

**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

<b>CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....</b>	<b>1</b>
1.1 Đặt vấn đề.....	1
1.2 Mục tiêu và phạm vi đề tài.....	2
1.3 Định hướng giải pháp.....	2
1.4 Bố cục đồ án .....	3
<b>CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU.....</b>	<b>5</b>
2.1 Khảo sát hiện trạng .....	5
2.2 Khảo sát về quyền theo dõi .....	6
2.3 Tổng quan chức năng .....	7
2.3.1 Biểu đồ use case tổng quát .....	7
2.3.2 Biểu đồ use case phân rã Quản lý thông tin người dùng .....	9
2.3.3 Biểu đồ use case phân rã Quản lý thông tin phim .....	10
2.3.4 Biểu đồ use case phân rã Quản lý tài khoản .....	10
2.3.5 Biểu đồ use case phân rã Tương tác với phim.....	11
2.3.6 Biểu đồ use case phân rã Review, comment .....	11
2.4 Đặc tả chức năng .....	12
2.4.1 Đặc tả use case Đăng nhập.....	12
2.4.2 Đặc tả use case Chính sửa thông tin cá nhân .....	13
2.4.3 Đặc tả use case Tìm kiếm phim .....	14
2.4.4 Đặc tả use case Đánh dấu phim mình thích.....	15
2.4.5 Đặc tả use case Đánh giá phim .....	15
2.4.6 Đặc tả use case CRUD phim .....	17
2.4.7 Đặc tả use case Gợi ý phim theo nội dung .....	18
2.5 Yêu cầu phi chức năng .....	18

## **CHƯƠNG 3. CƠ SỞ LÝ THUYẾT CHO GỢI Ý PHIM..... 20**

3.1 Content-Based Filtering .....	20
3.1.1 Khái niệm .....	20
3.1.2 Các thuật toán.....	20

## **CHƯƠNG 4. CÔNG NGHỆ SỬ DỤNG..... 22**

4.1 Công nghệ sử dụng phía frontend .....	22
4.1.1 React .....	22
4.1.2 Tailwind CSS .....	22
4.2 Công nghệ sử dụng phía backend .....	23
4.2.1 Django .....	23
4.2.2 MySQL.....	23
4.3 Công nghệ sử dụng trong việc lưu trữ code .....	24
4.3.1 Git .....	24
4.3.2 GitHub.....	24
4.4 Công nghệ sử dụng trong việc crawl dữ liệu .....	25
4.4.1 Selenium và Requests .....	25
4.4.2 Công nghệ sử dụng để update dữ liệu.....	25

## **CHƯƠNG 5. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG .... 27**

5.1 Thiết kế kiến trúc.....	27
5.1.1 Lựa chọn kiến trúc phần mềm .....	27
5.1.2 Thiết kế tổng quan.....	28
5.1.3 Thiết kế chi tiết gói .....	30
5.2 Thiết kế chi tiết.....	33
5.2.1 Thiết kế giao diện .....	33
5.2.2 Thiết kế cơ sở dữ liệu .....	39
5.2.3 Thiết kế hệ thống crawl dữ liệu.....	40

5.2.4 Tìm kiếm phim bằng ngôn ngữ tự nhiên .....	41
5.2.5 Gợi ý phim bằng genAI dựa vào lịch sử hoạt động của người dùng .	43
5.3 Xây dựng ứng dụng.....	45
5.3.1 Thư viện và công cụ sử dụng .....	45
5.3.2 Kết quả đạt được .....	46
5.3.3 Minh họa các chức năng chính .....	49
5.4 Kiểm thử.....	52
5.5 Triển khai .....	55
<b>CHƯƠNG 6. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT .....</b>	<b>57</b>
6.1 Thiếu khá nhiều dữ liệu phim ảnh .....	57
6.2 Thiết kế database khá khó khăn với quá nhiều trường .....	57
6.3 Trang web có dữ liệu update liên tục theo ngày khiến link ảnh và trailer bị hỏng.....	57
6.4 Lựa chọn thuật toán gợi ý phù hợp.....	60
<b>CHƯƠNG 7. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....</b>	<b>64</b>
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>67</b>



## DANH MỤC HÌNH VẼ

Hình 2.1	Khảo sát người dùng về các ứng dụng . . . . .	5
Hình 2.2	Khảo sát về quyền theo dõi . . . . .	6
Hình 2.3	Use case tổng quan . . . . .	9
Hình 2.4	Phân rã use case Quản lý thông tin người dùng . . . . .	9
Hình 2.5	Phân rã use case Quản lý thông tin phim . . . . .	10
Hình 2.6	Phân rã use case Quản lý tài khoản . . . . .	10
Hình 2.7	Phân rã use case Tương tác với phim . . . . .	11
Hình 2.8	Phân rã use case Review, comment . . . . .	11
Hình 5.1	Biểu đồ phụ thuộc gói . . . . .	28
Hình 5.2	Thiết kế chi tiết gói ở tầng Model . . . . .	30
Hình 5.3	Thiết kế chi tiết gói ở tầng View . . . . .	31
Hình 5.4	Thiết kế chi tiết gói ở tầng Template . . . . .	33
Hình 5.5	Màn hình <i>đăng nhập</i> . . . . .	34
Hình 5.6	Màn hình <i>đăng ký</i> . . . . .	35
Hình 5.7	Màn hình <i>profile</i> . . . . .	35
Hình 5.8	Màn hình <i>chính</i> . . . . .	36
Hình 5.9	Màn hình <i>gọi ý phim ở trang chủ</i> . . . . .	36
Hình 5.10	Màn hình <i>thông tin chi tiết 1</i> . . . . .	37
Hình 5.11	Màn hình <i>thông tin chi tiết 2</i> . . . . .	37
Hình 5.12	Màn hình <i>tổng hợp đạo diễn, diễn viên, ...</i> . . . . .	38
Hình 5.13	Màn hình <i>tổng hợp đạo diễn, diễn viên, ...</i> . . . . .	38
Hình 5.14	Màn hình <i>review phim</i> . . . . .	39
Hình 5.15	Sơ đồ ERD . . . . .	39
Hình 5.16	Hình ảnh các table trong database 1 . . . . .	40
Hình 5.17	Hình ảnh các table trong database 2 . . . . .	40
Hình 5.18	Hình ảnh các table trong database 3 . . . . .	40
Hình 5.19	Chức năng tìm kiếm phim theo ngôn ngữ người dùng . . . . .	47
Hình 5.20	Review phim . . . . .	48
Hình 5.21	Gợi ý phim theo lịch sử review . . . . .	48
Hình 5.22	Tìm kiếm phim . . . . .	49
Hình 5.23	Gợi ý phim theo lịch sử tìm kiếm . . . . .	49
Hình 5.24	Chức năng gợi ý phim dựa vào lịch sử thao tác của người dùng	50
Hình 5.25	Chức năng gợi ý phim dựa vào nội dung phim đang xem . . . . .	50
Hình 5.26	Chức năng thêm phim vào danh sách yêu thích . . . . .	51

Hình 5.27	Chức năng review phim . . . . .	51
Hình 5.28	Chức năng xem toàn bộ ảnh của phim . . . . .	52
Hình 6.1	Tạo PAT Github . . . . .	58
Hình 6.2	Tạo Secrets Repository . . . . .	58
Hình 6.3	Hệ thống tự động crawl dữ liệu . . . . .	59
Hình 6.4	Cập nhật dữ liệu mới thành công . . . . .	60
Hình 6.5	Kiểm tra kết quả 1 . . . . .	61
Hình 6.6	Kiểm tra kết quả 2 . . . . .	62
Hình 6.7	Kiểm tra kết quả 3 . . . . .	63

## **DANH MỤC BẢNG BIỂU**

Bảng 2.1	Khảo sát các hệ thống đã có . . . . .	7
Bảng 2.2	Đặc tả use case Đăng nhập . . . . .	12
Bảng 2.3	Đặc tả use case Chính sửa thông tin cá nhân . . . . .	13
Bảng 2.4	Đặc tả use case Tìm kiếm phim . . . . .	14
Bảng 2.5	Đặc tả use case Đánh dấu phim mình thích . . . . .	15
Bảng 2.6	Đặc tả use case Đánh giá phim . . . . .	16
Bảng 2.7	Đặc tả use case CRUD phim . . . . .	17
Bảng 2.8	Đặc tả use case Gợi ý phim theo nội dung . . . . .	18
Bảng 5.1	Tổng hợp các thông tin cần lấy . . . . .	41
Bảng 5.2	Danh sách thư viện và công cụ sử dụng . . . . .	46
Bảng 5.3	Kiểm thử chức năng . . . . .	53
Bảng 5.4	Kiểm thử chức năng . . . . .	53
Bảng 5.5	Bảng danh sách kiểm thử hiệu năng . . . . .	54
Bảng 5.6	Bảng kết quả kiểm thử hiệu năng . . . . .	55
Bảng 7.1	Kết quả so sánh so với các ứng dụng khác . . . . .	64

## DANH MỤC THUẬT NGỮ VÀ TỪ VIỆT TẮT

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

# CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

## 1.1 Đặt vấn đề

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

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

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

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

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

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

## 1.2 Mục tiêu và phạm vi đề tài

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

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

## 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 kém:** Cải thiện hiệu năng của hệ thống bằng cách rút gọn mã nguồn, thông tin, tối ưu hóa các thuật toán và sử dụng các kỹ thuật lập trình hiệu quả nhất.
- **Thiếu hệ thống gợi ý phim:** Sử dụng thuật toán cho hệ thống gợi ý phim: Content-Based Filtering. Content-Based Filtering sẽ được áp dụng khi người dùng xem thông tin về một bộ phim cụ thể, từ đó gợi ý các bộ phim có nội dung tương tự.
- **Chức năng bị phân tán:** Phát triển một nền tảng web tích hợp các tính năng

quan trọng như xem chi tiết phim, gợi ý phim cá nhân hóa, đánh giá và review từ cộng đồng, cùng với việc xem trailer và hình ảnh phim.

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

#### 1.4 **Bố cục đồ án**

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

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

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

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

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

Chương 6 tôi sẽ nêu ra những giải pháp và đóng góp nổi bật trong quá trình làm sản phẩm này. Đồng thời phân tích bài toán gợi ý phim một cách đầy đủ nhất.

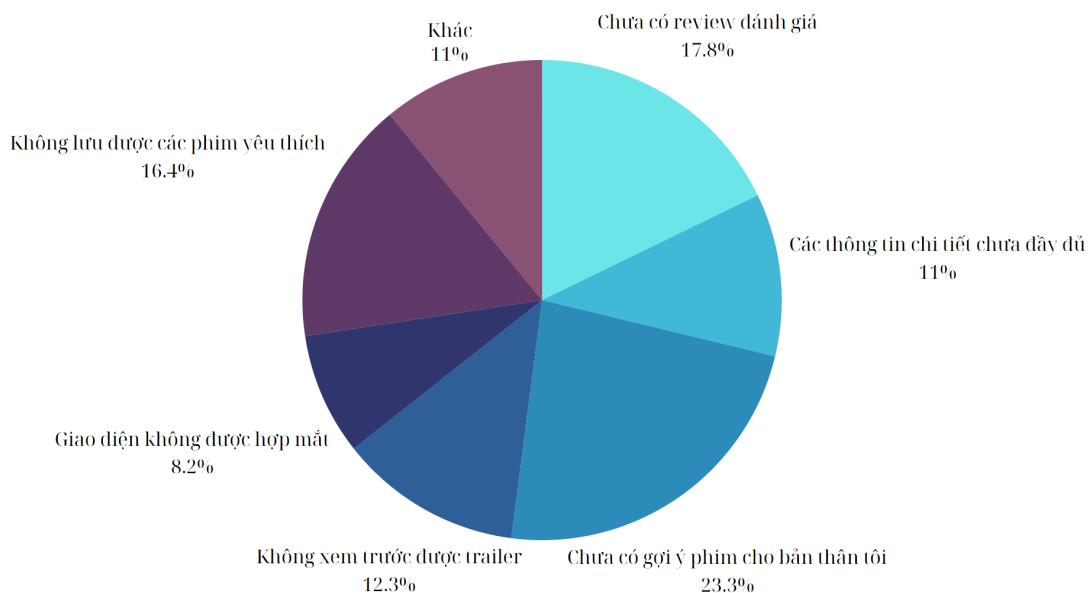
Chương 7 tôi sẽ kết luận và đề xuất hướng phát triển. Tôi sẽ tóm tắt lại những kết quả chính đã đạt được trong đồ án, đồng thời đề xuất các hướng phát triển tiềm

năng cho hệ thống trong tương lai. Chương này cung cấp cái nhìn tổng quát về đóng góp của đồ án và các bước thực hiện tiếp theo để nâng cao hệ thống.

## CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

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

### 2.1 Khảo sát hiện trạng



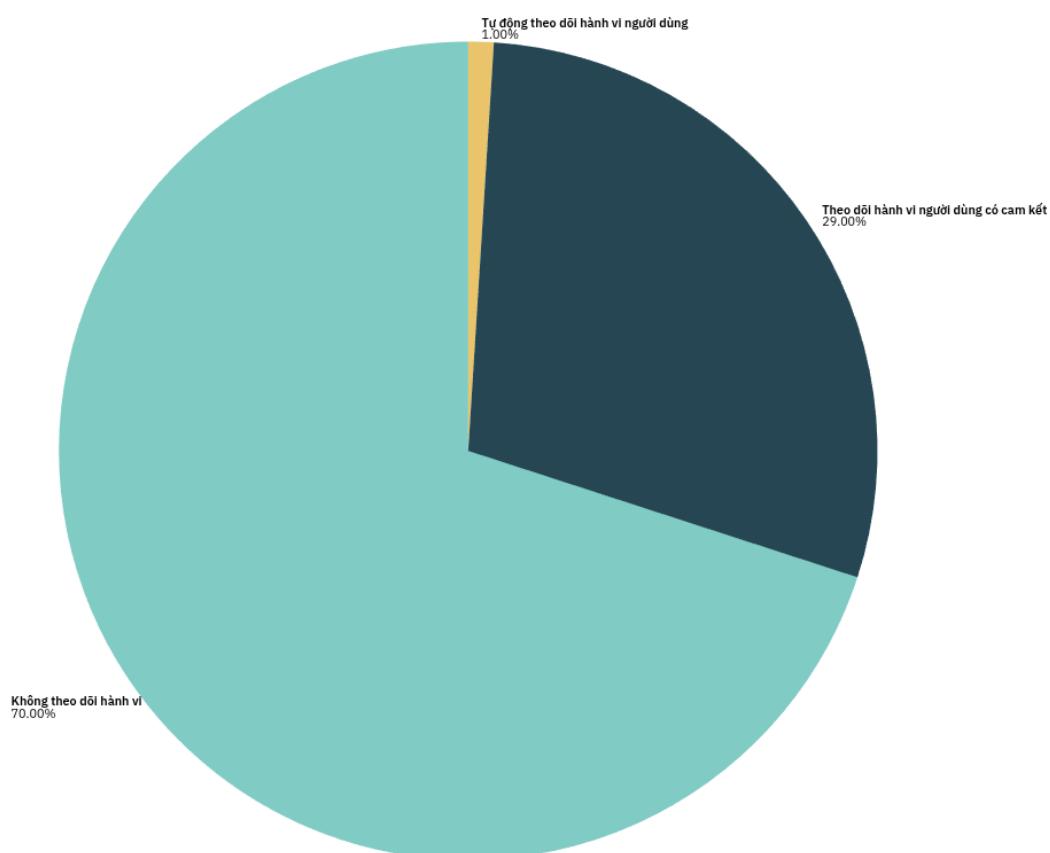
**Hình 2.1:** Khảo sát người dùng về các ứng dụng

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

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

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

## 2.2 Khảo sát về quyền theo dõi



**Hình 2.2:** Khảo sát về quyền theo dõi

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

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

**Bảng 2.1:** Khảo sát các hệ thống đã có

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

Cuối cùng, tôi cũng đã khảo sát một số ứng dụng giới thiệu phim trên nền tảng di động và web, từ đó nhận thấy một số điểm chung về giao diện thân thiện và tính năng gợi ý phim dựa trên hành vi người dùng như trong Table 2.1. Tuy nhiên, nhiều ứng dụng vẫn cần cải thiện về mặt hiệu năng và tính năng phong phú.

Dựa trên kết quả khảo sát từ các nguồn trên, tôi xác định rõ các tính năng cần thiết để phát triển một ứng dụng gợi ý phim toàn diện, đáp ứng tốt nhu cầu của người dùng.

## 2.3 Tổng quan chức năng

### 2.3.1 Biểu đồ use case tổng quát

Các tác nhân tham gia:

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

### Vai Trò của Mỗi Tác Nhân:

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

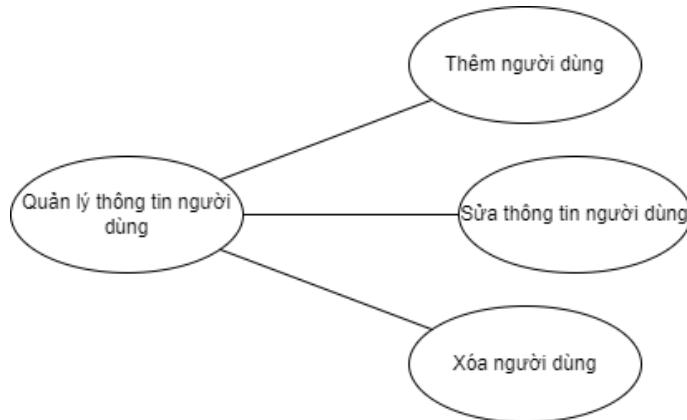
### Các use case chính:

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



**Hình 2.3:** Use case tổng quan

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



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

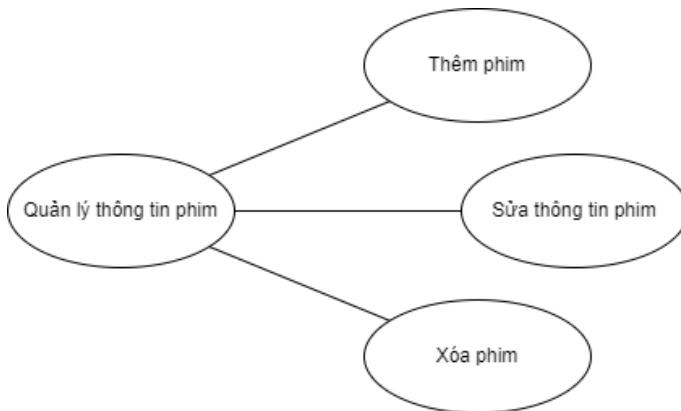
Trong biểu đồ use case phân rã "Quản lý thông tin người dùng", các use case được phân rã và mô tả như sau:

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

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

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

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

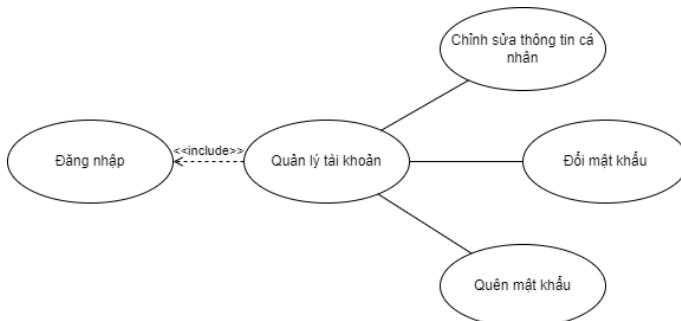


**Hình 2.5:** Phân rã use case Quản lý thông tin phim

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

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

### 2.3.4 Biểu đồ use case phân rã Quản lý tài khoản

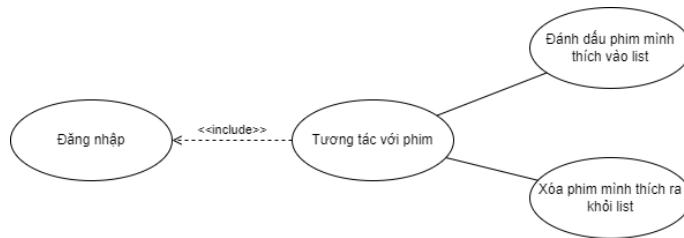


**Hình 2.6:** Phân rã use case Quản lý tài khoản

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

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

### 2.3.5 Biểu đồ use case phân rã Tương tác với phim



**Hình 2.7:** Phân rã use case Tương tác với phim

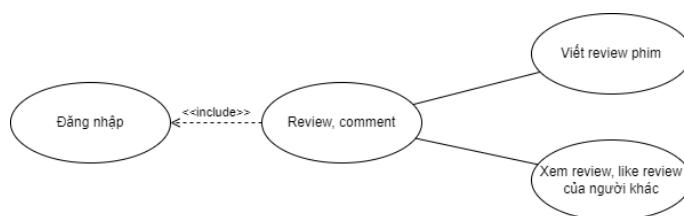
Trong biểu đồ use case phân rã "Tương tác với phim", các use case được phân rã và mô tả như sau:

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

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

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

### 2.3.6 Biểu đồ use case phân rã Review, comment



**Hình 2.8:** Phân rã use case Review, comment

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

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

## 2.4 Đặc tả chức năng

### 2.4.1 Đặc tả use case Đăng nhập

<b>Mã Use case</b>	UC002		<b>Tên Use case</b>	Đăng nhập
<b>Tác nhân</b>	User			
<b>Tiền điều kiện</b>	Người dùng đã có tài khoản trên hệ thống.			
<b>Luồng sự kiện chính (Thành công)</b>	STT	Thực hiện bởi	<b>Hành động</b>	
	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.	
<b>Luồng sự kiện thay thế</b>	STT	Thực hiện bởi	<b>Hành động</b>	
	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.	
<b>Hậu điều kiện</b>				

Bảng 2.2: Đặc tả use case Đăng nhập

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

#### 2.4.2 Đặc tả use case Chính sửa thông tin cá nhân

<b>Mã Use case</b>	UC009		<b>Tên Use case</b>	Chỉnh sửa thông tin cá nhân
<b>Tác nhân</b>	Guest, Admin			
<b>Tiêu điều kiện</b>	Người dùng đã đăng nhập vào hệ thống và có thông tin cá nhân cần cập nhật.			
<b>Luồng sự kiện chính (Thành công)</b>	STT	Thực hiện bởi	<b>Hành động</b>	
	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.	
<b>Luồng sự kiện thay thế</b>	STT	Thực hiện bởi	<b>Hành động</b>	
	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.	
<b>Hậu điều kiện</b>				

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

Bảng trên (Table 2.3) là đặc tả use case Chính sửa thông tin cá nhân từ người dùng. Trong đó, người dùng có thể chỉnh sửa thông tin cá nhân của mình như tên, địa chỉ email, ảnh đại diện, ... Hệ thống sẽ kiểm tra thông tin cá nhân của người dùng và trả về kết quả chỉnh sửa. Sẽ validate thông tin cá nhân để tránh lỗi input không hợp lệ.

### 2.4.3 ĐẶC TẢ USE CASE TÌM KIẾM PHIM

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

Bảng 2.4: Đặc tả use case Tìm kiếm phim

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

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

#### 2.4.4 Đặc tả use case Đánh dấu phim mình thích

<b>Mã Use case</b>	UC010	<b>Tên Use case</b>	Đánh dấu phim mình thích
<b>Tác nhân</b>	User		
<b>Tiền điều kiện</b>	Người dùng đã đăng nhập vào hệ thống.		
<b>Luồng sự kiện chính (Thành công)</b>	<b>STT</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>
	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
<b>Luồng sự kiện thay thế</b>	<b>STT</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>
<b>Hậu điều kiện</b>			

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

Bảng trên (Table 2.5) là đặc tả use case Đánh dấu phim mình thích từ người dùng. Trong đó, người dùng có thể đánh dấu phim mình thích vào danh sách riêng của mình để dễ dàng theo dõi sau này và hệ thống có thể đưa ra gợi ý những bộ phim có khả năng mà người dùng thích.

#### 2.4.5 Đặc tả use case Đánh giá phim

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

**Bảng 2.6:** Đặc tả use case Đánh giá phim

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

### 2.4.6 ĐẶC TẢ USE CASE CRUD PHIM

<b>Mã Use case</b>	UC013	<b>Tên Use case</b>	CRUD phim
<b>Tác nhân</b>	Admin		
<b>Tiền điều kiện</b>	Người dùng đăng nhập với vai trò Admin		
<b>Sửa:</b>			
<b>Luồng sự kiện chính</b>	<b>STT</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>
	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
<b>Luồng sự kiện thay thế</b>	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
<b>Xoá:</b>			
<b>Luồng sự kiện chính</b>	<b>STT</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>
	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
<b>Luồng sự kiện thay thế</b>	4a	Hệ thống	Kết thúc use case nếu Admin xác nhận không xoá
<b>Thêm:</b>			
<b>Luồng sự kiện chính</b>	<b>STT</b>	<b>Thực hiện bởi</b>	<b>Hành động</b>
	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
<b>Luồng sự kiện thay thế</b>	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
<b>Hậu điều kiện</b>	Không		

Bảng 2.7: Đặc tả use case CRUD phim

Bảng trên (Table 2.7) là đặc tả use case CRUD phim cho admin. Trong đó, admin có thể thêm, sửa, xóa phim vào danh sách phim của hệ thống. Hệ thống sẽ kiểm tra phim của admin và trả về kết quả CRUD.

#### 2.4.7 Đặc tả use case Gọi ý phim theo nội dung

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

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

Bảng trên (Table 2.8) là đặc tả use case Gọi 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

Trong phần này, tôi xác định được các yêu cầu phi chức năng và kỹ thuật cho ứng dụng như sau:

**Hiệu năng:** Ứng dụng cần đảm bảo hiệu suất cao và thời gian phản hồi nhanh chóng (với các CRUD thông thường thời gian phản hồi < 3s, còn với chức năng tìm kiếm và gợi ý phim bằng AI thì thời gian phản hồi < 10s). Thời gian tải trang cần được giảm thiểu tối đa để tăng trải nghiệm người dùng.

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

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

**Cơ sở dữ liệu:** Cần sử dụng cơ sở dữ liệu để lưu trữ và quản lý thông tin phim và

người dùng một cách hiệu quả và linh hoạt. Ứng dụng cần hỗ trợ các thao tác truy vấn nhanh chóng.

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

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

### 3.1 Content-Based Filtering

#### 3.1.1 Khái niệm

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

#### 3.1.2 Các thuật toán

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

**TF-IDF (Term Frequency-Inverse Document Frequency):** Thuật toán này được sử dụng để đánh giá tầm quan trọng của từ khóa đó trong văn bản. Tần suất xuất hiện của từ trong văn bản (Term Frequency) và mức độ phổ biến của từ trong toàn bộ tập văn bản (Inverse Document Frequency) được kết hợp để xác định trọng số của từ khóa đó.

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

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

Chương này đã cung cấp một cái nhìn tổng quan và chi tiết về phương pháp gợi

ý phim: Content-Based Filtering. Qua các phần trình bày về kiến thức nền tảng, cơ sở lý thuyết, các thuật toán và phương pháp nghiên cứu, chúng ta đã hiểu rõ cách thức hoạt động, ưu điểm và hạn chế của phương pháp.

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

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

## CHƯƠNG 4. CÔNG NGHỆ SỬ DỤNG

### 4.1 Công nghệ sử dụng phía frontend

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

#### 4.1.1 React

React là một thư viện JavaScript mã nguồn mở được phát triển bởi Meta, được sử dụng để xây dựng giao diện người dùng dựa trên các thành phần. React sử dụng cơ chế Virtual DOM để tối ưu hóa việc cập nhật giao diện, cho phép xây dựng các ứng dụng web động và tương tác cao.

Tôi chọn sử dụng React vì nó cho phép chia nhỏ giao diện thành các thành phần độc lập, giúp dễ dàng quản lý, bảo trì và tái sử dụng mã nguồn. Ngoài ra, cơ chế ràng buộc dữ liệu một chiều (one-way data binding) và hệ sinh thái phong phú của React giúp việc phát triển các tính năng trở nên linh hoạt và hiệu quả hơn. Với cơ chế Virtual DOM, React đảm bảo ứng dụng hoạt động mượt mà ngay cả khi xử lý khối lượng dữ liệu phim ảnh lớn và các tương tác phức tạp từ người dùng.

Trong ứng dụng này, React được sử dụng để xây dựng toàn bộ giao diện frontend, bao gồm các tính năng như tìm kiếm phim, lọc phim, hiển thị danh sách phim, trang chi tiết phim, và các chức năng tương tác khác. Các component React được tổ chức theo cấu trúc module, giúp mã nguồn dễ đọc và bảo trì.

#### 4.1.2 Tailwind CSS

Tailwind CSS là một framework CSS theo hướng utility-first, cung cấp các lớp (classes) tiện ích cấp thấp để xây dựng giao diện trực tiếp trong mã nguồn mà không cần rời khỏi tệp HTML hoặc JSX. Khác với các framework truyền thống, Tailwind CSS không áp đặt các thành phần giao diện cố sẵn, cho phép kiểm soát thiết kế một cách linh hoạt.

Tôi chọn sử dụng Tailwind CSS vì nó giúp tối ưu hóa kích thước tệp CSS cuối cùng thông qua cơ chế loại bỏ mã không sử dụng, đồng thời hỗ trợ thiết kế đáp ứng một cách trực quan và nhanh chóng. Framework này giúp tôi có toàn quyền kiểm soát thiết kế và tùy biến giao diện hệ thống mà không bị ràng buộc bởi các thành phần cố sẵn.

Trong ứng dụng này, Tailwind CSS được sử dụng để xây dựng toàn bộ giao diện styling, bao gồm các component như nút bấm, form input, card hiển thị phim, và

layout responsive. Tailwind CSS đảm bảo giao diện hiện đại, tinh tế và tương thích tốt trên nhiều loại thiết bị, từ máy tính để bàn đến điện thoại di động.

## 4.2 Công nghệ sử dụng phía backend

### 4.2.1 Django

Django là một framework phát triển bằng Python, nổi tiếng với khả năng giúp xây dựng các ứng dụng web một cách nhanh chóng và hiệu quả. Django cung cấp một kiến trúc MVT (Model-View-Template) rõ ràng, giúp tổ chức mã nguồn một cách khoa học và logic. Framework này hỗ trợ tích hợp với nhiều cơ sở dữ liệu khác nhau và đi kèm với nhiều tính năng mạnh mẽ như ORM và hệ thống quản trị admin tự động.

Tôi chọn sử dụng Django vì nó giúp dễ dàng thao tác với cơ sở dữ liệu thông qua ORM mà không cần viết SQL thủ công, giảm thiểu thời gian phát triển và lỗi lập trình. Ngoài ra, Django còn hỗ trợ nhiều tính năng bảo mật tích hợp sẵn như chống tấn công CSRF, XSS hay SQL Injection, giúp đảm bảo an toàn cho hệ thống. Framework này cũng có cộng đồng lớn và tài liệu phong phú, giúp việc phát triển và khắc phục vấn đề trở nên dễ dàng hơn.

Trong ứng dụng này, Django được sử dụng để xây dựng toàn bộ phần backend, bao gồm các API endpoints để xử lý yêu cầu từ frontend, quản lý dữ liệu 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 cho phép giao tiếp giữa frontend React và backend Django.

### 4.2.2 MySQL

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở, được sử dụng rộng rãi trong các ứng dụng web vì tính ổn định và hiệu suất cao. MySQL hỗ trợ các tính năng mạnh mẽ như quản lý dữ liệu phức tạp, thực hiện truy vấn nhanh chóng, bảo mật cao, và khả năng mở rộng tốt. Hệ quản trị này sử dụng ngôn ngữ SQL chuẩn và hỗ trợ nhiều kiểu dữ liệu khác nhau.

Tôi chọn sử dụng MySQL vì nó phù hợp với yêu cầu 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 tính năng sao lưu và phục hồi dữ liệu tốt, giúp bảo vệ dữ liệu quan trọng của ứng dụng. Ngoài ra, MySQL tích hợp tốt với Django thông qua ORM, giúp việc phát triển trở nên thuận tiện hơn.

Trong ứng dụng này, MySQL được sử dụng để lưu trữ toàn bộ dữ liệu của hệ thống, bao gồm 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 an toàn và 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 hệ thống quản lý phiên bản phân tán, cho phép theo dõi sự thay đổi của mã nguồn trong quá trình phát triển. Git lưu trữ lịch sử thay đổi của mã nguồn dưới dạng các snapshot, cho phép quay lại bất kỳ phiên bản nào trước đó. Hệ thống này hoạt động trên mô hình phân tán, mỗi bản sao repository đều chứa toàn bộ lịch sử dự án.

Tôi chọn sử dụng Git vì nó giúp quản lý mã nguồn một cách hiệu quả, dễ dàng quay lại các phiên bản trước khi gặp lỗi hoặc cần so sánh thay đổi. Git cung cấp các tính năng mạnh mẽ như nhánh, hợp nhất và theo dõi lịch sử, giúp quản lý quá trình phát triển và xử lý xung đột mã nguồn một cách hiệu quả. Ngoài ra, Git cho phép phối hợp làm việc nhóm một cách mượt mà, mỗi thành viên có thể làm việc độc lập trên các nhánh khác nhau.

Trong ứng dụng này, Git được sử 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 giúp theo dõi mọi thay đổi trong quá trình phát triển và đảm bảo mã nguồn luôn được sao lưu an toàn.

#### 4.3.2 GitHub

GitHub là một dịch vụ lưu trữ mã nguồn dựa trên Git, cung cấp nền tảng trực tuyến để chia sẻ, lưu trữ và quản lý mã nguồn. GitHub cung cấp giao diện web thân thiện để quản lý các kho lưu trữ, theo dõi các vấn đề, yêu cầu tính năng và xem lại mã nguồn. Nền tảng này cũng hỗ trợ nhiều công cụ hỗ trợ như quản lý dự án, wiki, và tích hợp liên tục CI/CD.

Tôi chọn sử dụng GitHub vì nó giúp dễ dàng chia sẻ mã nguồn và phối hợp phát triển sản phẩm một cách hiệu quả, đặc biệt khi làm việc với nhóm hoặc cần backup mã nguồn trên đám mây. GitHub cung cấp các tính năng review mã và quản lý dự án, giúp quy trình phát triển trở nên chuyên nghiệp và linh hoạt hơn. Ngoài ra, GitHub hỗ trợ tích hợp với nhiều dịch vụ và công cụ khác, giúp tự động hóa các quy trình phát triển và triển khai sản phẩm thông qua GitHub Actions.

Trong ứng dụng này, GitHub được sử 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 sửa. Ngoài ra, GitHub Actions được sử dụng để tự động hóa quy trình cập nhật dữ liệu phim hàng ngày, chạy các script crawl dữ liệu từ IMDb và BoxOfficeMojo.

#### 4.4 Công nghệ sử dụng trong việc crawl dữ liệu

Trong quá trình xây dựng ứng dụng gợi ý phim, việc thu thập và cập nhật dữ liệu từ các nguồn uy tín là một phần quan trọng nhằm đảm bảo tính chính xác và cập nhật của thông tin phim ảnh. Để thực hiện việc này, tôi đã sử dụng các công cụ mạnh mẽ như Selenium và Requests, kết hợp với việc tự động hóa quy trình cập nhật dữ liệu bằng GitHub Actions.

##### 4.4.1 Selenium và Requests

Selenium là một công cụ tự động hóa web mạnh mẽ, cho phép điều khiển trình duyệt web và tương tác với các trang web động giống như người dùng thật. Selenium có thể thực hiện các hành động như click, nhập liệu, cuộn trang, và chờ đợi các phần tử động tải. Requests là một thư viện HTTP đơn giản nhưng mạnh mẽ trong Python, giúp gửi các yêu cầu HTTP và truy xuất dữ liệu từ các trang web. Requests hỗ trợ nhiều tính năng như duyệt qua các trang web, gửi dữ liệu dưới dạng biểu mẫu và xử lý cookies.

Tôi chọn sử dụng Selenium và Requests vì chúng cho phép thu thập dữ liệu từ các trang web uy tín như IMDb và BoxOfficeMojo một cách tự động và hiệu quả. Selenium được sử dụng để xử lý các trang web động yêu cầu người dùng phải cuộn trang hoặc nhấn nút để tải thêm dữ liệu, trong khi Requests được sử dụng cho các trang web tĩnh để truy xuất dữ liệu nhanh chóng hơn. Sự kết hợp của hai công cụ này giúp tôi có thể thu thập được dữ liệu phong phú và chính xác về phim ảnh cho ứng dụng.

Trong ứng dụng này, Selenium được sử dụng để crawl dữ liệu từ IMDb, bao gồm 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 chạy tự động hàng ngày thông qua GitHub Actions để đảm bảo dữ liệu luôn được cập nhật mới nhất.

##### 4.4.2 Công nghệ sử dụng để update dữ liệu

Việc duy trì và cập nhật dữ liệu phim ảnh một cách liên tục là điều cần thiết để ứng dụng gợi ý phim luôn cung cấp thông tin mới nhất và chính xác. Để tự động

hóa quy trình này, tôi đã sử dụng GitHub Actions. GitHub Actions là một công cụ CI/CD tích hợp sẵn trong GitHub, cho phép thiết lập các workflow tự động chạy khi có sự kiện cụ thể.

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

Cụ thể, mỗi ngày GitHub Actions sẽ kích hoạt các script crawl dữ liệu tự động để kiểm tra và cập nhật những thay đổi mới nhất trên IMDb. Điều này bao gồm các trường thông tin như doanh thu, trailer và hình ảnh và các thông tin liên quan. Workflow GitHub Actions được tôi cài đặt các bước sau:

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

Đọc, xử lý và làm sạch dữ liệu thu thập được, đảm bảo dữ liệu đúng định dạng và cấu trúc.

Cập nhật vào cơ sở dữ liệu MySQL với các thông tin mới nhất.

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

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

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

### 5.1 Thiết kế kiến trúc

#### 5.1.1 Lựa chọn kiến trúc phần mềm

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

##### Giải thích sơ bộ về MVT:

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

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

##### Mô tả kiến trúc cụ thể cho ứng dụng:

###### • Model (M):

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

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

###### • View (V):

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

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

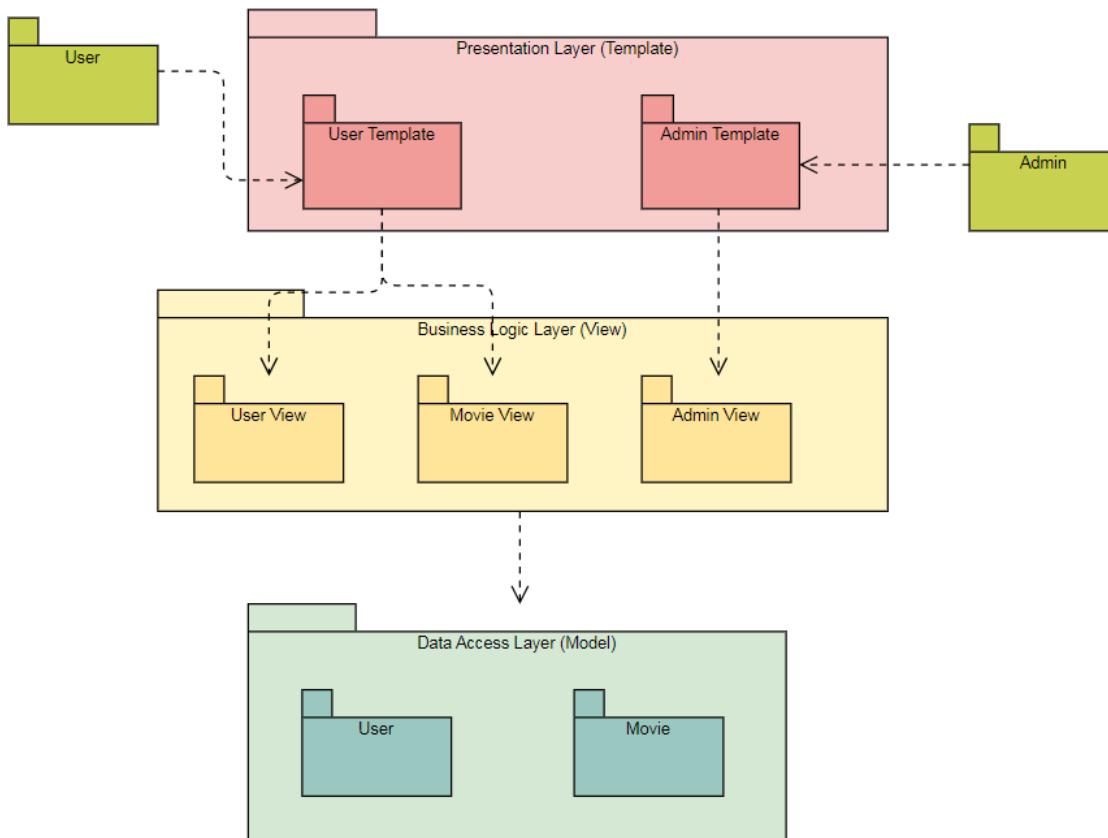
###### • Template (T):

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

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

### 5.1.2 Thiết kế tổng quan

Các gói và sự phụ thuộc giữa các gói:



**Hình 5.1:** Biểu đồ phụ thuộc gói

#### 1. Data Access Layer (Model)

**Mục đích:** Ở tầng này, chứa các định nghĩa về dữ liệu và cấu trúc dữ liệu của ứng dụng.

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

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

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

#### 2. Business Logic Layer (View)

**Mục đích:** Ở tầng này, xử lý logic nghiệp vụ và quản lý các yêu cầu từ người dùng.

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

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

**Phụ thuộc:** Tầng này phụ thuộc vào gói Models. View sẽ truy xuất dữ liệu từ model và xử lý logic nghiệp vụ ở dưới, không phụ thuộc vào gói Templates trực tiếp. Dữ liệu được chuẩn bị trong view sẽ được truyền đến template để hiển thị cho người dùng.

### 3. Presentation Layer (Template)

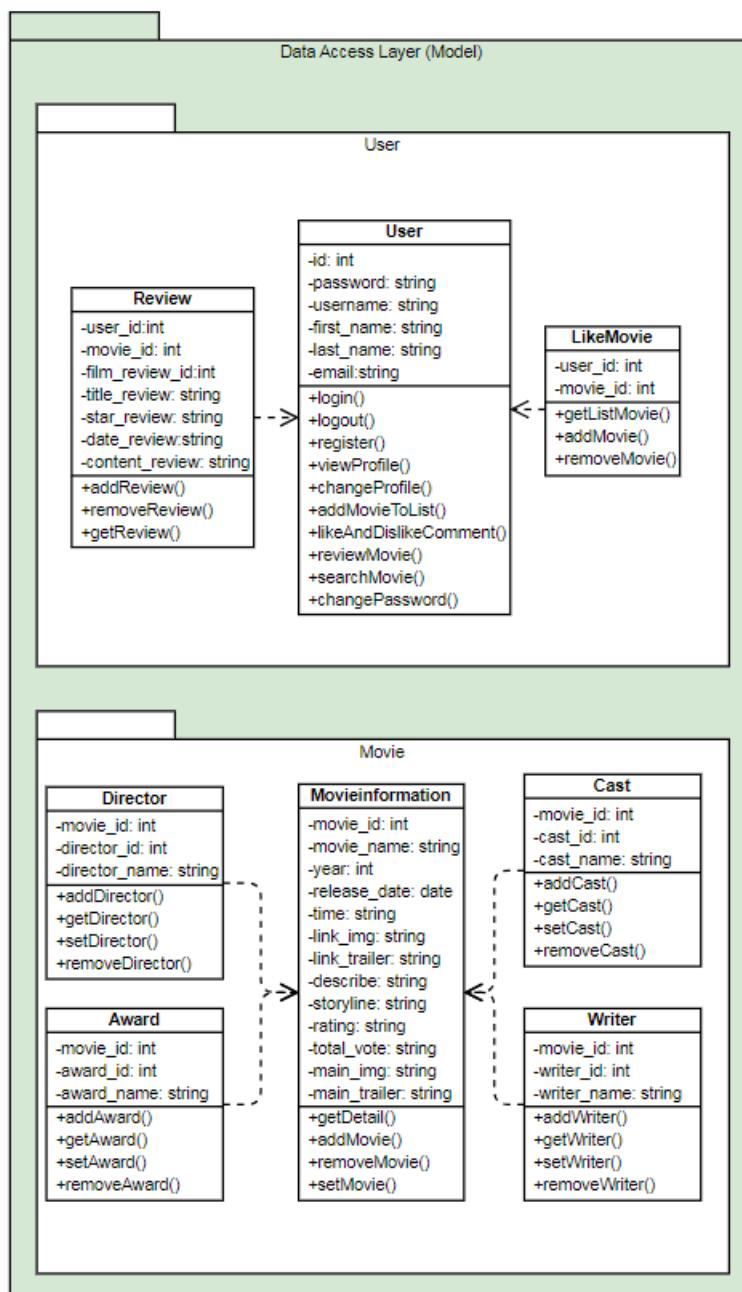
**Mục đích:** Ở tầng này, định nghĩa giao diện người dùng.

**Nhiệm vụ:** Hiển thị dữ liệu được truyền từ view dưới dạng HTML, CSS và JavaScript. Template sẽ nhận dữ liệu từ view và hiển thị nó.

**Các thành phần:** Các tệp HTML template cho các trang web khác nhau của ứng dụng, ví dụ như trang đăng nhập, đăng ký, trang chủ, trang chi tiết phim.

**Phụ thuộc:** Tầng này phụ thuộc vào gói Views. Templates nhận dữ liệu từ view và hiển thị nó, không phụ thuộc vào gói Models trực tiếp. Mọi dữ liệu hiển thị trên giao diện đều được truyền từ view.

### 5.1.3 Thiết kế chi tiết gói



**Hình 5.2:** Thiết kế chi tiết gói ở tầng Model

#### Gói Model:

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

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

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

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

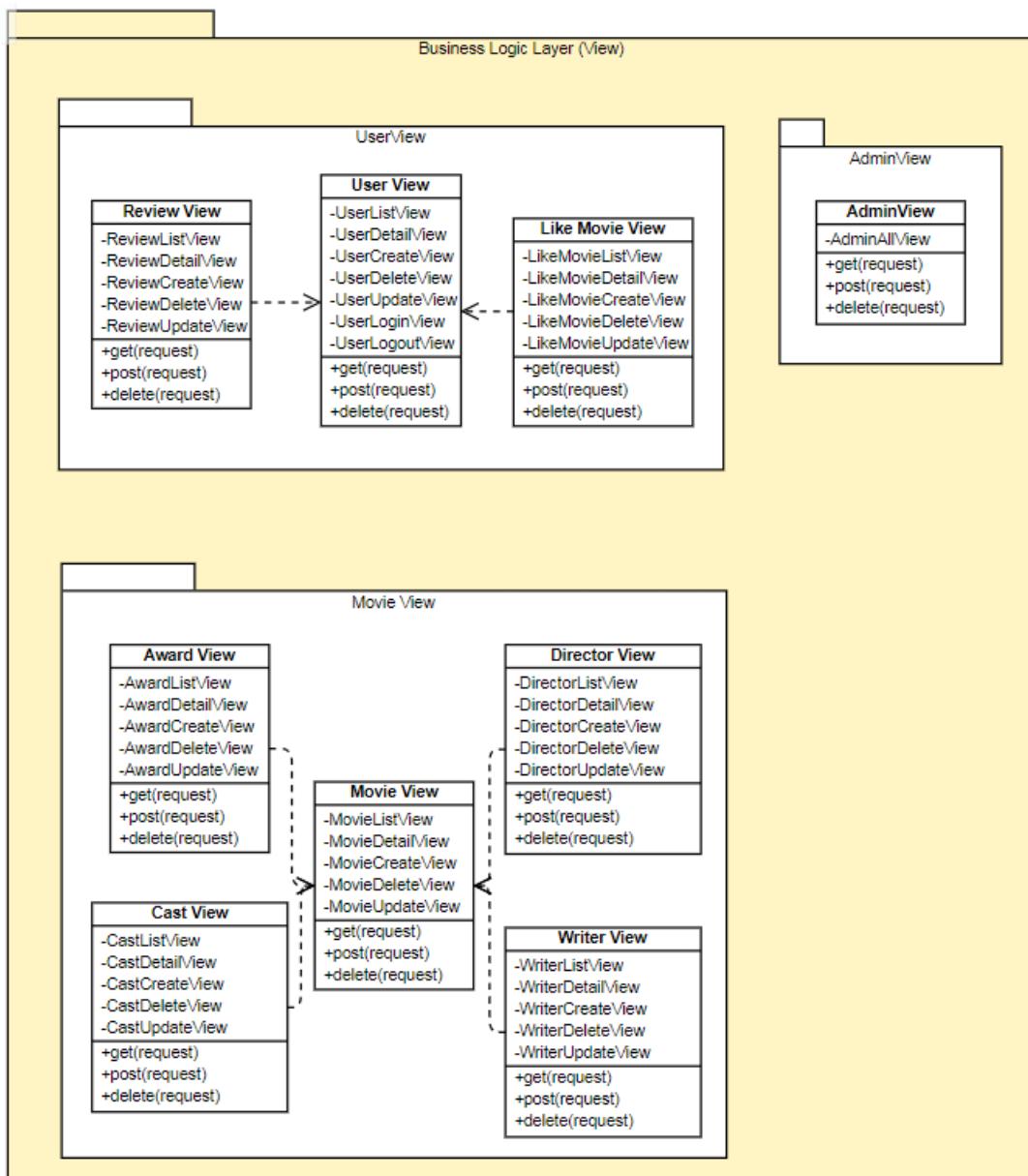
User.

- **Gói Movie:**

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

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

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



Hình 5.3: Thiết kế chi tiết gói ở tầng View

### Gói View:

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

- **Gói UserView:** Gồm 3 class là ReviewView, UserView và LikeMovieView.

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

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

- **Gói MovieView:**

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

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

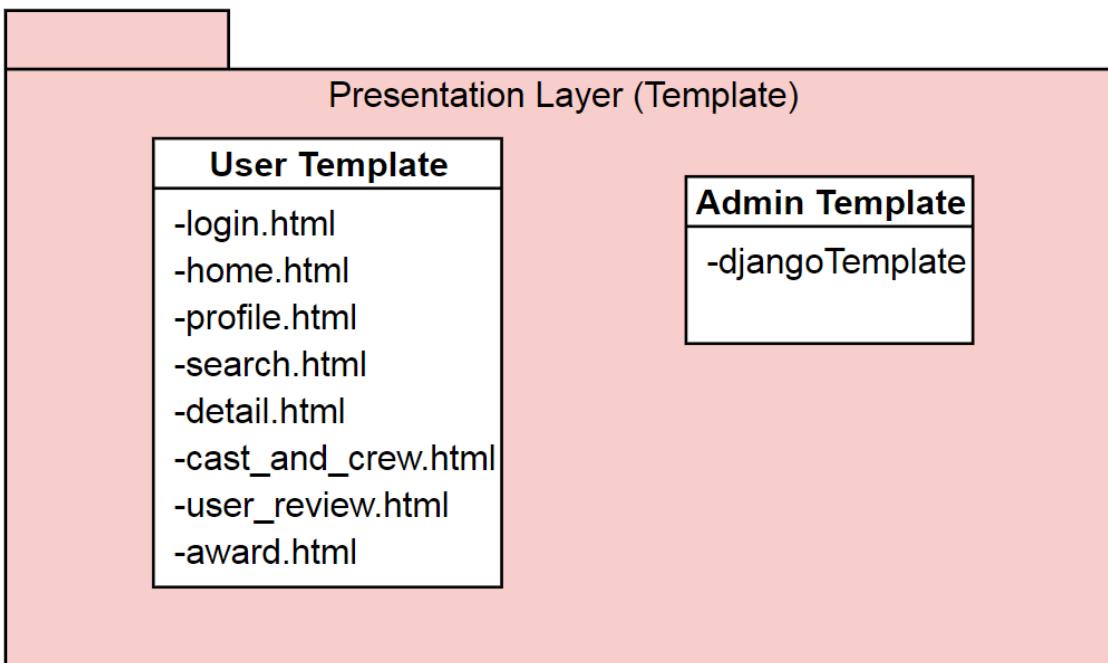
Phụ thuộc: 4 class còn lại có quan hệ phụ thuộc với Movieinfomation, lấy Movieinfomation làm gốc, các thông tin còn lại đều là thông tin bổ sung.

- **Gói AdminView:**

Gồm 1 class là AdminView.

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

Phụ thuộc: Không phụ thuộc vào class nào.

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

### Gói Template:

Gồm 2 class là UserTemplate và AdminTemplate.

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

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

## 5.2 Thiết kế chi tiết

### 5.2.1 Thiết kế giao diện

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

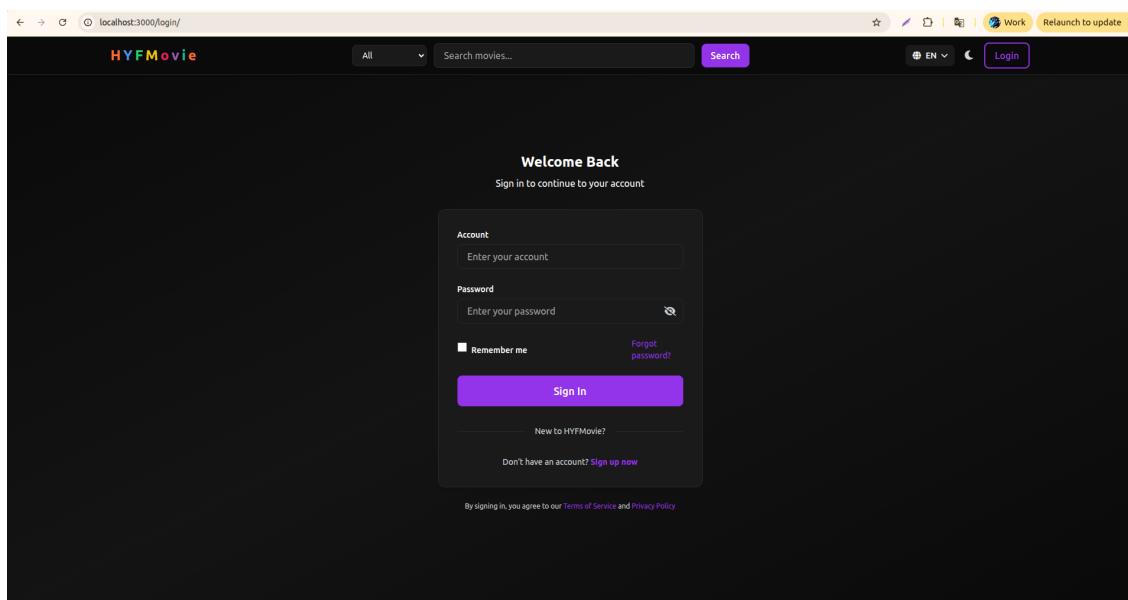
Trong quá trình thiết kế, tôi đã áp dụng các quy chuẩn cụ thể để tạo ra giao diện

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

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

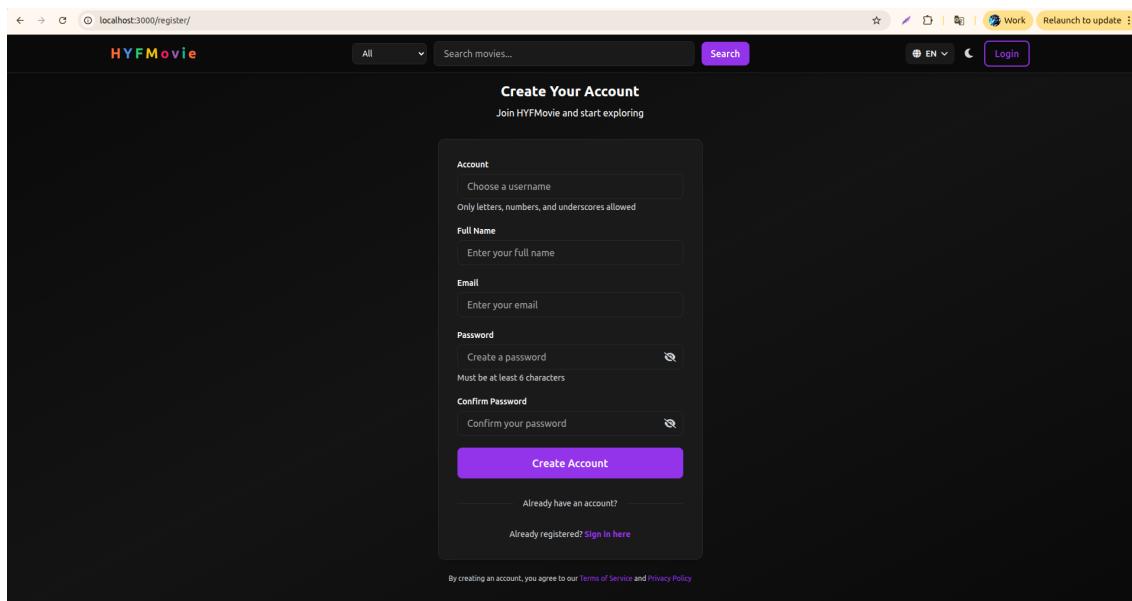
Dưới đây là một số hình ảnh minh họa cho thiết kế giao diện của các chức năng quan trọng trong ứng dụng của tôi:

### Màn hình đăng nhập



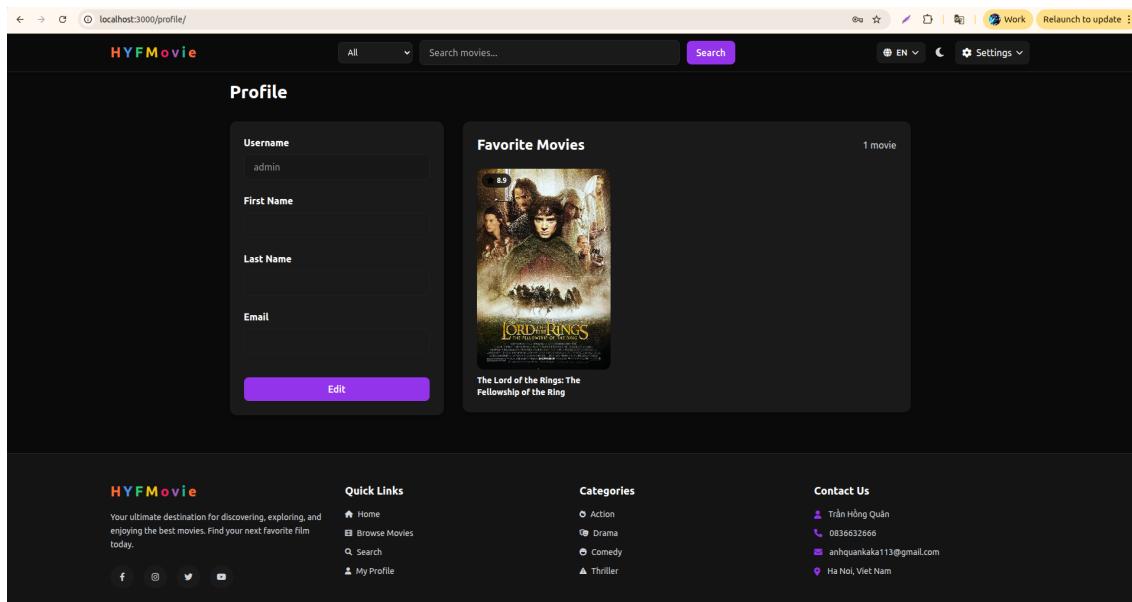
**Hình 5.5:** Màn hình *đăng nhập*

### Màn hình đăng ký



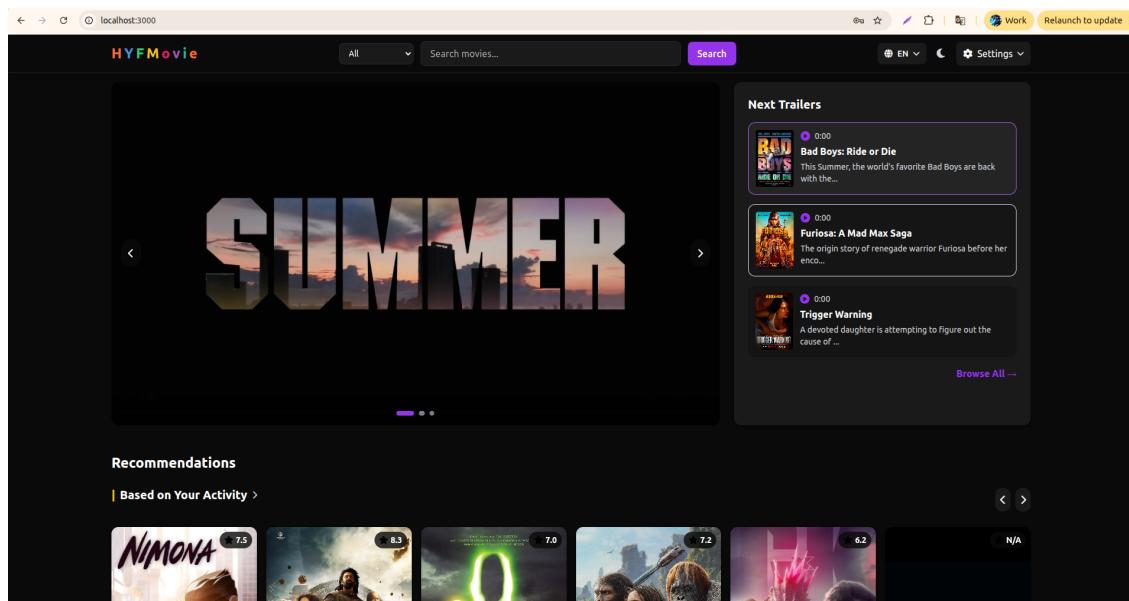
**Hình 5.6:** Màn hình *đăng ký*

## Màn hình profile



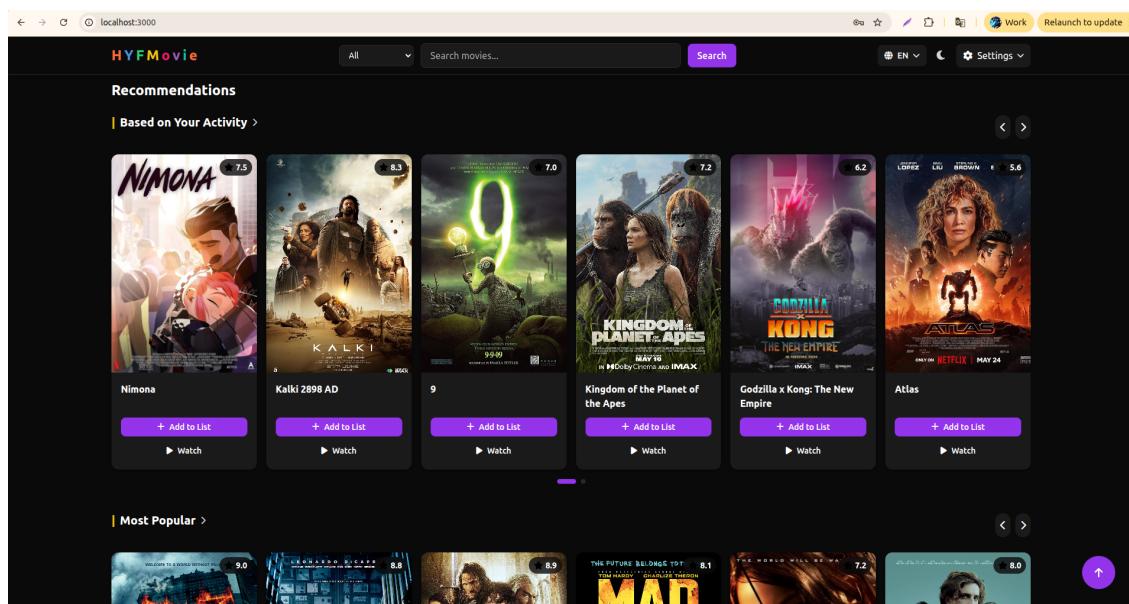
**Hình 5.7:** Màn hình *profile*

## Màn hình chính



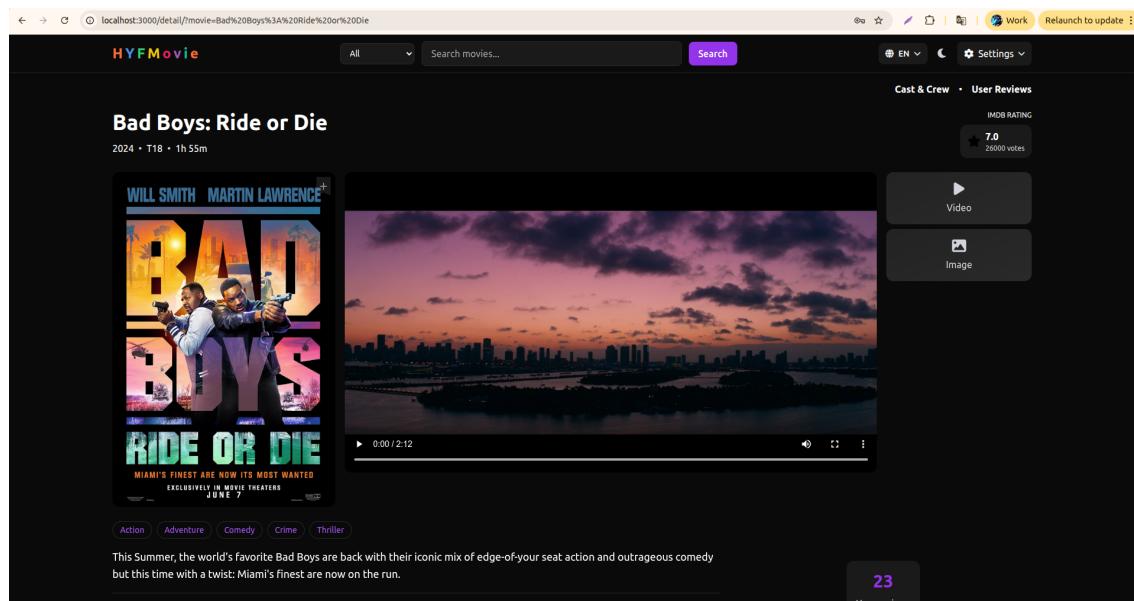
**Hình 5.8:** Màn hình *chính*

### Màn hình gợi ý phim ở trang chủ

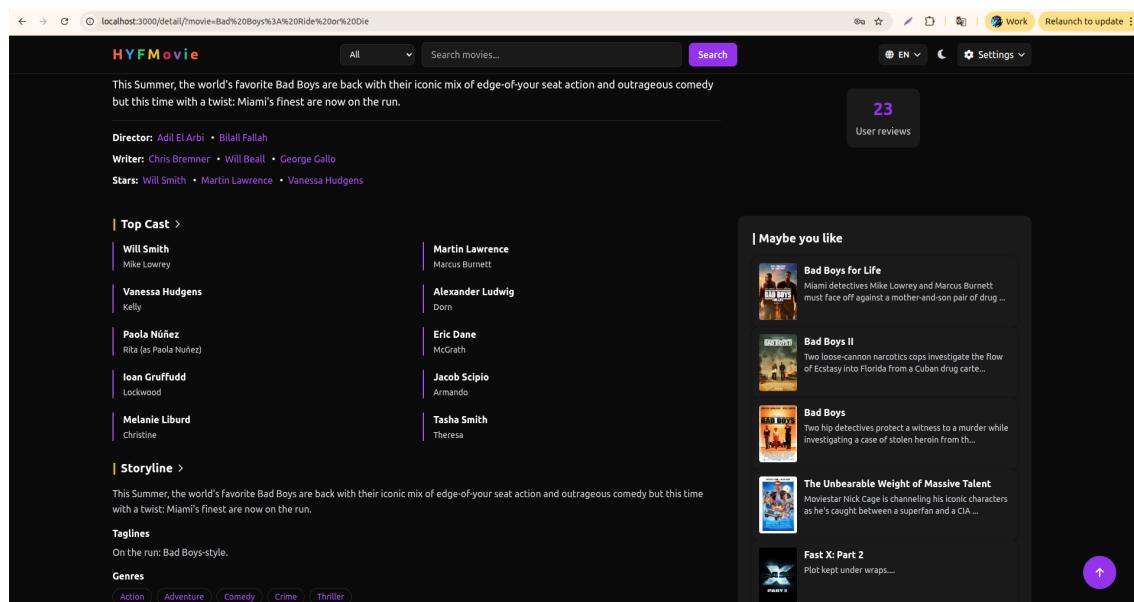


**Hình 5.9:** Màn hình *gợi ý phim* ở trang chủ

### Màn hình thông tin chi tiết

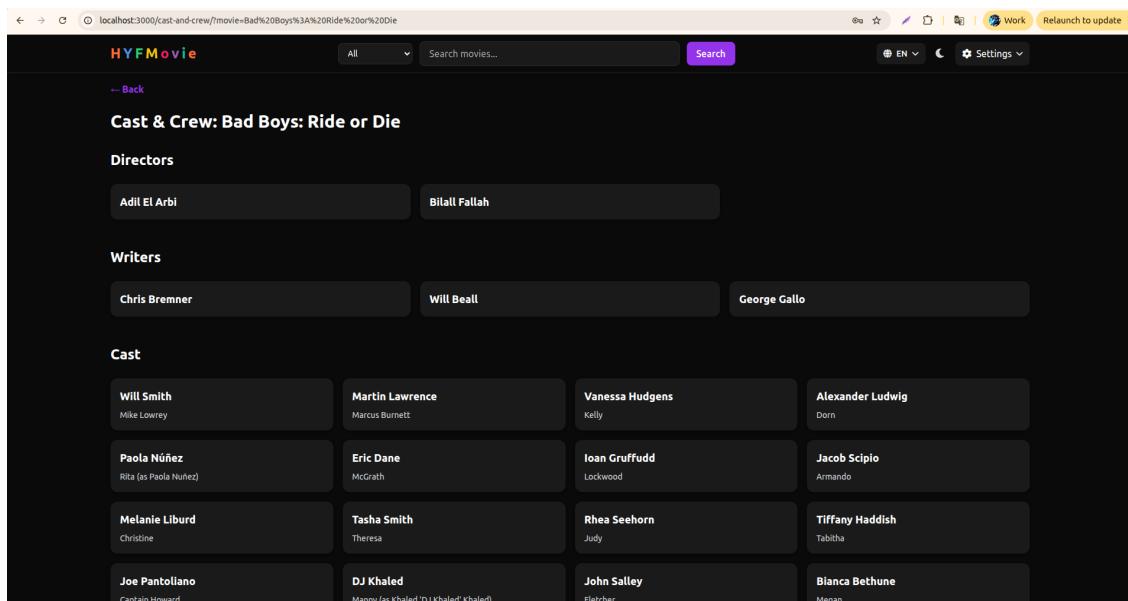


**Hình 5.10:** Màn hình thông tin chi tiết 1



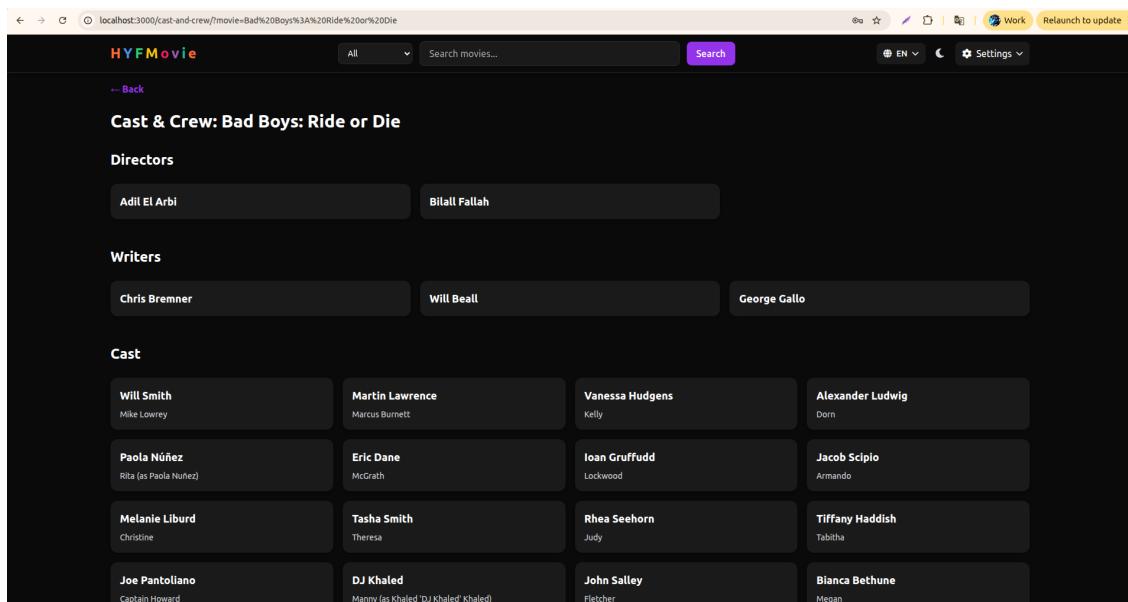
**Hình 5.11:** Màn hình thông tin chi tiết 2

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



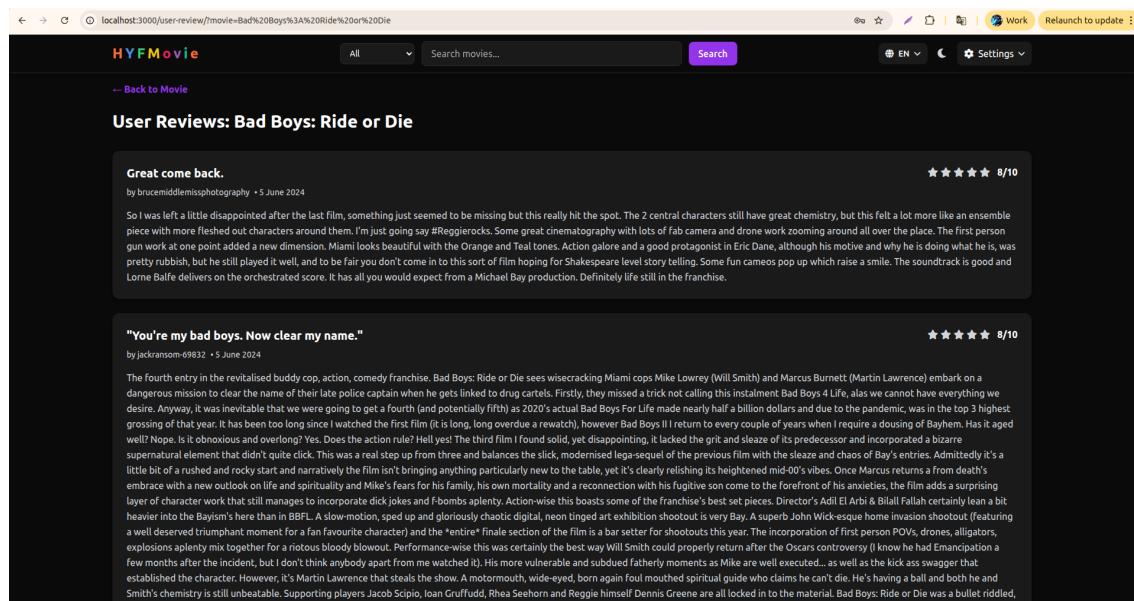
**Hình 5.12:** Màn hình tổng hợp đạo diễn, diễn viên, ...

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



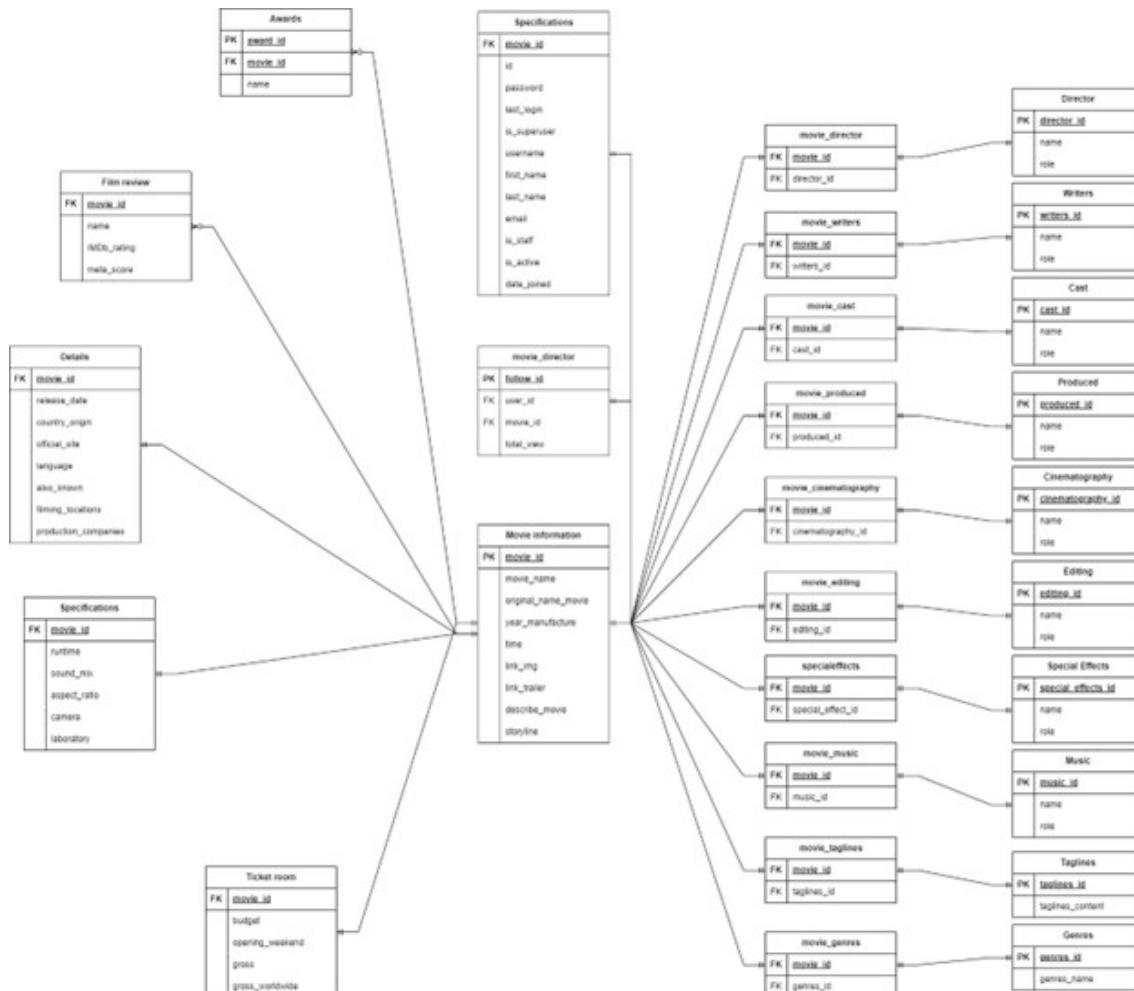
**Hình 5.13:** Màn hình tổng hợp đạo diễn, diễn viên, ...

Màn hình review phim

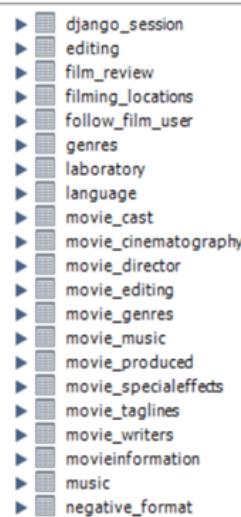
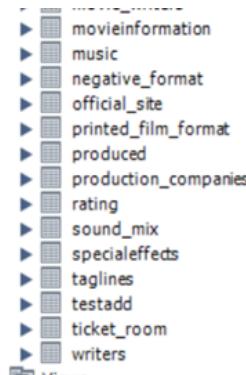


Hình 5.14: Màn hình review phim

## 5.2.2 Thiết kế cơ sở dữ liệu



Hình 5.15: Sơ đồ ERD

**Hình 5.16:** Hình ảnh các table trong database 1**Hình 5.17:** Hình ảnh các table trong database 2**Hình 5.18:** Hình ảnh các table trong database 3

### 5.2.3 Thiết kế hệ thống crawl dữ liệu

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

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

**Bảng 5.1:** Tổng hợp các thông tin cần lấy

**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

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

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.

Mục đích	Công cụ	Tài liệu tham khảo
IDE lập trình	Visual Studio Code (VSCode)	[7]
API Framework	Django Rest Framework	[8]
Frontend	HTML	[9]
Style	Tailwind CSS	[10]
Frontend Framework	React	[11]
Backend Framework	Django	[12]
Database	MySQL	[13]
Công cụ crawl dữ liệu	Selenium	[14]
Thư viện HTTP	Requests	[15]
Công cụ CI/CD	GitHub Actions	[16]

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

### 5.3.2 Kết quả đạt được

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

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

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

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

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

#### Thông kê các thông tin của ứng dụng:

Số dòng code crawl dữ liệu: 2000 dòng

Số dòng code phía backend: 4000 dòng

Số dòng code phía frontend: 8000 dòng

Số lớp: 180 lớp

Số gói: 8 gói

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

Ứng dụng đã hoàn thành với tổ chức cấu trúc rõ ràng, mã nguồn được tổ chức

tốt trong các gói và lớp. Thông kê chi tiết về số dòng code, số lớp và dung lượng mã nguồn giúp đánh giá và theo dõi tiến trình phát triển của dự án. Ứng dụng đáp ứng các yêu cầu đặt ra và sẵn sàng để triển khai và sử dụng trong thực tế.

### Crawl dữ liệu từ trang web:

Nguồn dữ liệu: IMDb, BoxOfficeMojo

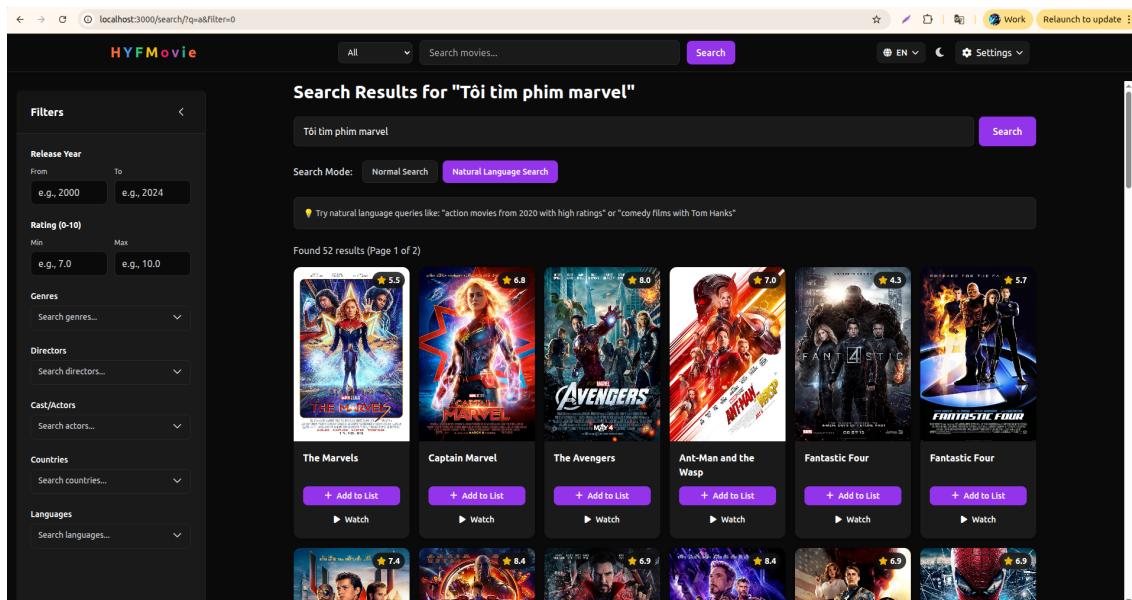
Số bộ phim đã crawl: 5000 bộ phim

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.

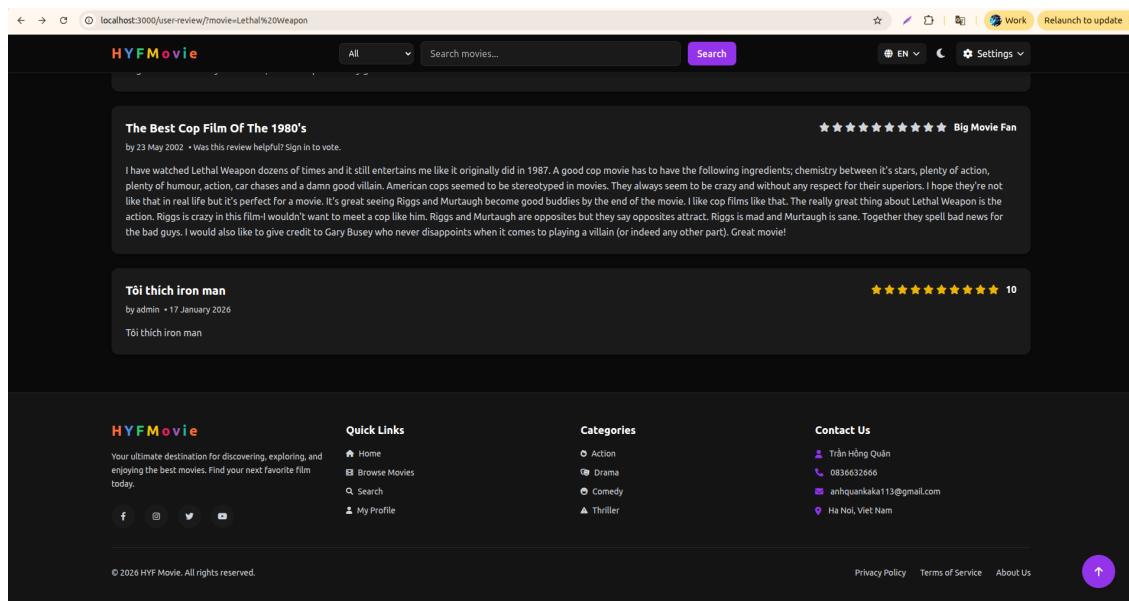


**Hình 5.19:** Chức năng tìm kiếm phim theo ngôn ngữ người dùng

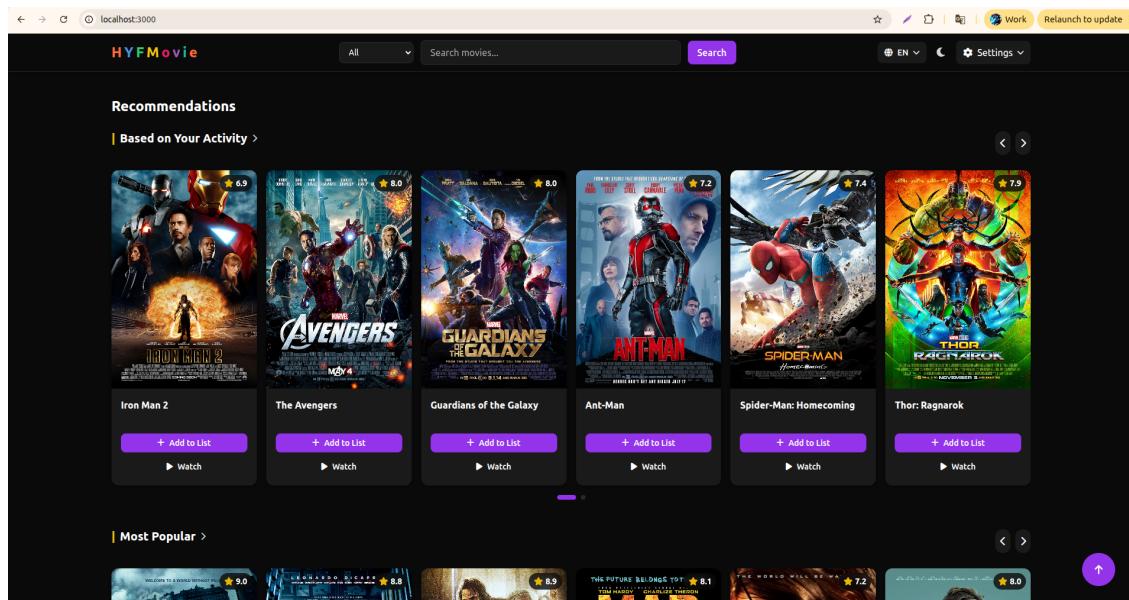
Dựa vào hình ảnh kết quả trên, có thể thấy hệ thống đã đưa ra kết quả phù hợp với yêu cầu của người dùng. Ở trên là các phim marvel hệ thống đã đưa ra kết quả.

### 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.

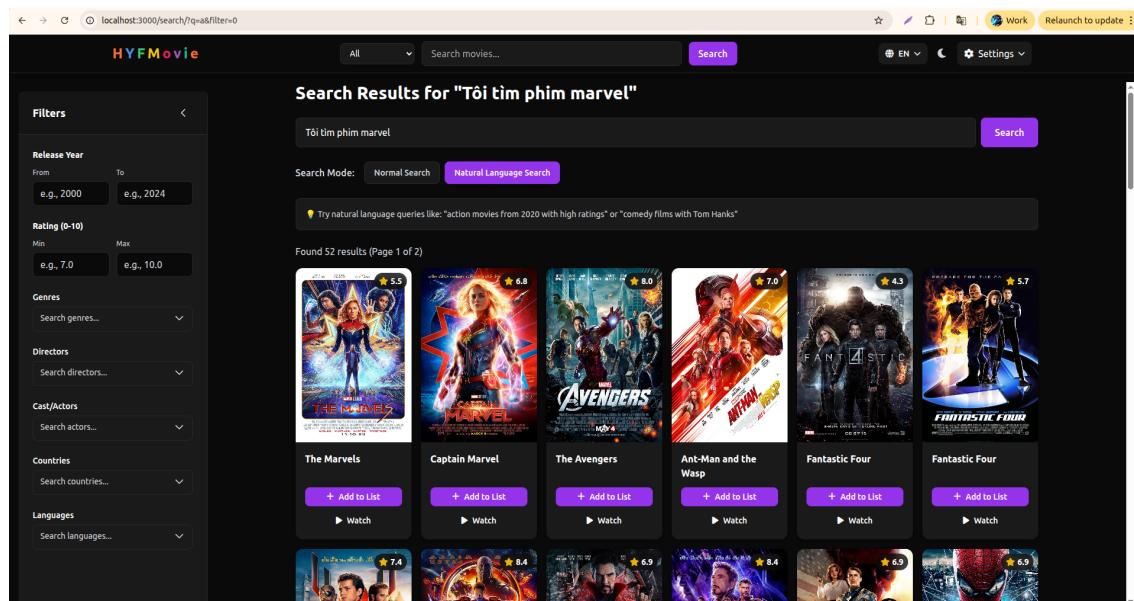


**Hình 5.20:** Review phim

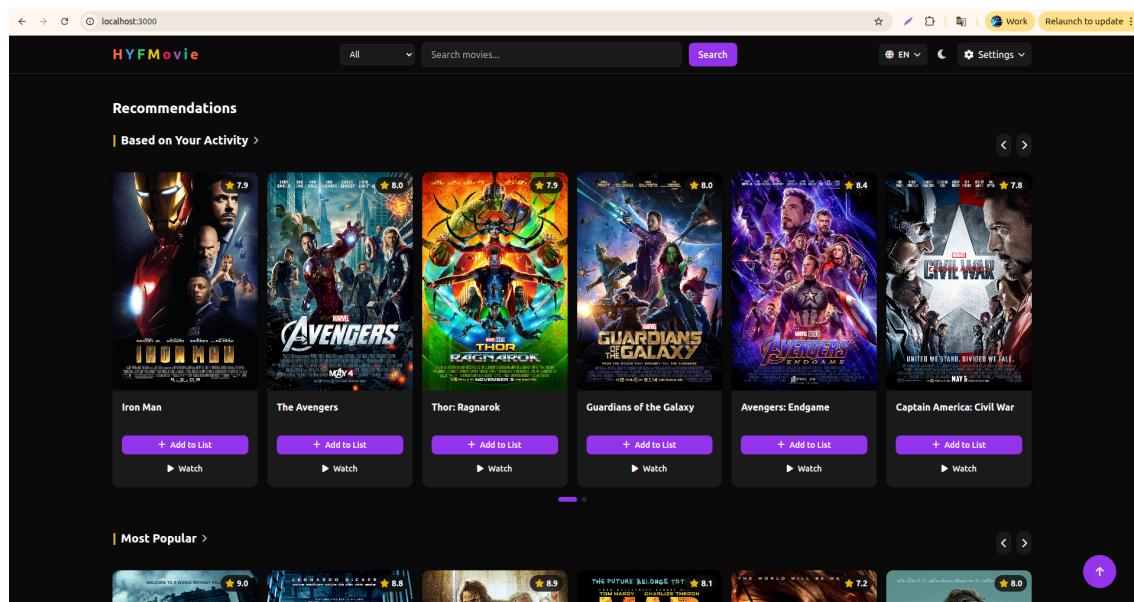


**Hình 5.21:** Gợi ý phim theo lịch sử review

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 5.22:** Tìm kiếm phim

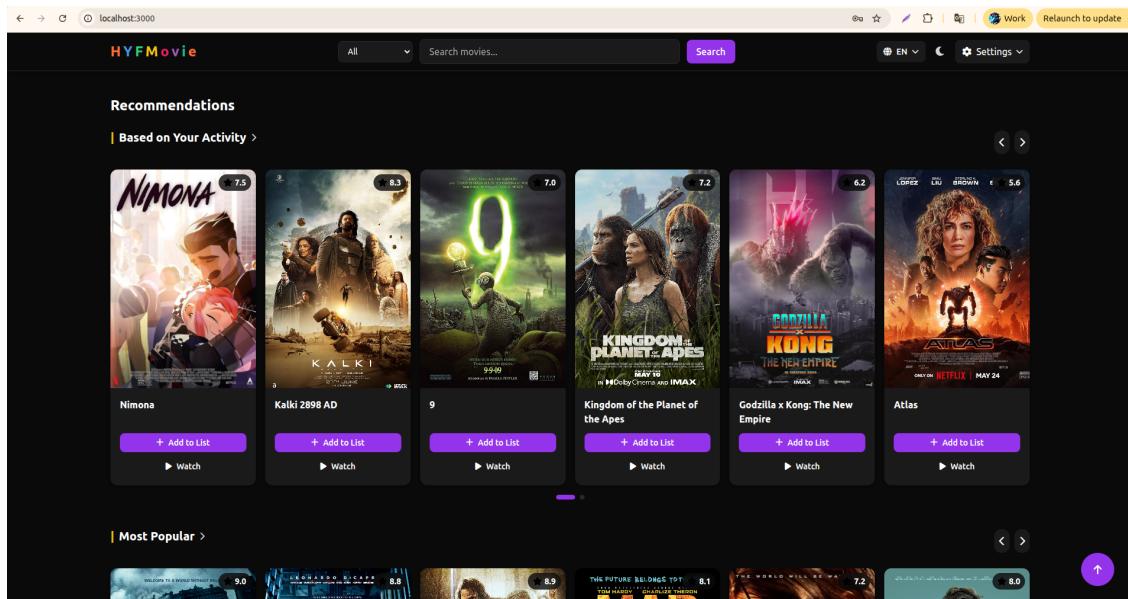


**Hình 5.23:** Gợi ý phim theo lịch sử tìm kiếm

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

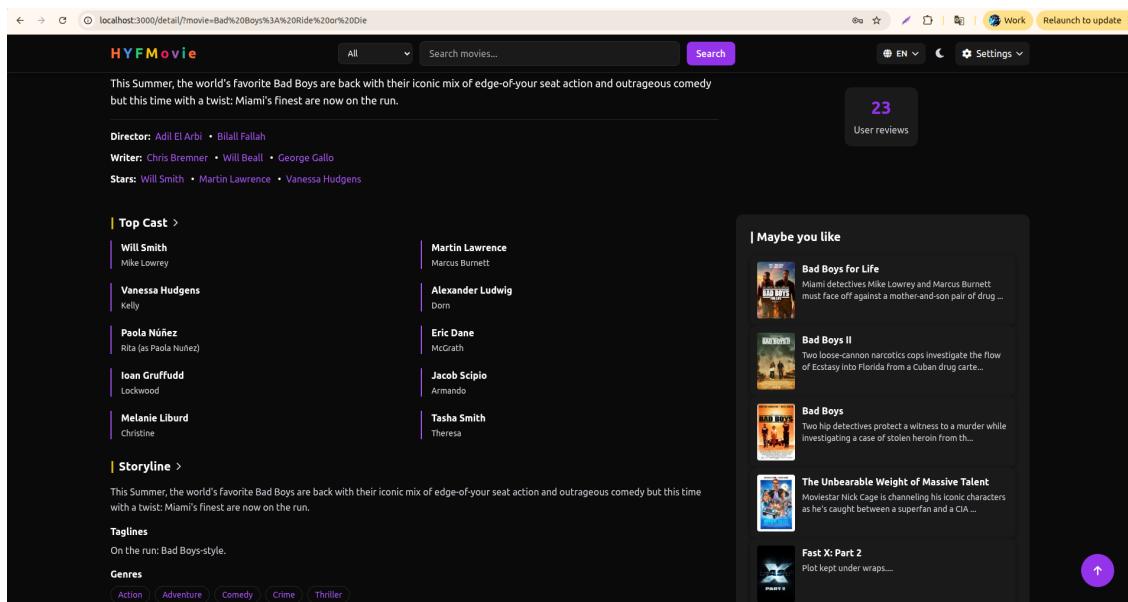
#### Chức năng gợi ý phim dựa vào lịch sử thao tác của người dùng



**Hình 5.24:** Chức năng gợi ý phim dựa vào lịch sử thao tác của người dùng

Dựa vào lịch sử thêm vào danh sách yêu thích của người dùng, ứng dụng có thể tự học và đưa ra được kết quả mới nhất phù hợp với sở thích của người dùng.

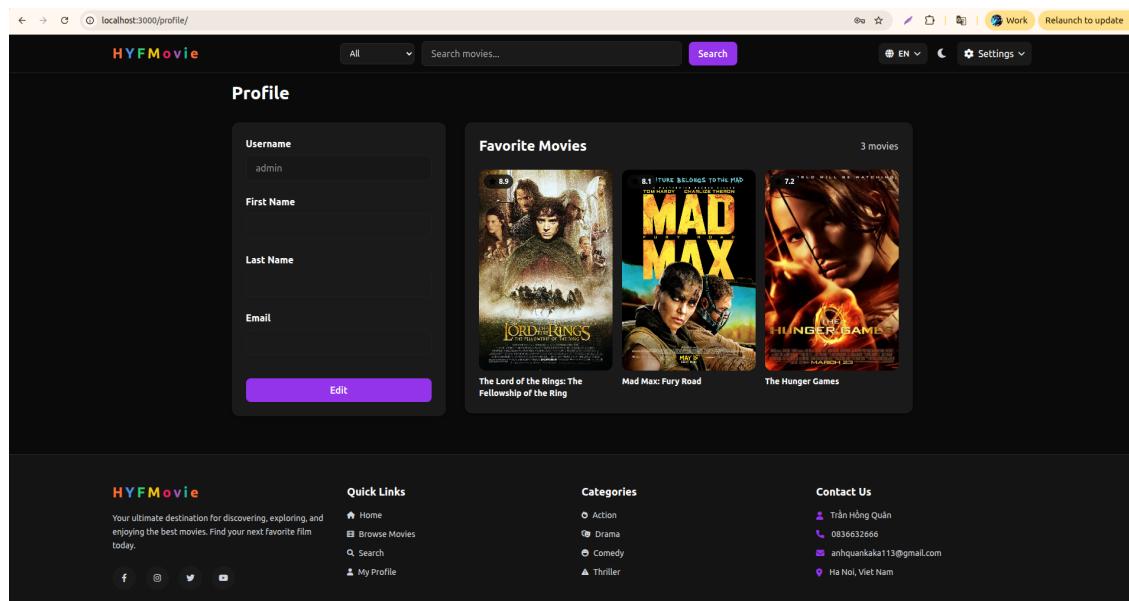
### Chức năng gợi ý phim dựa vào nội dung phim đang xem



**Hình 5.25:** Chức năng gợi ý phim dựa vào nội dung phim đang xem

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

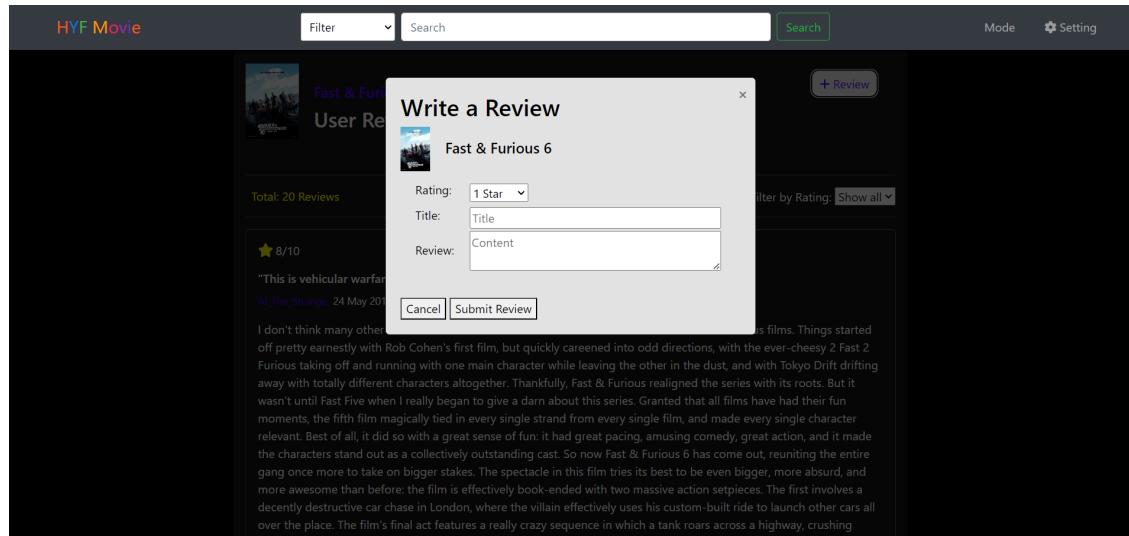
### Chức năng thêm phim vào danh sách yêu thích



**Hình 5.26:** Chức năng thêm phim vào danh sách yêu thích

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

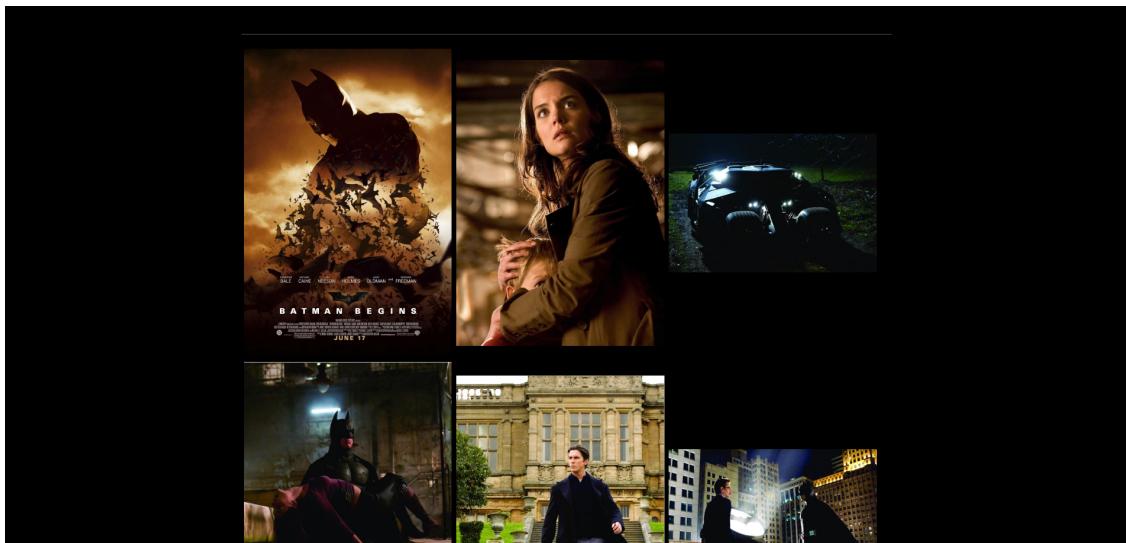
### Chức năng review phim



**Hình 5.27:** Chức năng review phim

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

### Chức năng xem toàn bộ ảnh của phim



**Hình 5.28:** Chức năng xem toàn bộ ảnh của phim

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

#### 5.4 Kiểm thử

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

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 5.3:** Kiểm thử chức năng

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

**Bảng 5.4:** Kiểm thử chức năng

**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 những bước kiểm thử trên (Table 5.3, Table 5.4, Table 5.5, Table 5.6), có thể thấy ứng dụng đã được kiểm thử thành công và đã sẵn sàng đi vào hoạt động khi toàn bộ test case đều được thông qua và hoạt động tốt.

## 5.5 Triển khai

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

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

Hệ điều hành: Windows, Linux, MacOS

CPU: Intel i3 hoặc tương đương

RAM: 4GB

Ổ cứng: 128GB SSD

Kết nối mạng: Internet tốc độ trung bình

### Triển khai

Clone repository từ GitHub về máy tính cá nhân.

Cài đặt docker vào máy.

Cd vào thư mục project và chạy lệnh docker-compose up -d để chạy ứng dụng. Lúc này thì toàn bộ ứng dụng đã được chạy trên máy tính cá nhân. Vào cổng localhost:3000 để test ứng dụng.

### Kết quả triển khai thử nghiệm

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

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

**Thời gian phản hồi:** Thời gian phản hồi trung bình dưới 2 giây cho các yêu cầu cơ bản, còn gợi ý phim thì trung bình dưới 5s, tìm kiếm phim thì dưới 10s.

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

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

## CHƯƠNG 6. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT

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

### 6.1 Thiếu khá nhiều dữ liệu phim ảnh

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

### 6.2 Thiết kế database khá khó khăn với quá nhiều trường

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

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

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

Ở cách giải quyết là tải toàn bộ ảnh và trailer xuống thì tôi đã loại bỏ ngay sau khi quyết định số bộ phim để demo ứng dụng. Với 5000 bộ phim thì sẽ có tổng cộng hơn 250000 ảnh và 5000 video demo, nếu tải về thì rất nặng máy và không tối ưu. Vì thế, tôi đã nghiên cứu và tìm ra cách để ứng dụng tự update data, đó chính là dùng github action.

#### Giới thiệu Github Action:

Github Actions là một dịch vụ được cung cấp bởi GitHub để tự động hóa các quy trình phát triển phần mềm. Bằng cách sử dụng các action, tôi có thể thiết lập các luồng làm việc tự động, từ việc kiểm tra mã nguồn, triển khai ứng dụng, tự

đóng hóa các tác vụ như cập nhật dữ liệu trong ứng dụng.

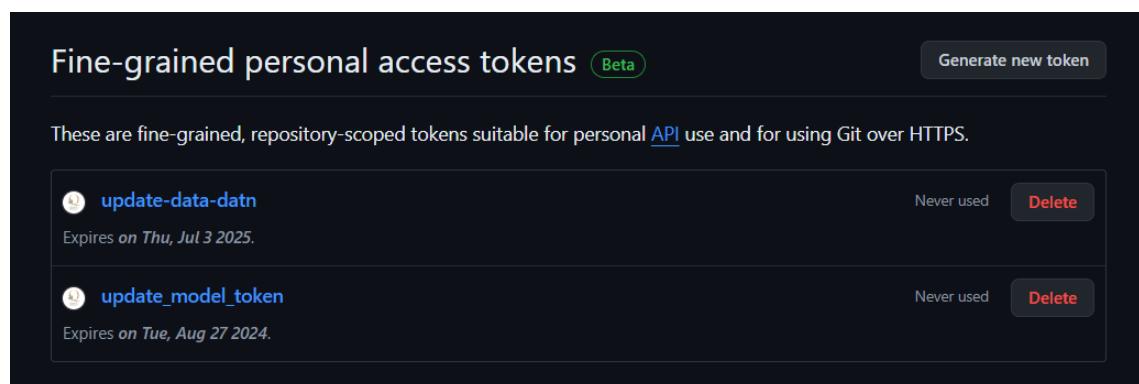
### Triển khai:

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

Để ứng dụng có thể tự động update được, sẽ có những bước như sau:

#### 1. Tạo PAT (Personal Access Token) Github

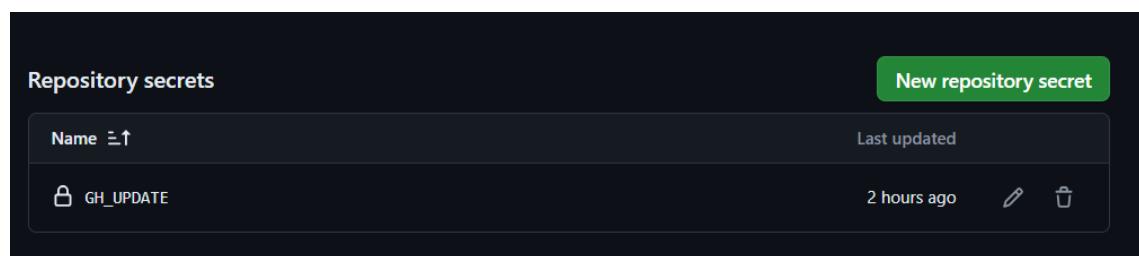
Personal Access Token (PAT) trên GitHub được sử dụng để cung cấp quyền truy cập cho các ứng dụng hoặc dịch vụ bên thứ ba vào tài khoản GitHub, mà không cần phải sử dụng mật khẩu. Ở trong ứng dụng này, dịch vụ bên thứ 3 chính là bot tự động.



**Hình 6.1:** Tạo PAT Github

#### 2. Tạo Secrets Repository

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



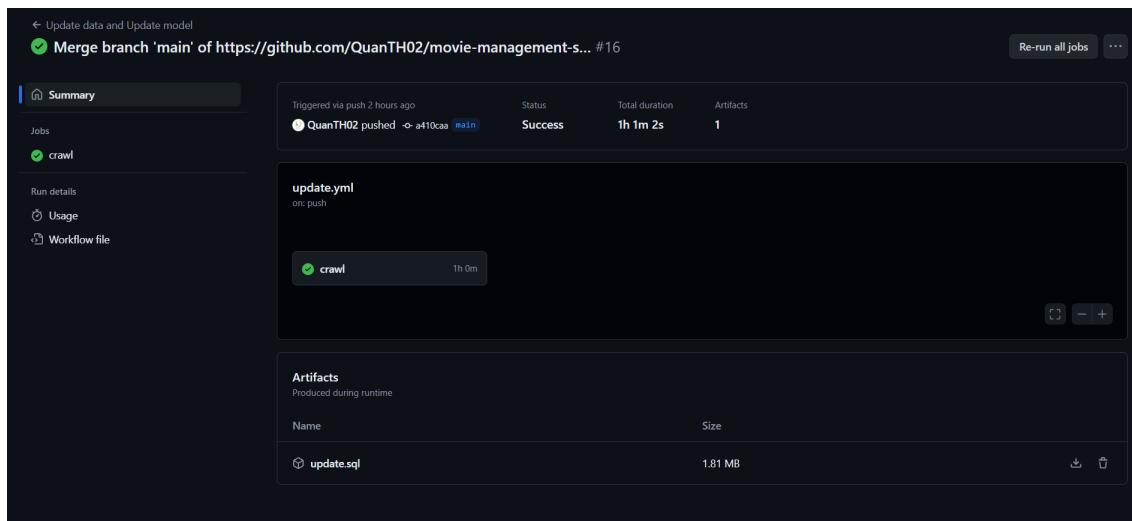
**Hình 6.2:** Tạo Secrets Repository

### 3. Viết file update.yml

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

#### Kết quả:

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



**Hình 6.3:** Hệ thống tự động crawl dữ liệu

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

The screenshot shows a MySQL Workbench interface. The top window is titled "link\_img" and contains the SQL query: "SELECT count(\*) FROM moviedata.link\_img;". Below this is a "Result Grid" showing a single row with the value "49030". The interface includes standard database management tools like file operations, search, and refresh.

count(*)
49030

**Hình 6.4:** Cập nhật dữ liệu mới thành công

#### 6.4 Lựa chọn thuật toán gợi ý phù hợp

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

##### Giới thiệu thuật toán Content-based:

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

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

##### Phân tích, giải pháp:

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

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

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

Trong đó:

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

### Kiểm tra tính chính xác:

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

```

Movie id: 136 Untitled-1
1 Movie id: 136
2 Movie name: Guardians of the Galaxy
3 Describe: A group of intergalactic criminals must pull together to stop a fanatical warrior with plans to purge the universe.
4 Genres:
5 Director: James Gunn
6 Cast: Chris Pratt Zoe Saldana Dave Bautista Vin Diesel
Bradley Cooper Lee Pace Michael Rooker Karen Gillan Djimon
Hounsou John C. Reilly Glenn Close Benicio Del Toro Laura
Haddock Sean Gunn Peter Serafinowicz
7 Movie id result:
8 21 0.3526438817427592
9 109 0.21987801134666684
10 48 0.11113121522658766
11 26 0.10923877357827322
12 120 0.08440632301936431
13 19 0.05394264748270165
14 86 0.04964744394508321
15 78 0.047996604986685044
16 32 0.047196291445337255
17 74 0.04686329421081032
18
19 genres
20 45 0.0
21 51 0.0
22 50 0.0
23 49 0.0
24 48 0.0
25 47 0.0
26 46 0.0
27 44 0.0
28 36 0.0
29 43 0.0
30
31 des
32 86 0.13878849563162084
33 63 0.10159744616742258
34 58 0.08755433976577434
35 48 0.0831740809398962
36 80 0.06715573133762048
37 120 0.06481968158583065
38 87 0.06434887142048598
39 43 0.06091447365093486
40 10 0.05759624460829198
41 21 0.05458436118761168
42

cast Untitled-2
1 cast
2 109 0.3647058800017843
3 21 0.338707928444981
4 48 0.1644614398256814
5 120 0.10773013419658836
6 26 0.1055610662972952
7 70 0.10167238581379695
8 32 0.07255420776372298
9 63 0.06604964845434892
10 46 0.06358163683880483
11 48 0.062483862371309436
12
13 director
14 21 1.0
15 26 1.0
16 89 0.4032060238565666
17 46 0.4032060238565666
18 59 0.38328757973316574
19 22 0.38328757973316574
20 50 0.0
21 49 0.0
22 48 0.0
23 47 0.0
24
25 Movie name: Guardians of the Galaxy Vol. 3
26 Describe: Still reeling from the loss of Gamora, Peter Quill rallies his team to defend the universe and one of their own - a mission that could mean the end of the Guardians if not successful.
27 Genres:
28 Director: James Gunn
29 Cast: Chukwudi Iwuji Bradley Cooper Pom Klementieff
Dave Bautista Karen Gillan Vin Diesel Austin Freeman
Stephen Blackehart Terence Rosemore Maria Bakalova Sean
Gunn Sarah Alami Jasmine Munoz Chris Pratt Giovannie
Cruz
30
31 Chris Pratt
32 Bradley Cooper
33 Dave Bautista
34 Karen Gillan
35 Vin Diesel
36 Sean Gunn
37
38

```

Hình 6.5: Kiểm tra kết quả 1

Ở hình ảnh 6.5 có thể thấy bộ phim được đề xuất gần nhất với bộ phim Guardians

of the Galaxy chính là bộ phim Guardians of the Galaxy Vol. 3. Hai bộ phim này có movie name, describe, cast và director tương tự nhau.

Hình 6.6: Kiểm tra kết quả 2

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

```

Movie id: 15 Untitled-1 ...
1 Movie id: 15
2 Movie name: Meg 2: The Trench
3 Describe: A research team encounters multiple threats while exploring the depths of the ocean, including a malevolent mining operation.
4 Genres: Action Adventure Horror Sci-Fi Thriller
5 Director: Ben Wheatley
6 Cast: Jason Statham Jing Wu Shuya Sophia Cai Cliff Curtis Page Kennedy Sergio Peris-Mencheta Skyler Samuels Melissanthi Mahut Whoopie Van Raam Kiran Sonia Samar Felix Mayr Sienna Guillory Tao Guo Robin Hill Lele Dai
7 Movie id result:
8 88 0.5504308169597662
9 38 0.35319267661183157
10 2 0.333988033722691944
11 105 0.3223983287172102
12 9 0.30436133874364807
13 75 0.3006358473993846
14 61 0.2873170128093882
15 140 0.28669542992067987
16 54 0.2861652412881092
17 34 0.28115353309874935
18
19
20 genres
21 80 0.932862414224352
22 38 0.8304865093064482
23 105 0.7549184158987879
24 2 0.7320245466125233
25 22 0.7256822305611071
26 61 0.7256822305611071
27 75 0.7256822305611071
28 140 0.7256822305611071
29 73 0.6431597835862669
30 4 0.6239914216557716
31
32 des
33 90 0.17469314530554264
34 11 0.0990214542750533
35 83 0.07411479933063567
36 80 0.07368250880623025
37 84 0.07172309901182033
38 42 0.06981388044848848
39 102 0.06898968252852719
40 103 0.06898968252852719
41 4 0.05255358144650688
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
229
230
231
232
233
234
235
236
237
238
239
239
240
241
242
243
244
245
246
247
248
249
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
287
288
289
289
290
291
292
293
294
295
296
297
297
298
299
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
312
313
314
315
316
317
318
319
319
320
321
322
323
324
325
326
327
328
329
329
330
331
332
333
334
335
336
337
338
339
339
340
341
342
343
344
345
346
347
348
349
349
350
351
352
353
354
355
356
357
358
359
359
360
361
362
363
364
365
366
367
368
369
369
370
371
372
373
374
375
376
377
378
379
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
399
400
401
402
403
404
405
406
407
408
409
409
410
411
412
413
414
415
416
417
418
419
419
420
421
422
423
424
425
426
427
428
429
429
430
431
432
433
434
435
436
437
438
439
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
518
519
519
520
521
522
523
524
525
526
527
528
529
529
530
531
532
533
534
535
536
537
538
539
539
540
541
542
543
544
545
546
547
548
549
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
698
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
896
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
988
988
989
989
990
991
992
993
994
995
996
997
998
999
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1089
1090
1091
1092
1093
1094
1095
1096
1097
1097
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1189
1190
1191
1192
1193
1194
1195
1196
1197
1197
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1288
1289
1289
1290
1291
1292
1293
1294
1295
1295
1296
1297
1297
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1389
1390
1391
1392
1393
1394
1395
1396
1397
1397
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1488
1489
1489
1490
1491
1492
1493
1494
1495
1495
1496
1497
1497
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1588
1589
1589
1590
1591
1592
1593
1594
1595
1595
1596
1597
1597
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1688
1689
1689
1690
1691
1692
1693
1694
1695
1695
1696
1697
1697
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1788
1789
1789
1790
1791
1792
1793
1794
1795
1795
1796
1797
1797
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1888
1889
1889
1890
1891
1892
1893
1894
1895
1895
1896
1897
1897
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1988
1989
1989
1990
1991
1992
1993
1994
1995
1995
1996
1997
1997
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2069
2070
2071
2072
2073
20
```

## CHƯƠNG 7. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

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

### Kết quả so sánh

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

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

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

### Phân tích quá trình thực hiện

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

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

### Những điểm chưa làm được

- Chưa triển khai, deploy lên server thực tế. Hiện tại ứng dụng chỉ chạy trên môi trường local, chưa được kiểm thử với lượng người dùng lớn trong điều kiện tải cao.
- Tổng quan ứng dụng đang còn đơn giản. Nhìn qua, ứng dụng có giao diện, luồng thao tác còn đang đơn giản, hầu như chỉ view.

### **Đóng góp nổi bật**

- Thiết kế và triển khai ứng dụng được thiết kế ngay từ đầu: Tôi đã hoàn thành các chức năng cơ bản và đảm bảo ứng dụng hoạt động ổn định lâu dài.
- Dữ liệu đảm bảo chính xác, nhất quán, đầy đủ và cập nhật liên tục. Tôi đã crawl được bộ dữ liệu hoàn chỉnh phục vụ cho mục đích thử nghiệm, xây dựng ứng dụng của bản thân.
- Xây dựng được chức năng gợi ý phim theo lịch sử hoạt động cũng như tìm kiếm phim bằng ngôn ngữ tự nhiên của người dùng.
- Cấu trúc mã nguồn của tôi cho phép mở rộng và có thể phát triển các tính năng nâng cao, dễ dàng kế thừa và phát triển tiếp.

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

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

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

- Đầu tiên là triển khai trên môi trường thực tế. Ứng dụng của tôi hiện tại đã phân chia ra đúng các gói, các lớp, các thành phần và đó là cơ sở để có thể dễ dàng deploy trong tương lai. Đây là một nhiệm vụ cần thiết để có thể phổ biến được ứng dụng của bản thân.
- Tôi cần cải thiện về giao diện ứng dụng. Ứng dụng của tôi hiện tại đang có

những tông màu rất đơn giản như đen, xanh, xanh lá cây. Để ứng dụng bắt mắt, được sự chú ý của người khác hơn thì tôi nghĩ nên cải thiện nhiều hơn về giao diện ứng dụng.

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

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

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

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

## TÀI LIỆU THAM KHẢO

- [1] *Boxofficemojo*, <https://www.boxofficemojo.com/>, Truy cập lần cuối vào ngày 01/06/2024.
- [2] *The numbers*, <https://www.the-numbers.com/>, Truy cập lần cuối vào ngày 01/06/2024.
- [3] *Imdb*, <https://www.imdb.com/>, Truy cập lần cuối vào ngày 01/06/2024.
- [4] *Metacritic*, <https://www.metacritic.com/>, Truy cập lần cuối vào ngày 01/06/2024.
- [5] *Rottentomatoes*, <https://www.rottentomatoes.com/>, Truy cập lần cuối vào ngày 01/06/2024.
- [6] *The moviedb*, <https://www.themoviedb.org/>, Truy cập lần cuối vào ngày 18/01/2026.
- [7] Microsoft, *Visual studio code (vscode)*, <https://code.visualstudio.com/>, Truy cập lần cuối vào ngày 01/02/2024.
- [8] Tom Christie, *Django rest framework*, <https://www.djangoproject-rest-framework.org/>, Truy cập lần cuối vào ngày 01/06/2024.
- [9] *Html*, <https://developer.mozilla.org/en-US/docs/Web/HTML>, Truy cập lần cuối vào ngày 15/04/2024.
- [10] *Tailwind css*, <https://tailwindcss.com/>, Truy cập lần cuối vào ngày 18/01/2026.
- [11] *React*, <https://react.dev/>, Truy cập lần cuối vào ngày 18/01/2026.
- [12] Django Software Foundation, *Django*, <https://www.djangoproject.com/>, Truy cập lần cuối vào ngày 01/06/2024.
- [13] *Mysql*, <https://www.mysql.com/>, Truy cập lần cuối vào ngày 01/06/2024.
- [14] *Selenium*, <https://www.selenium.dev/>, Truy cập lần cuối vào ngày 01/02/2024.
- [15] *Requests*, <https://requests.readthedocs.io/>, Truy cập lần cuối vào ngày 01/03/2024.
- [16] *Github actions*, <https://github.com/features/actions>, Truy cập lần cuối vào ngày 30/06/2024.

# **PHỤ LỤC**