Smart Care Assistant for Small Clinics

Project Summary

What We Are Building:

We are developing a Smart Care Assistant web application designed to assist junior doctors and chronic illness patients in small or resource-limited clinics. The platform features two user roles: Doctor and Patient, each with a customized dashboard and tools to improve healthcare workflows.

Who It's For:

- Doctors (especially junior clinicians) managing multiple chronic care patients daily.
- Patients with chronic conditions who need help tracking symptoms and medication.

Project Aim:

To create a functional, user-friendly healthcare assistant that supports real-time decision-making, improves communication between doctors and patients, and detects adverse drug reactions (ADRs) early using smart logic.

PHASE 1: Project Setup (1-2 hours)

1. Initialize the Project

- Set up a GitHub repo
- Use React (for web) or React Native (for mobile) with Vite or Expo
- Setup Tailwind CSS
- Install: npm install tailwindcss chart.js react-router-dom axios

PHASE 2: Doctor Side (5-6 hours)

2. Secure Login & Dashboard

- Frontend: Create login/signup page, routing (/login, /dashboard, /patient/:id)
- Backend: Firebase Auth or Express + JWT; Supabase/Firebase for patient data

3. Patient Dashboard

- /dashboard: List patients
- Add form for new patients

- Each row links to patient details

4. Quick Entry Form

- /patient/:id: Form for vitals, symptoms, prescriptions, follow-up

5. Voice-to-Text Input (Optional)

- Use Web Speech API for dictation

6. Smart Alerts (Rule-based)

- E.g., Alert if 'rash' + 'Amoxicillin'

7. End-of-Day Summary Generator

- Compile notes -> Table -> Export to PDF (jspdf)

PHASE 3: Patient Side (5-6 hours)

8. Patient Login & Logbook

- Daily symptom log, medication checkbox, free notes

9. Trend Visualization

- Chart.js: Line graph of symptoms over time

10. Medication Reminder

- Local notifications or push with Firebase Cloud Messaging

PHASE 4: Smart Logic (ADR & Risk Tags) (3 hours)

11. Rule-Based ADR Detector

- JSON mapping of medication to symptoms

12. Risk Level Tagging

- Detect critical conditions and tag priority as 'high'

Smart Care Assistant for Small Clinics

PHASE 5: Report Generation & Polish (3-4 hours)

13. Generate Discharge or Visit Report

- Export data with jspdf or html2pdf

14. Doctor Handoff Mode

- Printable summary of all patients with flags, prescriptions, and follow-ups

PHASE 6: Final Touches (1-2 hours)

- UI cleanup, color-coded alerts, loading states
- Add icons, dummy patient data
- Deploy using Vercel/Netlify for frontend, Render for backend

Optional:

- Offline mode (IndexedDB/Service Workers)
- AI-based symptom predictor (future)