Technical Assessment

Summary

The assessment below has two technical tests namely:

1. Next js assessment
2. React Native assessment

Please select the applicable test(s) and kindly read the instructions carefully and submit the result after completing the assessment at careers@cabsandmore.com.

Only select any of below or both based on your preference.

Deadline for submitting the assessment is on 25/03/2025

Feel free to ask any question for clarification.

1. Next.js Takeaway Test: Github Gist Tracker App

Objective:

Develop a **Github Gist Tracker App** using **Next.js, MongoDB, and Tailwind CSS**. The application should allow github users to **sign up, log in, create, update, delete, and view github gists**. Users should also be able to manage their **profile information**.

Requirements:

1. Tech Stack:

- **Frontend:** Next.js 15, Tailwind CSS V4

- **Backend:** API routes/ Server Action in Next.js
- **Database:** MongoDB (via Mongoose) for handling user profile
- **Form Handling & Validation:** React Hook Form + Zod
- **APIS**: GitHub Gist API (https://docs.github.com/en/rest/gists)
- **Deployment:** Any platform (Vercel, Netlify, etc.)

2. User Features:

- **Authentication**
    - User can register with email & password
    - User can log in and log out
    - Protected routes for authenticated users
- **Profile Management (CRUD)**
    - View, edit, and update profile information (name, bio, avatar, etc.)
    - Delete account
    - Save profile on database (CRUD)
- **Gist Management (CRUD)**
    - Create a new gist (title, description, code snippet)
    - View a list of gists
    - Edit & delete own gists
    - View individual gist details
    - Search/filter gists

Responsiveness

- The app should work on mobile, tablet, and desktop screens

3. Bonus (Optional):

- Allow users to **star/favorite** gists
- Implement **pagination** for gist listing
- Use **Mapbox** to show the user's location (if allowed)
- Deploy the application

Submission Guidelines:

- Host the code on **GitHub** (private/public repo) & share access
- Deploy the app and provide a **live URL**

- Submit a **README.md** with setup instructions and any extra notes

Good luck! 🚀

2. React Native Takeaway Test: Soccer Games App

Objective:

Develop a **React Native App using Expo** that fetches upcoming soccer games from the **Free Sports API (https://www.thesportsdb.com/)**. The soccer match should be from American League Major League Soccer. The app should have **authentication, profile management, and search functionality** implemented with **Firebase and Redux**.

Requirements:

1. Tech Stack:

- **Frontend:** React Native (Expo 50+)
- **State Management:** Redux Toolkit
- **Backend:** Firebase (Firestore for user profiles, Firebase Auth for authentication)
- **API:** Free Sports API (fetch upcoming soccer games)
- **Navigation:** React Navigation/Expo router (Bottom Tabs)
- **Styling:** GlueStack UI

2. App Features:

- **Authentication**
    - User can **sign up** and **log in** using Firebase Auth
    - Protected routes to ensure only logged-in users can access certain features
- **Bottom Tab Navigation**
    - **Home Tab:** Displays upcoming soccer games fetched from Free Sports API
        - Implement **search functionality** using Redux to filter games by team name or league
    - **Profile Tab:** Allows the user to **manage their profile (CRUD operations)**
        - Edit profile details (name, favorite team, profile picture, etc.)

        - Delete account
    - **Favourite:** Allow users to **favorite matches** and store them in database

- **Dynamic Search Logic**
    - Use Redux to manage the search state and filter soccer games dynamically
- **Real-time Updates**
    - Fetch upcoming matches dynamically from Free Sports API
    - Update user profile in Firebase Firestore and reflect changes in real time
- **Responsive Design**
    - Ensure the app works smoothly on both Android and iOS

3. Bonus (Optional):

- Show **match details page** when clicking on a game
- Implement **pagination or infinite scroll** for listing soccer games
- Deploy the app (Expo or APK build)

Submission Guidelines:

- Upload the code to **GitHub** (public/private repo) and share access
- Provide a **README.md** with setup instructions and API usage details
- Share a **demo video or APK file** for review (optional but preferred)

Good luck! ⚽🚀