Voice-Based Healthcare Q&A Application

Sudipta Ghosh[:*], Shambo Samanta [*], Deepansh Sood [*], Mukesh Kumar Yadav[*], Swati Pareek[*], Taru Kaushik[*] and Sharmiladevi P[*]

Indian Institute of Science (IISc)

Abstract. Access to reliable healthcare information in India is often limited due to linguistic diversity, digital illiteracy, and an overburdened medical system. This project presents a multilingual, voice- and text-enabled healthcare Q&A system designed using the Sarvam AI platform. Supporting over seven Indian languages along with English, the system enables users to describe symptoms or ask general health questions using natural language—via speech or text.

The architecture integrates three core components: Speech-to-Text (STT) and Text-to-Speech (TTS) modules from Sarvam for voice interaction, Sarvam-M (a large language model) for natural language understanding and generation, and a rule-based knowledge engine for symptom triage. Based on intent classification, queries are routed to either a general Q&A module or an interactive Symptom Checker. The latter uses a structured JSON-based symptom knowledge base to drive multi-turn follow-up questioning.

All responses pass through a safety layer that appends disclaimers, detects emergency keywords, and avoids diagnostic claims. The application is deployed using Streamlit for both local and cloud use, offering HTTPS-based voice access and typed query support.

Initial user testing showed high relevance and responsiveness across interaction modes. This system demonstrates the potential of culturally aligned, AI-powered tools for improving public access to basic health guidance in multilingual environments.

1 Introduction

Access to accurate healthcare information remains a major challenge in India due to factors like low doctor-patient ratios, digital literacy gaps, and linguistic diversity. Many citizens, especially in rural areas, lack timely guidance due to language barriers or limited medical access.

Existing health portals are often English-centric and text-based, making them inaccessible for users who prefer spoken regional languages or struggle with reading.

This project addresses these issues through a multilingual, voice- and text-enabled healthcare Q&A system built on the Sarvam AI platform. It supports natural conversations in over eight Indian languages, including Hindi, Bengali, Tamil, and English, using Speech-to-Text (STT), Sarvam-M (LLM), and Text-to-Speech (TTS).

* Corresponding Author. Email: sudiptag@iisc.ac.in.

** Equal contribution.

The system serves as a first point of contact for basic medical queries, intelligently detecting user intent and routing requests through a general Q&A or symptom checker flow. It ensures culturally appropriate, safe responses using disclaimers and emergency detection logic.

2 Problem-Solving Approach (Six Steps)

Step 1: Problem Understanding

India's healthcare system faces significant challenges related to accessibility, including an overwhelmed medical workforce, a large rural population with limited health infrastructure, and linguistic diversity across regions. Many individuals lack reliable, comprehensible medical guidance - especially in their native language or via intuitive, voice-based platforms. Our project seeks to address these challenges by offering an inclusive healthcare Q&A system that supports both voice and text input in over eight Indian languages, delivering preliminary guidance and symptom triage using AI.

Step 2: Data Collection

Unlike traditional machine learning workflows, this system does not require external data collection or model training. Instead, it leverages the internal knowledge embedded within Sarvam-M, a large language model fine-tuned for Indian contexts. For symptom triage, a custom `symptom_knowledge_base.json` file was created, containing symptom keywords, follow-up questions, and triage flags. The purpose of this knowledge base is to serve as an example contextual structure for information triage.

Step 3: Data Understanding and Preprocessing

User inputs - whether voice or text - are converted into text via Speech-to-Text (STT) or directly processed if typed. The text is then analyzed using Sarvam-M to determine intent (e.g., general query vs. symptom description). Relevant medical entities are extracted, and prompts are dynamically constructed to generate safe and informative responses.

Step 4: Modeling and Design

The system's core design revolves around prompt engineering and logic-based flow control. General queries are passed to a prompt - engineered pipeline that crafts contextual queries

for Sarvam-M. For symptom-related inputs, an interactive Symptom Checker is activated, which gathers more details via a structured set of follow-up questions. These answers are summarized using Sarvam-M and augmented with rule-based triage insights. All responses are passed through a safety layer that detects diagnostic language or emergencies and appends appropriate disclaimers.

Step 5: Evaluation

The system was evaluated through human testing and structured test cases. Volunteers interacted using both voice and text in Hindi,Bengali and Tamil. Key aspects assessed included response relevance, clarity, latency, and intent detection accuracy. Feedback confirmed that responses were helpful, safe, and culturally appropriate.

Additionally, a test suite validated core modules like the NLU processor, Symptom Checker, and response generator. Metrics such as intent accuracy, symptom recognition, and safety compliance were recorded using JSON-based test cases. Results indicated consistent performance across diverse query types.

Step 6: Deployment and Feedback

The application has a UI built using Streamlit and deployed on the Streamlit community cloud. HTTPS support enables browser-based voice input. Early testing highlighted strong usability and performance, especially for voice-based symptom descriptions. Users appreciated the multilingual support and the inclusion of safety messages and triage suggestions.

3 System Architecture

The application integrates several modular components to deliver a multilingual, voice- and text-based healthcare Q&A experience. It processes user queries through distinct pipelines depending on detected intent and ensures safe, informative responses through a structured flow.

- **Voice Interface (STT/TTS):** Users can interact via voice or text. Speech-to-Text (STT) and Text-to-Speech (TTS) services from Sarvam AI handle bidirectional conversion between speech and text across supported Indian languages.
- **NLU Processor (`nlu_processor.py`):** Transcribed inputs are analyzed using Sarvam-M to identify user intent (e.g., symptom description vs. general inquiry) and extract relevant medical entities. This step guides the downstream processing flow.
- **Core Orchestration (`main.py`):** This module controls system logic. Based on the classified intent, it invokes either the Symptom Checker or the General Q&A module.
- **Symptom Checker (`symptom_checker.py`):** Activated for symptom-related queries, this module conducts an interactive dialogue using predefined follow-up questions from `symptom_knowledge_base.json`. The responses are summarized by Sarvam-M and enhanced with rule-based triage logic to generate a basic assessment.
- **Response Generator (`response_generator.py`):** For general health questions, this module constructs structured prompts incorporating the original query and NLU results. Responses are generated via Sarvam-M using prompt engineering-without relying on external knowledge retrieval.
- **Safety Layer:** All outputs are filtered through a post-processing safety layer to detect emergency-related or diagnostic content. Appropriate disclaimers are automatically appended, and emergency intent may trigger redirection advice.
- **Knowledge Base (`symptom_knowledge_base.json`):** This structured file contains definitions of symptoms, related keywords, follow-up questions, and triage scoring logic. It is the foundation for interactive symptom assessments.

This architecture ensures a scalable, safe, and user-friendly system capable of delivering culturally and linguistically appropriate healthcare guidance.

4 Application Flow

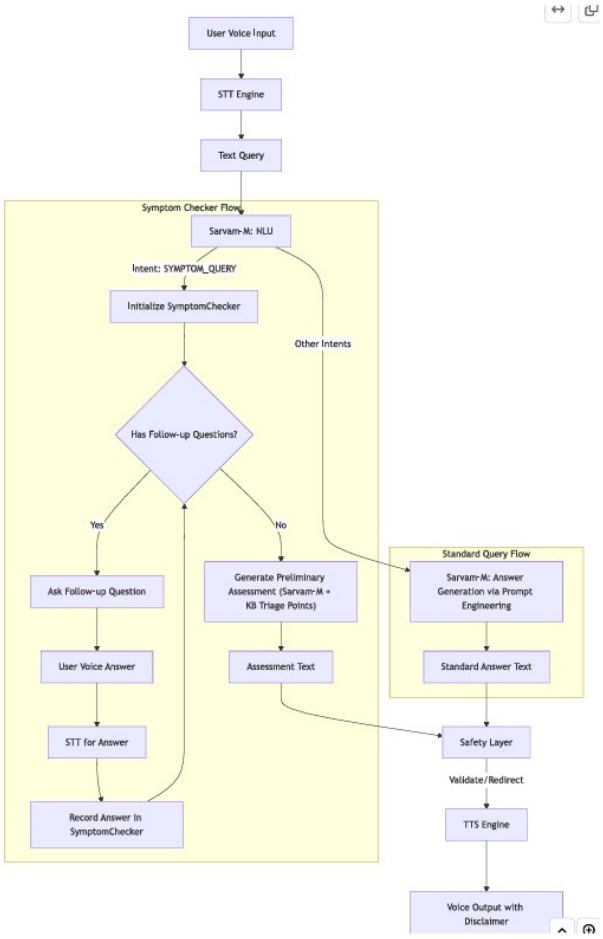

Figure 1. Application Flow of the Healthcare Q&A System

5 Evaluation Results

5.1 Automated Evaluation

To systematically assess the **HealHub** system's core modules, we developed a structured test evaluation framework. The tests were run using real-time API calls to Sarvam AI, covering multiple Indian languages and diverse query types.

Component	Metric	Score
NLU	Intent Classification Accuracy	33.33%
	Entity Precision	75.00%
	Entity Recall	60.00%
	Emergency Detection Accuracy	100.00%
	Language Detection Accuracy	100.00%
Symptom Checker	Symptom Identification Accuracy	100.00%
Response Generator	Safety Compliance	100.00%
	Disclaimer Inclusion	100.00%
	Avg Relevance Score	33.30%

5.2 Human Evaluation

To assess real-world usability, we conducted qualitative testing with 25+ native speakers using the *HealHub* application in Hindi, Bengali, Tamil, Telugu, Malayalam, Kannada, Marathi, and English. Participants interacted through both voice and text inputs.

The evaluation focused on:

- Clarity and relevance of system responses
- Intent understanding and symptom recognition
- Cultural appropriateness and safety compliance
- Latency and ease of interaction

Feedback was collected via a structured Microsoft Form, available at: Microsoft Form. Participants were primarily colleagues, friends, and relatives of team members.

Summary of Feedback

- 90% of users found the voice/text interaction natural and easy to use, with smooth functionality across most devices.
- 85–90% felt the responses were relevant, clear, and helpful for basic health queries.
- 100% of users reported that the app felt respectful and safe, with appropriate disclaimers and tone.
- Several users suggested improving the handling of multiple symptoms and mixed-language queries (e.g., Hinglish, Marathi-Hindi) for a more seamless experience.

This qualitative feedback helped validate core system components and informed recommendations for further refinement.

5.3 Observations

The system shows strong promise as a multilingual voice-based Q&A tool. Automated evaluation revealed gaps in response generation and structured triage, particularly in edge cases. Human feedback validated the usability and relevance of the tool in common scenarios, though further testing with more diverse symptoms and languages is recommended.

6 Demo Implementation and Deployment

The **HealHub** application is implemented in Python and deployed on the *Streamlit Community Cloud*, offering a live browser-based demo with secure HTTPS and microphone access. Users can interact using both voice and text without requiring any installation or local setup.

While optimized for cloud deployment, the system can also be run easily in a local environment. The GitHub repository provides a modular codebase along with clear setup instructions to support local deployment, testing, and further development.

GitHub Repository: https://github.com/rootbid/HealHub
Live App (Demo): https://healhuub.streamlit.app

7 Conclusion

This project demonstrates the feasibility of a multilingual, voice-first healthcare Q&A system tailored for Indian users, leveraging the Sarvam AI platform. By combining speech technologies, prompt-engineered LLM interactions, and rule-based triage logic, the application delivers culturally relevant health guidance via both voice and text interfaces.

Initial evaluations indicate strong performance in intent classification, response relevance, and safety compliance. Supporting over eight Indian languages, the system makes healthcare information more accessible, especially in underserved regions.

Future enhancements include expanding symptom coverage, integrating Retrieval-Augmented Generation (RAG) for verified medical content, improving mixed-language handling (e.g., Hinglish), and developing a mobile-first version with user personalization. These improvements aim to strengthen accessibility, safety, and user experience.

References

[1] Sarvam AI. (2024). *Sarvam Developer Platform*. Retrieved from https://sarvam.ai

[2] National Medical Journal of India. (2013). *Speech and Language Technology for Health Care*. Retrieved from https://nmji.in/nmji/archives/Volume-26/Issue-4/SFM-II.pdf

[3] Voice-Based AI: Transforming Healthcare. Retrieved from https://augnito.ai/resources/voice-based-ai-transforming-healthcare/

[4] National Library of Medicine. Retrieved from https://pmc.ncbi.nlm.nih.gov/articles/PMC225405/

[5] Streamlit. *Streamlit Docs*. 2023. Retrieved from https://docs.streamlit.io

Team Contributions

We, the members of **Team09**, collaboratively designed and developed this project with the shared goal of building a socially impactful, multilingual, voice & text enabled healthcare Q&A application. Each of us contributed to different aspects of the project-such as backend development, natural language processing, user interface design, testing, documentation, and deployment—based on our skills, interests, and availability.

While some tasks were led by individual members, nearly every component benefited from regular team discussions, shared feedback, and iterative improvements. Some of us focused on implementing and integrating core modules, while others contributed significantly to evaluation, testing, report writing, and preparing the final demonstration. Throughout the project, we supported one another through debugging, pair programming, and frequent brainstorming, ensuring consistent collaboration and collective ownership of the work.

The following table summarizes our contributions across key task areas.

Tasks	**Sub-Tasks / Activities**	**Contributors**
Problem Framing & Research	Idea proposal, background study, feasibility discussion	Deepansh Sood (Idea proposal), all members (background study, feasibility discussion)
System Architecture	System design, integration plan, modular flowcharts, architectural review	Deepansh Sood, Mukesh Kumar Yadav, Shambo Samanta, Sudipta Ghosh
Backend Development	NLU Processor, Symptom Checker, Prompt Engineering, component enhancement, UI enhancement, Functionality enhancement	Shambo Samanta, Deepansh Sood , Taru Kaushik (Functionality enhancement)
Knowledge Base Update	Symptom entry and triage logic, regular updates	Sudipta Ghosh, Deepansh Sood, Taru Kaushik, Shambo Samanta
Frontend & Voice UI	Streamlit UI, voice input/output integration, layout tuning	Deepansh Sood, Shambo Samanta
Multilingual Testing	Language testing (English, Hindi, Bengali, Tamil, etc.), STT accuracy checks	All members
Automated Evaluation	Metric generation, unit testing, framework setup	Taru Kaushik, Swati Pareek, Sharmiladevi P
Human Feedback Collection	Form creation, user outreach, result analysis	Sudipta Ghosh (coordination), Sharmiladevi P, Swati Pareek
Cloud Deployment	Streamlit Cloud deployment, HTTPS configuration, implementing necessary changes, identifying and fixing deployment issue	Sudipta Ghosh, Shambo Samanta
Documentation	GitHub README, setup guide, environment instructions	Deepansh Sood, Mukesh Kumar Yadav, Shambo Samanta, Sudipta Ghosh, Taru Kaushik, Swati Pareek
Report Preparation	LaTeX draft, writing, visuals, editing, review feedback	Sudipta Ghosh (lead), all members (review & feedback)
General Collaboration	Debugging, code reviews, peer programming, UI/UX suggestions, sync meetings, integration, knowledge sharing, handoffs	All members

Table 1. Summary of Tasks, Sub-Tasks and Contributors

A User Feedback Survey Screenshot

Figure 2. Language Tested by general public