

FAP NextGen App

Comprehensive Documentation & User Guide

For: MBBS Students, Faculty, and Developers
Context: Competency-Based Medical Education (CBME)
Family Adoption Programme (FAP)

Table of Contents

1. Introduction & Context
2. Understanding Key Terms (FAP, CBME)
3. User Manual (For Students)
4. Features Overview
5. Technical Documentation (For Developers)
6. Glossary

1. Introduction & Context

The **FAP NextGen App** is a digital solution designed to modernize the Family Adoption Programme (FAP) within the Indian medical curriculum. It replaces traditional paper logbooks with a robust, offline-first mobile application that allows MBBS students to track community health data effectively.

Why was this app built?

Under the new CBME curriculum, every medical student is required to "adopt" families in rural areas and monitor their health longitudinally over 4 years. Managing this data on paper is inefficient, prone to data loss, and lacks analytical value. This app solves these problems by digitizing the entire workflow.

2. Understanding Key Terms

What is FAP (Family Adoption Programme)?

The Family Adoption Programme is a community medicine initiative mandated by the National Medical Commission (NMC). It requires medical students to visit adopted families regularly, monitor their health, provide education, and track parameters like immunization, hygiene, and nutrition.

What is CBME?

Competency-Based Medical Education (CBME) is an outcome-driven training approach. Unlike traditional education which focuses on knowledge acquisition, CBME focuses on the learner's ability to successfully perform specific medical tasks (competencies). This app helps students demonstrate these competencies through verified field visits and reflective journaling.

3. User Manual (For Students)

3.1 Getting Started

1. Open the app in your browser (works on Chrome/Edge on Mobile & Desktop).
2. Login with your student credentials (Email & Password).
3. Upon first login, ensure you have a stable internet connection to sync initial data.

3.2 The Dashboard

The main dashboard gives you a quick overview of your fieldwork progress:

- Total Families Adopted
- Total Village Profiles Mapped
- Pending Visits or Reports

3.3 Managing Families (Digital Family Folder)

This is the core feature. Instead of carrying physical files, you manage everything here.

Adding a Family

Navigate to "My Families" and click "Add Family". You will need to enter:

- Head of Family Name
- Village Name
- Contact Details
- Geo-location (if enabled)

Family Details & Members

Inside a family folder, you can add individual members. For each member/patient, you can track:

- Demographics (Age, Gender, Education, Occupation)
- Health Status
- Chronic Diseases

Socio-Economic Verification

The app automatically calculates the Socio-Economic Class (SEC) using the BG Prasad Scale (updated for 2024). You simply input the income, and the app tells you the class (Class I to V).

3.4 Logging Visits (The Logbook)

Every time you visit a family, you must log it.

1. Go to the Family Card.
2. Click "Add Visit".
3. Enter Date, Activity Done (e.g., "Health Education", "BP Check"), and Outcome.
4. Save. This records your attendance.

3.5 Community Diagnosis (Village Profile)

Students must also understand the community. The "Village Profile" section allows you to survey the village infrastructure: Water sources, Waste disposal, Health centers availability, etc.

4. Special Features

4.1 AI Medical Coach & Reflections

This is a unique feature. After a visit, students are encouraged to write a "Reflection".

- How it works:
- You write about your experience using the Gibbs Reflective Cycle (Description -> Feelings -> Evaluation -> Analysis -> Conclusion -> Action Plan).
- The AI Coach (powered by Google Gemini) reads your reflection.
- It gives you instant feedback on your clinical reasoning and empathy.
- It suggests medical topics you should read up on based on what you saw (e.g., if you mention a coughing patient, it might suggest reading TB guidelines).

4.2 Offline Mode

The app is a PWA (Progressive Web App). You can load it once while online, and then go to rural areas with no internet. You can view families and add new data. The app will sync ("Base64 Bypass Technology") when you are back online.

5. Technical Documentation (For Developers)

This section is for IT teams or future developers maintaining the app.

5.1 Technology Stack

Component	Technology
Frontend	React 18, Vite
Language	JavaScript (ES6+)
Styling	Vanilla CSS (Performance optimized)
Backend (BaaS)	Supabase (PostgreSQL)
Authentication	Supabase Auth
AI Integration	Google Gemini via OpenRouter
Hosting	Vercel

5.2 Database Schema (Supabase)

The backend is a PostgreSQL database managed by Supabase. Key tables include:

families: Stores core family units. Linked to `student_id`.

family_members: Individual people within a family. Linked to `family_id`.

villages: Community profile data. One per student or shared.

family_visits: Transaction table recording every student visit.

health_measurements: Specific vitals (BP, Weight) linked to a visit and member.

reflections: Student journal entries linked to AI analysis.

teacher_student_mappings: Many-to-Many relationship for teacher oversight.

5.3 Security & RLS

Row Level Security (RLS) is strictly enforced.

- Students can ONLY view/edit their own families.
- Teachers can view families of students mapped to them.
- Admins have full access.

5.4 Setup Instructions

To run this project locally:

1. Clone the repository.
2. Run ``npm install`` to install dependencies.
3. Create a ``.env`` file with Supabase credentials (``VITE_SUPABASE_URL``, ``VITE_SUPABASE_ANON_KEY``).
4. Run ``npm run dev`` to start the local server.

6. Glossary

FAP: Family Adoption Programme

CBME: Competency-Based Medical Education

NMC: National Medical Commission (India)

PWA: Progressive Web App (Web app that works like a native app)

RLS: Row Level Security (Database security policy)

Supabase: An open-source Firebase alternative providing Database and Auth services.