# SafeTrack Technical Stack Overview (English)

## 1. 📱 Frontend Stack (Mobile - Flutter)

• Flutter SDK: Build cross-platform native apps (Android/iOS) — Fast development, unified UI

• provider / riverpod: State management — Control health data, alerts, etc.

• sqflite: Local database — Store health history, thresholds

• shared\_preferences: Key-value storage — Save user settings like preferences or thresholds

• http / dio: HTTP requests — Communicate with FastAPI and third-party APIs

• fl\_chart: Chart display — Visualize health trends (Should have)

• google\_fit: Access heart rate, steps — Android only (Could have)

• flutter\_local\_notifications: Local alerts — Threshold alerts (Should have)

## 2. 🔙 Backend Stack (Python + FastAPI)

• FastAPI: Create RESTful API services — Handle health data, alert logic, etc.

• Uvicorn: Run FastAPI apps efficiently — ASGI server for local development

• Pydantic: Data model validation — Ensure safe input/output data structure

• CORS Middleware: Allow API access from Flutter — Support cross-origin requests

• JWT (optional): User authentication — If login is enabled

## 3. 🗂 Database

• SQLite: Lightweight local database — Used by both frontend and backend in prototypes

• sqflite (Flutter plugin): Operate SQLite on device — Store history, thresholds

## 4. 🌐 Third-Party API Integration

• OpenWeather API: Get temperature, humidity, air quality — ✅ Should have | Used for environmental health info

• Google Fit API: Get step/heart rate data — ⚠ Could have | Android-only, requires OAuth

• Firebase API: Cloud storage and account system — ❌ Won’t have | Disabled in current version

• NHS API (UK): Retrieve health conditions, symptoms — ❌ Won’t have | Reserved for future use

## 5. 📁 Recommended Project Structure (Flutter)

/lib

┣ /screens # Home, Input, Alerts, etc.

┣ /models # HealthData, WeatherData, Threshold

┣ /services # API service, local DB service

┣ /providers # State management

┣ /widgets # Reusable UI components

┗ main.dart # App entry point

## 6. ⚙ Development Tools

• Postman: Test API endpoints

• Git + GitHub: Version control

• VS Code or Android Studio: Flutter development environment

• Render / Railway (optional): Deploy FastAPI backend for testing