TRƯỜNG ĐẠI HỌC PHENIKA

KHOA CÔNG NGHỆ THÔNG TIN

Báo Cáo Dự Án Học Phần Đồ Án Cơ Sở

Đề tài: Website xem phim online

Thành viên

- **Nguyễn Như Bắc– 21012038**
- **Nguyễn Thái Việt Huy-21012500**
- **Nguyễn Lê Minh-21012884**
- **Đỗ Minh Quân-20010879**

Giảng viên hướng dẫn: **Vũ Minh Đức**

Hà Nội, tháng 11 năm 2023

Mục lục

LỜI MỞ ĐẦU

Trong thế giới số hóa ngày nay, việc xem phim trực tuyến đã trở thành một phần không thể thiếu của cuộc sống hàng ngày. Với sự tiện lợi và đa dạng về thể loại, các website xem phim trực tuyến đã mang đến cho người dùng trải nghiệm xem phim tuyệt vời ngay tại nhà. Bài báo cáo này sẽ khám phá và phân tích về các yếu tố quan trọng tạo nên sự thành công của các website xem phim trực tuyến, từ giao diện người dùng, chất lượng nội dung, đến cách thức quảng cáo và thu hút người dùng. Mục tiêu của bài báo cáo là cung cấp cái nhìn sâu sắc về ngành công nghiệp xem phim trực tuyến, đồng thời đưa ra những gợi ý và khuyến nghị để cải thiện và phát triển hơn nữa trong tương lai.

1. Giới thiệu

1.1 Đặt vấn đề

Dữ liệu lớn và phức tạp: Quản lý một cơ sở dữ liệu lớn với thông tin về hàng ngàn phim, diễn viên, đạo diễn và nhiều loại dữ liệu khác có thể là một thách thức.

Bản quyền và cấp phép: Thu thập thông tin chi tiết về phim, bao gồm cả hình ảnh và video, có thể liên quan đến vấn đề bản quyền và cần phải có cơ chế cấp phép phù hợp.

Tương tác người dùng: Xây dựng trải nghiệm người dùng một cách mượt mà, tránh giật lag, giao diện ưu nhìn để không cảm thấy nhàm chán

Tính tương thích và đa nền tảng: Đảm bảo trang web hoạt động trơn tru trên nhiều trình duyệt và thiết bị khác nhau là một thách thức kỹ thuật.

Tối ưu hóa tốc độ trang web: Làm thế nào để đảm bảo rằng trang web hoạt động nhanh và mượt, không làm mất khách hàng vì lỗi trang hoặc tải trang chậm?

Dữ liệu liên kết và cập nhật liên tục: Liên tục cập nhật và duy trì dữ liệu, bao gồm cả thông tin về các bộ phim mới và thông tin về nghệ sĩ, để đảm bảo trang web luôn cung cấp thông tin chính xác và mới nhất.

- **Tìm kiếm và lọc thông tin:** Cung cấp cơ chế tìm kiếm mạnh mẽ và khả năng lọc thông tin sẽ giúp người dùng dễ dàng tìm kiếm và xem thông tin mà họ quan tâm**.**
- **Tích hợp xã hội:** Kết nối với các nền tảng mạng xã hội để người dùng có thể chia sẻ thông tin về phim và tương tác với cộng đồng một cách dễ dàng.
- **Responsive Design:** thiết kế và triển khai trang web để nó có thể hoạt động mượt mà trên cả máy tính và điện thoại di động?
- **Đa ngôn ngữ và văn hóa:** Xây dựng một giao diện hỗ trợ nhiều ngôn ngữ và văn hóa sẽ làm cho trang web hấp dẫn đối với một đối tượng rộng lớn người dùng trên toàn thế giới.

1.2 Các giải pháp đã có

- **Thiết kế giao diện và trải nghiệm người dùng:** Áp dụng các nguyên tắc thiết kế UX/UI để đảm bảo trải nghiệm tốt cho người xem.
- **Phân trang và Tải trước:** Sử dụng React Router để phân trang giữa các danh sách phim. Tối ưu hóa hình ảnh và sử dụng kỹ thuật tải trước để giảm thời gian tải trang.
- **Xử lý video:** Sử dụng thư viện React Video để phát video một cách linh hoạt. Tích hợp các API như YouTube API hoặc Vimeo API để phát video từ các nguồn bên ngoài.
- **Tìm kiếm và Lọc Phim:** Sử dụng thư viện ReactJS như React-Select để xây dựng giao diện tìm kiếm và lọc đa dạng.
- **Đề xuất Phim:** Sẽ đề xuất phim theo từng danh mục mà tất cả người dùng đã xem trước đó.
- **Thẻ và Danh mục Phim:** Sử dụng Redux để quản lý trạng thái của các thẻ và danh mục. Tạo các component tái sử dụng cho thẻ và danh mục để giảm lặp code.
- **Bảo mật mạnh mẽ**: Sử dụng mã hóa SSL, kiểm soát truy cập và theo dõi các vấn đề bảo mật để bảo vệ thông tin người dùng và dữ liệu trang web.
- **Responsive Design:** Sử dụng media queries và responsive design để tối ưu hóa trang web cho các thiết bị di động.

1.3 Giải pháp đề xuất

- **Tích hợp Dịch vụ Phim Nổi tiếng:** Sử dụng API từ các dịch vụ như IMDb hoặc TMDB để lấy thông tin chi tiết và đánh giá. Hiển thị thông tin này một cách thân thiện và nhất quán trên trang web của bạn.
- **Đa Ngôn ngữ:** Sử dụng thư viện react-i18next để hỗ trợ đa ngôn ngữ và quản lý chuỗi ngôn ngữ.

- **Chia sẻ và Bình luận:** Sử dụng các API chia sẻ xã hội như ShareThis để tích hợp chức năng chia sẻ.
- **Tích hợp đa ngôn ngữ**: Hỗ trợ nhiều ngôn ngữ và văn hóa để mở rộng đối tượng người dùng trên toàn thế giới.
- **Chế độ Tối và Chế độ Ban Ngày:** Sử dụng thư viện React Context để quản lý trạng thái chế độ tối/ban ngày. Tích hợp cơ chế lưu trữ cục bộ để lưu trạng thái chế độ tối/ban ngày của người dùng.
- **Tích hợp xã hội**: Kết nối với các nền tảng mạng xã hội và cung cấp tính năng chia sẻ để tăng sự tương tác và lan truyền thông tin.
- **Đánh giá và Đánh giá Phim:** Sử dụng thư viện như React Rating để tạo giao diện đánh giá dễ sử dụng.
- **Hỗ trợ Nền tảng đa phương tiện:** Sử dụng thư viện như react-player để tích hợp video từ nhiều nguồn khác nhau. Kết hợp với các API như YouTube API hoặc Vimeo API để phát video từ các dịch vụ phổ biến.
- **Tối ưu hóa hiệu suất**: Sử dụng kỹ thuật tối ưu hóa hình ảnh, tải dữ liệu theo đợt, và sử dụng các Content Delivery Network (CDN) để cải thiện tốc độ tải trang.

1.4 Giới thiệu một vài công nghệ sử dụng chính

ReactJS

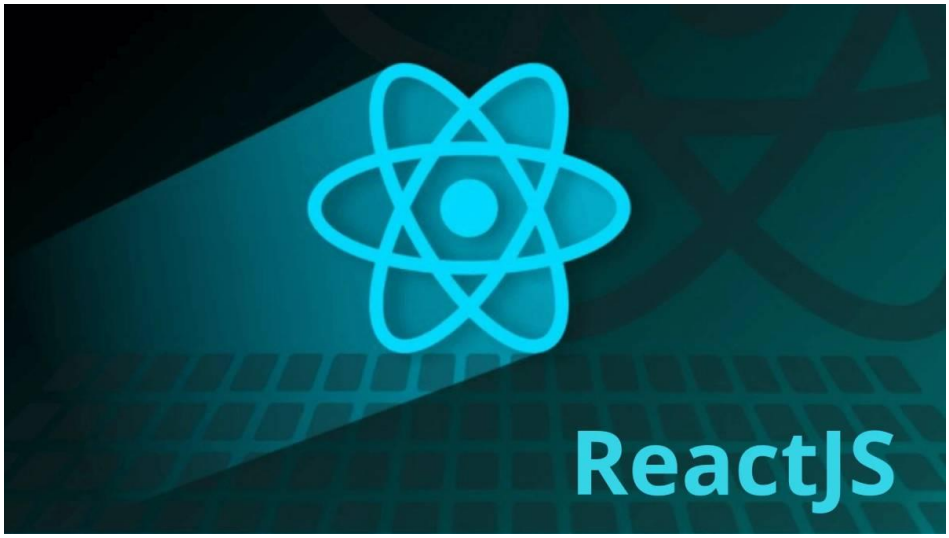
ReactJS là thư viện Javascript được xây dựng bởi các kỹ sư của Facebook, đang được rất nhiều công ty nổi tiếng sử dụng để phát triển các sản phẩm của họ như Yahoo, Airbnb và tất nhiên là trong nội tại Facebook, Instagram. Nó phù hợp với các dự án lớn có tính mở rộng hơn là các dự án nhỏ.

Tư tưởng ReactJS là xây dựng lên các components có tính tái sử dụng, dễ dàng cho việc chia nhỏ vấn đề, testing. Nó giúp chúng ta dễ dàng quản lý, mở rộng

hệ thống, điều này nếu là Angular thì đòi hỏi cấu trúc, cách viết code phải thật tối ưu.ReactJS luôn giữ các components ở trạng thái stateless khiến ta dễ dàng quản lý bởi nó chẳng khác gì một trang HTML tĩnh. Bản thân các components này không có trạng thái (state) nó nhận đầu vào từ bên ngoài và chỉ hiện thị ra dựa vào các đầu vào đó, điều này lý giải tại sao nó lại mang tính tái sử dụng và dễ dàng cho việc bảo trì.

ReactJS là một thư viện hiển thị view chú ý đến hiệu năng(performance-minded). Rất nhiều đối thủ nặng ký về framework MVVM (Model-View-ViewModel)mất một thời gian lớn để hiển thị những lượng dữ liệu lớn, như trong trường hợp hiển thị những danh sách và tương tự. Nhưng React đó không còn là vấn đề, vì nó chỉ hiển thị những gì thay đổi.Một trong những điểm mạnh nữa của ReactJS là virtual DOM - thứ nằm ẩn bên trong mỗi view và là lí do khiến cho React đạt được hiệu năng tốt. Khi một view yêu cầu gọi, tất cả mọi thứ sẽ được đưa vào trong một bản sao ảo của DOM. Sau khi việc gọi hoàn thành, React tiến hành một phép so sánh giữa DOM ảo và DOM thật, và thực hiện những thay đổi được chỉ ra trong phép so sánh trên.

Ví dụ: nếu chúng ta đang xem một danh sách có 20 sản phẩm được hiển thị bởi ReactJS, và chúng ta thay đổi sản phẩm thứ 2, thì chỉ sản phẩm đó sẽ được hiển thị lại,và 19 sản phẩm còn lại vẫn giữ nguyên (không cần hiển thị lại hay reload lại trang).React đã dùng cái gọi là “virtual DOM” để tăng hiệu năng bằng cách xuất ra một hiển thị ảo, sau đó kiểm tra sự khác biệt giữa hiển thị ảo và những gì có trên DOM và tạo một bản vá.



Giới thiệu về Redux

Khi làm việc với React hay các dự án ứng dụng Single Page nói chung, có một vấn đề khá đau đầu là làm sao quản lý được trạng thái của ứng dụng đó. Sau khi xem qua giới thiệu về một số thư viện hỗ trợ công việc này, em đã quyết định chọn Redux.

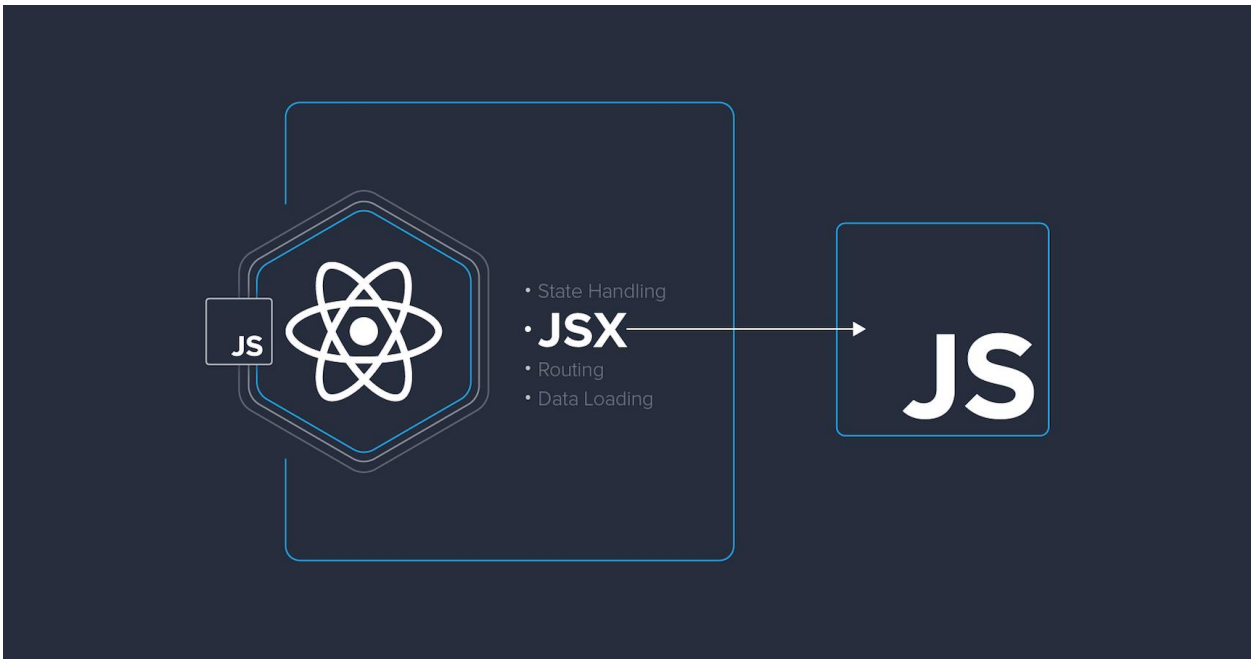
Redux là một thư viện Javascript giúp tạo ra thành một lớp quản lý trạng thái của ứng dụng.Redux được xây dựng dựa trên nền tảng tư tưởng của ngôn ngữ Elm và kiến trúc Flux do Facebook giới thiệu. Do vậy Redux thường là bộ đôi kết hợp hoàn hảo với React. Tuy nhiên hoàn toàn có thể sử dụng với các framework khác như Angular,Backbone.

ReactJS là một thư viện Javascript giúp xây dựng giao diện người dùng, được xây dựng xung quanh các component (thành phần) nhỏ ghép lại với nhau, có hai kiểu dữ liệu trong React đó là state và props. State là trạng thái, mang tính private, chỉ có thể thay đổi ở trong chính component. Còn props mang tính external, không bị kiểm soát bởi component, được truyền từ component cao (cha) đến component thấp (con).

ReactJS xây dựng lên các single-page-app, tức là chỉ render ra 1 trang, và tất cả các thành phần của ứng dụng sẽ được lưu trữ trong đó. Vì thế, nếu ứng dụng phức tạp lên theo thời gian, việc quản lý state của chúng ta sẽ ngày một lớn dần.

Sự khác biệt với việc sử dụng Redux và không sử dụng Redux được miêu tả bằng hình vẽ dưới đây:

Giới thiệu về JSX



JSX là một cú pháp mở rộng cho JavaScript, là kết hợp của JavaScript và XML. Nó chuyển đổi cú pháp dạng gần như XML về thành JavaScript, giúp chúng ta có thể code ReactJS bằng cú pháp của XML thay vì phải dùng JavaScript.

Thay vì phải truy xuất đến một thẻ HTML bất kì thông qua các thuộc tính như id, class, name... thì ta đã có thể thoải mái sử dụng các thẻ HTML trong code JavaScript rồi. Tất cả những gì ta cần làm đó là viết 1 đống thẻ html và sau đó đặt chúng vào trong một biến, một function hoặc một class chẳng hạn.

Ví dụ:

```
const name = 'Josh Perez';
const element = <h1>Hello, {name}</h1>;

ReactDOM.render(
  element,
  document.getElementById('root')
);
```

Giới thiệu react-router-dom:

React-router-dom là một thư viện React được sử dụng để quản lý các route trong ứng dụng React. Nó cung cấp các thành phần giúp bạn định nghĩa các route và quản lý việc chuyển đổi giữa chúng một cách dễ dàng. Thư viện này là một phần của hệ sinh thái react-router, cung cấp các giải pháp cho việc điều hướng và quản lý URL trong ứng dụng React.

Dưới đây là một số đặc điểm quan trọng của react-router-dom:

- BrowserRouter và HashRouter:
- BrowserRouter: Sử dụng HTML5 History API để theo dõi các thay đổi trên URL.

- HashRouter: Sử dụng hash URL để theo dõi các thay đổi.
- Route:

Dùng để định nghĩa một route trong ứng dụng React. Mỗi Route có một đường dẫn và một component tương ứng để hiển thị khi đường dẫn này được truy cập.

- Link và NavLink:

Link: Dùng để tạo các liên kết giữa các route mà không làm tải lại toàn bộ trang.

NavLink: Tương tự như Link, nhưng cung cấp thêm các tính năng như xác định một phong cách CSS khi route được chọn.

- Switch:

Switch là một component giúp bạn bao bọc các Route và đảm bảo chỉ có một Route đầu tiên khớp với URL được render. Điều này giúp tránh việc hiển thị nhiều components cùng một lúc.

- Tham số Đường dẫn và Tham số Truy vấn (Params và Query):

Bạn có thể định nghĩa tham số trong đường dẫn và truy vấn URL để truyền dữ liệu giữa các components.

- Redirect:

Dùng để chuyển hướng người dùng từ một router khác.

Để sử dụng react-router-dom, bạn cần cài đặt nó thông qua npm hoặc yarn:

npm install react-router-dom hoặc

yarn add react-router-dom

Giới thiệu Axios:

Axios là một thư viện HTTP Client dựa trên Promise dành cho node.js và trình duyệt. Nó có tính đẳng hình (tức là cùng codebase có thể chạy trong cả trình duyệt và node.js). Ở phía server thì nó sử dụng native module http trong node.js, còn ở phía client (trình duyệt) thì nó sử dụng XMLHttpRequest.

Tính năng:

- Tạo request từ trình duyệt bằng XMLHttpRequest
- Tạo request từ node.js bằng http
- Hỗ trợ Promise API
- Đón chặn request và response
- Biến đổi dữ liệu request và response
- Bãi bỏ request
- Tự động chuyển đổi cho dữ liệu JSON
- Hỗ trợ phía client bảo vệ chống lại XSRF

- **Tool Vite**

Các công cụ xây dựng Javascript có thể gây phiền toái đối với các developers, đặc biệt là khi tất cả những gì bạn muốn làm là tập trung vào phát triển. Các developers muốn có một cách đơn giản hóa để xây dựng và phát triển các ứng dụng của họ và Vite.js là một trong những công cụ phổ biến giải quyết vấn đề này.

Trong phần này, chúng ta sẽ nói về các tính năng chính mà Vite cung cấp và lý do tại sao bạn nên bắt đầu sử dụng nó. Vite.js, là một công cụ xây dựng phát triển giao diện người dùng hiện đại, cung cấp một số tính năng chính phân biệt nó với các trình bundling truyền thống. Đây là một số tính năng:

- Máy chủ phát triển siêu nhanh: Vite.js giới thiệu một máy chủ phát triển được tối ưu hóa cao, tận dụng việc nhập khẩu native ES module trên trình duyệt hiện đại. Nó sử dụng phương pháp biên dịch theo yêu cầu, cho phép thay thế module nhanh chóng (HMR) gần như tức thì và tải lại trang nhanh chóng. Điều này đáng kể tăng tốc quá trình phát triển, cung cấp chu trình phản hồi nhanh hơn và nâng cao năng suất của nhà phát triển.
- Hỗ trợ Module ES gốc: Vite.js hoàn toàn hướng tới hỗ trợ Module ES gốc, được hỗ trợ một cách tự nhiên trong các trình duyệt hiện đại. Trong quá trình phát triển, Vite.js tận dụng khả năng tải các Module ES thành các tệp riêng biệt mà không cần phải bundle chúng lại. Phương pháp này loại bỏ việc bundle dữ liệu trong quá trình phát triển, dẫn đến thời gian khởi động nhanh hơn và khả năng lưu trữ dữ liệu tốt hơn. Nó cũng cho phép trình duyệt song song hóa việc tải các module, làm tăng hiệu suất tổng thể.
- Blazing-Fast Production Builds: Vite.js sử dụng trình bundling “esbuild”, được biết đến với tốc độ vượt trội, trong quá trình xây dựng sản xuất. “esbuild” tạo ra các bundle mã được tối ưu hóa và thu nhỏ, dẫn đến giảm đáng kể thời gian xây dựng so với các trình đóng gói truyền thống. Quá trình bundling nhanh

chóng này nâng cao hiệu quả của developer và cho phép chu kỳ triển khai nhanh hơn.

- Không cần cấu hình: Vite.js tuân theo triết lý không cần cấu hình, mang đến một trải nghiệm ngay lập tức ngay sau khi cài đặt. Bằng cách giảm thiểu việc cần phải cấu hình thủ công, developer có thể nhanh chóng thiết lập dự án mới mà không phải tốn thời gian vào các cấu hình phức tạp. Tuy nhiên, Vite.js cũng cung cấp một tệp cấu hình linh hoạt (vite.config.js) để tùy chỉnh cao cấp khi cần thiết.
- Tích hợp Devtool: Vite.js tích hợp liền mạch với các công cụ phát triển trình duyệt phổ biến. Nó cung cấp trải nghiệm gỡ lỗi nâng cao bằng cách ánh xạ mã nguồn gốc đến trình duyệt, cho phép developers gỡ lỗi trực tiếp mã của họ mà không cần bất kỳ thiết lập hoặc công cụ bổ sung nào.
- Hệ sinh thái Plugin: Vite.js có một hệ sinh thái plugin đang phát triển mở rộng chức năng và tích hợp với các framework front -end phổ biến như Vue.js, React và Preact. Các plugin này nâng cao trải nghiệm phát triển và cung cấp các tính năng bổ sung, tối ưu hóa và tích hợp với các công cụ và thư viện.

- **Vercel**

Vercel là một platform tốt cho phép bạn build, preview và deploy websites một cách dễ dàng cho những trang web được tạo bởi kiến trúc JavaScript như: Next.js, Gatsby.js, React, Vue.js, Nuxt, Angular,...

Vercel có một số tính năng miễn phí như:

- Cải thiện hiệu suất trang web với Edge Network.
- Không giới hạn website và API.
- Có thể custom domain với đầy đủ chức năng SSL.
- Có cung cấp Serverless Functions.

- 100GB bandwidth/tháng
- Tự động tối ưu hoá hình ảnh(1000 image cho phiên bản free)

- **APIs**

API là các phương thức, giao thức kết nối với các thư viện và ứng dụng khác. Nó là viết tắt của Application Programming Interface – giao diện lập trình ứng dụng. API cung cấp khả năng truy xuất đến một tập các hàm hay dùng. Và từ đó có thể trao đổi dữ liệu giữa các ứng dụng.

API sử dụng mã nguồn mở, dùng được với mọi client hỗ trợ XML, JSON.

API có khả năng đáp ứng đầy đủ các thành phần HTTP: URI, request/response headers, caching, versioning, content forma…. Bạn có thể sử dụng các host nằm trong phần ứng dụng hoặc trên IIS.

Mô hình web API dùng để hỗ trợ MVC như: unit test, injection, ioc container, model binder, action result, filter, routing, controller. Ngoài ra, nó cũng hỗ trợ RESTful đầy đủ các phương thức như: GET, POST, PUT, DELETE các dữ liệu.

Được đánh giá là một trong những kiểu kiến trúc hỗ trợ tốt nhất với các thiết bị có lượng băng thông bị giới hạn như smartphone, tablet…

Ưu điểm:

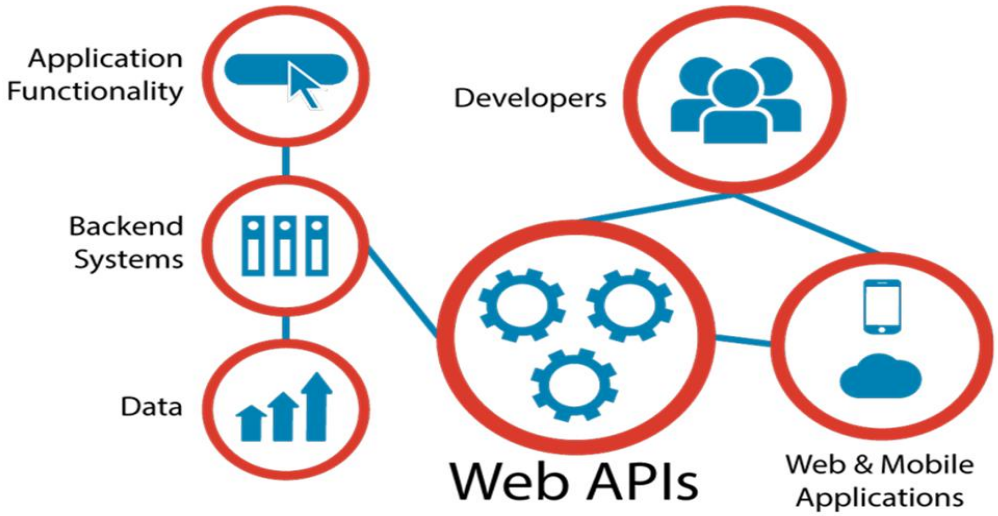
- Giao tiếp hai chiều phải được xác nhận trong các giao dịch sử dụng API.
- Cũng chính vì vậy mà các thông tin rất đáng tin cậy.
- API là công cụ mã nguồn mở, có thể kết nối mọi lúc nhờ vào Internet.
- Hỗ trợ chức năng RESTful một cách đầy đủ.
- Cấu hình đơn giản khi được so sánh với WCF (Window Communication Foundation). Cung cấp cấp trải nghiệm thân thiện với người dùng.

Nhược điểm:

-Tốn nhiều chi phí phát triển, vận hành, chỉnh sửa.

-Đòi hỏi kiến thức chuyên sâu.

-Có thể gặp vấn đề bảo mật khi bị tấn công hệ thống.



- **SCSS**

SCSS được biết đến là sử dụng cú pháp giống với Ruby. Bởi lẽ nó được thiết kế bởi chính xác lập trình viên Ruby. Với phần mở rộng là .scss, nó được cho ra đời sau SASS và có cú pháp viết tương tự như cách viết CSS thông thường. Tuy nhiên, cú pháp này lại có thể thu hẹp khoảng cách giữa SASS và CSS bằng cách mang lại môi trường thân thiện.

Nói một cách dễ hiểu, SCSS được coi như một bản nâng cấp của SASS khi viết CSS. Nó đem tới cho người dùng môi trường làm việc thân thiện, dễ hiểu hơn. Đây chính là lý do mặc dù khá giống nhau nhưng SCSS lại được lựa chọn. Khi bạn hỏi về SCSS là gì. Người ta vẫn sẽ giải đáp từ SASS sau đó mới đến nền tảng này. Bởi lẽ chúng luôn đi song song và có liên quan đến nhau, bổ trợ cho nhau. Các lập trình viên lâu năm hoàn toàn có thể sử dụng cả hai nền tảng này để viết CSS.

1.5 Các tính năng cơ bản của SCSS

SCSS đem đến những tính năng tuyệt vời dành cho người dùng. Với những tính năng này, chúng ta đều sẽ thấy được điểm khác biệt của nó với SASS.

1.5.1 Xếp chồng (Nested Rules) – SCSS là gì?

- Đây chính là một tính năng được đánh giá cao của SCSS. Tính năng này được sử dụng rất thường xuyên trong việc lập trình. Các bạn có thể tham khảo những đoạn code có dùng tính năng này để nhận ra sự khác biệt.

1.5.2 Biến (Variable):

- Sử dụng biến với SCSS vô cùng đơn giản và dễ hiểu. Các bạn chỉ cần đặt tên cho biến và bắt đầu bằng $. Biến chứa đựng các giá trị mà người dùng sử dụng nhiều lần. Ví dụ như các mã màu, font chữ, kiểu chữ,…

1.5.3 Quy tắc Mixin – SCSS là gì?

- Mixin là một trong những tính năng phải nói đến khi giải đáp SCSS là gì. Điều này giúp cho người dùng tạo ra các hàm sử dụng được trong nền tảng này. Từ đó người dùng có thể truyền các tham số vào bên trong nó để sử dụng. Mixin được biết đến là một cơ chế khá phổ biến trong SASS. Công dụng mà Mixin đem lại chính là mang nhiều thuộc tính đã quy ước vào một thành phần bất kỳ.

1.5.4 Kế thừa (Extends):

- Khi nhắc đến tính năng này, người dùng sẽ nghĩ ngay đến OOP. Lập trình hướng đối tượng với các class được lập sắn. Những tag cần thì người dùng chỉ việc @extend class vào là hoàn chỉnh.

1.5.5 Import:

- Tính năng này của SCSS cũng có trên cả SASS. Cú pháp này vô cùng hữu dụng và thường xuyên được dùng trong các project. Nó giống như việc bạn require hay include file vào các file khác.

2. Thiết kế và triển khai

2.1 Các yêu cầu chức năng

- **Tìm kiếm và Duyệt Phim:**
  Công cụ tìm kiếm mạnh mẽ với khả năng lọc và sắp xếp.
  Trình duyệt phim dựa trên nhiều tiêu chí như thể loại, năm sản xuất, đạo diễn, và diễn viên.
- **Đánh Giá:**
  Đánh giá rating từ người xem.
- **Thông Tin Chi Tiết:**
  Trang thông tin chi tiết về mỗi phim, bao gồm diễn viên, đạo diễn, thể loại, ngày phát sóng, đánh giá, và hình ảnh.
- **Tương Thích và Đa Nền Tảng:**
  Tích hợp đa nền tảng, từ máy tính đến thiết bị di động.
  Tương thích với nhiều trình duyệt web khác nhau.
- **Bảng Xếp Hạng :**
  Bảng xếp hạng các phim phổ biến và được đánh giá cao.
- **Cập Nhật Liên Tục:**
  Cập nhật thông tin mới về phim, diễn viên, và movieTV,..
- **An Toàn và Bảo Mật:**
  Bảo vệ thông tin người dùng và đảm bảo an toàn về mặt bảo mật.

2.2 Các yêu cầu phi chức năng

- **Hiệu Năng:**

Trang web phải phản hồi nhanh chóng và không có độ trễ lớn khi người dùng thực hiện các thao tác.

Hỗ trợ một lượng lớn người dùng đồng thời.

- **Bảo Mật:**

  Cung cấp các biện pháp bảo mật như mã hóa dữ liệu người dùng và giải pháp chống tấn công.

- **Đồng Bộ Hóa Dữ Liệu:**

  Đảm bảo dữ liệu giữa các thành phần khác nhau của hệ thống được đồng bộ hóa và không có sự mất mát thông tin.

- **Khả Năng Mở Rộng:**

  Dự tính và triển khai cơ sở hạ tầng để có thể mở rộng hệ thống khi có nhu cầu.

- **Tương Thích Nền Tảng:**

  Hỗ trợ nhiều trình duyệt và hệ điều hành khác nhau để đảm bảo rằng nhiều người dùng có thể truy cập trang web một cách thuận lợi.

- **Tương Thích Trình Duyệt:**

  Đảm bảo trang web hoạt động mượt mà và có giao diện thân thiện trên các thiết bị di động.

- **Tuân Thủ Pháp Luật:**

  Đảm bảo rằng trang web tuân thủ các quy định pháp luật liên quan, đặc biệt là về bản quyền nội dung và bảo vệ thông tin cá nhân.

2.3 Các ràng buộc (Constraints)

2.3.1 Các ràng buộc về triển khai

- **Ràng Buộc Tương Thích:**

Ứng dụng cần hoạt động được trên nhiều thiết bị và trình duyệt khác nhau

- **Ràng Buộc Giao Tiếp:**

  Tiêu Chuẩn Giao Tiếp: Sử dụng các tiêu chuẩn giao tiếp nhất định (ví dụ: RESTful API).

- **Ràng Buộc Hiệu Suất:**

  Ứng dụng cần đảm bảo khả năng phát video mượt mà, không giật lag, đặc biệt khi số lượng người dùng truy cập cùng lúc lớn.

- **Ràng Buộc Môi Trường:**

  Khả năng Mở Rộng: Hệ thống phải có khả năng mở rộng khi cần thiết.

  Duyệt Di Động: Phải tương thích và thân thiện với các thiết bị di động.

2.3.2 Các ràng buộc kinh tế

- **Ràng Buộc Tài Nguyên:**

  Việc phát triển và duy trì ứng dụng cần tài nguyên nhất định

2.3.3 Các ràng buộc về đạo đức

- **Ràng Buộc Pháp Luật và Chuẩn Mực:**

Bản Quyền: Tuân thủ các quy định bản quyền và sử dụng dữ liệu theo quy định pháp luật.

An Toàn Thông Tin: Bảo vệ thông tin cá nhân theo các chuẩn mực an ninh.

- **Ràng Buộc Bảo mật:**

Ứng dụng cần bảo vệ thông tin cá nhân của người dùng, như thông tin tài khoản, lịch sử xem phim, v.v.

2.4 Mô hình hệ thống / Thiết kế giải pháp

Các công nghệ sử dụng chính:

- **Client-Side (Frontend):**

  Sử dụng ReactJS để xây dựng giao diện người dùng.

  Các thành phần UI như danh sách phim, chi tiết phim, tìm kiếm, và giao diện người dùng khác.

  Sử dụng thư viện quản lý trạng thái như Redux để quản lý trạng thái toàn cầu.

- **Server-Side (Backend):**

  Một server backend để xử lý các yêu cầu từ frontend và gọi API để lấy dữ liệu phim.

  Quản lý người dùng, đánh giá, và các tính năng khác.

- **APIs:**

  Gọi các API từ các nguồn dữ liệu bên ngoài để lấy thông tin về phim (ví dụ: The Movie Database API).

  Xây dựng các API nội bộ để xử lý các yêu cầu từ frontend.

- **Communication with API:**

  Sử dụng thư viện như Axios để thực hiện các yêu cầu HTTP đến API.

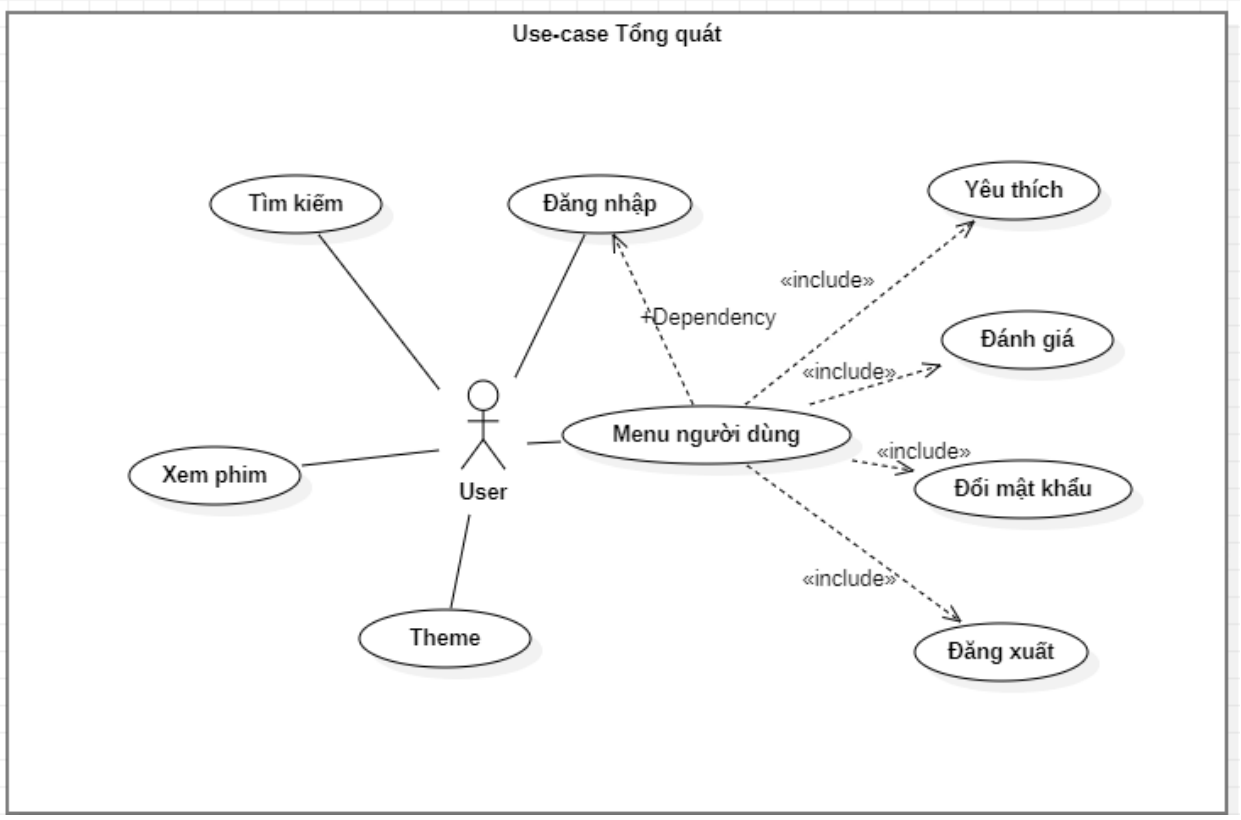
  Xử lý các trạng thái như loading, thành công, và lỗi khi gọi API.

- **Caching and Optimizations:**

  Sử dụng caching để giảm thời gian tải dữ liệu và cải thiện hiệu suất.

  Tối ưu hóa hình ảnh và tài nguyên để giảm dung lượng trang.

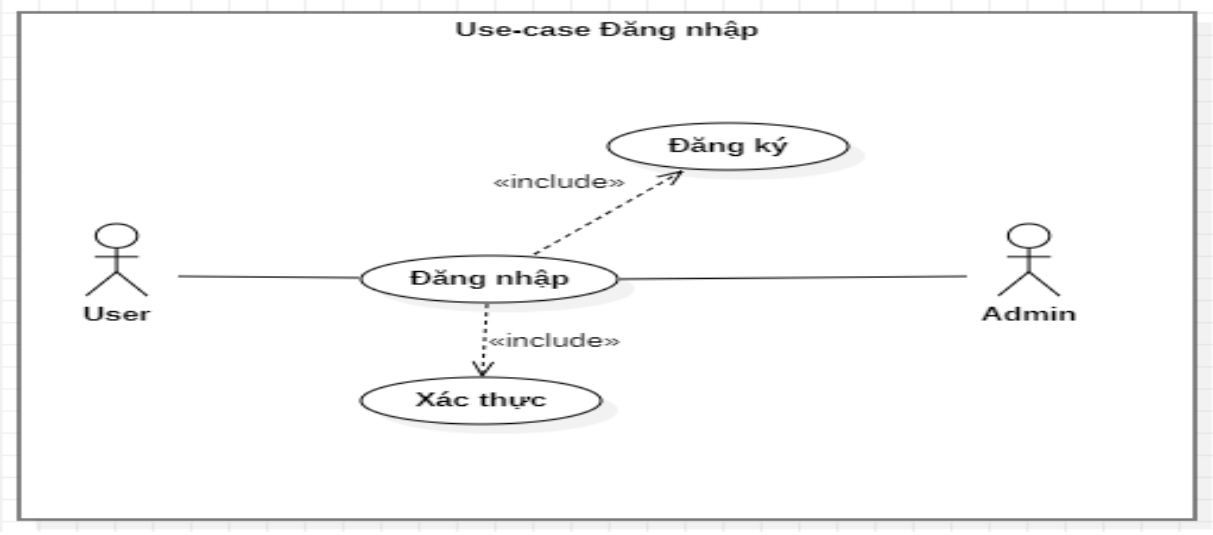
2.4.1 Các kịch bản của hệ thống (Use-cases)



Hình 1 Biểu đồ Use-case Tổng quát

2.4.2 Mô hình Use-case

2.4.2.1 Use-case Đăng nhập



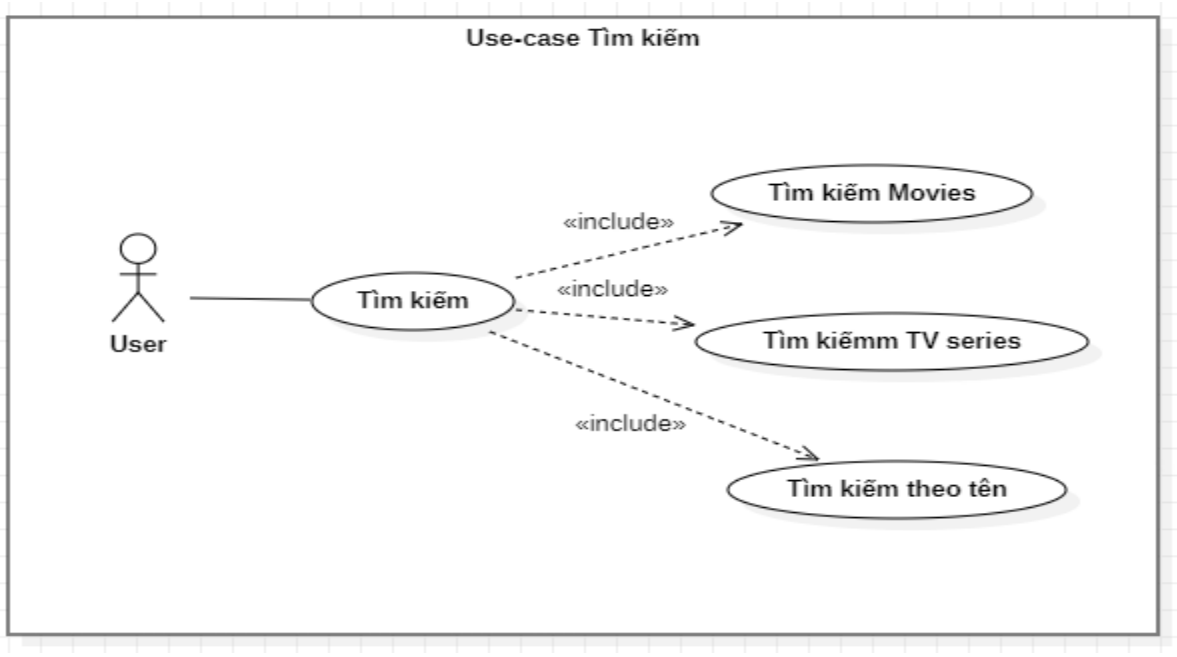
Hình 2 Use-case Đăng nhập

Mô tả:

Khi người dùng vào trang web có thể đăng nhập tài khoản mình. Nếu nhập đúng thông tin thì hệ thống sẽ thông báo thành công và cập nhật lại màn hình tương tác.

Nếu nhập sai thông tin thì hệ thống sẽ báo lỗi và yêu cầu nhập lại.

2.4.2.2 Use-case Tìm kiếm



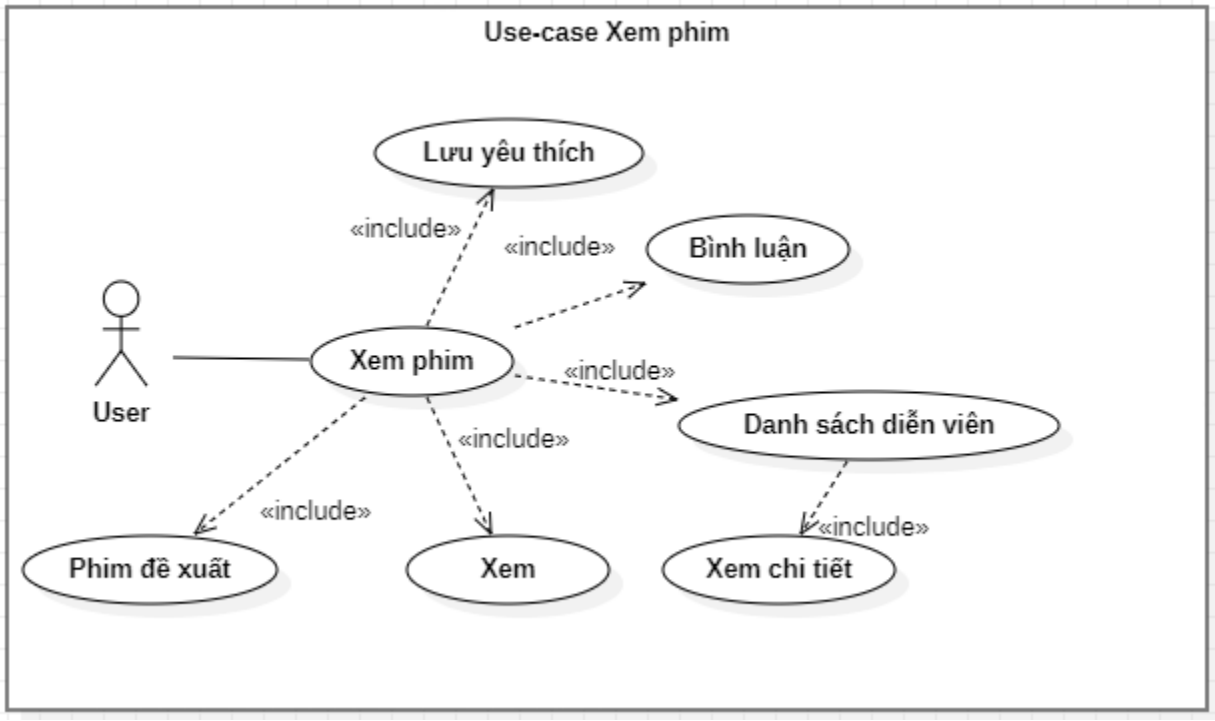
Hình 3 Use-case Tìm kiếm

Mô tả:

Khi người dùng nhấn chức năng tìm kiếm thì giao diện tìm kiếm sẽ hiện ra. Người dùng cần chọn tìm kiếm theo Movie, TV Series hay People. Sau khi người dùng nhập từ khóa và chọn tìm kiếm thì hệ thống sẽ hiển thị những bộ phim phù hợp với tiêu chí tìm kiếm.

Nếu không tìm thấy phim phù hợp thì sẽ thông báo không tìm thấy.

2.4.2.3 Use-case Xem phim



Hình 4 Use-case Xem Phim

Mô tả:

Khi người dùng chọn vào phim bất kỳ thì màn hình sẽ hiển thị ra thông tin chi tiết phim đó. Trong màn hình này sẽ có các thông tin về diễn viên, tên phim, năm sản xuất, phim đề xuất....

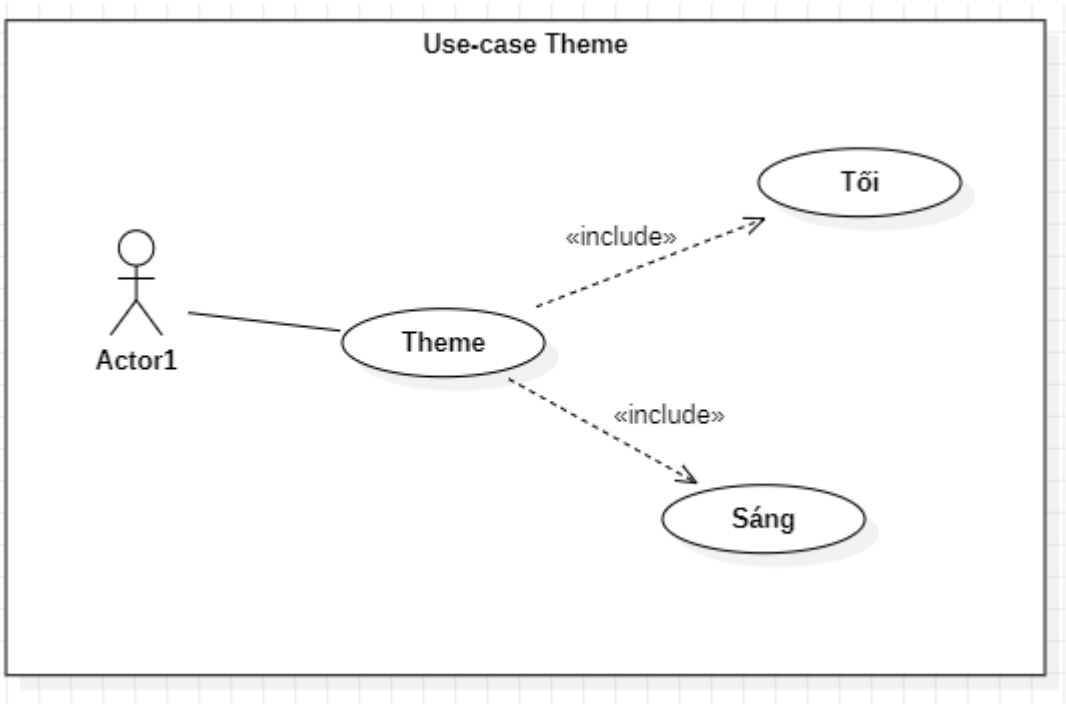
Người dùng có thể nhấn vào hình diễn viên để xem thông tin về diễn viên đó và các bộ phim của người đó.

Khi người dùng nhấn vào lưu yêu thích thì hệ thống sẽ thêm phim này vào dữ liệu của người dùng

Khi nhấn nút “xem” thì sẽ chuyển tới phần xem video.

Kéo xuống dưới là phần bình luận

2.4.2.4 Use-case Theme

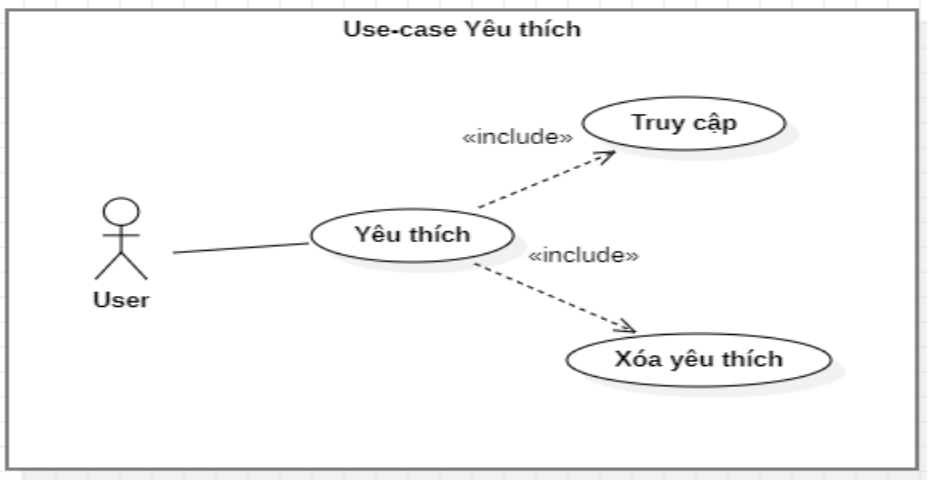


Hình 5 Use-case Theme

Mô tả:

Khi người dùng nhấn vào biểu tượng theme thì giao diện sẽ chuyển sang chủ đề sáng hoặc tối.

2.4.2.5 Use-case Yêu thích

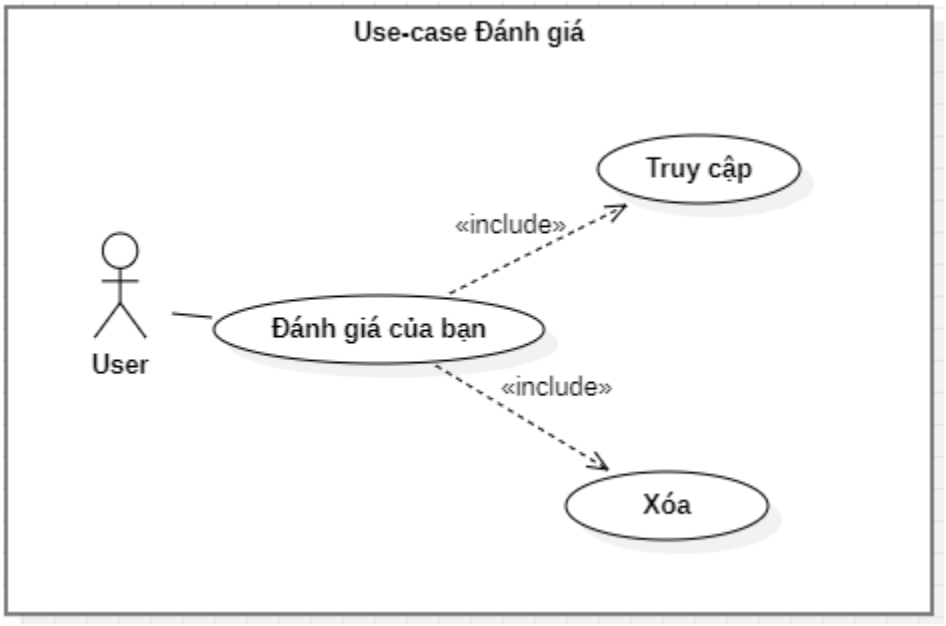


Hình 6 Use-case Yêu thích

Mô tả:

Khi người dùng nhấn vào “Favourist” thì sẽ đưa tới giao diện phim yêu thích. ở đây người dùng có thể xóa yêu thích hoặc truy cập vào trang thông tin chi tiết của phim đó .

2.4.2.6 Use-case Đánh giá

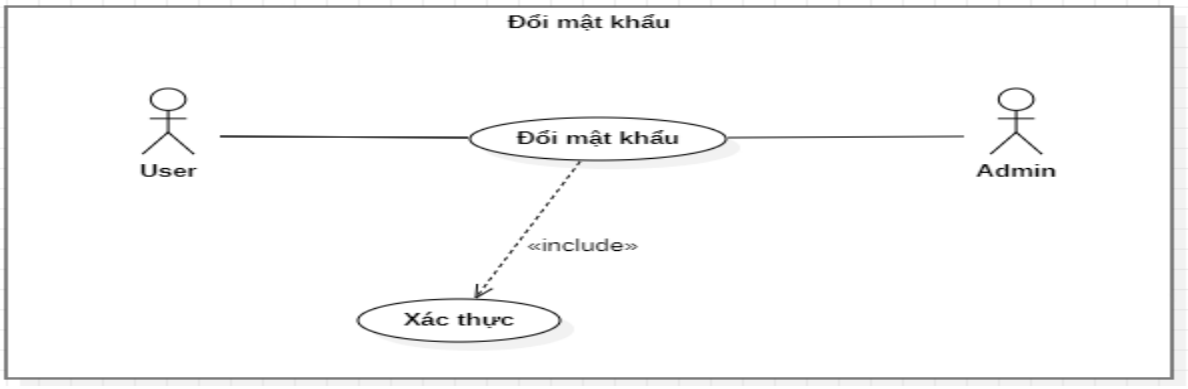


Hình 7 Use-case Đánh giá

Mô tả:

Khi người dùng bình luận trên trang của phim nào đó thì sẽ có dữ liệu trong mục “Reviews” Khi nhấn vào đây màn hình sẽ hiển thị thông tin các bình luận cũ của người dùng. Người dùng có thể xóa hoặc truy cập lại bộ phim chứa bình luận đó.

2.4.2.7 Use-case Đổi mật khẩu



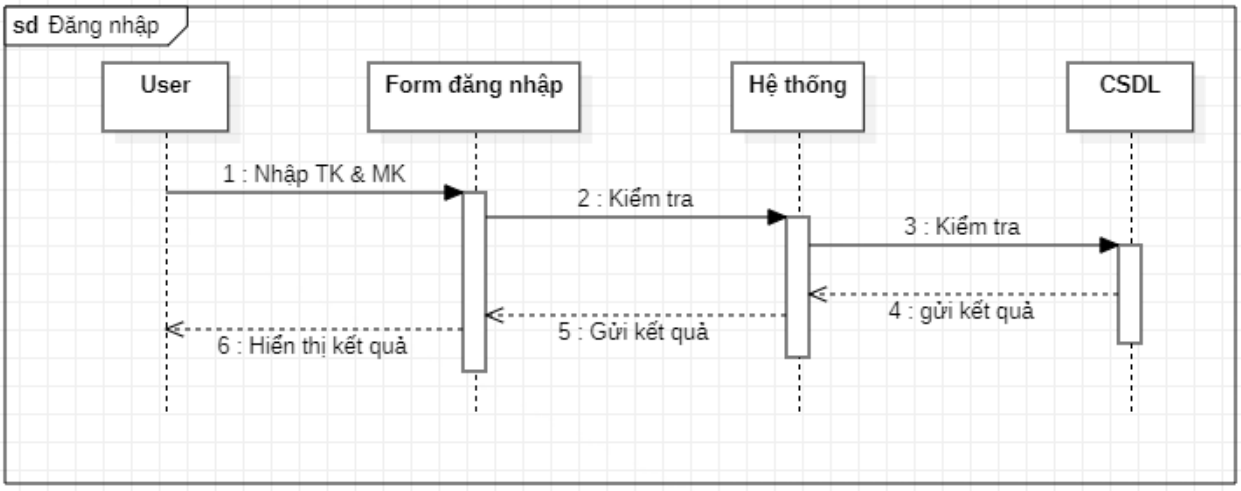
Hình 8 Use-case Đổi mật khẩu

Mô tả:

Khi người dùng nhấn vào mục đổi mật khẩu thì giao diện đổi mật khẩu sẽ hiện lên. Người dùng cần phải nhập mật khẩu hiện tại, mật khẩu mới, nhập lại mật khẩu mới sau đó nhấn ok. Nếu mật khẩu cũ đúng và hai mật khẩu mới trùng nhau thì hiển thị thông báo thành công. Nếu không thì sẽ yêu cầu người dùng nhập lại.

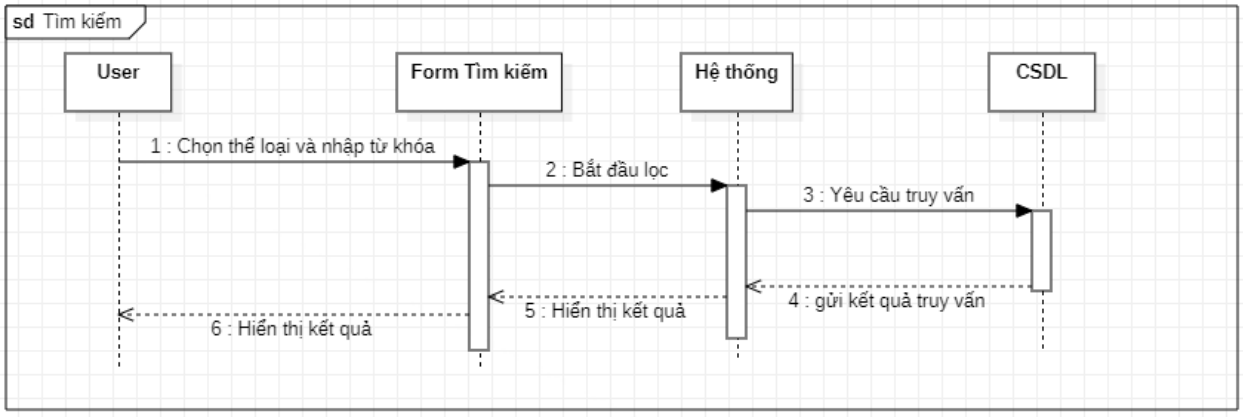
2.4.3 Các biểu đồ tuần tự

2.4.3.1 Biểu đồ tuần tự cho Use-case Đăng nhập



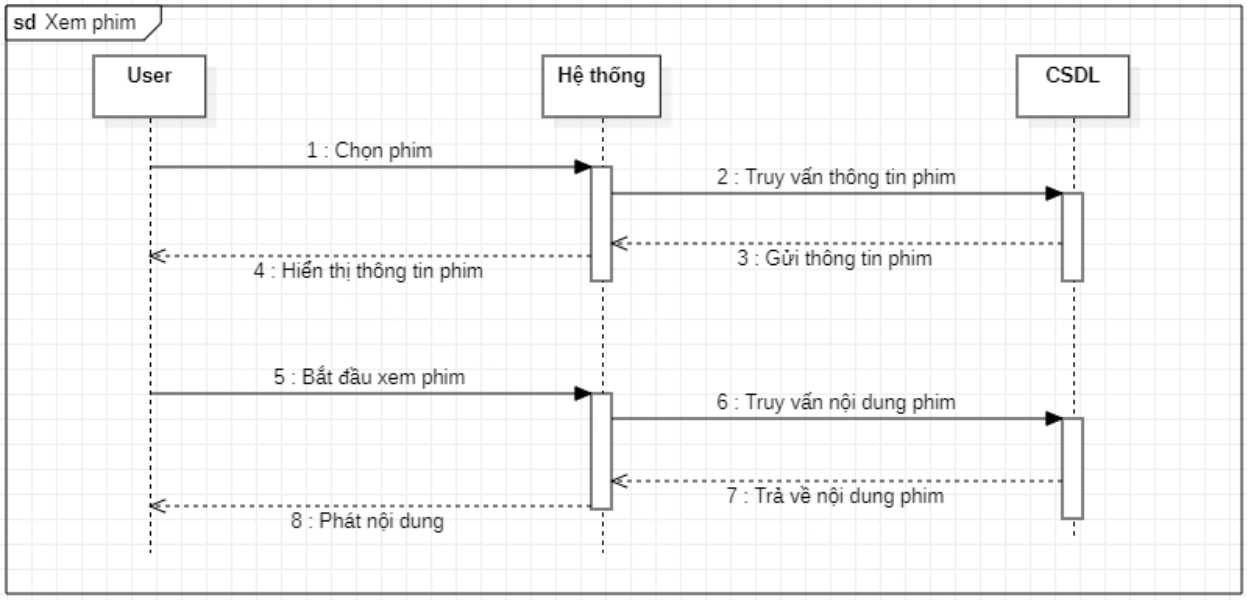
Hình 9 Biểu đồ tuần tự cho Use--case Đăng nhập

2.4.3.2 Biểu đồ tuần tự cho Use-case Tìm kiếm



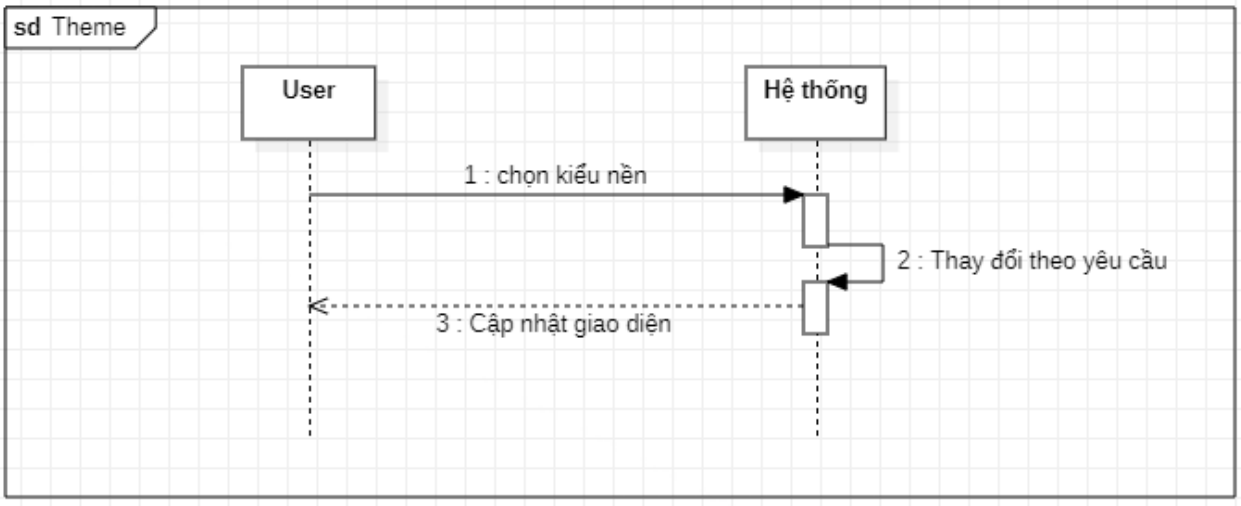
Hình 10 Biểu đồ tuần tự cho Use-case Tìm kiếm

2.4.3.3 Biểu đồ tuần tự cho Use-case Xem phim



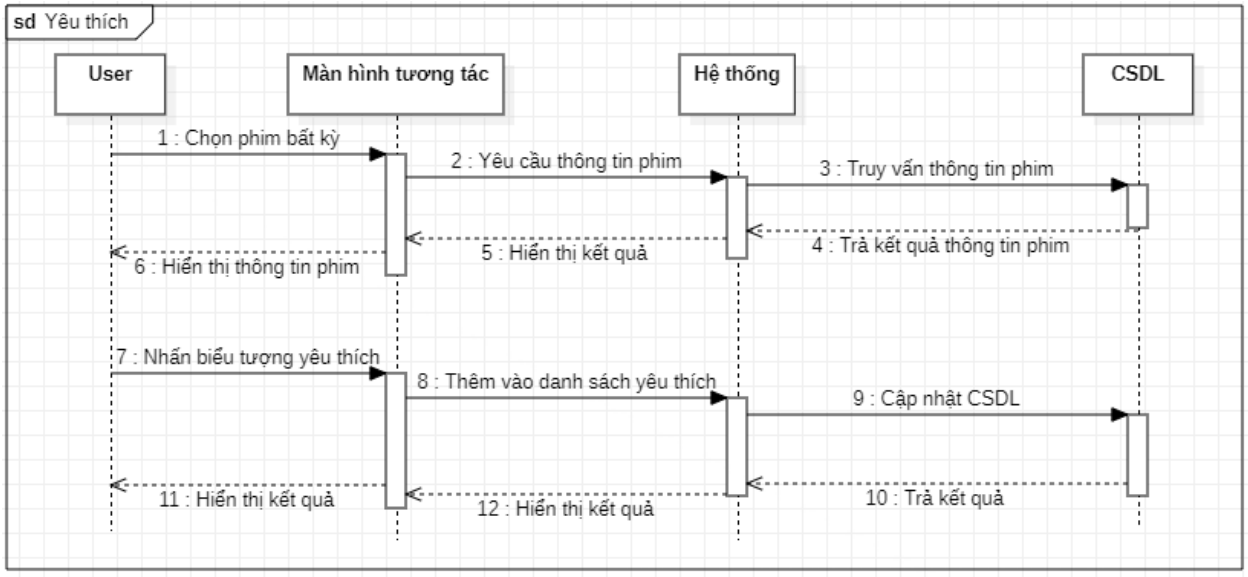
Hình 11 Biểu đồ tuần tự cho Use-case Xem phim

2.4.3.4 Biểu đồ tuần tự cho Use-case Theme



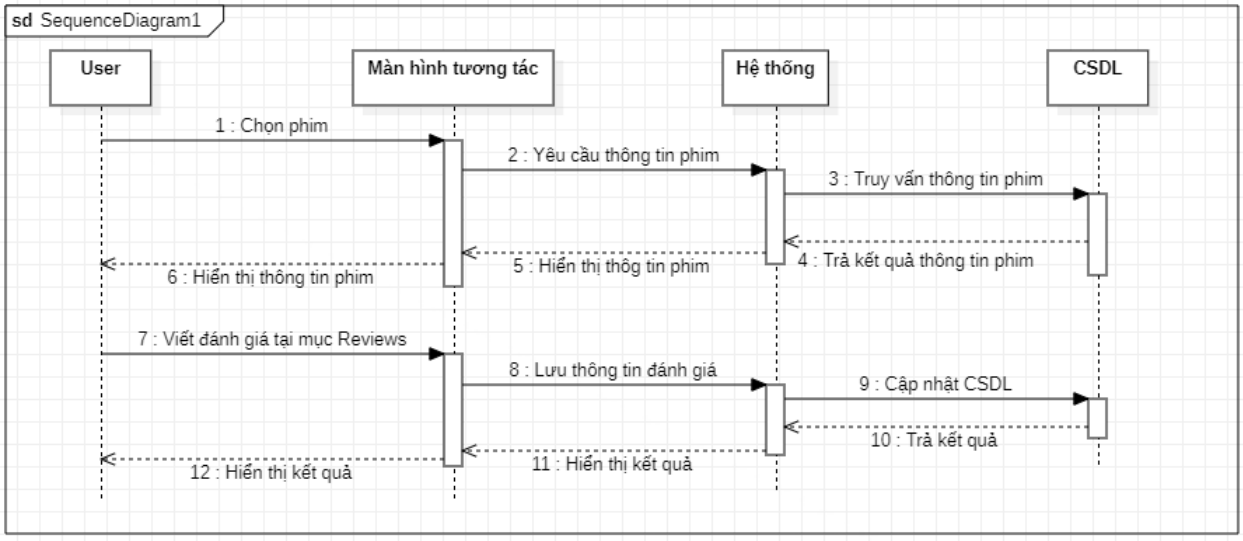
Hình 12 Biểu đồ tuần tự cho Use-case Theme

2.4.3.5 Biểu đồ tuần tự cho Use-case Yêu thích



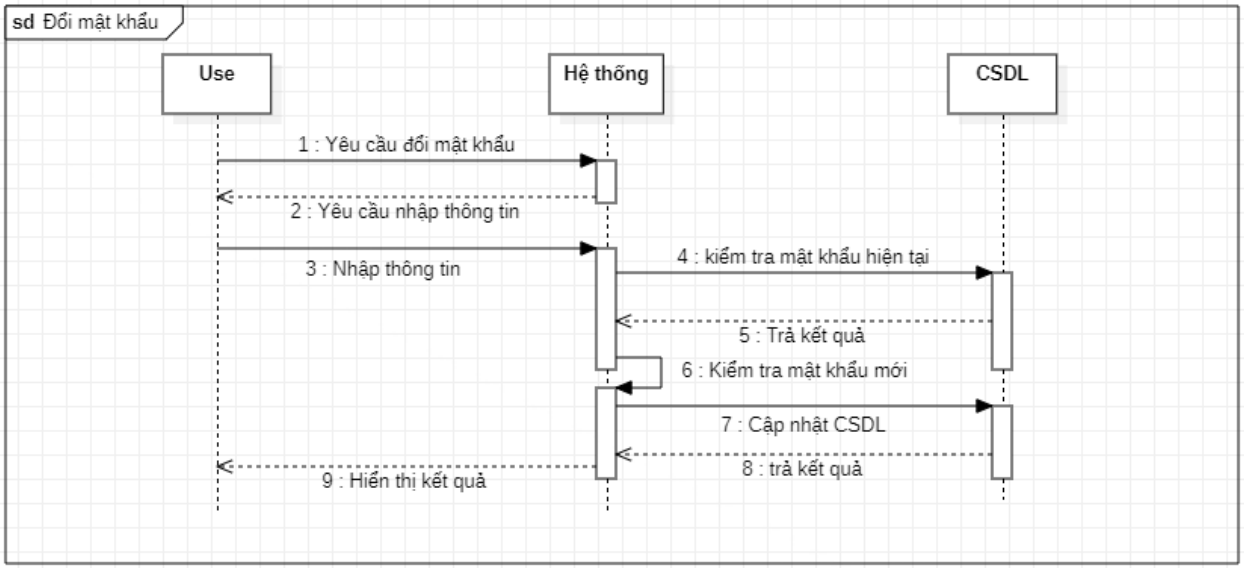
Hình 13 Biểu đồ tuần tự cho Use-case Yêu thich

2.4.3.6 Biểu đồ tuần tự cho Use-case Đánh giá



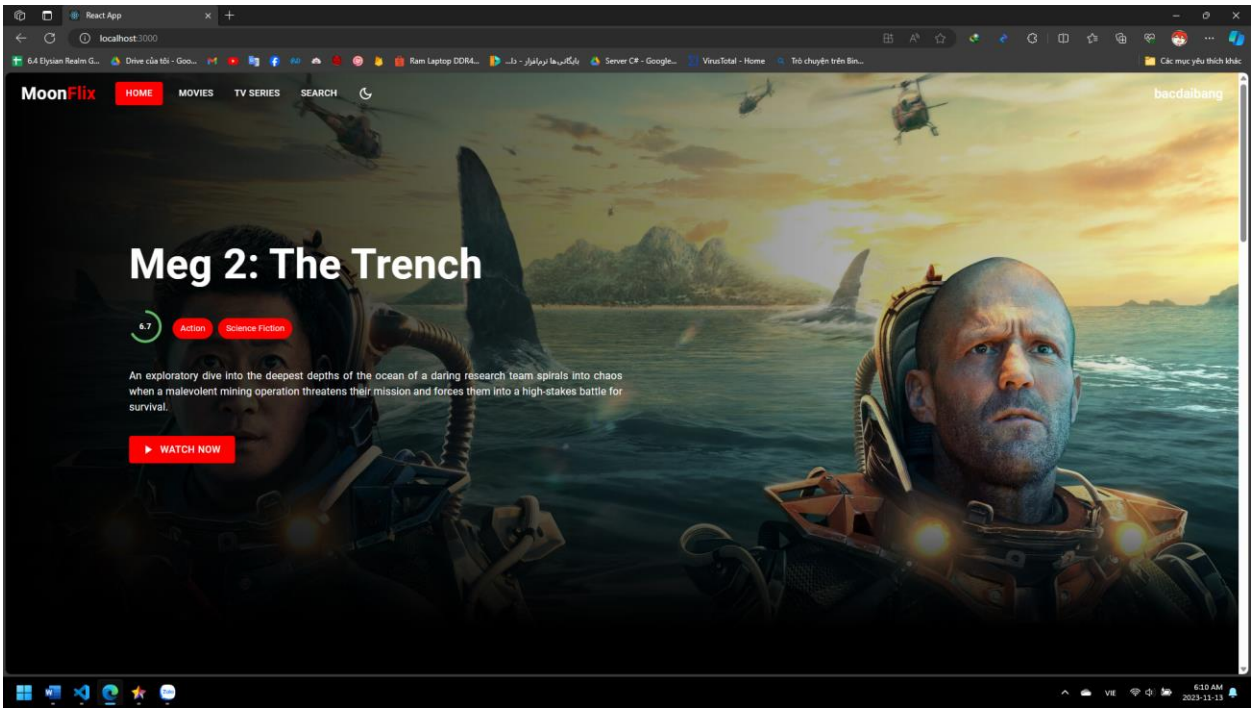
Hình 14 Biểu đồ tuần tự cho Use-case Đánh giá

2.4.3.7 Biểu đồ tuần tự cho Use-case Đổi mật khẩu

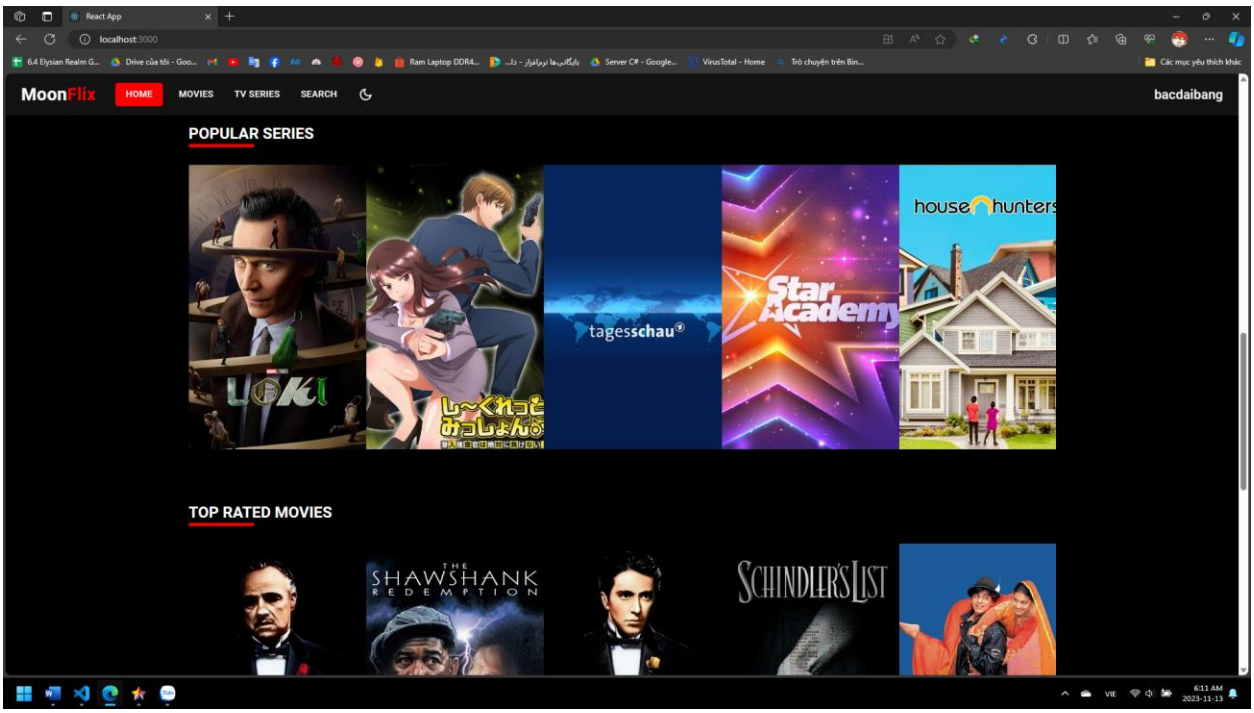


Hình 15 Biểu đồ tuần tự cho Use-case Đổi mật khẩu

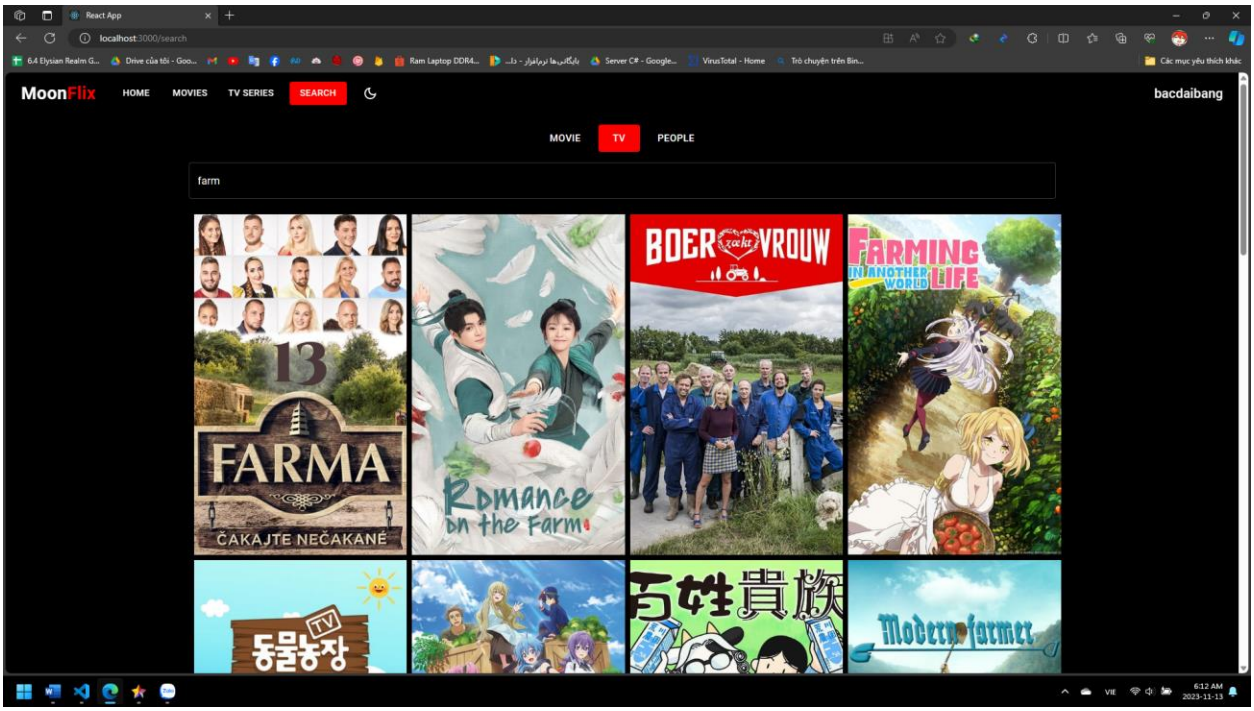
2.4.4 Các màn hình giao diện người dùng



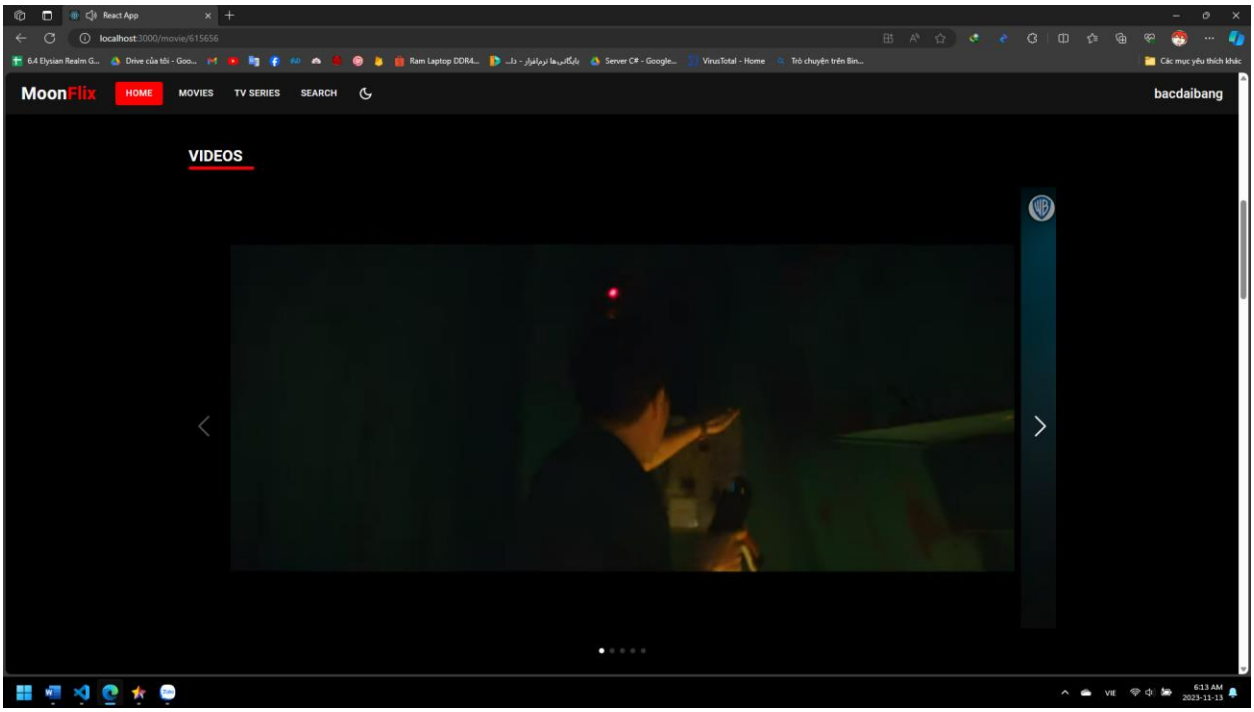
Hình 16 Màn hình chính



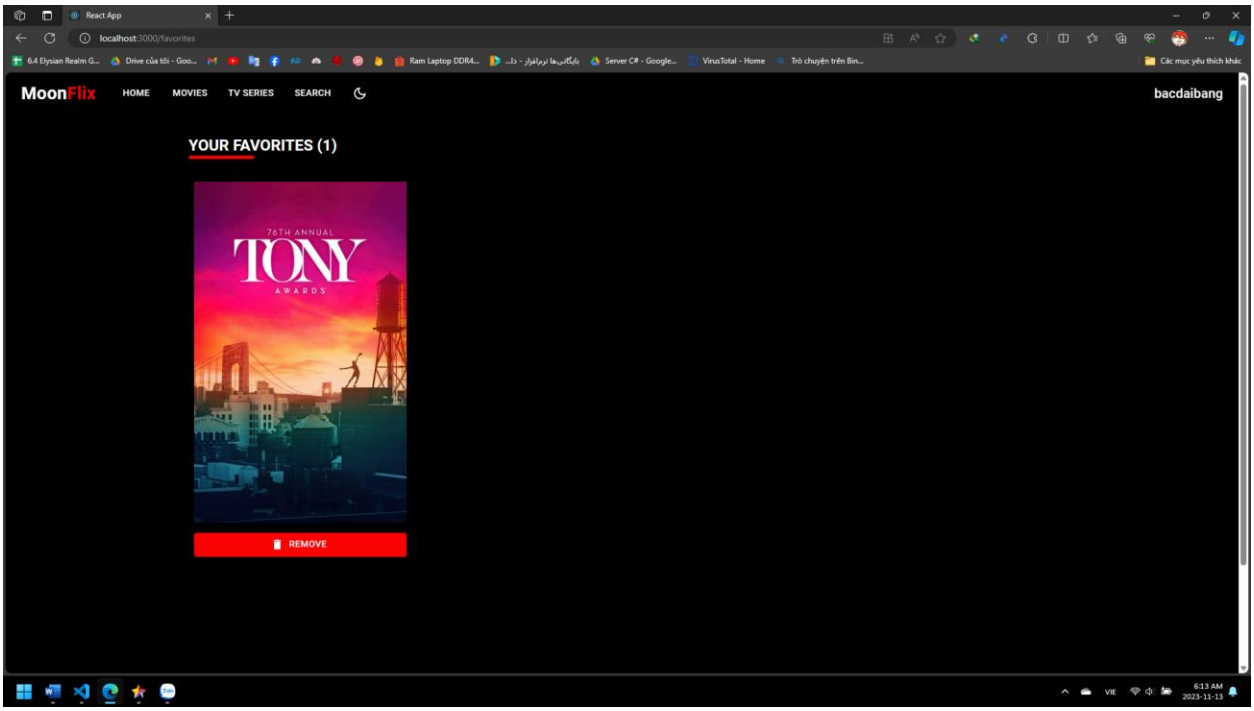
Hình 17 Bảng xếp hạng trong màn hình chính



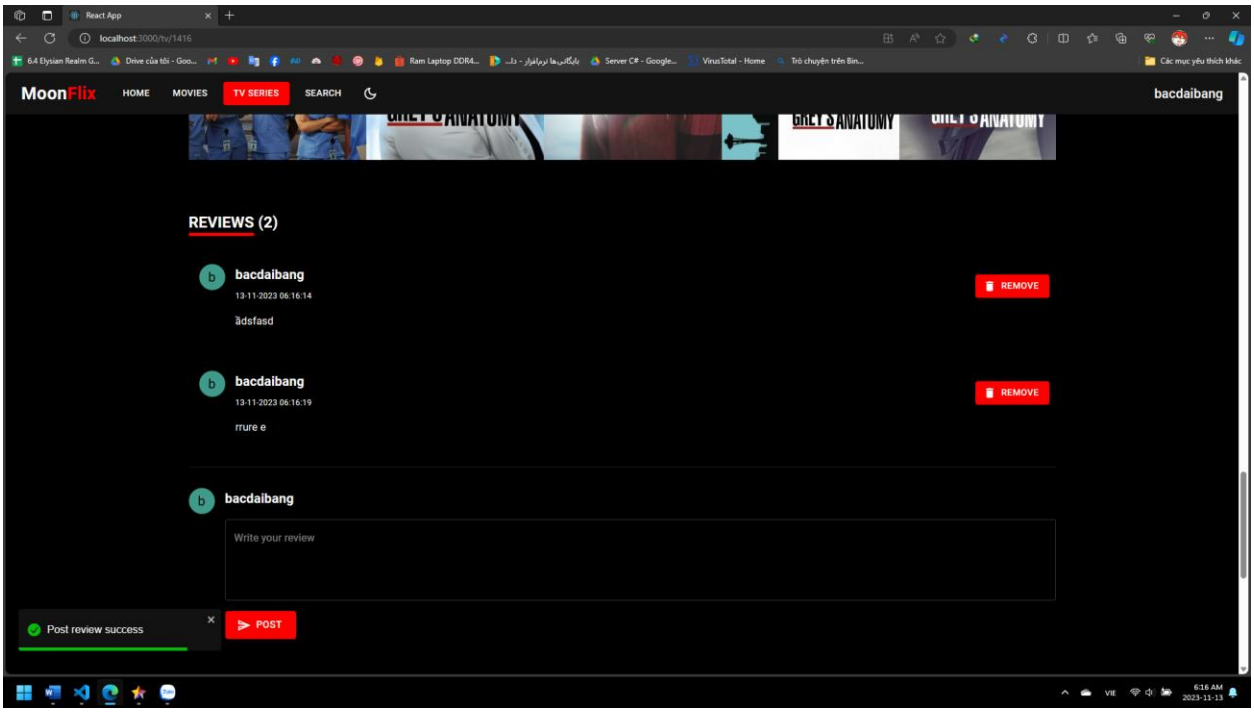
Hình 18 Tìm kiếm



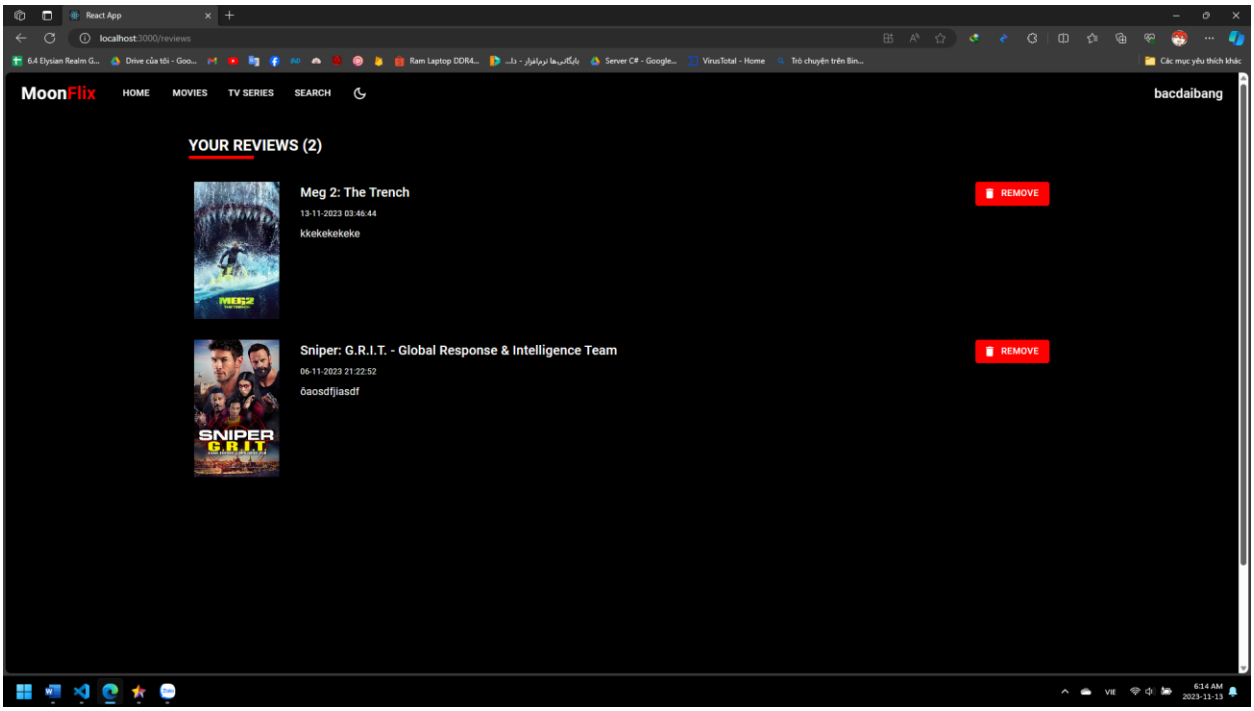
Hình 19 Xem phim



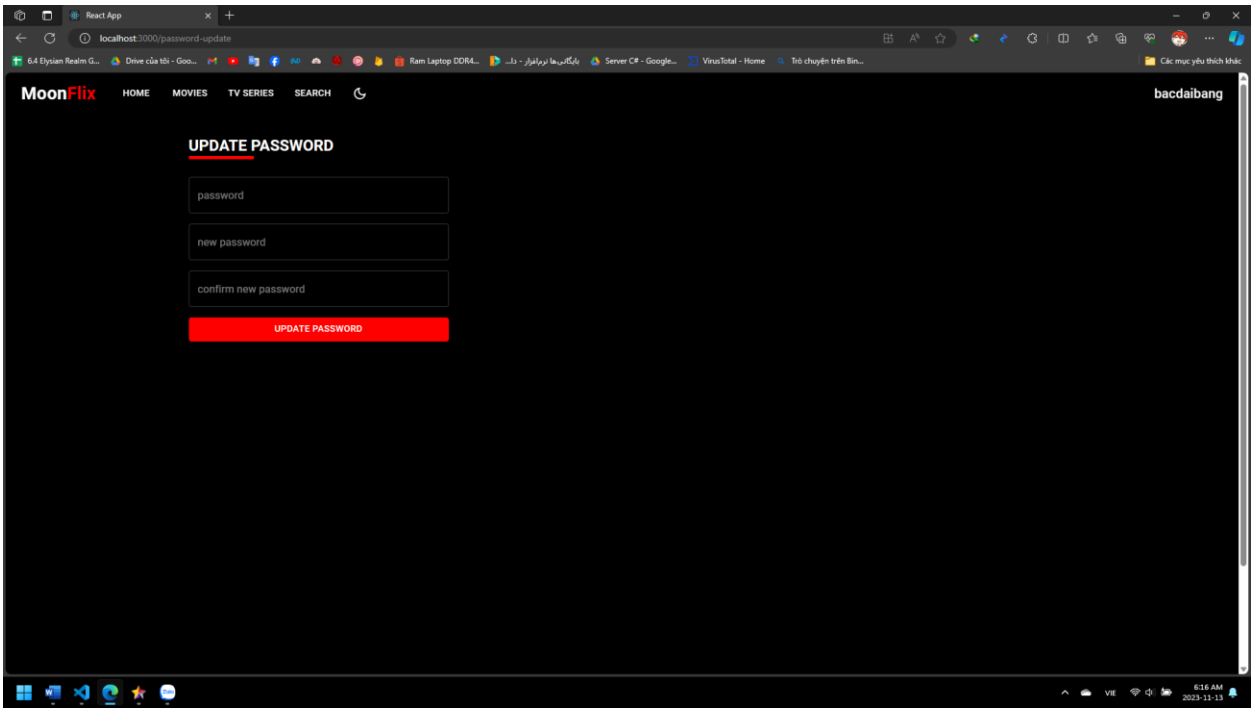
Hình 20 Danh sách phim yêu thích



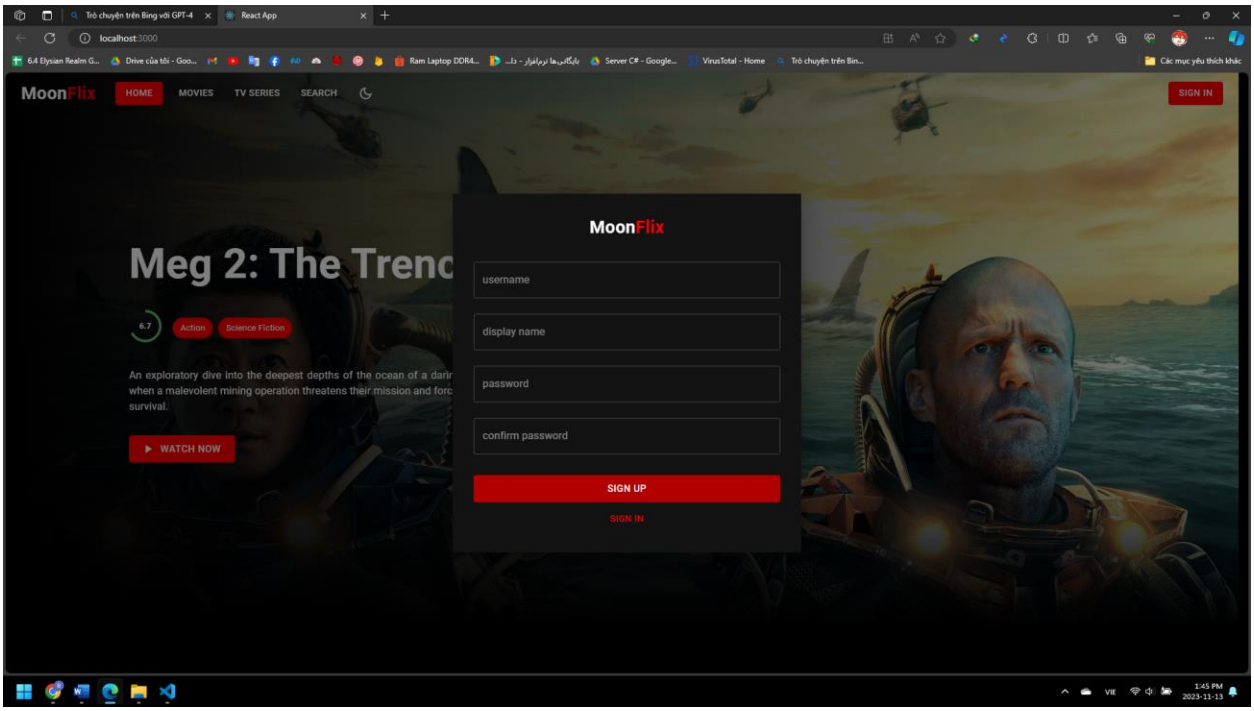
Hình 21 Bình luận



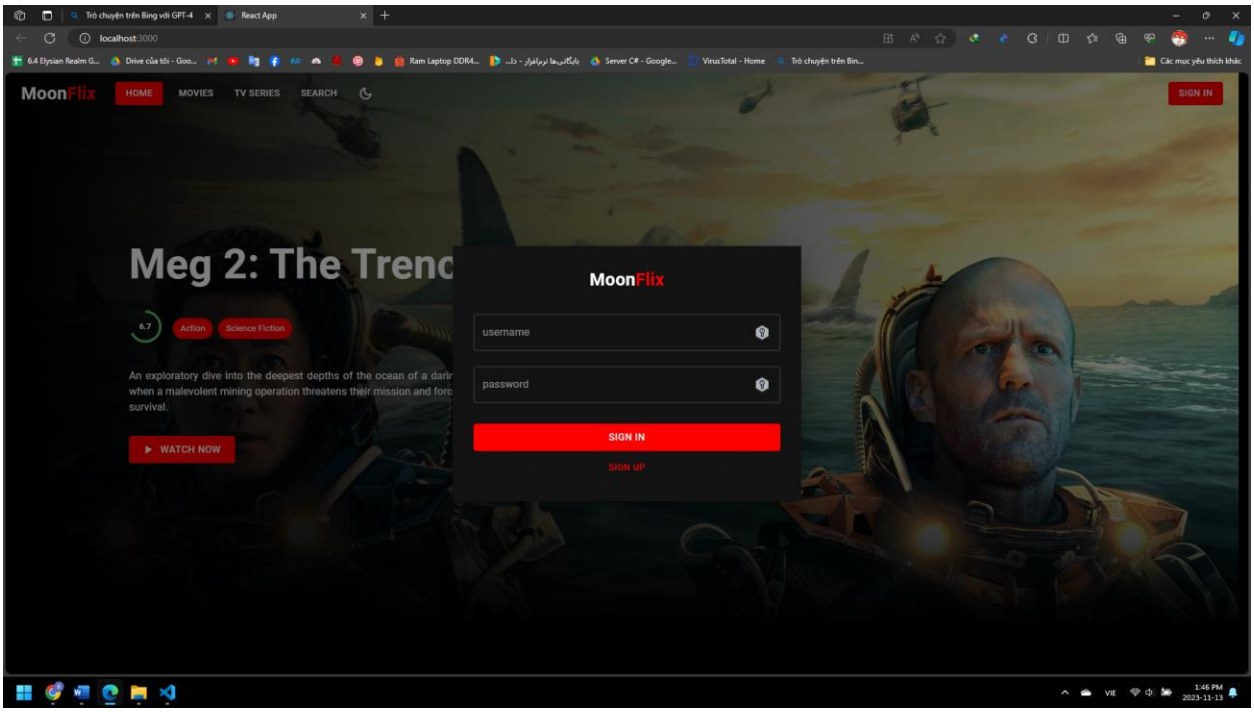
Hình 22 Danh sách các đánh giá của người dùng



Hình 23 Đổi mật khẩu



Hình 24 Đăng ký



Hình 25 Đăng nhập

3. Một số thành phần khác của đồ án

3.1 Kế hoạch dự án

SỐ CẤU TRÚC CÔNG VIỆC	TIÊU ĐỀ CÔNG VIỆC	NGƯỜI PHỤ TRÁCH	NGÀY BẮT ĐẦU	NGÀY ĐẾN HẠN	THỜI GIAN	ĐỘ HOÀN THIỆN
1	**Lên ý tưởng và thiết lập dự án**					
1.1	Đặt vấn đề, tìm kiếm giải pháp	Cả nhóm	12/09/23	######	2	100%
1.2	Xây dựng kế hoạch, và công nghệ	Cả nhóm	14/09/23	######	9	100%
1.3	Xây dựng các chức năng	Cả nhóm	15/09/23	######	6	100%
	Xây dựng kịch bản hệ thống	Quân, Huy	15/09/23	######	5	90%
1.1	Tìm hiểu về Reactjs	Cả nhóm	21/09/23	######	2	100%
1.2.3	Tìm hiểu về backend - MongoDB	Minh, Bắc	20/09/23	######	3	100%
2	**Tiến hành xây dựng dự án và code**					
2.1.1	Xây dựng sơ đồ usecase	Quân, Minh	25/09/23	######	3	75%
2.1.2	Thiết kế giao diện	Cả nhóm	28/09/23	######	9	100%
2.1.4	Thiết kế cơ sở dữ liệu	Huy, Bắc	02/10/23	######	2	100%
2.2.1	Xây dựng backend	Quân, Minh	04/10/23	######	16	100%
2.2.2	Xây dựng giao diện	cả nhóm	06/10/23	######	14	80%
2.3.3	Kết nối api tới web	Huy	20/10/23	######	8	95%
3	**Kiểm Thử, sửa lỗi và hoàn thiện báo cáo**					
3.1.2	Kiểm thử chức năng web	Cả nhóm	30/10/23	######	1	88%
3.1.3	Kiểm thử chức năng	Minh, Bắc	01/11/23	######	1	90%
3.1.4	Kiểm tra độ bảo mật và logic	Cả nhóm	02/11/23	######	1	100%
3.2.1	Sửa và kiểm thử tính năng	Quân, Huy	01/11/23	######	2	100%
3.3.1	Viết báo cáo: công nghệ sử dụng	Cả nhóm	03/11/23	######	1	100%
3.3.2	Viết báo cáo: demo sản phẩm	Cả nhóm	06/11/23	######	1	100%
3.4	Kiểm tra và hoàn thiện báo cáo	Cả nhóm	08/11/23	######	1	100%
3.5	Viết slider và chuẩn bị thuyết trình	Bắc	09/11/23	######	2	100%

Hình 26 Bảng phân chia công việc

- Biểu đồ kế hoạch dự án

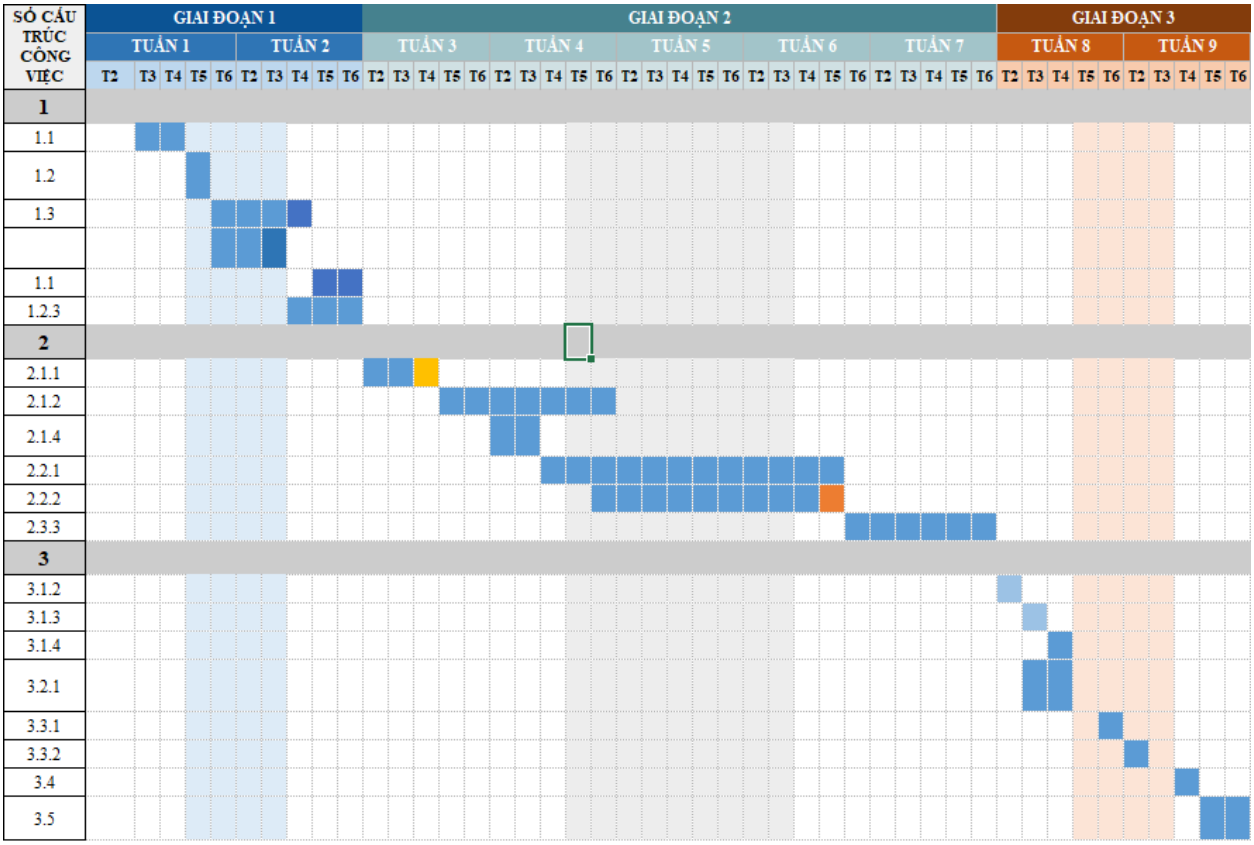


Figure 1. Biểu đồ kế hoạch thực hiện dự án

3.2 Đảm bảo thực hiện đúng làm việc nhóm

Dự án của chúng em tuân theo một quy trình rõ ràng và hiệu quả. Mỗi thành viên trong nhóm đều có một vai trò cụ thể, từ lãnh đạo nhóm, người quản lý dự án, đến chuyên gia kỹ thuật. Chúng em đã xây dựng một kế hoạch công việc chi tiết, bao gồm các công việc cụ thể, mục tiêu và thời hạn. Chúng em cũng đã lên lịch cho các cuộc họp, meeting minutes, xây dựng phần mềm và các sự kiện quan trọng khác.

Công cụ truyền thông giao tiếp chúng em sử dụng là Zalo và tổ chức các cuộc họp định kỳ để cập nhật tình trạng công việc và giải quyết vấn đề. Thống nhất sử dụng HTML, JavaScripts, ReactJS, MongoDB và sử dụng Github để trao đổi mã nguồn. Chia

nhỏ công việc thành các phần nhỏ hơn để mỗi thành viên có thể dễ dàng theo dõi và đóng góp. Mọi người đều hiểu rõ về công việc của họ và các phụ thuộc giữa các nhiệm vụ.

Chúng em sử dụng Github và Jira để quản lý dự án. Các thành viên đã xác định quy trình làm việc chung, từ việc kiểm tra mã nguồn, kiểm thử, đến triển khai. Tất cả thành viên đều tuân theo quy trình này. Chúng em đảm bảo có quy trình kiểm thử đầy đủ, chi tiết để đảm bảo không bỏ sót vấn đề nào của sản phẩm. Và cũng thực hiện đánh giá các thành viên để kịp thời cải thiện dự án, quy trình làm việc.

Nhóm chúng em cũng khuyến khích sự sáng tạo và chia sẻ ý kiến. Tạo một môi trường làm việc tích cực và thoải mái để mọi người có thể làm việc hiệu quả. Với những nguyên tắc và quy trình này, hy vọng sẽ đạt được mục tiêu của dự án một cách hiệu quả và kịp thời.

3.3 Các vấn đề về đạo đức và làm việc chuyên nghiệp

Nhóm chúng em xin đảm bảo các điều sau:

- **Bản quyền và sở hữu trí tuệ**: Đảm bảo tất cả các bộ phim và nội dung video khác trên website đều tuân thủ các quy định về bản quyềnĐây là một trong những vấn đề lớn nhất khi xây dựng một trang web xem phim đều tuân thủ các quy định về bản quyền .
- **Quyền riêng tư của người dùng:** Đảm bảo thông tin cá nhân của người dùng được bảo vệ và không được tiết lộ khi không có sự đồng ý của họ.
- **Chất lượng nội dung:** Đảm bảo nội dung được cung cấp đáp ứng được yêu cầu về chất lượng hình ảnh.
- **Đạo đức kinh doanh:** Đảm bảo thông tin chính xác về các sản phẩm và dịch vụ, không lừa dối khách hàng và tuân thủ quy định pháp luật.
- **Chuyên nghiệp:** Đảm bảo trang web dễ dàng sử dụng, giao diện thân thiện và cung cấp trải nghiệm tốt cho người dùng.
- **Văn hóa và chuẩn mực xã hội:** Đảm bảo tôn trọng các giá trị văn hóa, không đăng tải nội dung phân biệt đối xử hoặc xúc phạm.

Chúng tôi cam đoan đồ án do chúng tôi tự làm, và các nội dung tham khảo, trich dẫn trong luận văn được đề cập đầy đủ ở mục tham khảo.
Chúng tôi cũng cam đoan không sử dụng sản phẩm của đồ án để thực hiện những việc trái pháp luật, đạo đức, gây anh hưởng xấu đến cộng đồng, xã hội.

Ký tên
Bắc
Minh
Huy
Quân

3.4 Tác động xã hội

Nội dung: Website cung cấp cho người dùng khả năng truy cập vào một lượng lớn nội dung giải trí. Điều này có thể tạo ra cơ hội cho người dùng khám phá các thể loại phim mới, các văn hóa khác nhau và các quan điểm sáng tạo.

Tương tác xã hội: Cung cấp các tính năng tương tác xã hội cho phép người dùng bình luận, thảo luận về các bộ phim. Điều này có thể tạo ra cơ hội cho người dùng kết nối với nhau qua sở thích chung.

Sức khỏe tâm thần: Việc xem phim trực tuyến quá mức có thể dẫn đến các vấn đề về sức khỏe tâm thần, bao gồm cảm giác đố kỵ, thua kém và kém hài lòng với cuộc sống. Nghiên cứu cũng cho thấy rằng việc xem phim trực tuyến quá mức có thể dẫn đến các triệu chúng ADHD, trầm cảm, lo âu và thiếu ngủ.

Vấn đề bản quyền: Việc xem phim trực tuyến có thể dẫn đến các vấn đề về bản quyền, đặc biệt là các bộ phim được phát sóng mà chưa được sự cho phép của người sở hữu bản quyền

Tác động đến ngành công nghiệp phim điện ảnh: Xem phim trực tuyến có thể tạo ra cơ hội mới cho ngành công nghiệp phim điện ảnh, nhưng cũng có thể tạo ra thách thức khi cạnh tranh với các dịch vụ trực tuyến

3.5 Kế hoạch cho kiến thức mới và kế hoạch học tập

Trong quá trình thực hiện dự án xây dựng website xem phim trực tuyến, mỗi thành viên trong nhóm đều có nhiều cơ hội để học hỏi và phát triển kỹ năng. Một số thành viên đã học ngôn ngữ lập trình mới như JavaScript và ReactJS từ các nguồn học trực tuyến để phát triển front-end. Một thành viên trong nhóm có kinh nghiệm với kiến trúc MVC, vì vậy anh ấy đang chia sẻ kiến thức này với những người còn lại trong nhóm back-end.

Chúng em sẽ sử dụng Node.js trong phần back-end, vì vậy tất cả các thành viên back-end đều đang tìm hiểu Node.Jstrực tuyến. Ngoài ra, nhóm của chúng em cũng đang nỗ lực nâng cao kiến thức về thiết kế kiến trúc cho dự án. Chúng em dự định học hỏi từ các anh chị khóa trên, cũng như tự học từ các nguồn trực tuyến.

Nhóm back-end cũng đang học JavaScripts từ các nguồn học trực tuyến như Udemy để chúng em có thể giúp nhau gỡ lỗi và phát triển dự án một cách hiệu quả. Với sự hỗ trợ của các công nghệ và công cụ này, hy vọng nhóm chúng em sẽ hoàn thành dự án một cách thành công.

Tài liệu tham khảo

https://react-icons.github.io/react-icons/

https://www.npmjs.com/package/react-infinite-scroll-component

https://www.npmjs.com/package/react-select

https://www.npmjs.com/package/react-tag-autocomplete

https://sass-lang.com/

https://redux-toolkit.js.org/

https://legacy.reactjs.org/docs/react-dom.html

https://www.npmjs.com/package/react-circular-progressbar

https://www.npmjs.com/package/react-lazy-load-image-component

https://www.npmjs.com/package/react-player

https://www.npmjs.com/package/react-router-dom

github: iBihari/DACS_G6: nah (github.com)