Welcome to the project! This short guide introduces the overall structure of the codebase and highlights ke

## General Structure

- **App Entry & Navigation**
  - ‘App.js‘ sets up the React Navigation stack with screens like ‘Login‘, ‘Signup‘, ‘Events‘, ‘AdminEvents‘, a

- **Firebase Configuration**
  - ‘firebaseConfig.js‘ initializes Firebase services (Firestore, Authentication, and Storage) and exports ther

- **Push Notifications**
  - ‘expoPushNotifications.js‘ registers the device for Expo push notifications. It requests permissions and c

- **API Helpers (in ‘pages/api/‘ directory)**
  - ‘event.js‘ manages creating/editing events, scheduling reminders, and sending push notifications.
  - ‘users.js‘ includes authentication helpers and utilities to load the current user or store the device’s notific

- **UI Components**
  - ‘components/NavBar.js‘ renders the bottom navigation bar. It checks whether the user is an admin to sh

- **Screens (in ‘pages/‘ directory)**
  - ‘events.js‘ lists upcoming events with filters and navigation to individual event pages.
  - ‘signup.js‘ collects user details and interests before creating a user in Firebase Auth.
  - ‘waiver.js‘ stores emergency contact information in Firestore when the user accepts the waiver.
  - ‘profile.js‘ allows users to edit personal info, manage interests, and drop from events.
  - ‘notifications.js‘ displays push notifications retrieved from Firestore.
  - Admin-only screens live under ‘pages/admin/‘ for creating, editing, and viewing events, managing volunt

## Important Concepts

1. **Firebase Integration**
   - Firestore, Auth, and Storage are initialized in ‘firebaseConfig.js‘. Many modules import these directly.
2. **Push Notifications**
   - The device’s Expo notification token is registered on startup and stored in the user’s document. Event r
3. **Admin vs. Regular Users**
   - Screens check if the logged-in user is an admin to show different navigation options and event manage
4. **Events & Shifts**
   - Events store shift IDs referencing separate ‘shifts‘ documents. Users sign up for shifts, and reminders a
5. **Data Flow**
   - Screens fetch data from Firestore on mount or when focused. Updates are written back to Firestore dire

## Next Steps to Explore

- Review the Firestore schema for events, shifts, users, waivers, and notifications.
- Explore the admin-specific pages under ‘pages/admin/‘ to understand event creation and volunteer mana
- Examine ‘expoPushNotifications.js‘ and ‘pages/api/event.js‘ to see how notifications are registered and se
- Study ‘Login.js‘, ‘Signup.js‘, and ‘Waiver.js‘ to follow the user onboarding flow.
- Consider how error cases are handled throughout the screens and API helper functions.

We hope this overview helps you get comfortable with the project. Happy coding!