**BỘ CÔNG THƯƠNG**

**TRƯỜNG CAO ĐẲNG CÔNG THƯƠNG TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

**-----🙣🙢🕮🙠🙡-----**



**ĐỒ ÁN TỐT NGHIỆP**

**XÂY DỰNG WEB BÁN HÀNG QUA MẠNG**

GVHD: ThS. Trần Anh Khoa

Sinh Viên: Nguyễn Tiến Phương

MSSV: 2120110158

Lớp: CQQ2011E

**TP.HCM, ngày 17 tháng 6 năm 2023**

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN

LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành và sự tri ân sâu sắc với quý thầy cô Trường Cao đẳng Công thương TP.HCM, đặc biệt là quý thầy cô bộ môn khoa Công nghệ thông tin đã giúp em có những kiến thức cần thiết và bổ ích để em có thể hoàn thành được bài đồ án chuyên ngành. Em xin chân thành cảm ơn thầy Trần Anh Khoa đã nhiệt tình hướng dẫn, chỉ bảo và góp ý để em hoàn thành đồ án tốt nghiệp.

Trong quá trình làm đồ án chắc chắn em còn mắc phải nhiều sai sót rất mong thầy bỏ qua. Đồng thời, với kinh nghiệm và kiến thức còn hạn chế em xin sự đóng góp từ thầy để đồ án của em được hoàn thiện hơn.

Em xin chân thành cảm ơn.

Tp.HCM, ngày 17 tháng 6 năm 2023

Nguyễn Tiến Phương

LỜI MỞ ĐẦU

Trang web bán hàng qua mạng đã trở thành một phương tiện tiện lợi và thông dụng để mua sắm. Việc xây dựng một trang web bán hàng không chỉ đòi hỏi kiến thức kỹ thuật mà còn yêu cầu sự hiểu biết về thị trường và khách hàng. Vì vậy, tôi đã tìm hiểu và nghiên cứu về công nghệ, thiết kế, và xây dựng một trang web bán hàng nhằm phát triển sản phẩm tốt hơn, đáp ứng được nhu cầu của khách hàng.

Báo cáo đồ án tốt nghiệp có nội dung bao gồm 4 chương như sau:

- Chương I: Tổng quan về đề tài.

- Chương II: Cơ sở lý thuyết.

- Chương III: Phân tích thiết kế hệ thống.

- Chương IV: Triển khai cài đặt và đánh giá.

1. TỔNG QUAN VỀ ĐỀ TÀI
   * 1. Lý do chọn đề tài

Tôi đã chọn đề tài "Xây dựng trang web bán hàng qua mạng" vì đây là một chủ đề rất phù hợp với thời đại công nghệ số hiện nay. Trang web bán hàng qua mạng đã trở thành một phương tiện tiện lợi và thông dụng để mua sắm. Vì vậy, việc xây dựng một trang web bán hàng hiệu quả không chỉ đòi hỏi kiến thức kỹ thuật mà còn yêu cầu sự hiểu biết về thị trường và khách hàng.

Bên cạnh đó, với vai trò của một lập trình viên, tôi luôn muốn tìm hiểu và áp dụng những công nghệ mới nhất để xây dựng những sản phẩm tốt hơn, đáp ứng được nhu cầu của khách hàng. Việc nghiên cứu và thực hiện đề tài này giúp tôi phát triển các kỹ năng lập trình và cũng mở rộng kiến thức của tôi về ngành thương mại điện tử.

* + 1. Mục đích đề tài

- Ứng dụng sẽ giúp quản lý được tất cả hoạt động mua, bán và cũng quản lý được nhân viên, hàng hóa, kho và hóa đơn một cách tốt nhất có thể.

- Quản lý nhân viên dễ dàng và thuận tiện cho việc phân chia công việc.

- Quản lý hàng hóa giúp người dùng có thể biết được doanh thu của các loại hàng hóa.

- Quản lý kho dễ dàng kiểm tra, thống kê cũng như báo cáo về số lượng hàng trong kho và có thể quản lý được việc nhập, xuất hàng hóa tốt hơn.

- Tiết kiệm được nhiều thời gian và đem lại sự thuận tiện và hiệu quả cho người dùng.

* + 1. Đối tượng và phạm vi đề tài

Đối tượng: phần mềm chủ yếu sẽ dành cho các trang web thương mại điện tử.

Phạm vi không gian: đề tài sẽ chủ yếu nghiên cứu về việc quản lý các hoạt động của trang web.

Phạm vi thời gian: việc thực hiện và nghiên cứu đề tài sẽ được thực hiện từ ngày 14 tháng 3 năm 2023 đến ngày 17 tháng 6 năm 2023.

* + 1. Yêu cầu
       1. Yêu cầu chức năng

Yêu cầu chức năng của ứng dụng là:

* Quản lý khách hàng
* Quản lý danh mục
* Quản lý nhà cung cấp
* Quản lý nhân viên
* Quản lý sản phẩm
* Quản lý hóa đơn

Để đảm bảo được sự an toàn và bảo mật thông tin thì tất cả các chức năng khi muốn thực hiện đều phải đăng nhập bằng các tài khoản đã được tạo.

Trong các tài khoản được đăng nhập vào hệ thống và sử dụng các chức năng sẽ có một tài khoản sử dụng được tất cả chức năng, tài khoản đó sẽ tạo ra các tài khoản khác và có thể phân quyền cho các tài khoản được tạo ra.

* + - 1. Yêu cầu phi chức năng

Bên cạnh các yêu cầu về chức năng của ứng dụng thì cũng có những yêu cầu cầu phi chức năng như sau:

* Báo cáo
* Thống kê

1. CƠ SỞ LÝ THUYẾT
   * 1. HTML

HTML (viết tắt của từ HyperText Markup Language, hay là "Ngôn ngữ Đánh dấu Siêu văn bản") là một ngôn ngữ đánh dấu được thiết kế ra để tạo nên các trang web trên World Wide Web. Nó có thể được trợ giúp bởi các công nghệ như CSS và các ngôn ngữ kịch bản giống như JavaScript.

Các trình duyệt web nhận tài liệu HTML từ một web server hoặc một kho lưu trữ cục bộ và render tài liệu đó thành các trang web đa phương tiện. HTML mô tả cấu trúc của một trang web về mặt ngữ nghĩa và các dấu hiệu ban đầu được bao gồm cho sự xuất hiện của tài liệu.

Các phần tử HTML là các khối xây dựng của các trang HTML. Với cấu trúc HTML, hình ảnh và các đối tượng khác như biểu mẫu tương tác có thể được nhúng vào trang được hiển thị. HTML cung cấp một phương tiện để tạo tài liệu có cấu trúc bằng cách biểu thị ngữ nghĩa cấu trúc cho văn bản như headings, paragraphs, lists, links, quotes và các mục khác. Các phần tử HTML được phân định bằng các thẻ, được viết bằng dấu ngoặc nhọn. Các thẻ như <img /> và <input /> giới thiệu trực tiếp nội dung vào trang. Các thẻ khác như <p> bao quanh và cung cấp thông tin về văn bản tài liệu và có thể bao gồm các thẻ khác làm phần tử phụ. Các trình duyệt không hiển thị các thẻ HTML, nhưng sử dụng chúng để diễn giải nội dung của trang.

HTML có thể nhúng các chương trình được viết bằng scripting như JavaScript, điều này ảnh hưởng đến hành vi và nội dung của các trang web. Việc bao gồm CSS xác định giao diện và bố cục của nội dung. World Wide Web Consortium (W3C), trước đây là đơn vị bảo trì HTML và là người duy trì hiện tại của các tiêu chuẩn CSS, đã khuyến khích việc sử dụng CSS trên HTML trình bày rõ ràng kể từ năm 1997.

* + 1. CSS

Trong tin học, các tập tin định kiểu theo tầng – dịch từ tiếng Anh là Cascading Style Sheets (CSS) – được dùng để miêu tả cách trình bày các tài liệu viết bằng ngôn ngữ HTML và XHTML. Ngoài ra ngôn ngữ định kiểu theo tầng cũng có thể dùng cho XML, SVG, XUL. Các đặc điểm kỹ thuật của CSS được duy trì bởi World Wide Web Consortium (W3C). Thay vì đặt các thẻ quy định kiểu dáng cho văn bản HTML (hoặc XHTML) ngay trong nội dung của nó, bạn nên sử dụng CSS.

**Tác dụng của CSS**

Hạn chế tối thiểu việc làm rối mã HTML của trang Web bằng các thẻ quy định kiểu dáng (chữ đậm, chữ in nghiêng, chữ có gạch chân, chữ màu), khiến mã nguồn của trang Web được gọn gàng hơn, tách nội dung của trang Web và định dạng hiển thị, dễ dàng cho việc cập nhật nội dung.

Tạo ra các kiểu dáng có thể áp dụng cho nhiều trang Web, giúp tránh phải lặp lại việc định dạng cho các trang Web giống nhau.

Có 3 cách để sử dụng CSS.

* "Inline CSS": Áp dụng trực tiếp trên một đối tượng nhất định bằng thuộc tính style: <span style=”font-weight: bold;”>Đoạn văn in đậm</span>
* "Internal CSS": Đặt CSS ở đầu trang Web để áp dụng kiểu dáng cho toàn bộ trang ấy, khi đó chỉ cần đặt đoạn CSS vào trong cặp thẻ <style> rồi đặt vào trong phần header của Web (giữa <head> và </head>):
* "External CSS": Đặt các thuộc tính CSS vào một tệp tin riêng biệt (\*.css), khi đó có thể tham chiếu đến từ nhiều trang Web khác nhau: Tham chiếu tới tệp tin CSS trên từ trang Web bằng đoạn mã (mã có thể nằm ngoài thẻ <head>):<link rel=”stylesheet” type=”text/css” href=”style.css”/>
  + 1. JavaScript

JavaScript là ngôn ngữ lập trình được nhà phát triển sử dụng để tạo trang web tương tác. Từ làm mới bảng tin trên trang mạng xã hội đến hiển thị hình ảnh động và bản đồ tương tác, các chức năng của JavaScript có thể cải thiện trải nghiệm người dùng của trang web. Là ngôn ngữ kịch bản phía máy khách, JavaScript là một trong những công nghệ cốt lõi của World Wide Web. Ví dụ: khi duyệt internet, bất cứ khi nào bạn thấy quảng cáo quay vòng dạng hình ảnh, menu thả xuống nhấp để hiển thị hoặc màu sắc phần tử thay đổi động trên trang web cũng chính là lúc bạn thấy các hiệu ứng của JavaScript.

Trước đây, các trang web có dạng tĩnh, tương tự như các trang trong một cuốn sách. Một trang tĩnh chủ yếu hiển thị thông tin theo một bố cục cố định và không làm được mọi thứ mà chúng ta mong đợi như ở một trang web hiện đại. JavaScript dần được biết đến như một công nghệ phía trình duyệt để làm cho các ứng dụng web linh hoạt hơn. Sử dụng JavaScript, các trình duyệt có thể phản hồi tương tác của người dùng và thay đổi bố cục của nội dung trên trang web.

Khi ngôn ngữ này phát triển hoàn thiện, các nhà phát triển JavaScript đã thiết lập các thư viện, khung và cách thức lập trình cũng như bắt đầu sử dụng ngôn ngữ này bên ngoài trình duyệt web. Ngày nay, bạn có thể sử dụng JavaScript để thực hiện hoạt động phát triển cả ở phía máy khách và máy chủ. Dưới đây là một số trường hợp sử dụng phổ biến:

Trước đây, các trang web có dạng tĩnh, tương tự như các trang trong một cuốn sách. Một trang tĩnh chủ yếu hiển thị thông tin theo một bố cục cố định và không làm được mọi thứ mà chúng ta mong đợi như ở một trang web hiện đại. JavaScript dần được biết đến như một công nghệ phía trình duyệt để làm cho các ứng dụng web linh hoạt hơn. Sử dụng JavaScript, các trình duyệt có thể phản hồi tương tác của người dùng và thay đổi bố cục của nội dung trên trang web.

Khi ngôn ngữ này phát triển hoàn thiện, các nhà phát triển JavaScript đã thiết lập các thư viện, khung và cách thức lập trình cũng như bắt đầu sử dụng ngôn ngữ này bên ngoài trình duyệt web. Ngày nay, bạn có thể sử dụng JavaScript để thực hiện hoạt động phát triển cả ở phía máy khách và máy chủ. Dưới đây là một số trường hợp sử dụng phổ biến.

* + 1. SQL

Ngôn ngữ truy vấn có cấu trúc (SQL) là một ngôn ngữ lập trình phục vụ việc lưu trữ và xử lý thông tin trong cơ sở dữ liệu quan hệ. Cơ sở dữ liệu quan hệ lưu trữ thông tin dưới dạng bảng có các hàng và cột đại diện cho những thuộc tính dữ liệu và nhiều mối quan hệ khác nhau giữa các giá trị dữ liệu. Bạn có thể sử dụng các câu lệnh SQL để lưu trữ, cập nhật, loại bỏ, tìm kiếm và truy xuất thông tin từ cơ sở dữ liệu. Bạn cũng có thể sử dụng SQL để duy trì và tối ưu hóa hiệu suất cơ sở dữ liệu.

Ngôn ngữ truy vấn có cấu trúc (SQL) là một ngôn ngữ truy vấn phổ biến thường được sử dụng trong tất cả các loại ứng dụng. Các nhà phân tích và phát triển dữ liệu tìm hiểu và sử dụng SQL do ngôn ngữ này tích hợp hiệu quả với nhiều ngôn ngữ lập trình khác nhau. SQL cũng khá dễ tìm hiểu do nó sử dụng những từ khóa tiếng Anh phổ biến trong các câu lệnh.

SQL ra đời vào những năm 1970 dựa trên mô hình dữ liệu quan hệ. Ban đầu nó được gọi là ngôn ngữ truy vấn tiếng Anh có cấu trúc (SEQUEL). Thuật ngữ này sau đó được rút ngắn thành SQL. Oracle, trước đây được gọi là Relational Software, đã trở thành nhà cung cấp đầu tiên mang đến một hệ thống quản lý cơ sở dữ liệu quan hệ SQL thương mại.

* + 1. ReactJS

ReactJS là một opensource được phát triển bởi Facebook, ra mắt vào năm 2013, bản thân nó là một thư viện Javascript được dùng để để xây dựng các tương tác với các thành phần trên website. Một trong những điểm nổi bật nhất của ReactJS đó là việc render dữ liệu không chỉ thực hiện được trên tầng Server mà còn ở dưới Client nữa.

ReactJS là một thư viện JavaScript chuyên giúp các nhà phát triển xây dựng giao diện người dùng hay UI. Trong lập trình ứng dụng front-end, lập trình viên thường sẽ phải làm việc chính trên 2 thành phần sau: UI và xử lý tương tác của người dùng. UI là tập hợp những thành phần mà bạn nhìn thấy được trên bất kỳ một ứng dụng nào, ví dụ có thể kể đến bao gồm: menu, thanh tìm kiếm, những nút nhấn, card,… Giả sử bạn đang lập trình một website thương mại điện tử, sau khi người dùng chọn được sản phẩm ưng ý rồi và nhấn vào nút “Thêm vào giỏ hàng”, thì việc tiếp theo mà bạn phải làm đó là thêm sản phẩm được chọn vào giỏ hàng và hiển thị lại sản phẩm đó khi user vào xem.

Trước khi có ReactJS, lập trình viên thường gặp rất nhiều khó khăn trong việc sử dụng “vanilla JavaScript”(JavaScript thuần) và Jquery để xây dựng UI. Điều đó đồng nghĩa với việc quá trình phát triển ứng dụng sẽ lâu hơn và xuất hiện nhiều bug, rủi ro hơn. Vì vậy vào năm 2011, Jordan Walke – một nhân viên của Facebook đã khởi tạo ReactJS với mục đích chính là cải thiện quá trình phát triển UI.

* + 1. NodeJS

NodeJS là một môi trường runtime chạy JavaScript đa nền tảng và có mã nguồn mở, được sử dụng để chạy các ứng dụng web bên ngoài trình duyệt của client. Nền tảng này được phát triển bởi Ryan Dahl vào năm 2009, được xem là một giải pháp hoàn hảo cho các ứng dụng sử dụng nhiều dữ liệu nhờ vào mô hình hướng sự kiện (event-driven) không đồng bộ.

Ban đầu, tác giả đặt tên dự án là web.js vì mục đích đơn thuần chỉ là sử dụng như một ứng dụng web thay cho Apache hay các server khác. Tuy nhiên, dự án nhanh chóng phát triển rất mạnh, vượt qua những khả năng của một webserver thông thường và thậm chí có thể được xem như một nền tảng đa mục đích. Vì vậy, sau này dự án đã được đổi tên thành Node.JS và sử dụng đến ngày nay.

**Ưu điểm**

* IO hướng sự kiện không đồng bộ, cho phép xử lý nhiều yêu cầu đồng thời.
* Sử dụng JavaScript – một ngôn ngữ lập trình dễ học.
* Chia sẻ cùng code ở cả phía client và server.
* NPM(Node Package Manager) và module Node đang ngày càng phát triển mạnh mẽ.
* Cộng đồng hỗ trợ tích cực.
* Cho phép stream các file có kích thước lớn.

**Nhược điểm**

* Không có khả năng mở rộng, vì vậy không thể tận dụng lợi thế mô hình đa lõi trong các phần cứng cấp server hiện nay.
* Khó thao tác với cơ sử dữ liệu quan hệ.
* Mỗi callback sẽ đi kèm với rất nhiều callback lồng nhau khác.
* Cần có kiến thức tốt về JavaScript.
* Không phù hợp với các tác vụ đòi hỏi nhiều CPU.

1. PHÂN TÍCH THIẾT KẾ HỆ THỐNG
   * 1. Tác nhân chính của hệ thống

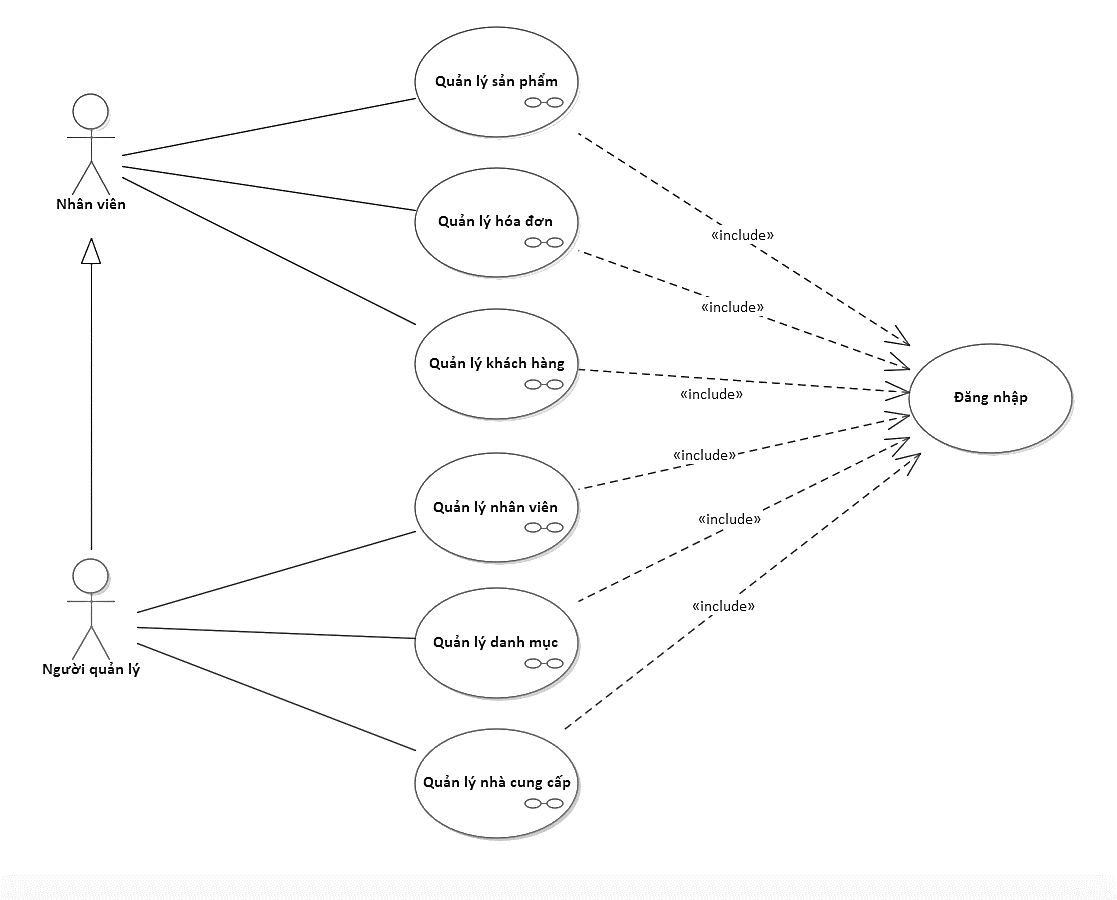
**Actor:** Người quản lý và nhân viên.

**Use Case:**

Người quản lý: đăng nhập, quản lý khách hàng, quản lý danh mục, quản lý nhà cung cấp, quản lý nhân viên, quản lý sản phẩm, quản lý hóa đơn.

Nhân viên: đăng nhập, quản lý khách hàng, quản lý sản phẩm, quản lý hóa đơn.

* + 1. Biểu đồ Use Case
       1. Biểu đồ Use Case tổng quát



* + - 1. Đặc tả Use Case

Đăng nhập

|  |  |
| --- | --- |
| **Use case** | **Nội Dung** |
| Tên use-case | Đăng Nhập. |
| Mô tả | Use case này cho phép người dùng đăng nhập vào hệ thống hệ thống. |
| Actors | Người quản lý, Nhân viên. |
| Điều kiện kích hoạt | Khi người dùng chọn Đăng nhập tại trang chủ. |
| Luồng cơ bản | 1. Tại trang chủ, người dùng chọn “Đăng nhập”.  2. Hệ thống hiển thị form đăng nhập.  3. Người dùng nhập các thông tin bao gồm: email, password.  4. Người dùng nhấn “Đăng nhập”.  5. Hệ thống kiểm tra thông tin đăng nhập (email, password).  A1: Email, password không hợp lệ.  6. Đăng nhập thành công, quay trang chủ. |
| Các luồng thay thế | A1: Email, password không hợp lệ:  1. Hệ thống hiển thị lại form đăng nhập kèm với thông báo email hoặc mật khẩu không hợp lệ.  2. Quay lại bước 2 trong luồng chính. |
| Pre-Condition(s):  Điều kiện đầu vào  (tiền điều kiện) | Người dùng đã có tài khoản trên hệ thống. |
| Post-Condition(s):  Điều kiện đầu ra  (hậu điều kiện) | Người dùng đăng nhập thành công. |

Quản lý khách hàng

|  |  |
| --- | --- |
| **Use case** | **Nội Dung** |
| Tên use-case | Quản lý khách hàng |
| Mô tả | Use case này cho phép người quản lý quản lý khách hàng của mình. |
| Actors | Người quản lý, Nhân viên. |
| Điều kiện kích hoạt | Khi người dùng chọn Quản lý khách hàng |
| Luồng cơ bản | 1. Tại trang chủ, người dùng chọn “Đăng nhập”.  2. Hệ thống hiển thị form đăng nhập.  3. Người dùng nhập các thông tin bao gồm: email, password.  4. Người dùng nhấn “Đăng nhập”.  5. Hệ thống kiểm tra thông tin đăng nhập (email, password).  A1: Email, password không hợp lệ.  6. Đăng nhập thành công, quay trang chủ.  7. Người dùng chọn Quản lý khách hàng.  8. Hệ thống hiển thị danh sách khách hàng và hiển thị chức năng.  9. Người dùng thực hiện chức năng Quản lý khách hàng.  10. Người dùng thực hiện thành công chức năng. |
| Các luồng thay thế | A1: Email, password không hợp lệ:  1. Hệ thống hiển thị lại form đăng nhập kèm với thông báo email hoặc mật khẩu không hợp lệ.  2. Quay lại bước 2 trong luồng chính. |
| **Pre-Condition(s):**  Điều kiện đầu vào  (tiền điều kiện) | Người dùng đã có tài khoản trên hệ thống. |
| **Post-Condition(s):**  Điều kiện đầu ra  (hậu điều kiện) | Người dùng đăng nhập thành công. |

Quản lý danh mục

|  |  |
| --- | --- |
| **Use case** | **Nội Dung** |
| Tên use-case | Quản lý danh mục. |
| Mô tả | Use case này cho phép người dùng Quản lý danh mục. |
| Actors | Người quản lý. |
| Điều kiện kích hoạt | Khi người dùng chọn Quản lý danh mục. |
| Luồng cơ bản | 1. Tại trang chủ, người dùng chọn “Đăng nhập”.  2. Hệ thống hiển thị form đăng nhập.  3. Người dùng nhập các thông tin bao gồm: email, password.  4. Người dùng nhấn “Đăng nhập”.  5. Hệ thống kiểm tra thông tin đăng nhập (email, password).  A1: Email, password không hợp lệ.  6. Đăng nhập thành công, quay trang chủ.  7. Người dùng chọn Quản lý danh mục.  8. Hệ thống hiển thị danh sách danh mục và hiển thị chức năng.  9. Người dùng thực hiện chức năng Quản lý danh mục.  10. Người dùng thực hiện thành công chức năng. |
| Các luồng thay thế | A1: Email, password không hợp lệ:  1. Hệ thống hiển thị lại form đăng nhập kèm với thông báo email hoặc mật khẩu không hợp lệ.  2. Quay lại bước 2 trong luồng chính. |
| **Pre-Condition(s):**  Điều kiện đầu vào  (tiền điều kiện) | Người dùng đã có tài khoản trên hệ thống. |
| **Post-Condition(s):**  Điều kiện đầu ra  (hậu điều kiện) | Người dùng đăng nhập thành công. |

Quản lý nhà cung cấp

|  |  |
| --- | --- |
| **Use case** | **Nội Dung** |
| Tên use-case | Quản lý nhà cung cấp. |
| Mô tả | Use case này cho phép người quản lý quản lý nhà cung cấp của mình. |
| Actors | Người quản lý. |
| Điều kiện kích hoạt | Khi người dùng chọn Quản lý nhà cung cấp. |
| Luồng cơ bản | 1. Tại trang chủ, người dùng chọn “Đăng nhập”.  2. Hệ thống hiển thị form đăng nhập.  3. Người dùng nhập các thông tin bao gồm: email, password.  4. Người dùng nhấn “Đăng nhập”.  5. Hệ thống kiểm tra thông tin đăng nhập (email, password).  A1: Email, password không hợp lệ.  6. Đăng nhập thành công, quay trang chủ.  7. Người dùng chọn Quản lý nhà cung cấp.  8. Hệ thống hiển thị danh sách nhà cung cấp và hiển thị chức năng.  9. Người dùng thực hiện chức năng Quản lý nhà cung cấp.  10. Người dùng thực hiện thành công chức năng. |
| Các luồng thay thế | A1: Email, password không hợp lệ:  1. Hệ thống hiển thị lại form đăng nhập kèm với thông báo email hoặc mật khẩu không hợp lệ.  2. Quay lại bước 2 trong luồng chính. |
| **Pre-Condition(s):**  Điều kiện đầu vào  (tiền điều kiện) | Người dùng đã có tài khoản trên hệ thống. |
| **Post-Condition(s):**  Điều kiện đầu ra  (hậu điều kiện) | Người dùng đăng nhập thành công. |

Quản lý nhân viên

|  |  |
| --- | --- |
| **Use case** | **Nội Dung** |
| Tên use-case | Quản lý nhân viên. |
| Mô tả | Use case này cho phép người quản lý quản lý nhân viên của mình. |
| Actors | Người quản lý. |
| Điều kiện kích hoạt | Khi người dùng chọn Quản lý nhân viên. |
| Luồng cơ bản | 1. Tại trang chủ, người dùng chọn “Đăng nhập”.  2. Hệ thống hiển thị form đăng nhập.  3. Người dùng nhập các thông tin bao gồm: email, password.  4. Người dùng nhấn “Đăng nhập”.  5. Hệ thống kiểm tra thông tin đăng nhập (email, password).  A1: Email, password không hợp lệ.  6. Đăng nhập thành công, quay trang chủ.  7. Người dùng chọn Quản lý nhân viên.  8. Hệ thống hiển thị danh sách nhân viên và hiển thị chức năng.  9. Người dùng thực hiện chức năng Quản lý nhân viên.  10. Người dùng thực hiện thành công chức năng. |
| Các luồng thay thế | A1: Email, password không hợp lệ:  1. Hệ thống hiển thị lại form đăng nhập kèm với thông báo email hoặc mật khẩu không hợp lệ.  2. Quay lại bước 2 trong luồng chính. |
| **Pre-Condition(s):**  Điều kiện đầu vào  (tiền điều kiện) | Người dùng đã có tài khoản trên hệ thống. |
| **Post-Condition(s):**  Điều kiện đầu ra  (hậu điều kiện) | Người dùng đăng nhập thành công. |

Quản lý sản phẩm

|  |  |
| --- | --- |
| **Use case** | **Nội Dung** |
| Tên use-case | Quản lý sản phẩm. |
| Mô tả | Use case này cho phép người dùng quản lý sản phẩm. |
| Actors | Người quản lý, Nhân viên. |
| Điều kiện kích hoạt | Khi người dùng chọn Quản lý sản phẩm |
| Luồng cơ bản | 1. Tại trang chủ, người dùng chọn “Đăng nhập”.  2. Hệ thống hiển thị form đăng nhập.  3. Người dùng nhập các thông tin bao gồm: email, password.  4. Người dùng nhấn “Đăng nhập”.  5. Hệ thống kiểm tra thông tin đăng nhập (email, password).  A1: Email, password không hợp lệ.  6. Đăng nhập thành công, quay trang chủ.  7. Người dùng chọn Quản lý Sản phẩm.  8. Hệ thống hiển thị danh sách sản phẩm và hiển thị chức năng.  9. Người dùng thực hiện chức năng Quản lý sản phẩm.  10. Người dùng thực hiện thành công chức năng. |
| Các luồng thay thế | A1: Email, password không hợp lệ:  1. Hệ thống hiển thị lại form đăng nhập kèm với thông báo email hoặc mật khẩu không hợp lệ.  2. Quay lại bước 2 trong luồng chính. |
| **Pre-Condition(s):**  Điều kiện đầu vào  (tiền điều kiện) | Người dùng đã có tài khoản trên hệ thống. |
| **Post-Condition(s):**  Điều kiện đầu ra  (hậu điều kiện) | Người dùng đăng nhập thành công. |

Quản lý hóa đơn

|  |  |
| --- | --- |
| **Use case** | **Nội Dung** |
| Tên use-case | Quản lý hóa đơn. |
| Mô tả | Use case này cho phép người dùng quản hóa lý hóa đơn. |
| Actors | Người quản lý, Nhân viên. |
| Điều kiện kích hoạt | Khi người dùng chọn Quản lý hóa đơn. |
| Luồng cơ bản | 1. Tại trang chủ, người dùng chọn “Đăng nhập”.  2. Hệ thống hiển thị form đăng nhập.  3. Người dùng nhập các thông tin bao gồm: email, password.  4. Người dùng nhấn “Đăng nhập”.  5. Hệ thống kiểm tra thông tin đăng nhập (email, password).  A1: Email, password không hợp lệ.  6. Đăng nhập thành công, quay trang chủ.  7. Người dùng chọn Đơn hàng.  8. Hệ thống hiển thị danh sách hóa đơn và hiển thị chức năng.  9. Người dùng thực hiện chức năng quản lý hóa đơn.  10. Người dùng thực hiện thành công chức năng. |
| Các luồng thay thế | A1: Email, password không hợp lệ:  1. Hệ thống hiển thị lại form đăng nhập kèm với thông báo email hoặc mật khẩu không hợp lệ.  2. Quay lại bước 2 trong luồng chính. |
| **Pre-Condition(s):**  Điều kiện đầu vào  (tiền điều kiện) | Người dùng đã có tài khoản trên hệ thống. |
| **Post-Condition(s):**  Điều kiện đầu ra  (hậu điều kiện) | Người dùng đăng nhập thành công. |

* + - 1. Biểu đồ phân rã Use Case

|  |  |
| --- | --- |
| |  | | --- | | Phân rã Use Case quản lý khách hàng | |
|  |
| |  | | --- | | Phân rã Use Case quản lý danh mục | |  | |
| |  | | --- | | Phân rã Use Case quản lý nhà cung cấp | |  | |
| |  | | --- | | Phân rã Use Case quản lý nhân viên | |  | |
| |  | | --- | | Phân rã Use Case quản lý sản phẩm | |  | |
| |  | | --- | | Phân rã Use Case quản lý hóa đơn | |  | |

* + 1. Biểu đồ lớp
       1. Lớp và thành phần của lớp

**Nhân viên:** mã nhân viên, họ tên, giới tính, email, số điện thoại, địa chỉ, ngày sinh, lương, quyền, ngày làm việc, trạng thái.

**Khách hàng:** mã khách hàng, họ tên, giới tính, email, số điện thoại, địa chỉ, ngày sinh.

**Danh mục:** mã danh mục, tên danh mục, trạng thái.

**Sản phẩm:** mã sản phẩm, mã danh mục, tên sản phẩm, số lượng, giá, trạng thái.

**Nhà cung cấp:** mã nhà cung cấp, tên nhà cung cấp, trạng thái.

**Hóa đơn:** mã hóa đơn, mã nhân viên, mã khách hàng, ngày xuất, tổng tiền.

**Hóa đơn chi tiết:** mã hóa đơn, mã sản phẩm, số lượng giá.

**Phiếu nhập:** mã phiếu nhập, mã nhà cung cấp, mã nhân viên, tổng tiền.

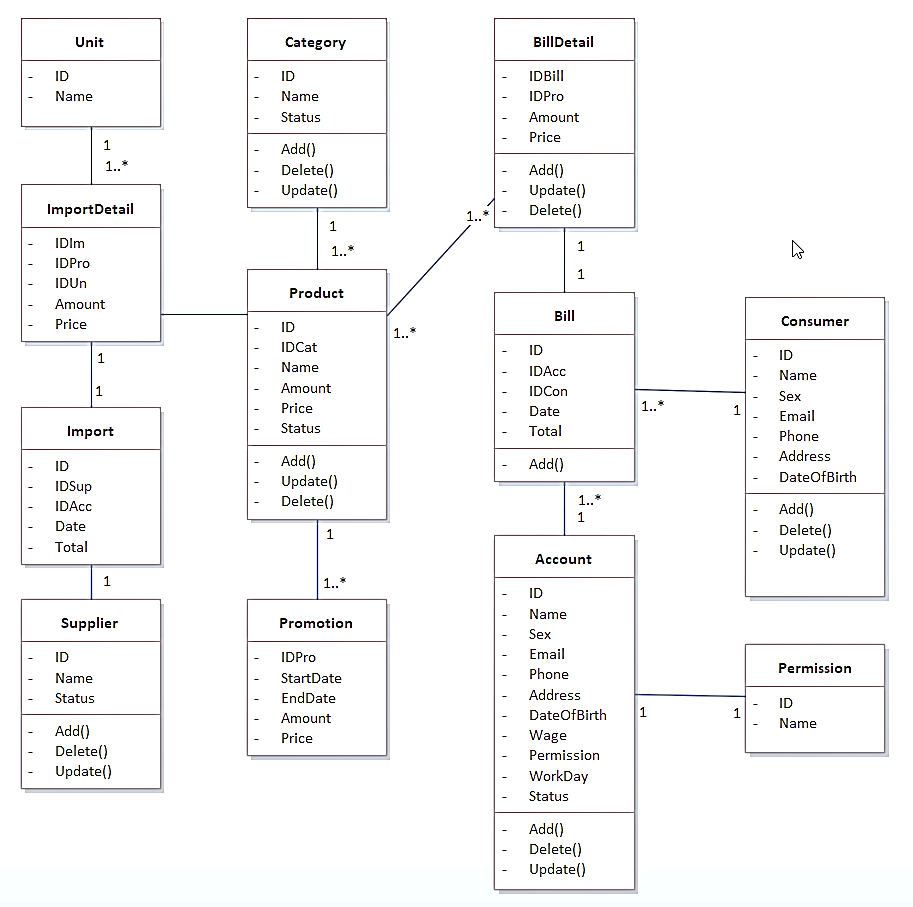
**Phiếu nhập chi tiết:** mã phiếu nhập, mã sản phẩm, mã đơn vị tính, số lượng, giá.

**Khuyến mãi:** mã sản phẩm, ngày bắt đầu, ngày kết thúc, số lượng, giá.

**Đơn vị tính:** mã đơn vị tính, tên đơn vị tính.

**Phân quyền:** mã quyền, tên quyền.

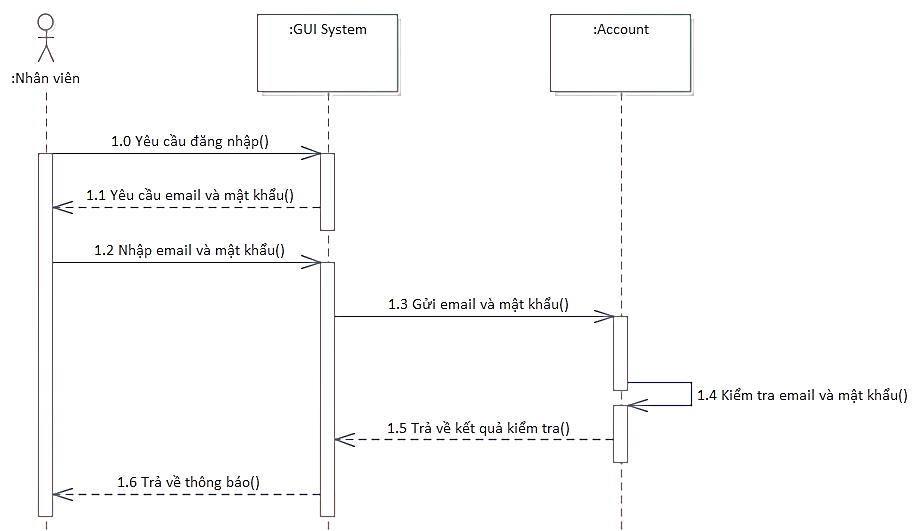
* + - 1. Biểu đồ lớp



* + 1. Biểu đồ dữ liệu



* + 1. Biểu đồ trình tự
       1. Đăng nhập



* + - 1. Thêm

|  |
| --- |
| 1. Thêm khách hàng |
|  |
| Thêm danh mục |
|  |
| Thêm sản phẩm |
|  |
| Thêm nhà cung cấp |
|  |
| Thêm nhân viên |
|  |
| Thêm hóa đơn |
|  |

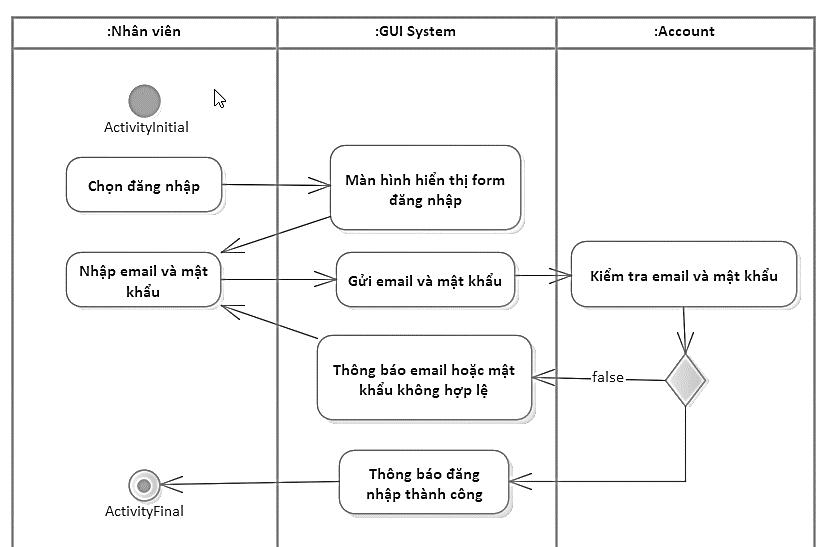
* + - 1. Sửa

|  |
| --- |
| 1. Sửa khách hàng |
|  |
| Sửa danh mục |
|  |
| Sửa sản phẩm |
|  |
| Sửa nhà cung cấp |
|  |
| Sửa nhân viên |
|  |
| Sửa hóa đơn |
|  |

* + - 1. Xóa

|  |
| --- |
| 1. Xóa khách hàng |
|  |
| Xóa danh mục |
|  |
| Xóa sản phẩm |
|  |
| Xóa nhà cung cấp |
|  |
| Xóa nhân viên |
|  |
| Xóa hóa đơn |
|  |

* + 1. Biểu đồ hoạt động
       1. Đăng nhập



* + - 1. Thêm

|  |
| --- |
| 1. Thêm khách hàng |
|  |
| Thêm danh mục |
|  |
| Thêm sản phẩm |
|  |

|  |
| --- |
| Thêm nhà cung cấp |
|  |
| Thêm nhân viên |
|  |
| Thêm hóa đơn |
|  |

* + - 1. Sửa

|  |
| --- |
| 1. Sửa khách hàng |
|  |
| Sửa danh mục |
|  |
| Sửa sản phẩm |
|  |
| Sửa nhà cung cấp |
|  |
| Sửa nhân viên |
|  |
| Sửa hóa đơn |
|  |

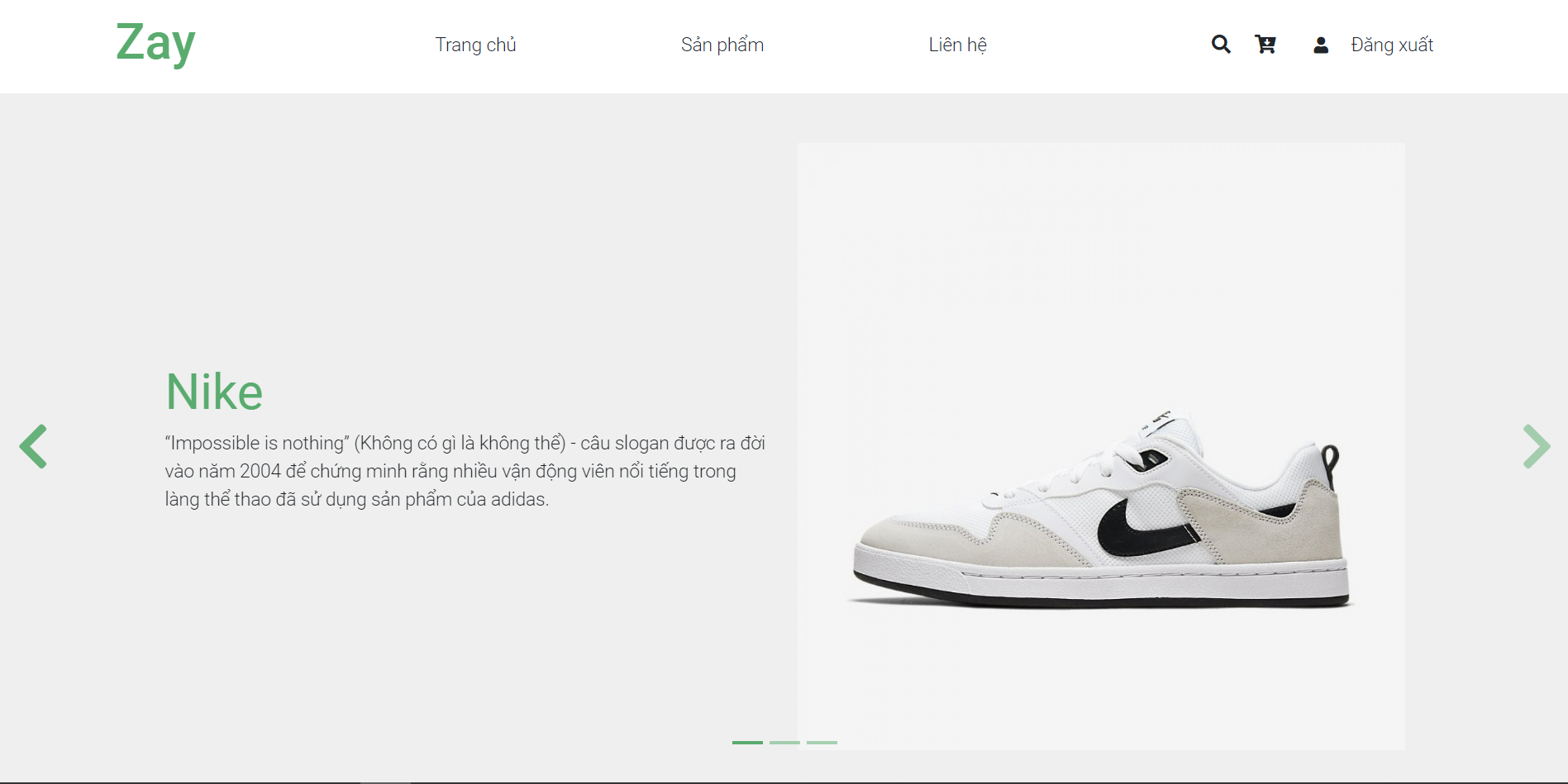
* + - 1. Xóa

|  |
| --- |
| 1. Xóa khách hàng |
|  |
| Xóa danh mục |
|  |
| Xóa sản phẩm |
|  |
| Xóa nhà cung cấp |
|  |
| Xóa nhân viên |
|  |
| Xóa hóa đơn |
|  |

1. TRIỂN KHAI CÀI ĐẶT VÀ ĐÁNH GIÁ
   * 1. Triển khai và cài đặt
        1. Giao diện người dùng

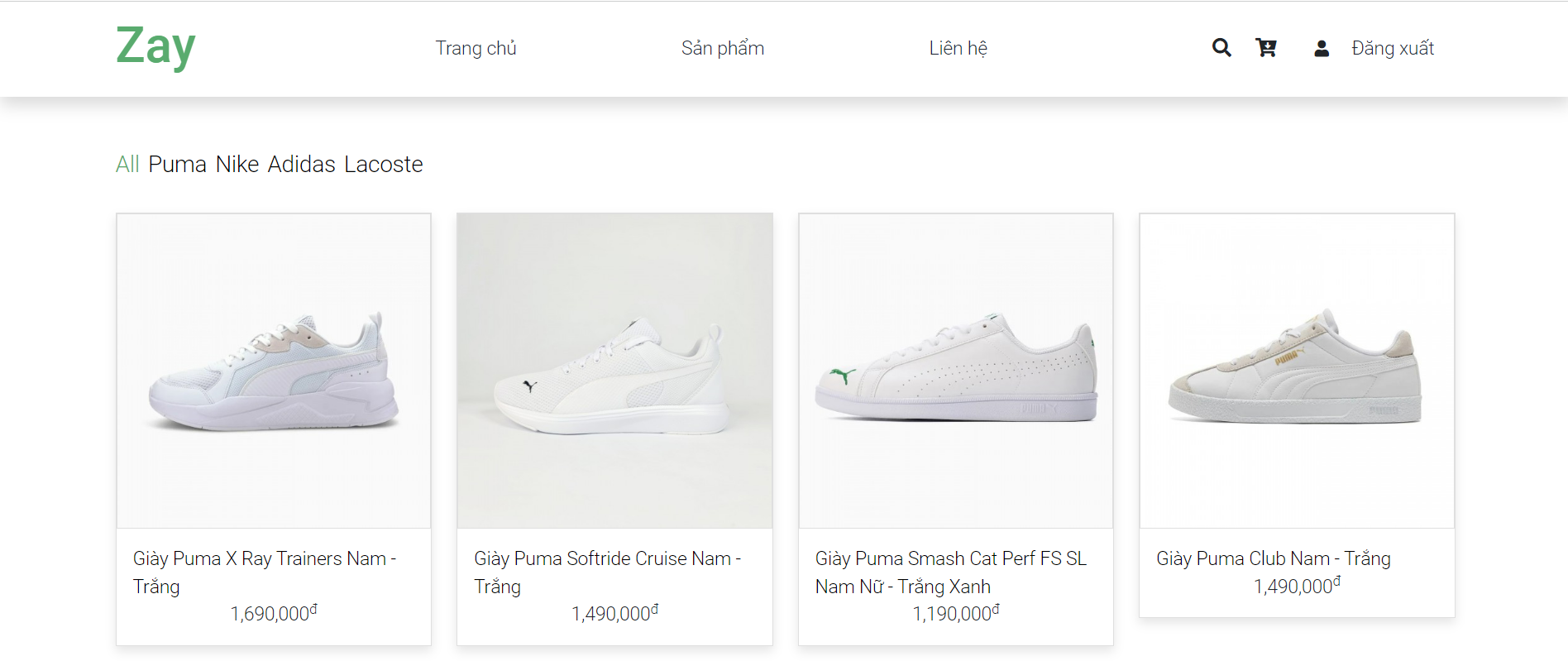
Trang chủ

Trang chủ hay còn gọi là “Homepage” là trang đầu tiên khi người dùng truy cập vào website của bạn. Đây là trang web mặc định khi bạn truy cập vào địa chỉ website thì chỉ chứa tên miền đó. Vì vậy, đây sẽ là trang được hiển thị ra những thành phần quan trọng như: Slider, Danh mục, Sản phẩm hot nhất, bài viết mới nhất,…



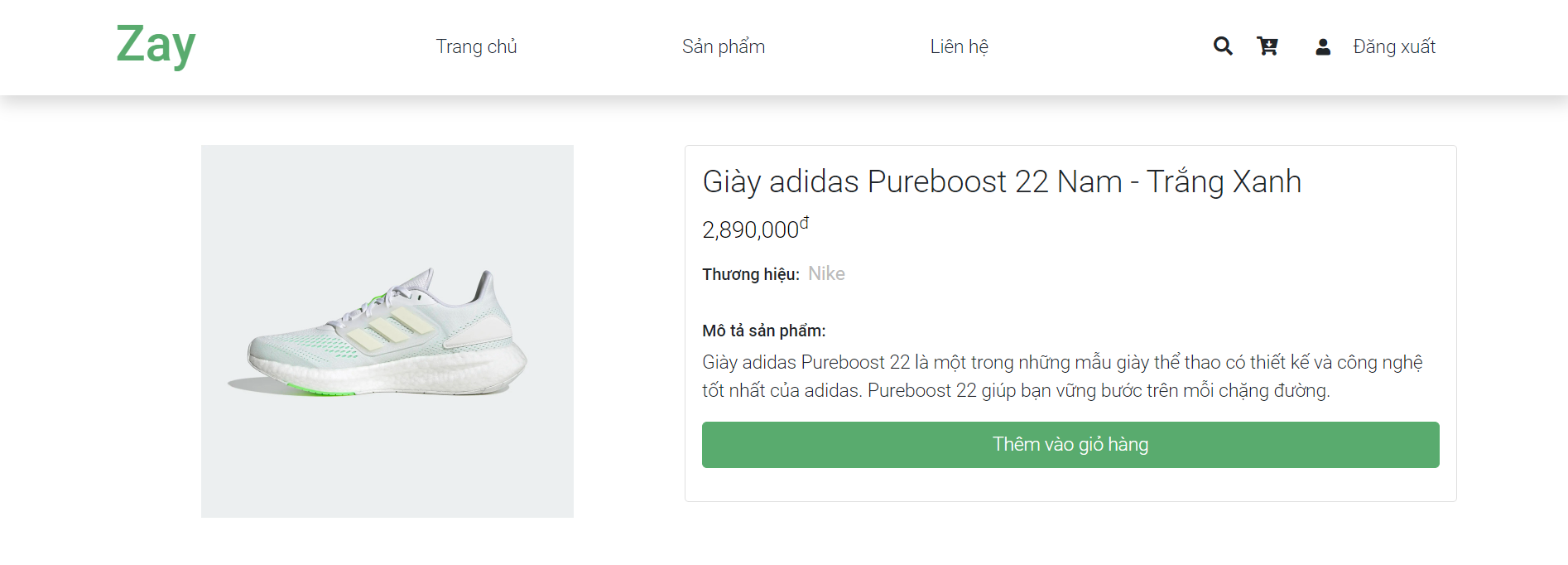
Trang danh sách sản phẩm

Trang sản phẩm là trang được hiển thị tất cả các sản phẩm trong hệ thống.



Trang chi tiết sản phẩm

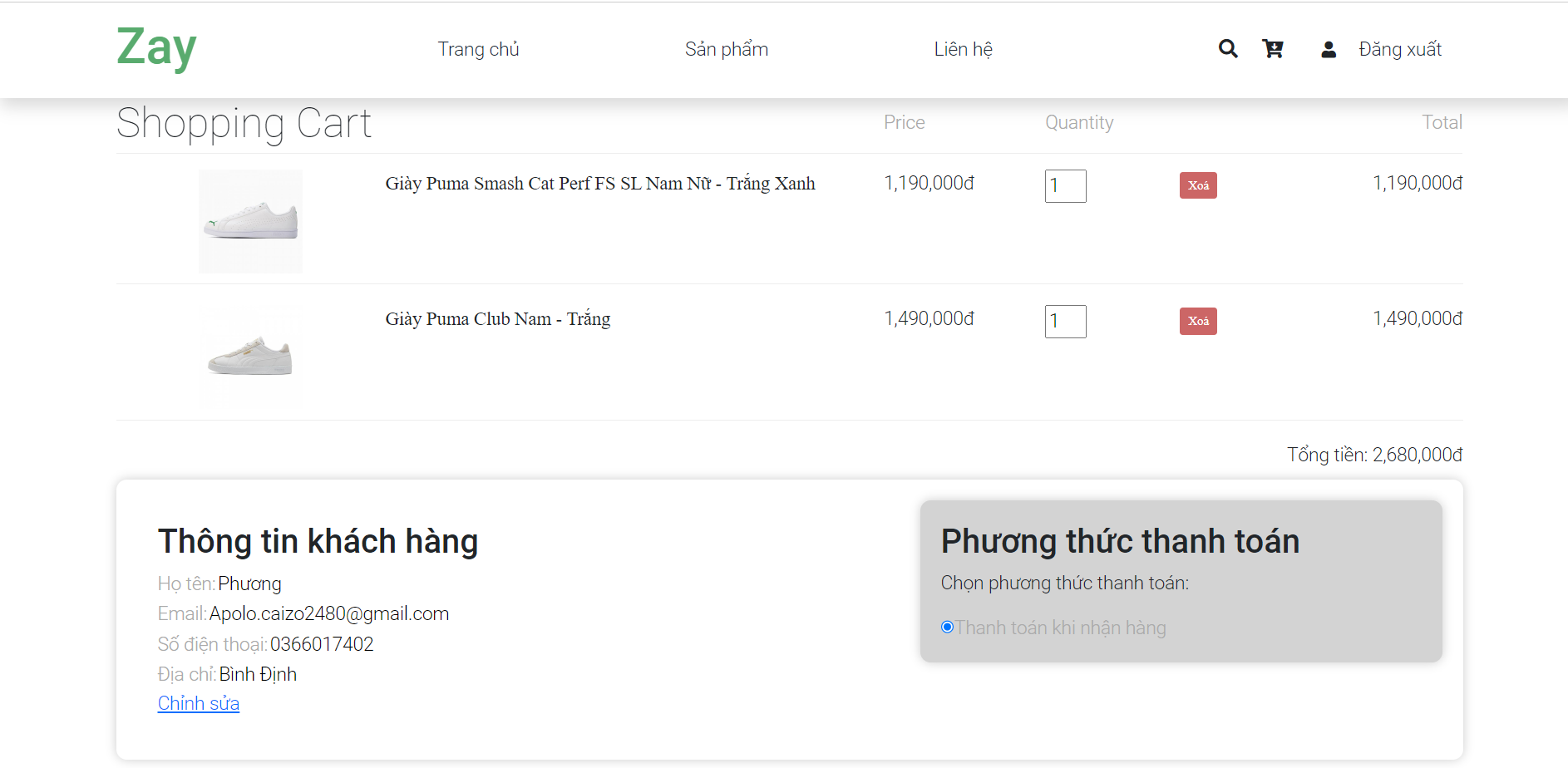
Đây là trang sẽ hiển thị đầy đủ thông tin chi tiết của sản phẩm.



Trang giỏ hàng và thanh toán

Thường thì các trang web sẽ có trang và trang thanh toán riêng biệt. Nhưng nếu gộp 2 trang này lại với nhau thì cũng có vài lí do để làm như vậy:

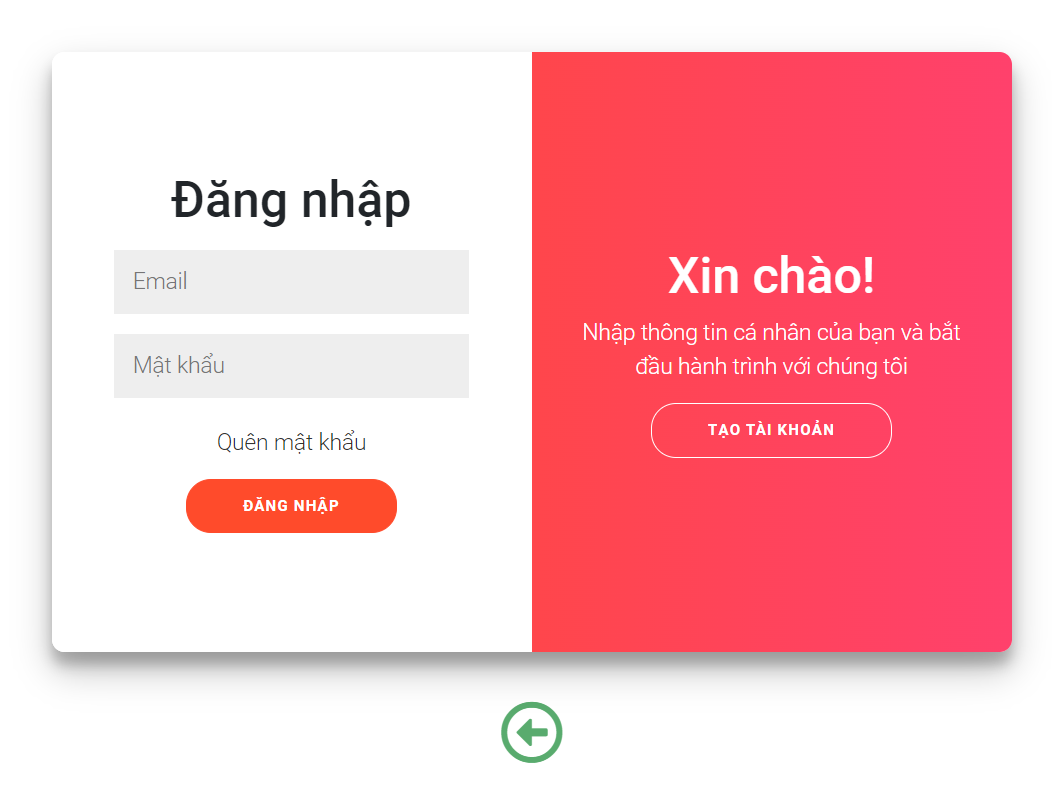
* Tăng tính tiện dụng: Gộp trang giỏ hàng và thanh toán lại với nhau tạo ra sự tiện dụng cho người dùng, giúp họ tiết kiệm thời gian vì không phải di chuyển giữa các trang khác nhau để hoàn thành quá trình mua sắm.
* Giảm tỷ lệ bỏ giỏ hàng: Khi khách hàng phải điều hướng qua nhiều trang để hoàn thành quá trình mua sắm, tỷ lệ bỏ giỏ hàng sẽ tăng cao vì nhiều khách hàng có thể cảm thấy thiếu kiên nhẫn. Khi gộp trang giỏ hàng và thanh toán, khách hàng có thể dễ dàng xem lại các sản phẩm trong giỏ hàng và thanh toán một lần duy nhất.
* Cải thiện trải nghiệm người dùng: Việc gộp trang giỏ hàng và thanh toán là một cách để cải thiện trải nghiệm người dùng của website. Người dùng sẽ cảm thấy thoải mái hơn khi mua sắm và có thể tập trung vào việc chọn sản phẩm thay vì phải lo lắng về việc thanh toán của mình.

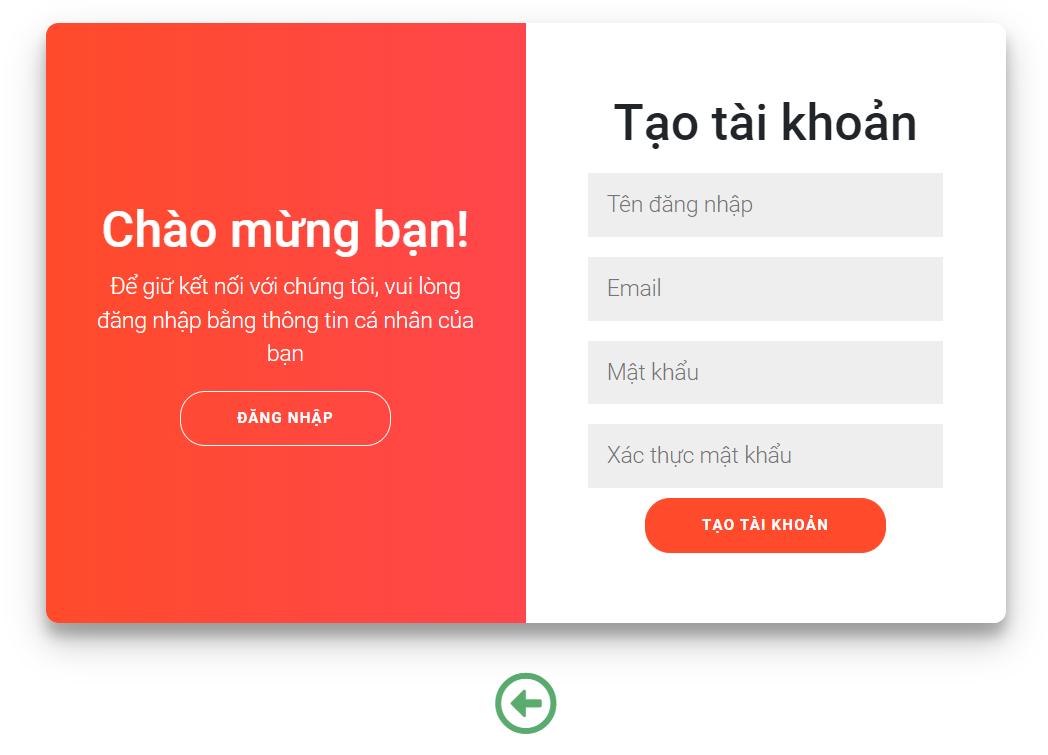


Trang đăng nhập và đăng ký

Khách hàng tiến hành đăng nhập/đăng ký bằng trang này.

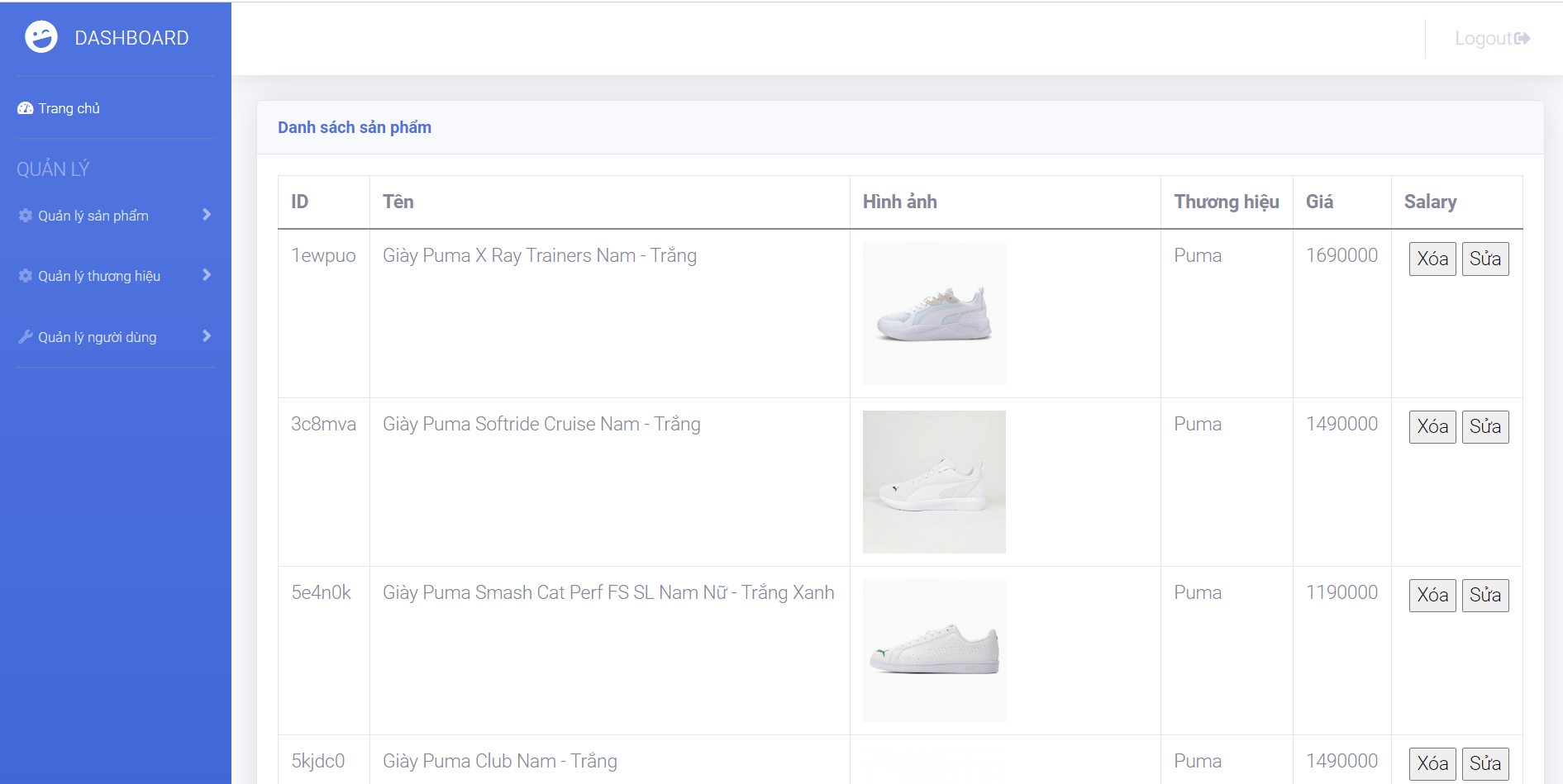
Trang đăng nhập sẽ được dùng chung cho cả người dùng và người quản trị.

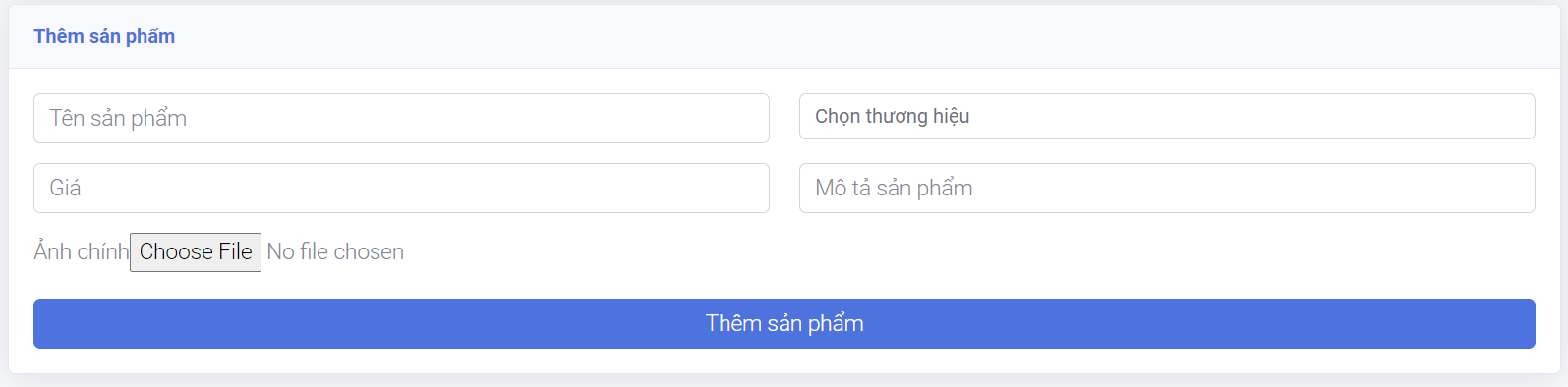




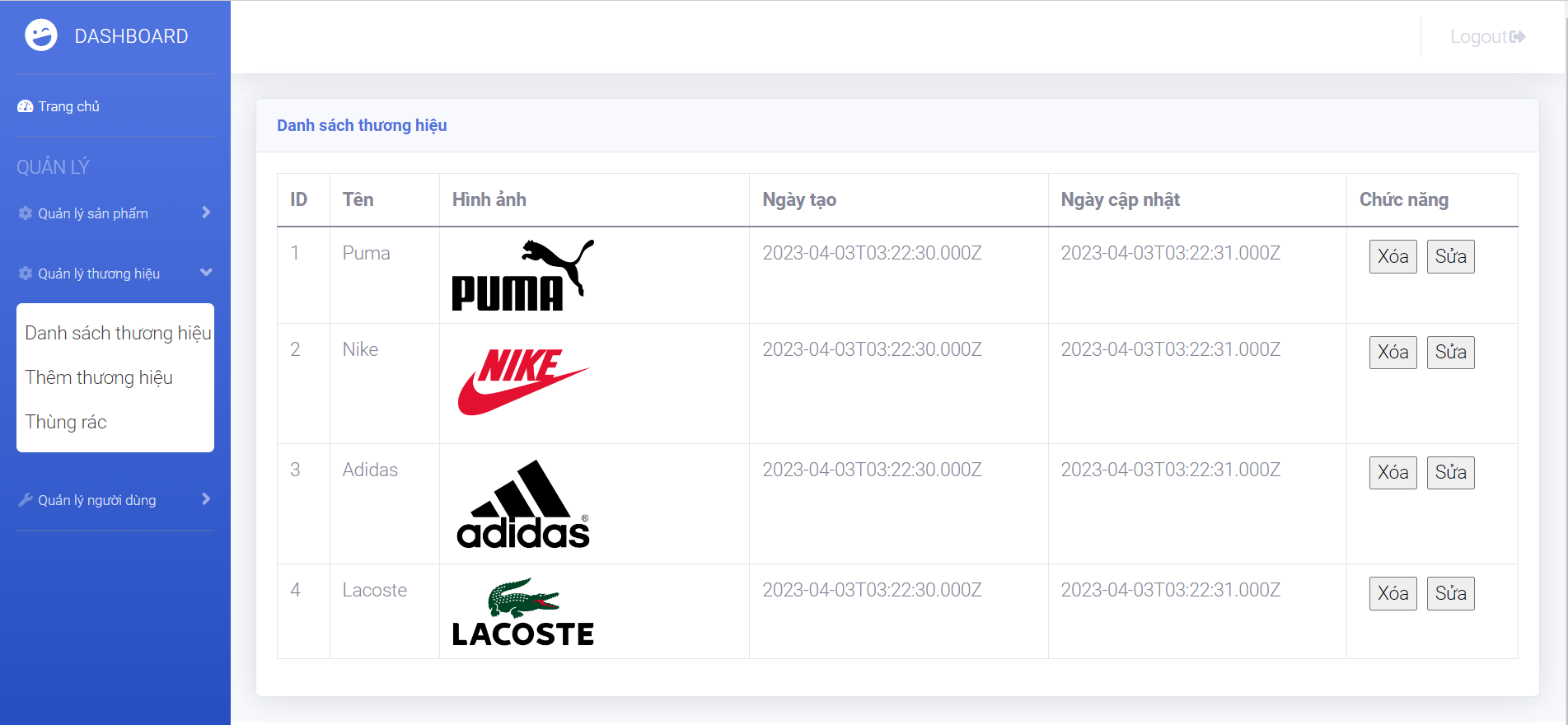
* + - 1. Giao diện quản trị

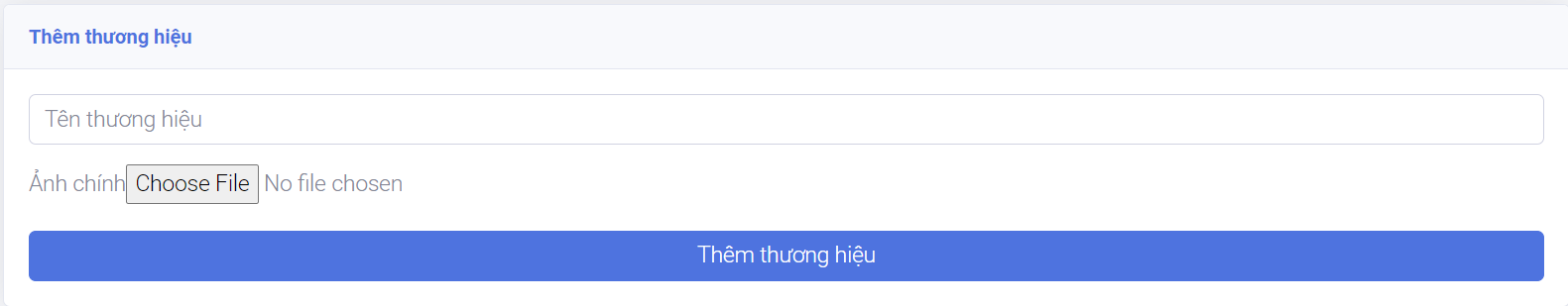
Trang quản lý sản phẩm



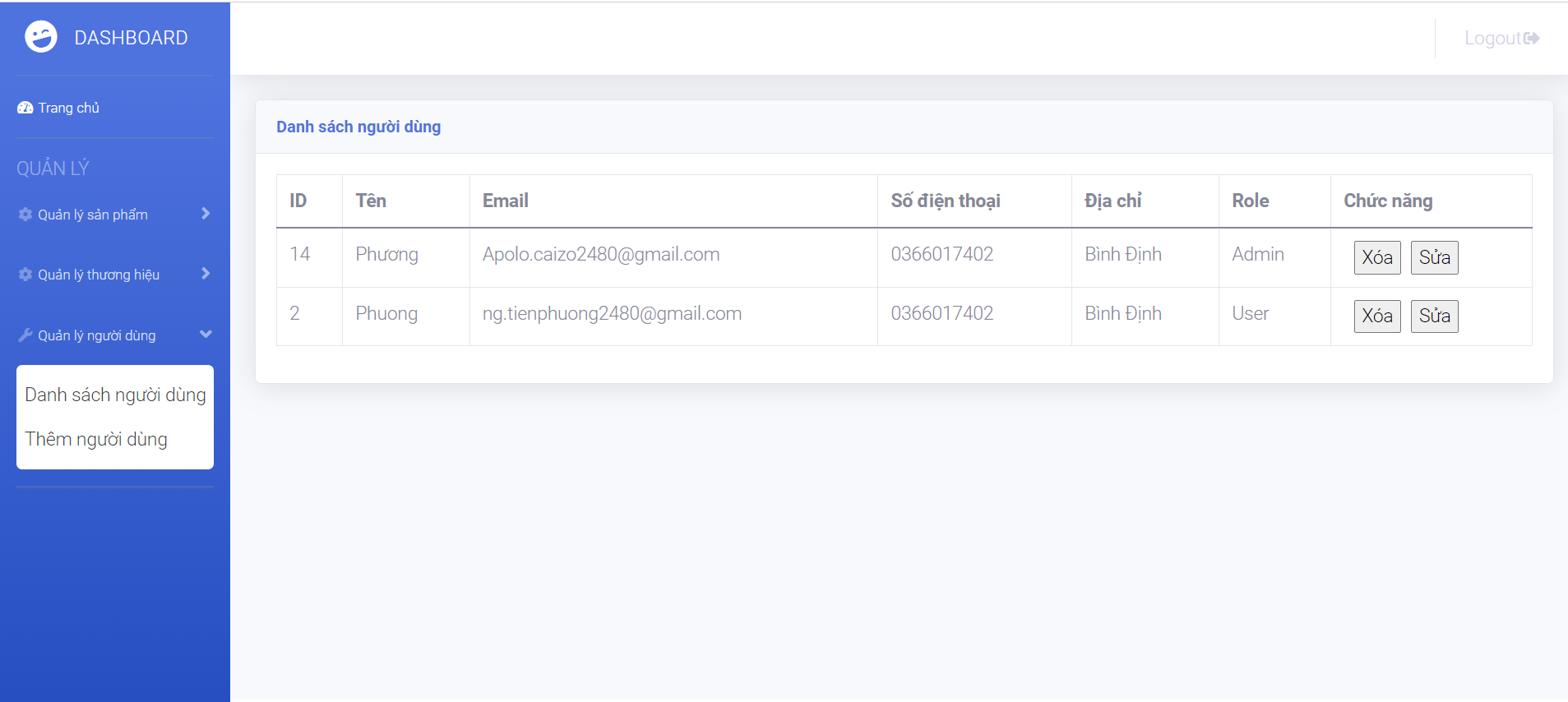


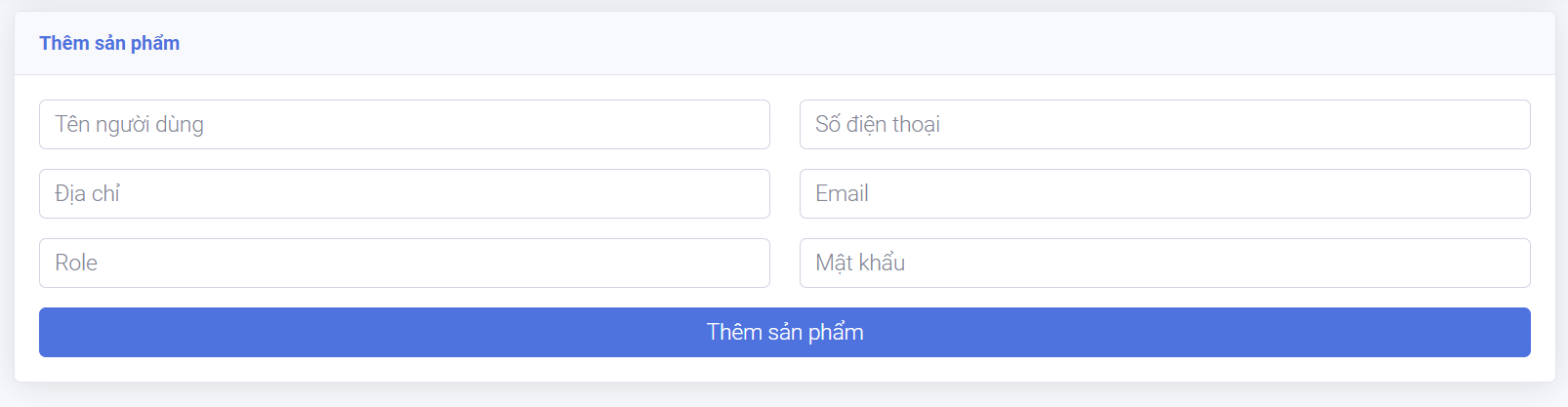
Trang quản lý thương hiệu





Trang quản lý người dùng





* + 1. Đánh giá kết quả
       1. Kết quả

Giao diện hiện đại, thân thiện

Trình bày được tất cả sản phẩm trong hệ thống cho khách hàng tùy ý lựa chọn

Đăng nhập, đăng ký

Và một số chức năng khác

Tóm lại, Website đã có đầy đủ những gì mà website thương mại điện tử có. Điều đó hỗ trỡ rất nhiều cho việc kinh doanh và thu về nhiều lợi nhuận

* + - 1. Hạn chế của đề tài

Do làm việc cá nhân nên việc thu thập dữ liệu, phân tích nghiệp vụ còn nhiều hạn chế

Chưa thể cập nhật hết được tất cả những sản phẩm hiện đang có mặt trên thị trường.

Vì kiến thức còn hạn hẹp nên website còn thô sơ, trong thời gian sắp tới tôi sẽ cố gắng mở rộng ứng dụng để có thể áp dụng cho những doanh nghiệp lớn thay vì chỉ áp dụng cho mô hình nhỏ như hiện tại

TÀI LIỆU THAM KHẢO

HTML

<https://vi.wikipedia.org/wiki/HTML>

CSS

<https://vi.wikipedia.org/wiki/CSS>

JavaScript

<https://aws.amazon.com/vi/what-is/javascript/>

SQL

<https://aws.amazon.com/vi/what-is/sql/>

ReactJS

<https://viblo.asia/p/reactjs-la-gi-nhung-dieu-co-the-ban-chua-biet-ve-reactjs-eW65G4RRKDO>

NodeJS

<https://vietnix.vn/nodejs-la-gi/>