TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO CUỐI KÌ**

**MÔN QUẢN TRỊ HỆ THỐNG THÔNG TIN**

**TÌM HIỂU  
TOÀN DIỆN VỀ SINGLE PAGE APPLICATION**

*Người hướng dẫn*: **THS NGUYỄN DUY THÁI**

*Người thực hiện*: **NGUYỄN CAO HẢI ĐĂNG – 51900306**

**NGUYỄN TRỌNG HIỂN – 51900332**

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**



**BÁO CÁO CUỐI KÌ**

**MÔN QUẢN TRỊ HỆ THỐNG THÔNG TIN**

**TÌM HIỂU  
TOÀN DIỆN VỀ SINGLE PAGE APPLICATION**

*Người hướng dẫn*: **THS NGUYỄN DUY THÁI**

*Người thực hiện*: **NGUYỄN CAO HẢI ĐĂNG – 51900306**

**NGUYỄN TRỌNG HIỂN – 51900332**

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 ThS. Nguyễn Duy Thái. 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 ***“Tìm hiểu toàn diện về Single Page Application"***.

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. 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.

CHÚNG TÔI XIN GỬI LỜI CẢM ƠN CHÂN THÀNH SÂU SẮC ĐẾN THẦY!

**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 Thạc sĩ Nguyễn Duy Thái. 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 tháng năm*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Nguyễn Cao Hải Đăng*

*Nguyễn Trọng Hiển*

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

Tìm hiểu toàn diện về SPA bao gồm: giới thiệu tổng quát về SPA, phân tích so sánh với những trang truyền thống như MPA, bên cạnh đó đi chi tiết về các thành phần cốt yếu tạo nên một SPA, sau cùng giới thiệu cũng như so sánh các thư viện và framework SPA phổ biến ở thời điểm hiện tạiđồng thời viết một trang web minh họa SPA sử dụng React library

Trang web minh họa về SPA này nên lấy dữ liệu từ một Web API của <https://www.themoviedb.org/>

MỤC LỤC

[LỜI CẢM ƠN i](#__RefHeading___Toc1183_1178145677)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN iii](#__RefHeading___Toc1203_1178145677)

[TÓM TẮT iv](#__RefHeading___Toc1201_1178145677)

[MỤC LỤC 1](#__RefHeading___Toc1199_1178145677)

[DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ 3](#__RefHeading___Toc1197_1178145677)

[CHƯƠNG 1 – TOÀN DIỆN VỀ SINGLE PAGE APPLICATION 1](#__RefHeading___Toc1195_1178145677)

[1.1 Giới thiệu Single Page Application (SPA) 1](#__RefHeading___Toc1213_1178145677)

[1.2 Kiến trúc và các thành phần cốt lõi xây dựng nên SPA 2](#__RefHeading___Toc1211_1178145677)

[1.2.1 AJAX 2](#__RefHeading___Toc1215_1178145677)

[1.2.2 REST 3](#__RefHeading___Toc1227_1178145677)

[1.2.3 Phân tách các mối quan tâm (SoC) 4](#__RefHeading___Toc1225_1178145677)

[1.2.4 Ràng buộc dữ liệu (Data binding) 4](#__RefHeading___Toc1223_1178145677)

[1.3 Tổng quan một số framework/ library dùng để xây dựng SPA 4](#__RefHeading___Toc1209_1178145677)

[1.3.1 ReactJS 5](#__RefHeading___Toc1221_1178145677)

[1.3.2 AngularJS 5](#__RefHeading___Toc1219_1178145677)

[1.3.3 EmberJS 5](#__RefHeading___Toc1217_1178145677)

[CHƯƠNG 2 – XÂY DỰNG CHƯƠNG TRÌNH VỚI THƯ VIỆN REACTJS 7](#__RefHeading___Toc1193_1178145677)

[2.1 Chi tiết về các đặc điểm của ReactJS 7](#__RefHeading___Toc1207_1178145677)

[2.3 Ứng dụng movie 10](#__RefHeading___Toc1205_1178145677)

[CHƯƠNG 3 – GIAO DIỆN CỦA ỨNG DỤNG 16](#__RefHeading___Toc1191_1178145677)

[CHƯƠNG 4 – KẾT LUẬN 18](#__RefHeading___Toc1189_1178145677)

[TÀI LIỆU THAM KHẢO 19](#__RefHeading___Toc1187_1178145677)

[ĐÁNH GIÁ MỨC ĐỘ HOÀN THÀNH 20](#__RefHeading___Toc1185_1178145677)

**DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT**

**CÁC CHỮ VIẾT TẮT**

|  |  |
| --- | --- |
| AJAX | Asynchronous JavaScript And XML |
| HTML | Ngôn ngữ đánh dấu siêu văn bản (HyperText Markup Language) |
| CSS | Cascading Style Sheets |
| DOM | Document Object Model |
| JSON | Javascript Object Notation |
| HTTP | Hypertext Transfer Protocol |
| REST | Representational state transfer |
| SoC | Separation of Concerns |
| SPA | Single Page Application |
| MPA | Multiple Page Application |
| XML | Extensible Markup Language |

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

**DANH MỤC HÌNH**

[Hình 1. 1 Request của website với phương thức AJAX 1](#_Toc90733933)

[Hình 1. 2 Request của website với phương thức thông thường 2](#_Toc90733934)

[Hình 1. 3 Cách nhận và gửi dữ liệu giữa Broswer và Server-side System 3](#_Toc90733935)

[Hình 1. 4 Các tương tác giữa các thành phần trong mô hình MVC 4](#_Toc90733936)

[Hình 2. 1 File App.js 10](#_Toc90735619)

[Hình 2. 2 File Movie.js 11](#_Toc90735620)

[Hình 2. 3 API data của FEATURED\_API 11](#_Toc90735621)

[Hình 2. 4 Website themoviedb.org cung cấp API về movie 12](#_Toc90735622)

[Hình 2. 5 Thông tin tài khoản sau khi tạo thành công tại website 12](#_Toc90735623)

[Hình 2. 6 Hàm getMovies và handleOnChange 13](#_Toc90735624)

[Hình 2. 7 JSX file Movie.js render ô tìm kiếm và danh sách thông qua MovieDetail Component 14](#_Toc90735625)

[Hình 2. 8 File MovieDetail.js sẽ nhận lại những props được truyền từ Movie.js và render ra theo các thẻ tương ứng 14](#_Toc90735626)

[Hình 2. 9 File MovieDe.js sẽ nhận id thông qua useParams được truyền từ file App.js và link API với video trailer tương ứng với id đó, file này sẽ trả về ReactPlayer hiện thị video trailer 15](#_Toc90735627)

[Hình 2. 10 File MovieDe.js mình sử dùng useState để lưu video và useEffect để thay đổi video theo id tương ứng ở API 15](#_Toc90735628)

[Hình 3. 1 Giao diện ban đầu khi vừa truy cập 16](#_Toc90735638)

[Hình 3. 2 Khi nhập vào ô search chỉ hiện phim spider man 16](#_Toc90735639)

[Hình 3. 3 Khi click vào 1 phim sẽ hiện ra trailer 17](#_Toc90735640)

[Hình 3. 4 view page source chỉ có thẻ <div id=”root”> 17](#_Toc90735641)

**DANH MỤC BẢNG**

[Bảng 1. 1 So sánh SPA và MPA 2](#_Toc90732910)

[Bảng 1. 2 Các phương thức của HTTP 4](#_Toc90732911)

[Bảng: Đánh giá mức độ hoàn thành 20](#_Toc90732912)

CHƯƠNG 1 – TOÀN DIỆN VỀ SINGLE PAGE APPLICATION

1.1 Giới thiệu Single Page Application (SPA)

1. *Single Page Application (SPA) là gì?*

Single Page Application, hay được viết tắt là SPA, là một giao diện web bao gồm các thành phần riêng lẻ có thể được cập nhật hoặc thay thế một cách độc lập, do đó toàn bộ trang không cần phải tải lại trên mỗi hành động, yêu cầu của người dùng đối với phía server. SPA sử dụng AJAX và HTML5 giúp cải thiện trãi nghiệm.

Khi người dung truy cập vào trang thì nội dung ban đầu của một trang web bao gồm các file Javascript, HTML, CSS sẽ được tải lên. Khi người dùng tương tác với ứng dụng hoặc điều hướng đến một trang, trang web không thực hiện tải lại toàn bộ trang. Thay vào đó, chỉ phân đoạn cụ thể của trang được được tải không đồng bộ thông quan những yêu cầu ajax và phần còn lại của nội dung trong trang vẫn không bị ảnh hưởng.

Graphical user interface, text, application, email

Description automatically generated

Hình 1. 1 Request của website với phương thức AJAX

1. *SPA so với trang truyền thống như Multiple-page application có điểm gì nổi bật?*

*Multiple-page application (MPA) là gì?*

Multiple-page ban đầu cũng tải lên toàn bộ source của trang web. Nhưng khác với SPA, khi người dùng yêu cầu tương tác như xem thông tin chi tiết của một bài viết thì toàn bộ trang web sẽ được load lại và bắt đầu tải các nội dụng mà phía client yêu cầu, do cơ chế đó khiến trãi nghiệm của người dùng không được tốt vì web khá chậm

Graphical user interface, application

Description automatically generated

Hình 1. 2 Request của website với phương thức thông thường

*So sánh SPA và MPA*

|  |  |  |
| --- | --- | --- |
| **Tiêu chí** | **Single Page Application** | **Multiple-page Application** |
| *SEO* | Không tốt | Tốt |
| *Trải nghiệm người dùng* | Tốt | Không tốt |
| *Hiệu suất* | Tốt | Không tốt |
| *Khả năng phát triển của ứng dụng* | Phát triển SPA dễ dàng hơn nhiều vì ít trang được tạo ra hơn, xây dựng và kiểm tra ít chức năng hơn và hiển thị ít nội dung hơn. Vì thế, mất ít công sức, thời gian và tiền bạc hơn để phát triển một sản phẩm SPA | Khó khăn hơn, vì MPA có số lượng tính năng cao hơn so với các giải pháp SPA, việc tạo ra chúng đòi hỏi nhiều nỗ lực và tài nguyên hơn. Thời gian phát triển tăng tỷ lệ thuận với số lượng trang được xây dựng và chức năng sẽ được thực hiện, cho nên cần thêm rất nhiều thời gian và tiền bạc |

Bảng 1.1 So sánh SPA và MPA

1.2 Kiến trúc và các thành phần cốt lõi xây dựng nên SPA

Để xây dựng được một ứng dụng web SPA hiệu quả thì ta cần phải nắm rõ một vài thành phần chính góp phần tạo ra SPA.

1.2.1 AJAX

AJAX, viết tắt của Asynchronous JavaScript And XML, là một công nghệ cho phép ứng dụng cập nhật dữ liệu mới vào DOM (Document Object Model hay Mô hình Đối tượng Tài liệu) mà không cần phải tải lại trang hiện tại. Các quy trình AJAX là không đồng bộ và do đó chúng cho phép nhiều hoạt động được thực hiện cùng một lúc. Giúp tăng tính trải nghiệm, tương tác của người dùng trong suốt quá trình sử dụng trang web

Diagram

Description automatically generated

Hình 1. 3 Cách nhận và gửi dữ liệu giữa Broswer và Server-side System

1.2.2 REST

REST (Representational state transfer) mô tả một phong cách kiến trúc thường được sử dụng kết hợp với các ứng dụng hỗ trợ AJAX. Ý tưởng đằng sau REST là làm cho dữ liệu ứng dụng có thể truy cập được tài nguyên.

HTTP giúp các ứng dụng web giao tiếp với các dịch vụ của REST bằng cách gửi các yêu cầu. HTTP xác định ngữ nghĩa giống như hành động cho các hoạt động được gọi là động từ. Dưới đây là các phương thức nổi bật của HTTP

|  |  |
| --- | --- |
| **Phương thức** | Miêu Tả |
| GET | Truy xuất tài nguyên |
| POST | Thêm một tài nguyên mới hoặc yêu cầu một hành động đối với một yêu cầu hiện có |
| PUT | Cập nhật tài nguyên hiện có. |
| DELETE | Xóa tài nguyên hiện có. |
| HEAD | Truy xuất thông tin tài nguyên. Hiệu quả giống như HEAD tiêu đề HTTP của yêu cầu. |

Bảng 1.2 Các phương thức của HTTP

1.2.3 Phân tách các mối quan tâm (SoC)

Phân tách các mối quan tâm (SoC) là một nguyên tắc kỹ thuật phần mềm ngụ ý tính mô đun: một mẫu trong thiết kế kiến trúc phân tách hành vi phần mềm trong các đơn vị được đóng gói.

MVC là một mẫu kiến trúc phần mềm phổ biến được sử dụng để tách phần biểu diễn dữ liệu trực quan khỏi mô hình bên dưới.

Diagram

Description automatically generated

Hình 1. 4 Các tương tác giữa các thành phần trong mô hình MVC

1.2.4 Ràng buộc dữ liệu (Data binding)

Ràng buộc dữ liệu là quá trình thiết lập kết nối giữa giao diện người dùng ứng dụng và logic nghiệp vụ. Có 2 loại ràng buộc dữ liệu đó chính là: 1 chiều và 2 chiều.

Liên kết dữ liệu hai chiều là một kỹ thuật được giới thiệu bởi các framework SPA hiện đại như AngularJS và Ember. Nó cho phép dữ liệu chảy theo cả hai hướng: thay đổi trong chế độ xem cập nhật Model và thay đổi trong Model cập nhật chế độ View

1.3 Tổng quan một số framework/ library dùng để xây dựng SPA

1.3.1 ReactJS

ReactJS là một thư viện rất mới do Facebook duy trì và phát triển. Phiên bản đầu tiên của React ra mắt công chúng vào khoảng năm 2013 khi Facebook quyết định xuất bản framework này dưới dạng mã nguồn mở. Kể từ đó nó đã được cộng đồng nhà phát triển chấp nhận nhanh chóng.

Các nguyên tắc cơ bản của React bao gồm ba ý tưởng sau:

* Tạo các thành phần đơn giản không làm thay đổi dữ liệu gốc.
* Chỉ hiển thị những gì đã được thay đổi thành DOM. Điều này giúp cải thiện hiệu suất
* Sử dụng luồng dữ liệu một chiều.

1.3.2 AngularJS

AngularJS là một frameword SPA được phát hành lần đầu tiên vào năm 2009. được duy trì và phát triển bởi Google. Theo các nhà phát triển, AngularJS được xây dựng dựa trên niềm tin rằng mã khai báo vượt trội hơn mã mệnh lệnh khi xây dựng giao diện người dùng và mã mệnh lệnh tốt hơn trong việc thể hiện logic nghiệp vụ. AngularJS có một số nguyên tắt sau:

* Logic nghiệp vụ được tách rời khỏi logic thao tác DOM, giúp kiểm tra tốt hơn.
* Kiểm tra phải được đánh giá cao như viết mã thực

AngularJS đã giới thiệu nhiều kỹ thuật và khái niệm mới giúp xây dựng các ứng dụng trang đơn một cách hiệu quả. AngularJS sử dụng nhiều liên kết dữ liệu hai chiều, nhưng cũng hỗ trợ liên kết một chiều.

* + 1. EmberJS

Ember là một framework SPA mã nguồn mở JavaScript phía client dùng cho phát triển các ứng dụng web và sử dụng mô hình MVC. Framework dựa trên các khái niệm cốt lõi sau :

* Thư viện mẫu: Ember sử dụng một thư viện tạo framework có tên là “Handlebars”[[1]](#footnote-2) , hỗ trợ các biểu thức, store (trình trợ giúp mẫu) và các component (các phần tử HTML tùy chỉnh).
* Router: Bộ định tuyến được sử dụng để dịch URL thành một loạt các mẫu lồng nhau. Ember tự động hóa tuần tự hóa URL dựa trên các model.
* Thành phần (component): Trong Ember, một component là một thẻ HTML tùy chỉnh được mô tả trong các mẫu Handlebars và hành vi của nó được triển khai trong JavaScript.
* Models: Mô hình là các đối tượng lưu trữ trạng thái liên tục.

Thiết kế của Ember rất giống trong AngularJS. Chúng bao gồm các khái niệm và nguyên tắc tương tự, chẳng hạn như ràng buộc dữ liệu hai chiều. Ember có nhiều chức năng soạn sẵn hơn và các khái niệm của nó ở mức cao hơn, tức là trừu tượng hơn, so với AngularJS. Điều này làm cho nó dễ dàng cho người mới bắt đầu, nhưng các nhà phát triển có kinh nghiệm có thể sẽ muốn hiểu những gì bên dưới của Ember.

CHƯƠNG 2 – XÂY DỰNG CHƯƠNG TRÌNH VỚI THƯ VIỆN REACTJS

2.1 Chi tiết về các đặc điểm của ReactJS

React là thư viện của Javascript giúp xây dựng giao diện người dùng. Sử dụng các phần biệt lập, các khối xây dựng nhỏ được gọi là component(thành phần) để tạo ra giao diện người dùng cho toàn bộ ứng dụng web.

*React Hooks*

React Hooks bản chất trong javscript thì đây là hàm được viết sẵn bởi thư viện ReactJS, được sử dụng cho các function component không dùng cho class component nâng cao hơn để kết nối các tính năng của React. Component trở nên đơn giản và dễ hiểu không bị chia ra như lifecycle của class component. Lời khuyên từ facebook thì những dự án mới thì sử dụng Hooks, còn những dự án cũ thì những component cũ giữ nguyên hoặc tối ưu sau còn những compoent mới thì sử dụng function component và hooks React Hooks được đặt tên theo quy ước là bắt đầu bằng từ “use” ví dụ useState, useEffect, useRef, useCallback, useContext ... muốn sử dụng mình phải import vào.



*useState Hook*

useState được sử dụng để bảo toàn trạng thái trong suốt sự tồn tại của các components, giúp đơn giản hóa thể hiện được trạng thái của dữ liệu ra giao diện người dùng, dữ liệu thay đổi thì giao diện thay đổi theo. useState được sử dụng khi muốn dữ liệu thay đổi thì giao diện sẽ tự render lại theo dữ liệu. useState hook và những hook khác có thể được sử dụng nhiều hơn 1 lần trong cùng 1 component.



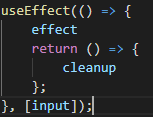
Đầu vào(input) là giá trị khởi tạo useState(initState) và trả về một mảng gồm 2 phần tử state và setState, trong lần đầu tiên thì initState sẽ đưa giá trị cho state, setState là một hàm được sử dụng để set lại state

useState được sử dụng trong phần demo:



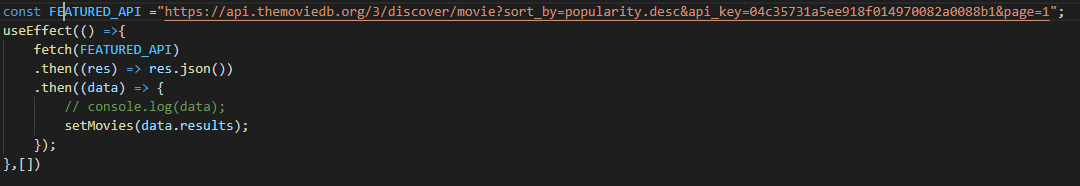
*useEffect Hook*

useEffect hook được sử dụng để thực thi lại 1 hàm cho mỗi lần re-render của component. Nó phục vụ cùng mục đích như componentDidMount, componentDidUpdate và componentWillUnmount trong các class React, nhưng được hợp nhất thành một API duy nhất.

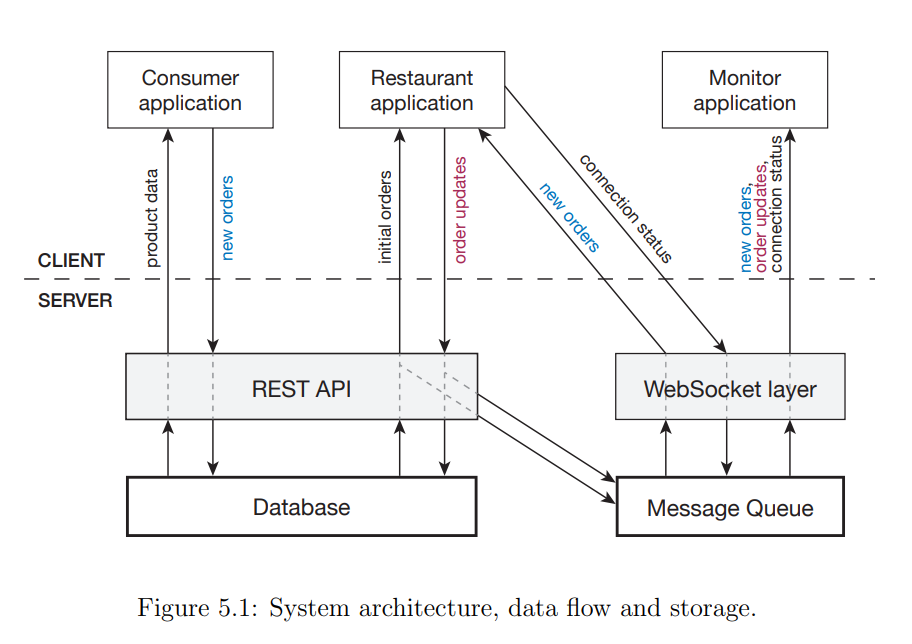


Khi dùng useEffect, yêu cầu React chạy effect sau khi thực hiện các thay đổi đối với DOM. Các effect được khai báo bên trong component để chúng có quyền truy cập vào các props và state. React chạy các effect sau mỗi lần render - bao gồm cả lần render đầu tiên. Các effect cũng có thể tùy chọn chỉ định cách “cleanup” sau chúng bằng cách trả về một hàm

useEffect được sử dụng trong phần demo:



Hình:



Hình trên minh họa dữ liệu từ database server truyền lên ứng dụng client sẽ thực hiện gọi lên api và lấy dữ liệu databse và hiện ra giao diện ứng dụng

**2.2 Create-react-app**

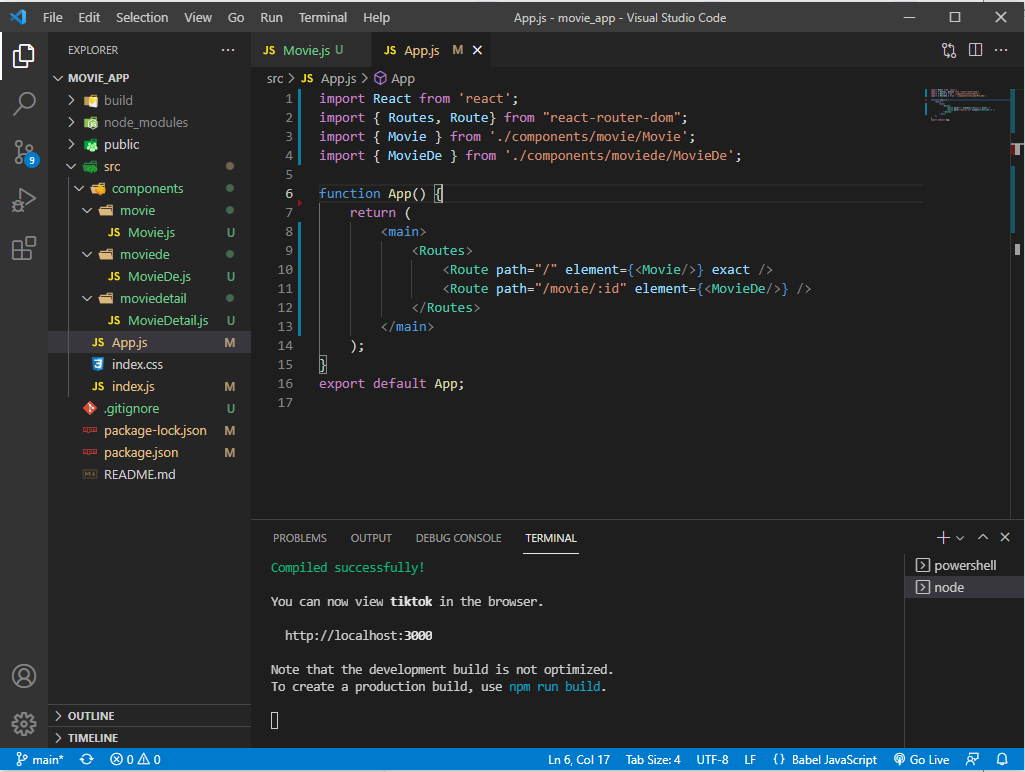
React cung cấp một công cụ có tên là create-react-app giúp thiết lập một ứng dụng React từ đầu bằng cách sử dụng Node. Nó bao gồm các công cụ hữu ích giúp tăng tốc độ phát triển như tải lại nóng và gỡ lỗi, cũng như các công cụ để xây dựng ứng dụng cho sản xuất. Nó cũng bao gồm một gói trình quản lý, trình gói và trình biên dịch trong số những thứ hữu ích khác

Điều kiện tiên quyết để sử dụng create-react-app là Node và Node Package Manager

Ứng dụng React được tạo chứa các thư mục sau:

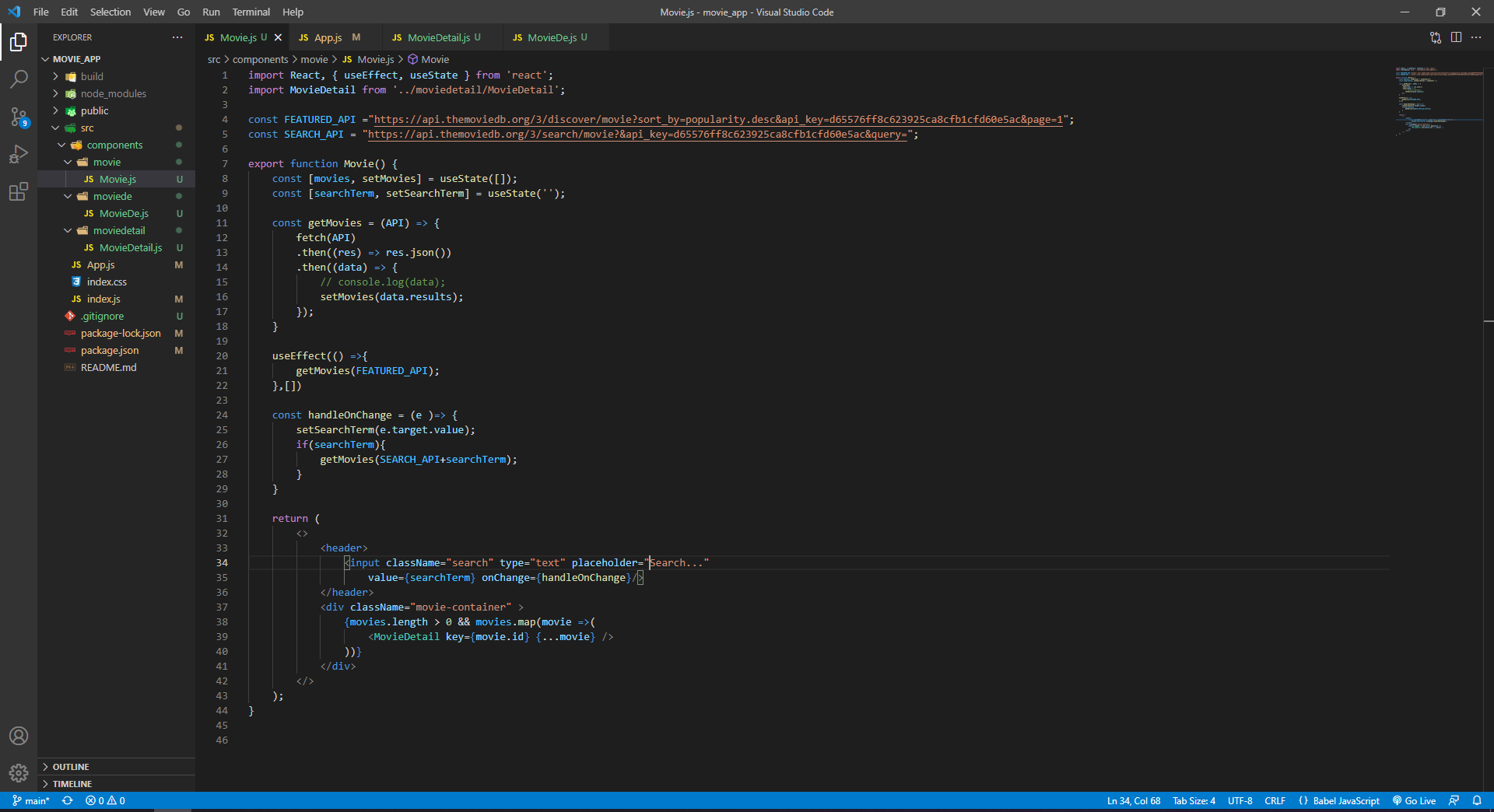
* node\_modules: chứa tất cả các folder file để phát triển các ứng dụng React.
* public: xử lý các file tĩnh trong ứng dụng, bao gồm một index.html chứa thẻ <div> nơi ứng dụng React được hiển thị, một liên kết icon ứng dụng, tiêu đề cho ứng dụng cũng như liên kết vào tệp kê khai .json nằm trong thư mục chung. Tệp manifest.json là một tệp JSON chứa thông tin về cách nó sẽ hoạt động khi được cài đặt trên điện thoại di động như một webapp.
* src: Thư mục chứa mã nguồn, bao gồm các tệp CSS và Javascript. Mục nhập điểm hiển thị ứng dụng tới DOM được đặt tên là index.js. Mình sẽ viết code ở trong thư mục này

2.3 Ứng dụng movie



Hình 2. 1 File App.js

Đây là ứng dụng để xem phim như Netflix nhưng chỉ có 2 chức năng đơn giản là search và xem trailer.Ở file App.js sẽ có 2 Route sẽ dẫn tới 2 trang là trang danh sách các movie và trailer của movie tương ứng



Hình 2. 2 File Movie.js

Ở file Movie.js sẽ có 2 link API:

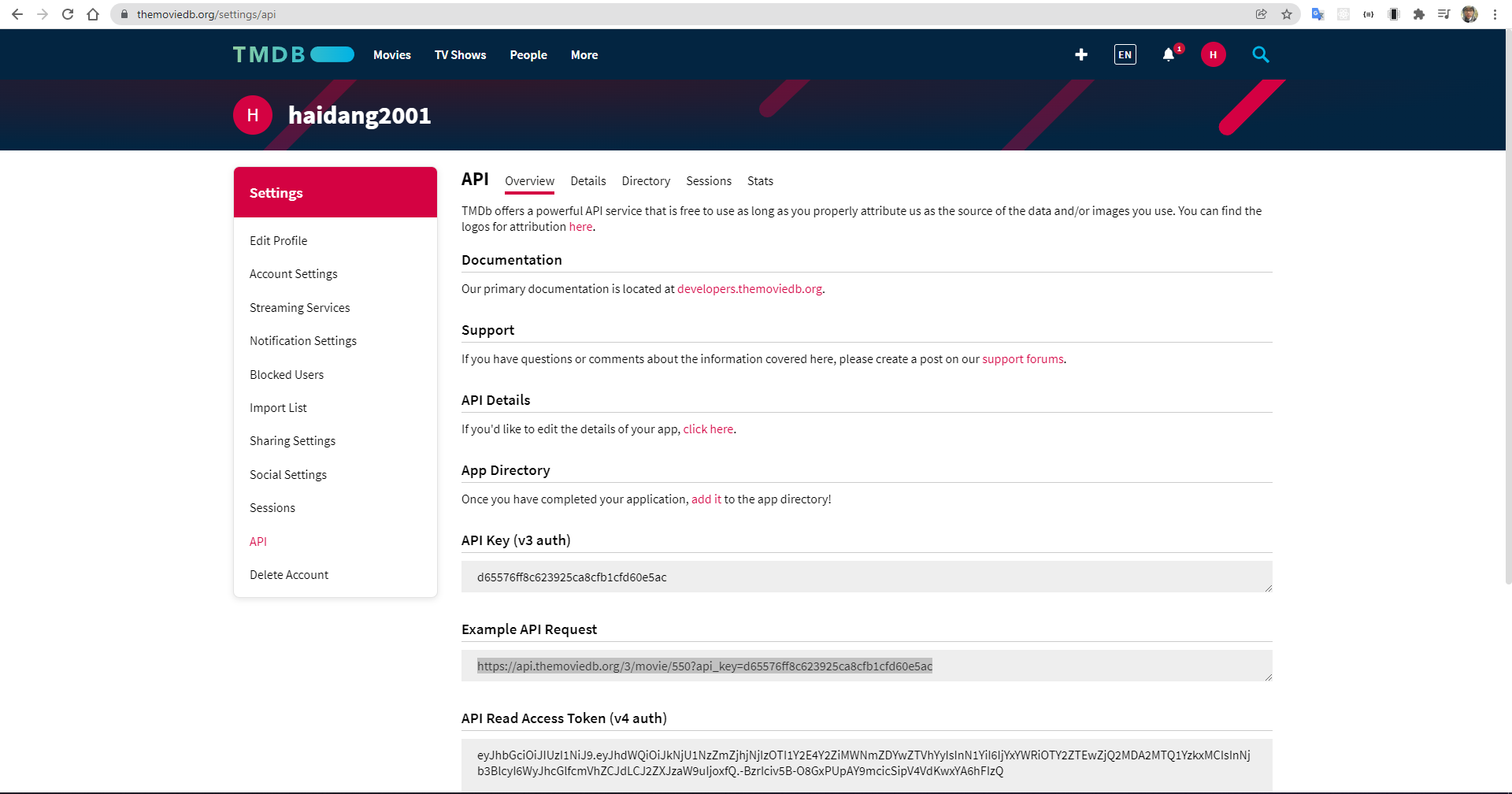
* FEATURED\_API: sẽ có api của danh sách phim, mình sẽ lấy những thông tin như id, poster\_path,title,vote\_average



Hình 2. 3 API data của FEATURED\_API

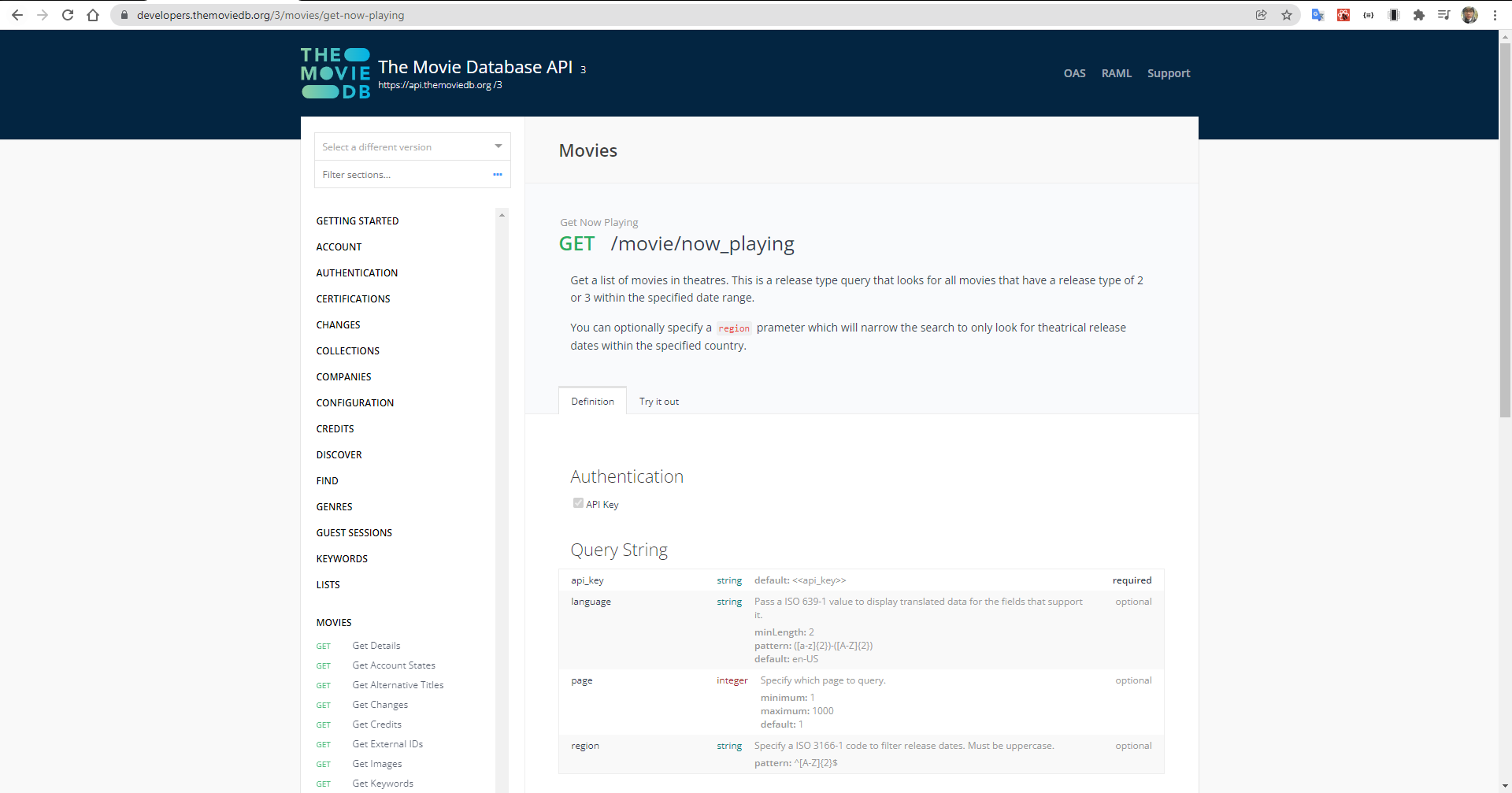
* SEARCH\_API: sẽ search phim theo tên phim

Cách lấy link API:

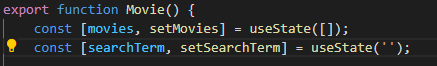


Hình 2. 4 Website themoviedb.org cung cấp API về movie

Sau khi tạo tài khoản tại link <https://www.themoviedb.org/> thành công mình sẽ cài đặt tài khoản và được cung cấp API key tiếp đến mình vào link https://developers.themoviedb.org/ để lấy dữ liệu tương ứng thông qua postman sử dụng GET, POST



Hình 2. 5 Thông tin tài khoản sau khi tạo thành công tại website



Mình sẽ có 2 state để lưu là:

- movies với giá trị khởi tạo là môt mảng rỗng để lưu danh sách các phim

- searchTerm với giá trị khởi tạo là chuỗi rỗng

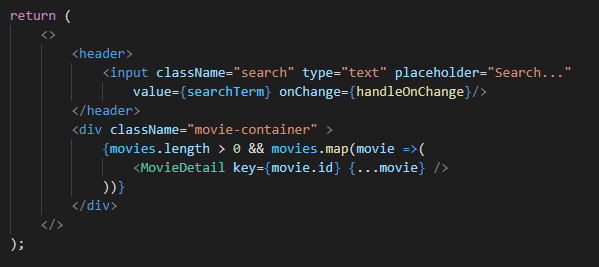


Hình 2. 6 Hàm getMovies và handleOnChange

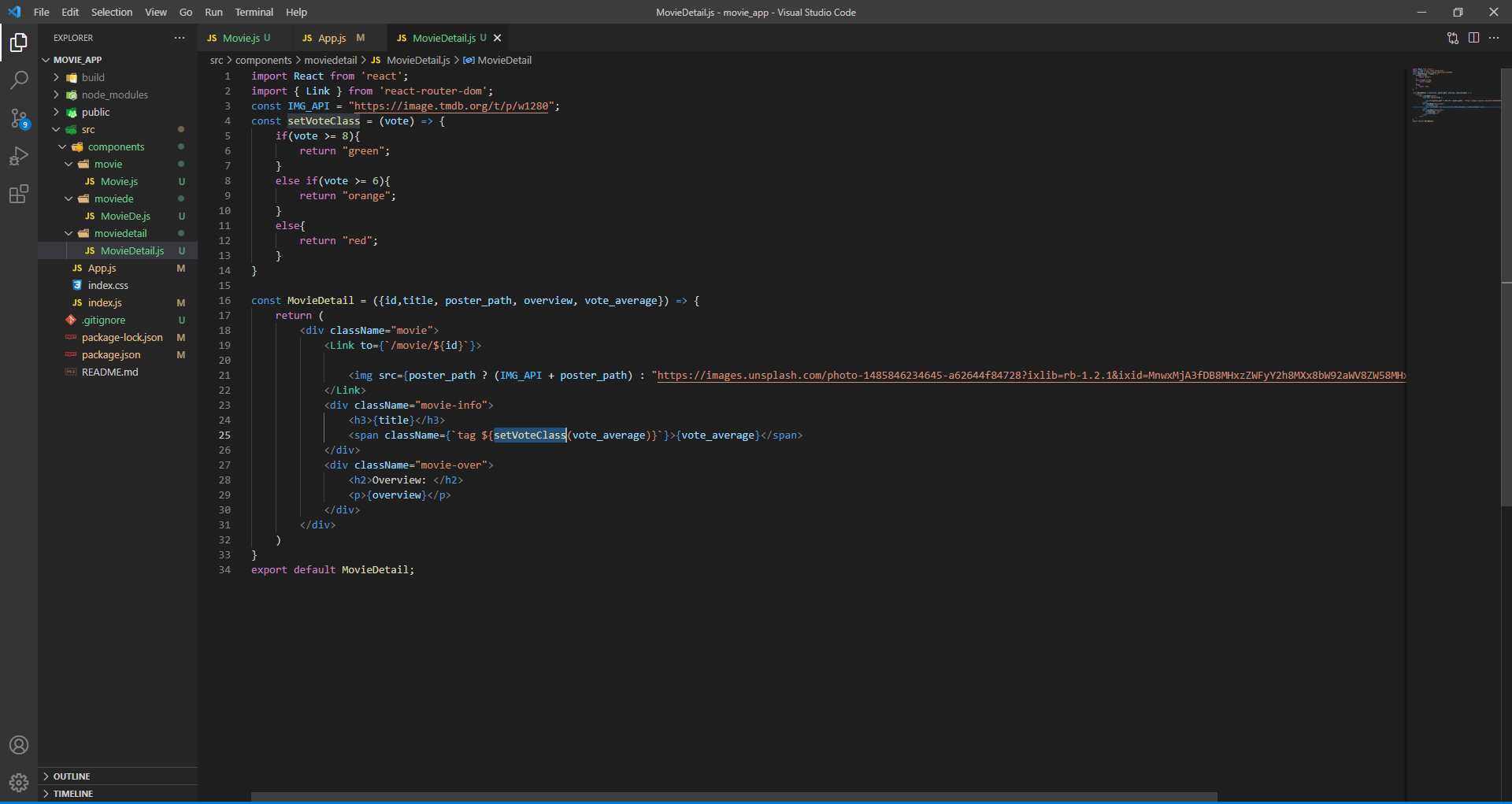
Mình có hàm getMovies để lấy dữ liệu ra từ api tương ứng

Sử dụng useEffect để thay đổi state movies thông qua hàm getMovies gọi setMovies và hiện ra giao diện

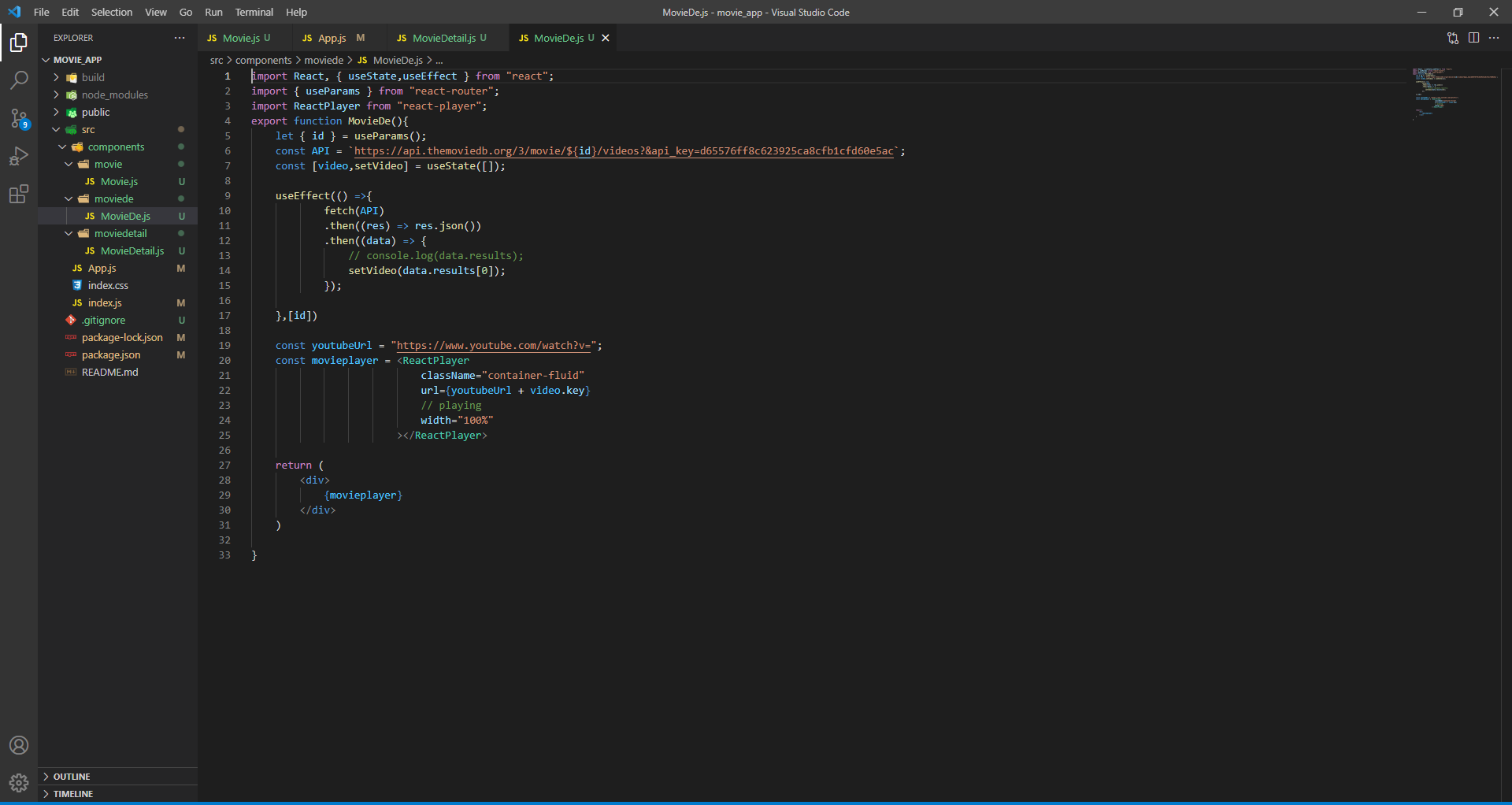
Mình có hàm handleOnChange để nhận lại giá trị người dùng gõ vào ô search và thay đổi theo movies tương ứng



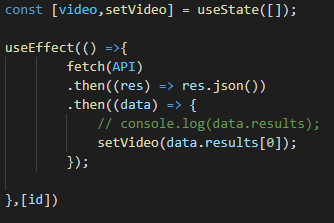
Hình 2. 7 JSX file Movie.js render ô tìm kiếm và danh sách thông qua MovieDetail Component



Hình 2. 8 File MovieDetail.js sẽ nhận lại những props được truyền từ Movie.js và render ra theo các thẻ tương ứng



Hình 2. 9 File MovieDe.js sẽ nhận id thông qua useParams được truyền từ file App.js và link API với video trailer tương ứng với id đó, file này sẽ trả về ReactPlayer hiện thị video trailer



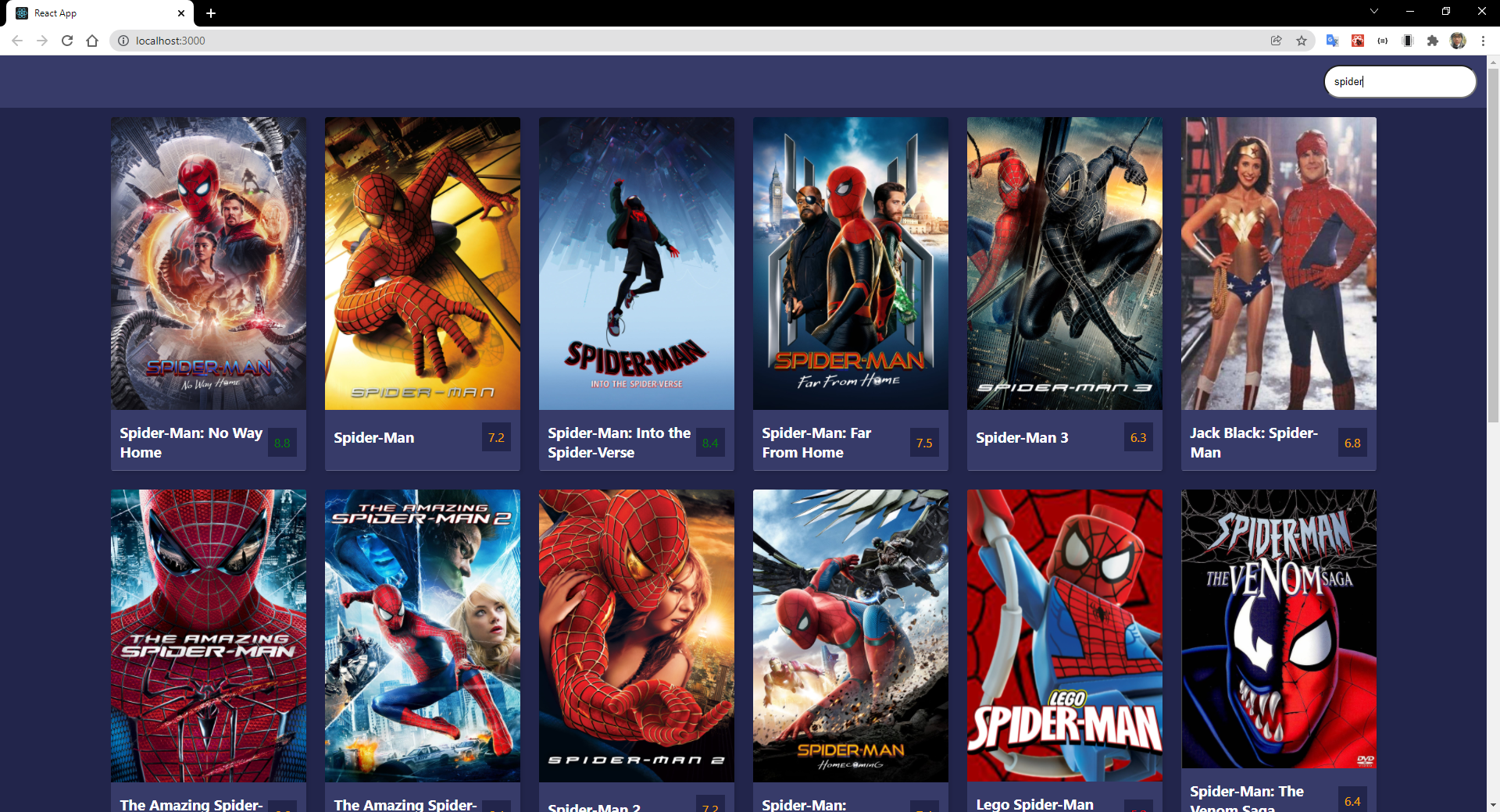
Hình 2. 10 File MovieDe.js mình sử dùng useState để lưu video và useEffect để thay đổi video theo id tương ứng ở API

CHƯƠNG 3 – GIAO DIỆN CỦA ỨNG DỤNG

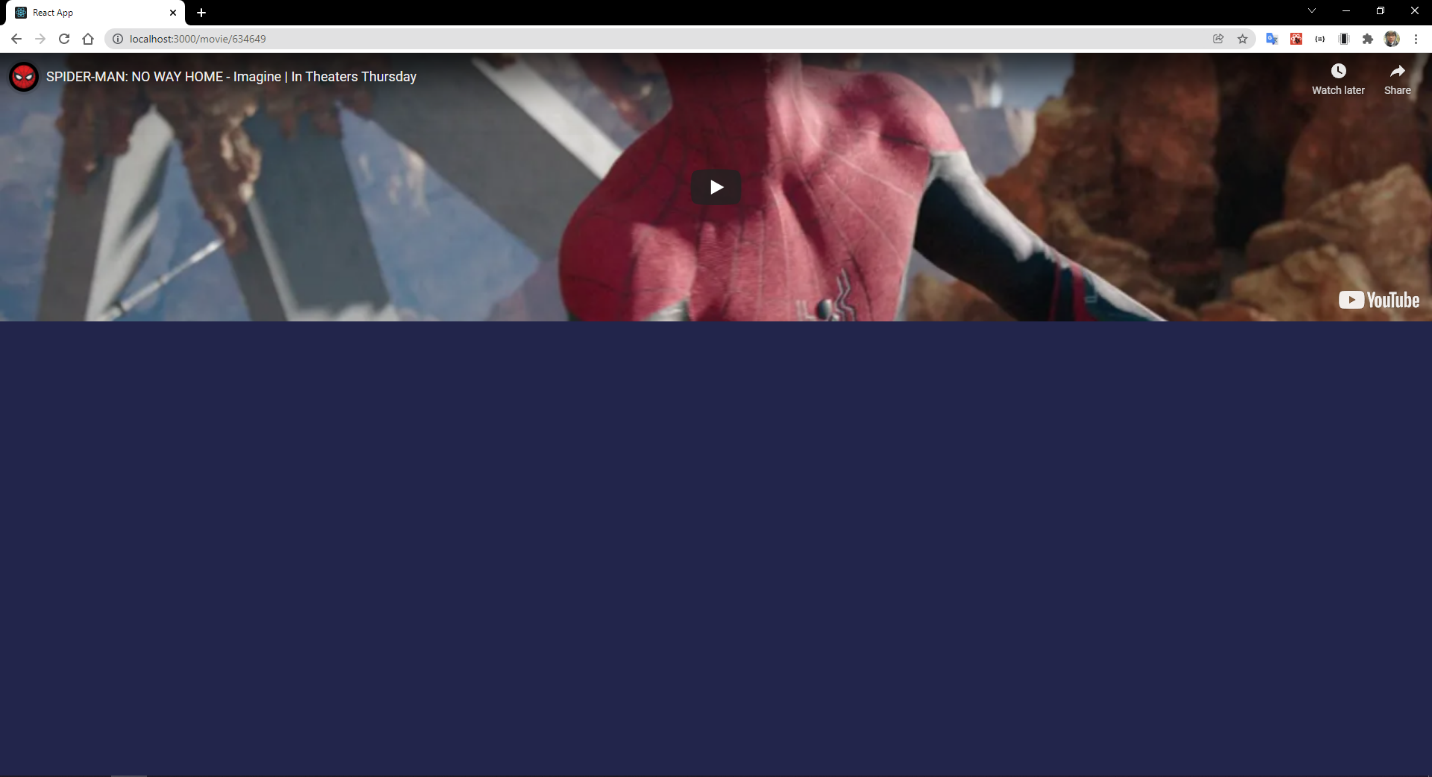
A picture containing text, bunch, different, screenshot

Description automatically generated

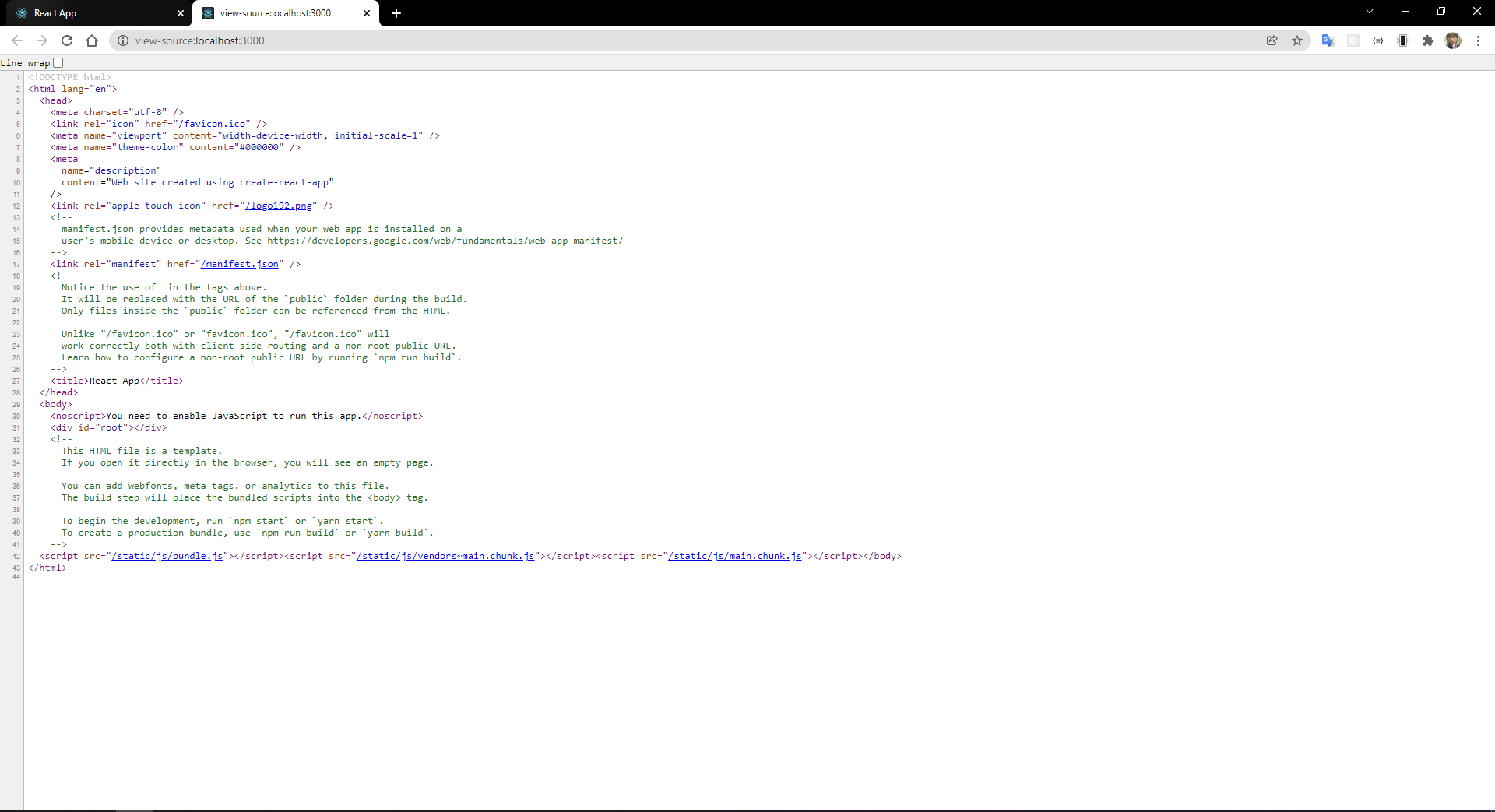
Hình 3. 1 Giao diện ban đầu khi vừa truy cập



Hình 3. 2 Khi nhập vào ô search chỉ hiện phim spider man



Hình 3. 3 Khi click vào 1 phim sẽ hiện ra trailer



Hình 3. 4 view page source chỉ có thẻ <div id=”root”>

CHƯƠNG 4 – KẾT LUẬN

Web là một trong những công nghệ phát triển nhanh chóng nhất trên thế giới. Điều này đã được chứng minh bởi sự xuất hiện của các công nghệ gần đây, chẳng hạn như HTML5, WebSockets, ECMAScript 6 và HTTP / 2. SPA là tâm điểm của sự phát triển ứng dụng web**,** chuyển việc xử lý logic từ máy chủ đến máy khách. Điều này mang lại cho các ứng dụng web một bước tiến gần hơn với các ứng dụng gốc về mặt kiến trúc. Do đó, hoàn toàn khác biệt và nổi bật hơn so với các ứng dụng web thông thường.

Dựa trên ứng dụng được xây dựng trên một thư viện ReactJS của Javascript giúp xây dựng SPA, chúng tôi nhận thấy hiệu suất của các SPA là rất tốt. Tôi cũng nhận thấy rằng việc sử dụng SPA làm cơ sở cho các ứng dụng giúp rất dễ dàng để xây dựng giao diện người dùng sử dụng các thành phần UI giống như ứng dụng và các hoạt ảnh chuyển tiếp đẹp mắt. Đồng thời giúp cho việc phát triển website sau này trở nên dễ dàng và tiết kiệm chi phí rất nhiều, mặc dù có một nhược điểm lớn là không tối ưu hóa SEO.

TÀI LIỆU THAM KHẢO

1. [Coverting+Multipage+Application+to+Single+Page+Application.pdf;jsessionid=93106E2885F3765E5CD79F67B86F7BE8 (theseus.fi)](https://www.theseus.fi/bitstream/handle/10024/106770/Coverting+Multipage+Application+to+Single+Page+Application.pdf;jsessionid=93106E2885F3765E5CD79F67B86F7BE8?sequence=1)
2. [Difference Between Vue.js ReactJS and AngularJS - Tutorial And Example](https://www.tutorialandexample.com/vue-js-vs-reactjs-vs-angularjs/)
3. [A Comparison of Single-Page and Multi-Page Applications - DZone Web Dev](https://dzone.com/articles/the-comparison-of-single-page-and-multi-page-appli)
4. [Single Page Application (devopedia.org)](https://devopedia.org/single-page-application" \l "qst-ans-1)
5. [Một ứng dụng một trang | là gì Bloomreach](https://www.bloomreach.com/en/blog/2018/07/what-is-a-single-page-application.html)
6. [Single page architecture as basis for web applications (aalto.fi)](https://aaltodoc.aalto.fi/bitstream/handle/123456789/17773/master_Nyg%E5rd_Klaus_2015.pdf?sequence=1)

ĐÁNH GIÁ MỨC ĐỘ HOÀN THÀNH

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Họ tên** | **MSSV** | **Nhiệm vụ** | **Đánh giá hoàn thành** |
| 1 | Nguyễn Cao  Hải Đăng | 51900306 |  | 100% |
| 2 | Nguyễn Trọng  Hiển | 51900332 |  | 100% |

Bảng: Đánh giá mức độ hoàn thành

1. [↑](#footnote-ref-2)