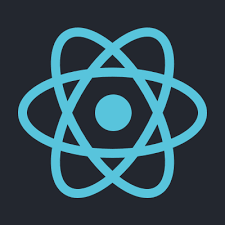
**1. React:**

**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.



- 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.

- 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.

**Các thành phần quan trọng trong ReactJS:**

- JSX (JavaScript XML) là một cú pháp mở rộng cho phép bạn viết mã giống như [HTML](https://200lab.io/blog/html-la-gi/) trong JavaScript. Trong các ngôn ngữ khác, bạn thường phải viết code HTML và JavaScript riêng rẽ. Tuy nhiên, với JSX, React cho phép bạn 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.

- **Component** là các đơn vị cơ bản trong React, cung cấp cấu trúc cho giao diện người dùng. Mỗi component được khuyến khích càng nhỏ gọn và độc lập càng tốt để có thể tái sử dụng trong nhiều phần của ứng dụng hoặc thậm chí trong các ứng dụng khác.

- Trong React, dữ liệu được quản lý và luân chuyển theo hướng từ component cha xuống component con. Các component cha có thể truyền dữ liệu cho component con thông qua props, nhưng các component con không thể trực tiếp thay đổi dữ liệu của component cha.

- Nếu muốn thay đổi dữ liệu, component con sẽ kích hoạt một hành động hoặc sự kiện (như nhấn nút), và component cha sẽ là nơi xử lý thay đổi dữ liệu đó. Giả sử chúng ta có một ứng dụng đơn giản với một button và số đếm. Mỗi khi người dùng nhấn button, số đếm sẽ tăng lên.

🡺 ReactJS là một thư viện JavaScript mạnh mẽ và linh hoạt, được thiết kế để giúp việc xây dựng giao diện người dùng trở nên đơn giản và hiệu quả hơn. Mặc dù có một số thách thức ban đầu với mô hình luồng dữ liệu một chiều (One-Way Data Binding), nhưng một khi nắm vững, React mang lại nhiều lợi ích vượt trội khi phát triển ứng dụng với quy mô lớn [1] .

**2. NodeJs:**

Node.js, là môi trường thời gian chạy (runtime environment) JavaScript đa nền tảng và mã nguồn mở. NodeJS cho phép các lập trình viên tạo cả ứng dụng front-end và back-end bằng

**JavaScript.**

NodeJS là mã nguồn mở 🡺mã nguồn của NodeJS được cung cấp công khai và được duy trì bởi những người đóng góp từ khắp nơi trên thế giới.

NodeJS không phụ thuộc vào bất kỳ phần mềm hệ điều hành nào mà đều có thể hoạt động trên Linux, macOS hoặc Windows.

NodeJS là môi trường thời gian chạy mã JavaScript 🡺 NodeJS cung cấp môi trường thời gian chạy bên ngoài trình duyệt. Nó cũng được xây dựng trên công cụ JavaScript của Chrome (V8 Engine). Điều này giúp bạn có thể xây dựng các ứng dụng back-end bằng cách sử dụng cùng ngôn ngữ lập trình JavaScript quen thuộc.



**Tính chất của NodeJS:**

- Không đồng bộ về bản chất và Hướng sự kiện (Asynchronous in Nature and Event-driven).

- Kiến trúc đơn luồng (Single Threaded Architecture).

**Ưu điểm của NodeJS:**

- NodeJS có thể mở rộng:

Các ứng dụng NodeJS có khả năng mở rộng cao vì chúng hoạt động không đồng bộ vì các yêu cầu đồng thời có thể được xử lý rất hiệu quả bằng NodeJS.

NodeJS hoạt động trên một luồng đơn nên khi có một yêu cầu đến, NodeJS sẽ bắt đầu xử lý yêu cầu đó và sẵn sàng xử lý yêu cầu tiếp theo.

- Thời gian thực thi code nhanh:

Công cụ thời gian chạy (runtime motor) JavaScript V8 được NodeJS sử dụng và cũng được Google Chrome sử dụng. Một trình bao bọc được trung tâm cung cấp cho JavaScript và vì lý do đó, công cụ thời gian chạy trở nên nhanh hơn.

Công cụ JavaScript V8 của Google Chrome là nền tảng của NodeJS, cho phép thực thi mã nhanh hơn. Công cụ này biên dịch code JavaScript thành code máy, giúp code được triển khai hiệu quả và dễ dàng hơn và nhanh hơn.

- Khả năng tương thích trên nhiều nền tảng

- NodeJS sử dụng JavaScript:

Hầu hết các lập trình viên đều quen thuộc với JavaScript, vì vậy đối với họ, việc hiểu NodeJS trở nên rất dễ dàng hơn.

- Truyền dữ liệu nhanh:

Thời gian xử lý những dữ liệu đã được truyền đến các luồng khác nhau thường sẽ mất nhiều thời gian. Trong khi đó, để xử lý dữ liệu, NodeJS chỉ mất một khoảng thời gian rất ngắn và thực hiện với tốc độ nhanh.

NodeJS tiết kiệm rất nhiều thời gian vì các tệp được NodeJS xử lý và tải lên đồng thời. Do đó, tốc độ tổng thể của truyền dữ liệu và video được cải thiện nhờ NodeJS.

- Không có bộ đệm

Dữ liệu không bao giờ được lưu vào bộ đệm trong ứng dụng NodeJS.

- Tiết kiệm thời gian, công sức và chi phí:

NodeJS được trang bị một kho thư viện khổng lồ – NPM (Node Package Manager). Các developer có thể sử dụng lại các module trong code và đưa các chức năng đa dạng vào bất kỳ ứng dụng nào.

- Một ngôn ngữ:

Một trong những lợi ích chính của NodeJS là khả năng viết toàn bộ cơ sở hạ tầng của bất kỳ ứng dụng web nào chỉ bằng một ngôn ngữ – JavaScript. Do đó, lập trình viên không cần tốn nhiều thời gian học các ngôn ngữ khác nhau để đáp ứng.

- Hỗ trợ cộng đồng mạnh mẽ:

NodeJS có sự hỗ trợ cộng đồng và hỗ trợ mạnh mẽ vì là mã nguồn mở. Vì vậy, các nhà phát triển có thể tìm kiếm sự trợ giúp từ các chuyên gia từ khắp nơi trên thế giới. Điều này thúc đẩy các dự án phát triển [2] .

**3.** [**Express.js**](https://expressjs.com/)**:**

**Express.js** là một framework đơn giản được xây dựng trên nền tảng [Node.js](https://200lab.io/blog/nodejs-la-gi/), ra đời với mục đích làm cho việc phát triển các ứng dụng web và [API](https://200lab.io/blog/api-la-gi/) trở nên đơn giản, hiệu quả và dễ bảo trì hơn. Bằng cách cung cấp các công cụ tiện lợi để xử lý yêu cầu HTTP, quản lý định tuyến (routing), và thêm các chức năng khác vào ứng dụng thông qua [middleware](https://200lab.io/blog/middleware-la-gi/).

Với Express.js, bạn có thể xây dựng các [API RESTful](https://200lab.io/blog/rest-api-la-gi/) dễ dàng hơn và quản lý dữ liệu hiệu quả mà không cần viết quá nhiều code phức tạp. Express.js giúp bạn tập trung vào logic ứng dụng thay vì xử lý chi tiết các yêu cầu HTTP.

A computer screen with icons and symbols

Description automatically generated

🡺 Express.js là một công cụ không thể thiếu cho các lập trình viên backend, nhất là khi bạn muốn xây dựng các ứng dụng web hoặc API một cách nhanh chóng và hiệu quả. Express.js là một lựa chọn tuyệt vời cho những ai đang làm việc với Node.js vì nó cân bằng tốt giữa hiệu năng và sự dễ sử dụng [3] .

**4. 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.
* 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 bạn 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.

**Khi nào nên dùng MongoDB ?**

🡺 Ví dụ như các hệ thống realtime (thời gian thực) yêu cầu phản hồi nhanh, Các hệ thống bigdata với yêu cầu truy vấn nhanh hay các hệ thống có lượng request lớn thì MongoDB sẽ là sự lựa chọn ưu tiên hơn CSDL quan hệ. Tùy theo dự án và trường hợp cụ thể để sử dụng CSDL quan hệ hay sử dụng MongoDB đem lại hiệu quả cao [4] .





Link tham khảo:

[1]

<https://200lab.io/blog/reactjs-la-gi/?srsltid=AfmBOoq8VAVDWFxST8akXaK9uE_MKePOQwkvihvUnueEOxZ2Zuvl4F6U>

[2] <https://itviec.com/blog/nodejs-la-gi/>

[3]

<https://200lab.io/blog/expressjs-la-gi/?srsltid=AfmBOoqK4R1T2Z2Fn0jYVbChTcoqJ5u3SnCpKh1bQeEoqblYwZZBj2x6>

[4] <https://viblo.asia/p/mongodb-la-gi-co-so-du-lieu-phi-quan-he-bJzKmgoPl9N>