**CineHub - User Stories**

**Tổng Quan Dự Án**

CineHub là một ứng dụng web xem phim hiện đại, cho phép người dùng duyệt, tìm kiếm và khám phá phim thông qua API của TMDB. Ứng dụng mang đến trải nghiệm người dùng mượt mà với giao diện hiện đại tương tự Netflix, hỗ trợ xác thực, tạo danh sách phim cá nhân hóa và cung cấp thông tin chi tiết về phim.

**Công Nghệ Sử dụng**

* **Frontend Framework**: Next.js với App Router
* **Ngôn Ngữ Lập Trình**: TypeScript
* **Styling**: Tailwind CSS với shadcn/ui components
* **Quản Lý State**: Zustand
* **Xác Thực**: NextAuth.js với JWT + Firebase (Google Auth, Facebook Auth)
* **Database**: MySQL (cho user data) + Firebase (Google auth, Facebook Auth)
* **API Tích Hợp**: TMDB API
* **Lấy Dữ Liệu**: Axios & TanStack Query
* **Triển Khai**: Vercel
* **Ngôn ngữ** : Tiếng anh

**User Stories**

**Xác Thực & Quản Lý Người Dùng**

**US-01: Đăng Ký Người Dùng**

**Là** một người dùng mới,  
**Tôi muốn** tạo một tài khoản bằng email và mật khẩu,  
**Để** có thể truy cập các tính năng cá nhân hóa và sidebar của ứng dụng.

**Tiêu Chí Chấp Nhận:**

* Người dùng có thể đăng ký bằng email và mật khẩu
* Mật khẩu phải đáp ứng yêu cầu bảo mật (tối thiểu 8 ký tự, một chữ hoa, một số)
* Thông tin được lưu vào MySQL với mật khẩu đã hash
* Hiển thị thông báo(Toast/Modal) thành công/lỗi với UI màu đã định nghĩa
* Người dùng được chuyển hướng đến trang đăng nhập sau khi đăng ký
* Giải thích rõ rằng đăng ký/đăng nhập sẽ mở khóa chức năng navbar và các tính năng cá nhân hóa

**US-02: Đăng Nhập Người Dùng**

**Là** một người dùng đã đăng ký,  
**Tôi muốn** đăng nhập vào tài khoản của mình,  
**Để** có thể truy cập sidebar và các tính năng cá nhân hóa.

**Tiêu Chí Chấp Nhận:**

* Người dùng có thể đăng nhập bằng email và mật khẩu đã đăng ký
* JWT được tạo và lưu trữ khi đăng nhập thành công
* Session được duy trì cho đến khi người dùng đăng xuất
* Hiển thị thông báo lỗi rõ ràng cho các lần đăng nhập thất bại
* Có tùy chọn quên mật khẩu cơ bản (extra)
* Sau khi đăng nhập, navbar được kích hoạt đầy đủ
* Hiển thị thông báo phải đăng nhập nếu user đã cố gắng truy cập các chức năng cần đăng nhập trên navbar
* Ghi nhớ đăng nhập trong 30 ngày (optional)

**US-03: Xác Thực Qua Google**

**Là** một người dùng,  
**Tôi muốn** đăng nhập bằng tài khoản Google của mình,  
**Để** có thể truy cập ứng dụng nhanh chóng mà không cần tạo tài khoản mới.

**Tiêu Chí Chấp Nhận:**

* Nút đăng nhập Google được hiển thị nổi bật với styling phù hợp BRD
* Sử dụng Firebase Authentication cho Google Auth
* Tích hợp với NextAuth.js để xử lý session
* Thông tin Google profile được sử dụng để tạo hoặc link với tài khoản CineHub
* Người dùng được chuyển hướng đến trang chủ sau khi xác thực thành công
* Xử lý trường hợp Google account đã được liên kết với email khác

**US-04: Xác Thực Qua Facebook**

**Là** một người dùng,  
**Tôi muốn** đăng nhập bằng tài khoản Facebook của mình,  
**Để** có thể truy cập ứng dụng nhanh chóng mà không cần tạo tài khoản mới.

**Tiêu Chí Chấp Nhận:**

* Nút đăng nhập Facebook được hiển thị nổi bật với styling phù hợp BRD
* Sử dụng Firebase Authentication cho Facebook Auth
* Tích hợp với NextAuth.js để xử lý session
* Thông tin Facebook profile được sử dụng để tạo hoặc link với tài khoản CineHub
* Người dùng được chuyển hướng đến trang chủ sau khi xác thực thành công
* Xử lý trường hợp Facebook account đã được liên kết với email khác

**US-05: Đăng Xuất Người Dùng**

**Là** một người dùng đã đăng nhập,  
**Tôi muốn** đăng xuất khỏi tài khoản của mình,  
**Để** phiên của tôi kết thúc và dữ liệu của tôi được bảo mật.

**Tiêu Chí Chấp Nhận:**

* Tùy chọn đăng xuất có sẵn trong user dropdown menu
* Phiên người dùng kết thúc khi đăng xuất
* JWT token bị clear khỏi client
* Người dùng được chuyển hướng đến trang chủ
* Sidebar trở về trạng thái bị hạn chế cho user chưa đăng nhập
* Xác nhận đăng xuất để tránh việc đăng xuất nhầm

**US-06: Quản Lý Người Dùng (Admin)**

**Là** một quản trị viên (Admin),  
**Tôi muốn** quản lý người dùng và theo dõi hệ thống,  
**Để** đảm bảo tài khoản và quyền hạn của người dùng được cập nhật chính xác.

**Tiêu Chí Chấp Nhận:**

* Admin có thể xem, tìm kiếm, tên/email
* Admin có thể phân quyền cho user theo cấp bậc tăng dần (chỉ có 1 admin): User => Moderator => Admin
* Các thay đổi được lưu vào MySQL database. Thông báo thành công/lỗi được hiển thị với màu sắc phù hợp BRD
* Admin và Moderator có thể tải ảnh lên để người dùng lựa chọn làm hình đại diện.
* Giao diện dashboard cho admin hiển thị biểu đồ lượt truy cập và thông tin cơ bản của user.
* Giao diện thiết kế đơn giản, dễ sử dụng, không bao gồm các chức năng phức tạp
* Khả năng vô hiệu hóa/kích hoạt tài khoản user (extra)
* Log hoạt động admin để theo dõi các thay đổi (extra)

**US-07: Quản Lý Hồ Sơ Người Dùng (Cơ Bản)**

**Là** một người dùng đã đăng ký,  
**Tôi muốn** cập nhật thông tin cơ bản của mình,  
**Để** tài khoản của tôi được cập nhật.

**Tiêu Chí Chấp Nhận:**

* Người dùng có thể cập nhật tên hiển thị và email
* Thay đổi mật khẩu yêu cầu xác minh mật khẩu hiện tại
* Các thay đổi được lưu vào MySQL database
* Thông báo thành công/lỗi được hiển thị với màu sắc phù hợp BRD
* Người dùng có thể chọn các hình đại diện có sẵn được admin và moderator tải lên.

**Duyệt & Khám Phá Phim**

**US-08: Duyệt Phim Từ TMDB**

**Là** một người dùng,  
**Tôi muốn** duyệt phim theo các danh mục từ TMDB,  
**Để** có thể khám phá nội dung mà tôi quan tâm.

**Tiêu Chí Chấp Nhận:**

* Trang chủ hiển thị phim thịnh hành/phổ biến từ TMDB API
* Phim được hiển thị theo các category: Phổ biến (Popular), Được đánh giá cao nhất (Top Rated), Đang chiếu (Now Playing), Sắp chiếu (Upcoming).
* Movie cards hiển thị poster, title, rating và release date
* Sử dụng TanStack Query để cache API responses
* Thiết kế theo mãu trong BRD

**US-09: Search Page**

**Là** một người dùng,  
**Tôi muốn** tìm kiếm phim hoặc TV shows theo từ khoá,  
**Để** nhanh chống tìm thấy nội dung muốn tìm.

**Tiêu Chí Chấp Nhận:**

* Thanh tìm kiếm( Trang ) có thể truy cập từ header
* Giao diện tìm kiếm có input và nút search rõ ràng
* Hỗ trợ tìm kiếm real-time với debounce để tránh gọi API liên tục
* Kết quả hiển thị dưới dạng moviecard hoặc tvshowcard
* Sử dụng TMDB Search API
* Hiển thị thông báo “Không tìm thấy kết quả” nếu không có phim nào phù hợp
* Lưu lịch sử tìm kiếm cho người dùng đã đăng nhập
* Responsive UI, tối ưu cả desktop lẫn mobile
* Sử dụng loading đã cấu hình sẵn
* Hỗ trợ tìm kiếm cho cả **Movie** và **TV Show**
* Có filter để chọn loại nội dung: All | Movie | TV Show

**US-10: Xem Chi Tiết Phim (Movie Detail)**

**Là** một người dùng**,**  
**Tôi muốn** xem thông tin chi tiết về một bộ phim,  
**Để** có thể tìm hiểu thêm về nó trước khi quyết định xem.

**Tiêu Chí Chấp Nhận:**

* Trang chi tiết phim lấy dữ liệu từ TMDB API
* Hiển thị: poster, backdrop, title, release date, runtime, genres, rating, overview
* Tabs nội dung gồm:
  + **Overall**: thông tin chung như mô tả, runtime, thể loại, ngày phát hành
  + **Cast**: diễn viên & đoàn làm phim
  + **Reviews**: đánh giá từ user (username, rating, comment)
  + **Media**: trailer từ YouTube (nếu có)
* Gợi ý phim tương tự từ TMDB (similar movies)
* Responsive design theo mobile-first
* Có nút share link phim(extra)
* Loading states cho từng tab

**US-11: Xem Chi Tiết TV Show**

**Là** một người dùng**,**  
**Tôi muốn** xem thông tin chi tiết về một TV Show,  
**Để** có thể khám phá các mùa và tập phim liên quan.

**Tiêu Chí Chấp Nhận:**

* Trang chi tiết TV Show lấy dữ liệu từ TMDB API
* Hiển thị: poster, backdrop, title, first air date, genres, rating, overview
* Tabs nội dung gồm:
  + **Overall**: mô tả chung, ngày chiếu đầu tiên, thể loại
  + **Cast**: diễn viên chính
  + **Reviews**: đánh giá từ người dùng
  + **Seasons**: danh sách các mùa, hiển thị season poster + số tập (drop down và tab riêng)
  + **Media**: trailer hoặc video liên quan từ Youtube
* Gợi ý TV show tương tự từ TMDB (similar TV Shows)
* Responsive design theo mobile-first
* Navigation giữa các mùa và tập

**US-12: Lọc Phim và TV show Cơ Bản**

**Là** một người dùng,  
**Tôi muốn** lọc phim hoặc tv show dựa trên tiêu chí cơ bản,  
**Để** có thể thu hẹp kết quả tìm kiếm của mình.

**Tiêu Chí Chấp Nhận:**

* Filter options: genres, year, runtime, release date(from… to...) (sử dụng TMDB discover API)
* Nhiều bộ lọc (Multiple filters) có thể được áp dụng đồng thời
* Hiển thị trực quan các bộ lọc đang hoạt động.
* Xoá bộ lọc (Clear Fillter)
* Filter state được maintain trong session và được quản lý bới zustand
* Tuỳ chọn short (most popularity, most rating, most recent)
* Reset về default filters

**Cá Nhân Hóa & Tương Tác Người Dùng**

**US-13: Tạo Watchlist**

**Là** một người dùng đã đăng nhập,  
**Tôi muốn** thêm phim vào watchlist của mình,  
**Để** có thể theo dõi các bộ phim tôi muốn xem sau.

**Tiêu Chí Chấp Nhận:**

* Nút “add to watchlist” trên detail pages
* Hiển thị trực quan các phim hoặc TV shows đã có trong watchlist
* Watchlist data được lưu trong MySQL
* Thông báo thành công/lỗi được hiển thị với màu sắc phù hợp BRD
* Watchlist có thể truy cập từ navbar/header (cho tất cả ai đã đăng nhập)

**US-14: Đánh Giá Phim Cơ Bản**

**Là** một người dùng đã đăng nhập,  
**Tôi muốn** đánh giá phim,  
**Để** có thể lưu trữ ý kiến của mình về phim.

**Tiêu Chí Chấp Nhận:**

* Hệ thống đánh giá phim cơ bản (1-5 stars)
* Rating component được hiển thị trên movie detail page
* User ratings được lưu trong MySQL
* User có thể update/delete ratings của mình
* Display user's personal rating riêng biệt với TMDB rating
* User có thể comment ý kiến riêng của mình về phim
* User Comments được lưu trong MySQL

**US-15: Đề Xuất Phim Cơ Bản**

**Là** một người dùng đã đăng nhập,  
**Tôi muốn** nhận đề xuất phim cơ bản,  
**Để** có thể khám phá các bộ phim phù hợp với sở thích của tôi.

**Tiêu Chí Chấp Nhận:**

* Recommendations dựa trên watchlist và ratings của user
* Sử dụng TMDB similar movies API
* Đề xuất được phân loại theo ("Because you liked X")
* Sử dụng thuật toán đơn giản dựa trên sở thích để phân loạ
* Không cần dùng AI để làm phức tạp
* Refresh recommendations định kỳ (extra)

**US-16: Lưu Lịch Sử Xem Phim**

**Là** một người dùng đã đăng nhập,  
**Tôi muốn** hệ thống lưu lại những phim và TV Show tôi đã xem,  
**Để** tôi có thể dễ dàng tiếp tục xem hoặc tìm lại các nội dung tôi đã quan tâm trước đó.

**Tiêu Chí Chấp Nhận:**

* Khi người dùng truy cập trang chi tiết của một phim hoặc TV Show và nhấn nút “Watch”, nội dung đó sẽ được thêm vào danh sách **History**.
* Mỗi mục trong lịch sử lưu:
  + Loại nội dung: Movie hoặc TV Show
  + Theo movie card và tv show card có sẵn
* Hiển thị danh sách “Đã xem gần đây” (**History**) trong mục avatar của header nếu user đã đăng nhập.
* Danh sách được sắp xếp theo thứ tự **xem gần nhất → cũ hơn**
* Người dùng có thể xóa từng mục khỏi lịch sử
* Dữ liệu được lưu vào **MySQL** để đảm bảo riêng biệt giữa người dùng
* Responsive UI trên cả desktop và mobile
* Lịch sử xem phim cũng sẽ hiển thị thông tin trong profile user nếu đã đăng nhập

**Giao Diện & Khả Năng Tiếp Cận**

**US-17: Thiết Kế Responsive Mobile-First**

**Là** một người dùng,  
**Tôi muốn** truy cập ứng dụng trên các thiết bị khác nhau,  
**Để** có thể sử dụng nó ở bất kỳ đâu và bất kỳ lúc nào.

**Tiêu Chí Chấp Nhận:**

* Mobile-first responsive design
* Hamburger menu cho navigation trên mobile
* Các yếu tố giao diện dễ tương tác bằng cảm ứng (touch-friendly)
* Kích thước văn bản và độ tương phản phù hợp theo bảng màu BRD
* Landing page riêng cho mobile users (first-time visitors)
* Optimize (extra)

**US-18: Hỗ Trợ Accessibility Cơ Bản (Optional)**

**Là** một người dùng có nhu cầu accessibility,  
**Tôi muốn** sử dụng ứng dụng với screen reader và keyboard navigation,  
**Để** có thể truy cập nội dung một cách hiệu quả.

**Tiêu Chí Chấp Nhận:**

* Cấu trúc HTML ngữ nghĩa (Semantic HTML structure)
* Nhãn ARIA phù hợp cho các yếu tố tương tác (interactive elements)
* Hỗ trợ điều hướng bằng bàn phím (Keyboard navigation support)
* Độ tương phản màu sắc đủ theo hướng dẫn BRD
* Alt text cho images
* Focus indicators rõ ràng

**Hiệu Suất tối ưu**

**US-19: Performance Optimization**

**Là** một người dùng,  
**Tôi muốn** trải nghiệm loading nhanh và smooth interactions,  
**Để** có thể sử dụng ứng dụng hiệu quả.

**Tiêu Chí Chấp Nhận:**

* Image lazy loading với next/image
* Code splitting với dynamic imports
* TanStack Query caching cho API responses
* Skeleton loaders cho content đang load
* Thêm Lazy Loading cho các page đang hoặc sắp được load

**Mobile & Bản Quyền**

**US-20: Mobile Landing Page**

**Là** một người dùng mobile,  
**Tôi muốn** thấy landing page giới thiệu khi truy cập lần đầu,  
**Để** hiểu về ứng dụng trước khi sử dụng.

**Tiêu Chí Chấp Nhận:**

* Landing page riêng cho mobile user truy cập lần đầu
* Giới thiệu các tính năng và lợi ích chính
* Nút CTA "Enter App" rõ ràng
* Tải nhanh với tài nguyên được tối ưu hóa
* Thương hiệu nhất quán với main app
* Tùy chọn bỏ qua cho người dùng quay lại

**US-21: Movie Viewing Interface Với Copyright Handling**

**Là** một người dùng,  
**Tôi muốn** thấy thông tin về khả năng xem phim,  
**Để** hiểu về hạn chế nội dung vì bản quyền.

**Tiêu Chí Chấp Nhận:**

* Trang chi tiết phim có phần về các tùy chọn xem
* Mẫu giao diện lỗi cho các hạn chế bản quyền
* Thông báo nhất quán về khả năng truy cập nội dung
* Liên kết đến các nền tảng xem hợp pháp khi có sẵn
* Nội dung giáo dục về việc xem phim hợp pháp
* Responsive error pages

**Extra Feature**

**US-22: Explore Page**

**Là** một người dùng,  
**Tôi muốn** khám phá các bộ phim hoặc TV shows nổi bật được lọc và phân loại,  
**Để** có thể dễ dàng tìm thấy nội dung mới hoặc phù hợp với sở thích.

**Tiêu Chí Chấp Nhận:**

* Trang “Explore” hiển thị nhiều nhóm phim hoặc TV shows (chia ra từng mục movie riêng tv show riêng):
  + Trending (xu hướng hiện tại)
  + Popular in your country (nếu TMDB hỗ trợ)
  + Đề xuất theo thể loại (Genre Recommendations)
* Lọc phim hoặc tv show cơ bản theo US-12
* Trên trang explore sẽ có 1 thanh search cho user nhập khi nhập xong chuyển hướng user về trang search
* Mỗi phim hiển thị theo moviecard hoặc tvshowcard
* Responsive UI với thiết kế theo brand CineHub
* Sử dụng TMDB API hoặc TMDB Discover endpoint
* Có nút "View More" để đi đến danh mục đầy đủ nếu applicable
* Caching bằng TanStack Query để giảm số lượng gọi API
* Loading state cho từng section
* Quản lý state bằng zustand
* Nếu không có dữ liệu (ví dụ không có phim trending), hiển thị thông báo “No Content Data”

**Các trường hợp sử dụng:**

* User click vào các menu item chưa hoàn thành
* Truy cập các routes đang trong quá trình phát triển
* Thay thế cho các placeholder components

**Ghi Chú Triển Khai Kỹ Thuật**

**Triển khai hệ thống**

* Triển khải bảng màu và kiểu chữ theo thông số bên BRD
* Sử dụng Tailwind CSS
* Các components bên shadcn/ui với kiểu dáng tùy chỉnh
* Khoảng cách, bán kính đường viền và phân cấp hình ảnh nhất quán

**Luồng xác thực**

* Thiết lập NextAuth.js với email/mật khẩu, Google và Facebook
* MySQL database cho user accounts
* Firebase chỉ cho Google và Facebook authentication
* JWT token management
* Middleware cho route protection

**Kết nối API**

* Wrapper API TMDB với xử lý lỗi
* TanStack Query cho data fetching và caching
* Nhận thức về giới hạn tốc độ (Rate limiting awareness)
* Xử lý fallback handling cho các lỗi API