

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP TIỂU LUẬN CUỐI KÌ MÔN LẬP TRÌNH WEB
VÀ ỨNG DỤNG**

SINGLE PAGE APPLICATION (SPA)

Người hướng dẫn: **THẦY TRẦN BẢO TÍN**

Người thực hiện: **MAI VĂN NGỌC – 51900147**

PHẠM HÙNG NGHĨA – 51900145

NGUYỄN THANH SƠN – 51900200

Lớp : 19050202

Khoá : 23

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2021

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP TIỂU LUẬN CUỐI KÌ MÔN LẬP TRÌNH WEB
VÀ ỨNG DỤNG**

SINGLE PAGE APPLICATION (SPA)

Người hướng dẫn: **THẦY TRẦN BẢO TÍN**

Người thực hiện: **MAI VĂN NGỌC – 51900147**

PHẠM HÙNG NGHĨA – 51900145

NGUYỄN THANH SƠN – 51900200

Lớp : 19050202

Khoá : 23

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2021

LỜI CẢM ƠN

Trước tiên, nhóm nghiên cứu xin gửi lời cảm ơn chân thành và lòng biết ơn sâu sắc đến Thầy Trần Bảo Tín. Thầy là người đã luôn hỗ trợ và hướng dẫn tận tình cho chúng tôi trong suốt quá trình nghiên cứu và hoàn thành bài nghiên cứu với đề tài ***“Single Page Application (SPA)”***.

Tiếp theo, nhóm chúng tôi xin gửi lời cảm ơn đến khoa Công Nghệ Thông Tin trường Đại học Tôn Đức Thắng vì đã tạo điều kiện cho chúng tôi được học tập và nghiên cứu môn học này. Khoa đã luôn sẵn sàng chia sẻ các kiến thức bổ ích cũng như chia sẻ các kinh nghiệm tham khảo tài liệu, giúp ích không chỉ cho việc thực hiện và hoàn thành đề tài nghiên cứu mà còn giúp ích cho việc học tập và rèn luyện trong quá trình học trực tuyến nói chung.

Cuối cùng, sau khoảng thời gian học tập trực tuyến chúng tôi đã hoàn tất đề tài nghiên cứu nhờ vào sự hướng dẫn, giúp đỡ và những kiến thức học hỏi được từ Quý thầy cô. Do giới hạn về mặt kiến thức và khả năng lý luận nên nhóm vẫn còn nhiều thiếu sót và hạn chế, kính mong sự chỉ dẫn và đóng góp của Quý thầy cô giáo để bài Nghiên cứu của chúng tôi được hoàn thiện hơn. Hơn nữa, nhờ những góp ý từ thầy cô và các bạn hữu, chúng tôi sẽ hoàn thành tốt hơn ở những bài nghiên cứu trong tương lai. Nhóm chúng tôi mong Quý thầy cô và các bạn bè – những người luôn quan tâm và hỗ trợ chúng tôi – luôn tràn đầy sức khỏe và sự bình an.

TẬP THỂ NHÓM NGHIÊN CỨU XIN CHÂN THÀNH CẢM ƠN!

ĐỒ ÁN ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Tôi xin cam đoan đây là sản phẩm đồ án của riêng chúng tôi và được sự hướng dẫn của thầy Trần Bảo Tín. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình. Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày 02 tháng 01 năm 2022

Tác giả

(ký tên và ghi rõ họ tên)

Mai Văn Ngọc

Phạm Hùng Nghĩa

Nguyễn Thanh Sơn

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

Phần xác nhận của GV hướng dẫn

Tp. Hồ Chí Minh, ngày tháng năm
(kí và ghi họ tên)

Phần đánh giá của GV chấm bài

Tp. Hồ Chí Minh, ngày tháng năm
(kí và ghi họ tên)

TÓM TẮT

Trong bài báo cáo này bọn em sẽ tìm hiểu toàn diện về Single Page Application bao gồm nguyên nhân ra đời, khái niệm, nguyên lý hoạt động,....

Sau đó bọn em sẽ tiến hành tìm hiểu về framework Vue và sẽ tiến hành viết một trang web “**Dự báo thời tiết**” minh họa Single page application sử dụng framework Vue. Trang web minh họa về Single page application của bọn em sẽ lấy dữ liệu từ một Web API đó là trang web OpenWeatherMap API.

MỤC LỤC

| | |
|---|-----|
| LỜI CẢM ƠN | i |
| PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN | iii |
| TÓM TẮT | iv |
| MỤC LỤC..... | 1 |
| DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT | 2 |
| CHƯƠNG 1 – MỞ ĐẦU | 3 |
| 1.1 Sơ lược về Single Page Application (SPA) | 3 |
| 1.1.1 Nguyên nhân mà Single Page Application ra đời | 3 |
| 1.1.2 Single Page Application là gì? | 4 |
| CHƯƠNG 2 – TỔNG QUAN VỀ SINGLE PAGE APPLICATION | 5 |
| 2.1 Cách thức hoạt động của Single Page Application (SPA)..... | 5 |
| 2.2 Những khác biệt trong hoạt động của Single Page Application và Web truyền thống | 6 |
| 2.2.1 . <i>Web Single Page Application có backend và frontend rõ ràng</i> | 7 |
| 2.2.2 . <i>Web Single Page Application đẩy mạnh xử lý ở frontend</i> | 7 |
| 2.3 Những ưu điểm của Single Page Application..... | 8 |
| 2.3.1 <i>Hạn chế truy vấn đến Server</i> | 8 |
| 2.3.2 <i>Tăng trải nghiệm người dùng</i> | 8 |
| 2.3.3 <i>Xây dựng frontend nhanh và có tính tương thích</i> | 8 |
| 2.3.4 <i>Giảm thiểu thời gian phát triển và chi phí hạ tầng</i> | 9 |
| 2.4 Những nhược điểm của Single Page Application..... | 9 |
| 2.4.1 <i>Content sẽ không có độ chi tiết cao</i> | 9 |
| 2.4.2 <i>Gia tăng khối lượng công việc</i> | 10 |
| 2.4.3 <i>Không sử dụng được các kỹ thuật SEO nâng cao</i> | 10 |
| CHƯƠNG 3 – SINGLE PAGE APPLICATION SỬ DỤNG VUE | 11 |
| 3.1 Lịch sử của Vue | 11 |
| 3.2 Sơ lược về Vue..... | 11 |
| 3.2.1 <i>Vue là gì?</i> | 11 |
| 3.2.2 <i>Mô hình của Vue</i> | 12 |
| 3.2.3 <i>Ưu nhược điểm của Vue</i> | 13 |
| TÀI LIỆU THAM KHẢO..... | 14 |

DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT

CÁC KÝ HIỆU

CÁC CHỮ VIẾT TẮT

- SPA: Single Page Application.
- SSR: server Side Rendering.
- SEO: Search Engine Optimization (Tối ưu hóa công cụ tìm kiếm)

CHƯƠNG 1 – MỞ ĐẦU

1.1 Sơ lược về Single Page Application (SPA)

1.1.1 Nguyên nhân mà Single Page Application ra đời

Trước khi Single Page Application ra đời thì việc lập trình web thường tuân theo mô hình MVC (viết tắt bởi 3 từ Model - View - Controller và là một kiểu lập trình truyền thống).

Cách hoạt động của nó thường là client gửi request tới server, server xử lý request và response cho client. Những gì mà client nhận được thường là nội dung của một trang web hoàn chỉnh với HTML, CSS và JS. Tóm lại server gần như làm tất cả mọi thứ, client chỉ việc nhận kết quả từ server. Thao tác phản hồi đầy đủ này của server được gọi là SSR - Server Side Rendering.

Ngày nay khi các trang web ngày càng trở nên phức tạp và tập trung nhiều hơn vào trải nghiệm người dùng thì SSR không còn đáp ứng được do SSR là tập trung xử lý ở server, trong khi trải nghiệm người dùng thì tập trung xử lý ở client.

Nếu muốn tối ưu hóa trải nghiệm người dùng, ứng dụng hay website sẽ buộc phải giảm tải bớt công việc trên server và tập trung xử lý ở client để gia tăng tốc độ, ga tăng trải nghiệm.

Người ta bắt đầu tách biệt vai trò của client với server vì họ nghĩ cái gì thuộc về client thì nên để riêng cho client xử lý. Lúc này thì jQuery và các jQuery plugin được tận dụng triệt để, ajax được mọi người biết đến nhiều hơn. Nhưng sau cùng, các thư viện xử lý ở client giai đoạn đó (chủ yếu là jQuery) vẫn còn thô sơ. Khó đáp ứng được các xử lý phức tạp tại client.

Sau giai đoạn này, ý tưởng về một kiểu lập trình mà có thể xử lý ở client nhiều hơn ra đời, đó chính là SPA và có thể coi SPA là một bước ngoặt lớn của lập trình Web bởi SPA đẩy mạnh hơn các thao tác xử lý ở client, nâng cao vai trò của frontend và giảm

bớt công việc cho backend (có thể nói SPA làm tách bạch frontend và backend). Các frontend developer lúc này cần quan tâm nhiều hơn tới các design pattern, cấu trúc dự án, cấu trúc dữ liệu thay vì chỉ quan tâm đến UI và UX.

1.1.2 Single Page Application là gì?

Single Page Application hay Single Page Web Application hay còn gọi tắt là SPA là tên gọi chung cho một kiểu lập trình ứng dụng web rất thân thiện với thiết bị di động mà ở đó tất cả các thao tác của người dùng chỉ diễn ra trên một trang duy nhất, tất cả cấu trúc của trang web (HTML) được load sẵn 1 lần và sẽ không load lại ngay cả khi chuyển trang.

Trang web khi lập trình theo kiểu SPA thường sẽ đem lại trải nghiệm mượt mà, khiến người dùng có cảm giác như đang sử dụng một ứng dụng Mobile chứ không phải một trang web, việc code một trang web theo kiểu SPA khá giống với việc code một ứng dụng Mobile.

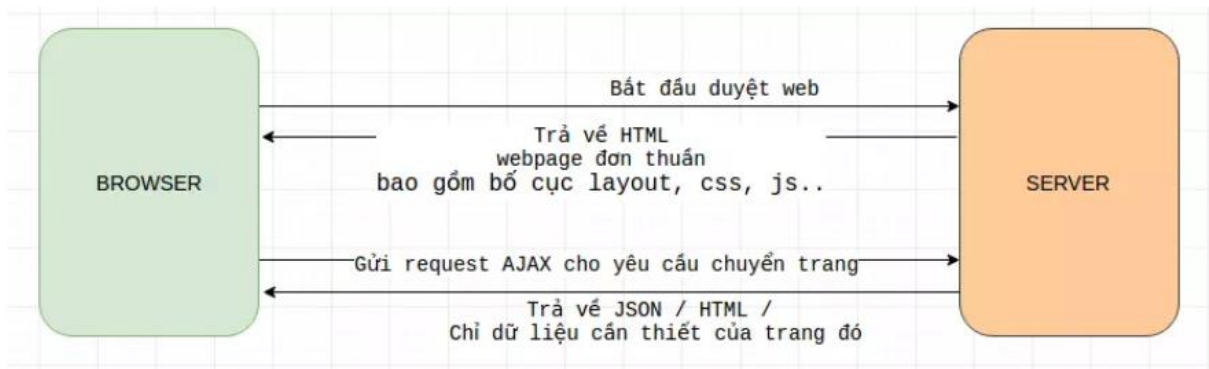
Chúng ta thường sử dụng các website code theo kiểu SPA hằng ngày mà có lẽ không nhận ra đó là các website lớn như facebook, google, youtube, Gmail,... Ta có thể dễ dàng nhận ra được các trang web kể trên đều mang lại trải nghiệm mượt mà cho người dùng mà nổi bật nhất ở đây đó chính là không bị khựng lại ở thao tác chuyển từ trang này đến trang khác.

Với Single Page Application người dùng sẽ có trải nghiệm tối ưu nhất, tận hưởng môi trường một cách tự nhiên chỉ việc “lướt lướt” trên 1 trang duy nhất mà không cần phải load lại trang.

CHƯƠNG 2 – TỔNG QUAN VỀ SINGLE PAGE APPLICATION

2.1 Cách thức hoạt động của Single Page Application (SPA)

Single page application là một ứng dụng web giúp nâng cao trải nghiệm của người dùng bằng cách xử dụng HTML5 và AJAX.



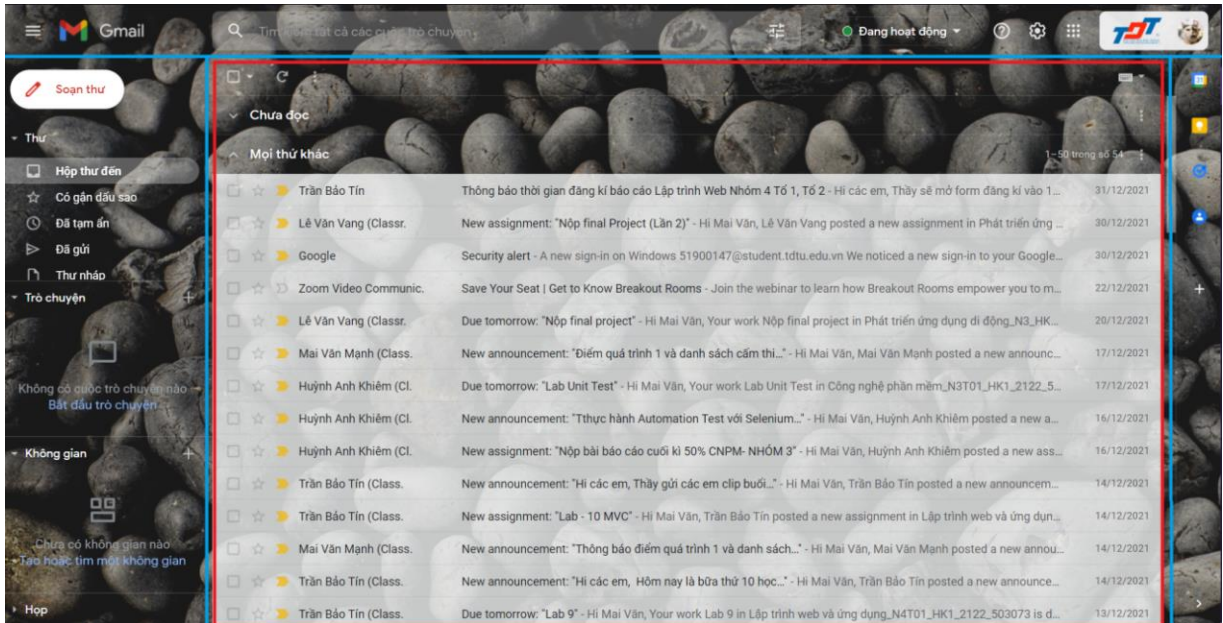
Hình 1. Sơ đồ cách thức hoạt động của SPA

Đầu tiên khi tải một trang web bất kì, SPA sẽ tải một trang HTML đơn, sau đó dựa trên request của người dùng mà SPA sẽ tiếp tục tải các HTML khác tương ứng với request của người dùng trong cùng một trang đó. SPA có thể sử dụng một số thư viện như JavaScript, AngularJS, VueJS,...

Ta có thể hiểu, toàn bộ resource của web bao gồm các file CSS, JavaScript, master layout hay cấu trúc của web page đều sẽ được load ngay trong lần đầu tiên khi chúng ta bắt đầu duyệt website. Trong những lần sau, khi chuyển sang một trang khác thì lúc đó client sẽ gửi những AJAX request để lấy những dữ liệu cần thiết.

Việc này sẽ mang đến trải nghiệm tốt hơn cho người dùng web, giảm bớt thời gian chờ đợi khi phải load lại trang web cũng như tiết kiệm băng thông. Việc này khác hoàn toàn so với một trang web thông thường khi toàn bộ trang web phải load lại mỗi khi chuyển trang.

Để dễ hiểu thì ta có thể xem một ví dụ minh họa dưới đây:



Hình 2. Minh họa về một trang web SPA

Gmail là một website lớn có sử dụng SPA. Ở đây phần có màu xanh đỏ chính là phần khung hay master page của website Gmail, phần này sẽ không bị thay thế mỗi khi chuyển trang. Phần khung màu đỏ là phần nội dung của website Gmail và sẽ hiển thị nội dung thay thế mỗi khi chuyển trang.

Có thể thấy, SPA luôn chỉ có một trang để tiếp nhận request và thực hiện routing, do đó SPA xử lý tất cả tại một trang nhanh chóng mà không bị quá tải.

2.2 Những khác biệt trong hoạt động của Single Page Application và Web truyền thống

Có hai đặc điểm lớn ảnh hưởng tạo nên sự khác biệt về cách hoạt động của web SPA so với web truyền thống đó là web SPA có backend và frontend rõ ràng và web SPA đẩy mạnh xử lý ở frontend hơn.

2.2.1 . Web Single Page Application có backend và frontend rõ ràng

So với kiểu lập trình truyền thống thì SPA có backend và frontend tách bạch tới mức chúng có thể nằm ở hai dự án khác nhau.

So với website code theo kiểu truyền thống chỉ cần một dự án thì website code theo kiểu SPA có thể cần đến hai dự án tách bạch bao gồm một cho backend và một cho frontend.

Việc trao đổi dữ liệu giữa backend và frontend thường qua các Restful API, định dạng dữ liệu thường là JSON.

2.2.2 . Web Single Page Application đẩy mạnh xử lý ở frontend

Với web SPA, frontend đảm nhiệm hoàn toàn việc render giao diện và xử lý các thay đổi giao diện, trong khi đó, backend chỉ làm theo yêu cầu là trả dữ liệu cần thiết cho frontend.

Frontend của web SPA thường là một framework hoặc một thư viện về SPA. Những framework hay thư viện này được code bằng JavaScript, và giúp cho Developer dễ dàng phát triển SPA hơn.

Để xử lý các yêu cầu phức tạp ở frontend, các framework SPA cũng hoạt động theo một kiến trúc cụ thể, có thể là MVC (Angular), MV - VM (VueJS, ReactJS),... Nhìn chung, code SPA bằng các framework chuyên về SPA sẽ đơn giản hơn nhiều so với việc code bằng jQuery hoặc bằng các kiến trúc khác thường thấy.

2.3 Những ưu điểm của Single Page Application

2.3.1 Hạn chế truy vấn đến Server

Bởi vì Server không cần phải tốn thời gian và hiệu năng để vẽ lên toàn bộ trang, SPA giảm tải sự ảnh hưởng trên toàn Server - đồng nghĩa với việc bạn có thể tiết kiệm chi phí dùng ít Server hơn cho cùng một lưu lượng yêu cầu.

2.3.2 Tăng trải nghiệm người dùng

Ngày càng nhiều tính năng được xây dựng như các dịch vụ mang tính module (kiến trúc microservice) có thể được cập nhật một cách độc lập, nó trở nên dễ dàng hơn để thử nghiệm chúng được hiển thị và sử dụng như thế nào.

Những SPA framework rất tuyệt vời khi sử dụng cùng với các dịch vụ này để tạo ra tính hấp dẫn, linh động và thậm chí phức tạp được cả trải nghiệm của người dùng.

Mặc dù rất nhiều người chỉ muốn phát triển trên một ngôn ngữ lập trình nhất định (rất nhiều SPA framework sử dụng JavaScript) nhưng nhờ có các API, SPA mà ta dựng trên một ngôn ngữ có thể làm việc tốt trên các dịch vụ từ backend được phát triển bằng nhiều ngôn ngữ lập trình khác nhau.

2.3.3 Xây dựng frontend nhanh và có tính tương thích

Cùng với thời gian xử lý nhanh hơn, SPA còn giúp cho nhà phát triển xây dựng hệ thống frontend nhanh chóng hơn.

Điều này dựa trên cấu trúc tách rời của SPA hoặc sự tách biệt giữa các dịch vụ backend và hiển thị ở frontend.

Với việc tách bạch giữa phần dữ liệu (backend) với cách chúng được trình bày (frontend) ta đã biến chúng thành một “dịch vụ” mà từ đó các nhà phát triển có thể xây dựng nhiều kiểu frontend cho việc trình bày và sử dụng dịch vụ đó.

Nhờ vào tách biệt hai thành phần này, các nhà phát triển có thể xây dựng, triển khai, trải nghiệm frontend một cách độc lập với nền tảng công nghệ của backend.

Lập trình viên có thể thiết kế sao cho người dùng có thể thấy và trải nghiệm theo những gì họ muốn trước, việc còn lại là kéo nội dung, dữ liệu và tính năng từ các dịch vụ (backend) về.

Việc này được hoàn thành bởi việc sử dụng các API, đây là những chuẩn mực được đặt ra giữa các ứng dụng nhằm thể hiện việc họ sẽ cấu trúc, trao đổi, và kết nối dữ liệu lại như thế nào.

Sự cài đặt API này giúp cho nhà phát triển làm việc nhanh hơn trên giao diện người dùng và không gây rủi ro tới nghiệp vụ quan trọng trong các công nghệ backend.

2.3.4 Giảm thiểu thời gian phát triển và chi phí hạ tầng

Các framework phổ biến như Angular, React, Vue,... là những framework thường được lựa chọn để tạo ra các SPA.

Ngoài việc chứa kiến trúc để xây dựng website theo vòng đời SPA, các framework này còn cung cấp một bộ các component thiết yếu được tích hợp sẵn và các component của những nhà phát triển khác đã đóng góp trên cộng đồng. Nếu bạn nghĩ về nó giống như xây một ngôi nhà, bạn có thể trộn đất sét, làm khô gạch, khai thác và tạo khuôn thép cho chính mình - hoặc bạn có thể sử dụng gạch và đường ống mà người khác đã thiết kế và tập trung thời gian vào thứ tạo nên ngôi nhà của bạn độc nhất. Nhờ vào tính kế thừa này mà các dự án SPA có thể giảm thời gian phát triển, cũng như chi phí hạ tầng và có khả năng nâng cấp dài hạn.

2.4 Những nhược điểm của Single Page Application

2.4.1 Content sẽ không có độ chi tiết cao

Một trong những điểm bất lợi của một Website sử dụng SPA đó là content sẽ không thể nào cụ thể và chi tiết được như Website thông thường có nhiều trang.

Ví dụ điển hình nhất chính là nội dung trên Facebook. Ta có thể thấy rằng, các bài viết trên Facebook chỉ hỗ trợ hiển thị các nội dung giới hạn của một bài viết, nội dung hình ảnh, nội dung chữ thông thường. Nếu ta muốn có các ký tự đặc biệt hay chia heading thì sẽ cần phải sử dụng bên thứ 3 để can thiệp. Việc lấp đầy content của Multiple pages trên Single page là không thể. Điều mà bạn có thể làm là chia nhỏ các mục của content ra. Mỗi mục này sẽ tương ứng với một trang riêng biệt nếu bạn làm Website có nhiều trang.

2.4.2 Gia tăng khối lượng công việc

Như đã đề cập ở trên thì một dự án website hướng đến SPA sẽ phải tách ra thành hai dự án backend và frontend riêng biệt. Điều này sẽ khiến các doanh nghiệp nhỏ với ít nhân lực trở nên “bất lực” khi khối lượng công việc tăng gấp đôi so với bình thường.

Vì vậy trong trường hợp này vẫn nên sử dụng mô hình truyền thống.

Việc sử dụng SPA sẽ giúp tăng tốc độ trải nghiệm của người dùng nhưng mặt khác sẽ phải đánh đổi bằng kinh nghiệm, công sức cùng rất nhiều tiền bạc nếu muốn một website/ ứng dụng theo mô hình Single Page Application thành công.

2.4.3 Không sử dụng được các kỹ thuật SEO nâng cao

Có những kỹ thuật SEO nâng cao mà chắc chắn chúng ta không thể sử dụng được trên single page. Một trong các kỹ thuật đó là kỹ thuật cấu trúc website thành các Category và Sub-Category để hiển thị nội dung tốt nhất cho người dùng và giúp website của chúng ta được chia theo độ tin cậy. Kỹ thuật này được gọi là siloing, một cấu trúc silo càng chứa nhiều nội dung liên quan đến chủ đề thì càng tăng độ liên quan của website trong mắt Google. Chắc chắn với single page, ta sẽ không thể nào áp dụng được kỹ thuật siloing này. Đây cũng có xem là một bất lợi của website single page.

CHƯƠNG 3 – SINGLE PAGE APPLICATION SỬ DỤNG VUE

3.1 Lịch sử của Vue

Vue được tạo bởi Evan You sau khi làm việc ở Google, khi đó Evan đang dùng AngularJS cho một số dự án thì đã có phát biểu rằng: “Điều gì sẽ xảy ra như trích một phần mà mình thích về Angular rồi xây dựng thêm một cái gì đó nhẹ nhàng hơn mà không cần phải chú ý đến những khái niệm bổ sung” và đây là phát biểu được xem là bước đệm giúp cho Vue đã và đang có mặt tại thời điểm hiện tại. Vue ban đầu được phát hành lần đầu vào tháng 2 năm 2014. Dự án này đã được đăng lên HackerNew, Echo Js trong ngày đầu ra mắt.

Hiện tại số star trên Github cho dự án của Vue đang ngày càng tăng nhanh. VueJS là một trong những project phổ biến nhất trên Github và thứ 2 trong số các JavaScript Framework (chỉ đứng sau React), Vue đã vượt qua các thư viện/ framework nổi tiếng khác như Angular 1.x, JQuery, Backbonejs,...

3.2 Sơ lược về Vue

3.2.1 Vue là gì?

Vue là một framework Javascript tiên tiến trong xây dựng giao diện người dùng, không giống như các framework khác, Vue được xây dựng từ những dòng code cơ bản nhất nhằm tối ưu tốc độ. Thư viện của Vue chỉ tập trung vào lớp hiển thị, rất đơn giản để tiếp cận và dễ dàng tích hợp vào các hệ thống khác. Vue cũng có khả năng cung cấp các ứng dụng web đơn trang Single Page Application (SPA) cho phép kết hợp với nhiều các công cụ hiện đại, như Laravel chẳng hạn.

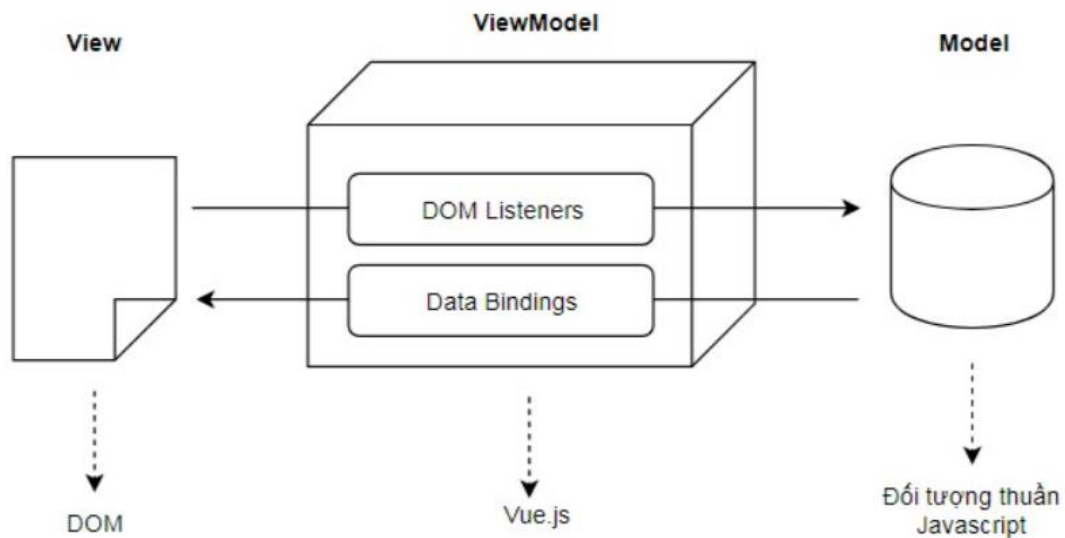
Là một công nghệ ra đời sau React và Angular vì vậy nó sẽ kế thừa lại một số tính năng nổi bật của Angular1 và đồng thời VueJs cũng có một số khái niệm khá giống với React.

Vue.js được sử dụng để xây dựng giao diện người dùng giống như React (sử dụng bởi Facebook), Angular (được hậu thuẫn bởi Google), Ember... Tuy nhiên, Vue.js có tốc độ tạo trang (render) rất nhanh và chiếm khá ít bộ nhớ.

Vue.js mới chỉ ra mắt năm 2015, nhưng Vue.js đã sớm khẳng định mình và sớm trở thành người thay thế Angular và React, đây cũng chính là lý do Laravel giới thiệu Vue.js trong thiết lập mặc định.

3.2.2 Mô hình của Vue

MVVM là viết tắt của Model-View-ViewModel là một mô hình được áp dụng trong framework Vue.js.



Hình 3. Mô hình Model-View-ViewModel

Trong mô hình này, dữ liệu mỗi khi được "gán" vào View hoặc Model sẽ đều được Vue.js tự động gắn cho phần còn lại. Tức là khi dữ liệu thay đổi ở Model nó sẽ tự động được "cập nhật" sang View và khi người dùng thay đổi dữ liệu trên View (ví dụ

nhập liệu vào ô địa chỉ email chẳng hạn) thì dữ liệu cũng được tự động cập nhật ngược lại Model. Trong cộng đồng Vue.js thường gọi mô hình này với một thuật ngữ khác là two-way data binding, tạm gọi là gán dữ liệu hai chiều.

3.2.3 Ưu nhược điểm của Vue

Vuejs là một JS Framework, mọi hoạt động của Vuejs đều được Render xử lý bằng JavaScript, Chính vì thế điều này làm cho website của chúng ta trở nên nhẹ hơn, và tốc độ cũng trở nên nhanh hơn.

Về phần nhược điểm đó là liên quan đến SEO. Vuejs có hỗ trợ Server Side Rendering hay còn gọi là SSR, tuy nhiên việc để Render được HTML từ Vuejs phía server ta cần tuân thủ sử dụng thuần Vuejs và các component của bên thứ 3 nhúng vào cũng phải tuân thủ quy định về SSR. Mà không phải đơn vị nào cũng hỗ trợ điều này ở thời điểm hiện tại. Vì vậy nên SEO trên Vuejs trở nên khó khăn hơn.

TÀI LIỆU THAM KHẢO

1. <https://vi.wikipedia.org/wiki/Vue.js>
2. <https://viblo.asia/p/gioi-thieu-framework-vuejs-gDVK2m0X5Lj>
3. <https://www.stdio.vn/javascript/spa-single-page-application-la-gi-3jnH51>
4. <https://itnavi.com.vn/blog/vuejs-la-gi-nhung-kien-thuc-co-ban-ve-vuejs>
5. <https://hethongbokhoe.com/single-page-application-la-gi-single-page-application-concept/>