**Project Report: MoviePedia App**

**Presented by:** 21SW89, 21SW108, 21SW152

**1. Introduction**

The Moviepedia app is a mobile application designed to provide users with detailed information about movies. This includes features such as viewing movie trailers, reading overviews, and exploring additional information related to various movies. The app was developed using the Flutter framework to ensure a smooth, cross-platform experience on both Android and iOS devices.

**2. Real World Problem Identification**

With limited platforms that combine detailed movie information with a social component, many users seek a comprehensive app where they can both explore movies and share their opinions. This lack of social interaction in traditional movie databases creates a need for a platform that encourages users to engage, post reviews, and discuss their favorite films in one place.

**3. Proposed Solution**

Moviepedia was developed as a social platform for movie enthusiasts to not only find information about movies but also share their reviews and opinions. By using Firebase as the central data storage, the app allows users to post, view, and engage with reviews, fostering a social media-like experience within a movie-centric environment.

**4. Technical Specifications**

* **Framework:** Flutter
* **Backend:** Firebase for Data Storage
* **Languages:** Dart (for Flutter)
* **Architecture:** MVVM (Model-View-ViewModel) for code organization and scalability
* **State Management:** Provider package for managing the app state efficiently

**5. Features**

1. **Movie Details and Overview:** Each movie page provides essential information, such as the title, release date, genre, and a short description.
2. **Post Review:** Users can post reviews of their favorite movies to database and store them.
3. **Dynamic UI Updates:** The UI is responsive, adjusting based on screen size to ensure an optimal viewing experience on both tablets and phones.
   1. **Data Storage**

Firebase Realtime Database is used for storing user reviews and movie-related data. Firebase was selected due to its ease of integration with Flutter, real-time data synchronization, and scalability, making it an ideal choice for handling user-generated content efficiently. This setup supports dynamic updates as users post reviews and engage with content.

**7. Development Process**

The Moviepedia app was built using the following development stages:

1. **Planning and Design:** Defined the app’s layout, key features, and target audience.
2. **Firebase Integration:** Utilized a Firebase database to fetch live data about movies, including images, reviews, and descriptions.
3. **UI Development:** Created a visually appealing and responsive user interface using Flutter widgets.
4. **Testing:** Conducted thorough testing to ensure that the app performs optimally on various devices, with attention to Database calls, UI responsiveness, and error handling.

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

* **Issue**: Managing real-time updates for user reviews without performance slowdowns.
  + Resolution: Utilized Firebase’s real-time synchronization, ensuring that new reviews are promptly visible to all users without delay.
* **Issue**: Ensuring the app’s responsiveness across different screen sizes.
  + Resolution: Used Flutter’s built-in widgets and layout options to make the app adaptable and visually consistent across devices.

**9. Conclusion**

The Moviepedia app successfully achieved its primary goals, providing users with a comprehensive, user-friendly, and visually appealing platform for exploring movies. The integration of trailers and real-time data significantly enhances user engagement, while the MVVM architecture and Provider state management ensure that the app is scalable and maintainable.