

MOVIE DISCOVERY APP

PROJECT OVERVIEW

TEAM MEMBERS & RESPONSIBILITIES

-**Farah Hamza Ghadady:** Network Layer of home screen

-**Shahd Khaled elsayed:** UI of home screen using jetpack compose

-**Yasmin Wael Mostafa:** ViewModel , search bar of search screen

-**Mohamed waleed :** Network of search screen : remote data source ,repository & UI of search screen using jetpack compose :UI search state , movie result list

- **Mohamed Ramadan :** Network Layer, ViewModel and uiState of details screen

- **Mustafa Bahaa :** repo & UI of details screen using jetpack compose

PROJECT CONCEPT & OBJECTIVES

The Idea:

An **Android movie discovery application** that enables users to browse popular movies, search for specific titles, and view detailed information about each film. The app provides a seamless experience for movie enthusiasts to explore and learn about movies using real-time data from TMDB API.

Project Goals:

Create a functional, user-friendly movie browsing application

Implement modern Android development practices and architecture

Integrate real-world API to fetch live movie data

Design an intuitive interface with Jetpack Compose

Develop essential features: browsing, searching, and detailed views

KEY FEATURES

1. Home Screen - Popular Movies

- What it does:

Displays a grid of popular/trending movies
Shows movie posters, titles, ratings, and release dates
Provides an attractive entry point to the app

- Implementation Highlights:

Clean red theme with white backgrounds
Star ratings displayed prominently
Grid layout for optimal viewing
Tap on any movie to see full details
Bottom navigation for easy screen switching

- Technical Details:

HomeViewModel manages movie data
HomeRepo fetches data from TMDB API
Efficient list rendering with LazyVerticalGrid
Loading states handled gracefully

2. Search Screen - Find Any Movie

- What it does:

Search bar for users to find specific movies
Real-time search results as users type
Displays matching movies with posters and ratings
Shows "Spider" search results in the demo

- Implementation Highlights:

Clean search interface with search icon
Grid display of search results
Consistent design with Home screen
Efficient search filtering

- Technical Details:

SearchViewModel processes search queries
SearchRepo handles API search requests
Debounced search to optimize API calls
Empty state handling for no results

3. Details Screen - Complete Movie Information

- What it does:

Shows full movie poster/backdrop

Displays movie title and rating

Presents complete movie overview/synopsis

Shows release date and other metadata

- Implementation Highlights:

Large, immersive poster display (as seen in Zootopia 2 example)

Clear typography for easy reading

Comprehensive movie description

Professional layout design

- Technical Details:

DetailsViewModel manages selected movie data

DetailsRepo fetches additional details if needed

Receives movie data via navigation arguments

Scrollable content for long descriptions

4. Navigation System

Bottom navigation bar with Home and Search tabs

Smooth transitions between screens

Back navigation support

Deep linking to specific movies

DESIGN & USER EXPERIENCE

- **Visual Design:**

Color Theme: Bold red primary color (#D32F2F or similar)

Layout: Clean, modern Material Design principles

Typography: Clear, readable fonts with proper hierarchy

Icons: Bottom navigation icons for Home and Search

- **User Flow:**

User opens app → sees Popular Movies

User can browse movies or tap Search

In Search, user types movie name → sees results

User taps any movie → sees full details

User can navigate back or to other sections

- **Responsive Design:**

Adapts to different screen sizes

Optimized for portrait orientation

Smooth scrolling and interactions

TECHNICAL IMPLEMENTATION

- **Architecture Pattern:** MVVM

UI Layer (Composables)



ViewModel Layer (Business Logic)



Repository Layer (Data Management)



Data Source Layer (API Calls)

- **User interacts with UI (Compose screen):**

UI triggers ViewModel function

ViewModel calls Repository

Repository requests data from

RemoteDataSource

RemoteDataSource makes API call via Retrofit

Data flows back through layers

UI updates with new state

- **API Integration - TMDB**

1. Base URL: api.themoviedb.org

2. Endpoints Used:

/movie/popular - for Home screen

/search/movie - for Search functionality

/movie/{id} - for Details screen

3. Authentication: API key in request headers

4. Response Format: JSON parsed to Kotlin data classes

KEY FILES STRUCTURE

HomeScreen.kt - Popular movies UI

SearchScreen.kt - Search interface

DetailsScreen.kt - Movie details UI

HomeViewModel.kt, SearchViewModel.kt, DetailsViewModel.kt - State management

HomeRepo.kt, SearchRepo.kt, DetailsRepo.kt - Data repositories

ApiClient.kt, ApiService.kt - Network configuration

Models.kt - Data classes for movies

Routes.kt - Navigation definitions

MainActivity.kt - App entry point

FUTURE ENHANCEMENTS

- **Short-term Improvements (Next Phase)**

1. Favorites/Watchlist

- Add ability to save favorite movies
- Create a dedicated Favorites tab
- Persist data locally with Room database

2. Enhanced Search

- Add filters (genre, year, rating)
- Search history
- Trending searches

- **Long-term Vision**

1. Content Expansion

- Add TV shows and series
- Include "Now Playing" movies
- "Upcoming" movies section
- Genre-based browsing

2. Social Features

- User ratings and reviews
- Share movies with friends
- Community discussions

PROJECT SUMMARY

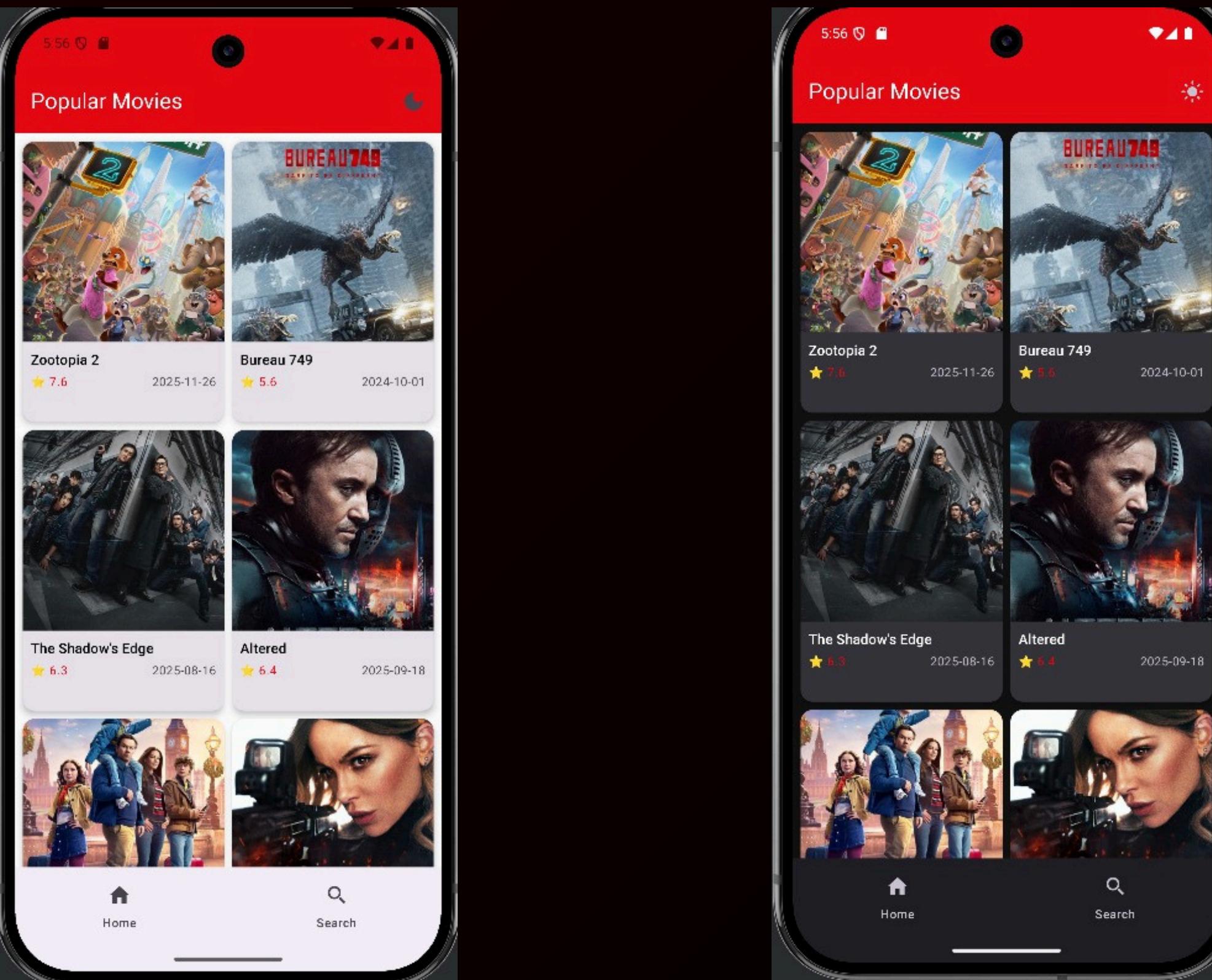
Our project has:

- Clean Architecture: Well-organized, scalable codebase
- Modern UI: Built with Jetpack Compose
- Real Data: Live integration with TMDB API
- Complete Features: Browse, search, and detailed views all working
- Professional Design: Consistent red theme, intuitive navigation

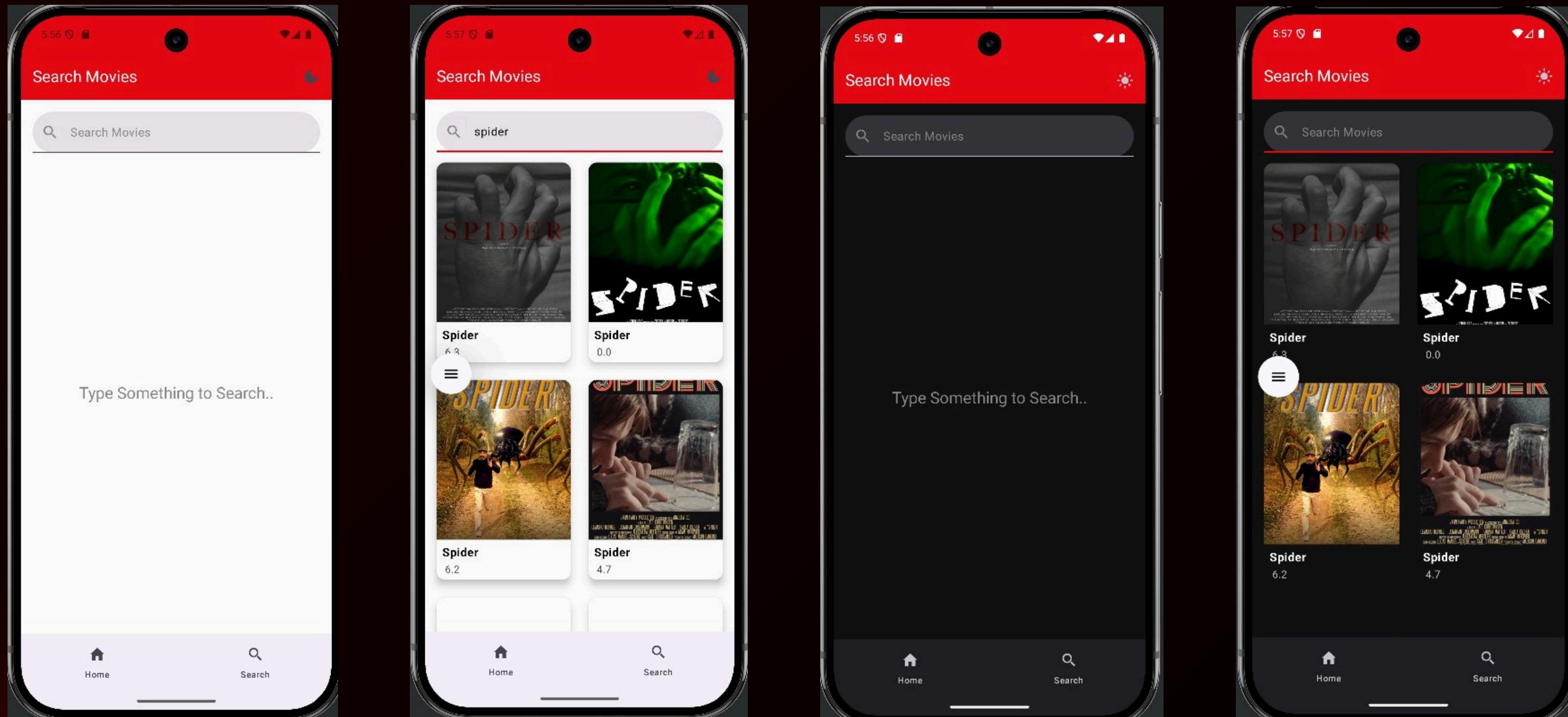
Key Achievements:

- Completed three fully functional screens
- Integrated real-time movie data
- Implemented smooth navigation flow
- Created an intuitive user interface

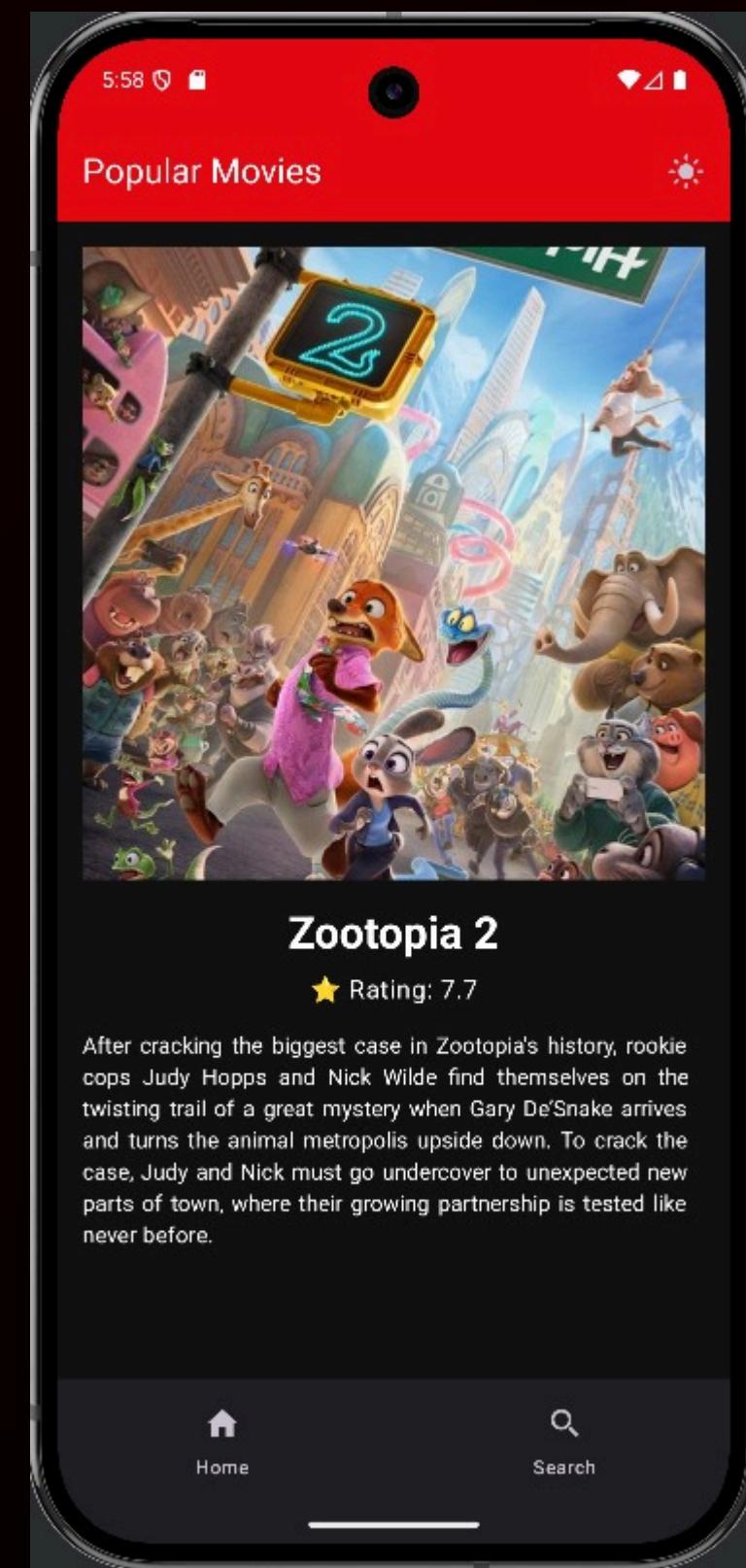
HOME SCREEN



SEARCH SCREEN



DETAILS SCREEN



THANK YOU

Questions & Feedback Welcome
Project Demonstration Available
We appreciate your time and feedback!