THE/NUDGE INSTITUTE | Meta

Pragati

AI for Impact Hackathon

Powered by H2S

Team Name : MediMinds AI

Team Leader Name : Sanskruti Udage

Which domain does your idea address? (Agriculture / Healthcare / Skilling / Education): Healthcare

THE/NUDGE INSTITUTE | Meta | H2S

What is the problem you are solving?

How might we leverage AI to revolutionise healthcare delivery by addressing India's deeply rooted challenges in accessibility, affordability, and quality of care — especially for underserved rural and low-income populations?

Key Issues:

- 33% of children under 5 are underweight, 60% anemic
- Late detection of high-risk pregnancies
- Low awareness and enrollment in healthcare schemes (only 14.1% insured)
- Scarcity of healthcare professionals in rural areas (~1:25,000 doctor-patient ratio)

We aim to use **AI as an equalizer** — empowering health workers with actionable insights, multilingual support, and automation to drive **inclusive, community-level impact at national scale.**

THE/NUDGE INSTITUTE | Meta | H2S

Describe your solution. How different is it from any of the other existing ideas? How will it be able to solve the problem? USP of the proposed solution? What is the intended impact of your solution?

Project Name: SwasthyaAI

An AI-powered assistant for frontline workers (ASHA, ANMs) to:
- Assess high-risk pregnancies and anemia
- Deliver local-language voice/text dietary and health info
- Auto-check healthcare scheme eligibility
- Organize past health data for structured, rapid review by doctors
- Function offline with sync on internet availability

USP (Unique Selling Proposition):

- **Offline + Voice + Structured Health Intelligence** — all combined in one lightweight app.
- Modular design means it scales across **languages, states, and use-cases** (e.g., child health, chronic care).

How It's Different:

- **Local-first, AI-powered**: Unlike generic telehealth apps, *SwasthyaAI* is built for **offline-first rural use**, functioning even in 2G zones and syncing when internet is available.
- **Frontline-first**: Tailored for **ASHA/ANM workers**, not just doctors—bridging last-mile access to maternal care, diet, and insurance.
- **Multilingual Voice Support**: Most apps are English/urban-focused. We support **low-literacy users** through voice-first AI in regional languages.
- **Integrated Scheme Recommender**: Helps families discover and apply for **state-specific health schemes**—a major gap in existing tools.

Intended Impact:

- Early detection of high-risk pregnancies and anemia = **reduced maternal mortality**
- Boost in **health scheme enrollment** = reduced out-of-pocket healthcare costs
- **Empowered frontline workers** with AI = scalable public health impact

Who is the primary user of your solution, and explain how your solution will leverage open-source AI to address the aspects mentioned in the Key Design Guidelines.

Primary User:
•ASHA and ANM workers in rural India
•Act as first responders and bridge between underserved communities and formal healthcare

How SwasthyaAI Leverages Open-Source AI (as per Key Design Guidelines):

- **Accessibility:**
  - Voice-first interface in regional languages using **open-source NLP models** (e.g., HuggingFace Transformers)
  - Offline-first design ensures usability in low/no internet zones
- **Inclusivity:**
  - Supports low-literacy users via local-language voice responses
  - Designed for socioeconomically disadvantaged rural populations

- **Transparency & Cost Efficiency:**
  Uses open-source ML frameworks like **PyTorch**, **Scikit-learn**, reducing dependency on proprietary tools
  Transparent model logic (e.g., rule-based eligibility engine)

- **Scalability:**
  Modular architecture enables rapid adaptation to new **states, languages, and health use-cases**
  Built using lightweight, open tools for easy deployment on low-end Android phones

- **User-Centric Design:**
  Frontline-first approach focuses on the workflows of ASHA/ANM, not just doctors
  Designed with continuous feedback from real rural health workers

How is this solution scalable?

Scalability of SwasthyaAI

- **Modular design** allows easy adaptation across states, health domains, and languages
- **Offline-first Progressive Web App** (PWA) ensures functionality in low-connectivity areas
- **Open-source AI models** enable fast training and deployment for new health needs
- **Voice/text interface** supports low-literacy and multilingual populations
- **Sync engine** ensures data is shared once connectivity is available
- **Doctor dashboards** and **API-based architecture** support seamless integration into state-level health systems

List of features offered by the solution

Multilingual Voice/Text Interface
→ Voice assistant for Hindi, Telugu, and other regional languages.

AI Risk Prediction
→ Questionnaire-based maternal & anemia risk analysis

Scheme Eligibility Checker
→ Auto-maps user profile to central/state health schemes

Doctor Teleconsultation
→ Works even in low-bandwidth (2G) areas

Offline-first App Design
→ App works without internet, syncs later

Structured Doctor Dashboard
→ Organized past health records for quick review

What open-source AI tools and technologies will you use to design the solution?

- **HuggingFace Transformers** – For multilingual NLP and voice assistant models.
- **TensorFlow / PyTorch** – For training AI models like risk prediction and eligibility classifiers.
- **spaCy / Stanza** – For Named Entity Recognition (NER) in structuring health reports.
- **DeepSpeech / Whisper (OpenAI)** – For speech-to-text and voice command handling in regional languages.
- **Scikit-learn** – For lightweight ML models (logistic regression, decision trees).
- **Flask / FastAPI** – For building backend APIs and microservices.
- **Firebase (Firestore + Auth)** – For real-time database, syncing, and authentication.
- **IndexedDB** – For offline data storage in web app.
- **Keras** – For prototyping health risk scoring models.
- **Tesseract OCR** – For digitizing paper records if needed (optional module).
- **Label Studio** – For annotating health data for model training.
- **Docker** – For containerized deployment and reproducibility.

Why are these open-source technologies the most appropriate for your solution?

• **Cost-effective**: Open-source tools like HuggingFace, TensorFlow, and PyTorch reduce development costs—ideal for scalable rural health applications.

• **Customizability**: These frameworks allow fine-tuning for regional languages, voice commands, and low-resource AI models.

• **Offline-friendly**: Tools like IndexedDB, Flask, and lightweight ML libraries (e.g., Scikit-learn) help build solutions that work offline-first and sync when internet is available.

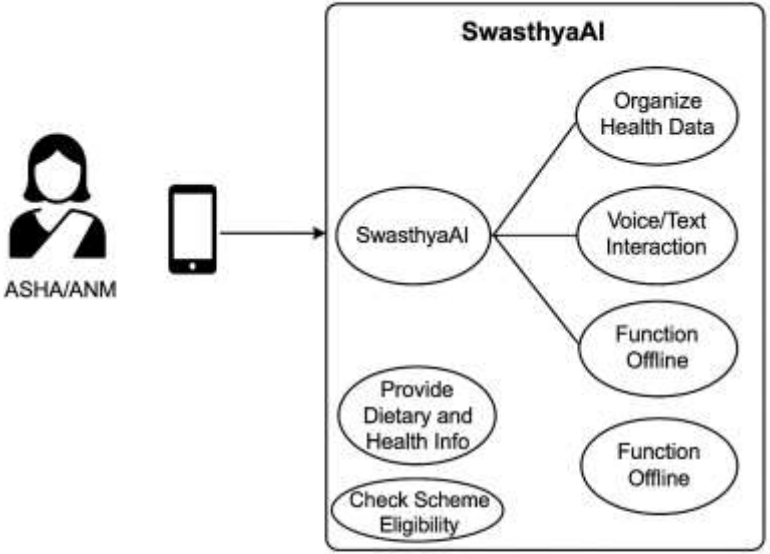
• **Community-driven innovation**: Large communities \are already used in medical annotation and OCR applications worldwide.

•pping

Describe the Solutions Architecture

- Frontend: Android Progressive Web App (PWA)
- Backend: Python Flask APIs
- NLP: HuggingFace transformers for local language understanding
- Storage: Firebase / AWS S3 for patient data
- Sync engine: IndexedDB + Cloud Functions
- AI Models: Trained for risk assessment, scheme ma

Provide a high-level architecture diagram or a use-case diagram of your proposed solution



What datasets will your solution use? Are they publicly available, synthetic, or user-generated?

Datasets Used

•**National Family Health Survey (NFHS-5)**
Publicly available; provides demographic and health indicators for maternal and child health.

•**Health Management Information System (HMIS)**
Government dataset with real-time health data used for model training and evaluation.

•**Public Scheme Eligibility Criteria**
Rule-based information scraped or derived from official portals (PMJAY, state health schemes).

•**Synthetic Datasets**
Generated for testing model behavior in edge cases (e.g., rare maternal complications).

•**User-Generated Data**
Collected via frontline workers during deployment (with consent) — includes vitals, symptoms, and demographics.

Does your solution require cloud-based computation, or can it work with on-device processing? If cloud-based, how do you plan to address connectivity challenges and cost constraints?

Cloud vs On-Device Computation

- **On-Device (Default)**
    - App designed to run AI tasks locally (voice interface, symptom checks, data collection)
    - Works in **low-connectivity or offline** rural areas
    - Data stored locally using IndexedDB until synced
- **Cloud-Based (Selective Use)**
    - For tasks like model updates, scheme database refresh, or doctor dashboards
    - Syncs when internet is available
    - Cloud providers (Firebase/AWS) help scale cost-effectively with usage-based billing
- **Connectivity & Cost Management**
    - Offline-first architecture avoids constant data usage
    - Lightweight models optimized for low RAM devices
    - Only minimal data (e.g., sync logs, alerts) sent to cloud when online

THE/NUDGE INSTITUTE | Meta

Pragati

AI for Impact Hackathon

Powered by H2S

THANK YOU