



Final Project Documentation

Safe-Kids Movies App



A Safe & Fun Kids Cartoon Application
Under the supervision of Digital Egyptian Pioneers Initiative

Team Members:

Basmala Nasser Mohamed

Basant Mohamed Shaban

Dareen Hasan Ali

Roaa Abdelghany

Salsabil Ahmed Hussien

Contents

1. Project Overview	II
1.1 Project Name.....	II
1.2 Overview.....	II
1.3 Objectives	II
1.4 Project Scope	III
1.5 Technologies Used	III
2. TMDB API Integration.....	IV
3. Stakeholder Analysis & Team Structure	V
3.1 Project Team Roles and Responsibilities	V
3.2 External Stakeholders	VI
4. Database / API Structure.....	VII
4.1 TMDB API (External Data Source).....	VII
4.2 Local Database – Room (Favorites Feature).....	VIII
5. System Architecture	II
5.1 Presentation Layer	IX
5.2 Business Logic Layer.....	II
5.3 Data Layer	II
5.4 Local Storage Layer	X
6. UI/UX Design (Figma)	X
7. Project Timeline (Summary).....	XVIII
8. Testing	XVIII
9. Conclusion	XVIII
10. Future Enhancements	XVIII
11. Links	XVIII

1. Project Overview

1.1 Project Name

Safe-Kids – Kids Movies Application



1.2 Overview

The Safe-Kids App is an Android application designed to provide children with a safe, controlled, and enjoyable environment to watch kids' animated movies.

The app uses the TMDB API to fetch a curated list of family-friendly animated content, ensuring high-quality images, updated descriptions, and secure access to movie data.

The app features a colorful, simple, and child-friendly UI, designed especially for young users. Development was completed using Android Studio



1.3 Objectives

The main objectives of the Safe-Kids App are:

- Providing a trusted collection of children-friendly movies and cartoons.**
 - Designing an interface that is fun, colorful, and easy to use for young children.**
 - Fetch movie data dynamically using TMDB API in a secure way**
 - Ensuring safety by preventing irrelevant or inappropriate content.**
 - Delivering a smooth and secure viewing experience.**
 - Provide users with the ability to save their favorite movies in a dedicated Favorites list.**
 - Improve user experience by storing favorites locally for fast access even without an internet connection.**
 - Offer a personalized and interactive experience by allowing users to manage their favorite animated movies.**
- 

1.4 Project Scope

The project covers:

- **Mobile Development:** Native Android app (**Kotlin / Jetpack Compose**).
 - **API Integration:** Connecting to TMDB API to fetch animated movies.
 - **UI/UX Design:** Creating easy, colorful screens for kids.
 - **Content Filtering:** Displaying only kids-appropriate animation content.
 - **Added a complete Favorites system,** including local storage with Room, an interactive heart button in the Movie Details screen, and a dedicated Favorites Screen to display all saved movies.
 - **Testing:** Ensuring quality, performance, and child safety
- 

1.5 Technologies Used

- **Frontend:** Android Studio (**Kotlin**).
 - **API:** TMDB API (**The Movie Database**).
 - **State Management:** ViewModel with StateFlow.
 - **Local Storage:** Room Database.
 - **Reactive Streams:** Kotlin Flows.
 - **Architecture:** MVVM with Local Persistence.
 - **Design:** Figma.
 - **Version Control:** Git & GitHub.
 - **Testing:** Unit Testing, Manual Testing
- 

2. TMDB API Integration

The Safe-Kids App integrates the TMDB API as the main data source for animated movies.

Why TMDB?

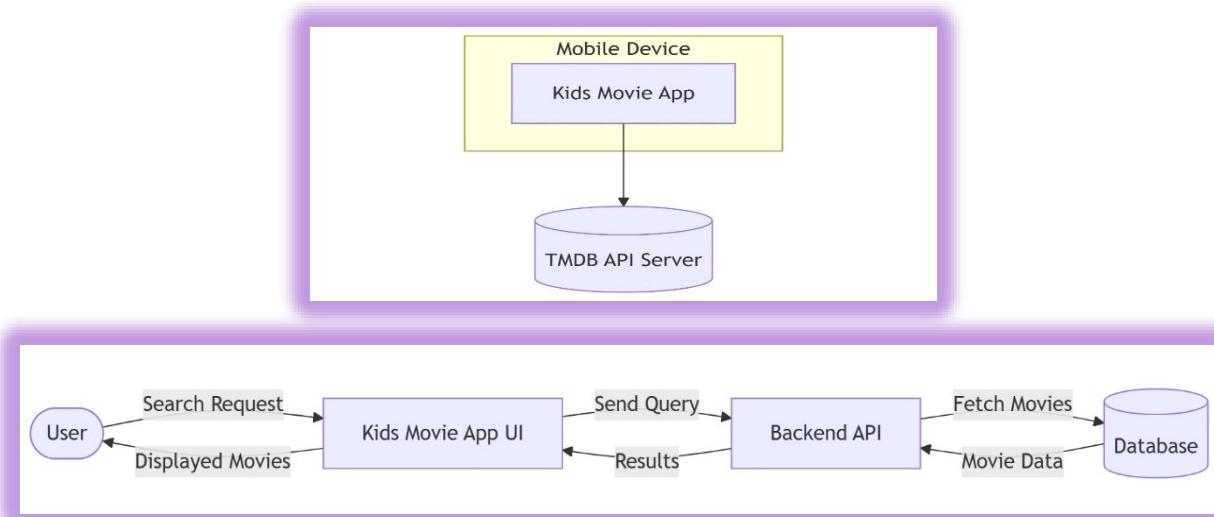
- Provides high-quality posters and images
- Offers accurate and updated movie descriptions
- Supports filtering by Animation / Family genres
- Reliable, scalable, and secure movie database

What do we fetch?

- Movie title
- Description (overview)
- Poster image
- ID for details
- Trailer key

Implementations

- Used TMDB's `/discover/movie` endpoint
- Filtered movies by Animation genre (`genre_id = 16`)
- Displayed results dynamically on the Home Screen
- Passed movie details to the Movie Details screen
- Ensured only kids-appropriate movies appear in the app



3. Stakeholder Analysis & Team Structure

3.1 Project Team Roles and Responsibilities

Team Member	Role	Responsibilities
Basmala Nasser	Team Leader	Managing the team workflow, distributing tasks, ensuring deadlines are met, and maintaining communication between members and supervisors.
Dareen Hasan Ali Roaa Abdelghany Salsabil Ahmed Hussien Basant Mohamed Basmala Nasser	UI Developers	Translating Figma designs into functional application screens and implementing front-end business logic.
Basmala Nasser Basant Mohamed	API Integration Developer/Local database	Responsible for integrating external API services, handling data requests and responses, ensuring secure communication with the server, and managing dynamic data operations within the application
Salsabil Ahmed Hussien	UI/UX Designer	Designing the mobile interface on Figma and ensuring user friendly experience.
Basmala Nasser	Tester	Testing app functionality, finding bugs.

3.2 External Stakeholders

Name / Type	Role in the Project
Digital Egypt Pioneers Initiative	Project supervision and evaluation
Academic Supervisor	Responsible for providing technical guidance and continuous feedback throughout the project.
End Users	Using the Safe-Kids Movies App and providing feedback about its features and performance.

4.Database / API Structure

The application now relies on two main data layers:

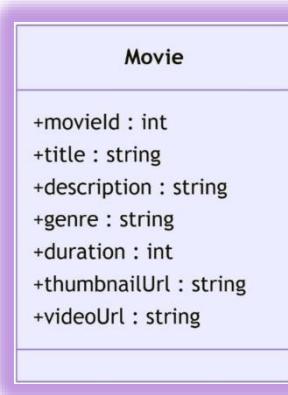
4.1 TMDB API (External Data Source)

Still used to fetch:

- Movie title
- Overview
- Poster image
- Rating
- Release date
- Genre IDs
- Trailer URL

Movie Data Model:

Description	Type	Field
Movie ID from TMDB	Int	id
Movie title	String	title
Description	String	overview
Poster URL	String	poster_path
Rating	Float	vote_average
Release date	String	release_date
Ensures it's animation	List<Int>	genre_ids



Content Safety

- Only movies under Animation genre are displayed.
- No user-generated content.
- No unsafe categories fetched from TMDB.

4.2 Local Database – Room (Favorites Feature)

A local Room database has been added to support offline saving of favorite movies.

FavoriteMovie Entity

Description	Type	Field
Movie ID	Int	id
Movie title	String	title
Poster URL	String	poster_path
Rating	Double	vote_average

DAO – MovieDao Functions

- `insertMovie(movie)` → Saves a movie to the favorites list
- `deleteMovie(movie)` → Removes a movie from favorites
- `getAllFavorites()` → Returns a Flow stream of all favorite movies
- `isFavorite(movieId)` → Checks whether a specific movie is already saved

Purpose of the Local Database

- Provides fast access to favorite movies
- Enables offline support
- Allows real-time updates via Kotlin Flow
- Supports the new Favorites Screen



5. System Architecture

5.1 Presentation Layer

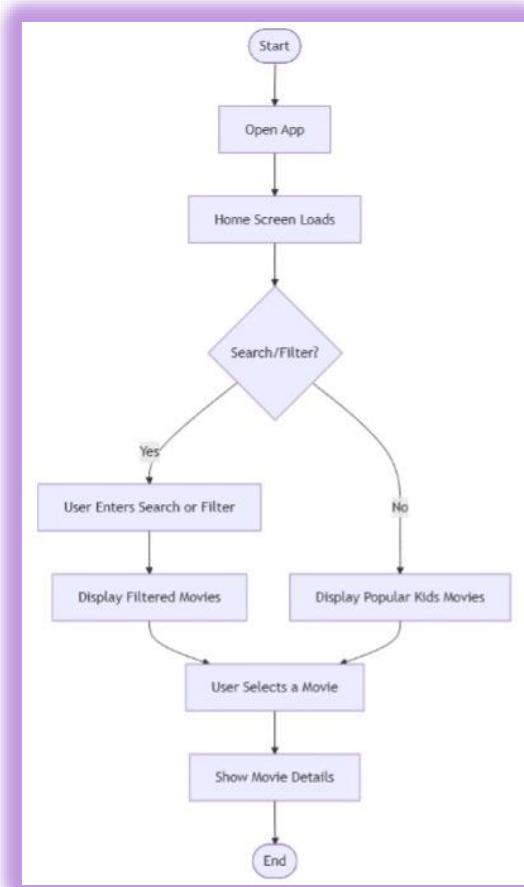
- Splash Screen
- Home Screen (dynamic movies list)
- Search Screen
- Movie Details Screen

5.2 Business Logic Layer

- ViewModel for managing UI state
- API handling (Retrofit)
- Search logic
- Data filtering (animation only)

5.3 Data Layer

- TMDB API endpoints
- JSON parsing
- Network handling



5.4 Local Storage Layer (New)

- The Local Storage Layer manages all favorite-movie operations using:
 - Room Database
 - FavoriteMovie Entity
 - MovieDao
 - AppDatabase instance
 - Integration with MovieViewModel using Kotlin Flow and StateFlow
 - This ensures a reactive and always-updated Favorites list.
- 

6.UI/UX Design (Figma)

Design Goals:

- Bright, child-friendly visuals
- Large clickable areas
- Clear and simple navigation
- Cartoon-inspired theme

Figma was used to design:

- Splash screen
 - Home movie list
 - Search UI
 - Movie details page
 - Favorites Screen
- 

Figma design screens



Figure 1: Splash screen

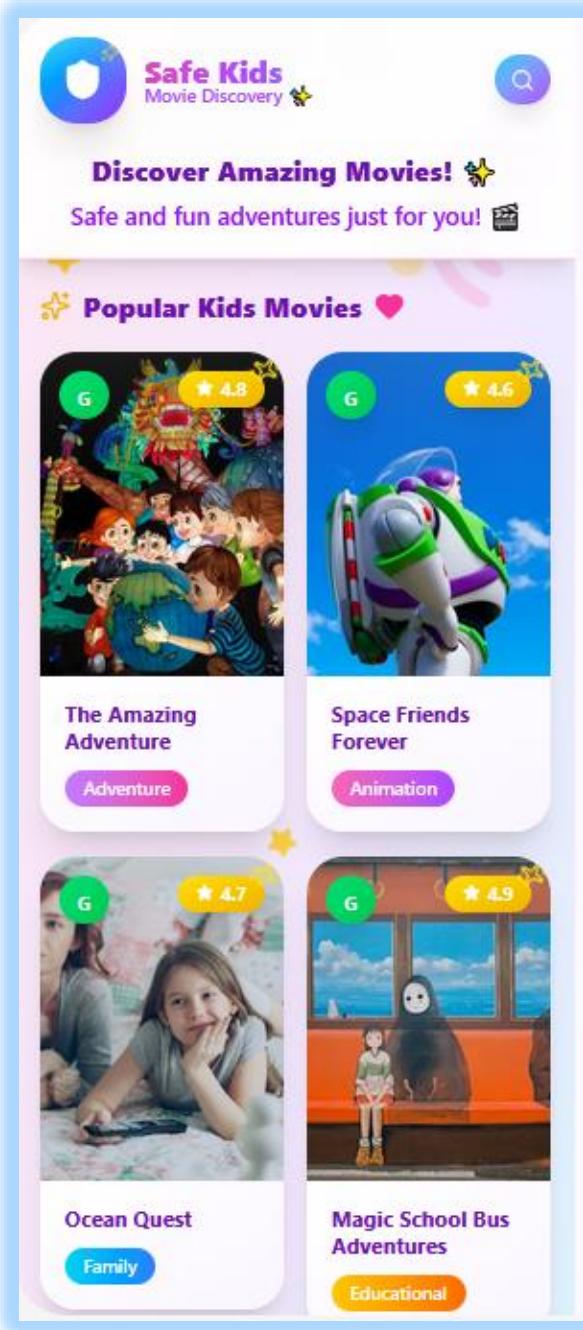


Figure 1: Home Screen

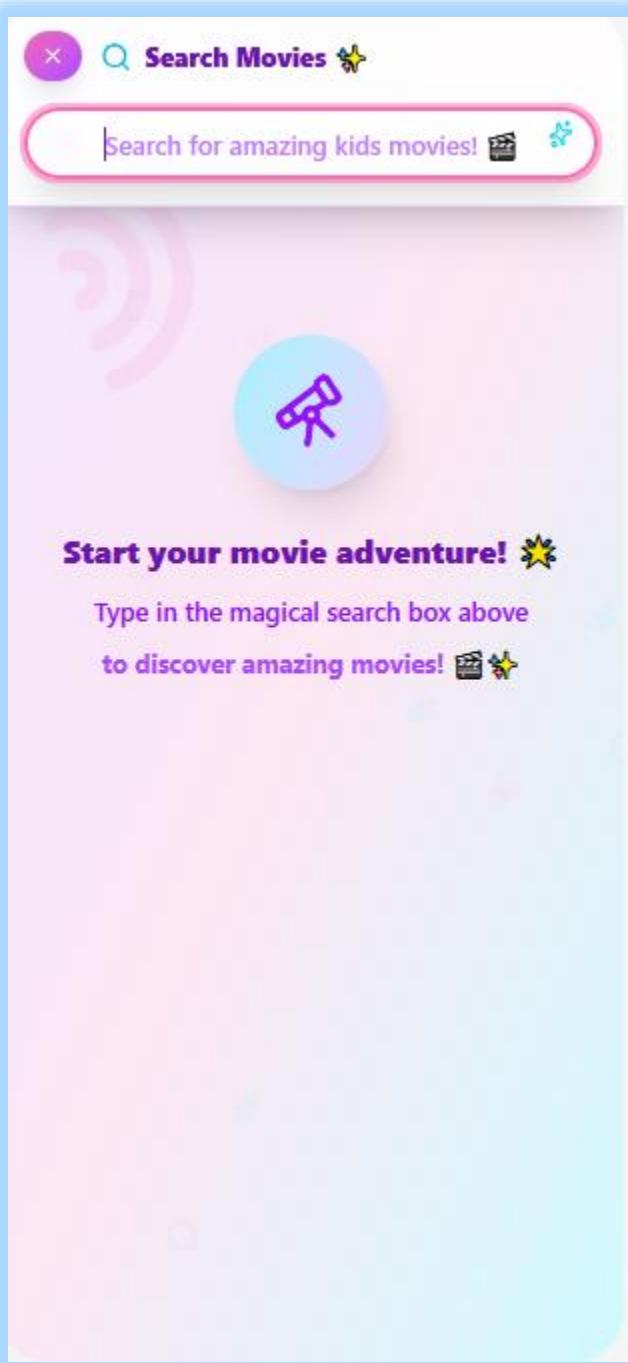


Figure 3: Empty Search Screen

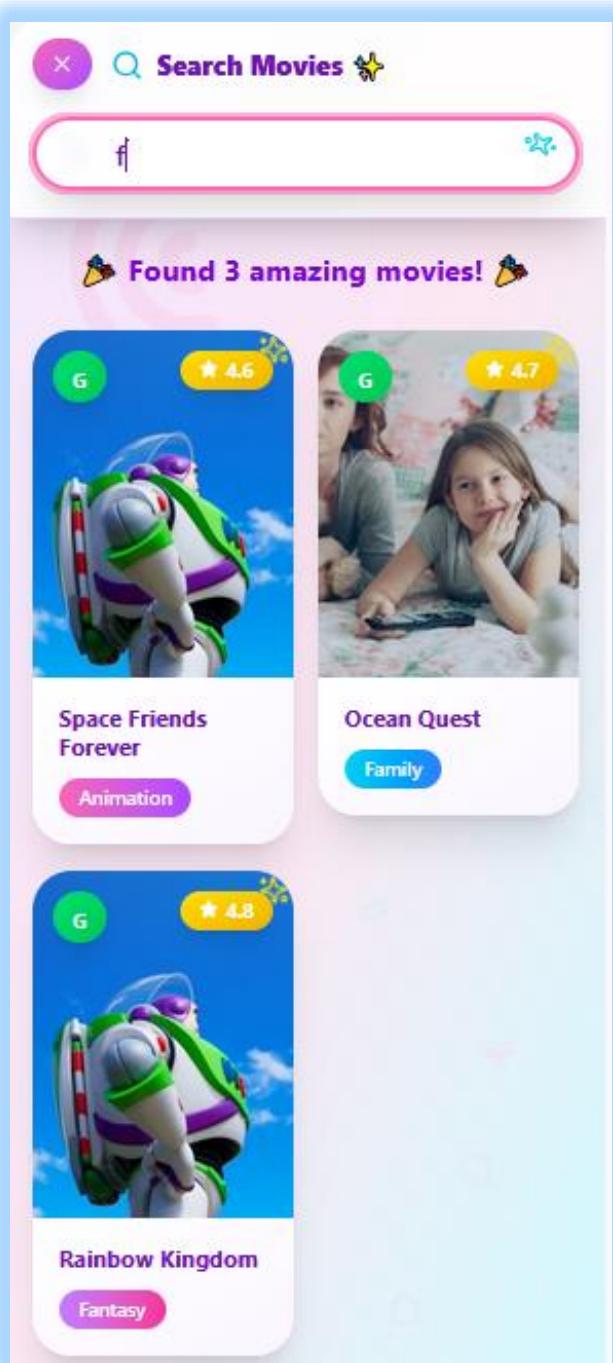


Figure 4: Search Screen after search



Figure 5: Detail Screen

Our App Screens

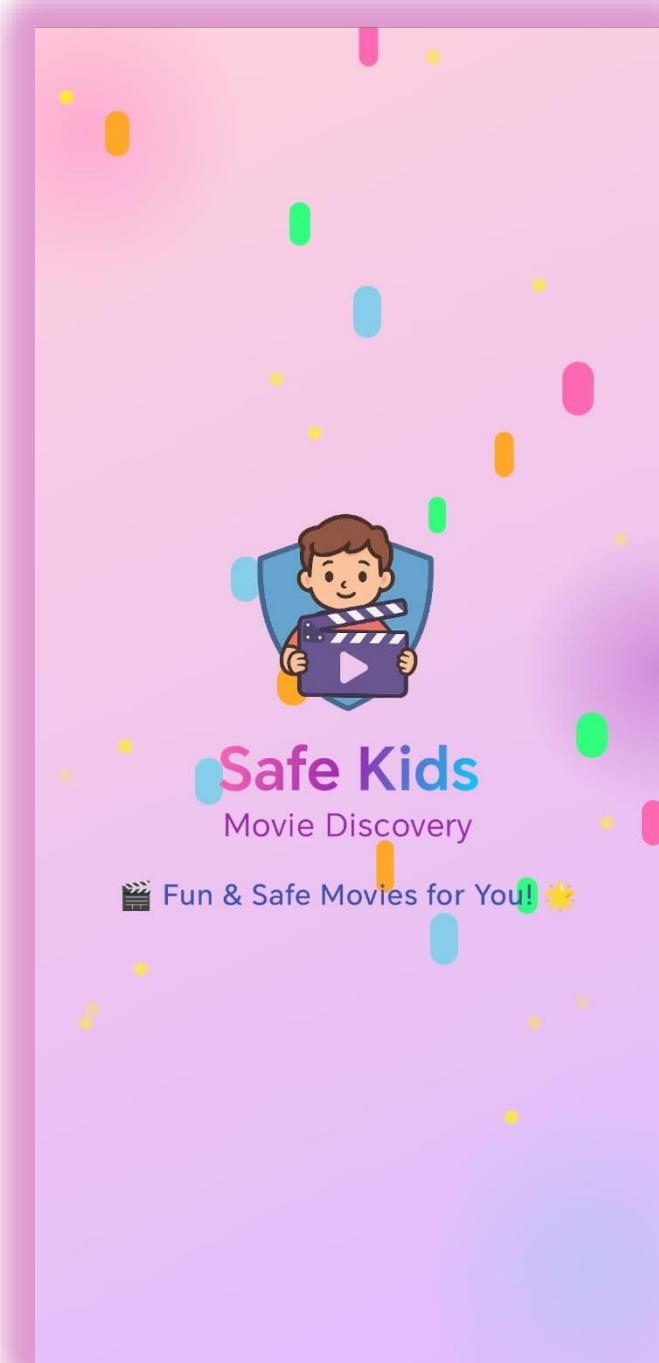


Figure 1: Splash Screen

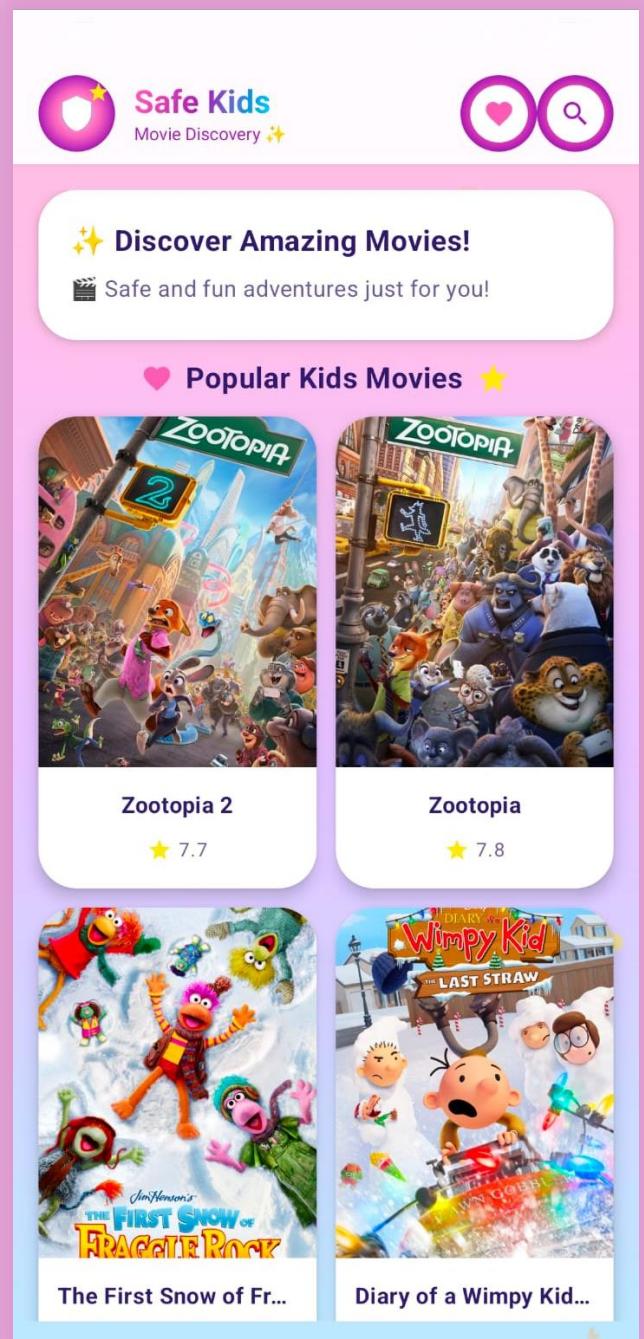


Figure 2: Home Screen

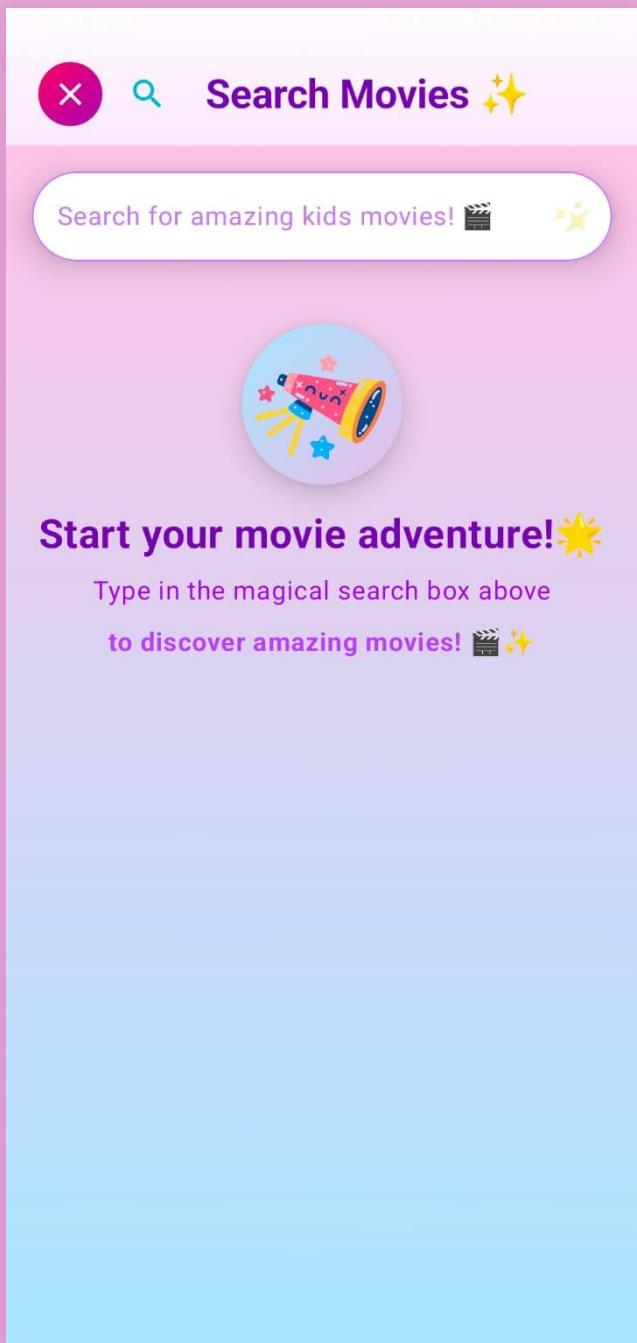


Figure 3: Empty Search Screen

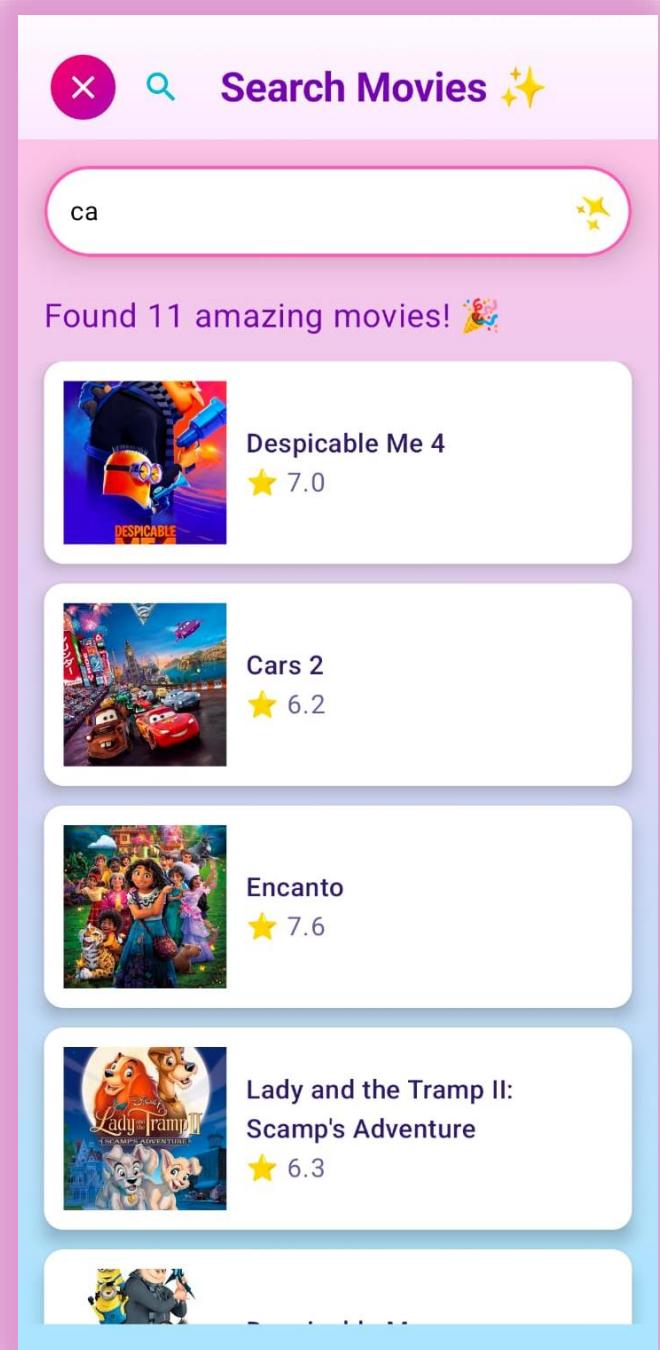


Figure 4: Search Screen after search



Figure 5: Detail Screen

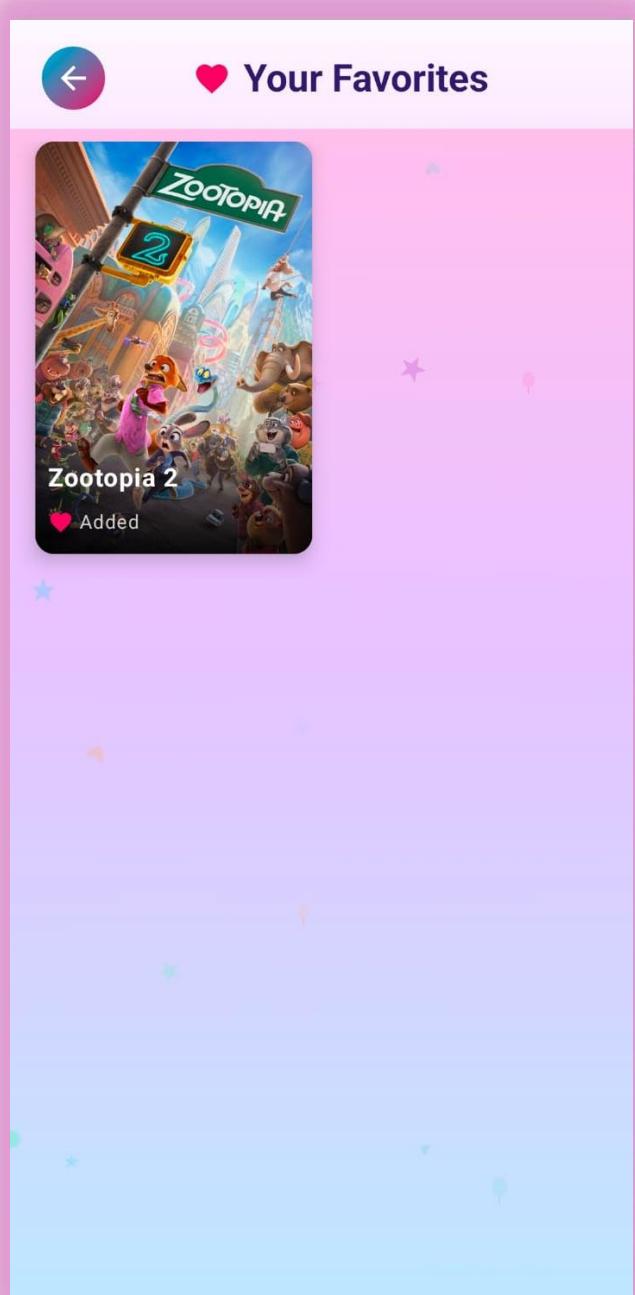


Figure 6: Favorites Screen

7. Project Timeline (Summary)

1. Week 1 – Setup & UI

- Initialize Android project
- Create GitHub repo
- Build wireframes
- Implement static screens

2. Week 2 – API Integration

- Connect TMDB API
- Fetch animated movies
- Display movies dynamically

3. Week 3 – Navigation & Features

- Implement navigation
- Build movie detail screen
- Link search → detail
- Add trailer button

4. Week 4 – Final Polish

- Add search improvements
- UI/UX refinements
- Loading & error states
- Implementation of Room Database
- Creation of FavoriteMovie entity
- Setup of MovieDao and AppDatabase
- Adding toggle favorite button on Movie Details Screen
- Designing and building the Favorites Screen
- Documentation



8. Testing

Testing Types:

- UI testing(manual)
- View model testing(automated)
- Navigation testing(automated)
- Search functionality testing(manual)



9. Conclusion

The Safe-Kids App successfully delivers a safe and enjoyable animated movie experience for young children.

By integrating TMDB API, the app provides dynamic, up-to-date, and family-friendly movies through a fun and visually engaging interface.



10. Future Enhancements

- Parental control dashboard
- Screen-time management
- Multi-language support



11. Links

<https://www.figma.com/community/file/1554038440572666820>

<https://github.com/slsbyl/SafeKids/blob/main/README.md>