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

KHOA ĐÀO TẠO CHẤT LƯỢNG CAO



CÔNG NGHỆ PHẦN MỀM MỚI

XÂY DỰNG WEBSITE BÁN SÁCH ONLINE

SVTH : VŨ NGỌC BỘI 17110102

NGUYỄN THÁI HOÁ 16110083

GVHD : NGUYỄN THANH PHƯỚC

TP. Hồ Chí Minh, tháng 12 năm 2020

# LỜI CẢM ƠN

Trước tiên em xin gửi lời cảm ơn chân thành sâu sắc tới các thầy cô giáo trong trường Đại học Sư phạm kĩ thuật – TP.Hồ Chí Minh nói chung và các thầy cô giáo trong khoa Công nghệ thông tin nói riêng đã tận tình giảng dạy, truyền đạt cho em những kiến thức, kinh nghiệm quý báu trong suốt thời gian bốn năm học vừa qua.

Đặc biệt em xin gửi lời cảm ơn đến thầy giáo Nguyễn Thanh Phước đã tận tình giúp đỡ, trực tiếp chỉ bảo, huớng dẫn em trong suốt quá trình thực hiện học phần Dự án. Trong thời gian làm việc, em không ngừng tiếp thu thêm nhiều kiến thức bổ ích mà còn học tập được tinh thần làm việc, thái độ nghiên cứu khoa học nghiêm túc, hiệu quả, đây là những điều rất cần thiết cho em trong quá trình học tập và công tác sau này.

Sau cùng xin gửi lời cảm ơn chân thành tới gia đình, bạn bè đã động viên, đóng góp ý kiến và giúp đỡ em trong quá trình học tập, nghiên cứu và hoàn thành khoá luận tốt nghiệp.

Một lần nữa, em xin chân thành cảm ơn.!

# LỜI MỞ ĐẦU

Ngày nay, với sự phát triển mạnh mẽ của khoa học công nghệ, Công nghệ Thông tin là một trong những ngành có vị thế dẫn đầu và có vai trò rất lớn trong sự phát triển chung đó. Các ứng dụng của công nghệ thông tin được áp dụng trong mọi lĩnh vực nghiên cứu khoa học cũng như trong mọi lĩnh vực của đời sống. Là một phần của Công nghệ Thông tin, Công nghệ web đang có được sự phát triển mạnh mẽ và phổ biến rất nhanh bởi những lợi ích mà nó mang lại cho cộng đồng là rất lớn.

Bằng việc lựa chọn và thực hiện đề tài “Xây dựng website bán sách online”, em muốn tìm hiểu và đưa ra một giải pháp tốt nhằm giải quyết công việc bán hàng trực tuyến cho công ty. Website mang lại cho công ty rất nhiều lợi ich như: khả năng quảng cáo, phổ biến tên tuổi công ty rất nhanh trên môi trường mạng toàn cầu, việc kinh doanh sẽ mở 24/24, giảm thiểu chi phí nhân viên cho tiếp thị, quảng cáo, dễ dàng nhận phải hồi từ phía khách hàng, cơ hôi mở rộng liên kết và hợp tác ở phạm vi quốc tế...v..v.

Website bán hàng giúp cho công ty cung cấp cho khách hàng những lựa chọn linh hoạt và tiện lợi trong việc tìm mua sản phẩm thông qua chức năng tìm kiếm và giỏ hàng. Các thông tin về sản phẩm được hiển thị chi tiết với từng sản phẩm, từ đó khách hàng dễ dàng nhận biết và lựa chọn được thứ mình cần. Về phía Công ty, hệ thống mang lại sự tiện dụng trong việc cập nhập và quản lý các thông tin cho website, tạo khả năng liên lạc nhanh và dễ dàng với khách hàng.

Nội dung đề tài gồm các chương:

Chương 1 : Tổng quan

Chương 2 : Cở sở lí thuyết

Chương 3 : Phân tích thiết kế hệ thống

Chương 4 : Cài đặt sản phẩm

Chương 5 : Kết luận và hướng phát triển

MỤC LỤC

[LỜI CẢM ƠN 1](#_Toc60625422)

[LỜI MỞ ĐẦU 2](#_Toc60625423)

[MỤC LỤC 3](#_Toc60625424)

[DANH MỤC CÁC TỪ VIẾT TẮT 5](#_Toc60625425)

[DANH MỤC CÁC BẢNG BIỂU 6](#_Toc60625426)

[DANH MỤC CÁC HÌNH ẢNH 1](#_Toc60625427)

[CHƯƠNG 1: TỔNG QUAN 2](#_Toc60625428)

[1.Lý do chọn đề tài 2](#_Toc60625429)

[2.Mục tiêu 2](#_Toc60625430)

[3.Công nghệ sử dụng: 2](#_Toc60625431)

[CHƯƠNG 2: CỞ SỞ LÍ THUYẾT 4](#_Toc60625432)

[1.Tìm hiểu các về ngôn ngữ lập trình 4](#_Toc60625433)

[1.1.HTML 4](#_Toc60625434)

[1.2.CSS 4](#_Toc60625435)

[1.4.JavaScript 5](#_Toc60625436)

[1.5.Reactjs 5](#_Toc60625437)

[1.6.Nodejs 7](#_Toc60625438)

[1.7.Mô hình MVC 8](#_Toc60625439)

[2.Một số Framework hỗ trợ 8](#_Toc60625440)

[2.1.Bootstrap 8](#_Toc60625441)

[2.2.Expess 9](#_Toc60625442)

[3.Một số Library hỗ trợ 9](#_Toc60625443)

[3.1.Material UI 9](#_Toc60625444)

[3.1.Body-Parser 9](#_Toc60625445)

[3.2.Mongoose 10](#_Toc60625446)

[3.3.Dotenv 10](#_Toc60625447)

[3.4.Axios 10](#_Toc60625448)

[3.5.Redux 10](#_Toc60625449)

[4.Cơ sở dữ liệu NoSQL 10](#_Toc60625450)

[4.1.NoSQL 10](#_Toc60625451)

[4.2.Mongodb 12](#_Toc60625452)

[CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 14](#_Toc60625453)

[1.Khảo sát và đặc tả yêu cầu 14](#_Toc60625454)

[1.1.Phần quản lí 14](#_Toc60625455)

[1.2.Giao diện người dùng 14](#_Toc60625456)

[2.Các yêu cầu phi chức năng (NFR) 14](#_Toc60625457)

[3.Các chứng năng của hệ thống (FR) 15](#_Toc60625458)

[4.Thiết kế cơ sở dữ liệu 15](#_Toc60625459)

[4.1.Collection Product 15](#_Toc60625460)

[4.2.Collection Category 16](#_Toc60625461)

[4.3.Collection User 16](#_Toc60625462)

[4.4.Collection Order 16](#_Toc60625463)

[4.5.Collection Favorite 17](#_Toc60625464)

[5.Yêu cầu chức năng 17](#_Toc60625465)

[5.1.Font-end 17](#_Toc60625466)

[5.1.1.Giao diện cho “Quản trị viên” và “Quản lí” 17](#_Toc60625467)

[5.1.2.Giao diện cho “Khách” 22](#_Toc60625468)

[5.2.Back-end 29](#_Toc60625469)

[5.2.1.User Endpoint 29](#_Toc60625470)

[5.2.2.Product Endpoint 29](#_Toc60625471)

[5.2.3.Order Endpoint 30](#_Toc60625472)

[5.2.4.Category Endpoint 30](#_Toc60625473)

[5.2.5.Favorie Endpoint 31](#_Toc60625474)

[5.2.6.Paypal Endpoint 31](#_Toc60625475)

[5.2.7.Search Endpoint 31](#_Toc60625476)

[6.Thiết kế xử lí đơn giản 32](#_Toc60625477)

[6.1.Font-end 32](#_Toc60625478)

[6.1.1.Liệt kê các Module 32](#_Toc60625479)

[6.1.2.Thiết kế ứng dụng 34](#_Toc60625480)

[6.2.Back-end 39](#_Toc60625481)

[6.2.1.Liệt kê thư viện 39](#_Toc60625482)

[CHƯƠNG 4: CÀI ĐẶT SẢN PHẨM 41](#_Toc60625483)

[1.Front-end 41](#_Toc60625484)

[2.Back-end 42](#_Toc60625485)

[3.Deployment 43](#_Toc60625486)

[3.1.Tạo database trên mongodb 43](#_Toc60625487)

[3.2.Tải ứng dụng lên Heroku 46](#_Toc60625488)

[CHƯƠNG 5: TỔNG KẾT 49](#_Toc60625489)

[1.Kết quả đạt được 49](#_Toc60625490)

[2.Ưu điểm 49](#_Toc60625491)

[3.Nhược điểm 49](#_Toc60625492)

[4.Khó khăn 49](#_Toc60625493)

[5.Hướng phát triển 49](#_Toc60625494)

# DANH MỤC CÁC TỪ VIẾT TẮT

|  |  |  |  |
| --- | --- | --- | --- |
| STT | TỪ VIẾT TẮT | TÊN ĐẦY ĐỦ | GIẢI THÍCH |
| 1 | MERN STACK | MongoDB, ExpressJS, React/React Native, NodeJS. | MERN stack là combo open source các công nghệ liên quan đến Javascript. |
| 2 | MVC | **Model – View – Controller**. | **MVC là 1 design pattern được sử dụng phổ biến trong lập tình web** |
| 3 | JSX | JavaScript XML | JSX là một cú pháp mở rộng cho JavaScrip |
| 4 | FR | Functional Requirements | Là những yêu cầu mô tả những gì hệ thống phải làm được.Ví dụ như: Hiển thị tên, màu sắc, .... |
| 5 | NFR | NonFunctional Requirements | Là các tiêu chí mà hệ thống phần mềm cần đạt được khi cho thực thi. Ví dụ: hiệu suất, khả năng mở rộng, tính bảo mật, tương thích,… |
| 6 | API | Application Programming Interface | Giao diện lập trình ứng dụng. |
| 7 | NoSQL | None-Relational Structured Query Language | Cơ sở dữ liệu phi quan hệ |

# DANH MỤC CÁC BẢNG BIỂU

Bảng 1.Collection Product

Bảng 2.Collection Category

Bảng 3.Collection User

Bảng 4.Collection Order

Bảng 5.Collection Favorite

# DANH MỤC CÁC HÌNH ẢNH

Hình 1.Menu Quản trị viên

Hình 2.Menu Quản lí

Hình 3.Giao diện danh sách sản phẩm

Hình 4.Giao diện cập nhật sản phẩm

Hình 5.Giao diện danh sách người dùng

Hình 6.Giao diện cập nhật người dùng

Hình 7.Giao diện danh sách loại sản phẩm

Hình 8.Giao diện cập nhật loại sản phẩm

Hình 9.Giao diện danh sách đơn đặt hàng

Hình 10.Giao diện thông tin chi tiết đơn hàng

Hình 11.Giao diện trang chủ

Hình 12.Giao diện trang chủ

Hình 13.Giao diện trang chi tiết sản phẩm

Hình 14.Giao diện giỏ hàng

Hình 15.Giao diện Shipping

Hình 16.Giao diện Payment Method

Hình 17.Giao diện PlaceOrder

Hình 18.Giao diện chi tiết đơn đặt hàng

Hình 19.Giao diện trang cập nhật thông tin cá nhân

Hình 20.Giao diện danh sách đơn hàng cá nhân

Hình 21.Giao diện trang toàn bộ sản phẩm

Hình 22.Giao diện trang tìm kiếm

Hình 23.Giao diện trang sản phẩm yêu thích

# CHƯƠNG 1: TỔNG QUAN

## 1.Lý do chọn đề tài

Việc lựa chọn và thực hiện đề tài “*Xây dựng website bán sách online” là vì* em muốn tìm hiểu và đưa ra một giải pháp tốt nhằm giải quyết công việc bán hàng trực tuyến cho công ty

2.Mục tiêu

* Xây dựng các chức năng cơ bản của một website bán hàng thương mại. Website có khả năng tự tương thích, hiển thị được trên tất cả các thiết bị hiện tại và có thể nâng cấp trong tương lai.
* Website hiển thị sản phẩm đẹp, thu hút người dùng.
* Hỗ trợ khách hàng một cách nhanh nhất khi nhận được yêu cầu liên hệ.
* Quản lý các mặt hàng sản phẩm, cấu hình website.
* Quản lý thống kê doanh thu, các đơn hàng sản phẩm, số lượng sản phẩm tồn đọng.
* Nắm bắt được công nghệ thiết kế web bằng:, HTML, CSS, JAVASCRIPT, NOSQL,…
* Ứng dụng mô hình MVC vào xây dựng trang web.
* Ứng dụng Nodejs, Reactjs vào xây dựng trang web.

## 3.Công nghệ sử dụng:

* Website xây dựng theo hướng MERN STACK, mô hình MVC.
* Font-End: Reactjs, HTML, CSS.
* Back\_end: Nodejs.
* Database: NoSQL là MongoDB.
* Một vài Framework:
  + Bootstrap
  + Express
* Một vài Library:
  + Material UI
  + Mongoose
  + Body-Parser
  + Dotenv
  + Axios
  + Redux
  + …..
* Service:
  + Cloudinary (Xử lí lưu trữ hình ảnh trực tuyến).
* Deployment: Triển khai ứng dụng lên nền tảng Heroku.

# CHƯƠNG 2: CỞ SỞ LÍ THUYẾT

## 1.Tìm hiểu các về ngôn ngữ lập trình

### 1.1.HTML

HTML (HyperText Markup Language) – Ngôn ngữ đánh dấu siêu văn bản được sử dụng để tạo các tài liệu có thể truy cập trên mạng. Tài liệu HTML được tạo nhờ dùng các thẻ và các phần tử của HTML. File được lưu trên máy chủ dịch vụ web với phần mở rộng “.htm” hoặc “.html”. Các trình duyệt sẽ đọc tập tin HTML và hiển thị chúng dưới dạng trang web. Các thẻ HTML sẽ được ẩn đi, chỉ hiển thị nội dung văn bản và các đối tượng khác: hình ảnh, media. Với các trình duyệt khác nhau đều hiển thị một tập HTML với một kết quả nhất định. Các trang HTML được gửi đi qua mạng internet theo giao thức HTTP. HTML không những cho phép nhúng thêm các đối tượng hình ảnh, âm thanh mà còn cho phép nhúng các kịch bản vào trong đó như các ngôn ngữ kịch bản như Javascript để tạo hiệu ứng động cho trang web. Để trình bày trang web hiệu quả hơn thì HTML cho phép sử dụng kết hợp với CSS. HTML không những cho phép nhúng thêm các đối tượng hình ảnh, âm thanh mà còn cho phép nhúng các kịch bản vào trong đó như các ngôn ngữ kịch bản như Javascript để tạo hiệu ứng động cho trang web.

HTML là một chuẩn ngôn ngữ internet được tạo ra và phát triển bởi tổ chức World Wide Web Consortium còn được viết tắc là W3C. Trước đó thì HTML xuất bản theo chuẩn của RFC. HTML được tương thích với mọi hệ điều hành cùng các trình duyệt của nó. Khả năng dễ học, dễ viết là một ưu điểm của HTML không những vậy việc soạn thảo đòi hỏi hết sức đơn giản, chúng ta có thể dùng word, notepad hay bất cứ một trình soạn thảo văn bản nào để viết và chỉ cần lưu với định dạng “.html “ hoặc “.htm” là đã có thể tạo ra một file chứa HTML. Hiện nay, phiên bản mới nhất của HTML là HTML5 với nhiều tính năng ưu việt so với các phiên bản cũ HTML cải tiến khá nhiều đặc biệt hỗ trợ mạnh mẽ các phần tử multimedia mà không cần các plugin. Một tập tin HTML bao gồm trong đó là các đoạn văn bản HTML, được tạo lên bởi các thẻ HTML. HTML5 nói chung mạnh mẽ hơn nhiều không chỉ về tốc độ và độ thích ứng cao mà chính là khả năng hỗ trợ API (Application Programming Interface - giao diện lập trình ứng dụng) và DOM (Document Object Model – các đối tượng thao tác văn bản).

### 1.2.CSS

CSS (Cascading Style Sheets) là một ngôn ngữ quy định cách trình bày cho các tài liệu viết bằng HTML, XHTML, XML, SVG, hay UML,…CSS quy định cách hiển thị của các thẻ HTML bằng cách quy định các thuộc tính của các thẻ đó (font chữ, kích thước, màu sắc...).

CSS có cấu trúc đơn giản và sử dụng các từ tiếng anh để đặt tên cho các thuộc tính. CSS khi sử dụng có thể viết trực tiếp xen lẫn vào mã HTML hoặc tham chiếu từ một file css riêng biệt. Hiện nay CSS thường được viết riêng thành một tập tin với mở rộng là “.css”. Chính vì vậy mà các trang web có sử dụng CSS thì mã HTML sẽ trở nên ngắn gọn hơn. Ngoài ra có thể sử dụng một tập tin CSS đó cho nhiều website tiết kiệm rất nhiều thời gian và công sức. Một đặc điểm quan trọng đó là tính kế thừa của CSS do đó sẽ giảm được số lượng dòng code mà vẫn đạt được yêu cầu.

Tuy nhiên, đối với CSS thì các trình duyệt hiểu theo kiểu riêng của nó. Do vậy, việc trình bày một nội dung trên các trình duyệt khác nhau là không thống nhất. CSS cung cấp hàng trăm thuộc tính trình bày dành cho các đối tượng với sự sáng tạo cao trong kết hợp các thuộc tính giúp mang lại hiệu quả.

### 1.4.JavaScript

JavaScript là một ngôn ngữ dạng script thường được sử dụng cho việc lập trình web ở phía client, nó tuân theo chuẩn ECMAScript. Là một ngôn ngữ linh động, cú pháp dễ sử dụng như các ngôn ngữ khác và dễ dàng lập trình. JavaScript không hề liên quan tới ngôn ngữ lập trình java, được hầu hết các trình duyệt ngày nay hỗ trợ. Với javascript, ứng dụng web của bạn sẽ trở nên vô cùng sinh động, mang tính trực quan và tương tác cao. JavaScript theo phiên bản hiện hành 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. Giống Java, JavaScript có cú pháp tương tự ngôn ngữ lập trình C. “.js” là phần mở rộng thường được dùng cho tập tin mã nguồn JavaScript.

### 1.5.Reactjs

ReactJS là thư viện JavaScript được sử dụng để xây dựng các thành phần UI có thể tái sử dụng. React được sử dụng tại Facebook trong production, và instagram được viết hoàn toàn trên React.

Một trong những điểm hấp dẫn của React là thư viện này không chỉ hoạt động trên phía client, mà còn được render trên server và có thể kết nối với nhau. React so sánh sự thay đổi giữa các giá trị của lần render này với lần render trước và cập nhật ít thay đổi nhất trên DOM. React cũng có thể hiển thị trên máy chủ bằng Node và nó có thể cung cấp năng lượng cho các ứng dụng gốc bằng React Native.

*Tính năng của React:*

* JSX - JSX là phần mở rộng cú pháp JavaScript. Không nhất thiết phải sử dụng JSX trong phát triển React, nhưng nó được khuyến khích.
* Component - React bao gồm tất cả các component. Là một đoạn mã ngắn có ý nghĩa thể hiện một chức năng hay đối tượng nhất định, có thể tái sử dụng. Một ứng dụng React có thể có hàng chục, hoặc hàng trăm các component liên kết, tương tác với nhau. Bạn cần nghĩ mọi thứ như một component. Điều này sẽ giúp bạn duy trì mã khi làm việc trên các dự án quy mô lớn hơn.
* Luồng dữ liệu một chiều và Flux - React triển khai luồng dữ liệu một chiều giúp bạn dễ dàng quản lý về ứng dụng của mình. Flux là một kiến trúc mà Facebook sử dụng trong khi làm việc với React. Flux không phải là một framework hay một thư viện (library). Nó đơn giản chỉ là một kiểu kiến trúc mới hỗ trợ thêm cho React, đồng thời xây dựng ý tưởng về luồng dữ liệu một chiều (Unidirectional Data Flow).
* Giấy phép - React được cấp phép theo Facebook Inc. Tài liệu được cấp phép theo CC BY 4.0.

*Ưu điểm của React:*

* Reactjs cực kì hiệu quả: Reactjs tạo ra cho chính nó DOM ảo – nơi mà các component thực sự tồn tại trên đó. Điều này sẽ giúp cải thiện hiệu suất rất nhiều, vì DOM ảo JavaScript nhanh hơn DOM thông thường.
* Có thể được sử dụng trên máy khách và máy chủ cũng như với các khuôn khổ khác.
* Các mẫu thành phần và dữ liệu cải thiện khả năng đọc, giúp duy trì các ứng dụng lớn hơn.

*Nhược điểm của React:*

* Reactjs chỉ phục vụ cho tầng View. React chỉ là View Library nó không phải là một MVC framework như những framework khác. Đây chỉ là thư viện của Facebook giúp render ra phần view. Vì thế React sẽ không có phần Model và Controller, mà phải kết hợp với các thư viện khác. Do đó bạn vẫn cần chọn các công nghệ khác để có được bộ công cụ hoàn chỉnh để phát triển.

*Lí do được sử dụng cho đồ án:*

* Xây dựng website theo Mern Stack.
* Tách website thành các component nhỏ hơn trong việc viết code, xử lí, maintain.
* Xây dựng website theo hướng SPA (Single Page Application).
* Được FaceBook và cộng đồng hỗ trợ nhiều.
* Theo cá nhân em thì Reactj dễ học và làm hơn một chút so với ngôn ngữ khác, cụ thể như Angular, Vuejs,.

### 1.6.Nodejs

Nodejs là một nền tảng (Platform) phát triển độc lập được xây dựng ở trên Javascript Runtime của Chrome mà chúng ta có thể xây dựng được các ứng dụng mạng một cách nhanh chóng và dễ dàng mở rộng.

Nodejs được xây dựng và phát triển từ năm 2009, bảo trợ bởi công ty Joyent, trụ sở tại California, Hoa Kỳ.

Phần Core bên dưới của Nodejs được viết hầu hết bằng C++ nên cho tốc độ xử lý và hiệu năng khá cao. Nodejs tạo ra được các ứng dụng có tốc độ xử lý nhanh, realtime thời gian thực. Nodejs áp dụng cho các sản phẩm có lượng truy cập lớn, cần mở rộng nhanh, cần đổi mới công nghệ, hoặc tạo ra các dự án Startup nhanh nhất có thể.

*Những ứng dụng nên viết bằng nodejs:*

Rõ ràng, không phải cứ hot và mới là Nodejs làm gì cũng tốt, ví dụ như một ứng dụng cần tính ổn định cao, logic phức tạp thì các ngôn ngữ PHP hay Ruby… vẫn là sự lựa chọn tốt hơn. Còn dưới đây là những ứng dụng có thể và nên viết bằng Nodejs:

* Websocket server: Các máy chủ web socket như là Online Chat, Game Server…
* Fast File Upload Client: là các chương trình upload file tốc độ cao.
* Ad Server: Các máy chủ quảng cáo.
* Cloud Services: Các dịch vụ đám mây.
* RESTful API: đây là những ứng dụng mà được sử dụng cho các ứng dụng khác thông qua API.
* Any Real-time Data Application: bất kỳ một ứng dụng nào có yêu cầu về tốc độ thời gian thực. Micro Services: Ý tưởng của micro services là chia nhỏ một ứng dụng lớn thành các dịch vụ nhỏ và kết nối chúng lại với nhau. Nodejs có thể làm tốt điều này.

*Lí do sử dụng cho đồ án:*

* Xây dựng website theo Mern Stack.
* Vì xây dựng website liên quan đến Reactjs, SPA, MongoDB,.. điều này sẽ giúp hệ thống hoàn toàn sử dụng JavaScript.
* Có sự hỗ trợ lớn của cộng đồng và đây là một nền tảng khá phát triển hiện nay.
* Theo cá nhân em thì Nodejs cũng dễ học, nhiều gói (package) hỗ trợ trong việc lập trình.

### 1.7.Mô hình MVC

MVC (viết tắt của Model-View-Controller) là một mẫu kiến trúc phần mềm để tạo lập giao diện người dùng trên máy tính. MVC chia một ứng dụng thành ba phần tương tác được với nhau để tách biệt giữa cách thức mà thông tin được xử lý nội hàm và phần thông tin được trình bày và tiếp nhận từ phía người dùng.

Khi sử dụng đúng cách, mẫu MVC giúp cho người phát triển phần mềm cô lập các nguyên tắc nghiệp vụ và giao diện người dùng một cách rõ ràng hơn. Phần mềm phát triển theo mẫu MVC tạo nhiều thuận lợi cho việc bảo trì vì các nguyên tắc nghề nghiệp và giao diện ít liên quan với nhau.

Trong mẫu Model-View-Controller, mô hình (model) tượng trưng cho dữ liệu của chương trình phần mềm. Tầm nhìn hay khung nhìn (view) bao gồm các thành phần của giao diện người dùng. Bộ kiểm tra hay bộ điều chỉnh (controller) quản lý sự trao đổi giữa dữ liệu và các nguyên tắc nghề nghiệp trong các thao tác liên quan đến mô hình.

## 2.Một số Framework hỗ trợ

### 2.1.Bootstrap

Bootstrap là một Framework có chứa HTML, CSS, JAVASCRIPT, Framework trong tiếng việt có nghĩa là “khuôn khổ” giúp tiết kiệm được thời gian, công sức hơn nữa việc xây dựng hai teamplate cho giao diện Desktop và Mobile đã lỗi thời thay vào đó là Responsive. Responsive sẽ giúp website của bạn hiển thị tương thích với mọi kích thước màn hình nhờ đó bạn sẽ tùy chỉnh hiện thị được nhiều hơn trên các loại màn hình khác nhau.

*Ưu điểm của Bootstrap:*

* Tiết kiệm thời gian: Boostrap giúp người thiết kế giao diện website tiết kiệm rất nhiều thời gian. Các thư viện Bootstrap có những đoạn mã sẵn sàng cho

bạn áp dùng vào website của mình. Bạn không phải tốn quá nhiều thời gian để tự viết code cho giao diện của mình.

* Tùy biến cao: Bạn hoàn toàn có thể dựa vào Bootstrap và phát triển nền tảng giao diện của chính mình. Bootstrap cung cấp cho bạn hệ thống Grid System mặc định bao gồm 12 bột và độ rộng 940px. Bạn có thể thay đổi, nâng cấp và phát triển dựa trên nền tảng này.
* Responsive Web Design: Với Bootstrap, việc phát triển giao diện website để phù hợp với đa thiết bị trở nên dễ dàng hơn bao giờ hết. Đây là xu hướng phát triển giao diện website đang rất được ưu chuộng trên thế giới

### 2.2.Expess

Express cung cấp những công cụ cần thiết để ta có thể xây dựng nên một ứng dụng, với rất nhiều những modules (các đơn vị làm những chức năng khác nhau) có sẵn trên ****npm****, những module này có thể giúp bạn dễ dàng thêm vào ứng dụng của mình, để giảm thời gian phải xây dựng các tính năng đã có sẵn cho ứng dụng của bạn.

Express hướng tới sự linh động, chỉ những gì bạn cần thì bạn mới phải thêm vào, không giống như những đối thủ như là Rails hay Django, hướng tới một cách cứng nhắc để xây dựng ứng dụng.

Bằng cách mở rộng các đặc tính của NodeJs, ExpressJs cung cấp các tính năng như middlewares (hàm xử xý trung gian), đơn giản hóa khả năng điều hướng của ứng dụng, kèm theo tích hợp hỗ trợ rất nhiều template engine, giúp bạn tương tác dễ dàng hơn với HTML.

## 3.Một số Library hỗ trợ

### 3.1.Material UI

Material UI là một thư viện các React Component đã được tích hợp thêm cả Google’s Material Design.

Material-UI cũng cấp cho bạn khá đầy đủ các component để có thể tạo ra một trang web một cách nhanh chóng hơn, mà không phải đi ngồi css từng tí một.

Material-UI cũng cung cấp cho bạn các icon để có thể sử dụng trong trang web.

### 3.1.Body-Parser

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

### 3.2.Mongoose

Mongoose là một thư viện mô hình hóa đối tượng (Object Data Model - ODM) cho MongoDB, nó giúp cho lập trình viên dễ dàng truy cập và tương tác với cơ sở dữ liệu trong ứng dụng Express.

### 3.3.Dotenv

Dotenv là một mô đun zero-dependency, tải các biến môi trường từ tệp env vào process.env

### 3.4.Axios

Axios là một thư viện HTTP Client dựa trên Promise. Cơ bản thì nó cung cấp một API cho việc xử lý XHR (XMLHttpRequests). Trong bài viết này, chúng ta sẽ tìm hiểu làm thế nào để thực hiện một HTTP request bằng cách sử dụng axios nhé.

### 3.5.Redux

Redux js là một thư viện Javascript giúp tạo ra thành một lớp quản lý trạng thái của ứng dụng.

Redux được xây dựng dựa trên nền tảng tư tưởng của ngôn ngữ Elm và kiến trúc Flux do Facebook giới thiệu. Redux thường là bộ đôi kết hợp hoàn hảo với React. Tuy nhiên hoàn toàn có thể sử dụng với các framework khác như Angular, Angular2, Backbone, Falcor, Deku.

Note: Reduxjs là một thư viện Javascript khác với Redux Framework là một WordPress Framework.

## 4.Cơ sở dữ liệu NoSQL

### 4.1.NoSQL

*NoSQL database là gì ?*

Thuật ngữ NoSQL được giới thiệu lần đầu vào năm 1998 sử dụng làm tên gọi chung cho các lightweight open source relational database (cơ sở dữ liệu quan hệ nguồn mở nhỏ) nhưng không sử dụng SQL cho truy vấn. Vào năm 2009, Eric Evans, nhân viên của Rackspace giới thiệu lại thuật ngữ NoSQL trong một hội thảo về cơ sở dữ liệu nguồn mở phân tán. Thuật ngữ NoSQL đánh dấu bước phát triển của thế hệ database mới: distributed (phân tán) + non-relational (không ràng buộc). Đây là 2 đặc tính quan trọng nhất.

*Một số đặc điểm chung:*

Có bốn loại chung (loại phổ biến nhất) của cơ sở dữ liệu NoSQL. Mỗi loại đều có các thuộc tính và giới hạn riêng. Không có một giải pháp duy nhất nào tốt hơn tất cả các giải pháp khác, tuy nhiên có một số cơ sở dữ liệu tốt hơn để giải quyết các vấn đề cụ thể.

* High Scalability: Gần như không có một giới hạn cho dữ liệu và người dùng trên hệ thống.
* Atomicity: Độc lập data state trong các operation.
* High Availability: Do chấp nhận sự trùng lặp trong lưu trữ nên nếu một node (commodity machine) nào đó bị chết cũng không ảnh hưởng tới toàn bộ hệ thống.
* Consistency: chấp nhận tính nhất quán yếu, có thể không thấy ngay được sự thay đổi mặc dù đã cập nhật dữ liệu.
* Durability: dữ liệu có thể tồn tại trong bộ nhớ máy tính nhưng đồng thời cũng được lưu trữ lại đĩa cứng.
* Deployment Flexibility: việc bổ sung thêm/loại bỏ các node, hệ thống sẽ tự động nhận biết để lưu trữ mà không cần phải can thiệp bằng tay. Hệ thống cũng không đòi hỏi cấu hình phần cứng mạnh, đồng nhất.
* Modeling flexibility: Key-Value pairs, Hierarchical data (dữ liệu cấu trúc), Graphs.
* Query Flexibility: Multi-Gets, Range queries (load một tập giá trị dựa vào một dãy các khóa).

*Phân loại NoSQL database*:

Có bốn loại chung (loại phổ biến nhất) của cơ sở dữ liệu NoSQL. Mỗi loại đều có các thuộc tính và giới hạn riêng. Không có một giải pháp duy nhất nào tốt hơn tất cả các giải pháp khác, tuy nhiên có một số cơ sở dữ liệu tốt hơn để giải quyết các vấn đề cụ thể.

* Key-Value Database
* Document Database
* Column-Family Database
* Graph Database

*So sánh giữa RDBMS và NoSQL:*

RDBMS

* Dữ liệu có cấu trúc và tổ chức
* Sử dụng ngôn ngữ SQL để truy vấn dữ liệu
* Dữ liệu và các mối quan hệ của nó được lưu trữ trong các bảng riêng biệt.
* Có tính chặt chẽ

NoSQL

* Không sử dụng SQL
* Không khai báo ngôn ngữ truy vấn dữ liệu
* Không định nghĩa schema
* Có 1 số nhóm dạng: Key-Value pair storage, Column Store, Document Store, Graph databases
* Dữ liệu phi cấu trúc và không thể đoán trước
* Ưu tiên cho hiệu năng cao, tính sẵn sàng cao và khả năng mở rộng

### 4.2.Mongodb

MongoDB là một hệ quản trị cơ sở dữ liệu mã nguồn mở, là CSDL thuộc NoSql và được hàng triệu người sử dụng.

MongoDB là một database hướng tài liệu (document), các dữ liệu được lưu trữ trong document kiểu JSON thay vì dạng bảng như CSDL quan hệ nên truy vấn sẽ rất nhanh.

Với CSDL quan hệ chúng ta có khái niệm bảng, các cơ sở dữ liệu quan hệ (như MySQL hay SQL Server...) sử dụng các bảng để lưu dữ liệu thì với MongoDB chúng ta sẽ dùng khái niệm là collection thay vì bảng

So với RDBMS thì trong MongoDB collection ứng với table, còn document sẽ ứng với row , MongoDB sẽ dùng các document thay cho row trong RDBMS.

Các collection trong MongoDB được cấu trúc rất linh hoạt, cho phép các dữ liệu lưu trữ không cần tuân theo một cấu trúc nhất định.

Thông tin liên quan được lưu trữ cùng nhau để truy cập truy vấn nhanh thông qua ngôn ngữ truy vấn MongoDB

*Ưu điểm của mongoDB:*

* Ít lược đồ: MongoDB là một cơ sở dữ liệu hướng document trong đó một collection chứa các document khác nhau. Số lượng trường, nội dung và kích thước của document có thể khác nhau từ document này sang document khác. Dữ liệu không đồng nhất, không ràng buộc.
* Trường “\_id” luôn được tự động đánh index giúp tốc độ truy vấn thông tin đạt hiệu năng cao nhất.
* Cấu trúc của một đối tượng được biểu diễn rõ ràng dưới dạng JSON.
* Hỗ trợ dữ liệu không gian địa lý.
* Không joins phức tạp.
* Khả năng truy vấn sâu: MongoDB hỗ trợ các truy vấn động trên các documents bằng ngôn ngữ truy vấn hướng document mạnh mẽ tương tự như SQL.
* Khả năng điều chỉnh, dễ phân tán.
* Dễ mở rộng theo chiều ngang, hiệu năng cao, tính sẵn sàng cao.
* Linh hoạt: cập nhật các trường sẽ ít hoặc không ảnh hưởng tới ứng dụng.
* Tích hợp dễ dàng với Big Data Hadoop.
* Chuyển đổi/ánh xạ các đối tượng trong ứng dụng thành các đối tượng cơ sở dữ liệu không cần thiết.
* Phân tán đám mây như AWS, Microsoft, RedHat,dotCloud and SoftLayer, ...
* Sử dụng bộ nhớ trong để lưu trữ bộ làm việc, cho phép truy cập dữ liệu nhanh hơn. Khi truy vấn dữ liệu, bản ghi kết quả được cached trên RAM, giúp lần truy vấn sau nhanh hơn mà không cần đọc lại từ ổ cứng.

*Nhược điểm của mongoDB:*

* MongoDB không có các tính chất ràng buộc mạnh mẽ như trong RDBMS (ACID) nên khi thao tác với MongoDB cần hết sức cẩn thận.
* Không hỗ trợ transaction hoặc transaction rất phức tạp.
* Không hỗ trợ join nên dễ trùng lặp dữ liệu.
* Sử dụng nhiều bộ nhớ do dữ liệu lưu dưới dạng key-value, các collection chỉ khác về value và key bị lặp lại.
* Giới hạn kích thước bản ghi (< 16Mb).
* Khi cập nhật dữ liệu, MongoDB sẽ không cập nhật ngay xuống ổ cứng, sau 60s mới thực hiện ghi toàn bộ dữ liệu thay đổi từ RAM sang ổ cứng. Do vậy, sẽ có khả năng mất dữ liệu khi xảy ra các trường hợp không mong muốn.
* Không hỗ trợ function/stored procedure để ràng buộc logic.

# CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## 1.Khảo sát và đặc tả yêu cầu

### 1.1.Phần quản lí

* Admin quản lý toàn bộ hoạt động của cửa hàng.
* Quản lý giao dịch thanh toán, mua hàng, xử lý các đơn đặt hàng.
* Quản lý hóa đơn: khi mua hàng khách hàng sẽ có hóa đơn mua gồm số tên khách hàng, địa chỉ người nhận, ngày đặt hàng,…
* Khi cửa hàng có nhu cầu nhập hàng thì chủ cửa hàng là người trực tiếp liên hệ với nhà cung cấp nhập thêm hàng vào cửa hàng.
* Khi nhà cung cấp mang đến, chủ cửa hàng có nhiệm vụ kê số lượng hàng nhập vào, kiểm tra chất lượng hàng nhập xem có đúng tiêu chuẩn và đúng với yêu cầu của cửa hàng cần hay không.
* Thống kê tổng số đơn đặt hàng trong ngày và trong tháng, thuận tiện cho người quản trị tiến hành xử lý các đơn hàng.
* Thống kê các mặt hàng tồn.
* Tư vấn trả lời các liên hệ của khách hàng.

### 1.2.Giao diện người dùng

* Giao diện các website đơn giản, dễ sủ dụng
* Phải có thông tin liên lạc với chủ cửa hàng (Số điện thoại, facebook, email,…)
* Hiển thị được các sản phẩm, các sản phẩm mới, sản phẩm đang sale,.. thì được gắn thêm thẻ để khách hàng biết.
* An toàn và bảo mật dữ liệu.
* Cho phép khách hàng xem thông tin chi tiết, tìm kiếm các sản phẩm, tin tức mà khách hàng quan tâm.
* Cho phép khách hàng đăng ký tài khoản để nhận các tin khuyến mãi, mua hàng được thuận tiện.
* Cho phép khách hàng thêm sản phẩm yêu thích vào tài khoản.
* Cho phép liên hệ với người quản trị thông qua chatbox ( Facebook hoặc bên thứ ba)

## 2.Các yêu cầu phi chức năng (NFR)

* Giao diện: giao diện hệ thống phải dễ sử dụng, trực quan, thân thiện với người dùng.
* Tốc độ xử lý: hệ thống phải xử lý nhanh chóng và chính xác.
* Bảo mật: tính bảo mật và độ an toàn cao.
* Tương thích: tương thích với đa phần các trình duyệt web hiện tại.
* Khả năng mở rộng: có thể thêm chức năng mới vào hệ thống.

## 3.Các chứng năng của hệ thống (FR)

* Quản trị viên (Admin):
  + Quản lý danh mục sản phẩm
  + Quản lý sản phẩm
  + Quản lý danh sách người dùng
  + Quản lý các đơn hàng các giao dịch
  + Thống kê đơn hàng và sản phẩm.
  + Chat với khách hàng thông qua messenger facebook
* Quản lí : Có tất cả chức năng như quản trị viên ngoại trừ “Quản lí người dùng”
* Khách hàng:
  + Khách hàng có thể xem thông tin, tìm kiếm sản phẩm, thêm sản phẩm vào giỏ hàng khi chưa đăng nhập.
  + Khách hàng có thể đặt hàng, xem thông tin đơn hàng, thanh toán, thêm sản phẩm yêu thích sau khi đăng nhập.
  + Khách hàng có thể chat với nhà bán hàng qua messenger facebook.

## 4.Thiết kế cơ sở dữ liệu

### 4.1.Collection Product

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Field | Type | Description |
| 1 | Name | String | Tên sản phẩm |
| 2 | Category | String | Loại sản phẩm |
| 3 | Image | String | Ảnh sản phẩm |
| 4 | Price | Number | Giá sản phẩm |
| 5 | Rating | Number | Số lượng bình chọn sản phẩm |
| 6 | Number Reviews | Number | Số lượng bình luận |
| 7 | Tag | String | Thẻ sản phẩm |
| 8 | Description | String | Mô tả sản phẩm |
| 9 | Count in Stock | Number | Số lượng sản phẩm |

*Bảng 1.Collection Product*

### 4.2.Collection Category

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Field | Type | Description |
| 1 | Name | String | Tên loại sản phẩm |

*Bảng 2.Collection Category*

### 4.3.Collection User

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Field | Type | Description |
| 1 | Name | String | Tên người dùng |
| 2 | Email | String | Email người dùng |
| 3 | Password | String | Mật khẩu |
| 4 | Phone | Number | Số điện thoại |
| 5 | Is Admin | Boolean | Phân loại tài khoản là Admin |
| 6 | Is Staff | Boolean | Phân loại tài khoản là Staff |

*Bảng 3.Collection User*

### 4.4.Collection Order

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Field | Type | Description |
| 1 | OrderItems | Object | Chứa thông tin sản phẩm( tên, số lượng, loại ,giá,..) |
| 2 | ShippingAddress | Object | Chứa các thông tin địa chỉ đặt hàng gồm: address, city, postalCode, country |
| 3 | Payment Method | String | Phương thức thanh toán |
| 4 | Payment Result | Object | Kết quả thanh toán |
| 5 | Items Price | Number | Giá tổng các sản phẩm cộng lại |
| 6 | Shipping Price | Number | Phí vận chuyển |
| 7 | Tax Price | Number | Phí Tax |
| 8 | Total Price | Number | Tổng sổ tiền 3 phí trên |
| 9 | User | mongoose.  Schema.  Types.  ObjectId | Lấy ID của người dùng để lưu thông tin các đơn hàng cho tài khoản. |
| 10 | Is Paid | Boolean | Đã thanh toán hay chưa |
| 11 | Paid At | Date | Thời gian thanh toán lúc nào |
| 12 | Is Deliverd | Boolean | Đã vận chuyển hàng hay chưa |
| 13 | Deliverd At | Date | Thời gian vận chuyển lúc nào |
| 14 | Is Received | Boolean | Đã nhận được hàng hay chưa |
| 15 | Received At | Date | Thời gian nhận hàng lúc nào |
| 16 | Create At | Date | Đơn hàng được tạo khi nào |

*Bảng 4.Collection Order*

### 4.5.Collection Favorite

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Field | Type | Description |
| 1 | ProductId | String | Id sản phẩm |
| 2 | Image | String | Ảnh sản phẩm |
| 3 | Name | String | Tên sản phẩm |
| 4 | Price | Number | Giá sản phẩm |
| 5 | Category | String | Loại sản phẩm |
| 6 | User | mongoose.  Schema.  Types.  ObjectId | Lấy ID của người dùng để lưu thông tin các sản phẩm yêu thích cho tài khoản. |

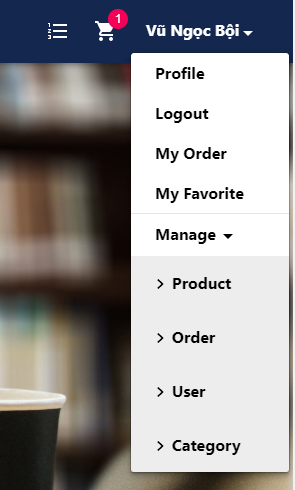
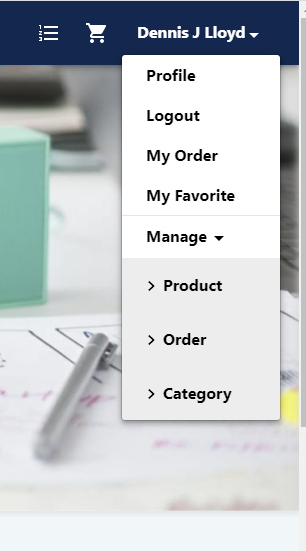
*Bảng 5.Collection Favorite*

## 5.Yêu cầu chức năng

### 5.1.Font-end

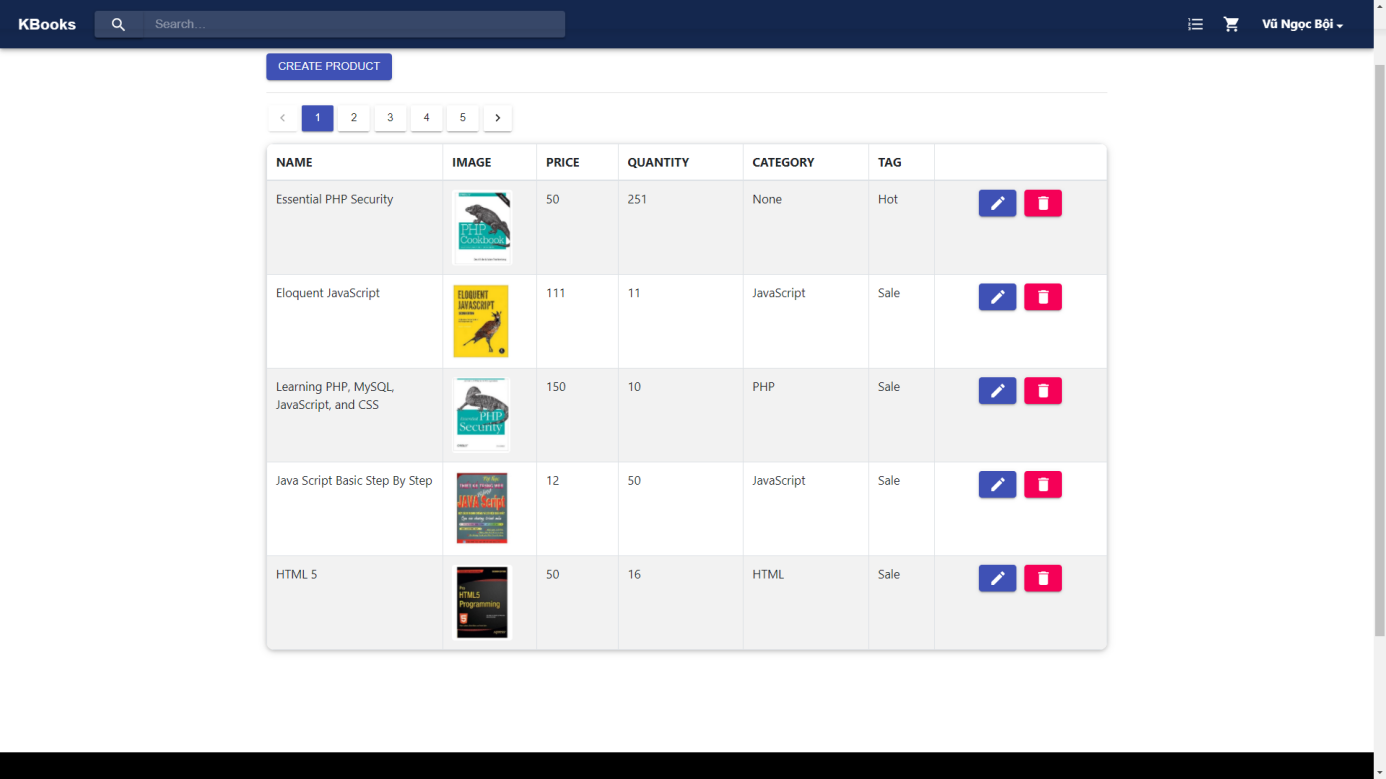
#### 5.1.1.Giao diện cho “Quản trị viên” và “Quản lí”

5.1.1.1.Menu quản lí

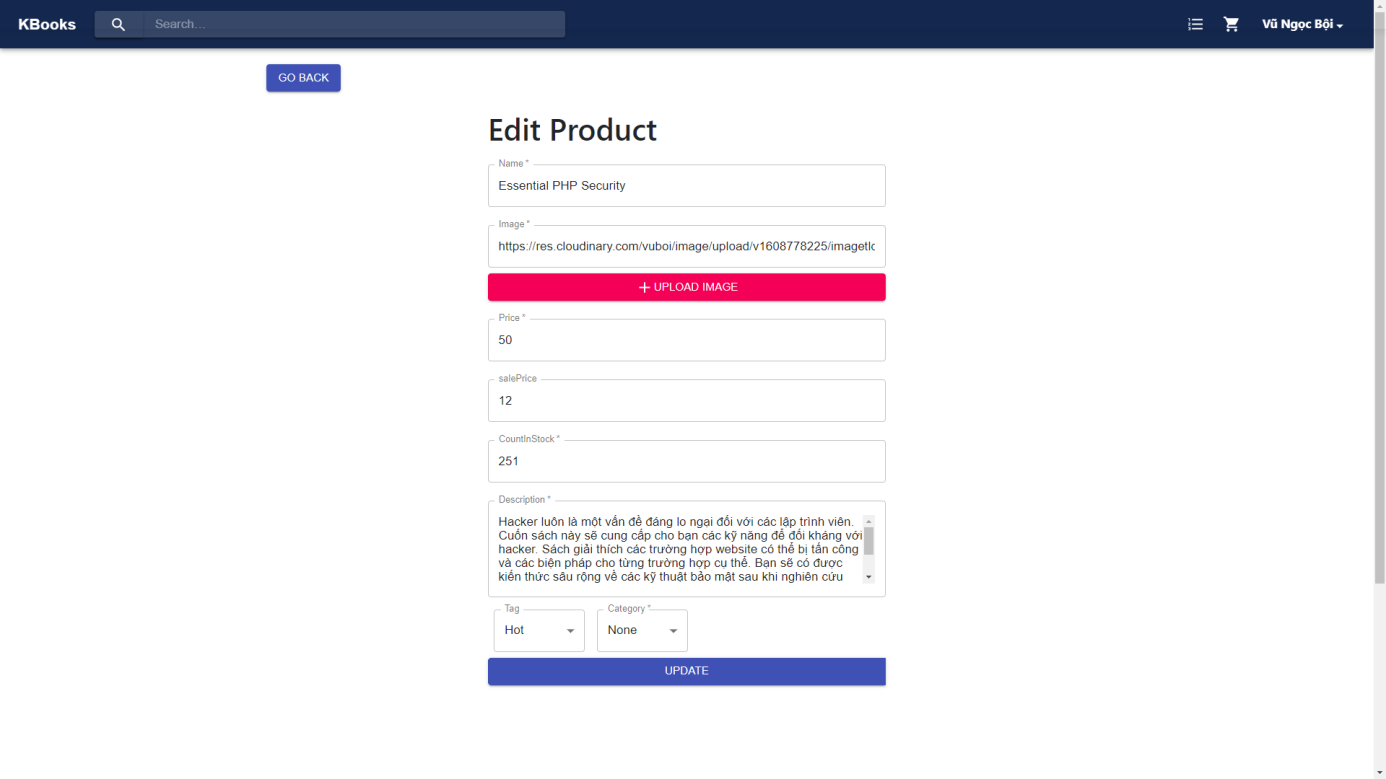


*Hình 1.Menu Quản trị viên Hình 2.Menu Quản lí*

5.1.1.2.Quản lí sản phẩm

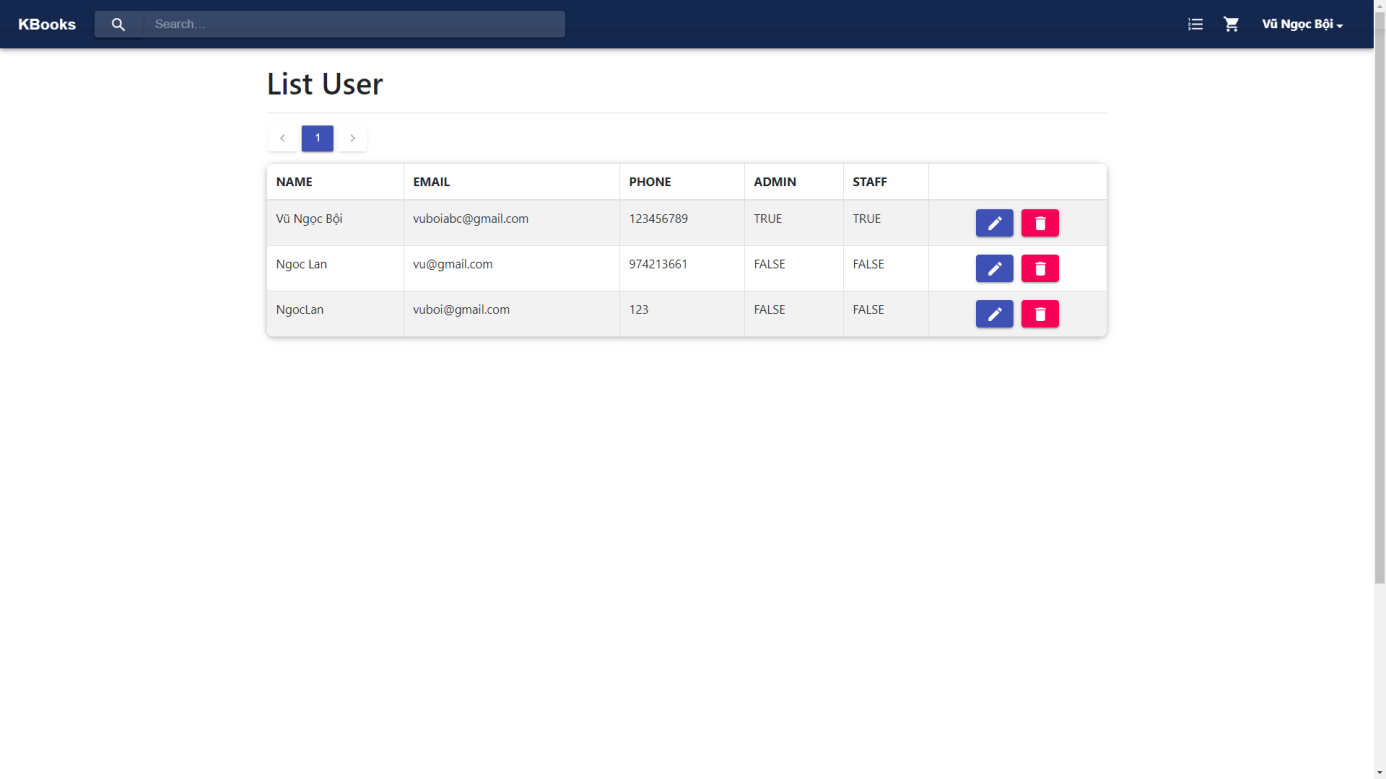


*Hình 3.Giao diện danh sách sản phẩm*

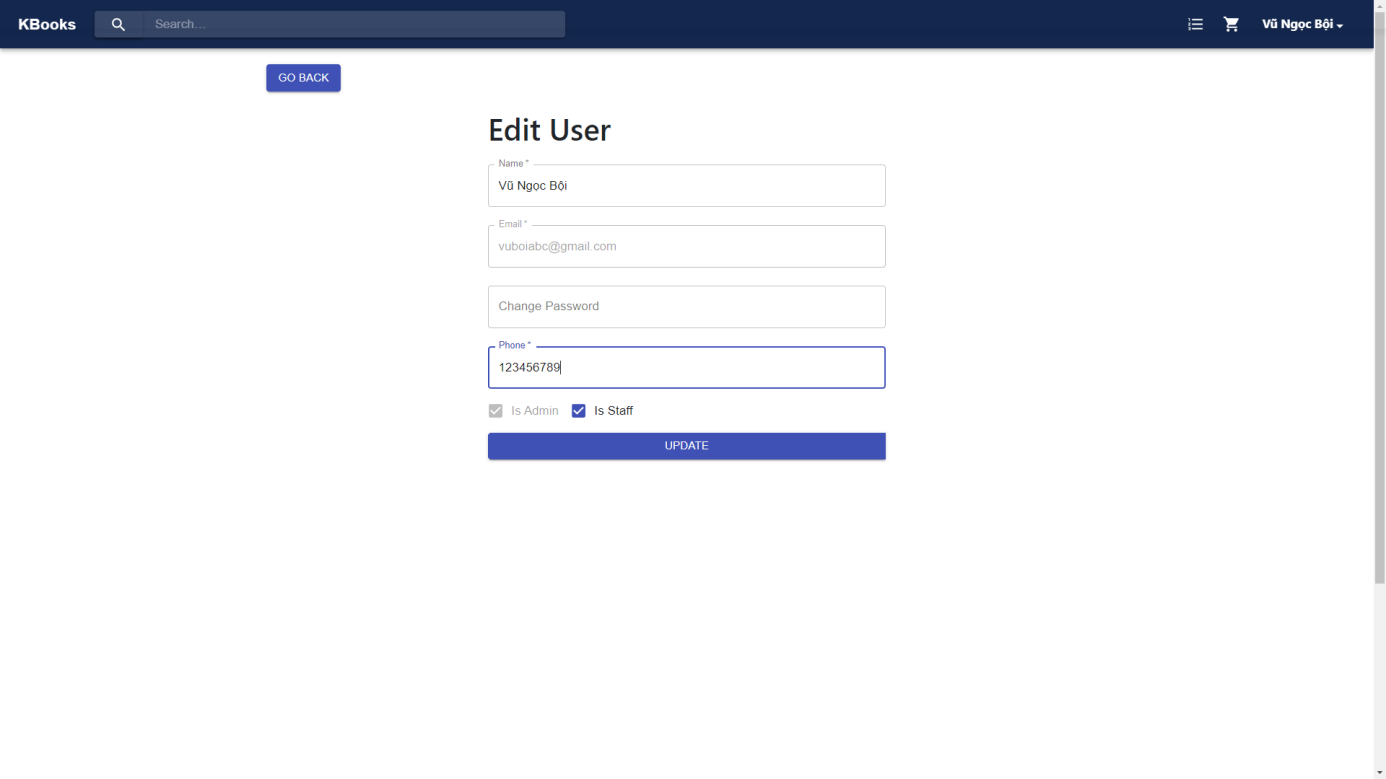


*Hình 4.Giao diện cập nhật sản phẩm (tạo sản phẩm tương tự)*

5.1.1.3.Quản lí người dùng

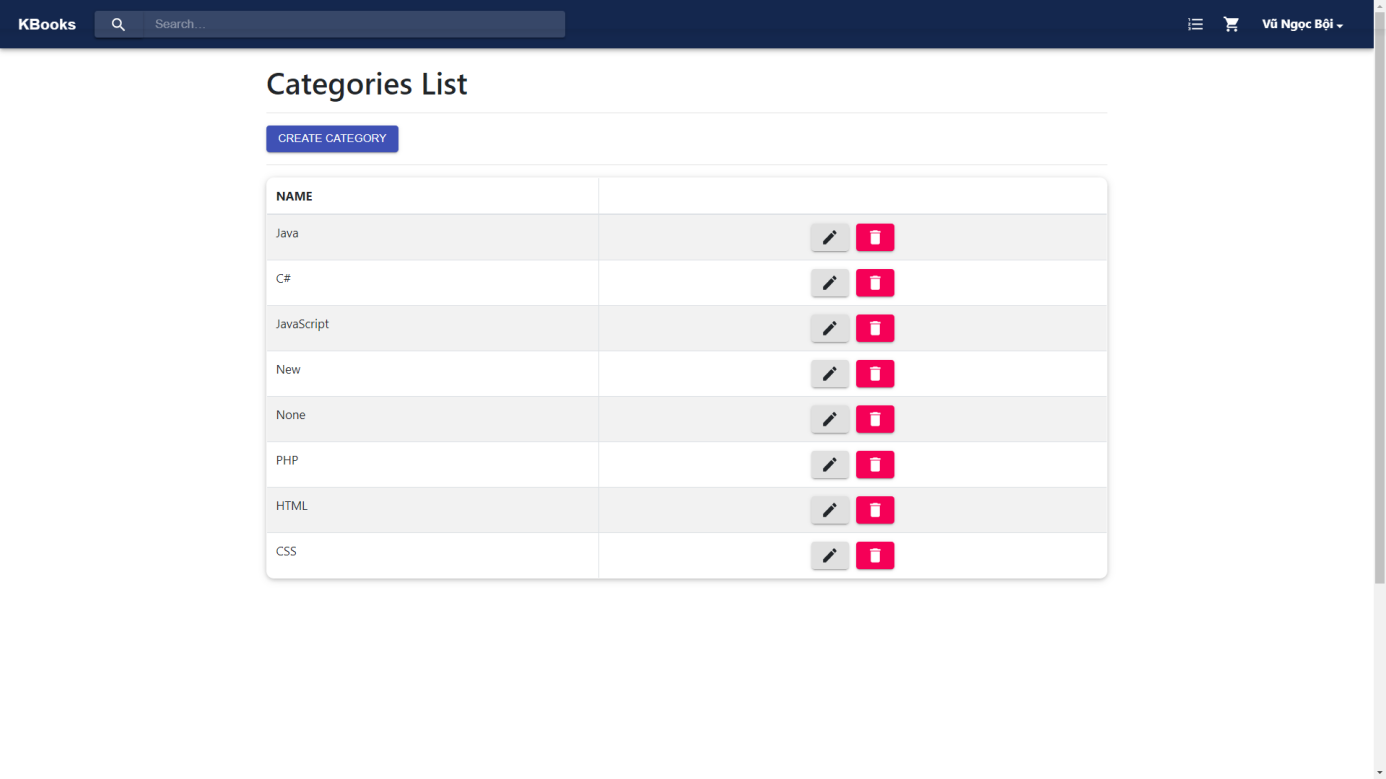


*Hình 5.Giao diện danh sách người dùng*

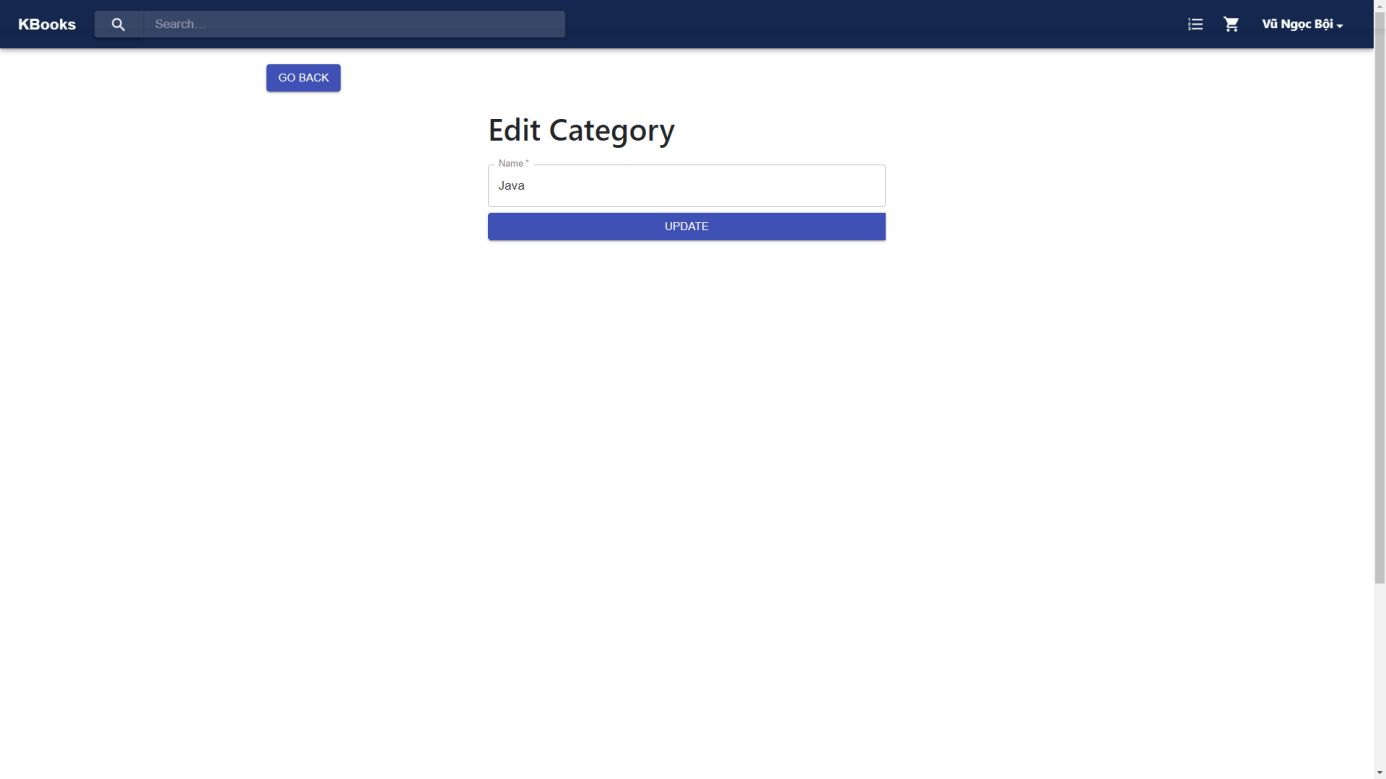


*Hình 6.Giao diện cập nhật người dùng*

5.1.1.4.Quản lí loại sản phẩm

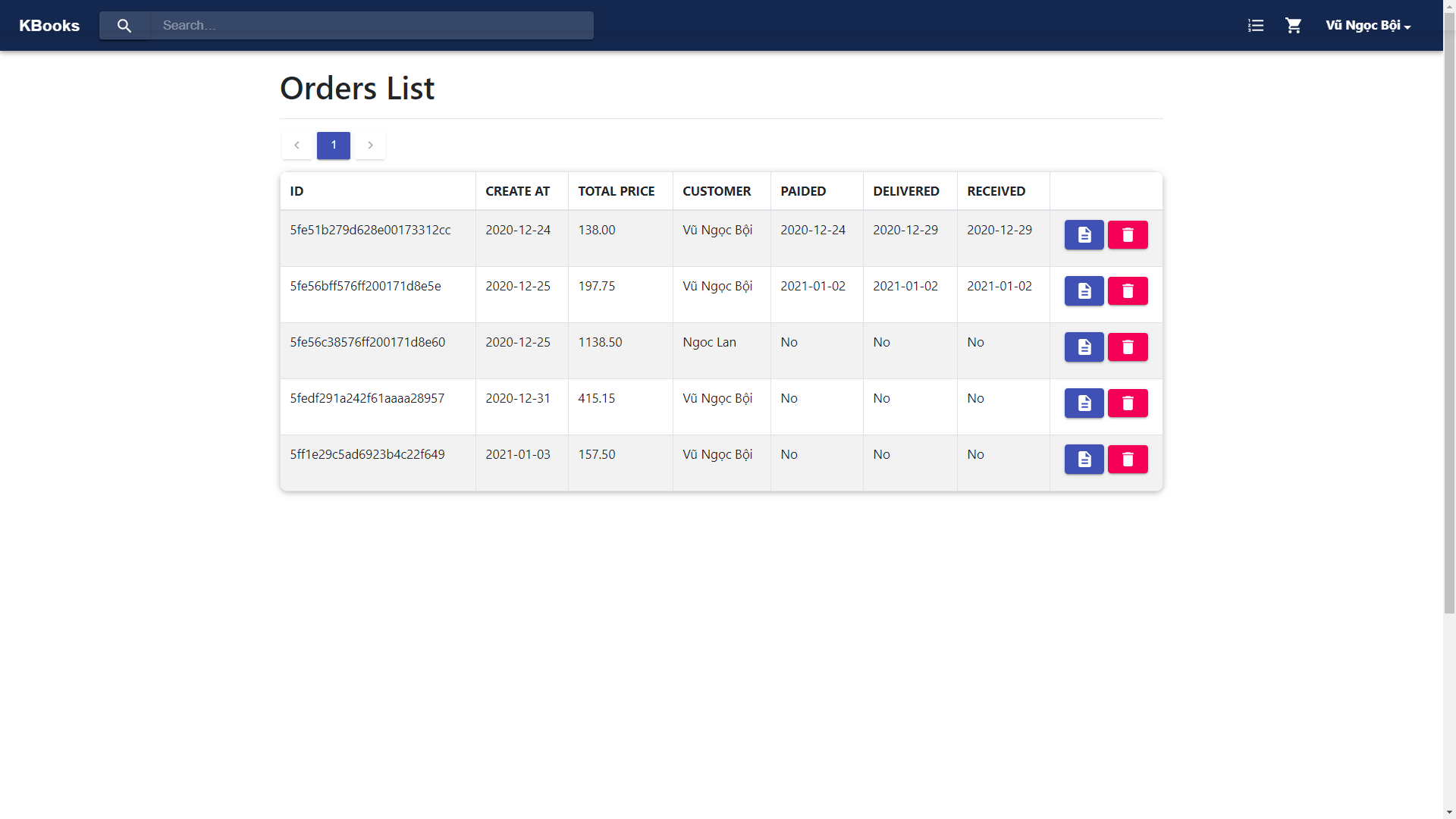


*Hình 7.Giao diện danh sách loại sản phẩm*

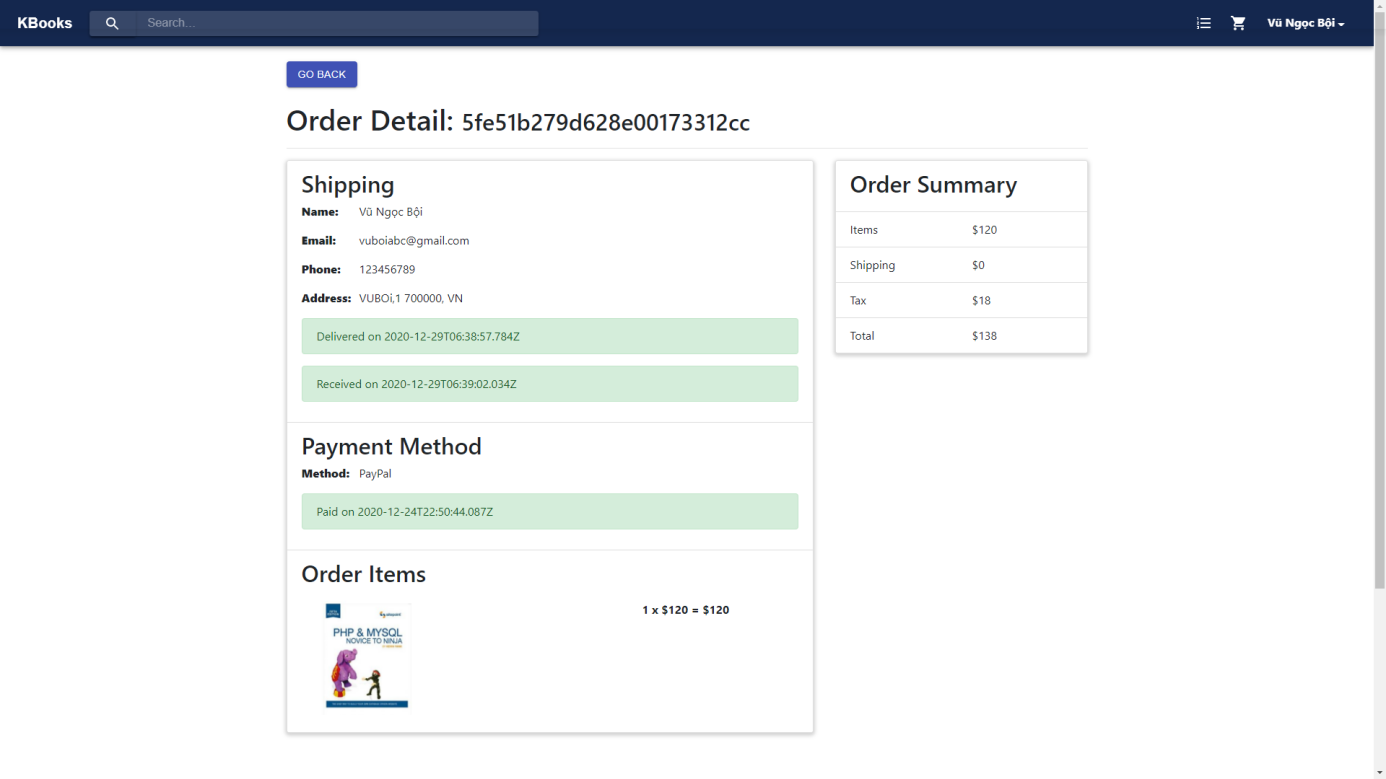


*Hình 8.Giao diện cập nhật loại sản phẩm (tạo loại sản phẩm tương tự)*

5.1.1.5.Quản lí đơn hàng



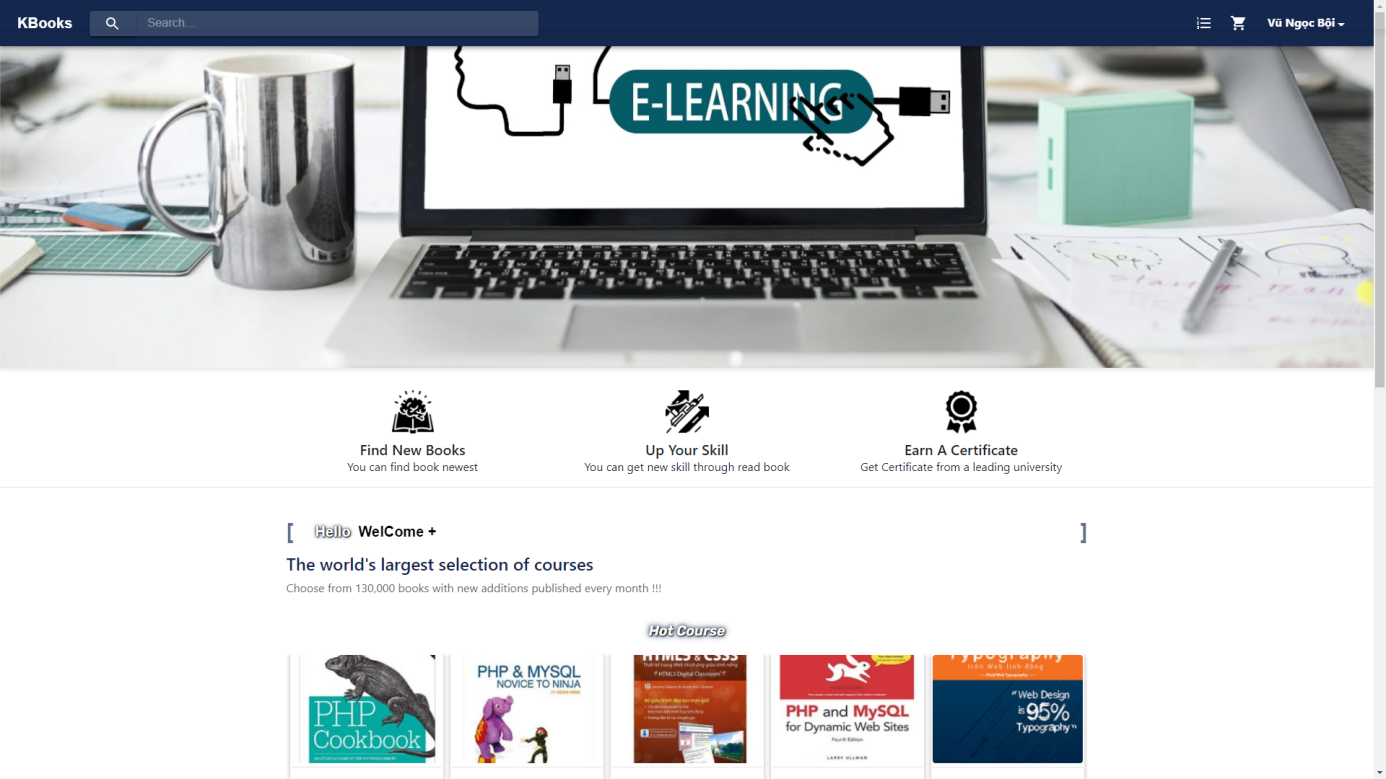
*Hình 9.Giao diện danh sách đơn đặt hàng*



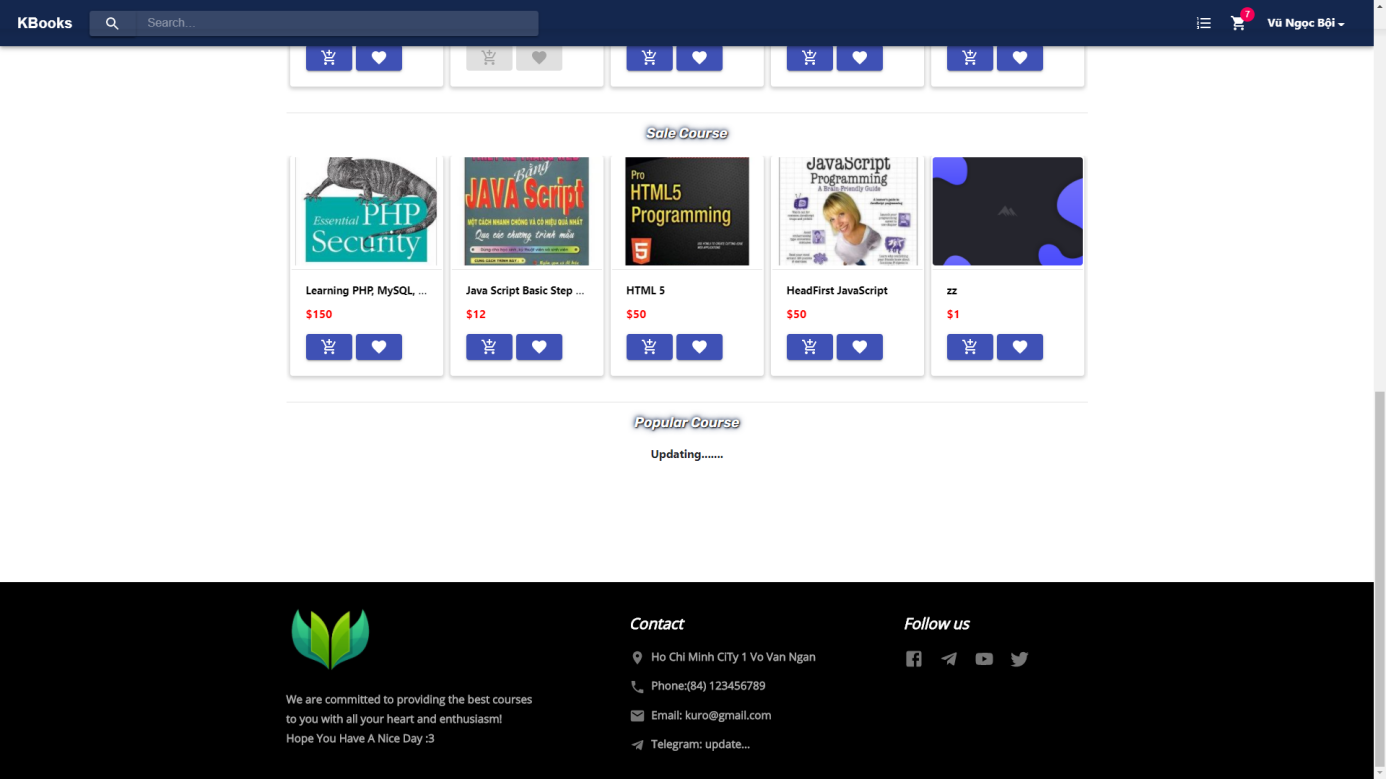
*Hình 10.Giao diện thông tin chi tiết đơn hàng*

#### 5.1.2.Giao diện cho “Khách”

5*.*1.2.1.Giao diện trang chủ

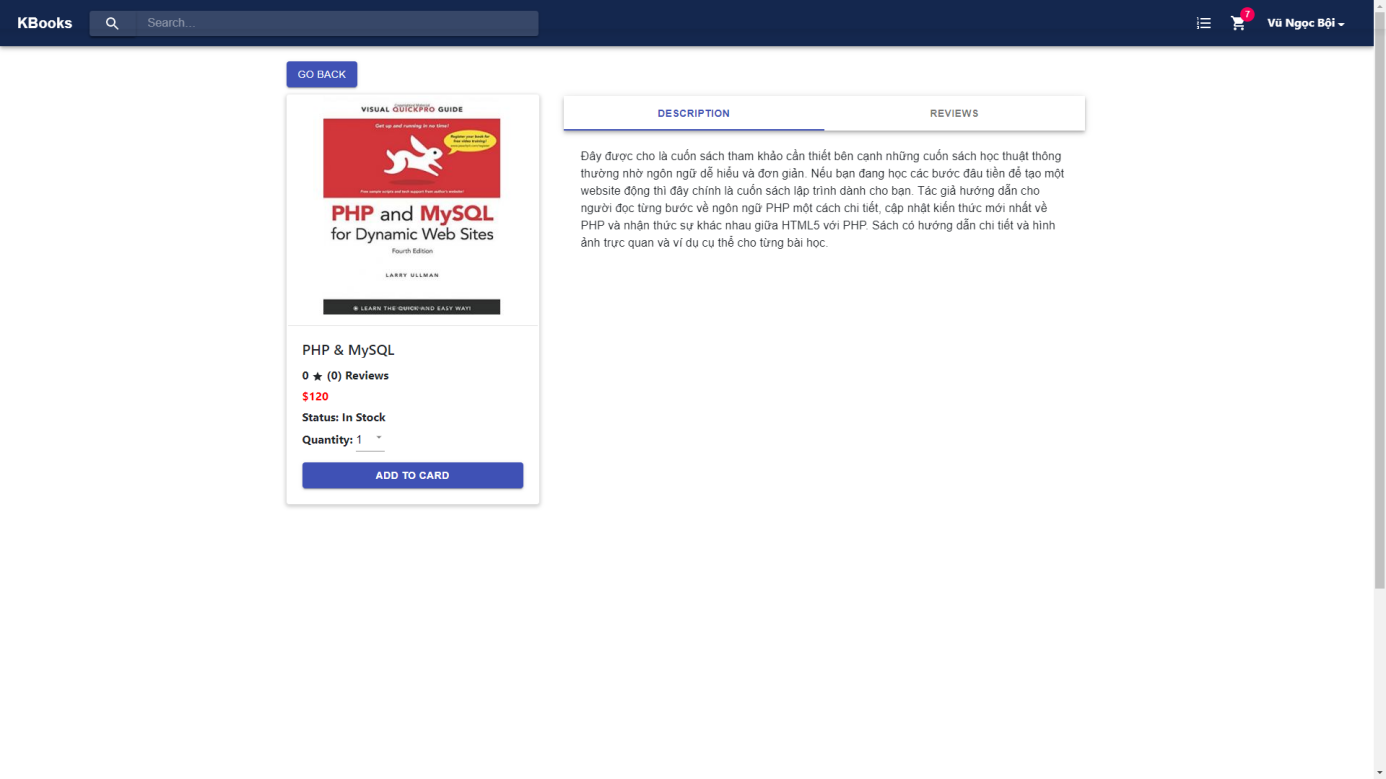


*Hình 11.Giao diện trang chủ*



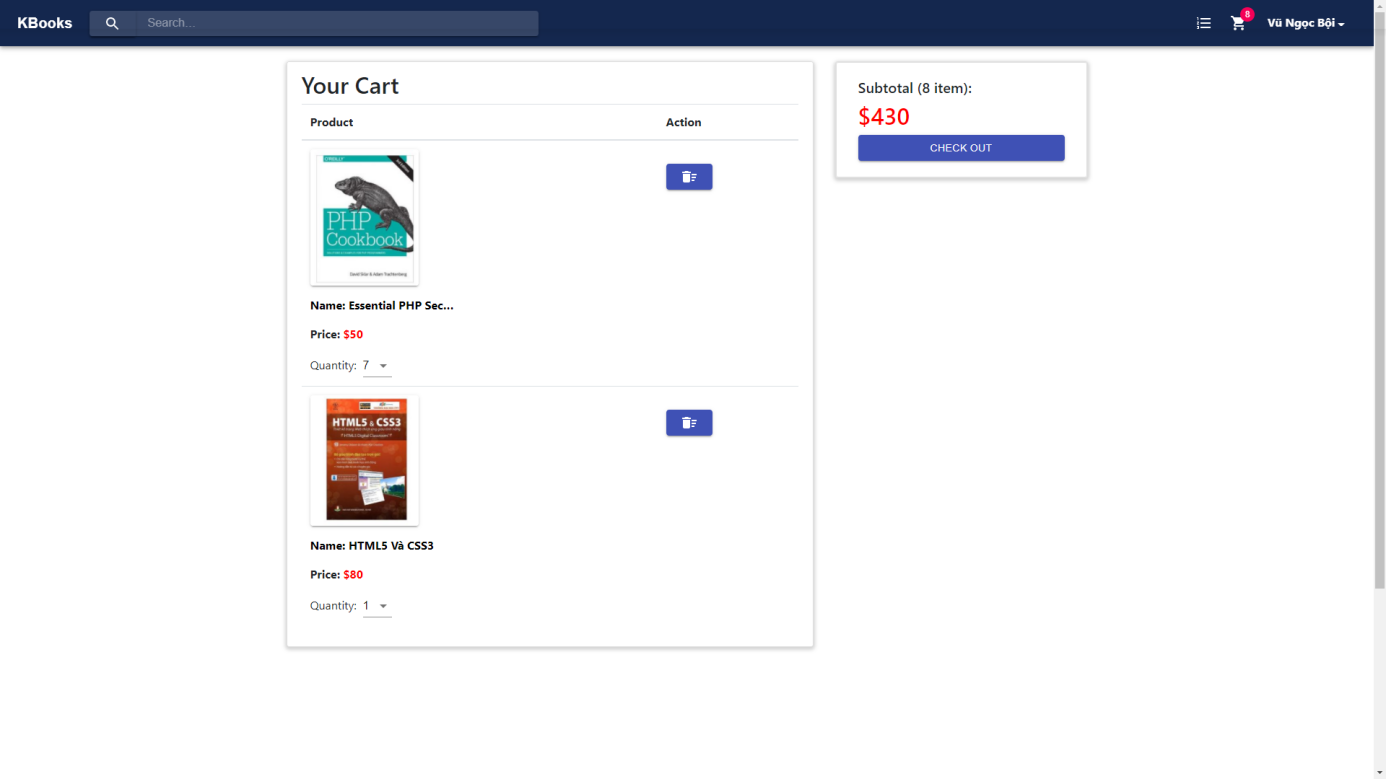
*Hình 12.Giao diện trang chủ*

5.1.2.2.Trang chi tiết sản phẩm



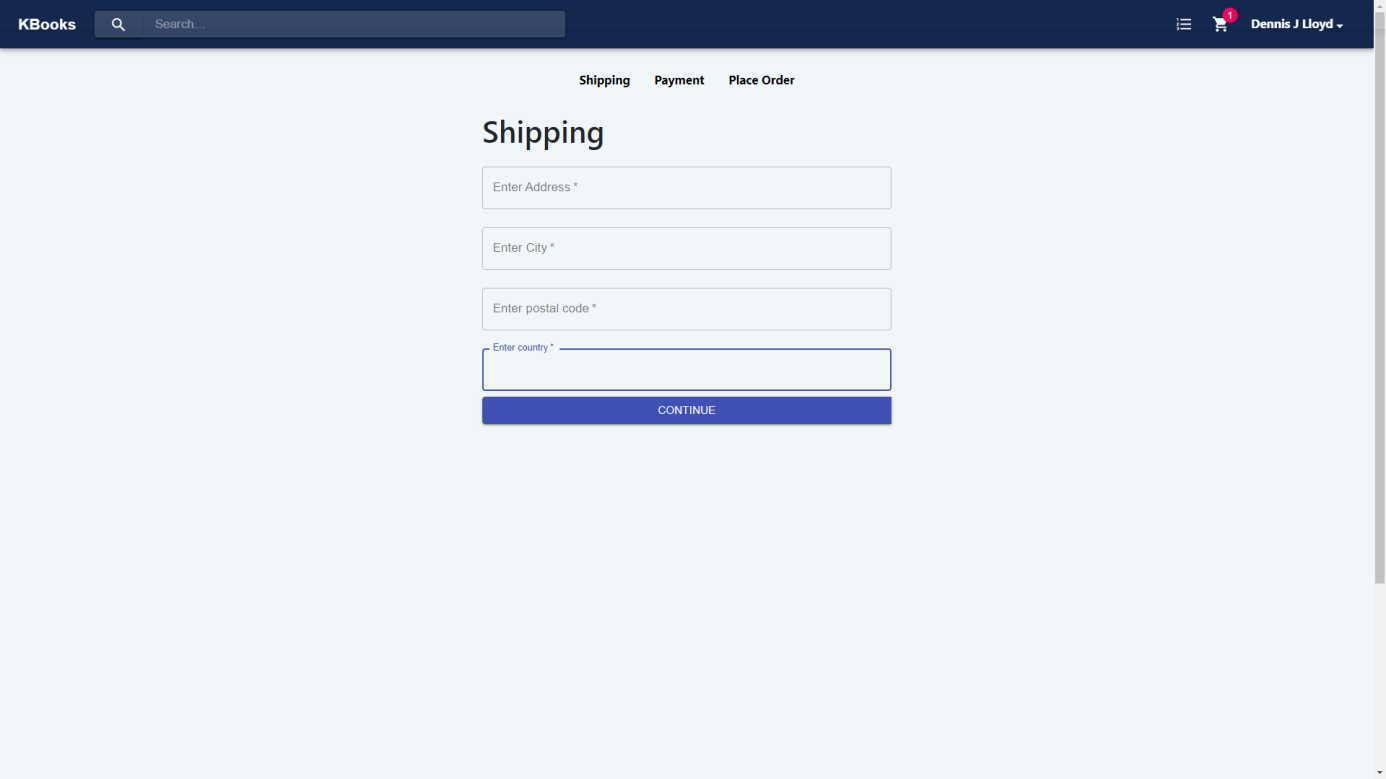
*Hình 13.Giao diện trang chi tiết sản phẩm*

5.1.2.3.Giỏ hàng

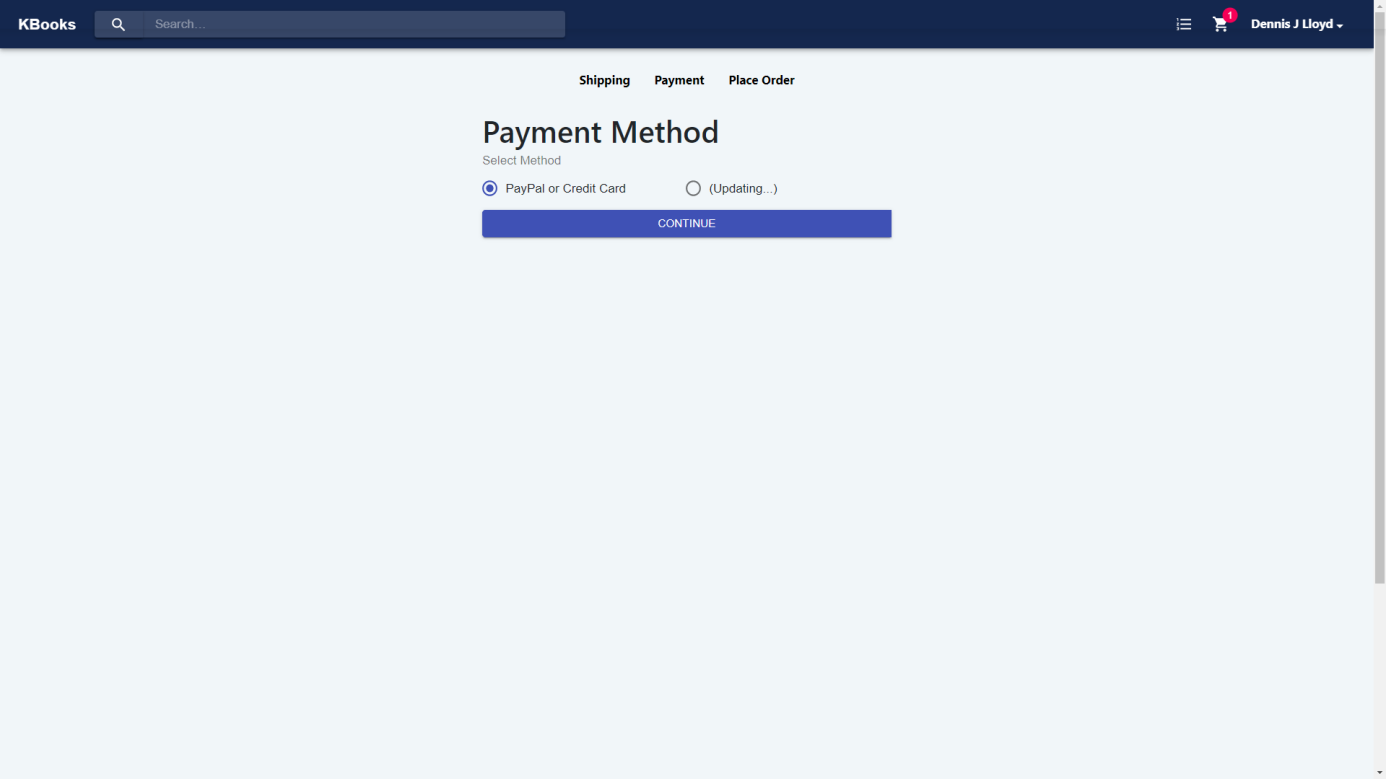


*Hình 14.Giao diện giỏ hàng*

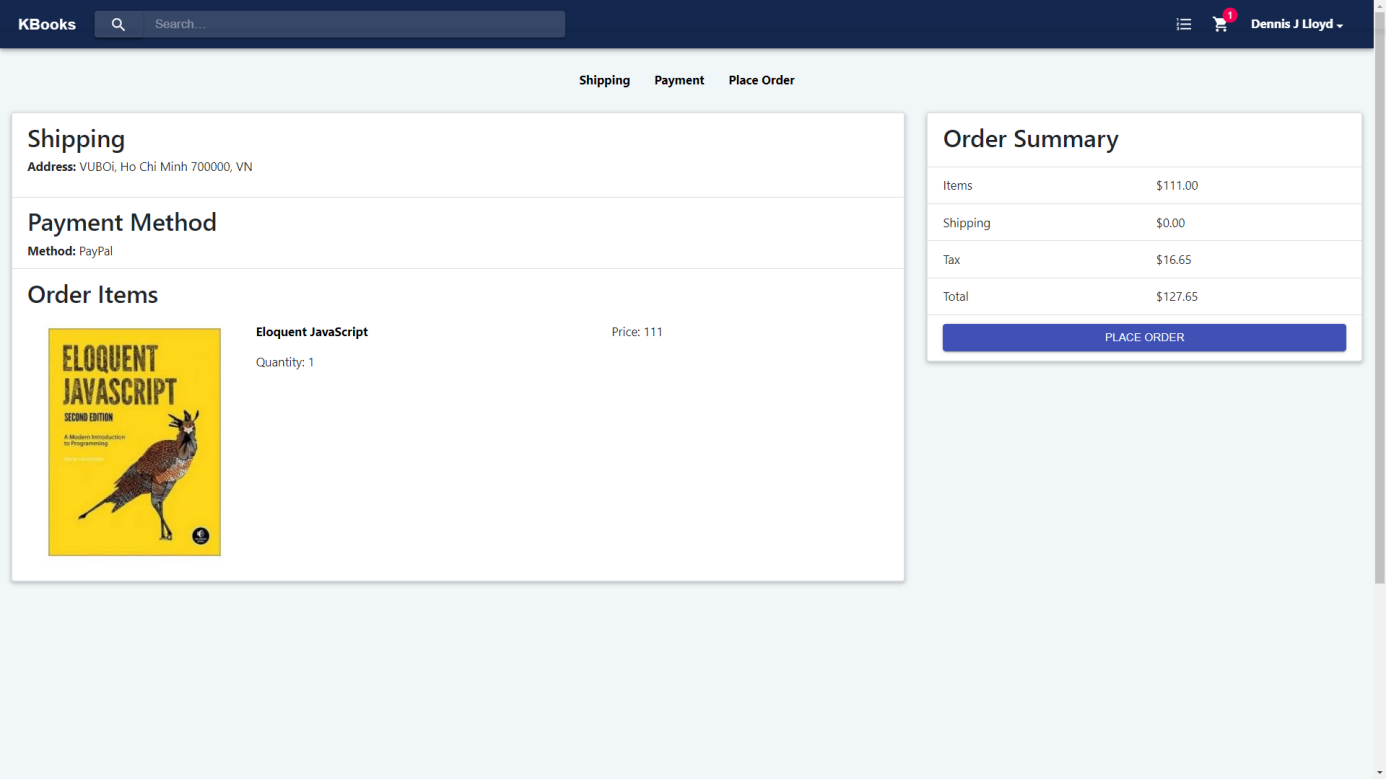
5.1.2.4.CheckoutStep



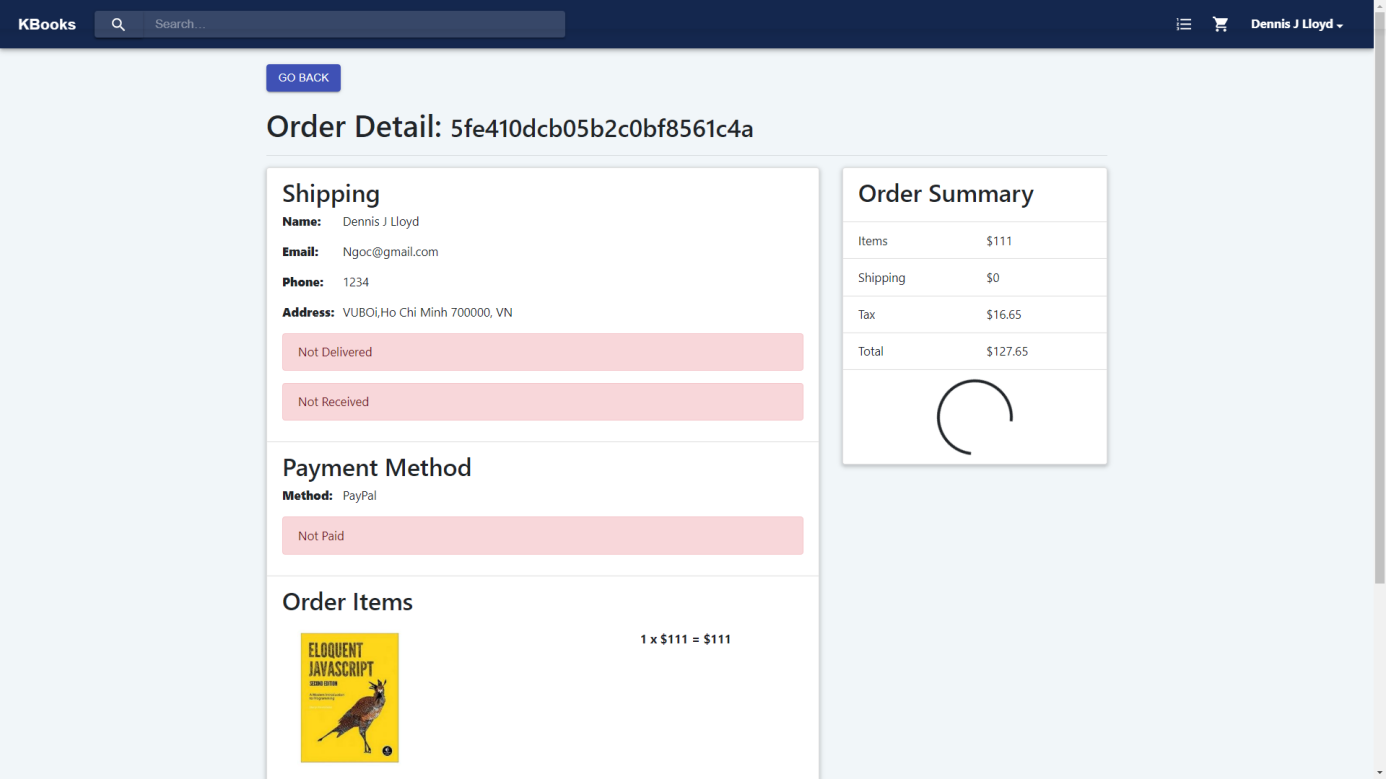
*Hình 15.Giao diện Shipping*



*Hình 16.Giao diện Payment Method*

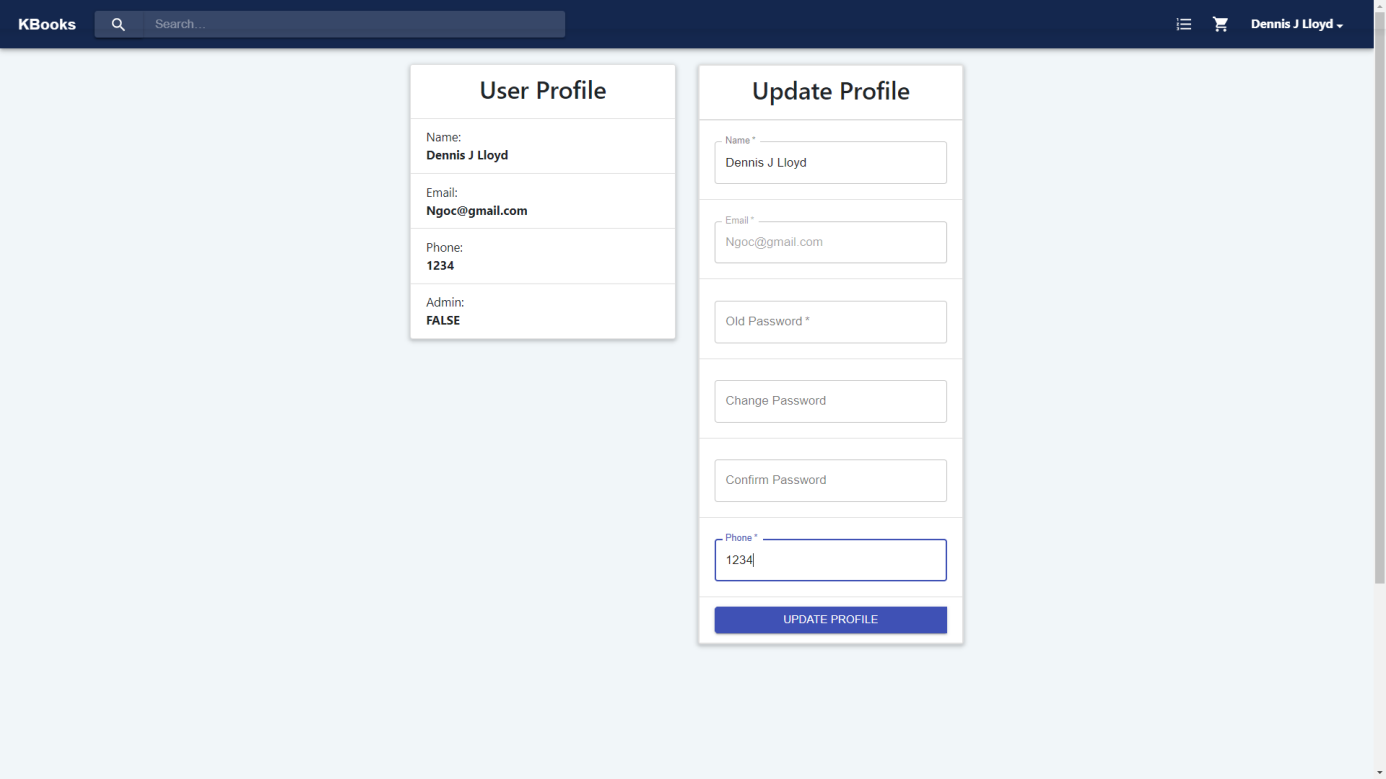


*Hình 17.Giao diện PlaceOrder*



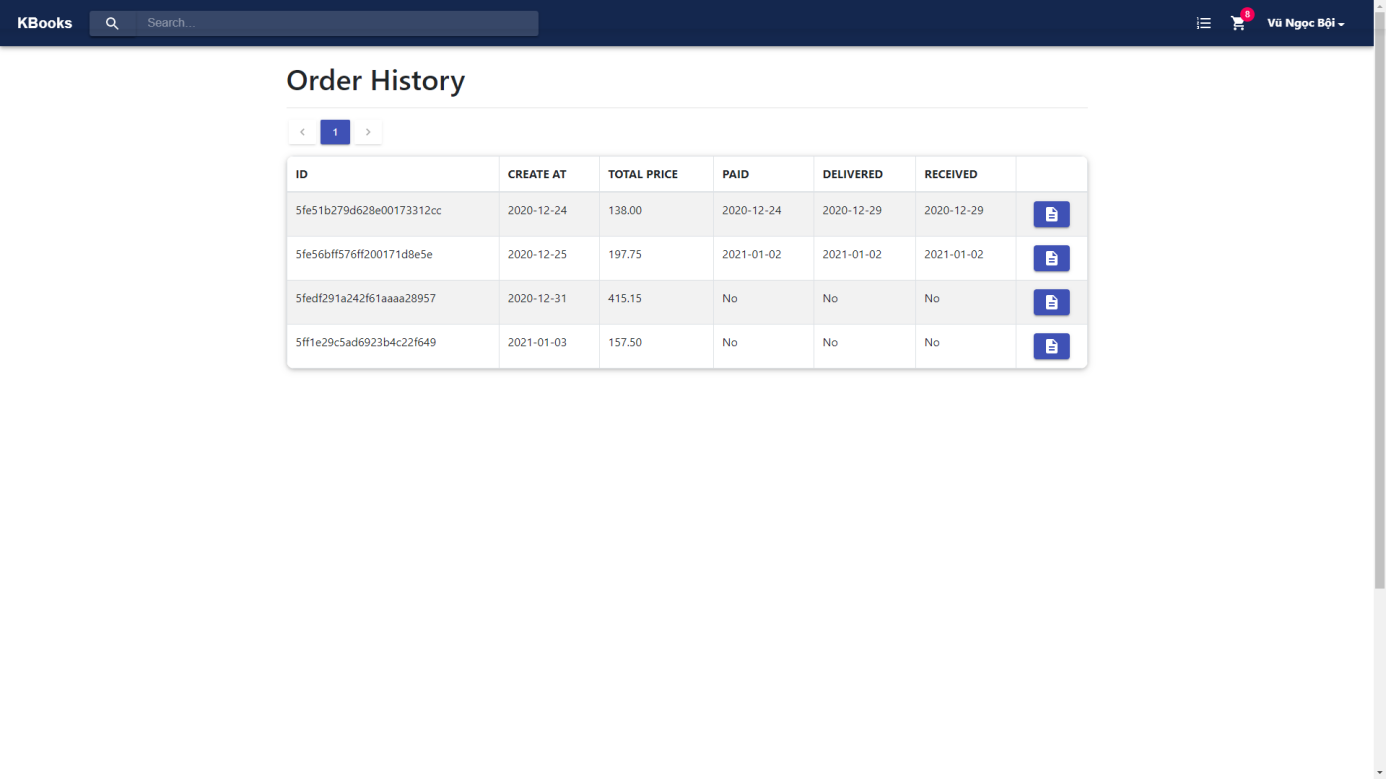
*Hình 18.Giao diện chi tiết đơn đặt hàng*

5.1.2.5.Thông tin cá nhân



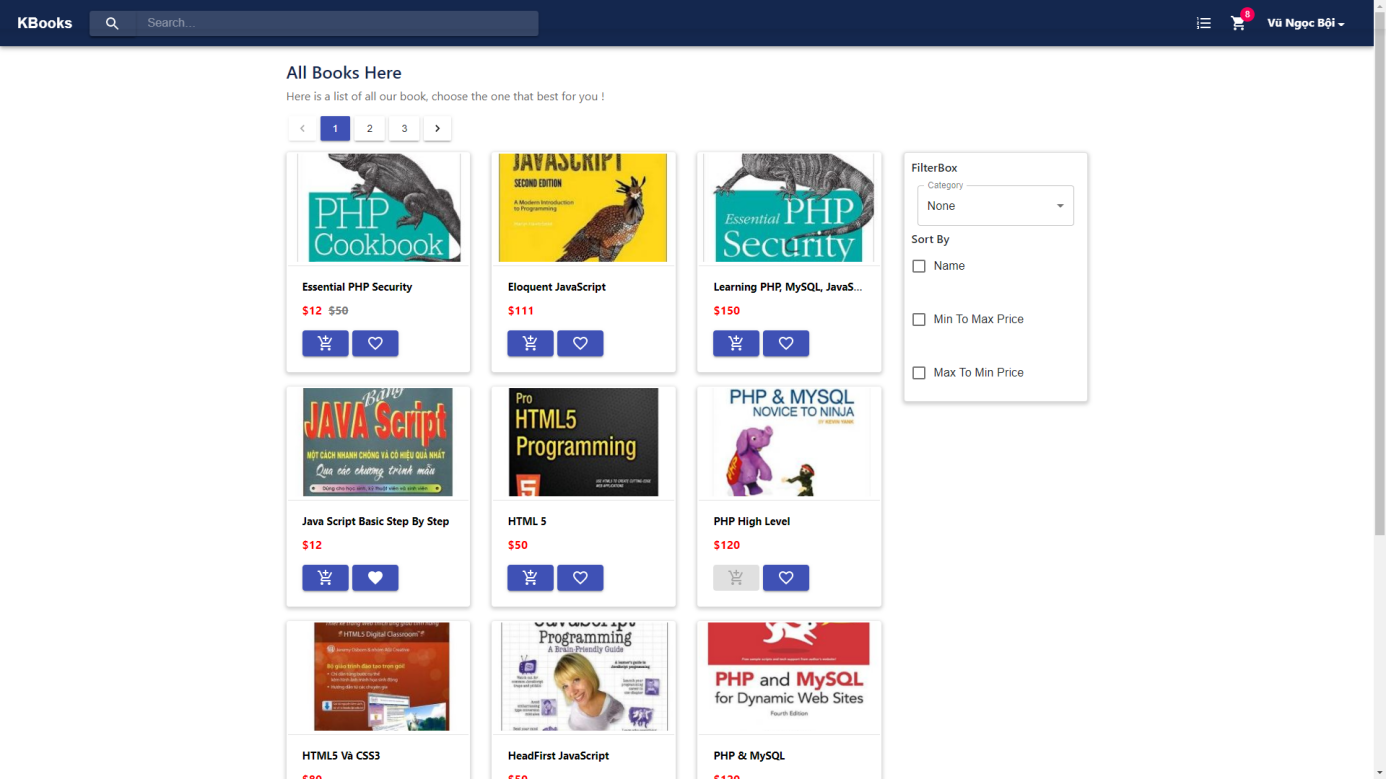
*Hình 19.Giao diện trang cập nhật thông tin cá nhân*

5.1.2.6.Danh sách đơn đặt hàng cá nhân



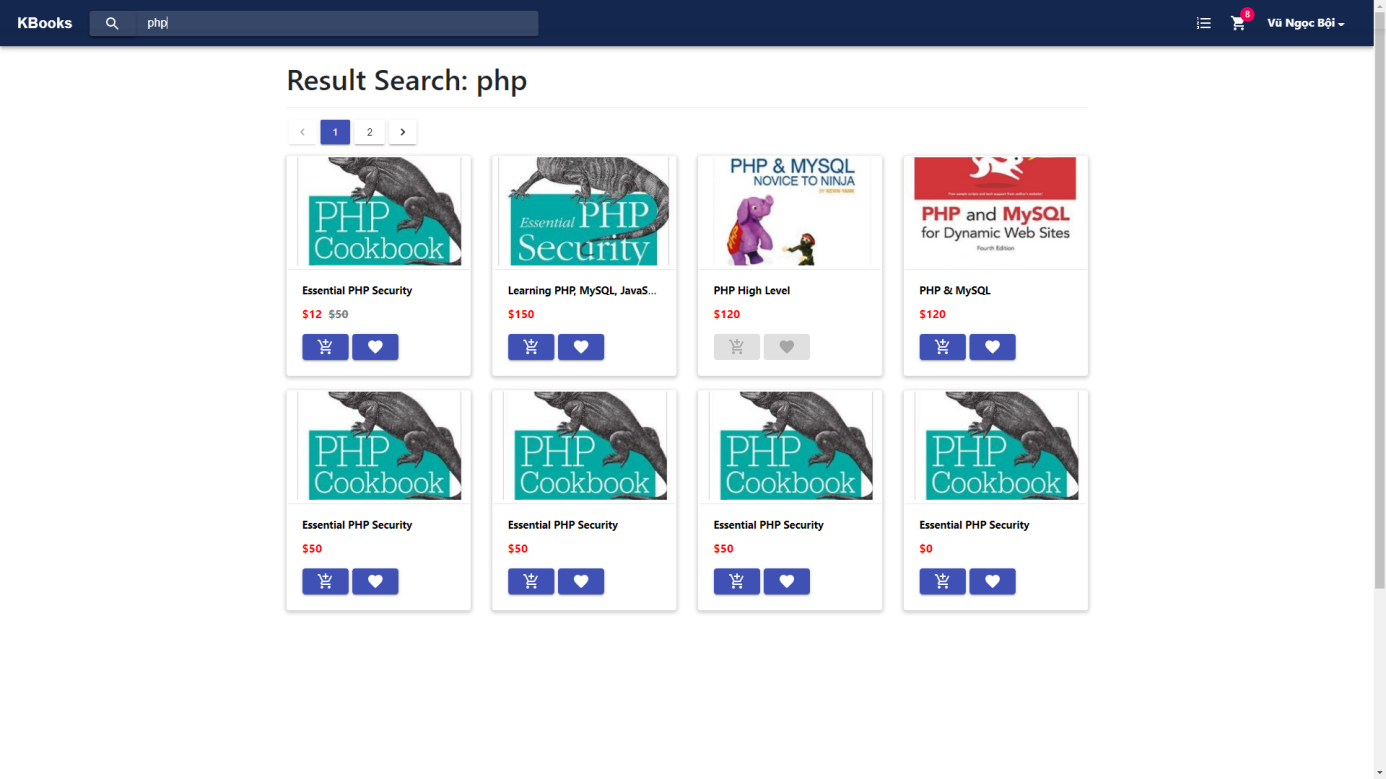
*Hình 20.Giao diện danh sách đơn hàng cá nhân*

5.1.2.7.Danh sách sản phẩm



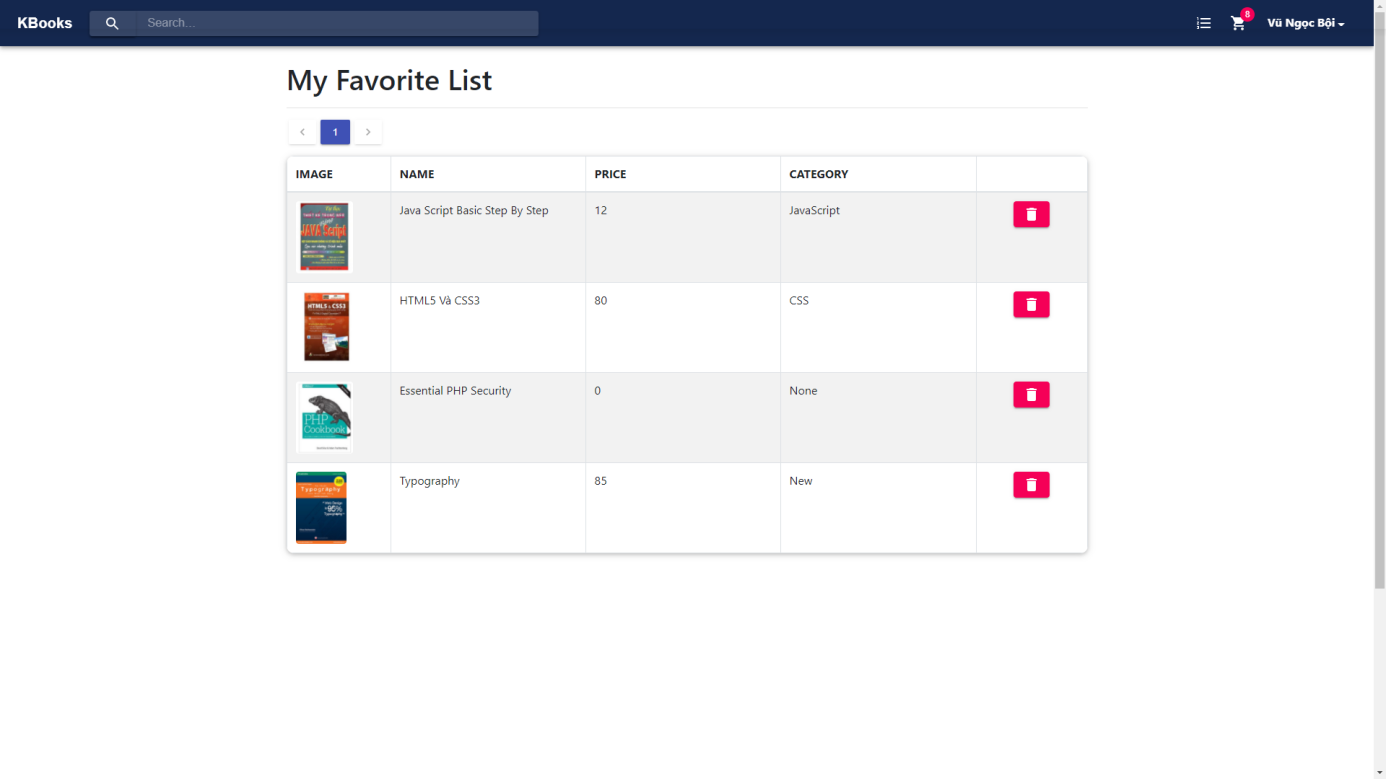
*Hình 21.Giao diện trang toàn bộ sản phẩm*

5.1.2.8.Trang tìm kiếm



*Hình 22.Giao diện trang tìm kiếm*

5.1.2.9.Trang sản phẩm yêu thích



*Hình 23.Giao diện trang sản phẩm yêu thích*

### 5.2.Back-end

#### 5.2.1.User Endpoint

*app.use('/api/users', userRoute);*

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Method | Path | Description |
| 1 | GET | '/' | Lấy danh sách các tài khoản user |
| 2 | GET | '/:id' | Lấy thông tin một user dựa vào tham số id |
| 3 | GET | '/create/admin' | Tạo tài khoản admin mặc định |
| 4 | POST | '/signin' | Gửi thông tin lên server để đăng nhập |
| 5 | POST | '/signup' | Gửi thông tin lên server để tạo tài khoản mới |
| 6 | PUT | '/:id' | Chỉnh sửa thông tin tài khoản trong chức năng quản lí tài khoản (admin) |
| 7 | PUT | '/profile' | Cập nhật thông tin tài khoản dành cho người dùng |
| 8 | DELETE | '/:id' | Xoá tài khoản trong chức năng quản lí tài khoản (admin) thông qua tham số id |

#### 5.2.2.Product Endpoint

*app.use('/api/products', productRoute);*

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Method | Path | Description |
| 1 | GET | '/' | Lấy danh sách các sản phẩm |
| 2 | GET | '/:id' | Lấy thông tin một sản phẩm dựa vào tham số id |
| 3 | POST | '/' | Gửi các thông tin cần thiết lên server để tạo sản phẩm mới trong chức năng quản lí sản phẩm (admin or staff) |
| 4 | PUT | '/:id' | Chỉnh sửa thông tin sản phẩm trong chức năng quản lí sản phẩm (admin or staff) thông qua tham số id |
| 5 | DELETE | '/:id' | Xoá sản phẩm trong chức năng quản lí sản phẩm (admin or staff) thông qua tham số id |

#### 5.2.3.Order Endpoint

*app.use('/api/orders', orderRoute);*

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Method | Path | Description |
| 1 | GET | '/' | Lấy danh sách các đơn đặt hàng cho chức năng quản lí đơn hàng (admin) |
| 2 | GET | '/myorders' | Lấy danh sách các đơn đặt hàng dành cho khách hàng |
| 3 | GET | '/:id' | Lấy thông tin chi tiết đơn hàng thông qua tham số id |
| 4 | POST | '/' | Gửi các thông tin cần thiết lên server để tạo đơn đặt hàng mới |
| 5 | PUT | '/:id/pay' | Cập nhật thông tin đơn hàng thanh toán hay chưa thanh toán thông qua tham số id |
| 6 | PUT | '/:id/deliver' | Cập nhật thông tin đơn đã di chuyển hay chưa, khách hàng phải thanh toán thì admin mới cập nhật được |
| 7 | PUT | '/:id/receive' | Cập nhật thông tin đơn hàng đã nhận được hay chưa, khách hàng phải thanh toán và admin đã cập nhật đơn hàng di chuyển thì mới được cập nhật |
| 8 | DELETE | '/:id' | Xoá đơn hàng trong quản lí đơn hàng (admin or staff) thông qua tham số id |

#### 5.2.4.Category Endpoint

*app.use('/api/categories', categoryRoute);*

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Method | Path | Description |
| 1 | GET | '/' | Lấy danh sách các loại sản phẩm |
| 2 | GET | '/:id' | Lấy thông tin một loại sản phẩm thông qua tham số id |
| 3 | POST | '/' | Gửi các thông tin cần thiết lên server để tạo loại sản phẩm mới trong chức năng quản lí loại sản phẩm (admin or staff) |
| 4 | PUT | '/:id' | Chỉnh sửa thông tin loại sản phẩm trong chức năng quản lí loại sản phẩm (admin or staff) thông qua tham số id |
| 5 | DELETE | '/:id' | Xoá loại sản phẩm trong chức năng quản lí loại sản phẩm (admin or staff) thông qua tham số id |

#### 5.2.5.Favorie Endpoint

*app.use('/api/favorites/', favoriteRoute);*

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Method | Path | Description |
| 1 | GET | '/ myfavorites ' | Lấy danh sách các sản phẩm yêu thích |
| 2 | POST | '/' | Gửi các thông tin cần thiết lên server để thêm sản phẩm yêu thích mới, dành cho người dùng |
| 3 | DELETE | '/:id' | Xoá sản phẩm yêu thích khỏi danh sách sản phẩm yêu thích thông qua tham số id, dành cho người dùng ở giao diện quản lí sản phẩm yêu thích |
| 4 | DELETE | '/ productId/:id' | Xoá sản phẩm yêu thích khỏi danh sách sản phẩm yêu thích thông qua tham số id, dành cho người dùng ở giao diện button click |

#### 5.2.6.Paypal Endpoint

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Method | Path | Description |
| 1 | GET | '/api/config/paypal' | Lấy thông tin Paypal client id |

#### 5.2.7.Search Endpoint

*app.use('/api/search', searchRoute);*

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Method | Path | Description |
| 1 | GET | '/products' | Lấy danh sách các sản phẩm tìm kiếm thông qua từ khoá |

## 6.Thiết kế xử lí đơn giản

### 6.1.Font-end

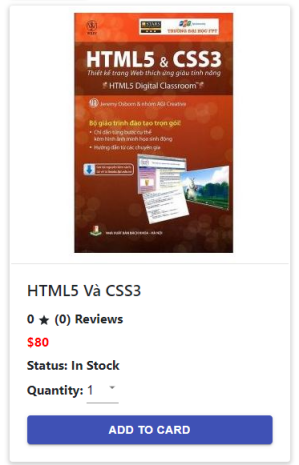
#### 6.1.1.Liệt kê các Module

|  |  |  |
| --- | --- | --- |
| Module | Path | Description |
| NotFound | '../pages/NotFound'; | Hiển thị khi truy cập đường dẫn không tồn tại |
| Home | '../pages/Home'; | Trang chủ |
| ProductList | '../pages/ProductList'; | Hiển thị trang danh sách tất cả sản phẩm |
| ProductDetail | '../pages/ProductDetail'; | Hiển thị trang thông tin chi tiết sản phẩm |
| Cart | '../pages/Cart'; | Hiển thị trang giỏ hàng |
| Signin | '../pages/Signin'; | Hiển thị trang đăng nhập |
| Signup | ../pages/Signup'; | Hiển thị trăng đăng kí |
| Shipping | '../pages/Shipping'; | Hiển thị trang thông tin vận chuyển |
| Payment | '../pages/Payment'; | Hiển thị trang phương thức thanh toán |
| PlaceOrder | '../pages/PlaceOrder'; | Hiển thị trang tiến hành đặt hàng |
| Profile | '../pages/Profile'; | Hiển thị trang thông tin cá nhân và cập nhật thông tin cho người dùng |
| Signout | '../pages/Signout'; | Điếu hướng tới trang đăng xuất để đăng xuất tài khoản |
| ProductEdit | '../pages/ProductEdit'; | Hiển thị trang chỉnh sửa sản phẩm cho admin |
| ProductListManage | '../pages//ProductListManage'; | Hiển thị trang quản lí danh sách sản phẩm cho admin |
| ProductCreate | '../pages/ProductCreate'; | Hiện thị trang tạo sản phẩm mới cho admin |
| Order | '../pages/Order'; | Hiển thị trang thông tin đơn hàng |
| OrderList | '../pages/OrderList'; | Hiển thị trang quản lí đơn đặt hàng cho admin |
| OrderHistoryList | '../pages/OrderHistoryList'; | Hiển thị trang quản lí đơn hàng cho người dùng |
| UserList | '../pages/UserList'; | Hiển thị trang quản lí tài khoản người dùng cho admin |
| UserEdit | '../pages/UserEdit'; | Hiển thị trang chỉnh sửa tài khoản người dùng cho admin |
| Search | '../pages/Search'; | Hiển thị trang tìm kiếm sản phẩm |
| CategoryListManage | '../pages/CategoryListManage'; | Hiển thị trang quản lí danh sách loại sản phẩm cho admin |
| CategoryEdit | '../pages/CategoryEdit'; | Hiển thị trang chỉnh sửa loại sản phẩm cho admin |
| CategoryCreate | '../pages/CategoryCreate'; | Hiển thị trang tạo loại sản phẩm mới cho admin |
| Favorite | '../pages/Favorite'; | Hiển thị trang danh sách sản phẩm yêu thích cho người dùng |

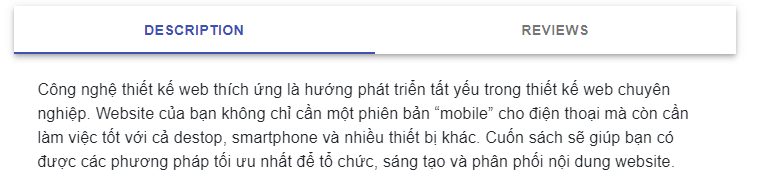
#### 6.1.2.Thiết kế ứng dụng

6.1.2.1.Các Component

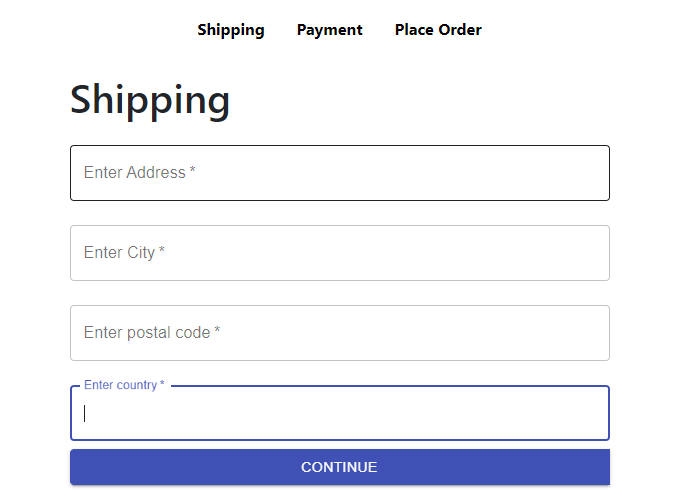
* CardProduct: chứa các thẻ card hiển thị thông tin sản phẩm
* CardProductDetail: thông tin sản phẩm trang thông tin chi tiết sản phẩm
* CardProductList: thông tin sản phẩm trang danh sách sản phẩm
* CardProductSearch: thông tin sản phẩm trang tìm kiếm sản phẩm



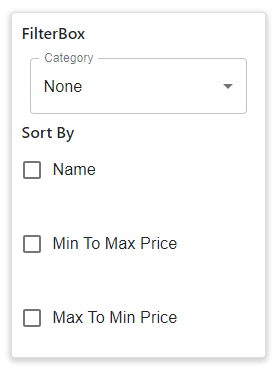
* InfoProduct: Thẻ tab Description & Review



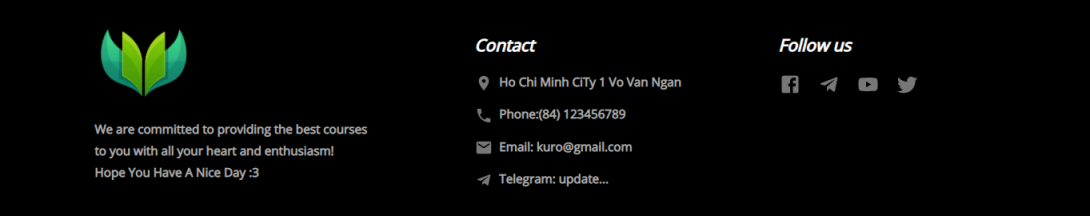
* CheckoutStep: Các bước điền thông tin trước khi đặt hàng



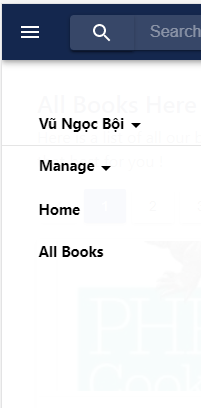
* FormContainer: Khung để giới hạn giao diện một số trang
* FilterBox: Hộp lọc sản phẩm



* Footer: Chân trang website



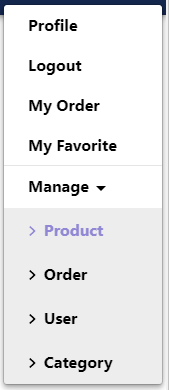
* Header:
* Drawer: Thanh menu bên trái cho giao diện mobile:



* Header: Phần đầu trang webiste



* MenuUserDesktop: Menu bên phải phần user



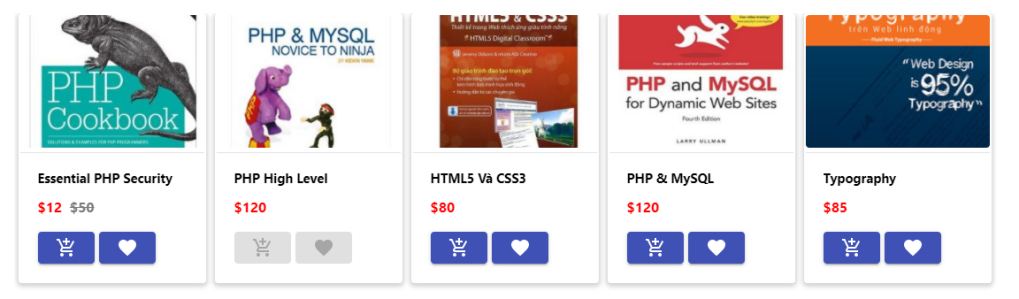
* Home:
* FeaturesBox: Các Featuresbox ở trang chủ



* SearchBox: Hộp tìm kiếm



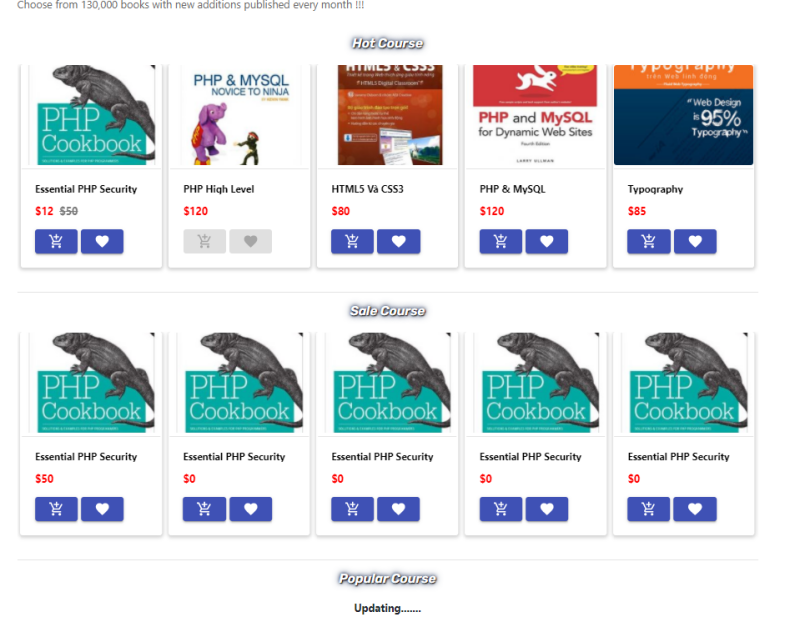
* SliderCourse: Slider sản phẩm



* SliderHome: Slider phần trên cùng header



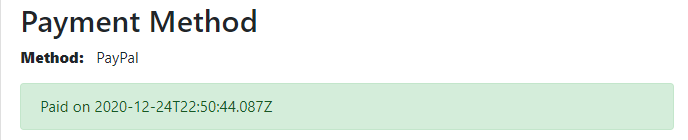
* TopCourse: Các sản phẩm có gán tag



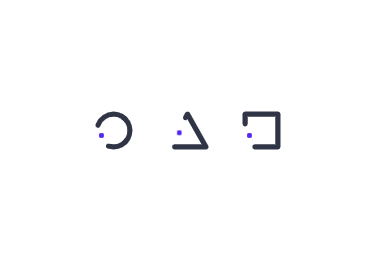
* Loading: Các dạng loading… trước khi dữ liệu chưa lấy về kịp.
* DefaultLoading



* Loader:



* Loading:



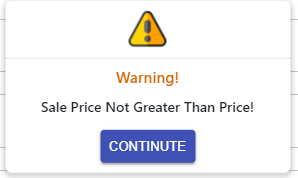
* Message: Các tin nhắn thông báo sau các action như sửa xoá, thêm
* ErrorMessage:



* SuccessMessage:



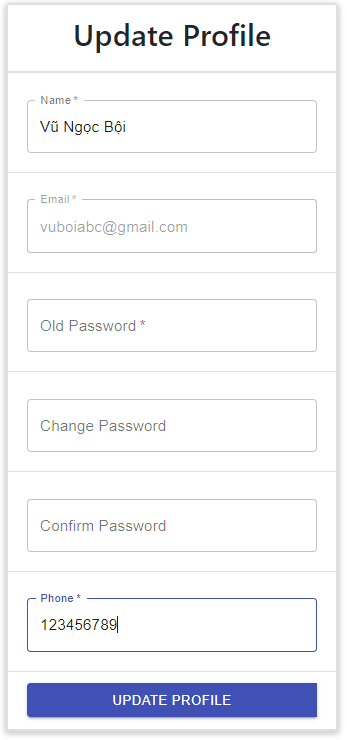
* WarningMessage:



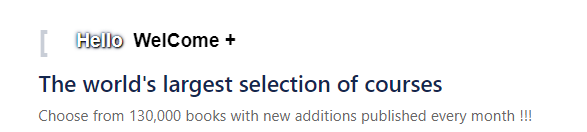
* Pagination: Phân trang



* ProfileEdit: Chỉnh sửa thông tin cá nhân

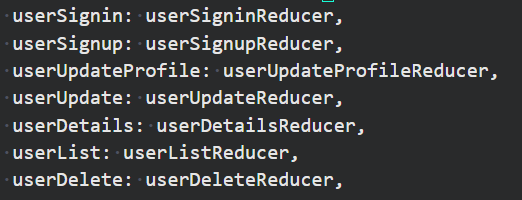


* TextHome: Trang chí text

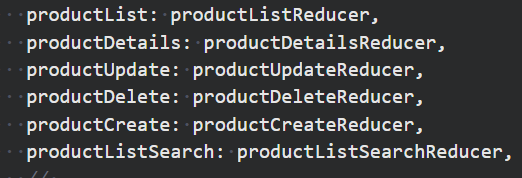


6.1.2.1.State Management

* User



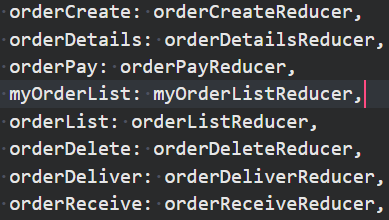
* Product



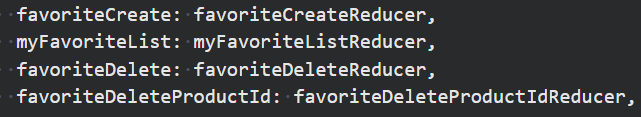
* Cart



* Order



* Favorite



### 6.2.Back-end

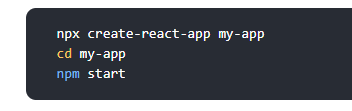
#### 6.2.1.Liệt kê thư viện

|  |  |  |
| --- | --- | --- |
| STT | Name | Description |
| 1 | @material-ui/core | Bộ thư viện Materia UI gồm icon, các component cung cấp sẵn,.. |
| 2 | @material-ui/icons |
| 3 | @material-ui/lab |
| 4 | axios | Axios là một thư viện HTTP Client dựa trên Promise cung cấp một API cho việc xử lý XHR (XMLHttpRequests). |
| 5 | bcryptjs | Mã hoá mật khẩu 1 chiều |
| 6 | bootstrap | Thư viện bootstrap |
| 7 | classnames | Đặt tên class linh động |
| 8 | react-bootstrap | Xử dụng các component được xây dựng từ bootstrap cho react |
| 9 | react-loading-skeleton | Hiệu ứng loading… |
| 10 | react-outside-click-handler | Phát hiện click ngoài phần tử component |
| 11 | react-redux | Quản lí state dễ dàng hơn |
| 12 | redux-thunk | Là middleware cho phép viết các Action trả về một function thay vì một plain javascript object bằng cách trì hoãn việc đưa action đến reducer. Redux Thunk được sử dụng để xử lý các logic bất đồng bộ phức tạp cần truy cập đến Store hoặc đơn giản là việc lấy dữ liệu như Ajax request |
| 13 | react-scripts | Cung cấp tập hợp các tập lệnh từ create-react-app gói khởi động. Khởi động các dự án mà không cần cấu hình, do đó không phải tự thiết lập trong dự án. |
| 14 | react-slick | Tạo các component dạng slider và setting tuỳ chỉnh theo ý thích. |
| 15 | slick-carousel |
| 16 | react-router-dom | Dùng để định tuyến (routing) tiêu chuẩn trong React. Dùng cho việc tạo SPA. |
| 17 | mongoose | Mongoose là một thư viện mô hình hóa đối tượng (Object Data Model - ODM) cho MongoDB, nó giúp cho lập trình viên dễ dàng truy cập và tương tác với cơ sở dữ liệu trong ứng dụng Express |
| 18 | body-parser | BodyParser trả về một function hoạt động như một middleware. Chức năng lắng nghe trên req.on (\'data\') và xây dựng req.body từ các đoạn dữ liệu mà nó nhận được. |
| 19 | dotenv | Dotenv là một mô đun zero-dependency, tải các biến môi trường từ tệp env vào process.env |
| 20 | jsonwebtoken | jsonwebtoken là package của Node phát triển dựa trên draft-ietf-jose-json-web-signature-08, dùng chủ yếu để xử lí Authentication, ví dụ: login,.. |

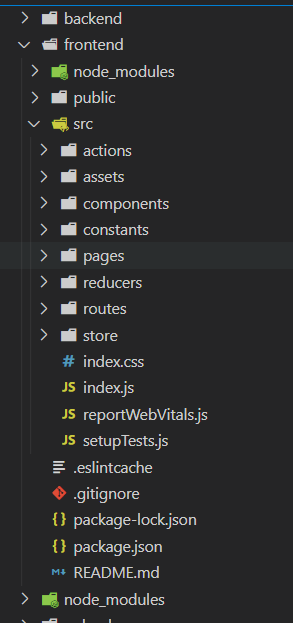
# CHƯƠNG 4: CÀI ĐẶT SẢN PHẨM

## 1.Front-end

* Sử dụng Visual Studio Code để xây dựng sản phẩm.
* Cài đặt Nodejs để có môi trường chạy ứng dụng : <https://nodejs.org/en/>
* Cài đặt ứng dụng Reactjs bằng câu lệnh bên dưới:

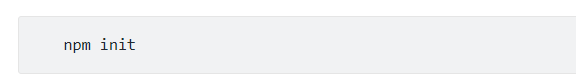


* Cấu trúc thư mục ứng dụng như hình sau:

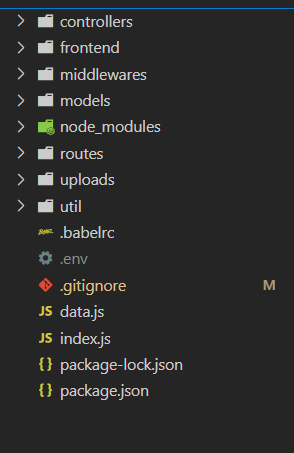


## 2.Back-end

* Khởi tạo ứng dụng bằng câu lệnh :



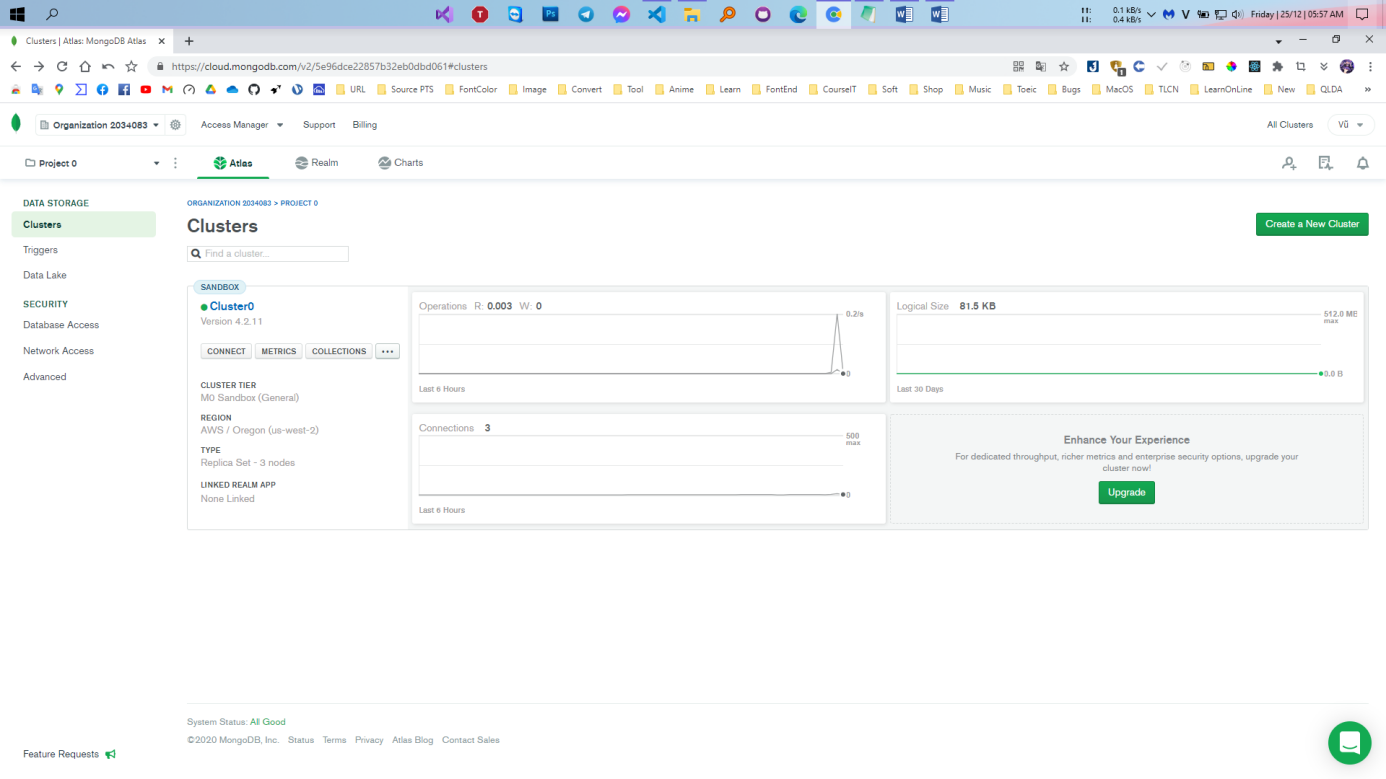
* Cấu trúc thư mục ứng dụng như hình sau:



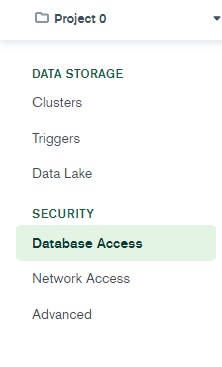
## 3.Deployment

### 3.1.Tạo database trên mongodb

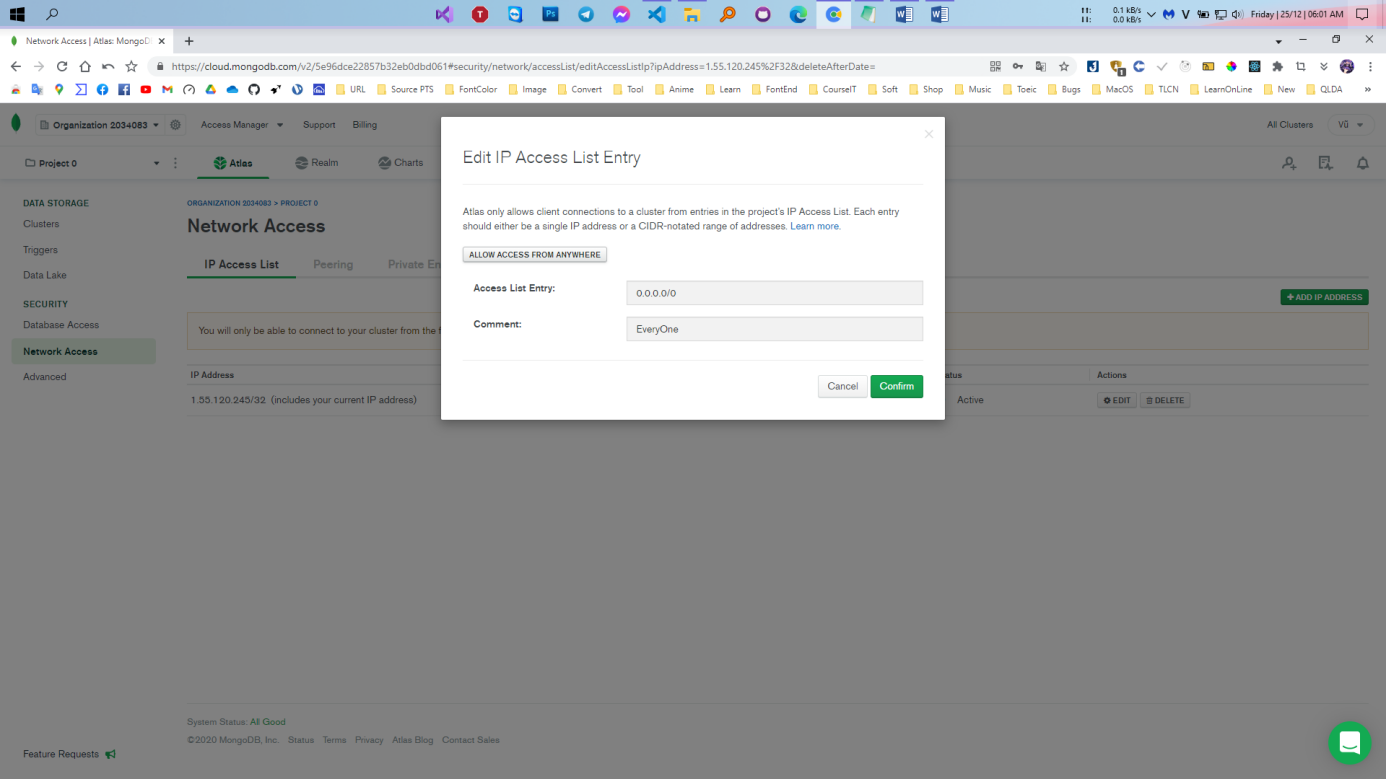
* Đăng kí tài khoản trên trang chủ <https://www.mongodb.com/>
* Sau khi đăng nhập tài khoản, ta có giao diện như bên dưới:



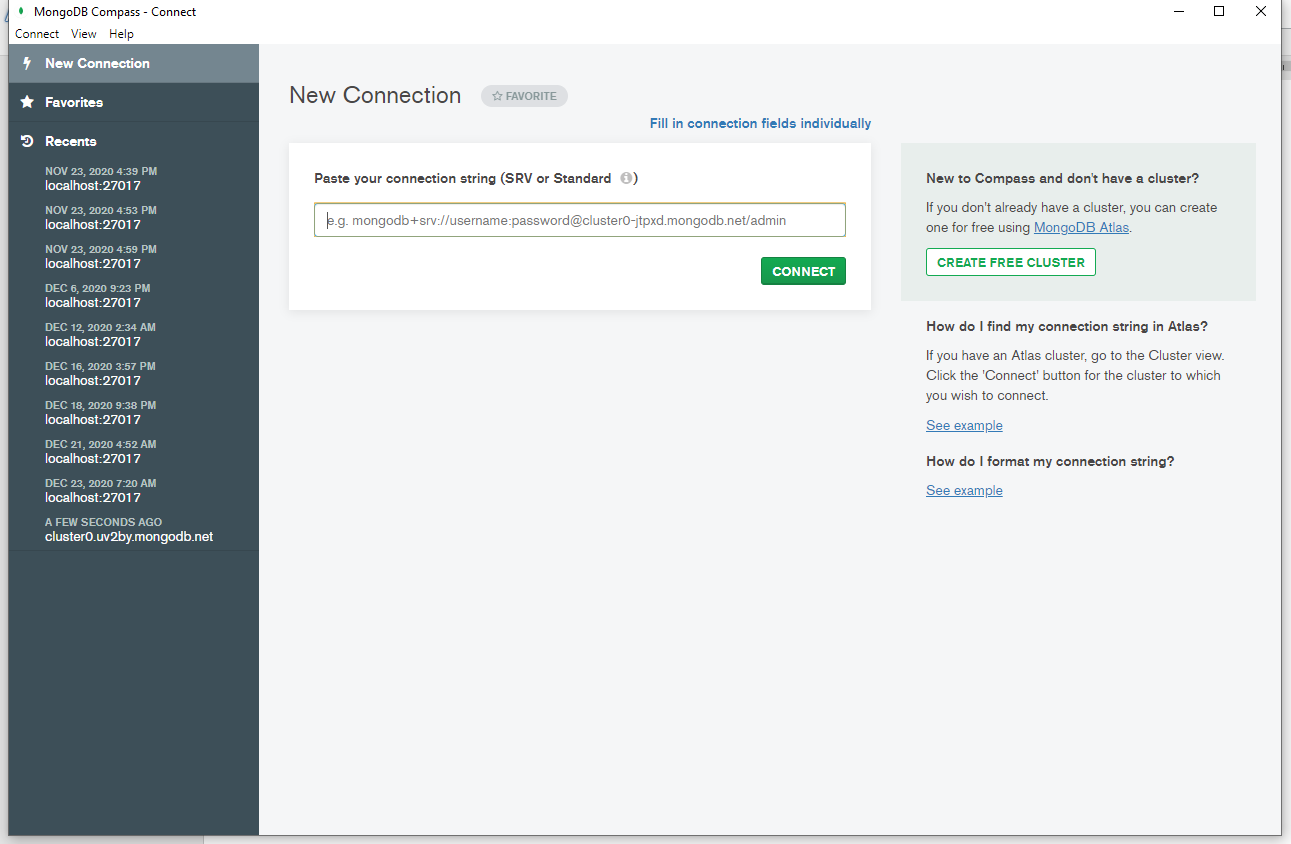
* Chọn Database Access để tạo tài khoản truy cập database:



* Tiếp đến chọn Network Access để quy định các địa chỉ IP có thể truy cập database, ở đây ta chọn “Allow access from any where”, tức là có thể truy cập không bị giới hạn địa chỉ IP.



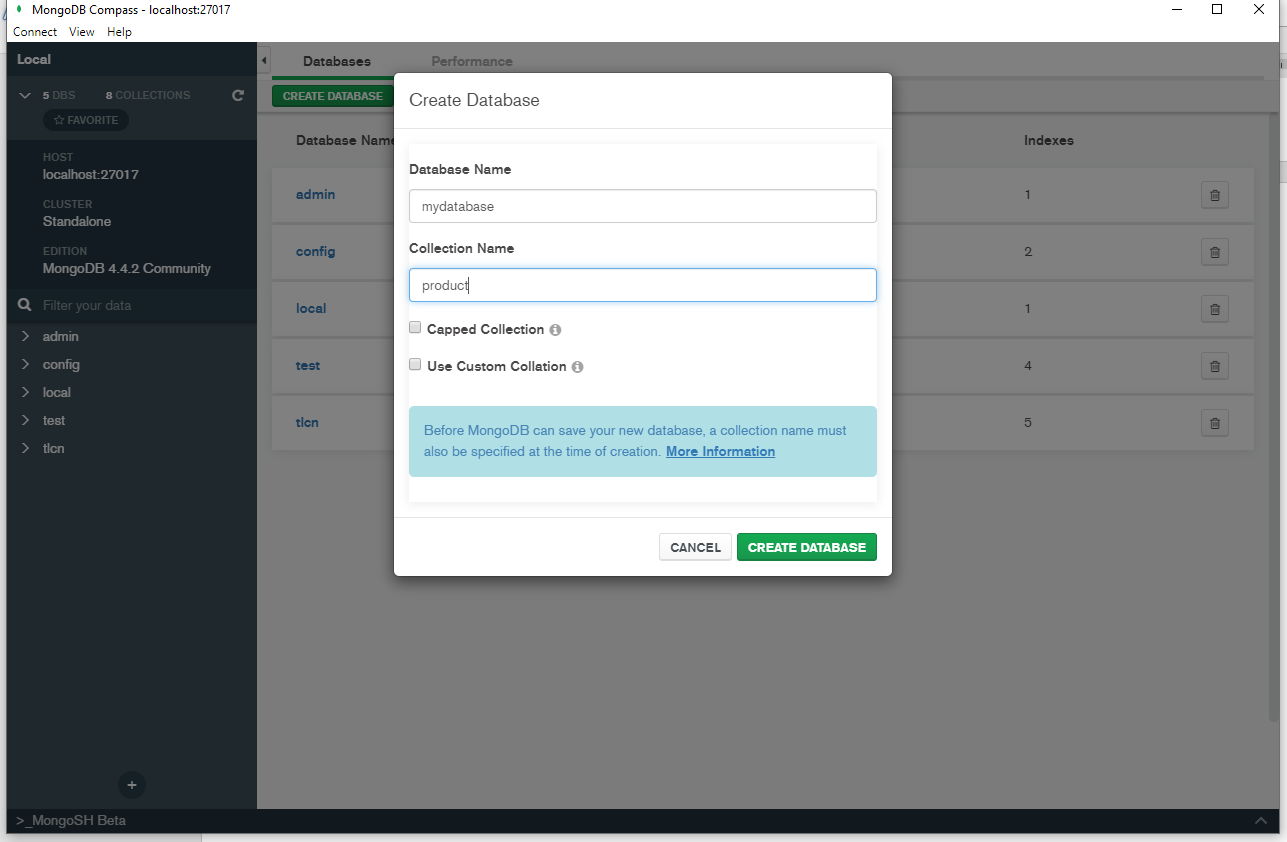
* Tải MongoDB Compass để quản lí database (thêm, sửa, xoá): <https://www.mongodb.com/try/download/compass>



* Mở ứng dụng lên và nhập đường dẫn theo cú pháp sau:

mongodb+srv://<username>:<password>@cluster0.uv2by.mongodb.net/test

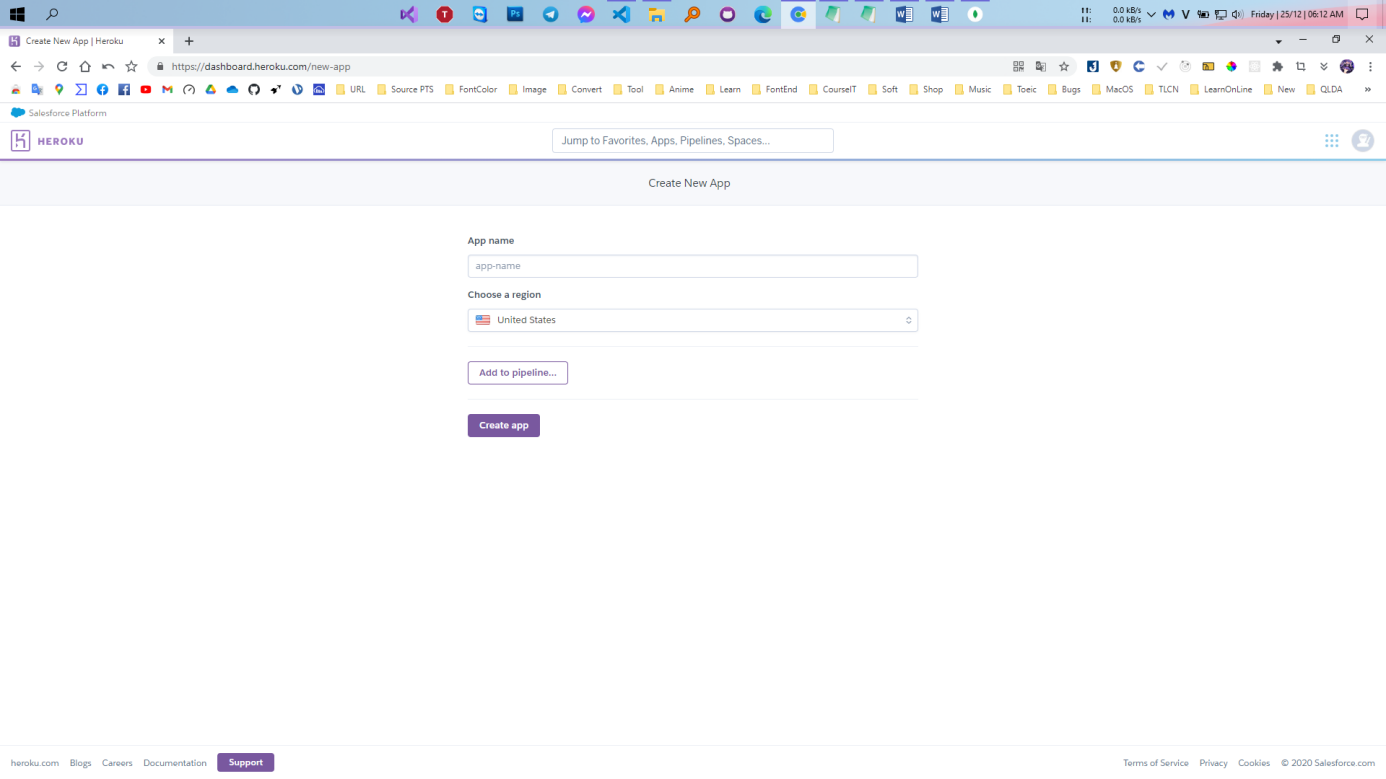
* Username và password là tên tài khoản đã tạo ở trên.
* Test là 1 database có sẵn mặc định đã được tạo bới mongodb để ta có thể truy cập và tạo database mới sau khi truy cập, ta sẻ đổi test thành tên database đã tạo.
* Ví dụ: mongodb+srv://vuboi:test@123 @cluster0.uv2by.mongodb.net/test



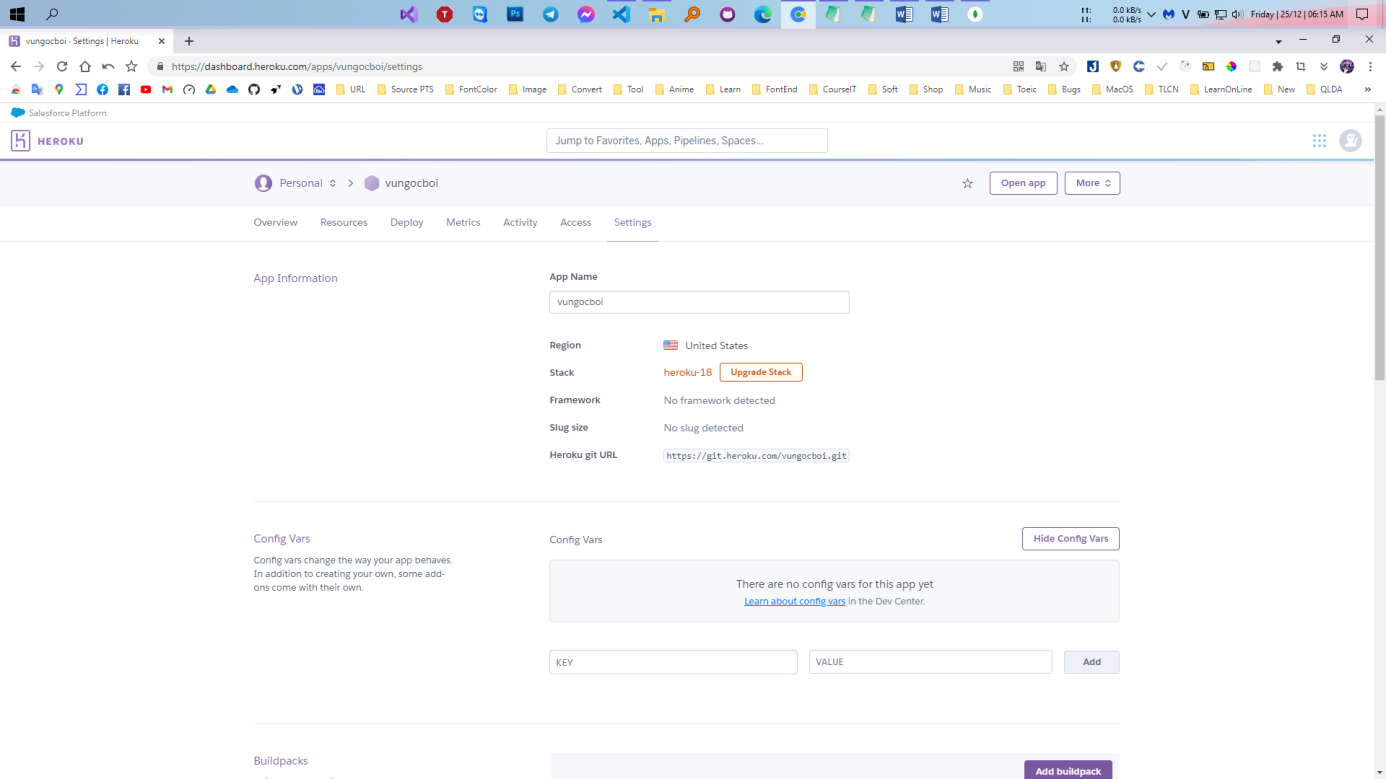
* Tạo mới database

### 3.2.Tải ứng dụng lên Heroku

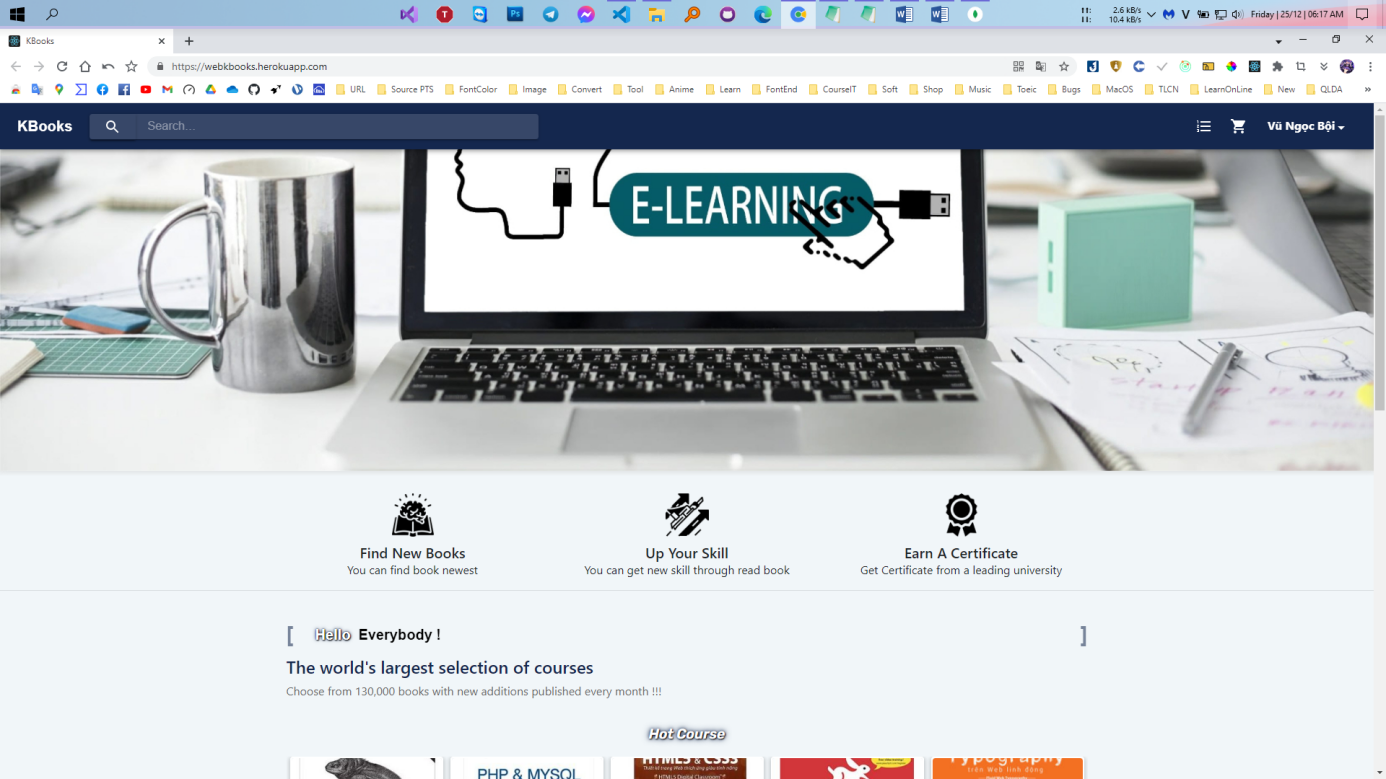
* Đăng kí tài khoản mới : <https://dashboard.heroku.com/>
* Tạo ứng dụng mới:



* Trong visual studio code tiến hành nhập các câu lệnh sau:
  + heroku login
  + cd my-project/
  + git init
  + heroku git:remote -a webkbooks
  + git add .
  + git commit -am "make it better"
  + git push heroku master
* Vào phần Setting app, thêm các biến môi trường (mongoURL, API Paypal,..)



* Kết quả:



# CHƯƠNG 5: TỔNG KẾT

## 1.Kết quả đạt được

* Xây dựng thành công website bán sách với các chức năng cơ bản.
* Tìm hiểu nắm bắt được các kiến thức về HTML, CSS, JAVASCRIPT, NOSQL,…
* Ứng dụng, phát triển được công nghệ thiết kế web bằng Nodejs, Reactjs
* Với quy mô trang web vừa và nhỏ, cộng thêm thời gian không cho phép nên đề tài “Xây dựng website bán sách online” có thể chưa có đầy đủ các chức năng hoàn thiện của một trang web bán hàng. Trong tương lai em sẽ cố gắng khắc phục những sai sót này.

## 2.Ưu điểm

* Ứng dụng đã có các chức năng cơ bản xem sản phẩm, đặt hàng, thanh toán, quản lí sản phẩm, giỏ hàng, đơn hàng, người dùng.
* Giao diện thân thiện người dùng, có cả trên máy tính và mobile.

## 3.Nhược điểm

* Xử lí logic còn sai sót, chưa tính đến nhiều trường hợp.
* Chưa hoàn thành chức đăng nhập xác thực, quên mật khẩu, bình luận, đánh giá sản phẩm.
* Tính bảo mật ứng dụng chưa cao, code chưa được tối ưu còn lộn xộn.

## 4.Khó khăn

* Hay gặp lỗi tình trạng dữ liệu bị Undefined.
* Giải quyết vấn đề còn chậm chạp khiến tiến độ ứng dụng không nhanh.

## 5.Hướng phát triển

* Hoàn thiện đầy đủ các chức năng cho ứng dụng.
* Tăng tính bảo mật của ứng dụng hơn.
* Bổ sung thêm một số chức năng mới như phân trang, thanh toán bằng Braintree, sản phẩm mới xem gần đây,…
* Điều chỉnh lại giao diện cho tối ưu, đẹp mặt hơn.