💧 Water Intake Reminder App Documentation

# 1. Project Overview

App Name: Water Intake Reminder  
  
Short Description: A React Native app that reminds users to stay hydrated by tracking their daily water intake and sending periodic notifications.  
  
Target Audience: Health-conscious individuals, fitness enthusiasts, and people aiming to develop healthier hydration habits.  
  
Platform: Android (via Expo), with potential for iOS support  
  
Technology Stack:  
  
Programming Language: TypeScript, JavaScript  
Framework: React Native (with Expo)  
Storage: AsyncStorage  
Notifications: Expo Notifications API

# ⚙️ 2. Functional Requirements

User Functionality  
- Register basic profile (e.g., weight, activity level)  
- Manually log water intake  
- Set cup sizes and preferred drink types  
- Enable/disable reminder notifications  
- View daily intake progress and history  
  
Other Functionalities  
- Push notifications using local scheduling  
- Offline access using local storage  
- Daily hydration goal calculation based on profile data

# 🛠 3. Technical Requirements

Supported OS: Android 8.0 (API 26) and above  
  
Required Permissions:  
- Notification access (foreground and background)  
  
Minimum Device Specs:  
- 2 GB RAM  
- 720x1280 screen resolution  
- Internet only required for initial installation

# 🧱 4. Architecture Diagram

Structure Overview  
App.tsx  
├── Tabs  
│ ├── layout.tsx,history.tsx,home.tsx  
│ └── Settings: layout,feedback,general,help,index,profile,reminder,tips.tsx  
├── Context  
│ ├── NotificationContext.tsx  
│  
├── Components  
│ ├── collapsible.tsx  
│ ├── externalLink.tsx  
│ └── HapticTAB.tsx  
├  
│ ├── \_layout.tsx  
│ ├── +not-found.tsx  
│   
  
State Management: React Context API  
Notification Handling: Managed by NotificationContext

# 📶 5. Data Flow Diagram (DFD)

User Input (logs intake)  
 ↓  
Local Storage (AsyncStorage)  
 ↓  
App State (updated via context)  
 ↓  
UI Components (e.g., progress ring)  
 ↓  
Reminders Scheduled (Expo Notification API)  
 ↓  
Notification Triggered  
 ↓  
User opens app from notification

# 🗃 6. Database Design

Local Storage via AsyncStorage  
  
Tables (Keys in AsyncStorage):  
@user\_profile:  
{ weight: number, activityLevel: string, climate: string }  
  
@daily\_intake:  
[ { date: string, amount: number, timestamp: string } ]  
  
@reminder\_settings:  
{ startTime: string, endTime: string, interval: number }

# 🌐 7. API Documentation

Expo Notifications (Local)  
  
scheduleNotificationAsync(config)  
Params:  
- title: "Time to drink water!"  
- body: "Stay hydrated and healthy."  
- trigger: { hour: number, minute: number, repeats: true }  
  
cancelAllScheduledNotificationsAsync()  
- Cancels all scheduled reminders.  
  
Authentication: Not required (no backend)

# 🧪 8. Testing Strategy

Manual Testing  
- Each feature tested on physical and emulated Android devices  
- Use case flows:  
 - Logging intake  
 - Editing settings  
 - Receiving notifications  
  
Automated Testing (optional for this MVP)  
- Planned for future using:  
 - Jest for unit testing  
 - Detox for E2E testing  
  
Tools Used:  
- Expo Go for real-device preview  
- Android Studio Emulator

# 🧑‍🎨 9. UI/UX Design

Design Goals:  
- Clean and calming (blues and whites)  
- Fast access to log water with minimal taps  
- Responsive layout for various screen sizes  
  
Navigation Flow:  
Home → [Log Intake] → Settings  
 ↓ ↓  
 History Profile / Reminder Settings  
  
Mockups & Wireframes:  
- Designed in Figma (optional placeholder if you used it)

# 🔐 10. Security Measures

- No sensitive data stored or transmitted  
- AsyncStorage is used locally on device  
- Notifications are local; no cloud messages  
- App permissions are explicitly requested

# 🪛 11. Setup Instructions

For Developers:  
Clone the Repository  
git clone https://github.com/yourusername/water-intake-reminder.git  
cd water-intake-reminder  
  
Install Dependencies  
npm install  
  
Run the App  
npx expo start  
  
Test on Device  
- Use Expo Go on Android  
- Or run via Android Studio emulator  
  
For Testers:  
- Install APK from release/ folder (if available)  
- Or use Expo Go to scan the QR code and run live preview