

**ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



BÁO CÁO MÔN CÔNG NGHỆ PHẦN MỀM
Ngành: Trí tuệ nhân tạo

WEB XEM PHIM MOLFIX

Giảng viên: PGS.TS. Trương Anh Hoàng
ThS. Nguyễn Quang Minh

Thành viên: Phạm Thành Long
Nguyễn Đức Minh
Phan Văn Hiếu
Trần Tiến Nam

HÀ NỘI – 2024

1 Tầm nhìn sản phẩm

Ứng dụng web xem phim của chúng tôi nhằm cung cấp cho người dùng nền tảng trực tuyến thuận tiện để tìm kiếm, xem thông tin chi tiết về các bộ phim, và quản lý danh sách phim yêu thích của họ với một giá cả hợp lý. Chúng tôi hướng tới việc tạo ra một trải nghiệm người dùng thân thiện, mượt mà, nhanh chóng, và bảo mật, giúp người dùng dễ dàng tiếp cận phim ảnh mọi lúc, mọi nơi.

2 Phương pháp phát triển

Chúng tôi áp dụng phương pháp Agile, cụ thể là Scrum, để đảm bảo quá trình phát triển phần mềm linh hoạt và phản hồi nhanh với thay đổi.

- Sprints: Mỗi sprint kéo dài 2 tuần. Trong mỗi sprint, chúng tôi phát triển và hoàn thiện một phần chức năng của hệ thống.
- Daily Standups: Các cuộc họp ngắn hàng ngày (15 phút) để các thành viên trong nhóm cập nhật tiến độ, nêu các trở ngại và lên kế hoạch cho ngày tiếp theo.
- Sprint Planning: Cuộc họp đầu sprint để lập kế hoạch và xác định các nhiệm vụ cần hoàn thành trong sprint.
- Sprint Review: Cuối mỗi sprint, nhóm trình bày kết quả làm việc cho các bên liên quan để nhận phản hồi.
- Sprint Retrospective: Đánh giá lại những gì đã làm tốt và những gì cần cải thiện trong sprint vừa qua để rút kinh nghiệm cho sprint tiếp theo.

3 Personas

Personas là các mẫu người dùng đại diện cho các nhóm người dùng chính của sản phẩm.

1. Persona: John Doe

- Tuổi: 25
- Nghề nghiệp: Software Developer
- Khả năng sử dụng công nghệ: Tốt
- Nhu cầu: Tìm kiếm các bộ phim mới và lưu trữ danh sách phim yêu thích.
- Không mong muốn: mất nhiều thời gian để tìm kiếm thông tin về phim.

2. Persona: Jane Smith

- Tuổi: 30
- Nghề nghiệp: Nội trợ
- Khả năng sử dụng công nghệ: Thấp
- Nhu cầu: Cập nhật thông tin chi tiết về các bộ phim và viết nhận xét.
- Không mong muốn: Giao diện phức tạp và khó sử dụng.

3. Persona: Helen Keller

- Tuổi: 27
- Nghề nghiệp: Quản trị web
- Khả năng sử dụng công nghệ: Khá
- Nhu cầu: Có thể dễ dàng chỉnh sửa thông tin phim với tư cách là admin.
- Không mong muốn: Dữ liệu mất quá lâu để load.

4 Câu chuyện người dùng

Câu chuyện người dùng giúp mô tả các yêu cầu từ góc nhìn của người dùng.

- **Câu chuyện người dùng 1:** Là người dùng tôi muốn thanh tìm kiếm phim có thể nhanh chóng tìm được những bộ phim tôi quan tâm.
- **Câu chuyện người dùng 2:** Là người dùng đã đăng ký tôi muốn lưu phim vào danh sách yêu thích của mình để tôi có thể truy cập chúng dễ dàng sau này.
- **Câu chuyện người dùng 3:** Với tư cách là quản trị viên tôi muốn thêm và cập nhật thông tin phim để cơ sở dữ liệu luôn được cập nhật.

5 Kịch bản

Kịch bản 1: John muốn tìm kiếm một bộ phim hành động.

- John mở ứng dụng và nhập tên phim hoặc từ khóa liên quan vào thanh tìm kiếm.
- Hệ thống hiển thị bộ phim khớp với từ khóa tìm kiếm và danh sách các phim có thể loại liên quan đến kết quả tìm kiếm.
- John chọn một bộ phim từ danh sách và xem thông tin chi tiết về bộ phim đó.

Kịch bản 2: Jane muốn thêm một bộ phim mới vào cơ sở dữ liệu.

- Jane đăng nhập vào hệ thống với tài khoản quản trị viên.
- Jane chọn tùy chọn "Thêm phim mới" và điền vào các thông tin chi tiết về bộ phim.
- Jane nhấn "Lưu" và bộ phim mới được thêm vào cơ sở dữ liệu và hiển thị lên trên hệ thống.

6 Kiến trúc hệ thống

Hệ thống được thiết kế theo kiến trúc MVC và được chia thành ba phần chính: frontend, backend và database.

6.1 Frontend (View)

Công nghệ sử dụng: HTML, CSS, JavaScript, Node.js, Express.

Chức năng chính:

- Hiển thị web giao diện người dùng.
- Gọi API từ backend để lấy dữ liệu, hiển thị dữ liệu, truyền dữ liệu và chỉnh sửa dữ liệu.
- Xử lý các sự kiện người dùng và sự kiện admin.

6.2 Backend (Controller)

Công nghệ sử dụng: Node.js, Express.js

Chức năng chính:

- Xử lý các yêu cầu từ frontend thông qua các API.
- Giao tiếp với database để lấy, lưu trữ, sửa đổi dữ liệu thông qua Controller.
- Bảo mật giao diện frontend và api backend với JWT (JSON Web Tokens).

Các sự kiện xử lý giữa Frontend và Backend:

Xử lý các sự kiện người dùng	Đăng nhập Đăng ký tài khoản Quên mật khẩu Đổi mật khẩu Tìm kiếm Chọn phim Chọn thể loại phim Lưu phim và danh sách yêu thích Lọc phim Bình luận Đánh giá Mua gói người dùng Chỉnh sửa trang các nhân Xem danh sách các phim đã xem và đánh giá gần đây
Xử lý các sự kiện admin	Đăng nhập Xem danh sách phim Chỉnh sửa thông tin phim Xóa phim Ẩn phim Thêm phim Xem danh sách người dùng Xem thông tin người dùng Chỉnh sửa trạng thái người dùng Xóa người dùng Đổi gói người dùng Xem danh sách giao dịch gói người dùng

6.3 Database (Model)

Công nghệ sử dụng: MariaDB.

Cấu trúc dữ liệu:

- Bảng users: Lưu trữ thông tin người dùng.
- Bảng movie: Lưu trữ thông tin về các bộ phim.
- Bảng comment: Lưu trữ các bình luận phim.
- Bảng genre: Lưu trữ các thể loại phim.
- Bảng label: Lưu trữ các nhãn về giới hạn độ tuổi của phim.
- Bảng movie_genre: Lưu trữ các thể loại của từng bộ phim.
- Bảng movie_label: Lưu trữ nhãn của từng bộ phim.
- Bảng movie_view: Lưu trữ lượt xem của từng bộ phim.
- Bảng pricing_plan: Lưu thông tin của các gói đăng kí.
- Bảng purchase_method: Lưu thông tin của các phương thức mua gói.
- Bảng sale: Lưu thông tin lịch sử giao dịch của người dùng.
- Bảng user_history: Lưu lịch sử xem phim của từng người dùng.
- Bảng user_plan: Lưu Thông tin về gói đăng ký của người dùng.
- Bảng user_purchase: Lưu gói mua và phương thức mua của người dùng.
- Bảng user_rating: Lưu lịch sử đánh giá của người dùng với bộ phim.

7 Quy trình phát triển

Chúng tôi sử dụng các công cụ và kỹ thuật sau để đảm bảo quy trình phát triển hiệu quả:

7.1 Version Control

Sử dụng: Git và GitHub.

Quy trình:

- Mỗi tính năng hoặc sửa lỗi được phát triển trên một nhánh riêng (branch).
- Sử dụng pull request để kiểm tra mã và hợp nhất vào nhánh chính (main branch).

7.2 Tích hợp liên tục/Triển khai liên tục

Công cụ: GitHub Actions.

Quy trình:

- Tự động kiểm tra và triển khai mã nguồn sau mỗi lần đẩy mã (push).
- Chạy các bộ kiểm thử tự động để đảm bảo chất lượng mã nguồn.

7.3 Code Review

Quy trình:

- Mỗi pull request phải được ít nhất một thành viên trong nhóm xem xét và phê duyệt trước khi hợp nhất.
- Đảm bảo mã nguồn được kiểm tra kỹ lưỡng và tuân theo các tiêu chuẩn mã nguồn.

8 Bảo mật

8.1 Xác thực và Phân quyền

Sử dụng: JWT (JSON Web Tokens).

Quy trình:

- Người dùng đăng nhập và nhận JWT.
- Mỗi yêu cầu đến API phải kèm theo JWT để xác thực.

9 Kiểm thử phần mềm

Để kiểm thử API, chúng tôi sử dụng phần mềm Postman. Postman cho phép chúng tôi kiểm tra từng API một cách chi tiết, bao gồm việc gửi yêu cầu (request) và nhận kết quả trả về (response). Chúng tôi kiểm tra và xác nhận các phản hồi (responses) từ API, đảm bảo rằng dữ liệu trả về đúng như mong đợi và đáp ứng các yêu cầu chức năng.

Ngoài ra, Postman còn hỗ trợ việc xử lý các lỗi khác nhau bằng cách cung cấp các công cụ mạnh mẽ để ghi lại, theo dõi và phân tích các lỗi. Chúng tôi tạo và chạy các test tự động để kiểm tra tính ổn định và hiệu suất của API theo các kịch bản khác nhau. Điều này giúp chúng tôi nhanh chóng phát hiện và khắc phục sự cố, cải thiện chất lượng và độ tin cậy của hệ thống.

Postman cũng hỗ trợ việc tạo bộ sưu tập các yêu cầu API (collections), cho phép chúng tôi tổ chức và quản lý các thử nghiệm một cách khoa học. Chúng tôi sử dụng môi trường (environments) để cấu hình các biến số khác nhau, giúp dễ dàng chuyển đổi giữa các môi trường phát triển, kiểm thử và sản xuất. Các tính năng này giúp tối ưu hóa quy trình kiểm thử API, tiết kiệm thời gian và nâng cao hiệu quả làm việc của nhóm.

Các lỗi API chúng tôi tìm ra trong quá trình kiểm thử:

Lỗi Forbidden: Người dùng chưa đăng nhập thành công và cố vào các trang của hệ thống thông qua link thì sẽ bị chặn.

Lỗi Unauthorized: Phiên đăng nhập của người dùng đã hết hạn và người dùng vẫn tiếp tục sử dụng các chức năng của web.

Lỗi truyền request: lỗi xuất hiện khi request không được truyền thành công giữa Frontend và Backend. Lỗi này có thể do header truyền vào không đúng, chưa truyền vào token của JWT hoặc header truyền vào sai với định dạng của controller.

Lỗi khi xử lý API: lỗi xuất hiện khi các dữ liệu được truyền vào trong body của request đã đúng định dạng nhưng không vượt qua được các lệnh kiểm tra. Ví dụ: Khi người dùng muốn đổi mật khẩu thì họ cần nhập mật khẩu cũ, mật khẩu mới và xác nhận mật khẩu mới. Nhưng mật khẩu cũ người dùng nhập không chính xác thì khi xử lý request ở backend sẽ trả về response lỗi.

10 Đánh giá và Kết luận

Trong quá trình phát triển, chúng tôi đã áp dụng các phương pháp và công cụ hiện đại để đảm bảo chất lượng sản phẩm. Kết quả đạt được cho thấy hệ thống hoạt động ổn định, đáp ứng các yêu cầu của người dùng và đảm bảo tính bảo mật. Các phản hồi từ người dùng và các bên liên quan đều tích cực, đặc biệt là về trải nghiệm người dùng và hiệu suất hệ thống.

10.1 Phụ lục

Danh sách các công nghệ sử dụng: Node.js, Express.js, MariaDB, React, JWT, Git, GitHub, GitHub Actions, Postman.