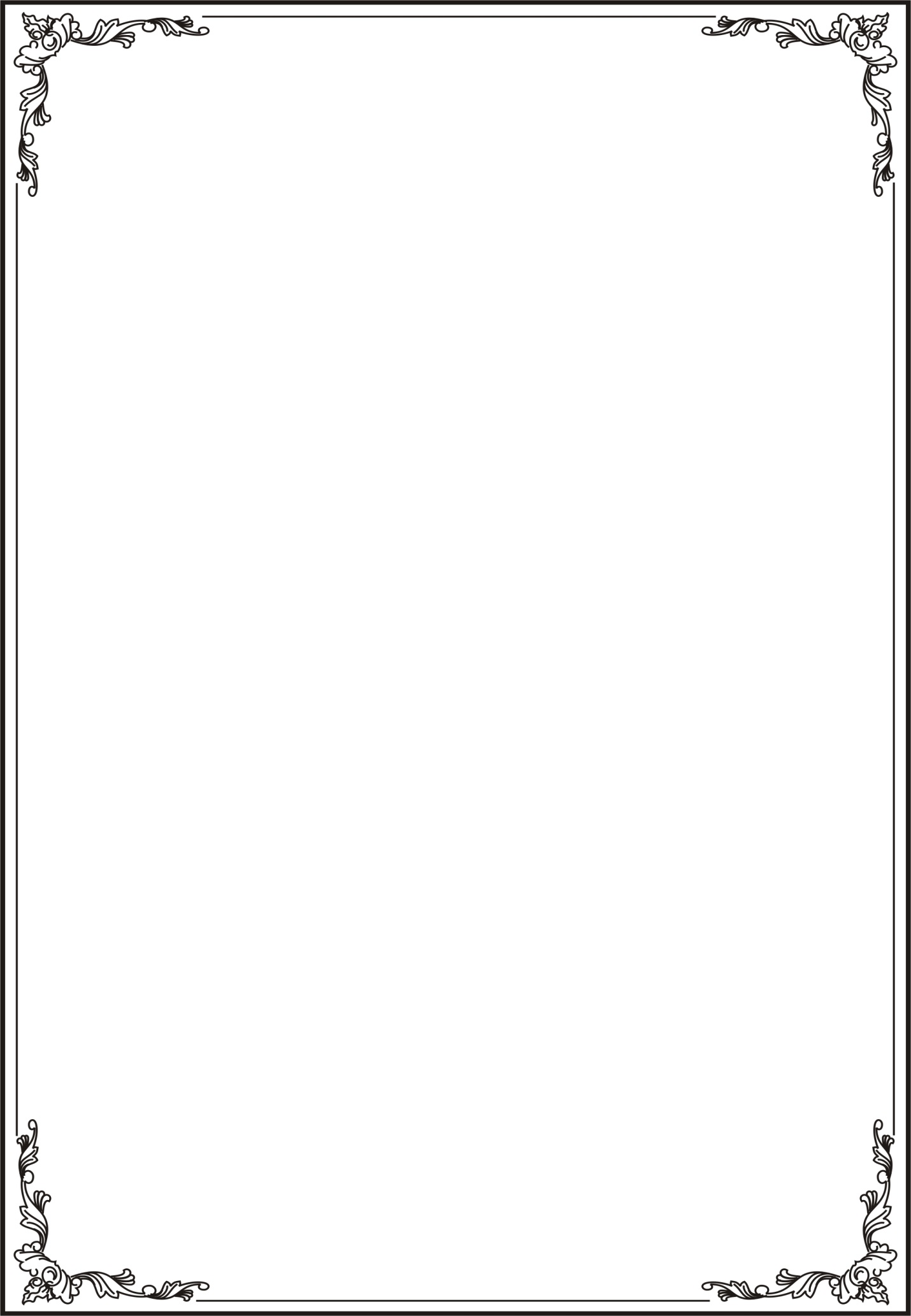
****

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

---------------🙦 🕮 🙤---------------

|  |  |  |
| --- | --- | --- |
| **Môn** | **:** | **Công Nghệ Phần Mềm Mới** |
| **GVHD** | **:** | **ThS. Nguyễn Minh Đạo** |
|  |  |

**XÂY DỰNG WEBSITE THƯƠNG MẠI ĐIỆN TỬ VỀ ĐỒ NỘI THẤT VÀ VẬT DỤNG GIA ĐÌNH**

***Đề tài:***

*TP. Hồ Chí Minh, tháng 12, năm 2022*

**PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

Họ tên sinh viên:Hà Văn Nhân MSSV: 20110531

**Giảng viên hướng dẫn:** ThS. Nguyễn Minh Đạo

Tên đề tài: Xây dựng website thương mại điện tử về đồ nội thất và vật dụng gia đình.

**NHẬN XÉT**

1. Về nội dung đề tài & khối lượng thực hiện:

2. Ưu điểm:

3. Khuyết điểm:

4. Đánh giá loại:

5. Điểm: (bằng chữ …………………………………………………………...)

|  |  |
| --- | --- |
|  | **GIẢNG VIÊN HƯỚNG DẪN** |
|  | (Kí và ghi rõ họ tên) |

**LỜI CẢM ƠN**

Lời đầu tiên, em 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 em thực hiện có cơ hội được tự do tiếp cận, tham khảo, mở rộng thêm kiến thức trong lĩnh vực Công nghệ thông tin.

Lời cảm ơn trân trọng nhất em xin chân thành gửi đến thầy Nguyễn Minh Đạo – người đã tận tình chỉ dẫn, hỗ trợ, luôn đồng hành, trực tiếp hướng dẫn và tạo mọi điều kiện thuận lợi giúp đỡ cho em 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. Trong quá trình xây dựng ứng dụng, vì kiến thức còn hạn chế nên còn gặp nhiều khó khăn nên ứng dụng khó tránh những sai sót. Do đó, em mong sẽ nhận được sự góp ý chân thành của thầy và các bạn đọc. Những ý kiến đóng góp về đề tài sẽ được em tiếp nhận để có thêm nhiều kinh nghiệm phát triển những nghiên cứu khác tốt hơn.

Cuối cùng em thực hiện kính chúc thầy dồi dào sức khỏe, niềm tin để tiếp tục sự nghiệp cao quý của mình là truyền đạt kiến thức cho thế hệ mai sau.

Em thực hiện xin chân thành cám ơn.

Mục lục

[CHƯƠNG 1 : TỔNG QUAN 1](#_Toc121606528)

[1.1 Lý do chọn đề tài 1](#_Toc121606529)

[1.2 Mục tiêu của đề tài 1](#_Toc121606530)

[CHƯƠNG 2 : KHẢO SÁT HIỆN TRẠNG 2](#_Toc121606531)

[2.1 Khảo sát hiện trạng 2](#_Toc121606532)

[2.1.1 Công nghệ MERN 2](#_Toc121606533)

[2.1.2 Thực Trạng Website E-commerce 2](#_Toc121606534)

[2.1.3 Những thứ cần có của một website thương mại điện tử bán hàng chuyên nghiệp hiệu quả 2](#_Toc121606535)

[2.1.4 Tính Năng Cần Có Của Website E-commerce 3](#_Toc121606536)

[CHƯƠNG 3 : CÔNG NGHỆ SỬ DỤNG 4](#_Toc121606537)

[**3. Ưu điểm, nhược điểm của MERN Stack** 4](#_Toc121606538)

[**1.** **Ưu điểm** 4](#_Toc121606539)

[**2. Nhược điểm** 4](#_Toc121606540)

[3.1 Kiến trúc thành phần ứng dụng 4](#_Toc121606541)

[3.2 Front-end (ReactJS) 6](#_Toc121606542)

[3.2.1 Cấu trúc thư mục 6](#_Toc121606543)

[3.2.2 Các thư viện sử dụng 9](#_Toc121606544)

[3.2.3 Đặc tả một số thư viện chính 11](#_Toc121606545)

[3.3 Back-end (NodeJS, ExpressJS) 19](#_Toc121606546)

[3.3.1 Cấu trúc thư mục 19](#_Toc121606547)

[3.3.2 Các thư viện sử dụng 22](#_Toc121606548)

[3.3.3 Đặc tả một số thư viện chính 24](#_Toc121606549)

[CHƯƠNG 4 PHÂN TÍCH DỮ LIỆU – TÍNH NĂNG ĐỀ ÁN 28](#_Toc121606550)

[4.1 Mô hình hóa yêu cầu 28](#_Toc121606551)

[4.1.1 Lược đồ Usecase 28](#_Toc121606552)

[CHƯƠNG 5 CÁCH SỬ DỤNG – CÀI ĐẶT 31](#_Toc121606553)

[**5.1 Cài đặt** 31](#_Toc121606554)

[**5.2 Hướng dẫn sử dụng website** 31](#_Toc121606555)

[**5.2.1 Các chức năng chính đối với quyền admin** 32](#_Toc121606556)

[**5.2.2 Các chức năng chính đối với quyền User** 36](#_Toc121606557)

[**5.2.3 Chức năng sử dụng trực tiếp khi truy cập website mà không cần đăng nhập tài khoản** 40](#_Toc121606558)

[TÀI LIỆU THAM KHẢO 46](#_Toc121606559)

# : TỔNG QUAN

## Lý do chọn đề tài

Như chúng ta cũng thấy trong thị trường hiện nay thì việc cạnh tranh về kinh doanh ngày càng trở nên quyết liệt và hầu hết những nhà kinh doanh, những công ty lớn đều rất chú tâm đến việc làm thoả mãn khách hàng một cách tốt nhất. So với kinh doanh truyền thống thì TMDT chi phí thấp hơn, hiệu quả đạt cao hơn. Hơn thế nữa, với lợi thế của công nghệ Internet nên việc truyền tải thông tin về sản phẩm nhanh chóng, thuận tiện.

Theo nghiên cứu thị trường thương mại điện tử nửa đầu năm 2022 của Công ty nghiên cứu dữ liệu Metric.vn, Việt Nam đang trở thành thị trường lớn thứ 2 trong khu vực Đông Nam Á, chỉ đứng sau Indonesia.

So với [thực trạng thương mại điện tử](https://magenest.com/vi/thuong-mai-dien-tu-tren-the-gioi/) thế giới, Việt Nam có mức tăng trưởng khá mạnh. Theo Statista, tốc độ phát triển thương mại điện tử toàn cầu có tốc độ phát triển là 16,24% vào năm 2021 và dự báo bứt phá lên 24.5% vào năm 2025. Còn tại Việt Nam, con số này năm 2021 là hơn 20%, với quy mô 16 tỷ USD. Đến năm 2025, tốc độ phát triển của thương mại điện tử ở nước ta có thể lên đến 29%, đạt 39 tỷ USD.

Năm 2022, thương mại điện tử tại Việt Nam được các chuyên gia dự báo sẽ tiếp tục phát triển mạnh mẽ, tạo bước đà cho phát triển kinh tế. Đây cũng là thời điểm để doanh nghiệp Việt Nam xây dựng những chiến lược và mô hình mới, từ đó khôi phục lại doanh nghiệp và mở rộng thị trường sau đại dịch COVID-19.

Trước thực tế đó chúng em đã chọn đề tài xây dựng website bán hàng trực tuyến cho website của mình

## Mục tiêu của đề tài

Mục tiêu của đề tài là có thể áp dụng các kiến thức đã học và áp dụng những công nghệ mới để có thể xây dựng lên một nền tảng website thương mại điện tử về đồ nội thất và vật dụng gia đình. Ngoài ra, giao diện phải vừa thân thiện, vừa dễ sử dụng cũng như đầy đủ các chức năng mang lại sự thoải mái tốt nhất cho người sử dụng, cũng như là người quản lý website.

# : KHẢO SÁT HIỆN TRẠNG

## Khảo sát hiện trạng

### Công nghệ MERN

MERN (MongoDB, Express, React, Node) là hai stack mạnh mẽ đáng kinh ngạc trong lĩnh vực phát triển ứng dụng. Tất cả bốn công nghệ này cung cấp một khuôn khổ hoàn chỉnh cho các nhà phát triển để tạo ra bất kỳ ứng dụng web nào. Các công nghệ này đang được phát triển nhanh chóng các framework và web mã nguồn mở. Những công nghệ được sử dụng bởi một số công ty công nghệ lớn nhất như Netflix, thúc đẩy sự phổ biến của nó. Toàn bộ trang web của Netflix dựa trên MERN để có trải nghiệm lướt web mượt mà, chất lượng tốt

Các công nghệ mới trong MERN:

Hot Reloading cho các React Components

Các đoạn mã được phân chia bởi React Router

* Hỗ trợ đa ngôn ngữ
* Hỗ trợ generate code
* Cấu trúc kiểu mô-đun
* Ava Test Runner
* Hỗ trợ Docker
* Có thể tùy biến phiên bản MERN cho riêng bạn

### Thực Trạng Website E-commerce

Nghiên cứu và khảo sát hiện trạng là bước mở đầu của quá trình phát triển hệ thống. Đối với website bán hàng ,vấn đề đặt hàng của khách hàng, vấn đề đặt hàng, quản lý thông tin hồ sơ khách hàng, quản lý dịch vụ cũng như quản lý hóa đơn, thống kê báo cáo doanh thu của cửa hàng.

Website bán hàng online - Thương Mại Điện Tử hiện đang trở thành xu hướng dẫn đầu tại Việt Nam đặc biệt trong thời đại công nghệ thông tin hiện nay.

Các trang web hương mại điện tử sẽ bùng nổ mạnh mẽ hơn nữa trong thời gian tới. Khi mà Internet đang ngày càng phủ sóng toàn cầu thì việc mua sắm sản phẩm online/dịch vụ online sẽ ngày càng trở nên phổ biến cho nên việc phát triển website E-commerce là rất cần thiết và phù hợp với nhu cầu hiện nay

### Những thứ cần có của một website thương mại điện tử bán hàng chuyên nghiệp hiệu quả

- Giới thiệu sản phẩm, dịch vụ trực quan

- Xây dựng hình ảnh chuyên nghiệp

- Thao tác tìm kiếm mua hàng một cách dễ dàng

- Giảm sự phụ thuộc vào các nền tảng khác

- Tối ưu chi phí quảng cáo

- Tạo dựng thương hiệu trên thị trường khi thiết kế web thương mại điện tử

### Tính Năng Cần Có Của Website E-commerce

Một mỗi lĩnh vực sẽ yêu cầu các tính năng phù hợp và không ngoại lệ, website thương mại điện tử sẽ cần có các tính năng như sau:

- Chức năng Responsive: Một website có khả năng tương thích với thiết bị di động như mobile, table, laptop....đem tới cho người dùng trải nghiệm tuyệt vời nhất.

- Giao diện phù hợp với sản phẩm, dịch vụ: Bạn cần lựa chọn giao diện website sẽ tạo ra được một trang web cung cấp sản phẩm, dịch vụ, chuyên nghiệp thân thiện với người dùng.

- Hỗ trợ đa ngôn ngữ: Ngoài Tiếng Việt thì các website thương mại điện tử nên hỗ trợ đa ngôn ngữ để tiếp cận nhiều đối tượng khách hàng hơn.

- Dễ dàng cập nhập, điều chỉnh, thay đổi nội dung: Các quản trị viên, bạn có quyền thay đổi nội dung, chỉnh sửa, xóa sửa hình ảnh, video, sản phẩm....

- Chức năng thêm, bớt, quản lý sản phẩm dịch vụ: Admin website có quyền thêm không hạn chế số lượng sản phẩm, tạo các bộ lọc tìm kiếm phân loại sản phẩm một cách khoa học.

- Cung cấp phương thức thanh toán đa dạng: Với tính năng thanh toán một website thương mại điện tử cần cho phép người dùng tự bổ sung thêm các cổng thành toán đa dạng như COD, Thanh toán qua thẻ visa/mastercard, Ví điện tử Momo, VNPAY....

- Tích hợp các mạng xã hội: Việc tích hợp các mạng xã hội như Facebook, Twitter, Youtube, Instagram... sẽ giúp tăng hiệu quả bán hàng.

- Hỗ trợ tích hợp với phương thức giao hàng, vận chuyển: Đa phần các sàn giao dịch thương mại điện tử lớn như Shopee, Lazada, Sendo, Tiki có khả năng kết nối với các công ty vận chuyển giao hàng rất hiệu quả. Và nếu doanh nghiệp bạn sẽ cần tích hợp và cố gắng hiển thị đầy đủ thông tin nhất dành cho khách hàng.

- Hỗ trợ tối ưu SEO: Website thương mại điện tử cần được thiết kế tối ưu về tốc độ tải trang, các sản phẩm, hình ảnh, video.... giúp website có được thứ hạng cao trên công cụ tìm kiếm.

- Bảo mật hệ thống: Website cần sử dụng các chứng chỉ bảo mật SSL để đảm bảo an toàn cho khách hàng khi giao dịch trên website. Ngoài ra các công ty nên lựa chọn hosting chất lượng, uy tín để hạn chế website bạn tấn công, nhiễm mã độc.

- Đánh giá sản phẩm/dịch vụ: Khách hàng thường có thói quen đánh giá, bình luận mua sản phẩm dịch vụ điều đó sẽ tạo được lòng tin cho khách hàng khi chọn mua sản phẩm của website bạn.

- Tốc độ tải trang và xử lý nhanh: Website thương mại điện tử cần có khả năng xử lý các giao dịch nhanh,, thời gian tải trang một cách hiệu quả, tối ưu mã nguồn của website.

# : CÔNG NGHỆ SỬ DỤNG

**3. Ưu điểm, nhược điểm của MERN Stack**

1. **Ưu điểm**

* Để phát triển trơn tru bất kỳ ứng dụng web hoặc ứng dụng di động nào, nó hỗ trợ kiến trúc MVC (Bộ điều khiển chế độ xem mô hình); Mục đích chính của kiến trúc này là tách các chi tiết trình bày với logic kinh doanh.
* Nó bao gồm tất cả các giai đoạn phát triển web bắt đầu từ phát triển front-end đến phát triển backend bằng JavaScript.
* Nó là một khuôn khổ mã nguồn mở chủ yếu được sử dụng để phát triển các ứng dụng dựa trên web hoặc di động và được cộng đồng hỗ trợ.
* Nó rất nhanh và hiệu quả so với MEAN Stack và chủ yếu phù hợp cho các ứng dụng nhỏ, trong khi MEAN Stack phù hợp để phát triển các ứng dụng lớn.

**2. Nhược điểm**

- Đều sử dụng các công nghệ mới nên còn gặp nhiều khó khăn trong việc triển khai và phát triển website

- Sự tiện lợi của việc sử dụng thư viện của bên thứ ba là một vấn đề khác cần phải suy nghĩ khi tham gia vào các ứng dụng cấp doanh nghiệp. Trong đó MERN yêu cầu các cấu hình bổ sung để thực hiện thêm các tiện ích mở rộng của bên thứ ba

- MERN phù hợp phát triển các ứng dụng nhỏ hơn là phát triển các ứng dụng lớn.

## Kiến trúc thành phần ứng dụng



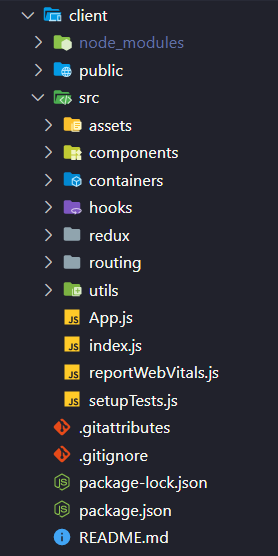
Hình 4.1: Kiến trúc thành phần ứng dụng

Ứng dụng gồm hai thành phần chính:

* Front-end: sử dụng công nghệ ReactJS với ngôn ngữ Javascript, sử dụng khái niệm Virtual DOM, nơi các thành phần (component) tồn tại giúp cải thiện hiệu suất ứng dụng. Sử dụng cú pháp JSX cho phép viết HTML ngay trên file javascript, giúp viết code dễ hiểu và dễ dàng hơn. React giúp tách các thành phần phức tạp của một trang web thành nhiều thành phần nhỏ hơn, chia nhỏ cấu trúc UX/UI phức tạp thành từng component đơn giản hơn. Ngoài ra, ReactJS hiệu năng cao với các ứng dụng có dữ liệu thay đổi liên tục, dễ dàng bảo trì, nâng cấp và sữa lỗi.
* Back-end: sử dụng công nghệ NodeJS kết hợp với ExpressJS framework, với sự hỗ trợ các phương thức HTTP và các middleware tạo ra các Rest API nhanh chóng, mạnh mẽ, dễ sử dụng và có khả năng tái sử dụng. Khả năng định nghĩa routing giữa các routes và các request method đến server dễ dàng. Ở phía cơ sở dữ liệu, đồ án sử dụng MongoDB bởi cấu trúc linh hoạt, tốc độ ghi và cập nhật nhanh, tiện lợi cũng như dữ liệu dễ nhìn và trực quan hơn. Ngoài ra, để phục vụ cho chức năng upload ảnh, đồ án sử dụng Drive giúp việc lưu ảnh nhanh chóng, có thể đồng bộ trực tiếp với website và kết hợp với Google API/OAuth2Client để gửi thực hiện các tác vụ liên quan đến việc gửi email cho người dùng (quên mật khẩu, xác nhận thanh toán…)

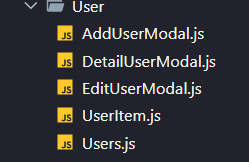
## Front-end (ReactJS)

### Cấu trúc thư mục



Hình 4.2: Cấu trúc thư mục front-end

Trang web ứng dụng công nghệ ReactJS, các components được phân nhóm theo module/feature, tất cả các thành phần chức năng đều ở một thư mục liên quan cụ thể đến chức năng của nó cho toàn bộ ứng dụng bao gồm: components, redux (actions, reducers, constants), containers. Ví dụ: các tính năng liên quan đến user sẽ ở trong thư mục User



Hình 4.3: Cấu trúc một thư mục chức năng (User)

Ứng dụng sử dụng công nghệ redux để quản lý state, giúp việc xử lý các state nhanh và dễ dàng hơn. Sử dụng thư viện react-router-domv6 để set up routing và thư viện react-bootstraps để thiết kế và xây dựng các giao diện người dùng.

Bảng 4.1: Chi tiết cấu trúc thư mục front-end

|  |  |  |
| --- | --- | --- |
| **Thư mục / File** | **Cấu trúc thư mục** | **Mô tả các file** |
| node\_module | … | Nơi lưu trữ các module đã được tải xuống. |
| public |  | Thư mục được tạo ra bới React, chứa file hệ thống |
| assets |  | Nơi chứa tất cả các tài nguyên của ứng dụng: file css, hình ảnh, thư viện icon, json data… |
| components |  | Nơi chức các folder bao gồm các UI component và routs ứng dụng. |
| containers |  | Nơi chứa các component chức năng của ứng dụng. Nơi xử lý dữ liệu, kết nối redux, xác định giữa state và props actions |
| redux |  | Nơi chứa các thư mục actions, constants, reducers, xử lý và quản lý các state của ứng dụng |
| utils |  | Thư mục define các hàm mà ứng dụng có thể tái sử dụng lại, bao gồm các hàm chuyển đổi, tính toán… |
| .gitignore |  | Nơi liệt kê những file mà mình không mong muốn cho vào git. |
| package-lock.json |  | Nơi lưu lại các phiên bản của thư viện của ứng dụng. |
| package.json |  | Chứa thông tin các dependencies cần cài đặt trong ứng dụng. |

### Các thư viện sử dụng

Bảng 4.2: Các thư viện sử dụng ở front-end

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Phiên bản** | **Licence** | **Mục đích** | **Áp dụng trong**  **dự án** |
| **Dependencies** | | | | | |
| 1 | apexcharts | 3.30.0 | ISC | Thư viện hỗ trợ cho việc vẽ biểu đồ | Compoent Dashboard.js |
| 2 | axios | 0.23.0 | ISC | thư viện HTTP Client dựa trên Promise, cung cấp một API cho việc xử lý XHR (XMLHttpRequests) | Trong những file call api trong thư mục redux/action |
| 3 | file-saver | 2.0.5 | ISC | Dùng để lưu file bên client-side | ExportToExel.js |
| 4 | lodash | 4.17.21 | ISC | Giúp viết code javascript ngắn gọn và dễ bảo trì hơn | Những file xử lý với array object: Room.js, Dashboard.js |
| 5 | moment | 2.29.1 | ISC | Cho object ngày tháng dễ dàng thể hiện hơn | Những hàm xử lý ngày: BookingModal.js, CheckInModal.js |
| 6 | react | 17.0.2 | ISC | Thư viện để xây dựng giao diện người dùng | Ứng dụng tất cả các file có hàm render jsx |
| 8 | react-apexcharts | 1.3.9 | ISC | Thư viện hỗ trợ cho việc xây dựng biểu đồ | Dashboard.js |
| 9 | react-bootstrap | 2.0.0 | ISC | Cung cấp sẵn một bộ các React component | Ứng dụng hầu hết các component có UI |
| 10 | react-datepicker | 4.3.0 | ISC | Component datepicker đơn giản và tái sử dụng cho react | Các file co xử dụng datepicker: BookingModal.js, CheckInModal.js |
| 11 | react-dom | 17.0.2 | ISC | Hỗ trợ định tuyến | Layout.js |
| 12 | react-redux | 7.2.5 | ISC | Thư viện Javascript để quản lý state của ứng dụng | Hầu hết trong các file HOC |
| 13 | react-router-dom | 6.0.2 | ISC | Chứa các bindings cho việc sử dụng react router trên các ứng dụng web | Các file định tuyến: App.js, Layout.js |
| 14 | react-scripts | 4.0.3 | ISC | Bộ script bắt đầu của create-react-app |  |
| 15 | react-select | 5.2.1 | ISC | Trình quản lý Select cho react | Sử dụng trong các file có sử dụng select input: BookingModal.js, CheckInModal.js |
| 16 | react-to-print | 2.14.1 | ISC | Hỗ trợ việc in ra nội dung của components | PrintBill.js |
| 17 | react-toastify | 8.0.3 | ISC | Hỗ trợ tạo thông báo | Chủ yếu trong các file gọi api trong folder redux/action |
| 18 | redux-devtools-extension | 2.13.9 | ISC | Tool được dùng để theo dõi flow của redux | File store.js |
| 19 | redux-thunk | 2.3.0 | ISC | Middleware cho phép viết các Action trả về một function | File store.js |
| 20 | web-vitals | 1.1.2 | ISC | Đo lường các chỉ số Web Vitals | Chưa áp dụng |
| 21 | xlsx | 0.17.4 | ISC | Hỗ trợ đọc file spreedsheet | Trong component Statistic.js |

### Đặc tả một số thư viện chính

#### Axios

Axios là một HTTP client được viết dựa trên Promises được dùng để hỗ trợ cho việc xây dựng các ứng dụng API từ đơn giản đến phức tạp và có thể được sử dụng cả ở trình duyệt hay Node.js.

Axios hỗ trợ: Tạo XMLHttpRequests từ trình duyệt, thực hiện các http request từ node.js, Hỗ trợ Promise API, chặn request và response, chuyển đổi dữ liệu request và response, tự động chuyển đổi về dữ liệu JSON, hỗ trợ phía client để chống lại XSRF.

* ***Cách sử dụng Axios:***
* **Axios GET:**

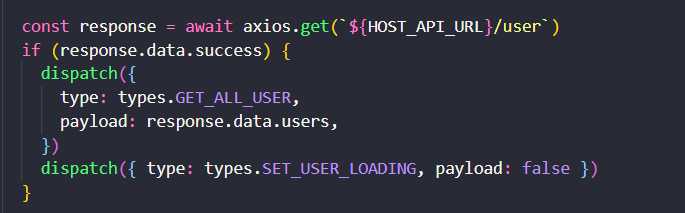
Graphical user interface, text, application

Description automatically generated

Hình 4.4: Cấu trúc đơn giản axios GET

Đoạn mã này gửi một request với phương thức GET đến URL https://jsonplaceholder.typicode.com/users, nếu thành công kết quả sẽ được render ra danh sách person.name và nếu lỗi thì in lỗi ra console trong .catch().

Áp dụng vào dự án:



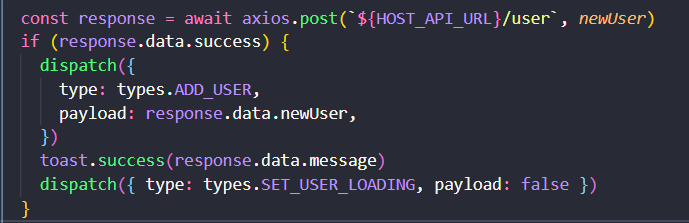
Hình 4.5: Axios GET lấy thông tin người dùng

**Mô tả:** mỗi khi có sự kiện gọi toàn bộ danh sách user, đoạn mã sẽ gửi một request với phương thức GET đến URL HOST\_API\_URL là một API được viết phía Backend, nếu thành công sẽ trả về một danh sách data user, và nếu lỗi thì in lỗi ra console và đồng thời hiển thị một thông báo cho người dùng.

* **Axios POST hoặc PUT:**

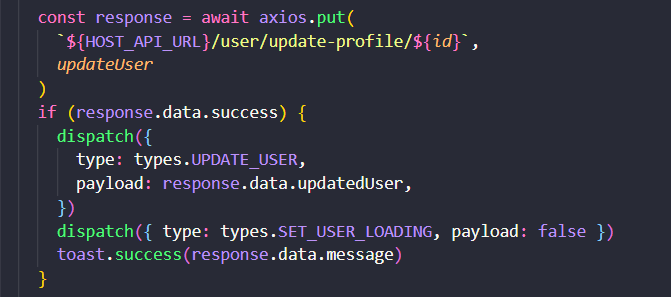
Với việc cho phép người dùng nhập dữ liệu vào form sau đó submit gửi nội dung lên API thì ta có thể sử dụng phương thức POST hoặc PUT.

POST dữ liệu: Đoạn mã dưới đây cho phép thêm một người dùng mới vào database.



Hình 4.6: Axios POST thêm mới người dùng

PUT dữ liệu: Đoạn mã dưới đây cho phép cập nhật thông tin người dùng



Hình 4.7: Axios PUT cập nhật thông tin người dùng

* **Axios DELETE:**

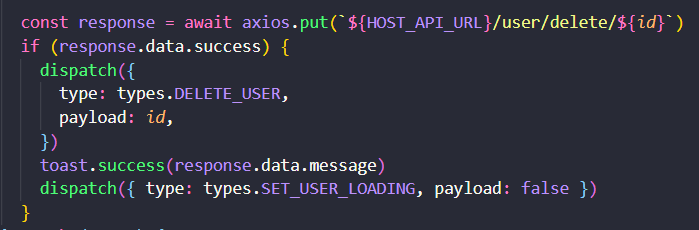
Ngoài ra, axios còn có phương thức DELETE,

Graphical user interface, text, application, email

Description automatically generated

Hình 4.8: Axios DELETE

Cấu trúc sử dụng trong dự án: Sử dụng PUT thay cho DELETE vì để đảm bảo dữ liệu không bị xóa bỏ hoàn toàn trong cơ sở dữ liệu, đảm bảo cho việc back-up cũng như tính toàn vẹn dữ liệu.



Hình 4.9: Axios PUT xóa người dùng

#### React-router-domv6

React không tích hợp sẵn router, thay vào đó, chúng ta cần sử dụng thư viện react-router-dom@v6. Routing là cách web chuyển hướng.

Router và Routes: để quyết định routes chúng ta muốn sử dụng và điều hướng phù hợp chúng ta sẽ sử dụng Route chứa các compents thông qua props element.

Text

Description automatically generated

Hình 4.10: Kiến trúc routing trong react-router-dom@v6

Phiên bản react-router-dom@v6 đã thay đổi cấu trúc của Route như trên, các phiên bản cũ hơn được viết như dưới đây:

Text, letter

Description automatically generated

Hình 4.11: Kiến trúc routing trong react-router-dom phiên bản cũ

#### React-redux

Redux là 1 thư viện Javascript để quản lý state của ứng dụng, thường được sử dụng với React. Việc tìm hiểu Redux nhìn chung khá khó hiểu vì sẽ gặp một loạt các thuật ngữ mới như dispatch, reducer, combineReducers, store… cùng flow khá phức tạp của nó:

Diagram

Description automatically generated

Hình 4.12: Kiến trúc redux

* **Mô tả các thành phần**
* **Store**: Store đơn giản là 1 object chứa tất cả state toàn cục của ứng dụng:

Text

Description automatically generated

Hình 4.13: Thành phần stores trong redux

* **Các Reducer**: 1 reducer tương đương với 1 state nhưng kèm theo các mô tả state sẽ thay đổi như thế nào khi các action khác nhau được gọi.

Text

Description automatically generated

Hình 4.14: Thành phần các reducers trong redux

* Các **dispatcher**: Khi cần dùng 1 action ở component, ta gọi action đó đơn giản bằng cách sử dụng phương thức dispatch của hook useDispatch của react-redux.

Text

Description automatically generated

Hình 4.15: Cách gọi dữ liệu thông qua hook useDispatch

* **Action:** Khi người dùng thực hiện một thao tác dispatch, redux sẽ thực hiện một action tương ứng

Text

Description automatically generated

Hình 4.16: Thành phần action trong redux

#### React-bootstrap

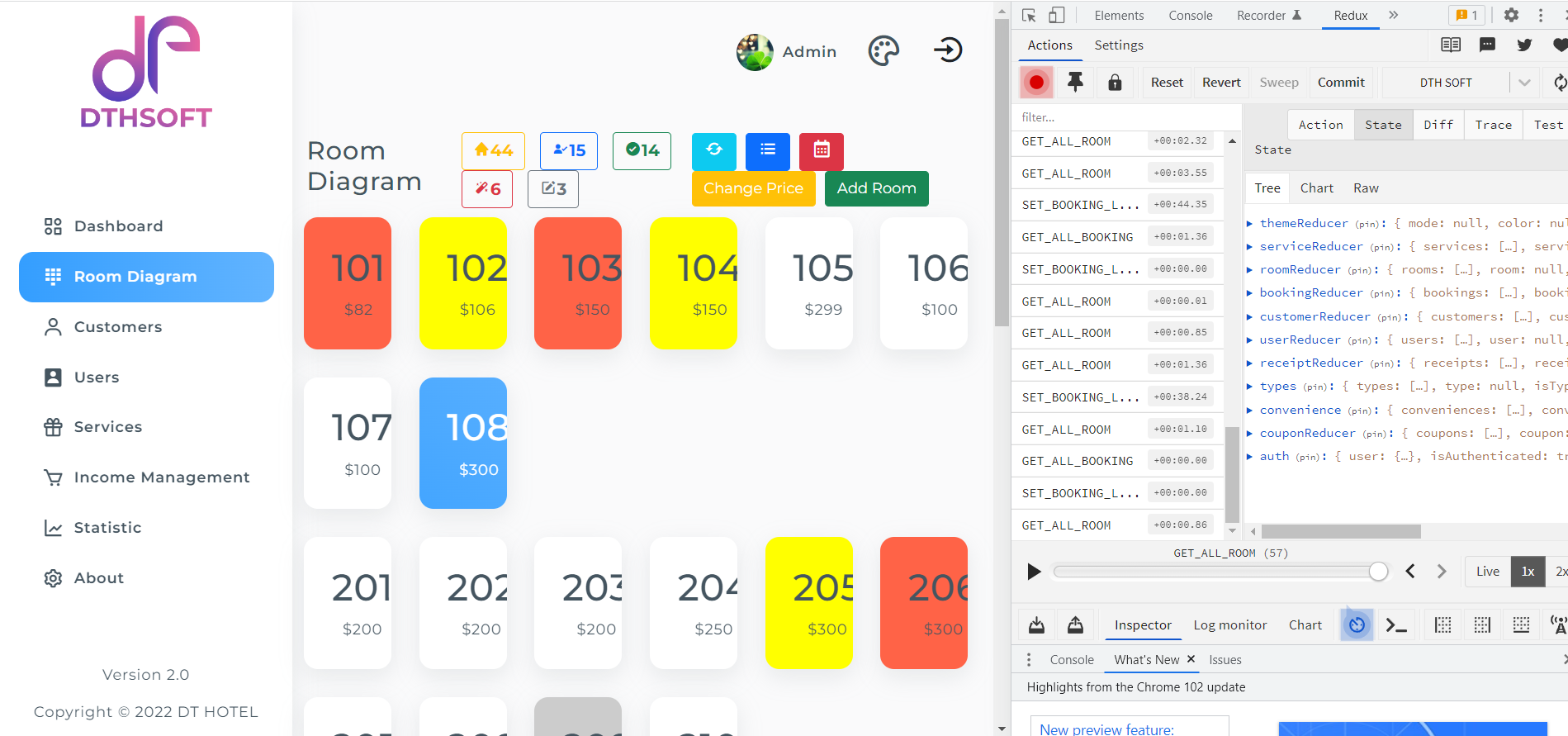
React-Bootstrap cung cấp sẵn một bộ các React component với look-and-feel của Twitter Bootstrap component từ đó giúp việc tạo dựng UI cho React app dễ dàng, tiện lợi hơn. Việc css inline cũng trở nên đơn giản hơn, giúp rút ngắn thời gian thiết kế cũng như xây dựng giao diện người dùng.



Hình 4.17: Thành phần Button của react-bootstrap

* **Công cụ Redux DevTools**

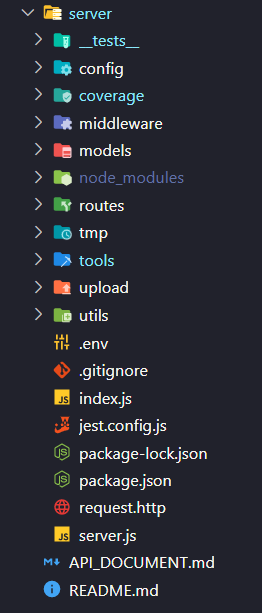
Redux DevTools là một Chrome DevTools extension, một tool mạnh mẽ được dùng để theo dõi flow của redux, theo dõi quá trình cập nhật state, quá trình dispatch action trong container hiện tại.



Hình 4.18: Công cụ redux devtools

## Back-end (NodeJS, ExpressJS)

### Cấu trúc thư mục



Hình 4.19: Cấu trúc thư mục back-end

Phần back-end ứng dụng công nghệ NodeJS, sử dụng ExpressJS framework để xây dựng các Restful API theo nhóm chức năng.

Bảng 4.3: Chi tiết cấu trúc thư mục back-end

|  |  |  |
| --- | --- | --- |
| **Thư mục / File** | **Cấu trúc thư mục** | **Mô tả các file** |
| configs |  | Chứa các file liên quan đến config hệ thống.  - File database.js: Config kết nối nối với database mongoDB.  - File key.js: Config các cấu hình cơ bản của web: name, api |
| middleware |  | Chức các file đóng vai trò là các middleware của back-end:  -File authentication.js: kiểm tra quyền người dùng để thực hiện các tác vụ.  - File authorization.js: kiểm tra token xác thực người dùng.  - File uploadImage.js: Kiểm tra sự hợp lệ của file ảnh được tải lên. |
| models |  | Chứa các mô hình lược đồ dữ liệu của đồ án bao gồm: booking, customer, receipt, room, service,user |
| node\_modules | … | Nơi lưu trữ các module đã được tải xuống. |
| routes |  | - Chứa các file định nghĩa bộ khung cơ sở dữ liệu, thực hiện các api các chức năng của ứng dụng.  - File index.js: Nơi tổng hợp tất cả các api |
| tools |  | Nơi chứa các file bao gồm các hàm đóng vai trò hỗ trợ, tính toán và thẩm định dữ liệu đầu vào, đầu ra. |
| utils |  | Nơi chứa các file xử lý các thao tác bổ sung cho các chức năng của ứng dụng. |
| \_\_test\_\_ |  | Nơi chứa những file unit test để kiểm thử các hàm, api chính của chương trình. |
| .env |  | Biến môi trường chứa các thông tin bảo mật ứng dụng và các thông tin dùng chung. |
| .gitignore |  | Nơi liệt kê những file mà mình không mong muốn cho vào git. |
| index.js |  | Là file quan trọng nhất của ứng dụng, nơi khai báo, cấu hình nên ứng dụng. |
| package-lock.json |  | Nơi lưu lại các phiên bản của thư viện của ứng dụng. |
| package.json |  | Chứa thông tin các dependencies cần cài đặt trong ứng dụng. |

### Các thư viện sử dụng

Bảng 4.4: Các thư viện sử dụng phía back-end

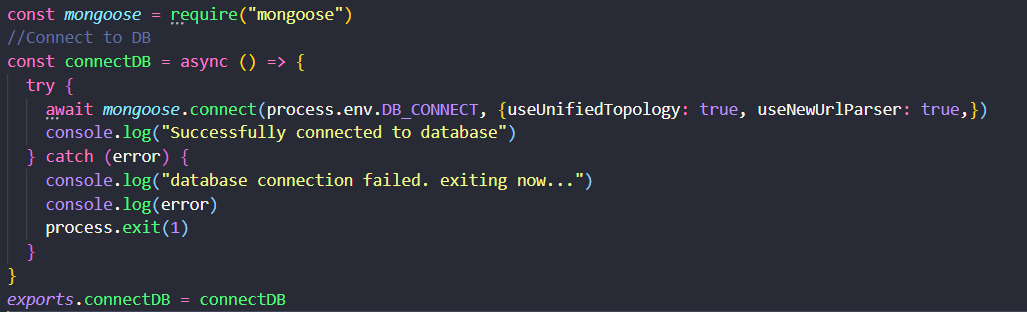
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Phiên bản** | **Licence** | **Mục đích** | **Áp dụng trong**  **dự án** |
| **dependencies** | | | | | |
| 1 | @hapi/joi | 17.1.1 | ISC | Định nghĩa dữ liệu đầu vào của các bảng cơ sở dữ liệu | File validation.js |
| 2 | bcryptjs | 2.4.3 | ISC | Công cụ hỗ trợ trong việc mã hóa | Mã hóa mật khẩu trong file auth.js, user.js |
| 3 | cloudinary | 1.27.1 | ISC | Một dịch vụ đám mây cung cấp giải pháp cho toàn bộ quy trình quản lý hình ảnh của ứng dụng web | File user.js |
| 4 | cors | 2.8.5 | ISC | Chia sẻ tài nguyên của các domain khác nhau cho nhau, hay đơn giản hơn là khi việc client gọi 1 api từ 1 nguồn khác có domain khác với trang hiện tại | File index.js |
| 5 | dotenv | 10.0.0 | ISC | Cấu hình environment variables để setting các giá trị cho từng môi trường. | Những file sử dụng biến môi trường: index.js, user.js, auth.js |
| 6 | express | 4.17.1 | ISC | Framework hỗ trợ, cung cấp các tính năng mạnh mẽ, thiết lập các lớp trung gian để trả về các HTTP request và định nghĩa router cho phép sử dụng với các hành động khác nhau dựa trên phương thức HTTP và URL | File index.js |
| 7 | express-fileupload | 1.2.1 | ISC | Xử lý dữ liệu multipart/form-data khi người dùng upload file. | File index.js |
| 8 | google-apis | 89.0.0 | ISC | Làm việc với các API Google Cloud Platform | File mailer.js |
| 9 | joi-objectid | 4.0.2 | ISC | Chuyển đổi kiểu dữ liệu ObjectId của mongoDB. | File validation.js |
| 10 | jsonwebtoken | 8.5.1 | ISC | Mã hóa dữ liệu, trao đổi thông tin và xác thực người dùng | Các file yêu cầu xác thực: auth.js,  authorization.js,  user.js |
| 11 | lodash | 4.17.21 | ISC | Làm việc với kiểu array object | Các file làm việc với array object: user.js, receipt.js,  validation.js, roomTool.js, serviceTool.js |
| 12 | moment | 2.29.1 | ISC | Xử lý, định dạng ngày tháng | Các file xử lý ngày: validation.js, roomTool.js |
| 13 | mongoose | 6.0.7 | ISC | Định nghĩa các schema làm việc với mongoDB | Các file trong folder models |
| 14 | nodemailer | 6.7.0 | ISC | Xử lý gửi email trong ứng dụng nodejs | File mailer.js |
| **devDependencies** | | | | | |
| 15 | nodemon | 2.0.14 | ISC | Tự động cập nhật và reload trang ứng dụng nodejs | File package.json |

### Đặc tả một số thư viện chính

#### Mongoose

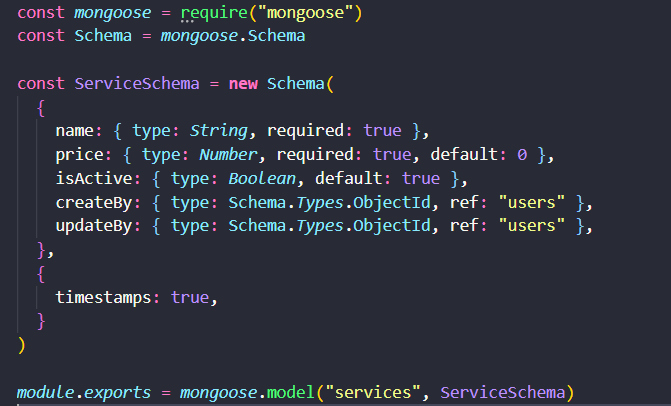
Là một thư viện lập trình hướng đối tượng Object Document Mapper (ODM). Thư viện cho phép định nghĩa các đối tượng (object) với một schema nhất định, được ánh tới một MongoDB document. Bằng cách tạo một model dựa trên một schema, mongoose chứa nhiều hàm cho phép xác thực tính hợp lệ, truy vấn, lưu, dữ liệu.

Đầu tiên, thư viện cho phép người sử dụng kết nối với một database MongoDB. Trong đồ án, kết nối database được sử dụng trong file database.js của thư mục config:



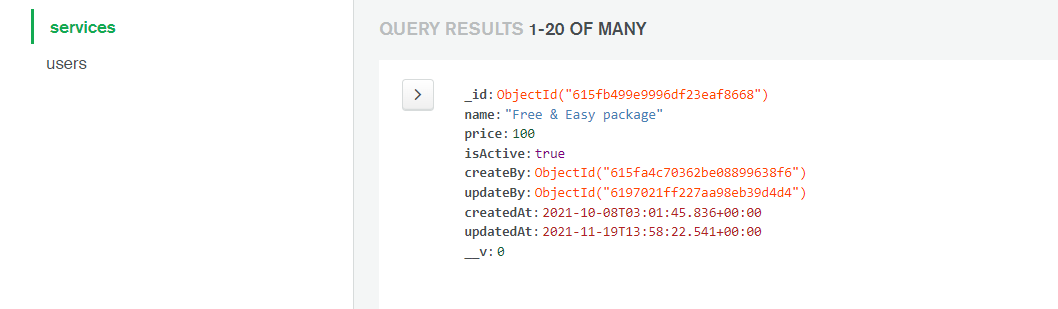
Hình 4.20: Cấu hình kết nối database

Một trong những công dụng chính của mongoose đó là định nghĩa schema linh hoạt để ánh xạ đến một MongoDB document. Ở đây, ví dụ schema services của ứng dụng:



Hình 4.21: Schema services

Một document “services” trong mongoDB được tạo ra với các thuộc tính được định nghĩa từ schema. Qua việc sử dụng phương thức save(), các object mới sẽ được lưu vào database của MongoDB.

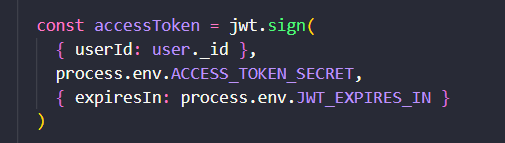


Hình 4.22: Dữ liệu được ánh xạ từ ứng dụng vào mongoDB

#### Jsonwebtoken

Thư viện chuẩn định nghĩa thông tin dưới dạng Json, sử dụng để mã hóa thông tin giúp xác thực người dùng. Token có 3 phần: header, payload, signature được ngăn cách bằng dấu chấm.

Trong ứng dụng, thư viện được sử dụng để trả về accessToken để xác thực người dùng truy cập vào ứng dụng:



Hình 4.23: Tạo accessToken với JWT

Thư viện cho phép mã hóa phần thông tin là userId và secret (khóa bí mật) bằng thuật toán HS256, cho phép người dùng được cấp quyền truy cập vào các url, service, và resource của hệ thống.

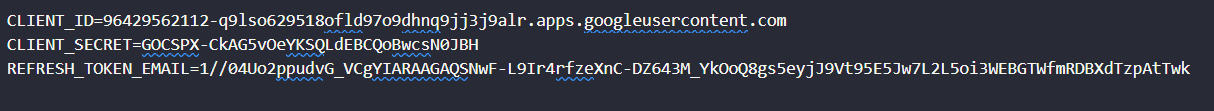
#### Nodemailer + Google-apis

Để thực hiện chức năng gửi email trong ứng dụng NodeJS, đồ án xây dựng API gửi email trong NodeJS với nodemailer, gmail, OAuth2Client.

Để cấu hình thiết lập ID Client OAuth2 cho ứng dụng:

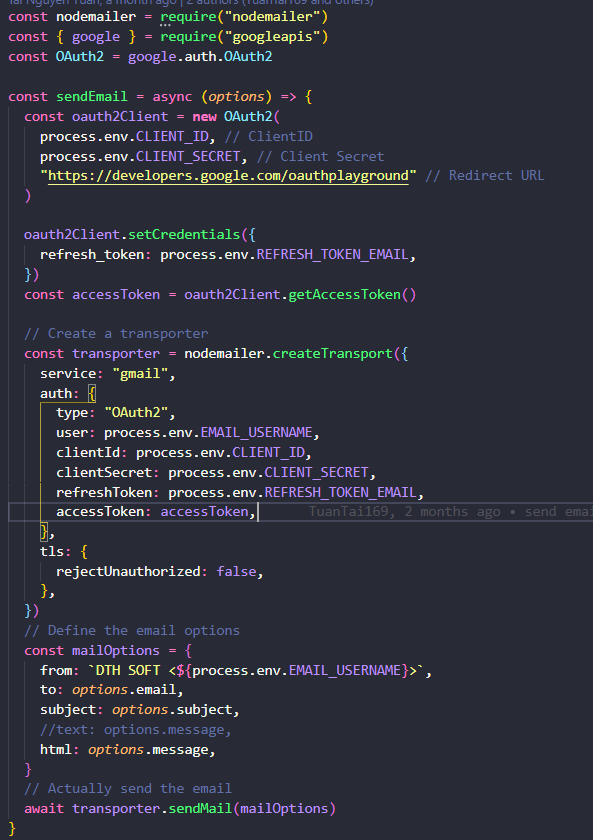
* Truy cập <https://console.developers.google.com/> để tạo một ứng dụng mới
* Tìm chọn APIs & Services 🡪 Credentials 🡪 Create credentials 🡪 OAuth client ID 🡪 Lưu ClientID và Client Secret
* Truy cập <https://developers.google.com/oauthplayground>
* Chọn vào Setting icon 🡪 Enable Use your own OAuth credentials 🡪 Sao chép ClientID và Client Secrets vào form
* Tại Select & Authorize APIs, gõ <https://mail.google.com/> 🡪 Authorize APIs 🡪 Đăng nhập với tài khoản email tạo ứng dụng ở trên.
* Click Exchange authorization code 🡪 Lưu Refresh token

Sau khi cấu hình xong, file .env sẽ có những dữ liệu như sau:



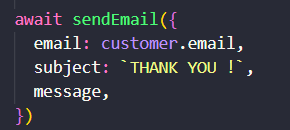
Hình 4.24: Thông tin cấu hình Google OAuth2 Client

Tiếp tục cấu hình nodemailer với Google OAuth2:



Hình 4.25: Cấu hình Google OAuth2 Client

Để gửi email, người dùng truyền vào các tham số gửi mail và kiểm tra kết quả:



Hình 4.26: Gửi email với Google OAuth2 Client

# PHÂN TÍCH DỮ LIỆU – TÍNH NĂNG ĐỀ ÁN

## 4.1 Mô hình hóa yêu cầu

### 4.1.1 Lược đồ Usecase

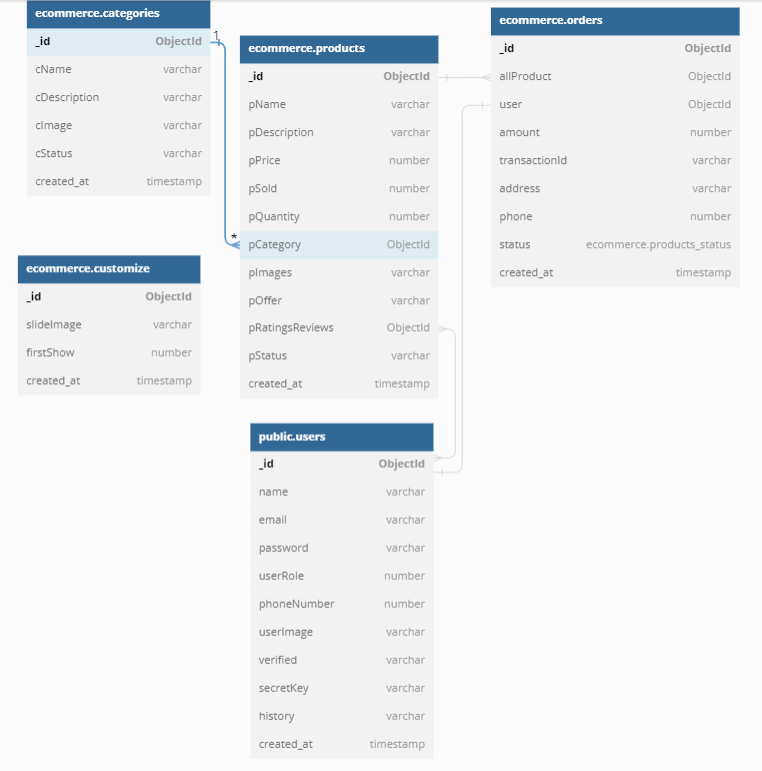
Bảng: Mô tả sơ đồ Usecase

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên chức năng** | **Mô tả** |
| DTH\_01 | Log in | Chức năng cho phép actor đăng nhập tài khoản vào website |

**Bảng: Usecase đăng nhập**

|  |  |
| --- | --- |
| **User Case ID** | DTH\_01 |
| **Name** | Log in |
| **Goal** | Đăng nhập thành công vào trang web |
| **Actors** | User, Manager, Admin |
| **Pre-conditions** | * Tài khoản người dùng đã được tạo sẵn. * Tài khoản người dùng đã được phân quyền. |
| **Post-conditions** | * Người dùng đăng nhập thành công và đến trang dashboard. |
| **Main Flow** | 1. Người dùng chọn chức năng đăng nhập từ website. 2. Hệ thống chuyển đến trang đăng nhập, hiển thị input để nhập email, mật khẩu. 3. Người dùng click vào nút “LOGIN”. 4. Hệ thống sẽ kiểm tra email, mật khẩu. (E1) |
| **Alternate Flow** |  |
| **Exception** | (E1) Hệ thống kiểm tra thất bại và thông báo sai email hoặc mật khẩu. Người dùng quay về bước số 2 để thực hiện đăng nhập lại. |
| **Open Issues** |  |
| **Relationship** |  |

**Link lược đồ quan hề dữ liệu:** [**https://dbdiagram.io/d/6367bef9c9abfc6111709862**](https://dbdiagram.io/d/6367bef9c9abfc6111709862)

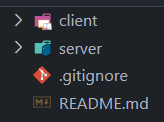
****

# CÁCH SỬ DỤNG – CÀI ĐẶT

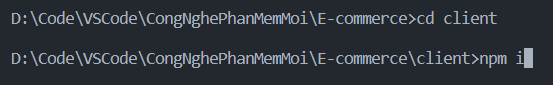
**5.1 Cài đặt**

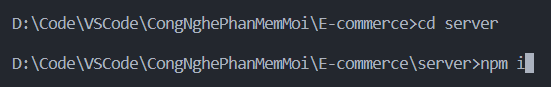
Link clone: <https://github.com/dipien2408/E-commerce.git>

Sau khi clone về và mở folder E-commerce bằng Visual Studio sẽ được 4 file như sau



Mở Terminal và lần lượt truy cập vào 2 file là client và server để cài đặt npm package





Đứng từ server chạy câu lệnh sau để khởi động server

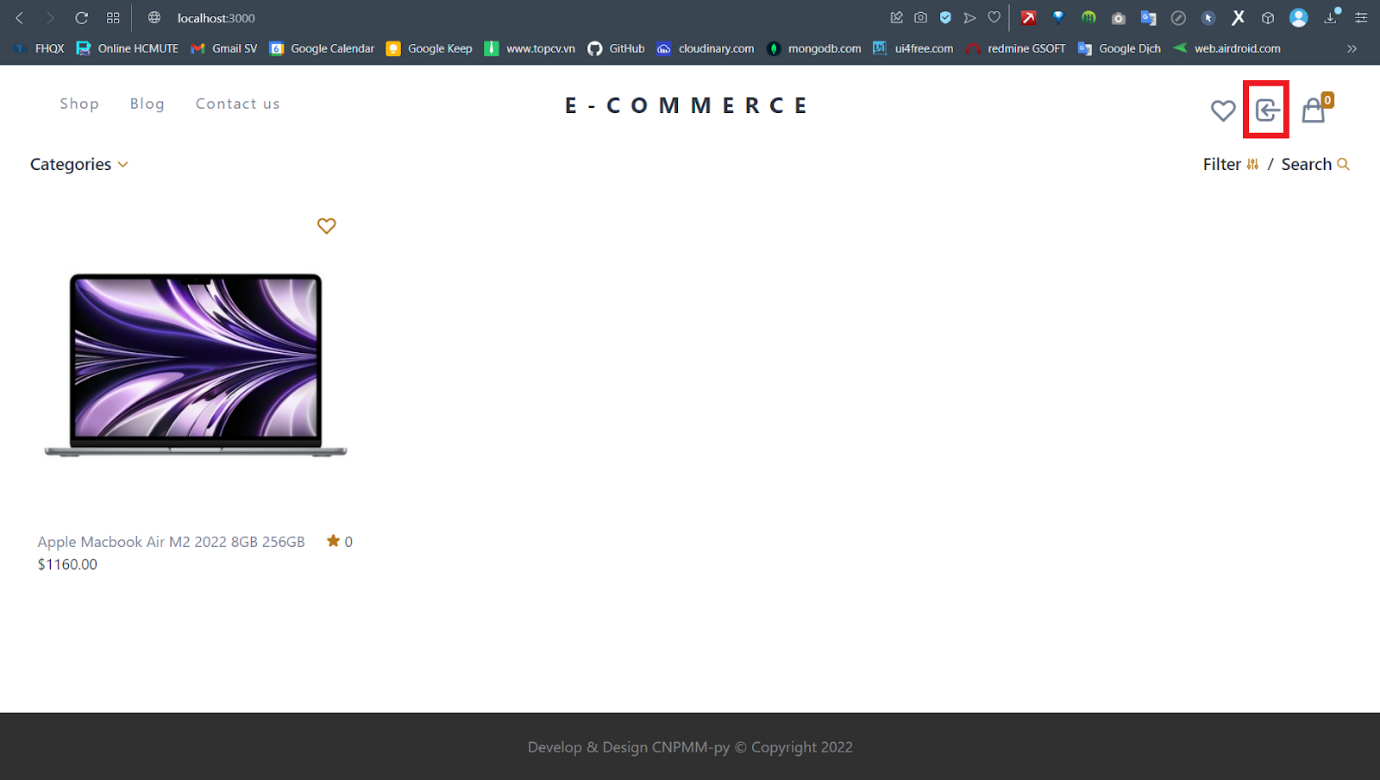


Đứng từ client chạy câu lệnh sau để khởi động client

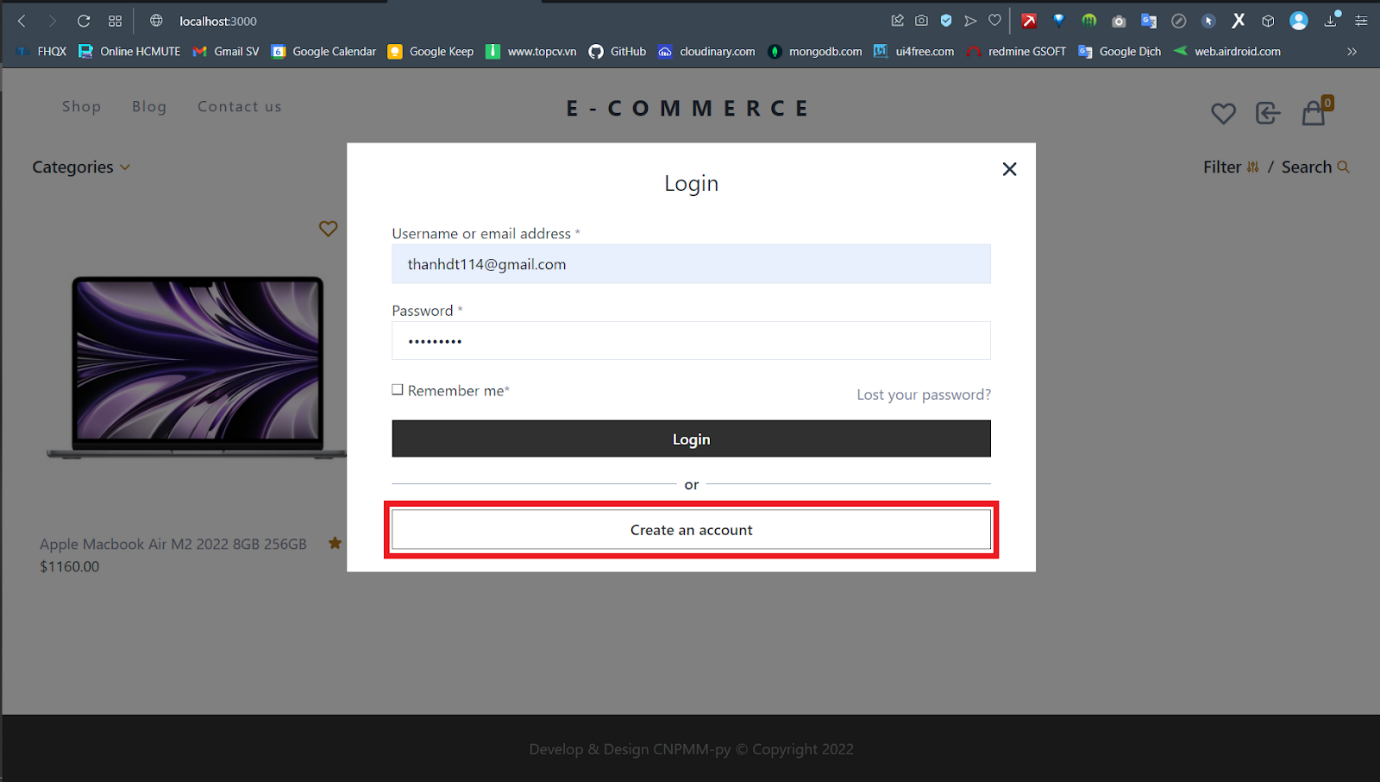


**5.2 Hướng dẫn sử dụng website**

* Giao diện khi lần đầu truy cập trang web

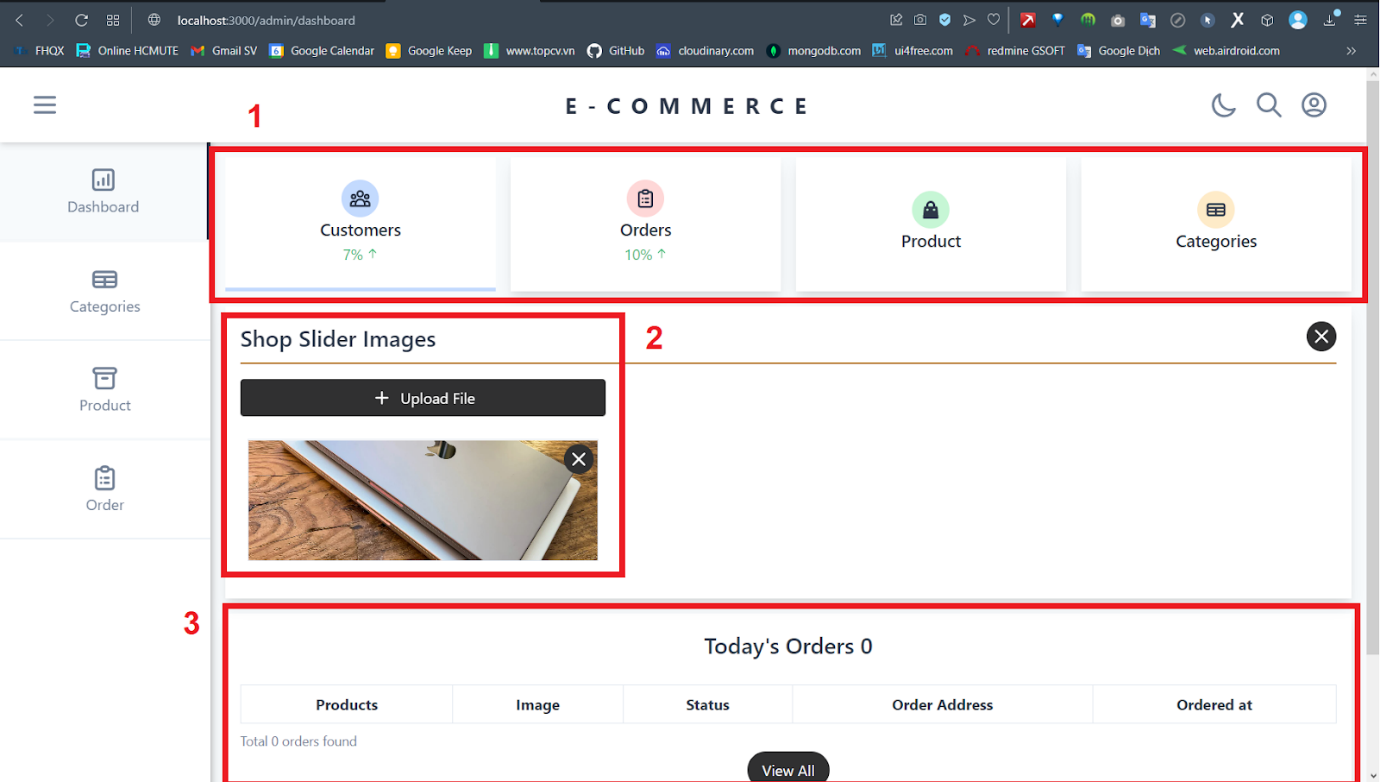


* Thực hiện đăng ký tài khoản để sử dụng các chức năng của trang web

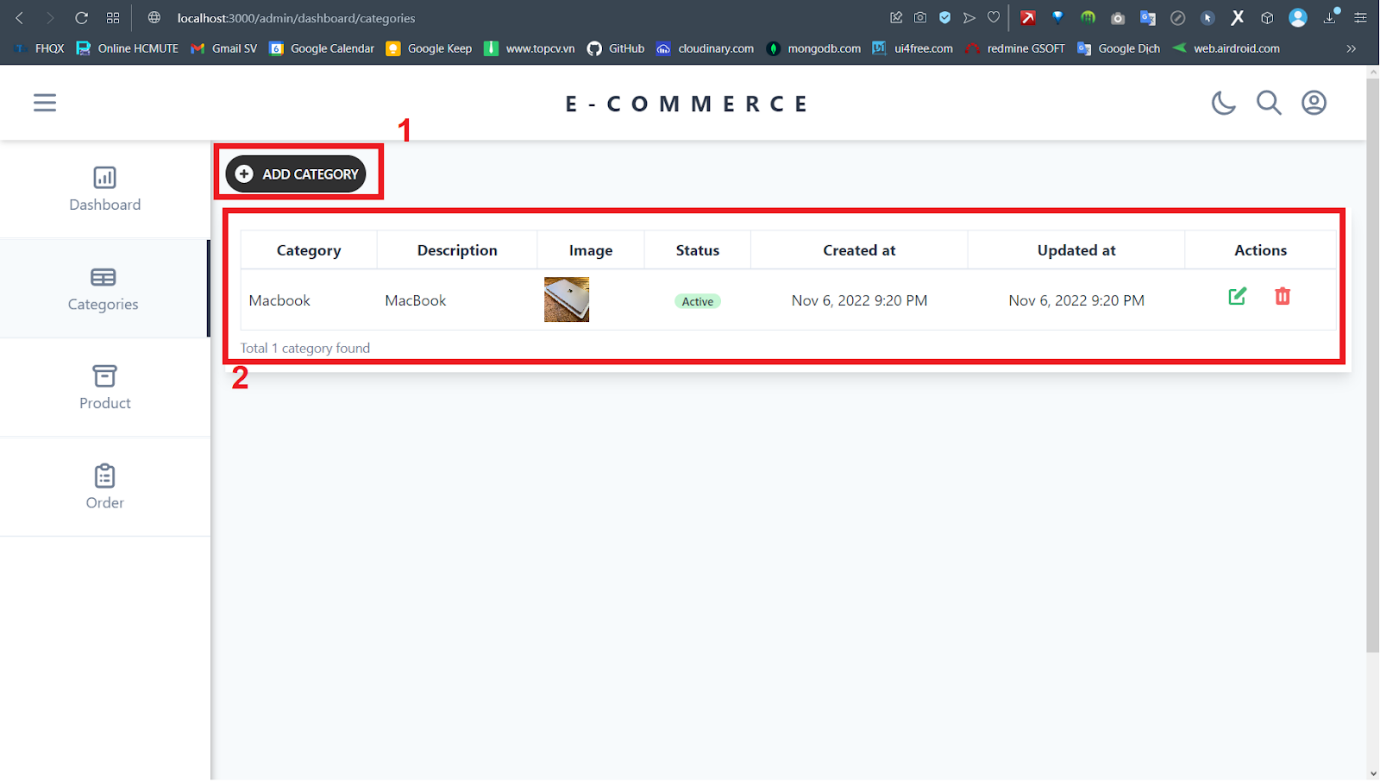


**5.2.1 Các chức năng chính đối với quyền admin**

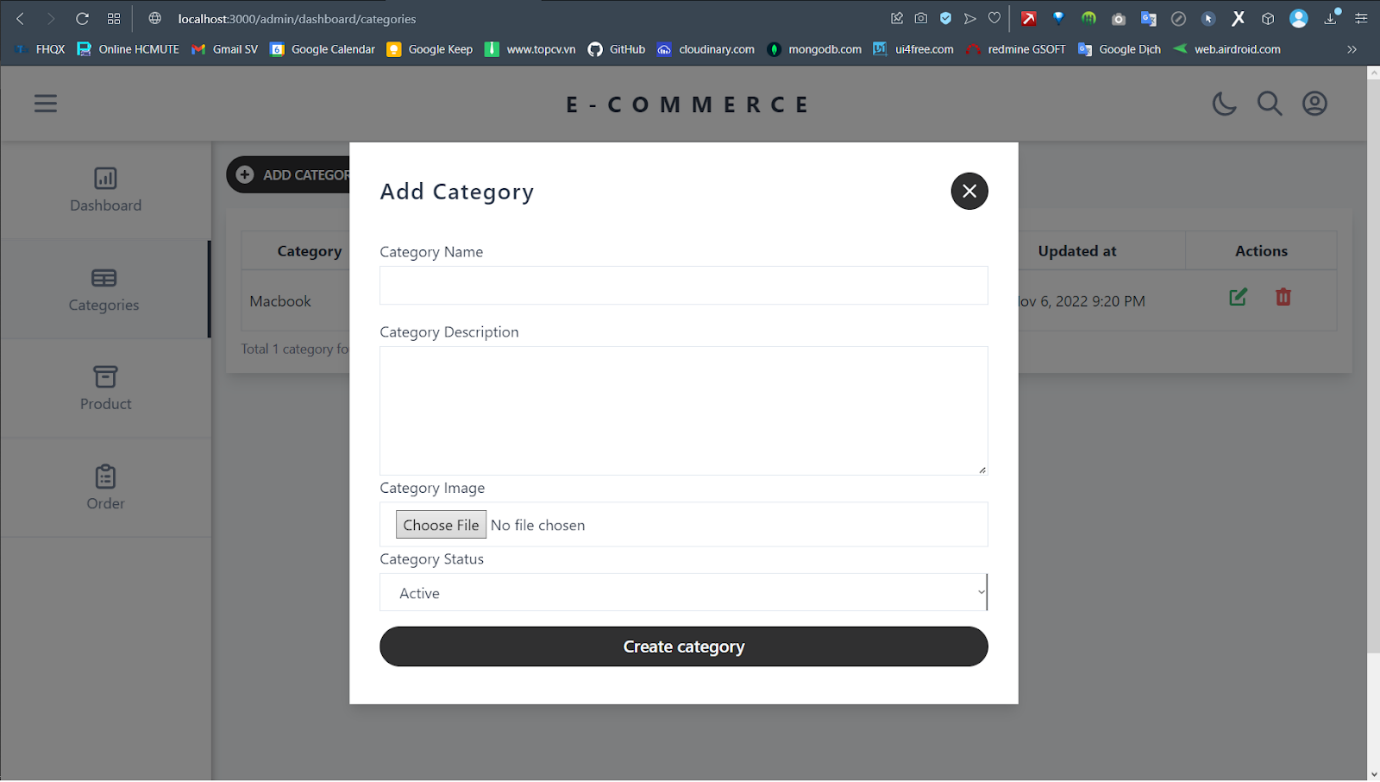
* Chức năng đầu tiên là quản lý Dashboard của website, ở trang chức năng này sẽ bao gồm 3 phần chính là:
* 1. Danh mục thống kê (Customers, Orders, Product, Categories)
* 2. Cài đặt hình ảnh cho trang chủ của website
* 3. Thống kê lượng đơn hàng trong ngày hôm nay



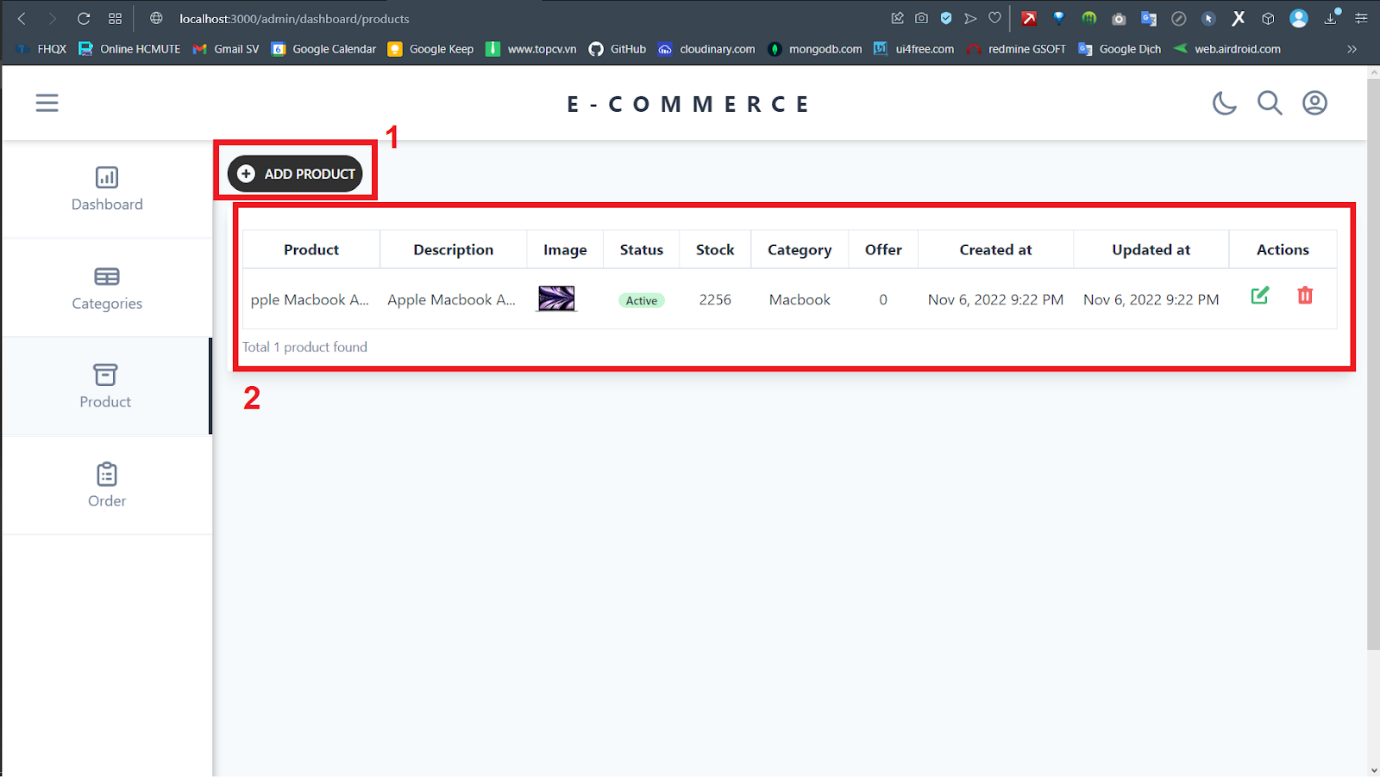
* Chức năng thứ 2 của admin là quản lý danh mục sản phẩm
* 1. Thêm các danh mục sản phẩm mới
* 2. Hiển thị danh sách các danh mục sản phẩm và cho phép chỉnh sửa hoặc xóa danh mục sản phẩm đó



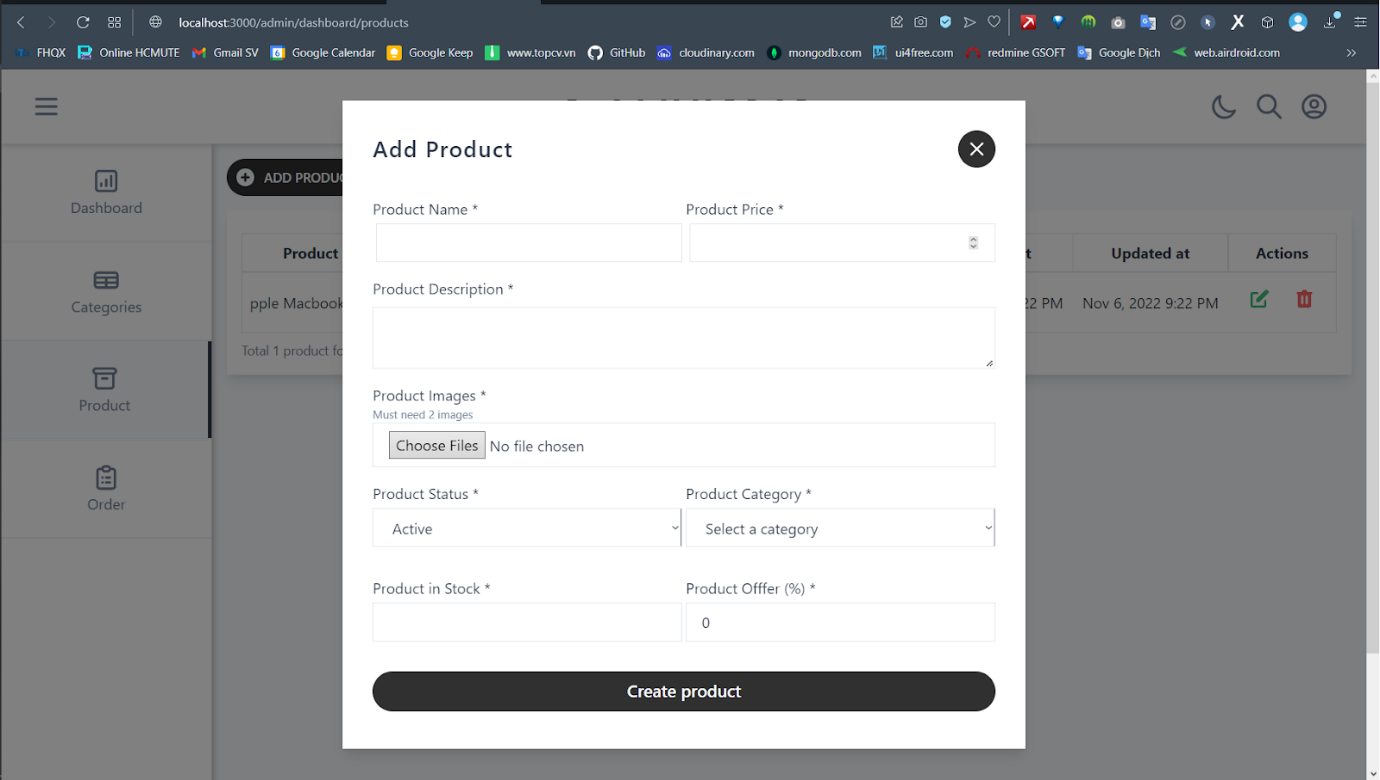
* Khi thêm mới danh mục sản phẩm có thể thêm các thông tin cơ bản và set trạng thái cho danh mục sản phẩm đó (tương tự như việc xóa bản mềm dữ liệu)



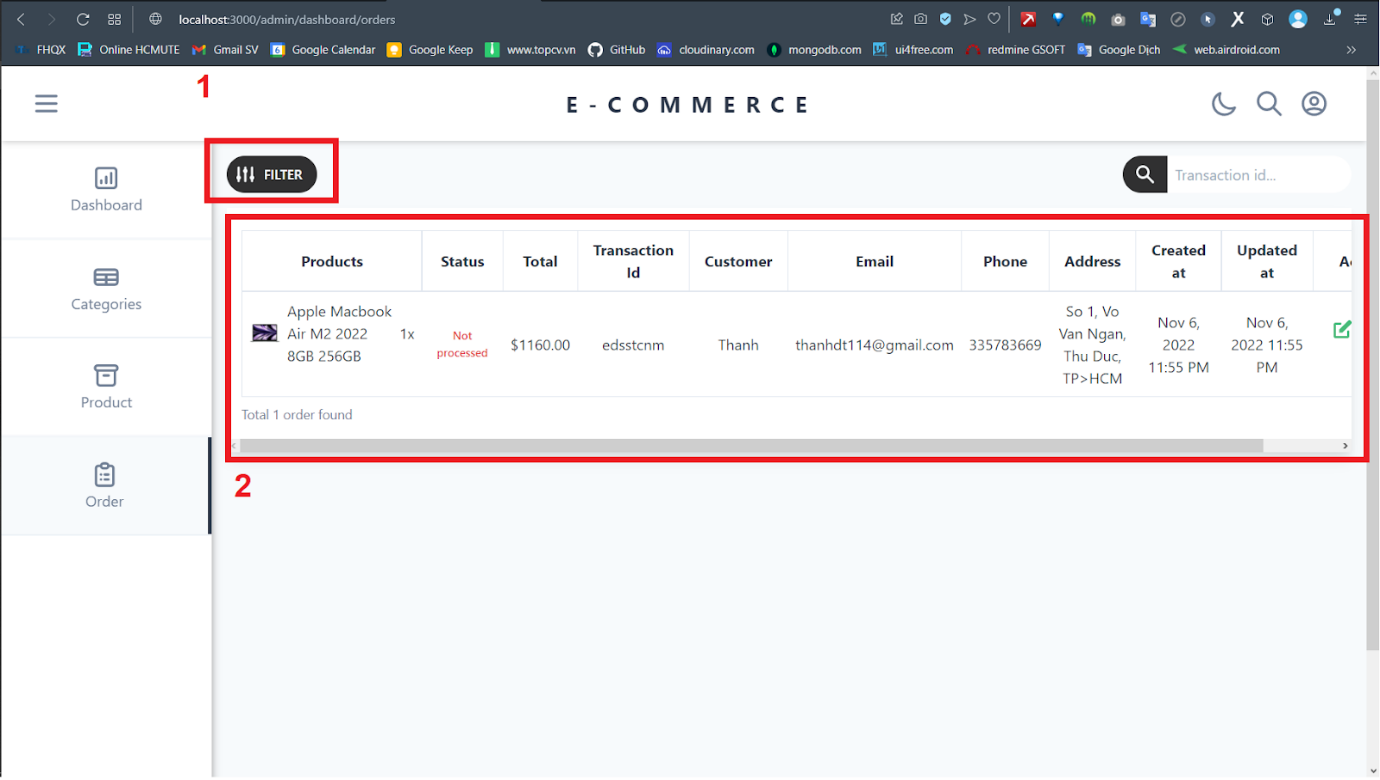
* Tương tự như quản lý danh mục sản phẩm thì quản lý sản phẩm cũng sẽ có 2 phần chính là:
* 1. Thêm mới sản phẩm
* 2. Chỉnh sửa hoặc xóa sản phẩm



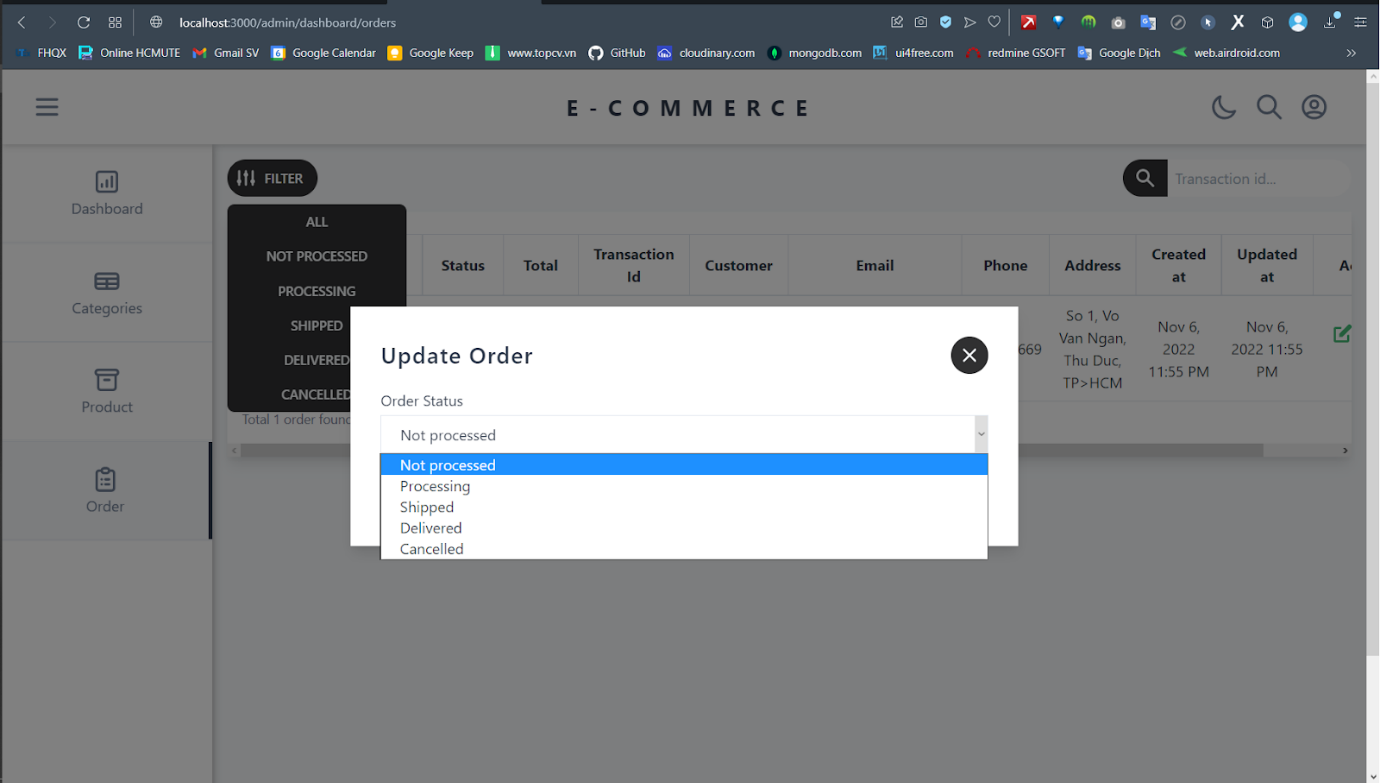
* Khi thêm mới sản phẩm sẽ cho nhập các thông tin cơ bản về sản phẩm và tải lên ít nhất 2 hình ảnh về sản phẩm đó (không được để trống bất kỳ 1 field nào khi thêm mới sản phẩm)



* Chức năng cuối cùng là quản lý đơn hàng:
* 1. Thực hiện lọc đơn hàng theo trạng thái
* 2. Danh sách đơn hàng và admin có quyền chỉnh sửa hoặc xóa đơn hàng

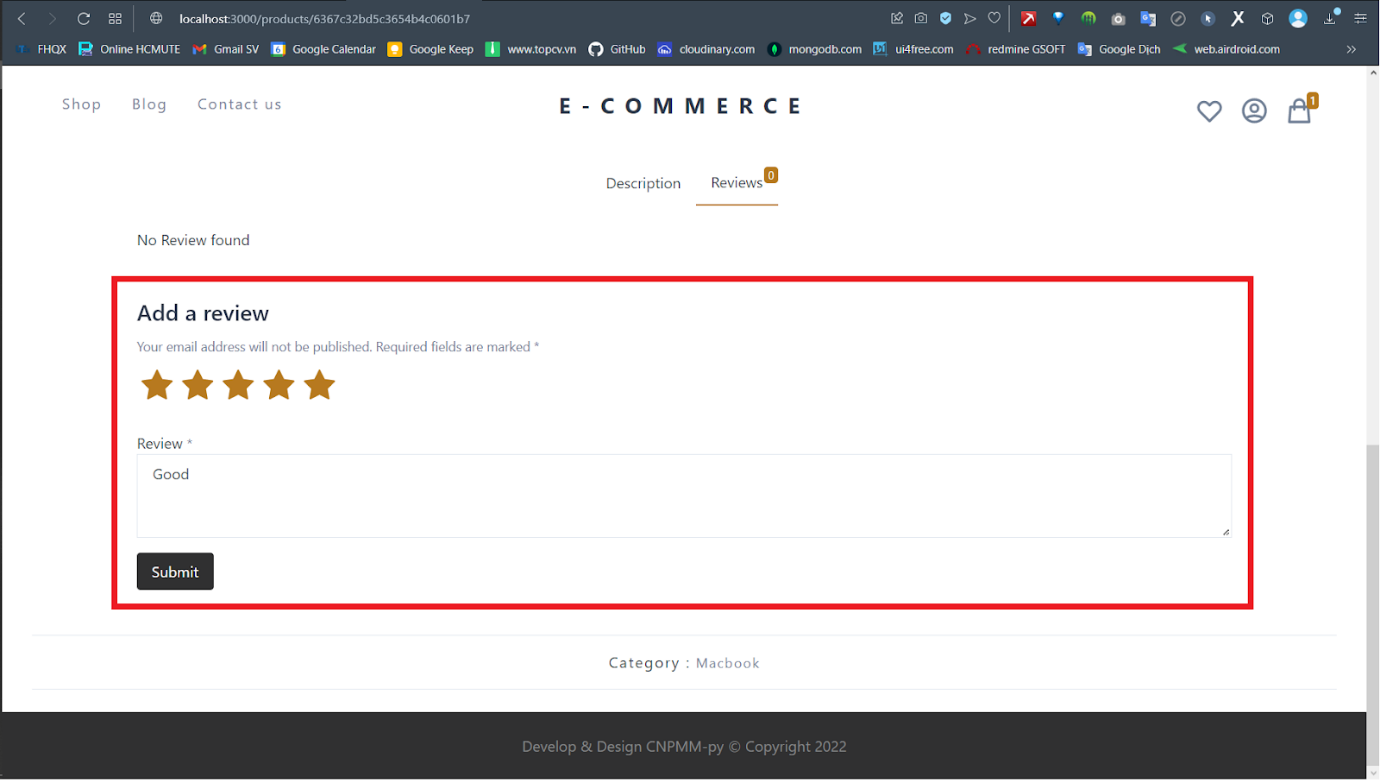


* Khi chỉnh sửa hoặc lọc đơn hàng thì chỉ theo trạng thái của đơn hàng

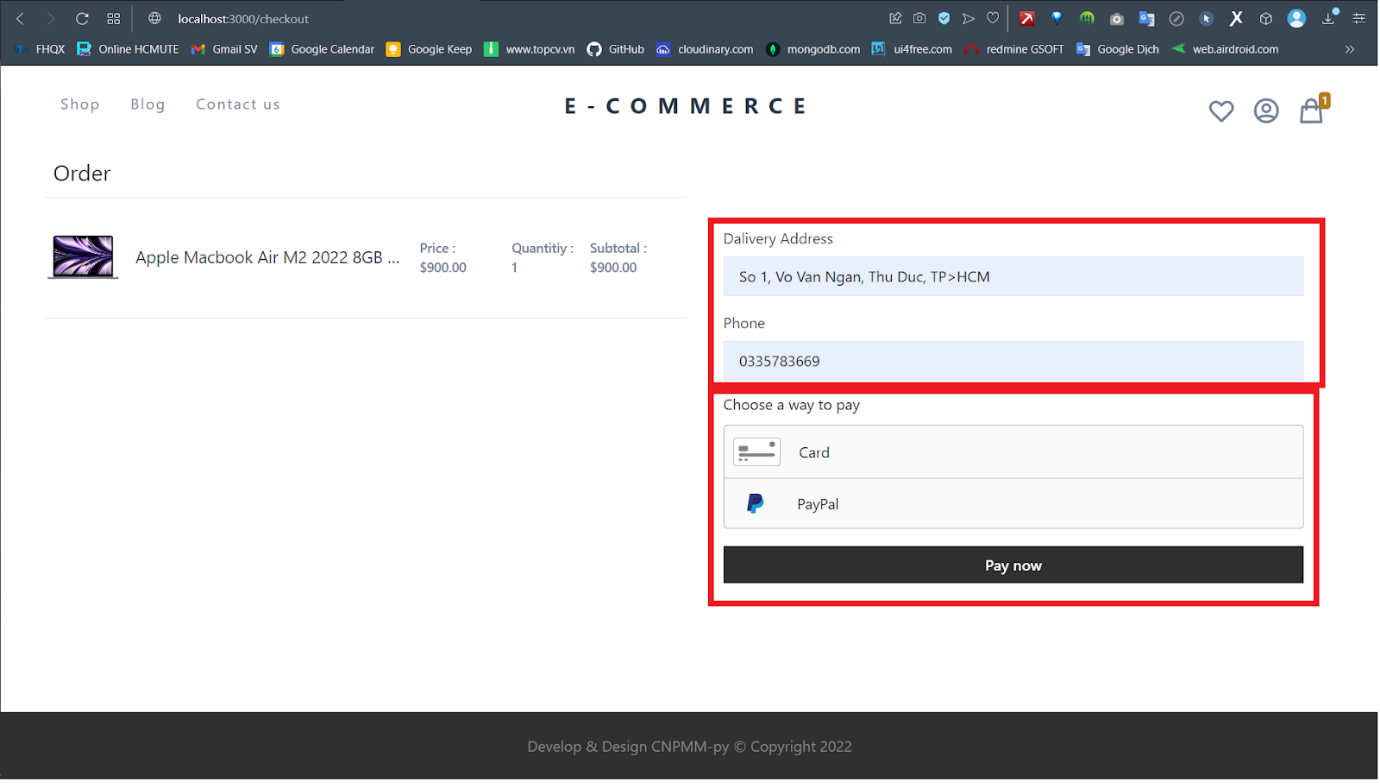


**5.2.2 Các chức năng chính đối với quyền User**

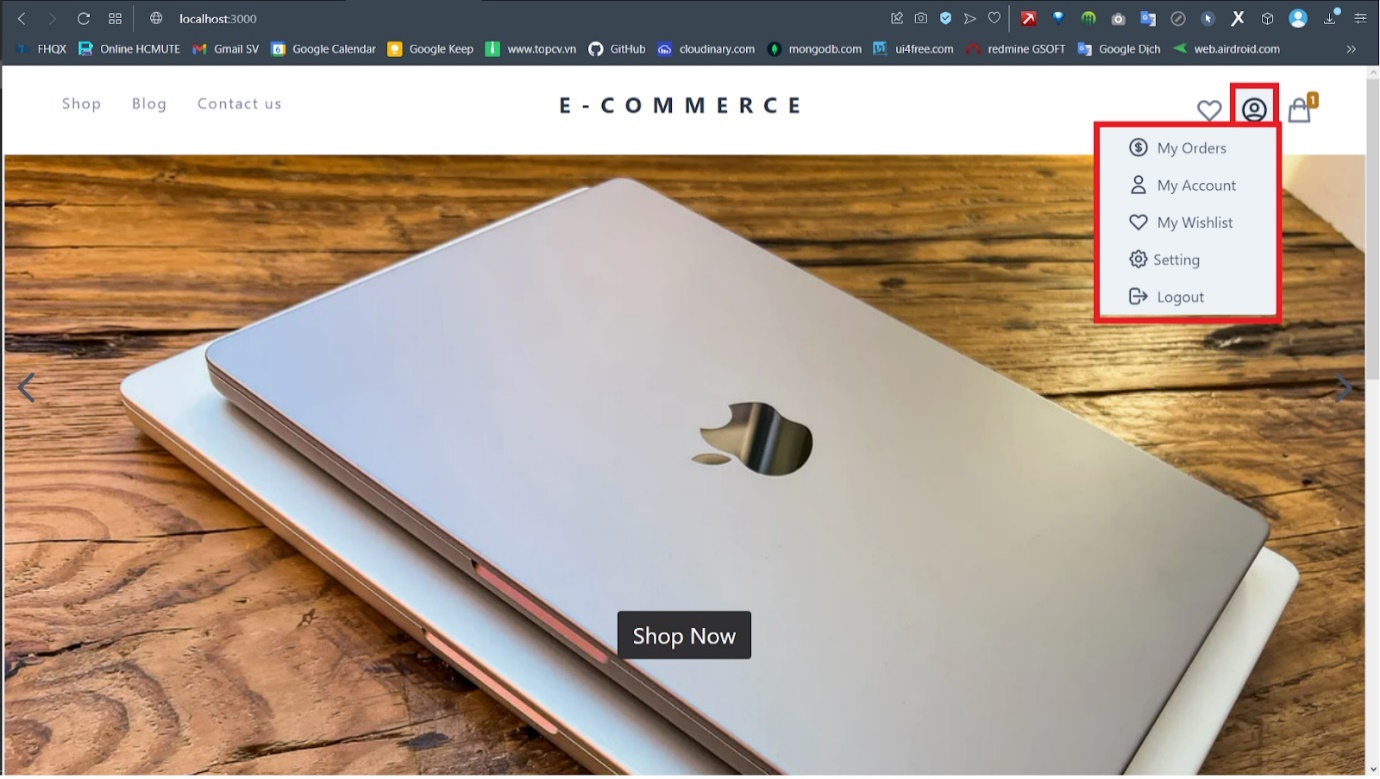
* Chức năng sử dụng trực tiếp khi đang xem sản phẩm là đánh giá (xem đánh giá) sản phẩm của người dùng là không thể thiếu để những người dùng khác có thể dựa vào những đánh giá này để đưa ra quyết định có nên mua hay không.



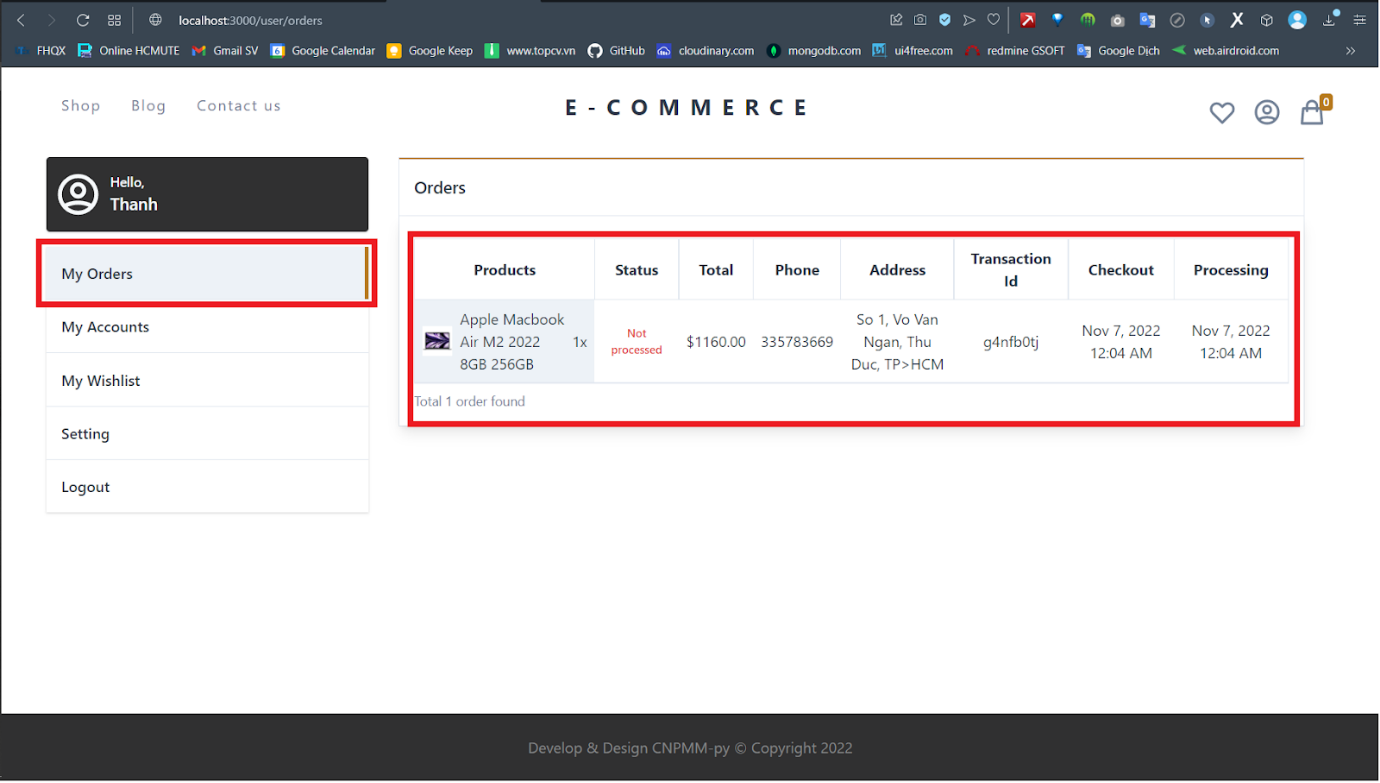
* Thực hiện đặt đơn hàng: khi tiến hành đặt hàng sẽ yêu cầu người dùng nhập các thông tin cơ bản và chọn phương thức thanh toán



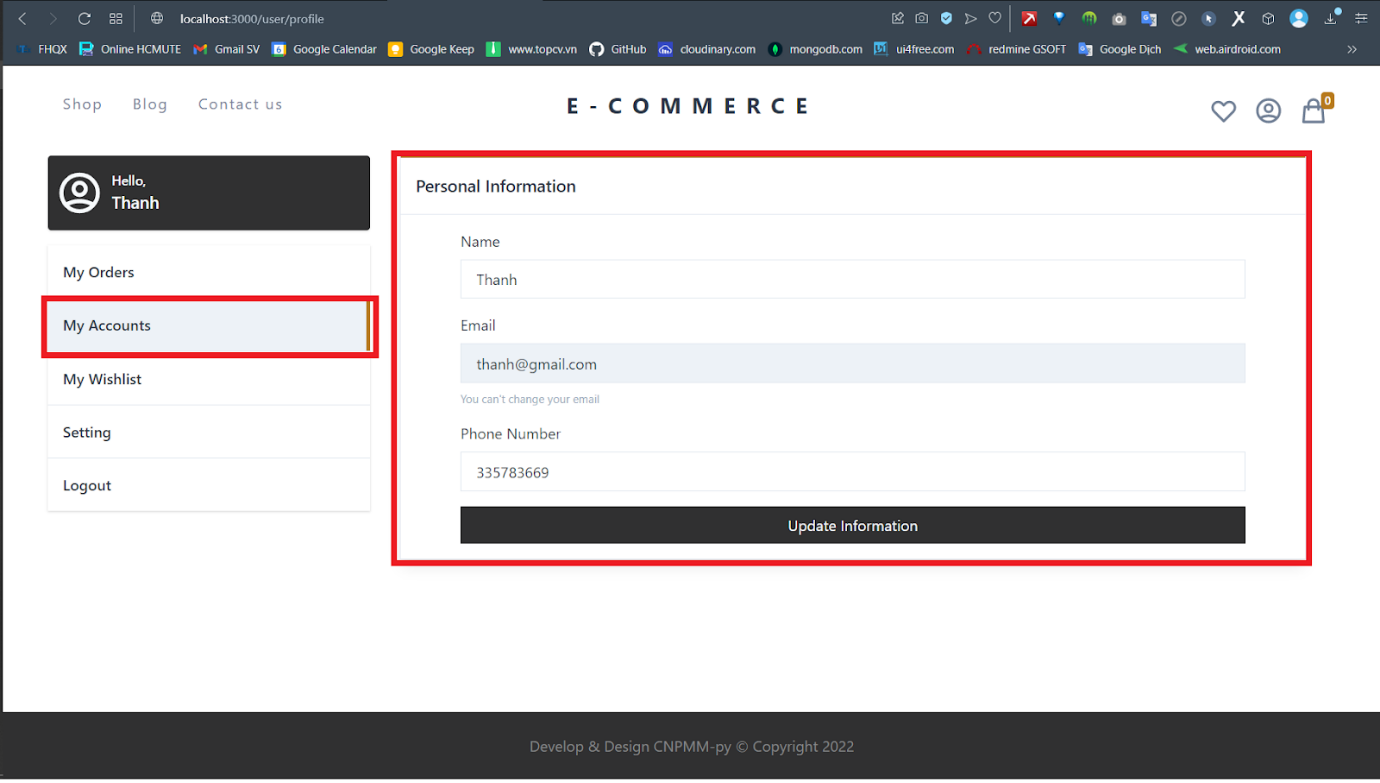
* Những chức năng tiếp theo sẽ cần truy cập vào icon user mới có thể truy cập được:



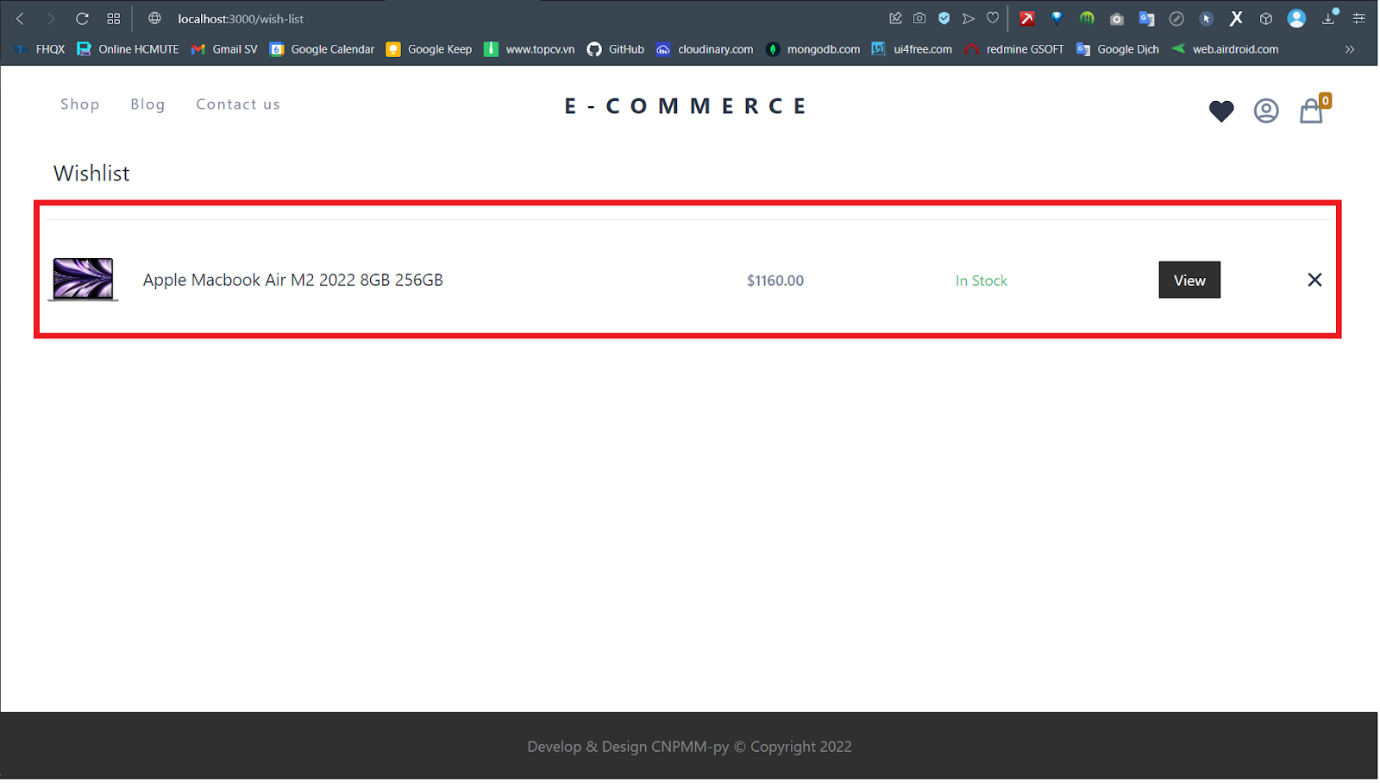
* Chức năng đầu tiên của user là quản lý đơn hàng mà mình đã đặt: hiển thị danh sách các đơn hàng đã đặt và có thể theo dõi trạng thái của đơn hàng.



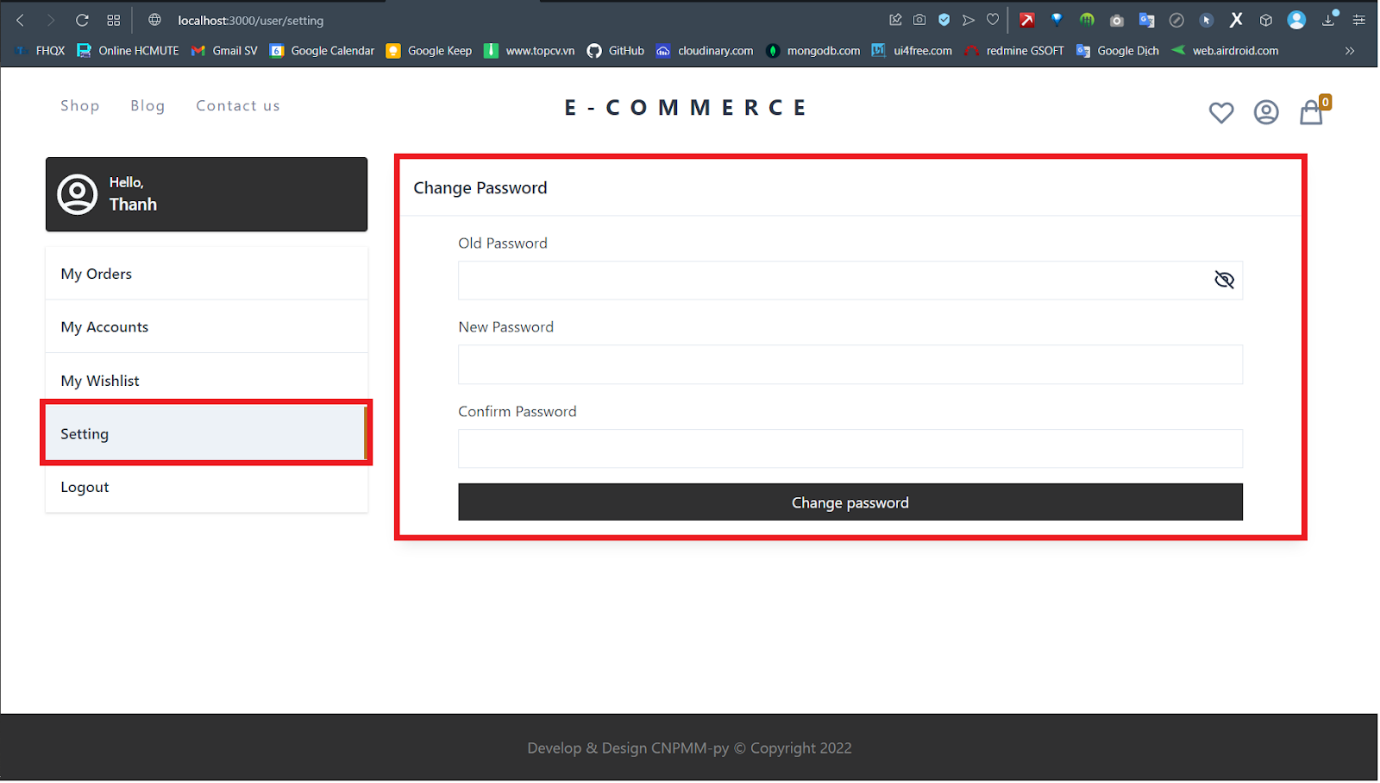
* Chức năng thứ 2 của user là quản lý tài khoản của mình: có thể cập nhật các thông tin cơ bản liên quan đến account



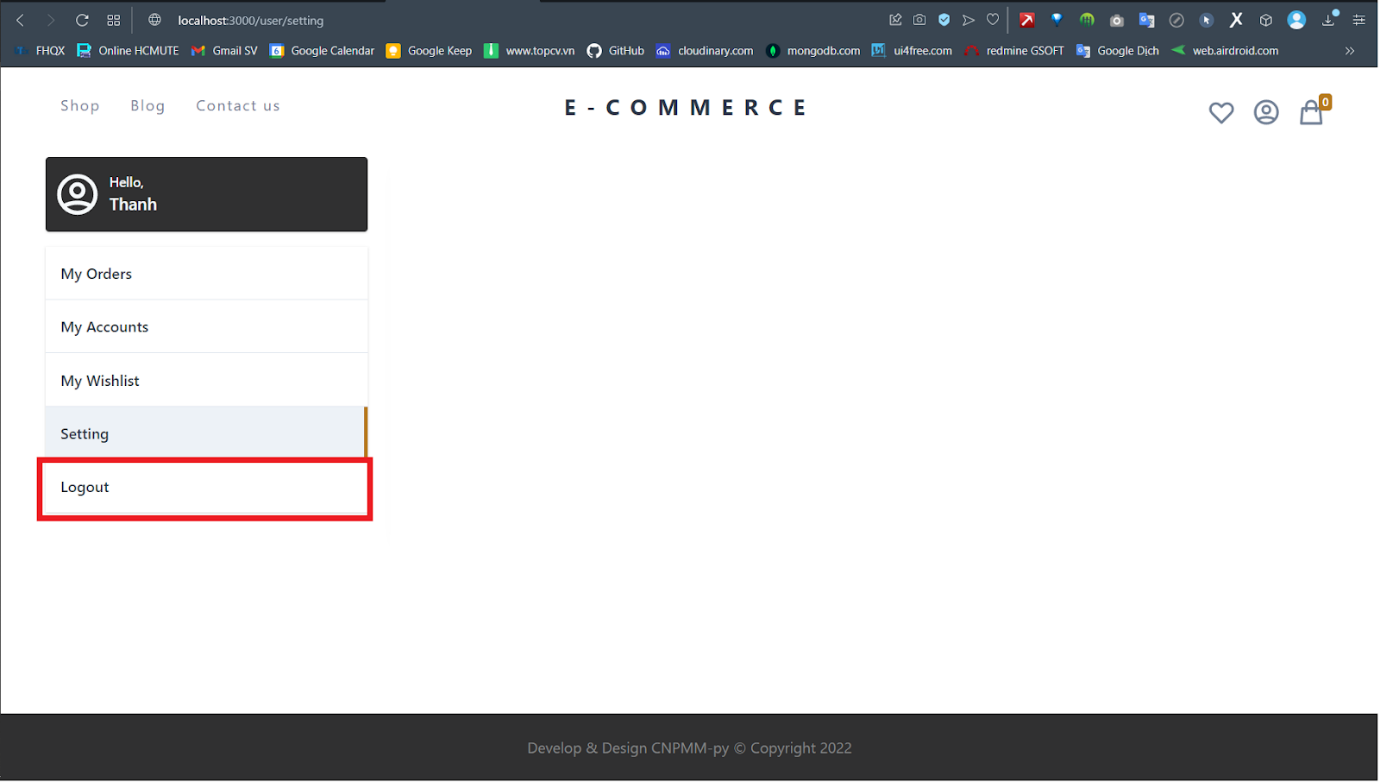
* Chức năng thứ 3 là quản lý danh sách những sản phẩm yêu thích: hiển thị danh sách sản phẩm yêu thích để người dùng dễ dàng truy cập



* Chức năng thứ 4 là đặt lại mật khẩu của người dùng: người dùng cung cấp mật khẩu cũ mới có thể thay đổi mật khẩu của mình

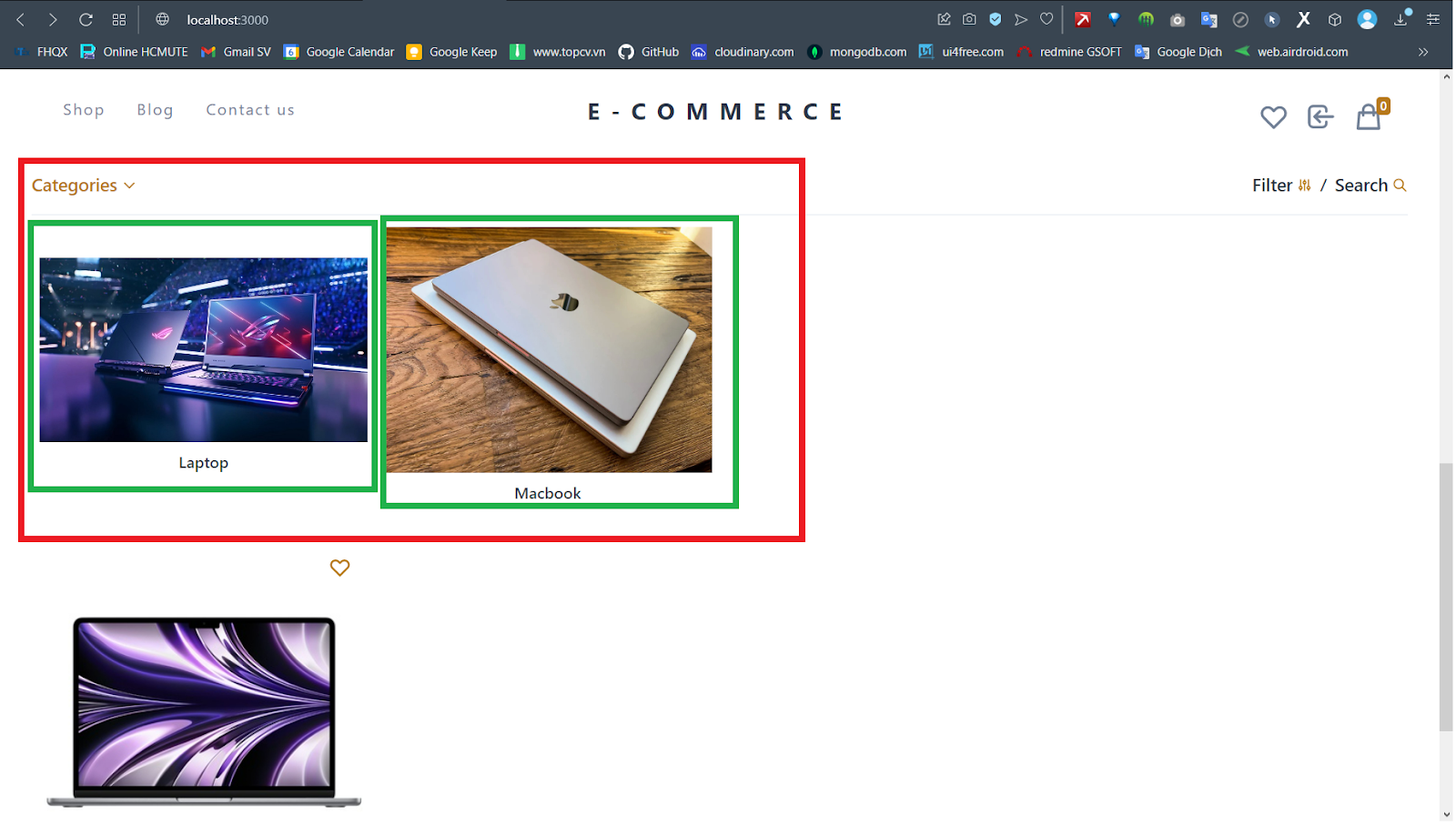


* Chức năng cuối cùng là đăng xuất tài khoản

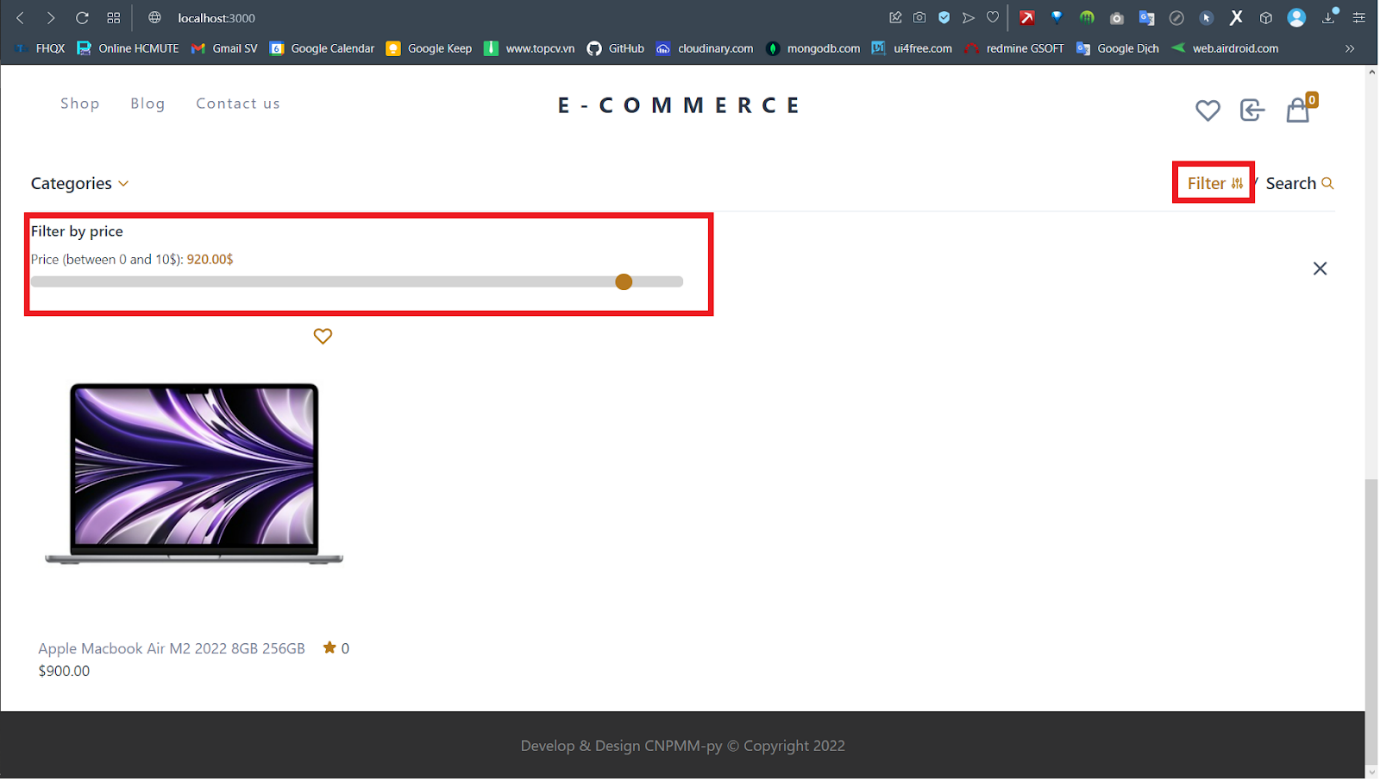


**5.2.3 Chức năng sử dụng trực tiếp khi truy cập website mà không cần đăng nhập tài khoản**

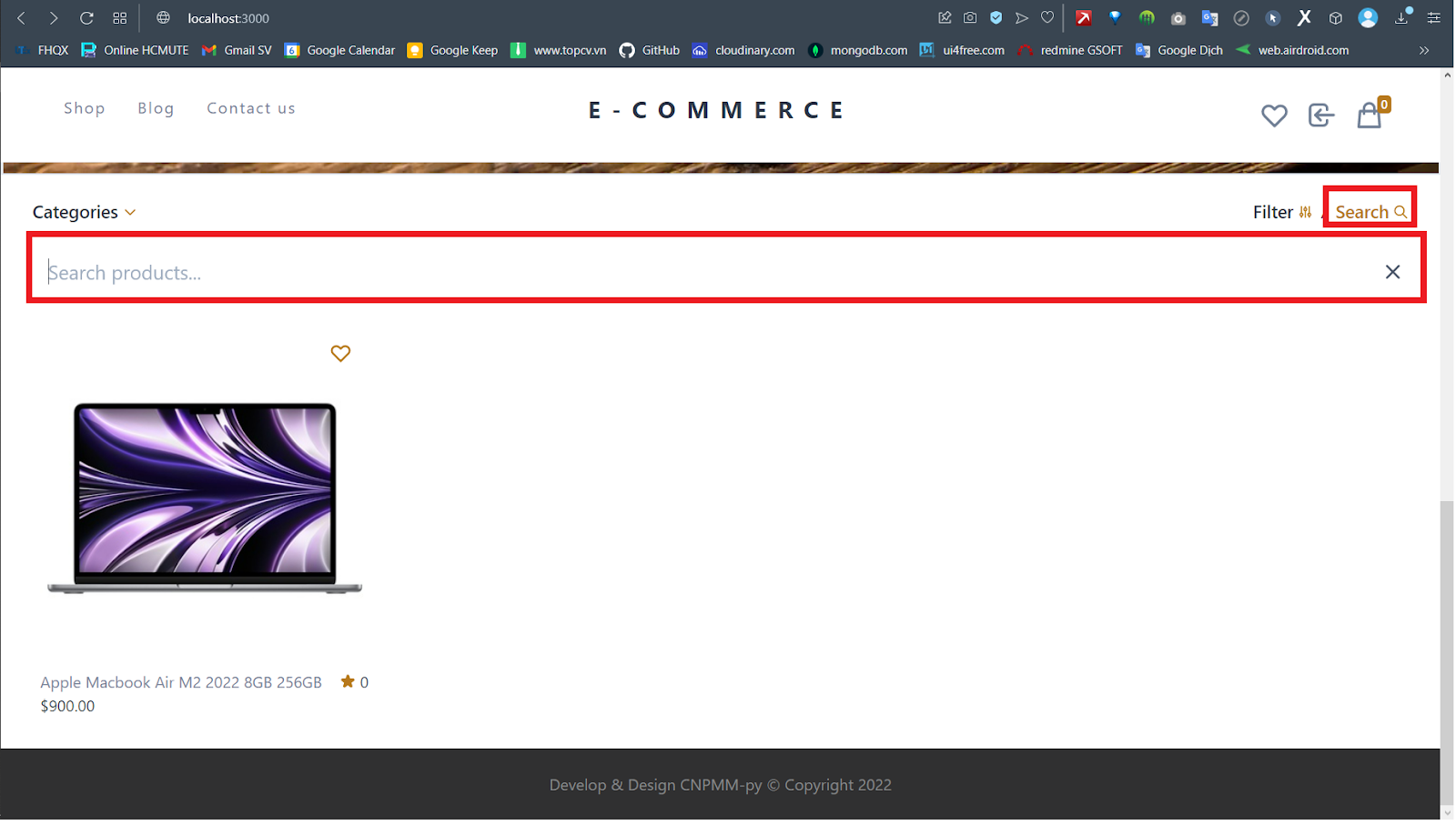
* Có thể tìm sản phẩm theo danh mục sản phẩm: chỉ cần chọn vào danh mục sản phẩm thì tất cả sản phẩm thuộc danh mục sản phẩm đó sẽ được hiển thị



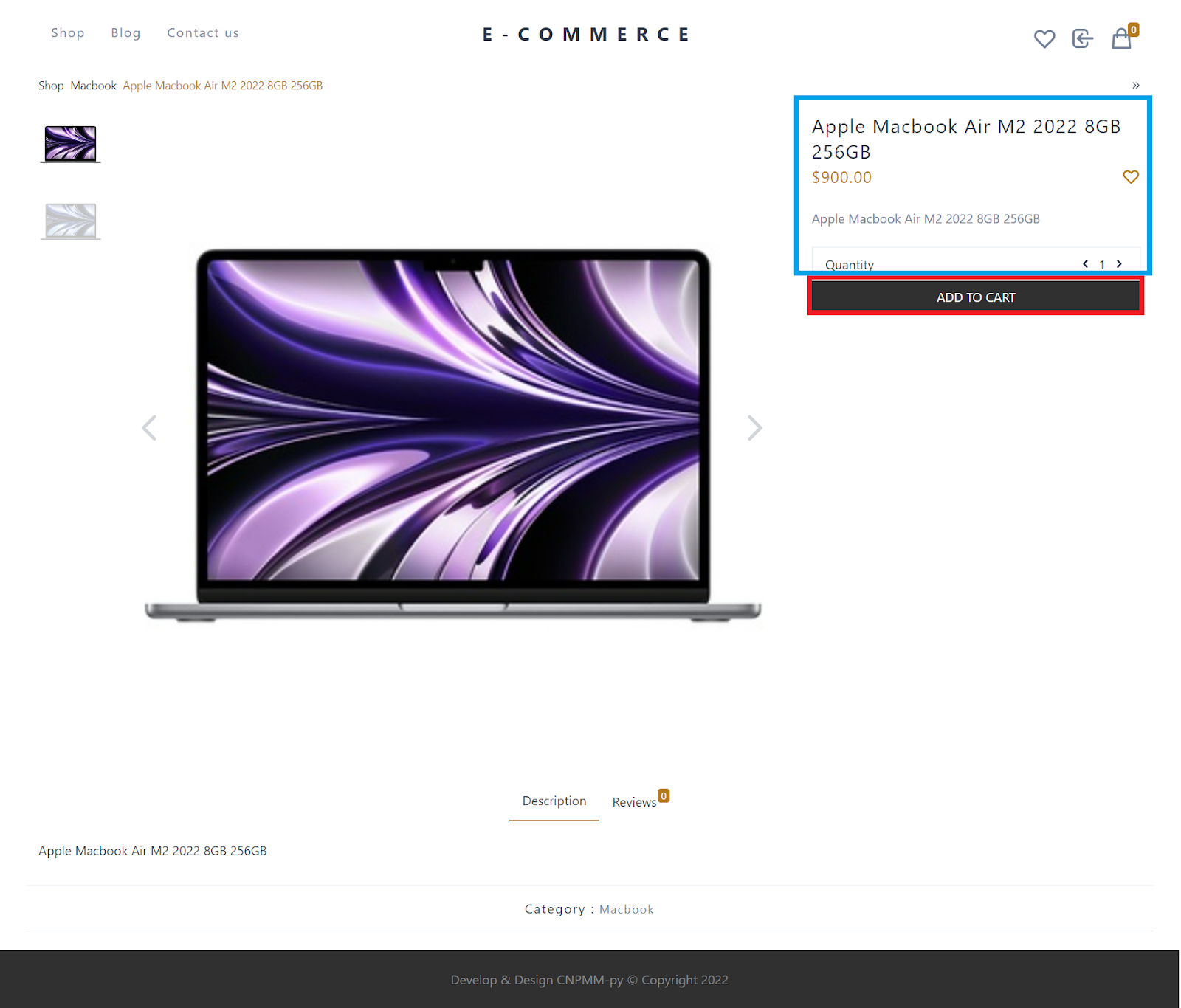
* Hoặc người dùng có thể tìm kiếm sản phẩm bằng cách lọc theo giá của sản phẩm: thực hiện kéo thanh giá để lọc giá từ 0 đến giá trị cần tìm



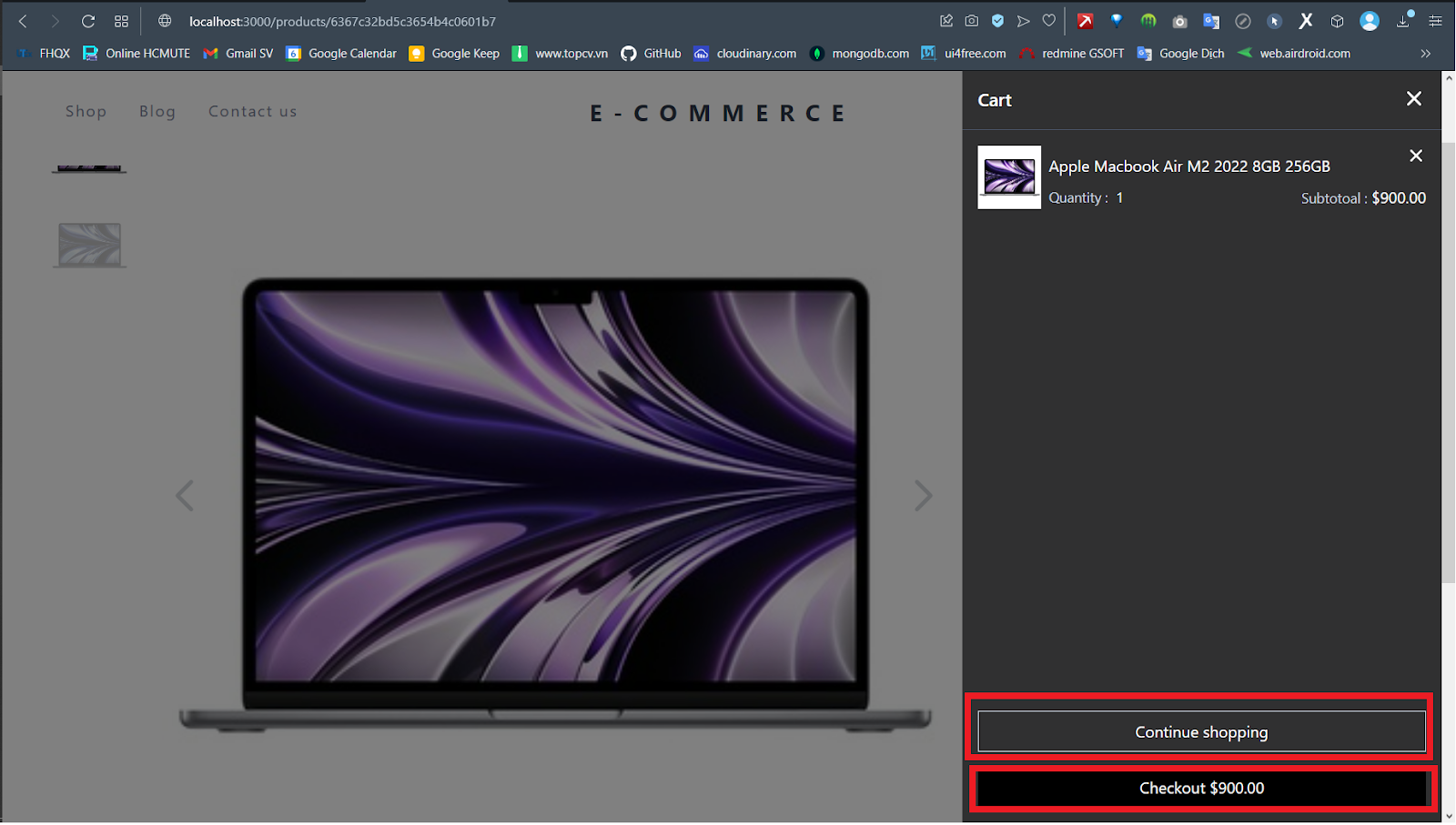
* Cách cơ bản nhất để tìm kiếm sản phẩm là tìm theo tên của sản phẩm: chỉ cần nhập tên sản phẩm và nhấn tìm kiếm là người dùng đã có thể tìm được những sản phẩm mà mình cần mua



* Chức năng tiếp theo khi sử dụng website mà không cần đăng nhập là xem chi tiết sản phẩm và thêm sản phẩm đó vào giỏ hàng: có thể lựa chọn số lượng sản phẩm và chọn thêm vào giỏ hàng



* Chức năng cuối cùng khi truy cập website mà không cần đăng nhập là xem giỏ hàng:
* 1. Chọn continue: người dùng sẽ được tiếp tục mua sắm
* 2. Chọn checkout: người dùng sẽ được yêu cầu đăng nhập để sử dụng chức năng này



**CHƯƠNG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

**Kết quả đạt được**

Dự án đã hoàn thành đầy đủ các chức năng cơ bản của một trang website thương mại điện tử về đồ nội thất và vật dụng gia đình.

Thông qua dự án em đã tìm hiểu và biết được cách mà một dự án MERN được tạo và hoạt động. Biết được cách xây dựng các API bằng NodeJS và ExpressJS. Xây dựng giao diện người dùng và tương tác với API bằng ReactJS. Hiểu được cách thao tác với cơ sở dữ liệu MongoDB và cách để Public một trang web lên Internet thông qua trình cung cấp hosting miễn phí Heroku.

Ngoài ra, trong quá trình xây dựng và phát triển dự án, em cũng đã học được cách quản lý dự án, quản lý tiến độ, cách để sử dụng Github quản lý Source Code và cập nhật các thay đổi của dự án thông qua từng phiên cập nhật.

Thông qua dự án, em cũng học được cách làm việc em trong một dự án phần mềm, cách phân chia thời gian cho từng phần việc cần phải làm.

**Ưu điểm**

- Thao tác đơn giản, dễ sử dụng và ổn định.

- Linh hoạt thay đổi giá món hàng, set up giá theo giờ cao điểm trong tương lai.

- Phân quyền chi tiết cho từng tài khoản, người sử dụng theo đúng chức năng và quyền hạn. Toàn bộ thông tin dữ liệu sẽ được đồng bộ, mã hóa để tăng tính bảo mật.

- Ứng dụng app mobile, tương thích trên 02 hệ điều hành android và ios, giúp các nhà quản lý yên tâm kiểm soát sản phẩm mọi lúc, mọi nơi.

- Xem được review trực tiếp của người dùng phản hồi ngay dưới phần mô tả sản phẩm

- Tích hợp kết nối OTA trên phần mềm

**Nhược điểm**

- Giao diện còn đơn giản

- Chức năng viết riêng nên khó đồng bộ

- Một số chức năng chưa hoàn thiện

**Khó khăn**

* MERN là một công nghệ mới và hiện tại là một trong những công nghệ phát triển nhất hiện nay cho nên các tài liệu về MERN không đồng nhất gây khó khăn cho những người mới tiếp cận.
* Thời gian còn hạn chế nên chưa phát triển tốt website.
* Cấu trúc thư mục chưa thực sự hiệu quả, còn hạn chế trong quá trình nâng cấp cũng như sửa lỗi.

**Bài học kinh nghiệm**

Tìm hiểu và xây dựng website dựa trên công nghệ MERN Stack đã giúp nhóm phân chia thời gian hợp lý và rút được một số kinh nghiệm sau:

* Về React: React được tạo ra xoay quanh khái niệm components tái sử dụng được. Tất cả các component đều có thể tái sử dụng. Tên component nên đặt theo chuẩn PascalCase. Tên component cần đề cập đúng mục đích của nó, phải rõ ràng và duy nhất trong ứng dụng, để dễ tìm thấy hơn và tránh những nhầm lẫn có thể xảy ra. nên chọn cách viết Stateless để định nghĩa component.
* Về NodeJS: NodeJs nó có tất cả các tính chất của javascript. Nodejs cũng xử lý bất đồng bộ. Nodejs chạy đơn luồng. Triển khai Node.js trên host không phải là điều dễ dàng.
* Về Expressjs: là một framework được xây dựng trên nền tảng của Nodejs. Cho phép thiết lập các lớp trung gian để trả về các HTTP request, trả về các trang HTML dựa vào các tham số.
* Về MongoDB: có thể insert một document với các key và value kiểu tuỳ ý vào một collection nào đó. Các mode khi sort với Mongodb lần lượt là: 1(tăng dần) và -1(giảm dần) thay vì esc và desc trong mysql. Trong SQL, count() và group by() là tương đương với Aggregation trong MongoDB

**Hướng phát triển**

* Tích hợp chatbot với các nền tảng mạng xã hội, tích hợp mã tracking.
* Thiết kế responsive, tạo giao diện ấn tượng với người dùng.

# TÀI LIỆU THAM KHẢO

[1] React – A JavaScript library for building user interfaces. (2021). Retrieved 24 December 2021, from <https://reactjs.org/>

[2] React-Bootstrap. (2021). Retrieved 24 December 2021, from <https://react-bootstrap.github.io/>

[3] Express.js and MongoDB REST API Tutorial. (2021). Retrieved 24 December 2021, from <https://www.mongodb.com/languages/express-mongodb-rest-api-tutorial>

[4] Hướng Dẫn Xây Dựng RESTful CRUD API Với NodeJS, Express và MongoDB. (2020). Retrieved 24 December 2021, from <https://www.thanhlongdev.com/huong-dan-xay-dung-restful-crud-api-voi-nodejs-express-va-mongodb/>

[5] JS, N, & OAuth2, N. (2021). NodeJS: Xây dựng API gửi mail với Nodemailer và OAuth2. Retrieved 24 December 2021, from <https://hocweb.vn/xay-dng-api-gi-mail-vi-nodemailer-va-oauth2-nodejs/>

[6] React Tutorials. (2021). Retrieved 24 December 2021, from <https://react-component-depot.netlify.app/button-loader>

[7] React Datepicker crafted by HackerOne. (2021). Retrieved 24 December 2021, from <https://reactdatepicker.com/>