

ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

Phát triển ứng dụng gợi ý phim tích hợp GenAI trong phân tích hành vi và truy vấn ngôn ngữ người dùng

TRẦN HỒNG QUÂN

quan.th240818E@sis.hust.edu.vn

Ngành Kỹ sư chuyên sâu Trí tuệ nhân tạo tạo sinh

Giảng viên hướng dẫn: TS. Đặng Tuấn Linh

Chữ ký GVHD

Khoa: Kỹ thuật máy tính

Trường: Công nghệ Thông tin và Truyền thông

HÀ NỘI, 01/2026

LỜI CẢM ƠN

Trong thời gian làm đồ án tốt nghiệp, tôi đã nhận được rất nhiều sự giúp đỡ, đóng góp ý kiến và sự chỉ bảo nhiệt tình của các thầy cô, gia đình và bạn bè.

Tôi xin gửi lời cảm ơn chân thành đến TS. Đặng Tuấn Linh, giảng viên khoa Kỹ thuật máy tính, trường Công nghệ thông tin và Truyền thông, Đại học Bách khoa Hà Nội - người đã tận tình chỉ bảo, hướng dẫn tôi trong quá trình làm đồ án tốt nghiệp.

Tôi xin chân thành cảm ơn các thầy cô giáo trong Đại học Bách khoa Hà Nội nói chung và các thầy cô trong trường Công nghệ thông tin và Truyền thông nói riêng đã dạy cho tôi những kiến thức các môn đại cương cũng như các môn chuyên ngành, giúp tôi có được cơ sở lý thuyết vững vàng cho tôi sự tự tin cho chặng đường sắp tới.

Tôi cũng xin chân thành cảm ơn gia đình và bạn bè, những người đã luôn đồng hành, luôn quan tâm, giúp đỡ, động viên tôi trong qua trình học tập.

Cuối cùng, tôi tự hào vì đã không ngừng nỗ lực và quyết tâm để đạt được kết quả tốt nhất. Sự chăm chỉ và kiên trì của chính bản thân đã hoàn thành bài đồ án tốt nghiệp này.

TÓM TẮT NỘI DUNG ĐỒ ÁN

Trong bối cảnh bùng nổ của ngành công nghiệp giải trí, việc lựa chọn phim phù hợp với sở thích cá nhân ngày càng trở nên khó khăn hơn do số lượng phim ngày càng tăng. Hiện tại, các hệ thống giới thiệu phim đã được phát triển để giúp người dùng tìm kiếm và khám phá nội dung mới một cách hiệu quả hơn. Tuy nhiên, các trang này thường chia tách các thông tin ra riêng biệt, tạo ra những trang thông tin không đầy đủ, đặc biệt chưa có chức năng gợi ý theo lịch sử hành động và tìm kiếm phim bằng truy vấn ngôn ngữ người dùng.

Trong đồ án tốt nghiệp này, tôi lựa chọn hướng tiếp cận là kết hợp tất cả thông tin vào ứng dụng của mình và phát triển thêm chức năng truy vấn ngôn ngữ và theo dõi hành động người dùng. Ứng dụng sẽ bao gồm tìm kiếm, xem chi tiết phim, gợi ý phim, review, đánh giá phim, xem trailer, hình ảnh của phim. Lý do tôi chọn hướng đi này vì nó giúp người dùng có trải nghiệm một cách tốt nhất từ các tính năng cần thiết và tích hợp vào một chỗ. Ngoài ra, chức năng gợi ý, tìm kiếm, theo dõi hành động giúp người dùng tìm được bộ phim yêu thích một cách nhanh nhất cho bản thân.

Giải pháp của tôi bao gồm việc xây dựng một ứng dụng giới thiệu phim với ba loại người dùng: Admin, User và Guest. Ứng dụng cho phép Guest tìm kiếm, sắp xếp và xem thông tin phim. User có thể đánh giá, comment và quản lý danh sách yêu thích của mình. Admin có quyền quản lý toàn bộ thông tin phim và người dùng. Điểm nổi bật của ứng dụng là khả năng gợi ý phim dựa trên độ tương tự, cập nhật tương tác của người dùng để đưa ra gợi ý một cách tốt nhất.

Đóng góp chính của đồ án là xây dựng một ứng dụng gợi ý phim bằng genAI toàn diện, kết hợp các phương pháp tiên tiến để cải thiện trải nghiệm người dùng. Kết quả đạt được là một ứng dụng có khả năng cung cấp nội dung phim một cách đầy đủ, gợi ý phim theo sở thích, mong muốn của người dùng một cách nhanh (10s) và chính xác, giúp người dùng dễ dàng tìm kiếm và khám phá những bộ phim phù hợp với sở thích cá nhân.

Sinh viên thực hiện
(Ký và ghi rõ họ tên)

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....	1
1.1 Đặt vấn đề.....	1
1.2 Mục tiêu và phạm vi đề tài.....	2
1.3 Định hướng giải pháp.....	2
1.4 Bố cục đồ án	3
CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU.....	5
2.1 Khảo sát hiện trạng	5
2.2 Khảo sát về quyền theo dõi	7
2.3 Tổng quan chức năng	9
2.3.1 Biểu đồ use case tổng quát	9
2.3.2 Biểu đồ use case phân rã Quản lý thông tin người dùng	10
2.3.3 Biểu đồ use case phân rã Quản lý thông tin phim	11
2.3.4 Biểu đồ use case phân rã Quản lý tài khoản	11
2.3.5 Biểu đồ use case phân rã Tương tác với phim.....	12
2.3.6 Biểu đồ use case phân rã Review, comment	12
2.4 Đặc tả chức năng	13
2.4.1 Đặc tả use case Đăng nhập.....	13
2.4.2 Đặc tả use case Chính sửa thông tin cá nhân	14
2.4.3 Đặc tả use case Tìm kiếm phim	15
2.4.4 Đặc tả use case Đánh dấu phim mình thích.....	16
2.4.5 Đặc tả use case Đánh giá phim	17
2.4.6 Đặc tả use case CRUD phim	18
2.4.7 Đặc tả use case Gợi ý phim theo nội dung	19
2.5 Yêu cầu phi chức năng	19

CHƯƠNG 3. CƠ SỞ LÝ THUYẾT CHO GỢI Ý PHIM..... 21

3.1 Content-Based Filtering	21
3.1.1 Khái niệm	21
3.1.2 Các thuật toán.....	21

CHƯƠNG 4. CÔNG NGHỆ SỬ DỤNG..... 23

4.1 Công nghệ sử dụng phía frontend	23
4.1.1 React	23
4.1.2 Tailwind CSS	23
4.2 Công nghệ sử dụng phía backend	24
4.2.1 Django	24
4.2.2 MySQL.....	24
4.3 Công nghệ sử dụng trong việc lưu trữ code	25
4.3.1 Git	25
4.3.2 GitHub.....	26
4.4 Công nghệ sử dụng trong việc crawl dữ liệu	26
4.4.1 Selenium và Requests	26
4.4.2 Công nghệ sử dụng để update dữ liệu.....	27

CHƯƠNG 5. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG 29

5.1 Thiết kế kiến trúc.....	29
5.1.1 Lựa chọn kiến trúc phần mềm	29
5.1.2 Thiết kế tổng quan.....	30
5.1.3 Thiết kế chi tiết gói	32
5.2 Thiết kế chi tiết.....	35
5.2.1 Thiết kế giao diện	35
5.2.2 Thiết kế cơ sở dữ liệu	41
5.2.3 Thiết kế hệ thống crawl dữ liệu.....	42

5.2.4 Tìm kiếm phim bằng ngôn ngữ tự nhiên	43
5.2.5 Gợi ý phim bằng genAI dựa vào lịch sử hoạt động của người dùng .	45
5.3 Xây dựng ứng dụng.....	48
5.3.1 Thư viện và công cụ sử dụng	48
5.3.2 Kết quả đạt được	48
5.3.3 Minh họa các chức năng chính	52
5.4 Kiểm thử.....	54
5.5 Triển khai	58
CHƯƠNG 6. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT	60
6.1 Thiếu khá nhiều dữ liệu phim ảnh	60
6.2 Thiết kế database khá khó khăn với quá nhiều trường	60
6.3 Trang web có dữ liệu update liên tục theo ngày khiến link ảnh và trailer bị hỏng.....	60
6.4 Lựa chọn thuật toán gợi ý phù hợp.....	63
CHƯƠNG 7. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	67
TÀI LIỆU THAM KHẢO.....	70

DANH MỤC HÌNH VẼ

Hình 2.1	Form khảo sát người dùng về các ứng dụng	5
Hình 2.2	Khảo sát người dùng về các ứng dụng	6
Hình 2.3	Form khảo sát về quyền theo dõi	7
Hình 2.4	Khảo sát về quyền theo dõi	8
Hình 2.5	Use case tổng quan	10
Hình 2.6	Phân rã use case Quản lý thông tin người dùng	10
Hình 2.7	Phân rã use case Quản lý thông tin phim	11
Hình 2.8	Phân rã use case Quản lý tài khoản	11
Hình 2.9	Phân rã use case Tương tác với phim	12
Hình 2.10	Phân rã use case Review, comment	12
Hình 5.1	Biểu đồ phụ thuộc gói	30
Hình 5.2	Thiết kế chi tiết gói ở tầng Model	32
Hình 5.3	Thiết kế chi tiết gói ở tầng View	33
Hình 5.4	Thiết kế chi tiết gói ở tầng Template	35
Hình 5.5	Màn hình <i>đăng nhập</i>	36
Hình 5.6	Màn hình <i>đăng ký</i>	37
Hình 5.7	Màn hình <i>profile</i>	37
Hình 5.8	Màn hình <i>chính</i>	38
Hình 5.9	Màn hình <i>gợi ý phim ở trang chủ</i>	38
Hình 5.10	Màn hình <i>thông tin chi tiết 1</i>	39
Hình 5.11	Màn hình <i>thông tin chi tiết 2</i>	39
Hình 5.12	Màn hình <i>tổng hợp đạo diễn, diễn viên, ...</i>	40
Hình 5.13	Màn hình <i>review phim</i>	40
Hình 5.14	Sơ đồ ERD	41
Hình 5.15	Hình ảnh các table trong database 1	41
Hình 5.16	Hình ảnh các table trong database 2	42
Hình 5.17	Hình ảnh các table trong database 3	42
Hình 5.18	Màn hình <i>tìm kiếm phim</i>	43
Hình 5.19	Màn hình <i>gợi ý phim bằng genAI</i>	46
Hình 5.20	Chức năng tìm kiếm phim theo ngôn ngữ người dùng	50
Hình 5.21	Review phim	50
Hình 5.22	Gợi ý phim theo lịch sử review	51
Hình 5.23	Tìm kiếm phim	51
Hình 5.24	Gợi ý phim theo lịch sử tìm kiếm	52

Hình 5.25	Chức năng gợi ý phim dựa vào lịch sử thao tác của người dùng	52
Hình 5.26	Chức năng gợi ý phim dựa vào nội dung phim đang xem	53
Hình 5.27	Chức năng thêm phim vào danh sách yêu thích	53
Hình 5.28	Chức năng review phim	54
Hình 5.29	Chức năng xem toàn bộ ảnh của phim	54
Hình 6.1	Tạo PAT Github	61
Hình 6.2	Tạo Secrets Repository	62
Hình 6.3	Hệ thống tự động crawl dữ liệu	62
Hình 6.4	Cập nhật dữ liệu mới thành công	63
Hình 6.5	Kiểm tra kết quả 1	64
Hình 6.6	Kiểm tra kết quả 2	65
Hình 6.7	Kiểm tra kết quả 3	66

DANH MỤC BẢNG BIỂU

Bảng 2.1	Khảo sát các hệ thống đã có	7
Bảng 2.2	Đặc tả use case Đăng nhập	13
Bảng 2.3	Đặc tả use case Chính sửa thông tin cá nhân	14
Bảng 2.4	Đặc tả use case Tìm kiếm phim	15
Bảng 2.5	Đặc tả use case Đánh dấu phim mình thích	16
Bảng 2.6	Đặc tả use case Đánh giá phim	17
Bảng 2.7	Đặc tả use case CRUD phim	18
Bảng 2.8	Đặc tả use case Gợi ý phim theo nội dung	19
Bảng 5.1	Tổng hợp các thông tin cần lấy	42
Bảng 5.2	Danh sách thư viện và công cụ sử dụng	48
Bảng 5.3	Kiểm thử chức năng	55
Bảng 5.4	Kiểm thử chức năng	56
Bảng 5.5	Bảng danh sách kiểm thử hiệu năng	57
Bảng 5.6	Bảng kết quả kiểm thử hiệu năng	58
Bảng 7.1	Kết quả so sánh so với các ứng dụng khác	67

DANH MỤC THUẬT NGỮ VÀ TỪ VIỆT TẮT

Thuật ngữ	Ý nghĩa
API	Giao diện lập trình ứng dụng (Application Programming Interface)
CBF	Một kỹ thuật gợi ý trong hệ thống đề xuất (Content-Based Filtering)
CF	Một kỹ thuật gợi ý trong hệ thống đề xuất (Collaborative Filtering)
CI/CD	Continuous Integration/Continuous Deployment
CRUD	Create, Read, Update, Delete
CSRF	Cross-Site Request Forgery
CSS	Ngôn ngữ được sử dụng để tìm và định dạng lại các phần tử được tạo ra bởi các ngôn ngữ đánh dấu (Cascading Style Sheets)
HTML	Ngôn ngữ đánh dấu siêu văn bản (HyperText Markup Language)
ORM	Object-Relational Mapping
RMSE	Độ lệch bình phương trung bình gốc (Root Mean Square Error)
XSS	Cross-Site Scripting

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Đặt vấn đề

Trong bối cảnh phát triển mạnh mẽ của ngành công nghiệp giải trí, số lượng phim sản xuất và phát hành ngày càng tăng, tạo ra một kho tàng nội dung khổng lồ và đa dạng. Điều này dẫn đến thách thức lớn cho người tiêu dùng khi lựa chọn những bộ phim phù hợp với sở thích cá nhân. Với hàng ngàn tựa phim mới xuất hiện hàng năm, việc tìm kiếm và chọn lọc phim không chỉ tốn thời gian mà còn gây khó khăn trong việc xác định đâu là những bộ phim đáng xem.

Các hệ thống giới thiệu, quản lý, gợi ý phim hiện nay đã phần nào giúp người dùng tìm kiếm và khám phá nội dung mới. Tuy nhiên, những hệ thống này thường chỉ tập trung vào một khía cạnh nhất định. Như trang web BoxOfficeMojo [1] tổng hợp các thông tin cơ bản về phim nhưng thiếu hệ thống gợi ý và review từ người dùng, cũng như thông tin chi tiết về nơi sản xuất. Tương tự, TheNumbers [2] cung cấp thông tin cơ bản như thể loại phim nhưng hạn chế hơn so với BoxOfficeMojo. IMDb [3] có hầu như toàn bộ tính năng cần thiết như gợi ý phim, đánh giá và review từ người dùng, nhưng tốc độ tải trang chậm làm ảnh hưởng đến trải nghiệm người dùng. Metacritic [4] và RottenTomatoes [5] tập trung chủ yếu vào tổng hợp đánh giá của các chuyên gia trước khi phim phát hành, thiếu thông tin phong phú từ cộng đồng người dùng. Themoviedb [6] cung cấp thông tin phim dưới dạng API dành cho nhà phát triển, không phải là một trang web dành cho người

Nhìn chung, các trang web hiện tại tuy có nhiều tính năng hữu ích nhưng đều có những hạn chế nhất định, như thiếu hệ thống gợi ý, thông tin không đầy đủ, tốc độ tải trang chậm, đặc biệt chưa có truy vấn bằng ngôn ngữ người dùng để tìm phim phù hợp. Sự phân tán thông tin này làm giảm hiệu quả và trải nghiệm người dùng, khi họ phải truy cập nhiều nguồn khác nhau để có cái nhìn toàn diện về một bộ phim. Điều này tạo ra nhu cầu cho một ứng dụng giới thiệu, gợi ý phim toàn diện hơn, tích hợp đầy đủ các chức năng cần thiết và tối ưu hóa trải nghiệm người dùng.

Nếu vấn đề này được giải quyết, người dùng sẽ tiết kiệm được thời gian và nâng cao trải nghiệm khi tìm kiếm phim. Một ứng dụng gợi ý phim toàn diện không chỉ giúp người xem dễ dàng tìm thấy phim yêu thích mà còn cung cấp đầy đủ thông tin như đánh giá, trailer, và review từ cộng đồng, tạo ra một nền tảng thông tin phong phú và tiện lợi.

Việc xây dựng một hệ thống như vậy sẽ đáp ứng nhu cầu ngày càng cao của người dùng, đồng thời mở ra cơ hội ứng dụng rộng rãi trong lĩnh vực giải trí nói

chung và mở rộng ứng dụng nói riêng, góp phần thúc đẩy sự phát triển của ngành công nghiệp giải trí và các ngành liên quan.

1.2 Mục tiêu và phạm vi đề tài

Dựa vào phân tích các vấn đề nêu trên, ta có thể kết luận ra rằng, vì muốn đáp ứng những nhu cầu cụ thể nên các trang web nêu trên chỉ có các chức năng chuyên biệt dựa vào các dữ liệu cụ thể.

Trên cơ sở đó, đề tài hướng tới việc phát triển một ứng dụng gợi ý phim toàn diện, khắc phục các hạn chế hiện tại bằng cách kết hợp các chức năng cũng như dữ liệu của những trang web trên vào một nền tảng duy nhất, đồng thời phát triển thêm các chức năng gợi ý, truy vấn kèm theo. Hệ thống này sẽ bao gồm tìm kiếm, xem chi tiết phim, gợi ý phim cá nhân hóa, đánh giá và review phim từ cộng đồng, xem trailer và hình ảnh phim. Mục tiêu là cung cấp một nền tảng tìm kiếm phim bằng ngôn ngữ người dùng và gợi ý phim chính xác với mong muốn của người dùng, giúp người dùng tìm được phim phù hợp với sở thích cá nhân một cách nhanh nhất và chính xác nhất. Ngoài ra, các bộ phim được gợi ý phải có thông tin đầy đủ, đáng tin cậy và được xếp hạng bởi cộng đồng người dùng.

Phạm vi là một ứng dụng gợi ý phim bằng genAI có số lượng phim khoảng 5000 bộ phim. Tập trung vào các bộ phim nổi bật nhất thời điểm hiện tại. Đối tượng sử dụng là người dùng thường xuyên quan tâm đến phim ảnh.

1.3 Định hướng giải pháp

Để giải quyết các hạn chế đã được xác định, tôi đề xuất định hướng giải pháp cho các vấn đề cụ thể như sau:

- **Tìm kiếm phim bằng ngôn ngữ người dùng:** Sử dụng AI để phân tích truy vấn, kết hợp vector search dựa trên embeddings và lọc có cấu trúc để trả về kết quả chính xác.
- **Gợi ý phim cá nhân hóa:** Theo dõi hành động người dùng để gợi ý phim cá nhân hóa.
- **Dữ liệu đầy đủ:** Thu thập và tổng hợp thông tin phim ảnh từ các nguồn uy tín để xây dựng cơ sở dữ liệu phong phú, đa dạng và chính xác.
- **Hiệu năng trang web tốt:** Cải thiện hiệu năng của hệ thống bằng cách rút gọn mã nguồn, thông tin, tối ưu hóa các thuật toán và sử dụng các kỹ thuật lập trình hiệu quả nhất, đảm bảo tốc độ tải trang nhanh và hiệu quả.
- **Có hệ thống gợi ý phim:** Sử dụng thuật toán cho hệ thống gợi ý phim: Content-Based Filtering. Content-Based Filtering sẽ được áp dụng khi người

dùng xem thông tin về một bộ phim cụ thể, từ đó gợi ý các bộ phim có nội dung tương tự.

- **Tích hợp các chức năng quan trọng:** Phát triển một nền tảng web tích hợp các tính năng quan trọng như xem chi tiết phim, gợi ý phim cá nhân hóa, đánh giá và review từ cộng đồng, cùng với việc xem trailer và hình ảnh phim.

Đóng góp chính của đồ án là xây dựng một ứng dụng gợi ý phim toàn diện, không chỉ cung cấp thông tin phong phú và đa dạng về phim mà còn mang lại trải nghiệm người dùng tối ưu thông qua các gợi ý phim chính xác và cá nhân hóa. Kết quả đạt được sẽ là một nền tảng duy nhất, tích hợp tất cả các chức năng cần thiết, giúp người dùng dễ dàng tìm kiếm và khám phá những bộ phim phù hợp với sở thích cá nhân một cách nhanh chóng và hiệu quả.

1.4 Bố cục đồ án

Phần còn lại của báo cáo đồ án tốt nghiệp này tôi sẽ tổ chức như sau:

Chương 2 tôi sẽ trình bày về phần khảo sát và phân tích yêu cầu. Trong chương này, tôi sẽ khảo sát hiện trạng của các hệ thống giới thiệu phim hiện có, từ đó xây dựng tổng quan các chức năng cần thiết. Nội dung bao gồm biểu đồ use case tổng quan, phân rã các chức năng và quy trình nghiệp vụ. Đồng thời, tôi cũng sẽ đặc tả các chức năng chính và xác định các yêu cầu phi chức năng, đặc biệt là chức năng gợi ý phim.

Chương 3 tôi sẽ giới thiệu về nền tảng lý thuyết cho gợi ý phim. Chương này tập trung vào việc trình bày thuật toán và phương pháp được sử dụng trong hệ thống gợi ý phim, đó là lọc cộng tác (Collaborative Filtering). Tôi sẽ giải thích cách thuật toán này hoạt động và lý do chọn thuật toán này cho hệ thống của mình.

Chương 4 tôi sẽ đi sâu vào các công nghệ sử dụng trong quá trình phát triển hệ thống. Tôi sẽ giới thiệu các công nghệ, công cụ và thư viện đã được lựa chọn để xây dựng hệ thống, cũng như lý do lựa chọn các công nghệ này. Chương này cung cấp cái nhìn tổng quan về nền tảng kỹ thuật của hệ thống.

Chương 5 tôi sẽ tập trung giới thiệu phần thiết kế, triển khai và đánh giá hệ thống. Nội dung bao gồm thiết kế kiến trúc hệ thống với việc lựa chọn kiến trúc phần mềm, thiết kế tổng quan và thiết kế chi tiết các gói. Tôi cũng sẽ trình bày thiết kế chi tiết về giao diện người dùng, lớp và cơ sở dữ liệu. Phần xây dựng ứng dụng sẽ minh họa các chức năng chính và kết quả đạt được sẽ bao gồm một phần đánh giá thực nghiệm của hệ thống gợi ý. Chương này cũng bao gồm các phần kiểm thử và triển khai hệ thống.

Chương 6 tập trung làm rõ các giải pháp đã được áp dụng và những đóng góp

quan trọng trong suốt quá trình phát triển dự án. Ngoài ra, chương cũng thực hiện phân tích toàn diện và chi tiết về bài toán gợi ý phim.

Chương 7 đóng vai trò là phần tổng kết và đề xuất các hướng phát triển tiếp theo. Nội dung chương bao gồm việc tổng hợp các kết quả quan trọng đã đạt được trong quá trình thực hiện đồ án, cùng với việc đưa ra các đề xuất về các hướng phát triển có tiềm năng cho hệ thống trong thời gian tới. Thông qua chương này, người đọc sẽ có được cái nhìn toàn diện về giá trị đóng góp của đồ án và những bước tiếp theo cần thực hiện nhằm cải thiện và nâng cấp hệ thống.

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

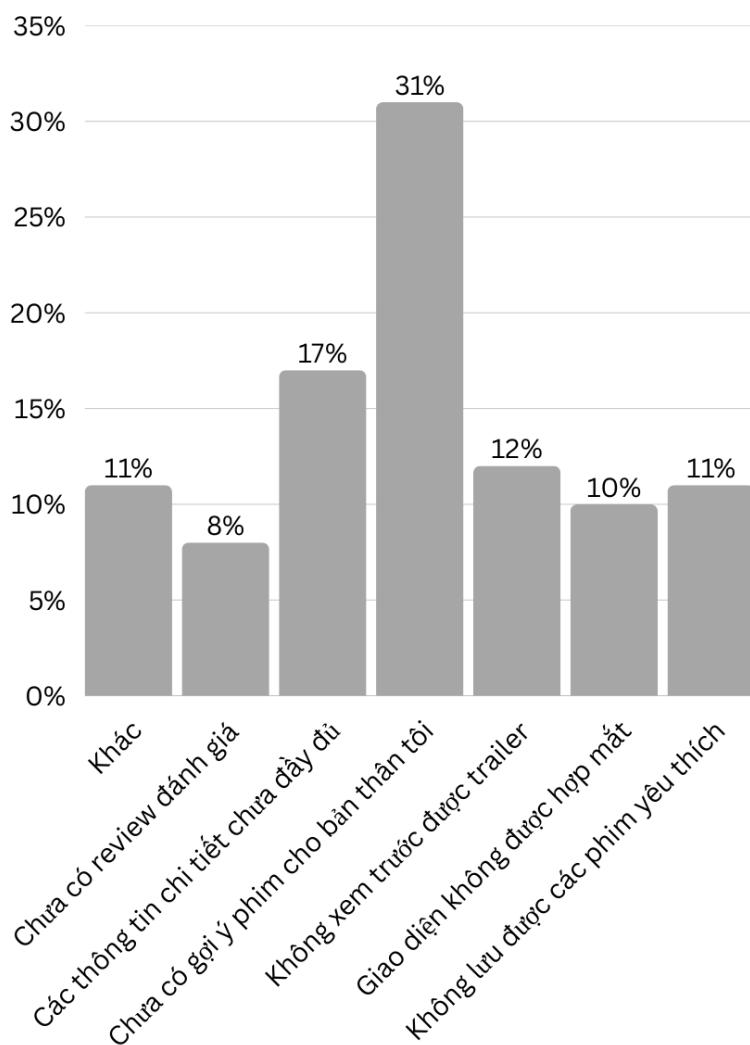
Chương này tập trung vào việc khảo sát và phân tích các yêu cầu cần thiết để xây dựng ứng dụng gợi ý phim. Nội dung chương được tổ chức theo trình tự bắt đầu từ việc nghiên cứu và đánh giá các hệ thống gợi ý phim đang tồn tại trên thị trường, qua đó xác định được những điểm mạnh và điểm yếu của từng hệ thống. Dựa trên kết quả khảo sát, tôi sẽ xây dựng tổng quan về các chức năng cần thiết thông qua việc mô hình hóa bằng biểu đồ use case tổng quan, đồng thời tiến hành phân rã chi tiết các chức năng và các quy trình nghiệp vụ liên quan. Tiếp theo, mỗi chức năng sẽ được đặc tả một cách chi tiết và cụ thể, đảm bảo mọi yêu cầu đều được xác định đầy đủ và chính xác. Phần cuối của chương sẽ trình bày về các yêu cầu phi chức năng, trong đó tập trung vào chức năng gợi ý phim, nhằm đảm bảo hệ thống không chỉ thỏa mãn các yêu cầu chức năng cơ bản mà còn cung cấp trải nghiệm người dùng tối ưu nhất.

2.1 Khảo sát hiện trạng

73	Bùi Ánh Tuyết	tuyet.buianh@gmail.com	2026-01-16	Chưa có review đánh giá
74	Ngô Bảo Châu	chau.ngobao@gmail.com	2026-01-03	Không xem trước được trailer
75	Hoàng Minh Khôi	khoi.hoangminh@gmail.com	2026-01-17	Chưa có gợi ý phim cho bản thân tôi
76	Đặng Việt Dũng	dung.dangviet@gmail.com	2026-01-01	Giao diện không được hợp mắt
77	Trịnh Thu Trang	trang.trinhthu@gmail.com	2026-01-19	Các thông tin chi tiết chưa đầy đủ
78	Lý Hải Nam	nam.lyhai@gmail.com	2026-01-05	Chưa có gợi ý phim cho bản thân tôi
79	Đinh Thúy Quỳnh	quynh.dinhthuy@gmail.com	2026-01-12	Khác
80	Dương Văn Hậu	hau.duongvan@gmail.com	2026-01-08	Không lưu được các phim yêu thích
81	Võ Hoài Lâm	lam.vohoai@gmail.com	2026-01-15	Chưa có gợi ý phim cho bản thân tôi
82	Nguyễn Lan Hương	huong.nguyenlan@gmail.com	2026-01-02	Các thông tin chi tiết chưa đầy đủ
83	Trần Tuấn Tú	tu.trantuan@gmail.com	2026-01-18	Không xem trước được trailer
84	Lê Diệp Anh	anh.lediep@gmail.com	2026-01-10	Chưa có gợi ý phim cho bản thân tôi
85	Phạm Tiến Thành	thanh.phamtien@gmail.com	2026-01-04	Giao diện không được hợp mắt
86	Vũ Ngọc Diệp	diep.vungoc@gmail.com	2026-01-14	Khác
87	Đỗ Hữu Cảnh	canh.dohuu@gmail.com	2026-01-07	Không lưu được các phim yêu thích
88	Phan Thị Mơ	mo.phantri@gmail.com	2026-01-11	Chưa có gợi ý phim cho bản thân tôi
89	Bùi Thế Anh	anh.buithe@gmail.com	2026-01-09	Các thông tin chi tiết chưa đầy đủ
90	Ngô Kiến Huy	huy.ngokien@gmail.com	2026-01-13	Không xem trước được trailer
91	Hoàng Thị Loan	loan.hoangthi@gmail.com	2026-01-06	Chưa có review đánh giá
92	Đặng Quang Thắng	thang.dangquang@gmail.com	2026-01-16	Chưa có gợi ý phim cho bản thân tôi
93	Trịnh Kim Chi	chi.trinhkim@gmail.com	2026-01-03	Khác
94	Lý Văn Sâm	sam.lyvan@gmail.com	2026-01-17	Các thông tin chi tiết chưa đầy đủ
95	Đinh Nhật Tân	tan.dinhnhat@gmail.com	2026-01-01	Chưa có gợi ý phim cho bản thân tôi
96	Dương Mỹ Linh	linh.duongmy@gmail.com	2026-01-19	Không lưu được các phim yêu thích
97	Võ Văn Kiệt	kiet.vovan@gmail.com	2026-01-05	Chưa có gợi ý phim cho bản thân tôi
98	Nguyễn Thị Định	dinh.nguyenthid@gmail.com	2026-01-12	Giao diện không được hợp mắt
99	Trần Văn Hùng	hung.tranvan@gmail.com	2026-01-08	Không xem trước được trailer
100	Lê Quang Liêm	liem.lequang@gmail.com	2026-01-15	Khác
101	Phạm Thị Thảo	thao.phamthi@gmail.com	2026-01-02	Chưa có review đánh giá
...				

Hình 2.1: Form khảo sát người dùng về các ứng dụng

Để thực hiện khảo sát hiện trạng và xác định các yêu cầu cho ứng dụng gợi ý phim, tôi đã tiến hành thu thập dữ liệu từ ba nguồn thông tin chính bao gồm: phản hồi từ người dùng và khách hàng, đánh giá các hệ thống hiện có trên thị trường,



Hình 2.2: Khảo sát người dùng về các ứng dụng

và nghiên cứu các ứng dụng tương tự. Dựa trên nguồn dữ liệu đã thu thập, tôi tiến hành phân tích một cách toàn diện, thực hiện so sánh giữa các giải pháp và đánh giá kỹ lưỡng các mặt ưu điểm cũng như hạn chế của từng sản phẩm và nghiên cứu liên quan.

Kết quả khảo sát từ phía người dùng và khách hàng cho thấy, đối tượng này kỳ vọng một ứng dụng có thể thực hiện tìm kiếm phim với tốc độ cao và độ chính xác tốt, kèm theo đó là hệ thống gợi ý phim được cá nhân hóa dựa trên sở thích riêng và lịch sử xem phim của từng người. Bên cạnh đó, các chức năng như đánh giá phim, viết nhận xét và xem trailer cũng nhận được sự quan tâm đáng kể từ phía người dùng. Ngoài ra, việc nâng cao hiệu năng hệ thống và thiết kế giao diện thân thiện, dễ sử dụng cũng là những yêu cầu quan trọng được đề xuất.

73	Bùi Ánh Tuyết	tuyet.buianh@gmail.com	2026-01-16	Không theo dõi hành vi người dùng
74	Ngô Bảo Châu	chau.ngobao@gmail.com	2026-01-03	Theo dõi hành vi người dùng có cam kết
75	Hoàng Minh Khôi	khoi.hoangminh@gmail.com	2026-01-17	Không theo dõi hành vi người dùng
76	Đặng Việt Dũng	dung.dangviet@gmail.com	2026-01-01	Không theo dõi hành vi người dùng
77	Trịnh Thu Trang	trang.trinhthu@gmail.com	2026-01-19	Theo dõi hành vi người dùng có cam kết
78	Lý Hải Nam	nam.lyhai@gmail.com	2026-01-05	Không theo dõi hành vi người dùng
79	Đinh Thúy Quỳnh	quynh.dinhthuy@gmail.com	2026-01-12	Không theo dõi hành vi người dùng
80	Dương Văn Hậu	hau.duongvan@gmail.com	2026-01-08	Theo dõi hành vi người dùng có cam kết
81	Võ Hoài Lâm	lam.vohoi@gmail.com	2026-01-15	Không theo dõi hành vi người dùng
82	Nguyễn Lan Hương	huong.nguyenlan@gmail.com	2026-01-02	Không theo dõi hành vi người dùng
83	Trần Tuấn Tú	tu.trantuan@gmail.com	2026-01-18	Theo dõi hành vi người dùng có cam kết
84	Lê Diệp Anh	anh.lediep@gmail.com	2026-01-10	Không theo dõi hành vi người dùng
85	Phạm Tiến Thành	thanh.phamtien@gmail.com	2026-01-04	Không theo dõi hành vi người dùng
86	Vũ Ngọc Diệp	diep.vungoc@gmail.com	2026-01-14	Theo dõi hành vi người dùng có cam kết
87	Đỗ Hữu Cảnh	canh.dohuu@gmail.com	2026-01-07	Không theo dõi hành vi người dùng
88	Phan Thị Mơ	mo.phanthi@gmail.com	2026-01-11	Không theo dõi hành vi người dùng
89	Bùi Thế Anh	anh.buithe@gmail.com	2026-01-09	Theo dõi hành vi người dùng có cam kết
90	Ngô Kiến Huy	huy.ngokien@gmail.com	2026-01-13	Không theo dõi hành vi người dùng
91	Hoàng Thị Loan	loan.hoangthi@gmail.com	2026-01-06	Không theo dõi hành vi người dùng
92	Đặng Quang Thắng	thang.dangquang@gmail.com	2026-01-16	Theo dõi hành vi người dùng có cam kết
93	Trịnh Kim Chi	chi.trinhkim@gmail.com	2026-01-03	Không theo dõi hành vi người dùng
94	Lý Văn Sâm	sam.lyvan@gmail.com	2026-01-17	Không theo dõi hành vi người dùng
95	Đinh Nhật Tân	tan.dinhnhat@gmail.com	2026-01-01	Theo dõi hành vi người dùng có cam kết
96	Dương Mỹ Linh	linh.duongmy@gmail.com	2026-01-19	Không theo dõi hành vi người dùng
97	Võ Văn Kiệt	kiet.vovan@gmail.com	2026-01-05	Không theo dõi hành vi người dùng
98	Nguyễn Thị Định	dinh.nguyenthi@gmail.com	2026-01-12	Theo dõi hành vi người dùng có cam kết
99	Trần Văn Hùng	hung.tranvan@gmail.com	2026-01-08	Không theo dõi hành vi người dùng
100	Lê Quang Liêm	liem.lequang@gmail.com	2026-01-15	Không theo dõi hành vi người dùng
101	Phạm Thị Thảo	thao.phamthi@gmail.com	2026-01-02	Theo dõi hành vi người dùng có cam kết
...				

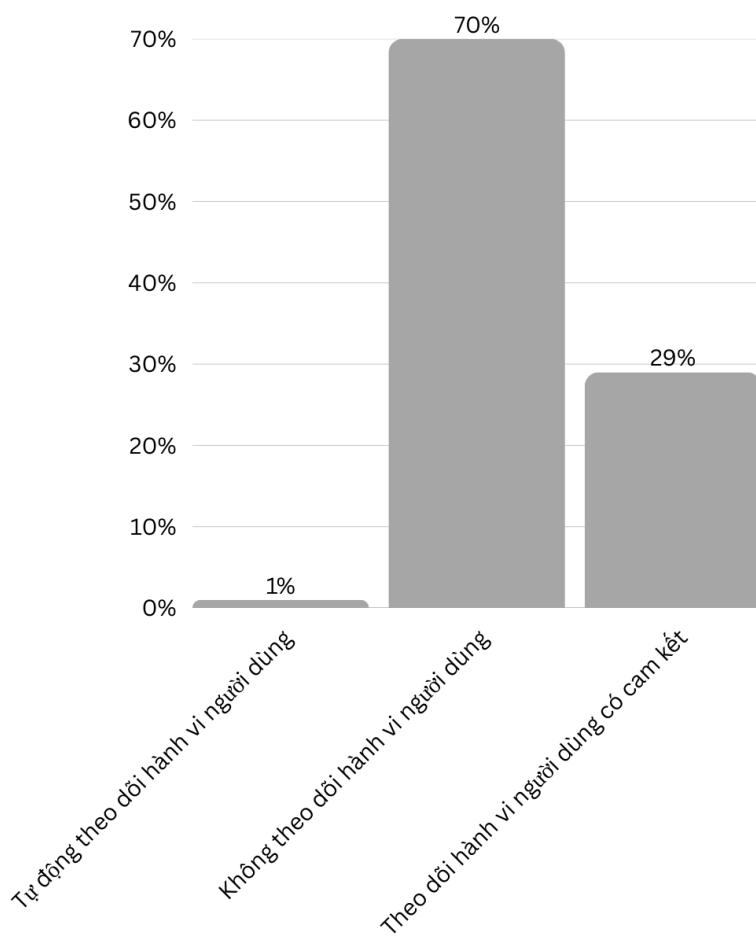
Hình 2.3: Form khảo sát về quyền theo dõi

2.2 Khảo sát về quyền theo dõi

Theo kết quả của khảo sát về quyền theo dõi, tôi nhận thấy rằng người dùng mong muốn một ứng dụng gợi ý phim, nhưng phần lớn lại không mong muốn bị theo dõi hành vi trên ứng dụng. Vì vậy, tôi sẽ phát triển thêm chức năng theo dõi hành vi người dùng để gợi ý phim một cách phù hợp nhất với mong muốn người dùng.

Bảng 2.1: Khảo sát các hệ thống đã có

Tính năng	IMDb	Metacritic	Rotten Tomatoes	TMDb	Box Office Mojo	The Numbers
Thông tin phim	Đầy đủ	Đầy đủ	Đầy đủ	Đầy đủ nhưng phải lấy bằng API	Cơ bản	Thiểu số
Gợi ý phim	Có	Không	Không	Không	Không	Không
Review	Có	Có	Có	Có	Không	Không
Trailer	Có	Có	Có	Có	Không	Không
Hiệu năng	Chậm	Vừa phải	Vừa phải	Ôn	Vừa phải	Ôn



Hình 2.4: Khảo sát về quyền theo dõi

Mặt khác, việc nghiên cứu và đánh giá các hệ thống hiện có bao gồm IMDb, Metacritic, Rotten Tomatoes, TMDb, Box Office Mojo và The Numbers đã được thực hiện nhằm so sánh và phân tích các tính năng, hiệu suất hoạt động cũng như khả năng cung cấp thông tin của mỗi hệ thống. Kết quả phân tích cho thấy rằng mỗi hệ thống đều sở hữu những ưu điểm và nhược điểm đặc trưng, và việc tích hợp các tính năng ưu việt từ các hệ thống này có tiềm năng tạo ra một giải pháp gợi ý phim mang tính toàn diện và hiệu quả.

Bên cạnh đó, một số ứng dụng giới thiệu phim trên các nền tảng di động và web cũng đã được khảo sát, qua đó phát hiện ra những đặc điểm tương đồng liên quan đến thiết kế giao diện thân thiện với người dùng và các tính năng gợi ý phim được xây dựng dựa trên phân tích hành vi người dùng như được thể hiện trong Table 2.1. Tuy vậy, phần lớn các ứng dụng này vẫn còn tồn tại những hạn chế về khả năng xử lý hiệu suất và sự đa dạng trong các tính năng được cung cấp.

Tổng hợp từ các kết quả khảo sát đã thu thập được, các tính năng quan trọng cần được tích hợp vào hệ thống đã được xác định một cách rõ ràng, nhằm mục tiêu phát triển một giải pháp gợi ý phim toàn diện có khả năng đáp ứng một cách hiệu quả

các nhu cầu đa dạng của người dùng.

2.3 Tổng quan chức năng

2.3.1 Biểu đồ use case tổng quát

Các tác nhân tham gia:

- Guest (Khách hàng): Người dùng không có tài khoản trong hệ thống, truy cập vào hệ thống để tìm kiếm thông tin về phim.
- User (Người dùng): Người dùng đã đăng nhập vào hệ thống, có thể thực hiện các tác vụ như đánh giá phim, lưu phim vào danh sách yêu thích, thay đổi thông tin cá nhân, và tương tác với các tính năng khác của hệ thống.
- Admin (Quản trị viên): Người quản lý hệ thống, có quyền truy cập và quản lý tất cả các thông tin trong hệ thống.

Vai Trò của Mỗi Tác Nhân:

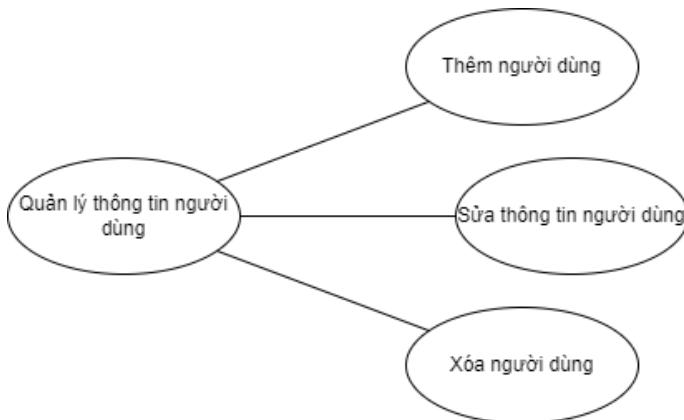
- Guest: Tác nhân này có vai trò chính là tìm kiếm thông tin về phim và xem các bộ phim mà không cần đăng nhập vào hệ thống.
- User: Tác nhân này có vai trò tương tự như Guest, nhưng cũng có thêm quyền đánh giá phim, thêm phim vào danh sách yêu thích, và quản lý thông tin cá nhân.
- Admin: Tác nhân này có vai trò quản lý hệ thống, bao gồm quản lý thông tin phim và quản lý thông tin người dùng.

Các use case chính:

- Xem thông tin các phim: Cả Guest và User đều có thể xem thông tin chi tiết về các bộ phim như nội dung, đạo diễn, diễn viên, phần thưởng, doanh thu, review, ...
- Tương tác với phim: User có thể đánh giá phim và lưu phim vào danh sách yêu thích.
- Gợi ý phim: Hệ thống đưa ra những gợi ý dựa vào thao tác của người dùng như khi người dùng vào xem một bộ phim nào đó hoặc đánh giá một bộ phim nào đó.

**Hình 2.5:** Use case tổng quan

2.3.2 Biểu đồ use case phân rã Quản lý thông tin người dùng

**Hình 2.6:** Phân rã use case Quản lý thông tin người dùng

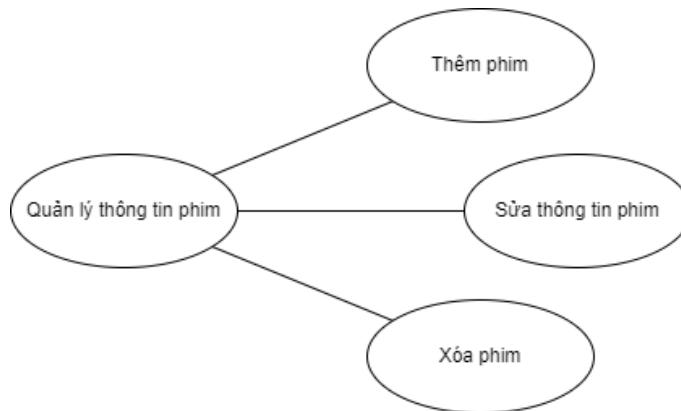
Trong biểu đồ use case phân rã "Quản lý thông tin người dùng", các use case được phân rã và mô tả như sau:

- Thêm người dùng: Admin có quyền thêm người dùng mới vào hệ thống bằng cách cung cấp thông tin cần thiết như tên đăng nhập, email và mật khẩu.
- Sửa thông tin người dùng: Admin có thể chỉnh sửa thông tin cá nhân của người

dùng, bao gồm cả tên đăng nhập, email, và mật khẩu.

- Xóa người dùng: Admin có khả năng xóa người dùng khỏi hệ thống.

2.3.3 Biểu đồ use case phân rã Quản lý thông tin phim

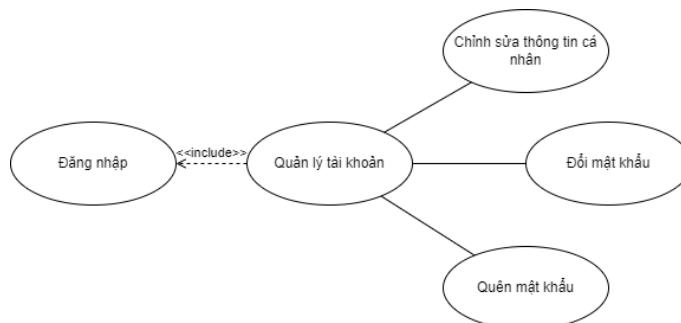


Hình 2.7: Phân rã use case Quản lý thông tin phim

Trong biểu đồ use case phân rã "Quản lý thông tin phim", các use case được phân rã và mô tả như sau:

- Thêm phim: Admin có khả năng thêm phim mới vào hệ thống bằng cách cung cấp thông tin chi tiết về phim như tiêu đề, nội dung, thể loại, đạo diễn, diễn viên, năm sản xuất, hình ảnh, ...
- Sửa thông tin phim: Admin có thể chỉnh sửa thông tin chi tiết của một bộ phim đã tồn tại trong hệ thống, bao gồm cả các thông tin như thể loại, đạo diễn, diễn viên và nội dung.
- Xóa phim: Admin có khả năng xóa một bộ phim khỏi hệ thống.

2.3.4 Biểu đồ use case phân rã Quản lý tài khoản

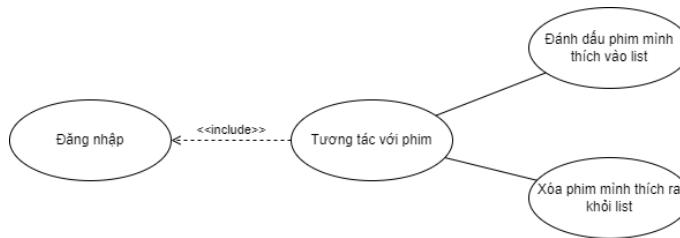


Hình 2.8: Phân rã use case Quản lý tài khoản

Trong biểu đồ use case phân rã "Quản lý tài khoản", các use case được phân rã và mô tả như sau:

- Quên mật khẩu: Người dùng có thể khôi phục mật khẩu bằng cách yêu cầu gửi liên kết đặt lại mật khẩu qua email.
- Đổi mật khẩu: Người dùng có thể thay đổi mật khẩu hiện tại bằng cách nhập mật khẩu cũ và mật khẩu mới.
- Chính sửa thông tin cá nhân: Người dùng có thể chỉnh sửa thông tin cá nhân của mình như tên, địa chỉ email, ảnh đại diện, ...

2.3.5 Biểu đồ use case phân rã Tương tác với phim



Hình 2.9: Phân rã use case Tương tác với phim

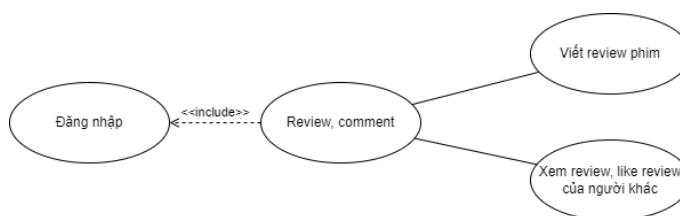
Trong biểu đồ use case phân rã "Tương tác với phim", các use case được phân rã và mô tả như sau:

- Đánh dấu phim yêu thích: Người dùng có khả năng đánh dấu các bộ phim yêu thích vào danh sách riêng của mình để dễ dàng theo dõi sau này và hệ thống có thể đưa ra gợi ý những bộ phim có khả năng mà người dùng thích.
- Xóa phim yêu thích: Người dùng có khả năng xóa đi những bộ phim đã từng thêm vào danh sách yêu thích.

Đánh dấu phim yêu thích: Người dùng có khả năng đánh dấu các bộ phim yêu thích vào danh sách riêng của mình để dễ dàng theo dõi sau này và hệ thống có thể đưa ra gợi ý những bộ phim có khả năng mà người dùng thích.

Xóa phim yêu thích: Người dùng có khả năng xóa đi những bộ phim đã từng thêm vào danh sách yêu thích.

2.3.6 Biểu đồ use case phân rã Review, comment



Hình 2.10: Phân rã use case Review, comment

Trong biểu đồ use case phân rã "Review, comment", các use case được phân rã và mô tả như sau:

- Viết review phim: Người dùng có khả năng viết và đăng review về một bộ phim sau khi xem. Họ có thể nhập nội dung review vào và đăng lên hệ thống để chia sẻ ý kiến của mình với cộng đồng người dùng khác.
- Like review: Người dùng có thể thể hiện sự đồng tình hoặc ủng hộ bằng cách like các review mà họ thấy ý nghĩa hoặc hữu ích.

2.4 Đặc tả chức năng

2.4.1 Đặc tả use case Đăng nhập

Bảng 2.2: Đặc tả use case Đăng nhập

Mã Use case	UC002		Tên Use case	Đăng nhập
Tác nhân	User			
Tiền điều kiện	Người dùng đã có tài khoản trên hệ thống.			
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động	
	1.	Người dùng	Người dùng truy cập trang đăng nhập (login page).	
	2.	Hệ thống	Hệ thống hiển thị giao diện đăng nhập yêu cầu người dùng nhập thông tin đăng nhập (username và password).	
	3.	Người dùng	Người dùng nhập thông tin đăng nhập và bấm nút "Đăng nhập".	
	4.	Hệ thống	Hệ thống xác thực thông tin đăng nhập của người dùng.	
	1.	Hệ thống	Nếu thông tin đăng nhập chính xác, hệ thống cho phép người dùng truy cập vào tài khoản và chuyển hướng người dùng đến trang chính (home page).	
	2.	Hệ thống	Nếu thông tin đăng nhập không chính xác, hệ thống hiển thị thông báo lỗi và yêu cầu người dùng nhập lại thông tin đăng nhập.	
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động	
	3a.	Người dùng	Nếu người dùng quên mật khẩu, họ có thể yêu cầu đặt lại mật khẩu bằng cách nhấn vào liên kết "Quên mật khẩu" trên trang đăng nhập.	
	4a.	Hệ thống	Nếu tên đăng nhập không tồn tại trong hệ thống, hệ thống hiển thị thông báo lỗi và yêu cầu người dùng nhập lại thông tin đăng nhập.	
Hậu điều kiện	5a.	Hệ thống	Nếu mật khẩu nhập vào không chính xác, hệ thống hiển thị thông báo lỗi và yêu cầu người dùng nhập lại thông tin đăng nhập.	

Bảng trên (Table 2.2) là đặc tả use case Đăng nhập từ người dùng. Trong đó, người dùng có thể đăng nhập vào hệ thống bằng cách nhập tên đăng nhập và mật khẩu. Hệ thống sẽ kiểm tra tên đăng nhập và mật khẩu của người dùng và trả về kết quả đăng nhập. Sẽ validate tên đăng nhập và mật khẩu để tránh lỗi input không hợp lệ.

2.4.2 Đặc tả use case Chính sửa thông tin cá nhân

Bảng 2.3: Đặc tả use case Chính sửa thông tin cá nhân

Mã Use case	UC009		Tên Use case	Chỉnh sửa thông tin cá nhân
Tác nhân	Guest, Admin			
Tiêu điều kiện	Người dùng đã đăng nhập vào hệ thống và có thông tin cá nhân cần cập nhật.			
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động	
	1.	Người dùng	Người dùng truy cập vào trang cập nhật thông tin cá nhân	
	2.	Hệ thống	Hệ thống hiển thị form thông tin cá nhân hiện tại của người dùng.	
	3.	Người dùng	Người dùng thay đổi hoặc bổ sung thông tin cá nhân cần cập nhật trên form.	
	4.	Người dùng	Người dùng xác nhận thông tin đã nhập chính xác và yêu cầu cập nhật.	
	5.	Hệ thống	Hệ thống kiểm tra và lưu thông tin cá nhân mới của người dùng	
	6.	Hệ thống	Hệ thống hiển thị thông báo cập nhật thành công và cung cấp cho người dùng xem lại thông tin cá nhân đã cập nhật.	
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động	
	5a.	Hệ thống	Nếu người dùng nhập thông tin không hợp lệ, hệ thống hiển thị thông báo lỗi và yêu cầu người dùng sửa lại thông tin trước khi cập nhật.	
	5b.	Hệ thống	Nếu hệ thống gặp lỗi khi lưu thông tin cá nhân mới của người dùng, hệ thống hiển thị thông báo lỗi và yêu cầu người dùng thực hiện lại lần nữa sau khi đã khắc phục lỗi.	
Hậu điều kiện				

Bảng Table 2.3 mô tả chi tiết use case Chính sửa thông tin cá nhân được thực hiện bởi người dùng. Use case này cho phép người dùng thực hiện việc cập nhật các thông tin cá nhân bao gồm tên, địa chỉ email, ảnh đại diện và các thông tin khác. Hệ thống sẽ tiến hành xác thực và kiểm tra tính hợp lệ của các thông tin được cập nhật, sau đó phản hồi lại kết quả của quá trình chỉnh sửa. Việc kiểm tra tính hợp lệ của dữ liệu đầu vào được thực hiện nhằm ngăn chặn các lỗi do dữ liệu không đúng

định dạng hoặc không đáp ứng các yêu cầu đã quy định.

2.4.3 Đặc tả use case Tìm kiếm phim

Bảng 2.4: Đặc tả use case Tìm kiếm phim

Mã Use case	UC005		Tên Use case	Tìm kiếm phim
Tác nhân	Guest, User, Admin			
Tiền điều kiện	Không có			
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động	
	1.	Người dùng	Yêu cầu chức năng tìm kiếm phim	
	2.	Hệ thống	Hiển thị thanh tìm kiếm phía trên giao diện	
	3.	Người dùng	Nhập thông tin cần tìm kiếm và chọn nút tìm kiếm	
	4.	Hệ thống	Thực hiện tìm kiếm trong cơ sở dữ liệu của người dùng và hiển thị danh sách kết quả tìm kiếm thỏa mãn điều kiện	
	5.	Người dùng	Chọn phim cần xem thông tin chi tiết	
	6.	Hệ thống	Hiển thị giao diện thông tin chi tiết của phim như ảnh, đạo diễn, diễn viên, đánh giá, comment, ...	
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động	
Hậu điều kiện	Không			

Bảng trên (Table 2.4) là đặc tả use case Tìm kiếm phim từ người dùng. Trong đó, người dùng có thể tìm kiếm phim bằng cách nhập tên phim, tên diễn viên, tên đạo diễn, ... Hệ thống sẽ kiểm tra tên phim, tên diễn viên, tên đạo diễn của người dùng và trả về kết quả tìm kiếm. Sẽ validate tên phim, tên diễn viên, tên đạo diễn để tránh lỗi input không hợp lệ.

Ngoài ra còn có chức năng Tìm kiếm phim bằng ngôn ngữ người dùng, khi người dùng nhập vào ngôn ngữ người dùng, hệ thống sẽ phân tích nội dung người dùng nhập vào và trả về kết quả tìm kiếm.

2.4.4 Đặc tả use case Đánh dấu phim mình thích

Bảng 2.5: Đặc tả use case Đánh dấu phim mình thích

Mã Use case	UC010			Tên Use case	Đánh dấu phim mình thích
Tác nhân	User				
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống.				
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động		
	1.	Người dùng	Ở màn hình chính hoặc bất cứ đâu có nút thêm phim vào danh sách yêu thích, người dùng click chuột vào đó		
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động		
Hậu điều kiện					

Bảng Table 2.5 mô tả chi tiết use case Đánh dấu phim yêu thích được thực hiện bởi người dùng. Thông qua use case này, người dùng có khả năng lưu các bộ phim yêu thích vào một danh sách cá nhân riêng biệt, tạo điều kiện thuận lợi cho việc quản lý và truy cập lại sau này. Đồng thời, hành động này cũng cung cấp dữ liệu quan trọng để hệ thống có thể phân tích và đề xuất các bộ phim phù hợp với sở thích của người dùng.

2.4.5 Đặc tả use case Đánh giá phim

Bảng 2.6: Đặc tả use case Đánh giá phim

Mã Use case	UC008	Tên Use case	Đánh giá phim
Tác nhân	User		
Tiền điều kiện	Khách đăng nhập thành công với vai trò là User		
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1.	Người dùng	Chọn chức năng Đánh giá phim sau khi vào xem chi tiết phim
	2.	Hệ thống	Hiển thị ra giao diện đánh giá
	3.	Người dùng	Chọn số sao đánh giá và viết comment, sau đó nhấn gửi
	4.	Hệ thống	Thông báo đánh giá thành công và quay về màn hình chi tiết phim đó
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	4a.	Hệ thống	Thông báo lỗi: Cần đánh giá số sao nếu Người dùng không đánh giá sao (không điền vào ô comment cũng không sao)
Hậu điều kiện	Không		

Bảng trên (Table 2.6) là đặc tả use case Đánh giá phim từ người dùng. Trong đó, người dùng có thể đánh giá sao, để lại bình luận sau khi xem. Hệ thống sẽ kiểm tra đánh giá của người dùng và trả về kết quả đánh giá. Sẽ validate đánh giá và bình luận để tránh lỗi input không hợp lệ.

2.4.6 ĐẶC TẢ USE CASE CRUD PHIM

Bảng 2.7: Đặc tả use case CRUD phim

Mã Use case	UC013	Tên Use case	CRUD phim
Tác nhân	Admin		
Tiền điều kiện	Người dùng đăng nhập với vai trò Admin		
Sửa:			
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1.	Admin	Chọn một phim và yêu cầu chỉnh sửa hoặc click chuột vào biểu tượng cây bút trong banner phim
	2.	Hệ thống	Hiển thị ra danh sách chi tiết nội dung phim cũ, và màn hình chỉnh sửa nội dung
	3.	Admin	Chỉnh sửa các thông tin của phim
	4.	Hệ thống	Kiểm tra các trường bắt buộc nhập
	5.	Hệ thống	Kiểm tra các trường cần đảm bảo duy nhất
	6.	Hệ thống	Cập nhật các thông tin cần chỉnh sửa và thông báo chỉnh sửa thành công.
Luồng sự kiện thay thế	5a	Hệ thống	Thông báo lỗi: chưa nhập đủ các trường cần thiết nếu Admin nhập thiếu trường
	6a	Hệ thống	Thông báo lỗi: thông báo cụ thể trường nào không duy nhất nếu Admin nhập trùng
Xoá:			
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1.	Admin	Chọn một phim và yêu cầu xoá hoặc click chuột vào biểu tượng thùng rác trong banner phim
	2.	Hệ thống	Hiển thị thông báo yêu cầu Admin xác nhận việc xoá
	3.	Admin	Xác nhận xoá phim
	4.	Hệ thống	Xoá phim ra khỏi database và thông báo xoá thành công
Luồng sự kiện thay thế	4a	Hệ thống	Kết thúc use case nếu Admin xác nhận không xoá
Thêm:			
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1.	Admin	Yêu cầu thêm phim mới
	2.	Hệ thống	Hiển thị giao diện thêm phim mới gồm nhiều trường thông tin cần thiết
	3.	Admin	Nhập các thông tin của phim và yêu cầu thêm
	4.	Hệ thống	Kiểm tra các trường bắt buộc nhập
	5.	Hệ thống	Kiểm tra các trường cần đảm bảo duy nhất
	6.	Hệ thống	Thêm phim vào database và thông báo thêm thành công
Luồng sự kiện thay thế	5a	Hệ thống	Thông báo lỗi: chưa nhập đủ các trường cần thiết nếu Admin nhập thiếu trường
	6a	Hệ thống	Thông báo lỗi: thông báo cụ thể trường nào không duy nhất nếu Admin nhập trùng
Hậu điều kiện	Không		

Bảng Table 2.7 trình bày chi tiết use case CRUD phim dành cho quản trị viên. Use case này cho phép quản trị viên thực hiện các thao tác quản lý cơ bản đối với danh sách phim trong hệ thống, bao gồm việc thêm mới, chỉnh sửa và xóa bỏ các bộ phim. Sau khi nhận được yêu cầu từ quản trị viên, hệ thống sẽ tiến hành xác thực và xử lý dữ liệu, sau đó phản hồi lại kết quả của các thao tác CRUD đã được thực hiện.

2.4.7 Đặc tả use case Gợi ý phim theo nội dung

Bảng 2.8: Đặc tả use case Gợi ý phim theo nội dung

Mã Use case	UC017		Tên Use case	Gợi ý phim theo nội dung
Tác nhân	Guest, User			
Tiền điều kiện	Xem chi tiết một bộ phim			
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động	
	1.	Người dùng	Nhấn vào một bộ phim để xem chi tiết	
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động	
Hậu điều kiện	Không			

Bảng trên (Table 2.8) là đặc tả use case Gợi ý phim theo nội dung từ người dùng. Trong đó, ứng dụng dựa vào tên phim, thể loại phim, diễn viên, đạo diễn, ... để gợi ý phim tương đồng với phim đang xem.

2.5 Yêu cầu phi chức năng

Phần này trình bày các yêu cầu phi chức năng và các yêu cầu kỹ thuật mà hệ thống cần đáp ứng, được xác định như sau:

Hiệu năng: Hệ thống phải đảm bảo đạt được hiệu suất hoạt động cao và có khả năng phản hồi nhanh chóng. Cụ thể, đối với các thao tác CRUD cơ bản, thời gian phản hồi yêu cầu phải nhỏ hơn 3 giây, trong khi đối với các chức năng tìm kiếm và gợi ý phim sử dụng công nghệ AI, thời gian phản hồi có thể lên đến 10 giây. Ngoài ra, việc tối ưu hóa thời gian tải trang là một yêu cầu quan trọng nhằm nâng cao chất lượng trải nghiệm của người dùng.

Tính dễ sử dụng: Giao diện người dùng cần được xây dựng với nguyên tắc đơn

giản, trực quan và dễ hiểu, tạo điều kiện thuận lợi cho người dùng trong việc tìm kiếm thông tin và thực hiện các tương tác với hệ thống một cách tự nhiên và thuận tiện nhất.

Tính dễ bảo trì: Mã nguồn của ứng dụng cần được tổ chức theo cấu trúc rõ ràng, logic và dễ dàng bảo trì, tạo điều kiện thuận lợi cho việc thực hiện các công việc nâng cấp hệ thống, khắc phục lỗi và mở rộng các tính năng mới trong tương lai một cách hiệu quả.

Cơ sở dữ liệu: Hệ thống cần được tích hợp với một hệ quản trị cơ sở dữ liệu phù hợp để thực hiện việc lưu trữ và quản lý dữ liệu về phim cũng như thông tin người dùng một cách hiệu quả và linh hoạt. Bên cạnh đó, hệ thống cần đảm bảo khả năng thực hiện các truy vấn dữ liệu với tốc độ cao để đáp ứng yêu cầu về hiệu năng.

CHƯƠNG 3. CƠ SỞ LÝ THUYẾT CHO GỢI Ý PHIM

Chương này được dành để giới thiệu về các nền tảng lý thuyết cũng như các thuật toán chính được sử dụng trong việc phát triển hệ thống gợi ý phim, với trọng tâm là phương pháp Content-Based Filtering. Đây được xem là một trong các kỹ thuật được sử dụng rộng rãi và đạt được hiệu quả đáng kể trong lĩnh vực đề xuất nội dung cá nhân hóa cho người dùng, dựa trên nguồn dữ liệu hiện có. Chương sẽ đi sâu vào việc giải thích các khái niệm nền tảng, các nguyên lý lý thuyết, các thuật toán cụ thể được triển khai, cùng với các phương pháp nghiên cứu có liên quan đến kỹ thuật gợi ý này. Mỗi nội dung sẽ giải thích rõ ràng về cách thức ứng dụng các thuật toán nhằm thỏa mãn các yêu cầu và khắc phục các vấn đề đã được nêu ra ở Chương 2. Nhờ vậy, người đọc có thể hiểu rõ về cơ chế hoạt động cũng như các lợi ích mà phương pháp gợi ý này đem lại cho hệ thống.

3.1 Content-Based Filtering

3.1.1 Khái niệm

Content-Based Filtering là một phương pháp gợi ý được triển khai trong các hệ thống đề xuất, hoạt động bằng cách khai thác và phân tích các đặc điểm cũng như các thuộc tính của các đối tượng cần được đề xuất. Điểm khác biệt so với các phương pháp gợi ý dựa trên sở thích và hành vi của cộng đồng người dùng, CBF chủ yếu dựa vào các thuộc tính bên trong và các đặc trưng độc đáo của mỗi sản phẩm để hình thành các gợi ý có tính phù hợp cao. Với phương pháp tiếp cận này, hệ thống có thể xác định và đề xuất các sản phẩm mang tính tương đồng cao dựa trên các sản phẩm mà người dùng đã bày tỏ sự quan tâm hoặc đưa ra đánh giá tốt. Ví dụ cụ thể, nếu một người dùng bày tỏ sự yêu thích đối với một bộ phim thuộc thể loại hành động và có sự tham gia của một diễn viên cụ thể, hệ thống sẽ tự động tìm kiếm và đề xuất các bộ phim khác có cùng thể loại và có sự góp mặt của diễn viên đó.

3.1.2 Các thuật toán

Các thuật toán chính được sử dụng trong Content-Based Filtering bao gồm:

TF-IDF (Term Frequency-Inverse Document Frequency): Thuật toán này được áp dụng nhằm xác định mức độ quan trọng của các từ khóa trong một văn bản. Việc tính toán dựa trên sự kết hợp giữa tần suất xuất hiện của từ trong văn bản (Term Frequency) và nghịch đảo của tần suất xuất hiện của từ đó trong toàn bộ tập văn bản (Inverse Document Frequency), từ đó xác định được trọng số phù hợp cho mỗi từ khóa.

Cosine Similarity: Đây là một phương pháp được sử dụng để đo lường mức độ tương đồng giữa hai vector trong không gian vector. Việc tính toán độ tương tự cosine được thực hiện thông qua việc xác định giá trị cosin của góc được tạo bởi hai vector đó. Kỹ thuật này thường được vận dụng để đánh giá mức độ tương đồng giữa hồ sơ người dùng và hồ sơ của các sản phẩm trong hệ thống.

Việc áp dụng Content-Based Filtering không chỉ giúp tối ưu hóa hiệu suất làm việc của hệ thống gợi ý phim mà còn đảm bảo rằng mỗi người dùng đều nhận được các đề xuất được cá nhân hóa phù hợp với sở thích và nhu cầu riêng của từng cá nhân.

Chương này đã trình bày một cách toàn diện và chi tiết về phương pháp gợi ý phim Content-Based Filtering. Thông qua việc trình bày các kiến thức cơ bản, nền tảng lý thuyết, các thuật toán cụ thể và các phương pháp nghiên cứu liên quan, chúng ta đã có được sự hiểu biết sâu sắc về nguyên lý hoạt động, những ưu điểm cũng như những hạn chế của phương pháp này.

Content-Based Filtering hoạt động dựa trên cơ sở phân tích các đặc trưng và thuộc tính của nội dung phim để từ đó đề xuất các bộ phim có tính chất tương đồng mà người dùng có khả năng quan tâm. Phương pháp này cho phép ứng dụng tạo ra các gợi ý có độ chính xác cao dựa trên các thông tin cụ thể về phim và các sở thích mang tính cá nhân của từng người dùng.

Việc hiểu rõ và áp dụng một cách chính xác các kỹ thuật của phương pháp này sẽ tạo điều kiện để cải thiện đáng kể hiệu suất vận hành của hệ thống gợi ý phim, qua đó tạo ra những trải nghiệm người dùng tốt hơn. Những kiến thức và phương pháp đã được trình bày trong chương này sẽ là cơ sở nền tảng để có thể tiếp tục phát triển và cải thiện hệ thống gợi ý phim trong các chương tiếp theo của báo cáo.

CHƯƠNG 4. CÔNG NGHỆ SỬ DỤNG

4.1 Công nghệ sử dụng phía frontend

Chương này được dành để giới thiệu về các công nghệ và các nền tảng kỹ thuật chính đã được sử dụng trong việc phát triển hệ thống gợi ý phim có tích hợp genAI. Những công nghệ này đóng một vai trò quan trọng và không thể thiếu trong toàn bộ quá trình xây dựng, triển khai và cải thiện hiệu suất của hệ thống, nhằm đảm bảo rằng giải pháp được đề xuất có tính khả thi và đạt được hiệu quả như mong đợi.

4.1.1 React

React là một thư viện JavaScript mã nguồn mở được phát triển bởi Meta, được thiết kế để xây dựng giao diện người dùng dựa trên mô hình component. React vận dụng cơ chế Virtual DOM nhằm tối ưu hóa quá trình cập nhật giao diện, tạo điều kiện để phát triển các ứng dụng web có tính động và tương tác cao.

Lý do tôi lựa chọn React là do nó cho phép phân tách giao diện thành các thành phần độc lập, tạo điều kiện thuận lợi cho việc quản lý, bảo trì và tái sử dụng mã nguồn. Bên cạnh đó, cơ chế ràng buộc dữ liệu một chiều (one-way data binding) cùng với hệ sinh thái đa dạng của React góp phần làm cho quá trình phát triển các tính năng trở nên linh hoạt và đạt hiệu quả cao hơn. Nhờ vào cơ chế Virtual DOM, React đảm bảo ứng dụng vận hành mượt mà ngay cả trong trường hợp phải xử lý khối lượng dữ liệu phim ảnh lớn và các tương tác phức tạp từ phía người dùng.

Trong dự án này, React được áp dụng để phát triển toàn bộ phần giao diện frontend, bao gồm các chức năng như tìm kiếm phim, lọc phim, hiển thị danh sách phim, trang thông tin chi tiết phim, và các chức năng tương tác khác. Các component React được tổ chức theo mô hình module, góp phần làm cho mã nguồn trở nên dễ đọc và dễ bảo trì hơn.

4.1.2 Tailwind CSS

Tailwind CSS là một framework CSS được xây dựng theo triết lý utility-first, cung cấp các lớp tiện ích cấp thấp để tạo giao diện trực tiếp trong mã nguồn mà không cần phải rời khỏi các tệp HTML hoặc JSX. Khác biệt so với các framework CSS truyền thống, Tailwind CSS không áp đặt các thành phần giao diện cố sẵn, tạo điều kiện để người phát triển có thể kiểm soát thiết kế một cách tự do và linh hoạt.

Lý do tôi lựa chọn Tailwind CSS là do nó có khả năng tối ưu hóa kích thước tệp CSS cuối cùng thông qua cơ chế loại bỏ các đoạn mã không được sử dụng, đồng thời hỗ trợ việc thiết kế giao diện đáp ứng một cách trực quan và nhanh chóng. Framework này cho phép tôi có toàn quyền kiểm soát thiết kế và tùy chỉnh giao

diện hệ thống mà không bị giới hạn bởi các thành phần có sẵn.

Trong dự án này, Tailwind CSS được áp dụng để xây dựng toàn bộ phần styling cho giao diện, bao gồm các component như nút bấm, form nhập liệu, card hiển thị phim, và bộ cục responsive. Tailwind CSS đảm bảo giao diện có tính hiện đại, tinh tế và tương thích tốt trên đa dạng các loại thiết bị, từ máy tính để bàn cho đến các thiết bị di động.

4.2 Công nghệ sử dụng phía backend

4.2.1 Django

Django là một framework phát triển web được xây dựng trên nền tảng ngôn ngữ lập trình Python, nổi tiếng với khả năng tạo ra các ứng dụng web trong thời gian ngắn và đạt được hiệu suất làm việc tốt. Framework này được thiết kế với kiến trúc MVT (Model-View-Template) có tính nhất quán và dễ hiểu, giúp cho việc sắp xếp và quản lý mã nguồn trở nên có hệ thống và hợp lý hơn. Django có khả năng kết nối với nhiều loại hệ quản trị cơ sở dữ liệu khác nhau và được tích hợp sẵn các công cụ mạnh mẽ như ORM cùng với hệ thống quản trị admin được tự động hóa.

Việc tôi quyết định sử dụng Django xuất phát từ khả năng của nó trong việc tương tác với cơ sở dữ liệu thông qua ORM mà không đòi hỏi phải viết các câu lệnh SQL một cách thủ công, nhờ đó có thể rút ngắn thời gian phát triển và hạn chế các sai sót trong quá trình lập trình. Ngoài ra, Django còn được tích hợp sẵn các cơ chế bảo mật nhằm ngăn chặn các hình thức tấn công phổ biến như CSRF, XSS và SQL Injection, từ đó nâng cao mức độ an toàn của hệ thống. Framework này còn có một cộng đồng người dùng và nhà phát triển đông đảo cùng với hệ thống tài liệu phong phú, tạo thuận lợi cho việc phát triển dự án và xử lý các vấn đề kỹ thuật khi phát sinh.

Trong dự án này, Django được áp dụng để phát triển toàn bộ phần backend, bao gồm các API endpoints để xử lý các yêu cầu từ frontend, quản lý dữ liệu về phim và người dùng thông qua ORM, xử lý logic nghiệp vụ cho các chức năng như tìm kiếm, gợi ý phim, và quản lý đánh giá. Django REST Framework được sử dụng kết hợp để xây dựng RESTful API, tạo điều kiện cho việc giao tiếp giữa frontend React và backend Django.

4.2.2 MySQL

MySQL là một hệ thống quản lý cơ sở dữ liệu quan hệ có mã nguồn mở, được sử dụng phổ biến trong nhiều ứng dụng web do đặc tính ổn định và khả năng xử lý hiệu quả. MySQL được tích hợp các chức năng mạnh mẽ bao gồm quản lý các dữ liệu có độ phức tạp cao, thực thi các truy vấn với hiệu suất cao, đảm bảo mức

độ bảo mật tốt, và có khả năng mở rộng dễ dàng. Hệ thống này hoạt động dựa trên ngôn ngữ SQL tiêu chuẩn và có khả năng hỗ trợ nhiều loại dữ liệu khác nhau.

Lý do tôi lựa chọn MySQL là do nó phù hợp với các yêu cầu về lưu trữ và truy xuất dữ liệu phim ảnh của ứng dụng, đảm bảo dữ liệu được tổ chức một cách có cấu trúc và dễ dàng quản lý. MySQL có khả năng sao lưu và phục hồi dữ liệu hiệu quả, góp phần bảo vệ các dữ liệu quan trọng của ứng dụng. Bên cạnh đó, MySQL có khả năng tích hợp tốt với Django thông qua ORM, tạo điều kiện thuận lợi cho quá trình phát triển.

Trong dự án này, MySQL được áp dụng để lưu trữ toàn bộ dữ liệu của hệ thống, bao gồm các thông tin về phim (tên phim, mô tả, thể loại, năm sản xuất), thông tin về đạo diễn, diễn viên, giải thưởng, dữ liệu người dùng, đánh giá phim, và lịch sử hoạt động của người dùng. Cơ sở dữ liệu này đảm bảo dữ liệu được lưu trữ một cách an toàn và có thể truy xuất hiệu quả cho các chức năng như tìm kiếm, gợi ý phim, và phân tích hành vi người dùng.

4.3 Công nghệ sử dụng trong việc lưu trữ code

4.3.1 Git

Git là một công cụ quản lý phiên bản hoạt động theo mô hình phân tán, có khả năng ghi nhận và theo dõi mọi thay đổi trong mã nguồn trong toàn bộ vòng đời phát triển dự án. Git lưu trữ toàn bộ lịch sử các thay đổi dưới hình thức các snapshot, cho phép người dùng có thể khôi phục lại bất kỳ phiên bản nào đã từng tồn tại trước đó. Công cụ này được thiết kế dựa trên nguyên lý phân tán, trong đó mỗi bản sao của repository đều lưu trữ đầy đủ toàn bộ lịch sử phát triển của dự án.

Quyết định của tôi trong việc chọn Git dựa trên khả năng quản lý mã nguồn hiệu quả của nó, giúp cho việc khôi phục các phiên bản cũ trở nên đơn giản khi xảy ra lỗi hoặc khi cần đổi chiều các thay đổi. Git được tích hợp các chức năng quan trọng như tạo nhánh, gộp mã và ghi chép lịch sử, đóng góp vào việc điều phối quá trình phát triển và giải quyết các mâu thuẫn trong mã nguồn một cách có hiệu quả. Hơn nữa, Git tạo điều kiện cho việc hợp tác nhóm được diễn ra suôn sẻ, cho phép mỗi thành viên có thể làm việc một cách độc lập trên các nhánh riêng của mình.

Trong dự án này, Git được áp dụng để quản lý toàn bộ mã nguồn của dự án, bao gồm mã frontend React, backend Django, các script crawl dữ liệu, và các tệp cấu hình. Tôi sử dụng các nhánh khác nhau để phát triển các tính năng riêng biệt, sau đó hợp nhất vào nhánh chính sau khi hoàn thành và kiểm thử. Git góp phần theo dõi mọi thay đổi trong quá trình phát triển và đảm bảo mã nguồn luôn được sao lưu một cách an toàn.

4.3.2 GitHub

GitHub là một dịch vụ lưu trữ mã nguồn được phát triển trên nền tảng Git, mang đến một môi trường trực tuyến để trao đổi, lưu giữ và điều hành mã nguồn. GitHub cung cấp một giao diện web dễ sử dụng để quản lý các kho chứa mã, giám sát các vấn đề kỹ thuật, xử lý các yêu cầu tính năng mới và thực hiện việc xem xét lại mã nguồn. Nền tảng này còn được bổ sung nhiều công cụ hỗ trợ như hệ thống quản lý dự án, wiki, và các tính năng tích hợp liên tục CI/CD.

Lý do tôi lựa chọn GitHub là do nó tạo điều kiện để dễ dàng chia sẻ mã nguồn và phối hợp phát triển sản phẩm một cách hiệu quả, đặc biệt trong trường hợp làm việc với nhóm hoặc cần sao lưu mã nguồn trên đám mây. GitHub cung cấp các tính năng review mã và quản lý dự án, góp phần làm cho quy trình phát triển trở nên chuyên nghiệp và linh hoạt hơn. Bên cạnh đó, GitHub hỗ trợ tích hợp với đa dạng các dịch vụ và công cụ khác, tạo điều kiện để tự động hóa các quy trình phát triển và triển khai sản phẩm thông qua GitHub Actions.

Trong dự án này, GitHub được áp dụng để lưu trữ toàn bộ mã nguồn của dự án trên đám mây, đảm bảo mã nguồn luôn được sao lưu và có thể truy cập từ bất kỳ đâu. Tôi sử dụng GitHub Issues để theo dõi các công việc cần thực hiện và các lỗi cần được khắc phục. Ngoài ra, GitHub Actions được áp dụng để tự động hóa quy trình cập nhật dữ liệu phim hàng ngày, thực thi các script crawl dữ liệu từ IMDb và BoxOfficeMojo.

4.4 Công nghệ sử dụng trong việc crawl dữ liệu

Trong quá trình phát triển hệ thống gợi ý phim, việc thu thập và duy trì cập nhật dữ liệu từ các nguồn có độ tin cậy cao là một thành phần thiết yếu để đảm bảo độ chính xác và tính cập nhật của các thông tin về phim ảnh. Nhằm thực hiện nhiệm vụ này, tôi đã sử dụng các công cụ có khả năng mạnh mẽ như Selenium và Requests, đồng thời kết hợp với việc tự động hóa toàn bộ quy trình cập nhật dữ liệu bằng cách sử dụng GitHub Actions.

4.4.1 Selenium và Requests

Selenium là một công cụ tự động hóa trình duyệt web có khả năng mạnh mẽ, cho phép điều khiển trình duyệt và thực hiện các tương tác với các trang web có nội dung động giống như cách một người dùng thực tế sẽ thao tác. Selenium có thể thực hiện nhiều loại hành động khác nhau bao gồm click chuột, nhập dữ liệu, cuộn trang, và chờ đợi các thành phần động được tải hoàn tất. Requests là một thư viện HTTP trong Python có thiết kế đơn giản nhưng sở hữu sức mạnh đáng kể, hỗ trợ việc gửi các yêu cầu HTTP và lấy dữ liệu từ các trang web. Requests được tích hợp nhiều chức năng hữu ích như điều hướng qua các trang web, gửi dữ liệu dưới định

dạng form và quản lý cookies.

Việc tôi quyết định sử dụng Selenium và Requests bắt nguồn từ khả năng của chúng trong việc tự động thu thập dữ liệu từ các trang web có độ tin cậy cao như IMDb và BoxOfficeMojo một cách hiệu quả. Selenium được sử dụng để xử lý các trang web có nội dung động đòi hỏi người dùng phải thực hiện các thao tác như cuộn trang hoặc nhấn nút để tải thêm nội dung, còn Requests được áp dụng cho các trang web tĩnh để lấy dữ liệu với tốc độ cao hơn. Việc kết hợp sử dụng hai công cụ này đã giúp tôi có thể thu thập được một lượng dữ liệu phong phú và đáng tin cậy về phim ảnh để phục vụ cho hệ thống.

Trong dự án này, Selenium được áp dụng để crawl dữ liệu từ IMDb, bao gồm các thông tin về phim (tên phim, mô tả, thể loại, năm sản xuất, đánh giá), thông tin về đạo diễn, diễn viên, giải thưởng, và hình ảnh. Requests được sử dụng để lấy dữ liệu tĩnh từ BoxOfficeMojo về doanh thu phòng vé của các bộ phim. Các script crawl này được thực thi tự động hàng ngày thông qua GitHub Actions để đảm bảo dữ liệu luôn được cập nhật mới nhất.

4.4.2 Công nghệ sử dụng để update dữ liệu

Việc duy trì và đảm bảo cập nhật dữ liệu phim ảnh một cách thường xuyên là một điều kiện quan trọng để hệ thống gợi ý phim có thể cung cấp các thông tin mới nhất và đáng tin cậy. Nhằm thực hiện tự động hóa quy trình này, tôi đã sử dụng GitHub Actions. GitHub Actions là một công cụ CI/CD được tích hợp sẵn trong nền tảng GitHub, có khả năng cho phép người dùng tạo ra các workflow tự động sẽ được kích hoạt khi xảy ra các sự kiện nhất định.

Bằng cách sử dụng GitHub Actions, tôi đã tạo ra các workflow có khả năng tự động cập nhật dữ liệu phim từ IMDb theo chu kỳ hàng ngày. Các workflow này thực hiện việc chạy các script Selenium và Requests để thu thập các dữ liệu mới, sau đó tiến hành cập nhật vào cơ sở dữ liệu của hệ thống. Quá trình tự động hóa này đảm bảo rằng dữ liệu luôn được cập nhật đều đặn theo đúng lịch trình đã định mà không cần có sự can thiệp từ con người.

Cụ thể hơn, mỗi ngày GitHub Actions sẽ tự động khởi chạy các script thu thập dữ liệu để kiểm tra và áp dụng các thay đổi mới nhất từ IMDb. Quá trình này bao gồm việc cập nhật các trường thông tin như doanh thu, trailer, hình ảnh và các thông tin liên quan khác. Workflow GitHub Actions mà tôi đã cấu hình bao gồm các bước sau đây:

Sử dụng thư viện Requests để gửi các yêu cầu HTTP và lấy các dữ liệu tĩnh từ các trang web có liên quan.

Thực hiện việc đọc, xử lý và làm sạch các dữ liệu đã được thu thập, đảm bảo rằng dữ liệu có đúng định dạng và cấu trúc yêu cầu.

Tiến hành cập nhật vào cơ sở dữ liệu MySQL với các thông tin mới nhất đã được xử lý.

Việc sử dụng GitHub Actions không chỉ giúp tự động hóa quy trình cập nhật dữ liệu mà còn đảm bảo tính đồng nhất và độ chính xác của các thông tin phim ảnh trong hệ thống. Điều này giúp cho hệ thống gợi ý phim có thể liên tục cung cấp các gợi ý và thông tin được cập nhật mới nhất, đáp ứng một cách tối ưu các nhu cầu của người dùng.

Chương này đã giới thiệu về các công nghệ và các nền tảng kỹ thuật chính đã được sử dụng trong quá trình phát triển hệ thống gợi ý phim. Sự lựa chọn Django và MySQL cho phần backend, React và Tailwind CSS cho phần frontend, Git và GitHub cho việc quản lý mã nguồn, cùng với các phương pháp thu thập dữ liệu từ IMDb và BoxOfficeMojo đã đóng góp vào việc xây dựng một hệ thống gợi ý phim có hiệu quả, ổn định và dễ sử dụng. Những công nghệ này không chỉ đáp ứng được các yêu cầu của hệ thống mà còn tạo ra những điều kiện thuận lợi cho việc mở rộng và cải tiến trong tương lai.

CHƯƠNG 5. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

5.1 Thiết kế kiến trúc

5.1.1 Lựa chọn kiến trúc phần mềm

Trong sản phẩm này, tôi lựa chọn kiến trúc phần mềm MVT trong Django [7].

Giải thích sơ bộ về MVT:

Kiến trúc MVT của Django tương tự như kiến trúc MVC (Model-View-Controller), nhưng có sự khác biệt về cách hoạt động.

- **Model (M):** Định nghĩa cấu trúc dữ liệu của ứng dụng và quản lý truy cập vào cơ sở dữ liệu.
- **View (V):** Xử lý logic xử lý nghiệp vụ và hiển thị dữ liệu cho người dùng.
- **Template (T):** Định nghĩa giao diện người dùng bằng cách sử dụng HTML cùng với các biến Python.

Mô tả kiến trúc cụ thể cho ứng dụng:

• Model (M):

Trong ứng dụng của em, các model như MovieInformation, Awards, Director, Cast, Genres, và các model liên quan khác sẽ đại diện cho các đối tượng dữ liệu chính về phim, đạo diễn, diễn viên, thể loại.

Mỗi model sẽ có các trường (fields) tương ứng và quan hệ (relationships) với các model khác để lưu trữ và quản lý thông tin một cách logic và cấu trúc hóa.

• View (V):

Các view như FilmListView, DirectorListView, GenreListView, ReviewView, và các view khác trong danh sách của tôi sẽ đảm nhiệm xử lý các yêu cầu từ người dùng.

View sẽ truy xuất dữ liệu từ model thông qua các queryset, xử lý logic nghiệp vụ và chuẩn bị dữ liệu để truyền cho template.

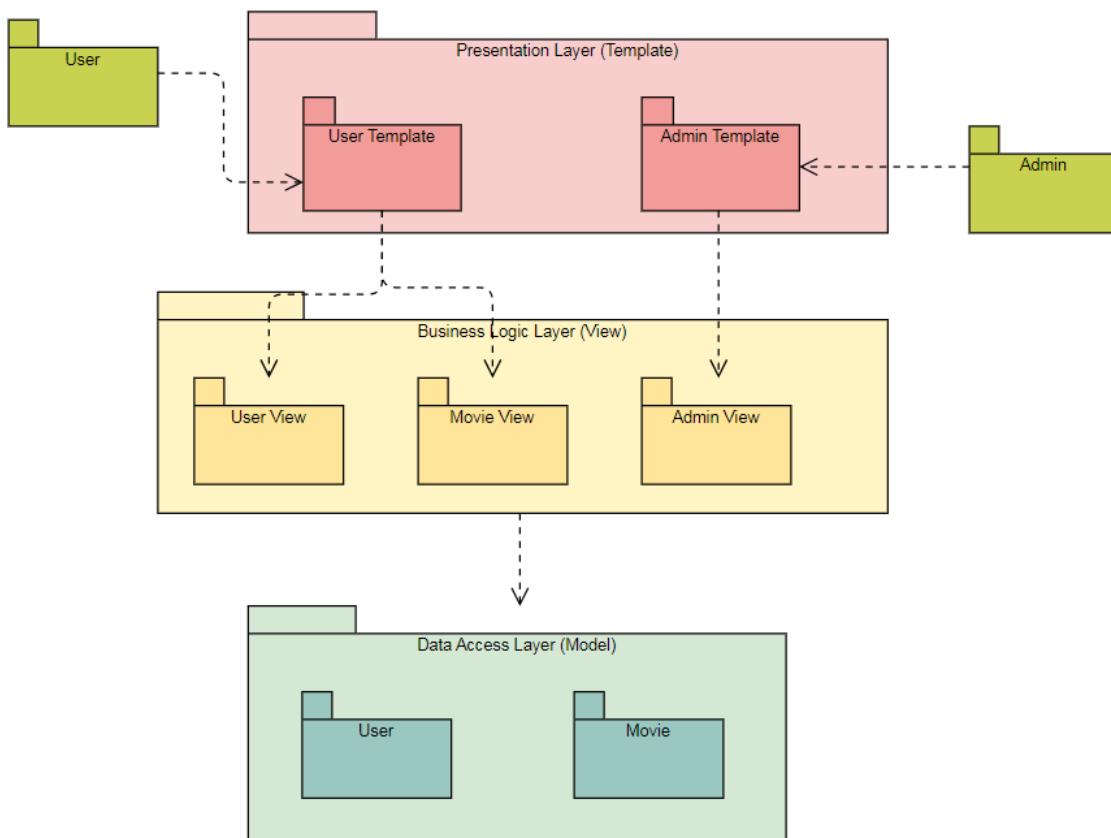
• Template (T):

Sử dụng các template HTML của Django kết hợp với template tags hiển thị dữ liệu cho người dùng.

Template sẽ nhận dữ liệu từ view và render ra các thành phần giao diện như danh sách phim, thông tin chi tiết, biểu đồ.

5.1.2 Thiết kế tổng quan

Các gói và sự phụ thuộc giữa các gói:



Hình 5.1: Biểu đồ phụ thuộc gói

1. Data Access Layer (Model)

- Mục đích: Tầng này chịu trách nhiệm chứa các định nghĩa về dữ liệu và cấu trúc dữ liệu của hệ thống.
- Nhiệm vụ: Thực hiện mô hình hóa các thực thể trong ứng dụng, bao gồm việc định nghĩa các bảng cơ sở dữ liệu và các mối quan hệ giữa các bảng đó.
- Các thành phần: MovieInformation, Awards, Director, Cast, Genres.
- Phụ thuộc: Tầng này không có sự phụ thuộc vào bất kỳ gói nào khác. Đây là tầng nằm ở mức thấp nhất trong kiến trúc hệ thống, nơi các dữ liệu được định nghĩa và quản lý thông qua các model.

2. Business Logic Layer (View)

- Mục đích: Tầng này đảm nhiệm việc xử lý logic nghiệp vụ và quản lý các yêu cầu đến từ phía người dùng.
- Nhiệm vụ: Thực hiện truy xuất dữ liệu từ các model, tiến hành xử lý và

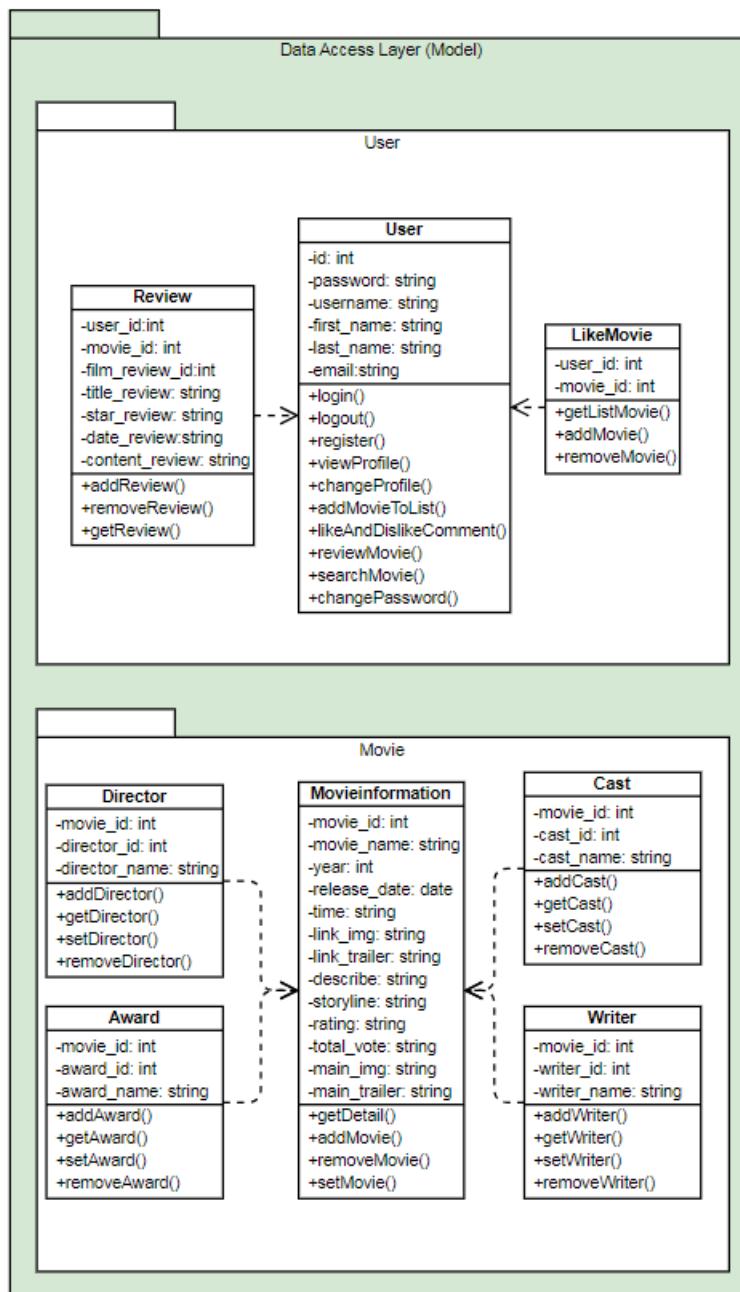
chuẩn bị dữ liệu để có thể hiển thị trong template.

- Các thành phần: FilmListView, DirectorListView, GenreListView, ReviewView.
- Phụ thuộc: Tầng này có sự phụ thuộc vào gói Models. View sẽ thực hiện truy xuất dữ liệu từ model và xử lý logic nghiệp vụ, không có sự phụ thuộc trực tiếp vào gói Templates. Dữ liệu sau khi được chuẩn bị trong view sẽ được chuyển đến template để hiển thị cho người dùng.

3. Presentation Layer (Template)

- Mục đích: Tầng này chịu trách nhiệm định nghĩa và xây dựng giao diện người dùng.
- Nhiệm vụ: Thực hiện hiển thị dữ liệu được truyền từ view dưới các định dạng HTML, CSS và JavaScript. Template sẽ tiếp nhận dữ liệu từ view và thực hiện việc hiển thị dữ liệu đó.
- Các thành phần: Các tệp HTML template được sử dụng cho các trang web khác nhau của ứng dụng, chẳng hạn như trang đăng nhập, đăng ký, trang chủ, trang chi tiết phim.
- Phụ thuộc: Tầng này có sự phụ thuộc vào gói Views. Templates tiếp nhận dữ liệu từ view và thực hiện hiển thị, không có sự phụ thuộc trực tiếp vào gói Models. Tất cả dữ liệu được hiển thị trên giao diện đều được truyền từ view.

5.1.3 Thiết kế chi tiết gói



Hình 5.2: Thiết kế chi tiết gói ở tầng Model

Gói Model:

Ở gói này, có các gói nhỏ hợp thành là User và Movie.

- **Gói User:** Gồm 3 class là Review, User và LikeMovie.

Mô tả: Class User: Đại diện cho thông tin người dùng. Class Review: Chứa thông tin về đánh giá phim của người dùng. Class LikeMovie: Lưu trữ những bộ phim mà người dùng thích.

Phụ thuộc: Class Review và class LikeMovie phụ thuộc vào thông tin của

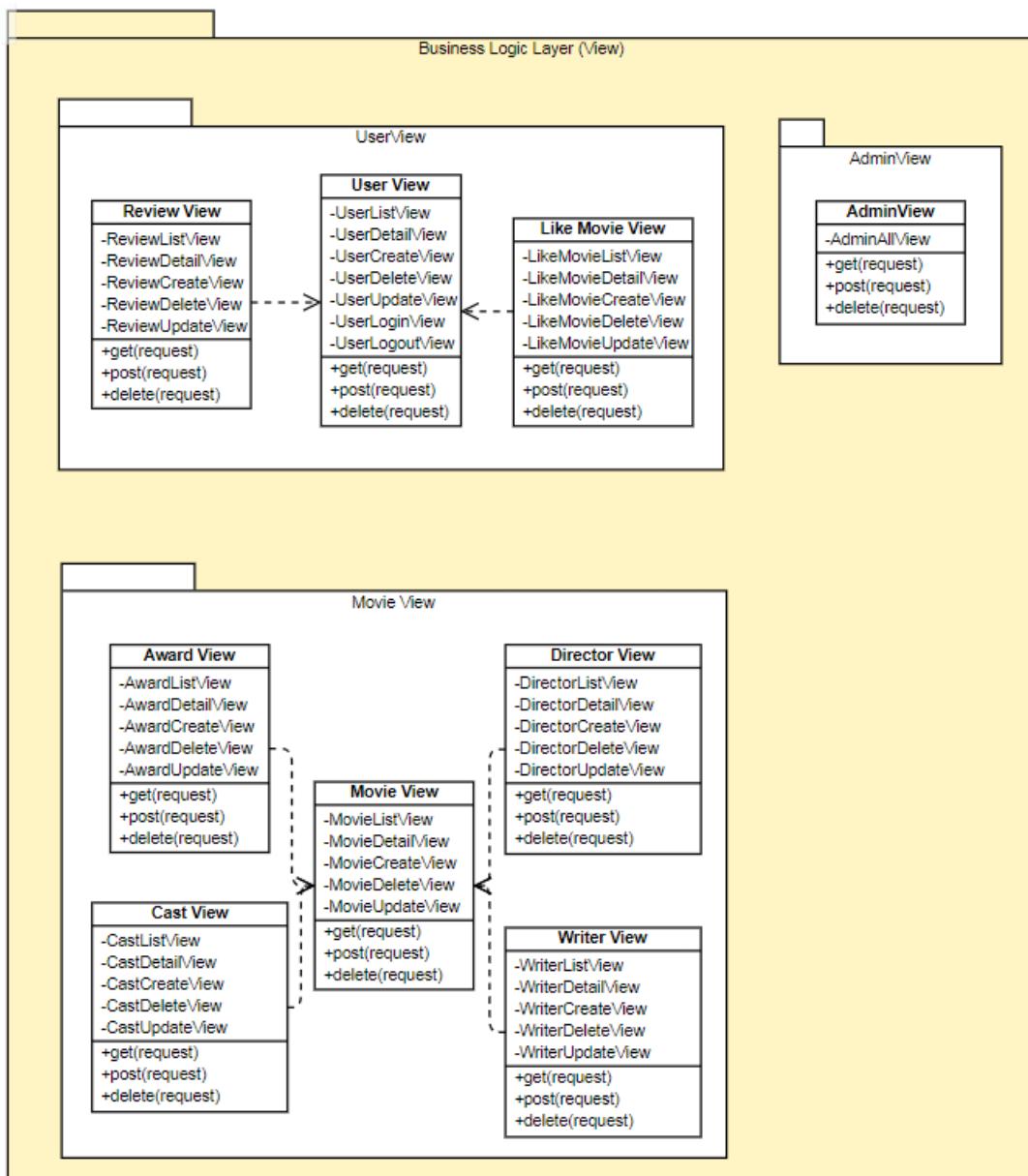
User.

- **Gói Movie:**

Gồm 5 class là Movieinfomation, Director, Award, Cast, Writer.

Mô tả: Class MovieInformation: Lớp chính quản lý thông tin tổng quát về phim. Class Director: Lưu trữ thông tin chi tiết về các đạo diễn của phim. Class Award: Lưu trữ thông tin về các giải thưởng phim đã nhận được. Class Cast: Lưu trữ thông tin về các diễn viên tham gia trong phim. Class Writer: Lưu trữ thông tin về các nhà biên kịch của phim.

Phụ thuộc: 4 class còn lại có quan hệ phụ thuộc với Movieinfomation, lấy Movieinfomation làm gốc, các thông tin còn lại đều là thông tin bổ sung.



Hình 5.3: Thiết kế chi tiết gói ở tầng View

Gói View:

Ở gói này, có các gói nhỏ hợp thành là UserView, MovieView và AdminView.

- Gói UserView:

Gồm 3 class là ReviewView, UserView và LikeMovieView.

Mô tả: Class UserView: Lớp này chịu trách nhiệm hiển thị và xử lý các thông tin liên quan đến người dùng, bao gồm đăng ký, đăng nhập, và thông tin cá nhân. Class ReviewView: Lớp này xử lý và hiển thị các đánh giá phim của người dùng. Nó phụ thuộc vào thông tin người dùng từ UserView. Class Like-MovieView: Lớp này quản lý và hiển thị các bộ phim mà người dùng đã thích. Nó cũng phụ thuộc vào thông tin người dùng từ UserView.

Phụ thuộc: Class ReviewView và class LikeMovieView phụ thuộc vào thông tin của UserView.

- Gói MovieView:

Gồm 5 class là MovieinfomationView, DirectorView, AwardView, CastView, WriterView.

Mô tả: Class MovieInformationView: Lớp chính quản lý, thao tác chính về phim. Class DirectorView: Xử lý dữ liệu về đạo diễn. Class AwardView: Xử lý dữ liệu, các thao tác về giải thưởng. Class CastView: Xử lý dữ liệu, các thao tác tính về diễn viên. Class Writer: Thao tác dữ liệu với biên kịch.

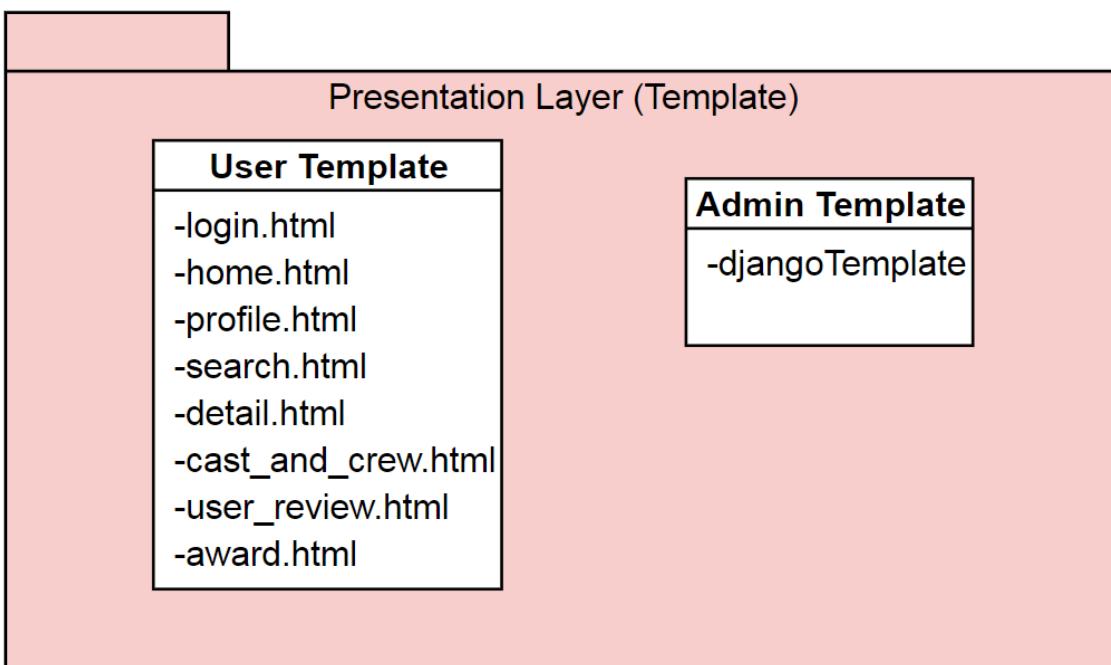
Phụ thuộc: 4 class còn lại có quan hệ phụ thuộc với Movieinfomation, lấy Movieinfomation làm gốc, các thông tin còn lại đều là thông tin bổ sung.

- Gói AdminView:

Gồm 1 class là AdminView.

Mô tả: Class AdminView có thể xử lý toàn bộ yêu cầu bao gồm các yêu cầu giống người dùng và có thể khác, admin quản lý toàn bộ trang web. Có thể thêm, sửa, xóa toàn bộ dữ liệu.

Phụ thuộc: Không phụ thuộc vào class nào.

**Hình 5.4:** Thiết kế chi tiết gói ở tầng Template

Gói Template:

Gói này chứa hai lớp là UserTemplate và AdminTemplate.

Mô tả: Class UserTemplate: Lớp này chịu trách nhiệm quản lý các file template HTML dành cho phần giao diện của người dùng cuối. Những template này gồm có các trang đăng nhập, đăng ký tài khoản, trang profile cá nhân, trang để đánh giá phim, cùng với các trang liệt kê danh sách phim, danh sách diễn viên và danh sách đạo diễn. Class AdminTemplate: Lớp này chịu trách nhiệm quản lý các file template HTML dành cho phần giao diện của quản trị viên. Những template này gồm có các trang để quản lý tài khoản người dùng, quản lý thông tin phim, phê duyệt các đánh giá và những trang khác mà admin sử dụng để điều hành toàn bộ hệ thống.

Phụ thuộc: Hai lớp này không tồn tại mối quan hệ phụ thuộc với nhau. Mỗi lớp hoạt động một cách tự chủ và đáp ứng những nhu cầu khác nhau: một lớp phục vụ cho người dùng thông thường, lớp còn lại phục vụ cho bộ phận quản trị.

5.2 Thiết kế chi tiết

5.2.1 Thiết kế giao diện

Khi tiến hành thiết kế giao diện cho hệ thống, tôi đã chọn lựa các tham số kỹ thuật, các tiêu chuẩn thiết kế và các mẫu template chuẩn để đảm bảo sự đồng nhất và tính tiện dụng. Những màn hình chính của hệ thống được xây dựng với độ phân giải 1920x1080 pixels, tương thích với các loại màn hình có kích thước từ 13 inch

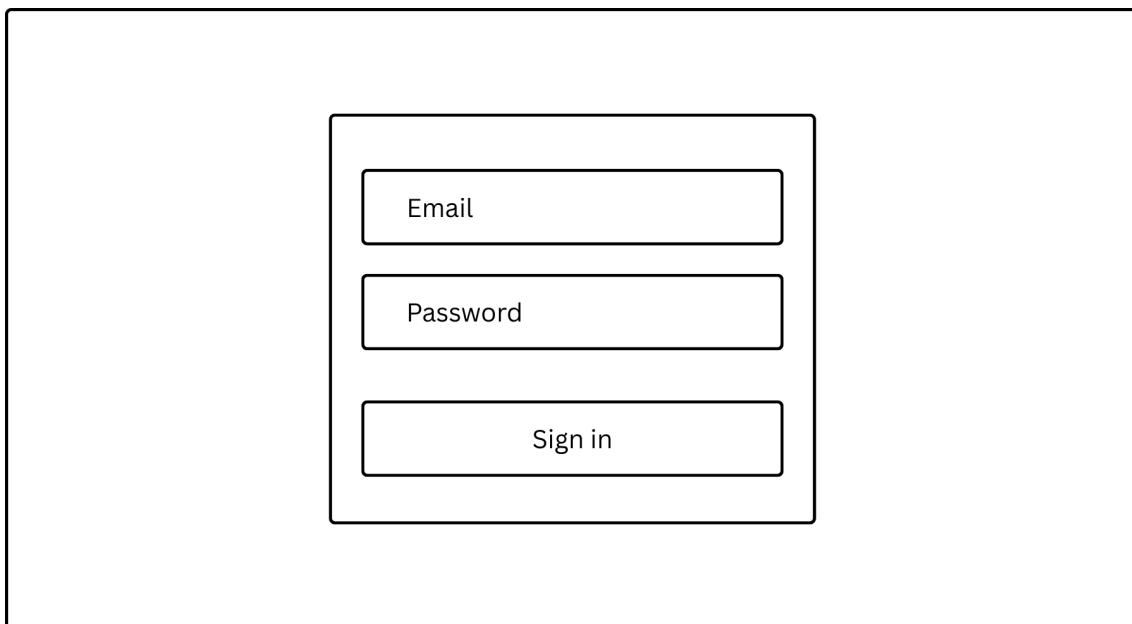
trở lên. Hệ thống có khả năng hiển thị lên đến 16 triệu màu để đảm bảo các thành phần đồ họa và hình ảnh hiển thị với chất lượng cao và chi tiết.

Trong giai đoạn thiết kế, tôi đã áp dụng các nguyên tắc thiết kế cụ thể để xây dựng một giao diện có sự thống nhất. Những nút điều khiển được xây dựng với kích thước tuân theo các tiêu chuẩn phổ biến, giúp người dùng có thể dễ dàng nhận diện và thực hiện các thao tác. Các thành phần điều khiển như menu thả xuống, hộp kiểm và nút radio được xây dựng để có diện mạo thống nhất, nhất quán và dễ phân biệt. Những thông báo phản hồi cho người dùng được đặt ở vị trí dễ nhìn thấy, cụ thể là góc trên bên phải màn hình.

Đối với việc chọn lựa màu sắc, tôi đã chọn một bảng màu cơ bản với hai tông màu chủ đạo là đen và tím, mang lại cảm giác thoải mái và thể hiện tính chuyên nghiệp. Những màu bổ sung được dùng để nhấn mạnh các thành phần quan trọng hoặc để báo hiệu các trạng thái khác nhau như thành công, cảnh báo, lỗi và trạng thái khi di chuột qua với các màu đỏ, vàng và xanh lá.

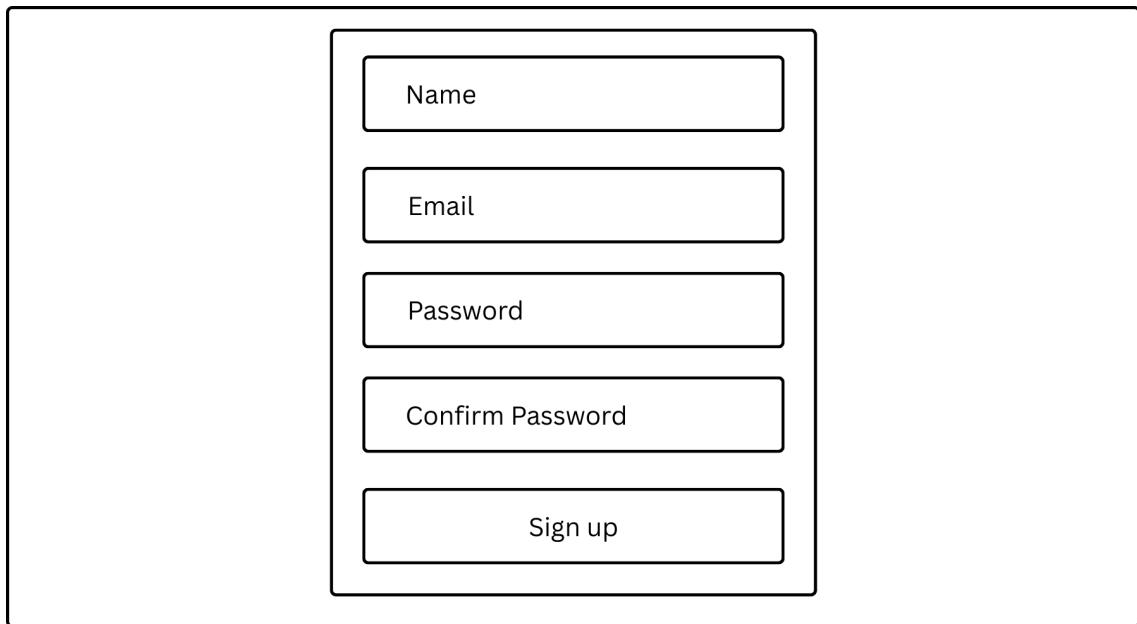
Dưới đây là các hình ảnh minh họa cho thiết kế giao diện của các chức năng quan trọng trong ứng dụng:

Màn hình đăng nhập



Hình 5.5: Màn hình *đăng nhập*

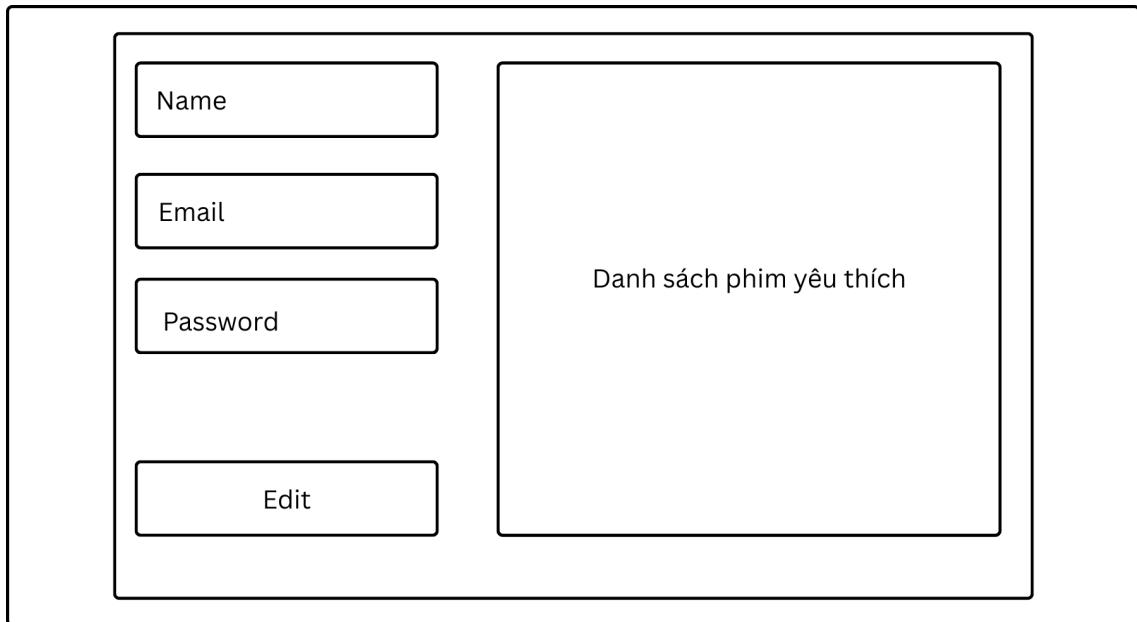
Màn hình đăng ký



The diagram shows a sign-up form within a large rectangular container. On the right side, there is a vertical stack of five input fields, each enclosed in a thin black border. The fields are labeled 'Name', 'Email', 'Password', 'Confirm Password', and 'Sign up' respectively. The 'Sign up' button is positioned at the bottom of the stack.

Hình 5.6: Màn hình *đăng ký*

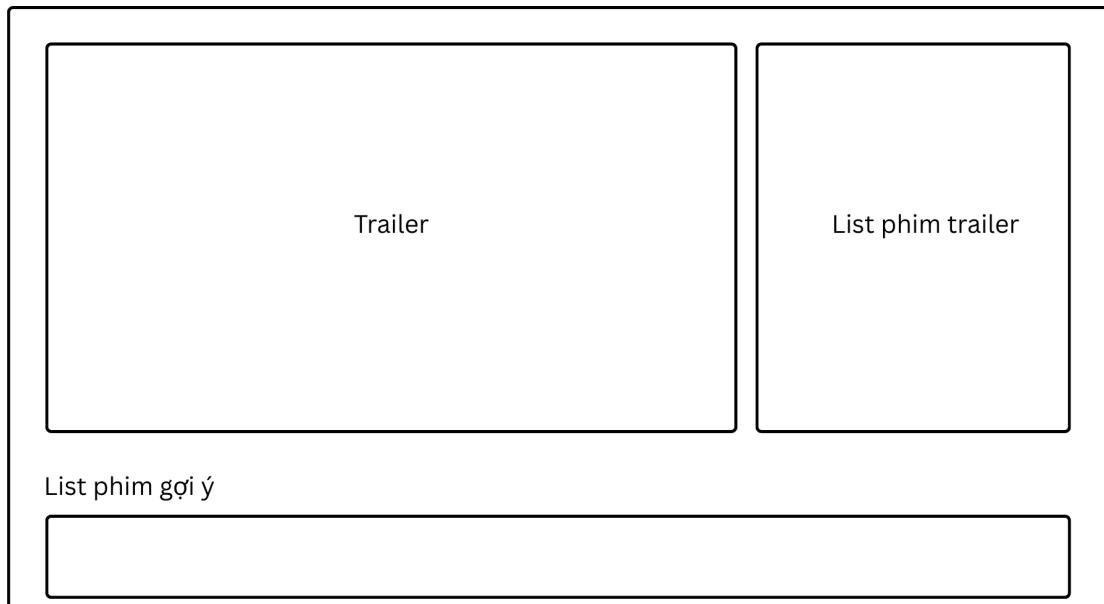
Màn hình profile



The diagram shows a profile management form within a large rectangular container. On the left side, there is a vertical stack of four input fields, each enclosed in a thin black border. The fields are labeled 'Name', 'Email', 'Password', and 'Edit' respectively. To the right of the 'Edit' button is a large, empty rectangular area with the text 'Danh sách phim yêu thích' centered in it.

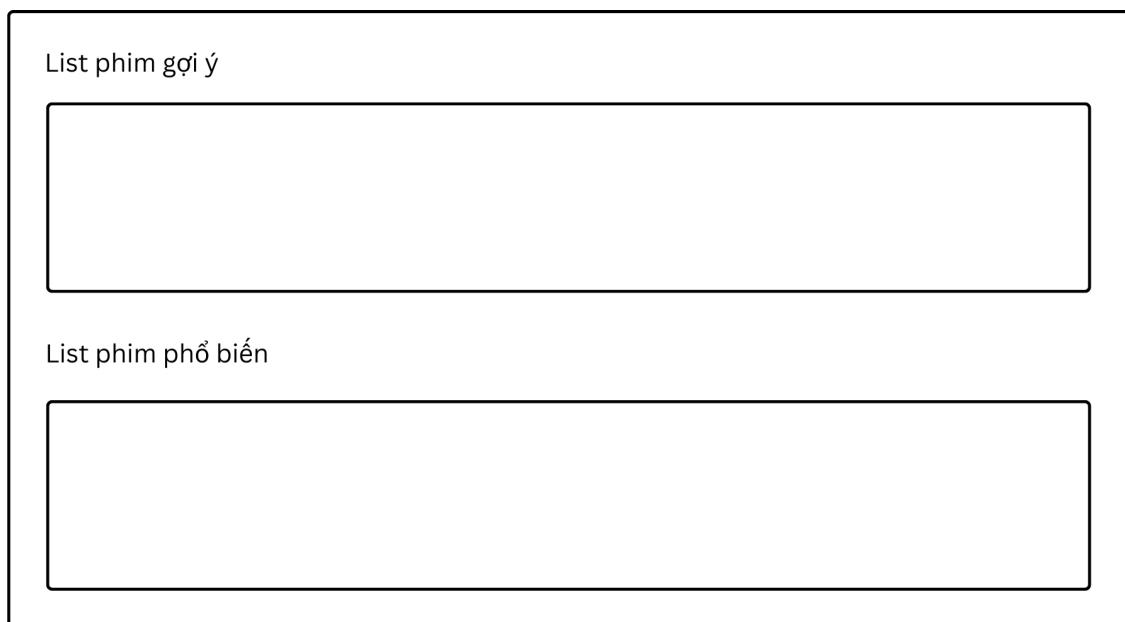
Hình 5.7: Màn hình *profile*

Màn hình chính



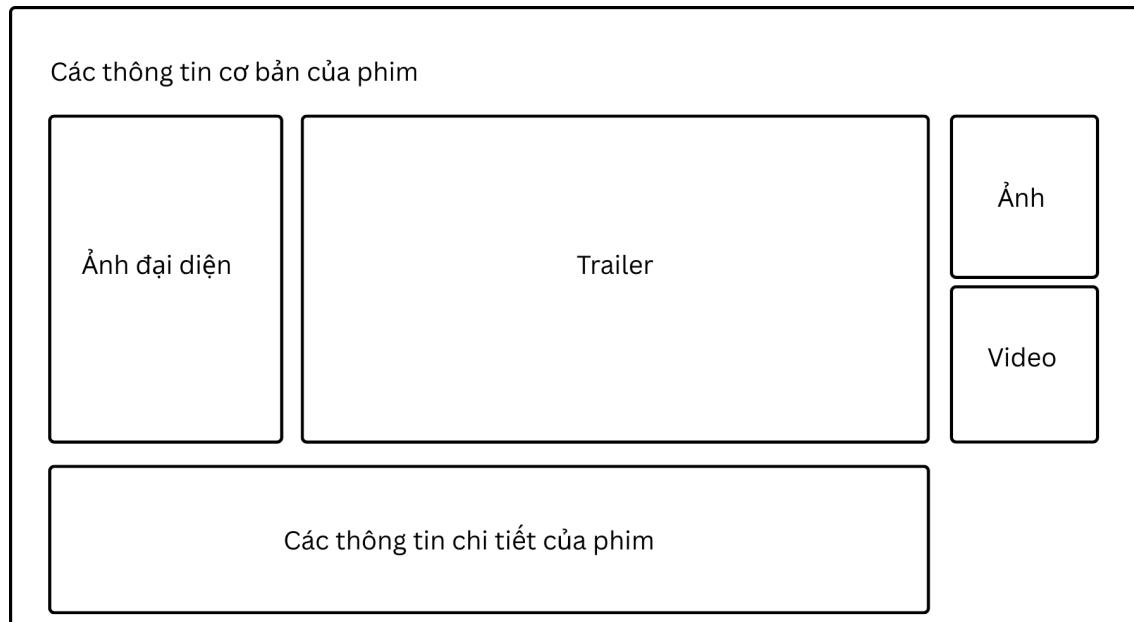
Hình 5.8: Màn hình *chính*

Màn hình gợi ý phim ở trang chủ

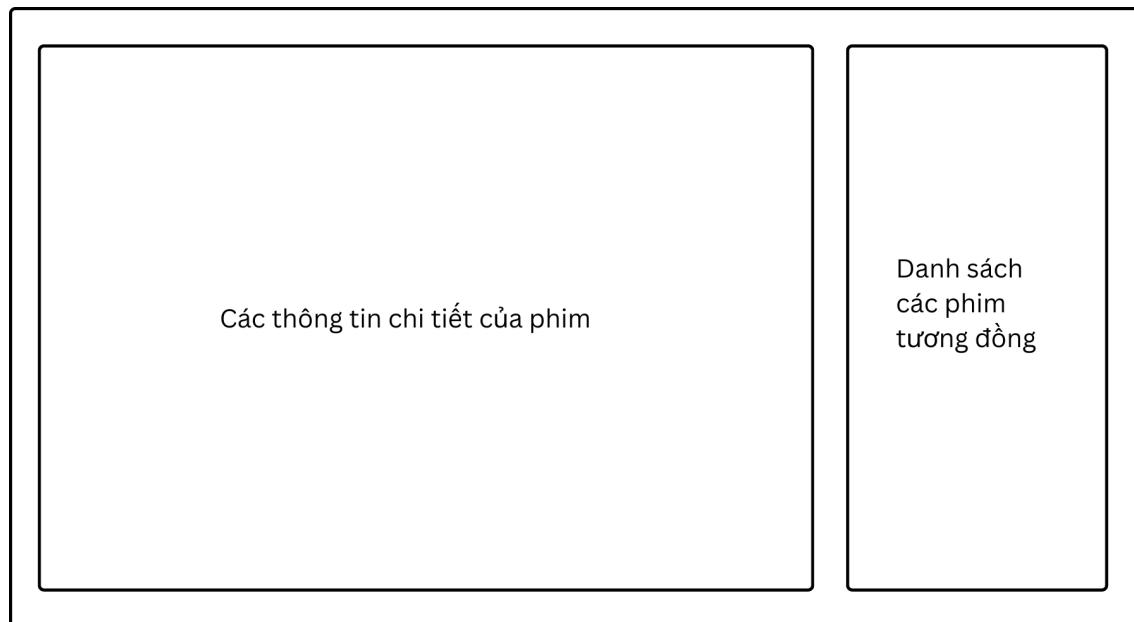


Hình 5.9: Màn hình *gợi ý phim* ở *trang chủ*

Màn hình thông tin chi tiết

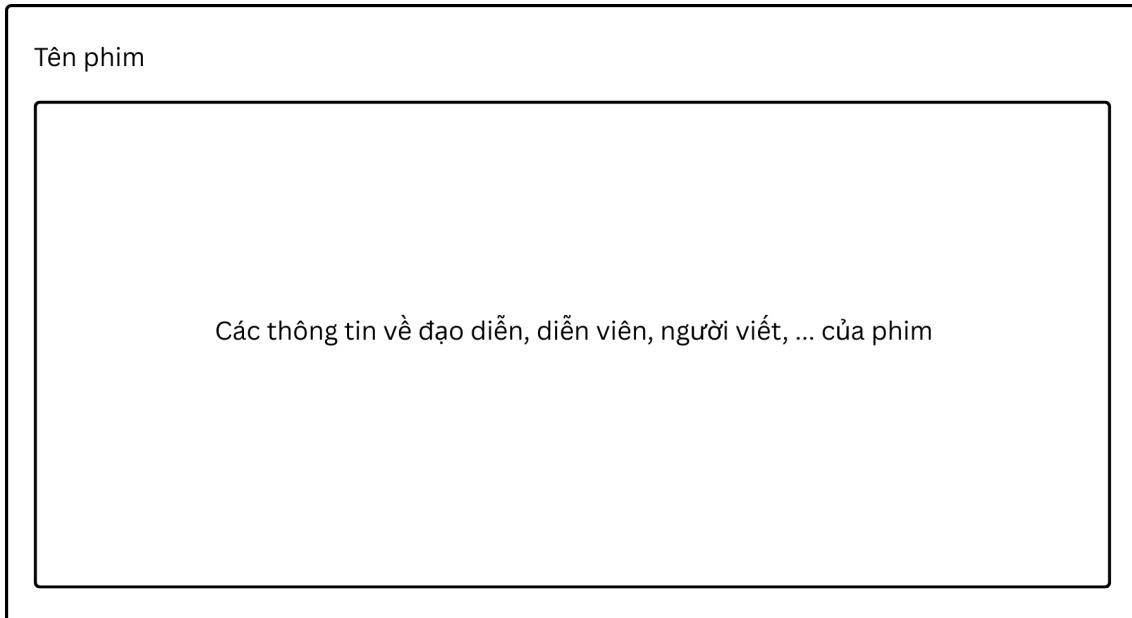


Hình 5.10: Màn hình *thông tin chi tiết 1*



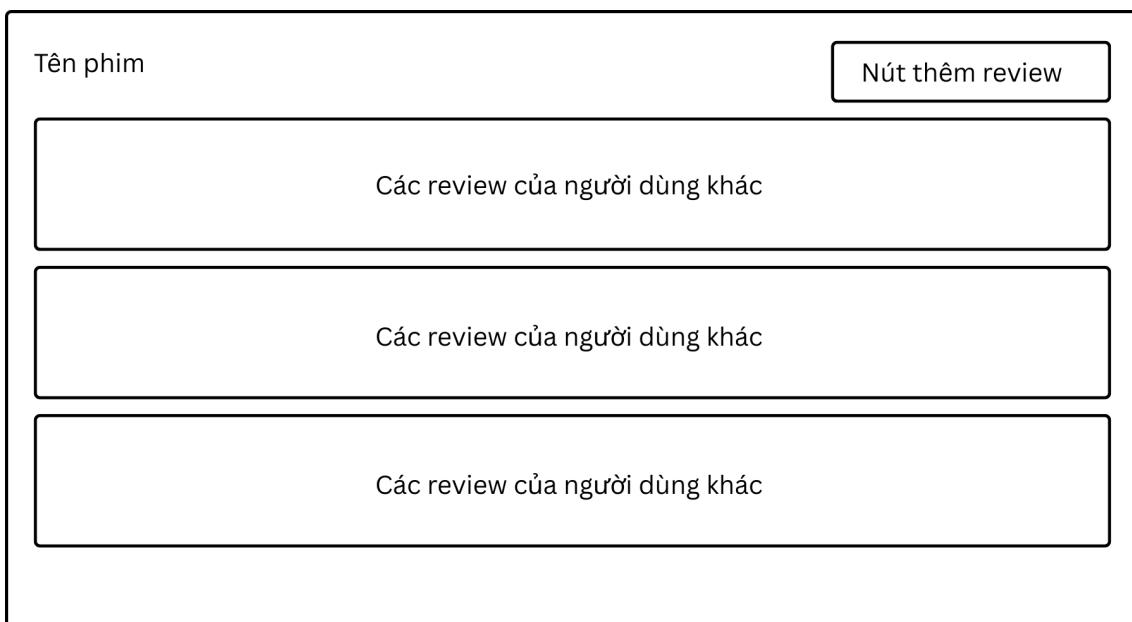
Hình 5.11: Màn hình *thông tin chi tiết 2*

Màn hình tổng hợp đạo diễn, diễn viên, ...



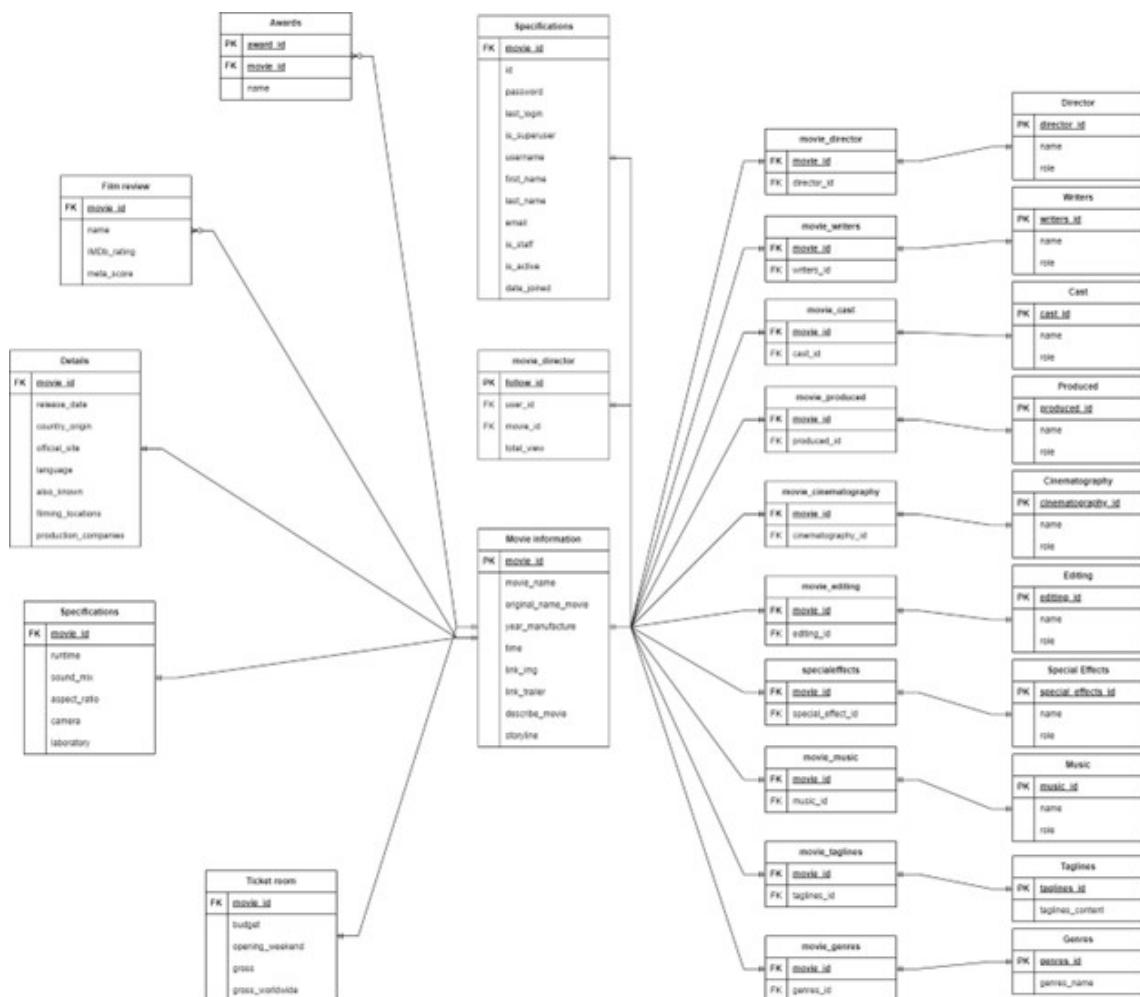
Hình 5.12: Màn hình *tổng hợp đạo diễn, diễn viên, ...*

Màn hình review phim



Hình 5.13: Màn hình *review phim*

5.2.2 Thiết kế cơ sở dữ liệu

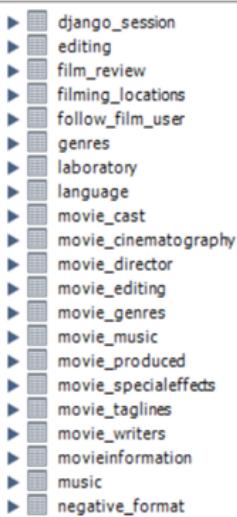
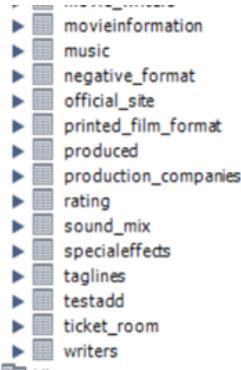


Hình 5.14: Sơ đồ ERD

Q. Filter objects

- ▶ aspect_ratio
- ▶ auth_group
- ▶ auth_group_permissions
- ▶ auth_permission
- ▶ auth_user
- ▶ auth_user_groups
- ▶ auth_user_user_permissions
- ▶ awards
- ▶ camera
- ▶ cast
- ▶ cinematographic_process
- ▶ cinematography
- ▶ color
- ▶ country_origin
- ▶ did_you_know
- ▶ director
- ▶ django_admin_log
- ▶ django_content_type
- ▶ django_migrations
- ▶ django_session

Hình 5.15: Hình ảnh các table trong database 1

**Hình 5.16:** Hình ảnh các table trong database 2**Hình 5.17:** Hình ảnh các table trong database 3

5.2.3 Thiết kế hệ thống crawl dữ liệu

IMDb: Ở trang web này, tôi chia ra các thông tin để crawl như sau:

Bảng 5.1: Tổng hợp các thông tin cần lấy

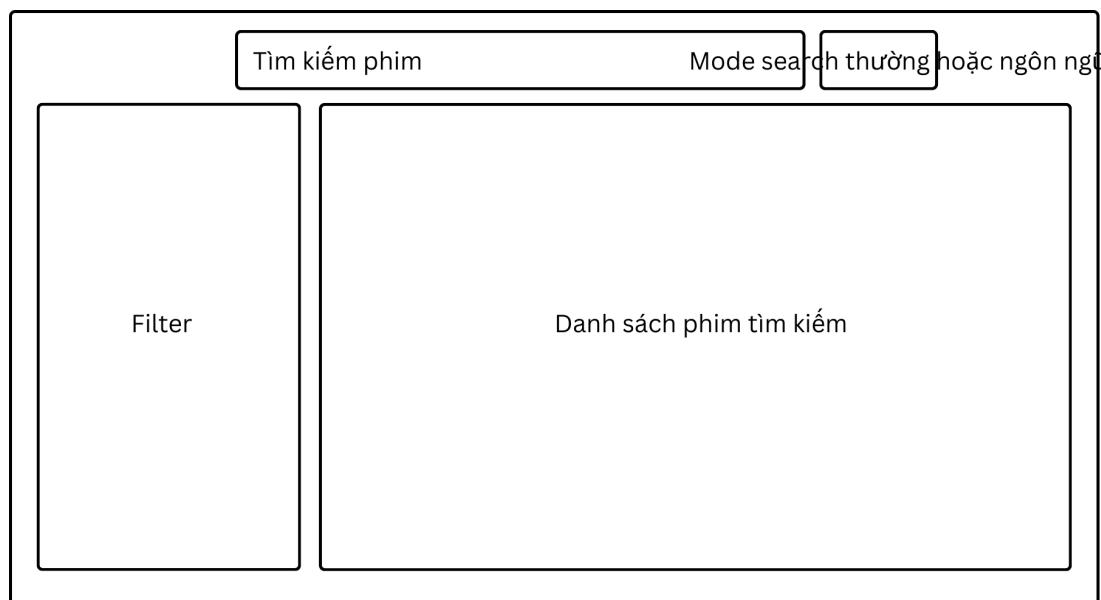
Page	Thông tin cần lấy
Home	Các thông tin cơ bản của một bộ phim như: Tên phim, ngày tháng năm sản xuất, mô tả
Award	Giải thưởng của phim
Director	Đạo diễn
Cast	Các diễn viên nổi tiếng và các diễn viên khác
Storyline	Cốt truyện, thể loại
Rating	Danh sách, số lượng đánh giá của người dùng
Review	Các đánh giá phim của người dùng khác

BoxofficeMojo: Ở trang web này, tôi sẽ tìm các bộ phim có trong Boxoffice-Mojo bằng cách lấy khóa chính là movie name có trong IMDb, sau đó sẽ lấy các thông tin về doanh thu và update vào data.

5.2.4 Tìm kiếm phim bằng ngôn ngữ tự nhiên

Ứng dụng hỗ trợ tìm kiếm phim bằng câu tự nhiên (natural language search), cho phép người dùng nhập yêu cầu tìm kiếm bằng ngôn ngữ thông thường thay vì phải sử dụng các từ khóa cụ thể. Tính năng này kết hợp hai phương pháp tìm kiếm: Vector Search (semantic search) và AI Filter (structured search) để đảm bảo độ chính xác và toàn diện trong kết quả tìm kiếm.

5.2.4.1 Tổng quan luồng xử lý



Hình 5.18: Màn hình *tìm kiếm phim*

Khi người dùng nhập câu tìm kiếm, ứng dụng sẽ xử lý theo các bước sau:

- Frontend gửi yêu cầu: Component React sẽ gửi câu tìm kiếm đến API endpoint.
- Backend xử lý song song: Ứng dụng thực hiện đồng thời hai phương pháp tìm kiếm:
 - Vector Search: Tìm kiếm theo ngữ nghĩa (semantic) dựa trên embedding
 - AI Filter: Tìm kiếm có cấu trúc (structured) dựa trên các thuộc tính cụ thể
- Kết hợp kết quả: Ứng dụng hợp nhất và sắp xếp kết quả từ hai phương pháp theo độ ưu tiên
- Trả về kết quả: Gửi danh sách phim kèm thông tin phân trang về frontend

5.2.4.2 Các thành phần kỹ thuật

Frontend: Frontend nhận sự kiện click button tìm kiếm và gửi yêu cầu đến API endpoint.

Backend API Endpoint: Backend kiểm tra tính hợp lệ của query, xử lý yêu cầu tìm kiếm. Xử lý phân trang và điều phối hai phương pháp tìm kiếm song song.

Vector Search (Semantic Search):

Thực hiện tìm kiếm theo ngữ nghĩa:

- Tạo embedding vector từ câu query bằng mô hình Gemini.
- Tải embeddings của các bộ phim đã được tính toán trước.
- Tính cosine similarity giữa embedding của query và embedding của mỗi phim
- Chọn top 200 kết quả có độ tương đồng cao nhất

Embedding của phim được tạo bởi kết hợp các thông tin: tên phim, năm sản xuất, mô tả, storyline, thể loại, đạo diễn, và diễn viên.

AI Filter (Structured Search):

Phân tích câu query bằng AI:

- Gửi câu query đến mô hình Gemini.
- AI trích xuất các thông tin có cấu trúc từ câu query thành định dạng JSON, bao gồm:
 - movie_name: Tên phim
 - year_min/year_max: Khoảng năm sản xuất
 - rating_min/rating_max: Khoảng đánh giá
 - genres: Danh sách thể loại
 - directors: Danh sách đạo diễn
 - cast: Danh sách diễn viên
 - countries: Danh sách quốc gia
 - languages: Danh sách ngôn ngữ
 - keywords: Các từ khóa khác
- Cuối cùng là thực hiện lọc phim dựa trên các tiêu chí được AI trích xuất

5.2.4.3 Chiến lược kết hợp kết quả

Ứng dụng hợp nhất kết quả từ hai phương pháp tìm kiếm theo ba mức độ ưu tiên:

- **Ưu tiên cao:** Các phim xuất hiện trong cả Vector Search và AI Filter - được sắp xếp theo độ tương đồng (similarity) từ cao xuống thấp

- Ưu tiên trung: Các phim chỉ xuất hiện trong Vector Search - tối đa 50 kết quả
- Ưu tiên thấp: Các phim chỉ xuất hiện trong AI Filter - tối đa 50 kết quả

Chiến lược này đảm bảo kết quả vừa phù hợp về mặt ngữ nghĩa, vừa đáp ứng các tiêu chí cụ thể mà người dùng yêu cầu.

5.2.4.4 Xử lý sau tìm kiếm

Sau khi có danh sách phim kết quả, ứng dụng thực hiện các bước xử lý sau:

- Serialize dữ liệu: Chuyển đổi đối tượng phim sang định dạng JSON để trả về cho frontend
- Lưu lịch sử tìm kiếm: Nếu người dùng đã đăng nhập, thì hệ thống sẽ lưu lại lịch sử tìm kiếm với định dạng JSON để phục vụ cho ứng dụng gợi ý phim trong tương lai

5.2.4.5 Kết quả trả về

API trả về response với cấu trúc:

- data: Danh sách phim đã được sắp xếp và lọc
- pagination: Thông tin phân trang (số trang hiện tại, tổng số trang, tổng số kết quả)
- filter_data: Thông tin các bộ lọc mà AI đã phân tích từ câu query, giúp người dùng hiểu rõ hơn

5.2.4.6 Điểm mạnh của ứng dụng

Tính năng tìm kiếm bằng ngôn ngữ người dùng có các ưu điểm sau:

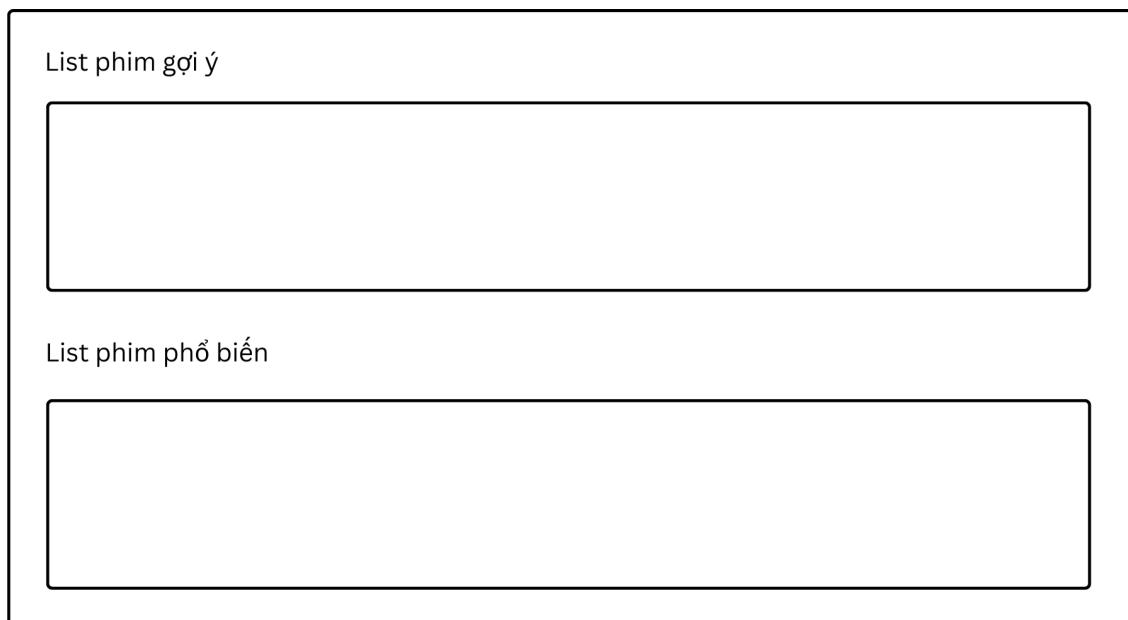
- Kết hợp hai phương pháp: Sự kết hợp giữa semantic search và structured search giúp cải thiện đáng kể độ chính xác của kết quả tìm kiếm
- Hỗ trợ đa ngôn ngữ: Thông qua mô hình Gemini, ứng dụng có thể xử lý câu tìm kiếm bằng nhiều ngôn ngữ khác nhau
- Lưu lịch sử: Ứng dụng tự động lưu lịch sử tìm kiếm của người dùng đã đăng nhập, hỗ trợ cho các tính năng gợi ý và phân tích hành vi
- Tối ưu hiệu suất: Embeddings được tính toán trước và lưu trong file pickle, giúp giảm thời gian xử lý khi tìm kiếm

5.2.5 Gợi ý phim bằng genAI dựa vào lịch sử hoạt động của người dùng

Ứng dụng sử dụng GenAI (Gemini hoặc OpenAI) để phân tích lịch sử hoạt động của người dùng và đưa ra gợi ý phim cá nhân hóa theo thời gian thực. Khác với các phương pháp gợi ý truyền thống yêu cầu training model, hệ thống này hoạt động

real-time dựa trên dữ liệu lịch sử của từng người dùng, không cần retrain hay cập nhật model.

5.2.5.1 Tổng quan luồng xử lý



Hình 5.19: Màn hình gợi ý phim bằng genAI

Khi người dùng yêu cầu gợi ý phim, ứng dụng sẽ xử lý theo các bước sau:

- Thu thập lịch sử người dùng: Ứng dụng lấy lịch sử tìm kiếm và lịch sử đánh giá của người dùng
- Xây dựng prompt cho GenAI: Ứng dụng tạo prompt chứa thông tin lịch sử và yêu cầu AI phân tích sở thích
- Gọi GenAI API: Ứng dụng gửi prompt đến GenAI (ưu tiên Gemini, fallback OpenAI) để nhận danh sách phim được gợi ý
- Tìm phim trong database: Ứng dụng tìm các phim được AI gợi ý trong database
- Trả về kết quả: Gửi danh sách phim đã được gợi ý về frontend

5.2.5.2 Các thành phần kỹ thuật

Frontend: Frontend gửi yêu cầu gợi ý phim đến API endpoint và hiển thị kết quả trả về.

Backend API Endpoint: Backend xử lý yêu cầu, thu thập lịch sử người dùng và gọi GenAI để tạo gợi ý.

Thu thập lịch sử hoạt động:

Ứng dụng thu thập ba loại dữ liệu lịch sử:

- Lịch sử tìm kiếm: Tự động lưu lại các câu tìm kiếm của người dùng, bao gồm cả tìm kiếm thường và tìm kiếm ngôn ngữ tự nhiên. Ứng dụng lấy tối đa 20 query gần nhất để phân tích
- Lịch sử đánh giá: Lưu trữ các review phim của người dùng, bao gồm điểm đánh giá (1-5 sao), tiêu đề và nội dung review. Ứng dụng lấy tối đa 20 review gần nhất, ưu tiên các review có rating cao (4-5 sao)
- Theo dõi hoạt động: Ghi nhận các hành động của người dùng như xem chi tiết phim, click vào card phim, xem trailer, hoặc click vào kết quả tìm kiếm. Mỗi loại hoạt động có mức độ quan trọng khác nhau

Xây dựng prompt cho GenAI:

Ứng dụng tạo prompt chứa:

- Lịch sử tìm kiếm: Liệt kê các câu tìm kiếm gần nhất của người dùng
- Lịch sử đánh giá: Liệt kê các phim đã được người dùng đánh giá kèm điểm số và tiêu đề review
- Yêu cầu phân tích: Genres người dùng tìm kiếm, themes và keywords, phim được đánh giá tích cực (4-5 sao), phim tương tự những gì đã review, xu hướng trong pattern tìm kiếm
- Yêu cầu trả về: Danh sách tên các phim được gợi ý dưới dạng JSON array

Gọi GenAI API:

Ứng dụng ưu tiên sử dụng Gemini API để phân tích lịch sử và đưa ra gợi ý. Nếu Gemini không khả dụng, ứng dụng sẽ fallback sang OpenAI API để đảm bảo tính khả dụng của hệ thống.

Xử lý response từ GenAI:

- Parse JSON từ response (loại bỏ markdown code blocks nếu có)
- Validate format (phải là array chứa tên phim)
- Giới hạn số lượng phim theo yêu cầu

Tìm phim trong database:

Với mỗi tên phim được AI gợi ý, ứng dụng tìm phim tương ứng trong database:

- Nếu phim không tìm thấy: Bỏ qua phim đó
- Chỉ trả về danh sách phim hợp lệ có trong database

5.2.5.3 Kết quả trả về

API trả về response với cấu trúc:

- message: Thông báo trạng thái
- data: Danh sách phim đã được gợi ý dưới dạng JSON array

5.3 Xây dựng ứng dụng

5.3.1 Thư viện và công cụ sử dụng

Sau đây là các công cụ, ngôn ngữ lập trình, API, thư viện, IDE, công cụ kiểm thử mà tôi sử dụng để phát triển ứng dụng này.

Bảng 5.2: Danh sách thư viện và công cụ sử dụng

Mục đích	Công cụ	Tài liệu tham khảo
IDE lập trình	Visual Studio Code (VSCode)	[8]
API Framework	Django Rest Framework	[9]
Frontend	HTML	[10]
Style	Tailwind CSS	[11]
Frontend Framework	React	[12]
Backend Framework	Django	[13]
Database	MySQL	[14]
Công cụ crawl dữ liệu	Selenium	[15]
Thư viện HTTP	Requests	[16]
Công cụ CI/CD	GitHub Actions	[17]

5.3.2 Kết quả đạt được

Mô tả kết quả đạt được: Sau khi hoàn thành quá trình phát triển, hệ thống Django đã được xây dựng với các sản phẩm được đóng gói như sau:

Gói Model: Bao gồm các mô hình dữ liệu phục vụ việc quản lý thông tin về phim, người dùng, và đánh giá.

Gói View: Bao gồm các class đảm nhiệm việc hiển thị và xử lý giao diện cho cả người dùng và quản trị viên.

Gói Template: Bao gồm các template HTML được sử dụng cho giao diện người dùng và giao diện quản trị.

Các sản phẩm được đóng gói đóng vai trò và có ý nghĩa quan trọng trong việc quản lý và hiển thị thông tin về phim, đạo diễn, diễn viên, và đánh giá phim. Hệ thống này cung cấp một nền tảng cho phép người dùng có thể thực hiện tìm kiếm và xem thông tin về các bộ phim, viết đánh giá, và tương tác với nội dung. Bên cạnh đó, hệ thống còn có khả năng đề xuất những bộ phim phù hợp nhất với từng

hành động của người dùng thông qua hệ thống gợi ý.

Thống kê các thông tin của ứng dụng:

Số dòng code crawl dữ liệu: 2000 dòng

Số dòng code phía backend: 4000 dòng

Số dòng code phía frontend: 8000 dòng

Số lớp: 180 lớp

Số gói: 8 gói

Dung lượng toàn bộ mã nguồn: Khoảng 310MB khi đã có hệ thống gợi ý và dữ liệu, còn nếu mỗi phần code thì dung lượng khoảng 25MB

Hệ thống đã được hoàn thiện với một kiến trúc có tổ chức rõ ràng, nơi mã nguồn được phân bổ hợp lý giữa các gói và các lớp. Những con số thống kê cụ thể liên quan đến số lượng dòng mã, số lượng lớp và kích thước mã nguồn đóng vai trò hỗ trợ trong công tác đánh giá và giám sát tiến độ phát triển dự án. Hệ thống đã thỏa mãn toàn bộ các yêu cầu đã được đặt ra và đã sẵn sàng cho việc triển khai và vận hành trong môi trường thực tế.

Crawl dữ liệu từ trang web:

Nguồn dữ liệu: IMDb, BoxOfficeMojo

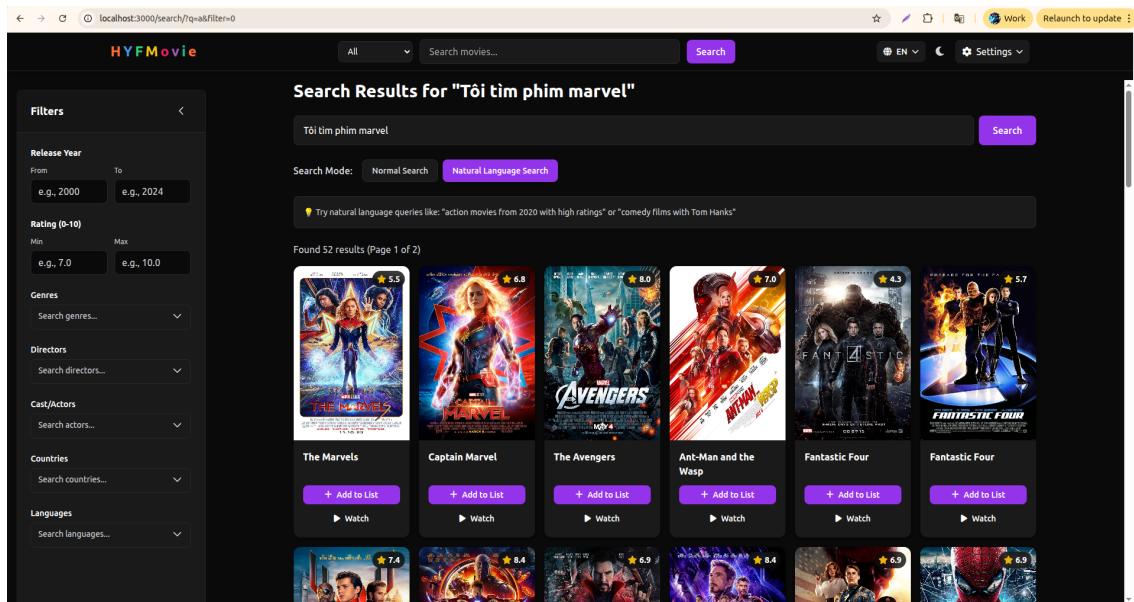
Số bộ phim đã crawl: 5000 bộ phim

Số ảnh đã crawl: 250000 ảnh

Số video đã crawl: 5000 video

Tìm kiếm phim theo ngôn ngữ người dùng:

Ở chức năng này, người dùng có thể tìm kiếm phim theo ngôn ngữ của mình. Ứng dụng sẽ nhận đầu vào, phân tích, embed text vào model để tìm kiếm phim theo ngôn ngữ của người dùng và đưa ra kết quả phù hợp.

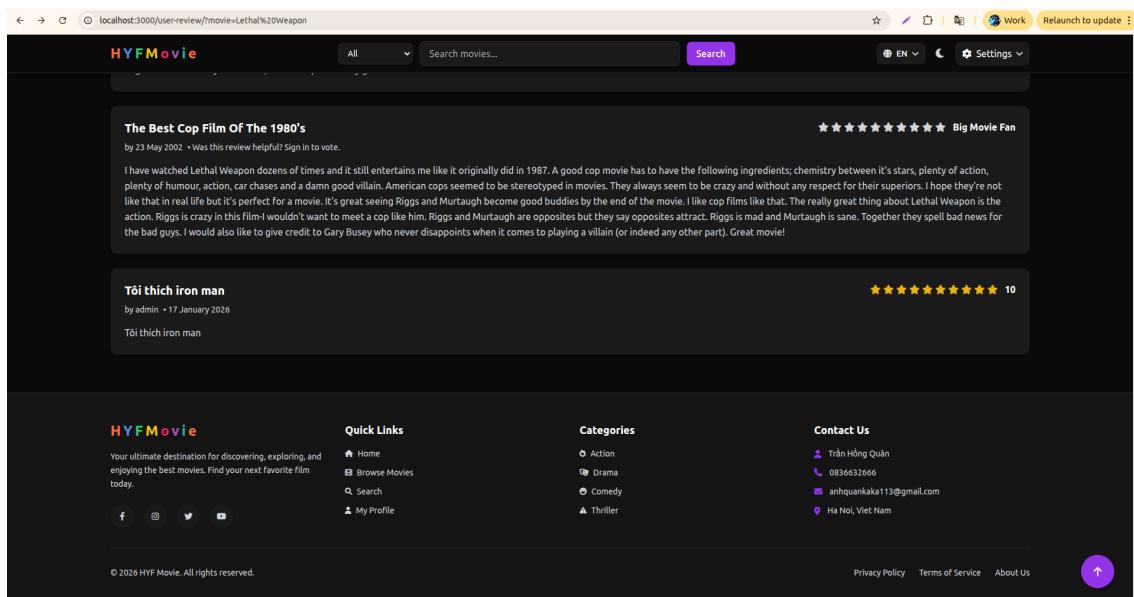


Hình 5.20: Chức năng tìm kiếm phim theo ngôn ngữ người dùng

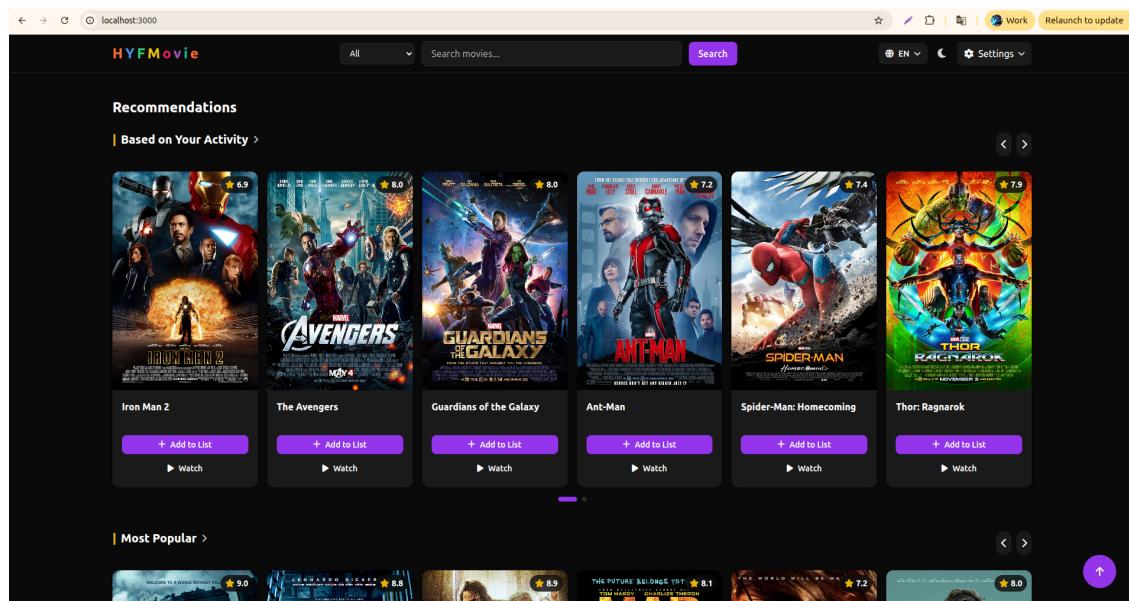
Dựa vào hình ảnh kết quả trên, có thể thấy hệ thống đã đưa ra kết quả phù hợp với yêu cầu của người dùng. Ở trên là các phim marvel hệ thống đã đưa ra kết quả.

Gợi ý phim theo lịch sử hoạt động của người dùng:

Ở chức năng này, hệ thống sẽ gợi ý phim dựa vào lịch sử hoạt động của người dùng. Hệ thống sẽ theo dõi lịch sử hoạt động của người dùng như review phim, tìm kiếm phim, xem chi tiết phim, ... và đưa ra gợi ý các bộ phim tương tự với lịch sử hoạt động của người dùng.

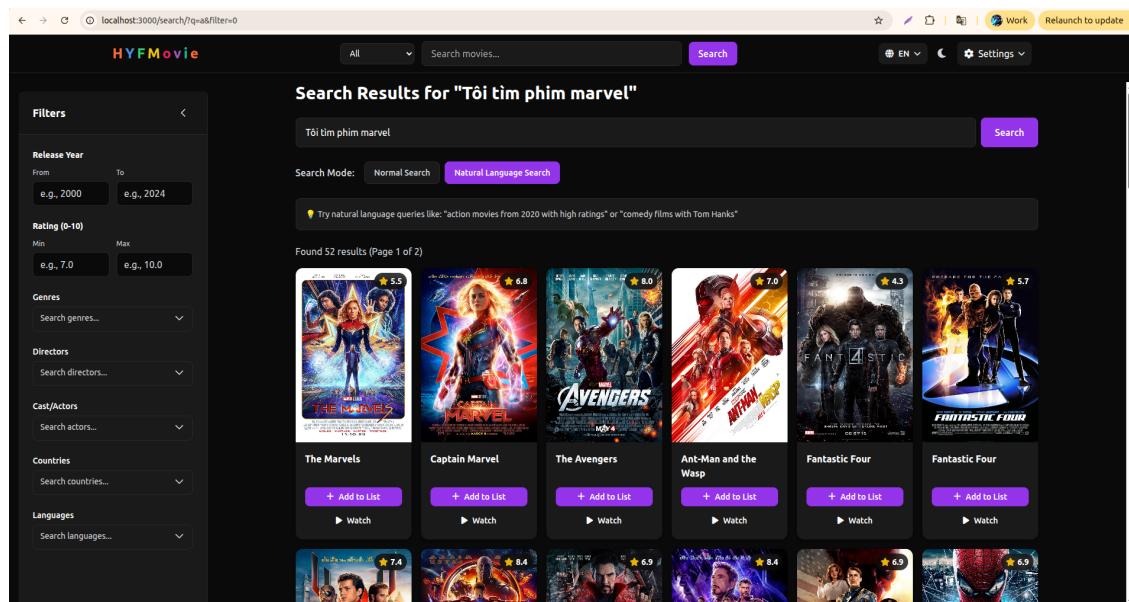


Hình 5.21: Review phim

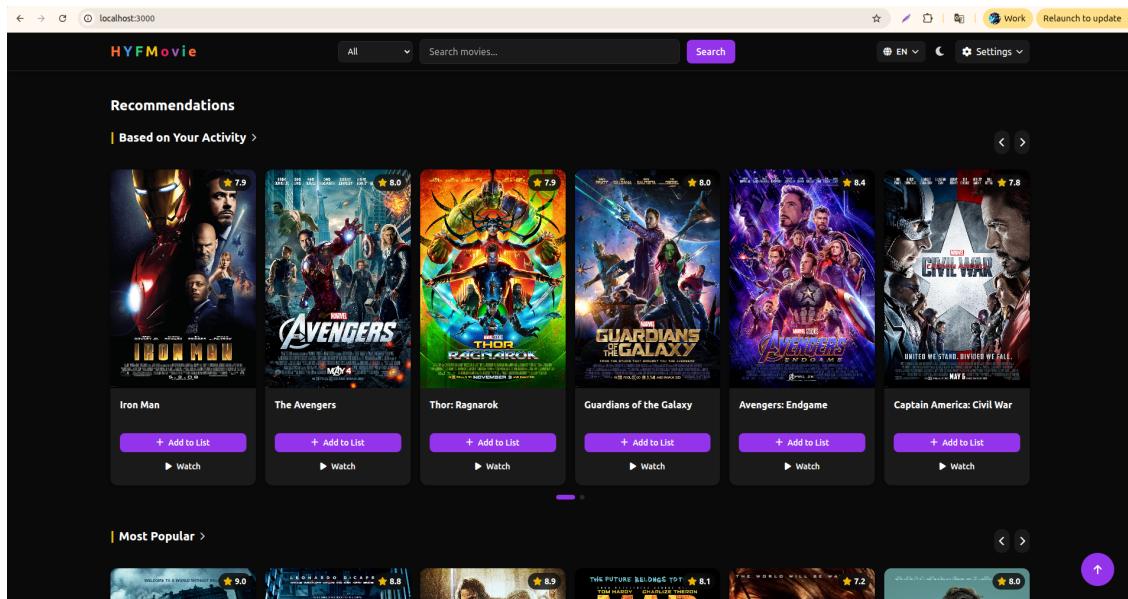


Hình 5.22: Gợi ý phim theo lịch sử review

Dựa vào lịch sử review của người dùng, hệ thống có thể tự học và đưa ra được kết quả mới nhất phù hợp với sở thích của người dùng. Ở phần trên, review phim người dùng bảo thích phim Iron Man, hệ thống đã đưa ra gợi ý phim tương tự là Iron man 1, 2, 3 và các bộ phim Marvel liên quan.



Hình 5.23: Tìm kiếm phim

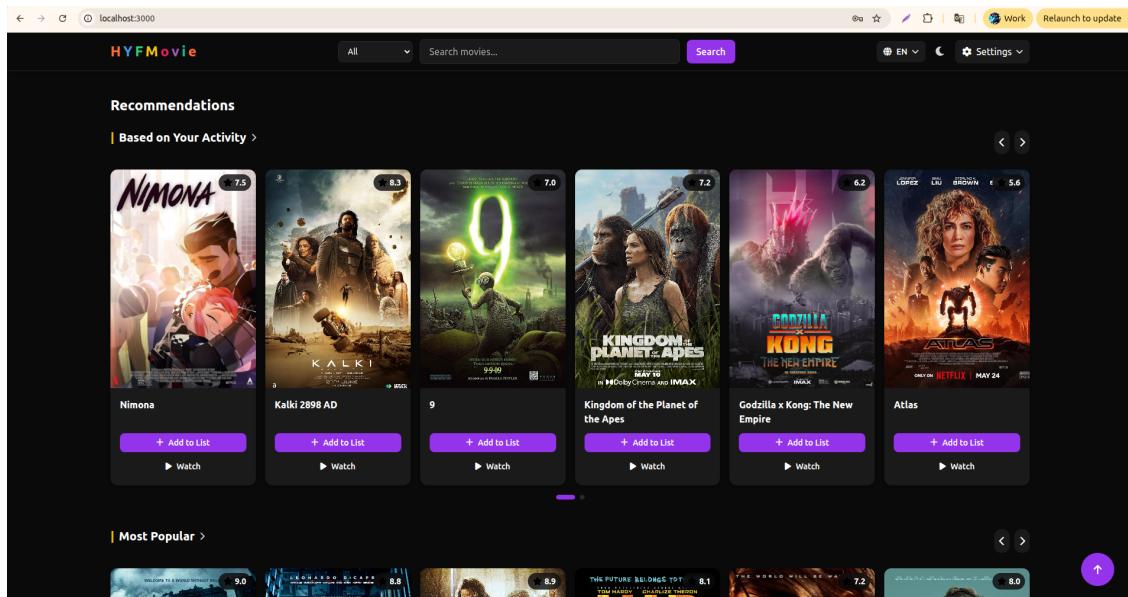


Hình 5.24: Gợi ý phim theo lịch sử tìm kiếm

Dựa vào lịch sử tìm kiếm của người dùng, hệ thống lưu lại lịch sử và đưa ra được kết quả mới nhất phù hợp với yêu cầu của người dùng. Ở phần trên, người dùng tìm kiếm phim Marvel, thì sau khi ra màn Home, hệ thống đã đưa ra gợi ý các phim Marvel liên quan.

5.3.3 Minh họa các chức năng chính

Chức năng gợi ý phim dựa vào lịch sử thao tác của người dùng

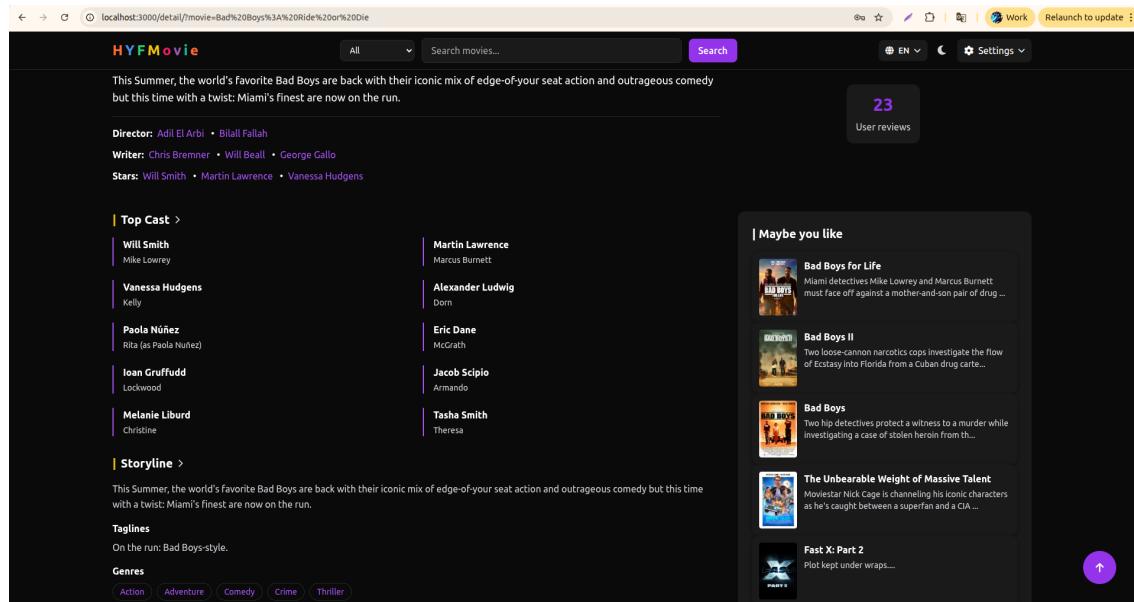


Hình 5.25: Chức năng gợi ý phim dựa vào lịch sử thao tác của người dùng

Thông qua việc phân tích lịch sử thêm phim vào danh sách yêu thích của người dùng, hệ thống có khả năng tự động học hỏi và đề xuất các kết quả mới nhất phù

hợp với sở thích cá nhân của từng người dùng.

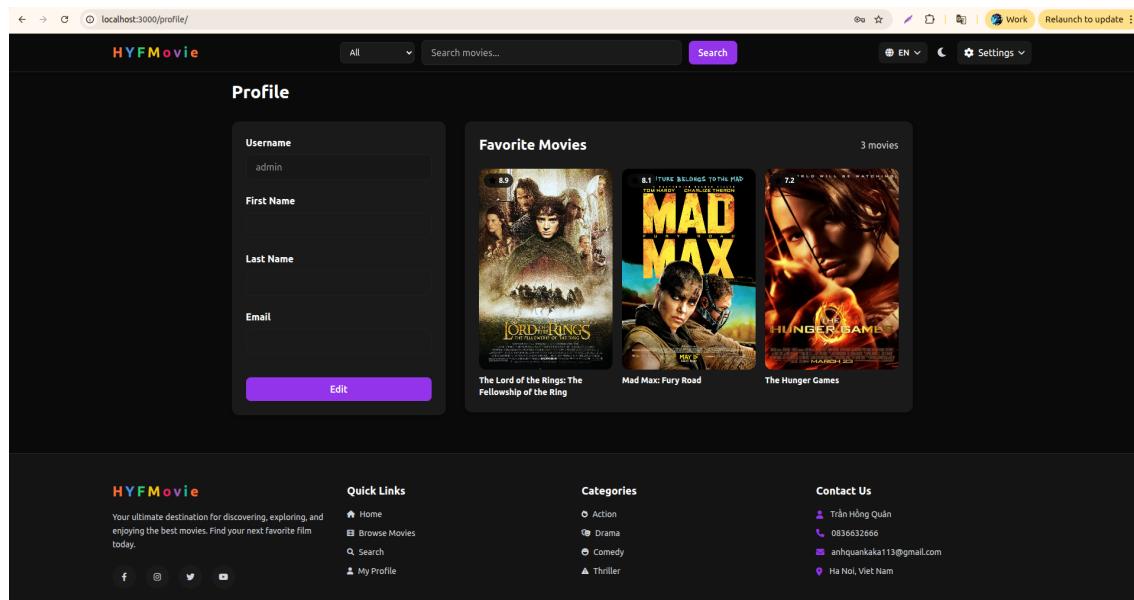
Chức năng gợi ý phim dựa vào nội dung phim đang xem



Hình 5.26: Chức năng gợi ý phim dựa vào nội dung phim đang xem

Dựa trên bộ phim mà người dùng đang quan tâm, hệ thống có khả năng đề xuất các bộ phim có tính chất tương đồng với bộ phim hiện tại.

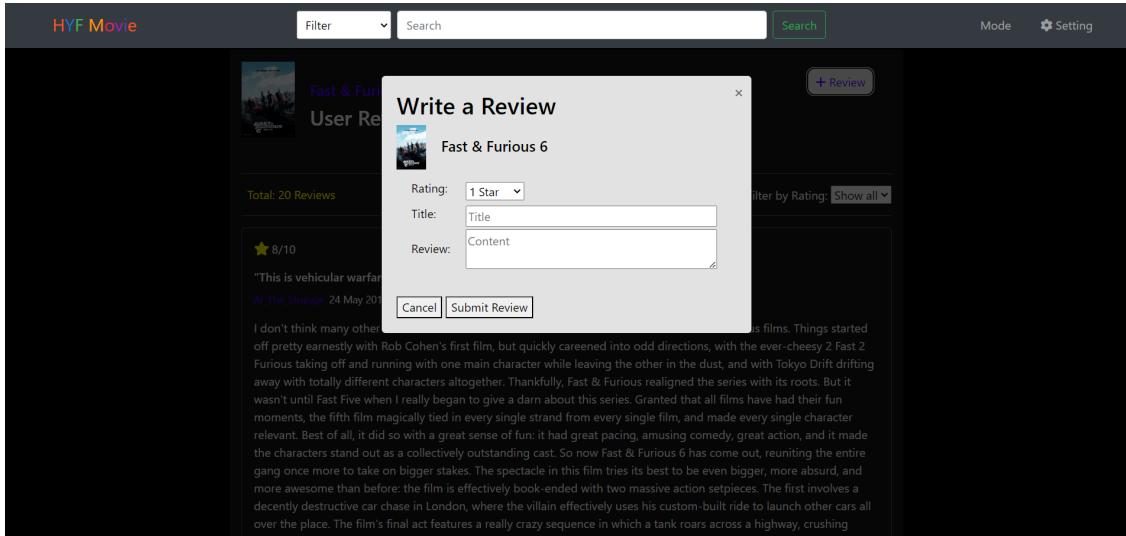
Chức năng thêm phim vào danh sách yêu thích



Hình 5.27: Chức năng thêm phim vào danh sách yêu thích

Khi người dùng thể hiện sự quan tâm đối với một bộ phim cụ thể, họ có thể sử dụng nút thêm vào danh sách yêu thích để lưu phim đó vào danh sách cá nhân của mình.

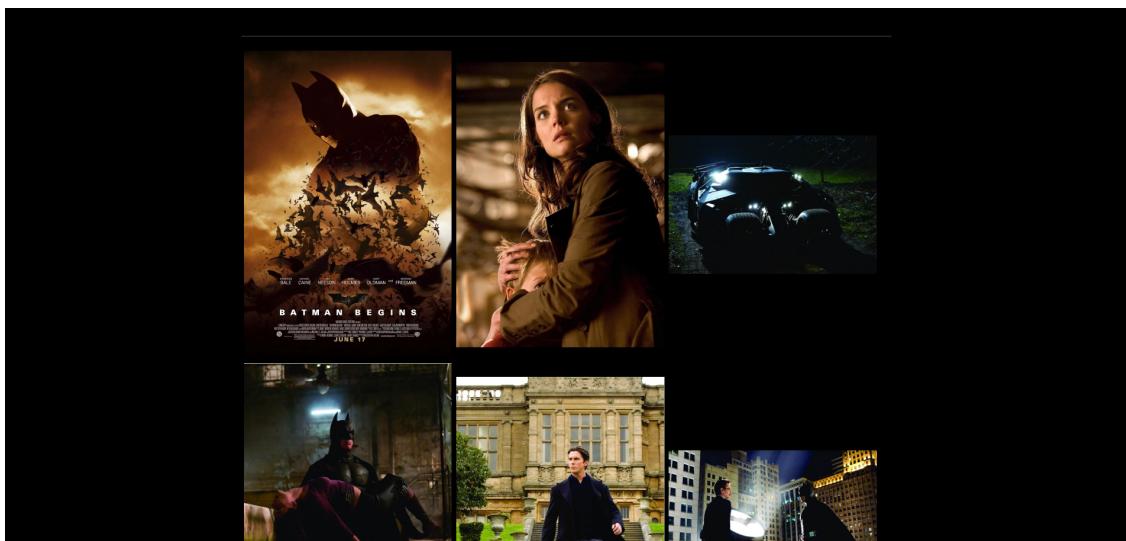
Chức năng review phim



Hình 5.28: Chức năng review phim

Người dùng có thể truy cập vào trang chi tiết phim để xem các bình luận và đánh giá từ cộng đồng, đồng thời có thể tự mình viết đánh giá về bộ phim đó.

Chức năng xem toàn bộ ảnh của phim



Hình 5.29: Chức năng xem toàn bộ ảnh của phim

Người dùng có thể truy cập vào trang chi tiết phim để xem toàn bộ danh sách hình ảnh liên quan đến bộ phim, các dữ liệu này được thu thập và tổng hợp từ các nguồn đáng tin cậy.

5.4 Kiểm thử

Bảng 5.3: Kiểm thử chức năng

Chức năng	Mô tả	Các bước kiểm thử	Kết quả mong đợi	Kết quả
Đăng ký người dùng mới	Kiểm thử chức năng đăng ký người dùng mới với thông tin hợp lệ.	Mở trang đăng ký. Nhập thông tin đăng ký hợp lệ (tên, email, mật khẩu). Nhấn nút "Đăng ký".	Người dùng được đăng ký thành công và chuyển hướng đến trang đăng nhập.	Thành công
Đăng ký người dùng với email đã tồn tại	Kiểm thử chức năng đăng ký người dùng với email đã tồn tại trong hệ thống.	Mở trang đăng ký. Nhập thông tin đăng ký với email đã tồn tại. Nhấn nút "Đăng ký".	Hiển thị thông báo lỗi "Email đã tồn tại".	Thành công
Đăng nhập người dùng	Kiểm thử chức năng đăng nhập với thông tin hợp lệ.	Mở trang đăng nhập. Nhập thông tin đăng nhập hợp lệ (email, mật khẩu). Nhấn nút "Đăng nhập".	Người dùng đăng nhập thành công và chuyển hướng đến trang chủ.	Thành công
Đăng nhập với mật khẩu sai	Kiểm thử chức năng đăng nhập với mật khẩu sai.	Mở trang đăng nhập. Nhập email hợp lệ và mật khẩu sai. Nhấn nút "Đăng nhập".	Hiển thị thông báo lỗi "Mật khẩu không đúng".	Thành công
Xem danh sách phim	Kiểm thử chức năng xem danh sách phim.	Mở trang danh sách phim.	Hiển thị danh sách phim với thông tin cơ bản (tên phim, đạo diễn, thể loại).	Thành công
Tìm kiếm phim	Kiểm thử chức năng tìm kiếm phim theo tên.	Nhập tên phim vào ô tìm kiếm. Nhấn nút "Tim kiem".	Hiển thị kết quả tìm kiếm với các phim có tên chứa từ khóa.	Thành công
Xem chi tiết phim	Kiểm thử chức năng xem chi tiết phim.	Chọn một phim từ danh sách.	Hiển thị trang chi tiết phim với thông tin đầy đủ về phim (tên, đạo diễn, diễn viên, mô tả, đánh giá, v.v.).	Thành công
Thêm đánh giá phim	Kiểm thử chức năng thêm đánh giá phim.	Đăng nhập vào hệ thống. Mở trang chi tiết phim. Nhập đánh giá và nhấn nút "Gửi đánh giá".	Đánh giá được thêm vào và hiển thị trên trang chi tiết phim.	Thành công
Thêm đánh giá phim khi chưa đăng nhập	Kiểm thử chức năng thêm đánh giá phim khi chưa đăng nhập.	Mở trang chi tiết phim. Nhập đánh giá và nhấn nút "Gửi đánh giá".	Hiển thị thông báo yêu cầu đăng nhập trước khi thêm đánh giá.	Thành công

Bảng 5.4: Kiểm thử chức năng

Thích phim	Kiểm thử chức năng thích phim.	Đăng nhập vào hệ thống. Mở trang chi tiết phim. Nhấn nút "Thích".	Phim được thêm vào danh sách yêu thích của người dùng.	Thành công
Xem thông tin cá nhân	Kiểm thử chức năng xem thông tin cá nhân.	Đăng nhập vào hệ thống. Mở trang thông tin cá nhân.	Hiển thị thông tin cá nhân của người dùng.	Thành công
Sửa thông tin cá nhân	Kiểm thử chức năng sửa thông tin cá nhân.	Đăng nhập vào hệ thống. Mở trang thông tin cá nhân. Sửa thông tin và nhấn nút "Lưu".	Thông tin cá nhân được cập nhật thành công.	Thành công
Đăng xuất	Kiểm thử chức năng đăng xuất.	Đăng nhập vào hệ thống. Nhấn nút "Đăng xuất".	Người dùng được đăng xuất và chuyển hướng đến trang đăng nhập.	Thành công
Xem danh sách phim	Kiểm thử chức năng xem danh sách phim yêu thích.	Đăng nhập vào hệ thống. Mở trang danh sách phim yêu thích.	Hiển thị danh sách phim mà người dùng đã thích.	Thành công
Xem phim gợi ý trong phần chi tiết phim	Kiểm thử chức năng gọi ý phim tương tự	Vào trang chi tiết phim	Hiển thị list phim tương tự với phim đang xem	Thành công
Xem phim gợi ý ở trang chủ	Kiểm thử chức năng gọi ý phim theo sở thích	Vào trang chủ	Hiển thị list phim gợi ý cho người dùng	Thành công

Bảng 5.5: Bảng danh sách kiểm thử hiệu năng

Tên kiểm thử	Endpoint/Method	Yêu cầu	Mã
GET Danh sách phim	GET /api/v1/movies/	< 3s	TC01
GET Chi tiết phim	POST /api/v1/movies/1/	< 3s	TC02
GET Lọc phim	GET /api/v1/movies/filter-options/	< 3s	TC03
GET Metadata phim	GET /api/v1/movies/1/cast/	< 3s	TC04
POST Đăng ký	POST /api/v1/users/register/	< 3s	TC05
POST Đăng nhập	POST /api/v1/users/login/	< 3s	TC06
Tìm kiếm AI (Mock)	POST /api/v1/movies/search/natural/	< 10s	TC07
Gợi ý AI (Mock)	GET /api/v1/movies/recommend/realtime/perf_test_user/	< 10s	TC08
Định dạng response nhất quán	GET /api/v1/movies/	Có	TC09
Thông báo lỗi rõ ràng	POST /api/v1/movies/999999/	Có	TC10
Validation đầu vào	POST /api/v1/users/login/	Có	TC11
Documentation trong code	Code structure	Có	TC12
Tổ chức code theo modules	Code organization	Có	TC13
Nhất quán RESTful API	Multiple endpoints	Có	TC14
Kết nối database	Database connection	< 1s	TC15
Hiệu năng query cơ bản	SELECT query	< 1s	TC16
Tính toàn vẹn dữ liệu	Data integrity	Có	TC17
Xử lý transaction	Transaction	Có	TC18

Bảng 5.6: Bảng kết quả kiểm thử hiệu năng

Mã	Yêu cầu	Thời gian thực tế (s)	Status Code	Kết quả
TC01	< 3s	0.009	200	Đạt
TC02	< 3s	0.004	200	Đạt
TC03	< 3s	0.099	200	Đạt
TC04	< 3s	0.007	200	Đạt
TC05	< 3s	0.001	400	Đạt
TC06	< 3s	0.184	200	Đạt
TC07	< 10s	4.6	200	Đạt
TC08	< 10s	2.9	200	Đạt
TC09	Có	0.000	200	Đạt
TC10	Có	0.000	404	Đạt
TC11	Có	0.000	400	Đạt
TC12	Có	0.000	200	Đạt
TC13	Có	0.000	200	Đạt
TC14	Có	0.000	200	Đạt
TC15	< 1s	0.000	200	Đạt
TC16	< 1s	0.005	200	Đạt
TC17	Có	0.000	200	Đạt
TC18	Có	0.000	200	Đạt

Thông qua các bước kiểm thử đã được trình bày ở trên (Table 5.3, Table 5.4, Table 5.5, Table 5.6), có thể nhận thấy rằng hệ thống đã được kiểm thử thành công và sẵn sàng để đưa vào vận hành khi tất cả các test case đều đạt yêu cầu và hoạt động ổn định.

5.5 Triển khai

Mô hình triển khai Hệ thống được triển khai dựa trên mô hình client-server, trong đó phần frontend được chạy trên live server và phần backend được triển khai sử dụng Django trên máy tính cá nhân.

Cấu hình triển khai: Máy tính cá nhân với thông số:

Hệ điều hành: Windows, Linux, MacOS

CPU: Intel i3 hoặc tương đương

RAM: 4GB

Ổ cứng: 128GB SSD

Kết nối mạng: Internet tốc độ trung bình

Triển khai

Thực hiện clone repository từ GitHub về máy tính cá nhân.

Tiến hành cài đặt Docker vào máy tính.

Di chuyển vào thư mục dự án và thực thi lệnh docker-compose up -d để khởi chạy hệ thống. Sau khi hoàn thành, toàn bộ hệ thống đã được vận hành trên máy tính cá nhân. Truy cập vào cổng localhost:3000 để kiểm thử hệ thống.

Kết quả triển khai thử nghiệm

Mặc dù việc triển khai chỉ được thực hiện trong môi trường local, nhưng một số kết quả thử nghiệm đã được ghi nhận như sau:

Khả năng chịu tải: Trong môi trường local, khả năng chịu tải của hệ thống chưa được đánh giá một cách toàn diện. Tuy nhiên, với số lượng truy cập thử nghiệm ở mức nhỏ, hệ thống vận hành mượt mà và không gặp phải các vấn đề liên quan đến hiệu suất.

Thời gian phản hồi: Thời gian phản hồi trung bình đạt dưới 2 giây đối với các yêu cầu cơ bản, trong khi đối với chức năng gợi ý phim thì thời gian trung bình dưới 5 giây, và đối với chức năng tìm kiếm phim thì dưới 10 giây.

Phản hồi người dùng: Các người dùng thử nghiệm đánh giá tích cực về giao diện đơn giản, dễ sử dụng và tính năng tìm kiếm, nhận định rằng hệ thống hoạt động hiệu quả. Đặc biệt, các chức năng tìm kiếm phim và gợi ý đưa ra các kết quả phù hợp với mong đợi của người dùng.

Việc triển khai hệ thống trên môi trường local đã góp phần xác minh các chức năng cơ bản và thu thập được những phản hồi tích cực từ phía người dùng thử nghiệm. Các kết quả thử nghiệm cho thấy hệ thống vận hành ổn định, có thời gian phản hồi nhanh và giao diện thân thiện với người dùng. Để có thể đánh giá hiệu suất và khả năng chịu tải một cách chính xác hơn, cần thực hiện triển khai trên server thực tế.

CHƯƠNG 6. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT

Chương này tập trung trình bày các vấn đề đã phát sinh trong quá trình thực hiện và các giải pháp cụ thể đã được áp dụng để giải quyết những vấn đề đó. Bên cạnh đó, chương cũng đề cập đến các đóng góp nổi bật được đánh giá cao trong suốt quá trình thực hiện đồ án tốt nghiệp.

6.1 Thiếu khá nhiều dữ liệu phim ảnh

Vấn đề này xuất hiện khi cần một khối lượng dữ liệu lớn về các bộ phim để phục vụ việc xây dựng ứng dụng gợi ý. Để khắc phục vấn đề này, tôi đã thực hiện tìm kiếm, thu thập và lựa chọn các trang web đáng tin cậy để thu thập dữ liệu. Quá trình này bao gồm việc phân tích cú pháp của các trang web, xác định phương thức truy cập, nghiên cứu cấu trúc và thực hiện thu thập dữ liệu một cách tự động.

6.2 Thiết kế database khá khó khăn với quá nhiều trường

Trong quá trình thiết kế cơ sở dữ liệu, tôi gặp phải thách thức khi phải xử lý một khối lượng lớn các trường thông tin liên quan đến phim và người dùng. Để khắc phục vấn đề này, tôi đã vận dụng các nguyên tắc chuẩn hóa cơ sở dữ liệu và phân tách các trường thông tin có liên quan thành các bảng dữ liệu con. Tiếp theo, tôi xác định các mối quan hệ giữa các bảng và liên kết chúng lại với nhau nhằm tạo ra một cấu trúc cơ sở dữ liệu hợp lý và đạt hiệu quả cao.

6.3 Trang web có dữ liệu update liên tục theo ngày khiến link ảnh và trailer bị hỏng

Một trong các trang web được sử dụng để thu thập dữ liệu là IMDb, trang web này chứa rất nhiều trường thông tin, tuy nhiên việc khó khăn nhất trong quá trình thu thập dữ liệu đối với tôi chính là việc lấy các liên kết ảnh và liên kết trailer. Trên trang web này, cứ mỗi ngày các liên kết ảnh và trailer sẽ được làm mới một lần, do đó các liên kết cũ sẽ trở nên không hợp lệ và không thể truy cập được vào ngày hôm sau. Tôi đã đề xuất hai phương án để giải quyết vấn đề này: Tải toàn bộ ảnh và trailer về máy hoặc tìm cách để hệ thống tự động cập nhật dữ liệu.

Về phương án tải xuống toàn bộ hình ảnh và video trailer, tôi đã từ bỏ ngay khi quyết định số lượng phim cần thiết cho bản demo. Nếu có 5000 bộ phim thì sẽ có hơn 250000 hình ảnh và 5000 video demo, việc tải xuống sẽ khiến máy tính trở nên cồng kềnh và không hiệu quả. Vì vậy, tôi đã tìm hiểu và phát hiện ra một giải pháp để hệ thống có thể tự động cập nhật dữ liệu, đó là việc sử dụng GitHub Actions.

Giới thiệu Github Action:

GitHub Actions là một dịch vụ được GitHub cung cấp nhằm tự động hóa các

quy trình phát triển phần mềm. Thông qua việc sử dụng các action, tôi có thể thiết lập các workflow tự động, bao gồm việc kiểm tra mã nguồn, triển khai ứng dụng, và tự động hóa các tác vụ như cập nhật dữ liệu trong hệ thống.

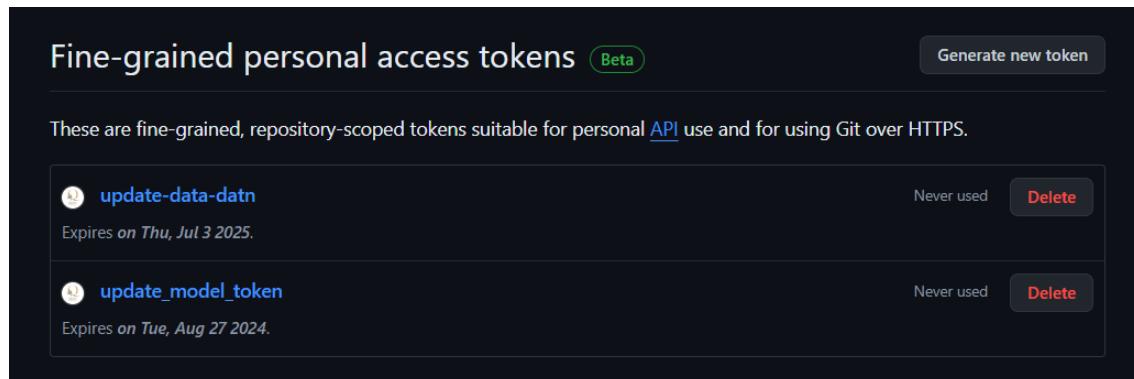
Triển khai:

Ở bước đầu tiên, tôi xây dựng một file crawlJson.py nhằm thu thập toàn bộ dữ liệu mới từ website IMDb và ghi vào file update.sql, một khi đã có file update.sql thì chỉ việc import vào môi trường local là có thể có dữ liệu mới ngay.

Nhằm cho phép hệ thống tự động cập nhật dữ liệu, quy trình thực hiện bao gồm các bước sau:

1. Tạo PAT (Personal Access Token) Github

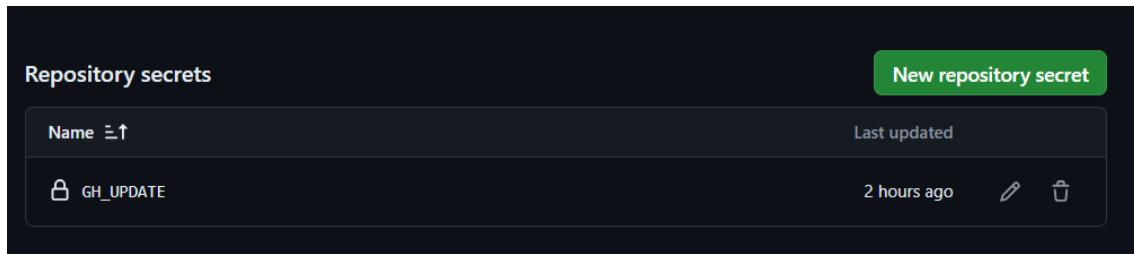
Personal Access Token (PAT) trên GitHub được sử dụng để cấp quyền truy cập cho các ứng dụng hoặc dịch vụ bên thứ ba vào tài khoản GitHub mà không cần sử dụng mật khẩu. Trong hệ thống này, dịch vụ bên thứ ba chính là bot tự động.



Hình 6.1: Tạo PAT Github

2. Tạo Secrets Repository

Mục đích của việc tạo Secrets trên GitHub là nhằm bảo vệ và quản lý các thông tin nhạy cảm như mã token, mật khẩu, khóa API cùng các thông tin quan trọng khác trong quá trình phát triển và triển khai hệ thống. Việc sử dụng Secrets góp phần đảm bảo rằng các thông tin này không bị lộ ra ngoài công cộng, chỉ có thể được truy cập và sử dụng bởi các công cụ tự động hóa như GitHub Actions hoặc các ứng dụng khác thông qua API được cấp phép.



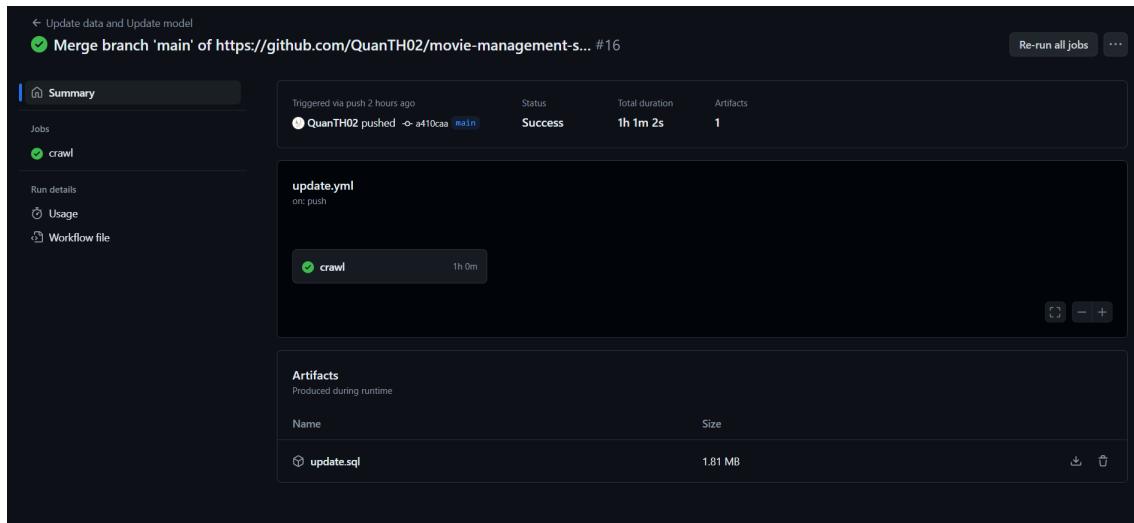
Hình 6.2: Tạo Secrets Repository

3. Viết file update.yml

Bước cuối cùng, tôi tạo file update.yml để GitHub Actions có thể thực thi tự động hàng ngày

Kết quả:

Vào lúc 00h00 mỗi ngày, hệ thống sẽ tự động thực hiện crawl dữ liệu mới, lưu vào file update.sql và đẩy code lên GitHub trong khoảng thời gian 1 giờ.



Hình 6.3: Hệ thống tự động crawl dữ liệu

Sau khi hệ thống tạo ra file update.sql, chỉ cần pull code về máy chủ và thực thi file Crawl/autoUpdate/update.py thì sẽ có dữ liệu mới ngay lập tức.

```

link_img x
1 •   SELECT count(*) FROM moviedata.link_img;

Result Grid | Filter Rows: | Export:
count(*)
▶ 49030

```

Hình 6.4: Cập nhật dữ liệu mới thành công

6.4 Lựa chọn thuật toán gợi ý phù hợp

Việc lựa chọn thuật toán gợi ý phù hợp là một thách thức quan trọng trong quá trình phát triển ứng dụng gợi ý phim. Tôi đã tham khảo các bài báo và tài liệu trên mạng, thực hiện thử nghiệm và đánh giá các thuật toán khác nhau để tìm ra thuật toán phù hợp nhất với yêu cầu của dự án. Cụ thể chính là thuật toán Content-based.

Giới thiệu thuật toán Content-based:

Trong hệ thống content-based này, dựa trên nội dung của mỗi movie, tôi xây dựng một bộ hồ sơ cho mỗi movie. Hồ sơ này được biểu diễn dưới dạng toán học là một feature vector. Features của một bộ phim trong hệ thống gợi ý tôi xây dựng này là:

- Movie name: Tên của bộ phim đó
- Genres: Thể loại của bộ phim đó
- Describe: Mô tả sơ lược về nội dung của bộ phim đó
- Cast: Diễn viên của bộ phim đó
- Director: Đạo diễn của bộ phim đó

Phân tích, giải pháp:

Content-based là thuật toán mà khi chọn một item, thì sẽ đưa ra các item tương

tự có những đặc điểm cụ thể giống với item đã chọn. Vì thế trong bài toán này tôi đã chọn cách tính độ tương đồng giữa các bộ phim dựa trên các feature ở trên.

$$\text{similarity}(A, B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$

Trong đó:

- A và B là hai vector đại diện cho hai phim hoặc hai người dùng.
- $A \cdot B$ là tích vô hướng của hai vector A và B .
- $\|A\|$ và $\|B\|$ lần lượt là độ lớn của vector A và vector B .

Kiểm tra tính chính xác:

Sau khi tính toán được độ tương đồng dựa vào công thức similarity trên, tôi kiểm tra tính chính xác của thuật toán bằng cách: Tính toán độ tương đồng riêng lẻ của các đặc trưng như movie name, genres, cast, director, describe. Sau đó kiểm tra kết quả cuối cùng với các kết quả riêng biệt xem có chính xác không.

```

Movie id: 136 Untitled-1
1 Movie id: 136
2 Movie name: Guardians of the Galaxy
3 Describe: A group of intergalactic criminals must pull together
to stop a fanatical warrior with plans to purge the universe.
4 Genres:
5 Director: James Gunn
6 Cast: Chris Pratt Zoe Saldana Dave Bautista Vin Diesel
Bradley Cooper Lee Pace Michael Rooker Karen Gillan Djimon
Hounsou John C. Reilly Glenn Close Benicio Del Toro Laura
Haddock Sean Gunn Peter Serafinowicz
7 Movie id result:
8 21 0.3526438817427592
9 109 0.21987801134660684
10 48 0.11113121522658766
11 26 0.10923877357827322
12 120 0.08440632301936431
13 19 0.05394264748270165
14 86 0.04964744394508321
15 78 0.047996684986685044
16 32 0.047196291445337255
17 74 0.04686329421881032
18
19 genres
20 45 0.0
21 51 0.0
22 58 0.0
23 49 0.0
24 48 0.0
25 47 0.0
26 46 0.0
27 44 0.0
28 36 0.0
29 43 0.0
30
31 des
32 86 0.13878049563162884
33 63 0.10159744616742258
34 58 0.08755433976577434
35 48 0.0831740893989962
36 80 0.06715573133762848
37 120 0.06481968158583065
38 87 0.06434887142048598
39 43 0.06891447365093486
40 10 0.05759624460829198
41 21 0.05458436118761168
42

cast Untitled-2
1 cast
2 109 0.364705880017843
3 21 0.3387079282444981
4 48 0.1644614390256814
5 120 0.10773013419658836
6 26 0.1055610662972952
7 70 0.10167238581379695
8 32 0.07255420776372298
9 63 0.06604964845434892
10 46 0.06358163683880483
11 40 0.062483862371309436
12
13 director
14 21 1.0
15 26 1.0
16 89 0.4032060238565666
17 46 0.4032060238565666
18 59 0.38328757973316574
19 22 0.38328757973316574
20 50 0.0
21 49 0.0
22 48 0.0
23 47 0.0
24
25 Movie name: Guardians of the Galaxy Vol. 3
26 Describe: Still reeling from the loss of Gamora, Peter
Quill rallies his team to defend the universe and one of
their own - a mission that could mean the end of the
Guardians if not successful.
27 Genres:
28 Director: James Gunn
29 Cast: Chukwudi Iwuji Bradley Cooper Pom Klementieff
Dave Bautista Karen Gillan Vin Diesel Austin Freeman
Stephen Blackehart Terence Rosemore Maria Bakalova Sean
Gunn Sarah Alami Jasmine Munoz Chris Pratt Giovannie
Cruz
30
31 Chris Pratt
32 Bradley Cooper
33 Dave Bautista
34 Karen Gillan
35 Vin Diesel
36 Sean Gunn
37 +
38

```

Hình 6.5: Kiểm tra kết quả 1

Từ hình ảnh 6.5, có thể nhận thấy rằng bộ phim được hệ thống đề xuất có mức

độ tương đồng cao nhất với bộ phim Guardians of the Galaxy chính là bộ phim Guardians of the Galaxy Vol. 3. Hai bộ phim này có sự tương đồng về tên phim, mô tả, diễn viên và đạo diễn.

```

Movie id: 132 Untitled-1 •
1 Movie id: 132
2 Movie name: The Hunger Games: Catching Fire
3 Describe: Katniss Everdeen and Peeta Mellark become
targets of the Capitol after their victory in the 74th
Hunger Games sparks a rebellion in the Districts of Panem.
4 Genres:
5 Director: Francis Lawrence
6 Cast: Jennifer Lawrence Liam Hemsworth Jack Quaid
Taylor St. Clair Sandra Ellis Lafferty Woody Harrelson
Josh Hutcherson Paula Malcomson Willow Shields Donald
Sutherland Elizabeth Banks Bruce Bundy Nelson
Ascencio Lenny Kravitz Stanley Tucci
7 Movie id result:
8 25 0.4886735665102322
9 9 0.12334146342856578
10 106 0.07472692295693872
11 74 0.04422493681947667
12 107 0.04188123767666646
13 94 0.04039795906845274
14 30 0.03660565782846335
15 135 0.031156937908150564
16 140 0.02794239278793543
17 44 0.0266216827827246
18
19 genres
20 45 0.0
21 51 0.0
22 58 0.0
23 49 0.0
24 48 0.0
25 47 0.0
26 46 0.0
27 44 0.0
28 36 0.0
29 43 0.0
30
31 des
32 25 0.3480833324437175
33 9 0.12522903952739983
34 45 0.0
35 51 0.0
36 58 0.0
37 49 0.0
38 48 0.0
39 47 0.0
40 46 0.0
41 45 0.0
42 46 0.0
43 47 0.0
44 48 0.0
45 49 0.0
46 50 0.0
47 51 0.0
48 52 0.0
49 53 0.0
50 54 0.0
51 55 0.0
52 56 0.0
53 57 0.0
54 58 0.0
55 59 0.0
56 60 0.0
57 61 0.0
58 62 0.0
59 63 0.0
60 64 0.0
61 65 0.0
62 66 0.0
63 67 0.0
64 68 0.0
65 69 0.0
66 70 0.0
67 71 0.0
68 72 0.0
69 73 0.0
70 74 0.0
71 75 0.0
72 76 0.0
73 77 0.0
74 78 0.0
75 79 0.0
76 80 0.0
77 81 0.0
78 82 0.0
79 83 0.0
80 84 0.0
81 85 0.0
82 86 0.0
83 87 0.0
84 88 0.0
85 89 0.0
86 90 0.0
87 91 0.0
88 92 0.0
89 93 0.0
90 94 0.0
91 95 0.0
92 96 0.0
93 97 0.0
94 98 0.0
95 99 0.0
96 100 0.0
97 101 0.0
98 102 0.0
99 103 0.0
100 104 0.0
101 105 0.0
102 106 0.0
103 107 0.0
104 108 0.0
105 109 0.0
106 110 0.0
107 111 0.0
108 112 0.0
109 113 0.0
110 114 0.0
111 115 0.0
112 116 0.0
113 117 0.0
114 118 0.0
115 119 0.0
116 120 0.0
117 121 0.0
118 122 0.0
119 123 0.0
120 124 0.0
121 125 0.0
122 126 0.0
123 127 0.0
124 128 0.0
125 129 0.0
126 130 0.0
127 131 0.0
128 132 0.0
129 133 0.0
130 134 0.0
131 135 0.0
132 136 0.0
133 137 0.0
134 138 0.0
135 139 0.0
136 140 0.0
137 141 0.0
138 142 0.0
139 143 0.0
140 144 0.0
141 145 0.0
142 146 0.0
143 147 0.0
144 148 0.0
145 149 0.0
146 150 0.0
147 151 0.0
148 152 0.0
149 153 0.0
150 154 0.0
151 155 0.0
152 156 0.0
153 157 0.0
154 158 0.0
155 159 0.0
156 160 0.0
157 161 0.0
158 162 0.0
159 163 0.0
160 164 0.0
161 165 0.0
162 166 0.0
163 167 0.0
164 168 0.0
165 169 0.0
166 170 0.0
167 171 0.0
168 172 0.0
169 173 0.0
170 174 0.0
171 175 0.0
172 176 0.0
173 177 0.0
174 178 0.0
175 179 0.0
176 180 0.0
177 181 0.0
178 182 0.0
179 183 0.0
180 184 0.0
181 185 0.0
182 186 0.0
183 187 0.0
184 188 0.0
185 189 0.0
186 190 0.0
187 191 0.0
188 192 0.0
189 193 0.0
190 194 0.0
191 195 0.0
192 196 0.0
193 197 0.0
194 198 0.0
195 199 0.0
196 200 0.0
197 201 0.0
198 202 0.0
199 203 0.0
200 204 0.0
201 205 0.0
202 206 0.0
203 207 0.0
204 208 0.0
205 209 0.0
206 210 0.0
207 211 0.0
208 212 0.0
209 213 0.0
210 214 0.0
211 215 0.0
212 216 0.0
213 217 0.0
214 218 0.0
215 219 0.0
216 220 0.0
217 221 0.0
218 222 0.0
219 223 0.0
220 224 0.0
221 225 0.0
222 226 0.0
223 227 0.0
224 228 0.0
225 229 0.0
226 230 0.0
227 231 0.0
228 232 0.0
229 233 0.0
230 234 0.0
231 235 0.0
232 236 0.0
233 237 0.0
234 238 0.0
235 239 0.0
236 240 0.0
237 241 0.0
238 242 0.0
239 243 0.0
240 244 0.0
241 245 0.0
242 246 0.0
243 247 0.0
244 248 0.0
245 249 0.0
246 250 0.0
247 251 0.0
248 252 0.0
249 253 0.0
250 254 0.0
251 255 0.0
252 256 0.0
253 257 0.0
254 258 0.0
255 259 0.0
256 260 0.0
257 261 0.0
258 262 0.0
259 263 0.0
260 264 0.0
261 265 0.0
262 266 0.0
263 267 0.0
264 268 0.0
265 269 0.0
266 270 0.0
267 271 0.0
268 272 0.0
269 273 0.0
270 274 0.0
271 275 0.0
272 276 0.0
273 277 0.0
274 278 0.0
275 279 0.0
276 280 0.0
277 281 0.0
278 282 0.0
279 283 0.0
280 284 0.0
281 285 0.0
282 286 0.0
283 287 0.0
284 288 0.0
285 289 0.0
286 290 0.0
287 291 0.0
288 292 0.0
289 293 0.0
290 294 0.0
291 295 0.0
292 296 0.0
293 297 0.0
294 298 0.0
295 299 0.0
296 300 0.0
297 301 0.0
298 302 0.0
299 303 0.0
300 304 0.0
301 305 0.0
302 306 0.0
303 307 0.0
304 308 0.0
305 309 0.0
306 310 0.0
307 311 0.0
308 312 0.0
309 313 0.0
310 314 0.0
311 315 0.0
312 316 0.0
313 317 0.0
314 318 0.0
315 319 0.0
316 320 0.0
317 321 0.0
318 322 0.0
319 323 0.0
320 324 0.0
321 325 0.0
322 326 0.0
323 327 0.0
324 328 0.0
325 329 0.0
326 330 0.0
327 331 0.0
328 332 0.0
329 333 0.0
330 334 0.0
331 335 0.0
332 336 0.0
333 337 0.0
334 338 0.0
335 339 0.0
336 340 0.0
337 341 0.0
338 342 0.0
339 343 0.0
340 344 0.0
341 345 0.0
342 346 0.0
343 347 0.0
344 348 0.0
345 349 0.0
346 350 0.0
347 351 0.0
348 352 0.0
349 353 0.0
350 354 0.0
351 355 0.0
352 356 0.0
353 357 0.0
354 358 0.0
355 359 0.0
356 360 0.0
357 361 0.0
358 362 0.0
359 363 0.0
360 364 0.0
361 365 0.0
362 366 0.0
363 367 0.0
364 368 0.0
365 369 0.0
366 370 0.0
367 371 0.0
368 372 0.0
369 373 0.0
370 374 0.0
371 375 0.0
372 376 0.0
373 377 0.0
374 378 0.0
375 379 0.0
376 380 0.0
377 381 0.0
378 382 0.0
379 383 0.0
380 384 0.0
381 385 0.0
382 386 0.0
383 387 0.0
384 388 0.0
385 389 0.0
386 390 0.0
387 391 0.0
388 392 0.0
389 393 0.0
390 394 0.0
391 395 0.0
392 396 0.0
393 397 0.0
394 398 0.0
395 399 0.0
396 400 0.0
397 401 0.0
398 402 0.0
399 403 0.0
400 404 0.0
401 405 0.0
402 406 0.0
403 407 0.0
404 408 0.0
405 409 0.0
406 410 0.0
407 411 0.0
408 412 0.0
409 413 0.0
410 414 0.0
411 415 0.0
412 416 0.0
413 417 0.0
414 418 0.0
415 419 0.0
416 420 0.0
417 421 0.0
418 422 0.0
419 423 0.0
420 424 0.0
421 425 0.0
422 426 0.0
423 427 0.0
424 428 0.0
425 429 0.0
426 430 0.0
427 431 0.0
428 432 0.0
429 433 0.0
430 434 0.0
431 435 0.0
432 436 0.0
433 437 0.0
434 438 0.0
435 439 0.0
436 440 0.0
437 441 0.0
438 442 0.0
439 443 0.0
440 444 0.0
441 445 0.0
442 446 0.0
443 447 0.0
444 448 0.0
445 449 0.0
446 450 0.0
447 451 0.0
448 452 0.0
449 453 0.0
450 454 0.0
451 455 0.0
452 456 0.0
453 457 0.0
454 458 0.0
455 459 0.0
456 460 0.0
457 461 0.0
458 462 0.0
459 463 0.0
460 464 0.0
461 465 0.0
462 466 0.0
463 467 0.0
464 468 0.0
465 469 0.0
466 470 0.0
467 471 0.0
468 472 0.0
469 473 0.0
470 474 0.0
471 475 0.0
472 476 0.0
473 477 0.0
474 478 0.0
475 479 0.0
476 480 0.0
477 481 0.0
478 482 0.0
479 483 0.0
480 484 0.0
481 485 0.0
482 486 0.0
483 487 0.0
484 488 0.0
485 489 0.0
486 490 0.0
487 491 0.0
488 492 0.0
489 493 0.0
490 494 0.0
491 495 0.0
492 496 0.0
493 497 0.0
494 498 0.0
495 499 0.0
496 500 0.0
497 501 0.0
498 502 0.0
499 503 0.0
500 504 0.0
501 505 0.0
502 506 0.0
503 507 0.0
504 508 0.0
505 509 0.0
506 510 0.0
507 511 0.0
508 512 0.0
509 513 0.0
510 514 0.0
511 515 0.0
512 516 0.0
513 517 0.0
514 518 0.0
515 519 0.0
516 520 0.0
517 521 0.0
518 522 0.0
519 523 0.0
520 524 0.0
521 525 0.0
522 526 0.0
523 527 0.0
524 528 0.0
525 529 0.0
526 530 0.0
527 531 0.0
528 532 0.0
529 533 0.0
530 534 0.0
531 535 0.0
532 536 0.0
533 537 0.0
534 538 0.0
535 539 0.0
536 540 0.0
537 541 0.0
538 542 0.0
539 543 0.0
540 544 0.0
541 545 0.0
542 546 0.0
543 547 0.0
544 548 0.0
545 549 0.0
546 550 0.0
547 551 0.0
548 552 0.0
549 553 0.0
550 554 0.0
551 555 0.0
552 556 0.0
553 557 0.0
554 558 0.0
555 559 0.0
556 560 0.0
557 561 0.0
558 562 0.0
559 563 0.0
560 564 0.0
561 565 0.0
562 566 0.0
563 567 0.0
564 568 0.0
565 569 0.0
566 570 0.0
567 571 0.0
568 572 0.0
569 573 0.0
570 574 0.0
571 575 0.0
572 576 0.0
573 577 0.0
574 578 0.0
575 579 0.0
576 580 0.0
577 581 0.0
578 582 0.0
579 583 0.0
580 584 0.0
581 585 0.0
582 586 0.0
583 587 0.0
584 588 0.0
585 589 0.0
586 590 0.0
587 591 0.0
588 592 0.0
589 593 0.0
590 594 0.0
591 595 0.0
592 596 0.0
593 597 0.0
594 598 0.0
595 599 0.0
596 600 0.0
597 601 0.0
598 602 0.0
599 603 0.0
600 604 0.0
601 605 0.0
602 606 0.0
603 607 0.0
604 608 0.0
605 609 0.0
606 610 0.0
607 611 0.0
608 612 0.0
609 613 0.0
610 614 0.0
611 615 0.0
612 616 0.0
613 617 0.0
614 618 0.0
615 619 0.0
616 620 0.0
617 621 0.0
618 622 0.0
619 623 0.0
620 624 0.0
621 625 0.0
622 626 0.0
623 627 0.0
624 628 0.0
625 629 0.0
626 630 0.0
627 631 0.0
628 632 0.0
629 633 0.0
630 634 0.0
631 635 0.0
632 636 0.0
633 637 0.0
634 638 0.0
635 639 0.0
636 640 0.0
637 641 0.0
638 642 0.0
639 643 0.0
640 644 0.0
641 645 0.0
642 646 0.0
643 647 0.0
644 648 0.0
645 649 0.0
646 650 0.0
647 651 0.0
648 652 0.0
649 653 0.0
650 654 0.0
651 655 0.0
652 656 0.0
653 657 0.0
654 658 0.0
655 659 0.0
656 660 0.0
657 661 0.0
658 662 0.0
659 663 0.0
660 664 0.0
661 665 0.0
662 666 0.0
663 667 0.0
664 668 0.0
665 669 0.0
666 670 0.0
667 671 0.0
668 672 0.0
669 673 0.0
670 674 0.0
671 675 0.0
672 676 0.0
673 677 0.0
674 678 0.0
675 679 0.0
676 680 0.0
677 681 0.0
678 682 0.0
679 683 0.0
680 684 0.0
681 685 0.0
682 686 0.0
683 687 0.0
684 688 0.0
685 689 0.0
686 690 0.0
687 691 0.0
688 692 0.0
689 693 0.0
690 694 0.0
691 695 0.0
692 696 0.0
693 697 0.0
694 698 0.0
695 699 0.0
696 700 0.0
697 701 0.0
698 702 0.0
699 703 0.0
700 704 0.0
701 705 0.0
702 706 0.0
703 707 0.0
704 708 0.0
705 709 0.0
706 710 0.0
707 711 0.0
708 712 0.0
709 713 0.0
710 714 0.0
711 715 0.0
712 716 0.0
713 717 0.0
714 718 0.0
715 719 0.0
716 720 0.0
717 721 0.0
718 722 0.0
719 723 0.0
720 724 0.0
721 725 0.0
722 726 0.0
723 727 0.0
724 728 0.0
725 729 0.0
726 730 0.0
727 731 0.0
728 732 0.0
729 733 0.0
730 734 0.0
731 735 0.0
732 736 0.0
733 737 0.0
734 738 0.0
735 739 0.0
736 740 0.0
737 741 0.0
738 742 0.0
739 743 0.0
740 744 0.0
741 745 0.0
742 746 0.0
743 747 0.0
744 748 0.0
745 749 0.0
746 750 0.0
747 751 0.0
748 752 0.0
749 753 0.0
750 754 0.0
751 755 0.0
752 756 0.0
753 757 0.0
754 758 0.0
755 759 0.0
756 760 0.0
757 761 0.0
758 762 0.0
759 763 0.0
760 764 0.0
761 765 0.0
762 766 0.0
763 767 0.0
764 768 0.0
765 769 0.0
766 770 0.0
767 771 0.0
768 772 0.0
769 773 0.0
770 774 0.0
771 775 0.0
772 776 0.0
773 777 0.0
774 778 0.0
775 779 0.0
776 780 0.0
777 781 0.0
778 782 0.0
779 783 0.0
780 784 0.0
781 785 0.0
782 786 0.0
783 787 0.0
784 788 0.0
785 789 0.0
786 790 0.0
787 791 0.0
788 792 0.0
789 793 0.0
790 794 0.0
791 795 0.0
792 796 0.0
793 797 0.0
794 798 0.0
795 799 0.0
796 800 0.0
797 801 0.0
798 802 0.0
799 803 0.0
800 804 0.0
801 805 0.0
802 806 0.0
803 807 0.0
804 808 0.0
805 809 0.0
806 810 0.0
807 811 0.0
808 812 0.0
809 813 0.0
810 814 0.0
811 815 0.0
812 816 0.0
813 817 0.0
814 818 0.0
815 819 0.0
816 820 0.0
817 821 0.0
818 822 0.0
819 823 0.0
820 824 0.0
821 825 0.0
822 826 0.0
823 827 0.0
824 828 0.0
825 829 0.0
826 830 0.0
827 831 0.0
828 832 0.0
829 833 0.0
830 834 0.0
831 835 0.0
832 836 0.0
833 837 0.0
834 838 0.0
835 839 0.0
836 840 0.0
837 841 0.0
838 842 0.0
839 843 0.0
840 844 0.0
841 845 0.0
842 846 0.0
843 847 0.0
844 848 0.0
845 849 0.0
846 850 0.0
847 851 0.0
848 852 0.0
849 853 0.0
850 854 0.0
851 855 0.0
852 856 0.0
853 857 0.0
854 858 0.0
855 859 0.0
856 860 0.0
857 861 0.0
858 862 0.0
859 863 0.0
860 864 0.0
861 865 0.0
862 866 0.0
863 867 0.0
864 868 0.0
865 869 0.0
866 870 0.0
867 871 0.0
868 872 0.0
869 873 0.0
870 874 0.0
871 875 0.0
872 876 0.0
873 877 0.0
874 878 0.0
875 879 0.0
876 880 0.0
877 881 0.0
878 882 0.0
879 883 0.0
880 884 0.0
881 885 0.0
882 886 0.0
883 887 0.0
884 888 0.0
885 889 0.0
886 890 0.0
887 891 0.0
888 892 0.0
889 893 0.0
890 894 0.0
891 895 0.0
892 896 0.0
893 897 0.0
894 898 0.0
895 899 0.0
896 900 0.0
897 901 0.0
898 902 0.0
899 903 0.0
900 904 0.0
901 905 0.0
902 906 0.0
903 907 0.0
904 908 0.0
905 909 0.0
906 910 0.0
907 911 0.0
908 912 0.0
909 913 0.0
910 914 0.0
911 915 0.0
912 916 0
```

```

Movie id: 15 Untitled-1 ...
1 Movie id: 15
2 Movie name: Meg 2: The Trench
3 Describe: A research team encounters multiple threats while exploring the depths of the ocean, including a malevolent mining operation.
4 Genres: Action Adventure Horror Sci-Fi Thriller
5 Director: Ben Wheatley
6 Cast: Jason Statham Jing Wu Shuya Sophia Cai Cliff Curtis Page Kennedy Sergio Peris-Mencheta Skyler Samuels Melissanthi Mahut Whoopie Van Raam Kiran Sonia Sawar Felix Mayr Sienna Guillory Tao Guo Robin Hill Lele Dai
7 Movie id result:
8 88 0.5584300169597662
9 38 0.35319267661183157
10 2 0.333988033722691944
11 105 0.3223983287172182
12 9 0.30436133874364807
13 75 0.30063584739093846
14 61 0.2873170128093802
15 140 0.28669542992067987
16 54 0.2861652412081892
17 34 0.28115353309874935
18
19
20 genres
21 88 0.932862414224352
22 38 0.83048650930644602
23 105 0.7549184158987879
24 2 0.7320245466125233
25 22 0.7256822305611071
26 61 0.7256822305611071
27 75 0.7256822305611071
28 140 0.7256822305611071
29 73 0.6431597835862669
30 4 0.6239914216557716
31
32 des
33 90 0.17469314530554264
34 11 0.0990214542750533
35 83 0.07411479933063567
36 88 0.07368250800623025
37 84 0.07172309901182033
38 42 0.06981308044848848
39 102 0.06898968252852719
40 103 0.06898968252852719
41 4 0.05255358144650688
42

cast Untitled-2 ...
1 cast
2 88 0.22183861524280854
3 54 0.05523016212903862
4 18 0.05310982691925622
5 32 0.052348176714640225
6 51 0.046815856484982065
7 46 0.046561172845036136
8 64 0.04160235477938892
9 56 0.04011477030059863
10 99 0.03699776836648799
11 8 0.034249912929519111
12
13 director
14 87 0.459885984332072
15 140 0.0
16 52 0.0
17 51 0.0
18 50 0.0
19 49 0.0
20 48 0.0
21 47 0.0
22 46 0.0
23 45 0.0
24
25 Movie name: The Meg
26 Describe: A group of scientists exploring the Marianas Trench encounter the largest marine predator that has ever existed - the Megalodon.
27 Genres: Action Horror Sci-Fi Thriller
28 Director: Jon Turteltaub
29 Cast: Jason Statham Bingbing Li Rainn Wilson Cliff Curtis Winston Chao Shuya Sophia Cai Ruby Rose Page Kennedy Robert Taylor Olafur Darri Olafsson Jessica McNamee Masi Oka Raymond Vinent Hongmei Mai Wei Yi
30
31
32
33

```

Line 33, Col 1 Spaces: 4 UTF-8 CR/LF Markdown Go Live tabnine starter

Hình 6.7: Kiểm tra kết quả 3

Quan sát hình ảnh 6.7, có thể kết luận rằng bộ phim được hệ thống gợi ý có độ tương đồng lớn nhất với bộ phim Meg 2: The Trench chính là bộ phim The Meg. Hai bộ phim này có nhiều điểm tương đồng về tên gọi, nội dung mô tả, dàn diễn viên và đạo diễn.

Dựa trên các kết quả đã được trình bày ở trên kết hợp với việc kiểm tra bằng phương pháp trực quan, có thể kết luận rằng hệ thống gợi ý phim dựa trên độ tương đồng đã được thực hiện với mức độ chính xác cao.

CHƯƠNG 7. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Chương này được dành để tổng hợp lại toàn bộ quá trình thực hiện và phát triển hệ thống gợi ý phim có tích hợp genAI. Chương sẽ mô tả các giai đoạn và các bước đã được tiến hành để xây dựng sản phẩm từ giai đoạn ý tưởng ban đầu cho đến giai đoạn triển khai và kiểm thử trong môi trường local. Bên cạnh đó, chương còn làm sáng tỏ các mục tiêu và các khó khăn đã phải đối mặt trong quá trình phát triển sản phẩm, cùng với những thành quả đạt được và các bài học kinh nghiệm thu được từ dự án.

Kết quả so sánh

Bảng 7.1: Kết quả so sánh so với các ứng dụng khác

Tính năng	IMDb	Metacritic	Rotten Tomatoes	TMDb	Box Office Mojo	The Numbers	HYF Movie
Thông tin phim	Đầy đủ	Đầy đủ	Đầy đủ	Đầy đủ nhưng phải lấy bằng API	Cơ bản	Thiếu một số	Đầy đủ
Gợi ý phim	Có	Không	Không	Không	Không	Không	Có
Review	Có	Có	Có	Có	Không	Không	Có
Trailer	Có	Có	Có	Có	Không	Không	Có
Hiệu năng	Chậm	Vừa phải	Vừa phải	Ôn	Vừa phải	Vừa phải	Ôn

Từ bảng Table 7.1, có thể nhận thấy rằng hệ thống hiện tại đã giải quyết được các vấn đề ban đầu đã được đề ra.

Phân tích quá trình thực hiện

Xuyên suốt quá trình thực hiện đồ án tốt nghiệp, tôi đã hoàn thành được những mục tiêu cốt lõi như sau:

- Thu thập được toàn bộ dữ liệu cần thiết để phục vụ việc xây dựng hệ thống. Để đạt được mục tiêu này, tôi đã tìm hiểu và nghiên cứu nhiều tài liệu chuyên sâu về chủ đề crawl dữ liệu để có thể áp dụng một cách hiệu quả vào dự án.
- Hoàn thành các chức năng chính của hệ thống. Tôi đã phát triển được nhiều chức năng mà ban đầu tôi nghĩ sẽ khó thực hiện, nhờ đó tôi đã tích lũy được nhiều kiến thức và công nghệ mới để áp dụng vào các dự án tương lai.
- Tiến hành triển khai và kiểm thử hệ thống trong môi trường local.

Những hạn chế còn tồn tại

- Chưa thực hiện triển khai và deploy lên server thực tế. Hiện tại hệ thống chỉ được vận hành trên môi trường local, chưa được kiểm thử với số lượng người dùng lớn trong điều kiện tải cao.
- Giao diện và luồng thao tác của hệ thống còn đơn giản. Nhìn tổng quan, hệ thống có giao diện và các luồng thao tác còn khá đơn giản, chủ yếu tập trung vào các chức năng xem và hiển thị.

Đóng góp nổi bật

- Thiết kế và triển khai hệ thống từ giai đoạn đầu: Tôi đã hoàn thành các chức năng cơ bản và đảm bảo hệ thống vận hành ổn định trong thời gian dài.
- Dữ liệu được đảm bảo về tính chính xác, tính nhất quán, tính đầy đủ và được cập nhật liên tục. Tôi đã thu thập được một bộ dữ liệu hoàn chỉnh phục vụ cho mục đích thử nghiệm và xây dựng hệ thống.
- Phát triển được chức năng gợi ý phim dựa trên lịch sử hoạt động của người dùng cũng như chức năng tìm kiếm phim bằng ngôn ngữ tự nhiên.
- Cấu trúc mã nguồn được thiết kế cho phép mở rộng và phát triển các tính năng nâng cao, tạo điều kiện thuận lợi cho việc kế thừa và phát triển tiếp trong tương lai.

Bài học kinh nghiệm rút ra Trong quá trình thực hiện dự án, tôi đã tích lũy được nhiều bài học quý giá như sau:

- Kỹ năng quản lý thời gian: Việc xây dựng kế hoạch chi tiết và quản lý thời gian một cách hiệu quả góp phần hoàn thành các mục tiêu đúng theo tiến độ đã đề ra. Nhờ hiểu rõ điều này, các buổi báo cáo với giảng viên hướng dẫn đã diễn ra thuận lợi và không gặp phải sự chậm trễ.
- Khả năng tự học và nghiên cứu: Kỹ năng tự học đóng vai trò rất quan trọng, bởi trong môi trường học tập, chúng ta chủ yếu tiếp thu lý thuyết và nhiệm vụ của sinh viên là về nhà tự tìm hiểu, thực hành thêm để củng cố và ghi nhớ kiến thức. Chỉ có việc tự ôn tập và nghiên cứu mới có thể giúp bản thân tiến bộ hơn.
- Tầm quan trọng của việc thu thập thông tin: Tôi nhận ra rằng đây là một bước vô cùng quan trọng trước khi bắt đầu xây dựng một hệ thống. Chúng ta cần phải hiểu rõ nhu cầu của người dùng thì mới có thể xây dựng được một hệ thống thành công.

Hiện tại, các công việc cần thiết để hoàn thiện sản phẩm bao gồm:

- Triển khai hệ thống trên môi trường thực tế. Hệ thống hiện tại đã được tổ chức

thành các gói, các lớp, các thành phần một cách hợp lý và đó là nền tảng để có thể dễ dàng triển khai trong tương lai. Đây là một nhiệm vụ quan trọng để có thể phổ biến hệ thống đến với người dùng.

2. Cải thiện giao diện của hệ thống. Hệ thống hiện tại đang sử dụng các tông màu khá đơn giản như đen, xanh, xanh lá cây. Để hệ thống trở nên thu hút và gây được sự chú ý hơn, cần cải thiện đáng kể về mặt giao diện.
3. Tích hợp thêm các tính năng phân tích và thống kê: Hiện tại hệ thống chỉ có thống kê về số lượng đánh giá. Người dùng thường quan tâm đến các số liệu để quyết định xem phim theo xu hướng số đông, do đó đây là một bước cần thiết trong giai đoạn hiện tại.

Trong tương lai, các hướng phát triển tiếp theo bao gồm:

1. Phát triển phiên bản ứng dụng trên thiết bị di động. Điện thoại di động là thiết bị không thể thiếu trong cuộc sống hiện đại, do đó việc phát triển ứng dụng trên nền tảng di động là một hướng đi quan trọng trong tương lai.
2. Tích hợp hệ thống dự đoán doanh thu phim mới. Đây là một chức năng đã được định hướng từ trước, tuy nhiên do lượng dữ liệu hiện tại trên hệ thống chưa đủ, nên trong tương lai cần thu thập thêm nhiều dữ liệu hơn để có thể phát triển tính năng này.

Đây là phần tổng kết và những định hướng phát triển tiếp theo của đồ án. Tôi sẽ duy trì nỗ lực để cải thiện sản phẩm đạt đến mức tốt nhất. Tôi xin chân thành cảm ơn!

TÀI LIỆU THAM KHẢO

- [1] *Boxofficemojo*, <https://www.boxofficemojo.com/>, Truy cập lần cuối vào ngày 20/12/2025.
- [2] *The numbers*, <https://www.the-numbers.com/>, Truy cập lần cuối vào ngày 20/12/2025.
- [3] *Imdb*, <https://www.imdb.com/>, Truy cập lần cuối vào ngày 20/12/2025.
- [4] *Metacritic*, <https://www.metacritic.com/>, Truy cập lần cuối vào ngày 20/12/2025.
- [5] *Rottentomatoes*, <https://www.rottentomatoes.com/>, Truy cập lần cuối vào ngày 20/12/2025.
- [6] *The movie db*, <https://www.themoviedb.org/>, Truy cập lần cuối vào ngày 20/12/2025.
- [7] *Django mvt*, <https://docs.djangoproject.com/en/6.0/>, Truy cập lần cuối vào ngày 20/12/2025.
- [8] Microsoft, *Visual studio code (vscode)*, <https://code.visualstudio.com/>, Truy cập lần cuối vào ngày 20/12/2025.
- [9] Tom Christie, *Django rest framework*, <https://www.django-rest-framework.org/>, Truy cập lần cuối vào ngày 20/12/2025.
- [10] *Html*, <https://developer.mozilla.org/en-US/docs/Web/HTML>, Truy cập lần cuối vào ngày 20/12/2025.
- [11] *Tailwind css*, <https://tailwindcss.com/>, Truy cập lần cuối vào ngày 20/12/2025.
- [12] *React*, <https://react.dev/>, Truy cập lần cuối vào ngày 20/12/2025.
- [13] Django Software Foundation, *Django*, <https://www.djangoproject.com/>, Truy cập lần cuối vào ngày 20/12/2025.
- [14] *Mysql*, <https://www.mysql.com/>, Truy cập lần cuối vào ngày 20/12/2025.
- [15] *Selenium*, <https://www.selenium.dev/>, Truy cập lần cuối vào ngày 20/12/2025.
- [16] *Requests*, <https://requests.readthedocs.io/>, Truy cập lần cuối vào ngày 20/12/2025.
- [17] *Github actions*, <https://github.com/features/actions>, Truy cập lần cuối vào ngày 20/12/2025.

PHỤ LỤC