

ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

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

TRẦN HỒNG QUÂN

quan.th240818E@sis.hust.edu.vn

Ngành Kỹ sư chuyên sâu Trí tuệ nhân tạo tạo sinh

Giảng viên hướng dẫn: TS. Đặng Tuấn Linh

Chữ ký GVHD

Khoa: Kỹ thuật máy tính

Trường: Công nghệ Thông tin và Truyền thông

HÀ NỘI, 01/2026

LỜI CẢM ƠN

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

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

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

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

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

TÓM TẮT NỘI DUNG ĐỒ ÁN

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

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

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

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

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

MỤC LỤC

| | |
|---|----------|
| CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI..... | 1 |
| 1.1 Đặt vấn đề..... | 1 |
| 1.2 Mục tiêu và phạm vi đề tài..... | 2 |
| 1.3 Định hướng giải pháp..... | 3 |
| 1.4 Bố cục đồ án | 3 |
| CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU..... | 5 |
| 2.1 Khảo sát hiện trạng | 5 |
| 2.2 Khảo sát về quyền theo dõi | 5 |
| 2.3 Tổng quan chức năng | 8 |
| 2.3.1 Biểu đồ use case tổng quát | 8 |
| 2.3.2 Biểu đồ use case phân rã Quản lý thông tin người dùng | 10 |
| 2.3.3 Biểu đồ use case phân rã Quản lý thông tin phim | 11 |
| 2.3.4 Biểu đồ use case phân rã Quản lý tài khoản | 12 |
| 2.3.5 Biểu đồ use case phân rã Tương tác với phim..... | 12 |
| 2.3.6 Biểu đồ use case phân rã bình luận, đánh giá phim..... | 13 |
| 2.4 Đặc tả chức năng | 14 |
| 2.4.1 Đặc tả use case Đăng nhập..... | 14 |
| 2.4.2 Đặc tả use case Chính sửa thông tin cá nhân | 15 |
| 2.4.3 Đặc tả use case Tìm kiếm phim | 16 |
| 2.4.4 Đặc tả use case Đánh dấu phim mình thích..... | 17 |
| 2.4.5 Đặc tả use case Đánh giá phim | 17 |
| 2.4.6 Đặc tả use case quản lý phim..... | 18 |
| 2.4.7 Đặc tả use case Gợi ý phim theo nội dung | 19 |
| 2.5 Yêu cầu phi chức năng | 19 |

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

| | |
|---|----|
| 3.1 Lý do lựa chọn Content-Based Filtering cho ứng dụng gợi ý phim..... | 21 |
| 3.2 Ứng dụng Content-Based Filtering trong hệ thống gợi ý phim..... | 22 |
| 3.2.1 Vai trò và mục đích sử dụng | 22 |
| 3.2.2 Cách thức hoạt động trong hệ thống..... | 23 |
| 3.2.3 Vị trí sử dụng trong ứng dụng..... | 23 |
| 3.3 So sánh các thuật toán trong Content-Based Filtering..... | 25 |
| 3.3.1 So sánh TF-IDF và các phương pháp trích xuất đặc trưng khác..... | 25 |
| 3.3.2 So sánh Cosine Similarity với các phương pháp đo độ tương đồng khác | 25 |
| 3.3.3 Lựa chọn thuật toán cho hệ thống | 26 |

CHƯƠNG 4. CÔNG NGHỆ SỬ DỤNG..... 28

| | |
|--|----|
| 4.1 Công nghệ sử dụng phía frontend | 28 |
| 4.1.1 React | 28 |
| 4.1.2 Tailwind CSS | 28 |
| 4.2 Công nghệ sử dụng phía backend | 29 |
| 4.2.1 Django | 29 |
| 4.2.2 MySQL..... | 29 |
| 4.3 Công nghệ sử dụng trong việc lưu trữ code | 30 |
| 4.3.1 Git | 30 |
| 4.3.2 GitHub..... | 31 |
| 4.4 Công nghệ sử dụng trong việc crawl và update dữ liệu | 31 |
| 4.4.1 Selenium và Requests | 31 |
| 4.4.2 Công nghệ sử dụng để cập nhật dữ liệu | 32 |

| | |
|---|-----------|
| CHƯƠNG 5. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG | 34 |
| 5.1 Thiết kế kiến trúc..... | 34 |
| 5.1.1 Lựa chọn kiến trúc phần mềm | 34 |
| 5.1.2 Thiết kế tổng quan..... | 35 |
| 5.1.3 Thiết kế chi tiết gói | 37 |
| 5.2 Thiết kế chi tiết..... | 41 |
| 5.2.1 Thiết kế giao diện | 41 |
| 5.2.2 Thiết kế cơ sở dữ liệu hệ thống | 47 |
| 5.2.3 Thiết kế hệ thống crawl dữ liệu..... | 48 |
| 5.2.4 Tìm kiếm phim bằng ngôn ngữ tự nhiên | 49 |
| 5.2.5 Gợi ý phim bằng genAI dựa vào lịch sử hoạt động của người dùng . | 51 |
| 5.3 Xây dựng ứng dụng..... | 54 |
| 5.3.1 Thư viện và công cụ sử dụng | 54 |
| 5.3.2 Kết quả đạt được | 54 |
| 5.3.3 Minh họa các chức năng chính | 57 |
| 5.4 Kiểm thử..... | 60 |
| 5.5 Triển khai | 64 |
| CHƯƠNG 6. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT | 66 |
| 6.1 Sự thiếu hụt trầm trọng về dữ liệu điện ảnh | 66 |
| 6.2 Thách thức trong việc thiết kế cơ sở dữ liệu với hệ thống thuộc tính phức tạp | 66 |
| 6.3 Trang web có dữ liệu update liên tục theo ngày khiến link ảnh và trailer bị hỏng..... | 66 |
| 6.4 Lựa chọn thuật toán gợi ý phù hợp..... | 69 |
| CHƯƠNG 7. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN | 73 |
| TÀI LIỆU THAM KHẢO..... | 76 |
| PHỤ LỤC..... | 78 |

CHƯƠNG A. PHỤ LỤC A..... 78

| | |
|---|----|
| A.1 Ngành học..... | 79 |
| A.2 Đánh dấu (bullet) và đánh số (numbering) | 79 |
| A.3 Cách thêm bảng | 80 |
| A.4 Chèn hình ảnh | 80 |
| A.5 Tài liệu tham khảo | 81 |
| A.6 Cách viết phương trình và công thức toán học..... | 81 |
| A.7 Qui cách đóng quyển..... | 81 |

CHƯƠNG B. PHỤ LỤC B..... 83

| | |
|--|----|
| B.1 Đặc tả use case “Thông kê tình hình mượn sách” | 83 |
| B.2 Đặc tả use case “Đăng ký làm thẻ mượn” | 83 |

DANH MỤC HÌNH VẼ

| | | |
|-----------|---|----|
| Hình 2.1 | Form khảo sát người dùng về các ứng dụng | 6 |
| Hình 2.2 | Khảo sát người dùng về các ứng dụng | 7 |
| Hình 2.3 | Form khảo sát về quyền theo dõi | 8 |
| Hình 2.4 | Khảo sát về quyền theo dõi | 9 |
| Hình 2.5 | Use case tổng quan | 10 |
| Hình 2.6 | Phân rã use case Quản lý thông tin người dùng | 10 |
| Hình 2.7 | Phân rã use case Quản lý thông tin phim | 11 |
| Hình 2.8 | Phân rã use case Quản lý tài khoản | 12 |
| Hình 2.9 | Phân rã use case Tương tác với phim | 12 |
| Hình 2.10 | Phân rã use case bình luận, đánh giá phim | 13 |
| | | |
| Hình 3.1 | So sánh Content-Based Filtering với Collaborative Filtering . | 22 |
| Hình 3.2 | Quy trình Content-Based Filtering | 23 |
| Hình 3.3 | So sánh Cosine Similarity và Euclidean Distance | 26 |
| | | |
| Hình 5.1 | Biểu đồ phụ thuộc gói | 35 |
| Hình 5.2 | Thiết kế chi tiết gói ở tầng Model | 37 |
| Hình 5.3 | Thiết kế chi tiết gói ở tầng View | 39 |
| Hình 5.4 | Thiết kế chi tiết gói Template | 41 |
| Hình 5.5 | Thiết kế màn hình <i>login</i> | 42 |
| Hình 5.6 | Thiết kế màn hình <i>signup</i> | 43 |
| Hình 5.7 | Thiết kế màn hình <i>user profile</i> | 43 |
| Hình 5.8 | Thiết kế màn hình <i>home</i> | 44 |
| Hình 5.9 | Thiết kế màn hình <i>gợi ý phim ở trang home</i> | 44 |
| Hình 5.10 | Thiết kế màn hình <i>detail 1</i> | 45 |
| Hình 5.11 | Thiết kế màn hình <i>detail 2</i> | 45 |
| Hình 5.12 | Thiết kế màn hình <i>cast and crew</i> | 46 |
| Hình 5.13 | Thiết kế màn hình <i>user review</i> | 46 |
| Hình 5.14 | Thiết kế sơ đồ ERD của hệ thống | 47 |
| Hình 5.15 | Hình ảnh các bảng trong cơ sở dữ liệu 1 | 47 |
| Hình 5.16 | Hình ảnh các bảng trong cơ sở dữ liệu 2 | 48 |
| Hình 5.17 | Hình ảnh các bảng trong cơ sở dữ liệu 3 | 48 |
| Hình 5.18 | Màn hình <i>tìm kiếm phim</i> | 49 |
| Hình 5.19 | Màn hình <i>gợi ý phim bằng genAI</i> | 52 |
| Hình 5.20 | Chức năng tìm kiếm phim theo ngôn ngữ người dùng | 55 |
| Hình 5.21 | Review phim | 56 |

| | |
|--|----|
| Hình 5.22 Gợi ý phim theo lịch sử review | 56 |
| Hình 5.23 Tìm kiếm phim | 57 |
| Hình 5.24 Gợi ý phim theo lịch sử tìm kiếm | 57 |
| Hình 5.25 Chức năng gợi ý phim dựa vào lịch sử thao tác của người dùng | 58 |
| Hình 5.26 Chức năng gợi ý phim dựa vào nội dung phim đang xem | 58 |
| Hình 5.27 Chức năng thêm phim yêu thích vào danh sách | 59 |
| Hình 5.28 Chức năng review phim | 59 |
| Hình 5.29 Chức năng xem toàn bộ ảnh của phim | 60 |
| | |
| Hình 6.1 Tạo PAT Github | 67 |
| Hình 6.2 Thiết lập Secrets trên GitHub | 68 |
| Hình 6.3 Hệ thống tự động crawl dữ liệu | 68 |
| Hình 6.4 Cập nhật dữ liệu mới thành công | 69 |
| Hình 6.5 Kiểm tra kết quả 1 | 70 |
| Hình 6.6 Kiểm tra kết quả 2 | 71 |
| Hình 6.7 Kiểm tra kết quả 3 | 72 |
| | |
| Hình A.1 Internet vạn vật | 80 |
| Hình A.2 Qui cách đóng quyền đồ án | 82 |
| Hình A.3 Qui cách đóng quyền đồ án | 82 |

DANH MỤC BẢNG BIỂU

| | | |
|----------|--|----|
| Bảng 2.1 | Khảo sát dựa trên các hệ thống giới thiệu phim hiện có | 6 |
| Bảng 2.2 | Đặc tả use case Đăng nhập | 14 |
| Bảng 2.3 | Đặc tả use case Chính sửa thông tin cá nhân | 15 |
| Bảng 2.4 | Đặc tả use case Tìm kiếm phim | 16 |
| Bảng 2.5 | Đặc tả use case Đánh dấu phim mình thích | 17 |
| Bảng 2.6 | Đặc tả use case Đánh giá phim | 17 |
| Bảng 2.7 | Đặc tả use case quản lý phim | 18 |
| Bảng 2.8 | Đặc tả use case Gợi ý phim theo nội dung | 19 |
| | | |
| Bảng 5.1 | Tổng hợp các thông tin cần lấy | 48 |
| Bảng 5.2 | Danh sách thư viện và công cụ sử dụng | 54 |
| Bảng 5.3 | Kiểm thử, đánh giá chức năng 1 | 61 |
| Bảng 5.4 | Kiểm thử, đánh giá chức năng 2 | 62 |
| Bảng 5.5 | Bảng danh sách kiểm thử hiệu năng | 63 |
| Bảng 5.6 | Bảng kết quả kiểm thử hiệu năng | 64 |
| | | |
| Bảng 7.1 | Kết quả so sánh so với các ứng dụng khác | 73 |
| | | |
| Bảng A.1 | Table to test captions and labels. | 80 |

DANH MỤC THUẬT NGỮ VÀ TỪ VIỆT TẮT

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

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Đặt vấn đề

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

Trong bối cảnh đó, Generative AI (genAI) - trí tuệ nhân tạo tạo sinh - đang nổi lên như một công nghệ tiềm năng để giải quyết những thách thức trên. GenAI là công nghệ AI có khả năng hiểu và xử lý ngôn ngữ tự nhiên một cách thông minh, cho phép hệ thống hiểu được ý định người dùng thông qua các truy vấn bằng ngôn ngữ tự nhiên thay vì từ khóa cứng nhắc. Điều này mở ra khả năng tìm kiếm và gợi ý phim một cách tự nhiên và chính xác hơn.

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

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

Để giải quyết những hạn chế này, việc tích hợp genAI vào hệ thống gợi ý phim

là một hướng tiếp cận đầy hứa hẹn. Bằng cách tận dụng khả năng xử lý ngôn ngữ tự nhiên của genAI, hệ thống có thể hiểu các truy vấn phức tạp của người dùng, từ đó đưa ra các gợi ý chính xác và phù hợp hơn so với các phương pháp tìm kiếm truyền thống. Khả năng học từ hành vi người dùng của genAI cũng cho phép hệ thống cải thiện chất lượng gợi ý theo thời gian, tạo ra trải nghiệm cá nhân hóa ngày càng tốt hơn. Do đó, việc xây dựng một hệ thống gợi ý phim tích hợp genAI sẽ không chỉ khắc phục các hạn chế của các hệ thống hiện có mà còn mang lại giá trị gia tăng đáng kể cho người dùng.

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

Việc xây dựng một hệ thống như vậy sẽ đáp ứng nhu cầu ngày càng cao của người dùng, đồng thời mở ra cơ hội ứng dụng rộng rãi trong lĩnh vực giải trí nói chung và mở rộng ứng dụng nói riêng, góp phần thúc đẩy sự phát triển của ngành công nghiệp giải trí và các ngành liên quan.

1.2 Mục tiêu và phạm vi đề tài

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

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

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

1.3 Định hướng giải pháp

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

- **Tìm kiếm phim bằng ngôn ngữ người dùng:** Sử dụng AI để phân tích truy vấn, kết hợp vector search dựa trên embeddings và lọc có cấu trúc để trả về kết quả chính xác.
- **Gợi ý phim cá nhân hóa:** Theo dõi hành động người dùng để gợi ý phim cá nhân hóa.
- **Dữ liệu đầy đủ:** Thu thập và tổng hợp thông tin phim ảnh từ các nguồn uy tín để xây dựng cơ sở dữ liệu phong phú, đa dạng và chính xác.
- **Hiệu năng trang web tốt:** Cải thiện hiệu năng của hệ thống bằng cách rút gọn mã nguồn, thông tin, tối ưu hóa các thuật toán và sử dụng các kỹ thuật lập trình hiệu quả nhất, đảm bảo tốc độ tải trang nhanh và hiệu quả.
- **Có hệ thống gợi ý phim:** Sử dụng thuật toán cho hệ thống gợi ý phim: Content-Based Filtering. Content-Based Filtering sẽ được áp dụng khi người dùng xem thông tin về một bộ phim cụ thể, từ đó gợi ý các bộ phim có nội dung tương tự.
- **Tích hợp các chức năng khác:** Phát triển một nền tảng web tích hợp các tính năng khác như xem chi tiết phim, gợi ý phim cá nhân hóa, đánh giá và review từ cộng đồng, cùng với việc xem trailer và hình ảnh phim.

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

1.4 Bố cục đồ án

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

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

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

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

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

Chương 6 tập trung làm rõ các giải pháp đã được áp dụng và những đóng góp quan trọng trong suốt quá trình phát triển dự án. Ngoài ra, chương cũng thực hiện phân tích toàn diện và chi tiết về bài toán gợi ý phim.

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

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

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

2.1 Khảo sát hiện trạng

Để thực hiện khảo sát hiện trạng và xác định các yêu cầu cho ứng dụng gợi ý phim Hình 2.1, tôi đã tiến hành thu thập dữ liệu từ ba nguồn thông tin chính bao gồm: phản hồi từ người dùng và khách hàng, đánh giá các hệ thống hiện có trên thị trường, và nghiên cứu các ứng dụng tương tự. Dựa trên nguồn dữ liệu đã thu thập, tôi tiến hành phân tích một cách toàn diện, thực hiện so sánh giữa các giải pháp và đánh giá kỹ lưỡng các mặt ưu điểm cũng như hạn chế của từng sản phẩm và nghiên cứu liên quan.

Kết quả khảo sát từ phía người dùng và khách hàng Hình 2.2 cho thấy, đối tượng này kỳ vọng một ứng dụng có thể thực hiện tìm kiếm phim với tốc độ cao và độ chính xác tốt, kèm theo đó là hệ thống gợi ý phim được cá nhân hóa dựa trên sở thích riêng và lịch sử xem phim của từng người. Bên cạnh đó, các chức năng như đánh giá phim, viết nhận xét và xem trailer cũng nhận được sự quan tâm đáng kể từ phía người dùng. Ngoài ra, việc nâng cao hiệu năng hệ thống và thiết kế giao diện thân thiện, dễ sử dụng cũng là những yêu cầu quan trọng được đề xuất.

2.2 Khảo sát về quyền theo dõi

Theo kết quả của khảo sát về quyền theo dõi Hình 2.4, tôi nhận thấy rằng người dùng mong muốn một ứng dụng gợi ý phim, nhưng phần lớn lại không mong muốn bị theo dõi hành vi trên ứng dụng. Vì vậy, tôi sẽ phát triển thêm chức năng theo dõi hành vi người dùng để gợi ý phim một cách phù hợp nhất với mong muốn người dùng.

Bạn tên gì ? *

Câu trả lời của bạn

Bạn có thích xem phim không ? *

Có
 Không

Bạn có thường xuyên tìm kiếm phim trên các hệ thống không ? *

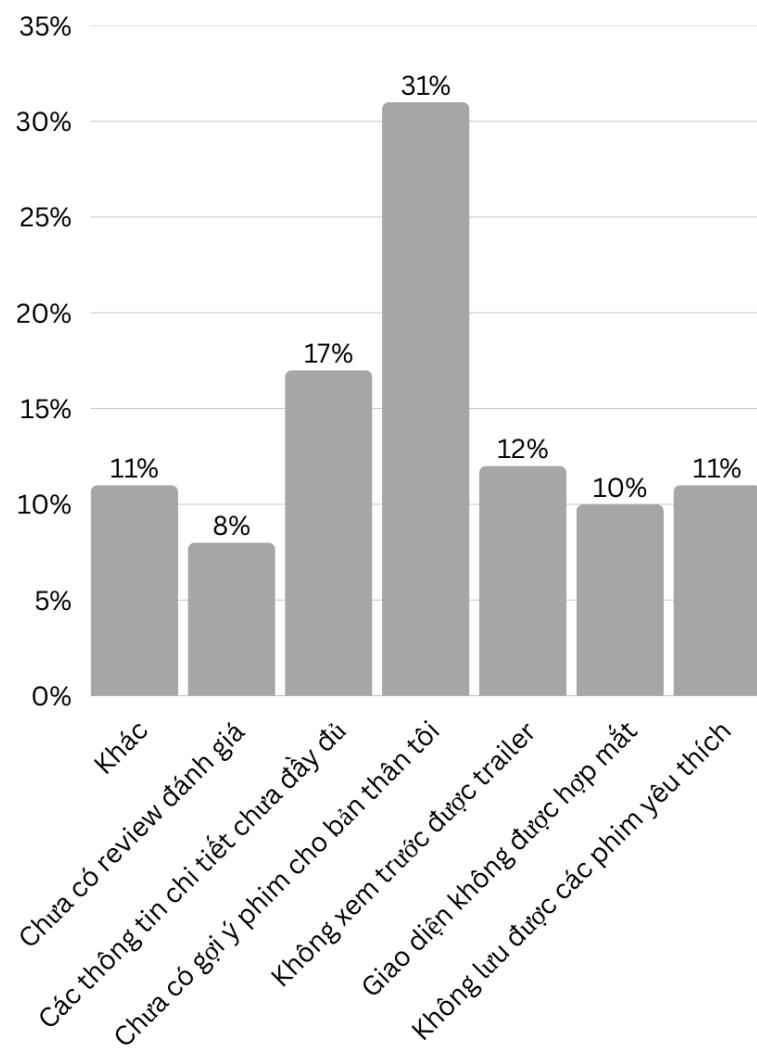
Có
 Không

Trong các hệ thống giới thiệu phim ảnh mà bạn đã từng tham khảo, thì bạn thấy chức năng nào bạn cần mà hệ thống đó không có ? *

Chưa có review đánh giá
 Các thông tin chi tiết chưa đầy đủ
 Chưa có gợi ý phim cho bản thân tôi
 Không xem trước được trailer
 Giao diện không được hợp mắt
 Không lưu được các phim yêu thích
 Khác

Hình 2.1: Form khảo sát người dùng về các ứng dụng**Bảng 2.1:** Khảo sát dựa trên các hệ thống giới thiệu phim hiện có

| Tính năng | IMDb | Metacritic | Rotten Tomatoes | TMDb | Box Office Mojo | The Numbers |
|----------------|--------|------------|-----------------|--------------------------------|-----------------|--------------|
| Thông tin phim | Đầy đủ | Đầy đủ | Đầy đủ | Đầy đủ nhưng phải lấy bằng API | Cơ bản | 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 người dùng về các ứng dụng

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

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

Bạn tên gì ? *

Câu trả lời của bạn

Bạn có thích xem phim không ? *

Có
 Không

Bạn có thường xuyên tìm kiếm phim trên các hệ thống không ? *

Có
 Không

Theo bạn, nếu trong một ứng dụng phim ảnh cần theo dõi hành động của bạn để gợi ý phim phù hợp nhất với sở thích thì bạn có đồng ý không ? *

Không theo dõi hành vi người dùng
 Tự động theo dõi hành vi người dùng
 Theo dõi hành vi người dùng có cam kết

Hình 2.3: Form khảo sát về quyền theo dõi

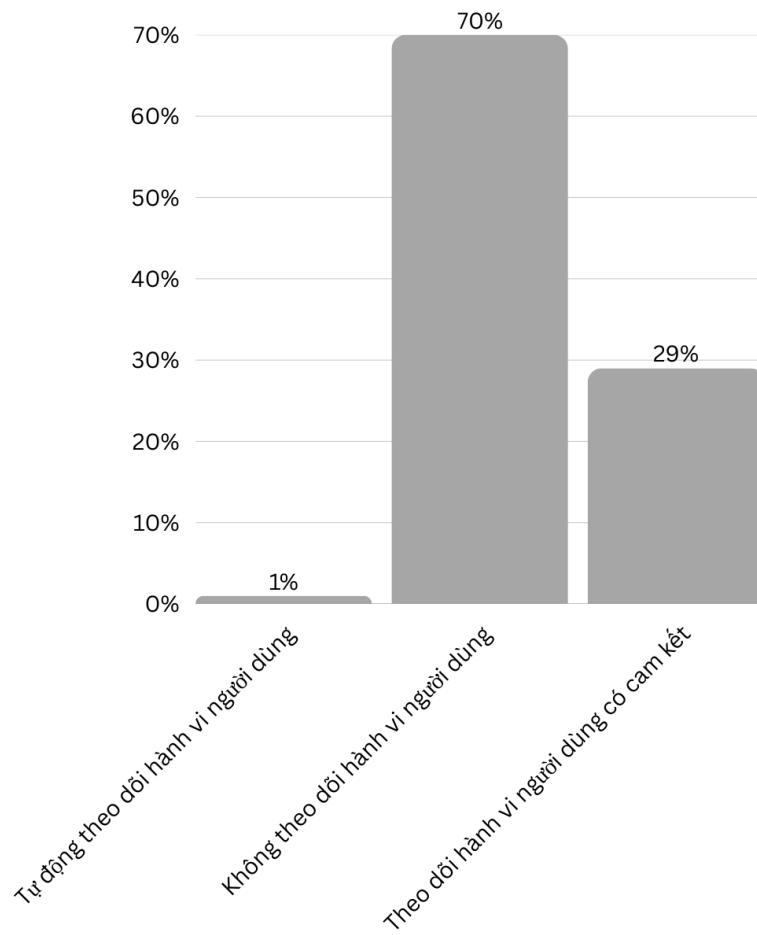
Tổng hợp từ các kết quả khảo sát đã thu thập được, các tính năng quan trọng cần được tích hợp vào hệ thống đã được xác định một cách rõ ràng, nhằm mục tiêu phát triển một giải pháp gợi ý phim toàn diện có khả năng đáp ứng một cách hiệu quả các nhu cầu đa dạng của người dùng.

2.3 Tổng quan chức năng

2.3.1 Biểu đồ use case tổng quát

Các tác nhân tham gia:

- Guest (Khách truy cập): Đối tượng chưa đăng ký thành viên, thực hiện truy cập ứng dụng nhằm mục đích tra cứu và tham khảo các dữ liệu liên quan đến phim ảnh.
- User (Thành viên): Những cá nhân đã xác thực tài khoản, được phép thực hiện các thao tác như gửi phản hồi về phim, quản lý danh mục phim ưa thích, cập nhật hồ sơ cá nhân và sử dụng đầy đủ các tiện ích mở rộng khác.
- Admin (Quản trị viên): Cá nhân nắm quyền điều hành cao nhất, thực hiện giám sát toàn diện và kiểm soát mọi dữ liệu cũng như hoạt động diễn ra trên nền tảng.

**Hình 2.4:** Khảo sát về quyền theo dõi

Vai Trò của Mối Tác Nhân:

- Guest: Vai trò chủ đạo của đối tượng này là tra cứu dữ liệu điện ảnh và thưởng thức các tác phẩm trực tuyến mà không yêu cầu thực hiện thao tác đăng nhập.
- User: Ngoài việc sở hữu các tính năng như Guest, thành viên này còn được cấp quyền bình luận, lưu trữ phim vào mục quan tâm và điều chỉnh hồ sơ cá nhân.
- Admin: Đây là chủ thể điều hành toàn bộ hệ thống, chịu trách nhiệm kiểm soát danh mục phim cũng như giám sát các hoạt động của người dùng.

Các use case chính:

- Truy xuất dữ liệu điện ảnh: Cả hai đối tượng Guest và User đều được phép tham khảo các thông số chi tiết của tác phẩm như cốt truyện, ê-kíp sản xuất, dàn sao, giải thưởng, chỉ số doanh thu và những nhận xét từ cộng đồng.
- Phản hồi và lưu trữ: Thành viên hệ thống có khả năng trực tiếp chấm điểm cho tác phẩm cũng như cá nhân hóa kho lưu trữ bằng cách thêm phim vào mục yêu

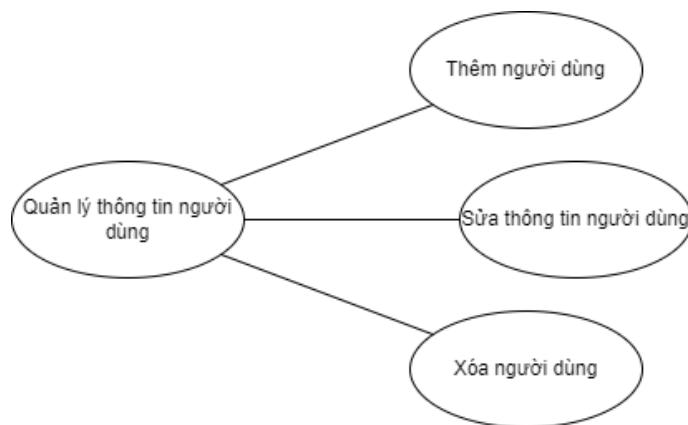
thích.

- Đề xuất nội dung: Dựa trên lịch sử truy cập và các lượt xếp hạng từ phía người dùng, nền tảng sẽ tự động hiển thị danh sách các bộ phim phù hợp với sở thích cá nhân.



Hình 2.5: Use case tổng quan

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

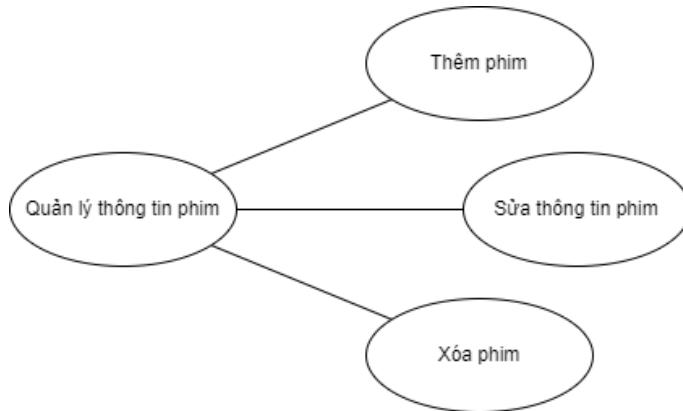


Hình 2.6: Phân rã use case Quản lý thông tin người dùng

Trong sơ đồ use case phân rã "Quản lý thông tin người dùng", các thành phần con được chia tách và giải thích cụ thể 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.
- Chỉnh sửa hồ sơ thành viên: Quản trị viên được quyền thay đổi các chi tiết cá nhân của người dùng, bao gồm tên truy cập, địa chỉ email cũng như mật khẩu hệ thống.
- Loại bỏ thành viên: Quản trị viên có quyền hạn gỡ bỏ tài khoản người dùng ra khỏi hệ thống.

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

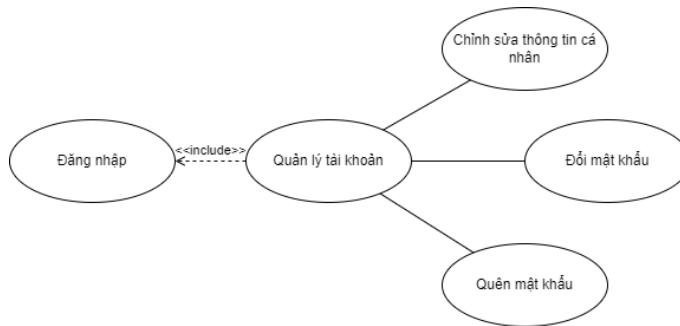


Hình 2.7: Phân rã use case Quản lý thông tin phim

Trong sơ đồ Use Case chi tiết thuộc phân hệ "Quản lý thông tin phim", các chức năng con được phân tách và diễn giải cụ thể như sau:

- Cập nhật phim mới: Quản trị viên có quyền thiết lập các bộ phim mới trên hệ thống thông qua việc cung cấp đầy đủ các dữ liệu như tên phim, tóm tắt, chủng loại, đạo diễn, dàn diễn viên, thời điểm phát hành, ảnh minh họa, ...
- Chỉnh sửa dữ liệu điện ảnh: Admin được phép thay đổi những thông số chi tiết của tác phẩm đã có sẵn, bao gồm các thành phần về thể loại, người dàn dựng, diễn viên tham gia và mô tả kịch bản.
- Gỡ bỏ phim: Người quản lý hệ thống có thẩm quyền xóa vĩnh viễn một đầu phim ra khỏi cơ sở dữ liệu.

2.3.4 Biểu đồ use case phân rã Quản lý tài khoản

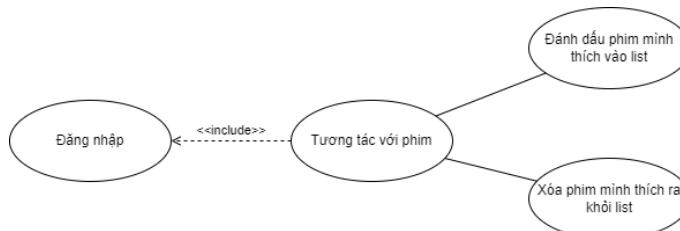


Hình 2.8: Phân rã use case Quản lý tài khoản

Trong biểu đồ Use case phân rã "Quản lý tài khoản", các chức năng con được phân tách và diễn giải cụ thể như sau:

- Quên mật khẩu: Thành viên có thể thiết lập lại mã bảo mật thông qua việc yêu cầu hệ thống gửi đường dẫn thay đổi mật khẩu tới hộp thư điện tử cá nhân.
- Đổi mật khẩu: Người dùng được phép thay thế mật khẩu đang sử dụng bằng cách xác nhận lại mật khẩu cũ và thiết lập chuỗi ký tự mới.
- Thay đổi hồ sơ cá nhân: Người dùng có quyền tự điều chỉnh các chi tiết về bản thân như họ tên, địa chỉ email, hình ảnh đại diện và các thông tin liên quan khác.

2.3.5 Biểu đồ use case phân rã Tương tác với phim

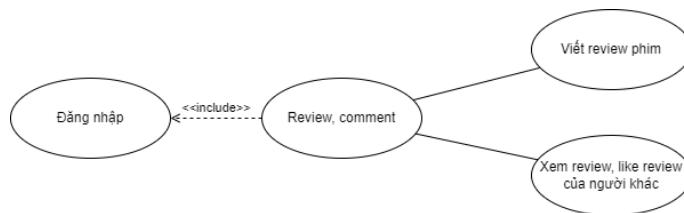


Hình 2.9: Phân rã use case Tương tác với phim

Trong biểu đồ use case phân rã "Tương tác với phim", các chức năng con được phân tách và diễn giải cụ thể như sau:

- Lưu phim vào mục yêu thích: Thành viên có thể thêm các tác phẩm điện ảnh vào bộ sưu tập cá nhân nhằm thuận tiện cho việc tìm kiếm lại, đồng thời giúp hệ thống đề xuất các nội dung tương đồng phù hợp với sở thích.
- Loại bỏ phim khỏi danh sách: Người dùng được phép gỡ bỏ những bộ phim đã lưu trước đó ra khỏi mục yêu thích của mình.

2.3.6 Biểu đồ use case phân rã bình luận, đánh giá phim



Hình 2.10: Phân rã use case bình luận, đánh giá phim

Trong biểu đồ use case phân rã "bình luận, đánh giá phim", các chức năng con được phân tách và diễn giải cụ thể như sau:

- Đánh giá phim: Sau khi thưởng thức, thành viên có thể soạn thảo và chia sẻ nhận xét về phim lên hệ thống để thảo luận cũng như đóng góp quan điểm cá nhân với cộng đồng.
- Tương tác bình luận: Người dùng có quyền biểu đạt sự tán thành hoặc khích lệ bằng cách nhấn nút "Thích" cho những bài review mà họ cảm thấy tâm đắc hoặc có giá trị tham khảo cao.

2.4 Đặc tả chức năng

2.4.1 Đặc tả use case Đăng nhập

Bảng 2.2: Đặc tả use case Đăng nhập

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

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

2.4.2 ĐẶC TẢ USE CASE CHỈNH SỬA THÔNG TIN CÁ NHÂN

Bảng 2.3: 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 | | |
| Tiền điều kiện | Người dùng đã đăng nhập vào hệ thống và có thông tin cá nhân cần cập nhật. | | |
| Luồng sự kiện chính (Thành công) | STT | Thực hiện bởi | Hành động |
| | 1. | Người dùng | Người dùng truy cập vào trang cập nhật thông tin cá nhân |
| | 2. | Hệ thống | Hệ thống hiển thị form thông tin cá nhân hiện tại của người dùng. |
| | 3. | Người dùng | Người dùng thay đổi hoặc bổ sung thông tin cá nhân cần cập nhật trên form. |
| | 4. | Người dùng | Người dùng xác nhận thông tin đã nhập chính xác và yêu cầu cập nhật. |
| | 5. | Hệ thống | Hệ thống kiểm tra và lưu thông tin cá nhân mới của người dùng |
| | 6. | Hệ thống | Hệ thống hiển thị thông báo cập nhật thành công và cung cấp cho người dùng xem lại thông tin cá nhân đã cập nhật. |
| Luồng sự kiện thay thế | STT | Thực hiện bởi | Hành động |
| | 5a. | Hệ thống | Nếu người dùng nhập thông tin không hợp lệ, hệ thống hiển thị thông báo lỗi và yêu cầu người dùng sửa lại thông tin trước khi cập nhật. |
| | 5b. | Hệ thống | Nếu hệ thống gặp lỗi khi lưu thông tin cá nhân mới của người dùng, hệ thống hiển thị thông báo lỗi và yêu cầu người dùng thực hiện lại lần nữa sau khi đã khắc phục lỗi. |
| Hậu điều kiện | | | |

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

2.4.3 ĐẶC TẢ USE CASE TÌM KIẾM PHIM

Bảng 2.4: Đặc tả use case Tìm kiếm phim

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

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

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

2.4.4 ĐẶC TẢ USE CASE ĐÁNH DẤU PHIM MÌNH THÍCH

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

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

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

2.4.5 ĐẶC TẢ USE CASE ĐÁNH GIÁ PHIM

Bảng 2.6: Đặc tả use case Đánh giá phim

| | | | | |
|-------------------------------------|--|---------------|--|---------------|
| Mã Use case | UC008 | | Tên Use case | Đánh giá phim |
| Tác nhân | User | | | |
| Tiền điều kiện | Khách đăng nhập thành công với vai trò là User | | | |
| Luồng sự kiện chính (Thành công) | STT | Thực hiện bởi | Hành động | |
| | 1. | Người dùng | Chọn chức năng Đánh giá phim sau khi vào xem chi tiết phim | |
| Luồng sự kiện thay thế | 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 | |
| Hậu điều kiện | 4. | Hệ thống | Thông báo đánh giá thành công và quay về màn hình chi tiết phim đó | |
| | 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) | |

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

2.4.6 Đặc tả use case quản lý phim

Bảng 2.7: Đặc tả use case quản lý phim

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

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

2.4.7 Đặc tả use case Gợi ý phim theo nội dung

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

| Mã Use case | UC017 | | Tên Use case | Gợi ý phim theo nội dung |
|-------------------------------------|--------------------------|---------------|--|--------------------------|
| Tác nhân | Guest, User | | | |
| Tiền điều kiện | Xem chi tiết một bộ phim | | | |
| Luồng sự kiện chính (Thành công) | STT | Thực hiện bởi | Hành động | |
| | 1. | Người dùng | Nhấn vào một bộ phim để xem chi tiết | |
| Luồng sự kiện thay thế | 2. | Hệ thống | Truy xuất dữ liệu bộ phim đó trong database, chạy thuật toán content-based để lấy ra danh sách 10 bộ phim có độ tương đồng cao nhất với bộ phim đang xem và hiển thị ra màn hình | |
| | STT | Thực hiện bởi | Hành động | |
| Hậu điều kiện | Không | | | |

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

2.5 Yêu cầu phi chức năng

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

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

Tính dễ sử dụng: Giao diện người dùng cần được xây dựng với nguyên tắc đơn giản, trực quan và dễ hiểu, tạo điều kiện thuận lợi cho người dùng trong việc tìm kiếm thông tin và thực hiện các tương tác với hệ thống một cách tự nhiên và thuận

tiện nhất.

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

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

CHƯƠNG 3. CƠ SỞ LÝ THUYẾT CHO GỢI Ý PHIM

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

Khái niệm Content-Based Filtering:

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

3.1 Lý do lựa chọn Content-Based Filtering cho ứng dụng gợi ý phim

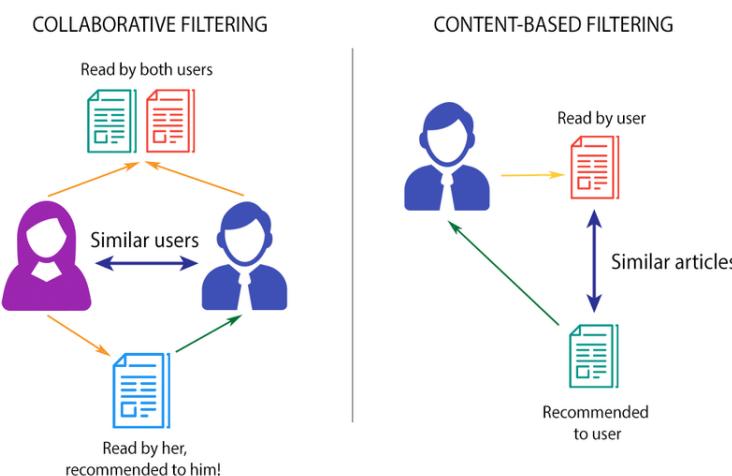
Việc lựa chọn Content-Based Filtering làm phương pháp chính cho hệ thống gợi ý phim được quyết định dựa trên nhiều yếu tố quan trọng. Đầu tiên, phương pháp này không yêu cầu dữ liệu lịch sử tương tác từ nhiều người dùng như Collaborative Filtering, giúp hệ thống có thể hoạt động ngay cả khi số lượng người dùng mới hoặc dữ liệu đánh giá còn hạn chế. Điều này đặc biệt quan trọng trong giai đoạn khởi động của ứng dụng, khi chưa có đủ dữ liệu về hành vi người dùng để áp dụng các phương pháp gợi ý khác.

Thứ hai, Content-Based Filtering cung cấp khả năng giải thích rõ ràng cho các gợi ý. Hệ thống có thể chỉ ra chính xác lý do tại sao một bộ phim được gợi ý, ví

dụ như "phim này có cùng thể loại và diễn viên với phim đã xem", giúp tăng độ tin cậy và trải nghiệm người dùng. Điều này khác với các phương pháp dựa trên Collaborative Filtering, thường khó giải thích vì dựa trên mối tương quan ẩn giữa các người dùng. Hình 3.1 minh họa sự khác biệt cơ bản giữa hai phương pháp này.

Thứ ba, phương pháp này phù hợp với đặc thù của dữ liệu phim ảnh, nơi mà các thuộc tính như thể loại, diễn viên, đạo diễn, năm sản xuất là những yếu tố quan trọng và dễ dàng trích xuất từ metadata của phim. Những thông tin này có sẵn và đáng tin cậy, giúp xây dựng profile phim một cách chính xác mà không cần phụ thuộc vào dữ liệu người dùng.

Cuối cùng, Content-Based Filtering tránh được vấn đề "cold start" đối với sản phẩm mới. Khi một bộ phim mới được thêm vào hệ thống, ngay lập tức có thể được gợi ý dựa trên các thuộc tính của nó, không cần chờ đợi người dùng đánh giá trước. Điều này đảm bảo hệ thống luôn cập nhật với các nội dung mới nhất và đáp ứng được nhu cầu khám phá của người dùng.



Hình 3.1: So sánh Content-Based Filtering với Collaborative Filtering

3.2 Ứng dụng Content-Based Filtering trong hệ thống gợi ý phim

3.2.1 Vai trò và mục đích sử dụng

Trong hệ thống gợi ý phim, Content-Based Filtering đóng vai trò là công cụ chính để cung cấp các gợi ý phim cá nhân hóa dựa trên nội dung. Khi người dùng xem chi tiết một bộ phim, hệ thống sẽ tự động hiển thị danh sách các phim tương tự ở phần "Phim liên quan" hoặc "Có thể bạn sẽ thích". Điều này giúp người dùng dễ dàng khám phá các nội dung mới mà không cần phải tìm kiếm thủ công.

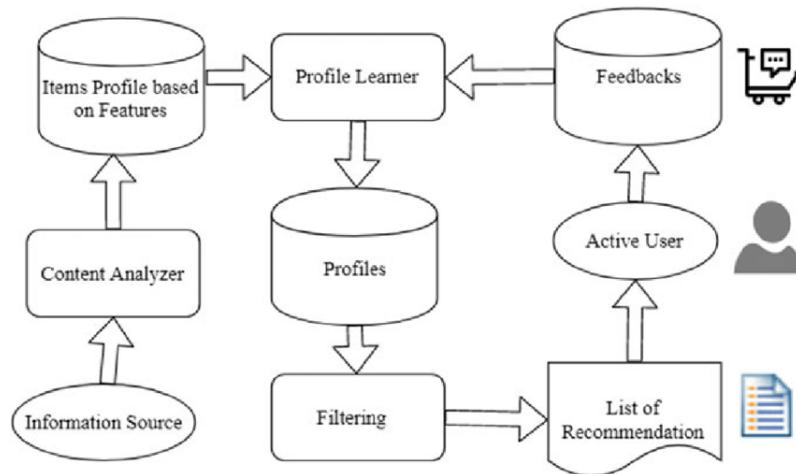
Hệ thống sử dụng Content-Based Filtering để phân tích các đặc trưng của phim bao gồm: thể loại (genres), dàn diễn viên (cast), đạo diễn (director), mô tả nội dung (description), năm sản xuất, và quốc gia sản xuất. Mỗi phim được biểu diễn dưới

dạng một vector đặc trưng, trong đó mỗi thành phần của vector đại diện cho một thuộc tính cụ thể. Hệ thống sau đó tính toán độ tương đồng giữa các phim dựa trên các vector này để xác định các phim có nội dung tương tự nhất.

3.2.2 Cách thức hoạt động trong hệ thống

Quy trình hoạt động của Content-Based Filtering trong hệ thống được thực hiện qua các bước như được minh họa trong Hình 3.2:

1. Trích xuất đặc trưng: Hệ thống thu thập và xử lý metadata của phim từ cơ sở dữ liệu, bao gồm tên phim, thể loại, dàn diễn viên, đạo diễn, mô tả, và các thông tin khác. Mỗi thuộc tính được mã hóa thành các giá trị số hoặc vector.
2. Xây dựng profile phim: Các thuộc tính được kết hợp để tạo thành profile hoàn chỉnh cho mỗi phim. Với dữ liệu văn bản như mô tả, hệ thống sử dụng TF-IDF để trích xuất các từ khóa quan trọng. Với các thuộc tính phân loại như thể loại và diễn viên, hệ thống sử dụng one-hot encoding hoặc embedding.
3. Tính toán độ tương đồng: Khi người dùng xem một phim, hệ thống tính toán độ tương đồng giữa phim đó với tất cả các phim khác trong cơ sở dữ liệu sử dụng Cosine Similarity. Các phim có độ tương đồng cao nhất sẽ được lựa chọn để gợi ý.
4. Lọc và sắp xếp kết quả: Hệ thống lọc ra các phim có độ tương đồng vượt qua ngưỡng nhất định, sau đó sắp xếp chúng theo thứ tự giảm dần của độ tương đồng. Top N phim (ví dụ: top 10 hoặc top 20) sẽ được hiển thị cho người dùng.



Hình 3.2: Quy trình Content-Based Filtering

3.2.3 Vị trí sử dụng trong ứng dụng

Content-Based Filtering được tích hợp vào hệ thống ở các điểm quan trọng sau:

- Trang chi tiết phim: Khi người dùng xem thông tin chi tiết của một bộ phim, hệ thống sẽ hiển thị phần "Phim tương tự" hoặc "Có thể bạn sẽ thích" ở cuối trang.

Đây là vị trí chính mà Content-Based Filtering được áp dụng, giúp người dùng khám phá các phim liên quan ngay sau khi quan tâm đến một phim cụ thể.

- **Trang gợi ý:** Trong trang gợi ý chuyên dụng, Content-Based Filtering có thể được kết hợp với các phương pháp khác để tạo ra danh sách gợi ý đa dạng và phong phú. Hệ thống có thể sử dụng lịch sử xem phim của người dùng để xây dựng profile người dùng, sau đó tìm các phim tương tự với các phim mà người dùng đã quan tâm.
- **Chức năng tìm kiếm nâng cao:** Content-Based Filtering cũng hỗ trợ chức năng tìm kiếm bằng cách cho phép người dùng tìm các phim tương tự với một phim mẫu. Người dùng có thể nhập tên phim hoặc chọn một phim, và hệ thống sẽ trả về danh sách các phim có nội dung tương tự.

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

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

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

Công thức tính Cosine Similarity giữa hai vector A và B được biểu diễn như sau:

$$\text{similarity}(A, B) = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$

Trong đó:

- $A \cdot B$ là tích vô hướng của hai vector
- $\|A\|$ và $\|B\|$ là độ dài (norm) của các vector
- Giá trị cosine similarity nằm trong khoảng $[-1, 1]$, với 1 biểu thị độ tương đồng hoàn toàn và -1 biểu thị độ khác biệt hoàn toàn

Cosine Similarity đặc biệt phù hợp cho việc so sánh các vector có chiều dài khác nhau, vì nó chỉ xem xét hướng của vector chứ không phụ thuộc vào độ lớn. Điều này rất hữu ích khi so sánh các phim có số lượng thể loại hoặc diễn viên khác nhau.

3.3 So sánh các thuật toán trong Content-Based Filtering

Trong Content-Based Filtering, có nhiều thuật toán khác nhau có thể được sử dụng để tính toán độ tương đồng giữa các phim. Mỗi thuật toán có những ưu điểm và nhược điểm riêng, phù hợp với các trường hợp sử dụng cụ thể. Dưới đây là so sánh chi tiết giữa các thuật toán phổ biến.

3.3.1 So sánh TF-IDF và các phương pháp trích xuất đặc trưng khác

TF-IDF là một trong những phương pháp phổ biến nhất để trích xuất đặc trưng từ dữ liệu văn bản. Tuy nhiên, trong bối cảnh của hệ thống gợi ý phim, việc so sánh TF-IDF với các phương pháp khác như Bag of Words (BoW) và Word Embeddings là cần thiết:

- Bag of Words (BoW): Đây là phương pháp đơn giản nhất, chỉ đếm tần suất xuất hiện của từ trong văn bản mà không xem xét tầm quan trọng của từ. BoW dễ triển khai nhưng có nhược điểm là không phân biệt được các từ quan trọng và không quan trọng, dẫn đến kết quả kém chính xác hơn TF-IDF.
- TF-IDF: Phương pháp này cải thiện BoW bằng cách nhân tần suất từ (TF) với nghịch đảo tần suất tài liệu (IDF), giúp giảm trọng số của các từ xuất hiện phổ biến và tăng trọng số của các từ đặc trưng. TF-IDF phù hợp cho việc trích xuất từ khóa quan trọng từ mô tả phim, giúp phân biệt được các phim có nội dung tương tự dựa trên các từ khóa đặc trưng.
- Word Embeddings (Word2Vec, GloVe, FastText): Các phương pháp này biểu diễn từ dưới dạng vector trong không gian đa chiều, giúp nắm bắt được ngữ nghĩa và mối quan hệ giữa các từ. Word Embeddings vượt trội hơn TF-IDF trong việc hiểu ngữ cảnh và ý nghĩa của văn bản, nhưng đòi hỏi dữ liệu huấn luyện lớn và chi phí tính toán cao hơn.

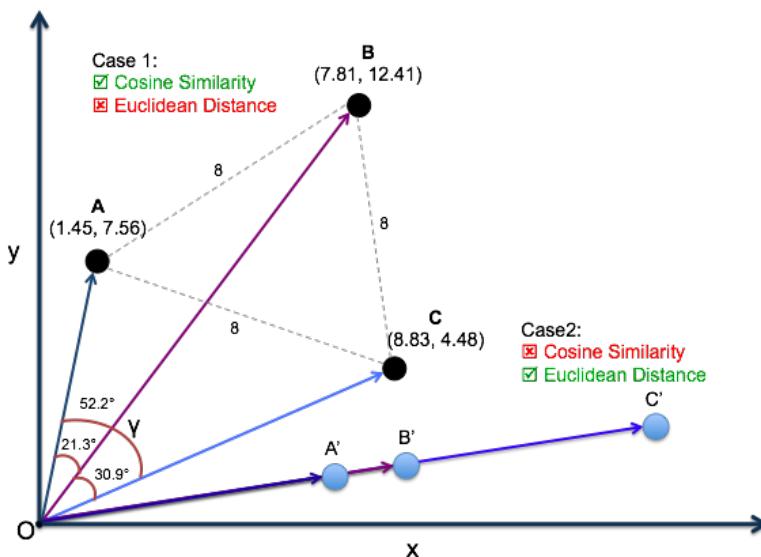
3.3.2 So sánh Cosine Similarity với các phương pháp đo độ tương đồng khác

Cosine Similarity là phương pháp phổ biến để đo độ tương đồng, nhưng cũng có nhiều phương pháp khác có thể được sử dụng:

- Euclidean Distance: Phương pháp này đo khoảng cách trực tiếp giữa hai điểm trong không gian vector. Euclidean Distance phù hợp khi các vector có cùng quy mô và ý nghĩa, nhưng nhạy cảm với sự khác biệt về độ lớn của vector. Trong trường hợp so sánh các phim có số lượng thuộc tính khác nhau, Eu-

clidean Distance có thể không phản ánh chính xác độ tương đồng.

- Cosine Similarity: Phương pháp này đo góc giữa hai vector, không phụ thuộc vào độ lớn của vector. Cosine Similarity đặc biệt phù hợp cho các vector sparse (thưa) như trong trường hợp của metadata phim, nơi mỗi phim chỉ có một số thuộc tính cụ thể. Phương pháp này cũng hiệu quả về mặt tính toán và dễ dàng chuẩn hóa về khoảng [0, 1]. Hình 3.3 minh họa sự khác biệt giữa Cosine Similarity và Euclidean Distance trong việc đo độ tương đồng giữa các vector.
- Jaccard Similarity: Phương pháp này đo độ tương đồng giữa hai tập hợp dựa trên giao và hợp của chúng. Jaccard Similarity rất phù hợp cho dữ liệu phân loại nhị phân (binary), ví dụ như danh sách thể loại hoặc diễn viên mà một phim có hoặc không có. Tuy nhiên, Jaccard không phù hợp cho dữ liệu có trọng số như TF-IDF vectors.
- Manhattan Distance (L1 Distance): Phương pháp này tính tổng giá trị tuyệt đối của sự khác biệt giữa các thành phần tương ứng của hai vector. Manhattan Distance ít nhạy cảm với outliers hơn Euclidean Distance, nhưng thường kém hiệu quả hơn Cosine Similarity trong các bài toán gợi ý.



Hình 3.3: So sánh Cosine Similarity và Euclidean Distance

3.3.3 Lựa chọn thuật toán cho hệ thống

Dựa trên đặc thù của hệ thống gợi ý phim, tôi đã lựa chọn sử dụng TF-IDF kết hợp với Cosine Similarity. Lý do cho sự lựa chọn này bao gồm:

1. Phù hợp với dữ liệu có sẵn: Metadata của phim chứa nhiều thông tin văn bản (mô tả, tóm tắt) và thông tin phân loại (thể loại, diễn viên). TF-IDF có thể xử lý hiệu quả cả hai loại dữ liệu này.

2. Hiệu suất tính toán: TF-IDF và Cosine Similarity có độ phức tạp tính toán thấp, cho phép hệ thống xử lý hàng ngàn phim trong thời gian ngắn. Điều này quan trọng để đảm bảo trải nghiệm người dùng mượt mà.
3. Dễ triển khai và bảo trì: Các thuật toán này đã được chuẩn hóa và có sẵn trong nhiều thư viện, giúp việc triển khai và bảo trì hệ thống trở nên dễ dàng.
4. Kết quả dễ giải thích: Với TF-IDF, hệ thống có thể giải thích rõ ràng các từ khóa nào đóng vai trò quan trọng trong việc xác định độ tương đồng, giúp người dùng hiểu tại sao một phim được gợi ý.

Việc tích hợp thuật toán Content-Based Filtering vào hệ thống giúp tối ưu hóa hiệu suất vận hành, đồng thời đảm bảo mỗi người dùng đều nhận được những đề xuất điện ảnh bám sát sở thích và nhu cầu cá biệt của từng cá nhân.

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

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

Việc hiểu rõ và áp dụng một cách chính xác các kỹ thuật của phương pháp này sẽ tạo điều kiện để cải thiện đáng kể hiệu suất vận hành của hệ thống gợi ý phim, qua đó tạo ra những trải nghiệm người dùng tốt hơn. Những kiến thức và phương pháp luận được trình bày trong chương này chính là nền tảng lý thuyết quan trọng để tôi tiếp tục triển khai và phát triển chuyên sâu hệ thống gợi ý phim trong các chương kế tiếp.

CHƯƠNG 4. CÔNG NGHỆ SỬ DỤNG

4.1 Công nghệ sử dụng phía frontend

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

4.1.1 React

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

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

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

4.1.2 Tailwind CSS

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

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

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

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

4.2 Công nghệ sử dụng phía backend

4.2.1 Django

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

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

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

4.2.2 MySQL

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

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

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

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

4.3 Công nghệ sử dụng trong việc lưu trữ code

4.3.1 Git

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

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

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

4.3.2 GitHub

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

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

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

4.4 Công nghệ sử dụng trong việc crawl và update dữ liệu

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

4.4.1 Selenium và Requests

Selenium là một công cụ tự động hóa trình duyệt web có khả năng mạnh mẽ, cho phép điều khiển trình duyệt và thực hiện các tương tác với các trang web có nội dung động giống như cách một người dùng thực tế sẽ thao tác. Selenium có thể thực hiện nhiều loại hành động khác nhau bao gồm click chuột, nhập dữ liệu, cuộn trang, và chờ đợi các thành phần động được tải hoàn tất. Với thiết kế tinh gọn nhưng hiệu quả, Requests là thư viện chuyên dụng trong Python giúp tối ưu hóa việc truyền tải các yêu cầu HTTP và khai thác dữ liệu từ các nền tảng trực tuyến. Requests được tích hợp nhiều chức năng hữu ích như điều hướng qua các trang

web, gửi dữ liệu dưới định dạng form và quản lý cookies.

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

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

4.4.2 Công nghệ sử dụng để cập nhật dữ liệu

Việc liên tục đồng bộ và làm mới cơ sở dữ liệu điện ảnh là yếu tố then chốt giúp thuật toán đề xuất duy trì tính thời sự cũng như độ chính xác cao cho các nội dung phản hồi tới người dùng. Nhằm thực hiện tự động hóa quy trình này, tôi đã sử dụng GitHub Actions. GitHub Actions là một công cụ CI/CD được tích hợp sẵn trong nền tảng GitHub, có khả năng cho phép người dùng tạo ra các workflow tự động sẽ được kích hoạt khi xảy ra các sự kiện nhất định.

Bằng cách sử dụng GitHub Actions, tôi đã tạo ra các workflow có khả năng tự động cập nhật dữ liệu phim từ IMDb theo chu kỳ hàng ngày. Những quy trình tự động này vận hành các tập lệnh Selenium và Requests nhằm khai thác thông tin mới, từ đó đồng bộ hóa và làm mới dữ liệu trực tiếp vào hệ thống lưu trữ. Quá trình tự động hóa này đảm bảo rằng dữ liệu luôn được cập nhật đều đặn theo đúng lịch trình đã định mà không cần có sự can thiệp từ con người.

Cụ thể hơn, mỗi ngày GitHub Actions sẽ tự động khởi chạy các script thu thập dữ liệu để kiểm tra và áp dụng các thay đổi mới nhất từ IMDb. Hệ thống thực hiện việc bổ sung và làm mới các thuộc tính như nguồn thu, đoạn giới thiệu, tư liệu hình ảnh cùng những dữ liệu phụ trợ liên quan thông qua quy trình GitHub Actions được thiết lập theo các giai đoạn cụ thể dưới đây:

Thông qua thư viện Requests, hệ thống thực hiện truyền tải các lệnh gọi HTTP nhằm trích xuất các nguồn tài nguyên tĩnh từ những nền tảng web mục tiêu.

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

Dữ liệu sau khi xử lý sẽ được đồng bộ trực tiếp vào hệ thống MySQL để đảm bảo tính cập nhật liên tục.

Việc tích hợp GitHub Actions không chỉ tối ưu hóa khả năng tự động hóa quy trình làm mới dữ liệu mà còn duy trì sự nhất quán và độ tin cậy cho kho thông tin điện ảnh. Cơ chế này hỗ trợ hệ thống đề xuất luôn phản hồi những kết quả mới nhất, từ đó đáp ứng trọn vẹn nhu cầu truy vấn của người dùng.

Phần này đã tổng hợp các giải pháp công nghệ cốt lõi được ứng dụng trong việc xây dựng hệ thống gợi ý phim. Sự kết hợp giữa Django, MySQL ở phía server cùng React và Tailwind CSS cho giao diện, đi kèm với quy trình quản lý mã nguồn qua Git/GitHub và phương thức khai thác dữ liệu từ IMDb, BoxOfficeMojo đã tạo nên một nền tảng vận hành ổn định, hiệu quả. Những lựa chọn kỹ thuật này không chỉ giải quyết tốt các yêu cầu hiện tại mà còn tạo tiền đề vững chắc cho việc nâng cấp và mở rộng quy mô 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 dự án này, tôi quyết định áp dụng mô hình kiến trúc MVT (Model-View-Template) được cung cấp bởi khung làm việc Django [7].

Giải thích sơ bộ về MVT:

Mô hình MVT trong Django có những nét tương đồng nhất định với kiến trúc MVC (Model-View-Controller) truyền thống, tuy nhiên quy trình vận hành và sự phân chia trách nhiệm giữa các thành phần lại có những điểm khác biệt đặc trưng. Trong kiến trúc MVT, Model (M) thiết lập khung sơ đồ dữ liệu cho ứng dụng và điều phối các thao tác truy xuất vào cơ sở dữ liệu. Các dữ liệu từ Model được xử lý bởi View (V), đảm nhận vai trò xử lý các quy trình nghiệp vụ hệ thống và điều hướng dữ liệu đến người dùng cuối. Cuối cùng, Template (T) xây dựng lớp giao diện hiển thị thông qua ngôn ngữ HTML kết hợp linh hoạt với các biến dữ liệu Python được truyền từ View, tạo ra một luồng xử lý dữ liệu từ Model qua View đến Template để hiển thị cho người dùng.

Mô tả kiến trúc cụ thể cho ứng dụng:

• Model (M):

Trong hệ thống của tôi, các thực thể như MovieInformation, Awards, Director, Cast và Genres đóng vai trò là những mô hình dữ liệu cốt lõi nhằm định nghĩa thông tin về phim ảnh, đội ngũ sản xuất và các phân loại liên quan.

Mỗi mô hình này được thiết lập các thuộc tính cụ thể và những mối liên kết đa chiều, giúp tối ưu hóa việc lưu trữ cũng như quản lý dữ liệu theo một cấu trúc logic và hệ thống.

• View (V):

Trong ứng dụng của tôi, các thành phần như FilmListView, DirectorListView, GenreListView và ReviewView đóng vai trò tiếp nhận và phản hồi các yêu cầu từ phía người dùng.

Tại đây, các View thực hiện việc trích xuất dữ liệu từ Model bằng các QuerySet, thực thi các quy trình nghiệp vụ cần thiết và đóng gói dữ liệu để chuyển tiếp đến lớp Template hiển thị.

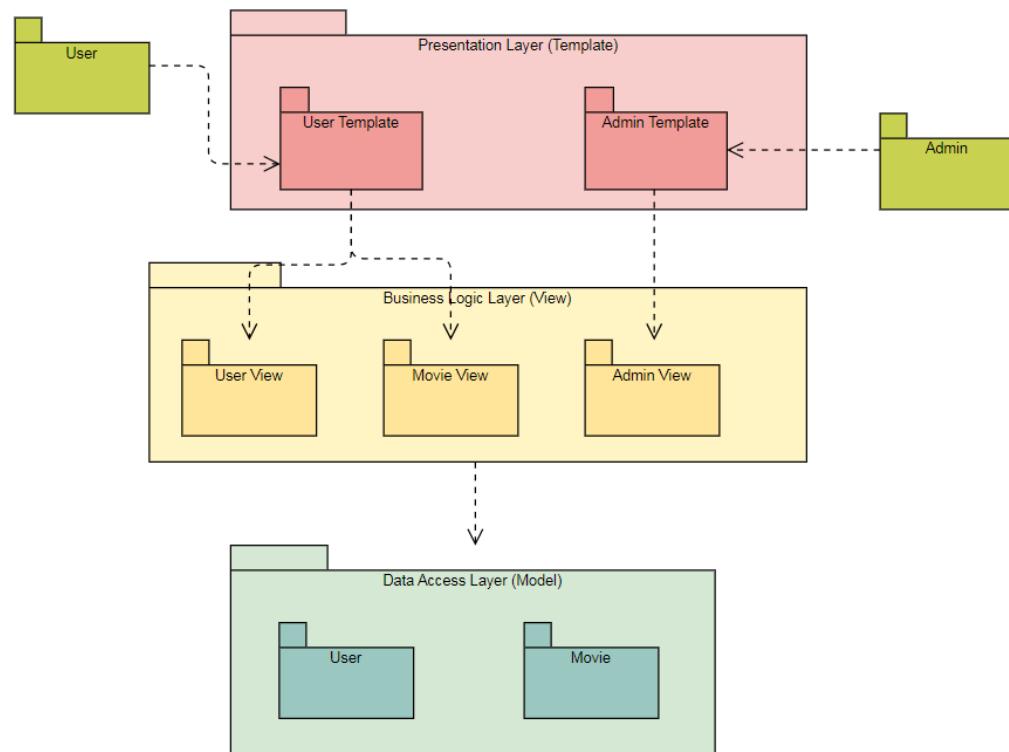
• Template (T):

Việc hiển thị thông tin tới người dùng được thực hiện thông qua hệ thống template HTML của Django kết hợp cùng các template tags chuyên dụng.

Các tệp giao diện này tiếp nhận dữ liệu từ View để kết xuất thành các thành phần trực quan như danh sách phim, thông tin chi tiết hoặc các sơ đồ minh họa.

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 tiêu: Lớp này đóng vai trò lưu trữ các định nghĩa và thiết lập cấu trúc dữ liệu cốt lõi cho toàn bộ hệ thống.
- Chức năng: Tiến hành mô phỏng các đối tượng thực tế trong ứng dụng bằng cách thiết lập các bảng dữ liệu cùng những liên kết logic giữa chúng.
- Đối tượng chính: MovieInformation, Awards, Director, Cast, Genres.
- Tính liên kết: Hoạt động độc lập và không dựa trên các gói bên ngoài, đây là thành phần nền tảng trong kiến trúc hệ thống, nơi dữ liệu được khởi tạo và điều phối qua các model.

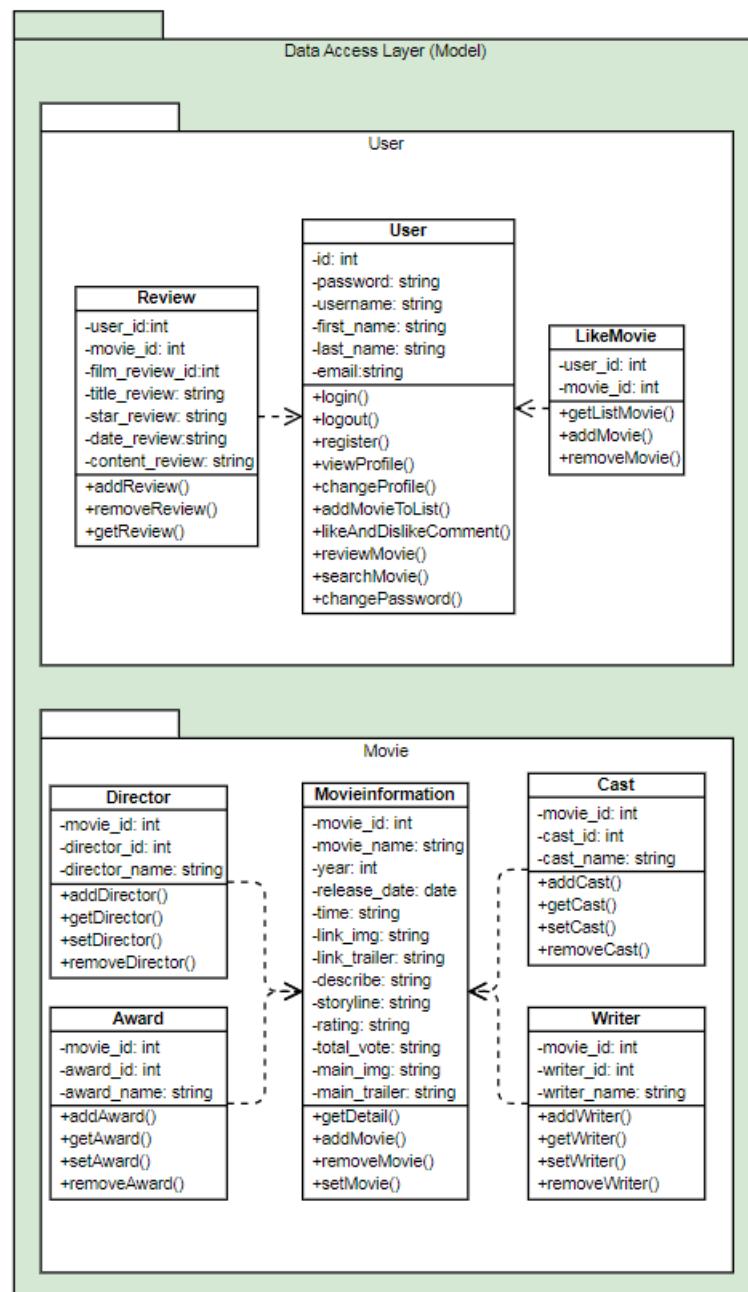
2. Business Logic Layer (View)

- Mục tiêu: Thành phần này tập trung vào việc điều phối các luồng xử lý nghiệp vụ và phản hồi các tác vụ do người dùng gửi đến.
- Vai trò: Trực tiếp lấy thông tin từ lớp Model, thực hiện các thao tác tính toán và định dạng lại dữ liệu trước khi chuyển giao cho lớp hiển thị.
- Các thành phần chính: FilmListView, DirectorListView, GenreListView, ReviewView.
- Liên kết hệ thống: Lớp này chịu sự phụ thuộc vào gói Models để truy xuất dữ liệu nhưng hoạt động tách biệt với gói Templates. Sau khi hoàn tất xử lý logic, View mới bắt đầu chuyển dữ liệu sang Template để kết xuất giao diện cho người dùng.

3. Presentation Layer (Template)

- Mục tiêu: Lớp này đóng vai trò thiết lập và trình bày giao diện đồ họa cho người sử dụng cuối.
- Chức năng: Chuyển đổi thông tin nhận được từ View thành các định dạng hiển thị trực quan thông qua HTML, CSS và JavaScript để người dùng có thể tương tác.
- Các thành phần: Hệ thống các tệp HTML được thiết kế riêng cho từng trang trong ứng dụng như giao diện đăng nhập, đăng ký, trang chủ và trang thông tin chi tiết phim.
- Liên kết hệ thống: Lớp này hoạt động dựa trên dữ liệu cung cấp từ gói Views mà không cần tương tác trực tiếp với gói Models. Toàn bộ thông tin hiển thị trên màn hình đều được luân chuyển và xử lý thông qua tầng View trước khi đến với Template.

5.1.3 Thiết kế chi tiết gói



Hình 5.2: Thiết kế chi tiết gói ở tầng Model

Gói Model:

Gói này được cấu thành từ hai nhóm nhỏ hơn là User và Movie.

- **Gói User:** Gói này được cấu thành từ ba lớp bao gồm Review, User và LikeMovie.

Về chức năng: Lớp User đóng vai trò mô hình hóa dữ liệu người dùng trong hệ thống. Lớp Review được thiết kế để quản lý các bản đánh giá mà người dùng đã tạo ra cho các bộ phim. Lớp LikeMovie có nhiệm vụ ghi nhận và lưu giữ

danh sách các bộ phim được người dùng đánh dấu là yêu thích.

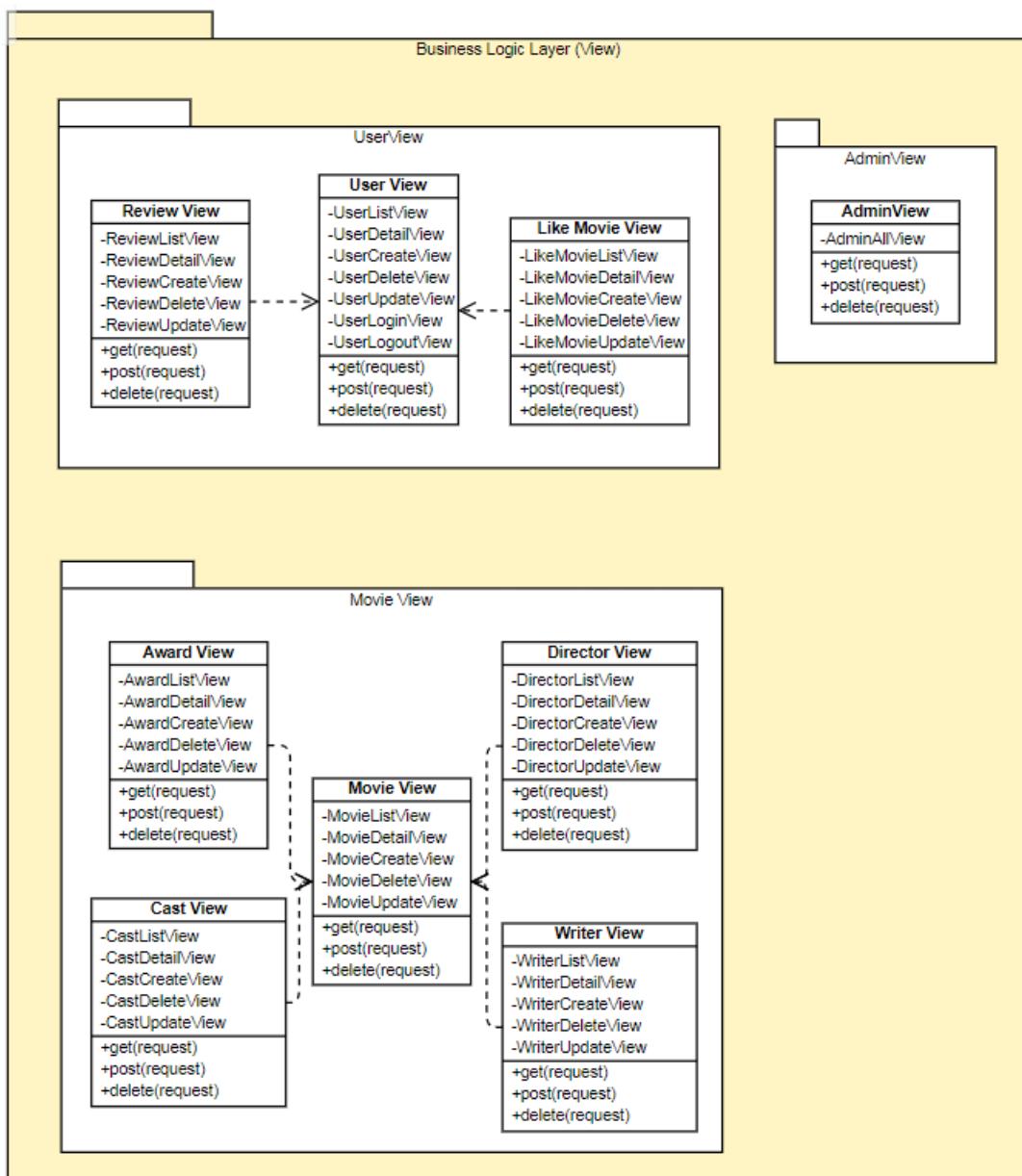
Về mối quan hệ: Lớp Review và lớp LikeMovie đều cần tham chiếu đến dữ liệu từ lớp User để hoạt động.

• **Gói Movie:**

Gói này bao gồm năm lớp là Movieinfomation, Director, Award, Cast và Writer.

Về chức năng: Lớp MovieInformation đóng vai trò trung tâm trong việc quản lý các thông tin cơ bản về phim. Lớp Director được dùng để lưu trữ các thông tin cụ thể liên quan đến đạo diễn của từng bộ phim. Lớp Award có chức năng ghi nhận các giải thưởng mà phim đã đạt được. Lớp Cast chịu trách nhiệm quản lý thông tin về đội ngũ diễn viên xuất hiện trong phim. Lớp Writer được thiết kế để lưu trữ thông tin về các tác giả kịch bản.

Về mối quan hệ: Bốn lớp còn lại đều có mối liên kết phụ thuộc với lớp Movieinfomation, trong đó Movieinfomation đóng vai trò là lớp cốt lõi, còn các lớp khác đóng vai trò bổ trợ thông tin.



Hình 5.3: Thiết kế chi tiết gói ở tầng View

Gói View:

Gói này được phân chia thành ba nhóm con bao gồm UserView, MovieView và AdminView.

- Gói UserView:

Gói này được tạo thành từ ba lớp là ReviewView, UserView và LikeMovieView.

Về chức năng: Lớp UserView có nhiệm vụ trình bày và điều phối các thông tin về người dùng, bao gồm các chức năng đăng ký tài khoản, đăng nhập hệ thống, và quản lý hồ sơ cá nhân. Lớp ReviewView đảm nhận việc xử lý và trình bày các bản đánh giá phim do người dùng tạo ra. Lớp này cần sử dụng dữ liệu từ UserView để hoạt động. Lớp LikeMovieView có chức năng điều

hành và hiển thị danh sách các bộ phim được người dùng đánh dấu yêu thích. Lớp này cũng cần tham chiếu đến thông tin từ UserView.

Về mối quan hệ: Lớp ReviewView và lớp LikeMovieView đều cần dữ liệu từ lớp UserView để thực hiện các chức năng của mình.

- Gói MovieView:

Gói này chứa năm lớp là MovieinfomationView, DirectorView, AwardView, CastView và WriterView.

Về chức năng: Lớp MovieInformationView đóng vai trò trung tâm trong việc điều hành và thực hiện các thao tác chính liên quan đến phim. Lớp DirectorView chuyên xử lý các dữ liệu về đạo diễn. Lớp AwardView tập trung vào việc xử lý dữ liệu và thực hiện các thao tác liên quan đến giải thưởng. Lớp CastView chuyên xử lý dữ liệu và thực hiện các thao tác tính toán về diễn viên. Lớp WriterView thực hiện các thao tác dữ liệu liên quan đến biên kịch.

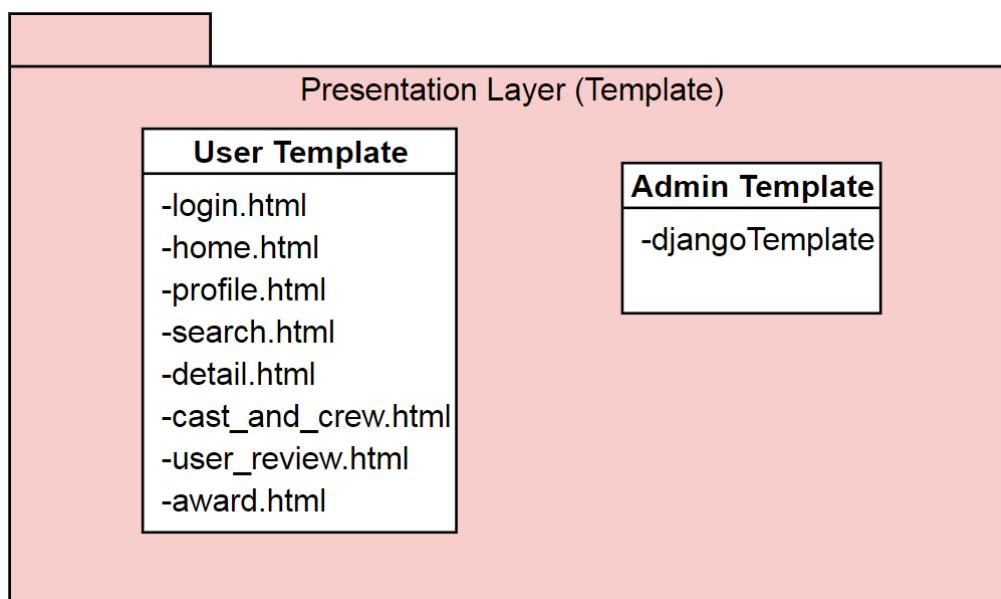
Về mối quan hệ: Bốn lớp còn lại đều có mối liên kết phụ thuộc với lớp Movieinfomation, trong đó Movieinfomation đóng vai trò là lớp nền tảng, còn các lớp khác cung cấp thông tin mở rộng.

- Gói AdminView:

Gói này chỉ chứa một lớp duy nhất là AdminView.

Về chức năng: Lớp AdminView có khả năng xử lý mọi loại yêu cầu trong hệ thống, bao gồm cả những yêu cầu tương tự như người dùng thông thường và các yêu cầu đặc biệt khác, cho phép quản trị viên điều hành toàn bộ website. Lớp này có đầy đủ quyền hạn để thực hiện các thao tác thêm mới, chỉnh sửa và xóa bỏ dữ liệu trong hệ thống.

Về mối quan hệ: Lớp này hoạt động độc lập và không cần phụ thuộc vào bất kỳ lớp nào khác.

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

Gói Template:

Hệ thống này được cấu thành từ hai bộ phận chủ chốt bao gồm AdminTemplate cùng với UserTemplate.

Về chức năng, lớp UserTemplate đảm nhận việc quản lý hệ thống các tệp HTML giao diện dành riêng cho người dùng cuối, bao gồm các trang chức năng như đăng nhập, đăng ký, hồ sơ cá nhân, khu vực đánh giá phim, cùng các danh mục hiển thị về phim ảnh, diễn viên và đạo diễn. Trong khi đó, lớp AdminTemplate tập trung vào việc quản lý các giao diện dành cho nhà quản trị, hỗ trợ các nghiệp vụ điều hành hệ thống như kiểm soát tài khoản người dùng, cập nhật dữ liệu phim, kiểm duyệt bình luận và các công cụ quản lý tổng thể khác.

Về tính liên kết, hai lớp này được thiết kế hoàn toàn độc lập và không có quan hệ phụ thuộc lẫn nhau. Mỗi lớp vận hành tự chủ nhằm đáp ứng các mục tiêu chuyên biệt: một bên tối ưu hóa trải nghiệm cho khách hàng và một bên cung cấp công cụ vận hành cho bộ phận quản trị.

5.2 Thiết kế chi tiết

5.2.1 Thiết kế giao diện

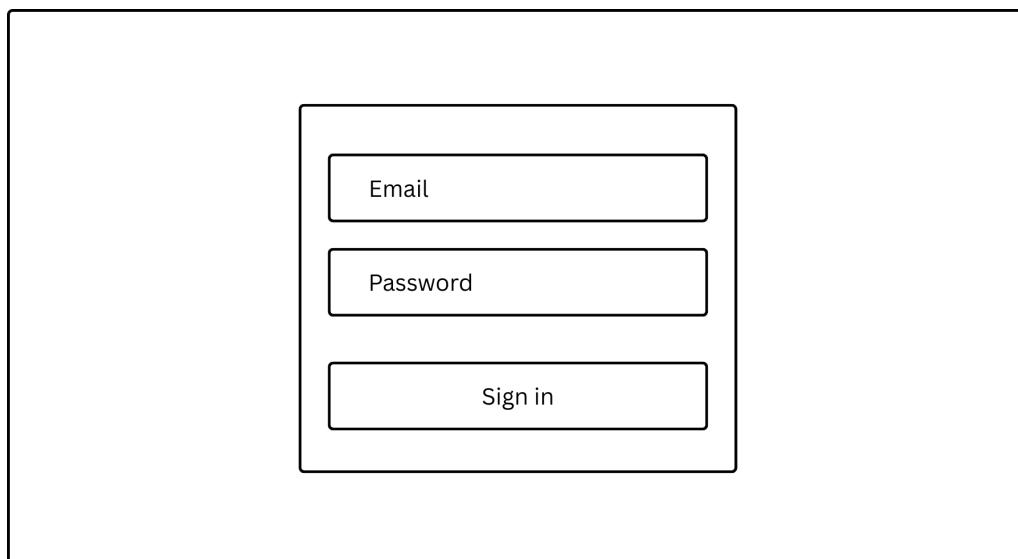
Khi tiến hành thiết kế giao diện cho hệ thống, tôi đã chọn lựa các tham số kỹ thuật, các tiêu chuẩn thiết kế và các mẫu template chuẩn để đảm bảo sự đồng nhất và tính tiện dụng. Những màn hình chính của hệ thống được xây dựng với độ phân giải 1920×1080 pixels, tương thích với các loại màn hình có kích thước từ 13 inch trở lên. Hệ thống có khả năng hiển thị lên đến 16 triệu màu để đảm bảo các thành phần đồ họa và hình ảnh hiển thị với chất lượng cao và chi tiết.

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

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

Phần tiếp theo trình bày các hình ảnh mô tả giao diện người dùng của những tính năng quan trọng nhất trong hệ thống:

Thiết kế màn hình login



Hình 5.5: Thiết kế màn hình *login*

Thiết kế màn hình signup

A wireframe diagram of a user interface. It features a large rectangular container with rounded corners. Inside, there is a vertical stack of five rectangular input fields. From top to bottom, the fields are labeled: 'Name', 'Email', 'Password', 'Confirm Password', and 'Sign up'. The 'Sign up' button is positioned at the bottom of the stack.

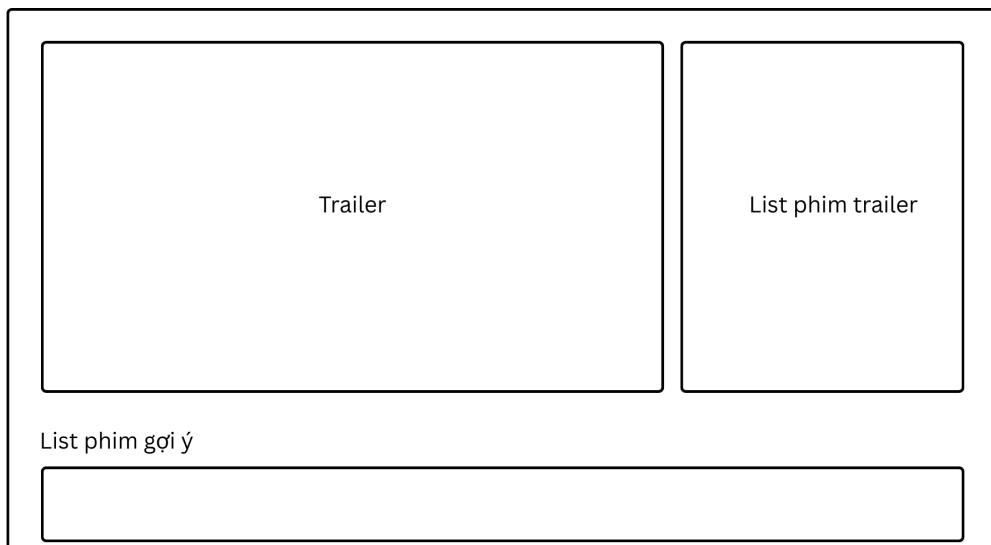
Hình 5.6: Thiết kế màn hình *signup*

Thiết kế màn hình user profile

A wireframe diagram of a user interface. It consists of a large rectangular container. On the left side, there is a vertical stack of four rectangular input fields: 'Name', 'Email', 'Password', and 'Edit'. To the right of these fields is a large, empty rectangular area with the text 'Danh sách phim yêu thích' centered in it.

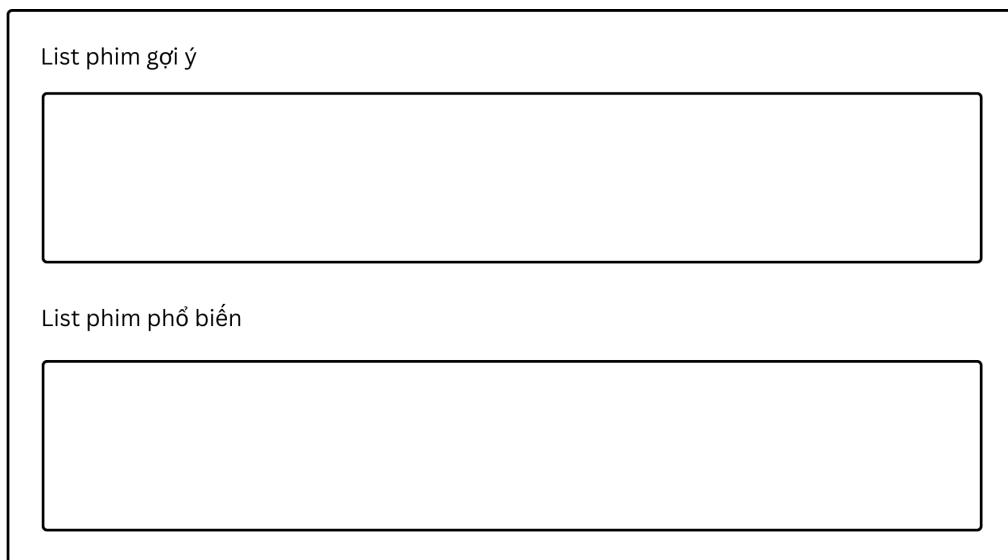
Hình 5.7: Thiết kế màn hình *user profile*

Thiết kế màn hình home



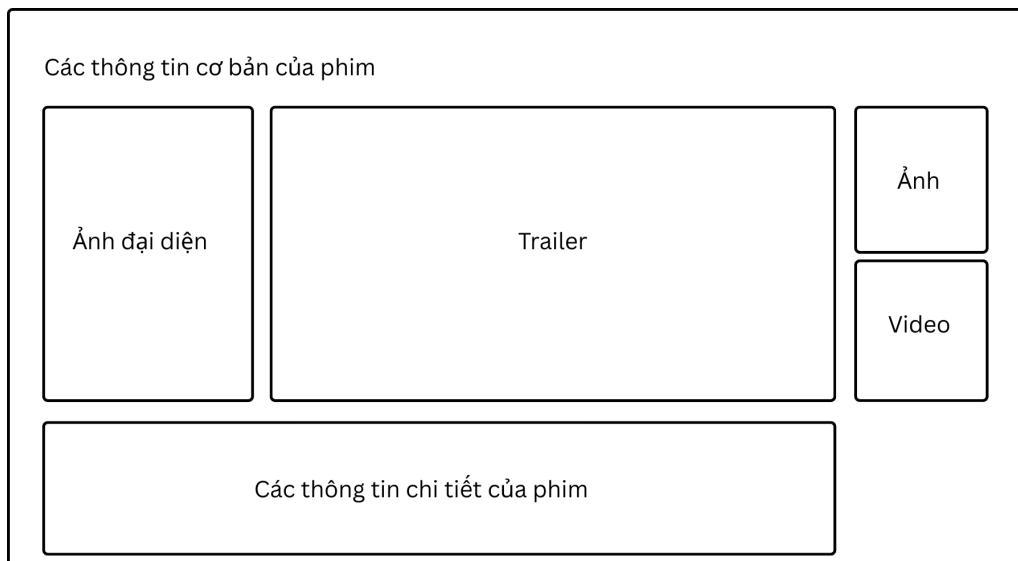
Hình 5.8: Thiết kế màn hình *home*

Thiết kế màn hình gợi ý phim ở trang home

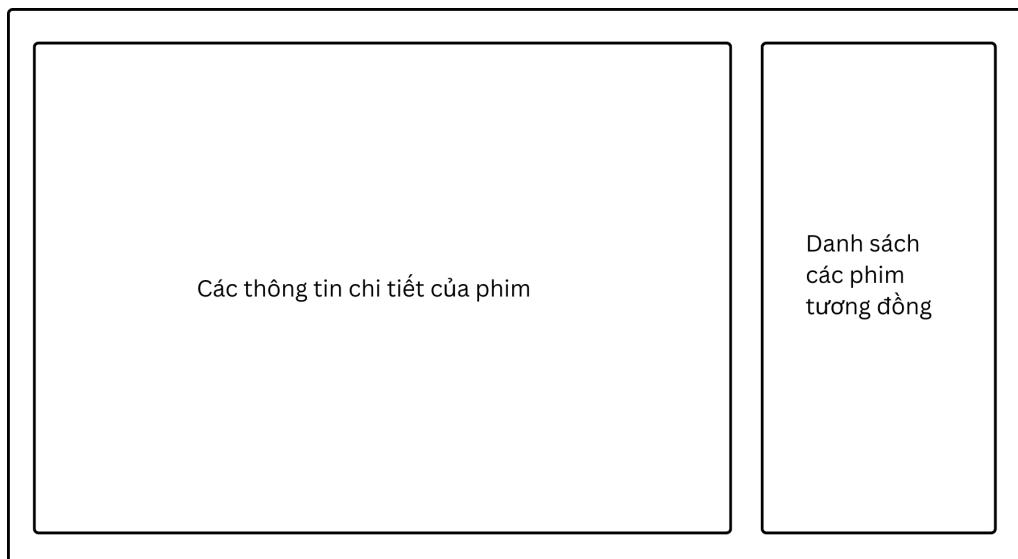


Hình 5.9: Thiết kế màn hình *gợi ý phim* ở trang *home*

Thiết kế màn hình detail

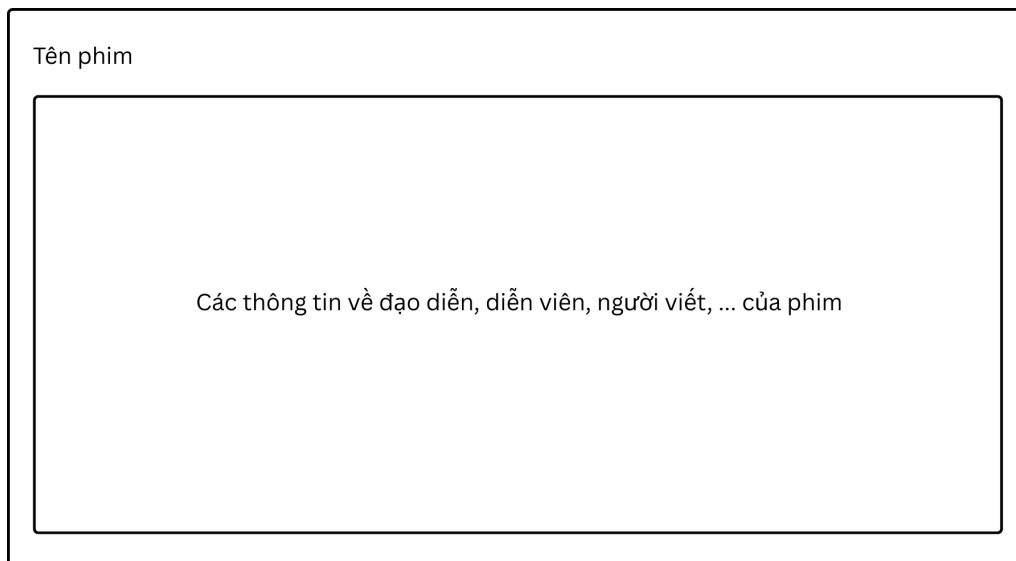


Hình 5.10: Thiết kế màn hình *detail 1*



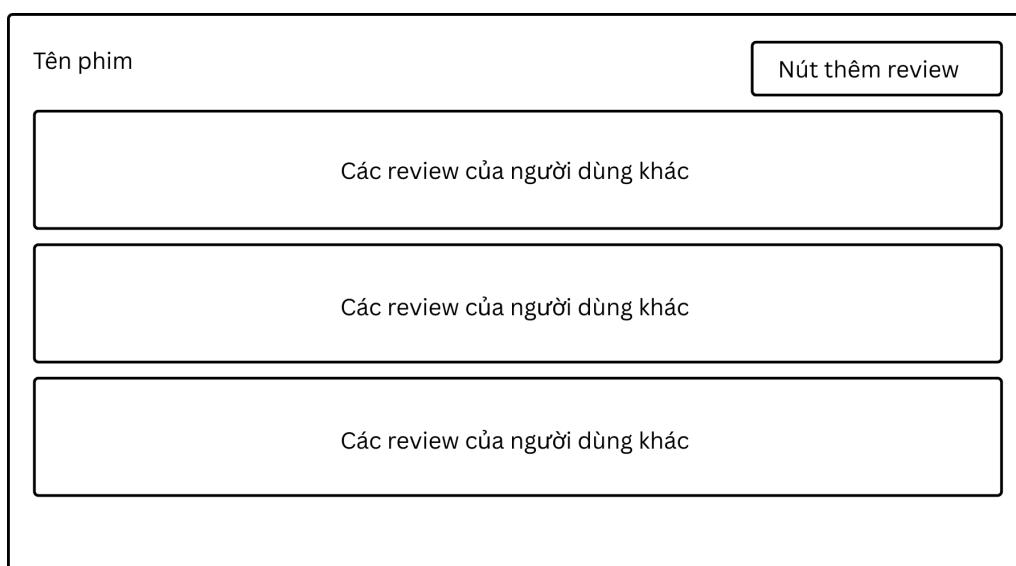
Hình 5.11: Thiết kế màn hình *detail 2*

Thiết kế màn hình cast and crew



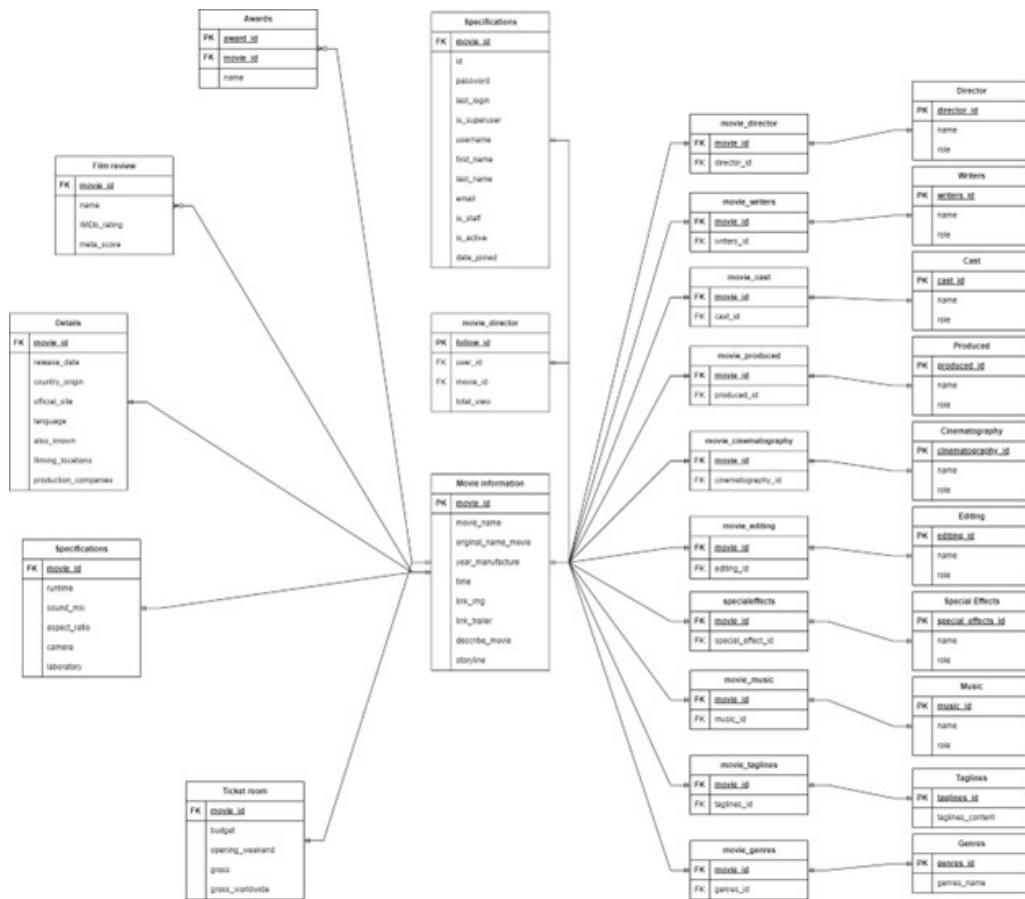
Hình 5.12: Thiết kế màn hình *cast and crew*

Thiết kế màn hình user review

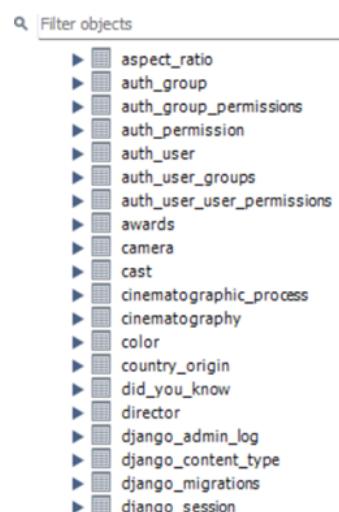


Hình 5.13: Thiết kế màn hình *user review*

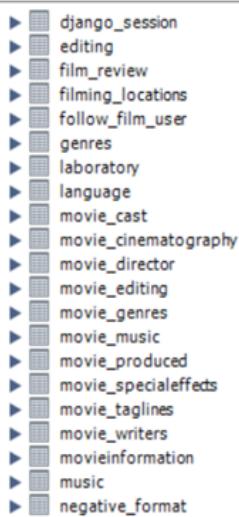
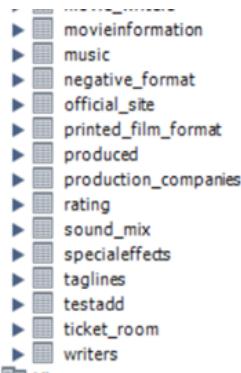
5.2.2 Thiết kế cơ sở dữ liệu hệ thống



Hình 5.14: Thiết kế sơ đồ ERD của hệ thống



Hình 5.15: Hình ảnh các bảng trong cơ sở dữ liệu 1

**Hình 5.16:** Hình ảnh các bảng trong cơ sở dữ liệu 2**Hình 5.17:** Hình ảnh các bảng trong cơ sở dữ liệu 3

5.2.3 Thiết kế hệ thống crawl dữ liệu

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

Bảng 5.1: Tổng hợp các thông tin cần lấy

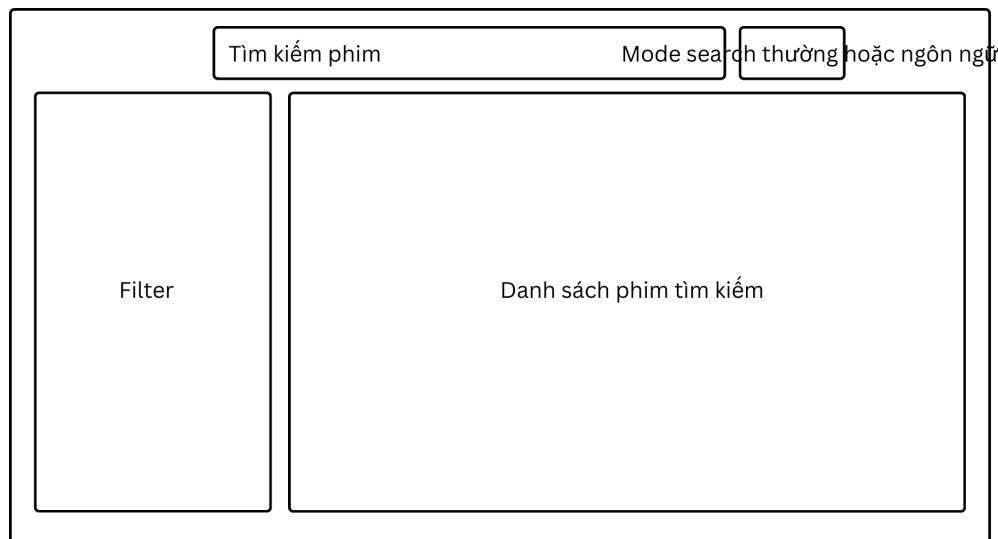
| Page | Thông tin cần lấy |
|-----------|--|
| Home | Các thông tin cơ bản của một bộ phim như: Tên phim, ngày tháng năm sản xuất, mô tả |
| Award | Giải thưởng của phim |
| Director | Đạo diễn |
| Cast | Các diễn viên nổi tiếng và các diễn viên khác |
| Storyline | Cốt truyện, thể loại |
| Rating |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

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

5.2.4 Tìm kiếm phim bằng ngôn ngữ tự nhiên

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

5.2.4.1 Tổng quan luồng xử lý



Hình 5.18: Màn hình *tìm kiếm phim*

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

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

5.2.4.2 Các thành phần kỹ thuật

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

Backend API Endpoint: Backend kiểm tra tính hợp lệ của query, xử lý yêu cầu

tìm kiếm. Xử lý phân trang và điều phối hai phương pháp tìm kiếm song song.

Vector Search (Semantic Search):

Thực hiện tìm kiếm theo ngữ nghĩa:

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

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

AI Filter (Structured Search):

Phân tích câu query bằng AI:

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

5.2.4.3 Chiến lược kết hợp kết quả

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

- **Ưu tiên cao:** Các phim xuất hiện trong cả Vector Search và AI Filter - được sắp xếp theo độ tương đồng (similarity) từ cao xuống thấp
- **Ưu tiên trung:** Các phim chỉ xuất hiện trong Vector Search - tối đa 50 kết quả

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

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

5.2.4.4 Xử lý sau tìm kiếm

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

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

5.2.4.5 Kết quả trả về

API trả về response với cấu trúc:

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

5.2.4.6 Điểm mạnh của ứng dụng

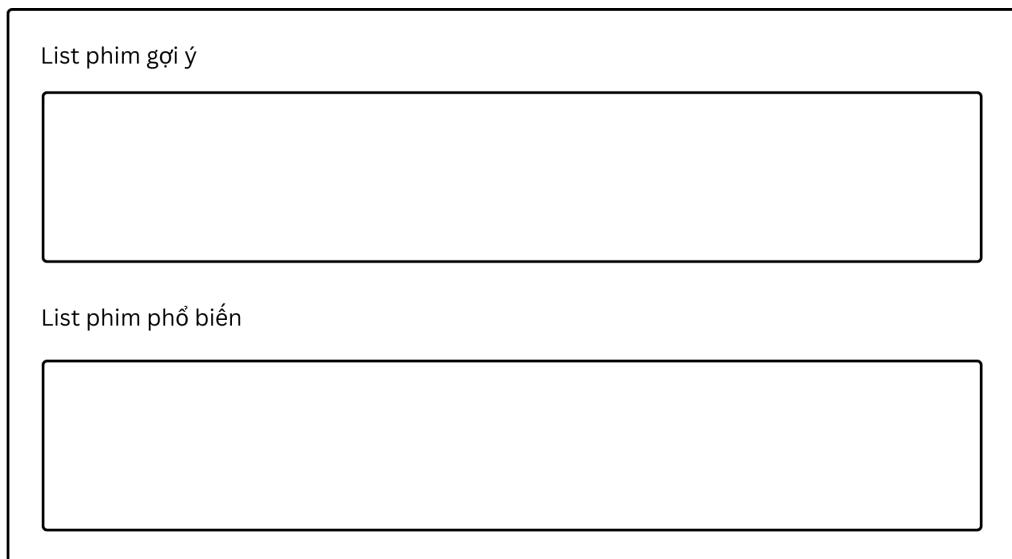
Tính năng tìm kiếm bằng ngôn ngữ người dùng có các ưu điểm sau:

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

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

Ứng dụng sử dụng GenAI (Gemini hoặc OpenAI) để phân tích lịch sử hoạt động của người dùng và đưa ra gợi ý phim cá nhân hóa theo thời gian thực. Khác với các phương pháp gợi ý truyền thống yêu cầu training model, hệ thống này hoạt động real-time dựa trên dữ liệu lịch sử của từng người dùng, không cần retrain hay cập nhật model.

5.2.5.1 Tổng quan luồng xử lý



Hình 5.19: Màn hình gợi ý phim bằng genAI

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

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

5.2.5.2 Các thành phần kỹ thuật

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

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

Thu thập lịch sử hoạt động:

Ứng dụng thu thập ba loại dữ liệu lịch sử:

- Lịch sử tìm kiếm: Tự động lưu lại các câu tìm kiếm của người dùng, bao gồm cả tìm kiếm thường và tìm kiếm ngôn ngữ tự nhiên. Ứng dụng lấy tối đa 20 query gần nhất để phân tích

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

Xây dựng prompt cho GenAI:

Ứng dụng tạo prompt chứa:

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

Gọi GenAI API:

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

Xử lý response từ GenAI:

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

Tìm phim trong database:

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

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

5.2.5.3 Kết quả trả về

API trả về response với cấu trúc:

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

5.3 Xây dựng ứng dụng

5.3.1 Thư viện và công cụ sử dụng

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

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

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

5.3.2 Kết quả đạt được

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

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

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

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

Việc đóng gói các sản phẩm có ý nghĩa then chốt trong công tác quản trị và trình diễn các dữ liệu liên quan đến tác phẩm điện ảnh, đội ngũ đạo diễn, dàn diễn viên cùng những phản hồi từ khán giả. Nền tảng này cho phép người dùng chủ động tra cứu, theo dõi dữ liệu điện ảnh, đóng góp ý kiến cá nhân và tương tác trực tiếp với các nội dung trên hệ thống. Song song đó, thông qua công cụ gợi ý thông minh, ứng dụng có khả năng phân tích hành vi để đề xuất những tác phẩm điện ảnh tối ưu và phù hợp nhất với sở thích riêng biệt của từng cá nhân.

Thống kê các thông tin của ứng dụng:

- Quy mô mã nguồn thu thập dữ liệu: 2000 dòng
- Quy mô mã nguồn phía backend: 4000 dòng

- Quy mô mã nguồn phía frontend: 8000 dòng
- Số lượng lớp: 180 lớp
- Số lượng gói: 8 gói
- Kích thước toàn bộ mã nguồn: Đạt mức 310MB khi tích hợp đầy đủ dữ liệu và thuật toán gợi ý, trong khi phần mã nguồn thuần túy chiếm khoảng 25MB

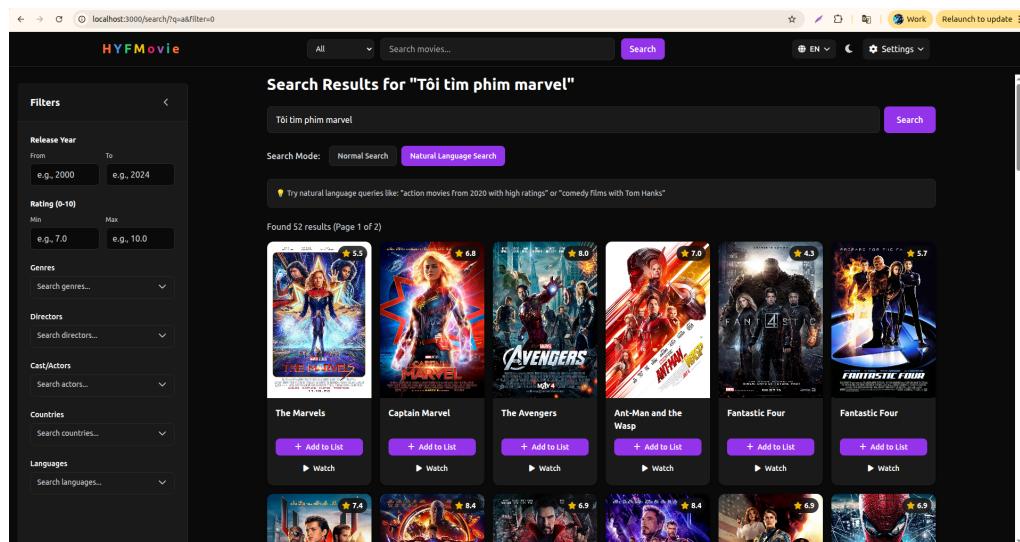
Nền tảng đã được kiện toàn với cấu trúc phân tầng rành mạch, đảm bảo sự khoa học trong việc phân chia giữa các gói và các lớp mã nguồn. Những số liệu thống kê về khối lượng mã và cấu trúc hệ thống là cơ sở quan trọng để đánh giá cũng như kiểm soát quá trình phát triển dự án. Hiện tại, hệ thống đã đáp ứng đầy đủ các tiêu chuẩn đề ra và hoàn toàn sẵn sàng cho việc vận hành trong môi trường thực tế.

Crawl dữ liệu từ trang web:

- Nguồn dữ liệu: IMDb, BoxOfficeMojo
- Số bộ phim đã crawl: 5000 bộ phim
- Số ảnh đã crawl: 250000 ảnh
- Số video đã crawl: 5000 video

Tìm kiếm phim theo ngôn ngữ người dùng:

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

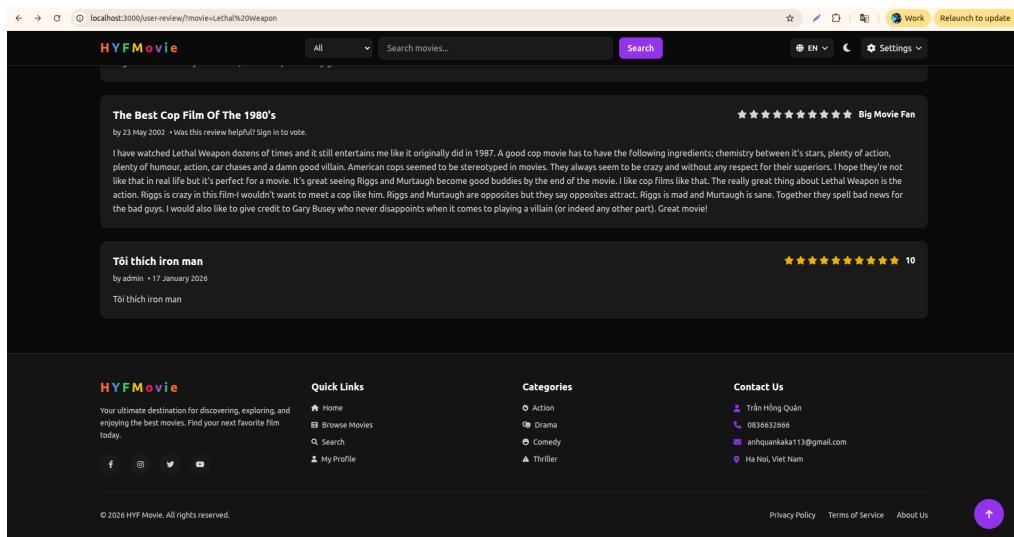


Hình 5.20: Chức năng tìm kiếm phim theo ngôn ngữ người dùng

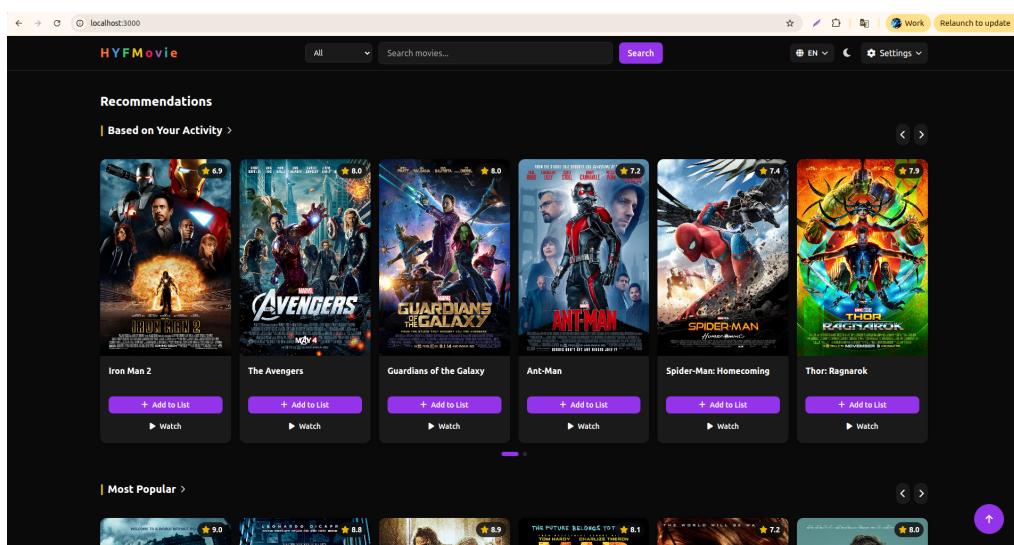
Dựa vào hình ảnh kết quả trên, có thể thấy hệ thống đã đưa ra kết quả phù hợp với yêu cầu của người dùng. Ở trên là các phim marvel hệ thống đã đưa ra kết quả.

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

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

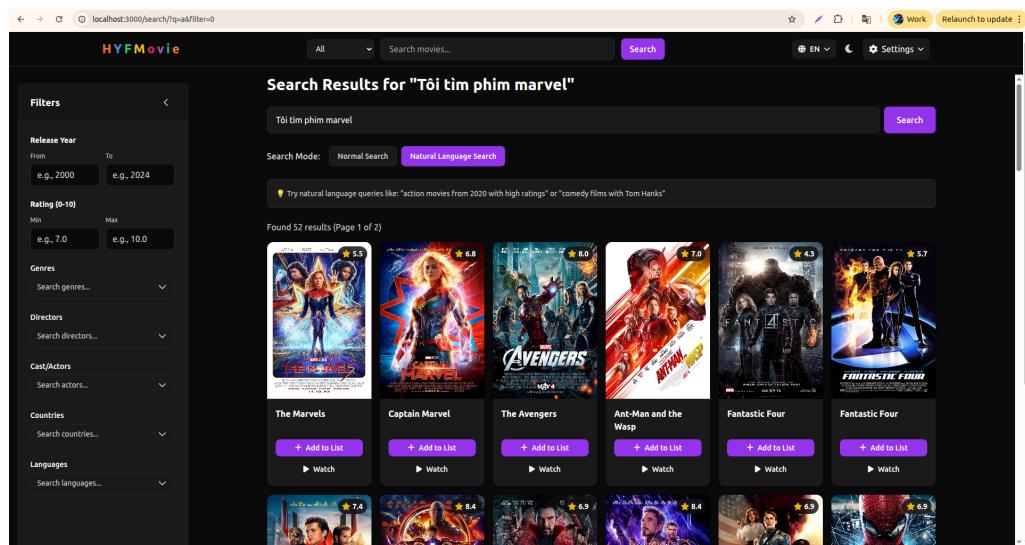


Hình 5.21: Review phim

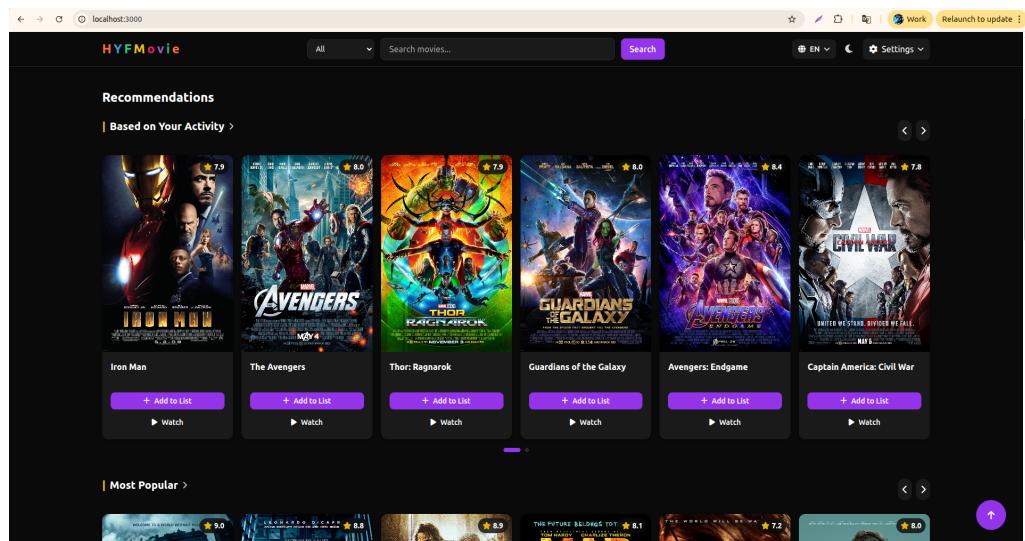


Hình 5.22: Gợi ý phim theo lịch sử review

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



Hình 5.23: Tìm kiếm phim

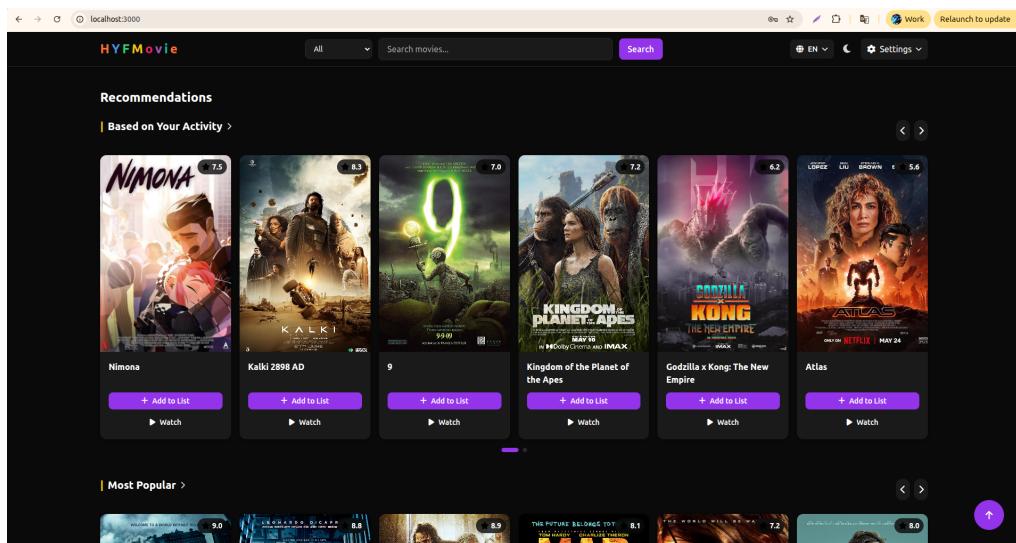


Hình 5.24: Gợi ý phim theo lịch sử tìm kiếm

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

5.3.3 Minh họa các chức năng chính

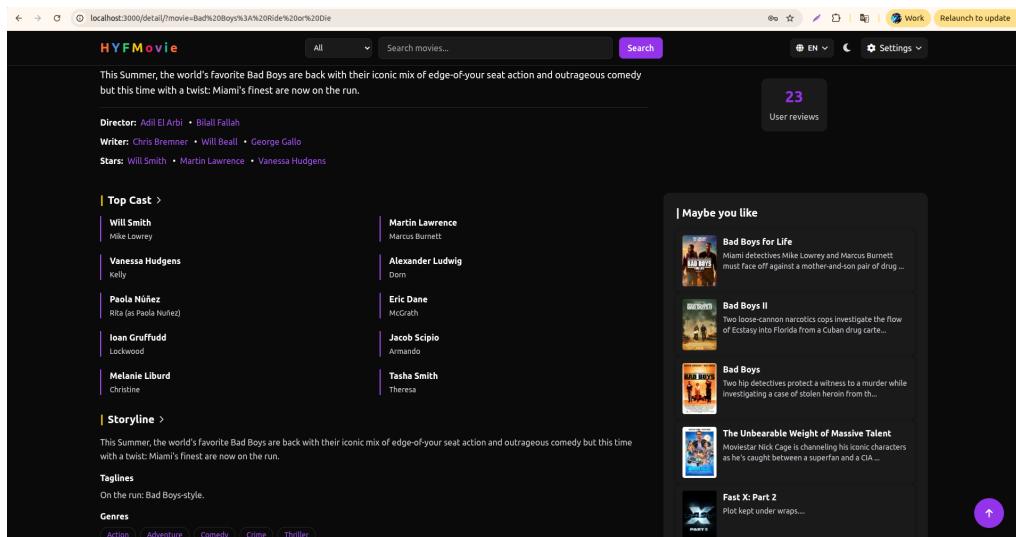
Chức năng gợi ý phim dựa vào lịch sử thao tác của người dùng



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

Dựa trên việc phân tích lịch sử xem phim của mỗi cá nhân, hệ thống sẽ tự động cập nhật các xu hướng thị trường và đề xuất những nội dung mới nhất phù hợp tối đa với sở thích riêng biệt của từng người dùng.

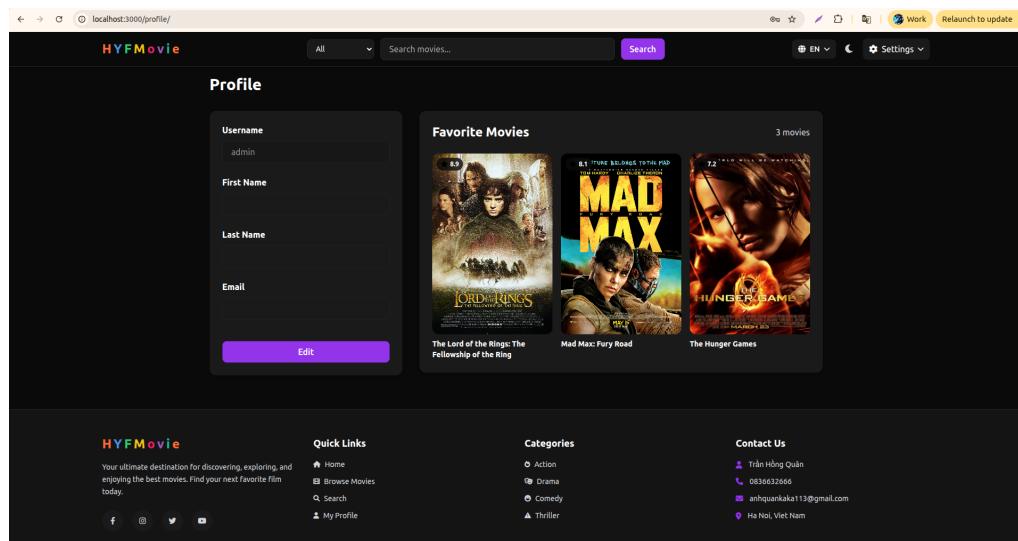
Chức năng gợi ý phim dựa vào nội dung phim đang xem



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

Hệ thống cho phép hiển thị các tác phẩm điện ảnh có đặc điểm tương đồng với bộ phim mà người dùng đang theo dõi hoặc tìm kiếm.

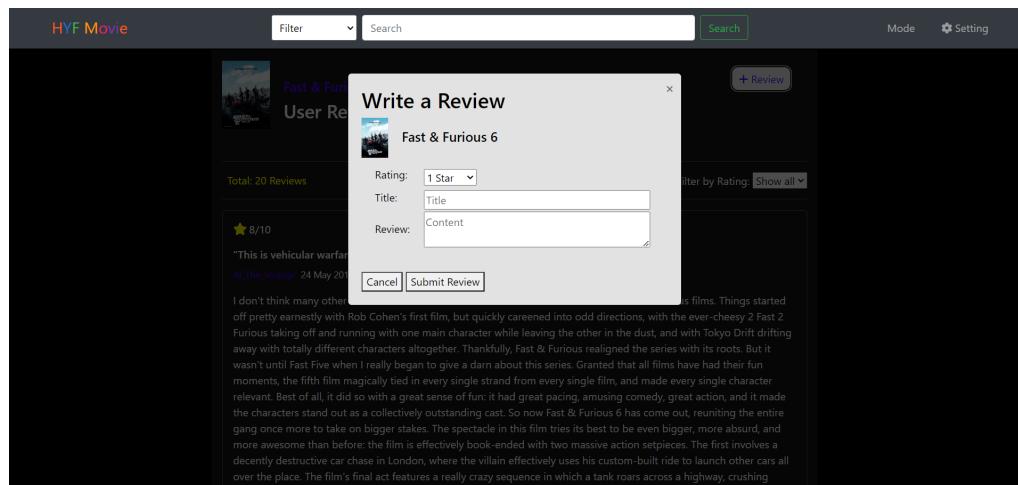
Chức năng thêm phim yêu thích vào danh sách



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

Thông qua công cụ thêm vào danh mục ưu tiên, người xem dễ dàng lưu trữ các tác phẩm điện ảnh tâm đắc vào kho lưu trữ riêng ngay khi tìm thấy nội dung ưng ý.

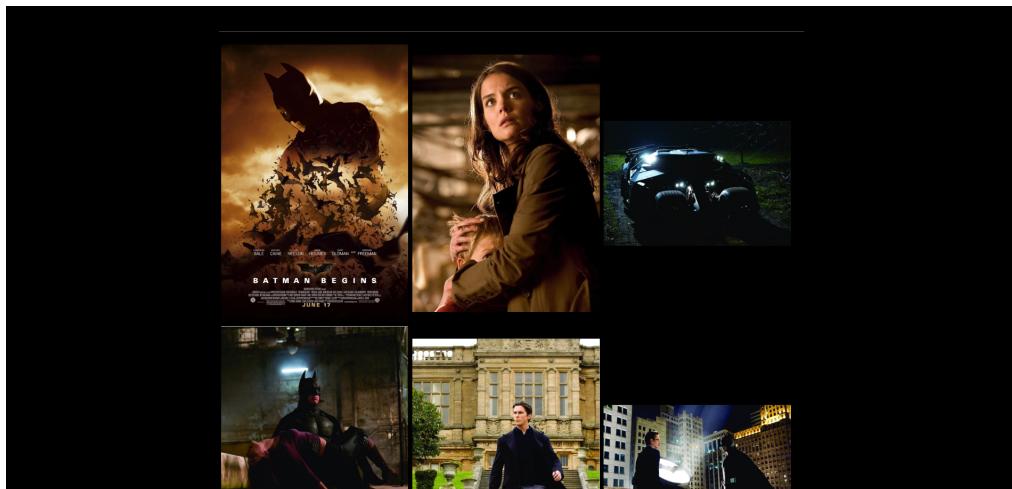
Chức năng review phim



Hình 5.28: Chức năng review phim

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

Chức năng xem toàn bộ ảnh của phim



Hình 5.29: Chức năng xem toàn bộ ảnh của phim

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

5.4 Kiểm thử

Bảng 5.3: Kiểm thử, đánh giá chức năng 1

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

Bảng 5.4: Kiểm thử, đánh giá chức năng 2

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

Bảng 5.5: Bảng danh sách kiểm thử hiệu năng

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

Bảng 5.6: Bảng kết quả kiểm thử hiệu năng

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

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

5.5 Triển khai

Mô hình triển khai Hệ thống được vận hành theo kiến trúc client-server, với giao diện người dùng (frontend) được khởi chạy trên live server và bộ phận xử lý trung tâm (backend) được thiết lập thông qua Django trên máy trạm cá nhân.

Thông số thiết lập: Máy tính cá nhân đáp ứng các tiêu chuẩn sau:

- Hệ điều hành: Tương thích Windows, Linux hoặc MacOS
- Bộ vi xử lý: Tối thiểu Intel i3 hoặc các dòng CPU có hiệu năng tương đương
- Bộ nhớ trong (RAM): 4GB
- Lưu trữ: Ổ cứng SSD dung lượng 128GB
- Yêu cầu đường truyền: Kết nối Internet ổn định ở mức băng thông trung bình

Triển khai

1. Thực hiện clone repository từ GitHub về máy tính cá nhân.
2. Tiến hành cài đặt Docker vào máy tính.
3. Di chuyển vào thư mục dự án và thực thi lệnh docker-compose up -d để khởi chạy hệ thống. Sau khi hoàn thành, toàn bộ hệ thống đã được vận hành trên máy tính cá nhân. Truy cập vào cổng localhost:3000 để kiểm thử hệ thống.

Đánh giá kết quả vận hành thực nghiệm

Dù quá trình thử nghiệm hiện mới chỉ dừng lại ở phạm vi môi trường nội bộ (local), các chỉ số ghi nhận được cho thấy những tín hiệu khả quan:

Hiệu suất xử lý: Trong điều kiện vận hành nội bộ, dù chưa có dữ liệu đánh giá áp lực tải ở quy mô lớn, hệ thống vẫn duy trì được sự ổn định tuyệt vời với lưu lượng truy cập thử nghiệm thấp, không xảy ra hiện tượng nghẽn mạng hay suy giảm hiệu năng.

Tốc độ phản hồi: Hệ thống ghi nhận độ trễ trung bình dưới 2 giây cho các thao tác thông thường. Riêng các tác vụ phức tạp hơn như thuật toán gợi ý phim đạt tốc độ dưới 5 giây và chức năng truy vấn tìm kiếm phim hoàn tất trong khoảng dưới 10 giây.

Đánh giá từ người dùng: Nhóm tham gia thử nghiệm có những phản hồi rất tốt về tính tinh gọn, dễ tiếp cận của giao diện cũng như độ chính xác của công cụ tìm kiếm. Đáng chú ý, các kết quả từ hệ thống đề xuất và tìm kiếm đều bám sát và thỏa mãn đúng kỳ vọng của người trải nghiệm.

Việc chạy thử nghiệm trên môi trường local đã giúp xác thực thành công các tính năng nền tảng và nhận về những tương tác tích cực từ phía người dùng. Các số liệu cho thấy một hệ thống hoạt động tin cậy, tốc độ xử lý nhanh cùng giao diện tối ưu. Tuy nhiên, để có cái nhìn chuyên sâu hơn về ngưỡng chịu tải và hiệu năng thực tế, việc đưa hệ thống lên các máy chủ chuyên dụng là bước đi cần thiết tiếp theo.

CHƯƠNG 6. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT

Chương này tập trung phân tích các thách thức nảy sinh trong giai đoạn triển khai và những biện pháp kỹ thuật tương ứng đã được áp dụng. Đồng thời, chương cũng tổng kết các kết quả đạt được và những đóng góp then chốt trong suốt quá trình hoàn thiện đồ án tốt nghiệp.

6.1 Sự thiếu hụt trầm trọng về dữ liệu điện ảnh

Khó khăn này phát sinh khi dự án yêu cầu một kho dữ liệu khổng lồ để vận hành thuật toán gợi ý một cách hiệu quả. Để giải quyết, tôi đã tiến hành khảo sát và lựa chọn các nền tảng trực tuyến uy tín làm nguồn tài nguyên chính. Quy trình xử lý bao gồm việc bóc tách cú pháp web, thiết lập giao thức truy cập, phân tích sơ đồ cấu trúc dữ liệu và triển khai các công cụ thu thập tự động để tối ưu hóa thời gian.

6.2 Thách thức trong việc thiết kế cơ sở dữ liệu với hệ thống thuộc tính phức tạp

Việc quản lý một lượng lớn các trường thông tin liên quan đến tác phẩm và người dùng đã tạo ra áp lực lớn trong khâu thiết kế sơ đồ dữ liệu. Nhằm khắc phục tình trạng này, tôi đã áp dụng các quy chuẩn hóa dữ liệu và chia tách các thuộc tính thành các bảng phụ chuyên biệt. Sau đó, bằng cách xác lập các mối quan hệ và liên kết logic giữa các bảng, tôi đã xây dựng thành công một cấu trúc cơ sở dữ liệu khoa học, đảm bảo tính toàn vẹn và hiệu suất truy xuất cao.

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

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

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

Giới thiệu Github Action:

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

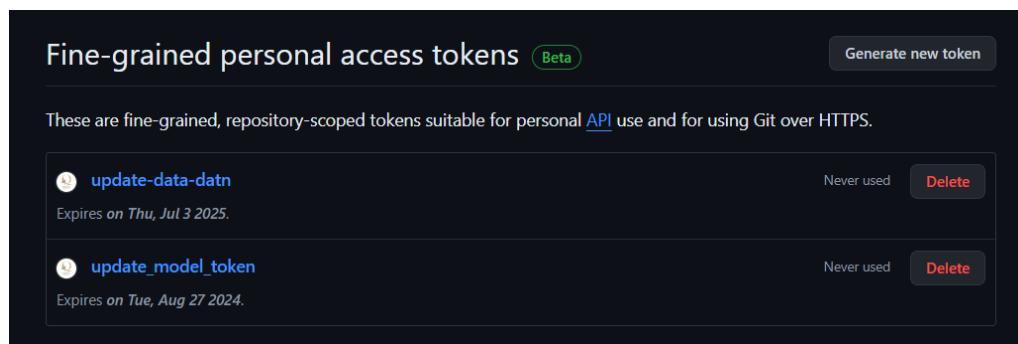
Triển khai:

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

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

1. Tạo PAT (Personal Access Token) Github

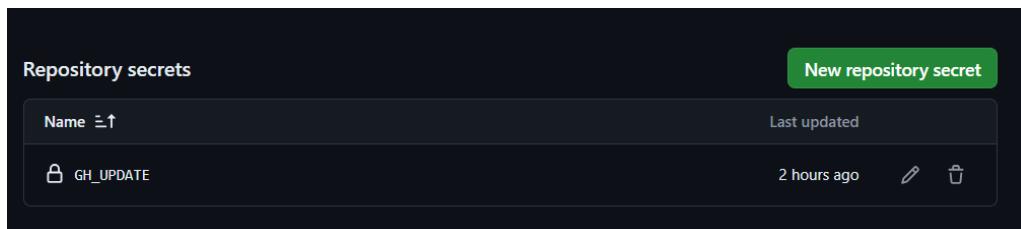
Personal Access Token (PAT) trên GitHub đóng vai trò như một phương thức xác thực, cho phép các ứng dụng hoặc dịch vụ trung gian truy cập vào tài khoản mà không cần tiết lộ mật khẩu chính. Trong kiến trúc này, đối tượng sử dụng mã PAT chính là các bot vận hành tự động.



Hình 6.1: Tạo PAT Github

2. Thiết lập Secrets Repository

Việc thiết lập GitHub Secrets nhằm mục tiêu bảo mật và quản trị tập trung các dữ liệu nhạy cảm như mã token, mật khẩu, hay khóa API trong suốt vòng đời phát triển dự án. Cơ chế này đảm bảo những thông tin quan trọng không bị công khai trên mã nguồn, đồng thời chỉ cho phép các quy trình tự động hóa như GitHub Actions hoặc các ứng dụng được ủy quyền truy xuất thông qua API an toàn.



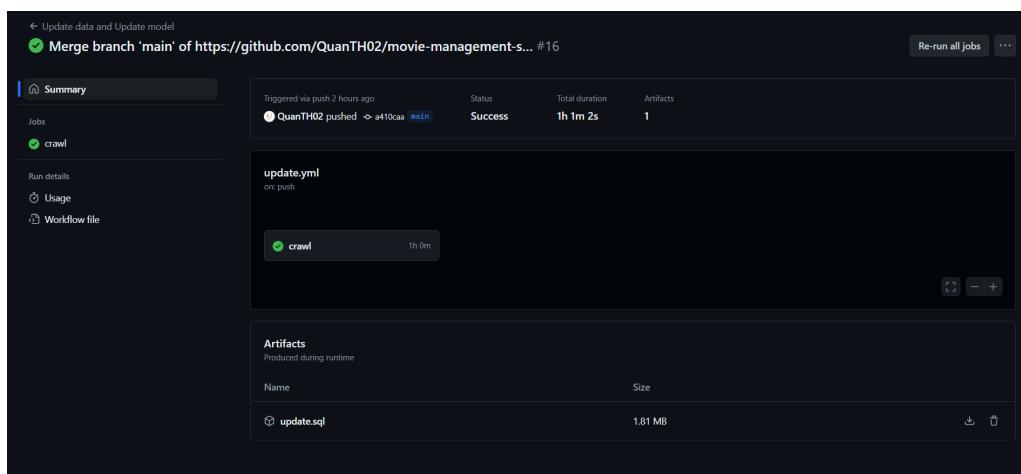
Hình 6.2: Thiết lập Secrets trên GitHub

3. Viết file update.yml

Giai đoạn cuối cùng, tôi thiết lập tệp cấu hình update.yml nhằm kích hoạt cơ chế vận hành tự động định kỳ cho GitHub Actions.

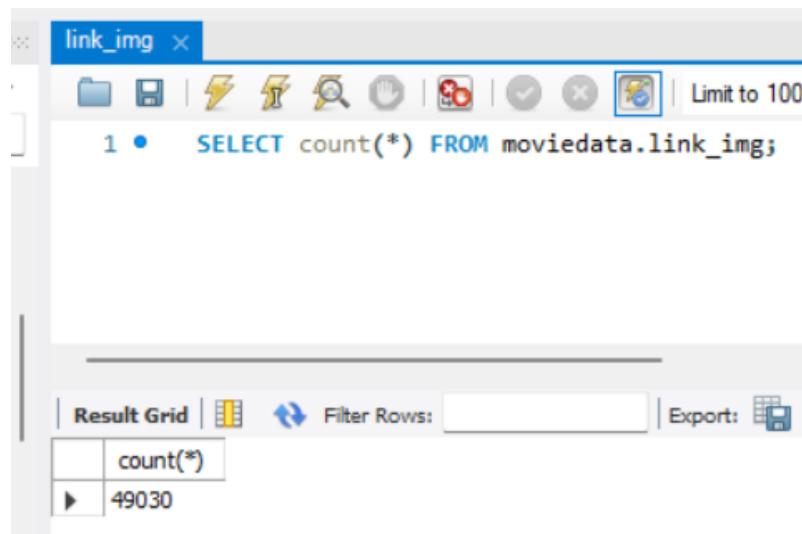
Kết quả:

Đúng 01:00 mỗi ngày, hệ thống sẽ tự động kích hoạt quy trình khai thác dữ liệu mới, kết xuất ra tệp update.sql và thực hiện cập nhật mã nguồn lên GitHub trong vòng một giờ đồng hồ.



Hình 6.3: Hệ thống tự động crawl dữ liệu

Ngay sau khi tệp update.sql được khởi tạo thành công, người dùng chỉ cần thực hiện thao tác kéo mã nguồn về máy chủ và khởi chạy tập lệnh Crawl/autoUpdate/update.py để cập nhật dữ liệu mới vào hệ thống tức thì.



The screenshot shows a MySQL Workbench interface with a query editor titled 'link_img'. The query is:

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

The results 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 xác định một phương pháp gợi ý tối ưu đóng vai trò then chốt và đầy thử thách trong quá trình xây dựng ứng dụng đề xuất điện ảnh. Sau khi nghiên cứu các công trình khoa học và tài liệu chuyên ngành, đồng thời tiến hành thực nghiệm để so sánh hiệu quả giữa các mô hình khác nhau, tôi đã quyết định lựa chọn thuật toán Content-based làm giải pháp chủ đạo cho dự án này.

Giới thiệu thuật toán Content-based:

Trong hệ thống lọc dựa trên nội dung này, tôi thiết lập một hồ sơ đặc trưng riêng biệt cho từng tác phẩm điện ảnh dựa trên các thuộc tính nội tại của chúng. Về mặt toán học, hồ sơ này được mô hình hóa thành một vectơ đặc trưng, đóng vai trò là cơ sở dữ liệu đầu vào để tính toán độ tương đồng giữa các bộ phim.

- Movie name: Tiêu đề của tác phẩm điện ảnh
- Genres: Phân loại thể loại mà tác phẩm thuộc về
- Describe: Nội dung tóm tắt về cốt truyện của tác phẩm
- Cast: Danh sách các nghệ sĩ tham gia diễn xuất trong tác phẩm
- Director: Người đảm nhiệm vai trò chỉ đạo sản xuất tác phẩm

Phân tích, giải pháp:

Content-based là phương thức gợi ý dựa trên việc đề xuất các đối tượng có những đặc tính tương đồng với đối tượng mà người dùng đã lựa chọn trước đó. Do vậy, trong phạm vi bài toán này, tôi đã thực hiện tính toán mức độ tương quan giữa các tác phẩm điện ảnh dựa trên tập hợp các vectơ đặc trưng đã xác định ở trên.

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

Trong đó:

- A và B là hai vector biểu diễn cho hai tác phẩm điện ảnh hoặc hai đối tượng người dùng.
- $A \cdot B$ là giá trị tích vô hướng được tính từ hai vector A và B .
- $\|A\|$ và $\|B\|$ tương ứng là giá trị chuẩn (norm) của vector A và vector B .

Kiểm tra tính chính xác:

Để đánh giá độ tin cậy của mô hình sau khi áp dụng công thức tính toán độ tương đồng, tôi tiến hành quy trình kiểm chứng bằng cách đo lường mức độ tương quan trên từng thuộc tính độc lập như tên phim, thể loại, dàn diễn viên, đạo diễn và phần mô tả. Cuối cùng, tôi thực hiện đối chiếu kết quả tổng hợp với các chỉ số riêng lẻ này để xác nhận tính nhất quán và độ chính xác của thuật toán.

```

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
9 199 0.21987801134666984
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 0.04686329421881832
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 0.13878049563162884
33 63 0.10159744616742258
34 58 0.08755433976577434
35 48 0.083174809398962
36 80 0.0671573133762048
37 129 0.06481968158583065
38 87 0.06434887142048598
39 43 0.06091447365093486
40 10 0.05759624460829198
41 21 0.05458436118761168
42
43

cast Untitled-2
1 cast
2 109 0.3647058800017843
3 21 0.3387079282444981
4 48 0.1644614390256814
5 128 0.10773013419658836
6 26 0.1055610662972952
7 78 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 Blackhart Terence Rosemore Maria Bakalova Sean Gunn Sarah Alami Jasmine Munoz Chris Pratt Giovannie Cruz
30
31 Chris Pratt
32 Bradley Cooper
33 Dave Bautista
34 Karen Gillan
35 Vin Diesel
36 Sean Gunn
37
38

```

Hình 6.5: Kiểm tra kết quả 1

Từ hình ảnh 6.5, có thể nhận thấy rằng bộ phim được hệ thống đề xuất có mức độ tương đồng cao nhất với bộ phim *Guardians of the Galaxy* chính là bộ phim *Guardians of the Galaxy Vol. 3*. Hai bộ phim này có sự tương đồng về tên phim,

mô tả, diễn viên và đạo diễn.

The image shows two side-by-side code editors. The left editor is titled 'Movie id: 132 Untitled-1' and contains the following text:

```

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 38 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 0.3480833324437175
33 9 0.1252903952739983
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
41 45 0.0

```

The right editor is titled 'cast Untitled-2' and contains the following text:

```

1 cast
2 25 0.49733589592274985
3 94 0.07541231794535201
4 30 0.0704472833428879
5 140 0.057882884735789555
6 74 0.053418775792282375
7 61 0.05041770984609579
8 107 0.049902341095944186
9 79 0.04983486654944694
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

```

At the bottom of the right editor, there are status indicators: In 45, Col 1, Spaces: 4, UTF-8, CRLF, Markdown, Go Live, tabnine starter.

Hình 6.6: Kiểm tra kết quả 2

Từ hình ảnh 6.6, có thể quan sát thấy bộ phim được hệ thống đề xuất có mức độ tương đồng cao nhất với bộ phim The Hunger Games: Catching Fire là bộ phim The Hunger Games. Hai bộ phim này thể hiện sự tương đồng đáng kể về tên phim, mô tả, diễn viên và đạo diễn.

```

1 Movie id: 15 Untitled-1
2 Movie name: Meg 2: The Trench
3 Describe: A research team encounters multiple threats while
exploring the depths of the ocean, including a malevolent mining
operation.
4 Genres: Action Adventure Horror Sci-Fi Thriller
5 Director: Ben Wheatley
6 Cast: Jason Statham Jing Wu Shuya Sophia Cai Cliff Curtis
Page Kennedy Sergio Peris-Mencheta Skyler Samuels Melissanthi
Mahut Whoopi Van Raam Kiran Sonia Sawar Felix Muyr Sienna
Guillory Tao Guo Robin Hill Lele Dai
7 Movie id result:
8 88 0.5504300169597662
9 38 0.35319267661183157
10 2 0.33398033722691944
11 105 0.322398328712102
12 9 0.30436133874364887
13 75 0.30063584739993846
14 61 0.2873170128093882
15 148 0.28669542992067987
16 54 0.2861652412081092
17 34 0.28115353309874935
18
19
20 genres
21 88 0.932862414224352
22 38 0.8304865093064482
23 105 0.7549104158987879
24 2 0.7320425466125233
25 22 0.7256822305611071
26 61 0.7256822305611071
27 75 0.7256822305611071
28 148 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 0.073682508800623825
37 84 0.07172309901182033
38 42 0.06981308844848848
39 102 0.06898968252852719
40 103 0.06898968252852719
41 4 0.05255358144650688
42

```

```

1 cast Untitled-2
2 80 0.22183861524280854
3 54 0.05523016212903862
4 18 0.05310982691925622
5 32 0.052348176714640225
6 51 0.0468198564084982865
7 46 0.046561172845036136
8 64 0.04160235447938892
9 56 0.04011477030059863
10 99 0.03699776836648769
11 8 0.03424991292951911
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 Ólafur Darri Ólafsson Jessica
McNamee Masi Oka Raymond Vinter Hongmei Mai Wei Yi
30
31
32
33

```

Hình 6.7: Kiểm tra kết quả 3

Dựa trên kết quả hiển thị tại hình 6.7, có thể thấy *The Meg* là tác phẩm được hệ thống đánh giá có mức độ tương quan cao nhất đối với bộ phim *Meg 2: The Trench*. Sự tương đồng này được thể hiện rõ nét qua các yếu tố như tiêu đề, cốt truyện mô tả, cùng sự trùng khớp về đội ngũ diễn viên và đạo diễn.

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

CHƯƠNG 7. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

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

Kết quả so sánh

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

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

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

Phân tích quá trình thực hiện

Trong suốt quá trình triển khai và hoàn thiện đồ án tốt nghiệp, tôi đã đạt được những kết quả trọng tâm như sau:

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

Những hạn chế còn tồn tại

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

Đóng góp nổi bật

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

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

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

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

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

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

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

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

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

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

TÀI LIỆU THAM KHẢO

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

PHỤ LỤC