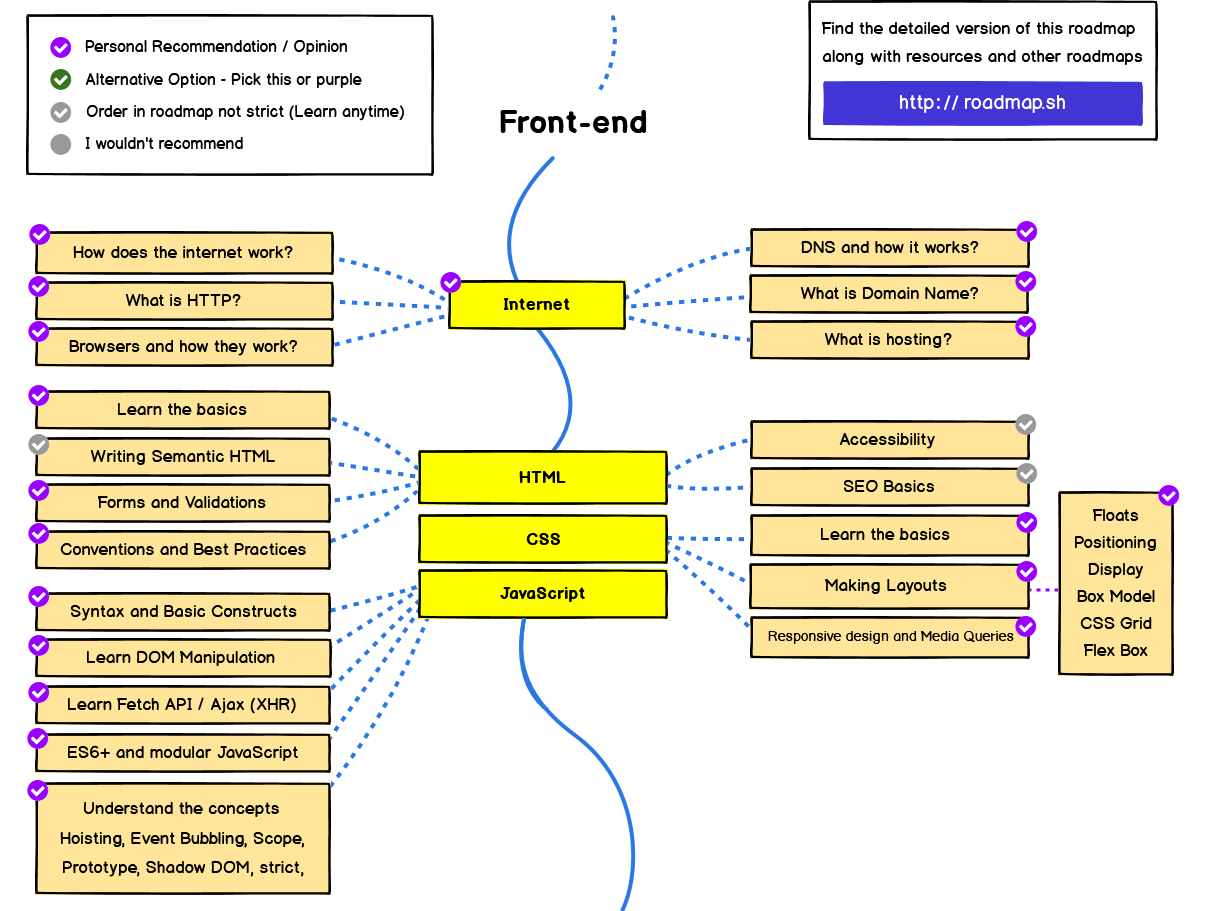
**FRONT END CAREER PATH**

**|**

**|**

1. **Nhập môn ban đầu**

****

Bắt đầu với những kiến thức cơ bản về Internet nói chung (IoT) và Website nói riêng. Cách Internet và website vận hành ra sao. Các giao thức mà website kết nối với Internet.

Domain name – Tên miền: Là tên bằng chữ của website giống như tên của một con người. ví dụ: [www.google.com](http://www.google.com)

Hosting: là nơi được lưu trữ dữ liệu được chia nhỏ từ các server, từ đó giúp upload và render website, app trên internet.

**HTML**

Kiến thức cơ bản nhất của Front End là HTML (Hyper Text Markup Language), dùng để cấu trúc các element để tạo hình cho một website.

HTML chỉ có tác dụng bố cục và định dạng trang web. HTML khi kết hợp với CSS và JavaScript sẽ trở thành một website hoàn chỉnh.

* Một số thuật ngữ cơ bản trong HTML:  
  Elements: Là các chỉ định xác định nội dung, cấu trúc của các đối tượng trong một Website. Tên Element được bao quang, xác định bằng dấu ngoặc < >. Những yếu tố được sử dụng phổ biến là đoạn văn ( <p>), các cấp độ tiêu đề (từ <h1> đến <h6>), danh sách tiếp tục bao gồm <a>, <div>, <span>, <strong>, và <em>,…
* Tags: Một Element được bao quanh bởi các dấu ngoặc < > sẽ tạo ra các thẻ. Ví dụ thẻ mở là dấu hiệu cho bạn biết sự bắt đầu của một Element (ví dụ: <div>). Thẻ đóng sẽ đánh dấu vào cuối của một Element, có hình thức là dấu ngặc nhỏ + dấu chéo + dấu ngoặc lớn (ví dụ: </div>). Ở giữa thẻ mở và thẻ đóng là nội dung của Element.
* Attributes: Là thuộc tính sử dụng để cung cấp thông tin bổ sung về một Element. Các Attributes bao gồm tên và giá trị, được xác định sau tên của một thành phần và trong thẻ mở. Attributes có định dạng như sau: tên thuộc tính + dấu bằng + giá trị thuộc tính được trích dẫn. Ví dụ Element <a> gồm một Attribute href: <a href=”http://shayhowe.com/”>Shay Howe</a>

Hiện tại HTML được nâng cấp lên chuẩn HTML5 với nhiều tag được thêm vào markup, mục đích là để xác định rõ nội dung thuộc loại là gì (ví dụ như: <article>, <header>, <[footer](https://wiki.matbao.net/footer/)>,…).

Theo Mozilla Developer Network (https://developer.mozilla.org/) thì HTML Element Reference hiện nay có khoảng hơn 140 tag.

HTML5 đã được bổ sung thêm rất nhiều tính năng quan trọng. Cụ thể, một số tính năng mới của HTML5 có thể kể đến như:

* Khả năng hỗ trợ video và audio: Thay vì phải sử dụng Flash Player để phát video/audio thì với HTML5, lập trình viên có thể nhúng trực tiếp file video/audio vào trang web bằng cặp tag <audio></audio> hoặc <video></video>.
* Hỗ trợ scalable vector graphic (SVG) và MathML cho các công thức toán học hoặc phương trình hóa học.
* Có thêm nhiều cải thiện về mặt ngôn ngữ. Chứa nhiều semantic tag giúp người dùng hiểu được nội dung chính của Website là gì (ví dụ như: <article></article>, <section></section>, <aside></aside>, <header></header>, <footer></footer>).
* Hỗ trợ nhiều loại kiểu form hơn so với HTML như: [email cho doanh nghiệp](https://www.matbao.net/email.html), ngày/giờ, số điện thoại, URL, tìm kiếm,…

**CSS**

CSS (Cascading Style Sheet) 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). Nói ngắn gọn hơn là ngôn ngữ tạo phong cách cho trang web. Bạn có thể hiểu đơn giản rằng, nếu HTML đóng vai trò định dạng các phần tử trên website như việc tạo ra các đoạn văn bản, các tiêu đề, bảng,…thì CSS sẽ giúp chúng ta có thể thêm style vào các phần tử HTML đó như đổi bố cục, màu sắc trang, đổi màu chữ, font chữ, thay đổi cấu trúc…

***Bố cục của một đoạn CSS***

Bố cục CSS thường chủ yếu dựa vào hình hộp và mỗi hộp đều chiếm những khoảng trống trên trang của bạn với các thuộc tính như:

* Padding (vùng đệm): Gồm không gian xung quanh nội dung (ví dụ: xung quanh đoạn văn bản).
* Border (đường viền): Là đường liền nằm ngay bên ngoài phần đệm.
* Margin (lề): Là khoảng cách xung quanh bên ngoài của phần tử.

***Cấu trúc một đoạn CSS***

Cấu trúc của một đoạn CSS sẽ bao gồm các phần cơ bản như sau:

Phần thứ 1: Vùng chọn  {

Phần thứ 2: thuộc tính :

Phần thứ 3: giá trị;

Phần thứ 4: }

Phần thứ 5: …..

***Ý nghĩa của các phần này như sau:***

* Bộ chọn (Selector): Tên phần tử HTML bắt đầu của bộ quy tắc và nó sẽ thực hiện chọn những phần tử đã được tạo kiểu. Từ đó, có thể tạo được kiểu cho những phần tử khác mà chỉ cần thực hiện thay đổi bộ chọn.
* Tuyên bố (Declaration): Một quy tắc duy nhất như: color: red; xác định thuộc tính của phần tử nào bạn muốn tạo kiểu.
* Thuộc tính (Properties): Những cách mà bạn có thể tạo kiểu cho một phần tử HTML. (Với trường hợp này thì color được xem là một trong những  thuộc tính của phần tử p). Chính vì vậy, với CSS thì bạn chỉ cần lựa chọn thuộc tính mà chính bạn muốn tác động nhất trong bộ quy tắc của mình.
* Giá trị thuộc tính: bên phải của thuộc tính sau dấu hai chấm(:), chúng ta sẽ sở hữu giá trị thuộc tính mà việc lựa chọn trong số đó sẽ xuất hiện nhiều lần để có thể cho một thuộc tính cụ thể nào đó.

Ngoài ra, còn có thể việt CSS theo một phương pháp chuyên nghiệp và nâng cao hơn là SASS.

SASS( Syntactically Awesome Style Sheets) là một tiện ích mở rộng của CSS. Các ngôn ngữ style sheets kiểm vị trí và cách thức văn bản hiện thị trên webpage, từ kích thước và màu sắc khung cho đến vị trí menu.

Code SASS được tổ chức tốt hơn so với CSS. SASS có khối lượng code nhỏ hơn, do đó, code của SASS dễ đọc và hiểu hơn, đặc biệt trong những dự án web với nhiều developer khác nhau.

SASS có khả năng tái sử dụng. SASS cho phép tái sử dụng các biên và đoạn code nhiều lần, giúp các developer tiết kiệm được nhiều thời gian và giảm thiểu bug trong code.

**JavaScript (JS)**

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. 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 bạn 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 bạn thấy các hiệu ứng của JavaScript.

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.

**Công cụ JavaScript**

Công cụ JavaScript là một chương trình máy tính chạy mã JavaScript. Các công cụ JavaScript đầu tiên chỉ là bộ diễn giải, nhưng tất cả các công cụ hiện đại đều sử dụng biên dịch tại chỗ hoặc thời gian chạy để cải thiện hiệu suất.

**JavaScript phía máy khách**

JavaScript phía máy khách đề cập đến cách thức JavaScript hoạt động trong trình duyệt của bạn. Trong trường hợp này, công cụ JavaScript nằm trong mã trình duyệt. Tất cả các trình duyệt web phổ biến đều có các công cụ JavaScript tích hợp riêng.

Các nhà phát triển ứng dụng web viết mã JavaScript với các hàm khác nhau được liên kết với nhiều loại sự kiện, chẳng hạn như nhấp chuột hoặc khi di chuột đến. Các hàm này thực hiện các thay đổi đối với HTML và CSS.

Sau đây là tổng quan về cách thức hoạt động của JavaScript phía máy khách:

1.   Trình duyệt tải một trang web khi bạn truy cập trang đó.

2.   Trong khi tải, trình duyệt chuyển đổi trang và tất cả các phần tử của trang, chẳng hạn như các nút, nhãn và hộp thả xuống, thành một cấu trúc dữ liệu được gọi là Mô hình đối tượng tài liệu (DOM).

3.   Công cụ JavaScript của trình duyệt chuyển đổi mã JavaScript thành mã bytecode. Mã này đóng vai trò là trung gian giữa cú pháp JavaScript và máy.

4.   Các sự kiện khác nhau, chẳng hạn như nhấp vào nút, sẽ kích hoạt việc thực thi khối mã JavaScript được liên kết. Sau đó, công cụ này sẽ diễn giải bytecode và thực hiện các thay đổi đối với DOM.

5.   Trình duyệt hiển thị DOM mới.

**JavaScript phía máy chủ**

JavaScript phía máy chủ đề cập đến việc sử dụng ngôn ngữ viết mã trong logic máy chủ back-end. Trong trường hợp này, công cụ JavaScript nằm trực tiếp trên máy chủ. Hàm JavaScript phía máy chủ có thể truy cập cơ sở dữ liệu, thực hiện các phép toán logic khác nhau và phản hồi nhiều loại sự kiện do hệ điều hành của máy chủ kích hoạt. Ưu điểm chính của viết tập lệnh phía máy chủ là bạn có khả năng tùy chỉnh linh hoạt phản hồi của trang web dựa trên yêu cầu và quyền truy cập của mình cũng như theo yêu cầu thông tin từ trang web.

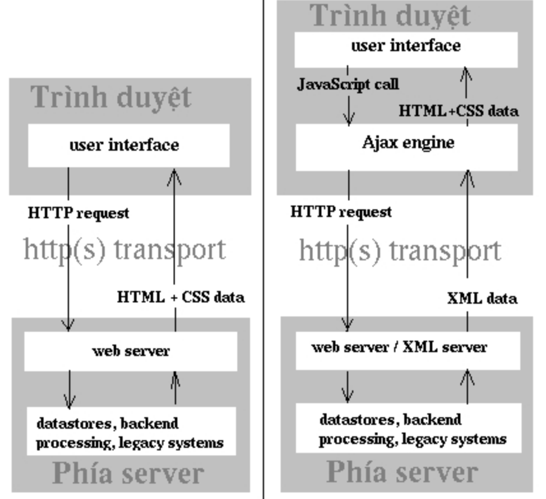
Lợi thế của JS nổi bật là việc tạo, cập nhật, thay đổi sửa và xoá dữ liệu cục bộ mà khong phải load lại toàn bộ trang web. Thời gian trước, xây dựng website thì công nghệ sử dụng dựa hoàn toàn vào HTML và mỗi thao tác từ người dùng thì đều phải tải lại trang từ trang chủ. Điều này làm cho trang web hoạt động kém hiệu quả và trải nghiệm không tốt đói với người dùng làm quá tải máy chủ, tăng băng thông truyền tải, lãng phí tài nguyên.

Vì thế cần công nghệ cải thiện việc này và AJAX đã ra đời bù đắp những nhược điểm của web truyền thống.

AJAX : Asynchronous Javascript And XML

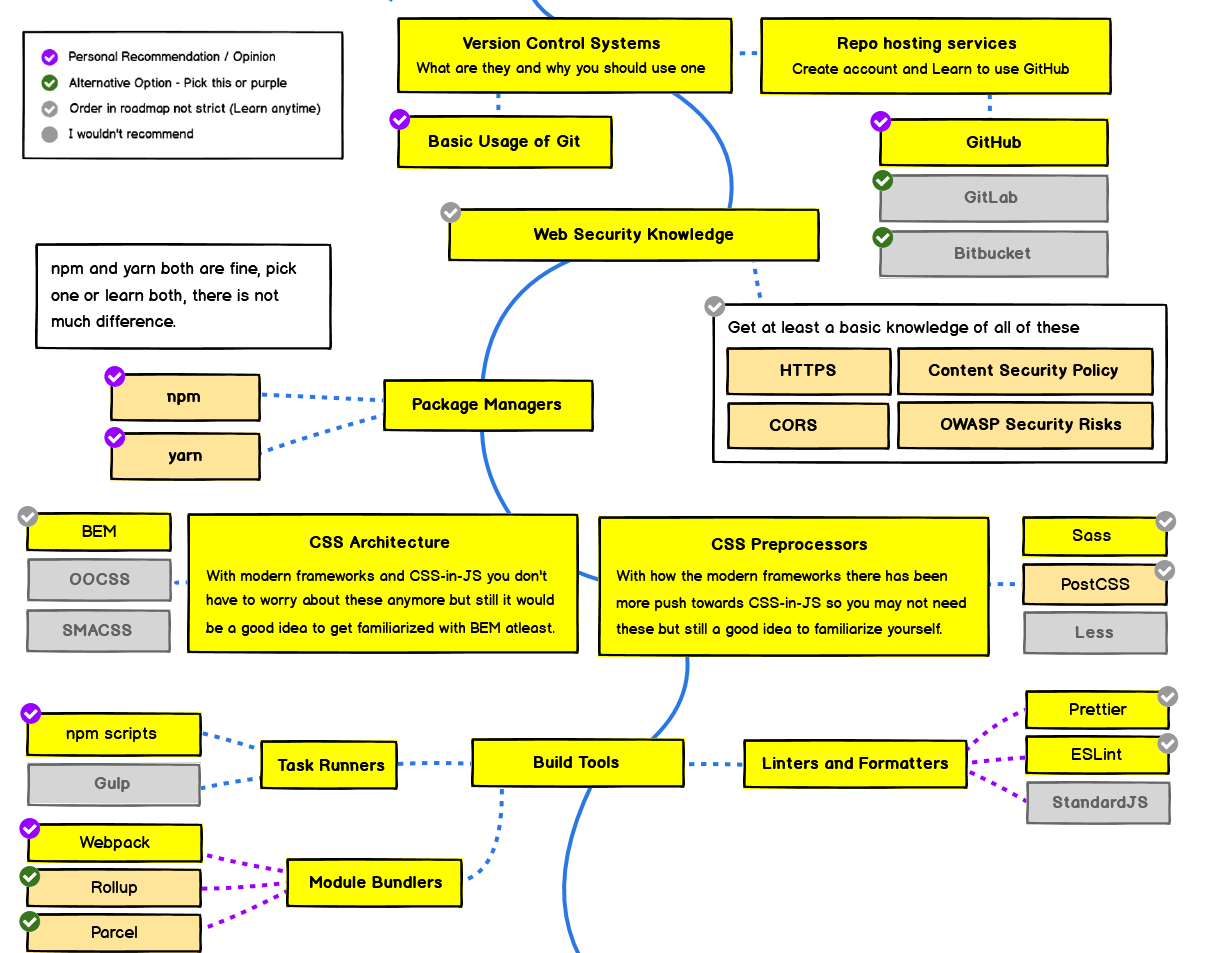
AJAX không phải ngôn ngữ lập trình mà kết hợp các công nghệ với nhau và điểm khác biệt cơ bản với cách truyền thống là xử lý thông tin trên client thay vì trên server vừa giảm tải cho serer lại tăng trải nghiệm của người dùng.

Cụ thể nhờ vào việc hoạt động bất đồng bộ asyncheonous để xử lý nhiều yêu cầu cùng một lúc dựa vào JS, HTML DOM (để hiển thị hoặc sử dụng dữ liệu) và dùng XML (Extensible Markup Language) để chứa dữ liệu trong quá trình trao đổi nhận và gửi dữ liệu từ browser tới server.

Qua hình có thể thấy khác biệt rõ ràng là với cách truyến thống thì với mỗi request từ client thì server phải gửi cả html, css và data cho client tải lại trang và hiển thị cho user.

Còn với AJAX thì 2 bên client và server chỉ trao đổi XML data còn lại việc hiển tại html, css thì do AJAX đảm nhiệm và việc này giảm tải cho server và tiết kiệm băng thông đồng thời tăng trải nghiệm cho người dùng giúp họ thấy việc tương tác với trang web nhanh hơn.

1. **Nâng cao chuyên sâu**



**Repo hosting services (Github):**

Đây là hệ thống quản lý dự án và phiên bản code. Cách thức hoạt động như trang mạng xã hội cho các lập trình viên clone lại mã nguồn từ Repository.

**Ưu điểm đến từ việc sử dụng GitHub**

Những lợi ích cụ thể các nhà lập trình được hưởng từ GitHub là gì? Bạn sẽ biết điều đó thông qua thế mạnh tổng hợp dưới đây. Các coder chắc chắn nhận thấy tiềm năng và động lực để ứng dụng ngay trong thực tế.

**Dễ dàng quản lý Source Code**

Dễ dàng quản lý Source Code là điều đã được nói đến khi giải thích GitHub là gì. Khi bạn tạo repo, toàn bộ thành quả sẽ được lưu trên công cụ này. Tại đây, bạn dễ dàng theo dõi lại quá trình mình đã làm việc.

Điều đó được nhìn thấy thông qua các comment sau mỗi lần commit. Điều thú vị hơn nữa chính là nhiều người có thể cùng làm một repo. Từ đó, bạn biết được cụ thể ai đã làm gì, chỉnh sửa ở đâu.

Tiếp theo, nguồn của bạn được phát triển theo nhiều nhánh khác nhau. Nguyên tắc làm việc với GitHub là gì? Bạn có thể rẽ các hướng để phát triển, mở rộng dự án. Đến cuối cùng, các phần phải merge lại để tạo ra project hoàn chỉnh.

**Tracking những thay đổi qua các version**

Đối với những dự án có nhiều thành viên cùng phát triển, hiểu GitHub là gì rất cần thiết. Khi làm theo nhóm, nhu cầu lớn nhất là theo dõi revisions – ai thay đổi cái gì, lúc nào và lưu tệp ở đâu.

Công cụ đã tính đến nhu cầu này và tối ưu tốt nhất có thể. Điều đó thể hiện qua việc luôn lưu lại những thay đổi đã push lên repository. Giống như Google Drive, bạn sẵn có phiên bản dự phòng để thay thế cho tình huống bị xóa mất dữ liệu chưa lưu.

**Markdown**

Markdown trong GitHub là gì? Đây là cách định dạng text trên web. Bạn có thể chỉnh sửa việc hiển thị của document, format,…Chẳng hạn như dạng in đậm, in nghiêng, thêm hình, tạo list cho những thứ cần Markdown.

Hầu hết, hình thức này chỉ là đoạn text đơn thuần với những ký tự đặc biệt được chèn thêm vào. Trong GitHub, bạn dễ dàng sử dụng ở các vị trí như: Git, Comments tại Issues và Pull Requests. File có đuôi .md hay .markdown extension cũng xuất hiện.

**Khẳng định chuyên môn**

Sau khi biết **GitHub là gì và cách sử dụng**, trình độ chuyên môn của bạn nâng lên tầm cao mới. Những Source Code luôn là minh chứng tốt nhất để bạn khẳng định mình là developer thực thụ.

Trong mắt nhà tuyển dụng, GitHub giống như chiếc máy liar – detech. Thông qua đó, họ dễ dàng phân biệt developer thật và giả. Vì thế, hãy đầu tư một tài khoản và dẫn link vào CV.

Thậm chí, nhiều công ty lớn trên thế giới còn xem đây là yêu cầu trong quy trình tuyển dụng. Những đóng góp của bạn cho cộng động Coder chính là lợi thế cạnh tranh không thể phủ nhận.

**Cải thiện khả năng Code**

**GitHub hướng dẫn** cung cấp cho bạn một cách học vô cùng hiệu quả. Điều này có được nhờ hàng vạn dự án mở, người đóng góp và hàng tỷ commit mỗi ngày.

Chỉ cần đăng nhập vào nền tảng, một kho tàng kiến thức update nhất đã mở ra. Việc tự phân tích, so sánh giúp tiếp cận kiến thức chuyên môn chủ động hơn. Bên cạnh đó, đây là cơ hội cọ sát với project gần gũi trong thực tế.

­­

**Kho tài nguyên tuyệt vời**

Khi tìm hiểu GitHub là gì, bạn sẽ thấy chức năng Explore tuyệt vời như thế nào. Người dùng dễ dàng theo dõi, tìm kiếm các open source projects theo chính xác technology pattern.

Công cụ hỗ trợ Code Search không kể ở dưới dạng dự án riêng hay website. Bên cạnh đó, kỹ thuật SEO khá tốt nên người dùng có thể tìm kiếm bất kỳ Code String nào được chia sẻ.

Ngoài ra, cần một số kiến thức về Package Manager, tiêu biểu có 2 package phổ biến là **npm và yarn.**

**NPM** là viết tắt của Node package manager là một công cụ tạo và quản lý các thư viện lập trình Javascript cho [Node.js](https://nodejs.org/). Trong cộng đồng Javascript, các lập trình viên chia sẻ hàng trăm nghìn các thư viện với các đoạn code đã thực hiện sẵn một chức năng nào đó. Nó giúp cho các dự án mới tránh phải viết lại các thành phần cơ bản, các thư viện lập trình hay thậm chí cả các [framework](https://topdev.vn/blog/framework-la-gi/).

Với npm , công việc sẽ đơn giản đi rất nhiều, chúng giúp bạn thực hiện việc quản lý đơn giản hơn rất nhiều. Các thư viện đều có sẵn trên npm, bạn chạy một dòng lệnh để tải về và dễ dàng include chúng hơn.

Mỗi đoạn code này có thể phụ thuộc vào rất nhiều các mã nguồn mở khác, thật may mắn khi các công cụ quản lý thư viện ra đời, nếu không sẽ mất rất nhiều công sức trong việc quản lý các thư viện này.

Cộng đồng sử dụng npm rất lớn, hàng nghìn các thư viện được phát hành, hỗ trợ Javascript ES6, React, Express, Grunt, Duo… Hiện nay cũng đã xuất hiện thêm Yarn một công cụ tương tự npm, được Facebook phát triển với nhiều tính năng vượt trội có khả năng sẽ thay thế npm.

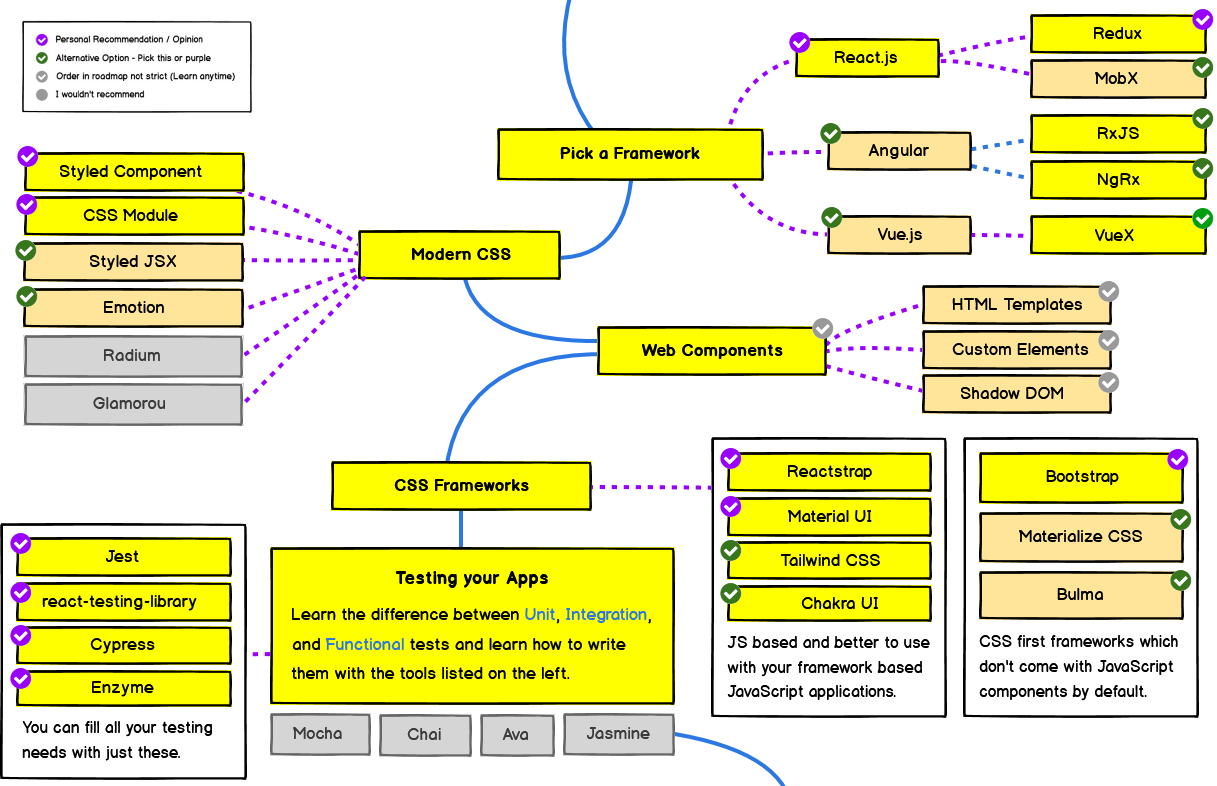
Để quản lý các gói cài đặt cục bộ bằng npm thì cách tốt nhất là thông qua file package.json, chính là file nằm trong thư mục gốc của project. File [JSON](https://topdev.vn/blog/json-la-gi/) này chứa các nội dung:

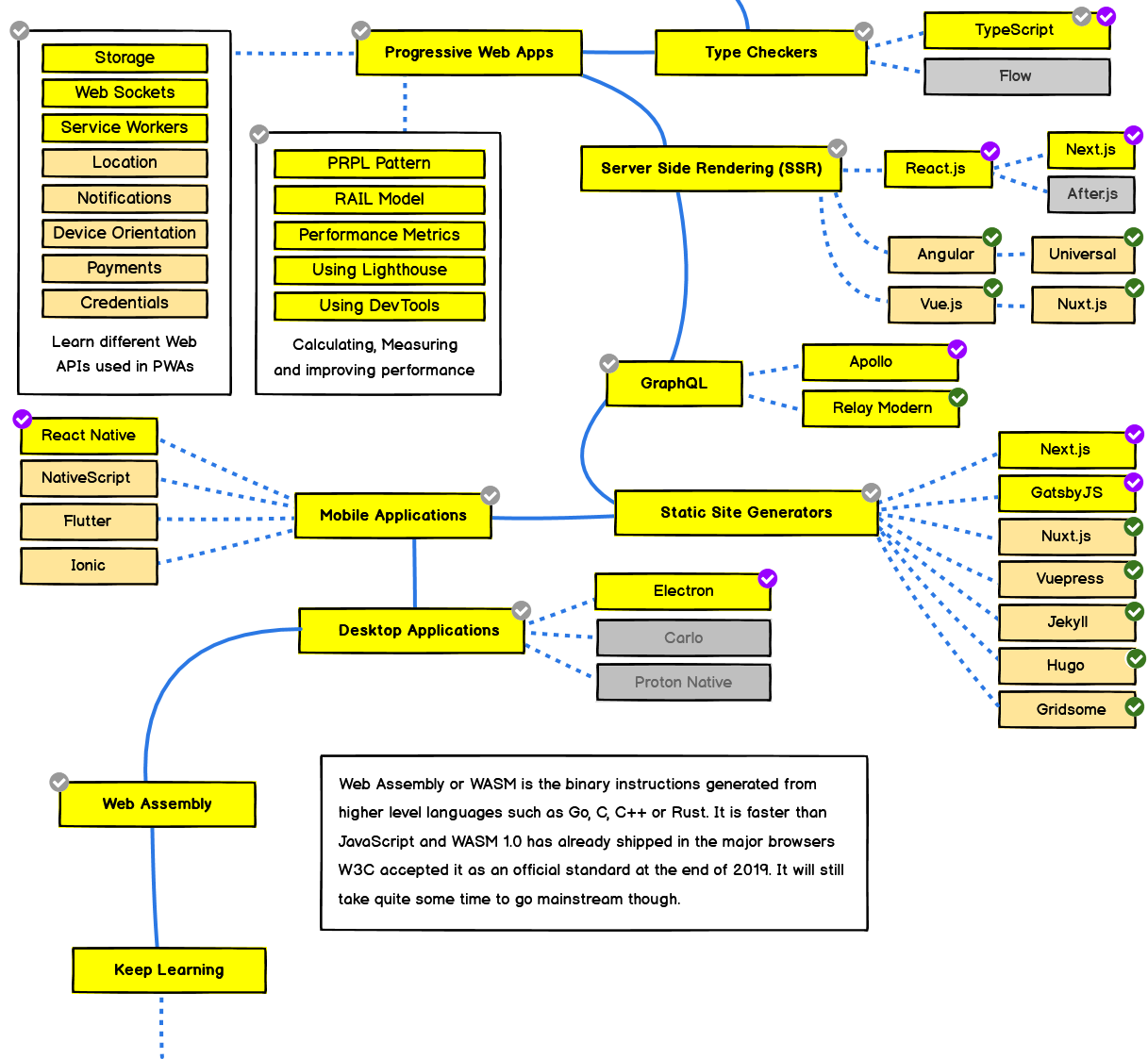
* Các gói thư viện lập trình mà project sử dụng.
* Cho phép xác định phiên bản chính xác của các gói thư viện lập trình được sử dụng.
* Các gói bạn xây dựng có thể chia sẻ dễ dàng với các lập trình viên khác trên toàn cầu thông qua npm.

Có một số các thuộc tính trong package.json, chúng ta cùng điểm qua:

* **name**: tên gói thư viện
* **version**: phiên bản gói
* **description**: phần mô tả về gói thư viện
* **homepage**: trang chủ của gói
* **author**: tác giả
* **contributors**: tên người đóng góp cho package
* **dependencies**: danh sách các gói phụ thuộc, tự động được cài theo.
* **repository**: loại repository và url của package, thông thường là git (Xem thêm [Git là gì](https://topdev.vn/blog/git-la-gi/)?)
* **main**: index.js
* **keywords**: các từ khóa

1. **Chuyên sâu**



Lựa chọn Framework - là các đoạn code đã được viết sẵn, cấu thành nên một bộ khung và các thư viện lập trình được đóng gói. Chúng cung cấp các tính năng có sẵn như mô hình, API và các yếu tố khác để tối giản cho việc phát triển các ứng dụng web phong phú, năng động. Các **framework**giống như là chúng ta có khung nhà được làm sẵn nền móng cơ bản, bạn chỉ cần vào xây dựng và nội thất theo ý mình.

**Framework** giúp tăng hiệu suất, mở rộng các chức năng và cung cấp các thư viện sẵn có để các nhà phát triển không phải làm lại từ đầu

Một framework cho web bao gồm:

* **Libraries**: Thư viện là các đoạn mã xây dựng sẵn cho một chức năng nào đó, bạn có thể tái sử dụng chức năng đó mà ko phải code lại. Có nhiều tools quản lý thư viện như [NPM](https://topdev.vn/blog/npm-la-gi/), [Composer](https://topdev.vn/blog/composer-la-gi/)…
* **API:**là phương thức trao đổi dữ liệu giữa các ứng dụng chính và ứng dụng khác. Đọc thêm [RESTful API là gì](https://topdev.vn/blog/restful-api-la-gi/) nhé.
* **Scaffolding:** Một bộ khung các quy tắc mà một framework MVC sử dụng quy hoạch cơ sở dữ liệu có thể được truy cập như thế nào.
* **AJAX:** update thông tin lên database mà không cần load lại trang.
* **Caching:** giúp giảm request đến máy chủ, tăng tốc độ load trang.
* **Security:** framework xác thực và ủy quyền user.
* **Compilers :**trình biên dịch từ code của bạn qua ngôn ngữ cho máy.

Các framework phổ biến hiện nay: ReactJS / AngularJS / VueJS

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

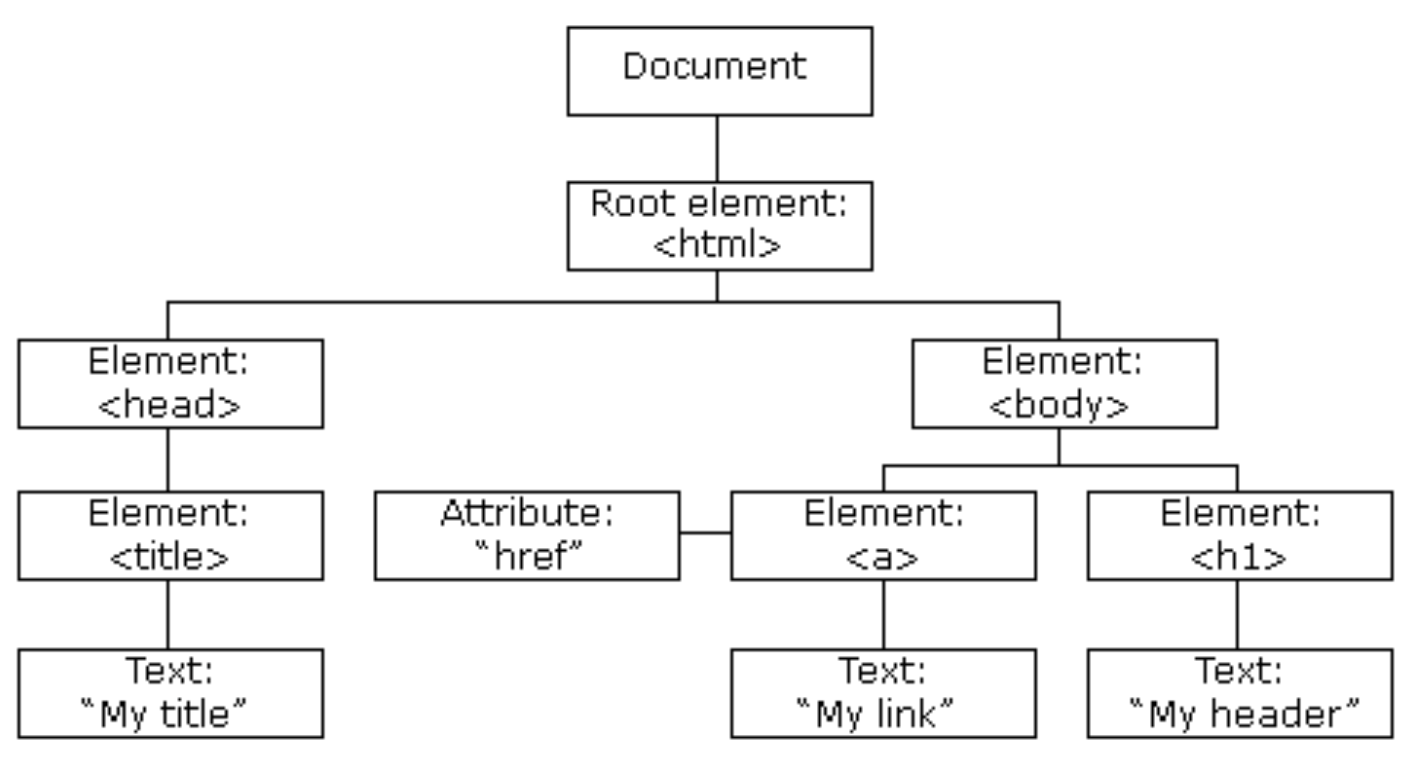
ReactJS là một thư viện JavaScript chuyên giúp các nhà phát triển xây dựng giao diện người dùng hay UI. Trong lập trình ứng dụng front-end, lập trình viên thường sẽ phải làm việc chính trên 2 thành phần sau: UI và xử lý tương tác của người dùng. UI là tập hợp những thành phần mà bạn nhìn thấy được trên bất kỳ một ứng dụng nào, ví dụ có thể kể đến bao gồm: menu, thanh tìm kiếm, những nút nhấn, card,… Giả sử bạn đang lập trình một website thương mại điện tử, sau khi người dùng chọn được sản phẩm ưng ý rồi và nhấn vào nút “Thêm vào giỏ hàng”, thì việc tiếp theo mà bạn phải làm đó là thêm sản phẩm được chọn vào giỏ hàng và hiển thị lại sản phẩm đó khi user vào xem => xử lý tương tác.

Trước khi có ReactJS, lập trình viên thường gặp rất nhiều khó khăn trong việc sử dụng “vanilla JavaScript”(JavaScript thuần) và JQuery để xây dựng UI. Điều đó đồng nghĩa với việc quá trình phát triển ứng dụng sẽ lâu hơn và xuất hiện nhiều bug, rủi ro hơn. Vì vậy vào năm 2011, Jordan Walke – một nhân viên của Facebook đã khởi tạo ReactJS với mục đích chính là cải thiện quá trình phát triển UI.

Hơn nữa, để tăng tốc quá trình phát triển và giảm thiểu những rủi ro có thể xảy ra trong khi coding, React còn cung cấp cho chúng ta khả năng Reusable Code (tái sử dụng code) bằng cách đưa ra 2 khái niệm quan trọng bao gồm:

* JSX.
* Virtual DOM.

**JSX**

****

Trọng tâm chính của bất kỳ website cơ bản nào đó là những HTML documents. Trình duyệt Web đọc những document này để hiển thị nội dung của website trên máy tính, tablet, điện thoại của bạn. Trong suốt quá trình đó, trình duyệt sẽ tạo ra một thứ gọi là Document Object Model (DOM) – một tree đại diện cho cấu trúc website được hiển thị như thế nào. Lập trình viên có thể thêm bất kỳ dynamic content nào vào những dự án của họ bằng cách sử dụng ngôn ngữ JavaScript để thay đổi cây DOM.

JSX (nói ngắn gọn là JavaScript extension) là một React extension giúp chúng ta dễ dàng thay đổi cây DOM bằng các HTML-style code đơn giản. Và kể từ lúc ReactJS browser hỗ trợ toàn bộ những trình duyệt Web hiện đại, bạn có thể tự tin sử dụng JSX trên bất kỳ trình duyệt nào mà bạn đang làm việc.

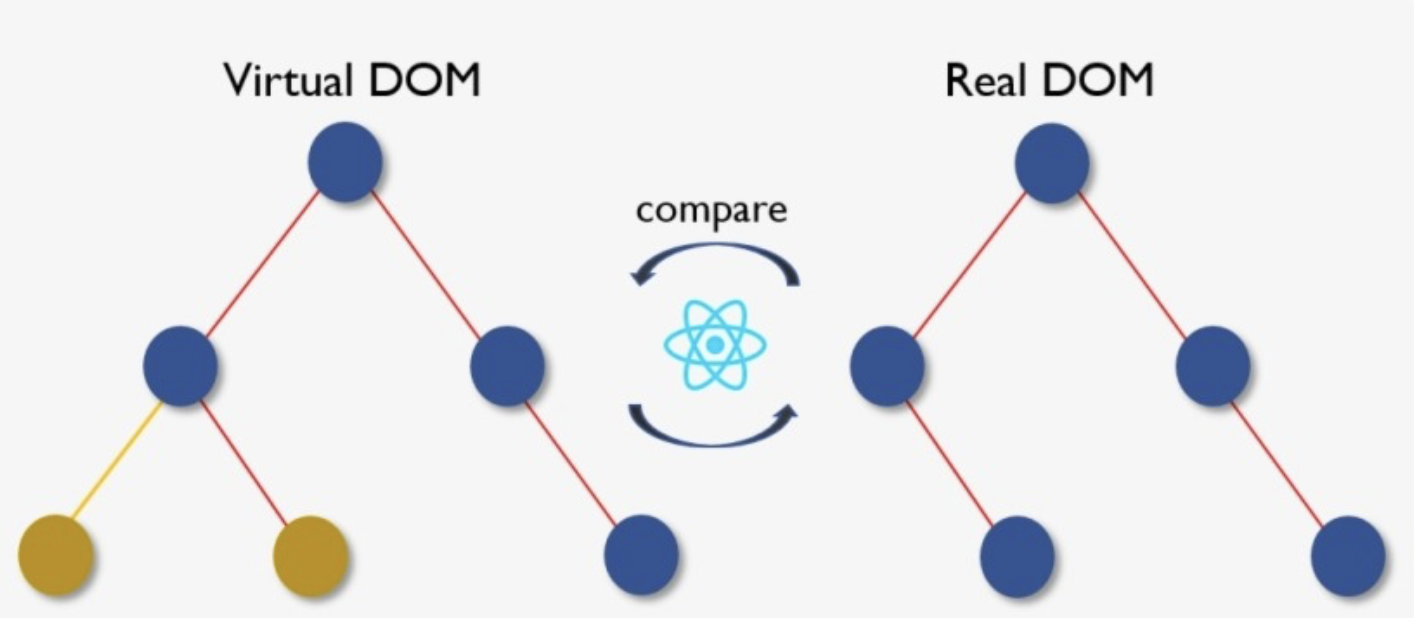
****

**Virtual DOM**

Nếu bạn không sử dụng ReactJS (và JSX), website của bạn sẽ sử dụng HTML để cập nhật lại cây DOM cho chính bản nó (quá trình thay đổi diễn ra tự nhiên trên trang mà người dùng không cần phải tải lại trang), cách làm này sẽ ổn cho các website nhỏ, đơn giản, static website. Nhưng đối với các website lớn, đặc biệt là những website thiên về xử lý các tương tác của người dùng nhiều, điều này sẽ làm ảnh hưởng performance website cực kỳ nghiêm trọng bởi vì toàn bộ cây DOM phải reload lại mỗi lần người dùng nhấn vào tính năng yêu cầu phải tải lại trang).

Tuy nhiên, nếu bạn sử dụng JSX thì bạn sẽ giúp cây DOM cập nhật cho chính DOM đó, ReactJS đã khởi tạo một thứ gọi là Virtual DOM (DOM ảo). Virtual DOM (bản chất của nó theo đúng tên gọi) là bản copy của DOM thật trên trang đó, và ReactJS sử dụng bản copy đó để tìm kiếm đúng phần mà DOM thật cần cập nhật khi bất kỳ một sự kiện nào đó khiến thành phần trong nó thay đổi (chẳng hạn như user nhấn vào một nút bất kỳ).

Ví dụ, khi người dùng bình luận vào khung comment vào bất kỳ bài Blog nào trên website của bạn và nhấn “Enter”. Dĩ nhiên, người dùng của bạn sẽ cần phải thấy được bình luận của mình đã được thêm vào danh sách bình luận. Giả sử trong trường hợp không sử dụng ReactJS, toàn bộ cây DOM sẽ phải cập nhật để báo hiệu sự thay đổi mới này. Còn khi bạn sử dụng React, nó sẽ giúp bạn scan qua Virtual DOM để xem những gì đã thay đổi sau khi người dùng thực hiện hành động trên (trong trường hợp này, thêm mới bình luận) và lựa chọn đúng nơi đúng chỗ cần cập nhật sự thay đổi mà thôi.



Với việc cập nhật đúng chỗ như vậy, khỏi phải nói nó tiết kiệm cho chúng ta rất nhiều tài nguyên cũng như thời gian xử lý. Ở các website lớn và phức tạp như thương mại điện tử, đặt món ăn, v.v bạn sẽ thấy việc này là vô cùng cần thiết và quan trọng để làm tăng trải trải nghiệm của khách hàng và performance được cải thiện đáng kể.

**Ưu điểm của ReactJS**

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

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

Bên cạnh framework để hỗ trợ các cấu trúc có sẵn thì vẫn còn các library thiết kế sẵn các component. Một số library như: Bootstrap, Tailwind CSS, MUI ….