**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG KHOA CÔNG NGHỆ THÔNG TIN**



BÁO CÁO THỰC TẬP THỰC TẾ HÈ

**ĐỀ TÀI: XÂY DỰNG WEBSITE BÁN HẢI SẢN**

**(CT471-02)**

|  |  |
| --- | --- |
| **Sinh viên thực hiện** | **Cán bộ hướng dẫn** |
| **Họ và tên:** Huỳnh Hữu Kiên | Huỳnh Thanh Dư |
| **MSSV:** B2014666 | **Giảng viên phụ trách** |
| **Lớp:** DI20V7A2 | ThS. Nguyễn Minh Trung |

# Học kỳ 3 – 2023-2024

**LỜI CẢM ƠN**

Với các kiến thức đã tiếp thu được trong quá trình học tập tại trường Đại Học Cần Thơ. Được sự cho phép của Khoa Công Nghệ Thông Tin và sự tiếp nhận của VNPT Cà Mau đã tạo điều kiện cho em được thực tập tại quý Công Ty. Khoảng thời gian thực tập tuy ngắn ngủi nhưng em đã được học hỏi, được trải nghiệm những công việc và nghiệp vụ thực tế. Thời gian này đã cho em những bài học kinh nghiệm quý báu, những kỹ năng cần có về chuyên nghành mà em đang học. Đồng thời đây là tiền đề để em tự tin bước và môi trường làm việc sau này.

Vì thời gian thực tập hạn hẹp cũng như hạn chế về mặt kiến thức chuyên môn, do đó bài báo cáo của em không thể tránh khỏi những sai sót nhất định. Đồng thời bản thân báo cáo là kết quả của một quá trình tổng kết, những bài học rút từ trong quá trình thực tập và làm việc của em. Em rất mong được những góp ý, đánh giá từ phía thầy cô để bài báo cáo của em trở nên hoàn chỉnh hơn.

Sau cùng, em xin cảm ơn thầy Nguyễn Minh Trung – giảng viên khoa Công Nghệ Thông Tin – Đại học Cần Thơ đã hỗ trợ em trong quá trình thực tập. Đồng thời, trong thời gian thực tập tại cơ quan, em xin cảm ơn anh Huỳnh Khánh Dư cùng các anh/chị trong cơ quan giúp đỡ và chỉ dẫn tận tình, tạo điều kiện để em hoàn thành báo cáo của mình.

Một lần nữa, em xin chân thành cảm ơn quý thầy cô tại trường và các anh chị trong quá trình thực tập tại cơ quan.

*Cà Mau, ngày 04 tháng 07 năm 2024*

Sinh viên thực hiện

**Huỳnh Hữu Kiên**

**MỤC LỤC**

[CHƯƠNG 1: TỔ CHỨC HÀNH CHÍNH, NHÂN SỰ, HOẠT ĐỘNG CHUYÊN NGÀNH VÀ MÔI TRƯỜNG CỦA CƠ QUAN NƠI THỰC TẬP 1](#_Toc170630180)

[1.1. Tổ chức hành chính và nhân sự 1](#_Toc170630181)

[**1.1.1.** **Cấu trúc tổ chức** 1](#_Toc170630182)

[**1.1.2. Lực lượng cán bộ và công chức** 1](#_Toc170630183)

[1.2. Hoạt động chuyên ngành 1](#_Toc170630184)

[**1.2.1 Cung cấp dịch vụ viễn thông, internet, CNTT** 1](#_Toc170630185)

[**1.2.2. Triển khai các giải pháp CNTT cho doanh nghiệp** 2](#_Toc170630186)

[1.3. Môi trường làm việc 2](#_Toc170630187)

[**1.3.1. Cơ sở làm việc** 2](#_Toc170630188)

[**1.3.2. Chế độ đãi ngộ** 2](#_Toc170630189)

[CHƯƠNG 2: CÔNG VIỆC ĐƯỢC PHÂN CÔNG 3](#_Toc170630190)

[2.1. Công việc 3](#_Toc170630191)

[2.2. Mô hình Client-Server 3](#_Toc170630192)

[2.3. Tìm hiểu về Javascript, Node.js và thư viện Express. 3](#_Toc170630193)

[**2.3.1. Javascript là gì?** 3](#_Toc170630194)

[**2.3.1.** **Next.js là gì?** 4](#_Toc170630195)

[**2.3.2.** **Node.js là gì?** 4](#_Toc170630196)

[**2.3.3. ExpressJS là gì?** 5](#_Toc170630197)

[**2.3.4.** **Visual Studio Code là gì?** 5](#_Toc170630198)

[2.4. Tìm hiểu về mô hình thiết kế MVC (Model-View-Controller) 6](#_Toc170630199)

[CHƯƠNG 3: PHƯƠNG PHÁP THỰC HIỆN 7](#_Toc170630200)

[3.1. Thiết kế Server xử lý các yêu cầu từ phía Client 7](#_Toc170630201)

[3.2. Kết quả từ phía người sử dụng 14](#_Toc170630202)

[**3.2.1.** **Giao diện trang chủ** 14](#_Toc170630203)

[3.2.2. Giao diện trang người dùng 15](#_Toc170630204)

[**3.2.3.** **Giao diện giỏ hàng** 16](#_Toc170630205)

[**3.2.4.** **Giao diện thanh toán** 16](#_Toc170630206)

[**3.2.5.** **Giao diện trang quản trị** 17](#_Toc170630207)

[**3.2.6.** **Giao diện quản lý đơn hàng** 17](#_Toc170630208)

[**3.2.7.** **Giao diện trang quản lý sản phẩm** 18](#_Toc170630209)

[**3.2.8.** **Giao diện trang quản lý người dùng** 18](#_Toc170630210)

[CHƯƠNG 4: KẾT QUẢ ĐẠT ĐƯỢC 19](#_Toc170630211)

[4.1. Những nội dung được củng cố 19](#_Toc170630212)

[4.2. Kinh nghiệm được tích lũy 19](#_Toc170630213)

[**4.2.1. Kỹ Thuật và Lập Trình** 19](#_Toc170630214)

[**4.2.2. Kiến Thức Về Thương Mại Điện Tử** 19](#_Toc170630215)

[**4.2.3. Kỹ Năng Quản Lý Dự Án** 20](#_Toc170630216)

[**4.2.4. Kỹ Năng Giải Quyết Vấn Đề** 20](#_Toc170630217)

[4.3. Kết quả công việc đã đóng góp cho cơ quan nơi thực tập 20](#_Toc170630218)

DANH MỤC HÌNH ẢNH

[Hình 1. 1 Giao diện trang chủ 15](#_Toc170630268)

[Hình 1. 2 Giao diện trang người dùng 15](#_Toc170630269)

[Hình 1. 3 Giao diện quản lý đơn hàng 16](#_Toc170630270)

[Hình 1. 4 Giao diện giỏ hàng 16](#_Toc170630271)

[Hình 1. 5 Giao diện trang thanh toán 17](#_Toc170630272)

[Hình 1. 6 Giao diện trang quản trị 17](#_Toc170630273)

[Hình 1. 7 Giao diện trang quản lý đơn hàng 18](#_Toc170630274)

[Hình 1. 8 Giao diện trang quản lý sản phẩm 18](#_Toc170630275)

[Hình 1. 9 Giao diện trang quản lý người dùng 18](#_Toc170630276)

# CHƯƠNG 1: TỔ CHỨC HÀNH CHÍNH, NHÂN SỰ, HOẠT ĐỘNG CHUYÊN NGÀNH VÀ MÔI TRƯỜNG CỦA CƠ QUAN NƠI THỰC TẬP

# Tổ chức hành chính và nhân sự

## **Cấu trúc tổ chức**

Trung tâm Công nghệ thông tin VNPT Cà Mau (TTCT VNPT Cà Mau) có cấu trúc tổ chức gồm các phòng ban chức năng chính sau:

* **Ban Giám đốc:** Chịu trách nhiệm chung về hoạt động của Trung tâm.
* **Phòng Kế hoạch - Tài chính:** Quản lý kế hoạch, tài chính, đầu tư của Trung tâm.
* **Phòng Kỹ thuật:** Phụ trách triển khai, quản lý hạ tầng mạng, hệ thống thông tin, phần mềm của Trung tâm.
* **Phòng Dịch vụ khách hàng:** Cung cấp dịch vụ viễn thông, internet, CNTT cho khách hàng.
* **Phòng An ninh mạng:** Bảo đảm an ninh mạng cho hạ tầng, hệ thống thông tin của Trung tâm.
* **Phòng Nhân sự - Hành chính:** Quản lý nhân sự, tổ chức, hành chính của Trung tâm.

## **1.1.2. Lực lượng cán bộ và công chức**

Trung tâm công nghệ thông tin VNPT Cà Mau có đội ngũ cán bộ, công chức có trình độ chuyên môn cao, giàu kinh nghiệm, nhiệt tình, năng động.

* **Đội ngũ kỹ thuật:** Là những kỹ sư CNTT, lập trình viên có chuyên môn cao, am hiểu về các lĩnh vực như mạng máy tính, hệ thống thông tin, phần mềm.
* **Đội ngũ nhân viên dịch vụ khách hàng:** Có kỹ năng giao tiếp tốt, chuyên nghiệp, sẵn sàng phục vụ khách hàng mọi lúc mọi nơi.
* **Đội ngũ cán bộ quản lý:** Có trình độ quản lý cao, khả năng lãnh đạo tốt, có tầm nhìn chiến lược.

# Hoạt động chuyên ngành

## **1.2.1 Cung cấp dịch vụ viễn thông, internet, CNTT**

Trung tâm công nghệ thông tin VNPT Cà Mau cung cấp đa dạng các dịch vụ viễn thông, internet, CNTT cho khách hàng cá nhân, doanh nghiệp, tổ chức trên địa bàn tỉnh Cà Mau.

* **Dịch vụ viễn thông:** Gồm dịch vụ di động (Vinaphone), dịch vụ internet cáp quang (MegaVNN), dịch vụ truyền hình (MyTV), dịch vụ internet cáp đồng (ADSL), dịch vụ điện thoại cố định.
* **Dịch vụ internet:** Cung cấp đường truyền internet tốc độ cao, ổn định, đáp ứng nhu cầu sử dụng internet của khách hàng.
* **Dịch vụ CNTT:** Cung cấp các dịch vụ CNTT như thiết kế website, lập trình phần mềm, bảo trì hệ thống máy tính, ...

## **1.2.2. Triển khai các giải pháp CNTT cho doanh nghiệp**

Trung tâm công nghệ thông tin VNPT Cà Mau cung cấp các giải pháp CNTT toàn diện cho doanh nghiệp, giúp doanh nghiệp nâng cao hiệu quả hoạt động, tiết kiệm chi phí.

* **Giải pháp hạ tầng mạng:** Thiết kế, thi công, bảo trì hạ tầng mạng cho doanh nghiệp.
* **Giải pháp hệ thống thông tin:** Triển khai, quản lý hệ thống thông tin cho doanh nghiệp.
* **Giải pháp phần mềm:** Cung cấp các phần mềm quản lý doanh nghiệp, phần mềm kế toán, phần mềm CRM, ...
* **Giải pháp an ninh mạng:** Bảo vệ hệ thống thông tin của doanh nghiệp khỏi các mối đe dọa an ninh mạng.

# Môi trường làm việc

## **1.3.1. Cơ sở làm việc**

Trung tâm công nghệ thông tin VNPT Cà Mau có trụ sở làm việc khang trang, hiện đại, được trang bị đầy đủ các thiết bị văn phòng, máy móc hiện đại. Trung tâm có hệ thống mạng lưới viễn thông rộng khắp, đảm bảo thông tin liên lạc thông suốt.

## **1.3.2. Chế độ đãi ngộ**

Trung tâm công nghệ thông tin VNPT Cà Mau áp dụng chế độ đãi ngộ cạnh tranh cho cán bộ, công chức, phù hợp với năng lực, trình độ chuyên môn. Trung tâm có nhiều chế độ phúc lợi tốt như bảo hiểm xã hội, bảo hiểm y tế,..

# CHƯƠNG 2: CÔNG VIỆC ĐƯỢC PHÂN CÔNG

# Công việc

Xây dựng một trang web hỗ trợ mua bán và quản lý cửa hàng hải sản, sử dụng NextJs, và NodeJS, Express để làm máy chủ (Server) xử lý các hoạt động trả về máy khách.

Đáp ứng được các nhu cầu cơ bản như quản lý bán hàng, quản lý sản phẩm, báo cáo doanh thu, cung cấp giao diện người dùng thân thiện với người dùng. Các công nghệ sử dụng như: NextJs được dùng cho frontend, Node.js và Express cho API và quản lý dữ liệu, MongoDB cho việc lưu trữ dữ liệu và nhiều thư viện hỗ trợ khác như: cloudinary, reCharts,…

Máy chủ được xây dựng bằng Node.js và Express, cung cấp các API để xử lý yêu cầu từ máy khách (Client). Các yêu cầu từ Client sẽ được xử lý và phản hồi dưới dạng JSON.

Client sẽ được xây dựng bằng NextJs để tạo nên giao diện thân thiện với người dùng. Client sẽ cho phép người dùng tìm kiếm, lọc, thêm sản phẩm vào giỏ hàng, quản lý hóa đơn và nhiều chức năng cơ bản khác.

# Mô hình Client-Server

**Mô hình Client - Server (Máy khách - Máy chủ)** là một kiến trúc mạng phổ biến và cơ bản được ứng dụng rộng rãi trong lĩnh vực công nghệ thông tin

* **Máy khách (Client)** đóng vai trò là người yêu cầu dịch vụ, thường là các thiết bị hoặc phần mềm như máy tính cá nhân, thiết bị di động, ứng dụng web,... Máy khách được trang bị giao diện người dùng thân thiện, dễ sử dụng, cho phép người dùng gửi yêu cầu đến máy chủ một cách trực quan.
* **Máy chủ (Server)** đóng vai trò là người cung cấp dịch vụ, chịu trách nhiệm xử lý các yêu cầu từ máy khách. Máy chủ thường là các máy tính mạnh mẽ hoặc cụm máy chủ được tối ưu hóa để xử lý nhiều yêu cầu đồng thời. Máy chủ cung cấp các dịch vụ đa dạng như lưu trữ dữ liệu, quản lý cơ sở dữ liệu, xử lý ứng dụng, cung cấp trang web,...

# Tìm hiểu về Javascript, Node.js và thư viện Express.

## **2.3.1. Javascript là gì?**

JavaScript là một ngôn ngữ lập trình thông dịch (interpreted programming language) được sử dụng chủ yếu để tạo ra các nội dung động trên các trang web. Ban đầu, JavaScript được phát triển để thêm tính năng tương tác động vào các trang web, nhưng sau đó đã mở rộng ra để phát triển ứng dụng web và ứng dụng di động.

JavaScript đã trở thành một ngôn ngữ lập trình quan trọng và không thể thiếu trong việc phát triển các ứng dụng web hiện đại, đóng vai trò quan trọng trong việc nâng cao trải nghiệm người dùng và khả năng tương tác của các ứng dụng trực tuyến.

* + 1. **Next.js là gì?**

Next.js là một **khung phát triển React** mạnh mẽ, giúp đơn giản hóa việc tạo các ứng dụng web React hiện đại, hiệu quả và dễ bảo trì. Nó cung cấp nhiều tính năng vượt trội so với React thuần túy, bao gồm:

* **Kết xuất phía máy chủ (SSR):** Tăng tốc độ tải trang và cải thiện SEO.
* **Tạo trang tĩnh (SSG):** Tạo các trang web tĩnh tối ưu hóa cho hiệu suất và SEO.
* **Hệ thống định tuyến linh hoạt:** Dễ dàng tạo cấu trúc URL và điều hướng.
* **Khả năng tương thích API cao:** Tích hợp dễ dàng với các dịch vụ bên ngoài.
* **Khuyến khích tái sử dụng thành phần:** Viết mã hiệu quả và dễ bảo trì.
* **Trải nghiệm nhà phát triển tuyệt vời:** Nạp lại nóng, công cụ gỡ lỗi mạnh mẽ.
* **Cộng đồng lớn và tích cực:** Hỗ trợ và chia sẻ kiến thức.

Next.js là lựa chọn lý tưởng cho các nhà phát triển web muốn tạo các ứng dụng web React hiệu suất cao, thân thiện với SEO và dễ sử dụng.

## **Node.js là gì?**

**Node.js** là một môi trường phần mềm mở nguồn, miễn phí, được xây dựng trên nền tảng JavaScript và sử dụng mô hình xử lý sự kiện phi đồng bộ (asynchronous event-driven). Node.js được phát triển bởi Ryan Dahl vào năm 2009, nhằm mục đích giải quyết vấn đề hiệu suất và khả năng mở rộng cho các ứng dụng web.

**Ưu điểm của Node.js:**

* **Hiệu suất cao:** Node.js sử dụng mô hình xử lý sự kiện phi đồng bộ, cho phép xử lý nhiều yêu cầu đồng thời một cách hiệu quả.
* **Khả năng mở rộng cao:** Node.js có thể dễ dàng mở rộng bằng cách thêm thêm máy chủ.
* **Dễ sử dụng:** Node.js sử dụng JavaScript, một ngôn ngữ lập trình phổ biến và dễ học.
* **Phù hợp cho các ứng dụng web thời gian thực:** Node.js lý tưởng cho các ứng dụng web thời gian thực như chat, game online, ...
* **Cộng đồng lớn và tích cực:** Node.js có cộng đồng nhà phát triển lớn và tích cực, luôn sẵn sàng hỗ trợ nhau.

Node.js được sử dụng để phát triển nhiều loại ứng dụng khác nhau, bao gồm:

* **Ứng dụng web:** Web server, chat, game online, ...
* **Ứng dụng desktop:** Giúp viết các ứng dụng desktop bằng JavaScript.
* **Công cụ dòng lệnh:** Viết các công cụ dòng lệnh mạnh mẽ và hiệu quả.
* **Internet of Things (IoT):** Phát triển các ứng dụng IoT bằng Node.js.

Nhìn chung, Node.js là một môi trường phần mềm mạnh mẽ, linh hoạt và dễ sử dụng, giúp bạn dễ dàng phát triển các ứng dụng web hiện đại, hiệu quả và dễ mở rộng.

## **2.3.3. ExpressJS là gì?**

Express.js là một framework Node.js phổ biến được sử dụng để xây dựng các ứng dụng web và API servers. Được phát triển từ Node.js, Express cung cấp các tính năng và công cụ hỗ trợ để phát triển ứng dụng web một cách nhanh chóng và hiệu quả.

Express.js cung cấp một cách tiếp cận linh hoạt để định tuyến (routing) các yêu cầu HTTP từ các client (khách hàng) đến các handler (xử lý) tương ứng. Điều này giúp phân phối các yêu cầu vào các định tuyến khác nhau dựa trên URL và phương thức HTTP (GET, POST, PUT, DELETE, vv.). Middleware là một khái niệm quan trọng trong Express.js, cho phép bạn mở rộng chức năng của ứng dụng bằng cách thêm các lớp xử lý trung gian giữa yêu cầu đến và từ server. Middleware có thể thực hiện các chức năng như xác thực, logging, xử lý lỗi, và nhiều công việc khác. Express.js là lựa chọn phổ biến cho việc phát triển các API servers, nhờ vào khả năng xử lý các yêu cầu HTTP một cách dễ dàng và hiệu quả.

## **Visual Studio Code là gì?**

Visual Studio Code (VSCode) là trình soạn thảo mã nguồn miễn phí và mã nguồn mở, được phát triển bởi Microsoft. VSCode nổi tiếng với khả năng hỗ trợ đa nền tảng (Windows, macOS, Linux), tốc độ khởi động nhanh, giao diện trực quan và khả năng mở rộng mạnh mẽ thông qua hệ thống plugin phong phú.

VSCode hỗ trợ nhiều ngôn ngữ lập trình phổ biến như C++, Java, Python, JavaScript, PHP, Go, ... và cung cấp nhiều tính năng hữu ích cho lập trình viên như:

* **Tô sáng cú pháp:** Giúp dễ dàng phân biệt các phần tử cú pháp trong mã nguồn.
* **Hoàn thành mã tự động:** Đề xuất các gợi ý hoàn thành mã dựa trên ngữ cảnh.
* **Tìm kiếm và sửa lỗi:** Tìm kiếm và sửa lỗi cú pháp, logic trong mã nguồn.
* **Refactoring code:** Tự động sắp xếp, chỉnh sửa cấu trúc code.
* **Debugging:** Gỡ lỗi code hiệu quả với nhiều tính năng hỗ trợ.
* **Tích hợp Git:** Quản lý phiên bản code trực tiếp trong VSCode.

Với những ưu điểm vượt trội, Visual Studio Code đã trở thành lựa chọn hàng đầu cho các lập trình viên trên toàn thế giới.

# Tìm hiểu về mô hình thiết kế MVC (Model-View-Controller)

Mô hình MVC (Model-View-Controller) là một kiến trúc phần mềm phổ biến được sử dụng để thiết kế các ứng dụng web và desktop. MVC chia ứng dụng thành ba thành phần chính: Model, View, và Controller, mỗi thành phần có một vai trò cụ thể và rõ ràng.

1. **Model**:
   * Chịu trách nhiệm quản lý dữ liệu và logic nghiệp vụ của ứng dụng.
   * Tương tác với cơ sở dữ liệu để lấy và lưu trữ dữ liệu.
   * Thực hiện các phép toán và xử lý dữ liệu.
2. **View**:
   * Chịu trách nhiệm hiển thị dữ liệu cho người dùng.
   * Nhận dữ liệu từ Model và định dạng để hiển thị trên giao diện người dùng.
   * Có thể bao gồm các thành phần như HTML, CSS, và JavaScript trong các ứng dụng web.
3. **Controller**:
   * Điều khiển luồng hoạt động của ứng dụng.
   * Nhận yêu cầu từ người dùng thông qua View, xử lý yêu cầu đó (thường bằng cách tương tác với Model), và trả lại kết quả qua View.
   * Xác định và điều phối các phản hồi thích hợp giữa View và Model.

Lợi ích của MVC:

* **Phân tách rõ ràng**: Giúp tách biệt các thành phần logic, giao diện và điều khiển, làm cho mã nguồn dễ hiểu và dễ bảo trì.
* **Dễ dàng bảo trì và mở rộng**: Các thay đổi trong một thành phần ít ảnh hưởng đến các thành phần khác, giúp việc bảo trì và mở rộng trở nên dễ dàng hơn.
* **Hỗ trợ phát triển theo nhóm**: Với sự phân chia rõ ràng, nhiều lập trình viên có thể làm việc đồng thời trên các phần khác nhau của ứng dụng mà không gây xung đột.
* **Tái sử dụng mã nguồn**: Các thành phần có thể được tái sử dụng trong các dự án khác nhau, giảm thiểu mã lặp và tiết kiệm thời gian phát triển.

MVC thường được sử dụng trong các framework web như Ruby on Rails, Laravel (PHP), Spring MVC (Java), ASP.NET MVC (C#), và Express.js (Node.js) để phân chia rõ ràng logic của ứng dụng. Sự tách biệt này không chỉ giúp mã nguồn trở nên rõ ràng hơn mà còn tạo điều kiện thuận lợi cho việc kiểm thử và phát triển theo hướng module hóa

# CHƯƠNG 3: PHƯƠNG PHÁP THỰC HIỆN

Làm việc cá nhân – Xây dựng website được theo thời gian với từng công việc cụ thể.

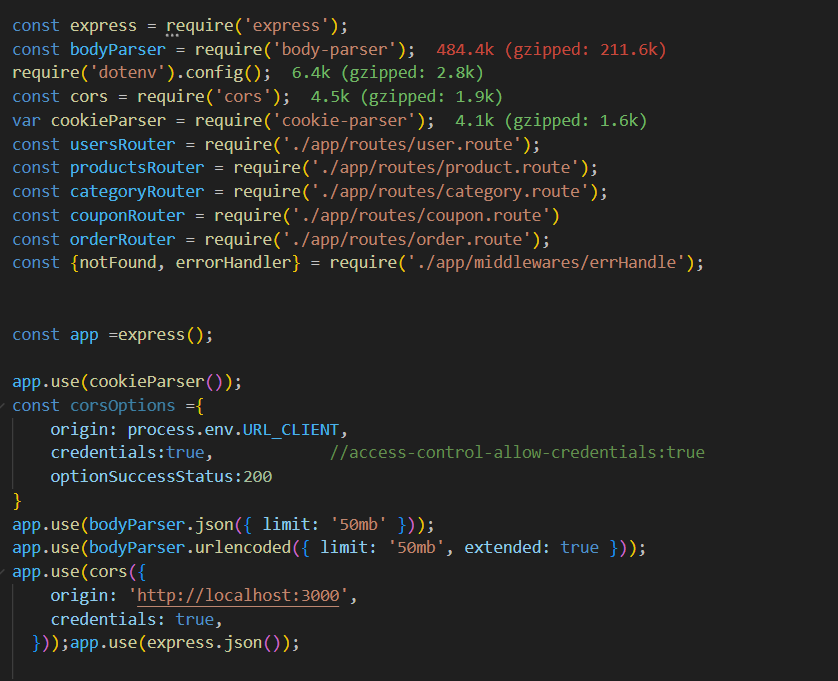
# Thiết kế Server xử lý các yêu cầu từ phía Client

Các cài đặt cần thiết:

Sử dụng Visual Studio Code cài đặt các gói: body-parser, cors, mongoose, multer, jsonwebtoken, nodemon.

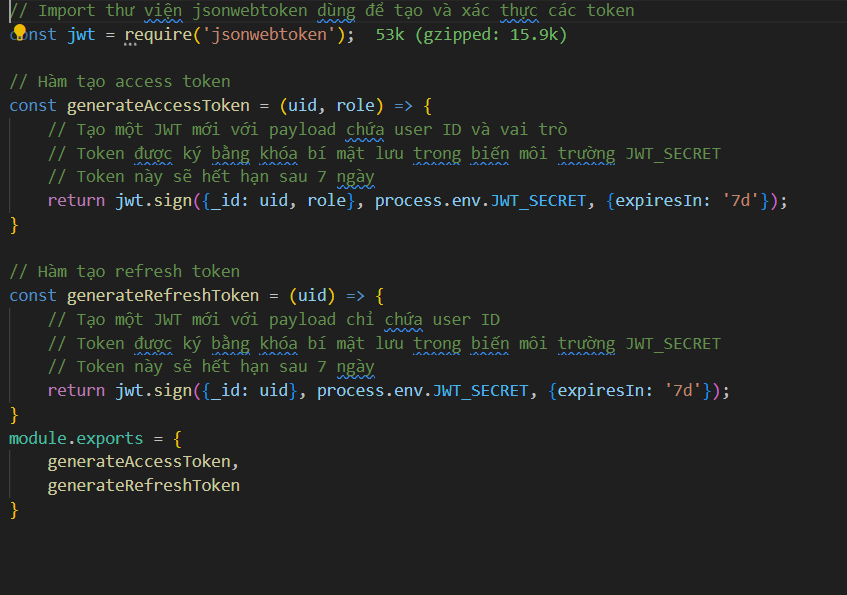
|  |
| --- |
| npm install body-parser |
| npm install cors |
| npm install mongoose |
| npm install --save multer |
| npm install jsonwebtoken |
| npm install -g nodemon |
| npm install dotenv |
| npm cookie-parser |
| npm multer-storage-cloudinary |

Import các thư viện cần thiết cho server:



Trong đó:

* bodyParser: Middleware để phân tích dữ liệu từ yêu cầu HTTP.
* dotenv: Để quản lý các biến môi trường từ tệp .env.
* cors: Để cấu hình chia sẻ tài nguyên giữa các nguồn gốc khác nhau.
* cookieParser: Middleware để xử lý cookie.
* Các tệp user.route, product.route, category.route, order.route: Các tuyến đường để xử lý yêu cầu liên quan đến người dùng, sản phẩm, danh mục, và đơn hàng.



// Import thư viện jsonwebtoken để xử lý JWT và express-async-handler để xử lý các hàm middleware bất đồng bộ

const jwt = require('jsonwebtoken');

const asyncHandler = require('express-async-handler');

// Middleware xác thực access token

const verifyAccessToken = asyncHandler(async (req, res, next) => {

    // Kiểm tra xem header Authorization có bắt đầu bằng "Bearer" không

    if (req?.headers?.authorization?.startsWith('Bearer')) {

        // Lấy token từ header Authorization

        const token = req.headers.authorization.split(' ')[1];

        // Xác thực token

        jwt.verify(token, process.env.JWT\_SECRET, (error, decode) => {

            if (error) {

                // Trả về lỗi nếu token không hợp lệ

                return res.status(401).json({

                    success: false,

                    message: 'Invalid access Token!'

                });

            }

            // In thông tin giải mã của token ra console (tùy chọn)

            console.log(decode);

            // Lưu thông tin người dùng vào req.user để sử dụng trong các middleware tiếp theo

            req.user = decode;

            // Chuyển sang middleware tiếp theo

            next();

        });

    } else {

        // Trả về lỗi nếu không có token trong header Authorization

        return res.status(401).json({

            success: false,

            message: 'Require authentication!'

        });

    }

});

// Middleware kiểm tra vai trò admin

const isAdmin = asyncHandler(async (req, res, next) => {

    // Lấy vai trò của người dùng từ req.user

    const { role } = req.user;

    // Kiểm tra nếu vai trò không phải admin (mã vai trò admin là 2002)

    if (+role != 2002) {

        // Trả về lỗi nếu người dùng không phải admin

        return res.status(401).json({

            success: false,

            message: 'Require admin role'

        });

    }

    // Chuyển sang middleware tiếp theo

    next();

});

module.exports = {

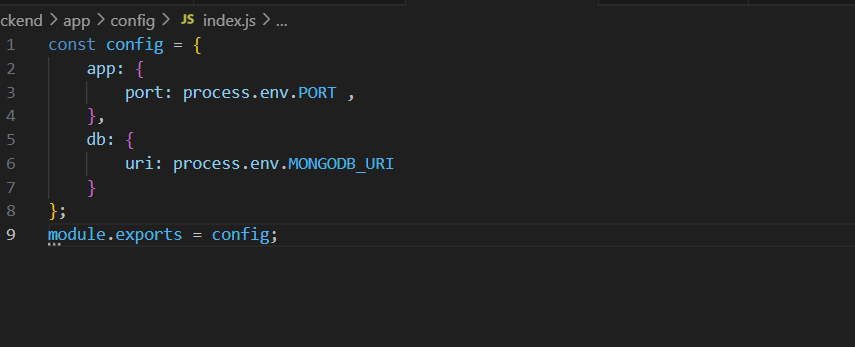
    verifyAccessToken,

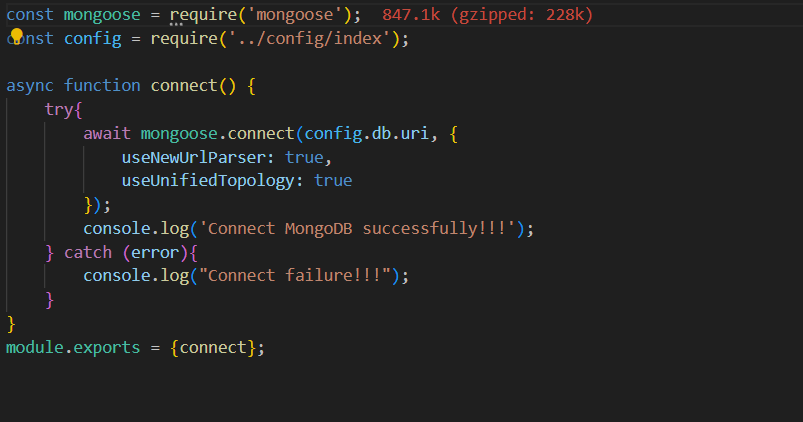
    isAdmin

}

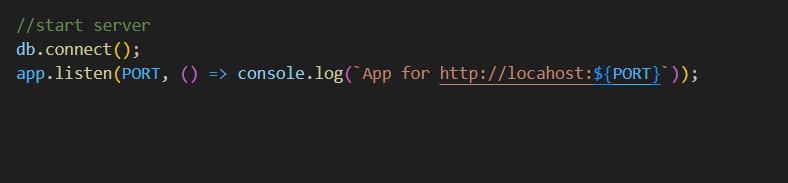


Cấu hình Cloudinary để lưu trữ hình ảnh và sử dụng multer để xử lý việc tải lên các tập tin (file uploads). Cloudinary là một dịch vụ đám mây chuyên về quản lý hình ảnh và video, còn multer là một middleware cho Express.js để xử lý việc tải lên tập tin.

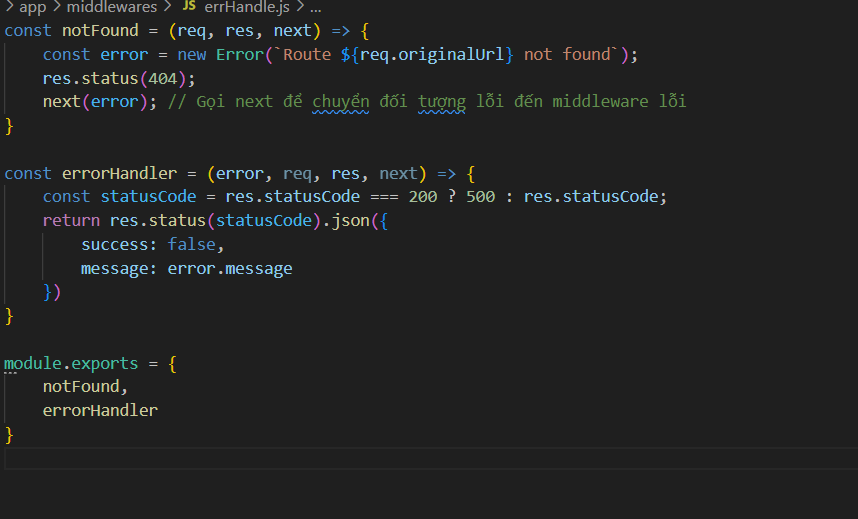




Trong config chứa app được sử dụng để khai báo cổng mà máy chủ sẽ lắng nghe các yêu cầu từ phía client. Đối với bd được sử dụng để khai báo cổng kêt nối tới MongoDB Atlas. Cấu hình và kết nối đến cơ sở dữ liệu MongoDB bằng mongoose. Hàm connect dùng để thiết lập kết nối. Trong trường hợp thành công, hiển thị thông báo xác nhận kết nối. Nếu kết nối thất bại, hiển thị thông báo lỗi.

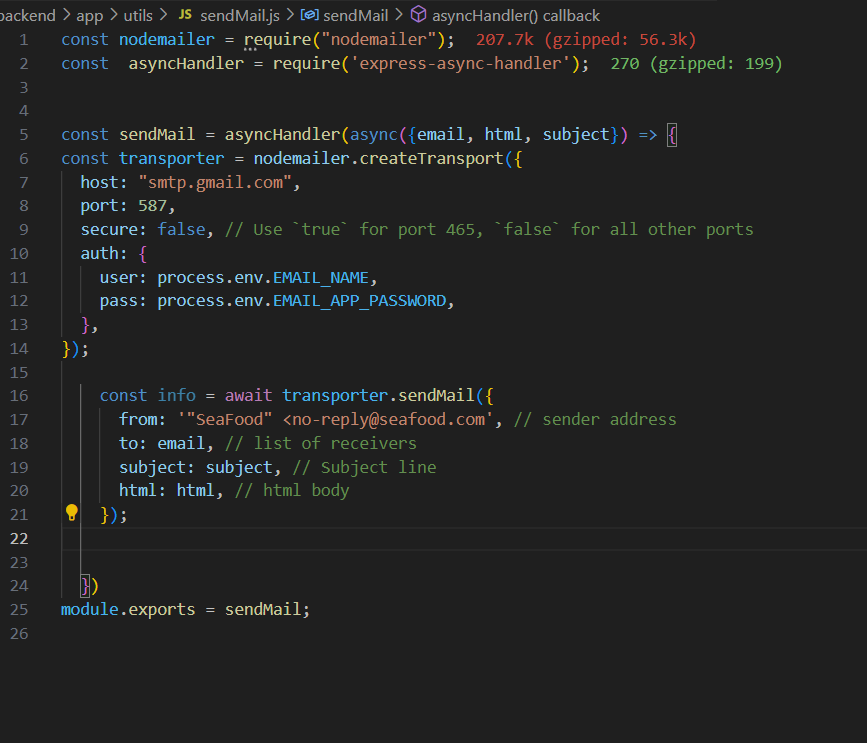


Hàm connect() được gọi để thiết lập kết nối đến cơ sở dữ liệu MongoDB thông qua thư viện mongoose.

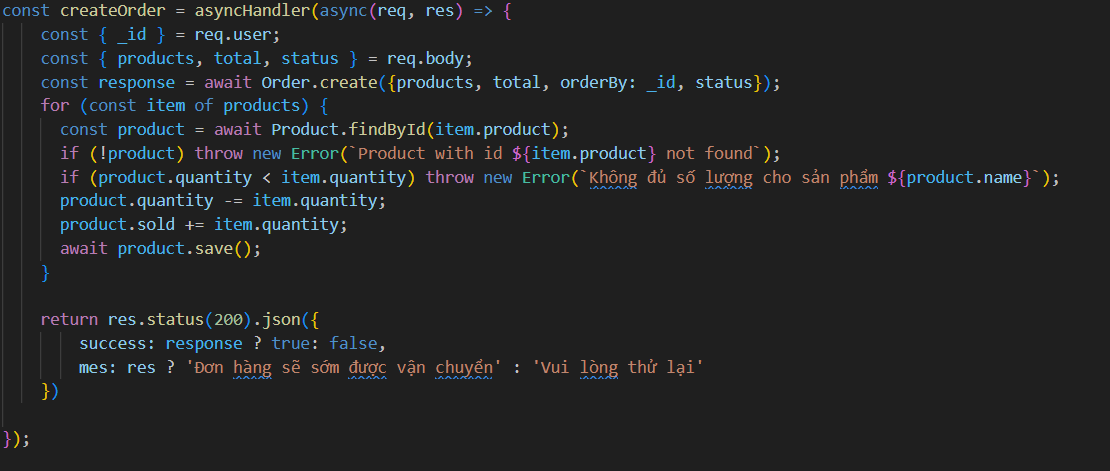


Đoạn mã này định nghĩa hai middleware để xử lý lỗi trong ứng dụng Express.js:

* notFound: Xử lý các yêu cầu đến các tuyến đường không tồn tại và chuyển lỗi 404 đến middleware tiếp theo.
* errorHandler: Xử lý tất cả các lỗi khác và trả về phản hồi JSON với mã trạng thái lỗi và thông báo lỗi.

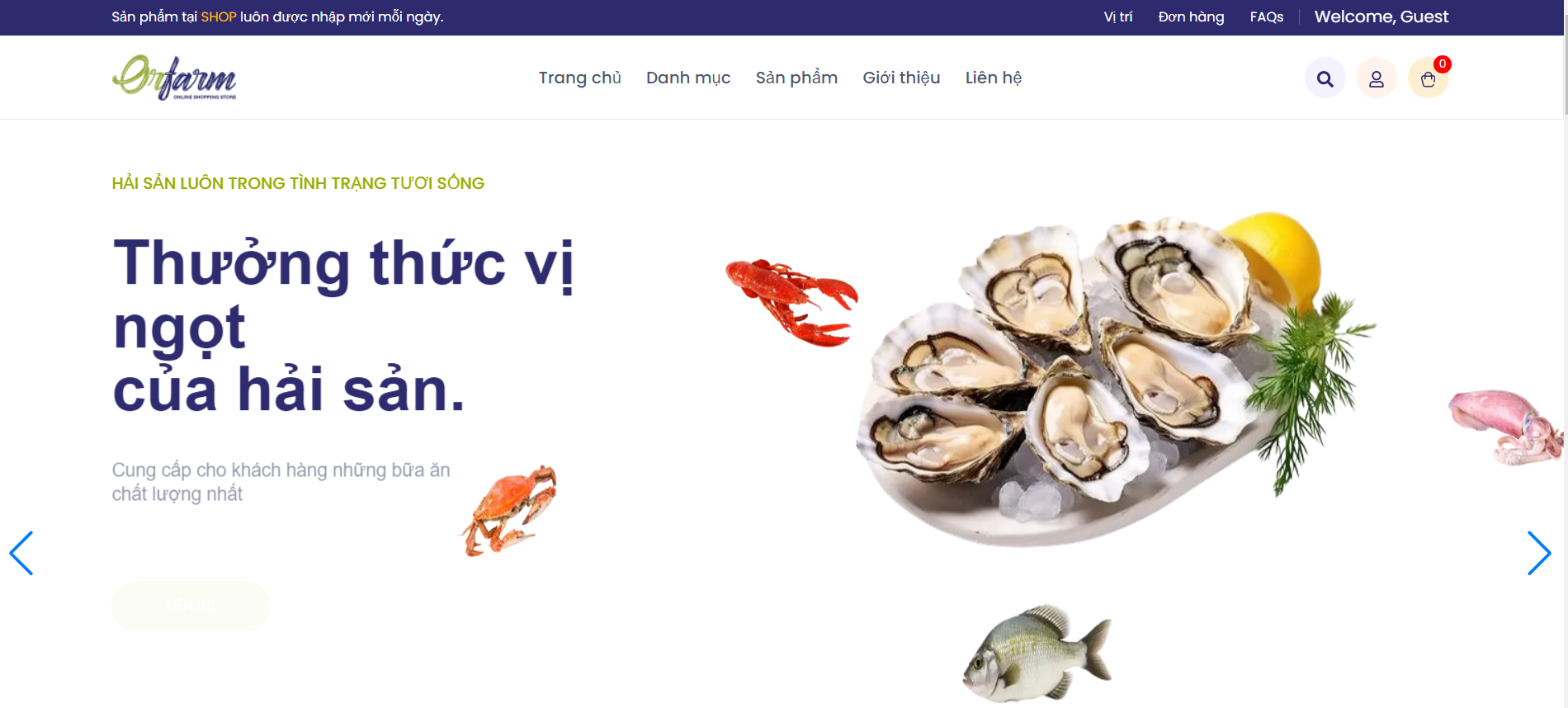


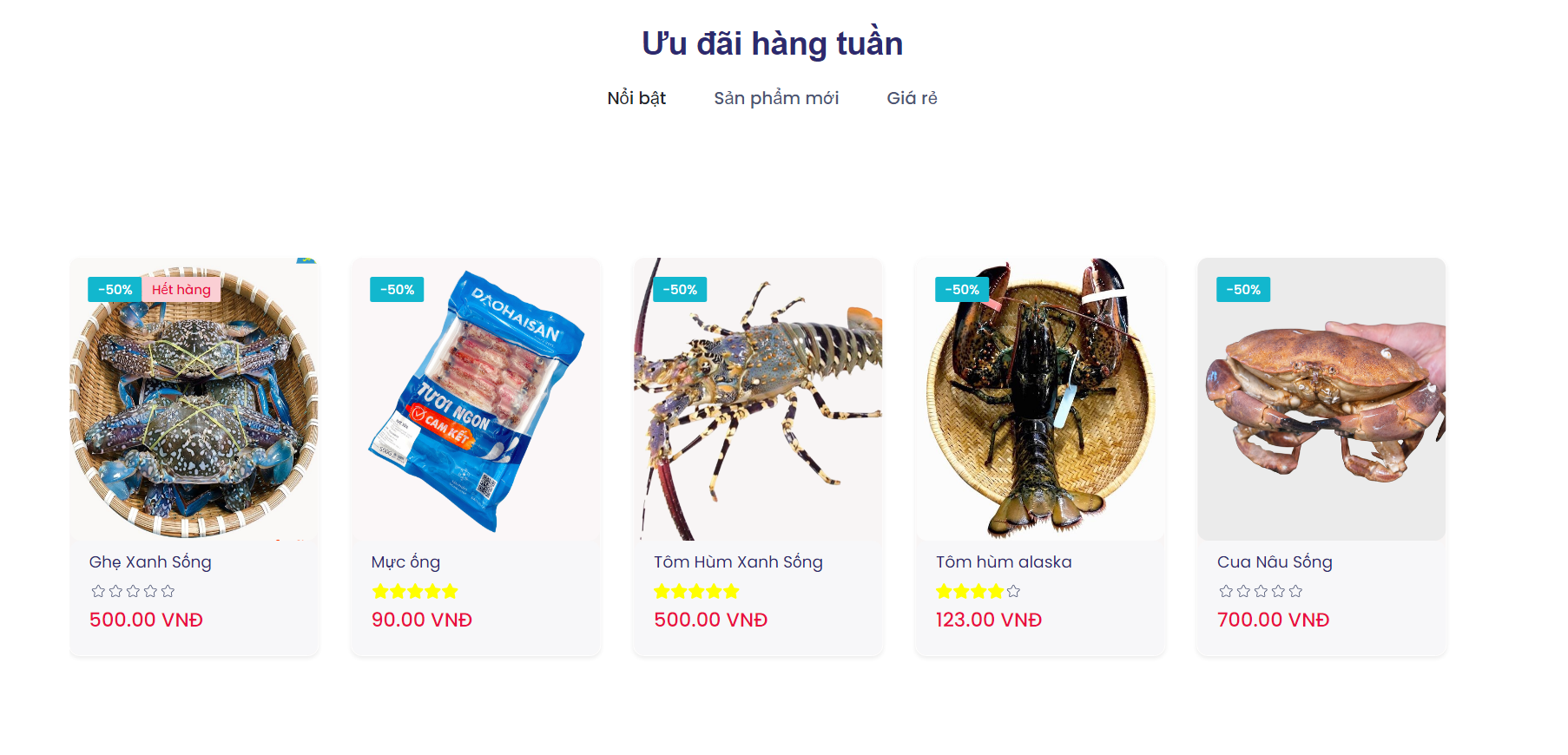
Hàm sendMail được định nghĩa để gửi email với nội dung HTML, tiêu đề và địa chỉ email người nhận được cung cấp. Việc sử dụng express-async-handler để xử lý lỗi bất đồng bộ một cách gọn gàng.



Hàm createOrder tạo một đơn hàng mới bằng cách nhận thông tin từ yêu cầu, kiểm tra và cập nhật số lượng sản phẩm trong kho. Nếu tất cả các bước thành công, nó trả về phản hồi với thông báo thành công. Nếu có lỗi trong quá trình, lỗi sẽ được xử lý bởi asyncHandler.

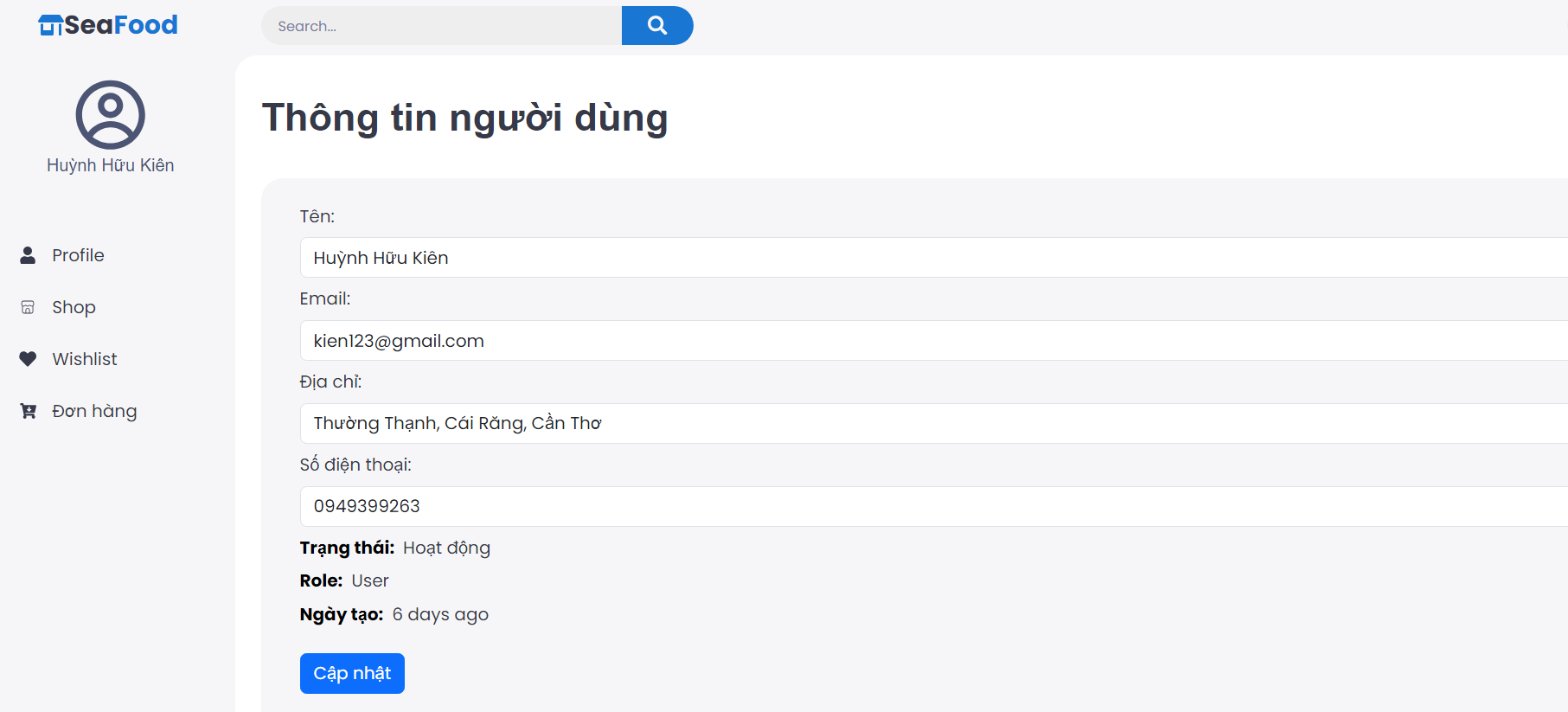
* 1. **Kết quả từ phía người sử dụng**
     1. **Giao diện trang chủ**

****

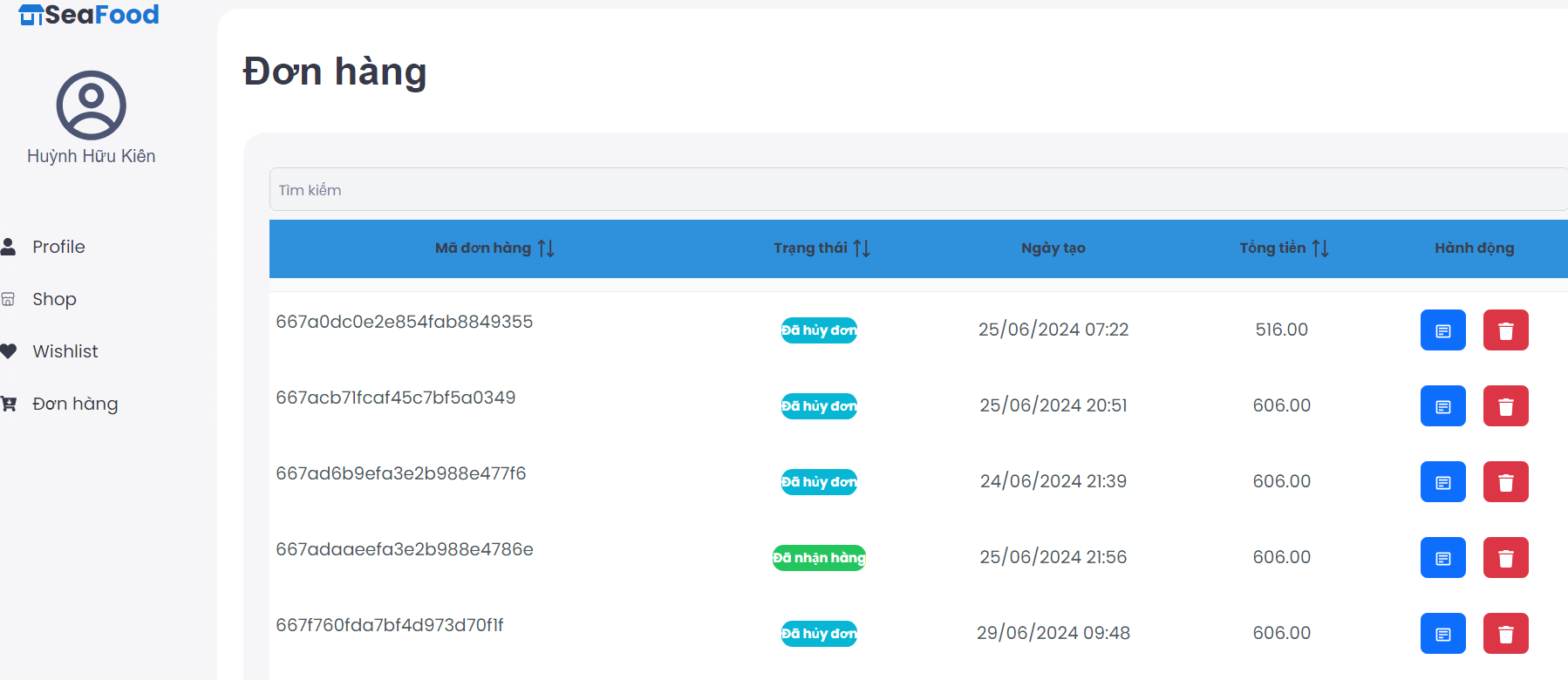
****

Hình 1. 1 Giao diện trang chủ

* + 1. **Giao diện trang người dùng**

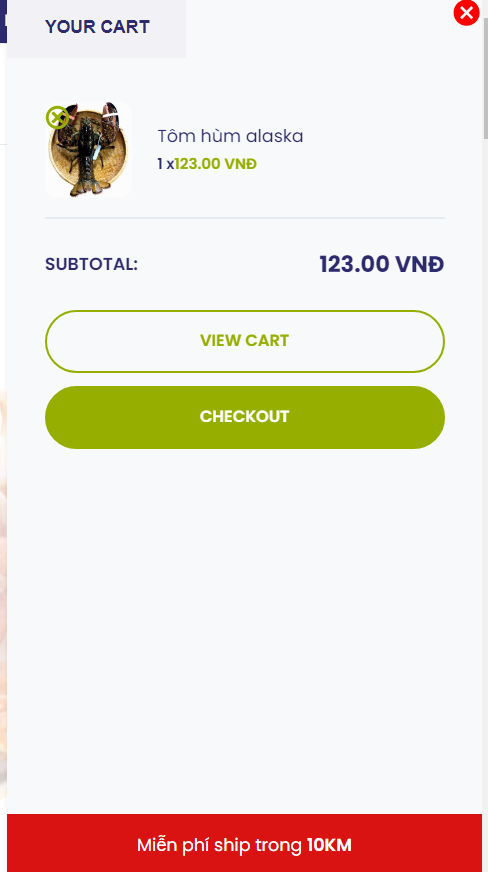
****

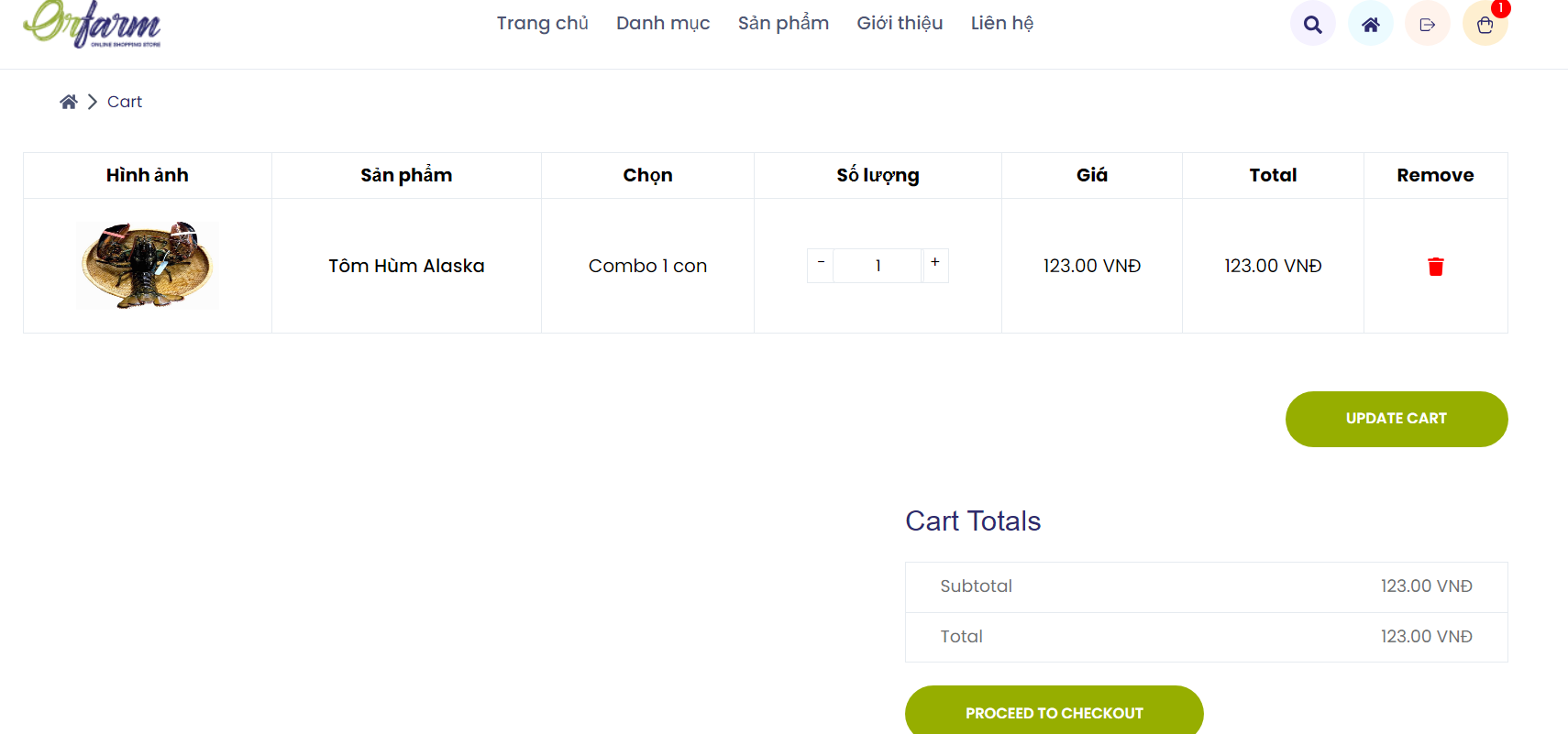
Hình 1. 2 Giao diện trang người dùng

****

Hình 1. 3 Giao diện quản lý đơn hàng

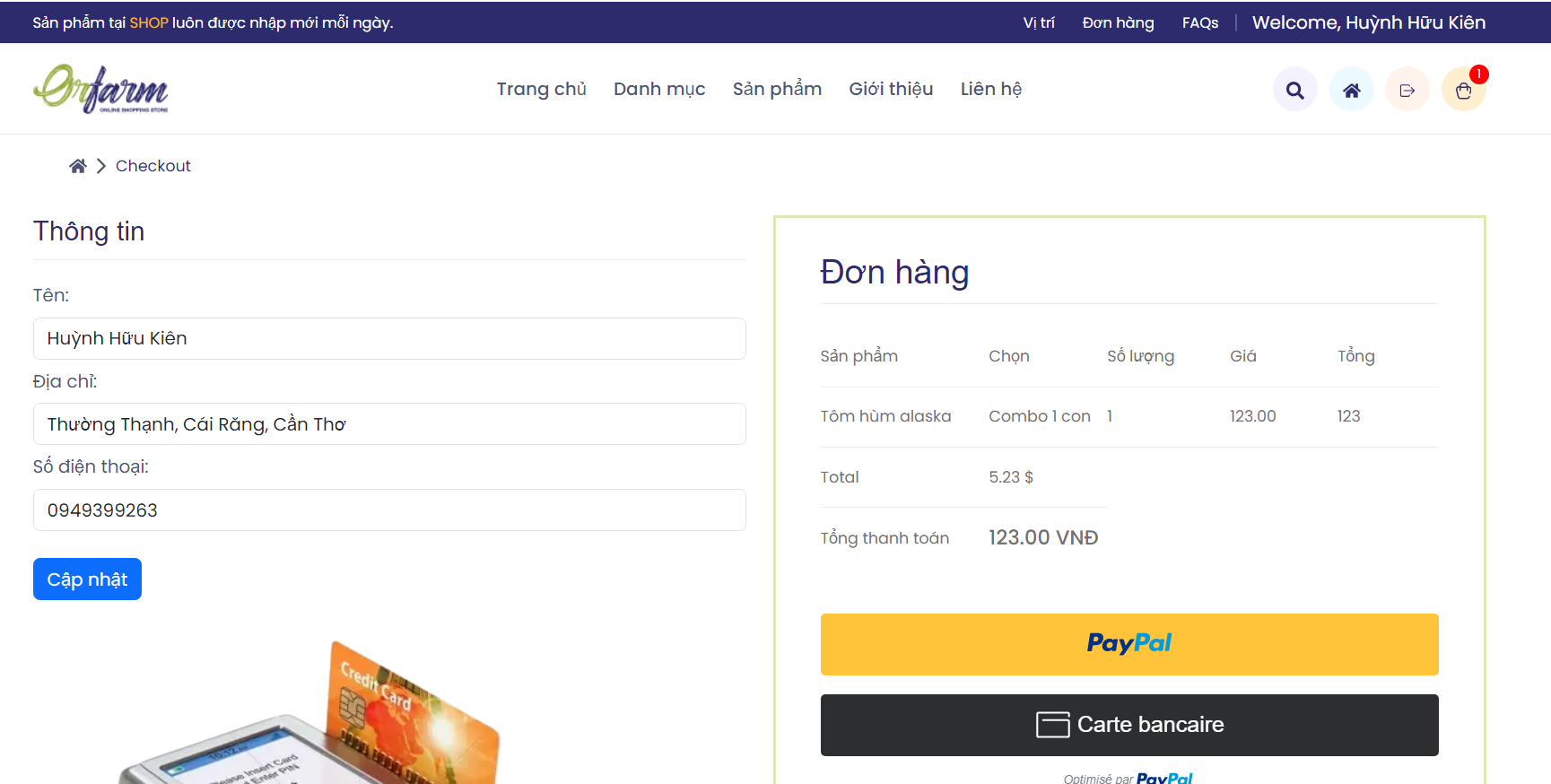
* + 1. **Giao diện giỏ hàng**

****

****

Hình 1. 4 Giao diện giỏ hàng

* + 1. **Giao diện thanh toán**



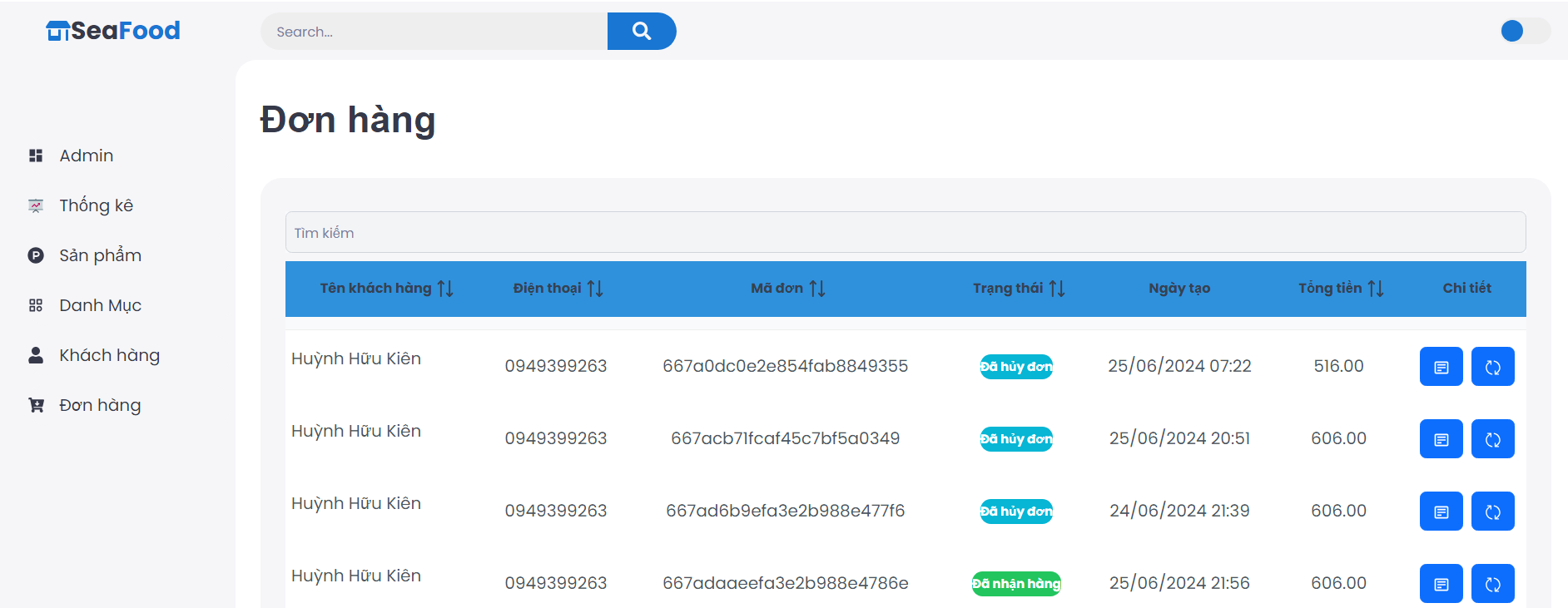
Hình 1. 5 Giao diện trang thanh toán

* + 1. **Giao diện trang quản trị**



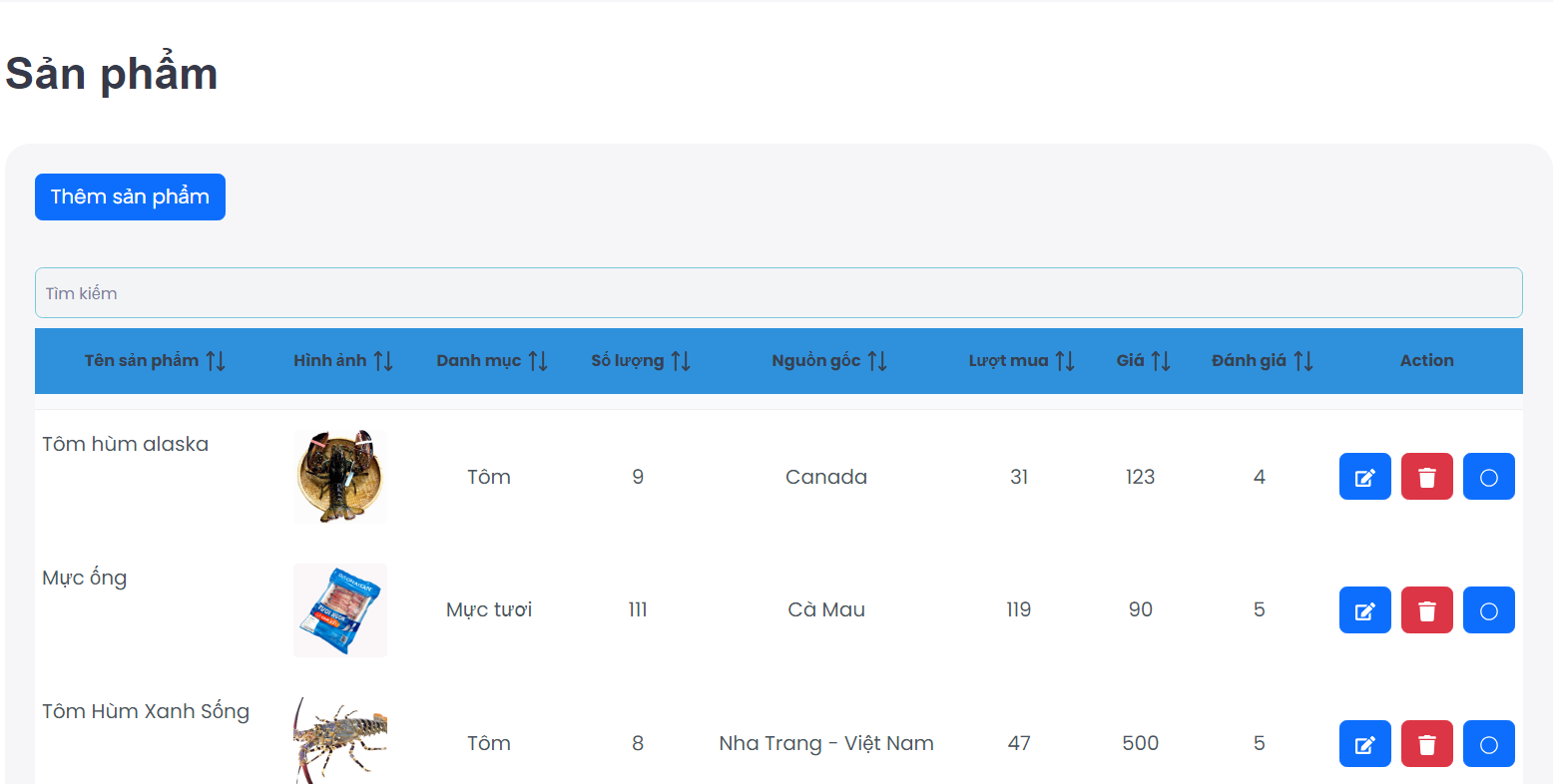
Hình 1. 6 Giao diện trang quản trị

* + 1. **Giao diện quản lý đơn hàng**



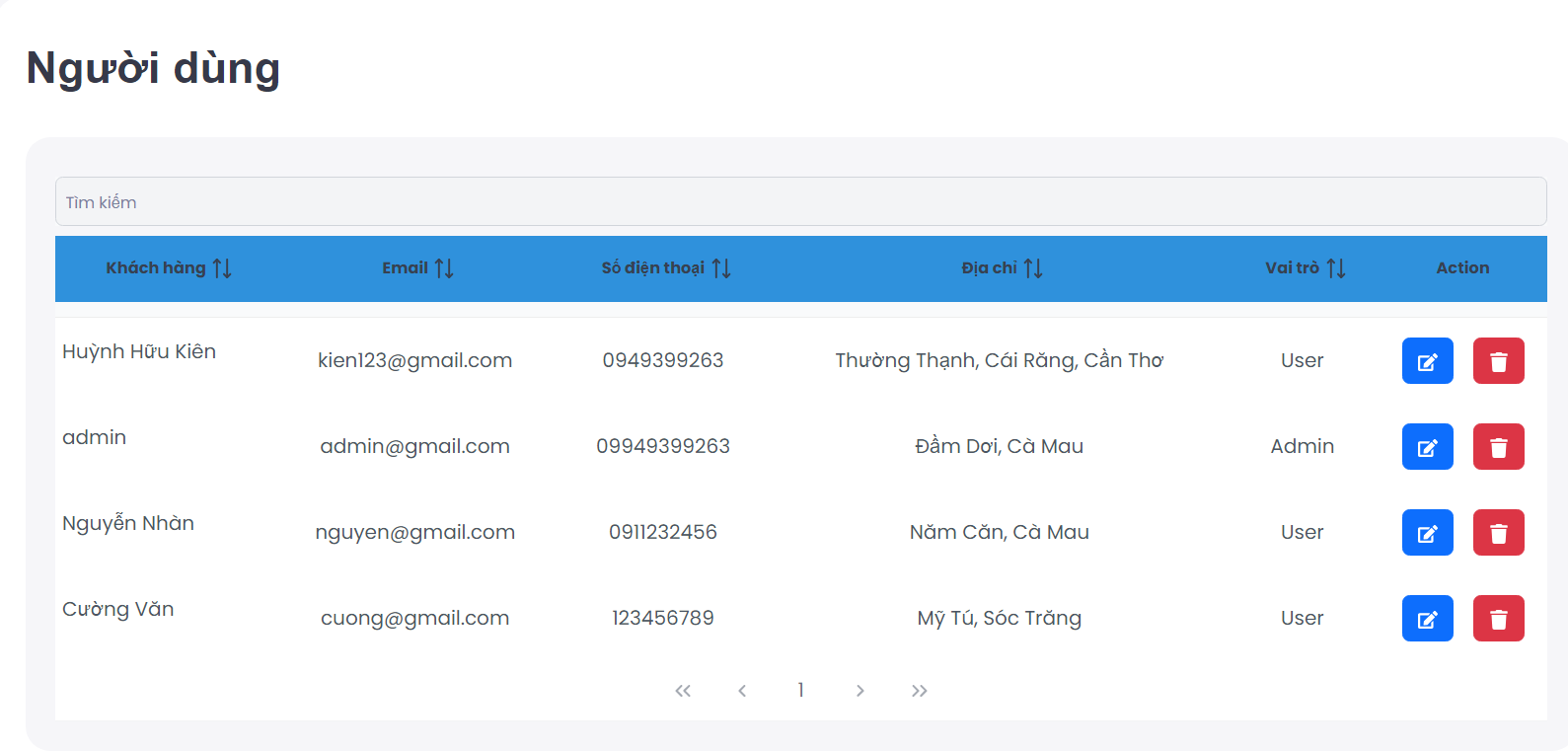
Hình 1. 7 Giao diện trang quản lý đơn hàng

* + 1. **Giao diện trang quản lý sản phẩm**



Hình 1. 8 Giao diện trang quản lý sản phẩm

* + 1. **Giao diện trang quản lý người dùng**



Hình 1. 9 Giao diện trang quản lý người dùng

# CHƯƠNG 4: KẾT QUẢ ĐẠT ĐƯỢC

# Những nội dung được củng cố

* + - Lập trình web frontend và backend: Hiểu về các nguyên lý cơ bản của HTML, CSS và JavaScript, áp dụng Next.js để xây dựng giao diện người dùng tương tác.
    - Quản lý dữ liệu với cơ sở dữ liệu NoSQL: Sử dụng MongoDB để lưu trữ dữ liệu sản phẩm, đơn hàng, người dùng các thông tin liên quan khác.
    - Phát triển API và backend với Node.js và Express.js: Xây dựn và triển khai các API để thực hiện các chức năng quản lý như đặt hàng, thanh toán, quản lý đơn hàng, quản lý sản phẩm và nhiều chức năng liên quan khác.
    - Phát triển giao diện người dùng: Thiết kế giao diện thân thiện để người dùng có thể dễ dàng sử dụng.

# Kinh nghiệm được tích lũy

## **4.2.1. Kỹ Thuật và Lập Trình**

**Front-end Development:**

* **Sử dụng các công nghệ hiện đại**: Thành thạo trong việc sử dụng các công nghệ như HTML, CSS, JavaScript, và các framework front-end như React.js để xây dựng giao diện người dùng.
* **Tối ưu hóa giao diện**: Hiểu cách tối ưu hóa giao diện người dùng để mang lại trải nghiệm tốt nhất, bao gồm việc đảm bảo trang web responsive trên các thiết bị khác nhau.

**Back-end Development:**

* **Quản lý dữ liệu**: Kỹ năng làm việc với các cơ sở dữ liệu như MongoDB, MySQL để quản lý dữ liệu sản phẩm, đơn hàng, và người dùng.
* **Xây dựng API**: Hiểu biết về RESTful API và GraphQL để tạo các dịch vụ web cho việc giao tiếp giữa client và server.
* **Sử dụng Node.js và Express**: Thành thạo trong việc sử dụng Node.js và Express.js để xây dựng các ứng dụng server-side mạnh mẽ.

## **4.2.2. Kiến Thức Về Thương Mại Điện Tử**

* **Hiểu biết về quy trình mua hàng**: Nắm vững các bước từ khi người dùng tìm kiếm sản phẩm, thêm vào giỏ hàng, đến khi hoàn tất thanh toán.
* **Quản lý sản phẩm**: Kỹ năng trong việc quản lý danh sách sản phẩm, bao gồm việc thêm, xóa, và cập nhật thông tin sản phẩm.
* **Quản lý đơn hàng**: Hiểu biết về cách xử lý đơn hàng từ lúc khách hàng đặt hàng đến khi giao hàng thành công.

## **4.2.3. Kỹ Năng Quản Lý Dự Án**

* **Phân chia công việc**: Kỹ năng phân chia công việc hiệu quả theo từn khoản thời gian cụ thể.
* **Giao tiếp**: Cải thiện kỹ năng giao tiếp và làm việc thông qua các cuộc họp, bàn luận về dự án

## **4.2.4. Kỹ Năng Giải Quyết Vấn Đề**

* **Debug và xử lý lỗi**: Kỹ năng tìm kiếm và khắc phục lỗi trong quá trình phát triển ứng dụng, bao gồm cả lỗi front-end và back-end.
* **Tìm kiếm và học hỏi**: Khả năng tìm kiếm thông tin và học hỏi các công nghệ mới, cũng như áp dụng vào dự án một cách hiệu quả.

# Kết quả công việc đã đóng góp cho cơ quan nơi thực tập

* + - Xây dựng thành công website bán hải sản, cung cấp các tính năng quan trọng như đặt hàng, quản lý sản phẩm và khách hàng.
    - Đóng góp vào việc phát triển kỹ năng lập trình, quản lý dự án và giải quyết vấn đề trong một môi trường thực tế.

**PHIẾU ĐÁNH GIÁ BÁO CÁO KẾT QUẢ THỰC TẬP** M-TT-04

**HỌC KỲ hè – 2023-2024**

**(Dùng cho giáo viên chấm báo cáo thực tập)**

Họ và tên cán bộ chấm báo cáo:………………………………………………………..

Họ tên sinh viên thực tập:………………………………………Mã số SV:…………...

|  |  |  |
| --- | --- | --- |
| **Nội dung đánh giá** | **Điểm tối đa** | **Điểm chấm** |
| **I. Hình thức trình bày** | **0.5** |  |
| I.1 Đúng format của khoa (Trang bìa, trang lời cảm ơn, trang đánh giá thực tập của khoa, trang mục lục và các nội dung báo cáo). Sử dụng đúng mã và font tiếng Việt (Unicode Times New  Roman, Size 13) | 0.25 |  |
| I.2 Trình bày mạch lạc, súc tích, không có lỗi chính tả | 0.25 |  |
| **II. Phiếu theo dõi** | **0.5** |  |
| II.1 Có lịch làm việc đầy đủ cho 8 tuần | 0.25 |  |
| II.2 Số buổi thực tập tại cơ quan trong 1 tuần >=6; ít hơn 6 buổi  0.0 điểm | 0.25 |  |
| **III. Đánh giá của công ty** | **8.0** |  |
| Cách tính điểm = (Điểm cộng của cán bộ hướng dẫn/100) x 8.0 | 8.0 |  |
| **IV. Nội dung thực tập (quyển báo cáo)** | **1.0** |  |
| * Có được sự hiểu biết tốt về cơ quan nơi thực tập. * Phương pháp thực hiện phù hợp với nội dung công việc được giao. * Kết quả củng cố lý thuyết. * Kết quả rèn luyện kỹ năng thực hành. * Kinh nghiệm thực tiễn thu nhận được. * Kết quả công việc có đóng góp cho cơ quan nơi thực tập | 0.5 |  |
| **TỔNG CỘNG** | **10.0** |  |
| Điểm trừ |  |  |
| **Điểm còn lại** |  |  |

………….., ngày….tháng… năm2024

GIÁO VIÊN CHẤM BÁO CÁO

*(ký tên)*