**TRƯỜNG ĐẠI HỌC AN GIANG  
KHOA CÔNG NGHỆ THÔNG TIN**

****

**ĐỒ ÁN CÔNG NGHỆ WEB PHP**

**Đề Tài**

**XÂY DỰNG TRANG WEB KINH DOANH**

**THIẾT BỊ IOT BẰNG FRAMWORK LARAVEL**

**Giảng viên hướng dẫn Sinh viên thực hiện**

Ths.Nguyễn Hoàng Tùng DTH225650 – Huỳnh Quốc Huy

**An Giang, tháng 11 năm 2025**

.............................................................................................................

**Giảng viên hướng dẫn**

( Ký và ghi rõ họ tên)

Nội dung nhận xét:

* **Đồng ý** hay **Không đồng ý**  cho sinh viên báo cáo; Nếu không đồng ý cần ghi rõ lý do.
* Kết quả được so với yêu cầu;

Ý kiến khác (nếu có)

**MỤC LỤC**

[**1.** **GIỚI THIỆU TỔNG QUAN ĐỀ TÀI** 6](#_Toc216741868)

[**1.1** **Giới Thiệu Đề Tài** 6](#_Toc216741869)

[**1.1.1** **Lý do chọn đề tài** 6](#_Toc216741870)

[**1.1.2** **Mục tiêu và kết quả** 7](#_Toc216741871)

[**1.1.3** **Phạm vi nghiên cứu của đồ án** 7](#_Toc216741872)

[**1.2** **Cơ Sở Lý Thuyết** 8](#_Toc216741873)

[**1.2.1** **Tổng quan về Hệ thống Internet of Thing (IoT)** 8](#_Toc216741874)

[**1.2.2** **Giới thiệu về Framework Laravel và mô hình MVC** 8](#_Toc216741875)

[**1.2.3** **Các công nghệ và thư viện hỗ trợ** 8](#_Toc216741876)

[**2.** **PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG** 9](#_Toc216741877)

[**2.1** **Phân Tích Yêu Cầu Hệ Thống** 9](#_Toc216741878)

[**2.1.1** **Phân tích yêu cầu chức năng** 9](#_Toc216741879)

[**2.1.2** **Phân tích yêu cầu phi chức năng** 10](#_Toc216741880)

[**2.2** **Thiết Kế Chức Năng** 11](#_Toc216741881)

[**2.2.1** **Sơ đồ Use Case** 11](#_Toc216741882)

[**2.2.2** **Đặc tả các Use Case chính** 12](#_Toc216741883)

[**2.3** **Thiết Kế Dữ Liệu** 13](#_Toc216741884)

[**2.3.1** **Sơ đồ quan hệ Database** 13](#_Toc216741885)

[**2.3.2** **Mô tả các bảng trong Database** 13](#_Toc216741886)

[**2.4** **Thiết Kế Cấu Trúc Hệ Thống** 14](#_Toc216741887)

[**2.4.1** **Sơ đồ Class** 14](#_Toc216741888)

[**2.4.2** **Thiết kết kiến trúc Laravel** 15](#_Toc216741889)

[**3.** **THIẾT KẾ VÀ TRIỂN KHAI HỆ THỐNG** 16](#_Toc216741890)

[**3.1** **Thiết Kế Giao Diện** 16](#_Toc216741891)

[**3.1.1** **Giao diện người dùng** 16](#_Toc216741892)

[**3.1.2** **Thiết kết giao diện quản lý** 18](#_Toc216741893)

[**3.2** **Thiết Kế Dữ Liệu Và Xử Lý** 18](#_Toc216741894)

[**3.2.1** **Chi tiết thiết kế dữ liệu** 19](#_Toc216741895)

[**3.2.2** **Chi tiết thiết kế xử lý** 19](#_Toc216741896)

[**3.3** **Triển khai chức năng** 20](#_Toc216741897)

[**3.3.1** **Module Quản lý Sản phẩm, Danh Mục, Nhà Sản Xuất** 20](#_Toc216741898)

[**3.3.2** **Module Giỏ hàng và Thanh toán** 22](#_Toc216741899)

[**3.3.3** **Module Quản lý Đơn hàng** 23](#_Toc216741900)

[**3.3.4** **Module Quản lý Người dùng và Phân quyền** 24](#_Toc216741901)

[**3.3.5** **Module Giám sát Thiết bị IoT** 25](#_Toc216741902)

[**3.4** **Hướng Dẫn Đăng Nhập Và Sử Dụng Hệ Thống** 26](#_Toc216741903)

[**4.** **TỔNG KẾT VÀ HƯỚNG PHÁT TRIỂN** 28](#_Toc216741904)

[**4.1** **Kết Quả Đạt Được** 28](#_Toc216741905)

[**4.2** **Đánh giá ưu nhược điểm đồ án** 29](#_Toc216741906)

[**4.2.1** **Ưu điểm** 29](#_Toc216741907)

[**4.2.2** **Nhược điểm** 29](#_Toc216741908)

[**4.3** **Hướng phát triển** 30](#_Toc216741909)

[**4.** **TÀI LIỆU THAM KHẢO** 31](#_Toc216741910)

**DANH SÁCH BẢNG**

[**Bảng 1: Bảng Use Case đặt hàng** 12](#_Toc216735596)

[**Bảng 2: Bảng mô tả Database MySQL** 14](#_Toc216735597)

**DANH SÁCH HÌNH**

[**Hình 1: Ảnh minh hoạt Cách mạng Công nghiệp 4.0** 6](#_Toc216748709)

[**Hình 2: Sơ đồ UseCase của hệ thống GreenTech** 11](#_Toc216748710)

[**Hình 3: Sơ đồ Entity Relationship của hệ thống GreenTech** 13](#_Toc216748711)

[**Hình 4: Sơ đồ Class của hệ thống GreenTech** 15](#_Toc216748712)

[**Hình 5: Giao diện đăng nhập** 17](#_Toc216748713)

[**Hình 6: Giao diện frontend** 17](#_Toc216748714)

[**Hình 7: Giao diện sản phẩm Frontend** 18](#_Toc216748715)

[**Hình 8: Giao diện tìm kiếm sản phẩm** 18](#_Toc216748716)

[**Hình 9:Chi tiết sản phẩm** 19](#_Toc216748717)

[**Hình 10: Giao diện dashboard của Administrator** 20](#_Toc216748718)

[**Hình 11: Giao diện quản lý sản phẩm** 23](#_Toc216748719)

[**Hình 12: Giao diện quản lý danh mục** 23](#_Toc216748720)

[**Hình 13: Giao diện quản lý nhà sản xuất** 24](#_Toc216748721)

[**Hình 14: Giao diện giỏ hàng** 25](#_Toc216748722)

[**Hình 15: Giao diện thanh toán** 25](#_Toc216748723)

[**Hình 16: Giao diện quản lý phân quyền** 27](#_Toc216748724)

[**Hình 17: Giao diện quản lý người dùng** 27](#_Toc216748725)

[**Hình 18: Thiết lập số liệu thiết bị IoT** 28](#_Toc216748726)

[**Hình 19: Thiết lập ngưỡng cảnh báo thiết bị IoT** 28](#_Toc216748727)

[**Hình 20: Tạo database greentech-commerce trên phpmyadmin** 29](#_Toc216748728)

[**Hình 21: Nội dung thư mục ImportFiles** 29](#_Toc216748729)

[**Hình 22: Nội dung file role.sql** 30](#_Toc216748730)

[**Hình 23: Giao diện register** 30](#_Toc216748731)

1. **GIỚI THIỆU TỔNG QUAN ĐỀ TÀI**
   1. **Giới Thiệu Đề Tài**
      1. **Lý do chọn đề tài**

Trong bối cảnh toàn cầu đang bước vào giai đoạn Cách mạng Công nghiệp 4.0, Internet of Things (IoT) đã nhanh chóng trở thành một xu hướng công nghệ chủ đạo, mang lại những thay đổi lớn trong các lĩnh vực từ nhà thông minh, nông nghiệp chính xác đến công nghiệp sản xuất.[1] Sự phát triển này kéo theo nhu cầu mua sắm và trao đổi các loại thiết bị IoT chuyên biệt như cảm biến, vi điều khiển, mô-đun truyền động ngày càng tăng cao, đòi hỏi một nền tảng thương mại điện tử phải chuyên biệt hóa.



**Hình 1: Ảnh minh hoạt Cách mạng Công nghiệp 4.0**

Do đó một trang web để kinh doanh thiết bị IoT là một ý tưởng hiệu quả để đáp ứng những vấn đề cần giải quyết hiện tại, với đề tài mộn học trang web này được xây dựng bằng Framework Laravel . Đầu tiên do thị trường thiết bị IoT tại Việt Nam đang có tốc độ phát triển mạnh mẽ. Tuy nhiên, vẫn còn thiếu các nền tảng E-commerce chuyên dụng, nơi người dùng không chỉ thực hiện giao dịch mua bán mà còn có thể tra cứu thông tin kỹ thuật chuyên sâu và tìm hiểu khả năng tích hợp của sản phẩm. Bên cạnh đó một hệ thống kinh doanh thiết bị IoT cần vượt qua các chức năng bán hàng truyền thống. Hệ thống phải có khả năng mô tả và phân loại chi tiết các thông số kỹ thuật, và quan trọng hơn, cần tích hợp các tính năng mô phỏng hoặc quản lý dữ liệu thiết bị và cấu hình ngưỡng cảnh báo an toàn để làm nổi bật giá trị cốt lõi của sản phẩm IoT. Như một kết quả tất yếu Framework Laravel được lựa chọn vì tính ổn định, hệ sinh thái phong phú và việc tuân thủ nghiêm ngặt mô hình MVC. Đây là nền tảng lý tưởng để xây dựng một ứng dụng phức tạp, bao gồm các module bán hàng, quản trị và giám sát IoT, đảm bảo tốc độ phát triển nhanh chóng và khả năng bảo trì lâu dài.

* + 1. **Mục tiêu và kết quả**

Đồ án hướng đến việc xây dựng hoàn chỉnh một hệ thống E-commerce trên nền tảng Laravel. Hệ thống cần triển khai đầy đủ các module giao dịch cốt lõi bao gồm quản lý sản phẩm, danh mục, giỏ hàng, thanh toán và quản lý đơn hàng, cho các đối tượng người dùng đa dạng (Khách hàng, Quản trị viên, Nhân viên Bán hàng, Giao hàng). Song song với đó, đồ án phát triển module quản lý thiết bị IoT chuyên biệt, cho phép mô hình hóa việc hiển thị các thông số đo được từ thiết bị (dữ liệu mẫu) và cung cấp giao diện để quản trị viên cấu hình ngưỡng cảnh báo. Tổng thể, mục tiêu là vận dụng thành thạo Framework Laravel và mô hình MVC để tạo ra một hệ thống có cấu trúc, dễ mở rộng, và đảm bảo tính bảo mật cao thông qua việc triển khai cơ chế phân quyền chi tiết.

Kết quả dự kiến đạt được là một trang web thương mại điện tử chuyên kinh doanh thiết bị IoT hoạt động ổn định, có giao diện hiện đại và thân thiện với người dùng. Hệ thống sẽ cung cấp một bảng điều khiển quản trị (Admin/Saler/Shipper) phân cấp, cho phép quản lý toàn diện các quy trình kinh doanh, từ quản lý kho hàng đến logistics. Đặc biệt, module IoT Device Management sẽ hoạt động như một bằng chứng về khả năng tích hợp, cho phép người dùng theo dõi các chỉ số quan trọng và nhận thông báo khi dữ liệu vượt ngưỡng cho phép

* + 1. **Phạm vi nghiên cứu của đồ án**

Phạm vi của đồ án được xác định chi tiết trên một số khía cạnh đặc trưng.Với chức năng đồ án tập trung vào hai nhóm chức năng chính. Nhóm chức năng dành cho Khách hàng bao gồm các quy trình cơ bản như đăng ký, đăng nhập, tìm kiếm, lọc, đặt hàng và theo dõi trạng thái đơn hàng. Nhóm chức năng Quản trị nội bộ bao gồm các thao tác Quản lý Dữ liệu (CRUD) đối với Products, Categories, Manufactures, Users, Roles, OrderStatuses và Quản lý chỉ số thiết bị IoT IoTDevice, DeviceMetric, AlertThreshold.

Về dữ liệu, hệ thống sẽ lưu trữ và xử lý các loại dữ liệu bao gồm: thông tin chi tiết sản phẩm (tên, giá, số lượng, mô tả,…và chi tiết kỹ thuật ProductDetail), hình ảnh (ProductImage), thông tin đơn hàng, thông tin người dùng, và các chỉ số đo lường mô phỏng của thiết bị IoT.

Phạm vi công nghệ của hệ thống được xây dựng trên nền tảng Backend là ngôn ngữ PHP và Framework Laravel, sử dụng hệ quản trị Database MySQL. Giao diện Frontend được xây dựng bằng HTML/CSS/JavaScript, sử dụng Blade Template Engine và tích hợp các thư viện giao diện Cartzilla/Bootstrap.

Giới hạn phạm vi đồ án chỉ tập trung vào việc mô phỏng các giao dịch thương mại và giao diện quản lý dữ liệu thiết bị IoT trên môi trường web. Đồ án không bao gồm việc thực hiện kết nối vật lý thời gian thực với các thiết bị IoT hoặc triển khai các tính năng phân tích dữ liệu chuyên sâu (như Machine Learning, Big Data).

* 1. **Cơ Sở Lý Thuyết**
     1. **Tổng quan về Hệ thống Internet of Thing (IoT)**

IoT được định nghĩa là một mạng lưới toàn cầu, nơi các đối tượng vật lý được nhúng các thành phần điện tử, phần mềm và khả năng kết nối mạng, cho phép chúng thu thập và trao đổi dữ liệu [2]. Việc hiểu rõ các thành phần này là nền tảng để xây dựng hệ thống thương mại điện tử chuyên biệt: Trong đồ án, thiết bị và cảm biến (Devices/Sensors) là các sản phẩm vật lý được bán trên website, từ bo mạch chủ (Arduino, Raspberry Pi, STM32…), cảm biến môi trường (DHT22, Light Sensor…), đến các mô-đun truyền động (actuator-relay, servo…). Các sản phẩm này được mô hình hóa trong Product model. Tiếp đên về kết nối (Connectivity) là cơ chế cần thiết cho thiết bị hoạt động (Wi-Fi, Zigbee, v.v.), được trình bày chi tiết trong mô tả kỹ thuật của sản phẩm. Bên cạnh đó, nền tảng đám mây (Platform/Cloud) chính là hệ thống web được xây dựng, đóng vai trò là nơi tiếp nhận, lưu trữ dữ liệu sử dụng DeviceMetric model để lưu trữ các chỉ số đo lường và là giao diện tương tác chính. Một phần Ứng dụng và Phân tích (Application & Analytics): Là các chức năng được triển khai trên website, cho phép người dùng hoặc quản trị viên xem dữ liệu, đặt các tham số hoạt động, và thiết lập ngưỡng cảnh báo (AlertThreshold) để ra quyết định dựa trên dữ liệu mô phỏng.

* + 1. **Giới thiệu về Framework Laravel và mô hình MVC**

Framework Laravel là nền tảng Back-end được chọn, là một PHP Framework nổi tiếng với cú pháp thanh lịch, tập trung vào tính năng và hiệu suất. Laravel cung cấp các công cụ mạnh mẽ như Eloquent ORM giúp tương tác với cơ sở dữ liệu một cách dễ dàng thông qua các đối tượng Model, và Blade Template Engine giúp xây dựng giao diện nhanh chóng.[3]

Toàn bộ đồ án được xây dựng theo Mô hình MVC (Model-View-Controller), một kiến trúc giúp phân tách rõ ràng các mối quan tâm trong ứng dụng [4]. Trong đó, Model chịu trách nhiệm về logic nghiệp vụ và thao tác với dữ liệu (các file trong thư mục app/Models). Kế đến, View sẽ xử lý hiển thị giao diện người dùng (các file .blade.php trong thư mục resources/views). Một phần quan trọng Controller đóng vai trò là cầu nối, tiếp nhận yêu cầu từ người dùng, gọi Model xử lý và điều hướng đến View phù hợp (các file trong thư mục app/Http/Controllers).

Việc áp dụng MVC là then chốt, giúp tổ chức mã nguồn khoa học, tách biệt giữa giao diện (View) và logic xử lý (Model, Controller), từ đó tối ưu hóa quá trình phát triển, mở rộng module và bảo trì hệ thống.

* + 1. **Các công nghệ và thư viện hỗ trợ**

Để đảm bảo khả năng mở rộng, tính ổn định và chất lượng sản phẩm cuối cùng, đồ án đã được phát triển dựa trên một hệ sinh thái công nghệ đa dạng. Về công nghệ cốt lõi, toàn bộ logic Back-end của ứng dụng được xây dựng trên ngôn ngữ lập trình PHP cùng Framework Laravel, trong khi JavaScript (ES6) được sử dụng cho các tương tác và logic xử lý ở phía Front-end. Mọi dữ liệu về sản phẩm, đơn hàng và quản lý thiết bị IoT đều được lưu trữ và quản lý bởi hệ quản trị cơ sở dữ liệu MySQL.

Về phát triển giao diện (Frontend), hệ thống áp dụng các tiêu chuẩn hiện đại như HTML5, CSS3, jQuery, và đặc biệt tích hợp bộ tài nguyên giao diện chuyên nghiệp Bootstrap/Cartzilla.[5] Sự kết hợp này nhằm đảm bảo giao diện người dùng và quản trị không chỉ hiện đại, bắt mắt mà còn có khả năng hiển thị responsive hoàn hảo trên mọi thiết bị.

Để tối ưu hóa các tác vụ quản trị và nghiệp vụ, một số thư viện mạnh mẽ của Laravel đã được tích hợp. Cụ thể, thư viện Maatwebsite/Laravel-Excel đóng vai trò quan trọng trong việc hỗ trợ các chức năng Import/Export dữ liệu quản lý (như danh sách Products, Categories), từ đó tối ưu hóa đáng kể quy trình nhập liệu và cập nhật dữ liệu [6]. Bên cạnh đó, trình soạn thảo trực quan CKEditor 5 được nhúng vào giao diện quản trị, giúp Admin dễ dàng soạn thảo mô tả sản phẩm với định dạng phong phú.[7]

Cuối cùng, các khía cạnh về truyền thông và bảo mật được xử lý nghiêm ngặt. Hệ thống sử dụng Laravel Mail để tự động xây dựng và gửi các email quan trọng đến khách hàng (điển hình là email xác nhận đặt hàng thông qua PlaceOrderSuccessEmail.php). Quan trọng nhất, tính năng bảo mật được tăng cường toàn diện thông qua các chức năng Authentication và cơ chế Authorization chặt chẽ của Laravel. Cơ chế này được triển khai chi tiết qua hệ thống Roles và RoleMiddleware, đảm bảo phân quyền truy cập rõ ràng, giới hạn quyền hạn cho từng nhóm người dùng (Admin, Saler, Shipper, Customer) vào các module tương ứng.

1. **PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**

Việc phân tích yêu cầu là bước nền tảng nhằm xác định chính xác những gì hệ thống cần làm để đáp ứng nhu cầu kinh doanh và người dùng, được chia thành yêu cầu chức năng và yêu cầu phi chức năng.

* 1. **Phân Tích Yêu Cầu Hệ Thống**
     1. **Phân tích yêu cầu chức năng**

Các chức năng của hệ thống được phân loại theo vai trò người dùng để đảm bảo tính rõ ràng và khả năng quản lý. Hệ thống phân quyền được xác định qua bảng roles và cột roles trong bảng users, bao gồm Administrator (1), Users (2 - Khách hàng), Saler (3 - Nhân viên Bán hàng), và Shipper (4 - Nhân viên Giao hàng).

Đầu tiên với nhóm chức năng dành cho Khách hàng (Users) là đối tượng tương tác với giao diện Front-end, với các nhu cầu chính là tìm kiếm, mua sắm và quản lý thông tin cá nhân. Hệ thống phải hỗ trợ người dùng đăng ký, đăng nhập và quản lý hồ sơ cá nhân. Quan trọng nhất, khách hàng phải dễ dàng tìm kiếm và duyệt sản phẩm thông qua các danh mục (categories) và nhà sản xuất (manufacturers), xem thông tin chi tiết kỹ thuật (product\_details). Quy trình mua hàng phải liền mạch, bao gồm thêm sản phẩm vào giỏ hàng, tiến hành thanh toán (tạo bản ghi trong bảng orders), và theo dõi trạng thái đơn hàng thông qua order\_statuses.

Thêm vào đó nhóm chức năng dành cho Quản trị viên (Administrator) và Bán hàng (Saler) là các module CRUD (Create, Read, Update, Delete) cho phép quản lý toàn diện cơ sở dữ liệu. Administrator và Saler có trách nhiệm quản lý thông tin sản phẩm (products), bao gồm giá (price), số lượng tồn kho (stock\_quantity), và hình ảnh (product\_images). Chức năng này cũng mở rộng sang việc quản lý chi tiết kỹ thuật chuyên biệt của thiết bị IoT qua bảng product\_details. Quản lý cấu trúc phân loại sản phẩm (categories) và thông tin đối tác (manufacturers). Xử lý, xác nhận, và cập nhật trạng thái đơn hàng (orders) từ Pending sang Confirmed hoặc Shipping (sử dụng order\_statuses). Chức năng này cũng bao gồm việc kiểm tra chi tiết từng mặt hàng trong đơn (order\_items).

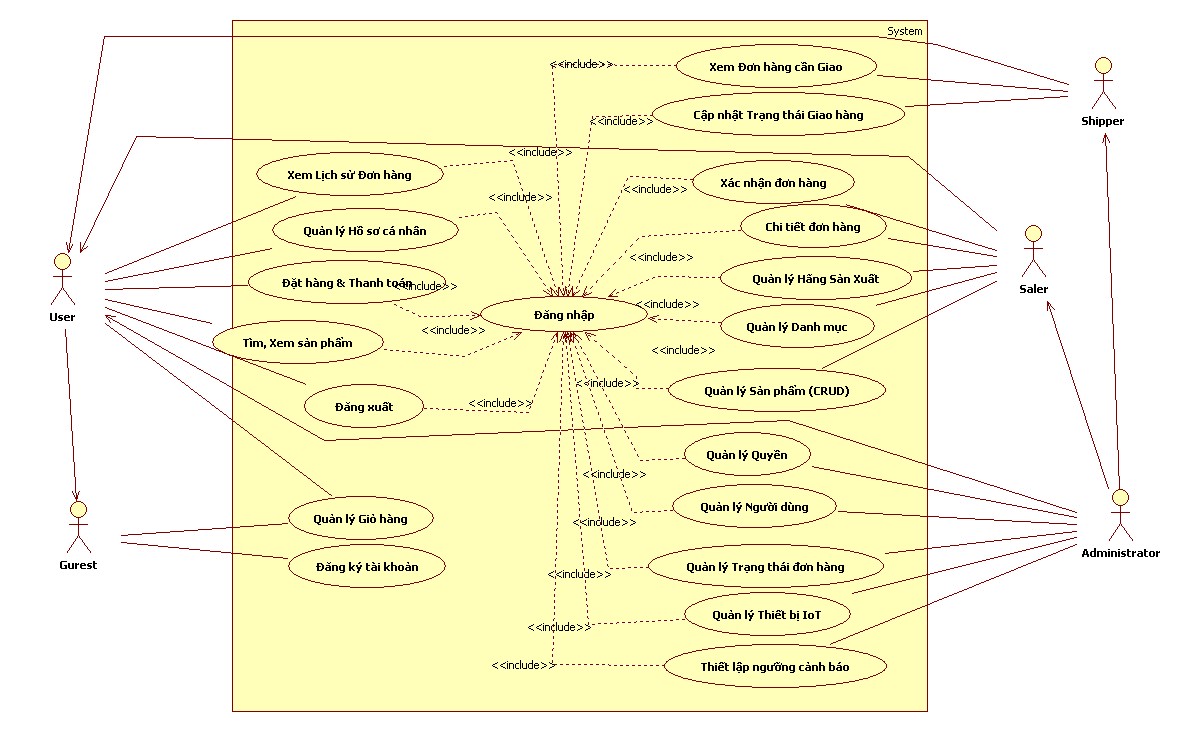
Ngoài ra nhóm chức năng Quản lý IoT chuyên biệt là điểm nhấn của đồ án, tích hợp khả năng quản lý các thiết bị IoT đã bán. Quản trị viên phải có khả năng thêm, quản lý các thiết bị IoT cụ thể (iot\_devices chẳng hạn một cảm biến đã được triển khai tại một location nào đó), định nghĩa các chỉ số đo lường (device\_metrics) và thiết lập các ngưỡng cảnh báo an toàn (alert\_thresholds với min\_value và max\_value) cho từng thiết bị.

* + 1. **Phân tích yêu cầu phi chức năng**

Các yêu cầu phi chức năng đảm bảo chất lượng, hiệu suất và độ tin cậy của hệ thống. Về bảo mật (Security) đây là yêu cầu tiên quyết. Hệ thống sử dụng cơ chế xác thực và phân quyền đa cấp của Laravel, được kiểm soát chặt chẽ bởi RoleMiddleware để ngăn chặn truy cập trái phép vào các khu vực quản trị. Mật khẩu người dùng phải được mã hóa, và các biện pháp chống tấn công cơ bản (như Cross-Site Scripting, SQL Injection) phải được framework xử lý. Thêm vào đó hiệu suất (Performance) phải có tốc độ tải nhanh, đặc biệt là trang chủ và trang danh mục sản phẩm, bất chấp việc phải xử lý lượng lớn hình ảnh sản phẩm và dữ liệu phức tạp. Việc sử dụng Laravel Eloquent ORM cần được tối ưu hóa để giảm thiểu số lượng truy vấn database. Thêm vào đó khả năng sử dụng (Usability) được đáp ứng qua giao diện phải trực quan và thân thiện. Front-end cần tuân thủ thiết kế E-commerce hiện đại (sử dụng thư viện Cartzilla/Bootstrap), đảm bảo trải nghiệm mua sắm mượt mà trên cả máy tính và thiết bị di động (Responsive Design). Giao diện quản trị phải rõ ràng, dễ dàng thao tác CRUD và tra cứu dữ liệu. Ngoài ra là khả năng mở rộng (Scalability) của hệ thống được thiết kế trên mô hình MVC để dễ dàng thêm các chức năng mới (như cổng thanh toán khác, module báo cáo nâng cao) mà không làm ảnh hưởng đến các module hiện có..

* 1. **Thiết Kế Chức Năng**
     1. **Sơ đồ Use Case**

Sơ đồ Use Case mô tả mối quan hệ giữa các tác nhân (Actors) và các chức năng chính của hệ thống. Trong đó tác nhân: Khách hàng (Users), Quản trị viên (Administrator), Nhân viên Bán hàng (Saler), Nhân viên Giao hàng (Shipper).



**Hình 2: Sơ đồ UseCase của hệ thống GreenTech**

Các Use Case chính trong hệ thống được thiết kế như sau:

Khách hà ng: Đăng nhập/Đăng ký, Xem Sản phẩm, Tìm kiếm Sản phẩm, Quản lý Giỏ hàng, Đặt hàng (OrdersController), Xem Lịch sử Đơn hàng, Quản lý Hồ sơ cá nhân.

Quản trị viên: Quản lý Toàn bộ Hệ thống (Users, Roles, Categories, Manufacturers, Products, Orders), Quản lý Cấu hình IoT Device.

Nhân viên Bán hàng (Saler): Quản lý Sản phẩm, Quản lý Danh mục, Quản lý Đơn hàng (Xác nhận, Cập nhật trạng thái).

Nhân viên Giao hàng (Shipper): Xem danh sách đơn hàng cần giao (trạng thái Shipping), Cập nhật trạng thái giao hàng (Đã giao/Hủy).

* + 1. **Đặc tả các Use Case chính**

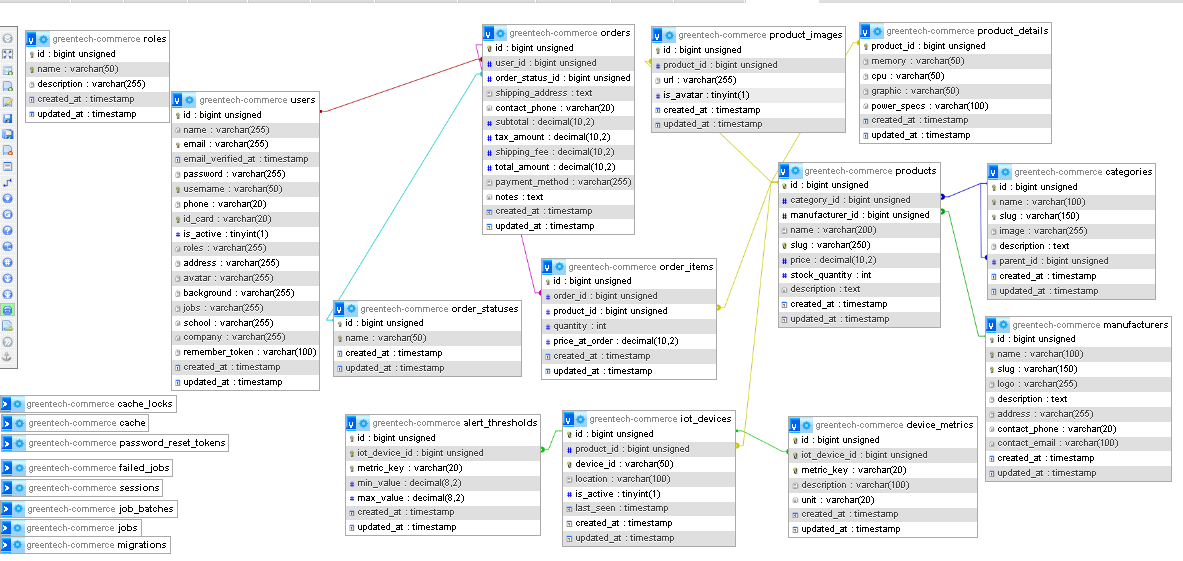
Hệ thống có quá trình Đặt hàng là một UseCase cốt lõi. Đây là bảng mô tả xử lý cho Use Case này

**Bảng 1: Bảng Use Case đặt hàng**

| **Thông tin** | **Nội dung** |
| --- | --- |
| **Tên Use Case** | Đặt hàng và Thanh toán |
| **Tác nhân** | Khách hàng |
| **Mục đích** | Cho phép Khách hàng hoàn tất giao dịch mua sản phẩm và tạo bản ghi đơn hàng. |
| **Tiền điều kiện** | Khách hàng đã đăng nhập và Giỏ hàng có ít nhất một sản phẩm. |
| **Luồng sự kiện chính** | 1. Khách hàng truy cập trang thanh toán (Checkout).  2. Hệ thống tải thông tin Khách hàng (users.address, users.phone,…).  3. Khách hàng điền/xác nhận địa chỉ giao hàng và thông tin liên hệ.  4. Khách hàng chọn phương thức thanh toán (payment\_method).  5. Khách hàng xác nhận đặt hàng.  6. Hệ thống thực hiện giao dịch: Tạo một bản ghi mới trong bảng orders (bao gồm user\_id, order\_status\_id ban đầu là Pending, total\_amount), Tạo các bản ghi trong bảng order\_items cho từng sản phẩm trong giỏ hàng (lưu lại product\_id, quantity, và price\_at\_order). Cập nhật giảm số lượng tồn kho (stock\_quantity) của các sản phẩm liên quan. Gửi email xác nhận đơn hàng (sử dụng PlaceOrderSuccessEmail.php).  7. Hệ thống chuyển hướng Khách hàng đến trang Order Success. |
| **Hậu điều kiện** | Một bản ghi đơn hàng mới được tạo trong database, tồn kho được cập nhật, và email xác nhận được gửi đi. |

* 1. **Thiết Kế Dữ Liệu**
     1. **Sơ đồ quan hệ Database**

Cơ sở dữ liệu được xây dựng xoay quanh các thực thể chính là Người dùng, Sản phẩm, Đơn hàng và Thiết bị IoT, thiết lập các mối quan hệ đa dạng.



**Hình 3: Sơ đồ Entity Relationship của hệ thống GreenTech**

Mối quan hệ 1-nhiều (One-to-Many): categories (id) và manufacturers (id) đều liên kết tới products (category\_id, manufacturer\_id), cho phép một danh mục hoặc nhà sản xuất có nhiều sản phẩm. users (id) và order\_statuses (id) liên kết tới orders (user\_id, order\_status\_id), cho phép một người dùng có nhiều đơn hàng và một trạng thái được áp dụng cho nhiều đơn.

Mối quan hệ 1-1 (One-to-One): products (id) liên kết tới product\_details (product\_id), cho phép lưu trữ các thông số kỹ thuật chi tiết chỉ có duy nhất cho một sản phẩm.

Mối quan hệ phức hợp (Many-to-Many qua bảng trung gian): orders (id) và products (id) liên kết qua bảng trung gian order\_items, mô tả chi tiết số lượng (quantity) và giá tại thời điểm đặt hàng (price\_at\_order).

Mối quan hệ IoT chuyên biệt: products (id) có thể liên kết tới iot\_devices (product\_id), đại diện cho các thiết bị đã được triển khai. iot\_devices lại liên kết tới device\_metrics (định nghĩa các chỉ số: temperature, humidity) và alert\_thresholds (định nghĩa ngưỡng cảnh báo min/max cho từng chỉ số), tạo nên cấu trúc dữ liệu cho module giám sát thiết bị.

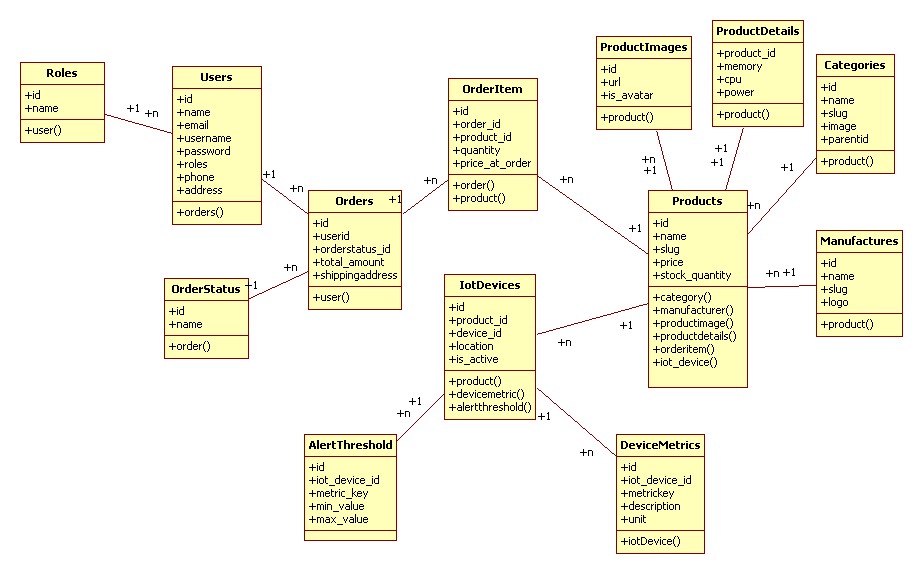
* + 1. **Mô tả các bảng trong Database**

**Bảng 2: Bảng mô tả Database MySQL**

| **Bảng** | **Mục đích** | **Các cột quan trọng và vai trò** |
| --- | --- | --- |
| **users** | Lưu trữ thông tin người dùng và phân quyền. | name, email, username, phone, roles (chứa ID vai trò: 1, 2, 3, 4), id\_card. |
| **roles** | Định nghĩa các vai trò hệ thống. | name (Administrator, Users, Saler, Shipper). |
| **products** | Thông tin sản phẩm cơ bản. | name, slug, price, stock\_quantity, category\_id, manufacturer\_id. |
| **product\_details** | Chi tiết kỹ thuật bổ sung (cho các thiết bị IoT/bo mạch). | product\_id (PK, FK), memory, cpu, power\_specs. |
| **orders** | Thông tin tổng quan của đơn hàng. | user\_id (FK), order\_status\_id (FK), total\_amount, shipping\_address, payment\_method. |
| **order\_items** | Chi tiết các mặt hàng trong từng đơn. | order\_id (FK), product\_id (FK), quantity, price\_at\_order. |
| **order\_statuses** | Định nghĩa trạng thái đơn hàng. | name (Pending, Shipping, Confirmed, Paid, Canceled). |
| **iot\_devices** | Quản lý các thiết bị IoT đã triển khai. | product\_id (FK), device\_id (ID duy nhất của thiết bị), location, is\_active. |
| **device\_metrics** | Định nghĩa các chỉ số mà thiết bị đo lường. | iot\_device\_id (FK), metric\_key (e.g., 'temperature'), unit. |
| **alert\_thresholds** | Thiết lập ngưỡng cảnh báo cho từng chỉ số. | iot\_device\_id (FK), metric\_key (cùng với iot\_device\_id là UNIQUE KEY), min\_value, max\_value. |

* 1. **Thiết Kế Cấu Trúc Hệ Thống**
     1. **Sơ đồ Class**

Hệ thống được thiết kế theo hướng đối tượng, với các thực thể trong cơ sở dữ liệu được ánh xạ thành các Model trong Laravel bằng Eloquent ORM. Các Model này không chỉ đại diện cho các bảng dữ liệu mà còn định nghĩa các mối quan hệ cốt lõi.



**Hình 4: Sơ đồ Class của hệ thống GreenTech**

Trong đó User liên kết tới Role (mối quan hệ 1-1 qua cột roles), Order (1-nhiều). Product liên kết tới Category và Manufacturer (nhiều-1), ProductDetail (1-1), ProductImage (1-nhiều), OrderItem (1-nhiều). Order liên kết tới User và OrderStatus (nhiều-1), OrderItem (1-nhiều). IoTDevice liên kết tới Product (nhiều-1), DeviceMetric và AlertThreshold (1-nhiều).

Các Controller chịu trách nhiệm xử lý logic nghiệp vụ và điều phối dữ liệu giữa các Model. Ví dụ: OrdersController sử dụng các Model Order, OrderItem và Product (để cập nhật tồn kho) để hoàn tất một giao dịch.

* + 1. **Thiết kết kiến trúc Laravel**

Đồ án tuân thủ nghiêm ngặt mô hình Kiến trúc Ba Lớp (Three-Tier Architecture) được triển khai thông qua mô hình Model-View-Controller (MVC) của Laravel.

Tầng Giao diện (Presentation Layer) được quản lý bởi các View (resources/views) sử dụng Blade Template Engine. Giao diện được tách biệt rõ ràng giữa khu vực người dùng (frontend/, user/) và khu vực quản trị (administrator/, saler/, shipper/), đảm bảo tính bảo mật và trải nghiệm người dùng phù hợp với từng vai trò.

Tầng Logic Nghiệp vụ (Business Logic Layer) được xử lý bởi các Controller và Service Provider. Các Controller như ProductsController, OrdersController, và IoTDevicesController chứa các logic chính như kiểm tra đầu vào, tính toán tổng tiền, và gọi các Model để thực hiện thay đổi dữ liệu.

Middleware (đặc biệt là RoleMiddleware.php) là thành phần cốt yếu của tầng này, chịu trách nhiệm xác thực và kiểm tra quyền hạn của người dùng trước khi cho phép truy cập vào các Controller nhạy cảm.

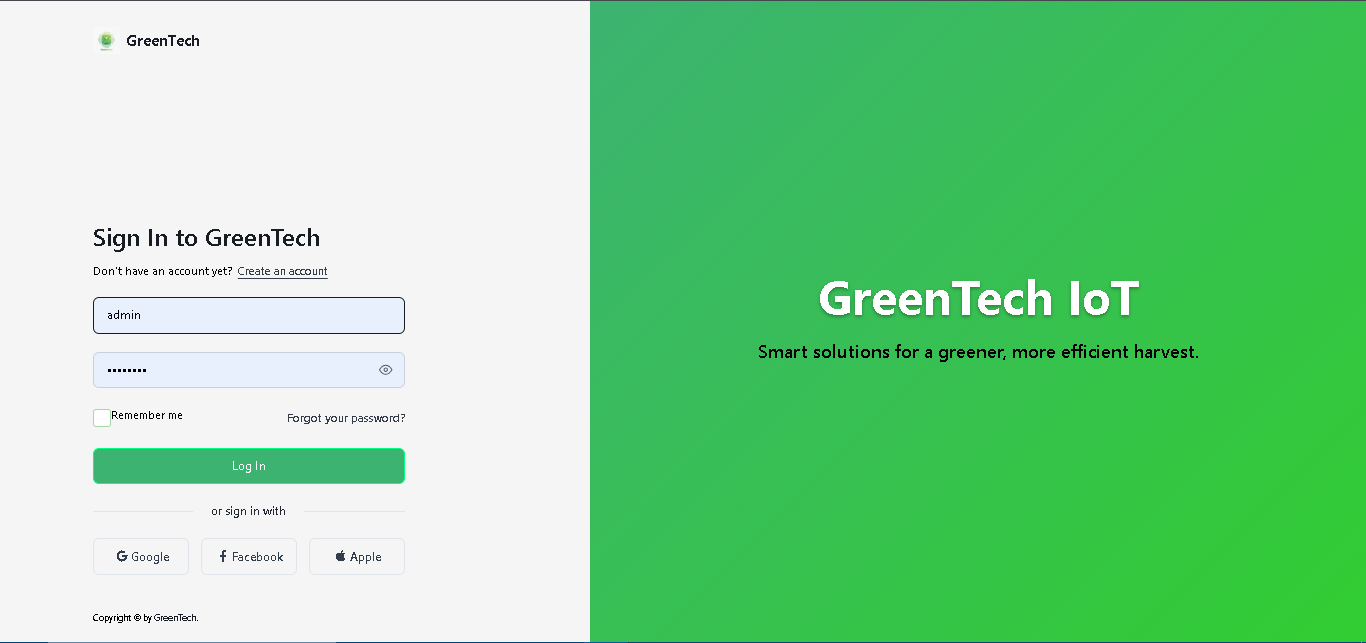
Tầng Truy cập Dữ liệu (Data Access Layer) được quản lý hoàn toàn bởi các Model (Eloquent ORM). Các Model này tương tác trực tiếp với Database để CRUD dữ liệu, đồng thời định nghĩa các ràng buộc và quan hệ giữa các bảng. Thiết kế này giúp logic truy vấn (ví dụ: tìm kiếm sản phẩm theo category\_id) được trừu tượng hóa, giữ cho Controller luôn gọn gàng và dễ bảo trì.

1. **THIẾT KẾ VÀ TRIỂN KHAI HỆ THỐNG**
   1. **Thiết Kế Giao Diện**

Thiết kế giao diện được xây dựng dựa trên nền tảng front-end hiện đại sử dụng thư viện Cartzilla/Bootstrap nhằm đảm bảo tính thẩm mỹ, thân thiện với người dùng và khả năng tương thích trên nhiều thiết bị Responsive Design. Hệ thống được tách biệt rõ ràng thành hai khu vực chính là giao diện người dùng Frontend và giao diện quản lý Backend tương ứng với các phân quyền đã thiết lập.

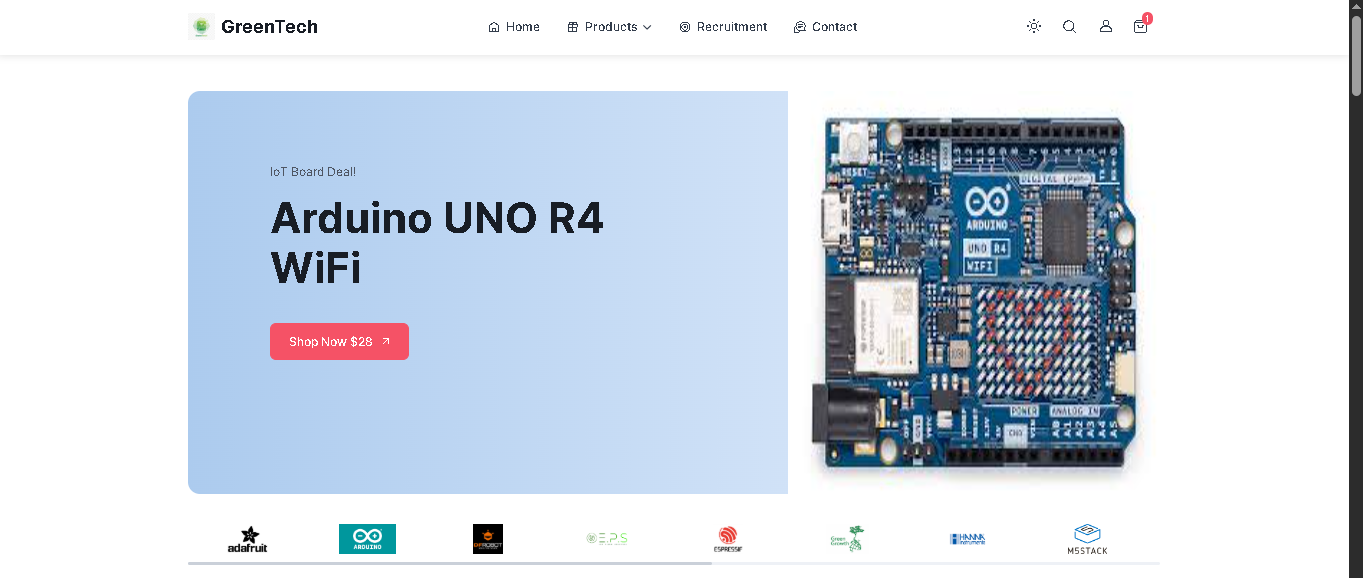
* + 1. **Giao diện người dùng**

Giao diện người dùng, được triển khai qua các view trong thư mục resources/views/frontend/ và resources/views/user/, được thiết kế nhằm mang lại trải nghiệm mua sắm tối ưu cho khách hàng các thiết bị công nghệ.

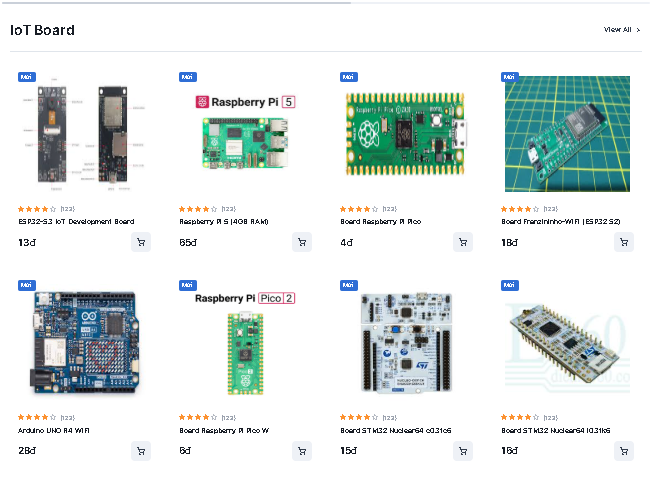


**Hình 5: Giao diện đăng nhập**

Trang chủ là cửa ngõ chính, nổi bật với các banner quảng cáo các sản phẩm IoT chủ đạo và các danh mục sản phẩm quan trọng như IoT Board, Sensor Module được lấy từ bảng categories. Trang Danh sách Sản phẩm products.blade.php cung cấp các bộ lọc nâng cao cho phép người dùng tìm kiếm theo tên, mức giá, và đặc biệt là theo Nhà sản xuất manufacturers, hỗ trợ việc tìm kiếm các linh kiện chuyên biệt một cách hiệu quả. Trang Chi tiết Sản phẩm là nơi hiển thị đầy đủ thông tin: giá, số lượng tồn kho (stock\_quantity), các hình ảnh liên quan (product\_images), và đặc biệt là bảng thông số kỹ thuật chi tiết (product\_details) như CPU, RAM, và Nguồn cấp (power\_specs), giúp khách hàng kỹ thuật dễ dàng đưa ra quyết định.



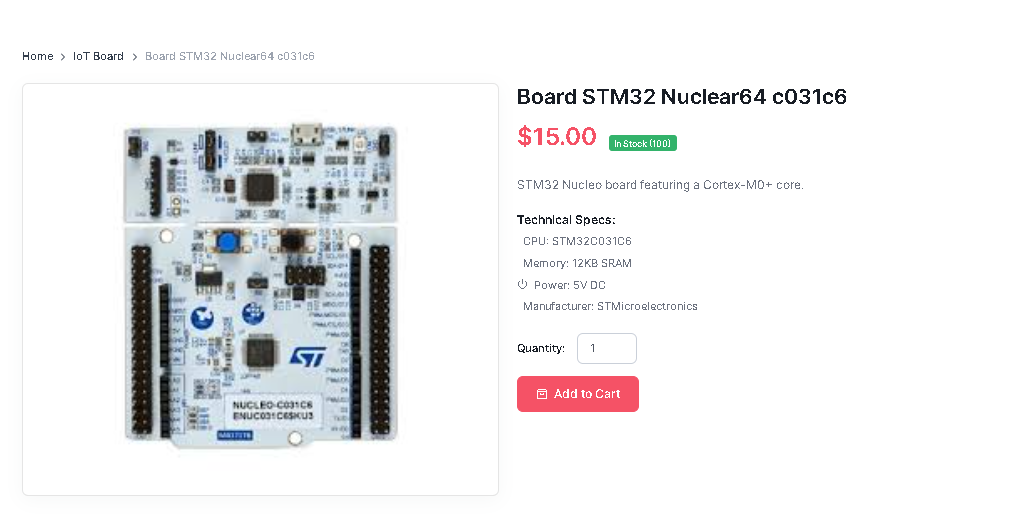
**Hình 6: Giao diện frontend**



**Hình 7: Giao diện sản phẩm Frontend**



**Hình 8: Giao diện tìm kiếm sản phẩm**

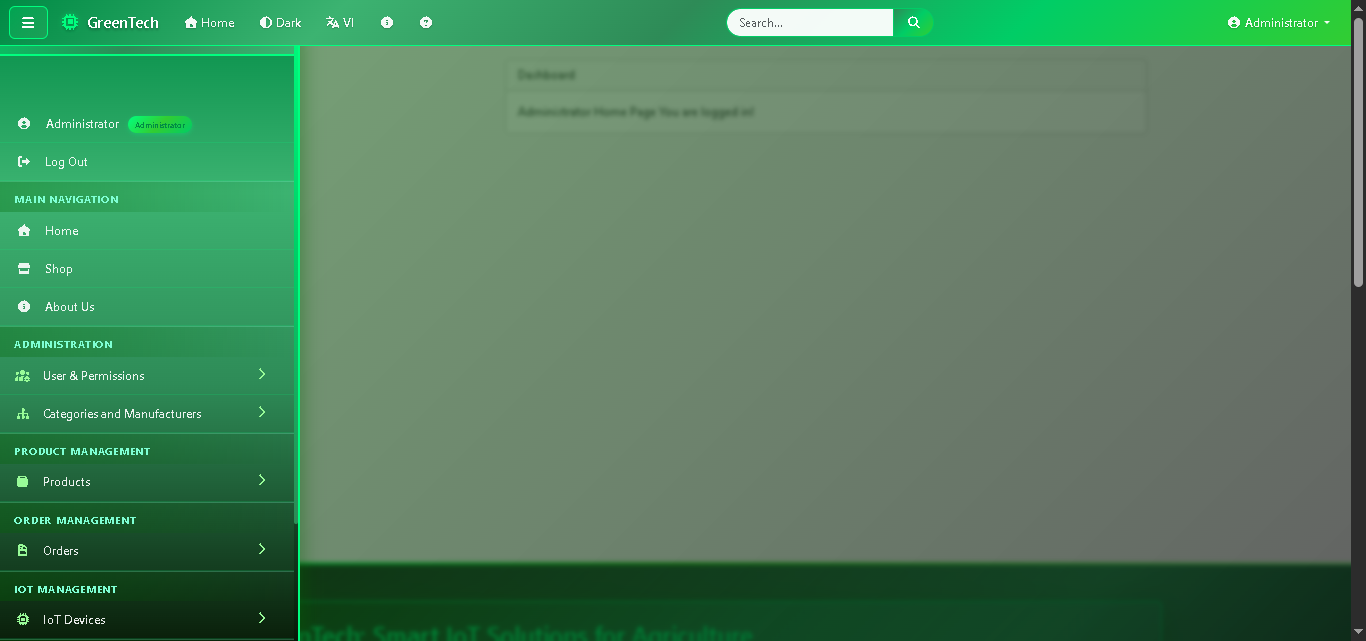


**Hình 9:Chi tiết sản phẩm**

Quy trình giao dịch được thiết kế logic qua các bước Giỏ hàng cho phép khách hàng điều chỉnh số lượng mua và tính toán tổng tiền tạm thời; sau đó chuyển đến trang Đặt hàng để xác nhận thông tin liên hệ, địa chỉ giao hàng (shipping\_address), và chọn phương thức thanh toán. Sau khi đặt hàng thành công, hệ thống sẽ chuyển đến trang thông báo và gửi email xác nhận.

* + 1. **Thiết kết giao diện quản lý**

Giao diện quản lý (Backend) được xây dựng trên một layout thống nhất và được chia thành các khu vực chức năng riêng biệt tương ứng với hệ thống phân quyền (Administrator, Saler, Shipper).



**Hình 10: Giao diện dashboard của Administrator**

Administrator Dashboard có quyền truy cập toàn bộ các module từ Quản lý Sản phẩm, Quản lý Danh mục, Quản lý Người dùng đến Quản lý Phân quyền. Đặc biệt, Admin có các form CRUD để quản lý module IoT Devices, bao gồm thêm thiết bị triển khai (administrator/iot\_devices/add.blade.php) và thiết lập ngưỡng cảnh báo. Các form nhập liệu cho Sản phẩm sử dụng thư viện CKEditor 5 để hỗ trợ mô tả chuyên sâu.

Saler Dashboard tập trung vào quản lý kho hàng và đơn hàng. Saler có quyền CRUD sản phẩm, danh mục và cập nhật trạng thái đơn hàng (ví dụ: chuyển từ Pending sang Confirmed), sử dụng các view chung nhưng với các quyền bị giới hạn bởi RoleMiddleware.

Shipper Dashboard có giao diện tối giản, tập trung vào danh sách các đơn hàng ở trạng thái Shipping, giúp Shipper dễ dàng theo dõi và cập nhật trạng thái đơn hàng thành Paid hoặc Canceled sau khi hoàn thành giao hàng.

* 1. **Thiết Kế Dữ Liệu Và Xử Lý**

Thiết kế dữ liệu và xử lý là yếu tố then chốt, đảm bảo hệ thống có thể xử lý các giao dịch thương mại phức tạp và module quản lý thiết bị IoT chuyên biệt.

* + 1. **Chi tiết thiết kế dữ liệu**

Kiến trúc dữ liệu được thiết kế tập trung vào sự phân tách giữa thông tin thương mại điện tử và thông tin kỹ thuật của thiết bị:

Tách biệt Thông số Kỹ thuật qua thiết kế bảng products lưu trữ thông tin thương mại cốt lõi (tên, giá, tồn kho…). Bảng product\_details được liên kết một-một (One-to-One) với products, chỉ chuyên biệt lưu trữ các thông số kỹ thuật cao cấp (memory, cpu, power\_specs) cho các thiết bị như bo mạch, vi điều khiển, mà không cần cho các sản phẩm đơn giản khác. Việc này giúp tối ưu hóa truy vấn cho việc duyệt danh mục thông thường.

Đảm bảo Tính toàn vẹn giao dịch thông qua bảng order\_items không chỉ lưu trữ product\_id và quantity, mà còn lưu trữ price\_at\_order (giá tại thời điểm đặt hàng). Điều này cực kỳ quan trọng, đảm bảo tổng số tiền của đơn hàng (orders.total\_amount) không bị thay đổi nếu giá sản phẩm (products.price) được cập nhật sau khi khách hàng đã đặt hàng.

Mô hình hóa Thiết bị IoT đã triển khai qua bảng iot\_devices là xương sống của module IoT. Nó liên kết một sản phẩm đã bán (product\_id) với một thực thể vật lý đã được triển khai (xác định bằng device\_id duy nhất và location). Bảng device\_metrics định nghĩa các chỉ số mà thiết bị đó có khả năng đo lường (ví dụ: metric\_key là 'temperature', unit là 'C'). Bảng alert\_thresholds liên kết trực tiếp với iot\_devices và metric\_key. Điều đặc biệt là việc sử dụng khóa duy nhất tổ hợp (iot\_device\_id, metric\_key) đảm bảo rằng mỗi chỉ số của mỗi thiết bị chỉ có thể có một cặp ngưỡng cảnh báo Min và Max được xác định, đảm bảo tính chính xác trong việc thiết lập logic cảnh báo.

* + 1. **Chi tiết thiết kế xử lý**

Luồng Xử lý Đặt hàng (Checkout Process) được quản lý chủ yếu bởi OrdersController.php và tuân thủ nguyên tắc ACID (Atomicity, Consistency, Isolation, Durability) thông qua việc sử dụng Database Transaction.

|  |
| --- |
| public function postPlaceOrder(Request $request)      {          // 1. Validation (Giữ nguyên)          $this->validate($request, [              'shipping\_address' => ['required', 'string', 'max:255'],              'contact\_phone' => ['required', 'string', 'max:20'],              'payment\_method' => ['required', 'string', 'max:50'],              'accept-terms' => ['accepted'],          ]);          if (Cart::count() == 0) {              return redirect()->route('frontend.shoppingcard')->with('error', 'Your cart is empty. Please add items before placing an order.');          }          $initialStatusId = OrderStatus::where('name', 'pending')->first()?->id ?? 1;          DB::beginTransaction();          try {              $subtotal = Cart::priceTotal(2, '.', '');              $taxAmount = Cart::tax(2, '.', '');              $shippingFee = 0.00;              $totalAmount = Cart::total(2, '.', '');              $order = new Order();              $order->user\_id = Auth::user()->id;              $order->order\_status\_id = $initialStatusId;              $order->shipping\_address = $request->shipping\_address;              $order->contact\_phone = $request->contact\_phone;              $order->payment\_method = $request->payment\_method;              $order->subtotal = $subtotal;              $order->tax\_amount = $taxAmount;              $order->shipping\_fee = $shippingFee;              $order->total\_amount = $totalAmount;              $order->notes = $request->notes ?? null;              $order->save();              foreach (Cart::content() as $value) {                  $productExists = Product::where('id', $value->id)->exists();                  if (!$productExists) {                      throw new \Exception("Product ID {$value->id} not found.");                  }                  $detail = new OrderItem();                  $detail->order\_id = $order->id;                  $detail->product\_id = $value->id;                  $detail->quantity = $value->qty;                  $detail->price\_at\_order = $value->price;                  $detail->save();              }              try {                  Mail::to(Auth::user()->email)->send(new PlaceOrderSuccessEmail($order));              } catch (\Exception $emailException) {                  Log::warning('Failed to send order confirmation email for order ID ' . $order->id . ': ' . $emailException->getMessage());              }              DB::commit();              return redirect()->route('user.place-order-success')->with('success', 'Your order has been successfully placed.');          } catch (\Exception $e) {              DB::rollBack();              Log::error('Order placement failed: ' . $e->getMessage(), ['user\_id' => Auth::id(), 'cart\_data' => Cart::content()->toArray()]);              return redirect()->back()                  ->with('error', 'Order failed. Error: ' . $e->getMessage())                  ->withInput();          }      } |

Kiểm tra và Xác thực với dữ liệu đầu vào từ form thanh toán (địa chỉ, số điện thoại, phương thức thanh toán) được xác thực. Khởi tạo Transaction bắt đầu một giao dịch database để đảm bảo toàn bộ quá trình đặt hàng là một đơn vị logic.Tạo Đơn hàng Chính bằng cách tạo một bản ghi mới được tạo trong bảng orders, ghi nhận thông tin tổng quan, bao gồm user\_id, địa chỉ (shipping\_address), tổng tiền (total\_amount), và trạng thái mặc định là Pending (ID=1) trong order\_status\_id. Tạo Chi tiết Đơn hàng và Cập nhật tồn kho bằng cách lặp qua từng mặt hàng trong giỏ hàng (Cart Session).Tạo bản ghi trong order\_items, lưu lại giá chính xác tại thời điểm đặt hàng (price\_at\_order).Giảm số lượng tồn kho (stock\_quantity) tương ứng trong bảng products. Nếu bất kỳ sản phẩm nào không đủ tồn kho, toàn bộ giao dịch sẽ được hoàn tác (Rollback). Kết thúc Transaction: Giao dịch được xác nhận (Commit).Cuối cùng, hệ thống gửi email xác nhận đặt hàng thành công đến khách hàng, sử dụng class PlaceOrderSuccessEmail.php và nội dung template tại resources/views/emails/place-order-success.blade.php.

Luồng Xử lý Quản lý Cảnh báo IoT, luồng này do Admin/Saler thực hiện, sử dụng các Controller IoTDevicesController.php và AlertThresholdsController.php.

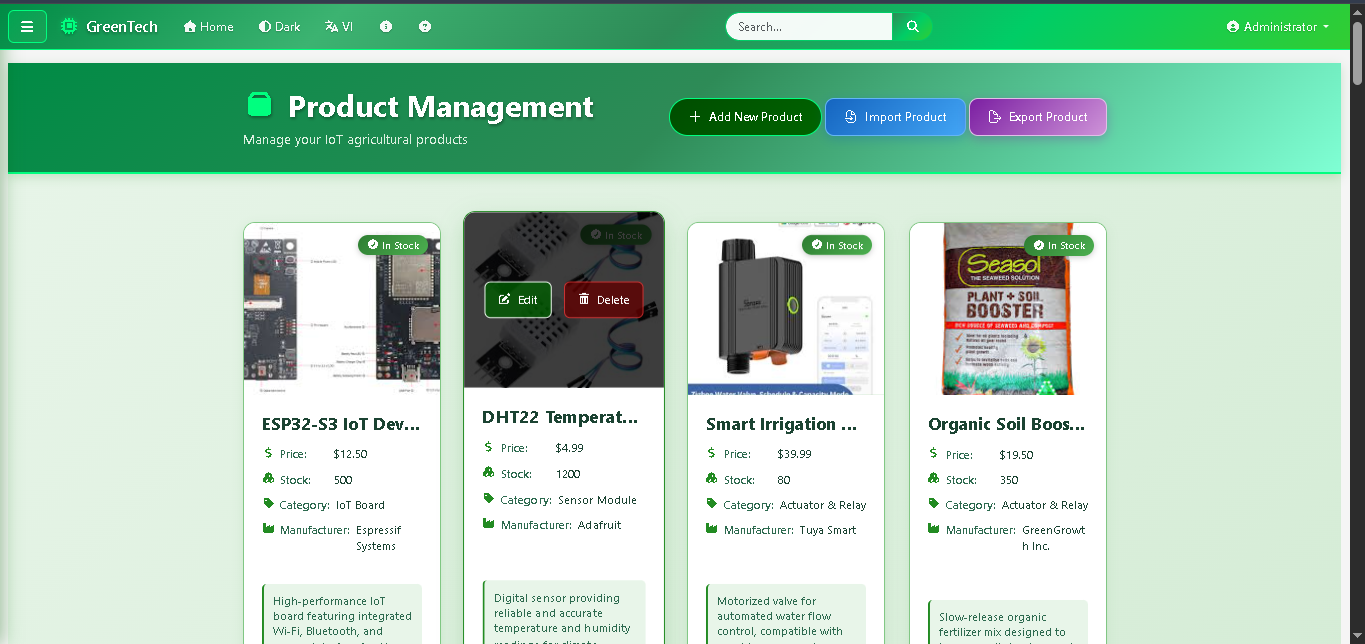
Đăng ký Thiết bị Triển khai khi một sản phẩm IoT được bán và đưa vào sử dụng, Admin tạo một bản ghi trong iot\_devices, gán product\_id và cung cấp device\_id (ID định danh) cùng location. Admin định nghĩa các chỉ số cần theo dõi cho thiết bị đó trong bảng device\_metrics. Sử dụng AlertThresholdsController, Admin tạo (hoặc cập nhật) các ngưỡng min\_value và max\_value cho từng metric\_key cụ thể của thiết bị đó. Chẳng hạn, thiết bị 'ESP\_SENSOR\_DHT22\_01' (ID=2) sẽ có ngưỡng nhiệt độ (temperature) Min 18.00 và Max 30.00. Bất kỳ dữ liệu nào nhận được sau này vượt ra ngoài phạm vi này sẽ kích hoạt cảnh báo, cho thấy logic nghiệp vụ của module giám sát đã sẵn sàng.

* 1. **Triển khai chức năng**

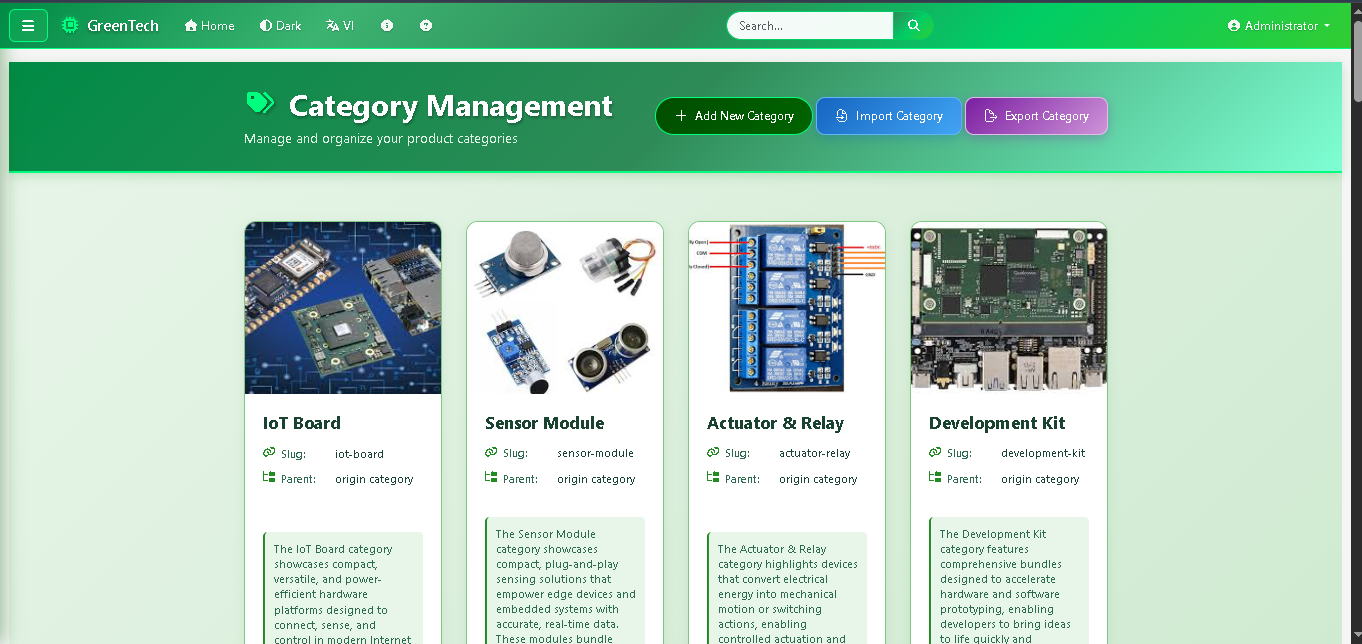
Phần này đi sâu vào cách thức các module chính của hệ thống được xây dựng và tích hợp trên Framework Laravel, đảm bảo đáp ứng các yêu cầu về thương mại điện tử chuyên biệt và quản lý thiết bị IoT.

* + 1. **Module Quản lý Sản phẩm, Danh Mục, Nhà Sản Xuất**

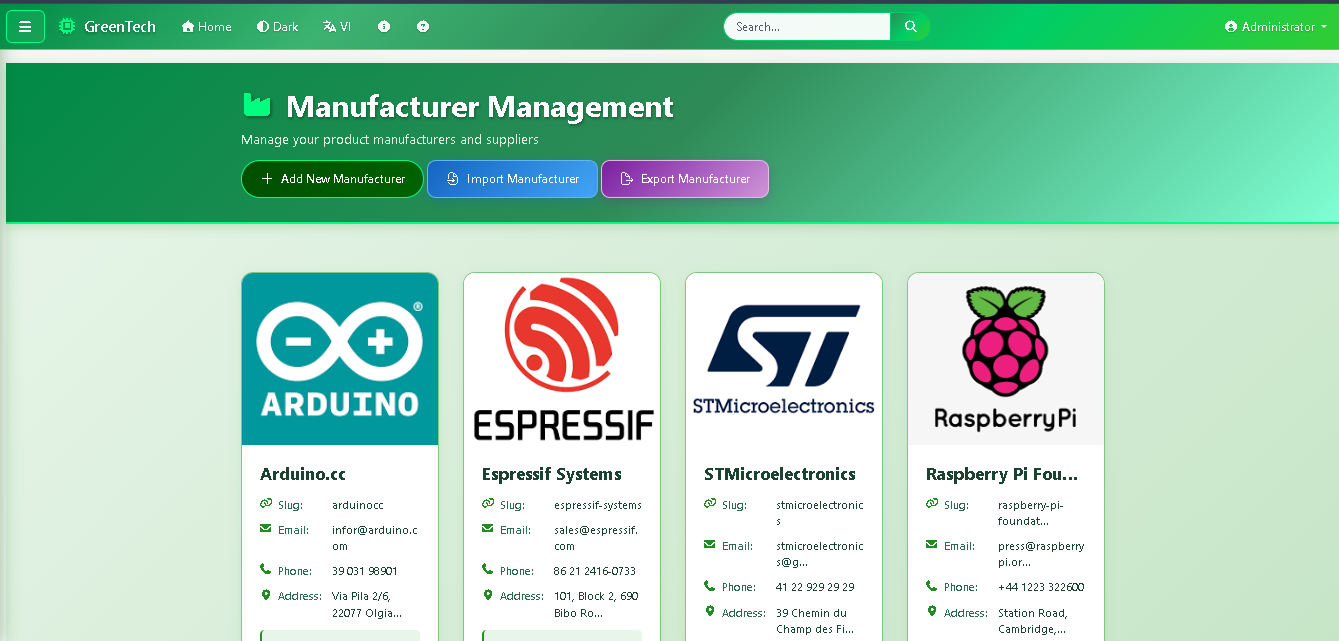
Đây là trái tim của hệ thống E-commerce, được quản lý toàn diện bởi các Controller tương ứng như ProductsController, CategoriesController, và ManufacturesController. Nhân viên Bán hàng (Saler) và Quản trị viên (Administrator) có trách nhiệm thực hiện các thao tác CRUD Tạo, Đọc, Cập nhật, Xóa trên các thực thể này. Để tối ưu hóa quy trình nhập liệu ban đầu, hệ thống đã tích hợp thư viện Maatwebsite/Laravel-Excel, cho phép Admin dễ dàng nhập và xuất dữ liệu sản phẩm, danh mục từ các file Excel/CSV. Điểm đặc trưng trong module này là việc quản lý thông số kỹ thuật chuyên sâu của thiết bị IoT: các thông tin chi tiết về CPU, RAM, và Nguồn cấp được lưu trữ riêng biệt trong bảng product\_details, được liên kết một-một với bảng products thông qua ProductDetailsController, giúp cấu trúc dữ liệu luôn gọn gàng và dễ dàng truy xuất thông số kỹ thuật cho các thiết bị công nghệ.



**Hình 11: Giao diện quản lý sản phẩm**



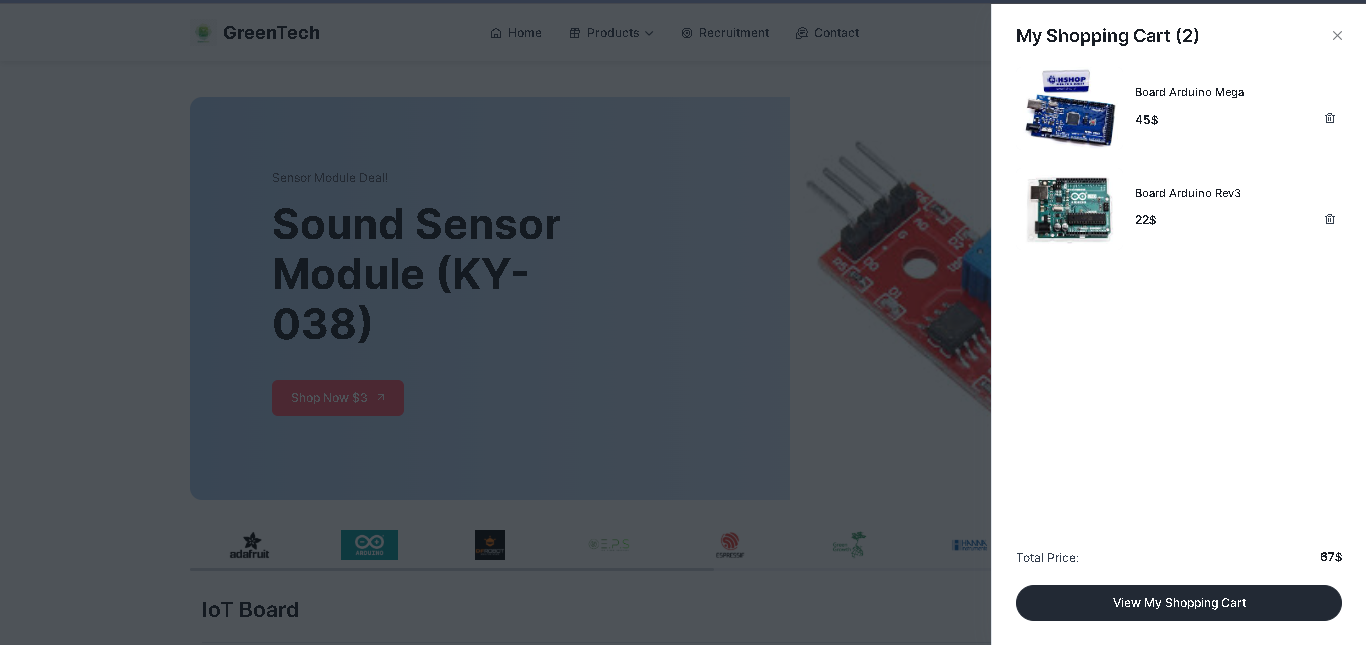
**Hình 12: Giao diện quản lý danh mục**



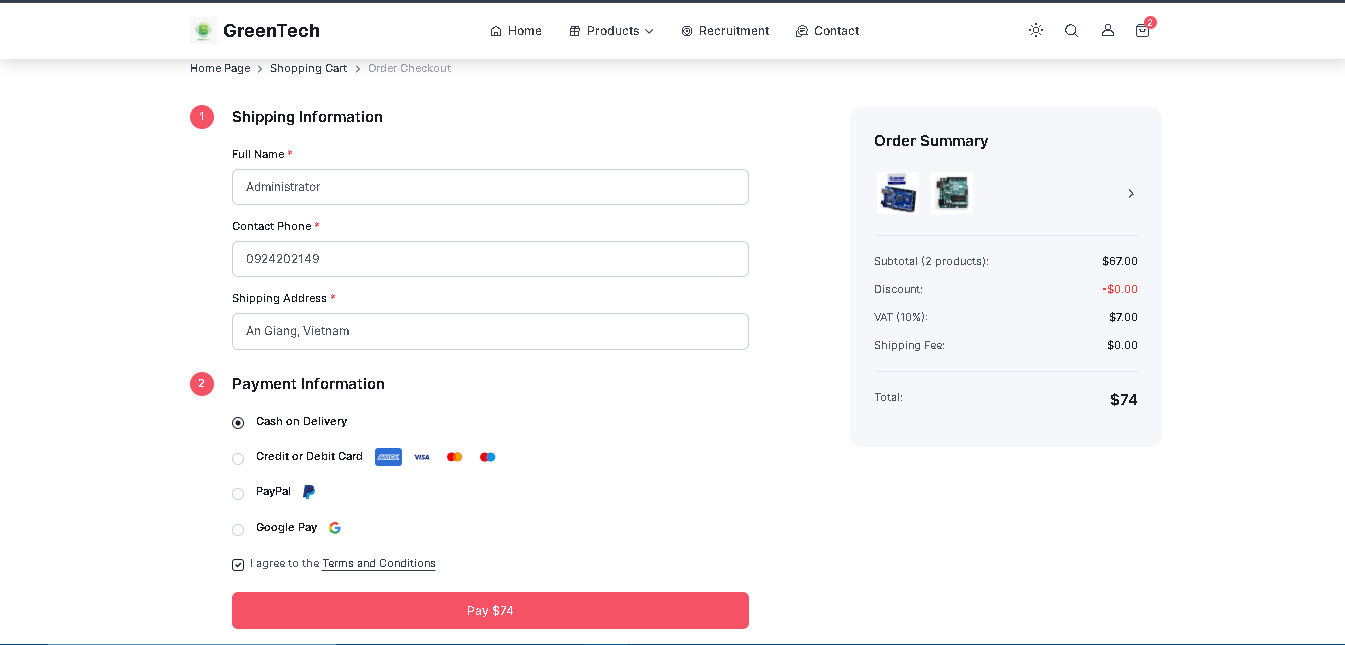
**Hình 13: Giao diện quản lý nhà sản xuất**

* + 1. **Module Giỏ hàng và Thanh toán**

Module này cung cấp trải nghiệm mua sắm không bị gián đoạn cho Khách hàng. Giỏ hàng được quản lý bằng phiên (Session) của Laravel, cho phép người dùng thêm, xóa, và điều chỉnh số lượng sản phẩm trước khi quyết định mua. Khi Khách hàng chuyển sang trang Thanh toán, OrdersController sẽ điều phối luồng xử lý chính. Toàn bộ quy trình đặt hàng được đóng gói trong một Database Transaction để đảm bảo tính toàn vẹn tuyệt đối: nếu việc tạo bản ghi đơn hàng trong bảng orders, tạo các bản ghi chi tiết trong order\_items và cập nhật giảm số lượng tồn kho (stock\_quantity) trong bảng products thành công, giao dịch mới được xác nhận (Commit). Ngay sau khi thanh toán, hệ thống sử dụng class PlaceOrderSuccessEmail.php để gửi một email tự động chi tiết về đơn hàng, hoàn thành chu trình giao dịch.



**Hình 14: Giao diện giỏ hàng**

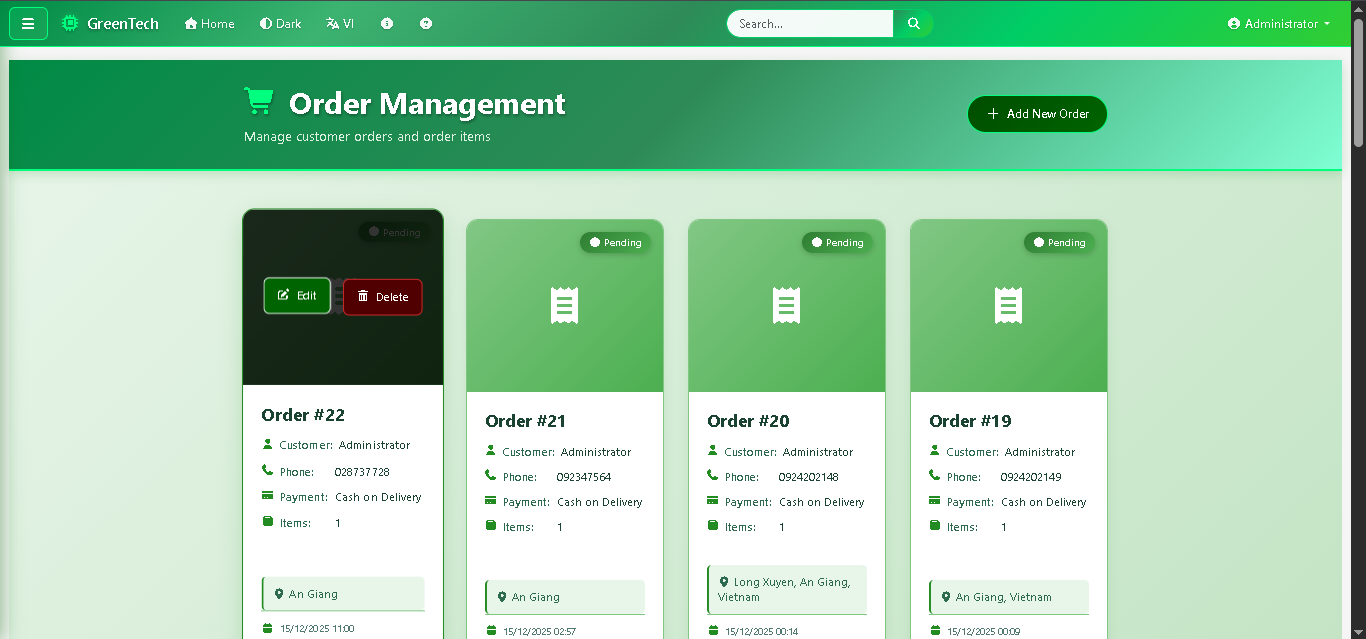


**Hình 15: Giao diện thanh toán**

* + 1. **Module Quản lý Đơn hàng**

Sau khi đơn hàng được tạo, nó được chuyển giao cho các nhóm nội bộ quản lý. Bảng orders liên kết với bảng order\_statuses để theo dõi chính xác hành trình của đơn hàng. Quản trị viên và Nhân viên Bán hàng có thể theo dõi danh sách đơn hàng và cập nhật trạng thái từ Pending sang Confirmed (OrdersController). Khi đơn hàng chuyển sang trạng thái Shipping (đang giao), trách nhiệm được chuyển giao cho Nhân viên Giao hàng (Shipper). Shipper có giao diện riêng biệt để theo dõi các đơn

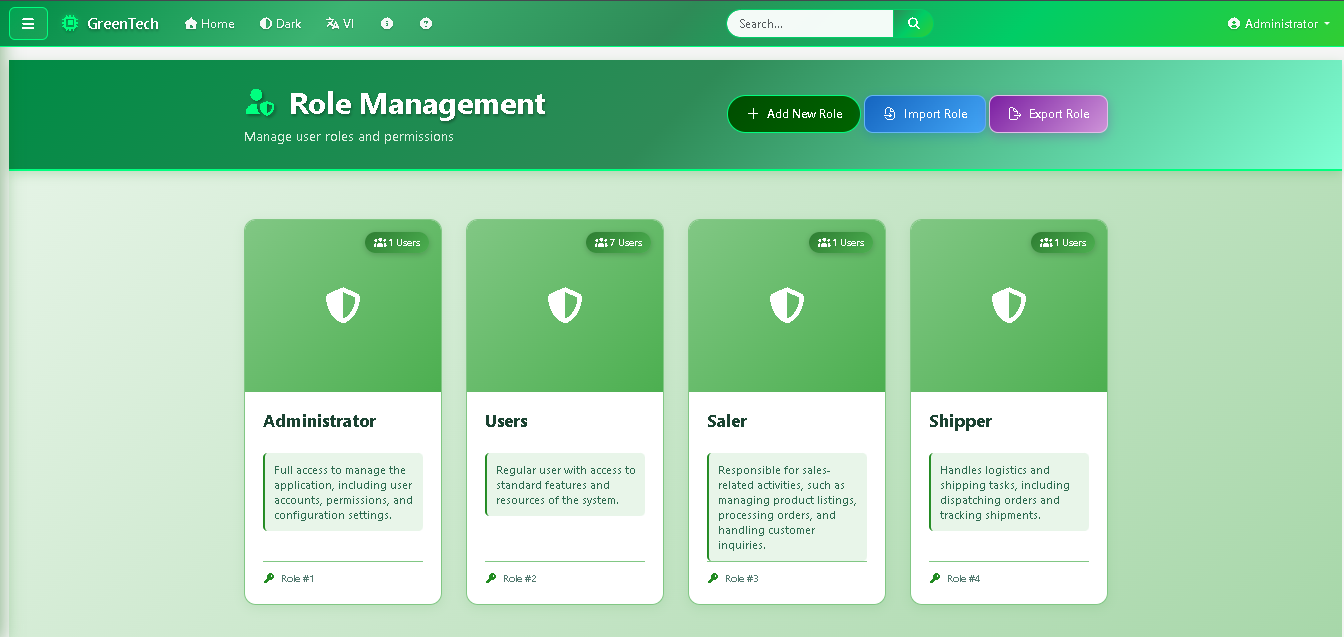
hàng đang vận chuyển và cập nhật trạng thái cuối cùng thành Paid (Đã thanh toán) hoặc Canceled (Đã hủy), đảm bảo luồng logistics được kiểm soát chặt chẽ.



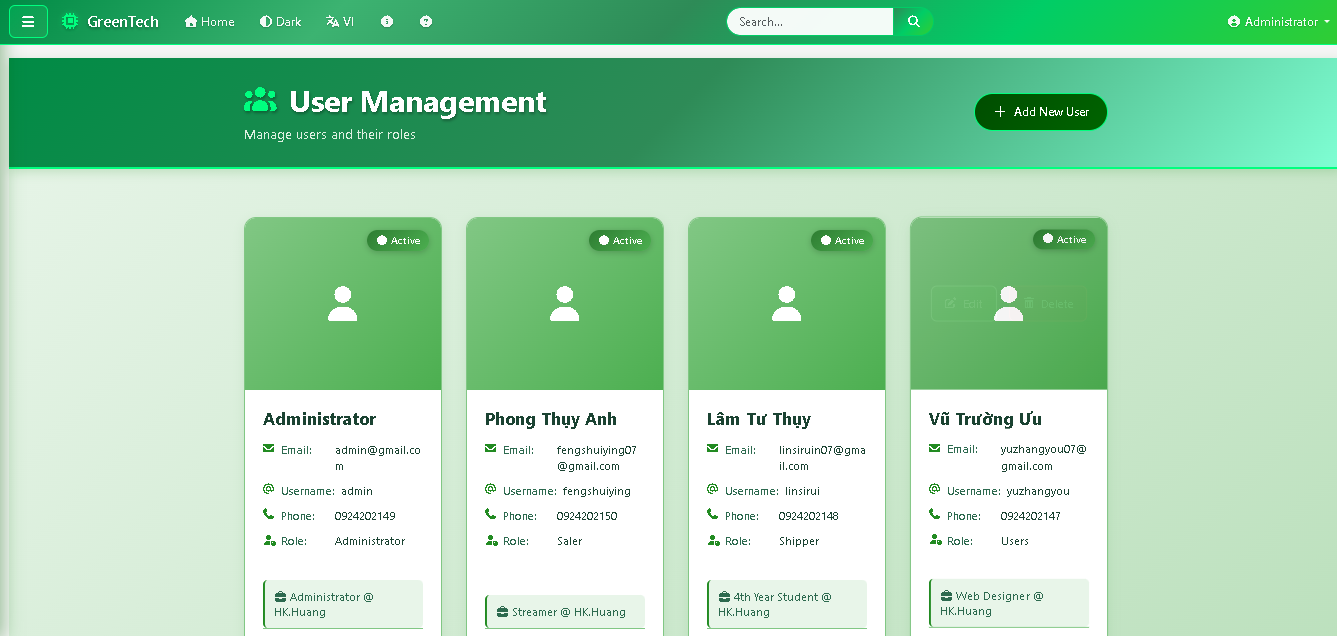
**Hình 16: Giao diện quản lý hóa đơn**

* + 1. **Module Quản lý Người dùng và Phân quyền**

Nền tảng bảo mật của hệ thống được xây dựng trên cơ chế phân quyền RBAC (Role-Based Access Control) của Laravel. Bảng roles định nghĩa bốn cấp độ chính: Administrator, Saler, Shipper, và Users (Khách hàng). Mỗi người dùng trong bảng users được gán một ID vai trò duy nhất. Mọi truy cập vào các khu vực quản trị được kiểm soát bởi RoleMiddleware.php, là một lớp bảo vệ cốt lõi, nó sẽ kiểm tra vai trò của người dùng hiện tại trước khi cho phép họ truy cập vào các Controller và View tương ứng. Điều này đảm bảo rằng Shipper không thể truy cập vào các chức năng quản lý sản phẩm của Saler, và chỉ Administrator mới có quyền quản lý toàn bộ hệ thống.



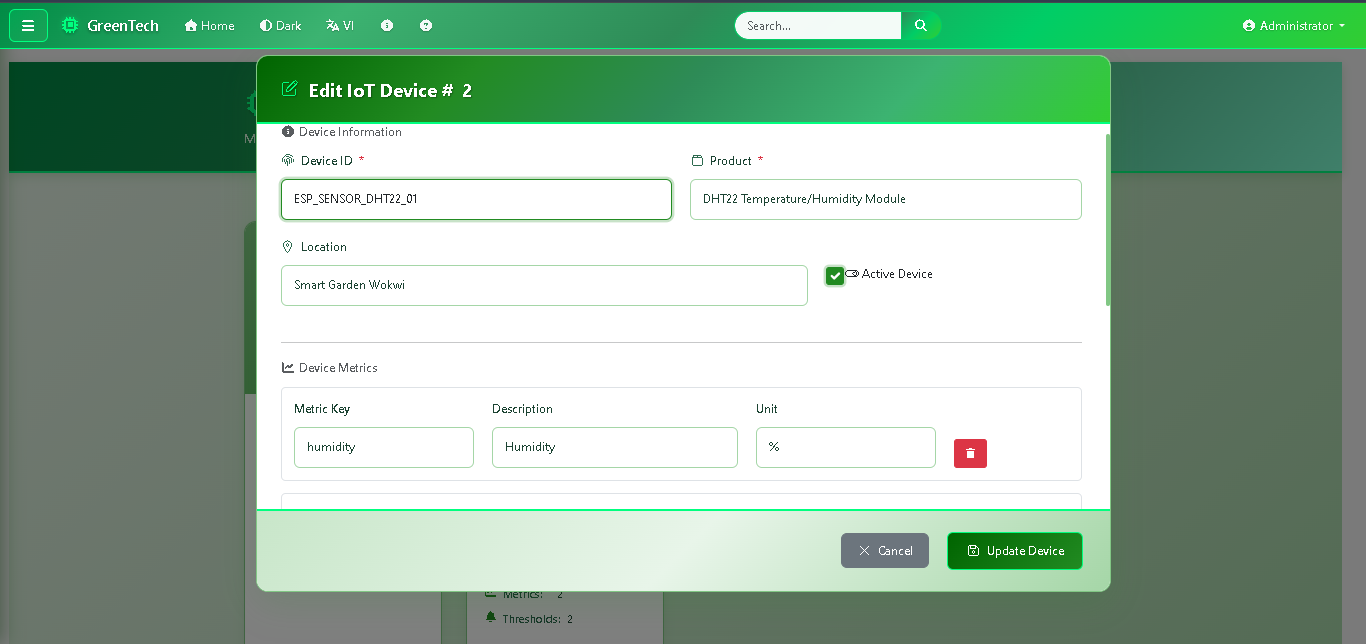
**Hình 17: Giao diện quản lý phân quyền**



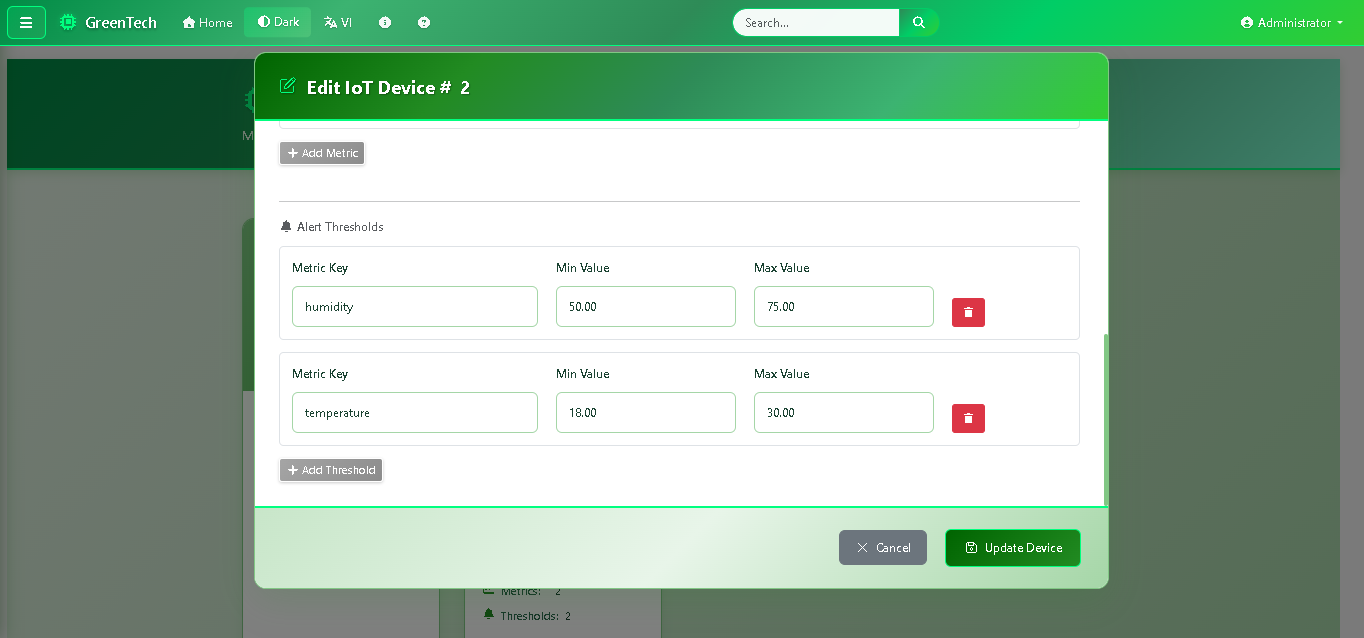
**Hình 18: Giao diện quản lý người dùng**

* + 1. **Module Giám sát Thiết bị IoT**

Đây là tính năng độc đáo của đồ án, mô hình hóa vòng đời sau bán hàng của thiết bị. Module này cho phép Admin quản lý các thiết bị IoT đã được triển khai (bảng iot\_devices), mỗi thiết bị được định danh bằng một device\_id và gán với product\_id tương ứng. Admin có thể định nghĩa các chỉ số đo lường (device\_metrics, ví dụ: 'light\_value' với đơn vị 'lux') mà thiết bị này cung cấp. Quan trọng hơn, thông qua bảng alert\_thresholds, Admin thiết lập các ngưỡng an toàn (min\_value và max\_value) cho từng chỉ số. Thiết kế này tạo ra một khung sườn logic: trong thực tế, nếu dữ liệu thật từ thiết bị vượt quá ngưỡng này, hệ thống sẽ kích hoạt một quy trình cảnh báo, minh chứng cho việc đồ án không chỉ là một trang web bán hàng mà còn là một nền tảng quản lý thiết bị thông minh.



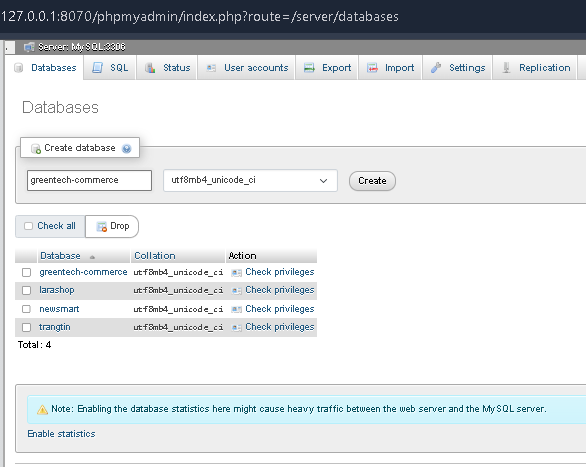
**Hình 19: Thiết lập số liệu thiết bị IoT**



**Hình 20: Thiết lập ngưỡng cảnh báo thiết bị IoT**

* 1. **Hướng Dẫn Đăng Nhập Và Sử Dụng Hệ Thống**

Truy cập trang 127.0.0.1/phpmyadmin chạy trên máy chủ Wampp tạo database với name: greentech-commerce

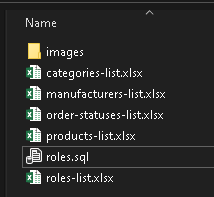


**Hình 21: Tạo database greentech-commerce trên phpmyadmin**

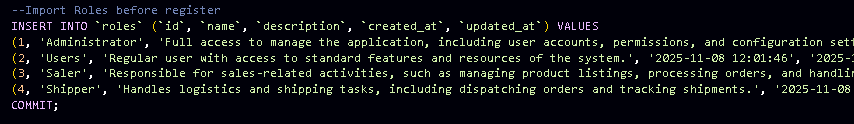
Truy cập thư mục gốc hệ thống chạy lệnh bên dưới để tạo database qua migration.

|  |
| --- |
| **php artisan migrate** |

Truy cập thư mục ImportFile trong hệ thống, copy đoạn sql Import và cơ sở dữ liệu greentech-commerce trước khi đăng ký tài khoản



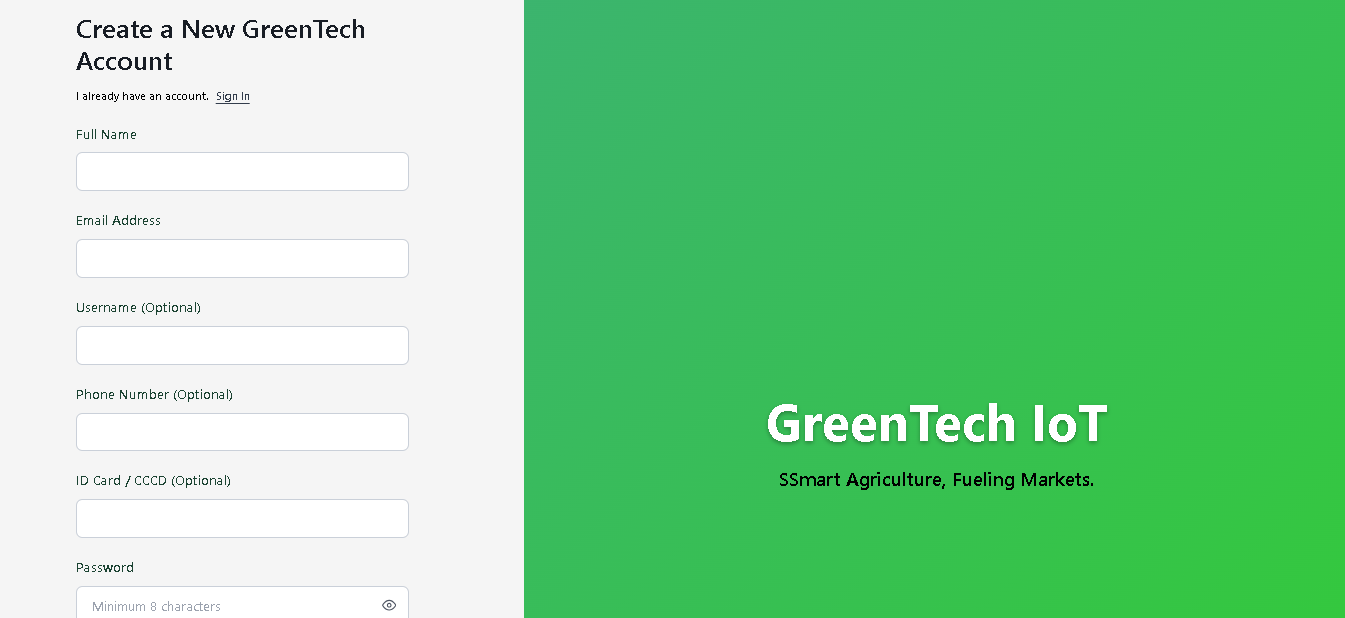
**Hình 22: Nội dung thư mục ImportFiles**



**Hình 23: Nội dung file role.sql**

Việc truy cập và sử dụng hệ thống được phân luồng rõ ràng theo vai trò:

Đăng ký và Đăng nhập (Khách hàng/Users): Khách hàng truy cập trang Đăng ký để tạo tài khoản, sau khi đăng ký thành công, vai trò mặc định được gán là Users (ID=2). Khách hàng sử dụng tên đăng nhập (username) hoặc email và mật khẩu để Đăng nhập. Sau khi đăng nhập, họ sẽ được chuyển hướng đến trang Hồ sơ cá nhân (user/profile.blade.php) và có thể bắt đầu quá trình mua sắm.



**Hình 24: Giao diện register**

Các tài khoản được tạo đều có quyền User, dùng phpmyadmin cập nhật một tài khoảng có roles là 1(quyền admin) để sử dụng chức năng quản trị



**Hình 25: Cập nhật quyền trong admin**

Truy cập Hệ thống Quản trị (Admin/Saler/Shipper):Các tài khoản nội bộ (Admin, Saler, Shipper) cũng sử dụng giao diện Đăng nhập chung.

Hệ thống sử dụng logic điều hướng sau đăng nhập:Administrator (ID=1) sẽ được chuyển đến trang tổng quan Admin (/administrator/home).Saler (ID=3) sẽ được chuyển đến trang tổng quan Bán hàng (/saler/home), tập trung vào quản lý sản phẩm và đơn hàng.Shipper (ID=4) sẽ được chuyển đến trang tổng quan Giao hàng (/shipper/home), chỉ hiển thị các đơn hàng đang ở trạng thái Shipping để tiện theo dõi và cập nhật.Mọi nỗ lực truy cập vào các URL quản trị không thuộc vai trò của người dùng (ví dụ: Shipper cố gắng truy cập trang Quản lý Người dùng của Admin) sẽ bị chặn bởi RoleMiddleware, hiển thị thông báo lỗi hoặc chuyển hướng về trang chủ tương ứng, đảm bảo tính an toàn và bảo mật cho hệ thống.

1. **TỔNG KẾT VÀ HƯỚNG PHÁT TRIỂN**
   1. **Kết Quả Đạt Được**

Sau quá trình nghiên cứu, phân tích và triển khai, đồ án đã hoàn thành mục tiêu đề ra, xây dựng thành công một nền tảng thương mại điện tử chuyên biệt cho thiết bị IoT, đáp ứng tốt các yêu cầu chức năng và phi chức năng cơ bản.

Xây dựng Hệ thống E-commerce hoàn chỉnh qua việc triển khai đầy đủ chu trình mua sắm trực tuyến. Khách hàng có thể đăng ký tài khoản, tìm kiếm, duyệt sản phẩm theo danh mục (categories) và nhà sản xuất (manufacturers), quản lý giỏ hàng và hoàn tất quy trình thanh toán (tạo bản ghi trong orders và order\_items). Việc sử dụng Database Transaction trong quy trình đặt hàng đảm bảo tính toàn vẹn dữ liệu và độ tin cậy của giao dịch.

Hệ thống được xây dựng theo mô hình MVC (Model-View-Controller) của Framework Laravel, giúp tách biệt rõ ràng các tầng logic, giao diện và dữ liệu, tạo ra một cấu trúc mã nguồn gọn gàng, dễ bảo trì và mở rộng.

Hệ thống Quản trị Đa cấp Chặt chẽ qua hệ thống phân quyền bốn cấp độ (Administrator, Saler, Shipper, Users) thông qua bảng roles và middleware bảo mật (RoleMiddleware.php). Điều này đảm bảo mỗi nhân viên nội bộ chỉ có quyền truy cập vào các module liên quan đến nghiệp vụ của mình chẳng hạn Shipper chỉ quản lý đơn hàng ở trạng thái Shipping, tăng cường bảo mật và kiểm soát hệ thống.

Đồ án đã thành công trong việc mô hình hóa dữ liệu thiết bị IoT sau bán hàng. Các bảng iot\_devices, device\_metrics, và alert\_thresholds đã được thiết lập, cho phép quản trị viên định nghĩa các chỉ số đo lường (như nhiệt độ, độ ẩm) và cài đặt ngưỡng an toàn (Min/Max). Mặc dù là dữ liệu mô phỏng, module này chứng minh khả năng sẵn sàng tích hợp với một nền tảng giám sát IoT thực tế.

Giao diện được thiết kế hiện đại, responsive (sử dụng Bootstrap/Cartzilla) và thân thiện, đặc biệt là trang chi tiết sản phẩm, nơi hiển thị các thông số kỹ thuật chuyên ngành (product\_details) quan trọng đối với đối tượng khách hàng kỹ thuật.

* 1. **Đánh giá ưu nhược điểm đồ án** 
     1. **Ưu điểm**

Tính chuyên biệt hóa cao vì đồ án không chỉ là một trang E-commerce thông thường mà còn giải quyết được yêu cầu đặc thù của ngành IoT thông qua module quản lý thiết bị và cảnh báo ngưỡng.

Cấu trúc kỹ thuật vững chắc thông qua việc sử dụng Laravel với Eloquent ORM và mô hình MVC giúp quản lý các mối quan hệ phức tạp giữa các bảng (như quan hệ giữa Product, Order và IoTDevice) một cách hiệu quả. Việc sử dụng Transaction trong thanh toán đảm bảo độ tin cậy cao cho các giao dịch.

Cơ chế phân quyền chi tiết cho Admin, Saler, và Shipper giúp bảo vệ dữ liệu nhạy cảm và tối ưu hóa luồng công việc nội bộ.

Tích hợp thư viện Maatwebsite/Laravel-Excel hỗ trợ nhập và xuất dữ liệu hàng loạt, giúp tiết kiệm thời gian quản lý cho Admin.

* + 1. **Nhược điểm**

Thiếu tính năng Real-time IoT do giới hạn về phạm vi và thời gian, module IoT chỉ dừng lại ở việc mô hình hóa dữ liệu và ngưỡng cảnh báo. Hệ thống chưa triển khai được kết nối vật lý thời gian thực qua MQTT hoặc WebSockets để thu thập và hiển thị dữ liệu thực tế từ thiết bị.

Chức năng Thanh toán đơn giản vì hiện tại, hệ thống mới chỉ hỗ trợ phương thức thanh toán cơ bản (như COD) và chưa tích hợp cổng thanh toán trực tuyến của bên thứ ba (như VNPay, PayPal), giới hạn phạm vi giao dịch.

Tính năng tìm kiếm và lọc sản phẩm chủ yếu dựa trên tên, slug và category. Khả năng tìm kiếm nâng cao dựa trên các thông số kỹ thuật cụ thể (ví dụ: tìm kiếm bo mạch có '4GB RAM') còn hạn chế.

Hệ thống chưa có các module báo cáo chuyên sâu (ví dụ: báo cáo doanh thu theo tháng, thống kê sản phẩm bán chạy nhất) cần thiết cho việc ra quyết định kinh doanh của Administrator.

* 1. **Hướng phát triển**

Tích hợp Real-time IoT Monitoring bằng cách triển khai các API (RESTful API hoặc MQTT Broker) để nhận dữ liệu thời gian thực từ thiết bị vật lý.

Xây dựng giao diện hiển thị dữ liệu dưới dạng biểu đồ động (sử dụng Chart.js hoặc thư viện tương tự để trực quan hóa DeviceMetric.

Phát triển cơ chế xử lý cảnh báo tự động khi dữ liệu vượt ngưỡng alert\_thresholds, hệ thống sẽ tự động gửi thông báo (Notification/Email) đến người quản lý.

Mở rộng Cổng Thanh toán qua việc tích hợp các cổng thanh toán trực tuyến phổ biến (VNPay, Momo, ZaloPay) để mở rộng phạm vi giao dịch và tăng tính tiện lợi cho Khách hàng.

Phát triển Chức năng Tìm kiếm Nâng cao bằng việc ap dụng công nghệ tìm kiếm Full-text Search (Elasticsearch/Laravel Scout) hoặc tối ưu hóa truy vấn SQL để cho phép người dùng lọc sản phẩm dựa trên các thông số kỹ thuật chi tiết trong bảng product\_details (ví dụ: tìm kiếm theo "CPU Dual-Core").

Xây dựng Module Quản lý Nội dung Động qua phát triển hệ thống cho phép Administrator quản lý các trang tĩnh (About Us, Contact, FAQ) và các banner trên trang chủ một cách linh hoạt mà không cần can thiệp vào mã nguồn (CMS Lite).

Xây dựng các Dashboard thống kê trực quan cho Administrator và Saler, hiển thị các chỉ số kinh doanh quan trọng (doanh thu, đơn hàng thành công, tỉ lệ chuyển đổi) theo thời gian.

1. **TÀI LIỆU THAM KHẢO**

[1] “Role of Internet of Things (IoT) in Adoption of Industry 4.0 | Journal of Industrial Integration and Management.” Accessed: Dec. 15, 2025. [Online]. Available: https://www.worldscientific.com/doi/10.1142/S2424862221500068

[2] “IoT là gì? Lợi ích và ứng dụng của Internet of Things.” Accessed: Dec. 15, 2025. [Online]. Available: https://viettelidc.com.vn/tin-tuc/iot-la-gi-nhung-ung-dung-noi-bat-cua-iot

[3] “Laravel - The PHP Framework For Web Artisans.” Accessed: Dec. 15, 2025. [Online]. Available: https://laravel.com/

[4] “What is MVC? - Laravel - The PHP Framework For Web Artisans.” Accessed: Dec. 15, 2025. [Online]. Available: https://laravel.com/learn/getting-started-with-laravel/what-is-mvc

[5] “Lập trình web với HTML5 - CSS3 - Bootstrap - JQuery - Ajax | Devmaster.” Accessed: Dec. 15, 2025. [Online]. Available: https://devmaster.edu.vn/cat-ghep-giao-dien-website-voi-html5-css3-jquery-ajax-bootstrap-devmaster-hcjabp.html

[6] “Laravel xuất file excel sử dụng thư viện maatwebsite.” Accessed: Dec. 15, 2025. [Online]. Available: https://viblo.asia/p/laravel-xuat-file-excel-su-dung-thu-vien-maatwebsite-Do754Qk3KM6

[7] “Ckeditor và Upload ảnh.” Accessed: Dec. 15, 2025. [Online]. Available: https://viblo.asia/p/ckeditor-va-upload-anh-MdZkAQwAkox