**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

*Trà Vinh, ngày ….. tháng …… năm ……*

**Giáo viên hướng dẫn**

*(Ký tên và ghi rõ họ tên)*

*Trà Vinh, ngày ….. tháng …… năm ……*

**Giáo viên hướng dẫn**

*(Ký tên và ghi rõ họ tên)*

**NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG**

*Trà Vinh, ngày ….. tháng …… năm ……*

**Thành viên hội đồng**

*(Ký tên và ghi rõ họ tên)*

**LỜI CẢM ƠN**

Đầu tiên, em xin gửi lời cảm ơn chân thành đến Ban Giám hiệu nhà trường, Ban lãnh đạo Khoa Kỹ Thuật và Công Nghệ, Trường Đại Học Trà Vinh đã tạo điều kiện tốt nhất để em có thể hoàn thành đồ án cơ sở ngành của mình.

Đồng thời, em xin gửi lời cảm ơn sâu sắc và chân thành nhất đến cô Phạm Thị Trúc Mai, cô đã hướng dẫn em trong suốt quá trình thực hiện đồ án, cô luôn tận tình hướng dẫn, chỉ bảo và cung cấp cho em những kiến thức chuyên môn, kỹ năng làm việc và truyền đạt cho em những kinh nghiệm quý báo. Cô là người đã đưa ra những nhận xét, góp ý vô cùng quý giá để giúp em hoàn thiện đồ án một cách tốt nhất.

Em cũng chân thành cảm ơn gia đình, tập thể bạn bè đã luôn giúp đỡ và động viên em trong suốt quá trình học tập cũng như thực hiện đồ án.

Tuy đã cố gắng hoàn thành đồ án trong phạm vi và khả năng cho phép, nhưng chắc chắn sẽ không tránh khỏi những thiếu sót, em rất mong nhận được sự góp ý, nhận xét của quý thầy cô và các bạn để nhờ vào đó em có thể tự cải thiện bản thân và chất lượng của đồ án.

Em xin chân thành cảm ơn!

*Trà Vinh, ngày ….. tháng …… năm ……*

**Sinh viên thực hiện**

*(Ký tên và ghi rõ họ tên)*

**MỤC LỤC**

[CHƯƠNG 1: TỔNG QUAN 8](#_Toc155429986)

[1.1 Giới thiệu 8](#_Toc155429987)

[1.2 Mục tiêu cần đạt được 8](#_Toc155429988)

[1.3 Hướng giải quyết 8](#_Toc155429989)

[CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT 10](#_Toc155429990)

[2.1 API 10](#_Toc155429991)

[2.1.1 API miễn phí 11](#_Toc155429992)

[2.1.2 Các API miễn phí sử dụng trong đề tài 12](#_Toc155429993)

[2.2 Vite 13](#_Toc155429994)

[2.2.1 Đặc điểm quan trọng 13](#_Toc155429995)

[2.2.2 Hướng dẫn cài đặt Vite 15](#_Toc155429996)

[2.3 Material UI 15](#_Toc155429997)

[2.4 Axios 17](#_Toc155429998)

[2.4.1 Đặc điểm quan trọng 18](#_Toc155429999)

[2.4.2 Cách cài đặt 19](#_Toc155430000)

[2.4.3 Cách cài đặt 19](#_Toc155430001)

[2.5 Ngôn ngữ lập trình JavaScript/HTML/CSS 21](#_Toc155430002)

[2.5.1 JavaScript 21](#_Toc155430003)

[2.5.2 HTML và CSS 22](#_Toc155430004)

[2.6 JSON 23](#_Toc155430005)

[2.7 Localhost 23](#_Toc155430006)

[2.8 Port 23](#_Toc155430007)

[CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU 23](#_Toc155430008)

[3.1 Xây dựng website minh họa 23](#_Toc155430009)

[3.1.1 Mô tả website 23](#_Toc155430010)

[3.1.2 Cài đặt môi trường 24](#_Toc155430011)

[3.1.3 Khởi tạo dự án 25](#_Toc155430012)

[3.1.4 Xây dựng giao diện 27](#_Toc155430013)

[CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU 30](#_Toc155430014)

[4.1 Kết quả đạt được 30](#_Toc155430015)

[4.2 Hạn chế của đề tài 30](#_Toc155430016)

[CHƯƠNG 5: HƯỚNG PHÁT TRIỂN 31](#_Toc155430017)

[DANH MỤC TÀI LIỆU THAM KHẢO 32](#_Toc155430018)

[PHỤ LỤC 33](#_Toc155430019)

**DANH MỤC HÌNH ẢNH**

[*Hình 2.1 Mô hình hoạt động của API* 11](#_Toc155431760)

[*Hình 2.2 Sơ đồ mô tả cách hoạt động của API* 13](#_Toc155431761)

[*Hình 2.3 Cài đặt Vite* 15](#_Toc155431761)

[*Hình 2.4 Cài đặt Axios* 19](#_Toc155431761)

[*Hình 2.5 Ví dụ sử dụng Axios* 20](#_Toc155431761)

[*Hình 2.6 Mô hình hoạt động JavaScript* 22](#_Toc155431761)

[*Hình 2.7 Ví dụ JSON* 23](#_Toc155431761)

[*Hình 3.1 Cài đặt và sử dụng Node.js* 27](#_Toc155431761)

[*Hình 3.2 Khởi tạo dự án* 27](#_Toc155431761)

[*Hình 3.3 Chọn framework cho dự án* 28](#_Toc155431761)

[*Hình 3.4 Chọn ngôn ngữ cho dự án* 28](#_Toc155431762)

[*Hình 3.5 Cài đặt các thư viện, dependencies cho dự án* 29](#_Toc155431762)

[*Hình 3.6 Cấu trúc thư mục dự án* 29](#_Toc155431762)

[*Hình 3.7 Giao diện website* 30](#_Toc155431762)

[*Hình 3.8 Giao diện website* 31](#_Toc155431762)

[*Hình 3.9 Giao diện website* 32](#_Toc155431762)

**DANH MỤC TỪ VIẾT TẮT**

|  |  |  |
| --- | --- | --- |
| STT | Ký hiệu chữ viết tắt | Chữ viết đầy đủ |
| 1 | API | Application Programming Interface |
| 2 | CSRF | Cross-site Request Forgery |
| 3 | CSS | Cascading Style Sheets |
| 4 | ECMA | European Computer Manufacturer's Association |
| 5 | HMR | Hot Module Replacement |
| 6 | HTML | HyperText Markup Language |
| 7 | HTTP | HyperText Transfer Protocol |
| 8 | IP | Internet Protocol |
| 9 | JSON | JavaScript Object Notation |
| 10 | REST | Representational State Transfer |
| 11 | RPC | Remote Procedure Call |
| 12 | SOAP | Simple Object Access Protocol |
| 13 | UI | User Interface |
| 14 | URL | Uniform Resource Locator |
| 15 | XML | Extensible Markup Language |

**TÓM TẮT ĐỒ ÁN CƠ SỞ NGÀNH**

Đồ án nhằm giới thiệu về Travel Advisor API, GeoDB Cities API, Weather API, Exchange Rates API và cách ứng dụng vào dự án website du lịch.

Sử dụng ViteJS và Material UI để triển khai và xây dựng website du lịch với các chức năng cơ bản như: tìm kiếm những địa điểm du lịch theo thành phố, hiển thị thông tin thời tiết, quy đổi tiền tệ cho người dùng.

Kết hợp sử dụng thư viện Axios và Fetch API để gửi yêu cầu và lấy dữ liệu phản hồi từ API.

Tìm hiều về Travel Advisor API, GeoDB Cities API, Weather API, Exchange Rates API thông qua Internet và cách ứng dụng chúng để thực hiện các chức năng trên.

Với mong muốn được tìm hiểu, học hỏi và sau thời gian nghiên cứu, thực hiện, tôi đã hoàn thành việc ứng dụng các API miễn phí để xây dựng website du lịch với các chức năng trên.

**MỞ ĐẦU**

Cùng với sự phát triển mạnh mẽ của khoa học và công nghệ thế giới, việc sử dụng API (Application Programming Interface) đã trở thành một phần quan trọng trong quá trình phát triển và ứng dụng và website. API không chỉ giúp kết nối và tương tác giữa các hệ thống khác nhau mà còn mở ra những cơ hội đáng kể cho các nhà phát triển.

Trong dự án này, tôi đã tập trung vào nghiên cứu và ứng dụng các API miễn phí như: Travel Advisor API, GeoDB Cities API, Weather API, Exchange Rates API. Đồng thời xây dựng một website du lịch minh họa để thể hiện sức mạnh và sự linh hoạt của những dịch vụ này.

Chúng ta sẽ tìm hiểu cách API có thể được tích hợp để cung cấp thông tin, dịch vụ, hoặc tương tác với các tài nguyên ngoại vi một cách thuận tiện và hiệu quả. Qua việc xây dựng một trang web minh họa, tôi mong muốn truyền đạt cách sử dụng API một cách sáng tạo và áp dụng chúng trong các ứng dụng thực tế.

Dự án này không chỉ là một hành trình khám phá về công nghệ API, mà còn là cơ hội để chúng ta đặt ra câu hỏi về cách chúng có thể tối ưu hóa trải nghiệm người dùng và mở rộng khả năng của một trang web.

Vì những nhu cầu cần thiết đấy nên tôi đã chọn đề tài “Tìm hiểu một số API miễn phí phổ biến hiện nay vào ứng dụng và dự án web minh họa”. Đề tài này giúp tôi có thêm các kiến thức về cách hoạt động của API cũng như cách ứng dụng chúng. Đồng thời đề tài cũng là một cơ hội học tập, trải nghiệm giúp bản thân tôi tích lũy được các kiến thức và kỹ năng chuyên môn cho công việc trong tương lai.

# CHƯƠNG 1: TỔNG QUAN

## Giới thiệu

Trong bối cảnh cuộc sống ngày nay, việc xây dựng và phát triển các ứng dụng web đòi hỏi sự linh hoạt, tương tác và tích hợp thông tin từ nhiều nguồn khác nhau. Để đáp ứng nhu cầu này, API (Application Programming Interface) đã trở thành trụ cột không thể thiếu, giúp kết nối và truyền thông giữa các hệ thống khác nhau. Mặc dù có nhiều loại API khác nhau, tuy nhiên mục đích của dự án này là tập trung vào việc nghiên cứu và ứng dụng API miễn phí.

Vấn đề chính mà tôi hướng đến là hiểu rõ cách API miễn phí có thể được tích hợp một cách hiệu quả trong quá trình phát triển website. API miễn phí không chỉ là nguồn cung cấp thông tin mà còn mang lại lợi ích về chi phí, đặc biệt là cho các dự án học tập, khởi nghiệp và cá nhân.

Ngoài ra, tôi cũng đặt ra câu hỏi về cách tích hợp API một cách sáng tạo nhằm tối ưu hóa trải nghiệm người dùng trên website. Sự linh hoạt và khả năng tương tác của API mở ra những khả năng mới, từ việc cung cấp thông tin đến việc tạo ra các trải nghiệm động và độc đáo.

Qua việc giải quyết những thách thức này, tôi hi vọng rằng dự án sẽ không chỉ mang lại cái nhìn tổng quát về API miễn phí mà còn cung cấp hướng đẫn thực tế cho việc tích hợp chúng vào quá trình phát triển website, đặc biệt là trong ngữ cảnh của các dự án nhỏ và tài nguyên hạn chế.

## Mục tiêu cần đạt được

Xây dựng được website du lịch với các chức năng hoàn chỉnh, minh họa được việc ứng dụng API vào website.

Học thêm những kiến thức mới, cải thiện trình độ bản thân, nâng cao khả năng tự học và giải quyết vấn đề.

## Hướng giải quyết

Nghiên cứu cách hoạt động của API, tìm hiểu và xác định các API miễn phí phù hợp với mục tiêu của dự án.

Nghiên cứu lý thuyết: Tìm hiểu về các công cụ lập trình và các ngôn ngữ lập trình (JavaScript, ViteJS, Material UI).

Phân tích và so sánh các tính năng, hạn chế và điều kiện sử dụng của từng API.

Lựa chọn framework, thư viện phù hợp cho việc xây dựng giao diện website minh họa.

Phát triển các chức năng tích hợp với các API miễn phí đã chọn.

Thiết kế giao diện người dùng thân thiện và dễ sử dụng.

# CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT

## API

API là cơ chế cho phép 2 thành phần phần mềm giao tiếp với nhau bằng một tập hợp các định nghĩa và giao thức. Ví dụ: hệ thống phần mềm của cơ quan thời tiết chứa dữ liệu về thời tiết hàng ngày. Ứng dụng thời tiết trên điện thoại sẽ “trò chuyện” với hệ thống này qua API và hiển thị thông tin cập nhật về thời tiết hàng ngày trên điện thoại. [1]

Kiến trúc API thường được giải thích dưới dạng máy chủ và máy khách. Ứng dụng gửi yêu cầu được gọi là máy khách, còn ứng dụng gửi phản hồi được gọi là máy chủ. Như vậy, trong ví dụ về thời tiết, cơ sở dữ liệu của cơ quan thời tiết là máy chủ còn ứng dụng di động là máy khách.

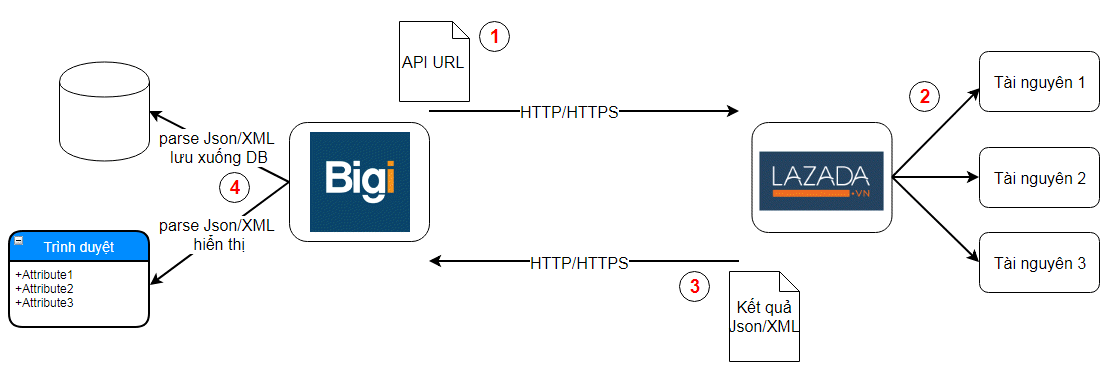
API hoạt động theo 4 cách khác nhau, tùy vào thời điểm và lý do chúng được tạo ra.

API SOAP: Các API này sử dụng Giao thức truy cập đối tượng đơn giản. Máy chủ và máy khách trao đổi thông đệp bằng XML. Đây là loại API kém linh hoạt được dùng phổ biến trước đây.

API RPC: Những API này được gọi là lệnh gọi thủ tục từ xa. Máy khách hoàn thành một hàm (hoặc thủ tục) trên máy chủ còn máy chủ gửi kết quả về cho máy khách.

API Websocket: API Websocket là một bản phát triển API web hiện đại khác sử dụng các đối tượng JSON để chuyển dữ liệu. API WebSocket hỗ trợ hoạt động giao tiếp hai chiều giữa ứng dụng máy khách và máy chủ. Máy chủ có thể gửi thông điệp gọi lại cho các máy khách được kết nối, điều này khiến loại API này hiệu quả hơn API REST.[1]

API REST Đây là loại API phổ biến và linh hoạt nhất trên web hiện nay. Máy khách gửi yêu cầu đến máy chủ dưới dạng dữ liệu. Máy chủ dùng dữ liệu đầu vào từ máy khách này để bắt đầu các hàm nội bộ và trả lại dữ liệu đầu ra cho máy khác.



*Hình 2.1 Mô hình hoạt động của API*

API đóng vai trò quan trọng trong việc tạo ra sự kết nối giữa các ứng dụng và hệ thống khác nhau, giúp chúng tương tác và làm việc cùng nhau một cách hiệu quả.

### API miễn phí

**Ưu điểm của việc sử dụng API miễn phí**

Tiết kiệm chi phí: API miễn phí giúp giảm bớt áp lực tài chính, đặc biệt là đối với các doanh nghiệp nhỏ, dự án cá nhân hoặc những người mới bắt đầu tìm hiểu về API.

Dễ dàng bắt đầu: Sử dụng API miễn phí thường đi kèm với quy trình đăng ký đơn giản, không yêu cầu chi phí ban đầu và thủ tục biên lai phức tạp, giúp người phát triển dễ dàng bắt đầu tích hợp chúng vào ứng dụng của mình.

Hỗ trợ cộng đồng: Nhiều API miễn phí được cung cấp bởi cộng đồng và do đó có sự hỗ trợ từ cồng đồng người sử dụng và nhà phát triển, có thể mang lại sự linh hoạt và cải thiện liên tục.

Tích hợp nhanh chóng: Do không có chi phí đầu vào, việc tích hợp API miễn phí thường nhanh chóng và giảm áp lực về tài nguyên.

**Hạn chế của việc sử dụng API miễn phí**

Hạn chế sử dụng: Các dịch vụ API miễn phí thường có giới hạn về số lần truy cập, dung lượng hoặc chức năng. Điều này có thể tạo ra rắc rối khi ứng dụng phát triển và cần nhiều tài nguyên hơn.

Không ổn định: API miễn phí có thể không được đảm bảo về độ ổn định và hiệu suất, đặc biệt là khi có nhiều người sử dụng cùng một lúc.

Thay đổi điều kiện sử dụng: Các nhà cung cấp API có thể thay đổi điều kiện sử dụng bất cứ lúc nào, bao gồm cả việc chuyển từ miễn phí sang trả phí hoặc giảm quyền truy cập.

Bảo mật: Một số API miễn phí có bảo mật yếu tố, và việc chia sẻ dữ liệu với dịch vụ không tin cậy có thể tăng nguy cơ về bảo mật.

Thiếu hỗ trợ: So với phiên bản trả phí, API miễn phí thường không được hỗ trợ chăm sóc khách hàng nhiều, điều này có thể là vấn đề khi phát sinh sự cố.

Tùy thuộc vào mục tiêu cụ thể của dự án, việc sử dụng API miễn phí có thể là một lựa chọn tốt, nhưng nhà phát triển cũng cần xem xét và đánh giá các hạn chế để đảm bảo rằng nó phù hợp với yêu cầu và kỳ vọng của họ.

### Các API miễn phí sử dụng trong đề tài

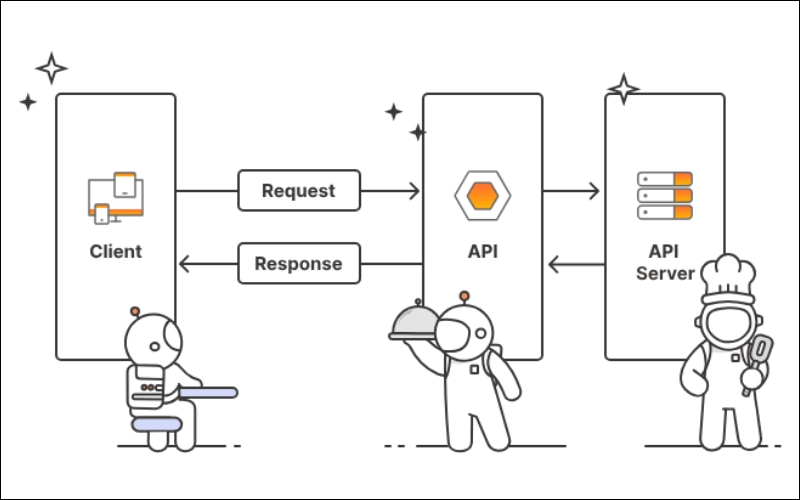
Trong quá trình tìm kiếm các API miễn phí phù hợp với dự án, tôi đã quyết định tiến hành nghiên cứu các API miễn phí sau để ứng dụng vào dự án website minh họa: Travel Advisor API, GeoDB Cites API, Weather API, Exchange Rates API.

Về chức mặt chức năng, các API miễn phí này đảm bảo cung cấp tương đối đầy đủ thông tin giúp tôi xây dựng các chức năng trên website minh họa. Cụ thể Travel Advisor API cung cấp thông tin du lịch và đánh giá người dùng trên toàn thế giới, cho phép truy cập đánh giá điểm du lịch, nhà hàng, khách sạn và các hoạt động giải trí. GeoDB cities API cung cấp thông tin về các thành phố trên khắp thế giới, cho phép tìn kiếm thông tin về thành phố, quốc gia, vùng lãnh thổ và các đặc điểm địa lý khác. Weather API cung cấp thông tin thời tiết hiện tại và dự báo trong tương lai, cho phép truy cập nhiệt độ, độ ẩm, tốc độ gió và dự báo thời tiết. Exchange Rates API cung cấp thông tin về tỷ giá hối đoái giữ các loại tiền tệ, cho phép truy cập tỷ giá hối đoái hiện tại và lịch sử tỷ giá.

Về dữ liệu đầu ra của các loại API này là dữ liêu JSON với các loại thông tin tương ứng.

Về ưu điểm của các API này là chúng cung cấp tương đối đầy đủ dữ liêu cho người dùng cơ bản, miễn phí và dễ dàng sử dụng.

Về mặt hạn chế các API miễn phí này đều giới hạn số lần truy cập và tính năng trong tài khoản miễn phí, yêu cầu xác thực API key, thời gian gửi phản hồi dữ liệu tương đối chậm hơn so với các API trả phí.



*Hình 2.2 Sơ đồ mô tả cách hoạt động của API*

## Vite

Vite là một công cụ xây dựng ứng dụng web hiệu suất cao được thiết kế để phát triển các ứng dụng Vue.js và cả các ứng dụng JavaScript khác. Tên "Vite" có nghĩa là "nhanh chóng" trong tiếng Pháp và đúng như tên gọi, Vite tập trung vào việc cung cấp một trải nghiệm phát triển nhanh chóng và hiệu suất tốt. [2]

### Đặc điểm quan trọng

Hiệu suất cao: Vite được xây dựng để có hiệu suất tối ưu trong quá trình phát triển và khi triển khai ứng dụng. Sử dụng ESModule để tải mô-đun, cũng như sử dụng Rollup để gói gọn mã nguồn, giúp giảm thời gian tải và tăng tốc độ khởi chạy ứng dụng.

Dev Server với HMR: Vite cung cấp một máy chủ phát triển tích hợp với Hot Module Replacement (HMR), cho phép thấy những thay đổi ngay lập tức khi chỉnh sửa mã nguồn, mà không cần làm mới trang.

Tích hợp Vue.js: Vite được xây dựng đặc biệt để hỗ trợ Vue.js, một thư viện JavaScript phổ biến cho xây dựng giao diện người dùng. Có thể tận dụng các tính năng nâng cao của Vue.js và kết hợp chúng với hiệu suất tốt của Vite.

Plugin hệ sinh thái: Vite hỗ trợ hệ sinh thái plugin phong phú, cho phép mở rộng chức năng của nó và tích hợp với các công nghệ khác. Có nhiều plugin sẵn có để giúp xử lý các loại file, tối ưu hóa ảnh và nhiều tính năng khác.

ESModule Native: Sử dụng ESModule một cách tự nhiên, cho phép sử dụng cú pháp import/export của ECMAScript module mà không cần biên dịch.

Cấu hình dễ dàng: Với tệp cấu hình (vite.config.js) có thể dễ dàng tùy chỉnh cài đặt của dự án, chẳng hạn như đường dẫn đầu ra, cấu hình proxy và nhiều hơn nữa.

Hỗ trợ TypeScript: Vite hỗ trợ TypeScript mà không cần cấu hình phức tạp. Có thể bắt đầu sử dụng TypeScript một cách nhanh chóng trong dự án của mình.

Với những đặc điểm này, ViteJS đã trở thành một lựa chọn phổ biến cho các nhà phát triển web muốn xây dựng ứng dụng Vue.js với trải nghiệm phát triển hiệu suất cao.

Mặc dù ViteJS mang lại nhiều lợi ích và điểm mạnh cho quá trình phát triển ứng dụng web, nhưng như mọi công nghệ khác, nó cũng có một số điểm yếu. Dưới đây là một số điểm yếu của Vite.

Chưa ổn định hoàn toàn: Vite là một dự án mới so với các công cụ phát triển truyền thống và có thể chưa hoàn toàn ổn định như chúng. Điều này có thể gây ra vấn đề đối với các dự án lớn hoặc yêu cầu tính ổn định cao.

Hệ sinh thái plugin đang phát triển: Mặc dù có nhiều plugin hữu ích, nhưng hệ sinh thái plugin của Vite vẫn đang phát triển. Điều này có thể làm cho việc tìm kiếm plugin cho các yêu cầu cụ thể trở nên khó khăn hơn so với một số công cụ khác.

Cộng đồng chưa lớn: So với một số công cụ phát triển khác như Webpack, cộng đồng xung quanh Vite chưa lớn. Điều này có thể dẫn đến việc có ít tài nguyên và hỗ trợ hơn từ cộng đồng.

Chưa hỗ trợ hoàn chỉnh cho mọi loại dự án: Mặc dù Vite là một lựa chọn tốt cho nhiều loại dự án, nhưng có thể không phải là lựa chọn tốt nhất cho tất cả mọi người. Đối với các dự án lớn, phức tạp và có yêu cầu tích hợp với nhiều công nghệ, Vite có thể cần được tích hợp với các công cụ và quy trình khác.

Tích hợp đôi khi khó khăn: Tích hợp Vite vào một số môi trường phát triển hoặc dự án có sẵn có thể đôi khi gặp khó khăn, đặc biệt là nếu đang chuyển từ các công cụ khác.

Tất cả các điểm này không có nghĩa là Vite không là một lựa chọn tốt. Tùy thuộc vào yêu cầu cụ thể của dự án, những điểm yếu này có thể được quản lý hoặc bù đắp bằng cách sử dụng plugin và giải pháp thích hợp.

### Hướng dẫn cài đặt Vite

Để cài đặt Vite cần có Node.js phiên bản 12.0.0 trở lên. Dưới đây là các bước cài đặt Vite sử dụng npm.

Mở terminal và thực hiện lệnh sau để tạo một dự án mới.

A white rectangular object with a black border

Description automatically generated

*Hình 2.3 Cài đặt Vite*

## Material UI

Material UI là một thư viện giao diện người dùng (UI) React.js được xây dựng dựa trên hướng dẫn thiết kế từ Google, được gọi là Material Design. Thư viện này cung cấp một bộ các thành phần giao diện người dùng đã được xây dựng sẵn, giúp xây dựng ứng dụng React đẹp mắt và hiệu quả một cách nhanh chóng.[3]

**Ưu điểm của Material UI**

Thiết kế theo hướng đẫn Material Design: Material UI được thiết kế theo hướng dẫn thiết kế Material Design của Google. Điều này mang lại một giao diện người dùng hiện đại, sáng tạo và dễ sử dụng.

Bộ các thành phần giao diện được xây dựng sẵn: Material UI cung cấp một loạt các thành phần giao diện như nút, hộp văn bản, đầu chủ đề, hộp thoại, biểu đồ, bảng, thanh trượt và nhiều thành phần khác, giúp xây dựng giao diện người dùng một cách dễ dàng.

Tùy biến dễ dàng: Mặc dù Material UI cung cấp các thành phần giao diện đã được thiết kế trước, nhưng có thể dễ dàng tùy chỉnh chúng để phù hợp với thiết kế của ứng dụng của . Các thành phố giao diện có thể được điều chỉnh thông qua CSS và các thuộc tính khác.

Hỗ trợ Responsiveness: Material UI được xây dựng với việc hỗ trợ độ phản hồi, đảm bảo rằng ứng dụng của hoạt động tốt trên nhiều loại thiết bị và kích thước màn hình.

Hệ thống Grid linh hoạt: Material UI cung cấp hệ thống grid linh hoạt để sắp xếp các thành phố giao diện trên trang web của . Hệ thống này giúp tổ chức layout của ứng dụng một cách hiệu quả.

Hỗ trợ Theme: Có thể tận dụng tính năng theme trong Material UI để áp dụng chủ đề màu sắc và kiểu cho toàn bộ ứng dụng của mình, giúp đảm bảo sự nhất quán trong giao diện.

Hỗ trợ TypeScript: Material UI hỗ trợ TypeScript, điều này giúp viết mã an toàn hơn và sử dụng tính năng kiểm soát kiểu trong quá trình phát triển.

Material UI là một trong những thư viện UI phổ biến và được sử dụng rộng rãi trong cộng đồng React, giúp nhà phát triển xây dựng giao diện người dùng hiện đại và thân thiện mà không phải xây dựng lại từ đầu nhiều thành phố giao diện.

**Hạn chế của Material UI**

Mặc dù Material UI là một thư viện giao diện người dùng phổ biến và mạnh mẽ trong cộng đồng React, nhưng cũng có một số hạn chế và điểm yếu cần xem xét.

Dung lượng (Bundle size): Material UI có thể tăng dung lượng của ứng dụng vì nó cung cấp nhiều thành phần giao diện đã được thiết kế sẵn và các tính năng mạnh mẽ. Điều này có thể ảnh hưởng đến thời gian tải của ứng dụng, đặc biệt là khi dung lượng bundle trở nên lớn.

Tùy chỉnh khó khăn: Mặc dù Material UI có thể được tùy chỉnh, nhưng việc điều chỉnh một số tính năng có thể trở nên phức tạp. Một số điều chỉnh giao diện cụ thể hoặc yêu cầu mức độ kiểm soát cao có thể đòi hỏi nhiều công sức hơn.

Phụ thuộc vào CSS-in-JS: Material UI thường sử dụng các giải pháp CSS-in-JS như styled-components hoặc @emotion/styled. Điều này có thể là một thách thức đối với những người muốn sử dụng CSS truyền thống hoặc các giải pháp quản lý kiểu khác.

Thiếu một số tính năng tích hợp: Mặc dù Material UI cung cấp nhiều thành phần giao diện, nhưng có một số chức năng tích hợp có sẵn trong các thư viện khác không được hỗ trợ mặc định.

Theme phức tạp: Đối với những người mới sử dụng Material UI , việc hiểu và tùy chỉnh Theme có thể trở nên phức tạp. Việc quản lý một chủ đề với nhiều thuộc tính có thể đòi hỏi sự hiểu biết sâu rộng về hệ thống chủ đề của thư viện.

Yêu cầu kĩ năng CSS: Việc làm chủ thành thạo CSS có thể là một lợi thế khi làm việc với Material UI . Người dùng có thể cảm thấy khó khăn nếu họ không quen với CSS trong React.

Tất cả những hạn chế này không có nghĩa là Material UI không phù hợp cho dự án. Nó vẫn là một thư viện mạnh mẽ và phổ biến trong cộng đồng React, và các điểm yếu này có thể được quản lý hoặc bù đắp bằng cách sử dụng các giải pháp và kỹ thuật thích hợp.

## Axios

Axios là một thư viện HTTP Client dựa trên Promise dành cho Node.js và trình duyệt. Nó có tính đẳng hình (tức là cùng codebase có thể chạy trong cả trình duyệt và Node.js). Ở phía server thì nó sử dụng native module http trong Node.js, còn ở phía client (trình duyệt) thì nó sử dụng XMLHttpRequest.[4]

### Đặc điểm quan trọng

**Ưu điểm của Axios**

Gửi HTTP Requests: Axios được sử dụng để thực hiện các yêu cầu HTTP, bao gồm GET, POST, PUT, DELETE và các phương thức khác.

Tích hợp Promise: Hỗ trợ Promise, giúp xử lý dễ dàng các yêu cầu bất đồng bộ và xử lý lỗi một cách thuận tiện.

Interceptors: Cho phép gắn các hàm middleware trước và sau khi yêu cầu được gửi hoặc sau khi nhận được phản hồi.

Chức năng tự động chuyển đổi dữ liệu: Có khả năng tự động chuyển đổi dữ liệu giữa JSON và các định dạng dữ liệu khác.

Hỗ trợ tính năng cancel request: Cung cấp khả năng hủy các yêu cầu HTTP, giúp quản lý tài nguyên và tăng hiệu suất.

Tương thích với môi trường trình duyệt và Node.js: Axios có thể chạy trên cả môi trường trình duyệt và Node.js, điều này làm cho nó phù hợp để sử dụng trong cả ứng dụng web và máy chủ.

Bảo mật: Axios có tính năng tự động xử lý các phương thức bảo mật như token CSRF, giúp tăng cường bảo mật trong ứng dụng.

Tất cả những điểm trên làm cho Axios trở thành một lựa chọn phổ biến khi làm việc với yêu cầu HTTP trong ứng dụng web và React.

**Hạn chế của Axios**

Mặc dù Axios là một thư viện HTTP client mạnh mẽ và phổ biến, nhưng cũng có một số hạn chế và điểm yếu mà người phát triển cần xem xét.

Dung lượng (Bundle size): Axios có kích thước khá lớn so với một số thư viện HTTP client nhỏ hơn. Điều này có thể làm tăng dung lượng của ứng dụng, đặc biệt là nếu chỉ cần một số tính năng cơ bản.

Không tự động hỗ trợ hủy yêu cầu (Cancellation): Mặc dù Axios hỗ trợ chức năng hủy yêu cầu thông qua Cancel Token, nhưng điều này không được kích hoạt mặc định. Phải thực hiện các bước cụ thể để hủy bỏ yêu cầu và có thể làm cho mã nguồn trở nên phức tạp hơn.

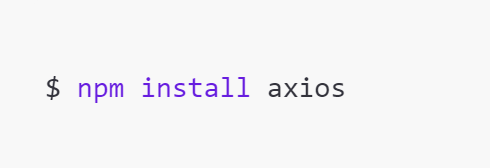
Không hỗ trợ JSONP: Dù Axios làm việc theo cú pháp Promise, nhưng nó vẫn có một lớp trừu tượng bổ sung khi so sánh với việc sử dụng fetch API mặc định trong JavaScript. Điều này có thể làm cho quá trình học hỏi trở nên phức tạp đối với những người mới bắt đầu.

Không hỗ trợ hình thức URL tốt hơn: Axios không hỗ trợ cú pháp URL mà các chuẩn gần đây như URLSearchParams hoặc URL.createObjectURL hỗ trợ. Điều này có thể là một hạn chế nếu muốn sử dụng những tính năng này mà không cần thư viện bên ngoài.

Tuy nhiên, những hạn chế này thường là tùy thuộc vào yêu cầu cụ thể của dự án và có thể được quản lý hoặc bù đắp bằng cách sử dụng các thư viện hoặc phương pháp khác nhau tùy thuộc vào tình huống cụ thể.

### Cách cài đặt

Cài đặt Axios thông qua npm trong terminal



*Hình 2.4 Cài đặt Axios*

### Cách cài đặt

Import Axios và sử dụng nó trong các file JavaScript hoặc TypeScript.

A screenshot of a computer program

Description automatically generated

*Hình 2.5 Ví dụ sử dụng Axios*

## Ngôn ngữ lập trình JavaScript/HTML/CSS

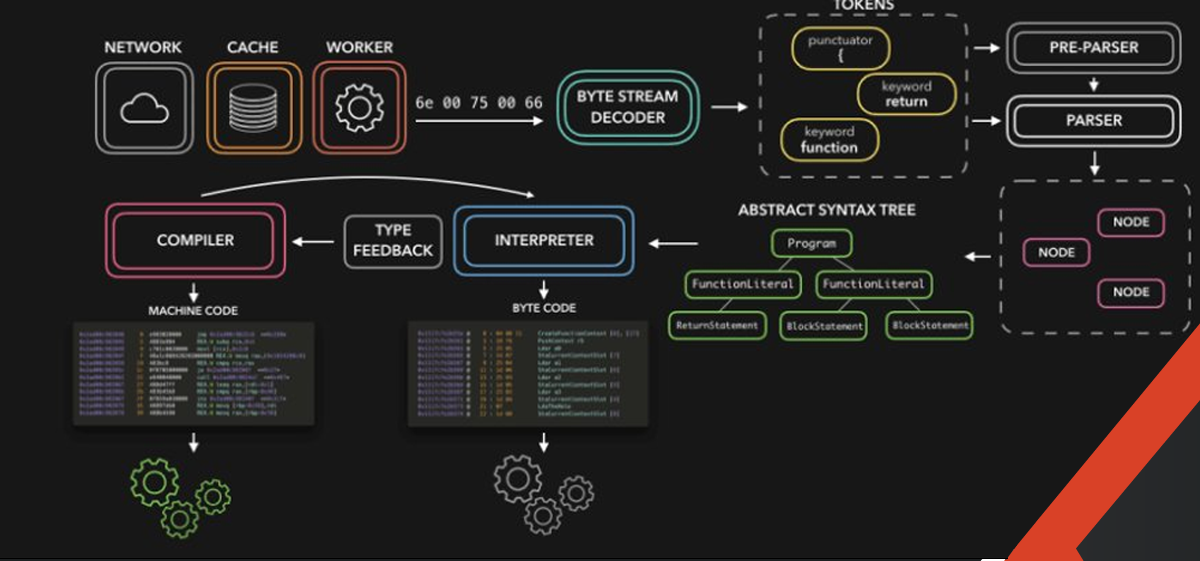
### JavaScript

JavaScript là ngôn ngữ lập trình được nhà phát triển sử dụng để tạo trang web tương tác. Từ làm mới bản tin trên trang mạng xã hội đến hiển thị hình ảnh động và bản đồ tương tác, các chức năng của JavaScript có thể cải thiện trải nghiệm người dùng của trang web. Là ngôn ngữ kịch bản phía máy khách, JavaScript là một trong những công nghệ cốt lõi của World Wide Web. Ví dụ: khi duyệt Internet, bất cứ khi nào thấy quảng cáo quay vòng dạng hình ảnh, menu thả xuống nhấp để hiển thị hoặc màu sắc phần tử thay đổi động trên trang web cũng chính là lúc thấy các hiệu ứng của JavaScript.[5]

Trước đây, các trang web có dạng tĩnh, tương tự như các trang trong một cuốn sách. Một trang tĩnh chủ yếu hiển thị thông tin theo một bố cục cố định và không làm được mọi thứ mà chúng ta mong đợi như ở một trang web hiện đại. JavaScript dần được biết đến như một công nghệ phía trình duyệt để làm cho các ứng dụng web linh hoạt hơn. Sử dụng JavaScript, các trình duyệt có thể phản hồi tương tác của người dùng và thay đổi bố cục của nội dung trên trang web.

Khi ngôn ngữ này phát triển hoàn thiện, các nhà phát triển JavaScript đã thiết lập các thư viện, khung và cách thức lập trình cũng như bắt đầu sử dụng ngôn ngữ này bên ngoài trình duyệt web. Ngày nay, có thể sử dụng JavaScript để thực hiện hoạt động phát triển cả ở phía máy khách và máy chủ.

Tất cả các ngôn ngữ lập trình đều hoạt động bằng cách dịch cú pháp giống tiếng Anh thành mã máy, hệ điều hành sau đó sẽ chạy mã này. JavaScript được phân loại khái quát là một ngôn ngữ viết tập lệnh hoặc một ngôn ngữ diễn giải. Mã JavaScript được diễn giải, tức là được công cụ JavaScript dịch trực tiếp sang mã ngôn ngữ máy cơ bản. Với các ngôn ngữ lập trình khác, trình biên dịch sẽ biên dịch toàn bộ mã thành mã máy trong một bước riêng. Như vậy, tất cả các ngôn ngữ viết tập lệnh đều là ngôn ngữ lập trình, nhưng không phải tất cả các ngôn ngữ lập trình đều là ngôn ngữ viết tập lệnh.



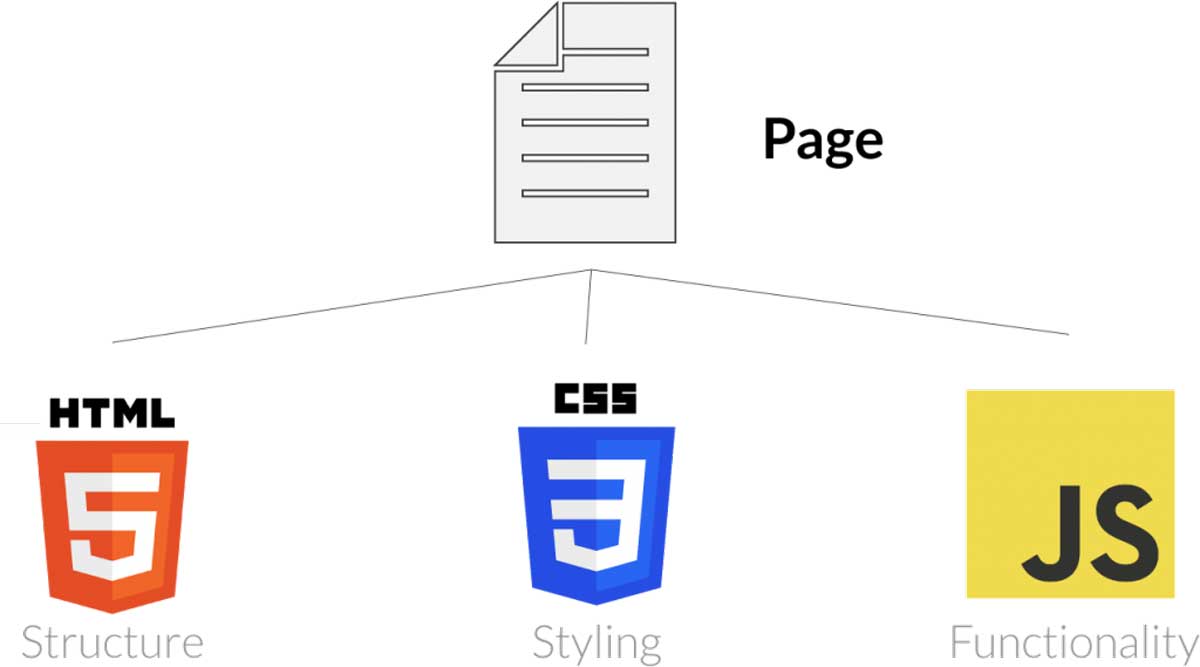
*Hình 2.6 Mô hình hoạt động JavaScript*

### HTML và CSS

Ngôn ngữ đánh dấu siêu văn bản (HTML) và ngôn ngữ định kiểu theo tầng (CSS) là hai ngôn ngữ lập trình khác mà các nhà phát triển sử dụng trong việc phát triển frontend. HTML là khối dựng cơ bản của hầu hết các trang web. Tất cả các đoạn văn, phần mục, hình ảnh, tiêu đề và văn bản đều được viết bằng HTML. Nội dung sẽ xuất hiện trên trang web theo thứ tự được viết bằng HTML.[5]

CSS là một ngôn ngữ của các quy tắc định kiểu mà chúng tôi sử dụng để định kiểu cho nội dung HTML của mình. Có thể sử dụng CSS để thiết kế các phần tử của trang web như màu nền, phông chữ, cột và đường viền.[5]

Cả ba ngôn ngữ cùng nhau tạo ra trải nghiệm người dùng tích cực trên bất kỳ trang web nào. Trong khi HTML và CSS chủ yếu có thể điều chỉnh nội dung tĩnh, chúng có thể tích hợp với mã JavaScript phía máy khách để cập nhật nội dung một cách linh hoạt.



*Hình 2.7 Tính liên quan giữa HTML, CSS và JavaScript*

## JSON

JSON là một định dạng dữ liệu nhẹ và độc lập ngôn ngữ. Tên đầy đủ là "JavaScript Object Notation" nhưng JSON không phụ thuộc vào JavaScript và được sử dụng rộng rãi trong nhiều ngôn ngữ lập trình khác.

JSON được tổ chức theo cặp "key-value" (khóa-giá trị) và có cú pháp tương tự với các ngôn ngữ lập trình khác như JavaScript, Python và Java.

Dữ liệu JSON có thể là một đối tượng (object), một mảng (array), một số, một chuỗi, một giá trị boolean, hoặc null.

A computer screen with green and blue text

Description automatically generated

*Hình 2.7 Ví dụ JSON*

Trong ví dụ này, có một đối tượng có các thuộc tính như "name", "age", "isStudent", "courses" và "address". Một số giá trị là chuỗi, một số, giá trị boolean, mảng, và một đối tượng con.

JSON là một định dạng dữ liệu linh hoạt, nhẹ và rất phổ biến trong phát triển web và truyền thông giữa các dịch vụ. Nó đã trở thành một phần quan trọng của các ứng dụng hiện đại và cung cấp một cách đơn giản để truyền tải và lưu trữ dữ liệu.

## Localhost

Localhost là một tên miền được sử dụng để chỉ đến địa chỉ IP loopback (127.0.0.1) trên máy tính. Trong ngữ cảnh phát triển web, localhost thường được sử dụng để đặt máy chủ web trên máy tính cục bộ để phát triển và kiểm thử ứng dụng web.

Khi phát triển một trang web hoặc ứng dụng web, thường cần một môi trường máy chủ để chạy mã nguồn và kiểm tra ứng dụng của mình.

Có thể thiết lập máy chủ cục bộ trên máy tính của mình thông qua localhost để chạy và kiểm thử ứng dụng web mà không cần triển khai trên môi trường sản xuất.

Khi chạy một ứng dụng web trên máy tính của mình và thiết lập máy chủ cục bộ, thường sẽ truy cập ứng dụng của mình qua địa chỉ http://localhost:port trong trình duyệt. Port thường là một số cụ thể, ví dụ: http://localhost:3000.

Khi sử dụng localhost, đảm bảo rằng các cổng được sử dụng là an toàn và không xung đột với các dịch vụ khác đang chạy trên máy tính.

## Port

Port (Cổng): Là một số nguyên dương được sử dụng để xác định các dịch vụ cụ thể trên một máy tính trong mạng.

Tại sao cần sử dụng Port: Máy tính có thể chạy nhiều dịch vụ và ứng dụng cùng một lúc. Sử dụng port giúp máy tính biết được dịch vụ hoặc ứng dụng nào đang được yêu cầu.

Port trong địa chỉ IP: Trong một địa chỉ IP, port thường được hiểu là một cổng nơi dữ liệu được gửi và nhận. Địa chỉ IP và port cùng nhau xác định một "địa chỉ đích" hoặc "địa chỉ nguồn" cho gói dữ liệu.

Có một số cổng được dùng phổ biến cho các dịch vụ cụ thể. Ví dụ, cổng 80 thường được sử dụng cho HTTP (truy cập web), cổng 443 cho HTTPS (truy cập web an toàn). Các cổng thường được quy định trong khoảng từ 0 đến 65535.

Ví dụ, http://localhost:3000 sử dụng cổng 3000. Cổng này có thể thay đổi tùy thuộc vào cách cấu hình môi trường phát triển của mình.

Port đóng vai trò quan trọng trong việc xác định và phân biệt các dịch vụ và ứng dụng trên một máy tính, đặc biệt là trong phát triển và triển khai các ứng dụng web.

# CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU

## Xây dựng website minh họa

### Mô tả website

Về mặt công nghệ: Website được thiết kế để giúp người dùng hiểu và thử nghiệm các API miễn phí một cách dễ dàng và thân thiện. Sự tích hợp giữa môi trường phát triển Visual Studio Code, framework Vite nhanh chóng, và giao diện Material UI tạo nên một trải nghiệm người dùng tốt và linh hoạt. Website tích hợp các API nổi tiếng như Travel Advisor, GeoDB Cities, Weather API và Exchange Rates API, giúp người dùng có cái nhìn toàn diện về các điểm đến và điều kiện thời tiết, cũng như tỷ giá hối đoái.

Về mặt chức năng: Website cho phép người dùng tìm kiếm các điểm đến dựa trên tên thành phố hoặc quốc gia sử dụng GeoDB Cities API. Hiển thị thông tin chi tiết về địa điểm, bao gồm hình ảnh, mô tả và đánh giá từ Travel Advisor API. Hiển thị thông tin thời tiết cho địa điểm được chọn, bao gồm các thông tin như nhiệt độ, tình trạng thời tiết, độ ẩm và tốc độ gió nhờ vào việc sử dụng Weather API. Cung cấp tỉ giá hối đoái giữa đồng tiền địa phương và một số đồng tiền quốc tế thông qua Exchange Rates API. Cho phép quy đổi tiền tệ dựa trên tỉ giá hối đoái.

Về mặt đối tượng sử dụng: Website được thiết kế, xây dựng nhầm phục vụ mục đích học tập, nghiên cứu các API miễn phí do đó đối tượng mà dự án hướng đến là những sinh viên, học sinh có nhu cầu nghiên cứu, học tập về cách sử dụng API và những người có sở thích du lịch, người muốn tìm kiếm thông tin chi tiết về các điểm đến, thời tiết, và tỉ giá hối đoái.

### Cài đặt môi trường

Đầu tiên để sử dụng được ViteJS ta cần phải cài đặt Node.js phiên bản 12.0.0 trở lên. Ở đây tôi cài đặt và sử dụng phiên bản Node.js 18.18.2.

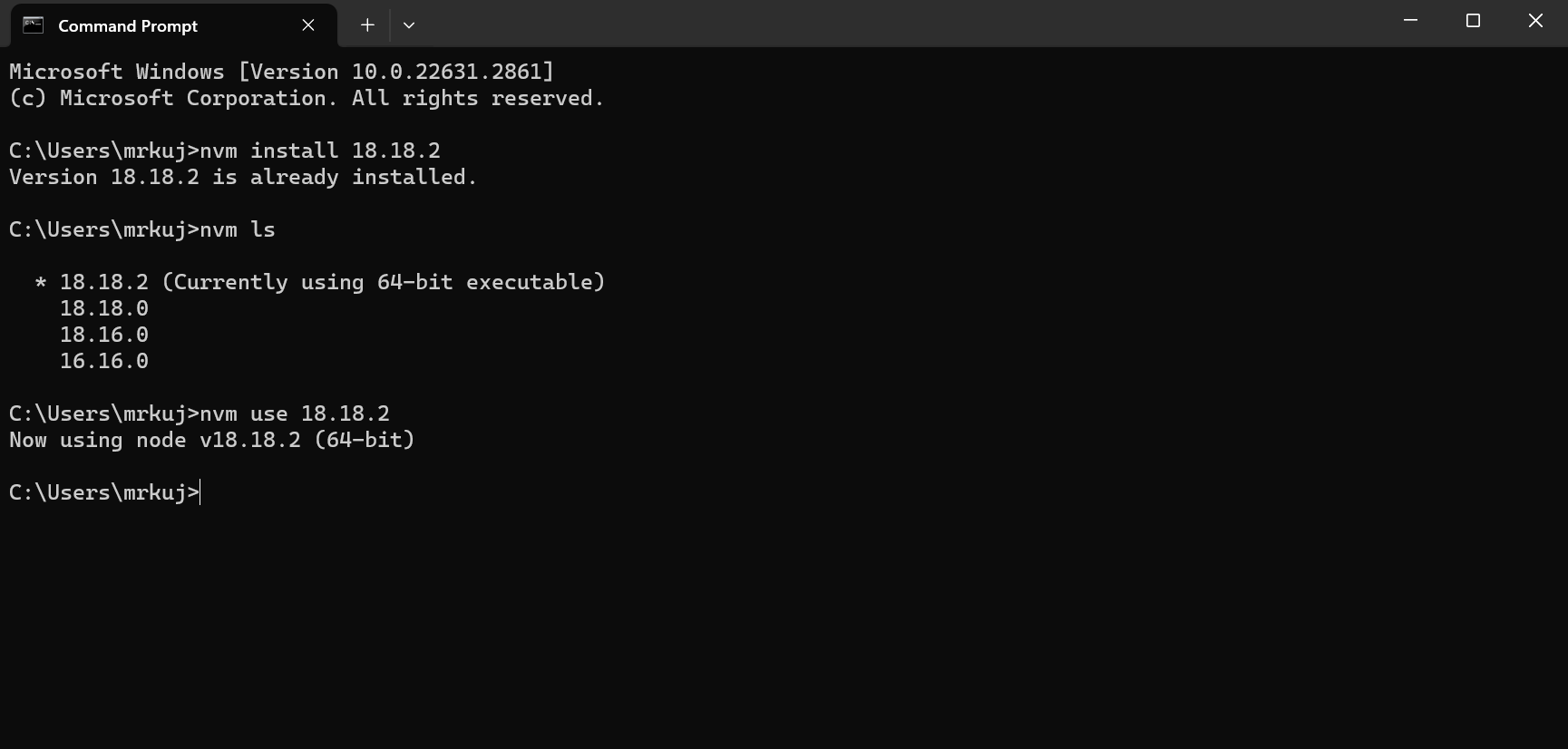
Sử dụng Node Version Manager (nvm) để tải và cài đặt phiên bản Node.js phù hợp. Truy cập vào đường link https://github.com/coreybutler/nvm-windows/releases để tải và tiến hành cài đặt nvm.

Sau khi cài đặt nvm, ta sử dụng Command Prompt để cài đặt Node.js 18.18.2 bằng các câu lệnh sau:

nvm install 18.18.2 (cài đặt Node.js phiên bản 18.18.2)

nvm ls (kiểm tra xem phiên phản Node.js đã được cài đặt thành công hay chưa)

nvm use 18.18.2 (sử dụng phiên bản Node.js 18.18.2)



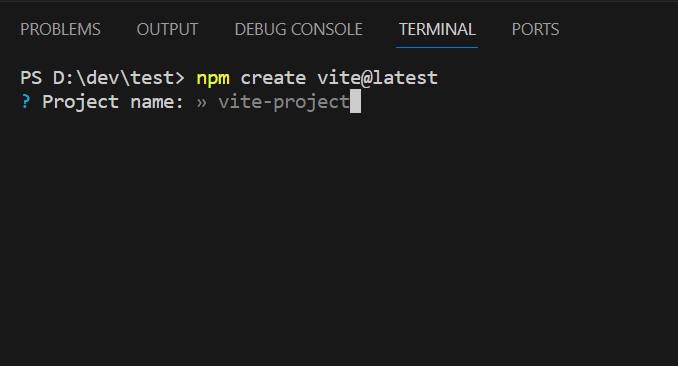
*Hình 3.1 Cài đặt và sử dụng Node.js*

Sau khi cài đặt thành công Nodej.js phiên bản 18.18.2, ta sử môi trường Visual Studio Code để tiến hành xây dựng giao diện website.

### Khởi tạo dự án

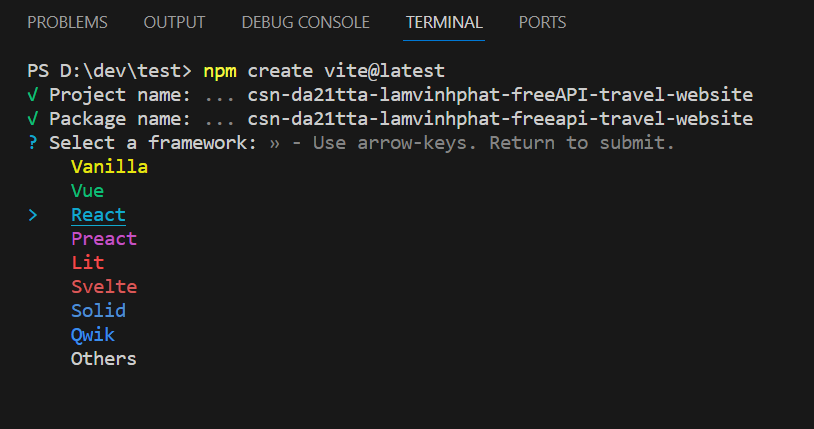
Để khởi tạo đự án ta sử dụng các lệnh trong terminal của môi trường Visual Studio Code.

Gõ lệnh nvm create vite@latest để khỏi tạo dự án.



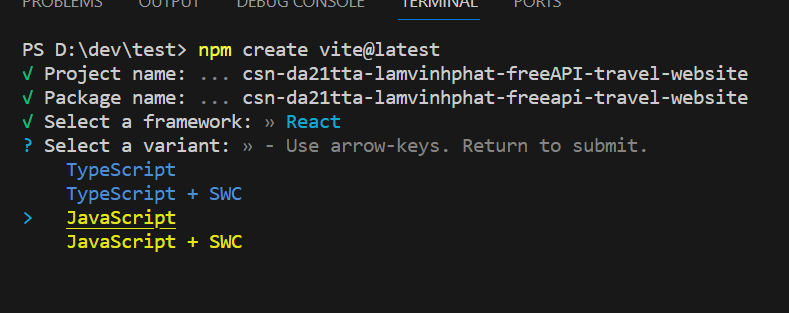
*Hình 3.2 Khởi tạo dự án*

Tiếp tục gõ tên dự án, ở đây tên dự án của tôi là csn-da21tta-lamvinhphat-freeAPI-travel-website.



*Hình 3.3 Chọn framework cho dự án*

Sử dụng phím mũi tên để chọn framework cho dự án, chọn React.

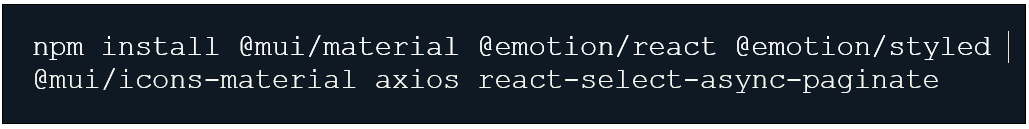


*Hình 3.4 Chọn ngôn ngữ cho dự án*

Chọn ngôn ngữ JavaScript để hoàn thành khởi tạo dự án.

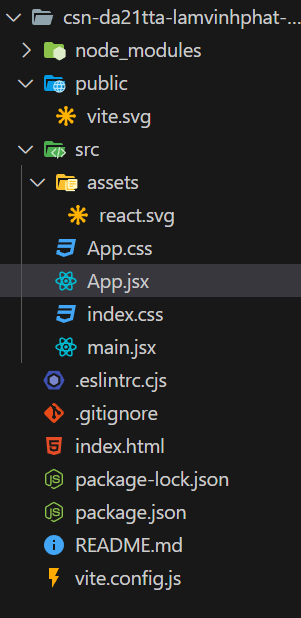
Sau khi khởi tạo dự án ta sử dụng lệnh cd csn-da21tta-lamvinhphat-freeAPI-travel-website để chuyển con trỏ vào thư mục dự án.

Tiếp tục cài đặt các thư viện, dependencies cho dự án bằng lệnh



*Hình 3.5 Cài đặt các thư viện, dependencies cho dự án*

Sau khi hoàng thành các bước trên ta sẽ được folder với cấu trúc như sau:



*Hình 3.6 Cấu trúc thư mục dự án*

### Xây dựng giao diện

Tiến hành xây dựng các components giao diện của website, dự án sử dụng Material UI để xây dựng giao diện người dùng một cách thân thiện và hiện đại. Giao diện của dự án được xây dựng dựa trên các components bao gồm:

Header component hiện thị câu khẩu hiệu của website và ô tìm kiếm autocomplete với chức năng tìm kiếm tên thành phố.

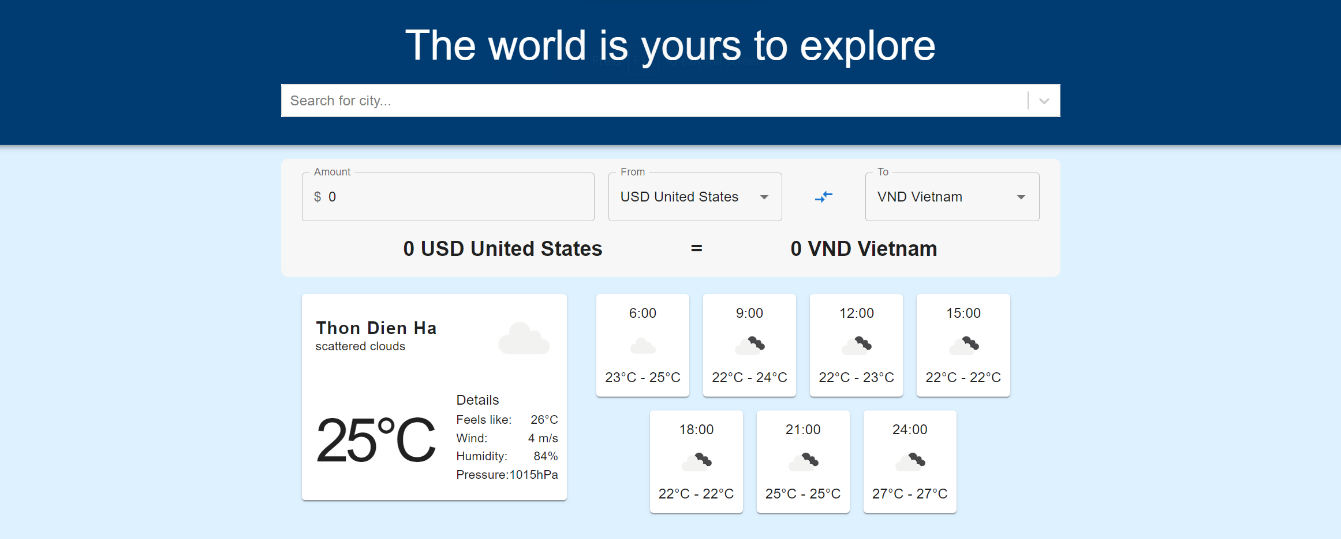
CurrencyConverter component hiện thị thông tin quy đổi tiền tệ giữa các loại tiền tệ được chọn.

Weather component hiện thị thông tin thời tiết theo giờ. Thông tin chi tiết về nhiệt độ, tốc độ gió, độ ẩm và áp lực.

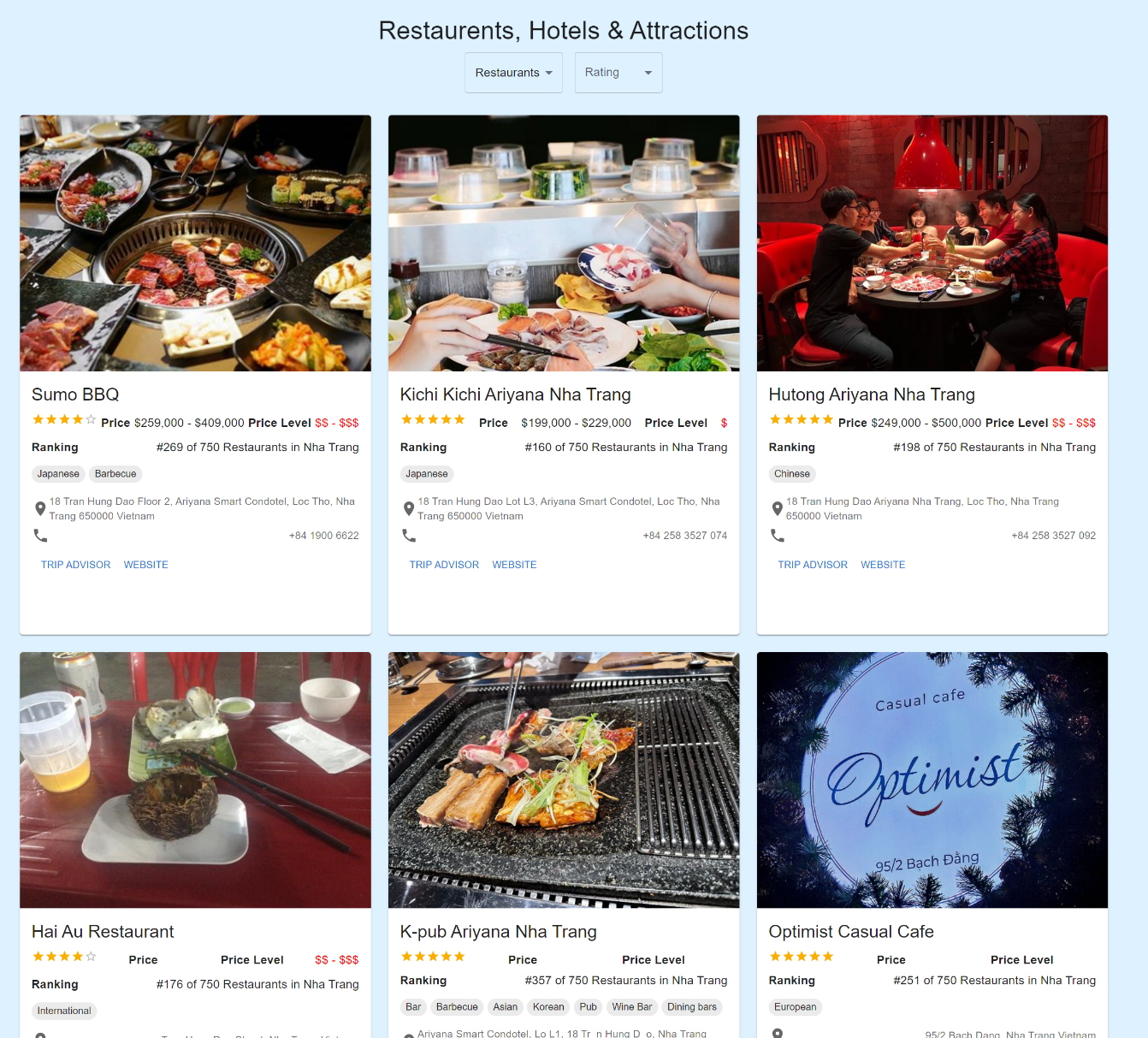
List component dùng để hiển thị thông tin địa điểm đã được lọc dựa trên đánh giá, nhà hàng, địa điểm du lịch hay khách sạn.

PlacesDetails component liệt kê danh sách các điểm đến, mô tả chi tiết điểm đến bao gồm đánh giá, hình ảnh, số điện thoại, địa chỉ, mức giá.

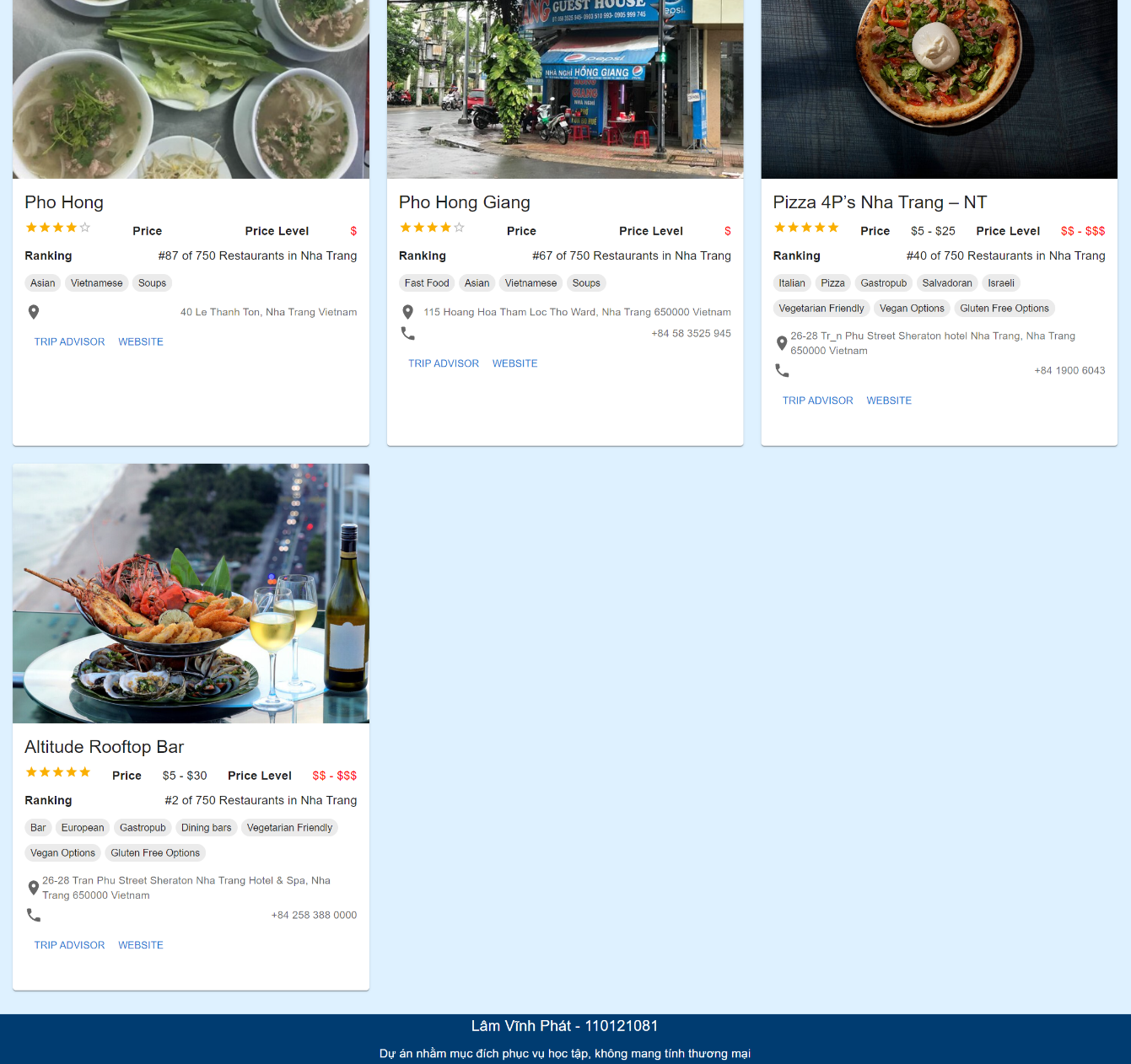
Footer component dùng để hiện thị thông tin của người phát triển dự án.



*Hình 3.7 Giao diện website*



*Hình 3.8 Giao diện website*



*Hình 3.9 Giao diện website*

# CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU

Sau một khoảng thời gian nghiên cứu, học tập và thực nghiệm. Đến nay đề tài “Tìm hiểu một số API miễn phí phổ biến hiện nay và ứng dụng vào dự án web minh họa.” đã hoàn thành về cơ bản theo đúng yêu cầu về nội dung và thời hạn. Trong quá trình nghiên cứu và thực hiện đề tài do những hạn chế về kinh nghiệm và kiên thức nên đề tài của tôi chắc chắn sẽ không thể tránh khỏi những thiếu sót. Tôi rất mong nhận được những ý kiến đóng góp từ quý thầy cô và các để đề tài được hoàn thiện hơn.

## Kết quả đạt được

**Về mặt công nghệ**

Xây dựng được một website minh họa tương đối hoàng chỉnh gồm giao diện và các chức năng tìm kiếm những địa điểm du lịch theo thành phố, hiển thị thông tin thời tiết, quy đổi tiền tệ cho người dùng.

Nắm vững được kiến thức về gọi API cũng như xử lý dữ liệu trả về.

Nâng cao kiến thức về giao diện Responsive, ViteJS, Material UI.

Công cụ sử dụng: Visual Studio Code.

**Về mặt kỹ năng mềm**

Kỹ năng tìm hiểu, phân tích, lên kế hoạch giải quyết vấn đề

Tính tự chủ trong công việc, khả năng tự học, tìm kiếm thông tin.

Học hỏi được những kinh nghiệm, kiến thức từ thầy cô và bè.

## Hạn chế của đề tài

Giao diện website còn khá đơn giản chưa được đẹp mắt.

Việc đặt tên file cũng như thư mục còn mang tính cá nhân hóa, làm cho người khác khó khăn khi tiếp cận với dự án.

Thiếu kinh nghiệm trong việc xây dựng giao diện website. Nội dung website còn hạn chế.

# CHƯƠNG 5: HƯỚNG PHÁT TRIỂN

Với những kiến thức tích lũy được trong quá trình nghiên cứu và thực nghiệm xây dựng đề tài. Tôi sẽ không ngừng trau dồi thêm kiến thức, kinh nghiệm để tiếp tục hoàn thiện đề tài. Khắc phục những hạn chế còn hiện hữu, bổ sung thêm các tính năng và tiện ích mới cho ứng dụng. Tích hợp các API mới như Google Maps API để hiển thị bản đồ và tạo trải nghiệm tìm kiếm dễ dàng hơn, Booking API để cung cấp dịch vụ đặt phòng khách sạn trực tuyến, Transportation API để cung cấp thông tin về phương tiện di chuyển như chuyến bay, tàu hỏa… Đảm bảo được đáp ứng đầy đủ nhu cầu cần thiết của người dùng.

Hiện tại đề tài chỉ dừng lại ở mức minh họa nên sau bài báo cáo này tôi sẽ cố gắng xây dựng được một ứng dụng website du lịch hoàn chỉnh dựa vào kiến thức đã học cũng như những kiến thức mới. Áp dụng được những kiến thức về API để tạo một cơ sở dữ liệu riêng cho mình, ứng dụng vào xây dựng một sản phẩm chỉn chu từ Front-end đến Back-end.

# DANH MỤC TÀI LIỆU THAM KHẢO

[1] API (giao diện lập trình ứng dụng) là gì?, website: https://aws.amazon.com/vi/what-is/api, truy cập ngày 20/12/2024.

[2] Vite Getting Started, webiste: https://vitejs.dev/guide, truy cập ngày 20/12/2024.

[3] Matertial UI Overview, website: https://mui.com/material-ui/getting-started, truy cập ngày 20/12/2024.

[4] Axios Getting Strated, website: https://axios-http.com/docs/intro, truy cập ngày 20/12/2024.

[5] JavaScript là gì?, webite: https://aws.amazon.com/vi/what-is/javascript, truy cập ngày 20/12/2024.

# PHỤ LỤC