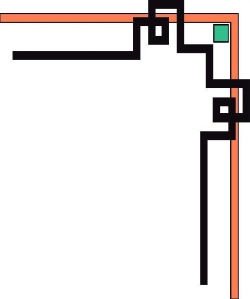
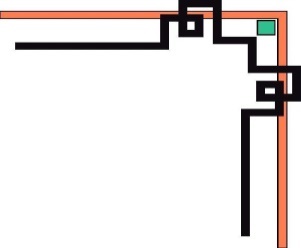
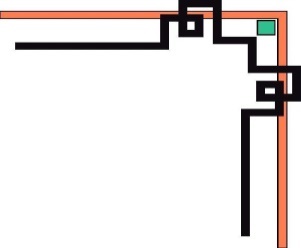
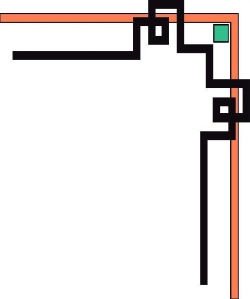
**ĐẠI HỌC ĐÀ NẴNG**



**TRƯỜNG ĐẠI HỌC SƯ PHẠM**

**KHOA TIN HỌC**A picture containing text, alcohol

Description automatically generated

**BÁO CÁO CUỐI KỲ**

**ĐỀ TÀI:**

**XÂY DỰNG WEBSITE SHOP BÁN GIÀY**

**GV hướng dẫn: VÕ CÔNG ĐÌNH**

**Sinh viên thực hiện: Phan Hữu Lân**

**Trần Thị Bảo Ngân**

**Lê Nguyễn Minh Trọng**

**Võ Đại Húa**

**Vannida Nanthavongsa**

**Nhóm: 10**

*Đà Nẵng, Tháng 5/202**2*

# LỜI CẢM ƠN

Để có thể hoàn thành đề tài đồ án này và có được kiến thức như ngày hôm nay, đầu tiên nhóm em xin gửi lời cảm ơn đến Ban Giám Hiệu nhà trường. Cùng toàn thể Thầy Cô Khoa Tin Học – Trường Đại Học Sư Phạm Đà Nẵng đã tận tình giảng dạy, truyền đạt kiến thức cũng như những kinh nghiệm quý báu cho em trong suốt quá trình học tập tại trường.

Đặc biệt nhóm em xin chân thành cảm ơn thầy VÕ CÔNG ĐÌNH đã tận tình hướng dẫn và quan tâm, động viên nhóm em trong quá trình thực hiện đề tài. Nhờ có thầy mà nhóm em có sự rèn luyện bản thân, thêm kiến thức cho bản thân để có được cách làm việc hiệu quả, đồng thời áp dụng vào các môn học, công việc khác trong cuộc sống.

Nhóm em cũng bày tỏ lòng biết ơn đến những bạn bè đã động viên và tạo mọi điều kiện giúp đỡ em trong quá trình học tập.

Trong quá trình nỗ lực để hoàn thành báo cáo cuối kỳ, cũng đã tìm hiểu nhiều tài liệu và nhờ những anh chị khóa trước và thầy cô của khoa Tin Học để hoàn thành tốt nhiệm vụ của mình. Nhóm em xin chân thành cảm ơn những sự giúp đỡ tận tình từ tất cả mọi người. Mặc dù nhóm em đã cố gắng hoàn thành tốt đề tài nhưng cũng không thể tránh khỏi những sai sót nhất định, rất mong được sự thông cảm và chia sẻ cùng quý Thầy Cô.

Nhóm em xin gửi lời chúc sức khỏe và thành đạt tới tất cả quý thầy cô cùng các bạn!

# MỤC LỤC

[LỜI CẢM ƠN 4](#_Toc104578053)

[MỤC LỤC 5](#_Toc104578054)

[BẢNG PHÂN CÔNG 7](#_Toc104578055)

[BẢNG TIẾN ĐỘ 8](#_Toc104578056)

[MỞ ĐẦU 10](#_Toc104578057)

[1. LÝ DO CHỌN ĐỀ TÀI: 10](#_Toc104578058)

[2. MỤC ĐÍCH NGHIÊN CỨU: 11](#_Toc104578059)

[3. NỘI DUNG NGHIÊN CỨU: 11](#_Toc104578060)

[4. PHƯƠNG PHÁP NGHIÊN CỨU: 11](#_Toc104578061)

[CHƯƠNG 1: GIỚI THIỆU CHUNG VỀ API, PHÂN TÍCH THIẾT KẾ CƠ SỞ DỮ LIỆU 12](#_Toc104578062)

[1.1. API LÀ GÌ? 12](#_Toc104578063)

[1.1.1. Khái niệm API: 12](#_Toc104578064)

[1.1.2. Những điểm nổi bật của API: 13](#_Toc104578065)

[1.1.3. Các loại API: 13](#_Toc104578066)

[1.1.4. REST và RESTful API là gì? 14](#_Toc104578067)

[1.1.5. Các Nguyên tắc của REST: 15](#_Toc104578068)

[1.2. WEB API LÀ GÌ? 16](#_Toc104578069)

[1.2.1. Một số ưu điểm vượt trội của Web API như: 17](#_Toc104578070)

[1.2.2. Web API trong lập trình, thiết kế website: 18](#_Toc104578071)

[1.3. CÁCH CÀI ĐẶT API: 18](#_Toc104578072)

[1.4. CƠ SỞ DỮ LIỆU: 19](#_Toc104578073)

[1.3.1. Lược đồ quan hệ: 19](#_Toc104578074)

[1.3.2. Chi tiết các bảng: 19](#_Toc104578075)

[CHƯƠNG 2: GIỚI THIỆU NGÔN NGỮ, TOOL, PHẦM MỀM ĐƯỢC SỬ DỤNG 24](#_Toc104578076)

[2.1. JAVASCRIPT: 24](#_Toc104578077)

[2.2. HTML: 25](#_Toc104578078)

[2.3. CSS: 26](#_Toc104578079)

[2.4. NOPEJS: 27](#_Toc104578080)

[2.5. MONGODB: 27](#_Toc104578081)

[2.6. REACTJS: 28](#_Toc104578082)

[CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ 30](#_Toc104578083)

[3.1. GIỚI THIỆU CHUNG: 30](#_Toc104578084)

[3.1.1. Mô tả: 30](#_Toc104578085)

[3.1.2. Đối tượng sử dụng: 30](#_Toc104578086)

[3.1.3. Yêu cầu hệ thống: 31](#_Toc104578087)

[3.1.3.1. Yêu cầu phi chức năng: 31](#_Toc104578088)

[3.1.3.2. Yêu cầu chức năng: 31](#_Toc104578089)

[3.2. PHÂN TÍCH THIẾT KẾ: 33](#_Toc104578090)

[3.2.1. Nhận diện các tác nhân: 33](#_Toc104578091)

[3.2.2. Lược đồ quan hệ: 34](#_Toc104578092)

[3.2.3. User case: 35](#_Toc104578093)

[3.3. KẾT QUẢ THỰC HIỆN: 36](#_Toc104578094)

[3.3.1. Giao diện người dùng: 36](#_Toc104578095)

[3.3.2. Giao diện admin: 40](#_Toc104578096)

[HƯỚNG PHÁT TRIỂN 44](#_Toc104578097)

[BÀI HỌC KINH NGHIỆM 45](#_Toc104578098)

[TÀI LIỆU THAM KHẢO 47](#_Toc104578099)

# MỤC LỤC HÌNH ẢNH

[Hình 1: Lược đồ quan hệ 35](#_Toc104844799)

[Hình 2: Sơ đồ user case 36](#_Toc104844800)

[Hình 3: Trang chủ web 37](#_Toc104844801)

[Hình 4: Chi tiết sản phẩm 38](#_Toc104844802)

[Hình 5: Trang giỏ hàng 39](#_Toc104844803)

[Hình 6: Trang thanh toán 39](#_Toc104844804)

[Hình 7: Hồ sơ đăng ký 40](#_Toc104844805)

[Hình 8: Trang đăng ký 40](#_Toc104844806)

[Hình 9: Trang đăng nhập 41](#_Toc104844807)

[Hình 10: Bảng điều khiển (Admin) 42](#_Toc104844808)

[Hình 11: Danh sách thành viên 42](#_Toc104844809)

[Hình 12: Trang thêm sản phẩm 43](#_Toc104844810)

[Hình 13: Trang xóa sửa sản phẩm 44](#_Toc104844811)

[Hình 14: Trang quản lý đơn hàng 44](#_Toc104844812)

# BẢNG PHÂN CÔNG

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Phan Hữu Lân** | **Trần Thị Bảo Ngân** | **Vannida Nanthavongsa** | **Võ Đại Húa** | **Lê Nguyễn Minh Trọng** |
| -Viết API User  -Connect Database  -Update User  -Đăng nhập  -Tìm kiếm  -Chi tiết sản phẩm  -Đánh giá  -Bảng điều khiển(Admin) | -Quản lí đơn hàng  -Viết API Product  -Làm Slide,  Viết Báo cáo | -Giỏ hàng  -Sửa, xóa sản phẩm  Viết Báo cáo | -Thanh toán  -Thêm sản phẩm | -Đăng kí |

# BẢNG TIẾN ĐỘ

|  |  |  |  |
| --- | --- | --- | --- |
| **Bước** | **Thời gian thực hiện** | **Nội dung thực hiện** |  |
| 1 | 14/3/2022 đến 21/3/2022 | -Tìm hiểu đề tài,nghiên cứu đề tài. |  |
| 2 | 22/3/2022 đến 28/3/2022 | -Thiết kế giao diện cho web.  -Tổng hợp giao diện |  |
| 3 | 29/3/2022 đến 4/4/2022 | - Nghiên cứu,xây dựng thuật toán,các chức năng của web |  |
| 4 | 5/4/2022 đến 11/4/2022 | - Nghiên cứu, tìm dữ liệu, hình ảnh, xây dựng API, cơ sở dữ liệu cho website |  |
| 5 | 12/4/2022 đến 25/4/2022 | Viết, xây dựng API cho website bằng ngôn ngữ nopeJs, mongoDb. |  |
| 6 | 26/4/2022 đến 9/5/2022 | Viết các chức năng, hiện thị dữ liệu bằng ngôn ngữ js, reactjs, kết hợp với các thuật toán để xử lý dữ liệu. |  |
| 7 | 9/5/2022 đến 20/5/2022 | -Hoàn thành chương trình và chạy thử, fix lỗi. |  |
| 8 | 20/5/2022 đến 27/5/2022 | -Viết báo cáo.  - Hoàn thành báo cáo |  |

# MỞ ĐẦU

## 1. LÝ DO CHỌN ĐỀ TÀI:

Chúng ta có thể nói rằng thế ký 21 đã và đang chứng kiến sự phát triển mạnh mẽ của ngành Công nghệ thông tin (CNTT). CNTT đã và đang thay đổi thế giới một cách nhanh chóng và tưng bước kéo nền tri thức của nhân loại xích lại gần nhau hơn. Đó là nhờ việc phát triển hệ thống website trên toàn thế giới.

Ngày nay, Website đã đóng một vai trò quan trọng đối với con người chúng ta từ giải trí cho đến quảng cáo, thương mại, quản lý…Website (thương mại điện tử) sẽ dần thay thế những phương thức kinh doanh cũ trong các doanh nghiệp bởi tính ưu việt mà Website mang lại như: nhanh hơn, rẻ hơn, tiện dụng hơn, hiệu quả hơn và không bị giới hạn không gian và thời gian.

API được ví như “cánh tay phải” đắc lực không hề thiếu so với lập trình website và phong cách thiết kế những ứng dụng ứng dụng di động. Nó có tính năng link những tính năng của web, app với những cơ sở tài liệu; tăng tương tác tối đa giữa người dùng và ứng dụng. Bên cạnh đó, công cụ này còn có rất nhiều công dụng, điều mê hoặc khác khiến chúng ta tò mò muốn tìm hiểu sâu về API hơn.

Từ những giao thức tách biệt, API như một cầu nối kết nối chúng, giúp người dùng hoàn toàn có thể tương tác giữ ứng dụng và website, cải tổ và lan rộng ra quản lý và vận hành trên mọi thiết bị và phần cứng dù trực tuyến hay ngoại tuyến. API giúp giảm tải hầu hết những việc làm, nâng cao sự đúng mực trong quy trình phong cách thiết kế website bán hàng trực tuyến và những loại web khác, tạo sự chuyên nghiệp, nhanh gọn và thuận tiện hơn cho những lập trình viên. Tất cả những ưu điểm đó giúp cho API và Web API trở thành những chiếc chìa khóa vạn năng, cải tổ đời sống thuận tiện, mưu trí và tân tiến hơn.

Để góp phần tìm hiểu sâu hơn và ứng dụng về data API, web API nhóm em đã thực hiện đề tài “ Xây dựng website shop bán giày”.

## 2. MỤC ĐÍCH NGHIÊN CỨU:

Nghiên cứu đề tài này chúng em nhằm mục đích học hỏi, tìm hiểu, rút ra kinh nghiệm và ứng dụng xây dựng dữ liệu API, web API vào công việc và cuộc sống sau này.

## 3. NỘI DUNG NGHIÊN CỨU:

* + Phân tích các yêu cầu cần thiết
  + Tìm hiểu cách xây dựng một website
  + Tìm hiểu và nghiên cứu ngôn ngữ lập trình: ReactJs,HTML,CSS….
  + Tìm hiểu và xây dựng cơ sở dữ liệu với NopeJs, MongoDB,…
  + Thiết kế giao diện cho website
  + Xây dựng website hoàn chỉnh

## 4. PHƯƠNG PHÁP NGHIÊN CỨU:

* Tài liệu về data API, web API.
* Xem, đọc các tài liệu về cách xây dựng website.
* Đọc sách tham khảo.
* Thảo luận ý kiến với bạn bè, thầy cô, trao đổi với các thành viên trong nhóm.

# CHƯƠNG 1: GIỚI THIỆU CHUNG VỀ API, PHÂN TÍCH THIẾT KẾ CƠ SỞ DỮ LIỆU

Đối với đồ án “Xây dựng website shop bán giày” này thì nhóm chúng em sử dụng ngôn ngữ NopeJs để viết api và dùng MogoBb để lưu trữ API của website.

Để biết cách cài đặt cơ sở dữ liệu, viết API như thế nào, trước tiên ta phải đi tìm hiểu khái niệm cơ bản về các thuật ngữ cơ bản trong lập trình website sau đây:

* API
* WEB API
* REST và RESTful API

## API LÀ GÌ?

### 1.1.1. Khái niệm API:

API là viết tắt của Application Programming Interface (giao diện lập trình ứng dụng) phương thức kết nối với các thư viện và ứng dụng khác. Windows có nhiều API, và Twitter cũng có web API, tuy nhiên chúng thực hiện các chức năng khác nhau, với mục tiêu khác nhau. Nó chính là một phần mềm giao tiếp được sử dụng bởi các ứng dụng khác nhau. Nó cũng giống như bàn phím là thiết bị dùng để giao tiếp giữa người sử dụng và máy tính, API là một phần mềm giao tiếp giữa chương trình và hệ điều hành.

Mỗi bộ API dành cho các hệ điều hành khác nhau là hoàn toàn khác nhau và không có sự tương thích với nhau. API dành cho các hệ điều hành Windows và Linux là hoàn toàn khác nhau. API cung cấp khả năng cung cấp khả năng truy xuất đến một tập các hàm hay dùng. Nói nôm na API là một loại công cụ để tạo ra phần mềm, cũng giống như các viên gạch xây nên một toà nhà.

Chúng ta chia việc xây nhà ra các phần khác nhau, API cũng như viên gạch là một bộ phận của ngôi nhà, nhưng việc tạo ra nó là một công đoạn riêng hoàn toàn, chúng ra có thể tự làm hoặc đơn giản là đi mua của các nhà cung cấp. Web API là một trong những công nghệ mới của Microsoft dùng để xây dựng dịch vụ thành phần phân tán. Web API là mô hình dùng để hỗ trợ MVC bao gồm: routing, controller, action result, filter, filter, loc container, model binder, unit test, injection. Bên cạnh đó nó còn hỗ trợ restful đầy đủ các phương thức: Get/Post/put/delete dữ liệu.

### 1.1.2. Những điểm nổi bật của API:

Đây là một trong những framework mới sẽ giúp ít cho bạn trong việc xây dựng các HTTP service một cách rất đơn giản và nhanh chóng. Mã nguồn mở nên bạn có thể được sử dụng bởi bất kì một client nào hỗ trợ XML, JSON. Nó cũng có khả năng hỗ trợ đầy đủ các thành phần HTTP: URI, request/response headers, caching, versioning, content forma. Bạn có thể sử dụng các host nằm trong phần ứng dụng hoặc trên IIS. Một kiểu kiến trúc vô cùng phù hợp dành cho các thiết bị trang bị băng thông giới hạn như smartphone, tablet. Thường nó có định dạng dữ liệu là JSON, XML hoặc một kiểu dữ liệu bất kỳ.

* **Tóm lại:** API giúp chúng tacấu hình đơn giản khi được so sánh với WCF Khả năng trình diễn cao Hỗ trợ chức năng RESTful một cách đầy đủ Hỗ trợ đầy đủ các thành phần MVC như: routing, controller, action result, filter, model binder, IoC container, dependency injection, unit test Mã nguồn mở.

### 1.1.3. Các loại API:

Có rất nhiều các loại API khác nhau tùy thuộc vào phân ngành trong CNTT:

* Hệ thống API trên nền tảng web còn gọi vắn tắt là web API: Hiện đang rất phổ biến, các website lớn đều cung cấp hệ thống API cho phép bạn kết nối, lấy dữ liệu hoặc cập nhật dữ liệu vào hệ thống. Ví dụ thực tế, nếu bạn sử dụng dịch vụ bán hàng trực tuyến của Lazada, bạn sẽ cần phải thực hiện một số các tác vụ như tạo sản phẩm mới, cập nhật sản phẩm mới. Giả sử bạn có 1000 mặt hàng (thực tế có nhiều đơn vị phân phối có số lượng sản phẩm rất lớn), việc cập nhật số lượng tồn kho bằng tay là cực kỳ vất vả và dễ dẫn đến nhầm lẫn. Lazada cung cấp hệ thống API và bạn có thể theo các đặc tả xây dựng một kết nối từ hệ thống của bạn sang Lazada và mọi thứ sẽ tự động hết.
* Hệ thống API trên Hệ điều hành: Khái niệm này có trước cả web API, khi Microsoft cung cấp các hệ điều hành Windows thì đều cung cấp thêm các tài liệu API là đặc tả các hàm, phương thức, lời gọi hàm cũng như các giao thức kết nối cho lập trình viên, giúp lập trình viên có thể tạo ra các phần mềm ứng dụng có thể tương tác trực tiếp với hệ điều hành.
* Các API của thư viện phần mềm hoặc framework: API mô tả và quy định các hành vi mong muốn mà các thư viện cung cấp, một API có thể có nhiều các triển khai khác nhau và nó cũng giúp cho một chương trình viết bằng ngôn ngữ này có thể sử dụng thư viện được viết bằng ngôn ngữ khác. API cũng có thể liên quan đến các framework khi framework được xây dựng trên nhiều các thư viện và thực thi nhiều các API khác nhau. Tuy nhiên việc sử dụng API trên framework không giống với thông thường, truy cập đến các API được xây dựng trong framework sẽ mở rộng nội dung của nó và các class mới được "cắm" vào (plug) khung tự nó. Kiểm soát tổng thể luồng ứng dụng có thể nằm ngoài tầm kiểm soát của thành phần gọi bằng cách đảo ngược kiểm soát - Inversion of Control

### 1.1.4. REST và RESTful API là gì?

Representational State Transfer, hoặc **REST**, mô tả một kiểu kiến ​​trúc cho các dịch vụ web. REST bao gồm một tập hợp các tiêu chuẩn hoặc các ràng buộc để chia sẻ dữ liệu giữa các hệ thống khác nhau và các hệ thống cài đặt REST được gọi là RESTful. REST là một khái niệm trừu tượng, không phải là ngôn ngữ, framework hoặc kiểu phần mềm nào đó.

REST giống như một bộ sưu tập vinyl so với sử dụng dịch vụ phát nhạc trực tuyến. Với bộ sưu tập vinyl vật lý, mỗi bản ghi phải được nhân lên toàn bộ để chia sẻ và phân phối các bản sao. Tuy nhiên, với một dịch vụ phát trực tuyến, cùng một bản nhạc có thể được chia sẻ vĩnh viễn thông qua một tham chiếu đến dữ liệu như tiêu đề bài hát. Trong trường hợp này, phát nhạc trực tuyến là một dịch vụ RESTful và bộ sưu tập vinyl là một dịch vụ không RESTful.

**API** là viết tắt của Application Programming Interface, là giao diện cho phép các chương trình phần mềm giao tiếp với nhau. Một **RESTful API** đơn giản là một API tuân thủ các nguyên tắc và ràng buộc của REST. Trong một Web API, máy chủ nhận **yêu cầu** thông qua một URL endpoint và gửi về **phản hồi**, thường là dữ liệu ở định dạng chẳng hạn như JSON.

### 1.1.5. Các Nguyên tắc của REST:

Sáu ràng buộc hướng dẫn xác định kiến ​​trúc REST, được nêu ở dưới đây.

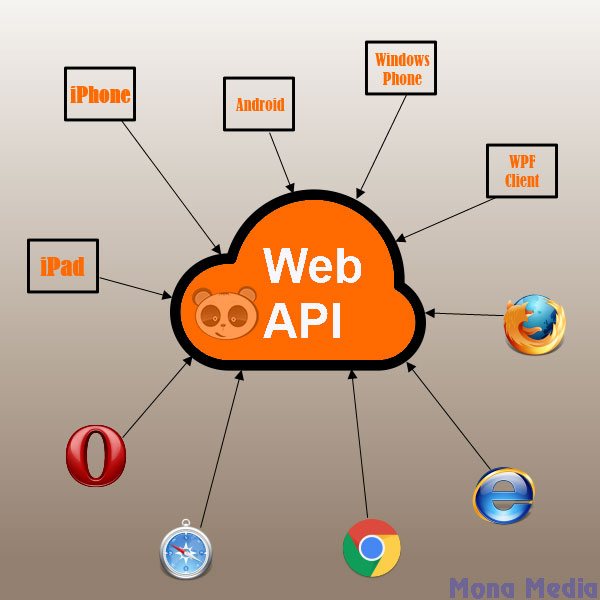
1. **Uniform Interface**: Giao diện của các thành phần phải giống nhau. Điều này có nghĩa là sử dụng URI tiêu chuẩn để xác định tài nguyên - nói cách khác, đường dẫn có thể được nhập vào thanh địa chỉ của trình duyệt.
2. **Client-Server**: Có sự phân tách các mối liên hệ giữa server, nơi lưu trữ và xử lý dữ liệu và client, gửi yêu cầu và hiển thị phản hồi.
3. **Stateless Interactions**: Tất cả các thông tin về mỗi yêu cầu được chứa trong mỗi yêu cầu riêng lẻ và không phụ thuộc vào trạng thái của session.
4. **Cacheable**: Client và server có thể lưu cache tài nguyên.
5. **Layered System**: Client có thể được kết nối với server cuối hoặc lớp trung gian chẳng hạn như một load-balancer.
6. **Code on Demand (Optional)**: Một client có thể tải về code, làm giảm khả năng hiển thị từ bên ngoài.

## WEB API LÀ GÌ?

Internet hiện nay đã quá phổ biến với rất nhiều các website, ứng dụng web... các ứng dụng trên web đã tiến dần đến phần mềm ứng dụng thông thường trong hệ điều hành, chính vì vậy khái niệm API giờ đây cũng gắn liền với các trang web. Khi bạn gặp thuật ngữ API ở đâu đó thì có tới 80% khả năng đó chính là web API. Tất cả những website lớn cung cấp nội dung, dịch vụ... đều đã xây dựng các hệ thống web API của riêng mình.

Vậy wed api là gì?

Web API hay ASP.NET Web API là một framework dùng để kiến thiết xây dựng và lập trình những dịch vụ web HTTP. Nó có dạng là một RESTful API văn minh, quy tụ đủ những điều kiện kèm theo của REST cũng như những tiêu chuẩn tựa như, được tối ưu cho những dịch vụ trực tuyến cũng như ứng dụng web lúc bấy giờ. Web API chiếm hữu những ưu điểm tiêu biểu vượt trội mà một API cũ có cùng nhiều ưu điểm mới tuyệt vời, giúp nó nhận được nhiều sự tin yêu từ những nhà tăng trưởng và lập trình website.



### 1.2.1. Một số ưu điểm vượt trội của Web API như:

* Dễ viết, tinh chỉnh tốt hơn so với các framework WCF, Web service trước đây
* Nâng cao trải nghiệm người dùng nhờ UX thân thiện
* Thỏa mãn những tiêu chuẩn phổ biến về REST và HTTP
* Hỗ trợ tốt các tính năng và thành phần của HTTP như: caching, versioning, HttpRequestMessage, HttpResponseMessage,…các MVC như: routing, container, controller,…
* Khả năng bảo mật cao, xác nhận 2 chiều khi có request
* Dữ liệu trả về ở nhiều định dạng phổ biến khác nhau từ XML cho tới JSON
* Host được nhiều loại client khác nhau từ ứng dụng nền web cho tới các phần mềm desktop,…

### 1.2.2. Web API trong lập trình, thiết kế website:

Từ những nghiên cứu và phân tích trên hoàn toàn có thể thấy, Web APi cung ứng những tính năng, cải tổ hiệu suất thao tác website cực tốt. Thay vì sử dụng những trang tĩnh với nội dung cứng ngắc, website văn minh ngày này hoàn toàn có thể ứng dụng Web API để cung ứng nội dung linh động với tính cá thể hóa cao hơn .Một số ứng dụng Web API trong nghành phong cách thiết kế và lập trình website điển hình nổi bật như :

* Chức năng tìm kiếm
* Khả năng tích hợp kênh mạng xã hội
* Xây dựng website bán hàng trực tuyến

## CÁCH CÀI ĐẶT API:

Yêu cầu đầu tiên của chúng ta là đảm bảo Node.js và npm được cài đặt toàn cục trên máy tính. Chúng ta có thể kiểm tra cả hai bằng cách sử dụng cờ -v, sẽ hiển thị phiên bản của chúng. Mở công cụ dòng lệnh của bạn và gõ như sau.

|  |  |
| --- | --- |
| 1 | node -v && npm -v |

v10.8.0

6.2.0

Phiên bản của bạn có thể hơi khác, nhưng miễn là có thì chúng ta có thể bắt đầu.

Sau đó chúng ta cách thiết lập một máy chủ HTTP tích hợp và máy chủ trong Node, yêu cầu thông qua bộ định tuyến và URL và sử dụng dữ liệu JSON với các yêu cầu GET.

Chúng ta sẽ kết nối server của chúng ta với MongoDB để tạo, xem, cập nhật và xóa người dùng trong cơ sở dữ liệu, hoàn thiện chức năng của API.

## CƠ SỞ DỮ LIỆU:

### 1.3.1. Lược đồ quan hệ:

### 1.3.2. Chi tiết các bảng:

**users:**  
user\_id  
name  
username  
date\_of\_birth  
gender  
phone  
address  
password

email

**orders:**  
order\_id  
Product\_id

user\_id

countInStock

orderItems

shippingAddress

paymentMethod

taxPrice

shippingPrice

totalPrice

**detail\_product:**  
detail\_id  
product\_id  
image

name

price

description

rating

numReviews  
countInStock

**product:**  
Product\_id  
catalog\_id  
name

image

rating

numReviews  
price

**catalogs:**  
Catalog\_id  
name

**Feedback**

id

chude

**detail\_report**

id

title

name

detail

email

phone

id\_chude

status

**Table users**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TT** | **Khóa** | **Tên trường** | **Kiểu dữ liệu** | **Null** | **Mô tả** |
| 1 | P | User\_id | Int(255) | Not null | Mã người dùng |
| 2 |  | Name | Varchar(100) | Not null | Họ tên người dùng |
| 3 | p | username | Varchar(100) | Not null | Tên tài khoản |
| 4 |  | Date\_of\_birth | Date | Null | Ngày sinh |
| 5 |  | Gender | Varchar(10) | Null | Giới tính |
| 6 |  | Phone | Varchar(15) | Null | Số điện thoại |
| 7 |  | address | Varchar(120) | Null | Địa chỉ |
| 8 |  | password | Varchar(40) | Not null | Mật khẩu |
| 9 |  | email | Varchar(120) | Not null | Địa chỉ email |

**Table product:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TT** | **Khóa** | **Tên trường** | **Kiểu dữ liệu** | **Null** | **Mô tả** |
| 1 | P | Product\_id | Int(255) | Not null | Mã sản phẩm |
| 2 | F | Catalog\_id | Int (255) | Not null | Mã hãng |
| 3 | P | name | Varchar(100) | Not null | Tên sản phẩm |
| 4 |  | price | Int (255) | Not Null | Giá |
| 5 |  | Image | Varchar(50) | Not Null | Hình ảnh |
| 6 |  | rating | Varchar(50) | Not Null | Xếp hạng |
| 7 |  | numReviews | Int(225) | Not Null | Số đánh giá |

**Table catalogs:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TT** | **Khóa** | **Tên trường** | **Kiểu dữ liệu** | **Null** | **Mô tả** |
| 1 | P | Catalog\_id | Int(255) | Not null | Mã hãng |
| 2 | p | name | Int (255) | Not null | Tên hãng |

**Table detail\_report:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TT** | **Khóa** | **Tên trường** | **Kiểu dữ liệu** | **Null** | **Mô tả** |
| 1 | P | Id | Int(255) | Not null | Mã phản hồi |
| 2 |  | title | Varchar (255) | Not null | Tiêu đề |
| 3 |  | name | Varchar (255) | Not null | Tên người gửi |
| 4 |  | Detail | text | Not Null | Nội dung |
| 5 |  | Email | Varchar (255) | Not Null | E-Mail người gửi |
| 6 |  | Phone | Varchar(20) | Not Null | Số điện thoại người gửi |
| 7 | F | Id\_chude | Int(225) | Not Null | Mã chủ đề |
| 8 |  | status | Int(2) | Not Null | Trạng thái |

**Table Orders**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TT** | **Khóa** | **Tên trường** | **Kiểu dữ liệu** | **Null** | **Mô tả** |
| 1 | P | Order\_id | Int(255) | Not null | Mã đơn hàng |
| 2 | F | User\_id | Int (255) | Not null | Mã người dùng |
| 3 | F | Product\_id | Int (255) | Not null | Mã sản phẩm |
| 4 |  | orderItems | Int(225) | Not null | Số lượng đặt |
| 5 |  | countInStock | Tinyint (2) | Not null | Tình trạng |
| 6 |  | shippingAddress | Varchar(2000) | Not null | Địa chỉ giao hàng |
| 7 |  | paymentMethod | Varchar(2000) | Not null | Phương thức trả tiền |
| 8 |  | taxPrice | Int(225) | Not null | Thuế |
| 9 |  | shippingPrice | Int(225) | Not null | Giá phí vận chuyển |
| 10 |  | totalPrice | Int(225) | Not null | Tổng giá đơn hàng |

**Table Feedback**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TT** | **Khóa** | **Tên trường** | **Kiểu dữ liệu** | **Null** | **Mô tả** |
| 1 | P | id | Int(255) | Not null | Mã chủ đề |
| 2 |  | chude | Varchar (255) | Null | Chủ đề |

**Table Detail Product**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TT** | **Khóa** | **Tên trường** | **Kiểu dữ liệu** | **Null** | **Mô tả** |
| 1 | P | detail\_id | Int(255) | Not null | Mã thông số |
| 2 | F | Product\_id | Int (255) | Not null | Mã sản phẩm |
| 3 |  | image | Varchar (50) | Null | Hình ảnh |
| 4 |  | name | Varchar (20) | Null | Tên sản phẩm |
| 5 |  | price | Varchar(100) | Null | Giá sản phẩm |
| 6 |  | Description | Varchar(2000) | Null | Miêu tả |
| 7 |  | rating | Varchar(20) | Null | Xếp hạng |
| 8 |  | numReviews | Varchar(100) | Null | Số đánh giá |
| 9 |  | countInStock | Varchar(2000) | Null | Số lượng có trong kho |

# CHƯƠNG 2: GIỚI THIỆU NGÔN NGỮ, TOOL, PHẦM MỀM ĐƯỢC SỬ DỤNG

## 2.1. JAVASCRIPT:

**JavaScript** là ngôn ngữ lập trình phổ biến nhất trên thế giới trong suốt 20 năm qua. Nó cũng là một trong số 3 ngôn ngữ chính của lập trình web:

[**HTML**](https://www.hostinger.vn/huong-dan/html-cheat-sheet-co-html5-tags-moi/): Giúp bạn thêm nội dung cho trang web.

[**CSS**](https://www.hostinger.vn/huong-dan/khac-biet-giua-inline-external-va-internal-style-css): Định dạng thiết kế, bố cục, phong cách, canh lề của trang web.

JavaScript: Cải thiện cách hoạt động của trang web.

JavaScript có thể [**học nhanh**](https://www.bitdegree.org/learn/javascript-basics/) và dễ dàng áp dụng cho nhiều mục đích khác nhau, từ việc cải thiện tính năng của website đến việc chạy game và tạo phần mềm nền web. Hơn nữa, có hàng ngàn mẫu template JavaScript và ứng dụng ngoài kia, nhờ vào sự cống hiến của cộng đồng, đặc biệt là Github.

**JavaScript**, theo phiên bản hiện hành, là một ngôn ngữ lập trình kịch bản dựa trên đối tượng được phát triển từ các ý niệm nguyên mẫu. Ngôn ngữ này được dùng rộng rãi cho các trang web, nhưng cũng được dùng để tạo khả năng viết script sử dụng các đối tượng nằm sẵn trong các ứng dụng. Nó vốn được phát triển bởi Brendan Eich tại hãng truyền thông Netscape với cái tên đầu tiên Mocha, rồi sau đó đổi tên thành LiveScript, và cuối cùng thành JavaScript. Giống Java, JavaScript có cú pháp tương tự C, nhưng nó gần với Self hơn Java. .js là phần mở rộng thường được dùng cho tập tin mã nguồn JavaScript

**Graphical user interface

Description automatically generated**

## 2.2. HTML:

Logo, icon

Description automatically generated

HTML (tiếng Anh, viết tắt cho HyperText Markup Language, hay là "Ngôn ngữ Đánh dấu Siêu văn bản") là một ngôn ngữ đánh dấu được thiết kế ra để tạo nên các trang web với các mẩu thông tin được trình bày trên World Wide Web. HTML được định nghĩa như là một ứng dụng đơn giản và được sử dụng trong các tổ chức cần đến các yêu cầu xuất bản phức tạp. HTML đã trở thành một chuẩn Internet do tổ chức World Wide Web Consortium (W3C) duy trì. Phiên bản chính thức mới nhất của HTML là HTML 4.01 (1999). Sau đó, các nhà phát triển đã thay thế nó bằng XHTML. Hiện nay, HTML đang được phát triển tiếp với phiên bản HTML5 hứa hẹn mang lại diện mạo mới cho Web.

HTML không phải là ngôn ngữ lập trình, nó là ngôn ngữ trình bày.

HTML là một **đánh dấu** cho ngôn ngữ **mô tả** các tài liệu web (trang web).

## ****2.3. CSS:****

Icon

Description automatically generated

**CSS** là chữ viết tắt của Cascading Style Sheets, nó là một ngôn ngữ được sử dụng để **tìm và định dạng** lại các phần tử được tạo ra bởi các ngôn ngữ đánh dấu ([HTML](https://topdev.vn/blog/html-la-gi/)). Nói ngắn gọn hơn là ngôn ngữ tạo phong cách cho trang web. Bạn có thể hiểu đơn giản rằng, nếu HTML đóng vai trò định dạng các phần tử trên website như việc tạo ra các đoạn văn bản, các tiêu đề, bảng,…thì CSS sẽ giúp chúng ta có thể thêm style vào các phần tử HTML đó như đổi bố cục, màu sắc trang, đổi màu chữ, font chữ, thay đổi cấu trúc…

CSS được phát triển bởi **W3C** ([World Wide Web Consortium](https://www.w3.org/)) vào năm 1996, vì HTML không được thiết kế để gắn tag để giúp định dạng trang web.Phương thức hoạt động của CSS là nó sẽ tìm dựa vào các vùng chọn, vùng chọn có thể là tên một thẻ HTML, tên một ID, class hay nhiều kiểu khác. Sau đó là nó sẽ áp dụng các thuộc tính cần thay đổi lên vùng chọn đó.

Mối tương quan giữa HTML và CSS rất mật thiết. HTML là ngôn ngữ markup (nền tảng của site) và CSS định hình phong cách (tất cả những gì tạo nên giao diện website), chúng là không thể tách rời.

## 2.4. NOPEJS:

**A picture containing text, clipart, vector graphics

Description automatically generated**

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.

Node.js 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.

## ****2.5. MONGODB:****

Logo, company name

Description automatically generated

**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.

* **Quản lý và truyền tải content** – Quản lý đa dạng nhiều product của content chỉ trong một kho lưu trữ data cho phép thay đổi và phản hồi nhanh chóng mà không chịu thêm phức tạp thêm từ hệ thống content.
* **Cấu trúc Mobile và Social** – MongoDB cung cấp một platform có sẵn, phản xạ nhanh, và dễ mở rộng cho phép rất nhiều khả năng đột phá, phân tích real-time, và hỗ trợ toàn cầu.
* **Quản lý data khách hàng** – Tận dụng khả năng query nhanh chóng cho phân tích real-time trên cơ sở dữ liệu người dùng cực lớn vớ các mô hình data phức tạp bằng các schema linh hoạt và tự động sharding cho mở rộng chiều ngang.

## 2.6. REACTJS:

**Graphical user interface

Description automatically generated with medium confidence**

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

**Ưu điểm của ReactJS:**

Ngoài việc hỗ trợ xây dựng giao diện nhanh, hạn chế lỗi trong quá trình code, cải thiện performance website thì những tính năng đặc biệt dưới đây có thể là lý do khiến bạn “chốt sale” với ReactJS và bắt đầu tìm hiểu nó từ bây giờ:

* **Phù hợp với đa dạng thể loại website**: ReactJS khiến cho việc khởi tạo website dễ dàng hơn bởi vì bạn không cần phải code nhiều như khi tạo trang web thuần chỉ dùng JavaScript, HTML và nó đã cung cấp cho bạn đủ loại “đồ chơi” để bạn có thể dùng cho nhiều trường hợp.
* **Tái sử dụng các Component**: Nếu bạn xây dựng các Component đủ tốt, đủ flexible để có thể thoả các “yêu cầu” của nhiều dự án khác nhau, bạn chỉ tốn thời gian xây dựng ban đầu và sử dụng lại hầu như toàn bộ ở các dự án sau. Không chỉ riêng mỗi ReactJS mà các framework hiện nay cũng đều cho phép chúng ta thực hiện điều đó, ví dụ Flutter chẳng hạn.
* **Có thể sử dụng cho cả Mobile application**: Hầu hết chúng ta đều biết rằng ReactJS được sử dụng cho việc lập trình website, nhưng thực chất nó được sinh ra không chỉ làm mỗi đều đó. Nếu bạn cần phát triển thêm ứng dụng Mobile, thì hãy sử dụng thêm React Native – một framework khác được phát triển cũng chính Facebook, bạn có thể dễ dàng “chia sẻ” các Component hoặc sử dung lại các Business Logic trong ứng dụng.
* **Thân thiện với SEO**: SEO là một phần không thể thiếu để đưa thông tin website của bạn lên top đầu tìm kiếm của Google. Bản chất ReactJS là một thư viện JavaScript, Google Search Engine hiện nay đã crawl và index được code JavaScript, tuy nhiên bạn cũng cần thêm một vài thư viện khác để hỗ trợ điều này nhé!
* **Debug dễ dàng**: Facebook đã phát hành 1 Chrome extension dùng trong việc debug trong quá trình phát triển ứng dụng. Điều đó giúp tăng tốc quá trình release sản phẩm cung như quá trình coding của bạn.
* **Công cụ phát triển web hot nhất hiện nay**: Nếu bạn nhìn vào số liệu thống kê từ [Google Trend](https://trends.google.com/trends/explore?geo=VN&q=reactjs,%2Fm%2F0j45p7w,VueJS) ở Việt Nam ở hình bên dưới, dạo lướt qua các trang tuyển dụng hàng đầu ở Việt Nam như Topdev, Itviec, v.v bạn sẽ thấy số lượng tuyển dụng cho vị trí React Developer là cực kỳ lớn cùng với mức lương vô cùng hấp dẫn và độ phổ biến hiện tại của ReactJS trên thị trường Việt Nam là như thế nào.

# CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ

## 3.1. GIỚI THIỆU CHUNG:

### 3.1.1. Mô tả:

Website bán giày dành cho tất cả mọi người ở mọi lứa tuổi, là nơi tìm mua một đôi ưng ý để phục vụ việc đi lại, học tập, làm việc và giải trí.

### 3.1.2. Đối tượng sử dụng:

**Ứng dụng được xây dựng nhằm phục vụ nhóm người dùng sau:**

* + **Người dùng ứng dụng( Người dùng):** Đây chính là đối tượng sử dụng chiếm đa số và quan trọng nhất của hệ thống.
  + **Lập trình viên:** Truy xuất dự liệu từ kho dữ liệu thông qua server được cung cấp bởi nhà quản lý dư liệu.

### 3.1.3. Yêu cầu hệ thống:

#### 3.1.3.1. Yêu cầu phi chức năng:

* + Hệ thống phải hỗ trợ người dùng trong việc tìm kiếm laptop và thông tin của laptop một cách nhanh chóng, hiệu quả và chính xác.
  + Hạn chế đến mức thấp nhất các sai sót có thể xảy ra trong quá trình sử dụng.
  + Giao diện thân thiện, đảm bảo được tính đơn giản để những người không có kinh nghiệm dùng máy tính có thể dùng.
  + Đảm bảo tính linh động nếu như sau này có sự thay đổi về số lượng danh mục
  + Xử lý lượng lớn thông tin một cách nhanh chóng.

#### 3.1.3.2. Yêu cầu chức năng:

* + **Các chức năng dành cho người dùng (user):**
* **Đăng ký**: Người dùng có thể đăng ký các thông tin cơ bản của mình như họ tên, số điện thoại… để đăng nhập vào hệ thống nếu họ chưa có tài khoản thành viên.
* **Đăng nhập**: Sau khi hoàn tất quá trình đăng ký thành viên ở trên hoặc người dùng đã có tài khoản từ trước, người dùng có thể lấy thông tin để đăng nhập hệ thống (thông tin để đăng nhập thường là username và password).
* **Xem thông tin sản phẩm:** Đây là chức năng cho phép người dùng xem trước một số thông tin cơ bản về sản phẩm.
* **Thêm sản phẩm vào giỏ hàng:** Sau khi lựa chọn được sản phẩm muốn mua người dùng có thể thêm sản phẩm vào giỏ hàng để thực hiện giao dịch.
* **Chỉnh sửa thông tin người dùng:** Cho phép người dùng chỉnh sửa thông tin cá nhân của chính tài khoản thành viên của mình.
* **Tìm kiếm:** giúp người dùng dễ dàng tìm sản phẩm thông qua tên của sản phẩm.
* **Liên hệ:** Giúp người dùng có thể phản hồi lại cho quản trị viên để khắc phục những vấn đề phát sinh khi sử dụng.
* **Đăng xuất:** Sau khi hoàn tất các thao tác trên hệ thống người dùng có thể đăng xuất để đảm bảo an toàn cho thông tin cá nhân của chính mình.
  + **Các chức năng dành cho quản trị hệ thống (admin)**
* **Đăng nhập**: Quản trị viên sẽ đăng nhập hệ thống bằng tài khoản cấp sẵn.
* **Quản lý thành viên:** Xem số lượng, chỉnh sửa thông tin hoặc xóa tài khoản của người dùng hiện có.
* **Quản lý sản phẩm:** Thêm, xóa, chỉnh sửa thông tin sản phẩm.
* **Quản lý đơn hàng:** Quản lý, tiếp nhận các giao dịch xảy ra.
* **Quản lý phản hồi:**  Quản lý, tiếp nhận các phản hồi của người dùng.

**3.3. Yêu cầu hệ thống**

* **Hệ quản trị cơ sở dữ liệu:** MongoDB.
* **Ngôn ngữ lập trình:** JavaScript, NopeJs, ReactJs.
* **Trình duyệt:** chạy tốt trên Google Chrome.

## 3.2. PHÂN TÍCH THIẾT KẾ:

### 3.2.1. Nhận diện các tác nhân:

Một tác nhân xác định một tập các vai trò khi người sử dụng tương tác với hệ thống. Người sử dụng có thể là một tác nhân hay một hệ thống khác.

Ở đây ta có 3 tác nhân chính là khách viếng thăm, người dùng (User) và người quản lý (Admin):

* **Khách viếng thăm**: Khách viếng thăm có quyền được xem sản phẩm, xem thông tin và tìm kiếm sản phẩm.
* **User**: Đối tượng trực tiếp sử dụng website. Thông qua giao diện website người dùng tương tác với kho dữ liệu như xem sản phẩm, tìm kiếm thông tin.
* **Admin**: Thực hiện quản lý sản phẩm, quản lý người dùng… thông qua giao diện của admin.

### 3.2.2. Lược đồ quan hệ:

Hình : Lược đồ quan hệ

**users:**  
user\_id  
name  
username  
date\_of\_birth  
gender  
phone  
address  
password

email

**orders:**  
order\_id  
Product\_id

user\_id

countInStock

orderItems

shippingAddress

paymentMethod

taxPrice

shippingPrice

totalPrice

**detail\_product:**  
detail\_id  
product\_id  
image

name

price

description

rating

numReviews  
countInStock

**product:**  
Product\_id  
catalog\_id  
name

image

rating

numReviews  
price

**catalogs:**  
Catalog\_id  
name

**Feedback**

id

chude

**detail\_report**

id

title

name

detail

email

phone

id\_chude

status

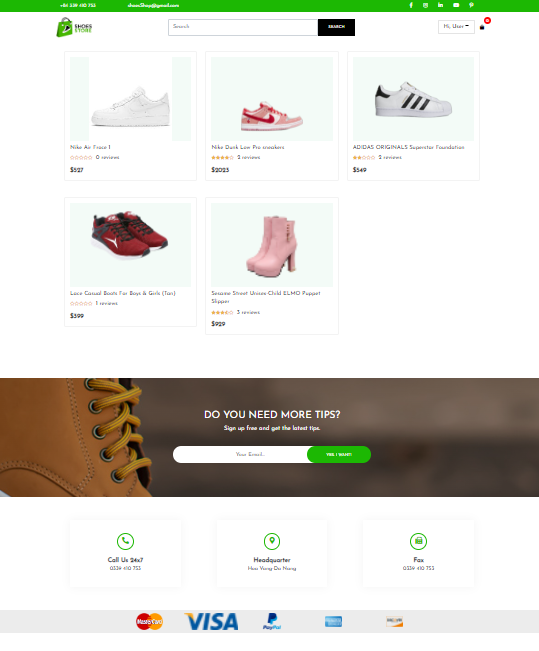
### 3.2.3. User case:

Hình : Sơ đồ user case

## 3.3. KẾT QUẢ THỰC HIỆN:

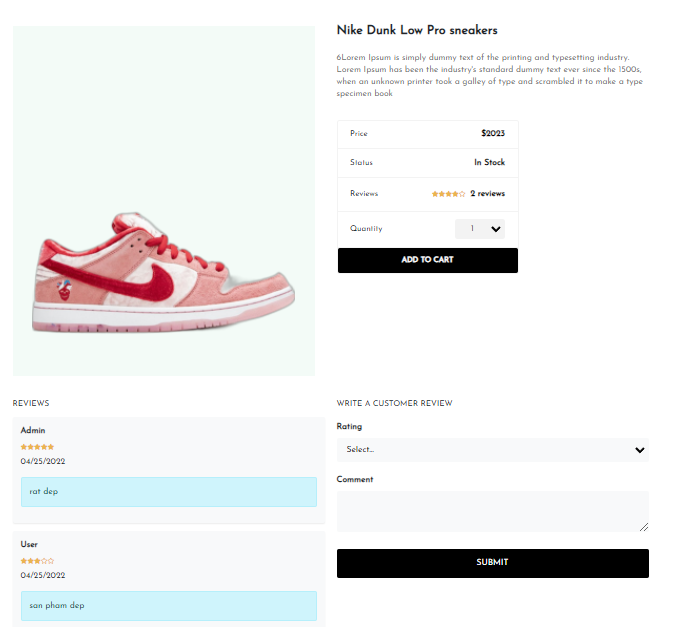
### 3.3.1. Giao diện người dùng:

**Trang chủ:**

****

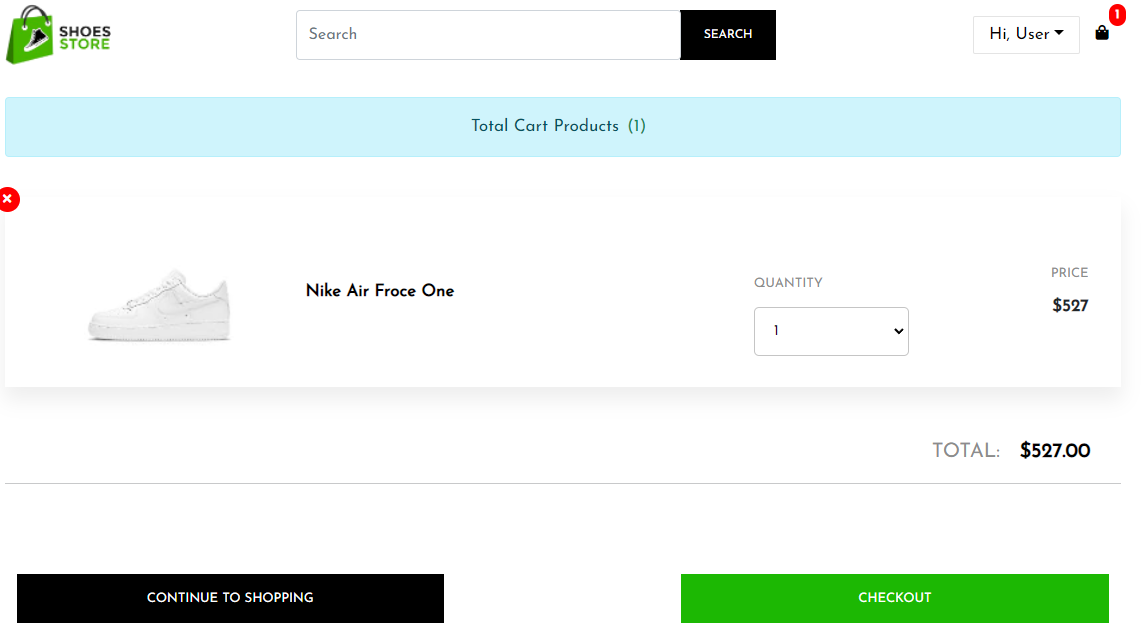
Hình : Trang chủ web

**Chi tiết sản phẩm:**

****

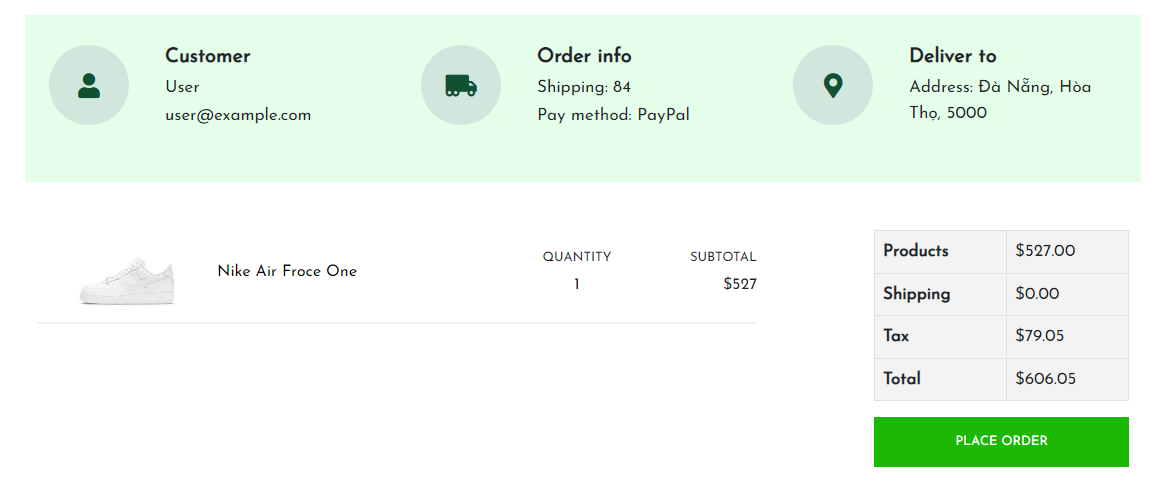
Hình : Chi tiết sản phẩm

**Giỏ hàng :**

****

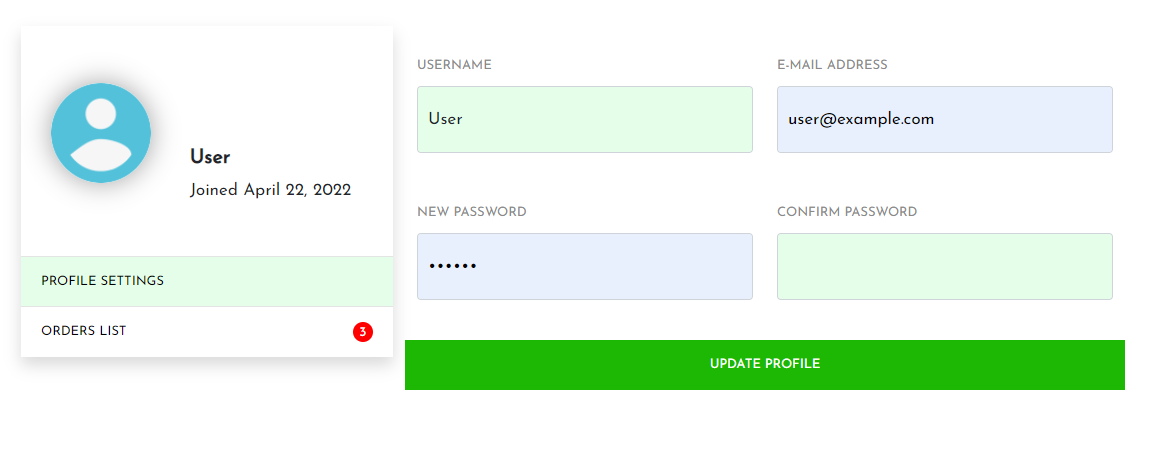
Hình : Trang giỏ hàng

**Thanh toán:**

****

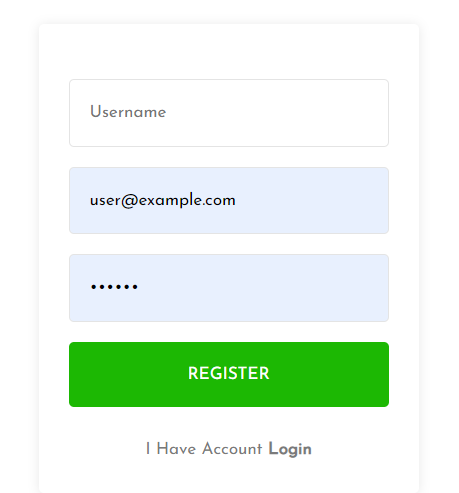
Hình : Trang thanh toán

**Hồ sơ tài khoản:**

****

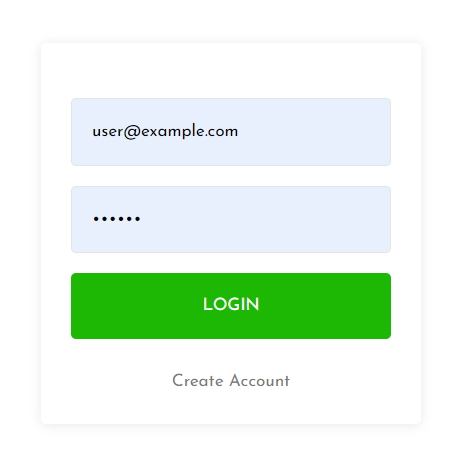
Hình : Hồ sơ đăng ký

**Đăng ký:**

****

Hình : Trang đăng ký

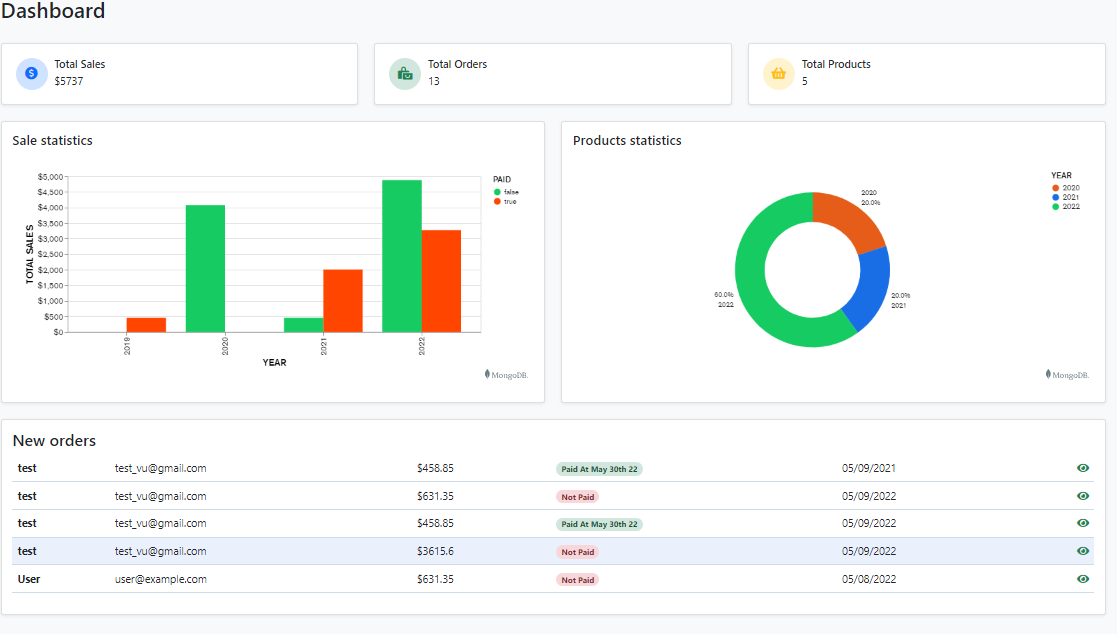
**Đăng nhập:**

****

Hình : Trang đăng nhập

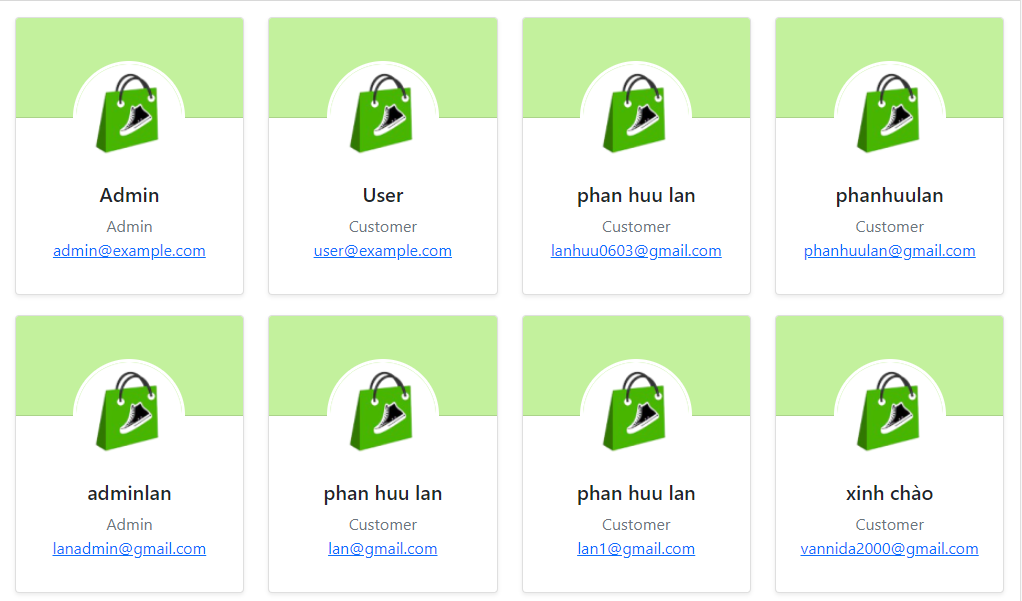
### 3.3.2. Giao diện admin:

**Bảng điều khiển:**

****

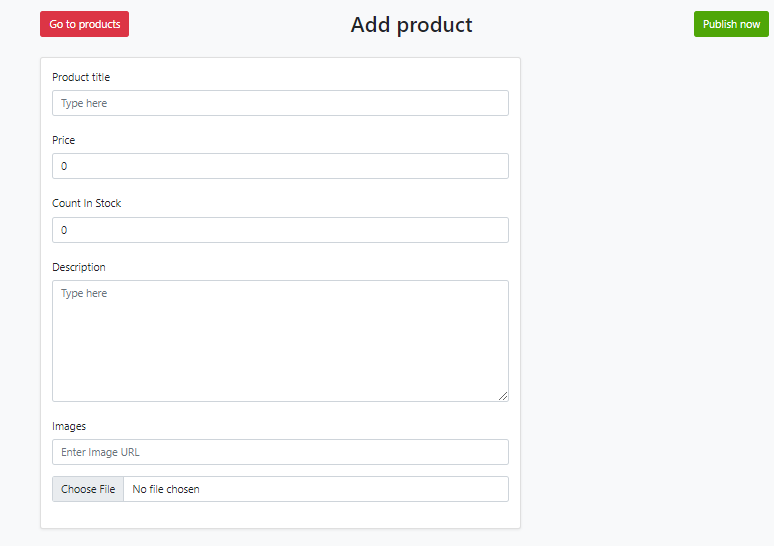
Hình : Bảng điều khiển (Admin)

**Danh sách thành viên:**

****

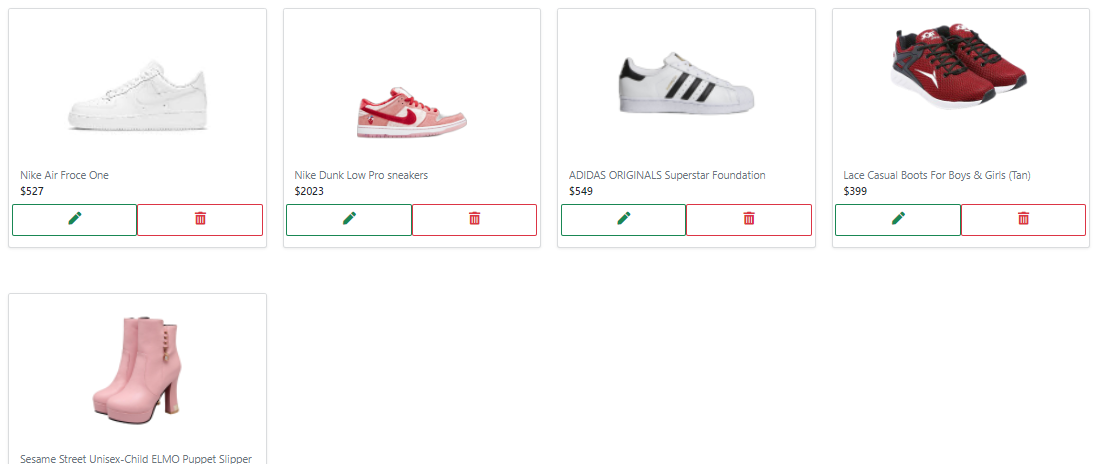
Hình : Danh sách thành viên

**Thêm sản phẩm:**

****

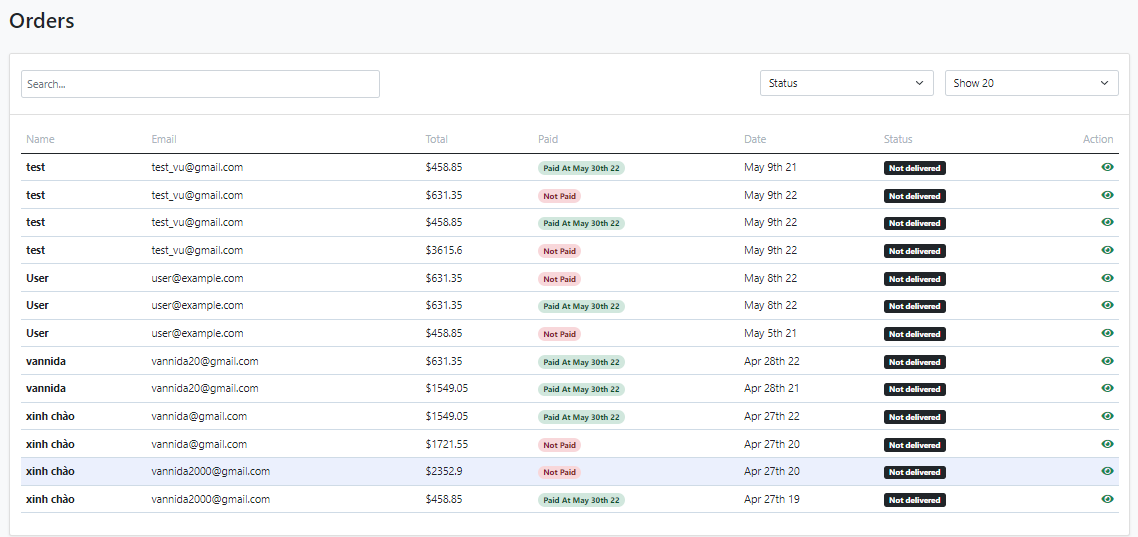
Hình : Trang thêm sản phẩm

**Xóa, sửa sản phẩm:**

****

Hình : Trang xóa sửa sản phẩm

**Quản lý đơn hàng:**

****

Hình : Trang quản lý đơn hàng

# HƯỚNG PHÁT TRIỂN

Sau một thời gian thực hiện đề tài này, cùng với sự hướng dẫn giảng viên chúng em đã hiểu được phần nào về mua bán giày qua mạng . Mặc dù chúng em muốn làm cho chương trình này tối ưu và khả thi hơn nữa nhưng trình độ của chúng em còn hạn chế nên không thể không tránh được những sai thiếu sót. Rất mong được ý kiến đóng góp bổ sung từ phía quy thầy cô và các bạn đặc biệt là những người sử dụng chương trình này.

Nhóm chúng em sẽ tìm ra nhiều giải pháp, phướng pháp làm cho website trở nên tối ưu hơn, nâng cao trải nghiệm người dùng, giao diện thân thiện dễ tiếp cận hơn. Nhóm em mong muốn wedsite sẽ phát triển, mở rộng quy mô hơn nữa để đến tay khách hàng. khi người dùng đến với website, họ biết ngay đây là ai, là về gì, sản phẩm làm về gì, và liệu họ có nên ở lạị hay tiếp tục tìm kiếm một website khác.

# BÀI HỌC KINH NGHIỆM

Để xây dựng được API phong phú và chuyên nghiệp thì có rất nhiều điều chúng ta cần xem xét từ khi bắt đầu, và cần phải tốn nhiều thời gian tìm hiểu và tự học. Cần sự phối hợp, tinh thần làm việc nhóm cao giữa các thành viên.

Để chuẩn bị phát triển dự án web REST API mà chia nhóm ra thành 2 team. Team frontend thì tập trung làm giao diện, gọi vào REST API để lấy ra dữ liệu. Team backend thì làm REST API, truy vấn cơ sở dữ liệu để trả ra dữ liệu cho team frontend.

Có 3 cách để 2 team phối hợp với nhau:

* Một là team frontend clone project của team backend về và chạy project của team backend song song
* Hai là team backend sử dụng ngrok để chuyển đổi localhost sang một domain public mà team frontend có thể dễ dàng gọi vào. (Có thể tìm hiểu ngrok [tại đây](https://ngrok.com/docs" \t "_blank))
* Cách thứ 3 là team backend xây dựng một API online đẩy lên Internet để team frontend ở bất cứ đâu có thể gọi trực tiếp vào đó.

Ở cách thứ 1 có thể đảm bảo 2 bên kết nối nhau an toàn mà không dính đến các vấn đề như cross-site request vì cùng dùng chung một máy localhost. Tuy nhiên cách này khiến cho một bên phụ thuộc vào bên kia, team frontend clone project về mỗi lần bật máy lên phải khởi chạy project của team backend. Giả sử team frontend là những người không biết gì về backend, không biết cách chạy thì team backend phải hướng dẫn từng bước cho team frontend chạy dự án của mình. Ngoài ra nếu mỗi bên làm việc ở xa việc hướng dẫn sẽ càng khó khăn.

Cách thứ 2 và 3 thì đều dính đến vấn đề cross-site request, vì team frontend gọi API ở máy local tới một domain hoàn toàn khác. Điều này khiến team backend phải tạm thời nới lỏng bảo mật trong quá trình develop để team fronend có thể gọi API và chọc được vào session cookie.

Ở cách 2 ưu điểm là team backend mỗi lần cập nhật và chạy code team frontend có thể nhận được ngay. Tuy nhiên điều này phụ thuộc team backend phải bật ngrok 24/24. Nếu tắt ngrok đi thì domain public đang gọi vào sẽ biến mất. Ngoài ra sau khi bật lại ngrok sẽ lại sinh ra domain mới hoàn toàn. Điều đó khiến cho team frontend phải chỉnh sửa lại toàn bộ domain ở từng API một.

Cách thứ 3 thì tiện lợi, team frontend có thể kết nối API bất cứ lúc nào mà không phải chỉnh lại nhiều. Tuy nhiên không như cách 2, chỉ khi team backend đẩy code mình lên thì team frontend mới gọi được thêm các API mới. Ngoài ra khi tạo ra một API online trên Internet trong trạng thái nới lỏng bảo mật có thể dễ dàng bị tấn công.

Tuy nhiên, không có cách nào không có rủi ro, không có những bất lợi riêng. Ta cần cân nhắc, xét đến nhiều yếu tố và làm sao 2 team càng ít phụ thuộc vào nhau càng tốt. Dưới đây quan điểm của nhóm em sau khi hoàn thành đồ án:

* Nếu tất cả đều là lập trình viên fullstack, có thể clone project chạy bình thường mà không gặp phải vấn đề khó khăn gì thì cách 1 là điều khuyến khích vì tính bảo mật của nó.
* Trường hợp khác nếu hai bên luôn làm việc cùng một thời gian, thời điểm như nhau thì cách 2 là điều khuyến khích. Vì 2 bên có thể kết nối trực tiếp, dễ dàng, nhanh chóng và khả năng bị tấn công cũng rất thấp nhờ domain sinh ngẫu nhiên.
* Trường hợp hai bên làm việc ở xa như đang đại dịch ngồi nhà làm, mỗi bên có chuyên môn riêng, thời gian làm việc ít bị quản thúc. Khi đó cách 3 sẽ là tiện nhất.

# TÀI LIỆU THAM KHẢO

https://code.tutsplus.com/vi/tutorials/code-your-first-api-with-nodejs-and-express-understanding-rest-apis--cms-31697

https://code.tutsplus.com/vi/tutorials/code-your-first-api-with-nodejs-and-express-set-up-the-server--cms-31698

<https://websolutions.com.vn/mongodb-la-gi-tong-quan-ve-mongodb/>

<https://vi.reactjs.org/tutorial/tutorial.html>

https://pokimobile.vn/api-web-la-gi-1639501048#Web\_API\_la\_gi

https://blog.codegym.vn/2020/12/24/uu-va-nhuoc-diem-cua-web-api/

<https://cloud.mongodb.com/>

**\*\*\*\*\*\*\*\*\*\*\*\*\* HẾT\*\*\*\*\*\*\*\*\*\*\***