

CHIKITSA — CLEAN PRODUCT & ENGINEERING TEMPLATE

(Mobile App + Web App + Backend + AI Integration)

1. PRODUCT OVERVIEW

Chikitsa is a cross-platform healthcare system designed to securely store medical records, enable emergency access, assist hospital discovery, and provide AI-based educational health insights.

The system does not diagnose or prescribe; it highlights reference-range deviations and assists users in seeking professional care.

2. PLATFORM SCOPE

- Mobile Application (Android & iOS)
- Web Application (Browser-based)
- Backend API (centralized)
- AI Service (separate, external ownership)

3. TECH STACK (MINIMAL & CLEAN)

Frontend (Mobile)

- Flutter

Frontend (Web)

- React (Next.js)

Backend

- Python
- FastAPI
- PostgreSQL

AI (External)

- Python-based AI service
- REST API integration

4. SYSTEM DESIGN (HIGH LEVEL)

Clients (Mobile/Web) → Backend API → Database
→ AI Service (async)

- Frontend handles UI and user interaction only
- Backend owns authentication, storage, permissions, notifications
- AI service processes reports and returns insights
- AI never directly talks to the database

5. CLEAN PROJECT FOLDER STRUCTURE

ROOT/

■

■■■ frontend/

■ ■■■ mobile_app/

■ ■■■■ src/

■ ■ ■■■■ screens/

■ ■ ■■■■ widgets/

■ ■ ■■■■ services/

■ ■ ■■■■ models/

■ ■ ■■■■ main.dart

■ ■

■ ■■■ web_app/

■ ■■■■ src/

■ ■ ■■■■ pages/

■ ■ ■■■■ components/

■ ■ ■■■■ services/

■ ■ ■■■■ models/

■ ■■■■ index.tsx

■

■■■ backend/

■ ■■■ app/

■ ■ ■■■ main.py

■ ■ ■■■ api/

■ ■ ■■■■ auth.py

■ ■ ■■■■ users.py

■ ■ ■■■■ records.py

■ ■ ■■■■ sos.py

■ ■ ■■■■ hospitals.py

■ ■ ■■■■ notifications.py

■ ■ ■

■ ■ ■■■■ core/

■ ■ ■■■■ config.py

■ ■ ■■■■ security.py

■ ■ ■

■ ■ ■■■■ models/

■ ■ ■■■■ user.py

■ ■ ■■■■ medical_record.py

■ ■ ■■■■ insight.py

■ ■ ■■■■ hospital.py

■ ■ ■

■ ■ ■■■■ services/

■ ■ ■■■■ ai_client.py

■ ■ ■■■■ notification_service.py

■ ■ ■■■■ storage_service.py

■ ■ ■

■ ■ ■■■■ database.py

■ ■

■ ■■■■ requirements.txt

■

■■■ ai_service/ (Owned by AI teammate)

■ ■■■■ api/

■ ■■■■ models/

■ ■■■■ logic/

■ ■■■■ main.py

■

■■■ docs/

■■■ product_requirements.pdf

■■■ system_design.md
■■■ api_contracts.md

6. DATABASE DESIGN

User

- id
- name
- email
- phone
- dob
- created_at

MedicalRecord

- id
- user_id
- record_type
- file_url
- report_date
- uploaded_at

HealthInsight

- id
- record_id
- parameter
- value
- reference_range
- status
- explanation

EmergencyContact

- id
- user_id
- name
- relation
- phone

Hospital

- id
- name
- phone
- latitude
- longitude

Notification

- id
- user_id
- message
- scheduled_at
- status

7. DEVELOPMENT PHASES

Phase 1: Auth + File Upload + DB
Phase 2: SOS + Emergency Access
Phase 3: Hospital Discovery
Phase 4: AI Integration
Phase 5: Security + Scaling

8. DESIGN PRINCIPLES

- Clean separation of concerns
- One responsibility per folder
- No hidden logic
- AI treated as an external dependency
- Production-first mindset

END OF DOCUMENT
