KHOA KỸ THUẬT VÀ CÔNG NGHỆ

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**

****

**THỰC TẬP ĐỒ ÁN CƠ SỞ NGÀNH**

**HỌC KỲ I, NĂM HỌC 2024-2025**

**THIẾT KẾ FRONT-END**

**CHO SÀN THƯƠNG MẠI SẢN PHẨM HANDMADE CỦA SINH VIÊN TVU**

*Giảng viên hướng dẫn:*

ThS. Phạm Thị Trúc Mai

*Sinh viên thực hiện:*

Họ tên: Thạch Thị Xuân Linh

MSSV: 110122013

Lớp: DA22TTA

***Trà Vinh, tháng 12 năm 2024***

KHOA KỸ THUẬT VÀ CÔNG NGHỆ

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**

****

**THỰC TẬP ĐỒ ÁN CƠ SỞ NGÀNH**

**HỌC KỲ I, NĂM HỌC 2024-2025**

**THIẾT KẾ FRONT-END**

**CHO SÀN THƯƠNG MẠI SẢN PHẨM HANDMADE CỦA SINH VIÊN TVU**

*Giảng viên hướng dẫn:*

ThS. Phạm Thị Trúc Mai

*Sinh viên thực hiện:*

Họ tên: Thạch Thị Xuân Linh

MSSV: 110122013

Lớp: DA22TTA

***Trà Vinh, tháng 12 năm 2024***

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

*Trà Vinh, ngày ….. tháng …… năm ……*

**Giáo viên hướng dẫn**

*(Ký tên và ghi rõ họ tên)*

*Trà Vinh, ngày ….. tháng …… năm ……*

**Giáo viên hướng dẫn**

*(Ký tên và ghi rõ họ tên)*

**LỜI CẢM ƠN**

**NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG**

*Trà Vinh, ngày ….. tháng …… năm ……*

**Thành viên hội đồng**

*(Ký tên và ghi rõ họ tên)*

Em xin gửi lời cảm ơn đến giảng viên hướng dẫn Cô Phạm Thị Trúc Mai vì đã dành thời gian quý báu, tận tình chỉ bảo từng chi tiết nhỏ, định hướng và hỗ trợ em hoàn thành bài báo cáo này. Những ý kiến đóng góp cô không chỉ giúp em hoàn thiện bài báo cáo mà còn mở ra nhiều góc nhìn mới, giúp em nâng cao kỹ năng nghiên cứu và áp dụng thực tế.

Xin chân thành cảm ơn Cô.

**MỤC LỤC**

[CHƯƠNG 1: TỔNG QUAN 10](#_Toc186390541)

[1.1 Giới thiệu tổng quan về vấn đề và cách giải quyết 10](#_Toc186390542)

[1.2 Thương mại điện tử 10](#_Toc186390543)

[1.3 Lập trình front-end cho web UX, UI 10](#_Toc186390544)

[CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT 11](#_Toc186390545)

[1.1 Ngôn ngữ HTML 11](#_Toc186390546)

[*1.1.1* *Giới thiệu* 11](#_Toc186390547)

[*1.1.2* *Cấu trúc của HTML* 11](#_Toc186390548)

[*1.1.3* *Lịch sử của HTML* 11](#_Toc186390549)

[*2.1.4 Các loại HTML hiện nay* 11](#_Toc186390550)

[1.2 Ngôn ngữ CSS 12](#_Toc186390551)

[*1.2.1* *Giới thiệu* 12](#_Toc186390552)

[*1.2.2* *Bố cục và cấu trúc của một đoạn CSS* 12](#_Toc186390553)

[*1.2.3* *Cách nhúng CSS vào website* 13](#_Toc186390554)

[1.3 Ngôn ngữ JavaScript 13](#_Toc186390555)

[*1.3.1* *Lịch sử phát triển* 14](#_Toc186390556)

[*1.3.2* *Cách hoạt động của ngôn ngữ JavaScript* 14](#_Toc186390557)

[*1.3.3* *Ứng dụng* 14](#_Toc186390558)

[1.4 API 15](#_Toc186390559)

[*1.4.1* *Giới thiệu* 15](#_Toc186390560)

[*1.4.2* *Ứng dụng* 15](#_Toc186390561)

[*1.4.3* *Web API* 15](#_Toc186390562)

[*1.4.4* *Cách Web API hoạt động* 16](#_Toc186390563)

[1.5 JSON API 16](#_Toc186390564)

[*1.5.1* *Giới thiệu JSON API* 16](#_Toc186390565)

[*1.5.2* *Cấu trúc JSON API* 17](#_Toc186390566)

[*1.5.3* *Hướng dẫn sử dụng JSON API* 17](#_Toc186390567)

[1.6 Bootstrap 18](#_Toc186390568)

[*1.6.1* *Giới thiệu* 18](#_Toc186390569)

[*1.6.2* *Lịch sử Bootstrap* 18](#_Toc186390570)

[*1.6.3* *Tính năng của Bootstrap* 19](#_Toc186390571)

[CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU 20](#_Toc186390572)

[3.1 Mô tả các bước nghiên cứu 20](#_Toc186390573)

[3.1.1 *Xây dựng file JSON* 20](#_Toc186390574)

[*3.1.2 Các phương thức để hiển thị sản phẩm từ file JSON lên web* 21](#_Toc186390575)

[*3.1.3 Các bản thiết kế* 22](#_Toc186390576)

[CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU 27](#_Toc186390577)

[4.1 Kết quả giao diện các trang web 27](#_Toc186390578)

[*4.1.1 Trang chủ* 27](#_Toc186390579)

[*4.1.2 Trang sản phẩm* 27](#_Toc186390580)

[*4.1.3 Trang chi tiết sản phẩm* 27](#_Toc186390581)

[*4.1.4 Giao diện các gian hàng* 28](#_Toc186390582)

[*4.1.5 Giao diện chi tiết gian hàng* 28](#_Toc186390583)

[*4.1.6 Giao diện giỏ hàng* 29](#_Toc186390584)

[*4.1.7 Giao diện đăng nhập của chủ gian hàng* 29](#_Toc186390585)

[*4.1.8 Giao diện quản lý gian hàng* 30](#_Toc186390586)

[5.1 Kết luận kết quả đạt được 31](#_Toc186390587)

[5.2 Nhược điểm 31](#_Toc186390588)

[5.3 Hướng phát triển 31](#_Toc186390589)

**DANH MỤC HÌNH ẢNH**

[**Hình 3.1** Giao diện bảng thiết kế trang chủ 23](#_Toc186390626)

[**Hình 3.2** Giao diện bảng thiết kế trang sản phẩm 24](#_Toc186390627)

[**Hình 3.3** Giao diện bảng thiết kế trang chi tiết sản phẩm 24](#_Toc186390628)

[**Hình 3.4** Giao diện bảng thiết kế trang gian hàng 25](#_Toc186390629)

[**Hình 3.5** Giao diện bảng thiết kế trang chi tiết gian hàng 25](#_Toc186390630)

[**Hình 3.6** Giao diện bảng thiết kế trang giỏ hàng 26](#_Toc186390631)

[**Hình 3.7** Giao diện bảng thiết kế trang đăng nhập chủ gian hàng 26](#_Toc186390632)

[**Hình 3.8** Giao diện bảng thiết kế trang quản lý gian hàng 27](#_Toc186390633)

[**Hình 4.1** Giao diện trang chủ web 28](#_Toc186390634)

[**Hình 4.2** Giao diện trang sản phẩm 28](#_Toc186390635)

[**Hình 4.3** Giao diện trang chi tiết sản phẩm 29](#_Toc186390636)

[**Hình 4.4** Giao diện các gian hàng 29](#_Toc186390637)

[**Hình 4.5** Giao diện trang chi tiết gian hàng 30](#_Toc186390638)

[**Hình 4.6** Giao diện giỏ hàng 30](#_Toc186390639)

[**Hình 4.7** Giao diện đăng nhập của chủ gian hàng 31](#_Toc186390640)

[**Hình 4.8** Giao diện quản lý gian hàng 31](#_Toc186390641)

**DANH MỤC TỪ VIẾT TẮT**

|  |  |
| --- | --- |
| **Từ viết tắt** | **Ý nghĩa** |
| API | Application Programming Interface |
| CSS | Cascading Style Sheets |
| DOM | Document Object Model |
| HTML | HyperText Markup Language |
| HTTP | HyperText Transfer Protocol |
| HTTPS | HyperText Transfer Protocol Secure |
| JSON | JavaScript Object Notation |
| MVC | Model-View-Controller |
| PDF | Portable Document Format |
| URI | Uniform Resource Identifier |
| URL | Uniform Resource Locator |
| W3C | World Wide Web Consortium |
| XML | Extensible Markup Language |
| XHTML | Extensible HyperText Markup Language |

**TÓM TẮT ĐỒ ÁN CƠ SỞ NGÀNH**

Hiện nay, công nghệ phát triển việc mua sắm online được áp dụng nhiều trong đời sống của mỗi người, từ đó các website về thương mại điện tử xuất hiện nhiều, bài báo cáo này em tập trung vào thiết kế front-end cho sàn thương mại sản phẩm handmade của sinh viên TVU, nhằm quảng bá sản phẩm do tự tay sinh viên sáng tạo ra.

Em sử dụng các ngôn ngữ lập trình như HTML, CSS, JavaScript,… kết hợp với Bootstrap framework để xây dựng trang web tạo dữ liệu mẫu dưới dạng JSON. Sử dụng json-server để tạo API giả lập từ dữ liệu mẫu. Các trang bao gồm: trang chủ, trang sản phẩm theo loại, trang gian hàng, trang chi tiết sản phẩm, giỏ hàng và form mua hàng. Tích hợp chức năng như tìm kiếm sản phẩm, quản lý giỏ hàng (thêm/xóa sản phẩm, tính tổng tiền).

Sử dụng Bootstrap để phát triển giao diện. Tạo file JSON chứa thông tin sản phẩm và sử dụng json-server để tạo API giả lập. Sử dụng JavaScript để lấy dữ liệu từ API và hiển thị trên giao diện người dùng. Tạo các chức năng tìm kiếm sản phẩm và quản lý giỏ hàng bằng JavaScript.

Hoàn thiện giao diện thương mại điện tử và một số chức năng.

**MỞ ĐẦU**

**Lý do chọn đề tài**

Trong thời đại số hóa ngày nay, thương mại điện tử đã trở thành một phần không thể thiếu trong đời sống kinh tế và xã hội. Tuy nhiên, các nền tảng thương mại điện tử hiện nay chủ yếu tập trung vào các sản phẩm công nghiệp, thời trang và công nghệ, mà chưa có nhiều sự chú trọng đến các sản phẩm handmade, đặc biệt là những sản phẩm do sinh viên tạo ra. Sinh viên thường có nhiều ý tưởng sáng tạo và sản phẩm độc đáo, nhưng lại thiếu môi trường để giới thiệu và bán chúng. Vì thế, em quyết định chọn đề tài “***Thiết kế front-end cho sàn thương mại sản phẩm handmade của sinh viên TVU***” không chỉ giúp họ quảng bá sản phẩm mà còn hỗ trợ họ trong việc học hỏi và phát triển kỹ năng kinh doanh, lập trình.

Việc chọn HTML, CSS, JavaScript và Bootstrap để lập trình trang web mang lại nhiều lợi ích. HTML, ngôn ngữ đánh dấu siêu văn bản, là nền tảng của bất kỳ trang web nào, cho phép tạo ra cấu trúc và nội dung như tiêu đề, đoạn văn, hình ảnh và liên kết. CSS giúp định dạng và thiết kế các phần tử HTML, tách biệt phần trình bày và nội dung để dễ dàng bảo trì. JavaScript, ngôn ngữ lập trình phổ biến, mang lại tính tương tác và động cho trang web, cho phép xử lý sự kiện, xác thực biểu mẫu và tạo các hiệu ứng động. Bootstrap, một framework CSS phổ biến, giúp tiết kiệm thời gian phát triển nhờ cung cấp các công cụ và thành phần giao diện sẵn có, đảm bảo tính nhất quán và phản hồi tốt trên nhiều thiết bị. Kết hợp các công cụ này, có thể tạo ra các trang web đẹp, phản hồi nhanh và tương tác cao, đáp ứng mọi nhu cầu và phong cách thiết kế.

**Mục tiêu nghiên cứu**

Mục tiêu chính của đề tài này là phát triển một giao diện thương mại điện tử để giới thiệu và bán các sản phẩm handmade do sinh viên TVU làm ra. Sử dụng framework bootstrap và các ngôn ngữ lập trình web như HTML, CSS, JavaScript để xây dựng

Tạo ra một nền tảng kinh doanh thực tế, giúp sinh viên thực hành các kỹ năng khởi nghiệp và kinh doanh online.

**Đối tượng và phạm vi nghiên cứu**

**Đối tượng nghiên cứu**

Sản phẩm handmade: Các sản phẩm thủ công như hoa đan, hoa đất sét, hoa vải, giỏ, áo len, v.v.

Công nghệ và công cụ lập trình: Nghiên cứu và ứng dụng các công cụ và công nghệ lập trình front-end như Bootstrap, JSON và API. Sử dụng Bootstrap để xây dựng giao diện. Sử dụng công cụ lập trình Visual Studio Code để chạy chương trình và kiểm tra.

**Phạm vi nghiên cứu**

Xây dựng giao diện người dùng: Sử dụng Bootstrap để thiết kế và phát triển các trang web cơ bản như trang chủ, trang sản phẩm, trang gian hàng, trang chi tiết sản phẩm, giỏ hàng và form mua hàng. Mô phỏng API: Tạo dữ liệu giả lập bằng file JSON và sử dụng json-server để tạo API giả lập, cung cấp dữ liệu sản phẩm cho giao diện. Tích hợp chức năng: Phát triển các chức năng tìm kiếm sản phẩm, quản lý giỏ hàng và tính toán tổng tiền, sử dụng JavaScript để xử lý và hiển thị dữ liệu từ API.

**Phương pháp nghiên cứu**

Nghiên cứu tài liệu về Bootstrap, JSON, API và JavaScript để xây dựng nền tảng lý thuyết. Phân tích yêu cầu người dùng, thiết kế giao diện bằng Bootstrap và tạo dữ liệu giả lập bằng file JSON, kết hợp json-server để mô phỏng API. Phát triển chức năng hiển thị sản phẩm, tìm kiếm, quản lý giỏ hàng và tính toán tổng tiền bằng JavaScript.

**Phương pháp lý thuyết**

Nghiên cứu và hiểu rõ các công nghệ, công cụ và nguyên tắc lập trình liên quan. Các bước chính trong phương pháp lý thuyết bao gồm: Nghiên cứu tài liệu, tìm hiểu về HTML, CSS, JavaScript và Bootstrap cũng như JSON và cách sử dụng API trong các ứng dụng web.

**Phương pháp thực nghiệm**

Phương pháp thực nghiệm tập trung vào việc áp dụng kiến thức lý thuyết vào thực tiễn để phát triển và kiểm tra các tính năng của dự án. Các bước chính trong phương pháp này bao gồm: Thiết kế giao diện người dùng bằng Bootstrap để phát triển các trang web cơ bản như trang chủ, trang sản phẩm, trang chi tiết sản phẩm, giỏ hàng và form mua hàng. Tạo dữ liệu giả lập bằng file JSON và sử dụng json-server để tạo API giả lập, cung cấp dữ liệu sản phẩm cho giao diện. Sử dụng JavaScript để phát triển các chức năng tìm kiếm sản phẩm, quản lý giỏ hàng và tính toán tổng tiền, đồng thời xử lý và hiển thị dữ liệu từ API trên giao diện người dùng.

# CHƯƠNG 1: TỔNG QUAN

## Giới thiệu tổng quan về vấn đề và cách giải quyết

Với sự phát triển của thương mại điện tử, việc xây dựng các trang web bán hàng trực tuyến trở nên ngày càng quan trọng, đặc biệt đối với các sản phẩm handmade, lĩnh vực cần sự sáng tạo và thể hiện rõ nét giá trị thủ công. Tuy nhiên, nhiều cá nhân và doanh nghiệp nhỏ gặp khó khăn trong việc thiết kế giao diện thân thiện và tích hợp các chức năng quản lý hiệu quả.

Trang web này tập trung vào việc ứng dụng các công nghệ lập trình front-end như Bootstrap, JSON, API, và JavaScript để xây dựng giao diện người dùng cơ bản cho trang web bán sản phẩm handmade. Mục tiêu là phát triển một hệ thống hiển thị sản phẩm, hỗ trợ tìm kiếm, quản lý giỏ hàng, và tính toán tổng tiền, qua đó cung cấp một giải pháp đơn giản nhưng hiệu quả cho các doanh nghiệp nhỏ hoặc cá nhân muốn tham gia kinh doanh trực tuyến.

## Thương mại điện tử

Thương mại điện tử (e-commerce) là quá trình mua bán hàng hóa, dịch vụ và truyền tải dữ liệu thông qua mạng internet. Nó đã trở thành một phần không thể thiếu trong đời sống kinh tế và xã hội hiện đại, giúp người tiêu dùng có thể mua sắm bất kỳ lúc nào, ở bất kỳ đâu chỉ với vài thao tác trên máy tính hoặc điện thoại di động. Thương mại điện tử không chỉ giúp doanh nghiệp tiếp cận một lượng lớn khách hàng tiềm năng mà còn giảm chi phí vận hành thông qua tự động hóa quy trình và tối ưu hóa chuỗi cung ứng. Đặc biệt, sự phát triển của các nền tảng thanh toán trực tuyến, dịch vụ vận chuyển nhanh chóng và chính sách bảo vệ người tiêu dùng đã thúc đẩy sự tin tưởng và gia tăng sự phổ biến của thương mại điện tử.

## Lập trình front-end cho web UX, UI

Lập trình front-end cho web là quá trình xây dựng và triển khai giao diện người dùng của một trang web, bao gồm tất cả các yếu tố trực quan mà người dùng tương tác trực tiếp. Các công nghệ phổ biến trong lập trình front-end bao gồm HTML, CSS và JavaScript, cùng với các thư viện và framework như React, Angular và Vue.js. Trong quá trình phát triển, các nhà lập trình front-end cần phải chú trọng đến UX (User Experience - Trải nghiệm người dùng) và UI (User Interface - Giao diện người dùng). UX đề cập đến cách người dùng tương tác và cảm nhận về sản phẩm, trong khi UI tập trung vào các yếu tố thị giác như màu sắc, kiểu chữ, và bố cục. Việc thiết kế UI đẹp mắt và thân thiện với người dùng, cùng với UX mượt mà và trực quan, sẽ giúp tăng cường sự hài lòng và giữ chân người dùng. Một giao diện front-end tốt không chỉ cần phải đẹp mà còn phải hiệu quả, dễ sử dụng và phản hồi nhanh, mang lại trải nghiệm tốt nhất cho người dùng.

# CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT

## Ngôn ngữ HTML

### *Giới thiệu*

HTML có tên đầy đủ là Hypertext Markup Language nghĩa là ngôn ngữ đánh dấu siêu văn bản. HTML thường được sử dụng để tạo và cấu trúc các phần trong trang web và ứng dụng, phân chia các đoạn văn, heading, link,…[1].

HTML không phải là một ngôn ngữ lập trình mà chỉ là một ngôn ngữ đánh dấu. Điều này đồng nghĩa với việc HTML không thể thực hiện các chức năng “động”. Nói cách khác, HTML tương tự như phần mềm Microsoft Word, chỉ có tác dụng định dạng các thành phần có trong website.

### *Cấu trúc của HTML*

Một file code HTML được cấu thành bởi các phần tử HTML và các cặp thẻ. Song song đó, HTML gồm có nhiều dạng thẻ khác nhau và mỗi thẻ sẽ có nhiệm vụ và ý nghĩa riêng. Ngoài ra, mỗi thẻ sẽ được bắt đầu và kết thúc bằng dấu ngoặc nhọn “<, >”. Các chữ giữa các dấu ngoặc này gọi là phần tử.  Về cơ bản, cấu trúc của một trang HTML sẽ gồm 3 phần như sau:

-Phần khai báo loại file code có cấu trúc thẻ là <!DOCTYPE html>: Xuất hiện ở đầu hoặc trên cùng của file HTML. Qua phần này, người dùng sẽ biết được trình duyệt đang sử dụng để tạo trang là phiên bản HTML nào.

-Phần khai báo ban đầu, khai báo về meta, little, javascript, css,... có cấu trúc bắt đầu bằng thẻ <head> và kết thúc với thẻ <head>: Chứa tiêu đề và các khai báo có thông tin nhằm phục vụ SEO. Trong đó, tiêu đề được hiển thị trên thanh điều hướng của trang web và là phần nội dung nằm giữa cặp thẻ <title> và </title>.

-Phần chứa và hiển thị nội dung của trang web, có cấu trúc bắt đầu bằng thẻ <body> và kết thúc bằng thẻ </body>: Bao gồm tiêu đề trang web, logo, điều hướng chính và thanh tìm kiếm; nội dung chính như tiêu đề hoặc tiêu đề bài viết, nội dung bài viết, ngày đăng, tác giả,..; thanh bên hiển thị tiện ích con và điều hướng thứ cấp; chân trang cung cấp thông tin liên hệ, liên kết xã hội, bản quyền và điều hướng

### *Lịch sử của HTML*

Vào cuối năm 1991, phiên bản HTML đầu tiên do Tim Berners-Lee phát triển đã được công khai với tên HTML Tags. Phiên bản này có thiết kế vô cùng đơn giản, mô tả 18 phần tử. Tiếp đến vào năm 1995, IETF đã hoàn thành  "HTML 2.0".

Sau đó phiên bản HTML 4.01 được công bố vào năm 1999. Đến năm 2000, các phiên bản HTML đã được các nhà phát triển thay thế bằng XHTML. Năm 2014, HTML được nâng cấp lên chuẩn HTML5  với sự cải tiến rõ rệt. Điều này được thể hiện trong việc đã có nhiều tag được thêm vào markup để giúp xác định rõ nội dung thuộc thể loại gì.

### *2.1.4 Các loại HTML hiện nay*

Hiện nay, có 3 loại HTML được ứng dụng nhiều nhất đó là HTML 4, XHTML, HTML5. Sau đây là một vài đặc điểm của các loại HTML này:

HTML 4 ra đời vào năm 1997 là phiên bản thứ 4 của ngôn ngữ đánh dấu siêu văn bản. Phiên bản HTML này được xuất bản dưới dạng như một W3C Recommendation. HTML 4 áp dụng cho nhiều phần tử và thuộc tính khác nhau cho trình duyệt web.

XHTML có tên đầy đủ là Extensible HyperText Markup Language, nghĩa là ngôn ngữ đánh dấu siêu văn bản mở rộng. Đây là một ngôn ngữ thay thế của HTML với cú pháp chặt chẽ hơn. Cụ thể, XHTML yêu cầu mọi phần tử được đóng bằng thẻ đóng hoặc cú pháp tự đóng riêng và phân biệt được chữ in hoa hoặc chữ in thường, trong khi đó HTML không có điều này.

HTML5 phiên bản thứ 5 của HTML được công bố bởi World Wide Web Consortium (W3C). HTML5 là sự kết hợp giữa HTML4, XHTML, DOM cấp 2 và JavaScript. HTML5 hỗ trợ chạy trên mọi trình duyệt.

## Ngôn ngữ CSS

### *Giới thiệu*

CSS là viết tắt của Cascading Style Sheets, là một [ngôn ngữ](https://glints.com/vn/blog/ban-can-hoc-bao-nhieu-ngon-ngu-lap-trinh-de-tro-thanh-developer/)thiết kế được sử dụng nhằm mục đích đơn giản hóa quá trình tạo nên một website. CSS được ra mắt vào năm 1996 bởi World Wide Web Consortium (W3C)[2].

CSS xử lý một phần giao diện của trang web. Sử dụng CSS, có thể kiểm soát màu sắc của văn bản, kiểu phông chữ, khoảng cách giữa các đoạn văn, cách các cột được đặt kích thước và bố cục, hình ảnh hoặc màu nền nào được sử dụng, thiết kế bố cục, các biến thể hiển thị cho các thiết bị và kích thước màn hình khác nhau cũng như hàng loạt các hiệu ứng khác.

Cách CSS hoạt động đó chính là tìm kiếm dựa trên vùng chọn chẳng hạn như thẻ HTML, ID, class, v.v. Sau đó, nó sẽ áp dụng những thuộc tính buộc phải thay đổi lên các vùng đã chọn.

CSS rất dễ học và dễ hiểu nhưng nó cung cấp khả năng kiểm soát mạnh mẽ việc trình bày tài liệu HTML. Thông thường nhất, CSS được kết hợp với các ngôn ngữ đánh dấu HTML hoặc XHTML.

HTML và CSS có mối quan hệ gắn bó mật thiết với nhau. Nếu HTML là nền tảng của một trang web thì CSS là tất cả tính thẩm mỹ của toàn bộ trang web đó.

### *Bố cục và cấu trúc của một đoạn CSS*

- Bố cục của một đoạn CSS chủ yếu dựa vào hình hộp với mỗi hộp chiếm những khoảng trống trên trang web với các thuộc tính chính như:

+Padding: Là các không gian xung quanh nội dung (ví dụ: không gian xung quanh đoạn văn bản).

+Border: Là các đường nằm ngoài phần đệm.

+Margin: Là khoảng cách bao quanh phía ngoài của phần tử.

-Cấu trúc của một đoạn CSS: Thông thường, một đoạn CSS sẽ bao gồm các phần:

|  |
| --- |
| vùng chọn {thuộc tính: giá trị; thuộc tính: giá trị;….. } |

Đoạn CSS sẽ được khai báo bằng vùng chọn, các thuộc tính, giá trị nằm trong dấu ngoặc nhọn. Mỗi thuộc tính là một giá trị riêng ở dạng số, hoặc chính là tên của các giá trị đã có trong danh sách của CSS.

Quy tắc khai báo đó chính là: thuộc tính và giá trị cần cách nhau bằng dấu hai chấm, mỗi dòng khai báo thuộc tính cần có dấu chấm phẩy cuối cùng. Các thuộc tính không bị giới hạn ở một vùng chọn.

Trong đó:

Bộ chọn (Selector): mẫu để chọn phần tử HTML mà muốn định nghĩa phong cách. có thể áp dụng các selector cho các trường hợp sau:

Tất cả những phần tử được định dạng theo một dạng cụ thể nào đó, ví dụ phần tử tiêu đề h2.

Thuộc tính id, class của phần tử.

Các phần tử có mối liên quan với các phần tử khác trong hệ thống cây phân cấp tài liệu.

Khai báo (Declaration): Khối khai báo có thể chứa một hoặc nhiều khai báo và chúng được phân tách với nhau bằng dấu chấm phẩy. Mỗi khai báo lại bao gồm tên & giá trị đặc tính CSS, dược phân tách với nhau bằng dấu phẩy. Quy tắc khai báo CSS là chúng luôn phải kết thúc bằng dấu chấm phẩy, và khối khai báo phải nằm trong các dấu ngoặc móc.

Thuộc tính (Properties): Thuộc tính là các cách thức mà có thể tạo kiểu cho một phần tử HTML. Vì vậy, với CSS, chỉ cần lựa chọn thuộc tính mà muốn tác động nhất trong bộ quy tắc đã tạo ra.

Giá trị thuộc tính: Được nằm ở bên phải của thuộc tính. Việc lựa chọn một thuộc tính trong số đó phụ thuộc vào số lần xuất hiện của thuộc tính.

### *Cách nhúng CSS vào website*

Có ba cách nhúng CSS vào website:

Nhúng CSS trực tiếp CSS vào tài liệu HTML (Inline CSS), với cách này, chúng ta đặt mã CSS vào thẳng thuộc tính style của phần tử. Và với cách nhúng trực tiếp, mã CSS sẽ chỉ tác động lên chính phần tử đó.

Để chỉ định nhiều quy tắc CSS, chúng ta có thể sử dụng dấu chấm phẩy để ngăn cách giữa các rules. Các rules này sẽ được đặt bằng cách sử dụng thuộc tính “style” kèm theo tên thuộc tính, giá trị thuộc tính.

Nội tuyến (Internal CSS), với cách nhúng nội tuyến, cần dùng thẻ <style> để tạo ra khu vực viết CSS. Có thể để CSS ở bất kỳ đâu trong HTML, nhưng nên đặt <style> trong thẻ <head>.

Ngoại tuyến (External CSS), trong External CSS, chúng ta sẽ sử dụng phần tử “link” để thêm các style sheet ở bên ngoài vào trong tài liệu HTML.

Trước hết chúng ta cần tạo các rules (quy tắc) trong một file riêng có phần đuôi mở rộng là .css. Tiếp theo, cần thêm file CSS này vào phần tử head trong tài liệu HTML.

Đây là phương pháp phổ biến nhất để nhúng CSS vào tài liệu HTML. Với cách chèn này, các lập trình viên có thể viết CSS cho nhiều trang web khác nhau và bổ sung cùng một file CSS cho tất cả các trang tương tự.

## Ngôn ngữ JavaScript

Javascript chính là một ngôn ngữ lập trình web rất phổ biến ngày nay. Javascript được tích hợp đồng thời nhúng vào HTML để hỗ trợ cho website trở nên sống động hơn. Chúng cũng đóng vai trò tương tự như một phần của website, cho phép Client-side Script từ người dùng tương tự máy chủ (Nodejs) để tạo ra những website động.[3]

### *Lịch sử phát triển*

Brendan Eich chính là người đã phát triển Javascript tại Netscape với tiền thân là Mocha. Sau đó, Mocha được đổi thành LiveScript và cuối cùng mới đổi thành JavaScript.

**Năm 1998**, JavaScript với phiên bản mới nhất là ECMAScript 2 phát hành và đến năm 1999 thì ECMAScript 3 được ra mắt.

**Năm 2016,** ứng dụng JavaScript đã đạt kỷ lục lên tới 92% website sử dụng, đồng thời cũng được đánh giá là một công cụ cực kỳ quan trọng đối với lập trình viên.

### *Cách hoạt động của ngôn ngữ JavaScript*

Thông thường, **JavaScript**sẽ được nhúng trực tiếp vào một website hoặc chúng được tham chiếu qua file .js hoặc .JavaScript.

Đây là một ngôn ngữ đến từ phía Client nên Script sẽ được download về máy client khi truy cập.

Tại đây, chúng sẽ được hệ thống xử ý. Vì vậy, không cần phải tải về máy server rồi chờ cho chúng xử lý xong mới phản hồi được kết quả đến client.

### *Ứng dụng*

**Thay đổi nội dung HTML:** một trong số nhiều phương thức HTML JavaScript chính là getElementById (). Chúng được sử dụng để tìm một phần tử của HTML với id =”demo" và dùng để thay đổi nội dung của phần từ (Internal HTML) sang thành “Hello JavaScript”.

**Thay đổi giá trị thuộc tính HTML:** Tổng quan về javascript còn có thể sử dụng để thay đổi các giá trị của thuộc tính. Ví dụ: thay đổi thuộc tính src (source) của tag<img>.

**Thay đổi kiểu HTML:**Đây chính là một hoạt động biến thể của việc thay đổi thuộc tính của HTML ở trên. Ví dụ: document.getElementById(‘demo’).style.fontSize = ’35px;

**Ẩn các phần tử HTML:**Một hoạt động tiếp theo là Javascript có thể ẩn được các phần tử HTML. Chúng có thể được thực hiện thông qua hoạt động thay đổi kiểu hiển thị các phần tử HTML.

**Hiển thị các phần tử HTML:**Một điểm đặc biệt là JavaScript có thể hiển thị được các yếu tố HTML ẩn. Đồng thời, cũng có thể thực hiện được thông qua cách thay đổi kiểu hiển thị phần tử.

Có 3 cách đặt thẻ script thường được sử dụng như dưới đây:

Với Internal, thông thường thẻ Script sẽ được đặt ở trong thẻ <head>, tuy nhiên cũng có thể đặt ở bất kỳ chỗ nào. Yêu cầu duy nhất là phải có chứa đầy đủ mở và đóng thẻ <script></script>. Bằng cách này, có thể đặt phần head ở trong file HTML hiện có của mình.

Đối với External, sẽ viết được các đoạn mã Javascript ra trong một file mới. Sau đó, lưu phần mở rộng này với đuôi .js. Kế đến, bán ẽ sử dụng thẻ Script để import file vào. Không cần đến thẻ đóng </script> vì trong file test.js đã có sự hiện diện của đuôi js. Vì vậy, trình duyệt sẽ tự động nhận dạng được đây là một file có chứa những mã Javascript.

Đối với Inline, cách sẽ sẽ cho phép viết trực tiếp các đoạn mã JS vào bên trong thẻ HTML một cách nhanh chóng.

## API

### *Giới thiệu*

API là các phương thức, giao thức kết nối với các thư viện và ứng dụng khác. Nó là viết tắt của Application Programming Interface – giao diện lập trình ứng dụng. API cung cấp khả năng truy xuất đến một tập các hàm hay dùng. Và từ đó có thể trao đổi dữ liệu giữa các ứng dụng[5].

### *Ứng dụng*

Web API: là hệ thống API được sử dụng trong các hệ thống website. Hầu hết các website đều ứng dụng đến Web API cho phép kết nối, lấy dữ liệu hoặc cập nhật cơ sở dữ liệu. Ví dụ: thiết kế chức năng login thông qua Google, Facebook, Twitter, Github… Điều này có nghĩa là đang gọi đến API của. Hoặc như các ứng dụng di động đều lấy dữ liệu thông qua API.

API trên hệ điều hành: Windows hay Linux có rất nhiều API, họ cung cấp các tài liệu API là đặc tả các hàm, phương thức cũng như các giao thức kết nối. Nó giúp lập trình viên có thể tạo ra các phần mềm ứng dụng có thể tương tác trực tiếp với hệ điều hành.

API của thư viện phần mềm hay framework: API mô tả và quy định các hành động mong muốn mà các thư viện cung cấp. Một API có thể có nhiều cách triển khai khác nhau và nó cũng giúp cho một chương trình viết bằng ngôn ngữ này có thể sử dụng thư viện được viết bằng ngôn ngữ khác. Ví dụ có thể dùng Php để yêu cầu một thư viện tạo file PDF được viết bằng C++.

### *Web API*

Web API là một phương thức dùng để cho phép các ứng dụng khác nhau có thể giao tiếp, trao đổi dữ liệu qua lại. Dữ liệu được Web API trả lại thường ở dạng [JSON](https://topdev.vn/blog/json-la-gi/) hoặc XML thông qua giao thức HTTP hoặc HTTPS.

Web API hỗ trợ restful đầy đủ các phương thức: Get/Post/put/delete dữ liệu. Nó giúp xây dựng các HTTP service một cách rất đơn giản và nhanh chóng. Nó cũng có khả năng hỗ trợ đầy đủ các thành phần HTTP: URI, request/response headers, caching, versioning, content format.

Với web API, chúng ta sẽ tự động hóa quản lý công việc, cập nhật luồng công việc, giúp tăng năng suất và tạo hiệu quả công việc cao hơn.

API cho phép lấy nội dung từ bất kỳ website hoặc ứng dụng nào một cách dễ dàng nếu được cho phép, tăng trải nghiệm người dùng. API hoạt động như một chiếc cổng, cho phép các công ty chia sẻ thông tin được chọn nhưng vẫn tránh được những yêu cầu không mong muốn.

API có chức năng thay đổi và cập nhật thay đổi theo thời gian thực. Với công nghệ này, dữ liệu sẽ được truyền đi tốt hơn, thông tin chính xác hơn, dịch vụ cung cấp linh hoạt hơn.

Bất kỳ người dùng, công ty nào sử dụng cũng có thể điều chỉnh nội dung, dịch vụ mà họ sử dụng.

Hỗ trợ đầy đủ các thành phần MVC như: routing, controller, action result, filter, model binder, IoC container, [dependency injection](https://topdev.vn/blog/dependency-injection-la-gi/), unit test.

### *Cách Web API hoạt động*

Đầu tiên là xây dựng URL API để bên thứ ba có thể gửi request dữ liệu đến máy chủ cung cấp nội dung, dịch vụ thông qua giao thức HTTP hoặc HTTPS.

Tại web server cung cấp nội dung, các ứng dụng nguồn sẽ thực hiện kiểm tra xác thực nếu có và tìm đến tài nguyên thích hợp để tạo nội dung trả về kết quả.

Server trả về kết quả theo định dạng JSON hoặc XML thông qua giao thức HTTP/HTTPS.

Tại nơi yêu cầu ban đầu là ứng dụng web hoặc ứng dụng di động , dữ liệu JSON/XML sẽ được parse để lấy data. Sau khi có được data thì thực hiện tiếp các hoạt động như lưu dữ liệu xuống Cơ sở dữ liệu, hiển thị dữ liệu…

## JSON API

### *Giới thiệu JSON API*

JSON API là một tiêu chuẩn để xây dựng các API bằng cách sử dụng định dạng JSON. Mục tiêu của JSON API là làm cho các API trở nên đơn giản hơn và nhất quán hơn bằng cách tuân theo các quy ước chặt chẽ về cách yêu cầu và phản hồi dữ liệu.[6]

Định dạng JSON: JSON (JavaScript Object Notation) là một định dạng dữ liệu nhẹ, dễ đọc và viết. Nó sử dụng các cặp key-value để biểu diễn dữ liệu.

Phương thức HTTP: JSON API sử dụng các phương thức HTTP tiêu chuẩn như GET, POST, PUT, PATCH, và DELETE để thực hiện các thao tác CRUD (Create, Read, Update, Delete).

Quy ước rõ ràng: JSON API xác định rõ ràng các quy ước về cách đặt tên đường dẫn, định dạng yêu cầu và phản hồi, cũng như các mã trạng thái HTTP.

Ví dụ về yêu cầu và phản hồi JSON API:

|  |
| --- |
| {  "data": {  "type": "articles",  "id": "1",  "attributes": {  "title": "JSON API là gì?",  "content": "Đây là nội dung của bài viết."  },  "relationships": {  "author": {  "data": { "type": "people", "id": "9" }  }  }  }  } |

JSON API giúp chuẩn hóa cách thức giao tiếp giữa client và server, giảm thiểu sự khác biệt và tăng tính tương thích, giúp việc phát triển ứng dụng web trở nên dễ dàng và hiệu quả hơn.

### *Cấu trúc JSON API*

JSON (JavaScript Object Notation) là một định dạng dữ liệu nhẹ, dễ đọc và ghi. Nó được sử dụng chủ yếu để trao đổi dữ liệu giữa máy chủ và ứng dụng web. JSON có cấu trúc dựa trên hai yếu tố chính: đối tượng (object) và mảng (array).

Cấu trúc của JSON

-Đối tượng (Object): Được biểu diễn bằng dấu ngoặc nhọn ({}). Một đối tượng chứa các cặp tên/giá trị, với tên và giá trị được phân cách bởi dấu hai chấm (:), và các cặp tên/giá trị được phân tách bằng dấu phẩy ,.

-Mảng (Array): Được biểu diễn bằng dấu ngoặc vuông ([]). Một mảng chứa các giá trị, và các giá trị này được phân tách bởi dấu phẩy ,.

Cú pháp của JSON

-Keys: Phải là chuỗi và nằm trong dấu ngoặc kép "".

-Values**:** Có thể là chuỗi, số, đối tượng, mảng, boolean hoặc null.

### *Hướng dẫn sử dụng JSON API*

JSON API là một tiêu chuẩn để xây dựng các API RESTful, giúp tối ưu hóa việc trao đổi dữ liệu giữa client và server. Dưới đây là hướng dẫn từng bước để sử dụng JSON API.

**Cài đặt môi trường**

Đầu tiên, cần cài đặt Node.js và npm (Node Package Manager). Sau khi cài đặt, kiểm tra phiên bản bằng lệnh: node -v và npm -v

**Cài đặt JSON Server**

Cài đặt JSON Server toàn cục bằng npm:

|  |
| --- |
| npm install -g json-server |

**Tạo file JSON**

Tạo một file data.json chứa dữ liệu muốn phục vụ qua API. Ví dụ:

|  |
| --- |
| {  "employees": [  { "id": 1, "name": "John Doe", "role": "Developer" },  { "id": 2, "name": "Jane Doe", "role": "Designer" }  ]  } |

**Khởi động JSON Server**

Chạy lệnh sau để khởi động server:

|  |
| --- |
| json-server --watch data.json |

Server sẽ chạy tại http://localhost:3000.

**Sử dụng các endpoints**

Các endpoints mặc định được tạo bởi JSON Server bao gồm:GET /employees, GET /employees/{id}, POST /employees, PUT /employees/{id}, DELETE /employees/{id}.

**Tùy chỉnh và mở rộng**

Có thể thêm các tham số vào URL để lọc dữ liệu, ví dụ:

<http://localhost:3000/employees?name=John>.

## Bootstrap

### *Giới thiệu*

Bootstrap là một framework front-end mã nguồn mở được phát triển bởi Twitter cung cấp một bộ công cụ và tài nguyên giúp thiết kế giao diện web nhanh chóng và dễ dàng. Bootstrap sử dụng HTML, CSS, và JavaScript để xây dựng các thành phần và giao diện người dùng đáp ứng, giúp đảm bảo rằng trang web của sẽ hiển thị một cách tốt trên nhiều thiết bị và kích thước màn hình khác nhau[4].

### *Lịch sử Bootstrap*

Ngày 19 tháng 8 năm 2011: Bootstrap được giới thiệu lần đầu tiên bởi Twitter với tên gọi Twitter Blueprint.

Ngày 24 tháng 1 năm 2012: Sau một thời gian sử dụng nội bộ tại Twitter, Bootstrap được chính thức công bố là một dự án mã nguồn mở.

Bootstrap 2 (Ngày 31 tháng 1 năm 2012): Bootstrap 2 được phát hành với nhiều cải tiến so với phiên bản trước đó, bao gồm cải thiện hiệu suất, bảng điều khiển (dashboard), và nhiều thành phần mới.

Bootstrap 3 (Ngày 19 tháng 8 năm 2013): Bootstrap 3 mang lại sự linh hoạt và tính đáp ứng cao hơn. Nó chuyển sang sử dụng Flat Design và loại bỏ hỗ trợ cho trình duyệt Internet Explorer 7 và 8.

Bootstrap 4 (Ngày 19 tháng 1 năm 2018): Bootstrap 4 là một bản cập nhật lớn với nhiều thay đổi đáng kể, bao gồm việc chuyển sang Flexbox, thay đổi đáng kể trong hệ thống lưới, và một loạt các cải tiến khác về hiệu suất và tính năng.

Bootstrap 5 (Ngày 5 tháng 5 năm 2021): Bootstrap 5 tiếp tục mở rộng tính năng và cải thiện hiệu suất. Một số thành phố như jQuery đã được loại bỏ và thay thế bằng Vanilla JavaScript. Bootstrap 5 cũng chú trọng vào tích hợp tốt hơn với CSS custom properties.

Trong quá trình phát triển, Bootstrap đã trở thành một trong những framework front-end phổ biến nhất và được sử dụng rộng rãi trên toàn cầu. Cộng đồng với đông đảo người dùng giúp framework ngày hoàn thiện và đáp ứng nhu cầu ngày càng đa dạng của các nhà phát triển web.

### *Tính năng của Bootstrap*

Framework này gồm có 3 file chính. Các file trong Bootstrap đã được nén lại, giúp người dùng dễ dàng tải, sử dụng và lưu trữ.

Ngoài ra, công cụ này cũng được thiết kế dưới dạng Module nên có thể dễ dàng tích hợp với nhiều loại mã nguồn mở, nổi tiếng, quen thuộc như WordPress, Joomla, Magento…. Bootstrap cũng có rất nhiều tính năng nổi bật khác như:

Cho phép người dùng truy cập vào thư viện: Thư viện của Bootstrap rất đa dạng với nhiều thành phần để người dùng dễ dàng thiết kế, tạo thành giao diện website. Người dùng có thể dễ dàng sử dụng font, typography, form, table, grid… để thiết kế một website như mong muốn.

Cho phép tùy chỉnh Framework: Với Bootstrap có thể dễ dàng tùy chỉnh Framework. Sau đó tải Framework về về tùy chỉnh tại khung của website đang được xây dựng.

Tái sử dụng: cũng có thể lưu trữ và sử dụng các thành phần đã được thiết kế cho những website tiếp theo.

Tích hợp Jquery: Như đã giới thiệu ở trên, Bootstrap được tích hợp jQuery, giúp tiết kiệm rất nhiều thời gian thiết kế, code và lập trình trang web.

Giảm thiểu sử dụng hình ảnh làm biểu tượng, tăng tốc độ tải trang: Người dùng có thể sử dụng file Glyphicons, lựa chọn các biểu tượng, giúp tăng tính thẩm mỹ cho website và tăng tốc độ tải của trang web.

CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU

## Mô tả các bước nghiên cứu

### 3.1.1 *Xây dựng file JSON*

- Cài đặt JSON Server toàn cục bằng npm:

|  |
| --- |
| npm install -g json-server |

-Tạo file JSON tên products.json

|  |
| --- |
| {   "featured\_products":[{  "id": 1, "name": "tên sản phẩm", "description": "mô tả sản phẩm.", "price": giá sản phẩm, "image": "đường dẫn hình ảnh ","store\_id": id của cửa hàng},],    "categories":[{  "id": id loại sản phẩm ,"name": "loại sản phẩm"},],    "stores":[{  "id": id cửa hàng,  "image": "hình ảnh cửa hàng", "owner": "tên chủ gian hàng", "store\_name": "tên gian hàng", "description": "mô tả gian hàng", "products": [mảng các id sản phẩm có trong gian hàng]},],  "products":[{   "id": id sản phẩm, "name": "tên sản phẩm", "description": "mô tả sản phẩm.", "price": giá sản phẩm, "stock": số lượng sản phẩm, "image": "đường dẫn hình sản phẩm", "category\_id": phân loại sản phẩm theo id loại sản phẩm, "store\_id": id gian hàng },],    " store\_owners":[{  "id": id chủ gian hàng, "username": "tên chủ gian hàng", "password": "mật khẩu", "store\_id": id gian hàng},],  } |

Chạy lệnh sau để khởi động server:

|  |
| --- |
| json-server --watch products.json |

Server sẽ chạy tại:

http://localhost:3000/featured\_products

http://localhost:3000/categories

http://localhost:3000/stores

http://localhost:3000/products

http://localhost:3000/store\_owners

### *3.1.2 Các phương thức để hiển thị sản phẩm từ file JSON lên web*

#### 3.1.2.1 Lấy dữ liệu từ API JSON

Để hiển thị dữ liệu từ file JSON trên giao diện web, chúng ta cần sử dụng JavaScript để gửi yêu cầu đến API được giả lập bởi json-server và lấy dữ liệu. Phương thức phổ biến nhất là sử dụng hàm fetch.

|  |
| --- |
| fetch('http://localhost:3000/products')  .then(response => response.json())  .then(data => {  console.log(data); // Xem dữ liệu trong console  // Gọi hàm hiển thị sản phẩm  displayProducts(data);  })  .catch(error => console.error('Lỗi:', error)); |

#### 3.1.2.2 Hiển thị dữ liệu lên giao diện web

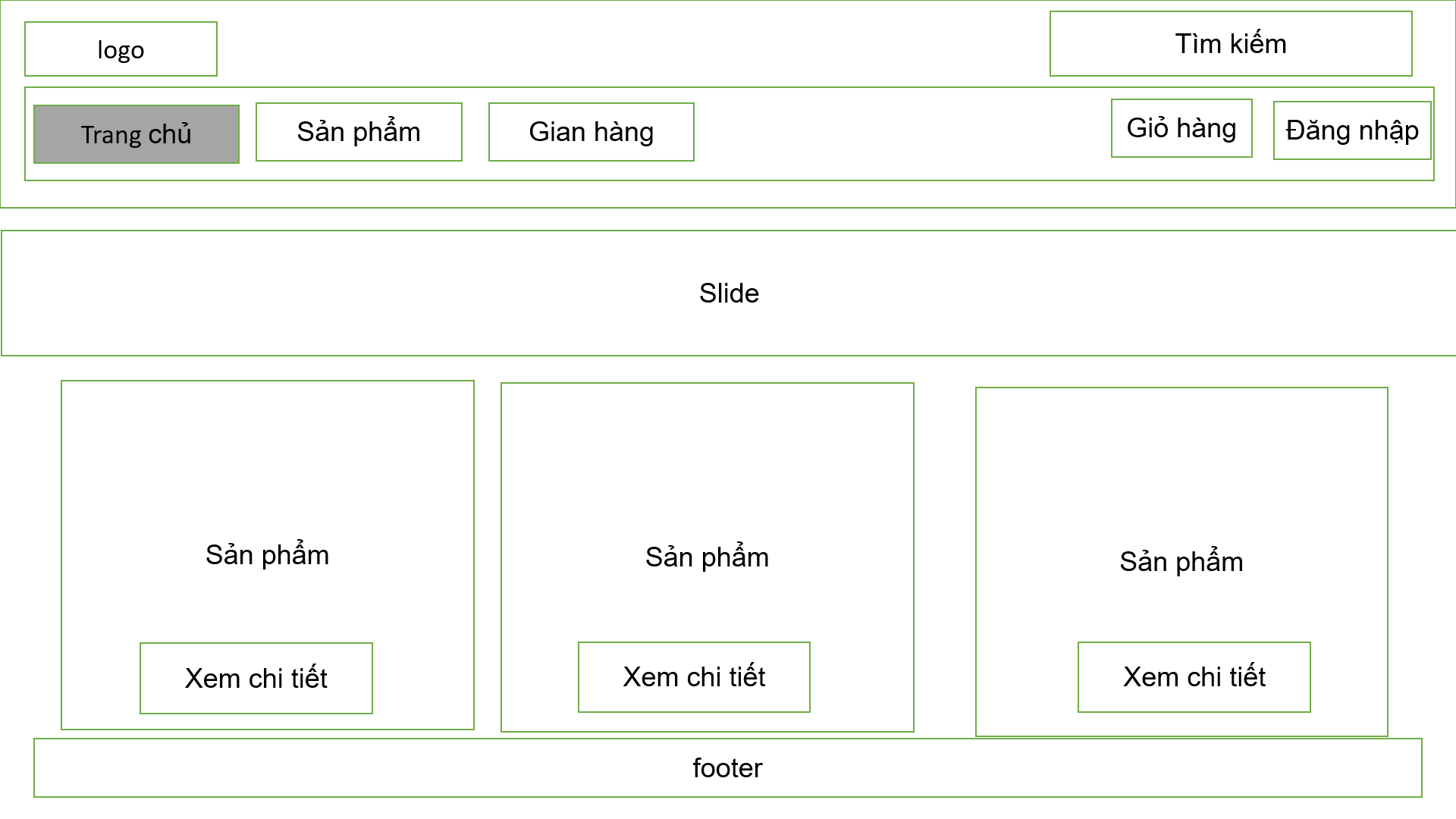
Sau khi lấy được dữ liệu, sử dụng JavaScript để tạo các phần tử HTML và chèn chúng vào DOM.

|  |
| --- |
| function displayProducts(products) {  const productContainer = document.getElementById('product-list');  products.forEach(product => {  const productElement = document.createElement('div');  productElement.classList.add('product');  productElement.innerHTML = `  <img src="${product.image}" alt="${product.name}" />  <h3>${product.name}</h3>  <p>${product.description}</p>  <span>Giá: ${product.price} VND</span>  <button>Thêm vào giỏ hàng</button>  `;  productContainer.appendChild(productElement);  });  } |

### *3.1.3 Các bản thiết kế*

#### 3.1.3.1 Giao diện trang chủ

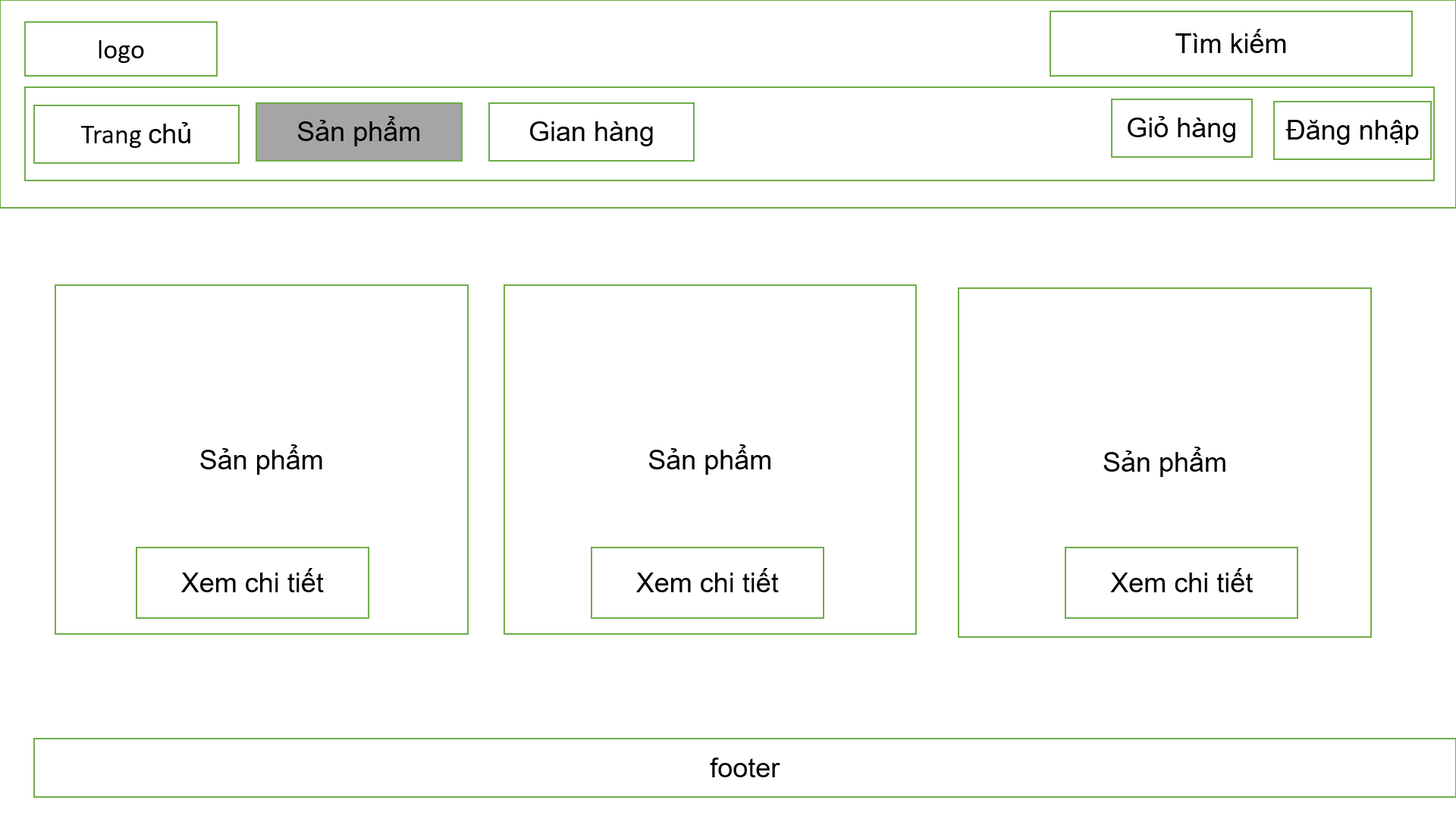
Trang chủ là nơi hiển thị các sản phẩm nổi bật, kèm theo thanh điều hướng đến các phần khác như sản phẩm, gian hàng, giỏ hàng, và đăng nhập.



**Hình 3.1** Giao diện bảng thiết kế trang chủ

#### 3.1.3.2 Giao diện trang sản phẩm

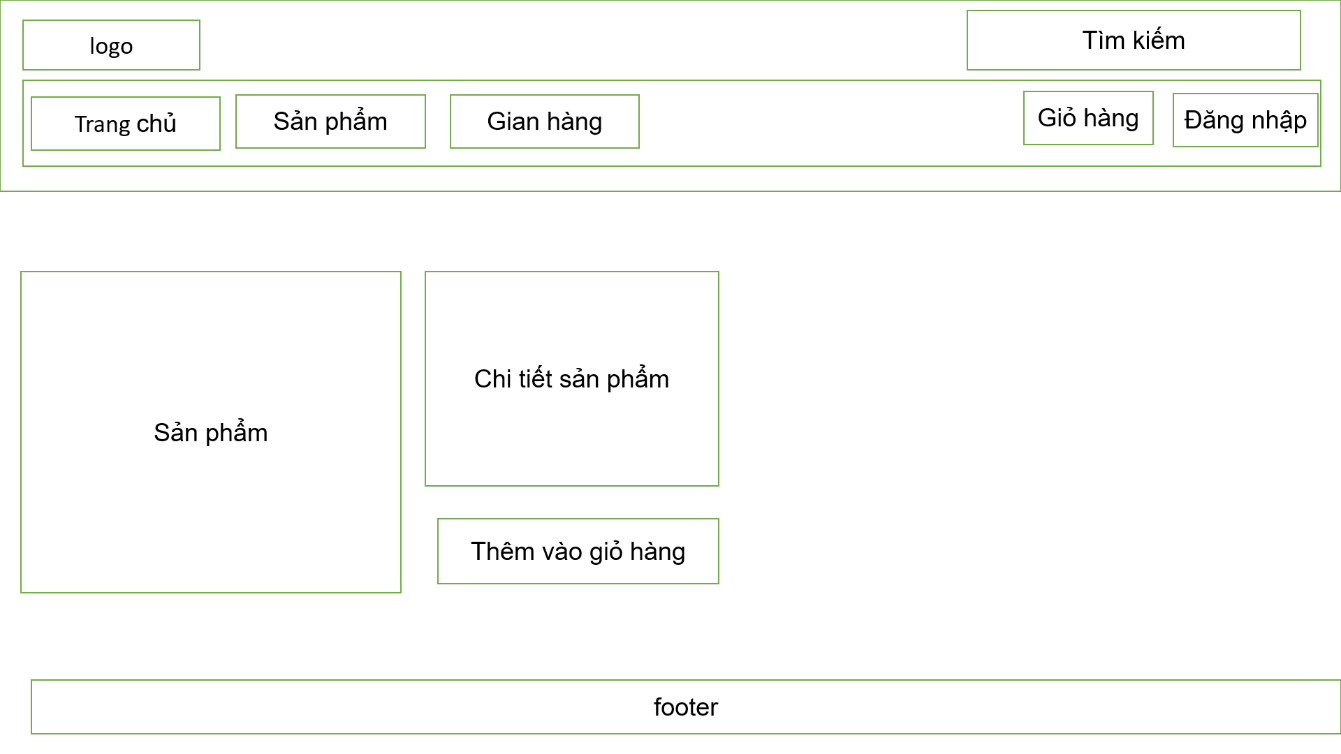
Hiển thị toàn bộ sản phẩm theo loại.



**Hình 3.2** Giao diện bảng thiết kế trang sản phẩm

#### 3.1.3.3 Giao diện chi tiết sản phẩm

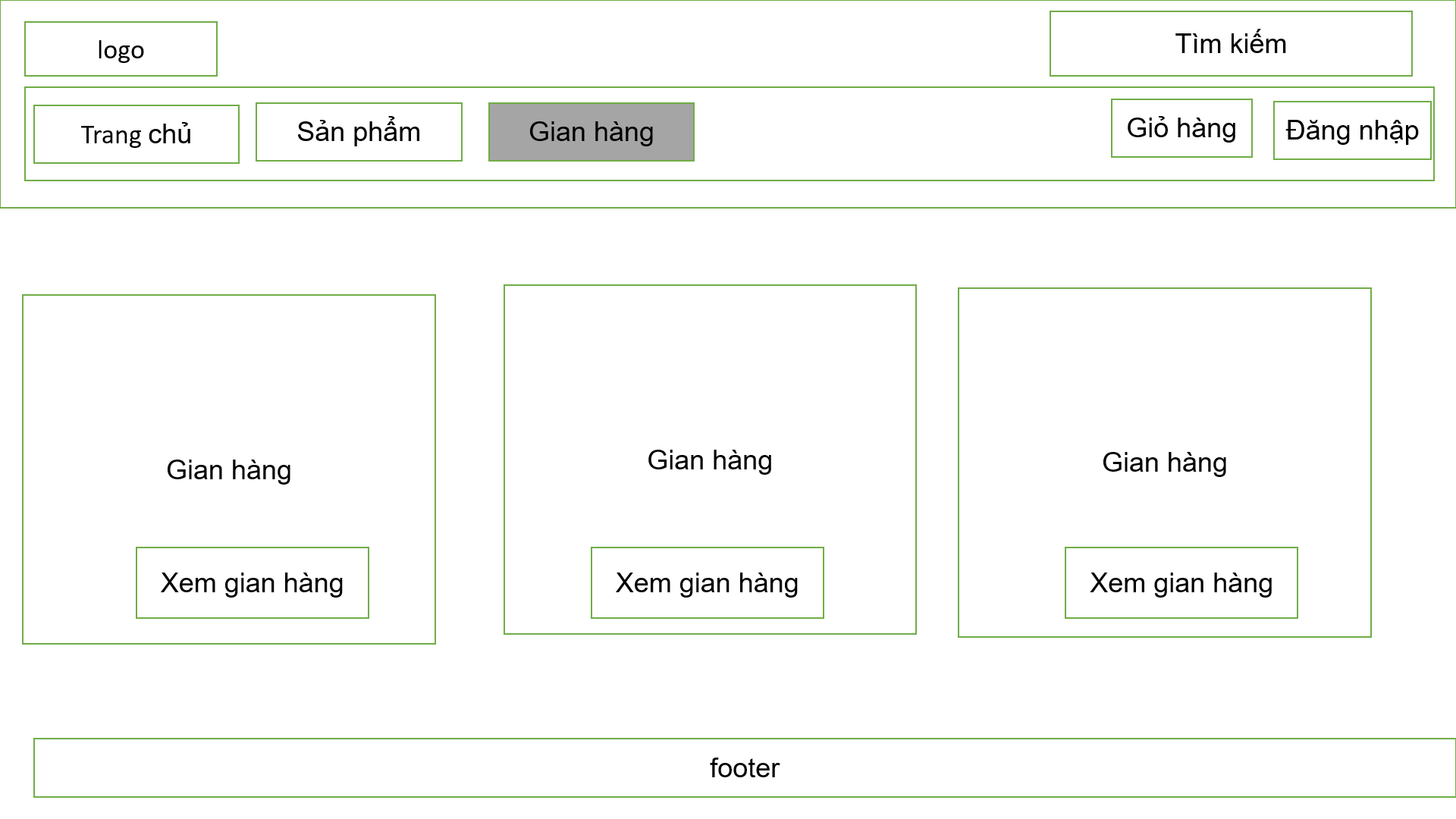
Hiển thị chi tiết về một sản phẩm, cho phép người dùng thêm sản phẩm vào giỏ hàng.



**Hình 3.3** Giao diện bảng thiết kế trang chi tiết sản phẩm

#### 3.1.3.4 Giao diện các gian hàng

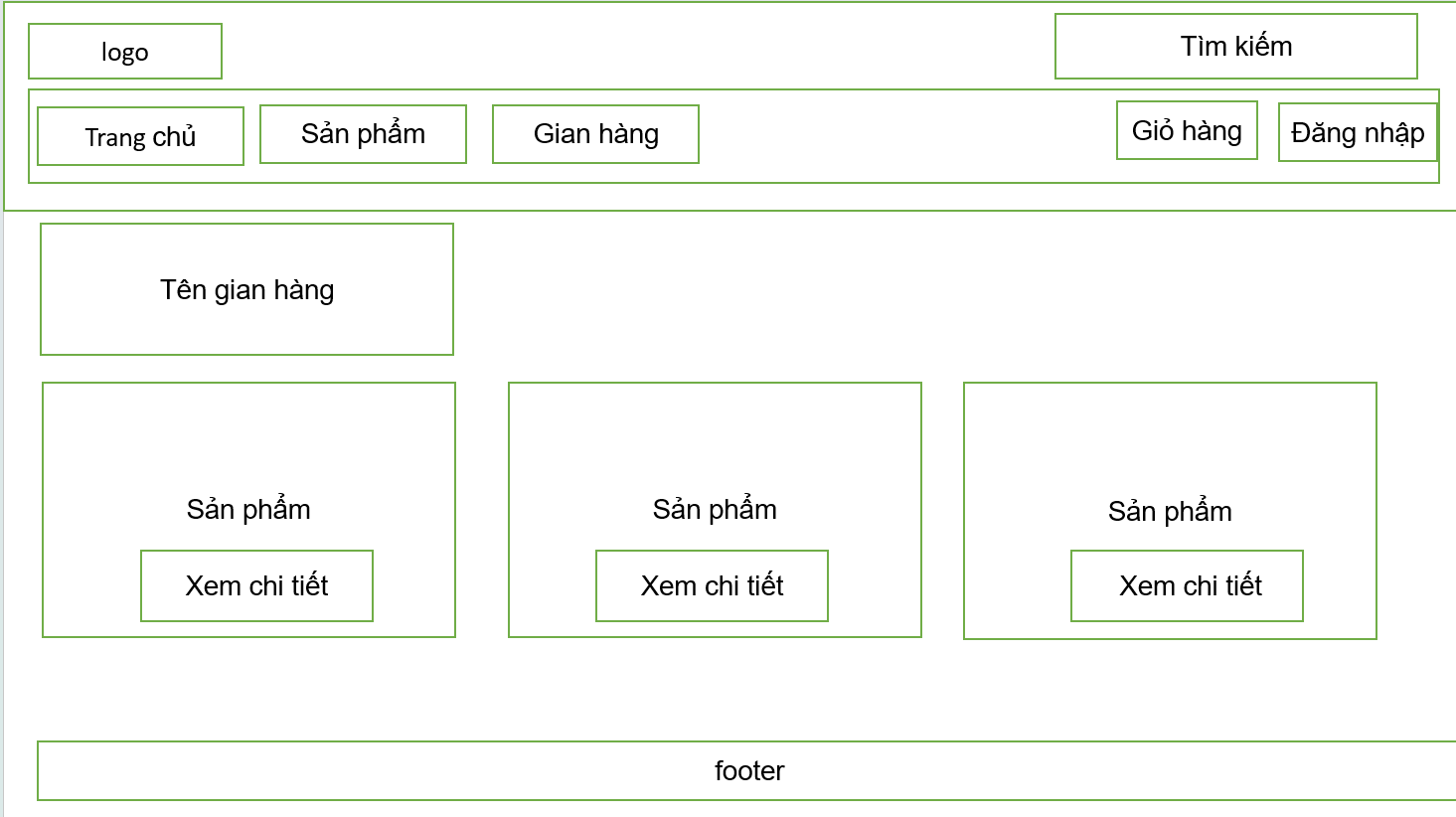
Hiển thị danh sách các gian hàng của sinh viên.



**Hình 3.4** Giao diện bảng thiết kế trang gian hàng

#### 3.1.3.5 Giao diện chi tiết các gian hàng

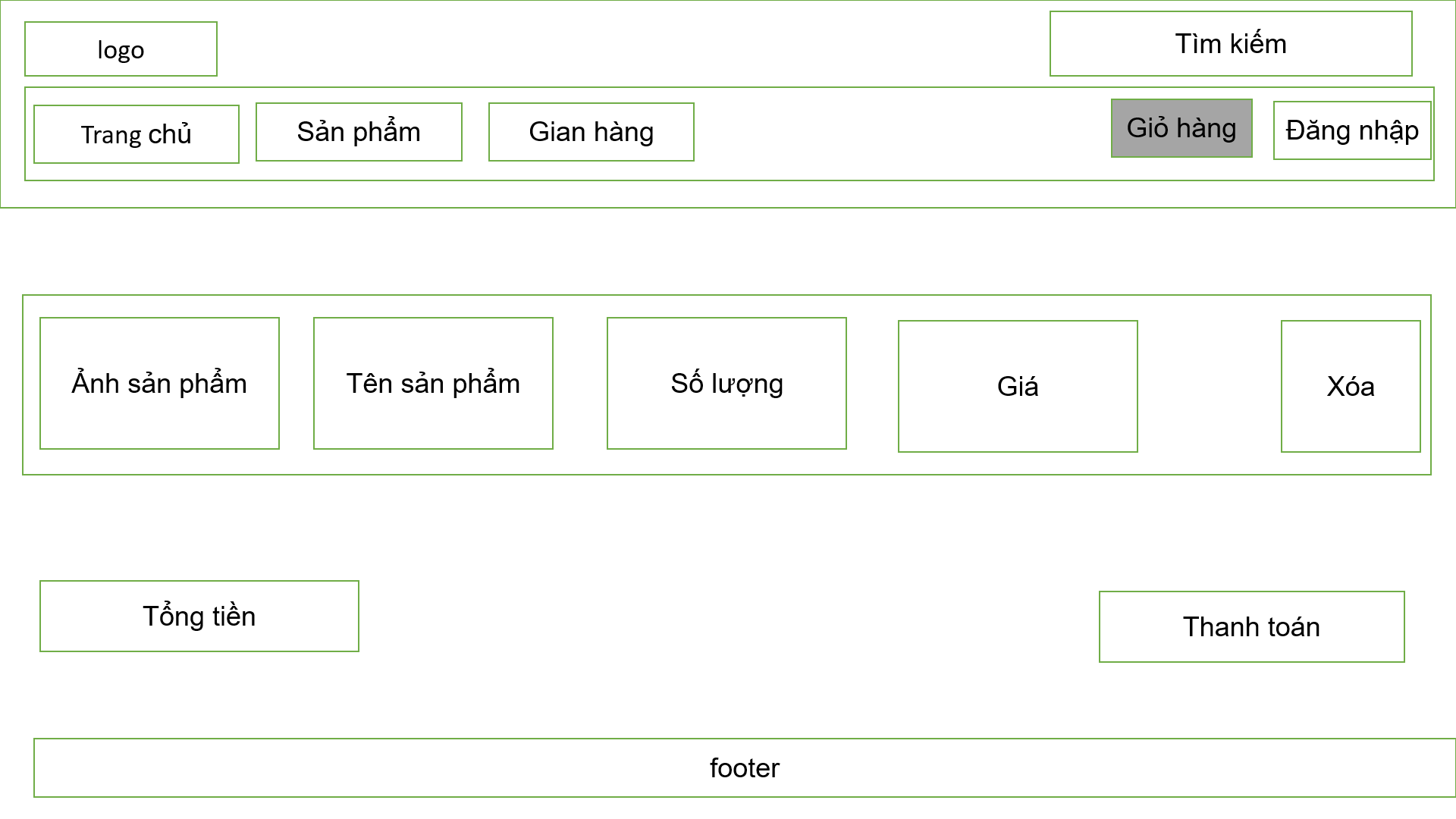
Hiển thị thông tin của một gian hàng và các sản phẩm mà gian hàng đó cung cấp.



**Hình 3.5** Giao diện bảng thiết kế trang chi tiết gian hàng

#### 3.1.3.6 Giao diện giỏ hàng

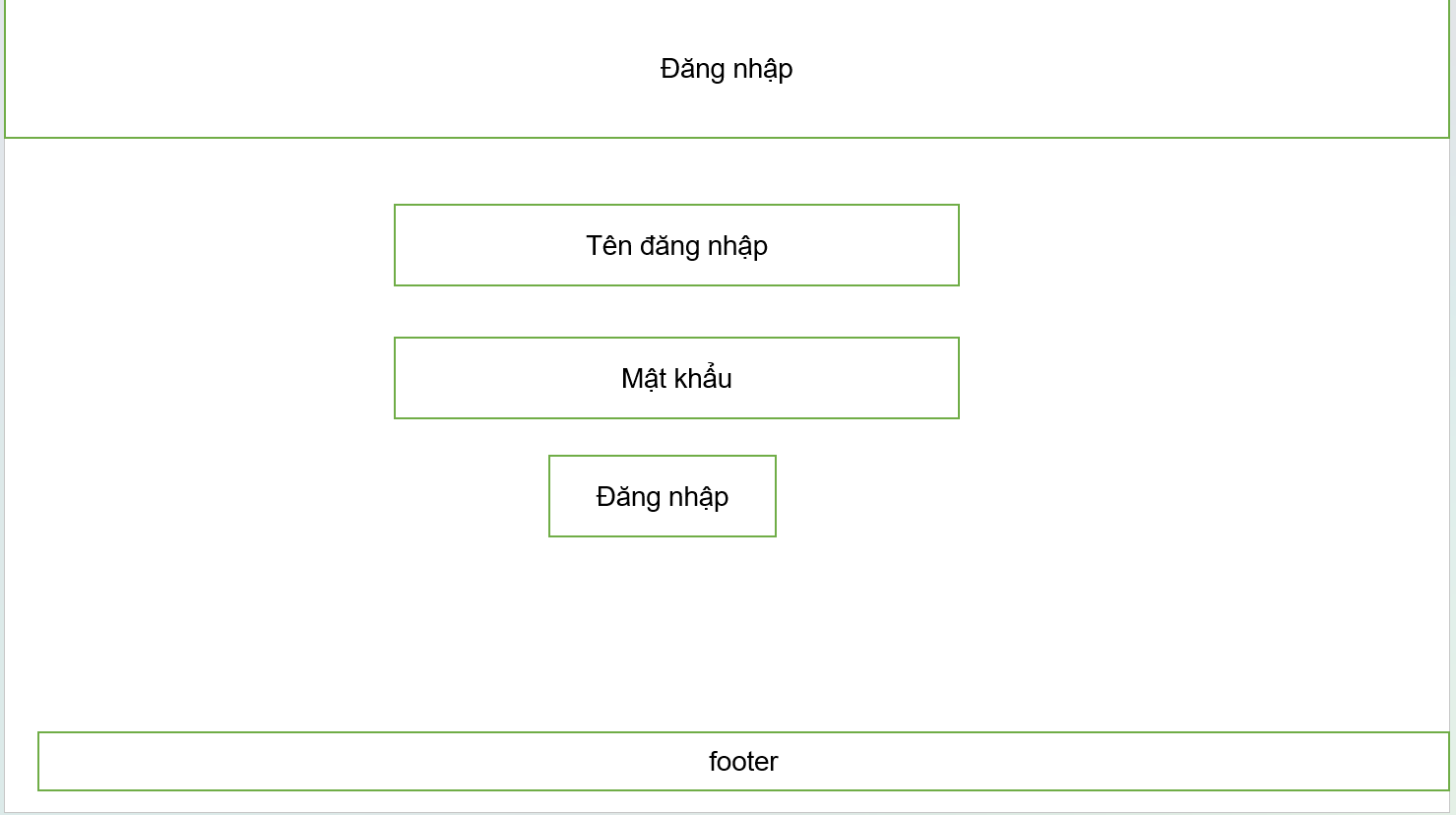
Hiển thị danh sách các sản phẩm mà người dùng đã thêm vào giỏ hàng và cho phép tính tổng tiền.



**Hình 3.6** Giao diện bảng thiết kế trang giỏ hàng

#### 3.1.3.7 Giao diện đăng nhập chủ gian hàng

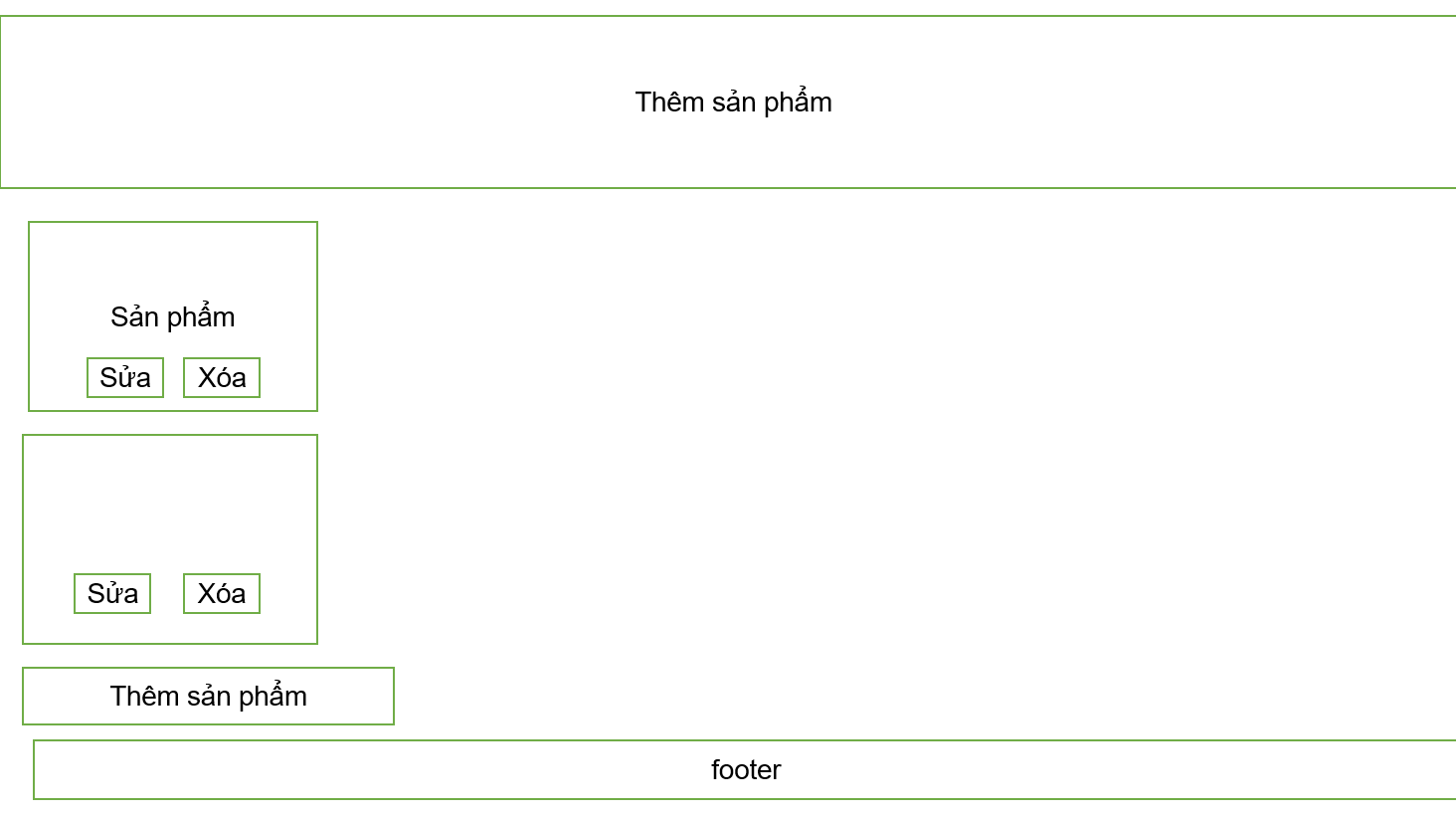
Trang để chủ gian hàng đăng nhập vào hệ thống.



**Hình 3.7** Giao diện bảng thiết kế trang đăng nhập chủ gian hàng

#### 3.1.3.8 Giao diện quản lý gian hàng

Trang dành cho chủ gian hàng quản lý sản phẩm của họ.



**Hình 3.8** Giao diện bảng thiết kế trang quản lý gian hàng

CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU

## 4.1 Kết quả giao diện các trang web

## *4.1.1 Trang chủ*

Kết quả: giao diện trang chủ hiển thị được danh sách các sản phẩm nổi bật (featured products) từ file JSON.

Chức năng: điều hướng đến các trang khác qua thanh menu, hiển thị thông tin cơ bản của sản phẩm nổi bật như tên, giá, và hình ảnh.



**Hình 4.1** Giao diện trang chủ web

## *4.1.2 Trang sản phẩm*

Kết quả: Trang danh sách sản phẩm hiển thị toàn bộ sản phẩm từ file JSON, sắp xếp theo dạng lưới.

Chức năng: hiển thị các danh mục sản phẩm.



**Hình 4.2** Giao diện trang sản phẩm

## *4.1.3 Trang chi tiết sản phẩm*

Kết quả: hiển thị thông tin chi tiết về sản phẩm được chọn, bao gồm: tên, mô tả, giá, hình ảnh, và số lượng còn lại.

Chức năng: kết nối với file JSON để hiển thị thông tin chính xác theo ID sản phẩm, tương tác với giỏ hàng.

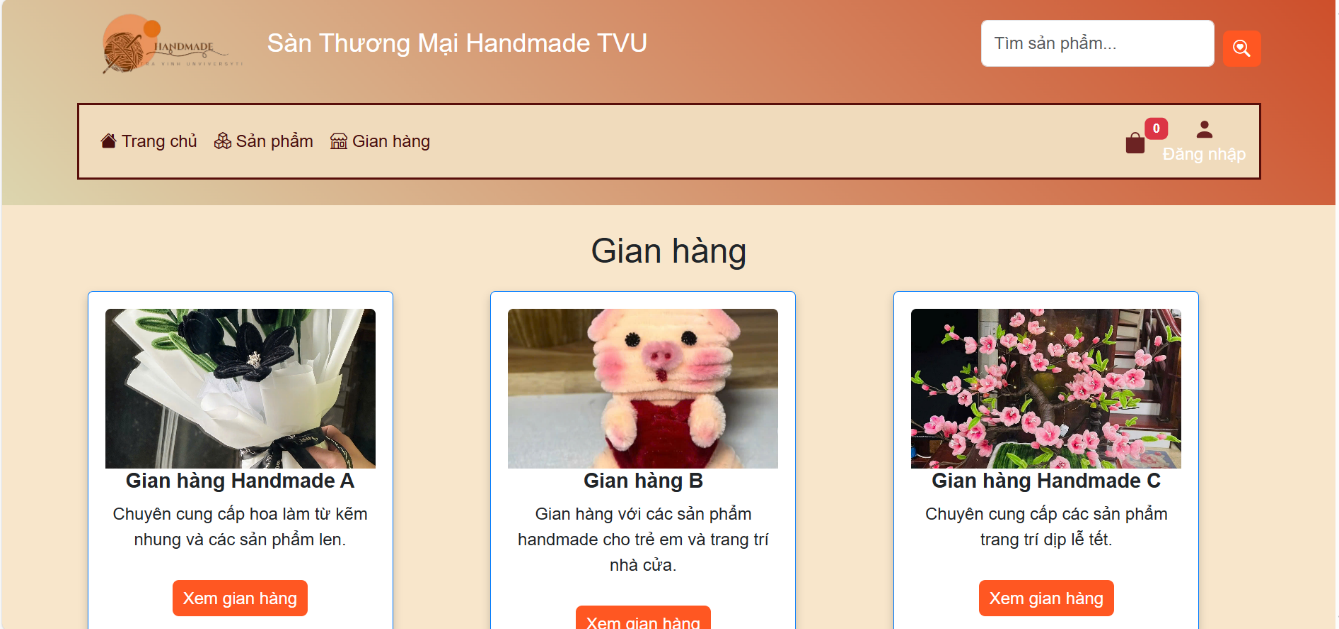


**Hình 4.3** Giao diện trang chi tiết sản phẩm

## *4.1.4 Giao diện các gian hàng*

Kết quả: hiển thị danh sách tất cả các gian hàng từ file JSON, mỗi gian hàng hiển thị tên, ảnh đại diện, và nút "Xem gian hàng".

Chức năng: điều hướng đến trang chi tiết của từng gian hàng.

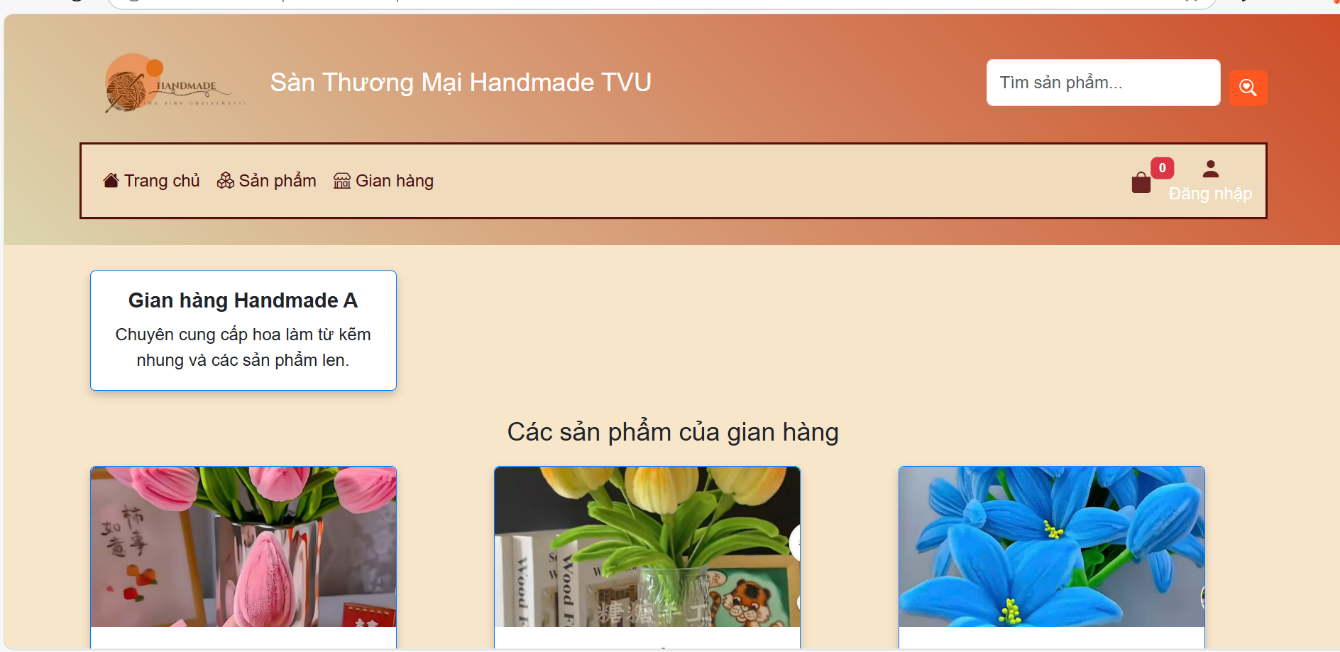


**Hình 4.4** Giao diện các gian hàng

## *4.1.5 Giao diện chi tiết gian hàng*

Kết quả: hiển thị thông tin cụ thể của gian hàng, bao gồm: tên, mô tả, và danh sách các sản phẩm được bán.

Chức năng: điều hướng đến chi tiết sản phẩm từ danh sách sản phẩm của gian hàng.

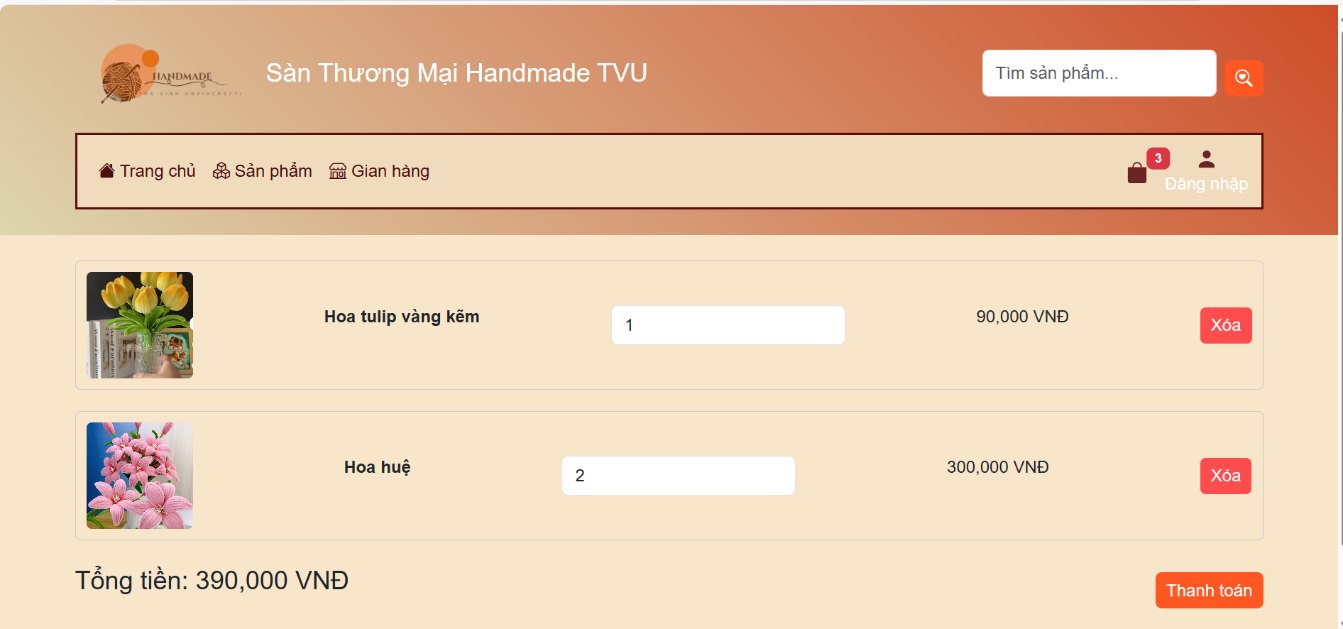


**Hình 4.5** Giao diện trang chi tiết gian hàng

## *4.1.6 Giao diện giỏ hàng*

Kết quả: hiển thị danh sách các sản phẩm đã thêm vào giỏ hàng, kèm thông tin như: tên, giá, số lượng, tính năng tính tổng tiền tự động cập nhật khi thêm/xóa sản phẩm.

Chức năng: xóa sản phẩm khỏi giỏ hàng, cập nhật tổng tiền theo số lượng sản phẩm và thanh toán sản phẩm.

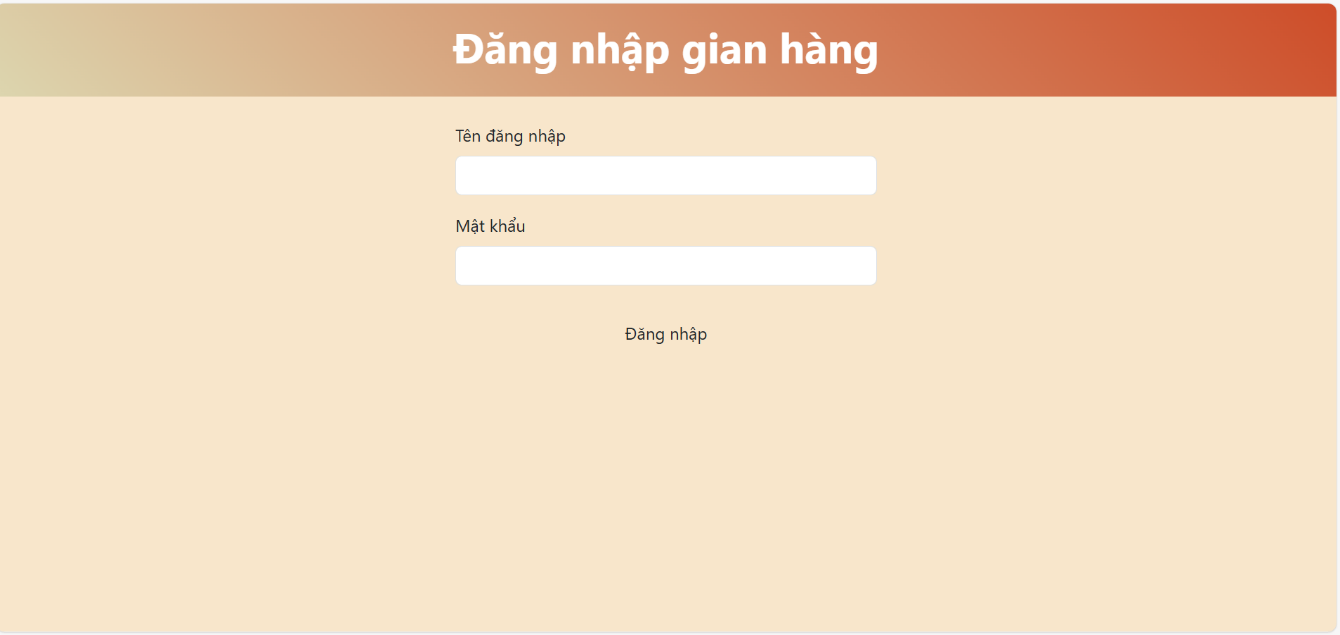


**Hình 4.6** Giao diện giỏ hàng

## *4.1.7 Giao diện đăng nhập của chủ gian hàng*

Kết quả: trang đăng nhập cho chủ gian hàng hoạt động đúng, kiểm tra tên đăng nhập và mật khẩu từ file JSON.

Chức năng: chuyển đến giao diện quản lý gian hàng sau khi đăng nhập thành công.

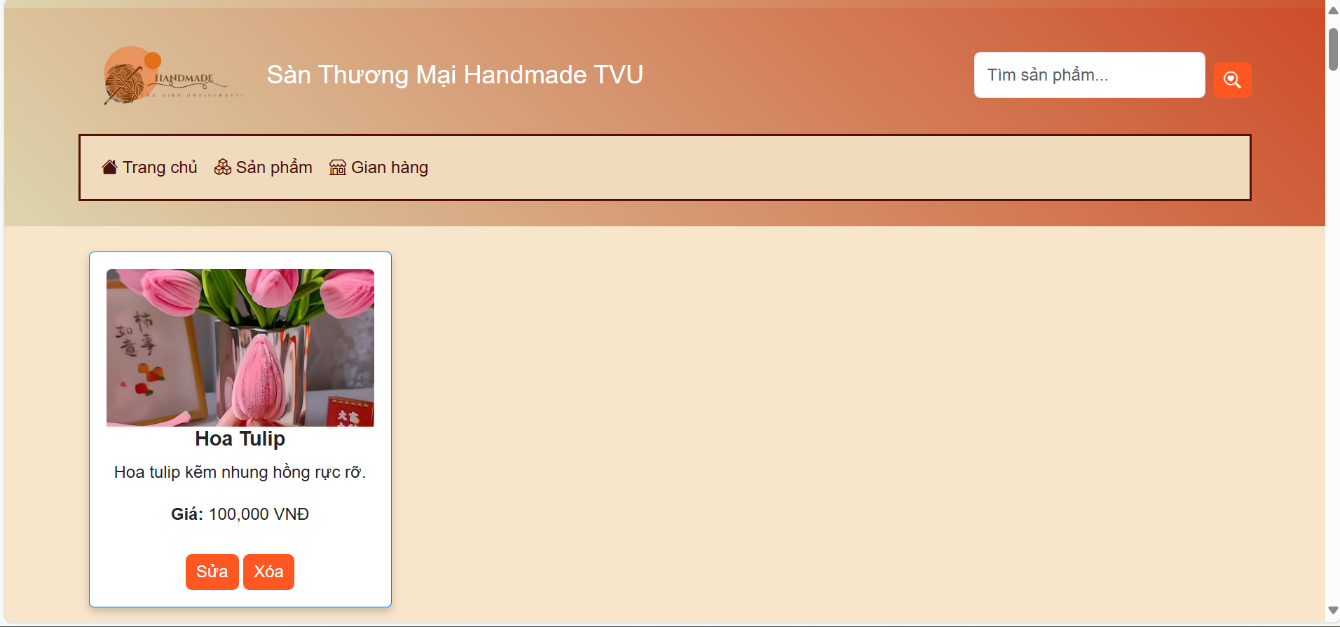


**Hình 4.7** Giao diện đăng nhập của chủ gian hàng

## *4.1.8 Giao diện quản lý gian hàng*

Kết quả: hiển thị danh sách sản phẩm mà chủ gian hàng đã đăng.

Chức năng: Sửa, xóa sản phẩm.



**Hình 4.8** Giao diện quản lý gian hàng

**CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

## 5.1 Kết luận kết quả đạt được

Thiết kế front-end cho sàn thương mại sản phẩm handmade của sinh viên TVU đã đạt được các mục tiêu đề ra, bao gồm:

Giao diện hoàn chỉnh: Đã xây dựng giao diện đầy đủ cho các trang web quan trọng, bao gồm: trang chủ, danh sách sản phẩm, chi tiết sản phẩm, danh sách gian hàng, chi tiết gian hàng, giỏ hàng, đăng nhập và quản lý gian hàng.

Chức năng cơ bản: Các chức năng như hiển thị sản phẩm từ API JSON, tìm kiếm, lọc sản phẩm, thêm/xóa sản phẩm trong giỏ hàng, và quản lý sản phẩm trong gian hàng đã được hiện thực hóa.

Tích hợp API: Kết nối thành công giữa front-end và file JSON thông qua json-server, đảm bảo dữ liệu được hiển thị chính xác và phản hồi nhanh chóng.

Hoàn thiện chức năng quản lý giỏ hàng, bao gồm thêm, xóa sản phẩm và tính toán tổng tiền.

## 5.2 Nhược điểm

Các giao diện các trang còn đơn giản, ít phần sinh động, kết hợp ít chức năng.

## 5.3 Hướng phát triển

Tích hợp thêm nhiều chức năng,

Cho phép khách hàng để lại đánh giá, nhận xét về sản phẩm và gian hàng, hiển thị điểm đánh giá trung bình trên các trang chi tiết sản phẩm và gian hàng.

Phát triển hệ thống đăng ký, đăng nhập người dùng với tính năng quản lý tài khoản và lịch sử mua hàng.

Tích hợp quy trình đặt hàng, theo dõi trạng thái đơn hàng và thông báo giao hàng.

**DANH MỤC TÀI LIỆU THAM KHẢO**

[1] FPT Smart Cloud, "HTML là gì? Tìm hiểu về ngôn ngữ HTML từ A đến Z", FPT Smart Cloud, 24/12/2024. Link: <https://fptcloud.com/html-la-gi/>

[2] T. Blog, "CSS là gì?", TopDev, 06/08/2024. Link: <https://topdev.vn/blog/css-la-gi/>

[3] Tiến C. L. V., “JavaScript là gì? Tổng quan kiến thức về JavaScript,” *Vietnix*, May 21, 2024. Link: https://vietnix.vn/javascript-la-gi/

[4] ICANTECH, "Bootstrap là gì? Tất cả những gì bạn cần biết về Bootstrap", ICANTECH, 14/12/2023. Link: <https://www.icantech.vn/kham-pha/bootstrap-la-gi>

[5] TopDev, "API là gì? Tại sao API được sử dụng nhiều hiện nay?", TopDev, 09/11/2023. Link: <https://topdev.vn/blog/api-la-gi/>

[6] T3H Team, “JSON File là gì? Ứng dụng và cách sử dụng trong Web development,” T3H, 24/12/2024. Link: https://t3h.com.vn/tin-tuc/json-file-la-gi-ung-dung-va-cach-su-dung-trong-web-development