BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC GIA ĐỊNH**

**KHOA: CÔNG NGHỆ THÔNG TIN**

Logo, company name

Description automatically generated

**TIỂU LUẬN**

**XÂY DỰNG TRANG WEB BÁN HÀNG ĐỒNG HỒ TRỰC TUYẾN SỬ DỤNG VUE3**

**MÔN: CÔNG CỤ VÀ MÔI TRƯỜNG PHÁT TRIỂN ỨNG DỤNG**

Ngành: **KỸ THUẬT PHẦN MỀM**

Giảng viên hướng dẫn: **CHÂU TRẦN TRÚC LY**

Sinh viên thực hiện: **VŨ ĐỨC THỊNH**

MSSV: **22140459**

Lớp học phần:**221402**

TP. Hồ Chí Minh, tháng 11 năm 2024

**Khoa/Viện: Công nghệ thông tin**

**NHẬN XÉT VÀ CHẤM ĐIỂM CỦA GIẢNG VIÊN**

**TIỂU LUẬN MÔN: CÔNG CỤ VÀ MÔI TRƯỜNG PHÁT TRIỂN ỨNG DỤNG**

1. **Họ và tên sinh viên: Vũ Đức Thịnh**
2. **Tên đề tài**: Xây Dựng Trang Web Bán Hàng Đồng Hồ Trực Tuyến Sử Dụng Vue3
3. **Nhận xét**:

***Những kết quả đạt được:***

***Những hạn chế:***

1. **Điểm đánh giá** *(theo thang điểm 10, làm tròn đến 0.5):*

Sinh viên:…………………………………………………………………………….

Điểm số: ……….…… Điểm chữ: …………………………………………………

|  |  |
| --- | --- |
|  | *TP. HCM, ngày … tháng 11 năm 2024*  **Giảng viên chấm thi**  *(Ký và ghi rõ họ tên)* |

**LỜI CAM ĐOAN**

Em xin cam đoan đề tài tiểu luận: “Xây Dựng Trang Web Bán Hàng Đồng Hồ Trực Tuyến Sử Dụng Vue3” do Vũ Đức Thịnh tìm hiểu và thực hiện.

Em đã kiểm tra dữ liệu theo quy định hiện hành.

Kết quả bài làm của đề tài “Xây Dựng Trang Web Bán Hàng Đồng Hồ Trực Tuyến Sử Dụng Vue3” là trung thực và không sao chép từ bất kì bài tập của các cá nhân khác.

*TP. HCM, …. tháng 11 năm 2024*

**Sinh viên cam đoan**

(Ký và ghi rõ họ tên)

**LỜI CẢM ƠN**

Lời đầu tiên, em xin được gửi lời cảm ơn chân thành nhất đến cô Châu Trần Trúc Ly. Trong quá trình học tập và tìm hiểu môn “Xây Dựng Trang Web Bán Hàng Đồng Hồ Trực Tuyến Sử Dụng Vue3”, em đã nhận được rất nhiều sự quan tâm, giúp đỡ, hướng dẫn tâm huyết và tận tình của cô. Cô đã giúp em tích lũy thêm nhiều kiến thức về môn học này để có thể hoàn thành được bài tiểu luận về đề tài “Xây Dựng Trang Web Bán Hàng Đồng Hồ Trực Tuyến Sử Dụng Vue3”.

Trong quá trình làm bài chắc chắn khó tránh khỏi những thiếu sót. Do đó, em kính mong nhận được những lời góp ý của cô để bài tiểu luận của em ngày càng hoàn thiện hơn.

Em xin chân thành cảm ơn!

MỤC LỤC

[**CHƯƠNG 1. CƠ SỞ LÝ THUYẾT 6**](#_Toc182873181)

[**1.1. Giới thiệu tổng quan về ngôn ngữ và framework 6**](#_Toc182873182)

[**1.1.1. Ngôn ngữ Javascript 6**](#_Toc182873183)

[**1.1.2. Framework VueJS 6**](#_Toc182873184)

[**1.1.3. Giới thiệu về JSON 7**](#_Toc182873185)

[**CHƯƠNG 2. PHÂN TÍCH YÊU CẦU VÀ MÔ TẢ HỆ THỐNG 9**](#_Toc182873186)

[**2.1. Mô tả hệ thống 9**](#_Toc182873187)

[**2.1.1. Kiến trúc hệ thống 9**](#_Toc182873188)

[**2.1.2. Chức năng của hệ thống 10**](#_Toc182873189)

[**2.1.3. Công nghệ sử dụng 10**](#_Toc182873190)

[**2.2. Yêu cầu chức năng. 11**](#_Toc182873191)

[**2.3. Yêu cầu phi chức năng. 12**](#_Toc182873192)

[**CHƯƠNG 3. THIẾT KẾ KIẾN TRÚC HỆ THỐNG 13**](#_Toc182873193)

[**3.1. Sơ đồ use case của chương trình 13**](#_Toc182873194)

[**3.2. Các components của chương trình 14**](#_Toc182873195)

[**3.2.1. Giới thiệu về các components 14**](#_Toc182873196)

[**3.2.2. Hướng đi của các component 14**](#_Toc182873197)

[**3.3. Cơ sở dữ liệu JSON-SERVER 14**](#_Toc182873198)

[**3.3.1. Bảng products 14**](#_Toc182873199)

[**3.3.2. Bảng cart 15**](#_Toc182873200)

[**3.3.3. Bảng orders 15**](#_Toc182873201)

[**3.4. Cài đặt và kiểm thử chương trình 16**](#_Toc182873202)

[**3.4.1. Giao diện của chương trình 16**](#_Toc182873203)

[**CHƯƠNG 4. TỔNG KẾT 20**](#_Toc182873204)

[**4.1. Kết quả đạt được 20**](#_Toc182873205)

[**4.2. Đánh giá về đề tài 21**](#_Toc182873206)

[**4.2.1. Ưu điểm của đề tài 21**](#_Toc182873207)

[**4.2.2. Khuyết điểm của đề tài 21**](#_Toc182873208)

[**4.2.3. Hướng phát triển 22**](#_Toc182873209)

[**TÀI LIỆU THAM KHẢO 23**](#_Toc182873210)

# CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

## ****Giới thiệu tổng quan về ngôn ngữ và framework****

* + 1. **Ngôn ngữ Javascript**

JavaScript là một ngôn ngữ lập trình bậc cao, động và hướng đối tượng, chủ yếu được sử dụng để phát triển ứng dụng web và tạo trang web tương tác. Ra đời vào năm 1995, JavaScript cho phép lập trình viên tạo nội dung động mà không cần tải lại trang, xử lý sự kiện từ người dùng, và tương tác với API để xây dựng các ứng dụng phức tạp. Với sự phát triển của các framework như React, Angular và Vue.js, JavaScript đã trở thành một trong những ngôn ngữ chính trong phát triển ứng dụng web hiện đại. Cộng đồng lớn và hệ sinh thái phong phú của JavaScript cung cấp nhiều thư viện và công cụ hỗ trợ, giúp lập trình viên dễ dàng phát triển và triển khai các ứng dụng.

JavaScript là một ngôn ngữ lập trình mạnh mẽ và linh hoạt, cung cấp nhiều chức năng và tiện ích hữu ích cho lập trình viên trong việc phát triển ứng dụng web. Đầu tiên, nó cho phép tạo nội dung động, giúp thay đổi các phần tử trên trang mà không cần tải lại, từ đó tạo ra trải nghiệm người dùng mượt mà. JavaScript cũng hỗ trợ xử lý sự kiện, cho phép lập trình viên phản hồi nhanh chóng trước các hành động của người dùng như nhấp chuột hay gõ phím, rất quan trọng cho các ứng dụng tương tác. Thêm vào đó, ngôn ngữ này có khả năng tương tác với API, giúp tích hợp các dịch vụ bên ngoài vào ứng dụng, và hỗ trợ xây dựng các ứng dụng một trang (SPA) với hiệu suất cao. JavaScript còn cung cấp các phương thức để xử lý dữ liệu, quản lý các tác vụ bất đồng bộ một cách hiệu quả và tương thích trên nhiều nền tảng và trình duyệt khác nhau. Cuối cùng, với một cộng đồng lớn và phong phú thư viện, JavaScript không chỉ giúp lập trình viên dễ dàng tìm kiếm giải pháp cho các vấn đề mà còn tăng tốc quá trình phát triển ứng dụng. Tóm lại, JavaScript là công cụ không thể thiếu cho việc xây dựng các sản phẩm web chất lượng cao, đáp ứng nhu cầu ngày càng cao của người dùng.

* + 1. **Framework VueJS**

Vue.js là một framework JavaScript nổi bật, được thiết kế để xây dựng giao diện người dùng (UI) và các ứng dụng web một cách dễ dàng và hiệu quả. Được phát triển bởi Evan You và ra mắt vào năm 2014, Vue.js nhanh chóng thu hút sự chú ý của cộng đồng lập trình viên nhờ vào tính đơn giản, linh hoạt và khả năng mở rộng của nó. Một trong những điểm mạnh của Vue.js là kiến trúc thành phần (component-based architecture), cho phép lập trình viên xây dựng các ứng dụng phức tạp bằng cách chia nhỏ chúng thành các thành phần độc lập và tái sử dụng được, từ đó cải thiện khả năng bảo trì mã nguồn và tăng cường tính tổ chức của dự án. Vue.js hỗ trợ hai cách tiếp cận trong việc xây dựng giao diện, giúp lập trình viên linh hoạt hơn trong việc lựa chọn phương pháp phát triển phù hợp. Hệ thống phản ứng (reactivity system) của Vue.js cho phép tự động cập nhật giao diện khi dữ liệu thay đổi, giúp giảm thiểu việc quản lý trạng thái phức tạp. Thêm vào đó, Vue Router và Vuex cung cấp các công cụ hữu ích cho việc điều hướng và quản lý trạng thái toàn cục trong ứng dụng. Cộng đồng Vue.js rất năng động và hỗ trợ, với nhiều tài liệu và plugin có sẵn, giúp lập trình viên dễ dàng tìm kiếm giải pháp cho các vấn đề mà họ gặp phải. Nhờ vào những ưu điểm này, Vue.js đã trở thành một trong những framework phổ biến nhất trong phát triển web hiện đại, được sử dụng bởi nhiều công ty lớn và dự án nổi bật trên toàn cầu.

* + 1. **Giới thiệu về JSON**

JSON (JavaScript Object Notation) là một định dạng dữ liệu nhẹ, được thiết kế để dễ đọc và dễ viết cho con người, đồng thời cũng dễ dàng được phân tích và xử lý bởi máy tính. Nó được phát triển dựa trên cú pháp của JavaScript, nhưng hiện nay đã trở thành một tiêu chuẩn phổ biến trong nhiều ngôn ngữ lập trình khác nhau như Python, Java, C#, và Ruby. JSON chủ yếu được sử dụng để trao đổi dữ liệu giữa máy khách và máy chủ trong các ứng dụng web, nhờ vào khả năng truyền tải thông tin một cách hiệu quả.

Cấu trúc của JSON rất đơn giản, bao gồm các cặp khóa-giá trị. Một đối tượng JSON được biểu diễn bằng dấu ngoặc nhọn {}, trong khi mảng được biểu diễn bằng dấu ngoặc vuông []. Mỗi khóa trong một đối tượng phải là một chuỗi, trong khi giá trị có thể là chuỗi, số, boolean, mảng hoặc thậm chí là một đối tượng khác.

Một trong những lý do chính khiến JSON trở thành định dạng phổ biến là tính dễ đọc và dễ viết của nó, giúp lập trình viên dễ dàng tạo ra và hiểu dữ liệu. Ngoài ra, JSON cũng hỗ trợ các kiểu dữ liệu cơ bản, cho phép người dùng biểu diễn hầu hết các cấu trúc dữ liệu phức tạp một cách trực quan và gọn gàng.

JSON thường được sử dụng trong các API (Application Programming Interface) để gửi và nhận dữ liệu giữa các ứng dụng. Điều này giúp việc phát triển và tích hợp hệ thống trở nên dễ dàng hơn, đặc biệt trong các ứng dụng web hiện đại, nơi mà việc trao đổi dữ liệu là rất quan trọng. Nhờ vào những ưu điểm này, JSON đã trở thành một tiêu chuẩn de facto cho việc truyền tải dữ liệu trên web, góp phần làm cho việc phát triển ứng dụng trở nên nhanh chóng và hiệu quả hơn..

# CHƯƠNG 2. PHÂN TÍCH YÊU CẦU VÀ MÔ TẢ HỆ THỐNG

## ****2.1. Mô tả hệ thống****

Hệ thống được xây dựng là một ứng dụng thương mại điện tử sử dụng công nghệ Vue.js, với mục tiêu cung cấp cho người dùng một trải nghiệm mua sắm trực tuyến tiện lợi và hiệu quả. Ứng dụng cho phép người dùng duyệt qua các sản phẩm, thêm sản phẩm vào giỏ hàng, thanh toán đơn hàng và theo dõi trạng thái đơn hàng. Dưới đây là mô tả chi tiết về các thành phần và chức năng của hệ thống.

**2.1.1. Kiến trúc hệ thống**

Hệ thống được xây dựng theo kiến trúc phân lớp, bao gồm các thành phần chính như sau:

Giao diện người dùng (Frontend): Sử dụng Vue.js để xây dựng giao diện người dùng tương tác. Các thành phần giao diện bao gồm:

ProductList.vue: Hiển thị danh sách sản phẩm.

ProductView.vue: Hiển thị chi tiết sản phẩm.

ComCart.vue: Quản lý giỏ hàng của người dùng.

ComPay.vue: Xử lý thanh toán đơn hàng.

ComInvoice.vue: Hiển thị thông tin hóa đơn.

ComInvoiceDetailInvoice.vue: Hiển thị chi tiết hoá đơn

ComContact.vue: Cung cấp thông tin liên hệ.

Quản lý trạng thái (State Management): Sử dụng Vuex để quản lý trạng thái của ứng dụng, bao gồm các module như:

cart.js: Quản lý giỏ hàng, bao gồm các hành động như thêm, xóa sản phẩm và khởi tạo giỏ hàng từ localStorage.

products.js: Quản lý danh sách sản phẩm, bao gồm các hành động như lấy danh sách sản phẩm từ API và tìm kiếm sản phẩm.

order.js: Quản lý đơn hàng, bao gồm việc tạo đơn hàng và lưu trữ thông tin đơn hàng.

Router: Sử dụng Vue Router để điều hướng giữa các trang trong ứng dụng. Các tuyến đường chính bao gồm:

Trang chủ (hiển thị danh sách sản phẩm)

Trang giỏ hàng

Trang thanh toán

Trang chi tiết sản phẩm

Trang liên hệ

Trang hoá đơn

Trang chi tiết hoá đơn

**2.1.2. Chức năng của hệ thống**

Hệ thống cung cấp các chức năng chính sau:

Duyệt sản phẩm: Người dùng có thể xem danh sách sản phẩm với các tùy chọn tìm kiếm và sắp xếp theo giá hoặc tên sản phẩm.

Chi tiết sản phẩm: Người dùng có thể nhấp vào từng sản phẩm để xem chi tiết, bao gồm thông tin về giá cả, mô tả, hình ảnh và số lượng có sẵn.

Quản lý giỏ hàng: Người dùng có thể thêm sản phẩm vào giỏ hàng và xóa sản phẩm không mong muốn.

Thanh toán: Hệ thống cho phép người dùng thực hiện thanh toán cho đơn hàng của họ, bao gồm việc nhập thông tin giao hàng và xác nhận đơn hàng.

Theo dõi đơn hàng: Người dùng có thể theo dõi trạng thái đơn hàng của mình sau khi thực hiện thanh toán.

**2.1.3. Công nghệ sử dụng**

Hệ thống được xây dựng dựa trên các công nghệ sau:

Vue.js: Thư viện JavaScript để xây dựng giao diện người dùng.

Vuex: Thư viện quản lý trạng thái cho Vue.js.

Vue Router: Thư viện để quản lý điều hướng trong ứng dụng Vue.js.

Axios: Thư viện để thực hiện các yêu cầu HTTP tới API.

LocalStorage: Sử dụng để lưu trữ giỏ hàng và thông tin đơn hàng tạm thời trên trình duyệt.

JSON Server:  Tạo một REST API giả lập từ một file JSON

Hệ thống được thiết kế để có khả năng mở rộng và bảo trì dễ dàng, với mã nguồn rõ ràng và có cấu trúc tốt. Điều này giúp cho việc phát triển thêm các tính năng mới trong tương lai trở nên thuận lợi hơn.

## ****2.2. Yêu cầu chức năng.****

Giao diện hợp lý và dễ sử dụng: Website cần được thiết kế với một bố cục rõ ràng và hợp lý, giúp người dùng dễ dàng tương tác với các chức năng mà không gặp phải bất kỳ khó khăn nào. Các trang quan trọng như trang chủ, danh sách sản phẩm, giỏ hàng và trang thanh toán nên được sắp xếp một cách khoa học, dễ nhìn và dễ sử dụng. Khi bố cục được tối ưu hóa, người dùng có thể nhanh chóng tìm thấy sản phẩm mình cần, truy cập vào giỏ hàng hoặc tiến hành thanh toán mà không phải lăn tăn. Điều này không chỉ giúp tiết kiệm thời gian mà còn nâng cao trải nghiệm mua sắm trực tuyến, khiến khách hàng cảm thấy thoải mái và hài lòng hơn khi duyệt web. Một website thân thiện với người dùng chính là chìa khóa để giữ chân khách hàng và tăng cường doanh thu.

Responsive Design: Giao diện của một website ngày nay cần phải linh hoạt và thích ứng với nhiều loại thiết bị khác nhau, từ máy tính để bàn cho đến điện thoại di động và máy tính bảng. Điều này có nghĩa là các thành phần trên giao diện phải tự động điều chỉnh kích thước và vị trí để phù hợp với màn hình của từng thiết bị. Nhờ vậy, người dùng sẽ luôn có trải nghiệm mượt mà và dễ chịu, với nội dung được hiển thị rõ ràng và các nút điều khiển dễ dàng thao tác, bất kể họ đang sử dụng thiết bị nào. Việc tối ưu hóa giao diện không chỉ giúp tăng cường sự hài lòng của người dùng mà còn góp phần nâng cao hiệu quả tương tác và chuyển đổi trên website.

Tái sử dụng component: Khi sử dụng Vue 3, việc áp dụng các thành phần (component) tái sử dụng là một trong những bí quyết giúp quản lý và phát triển giao diện trở nên dễ dàng và hiệu quả hơn bao giờ hết. Các component như danh sách sản phẩm, thẻ sản phẩm hay nút thêm vào giỏ hàng có thể được thiết kế một cách linh hoạt và tái sử dụng ở nhiều vị trí khác nhau trong ứng dụng. Điều này không chỉ giúp giảm đáng kể thời gian phát triển và bảo trì, mà còn nâng cao hiệu suất của toàn bộ ứng dụng. Thay vì phải viết lại mã cho mỗi phần tương tự, chúng ta chỉ cần tạo ra một component duy nhất và sử dụng nó ở bất kỳ đâu. Kết quả là, chúng ta có một giao diện nhất quán, dễ bảo trì và có thể dễ dàng mở rộng khi cần thiết. Chẳng hạn, một nút thêm vào giỏ hàng chỉ cần thay đổi một chút là đã có thể trở nên hoàn toàn mới mẻ cho từng sản phẩm.

## ****2.3. Yêu cầu phi chức năng.****

Hiệu suất : Ứng dụng phải đảm bảo thời gian phản hồi nhanh cho các yêu cầu của người dùng, đặc biệt là khi tải danh sách sản phẩm, thêm sản phẩm vào giỏ hàng, và thực hiện thanh toán. Các thao tác này cần được tối ưu hóa để không làm gián đoạn trải nghiệm người dùng.

Khả năng mở rộng : Hệ thống cần có khả năng mở rộng để xử lý một lượng người dùng lớn hơn và số lượng sản phẩm nhiều hơn mà không làm giảm hiệu suất. Điều này bao gồm khả năng mở rộng của Vuex để quản lý trạng thái khi số lượng sản phẩm và giỏ hàng tăng lên.

Bảo trì : Mã nguồn cần được tổ chức tốt với cấu trúc rõ ràng, giúp dễ dàng bảo trì và cập nhật trong tương lai. Việc sử dụng Vuex cho quản lý trạng thái và Vue Router cho điều hướng giúp cho mã nguồn trở nên dễ hiểu và dễ bảo trì hơn.

Tính khả dụng : Giao diện người dùng cần thân thiện và dễ sử dụng, giúp người dùng dễ dàng tìm kiếm sản phẩm, thêm vào giỏ hàng, và thực hiện thanh toán mà không gặp khó khăn. Các thành phần giao diện như ProductList.vue, ProductView.vue, và ComCart.vue cần được thiết kế để tối ưu hóa trải nghiệm người dùng.

# CHƯƠNG 3. THIẾT KẾ KIẾN TRÚC HỆ THỐNG

## ****3.1. Sơ đồ use case của chương trình****

A diagram of a person with text

Description automatically generated

Hình 3.1.1. Sơ đồ use case Bán hàng

## ****3.2. Các components của chương trình****

**3.2.1. Giới thiệu về các components**

ProductList.vue: Hiển thị danh sách sản phẩm.

ProductView.vue: Hiển thị chi tiết sản phẩm.

ComCart.vue: Quản lý giỏ hàng của người dùng.

ComPay.vue: Xử lý thanh toán đơn hàng.

ComInvoice.vue: Hiển thị thông tin hóa đơn.

ComInvoiceDetailInvoice.vue: Hiển thị chi tiết hoá đơn

ComContact.vue: Cung cấp thông tin liên hệ.

**3.2.2. Hướng đi của các component**

A diagram of a product

Description automatically generated

Hình 3.2.2.1. Hướng đi của các components

## ****3.3. Cơ sở dữ liệu JSON-SERVER****

**3.3.1. Bảng products**

Bảng products lưu trữ thông tin về các sản phẩm như id, tên sản phẩm, mô tả, giá, số lượng tồn kho, hình ảnh. Đóng vai trò quan trọng trong việc quản lý dữ liệu của sản phẩm.

|  |  |
| --- | --- |
| **Tên cột** | **Mô tả** |
| id | ID của sản phẩm |
| name | Tên của sản phẩm |
| price | Giá của sản phẩm |
| description | Mô tả của sản phẩm |
| stock | Số lượng tồn kho |
| img | Đường dẫn đến ảnh của sản phẩm |

*Bảng 3.3.1.1. Bảng products*

**3.3.2. Bảng cart**

Bảng cart (giỏ hàng) bao gồm thông tin như ID giỏ hàng, ID người dùng, danh sách sản phẩm (gồm ID, tên, hình ảnh, giá, số lượng và tổng giá), tổng giá giỏ hàng, thời gian tạo và trạng thái giỏ hàng. Bảng cart hỗ trợ các chức năng như thêm, cập nhật, xóa sản phẩm và thanh toán. Nó cũng kết nối với các thành phần khác như quản lý sản phẩm và đơn hàng.. Nó đóng vai trò như một trung gian giữa người dùng và quy trình đặt hàng.

|  |  |
| --- | --- |
| **Tên cột** | **Mô tả** |
| id | ID của sản phẩm trong giỏ hàng |
| productId | ID của sản phẩm |
| name | Tên của sản phẩm |
| price | Giá của sản phẩm |
| image | Đường dẫn ảnh |
| quantity | Số lượng trong giỏ hàng |

*Bảng 3.3.2.1. Bảng cart*

**3.3.3. Bảng orders**

Bảng order (đơn hàng) hỗ trợ các chức năng như lấy thông tin đơn hàng, cập nhật trạng thái đơn hàng, và quản lý lịch sử đơn hàng của khách hàng. Bảng này cũng kết nối với các thành phần khác trong hệ thống thương mại điện tử, chẳng hạn như quản lý sản phẩm và quản lý người dùng.

|  |  |
| --- | --- |
| **Tên cột** | **Mô tả** |
| id | ID của đơn hàng |
| customerName | Tên khách hàng |
| customerPhone | Số điện thoại |
| customerAddress | Địa chỉ |
| paymentMethod | Phương thức thanh toán |
| products | Bao gồm từng sản phẩm trong giỏ hàng (Mỗi sản phẩm đều có id của sản phẩm, trạng thái đặt hàng, đặt hàng vào ngày, phương thức thanh toán,…) |
| totalPrice | Tổng tiền |

Bảng 3.3.3.1. Bảng order

## ****3.4. Cài đặt và kiểm thử chương trình****

3.4.1. Giao diện của chương trình

A screenshot of a computer

Description automatically generated

Hình 3.4.1.1. Trang chủ của chương trình

A screenshot of a watch

Description automatically generated

Hình 3.4.1.2. Trang thông tin sản phẩm

A screenshot of a computer

Description automatically generated

*Hình 3.4.1.3. Trang giỏ hàng của người dùng*

*A screenshot of a computer

Description automatically generated*

*Hình 3.4.1.4. Trang đặt hàng*

*A screenshot of a computer

Description automatically generated*

*Hình 3.4.1.5. Trang hoá đơn*

*A screenshot of a computer

Description automatically generated*

*Hình 3.4.1.6. Trang thông tin chi tiết hoá đơn*

# CHƯƠNG 4. TỔNG KẾT

## ****4.1. Kết quả đạt được****

Sau quá trình phát triển ứng dụng thương mại điện tử, chúng tôi đã đạt được một số kết quả quan trọng như sau:

Kiến trúc kỹ thuật: Dự án đã thành công trong việc phát triển ứng dụng sử dụng Vue 3, một framework JavaScript hiện đại và mạnh mẽ. Việc sử dụng Vue 3 với Composition API đã mang lại những ưu điểm vượt trội trong quản lý state và logic của ứng dụng. Kiến trúc ứng dụng được xây dựng theo mô hình component-based, cho phép tái sử dụng và modular hóa các thành phần giao diện một cách linh hoạt và hiệu quả.

Các nguyên tắc quản lý state tiên tiến được áp dụng, giúp việc quản lý dữ liệu trở nên minh bạch và dễ dàng theo dõi. Mỗi component có thể độc lập về logic và trạng thái, đồng thời vẫn có thể giao tiếp và chia sẻ dữ liệu một cách mượt mà thông qua Vuex.

Quản lý sản phẩm: Hệ thống quản lý sản phẩm được xây dựng một cách chuyên nghiệp. Ứng dụng có khả năng hiển thị danh sách đồng hồ đa dạng với các thông tin chi tiết như hình ảnh, giá cả, mô tả sản phẩm. Chức năng tìm kiếm và lọc sản phẩm được tích hợp thông minh, cho phép người dùng dễ dàng tìm kiếm sản phẩm theo nhiều tiêu chí như giá, a-z, tên đồng hồ. Giao diện chi tiết sản phẩm được thiết kế tỉ mỉ, cung cấp đầy đủ thông tin để người dùng có thể đưa ra quyết định mua hàng một cách chính xác. Mỗi sản phẩm được trình bày với các thông số kỹ thuật, hình ảnh chất lượng cao và các tùy chọn liên quan.

Giỏ hàng: Chức năng giỏ hàng được triển khai với các tính năng linh hoạt và thuận tiện. Người dùng có thể dễ dàng thêm hoặc xóa sản phẩm, điều chỉnh số lượng một cách nhanh chóng. Cơ chế lưu trữ cục bộ (local storage) được tích hợp, đảm bảo trạng thái giỏ hàng được duy trì ngay cả khi người dùng làm mới trang hoặc đóng trình duyệt.

Đặt hàng: Quy trình đặt hàng được xây dựng một cách logic và dễ sử dụng. Hệ thống quản lý thông tin đơn hàng chuyên nghiệp, cung cấp cho người dùng trải nghiệm mua sắm trực tuyến mượt mà. Sau mỗi giao dịch, kho hàng sẽ được tự động cập nhật, đảm bảo tính chính xác về số lượng sản phẩm.

Quản lý trạng thái: Vuex được sử dụng như một giải pháp quản lý state toàn cục hiệu quả. Hệ thống được phân chia thành các module riêng biệt (products, cart, order), mỗi module có trách nhiệm quản lý một phần cụ thể của ứng dụng. Điều này giúp mã nguồn trở nên sáng sủa, dễ bảo trì và mở rộng.

Điều hướng: Vue Router được triển khai để tạo ra trải nghiệm điều hướng mượt mà giữa các trang. Các luồng chuyển trang được xây dựng một cách thông minh, hỗ trợ các chuyển đổi động và mượt mà giữa các view khác nhau trong ứng dụng.

**4.2. Đánh giá về đề tài**

**4.2.1. Ưu điểm của đề tài**

Kiến trúc linh hoạt: Dự án đã áp dụng Vue 3 Composition API, cho phép xây dựng kiến trúc linh hoạt và dễ dàng mở rộng. Cấu trúc module hóa cao giúp cho việc bảo trì mã nguồn trở nên đơn giản hơn, đồng thời cho phép các nhà phát triển dễ dàng thêm mới hoặc sửa đổi các tính năng mà không làm ảnh hưởng đến các phần khác của ứng dụng.

Quản lý dữ liệu: Việc quản lý state được thực hiện một cách tập trung thông qua Vuex, giúp đồng bộ hóa dữ liệu giữa các thành phần trong ứng dụng. Điều này không chỉ giúp giảm thiểu lỗi mà còn hỗ trợ cho việc debug hiệu quả hơn, khi mà mọi thay đổi về state đều được ghi nhận và có thể theo dõi dễ dàng.

Tích hợp backend: Dự án cho phép kết nối linh hoạt với máy chủ backend, xử lý các thao tác CRUD (Create, Read, Update, Delete) một cách hiệu quả. Điều này đảm bảo rằng mọi thay đổi trong dữ liệu đều được cập nhật kịp thời và chính xác, đồng thời quản lý trạng thái đơn hàng cũng được thực hiện một cách chính xác và nhanh chóng.

Giao diện: Ứng dụng được thiết kế với bố cục logic và rõ ràng, giúp người dùng dễ dàng tìm kiếm và điều hướng. Các thành phần giao diện được sắp xếp một cách hợp lý, tạo cảm giác thân thiện và dễ sử dụng cho người dùng.

**4.2.2. Khuyết điểm của đề tài**

Giới hạn trong phạm vi dự án: Do sử dụng JSON-server, hệ thống chỉ mô phỏng backend, không xử lý được các yêu cầu phức tạp như quản lý nhiều người dùng, bảo mật tài khoản, hoặc giao dịch thực tế.

Chưa tích hợp các tính năng nâng cao: Một số tính năng hữu ích như thông báo thời gian thực (real-time notifications), theo dõi trạng thái đơn hàng, hoặc đánh giá sản phẩm vẫn chưa được triển khai.

Phụ thuộc nhiều vào trình duyệt: Việc sử dụng local storage để lưu trạng thái

**4.2.3. Hướng phát triển**

Responsive Design: Mặc dù ứng dụng đã có những cải tiến nhất định về giao diện, nhưng vẫn chưa hoàn thiện trên mọi thiết bị. Cần có thêm thời gian và công sức để cải thiện giao diện trên các thiết bị di động, đảm bảo rằng người dùng có thể trải nghiệm ứng dụng một cách tốt nhất trên mọi nền tảng.

Tối ưu hiệu năng: Việc implement code splitting và áp dụng lazy loading sẽ giúp cải thiện hiệu suất của ứng dụng. Tối ưu hóa truy vấn API cũng là một yếu tố quan trọng, giúp giảm thiểu thời gian tải trang và nâng cao trải nghiệm người dùng.

Mở rộng tính năng: Dự án có thể mở rộng tính năng bằng cách tích hợp nhiều phương thức thanh toán khác nhau, phát triển hệ thống đánh giá sản phẩm và bổ sung tính năng so sánh sản phẩm. Những cải tiến này sẽ giúp nâng cao giá trị của ứng dụng và thu hút thêm người dùng.

# ****TÀI LIỆU THAM KHẢO****

1. LÊ CHI. Năm tài liệu được tạo ra 12/10/2023. [*Javascript là gì*](https://fptshop.com.vn/tin-tuc/danh-gia/javascript-la-gi-151984)*?.* Truy cập ngày 17/11/2024.
2. TRANG VŨ. Năm tài liệu được tạo ra 08/08/2023*.* [*VueJS là gì? Ưu và nhược điểm của Vue.js so với các framework khác*](https://stringee.com/vi/blog/post/vuejs-la-gi)*.* Truy cập ngày 17/11/2024.
3. KHÁNH KIM. Năm tài liệu được tạo ra 12/09/2022. [*Json là gì? Những điều lập trình viên nên biết về Json File.*](https://fptshop.com.vn/tin-tuc/danh-gia/mysql-la-gi-157657)Truy cập ngày 17/11/2024.
4. TRẦN ĐỨC GIANG. Năm tài liệu được tạo ra 01/05/2021. [*Tổng quan về json-server?*](https://viblo.asia/p/xay-dung-json-server-voi-thu-vien-json-server-07LKX7O45V4)*.* Truy cập ngày 17/11/2024.
5. CAO THÀNH LỰC. Năm tài liệu được tạo ra 28/07/2015. [*Tổng quan về NodeJS.*](https://viblo.asia/p/tong-quan-ve-node-js-AeJ1vOdQRkby)Truy cập ngày 17/11/2024.

6. Macrae, C. (2021). Vue.js: Up and running. O'Reilly Media.