Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

AI-Driven Public Health Chatbot for Disease Awareness

Harsh Nilve
School of Engineering
Ajeenkya DY Patil University

Kanhaiya Parihar
School of Engineering
Ajeenkya DY Patil University

Prof. Amit Nichat
Department of Computer Science
Ajeenkya DY Patil University

***Abstract* - This paper is about creating an AI-driven multilingual health chatbot targeting rural/semi-urban population where it will deliver a WhatsApp based chatbot integrated with features like text, image processing and voice chat. It uses the official data for processing by using LLMs, APIs and gathering data from government sites [5]. The project is implemented using hybrid architecture, using self-hosted Ollama LLM for model and using cloud API provided by services like Gemini [2]. This chatbot supports more than 5 regional languages making it user friendly even for non-English speakers by using the API [3]. The main objective for this chatbot is to provide disease info, vaccination schedules, scheme eligibility, etc. So, addressing the gap in health-related issues in the community is what this project is trying to solve.**

***Index Terms*—AI chatbot, public health, WhatsApp, multilingual, disease awareness, LLM, rural healthcare, telemedicine**

I. INTRODUCTION

68% Indian Population rural and semi-urban regions, accessing health information urgently is almost impossible, not every village has clinics which leads to limited healthcare facility, so if anyone wants to check out their symptoms at that moment then it's hard to the people in rural areas to use the latest technology like AI, Google to understand their symptoms better, because those advance technology is pretty much hard to understand by the regular rural people, that's why our chatbot is WhatsApp based, nowadays with enhancement of technology every person has WhatsApp installed in their devices which exceeds 500 million installation in India, WhatsApp is easy to use and easy to understand [1]; this project also includes the feature of image processing and voice chat, where if anyone wants to post picture of for example any rashes or any physical symptoms then this chatbot will answer that, same goes for voice chat if the person is not literate enough to chat then they can just send a voice chat to this chatbot. Here there is integration with government health APIs with hybrid with other APIs and Cloud LLM models hence guaranteeing the accuracy [4]. Here the symptoms are predicted but still we have to visit Doctor's to have proper checkups.

That's one of the problem this chatbot solves, this chatbot will have multiple options including the first one i.e. The symptom checker, then The Scheme Finder- where the user would have to put up the necessary information to check the eligibility of schemes and then the scheme finder will guide the user to apply for the same on Government websites [5]. Then the Vaccination center checker, where when the user will put up his address or the pin code then our chatbot will show nearby hospital to that location, helping them find nearby hospital on the same chatbot interface [6]. Then the last one is Outbreaks, here the latest outbreaks would be shown, like for example outbreaks like flu, dengue, etc. [7].

II. LITERATURE REVIEW

A. AI in Healthcare Chatbots

The integration of artificial intelligence into healthcare delivery has accelerated significantly over the past decade, with medical chatbots emerging as a prominent application. These systems leverage LLMs, Datasets, APIs to support preliminary diagnosis, enabling patients to describe symptoms in conversational language and receive structured guidance before engaging formal clinical services [1]. This capability reduces friction in the patient journey and holds particular promise for triaging non-emergency cases.

LLMs (Large Language Models) is now expanding into healthcare, where its making people accessibility to healthcare information more easily available over text/voice-chat/photo processing [2]. These systems reduce the cost of healthcare access while providing continuous 24/7 availability that improves patient engagement and adherence.

B. Multilingual Healthcare Systems

One of the major problems is language barriers when it comes to understanding the complexity of any technology is it's not available in regional language [3], here India is known for being multilingual, where each state has its own language hence this chatbot supports more than 5 languages on website which would help users to understand each core of the service provided by the chatbot easily in their own language for better understanding and on WhatsApp chatbot with the help of Gemini model it supports almost every language for chats [8]. Hence language barrier would be overcome helping each individual understand healthcare awareness in their own language. For websites individual pages for each language are made to get high accuracy over the translated text. We want our chatbot to deliver output with high accuracy so using other methods to translate the text would have been a problem, one wrong translated text can lead to many problems, so to solve this problem making different UI files for each language is

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

preferred. For now, the chatbot has 5 languages and over the time it would have more languages.

C. Research Gaps

WhatsApp has now entered with AI technology and has its own meta AI, where it's not personalized only for health-care information, there are other still other chatbot based on WhatsApp where healthcare is still not properly implemented and not available focusing the rural and semi-urban areas, not just healthcare information but other information like about vaccination centres and lacking many services what this chatbot provides to the user, proper information about available schemes provided by the Government and even if they want to check if they are eligible for those schemes then even for that there's no technology on WhatsApp they will provide all those information. There is minimal integration where modern tech is integrated with reliable government health APIs [4]. For low literacy rate voice chat or image input can be seen less where they are an important considering not all people can ask queries through texts.

III. METHODOLOGY

Due to restrictions by the Laws like HIPAA, GDPR, DPDP Act 2023 (India), IT Act 2000 (India), ISO 27001 compromising the patient healthcare data is strictly prohibited, failing to do so would strictly result in severe legal consequences. Therefore, instead of relying on private clinical data, the system is designed to use publicly available and government-authorized datasets to ensure both compliance and reliability [5]. While, including fines and sanctions. Hence, due to this collecting the data from hospitals/clinics nearby to use it to train model was not possible. These datasets include sources such as government health platforms, vaccination APIs, and outbreak monitoring systems, which provide structured and trustworthy data [6] [7]. That was the restriction that was faced while collecting the data, so other methods were chosen, using Gemini model and medical model hosted on Ollama that would be hosted on cloud for Disease Outbreak Monitoring, collecting data from available Government websites to collect the data and then using it by cleaning the data for Available Schemes [5]. Using APIs like Google News for Current outbreaks and Google Maps APIs to locate Vaccination centers. These APIs enable real-time data fetching, allowing the system to provide up-to-date information such as nearby vaccination centers and current disease outbreaks [4].

These challenges made us to explore more and more available options which led us to face the complexity of collecting data and then finding the alternative to get same type of result, this will make sure that delivering awareness to public is done with the latest and trustworthy data. Each service that is provided by the chatbot uses different way of data collection and different way of using that data and providing the right outcome for that service and option. The overall workflow of the system involves receiving user input through the WhatsApp Cloud API, processing it via a Node.js-based backend, and generating responses using AI models and external data sources [2]. All user data is handled securely and stored in a controlled environment, ensuring compliance with data protection regulations while maintaining system performance and scalability.

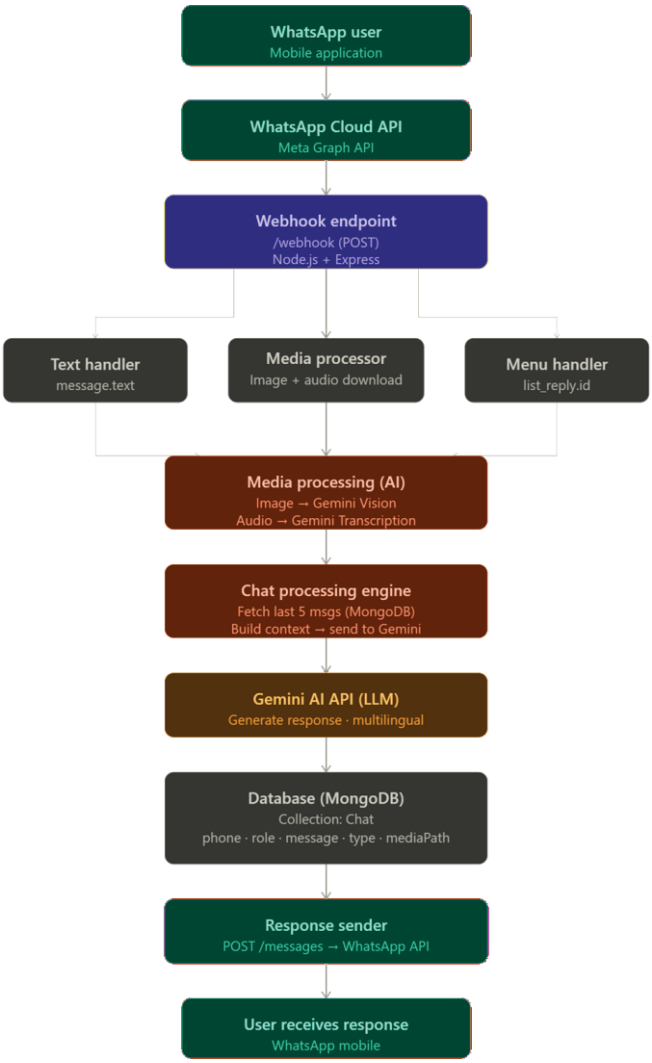

Fig. 1. System Architecture – WhatsApp Chatbot Approach

IV. SYSTEM ARCHITECTURE

Fig. 1 explains the overall proposed system architecture for the proposed WhatsApp Chatbot health assistance platform, where user can interact with the chatbot on WhatsApp and on web interface for the same.

The user interface layer consists of web application and the WhatsApp chatbot interface. Where user can interact with the system with such interactions, chatbot specifically made for the rural population considering the literacy rate and considering the exposure to the technology, whereas the web interface is made considering the semi-urban or urban population, where its more interactive and with more information available there. To connect with the WhatsApp, this chatbot uses WhatsApp Cloud API, where it helps backend connect to WhatsApp

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

using this API, so that all the services and option can be provided into the WhatsApp chat directly. WhatsApp provides this API to the business so that they can use it send and receive messages at scale without any need to host their own servers or to use any third-party Business Solution Providers (BSPs). For now, this chatbot is developed on Development (Sandbox / Test Mode) where meta provides a temporary test phone number, so can be used for testing. But for Production meta requests to have business to have their own business phone number, where meta verifies it in Meta Business Manager.

The Webhook is implemented using Node.js and Express, which function as both the application server and REST API gateway. In this layer it processes the incoming requests like auth service, chat service, prediction and alert endpoints, where it would route the users request to the appropriate services in the backend. Then there are three handlers, text handler to handle all the text receive, media processor to handle and process the received text and image and then the menu handler to handle the menu tabs for interactive menu on list_reply.id. The service implements the core business logic for the system, where it includes all the services like auth service for authentication, chat service for WhatsApp chatbot interactions, Prediction service to predict with the symptoms provided by the user, alert service to find the outgoing outbreaks and other services for other health related information.

The Media processing by AI is responsible for the disease prediction, where it uses multi model techniques, making use of Google Gemini AI model where image is processed by Gemini Vision and audio is processed by Gemini Transcription [2], and also using health models on Self-hosted OLLAMA on cloud with to run LLMs locally, models here are restricted to give health information and to reduce the hallucinations.

While the data the user is providing to the chatbot is safely stored in the MongoDB Database which is the primary database, Chats would be fetched onto the web interface UI showing history chats so user can go back to the past chats. The chat can also help us train our own LLM with the real user's data, where it would be stored securely and help us train the LLM so that we could use our model instead for more personalized prediction. The chat history would be shown only on Website while WhatsApp already stores the chat history on its own application.

This data would be responded and would be sent as Post method to the user with help of the same WhatsApp Cloud API, then the user receives the message and response on the same chat.

The website approach in Fig. 2 is designed keeping the semi-urban and urban population in mind, where the user interacts through a proper web interface rather than WhatsApp. When the user submits any report or inputs data through the website UI, it hits the Backend API which is built on Node.js, this layer handles all the incoming requests and routes them to the right place. All this data that comes in gets stored in MongoDB, which is the primary database of the system, it keeps everything safe and structured for further processing.

Fig. 2. System Architecture – Web-Based Monitoring Approach

Once the data is stored, the Aggregation Engine kicks in, where it groups and summarizes the data based on things like location, symptom type, or time period, basically it cleans and organizes the raw data to make it useful.

Then comes the Pattern Detection layer, this is where the spike logic runs, for example if suddenly 50 people from the same pin code report similar symptoms within 24 hours, that's a spike, and the system catches it automatically. When a spike is detected, the Alert System triggers, it doesn't wait for someone to manually check the dashboard, it fires automatically the moment a threshold is breached. The Notification layer then pushes this alert out, either as a WhatsApp message to the concerned health authority or as an in-app push notification, making sure the right people are informed immediately [7].

Finally, everything gets visualized on the Dashboard, where admins and health workers can see real-time charts, trends, and heatmaps showing what's happening, where it's happening, and how fast it's spreading.

One thing that is important here is that both the WhatsApp chatbot and the website are not two separate systems running independently, they are actually connected to the same back-end and the same MongoDB database, so whatever the user

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

does on WhatsApp, the history and the data of that would also be visible on the website when they login, this makes it very convenient because the user does not have to start fresh every time they switch between the two interfaces. Also since both are connected to the same database, the outbreak alerts and the vaccination reminders would work across both platforms at the same time.

Another thing worth mentioning here is the security layer of the overall system, so when a user sends any data whether it is a text, image or voice, none of that is stored in plain format, the sensitive fields are encrypted before storing into MongoDB, and the communication between the frontend and the backend happens over HTTPS so there is no chance of the data being intercepted in between. The WhatsApp Cloud API itself is also end-to-end encrypted by Meta so the messages are already protected at that level too, this was one of the main reasons why WhatsApp was chosen as the primary interface because it already has a strong security layer built in and the rural users would not have to worry about anything, they are already comfortable using WhatsApp for daily communication so the trust is already there.

V. PROPOSED SYSTEM FEATURES

A. Disease Prediction Module

This module helps user predict the diseases where they have to put the input as texts, voice and images. It would be easy for people where they would just send photo to the chatbot for example if there are any rashes or any visible symptoms sent as a photo then the chatbot would predict what might be the case here, same goes for voice chat as text however the user is comfortable explaining their symptoms this chatbot helps them accordingly. It uses AI-ML based prediction where this uses the agents like Google Gemini API or other LLM hosted on Ollama [2], then the user data would be saved into the MongoDB database to train a new LLM and with that using already trained healthcare model that can also be hosted on Ollama which would be more personalized and would follow the health-related Laws more securely with laws like HIPAA, GDPR, DPDP Act 2023 (India), IT Act 2000 (India), ISO 27001, also the same data would be used and showed into history chats on website. This chatbot also has context-awareness factor, where the main goal is making the chatbot remember the context of the whole chat where it would need to remember all the history of the conversation happened in that chat.

Note: Here the primary objective of the proposed chatbot is to assist users by predicting based on the symptoms provided by the user to promote early awareness and preventive health-care practices, so that healthcare tips would be available over the finger-tips and easily available for the users who are in underserved regions.

It is important to note that the chatbot does not provide medical diagnoses or replaces professional medical consulting. The prediction generated by the AI system is solely made for the educational purpose. Users are strongly encouraged to seek qualified professional help for accurate diagnosis and treatment.

B. Government Scheme Finder and Eligibility Checker

This module helps user to find the available schemes by checking the eligibility and guide them to the official Government website and the process [5]. Schemes like Mahatma Jyotirao Phule Jan Arogya Yojana (MJPJAY), often combined with Ayushman Bharat (AB-PMJAY), is Maharashtra's primary health scheme. So, the goal for the chatbot scheme module is where the user will input things like age, income or any other category to find out the available schemes provided by the government and then the official link guiding to the main process. Government schemes have so many benefits to the people, it can cover a lot of cost where they are to help people financially. Here government dataset would be used and then converting that dataset into csv and then it would be fetched accordingly. Detailed information of each scheme would be collected by following this method instead of relying on LLMs or Gemini models, because the scheme data needs to be accurate, their website, their eligibility and all the other options need to be extremely accurate and hence each data has been collected – transferred into csv and then it is fetched from there.

C. Vaccination Center Locator

The vaccination center locator module enables user to find nearby vaccination centers using pincode or address [6]. This is integrated with the Google Maps API integrated with the system, where it would reveal the vaccination center nearby by filtering the geographical areas. The user just has to input the pincode and then the chatbot will locate the nearby centers, when it comes to chatbot it sends the available centers as a text but whereas in the website it can be located on maps directly making it more flexible to the users. It sorts the centers on distance-based location, where it will show the centers nearby and the centers available in that particular pincode. It fetches the real-time data updating to the time and availability of the vaccines on that center.

D. Disease Outbreak Monitoring

The outbreak module monitors to utilize the official disease surveillance data like for example, along with the environment location-based risk [7]. Here the chatbot will provide the on-going outbreak which is happening at that time, and will also filter it out based on the geolocation, for example if city Pune is having flu outbreak. This will help to spread awareness in public by providing up-to-date information in users vicinity and what preventive measures can be done to avoid the outbreak. On the chatbot this would be either available as the option or it will be auto sent to public to keep them up-to-date, but for the website which has a web interface the outbreaks would be presented as scrolling news ticker format, which would be displaying real-time alerts. This approach ensures high visibility without disrupting the user's experience.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

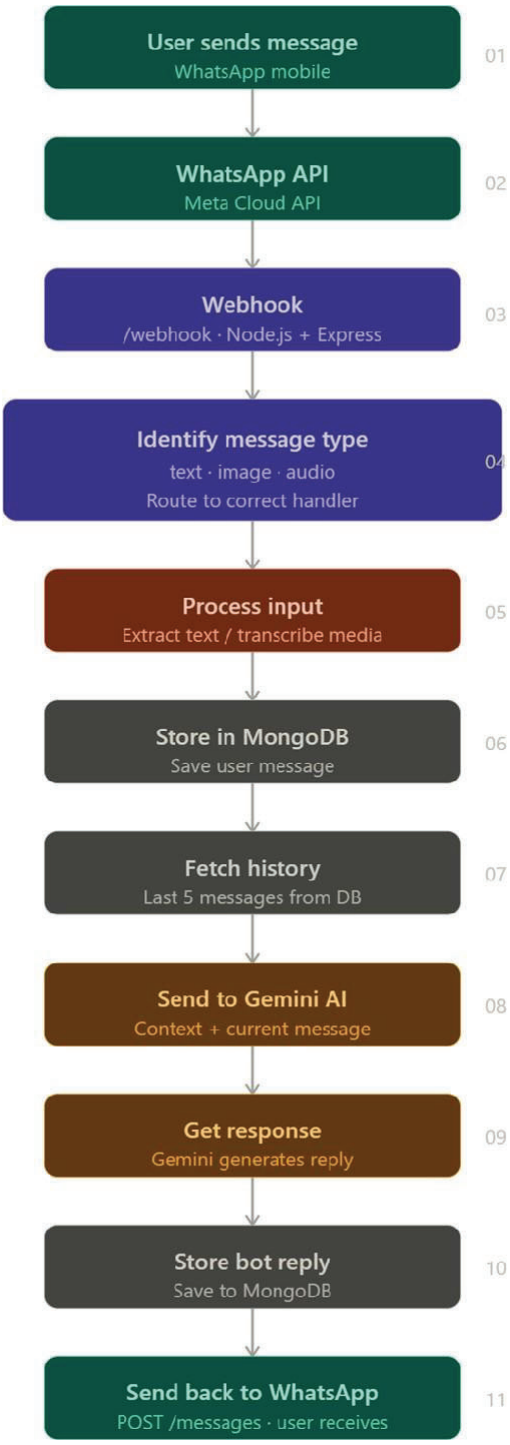

Fig. 3. WhatsApp Chatbot: Message Processing Flow

VI. DATA-FLOW PIPELINES

A. WhatsApp Dataflow Pipeline

On the chat interface of WhatsApp when the user sends a message on WhatsApp the message first goes and hits the WhatsApp Cloud API provided by Meta, then through that it gets forwarded to the backend through a Webhook endpoint built on Node.js and Express, this would be the entry point for the entire chatbot system where every single message that comes will have to pass through here.

Once the message is received by the webhook, here it identifies what type of message it is, if text, image or audio message, this is important because each type is handled differently, for example if it's an image then it goes through Gemini Vision to extract what's in the image [2], if it's audio then it gets transcribed, and if it's plain text then it's directly ready to be processed. The full message processing flow is shown in Fig. 3.

After identifying and processing the input, the message gets stored in the database MongoDB, so nothing is lost and everything is tracked – the extracted text from chat, transcribed text extracted from audio and then extracted details from the image. For WhatsApp, it already has its own storage so if user wants to check their past messages, then they can scroll up to check on the WhatsApp chat interface. For context this data can be used to train a new LLM but mostly used for website chat history. Then the data stored here in MongoDB is sent to Gemini AI for context building so that Gemini can remember past context of chat history and then it will help to answer the user with old context + new chat, so that user won't have to start again explaining everything from the beginning.

That response then gets saved back into MongoDB as the bot's reply, maintaining the full conversation history, and finally it is sent back to the user through the WhatsApp API, completing the full cycle from the moment the user typed something to the moment they receive a reply. Then this message is sent to the user on the WhatsApp application chat and then repeating the cycle for every text.

B. Each Feature Dataflow Pipeline

These features are explained here into five sections, in Fig. 4 for login and register, for WhatsApp user is already logged in into WhatsApp with their phone number where WhatsApp does its own authentication, for website user needs to first register and by using the same credentials the user can login into the website again. Here the user can see their previous history of chats and other options they have used, this data would be stored into MongoDB in encrypted form where no one can see username and their password, while logging in the user data would be fetched out from MongoDB.

For vaccination centers data module, the user would have to visit vaccination center page where they would fetch out nearby vaccination centers where they will have a map they can see to checkout the location directly on map and then the data would be presented below the map where user can have detailed information about the centers, here the Google Maps API has been used to locate the locations and their data [4], that page also provides the detailed updated vaccine requirement based on the age, gender and conditions [6].

The next module, searching the government schemes the data has been collected from government datasets [5], collected into csv and then stored into the MongoDB and then

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

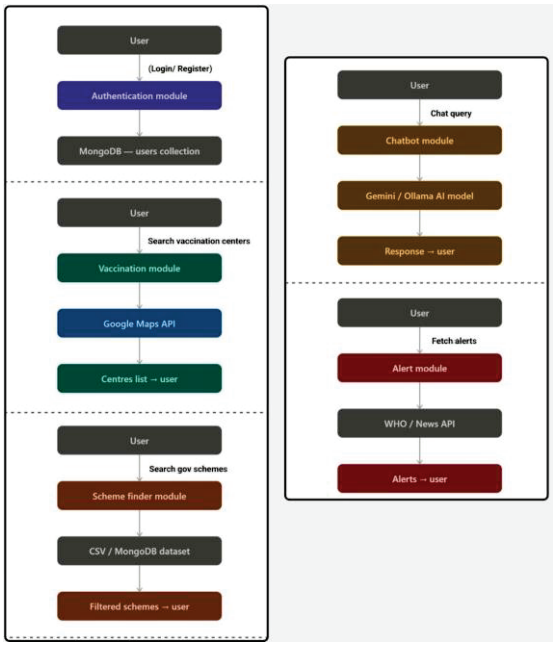

Fig. 4. Feature Dataflow Pipeline Overview

fetched when the data is requested, here the chatbot does not provide schemes data generated from LLMs and Gemini AI, rather it uses this data from trustful sources which are checked and then stored into the Database; the data needs to be without any error and should provide correct rightful information that Government of India provides.

The chat query module specially used for disease prediction when the user sends chat then that query gets to the LLMs, either Gemini API is used or other healthcare models are used which would be hosted on OLLAMA where they would be hosted on cloud infrastructure like Amazon Web Services [2], here the data would be sent to these models and then the generated answer would be sent to the user.

The last module fetches the alert, by using News API or WHO the correct outbreak which might be global or regional would be sent on WhatsApp chats for the chatbot and on website it would be notified on the homepage itself [7], so the awareness would be created within users so that they can take enough precautions for those outbreaks. By taking enough precautions the viral would spread less into the public and then public might overcome it before the viral reaching its critical conditions.

VII. ENTITY-RELATIONSHIP DIAGRAM

The ER diagram in Fig. 5 shows how the main data of the system is structured and connected with each other. Here there are main four tables in our MongoDB database which are user, chat_logs, reminders, alerts. So, in these four tables the user table is the central one, because everything is linked to the user. It stores basic data like name, email, password hash, phone number, role, language, and whether the user is active or not. Each user also has a unique ID (uuid) and timestamp for when

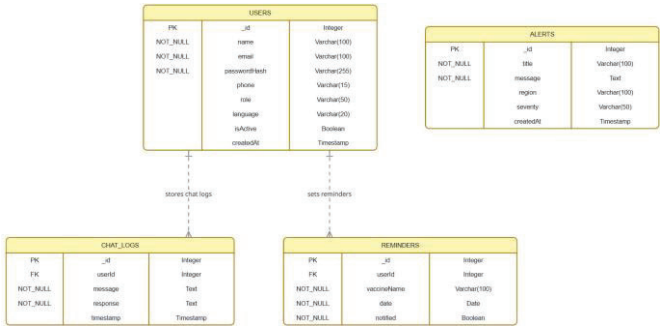

Fig. 5. Entity-Relationship Diagram of the Proposed Healthcare Chatbot System

the account was created. Since the chatbot needs to support multiple languages and personalized interaction, storing these details becomes important [8].

Now coming to Chat Logs, here it stores the conversation between the user and the chatbot each and every conversation from both the ends i.e. the data sent from the user and the data generated by the system is saved here along with a timestamp. Like the data like text data, voice data which would be transcribed into text i.e. string, image data where the data would be stored. This is linked to the Users table using userId, so each chat can be traced back to a specific user so this helps in maintaining chat history and also useful if we want to improve the model later using real interactions.

Then there is the Reminders table, which is mainly used for vaccination reminders. Each reminder is connected to a user, and it stores details like vaccine name, date, and whether the user has been notified or not. This feature is useful for keeping track of important health-related schedules and making sure the user doesn't miss them [6].

Lastly, the Alerts table stores information about disease outbreaks or important health alerts. It contains fields like title, message, region, severity, and created timestamp. Unlike other tables, this is more system-generated and not directly linked to a specific user, since alerts are generally broadcasted to multiple users based on location or relevance [7]. So whenever there are outbreaks or viral, the data would be stored here and the awareness would be shared from here, including precautions that the user can take.

Overall, the relationships are quite simple: one user can have multiple chat logs and multiple reminders, which creates a one-to-many relationship. This design keeps the data organized and makes it easy to scale the system when more users and features are added.

VIII. DEPLOYMENT WORKFLOW

The deployment architecture goes in two ways parallel but both end up to the same database, website architecture flow it is deployed on cloud services that is the AWS (Amazon Web Services). It follows a multi-machine architecture where the load would be planned around two virtual machines so that if one virtual machine goes down there would be another virtual

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

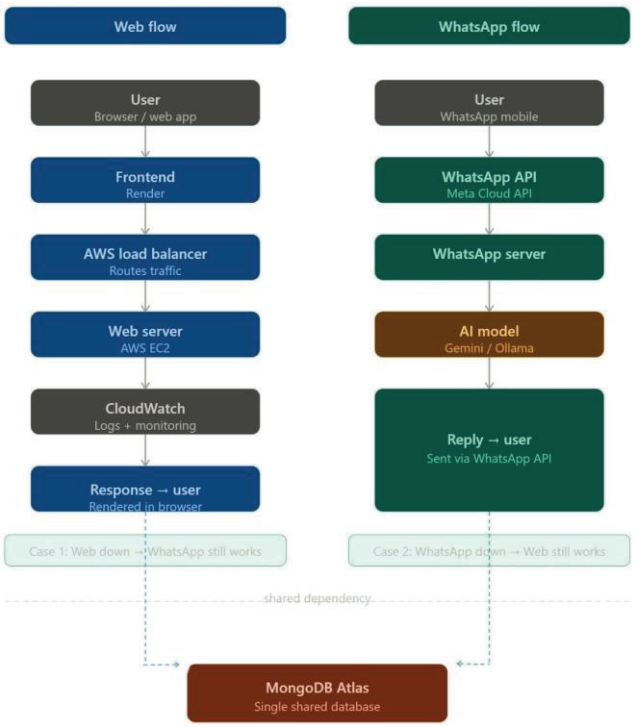

Fig. 6. Deployment Workflow – Both WhatsApp and Website

and the traffic would be controlled by AWS load balancer, so the number of virtual machines would be auto scaled so when the traffic would increase the virtual machines would go up in number and when traffic would decrease again the virtual machine number will come back to 2, thanks to auto scaling and load balancing the traffic would be handled here by using the native AWS services. For the security, AWS makes sure that there are less attacks, are protected public keys and private keys to get access to the virtual machines and about the other AWS configuration so if we want to watch traffic, logs and want to monitor it then we can use the native service that is the CloudWatch service, then the response would be generated and will go down to the database. Here we are using SAAS MongoDB database, so it already has high level security and MongoDB itself makes sure that no one can get pass through the security.

The full deployment flow is illustrated in Fig. 6. For the WhatsApp deployment workflow, would be connected through the WhatsApp Cloud API, and with help of the WhatsApp Cloud API – backend connects to the WhatsApp so that it could interact with users and can send message to the users through WhatsApp, so this chatbot does not need any third-party services for backend to interact with users, using the same API service. The response would be generated using AI model and whenever the user will text, it will go back to the LLM and the answer would be generated again [2], all this conversation of system that is the generated by the system and user would be stored into the same SAAS MongoDB database.

Hence with this availability architecture if the WhatsApp goes down then the website would be still functioning and there are chances for the website to go down due to autoscaling, and even if somehow the website goes down then there would be still WhatsApp functioning.

IX. FUTURE WORK

So right now the chatbot is working on a development/sandbox environment where it is being tested with a temporary phone number provided by Meta, the next step would be to move it to production where a proper verified business phone number would be registered and the chatbot would be available to actual users. That is the most immediate thing that needs to be done.

Apart from that, one of the bigger things that we want to look into is making the chatbot work without internet, because the whole point of this chatbot is to help rural population but the reality is that a lot of villages in India still don't have proper internet connectivity or the data is very limited, so if the chatbot can work over SMS then it would reach even more people who can't access WhatsApp, the idea here is to integrate an SMS gateway service like Twilio or MSG91 so that users can just send a plain text SMS with their symptoms or query and the backend would process it the same way it processes a WhatsApp message and send back the response as an SMS, no internet needed on the user's end at all.

Then for the disease prediction part, right now it depends on Gemini API and OLLAMA hosted healthcare models, but over time as the chatbot collects more and more real user interaction data which is stored in MongoDB, that data can be used to fine-tune and train our own custom LLM which would be more personalized specifically for Indian health conditions and the kind of symptoms that rural population mostly faces, this is one of the bigger goals and would take time but the groundwork for it is already being laid by storing each conversation.

Also on the language side, right now 5 regional languages are supported on the website and Gemini handles most languages on WhatsApp, but adding more dedicated language UI pages especially for languages like Tamil, Bengali, Telugu, Odia would make the chatbot reach even more people. India has more than 20 scheduled languages and ideally each of them should have a proper interface.

One more thing that can be looked at in the future is integrating with the Aarogya Setu API and the ABHA health ID system, because these are the government's own digital health infrastructure and if this chatbot could connect to those then the users would not have to enter their details again and again, their health ID would already have their history and the chatbot could give even more personalized responses based on that.

X. CONCLUSION

As discussed in the Future Work section, the system has several planned improvements including production deployment, ASHA worker dashboard, custom LLM training, and

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

deeper government API integration. In conclusion, this system uses the hybrid architecture using both WhatsApp approach and website interface. For that users can use it comfortably like rural population can use WhatsApp for more friendly use and semi-urban or urban population can use anything as they are comfortable with. It provides with multiple services like disease prediction, government scheme finder and eligibility checker, vaccination centre locator, and diseases outbreak monitoring with outbreaks awareness and prevention [1] [2]. Since for many people it is hard to understand English so this chatbot has more than five regional languages [3] [8] and when it comes to the complexity this chatbot has options where the user can send data as per the comfortability like in text, image and voice.

This uses source from database of government [5], APIs [4], cloud-based models and Gemini AI [2] helping the system by making it more reliable and up to date. At the same time the system has a proper architecture to store the data in a secure way. This project does not aim to replace doctors or medical consultation, but it can work as a first step for awareness, guidance, and early support. Overall, this chatbot can be a helpful digital tool for improving health information reach, especially in areas where medical guidance is not easily available.

REFERENCES

[1] S. Kumar et al., “An AI-Based Medical Chatbot Model for Infectious Disease Prediction,” *IEEE Access*, vol. 10, pp. 128278–128289, 2022.
[2] R. Chen and M. Zhang, “An LLM-Based Multimodal Chatbot for Preliminary Diagnosis,” *IEEE Access*, vol. 13, pp. 45623–45635, 2025.
[3] P. Desai et al., “Multilingual Healthcare Chatbot Using Machine Learning,” in *Proc. 2nd Int. Conf. Emerging Technology (INCET)*, Belagavi, India, 2021, pp. 1–6.
[4] Office of the National Coordinator for Health Information Technology, “Digital Health Company Experiences Using EHR APIs,” HealthIT.gov, Jan. 2026. [Online]. Available: https://healthit.gov/blog/research-and-scientific-advancement/digital-health-company-experiences-using-ehr-apis/
[5] National Health Portal India, “Open Government Data Platform,” data.gov.in, 2025. [Online]. Available: https://data.gov.in/
[6] Ministry of Health and Family Welfare, “CoWIN: COVID-19 Vaccine Intelligence Network,” cowin.gov.in, 2024.
[7] National Centre for Disease Control, “Integrated Disease Surveillance Programme,” ncdc.gov.in, 2025.
[8] Ministry of Electronics and IT, “Bhashini: National Language Translation Mission,” bhashini.gov.in, 2024.