

FLUTTER PROJECT

TUNEWAVE

A FLUTTER-BASED MUSIC STREAMING APPLICATION

PRAWINRAJ VM

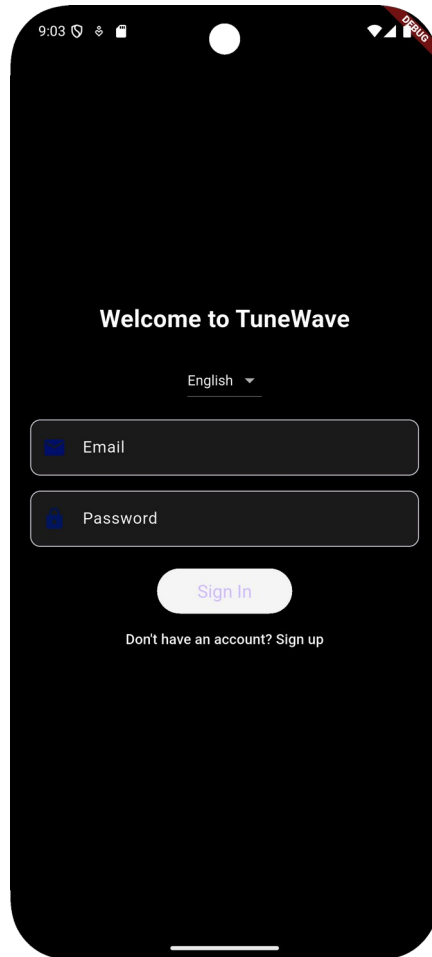
INTRODUCTION

- A Flutter-based music streaming application
- Provides a simple and efficient way for users to listen to songs
- Includes song lists, home page, favorites page, profile page, and user authentication

APP FEATURES

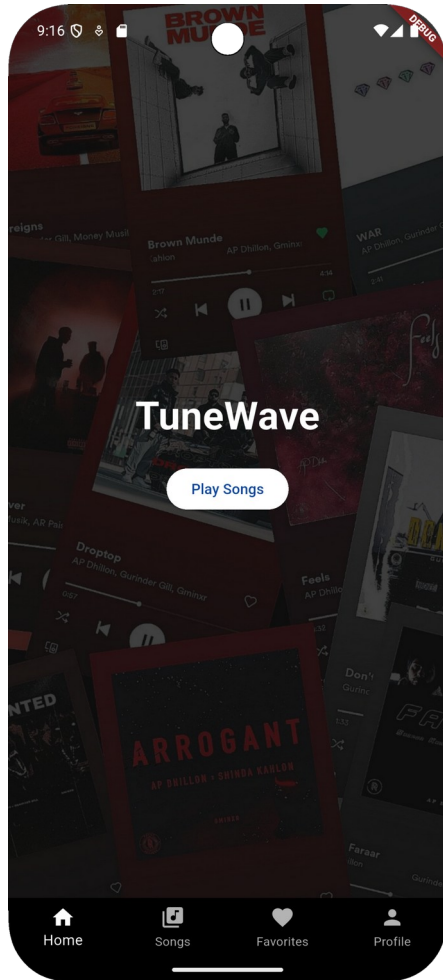
- Music Streaming : Play songs seamlessly with smooth playback controls .
- Home Page : Displays available songs categorized for easy browsing.
- Song Lists Page : Shows all available songs for users to explore.
- Favorites Page : Allows users to mark and access their favorite songs quickly.
- User Authentication: Secure login and signup using Firebase authentication.
- Profile Page : Manage user details and preferences.

LOGIN PAGE



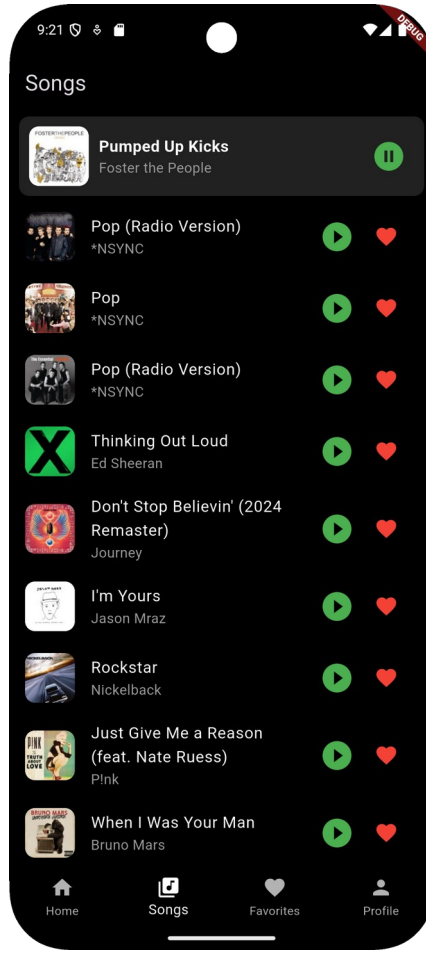
- This login page implements secure user authentication using Firebase.
- It supports email and password login, signup, and stores user details in Firestore.
- User input validation ensures proper email format and password strength, while error handling provides clear feedback.
- The UI features a modern dark theme with intuitive navigation and a language selection dropdown for multi-language support.
- Firestore updates login timestamps for tracking user activity.
- Upon successful authentication, users are redirected to the homepage. The app ensures a seamless user experience with proper state management.

HOME PAGE



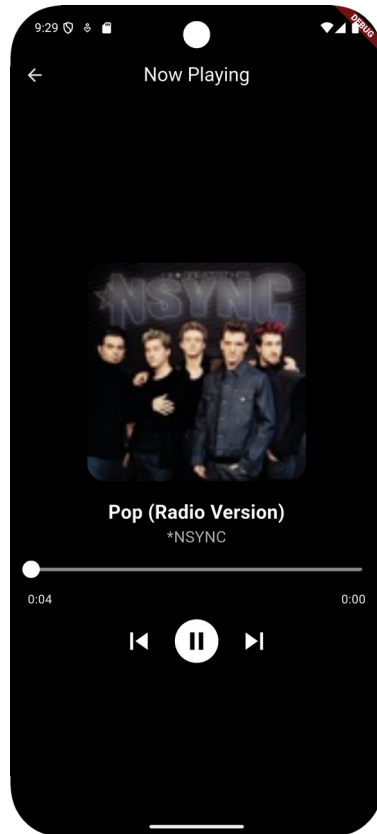
- The home page of this Flutter music app provides a seamless user experience with a modern dark-themed UI.
- It features a background image with transparency, enhancing the aesthetic appeal. Users can navigate through different sections: Home, Songs, Favorites, and Profile using a bottom navigation bar.
- A floating mini-player appears when a song is playing, allowing easy control of playback.
- The SongListPage enables browsing and selecting music, while the FavoritesPage lets users save preferred tracks.
- Localization support ensures a multilingual experience. Future enhancements include playlist creation, real-time lyrics, and AI-based music recommendations.

SONGS LIST PAGE



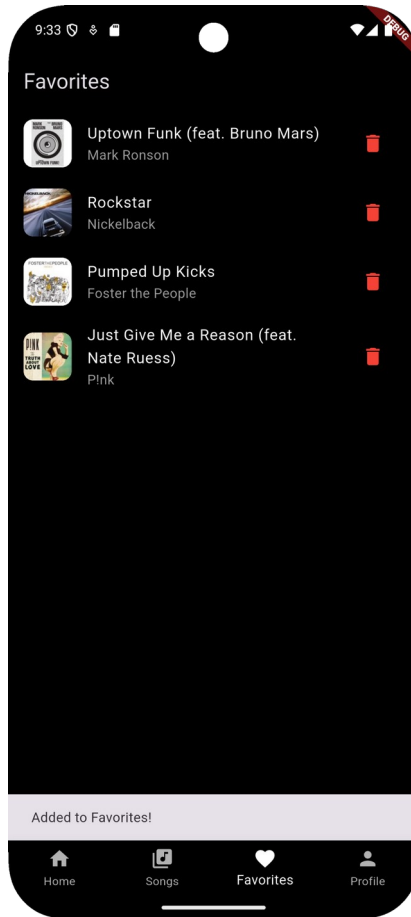
- The Song List Page allows users to browse and play music from an iTunes API.
- It features a sleek dark-themed UI with a dynamic song list, displaying track names, artists, and album artwork.
- Users can play, pause, and navigate between songs seamlessly.
- A mini-player shows the currently playing track, enhancing the user experience.
- Songs can also be added to favorites for quick access.
- Smooth navigation to the Music Player Page ensures an immersive listening experience.

MUSIC PLAYER



- The Music Player Page provides an immersive audio experience with a sleek dark-themed UI.
- It displays album artwork, song title, artist name, and a progress slider for seamless control.
- Users can play, pause, skip tracks, and seek within songs. The app maintains audio state continuity using a shared `AudioPlayer` instance.
- Smooth transitions between tracks enhance user engagement. Future improvements include shuffle, repeat, equalizer settings, and offline playback for a more feature-rich experience.

FAVOURITES PAGE

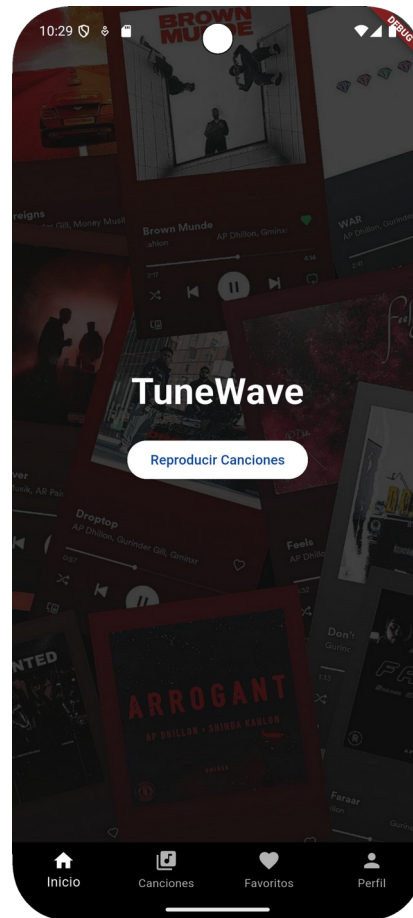
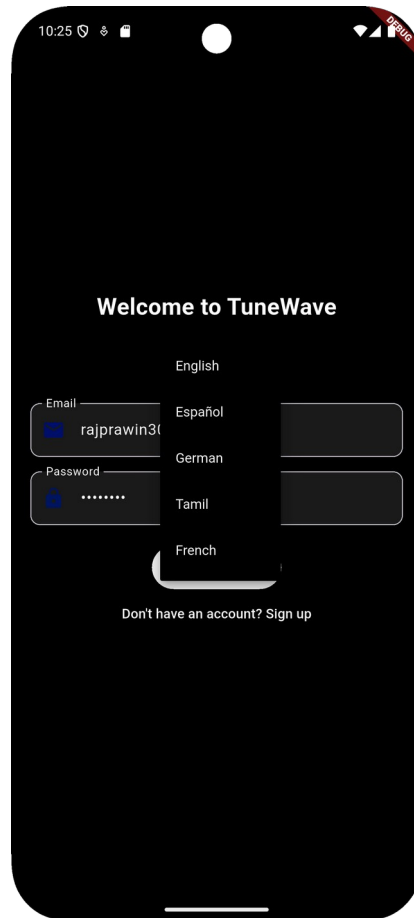


- The Favorites Manager efficiently handles user favorites using Firebase Firestore for persistent storage and Firebase Authentication for user-specific data management.
- It allows users to add, remove, and retrieve their favorite songs in real time, with UI updates powered by ChangeNotifier.
- The system ensures that favorites remain saved even after app restarts while preventing unauthorized access for non-logged-in users.
- The Favorites Page provides a visually appealing interface, displaying album artwork, song details, and a delete option.

WIDGETS

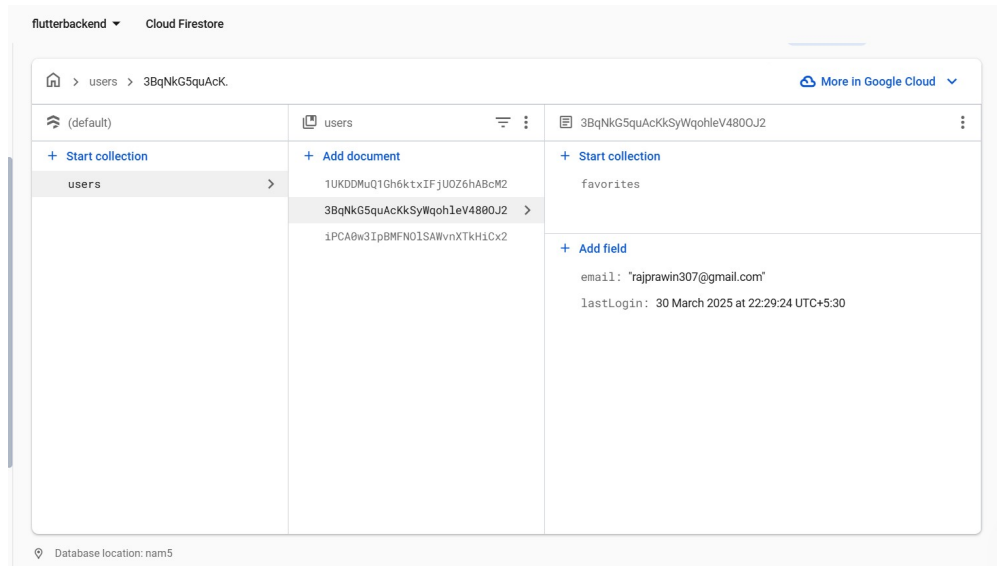
- The SongControls widget is a reusable Flutter component designed to manage audio playback using the audioplayers package.
- It provides a minimalist control interface with a single play/pause button.
- The widget takes an `AudioPlayer` instance and a `previewUrl`, allowing it to stream music from a provided URL.
- Two callback functions (`onPlay` and `onStop`) notify the parent widget (such as a homepage) about playback state changes.
- The `isPlaying` state toggles between `true` and `false`, ensuring smooth switching between playing and pausing.
- This widget can be easily integrated into music player apps to provide an intuitive playback experience, and it can be extended to include additional controls like stop, rewind, fast-forward, or a progress bar.

APP LOCALIZATION

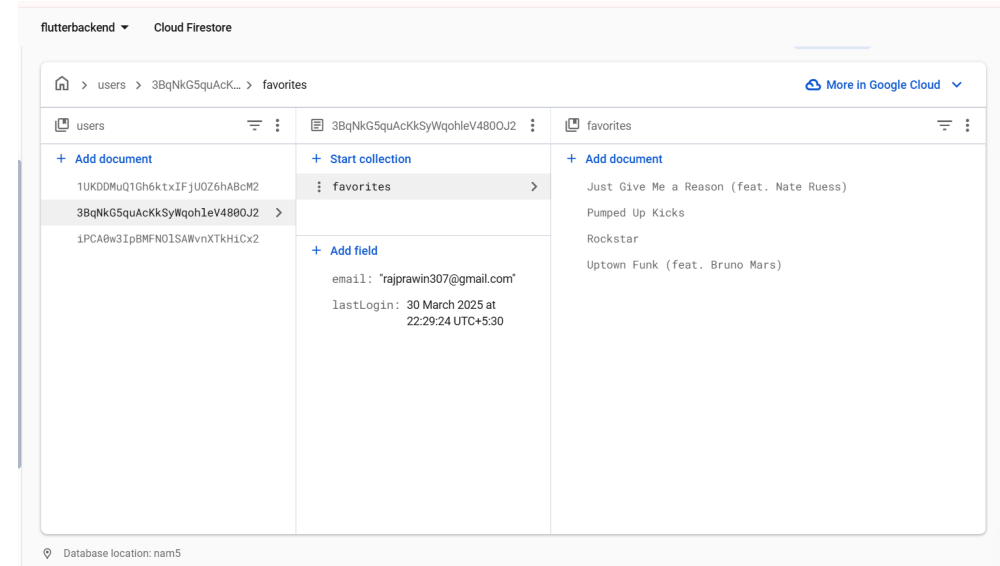


- Tunewave app implements localization to support multiple languages, including English, German, Spanish, French, and Tamil, ensuring a seamless experience for users worldwide.
- This is achieved using Flutter's `flutter_localizations` and `intl` packages, which enable dynamic language adaptation based on the user's device settings.
- The localization process is managed through the `AppLocalizations` class and `LocalizationsDelegate`, which handle the retrieval of translated strings.
- The app automatically detects the device's locale and loads the appropriate translations, enhancing usability and accessibility.

FIREBASE DATABASE



User login details stored once they signed up



Users favorites songs added to the database

THANK YOU