TRƯỜNG ĐẠI HỌC TRÀ VINH

KHOA KỸ THUẬT VÀ CÔNG NGHỆ

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**

**----------🕮-----------**

****

BÁO CÁO KẾT THÚC MÔN HỌC

PHÁT TRIỂN ỨNG DỤNG WEB VỚI MÃ NGUỒN MỞ

MÃ ĐỀ TÀI:

**ĐỀ TÀI: XÂY DỰNG WEBSITE BÁN PC**

|  |  |
| --- | --- |
| Giảng viên hướng dẫn:  **Họ và tên: Phạm Thị Trúc Mai** | Sinh viên thực hiện:  **Nguyễn Tín Thành** (110121104)  **Hồ Hoàng Phúc** (110121242)  **Nguyễn Lâm Quốc Bảo** (110121007)  Lớp: Công Nghệ Thông Tin A (DA21TTA)  Lớp theo TKB: DA21TTA |

**Trà Vinh, tháng 12 năm 2024**

# **LỜI CẢM ƠN**

Chúng tôi xin gửi lời cảm ơn sâu sắc đến Trường Đại học Trà Vinh đã tạo điều kiện cho chúng tôi thực hiện dự án này, những người đã đưa ra sự hỗ trợ nhỏ nhất cho chúng tôi dù là một lời gợi ý. Cũng như là sự hỗ trợ và giúp đỡ của cô Phạm Thị Trúc Mai người đã đóng vai trò quan trọng trong việc phát triển và làm cho dự án trở nên ngày càng thành công.

Chúng tôi biết ơn tất cả mọi người đã dành thời gian, công sức và kiến thức của mình để hỗ trợ chúng tôi. Sự giúp đỡ của các bạn là nguồn động lực giúp chúng tôi phát triển bản thân. Các ý kiến đóng góp và sự hợp tác của mọi người đã tạo nên một môi trường làm việc tích cực.

Không có gì quý báu hơn những người đồng đội, những người đã góp ý và những người đã hỗ trợ chúng tôi như mọi người đặc biệt là sự giúp đỡ của cô Phạm Thị Trúc Mai, người đã dìu dắt chúng tôi trong dự án này . Nhờ có cô và những người bạn đã giúp đỡ mà dự án của chúng tôi mới có thể phát triển đến mức này.Chúng tôi hạnh phúc và tự hào về mối quan hệ mà chúng ta đã xây dựng cùng nhau và tin rằng những thành công trong tương lai sẽ tiếp tục phát triển từ sự đồng lòng, cống hiến và sự giúp đỡ của mọi người đối với chúng tôi.

Một lần nữa, xin chân thành cảm ơn Trường Đại học Trà Vinh và cô Phạm Thị Trúc Mai đã giúp đỡ và chúng tôi mong sẽ tiếp tục nhận được sự ủng hộ của mọi người trong những dự án sắp tới.

Trân trọng,

Sinh viên ký và ghi rõ họ và tên

|  |  |  |
| --- | --- | --- |
| **Sinh viên 1** | **Sinh viên 2** | **Sinh viên 3** |

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

***Trà Vinh, ngày…...tháng…...năm 2012***

**GIÁO VIÊN CHẤM BÁO CÁO**

*(ký, ghi rõ họ tên)*

**NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN**

# **MỤC LỤC**

Trang

[**LỜI CẢM ƠN 2**](#_Toc187355509)

[**MỤC LỤC 1**](#_Toc187355510)

[**BẢNG PHÂN CÔNG CÔNG VIỆC 1**](#_Toc187355511)

[**BÁO CÁO KẾT QUẢ THỰC HIỆN TỪNG THÀNH VIÊN 2**](#_Toc187355512)

[**LÝ DO CHỌN ĐỀ TÀI 3**](#_Toc187355513)

[**Phần 1: CƠ SỞ LÝ THUYẾT 4**](#_Toc187355514)

[**1.1. GIỚI THIỆU VỀ HTML/CSS 4**](#_Toc187355515)

[**1.1.1. Giới thiệu về HTML 4**](#_Toc187355516)

[**1.1.2. Giới thiệu về CSS 5**](#_Toc187355517)

[**1.1.3. Các phương thức áp dụng CSS 5**](#_Toc187355518)

[**1.1.4. Cấu trúc cơ bản của CSS 5**](#_Toc187355519)

[**1.1.5. Các thuộc tính CSS phổ biến 5**](#_Toc187355520)

[**1.1.6. Responsive Web Design 6**](#_Toc187355521)

[**1.2. GIỚI THIỆU VỀ BACKEND PHP/LARAVEL 6**](#_Toc187355522)

[**1.2.1. Giới thiệu về PHP 6**](#_Toc187355523)

[**1.2.2. Cấu trúc và cách hoạt động của PHP 7**](#_Toc187355524)

[**1.2.3. Kết nối với cơ sở dữ liệu 8**](#_Toc187355525)

[**1.2.4. Các thư viện và framework PHP 8**](#_Toc187355526)

[**1.2.5. Giới thiệu về framework Laravel 8**](#_Toc187355527)

[**1.2.6. Lý do Laravel được ưa chuộng 10**](#_Toc187355528)

[**1.3. GIỚI THIỆU VỀ REACTJS 11**](#_Toc187355529)

[**Phần 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 14**](#_Toc187355530)

[**2.1. MÔ TẢ ĐỀ TÀI 14**](#_Toc187355531)

[**2.2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG 14**](#_Toc187355532)

[**2.2.1. Mô hình ERD/MCD/UML (Mô hình dữ liệu mức quan niệm) 14**](#_Toc187355533)

[**2.2.2. Mô hình MLD/Lược đồ CSDL (Mô hình dữ liệu mức logic, mức vật lý) 15**](#_Toc187355534)

[**Phần 3: XÂY DỰNG WEBSITE 20**](#_Toc187355535)

[**3.1. GIỚI THIỆU GIAO DIỆN 20**](#_Toc187355536)

[**3.1.1. Trang người dùng 20**](#_Toc187355537)

[**3.1.2. Trang quản trị 23**](#_Toc187355538)

[**3.2. CÁC CHỨC NĂNG CỦA WEBSITE 25**](#_Toc187355539)

[**3.2.1. Chức năng 1: Chức năng đăng nhập, đăng ký 25**](#_Toc187355540)

[**3.2.2. Chức năng 2: Chức năng giỏ hàng 25**](#_Toc187355541)

[**3.2.3. Chức năng 3: Chức năng thanh toán 26**](#_Toc187355542)

[**3.2.4. Chức năng 4: Chức năng thống kê 26**](#_Toc187355543)

[**3.2.5. Chức năng 5: Chức năng quản lý sản phẩm 27**](#_Toc187355544)

[**3.2.6. Chức năng 6: Chức năng quản lý đơn hàng 27**](#_Toc187355545)

[**3.2.7. Chức năng 7: Chức năng quản lý thông tin cá nhân 27**](#_Toc187355546)

[**Phần 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 28**](#_Toc187355547)

[**4.1. KẾT QUẢ ĐẠT ĐƯỢC 28**](#_Toc187355548)

[**4.2. ƯU NHƯỢC ĐIỂM 28**](#_Toc187355549)

[**4.3. HƯỚNG PHÁT TRIỂN 29**](#_Toc187355550)

[**TÀI LIỆU THAM KHẢO 30**](#_Toc187355551)

**DANH MỤC TỪ VIẾT TẮT (nếu có)**

|  |  |
| --- | --- |
| Từ viết tắt | Giải nghĩa |
| API | Application Programming Interface - Giao diện lập trình ứng dụng, giúp các hệ thống giao tiếp với nhau. |
| CLI | Command Line Interface - Giao diện dòng lệnh, giúp thực hiện các tác vụ thông qua lệnh. |
| CSS | Cascading Style Sheets - Ngôn ngữ dùng để định kiểu cho các trang web. |
| CRUD | Create, Read, Update, Delete - Các thao tác cơ bản trong quản lý dữ liệu. |
| DBMS | Database Management System - Hệ quản trị cơ sở dữ liệu, giúp quản lý và xử lý dữ liệu hiệu quả. |
| HTML | HyperText Markup Language - Ngôn ngữ đánh dấu chuẩn dùng để cấu trúc nội dung trên các trang web. |
| HTTP | HyperText Transfer Protocol - Giao thức truyền tải siêu văn bản, dùng để giao tiếp giữa client và server. |
| MVC | Model-View-Controller - Kiến trúc phần mềm phân chia ứng dụng thành ba thành phần chính: Model, View và Controller. |
| ORM | Object-Relational Mapping - Công cụ ánh xạ giữa các đối tượng trong mã nguồn và cơ sở dữ liệu quan hệ. |
| PHP | Hypertext Preprocessor - Ngôn ngữ lập trình kịch bản phía máy chủ dùng để phát triển các ứng dụng web động. |
| URL | Uniform Resource Locator - Địa chỉ duy nhất để xác định vị trí của một tài nguyên trên internet. |

**DANH MỤC HÌNH ẢNH**

[Hình 1 Minh chứng hoạt động github 2](#_Toc187410042)

[Hình 2. 1 Mô hình thực thể ERD 15](#_Toc187355552)

[Hình 2. 2 Mô hình dữ liệu mức vật lý 15](#_Toc187355553)

[Hình 2. 3 Mô hình dữ liệu mức logic 16](#_Toc187355554)

[Hình 3. 1 Giao diện đăng nhập 20](#_Toc187409593)

[Hình 3. 2 Giao diện đăng ký 20](#_Toc187409594)

[Hình 3. 3 Giao diện trang chủ 21](#_Toc187409595)

[Hình 3. 4 Giao diện giỏ hàng 21](#_Toc187409596)

[Hình 3. 5 Giao diện thanh toán 22](#_Toc187409597)

[Hình 3. 6 Giao diện thông tin cá nhân 22](#_Toc187409598)

[Hình 3. 7 Giao diện quản lý đơn hàng cá nhân 22](#_Toc187409599)

[Hình 3. 8 Giao diện chi tiết sản phẩm đã đặt 23](#_Toc187409600)

[Hình 3. 9 Giao diện thống kê quản trị 23](#_Toc187409601)

[Hình 3. 10 Giao diện quản trị người dùng 24](#_Toc187409602)

[Hình 3. 11 Giao diện quản lý sản phẩm 24](#_Toc187409603)

[Hình 3. 12 Giao diện quản lý danh mục 25](#_Toc187409604)

[Hình 3. 13 Giao diện đăng nhập 25](#_Toc187409605)

[Hình 3. 14 Giao diện đăng ký 26](#_Toc187409606)

[Hình 3. 15 Giao diện giỏ hàng 26](#_Toc187409607)

[Hình 3. 16 Giao diện thanh toán 26](#_Toc187409608)

[Hình 3. 17 Giao diện thống kê 27](#_Toc187409609)

[Hình 3. 18 Giao diện quản lý sản phẩm 27](#_Toc187409610)

[Hình 3. 19 Giao diện quản lý đơn hàng 28](#_Toc187409611)

[Hình 3. 20 Giao diện chi tiết đơn hàng 28](#_Toc187409612)

[Hình 3. 21 Quản lý thông tin cá nhân 28](#_Toc187409613)

**DANH MỤC BẢNG BIỂU (nếu có)**

[Bảng 1. 1 Một số thẻ HTML thông dụng 6](#_Toc187355575)

[Bảng 1. 2 Một số tính năng sử dụng trong bài báo cáo 10](#_Toc187355576)

[Bảng 2. 1 Chi tiết các bảng trong hệ thống 16](#_Toc187355577)

[Bảng 2. 2 Bảng lưu trữ thông tin chi tiết hóa đơn 17](#_Toc187355578)

[Bảng 2. 3 Bảng lưu trữ thông tin giỏ hàng 17](#_Toc187355579)

[Bảng 2. 4 Bảng lưu trữ thông tin hóa đơn 17](#_Toc187355580)

[Bảng 2. 5 Bảng lưu trữ thông tin khách hàng 17](#_Toc187355581)

[Bảng 2. 6 Bảng lưu trữ thông tin các phân quyền 18](#_Toc187355582)

[Bảng 2. 7 Bảng lưu trữ thông tin sản phẩm 18](#_Toc187355583)

[Bảng 2. 8 Bảng lưu trữ thông tin tài khoản 19](#_Toc187355584)

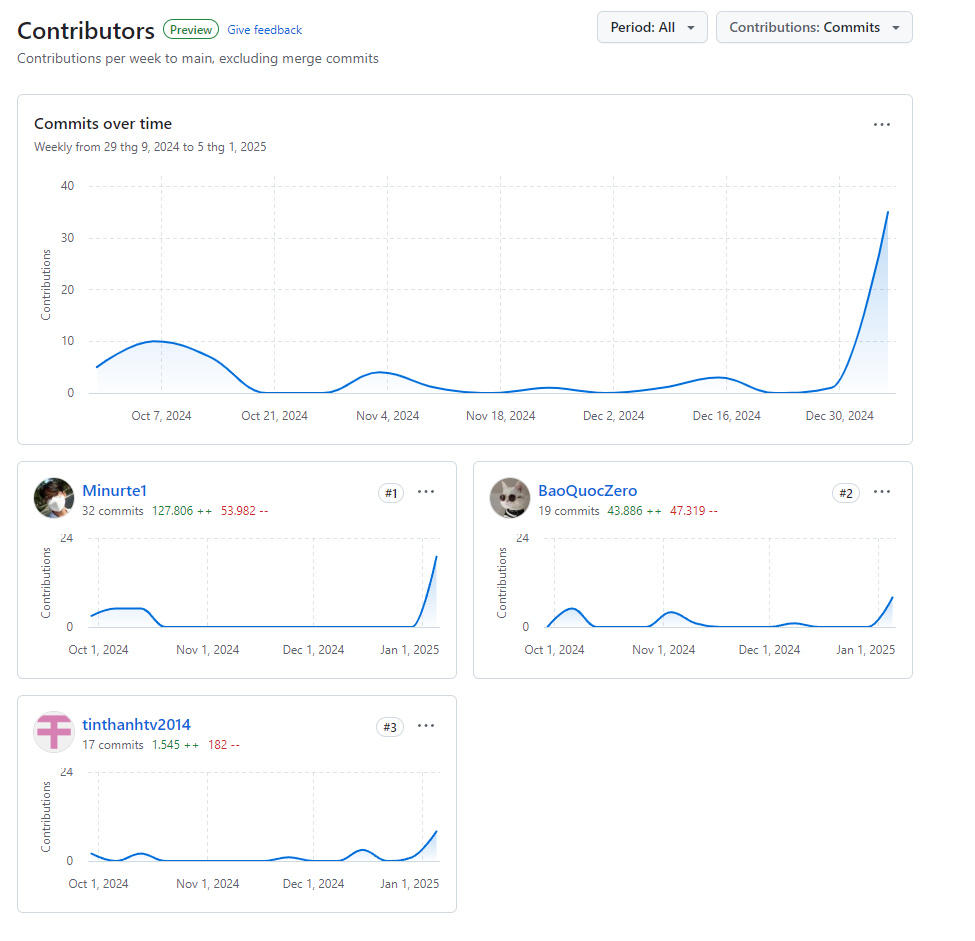
[Bảng 2. 9 Bảng lưu trữ thông tin thể loại 19](#_Toc187355585)

# **BẢNG PHÂN CÔNG CÔNG VIỆC**

|  |  |  |
| --- | --- | --- |
| **HỌ TÊN SINH VIÊN** | **CÔNG VIỆC THỰC HIỆN** | **THỜI GIAN**  **THỰC HIỆN** |
| Nguyễn Tín Thành  110121104 | - Viết báo cáo  - Lập trình backend  - Thuyết trình | 01/11/2024  đến  10/01/2025 |
| Hồ Hoàng Phúc  110121242 | - Tạo slide powerpoint  - lập trình frontend  - Thuyết trình | 01/11/2024  đến  10/01/2025 |
| Nguyễn Lâm Quốc Bảo  110121007 | - Viết báo cáo  - Lập trình backend  - Thuyết trình | 01/11/2024  đến  10/01/2025 |

# **BÁO CÁO KẾT QUẢ THỰC HIỆN TỪNG THÀNH VIÊN**

[Báo cáo tóm tắt kết quả thực hiện và minh chứng cho công việc được phân công của từng thành viên trong nhóm. Gợi ý: chụp hình thể hiện thành quả đóng góp của từng thành viên trên GitHub.]



Hình 1 Minh chứng hoạt động github

# **LÝ DO CHỌN ĐỀ TÀI**

Trong bối cảnh công nghệ phát triển nhanh chóng, nhu cầu sử dụng máy tính cá nhân (PC) đang ngày càng gia tăng. Các cá nhân, doanh nghiệp, và tổ chức đều có nhu cầu mua sắm máy tính để phục vụ công việc, học tập, giải trí và các hoạt động khác. Tuy nhiên, việc lựa chọn sản phẩm phù hợp với nhu cầu và ngân sách đôi khi gặp phải nhiều khó khăn do có quá nhiều lựa chọn và thiếu thông tin rõ ràng.

Vì vậy, việc xây dựng một website bán PC nhằm cung cấp một nền tảng trực tuyến tiện lợi và dễ sử dụng cho người tiêu dùng là một nhu cầu thực tế. Website này sẽ không chỉ giúp người dùng dễ dàng tìm kiếm các sản phẩm máy tính phù hợp, mà còn cung cấp các thông tin chi tiết về cấu hình, tính năng, và giá cả của từng sản phẩm. Bằng cách này, người tiêu dùng có thể đưa ra quyết định mua sắm thông minh và hiệu quả hơn.

Website bán PC này không chỉ đáp ứng nhu cầu về sản phẩm mà còn chú trọng đến trải nghiệm người dùng (UX/UI), giúp người dùng dễ dàng duyệt các sản phẩm, so sánh, và thực hiện các giao dịch trực tuyến một cách nhanh chóng và an toàn.

# **Phần 1: CƠ SỞ LÝ THUYẾT**

1. GIỚI THIỆU VỀ HTML/CSS

### Giới thiệu về HTML

HTML (HyperText Markup Language) là ngôn ngữ đánh dấu chuẩn dùng để cấu trúc nội dung trên các trang web. HTML không phải là một ngôn ngữ lập trình mà là một ngôn ngữ đánh dấu, giúp xác định các thành phần cấu trúc của trang web và làm cho các trình duyệt web có thể hiểu và hiển thị các phần tử trên trang một cách chính xác.

#### **Cấu trúc cơ bản của HTML**

Một tài liệu HTML cơ bản có cấu trúc như sau:

* <!DOCTYPE html>: Khai báo loại tài liệu HTML5, giúp trình duyệt biết rằng trang sử dụng chuẩn HTML5.
* <html lang="vi">: Bao quanh toàn bộ nội dung của trang HTML, thuộc tính lang xác định ngôn ngữ của trang (trong trường hợp này là tiếng Việt).
* <head>: Phần đầu của tài liệu, chứa các thông tin metadata như charset, tiêu đề trang, liên kết đến CSS và JavaScript.
* <meta charset="UTF-8">: Đảm bảo trang sử dụng bộ mã hóa ký tự UTF-8, giúp hỗ trợ mọi ký tự, bao gồm các ký tự đặc biệt trong tiếng Việt.
* <meta name="viewport" content="width=device-width, initial-scale=1.0">: Thiết lập chiều rộng trang phù hợp với thiết bị di động, giúp trang web hiển thị tốt trên các màn hình khác nhau.
* <title>: Chứa tiêu đề trang web, được hiển thị trên tab của trình duyệt.
* <body>: Chứa nội dung chính của trang web, là nơi người dùng sẽ nhìn thấy các thành phần như văn bản, hình ảnh, liên kết.

HTML là ngôn ngữ cơ bản và không thể thiếu trong phát triển web. Dù không phải là ngôn ngữ lập trình, HTML giúp xây dựng cấu trúc và tổ chức nội dung của trang web, làm nền tảng cho các ngôn ngữ khác như CSS và JavaScript. Sự phát triển của HTML5 mang lại nhiều tính năng mới, giúp trang web hoạt động hiệu quả trên mọi thiết bị và hỗ trợ các tính năng đa phương tiện một cách mượt mà.

### Giới thiệu về CSS

CSS (Cascading Style Sheets) là một ngôn ngữ dùng để định kiểu cho các trang web. CSS cho phép tách biệt nội dung của trang web (được viết bằng HTML) và cách thức hiển thị của nó. Thông qua CSS, người phát triển web có thể dễ dàng thay đổi màu sắc, kích thước, font chữ, bố cục, hiệu ứng chuyển động, và rất nhiều yếu tố khác của các phần tử trên trang.

CSS giúp làm đẹp và tạo ra giao diện người dùng (UI) cho các website, cung cấp khả năng kiểm soát chi tiết các yếu tố trực quan của trang web mà không ảnh hưởng đến cấu trúc dữ liệu bên trong HTML. Điều này không chỉ giúp cải thiện trải nghiệm người dùng mà còn giúp việc duy trì mã nguồn trở nên dễ dàng và hiệu quả hơn.

### Các phương thức áp dụng CSS

CSS có thể được áp dụng vào HTML theo ba cách chính:

* Inline CSS: CSS được viết trực tiếp trong thuộc tính style của thẻ HTML. Phương pháp này thích hợp khi cần áp dụng một kiểu dáng duy nhất cho một phần tử cụ thể.
* Internal CSS: CSS được viết trong thẻ <style> trong phần <head> của tài liệu HTML. Phương pháp này thích hợp cho việc áp dụng các kiểu dáng cho toàn bộ trang web mà không cần tạo tệp CSS riêng biệt.
* External CSS: CSS được viết trong một tệp CSS riêng biệt và liên kết với tệp HTML thông qua thẻ <link> trong phần <head> của tài liệu. Phương pháp này giúp tái sử dụng các kiểu dáng trên nhiều trang khác nhau và giữ mã nguồn HTML gọn gàng.

### Cấu trúc cơ bản của CSS

Một quy tắc CSS cơ bản bao gồm hai phần: **Selector** (bộ chọn) và **Declaration Block** (khối khai báo).

* Selector: Chọn phần tử HTML mà muốn áp dụng kiểu dáng.
* Declaration Block: Chứa các thuộc tính và giá trị để thay đổi kiểu dáng của phần tử.

### Các thuộc tính CSS phổ biến

CSS cung cấp hàng nghìn thuộc tính để thay đổi giao diện của các phần tử HTML. Một số thuộc tính cơ bản thường được sử dụng bao gồm:

* Màu sắc và nền: Các thuộc tính như color, background-color, và background-image giúp thay đổi màu sắc của văn bản và nền của phần tử.
* Bố cục: Thuộc tính như width, height, margin, padding, và border giúp điều chỉnh kích thước, khoảng cách và viền của phần tử.
* Font chữ và văn bản: Các thuộc tính như font-family, font-size, font-weight, và text-align giúp kiểm soát kiểu chữ và cách văn bản được căn chỉnh.
* Vị trí: Thuộc tính position, top, right, bottom, và left giúp điều chỉnh vị trí của phần tử trong trang web.

### Responsive Web Design

Một trong những tính năng quan trọng của CSS là khả năng hỗ trợ **Responsive Web Design** (Thiết kế đáp ứng). Đây là phương pháp giúp các trang web tự động điều chỉnh giao diện để phù hợp với mọi loại thiết bị, từ máy tính để bàn cho đến điện thoại di động. Điều này được thực hiện chủ yếu thông qua các **media queries**, cho phép thay đổi kiểu dáng của trang web tùy thuộc vào kích thước màn hình của thiết bị người dùng.

CSS là công cụ không thể thiếu trong việc xây dựng giao diện người dùng cho các trang web. Nó giúp tạo ra các trang web đẹp mắt, dễ sử dụng và dễ bảo trì. Bằng cách tách biệt giao diện khỏi nội dung, CSS không chỉ làm cho mã nguồn gọn gàng hơn mà còn giúp việc thay đổi giao diện trở nên dễ dàng mà không cần phải chỉnh sửa nội dung HTML.

Bảng 1. 1 Một số thẻ HTML thông dụng

|  |  |
| --- | --- |
| **Tên thẻ** | **Chức năng** |
| <div> | Tạo khối |
| <h> | Tạo kích thước của đoạn text |
| <span> | Tạo nội dung cho đoạn text |

1. GIỚI THIỆU VỀ BACKEND PHP/LARAVEL

### Giới thiệu về PHP

PHP (Hypertext Preprocessor) là một ngôn ngữ lập trình kịch bản phía máy chủ (server-side scripting language) được sử dụng rộng rãi trong phát triển web. Được phát triển lần đầu tiên vào năm 1994 bởi Rasmus Lerdorf, PHP hiện nay đã trở thành một trong những ngôn ngữ phổ biến nhất để xây dựng các ứng dụng web động.

**Đặc điểm nổi bật của PHP**

* Mã nguồn mở: PHP là ngôn ngữ mã nguồn mở, có thể tải xuống và sử dụng miễn phí. Điều này giúp các lập trình viên có thể dễ dàng triển khai và phát triển ứng dụng mà không gặp phải chi phí bản quyền.
* Tính năng mạnh mẽ cho phát triển web: PHP được tích hợp tốt với các cơ sở dữ liệu như MySQL, PostgreSQL và SQLite, giúp việc xây dựng các ứng dụng web động (dynamic web applications) trở nên dễ dàng. PHP có thể xử lý các tác vụ như quản lý phiên làm việc (session management), gửi email, và xử lý biểu mẫu (form handling).
* Khả năng tương thích với các hệ điều hành khác nhau: PHP có thể chạy trên nhiều hệ điều hành, bao gồm Windows, Linux, và macOS, và tương thích với nhiều máy chủ web như Apache và Nginx.
* Cộng đồng lớn và tài liệu phong phú: Với sự phát triển mạnh mẽ và phổ biến của PHP, cộng đồng lập trình viên luôn hỗ trợ nhau với vô số tài liệu, hướng dẫn và thư viện mã nguồn mở. Điều này giúp cho việc phát triển và giải quyết vấn đề trở nên dễ dàng hơn.

### Cấu trúc và cách hoạt động của PHP

PHP có thể nhúng trực tiếp vào mã HTML, giúp tạo ra các trang web động một cách dễ dàng. Các đoạn mã PHP được viết trong các thẻ <?php ?>, và khi trang web được tải, mã PHP sẽ được xử lý trên máy chủ. Sau đó, kết quả của mã PHP sẽ được gửi đến trình duyệt của người dùng dưới dạng HTML thuần túy.

**Các ứng dụng Của PHP**

PHP chủ yếu được sử dụng trong việc phát triển các trang web động và ứng dụng web. Một số ứng dụng phổ biến của PHP bao gồm:

* Quản lý cơ sở dữ liệu: PHP có thể kết nối với các hệ quản trị cơ sở dữ liệu (DBMS) như MySQL hoặc PostgreSQL, giúp người dùng lưu trữ và truy xuất dữ liệu một cách hiệu quả.
* Hệ thống quản lý nội dung (CMS): Các hệ thống quản lý nội dung như WordPress, Joomla và Drupal đều được xây dựng bằng PHP, cho phép người dùng dễ dàng tạo và quản lý các trang web mà không cần phải viết mã từ đầu.
* E-commerce: PHP được sử dụng trong các ứng dụng thương mại điện tử để xử lý các giao dịch mua bán trực tuyến, quản lý giỏ hàng, và thanh toán trực tuyến.
* Các ứng dụng web khác: PHP còn được sử dụng trong nhiều loại ứng dụng web khác, từ các diễn đàn, blog đến các hệ thống quản lý học sinh, quản lý khách hàng, và nhiều hơn nữa.

### Kết nối với cơ sở dữ liệu

PHP có khả năng kết nối và tương tác với các hệ quản trị cơ sở dữ liệu (DBMS), giúp lưu trữ và truy xuất dữ liệu cho ứng dụng web. Một trong những hệ quản trị cơ sở dữ liệu phổ biến mà PHP hỗ trợ là MySQL.

### Các thư viện và framework PHP

PHP có một số thư viện và framework mạnh mẽ giúp việc phát triển ứng dụng web trở nên nhanh chóng và dễ dàng. Một số framework PHP phổ biến bao gồm:

* Laravel: Một framework PHP mạnh mẽ với các tính năng như ORM (Object-Relational Mapping), hệ thống routing, và các công cụ để phát triển nhanh chóng các ứng dụng web.
* Symfony: Một framework PHP hướng đối tượng được sử dụng để xây dựng các ứng dụng web phức tạp và dễ bảo trì.
* CodeIgniter: Một framework nhẹ và dễ sử dụng, thích hợp cho các dự án nhỏ và vừa.
* Zend Framework: Một framework mạnh mẽ, hỗ trợ các ứng dụng web phức tạp và các API.

### Giới thiệu về framework Laravel

Laravel là một framework mã nguồn mở (open-source) được viết bằng ngôn ngữ PHP, được tạo ra với mục tiêu giúp các lập trình viên xây dựng ứng dụng web nhanh chóng và hiệu quả. Laravel cung cấp một cấu trúc mã nguồn mạnh mẽ và dễ sử dụng, đồng thời giúp giảm thiểu những công việc lặp đi lặp lại trong quá trình phát triển phần mềm. Laravel nổi bật với việc mang lại một môi trường phát triển tiện lợi, dễ học, và có thể ứng dụng vào nhiều loại dự án, từ các ứng dụng nhỏ đến các hệ thống lớn, phức tạp.

**Các tính năng chính của Laravel**

* Routing và URL Management: Laravel cung cấp một hệ thống routing linh hoạt, giúp lập trình viên dễ dàng định nghĩa các tuyến đường (routes) trong ứng dụng. Các routes này có thể xử lý các yêu cầu HTTP và liên kết với các controller hoặc hành động cụ thể trong ứng dụng. Laravel cũng hỗ trợ quản lý URL đơn giản và mạnh mẽ.
* Migrations và Database Management: Laravel hỗ trợ hệ thống migration, giúp dễ dàng tạo và quản lý cấu trúc cơ sở dữ liệu. Với Laravel, có thể quản lý và thay đổi cấu trúc của cơ sở dữ liệu một cách dễ dàng mà không lo mất dữ liệu. Laravel cũng cung cấp một ORM (Object-Relational Mapping) mạnh mẽ mang tên Eloquent, giúp thao tác với cơ sở dữ liệu một cách trực quan, dễ dàng mà không cần viết quá nhiều mã SQL.
* Blade Template Engine: Laravel sử dụng một engine template tên là Blade, cho phép tách biệt phần logic và giao diện trong ứng dụng. Blade cung cấp một cú pháp đơn giản và dễ đọc để xử lý các phần tử HTML động, ví dụ như vòng lặp, điều kiện, và hiển thị dữ liệu từ controller vào view.
* Artisan CLI: Laravel đi kèm với một công cụ dòng lệnh mạnh mẽ tên là Artisan, giúp lập trình viên thực hiện các tác vụ thường xuyên như tạo controller, model, migration, seeder, v.v., mà không cần phải làm thủ công. Artisan giúp đơn giản hóa quá trình phát triển ứng dụng và tiết kiệm thời gian.
* Middleware: Middleware trong Laravel giúp xử lý các yêu cầu HTTP trước khi chúng được xử lý bởi các route hoặc controller. Đây là nơi có thể thực hiện các tác vụ như xác thực, kiểm tra quyền hạn người dùng, hoặc xử lý dữ liệu trước khi trả về cho người dùng.
* Authentication và Authorization: Laravel hỗ trợ hệ thống xác thực người dùng mạnh mẽ, bao gồm việc đăng nhập, đăng ký, quên mật khẩu, và bảo mật thông tin người dùng. Laravel cũng cung cấp các công cụ để kiểm tra quyền truy cập và xác thực quyền hạn người dùng (Authorization), giúp tạo ra các ứng dụng web an toàn và bảo mật.
* Queues và Jobs: Laravel hỗ trợ cơ chế xử lý các tác vụ nền (background jobs) thông qua hệ thống queues. Điều này cho phép các tác vụ tốn thời gian như gửi email, xử lý tệp tải lên, hoặc các tác vụ tính toán phức tạp có thể được xử lý ngoài luồng, giúp giảm tải cho ứng dụng và cải thiện hiệu suất.
* Testing: Laravel đi kèm với một bộ công cụ kiểm thử tích hợp, giúp lập trình viên dễ dàng viết các bài kiểm tra tự động cho ứng dụng của mình. Laravel hỗ trợ cả kiểm tra đơn vị (unit testing) và kiểm tra chức năng (feature testing), giúp đảm bảo rằng ứng dụng hoạt động đúng như mong đợi trong suốt quá trình phát triển.
* Task Scheduling: Laravel cung cấp một hệ thống lập lịch các tác vụ định kỳ dễ dàng thông qua cấu hình trong một file duy nhất. Điều này giúp lập trình viên dễ dàng lên lịch và quản lý các tác vụ tự động trong ứng dụng như sao lưu cơ sở dữ liệu, gửi email hàng ngày, v.v.
* Package Ecosystem: Laravel có một hệ sinh thái các package phong phú, bao gồm các thư viện bổ sung và các công cụ hữu ích cho nhiều mục đích khác nhau, từ xử lý thanh toán, gửi email, đến các công cụ tích hợp API. Điều này giúp lập trình viên có thể nhanh chóng mở rộng tính năng của ứng dụng mà không cần phải phát triển từ đầu.

### Lý do Laravel được ưa chuộng

* Cộng đồng lớn và hỗ trợ mạnh mẽ: Laravel có một cộng đồng người dùng và phát triển rất lớn, với nhiều tài nguyên hỗ trợ như tài liệu, diễn đàn, bài viết, video hướng dẫn. Điều này giúp người dùng dễ dàng tiếp cận và giải quyết các vấn đề gặp phải khi phát triển ứng dụng.
* Hỗ trợ MVC (Model-View-Controller): Laravel sử dụng mô hình kiến trúc MVC, giúp phân tách rõ ràng giữa các phần của ứng dụng (logic, giao diện người dùng và cơ sở dữ liệu), giúp mã nguồn dễ bảo trì, dễ mở rộng và dễ hiểu.
* Quản lý phụ thuộc đơn giản với Composer:Laravel tích hợp tốt với Composer, công cụ quản lý phụ thuộc trong PHP, giúp dễ dàng cài đặt và cập nhật các thư viện bên ngoài.
* Cập nhật thường xuyên: Laravel liên tục nhận được các bản cập nhật mới, bổ sung tính năng và cải thiện bảo mật, giúp framework luôn đáp ứng được yêu cầu của các dự án phát triển web hiện đại.
* Tính mở rộng và linh hoạt: Laravel được thiết kế để có thể mở rộng dễ dàng. Có thể thêm các tính năng, dịch vụ và packages bổ sung tùy theo nhu cầu của dự án.

Bảng 1. 2 Một số tính năng sử dụng trong bài báo cáo

|  |  |
| --- | --- |
| **STT** | **Tính năng** |
|  | Đăng nhập, đăng ký |
|  | Giỏ hàng |
|  | Thanh toán |
|  | Trang chủ |
|  | Thống kê |
|  | Quản trị |

1. GIỚI THIỆU VỀ REACTJS

React.js (thường được gọi đơn giản là React) là một thư viện JavaScript mã nguồn mở, được phát triển bởi Facebook. Nó cho phép xây dựng các giao diện người dùng (UI) động và có thể tái sử dụng được thông qua các component. React tập trung vào việc tối ưu hóa việc cập nhật và render giao diện khi dữ liệu thay đổi, nhờ vào cơ chế Virtual DOM.

React chủ yếu được sử dụng để xây dựng các ứng dụng single-page applications (SPA) và có thể được kết hợp với nhiều công nghệ khác như Redux, React Router, và các API khác để xây dựng ứng dụng phức tạp.

**Tính Năng Chính của React.js**

* Component-based architecture (Kiến trúc dựa trên Component): React sử dụng mô hình component để chia giao diện người dùng thành các phần nhỏ, tái sử dụng được. Mỗi component có thể có trạng thái riêng và có thể được kết hợp để tạo thành các ứng dụng phức tạp.
* Virtual DOM (DOM ảo): React sử dụng Virtual DOM, giúp giảm thiểu việc thao tác trực tiếp với DOM thật. Khi có sự thay đổi trong state hoặc props của một component, React sẽ cập nhật Virtual DOM trước, so sánh với DOM cũ, và chỉ thay đổi những phần cần thiết trong DOM thật. Điều này giúp cải thiện hiệu suất của ứng dụng.
* One-way data flow (Dòng dữ liệu một chiều): Dữ liệu trong React di chuyển theo một chiều: từ parent component xuống child component thông qua props. Điều này giúp dễ dàng kiểm soát luồng dữ liệu và tránh được các vấn đề khó kiểm soát như trong các framework khác với data-binding hai chiều.
* JSX (JavaScript XML): React sử dụng JSX, một cú pháp mở rộng cho JavaScript cho phép viết HTML trong JavaScript. JSX giúp dễ dàng xác định cấu trúc giao diện người dùng trong code JavaScript.
* State và Props: State là dữ liệu có thể thay đổi trong một component. Mỗi khi state thay đổi, React sẽ tự động render lại component để phản ánh sự thay đổi đó. Props là các tham số truyền vào từ component cha (parent component) vào component con (child component). Props không thể thay đổi trong component nhận, và chúng giúp truyền tải dữ liệu giữa các component.
* Hooks: Hooks là một tính năng mới trong React (bắt đầu từ phiên bản 16.8) cho phép sử dụng state và các tính năng khác của React mà không cần phải viết class components. Các hooks phổ biến như useState, useEffect, useContext giúp xử lý trạng thái và side effects trong component một cách dễ dàng.
* React Router: React Router là một thư viện cho phép định tuyến (routing) trong các ứng dụng React, giúp chuyển hướng giữa các trang mà không cần tải lại toàn bộ trang, mang lại trải nghiệm người dùng mượt mà.
* Context API: Context API trong React giúp quản lý trạng thái ứng dụng mà không cần phải truyền dữ liệu qua nhiều lớp component. Nó rất hữu ích trong việc chia sẻ dữ liệu giữa các component mà không cần props drilling.

**Lợi Ích và Ưu Điểm của React**

* Tái sử dụng mã nguồn (Reusability): Các component trong React có thể được tái sử dụng nhiều lần trong ứng dụng, giúp tiết kiệm thời gian và công sức phát triển.
* Hiệu suất cao: Nhờ vào Virtual DOM, React có khả năng tối ưu hóa việc cập nhật giao diện người dùng, giúp ứng dụng nhanh hơn và mượt mà hơn so với các framework sử dụng DOM thật.
* Dễ học và dễ sử dụng: Nếu đã quen thuộc với JavaScript, React dễ dàng học và làm quen. JSX cung cấp một cú pháp dễ hiểu, gần gũi với HTML.
* Cộng đồng mạnh mẽ: React có một cộng đồng rất lớn và phát triển mạnh mẽ, với rất nhiều tài liệu học tập, thư viện hỗ trợ và các giải pháp đã được kiểm chứng.
* Quản lý trạng thái (State management): React hỗ trợ các thư viện quản lý trạng thái như Redux và Context API, giúp dễ dàng quản lý dữ liệu trong các ứng dụng lớn và phức tạp.
* Kết hợp tốt với các công nghệ khác: React có thể dễ dàng kết hợp với các công nghệ khác như Node.js, GraphQL, Firebase, giúp xây dựng các ứng dụng full-stack mạnh mẽ.
* Hỗ trợ cho các ứng dụng di động: React Native là một framework dựa trên React cho phép phát triển ứng dụng di động cho cả iOS và Android từ một codebase duy nhất.

**Công Cụ và Thư Viện Hữu Ích Khi Làm Việc Với React**

* Create React App: Là công cụ giúp nhanh chóng khởi tạo một dự án React mà không cần phải cấu hình Webpack, Babel, hay các công cụ build khác.
* Cài đặt Create React App: npx create-react-app my-app
* Redux: Thư viện giúp quản lý trạng thái của ứng dụng ở mức toàn cục (global state), rất hữu ích cho các ứng dụng phức tạp với nhiều state và nhiều component cần truy cập.
* React Router: Thư viện quản lý điều hướng (routing) trong ứng dụng React, giúp tạo các trang và chuyển hướng giữa các trang mà không cần phải tải lại toàn bộ trang.
* Axios: Thư viện giúp gửi các yêu cầu HTTP dễ dàng từ frontend để kết nối với backend API.

**Lợi Ích và Hạn Chế của React.js**

**Lợi ích:**

* Tái sử dụng component giúp tiết kiệm thời gian phát triển.
* Virtual DOM giúp tăng hiệu suất ứng dụng.
* Cộng đồng lớn với nhiều tài nguyên hỗ trợ.
* Quản lý trạng thái dễ dàng với Redux hoặc Context API.

**Hạn chế:**

* Cần thời gian học hỏi đối với người mới bắt đầu, nhất là khi sử dụng các tính năng như hooks, context hoặc Redux.
* JSX có thể khiến người mới cảm thấy lạ lẫm vì cú pháp HTML kết hợp với Javascript.

# **Phần 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG**

1. MÔ TẢ ĐỀ TÀI

Giới thiệu chung: Đề tài "Xây dựng Website Bán PC" nhằm phát triển một nền tảng thương mại điện tử chuyên cung cấp các sản phẩm PC, máy tính và linh kiện điện tử. Website này sẽ giúp người dùng dễ dàng tìm kiếm, lựa chọn, và mua sắm các sản phẩm PC, từ các bộ máy tính hoàn chỉnh đến các linh kiện, phụ kiện hỗ trợ. Người dùng sẽ có thể tham khảo thông tin chi tiết về sản phẩm, đánh giá từ các khách hàng khác, và thực hiện các giao dịch mua hàng trực tuyến.

**Chức năng chính của website**:

* Trang chủ: Hiển thị các sản phẩm nổi bật và thông tin về các sản phẩm mới nhất.
* Danh mục sản phẩm: Cung cấp các bộ lọc cho phép người dùng tìm kiếm sản phẩm theo các tiêu chí như loại sản phẩm, giá cả, thương hiệu, v.v.
* Chi tiết sản phẩm: Mỗi sản phẩm sẽ có trang riêng với đầy đủ thông tin chi tiết, bao gồm mô tả, hình ảnh, giá, thông số kỹ thuật, đánh giá, và các sản phẩm liên quan.
* Giỏ hàng: Cho phép người dùng thêm các sản phẩm vào giỏ hàng, chỉnh sửa số lượng và thanh toán đơn hàng.
* Tài khoản người dùng: Cho phép người dùng đăng ký, đăng nhập, theo dõi đơn hàng và quản lý thông tin cá nhân.

**Công nghệ sử dụng:**

* Frontend: Sử dụng thư viện Reactjs để tạo giao diện người dùng tương tác.
* Backend: Sử dụng framework của PHP là Laravel với cơ sở dữ liệu MySQL.
* Bảo mật: Đảm bảo an toàn thông tin khách hàng và bảo mật các giao dịch thanh toán.
* Lợi ích: Website sẽ giúp kết nối người mua và người bán trong lĩnh vực PC, máy tính và linh kiện điện tử, tăng cường khả năng tiếp cận khách hàng, cải thiện trải nghiệm mua sắm trực tuyến và thúc đẩy doanh số bán hàng.

1. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

### Mô hình ERD/MCD/UML (Mô hình dữ liệu mức quan niệm)

Mô hình ERD



Hình 2. 1 Mô hình thực thể ERD

### Mô hình MLD/Lược đồ CSDL (Mô hình dữ liệu mức logic, mức vật lý)

Mô hình dữ liệu mức vật lý



Hình 2. 2 Mô hình dữ liệu mức vật lý

Mô hình dữ liệu mức logic



Hình 2. 3 Mô hình dữ liệu mức logic

Bảng 2. 1 Chi tiết các bảng trong hệ thống

|  |  |
| --- | --- |
| Tên bảng | Diễn giải |
| Chi\_tiet\_hoa\_don | Chi tiết hóa đơn |
| Gio\_hang | Giỏ hàng |
| Hoadon | Hóa đơn |
| Khachhang | Khách hàng |
| Phan\_quyen | Phân quyền |
| Sanpham | Sản phẩm |
| Tai\_khoan | Tài khoản |
| Theloai | Thể loại |

Bảng 2. 2 Bảng lưu trữ thông tin chi tiết hóa đơn

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Ràng buộc** |
| MA\_CTHD | Mã chi tiết hóa đơn | Int | Khóa chính |
| MASP | Mã sản phẩm | Int | Khóa ngoại |
| MAHD | Mã hóa đơn | Int | Khóa ngoại |
| SO\_LUONG | Số lượng | Int |  |
| GIAM\_GIA | Giảm giá | Varchar |  |
| GHI\_CHU\_CTHD | Ghi chú | Text |  |

Bảng 2. 3 Bảng lưu trữ thông tin giỏ hàng

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Ràng buộc** |
| MA\_KH | Mã khách hàng | Int | Khóa chính |
| MASP | Mã sản phẩm | Int | Khóa chính, khóa ngoại |
| SO\_LUONG\_SP | Số lượng sản phẩm | Int |  |

Bảng 2. 4 Bảng lưu trữ thông tin hóa đơn

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Ràng buộc** |
| MAHD | Mã hóa đơn | Int | Khóa chính |
| MA\_KH | Mã khách hàng | Int | Khóa ngoại |
| DIA\_CHI\_SHIP | Địa chỉ ship | Varchar |  |
| SDT\_LIEN\_HE\_KH | Số điện thoại | Varchar |  |
| GHI\_CHU\_HOA\_DON | Ghi chú |  |  |

Bảng 2. 5 Bảng lưu trữ thông tin khách hàng

| **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Ràng buộc** |
| --- | --- | --- | --- |
| MA\_KH | Mã khách hàng | Int | Khóa chính |
| SDT\_KH | Số điện thoại khách hàng | Varchar |  |
| TEN\_KHACH\_HANG | Tên khách hàng | Varchar |  |
| DIA\_CHI | Địa chỉ | Varchar |  |
| GHI\_CHU\_KH | Ghi chú | Text |  |

Bảng 2. 6 Bảng lưu trữ thông tin các phân quyền

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Ràng buộc** |
| MA\_PHAN\_QUYEN | Mã phân quyền | Int | Khóa chính |
| TEN\_PHAN\_QUYEN | Tên phân quyền | Varchar |  |
| GHI\_CHU\_PHAN\_QUYEN | Ghi chú | Varchar |  |

Bảng 2. 7 Bảng lưu trữ thông tin sản phẩm

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Ràng buộc** |
| MASP | Mã sản phẩm | Int | Khóa chính |
| MATL | Mã thể loại | Int | Khóa ngoại |
| TENSP | Tên sản phẩm | Varchar |  |
| DON\_GIA | Đơn giá | Varchar |  |
| TON\_KHO\_SP | Tồn kho | int |  |
| CHIP | Con chíp | Varchar |  |
| Main | Mainboard | Varchar |  |
| VGA | Card màn hình | Varchar |  |
| NHA\_SAN\_XUAT | Nhà sản xuất | Varchar |  |
| RAM | Ram | Varchar |  |
| ROM | Rom | Varchar |  |
| ANHSP | ảnh sản phẩm | Text |  |
| GHI\_CHU\_SP | Ghi chú | Text |  |

Bảng 2. 8 Bảng lưu trữ thông tin tài khoản

| **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Ràng buộc** |
| --- | --- | --- | --- |
| MA\_TK | Mã tài khoản | Int | Khóa chính |
| MA\_PHAN\_QUYEN | Mã phân quyền | Int | Khóa ngoại |
| MA\_KH | Mã khách hàng | Int | Khóa ngoại |
| TEN\_DANG\_NHAP | Tên đăng nhập | Varchar |  |
| MAT\_KHAU | Mật khẩu | Varchar |  |

Bảng 2. 9 Bảng lưu trữ thông tin thể loại

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **Ràng buộc** |
| MATL | Mã thể loại | Int | Khóa chính |
| TENTL | Tên thể loại | Varchar |  |
| MO\_TA\_TL | Mô tả thể loại | Text |  |
| GHI\_CHU\_TL | Ghi chú | Text |  |

# **Phần 3: XÂY DỰNG WEBSITE**

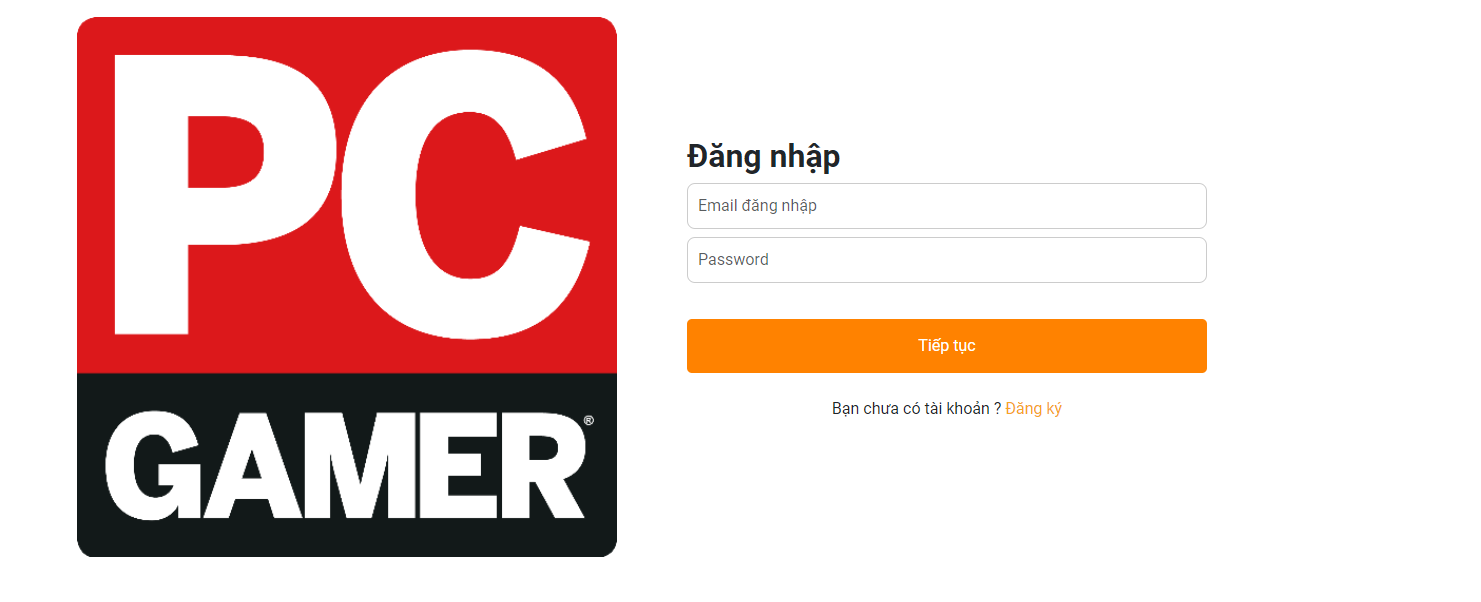
1. GIỚI THIỆU GIAO DIỆN

### Trang người dùng

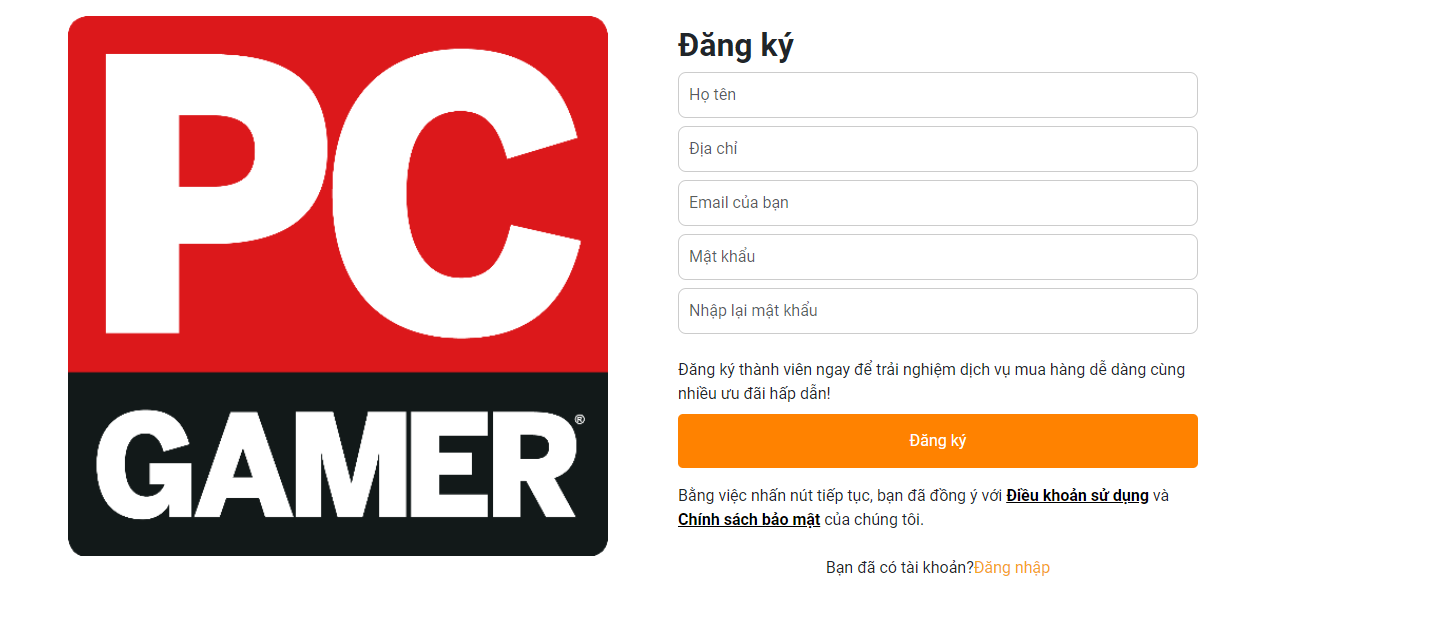
* 1. **Giao diện đăng nhập, đăng ký**

Đây là trang đăng nhập cho người dùng, có hai chức năng chính là đăng nhập và đăng ký, khi đăng nhập cần phải nhập email và mật khẩu để tiến hành vào hệ thống

Trường hợp không có tài khoản người dùng có thể bấm vào nút đăng ký để đăng ký tài khoản mới, người dùng cần phải điền tên, địa chỉ, email, mật khẩu và xác nhận mật khẩu để đăng ký



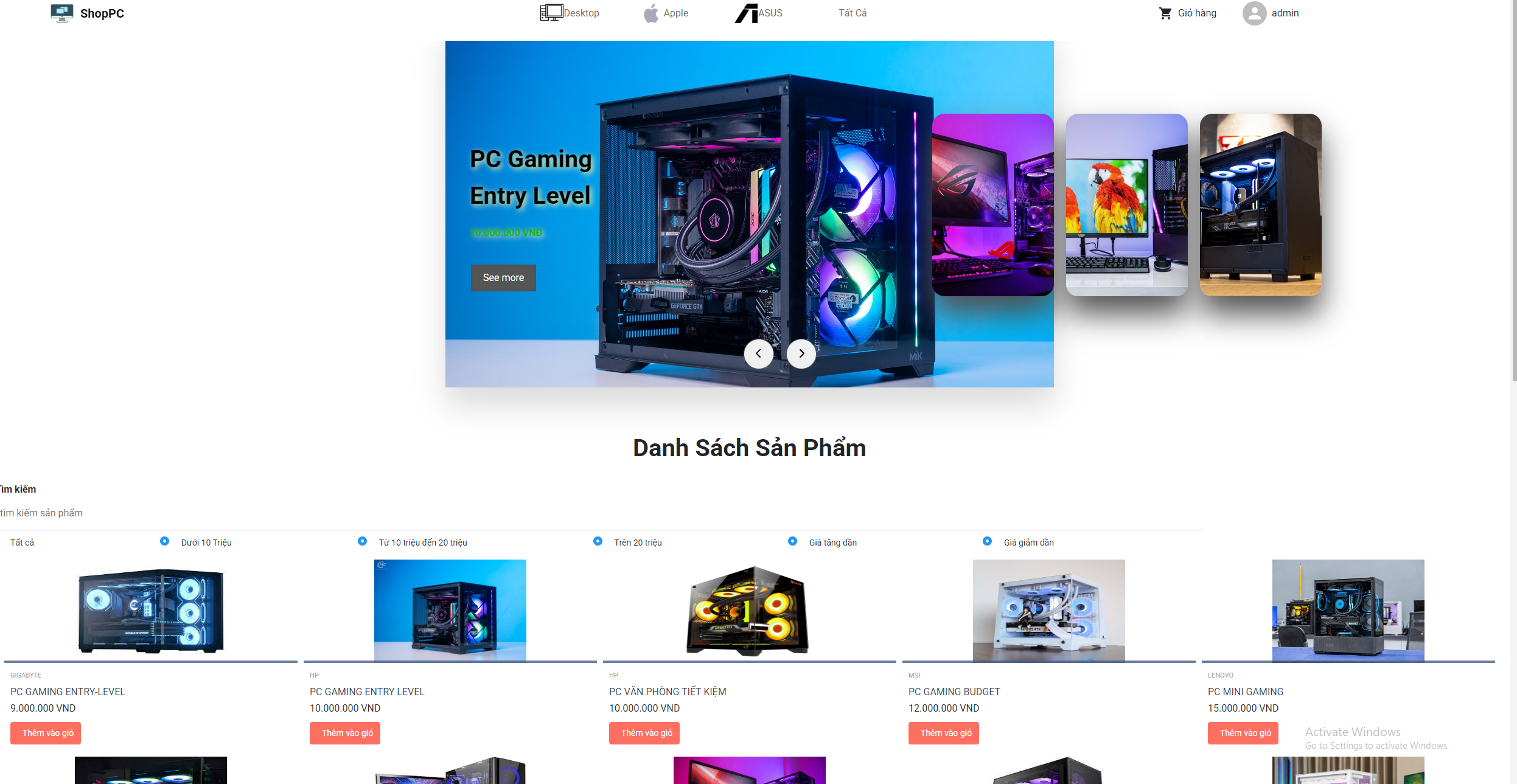
Hình 3. 1 Giao diện đăng nhập



Hình 3. 2 Giao diện đăng ký

* 1. **Giao diện trang chủ**

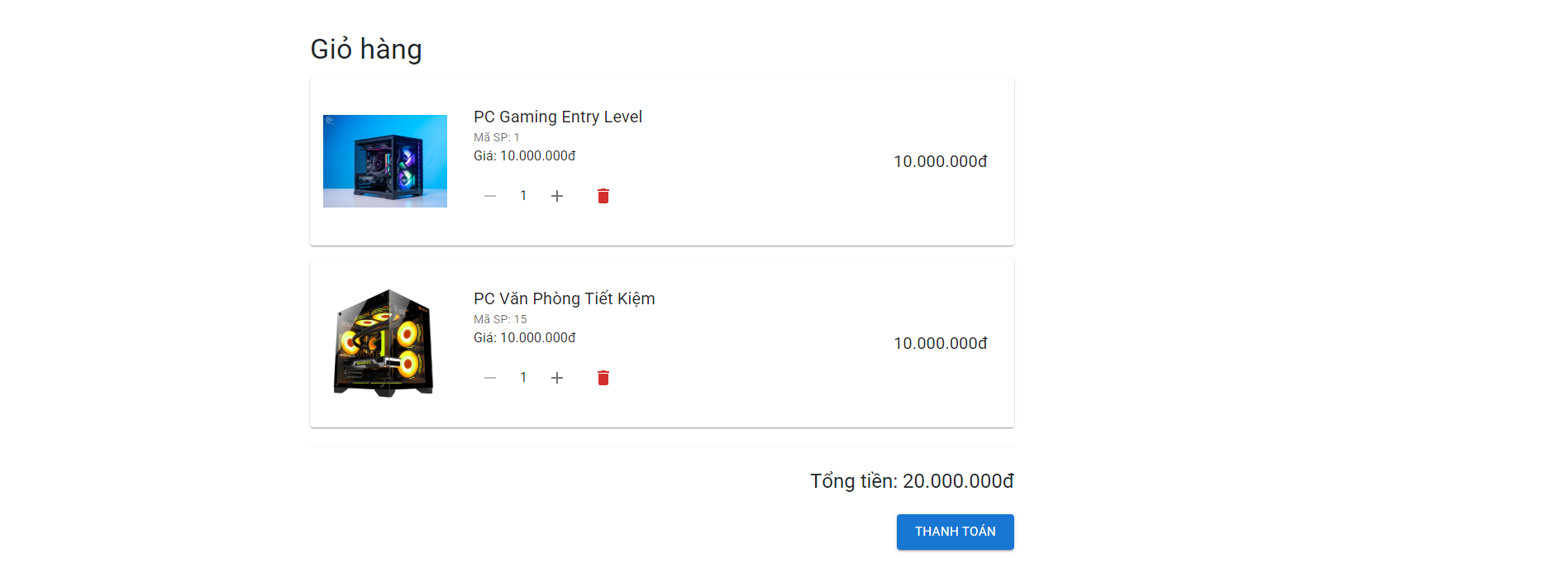
Giao diện trang chủ của website sẽ gồm slide và danh sách sản phẩm có các bộ lọc và sắp xếp theo giá tiền, ngoài ra còn có xem sản phẩm theo hãng.



Hình 3. 3 Giao diện trang chủ

* 1. **Giao diện giỏ hàng**

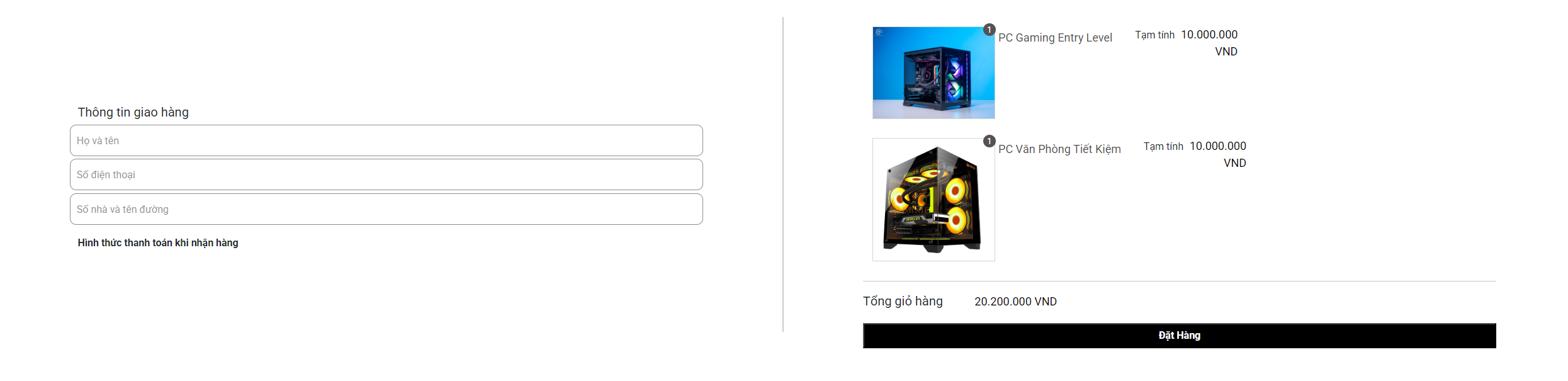
Giao diện giỏ hàng bên trong có số lượng sản phẩm, giá tiền, tổng tiền cần phải thanh toán, nếu đã xác nhận mua thì chỉ cần bấm nút thanh toán.



Hình 3. 4 Giao diện giỏ hàng

* 1. **Giao diện thanh toán**

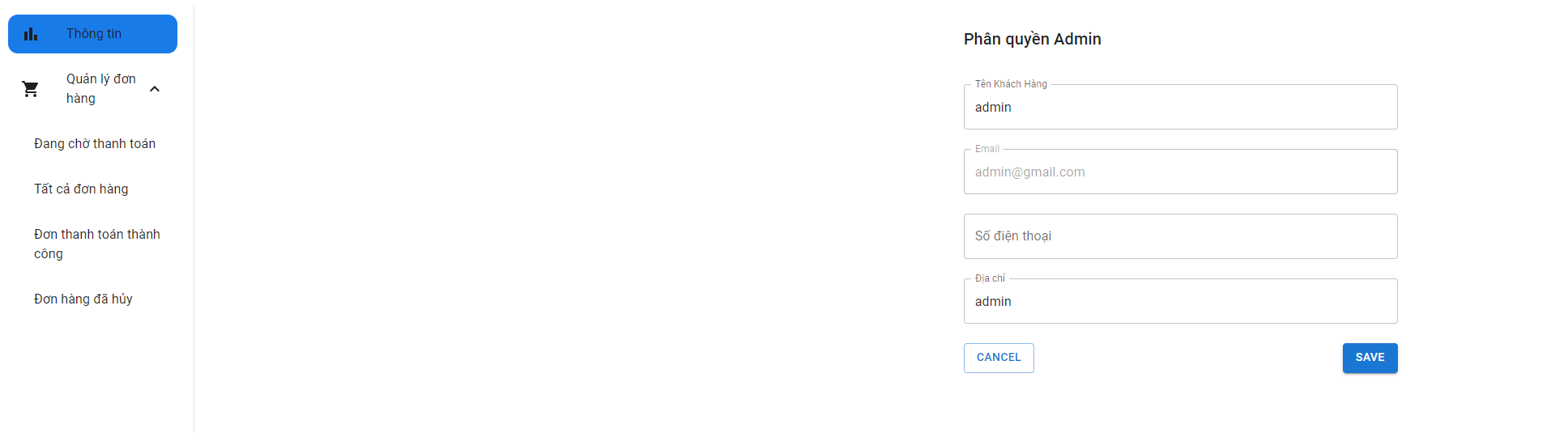
Giao diện xác nhận thanh toán sản phẩm, cần có các thông tin cơ bản như họ tên, số điện thoại, địa chỉ giao hàng và ghi chú xác nhận, khi đã xác nhận chỉ cần bấm nút đặt hàng sẽ đặt được.



Hình 3. 5 Giao diện thanh toán

* 1. **Giao diện thông tin cá nhân**

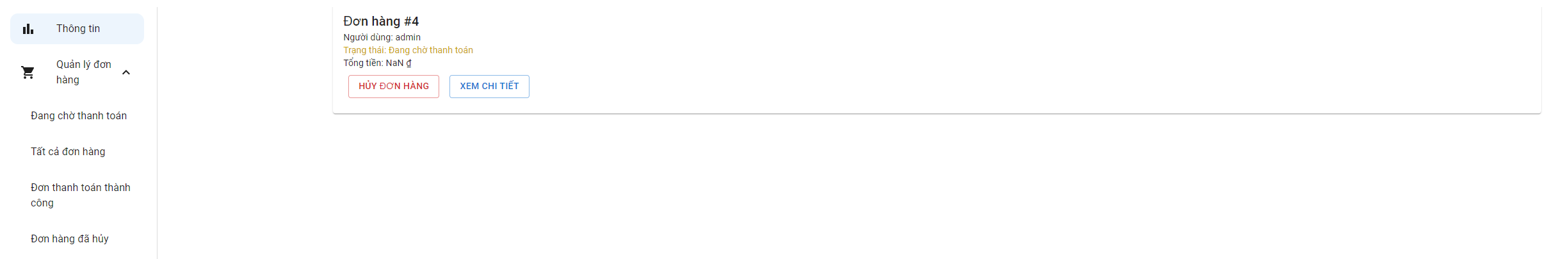
Giao diện cập nhật thông tin cá nhân cho biết thông tin về tên của người dùng, email, số điện thoại và địa chỉ của khách hàng.



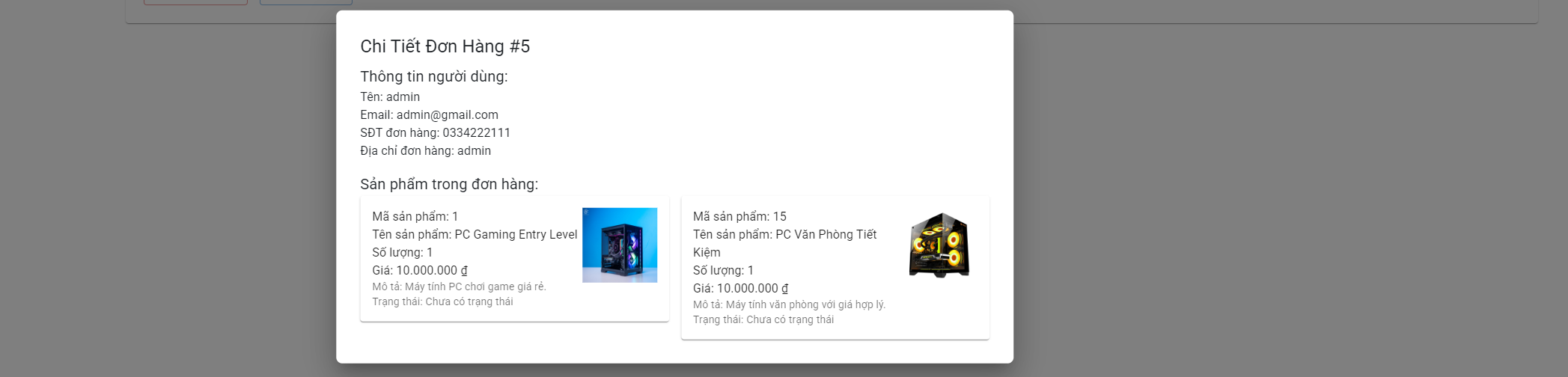
Hình 3. 6 Giao diện thông tin cá nhân

* 1. **Giao diện quản lý đơn hàng cá nhân**

Giao diện quản lý đơn hàng bên trong sẽ quản lý được đơn hàng của bản thân, có thể hủy đơn hàng và xem chi tiết đơn hàng của mình



Hình 3. 7 Giao diện quản lý đơn hàng cá nhân

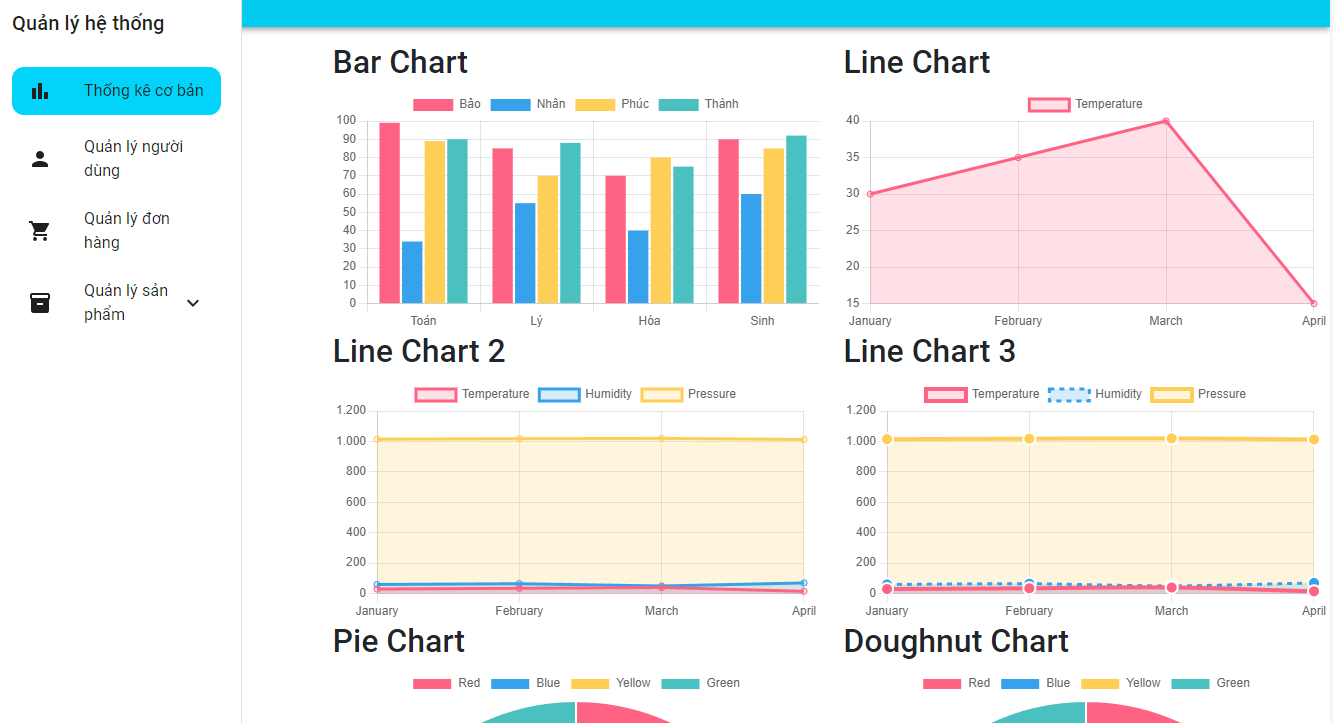


Hình 3. 8 Giao diện chi tiết sản phẩm đã đặt

### Trang quản trị

* 1. **Trang thống kê cơ bản**

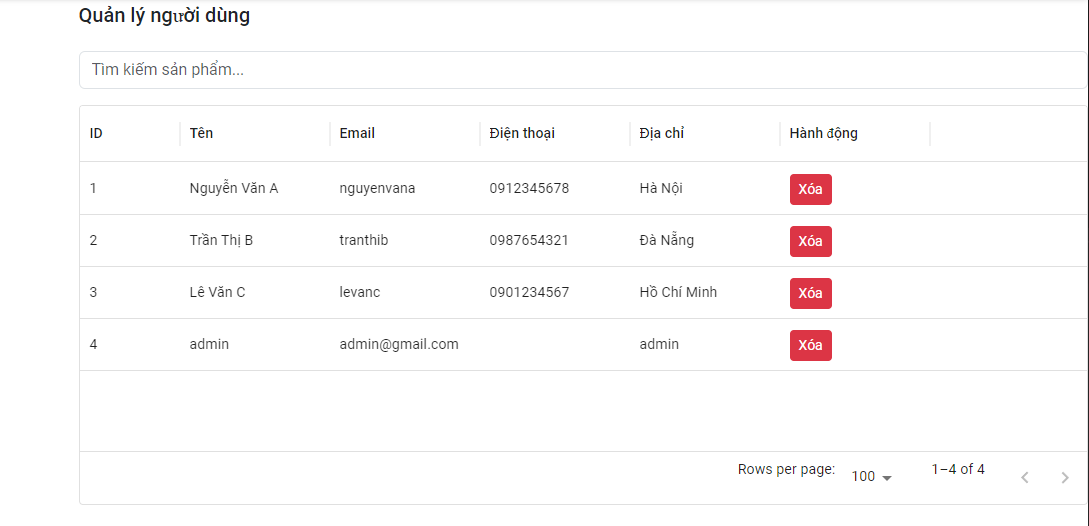
Giao diện thống kê cơ bản của quản trị viên, sẽ có các biểu đồ



Hình 3. 9 Giao diện thống kê quản trị

* 1. **Trang quản lý người dùng**

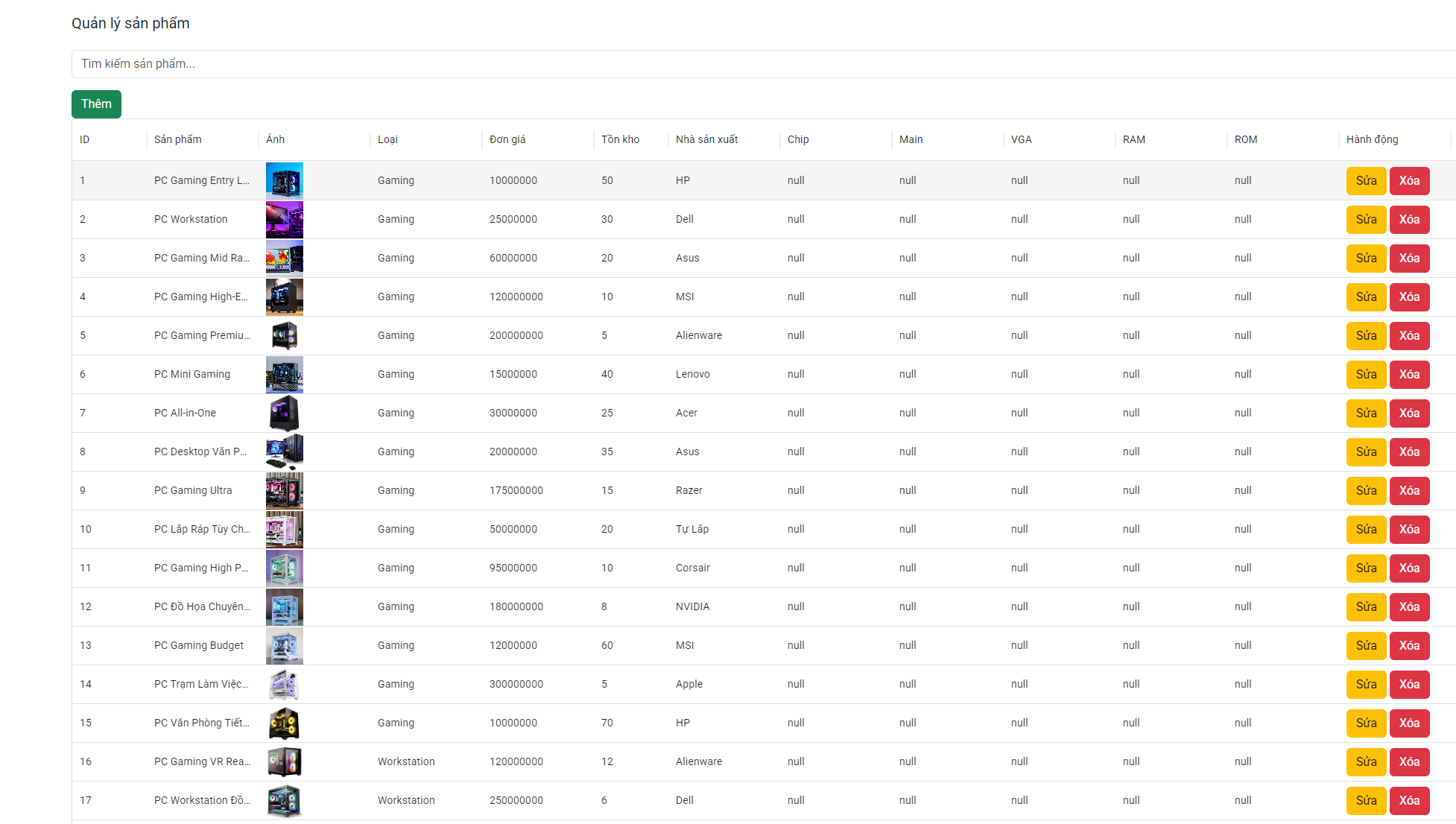
Giao diện quản lý người dùng sẽ xem thông tin quản lý của tất cả người dùng hiện tại



Hình 3. 10 Giao diện quản trị người dùng

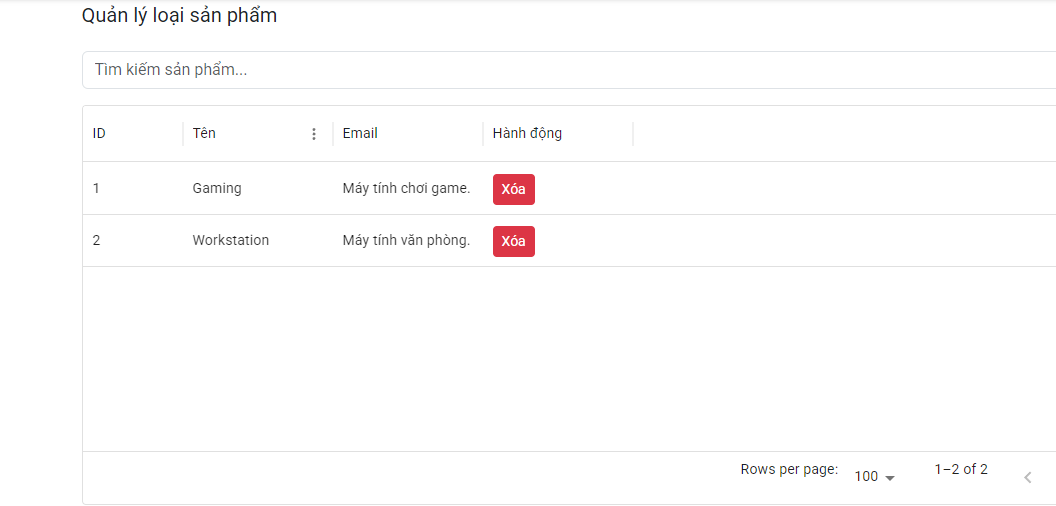
* 1. **Giao diện quản lý sản phẩm, quản lý danh mục sản phẩm**

Giao diện quản lý sản phẩm có thể thêm sản phẩm mới, xem thông tin của sản phẩm hiện tại và có thể xóa sản phẩm đó đi



Hình 3. 11 Giao diện quản lý sản phẩm

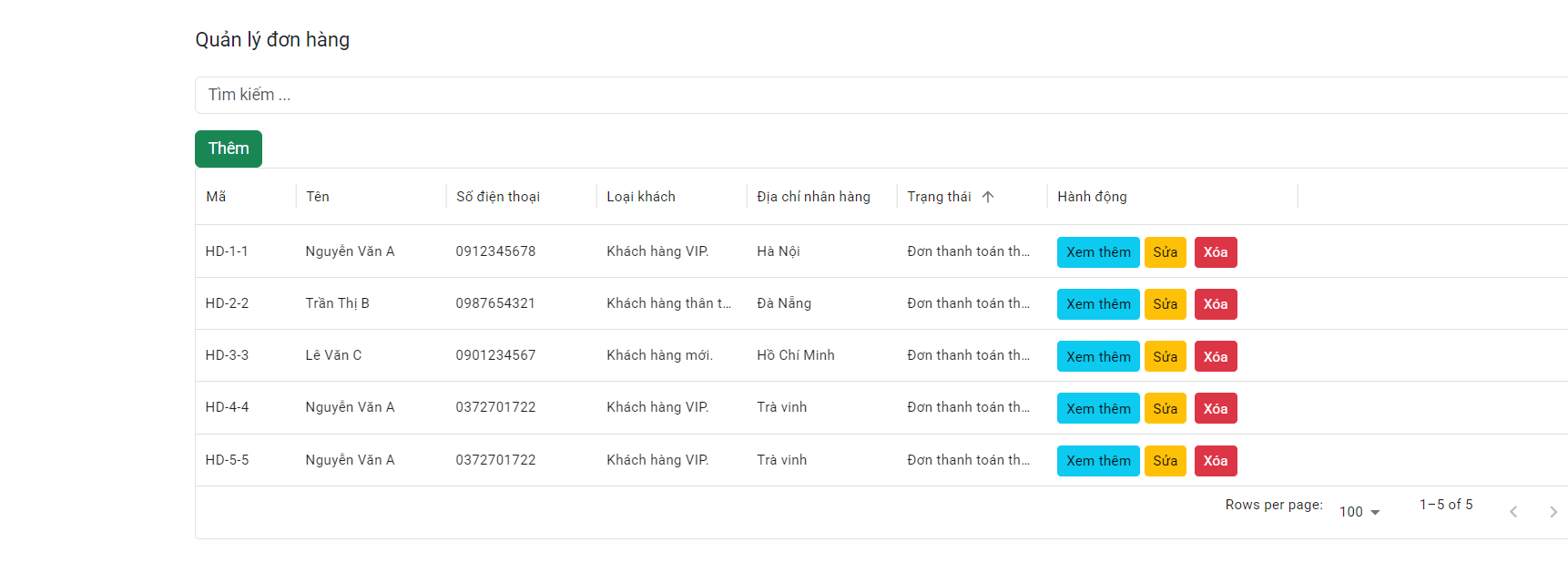
Giao diện quản lý danh mục sản phẩm sẽ có thể quản lý các danh mục của sản phẩm



Hình 3. 12 Giao diện quản lý danh mục

* 1. **Giao diện quản lý đơn hàng**

Giao diện quản lý đơn hàng của admin có thể xem thông tin các đơn hàng đã được đặt với nhiều trạng thái khác nhau, có thể xem thông tin của đơn hàng đó.



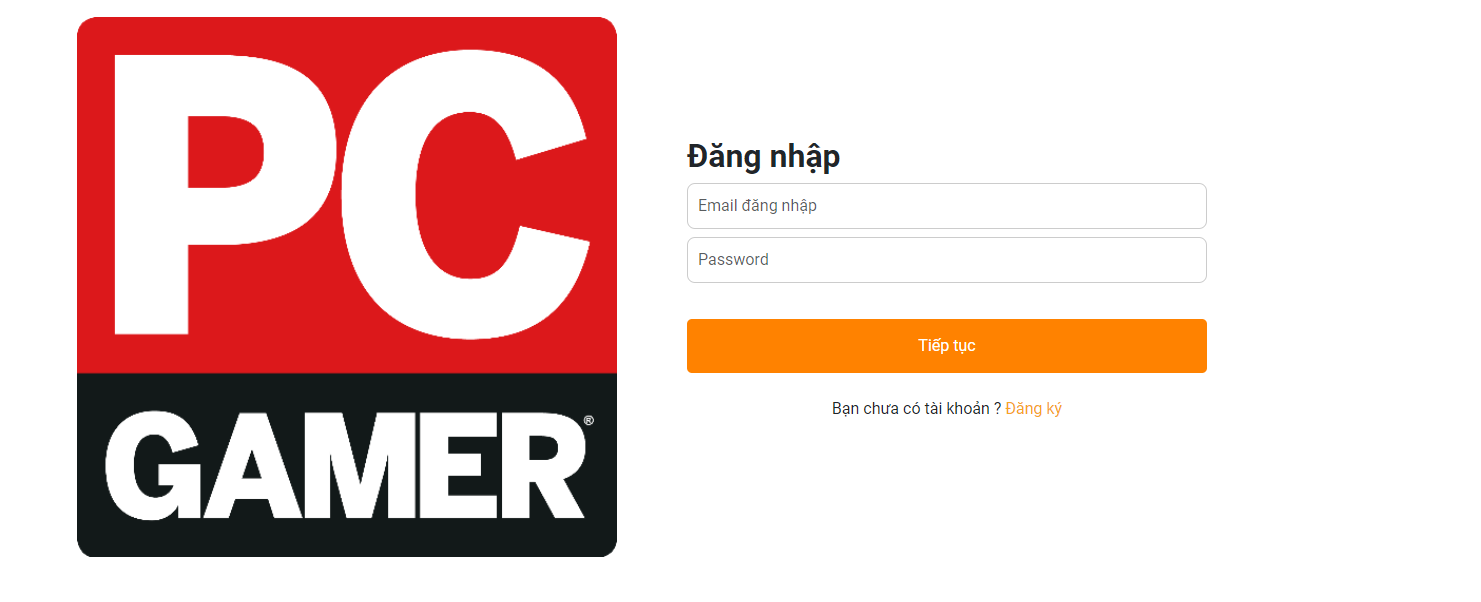
Hình 3. 13 Giao diện quản lý đơn hàng

1. CÁC CHỨC NĂNG CỦA WEBSITE

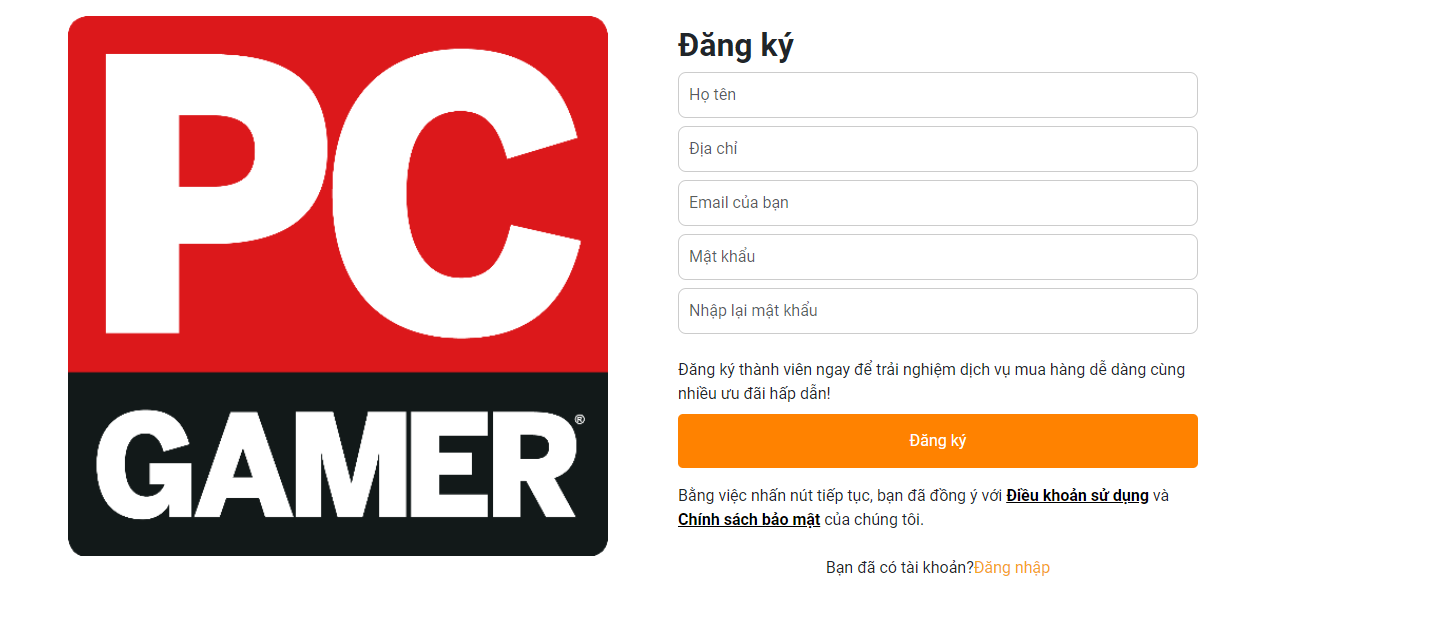
### Chức năng 1: Chức năng đăng nhập, đăng ký

Đây là trang đăng nhập cho người dùng, có hai chức năng chính là đăng nhập và đăng ký, khi đăng nhập cần phải nhập email và mật khẩu để tiến hành vào hệ thống

Trường hợp không có tài khoản người dùng có thể bấm vào nút đăng ký để đăng ký tài khoản mới, người dùng cần phải điền tên, địa chỉ, email, mật khẩu và xác nhận mật khẩu để đăng ký



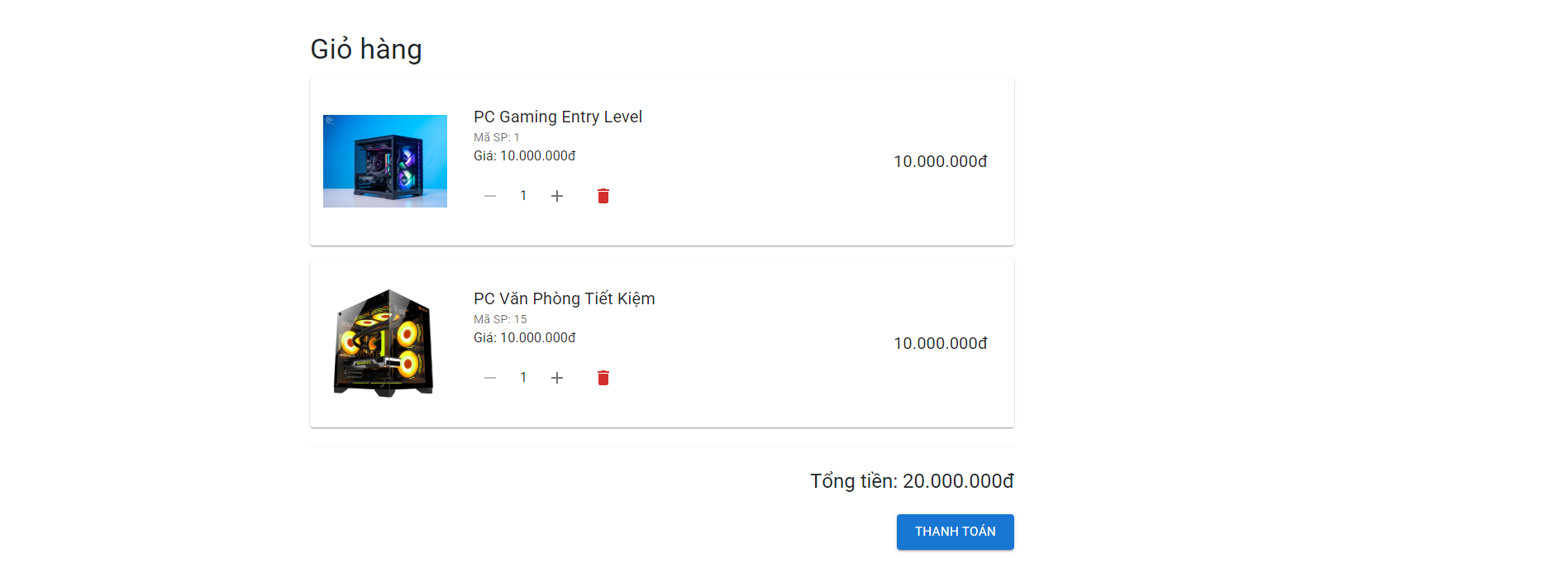
Hình 3. 14 Giao diện đăng nhập



Hình 3. 15 Giao diện đăng ký

### Chức năng 2: Chức năng giỏ hàng

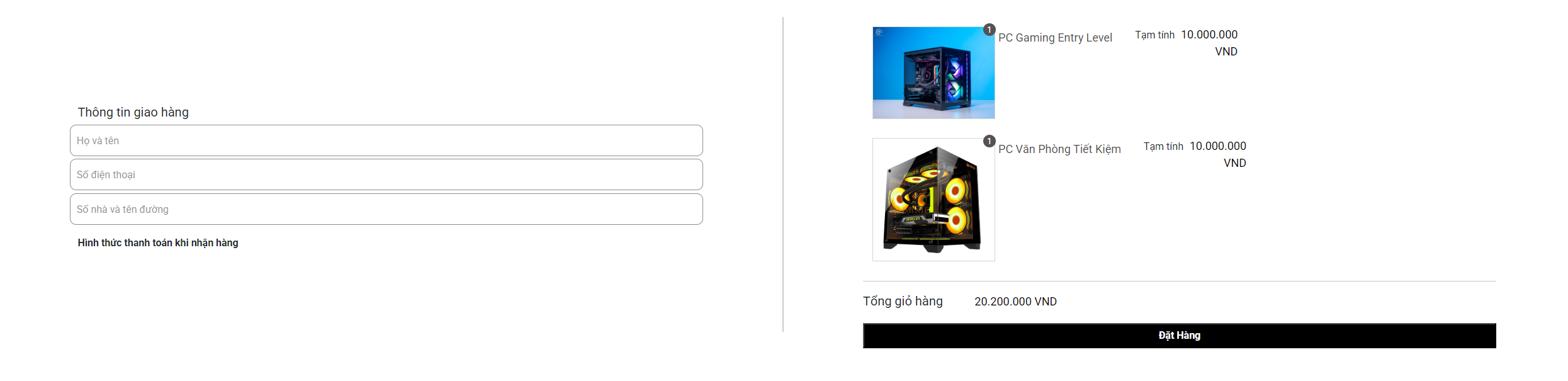
Chức năng giỏ hàng cho phép bỏ các sản phẩm mong muốn vào giỏ hàng, bên trong giỏ hàng có thể xem giá tiền của từng sản phẩm và tổng tiền của tất cả sản phẩm trong giỏ hàng, có thể xác nhận mua tất cả bằng cách bấm thanh toán.



Hình 3. 16 Giao diện giỏ hàng

### Chức năng 3: Chức năng thanh toán

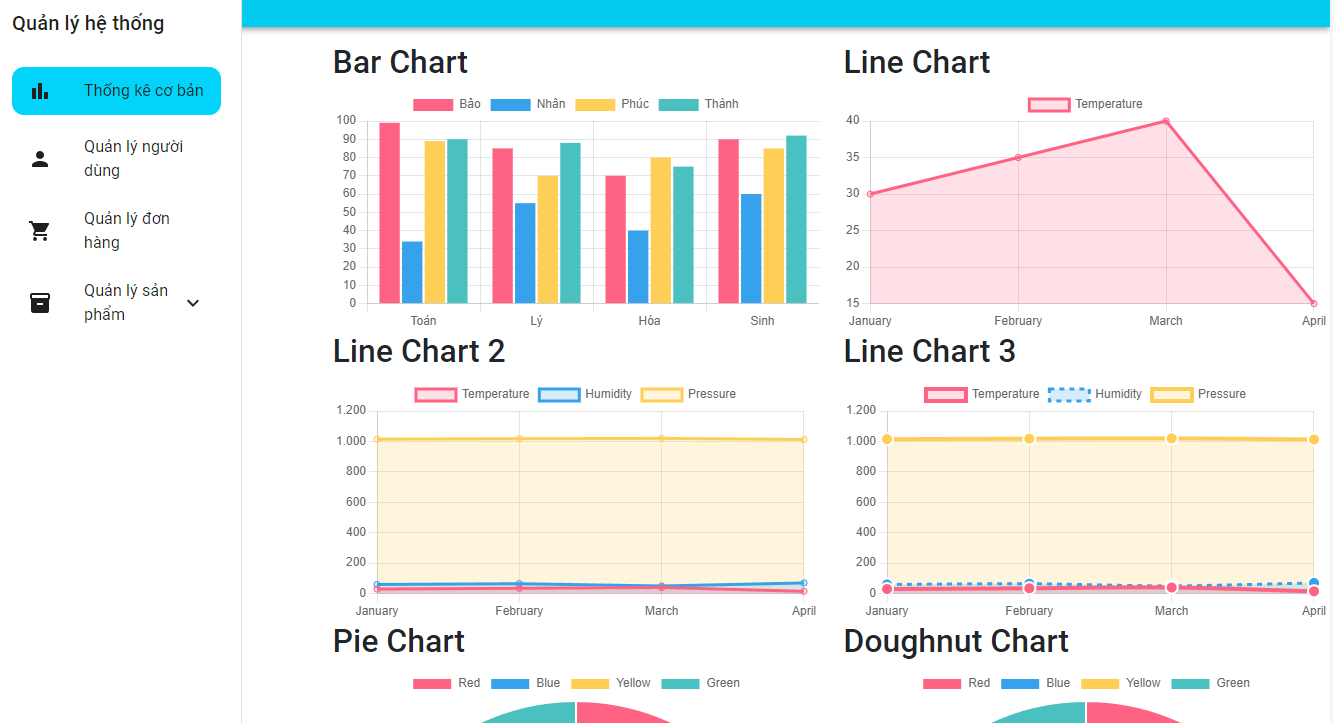
Chức năng thanh toán yêu cầu nhập thông tin cá nhân để giao hàng, bao gồm họ tên, số điện thoại, địa chỉ để có thể đặt hàng, khi thông tin đã nhập xong chỉ cần bấm đặt hàng là có thể đặt được.



Hình 3. 17 Giao diện thanh toán

### Chức năng 4: Chức năng thống kê

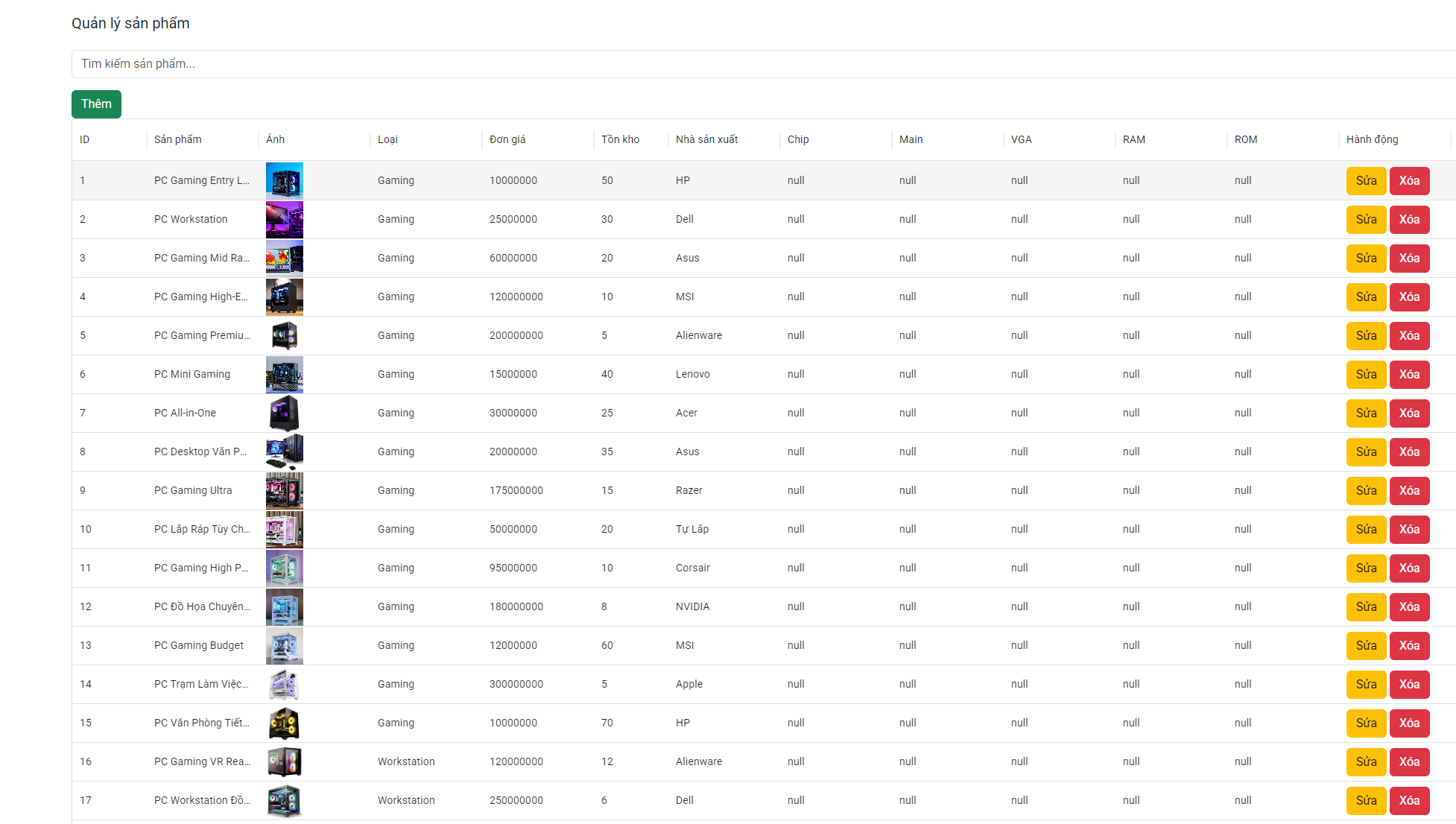
Chức năng thống kê tại trang quản trị cho phép xem thông tin, thống kê cơ bản về số lượng sản phẩm bán được trong ngày hay trong tháng, hoặc thậm chí là trong năm, có thể thống kê theo nhiều tiêu chí.



Hình 3. 18 Giao diện thống kê

### Chức năng 5: Chức năng quản lý sản phẩm

Chức năng quản lý sản phẩm của quản trị viên có thể xem được số lượng sản phẩm còn hiện tại, thông tin chi tiết của sản phẩm, có thể thêm một sản phẩm mới, hoặc cập nhật sản phẩm hiện tại, hoặc là xóa sản phẩm hiện tại.



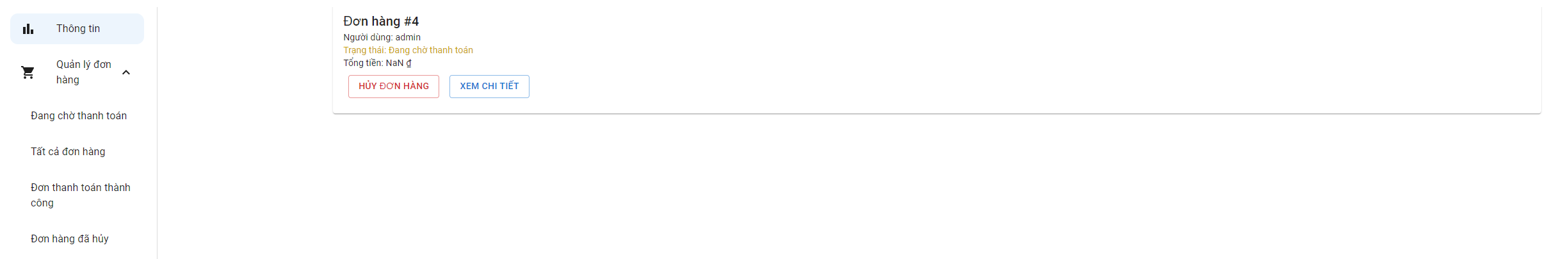
Hình 3. 19 Giao diện quản lý sản phẩm

### Chức năng 6: Chức năng quản lý đơn hàng

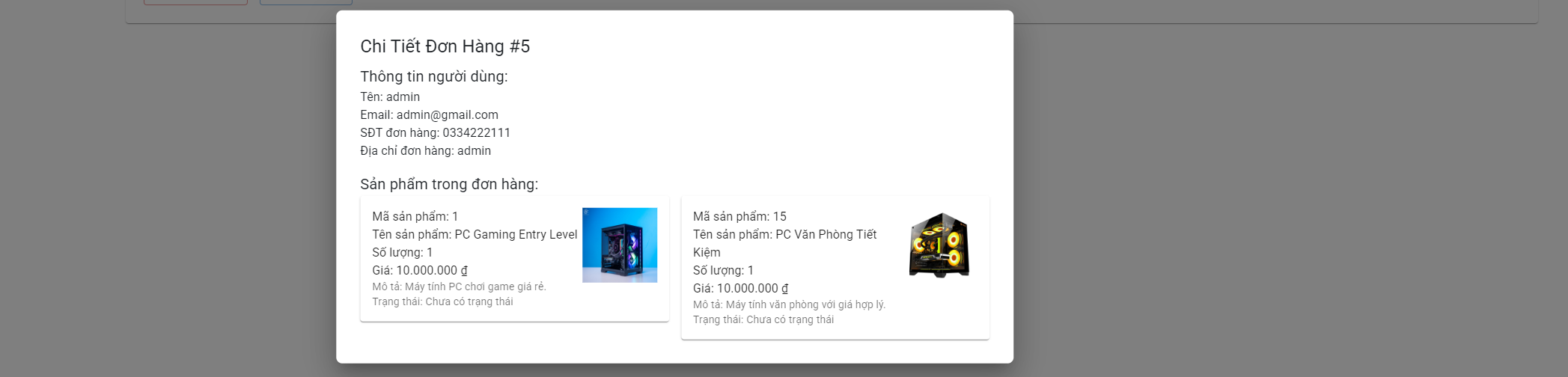
Quản lý đơn hàng cho phép quản lý đơn hàng của cá nhân, quản trị viên.

- Quản trị viên được quyền xác nhận đơn hàng cho người dùng.

- Người dùng có thể xem xem đơn hàng của mình đã được duyệt hay chưa, hoặc xóa cả cái đơn hàng đó đi không đặt nữa.



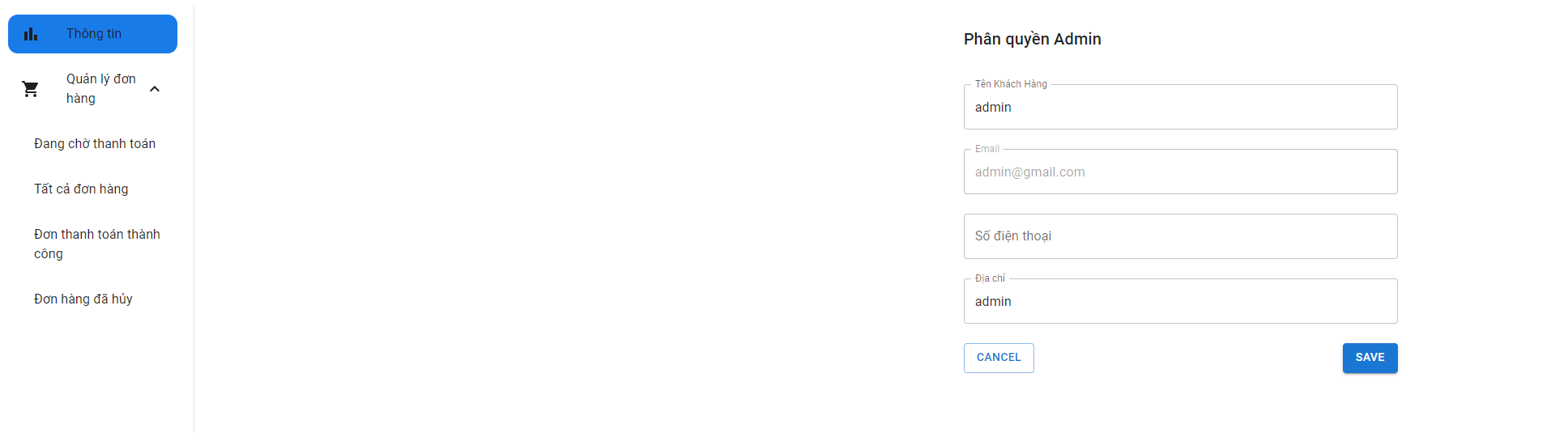
Hình 3. 20 Giao diện quản lý đơn hàng



Hình 3. 21 Giao diện chi tiết đơn hàng

### Chức năng 7: Chức năng quản lý thông tin cá nhân

Chức năng quản lý thông tin cá nhân của người dùng, người dùng có thể cập nhật thông tin cá nhân của mình, có thể xác nhận đổi lại thông tin cá nhân như họ tên, số điện thoại, địa chỉ.



Hình 3. 22 Quản lý thông tin cá nhân

# **Phần 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

1. KẾT QUẢ ĐẠT ĐƯỢC

Sau quá trình phát triển và triển khai, website bán PC đã hoàn thành và đạt được các mục tiêu chính đề ra. Các kết quả cụ thể bao gồm:

* Hoàn thành giao diện người dùng: Website có giao diện thân thiện, dễ sử dụng, giúp người dùng dễ dàng tìm kiếm và lựa chọn sản phẩm.
* Tính năng tìm kiếm và lọc sản phẩm: Người dùng có thể dễ dàng tìm kiếm các sản phẩm PC và linh kiện với các tiêu chí như giá, cấu hình, thương hiệu, và các tính năng đặc biệt.
* Hệ thống quản lý người dùng: Tính năng đăng ký, đăng nhập và theo dõi đơn hàng giúp khách hàng quản lý thông tin cá nhân và các đơn hàng đã thực hiện.
* Quản lý sản phẩm: Website cung cấp hệ thống quản lý sản phẩm, cho phép thêm, sửa, và xóa sản phẩm dễ dàng từ phía người quản trị.

Với các tính năng trên, website đã đáp ứng được yêu cầu cơ bản về một nền tảng thương mại điện tử bán PC, tạo điều kiện thuận lợi cho khách hàng trong việc mua sắm trực tuyến.

1. ƯU NHƯỢC ĐIỂM

#### **Ưu điểm:**

* Giao diện người dùng dễ sử dụng: Website có thiết kế đơn giản, dễ sử dụng, thân thiện với người dùng, kể cả đối với những người không quen với công nghệ.
* Khả năng mở rộng: Cấu trúc website cho phép dễ dàng mở rộng các tính năng mới, như tích hợp thêm phương thức thanh toán, hỗ trợ khách hàng trực tuyến, hoặc mở rộng danh mục sản phẩm.
* Tính năng tìm kiếm mạnh mẽ: Với các bộ lọc tìm kiếm chi tiết, người dùng có thể dễ dàng tìm thấy sản phẩm phù hợp với nhu cầu của mình.
* Bảo mật thông tin: Các biện pháp bảo mật được áp dụng để bảo vệ thông tin cá nhân và giao dịch của khách hàng.

#### **Nhược điểm:**

* Chưa tối ưu hóa cho di động: Website hiện tại chưa được tối ưu hóa trên các thiết bị di động, điều này có thể gây khó khăn cho người dùng khi truy cập từ các thiết bị nhỏ.
* Thiếu hỗ trợ khách hàng trực tuyến: Mặc dù website có các phương thức thanh toán trực tuyến, nhưng việc thiếu tính năng hỗ trợ khách hàng trực tuyến (chat trực tiếp) có thể làm giảm trải nghiệm người dùng.
* Chưa có thanh toán trực tuyến: Chưa thể tích hợp các phương thức trực tuyến như Momo, Paypal vào hệ thống dẫn đến một số hạn chế và bất cập cho người dùng.

1. HƯỚNG PHÁT TRIỂN

Để nâng cao trải nghiệm người dùng và mở rộng khả năng kinh doanh, website có thể phát triển thêm các tính năng sau:

* Tối ưu hóa di động: Cải thiện giao diện và tính năng của website trên các thiết bị di động để đảm bảo rằng người dùng có thể dễ dàng duyệt web và mua sắm từ bất kỳ đâu.
* Hỗ trợ khách hàng trực tuyến: Tích hợp các công cụ hỗ trợ khách hàng trực tuyến như chatbot hoặc hệ thống live chat để giải đáp nhanh chóng các thắc mắc của khách hàng.
* Mở rộng phương thức thanh toán và giao hàng: Hỗ trợ thêm các phương thức thanh toán và giao hàng quốc tế hoặc các phương thức giao hàng linh hoạt hơn, đáp ứng nhu cầu của khách hàng.
* Phát triển tính năng đánh giá và nhận xét: Mở rộng hệ thống đánh giá sản phẩm, cho phép khách hàng để lại nhận xét chi tiết về sản phẩm, giúp những người mua sau có thêm thông tin tham khảo.
* Tích hợp AI và MachineLearning: Áp dụng công nghệ trí tuệ nhân tạo và học máy để cải thiện khả năng gợi ý sản phẩm, phân tích hành vi người dùng và tối ưu hóa trải nghiệm mua sắm.

# **TÀI LIỆU THAM KHẢO**

|  |  |
| --- | --- |
| [1] | Nguyễn Khắc Quốc, tài liệu giảng dạy môn “Công nghệ phần mềm”, 2015, trường Đại học Trà Vinh. |
| [2] | Phạm Minh Đương, tài liệu giảng dạy môn “Phân tích và thiết kế hệ thống, trường Đại học Trà Vinh, 2014. |
| [3] | Phan Thị Phương Nam, tài liệu giảng dạy môn “Hệ quản trị Cơ sở dữ liệu”, trường Đại học Trà Vinh, 2015. |
| [4] | R. Team, "React – Learn React," 17 December 2024. [Online]. Available: https://reactjs.org/docs/getting-started.html. |
| [5] | L. D. Team, "Laravel API Documentation," 17 December 2024. [Online]. Available: https://laravel.com/api/. |
| [6] | R. Team, "React Hooks," 17 December 2024. [Online]. Available: https://reactjs.org/docs/hooks-intro.html. |
| [7] | L. D. Team, "Laravel Eloquent ORM," 17 December 2024. [Online]. Available: https://laravel.com/docs/8.x/eloquent. |