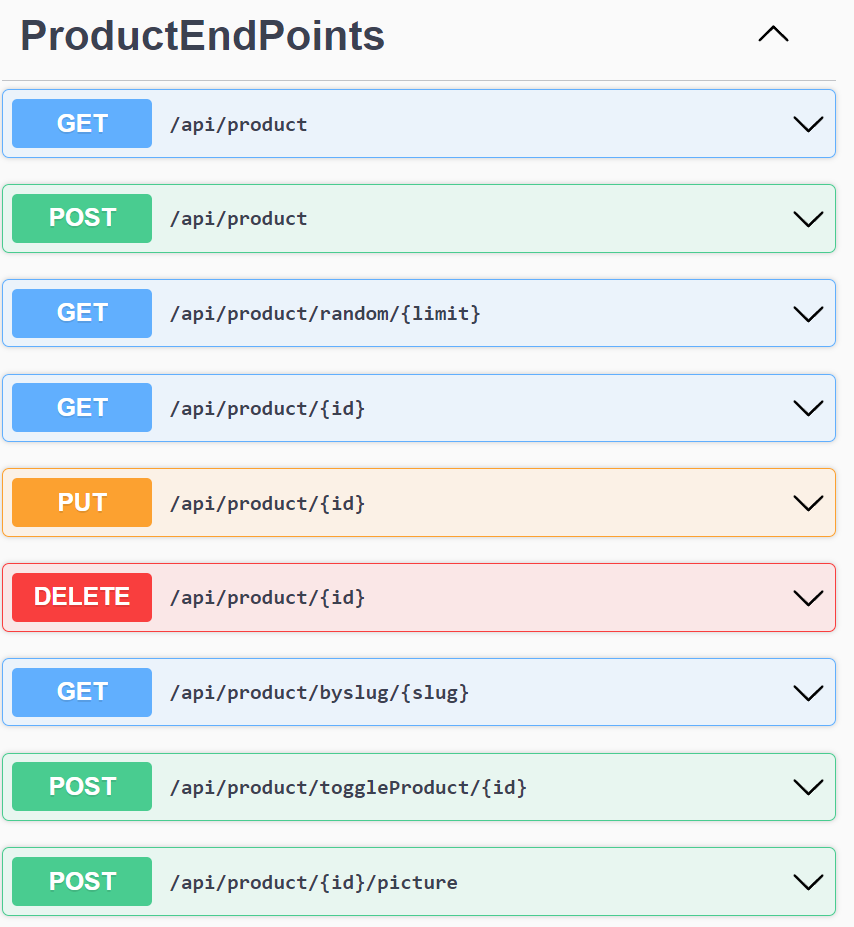
- GET /api/category/{id}: Endpoint sẽ trả về danh mục với “id” tương ứng.

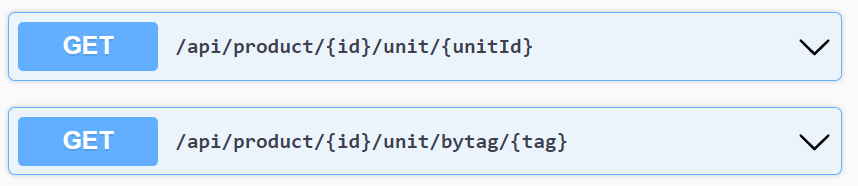
- PUT /api/category/{id}: Endpoint sẽ cập nhập thông tin danh mục với “id” tương ứng.

- DELETE /api/category/{id}: Endpoints sẽ xóa danh mục với “id” tương ứng.

- GET /api/category/{slug}/product: Endpoint sẽ trả về thông tin chi tiết của các sản phẩm có slug tương ứng với slug được gửi yêu cầu từ phía người dùng.

1. **Product API:**





- GET /api/product: Endpoint sẽ trả về danh sách các sản phẩm.

- POST /api/product: Endpoint dùng để thêm sản phẩm mới.

- GET /api/product/random/{limit}: Endpoint sẽ lấy ngẫu nhiên sản phẩm với số lượng {limit}.

- GET /api/product/{id}: Endpoint sẽ trả về sản phẩm với “id” tương ứng.

- PUT /api/product/{id}: Endpoint sẽ cập nhập thông tin sản phẩm với “id” tương ứng.

- DELETE /api/product/{id}: Endpoints sẽ xóa sản phẩm với “id” tương ứng.

- GET /api/product/byslug/{slug}: Endpoint sẽ trả về sản phẩm với “urlSlug” tương ứng với “slug”.

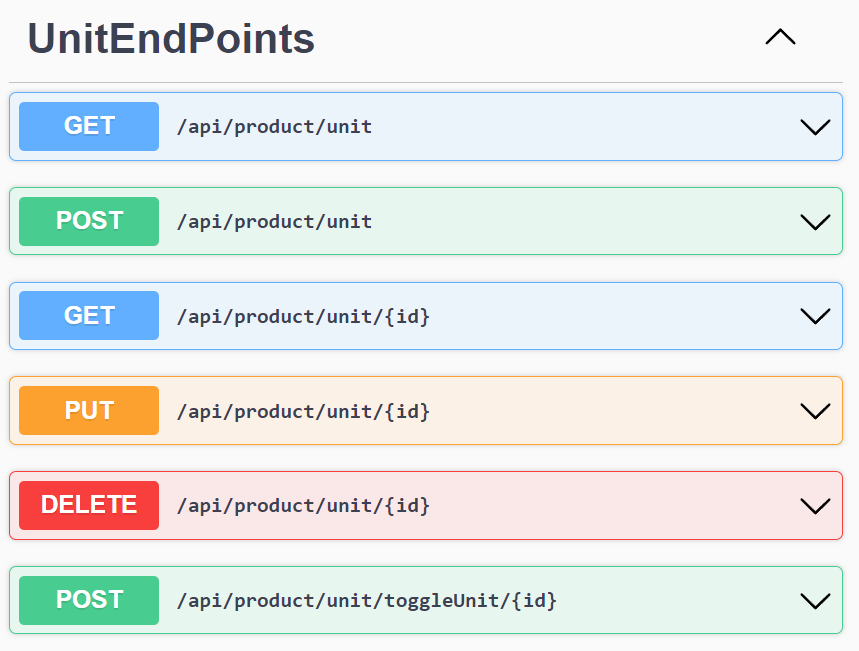
- POST /api/product/toggleProduct/{id}: Endpoints sẽ thay đổi trạng thái còn hàng của sản phẩm với “id” tương ứng.

- POST /api/product/{id}/picture: Endpoints sẽ thay đổi ảnh của sản phẩm với “id” tương ứng.

- GET /api/product/{id}/unit/{unitId}: Endpoints sẽ trả về thông tin của loại sản phẩm với “id” tương ứng với “unitId” của sản phẩm có “id”.

- GET /api/product/{id}/unit/bytag/{tag}: Endpoints sẽ trả về thông tin của loại sản phẩm với “unitTag” tương ứng với “tag” của sản phẩm có “id”.

1. **Product Unit API:**



- GET /api/product/unit: Endpoint sẽ trả về danh sách các loại sản phẩm.

- POST /api/product/unit: Endpoint dùng để thêm loại sản phẩm mới.

- GET /api/product/unit/{id}: Endpoint sẽ trả về loại sản phẩm với “id” tương ứng.

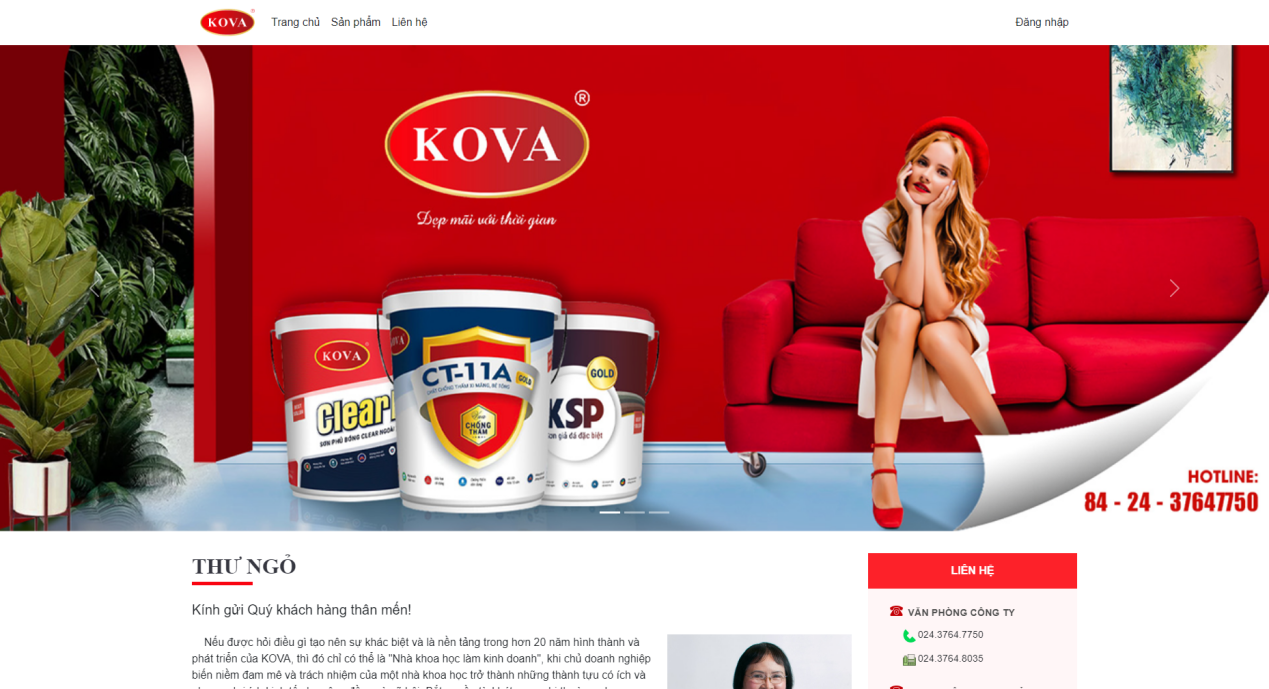
- PUT /api/product/unit/{id}: Endpoint sẽ cập nhập thông tin loại sản phẩm với “id” tương ứng.

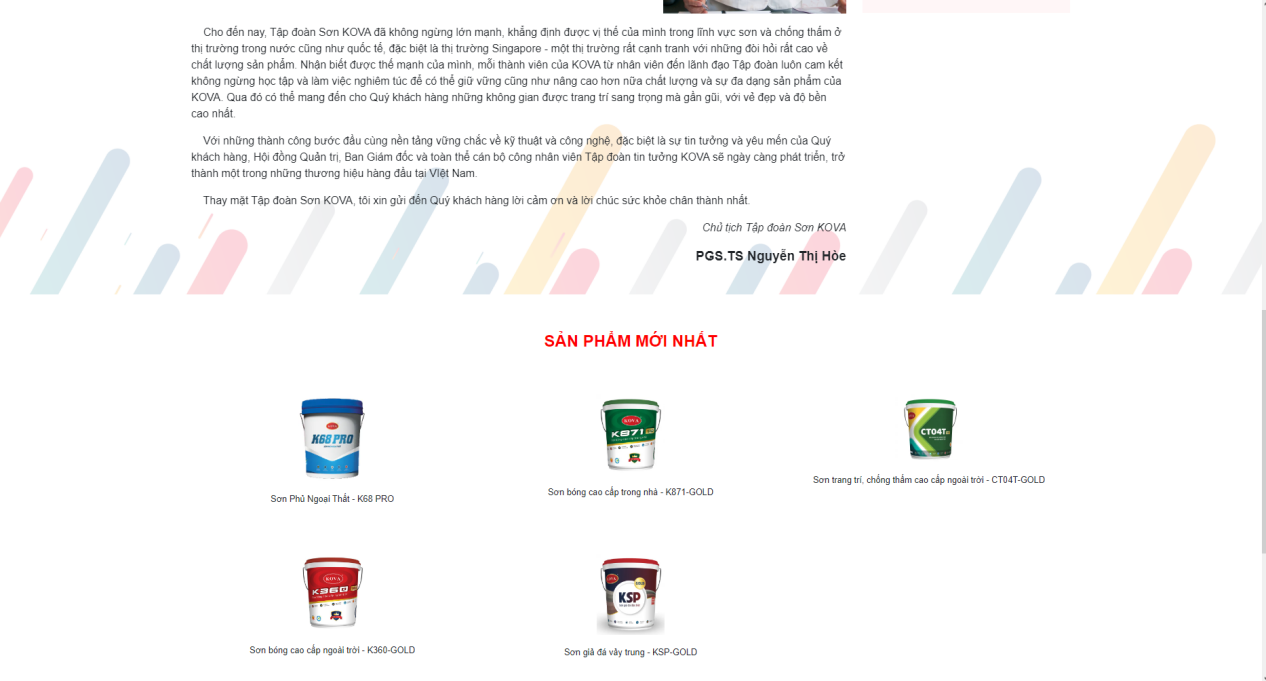
- DELETE /api/product/unit/{id}: Endpoints sẽ xóa loại sản phẩm với “id” tương ứng.

- POST /api/product/unit/toggleProduct/{id}: Endpoints sẽ thay đổi trạng thái còn hàng của loại sản phẩm với “id” tương ứng.

1. **Xây dựng ứng dụng**
2. **Giao diện trang chủ ứng dụng**

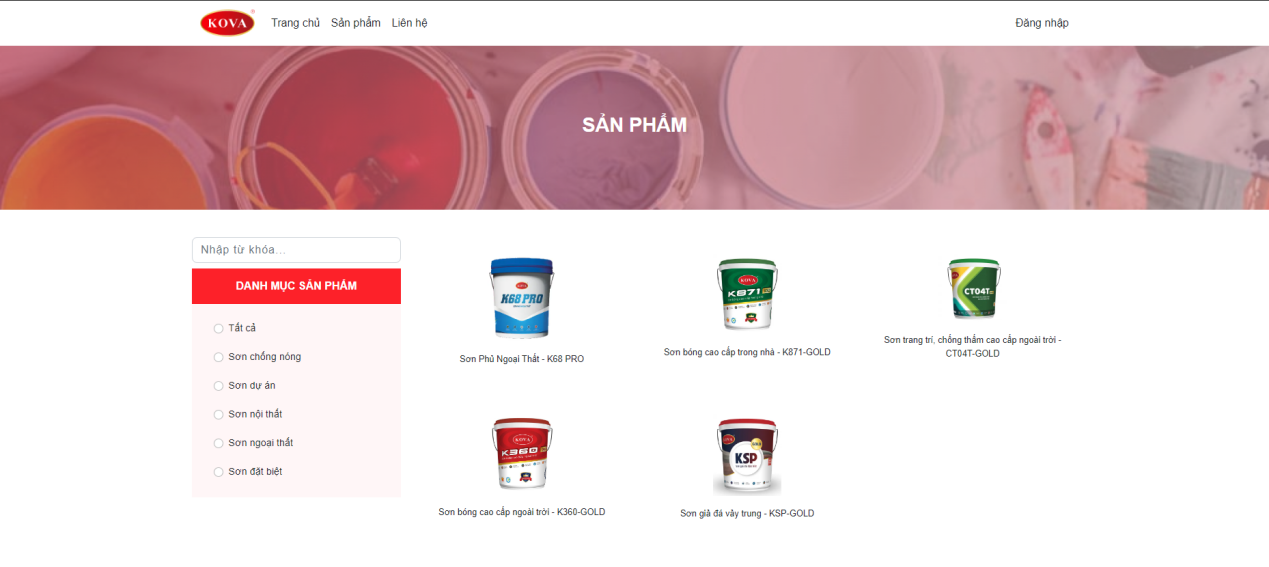
Trang chủ website hiện thị thông tin quảng cáo về cửa hàng, giới thiệu và các sản phẩm mới nhất của cửa hàng.

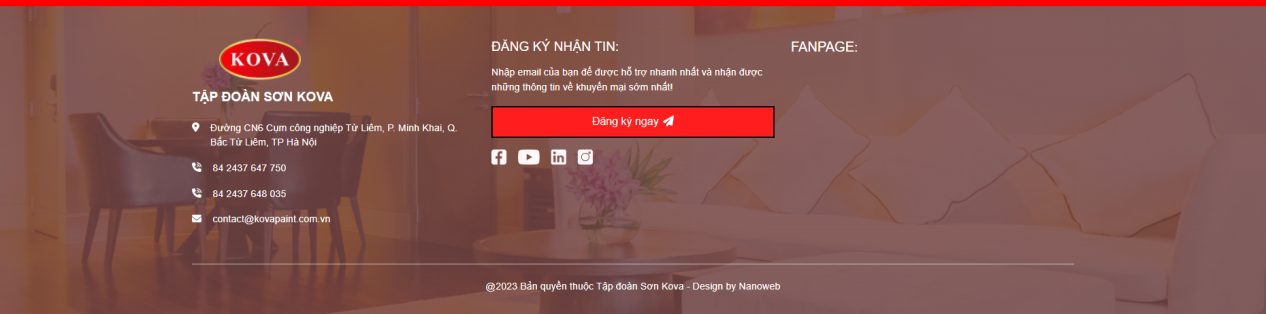




1. **Giao diện và chức năng trang sản phẩm**

Xem được toàn bộ sản phẩm hiện tại, có thể lọc sản phẩm qua phần “Danh mục sản phẩm” hoặc tìm kiếm sản phẩm qua từ khóa.





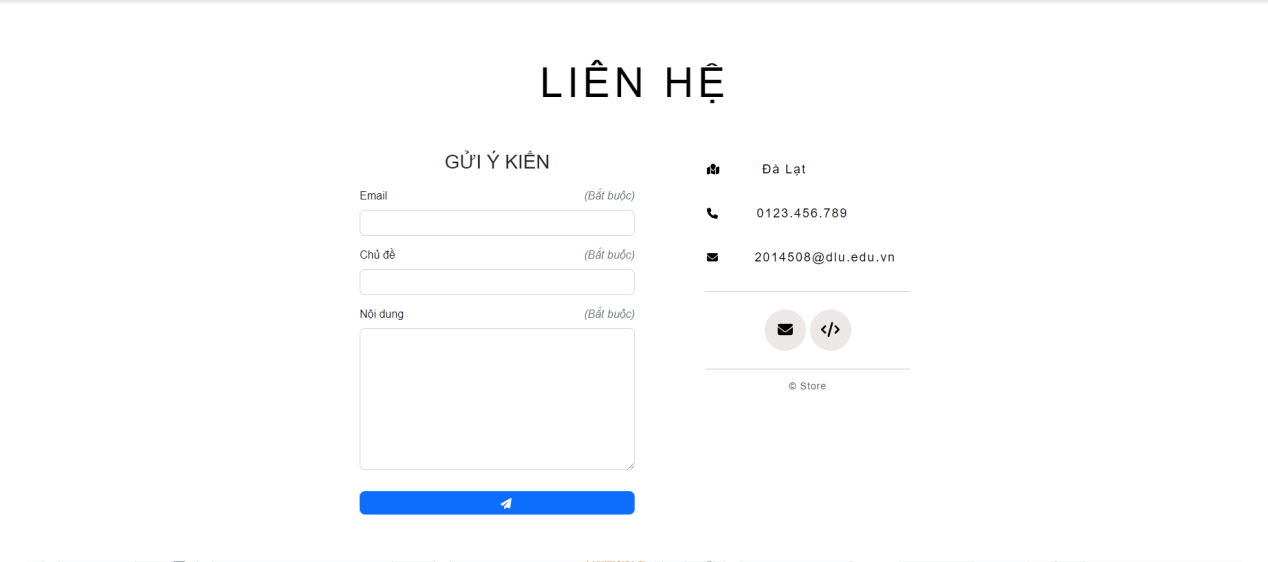
1. **Giao diện xem chi tiết sản phẩm**

Trang chi tiết sản phẩm hiện thị đầy đủ thông tin của sản phẩm và tất cả các loại sản phẩm của sản phẩm cùng với giá tiền tương ứng.

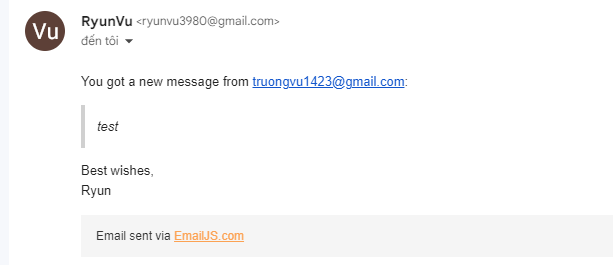


1. **Giao diện và chức năng trang liên hệ**

Trang sẽ gồm 2 phần là phần “Liên hệ” và phần bản đồ hiển thị vị trí shop.

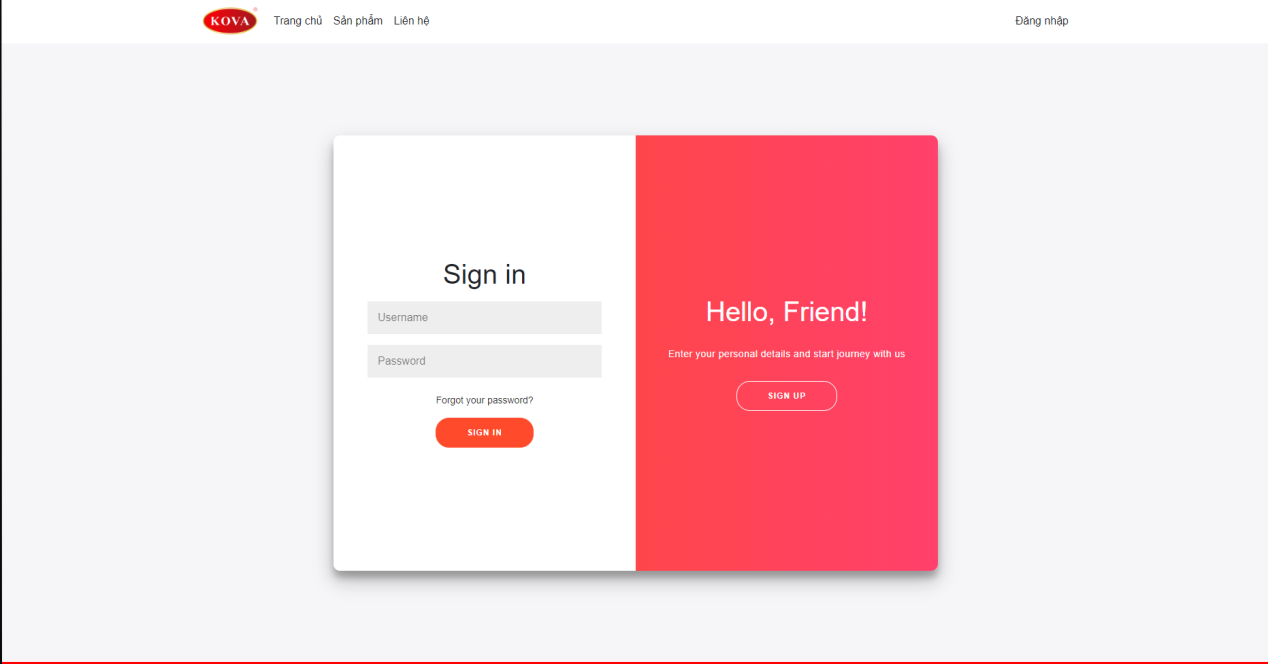


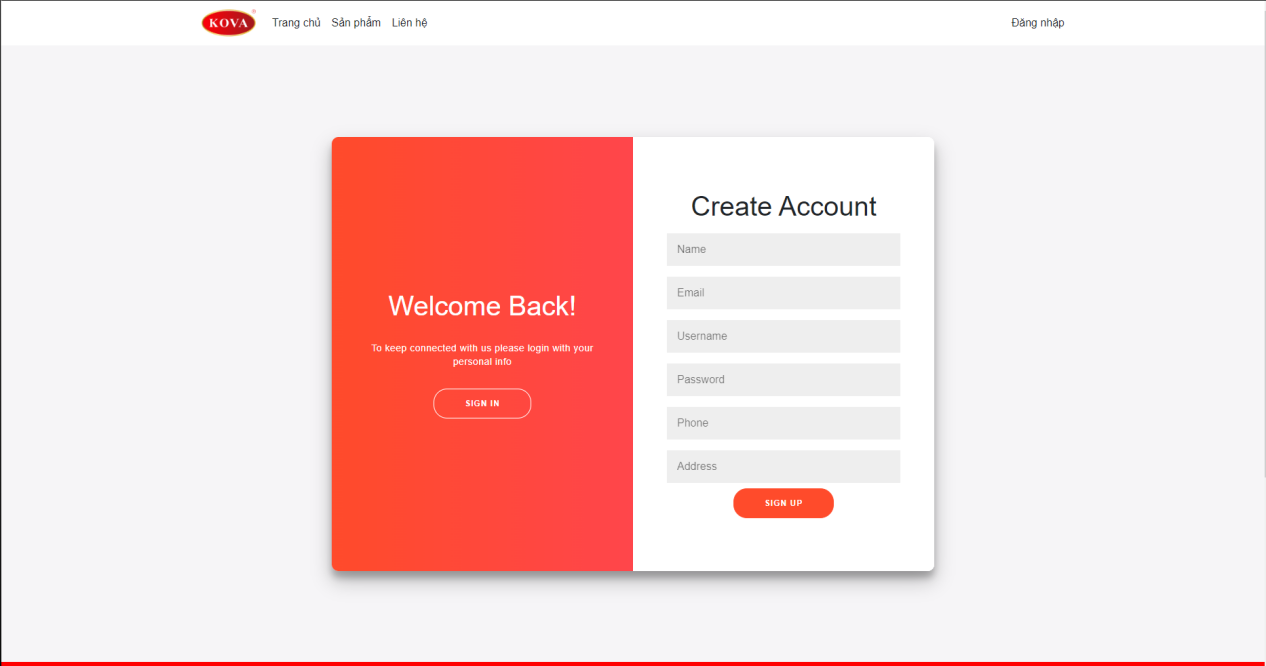




1. **Giao diện và chức năng trang đăng nhập và đăng ký:**

Hiện thị giao diện đăng nhập và khi người dùng muốn đăng ký sẽ bấm vào nút “SIGN UP” để hiển thị phần đăng ký và ngược lại.





1. **Giao diện Admin và các chức năng:**

- Gồm hiển thị giao diện;

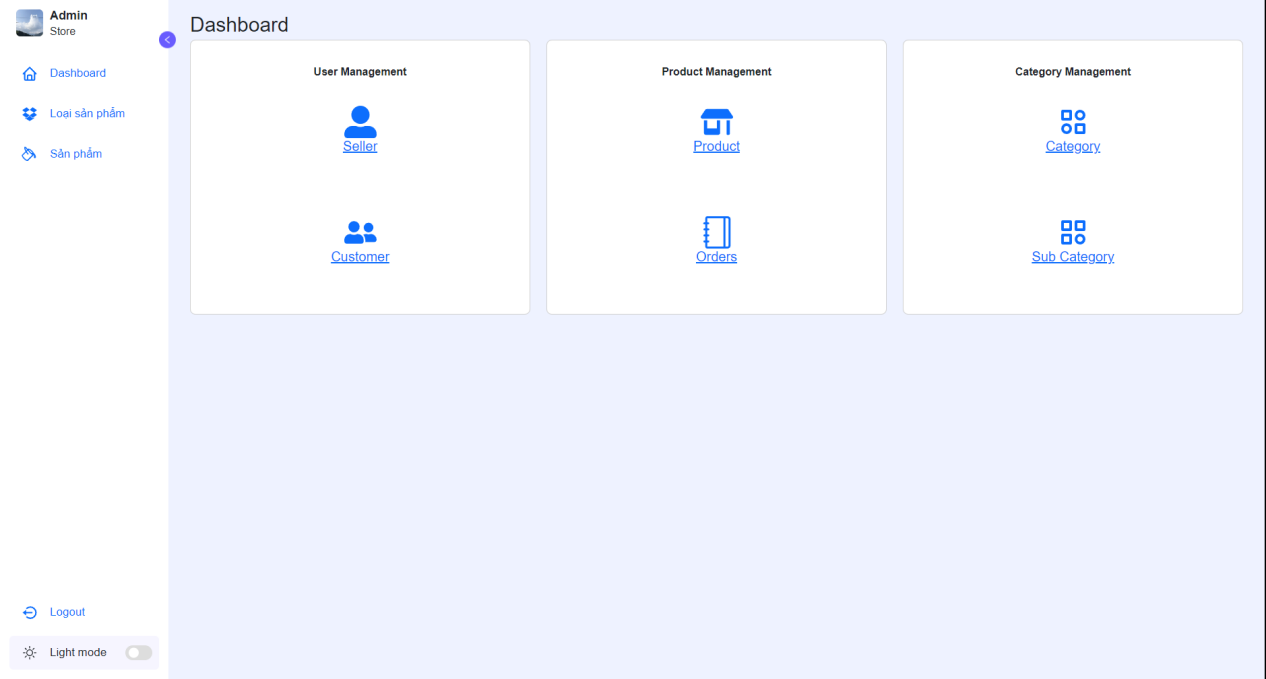
- Các chức năng điều hướng đến các trang quản lý tương ứng;

- Chức năng thu nhỏ/ phóng to sidebar;

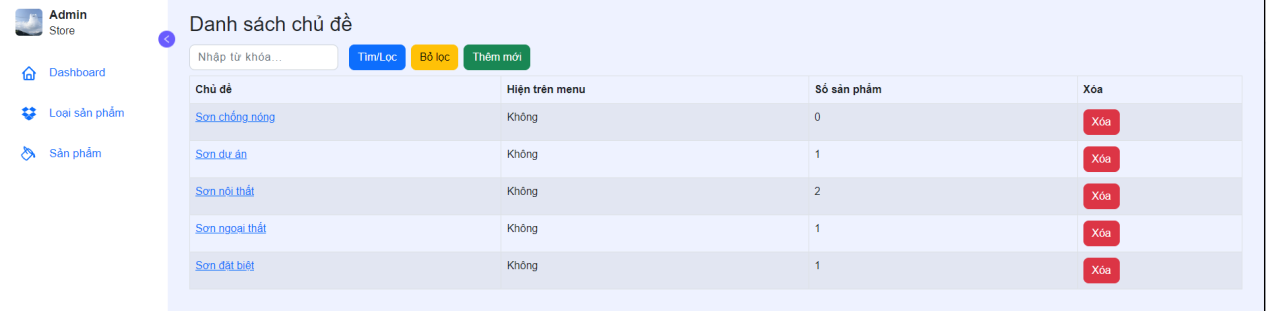
- Chức năng đăng xuất;

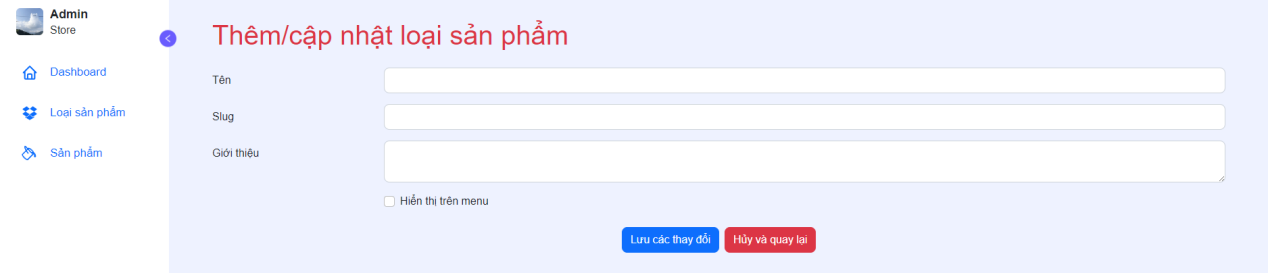
- Chức năng chuyển sang Dark mode và ngược lại;

* Trang dashboard: Quản lý thông tin cửa hàng

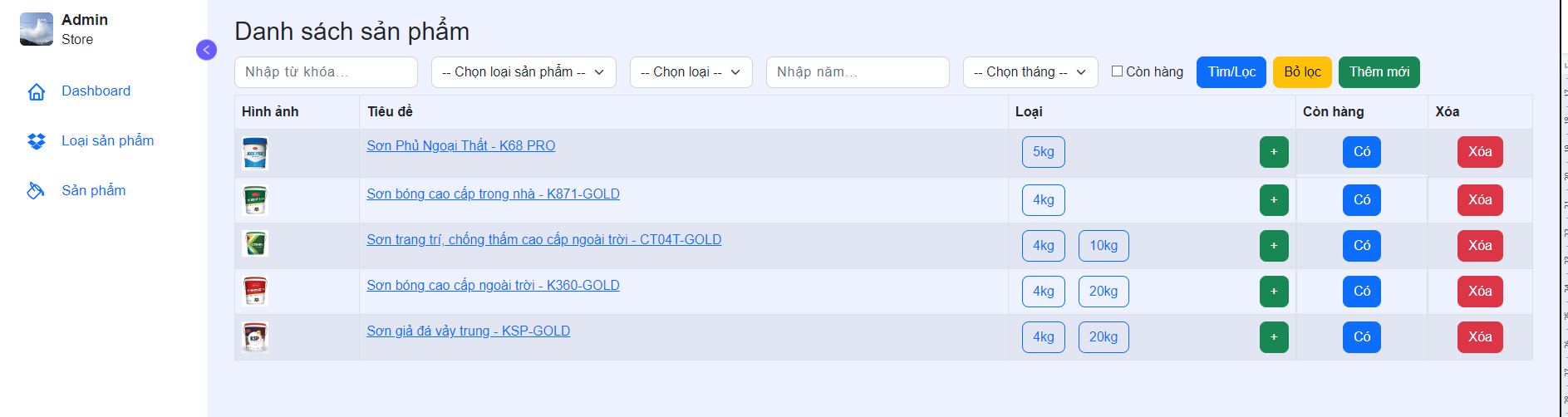


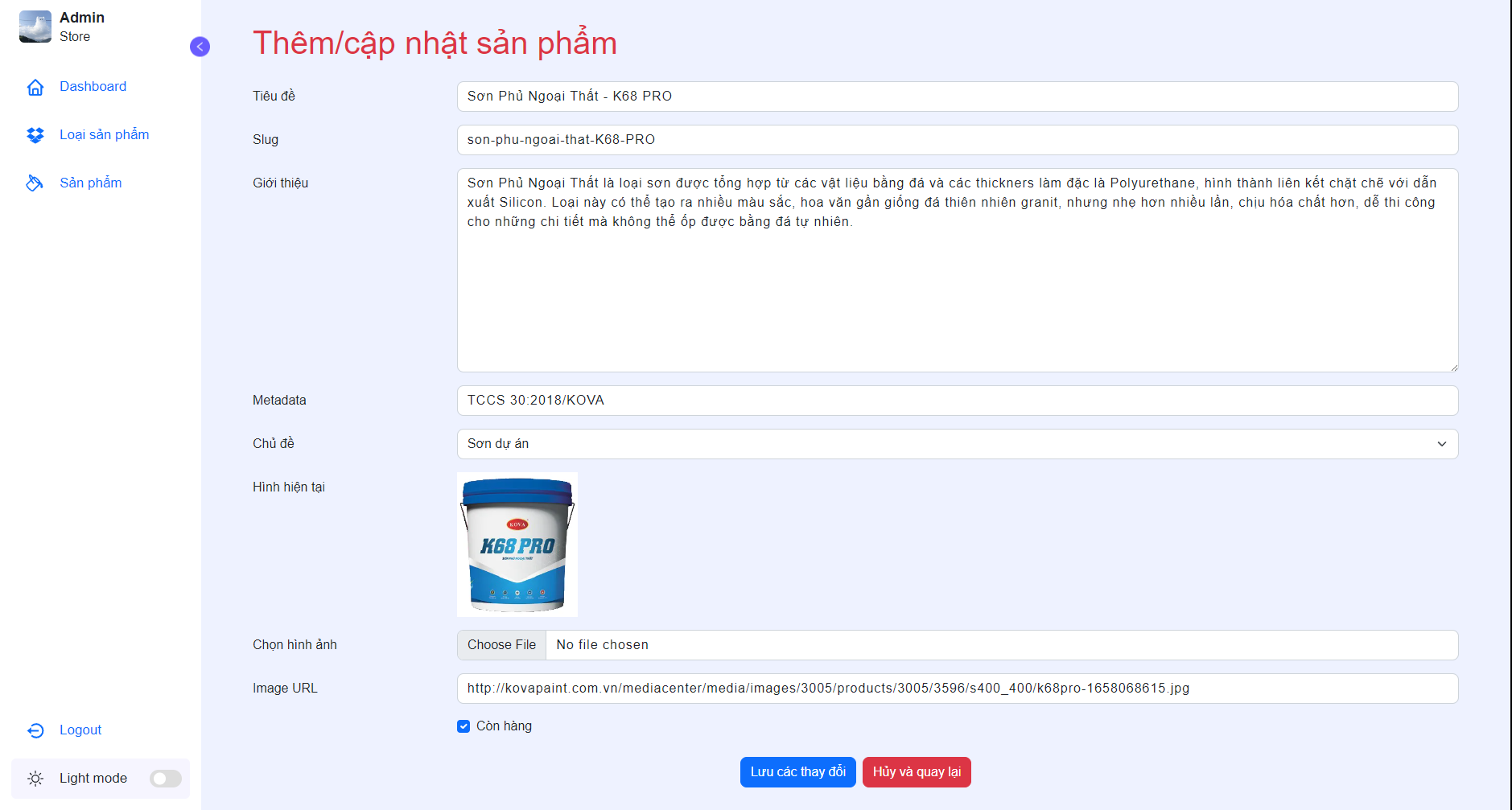
* Trang loại sản phẩm: Quản lý thông tin danh mục và các chức năng như hiện thị, thêm sửa, xóa. Tìm kiếm theo từ khóa.

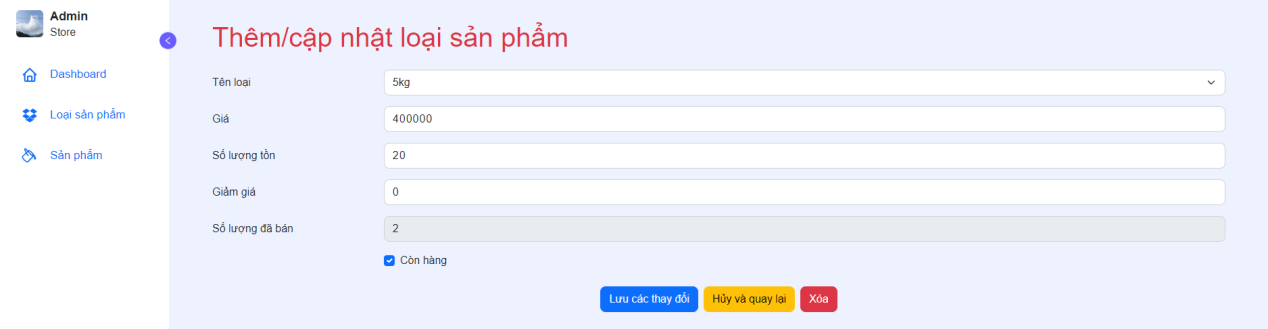




* Trang sản phẩm: Quản lý thông tin sản phẩm và các chức năng như hiện thị, thêm sửa, xóa. Tìm kiếm theo từ khóa. Đồng thời quản lý thông tin loại sản phẩm ứng với mỗi sản phẩm tương ứng.







1. **Hướng dẫn cài đặt và chạy chương trình**
2. **Yêu cầu phần mềm**

- Máy tính có cài đặt **nodejs** phiên bản **18.13** trở lên và **npm** phiên bản **8.19** trở lên

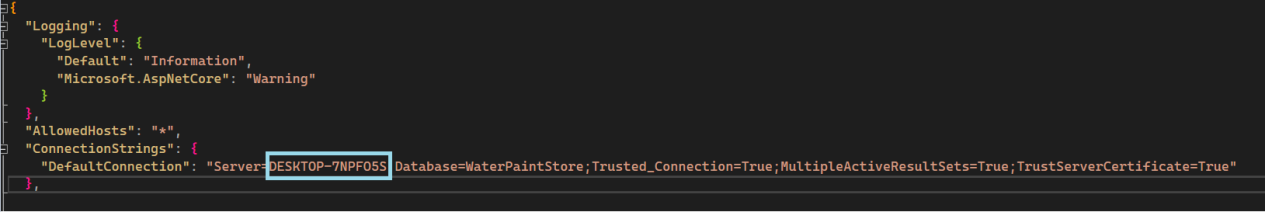
- Hệ quản trị cơ sở dữ liệu **SQL Server 2019**

- Phần mềm **Visual Studio 2022** để biên dịch và chạy mã nguồn.

1. **Các bước thực hiện**
   1. **Chỉnh sửa tên server cơ sở dữ liệu**

Mở dự án theo đường dẫn **/src/WebWaterPaintStore.sln** bằng phần mềm **Visual Studio 2022.**

Mở tập tin **appsettings.json** trong dự án chương trình **WebWaterPaintStore.API** sau đó chỉnh sửa tên của server cơ sở dữ liệu phù hợp với máy.



* 1. **Chọn dự án khởi chạy mặt định**

Click chuột phải vào dự án **WebWaterPaintStore.WebApi** và chọn **Set as Startup Project**

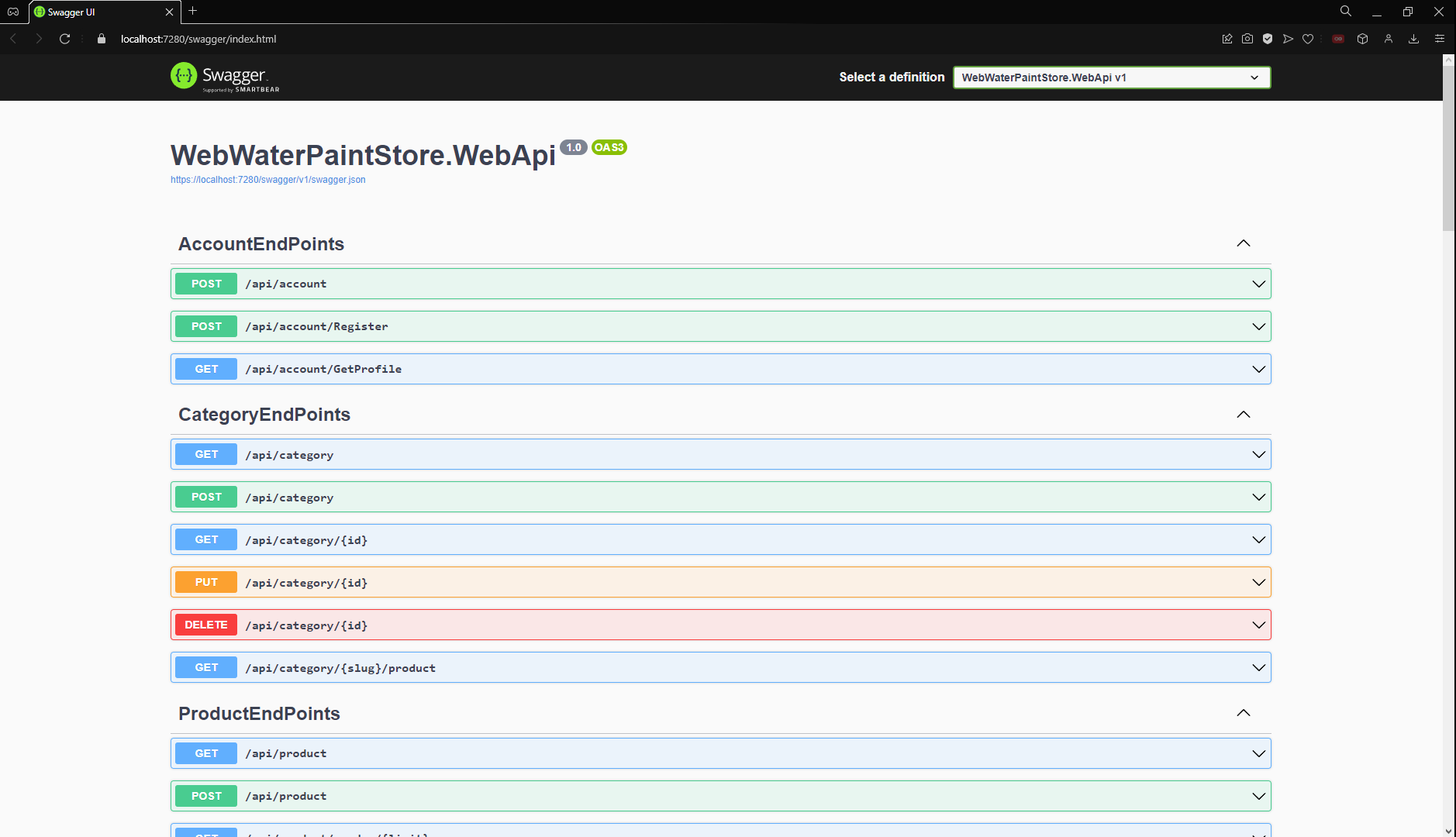


* 1. **Chạy chương trình và thay đổi địa chỉ API Endpoint**

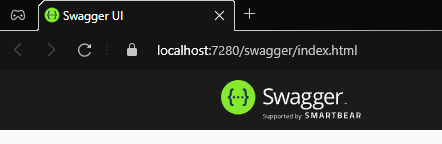
Nhấn tổ hợp phím **Ctrl + F5** hoặc nút **https** trên thanh công cụ để biên dịch và chạy chương trình Api.



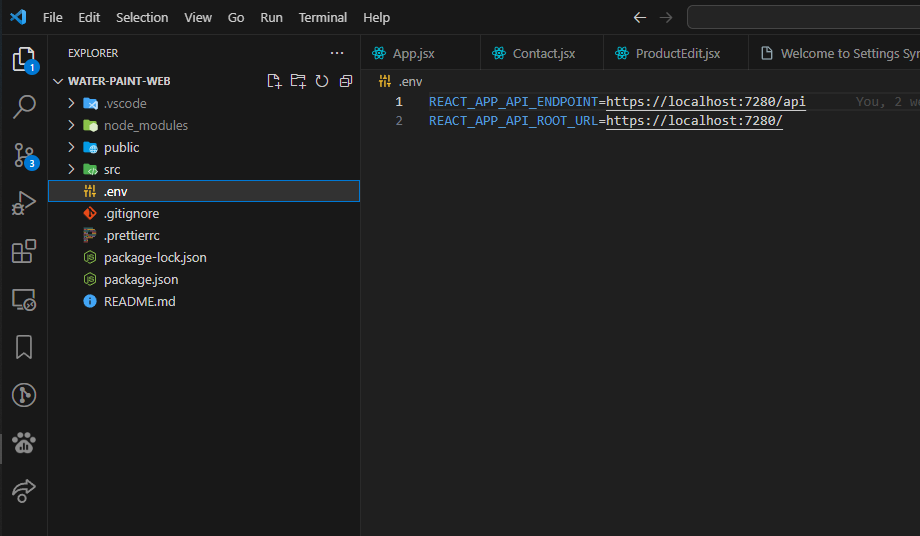
Một trình duyệt sẽ tự động mở một trang web có giao diện tương ứng như sau.



Quan sát thanh url sẽ thấy sau phần localhost: là phần 7280 (Có thể khác nhau).

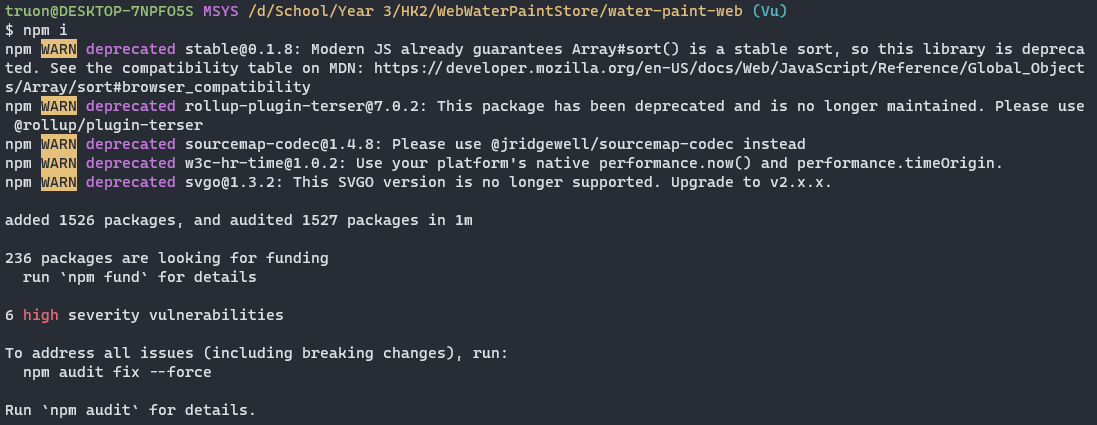


Thay thế số sau localhost: từ url vào file **.env** từ đường dẫn **/water-paint-web/.env**



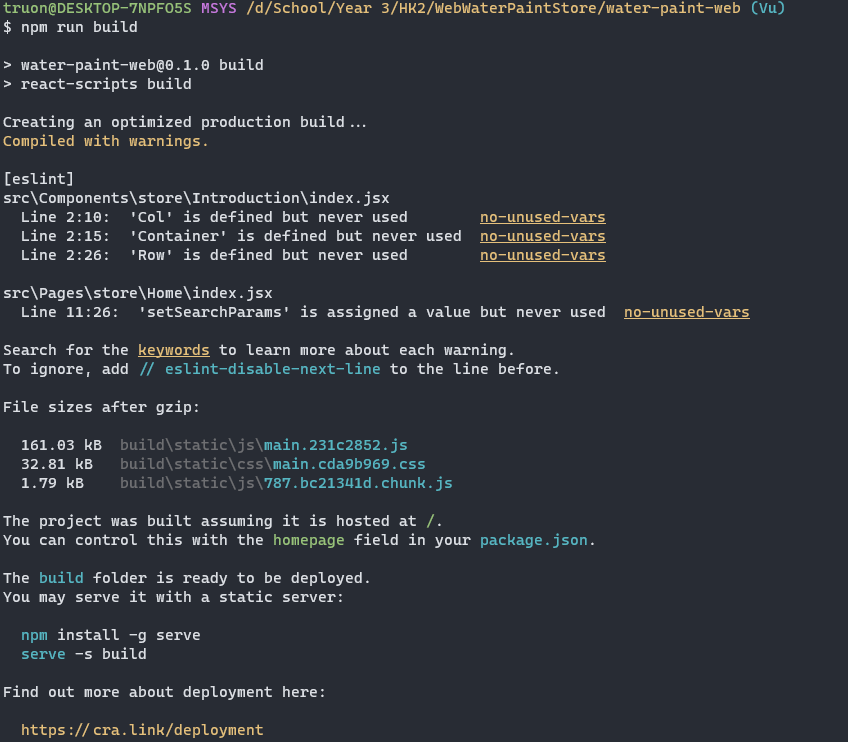
* 1. **Cài đặt các gói tin**

Mở cửa sổ dòng lệnh tại thư mục **/water-paint-web/** và gõ lệnh **npm install** hoặc **npm i** và nhấn phím Enter để cài đặt tất cả các gói thư viện.



* 1. **Biên dịch chương trình**

Sau khi cài đặt hoàn tất, gõ lệnh **npm run build** và nhấn phím **Enter** để biên dịch dự án.



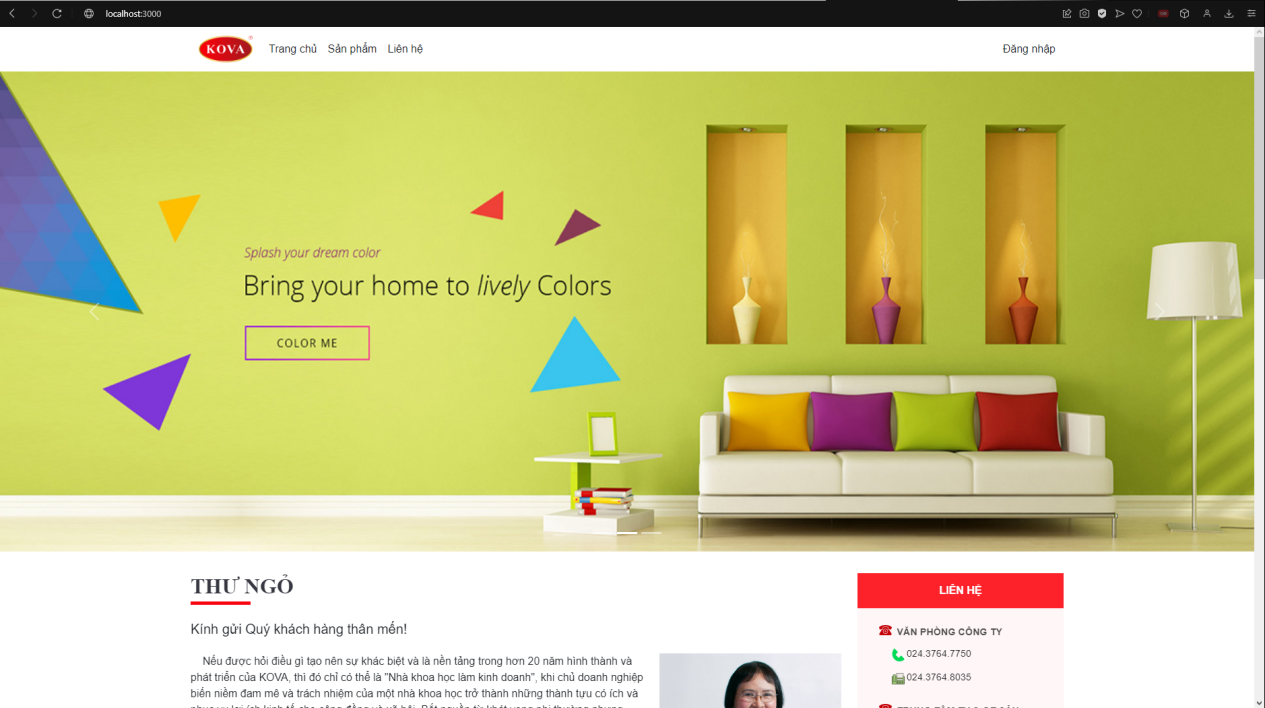
* 1. **Chạy chương trình**

Sau khi biên dịch hoàn tất, gõ lệnh **npm run start** và nhấn phím **Enter** để chạy chương trình.



Trình duyệt web sẽ mở và truy cập vào địa chỉ tương ứng trong cửa sổ dòng lệnh, trong trường hợp này là **http://localhost:3000**

Kết quả của chương trình:



* 1. **Đăng nhập:**

Có thể sử dụng tài khoản của Admin như sau:

Tài khoản: **admin**

Mật khẩu: **admin#123**

Hoặc đăng ký một tài khoản mới (Không thể truy cập vào trang admin do không có quyền).

1. **Kết luận và hướng phát triển**
2. **Kết luận:**

Trong quá trình làm và hoàn thiện các tính năng đã giúp cho bọn em có thêm kiến thực môn học và đạt được các thành quả như sau:

* Tìm hiểu sâu hơn quá trình hình thành một trang web;
* Tìm hiểu tổng quan về React và các cách thức xây dựng và phát triển ứng dụng;
* Hiểu được cách thức xây dựng và làm việc với API bằng ASP.Net core, sử dụng Minimal API để dựng các endpoint.

1. **Hướng phát triển:**

* Xây dựng và hoàn thiện các chức năng đã nêu ở phần ghi chú mục “Phân công nhiệm vụ”;
* Cải thiện và sửa lỗi hệ thống.

1. **Bảng phân công công việc**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Người thực hiện** | **Công việc** | **Mức độ hoàn thành (%)** |
| **1** | Nguyễn Thị Huế | Giao diện cho trang chủ và sản phẩm | 100% |
| **2** | Nghiêm Xuân Anh Quân | Giao diện cho trang tin tức | 0% |
| **3** | Nguyễn Trường Vũ | Tất cả các phần còn lại | 80% |

Ghi chú:

- Do lần đầu làm trưởng nhóm nên em chưa phân công công việc tốt mà vào làm luôn phần API và chỉ kêu các bạn học React để làm phần giao diện.

- Phần Admin:

* Chưa hoàn thành dashboard;
* Chưa xây dựng quản lý thông tin người dùng (kể cả phần API)
* Chưa xây dựng quản lý thông tin đặt hàng (kể cả phần API)
* Chưa hoàn thành phần giao diện Dark mode.

- Phần trang cửa hàng:

* Tính đăng nhập chưa có refresh token, khi đăng nhập thành công thì không điều hướng ra trang home.
* Chưa xây dựng trang order.
* Chưa xây dựng trang quản lý thông tin khách hàng.
* Chưa làm tính năng quản mật khẩu.

**-- HẾT --**