International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 11 | Nov - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

AarogyaSathi: Design and Preliminary Evaluation of a Multilingual Public Health Information Chatbot

Salman Mohammad Khan, Dhairya Summi, Sidharth Singh, and Aditya Sam Koshy

***Abstract*—AarogyaSathi is a public health information chatbot that our team built in response to Smart India Hackathon problem statement SIH25049. It brings together a curated knowledge base, multilingual retrieval, language model answers, claim verification, emergency phrase rules, and an experimental detector that tracks symptom reports by pincode. Here we describe the version running on Oracle Cloud and ask how far the available tests support it. We read the source code, inspected the live service, reran the repository's component benchmarks, and sent the system synthetic conversations. All 37 automated tests in the 26 October run passed, and the supplied evaluation sets gave 30/30 correct intent labels, 15/15 expected retrieval targets, and 34/34 expected claim verdicts. Training overlap and configuration differences limit the interpretation of these scores. Of the 30 intent questions, 14 also appear in the training data, and the benchmark thresholds differ from the production settings. The conversations turned up problems the benchmarks missed. A sentence saying the user had no chest pain triggered an emergency response, and an Odia question about prevention got no useful answer. Reading the code revealed a further error, in how vaccination ages are converted. The web interface worked, and a subsequent Meta WhatsApp test-number demonstration confirmed replies for greeting, dengue-information, and claim-checking prompts. Integration with official outbreak data remained unverified. Our findings are preliminary engineering evidence about how the implemented response paths behave and where further validation is needed. Clinical effectiveness was not assessed.**

***Index Terms*—public health information; multilingual chatbot; retrieval augmented generation; health misinformation; prototype evaluation**

I. Introduction

People looking for health information online get preventive advice, personal stories, and misleading claims mixed together in the same channels. The World Health Organization (WHO) calls this an infodemic, meaning so much information that reliable guidance becomes hard to pick out [1]. Smart India Hackathon problem SIH25049, set under the Government of Odisha's health-technology theme, asks for an AI-driven chatbot for public health disease awareness [2]. AarogyaSathi is our answer to that brief, with disease and vaccination information, health-claim checking, and accessible conversational interfaces.

The application combines several ways of producing a response. Stored health records handle routine questions, and an intent classifier with entity rules picks out common requests. Multilingual retrieval pulls relevant material and passes it to a language model. Emergency phrases follow a separate path, and a geographic module links symptom events to pincodes. Users reach the system through a browser interface on Oracle Cloud and, in the subsequent channel demonstration, through a Meta WhatsApp test number.

We examine what the available tests establish about the combined system. An earlier audit of symptom checkers found mistakes in both diagnostic suggestions and urgency advice [3]. Those tools are not the same as an educational chatbot, but the audit still makes a useful point. An answer that reads well is not evidence that the guidance is safe. We therefore examine AarogyaSathi at two levels, individual components and whole conversations.

This paper makes three contributions: a description of the deployed architecture, a reproduction of its supplied component benchmarks, and an analysis of the failures we found through synthetic API requests and code inspection. Throughout, we keep two things apart. One is what the code implements, and the other is what we actually saw working in the deployment. We evaluate an existing prototype. We do not propose a new retrieval algorithm, and we do not measure patient outcomes.

II. Background and Related Work

Retrieval-augmented generation pairs a language model with access to outside documents [4]. In AarogyaSathi those documents are local records that supply context for answers on routine health topics. One practical benefit is that the reference collection can be edited without touching the language model. For cross-language retrieval we use the pretrained Language-agnostic BERT Sentence Embedding (LaBSE) transformer, which builds on the multilingual sentence embedding work of Feng et al. [5]. Both LaBSE and the retrieval pattern are existing methods that we apply here, and neither is new to this project.

The evidence on healthcare conversational agents is mixed. Laranjo et al. found limited support for the efficacy and safety of systems that accept unconstrained natural language [6]. Milne-Ives et al. reviewed 31 studies and reported promising results, but also weak evaluation methods and little attention to safety, privacy, and cost effectiveness [7]. These reviews shaped how we report results. Passing software checks and giving good medical answers are different things, so intent classification, record retrieval, and claim-label agreement each get their own measure.

S. M. Khan, D. Summi, and S. Singh are with the Department of Computer Science and Engineering (Data Science), Bhagwan Parshuram Institute of Technology, Delhi, India. This work was carried out under the supervision of Aditya Sam Koshy. Aditya Sam Koshy is with the Department of Computer Science and Engineering, Bhagwan Parshuram Institute of Technology, Delhi, India.

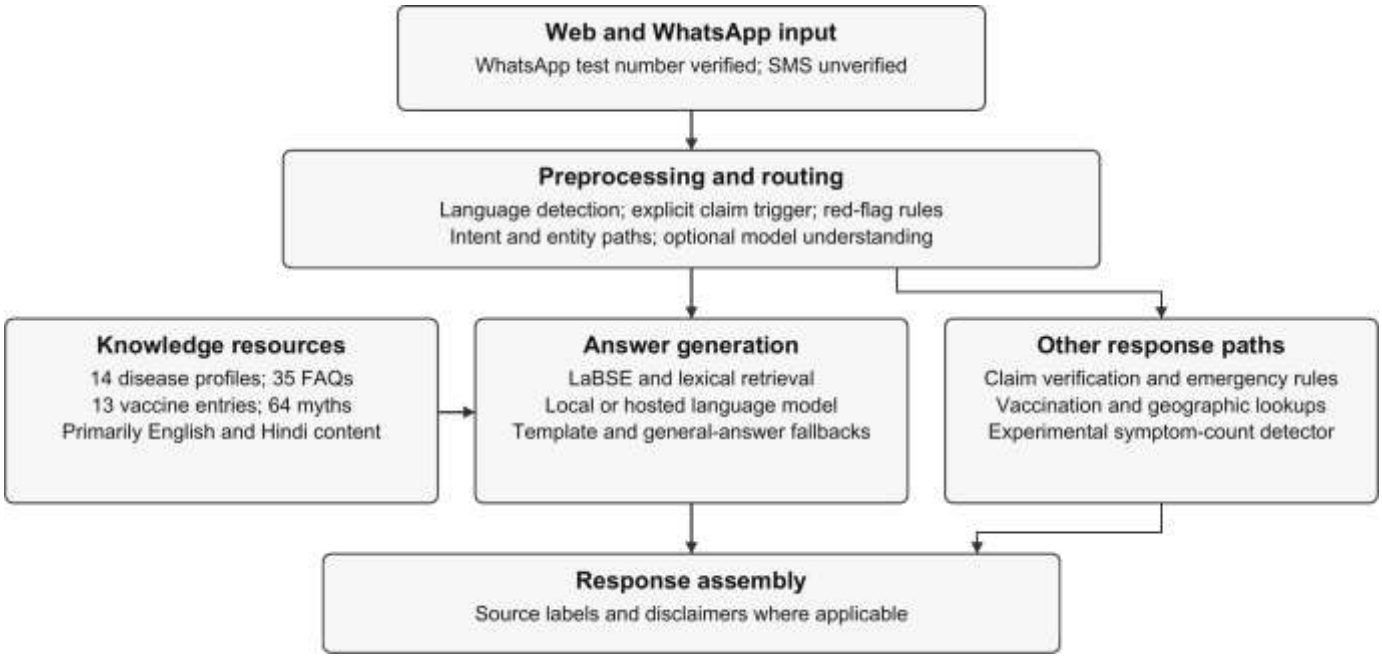


Fig. 1. Implemented architecture of AarogyaSathi. Web access and a Meta WhatsApp test-number exchange were demonstrated; SMS delivery was not verified.

Some requirements depend on outside authorities as much as on software. Vaccination information requires precise age interpretation and an identified version of the official National Immunization Schedule [8]. Outbreak reporting belongs to surveillance systems such as India's Integrated Disease Surveillance Programme [9], and a count of chatbot queries means something quite different from a surveillance report. WHO guidance on AI for health gives us a further reference for thinking about oversight and validation [10]. We use these sources to frame the requirements we examine. They do not certify our implementation.

Nilve, Parihar, and Nichat describe a related chatbot aligned with the Smart India Hackathon, with WhatsApp and web access, multilingual interaction, and outbreak information [11]. The general idea is therefore already established in prior work. We examine a separate deployment, focusing on the gaps between its component scores and live responses. A similar feature list does not mean similar performance, and we do not assume it does.

III. System and Implementation

A. Deployed Architecture

Fig. 1 shows the main response paths. The application is built on FastAPI, which serves both the browser chat interface and the API. Session state supports conversations, a local SQLite database stores aggregate symptom events, and model requests can optionally go through Ollama. The configured local model was Qwen3 (`qwen3:1.7b`), with `gpt-oss:20b` available through Ollama Cloud. A read-only inspection on 29 October 2026 recorded Ollama 0.32.15 and Python 3.12.3. These describe the inspected configuration; a complete immutable environment manifest and hosted model revision were not retained for the earlier requests. When we inspected the deployment on 26 October 2026, the Oracle Cloud virtual machine (VM) was running Ubuntu with 2 OCPUs and roughly 12 GiB of RAM. The healthbot and ollama services were both active, and the API answered on port 8000.

We also checked that the code we reviewed was the code actually running. We compared file hashes between our local project and the VM for the main router, the configuration, the WhatsApp handler, the evaluation labels, and the API entry point, and the selected files matched. The comparison does not cover every dependency, and it says nothing about availability beyond the inspection period.

B. Knowledge and Response Paths

The knowledge base (KB) holds 14 disease profiles, 13 vaccine entries, 35 frequently asked questions (FAQs), and 64 myth entries. Most of the core descriptions are written in English and Hindi. Incoming messages are first checked for explicit claim-verification triggers and emergency phrases. Common requests can then be handled by the intent classifier and entity rules, and a message that neither can resolve may be sent to a language model for intent extraction.

For health questions that need it, retrieved records can be supplied to the model as context for its answer. When the local material is not enough, a general-answer fallback lets the model answer from its own knowledge, so retrieval does not confine every answer to the curated collection. If generation fails, a template can present the stored information instead.

Claim verification runs in its own module. Curated matches and optional online evidence paths produce one of four verdict categories. Some first-aid rules return fixed verdicts. The percentage shown to the user is a heuristic score, and

the score was not externally calibrated in this evaluation. Many knowledge records name a source, yet nothing in the implementation shows sentence-by-sentence checking against the current edition of that source.

C. Retrieval and Routing Configuration

The fallback intent classifier is trained on 289 examples spread across 14 labels. It uses a term frequency–inverse document frequency (TF-IDF) representation built from character-boundary n-grams of length 2–5, with sublinear term frequency and a cap of 30,000 features. On top of that sits a logistic regression model with $C = 8$, the LBFGS solver, and a limit of 1,000 iterations. The implementation uses scikit-learn 1.9.0, as recorded in the 29 October environment inspection. Trigger and entity rules can bypass this classifier or supplement it, which means its standalone score covers only one part of routing. In production, the classifier's confidence threshold is 0.55.

The general retrieval index has 49 records: the 14 disease profiles and the 35 FAQs. Vaccine and myth records are reached through other paths in the application. For embeddings, we take token representations from the pretrained LaBSE transformer [5] (checkpoint `sentence-transformers/LaBSE`, loaded through the Transformers AutoTokenizer and AutoModel interfaces), apply mean pooling weighted by the attention mask, and normalize the resulting vectors. Inputs are truncated to 128 tokens. We describe only the pooling used in our project here; we did not reproduce the original LaBSE experiments. The inspected environment contained Transformers 5.15.1 and PyTorch 2.6.0; the cached checkpoint revision is recorded in the accompanying environment manifest.

For a query q and an indexed record d, the ranking score is

$$S(q, d) = \cos(e_q, e_d) + 0.22\, L(q, d) \tag{1}$$

Here $L(q, d)$ is the fraction of lowercased query terms longer than two characters that appear as substrings of d, and it is zero when no terms qualify. Production retrieval returns up to five records with a score above 0.52. Since the lexical term can push $S(q, d)$ above one, the score serves as a ranking value only. We did not test how much the lexical term contributes on its own.

D. Safety, Geographic Information, and Channels

Emergency phrases bypass language generation and return urgent-care guidance. This gives the application a fast, direct response path, but its lexical matching does not reliably tell a reported symptom from a denied one. We tested it with an affirmative and a negated chest-pain request.

The geographic module connects a small pincode registry to stored advisories and counts symptom events by location. Its bundled outbreak records are explicitly illustrative seasonal material, not a synchronized government feed. A separate detector compares recent submissions with a seven-day baseline. Let $c_{p,s}(t)$ be the count for pincode p and symptom s in the latest 24 hours, and $b_{p,s}(t)$ the mean daily count over the preceding seven complete 24-hour windows. An alert candidate is created when

$$c_{p,s} \geq 6 \ \wedge \ (b_{p,s} < 1 \ \vee \ c_{p,s} \geq 3b_{p,s}) \tag{2}$$

With a baseline of at least one, a candidate is marked high severity when the count reaches five times the baseline, and medium otherwise. If the baseline is below one, the candidate is medium. After an alert is raised for a pincode and symptom pair, further alerts for that pair are suppressed for 24 hours. We have not calibrated any of these constants against epidemiological observations. Submitted symptom events also have no population denominator and can include repeat submissions, so they mean something different from confirmed disease incidence.

Webhook routes for Meta WhatsApp and Twilio are implemented, along with a simulator. Messaging was unverified during the initial inspection. On 31 October 2026, the Oracle service was connected to a Meta WhatsApp test number through an HTTPS tunnel. The callback verified incoming webhook signatures before dispatch. A registered test recipient confirmed replies to a greeting, a dengue-symptoms question, and a garlic-dengue claim check; server logs recorded three incoming messages and three outbound replies accepted by Meta. Five isolated channel tests passed locally and on the VM, covering signature validation, duplicate suppression, and callback filtering. These checks establish basic channel operation for an approved recipient, rather than public availability or clinical answer accuracy.

E. Example Execution Paths

A routine disease question shows how routing and generation fit together. The application first works out the likely intent and disease term, then builds context from the stored disease profile and asks the model for a concise answer. It adds the source names and a medical disclaimer to the result. If the model is unavailable, a template can present the stored profile instead. Even when the right profile is retrieved, the wording the model generates still has to be checked for omissions and unsupported additions.

A message that starts with "fact check," for example "fact check: vaccines cause autism," goes to the verifier. The module compares the claim against curated material and, when it finds no adequate local match, may look for additional evidence. What comes back is a verdict, an explanation, and source information. The percentage that accompanies the verdict comes from rules and model judgments, so it should not be treated as measured certainty. Showing the supporting evidence directly would make the output easier to interpret.

IV. Evaluation Design

We combined four kinds of evidence: source inspection, deployment checks, repository tests, and synthetic conversations. The existing automated tests were run in the VM environment, and we reran `scripts/eval.py`. For the live requests, we posted to `/api/chat` with fresh session identifiers and no pincode. No patient records were used, and no participants were recruited.

TABLE I
REPRODUCIBLE PROTOTYPE CHECKS ON THE DEPLOYED VM.

Measure	Observed result	Important limit
Automated tests	37 passed (24.24 s, 26 Oct)	Checks encoded behavior only
Fallback intent classifier	30/30 exact labels	14/30 inputs duplicate training text
LaBSE retrieval	15/15 expected top result	Known-KB queries; lower evaluation threshold
Claim verdict module	34/34 expected labels; mean 387 ms	24/34 examples are MYTH; developer-selected

TABLE II
SELECTED LIVE API OBSERVATIONS ON 26 OCTOBER 2026. TIMES ARE SINGLE WALL-CLOCK OBSERVATIONS.

Synthetic input	Observed route / outcome	Time
Chest pain and cannot breathe	Emergency response delivered	0.29 s
No chest pain; mild headache	Incorrect emergency response	0.08 s
Fact check: vaccines cause autism	MYTH verdict with explanation	0.15 s
What are dengue symptoms?	Disease information returned	2.32 s
Odia dengue-prevention question	Fallback; no useful prevention answer	11.41 s

The component questions and their expected labels are stored in `data/training/eval.json`, and they were put together during development. Intent accuracy is the share of 30 examples where the predicted class matches the expected one exactly. Retrieval accuracy is the share of 15 questions where the expected item ranks first. Claim-verdict accuracy is exact agreement with the assigned category on 34 claims. Table I lists these checks together with their main limits. The measures address different outputs, so we do not combine them into a single chatbot accuracy.

The evaluation and production settings are not the same. The intent script uses a confidence threshold of 0.45, while production uses 0.55. The retrieval evaluation asks for up to three records above a score of 0.35, whereas production defaults to five records above 0.52. We kept the evaluation settings when reproducing the supplied benchmark and report the differences openly. As a result, the numbers describe the benchmark configuration and not every deployed response path.

We also ran five live synthetic conversations to probe behavior that the benchmark labels do not cover: affirmative chest pain with breathing difficulty, negated chest pain with a mild headache, a prefixed vaccine-autism claim, an English dengue-symptoms question, and an Odia dengue-prevention question. Table II lists them with their outcomes. We picked these as exploratory cases to test particular routes, so they are not a random sample of users. Two additional vaccination checks used “Vaccines for my 2 month old baby” and “Which vaccines are due for a 1 year old child?”. Both followed the vaccination route and declined to supply a verified age-specific schedule (Section V). Each response time we report is a single wall-clock observation, so it reflects the state of the network and the model backend at that moment.

A. Assessment Scope

The component checks tell us whether the software returned the output we expected. They say nothing about whether a generated answer is complete or medically accurate. A retrieval hit can still point to outdated source text, and a correct claim category can accompany an inadequately supported explanation. We read through the live conversations for route selection, relevance, and visible failure. We did not run a blinded clinical rating or an independent annotation study.

We also compared the intent evaluation inputs with the training examples, trimming whitespace and folding case first. The check looks only for exact matches. It tests the evaluation file’s claim to be held out from training, and it does not catch every paraphrase or other dependency between the two sets. Separately, we looked at label coverage and class counts to see what the small benchmark actually contains.

V. RESULTS

All 37 automated tests passed in the 26 October run, and the five channel tests added for the WhatsApp follow-up passed separately on 31 October. The reproduced benchmarks gave the expected label or first-ranked record on every supplied example (Table I). A perfect aggregate on the intent set is not independent evidence of generalization, though. Of the 30 questions, 14 match training inputs exactly, and the set covers 13 of the 14 training labels; smalltalk is the one missing. A separate reproduction on 29 October 2026, using the benchmark confidence threshold of 0.45, classified all 16 questions without an exact training-text match correctly. This development-selected subset was not independently assembled and does not provide an unbiased estimate of generalization. The retrieval set holds five disease targets and ten FAQ targets. The 34 claim labels break down as 24 MYTH, four TRUE, five PARTLY_TRUE, and one UNVERIFIABLE example, which gives us very little evidence about the rarer verdict categories. Both sets were chosen during development, and both results stay tied to the evaluation settings described above.

The live synthetic conversations (Table II) exposed failures that the component scores had missed. The input “I have no chest pain, only a mild headache” triggered an emergency response. Both chest-pain responses also carried snakebite instructions that had nothing to do with the request. The error sits in the lexical safety path, which shows that a fast deterministic response can still be wrong for its context. The English dengue question returned disease information. The Odia prevention question received a fallback with no useful prevention advice. One case cannot give an Odia failure rate. What it does show is a gap between recognizing a language and answering a health question in it. That response also took 11.41 s, the longest of the five observations.

TABLE III
FEATURE STATUS AGAINST THE TEAM'S SIH25049 PROJECT BRIEF: INITIAL INSPECTION ON 26 OCTOBER AND WHATSAPP FOLLOW-UP ON 31 OCTOBER 2026.

Project requirement	Current evidence	Status
Disease and prevention information	14 local disease profiles; English API response checked	Prototype available
Vaccination schedules	13 vaccine records; age conversion defect found	Needs correction
Multilingual access	12 detected/welcome language codes; Odia answer failed	Partial
WhatsApp and SMS reach	WhatsApp test-number replies confirmed; SMS unverified	Restricted demo
Outbreak alerts	Illustrative advisories and count detector	No official live feed
Government database linkage	No verified integration observed in this audit	Unmet

Code inspection found a further defect in vaccination age conversion. An age of 2 months maps to the 9–12 month bucket, and an age of 1 year maps to the 5–6 year bucket. In two additional live vaccination prompts, the model declined to give an age-specific schedule. So we saw the faulty mapping in the code, but we did not see an incorrect schedule in either response. A model refusal can hide a broken deterministic fallback.

A. Alignment with the Original Challenge

Table III separates the functions we found in the implementation from those we saw demonstrated in operation. The health endpoint lists 12 detected or welcome language codes, which is not a measure of answer quality in those languages. WhatsApp replies were subsequently demonstrated with an approved test recipient, while SMS remained unverified and the outbreak records remained illustrative. No integration with a government database was verified during the inspection.

VI. DISCUSSION

A. Interpretation of the Findings

A student prototype can bring local health records, retrieval, rules, and an optional language model together on the inspected VM. The English disease question and the prefixed claim check exercised useful response paths. Those demonstrations have to be read next to the failures, because label agreement gave no hint of the negation error, the irrelevant emergency instructions, or the failed regional-language answer.

The tests sit at different levels. The intent benchmark assesses a classifier and the retrieval benchmark assesses record ranking, while a live request passes through the whole application, where routing rules, thresholds, and fallbacks can change what the user receives. In a health application, a working endpoint and a fluent answer still leave factual support and urgency advice open.

B. Threats to Validity

The evaluation questions were prepared during development, and exact overlap with the training data touches almost half the intent set (14 of 30). Removing those duplicates would not make the set independent, since near paraphrases and familiarity with the knowledge base may remain. The retrieval set is small and the claim categories are uneven, which limits coverage. No separate annotator agreement, calibration study, retrieval ablation, or external multilingual test set was available.

The live synthetic conversations were deliberately chosen examples, not a representative sample. They point to specific faults but cannot tell us how common those faults are or how often an emergency would be missed. Response times come from single requests, so they say nothing about a latency distribution. We inspected availability once and ran no load test or longitudinal uptime study. We draw no inference about clinical effectiveness or population health outcomes.

C. Safety and Deployment Priorities

The faults suggest an order for further work. Vaccination ages need a consistent unit, and the conversion needs tests at schedule boundaries against the current guidance from the Ministry of Health and Family Welfare (MoHFW) [8]. Emergency tests need to separate reported, denied, historical, and uncertain symptoms. The unrelated snakebite content also means we must check how emergency instructions are selected, and not only whether the emergency route was entered.

The general-answer fallback needs its own evidence policy. When local records do not support an answer, the current code may ask the model to answer from its medical knowledge. That behavior should be assessed explicitly and made clear to users; a disclaimer is not evidence for the answer. Content review should also record source URLs, editions or retrieval dates, review dates, and responsibility for updates. These controls fit the oversight concerns raised in WHO guidance [10].

The WhatsApp follow-up added HTTPS transport and signature verification, and removed message text and sender numbers from normal channel information logs. The temporary tunnel and restricted test sender remain demonstration infrastructure. A public pilot requires a stable HTTPS endpoint, a registered production sender, durable message processing, authenticated administrator and database-changing endpoints, and a defined retention policy across the service. Channel verification does not resolve the answer-quality and safety failures identified in the earlier evaluation.

D. Independent Validation

Table IV outlines follow-up checks for the risks we observed. Questions should be written independently of the training examples and the knowledge-base phrasing. A predefined rubric can score relevance, factual support, completeness, source traceability, language clarity, and urgency advice. Clinicians should review the clinical questions and native-language reviewers each language separately, and reviewers

TABLE IV
PROPOSED VALIDATION CHECKS FOR SUBSEQUENT INDEPENDENT EVALUATION.

Risk area	Independent check	Report
Emergency phrases	Clinician-labelled urgent and nonurgent vignettes, including negation	False negatives and false positives by language
Vaccination	Age conversion and answer review against a dated MoHFW schedule	Errors by age unit and schedule milestone
Health facts	Blinded source-supported answer ratings	Support, completeness, and unsafe advice rates
Multilingual answers	Native-language review beyond translated prompts	Per-language comprehension and safety
Outbreak alerts	Historical comparison with confirmed official events	Alert precision, recall, and time lead

should record disagreements without leaning on the system's expected labels.

Table IV is a proposed validation plan, not a set of completed results. Thresholds for acceptable errors would need to be agreed before new outputs are examined. Negation, code mixing, misspellings, forwarded claims, and missing context deserve explicit coverage, since routine wording puts little stress on routing. For geographic alerts, precision, recall, or lead time could be interpreted only against confirmed historical events and a defined surveillance setting. SMS and government-data integration should be reported as demonstrated only after the respective interfaces have been tested.

E. Reproducibility

The repository keeps the supplied evaluation questions and labels, and the inspection recorded deployment resources, threshold differences, and matching hashes for selected files. Even so, the VM ran copied files rather than a recorded release commit, and no complete immutable manifest of dependencies and runtime settings existed. Optional external model services and online evidence results also affect reproduction over time.

A later release should pair the knowledge-base snapshot and model versions with the prompts, a redacted configuration, raw predictions, and the review rubric. Publishing those materials would support stronger comparison than the current development benchmark and keep software behavior distinct from clinically reviewed answer quality. Timestamped responses would also help separate code changes from external service changes.

VII. CONCLUSION

AarogyaSathi is a deployed web prototype with a demonstrated WhatsApp test-number interface that combines curated health information, multilingual retrieval, claim checking, emergency rules, and experimental geographic signals. We reproduced the supplied component benchmarks successfully, but training overlap and configuration differences limit how far they can be interpreted. The live synthetic conversations exposed an emergency-negation error, unrelated emergency advice, and a failed Odia answer, and source inspection found faulty vaccination age mapping. These findings give us concrete priorities for improving the system. The WhatsApp follow-up established basic bidirectional delivery for an approved test recipient. Independent answer review, corrected safety paths, production-channel validation, and verified data integrations are needed before broader claims about public health use can be supported.

DATA AVAILABILITY

The supplied evaluation questions and labels are kept with the project source in `data/training/eval.json`. The live conversations reported here used synthetic prompts only, with no patient records and no recruited participants. No public archive identifier was available at the time of the study.

REFERENCES

[1] World Health Organization et al., "Managing the COVID-19 infodemic: Promoting healthy behaviours and mitigating the harm from misinformation and disinformation," Sep. 23, 2020. [Online]. Available: https://www.who.int/news/item/23-09-2020-managing-the-covid-19-infodemic-promoting-healthy-behaviours-and-mitigating-the-harm-from-misinformation-and-disinformation. Accessed: Oct. 29, 2026.

[2] DRIEMS University, "Smart India Hackathon problem statements," SIH25049, Government of Odisha, 2025. [Online]. Available: https://www.driems.ac.in/wp-content/uploads/2025/09/Smart-India-Hackathon_PROBLEM-STATEMENTS.pdf. Accessed: Oct. 29, 2026.

[3] H. L. Semigran, J. A. Linder, C. Gidengil, and A. Mehrotra, "Evaluation of symptom checkers for self diagnosis and triage: Audit study," BMJ, vol. 351, Art. no. h3480, 2015, doi: 10.1136/bmj.h3480.

[4] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in Advances in Neural Information Processing Systems, vol. 33, 2020. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html

[5] F. Feng, Y. Yang, D. Cer, N. Arivazhagan, and W. Wang, "Language-agnostic BERT sentence embedding," in Proc. ACL, 2022, pp. 878–891, doi: 10.18653/v1/2022.acl-long.62.

[6] L. Laranjo et al., "Conversational agents in healthcare: A systematic review," Journal of the American Medical Informatics Association, vol. 25, no. 9, pp. 1248–1258, 2018, doi: 10.1093/jamia/ocy072.

[7] M. Milne-Ives et al., "The effectiveness of artificial intelligence conversational agents in health care: Systematic review," Journal of Medical Internet Research, vol. 22, no. 10, Art. no. e20346, 2020, doi: 10.2196/20346.

[8] National Health Mission, Ministry of Health and Family Welfare, Government of India, "National Immunization Schedule." Accessed: Oct. 29, 2026. [Online]. Available: https://nhm.gov.in/New_Updates_2018/NHM_Components/Immunization/report/National_%20Immunization_Schedule.pdf

[9] National Centre for Disease Control, Government of India, "Integrated Disease Surveillance Programme." Accessed: Oct. 26, 2026. [Online]. Available: https://ncdc.mohfw.gov.in/includes/About/CentresAndDivision/IDSP.php

[10] World Health Organization, Ethics and Governance of Artificial Intelligence for Health: Guidance on Large Multi-Modal Models, 2024, ISBN 978-92-4-008475-9. [Online]. Available: https://www.who.int/publications/i/item/9789240084759. Accessed: Oct. 29, 2026.

[11] H. Nilve, K. Parihar, and A. Nichat, "AI-driven public health chatbot for disease awareness," International Journal of Engineering Research & Technology, vol. 15, no. 3, Art. no. IJERTV15IS031825, 2026. [Online]. Available: https://www.ijert.org/research/ai-driven-public-health-chatbot-for-disease-awareness-IJERTV15IS031825.pdf. Accessed: Oct. 29, 2026.