**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: LẬP TRÌNH MOBILE**

ĐỀ TÀI: Xây dựng ứng dụng xem phim trên nền tảng Android sử dụng Android Studio

**Giáo viên hướng dẫn: ThS. Đặng Khánh Trung**

**Sinh viên thực hiện:**

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Mã SV | Họ và tên | Lớp |
| 1 | 1671020267 | Vũ Đình Quân | CNTT 16-04 |

**Hà Nội, năm 2025**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: LẬP TRÌNH MOBILE**

**ĐỀ TÀI: XÂY DỰNG ỨNG DỤNG XEM PHIM TRÊN NỀN TẢNG ANDROID SỬ DỤNG ANDROID STUDIO**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Mã Sinh Viên | Họ và Tên | Ngày Sinh | Điểm | |
| Bằng Số | Bằng Chữ |
| 1 | 1671020267 | Vũ Đình Quân | 08/04/2004 |  |  |

**CÁN BỘ CHẤM THI 1 CÁN BỘ CHẤM THI 2**

**Hà Nội, năm 2025**

**LỜI NÓI ĐẦU**

Ngày nay, nhu cầu giải trí thông qua các nền tảng xem phim trực tuyến ngày càng phổ biến, đặc biệt là trên các thiết bị di động chạy hệ điều hành Android. Với sự phát triển mạnh mẽ của công nghệ, việc xây dựng một ứng dụng xem phim trên Android không chỉ mang lại trải nghiệm tiện ích cho người dùng mà còn giúp các nhà phát triển tiếp cận với các công nghệ hiện đại như phát trực tuyến video, tối ưu hóa giao diện người dùng và tích hợp API từ bên thứ ba. Ứng dụng này được phát triển bằng Android Studio, môi trường lập trình mạnh mẽ, hỗ trợ nhiều công cụ hữu ích giúp lập trình viên dễ dàng xây dựng và tối ưu hóa sản phẩm.

Dự án tập trung vào các tính năng quan trọng như phát video trực tuyến từ máy chủ hoặc các nền tảng video phổ biến, giao diện thân thiện hỗ trợ tìm kiếm và phân loại phim, quản lý danh sách yêu thích, lịch sử xem phim cũng như tích hợp Firebase để lưu trữ dữ liệu và quản lý người dùng. Thông qua ứng dụng này, người dùng có thể trải nghiệm xem phim mượt mà, chất lượng cao ngay trên thiết bị Android của mình. Đồng thời, đây cũng là một dự án giúp lập trình viên nâng cao kỹ năng phát triển ứng dụng Android, từ việc xử lý dữ liệu, tối ưu hiệu suất cho đến triển khai và bảo trì ứng dụng trên nền tảng di động. Chính vì vậy, đề tài “Xây dựng ứng dụng xem phim trên nền tảng Android sử dụng Android Studio” được lựa chọn nhằm nghiên cứu, phát triển một ứng dụng đáp ứng tốt hơn nhu cầu giải trí của người dùng.

**MỤC LỤC**

[**CHƯƠNG 1. CƠ SỞ LÝ THUYẾT** 5](#_Toc193776260)

[**1.** **Lập trình Mobile là gì ?** 5](#_Toc193776261)

[**2.** **Android studio là gì ?** 6](#_Toc193776262)

[**3.** **Flutter là gì ?** 7](#_Toc193776263)

[**4.** **Fire base là gì ?** 8](#_Toc193776264)

[**CHƯƠNG 2. XÂY DỰNG CHƯƠNG TRÌNH** 10](#_Toc193776265)

[**1.** **Giới thiệu các thư viện** 10](#_Toc193776266)

[**2.** **Xây dựng ứng dụng** 11](#_Toc193776267)

[3. **Tổ chức mã nguồn (lib)** 12](#_Toc193776268)

[4. **Chi tiết mã nguồn** 13](#_Toc193776269)

[a) Trang main.dart 13](#_Toc193776270)

[b) Trang dangnhap.dart 18](#_Toc193776271)

[c) Trang api.service.dart 19](#_Toc193776273)

[d) Trang movie.dart 20](#_Toc193776274)

[e) Trang watchlistScreen.dart 21](#_Toc193776275)

[f) Trang movie\_detail\_screen.dart 24](#_Toc193776276)

[**CHƯƠNG 3. KẾT QUẢ CHƯƠNG TRÌNH** 25](#_Toc193776277)

[**1.** **Trang chủ** 25](#_Toc193776278)

[**2.** **Trang đăng nhập** 26](#_Toc193776279)

[**3.** **Trang thông báo** 26](#_Toc193776280)

[**4.** **Trang đánh giá** 27](#_Toc193776281)

[**5.** **Trang phim bộ** 28](#_Toc193776282)

[**6.** **Trang phim điện ảnh** 29](#_Toc193776283)

[**7.** **Trang phim hành động** 30](#_Toc193776284)

[**8.** **Trang phim xem sau** 31](#_Toc193776285)

[**KẾT LUẬN** 32](#_Toc193776286)

[**DANH MỤC TÀI LIỆU THAM KHẢO** 34](#_Toc193776287)

# **CHƯƠNG 1. CƠ SỞ LÝ THUYẾT**

## **Lập trình Mobile là gì ?**

* Lập trình mobile là quá trình phát triển ứng dụng cho thiết bị di động trên nền tảng Android và iOS. Nó bao gồm thiết kế giao diện, xử lý dữ liệu, tối ưu hiệu suất và triển khai ứng dụng. Có hai phương pháp chính: native (Android dùng Java/Kotlin, iOS dùng Swift) và cross-platform (Flutter, React Native). Lập trình mobile được ứng dụng rộng rãi trong thương mại, giải trí, tài chính và giáo dục, mang lại trải nghiệm tiện ích cho người dùng và cơ hội lớn cho lập trình viên.



* Các nền tảng lập trình mobile phổ biến:
* Android: Sử dụng ngôn ngữ Java, Kotlin, hoặc Flutter (Dart), phát triển trên Android Studio.
* iOS: Sử dụng Swift hoặc Objective-C, phát triển trên Xcode.
* Hybrid & Cross-Platform: Các công nghệ như Flutter, React Native, Ionic giúp xây dựng ứng dụng chạy trên cả Android và iOS từ một codebase duy nhất.
* Lập trình Mobile không chỉ yêu cầu kỹ năng về ngôn ngữ lập trình mà còn đòi hỏi kiến thức về giao diện người dùng (UI/UX), cơ sở dữ liệu, API, bảo mật và tối ưu hiệu suất ứng dụng. Với sự hỗ trợ của các công cụ và nền tảng phát triển hiện đại, lập trình viên có thể dễ dàng xây dựng các ứng dụng di động mạnh mẽ, tiện ích và đáp ứng nhu cầu người dùng một cách hiệu quả.
* Nhờ sự phát triển không ngừng của công nghệ di động, lập trình Mobile tiếp tục là một lĩnh vực đầy tiềm năng, mang đến cơ hội nghề nghiệp hấp dẫn cho các lập trình viên trên toàn thế giới.

## **Android studio là gì ?**

* Android Studio là môi trường phát triển tích hợp (IDE) chính thức do Google phát triển dành cho lập trình viên Android. Đây là công cụ mạnh mẽ giúp lập trình viên xây dựng, kiểm thử và triển khai ứng dụng Android một cách hiệu quả. Android Studio được xây dựng dựa trên IntelliJ IDEA, cung cấp nhiều tính năng hữu ích như trình giả lập (Emulator), hỗ trợ lập trình bằng Java, Kotlin, C++, giao diện kéo-thả thiết kế UI, kiểm thử ứng dụng và công cụ tối ưu hiệu suất.



* Các tính năng chính của Android Studio:
* **Trình giả lập (Android Emulator)** giúp kiểm thử ứng dụng mà không cần thiết bị thật.
* **Hỗ trợ nhiều ngôn ngữ lập trình** như **Java, Kotlin, C++**.
* **Giao diện thiết kế UI trực quan** với chế độ kéo-thả (Layout Editor).
* **Công cụ kiểm thử và debug mạnh mẽ**, hỗ trợ kiểm tra lỗi nhanh chóng.
* **Tích hợp Firebase, Google Cloud** giúp lưu trữ dữ liệu và xác thực người dùng dễ dàng.
* **Hệ thống build mạnh mẽ với Gradle**, tối ưu hiệu suất phát triển.

## **Flutter là gì ?**

* Flutter là một bộ công cụ phát triển giao diện (UI toolkit) mã nguồn mở do Google phát triển, giúp lập trình viên xây dựng ứng dụng cross-platform (đa nền tảng) chạy trên Android, iOS, web, desktop chỉ với một codebase duy nhất. Flutter sử dụng ngôn ngữ lập trình Dart và cung cấp UI framework mạnh mẽ, giúp tạo giao diện mượt mà, đẹp mắt với hiệu suất cao.



* Tính năng nổi bật của Flutter:
* Viết một lần, chạy trên nhiều nền tảng (Android, iOS, web, desktop).
* Hiệu suất cao, sử dụng đồ họa Skia để vẽ giao diện trực tiếp lên màn hình mà không cần qua trung gian như các framework khác.
* Hot Reload, giúp lập trình viên xem ngay thay đổi khi chỉnh sửa code mà không cần khởi động lại ứng dụng.
* Thư viện widget phong phú, hỗ trợ tùy chỉnh linh hoạt, tạo giao diện đẹp mắt.
* Hỗ trợ Firebase, API bên thứ ba, dễ dàng kết nối cơ sở dữ liệu và dịch vụ đám mây.

## **Fire base là gì ?**

* Firebase là nền tảng dịch vụ đám mây do Google phát triển, cung cấp nhiều công cụ hỗ trợ lập trình viên xây dựng, phát triển và quản lý ứng dụng web & mobile một cách dễ dàng. Firebase giúp giảm tải công việc backend, cho phép tập trung vào phát triển tính năng của ứng dụng mà không cần lo về hạ tầng máy chủ.\



* Các dịch vụ chính của Firebase:
* Realtime Database & Firestore: Cơ sở dữ liệu NoSQL giúp lưu trữ và đồng bộ dữ liệu theo thời gian thực.
* Firebase Authentication: Xác thực người dùng bằng Google, Facebook, Email, SĐT…
* Firebase Storage: Lưu trữ hình ảnh, video, file trên đám mây.
* Cloud Messaging (FCM): Gửi thông báo đẩy (Push Notification) đến thiết bị.
* Firebase Hosting: Lưu trữ và triển khai ứng dụng web nhanh chóng.
* Crashlytics: Giám sát và báo lỗi ứng dụng theo thời gian thực.

# **CHƯƠNG 2. XÂY DỰNG CHƯƠNG TRÌNH**

## **Giới thiệu các thư viện**

**Ảnh có chứa văn bản, ảnh chụp màn hình

Nội dung do AI tạo ra có thể không chính xác.**

Tệp Pubspec.yaml của dự án Flutter bao gồm:

* Flutter:
* Là thư viện cốt lõi của Flutter, cung cấp các công cụ, widget và API để xây dựng giao diện người dùng.
* Đây là thành phần bắt buộc trong mọi dự án Flutter.
* cupertino\_icons:
* Cung cấp bộ icon theo phong cách iOS (Cupertino) để sử dụng trong ứng dụng Flutter.
* Giúp hiển thị biểu tượng UI đẹp mắt, phù hợp khi thiết kế giao diện giống iOS.
* youtube\_player\_flutter:
* Thư viện giúp nhúng và phát video từ YouTube trực tiếp trong ứng dụng Flutter.
* Hỗ trợ điều khiển phát lại, toàn màn hình, tua nhanh, tạm dừng…
* Rất hữu ích trong ứng dụng xem phim có tích hợp video từ YouTube.
* shared\_preferences:
* Dùng để lưu trữ dữ liệu cục bộ trên thiết bị như cài đặt, trạng thái đăng nhập, lịch sử xem phim.
* Hoạt động như một hệ thống lưu trữ key-value, phù hợp để lưu dữ liệu nhỏ.
* http:
* Thư viện giúp gửi và nhận dữ liệu từ API thông qua giao thức HTTP.
* Hỗ trợ các phương thức như GET, POST, PUT, DELETE, giúp ứng dụng lấy danh sách phim từ server.

## **Xây dựng ứng dụng**

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, thiết kế

Nội dung do AI tạo ra có thể không chính xác.

* Tệp Lib của dự án bao gồm 1 thư mục con tên (models, services, widgets) cùng với các tệp quan trọng như movie.dart và api\_service.dart và nhiều tệp con khác.

## **Tổ chức mã nguồn (lib)**

Thư mục lib/ chứa các tệp chính để xây dựng chức năng của app:

|  |  |
| --- | --- |
| **Tệp** | **Chức năng chính** |
| main.dart | Điểm khởi chạy của app, cấu hình routes, giao diện chính. |
| movie.dart | Làm trung gian giữa API và ứng dụng Flutter, giúp chuyển đổi dữ liệu từ JSON sang đối tượng dễ xử lý và hỗ trợ hiển thị danh sách phim |
| api\_service.dart | Gửi yêu cầu API để lấy danh sách phim từ server. |
| dangnhap.dart | Đăng nhập và đăng kí |
| danhgia.dart | Hiện thông tin đánh giá |
| dienanh.dart | Hiện các banner phim điện ảnh |
| hanhdong.dart | Hiện các banner phim hành động |
| phimbo.dart | Hiện các banner phim bộ |
| thongbao.dart | Hiện các thông báo cập nhật hoặc phản hồi |
| movie\_detail\_screen.dart | - Xem chi tiết phim (hình ảnh, tiêu đề, mô tả). - Xem trailer phim với trình phát YouTube nhúng. - Lưu phim vào danh sách "Xem Sau", giúp người dùng theo dõi phim yêu thích. |
| watchlist\_screen.dart | Trang lưu trữ những phim được bấm xem sau |

## **Chi tiết mã nguồn**

### Trang main.dart

**Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.**

* Ứng dụng sử dụng MaterialApp.
* Giao diện tối (ThemeData.dark()).
* Màn hình chính là MovieScreen().

**Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.**

* Đây là StatefulWidget, cho phép cập nhật UI khi dữ liệu thay đổi.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

* chứa danh sách ảnh banner.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

* Cứ 3 giây, banner sẽ tự động chuyển sang ảnh tiếp theo.
* Nếu đã đến ảnh cuối cùng → Quay lại ảnh đầu.

**Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm

Nội dung do AI tạo ra có thể không chính xác.**

* Có thanh tìm kiếm phim
* Có thông báo, tin nhắn, tài khoản cá nhân

**Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm

Nội dung do AI tạo ra có thể không chính xác.**

* Có danh mục như "Lịch sử", "Phim yêu thích", "Xem sau"
* Bấm vào "Xem sau" → Chuyển sang WatchlistScreen()

**Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.**

* Hiển thị danh sách phim hot
* Dạng cuộn ngang

**Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.**

* Hiển thị ảnh bấm vào chuyển sang (moviedetailscreen)

### Trang dangnhap.dart

### Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, màn hình Nội dung do AI tạo ra có thể không chính xác.

* Chuyển đổi giữa đăng nhập & đăng ký
* Form nhập tài khoản, mật khẩu
* Hiệu ứng UI đẹp mắt với gradient
* Chuyển hướng sang màn hình chính sau khi đăng nhập

### Trang api.service.dart

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Nội dung do AI tạo ra có thể không chính xác.

* Gọi API từ MockAPI.
* Parse dữ liệu JSON thành danh sách Movie.
* Xử lý lỗi khi API thất bại.

### Trang movie.dart

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình

Nội dung do AI tạo ra có thể không chính xác.

* Làm trung gian giữa API và ứng dụng Flutter, giúp chuyển đổi dữ liệu từ JSON sang đối tượng dễ xử lý và hỗ trợ hiển thị danh sách phim

### Trang watchlistScreen.dart

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Nội dung do AI tạo ra có thể không chính xác.

* flutter/material.dart: Thư viện UI chính của Flutter.
* dart:convert: Dùng để chuyển đổi JSON.
* shared\_preferences: Dùng để lưu dữ liệu phim vào bộ nhớ thiết bị.
* models/movie.dart: Import model Movie để map dữ liệu.
* movie\_detail\_screen.dart: Màn hình hiển thị chi tiết phim.
* WatchlistScreen kế thừa StatefulWidget → Cần setState() để cập nhật UI khi danh sách thay đổi.
* watchlist: Danh sách lưu phim.
* watchlistKey: Khóa để lưu dữ liệu vào SharedPreferences.
* Lấy dữ liệu từ SharedPreferences.
* Nếu có dữ liệu, parse JSON thành danh sách Movie rồi cập nhật UI.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

* Tìm và xóa phim khỏi danh sách watchlist.
* Cập nhật SharedPreferences để lưu danh sách mới.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

* Tiêu đề: Danh sách xem sau.
* Nếu không có phim → Hiển thị text thông báo.
* Hiển thị ảnh phim.
* Nếu ảnh là URL → Dùng NetworkImage.
* Nếu ảnh là file cục bộ → Dùng AssetImage.
* Hiển thị tên phim.

### Trang movie\_detail\_screen.dart

Ảnh có chứa văn bản, ảnh chụp màn hình

Nội dung do AI tạo ra có thể không chính xác.

* Thêm phim vào danh sách "Xem Sau".
* Lưu danh sách phim vào SharedPreferences.
* Hiển thị danh sách phim "Xem Sau".
* Xóa phim khỏi danh sách.
* Phát trailer phim bằng YoutubePlayer**.**

# **CHƯƠNG 3. KẾT QUẢ CHƯƠNG TRÌNH**

## **Trang chủ**

Ảnh có chứa văn bản, trang phục, ảnh chụp màn hình, Mặt người

Nội dung do AI tạo ra có thể không chính xác.

## **Trang đăng nhập**

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Nội dung do AI tạo ra có thể không chính xác.

## **Trang thông báo**

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

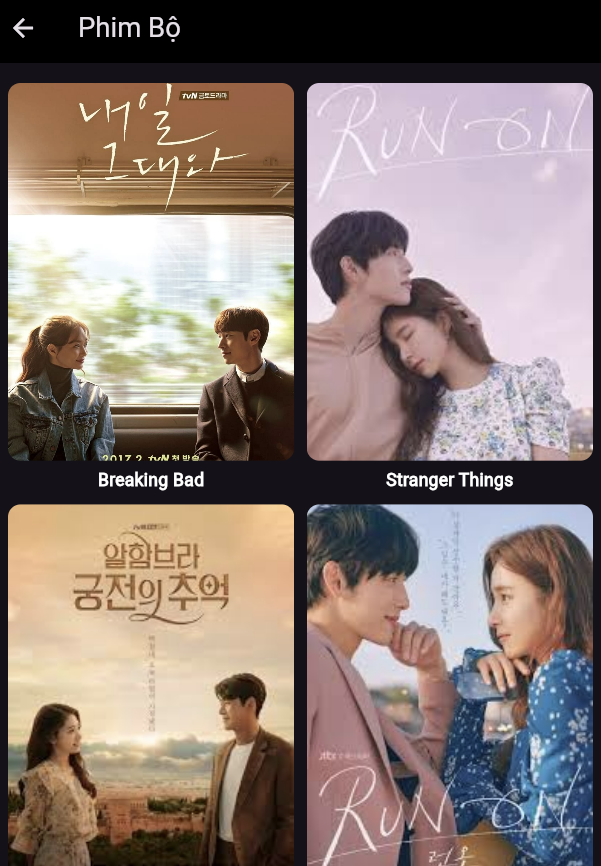
Nội dung do AI tạo ra có thể không chính xác.

## **Trang đánh giá**

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

## **Trang phim bộ**



## **Trang phim điện ảnh**

Ảnh có chứa văn bản, ảnh chụp màn hình, đàn ông, hư cấu

Nội dung do AI tạo ra có thể không chính xác.

## **Trang phim hành động**

Ảnh có chứa văn bản, hư cấu, áp phích, thiết kế đồ họa

Nội dung do AI tạo ra có thể không chính xác.

## **Trang phim xem sau**

Ảnh có chứa văn bản, Mặt người, trang phục, phụ nữ

Nội dung do AI tạo ra có thể không chính xác.

# **KẾT LUẬN**

**Ưu điểm**

* Ứng dụng xem phim trên Android mang lại nhiều lợi ích quan trọng, đặc biệt là tính tiện lợi và khả năng tiếp cận người dùng rộng rãi. Với sự hỗ trợ của Android Studio, nhà phát triển có thể dễ dàng tạo ra một ứng dụng có giao diện đẹp mắt, thân thiện với người dùng. Ứng dụng có thể tích hợp API để lấy dữ liệu phim từ nhiều nguồn khác nhau, giúp cập nhật nội dung một cách nhanh chóng. Ngoài ra, việc sử dụng các thư viện phát video như ExoPlayer giúp cải thiện trải nghiệm xem phim với chất lượng cao và tốc độ tải nhanh. Một ưu điểm khác là ứng dụng có thể cá nhân hóa nội dung, cho phép người dùng lưu danh sách phim yêu thích hoặc đề xuất phim dựa trên sở thích, giúp tăng sự gắn kết với ứng dụng.

**Nhược điểm**

* Bên cạnh những lợi ích, ứng dụng cũng gặp phải một số hạn chế đáng kể. Một trong những thách thức lớn nhất là vấn đề bản quyền phim, vì nếu không có giấy phép hợp pháp, ứng dụng có thể bị gỡ bỏ hoặc gặp rắc rối pháp lý. Ngoài ra, nếu sử dụng API miễn phí để lấy dữ liệu phim, có thể bị giới hạn về số lượng request hoặc tốc độ truy cập, ảnh hưởng đến trải nghiệm người dùng. Việc phát video cũng yêu cầu hiệu suất thiết bị cao và đường truyền ổn định, gây khó khăn cho người dùng có thiết bị yếu hoặc kết nối Internet không ổn định. Hơn nữa, nếu ứng dụng không có nội dung độc quyền hoặc tính năng nổi bật, khả năng cạnh tranh với các nền tảng lớn như Netflix hay Disney sẽ rất thấp.

Kết luận

* Dù có một số nhược điểm, nhưng với hướng đi đúng đắn, ứng dụng xem phim trên Android vẫn có thể trở thành một nền tảng giải trí hấp dẫn. Nhà phát triển cần tập trung vào tối ưu hiệu suất, đảm bảo bản quyền nội dung và cải thiện trải nghiệm người dùng để nâng cao khả năng cạnh tranh. Ngoài ra, nếu có thể bổ sung các tính năng độc đáo như xem phim offline, gợi ý thông minh hoặc hỗ trợ đa nền tảng, ứng dụng sẽ thu hút được nhiều người dùng hơn và tạo ra giá trị lâu dài trên thị trường.

# **DANH MỤC TÀI LIỆU THAM KHẢO**

1. Trần Duy Thanh (2020), Lập trình di động với Android Studio, NXB Bách Khoa Hà Nội.
2. Nguyễn Văn Tuấn (2019), Phát triển ứng dụng di động sử dụng Firebase, NXB Khoa học và Kỹ thuật.
3. Firebase Team (2024), Firebase Authentication & Firestore Guide, Google Developers. Truy cập tại: <https://firebase.google.com/docs>