

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



**TIỂU LUẬN**  
**MÔN LẬP TRÌNH WEB NÂNG CAO**

# **ECOMMERCE WEBSITE**

*Người hướng dẫn:* **ThS. NGUYỄN THANH QUÂN**

*Người thực hiện:* **Nguyễn Võ Hoàng Vũ - 51900286**

**Nguyễn Trường Anh - 51900699**

**Khoá : 23**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022**

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



**TIỂU LUẬN**  
**MÔN LẬP TRÌNH WEB NÂNG CAO**

# **ECOMMERCE WEBSITE**

*Người hướng dẫn:* **ThS. NGUYỄN THANH QUÂN**

*Người thực hiện:* **Nguyễn Võ Hoàng Vũ - 51900286**

**Nguyễn Trường Anh - 51900699**

**Khoá : 23**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022**

## LỜI CẢM ƠN

Trước tiên, nhóm nghiên cứu xin gửi lời cảm ơn chân thành và lòng biết ơn sâu sắc đến ThS.Nguyễn Thanh Quân. 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 *“eCommerce Website”*.

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 thực hành tại trường Đại học Tôn Đức Thắng nói chung.

Cuối cùng, sau khoảng thời gian học tập trên lớp 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**

Chúng tôi xin cam đoan đây là sản phẩm đồ án của riêng chúng tôi và được sự hướng dẫn của ThS.Nguyễn Thanh Quân. 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 chúng 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 chúng tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày tháng năm 2022*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

Nguyễn Trường Anh

Nguyễn Võ Hoàng Vũ

## 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

Hiện nay, công nghệ thông tin ngày càng phát triển và đã trở thành một phần quan trọng trong cuộc sống của con người. Đặc biệt là trong những năm gần đây, với tình hình dịch bệnh phức tạp thì việc áp dụng công nghệ thông tin vào việc kinh doanh, buôn bán trực tuyến đã trở thành một phần tất yếu và quan trọng để duy trì doanh thu cũng như để phát triển kinh tế cho các doanh nghiệp và đất nước.

Tuy nhiên, để có thể thực hiện kinh doanh trực tuyến thì các doanh nghiệp cần phải có một môi trường để có thể trao đổi, buôn bán với khách hàng; cũng như tạo điều kiện thuận lợi cho khách hàng có thể tham khảo và tìm kiếm sản phẩm mà doanh nghiệp kinh doanh. Từ đó, nhu cầu về các website thương mại điện tử xuất hiện ngày càng nhiều và đi theo đó là các yêu cầu về sự thuận tiện, đẹp, nhanh chóng từ phía các doanh nghiệp; nhằm tăng thêm sức hút cho các sản phẩm và gây ấn tượng với khách hàng của họ.

Nắm bắt được nhu cầu đó, nhóm đã quyết định nghiên cứu và thực hiện đề tài “eCommerce Website” với sản phẩm được bán là các quần áo và phụ kiện phương Tây. Website mà nhóm thực hiện sẽ thuộc dạng là Single Page Application được xây dựng dựa trên Framework Reactjs; kết hợp với Nodejs và Expressjs cho phía Server.

# MỤC LỤC

LỜI CẢM ƠN .....	3
PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN .....	5
TÓM TẮT .....	6
MỤC LỤC.....	7
DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT .....	9
DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ.....	10
A. PHẦN ĐỀ TÀI.....	11
CHƯƠNG 1 – GIỚI THIỆU ĐỀ TÀI .....	11
1.1 ĐẶT VẤN ĐỀ .....	11
1.2 GIỚI THIỆU ĐỀ TÀI.....	11
1.3 LÝ DO CHỌN ĐỀ TÀI .....	11
CHƯƠNG 2 – CƠ SỞ LÝ THUYẾT.....	13
2.1 TỔNG QUAN VỀ SPA .....	13
2.1.1 <i>Single Page Application</i> là gì ?.....	13
2.1.2 <i>Lý do ra đời của SPA</i> .....	14
2.1.3 <i>Cách hoạt động của SPA</i> .....	16
2.1.4 <i>Sự khác biệt của SPA và MPA</i> .....	19
2.1.5 <i>Tại sao SPA trở nên xu hướng lập trình ?</i> .....	20
2.1.6 <i>Chọn SPA hay MPA ?</i> .....	20
2.2 REACTJS.....	21
2.2.1 <i>ReactJS</i> là gì ?.....	21
2.2.2 <i>Các tính năng nổi bật của ReactJS</i> .....	21
2.2.3 <i>Ưu điểm và nhược điểm của ReactJS</i> .....	22
2.3 NODEJS:.....	23
2.3.1 <i>Khái niệm Nodejs:</i> .....	23

2.3.2 Lý do nên sử dụng Nodejs:.....	23
2.3.3 Ưu và nhược điểm của Nodejs: .....	24
2.3.4 Các trường hợp nên và không nên sử dụng Nodejs: .....	25
2.4 EXPRESSJS: .....	25
2.3.1 Khái niệm Expressjs: .....	25
2.3.2 Lý do nên sử dụng Expressjs:.....	26
2.3.3 Các tính năng nổi bật của Expressjs:.....	26
2.5 MÔ HÌNH MVC: .....	27
2.5.1 Khái niệm mô hình MVC: .....	27
2.5.2 Các thành phần chính của mô hình MVC: .....	27
2.5.3 Luồng xử lý của MVC:.....	28
2.5.4 Ưu và nhược điểm của mô hình MVC: .....	28
2.5.5 Lý do nên sử dụng mô hình MVC:.....	29
CHƯƠNG 3 – TRIỂN KHAI DỰ ÁN .....	30
3.1. GIỚI THIỆU VỀ HỆ THỐNG: .....	30
3.2. CHỨC NĂNG CỦA HỆ THỐNG:.....	30
3.3. CÁC NGUỒN TÀI NGUYÊN DÙNG ĐỂ XÂY DỰNG HỆ THỐNG: .....	31
CHƯƠNG 4 – TỔNG KẾT .....	32
4.1. KẾT QUẢ ĐẠT ĐƯỢC:.....	32
4.2. NHỮNG MẶT HẠN CHẾ:.....	32
4.3. HƯỚNG PHÁT TRIỂN TRONG TƯƠNG LAI:.....	32
B. PHẦN LÝ THUYẾT .....	33
TÀI LIỆU THAM KHẢO .....	37



# **DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT**

## **CÁC CHỮ VIẾT TẮT**

SPA	Single Page Application
MPA	Multi Page Application
CSR	Client Side Rendering
SSR	Server Side Rendering
SEO	Search Engine Optimization
UX	User Experience
JS	Javascript
MVC	Model-View-Controller

# **DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ**

## **DANH MỤC HÌNH**

Hình 1 Hình ảnh giao diện và url của trang web SPA .....	13
Hình 2 Server Side Rendering (SSR) .....	15
Hình 3 Client Side Render (CSR) .....	16
Hình 4 File html nhận được từ server Facebook.....	17
Hình 5 Giao diện hiển thị mà người dùng thấy.....	17
Hình 6 Cách hoạt động của SPA.....	18
Hình 7 Survey Front-end Frameworks .....	21
Hình 8 Nodejs .....	23
Hình 9 Expressjs.....	26
Hình 10 Mô hình MVC.....	27
Hình 11 Giao diện của trang Cổng thông tin sinh viên Trường Đại học Tôn Đức Thắng .....	33

## **DANH MỤC BẢNG**

Bảng 1 So sánh giữa SPA và MPA .....	19
Bảng 2 So sánh xử lý request file giữa PHP, ASP với Nodejs .....	24

# A. PHẦN ĐỀ TÀI

## CHƯƠNG 1 – GIỚI THIỆU ĐỀ TÀI

### 1.1 Đặt vấn đề

Với sự phát triển ngày càng vượt trội của công nghệ thông tin, kết hợp với tình hình dịch bệnh những năm gần đây thì hầu hết tất cả các hoạt động buôn bán, kinh doanh hầu hết đều diễn ra trên nền tảng trực tuyến. Mà trong đó đều diễn ra thông qua các trang mạng xã hội và đặc biệt là các website thương mại điện tử. Do đó, nhu cầu đặt ra với việc xây dựng và phát triển một website thương mại điện tử hiện nay sẽ không còn dừng lại ở thiết kế đẹp và chất lượng tốt, mà còn phải yêu cầu hiệu suất nhanh cũng như mang đến một trải nghiệm mượt mà cho người sử dụng; từ đó tạo sự thoải mái và ấn tượng tốt từ khách hàng đối với doanh nghiệp. Hiệu suất và trải nghiệm tốt ở đây tức website sẽ phải tải các thông tin một cách nhanh chóng, tránh tình trạng khiến người dùng chờ lâu. Vậy câu hỏi đặt ra là làm thế nào để xây dựng một website có thể đáp ứng nhu cầu đặt ra ở trên?

### 1.2 Giới thiệu đề tài

Để giải quyết các câu hỏi đã đặt ra, nhóm đã quyết định thực hiện đề tài có nội dung về “eCommerce Website”, với giao diện sẽ được xây dựng dựa trên Framework Reactjs, kết hợp với Server sẽ được áp dụng Nodejs và Expressjs để thực hiện; đồng thời dữ liệu sẽ được nhóm lưu trữ bằng MongoDB.

### 1.3 Lý do chọn đề tài

Website thương mại điện tử đã và đang trở thành một loại hình website rất “hot” và phổ biến, được hầu hết các doanh nghiệp sử dụng để kinh doanh các sản phẩm của họ. Vì vậy, để đáp ứng cũng như bắt kịp xu hướng hiện nay, nhóm đã quyết định nghiên cứu, tìm hiểu và xây dựng một Website thương mại điện tử với chủ trương là: đẹp, nhanh và thuận tiện.

Để đáp ứng được các chủ trương đó, Website của nhóm sẽ được xây dựng là một Single Page Application dựa trên framework Reactjs với các ưu điểm nổi bật và hiện đại

hơn so với các Website truyền thống thuộc dạng Multiple Page Application. Đồng thời, nhóm sẽ sử dụng Nodejs và Expressjs cho việc xây dựng Server cung cấp thông tin cho phía giao diện; và các thông tin sẽ được lưu trữ bằng MongoDBDatabase.

Website thương mại sẽ được nhóm xây dựng dựa trên mô hình MVC (Module – View – Controller), giúp tách biệt giữa giao diện so với database và sẽ dễ dàng quản lý cũng như bổ sung code hơn.

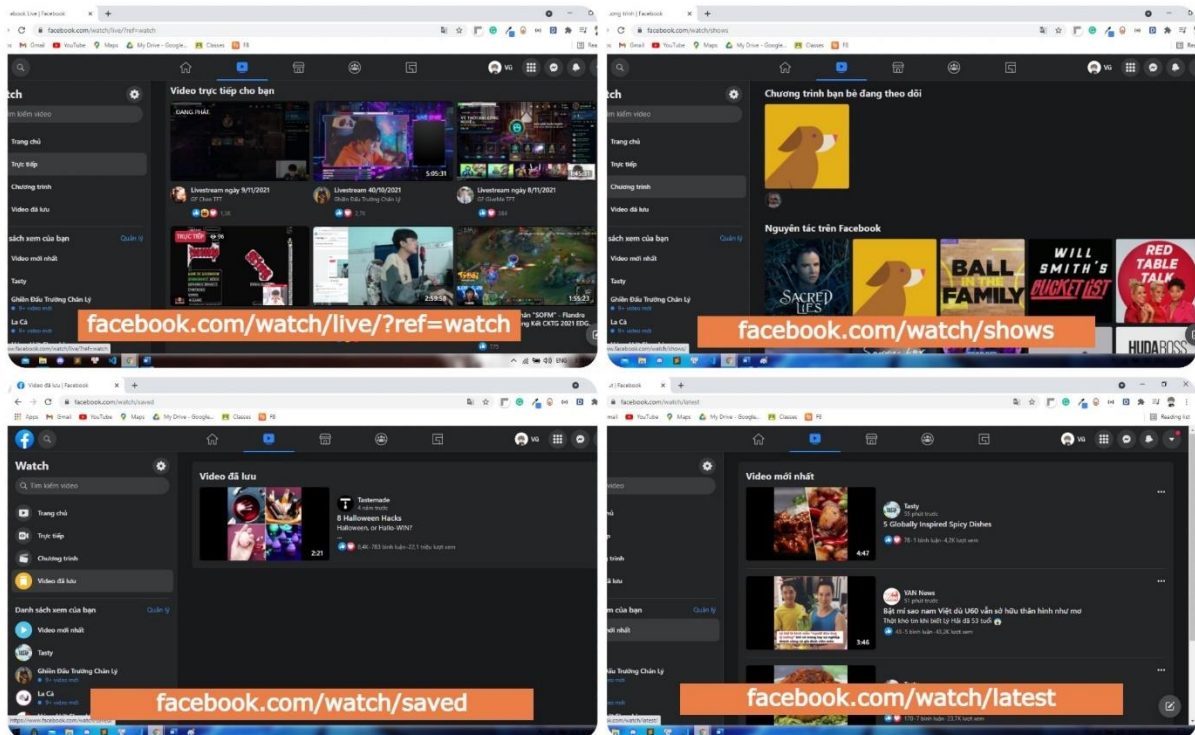
## CHƯƠNG 2 – CƠ SỞ LÝ THUYẾT

### 2.1 Tổng quan về SPA

#### 2.1.1 Single Page Application là gì ?

Single Page Application (viết tắt là SPA) là ứng dụng web một trang hay còn được gọi là ứng dụng trang đơn. Khi một trang web được xây dựng theo SPA sẽ mang lại trải nghiệm người dùng mượt mà hơn so với Multiple Page Application (viết tắt là MPA).

Khác với MPA phải tải lại toàn bộ trang web khi người dùng di chuyển qua lại giữa các trang, thì đối với SPA thì không cần phải tải lại toàn bộ trang mà chỉ tải các thành phần cần thiết (các thành phần có sự thay đổi như: nội dung chính, danh sách sản phẩm, ...), còn các thành phần không thay đổi như: header, footer, ... thì sẽ không cần thiết phải tải lại, từ đó tiết kiệm được thời gian render cho trang web. Điều đó thể hiện qua url như sau:



Hình 1. Hình ảnh giao diện và url của trang web SPA

Có thể thấy các trang trên đều xử lý tại trang ‘<https://www.facebook.com/watch>’ và mỗi tính năng nó sẽ render ra giao diện khác nhau mà không cần phải tải lại trang. Để có thể hiểu hơn về điều này ta cần phải biết tại sao SPA ra đời.

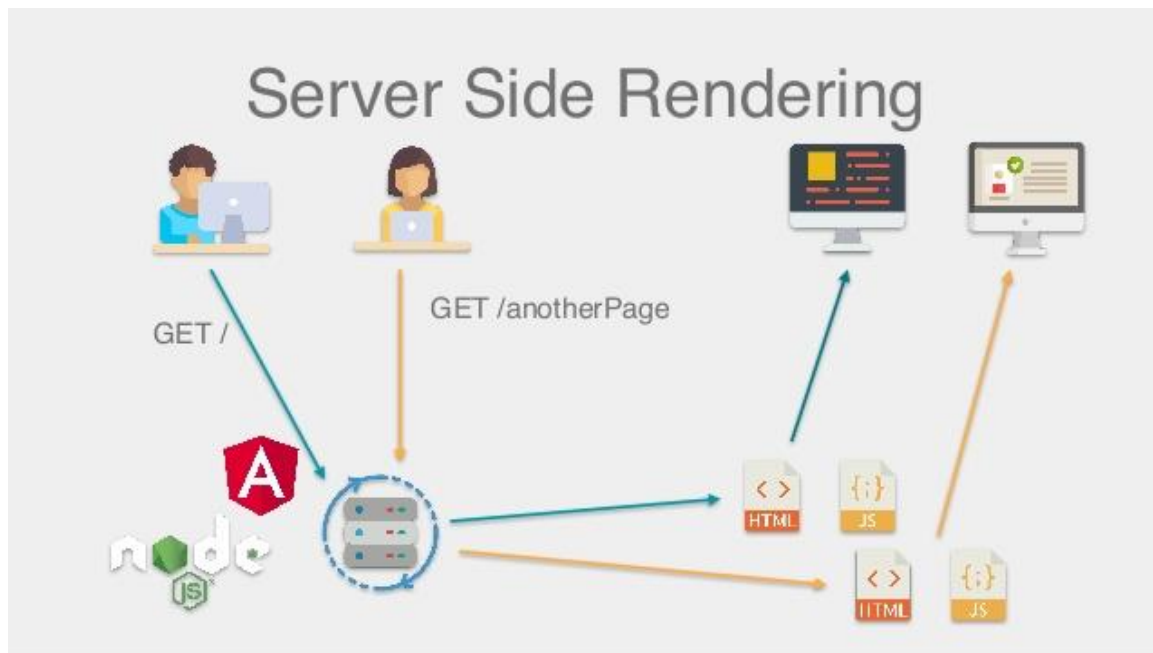
### ***2.1.2 Lý do ra đời của SPA***

Trước khi Single Page Application ra đời thì toàn bộ các webstie đều thuộc dạng là Multiple Page Application; mỗi website khi đó sẽ bao gồm rất nhiều các trang web tùy vào kích thước mà website đó được xây dựng, và mỗi trang web lại có chứa các thành phần bị lặp đi lặp lại nhiều lần như: header, footer, navbar, ....

Với các website thuộc MPA thì quá trình hoạt động sẽ diễn ra như sau:

- 1/ Người dùng chuyển qua lại giữa các trang trong website.
- 2/ Phía trình duyệt người dùng (client) sẽ gửi một request lên phía máy chủ (server) để lấy các thông tin về trang web mà người dùng truy cập.
- 3/ Máy chủ (server) sẽ nhận request từ trình duyệt người dùng (client) và phản hồi response về cho phía máy khách (client); trong đó response sẽ chứa các thông tin gồm: HTML, CSS, JS.
- 4/ Trình duyệt người dùng (client) sẽ nhận response từ máy chủ (server), sau đó tiến hành đọc dữ liệu và render ra giao diện.

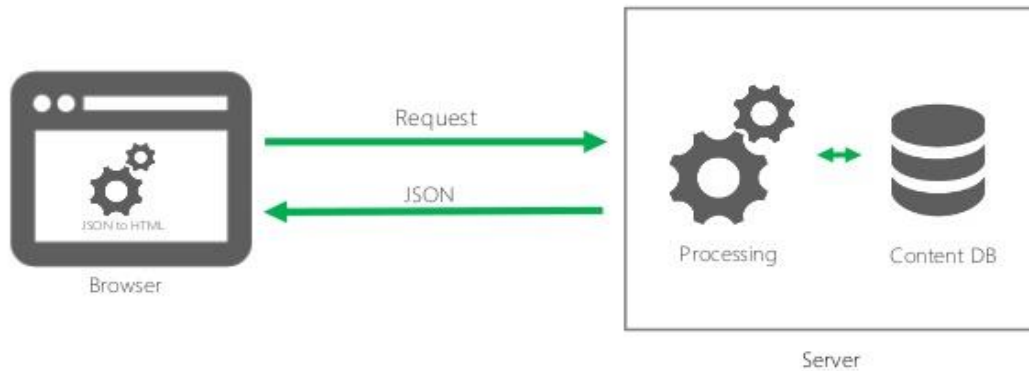
Có thể thấy mọi việc điều do phía server xử lý và trả về cho phía client, điều này còn được gọi là Server Side Rendering (SSR).



Hình 2. Server Side Rendering (SSR)

Nhưng đối với thời đại ngày nay các trang web trở nên phức tạp và yêu cầu về mặt người dùng ngày càng tăng thì SSR không đáp ứng được. Hãy tưởng tượng khi có một lượng lớn người dùng truy cập cùng lúc vào website và gửi hàng trăm, hàng ngàn request lên server có thể sẽ khiến server bị quá tải dẫn đến người dùng sẽ phải chờ phản hồi rất lâu, đôi khi là khiến cho trình duyệt hiện lỗi về trang web do quá thời gian chờ. Chính vì lý do trên nên SPA ra đời bởi nó mang đến ý tưởng hướng việc xử lý phía client nhiều hơn. Đánh dấu ra đời khái niệm mới đó là Client Side Rendering (CSR). CSR có nghĩa là phía client sẽ phụ trách việc render giao diện thay vì server sẽ xử lý tất cả.

## Client Side Render (CSR)



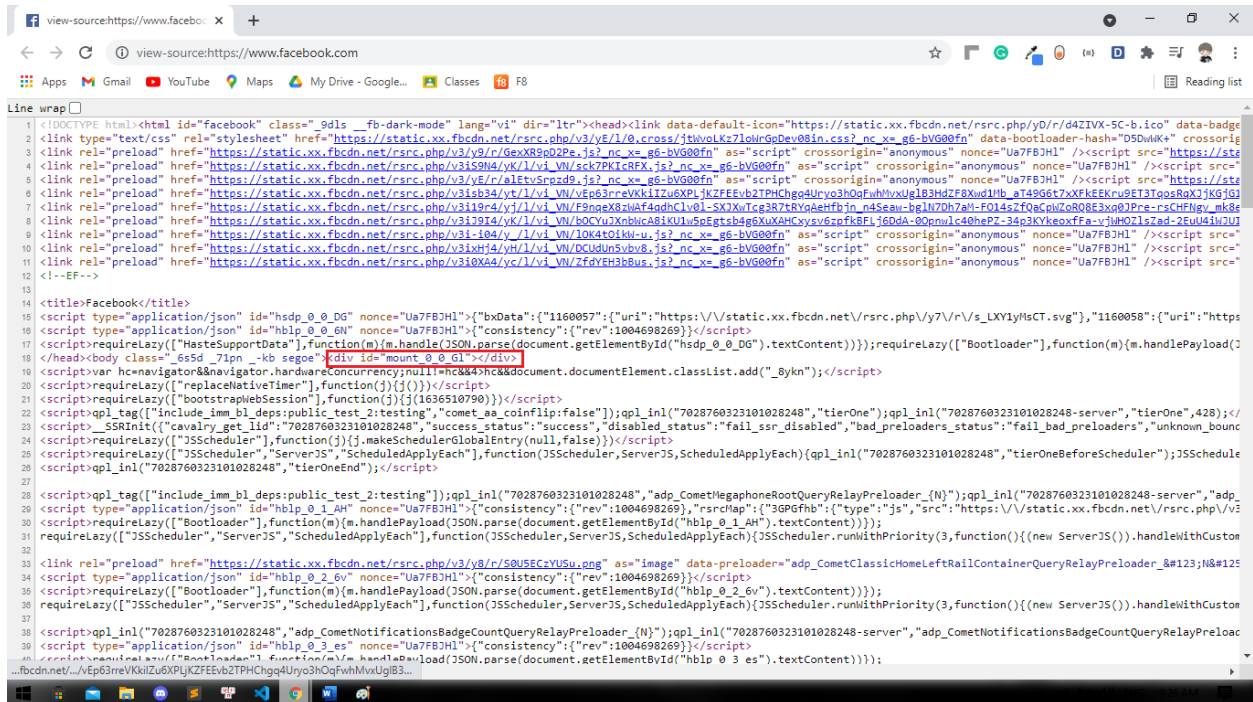
Hình 3. Client Side Render (CSR)

Do SPA đẩy mạnh xử lý ở phía client nên sẽ giảm bớt gánh nặng cho Back-end đồng nghĩa vai trò của Front-end được nâng lên. Việc này mang lại ưu điểm là tách biệt Front-end và Back-end sẽ được đề cập ở phần sau.

### ***2.1.3 Cách hoạt động của SPA***

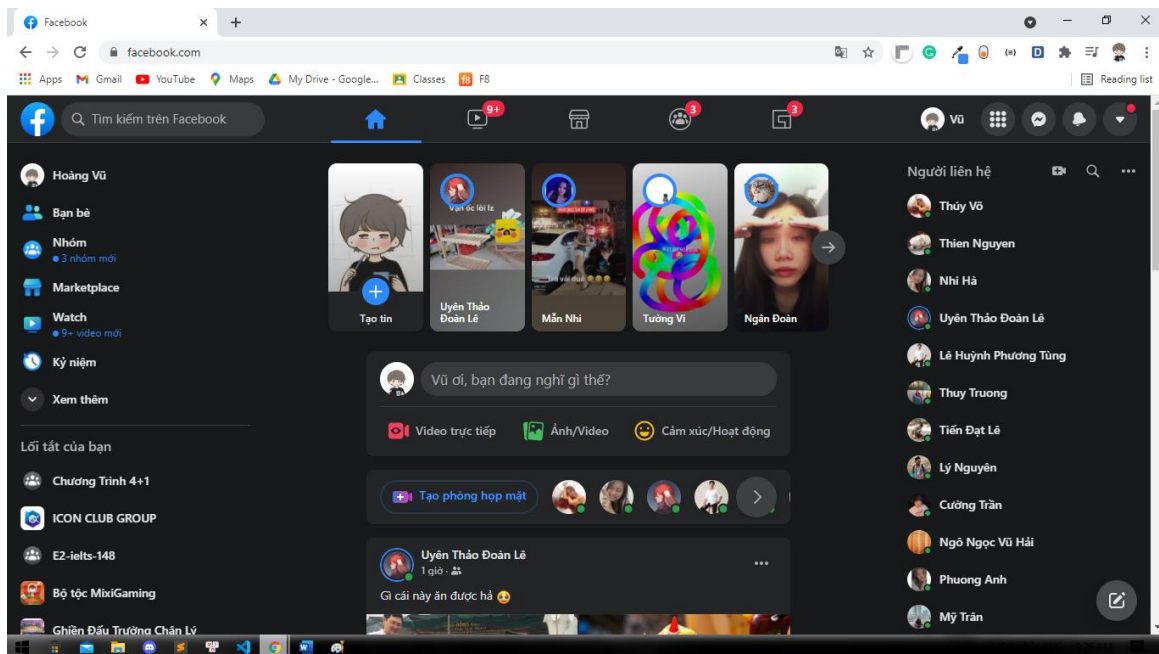
Khi xây dựng ứng dụng bằng SPA thì Back-end và Front-end sẽ được tách ra riêng biệt. Cách hoạt động SPA dựa trên việc trao đổi dữ liệu giữa Back-end và Front-end có thể được mô tả như sau: khi người dùng truy cập lần đầu tiên vào trang web, trình duyệt sẽ nhận được file html có cấu trúc như hình 4.





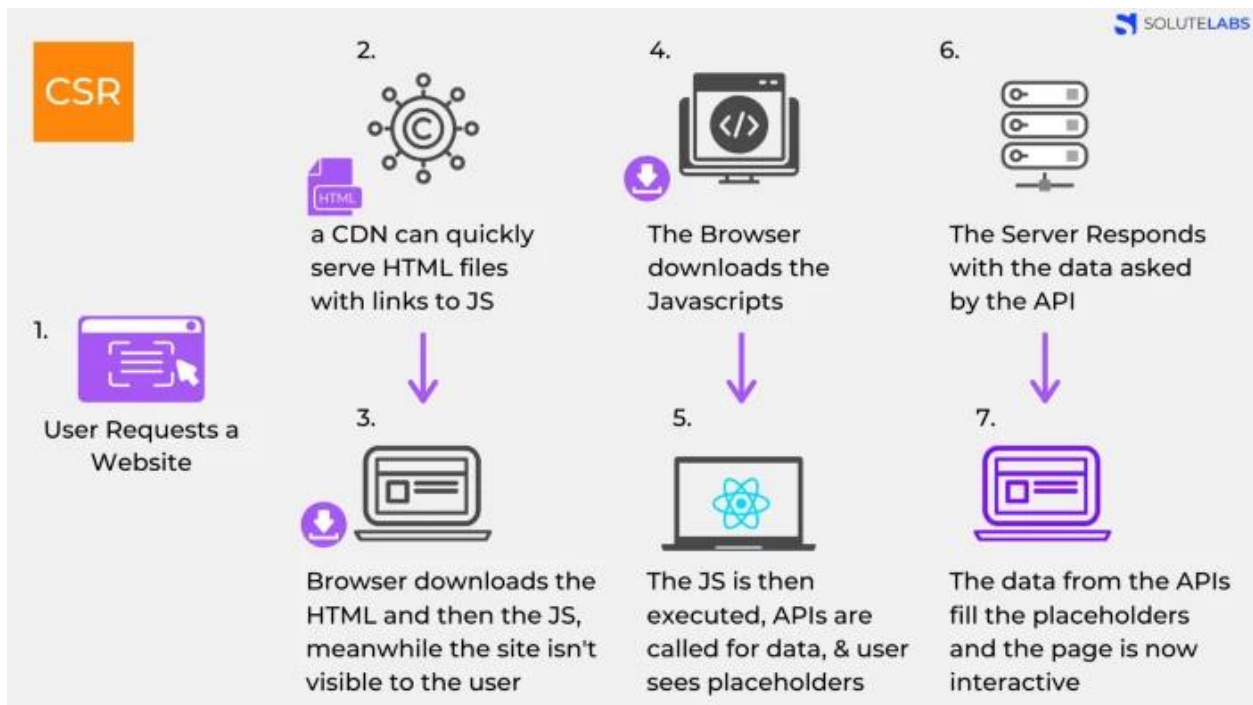
Hình 4. File html nhận được từ server Facebook

Chúng ta có thể thấy rằng là file HTML phía client nhận được trong thẻ body là một thẻ <div> rỗng nhưng tại sao giao diện người dùng thấy lại là như hình 5.



Hình 5. Giao diện hiển thị mà người dùng thấy

Vì khi nhận được file html thì trình duyệt sẽ không có gì để render cả thì lúc đó trình duyệt sẽ tải các file javascript (có thể thấy ở hình 4) và thực thi. Sau khi các file javascript được thực thi thì các DOM Element được tạo ra bởi các đoạn mã trong các file javascript và thêm vào thẻ <div> rỗng phía trên. Từ đó trình duyệt mới hiển thị ra giao diện cho người dùng.



Hình 6. Cách hoạt động của SPA

Để chứng minh cho việc trên, khi chúng ta truy cập vào trang web như là Facebook, Google thì người dùng có thể thấy trang trống trơn trong tích tắc sau đó mới thấy được giao diện. Tại vì khi nhận file html là thẻ <div> rỗng không có gì để hiển thị cùng lúc đó trình duyệt đang tải file javascript và thực thi sau đó render ra giao diện. Đó cũng là lý do tại sao mà nó được gọi là single page vì nó có đúng một trang mà có thể thấy như có nhiều trang, tất cả đều do javascript thực thi và tạo ra giao diện mới, ngay cả việc thay đổi url cũng do javascript thực hiện.

Vậy tại sao lại nói là sự giao tiếp của Front-end và Back-end ? Để giải thích thì chúng ta sẽ nói chi tiết hơn khi người dùng click vào nút chuyển hướng trên thanh điều hướng. Khi người dùng click vào nút chuyển hướng thì việc nhận request lúc này là Front-end chứ không phải Back-end. Sau khi tiếp nhận request, Front-end sẽ biết được người

dùng muốn chuyển hướng đến trang nào và trang này cần những dữ liệu gì thì mới gửi request đến Back-end. Front-end nhận được response từ Back-end, dựa vào dữ liệu nhận được Front-end lại render ra giao diện người dùng. Đó là lý do vì sao việc này được gọi là Client Side Rendering (CSR).

#### 2.1.4 Sự khác biệt của SPA và MPA

Để có thể hiểu rõ về ưu điểm và nhược điểm của SPA, chúng ta sẽ so sánh sự khác biệt của SPA và MPA thông qua bảng sau.

Bảng 1: So sánh giữa SPA và MPA

	SPA	MPA
<b>Tốc độ</b>	-Phần lớn tài nguyên được tải trong lần đầu. (*) -Trang chỉ tải thêm dữ liệu mới khi cần => Nhanh hơn khi sử dụng	-Luôn tải lại toàn bộ trang khi truy cập và chuyển hướng.  => Chậm hơn khi sử dụng
<b>Bóc tách</b>	-Có phần Front-end riêng biệt với Back-end => Có thể chia team để phát triển	-Front-end và Back-end phụ thuộc vào nhau nhiều hơn, được đặt trong cùng 1 dự án
<b>SEO</b>	-Không thân thiện với SEO như MPA (**)	- Áp dụng được kỹ thuật SEO nâng cao
<b>UX</b>	-Trải nghiệm tốt hơn, nhất là khi chuyển trang -Trải nghiệm di động tốt hơn	- Trải nghiệm không tốt như SPA do phải tải lại toàn bộ trang.
<b>Quá trình phát triển</b>	-Dễ dàng tái sử dụng code(component) -Chia team phát triển song song FE & BE -Phát triển thêm mobile app dễ dàng	- Khó triển khai để có thể tái sử dụng code như SPA - Cần viết lại API nếu muốn phát triển mobile app
<b>Phụ thuộc vào Javascript</b>	-Phụ thuộc hoàn toàn javascript (***)	-Có thể không cần javascript
<b>Performance</b>	-Giảm tải cho sever khi có nhiều người truy cập bởi vì render dữ liệu sẽ do phía client xử lý	-Khi nhiều người sử dụng sẽ gửi rất nhiều request lên sever => tốn nhiều chi phí hơn cho sever

(\*) Đây cũng có thể là nhược điểm của SPA nếu lần đầu tải quá nhiều sẽ dẫn đến tải chậm. Điều này phụ thuộc vào việc tối ưu tốt khi code bằng cách sử dụng phương pháp Code Splitting (băm nhỏ code) thì trong lần đầu tải lượng nhỏ dữ liệu thì trang web vẫn sẽ thực hiện rất nhanh. Tuy vậy, những lần render sau thì sẽ rất nhanh do chỉ cần nhận data

theo dạng json từ server gửi về để render giao diện, cải thiện rất nhiều về tốc độ so với MPA.

(\*\*) Do khi request url thì SPA chỉ nhận về page source với một thẻ <div> rỗng nó sẽ không hiểu được. Hiện nay google bot và các công cụ tìm kiếm khác cũng đang dần nâng cấp và hỗ trợ tốt hơn cho các SPA trong việc SEO. Nhưng sẽ phức tạp hơn việc xử lý bằng MPA.

(\*\*\*) Vì SPA phụ thuộc hoàn toàn vào javascript đòi hỏi đội ngũ lập trình phải nắm chắc cách sử dụng javascript và các framework hỗ trợ liên quan như ReactJS hay AngularJS.

### ***2.1.5 Tại sao SPA trở nên xu hướng lập trình ?***

Single Page Application trở nên xu hướng trong vòng 5 năm trở lại đây. Vậy tại sao nó trở nên trending ? Có hai lý do sau:

- SPA là cách tiếp cận hiện đại hơn do các lập trình viên nhận ra được các ưu điểm mà SPA mang lại, nhất là về mặt trải nghiệm người dùng.
- Các ‘ông lớn’ công nghệ như Google, Facebook, Twitter áp dụng SPA vào sản phẩm của họ và các sản phẩm đó đang được cả thế giới sử dụng. Việc này cũng thúc đẩy các thư viện, framework về Javascript phát triển như ReactJS, AngularJS.

### ***2.1.6 Chọn SPA hay MPA ?***

Tất nhiên không có gì là hoàn hảo trong mọi trường hợp nên việc chọn SPA hay MPA để phát triển ứng dụng nên phụ thuộc vào yêu cầu khách hàng:

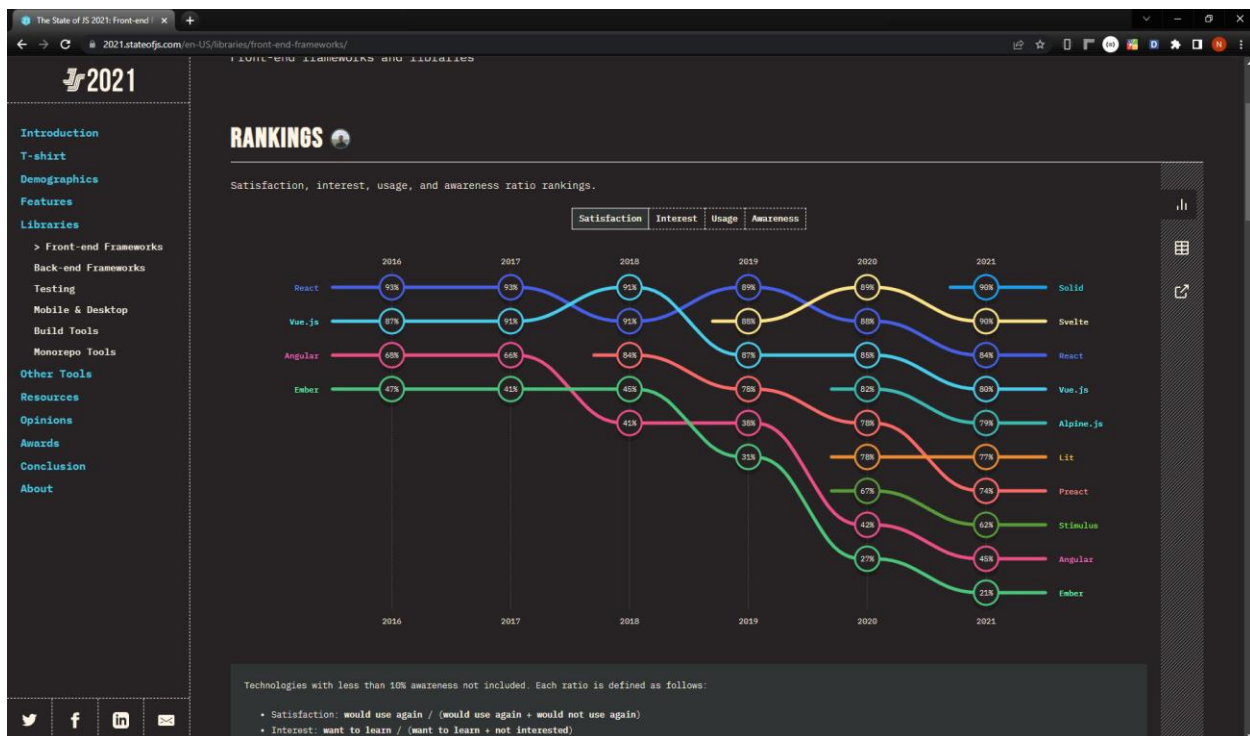
- Nếu khách hàng muốn một trang web đơn giản để giới thiệu công ty hoặc trang bán hàng đơn giản và tương lai họ cũng không có ý định phát triển ứng dụng rộng rãi, tăng quy mô lên, không cần mobile app thì chúng ta có thể phát triển theo MPA.
- Nhưng ngược lại thì khách hàng rất quan tâm tới trải nghiệm người dùng, khách hàng của họ dùng phải mượt mà, thậm chí muốn phát triển ngày càng tăng quy mô và phát triển cả mobile app, thêm các chức năng mới hiện đại trong tương lai thì chúng ta nên xây dựng theo kiểu SPA.

- Một vấn đề nữa cần được quan tâm đó là chi phí. Tất nhiên chi phí của MPA sẽ thấp hơn so với SPA vì SPA sẽ phải chia ra team Front-end và Back-end, bỏ ra nhiều chất xám hơn so với MPA.

## 2.2 ReactJS

### 2.2.1 ReactJS là gì ?

ReactJS là một thư viện Javascript mã nguồn mở, dùng để xây dựng giao diện người dùng. ReactJS được phát triển bởi ‘ông lớn’ công nghệ Facebook nhằm tạo ra một hệ sinh thái React. ReactJS đang được sử dụng rất rộng rãi với một cộng đồng lớn mạnh.



Hình 7. Survey Front-end Frameworks

Dựa vào hình 7 ta có thể thấy React được số lượng người sử dụng và yêu thích lên đến 84% vào năm 2021. Điều này chứng minh rằng React là một trong những framework đang hot hiện nay.

### 2.2.2 Các tính năng nổi bật của ReactJS

ReactJS cung cấp cho lập trình viên các tính năng nổi bật như:

- + JSX (viết tắt Javascript Extension): là một React Extension, giúp cho việc thay đổi các thành phần trên cây DOM (Document Object Module) dễ dàng hơn, linh hoạt hơn. Có thể pha trộn giữa Javascript và HTML, từ đó có thể áp dụng các logic mà Javascript cung cấp để xử lý giao diện HTML.
- + Bóc tách các Components: một trang web được xây dựng dựa trên ReactJS sẽ là sự kết hợp nhiều Component lại với nhau chứ không sử dụng chung một Template, giúp chúng ta dễ dàng tái sử dụng các Component, từ đó linh hoạt và dễ dàng hơn trong việc điều chỉnh giao diện hiển thị. Ngoài ra, các Component sẽ giúp bóc tách Javascript, tạo nên các đoạn code độc lập và giúp dễ dàng kiểm soát hơn.
- + Tạo DOM ảo: ReactJS sẽ tạo nên một môi trường DOM ảo giúp mô phỏng lại giống hoàn toàn so với DOM của trình duyệt. DOM ảo sẽ giúp tìm đúng các thành phần có sự thay đổi và cần cập nhật, trong khi các thành phần khác không có sự thay đổi sẽ giữ nguyên.

### ***2.2.3 Ưu điểm và nhược điểm của ReactJS***

Một số ưu điểm của ReactJS mang lại khi sử dụng đó là:

- Được đánh giá là thân thiện với SEO (trong khi đây là một nhược điểm của SPA).
- Khả năng mở rộng tốt, tái sử dụng cao (chia các Component).
- Hiệu suất cao.
- Phát triển nhanh chóng.
- Khả năng tương thích ngược (nâng cấp dự án cũ dễ dàng).

Một số nhược điểm của ReactJS:

- Kích thước file lớn.
- Các thành phần thường được viết bằng JSX (kết hợp giữa HTML và CSS với Javascript) nên sẽ gây chút khó khăn cho những người mới khi nhầm lẫn giữa HTML với JSX

## 2.3 Nodejs:

### 2.3.1 *Khái niệm Nodejs:*

Nodejs là một nền tảng (Platform) được tạo ra bởi Ryan Dahl và phát triển độc lập dựa trên V8 Javascript Engine của Chrome, giúp xây dựng các ứng dụng web một cách linh hoạt, nhanh chóng và có thể dễ dàng mở rộng.

Nodejs hoàn toàn miễn phí và có thể thực thi trên nhiều hệ thống khác nhau như: Window, MacOS, Linux, ...

Nodejs sử dụng Javascript sử dụng chạy trên môi trường Server.



Hình 8. Nodejs

### 2.3.2 *Lý do nên sử dụng Nodejs:*

Nodejs sử dụng ngôn ngữ Javascript, giúp lập trình viên website dễ dàng tiếp cận cũng như có một cộng đồng lớn mạnh trên toàn thế giới.

Nodejs chạy đơn luồng và sử dụng cơ chế xử lý bất đồng bộ, từ đó giúp người lập trình viên dễ dàng hơn trong việc viết code không cần phải tạo luồng hay tiến trình cho các hoạt động xử lý. Có thể được minh họa bằng việc Nodejs xử lý một request so với PHP và ASP như sau:

Bảng 2: So sánh xử lý request file giữa PHP, ASP với Nodejs

PHP và ASP	Nodejs
1/ Gửi một tác vụ đến hệ thống tệp của máy tính.	1/ Gửi một tác vụ đến hệ thống tệp của máy tính.
2/ Chờ hệ thống mở và đọc file.	2/ Sẵn sàng tiếp nhận request tiếp theo.
3/ Trả về nội dung cho phía client.	3/ Khi hệ thống mở và đọc file, hệ thống trả về nội dung cho client.
4/ Sẵn sàng cho lần request tiếp theo	

Qua đó, Nodejs sẽ không phải chờ lâu trong việc xử lý các tác vụ được gửi đến và giúp tiết kiệm bộ nhớ.

### 2.3.3 Ưu và nhược điểm của Nodejs:

Một số ưu điểm của Nodejs:

- + Cải thiện tốc độ xử lý nhờ áp dụng cơ chế bất đồng bộ, từ đó có thể tiếp nhận nhiều kết nối và xử lý trong khoảng thời gian ngắn.
- + Dễ dàng mở rộng khi cần phát triển website.
- + Sử dụng Single-thread (đơn luồng) giúp tiết kiệm bộ nhớ, kết hợp với Non-Blocking của Javascript giúp tận dụng tối đa tài nguyên của server.
- + Có thể xử lý nhiều request trong khoảng thời gian ngắn.
- + Dễ dàng xây dựng các ứng dụng Single Page Application bởi việc xử lý các request diễn ra nhanh chóng và tốn ít thời gian.
- + Phù hợp trong việc xây dựng các ứng dụng thời gian thực như: Facebook, Instagram, ...

Một số nhược điểm của Nodejs:

- + Tiêu tốn tài nguyên lớn, do Nodejs được viết bằng C# và Javascript nên cần quá trình biên dịch khi xử lý. Do đó, với những thao tác xử lý cần tiêu



tốn nhiều tài nguyên của CPU thì việc sử dụng Nodejs không phải là lựa chọn tối ưu.

+ Nodejs không có sự khác biệt và chênh lệch quá lớn so với các ngôn ngữ khác như PHP, Ruby, Python, .NET, ...

### ***2.3.4 Các trường hợp nên và không nên sử dụng Nodejs:***

Các trường hợp nên sử dụng Nodejs:

- + Xây dựng RESTful API (json). Do áp các API server thường không xử lý các tác vụ phức tạp tốn nhiều tài nguyên và do Nodejs sử dụng ngôn ngữ Javascript sẽ giúp xử lý JSON rất dễ dàng, đồng thời với Non-Blocking giúp tiếp nhận và phản hồi nhanh các request được gửi đến. Từ đó giúp Nodejs rất phù hợp trong việc viết các RESTful API.
- + Xây dựng các ứng dụng thời gian thực.

Các trường hợp không nên sử dụng Nodejs:

- + Xây dựng các ứng dụng có các xử lý tốn nhiều tài nguyên như: xử lý ảnh, chuyển đổi video, ....
- + Một ứng dụng chỉ toàn là CRUD: vì Nodejs sẽ không nhanh hơn so với PHP đối với các tác vụ mang nặng tính I/O.

## **2.4 Expressjs:**

### ***2.3.1 Khái niệm Expressjs:***

Expressjs là một framework mã nguồn mở dành cho Nodejs, cung cấp các tính năng mạnh mẽ giúp việc xây dựng các ứng dụng web hoặc mobile diễn ra dễ dàng và nhanh chóng hơn. Ngoài ra, Expressjs còn hỗ trợ các giao thức HTTP và các middleware tạo API vô cùng mạnh mẽ và dễ sử dụng.



Hình 9. Expressjs

### ***2.3.2 Lý do nên sử dụng Expressjs:***

Expressjs hỗ trợ ngôn ngữ Javascript, từ đó tạo sự thuận tiện hơn cho lập trình viên trong việc sử dụng cũng như có một cộng đồng lớn mạnh trên toàn thế giới.

Expressjs là một framework mã nguồn mở hoàn toàn miễn phí.

Dùng Expressjs kết hợp với Nodejs sẽ giúp thời gian phát triển một website giảm đi một cách đáng kể, giúp tăng hiệu quả cũng như tốc độ phát triển ứng dụng.

### ***2.3.3 Các tính năng nổi bật của Expressjs:***

Expressjs cung cấp nhiều tính năng nổi bật như:

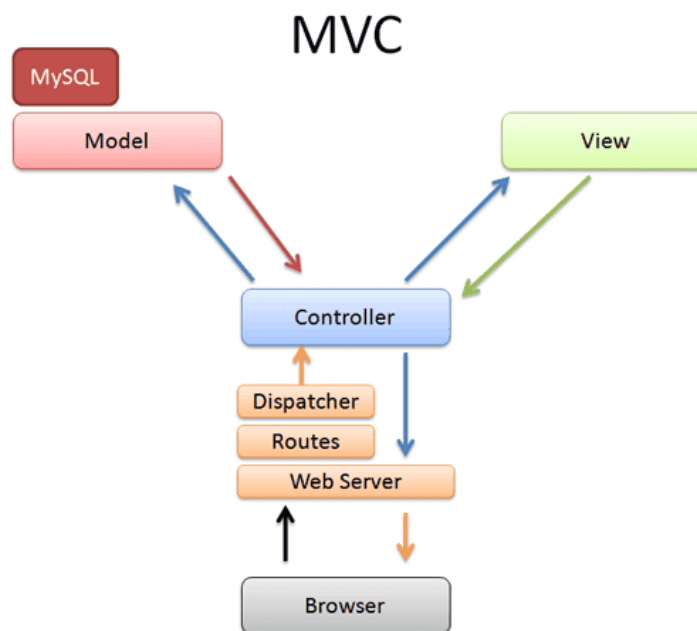
- + Giúp phát triển máy chủ một cách nhanh chóng: với việc cung cấp các tính năng dưới dạng các hàm từ đó có thể sử dụng ở nhiều nơi trong chương trình, làm giảm nhu cầu viết mã để từ đó gia tăng tốc độ xây dựng máy chủ.
- + Sử dụng các Middleware: là các phần mềm trung gian có quyền truy cập vào cơ sở dữ liệu, tiếp nhận xử lý các yêu cầu từ người dùng. Có trách nhiệm tổ chức có hệ thống các chức năng của Expressjs.
- + Định tuyến – Routing: cho phép thực hiện các hành động khác nhau dựa vào các phương thức HTTP (Get, Post, Put, Delete,...) và các đường dẫn URL.
- + Cung cấp các mẫu – Template: Expressjs cũng cấp các Template engine như: handlebars, ... cho phép lập trình viên xây dựng các website động bằng việc cung cấp các HTML ở phía máy chủ.

+ Dễ dàng gỡ lỗi: Expressjs hỗ trợ việc gỡ lỗi một cách nhanh chóng bằng việc xác định chính xác vị trí các lỗi trong các thành phần của website.

## 2.5 Mô hình MVC:

### 2.5.1 Khái niệm mô hình MVC:

MVC là từ viết tắt của cụm từ “Model – View – Controller”, đây là một mẫu kiến trúc phần mềm giúp phát triển phần mềm nhanh chóng thông qua việc chia source code thành ba phần tách biệt nhau gồm: Model (dữ liệu), View (giao diện) và Controller (điều khiển).



Hình 10. Mô hình MVC

### 2.5.2 Các thành phần chính của mô hình MVC:

Mô hình MVC bao gồm 3 thành phần chính gồm:

+ Model: là bộ phận chứa toàn bộ dữ liệu của ứng dụng (như: thông tin đơn hàng, thông tin tài khoản, sản phẩm, ...), có nhiệm vụ lưu trữ và cung cấp thông tin hiển thị phía View thông qua Controller.

- + View: là giao diện của ứng dụng, nơi người dùng sẽ thực hiện các thao tác với ứng dụng (như: nhấn xem video, nhấn vào nút đăng ký, ...).
- + Controller: là bộ phận giúp tiếp nhận và xử lý các yêu cầu từ phía người dùng gửi lên, khi họ tương tác với View, sau đó sẽ tiến hành truy cập vào Model để lấy các thông tin cần thiết và trả về cho View.

### ***2.5.3 Luồng xử lý của MVC:***

Để có thể hiểu thêm việc các thành phần của mô hình MVC tương tác với nhau như thế nào thì chúng ta sẽ đi qua một luồng xử lý như sau:

- + Bạn đang ở trang tìm kiếm Google (View) và sau đó bạn gõ dòng chữ “Trường đại học Tôn Đức Thắng” và nhấn Enter.
- + Lúc này request của bạn sẽ được gửi lên phía server và được một Controller tiếp nhận và tiến hành xử lý request đó.
- + Dựa vào dữ liệu gửi lên là “Trường đại học Tôn Đức Thắng” thì Controller sẽ tiến hành truy cập vào cơ sở dữ liệu (Model) để tìm kiếm các thông tin liên quan đến request của bạn và trả về cho phía Controller.
- + Vì lúc này bạn đang tìm kiếm kết quả của “Trường đại học Tôn Đức Thắng” nên Controller sẽ trả về cho trình duyệt của bạn một View mới cùng với dữ liệu nhận được từ Model, lúc này bạn sẽ thấy View không còn là trang tìm kiếm mà sẽ là một danh sách các thông tin liệt kê liên quan đến từ khóa “Trường đại học Tôn Đức Thắng”.

### ***2.5.4 Ưu và nhược điểm của mô hình MVC:***

Các ưu điểm của mô hình MVC:

- + Giúp bóc tách các thành phần dự án thành các phần riêng biệt: Model, View, Controller. Giúp việc quản lý, kiểm tra, chỉnh sửa hơn và bảo trì ứng dụng diễn ra dễ dàng và nhanh chóng hơn.
- + Trình tự xử lý rõ ràng, nhiệm vụ riêng biệt và tách biệt giữa các thành phần, các thành phần có thể tương tác với nhau.

- + Có thể chia công việc cho nhiều lập trình viên thực hiện cùng lúc do các thành phần tách biệt với nhau.

Các nhược điểm của mô hình MVC:

- + Không phù hợp với các dự án nhỏ do việc triển khai sẽ khá cồng kềnh.
- + Việc kiểm thử khá khó khăn do các thành phần có liên hệ với nhau, như để kiểm thử View thì phải giả lập cả Model và Controller.
- + Khó triển khai.

#### ***2.5.5 Lý do nên sử dụng mô hình MVC:***

Đối với các dự án vừa và lớn, việc sử dụng MVC sẽ mang đến nhiều lợi ích như:

- + Tách biệt các thành phần với nhau: View, Model, Controller.
- + Giúp giảm thiểu thời gian phát triển, do các thành phần được tách biệt với nhau nên một lập trình viên có thể làm View trong khi một lập trình viên khác thực hiện phần Controller hoặc Model cho ứng dụng.
- + Việc sửa đổi trên Website sẽ không làm ảnh hưởng toàn bộ mô hình MVC vì phần View tách biệt hoàn toàn so với Model, vì vậy việc thêm một View mới tùy theo nhu cầu và xu hướng sẽ rất dễ dàng. Ngoài ra, bạn cũng có thể thực hiện thay đổi trong Model mà không phải lo ảnh hưởng toàn bộ kiến trúc của website.
- + Tạo sự thân thiện và cải thiện SEO rất tốt.

## CHƯƠNG 3 – TRIỂN KHAI DỰ ÁN

### 3.1. Giới thiệu về hệ thống:

Với tình hình dịch bệnh diễn ra trong một năm trở lại đây nên các hoạt động buôn bán, kinh doanh hầu hết diễn ra trên nền tảng trực tuyến. Từ đó khiến cho các hoạt động mua sắm của mọi người trên các website thương mại điện tử hiện nay rất phổ biến, bên cạnh đó các hệ thống cửa hàng xây dựng website bán hàng riêng cho mình để tiếp cận tới nhiều khách hàng hơn.

Nắm bắt được tình hình đó, nhóm chúng tôi quyết định sẽ tạo ra một trang web với tên gọi là “Anh Vũ Store”, một website chuyên cung cấp các bộ đồ vest sang trọng, phục vụ việc mua sắm online tại nhà của nhiều khách hàng.

“Anh Vũ Store” thuộc loại Single Page Application (SPA) và được viết dựa trên framework ReactJs. Mang đến các trải nghiệm về UI/UX mượt mà, thú vị. “AV Store” sẽ có trang website cho phía khách hàng và cả phía người quản lý.

### 3.2. Chức năng của hệ thống:

Trang web “Anh Vũ Store” phía khách hàng cung cấp cho người dùng các tính năng cơ bản nhất mà một website thương mại điện tử thường có, bao gồm:

- + Xem thông tin về cửa hàng và sản phẩm: Khi người dùng truy cập website AV Store xem thông tin và xem các sản phẩm hiện có bằng cách ấn vào sản phẩm.
- + Đăng ký tài khoản: Cung cấp chức năng tạo tài khoản, giúp người dùng có được tài khoản riêng trong hệ thống và thực hiện các chức năng mà một tài khoản mang lại.
- + Đăng nhập / Đăng xuất: Đăng nhập và đăng xuất trong hệ thống khi đã có tài khoản.
- + Thêm sản phẩm vào giỏ hàng: Giúp người dùng lưu lại các sản phẩm cần mua, thuận lợi hơn cho việc tìm kiếm sau này khi chỉ cần vào giỏ hàng.
- + Đặt hàng: Người dùng có thể đặt những sản phẩm đã thêm vào giỏ hàng.

- + **Chỉnh sửa thông tin cá nhân:** Người dùng có thể chỉnh sửa thông tin cá nhân của mình.
- + **Xem danh sách đơn hàng:** Người dùng có thể xem danh sách đơn hàng mà mình đã đặt và tình trạng của các đơn hàng.
- + **Đổi mật khẩu:** Người dùng có thể đổi mật khẩu mới cho tài khoản của mình.

Đối với website quản lý sẽ có các chức năng như sau:

- + **Đăng nhập / Đăng xuất:** Đăng nhập và đăng xuất trong hệ thống.
- + **Thêm, sửa, xóa sản phẩm:** Người quản lý có thể thêm sản phẩm, chỉnh sửa thông tin sản phẩm, xóa sản phẩm.
- + **Duyệt đơn hàng:** Duyệt các đơn hàng được đặt bởi khách hàng.
- + **Lọc sản phẩm theo loại:** Chức năng lọc các sản phẩm theo loại giúp thuận lợi hơn cho việc tìm kiếm, quản lý.
- + **Lọc đơn hàng theo tình trạng:** Chức năng lọc đơn hàng theo tình trạng giúp thuận lợi hơn cho việc tìm kiếm.

### **3.3. Các nguồn tài nguyên dùng để xây dựng hệ thống:**

“Anh Vũ Store” sử dụng framework chính là ReactJS, hỗ trợ việc tạo nên một trang web Single Page Application (SPA) một cách nhanh chóng, thuận lợi và dễ dàng.

REST Api sử dụng cho trang website sử dụng framework ExpressJS kết với với MongoDB.

Các công cụ hỗ trợ xây dựng dự án: ReactJS, ExpressJS, Git, Visual Studio Code, Postman.

## CHƯƠNG 4 – TỔNG KẾT

### 4.1. Kết quả đạt được:

Hiểu về trang web Single Page Application và ứng dụng được Framework React kết hợp REST Api dựa trên Framework Express để xây dựng thành một trang web hoàn chỉnh.

Tạo ra được trang web bán hàng với tên gọi là “Anh Vũ Store” và trang web dành cho phía quản lý cửa hàng với các chức năng đã được liệt kê ở trên.

Trang web hoạt động mượt mà, mang đến trải nghiệm thoải mái, dễ chịu cho người sử dụng, không phát sinh lỗi trong quá trình sử dụng.

### 4.2. Những mặt hạn chế:

Chưa có các chức năng nâng cao như: Đánh giá sản phẩm, Quản lý tài khoản khách hàng, Đăng ký bằng tài khoản Google hoặc Facebook, Quên mật khẩu.

Giao diện thiết kế chưa đẹp và chuyên nghiệp.

### 4.3. Hướng phát triển trong tương lai:

Thêm các chức năng nâng cao còn thiếu.

Thiết kế lại giao diện để bắt mắt và chuyên nghiệp hơn.

Hỗ trợ người dùng nhiều hơn với việc gợi ý các sản phẩm kèm theo để gợi ý cho khách hàng.

Tìm hiểu thêm về SEO và triển khai.



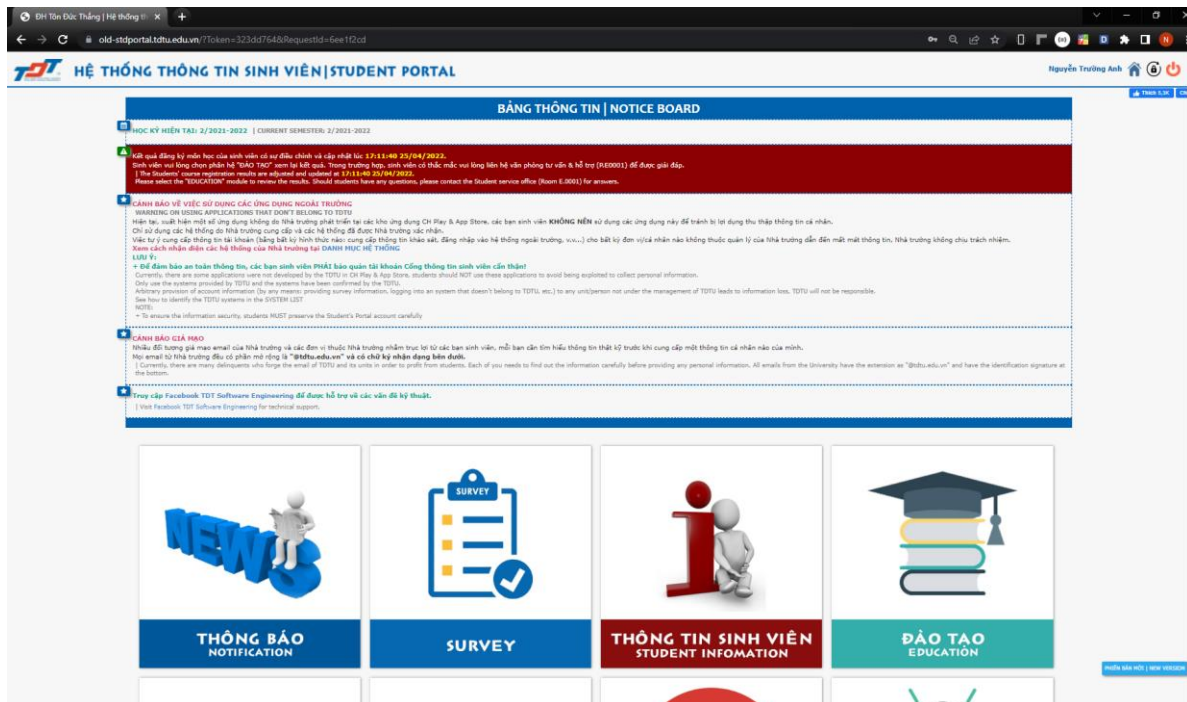
## B. PHẦN LÝ THUYẾT

### Câu hỏi 1: Anh/chị hãy cho biết front-end, back-end là gì? Cho ví dụ

Front-end là giao diện dùng để tương tác giữa người dùng với hệ thống hoặc ứng dụng, nó bao gồm tất cả những gì mà người dùng thấy khi thao tác trên Internet như: màu sắc, chữ viết, nút bấm, hình ảnh, menu, .... Front-end là sự kết hợp giữa HTML, CSS, Javascript và được điều khiển bởi trình duyệt mà bạn sử dụng (Chrome, Edge, Safari, ...).

Ví dụ:

Khi bạn truy cập vào “Cổng thông tin sinh viên Trường Đại học Tôn Đức Thắng” thì toàn bộ những gì bạn nhìn thấy từ màu sắc, logo cho đến các kiểu chữ, nút bấm, thông báo,... tất cả đều là Front-end, được thiết kế bởi một đội ngũ gồm các: nhà thiết kế cung cấp các logo, nhiếp ảnh gia cung cấp hình ảnh, lập trình viên front-end tạo nên bố cục giao diện trên website.



Hình 11. Giao diện của trang Cổng thông tin sinh viên Trường Đại học Tôn Đức Thắng

Back-end là phần sẽ cung cấp và lưu trữ các thông tin cho phía front-end, và cũng là nơi tiếp nhận các yêu cầu của phía người dùng để xử lý và trả về các kết quả phù hợp. Thành phần của back-end sẽ gồm có: máy chủ, ứng dụng và cơ sở dữ liệu.

Ví dụ:

Khi truy cập vào trang đăng nhập của “Cổng thông tin sinh viên Trường Đại học Tôn Đức Thắng”, người dùng sẽ phải tiến hành đăng nhập để có thể vào được hệ thống bằng việc cung cấp tên tài khoản cũng như mật khẩu và sau đó nhấn nút gửi. Khi nhập đúng tài khoản thì bạn sẽ có thể vào được trang chủ của cổng thông tin, nhưng khi nhập sai thì bạn sẽ nhận được thông báo là “Sai tên đăng nhập hoặc mật khẩu”, quá trình đó được thực hiện với back-end và được đội ngũ lập trình viên back-end thực hiện.

## **Câu hỏi 2: Theo anh/chị dùng nodejs để xây dựng front-end hay back-end? Vì sao?**

Nodejs được xem là một runtime environment được xây dựng dựa trên V8 Javascript Engine của Chrome giúp Nodejs có thể thực thi các đoạn mã Javascript bên ngoài các browser. Do đó, Nodejs có cơ chế non-blocking I/O giúp giảm tải đáng kể các thread xử lý các I/O, từ đó tạo ra một môi trường chiếm ít dung lượng sử dụng, do đó hỗ trợ rất tốt trong việc lập trình phía back-end được nhanh gọn và tiện lợi hơn:

- + Network và API call: Nodejs có nhiều thư viện hỗ trợ cho phép kiểm soát các câu lệnh gọi HTTP, giúp tạo ra các RESTful API dễ dàng.
- + Thao tác với database: Nodejs cung cấp nhiều thư viện giúp tương tác với đa dạng cơ sở dữ liệu (Mongo, MySQL, SQL Server,...). Từ đó giúp chúng ta có thể tạo ra các câu lệnh tương tác với cơ sở dữ liệu bằng ngôn ngữ Javascript, khiến dễ nắm bắt và làm quen hơn.
- + Xây dựng ứng dụng thời gian thực.

Ngoài ra, Nodejs cũng hỗ trợ rất nhiều trong việc lập trình phía front-end với các chức năng như:

- + Module Bundle: Hỗ trợ việc gom các file với nhiều định dạng khác nhau về chung một file, từ đó giảm dung lượng và tăng tốc độ tải dữ liệu cho các trang web.
- + Styling: hỗ trợ việc code css dễ dàng hơn, như styled-component được dùng cho Reactjs, giúp xây dựng các kiểu mẫu dễ dàng bằng việc tích hợp Javascript vào.
- + Packages: node package management (npm) cung cấp nhiều package giúp việc lập trình trở nên dễ dàng và quản lý hiệu quả hơn.

Do đó, Nodejs có thể được sử dụng cho lập trình cả phía front-end lẫn back-end. Tuy nhiên, Nodejs thường được dùng cho phía back-end hơn là front-end như việc tạo các RESTful API cho web app và mobile app.

### **Câu hỏi 3: Theo anh/chị dùng expressjs để xây dựng font-end hay back-end? Vì sao?**

Expressjs là một web application framework mã nguồn mở, miễn phí dành cho NodeJs, được dùng để phát triển các ứng dụng web và các API một cách dễ dàng và nhanh chóng. Express là thành phần back-end của MEAN stack, trong đó bao gồm: MongoDB là cơ sở dữ liệu, AngularJS cho front-end và Nodejs cho Javascript runtime environment.

Do đó, Expressjs được dùng để xây dựng back-end.

### **Câu hỏi 4: Theo anh/chị dùng expressjs là thư viện mã nguồn mở hay framework hay là ngôn ngữ lập trình web hay công cụ xây dựng web? Vì sao**

Expressjs là một web application framework mã nguồn mở và hoàn toàn miễn phí dùng cho Nodejs, được dùng để xây dựng các ứng dụng web, các API dễ dàng và nhanh chóng. Đồng thời, Expressjs sử dụng ngôn ngữ Javascript giúp cho các lập trình viên dễ dàng tiếp cận và làm quen dễ dàng, đồng thời có cộng đồng lớn mạnh trên toàn thế giới.

Vì Expressjs là một framework dành cho Nodejs nên hầu hết các đoạn code đã được viết sẵn thông qua các thư viện, giúp lập trình viên không cần tốn nhiều thời gian và công

sức cho các chức năng cơ bản. Khi đó, lập trình viên có thể tạo ra các single-page, multi-page hoặc hybrid page application thông qua Expressjs.

Ngoài ra, Expressjs có dung lượng nhẹ và giúp ích rất lớn trong việc tổ chức code ở phía máy chủ theo mô hình MVC (Model – View – Controller) được tốt hơn.

Bên cạnh đó, Expressjs được xem là thành phần dùng cho phía back-end trong MEAN stack (gồm: MongoDB là cơ sở dữ liệu, AngularJS cho front-end và Nodejs cho Javascript runtime environment) giúp quản lý các HTTP request, routing, session, xử lý các lỗi, ....Việc kết hợp dùng Nodejs với Expressjs sẽ giúp xây dựng web và các API được hiệu quả và nhanh chóng; nếu không có Expressjs thì lập trình viên phải xử lý những tính năng phức tạp trong việc thực thi tạo các API.

Expressjs cung cấp các thư viện hỗ trợ nhiều tính năng như:

- + Phát triển Server-side được nhanh hơn: Expressjs cung cấp các tính năng thường được sử dụng của Nodejs thành các hàm và các hàm đó có thể được sử dụng ở bất kỳ đâu trong chương trình; từ đó giảm thời gian viết mã.
- + Các middleware: Expressjs cung cấp các middleware hỗ trợ việc truy cập các cơ sở dữ liệu, tiếp nhận các request từ người dùng và của các phần mềm trung gian khác. Từ đó giúp tổ chức có hệ thống các chức năng của Expressjs.
- + Routing: Expressjs cho phép thực hiện các hành động khác nhau dựa vào các phương thức HTTP (Get, Post, Put, Delete,..) và các đường dẫn URL.
- + Templating: Expressjs cung cấp các template engine giúp xây dựng các nội dung động trên trang web bằng việc cung cấp các HTML template ở phía server.
- + Debugging: Expressjs hỗ trợ việc gỡ lỗi được tốt hơn bằng việc cung cấp công cụ hỗ trợ giúp xác định chính xác thành phần bị lỗi của hệ thống.

Vì vậy, kết luận chung thì Expressjs là một framework.

## TÀI LIỆU THAM KHẢO

- [1] Tài liệu về ReactJs trên trang: <https://reactjs.org/>
- [2] <https://phambinh.net/bai-viet/single-page-application-la-gi-co-phai-la-xu-huong-lap-trinh-web-trong-tuong-lai/>
- [3] <https://vn.got-it.ai/blog/single-page-application-la-gi-loi-ich-khi-su-dung-single-page-application>
- [4] <https://fullstack.edu.vn/learning/reactjs?id=2729>
- [5] <https://2020.stateofjs.com/en-US/technologies/front-end-frameworks/>
- [6] <https://vietnix.vn/tim-hieu-mo-hinh-mvc-la-gi/>
- [7] <https://viblo.asia/p/tim-hieu-ve-node-js-co-ban-ojaqG0dGEKwZ>
- [8] <https://viblo.asia/p/phan-1-tim-hieu-express-js-framework-Qbq5Qq7m5D8>
- [9] <https://itnavi.com.vn/blog/expressjs-la-gi/?amp>