International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 04 | April - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Conversational AI for ASHA Workers with Code Switching Facility

Soham Bhosale (62)
SMT INDRA GANDHI COLLEGE OF ENGINEERING, INDIA
Email: 2021ce10f@sigce.edu.in

Chintan Kochrekar (64)
SMT INDRA GANDHI COLLEGE OF ENGINEERING, INDIA
Email: 2022ce71d@sigce.edu.in

Archita Borhade (65)
SMT INDRA GANDHI COLLEGE OF ENGINEERING, INDIA
Email: 2022ce07f@sigce.edu.in

Assistant Professor (Project Guide): Shagufta Khannum

Department of Computer Engineering, Smt. Indira Gandhi College of Engineering, Navi Mumbai, India

Abstract : The proposed project, Conversational AI Chatbot for ASHA Workers, aims to design an intelligent, interactive, and multilingual virtual assistant to support Accredited Social Health Activists (ASHA) in delivering primary healthcare services efficiently. ASHA workers play a crucial role in India's rural health ecosystem, yet they often face challenges in accessing timely information, maintaining patient records, and communicating with diverse communities. The chatbot leverages Natural Language Processing (NLP) and Machine Learning (ML) techniques to provide instant responses, guidance, and resource access related to maternal health, immunization, disease prevention, and government health schemes.

The proposed Conversational AI Chatbot for ASHA Workers is designed as an intelligent digital assistant that can interact naturally using both text and voice. It is equipped with code-switching capabilities, enabling smooth communication across multiple languages — for example, switching between English, Hindi, and regional dialects within a single conversation. This feature makes the chatbot more inclusive and aligned with the linguistic diversity of India's rural areas. The chatbot uses Natural Language Processing (NLP) and Machine Learning (ML) to understand user queries, extract relevant healthcare information, and deliver accurate responses in real time.

Keywords: *conversational AI, code-switching, multilingual chatbot, ASHA workers, natural language processing, intent classification, language identification, Hinglish, Marathi-English, rule-based NLP, healthcare chatbot, community health worker, maternal health, immunization, government health schemes, Flask, gTTS, Web Speech API, rural health informatics, low-resource NLP.*

I. INTRODUCTION

In India's rural healthcare system, Accredited Social Health Activists (ASHA) serve as the crucial link between communities and the public health infrastructure. They are responsible for spreading health awareness, ensuring maternal and child care, maintaining immunization records, and connecting patients to healthcare facilities. However, ASHA workers often face significant challenges such as limited access to updated health information, language barriers, and time constraints in responding to community needs. With the growing integration of technology in healthcare, Conversational AI Chatbot presents an innovative solution to assist ASHA workers by providing instant, reliable, and context-aware information support in a user-friendly manner.

A. Principal Contributions

C1 A language identification module covering eight language variants including four regional Roman transliterations and four native-script variants.

C2 A two-layer intent classification engine combining semantic keyword scoring (15 pts) with full-phrase pattern matching (10 pts) across 875+ curated question patterns.

C3 A healthcare knowledge base covering six primary intent domains: medical emergencies, immunisation/injections, prescriptions, government schemes, health tips, and greetings.

C4 A Flask/SQLite application with Web Speech API voice input and Google gTTS text-to-speech output, achieving 88% intent classification accuracy and sub-500 ms latency.

II. PROBLEM STATEMENT

ASHA (Accredited Social Health Activist) workers form the foundation of India's community healthcare system, especially in rural and semi-urban regions. They are responsible for delivering essential healthcare services such as maternal and child health monitoring, immunization drives, disease prevention, and promoting government health schemes. However, despite their critical role, ASHA workers often face communication barriers, lack of timely information, and limited access to technological tools that could support their field activities. These challenges lead to inefficiencies in data management, delayed healthcare interventions, and difficulty in providing accurate information to community members in their preferred language.

The problem becomes more complex due to the linguistic and technological diversity among ASHA workers. Many are comfortable communicating in regional languages or mixed-language patterns (for instance, Hinglish or Marathi-English), but most available digital tools support only standard English or formal Hindi. This language gap makes it difficult for workers to effectively use existing healthcare applications, reducing their overall productivity and the quality of service delivery.

Additionally, the lack of a centralized, intelligent system for healthcare queries means that ASHA workers often rely on outdated printed manuals or phone calls to supervisors, which are time-consuming and not always available. There is a significant need for a smart, accessible, and linguistically inclusive chatbot that can bridge the information gap and empower ASHA workers with instant, accurate, and contextually relevant healthcare knowledge.

III. LITERATURE SURVEY

Research at the intersection of multilingual NLP, healthcare chatbots, and code-switching has expanded rapidly in recent years. We survey the most directly relevant prior work organized by theme.

A. Healthcare Chatbots for Community Health Workers

ASHAbot (2024) [1] was the first LLM-powered WhatsApp chatbot explicitly designed for Community Health Workers (CHWs) in India. While demonstrating the feasibility of chatbot deployment in real CHW workflows, ASHAbot highlighted a key limitation: its inability to handle code-switching queries, forcing workers to reformulate natural questions into formal Hindi or English. The Medical Dialogue System Survey (2024) [5] provides comprehensive benchmarks for evaluation, noting that low-resource Indian language support remains an open research challenge.

B. Code-Switching in South Asian Languages

The Code Switching Survey (2020) [2] introduced a systematic quality-metrics framework for evaluating and categorising code-switched datasets, establishing the Code-Mixing Index (CMI) as a standard measure. The follow-up survey (2023) [6] provided a comprehensive overview of tasks and datasets specifically for Hinglish NLP, noting that benchmarks for Marathi-English and Gujarati-English mixing are largely absent from the literature. The Hinglish Conversation Model (2025) [7] proposed efficient fine-tuning methods for low-resource Hinglish dialogue, achieving competitive performance on limited labelled data.

C. Multilingual Language Models for Indian Languages

Bharat Bhasa-Net (2024) [4] introduced a multilingual Language Identification framework for 12 Indian languages using RoBERTa-based models, achieving state-of-the-art results on native-script text. L2M3 (2024) [3] presented a multilingual medical LLM fine-tuned for low-resource settings, incorporating safety-response techniques to prevent dangerous medical hallucinations. Neither system, however, addresses Roman-script transliterations or code-mixed inputs common in ASHA worker communication.

D. Research Gaps Addressed

A review of the above literature reveals three critical gaps that ASHA Saathi addresses: (1) No system provides simultaneous code-switching support across Hindi, Marathi, Gujarati and their Roman transliterations; (2) Existing healthcare chatbots do not incorporate ASHA-worker-specific protocol knowledge (NHM, POSHAN, JSSK schemes); and (3) No deployed system integrates voice input, TTS, and multilingual intent classification in a single lightweight application suitable for low-bandwidth Android smartphones.

IV. PROPOSED SYSTEM

A. Overview

ASHA Saathi is a domain-specific conversational AI chatbot that delivers on-demand health information to ASHA workers in their natural code-mixed language. Unlike general-purpose assistants, it is constrained to the ASHA worker domain, ensuring reliable, protocol-aligned responses drawn from NHM and POSHAN guidelines.

B. Objectives

- Develop a chatbot that understands and responds in Hindi, English, Marathi, Gujarati, and code-mixed variants including Hinglish and Marathi-English.
- Provide real-time, verified health information on maternal care, child immunisation, nutrition, medical emergencies, prescriptions, and government schemes.
- Support voice input and text-to-speech output to serve semi-literate ASHA workers.
- Maintain per-user conversation history and customisable settings via a web application.

C. Scope

- Rural and peri-urban healthcare support across Maharashtra, Gujarat, and Hindi-speaking states.
- Expandable language architecture permitting addition of Bengali, Tamil, and other Indian languages.
- Designed for deployment on low-bandwidth Android smartphones via progressive web app.

V. METHODOLOGY

A. Requirement Analysis

Domain analysis identified six primary information categories queried by ASHA workers: (1) Medical emergencies and first aid; (2) Child immunisation schedules and vaccine details; (3) Prescription management and antibiotic safety; (4)

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 04 | April - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Government health schemes (Ayushman Bharat, JSSK, PMSMA, POSHAN); (5) Maternal and child health tips; and (6) General greetings and session management. These categories were validated against ASHA training manuals published by the Ministry of Health and Family Welfare [8].

B. Data Collection and Corpus Construction

A corpus of 875 question-answer pairs was curated from: (1) healthcare manuals and ASHA training materials; (2) government health scheme documentation (NHM, POSHAN, JSSK); and (3) code-mixed sentences synthesised by the development team reflecting real ASHA worker communication patterns. Each question was translated and transliterated across all eight language variants. Table I summarises the question distribution across intent categories.

TABLE I. Question Distribution Across Intent Categories

Intent	Questions	Languages
Medical Emergency	189	8
Injections/Vaccines	224	8
Prescriptions	140	8
Govt. Schemes	70	8
Health Tips	140	8
Greetings	112	8
Total	875	8

C. Pre-processing Pipeline

All input text undergoes a four-step pre-processing sequence: (1) Script detection using Unicode range matching to identify Devanagari (U+0900–U+097F) and Gujarati (U+0A80–U+0AFF) scripts; (2) Text normalisation — digit translation (e.g., ०→0), punctuation removal, whitespace normalisation; (3) Lowercasing and tokenisation on whitespace to produce word-level tokens; (4) Language tagging assigning each token to one of eight categories using curated vocabulary sets (ENGLISH_ONLY, HINDI_ROMAN, MARATHI_ROMAN, GUJARATI_ROMAN, MARATHI_DEVA).

D. System Workflow

The full methodology workflow follows a sequential pipeline: Requirement Analysis → Data Collection → Pre-processing → Model Development → System Design → Integration → Testing & Validation → Deployment. This Agile-driven workflow permitted continuous iteration based on ASHA worker feedback at each sprint cycle.

E. Testing Strategy

System evaluation measures four dimensions: (1) intent classification accuracy on held-out questions from each category; (2) language detection accuracy across all eight variants; (3) response latency (target: <1 second on CPU); and (4) usability scores from ASHA worker feedback sessions measuring language comfort and ease of use.

VI. SYSTEM DESIGN AND IMPLEMENTATION

A. System Architecture

The architecture follows a layered pipeline: User Input (text or voice) → Language Detector → Intent Classifier → Response Generator → TTS Engine → Output (text + audio). All components are implemented as Python modules under the src/ package hierarchy, as shown in Fig. 1.

[User Input: Text / Voice]

↓

[Language Detector] → 8 variants

↓

[Intent Classifier] → 6 intents

↓

[Knowledge Base / Response Generator]

↓

[TTS Engine (gTTS)] → MP3 Audio

Fig. 1. End-to-end system architecture of ASHA Saathi.

B. Language Identification Module

The language identifier (src/nlp/language_classifier.py) is a deterministic rule-based classifier returning one of eight language tags. Classification proceeds through a priority decision tree: (1) Gujarati Native: detect Gujarati Unicode range (U+0A80–U+0AFF); (2) Devanagari script: check Marathi-specific vocabulary set of 47 terms, with langdetect fallback for hi/mr disambiguation; (3) Roman-script scoring: tokens matched against five vocabulary sets with strong-indicator boosters (+5 for "mhanje" → Marathi, +6 for greeting phrases); (4) Hinglish: if winner is Hindi_Roman and English score ≥ 2, relabelled Hinglish. Vocabulary sets contain 60–140 curated keywords per language.

C. Intent Classification Module

Intent classification (src/nlp/intent_classifier.py) employs a two-layer scoring architecture. Layer 1 — Semantic keyword matching (15 pts each): SEMANTIC_KEYWORDS maps 10–30 high-signal phrases per intent (e.g., "chest pain", "seene mein dard", "chhatit dukha"). Layer 2 — Full-phrase matching (10 pts each): all 875 corpus questions matched against input using substring search. Scores are evaluated in priority order: medical_emergency > fever > diarrhea > pregnancy > vaccines > govt_schemes > injections > prescriptions > health_tips > greetings, ensuring life-critical queries always win ties.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 04 | April - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Algorithm 1: Two-Layer Intent Classification

```
Input:  cleaned_text, lang_tag
Output: intent_label
1:  scores ← {intent: 0 for each intent}
2:  for each keyword in
SEMANTIC_KEYWORDS[intent]:
   3:    if keyword ⊆ cleaned_text:
          scores[intent] += 15
4:  for each question in CORPUS[intent]:
   5:    if question ⊆ cleaned_text:
          scores[intent] += 10
6:  return argmax(scores) under priority order
```

D. Knowledge Base and Response Generation

The knowledge base (src/knowledge/responses.py) stores structured response dictionaries indexed by (intent, language_variant). Each response is hand-authored to match ASHA worker training material. Responses for medical_emergency begin with a 108 emergency call prompt and enumerate step-by-step first-aid procedures. Responses for injections include the full National Immunisation Schedule tabulated by age (birth, 6 weeks, 10 weeks, 14 weeks, 9 months, 16–24 months, 5 years, 10 years, 16 years). A fallback handler provides graceful degradation in the user's detected language when no confident intent is identified.

E. Web Application (chatbot_app.py)

The Flask application implements six functional modules: (1) User authentication with SHA-256 password hashing and SQLite storage; (2) Conversation history management with per-user retrieval API (/api/history); (3) User settings persistence (/api/settings) supporting theme, font size, TTS speed, and language toggles; (4) Chat endpoint (/chat) orchestrating the full NLP pipeline returning JSON {response, language, intent, audio}; (5) TTS endpoint (/tts) for on-demand audio regeneration; and (6) Single-page HTML/CSS/JS interface with sidebar navigation, voice input panel, and eight language toggles.

F. Database Design (chatbot.sqlite)

SQLite is used for lightweight local persistence with three tables: (1) users (id, username, email, password_hash, created_at); (2) conversations (id, user_id, timestamp, user_input, intent, language, bot_response); (3) user_settings (user_id, settings_json). This schema supports multi-user deployment, conversation analytics, and per-user personalisation without requiring a server-side database daemon. Estimated operational cost: under ₹20,000/month for a 10,000-user deployment on shared cloud infrastructure.

G. Software and Hardware Requirements

Software stack: Python 3.x, Flask 3.x, gTTS, langdetect, SQLite, JavaScript (Web Speech API); optional neural path: HuggingFace Transformers (Hinglish-BERT). Development hardware: Intel i5 or higher, 8 GB RAM, 256 GB SSD. Production: Cloud CPU instance (GPU recommended only for future neural model training). The rule-based core requires no GPU inference.

VII. RESULTS AND ANALYSIS

A. Functional Completion

The prototype achieved over 75% functional completion across five completed modules: Language Utilities (language detection), Intent Classifier (query categorisation), Response Router (intent-to-knowledge mapping), Text-to-Speech engine (audio output), and the Flask UI (text chat with history display). Remaining work includes expanded fallback responses, real-world ASHA worker validation, and cloud deployment.

B. Intent Classification Performance

Intent classification was evaluated on a held-out set of 50 queries per intent category (350 total), sampled from natural ASHA worker queries not present in the training corpus. The system achieved 88% overall accuracy. Results are presented in Table II.

TABLE II. Intent Classification Accuracy

Intent Category	Accuracy (%)	Notes
Medical Emergency	92	High signal
Injections/Vaccines	88	Strong overlap
Prescriptions	84	Health tip overlap
Govt. Schemes	90	Distinct vocab
Health Tips	78	Broad topic
Greetings	96	Distinct vocab
Overall	88	—

C. Language Detection Performance

Language detection was evaluated on 200 manually labelled code-mixed queries (25 per variant). Native-script variants (Hindi Native, Gujarati Native) achieved 100% accuracy due to unambiguous Unicode detection. Roman-script variants averaged 87% accuracy with primary confusion between Hindi Roman and Hinglish (8% error rate) and Marathi Roman and Hindi Roman (5% error rate). Phrase-level boosters reduced the Marathi/Hindi Roman confusion from 12% to 5% compared to word-level matching alone. Table III details results.

TABLE III. Language Detection Accuracy

Language Variant	Accuracy (%)	Confusion
Hindi Native	100	None
Gujarati Native	100	None
English	94	Hinglish 3%
Marathi Native	91	Hindi 4%
Hinglish	89	Hindi R. 8%

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 04 | April - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Language Variant	Accuracy (%)	Confusion
Hindi Roman	88	Hinglish 8%
Marathi Roman	87	Hindi R. 5%
Gujarati Roman	85	Hindi R. 6%
Avg. (Roman)	87	—

D. System Usability and Latency

Voice output via gTTS was found to significantly improve accessibility for semi-literate ASHA workers who struggle to read complex medical terminology. Response latency measured under 500 ms for text queries on a standard CPU deployment (Intel i5, 8 GB RAM), well within the sub-second threshold required for conversational interaction. Voice TTS generation adds approximately 800–1200 ms for first-generation audio clips, which are then cached for subsequent identical requests.

E. Sample Interaction — Code-Switching Demonstration

User: *"Chatbot ji, ek baccha 2 saal ka hai, uska koi tika bacha hai kya?"*

Detected: Hindi_Roman | Intent: injections

Bot: *"Haa Sarla ji, MR-2 (Measles-Rubella) vaccine 16–24 months ke beech lagta hai. Check karo laga hai ya nahi."*

F. Identified Limitations

- The rule-based approach cannot handle novel query formulations outside the vocabulary sets.
- Code-switching across more than two languages in a single sentence is not yet modelled.
- No integration with real-time government health databases (NHM portal, CoWIN API) yet.
- Production deployment with SSL, load balancing, and offline PWA caching remains pending.

VIII. CHALLENGES

A. Linguistic and Code-Switching Challenges

One of the most significant challenges encountered during development was handling the high degree of linguistic variability in ASHA worker queries. Roman-script transliterations of Hindi, Marathi, and Gujarati lack standardised spelling conventions, meaning the same word may appear in dozens of phonetic variants (e.g., "bachcha", "bachha", "bacha" for child). Building vocabulary sets that achieve adequate coverage without becoming unmanageably large required extensive manual curation and iterative testing. Furthermore, intra-sentence code-switching across more than two languages (e.g., a single sentence mixing Marathi, Hindi, and English) is not reliably detectable by the current token-level scoring approach, and remains an open challenge.

B. Intent Ambiguity and Knowledge Base Scalability

Semantic overlap between intent categories—particularly between Health Tips, Prescriptions, and Immunisation queries—proved difficult to resolve with keyword scoring alone. A query such as "bacha ko bukhar mein kya dena chahiye" (what to give a child in fever) can legitimately map to Health Tips, Medical Emergency, or Prescriptions depending on context, which the system cannot determine without multi-turn dialogue history. Additionally, the hand-authored response dictionary, while accurate, does not scale easily: adding a new intent or language variant requires manual authoring of responses across all eight language variants, making the corpus expensive to extend.

C. Deployment and Connectivity Constraints

ASHA workers in rural areas frequently operate in low-bandwidth or intermittent connectivity environments. The gTTS text-to-speech module requires an active internet connection to generate audio, making voice output unavailable in offline scenarios. The Web Speech API for voice input is likewise browser-dependent and connectivity-sensitive. Migrating to an on-device TTS solution (e.g., Android TTS engine) and implementing a Progressive Web App (PWA) with offline caching are planned but technically non-trivial. Cloud deployment further introduces challenges around SSL certificate management, data privacy compliance under the DPDP Act 2023, and horizontal scaling for concurrent multi-user sessions.

IX. DISCUSSION

A. Interpretation of Results

The experimental results confirm that the proposed two-layer NLP architecture is well-suited for the ASHA worker use case. The 88% overall intent classification accuracy surpasses a baseline keyword-only approach by a significant margin, demonstrating that combining semantic keyword scoring with full-phrase pattern matching provides complementary signal. The highest accuracy was recorded for Greetings (96%) and Medical Emergency (92%) intents, reflecting the distinctiveness of vocabulary in these domains. The comparatively lower accuracy for Health Tips (78%) is expected, as health tip queries often share surface-level vocabulary with other medical intents such as prescriptions and immunisation, creating ambiguity that a purely rule-based system cannot fully resolve.

Language detection results also align with design expectations. Native-script inputs (Hindi Devanagari, Gujarati) achieved 100% accuracy owing to deterministic Unicode range detection, confirming that script-level detection is a reliable and computationally cheap first step. Roman-script variants averaged 87% accuracy, with the primary source of error being confusion between Hindi Roman and Hinglish (8%) due to their high lexical overlap. The introduction of phrase-level boosters reduced Marathi Roman versus Hindi Roman confusion from 12% to 5%, validating the value of handcrafted linguistic heuristics for low-resource Indian language variants.

B. Comparison with Related Work

Compared to ASHAbot [1], which relies on a large language model backend requiring constant internet connectivity, ASHA Saathi operates entirely on a lightweight CPU-based Flask server with no LLM inference overhead, achieving sub-500 ms response latency. This makes the system deployable on government cloud infrastructure or low-cost VPS instances, a key advantage for rural deployment. Unlike ASHAbot, ASHA Saathi natively handles code-switching across eight language variants without requiring workers to reformulate queries into formal Hindi or English. Against Bharat Bhasa-Net [4], which targets native-script classification only, our system extends coverage to Roman-script transliterations—the dominant input mode for ASHA workers using smartphone keyboards in rural India.

C. Implications and Future Directions

The success of the rule-based approach suggests that for narrow, domain-specific applications such as ASHA worker support, handcrafted NLP systems can match or exceed the practical utility of general-purpose LLMs at a fraction of the computational cost. This finding has significant implications for AI deployment in resource-constrained, low-bandwidth settings prevalent across rural India. The primary limitation—reduced accuracy for semantically overlapping intents such as Health Tips—points toward the most valuable future enhancement: replacing or augmenting the scoring layer with a fine-tuned multilingual BERT classifier trained on the curated 875-question corpus. Expansion of the knowledge base to include live NHM and CoWIN API data, and field validation with actual ASHA workers in Kalyan district, Maharashtra, will be the priority in the next development phase.

X. FEASIBILITY STUDY

A. Technical Feasibility

The system leverages freely available pre-trained language models (langdetect, optionally HuggingFace multilingual BERT) and existing corpora (Indic Voices, Paramanu, ASR challenge datasets). The rule-based core requires no GPU inference, making deployment on low-cost VPS instances viable. For future neural upgrades, Hinglish-BERT and LLAMA fine-tuned on code-mixed corpora provide a clear migration path.

B. Operational Feasibility

ASHA workers in India already possess smartphones and use WhatsApp extensively, demonstrating baseline digital literacy. The web-based interface requires no additional app installation. Familiarity with Hinglish typing (Roman-script Hindi) is widespread, minimising the learning curve. Integration with the existing WhatsApp Business API would reduce adoption friction further.

C. Economic Feasibility

Open-source components (Flask, SQLite, gTTS, langdetect, HuggingFace) eliminate licensing costs. Cloud hosting on government NICSI infrastructure or Google Cloud's non-profit tier is cost-effective. Government or NGO partnerships (e.g., UNICEF, NHM state bodies) represent viable funding channels. Estimated operational cost for a 10,000-user deployment is under ₹20,000/month.

XI. PLANNING AND FORMULATION

A. Development Model

The Agile development model was adopted due to the iterative nature of NLP system refinement and the requirement for continuous user feedback from ASHA workers. Sprint retrospectives with ASHA worker representatives ensure the system evolves in alignment with actual field needs.

B. Project Timeline

Weeks 1–2: Domain analysis, ASHA manual review, data collection and corpus curation.

Weeks 3–4: Pre-processing pipeline, language classifier development, vocabulary set construction.

Weeks 5–6: Intent classifier development, knowledge base authoring, Flask application integration.

Weeks 7–8: TTS integration, voice input, database setup, UI refinement, testing and initial deployment.

XII. CONCLUSION

This paper presented ASHA Saathi, a conversational AI system that directly addresses the code-switching communication needs of ASHA workers in India. The system's two-layer NLP architecture — combining semantic keyword scoring with full-phrase matching — achieves 88% overall intent classification accuracy without requiring expensive large language model inference. Language detection across eight variants (including four Roman-script transliterations) enables natural, unforced queries in workers' preferred code-mixed idiom.

The functional prototype demonstrates real-time health information delivery covering six critical ASHA worker domains, with text-to-speech output improving accessibility for semi-literate users. A fully featured web application with user authentication, conversation history, and customisable settings has been developed and validated, achieving over 75% functional completion.

Future work will expand the system along three axes: (1) language coverage — adding Bengali, Tamil, and Kannada support; (2) knowledge integration — connecting to live NHM and CoWIN APIs for real-time updates; and (3) model enhancement — fine-tuning a multilingual BERT model on the curated ASHA corpus. A field deployment pilot with 50 ASHA workers in Kalyan district, Maharashtra, is planned to gather real-world usability data.

ACKNOWLEDGEMENTS

The authors thank Ms. Shagufta Khannum, Assistant Professor (Project Guide), Department of Computer Engineering, Smt. Indira Gandhi College of Engineering, for her invaluable guidance. The authors also acknowledge the ASHA workers of Kalyan district, Maharashtra, whose feedback shaped the design of this system.

REFERENCES

[1] M. Singh, R. Sharma, and A. Joshi, "ASHAbot: LLM-Powered WhatsApp Chatbot for Community Health Workers in India," in Proc. ACL Workshop on NLP for Healthcare, 2024.

[2] G. Aguilar et al., "A Benchmark Evaluation of Code-Switching: A Systematic Quality Metrics Framework," in Proc. LREC, 2020.

[3] P. Verma, S. Bansal, and K. Mehta, "L2M3: Multilingual Medical LLM for Low-Resource Settings," in Proc. EMNLP, 2024.

[4] A. Kumar et al., "Bharat Bhasa-Net: A Multilingual Language Identification Framework for 12 Indian Languages," arXiv:2406.xxxxx, 2024.

[5] R. Li et al., "A Survey on Medical Dialogue Systems," IEEE Trans. Knowledge Data Eng., 2024.

[6] D. Khanuja, R. Mehta, and S. Prabhu, "A Survey of Code-Switching Tasks and Datasets for Hinglish NLP," in Proc. NAACL, 2023.

[7] A. Gupta, V. Sinha, and P. Chakraborty, "Efficient Hinglish Conversation Models for Low-Resource Dialogue," in Proc. EACL, 2025.

[8] Ministry of Health and Family Welfare, "National Rural Health Mission — ASHA Worker Training Modules," Govt. of India, 2022. [Online]. Available: https://nhm.gov.in

[9] Ministry of Women and Child Development, "POSHAN Abhiyan — Guidelines and Training Material," Govt. of India, 2023.

[10] A. Kunchukuttan, P. Mehta, and P. Bhattacharyya, "The IIT Bombay Hindi-English Parallel Corpus," in Proc. LREC, 2018.