TRƯỜNG ĐẠI HỌC TRÀ VINH

KHOA KỸ THUẬT VÀ CÔNG NGHỆ

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



ĐỒ ÁN MÔN HỌC

**PHÁT TRIỂN ỨNG DỤNG WEB VỚI MÃ NGUỒN MỞ**

**XÂY DỰNG WEBSITE BÁN LAPTOP**

|  |  |
| --- | --- |
| **Người hướng dẫn:**  Nguyễn Ngọc Đan Thanh | **Nhóm sinh viên thực hiện:**   1. 110121273 – Võ Trọng Nghĩa 2. 110121141 – Kim Hữu Ngân 3. 110121145 – Cao Khải Minh |

TRÀ VINH, THÁNG 1 NĂM 2025

TRƯỜNG ĐẠI HỌC TRÀ VINH

KHOA KỸ THUẬT VÀ CÔNG NGHỆ

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



ĐỒ ÁN MÔN HỌC

**PHÁT TRIỂN ỨNG DỤNG WEB VỚI MÃ NGUỒN MỞ**

**XÂY DỰNG WEBSITE BÁN LAPTOP**

|  |  |
| --- | --- |
| **Người hướng dẫn:**  Nguyễn Ngọc Đan Thanh | **Nhóm sinh viên thực hiện:**   1. 110121273 – Võ Trọng Nghĩa 2. 110121141 – Kim Hữu Ngân 3. 110121145 – Cao Khải Minh |

TRÀ VINH, THÁNG 1 NĂM 2025

# LỜI CẢM ƠN

Chúng em xin chân thành cảm ơn quý Thầy Cô Bộ môn Công nghệ Thông tin Khoa Kỹ thuật và Công nghệ đã tận tình hướng dẫn, truyền đạt kiến thức, kỹ năng chuyên môn và tạo điều kiện thuận lợi để chúng em hoàn thành đồ án môn học này.

Chúng em xin gửi lời cảm ơn đến cô Nguyễn Ngọc Đan Thanh người đã hướng dẫn, giúp đỡ và động viên em trong suốt quá trình thực hiện đồ án môn học này.

Cuối cùng, chúng em xin bày tỏ lòng biết ơn sâu sắc đến gia đình, bạn bè và những người đã luôn ủng hộ, động viên chúng em trong suốt quá trình học tập và thực hiện đồ án môn học này.

Chúng em xin chân thành cảm ơn !

# MỤC LỤC

[CHƯƠNG 1. TỔNG QUAN NGHIÊN CỨU 5](#_Toc183954377)

[1.1. Tổng quan về đề tài 5](#_Toc183954378)

[1.2. Tổng quan các hệ thống 5](#_Toc183954379)

[1.2.1. Laptop Giá Sỉ 5](#_Toc183954380)

[1.2.2. MemoryZone 5](#_Toc183954381)

[1.2.3. Laptop123 6](#_Toc183954382)

[1.2.4. Ngọc Laptop 8](#_Toc183954383)

[1.2.5. SV STORE 9](#_Toc183954384)

[1.3. Công nghệ sử dụng 11](#_Toc183954385)

[1.4. Mục tiêu nghiên cứu 12](#_Toc183954386)

[1.5. Phạm vi nghiên cứu 12](#_Toc183954387)

[1.6. Phương pháp nghiên cứu 12](#_Toc183954388)

[1.6.1. Phương pháp nghiên cứu lý thuyết 12](#_Toc183954389)

[1.6.2. Phương pháp nghiên cứu thực nghiệm 13](#_Toc183954390)

[CHƯƠNG 2. NGHIÊN CỨU LÝ THUYẾT 14](#_Toc183954391)

[2.1. Node.js 14](#_Toc183954392)

[2.1.1. Khái niệm 14](#_Toc183954393)

[2.1.2. Ứng dụng của Node.js 14](#_Toc183954394)

[2.1.3. Nguyên lý hoạt động của Node.js 15](#_Toc183954395)

[2.2. MongoDB 16](#_Toc183954396)

[2.2.1. Ưu điểm của MongoDB 17](#_Toc183954397)

[2.2.2. Nhược điểm của MongoDB 17](#_Toc183954398)

[2.3. Express JS 18](#_Toc183954399)

[2.3.1. Khái niệm 18](#_Toc183954400)

[2.3.2. Đặc điểm của Express.js 18](#_Toc183954401)

[2.3.3. Ưu điểm của Express.js 19](#_Toc183954402)

[2.4. RESTful API 20](#_Toc183954403)

[2.4.1. Khái niệm 20](#_Toc183954404)

[2.4.2. RESTful hoạt động như thế nào 21](#_Toc183954405)

[2.4.3. API RESTful mang lại những lợi ích 21](#_Toc183954406)

[2.5. React 22](#_Toc183954407)

[2.5.1. Khái niệm 22](#_Toc183954408)

[2.5.2. Những đặc điểm nổi bật của React 22](#_Toc183954409)

[2.5.3. Các thành phần quan trọng trong ReactJS 23](#_Toc183954410)

[2.5.4. Những lợi ích tuyệt vời mà ReactJS mang lại cho lập trình viên 24](#_Toc183954411)

[2.5.5. Các tính năng nổi bật của ReactJS 25](#_Toc183954412)

[2.5.6. Cách sử dụng ReactJS trong phát triển web 26](#_Toc183954413)

[CHƯƠNG 3. HIỆN THỰC HÓA NGHIÊN CỨU 27](#_Toc183954414)

[3.1. Mô tả bài toán 27](#_Toc183954415)

[3.2. Phân tích đặc tả hệ thống 27](#_Toc183954416)

[3.2.1. Đặc tả yêu cầu chức năng 27](#_Toc183954417)

[3.2.2. Đặc tả yêu cầu phi chức năng 28](#_Toc183954418)

[3.3. Thiết kế hệ thống 29](#_Toc183954419)

[3.3.1. Kiến trúc hệ thống 29](#_Toc183954420)

[3.3.2. Thiết kế dữ liệu 29](#_Toc183954421)

[3.3.2.1 Mô hình dữ liệu 37](#_Toc183954422)

[3.3.2.2 Danh sách các thực thể và mối kết hợp 37](#_Toc183954423)

[3.3.2.3 Chi tiết thực thể và mối kết hợp 37](#_Toc183954424)

[3.3.3. Thiết kế xử lý 38](#_Toc183954425)

[3.3.3.1 Mô hình DFD mức ngữ cảnh 38](#_Toc183954426)

[3.3.3.2 Mô hình DFD cấp 1 38](#_Toc183954427)

[3.3.3.3 Mô hình DFD cấp 2 38](#_Toc183954428)

[3.3.4. Thiết kế giao diện 38](#_Toc183954429)

[3.3.4.1 Sơ đồ website 38](#_Toc183954430)

[3.3.4.2 Giao diện trang … 38](#_Toc183954431)

[3.3.4.3 Giao diện … 38](#_Toc183954432)

[CHƯƠNG 4. KẾT QUẢ NGHIÊN CỨU 38](#_Toc183954433)

[4.1. Bộ dữ liệu thử nghiệm 38](#_Toc183954434)

[4.2. Kết quả thực nghiệm 38](#_Toc183954435)

[4.2.1. Chức năng … 38](#_Toc183954436)

[4.2.2. Chức năng … 38](#_Toc183954437)

[4.2.3. Chức năng … 38](#_Toc183954438)

[CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 38](#_Toc183954439)

[5.1. Kết luận 38](#_Toc183954440)

[5.2. Hướng phát triển 38](#_Toc183954441)

# DANH MỤC HÌNH ẢNH

[Hình 2.1 Tiêu đề của hình 4](#_Toc181621202)

[Hình 2.2 Gõ tiêu đề cho cái hình 5](#_Toc181621203)

# DANH MỤC BẢNG BIỂU

[Bảng 3.1 Danh sách các thực thể và mối kết hợp 6](#_Toc181621211)

[Bảng 3.2 Chi tiết thực thể sinhvien 6](#_Toc181621212)

# TỔNG QUAN NGHIÊN CỨU

## Tổng quan về đề tài

Với sự phát triển không ngừng của công nghệ thông tin, nhu cầu mua sắm trực tuyến ngày càng tăng, đặc biệt trong lĩnh vực thiết bị công nghệ như laptop. Laptop đã trở thành một công cụ thiết yếu cho học tập, công việc, và giải trí. Việc xây dựng một website bán laptop không chỉ giúp kết nối giữa người bán và người mua mà còn mang lại nhiều tiện ích như so sánh sản phẩm, tìm kiếm thông tin, và thanh toán dễ dàng.

Website bán laptop sẽ là một nền tảng giúp người dùng có thể lựa chọn sản phẩm phù hợp với nhu cầu và tài chính của mình một cách thuận tiện, dễ dàng hơn.

## Tổng quan các hệ thống

### Laptop Giá Sỉ

Tác giả: Đội ngũ phát triển của Laptop Giá Sỉ

URL: https://laptopgiasi.vn/danh-muc/laptop/

Nội dung: Cung cấp thông tin và bán các loại laptop với mức giá sỉ.

Hạn chế:

Thiếu tính năng so sánh sản phẩm: Không có tùy chọn cho phép người dùng so sánh các sản phẩm cùng loại, làm giảm khả năng đưa ra quyết định mua sắm.

Thiếu bộ lọc tìm kiếm: Trang web có thể không cung cấp đủ các bộ lọc (như theo giá, thương hiệu, hoặc thông số kỹ thuật), gây khó khăn cho việc tìm kiếm sản phẩm phù hợp.

Không có đánh giá sản phẩm: Thiếu phần đánh giá và nhận xét từ người dùng, làm giảm độ tin cậy của thông tin.

Thiếu tính năng xem nhanh: Người dùng không thể xem nhanh thông tin sản phẩm mà không cần truy cập vào trang riêng.

### MemoryZone

Tác giả: Đội ngũ phát triển của MemoryZone

URL: https://memoryzone.com.vn/laptop?q=collections:2828130&view=grid

Nội dung: Trang web chuyên cung cấp và bán các sản phẩm laptop, với nhiều lựa chọn từ các thương hiệu khác nhau. Giao diện được thiết kế thân thiện và dễ dàng cho người dùng tìm kiếm sản phẩm.

Hạn chế:

Thiếu tính năng so sánh:Không có chức năng so sánh trực tiếp giữa các sản phẩm, điều này hạn chế khả năng ra quyết định của người dùng.

Bộ lọc tìm kiếm hạn chế: Mặc dù có tính năng tìm kiếm, nhưng bộ lọc có thể không đủ chi tiết (như phân loại theo kích thước, trọng lượng, hay thời gian bảo hành).

Thiếu đánh giá từ người dùng: Trang không cung cấp đánh giá hoặc nhận xét từ người dùng, làm cho việc lựa chọn sản phẩm khó khăn hơn.

Không có hướng dẫn sử dụng: Thiếu thông tin hoặc video hướng dẫn sử dụng cho các sản phẩm, ảnh hưởng đến sự tin tưởng của người dùng.

### Laptop123

Tác giải: Công Ty TNHH Thương Mại Dịch Vụ Tin Học Xuân Hàm

Laptop123 là một trang web chuyên cung cấp các sản phẩm laptop, giúp người tiêu dùng dễ dàng tìm kiếm, so sánh và mua sắm laptop trực tuyến. Trang web nhắm đến việc cung cấp thông tin chi tiết về sản phẩm, hỗ trợ người tiêu dùng trong việc đưa ra quyết định mua hàng.

URL: https://laptop123.com.vn/lien-he.html

**Công nghệ sử dụng:**

Ngôn ngữ lập trình:

HTML, CSS, JavaScript: Để xây dựng giao diện người dùng.

React.js hoặc Vue.js: Nếu có tính năng tương tác cao, giúp cải thiện trải nghiệm người dùng.

Backend:

Node.js: Xử lý các yêu cầu từ người dùng và tương tác với cơ sở dữ liệu.

Django (Python) hoặc Ruby on Rails: Các framework có thể được sử dụng để phát triển ứng dụng web nhanh chóng và dễ bảo trì.

Cơ sở dữ liệu:

MySQL hoặc PostgreSQL: Lưu trữ thông tin sản phẩm, người dùng và đơn hàng.

Máy chủ web:

Apache hoặc Nginx: Để phục vụ trang web đến người dùng.

Công nghệ bảo mật:

SSL (Secure Sockets Layer): Đảm bảo rằng dữ liệu giao dịch giữa người dùng và trang web được mã hóa an toàn.

Ưu điểm

Danh mục sản phẩm rõ ràng: Giúp người dùng dễ dàng tìm kiếm và lựa chọn sản phẩm theo nhu cầu.

Thông tin chi tiết: Cung cấp đánh giá và thông số kỹ thuật đầy đủ, hỗ trợ người dùng trong quá trình quyết định mua sắm.

Tính năng tìm kiếm hiệu quả: Có thể cho phép người dùng tìm kiếm sản phẩm theo nhiều tiêu chí khác nhau (giá, thương hiệu, tính năng).

Nhược điểm

Thiếu sự đa dạng về sản phẩm: Sản phẩm không phong phú so với các trang thương mại điện tử lớn, hạn chế sự lựa chọn cho người dùng.

Giao diện chưa hiện đại: Thiết kế giao diện có thể không thu hút, ảnh hưởng đến trải nghiệm người dùng.

Thiếu các tính năng so sánh sản phẩm: Không có công cụ cho phép người dùng so sánh các sản phẩm tương tự, điều này có thể khiến quyết định mua sắm khó khăn hơn.

Kết luận

Để cải thiện vị thế cạnh tranh của Laptop123, cần tập trung vào việc mở rộng danh mục sản phẩm, nâng cao giao diện người dùng, và tích hợp các tính năng hỗ trợ mua sắm như so sánh sản phẩm và dịch vụ người dùng tốt hơn.

### Ngọc Laptop

Tác giải: CÔNG TY CỔ PHẦN BÁN LẺ NGỌC NGUYỄN

Ngọc Laptop là một trang web bán laptop kết hợp giữa bán hàng online và cửa hàng vật lý. Điều này cho phép người dùng không chỉ mua sắm trực tuyến mà còn có thể trải nghiệm sản phẩm trực tiếp tại cửa hàng. Trang web nhằm cung cấp thông tin đầy đủ và dịch vụ hỗ trợ cho người dùng, từ tư vấn sản phẩm đến bảo trì.

URL: https://ngocnguyen.vn/?srsltid=AfmBOopXtOGHnab5eM98NOXvZI4-GBTkPqR1PFDOTFrjGmNGh95G6AMJ

**Công nghệ sử dụng:**

Ngôn ngữ lập trình:

HTML, CSS, JavaScript: Cơ bản để xây dựng giao diện người dùng.

React.js hoặc Vue.js: Nếu có tính năng tương tác cao, giúp cải thiện trải nghiệm người dùng.

Backend:

Node.js: Được sử dụng để xử lý yêu cầu từ người dùng và tương tác với cơ sở dữ liệu.

Django (Python) hoặc Ruby on Rails: Các framework có thể được sử dụng để phát triển ứng dụng web nhanh chóng và dễ bảo trì.

Cơ sở dữ liệu:

MySQL hoặc PostgreSQL: Lưu trữ thông tin sản phẩm, người dùng và đơn hàng.

Máy chủ web:

Apache hoặc Nginx: Để phục vụ trang web đến người dùng.

Công nghệ bảo mật:

SSL (Secure Sockets Layer): Đảm bảo rằng dữ liệu giao dịch giữa người dùng và trang web được mã hóa an toàn.

Ưu điểm

Kết hợp bán hàng online và cửa hàng vật lý: Người dùng có thể trải nghiệm sản phẩm trước khi mua, tăng tính tin cậy và sự hài lòng.

Dịch vụ tư vấn và bảo trì tốt: Đội ngũ nhân viên có thể hỗ trợ người dùng một cách tận tình, từ việc tư vấn sản phẩm đến bảo trì sau khi mua.

Khả năng tương tác với người dùng: Người dùng có thể dễ dàng đặt câu hỏi và nhận phản hồi nhanh chóng.

Nhược điểm

Trang web chưa tối ưu hóa cho mua sắm online: Giao diện và tính năng có thể không thân thiện hoặc dễ sử dụng, ảnh hưởng đến trải nghiệm người dùng.

Thiếu tính năng so sánh sản phẩm: Người dùng không thể dễ dàng so sánh các sản phẩm khác nhau, điều này có thể gây khó khăn trong việc đưa ra quyết định mua hàng.

Khả năng cập nhật thông tin sản phẩm: Có thể không thường xuyên cập nhật thông tin về sản phẩm mới hoặc khuyến mãi, khiến người dùng không biết đến những lựa chọn tốt hơn.

Kết luận

Để nâng cao khả năng cạnh tranh và cải thiện trải nghiệm người dùng, Ngọc Laptop có thể tập trung vào việc tối ưu hóa giao diện trang web, tích hợp tính năng so sánh sản phẩm, và thường xuyên cập nhật thông tin sản phẩm.

https://ngocnguyen.vn/?srsltid=AfmBOopXtOGHnab5eM98NOXvZI4-GBTkPqR1PFDOTFrjGmNGh95G6AMJ

### SV STORE

Tác giả: SV STORE

Đội ngũ phát triển: Trang web thường được phát triển bởi một nhóm kỹ sư phần mềm, nhà thiết kế UX/UI, và các chuyên gia marketing. Họ có thể thuộc một công ty hoặc là nhóm độc lập.

là nhóm độc lập.

URL: https://svstore.com.vn/danh-muc/laptop/

**Công nghệ sử dụng:**

Ngôn ngữ lập trình: Có thể sử dụng HTML, CSS, JavaScript cho frontend. Backend có thể sử dụng PHP, Python, Ruby hoặc Node.js.

Cơ sở dữ liệu: Thường sử dụng MySQL, PostgreSQL hoặc MongoDB để lưu trữ thông tin sản phẩm và người dùng.

CMS: Nhiều trang web thương mại điện tử sử dụng hệ thống quản lý nội dung (CMS) như WordPress hoặc Shopify để dễ dàng quản lý nội dung và sản phẩm.

Ưu điểm:

Dễ tiếp cận: Người dùng có thể dễ dàng tìm kiếm và mua sản phẩm từ bất kỳ đâu.

Đa dạng sản phẩm: Cung cấp nhiều lựa chọn từ nhiều thương hiệu khác nhau.

Tiện lợi: Hỗ trợ đặt hàng và thanh toán trực tuyến, tiết kiệm thời gian.

Khuyến mãi: Thường có nhiều chương trình khuyến mãi và giảm giá hấp dẫn.

Nhược điểm:

Thiếu trải nghiệm trực tiếp: Người dùng không thể xem và cảm nhận sản phẩm trước khi mua.

Rủi ro khi thanh toán: Có thể gặp phải các vấn đề về bảo mật thông tin cá nhân và tài chính.

Vấn đề giao hàng: Đôi khi có thể gặp phải chậm trễ trong giao hàng hoặc sản phẩm không đúng như mô tả.

Dịch vụ người dùng: Nếu có vấn đề phát sinh, việc hỗ trợ từ xa có thể không hiệu quả như khi đến trực tiếp cửa hàng.

Kết luận:

SVStore là một trang web thương mại điện tử tại Việt Nam cung cấp nhiều sản phẩm công nghệ như laptop và linh kiện. Với những ưu điểm như sự tiện lợi, đa dạng sản phẩm và thường xuyên có khuyến mãi, trang web này là lựa chọn hấp dẫn cho người tiêu dùng. Tuy nhiên, cũng cần lưu ý những nhược điểm như thiếu trải nghiệm trực tiếp, rủi ro bảo mật, và vấn đề giao hàng.

- Phân tích các công nghệ có thể sử dụng để giải quyết vấn đề và giải thích vì sao em chọn chúng .

- Chốt lại những điểm hạn chế của những hệ thống đã tồn tại và đề xuất giải pháp mà đề tài em sẽ thực hiện.

- Nhớ trích dẫn tài liệu tham khảo cho các nội dung em sẽ trình bày trong phần tổng quan này.

## Công nghệ sử dụng

**Tổng quan về các công nghệ sẽ sử dụng:**

JavaScript là ngôn ngữ lập trình được nhà phát triển sử dụng để tạo trang web tương tác. Từ làm mới bảng 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.

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

Node.js là một nền tảng JavaScript phía máy chủ (server-side) cho phép chạy mã JavaScript ngoài trình duyệt, chủ yếu dùng để phát triển ứng dụng web, ứng dụng mạng, và các dịch vụ web. Node.js được xây dựng trên V8 JavaScript engine của Google Chrome, giúp mã JavaScript chạy nhanh và hiệu quả.

Express.js là một framework web phổ biến được sử dụng để phát triển các ứng dụng web và API qua Node.js. Nền tảng được xem là một phương thức xử lý các yêu cầu HTTP, quản lý các tuyến đường, xử lý phần mềm trung gian và nhiều tính năng khác để phát triển hiệu quả ứng dụng web.

CSS là 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).

MongoDB là một hệ quản trị cơ sở dữ liệu NoSQL mã nguồn mở, rất phổ biến hiện nay. Thay vì lưu trữ dữ liệu trong các bảng và dòng như các hệ quản trị SQL truyền thống, MongoDB lưu dữ liệu dưới dạng các tài liệu (documents) trong các bộ sưu tập (collections). Các dữ liệu được lưu trữ trong tài liệu kiểu JSON nên tốc độ truy vấn sẽ rất nhanh.

## Mục tiêu nghiên cứu

Mục tiêu là tìm hiểu và xây dựng một website bán laptop hiệu quả, đáp ứng nhu cầu của người tiêu dùng trong việc tìm kiếm và mua sắm các loại laptop với đa dạng tính năng, thương hiệu và mức giá.

## Phạm vi nghiên cứu

Sản phẩm: Website sẽ cung cấp một loạt các sản phẩm laptop từ các thương hiệu nổi tiếng, bao gồm các phân khúc giá khác nhau từ laptop giá rẻ, trung cấp đến laptop cao cấp.

Người dùng: Mục tiêu của nghiên cứu là phục vụ các người dùng có nhu cầu tìm kiếm và mua laptop trực tuyến.

Chức năng của website: Phát triển các tính năng của website như tìm kiếm sản phẩm, lọc theo các tiêu chí như giá, cấu hình, và thương hiệu, đăng ký tài khoản người dùng, đặt hàng, theo dõi đơn hàng, và dịch vụ người dùng. Website cũng có thêm một số chức năng quản lý sản phẩm, người dùng, danh mục sản phẩm, hãng sản xuất, đơn hàng giúp admin thuận tiện trong việc quản lý và vận hành website.

## Phương pháp nghiên cứu

### Phương pháp nghiên cứu lý thuyết

Tìm kiếm và tham khảo các tài liệu nghiên cứu, sách chuyên ngành, báo cáo ngành và các bài viết về phát triển website thương mại điện tử.

Tìm hiểu về các công cụ có liên quan và cần thiết để thực hiện xây dựng phần mềm.

### Phương pháp nghiên cứu thực nghiệm

Thiết kế giao diện người dùng: Sử dụng công cụ figma để phác thảo giao diện người dùng.

Phát triển phần mềm: Lựa chọn công nghệ để phù hợp, tiến hành lập trình và tích hợp các tính năng.

# NGHIÊN CỨU LÝ THUYẾT

## Node.js

Khái niệm



Hình 1 Node.js

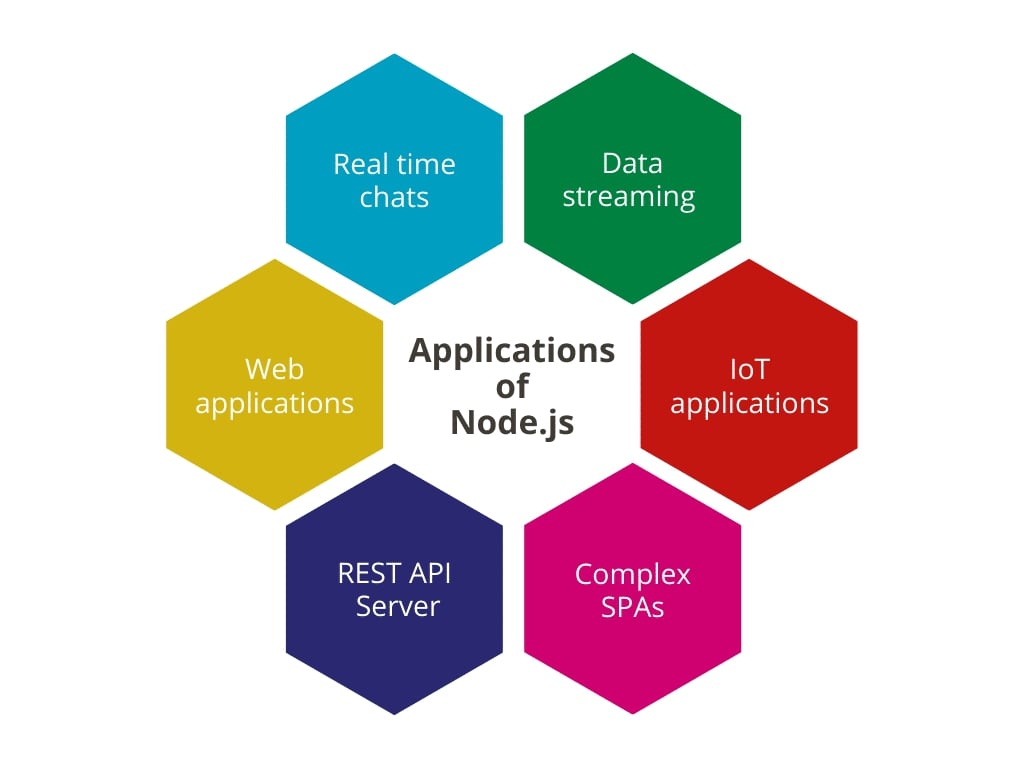
Node.js là một nền tảng (Platform) phát triển độc lập được xây dựng ở trên Javascript Runtime của Chrome mà chúng ta có thể xây dựng được các ứng dụng mạng một cách nhanh chóng và dễ dàng mở rộng.

Node.js được xây dựng và phát triển từ năm 2009, bảo trợ bởi công ty Joyent, trụ sở tại California, Hoa Kỳ.

Phần Core bên dưới của Node.js được viết hầu hết bằng C++ nên cho tốc độ xử lý và hiệu năng khá cao.

Node.js tạo ra được các ứng dụng có tốc độ xử lý nhanh, realtime thời gian thực.

Ứng dụng của Node.js



Hình 2 Ứng dụng của Node.js

Ứng dụng Web Thời Gian Thực (Real-time Web Applications): Node.js là lựa chọn lý tưởng cho các ứng dụng web thời gian thực như trò chuyện trực tuyến và trò chơi trực tuyến do khả năng xử lý các sự kiện I/O một cách nhanh chóng và hiệu quả.

APIs Server-side: Node.js thường được sử dụng để xây dựng RESTful APIs do khả năng xử lý đồng thời lớn và tốc độ phản hồi nhanh, làm cơ sở cho các ứng dụng di động và web.

Streaming Data: Node.js hỗ trợ xử lý dữ liệu dạng stream, cho phép ứng dụng xử lý các tệp video, âm thanh hoặc các dữ liệu khác trong khi chúng vẫn đang được truyền, thay vì phải chờ cho đến khi toàn bộ tệp được tải về.

Ứng dụng Một Trang (Single Page Applications): Node.js phù hợp với việc phát triển các ứng dụng một trang (SPA) như Gmail, Google Maps, hay Facebook, nơi mà nhiều tương tác xảy ra trên một trang duy nhất mà không cần tải lại trang.

Công cụ và Tự Động Hóa: Node.js cũng được sử dụng để phát triển các công cụ dòng lệnh và các script tự động hóa quy trình làm việc, nhờ vào các gói NPM hỗ trợ đa dạng và khả năng tích hợp dễ dàng với các công nghệ khác.

Microservices Architecture: Node.js là một lựa chọn phổ biến cho kiến trúc microservices, nơi các ứng dụng lớn được chia thành các dịch vụ nhỏ, độc lập và dễ quản lý hơn.

Ứng dụng IoT (VARCHARernet of Things): Node.js thường được sử dụng trong các ứng dụng IoT, nơi cần xử lý một lượng lớn các kết nối đồng thời và các sự kiện từ các thiết bị IoT.

Dashboard và Monitoring: Node.js được sử dụng để xây dựng các dashboard hiển thị dữ liệu thời gian thực và các công cụ giám sát, giúp doanh nghiệp dễ dàng theo dõi hiệu suất và tình trạng của các hệ thống.

Nguyên lý hoạt động của Node.js

Node.js là một nền tảng phát triển phía server được xây dựng trên nền tảng JavaScript. Khi một yêu cầu mạng đến từ một client, Node.js sẽ xử lý yêu cầu đó bằng cách thực hiện các bước như sau:

Node.js tạo một event loop để theo dõi các yêu cầu mạng đến và đi.

Khi một yêu cầu mạng đến, Node.js sẽ tạo một worker thread (luồng làm việc) để xử lý yêu cầu đó.

Trong worker thread, Node.js sẽ thực hiện các tác vụ xử lý yêu cầu, chẳng hạn như đọc và ghi vào cơ sở dữ liệu, đọc và ghi file, tương tác với API, ...

Khi worker thread đã hoàn thành các tác vụ, Node.js sẽ trả về kết quả cho client qua mạng.

Nếu có yêu cầu mạng mới đến, Node.js sẽ tạo một worker thread mới để xử lý yêu cầu đó.

Các yêu cầu mạng đến và đi trong Node.js được xử lý bằng cách sử dụng các hàm callback, Promise, async/await để đảm bảo tính phi đồng bộ và tăng hiệu suất của ứng dụng. Các yêu cầu mạng được xử lý một cách độc lập, giúp tránh tình trạng "blocking" (chặn) trong quá trình xử lý yêu cầu. Node.js cũng có thể hoạt động với các module và thư viện khác để hỗ trợ cho việc phát triển ứng dụng web.

## MongoDB



Hình 3 MongoDB

MongoDB là một hệ quản trị cơ sở dữ liệu mã nguồn mở, là CSDL thuộc NoSql và được hàng triệu người sử dụng.

MongoDB là một database hướng tài liệu (document), các dữ liệu được lưu trữ trong document kiểu JSON thay vì dạng bảng như CSDL quan hệ nên truy vấn sẽ rất nhanh.

Với CSDL quan hệ chúng ta có khái niệm bảng, các cơ sở dữ liệu quan hệ (như MySQL hay SQL Server...) sử dụng các bảng để lưu dữ liệu thì với MongoDB chúng ta sẽ dùng khái niệm là collection thay vì bảng.

Các collection trong MongoDB được cấu trúc rất linh hoạt, cho phép các dữ liệu lưu trữ không cần tuân theo một cấu trúc nhất định.

Thông tin liên quan được lưu trữ cùng nhau để truy cập truy vấn nhanh thông qua ngôn ngữ truy vấn MongoDB.

Ưu điểm của MongoDB

Do MongoDB sử dụng lưu trữ dữ liệu dưới dạng Document JSON nên mỗi một collection sẽ có các kích cỡ và các document khác nhau, linh hoạt trong việc lưu trữ dữ liệu, nên người dùng muốn gì thì cứ insert vào thoải mái.

Dữ liệu trong MongoDB không có sự ràng buộc lẫn nhau, không có join như trong RDBMS nên khi insert, xóa hay update nó không cần phải mất thời gian kiểm tra xem có thỏa mãn các ràng buộc dữ liệu như trong RDBMS.

MongoDB rất dễ mở rộng (Horizontal Scalability). Trong MongoDB có một khái niệm cluster là cụm các node chứa dữ liệu giao tiếp với nhau, khi muốn mở rộng hệ thống ta chỉ cần thêm một node với vào cluster:

Trường dữ liệu “\_id” luôn được tự động đánh index (chỉ mục) để tốc độ truy vấn thông tin đạt hiệu suất cao nhất.

Khi có một truy vấn dữ liệu, bản ghi được cached lên bộ nhớ Ram, để phục vụ lượt truy vấn sau diễn ra nhanh hơn mà không cần phải đọc từ ổ cứng.

Hiệu năng cao: Tốc độ truy vấn (find, update, insert, delete) của MongoDB nhanh hơn hẳn so với các hệ quản trị cơ sở dữ liệu quan hệ (RDBMS). Với một lượng dữ liệu đủ lớn thì thử nghiệm cho thấy tốc độ insert của MongoDB có thể nhanh tới gấp 100 lần so với MySQL.

Nhược điểm của MongoDB

MongoDB không có các tính chất ràng buộc như trong RDBMS nên khi thao tác với mongoDB thì phải hết sức cẩn thận.

Tốn bộ nhớ do dữ liệu lưu dưới dạng key-value, các collection chỉ khác về value do đó key sẽ bị lặp lại. Không hỗ trợ join nên dễ bị dữ thừa dữ liệu.

Khi insert/update/remove bản ghi, MongoDB sẽ chưa cập nhật ngay xuống ổ cứng, mà sau 60 giây MongoDB mới thực hiện ghi toàn bộ dữ liệu thay đổi từ RAM xuống ổ cứng điều này sẽ là nhược điểm vì sẽ có nguy cơ bị mất dữ liệu khi xảy ra các tình huống như mất điện...

## Express JS

### Khái niệm



Hình 2.2. Express.js

Express.js là một framework web phổ biến được sử dụng để phát triển các ứng dụng web và API qua Node.js. Nền tảng được xem là một phương thức xử lý các yêu cầu HTTP, quản lý các tuyến đường, xử lý phần mềm trung gian và nhiều tính năng khác để phát triển hiệu quả ứng dụng web.

Express.js tập trung vào công việc tối ưu hóa việc xây dựng web ứng dụng bằng cách cung cấp một cấu trúc hoạt động và chỉ định rõ ràng việc xử lý yêu cầu và phản hồi. Nền tảng cũng hỗ trợ tích hợp các phần mềm trung gian bên ngoài để mở rộng chức năng của ứng dụng.

### Đặc điểm của Express.js

Định tuyến (Routing): Express.js cho phép định tuyến các yêu cầu HTTP tới các endpoVARCHAR cụ thể dựa trên đường dẫn và phương thức yêu cầu. Điều này giúp tách biệt logic xử lý cho từng phần của ứng dụng và quản lý các route một cách linh hoạt.

Trích xuất dữ liệu từ yêu cầu (Request parsing): Express.js cung cấp các công cụ để trích xuất dữ liệu từ yêu cầu HTTP như tham số đường dẫn, tham số truy vấn, thân yêu cầu, header và các loại dữ liệu khác.

Phản hồi yêu cầu (Response handling): Express.js cho phép phản hồi yêu cầu với headers và status code tương ứng, cũng như trả về dữ liệu dưới nhiều định dạng như JSON, HTML hoặc phản hồi khác.

Middleware: Trung gian (middleware) trong Express.js có thể được sử dụng để xử lý các yêu cầu trước khi chúng đến các endpoVARCHAR cụ thể, giúp thực hiện các thao tác như xác thực người dùng, xử lý lỗi, nén dữ liệu và nhiều tác vụ khác

RESTful API Development: Express.js thường được sử dụng để xây dựng các RESTful API với sự hỗ trợ tốt cho việc tạo, lấy, cập nhật và xóa dữ liệu thông qua các phương thức HTTP.

### Ưu điểm của Express.js

Dễ sử dụng: Với cú pháp đơn giản và dễ hiểu, Express.js giúp việc phát triển ứng dụng trở nên nhanh chóng và hiệu quả, giảm thiểu độ phức tạp trong việc quản lý yêu cầu và phản hồi.

Hiệu suất cao: Vì được xây dựng trên Node.js, Express.js tận dụng các đặc tính của Node.js như xử lý bất đồng bộ và hiệu suất cao, giúp ứng dụng web hoặc API chạy nhanh.

Phát triển nhanh chóng: Express.js cung cấp nhiều công cụ và tính năng hỗ trợ việc phát triển web nhanh chóng, như hỗ trợ các template engine, middleware và khả năng xử lý lỗi dễ dàng.

Cộng đồng lớn: Express.js có một cộng đồng phát triển rộng lớn và rất nhiều tài liệu hỗ trợ, giúp người dùng dễ dàng tìm kiếm giải pháp và nhận sự trợ giúp khi gặp vấn đề.

Mở rộng linh hoạt: Express.js cho phép dễ dàng mở rộng ứng dụng bằng cách sử dụng các module hoặc middleware từ bên thứ ba, tăng tính linh hoạt của ứng dụng.

## RESTful API

### Khái niệm



Hình 4 REST API

RESTful API là một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web (thiết kế Web services) để tiện cho việc quản lý các resource. Nó chú trọng vào tài nguyên hệ thống (tệp văn bản, ảnh, âm thanh, video, hoặc dữ liệu động…), bao gồm các trạng thái tài nguyên được định dạng và được truyền tải qua HTTP.

API (Application Programming VARCHARerface) là một tập các quy tắc và cơ chế mà theo đó, một ứng dụng hay một thành phần sẽ tương tác với một ứng dụng hay thành phần khác. API có thể trả về dữ liệu mà bạn cần cho ứng dụng của mình ở những kiểu dữ liệu phổ biến như [JSON](https://topdev.vn/blog/json-la-gi/) hay XML.

REST (REpresentational State Transfer) là một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API. Nó sử dụng phương thức 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.

Chức năng quan trọng nhất của REST là quy định cách sử dụng các HTTP method (như GET, POST, PUT, DELETE…) và cách định dạng các URL cho ứng dụng web để quản các resource. RESTful không quy định logic code ứng dụng và không giới hạn bởi ngôn ngữ lập trình ứng dụng, bất kỳ ngôn ngữ hoặc framework nào cũng có thể sử dụng để thiết kế một RESTful API.

RESTful hoạt động như thế nào



Hình 5 Cách hoạt động của RESTful API

REST hoạt động chủ yếu dựa vào giao thức HTTP. Các hoạt động cơ bản nêu trên sẽ sử dụng những phương thức HTTP riêng.

GET (SELECT): Trả về một Resource hoặc một danh sách Resource.

POST (CREATE): Tạo mới một Resource.

PUT (UPDATE): Cập nhật thông tin cho Resource.

DELETE (DELETE): Xoá một Resource.

Những phương thức hay hoạt động này thường được gọi là CRUD tương ứng với Create, Read, Update, Delete – Tạo, Đọc, Sửa, Xóa.

API RESTful mang lại những lợi ích

API RESTful có những lợi ích sau:

1. **Khả năng thay đổi quy mô**

Các hệ thống triển khai API REST có thể thay đổi quy mô một cách hiệu quả vì REST tối ưu hóa các tương tác giữa client và máy chủ. Tình trạng phi trạng thái loại bỏ tải của máy chủ vì máy chủ không phải giữ lại thông tin yêu cầu của client trong quá khứ. Việc lưu bộ nhớ đệm được quản lý tốt sẽ loại bỏ một phần hoặc hoàn toàn một số tương tác giữa client và máy chủ. Tất cả các tính năng này hỗ trợ khả năng thay đổi quy mô mà không gây ra tắc nghẽn giao tiếp làm giảm hiệu suất.

1. **Sự linh hoạt**

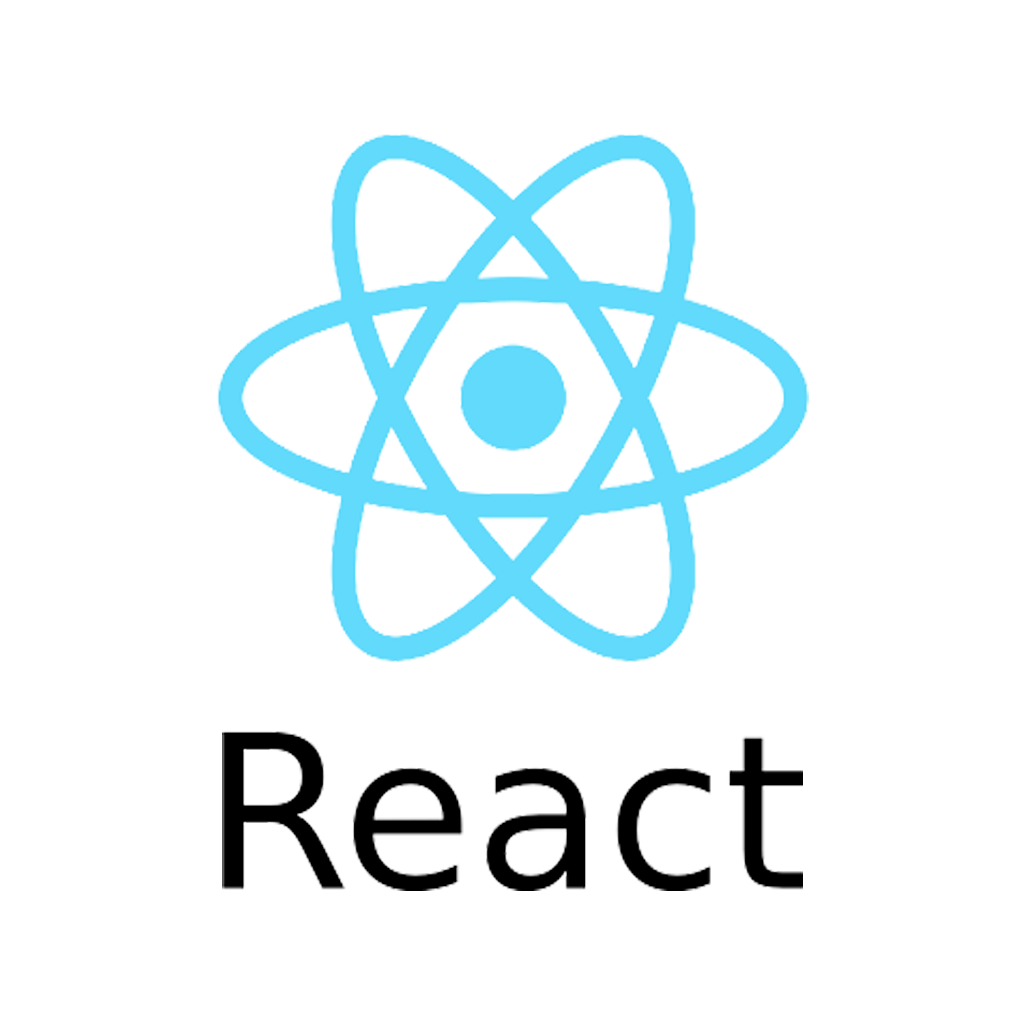
Các dịch vụ web RESTful hỗ trợ phân tách hoàn toàn giữa client và máy chủ. Các dịch vụ này đơn giản hóa và tách riêng các thành phần máy chủ khác nhau để mỗi phần có thể phát triển độc lập. Các thay đổi ở nền tảng hoặc công nghệ tại ứng dụng máy chủ không ảnh hưởng đến ứng dụng client. Khả năng phân lớp các chức năng ứng dụng làm tăng tính linh hoạt hơn nữa. Ví dụ: các nhà phát triển có thể thực hiện các thay đổi đối với lớp cơ sở dữ liệu mà không cần viết lại logic ứng dụng.

1. **Sự độc lập**

Các API REST không phụ thuộc vào công nghệ được sử dụng. Bạn có thể viết cả ứng dụng client và máy chủ bằng nhiều ngôn ngữ lập trình khác nhau mà không ảnh hưởng đến thiết kế API. Bạn cũng có thể thay đổi công nghệ cơ sở ở hai phía mà không ảnh hưởng đến giao tiếp.

## React

### Khái niệm



Hình 6 React

React (ReactJS) là một thư viện [JavaScript](https://200lab.io/blog/javascript-la-gi/) mã nguồn mở, được dùng để xây dựng giao diện người dùng (frontend) cho web. React chỉ tập trung vào phần hiển thị giao diện (view), chứ không can thiệp vào cách sắp xếp logic nghiệp vụ hoặc cấu trúc ứng dụng.

Những đặc điểm nổi bật của React

1. **Linh hoạt trong thiết kế kiến trúc**

React tập trung vào việc hiển thị giao diện người dùng và cho phép lập trình viên tự do quyết định cách sắp xếp logic nghiệp vụ. Chính sự linh hoạt này làm cho React trở nên phổ biến, vì nó phù hợp với nhiều loại dự án và phong cách phát triển khác nhau.

Khác với các framework có kiến trúc cố định như Angular, React không ép buộc người dùng vào một mô hình cụ thể, khiến nó linh hoạt cho nhiều dự án khác nhau. Mặc dù điều này trong React đem lại lợi ích cho lập trình viên có kinh nghiệm, nhưng lại gây khó khăn cho người mới bắt đầu vì thiếu sự hướng dẫn cụ thể.

1. **Kiến trúc Component đơn giản và nhẹ**

React được xây dựng dựa trên kiến trúc **component**, nơi mỗi thành phần có thể được tái sử dụng, giúp ứng dụng dễ mở rộng và duy trì. Các component trong React rất nhẹ và có thể chỉ là các hàm đơn giản trả về JSX.

Điều này giúp các lập trình viên dễ dàng phát triển các ứng dụng lớn hơn bằng cách chia nhỏ thành các thành phần độc lập, có thể tái sử dụng mà không cần phải lo về việc tích hợp phức tạp.

1. **Cộng đồng hỗ trợ lớn**

Một trong những lý do chính khiến React trở nên phổ biến là nhờ vào số lượng người dùng lớn và sự hỗ trợ từ cộng đồng cũng như các công cụ học tập. Đây là yếu tố quan trọng giúp các lập trình viên, đặc biệt là người mới bắt đầu, dễ dàng học và sử dụng React một cách hiệu quả.

Các diễn đàn lớn như Stack Overflow, **Reddit**, **Dev.to**, và **Github** chứa đầy các câu hỏi, câu trả lời và bài viết liên quan đến React, giúp người học dễ dàng tìm thấy sự trợ giúp khi gặp khó khăn.

React được duy trì và phát triển bởi **Facebook** (Meta), một trong những công ty công nghệ lớn nhất thế giới. Điều này đảm bảo rằng React sẽ nhận được sự hỗ trợ liên tục và các bản cập nhật mới, giúp lập trình viên yên tâm rằng React sẽ tiếp tục phát triển và phù hợp với các dự án dài hạn.

Các thành phần quan trọng trong ReactJS

JSX (JavaScript XML) là một cú pháp mở rộng cho phép lập trình viên viết mã giống như [HTML](https://200lab.io/blog/html-la-gi/) trong JavaScript. Trong các ngôn ngữ khác, lập trình viên thường phải viết code HTML và JavaScript riêng rẽ. Tuy nhiên, với JSX, React cho phép lập trình viên kết hợp cả hai trong cùng một mã nguồn, giúp quản lý dễ dàng hơn, đặc biệt là trong các ứng dụng phức tạp.

**Virtual DOM** (Document Object Model ảo) là một bản sao nhẹ hơn của DOMthật. DOM thật là cấu trúc cây chứa tất cả các thành phần HTML trong trang web. Khi người dùng tương tác với ứng dụng (ví dụ: nhập văn bản, nhấn nút), ứng dụng sẽ thay đổi nội dung và DOM thật phải được cập nhật.

Tuy nhiên, việc cập nhật trực tiếp DOM thật có thể gây ra tình trạng chậm chạp và kém hiệu quả, đặc biệt trong các ứng dụng lớn với nhiều phần tử. Để giải quyết vấn đề này, React sử dụng Virtual DOM, với mục đích chỉ những phần có sự thay đổi mới được cập nhật lên DOM thật, giúp tiết kiệm thời gian và tài nguyên.

Những lợi ích tuyệt vời mà ReactJS mang lại cho lập trình viên

* Hiệu suất cao: ReactJS sử dụng Virtual DOM để tối ưu hóa hiệu suất của ứng dụng. Virtual DOM cho phép ReactJS cập nhật các thay đổi trên trang web một cách nhanh chóng và hiệu quả hơn so với cách truyền thống, giúp tăng tốc độ và hiệu suất của ứng dụng.
* Tái sử dụng: ReactJS cho phép tái sử dụng các thành phần UI, giúp giảm thiểu thời gian và chi phí phát triển. Các thành phần UI có thể được sử dụng lại trong nhiều phần khác nhau của ứng dụng, giúp tăng tính linh hoạt và khả năng mở rộng của ứng dụng.
* Dễ dàng quản lý trạng thái: ReactJS giúp quản lý trạng thái của ứng dụng một cách dễ dàng. Sử dụng State và Props, ReactJS cho phép các nhà phát triển quản lý trạng thái của các thành phần UI một cách chính xác và dễ dàng.
* Hỗ trợ tốt cho SEO: ReactJS cho phép các nhà phát triển xây dựng ứng dụng web với khả năng tương thích tốt với SEO. Với sự hỗ trợ của các thư viện như React Helmet, ReactJS cho phép các nhà phát triển tùy chỉnh và quản lý các phần tử meta và title cho từng trang web.
* Hỗ trợ đa nền tảng: ReactJS không chỉ được sử dụng để phát triển các ứng dụng web, mà còn được sử dụng để phát triển các ứng dụng di động với React Native. Sử dụng React Native, các nhà phát triển có thể xây dựng ứng dụng di động cho cả iOS và Android sử dụng cùng một mã nguồn.

Các tính năng nổi bật của ReactJS

* **Components**: ReactJS cho phép phát triển ứng dụng web theo mô hình component. Các component là các phần tử UI độc lập có thể được tái sử dụng trong nhiều phần khác nhau của ứng dụng.
* **Virtual DOM**: ReactJS sử dụng Virtual DOM để tối ưu hóa hiệu suất của ứng dụng. Virtual DOM là một bản sao của DOM được lưu trữ trong bộ nhớ và được cập nhật một cách nhanh chóng khi có thay đổi, giúp tăng tốc độ và hiệu suất của ứng dụng.
* **JSX**: JSX là một ngôn ngữ lập trình phân biệt được sử dụng trong ReactJS để mô tả các thành phần UI. JSX kết hợp HTML và JavaScript, giúp cho việc viết mã dễ hiểu và dễ bảo trì hơn.
* **State và Props**: ReactJS cho phép quản lý trạng thái của các thành phần UI thông qua State và Props. State là trạng thái của một thành phần được quản lý bởi nó chính, trong khi Props là các giá trị được truyền vào từ bên ngoài để tùy chỉnh hoặc điều khiển hành vi của một thành phần.
* **Hỗ trợ tốt cho SEO**: ReactJS hỗ trợ tốt cho việc tối ưu hóa SEO. Với các thư viện như React Helmet, các nhà phát triển có thể quản lý các phần tử meta và title cho từng trang web, giúp tăng khả năng tìm kiếm và tăng cường trải nghiệm người dùng.
* **Hỗ trợ đa nền tảng**: ReactJS không chỉ được sử dụng để phát triển ứng dụng web, mà còn được sử dụng để phát triển ứng dụng di động với React Native. Sử dụng React Native, các nhà phát triển có thể xây dựng ứng dụng di động cho cả iOS và Android sử dụng cùng một mã nguồn.
* **Redux**: Redux là một thư viện quản lý trạng thái cho các ứng dụng ReactJS. Nó giúp quản lý trạng thái của ứng dụng một cách chính xác và dễ dàng, đồng thời giúp tăng tính linh hoạt và khả năng mở rộng của ứng dụng.

Cách sử dụng ReactJS trong phát triển web

**Bước 1 Cài đặt Node.js và npm**

ReactJS được xây dựng trên nền tảng Node.js, do đó lập trình viên cần cài đặt Node.js và npm để phát triển ứng dụng ReactJS.

**Bước 2 Tạo một ứng dụng React**

lập trình viên có thể tạo một ứng dụng React bằng cách sử dụng lệnh "create-react-app" trong Command Prompt hoặc Terminal.

**Bước 3 Tạo các component**

Tạo các component để xây dựng giao diện người dùng cho ứng dụng. Lập trình viên có thể tạo component bằng cách sử dụng class hoặc hàm.

**Bước 4 Xây dựng giao diện người dùng**

Sử dụng JSX để xây dựng giao diện người dùng cho ứng dụng. JSX là một ngôn ngữ phân biệt được sử dụng trong ReactJS để mô tả các thành phần UI.

**Bước 5 Quản lý trạng thái**

Sử dụng State và Props để quản lý trạng thái của các thành phần UI. State là trạng thái của một thành phần được quản lý bởi nó chính, trong khi Props là các giá trị được truyền vào từ bên ngoài để tùy chỉnh hoặc điều khiển hành vi của một thành phần.

**Bước 6 Kết nối với API**

Sử dụng thư viện như Axios để kết nối với API và lấy dữ liệu từ server.

**Bước 7 Build và triển khai ứng dụng**

Sử dụng lệnh "npm run build" để build ứng dụng và triển khai nó trên môi trường sản phẩm.

## Tương tác với cơ sở dữ liệu

### Kết nối cơ sở dữ liệu

Để kết nối với cơ sở dữ liệu mongoDB cần phải cài đặt thư viện mongoose, sử dụng lệnh “npm install mongoose” để thực hiện việc cài đặt.

Sau khi cài đặt Mongoose xong cần phải thiết lập kết nối với cơ sở dữ liệu MongoDB . Ví dụ dưới đây là cách kết nối cơ bản trong Node.js:

**Khai báo thư viện Mongoose**

const mongoose = require('mongoose');

**Thiết lập kết nối tới cơ sở dữ liệu MongoDB**

mongoose.connect('mongodb://localhost:27017/tenCoSoDuLieu', { useNewUrlParser: true, useUnifiedTopology: true })

.then(() => {

console.log('Kết nối cơ sở dữ liệu thành công!');

})

.catch(err => {

console.log('Kết nối cơ sở dữ liệu thất bại:', err);

});

**mongodb://localhost:27017/** **tenCoSoDuLieu** là chuỗi kết nối tới cơ sở dữ liệu MongoDB, tên của cơ sở dữ liệu ở đây là **tenCoSoDuLieu**.

### Chọn và hiển thị dữ liệu

Sau khi đã kết nối thành công với cơ sở dữ liệu để có thể thực hiện các truy vấn thì cần thực hiện các phương thức của Mongoose như find(), findOne(),...

Ví dụ về việc truy vấn dữ liệu:

**Tạo một mô hình User từ một schema định nghĩa trước**

const User = mongoose.model('User', new mongoose.Schema({

name: String, // tên của người dùng

age: Number // tuổi của người dùng

}));

**Truy vấn tất cả người dùng trong collection "users" bằng lệnh find()**

User.find()

.then(users => {

console.log('Danh sách người dùng:', users);

})

.catch(err => {

console.log('Lỗi khi truy vấn dữ liệu:', err);

});

### Thêm, xóa, sửa dữ liệu

Để thêm dữ liệu vào cơ sở dữ liệu có thể sử dụng phương thức save() hoặc phương thức create() của Mongoose.

Ví dụ về sử dụng phương thức save() để thêm dữ liệu:

**Thêm một người dùng mới vào collection "users"**

const newUser = new User({

name: 'Nguyen Thi B',

age: 30

});

newUser.save()

.then(() => {

console.log('Thêm người dùng thành công!');

})

.catch(err => {

console.log('Lỗi khi thêm người dùng:', err);

});

Ví dụ về sử dụng phương thức create() để thêm dữ liệu:

User.create([

{ name: 'Nguyen Thi C', age: 25 },

{ name: 'Le Thi D', age: 28 }

])

.then(() => {

console.log('Thêm nhiều người dùng thành công!');

})

.catch(err => {

console.log('Lỗi khi thêm dữ liệu:', err);

});

Để sửa dữ liệu trong MongoDB có thể sử dụng phương thức updateOne() hoặc

findByIdAndUpdate().

Ví dụ:

**Sửa thông tin người dùng có id cụ thể**

User.findByIdAndUpdate('id-user', { age: 35 })

.then(() => {

console.log('Cập nhật người dùng thành công!');

})

.catch(err => {

console.log('Lỗi khi cập nhật:', err);

});

Để xóa dữ liệu có thể sử dụng phương thức findByIdAndDelete() hoặc deleteOne().

Ví dụ:

**Xóa người dùng theo id**

User.findByIdAndDelete('id-user')

.then(() => {

console.log('Xóa người dùng thành công!');

})

.catch(err => {

console.log('Lỗi khi xóa dữ liệu:', err);

});

# HIỆN THỰC HÓA NGHIÊN CỨU

## Mô tả bài toán

Website bán laptop là một nền tảng thương mại điện tử cung cấp các tính năng tiện lợi cho người dùng và quản trị viên. Người dùng có thể tìm kiếm và xem thông tin chi tiết về sản phẩm, lọc laptop theo giá, hãng, và cấu hình. Người dùng có thể thêm sản phẩm vào giỏ hàng, quản lý số lượng, xem tổng tiền và thực hiện đặt hàng. Hệ thống hỗ trợ đăng ký, đăng nhập và cập nhật thông tin cá nhân. Người quản trị admin có thể quản lý người dùng, sản phẩm, danh mục, hãng sản xuất, và đơn hàng, bao gồm thêm, sửa, xóa thông tin, cùng việc cập nhật trạng thái đơn hàng. Trang chủ hiển thị danh sách laptop, nổi bật với các sản phẩm bán chạy hoặc khuyến mãi, mang đến trải nghiệm mua sắm dễ dàng và hiệu quả.

## Phân tích đặc tả hệ thống

### Đặc tả yêu cầu chức năng

**Chức năng của người dùng:**

Đăng ký tài khoản: Người dùng cần nhập thông tin (tên, email, mật khẩu), hệ thống sẽ lưu thông tin và xác thực người dùng đã đăng ký thành công.

Đăng nhập và quản lý thông tin cá nhân: Đăng nhập bằng email và mật khẩu, sau khi đăng nhập người dùng có thể cập nhật thông tin cá nhân như tên, địa chỉ, số điện thoại.

Tìm kiếm và lọc sản phẩm: Tìm kiếm sản phẩm qua tên hoặc từ khóa, người dùng cũng có thể lọc sản phẩm theo giá, hãng, cấu hình (RAM, CPU, dung lượng ổ cứng).

Xem chi tiết sản phẩm: Hiển thị thông tin chi tiết của sản phẩm.

Quản lý giỏ hàng: Người dùng có thể thêm sản phẩm vào giỏ hàng, xóa hoặc cập nhật số lượng sản phẩm trong giỏ hàng. Tiền trong giỏ hàng sẽ tự động tính để người dùng biết tổng giá trị đơn hàng.

Đặt hàng: Người dùng cung cấp thông tin giao hàng và xác nhận đơn hàng sau khi đặt hàng xong có thể theo dõi trạng thái đơn hàng.

**Chức năng của quản trị viên:**

Quản lý người dùng: Xem danh sách người dùng và thông tin chi tiết. Xóa tài khoản người dùng sai phạm.

Quản lý sản phẩm: Thêm mới, chỉnh sửa, hoặc xóa sản phẩm.

Quản lý danh mục sản phẩm: Thêm, sửa, hoặc xóa danh mục.

Quản lý hãng sản xuất: Thêm, sửa, hoặc xóa hãng sản xuất.

Quản lý đơn hàng: Xem danh sách đơn hàng, chi tiết đơn hàng và có thể cập nhật trạng thái đơn hàng (chờ xử lý, đang giao, hoàn tất, hủy).

### Đặc tả yêu cầu phi chức năng

Hiệu năng của hệ thống: Tốc độ tải trang phải nhanh và có khả năng chịu tải ở mức khá.

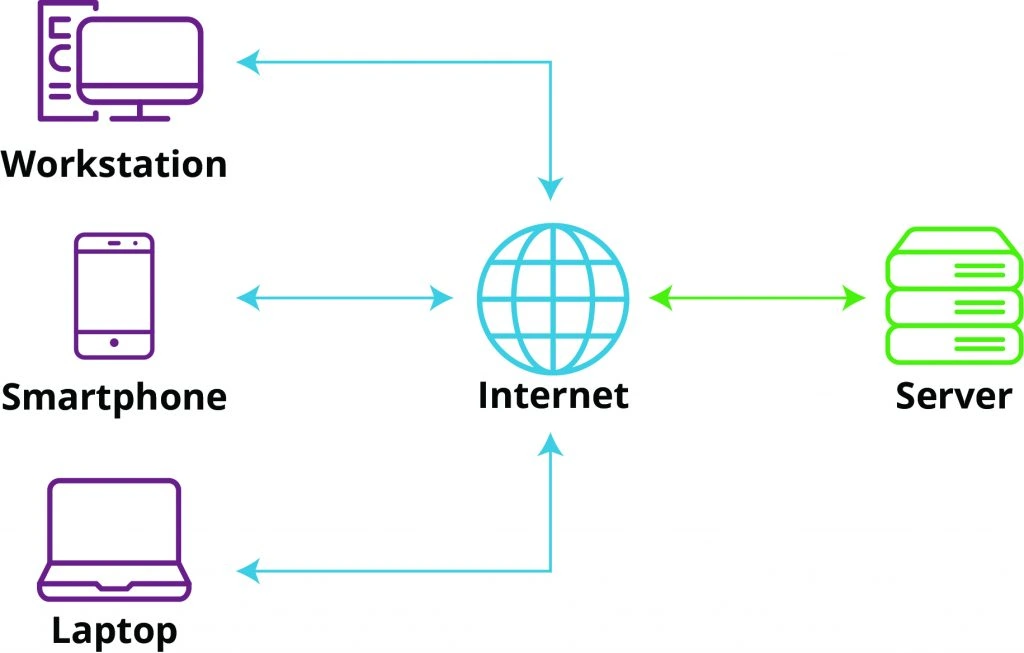
Khả năng mở rộng: Cấu trúc hệ thống linh hoạt để có thể tích hợp thêm tính năng trong cho hệ thống trong tương lai.

Khả năng tương thích: Có khả năng hỗ trợ các trình duyệt phổ biến như Chrome, Edge, Fire Fox, Cốc Cốc,...

Tính bảo mật: Mật khẩu của người dùng sẽ được mã hóa bằng thuật toán mạnh như bcrypt.

## Thiết kế hệ thống

### Kiến trúc hệ thống



Hệ thống áp dụng kiến trúc Client-Server.

Frontend (Client): Sử dụng ReactJS (hoặc HTML, CSS, JavaScript). Giao tiếp với backend thông qua RESTful API.

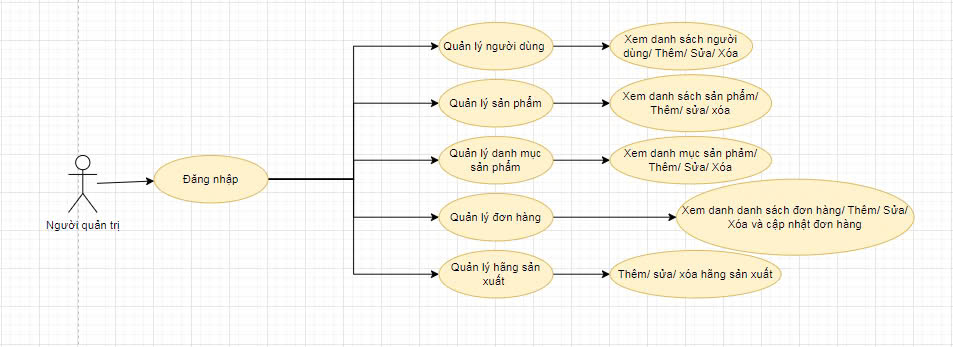
Backend (Server):Sử dụng Node.js với Express để xử lý logic và kết nối cơ sở dữ liệu. Xây dựng các API cho các chức năng của người dùng và admin.

Cơ sở dữ liệu (Database): Sử dụng MongoDB làm cơ sử dữ liệu để lưu trữ các thông tin của sản phẩm, người dùng,...

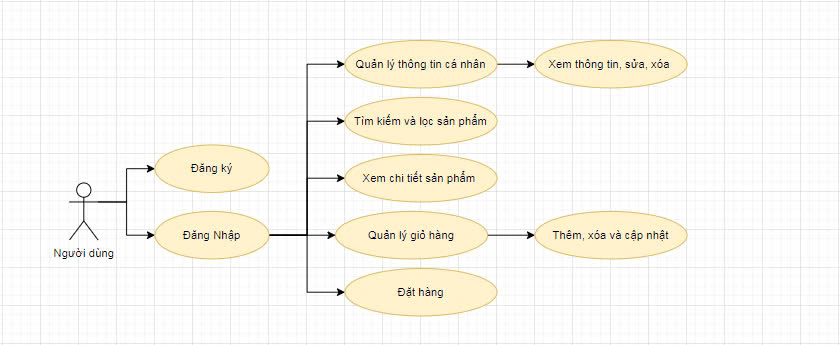
### Thiết kế dữ liệu

#### Sơ đồ Use case

Sơ đồ Use case của người quản trị



Sơ đồ Use case của người dùng



#### Danh sách các thực thể và mối kết hợp

Collection danh mục sản phẩm

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Kiểu dữ liệu | Mô tả |
| ma\_danh\_muc | VARCHAR | Mã danh mục |
| ma\_san\_pham | VARCHAR | Mã sản phẩm liên kết |
| ten\_danh\_muc | VARCHAR | Tên danh mục |

Collection sản phẩm

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Kiểu dữ liệu | Mô tả |
| ma\_san\_pham | VARCHAR | Mã sản phẩm (Khóa chính) |
| ten\_san\_pham | VARCHAR | Tên sản phẩm |
| ma\_danh\_muc | VARCHAR | Mã danh mục (Khóa ngoại tham chiếu đến bảng danh mục) |
| ma\_hang\_sx | VARCHAR | Mã hãng sản xuất (Khóa ngoại tham chiếu đến bảng hãng sản xuất) |
| gia | DOUBLE | Giá sản phẩm |
| so\_luong\_ton | VARCHAR | Số lượng tồn kho |
| mo\_ta | VARCHAR | Mô tả sản phẩm |
| anh\_dai\_dien | VARCHAR | Đường dẫn hình ảnh sản phẩm |
| danh\_sach\_hinh\_anh | VARCHAR | Danh sách đường dẫn hình ảnh của sản phẩm |

Collection chi tiết đơn hàng

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Kiểu dữ liệu | Mô tả |
| ma\_chi\_tiet | VARCHAR | Mã chi tiết đơn hàng (Khóa chính) |
| ma\_don\_hang | VARCHAR | Mã đơn hàng (Khóa ngoại liên kết đến bảng đơn hàng) |
| ma\_san\_pham | VARCHAR | Mã sản phẩm (Khóa ngoại liên kết đến bảng sản phẩm) |
| so\_luong | VARCHAR | Số lượng sản phẩm trong đơn hàng |

Collection đơn hàng

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Kiểu dữ liệu | Mô tả |
| ma\_don\_hang | VARCHAR | Mã đơn hàng (Khóa chính) |
| ma\_chi\_tiet | VARCHAR | Mã chi tiết đơn hàng (Khóa ngoại liên kết đến bảng chi tiết đơn hàng) |
| ma\_nguoi\_dung | VARCHAR | Mã khách hàng (Khóa ngoại liên kết đến bảng người dùng) |
| ngay\_dat | DATE | Ngày đặt hàng |
| trang\_thai | VARCHAR | Trạng thái đơn hàng (Đang xử lý, đã giao, đã hủy) |
| tong\_tien | DECIMAL | Tổng tiền của đơn hàng |

Collection người dùng

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Kiểu dữ liệu | Mô tả |
| ma\_nguoi\_dung | VARCHAR | Mã người dùng (Khóa chính) |
| ma\_don\_hang | VARCHAR | Mã đơn hàng (Khóa ngoại liên kết đến bảng DonHang) |
| ho\_ten | VARCHAR | Họ và tên người dùng |
| email | VARCHAR | Địa chỉ email của người dùng |
| mat\_khau | VARCHAR | Mật khẩu người dùng |
| so\_dien\_thoai | VARCHAR | Số điện thoại của người dùng |
| dia\_chi | VARCHAR | Địa chỉ người dùng |
| vai\_tro | VARCHAR | Vai trò của người dùng ( admin, customer) |

Collection hãng sản xuất

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Kiểu dữ liệu | Mô tả |
| ma\_hang\_sx | VARCHAR | Mã hãng sản xuất |
| Ten\_hang\_sx | VARCHAR | Tên hãng sản xuất |

### Thiết kế xử lý

#### Mô hình DFD mức ngữ cảnh

#### Mô hình DFD cấp 1

#### Mô hình DFD cấp 2

### Thiết kế giao diện

#### Sơ đồ website

#### Giao diện trang …

#### Giao diện …

# KẾT QUẢ NGHIÊN CỨU

## Bộ dữ liệu thử nghiệm

## Kết quả thực nghiệm

### Chức năng …

### Chức năng …

### Chức năng …

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

## Hướng phát triển

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | T. Dev, "Xampp là gì? Cách sử dụng và cài đặt XAMPP," 4 11 2024. [Online]. Available: https://topdev.vn/blog/cai-dat-xampp/. [Accessed 4 11 2024]. |
| [2] | T. Dev, "MySQL là gì? Giải đáp chi tiết và hướng dẫn cài đặt MySQL," 4 11 2024. [Online]. Available: https://topdev.vn/blog/gioi-thieu-ve-mysql/. [Accessed 4 11 2024]. |
| [3] | T. Dev, "HTML là gì?," 4 11 2024. [Online]. Available: https://topdev.vn/blog/html-la-gi/. [Accessed 4 11 2024]. |
| [4] | T. Dev, "CSS là gì?," 4 11 2024. [Online]. Available: https://topdev.vn/blog/css-la-gi/. [Accessed 4 11 2024].  5[Laravel là gì? Tổng quan về Laravel A-Z cho người mới bắt đầu - ITviec Blog](https://itviec.com/blog/laravel-la-gi/) |

6 [Làm việc với cơ sở dữ liệu trong Laravel | TopDev](https://topdev.vn/blog/lam-viec-voi-co-so-du-lieu-trong-laravel/#thiet-lap)

7 [Laravel là gì? Những ứng dụng của Laravel phổ biến nhất](https://topdev.vn/blog/laravel-la-gi-nhung-ung-dung-cua-laravel-pho-bien-nhat/#nhung-tinh-nang-chinh-cua-laravel-framework-la-gi)

8 [Laravel - Kết nối với Cơ sở dữ liệu](https://hiepsiit.com/public/detail/laravel/laravel/ket-noi-voi-co-so-du-lieu)

**9** [**Laravel - Model**](https://hiepsiit.com/public/detail/laravel/laravel/model-1)

**10** [**Laravel - Query Builder**](https://hiepsiit.com/public/detail/laravel/laravel/query-builder)