

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

Vĩnh Long, ngày tháng năm
Giảng viên hướng dẫn
(Ký tên và ghi rõ họ tên)

NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG

Vĩnh Long, ngày tháng năm
Thành viên hội đồng
(Ký tên và ghi rõ họ tên)

LỜI CẢM ƠN

Tôi xin chân thành cảm ơn quý thầy cô đã tận tình hướng dẫn em trong quá trình thực hiện đồ án. Đặc biệt, cô Phạm Thị Trúc Mai đã tận tâm hướng dẫn, chỉ bảo tận tình và hỗ trợ em trong suốt thời gian thực hiện đồ án cơ sở ngành.

Tôi xin trân trọng cảm ơn các thầy cô Trường Đại học Trà Vinh đã truyền đạt cho em những kiến thức quý báu, những kinh nghiệm thực tiễn cùng với những phương pháp học tập và làm việc hiệu quả, giúp em có nền tảng xây dựng đồ án này.

Bên cạnh đó, tôi xin được gửi lời cảm ơn sâu sắc đến gia đình đã luôn động viên, ủng hộ và các bạn bè đã gắn bó, chia sẻ rất nhiều kinh nghiệm và những kiến thức nhất là trong thời gian thực hiện đồ án, để đồ án có thể hoàn thành một cách tốt nhất.

Tuy nhiên đã cố gắng hết sức, nhưng do thời gian có hạn và kinh nghiệm còn hạn chế, bài đồ án chắc chắn không tránh khỏi những thiếu sót. Tôi rất mong nhận được sự cảm thông, chỉ dẫn và góp ý từ quý thầy cô để tôi có thể hoàn thiện hơn trong tương lai.

Tôi xin chân thành cảm ơn.

Vĩnh Long, ngày... tháng... năm...

Sinh viên thực hiện

Trần Thị Yến Khoa

MỤC LỤC

TÓM TẮT ĐỒ ÁN CƠ SỞ NGÀNH	8
MỞ ĐẦU	9
CHƯƠNG 1: TỔNG QUAN	11
1.1 Giới thiệu tổng quan về vấn đề nghiên cứu	11
1.2 Giải quyết vấn đề	11
1.3. Một số khái niệm liên quan	12
1.3.1. Khái niệm về website tĩnh (Static Website)	12
1.3.2. Khái niệm về website động (Dynamic Website)	12
CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT	13
2.1 HTML - HyperText Markup Language	13
2.2 CSS – Cascading Style Sheets	13
2.3 JavaScript và xử lý logic	13
2.4 Bootstrap 5 – UI Framework	13
2.5 Font Awesome	13
2.6 Khái niệm JSON Server	14
2.7 Fetch API	14
CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU	16
3.1. Mô tả hệ thống	16
3.2. Thiết kế hệ thống	16
3.2.1. Thiết kế các chức năng của website	16
3.2.2. Thiết kế giao diện và bố cục	17
3.3. Thiết kế cấu trúc dữ liệu JSON	18
3.3.1 Các thuộc tính để xây dựng dữ liệu sản phẩm	18
3.4. Hiện thực hóa chức năng	20
3.4.1. Cấu trúc thư mục dự án	20
3.4.2. Hiển thị dữ liệu động bằng Fetch API	20
3.5. Cài đặt và triển khai chương trình	23
3.5.1. Cài đặt môi trường	23
3.5.2. Cài đặt Node.js và JSON Server	23
3.5.3. Khởi chạy JSON Server	24
CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU	25
4.1. Kết quả đạt được	25
4.2. Đánh giá về hiệu năng và trải nghiệm của người dùng	29
CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	30
5.1. Những kết quả đạt được	30
5.2. Hạn chế và hướng phát triển	30

DANH MỤC HÌNH ẢNH

Hình 3.1 Phát thảo bộ cục giao diện.....	17
Hình 3.2 Cấu trúc cây thư mục.....	20
Hình 3.3 Quá trình Fetch API.....	21
Hình 4.1 Thanh menu.....	25
Hình 4.2 Giao diện sản phẩm trang chủ.....	25
Hình 4.3 Giao diện sản phẩm.....	26
Hình 4.4 Giao diện chi tiết sản phẩm	27
Hình 4.5 Giao diện giỏ hàng.....	28

DANH MỤC BẢNG BIỂU

Bảng 3.1 Mô tả thuộc tính tệp JSON	18
--	----

DANH MỤC TỪ VIẾT TẮT

Từ viết tắt	Tên đầy đủ	Ý nghĩa
HTML	HyperText Markup Language	Ngôn ngữ đánh dấu dùng để xây dựng cấu trúc nội dung website
CSS	Cascading Style Sheets	Ngôn ngữ định dạng giao diện website
JS	JavaScript	Ngôn ngữ lập trình xử lý logic và tương tác người dùng
UI	User Interface	Giao diện người dùng
UX	User Experience	Trải nghiệm người dùng
API	Application Programming Interface	Giao diện lập trình cho phép trao đổi dữ liệu giữa các hệ thống
JSON	JavaScript Object Notation	Định dạng dữ liệu nhẹ dùng để trao đổi dữ liệu
DOM	Document Object Model	Mô hình đối tượng tài liệu, cho phép JavaScript thao tác với HTML
CDN	Content Delivery Network	Mạng phân phối nội dung, dùng để tải thư viện nhanh
HTTP	HyperText Transfer Protocol	Giao thức truyền tải dữ liệu trên web

TÓM TẮT ĐỀ ÁN CƠ SỞ NGÀNH

Đề tài tập trung phát triển và xây dựng một website tĩnh (Static Website) phục vụ cho bán trang sức trực tuyến. Trong đó dữ liệu sản phẩm được tải tự động từ JSON Server hoặc tệp JSON thông qua Fetch API thay vì dùng Backend.

Mục tiêu đề tài hướng đến là tạo một website tĩnh những vẫn có khả năng tìm kiếm, lọc sản phẩm, thêm sản phẩm vào giỏ hàng và hiện dữ liệu động, mô phỏng quá trình đăng nhập và đăng ký. Từ đó mô phỏng các chức năng cơ bản của một website mà không cần sử dụng máy chủ Backend.

Đề tài được tiếp cận theo mô hình Front-end thuần và tập trung vào các công cụ đơn giản: HTML, CSS, Bootstrap 5 để định dạng giao diện người dùng (UI) và bố cục responsive. JavaScript thuần (Vanilla JS) được áp dụng để xử lý sự kiện và tương tác với người dùng. Dữ liệu sản phẩm được tải từ tệp products.json thông qua Fetch API và được render động lên giao diện bằng thao tác DOM mà không cần Backend.

Để thực hiện được đề tài, đầu tiên phân tích và thiết kế, xác định các chức năng cơ bản của website: trang chủ, trang sản phẩm, chi tiết sản phẩm, tìm kiếm, lọc sản phẩm, trang đăng nhập và đăng ký tài khoản và footer thông tin liên hệ. Bên cạnh đó, xây dựng cấu trúc dữ liệu JSON gồm các thuộc tính sản phẩm: mã sản phẩm, tên sản phẩm, giá bán hiện tại, mô tả, hình ảnh,...

Thứ hai triển khai kỹ thuật bao gồm xây dựng cấu trúc thư mục theo chuẩn Front-end. Xây dựng giao diện bằng HTML kết hợp Bootstrap. Sử dụng JavaScript để fetch dữ liệu, render danh sách sản phẩm, hiển thị chi tiết sản phẩm theo ID. Cài đặt chức năng lọc, tìm kiếm sản phẩm và thêm sản phẩm vào giỏ hàng bằng cách lưu trữ tạm thời bằng LocalStorage trên trình duyệt. Cuối cùng là sử dụng Fetch API cho trao đổi dữ liệu JSON và LocalStorage API để lưu trữ trạng thái người dùng mà không cần Backend.

Sau quá trình thực hiện, đề tài đạt được một số kết quả sau: hoàn thiện website tĩnh có khả năng hiển thị dữ liệu động từ API JSON, giao diện hiện đại và tương thích nhiều thiết bị; triển khai thành công các chức năng mô phỏng cơ bản của một website động; biết được cơ chế kết nối dữ liệu bất đồng bộ (asynchronous data fetching) bằng Fetch API, cùng với kỹ năng thao tác DOM và nâng cao kỹ năng thiết kế giao diện với Bootstrap.

MỞ ĐẦU

Trong thời đại chuyển đổi số ngày càng được phát triển mạnh mẽ trong nhiều lĩnh vực, thị trường cạnh tranh ngày càng mãnh liệt các công ty lớn đều quan tâm đến việc làm thỏa mãn nhu cầu của khách hàng một cách tốt nhất có thể. Đặc biệt là lĩnh vực thương mại điện tử, những website bán hàng ngày càng phổ biến và góp phần đóng vai trò quan trọng trong việc quảng bá sản phẩm và tương tác với khách hàng.

Trước đây các doanh nghiệp kinh doanh theo mô hình truyền thống, khách đến cửa hàng chọn sản phẩm. Hình thức này còn nhiều bất lợi như số lượng mẫu mã chưa được cập nhật thường xuyên, việc tìm kiếm mất nhiều thời gian, từ đó ảnh hưởng đến trải nghiệm mua sắm của khách hàng. Đặc biệt là trong lĩnh vực trang sức, đòi hỏi doanh nghiệp phải cập nhật liên tục mẫu trang sức theo xu hướng, tìm kiếm và lọc sản phẩm còn mất nhiều thời gian.

Để làm thỏa mãn mọi nhu cầu của người tiêu dùng về tính thẩm mỹ, tìm kiếm nhanh, lọc theo yêu cầu một cách nhanh chóng. Xây dựng cấu trúc nội dung của website bằng HTML để lập trình. CSS kết hợp với Bootstrap 5 nhằm thiết kế giao diện nhanh chóng, chuyển đổi kích thước linh hoạt đa thiết bị. Dùng JavaScript thuận thực hiện các sự kiện, nhằm nâng cao chất lượng tương tác người dùng và thao tác DOM. Bên cạnh đó, Fetch API được áp dụng để lấy dữ liệu từ tệp JSON hoặc JSON Server hiển thị dữ liệu nhanh chóng một cách bất đồng bộ. Giữ hàng lưu trữ sản phẩm và người dùng tạm thời trên trình duyệt bằng LocalStorage API, phù hợp với website tĩnh.

Nhằm đáp ứng nhu cầu mua sắm, tối ưu hóa trải nghiệm người dùng và khắc phục những bất lợi của mô hình kinh doanh truyền thống, tôi chọn đề tài “**Phát triển website tĩnh bán trang sức với dữ liệu động API JSON**”. Mục tiêu xây dựng một hệ thống có giao diện hiện đại, thẩm mỹ cao dễ triển khai, tốc độ truy cập nhanh đáp ứng trải nghiệm người dùng. Sử dụng Fetch API để lấy dữ liệu động từ API JSON, nhằm mô phỏng hoạt động của một website thương mại điện tử cơ bản.

Đối tượng nghiên cứu: Quy trình và kỹ thuật xây dựng website tĩnh có dữ liệu động từ API JSON.

Phạm vi nghiên cứu: Website tĩnh được xây dựng từ HTML, CSS, JavaScript, Bootstrap và Fetch API thông qua dữ liệu mô phỏng qua JSON, không có Backend.

Quá trình thực hiện đề tài được triển khai qua phương pháp nghiên cứu lý và thực nghiệm. Trong phương pháp lý thuyết tôi tìm hiểu về HTML, CSS, JavaScript cùng với Bootstrap 5 nhằm nắm các kiến thức cơ bản, xây dựng bối cảnh. Đồng thời tìm hiểu cách dùng JSON Server làm API giả lập mà không cần cơ sở dữ liệu phức tạp. Tìm cách dùng Fetch API để lấy dữ liệu, thực hiện các chức năng.

Giai đoạn dùng phương pháp thực nghiệm để xây dựng giao diện chi tiết cho các trang web bằng thư viện Bootstrap 5 kết hợp JavaScript để xử lý logic. Tiếp theo là dùng JSON Server làm API giả lập, kết hợp Fetch API render lên trên website.

CHƯƠNG 1: TỔNG QUAN

1.1 Giới thiệu tổng quan về vấn đề nghiên cứu

Trong thời đại chuyển đổi số ngày càng phát triển, người dùng Internet càng tăng mạnh theo [1] số lượng người dùng Internet toàn cầu đạt 5.35 tỷ, tương đương 66,2% dân số thế giới, tạo điều kiện thuận lợi cho các hoạt động trực tuyến. Cùng với sự phát triển của thời đại công nghệ, các doanh nghiệp chú trọng việc xây dựng các website thương mại điện tử đóng vai trò quan trọng trong kinh doanh trực tuyến. Các doanh nghiệp phần lớn cần những trang web để quảng bá và giới thiệu sản phẩm và tiếp cận với khách hàng một cách tối ưu, nhanh chóng và linh hoạt nhất.

Tuy nhiên, không phải doanh nghiệp nào cũng có đủ nguồn tài nguyên và nhân lực để xây dựng và phát triển một hệ thống phức tạp. Những kiểu kinh doanh nhỏ chỉ cần website tĩnh quảng bá sản phẩm nhưng có thể cập nhật dữ liệu linh hoạt. Chính vì vậy, để tài chủ yếu phát triển website tĩnh có khả năng hiển thị thông tin sản phẩm thông qua dữ liệu giả lập API JSON.

Vấn đề sẽ thực hiện là làm sao để phát triển một website tĩnh nhưng vẫn có khả năng hiển thị thông tin sản phẩm một cách linh hoạt, cập nhật nguồn dữ liệu API JSON, không cần sử dụng cơ sở dữ liệu.

1.2 Giải quyết vấn đề

Để giải quyết những vấn đề trên, trong đề tài sẽ sử dụng các công nghệ thường dùng khi lập trình, đầu tiên thực hiện được phần giao diện (Front-end) bằng các ngôn ngữ sau: HTML, CSS, JavaScript, Bootstrap 5. Giúp trang web tương thích với đa thiết bị, tốc độ tải nhanh.

Tiếp theo là xây dựng website không dùng cơ sở dữ liệu, thay vào đó sử dụng JavaScript lấy dữ liệu từ API JSON để xử lý dữ liệu thông qua phương thức Fetch(), hiển thị kết quả lên trang web. Giúp giảm chi phí thực triển khai và cập nhật dữ liệu tự động.

Mô phỏng quá trình lấy dữ liệu động thông qua API JSON để hiển thị danh sách sản phẩm, chi tiết sản phẩm, tìm kiếm và các chức năng khác của một website động cơ bản.

1.3. Một số khái niệm liên quan

1.3.1. Khái niệm về website tĩnh (Static Website)

Là trang web mà toàn bộ nội dung được viết cố định bằng HTML, CSS và JavaScript, không có chức năng lưu trữ và tương tác với dữ liệu. Thường dùng để mô tả giao diện hoặc chức năng của trang web. Ưu điểm của website tĩnh có tốc độ tải nhanh, dễ triển khai. Tuy nhiên, website tĩnh không có khả năng tương tác với người dùng và không phụ thuộc vào máy chủ xử lý dữ liệu.

1.3.2. Khái niệm về website động (Dynamic Website)

Là trang web có nội dung được tạo ra tự động từ máy chủ thông qua cơ sở dữ liệu. Người dùng có thể tương tác, tìm kiếm, lọc dữ liệu và trải nghiệm nội dung thay đổi theo thời gian thực. Hạn chế của website động là phức tạp và yêu cầu hệ thống Back-end.

Từ sự khác biệt về ưu điểm và hạn chế của website tĩnh và website động, đề tài tập trung phát triển ưu điểm website tĩnh kết hợp với khả năng truy xuất dữ liệu linh hoạt từ API JSON. Từ đó mô phỏng các chức năng cơ bản của website như một website bán hàng trực tuyến.

CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT

2.1 HTML - HyperText Markup Language

HTML là ngôn ngữ đánh dấu, chú thích văn bản, hình ảnh được dùng để xây dựng cấu trúc nội dung cho trang web. HTML gồm nhiều thẻ như: <header>, <footer>, <head>, <body>, biểu diễn từng thành phần như đoạn văn, hình ảnh, liên kết, danh sách, bố cục,...[2], [3].

2.2 CSS – Cascading Style Sheets

CSS là một ngôn ngữ định dạng được sử dụng để mô tả cách trình bày tài liệu được viết bằng HTML hoặc XML (bao gồm các XML như SVG, MathML hoặc XHTML). CSS mô tả cách các thành phần được hiển thị trên màn hình, trên giấy, trong giọng nói hoặc trên các phương tiện khác [4].

2.3 JavaScript và xử lý logic

JavaScript là một ngôn ngữ kịch bản hướng đối tượng, đa nền tảng, được sử dụng để tạo tính tương tác cho các trang web (ví dụ: có các hoạt ảnh phức tạp, nút bấm có thể nhấp, menu bật lên) [5]. Có thể tương tác với các đối tượng, cung cấp khả năng điều khiển theo chương trình.

2.4 Bootstrap 5 – UI Framework

Là một framework front-end mã nguồn mở dùng HTML, CSS và JavaScript để xây dựng giao diện website (UI) nhanh chóng. Framework này cung cấp hệ thống lưới linh hoạt, các phần giao diện có sẵn như navbar, card, button,... và hỗ trợ responsive giúp website hiển thị tốt trên đa dạng thiết bị [6].

Trong đề tài này, Bootstrap 5 được áp dụng để định dạng bố cục ở tất cả các trang. Giúp bố cục website dễ nhìn, thẩm mỹ và phù hợp với nét thẩm mỹ sang trọng của ngành trang sức. Việc lựa chọn Bootstrap 5 là một giải pháp tối ưu để đẩy nhanh tiến độ thiết kế website, đảm bảo tính tương thích nhiều thiết bị, dễ chỉnh sửa và nâng cấp trong tương lai.

2.5 Font Awesome

Font Awesome là một thư viện biểu tượng (icon) phổ biến, cung cấp nhiều biểu tượng đa dạng, cho phép chèn các biểu tượng (icon) vào website thông qua CSS hoặc CDN. Thư viện cung cấp nhiều icon phục vụ cho giao diện người dùng như biểu tượng tìm kiếm, giỏ hàng, người dùng [7]

Để tối ưu hóa trải nghiệm khách hàng, trong đề tài có tích hợp thư viện này vào nhằm giúp người dùng dễ dàng nhận diện và thao tác. Việc lựa chọn Font Awesome không chỉ đảm bảo tính thẩm mỹ tinh tế cho một website trang sức mà còn tối ưu hóa dung lượng và khả năng hiển thị đa dạng sắc nét trên mọi độ phân giải.

2.6 Khái niệm JSON Server

JSON Server là một công cụ giả lập API bằng tệp JSON (JavaScript Object Notation) dựa trên Node.js. Cho phép thử nghiệm giao diện ứng dụng mà không cần mã Backend. [8]

Tệp JSON là định dạng dữ liệu dễ đọc và dễ trao đổi giữa Client và server.

JSON Server đóng vai trò giả lập API: Giúp mô phỏng quá trình lấy dữ liệu từ máy chủ, sau đó trả về dưới dạng JSON, dễ dàng phát triển website mà không cần xây dựng Backend phức tạp.

2.7 Fetch API

Fetch API là một giao diện lập trình ứng dụng (API) được cung cấp sẵn trong JavaScript, người dùng được cho phép gửi các yêu cầu HTTP (GET, POST,...) đến máy chủ hoặc nguồn dữ liệu và nhận phản hồi dưới dạng bất đồng bộ. Ưu điểm của công cụ này là khả năng truy xuất dữ liệu từ các nguồn API JSON mà không cần tải lại toàn bộ trang web, từ đó tối ưu hóa hiệu năng của hệ thống và trải nghiệm người dùng. Nói ngắn gọn API Fetch trong JavaScript cho phép gửi yêu cầu đến máy chủ và nhận dữ liệu JSON [9].

Cách hoạt động của Fetch. Đầu tiên Fetch API sẽ gửi yêu cầu đến một nguồn dữ dữ liệu, sau đó nhận phản hồi (response), chuyển đổi dữ liệu sang định dạng JSON và xử lý hiển thị nội dung lên giao diện thông qua JavaScript. Quá trình này diễn ra bất đồng bộ, đảm bảo website vẫn hoạt động mượt mà trong khi dữ liệu đang được tải.

Trong khuôn khổ đề tài này, Fetch API được áp dụng để lấy danh sách sản phẩm, thông tin chi tiết sản phẩm và các dữ liệu liên quan từ API JSON, sau đó hiển thị động lên website. Thông qua Fetch API, website tĩnh vẫn có thể mô phỏng các chức năng cơ bản của một website thương mại điện tử như hiển thị sản phẩm, tìm kiếm và lọc dữ liệu, đăng ký và đăng nhập tài khoản.

Việc lựa chọn Fetch API giúp giảm sự phụ thuộc vào thiết kế Backend, đơn giản hóa quá trình phát triển, dễ triển khai và phù hợp với mô hình website tĩnh sử dụng dữ liệu giả lập API JSON.

CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU

3.1. Mô tả hệ thống

Website tĩnh bán trang sức gồm các chức năng: trang chủ, hiển thị danh sách sản phẩm theo loại, xem chi tiết từng sản phẩm, tìm kiếm và lọc sản phẩm theo nhu cầu người dùng, thêm sản phẩm vào giỏ hàng, đăng nhập và đăng ký tài khoản. Website không dùng Backend và cơ sở dữ liệu để lưu trữ mà thay vào đó là sử dụng API giả lập từ tệp JSON để fetch API lấy dữ liệu bất đồng bộ hiện trang web. Giỏ hàng và trạng thái người dùng được lưu trữ tạm thời trên trình duyệt thông qua LocalStorage API. Mục tiêu của hệ thống là mô phỏng các chức năng cơ bản của một website thương mại điện tử, tập trung vào khả năng hiển thị dữ liệu động, tối ưu tốc độ tải trang và nâng cao trải nghiệm người dùng.

3.2. Thiết kế hệ thống

3.2.1. Thiết kế các chức năng của website

Dựa trên những gì đã phân tích thì trang web được thiết kế gồm các chức năng chính sau:

Trang chủ: Dùng để giới thiệu sản phẩm, banner, sliders sản phẩm nổi bật và các sản phẩm bán chạy để thu hút người dùng.

Trang sản phẩm: Dùng để hiện thị tất cả sản phẩm từ file JSON, có thanh menu để xem từng trang sản phẩm riêng biệt (ví dụ: các sản phẩm về nhẫn, vòng tay,...). Cho phép người dùng xem tất cả thông tin chi tiết liên quan đến sản phẩm.

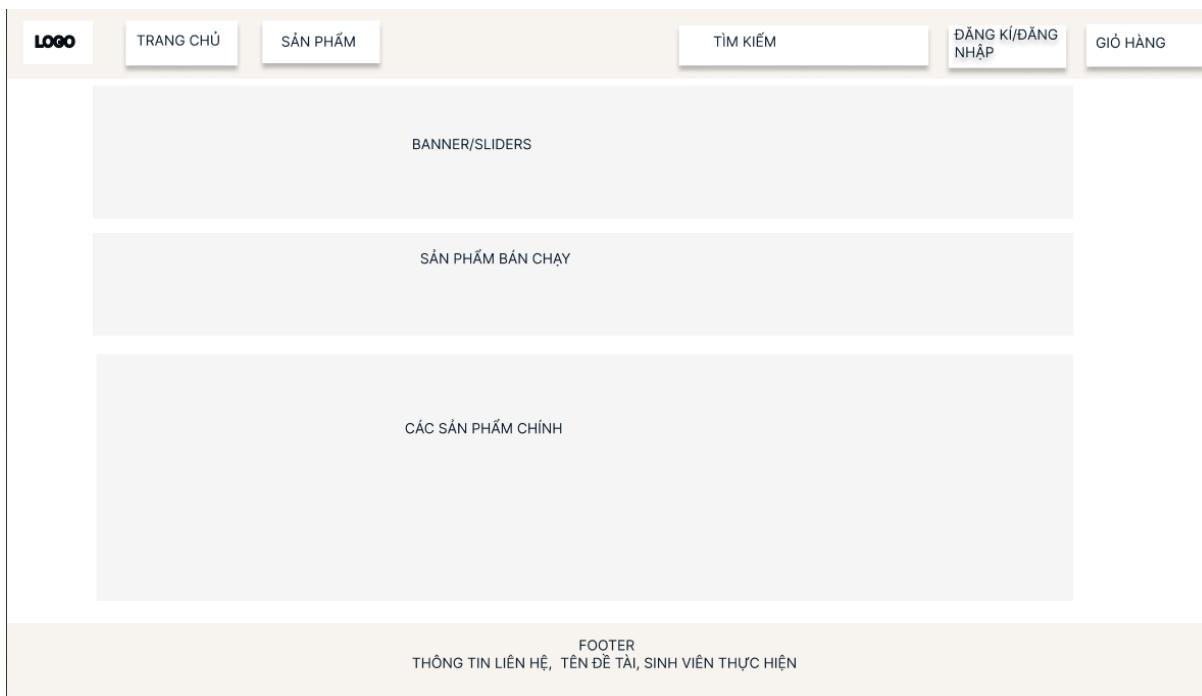
Chức năng tìm kiếm: Cho phép người dùng tìm kiếm sản phẩm theo tên.

Chức năng lọc sản phẩm: Lọc sản phẩm theo giá, vật liệu trang sức.

Giỏ hàng: Cho phép thêm sản phẩm vào giỏ hàng, xem sản phẩm có trong giỏ hàng, tùy chỉnh theo số lượng và được lưu dữ liệu giỏ hàng bằng LocalStorage.

Footer: Hiển thị thông tin liên hệ và tên đê tài, người thực hiện.

3.3.2. Thiết kế giao diện và bố cục



Hình 3.1 Phát thảo bố cục giao diện

Bố cục tổng thể của website được phát thảo theo hướng đơn giản, trực quan:

Header: đặt ở phía trên cùng, bao gồm logo thương hiệu, thanh menu (trang chủ, trang sản phẩm, giỏ hàng, icon đăng ký và đăng nhập), ô tìm kiếm và icon tìm kiếm.

Main content:

- Trang chủ hiển thị banner, sliders giới thiệu và các sản phẩm nổi bật, sản phẩm bán chạy.
- Trang sản phẩm hiển thị danh sách trang sức theo dạng lưới, mỗi sản phẩm gồm hình ảnh, tên, giá bán hiện tại và nút xem chi tiết.
- Trang chi tiết sản phẩm hiển thị hình ảnh lớn, thông tin chi tiết sản phẩm và nút thêm vào giỏ hàng.
- Trang giỏ hàng hiển thị các sản phẩm được chọn bao gồm ảnh, tên, giá bán của sản phẩm, nút tăng giảm số lượng sản phẩm, nút xóa.

- Trang đăng ký và đăng nhập hiển thị danh sách các ô điền thông tin.

Footer: Hiển thị thông tin liên hệ, thông tin của hàng, tên người làm và tên đề tài.

3.3. Thiết kế cấu trúc dữ liệu JSON

3.3.1 Các thuộc tính để xây dựng dữ liệu sản phẩm

Các dữ liệu sản phẩm được lưu trong file products.json. Chi tiết các thuộc tính sau của một sản phẩm:

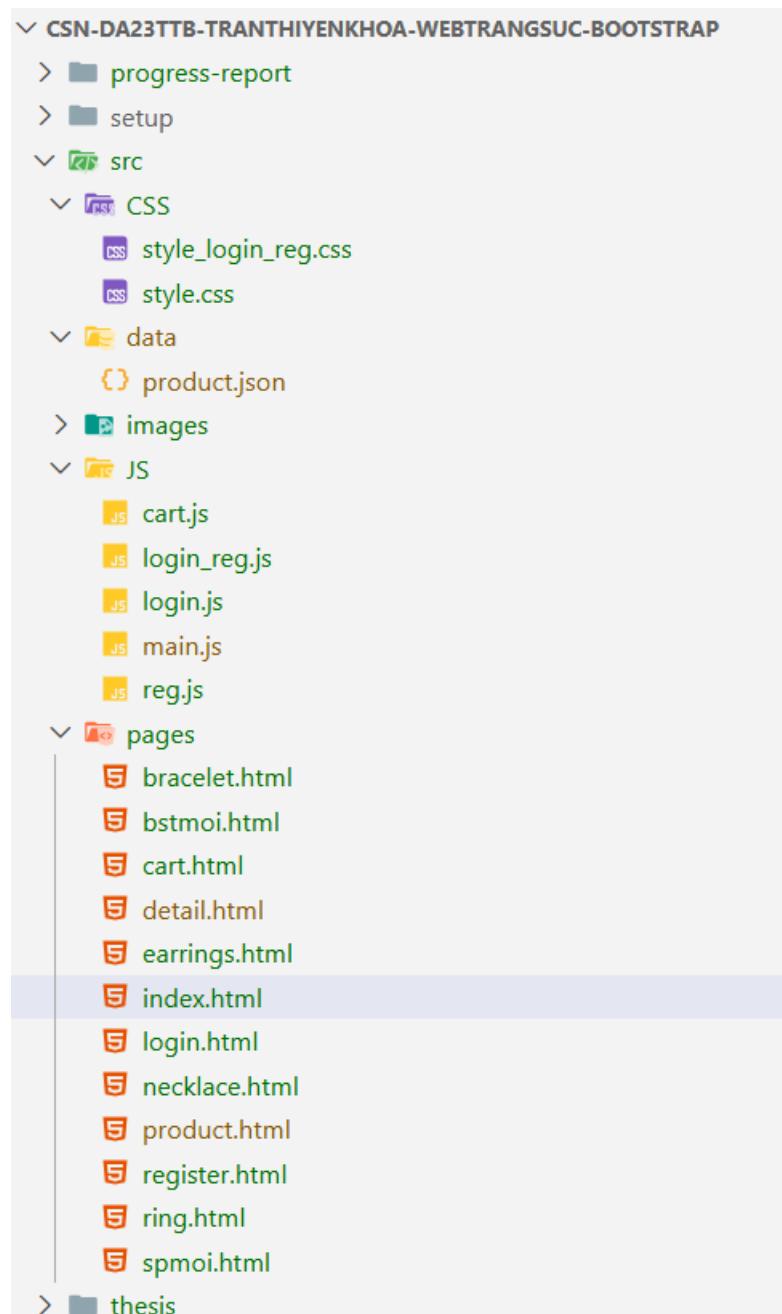
Bảng 3.1 Mô tả thuộc tính tệp JSON

Tên thuộc tính	Ý nghĩa	Mục đích
id	Mã sản phẩm	Phân biệt sản phẩm, dùng khi tìm kiếm, thêm giỏ hàng
name	Tên sản phẩm	Hiển thị tên sản phẩm lên giao diện
original_price	Giá bán gốc của sản phẩm	Giá bán trước khi được giảm
price	Giá bán hiện tại	Giá thanh toán, tính tổng tiền, lọc theo giá bán
category	Danh mục sản phẩm	Phân loại trang sức (Nhẫn, Hoa tai, Vòng tay,...), lọc sản phẩm
material	Vật liệu sản phẩm	Hiển thị thông tin, lọc theo vật liệu.

Tên thuộc tính	Ý nghĩa	Mục đích
image	Đường dẫn hình ảnh sản phẩm	Hiển thị hình ảnh sản phẩm
descripton	Thông tin chi tiết của sản phẩm	Mô tả chi tiết sản phẩm
quantity	Số lượng sản phẩm đang có	Kiểm soát số lượng sản phẩm
featured	Trạng thái các sản phẩm nổi bật	Hiển thị sản phẩm ở trang chủ

3.4. Hiển thực hóa chức năng

3.4.1. Cấu trúc thư mục dự án



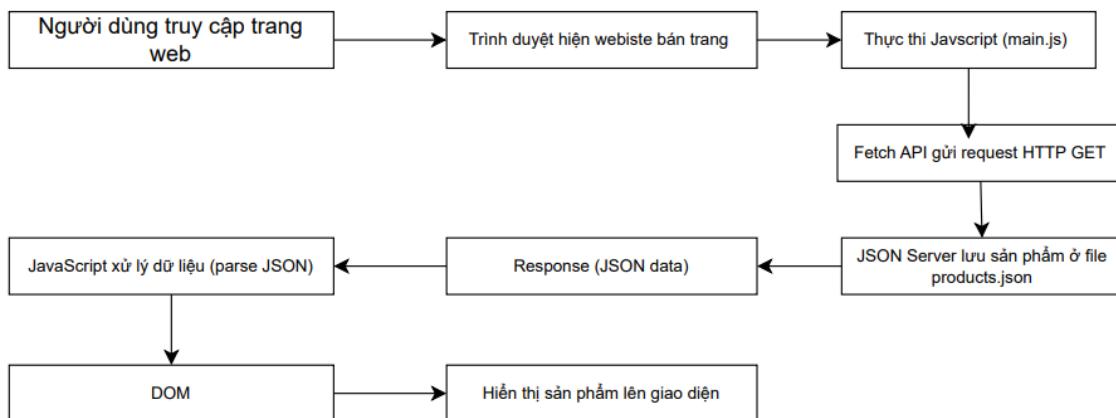
Hình 3.2 Cấu trúc cây thư mục

3.4.2. Hiển thị dữ liệu động bằng Fetch API

Website sử dụng Fetch API để gửi yêu cầu HTTP GET đến API giả lập từ JSON Server hoặc tệp JSON. Dữ liệu sau khi được phản hồi dưới dạng JSON sẽ được JavaScript xử lý, chuyển đổi thành các phần tử HTML thông qua thao tác DOM và hiển thị động lên giao diện người dùng.

Phát triển website tĩnh bán trang sức với dữ liệu động từ API JSON

Quá trình xử lý dữ liệu được minh họa thông qua sơ đồ sau:



Hình 3.3 Quá trình Fetch API

Website sử dụng Fetch API để lấy dữ liệu từ json. Dữ liệu sau khi được fetch được xử lý bằng JavaScript và hiển thị ra giao diện. Giúp hiển thị dữ liệu linh hoạt, cập nhật sản phẩm nhanh.

```
const API_URL = "http://localhost:3000/products";
let allProduct = [];
let currentProduct = null;

async function fetchProducts() {
    try {
        const res = await fetch(API_URL);
        const data = await res.json();

        console.log("DATA API:", data);
        return data; // LẤY ĐÚNG MẢNG
    } catch (error) {
        console.error("Lỗi API:", error);
        return [];
    }
}
```

Đoạn mã trên áp dụng Fetch API để gửi yêu cầu HTTP GET đến API giả lập từ JSON Server. Dữ liệu nhận về dưới dạng JSON được xử lý và truyền vào hàm renderProducts() để hiển thị lên giao diện, catch() được dùng để xử lý lỗi trong trường hợp không thể truy xuất dữ liệu.

```
function renderProductList(products) {
```

```
const productList = document.getElementById("product-list");
if (!productList) return;

if (products.length === 0) {
    productList.innerHTML =
        <p class="text-center text-danger">
            Không có sản phẩm phù hợp
        </p>;
    return;
}

productList.innerHTML = products.map(item => `

<div class="col-lg-3 col-md-4 col-sm-6 mb-4">
    <div class="card h-100 shadow-sm border-0 position-relative">

        
        <div class="card-body text-center">
            <h6 class="card-title">${item.name}</h6>

            ${item.original_price ? `

                <p class="mb-1 text-muted text-decoration-line-through">
                    ${item.original_price.toLocaleString()} VND
                </p>` : ""}

            <p class="text-danger fw-bold">
                ${item.price.toLocaleString()} VND
            </p>

            <a href="detail.html?id=${item.id}" class="btn btn-outline-warning btn-sm">
                Xem chi tiết
                <i class="fa-solid fa-eye"></i>
            </a>
        </div>
    </div>
`).join(``;
```

Hàm renderProducts() thực hiện thao tác DOM để chuyển đổi dữ liệu JSON thành các phần tử HTML và hiển thị danh sách sản phẩm theo dạng lưới. Việc render động giúp website tĩnh có khả năng hiển thị dữ liệu linh hoạt mà không cần tải lại trang.

3.5. Cài đặt và triển khai chương trình

3.5.1. Cài đặt môi trường

Cần cài các phần mềm và công cụ sau:

Node.js môi trường chạy JavaScript phía máy chủ, dùng để cài đặt và chạy JSON Server.

JSON Server công cụ tạo API giả lập từ tệp JSON, phục vụ cho việc đọc dữ liệu sản phẩm.

Trình duyệt: Google Chrome, Microsoft Edge hoặc Firefox để chạy giao diện website.

3.5.2. Cài đặt Node.js và JSON Server

Bước 1: Cài đặt Node.js

Truy cập trang chủ Node.js và tải phiên bản LTS (Long Term Support).

Tiến hành cài đặt theo hướng dẫn mặc định.

Sau khi cài đặt, kiểm tra bằng lệnh:

```
node -v  
npm -v
```

Bước 2: Cài đặt JSON Server

Mở Command Prompt hoặc Terminal.

Thực hiện lệnh sau để cài đặt JSON Server toàn cục:

```
npm install -g json-server
```

Kiểm tra JSON Server đã được cài đặt:

```
json-server -version
```

3.5.3. Khởi chạy JSON Server

Trong thư mục dự án, mở thư mục **setup/**.

File db.json chứa dữ liệu sản phẩm giả lập.

Tại thư mục này, mở Command Prompt và chạy lệnh:

```
json-server --watch db.json --port 3000
```

Sau khi chạy thành công, hệ thống tạo API giả lập như:

<http://localhost:3000/products>

CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU

4.1. Kết quả đạt được

Sau quá trình triển khai, website bán trang sức đã được xây dựng hoàn chỉnh với các chức năng cơ bản. Các giao diện chính của hệ thống được minh họa trong các hình sau.



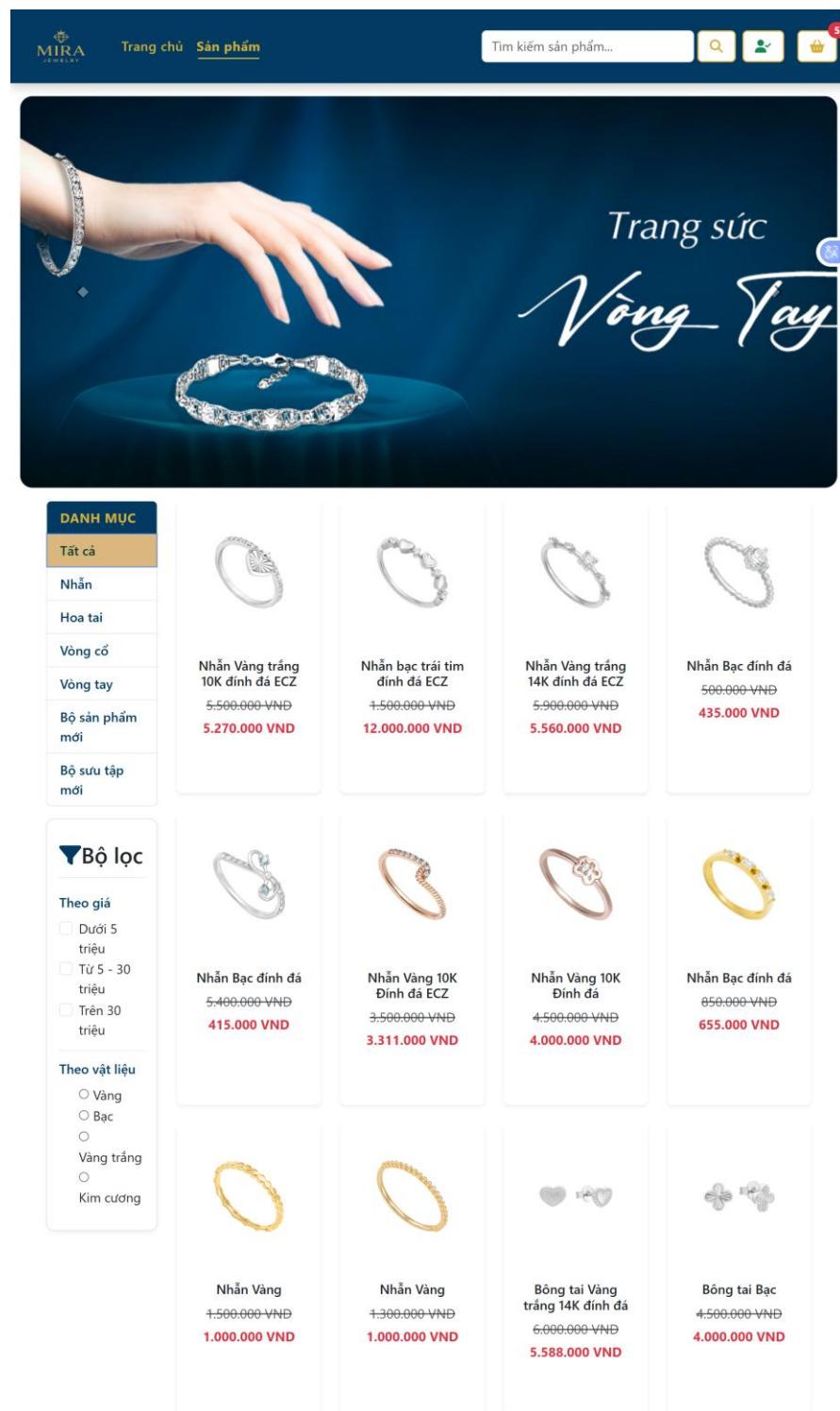
Hình 4.1 Thanh menu

Thanh navbar được thiết kế theo phong cách hiện đại và tối giản có vai trò nêu nhận diện website. Bao gồm logo thương hiệu, menu điều hướng các trang chức năng, thanh tìm kiếm sản phẩm và các biểu tượng tiện ích như tài khoản người dùng và giỏ hàng. Tông màu chủ đạo là xanh than và vàng đồng để làm nổi bật dự sang trọng phù hợp với kinh doanh trang sức.



Hình 4.2 Giao diện sản phẩm trang chủ

Trang chủ của website đóng vai trò là nơi giới thiệu tổng quan các danh mục và sản phẩm nổi bật. Phần đầu trang hiển thị khu vực gợi ý tìm kiếm với các danh mục trang sức giúp người dùng nhanh chóng chọn loại sản phẩm mong muốn. Bên dưới là khu vực Sản phẩm mới, được trình bày dưới dạng các thẻ sản phẩm (card) rõ ràng, bao gồm hình ảnh, tên sản phẩm và giá bán nổi bật.



Hình 4.3 Giao diện sản phẩm

Trong giao diện trên, khu vực banner phía trên cùng dùng để giới thiệu loại sản phẩm, giúp thu hút sự chú ý của khách hàng. Bên trái màn hình là menu danh mục và bộ lọc, cho phép người dùng lọc sản phẩm theo loại trang sức, mức giá và chất liệu. Khu vực trung tâm hiển thị danh sách sản phẩm dưới dạng lưới. Người dùng có thể cuộn trang để xem thêm sản phẩm, nhấp vào từng sản phẩm để xem chi tiết hoặc sử dụng bộ lọc để thu hẹp phạm vi tìm kiếm theo nhu cầu.

The screenshot shows a product detail page for a diamond necklace. At the top, there's a navigation bar with the Mira Jewelry logo, a search bar, and user icons. The main content area features a large image of a woman wearing the necklace. To the right, the product title is "Bộ Trang Sức Cổ Điển". Below it is the original price "750.000.000 VND" and the discounted price "600.000.000 VND" in red. A detailed description box contains information about the weight (7.700 ct), main stone type (Cushion Cut), and brand (Mira). A "Thêm giỏ hàng" button is at the bottom left. Below this, a section titled "Các phẩm tương tự" displays four similar products with images and details.

Tên Sản Phẩm	Giá
Bộ Trang Sức Cổ Điển	700.000.000 VND
Bộ Trang Sức Thiên Thần	600.000.000 VND
Bộ Trang Sức Thiên Nhiên Dịu Dàng	500.000.000 VND
Bộ Ngọc Trai Baroque Vàng	300.000.000 VND

The footer contains the Mira Jewelry logo, address (Đường Nguyễn Thị Thành, Vĩnh Long), phone number (0213 123 456), email (support@mirajewelry.com), and copyright information (© 2025 Mira Jewelry). It also includes a note about the project: "Đề tài đồ án cơ sở ngành: Phát triển website tĩnh bán trang sức với dữ liệu di động từ JSON" and "Sinh viên thực hiện: Trần Thị Yến Khoa - 110123020".

Hình 4.4 Giao diện chi tiết sản phẩm

Trong giao diện chi tiết sản phẩm, khu vực bên trái hiển thị hình ảnh sản phẩm, giúp người dùng quan sát chi tiết thiết kế trang sức. Bên phải trình bày đầy đủ thông tin sản phẩm bao gồm tên sản phẩm, giá bán, thông tin khuyến mãi và các thông số kỹ thuật. Phía dưới là phần mô tả sản phẩm, giúp người dùng hiểu rõ hơn về giá trị và mục đích sử dụng. Nút thêm vào giỏ hàng cho phép người dùng lưu sản phẩm để tiếp tục mua sắm hoặc xem sản phẩm hiện có trong giỏ. Ngoài ra, trang còn hiển thị sản phẩm tương tự, hỗ trợ tương đối để người dùng tham khảo thêm các lựa chọn khác.

← Quay lại

GIỎ HÀNG (5 SẢN PHẨM)

	Bông tai Vàng 18K Đính ngọc trai Chất liệu: Vàng Kích thước: 54 Xóa	17.000.000đ	<input type="button" value="-"/> <input type="button" value="1"/> <input type="button" value="+"/>
	Vòng tay Vàng 18K Chất liệu: Vàng Kích thước: 54 Xóa	30.000.000đ	<input type="button" value="-"/> <input type="button" value="1"/> <input type="button" value="+"/>
	Bộ Trang Sức Vàng Vàng Hiện Đại Chất liệu: Vàng Kích thước: 54 Xóa	250.000.000đ	<input type="button" value="-"/> <input type="button" value="1"/> <input type="button" value="+"/>
	Bộ sưu tập Tuyết Liên Hoa Chất liệu: Diamond Kích thước: 54 Xóa	850.000.000đ	<input type="button" value="-"/> <input type="button" value="1"/> <input type="button" value="+"/>
	Nhẫn Bạc đính đá Chất liệu: Bạc Kích thước: 54 Xóa	435.000đ	<input type="button" value="-"/> <input type="button" value="1"/> <input type="button" value="+"/>

Tổng tiền: 1.147.435.000 VND

Trang chủ Sản phẩm

MIRA JEWELRY Thời trang trang sức cao cấp
Địa chỉ: Đường Nguyễn Thiện Thành, Vĩnh Long
Hotline: 0213 123 456
Email: support@mirajewelry.com
© 2025 Mira Jewelry

Đề tài đồ án cơ sở ngành: Phát triển website tĩnh bán trang sức với dữ liệu di động từ JSON
Sinh viên thực hiện: Trần Thị Yến Khoa - 110123020.

Hình 4.5 Giao diện giỏ hàng

Hình trên minh họa giao diện trang giỏ hàng của website, nơi hiển thị danh sách các sản phẩm mà người dùng đã chọn. Mỗi sản phẩm bao gồm hình ảnh, tên, thông tin cơ bản, giá bán và các nút điều chỉnh số lượng hoặc xóa sản phẩm khỏi giỏ hàng. Phần cuối trang hiển thị tổng tiền, giúp người dùng kiểm soát chi phí trước khi tiến hành các bước mua sắm tiếp theo.

Sau quá trình thực hiện đồ án, đã hoàn thành xây dựng hệ thống website tịnh bán trang sức với dữ liệu JSON, có đầy đủ các chức năng cơ bản như đăng nhập, đăng ký tài khoản, hiện sản phẩm, xem chi tiết, tìm kiếm, lọc sản phẩm và thêm vào giỏ hàng.

4.2 . Đánh giá về hiệu năng và trải nghiệm của người dùng

Giao diện trang web đơn giản, hiện đại, dễ sử dụng, bố cục rõ ràng. Website hiển thị linh hoạt trên nhiều thiết bị nhờ vào Bootstrap 5.

Website có tốc độ tải nhanh do không có sử dụng cơ sở dữ liệu hay Backend phức tạp. Dữ liệu JSON được triển khai và xử lý nhanh chóng bằng Fetch API.

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Những kết quả đạt được

Hoàn thành xây dựng website với đầy đủ tính năng và giao diện trực quan, dễ sử dụng. Tối ưu hóa giao diện, giúp người dùng dễ dàng làm quen và tìm kiếm thông tin. Đặc biệt, trang web có khả năng hiển thị linh hoạt (responsive), đảm bảo trải nghiệm tốt nhất trên mọi nền tảng thiết bị.

Thông tin sản phẩm được trình bày trực quan, cho phép người dùng tìm hiểu chi tiết nhanh chóng mà không cần tốn nhiều công sức. Kết hợp bộ lọc và công cụ tìm kiếm thông minh không chỉ tiết kiệm thời gian lựa chọn mà còn giúp khách hàng dễ dàng chọn sản phẩm ưng ý.

Website tích hợp đầy đủ chức năng quản lý tài khoản như đăng xuất và đăng nhập, giúp trang web mô phỏng các chức năng như website động. Cùng với đó là khả năng lưu trữ dữ liệu giỏ hàng tự động thông qua LocalStorage, giúp người dùng thoải mái thao tác hoặc tải lại trang mà vẫn không bị mất sản phẩm đã chọn.

Tóm lại, website không chỉ có giao diện thân thiện, dễ sử dụng, bố cục rõ ràng phù hợp đa thiết bị. Mà còn có khả năng ứng dụng thực tiễn, tạo thuận lợi cho việc phát triển và mở rộng thêm các chức năng nâng cao hơn tương lai.

5.2. Hạn chế và hướng phát triển

Đề tài đã thực hiện được đầy đủ các mục tiêu đã phân tích, tuy nhiên website vẫn còn một số hạn chế và có thể được phát triển trong tương lai. Hướng phát triển có thể thực hiện như sau:

Lập trình thêm cơ sở dữ liệu và Backend với dữ liệu thực như MySQL để quản lý dữ liệu sản phẩm và đơn hàng hiệu quả hơn.

Bổ sung chức năng đặt hàng, thanh toán để giống một website thương mại điện tử như Shoppe, Tiki,...

Nâng cao trải nghiệm khách hàng thông qua tối ưu giao diện, bổ sung các hiệu ứng tương tác.

DANH MỤC TÀI LIỆU THAM KHẢO

- [1] DataReportal – Global Digital Insights, “*Digital 2024: Global Overview Report*”, <https://datareportal.com/reports/digital-2024-global-overview-report> [Truy cập: 20/11/2025].
- [2] MDN Web Docs, “*HTML: HyperText Markup Language*”, <https://developer.mozilla.org/en-US/docs/Web/HTML>. [Truy cập: 19/11/2025].
- [3] W3Schools, “*Introduction to HTML*”, https://www.w3schools.com/html/html_intro.asp. [Truy cập: 19/11/2025].
- [4] MDN Web Docs “*CSS: Cascading Style Sheets*”, <https://developer.mozilla.org/en-US/docs/Web/CSS> [Truy cập 19/11/2025].
- [5] MDN Web Docs, “*Introduction - JavaScript*”, <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Introduction> [Truy cập 19/11/2025].
- [6] Bootstrap 5, “*Grid system*”, <https://getbootstrap.com/docs/5.3/layout/grid/>, [Truy cập 03/01/2026].
- [7] Font Awesome, “*Font Awesome Icons*”, <https://fontawesome.com> [Truy cập 03/01/2026].
- [8] Typicode, “*json-server*”, <https://github.com/typicode/json-server> [Truy cập: 03/01/2026]
- [9] MDN Web Docs, “*Fetch API - Web APIs*”, https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API [Truy cập 07/11/2025].