

Take-Home Project: React Native Music Search App

Assignment Overview

Build a simple **React Native** application (written in **TypeScript**) that allows users to search for music and listen to short previews. This assignment should take roughly **4–6 hours** of your time. The focus is on demonstrating coding style, architectural decisions, and handling data flow in a mobile environment.

Core Requirements

1. **Search Feature**
 - Implement a search bar where users can enter an artist or track name.
 - Fetch matching music data from a public API (e.g., [Spotify Web API](#) or any open music service).
2. **Results List**
 - Display the results in a scrollable list, showing basic info (e.g., track name, artist, album art).
3. **Details & Preview**
 - Allow users to tap on a result to see more details on a separate screen.
 - If available, provide a way to play a short audio preview.
4. **TypeScript**
 - The entire app must be written in TypeScript, demonstrating proper use of types, interfaces, and/or generics.

Bonus (Optional)

- **Favorites**
 - Let users "favorite" tracks, storing them locally with AsyncStorage or any state management solution (e.g., Redux or React Context).
- **Testing**

Add a few unit or integration tests with a testing framework (e.g., Jest, React Native Testing Library).

What to Submit

1. **GitHub Repository Link**
 - Provide a public or private repository link (if private, grant access).
2. **README**
 - Briefly describe how to install and run the project.
 - Outline any design decisions or major libraries used.

- Note any known limitations or areas where you'd like to improve if given more time.

Evaluation Criteria

- **Code Quality & Organization:** Is the project structured logically? Are components, services, and types well organized?
- **Correctness & Functionality:** Does the search feature work? Can users view track details and play previews without errors?
- **TypeScript Usage:** Are type definitions used appropriately and consistently?
- **UX & UI:** Is the interface intuitive, responsive, and visually consistent?
- **Error Handling:** How are unexpected API results or network errors handled?

Estimated Timeframe

We recommend spending around **4–6 hours** on this project. We do **not** expect a perfectly polished product—please focus on showing how you think through problems and implement solutions.

Thank you, and good luck with the assignment!