**MOVIE APP**

**Real World Problem Identification**

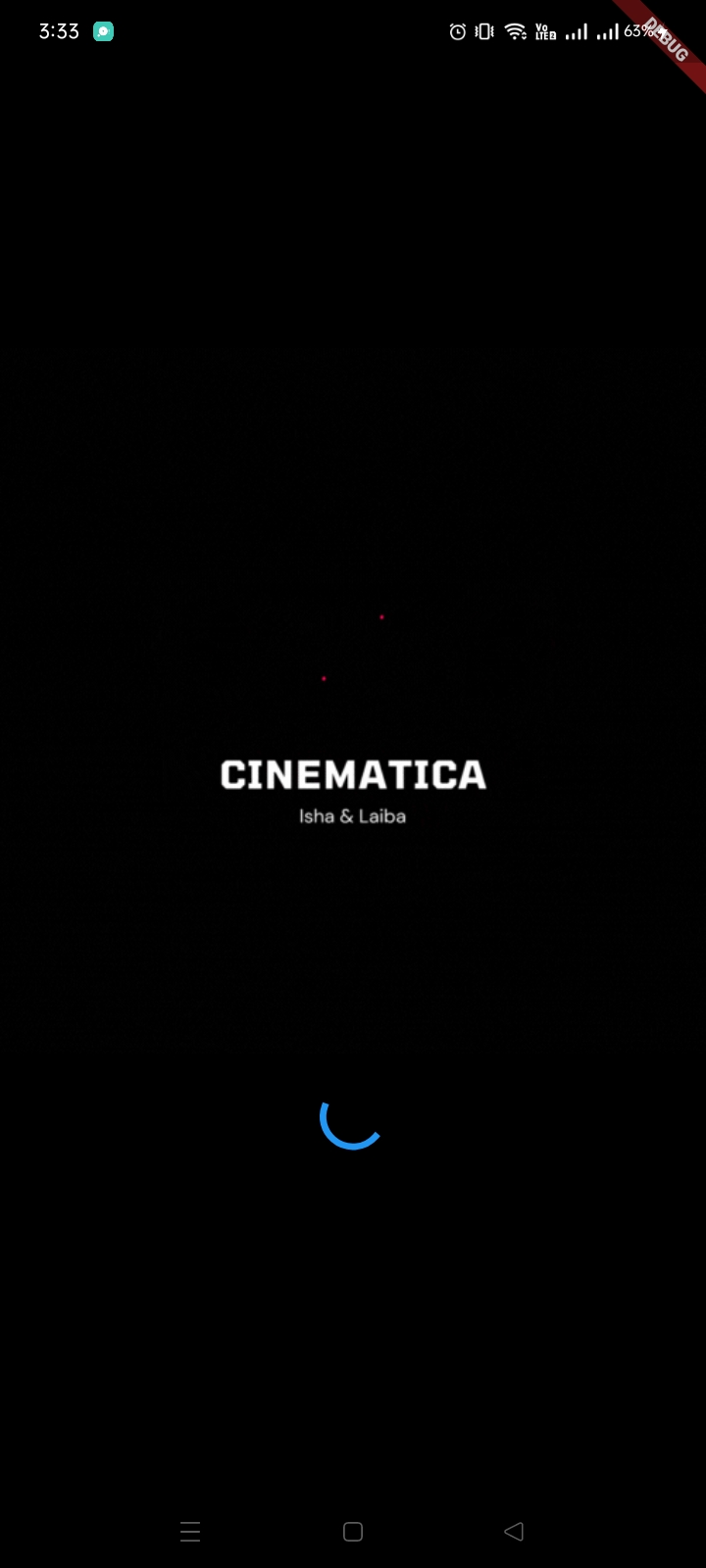
The primary real-world problem we aim to tackle is the inconvenience experienced by movie enthusiasts when trying to access and enjoys movies effortlessly. Currently, users often face the challenge of navigating complex interfaces and encountering overwhelming options while searching for movies that align with their preferences. This creates a need for a simplified movie app that streamlines the movie discovery and viewing process, making it user-friendly and accessible to a wide audience.

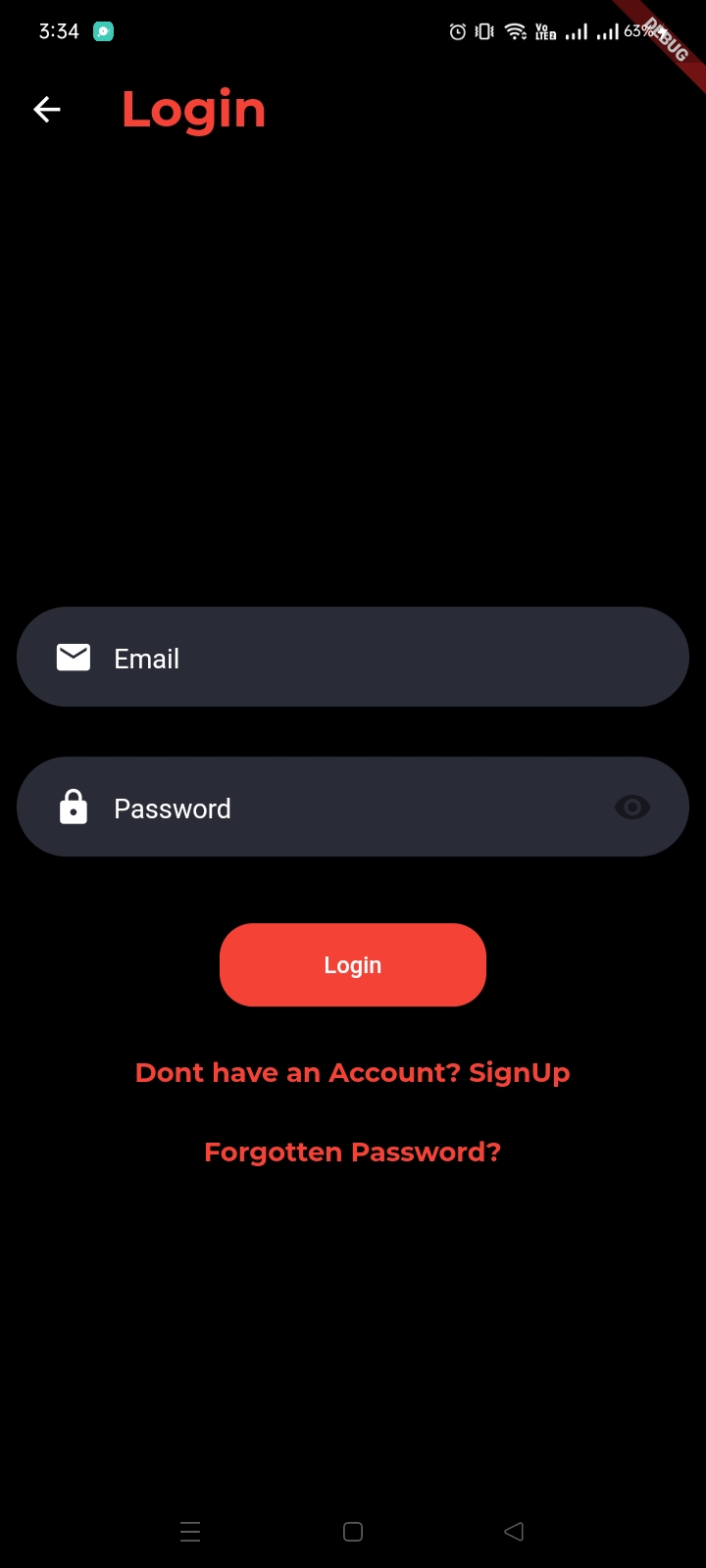
**Proposed Solution:**

Our proposed solution is to design and develop a user-friendly movie app that caters to the needs of movie enthusiasts by simplifying the process of finding and enjoying movies of their choice. This app will focus on providing a seamless and intuitive user experience, making movie discovery and playback straightforward and enjoyable. To achieve this, we will implement the following key features:

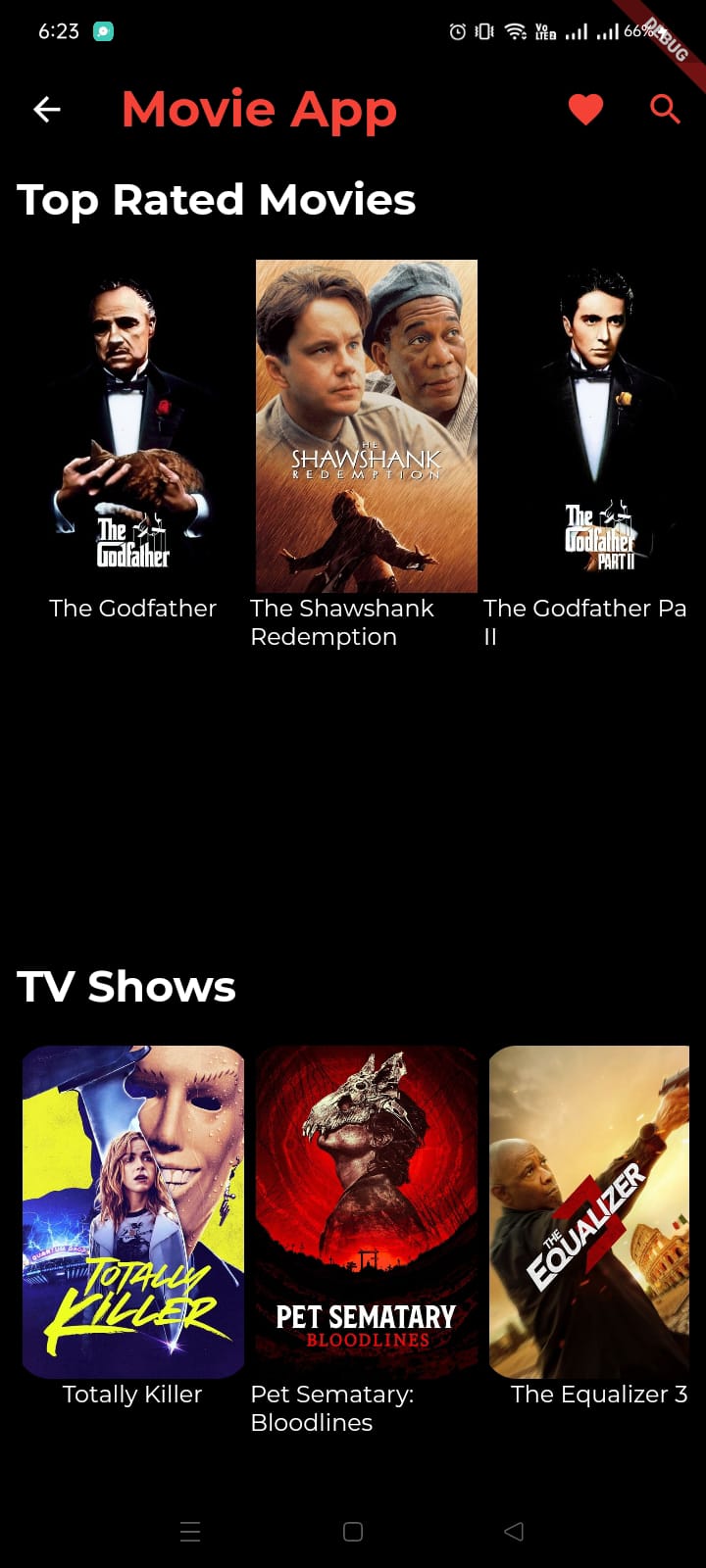
1. **Login and Signup:**
   * Create a user-friendly authentication system with dedicated login and signup screens.
   * Ensure secure user registration and login processes, with data encryption for user data privacy and security.
2. **TMDB API Integration:**
   * Implement TMDB API integration to access a vast database of movies and retrieve essential movie information such as titles, genres, release years, synopses, cast, crew, and more.
   * Ensure seamless integration to provide users with a rich and up-to-date movie catalog.
3. **Search Feature:**
   * Develop a robust search functionality that enables users to easily find movies based on keywords, titles, genres, and other criteria.
   * Provide instant and relevant search results to enhance the user experience.
4. **Favorite Movie Feature:**
   * Allow users to mark movies as favorites or add them to a watch list for easy access later.
   * Create a user-friendly interface for managing and viewing their list of favorite movies.

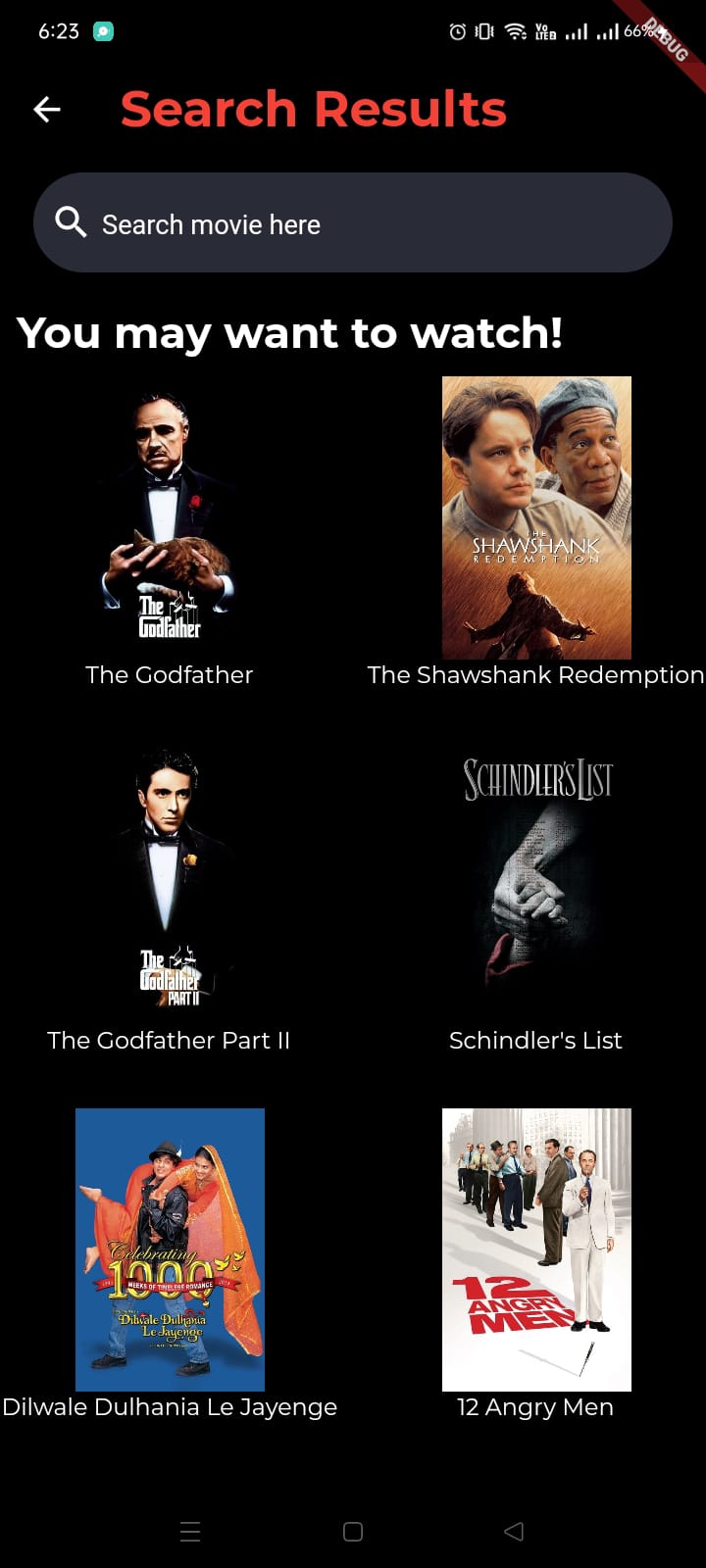
**Responsive User Interfaces (Screenshots of your app on different screens & platforms)**

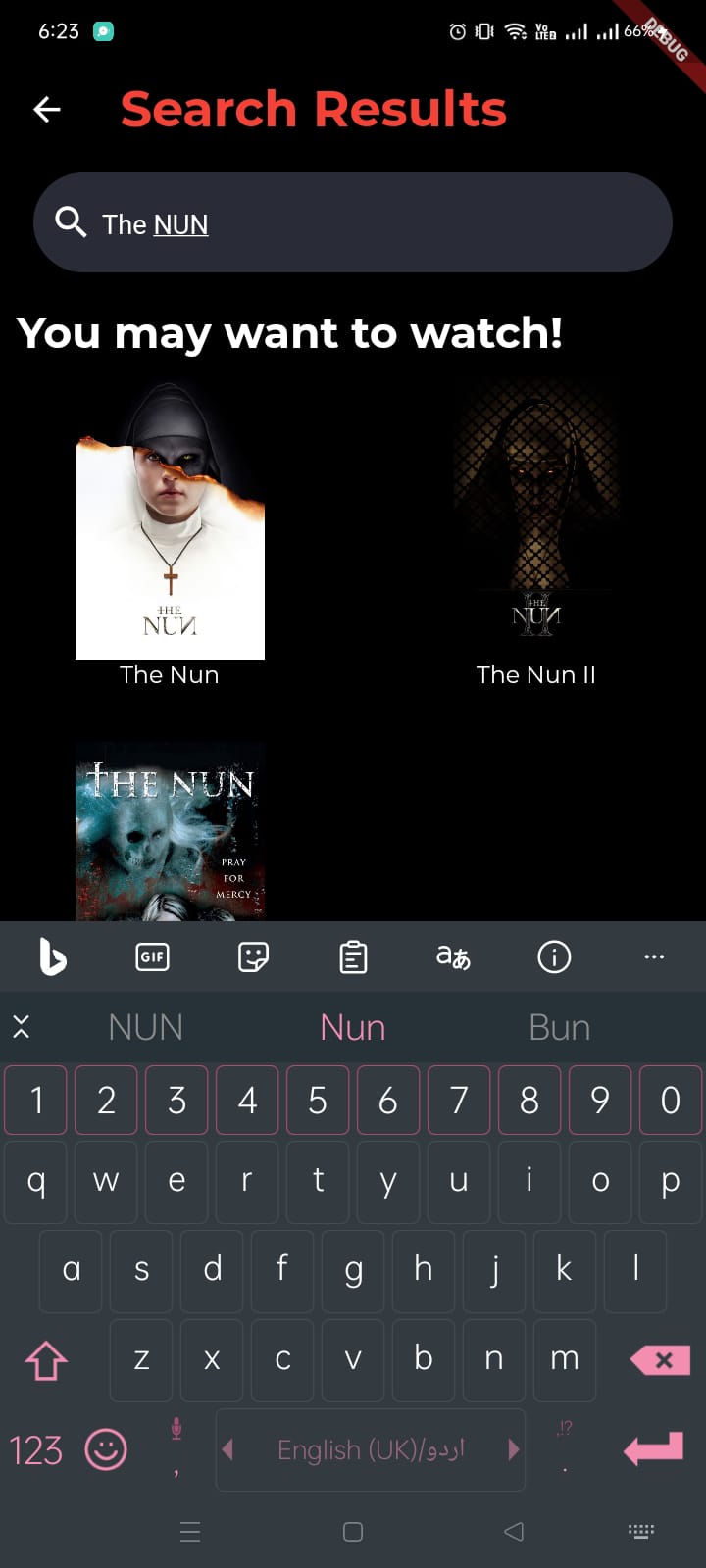
****

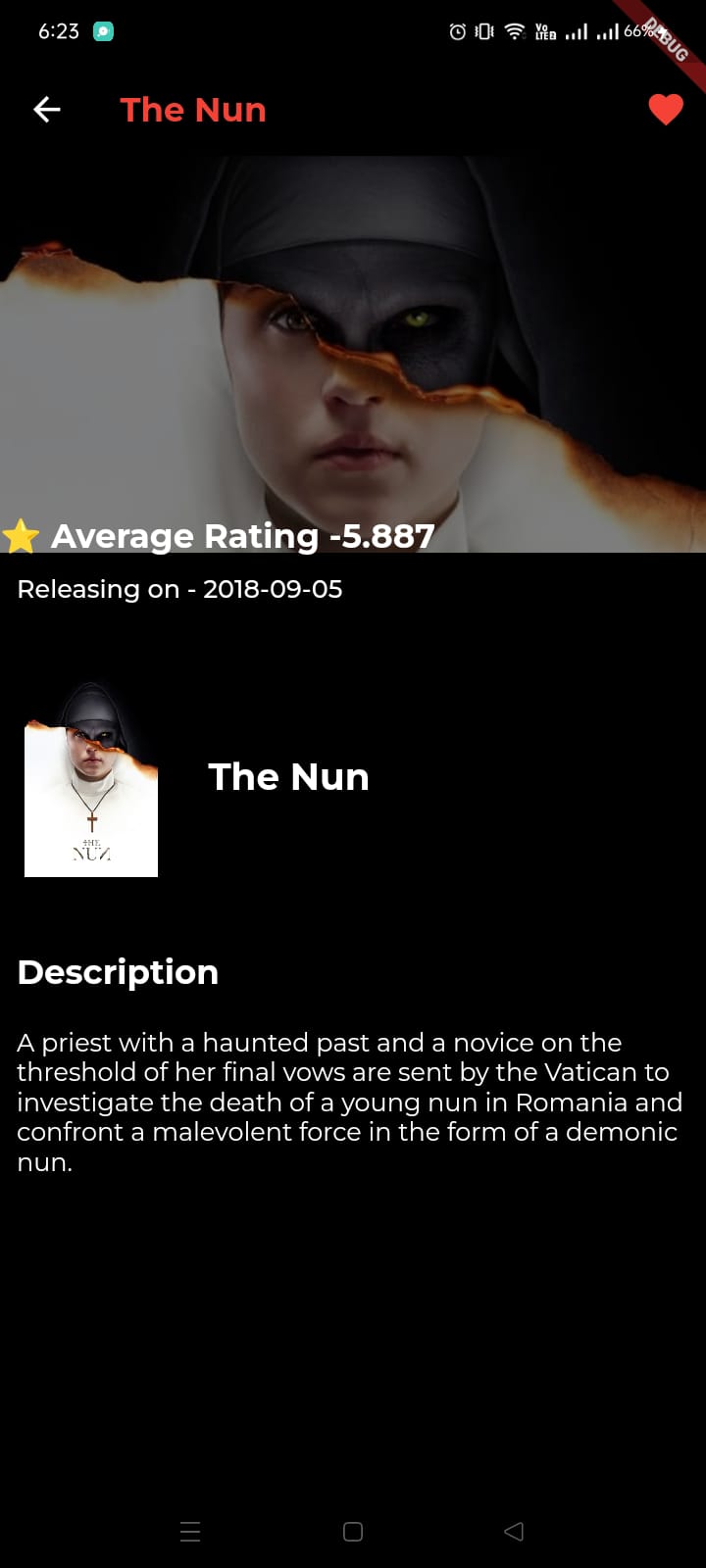
****

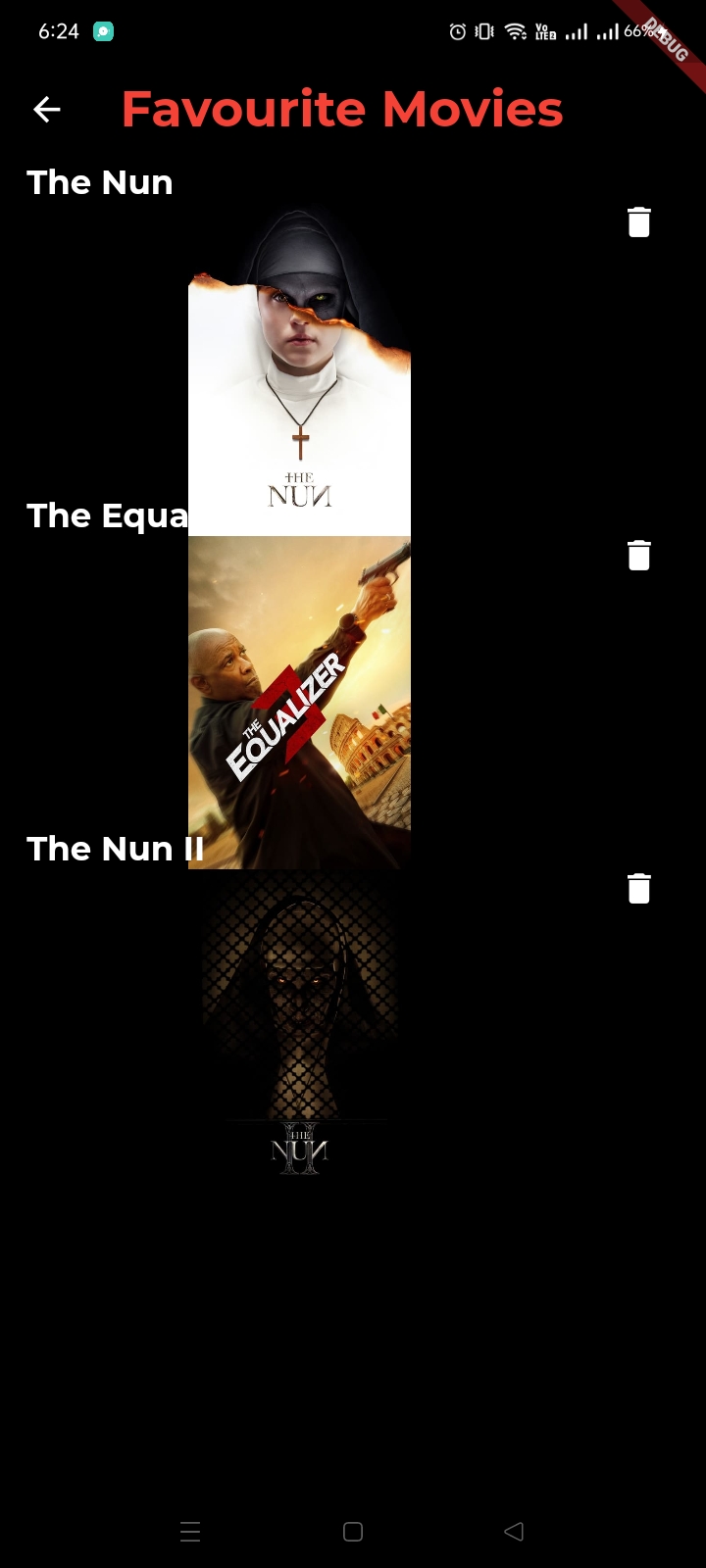
****

****

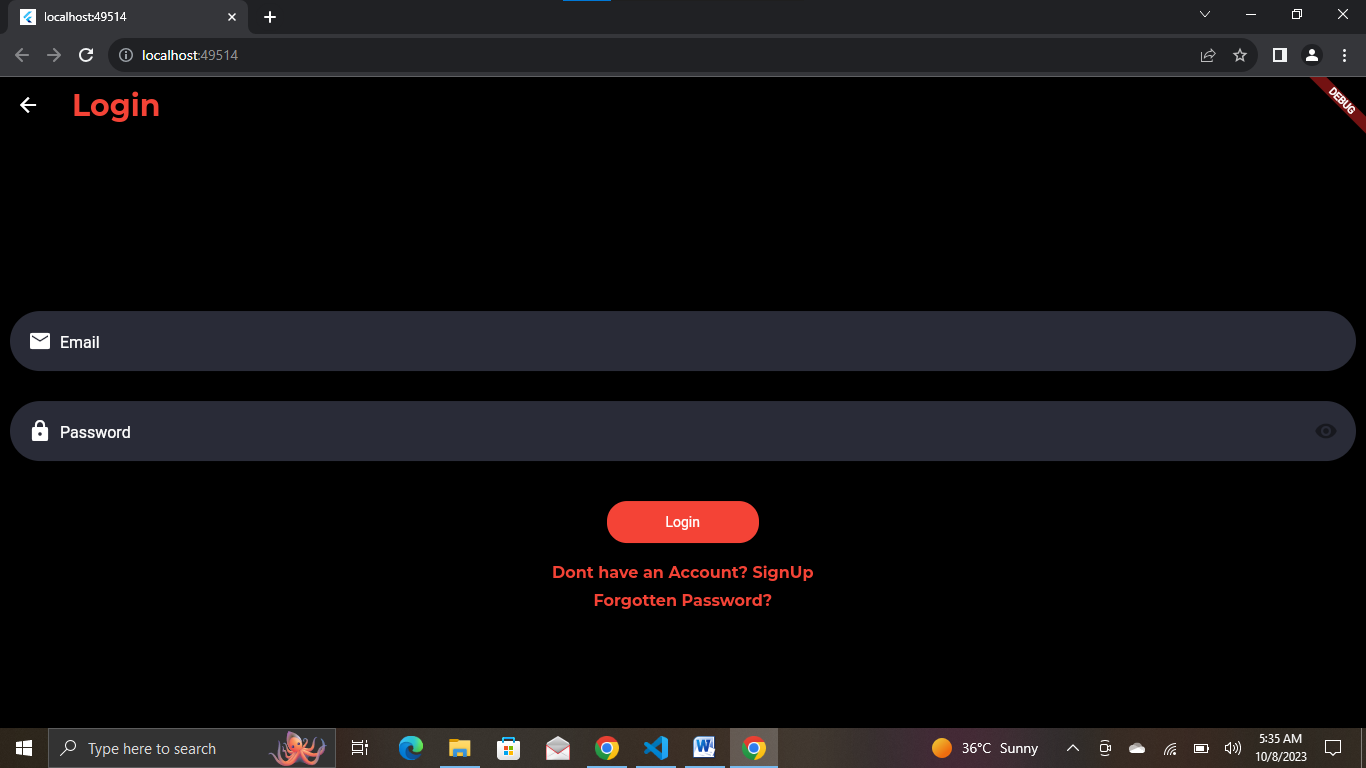
****

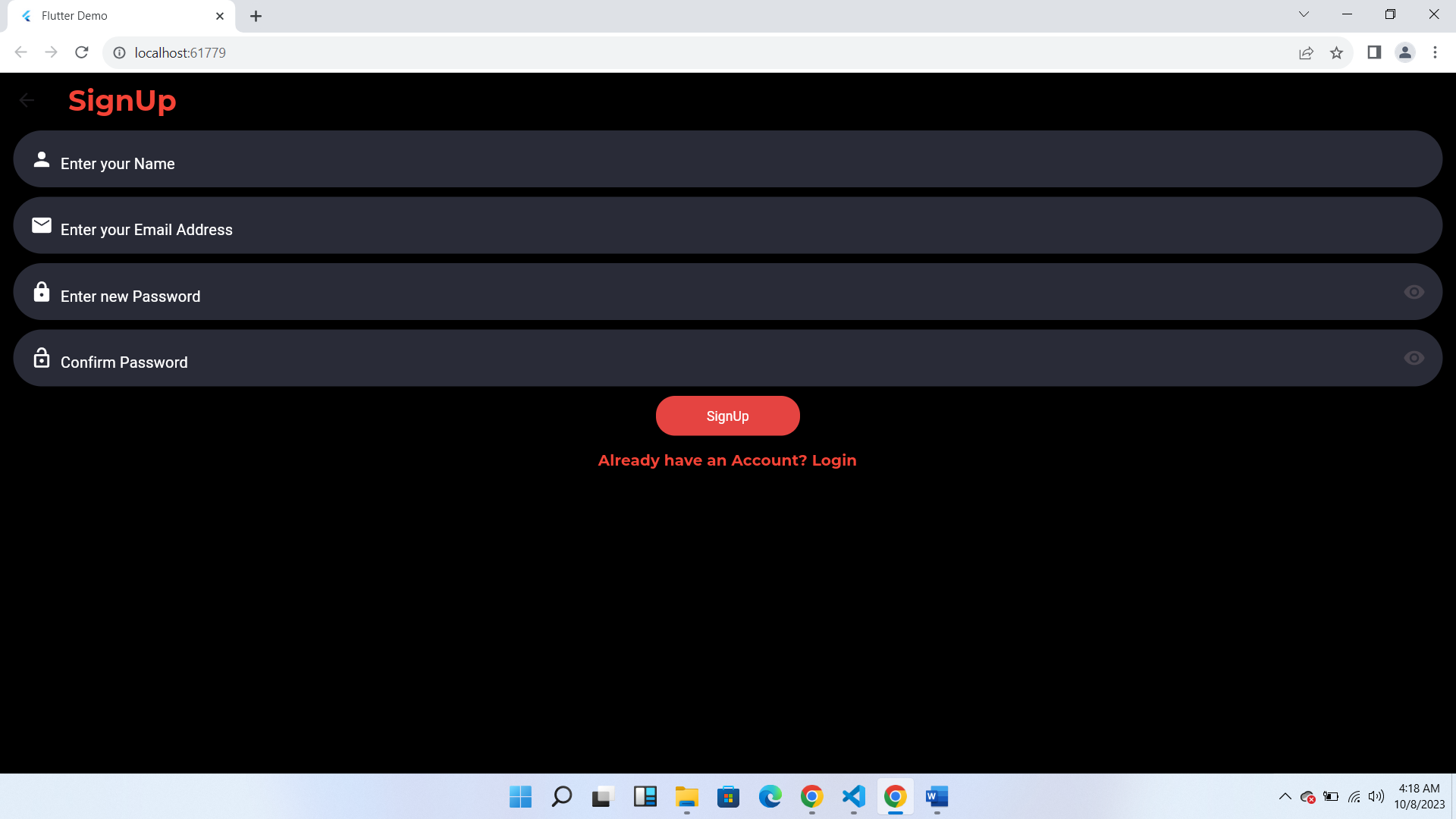
****

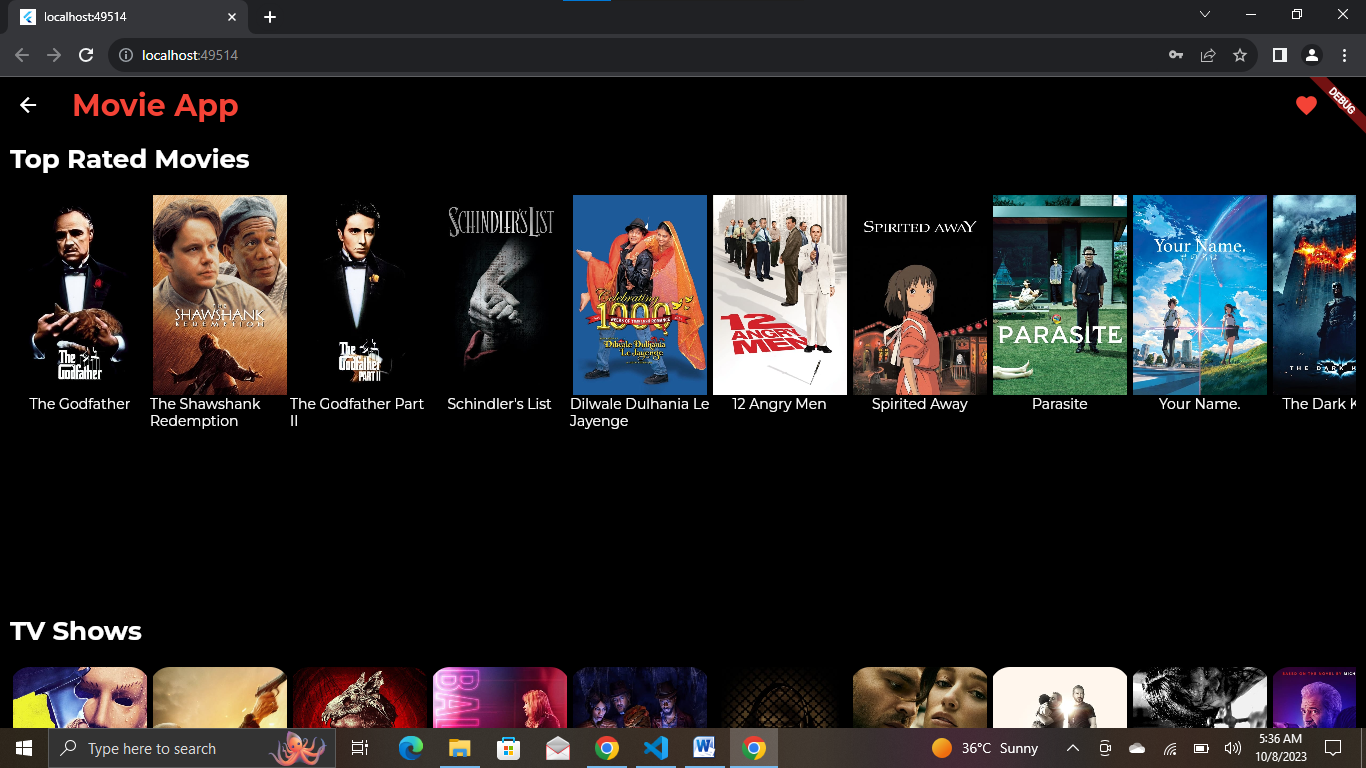
****

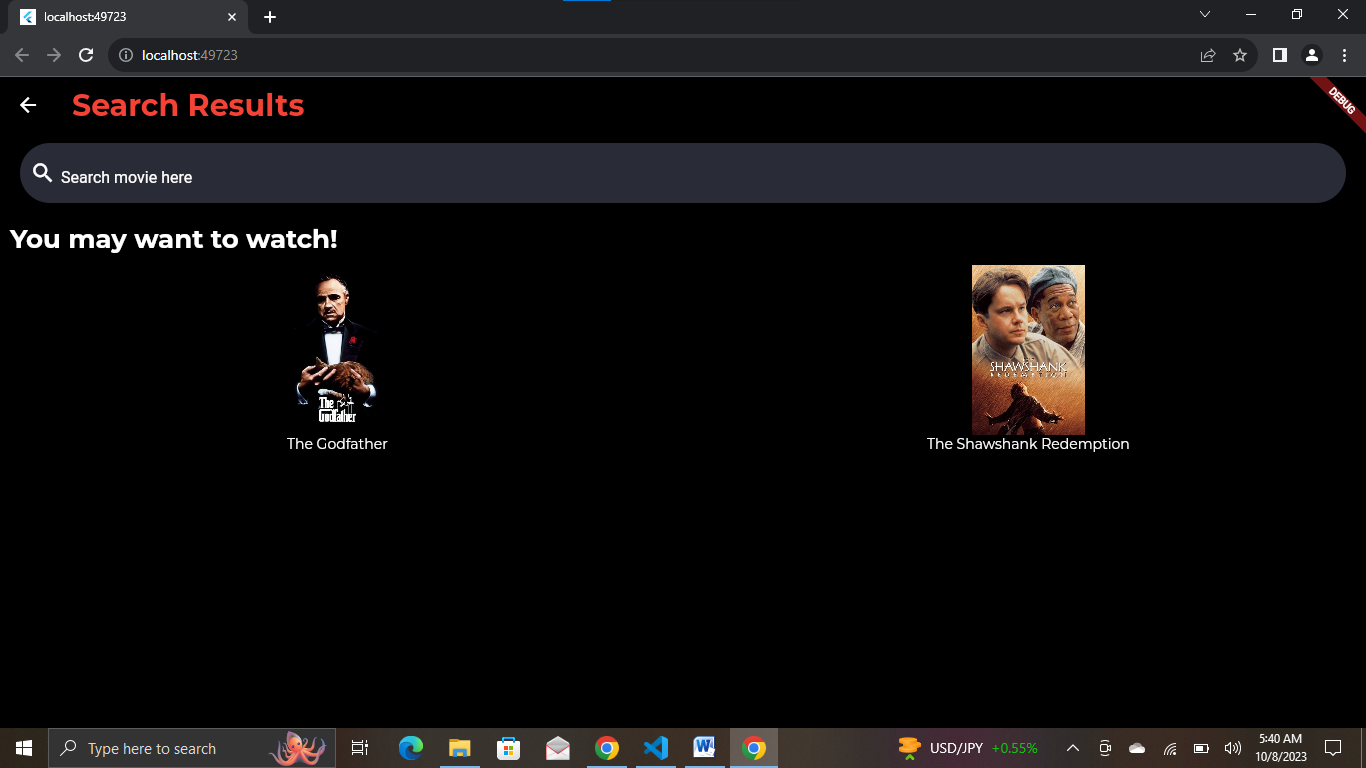
****

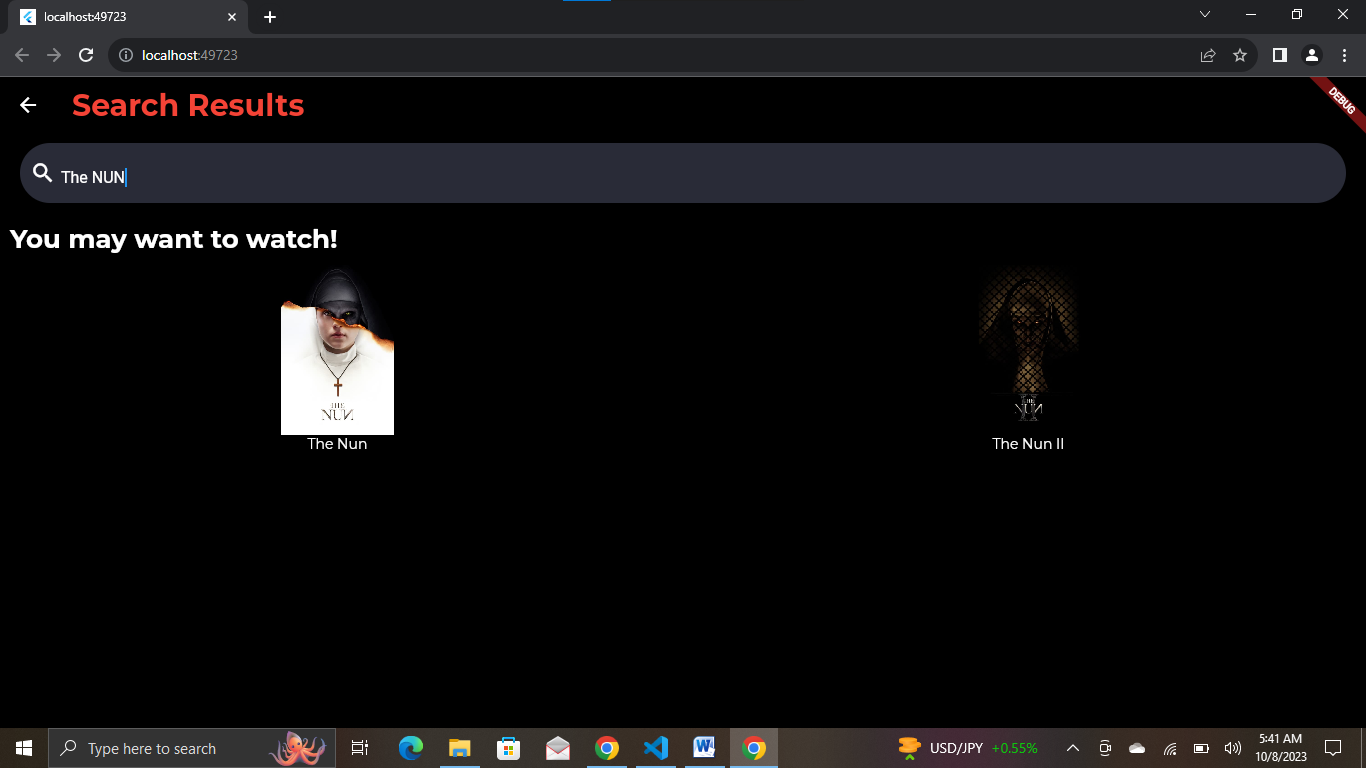
**Web:**

****

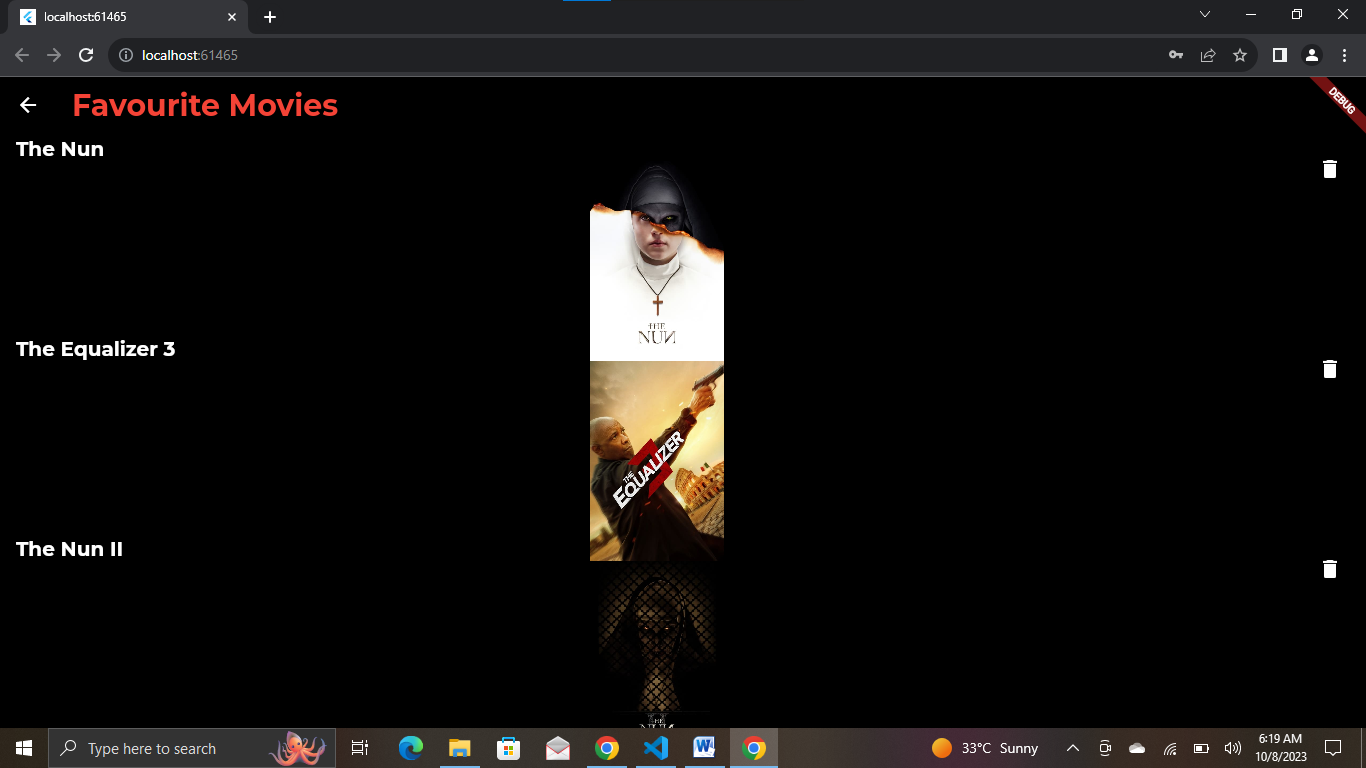




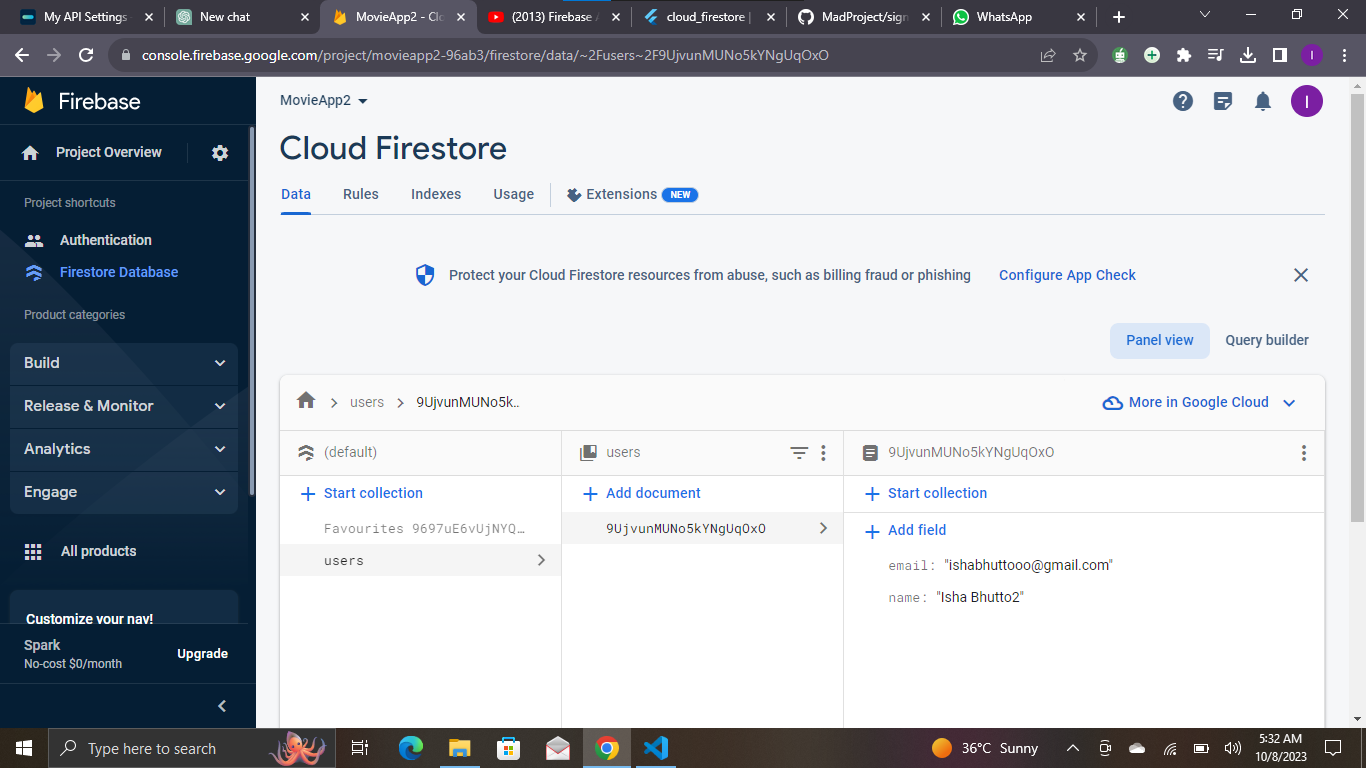
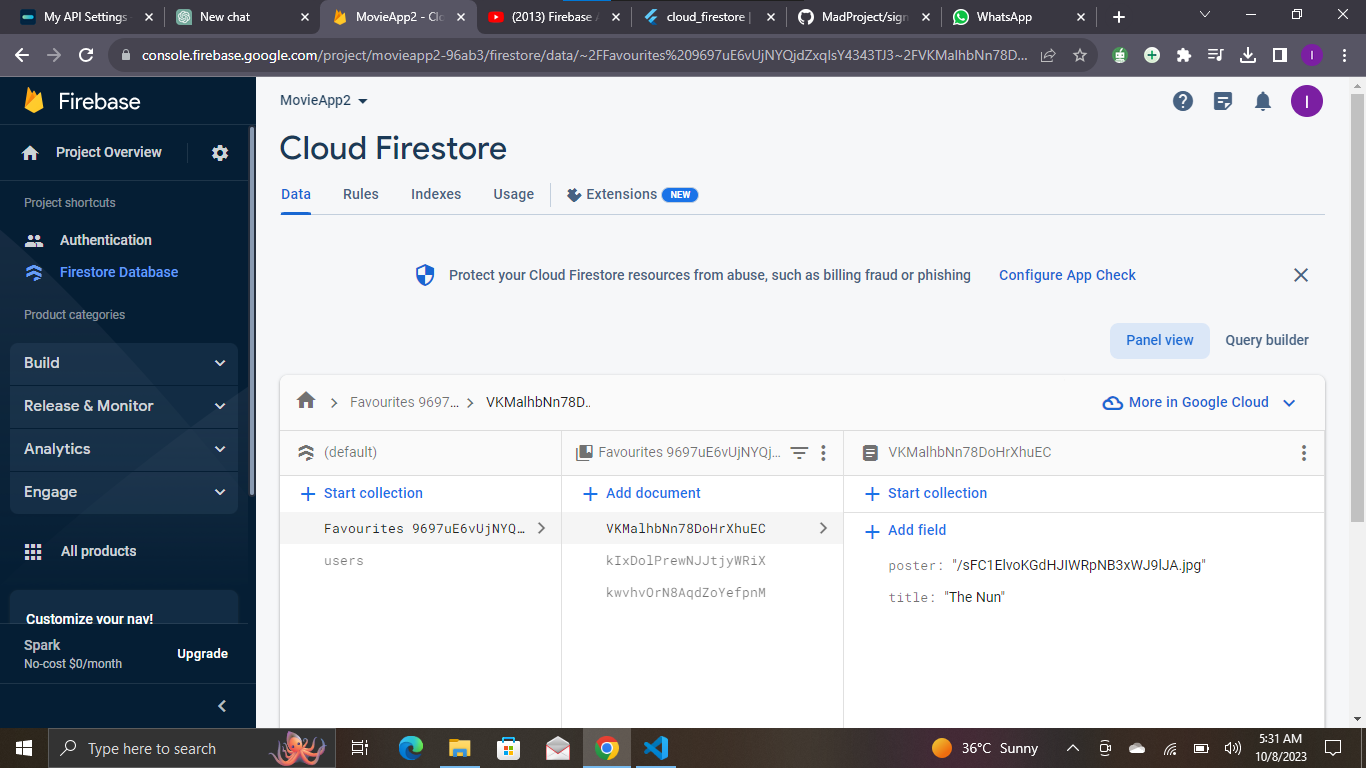
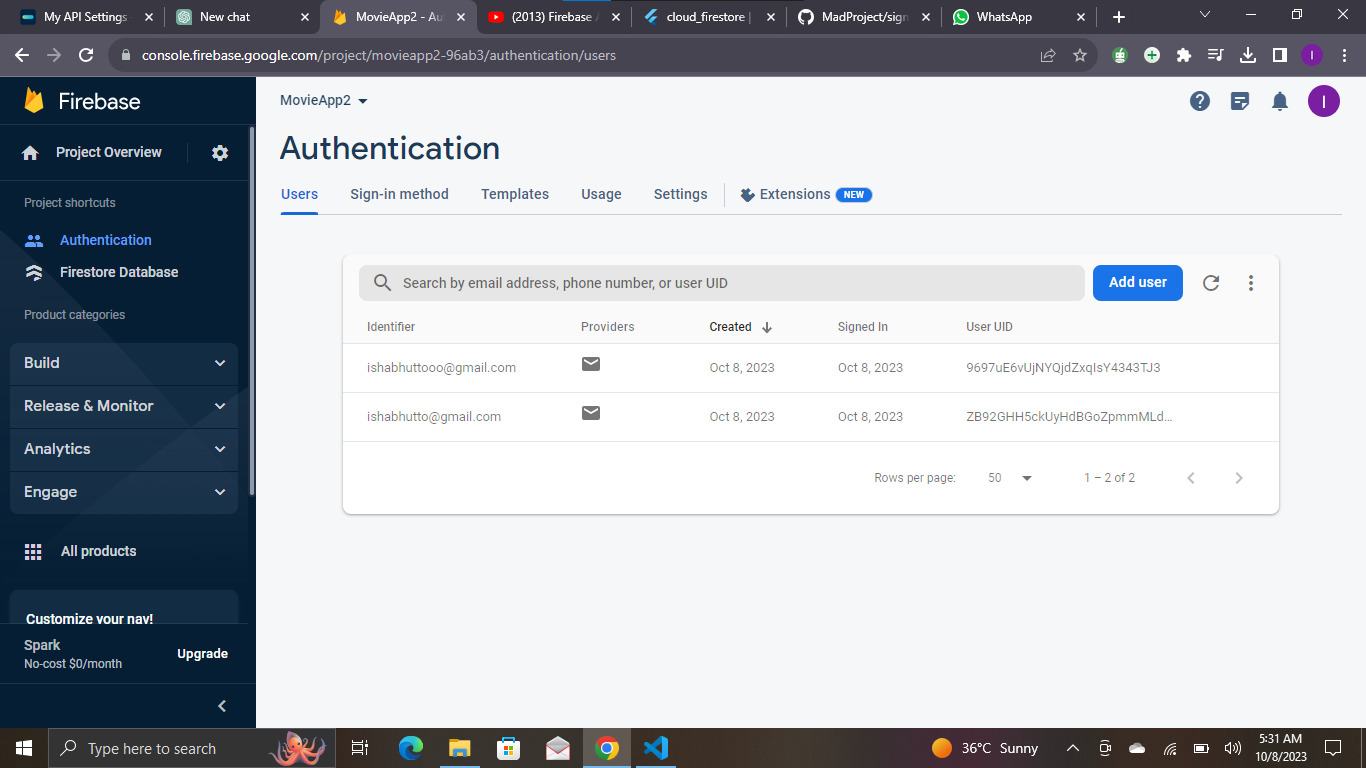


****





**FirBase**

****

**Data Storage (With justification for using a particular database)**

We have used Firebase for 2 purposes

1. Authentication: For authentication with email and password in login credentials and signup. User can also update password with the link if click on forgot passwords the credentials will be updated automatically.
2. Data Storage: We have implemented database for saving information of favorite movies and retrieving and displaying information. User can also delete the items from favorites. There are separate favorite lists for every user on the basis of their uid.

**APIs/Packages/Plugins Used in the Movie App**

In the development of our movie app, we have leveraged a variety of APIs, packages, and plugins to enhance functionality, streamline development, and provide a better user experience. Below is a list of the key APIs and packages we have integrated, along with justifications for their use:

1. **TMDB API (The Movie Database API):**

**Justification:** We have integrated the TMDB API to access an extensive movie database. This API provides us with a rich source of movie information, including details such as movie titles, genres, release years, synopses, cast and crew data, ratings, and images. By using this API, we ensure that our app offers users an up-to-date and comprehensive catalog of movies, enhancing the overall user experience.

1. **Authentication APIs (e.g., OAuth for Social Logins):**

**Justification**: We have incorporated authentication APIs, such as OAuth for social logins, to simplify the user onboarding process. This feature allows users to sign up or log in using their existing social media accounts, eliminating the need for them to create separate credentials. By doing so, we reduce friction during the registration process, leading to increased user adoption and a more seamless user experience.

1. **Firebase (Backend Services):**

**Justification**: Firebase is used for backend services, including real-time database management, user authentication, and cloud storage. Firebase's authentication services provide robust security, while cloud storage simplifies the handling of user-generated content such asin the form of documents collection.

**GENERAL PACKAGES:**

firebase\_auth

cloud\_firestore

http

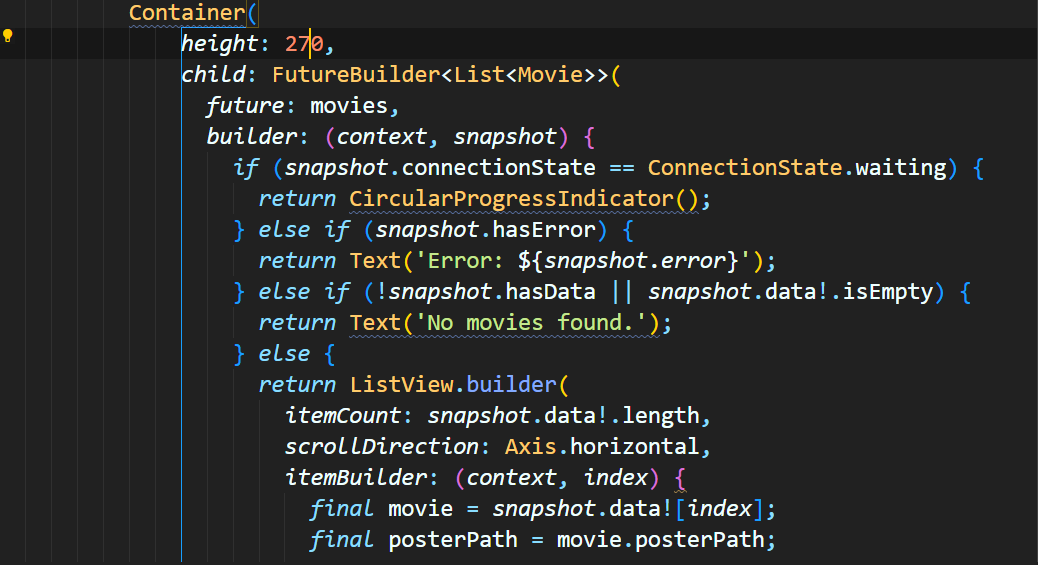
firebase\_core

google fonts

**Issues and Bugs Encountered and Resolved during Development**

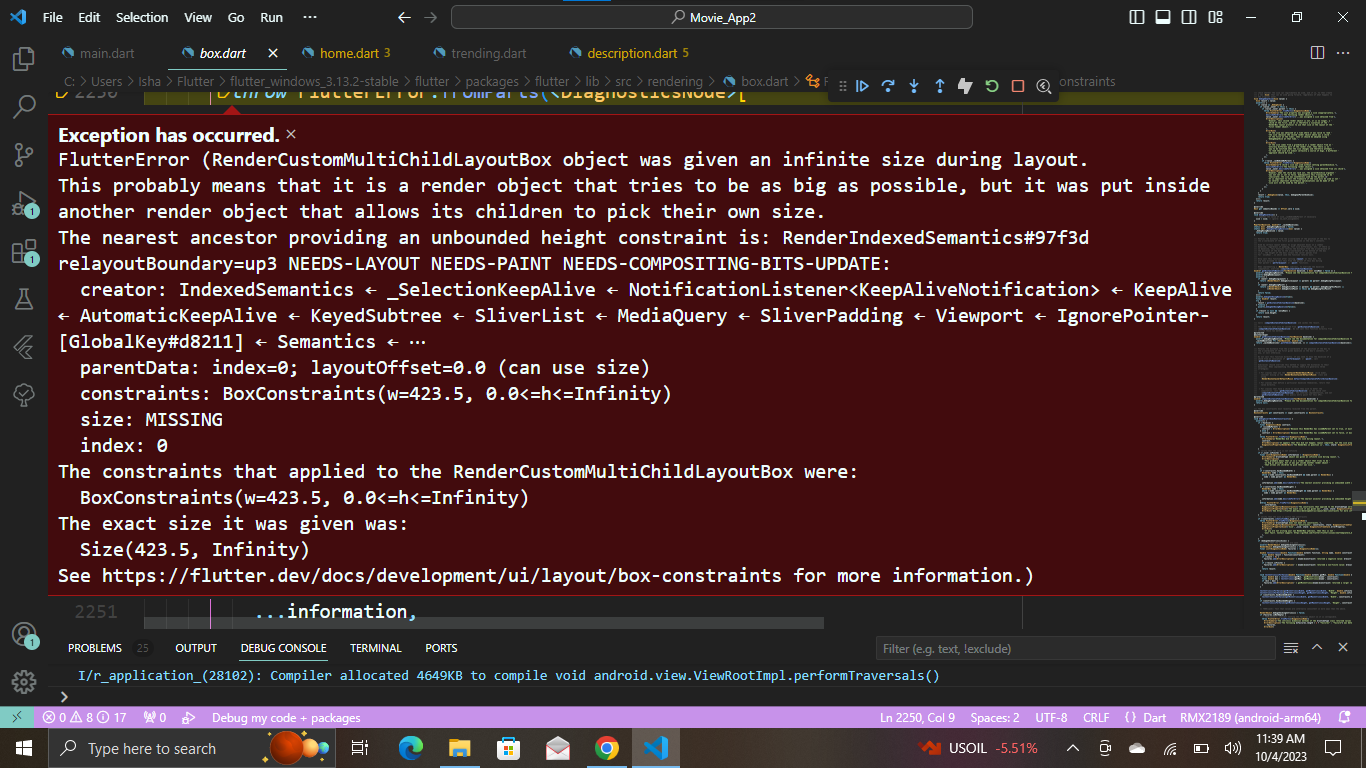
***RenderCustomMultiChildLayoutBox object was given an infinite size during layout*** *typically occurs when a widget inside your layout is trying to occupy an infinite height, and the parent widget does not provide any constraints to limit the height.*

the issue seems to be in this part:

**

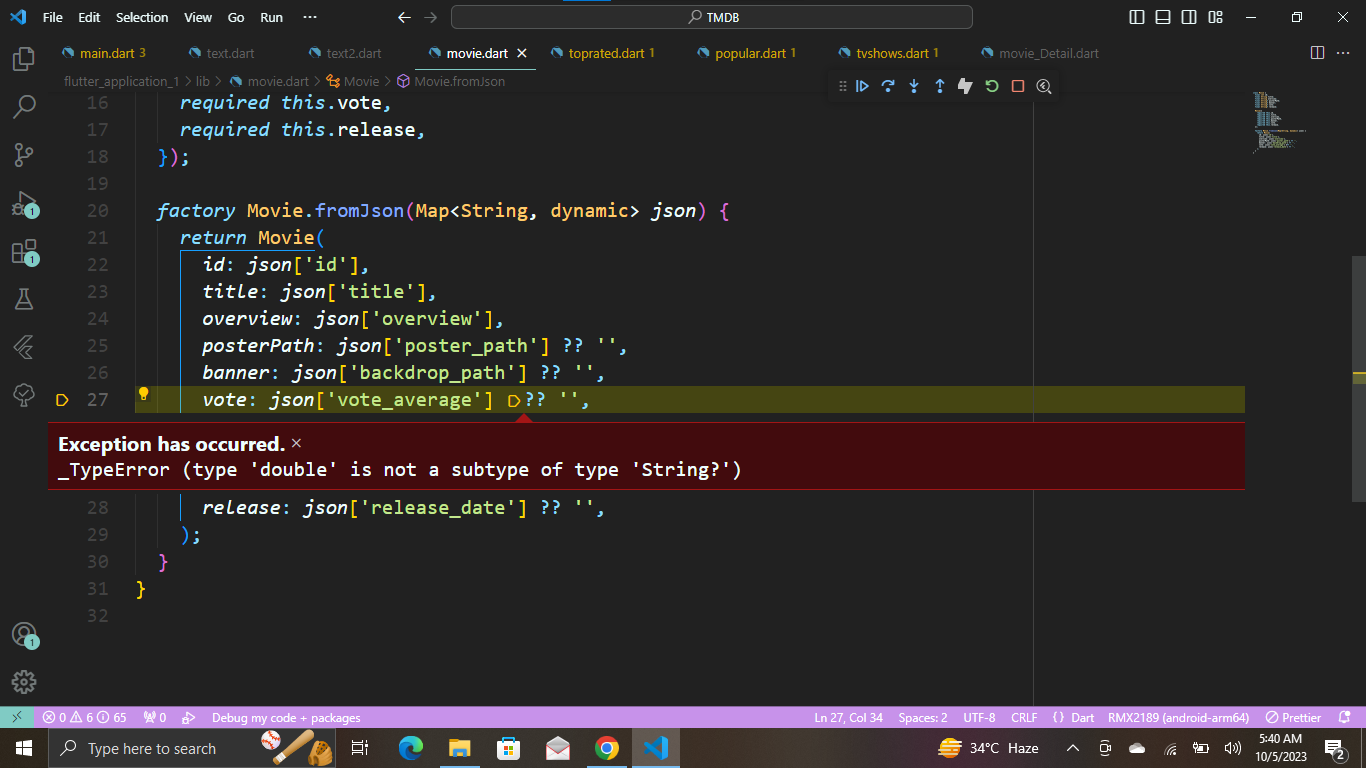
The **Container** widget is trying to set a fixed height of 270, but the **ListView.builder** inside it can potentially have more items than what can fit in that fixed height. This is causing the error because the **ListView.builder** is trying to occupy an infinite height.

To fix this issue, you have a few options:

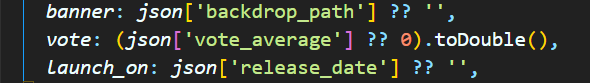
****

****

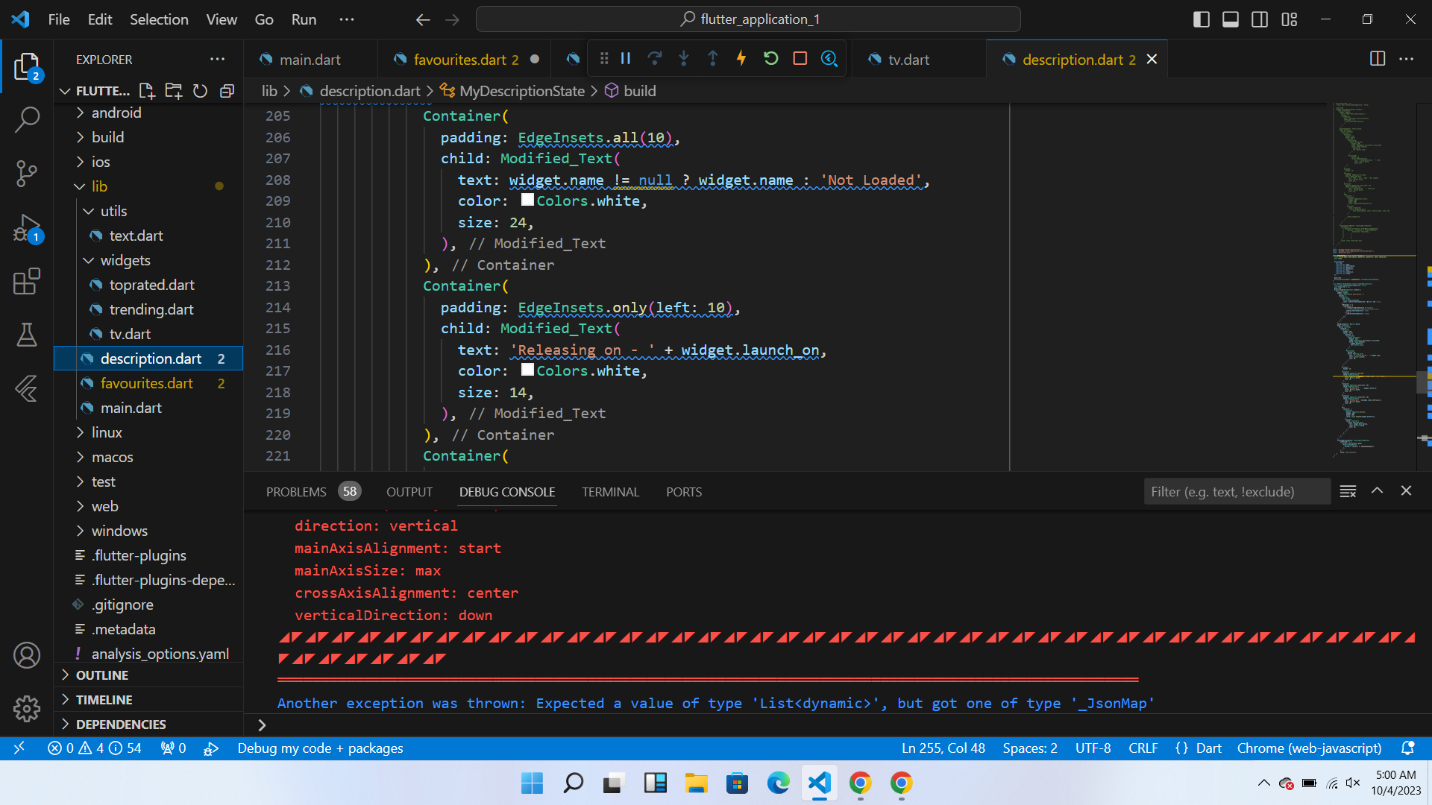
*The error you're encountering \_TypeError (type 'int' is not a subtype of type 'double?') is because the vote\_average field in the JSON response is of type double, but in your Movie class, you've defined it as double? (nullable double).*

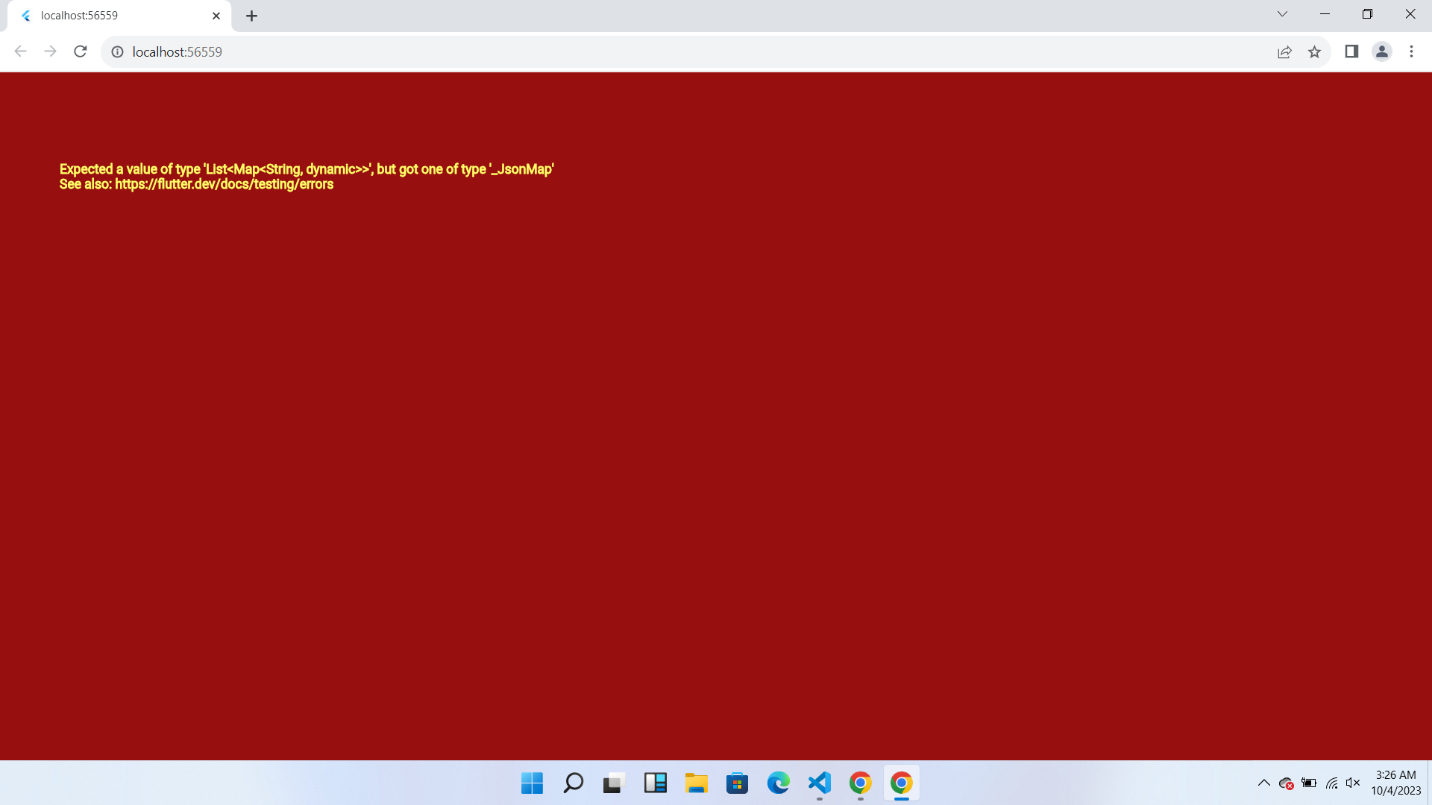
****

To handle this, you should convert the vote field in your Movie class to be of type double.



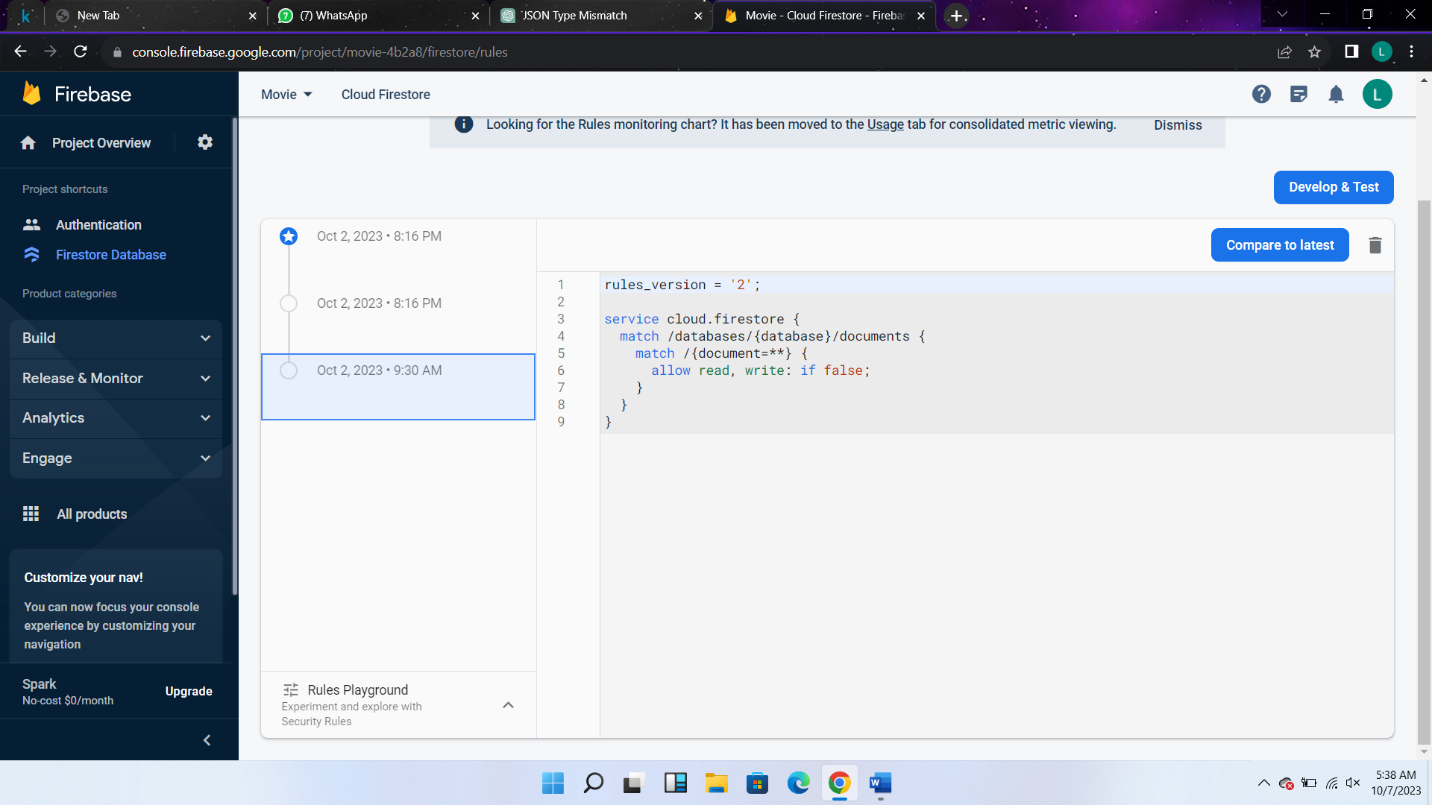
* This error occurred when traversing the data from APIs as the data is in the form of json keypairs. To solve this we decode it into json format.





* The error occurred when I connected firebase to the movie app and when selecting functionalities. I selected production mode afterwards I have an access issue. I tried to write data several times but it was not writing.

So I checked the rules and find out that I have not given read and write access.

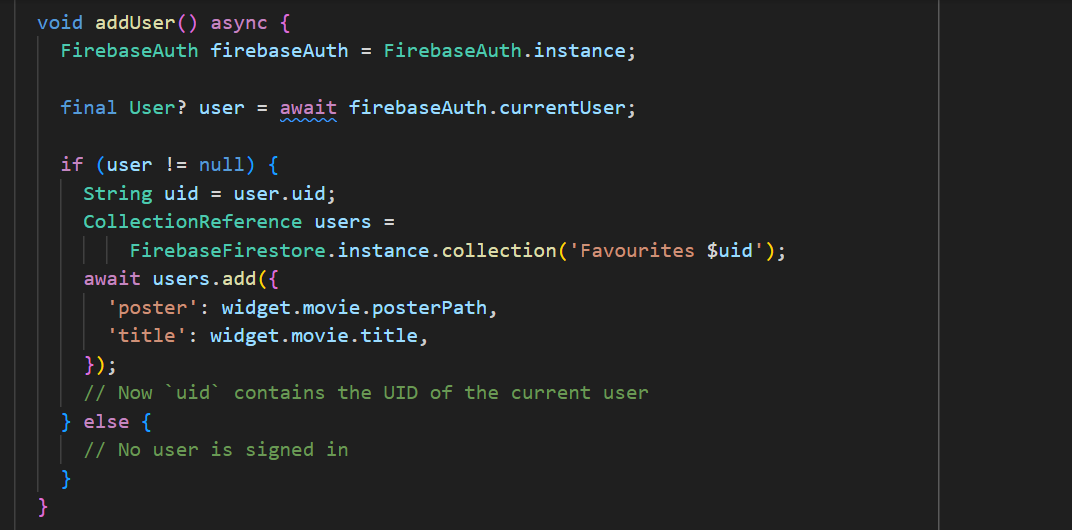


So I changed the rules



* While working with the authentication with firebase it was so easy but storing and fetching each user data of favorites become difficult. First I was getting same favorite collection for every user.

But afterwards I integrated the unique identification of firebase with the every collection of user.

like that. Same procedure is done on retrieving the favorites and deleting



