

**Solution Approach AI-Powered Health Insurance Assistance Platform**

🧩 **Problem Statement**

Millions of Indians struggle to understand health insurance policy terms, locate network hospitals, and file claims — resulting in confusion, claim denials, and erosion of trust in the insurance ecosystem. With a diversity of languages, low digital literacy, and complex insurance jargon, navigating the system is overwhelming for the average citizen. There's a dire need for an accessible, multilingual assistant that empowers users to make sense of their health insurance and confidently utilize it.

**Proposed Solution: AarogyaMitra AI**

**AarogyaMitra AI** is a GenAI-powered, multilingual virtual assistant designed to:

* Simplify insurance jargon into plain language.
* Translate policies into regional Indian languages.
* Answer policy-related questions intelligently.
* Guide users step-by-step in claim filing.
* Assist users in checking policy coverage based on symptoms.
* Be accessible via mobile app, web app, or embedded widget in insurer portals.

By empowering users through conversational AI, AarogyaMitra AI bridges the gap between policy complexity and user understanding.

**🏗️ Architecture Overview**

**🔧 Components:**

1. **GenAI Engine (LLM Layer)**:
   * Powered by LLaMA-3 (Groq via LangChain).
   * Fine-tuned prompts for medical insurance policy question answering.
2. **Vector Database (FAISS)**:
   * Indexes embeddings of insurance policy documents.
   * Enables semantic search for contextual policy Q&A.
3. **Embedding Layer**:
   * BAAI/bge-small-en-v1.5 used via HuggingFace for document embeddings.
   * Multilingual capability planned in Phase 2 using IndicBERT and Langchain's language adapters.
   * Voice input/output in local languages.
4. **Streamlit Interface (MVP)**:
   * Interactive frontend for users to type questions.
   * Answers policy questions, guides claim process, and supports symptom-based queries.
5. **Translation Layer (In progress)**:
   * Use of NLLB or IndicTrans2 for high-accuracy translation to Indian languages.
6. **Claim Assistance Bot Logic**:
   * Step-by-step flow to guide users on claim initiation, documentation, and submission.

**Agentic AI Overview**

The **AI-Powered Health Insurance Assistance Platform** is a multi-agent system designed to simplify health insurance interactions for policyholders. It leverages **Generative AI (GenAI), Optical Character Recognition (OCR), Natural Language Understanding (NLU), and Retrieval-Augmented Generation (RAG)** to provide real-time, multilingual assistance for:

* **Symptom-to-policy coverage mapping**
* **Prescription/lab report diagnosis extraction**
* **Voice-first multilingual support (Hindi, Tamil, Marathi, etc.)**
* **Claim filing guidance**
* **Centralized admin monitoring & feedback-driven improvements**

The system is built on a **fully open-source** tech stack, ensuring scalability, cost-efficiency, and compliance with IRDAI regulations.

**2. AI Agents & Their Roles**

**Agent 1: Symptom-to-Coverage Agent**

**Purpose:** Maps user-reported symptoms to possible ICD-10 diagnoses and checks policy coverage.

**Workflow:**

1. **Input:** User describes symptoms via **text/voice** or uploads a prescription.
2. **Medical LLM Processing:**
   * Uses **Grok (LLMA3.1) / Mistral 3B** to generate ICD-10-compliant diagnoses.
   * Cross-references with **IRDAI policy embeddings** (FAISS vector DB).
3. **Output:**
   * Simple coverage explanation (e.g., *"Your policy covers this under daycare procedures."*)
   * Denial reasoning with policy clause references (if not covered).

**Tech Stack:**

* **GenAI:** Grok / Mistral (local inference)
* **Vector DB:** FAISS (IRDAI policy embeddings)
* **RAG:** LangChain with PyMuPDF for policy PDF parsing

**Agent 2: Policy Translator + QA Agent**

**Purpose:** Extracts diagnoses from prescriptions/lab reports and answers policy-related queries.

**Workflow:**

1. **OCR Processing:**
   * User uploads **PDF/image** of prescription/lab report.
   * **Tesseract/DocTR** extracts text.
2. **GenAI Summarization:**
   * Mistral/Grok summarizes diagnosis in plain language.
3. **Policy Matching:**
   * Checks coverage against insurer-specific rules via RAG.
4. **QA Interface:**
   * Answers follow-up questions in simple English.

**Tech Stack:**

* **OCR:** Tesseract
* **Translation:** IndicTrans2 (if regional language)
* **Document Parsing:** PyMuPDF, pdfminer.six

**Agent 3: Claim Navigation Agent (Voice-First Bharat Assistant)**

**Purpose:** Multilingual voice assistant for claim-related queries.

**Workflow:**

1. **Speech-to-Text:**
   * **Whisper (OpenAI)** converts Hindi/Tamil/Marathi voice to text.
2. **NLU + Policy Retrieval:**
   * LangChain routes query to insurer-specific claim manuals.
3. **Response Generation:**
   * Provides step-by-step claim filing guidance.

**Tech Stack:**

* **STT:** Whisper
* **Translation:** MarianMT
* **NLU:** LangChain + custom intents

**Agent 4: Agent Coordinator / Orchestrator**

**Purpose:** Routes user requests to the appropriate agent and manages workflows.

**Key Functions:**

* Detects input type (**text/voice/upload**) and assigns to relevant agent.
* Maintains session history for contextual follow-ups.
* Ensures seamless handoff between agents (e.g., OCR → Policy QA).

**Tech Stack:**

* **Backend:** FastAPI
* **Orchestration:** LangChain Agents

**Agent 5: Evaluation and Feedback Agent**

**Purpose:** Monitors platform performance and user feedback for continuous improvement.

**Workflow:**

1. **Admin Portal:**
   * **Streamlit dashboard** displays KPIs (coverage success rate, claim resolution time).
2. **Automated Insights:**
   * Alerts on negative trends (e.g., high denials at a specific hospital).
3. **Feedback Loop:**
   * Collects user ratings, retrains models with new insurer policy updates.

**Tech Stack:**

* **Analytics:** Python (Pandas, Plotly)
* **Alerting:** Custom logic + Slack API

**3. Technology Stack**

| **Component** | **Technology** |
| --- | --- |
| **Frontend** | Streamlit (Open Source) |
| **Backend** | Python, FastAPI, LangChain |
| **GenAI Models** | Grok (LLMA3.1), Mistral 3B |
| **OCR** | Tesseract |
| **Speech-to-Text** | Whisper (OpenAI) |
| **Translation** | IndicTrans2 / MarianMT |
| **Vector DB** | FAISS |
| **RAG Layer** | LangChain + IRDAI embeddings |

**4. Key Benefits**

✅ **Automated Policy Interpretation** – Eliminates manual document checks.  
✅ **Voice-First for Bharat** – Supports Hindi, Tamil, Marathi, and more.  
✅ **Real-Time Claim Assistance** – Reduces claim rejection rates.  
✅ **Open-Source & Scalable** – No vendor lock-in, cost-efficient.  
✅ **Admin-Driven Improvements** – Data-backed optimizations.

**5. Future Enhancements**

* **Hospital-Policy Benchmarking** – Compare claim acceptance rates across providers.
* **AI-Powered Dispute Resolution** – Auto-generate appeal letters for denied claims.
* **WhatsApp/Telegram Integration** – Expand accessibility.

**Security & Privacy Considerations**

* **Data anonymization** and no storage of sensitive user info by default.
* **Policy documents processed client-side** or encrypted in transit.
* **Open to integrating with Aadhaar** and IRDAI-compliant APIs in the future.

**Conclusion**

This **AI-powered multi-agent system** revolutionizes health insurance interactions by providing **real-time, multilingual, policy-aware assistance** while ensuring **transparency, compliance, and continuous improvement** through data-driven insights.