

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: Khoa học 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, em 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.

Em 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 em trong quá trình làm đồ án tốt nghiệp.

Em 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 em những kiến thức các môn đại cương cũng như các môn chuyên ngành, giúp em có được cơ sở lý thuyết vững vàng cho em sự tự tin cho chặng đường sắp tới.

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

Cuối cùng, em 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 tìm kiếm phim bằng truy vấn ngôn ngữ người dùng.

Trong đồ án tốt nghiệp này, em lựa chọn hướng tiếp cận là kết hợp tất cả thông tin vào web 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 em 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 em bao gồm việc xây dựng một hệ thống giới thiệu phim với ba loại người dùng: Admin, User và Guest. Hệ thống 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 hệ thống 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 hệ thống giới thiệu phim toàn diện, kết hợp các phương pháp tiên tiến để cải thiện trải nghiệm người dùng. Kết quả đạt được là một hệ thống có khả năng cung cấp nội dung phim một cách đầy đủ, gợi ý phim một cách 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.....	4
2.1 Khảo sát hiện trạng	4
2.2 Tổng quan chức năng	5
2.2.1 Biểu đồ use case tổng quát	5
2.2.2 Biểu đồ use case phân rã Quản lý thông tin người dùng	7
2.2.3 Biểu đồ use case phân rã Quản lý thông tin phim	8
2.2.4 Biểu đồ use case phân rã Quản lý tài khoản	8
2.2.5 Biểu đồ use case phân rã Tương tác với phim.....	9
2.2.6 Biểu đồ use case phân rã Review, comment	9
2.3 Đặc tả chức năng	10
2.3.1 Đặc tả use case Đăng nhập.....	10
2.3.2 Đặc tả use case Chính sửa thông tin cá nhân	11
2.3.3 Đặc tả use case Tìm kiếm phim	12
2.3.4 Đặc tả use case Đánh dấu phim mình thích.....	12
2.3.5 Đặc tả use case Đánh giá phim	12
2.3.6 Đặc tả use case CRUD phim	14
2.3.7 Đặc tả use case Gợi ý phim theo nội dung	15
2.4 Yêu cầu phi chức năng	15

CHƯƠNG 3. CƠ SỞ LÝ THUYẾT CHO GỢI Ý PHIM..... 16

3.1 Content-Based Filtering	16
3.1.1 Kiến thức nền tảng	16
3.1.2 Cơ sở lý thuyết.....	16
3.1.3 Các thuật toán.....	16
3.1.4 Phương pháp nghiên cứu.....	17
3.2 Collaborative Filtering	18
3.2.1 Kiến thức nền tảng	18
3.2.2 Cơ sở lý thuyết.....	18
3.2.3 Các thuật toán.....	18
3.2.4 Phương pháp nghiên cứu.....	19

CHƯƠNG 4. CÔNG NGHỆ SỬ DỤNG..... 21

4.1 Công nghệ sử dụng phía frontend	21
4.1.1 Bootstrap.....	21
4.1.2 JQuery	21
4.2 Công nghệ sử dụng phía backend	21
4.2.1 Django	21
4.2.2 MySQL.....	22
4.3 Công nghệ sử dụng trong việc lưu trữ code	22
4.3.1 Git	22
4.3.2 Github.....	22
4.4 Công nghệ sử dụng trong việc crawl dữ liệu	22
4.4.1 Selenium và Requests	23
4.4.2 Công nghệ sử dụng để update dữ liệu.....	23

CHƯƠNG 5. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG	25
5.1 Thiết kế kiến trúc.....	25
5.1.1 Lựa chọn kiến trúc phần mềm	25
5.1.2 Thiết kế tổng quan.....	26
5.1.3 Thiết kế chi tiết gói	28
5.2 Thiết kế chi tiết.....	31
5.2.1 Thiết kế giao diện	31
5.2.2 Thiết kế cơ sở dữ liệu	37
5.2.3 Thiết kế hệ thống crawl dữ liệu.....	38
5.3 Xây dựng ứng dụng.....	39
5.3.1 Thư viện và công cụ sử dụng	39
5.3.2 Kết quả đạt được	39
5.3.3 Minh họa các chức năng chính	40
5.4 Kiểm thử.....	42
5.5 Triển khai	44
CHƯƠNG 6. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT	46
6.1 Thiếu dữ liệu phim ảnh	46
6.2 Thiết kế database khó khăn với quá nhiều trường	46
6.3 Trang web có dữ liệu update liên tục theo ngày.....	46
6.3.1 Github Action	46
6.4 Lựa chọn thuật toán gợi ý phù hợp.....	49
6.4.1 Content-based.....	49
6.4.2 Collaborative	53
CHƯƠNG 7. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	58
TÀI LIỆU THAM KHẢO.....	61

DANH MỤC HÌNH VẼ

Hình 2.1	Khảo sát người dùng	4
Hình 2.2	Khảo sát các hệ thống đã có	5
Hình 2.3	Use case tổng quan	7
Hình 2.4	Phân rã use case Quản lý thông tin người dùng	7
Hình 2.5	Phân rã use case Quản lý thông tin phim	8
Hình 2.6	Phân rã use case Quản lý tài khoản	8
Hình 2.7	Phân rã use case Tương tác với phim	9
Hình 2.8	Phân rã use case Review, comment	9
Hình 2.9	Đặc tả use case Đăng nhập	10
Hình 2.10	Đặc tả use case Chính sửa thông tin cá nhân	11
Hình 2.11	Đặc tả use case Tìm kiếm phim	12
Hình 2.12	Đặc tả use case Đánh dấu phim mình thích	12
Hình 2.13	Đặc tả use case Đánh giá phim	13
Hình 2.14	Đặc tả use case CRUD phim	14
Hình 2.15	Đặc tả use case Gợi ý phim theo nội dung	15
Hình 5.1	Biểu đồ phụ thuộc gói	26
Hình 5.2	Thiết kế chi tiết gói ở tầng Model	28
Hình 5.3	Thiết kế chi tiết gói ở tầng View	29
Hình 5.4	Thiết kế chi tiết gói ở tầng Template	31
Hình 5.5	Màn hình đăng nhập	32
Hình 5.6	Màn hình đăng ký	33
Hình 5.7	Màn hình profile	33
Hình 5.8	Màn hình chính	34
Hình 5.9	Màn hình gợi ý phim ở trang chủ	34
Hình 5.10	Màn hình thông tin chi tiết 1	35
Hình 5.11	Màn hình thông tin chi tiết 2	35
Hình 5.12	Màn hình tổng hợp đạo diễn, diễn viên,	36
Hình 5.13	Màn hình tổng hợp đạo diễn, diễn viên,	36
Hình 5.14	Màn hình review phim	37
Hình 5.15	Sơ đồ ERD	37
Hình 5.16	Hình ảnh các table trong database	38
Hình 5.17	Hình ảnh các table trong database	38
Hình 5.18	Hình ảnh các table trong database	38
Hình 5.19	Tổng hợp các thông tin cần lấy	39

Hình 5.20	Chức năng gợi ý phim dựa vào sở thích người dùng	40
Hình 5.21	Chức năng gợi ý phim dựa vào nội dung phim đang xem	41
Hình 5.22	Chức năng thêm phim vào danh sách yêu thích	41
Hình 5.23	Chức năng review phim	42
Hình 5.24	Chức năng xem toàn bộ ảnh của phim	42
Hình 5.25	Kiểm thử chức năng 1	43
Hình 5.26	Kiểm thử chức năng 2	43
Hình 6.1	Tạo PAT Github	47
Hình 6.2	Tạo Secrets Repository	48
Hình 6.3	Hệ thống tự động crawl dữ liệu	48
Hình 6.4	Cập nhật dữ liệu mới thành công	49
Hình 6.5	Kiểm tra kết quả 1	51
Hình 6.6	Kiểm tra kết quả 2	52
Hình 6.7	Kiểm tra kết quả 3	53
Hình 6.8	Tính giá trị trung bình rating của các user	54
Hình 6.9	Tính lại rating theo giá trị trung bình	55
Hình 6.10	Tính độ tương đồng giữa các user theo rating mới	55
Hình 6.11	Dự đoán rating cho từng item	56
Hình 6.12	Kết quả dự đoán rating cuối cùng của hệ thống	56
Hình 6.13	Giá trị RMSE	56
Hình 6.14	Kết quả gợi ý item cho user	57
Hình 7.1	Kết quả so sánh so với các ứng dụng khác	58

DANH MỤC BẢNG BIỂU

Bảng 5.1 Danh sách thư viện và công cụ sử dụng 39

DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT 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)
CRUD	Create, Read, Update, Delete
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)
RMSE	Độ lệch bình phương trung bình gốc (Root Mean Square Error)

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 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, the-numbers cung cấp thông tin cơ bản như thể loại phim nhưng hạn chế hơn so với boxofficemojo. IMDb 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 và rottentomatoes 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 cung cấp thông tin phim dưới dạng API dành cho nhà phát triển, không phải là một trang web dành cho người dùng cuối để tìm kiếm và gợi ý phim.

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 thông tin phong phú, nhanh chóng và dễ dàng sử dụng, giúp người dùng tìm kiếm và khám phá phim phù hợp với sở thích cá nhân một cách hiệu quả nhất.

1.3 Định hướng giải pháp

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

- **Dữ liệu không 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 kém:** 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.
- **Thiếu 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.
- **Gợi ý phim cá nhân hóa:** Theo dõi hành động người dùng để gợi ý phim cá nhân hóa.
- **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.
- **Chức năng bị phân tán:** 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 em sẽ tổ chức như sau:

Chương 2 em sẽ trình bày về phần khảo sát và phân tích yêu cầu. Trong chương này, em 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, em 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 em 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). Em sẽ giải thích cách thuật toán này hoạt động và lý do chọn thuật toán này cho hệ thống của mình.

Chương 4 em sẽ đi sâu vào các công nghệ sử dụng trong quá trình phát triển hệ thống. Em 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 em 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. Em 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 em sẽ nêu ra những giải pháp và đóng góp nổi bật trong quá trình làm sản phẩm này. Đồng thời phân tích bài toán gợi ý phim một cách đầy đủ nhất.

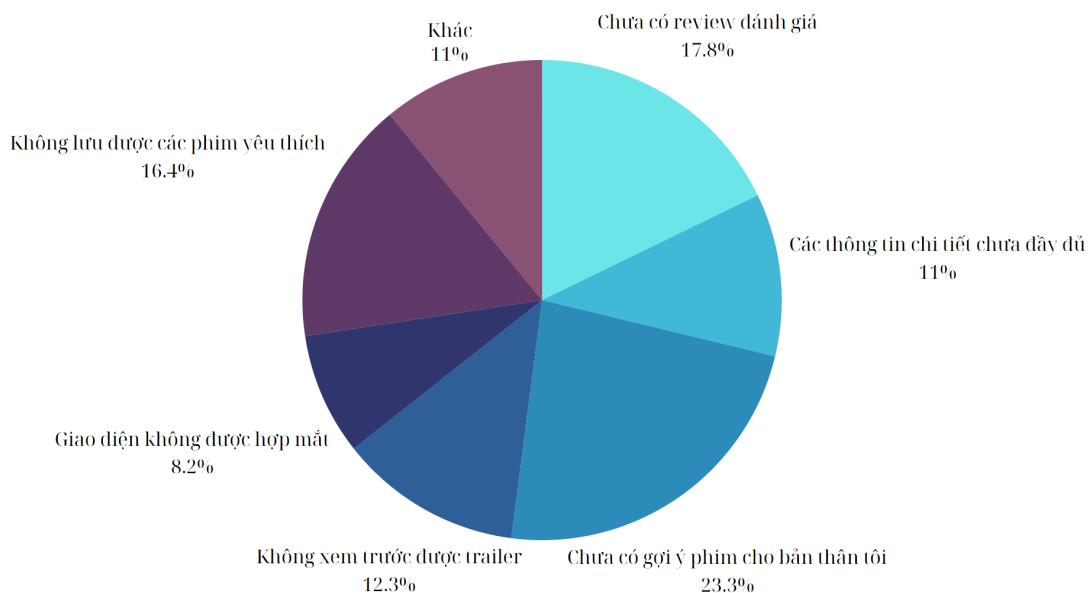
Chương 7 em sẽ kết luận và đề xuất hướng phát triển. Em sẽ tóm tắt lại những kết quả chính đã đạt được trong đồ án, đồng thời đề xuất các hướng phát triển tiềm năng cho hệ thống trong tương lai. Chương này cung cấp cái nhìn tổng quát về

đóng góp của đồ án và các bước tiếp theo có thể thực hiện để nâng cao hệ thống.

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

Trong chương này, em sẽ tiến hành khảo sát và phân tích yêu cầu cho hệ thống giới thiệu phim. Chương này sẽ bắt đầu bằng việc khảo sát hiện trạng của các hệ thống giới thiệu phim hiện có, nhằm hiểu rõ những ưu điểm và hạn chế của chúng. Tiếp theo, em sẽ trình bày tổng quan các chức năng cần thiết cho hệ thống mới thông qua biểu đồ use case tổng quan, phân rã các chức năng và quy trình nghiệp vụ. Sau đó, các chức năng sẽ được đặc tả chi tiết để đảm bảo rằng tất cả các yêu cầu đều được xác định một cách rõ ràng và chính xác. Cuối cùng, em sẽ đề cập đến các yêu cầu phi chức năng, đặc biệt là chức năng gợi ý phim, để đảm bảo rằng hệ thống không chỉ đáp ứng được các yêu cầu cơ bản mà còn mang lại trải nghiệm người dùng tốt nhất.

2.1 Khảo sát hiện trạng



Hình 2.1: Khảo sát người dùng

Trong quá trình khảo sát hiện trạng và yêu cầu của hệ thống giới thiệu phim, em đã thu thập thông tin từ ba nguồn chính là người dùng/khách hàng, các hệ thống đã có và các ứng dụng tương tự. Từ những phản hồi và dữ liệu thu thập được, em đã thực hiện phân tích, so sánh và đánh giá chi tiết về ưu nhược điểm của các sản phẩm và nghiên cứu hiện có.

Từ khảo sát người dùng/khách hàng, em nhận thấy rằng người dùng mong muốn một hệ thống có khả năng tìm kiếm phim nhanh chóng và chính xác, cùng với các tính năng gợi ý phim dựa trên sở thích cá nhân và hành vi xem phim trước đó.

Các tính năng như đánh giá, nhận xét phim và xem trailer cũng được đánh giá cao. Đồng thời, hiệu năng hệ thống cần được cải thiện, và giao diện người dùng phải thân thiện và dễ sử dụng.

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	Thiếu một 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 2.2: Khảo sát các hệ thống đã có

Trong khi đó, thông qua khảo sát các hệ thống đã có như IMDb, Metacritic, Rotten Tomatoes, TMDb, Box Office Mojo và The Numbers, em đã so sánh và đánh giá các tính năng, hiệu năng và khả năng cung cấp thông tin của từng hệ thống. Từ đó, em nhận thấy mỗi hệ thống có điểm mạnh và điểm yếu riêng, và việc kết hợp các tính năng của chúng có thể tạo ra một hệ thống giới thiệu phim toàn diện.

Cuối cùng, em cũng đã khảo sát một số ứng dụng giới thiệu phim trên nền tảng di động và web, từ đó nhận thấy một số điểm chung về giao diện thân thiện và tính năng gợi ý phim dựa trên hành vi người dùng như trong bảng 2.2. Tuy nhiên, nhiều ứng dụng vẫn cần cải thiện về mặt hiệu năng và tính năng phong phú.

Dựa trên kết quả khảo sát từ các nguồn trên, em xác định rõ các tính năng cần thiết để phát triển một hệ thống giới thiệu phim toàn diện, đáp ứng tốt nhu cầu của người dùng.

2.2 Tổng quan chức năng

2.2.1 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 Mối Tác Nhân:

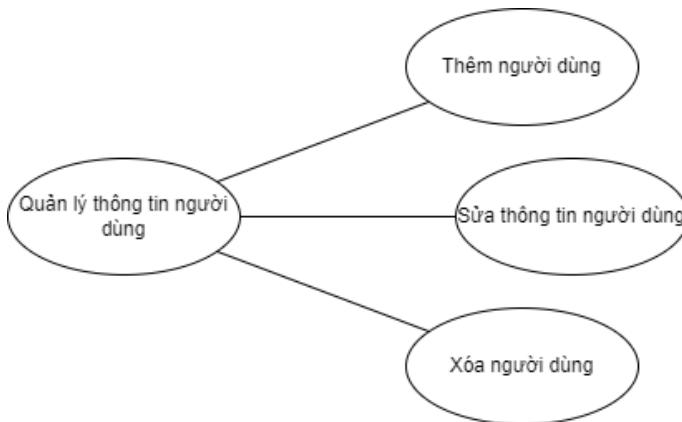
- Guest: Tác nhân này có vai trò chính là tìm kiếm thông tin về phim và xem các bộ phim mà không cần đăng nhập vào hệ thống.
- User: Tác nhân này có vai trò tương tự như Guest, nhưng cũng có thêm quyền đánh giá phim, thêm phim vào danh sách yêu thích, và quản lý thông tin cá nhân.
- Admin: Tác nhân này 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 2.3:** Use case tổng quan

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

**Hình 2.4:** Phân rã use case Quản lý thông tin người dùng

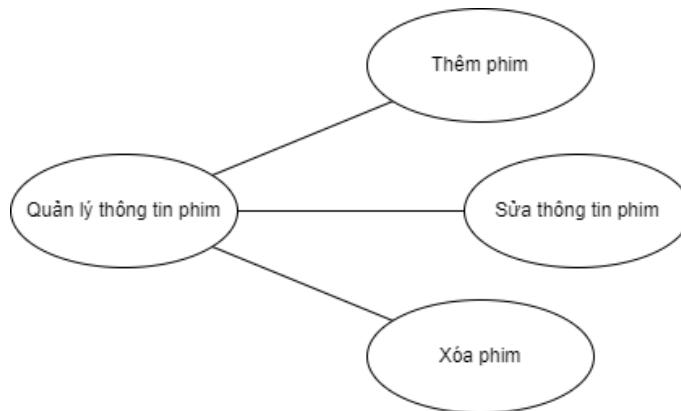
Trong biểu đồ use case phân rã "Quản lý thông tin người dùng", các use case được phân rã và mô tả như sau:

- 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.

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

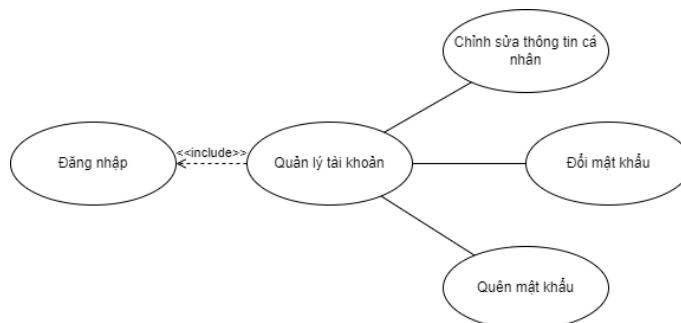


Hình 2.5: Phân rã use case Quản lý thông tin phim

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.

2.2.4 Biểu đồ use case phân rã Quản lý tài khoản

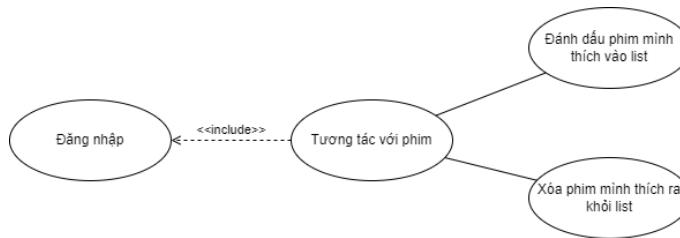


Hình 2.6: Phân rã use case Quản lý tài khoản

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, ...

2.2.5 Biểu đồ use case phân rã Tương tác với phim



Hình 2.7: Phân rã use case Tương tác với phim

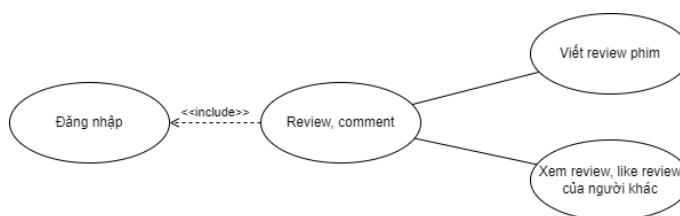
Trong biểu đồ use case phân rã "Tương tác với phim", các use case được phân rã và mô tả như sau:

- 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.

2.2.6 Biểu đồ use case phân rã Review, comment



Hình 2.8: 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.

2.3 Đặc tả chức năng

2.3.1 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.	
	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.	
Hậu điều kiện				

Hình 2.9: Đặc tả use case Đăng nhập

2.3.2 ĐẶC TẢ USE CASE CHỈNH SỬA THÔNG TIN CÁ NHÂN

Mã Use case	UC009		Tên Use case	Chỉnh sửa thông tin cá nhân
Tác nhân	Guest, Admin			
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				

Hình 2.10: Đặc tả use case Chỉnh sửa thông tin cá nhân

2.3.3 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			

Hình 2.11: Đặc tả use case Tìm kiếm phim

2.3.4 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	
	2.	Hệ thống	Hệ thống cập nhật list yêu thích của người dùng trong database và hiển thị thông báo đã thêm thành công	
	STT	Thực hiện bởi	Hành động	
Hậu điều kiện				

Hình 2.12: Đặc tả use case Đánh dấu phim mình thích

2.3.5 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		

Hình 2.13: ĐẶC TẢ USE CASE ĐÁNH GIÁ PHIM

2.3.6 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		

Hình 2.14: Đặc tả use case CRUD phim

2.3.7 ĐẶC TẢ USE CASE GỌI Ý PHIM THEO NỘI DUNG

Mã Use case	UC017		Tên Use case	Gọi 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			

Hình 2.15: Đặc tả use case Gọi ý phim theo nội dung

2.4 Yêu cầu phi chức năng

Trong phần này, em xác định được các yêu cầu phi chức năng và kỹ thuật cho hệ thống như sau:

Hiệu năng: Hệ thống cần đảm bảo hiệu suất cao và thời gian phản hồi nhanh chóng, đặc biệt khi xử lý các yêu cầu tìm kiếm, sắp xếp và gợi ý phim. Thời gian tải trang cần được giảm thiểu để tăng trải nghiệm người dùng.

Độ tin cậy: Hệ thống phải ổn định và đáng tin cậy, đảm bảo rằng dữ liệu không bị mất mát, các chức năng hoạt động một cách mượt mà mọi lúc, đặc biệt là phải chính xác.

Tính dễ dùng: Giao diện người dùng cần được thiết kế sao cho dễ sử dụng và trực quan, giúp người dùng dễ dàng tìm kiếm thông tin và tương tác với hệ thống một cách tự nhiên.

Tính dễ bảo trì: Hệ thống cần có cấu trúc code rõ ràng và dễ bảo trì để cho phép việc nâng cấp, sửa lỗi và mở rộng dễ dàng trong tương lai.

Cơ sở dữ liệu: Cần sử dụng cơ sở dữ liệu hiệu quả và linh hoạt để lưu trữ và quản lý thông tin phim và người dùng một cách hiệu quả. Hệ thống cần hỗ trợ các thao tác truy vấn nhanh chóng và bảo mật dữ liệu.

CHƯƠNG 3. CƠ SỞ LÝ THUYẾT CHO GỢI Ý PHIM

Chương này trình bày nền tảng lý thuyết và các thuật toán được sử dụng để xây dựng hệ thống gợi ý phim, bao gồm hai phương pháp chính là Content-Based Filtering và Collaborative Filtering. Đây là các phương pháp phổ biến và hiệu quả trong việc đề xuất nội dung cho người dùng dựa trên dữ liệu hiện có. Nội dung chương sẽ đi sâu vào các kiến thức nền tảng, cơ sở lý thuyết, các thuật toán cụ thể, và phương pháp nghiên cứu của từng phương pháp gợi ý. Mỗi phần sẽ phân tích rõ cách thức áp dụng từng thuật toán để giải quyết các yêu cầu và vấn đề đã được xác định trong Chương 2. Qua đó, sẽ có cái nhìn tổng quan và chi tiết về cách thức hoạt động cũng như lợi ích của từng phương pháp gợi ý trong hệ thống.

3.1 Content-Based Filtering

3.1.1 Kiến thức nền tảng

Content-Based Filtering (CBF) là một kỹ thuật gợi ý trong hệ thống đề xuất, dựa trên các đặc trưng của các đối tượng cần đề xuất. Thay vì dựa vào hành vi và sở thích của người dùng khác, CBF sử dụng các thuộc tính và đặc điểm cụ thể của sản phẩm để đưa ra gợi ý. Điều này giúp hệ thống có thể gợi ý các sản phẩm tương tự dựa trên những sản phẩm mà người dùng đã quan tâm hoặc đánh giá cao. Ví dụ, nếu một người dùng thích một bộ phim có chủ đề khoa học viễn tưởng và có sự tham gia của một diễn viên cụ thể, hệ thống sẽ gợi ý các bộ phim khác có chủ đề tương tự và có sự tham gia của diễn viên đó.

3.1.2 Cơ sở lý thuyết

Cơ sở lý thuyết của Content-Based Filtering dựa trên việc phân tích và mô tả các đặc trưng của sản phẩm. Mỗi sản phẩm được mô tả bằng một tập hợp các thuộc tính đặc trưng, ví dụ như thể loại, diễn viên, đạo diễn, hoặc từ khóa mô tả nội dung phim. Các đặc trưng này được sử dụng để tạo ra một hồ sơ (profile) cho mỗi sản phẩm. Hồ sơ của người dùng được xây dựng dựa trên các đặc trưng của những sản phẩm mà người dùng đã thích hoặc đánh giá cao. Hệ thống sử dụng các hồ sơ này để tìm kiếm và gợi ý các sản phẩm tương tự. Một số kỹ thuật thường được sử dụng trong CBF bao gồm vector hóa từ khóa (TF-IDF), phân tích tương quan Cosine, và các phương pháp học máy như K-Nearest Neighbors (KNN) để tính toán độ tương tự giữa các hồ sơ.

3.1.3 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 sử dụng để đánh giá tầm quan trọng của từ khóa trong văn bản. Tần suất xuất hiện của từ trong văn bản (Term Frequency) và mức độ phổ biến của từ trong toàn bộ tập văn bản (Inverse Document Frequency) được kết hợp để xác định trọng số của từ khóa.

Cosine Similarity: Đây là một phương pháp đo độ tương tự giữa hai vector trong không gian vector. Độ tương tự cosine này được xác định bằng cách tính giá trị cosin của góc giữa hai vector. Phương pháp này thường được sử dụng để đo độ tương tự giữa hồ sơ của người dùng và hồ sơ của sản phẩm.

K-Nearest Neighbors (KNN): Đây là một thuật toán học máy dùng để phân loại và hồi quy. Trong CBF, KNN được sử dụng để tìm ra k sản phẩm tương tự nhất với sản phẩm mà người dùng đã đánh giá cao.

Naive Bayes: Đây là một thuật toán phân loại dựa trên lý thuyết xác suất Bayes. Naive Bayes có thể được sử dụng để dự đoán xác suất một sản phẩm thuộc một loại cụ thể dựa trên các đặc trưng của nó.

3.1.4 Phương pháp nghiên cứu

Để triển khai Content-Based Filtering, cần thực hiện các bước sau:

Thu thập dữ liệu: Dữ liệu về các đặc trưng của sản phẩm cần được thu thập từ các nguồn uy tín như IMDb và BoxOfficeMojo. Dữ liệu này bao gồm các thuộc tính của phim như thể loại, diễn viên, đạo diễn, và các từ khóa mô tả nội dung.

Xây dựng hồ sơ sản phẩm và người dùng: Sử dụng các đặc trưng đã thu thập để xây dựng hồ sơ chi tiết cho mỗi sản phẩm. Đồng thời, xây dựng hồ sơ người dùng dựa trên các sản phẩm mà họ đã đánh giá cao hoặc yêu thích.

Tính toán độ tương tự: Sử dụng các thuật toán như TF-IDF và Cosine Similarity để tính toán độ tương tự giữa hồ sơ của người dùng và hồ sơ của các sản phẩm.

Gợi ý sản phẩm: Dựa trên độ tương tự đã tính toán, gợi ý các sản phẩm tương tự mà người dùng có thể quan tâm.

Đánh giá và cải thiện: Đánh giá hiệu quả của hệ thống gợi ý thông qua các chỉ số như độ chính xác, độ bao phủ, và độ tin cậy. Dựa trên kết quả đánh giá, điều chỉnh các tham số của thuật toán và cập nhật dữ liệu để cải thiện hệ thống.

Việc áp dụng Content-Based Filtering không chỉ giúp hệ thống gợi ý phim hoạt động hiệu quả mà còn đảm bảo rằng người dùng nhận được các đề xuất phù hợp với sở thích và nhu cầu cá nhân.

3.2 Collaborative Filtering

3.2.1 Kiến thức nền tảng

Collaborative Filtering (CF) là một kỹ thuật gợi ý dựa trên sự hợp tác giữa các người dùng trong hệ thống. Phương pháp này không chỉ dựa trên đặc trưng của các sản phẩm mà còn dựa trên hành vi và sở thích của người dùng. Cụ thể, CF sử dụng thông tin từ các người dùng khác có sở thích tương tự để đưa ra gợi ý. Đây là một trong những phương pháp gợi ý phổ biến nhất và được áp dụng rộng rãi trong các hệ thống khuyến nghị như Amazon, Netflix, và Spotify.

3.2.2 Cơ sở lý thuyết

Phương pháp Collaborative Filtering (CF) là một trong những phương pháp quan trọng trong lọc cộng tác trong lĩnh vực học máy và khai phá dữ liệu. Cơ sở lý thuyết của CF dựa trên hai giả định chính là User similarity và Item similarity. CF được triển khai thông qua hai phương pháp chính:

User-Based Collaborative Filtering tìm kiếm những người dùng có sở thích tương tự với người dùng hiện tại và gợi ý những sản phẩm mà họ đã thích nhưng người dùng hiện tại chưa khám phá.

Ngược lại, Item-Based Collaborative Filtering tìm kiếm các sản phẩm tương tự dựa trên các sản phẩm mà người dùng đã đánh giá cao và gợi ý các sản phẩm tương tự này cho người dùng.

3.2.3 Các thuật toán

Các thuật toán chính được sử dụng trong Collaborative Filtering bao gồm:

User-Based Collaborative Filtering (UBCF): Thuật toán này tính toán độ tương tự giữa các người dùng bằng cách so sánh lịch sử đánh giá của họ. Một phương pháp phổ biến để đo độ tương tự là sử dụng Cosine Similarity hoặc Pearson Correlation. Sau khi tìm ra những người dùng có độ tương tự cao, hệ thống sẽ gợi ý các sản phẩm mà họ đã thích nhưng người dùng hiện tại chưa đánh giá.

Item-Based Collaborative Filtering (IBCF): Thay vì so sánh giữa các người dùng, IBCF so sánh giữa các sản phẩm. Hệ thống sẽ tìm ra các sản phẩm tương tự dựa trên lịch sử đánh giá của tất cả người dùng. Các phương pháp như Cosine Similarity hoặc Jaccard Similarity có thể được sử dụng để đo độ tương tự giữa các sản phẩm.

Matrix Factorization: Đây là một kỹ thuật học máy nâng cao được sử dụng rộng rãi trong CF. Matrix Factorization, như SVD (Singular Value Decomposition), phân rã ma trận đánh giá người dùng-sản phẩm thành các ma trận nhỏ hơn, từ đó phát hiện ra các yếu tố ẩn định nghĩa sự tương đồng giữa người dùng và sản phẩm.

k-Nearest Neighbors (k-NN): Thuật toán k-NN có thể được áp dụng cả trong UBCF và IBCF. Trong UBCF, k-NN tìm ra k người dùng tương tự nhất, trong khi ở IBCF, nó tìm ra k sản phẩm tương tự nhất.

3.2.4 Phương pháp nghiên cứu

Để triển khai Collaborative Filtering, cần thực hiện các bước sau:

Thu thập dữ liệu: Thu thập dữ liệu đánh giá từ người dùng về các sản phẩm. Dữ liệu này có thể bao gồm các đánh giá sao, bình luận, và lịch sử mua hàng.

Xây dựng ma trận người dùng-sản phẩm: Tạo ra một ma trận mà mỗi hàng đại diện cho một người dùng và mỗi cột đại diện cho một sản phẩm. Các giá trị trong ma trận là các đánh giá mà người dùng đã cho sản phẩm đó.

Tính toán độ tương tự: Sử dụng các phương pháp như Cosine Similarity hoặc Pearson Correlation để tính toán độ tương tự giữa các người dùng hoặc các sản phẩm.

Gợi ý sản phẩm: Dựa trên độ tương tự đã tính toán, hệ thống gợi ý các sản phẩm mà người dùng tương tự đã đánh giá cao hoặc các sản phẩm tương tự với những gì người dùng đã thích.

Đánh giá và cải thiện: Đánh giá hiệu quả của hệ thống gợi ý bằng các chỉ số như RMSE (Root Mean Squared Error), MAE (Mean Absolute Error), hoặc Precision and Recall. Điều chỉnh các tham số và mô hình dựa trên kết quả đánh giá để cải thiện độ chính xác của hệ thống.

Collaborative Filtering là một phương pháp mạnh mẽ trong hệ thống gợi ý phim, giúp cá nhân hóa trải nghiệm người dùng dựa trên sự hợp tác và chia sẻ thông tin giữa các người dùng. Việc áp dụng đúng các kỹ thuật và thuật toán sẽ giúp cải thiện độ chính xác và hiệu quả của các đề xuất, từ đó tăng cường sự hài lòng của người dùng.

Chương này đã cung cấp một cái nhìn tổng quan và chi tiết về hai phương pháp gợi ý phim chủ đạo: Content-Based Filtering và Collaborative Filtering. Qua các phần trình bày về kiến thức nền tảng, cơ sở lý thuyết, các thuật toán và phương pháp nghiên cứu, chúng ta đã hiểu rõ cách thức hoạt động, ưu điểm và hạn chế của từng phương pháp.

Content-Based Filtering dựa vào phân tích các đặc trưng của nội dung phim để gợi ý các bộ phim tương tự mà người dùng có thể quan tâm. Phương pháp này giúp hệ thống đưa ra các gợi ý chính xác dựa trên thông tin cụ thể của phim và sở thích cá nhân của người dùng.

Collaborative Filtering dựa trên sự hợp tác và chia sẻ thông tin giữa các người dùng để gợi ý phim. Bằng cách phân tích hành vi và sở thích của nhiều người dùng, phương pháp này có thể đưa ra các gợi ý cá nhân hóa và phù hợp hơn, đặc biệt khi người dùng có xu hướng thích các phim mà những người dùng tương tự đã đánh giá cao.

Việc hiểu và áp dụng đúng các kỹ thuật của cả hai phương pháp này sẽ giúp hệ thống gợi ý phim hoạt động hiệu quả, mang lại trải nghiệm tốt hơn cho người dùng. Các kiến thức và phương pháp trình bày trong chương này sẽ là nền tảng vững chắc để phát triển và tối ưu hóa hệ thống gợi ý phim trong các chương tiếp theo.

CHƯƠNG 4. CÔNG NGHỆ SỬ DỤNG

4.1 Công nghệ sử dụng phía frontend

Trong chương này, em sẽ giới thiệu các công nghệ và nền tảng đã sử dụng trong quá trình phát triển hệ thống giới thiệu phim ảnh. Những công nghệ này đóng vai trò quan trọng trong việc xây dựng, triển khai và tối ưu hóa hệ thống, đảm bảo tính khả thi và hiệu quả của giải pháp.

4.1.1 Bootstrap

Bootstrap là một framework CSS phổ biến, giúp thiết kế giao diện người dùng một cách nhanh chóng và dễ dàng. Với Bootstrap, em có thể tạo ra các giao diện web đáp ứng, đẹp mắt và tương thích với nhiều thiết bị khác nhau, từ máy tính bàn đến điện thoại di động. Việc sử dụng Bootstrap giúp tiết kiệm thời gian thiết kế và đảm bảo tính nhất quán trong giao diện người dùng của hệ thống.

Bootstrap cung cấp một hệ thống lưới linh hoạt và nhiều thành phần UI có sẵn như nút, biểu mẫu, bảng, và modal, giúp xây dựng giao diện nhanh chóng và hiệu quả. Ngoài ra, Bootstrap còn hỗ trợ nhiều tiện ích JavaScript giúp tăng cường tính tương tác của trang web.

4.1.2 JQuery

JQuery là một thư viện JavaScript nhỏ gọn nhưng mạnh mẽ, hỗ trợ thao tác DOM, xử lý sự kiện và Ajax. Với JQuery, em có thể dễ dàng thêm các tính năng động và tương tác cho trang web, cải thiện trải nghiệm người dùng. Việc kết hợp Bootstrap và JQuery giúp giao diện người dùng trở nên linh hoạt, trực quan và thân thiện hơn.

JQuery giúp đơn giản hóa các tác vụ phức tạp trong JavaScript, như thao tác DOM và xử lý sự kiện, bằng cách cung cấp một API dễ sử dụng. JQuery cũng hỗ trợ Ajax, giúp trang web có thể tải dữ liệu động mà không cần tải lại trang, cải thiện hiệu suất và trải nghiệm người dùng.

4.2 Công nghệ sử dụng phía backend

4.2.1 Django

Django là một framework web phát triển bằng Python, nổi tiếng với khả năng giúp xây dựng các ứng dụng web một cách nhanh chóng và hiệu quả. Lý do chọn Django cho hệ thống gợi ý phim là vì nó cung cấp một kiến trúc MVT (Model-View-Template) rõ ràng, giúp tổ chức mã nguồn một cách khoa học và dễ dàng bảo trì. Django cũng hỗ trợ tích hợp với nhiều cơ sở dữ liệu khác nhau, trong đó MySQL được chọn làm hệ quản trị cơ sở dữ liệu cho hệ thống.

Django cung cấp nhiều tính năng mạnh mẽ như ORM (Object-Relational Mapping), giúp dễ dàng thao tác với cơ sở dữ liệu mà không cần viết SQL thủ công. Ngoài ra, Django còn hỗ trợ nhiều tính năng bảo mật tích hợp sẵn như chống tấn công CSRF, XSS, và SQL Injection, giúp đảm bảo an toàn cho hệ thống.

4.2.2 MySQL

MySQL là một hệ quản trị cơ sở dữ liệu mã nguồn mở, được sử dụng rộng rãi trong các ứng dụng web vì tính ổn định và hiệu suất cao. MySQL hỗ trợ các tính năng mạnh mẽ như quản lý dữ liệu phức tạp, thực hiện truy vấn nhanh chóng và bảo mật cao, phù hợp với yêu cầu lưu trữ và truy xuất dữ liệu phim ảnh của hệ thống.

Sử dụng MySQL trong hệ thống giúp đảm bảo dữ liệu được tổ chức một cách có cấu trúc, dễ dàng truy xuất và quản lý. MySQL cũng hỗ trợ tính năng sao lưu và phục hồi dữ liệu, giúp bảo vệ dữ liệu quan trọng của hệ thống.

4.3 Công nghệ sử dụng trong việc lưu trữ code

4.3.1 Git

Git là một hệ thống quản lý phiên bản phân tán, cho phép theo dõi sự thay đổi của mã nguồn trong quá trình phát triển. Việc sử dụng Git giúp em quản lý mã nguồn một cách hiệu quả, dễ dàng quay lại các phiên bản trước và phối hợp làm việc nhóm một cách mượt mà. Git cung cấp các tính năng mạnh mẽ như nhánh (branching), hợp nhất (merging) và theo dõi lịch sử (history), giúp quản lý quá trình phát triển và xử lý xung đột mã nguồn một cách hiệu quả.

4.3.2 Github

GitHub là một dịch vụ lưu trữ mã nguồn dựa trên Git, cung cấp nền tảng trực tuyến để chia sẻ, lưu trữ và quản lý mã nguồn. Sử dụng GitHub, em có thể dễ dàng chia sẻ mã nguồn với các thành viên khác trong nhóm, theo dõi các vấn đề, yêu cầu tính năng và phối hợp phát triển một cách hiệu quả. GitHub cũng cung cấp nhiều công cụ hỗ trợ như review mã, quản lý dự án và tích hợp liên tục (CI/CD), giúp quy trình phát triển trở nên chuyên nghiệp và hiệu quả hơn.

GitHub cung cấp một giao diện web thân thiện, giúp dễ dàng quản lý các kho lưu trữ và theo dõi tiến độ dự án. Ngoài ra, GitHub cũng hỗ trợ tích hợp với nhiều dịch vụ và công cụ khác, giúp tự động hóa các quy trình phát triển và triển khai.

4.4 Công nghệ sử dụng trong việc crawl dữ liệu

Trong quá trình xây dựng hệ thống gợi ý phim, việc thu thập và cập nhật dữ liệu từ các nguồn uy tín là một phần quan trọng nhằm đảm bảo tính chính xác và cập nhật của thông tin phim ảnh. Để thực hiện việc này, em đã sử dụng các công cụ

mạnh mẽ như Selenium và Requests, kết hợp với việc tự động hóa quy trình cập nhật dữ liệu bằng GitHub Actions.

4.4.1 Selenium và Requests

Để có được dữ liệu phong phú và chính xác cho hệ thống gợi ý phim, em đã sử dụng kỹ thuật crawl dữ liệu từ các trang web uy tín như IMDb và BoxOfficeMojo. IMDb là một cơ sở dữ liệu trực tuyến về phim ảnh, chương trình truyền hình và người nổi tiếng, cung cấp thông tin chi tiết và đa dạng về các bộ phim. BoxOffice-Mojo là một trang web chuyên về doanh thu phòng vé, cung cấp dữ liệu chính xác về doanh thu của các bộ phim.

Selenium là một công cụ tự động hóa web mạnh mẽ, giúp em có thể điều khiển trình duyệt web và tương tác với các trang web động. Sử dụng Selenium, em có thể xử lý các trang web yêu cầu người dùng phải cuộn trang hoặc nhấn nút để tải thêm dữ liệu. Selenium giúp em tự động thu thập thông tin từ các trang web này một cách hiệu quả.

Requests là một thư viện HTTP đơn giản nhưng mạnh mẽ trong Python, giúp gửi các yêu cầu HTTP/1.1 dễ dàng và truy xuất dữ liệu từ các trang web. Requests hỗ trợ nhiều tính năng hữu ích như duyệt qua các trang web, gửi dữ liệu dưới dạng biểu mẫu và xử lý cookies, giúp em có thể thu thập dữ liệu từ các trang web tinh một cách dễ dàng.

4.4.2 Công nghệ sử dụng để update dữ liệu

Việc duy trì và cập nhật dữ liệu phim ảnh một cách liên tục là điều cần thiết để hệ thống gợi ý phim luôn cung cấp thông tin mới nhất và chính xác. Để tự động hóa quy trình này, em đã sử dụng GitHub Actions. GitHub Actions là một công cụ CI/CD (Continuous Integration/Continuous Deployment) tích hợp sẵn trong GitHub, cho phép thiết lập các workflow tự động chạy khi có sự kiện cụ thể.

Bằng cách sử dụng GitHub Actions, em đã thiết lập các workflow để tự động cập nhật dữ liệu phim từ IMDb hàng ngày. Các workflow này bao gồm việc chạy các script Selenium và Requests để thu thập dữ liệu mới, sau đó cập nhật vào cơ sở dữ liệu của hệ thống. Việc tự động hóa này giúp đảm bảo dữ liệu luôn được cập nhật một cách liên tục mà không cần sự can thiệp thủ công.

Cụ thể, mỗi ngày GitHub Actions sẽ kích hoạt các script crawl dữ liệu để kiểm tra và cập nhật những thay đổi mới nhất trên IMDb. Điều này bao gồm các trường thông tin như doanh thu, trailer và hình ảnh. Workflow GitHub Actions sẽ thực hiện các bước sau: Sử dụng Requests để gửi các yêu cầu HTTP và lấy dữ liệu tinh từ các trang web liên quan. Xử lý và làm sạch dữ liệu thu thập được, đảm bảo dữ

liệu đúng định dạng và cấu trúc. Cập nhật cơ sở dữ liệu MySQL với các thông tin mới nhất. Việc sử dụng GitHub Actions không chỉ giúp tự động hóa quy trình cập nhật dữ liệu mà còn đảm bảo tính nhất quán và chính xác của thông tin phim ảnh trong hệ thống. Điều này giúp hệ thống gợi ý phim luôn cung cấp các gợi ý và thông tin mới nhất, đáp ứng nhu cầu của người dùng một cách tốt nhất.

Chương này đã trình bày các công nghệ và nền tảng mà em sử dụng trong quá trình phát triển hệ thống gợi ý phim. Việc lựa chọn Django và MySQL cho backend, Bootstrap và JQuery cho frontend, Git và GitHub cho việc lưu trữ mã nguồn, cùng với kỹ thuật crawl dữ liệu từ IMDb và BoxOfficeMojo đã giúp em xây dựng một hệ thống gợi ý phim hiệu quả, ổn định và thân thiện với người dùng. Những công nghệ này không chỉ đáp ứng yêu cầu của hệ thống mà còn tạo điều kiện thuận lợi cho việc mở rộng và nâng cấp 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, em lựa chọn kiến trúc phần mềm MVT trong Django.

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, v.v.

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 em 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.

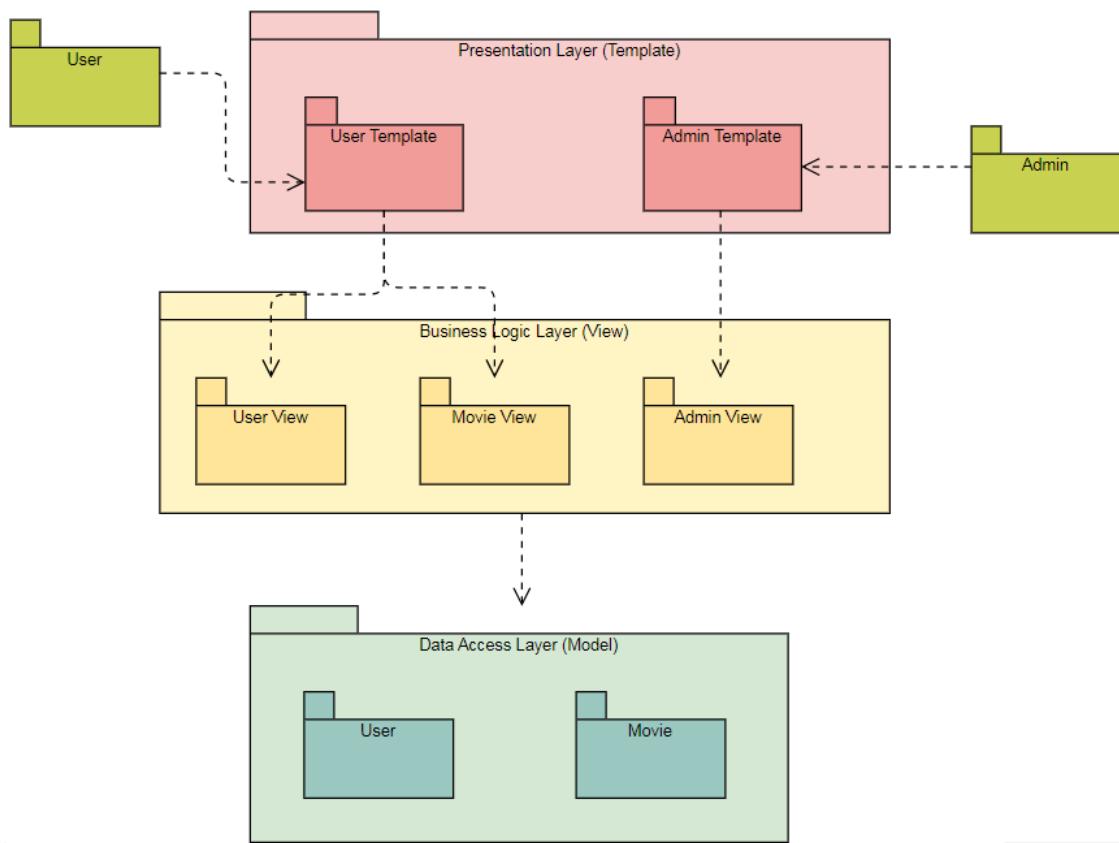
• Template (T):

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, v.v.

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: Chứa các định nghĩa về dữ liệu và cấu trúc dữ liệu của ứng dụng.

Nhiệm vụ: Mô hình hóa các thực thể trong hệ thống, bao gồm các bảng cơ sở dữ liệu và các mối quan hệ giữa chúng.

Các thành phần: MovieInformation, Awards, Director, Cast, Genres, v.v.

Phụ thuộc: Không phụ thuộc vào bất kỳ gói nào khác. Đây là tầng thấp nhất trong kiến trúc, nơi dữ liệu được định nghĩa và quản lý.

2. Business Logic Layer (View)

Mục đích: Xử lý logic nghiệp vụ và quản lý các yêu cầu từ người dùng.

Nhiệm vụ: Truy xuất dữ liệu từ các model, xử lý và chuẩn bị dữ liệu để hiển thị trong các template.

Các thành phần: FilmListView, DirectorListView, GenreListView, ReviewView, v.v.

Phụ thuộc: Phụ thuộc vào gói Models. View sẽ truy xuất dữ liệu từ model và xử lý logic nghiệp vụ. Không phụ thuộc vào gói Templates trực tiếp. Dữ liệu được chuẩn bị trong view sẽ được truyền đến template để hiển thị.

3. Presentation Layer (Template)

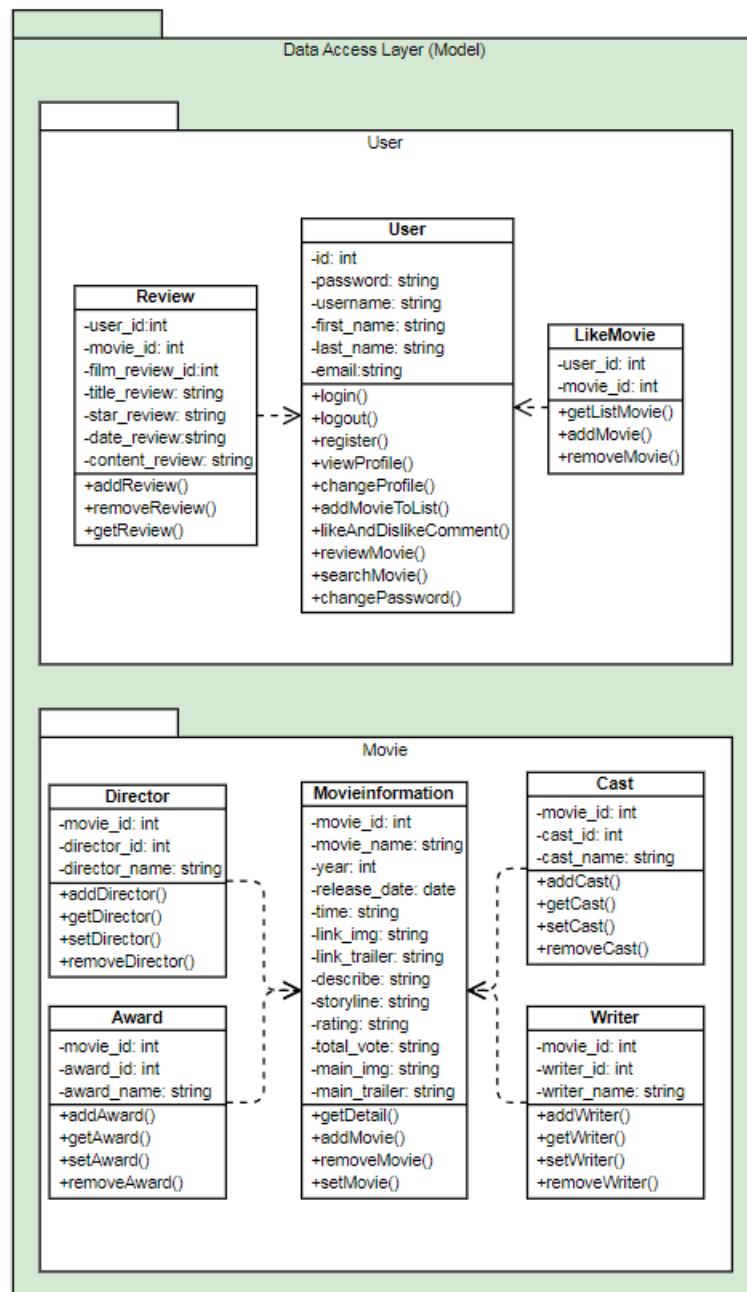
Mục đích: Định nghĩa giao diện người dùng.

Nhiệm vụ: Hiển thị dữ liệu được truyền từ view dưới dạng HTML, CSS và JavaScript.

Các thành phần: Các tệp HTML template cho các trang web khác nhau của ứng dụng.

Phụ thuộc: Phụ thuộc vào gói Views. Templates nhận dữ liệu từ view và hiển thị nó. Không phụ thuộc vào gói Models trực tiếp. Mọi dữ liệu 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

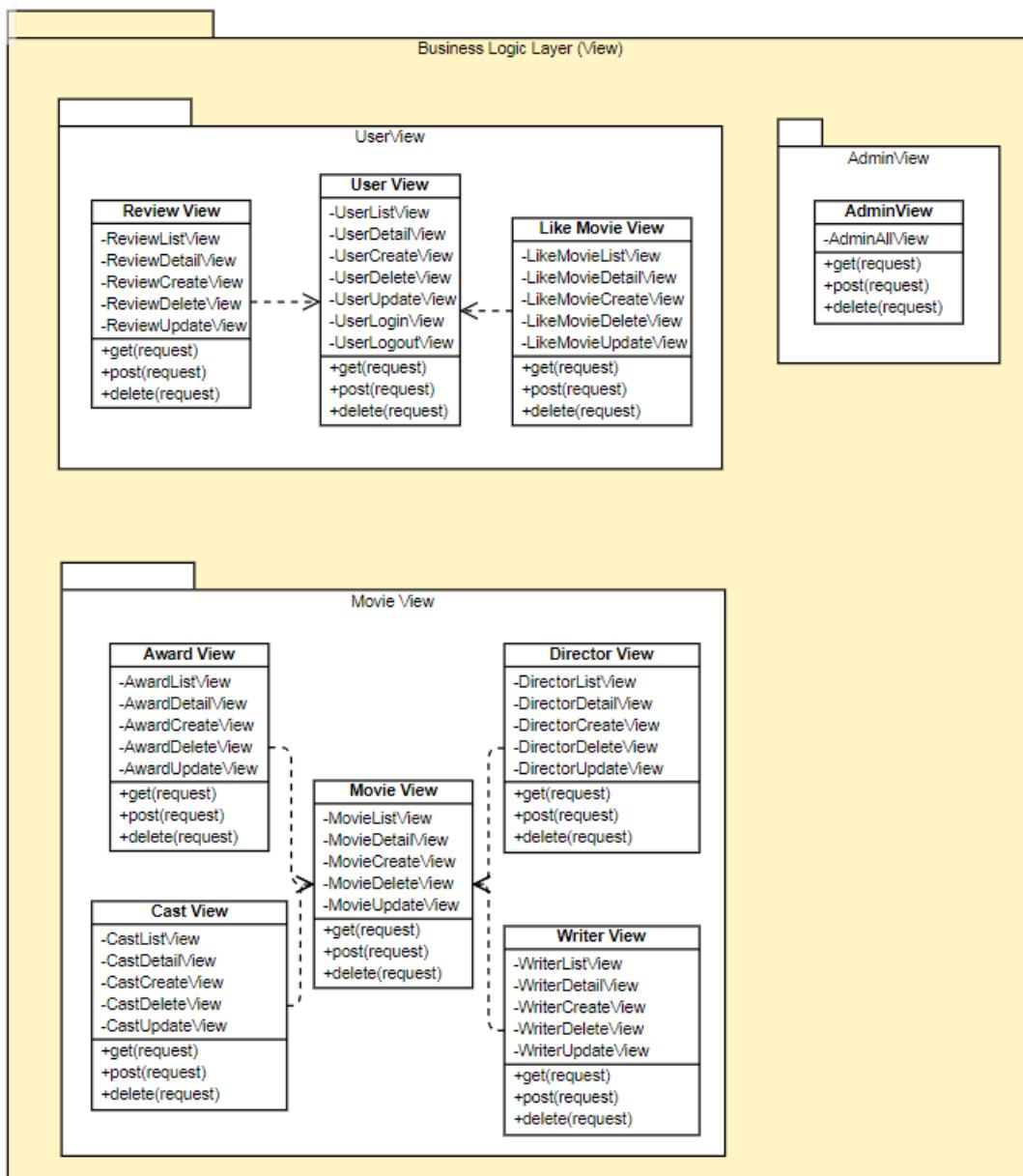
User.

- **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 LikeMovieView: 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.

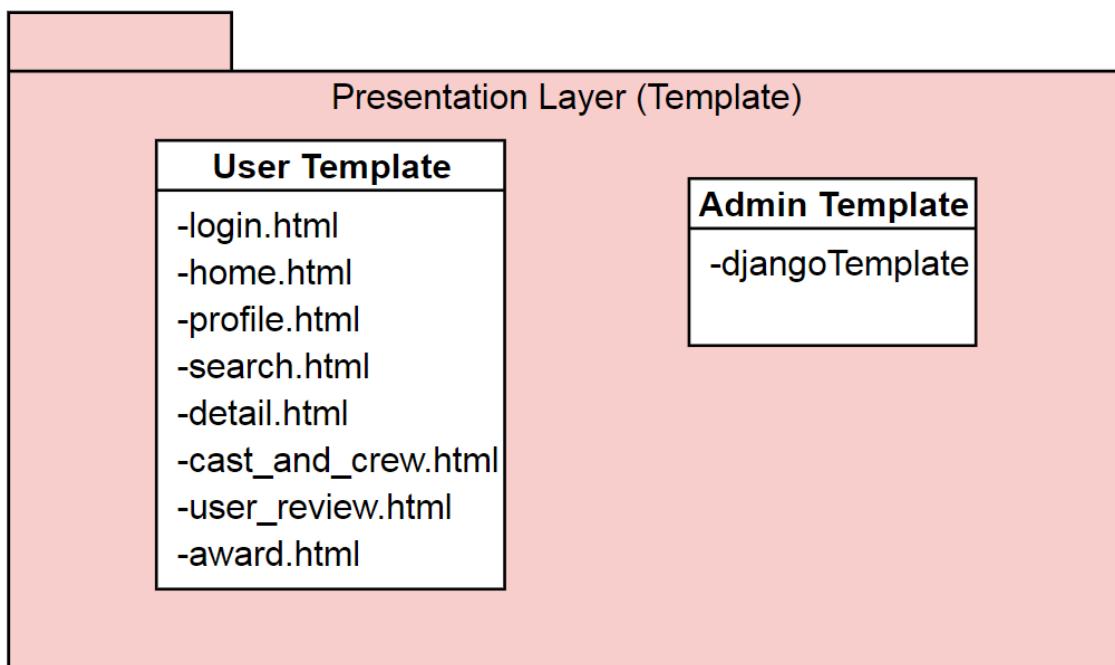
Phụ thuộc: 4 class còn lại có quan hệ phụ thuộc với Movieinfomation, lấy Movieinfomation làm gốc, các thông tin còn lại đều là thông tin bổ sung.

- **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:

Gồm 2 class là UserTemplate, AdminTemplate.

Mô tả: Class UserTemplate: Lớp này chịu trách nhiệm quản lý các mẫu HTML dành cho giao diện người dùng. Các mẫu này bao gồm các trang đăng nhập, đăng ký, trang cá nhân, trang đánh giá phim, và các trang hiển thị danh sách phim, diễn viên, đạo diễn, v.v. Class AdminTemplate: Lớp này chịu trách nhiệm quản lý các mẫu HTML dành cho giao diện admin. Các mẫu này bao gồm các trang quản lý người dùng, quản lý phim, duyệt đánh giá, và các trang khác mà admin cần sử dụng để quản lý hệ thống.

Phụ thuộc: 2 class này không có mối quan hệ phụ thuộc với nhau. Chúng độc lập và phục vụ các mục đích khác nhau: một bên là giao diện người dùng và một bên là giao diện admin.

5.2 Thiết kế chi tiết

5.2.1 Thiết kế giao diện

Khi thiết kế giao diện cho ứng dụng, em đã chọn các thông số và tiêu chuẩn nhất định để đảm bảo tính nhất quán và tiện dụng. Các màn hình mục tiêu cho ứng dụng này có độ phân giải 1920x1080 pixels, phù hợp với các kích thước màn hình từ 13 inch trở lên. Ứng dụng sẽ hỗ trợ hiển thị 16 triệu màu để đảm bảo các yếu tố đồ họa và hình ảnh hiển thị sắc nét.

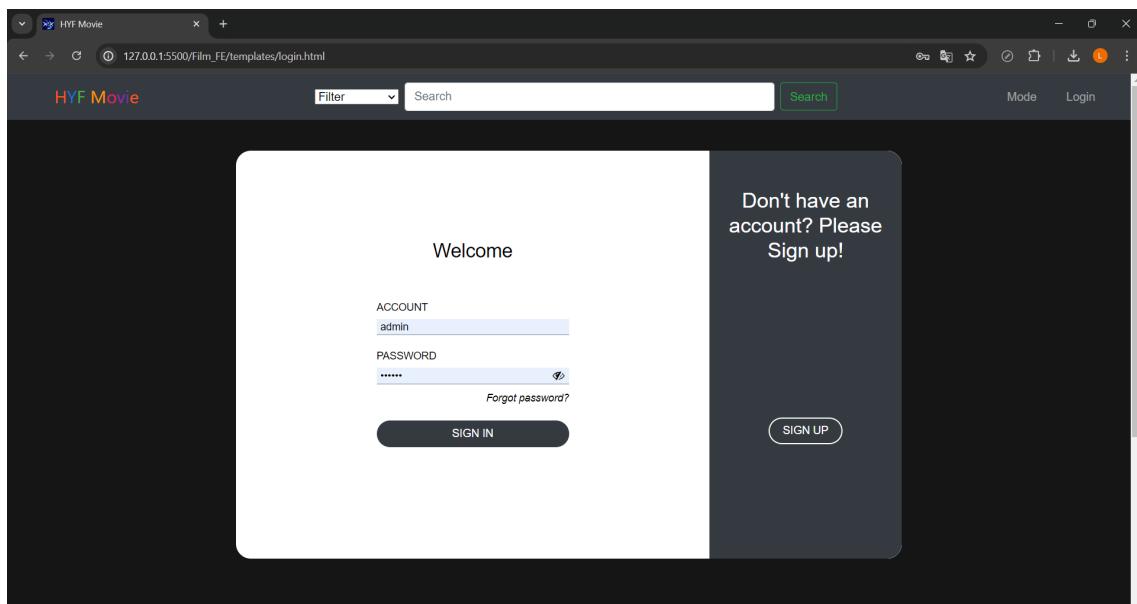
Trong quá trình thiết kế, em đã áp dụng các quy chuẩn cụ thể để tạo ra giao

diện người dùng nhất quán. Các nút bấm được thiết kế với các kích thước chuẩn, dễ dàng nhận biết và sử dụng. Các điều khiển như dropdowns, checkboxes, và radio buttons được thiết kế để có giao diện đồng nhất và rõ ràng. Các thông điệp phản hồi cho người dùng sẽ được hiển thị ở vị trí dễ thấy, ở phía trên bên phải.

Về phối màu, em đã chọn một bảng màu chính với các tông màu xanh dương và lá cây và đen, tạo cảm giác dễ chịu và chuyên nghiệp. Các màu phụ được sử dụng để nhấn mạnh các yếu tố quan trọng hoặc thông báo trạng thái khác nhau như thành công, cảnh báo, lỗi và hover.

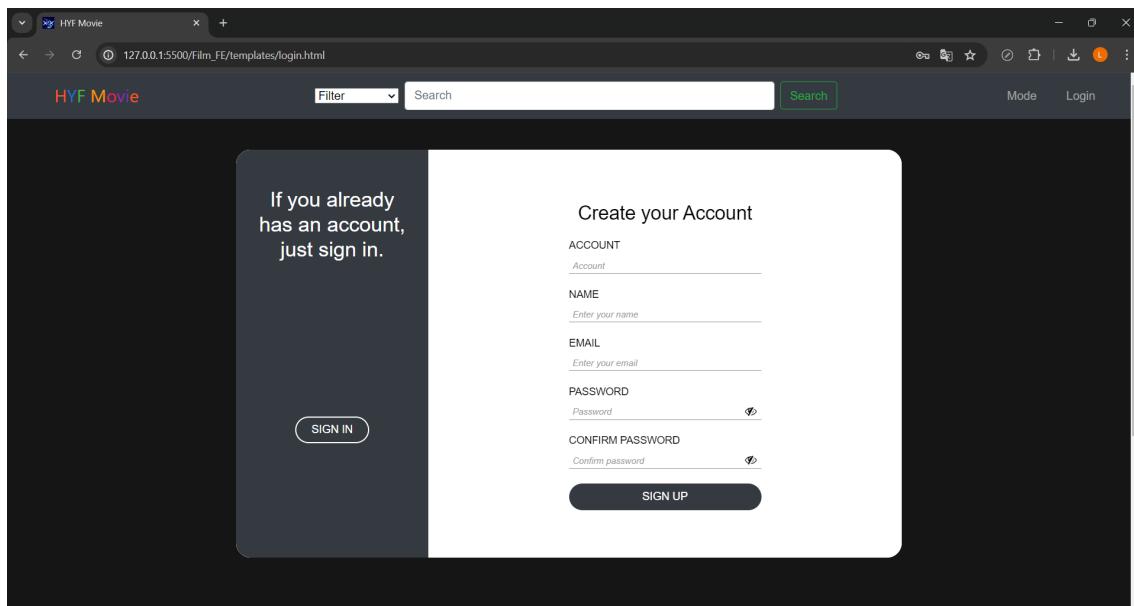
Dưới đây là một số hình ảnh minh họa cho thiết kế giao diện của các chức năng quan trọng trong ứng dụng:

Màn hình đăng nhập



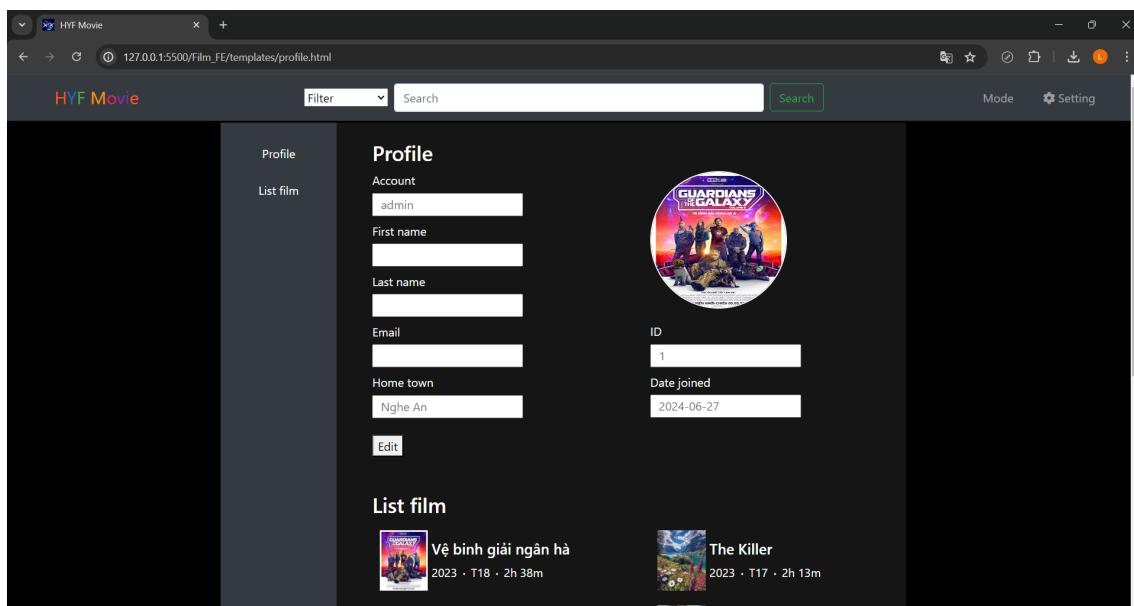
Hình 5.5: Màn hình đăng nhập

Màn hình đăng ký



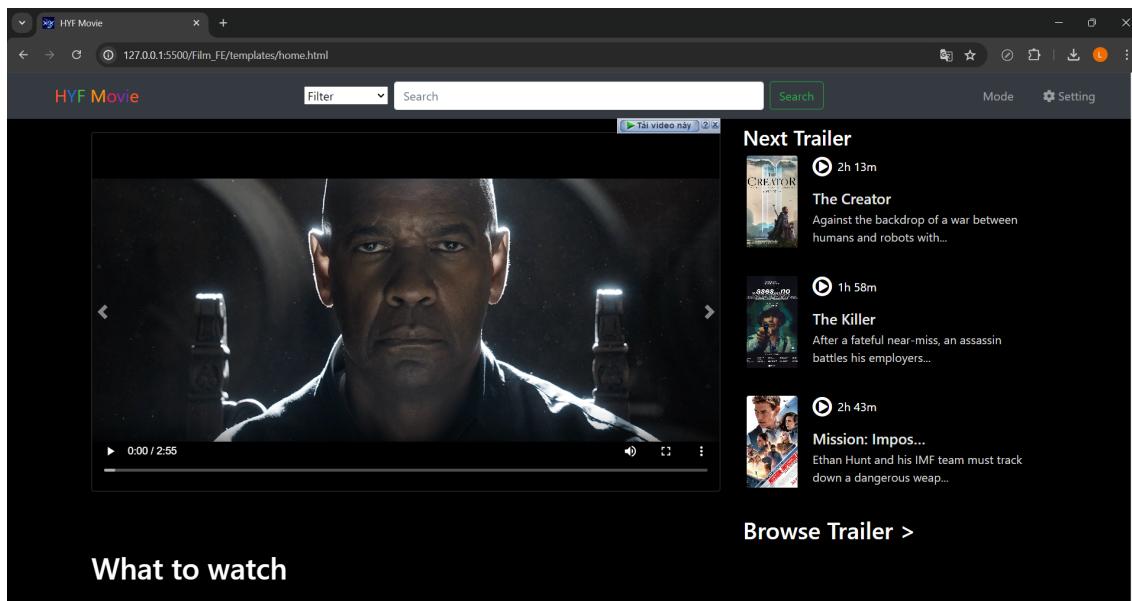
Hình 5.6: Màn hình đăng ký

Màn hình profile



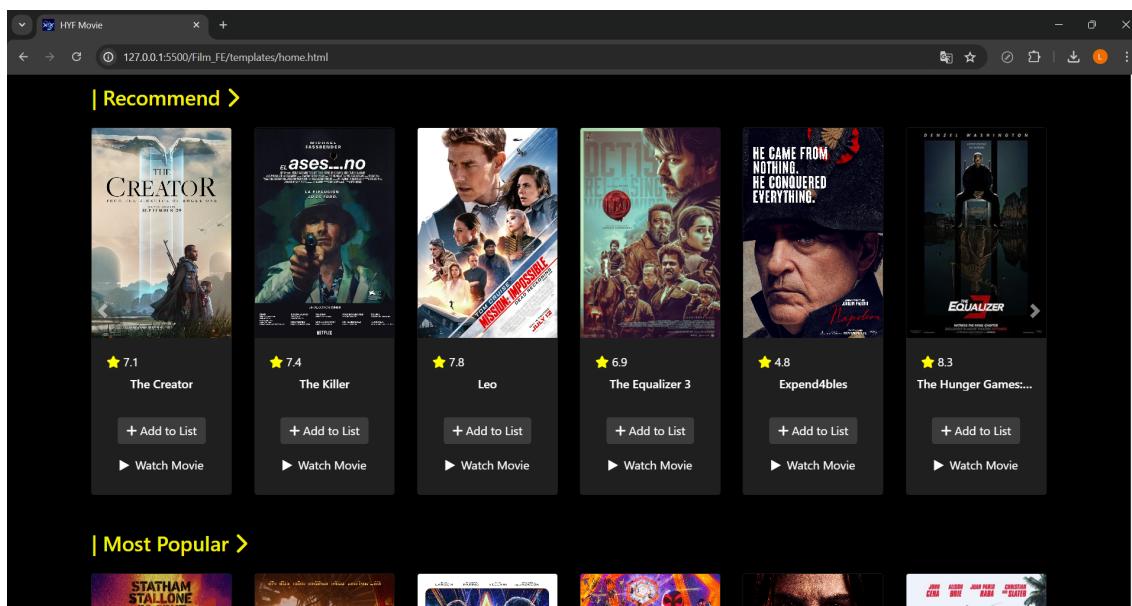
Hình 5.7: Màn hình profile

Màn hình chính



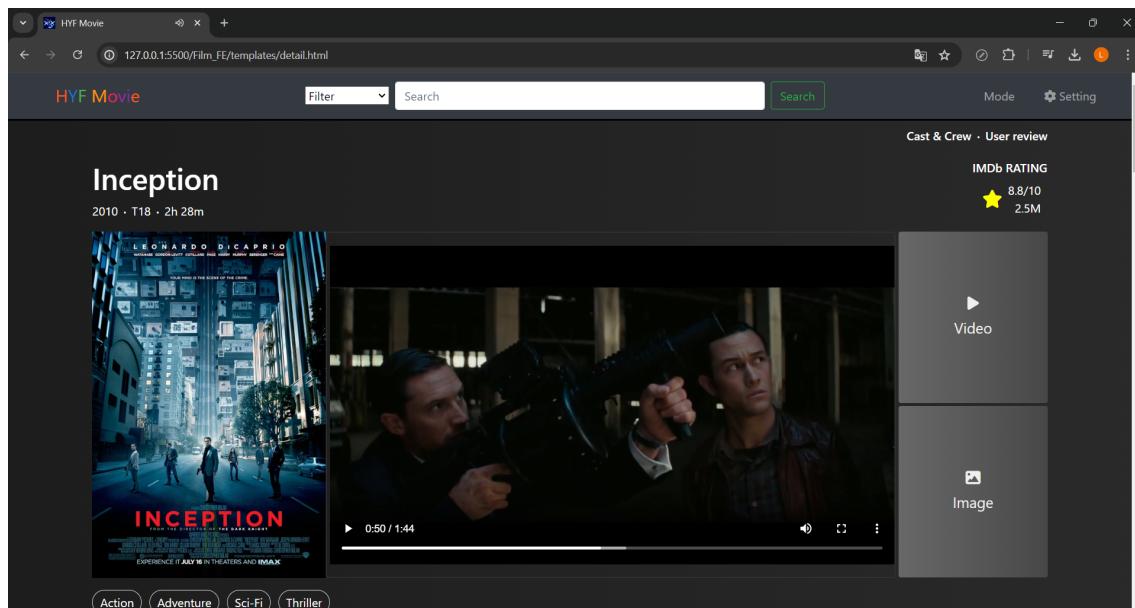
Hình 5.8: Màn hình chính

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

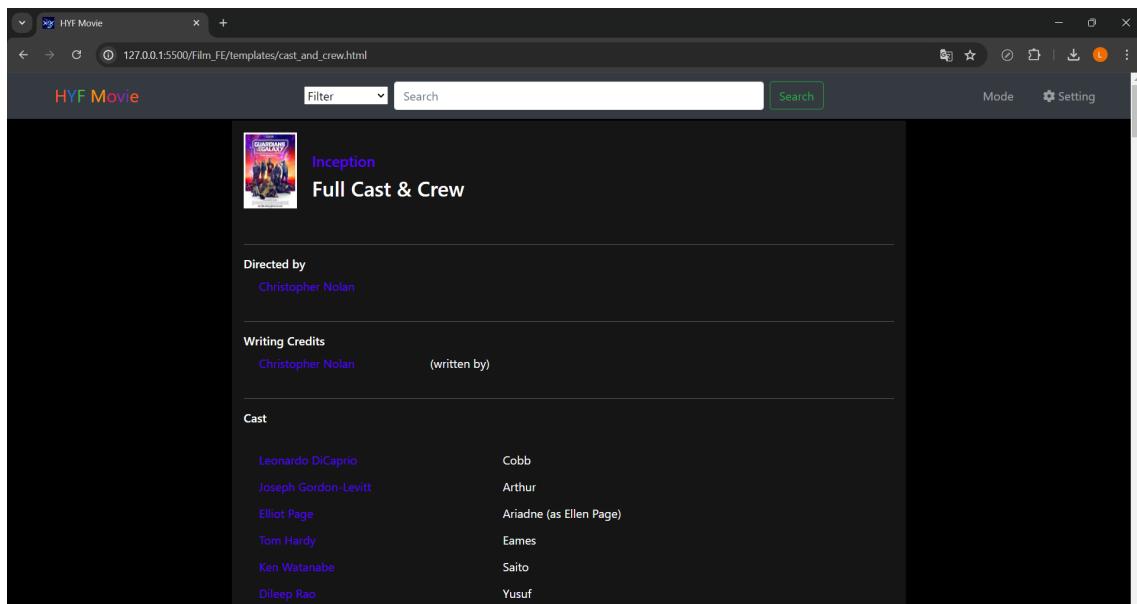


Hình 5.10: Màn hình thông tin chi tiết 1

Awards	Maybe you like
Top cast <ul style="list-style-type: none"> Leonardo DiCaprio (Cobb) Elliot Page (Ariadne (as Ellen Page)) Ken Watanabe (Saito) Cillian Murphy (Robert Fischer) Marion Cotillard (Mal) 	The Creator 2023 - 2h 13m Meg 2: The Trench 2023 - 1h 56m Black Widow 2021 - 2h 14m The Hunger Games: The Ballad of Songbirds and Snakes 2023 - 2h 45m Venom 2018 - 1h 52m The Meg 2018 - 1h 53m

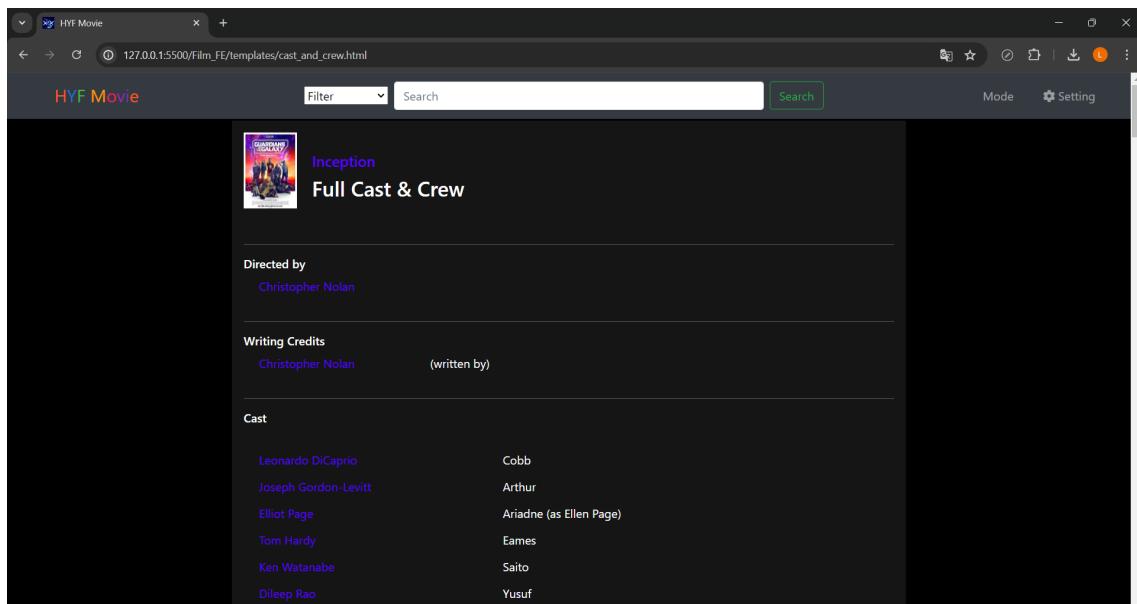
Hình 5.11: Màn hình thông tin chi tiết 2

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



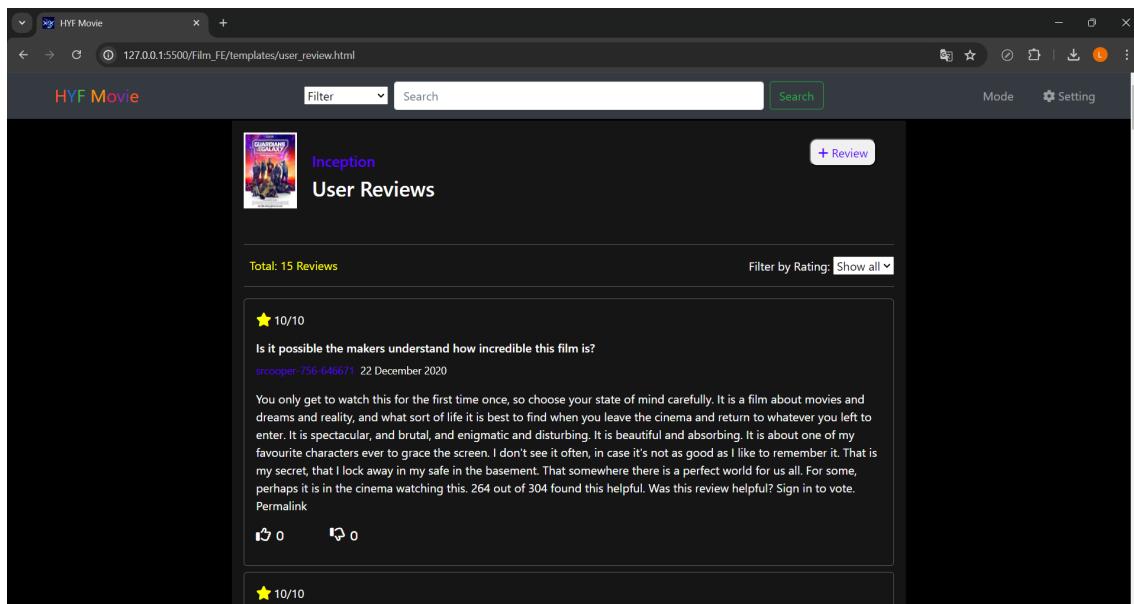
Hình 5.12: Màn hình tổng hợp đạo diễn, diễn viên, ...

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



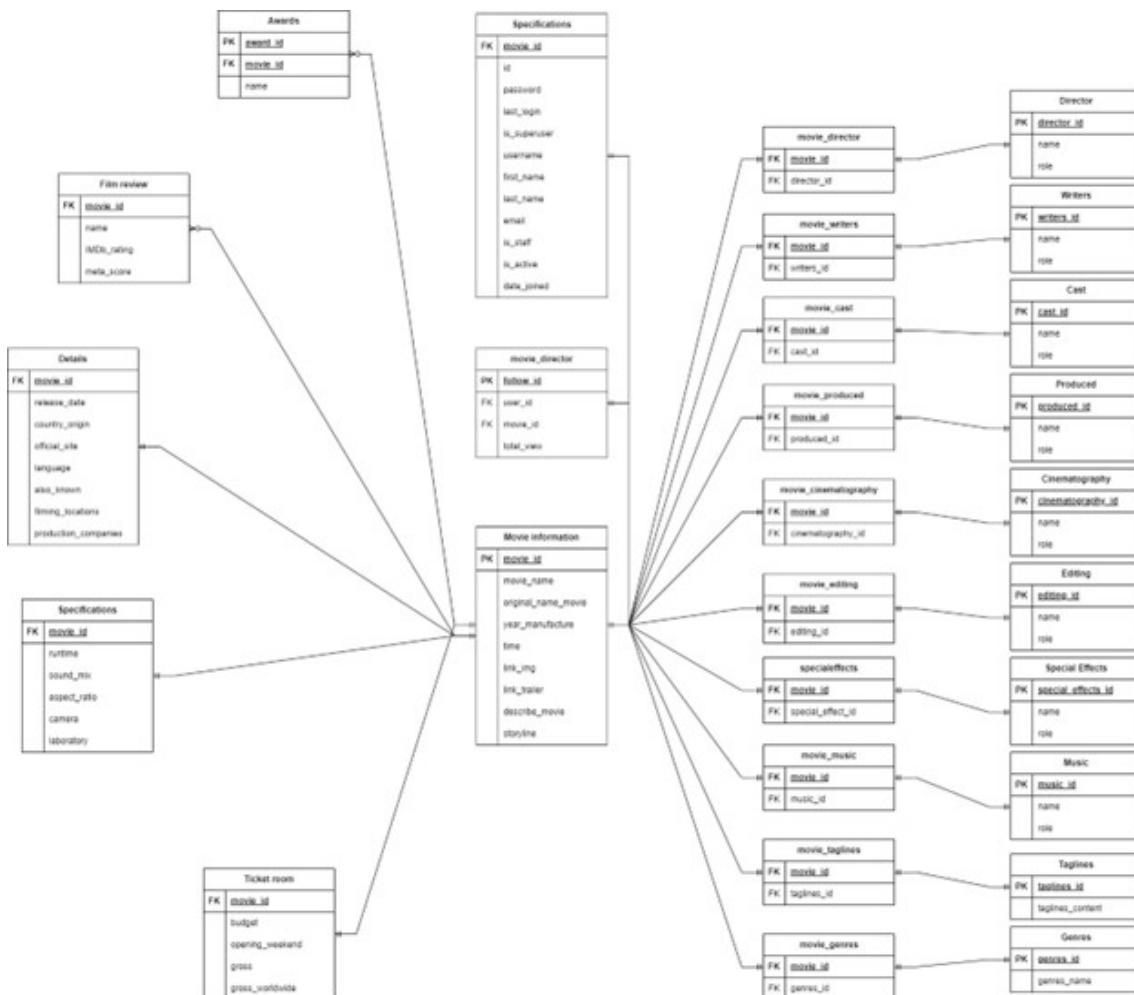
Hình 5.13: Màn hình tổng hợp đạo diễn, diễn viên, ...

Màn hình review phim

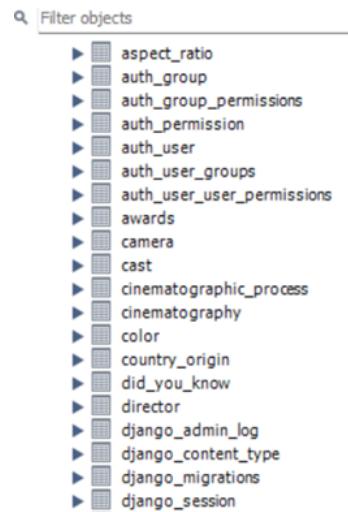
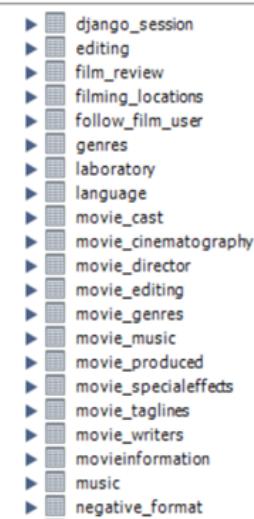
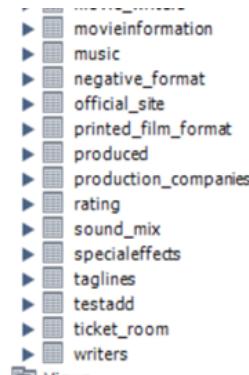


Hình 5.14: Màn hình review phim

5.2.2 Thiết kế cơ sở dữ liệu



Hình 5.15: Sơ đồ ERD

**Hình 5.16:** Hình ảnh các table trong database**Hình 5.17:** Hình ảnh các table trong database**Hình 5.18:** Hình ảnh các table trong database

5.2.3 Thiết kế hệ thống crawl dữ liệu

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

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

Hình 5.19: Tổng hợp các thông tin cần lấy

BoxofficeMojo: Ở trang web này, em 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.

5.3 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, v.v. mà em sử dụng để phát triển ứng dụng.

Mục đích	Công cụ	Tài liệu tham khảo
IDE lập trình	Visual Studio Code (VSCode)	[1]
Backend Framework	Django	[2]
API Framework	Django Rest Framework	[3]
Frontend	HTML	[4]
Styling	CSS	[5]
Frontend Framework	Bootstrap	[6]
JavaScript Library	jQuery	[7]
Phiên bản kiểm soát	Git	[8]
Kho lưu trữ mã nguồn	GitHub	[9]
Công cụ crawl dữ liệu	Selenium	[10]
HTTP Library	Requests	[11]
Công cụ CI/CD	GitHub Actions	[12]
Cơ sở dữ liệu	MySQL	[13]

Bảng 5.1: Danh sách thư viện và công cụ sử dụng

5.3.2 Kết quả đạt được

Mô tả kết quả đạt được: Sau quá trình phát triển, ứng dụng Django của em đã hoàn thành với các sản phẩm được đóng gói bao gồm:

Gói Model: Chứa các mô hình dữ liệu (models) quản lý thông tin phim, người dùng, đánh giá, v.v.

Gói View: Chứa các lớp hiển thị và xử lý giao diện cho người dùng và admin.

Gói Template: Chứa các mẫu HTML dành cho giao diện người dùng và admin.

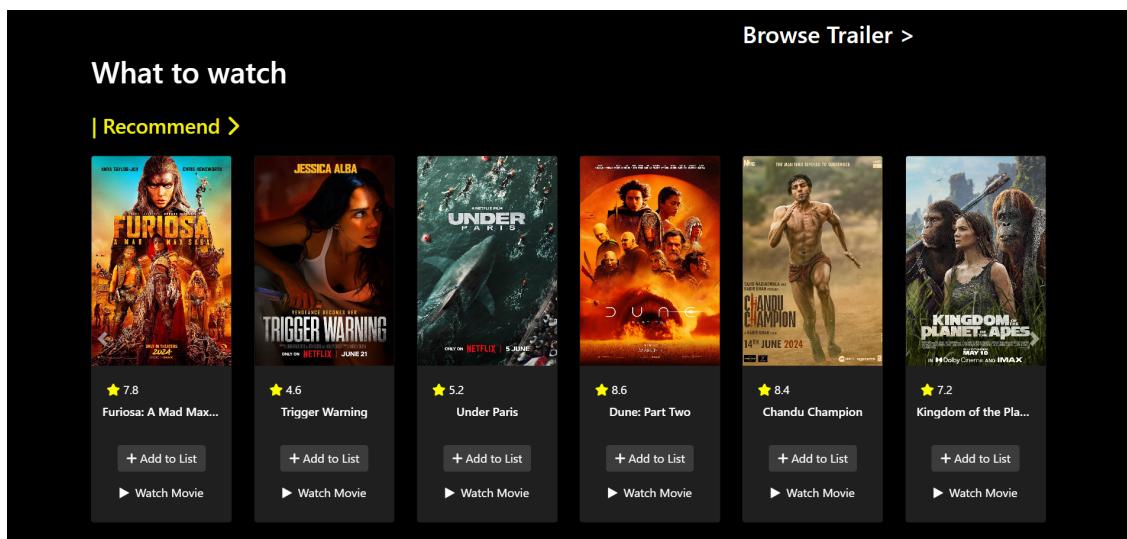
Các sản phẩm được đóng gói có ý nghĩa và vai trò cụ thể trong việc quản lý và hiển thị thông tin về phim, đạo diễn, diễn viên, đánh giá phim, và các thông tin liên quan khác. Ứng dụng này cung cấp nền tảng để người dùng có thể tìm kiếm và xem thông tin về các bộ phim, viết đánh giá, và tương tác với nội dung. Ngoài ra, ứng dụng còn có thể đưa ra những bộ phim phù hợp nhất với từng thao tác của người dùng.

Thống kê các thông tin của ứng dụng: Số dòng code crawl dữ liệu: 1500 dòng Số dòng code phía backend: 3000 dòng Số dòng code phía frontend: 7500 dòng Số lớp: 170 lớp Số gói: 7 gói Dung lượng toàn bộ mã nguồn: Khoảng 300MB khi đã có hệ thống gợi ý và dữ liệu, còn nếu mỗi phần code thì dung lượng khoảng 20MB

Ứng dụng đã hoàn thành với cấu trúc rõ ràng, mã nguồn được tổ chức tốt trong các gói và lớp. Thống kê chi tiết về số dòng code, số lớp, và dung lượng mã nguồn giúp đánh giá và theo dõi tiến trình phát triển của dự án. Ứng dụng đáp ứng các yêu cầu đặt ra và sẵn sàng để triển khai và sử dụng.

5.3.3 Minh họa các chức năng chính

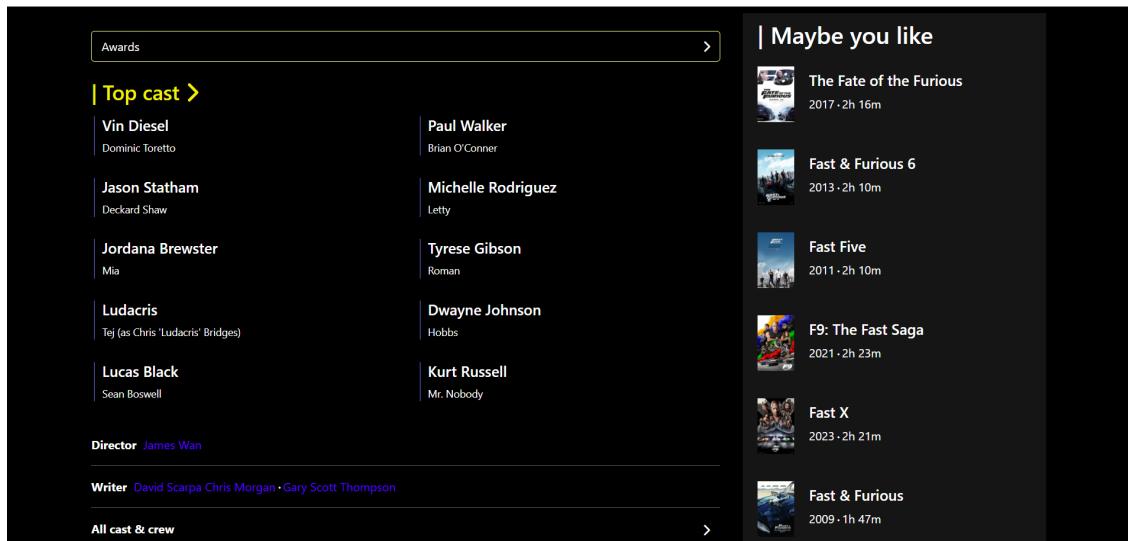
Chức năng gợi ý phim dựa vào sở thích người dùng



Hình 5.20: Chức năng gợi ý phim dựa vào sở thích người dùng

Dựa vào lịch sử thêm vào danh sách yêu thích 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.

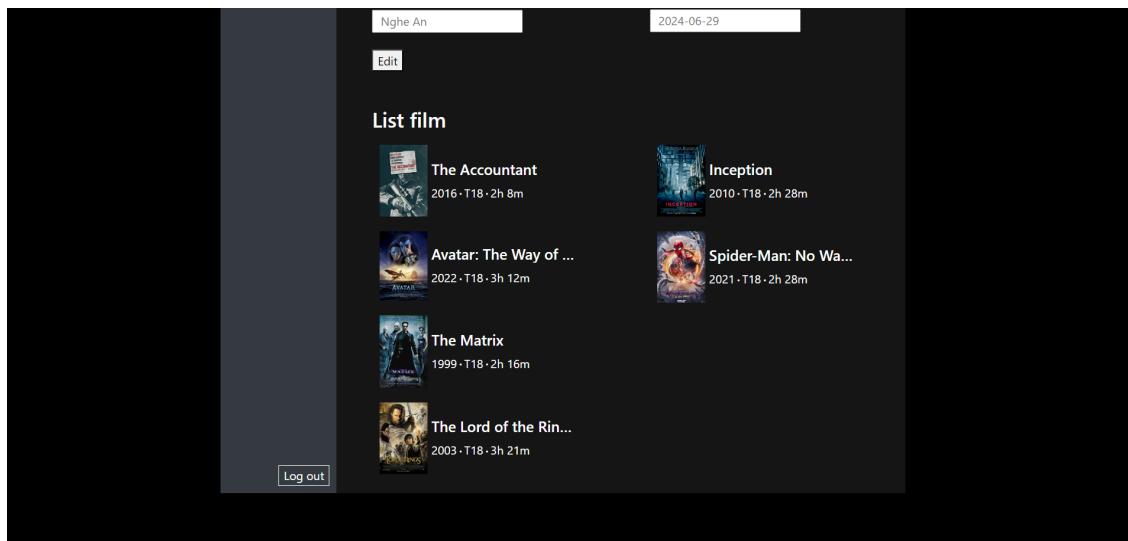
Chức năng gợi ý phim dựa vào nội dung phim đang xem



Hình 5.21: Chức năng gợi ý phim dựa vào nội dung phim đang xem

Dựa vào bộ phim mà người dùng đang tìm hiểu, hệ thống có thể đưa ra các bộ phim tương tự với bộ phim đang xem.

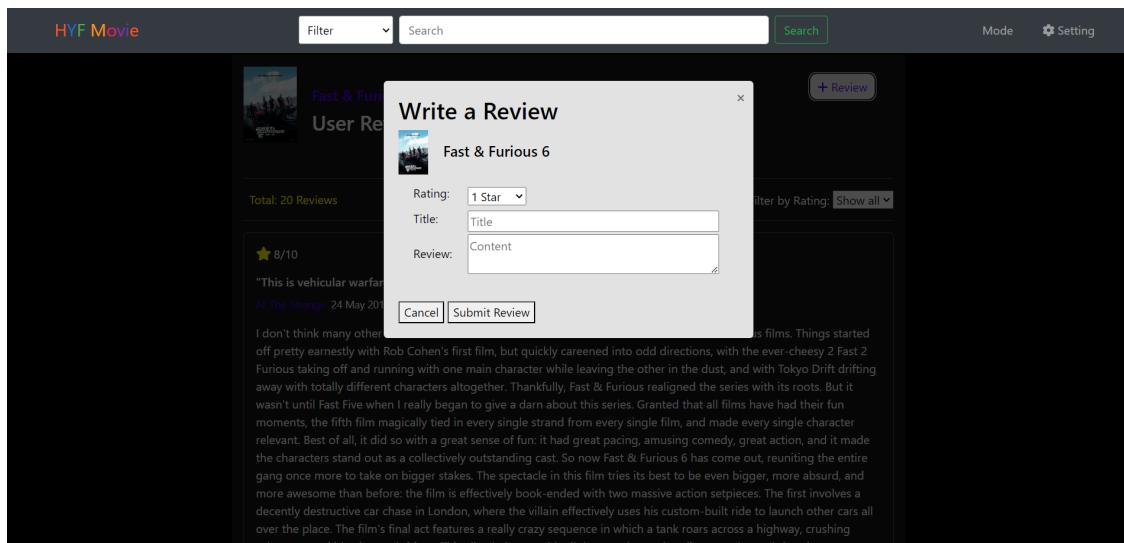
Chức năng thêm phim vào danh sách yêu thích



Hình 5.22: Chức năng thêm phim vào danh sách yêu thích

Khi người dùng có hứng thú với một bộ phim nào đó, họ có thể thêm phim đó vào danh sách yêu thích của mình.

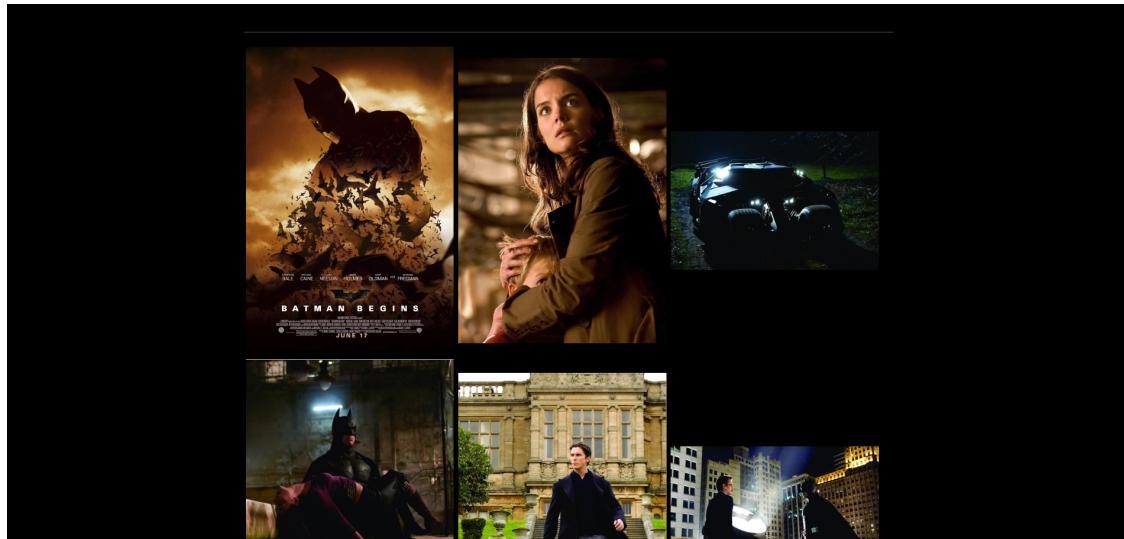
Chức năng review phim



Hình 5.23: Chức năng review phim

Người dùng có thể đọc bình luận, đánh giá của một bộ phim và có thể tự mình đánh giá về bộ phim đó.

Chức năng xem toàn bộ ảnh của phim



Hình 5.24: Chức năng xem toàn bộ ảnh của phim

Người dùng có thể xem danh sách ảnh có trong bộ phim đó, dữ liệu được tổng hợp từ các nguồn uy tín.

5.4 Kiểm thử

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

Hình 5.25: Kiểm thử chức năng 1

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 phân chi tiết phim	Kiểm thử chức năng gợi 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 gợi ý phim theo sở thích	Vào trang chủ	Hiển thị list phim gợi ý cho người dùng	Thành công

Hình 5.26: Kiểm thử chức năng 2

Qua những bước kiểm thử trên, có thể thấy ứng dụng đã sẵn sàng đi vào hoạt động khi toàn bộ test case đều được thông qua.

5.5 Triển khai

Mô hình triển khai Ứng dụng được triển khai theo mô hình client-server với phần frontend (FE) chạy trên live server và phần backend (BE) được triển khai bằng Django trên máy tính cá nhân (local).

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

Hệ điều hành: Windows/Linux/MacOS

CPU: Intel i5 hoặc tương đương

RAM: 8GB

Ổ cứng: 256GB SSD

Kết nối mạng: Internet tốc độ cao

Các bước triển khai

1. Triển khai frontend (FE):

Sử dụng Live Server trong Visual Studio Code (VS Code) để chạy ứng dụng frontend.

Live Server cung cấp một máy chủ phát triển đơn giản, tự động nạp lại trang khi có thay đổi trong mã nguồn.

2. Triển khai backend (BE):

Cài đặt và cấu hình Django trên máy tính cá nhân.

Cài đặt cơ sở dữ liệu MySQL.

Chạy server Django bằng lệnh `python manage.py runserver`.

3. Cơ sở dữ liệu:

Sử dụng cơ sở dữ liệu MySQL cho môi trường phát triển.

Tạo và áp dụng các migrations bằng lệnh `python manage.py migrate`.

Kết quả triển khai thử nghiệm

Mặc dù triển khai chỉ giới hạn trong môi trường local, nhưng một số kết quả thử nghiệm đã được ghi nhận:

Khả năng chịu tải: Với môi trường local, khả năng chịu tải chưa được kiểm thử kỹ lưỡng. Tuy nhiên, với số lượng truy cập thử nghiệm nhỏ, ứng dụng hoạt động mượt mà, không gặp vấn đề về hiệu suất.

Thời gian phản hồi: Thời gian phản hồi trung bình dưới 2 giây cho các yêu cầu cơ bản (truy cập trang chủ, tìm kiếm phim, xem chi tiết phim).

Phản hồi người dùng: Người dùng thử nghiệm đánh giá cao về giao diện đơn giản, dễ sử dụng và tính năng tìm kiếm, đánh giá phim hoạt động tốt. Đặc biệt, hệ thống gợi ý cho ra kết quả rất đúng với mong muốn của người dùng.

Triển khai ứng dụng trên môi trường local đã giúp xác minh các chức năng cơ bản và thu thập phản hồi từ người dùng thử nghiệm. Các kết quả thử nghiệm cho thấy ứng dụng hoạt động ổn định, phản hồi nhanh và giao diện thân thiện với người dùng. Để kiểm thử hiệu suất và khả năng chịu tải tốt hơn, cần triển khai trên server thực tế hoặc môi trường cloud.

CHƯƠNG 6. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT

Trong chương này, em sẽ trình bày các vấn đề mà em đã gặp phải và những giải pháp cụ thể mà em đã áp dụng để giải quyết chúng. Đồng thời, em cũng sẽ trình bày các đóng góp đáng chú ý mà em đánh giá cao trong quá trình thực hiện đồ án tốt nghiệp.

6.1 Thiếu dữ liệu phim ảnh

Vấn đề này phát sinh khi em cần một lượng lớn dữ liệu về các bộ phim để xây dựng hệ thống gợi ý. Để giải quyết vấn đề này, em đã tiến hành tìm kiếm và lựa chọn các trang web có uy tín để lấy dữ liệu. Việc này bao gồm việc phân tích cú pháp của các trang web, xác định cách truy cập và lấy dữ liệu một cách tự động.

6.2 Thiết kế database khó khăn với quá nhiều trường

Khi thiết kế cơ sở dữ liệu, em gặp khó khăn khi phải xử lý một lượng lớn các trường thông tin về phim và người dùng. Để giải quyết vấn đề này, em đã áp dụng nguyên tắc chuẩn hóa cơ sở dữ liệu và chia nhỏ các trường thông tin có liên quan thành các bảng dữ liệu con. Sau đó, em xác định mối liên hệ giữa các bảng và kết hợp chúng lại với nhau để tạo ra cấu trúc cơ sở dữ liệu hợp lý và hiệu quả.

6.3 Trang web có dữ liệu update liên tục theo ngày

Một trong số các trang web mà em lấy dữ liệu đó là IMDb, ở trang web này, có rất nhiều trường thông tin, nhưng việc khó khăn để lấy dữ liệu nhất đối với em chính là lấy link ảnh và link trailer. Ở trang web này, cứ 1 ngày thì các link ảnh và trailer sẽ được làm mới 1 lần, cho nên qua ngày khác thì link cũ sẽ bị hỏng và không truy cập được nữa. Em đã nghĩ ra 2 phương án để giải quyết đó là: Tải toàn bộ ảnh và trailer xuống hoặc tìm cách để hệ thống tự update data.

Ở cách giải quyết là tải toàn bộ ảnh và trailer xuống thì em đã loại bỏ ngay sau khi quyết định số bộ phim để demo hệ thống. Với 1000 bộ phim thì sẽ có tổng cộng hơn 50000 ảnh và 1000 video demo, nếu tải về thì rất nặng máy và không tối ưu. Vì thế, em đã nghiên cứu và tìm ra cách để hệ thống tự update data, đó chính là dùng github action.

6.3.1 Github Action

Giới thiệu:

Github Actions là một dịch vụ được cung cấp bởi GitHub để tự động hóa các quy trình phát triển phần mềm trong các repository. Bằng cách sử dụng các "actions" (hành động), em có thể thiết lập các luồng làm việc tự động, từ việc kiểm tra mã

nguồn, triển khai ứng dụng, đến tự động hóa các tác vụ như cập nhật dữ liệu trong hệ thống.

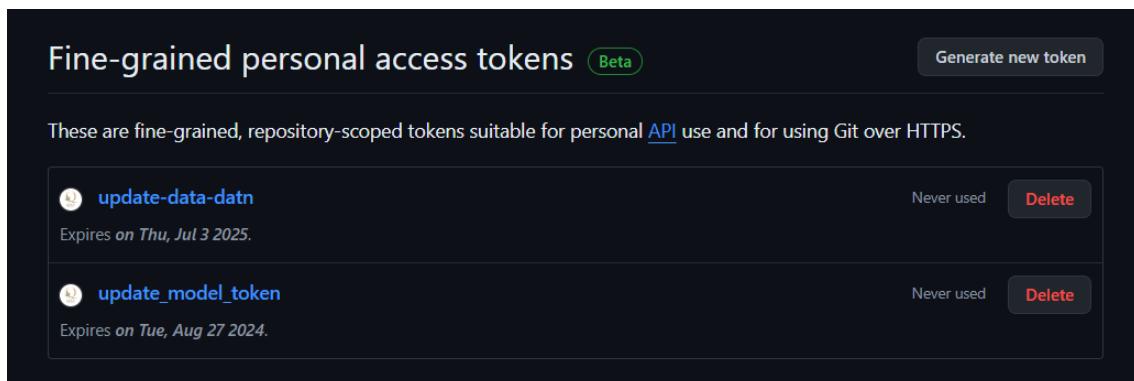
Triển khai:

Đầu tiên, em viết 1 file crawlJson.py, mục đích để crawl toàn bộ dữ liệu mới từ trang IMDb về và lưu vào file update.sql, khi có file update.sql rồi thì chỉ việc cập nhật vào local của mình là sẽ có dữ liệu mới.

Để hệ thống có thể tự động update được, sẽ có những bước như sau:

1. Tạo PAT (Personal Access Token) Github

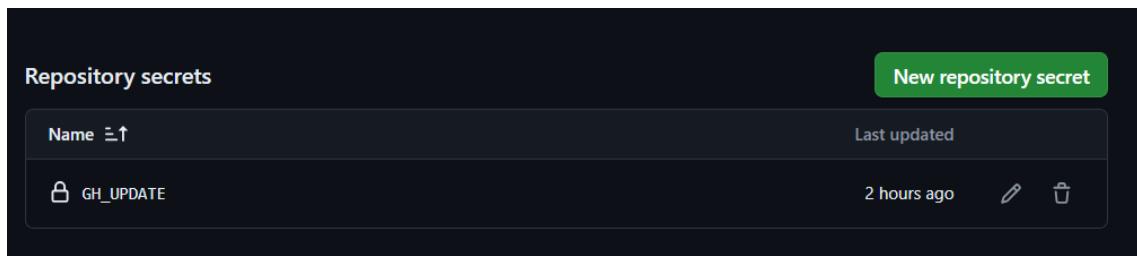
Personal Access Token (PAT) trên GitHub được sử dụng để cung cấp quyền truy cập cho các ứng dụng hoặc dịch vụ bên thứ ba vào tài khoản GitHub, mà không cần phải sử dụng mật khẩu. Ở trong hệ thống này, dịch vụ bên thứ 3 chính là bot tự động.



Hình 6.1: Tạo PAT Github

2. Tạo Secrets Repository

Mục đích của việc tạo Secrets trên GitHub là để bảo vệ và quản lý các thông tin nhạy cảm như mã token, mật khẩu, khóa API, và các thông tin quan trọng khác trong quá trình phát triển và triển khai phần mềm. Việc sử dụng Secrets giúp đảm bảo rằng các thông tin này không được 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 cho phép.



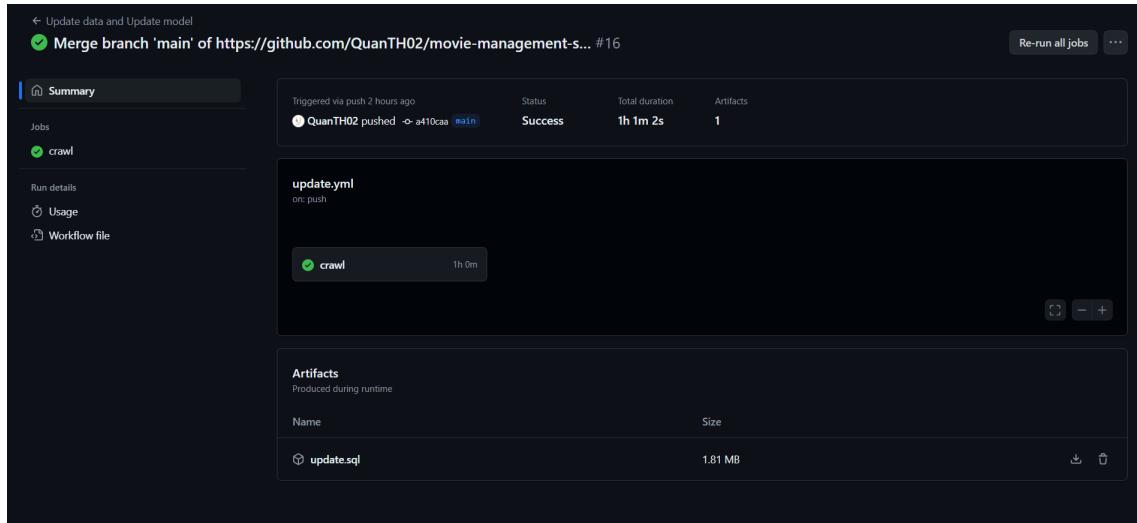
Hình 6.2: Tạo Secrets Repository

3. Viết file update.yml

Bước cuối cùng, em sẽ viết file update.yml để Github Action có thể chạy tự động

Kết quả:

Cứ 00h00 hằng ngày, hệ thống sẽ tự động crawl dữ liệu mới, lưu vào file update.sql và push code lên github trong khoảng 1 tiếng.



Hình 6.3: Hệ thống tự động crawl dữ liệu

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

The screenshot shows a MySQL Workbench interface. The title bar says "link_img". Below it is a toolbar with various icons. A query window displays the SQL command: "SELECT count(*) FROM moviedata.link_img;". The result grid shows one row with the value "49030".

	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 hệ thống gợi ý phim. Em 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 2 thuật toán Content-based và Collaborative.

6.4.1 Content-based

Giới thiệu:

Trong hệ thống content-based này, dựa trên nội dung của mỗi movie, em 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 em 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 em 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, em 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
12 128 0.08440632301936431
13 19 0.05394264748270165
14 86 0.04964744394500321
15 70 0.047996604986685044
16 32 0.047196291445337255
17 74 0.04686329421081032
18
19 genres
20 45 0.0
21 51 0.0
22 50 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 0.13878049563162084
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 0.06091447365093486
40 10 0.05759624460829198
41 21 0.05458436118761168
42

cast Untitled-2
1 cast
2 109 0.3647058800017843
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

```

Ln 38, Col 1 Spaces: 4 UTF-8 CRLF Markdown Go Live tabnine starter

Hình 6.5: Kiểm tra kết quả 1

Ở hình ảnh 6.5 có thể thấy bộ phim được gợi ý gần nhất với bộ phim Guardians of the Galaxy là bộ phim Guardians of the Galaxy Vol. 3. Hai bộ phim này có movie name, describe, cast và director khá tương tự nhau.

```

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 0.03660565702846335
15 135 0.031156937908150564
16 140 0.02794239278793543
17 44 0.0266216827027246
18
19 genres
20 45 0.0
21 51 0.0
22 50 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 0.3488033324437175
33 9 0.12522903952739983
34 45 0.0
35 51 0.0
36 50 0.0
37 49 0.0
38 48 0.0
39 47 0.0
40 46 0.0
``
```

```

cast Untitled-2
1 cast
2 25 0.49733589592274985
3 94 0.07541231794535281
4 30 0.07044728333428879
5 140 0.057882084735789555
6 74 0.053418775792202375
7 61 0.05041770904609579
8 107 0.049902341895944186
9 79 0.0498348654944694
10 34 0.0458437582028299
11 106 0.04566657169335134
12
13 director
14 106 1.0000000000000002
15 132 1.0000000000000002
16 18 0.26717567426324096
17 39 0.0
18 40 0.0
19 41 0.0
20 42 0.0
21 38 0.0
22 52 0.0
23 43 0.0
24
25 Movie name: The Hunger Games
26 Describe: Katniss Everdeen voluntarily takes her younger
sister's place in the Hunger Games: a televised
competition in which two teenagers from each of the twelve
Districts of Panem are chosen at random to fight to the
death.
27 Director: Gary Ross
28 Cast: Stanley Tucci Wes Bentley Jennifer Lawrence
Willow Shields Liam Hemsworth Elizabeth Banks Sandra
Ellis Lafferty Paula Malcomson Rhoda Griffis Sandino
Moya-Smith Josh Hutcherson Raiko Bowman Dwayne Boyd
Anthony Reynolds Judd Lormand
29
30 Jennifer Lawrence
31 Liam Hemsworth
32 Sandra Ellis
33 Lafferty
34 Hutcherson
35 Paula Malcomson
36 Willow Shields
37 Elizabeth Banks
38 Stanley Tucci
```

```

Ln 45 Col 1 Spaces: 4 UTF-8 CRLF Markdown Go Live tabnine starter

Hình 6.6: Kiểm tra kết quả 2

Ở hình ảnh 6.6 có thể thấy bộ phim được gợi ý gần nhất với bộ phim The Hunger Games: Catching Fire là bộ phim The Hunger Games. Hai bộ phim này có movie name, describe, cast và director khá tương tự nhau.

```

Movie id: 15
Movie name: Meg 2: The Trench
Describe: A research team encounters multiple threats while exploring the depths of the ocean, including a malevolent mining operation.
Genres: Action Adventure Horror Sci-Fi Thriller
Director: Ben Wheatley
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
Movie id result:
88 0.5584300169597662
38 0.35319267661183157
2 0.333988033722691944
105 0.3223983287172182
9 0.30436133874364807
13 75 0.30063584739093846
14 61 0.2873170128093802
15 140 0.28669542992067987
16 54 0.2861652412881892
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  
88 0.22183861524280854  
3 0.05523016212903862  
4 18 0.05310982691925622  
5 32 0.052348176714640225  
6 51 0.046815856484982065  
7 46 0.046561172845036136  
8 64 0.04160235447938892  
9 56 0.04011477030059863  
10 99 0.03699776836648709  
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

**Hình 6.7:** Kiểm tra kết quả 3

Ở hình ảnh 6.7 có thể thấy bộ phim được gợi ý gần nhất với bộ phim Meg 2: The Trench là bộ phim The Meg. Hai bộ phim này có movie name, describe, cast và director khá tương tự nhau.

Dựa trên các kết quả cho ra ở trên cùng với kiểm tra bằng tính trực quan có thể thấy hệ thống gợi ý phim bằng độ tương đồng được thực hiện khá chính xác.

#### 6.4.2 Collaborative

##### Giới thiệu:

Ý tưởng cơ bản của Collaborative Filtering là xác định mức độ quan tâm của một user tới một item dựa trên các users khác gần giống với user này. Đối với những users dễ tính, họ có thể đánh giá phim yêu thích là 5 sao, và nếu không thích thì đánh giá có thể thấp hơn một chút, chẳng hạn như 3 sao. Ví dụ, A, B đều thích phim Cảnh sát hình sự, tức đều rate bộ phim này 5 sao. Ta đã biết A cũng thích Người phán xử, vậy nhiều khả năng B cũng thích bộ phim này.

Vấn đề đặt ra là:

- Làm thế nào xác định được sự giống nhau giữa hai users?

- Khi đã xác định được các users gần giống nhau (similar users) rồi, làm thế nào dự đoán được mức độ quan tâm của một user lên một item?

### Phân tích, giải pháp:

Công việc quan trọng nhất phải làm trước tiên trong User-user Collaborative Filtering là phải xác định được sự giống nhau (similarity) giữa hai users.

Giả sử chúng ta có các users từ u0 đến u6 và các items từ i0 đến i4 trong đó các ô vuông trong bảng là số rating mà user đã đánh giá cho item đó. Trong một hệ thống có số lượng người dùng và số lượng item rất lớn, rất khó để có dữ liệu người dùng đánh giá nhiều item, một số người dùng thậm chí còn không đánh giá hoặc chỉ đánh giá một số sản phẩm. Những điều này sẽ ảnh hưởng rất lớn để xác định được sự giống nhau giữa hai user. Trong hình 6.8, các giá trị bị trống là user chưa đánh giá, khi đó hệ thống sẽ phải đi tìm những giá trị đó.

|    | U0 | U1 | U2 | U3 | U4 | U5 | U6 |
|----|----|----|----|----|----|----|----|
| I0 | 5  | 5  | 2  | 0  | 1  |    |    |
| I1 | 4  |    |    | 0  |    | 2  |    |
| I2 |    | 4  | 1  |    |    | 1  | 1  |
| I3 | 2  | 2  | 3  | 4  | 4  |    | 4  |
| I4 | 2  | 0  | 4  |    |    |    | 5  |

|    |      |      |     |      |     |     |      |
|----|------|------|-----|------|-----|-----|------|
| TB | 3.25 | 2.75 | 2.5 | 1.33 | 2.5 | 1.5 | 3.33 |
|----|------|------|-----|------|-----|-----|------|

**Hình 6.8:** Tính giá trị trung bình rating của các user

Để điền giá trị vào các ô trống trên, có thể điền giá trị an toàn là 2.5, vì nó chính là giá trị trung bình của 0 và 5, mức thấp nhất và mức cao nhất. Tuy nhiên, giá trị này vẫn có hạn chế. Đối với những users dễ tính, họ có thể đánh giá phim yêu thích là 5 sao, và nếu không thích thì đánh giá có thể thấp hơn một chút, chẳng hạn như 3 sao. Khi đó việc chọn giá trị 2.5 hơi thấp đối với những user đó. Và tương tự, những user khó tính có thể đánh giá 3 sao cho những bộ phim họ thích và thấp hơn thế nếu họ không thích.

Một giá trị có thể chấp nhận được là trung bình cộng của các ratings mà user tương ứng đã thực hiện Hình 6.8. Việc này sẽ tránh được tình trạng bị ảnh hưởng bởi những users quá khó tính hoặc dễ tính. Vì đối với mỗi user, họ lúc nào cũng có những items mà họ ưu tiên hơn những items khác, và được đánh giá cao hơn.

|    | U0    | U1    | U2   | U3    | U4   | U5   | U6    |
|----|-------|-------|------|-------|------|------|-------|
| I0 | 1.75  | 2.25  | -0.5 | -1.33 | -1.5 | 0    | 0     |
| I1 | 0.75  | 0     | 0    | -1.33 | 0    | 0.5  | 0     |
| I2 | 0     | 1.25  | -1.5 | 0     | 0    | -0.5 | -2.33 |
| I3 | -1.25 | -0.75 | 0.5  | 2.67  | 1.5  | 0    | 0.67  |
| I4 | -1.25 | -2.75 | 1.5  | 0     | 0    | 0    | 1.67  |

**Hình 6.9:** Tính lại rating theo giá trị trung bình

Việc tính giá trị trung bình này có thể xác định được user đó là dễ tính hay khó tính và có thể xem được item cụ thể user đó thích hay ghét bằng cách lấy giá trị rating item đó trừ đi giá trị trung bình Hình 6.9. Khi đó có thể điền các giá trị trống bằng giá trị 0, biểu hiện cho không thích cũng không ghét.

|    | U0    | U1    | U2    | U3    | U4    | U5    | U6    |
|----|-------|-------|-------|-------|-------|-------|-------|
| U0 | 1     | 0.83  | -0.58 | -0.79 | -0.82 | 0.2   | -0.36 |
| U1 | 0.86  | 1     | -0.87 | -0.4  | -0.55 | -0.23 | -0.71 |
| U2 | -0.58 | -0.87 | 1     | 0.27  | 0.32  | 0.47  | 0.96  |
| U3 | -0.79 | -0.4  | 0.27  | 1     | 0.87  | -0.29 | 0.18  |
| U4 | -0.82 | -0.55 | 0.32  | 0.87  | 1     | 0     | 0.16  |
| U5 | 0.2   | -0.23 | 0.47  | -0.29 | 0     | 1     | 0.56  |
| U6 | -0.36 | -0.71 | 0.96  | 0.18  | 0.16  | 0.56  | 1     |

**Hình 6.10:** Tính độ tương đồng giữa các user theo rating mới

Sau đó, ta có thể tính được độ tương đồng giữa các user theo rating mới Hình 6.10. Nhìn qua thì có thể thấy người có độ tương đồng cao nhất và u0 là u1.

Dựa vào độ tương đồng này, ta có thể dự đoán được rating cho item mà user này trước đó chưa đánh giá. Theo công thức như sau:

$$\hat{y}_{i,u} = \frac{\sum_{u_j \in \mathcal{N}(u,i)} \bar{y}_{i,u_j} \text{sim}(u, u_j)}{\sum_{u_j \in \mathcal{N}(u,i)} |\text{sim}(u, u_j)|} \quad (6.1)$$

Trong đó:

- $\hat{y}_{i,u}$ : Dự đoán giá trị đánh giá của người dùng  $u$  cho item  $i$ .
- $\mathcal{N}(u, i)$ : Tập hợp các người dùng đã đánh giá item  $i$  và có độ tương đồng với người dùng  $u$ .
- $\bar{y}_{i,u_j}$ : Giá trị đánh giá mà người dùng  $u_j$  đưa ra cho item  $i$ .
- $\text{sim}(u, u_j)$ : Độ tương đồng giữa người dùng  $u$  và người dùng  $u_j$ .

|    | U0    | U1    | U2    | U3    | U4    | U5   | U6    |
|----|-------|-------|-------|-------|-------|------|-------|
| I0 | 1.75  | 2.25  | -0.5  | -1.33 | -1.5  | 0.18 | -0.63 |
| I1 | 0.75  | 0.48  | -0.17 | -1.33 | -1.33 | 0.5  | 0.05  |
| I2 | 0.91  | 1.25  | -1.5  | -1.84 | -1.78 | -0.5 | -2.33 |
| I3 | -1.25 | -0.75 | 0.5   | 2.67  | 1.5   | 0.59 | 0.67  |
| I4 | -1.25 | -2.75 | 1.5   | 1.57  | 1.56  | 1.59 | 1.67  |

**Hình 6.11:** Dự đoán rating cho từng item

Sau khi dự đoán bằng công thức trên, ta có các kết quả dự đoán như Hình 6.11

|    | U0   | U1   | U2   | U3   | U4   | U5   | U6   |
|----|------|------|------|------|------|------|------|
| I0 | 5    | 5    | 2    | 0    | 1    | 1.47 | 2.27 |
| I1 | 4    | 3.47 | 2.61 | 0    | 1.8  | 2    | 4.54 |
| I2 | 3.85 | 4    | 1    | 1.54 | 2.16 | 1    | 1    |
| I3 | 2    | 2    | 3    | 4    | 4    | 1.3  | 4    |
| I4 | 2    | 0    | 4    | 2.6  | 2.94 | 1.46 | 5    |

**Hình 6.12:** Kết quả dự đoán rating cuối cùng của hệ thống

Sau khi có kết quả dự đoán dựa trên kết quả rating trung bình, ta cộng giá trị dự đoán đó với rating trung bình của user và được kết quả cuối cùng như Hình 6.12.

### Kiểm tra tính chính xác:

Sau khi chạy thuật toán, kiểm tra giá trị RMSE với bộ data 100000 lượt rate ta có được kết quả như sau:

```
rs = CF(rate_train, k = 30, uuCF = 1)
rs.fit()

n_tests = rate_test.shape[0]
SE = 0 # squared error
for n in range(n_tests):
 pred = rs.pred(rate_test[n, 0], rate_test[n, 1], normalized = 0)
 # print(pred, rate_test[n, 2])
 SE += (pred - rate_test[n, 2])**2

RMSE = np.sqrt(SE/n_tests)
print('User-user CF, RMSE =', RMSE)

[19]: ✓ 1.4s
...
User-user CF, RMSE = 1.0369740376881258
```

**Hình 6.13:** Giá trị RMSE

Dưới đây là kết quả chạy Collaborative đối với dữ liệu trên:

**Recommendation:**

```

Recommend item 0 to user(s) : []
Recommend item 1 to user(s) : [1]
Recommend item 2 to user(s) : [0]
Recommend item 3 to user(s) : [5]
Recommend item 4 to user(s) : [3, 4, 5]

```

**Hình 6.14:** Kết quả gợi ý item cho user

Có thể thấy kết quả chạy đúng như mong muốn như ở 6.12.

Trong chương này, em đã trình bày những vấn đề mà em đã gặp phải và những giải pháp cụ thể mà em đã áp dụng để giải quyết chúng. Những kinh nghiệm này không chỉ giúp em vượt qua các thách thức trong quá trình làm đồ án tốt nghiệp mà còn giúp em phát triển kỹ năng phân tích, tự giải quyết vấn đề và lập luận logic.

## CHƯƠNG 7. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Chương này em sẽ giới thiệu về quá trình thực hiện và phát triển hệ thống giới thiệu phim ảnh. Em sẽ trình bày các bước và quy trình đã thực hiện để xây dựng sản phẩm từ các ý tưởng ban đầu đến việc triển khai và thử nghiệm trên môi trường local. Bên cạnh đó, chương cũng nhấn mạnh các mục tiêu và những thách thức trong quá trình này, cùng với những kết quả đạt được và học hỏi được từ dự án.

### Kết quả so sánh

| 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        |

**Hình 7.1:** Kết quả so sánh so với các ứng dụng khác

Nhìn vào hình 7.1 có thể thấy ứng dụng hiện tại đã giải quyết được hết những vấn đề ban đầu đặt ra.

### Phân tích quá trình thực hiện

Trong suốt quá trình thực hiện ĐATN, em đã hoàn thành được những mục tiêu chính sau:

1. Lấy được tất cả những dữ liệu cần thiết phục vụ cho việc xây dựng ứng dụng. Để làm được mục tiêu này, em đã tìm rất nhiều tài liệu riêng về chủ đề crawl để có thể nghiên cứu và áp dụng được hoàn toàn vào đồ án của bản thân.
2. Hoàn thành chức năng chính của ứng dụng. Em đã thêm được rất nhiều chức năng mà bản thân đã nghĩ khó có thể làm, nhờ đó em có thể biết được nhiều công nghệ, kiến thức hay ho hơn để áp dụng vào những dự án trong tương lai.
3. Triển khai và thử nghiệm ứng dụng trên môi trường local.

### Những điểm chưa làm được

1. Chưa triển khai trên server thực tế hoặc môi trường cloud. Hiện tại ứng dụng chỉ chạy trên môi trường local, chưa được kiểm thử với lượng người dùng lớn

hoặc trong điều kiện tải cao.

- Ứng dụng đang còn đơn giản. Nhìn qua, ứng dụng chỉ có vài tính năng CRUD đơn giản, không quá phức tạp.

### **Đóng góp nổi bật**

- Thiết kế và triển khai ứng dụng được thiết kế ngay từ đầu: Em đã hoàn thành các chức năng cơ bản và đảm bảo ứng dụng hoạt động ổn định lâu dài.
- Dữ liệu đảm bảo, nhất quán, đầy đủ. Em đã crawl được bộ dữ liệu hoàn chỉnh phục vụ cho mục đích xây dựng ứng dụng của bản thân.
- Xây dựng được 2 hệ thống gọi ý riêng biệt áp dụng vào trang home và trang detail.
- Cấu trúc mã nguồn của em cho phép mở rộng và có thể phát triển các tính năng nâng cao.

**Bài học kinh nghiệm cho bản thân** Trong quá trình thực hiện đồ án, em đã rút ra được rất nhiều bài học như:

- Cách quản lý thời gian: Việc lập kế hoạch chi tiết và quản lý thời gian hiệu quả giúp hoàn thành các mục tiêu đúng tiến độ. Vì hiểu được điều này, các buổi báo cáo với cô giáo diễn ra rất suôn sẻ và không bị chậm trễ.
- Khả năng học hỏi và tự nghiên cứu: Khả năng tự học là rất quan trọng, bởi trên nhà trường, chỉ học lý thuyết và nhiệm vụ của chúng em là về nhà tìm tòi, mà mò thực hành thêm để ghi nhớ những kiến thức đó. Chỉ có tự ôn tập, nghiên cứu mới có thể khiến bản thân tiến bộ hơn.
- Thu thập thông tin: Em rút ra được bài học rằng, đây là bước rất quan trọng trước khi xây dựng một ứng dụng. Chúng ta phải biết người dùng muốn gì, chúng ta mới xây dựng được một ứng dụng thành công.

Hiện tại, em nghĩ các công việc cần thiết để hoàn thiện sản phẩm của mình là:

- Đầu tiên là triển khai trên môi trường thực tế. Ứng dụng của em hiện tại đã phân chia ra đúng các gói, các lớp, các thành phần và đó là cơ sở để có thể dễ dàng deploy trong tương lai. Đây là một nhiệm vụ cần thiết để có thể phổ biến được ứng dụng của bản thân.
- Em cần cải thiện về giao diện ứng dụng. Ứng dụng của em hiện tại đang có những tông màu rất đơn giản như đen, xanh, xanh lá cây. Để ứng dụng bắt mắt, được sự chú ý của người khác hơn thì em nghĩ nên cải thiện nhiều hơn về giao diện ứng dụng.

3. Tích hợp thêm tính năng phân tích và thống kê: Hiện tại ứng dụng chỉ có thống kê số lượng đánh giá. Mà người dùng thì lại rất quan tâm đến số liệu để xem phim theo số đông, nêm em nghĩ đây là bước cần thiết hiện tại.

Trong tương lai, em muốn phát triển thêm vài chức năng khác như:

1. Phát triển ứng dụng trên thiết bị di động. Điện thoại di động là thiết bị không thể thiếu đối với mọi người, vậy nên phát triển ứng dụng trên thiết bị di động trong tương lai là một việc rất quan trọng.
2. Tích hợp thêm hệ thống dự đoán doanh thu phim mới. Đây là một chức năng mà em đã định hướng từ trước, nhưng do dữ liệu em triển khai trên hệ thống là không đủ, nên em muốn trong tương lai, em có thể lấy nhiều dữ liệu hơn và phát triển tính năng này

Trên đây là kết luận và hướng phát triển trong tương lai của em. Em sẽ luôn cố gắng hết sức mình để hoàn thiện sản phẩm của mình một cách tốt nhất, em xin chân thành cảm ơn!

## TÀI LIỆU THAM KHẢO

- [1] Microsoft, *Visual studio code (vscode)*, <https://code.visualstudio.com/>, Truy cập lần cuối vào ngày 01/02/2024.
- [2] Django Software Foundation, *Django*, <https://www.djangoproject.com/>, Truy cập lần cuối vào ngày 01/06/2024.
- [3] Tom Christie, *Django rest framework*, <https://www.django-rest-framework.org/>, Truy cập lần cuối vào ngày 01/06/2024.
- [4] *Html*, <https://developer.mozilla.org/en-US/docs/Web/HTML>, Truy cập lần cuối vào ngày 15/04/2024.
- [5] *Css*, <https://developer.mozilla.org/en-US/docs/Web/CSS>, Truy cập lần cuối vào ngày 15/04/2024.
- [6] *Bootstrap*, <https://getbootstrap.com/>, Truy cập lần cuối vào ngày 15/04/2024.
- [7] *Jquery*, <https://jquery.com/>, Truy cập lần cuối vào ngày 15/05/2024.
- [8] *Git*, <https://git-scm.com/>, Truy cập lần cuối vào ngày 01/04/2024.
- [9] *Github*, <https://github.com/>, Truy cập lần cuối vào ngày 30/06/2024.
- [10] *Selenium*, <https://www.selenium.dev/>, Truy cập lần cuối vào ngày 01/02/2024.
- [11] *Requests*, <https://requests.readthedocs.io/>, Truy cập lần cuối vào ngày 01/03/2024.
- [12] *Github actions*, <https://github.com/features/actions>, Truy cập lần cuối vào ngày 30/06/2024.
- [13] *Mysql*, <https://www.mysql.com/>, Truy cập lần cuối vào ngày 01/06/2024.

# **PHỤ LỤC**