Flutter developer

Task: Create a MP3 Player with Equalizer Visualization in Flutter

Description:

You are tasked with building an MP3 player in Flutter. The player should feature a basic interface with "Play" and "Pause" controls, along with a static equalizer that visualizes the audio's frequency with a dynamic progress bar.

Figma link:

https://www.figma.com/design/UT91VdEhsaXx3R7V0I7NfR/Flutter-Assignment?node-id=0-1&node-type=canvas&t=ouVPoIvkPb5EM99H-0

Requirements:

1. Audio Playback:

- The player should stream an audio file from the following URL
- Use the http package to download the audio file.
- The player should include only two controls:
  - Play: Starts audio playback.
  - Pause: Pauses the audio

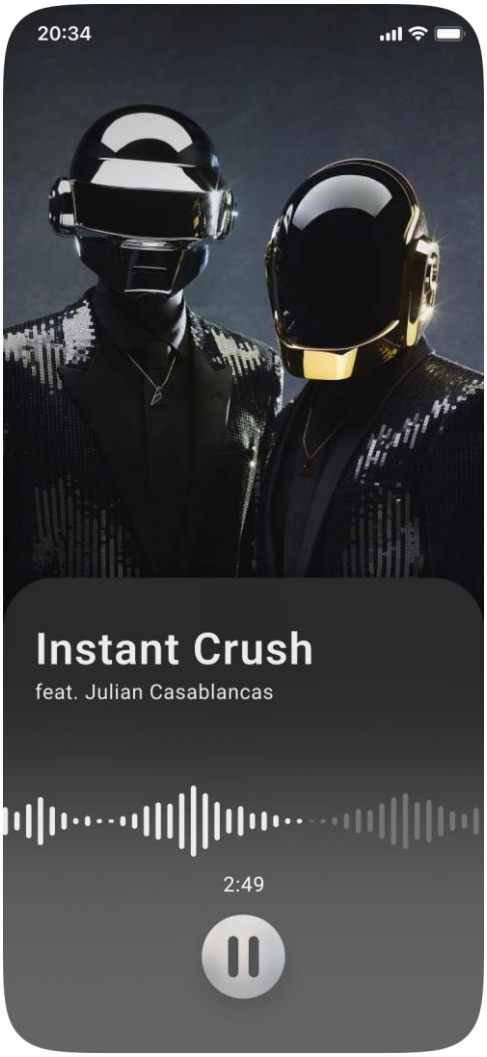
2. Wave Visualization:

- Generate a wave form from the audio using any flutter library

3. State Management:

- Use flutter_bloc for state management throughout the app.

Reference image on the right



Evaluation criteria:

- UI Implementation & Design Fidelity
    - Correctness against Figma Design: How closely does the UI match the provided Figma design? Consider spacing, colors, fonts, and layouts.

- API Integration
    - Correct API Usage: Does the app successfully fetch and display product data from the API? ○ Error Handling: How does the app handle failed requests, timeouts, or API errors (e.g., network issues, invalid responses)?
    - Loading States: Are loading indicators shown appropriately when fetching data?
    - Data Display: Is the product data correctly displayed in the UI according to the Figma design?

- State Management
    - Correct Implementation of BLoC Pattern: Is the state managed using the BLoC pattern effectively? Are states separated logically, and is the business logic decoupled from UI?
    - State Handling: Are state transitions smooth, and are all possible states (loading, success, error) handled properly?
    - Modularity of BLoC Components: Is the code organized into clear and distinct parts (Events, States, Bloc classes

- Code Quality & Best Practices
    - Clean Code: Is the code well-organized, readable, and maintainable, following Dart and Flutter best practices?
    - Modularity: Is the code broken down into reusable components, widgets, and classes?
    - Extensibility & Scalability: Is the code structured in a way that allows for easy expansion or modification of features in the future?
    - Use of Comments & Documentation: Are the codebase and complex logic well-documented with comments where necessary?