

**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

-----











-----

**BÀI TẬP LƠN MÔN PHÁT TRIỂN ỨNG DỤNG WEB**

**Mã số học phần: CT449**

**ĐỀ TÀI: WEBSITE BÁN GIÀY SỬ DỤNG CÔNG NGHỆ MEVN-STACK**

**Sinh Viên: Nguyễn Phúc Vinh**

**MSSV: C2000031**

**Cần Thơ, tháng 5 năm 2023**



**KHÓA: 46**

**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

-----











-----

**BÀI TẬP LƠN MÔN PHÁT TRIỂN ỨNG DỤNG WEB**

**Mã số học phần: CT449**

**ĐỀ TÀI: WEBSITE BÁN GIÀY SỬ DỤNG CÔNG NGHỆ MEVN-STACK**

**Cần Thơ, tháng 5 năm 2023**



**Sinh Viên Thực Hiện:**

**Nguyễn Phúc Vinh**

**MSSV: C2000031**

**Giảng Viên Hướng Dẫn:**

**ThS.Nguyễn Minh Trung**

**LỜI CẢM ƠN**

Quá trình thực hiện niên luận cơ sở là giai đoạn đầu quan trọng trong quãng đời mỗi sinh viên khoa Công Nghệ Thông Tin – Truyền Thông. Niên luận cơ sở là tiền đề nhằm trang bị cho chúng em những kỹ năng nghiên cứu, những kiến thức quý báu.

Trước hết, em xin chân thành cám ơn Trường Công Nghệ Thông Tin và Truyền Thông Trường Đại học Cần Thơ đã tạo điều kiện thuận lợi cho chúng em học tập và thực hiện đề tài này.

Em xin chân thành cảm ơn quý Thầy, Cô Khoa Công Nghệ Thông Tin và Truyền Thông Trường Đại học Cần Thơ đã tận tình chỉ dạy và trang bị cho em những kiến thức cần thiết trong suốt thời gian ngồi trên ghế giảng đường, lấy đó làm nền tảng cho em có thể hoàn thành được bài tập lớn này.

Em xin trân trọng cảm ơn thầy Nguyễn Minh Trung đã tận tình giúp đỡ, định hướng cách tư duy và cách làm việc. Đó là những góp ý hết sức quý báu không chỉ trong quá trình thực hiện niên luận mà còn là hành trang tiếp bước không thể nào quên cho em trong quá trình học tập và lập nghiệp sau này.

Và cuối cùng, xin gửi lời cảm ơn đến gia đình, bạn bè, những người luôn sẵn sàng sẻ chia và giúp đỡ trong học tập và cuộc sống. Mong rằng, chúng ta sẽ mãi mãi gắn bó với nhau.

Dù đã cố gắng hoàn thành đề tài trong phạm vi và khả năng cho phép nhưng chắc chắn sẽ không tránh khỏi những thiếu sót. Rất mong nhận được sự thông cảm, góp ý và tận tình chỉ bảo của quý Thầy Cô.

Lời cuối cùng, em xin kính chúc các thầy cô nhiều sức khỏe, thành công và hạnh phúc.

**MỤC LỤC**

[**CHƯƠNG I 1**](#_Toc134105783)

[**TỔNG QUAN 1**](#_Toc134105784)

[**1.1 MỤC TIÊU ĐỀ TÀI 1**](#_Toc134105785)

[**1.2 YÊU CẦU ĐỀ TÀI 1**](#_Toc134105786)

[**1.2.1 Yêu cầu chung 1**](#_Toc134105787)

[**1.2.2 Xây dựng website thương mại điện tử 2**](#_Toc134105788)

[**1.2.3 Các chức năng chính 2**](#_Toc134105789)

[**1.2.4 Đối tượng và phạm vi nghiên cứu 2**](#_Toc134105790)

[**1.2.5 Phương pháp nghiên cứu 3**](#_Toc134105791)

[**1.2.6 Nội dung nghiên cứu 3**](#_Toc134105792)

[**CHƯƠNG II 4**](#_Toc134105793)

[**CƠ SỞ NGHIÊN CỨU LÝ THUYẾT 4**](#_Toc134105794)

[**2.1 GIỚI THIỆU VỀ API VÀ RESTFUL API 4**](#_Toc134105795)

[**2.2 NODEJS 6**](#_Toc134105796)

[**2.3 MONGODB 8**](#_Toc134105797)

[**2.4 EXPRESSJS 10**](#_Toc134105798)

[**2.5 VueJS 12**](#_Toc134105799)

[**2.6 AXIOS 14**](#_Toc134105800)

[**CHƯƠNG III 16**](#_Toc134105801)

[**NỘI DUNG 16**](#_Toc134105802)

[**3.1 YÊU CẦU TỐI THIỂU CỦA ĐỀ TÀI 16**](#_Toc134105803)

[**3.2 XÂY DỰNG CHỨC NĂNG WEBSITE THƯƠNG MẠI ĐIỆN TỬ 16**](#_Toc134105804)

[**3.3 MÔ TẢ NGHIỆP VỤ 17**](#_Toc134105805)

[**3.3.1 Dành cho khách hàng nói chung 17**](#_Toc134105806)

[**3.3.2 Dành cho quản trị viên 18**](#_Toc134105807)

[**3.4 YÊU CẦU HỆ THỐNG 18**](#_Toc134105808)

[**3.4.1 Yêu cầu chức năng 18**](#_Toc134105809)

[**3.4.2 Yêu cầu phi chức năng 19**](#_Toc134105810)

[**3.5. CÁC SƠ ĐỒ 20**](#_Toc134105811)

[**3.5.1 Use case diagram 20**](#_Toc134105812)

[**3.5.2 Entity Diagram (sơ đồ thực thể) 22**](#_Toc134105813)

[**3.5.3 Sequence diagram (Sơ đồ tuần tự) 24**](#_Toc134105814)

[**3.5.4 Functional diagram (Sơ đồ chức năng) 26**](#_Toc134105815)

[**3.6 KIỂM THỬ 29**](#_Toc134105816)

[**3.6.1 Giao diện của khách hàng vãng lai và khách hàng thành viên. 29**](#_Toc134105817)

[**3.6.2 Đối với giao diện của Quản Trị Viên. 36**](#_Toc134105818)

[CHƯƠNG IV 38](#_Toc134105819)

[KẾT LUẬN 38](#_Toc134105820)

[4.1 KẾT QUẢ ĐẠT ĐƯỢC 38](#_Toc134105821)

[4.1.1 Ưu điểm 38](#_Toc134105822)

[4.1.2 Nhược điểm 38](#_Toc134105823)

[4.2. HƯỚNG PHÁT TRIỂN 38](#_Toc134105824)

# CHƯƠNG I

# TỔNG QUAN

## 1.1 MỤC TIÊU ĐỀ TÀI

Những tiến bộ to lớn về công nghệ hiện đại, đặc biệt là công nghệ thông tin trong những thập niên cuối của thế kỷ 20 đã tạo ra bước ngoặt mới cho sự phát triển kinh tế xã hội toàn cầu. Chính vì điều đó, một phương thức thương mại mới đã xuất hiện và phát triển nhanh chóng, đó là thương mại điện tử. Thương mại điện tử chính là một công cụ hiện đại sử dụng mạng Internet giúp cho các doanh nghiệp có thể thâm nhập và nhanh chóng phát triển vào thị trường thế giới, thu thập thông tin nhanh hơn, nhiều hơn và chính xác hơn. Với thương mại điện tử, các doanh nghiệp cũng có thể đưa các thông tin về sản phẩm của mình đến các đối tượng khách hàng tiềm năng lớn khác nhau ở mọi nơi trên thế giới với chi phí thấp hơn nhiều so với các phương pháp truyền thống.

Trong xu thế phát triển mạnh mẽ của thương mại điện tử trên thế giới, các doanh nghiệp Việt Nam đặc biệt là các doanh nghiệp vừa và nhỏ, bộ phận chiếm tỷ lệ lớn trong tổng số các doanh nghiệp Việt Nam, cũng đã bước đầu nhận thức được ích lợi và tầm quan trọng của việc ứng dụng thương mại điện tử. Tuy nhiên, do còn nhiều hạn chế trong nhận thức của bản thân các doanh nghiệp cũng như các điều kiện cơ sở hạ tầng, việc ứng dụng thương mại điện tử trong các doanh nghiệp vừa và nhỏ ở Việt Nam có thể nói mới ở mức độ sơ khởi. Vì thế, các doanh nghiệp vừa và nhỏ cần xây dựng cho mình một chiến lược kinh doanh phù hợp với điều kiện thực tế hiện nay nhằm tiếp cận nhiều hơn nữa với thương mại điện tử để có thể khai thác tối đa lợi ích mà phương thức kinh doanh này đem lại.

Ngày nay, các cửa hàng trực tuyến xuất hiện càng nhiều. Cửa hàng trực tuyến sẽ đem lại nhiều lợi ích về cho người sử dụng. Đối với doanh nghiệp, tiết kiệm được không gian trưng bày sản phẩm, tiết kiệm nguồn nhân lực, tiết kiệm chi phí mặt bằng… Đối với khách hàng, tiết kiệm được thời gian đi lại, lựa chon được những sản phẩm ưa thích…Đây cũng là lý do để em chọn đề tài: **“Xây dựng website thương mại điện tử bán giày”** làm đề tài môn học “Phát triển ứng dụng web”.

Chính vì vậy, việc xây dựng website V-SHOP được xây dựng để nắm bắt xu hướng thời đại, cũng là để doanh nghiệp tiếp cận với số lượng khách hàng mới, giúp cho các sản phẩm tiếp cận được với nhiều khách hàng, đem lại doanh thu lớn cho doanh nghiệp.

Không những thế, đây cũng là một cơ hội tốt để có thể trao dồi học hỏi những điều mới từ những kiến thức đã học và sẽ học để đủ điều kiện nâng cấp bản thân.

## 1.2 YÊU CẦU ĐỀ TÀI

### 1.2.1 Yêu cầu chung

- Tìm hiểu về cách thức hoạt động kinh doanh thương mại.

- Tìm hiểu về các giải pháp để xây dựng một website thương mại điện tử như thế nào.

- Xây dựng một website mang tính thương mại hóa.

- Xây dựng một trang web với đầy đủ các chức năng để phục vụ cho Quản Trị viên nói riêng và Khách Hàng nói chung.

- Đòi hỏi một môi trường thân thiện với người tiêu dùng.

### 1.2.2 Xây dựng website thương mại điện tử

- Webiste thương mại điện tử có một số chức năng như sau:

+ Đối với khách hàng: Xem danh sách các sản phẩm, Xem thông tin chi tiết một sản phẩm, Tìm kiếm sản phẩm, Lọc sản phẩm, Đưa sản phẩm vào giỏ hàng, Đặt mua, Thanh toán.

+ Đối với quản trị viên: Quản lý sản phẩm, Quản lý danh mục, Quản lý đơn hàng.

### 1.2.3 Các chức năng chính

- Website thời trang được xây dựng với tiêu chí phục vụ cho tất cả mọi người, trong lĩnh vực thời trang với mong muốn đáp ứng đầy đủ các mặt hàng, các chức năng của một website bán hàng trực tuyến.

- Website bao gồm các loại người dùng chính: Khách hàng viếng thăm, Khách hàng thành viên, Quản trị viên:

+ Đối với Khách hàng viếng thăm: Xem danh sách sản phẩm, Xem chi tiết sản phẩm, Tìm kiếm, Lọc, Đăng Ký…

+ Đối với Khách hàng thành viên: Ngoài những chức năng như khách hàng viếng thăm còn có thêm một số chức năng như Đưa sản phẩm vào giỏ hàng, Cập nhật giỏ hàng Đặt mua, Xem danh sách đơn hàng, Quản lý đơn hàng, Thanh Toán, Đăng nhập…

+ Quản trị viên: Được xem như là một khách hàng thành viên và còn có chức năng quản lý tùy thuộc vào chức năng như: Quản lý sản phẩm, danh mục, đơn hàng, quản lý thông tin khác.

### 1.2.4 Đối tượng và phạm vi nghiên cứu

Website thời trang được xây dựng với tiêu chí phục vụ cho tất cả mọi người có đam mê về thời trang với mong muốn đáp ứng đầy đủ các mặt hàng, các chức năng của một website bán hàng trực tuyến.

Nghiên cứu và tham khảo những trang web về thời trang nổi bật hiện nay để lấy đó làm nền tảng mở rộng kiến thức.

Nắm bắt được những điểm nổi bậc của công nghệ về thời trang hiện nay, những sản phẩm tốt và tiên quyết để làm nên thương hiệu riêng của trang web từ đó phát triển duy trì trang web một cách thuận lợi.

### 1.2.5 Phương pháp nghiên cứu

Nội dung nghiên cứu khá phổ biến và đầy đủ, do đó tài liệu nghiên cứu rất nhiều nên không tránh khỏi tình trạng không biết bắt đầu từ đâu và như thế nào nhưng chủ yếu là tìm hiểu thông tin tài liệu thông qua các video có nhiều lượt xem và đầy đủ nội dung, sách báo và các bài viết bổ ích đáp ứng đủ tiêu chí trên mạng.

Từ đó, vận dụng các kiến thức đã tiếp thu được vào đề tài. Tương ứng với mỗi thành phần của hệ thống sẽ có những phương pháp nghiên cứu khác nhau.

### 1.2.6 Nội dung nghiên cứu

|  |  |  |
| --- | --- | --- |
| STT | Đối tượng | Nội dung |
| 1 | API và RESTful API | - Tìm hiểu về API  - Rest là gì ?  - Restful API là gì ?  - Các phương thức cơ bản ?  - Nguyên tắc hoạt động của Restful API |
| 2 | NodeJS | - Tìm hiểu NodeJs  - NodeJs hoạt động như thế nào?  - Những ứng dụng sử dụng NodeJs |
| 3 | MongoDB | - Tìm hiểu hệ quản trị cơ sở dữ liệu NOSQL – MongoDB  - So sánh một số khái niệm trong MongoDB và các cơ sở dữ liệu quan hệ (RDBMS) khác  - Khi nào nên sử dụng MongoDB |
| 4 | ExpressJS | - ExpressJs là gì?  - Phân biệt giữa ExpressJs và NodeJs |
| 5 | VueJS | - Tìm hiểu VueJS  - Xử ký sự kiện trong Vue  - Ví dụ về Vue  - Ưu điểm và nhược điểm của Vue  - Dom ảo là như thế nào? |
| 6 | Axios | - Tìm hiểu về Axios  - Tại sao nên sử dụng Axios thay vì Fetch |

# CHƯƠNG II

# CƠ SỞ NGHIÊN CỨU LÝ THUYẾT

## 2.1 GIỚI THIỆU VỀ API VÀ RESTFUL API



REST (**RE**presentational **S**tate **T**ransfer) được đưa ra vào năm 2000, trong luận văn tiến sĩ của Roy Thomas Fielding (đồng sáng lập giao thức HTTP). Nó là một dạng chuyển đổi cấu trúc dữ liệu, là một phong cách kiến ​​trúc cho việc thiết kế các ứng dụng có kết nối. Nó sử dụng HTTP đơn giản để tạo cho giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, DELETE, vv đến một URL để [**xử lý dữ liệu**](https://movan.vn/quan-ly-co-du-lieu-cho-pheu-ban-hang/).

**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 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.

Vì một phần mềm chứa rất nhiều logic phức tạp, nên người ta tìm cách chia nhỏ nó ra thành nhiều phần, mỗi phần này tạm gọi là một component. Mỗi component sẽ có tính độc lập cao, ít phụ thuộc hoặc có thể không phục thuộc vào các thành phần khác. Tuy là có tính độc lập cao, nhưng để có thể kết nối được với nhau mà một phần mềm hoàn chỉnh, buộc chúng vẫn phải tuân theo một hoặc một số chuẩn nào đó. Thì mỗi cái chuẩn đó được gọi là một giao diện lập trình ứng dụng – hay chính là một API.

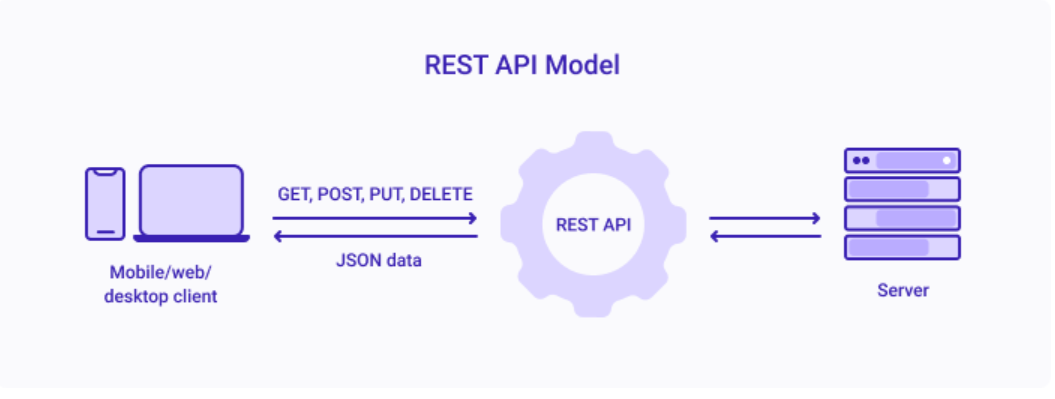
**RESTful API** (hay còn gọi là REST API) là một giao diện lập trình ứng dụng (API hay web API) tuân theo các ràng buộc của kiểu kiến trúc REST, cho phép tương tác với các dịch vụ web RESTful. Hay nói đơn giản, RESTful API là một tiêu chuẩn được dùng trong việc thiết kế API dành cho các ứng dụng web (thiết kế Web Services) để hỗ trợ cho việc quản lý các resource. **REST** là viết tắt của cụm từ **Representational state transfer** (**ứng dụng chuyển đổi cấu trúc dữ liệu**), được tạo ra lần đầu bởi nhà khoa học máy tính Roy Fielding ở những năm 2000.. REST API không được xem là một công nghệ, nó là một giải pháp để tạo ra các ứng dụng web services thay thế cho các kiểu khác như SOAP, WSDL (Web Service Definition Language),…

|  |  |
| --- | --- |
| Các phương thức HTTP | Tác vụ |
| GET | Lấy dữ liệu |
| POST | Tạo mới dữ liệu |
| PUT | Cập nhật dữ liệu (toàn bộ) |
| PATCH | Cập nhật dữ liệu (một phần) |
| DELETE | Xóa dữ liệu |

Các tác vụ đọc, tạo, cập nhật, xóa được gọi là CRUD service ( Create, Read, Update, Delete ). Mỗi tác vụ trên phải được thông qua địa chỉ URI ( Uniform Resource Identifier ) kèm theo phương thức payload ( có thể có hoặc không, thường được định dạng kiểu XML hoặc JSON).

**RESTful API** sử dụng giao thức stateless ( là một giao thức truyền thông không sử dụng session ) và theo chuẩn nên hệ thống sẽ nhanh, đáng tin cậy và có thể mở rộng dễ dàng. Thông thường, RESTful API sẽ xác thực người dùng khi gửi yêu cầu đối với những tác vụ nguy hiểm như cập nhật hoặc xóa dữ liệu chỉ cho phép người quản trị.

**RESTful API** hoạt động như thế nào?



Bước 1: Client gửi một yêu cầu đến máy chủ. Client làm theo tài liệu API để định dạng yêu cầu theo cách mà máy chủ hiểu được.

Bước 2: Máy chủ xác thực và xác nhận máy khách có quyền đưa ra yêu cầu đó.

Bước 3: Máy chủ nhận yêu cầu và xử lý trong nội bộ.

Bước 4: Máy chủ trả về một phản hồi đến client. Phản hồi chứa thông tin cho client biết liệu yêu cầu có thành công hay không. Phản hồi cũng bao gồm bất kỳ thông tin nào mà client yêu cầu.

Chi tiết về phản hồi và yêu cầu API REST sẽ khác nhau đôi chút tùy thuộc vào cách các nhà phát triển API thiết kế API.

## 2.2 NODEJS



**NodeJS** là một nền tảng được xây dựng trên “ V8 Javascript engine ” được viết bằng C++ và Javascript. Nền tảng này được phát triển bởi Ryan Lienhart Dahl vào năm 2009.

NodeJS ra đời khi các developer đời đầu của JavaScript mở rộng nó từ một thứ bạn chỉ chạy được trên trình duyệt thành một thứ bạn có thể chạy trên máy của mình dưới dạng ứng dụng độc lập.

Giờ đây bạn có thể làm được nhiều thứ với JavaScript hơn là chỉ tương tác với các Website.

Cả trình duyệt JavaScript và Node.js đều chạy trên JavaScript runtime V8 engine. Công cụ này lấy code JavaScript của bạn và convert nó sang mã máy (bytecode) cho việc thực thi nhanh hơn. Mã máy là loại code thấp cấp hơn để máy tính có thể chạy mà không cần biên dịch nó.

Chính vì thế đây là môi trường hoàn hảo để chạy các ứng dụng, web ngoài trình duyệt mà người dùng đang sử dụng. Đây cũng được coi là giải pháp thiết yếu giúp ứng dụng sử dụng dữ liệu hiệu quả nhờ mô hình event driven (Mô hình hướng sự kiện) một cách không đồng bộ.

NodeJS hoạt động như thế nào?

Bước 1: Khi sử dụng Node JS các câu lệnh sẽ được đưa vào queue, hàng đợi. Đồng thời chạy từ trên xuống dưới.

Bước 2: Lúc này Event loop lấy 1 tác vụ ở queue cho vào stack, và trước đó stack hoàn toàn trống. Điều này có thể hiểu rằng câu lệnh đầu tiên – console.log sẽ được stack xử lý và đưa ra “caulenh1”

Bước 3: Sau khi stack xử lý xong, tác vụ đầu sẽ được lấy ra và tiếp tục đưa tác vụ tiếp theo vào để xử lý tiếp. Cụ thể có thể là setTimeout(function(){ console.log(“cau lenh 2”); },0);. Lúc này có thể thấy SetTimeout trở thành 1 hàm tính toán thời gian nằm trong web AIPS và đưa vào web AIPs để đợi tác vụ tiếp theo. Nó sẽ được đưa trở lại hàng đợi khi queue đã trống.

Bước 4: Tương tự khi thực hiện tác vụ 1, tác vụ 3 sẽ in câu lệnh và hàng đợi ở trạng thái rỗng. Khi đó tác vụ 2 sẽ được đưa trở lại queue, tiếp đến sang stack để đi qua xử lý.

Những ứng dụng nào nên viết bằng NodeJs:

* Fast File Upload: Đây là những tool được viết để hỗ trợ tải lên file tốc độ cao
* Websocket server: Là các dạng máy chủ dành cho có lượng truy cập lớn và tương tác khủng như hệ thống chat online, Game online, Game server….
* Restful API: Là ứng dụng được dùng để hỗ trợ các phần mềm, ứng dụng chính qua API
* Any Real-time Data Application: Đây là ứng dụng yêu cầu tốc độ tải cao hoặc chia nhỏ các big app thành các site app nhỏ hơn.
* Ad server: Là những loại máy chủ quảng cáo phổ biến hiện nay.

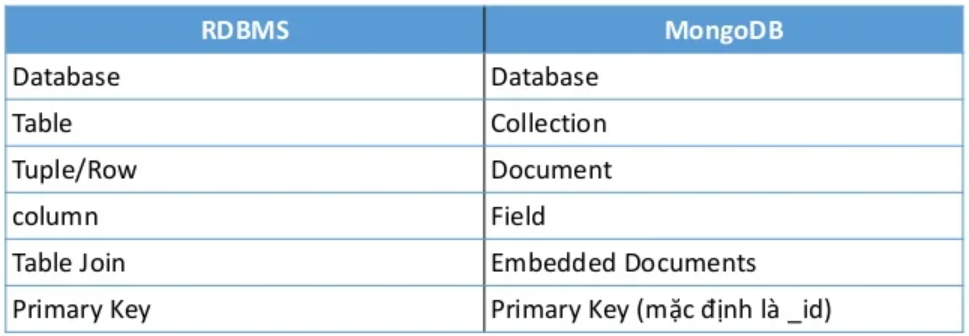
## 2.3 MONGODB



**MongoDB** lần đầu ra đời bởi MongoDB Inc., tại thời điểm đó là thế hệ 10, vào tháng Mười năm 2007, nó là một phần của sản phẩm PaaS (Platform as a Service) tương tự như Windows Azure và Google App Engine. Sau đó nó đã được chuyển thành nguồn mở từ năm 2009.

**MongoDB** là một database hướng tài liệu (document), một dạng NoSQL database. Vì thế, MongoDB sẽ tránh cấu trúc table-based của relational database để thích ứng với các tài liệu như JSON có một schema rất linh hoạt gọi là BSON. [**MongoDB**](https://topdev.vn/viec-lam-it/mongodb-kt90) sử dụng lưu trữ dữ liệu dưới dạng Document JSON nên mỗi một collection sẽ các các kích cỡ và các document khác nhau. Các dữ liệu được lưu trữ trong document kiểu JSON nên truy vấn sẽ rất nhanh.

So sánh một số khái niệm trong MongoDB và các cơ sở dữ liệu quan hệ (RDBMS) khác:



=> Trong đó ta thấy key **\_id** là do hệ thống tự tạo ra một field như vậy làm khóa chính.

Khi nào nên sử dụng MongoDB:

MongoDB thật sự rất hot nhưng không phải lúc nào ta sử dụng nó cũng tốt, có những trường hợp không nên sử dụng và nên sử dụng, chi tiết thế nào thì chúng ta cùng thảo luận.

- Nếu website của bạn có tính chất INSERT cao, bởi vì mặc định MongoDB có sẵn cơ chế ghi với tốc độ cao và an toàn.

- Website của bạn ở dạng thời gian thực nhiều, nghĩa là nhiều người thao tác với ứng dung. Nếu trong quá trình load bị lỗi tại một điểm nào đó thì nó sẽ bỏ qua phần đó nên sẽ an toàn.

- Website bạn có nhiều dữ liệu quá, giả sử web bạn có đến 10 triệu records thì đó là cơn ác mộng với MYSQL. Bởi vì MongoDB có khả năng tìm kiến thông tin liên quan cũng khá nhanh nên trường hợp này nên dùng nó.

- Máy chủ không có hệ quản trị CSDL, trường hợp này thường bạn sẽ sử dụng SQLITE hoặc là MongoDB.

## 2.4 EXPRESSJS



Expressjs hay còn được viết là Express js, Express.js. Đây là một framework mã nguồn mở miễn phí cho Node.js. Express.js được sử dụng trong thiết kế và xây dựng các ứng dụng web một cách đơn giản và nhanh chóng.

Vì Express js chỉ yêu cầu ngôn ngữ lập trình Javascript nên việc xây dựng các ứng dụng web và API trở nên đơn giản hơn với các lập trình viên và nhà phát triển.Expressjs cũng là một khuôn khổ của Node.js do đó hầu hết các mã code đã được viết sẵn cho các lập trình viên có thể làm việc.

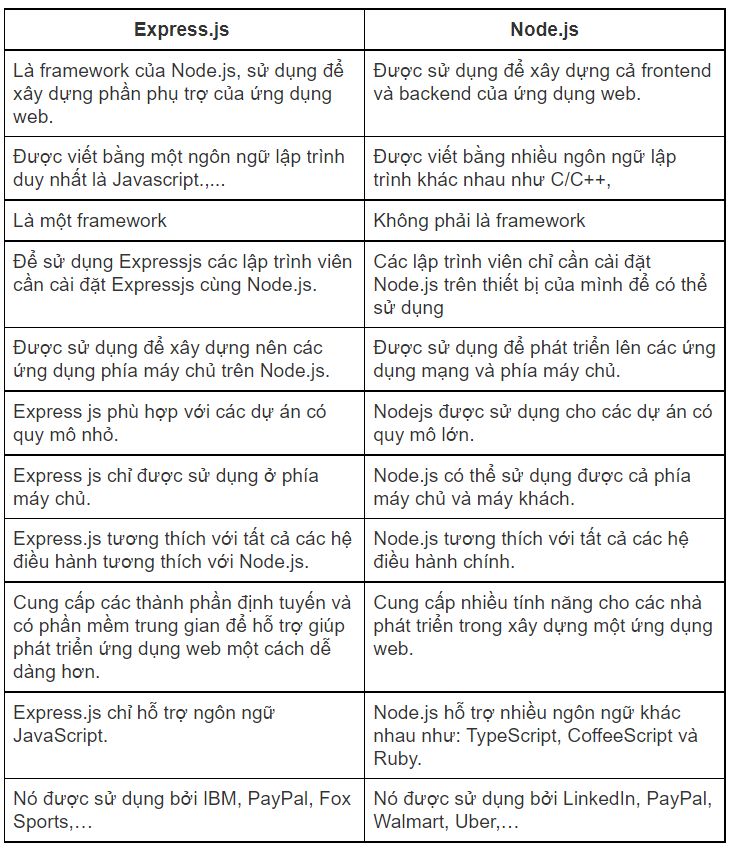
Có Expressjs mà các nhà lập trình có thể dễ dàng tạo các ứng dụng 1 web, nhiều web hoặc kết hợp. Do có dung lượng khá nhẹ, Expressjs giúp cho việc tổ chức các ứng dụng web thành một kiến trúc MVC có tổ chức hơn.Để có thể sử dụng được mã nguồn này, chúng ta cần phải biết về Javascript và HTML.

Expressjs cũng là một phần của công nghệ giúp quản lý các ứng dụng web một cách dễ dàng hơn hay còn được gọi là ngăn xếp phần mềm MEAN.Nhờ có thư viện Javascript của Express js đã giúp cho các nhà lập trình xây dựng nên các ứng dụng web hiệu quả và nhanh chóng hơn. Expressjs cũng được sử dụng để nâng cao các chức năng của Node.js.

Trên thực tế, nếu không sử dụng Express.js, bạn sẽ phải thực hiện rất nhiều bước lập trình phức tạp để xây dựng nên một [API](https://itnavi.com.vn/blog/api-la-gi/) hiệu quả. Express js đã giúp cho việc lập trình trong Node.js trở nên dễ dàng hơn và có nhiều tính năng mới bổ sung.

Phân biệt giữa ExpressJS và NodeJS

Sở hữu các tính năng gần như tương tự nhau do vậy ExpressJS và NodeJS thường được đưa lên bàn cân.



## 2.5 VueJS

**VueJS là gì?**

VueJS là một framework mã nguồn mở của JavaScript được sử dụng để phát triển các giao diện web tương tác. Nó là một trong những framework nổi tiếng được sử dụng để đơn giản hóa việc phát triển web. VueJS tập trung vào view layer. Nó có thể dễ dàng tích hợp vào các dự án lớn để phát triển front-end mà không gặp bất kỳ sự cố nào.

**Ưu điểm**

**Kích thước nhỏ:** Tệp zip được tải xuống của framework này chỉ nặng 18 KB. Điều này khiến nó không chỉ cài đặt nhanh mà còn tác động tích cực đến SEO và UX của bạn.

**Kết xuất và hiệu suất DOM ảo:** Mô hình đối tượng tài liệu (DOM) là thứ bạn có thể gặp phải khi kết xuất các trang web. DOM đại diện cho một trang HTML với các kiểu, thành phần và nội dung dưới dạng cấu trúc cây của các đối tượng (nút). Các đối tượng cây DOM lưu trữ dưới dạng cây và được tạo bởi trình duyệt khi tải trang.

vuejs là gì


Khi người dùng tương tác với trang, các đối tượng sẽ thay đổi trạng thái của chúng, do đó trình duyệt sẽ phải cập nhật thông tin và hiển thị trên màn hình. Tuy nhiên, việc cập nhật toàn bộ DOM rất phức tạp. Ưu tiên tốc độ load, VueJS sử dụng DOM ảo. Hãy coi đây là một bản sao của DOM gốc giúp tìm ra những phần tử cần cập nhật mà không cần kết xuất lại toàn bộ cây nút. Cách tiếp cận này giúp hiển thị trang khá nhanh và cải thiện hiệu suất ứng dụng.

Hệ thống phản ứng và các tùy chọn ràng buộc dữ liệu: Liên kết dữ liệu là kết nối giữa mô hình dữ liệu (nguồn dữ liệu) và mẫu DOM hoặc HTML của chế độ xem. Liên kết dữ liệu một chiều cho phép thông tin truyền theo một hướng, từ mô hình sang chế độ xem hoặc ngược lại. Trong trường hợp đầu tiên, các thay đổi đối với nguồn sẽ tự động cập nhật DOM, nhưng nó không hoạt động ngược lại vì DOM có quyền truy cập chỉ đọc vào mô hình.

Liên kết dữ liệu hai chiều cho phép trao đổi dữ liệu giữa mô hình và chế độ xem theo cả hai hướng. Nói cách khác, mô hình cũng lắng nghe các sự kiện trên DOM và bất kỳ cập nhật nào ở một bên sẽ phản ánh ngay lập tức ở bên kia. Cách tiếp cận này loại bỏ mã soạn sẵn và đơn giản hóa việc phát triển ứng dụng. Tuy nhiên, việc khó Debug và dễ xảy ra lỗi khiến luồng hai chiều không phù hợp cho các dự án lớn.

**Nhược điểm**

**Thiếu hỗ trợ cho các dự án quy mô lớn:** Quy mô nhóm phát triển và cộng đồng của VueJS vẫn không thể so sánh với Angular hay React. Framework này cũng không được hỗ trợ tài chính từ các doanh nghiệp lớn. Để được áp dụng trong các dự án quy mô lớn, công nghệ phải ổn định và được hỗ trợ mạnh mẽ để các vấn đề có thể được giải quyết nhanh chóng. Mặc dù VueJS không gặp nhiều vấn đề và thậm chí còn có nhu cầu đến từ các doanh nghiệp như IBM và Adobe, nhưng nó chủ yếu được sử dụng trong các dự án tương đối nhỏ.

**Nguy cơ đến từ việc quá linh hoạt:** Tính linh hoạt là một đặc tính gây tranh cãi của một dự án lớn. Cung cấp cho nhóm phát triển của bạn quá nhiều tùy chọn có thể dẫn đến các cách tiếp cận lập trình khác nhau trong một nhóm. Và kết quả là, nó trở thành một công cụ vô hiệu hóa cuối cùng thay vì một phần mềm hoạt động.

**Nguồn tài nguyên giới hạn:** Mặc dù hệ sinh thái khá rộng và có tất cả các công cụ cần thiết để bắt đầu phát triển với VueJS, nhưng framework này không lớn bằng React hay Angular. Nói chính xác hơn, chỉ cần so sánh số lượng plugin có sẵn cho React và Vue.js, sự khác biệt là ở hàng trăm đơn vị. Các plugin hiện có có thể được sử dụng với các framework khác cũng thường không được hỗ trợ.

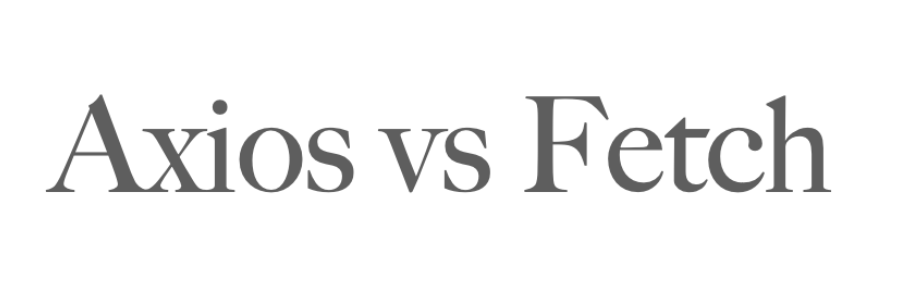
## 2.6 AXIOS



Axios là một HTTP client được viết dựa trên Promises được dùng để hỗ trợ cho việc xây dựng các ứng dụng API từ đơn giản đến phức tạp và có thể được sử dụng cả ở trình duyệt.

Việc tạo ra một HTTP request dùng để fetch hay lưu dữ liệu là một trong những nhiệm vụ thường thấy mà một ứng dụng Javascript phía client cần phải làm khi muốn giao tiếp với phía server.

Tại sao nên sử dụng Axios thay vì Fetch ?



Các browser hiện đại bây giờ thường đi kèm và xây dựng sẵn các tính năng Fetch API mới hơn, vậy tạo sao ta lại không dùng nó luôn cho xong? Rất nhiều những sự khác biệt giữa **Axios** và **Fetch** khiến rất nhiều người vẫn lựa chọn **Axios** thay vì **Fetch**.

Một trong những đặc điểm lớn ta có thể thấy đó là cách mà 2 thư viện xử lý với các HTTP error code. Khi sử dụng Fetch, nếu khi server trả về các mã lỗi 4xx hay 5xx, thì hàm catch() của bạn sẽ không được gọi đến và người lập trình viên sẽ có nhiệm vụ phải tự kiểm tra trạng thái của mã trả về để xác định xem liệu request đó có thành công hay không. Trong khi đó, **Axios** sẽ reject tất cả các promise của request nếu một trong các mã lỗi trên được trả về.

Một điểm khác biệt nho nhỏ khác, điều mà không ít các lập trình viên mới trong việc xây dựng API gặp phải đó là việc **Fetch** không tự động gủi trả cookies về cho server khi tạo một request. Ta sẽ cần phải truyền một cách trực tiếp các option để cho cookies có thể được include. Còn với **Axios** thì không phải lo về vấn đề này.

Sự khác biệt có thể được coi là vô cùng đáng chú ý đối với rất nhiều lập trình viên đó là khả năng cập nhật tiến độ của những lần uploads/downloads. Vì **Axios** được xây dựng dựa trên các XHR API cũ hơn, nên có thể khai báo các hàm callback cho onUploadProgress và onDownloadProgress để hiện thị phần trăm thành công tại giao diện cho app. Và cho đến hiện tại thì Fetch vẫn chưa hỗ trợ gì cho việc này.

Cuối cùng, **Axios** có thể được xử dụng ở cả browser và chính điều này đã tạo cơ hội cho việc chia sẻ code Javascript giữa các trình duyệt và phần back-end hay việc thực hiện render cho ứng dụng front-end của bạn ở phía server.

Một trong các cách thường thấy để cài đặt **Axios** đó là thông qua trình quản lý các npm package.

# CHƯƠNG III

# NỘI DUNG

## 3.1 YÊU CẦU TỐI THIỂU CỦA ĐỀ TÀI

* Tìm hiểu về cách thức hoạt động kinh doanh thương mại.
* Tìm hiểu về các giải pháp để xây dựng một website thương mại điện tử như thế nào.
* Xây dựng một website mang tính thương mại hóa.
* Xây dựng một trang web với đầy đủ các chức năng để phục vụ cho Quản Trị viên nói riêng và Khách Hàng nói chung.
* Đòi hỏi một môi trường thân thiện với người tiêu dùng.

## 3.2 XÂY DỰNG CHỨC NĂNG WEBSITE THƯƠNG MẠI ĐIỆN TỬ

Website thương mại điện tử có một số chức năng như sau tương ứng với từng đối tượng tương ứng:

+ Đối với khách hàng viếng thăm:

* Xem danh sách sản phẩm
* Xem thông tin chi tiết từng sản phẩm
* Tìm kiếm sản phẩm
* Lọc sản phẩm theo danh mục
* Đăng ký thành viên

+ Đối với khách hàng thành viên:

* Xem danh sách sản phẩm
* Xem thông tin chi tiết từng sản phẩm
* Tìm kiếm sản phẩm
* Lọc sản phẩm theo danh mục
* Đăng nhập
* Đăng xuất
* Đăng ký thành viên
* Đưa sản phẩm vào giỏ hàng
* Cập nhật giỏ hàng
* Đặt mua
* Thanh toán
* Xem lịch sử mua hàng

+ Đối với quản trị:

* Quản trị sản phẩm
* Quản trị danh mục
* Quản trị đơn hàng

## 3.3 MÔ TẢ NGHIỆP VỤ

### 3.3.1 Dành cho khách hàng nói chung

Website được xây dựng với mục tiêu là nơi trưng bài, giới thiệu các Sản Phẩm và các dịch vụ hỗ trợ Khách Hàng như: Tìm Kiếm, Lọc Sản Phẩm…

Khách Hàng bắt đầu tìm kiếm những Sản Phẩm thông qua *title* mình cần trên website, các Sản Phẩm được bố trí, bày bán theo từng *categories,* từng *price* hoặc đa dạng như Sản Phẩm mới, Sản Phẩm bán chạy… Khi Khách Hàng đã chọn được một Sản Phẩm ưng ý, Khách Hàng có thể nhấp chọn vào nút “Xem” để xem chi tiết thông tin Sản Phẩm như: *title, product\_id, price*, *categories*, *description, content, sold*…Sau đó Khách Hàng có thể chọn nút “Thêm vào vỏ hàng” nếu quá trình mua sắm vẫn chưa kết thúc. Nếu muốn Thanh Toán chọn nút có biểu tượng “Giỏ Hàng” để tiến hành thanh toán. Giỏ hàng đơn thuần là danh sách các Sản Phẩm mà Khách Hàng đã chọn bao gồm những thông tin như: *title*, *price*, Số lượng, Thành tiền.

* Giỏ hàng cung cấp chức năng xóa các Sản Phẩm trong gió hàng, cập nhật số lượng Sản Phẩm muốn mua.
* Ngay sau khi Khách Hàng đã chọn được các Sản Phẩm cần mua, có thể chọn nút “Đặt Mua”. Lúc này website sẽ lấy thông tin từ tài khoản Paypal thông tin về địa chỉ giao hàng (nếu có), sau cùng là thông tin về giỏ hàng (*product\_id*, *title*, số lượng, thành tiền…). Nếu Khách Hàng chưa đăng nhập trước đó thì cần phải đăng nhập để có thể đặt hàng.
* Sau khi Khách Hàng đã đăng nhập, chọn nút “Đặt Hàng” để hoàn tất quá trình mua hàng.

### 3.3.2 Dành cho quản trị viên

* Website cung cấp cho các Quản Trị Viên (admin) toàn quyền quản lý website như: CRUD Sản Phẩm, Quản lý các đơn hàng, Quản lý danh mục…

## 3.4 YÊU CẦU HỆ THỐNG

### 3.4.1 Yêu cầu chức năng

Mô tả chi tiết các chức năng dành cho từng đối tượng chính:

* *Khách hàng vãng lai:*
* Xem danh sách các sản phẩm: Khi người dùng truy cập vào website hệ thống sẽ hiển thị tự động các danh sách Sản Phẩm. Bên cạnh đó người dùng có thể chọn các danh mục Sản Phẩm để có thể lựa chọn các loại Sản Phẩm phù hợp.
* Xem thông tin chi tiết từng sản phẩm: Sau khi tìm thấy Sản Phẩm phù hợp Khách Hàng có thể nhấn chọn vào Sản Phẩm đó để xem chi tiết của Sản Phẩm.
* Tìm kiếm sản phẩm: Nếu Khách Hàng muốn tìm kiếm bất kì một Sản Phẩm nào đó thì có thể nhập thông tin Sản Phẩm muốn tìm lên thanh công cụ tìm kiếm để thực hiện chức năng tìm kiếm. Sản Phẩm sẽ được hiển thị lên giao diện nếu có. Ngược lại sẽ hiển thị “Không tìm thấy kết quả”.
* Đăng ký thành viên: Khi khách hàng viếng thăm muốn mua hàng thì có thể thực hiện chức năng đăng ký để trở thành thành viên của website và thực hiện các chức năng mua hàng.
* *Khách hàng thành viên:* Ngoài có những các chức năng của khách hàng vãng lai, khách hàng thành viên còn có những chức năng sau:
* Đưa các sản phẩm vào giỏ hàng: Sau khi khách hàng tìm thấy sản phẩm đáp ứng nhu cầu của mình thì có thể chọn đưa sản phẩm vào giỏ hàng.
* Cập nhật giỏ hàng: Khi người dùng đã lựa chọn một sản phẩm nhưng tìm được một sản phẩm hợp lý hơn thì có thể xóa sản phẩm đã có trong giỏ hàng vào chọn lại sản phẩm mới. Ngoài ra, khi cần mua sản phẩm đó với một số lượng khác thì người dùng cũng có thể cập nhật trong giỏ hàng.
* Đặt mua: Sau khi xem thông tin sản phẩm đúng với nhu cầu thì người dùng có thể chọn đặt mua để đưa sản phẩm vào giỏ hàng và có thể thanh toán.
* Thanh toán sản phẩm: Khi đã chọn Sản Phẩm và quyết định chắc chắn mua chúng, Khách Hàng có thể chọn chức năng thanh toán để thực hiện xác nhận đơn hàng cho mình. Muốn thực hiện này thì yêu cầu người dùng phải đăng nhập vào hệ thống. Trong quá trình thực hiện chức năng thanh toán thì người dùng có thể thấy được danh sách sản phẩm mà mình đặt mua trong giỏ hàng.
* Xem lịch sử mua hàng: Khách hàng có thể xem lại các Sản Phẩm mà đã từng mua trên website.
* Bình luận sản phẩm: Chức năng này cho phép Khách Hàng đánh giá các Sản Phẩm mình đã từng mua trên website.
* *Quản trị viên website:* Có tất cả các chức năng của Khách Hàng Thành Viên và có thêm chức năng quản lý khác như:
* Quản lý Sản Phẩm, Đơn Hàng, Danh Mục và một số thông tin khác. Để thực hiện chức năng quản lý thì yêu cầu các Quản Trị Viên phải đăng nhập hệ thống. Quản Trị Viên chọn mục mình cần thực hiện quản lý như Sản Phẩm. Trong quản lý Sản Phẩm có xem, thêm, sửa, xóa Sản Phẩm. Chọn thao tác cần thực hiện, điền đầy đủ thông tin trong quá trình thực hiện rồi xác nhận thao tác.

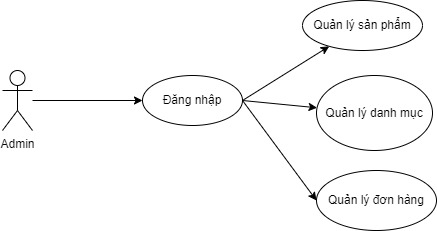
### 3.4.2 Yêu cầu phi chức năng

* Dung lượng website phải vừa phải, tốc độ truy xuất cao, hạn chế downtime không cần thiết. Ngoài ra, cần đảm bảo chế độ bảo mật cho trang web nói riêng và thông tin Khách Hàng nói chung, không chấp nhận sai sót.
* Cơ sở dữ liệu phải hoạt động tốt khi hệ thống đang làm việc.
* Yêu cầu các bước chuyển tiếp phải rõ ràng, rành mạch, không rắc rối.
* Môi trường, hình ảnh, màu sắc ánh sáng thân thiện, dễ chịu đối với Khách Hàng và cả Quản Trị Viên.

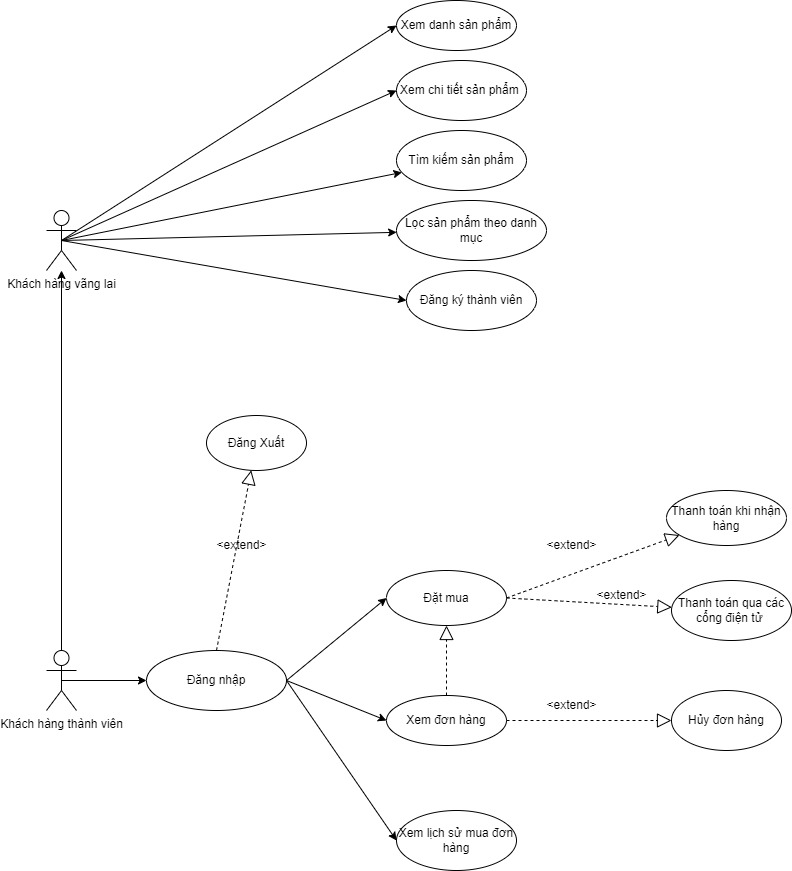
## 3.5. CÁC SƠ ĐỒ

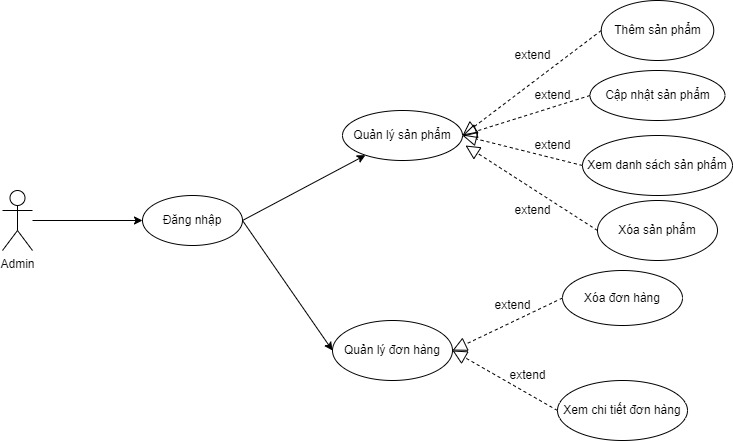
### 3.5.1 Use case diagram





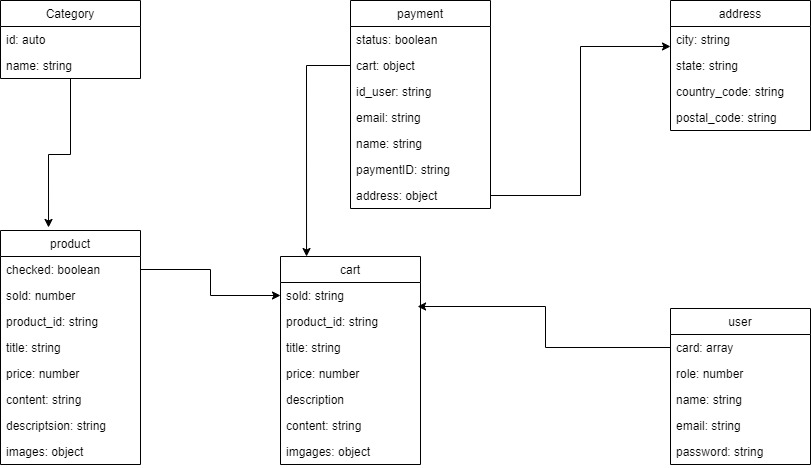
* Người dùng khách (khách hàng không có tài khoản) có thể thực hiện các chức năng sau: xem danh sách các sản phẩm, xem chỉ tiết các sản phẩm, tìm kiếm sản phẩmvà đăng ký thành viên.
* Người dùng thành viên (khách hàng thành viên) có thể thực hiện các chức năng như sau: đăng nhập, đăng ký, đăng xuất, mua hàng, xem đơn hàng, xem danh sách các sản phẩm, xem chi tiết các sản phẩm, tìm kiếm sản phẩm, đưa sản phẩm vào giỏ hàng, cập nhật giỏ hàng.





* Quản trị viên có thể thực hiện các chức năng sau: đăng nhập, đăng xuất, quản lý đơn hàng (xóa đơn hàng, xem chi tiết đơn hàng, cập nhật đơn hàng), quản lý danh mục (xem danh sách các danh mục, thêm danh mục, cập nhập danh mục, xóa danh mục), quản lý sản phẩm (xem danh sách các sản phẩm, thêm sản phẩm, cập nhập sản phẩm, xóa sản phẩm).

### 3.5.2 Entity Diagram (sơ đồ thực thể)



Danh sách các bảng thực thể:

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên bảng** | **Diễn giải** |
| 1 | User | Thành Viên |
| 2 | Categories | Danh Mục |
| 3 | Products | Sản Phẩm |
| 4 | Payment | Đơn Hàng |

Bảng User: dùng để lưu thông tin và quản lý thông tin của khách hàng và quản trị viên.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User** | | | | |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| 1 | name | string | Not Null | Tên khách hàng |
| 2 | email | string | Not Null | Email khách hàng |
| 3 | password | string | Not Null | Mật khẩu |
| 4 | cart | array |  | Giỏ hàng |
| 5 | role | number | Not Null | Phân quyền |

Bảng Categories: dùng để lưu thông tin và quản lý thông tin danh mục sản phẩm.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Categories** | | | | |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| 1 | name | string | Not Null | Tên danh mục |

Bảng Products: dùng để lưu thông tin và quản lý thông tin sản phẩm.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Products | | | | |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | checked | boolean | Not Null | Trạng thái |
| 2 | sold | number | Not Null | Số lượng đã bán |
| 3 | product\_id | string | Not Null | Mã sản phẩm |
| 4 | title | string | Not Null | Tên sản phẩm |
| 5 | price | number | Not Null | Giá sản phẩm |
| 6 | description | string | Not Null | Chi tiết sản phẩm |
| 7 | content | string | Not Null | Mô tả sản phẩm |
| 8 | images | object | Not Null | Hình ảnh |

Bảng Payment: dùng để lưu thông tin và quản lý thông tin đơn hàng.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Payment** | | | | |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| 1 | status | boolean | Not Null | Tình trạng |
| 2 | cart | object | Not Null | Giỏ hàng |
| 3 | id\_user | string | Not Null | Mã khách hàng |
| 4 | email | string | Not Null | Email khách hàng |
| 5 | name | string | Not Null | Tên khách hàng |
| 6 | paymentID | String | Not Null | Địa chỉ ví Paypal |
| 7 | address | Object | Not Null | Địa chỉ giao hàng |

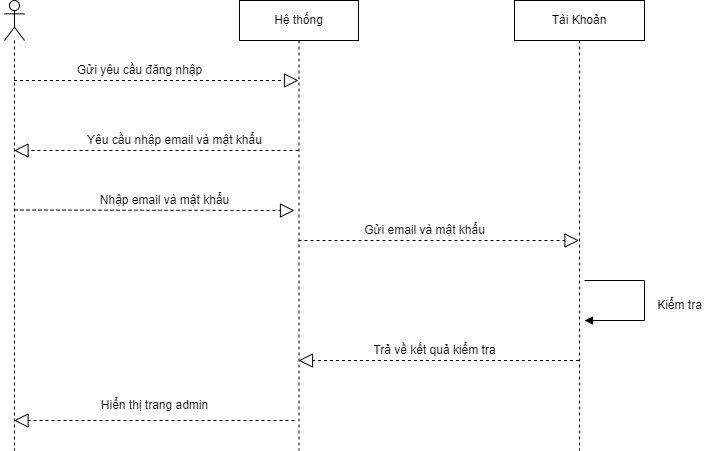
Bảng Cart: dùng để lưu thông tin và quản lý thông tin giỏ hàng của khách hàng và đơn hàng.

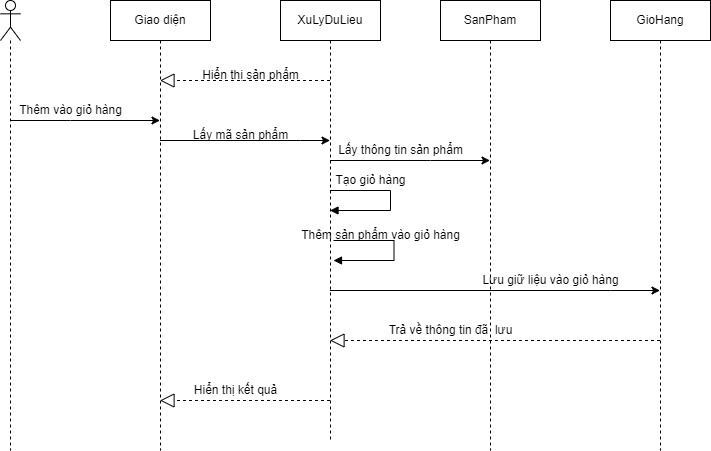
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Cart** | | | | |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| 1 | sold | number | Not Null | Số lượng đã bán |
| 2 | product\_id | string | Not Null | Mã sản phẩm |
| 3 | title | string | Not Null | Tên sản phẩm |
| 4 | price | number | Not Null | Giá sản phẩm |
| 5 | description | string | Not Null | Chi tiết sản phẩm |
| 6 | content | string | Not Null | Mô tả sản phẩm |
| 7 | images | object | Not Null | Hình ảnh |

Bảng Address: dùng để lưu thông tin và quản lý địa chỉ khách hàng tron đơn hàng.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Address** | | | | |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| 1 | city | string | Not Null | Thành phố |
| 2 | state | string | Not Null | Quận/huyện |
| 3 | country\_code | string | Not Null | Mã quốc gia |
| 4 | postal\_code | string | Not Null | Mã bưu chính |

### 3.5.3 Sequence diagram (Sơ đồ tuần tự)

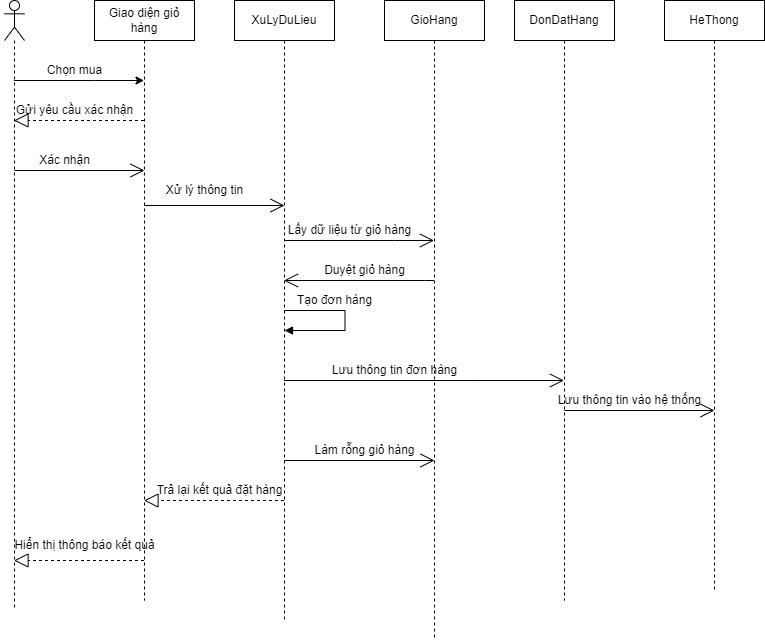




Miêu tả chi tiết các trình tự thêm sản phẩm vào giỏ hàng:

* Sau khi trang web sử lý dữ liệu sẽ hiện ra giao diện sản phẩm.
* Khách hàng có thể lựa chọn tùy ý các sản phẩm yêu thích và thêm vào giỏ hàng.
* Web sẽ lấy id của sản phẩm và tìm sản phẩm đó theo id và lưu nó vào LocalStorage tạm thời.
* Tạo giỏ hàng mới và lấy dữ liệu đã lưu trong LocalStorage ra giỏ hàng.

Sau khi khách hàng hoàn tất quá trình mua hàng, clip vào giỏ hàng hàng sẽ nhận được các sản phẩm đã chọn từ trước.



Miêu tả chi tiết trình tự mua hàng của khách hàng

* Sau quá trình chọn lựa sản phẩm khách hàng chọn Đặt Mua
* Trang web sẽ gửi một yêu cầu xác nhận có đồng ý mua hàng hay không, chọn đồng ý xác nhận
* Lúc này trang web sẽ xử lý thông tin lấy dữ liệu các sản phẩm và số lượng
* Tiếp theo sẽ tạo đơn hàng và lưu vào hệ thống với thông tin gồm: tất cả sản phẩm trong giỏ hàng và số lượng cho các sản phẩm đó, thông tin khách hàng mua, tổng tiền của háo đơn và thời gian đặt
* Hệ thống sẽ làm rỗng lại giỏ hàng (tức làm rỗng lại Local Storage) sau khi lưu thông tin đơn hàng thành công
* Trả lại kết quả đơn hàng khách hàng đã đặt mua gồm các thông tin như: các sản phẩm đã đặt, số lượng, tổng tiền của đơn hàng, ngày đặt và trạng thái của đơn hàng.

### 3.5.4 Functional diagram (Sơ đồ chức năng)

Sơ đồ chức năng của khách hàng vãng lai:

Sơ đồ chức năng của khách hàng thành viên:

Sơ đồ chức năng của quản trị viên:

## 3.6 KIỂM THỬ

### 3.6.1 Giao diện của khách hàng vãng lai và khách hàng thành viên.

Graphical user interface, website

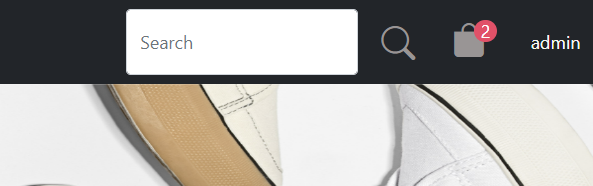
Description automatically generated

Mô tả: Giao diện trang chủ của website.

Graphical user interface, website

Description automatically generated

Mô tả: Giao diện lọc sản phẩm theo danh mục.



Mô tả: Giao diện tìm kiếm sản phẩm.

Graphical user interface

Description automatically generated

Mô tả: Nếu nhấn vào “Mua” nhưng chưa đăng nhập sẽ xuất hiện hộp thoại thông báo.

A picture containing graphical user interface

Description automatically generated

Mô tả: Giao diện khi nhấn vào nút “Xem” để xem chi tiết sản phẩm.

Graphical user interface

Description automatically generated

Mô tả: Giao diện đăng ký.

* + Khi điền thiếu thông tin hoặc không đúng chuẩn Khách Hàng sẽ nhận được một thông báo.

Graphical user interface, application, Teams

Description automatically generated

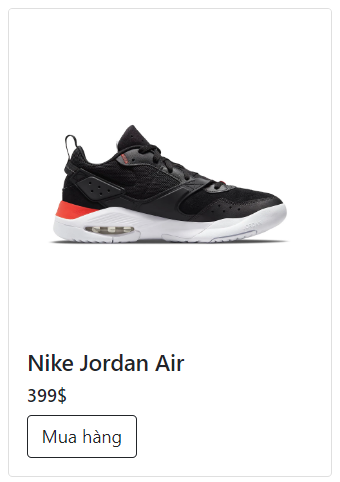
Mô tả: Giao diện đăng nhập.

* + Khi điền thông tin không đúng web sẽ hiển thị thông báo “Mật khẩu không đúng” hoặc “Người dùng không tồn tại”
  + Ngược lại, khi đúng thông tin Khách Hàng sẽ được chuyển đến trang như phía dưới đây.

Graphical user interface, application

Description automatically generated

Mô tả: Giao diện sau khi đăng nhập thành công.



Giả sử, nhấp chọn “Mua hàng” như bình bên cho hai Sản Phẩm đầu tiên bạn có thể thấy giỏ hàng hiển thị số lượng sản phẩm thêm  vào giỏ hàng.

Và khi nhấp chọn sẽ được chuyển đến trang giỏ hàng như hình bên dưới và như vậy, hai sản phẩm đầu tiên đã được thêm vào giỏ hàng

Graphical user interface, application

Description automatically generated

Tại đây khách hàng có thể chỉnh sửa giỏ hàng, nếu hoàn tất nhấp “Đặt hàng” để tiến hành thanh toán.

### 3.6.2 Đối với giao diện của Quản Trị Viên.

Tại trang đăng nhập sao khi Quản Trị Viên đăng nhập đúng tài khoản và mật khẩu của Quản Trị Viên sẽ được chuyển đến trang quản lý như phía dưới:

Graphical user interface, application

Description automatically generated

Chọn “Thêm Sản Phẩm” trên menu để thêm sản phẩm mới hoặc chọn vào nút “Hiệu chỉnh” trên sản phẩm tương ứng để sửa và chọn “Xóa” để xóa sản phẩm.

Application

Description automatically generated with medium confidence

Nhập vào đầy đủ thông tin và nhấn “Lưu” để thêm sản phẩm.

Graphical user interface, text

Description automatically generated with medium confidence

Giao diện sửa sản phẩm khi nhấn vào nút “Hiệu chỉnh”.

# CHƯƠNG IV

# KẾT LUẬN

## 4.1 KẾT QUẢ ĐẠT ĐƯỢC

* Tìm hiểu về cách thức hoạt động kinh doanh thương mại.
* Tìm hiểu về các giải pháp để xây dựng một website thương mại như thế nào.
* Xây dựng một website mang tính thương mại hóa.
* Xây dựng một trang web với đầy đủ các chức năng để phục vụ cho Quản Trị viên nói riêng và Khách Hàng nói chung.
* Đòi hỏi một môi trường thân thiện với người tiêu dùng.

### 4.1.1 Ưu điểm

* Giới thiệu sản phẩm đến khách hàng. Đối với khách hàng: cho phép thực hiện các chức năng xem hàng hóa, góp ý.
* Sau khi mua sản phẩm, đặt mua sản phẩm và thanh toán bằng nhiều cách khác nhau.
* Khách hàng có thể thanh toán trực tuyến xem thông tin đơn hàng của mình.
* Hoàn thành hầu hết các chức năng cơ bản của một trang web bán hàng.
* Xây dựng trên mô hình MVC dễ dàng quản lý việc thay đổi cấu trúc sau này.
* Website áp dụng giao thức bảo mật SSL nên thông tin đã bảo mật hơn.

### 4.1.2 Nhược điểm

Về dữ liệu

**+** Dữ liệu chưa được đầy đủ.

**+** Dữ liệu trên trang web hiện tại không đúng với thực tế, sử dụng nguồn dữ liệu ảo.

Về giao diện

+ Giao diện chưa sinh động

+ Giao diện cứng, không thể thay đổi template cho giao diện.

## 4.2. HƯỚNG PHÁT TRIỂN

* Với những thiếu sót như phân tích trên, để phần mềm ngày càng hoàn thiện, phục vụ tốt cho nhu cầu của người dùng thì nên có những cập nhật sau đây:
* Thiết kế trang web đẹp hơn, tiện dụng hơn, có nhiều chức năng và sinh động hơn. Xây dựng thêm một diễn đàn trên trang web để khách hàng có thể trao đổi bình luận về sản phẩm. Thiết kế thêm vị trí để cập nhật giá cả thị trường. Nâng cao khả năng bảo mật cho chương trình. Ngăn ngừa tối đa các trường hợp đột nhập dữ liệu hệ thống. Chỉnh sữa dữ liệu cho chính xác với thực tế và sẽ nâng cấp dần website để đáp ứng được những nhu cầu của khách hàng trong thời đại mới.

**-------HẾT-------**