**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: CÔNG NGHỆ PHẦN MỀM**

**ĐỀ TÀI: XÂY DỰNG WEBSITE XEM PHIM MIỄN PHÍ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Mã Sinh Viên | Họ và Tên | Ngày Sinh | Lớp |
| 1 | 1671020184 | Đỗ Duy Long | 22/11/2004 | CNTT 16-05 |
| 2 | 1671020237 | Đào Hải Phong | 09/09/2004 | CNTT 16-05 |
| 3 | 1671020286 | Nguyễn Đức Thanh | 08/10/2004 | CNTT 16-05 |
| 4 | 1671020303 | Phạm Quyết Thắng | 20/09/2004 | CNTT 16-05 |

### 

**Hà Nội, năm 2025**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: CÔNG NGHỆ PHẦN MỀM**

**ĐỀ TÀI: XÂY DỰNG WEBSITE XEM PHIM MIỄN PHÍ**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Mã Sinh Viên | Họ và Tên | Ngày Sinh | Điểm | |
| Bằng Số | Bằng Chữ |
| 1 | 1671020184 | Đỗ Duy Long | 22/11/2004 |  |  |
| 2 | 1671020237 | Đào Hải Phong | 09/09/2004 |  |  |
| 3 | 1671020286 | Nguyễn Đức Thanh | 08/10/2004 |  |  |
| 4 | 1671020303 | Phạm Quyết Thắng | 20/09/2004 |  |  |

### 

### CÁN BỘ CHẤM THI 1 CÁN BỘ CHẤM THI 2

**Hà Nội, năm 2025**

**LỜI NÓI ĐẦU**

Trong thời đại công nghệ thông tin phát triển vượt bậc, việc sử dụng các ứng dụng phần mềm và nền tảng trực tuyến đã trở thành một phần thiết yếu của cuộc sống. Trong số đó, các website cung cấp dịch vụ giải trí, đặc biệt là xem phim trực tuyến, đang ngày càng được ưa chuộng nhờ vào sự tiện lợi, đa dạng nội dung và khả năng tiếp cận rộng rãi.

Với mong muốn áp dụng các kiến thức đã học trong môn **Công nghệ phần mềm**, nhóm chúng tôi đã lựa chọn đề tài **Xây dựng website xem phim miễn phí** để làm dự án thực hành. Đề tài không chỉ giúp chúng tôi hiểu sâu hơn về quy trình phát triển phần mềm từ khâu phân tích, thiết kế, đến triển khai mà còn cung cấp cơ hội để áp dụng các công nghệ xây dựng web như Reactjs, Nodejs và cơ sở dữ liệu MySQL.

Thông qua dự án này, nhóm mong muốn xây dựng một nền tảng xem phim miễn phí với giao diện thân thiện, dễ sử dụng, tích hợp các tính năng cơ bản như xem phim, tìm kiếm phim, phân loại theo thể loại, đăng nhập, đăng ký, đăng xuất, quản lý phim yêu thích.

Chúng tôi xin gửi lời cảm ơn chân thành đến giảng viên hướng dẫn **Phạm Thị Tố Nga** đã tận tình chỉ bảo trong suốt quá trình thực hiện dự án. Đồng thời, chúng tôi cũng rất mong nhận được các ý kiến đóng góp từ thầy cô và các bạn để hoàn thiện sản phẩm của mình.

Tuy thời gian thực hiện dự án còn hạn chế và kinh nghiệm thực tế chưa nhiều, nhóm đã cố gắng hết sức để hoàn thành sản phẩm. Hy vọng đề tài này sẽ mang lại giá trị thực tiễn và giúp nhóm chúng tôi nâng cao hơn nữa các kỹ năng chuyên môn.

**Trân trọng cảm ơn.**

**Nhóm 01**  
**Ngày 09, tháng 01, năm 2025**

**MỤC LỤC**

[**LỜI NÓI ĐẦU** 3](#_Toc191911221)

[**CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI** 7](#_Toc191911222)

[**1.1** **Giới thiệu chung về đề tài** 7](#_Toc191911223)

[1.1.1 Đề tài nhóm 1 7](#_Toc191911224)

[1.1.2 Các tính năng chính. 7](#_Toc191911225)

[1.1.3 Công nghệ và nền tảng 7](#_Toc191911226)

[1.1.4 Đối tượng sử dụng 7](#_Toc191911227)

[1.1.5 Kết quả mong đợi 7](#_Toc191911228)

[**1.2** **Lý do chọn đề tài** 7](#_Toc191911229)

[**1.3** **Mục tiêu** 8](#_Toc191911230)

[1.2.1 Áp dụng kiến thức và công nghệ web: 8](#_Toc191911231)

[1.3.2. Tăng cường kỹ năng phát triển phần mềm: 8](#_Toc191911232)

[1.3.3. Hỗ trợ cộng đồng giải trí trực tuyến: 9](#_Toc191911233)

[1.3.4. Thử nghiệm và cải tiến hệ thống: 9](#_Toc191911234)

[**1.4 Phạm vi nghiên cứu của đề tài** 9](#_Toc191911235)

[1.4.1 Nhu cầu và Mục tiêu 9](#_Toc191911236)

[1.4.2 Công nghệ và Phương pháp Xây dựng 9](#_Toc191911237)

[1.4.3 Quản lý Nội dung và Phân loại 9](#_Toc191911238)

[1.4.4 Bảo mật và Quyền riêng tư 10](#_Toc191911239)

[1.4.5 Trải nghiệm Người Dùng 10](#_Toc191911240)

[1.4.6 Tiện Ích và Tính Năng Hỗ Trợ 10](#_Toc191911241)

[1.4.7 Dự báo và Tiềm Năng Phát Triển 10](#_Toc191911242)

[**CHƯƠNG 2. CƠ SỞ LÍ THUYẾT** 11](#_Toc191911243)

[**2.1. Nghiên cứu và xác định yêu cầu** 11](#_Toc191911244)

[**2.2 Thiết kế hệ thống và giao diện** 11](#_Toc191911245)

[**2.3 Phát triển phần mềm** 11](#_Toc191911246)

[**2.4 Kiểm thử và sửa lỗi** 11](#_Toc191911247)

[**2.5 Triển khai và vận hành** 12](#_Toc191911248)

[**2.6 Bảo trì và nâng cấp** 12](#_Toc191911249)

[**2.7 Cài đặt hệ thống** 12](#_Toc191911250)

[**CHƯƠNG 3. CÀI ĐẶT PHẦN MỀM** 14](#_Toc191911251)

[**3.1 Các công nghệ sử dụng** 14](#_Toc191911252)

[**3.2 Cài đặt phần mềm** 15](#_Toc191911253)

**MỤC LỤC HÌNH ẢNH**

[Hình 1: Cài đặt dự án Reactjs 19](#_Toc191910857)

[Hình 2: Cài đặt các thư viện cần thiết 20](#_Toc191910858)

[Hình 3: Import thư viện xây dựng trang chủ 21](#_Toc191910859)

[Hình 4: Render dữ liệu trang chủ 21](#_Toc191910860)

[Hình 5: Import thư viện xây dựng trang phim lẻ 22](#_Toc191910861)

[Hình 6: Render dữ liệu trang phim lẻ 23](#_Toc191910862)

[Hình 7: Import các thư viện cần thiết để xây dựng trang phim bộ 24](#_Toc191910863)

[Hình 8: Render dữ liệu danh sách phim bộ ra trang phim bộ 24](#_Toc191910864)

[Hình 9: Import các thư viện cần thiết để xây dựng trang phim anime 25](#_Toc191910865)

[Hình 10: Render dữ liệu danh sách phim anime ra trang phim anime 26](#_Toc191910866)

[Hình 11: Import các thư viện cần thiết để xây dựng trang phim truyền hình 27](#_Toc191910867)

[Hình 12: Render dữ liệu danh sách phim truyền hình ra trang phim truyền hình 27](#_Toc191910868)

[Hình 13: Import các thư viện cần thiết để xây dựng trang phim yêu thích 28](#_Toc191910869)

[Hình 14: Render dữ liệu danh sách phim yêu thích ra trang phim yêu thích 29](#_Toc191910870)

[Hình 15: Import các thư viện cần thiết để xây dựng trang đăng nhập 30](#_Toc191910871)

[Hình 16: Code giao diện form đăng nhập 30](#_Toc191910872)

[Hình 17: Import các thư viện cần thiết để xây dựng trang đăng ký 31](#_Toc191910873)

[Hình 18: Code giao diện form đăng ký 32](#_Toc191910874)

[Hình 19: Import các thư viện cần thiết để xây dựng trang hiển thị kết quả tìm kiếm 33](#_Toc191910875)

[Hình 20: Hiển thị dữ liệu kết quả tìm kiếm 34](#_Toc191910876)

**CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI**

* 1. **Giới thiệu chung về đề tài** 
     1. **Đề tài nhóm 1**
* Xây dựng website xem phim miễn phí, website xem phim miễn phí là một nền tảng trực tuyến cho phép người dùng truy cập và thưởng thức các bộ phim mà không cần trả phí. Mục tiêu chính của đề tài bao gồm:
* Cung cấp kho phim đa dạng: Từ phim điện ảnh, phim truyền hình, đến phim hoạt hình hoặc các thể loại khác.
* Tạo trải nghiệm người dùng tối ưu: Giao diện thân thiện, dễ sử dụng, tối ưu trên nhiều thiết bị (desktop, tablet, mobile).
* Đảm bảo bảo mật: Đáp ứng bảo vệ dữ liệu cá nhân người dùng.

### 1.1.2 Các tính năng chính.

* Trang chủ: Hiển thị danh mục phim nổi bật, phim mới cập nhật và phim theo thể loại.
* Tìm kiếm và lọc phim: Người dùng có thể tìm kiếm phim theo tên.
* Xem phim: Trình phát video full hd có phụ đề tiếng việt, tua nhanh.
* Tài khoản người dùng: Đăng ký/đăng nhập, quản lý danh sách phim yêu thích. Tích hợp quản trị: Hệ thống quản lý phim, thể loại dành cho admin.

### 1.1.3 Công nghệ và nền tảng

* Front-end: React.js, scss xây dựng giao diện đẹp mắt.
* Back-end: Node.js với Express.js để xử lý logic và API.
* Cơ sở dữ liệu: MySQL để lưu trữ thông tin phim, người dùng.

### 1.1.4 Đối tượng sử dụng

* Người dùng yêu thích xem phim trực tuyến, đặc biệt là các bạn trẻ có nhu cầu giải trí nhưng không muốn trả phí.
* Quản trị viên hệ thống để quản lý nội dung và người dùng.

### 1.1.5 Kết quả mong đợi

* Một nền tảng trực tuyến đáp ứng tốt nhu cầu giải trí của người dùng.
* Giao diện bắt mắt, tính năng tiện lợi và ổn định.
* Xây dựng cộng đồng người dùng đông đảo, góp phần phát triển hệ sinh thái giải trí trực tuyến.
  1. **Lý do chọn đề tài**

Việc lựa chọn đề tài "Xây dựng website xem phim miễn phí" xuất phát từ những lý do thực tiễn và giá trị ứng dụng mà nó mang lại. Trước hết, trong bối cảnh nhu cầu giải trí trực tuyến ngày càng tăng, đặc biệt với sự phổ biến của internet và công nghệ, một nền tảng xem phim miễn phí sẽ đáp ứng được nhu cầu của đa dạng người dùng, từ học sinh, sinh viên đến người lao động, mang lại giá trị giải trí cao. Đây không chỉ là một xu hướng, mà còn là một nhu cầu thiết yếu trong cuộc sống hiện đại.

Đề tài mang tính ứng dụng cao, giúp người thực hiện áp dụng các kiến thức lý thuyết vào thực tế, từ thiết kế giao diện người dùng (UI/UX), lập trình web, đến xử lý cơ sở dữ liệu và tối ưu hóa hiệu suất website. Việc xây dựng một hệ thống quản lý kho phim lớn, tổ chức và phân loại nội dung hiệu quả đòi hỏi sự sáng tạo và khả năng giải quyết vấn đề, tạo cơ hội phát triển kỹ năng chuyên môn toàn diện.

Bên cạnh đó, thiết kế giao diện Responsive để tương thích với mọi thiết bị. Điều này không chỉ nâng cao giá trị trải nghiệm người dùng mà còn giúp website cạnh tranh với các nền tảng tương tự.

Cuối cùng, website xem phim miễn phí không chỉ đơn thuần là một dự án công nghệ mà còn có thể tạo ra một cộng đồng chia sẻ về các bộ phim, từ đó gia tăng giá trị kết nối xã hội. Những yếu tố trên cho thấy đề tài vừa có tính thực tiễn, vừa mang lại cơ hội phát triển toàn diện về kỹ năng chuyên môn và nhận thức xã hội.

* 1. **Mục tiêu**

### 1.2.1 Áp dụng kiến thức và công nghệ web:

Sử dụng các ngôn ngữ và công nghệ hiện đại như Reactjs để thiết kế và phát triển giao diện người dùng thân thiện, tương tác tốt.

Tận dụng Node.js để phát triển phần backend của website, hỗ trợ xử lý dữ liệu và giao tiếp với cơ sở dữ liệu.

Quản lý và lưu trữ thông tin phim, người dùng, và các chức năng khác thông qua hệ thống cơ sở dữ liệu MySQL.

. Phát triển nền tảng xem phim trực tuyến:

Xây dựng một website với các tính năng chính như:

* Tìm kiếm và phân loại phim theo thể loại, quốc gia, năm phát hành.
* Xem phim trực tuyến với chất lượng video tốt.
* Đăng ký/đăng nhập tài khoản người dùng để cá nhân hóa trải nghiệm (quản lý danh sách phim yêu thích).
* Đảm bảo website hoạt động mượt mà trên các thiết bị như máy tính và điện thoại di động.

### 1.3.2. Tăng cường kỹ năng phát triển phần mềm:

Làm quen với việc xây dựng ứng dụng web full-stack, kết hợp giữa frontend (Reactjs) và backend (Node.js, MySQL).

Hiểu và áp dụng các khái niệm cơ bản về API RESTful để kết nối giữa client và server.

### 1.3.3. Hỗ trợ cộng đồng giải trí trực tuyến:

Tạo ra một nền tảng giải trí miễn phí, dễ sử dụng, và phục vụ được nhiều người dùng với nội dung phong phú và đa dạng.

### 1.3.4. Thử nghiệm và cải tiến hệ thống:

Thực hiện các bài kiểm thử chức năng (functional testing), hiệu suất (performance testing) và bảo mật (security testing) để đảm bảo website hoạt động ổn định trong thực tế.

Nhận phản hồi từ người dùng để cải tiến sản phẩm.

## **1.4 Phạm vi nghiên cứu của đề tài**

### 1.4.1 Nhu cầu và Mục tiêu

- Nhu cầu người dùng  
Mục tiêu đầu tiên là tìm hiểu rõ nhu cầu của người dùng khi họ tìm kiếm các dịch vụ xem phim trực tuyến miễn phí. Những yếu tố quan trọng bao gồm: chất lượng video, tốc độ tải trang, và trải nghiệm người dùng. Họ cũng cần một giao diện dễ sử dụng, thân thiện và hỗ trợ tính năng tìm kiếm phim hiệu quả để nhanh chóng tìm ra bộ phim mình muốn xem.

- Mục tiêu phát triển  
Mình muốn xây dựng một website xem phim miễn phí với những đặc điểm nổi bật như:

* Chất lượng video ổn định và hỗ trợ đa dạng thể loại phim.
* Tính năng tìm kiếm phim theo tên phim.
* Tương thích trên nhiều nền tảng: website phải hoạt động mượt mà trên cả PC và thiết bị di động, giúp người dùng dễ dàng xem phim mọi lúc, mọi nơi.

### 1.4.2 Công nghệ và Phương pháp Xây dựng

- Công nghệ Front-end  
Để tạo ra giao diện người dùng đẹp mắt, tương thích với nhiều loại thiết bị và dễ sử dụng, tôi sẽ sử dụng Reactjs.

- Công nghệ Back-end  
Phần back-end sẽ được xây dựng với công nghệ Node.js với Expressjs để xử lý các yêu cầu từ người dùng và quản lý cơ sở dữ liệu. Những công nghệ này sẽ giúp website hoạt động mượt mà, đảm bảo xử lý đồng thời nhiều yêu cầu của người dùng mà không gặp phải sự cố.

- Cơ sở dữ liệu  
Website cần một hệ thống cơ sở dữ liệu mạnh mẽ để lưu trữ thông tin phim, người dùng và các tương tác khác. Tôi sẽ lựa chọn hệ quản trị cơ sở dữ liệu MySQL với khả năng mở rộng tốt

- Quản lý phim   
Để người dùng có thể xem phim mượt mà, tôi sẽ xây dựng một hệ thống quản lý phim, hỗ trợ upload, phân loại, tìm kiếm phim dễ dàng.

### 1.4.3 Quản lý Nội dung và Phân loại

- Cập nhật phim mới  
Website cần một hệ thống quản lý nội dung (CMS) để dễ dàng thêm mới, chỉnh sửa, và xóa các bộ phim. Điều này giúp giữ cho nội dung luôn được cập nhật với những bộ phim mới nhất.

- Phân loại phim  
Website sẽ cung cấp các bộ phim được phân loại rõ ràng theo thể loại. Điều này giúp người dùng dễ dàng tìm kiếm phim phù hợp với sở thích của mình.

- Tính năng tìm kiếm  
Tính năng tìm kiếm sẽ được tối ưu hóa để người dùng có thể tìm thấy bộ phim mình muốn một cách nhanh chóng và chính xác nhất.

### 1.4.4 Bảo mật và Quyền riêng tư

- Bảo mật người dùng  
Website sẽ có hệ thống đăng ký, đăng nhập và bảo mật tài khoản người dùng với các biện pháp mã hóa dữ liệu để đảm bảo an toàn cho thông tin cá nhân của người dùng.

### 1.4.5 Trải nghiệm Người Dùng

- Giao diện và trải nghiệm người dùng  
Giao diện của website cần phải đơn giản, dễ hiểu và dễ sử dụng. Người dùng phải dễ dàng tìm thấy phim và trải nghiệm xem phim mượt mà mà không gặp phải khó khăn nào.

- Tối ưu hóa cho di động  
Website sẽ được tối ưu hóa để tương thích và hoạt động mượt mà trên các thiết bị di động, giúp người dùng xem phim mọi lúc, mọi nơi.

- Tốc độ và hiệu suất  
Tôi cũng đặc biệt chú trọng đến tốc độ tải trang và chất lượng video khi phát. Website cần có khả năng tải nhanh và không gây độ trễ khi người dùng bắt đầu phát video.

### 1.4.6 Tiện Ích và Tính Năng Hỗ Trợ

Danh sách phim yêu thích.   
Tính năng này sẽ giúp người dùng theo dõi phim yêu thích.

### 1.4.7 Dự báo và Tiềm Năng Phát Triển

Dự báo xu hướng phát triển:   
 Với sự phát triển mạnh mẽ của các nền tảng streaming như Netflix và YouTube, tôi dự đoán rằng thị trường phim trực tuyến miễn phí sẽ tiếp tục phát triển mạnh mẽ trong tương lai, và người dùng sẽ yêu cầu chất lượng tốt hơn, dịch vụ ổn định hơn.

**CHƯƠNG 2. CƠ SỞ LÍ THUYẾT**

**2.1. Nghiên cứu và xác định yêu cầu**

Trước tiên, mình cần xác định rõ người dùng thực sự muốn gì. Mình có thể bắt đầu bằng cách làm một số khảo sát hoặc tìm hiểu những website xem phim miễn phí hiện tại để xem người dùng đang kỳ vọng gì, cũng như những điểm họ không hài lòng. Ví dụ, nhiều người muốn chất lượng video ổn định, không lag, có thể dễ dàng tìm kiếm phim mình yêu thích, giao diện đơn giản và dễ dùng. Mình cũng phải xác định rõ các tính năng cần thiết, như tìm kiếm phim, phân loại phim theo thể loại. Ngoài các tính năng, mình cũng cần hiểu rõ về yêu cầu phi chức năng, như tốc độ tải trang phải nhanh, website phải dễ sử dụng trên cả điện thoại và máy tính, và bảo mật thông tin người dùng. Mình phải làm rõ tất cả những điều này ngay từ đầu để đảm bảo không bị thiếu sót trong quá trình phát triển.

**2.2 Thiết kế hệ thống và giao diện**

Khi đã có yêu cầu rõ ràng, mình chuyển sang giai đoạn thiết kế. Đầu tiên là giao diện người dùng (UI). Mình sẽ vẽ ra các bản vẽ sơ bộ về giao diện của trang chủ, trang tìm kiếm phim, trang chi tiết phim, và các phần khác của website. Mình muốn website nhìn trực quan, dễ hiểu, và không có quá nhiều yếu tố gây nhiễu, để người dùng có thể tìm phim nhanh chóng mà không phải mất thời gian.

Mình cũng phải chú ý đến trải nghiệm người dùng (UX), làm sao để các thao tác trên website thật mượt mà. Ví dụ, khi người dùng bấm vào một thể loại phim nào đó, phim phải được hiển thị ngay lập tức, không bị chậm trễ.

Bên cạnh giao diện, mình cũng cần phải thiết kế kiến trúc của hệ thống. Mình nghĩ sẽ sử dụng mô hình ba lớp (client-server-database), để tách biệt rõ ràng phần giao diện, xử lý logic và lưu trữ dữ liệu. Sơ đồ luồng dữ liệu là một phần không thể thiếu để mình hình dung được cách mà dữ liệu sẽ được di chuyển trong hệ thống.

**2.3 Phát triển phần mềm**

Giờ là lúc bắt tay vào lập trình. Đầu tiên là phần Front-end. Mình sẽ dùng Reactjs, scss cho phần này, vì đây là công nghệ cơ bản mà mình cần dễ duy trì và dễ phát triển sau này.

Tiếp theo là phần Back-end. Mình sẽ dùng Node.js với Expressjs để xử lý các yêu cầu từ người dùng, quản lý cơ sở dữ liệu. Mình cũng sẽ chọn một hệ quản trị cơ sở dữ liệu như MySQL để lưu trữ thông tin phim, người dùng.

**2.4 Kiểm thử và sửa lỗi**

Sau khi phát triển xong, mình sẽ tiến hành kiểm thử để đảm bảo rằng website hoạt động đúng như mong đợi. Trước hết, mình phải kiểm tra các tính năng cơ bản như tìm kiếm phim, đăng ký tài khoản, xem phim, lưu phim yêu thích. Mọi thứ phải chạy mượt mà, không gặp phải lỗi gì.

Ngoài ra, kiểm thử hiệu suất cũng cần phải được thực hiện. Mình sẽ kiểm tra tốc độ tải trang và chất lượng video khi có nhiều người truy cập cùng một lúc. Tốc độ tải phải nhanh, và video phải phát mượt mà mà không bị giật hay dừng lại.

**2.5 Triển khai và vận hành**

Sau khi kiểm thử và mọi thứ đã ổn định, mình sẽ triển khai website lên môi trường thực tế. Mình sẽ sử dụng các dịch vụ Vercel để hosting website, vì chúng hỗ trợ mở rộng và bảo mật tốt. Sau khi website chạy ổn định, mình sẽ tiếp tục theo dõi, đảm bảo mọi thứ hoạt động tốt.

**2.6 Bảo trì và nâng cấp**

Sau khi website đã ra mắt, mình sẽ tiếp tục theo dõi và bảo trì hệ thống. Dữ liệu người dùng, các bộ phim mới, hay thậm chí là các tính năng mới cần được cập nhật thường xuyên. Mình cũng cần phải bảo mật liên tục, tránh các cuộc tấn công từ bên ngoài.

Sau khi website đã ra mắt, mình sẽ tiếp tục theo dõi và bảo trì hệ thống. Dữ liệu người dùng, các bộ phim mới, hay thậm chí là các tính năng mới cần được cập nhật thường xuyên.

Mình sẽ luôn thu thập phản hồi từ người dùng và cải tiến giao diện, tốc độ cũng như các tính năng để website ngày càng tốt hơn.

**2.7 Cài đặt hệ thống**

**Front-end (React.js + SCSS)**

React.js: Dùng cho giao diện người dùng, giúp tạo component tái sử dụng và tối ưu hiệu suất.

SCSS: Hỗ trợ viết CSS dễ quản lý hơn với biến, mixins, nesting.

React Router: Quản lý các route của ứng dụng như trang chủ, danh sách phim, chi tiết phim, đăng nhập, đăng ký, v.v.

Axios: Gửi yêu cầu API tới Back-end.

**Back-end (Node.js + Express.js)**

Express.js: Framework giúp xử lý các request từ Front-end.

Middleware: Dùng cors và jsonwebtoken để xử lý dữ liệu gửi lên và bảo mật API.

Bcrypt: Mã hóa mật khẩu người dùng trước khi lưu vào DB.

Sequelize (ORM): Giúp làm việc với MySQL dễ dàng hơn.

**Database (MySQL)**

Bảng Users: Lưu thông tin người dùng (id, email, mật khẩu, role, v.v.).

Bảng Phims: Lưu thông tin phim (id, tiêu đề, mô tả, thể loại, năm sản xuất, hình ảnh, v.v.).

Bảng Danhmucs: Lưu danh mục phim (id, tên danh mục).

Bảng TapPhims: Lưu danh sách các tập phim của từng phim

Bảng YeuThichs: Lưu danh sách các phim yêu thích của người dùng.

**CHƯƠNG 3. CÀI ĐẶT PHẦN MỀM**

## **3.1** **Các công nghệ sử dụng**

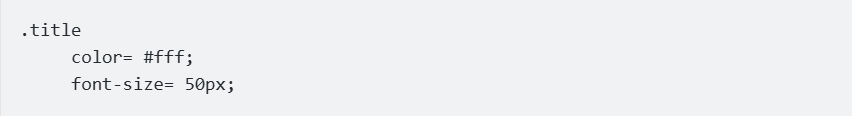
**Frond-end (Reactjs + SCSS)**

1 Reactjs là gì ?

ReactJS 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.

2 SCSS là gì ?

SASS được thiết kế và viết bởi các lập trình viên Ruby. Bởi vậy, Sass stylesheets sử dụng cú pháp giống như Ruby với việc không có các dấu ngoặc nhọn {}, dấu chấm phẩy, việc viết CSS như vậy không 'gần' CSS thuần nên việc hiểu và đọc code SASS rất khó hiểu, kiểu như này:



Sass giống như vậy cho đến khi phiên bản 3.0 được phát hành vào tháng 5/2010, nó giới thiệu một cú pháp mới được gọi là SCSS (Sassy CSS). Cú pháp này nhằm thu hẹp khoảng cách giữa Sass và CSS bằng cách mang tới một cú pháp thân thiện với CSS.

Hiểu 1 cách đơn giản, SCSS là một bản nâng cấp của SASS giúp chúng ta viết CSS một cách dễ dàng và dễ hiểu hơn.

**Back-end (Node.js + Express.js)**

1 Nodejs là gì?

NodeJS là một nền tảng được xây dựng trên “[V8 Javascript engine](https://topdev.vn/blog/cach-thuc-hoat-dong-cua-javascript-v8-engine-va-5-meo-toi-uu-hoa/)” được viết bằng c++ và Javascript. Nền tảng này được phát triển bởi Ryan Lienhart Dahl vào năm 2009.

2 Expressjs là gì?

Expressjs là một framework được xây dựng trên nền tảng của Nodejs. Nó cung cấp các tính năng mạnh mẽ để phát triển web hoặc mobile. Expressjs hỗ trợ các method HTTP và midleware tạo ra API vô cùng mạnh mẽ và dễ sử dụng.

**Database(MySQL)**

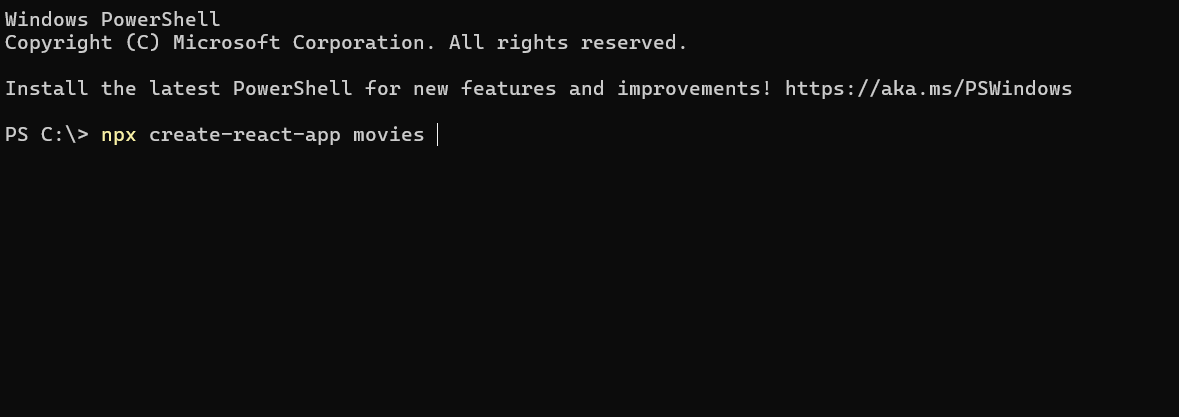
MySQL là gì?

MySQL là một hệ thống quản trị cơ sở dữ liệu (Relational Database Management System - RDBMS) mã nguồn mở hoạt động theo mô hình client - server. MySQL được phát triển bởi Oracle Corporation và được phát hành miễn phí cho cộng đồng người dùng.

**3.2 Cài đặt phần mềm**

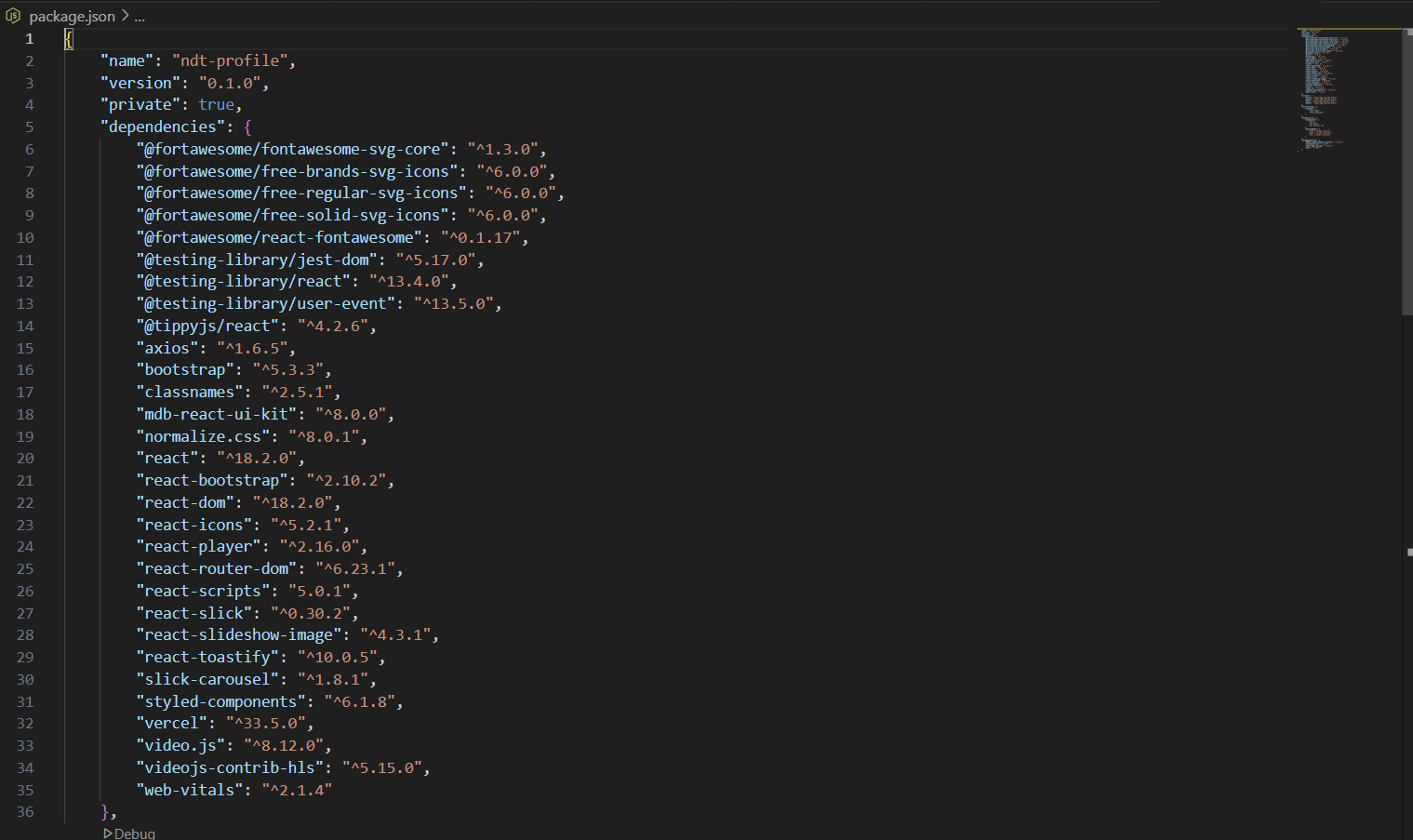
1 Cài đặt dự án Reactjs để xây dựng Frond-end

Sử dụng dòng lệnh: npx create-react-app movies



Hình 1: Cài đặt dự án Reactjs

Cài đặt các thư viện cần thiết để xây dựng giao diện



Hình 2: Cài đặt các thư viện cần thiết

*Xây dựng trang chủ:*

Mô tả: Tại trang chủ sẽ có phần header, footer, banner giới thiệu phim và truy vấn để render hiển thị ra danh sách phim của từng loại phim bằng cách gọi API đã viết ở phía Back-end.

Các chức năng:

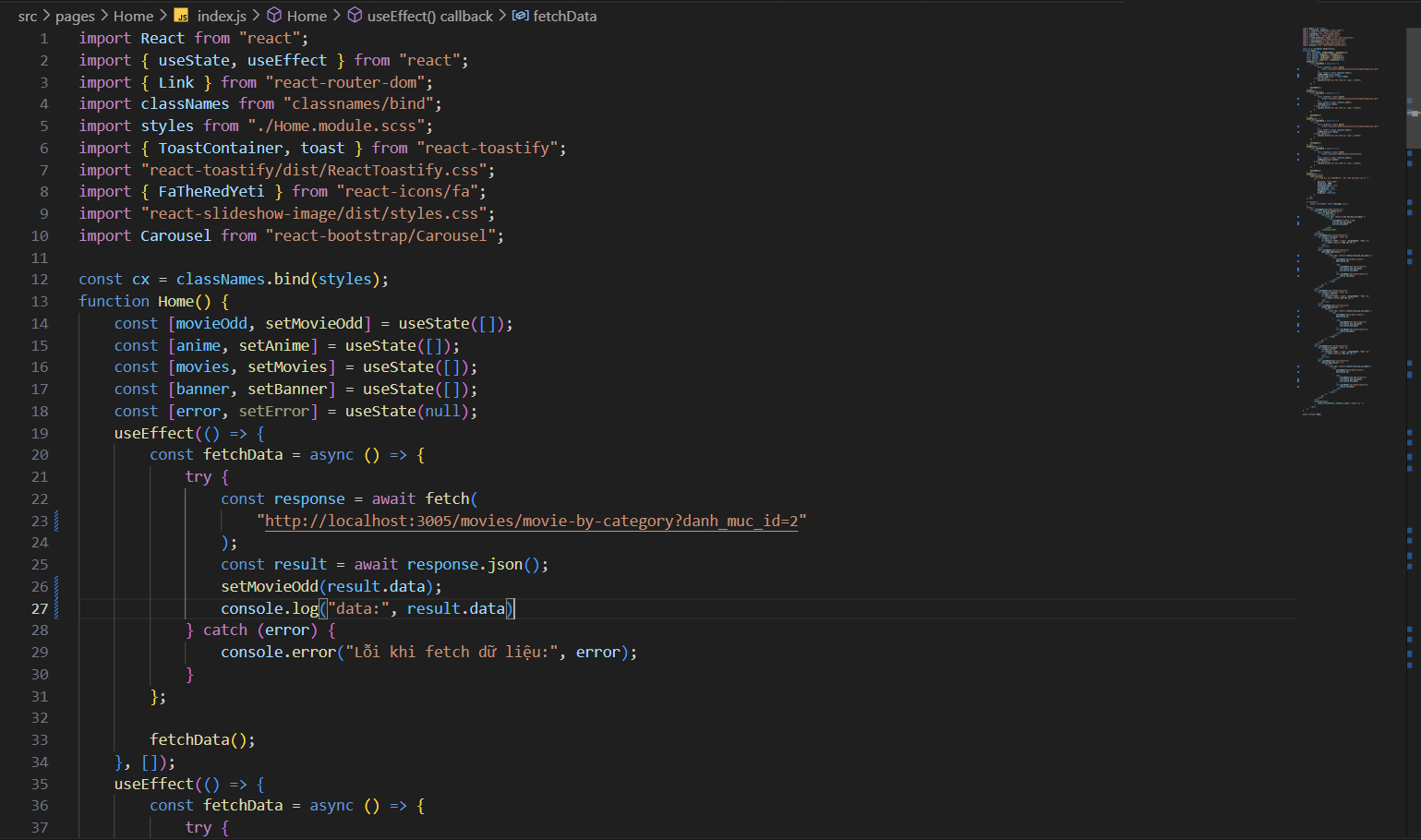
Hiển thị banner với các phim mới nhất trong một Carousel (dùng thư viện react-bootstrap).

Hiển thị danh sách phim theo 3 danh mục: Phim lẻ, Hoạt hình, Phim bộ.

Thông báo chào mừng khi người dùng vào trang (react-toastify).

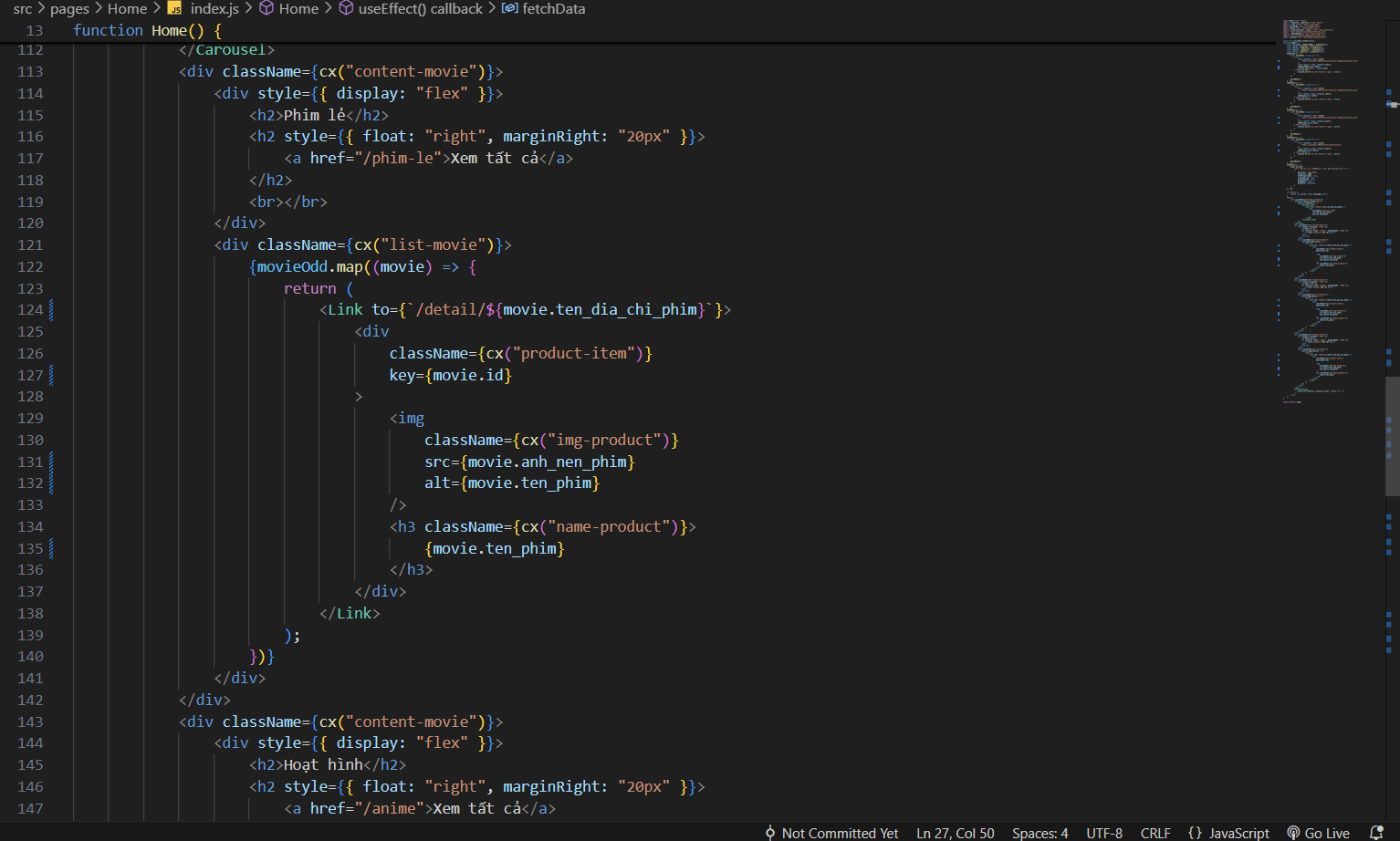
Sử dụng SCSS module (Home.module.scss) để tạo class động với classnames/bind.

Import các thư viện cần thiết để xây dựng trang chủ.



Hình 3: Import thư viện xây dựng trang chủ

Render dữ liệu các danh mục phim ra màn hình trang chủ



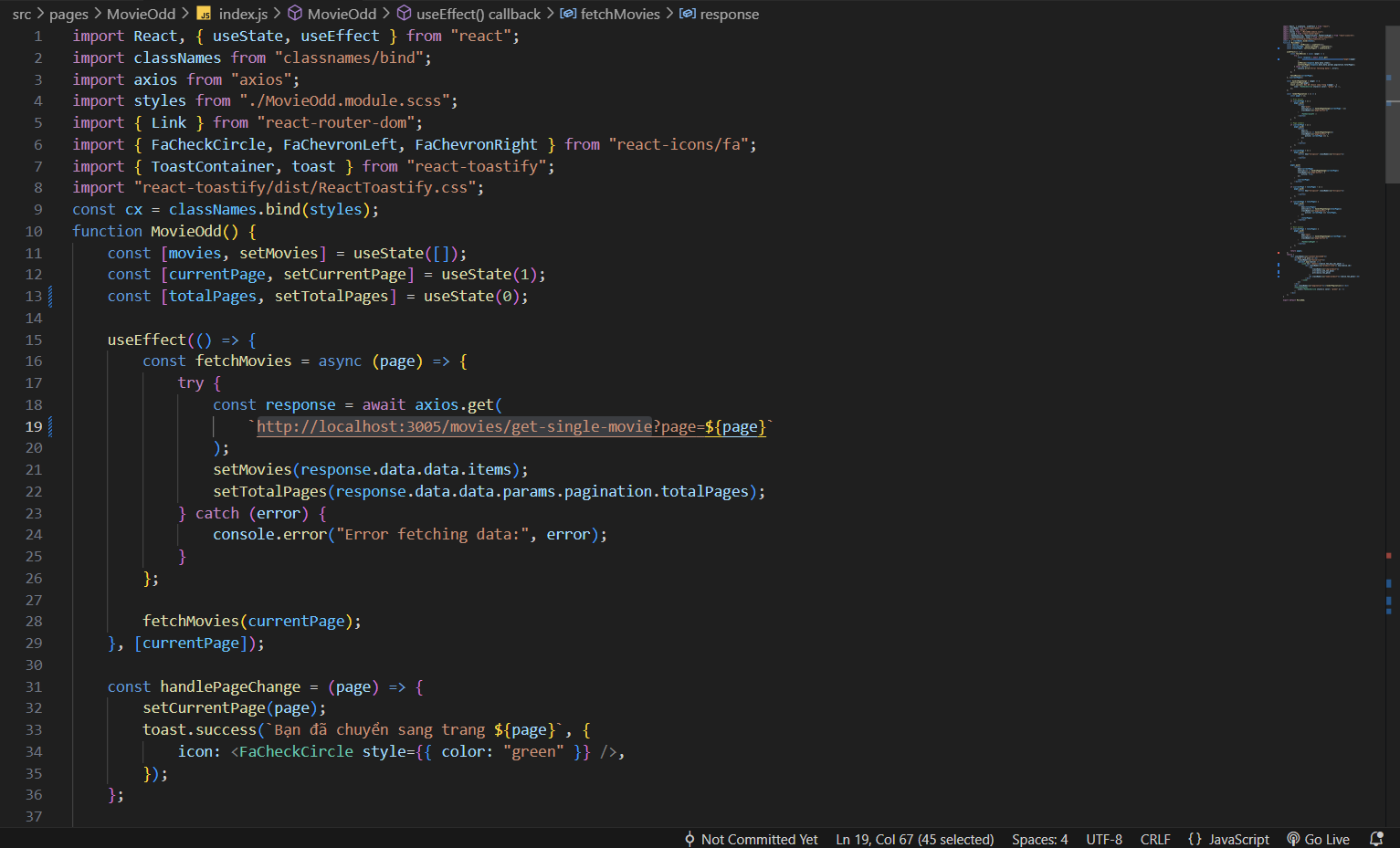
Hình 4: Render dữ liệu trang chủ

*Xây dựng trang phim lẻ:*

Mô tả: Trang phim lẻ vẫn sẽ có header và footer và sẽ render hiển thị ra danh sách các bộ phim trong danh mục phim lẻ bằng cách gọi API để lấy dữ liệu từ phía Back-end.

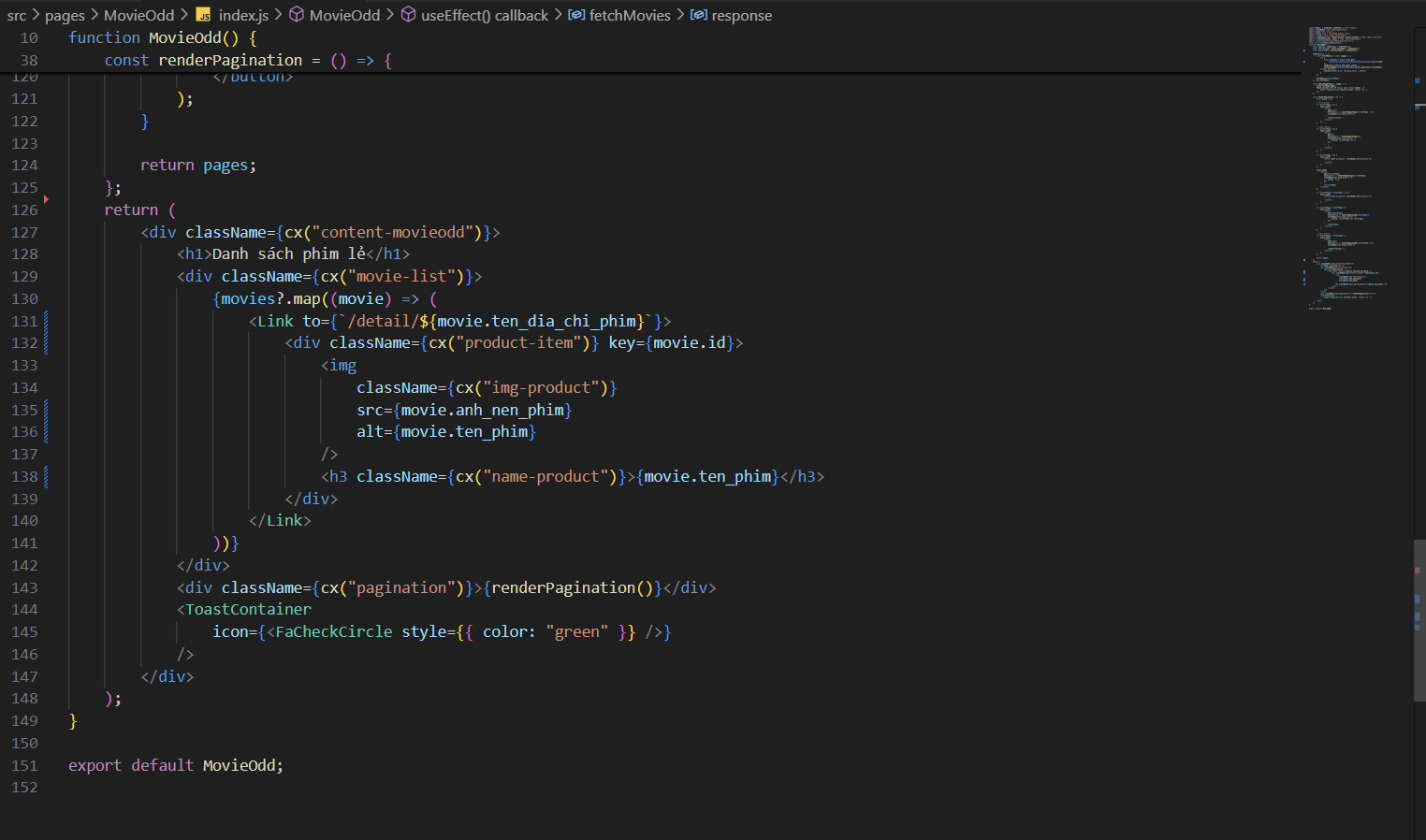
Import các thư viện cần thiết để xây dựng trang phim lẻ:

* useState, useEffect: Quản lý state và xử lý dữ liệu khi component mount.
* axios: Gửi request API lấy dữ liệu phim.
* classnames/bind: Hỗ trợ viết className dễ dàng hơn với CSS Modules.
* react-router-dom: Điều hướng đến trang chi tiết phim (Link).
* react-icons/fa: Dùng icon như FaCheckCircle, FaChevronLeft, FaChevronRight.
* react-toastify: Hiển thị thông báo khi chuyển trang.



Hình 5: Import thư viện xây dựng trang phim lẻ

Render dữ liệu danh sách phim lẻ ra trang phim lẻ



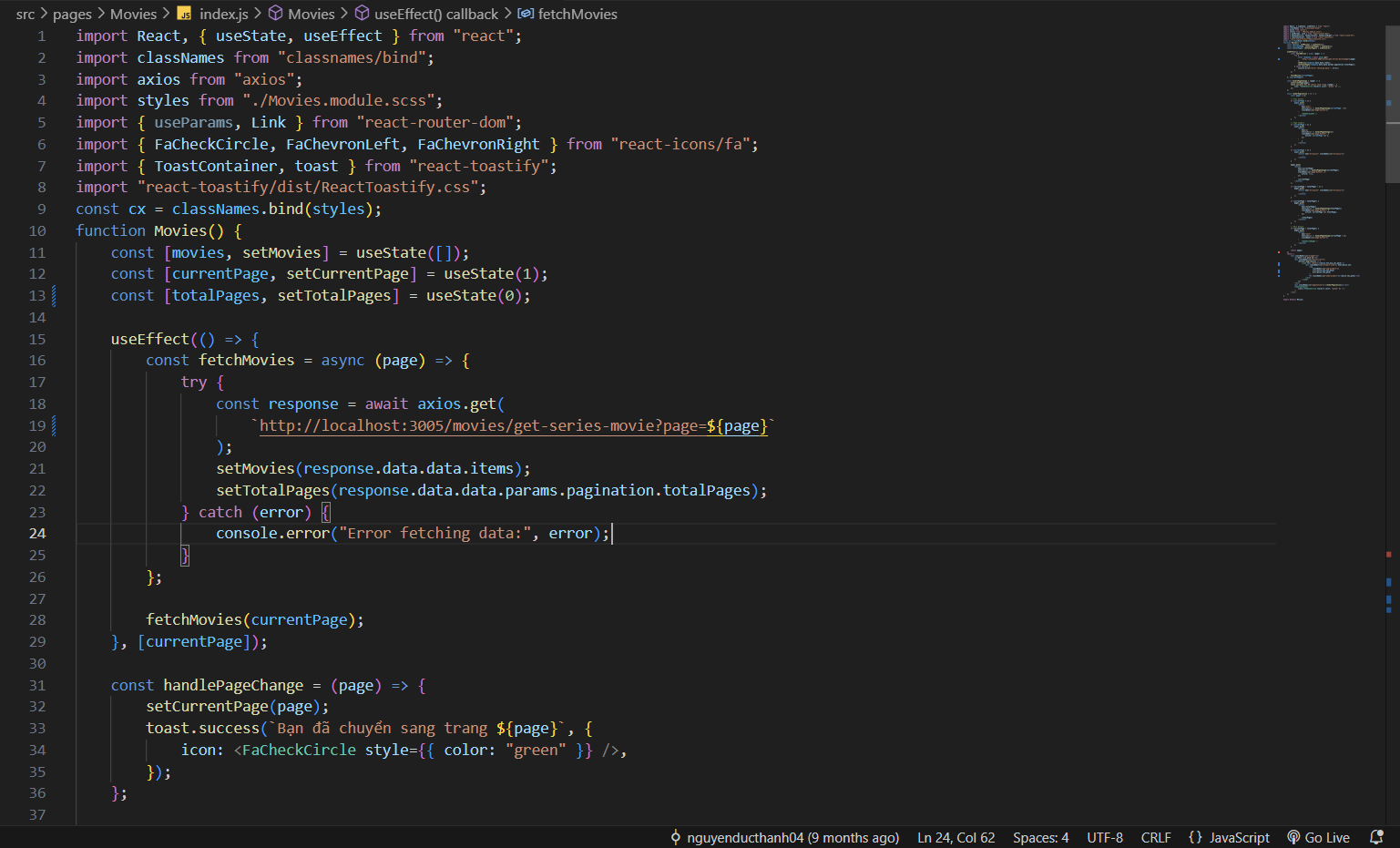
Hình 6: Render dữ liệu trang phim lẻ

*Xây dựng trang phim bộ:*

Mô tả: Trang phim lẻ vẫn sẽ có header và footer và sẽ render hiển thị ra danh sách các bộ phim trong danh mục phim bộ bằng cách gọi API để lấy dữ liệu từ phía Back-end.

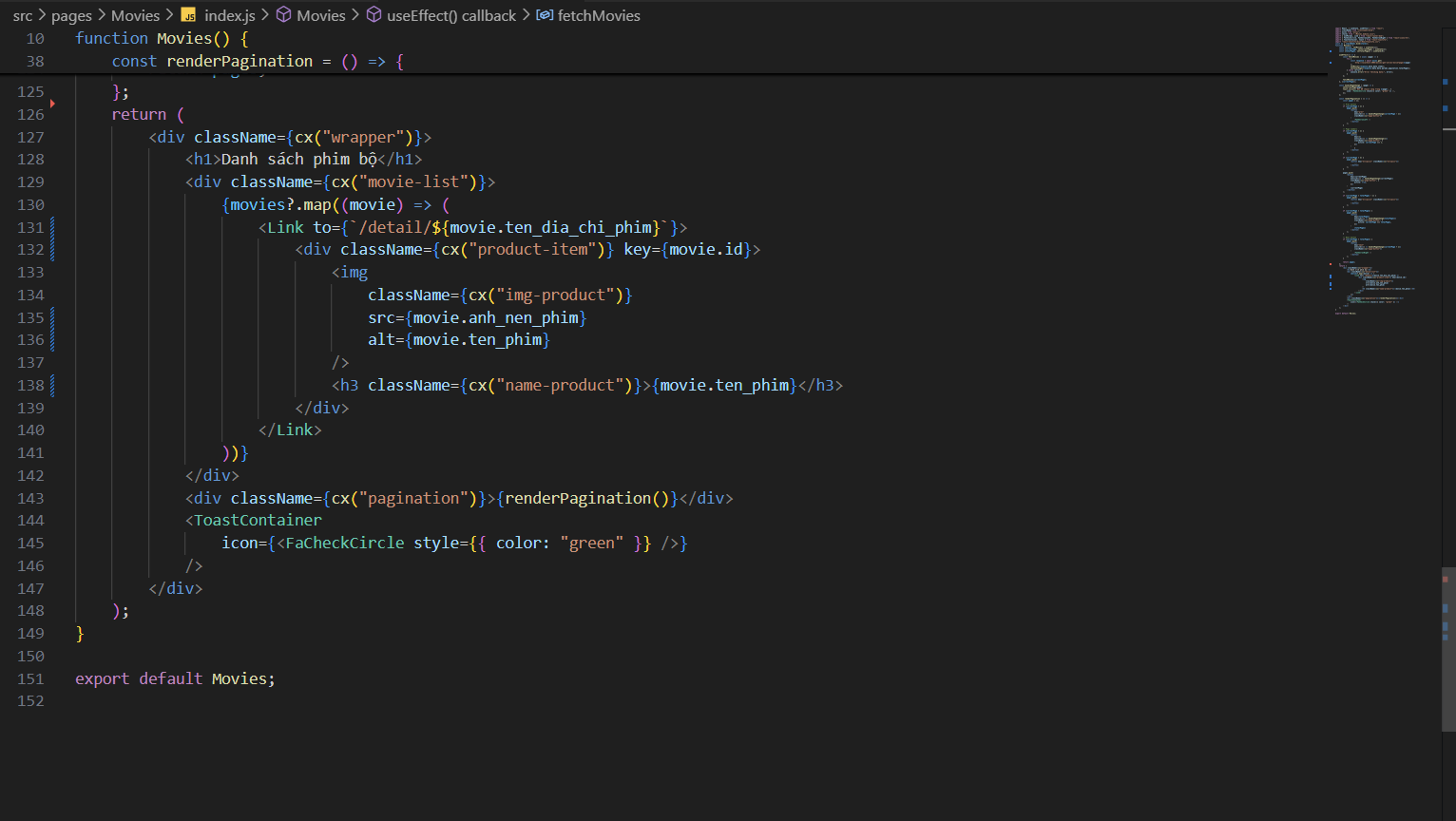
Import các thư viện cần thiết để xây dựng trang phim bộ:

* useState, useEffect: Quản lý state và xử lý dữ liệu khi component mount.
* axios: Gửi request API lấy dữ liệu phim.
* classnames/bind: Hỗ trợ viết className dễ dàng hơn với CSS Modules.
* react-router-dom: Điều hướng đến trang chi tiết phim (Link).
* react-icons/fa: Dùng icon như FaCheckCircle, FaChevronLeft, FaChevronRight.
* react-toastify: Hiển thị thông báo khi chuyển trang.

**

Hình 7: Import các thư viện cần thiết để xây dựng trang phim bộ

Render dữ liệu danh sách phim bộ ra trang phim bộ



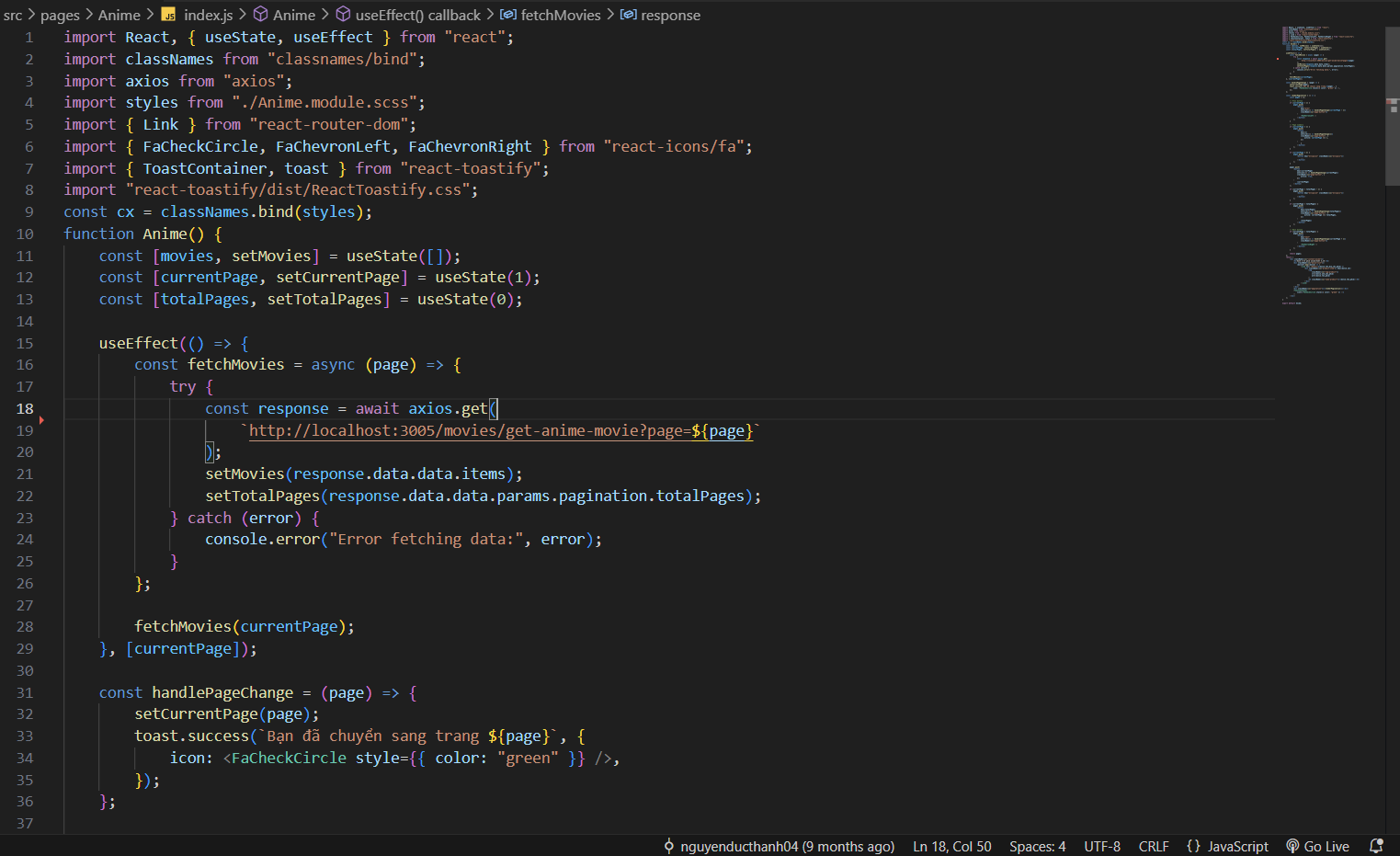
Hình 8: Render dữ liệu danh sách phim bộ ra trang phim bộ

*Xây dựng trang phim anime:*

Mô tả: Trang phim anime vẫn sẽ có header và footer và sẽ render hiển thị ra danh sách các bộ phim trong danh mục phim anime bằng cách gọi API để lấy dữ liệu từ phía Back-end.

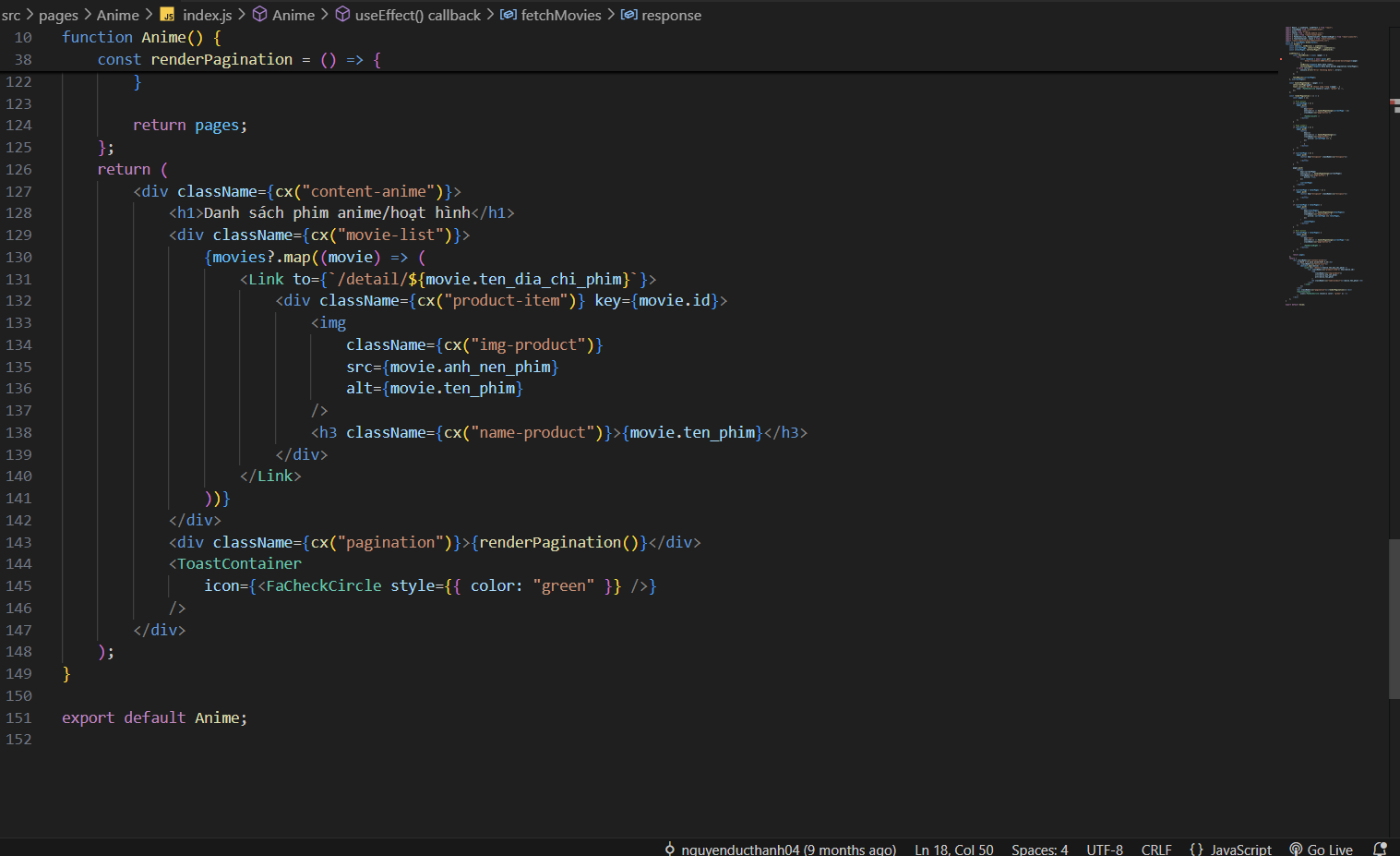
Import các thư viện cần thiết để xây dựng trang phim anime:

* useState, useEffect: Quản lý state và xử lý dữ liệu khi component mount.
* axios: Gửi request API lấy dữ liệu phim.
* classnames/bind: Hỗ trợ viết className dễ dàng hơn với CSS Modules.
* react-router-dom: Điều hướng đến trang chi tiết phim (Link).
* react-icons/fa: Dùng icon như FaCheckCircle, FaChevronLeft, FaChevronRight.
* react-toastify: Hiển thị thông báo khi chuyển trang.

**

Hình 9: Import các thư viện cần thiết để xây dựng trang phim anime

Render dữ liệu danh sách phim anime ra trang phim anime

**

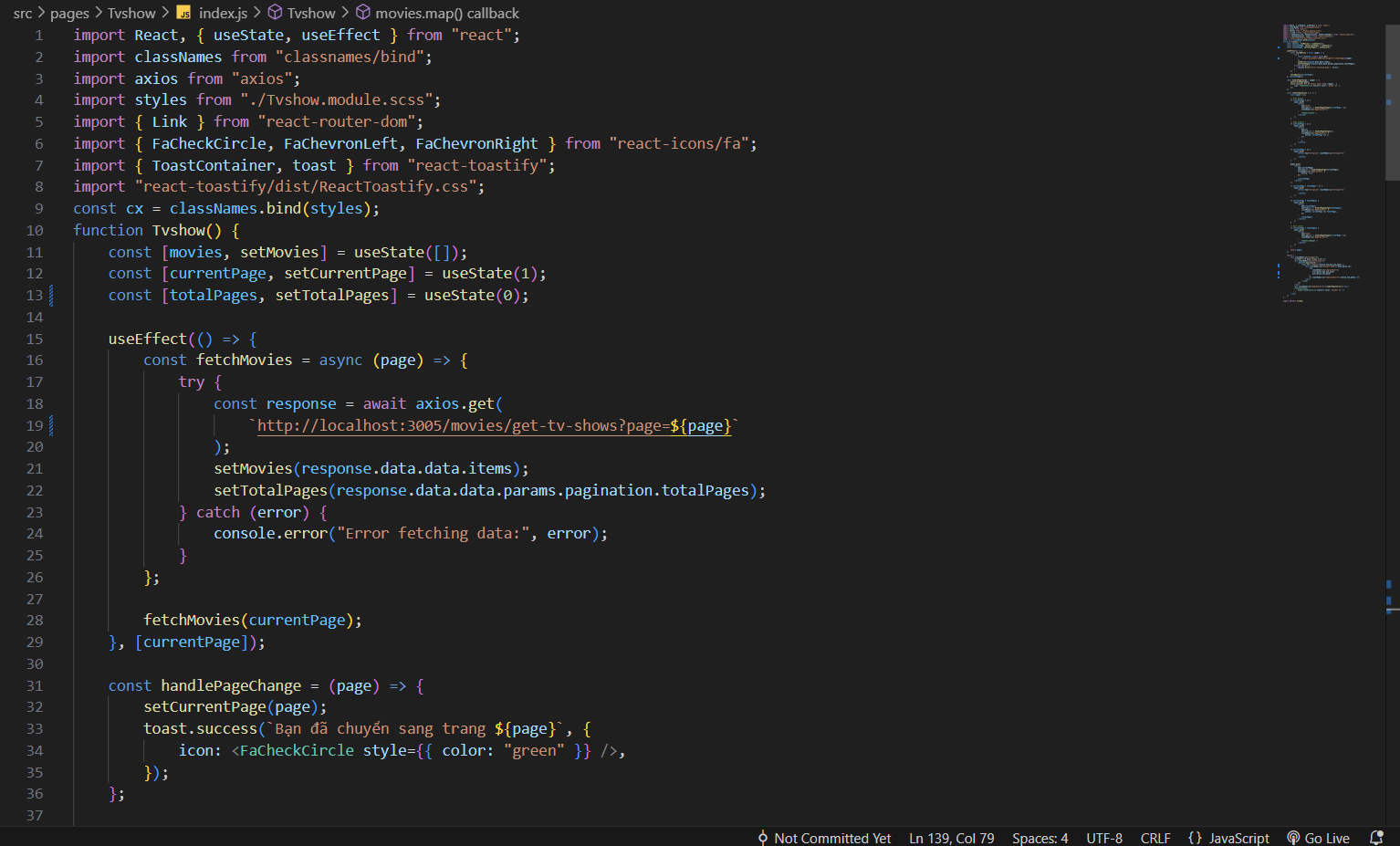
Hình 10: Render dữ liệu danh sách phim anime ra trang phim anime

*Xây dựng trang phim truyền hình:*

Mô tả: Trang phim truyền hình vẫn sẽ có header và footer và sẽ render hiển thị ra danh sách các bộ phim trong danh mục phim truyền hình bằng cách gọi API để lấy dữ liệu từ phía Back-end.

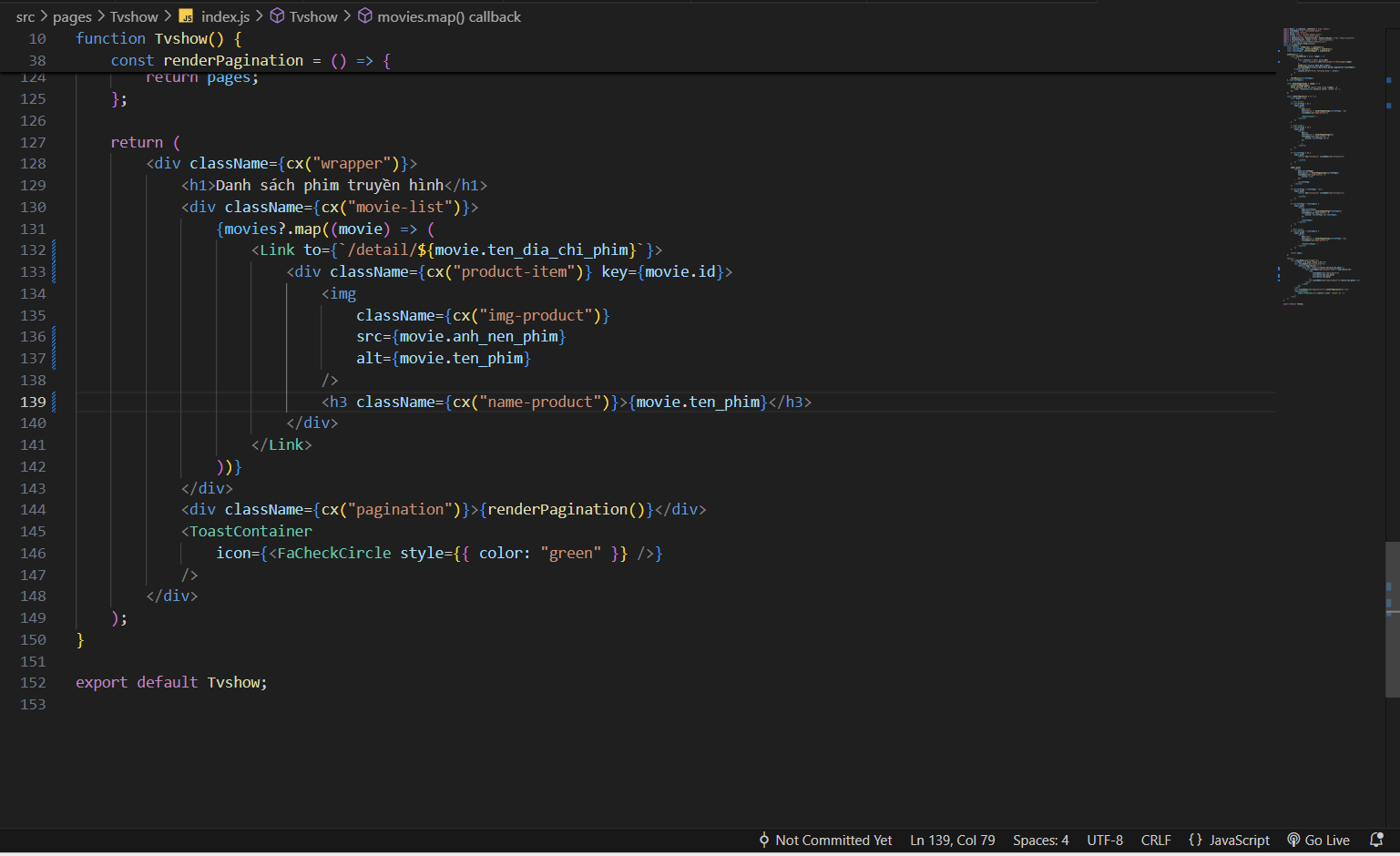
Import các thư viện cần thiết để xây dựng trang phim truyền hình:

* useState, useEffect: Quản lý state và xử lý dữ liệu khi component mount.
* axios: Gửi request API lấy dữ liệu phim.
* classnames/bind: Hỗ trợ viết className dễ dàng hơn với CSS Modules.
* react-router-dom: Điều hướng đến trang chi tiết phim (Link).
* react-icons/fa: Dùng icon như FaCheckCircle, FaChevronLeft, FaChevronRight.
* react-toastify: Hiển thị thông báo khi chuyển trang.



Hình 11: Import các thư viện cần thiết để xây dựng trang phim truyền hình

Render dữ liệu danh sách phim truyền hình ra trang phim truyền hình



Hình 12: Render dữ liệu danh sách phim truyền hình ra trang phim truyền hình

*Xây dựng trang phim yêu thích:*

Mô tả: Trang phim yêu thích vẫn sẽ có header và footer và sẽ render hiển thị ra danh sách các bộ phim trong danh mục phim yêu thích của người dùng bằng cách gọi API để lấy dữ liệu từ phía Back-end.

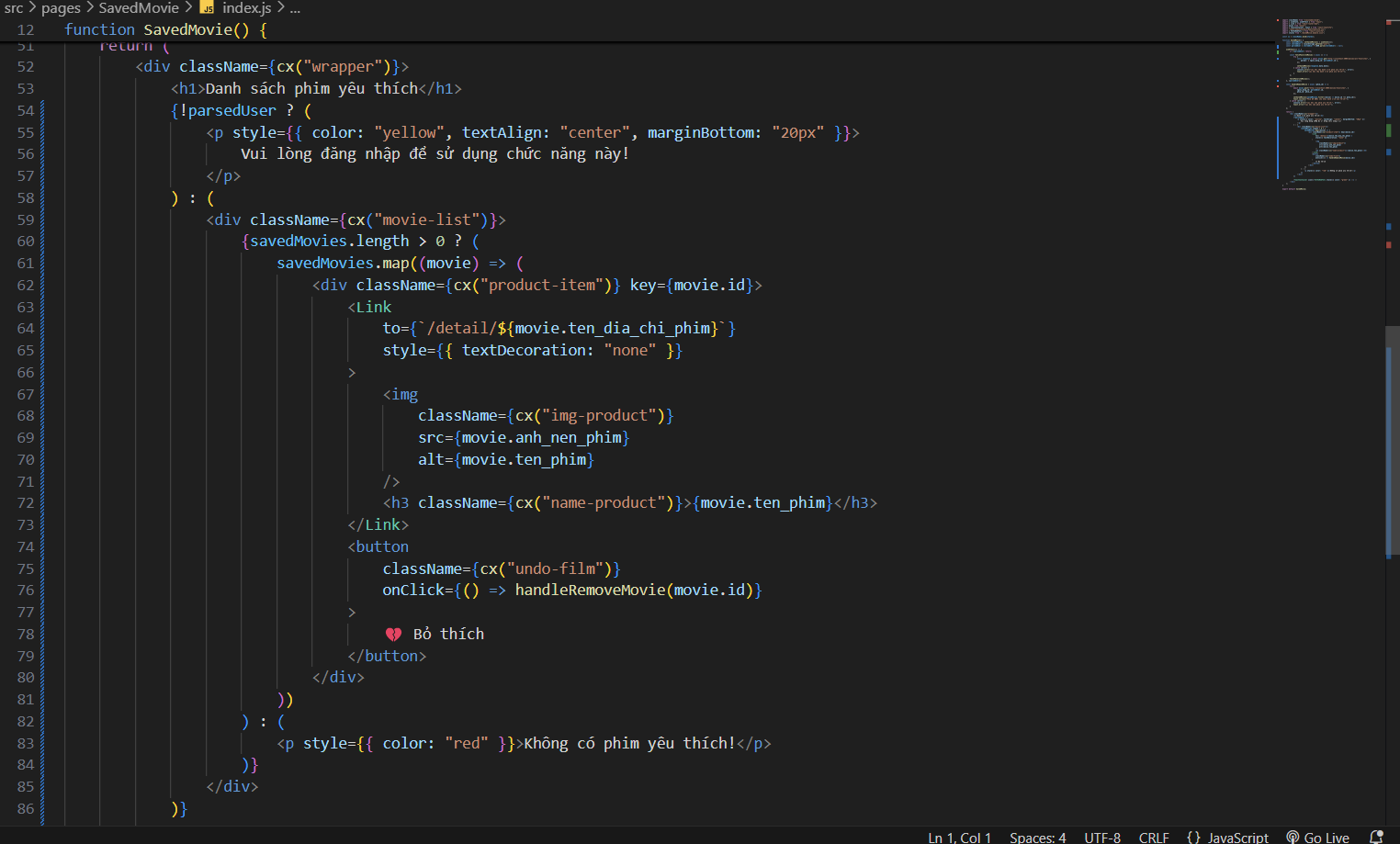
Import các thư viện cần thiết để xây dựng trang phim yêu thích:

* useState, useEffect: Quản lý state và xử lý dữ liệu khi component mount.
* axios: Gửi request API lấy dữ liệu phim.
* classnames/bind: Hỗ trợ viết className dễ dàng hơn với CSS Modules.
* react-router-dom: Điều hướng đến trang chi tiết phim (Link).
* react-icons/fa: Dùng icon FaTheRedYeti.
* react-toastify: Hiển thị thông báo khi chuyển trang.



Hình 13: Import các thư viện cần thiết để xây dựng trang phim yêu thích

Render dữ liệu danh sách phim yêu thích ra trang phim yêu thích



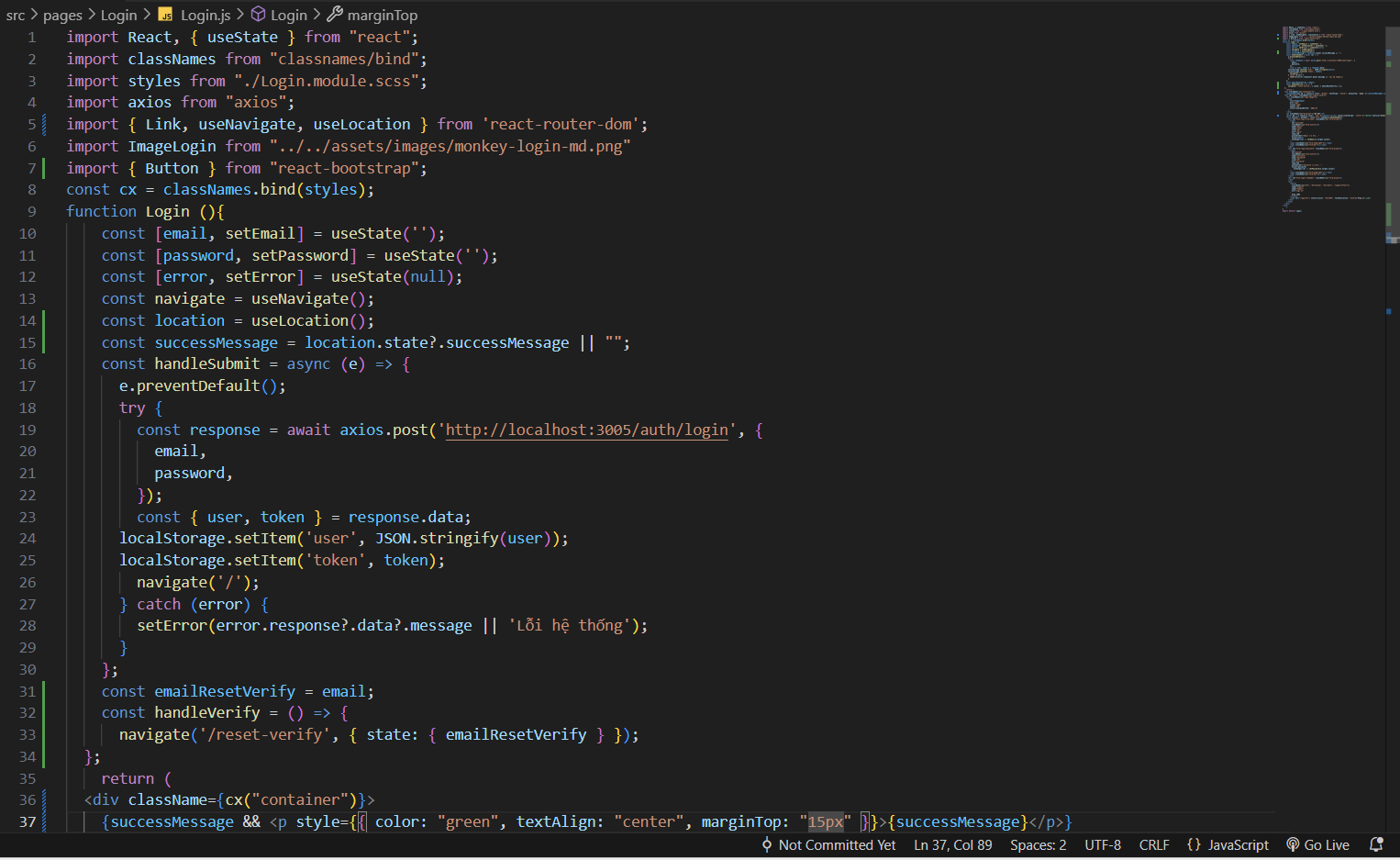
Hình 14: Render dữ liệu danh sách phim yêu thích ra trang phim yêu thích

*Xây dựng trang đăng nhập:*

Mô tả: Trang đăng nhập vẫn sẽ có header và footer và sẽ giúp cho người dùng đăng nhập vào website bằng cách gọi API Login để thực hiện Login từ phía Back-end.

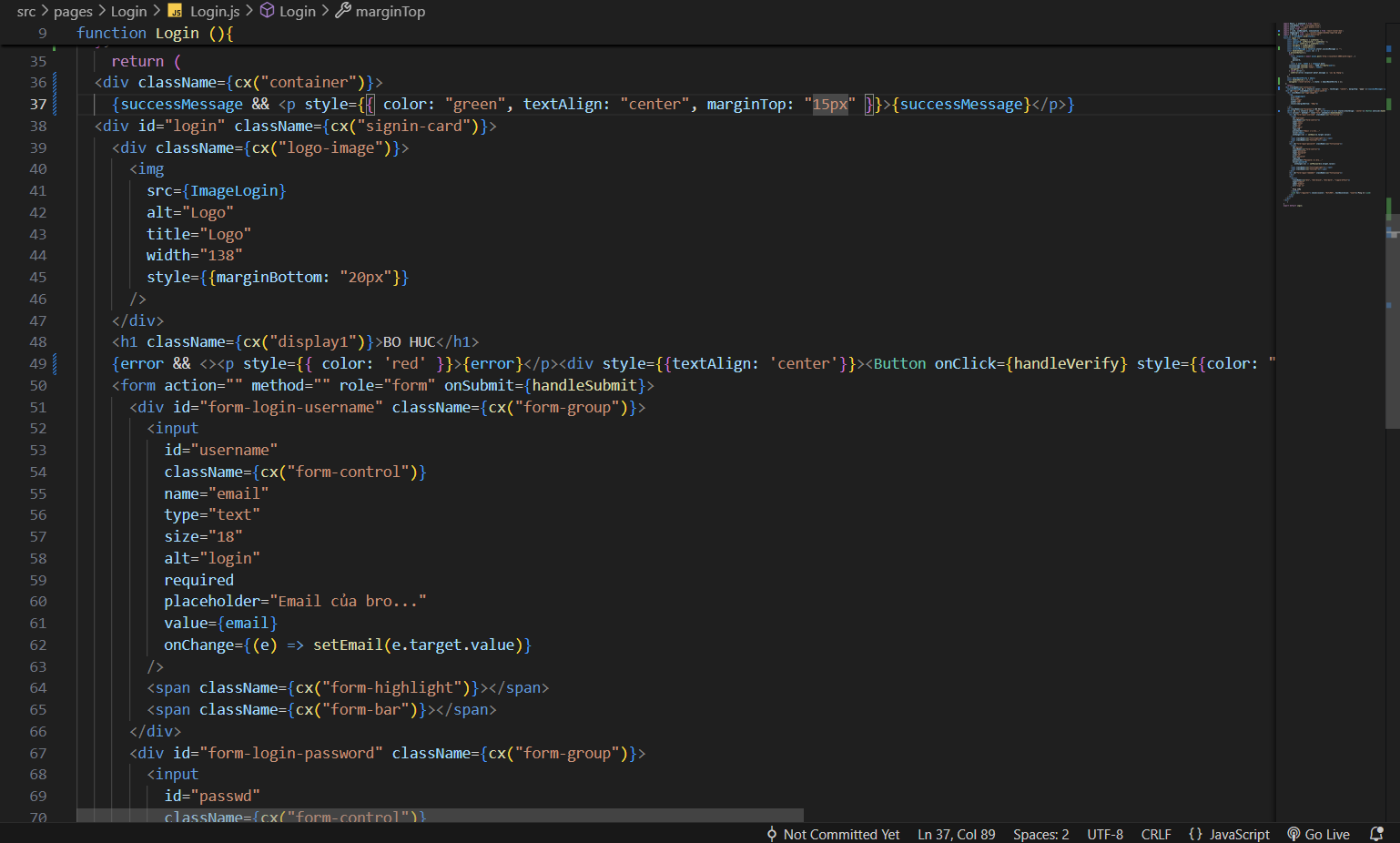
Import các thư viện cần thiết để xây dựng trang đăng nhập:

* React, useState: Cung cấp khả năng tạo component và quản lý state trong form đăng nhập.
* classnames/bind: Hỗ trợ viết className dễ dàng hơn với CSS Modules.
* axios: Gửi request API đến server để thực hiện đăng nhập.
* useNavigate(): Điều hướng người dùng sau khi đăng nhập thành công.
* useLocation(): Lấy dữ liệu từ trạng thái trước đó (ví dụ: thông báo thành công từ trang đăng ký).
* Link: Chuyển hướng đến trang đăng ký.
* react-bootstrap: Dùng Button để tạo nút "Xác thực tài khoản".
* SCSS Modules: Tăng khả năng tùy chỉnh giao diện và tổ chức CSS chuyên nghiệp.
* localStorage: Lưu thông tin người dùng và token sau khi đăng nhập thành công.



Hình 15: Import các thư viện cần thiết để xây dựng trang đăng nhập

Code giao diện form đăng nhập



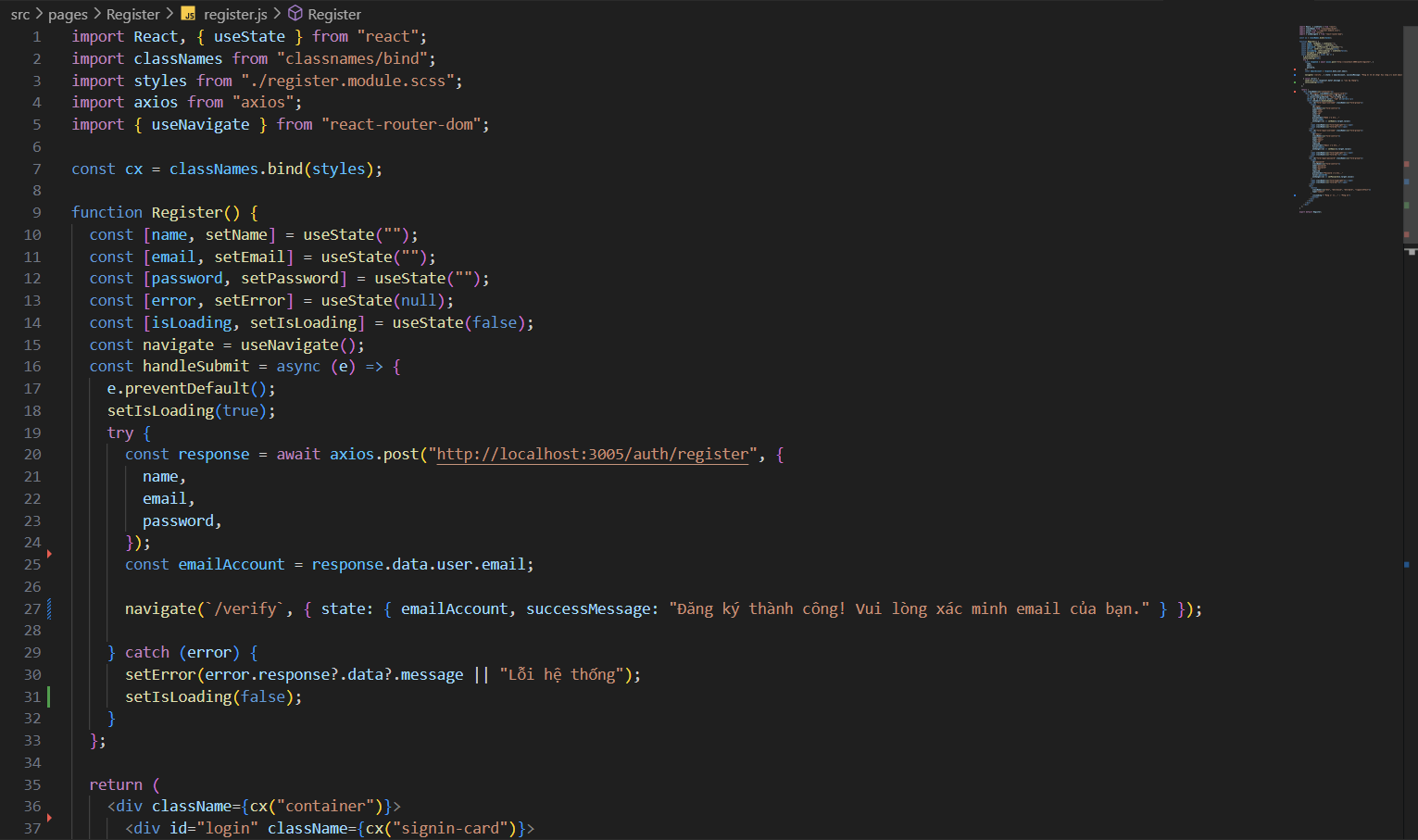
Hình 16: Code giao diện form đăng nhập

*Xây dựng trang đăng ký:*

Mô tả: Trang đăng ký vẫn sẽ có header và footer và sẽ giúp cho người dùng đăng ký tài khoản vào website bằng cách gọi API Register để thực hiện Register từ phía Back-end.

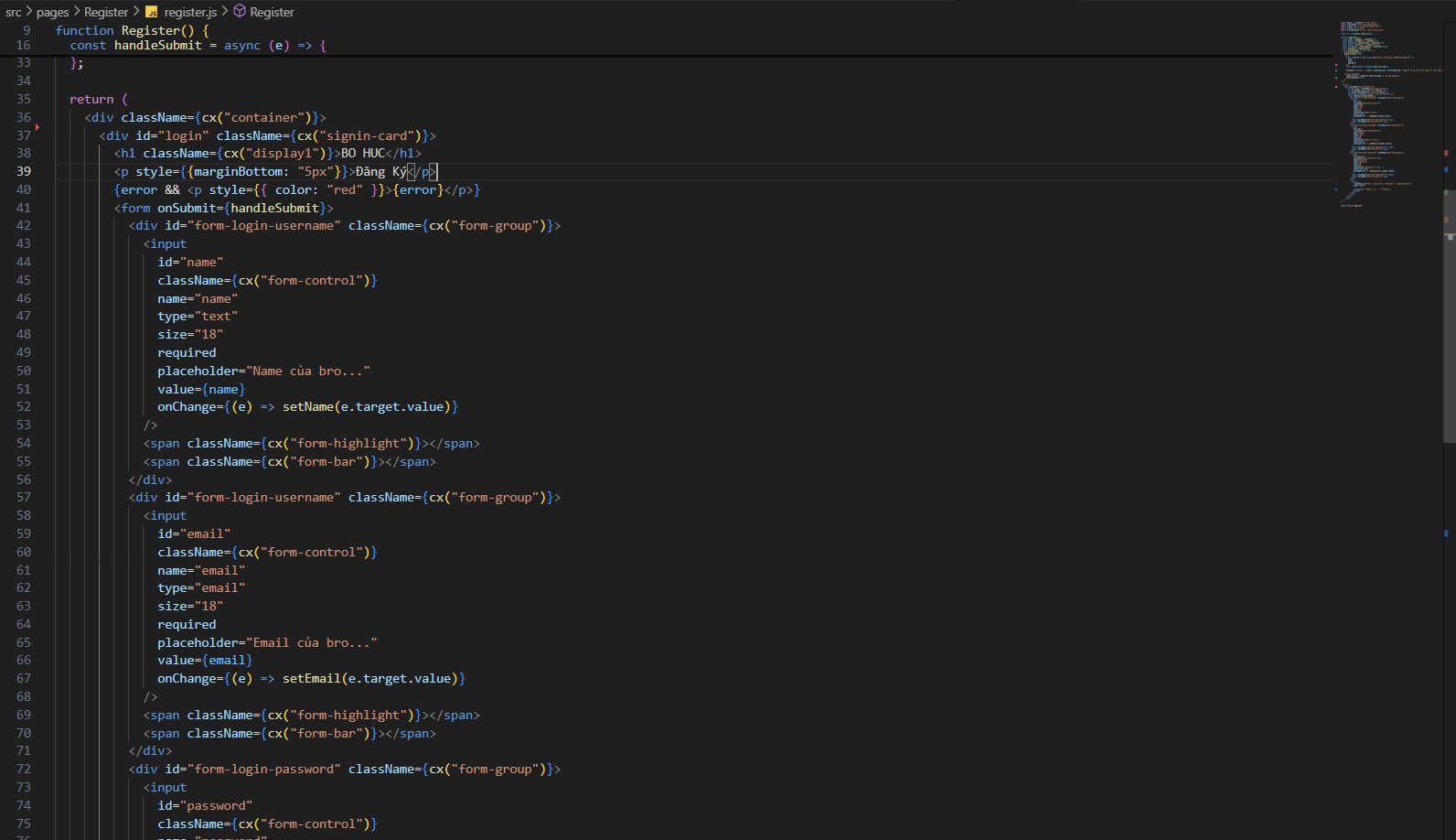
Import các thư viện cần thiết để xây dựng trang đăng ký:

* React, useState: Quản lý state của form đăng ký (tên, email, mật khẩu, lỗi, trạng thái loading).
* classnames/bind: Giúp viết className dễ dàng hơn với CSS Modules.
* axios: Gửi request API để tạo tài khoản mới.
* useNavigate(): Điều hướng người dùng đến trang xác minh email sau khi đăng ký thành công.
* SCSS Modules: Sử dụng để tạo styles cho giao diện đăng ký.
* localStorage: Không được sử dụng trong component này, nhưng có thể dùng để lưu token nếu cần.
* useState (isLoading): Trạng thái để hiển thị "Đang xử lý..." khi form đang gửi dữ liệu.

**

Hình 17: Import các thư viện cần thiết để xây dựng trang đăng ký

Code giao diện form đăng ký



Hình 18: Code giao diện form đăng ký

*Xây dựng giao diện trang hiển thị kết quả tìm kiếm:*

Mô tả: Trang hiển thị kết quả tìm kiếm vẫn sẽ có header và footer và sẽ render hiển thị ra danh sách các kết quả được tìm kiếm theo keyword bằng cách gọi API để lấy dữ liệu lọc dữ liệu tìm kiếm từ phía Back-end.

Import các thư viện cần thiết để xây dựng trang hiển thị kết quả tìm kiếm:

React, useState, useEffect:

* useState: Quản lý state của danh sách phim (movies) và trạng thái loading (loading).
* useEffect: Fetch dữ liệu phim từ API khi từ khóa tìm kiếm (keyword) thay đổi.

react-router-dom:

* useLocation(): Lấy URL hiện tại để trích xuất từ khóa tìm kiếm (keyword).
* Link: Điều hướng đến trang chi tiết phim (/detail/:id).

classnames/bind:

* Giúp quản lý classnames dễ dàng khi sử dụng SCSS Modules.

Axios:

* Gửi request đến API (http://127.0.0.1:3005/movies/search) để lấy danh sách phim dựa trên từ khóa.

react-icons (FaCheckCircle):

* Icon hiển thị trong thông báo (ToastContainer).

react-toastify:

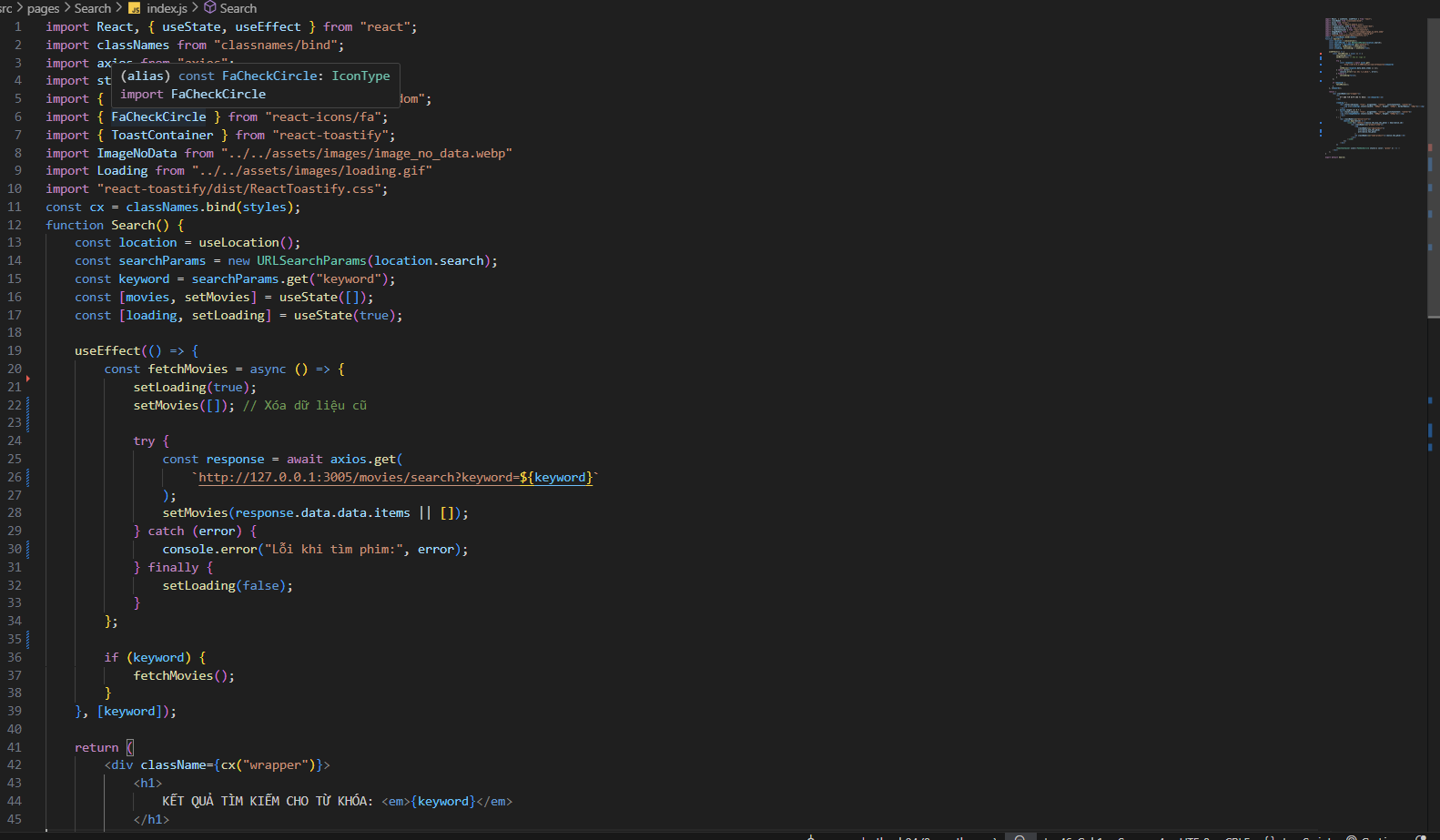
* Hiển thị thông báo người dùng khi có sự kiện xảy ra.

SCSS Modules:

* Tạo styles riêng cho component Search.

Hình ảnh assets (image\_no\_data.webp, loading.gif):

* Hiển thị ảnh khi không có kết quả hoặc khi dữ liệu đang tải.



Hình 19: Import các thư viện cần thiết để xây dựng trang hiển thị kết quả tìm kiếm

Hiển thị dữ liệu kết quả tìm kiếm



Hình 20: Hiển thị dữ liệu kết quả tìm kiếm