Volume 2 Issue 2 (June 2026)
DOI: https://doi.org/10.64189/ict.26309
J. Inf. Commun. Technol. Algorithms, Syst. Appl., 2026, **2**(2), 26309

Research Article

OPEN ACCESS

An AI-Based Menu-Driven Healthcare Chatbot for Underserved Communities

Sayyed Mohammed Abbas,* Amaan Shaikh, Shrushti Sapte, Zara Shaikh, Yumna Qureshi and Awij Shaikh

Department of Electronics and Telecommunication, Anjuman-I-Islam's M.H. Saboo Siddik College of Engineering, Byculla, Maharashtra, 400008, India

*Corresponding Author
Sayyed Mohammed Abbas
abbas.241644.et@mhssce.ac.in

Received: 04 May 2026
Revised: 21 June 2026
Accepted: 29 June 2026
Published Online: 30 June 2026.

Citation

S. M. Abbas, A. Shaikh, S. Sapte, Z. Shaikh, Y. Qureshi, A. Shaikh, An AI-Based Menu-Driven Healthcare Chatbot for Underserved Communities, *Journal of Information and Communications Technology: Algorithms, Systems and Applications,* 2026, 2(2), 26309, https://doi.org/10.64189/ict.26309.

Abstract

Language barriers, low levels of digital literacy, and inadequate medical resources limit people's access to quality healthcare information, particularly in rural and semi-urban areas. In this research study, we present SehatBot, an AI-driven healthcare chatbot that utilizes the WhatsApp Business platform as its delivery vehicle. The goal of SehatBot is to provide multilingual, structured healthcare services in developing regions. Users can ask questions about symptoms they are experiencing, view vaccination schedules, and access health alerts issued by local governments through a user-friendly messaging interface. SehatBot was developed using Flask as the backend framework with a modular architecture. Additionally, it uses webhook technology to communicate with the Meta WhatsApp Business Cloud API, eliminating the need for users to install any additional applications. To provide reliable and context-aware responses, a hybrid NLP approach has been implemented by combining rule-based methods, a menu-driven finite-state mechanism, and optional machine learning-based intent classification. The healthcare knowledge base is stored in JSON format and managed using PostgreSQL in the production environment and SQLite in the development environment. Preliminary evaluations were conducted with 10–15 participants, who reported that SehatBot operated reliably, effectively supported multiple languages, and responded within 0.5 to 1 second. These findings suggest that SehatBot provides a scalable solution for improving health awareness, reducing misinformation, and increasing access to trusted healthcare information in underserved communities.

Keywords: Healthcare chatbot; WhatsApp business API; Natural language processing; Multilingual healthcare; Underserved communities.

1. Introduction

The lack of access to well-organized and trustworthy healthcare information remains a significant challenge in rural and semi-rural areas due to limited healthcare services, language diversity, and varying levels of digital literacy. [1] Although digital health technologies have improved the delivery of healthcare information in many urban areas, their benefits have not been distributed equitably.[1] However, the widespread adoption of smartphones and improved mobile internet connectivity have created new opportunities to deliver healthcare information through interactive messaging platforms, particularly in developing regions. Chatbots, which are conversational interfaces integrated into commonly used messaging applications, enable users to interact directly with healthcare systems without requiring advanced technical skills. As a result, they offer an accessible and user-friendly solution for delivering reliable healthcare information to underserved communities.[1,2] Early chatbot systems were primarily based on rule-based logic. These systems used predefined scripts and keyword matching to generate responses.[2-4] While they were predictable and reliable, they lacked the flexibility to handle the wide variety of user inputs. Advances in Natural Language Processing (NLP) have enabled more sophisticated conversational interactions by incorporating intent recognition and contextual awareness.[5] However, deploying such systems in resource-constrained environments remains challenging.[1] Open-ended conversational models may produce unclear or unreliable responses, which is a significant concern in healthcare applications where accuracy, clarity, and reliability are essential. Furthermore, limited support for multilingual communication and unstructured conversation flows can reduce usability for users with low digital literacy, potentially leading to confusion and the spread of misinformation.

Despite significant advances in chatbot technology, relatively few healthcare chatbot systems successfully integrate controlled interaction, multilingual support, and adaptability to resource-constrained environments. Existing solutions often prioritize conversational capabilities at the expense of response consistency, while others fail to address the accessibility needs of underserved populations. Consequently, there is a need for a solution that balances the delivery of structured and reliable healthcare information with ease of use, ensuring that healthcare communication is accurate, accessible, and inclusive for all users. To address these challenges, this research develops and evaluates SehatBot, an AI-powered hybrid conversational chatbot designed specifically for the WhatsApp messaging platform. SehatBot employs a menu-driven, rule-based finite-state architecture, chosen for its simplicity, ease of implementation, and high usability among users. It also selectively incorporates AI-based intent analysis to provide greater flexibility in interpreting user inputs. This hybrid design combines a structured healthcare knowledge base with intelligent intent recognition to ensure the consistent, accurate, and clear dissemination of healthcare information through a scalable and adaptable system. The feasibility and effectiveness of the proposed SehatBot are evaluated through a controlled pilot study conducted at the prototype level. This research contributes to the development of a multilingual healthcare interaction framework, a structured healthcare knowledge organization strategy, and a messaging-based deployment model designed for low-bandwidth and resource-constrained environments. The study focuses on the conceptual design, system architecture, implementation, and prototype evaluation of the proposed chatbot. Consequently, the system is not intended to perform clinical diagnoses, provide medical prescriptions, or initiate emergency interventions.[6] Furthermore, the proposed solution is designed exclusively for deployment on the WhatsApp messaging platform and does not include integration with other digital communication channels. Due to the exploratory nature of this research, the study is limited to prototype-level evaluation and does not include large-scale field deployment or statistical benchmarking of the system's performance.

2. Materials and methods

The proposed SehatBot system will be a modular, messaging-based healthcare chatbot that can provide multilingual medical information via a structured conversational interface. The architecture of the system is layered in separate modules or layers so each layer can be developed independently, scaled as needed, and maintained separately. These include Messaging interface, Processing unit, Knowledge base, Data storage layer. The backend of the system will utilize Python programming language and the Flask web framework to handle incoming http request(s) and webhook communication.[7] Incoming user messages from the WhatsApp Business API will be accepted via secure POST request.[8] Those POST requests will be passed and sent to the processing layer.[9] For development & testing purposes, Ngrok will be used to expose the local Flask server to the internet, allowing real-time webhook communication. However, in a production-oriented setup, deployment will be configured on an Ubuntu-based virtual private server (VPS), where NGINX will act as a reverse proxy and Gunicorn will manage application execution.[10,11] In addition to configuration of the VPS, Docker containerization will be utilized to ensure portability of the application across multiple environments and consistent deployment across multiple

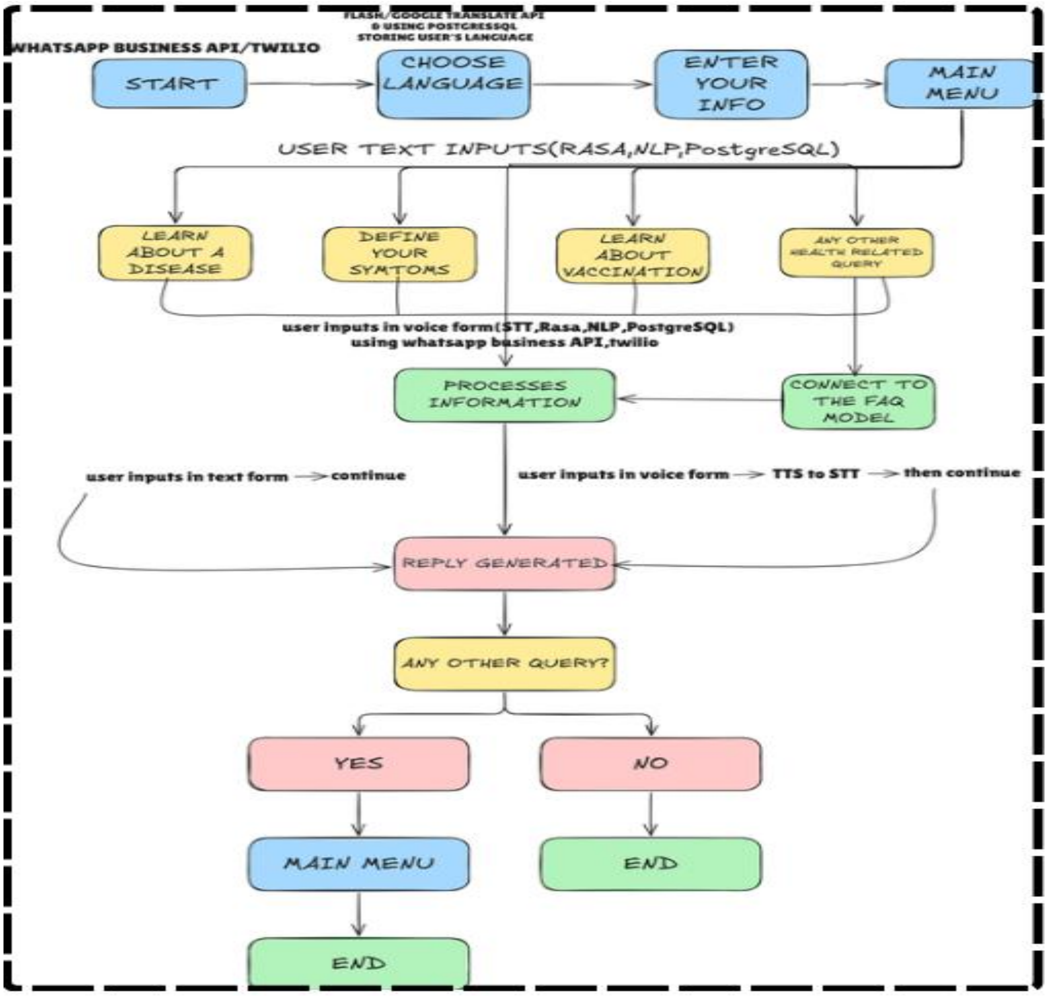

Fig.1: Technical flow chart of the system.

platforms.[12] The messaging layer is responsible for integrating with the WhatsApp Business API, which provides encrypted communication using HTTPS & JSON payloads. It handles only message transmission & reception, separated from internal processing logic. Once the received message is processed by the core processing module, it performs intent recognition, state tracking, and response selection.

2.1 System workflow representation

In order to give a holistic overview of the interaction pipeline, the overall operational workflow of the chatbot is depicted in Fig. 1. Fig. 1 gives a representation of the stages of processing in a sequential manner from the user initiation to the response generation and then to the continuation of the session. Instead of giving a description of the technologies, the workflow is centered on transitions between user interaction and response generation.

2.1.1 Explanation

User input either via text or voice is routed through a central processing unit. Voice input is converted to text by Speech-To-Text (STT), whereas text input is sent through to analysis. To generate the appropriate response path, a finite state machine (FSM) is utilized to control conversation flow; it will ensure that users advance through a series of steps in a logical sequence that includes collecting input, analyzing input and generating a response. [2,13] Once the relevant information has been retrieved, the response generation process utilizes contextually tracked state along with knowledge retrieval to format the response into readable sections in the form of structured messages suited for messaging interfaces. Upon providing a response, the system presents options to allow users to proceed with their interaction or end the interaction thus controlling the conversational loop. The Natural Language Processing (NLP) component of the system employs a hybrid method that combines rule-based methodology with NLP based on models. Rule-based methodologies are employed when there are predefined questions/queries and when consistency in medical responses is required, whereas NLP models can be applied to more general user input to analyze them. In this manner, the system maintains both reliability and flexibility in its usage scenarios. Multilingual support is implemented within the system utilizing a translation layer. Internal processing is conducted utilizing English, while all user input from other languages is translated prior to processing and then translated back after processing to provide the final response. Pretranslated static content related to critical medical information are stored to ensure accurate translations; however, dynamic conversational components are translated in real-time. Thus, this hybrid translation approach strikes balance between efficiency and reliability.

2.2 Knowledge base representation

The knowledge base uses Hierarchical JSON Schemas. Each disease entry in the schema includes attributes for symptoms, cause, preventative measure, and treatment guidance. The hierarchical structure enables efficient retrieval and consistent formatting of responses during interactions. When developing the knowledge base, SQLite was utilized.[14] The prototype interface was developed using a WhatsApp-based interface that provides an easy to follow structured and user-friendly interaction process; including language selection and then a menu driven navigation process. The design minimizes user confusion, especially among those who are new to chatbots. Manual testing evaluates multilingual responses, and how the application handles different user scenarios, including state transitions. The end-to-end testing tests the entire interaction pipeline from when the message is received to when it is processed and delivered back to the user. Using third-party tools like PostMan to perform API testing verifies proper webhook communication, as well as the format and structure of payloads. Using unit testing on each Flask route ensures the applications can correctly handle requests. For production purposes, we would use PostgreSQL (a large-scale relational database) for managing our larger dataset and supporting JSON based queries.[15] For our application we will store temporary conversation states in memory and record all interactions logged into our database for further analysis and improving our overall system.

2.3 Prototype interface and user interaction flow

The prototype has some functional capabilities; for example, it enables users to retrieve data about diseases by entering the name of a disease and obtain a category-based list of symptoms associated with the entered disease, also preventive measures associated with the entered disease, and general health related information for the entered disease. Each response is broken up into smaller sections that are easily distinguishable from one another so they are easy to read on a limited screen like those found on mobile devices or messaging platforms. The prototype also demonstrates the ability of an integrated system (for example it uses state management, multi-lingual processing, and database driven information) which is a good indicator of whether the system design will be effective in a deployed environment. In addition, this prototype includes actual images of the screens used during interactions with the system, which provide concrete evidence that the prototype was tested in a real-world deployed environment versus simulated testing, therefore increasing the amount of confidence that other developers would have in implementing the same system design.

3. Results and discussion

This pilot test had about 10-15 individuals from the development team. Functional correctness, multi-language capabilities to respond to user queries, menu-based navigation and the ability of the system to operate stably and perform reliably under normal operating conditions were tested. Realtime testing occurred using a smartphone device as the client (via WhatsApp) and the server was running in development mode. Automated benchmarking tool(s) were not used – manual observation records were kept. Working Prototype of the System is shown in Fig. 2.

The system performed well in all tests that were run. There were no navigational "dead ends" in the workflow of core functions such as accessing disease information and checking symptoms. The Webhook communication pipeline successfully sent and received messages (requests & responses) via the WhatsApp Business API. Average response time per query ranged approximately 0.5 seconds when there were minimal network delays, which is comparable to near real-time interaction that can be practically used.[16] Testing for multiple languages was done in separate sessions. When an individual chose a language for their session, that language choice continued for the entire length of their session. There were no apparent problems with grammar or semantics related to how the system responded back to users. A hybrid translation method was employed where pre-translated medical content was provided with some degree of dynamically translated content, and it served its purpose to provide clarity and reliability while minimizing potential problems with full automation translation. On average, users could find what they needed by asking questions and receiving answers in 2 to 3 iterations. Because the finite state machine-based design of the system defined the order in which things should happen as part of a conversation and/or transaction, it helped create a structure and direction to conversations thereby reducing confusion for users with less experience with digital technology.[1] No failures due to loss of connectivity were experienced during testing but no extreme load testing or very large scale testing has been completed at this point. In terms of being flexible yet consistent, the Hybrid Conversational Model worked well because it allowed for rule-based logic to generate

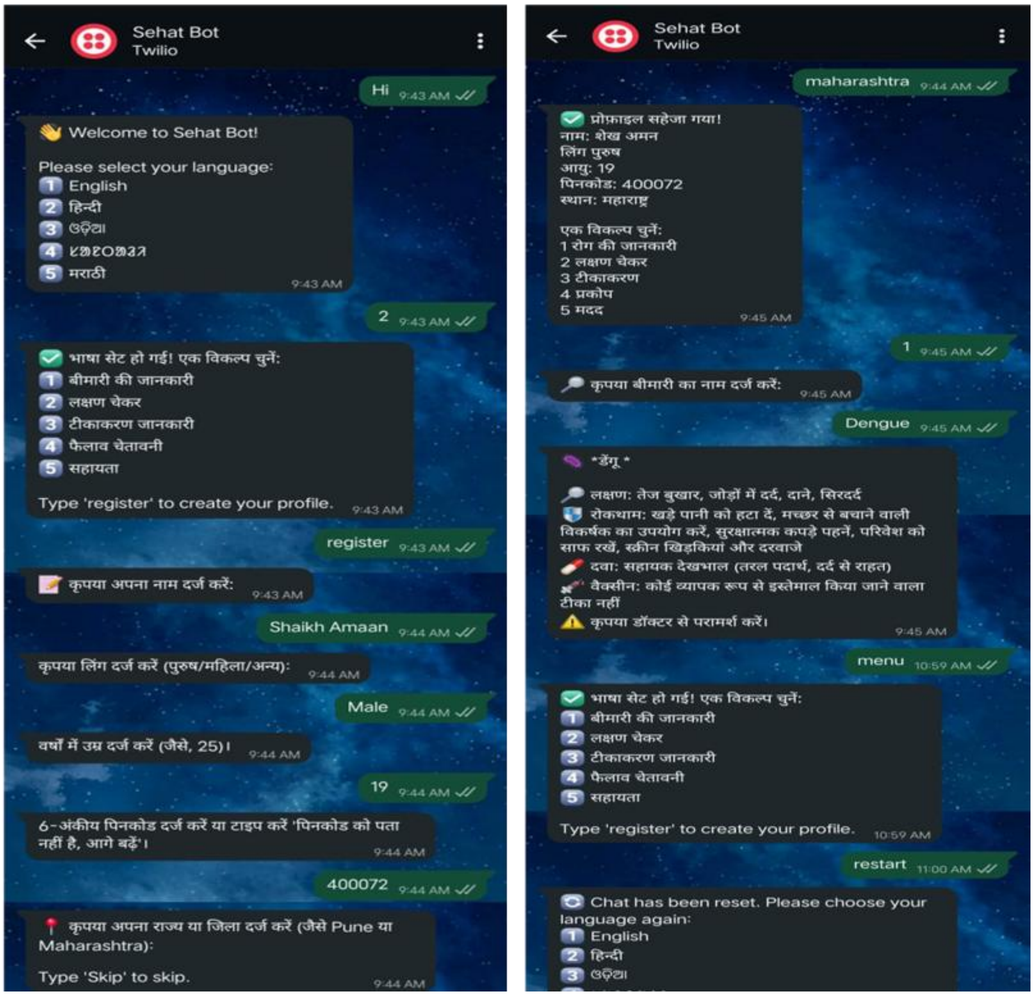

Fig.2: Working prototype of the system.

Table 1: Comprehensive feature comparison between SehatBot, MFine, DocsApp and Tata1mg

Parameter	SehatBot	MFine	DocsApp	Tata1mg
Platform Accessibility	WhatsApp-Based (no app installation)	Mobile application	Mobile application	Mobile application
Multilingual Support	Extensive (English, Hindi, Odia, Regional languages)	Partial	Limited	Partial
Support for low Digital Literacy	Menu-driven interaction	Limited	Limited	Limited
Disease Information and Awareness	Available	Available	Available	Available
Symptom based Disease Guidance	Available	Available	Available	Available
Vaccination Schedule Guidance	Available	Limited	Not Available	Partial
Government Health-Data Integration	Available (IDSP, MoHFW)	Not Available	Not Available	Not Available
Public Health Outbreak Alerts	Available	Not Available	Not Available	Not Available
Primary Healthcare Focus	Preventive healthcare and awareness	Online consultation	Online consultation	Pharmacy and consultation
Cost to End User	Free	Consultation charges apply	Consultation charges apply	Basic service free (Consultation charges apply)

consistent responses as well as allowed for limited Natural Language Processing (NLP) to allow for slight variations in inputs.

From an implementation standpoint the system is planned to be implemented upon lower cost infrastructure using well known communication systems like WhatsApp and SMS therefore decreasing the dependency upon custom applications or high-end user device requirements. A comprehensive feature comparison between SehatBot and existing digital health platforms is detailed in Table 1. The primary cost of operations will be related to message transmission services, back-end infrastructure, and maintenance. The system has been built around a sustainability model that does not charge the end-user; but rather generates income from relationships with health care organizations/providers, service integrations and partnerships with institutions. This strategy provides the necessary resources for the longevity of the system while at the same time allowing for accessibility. However, widespread use of SMS in low connectivity areas could increase operational costs and require optimization. While there have been many benefits, there are some significant limitations. Due to the fact that this study was based on a small number of participants who all worked within the company/organization, this study does not provide results on how the system would perform in a larger scale or in-field testing environment. In addition, since no formal quantitative measures (i.e., accuracy rate, completion times) were collected during testing, it is difficult to compare different types of input (e.g., keyboard vs. touch screen). Currently, the system is only able to send messages over the WhatsApp application and thus comparison data does not exist. In general terms, these results support the concept of utilizing a structurally organized multi-lingual chat bot deployed on a messaging platform to enhance access to quality healthcare information within resource-poor communities.[17] The system achieved its objectives of providing usable functionality, timely responses, and controlling what information is provided to the user while demonstrating potential for scalable and sustainable real-world implementations.

4. Conclusion

SehatBot was a messaging-based chat bot created as part of this research project to help provide access to structured/reliable health care information in low resource areas. To ensure both consistent and usable responses from the chat bot a hybrid conversation model was developed combining three different approaches including; Rule-Based Logic (RBL) Finite State Machine (FSM) Interaction Limited Natural Language Processing (LNLP). By using the WhatsApp platform to deploy the chat bot, the researchers were able to leverage an existing common mode of communication which would assist in reducing barriers to adoption. An internal pilot level assessment indicated that the system performed well under normal operating conditions provided effective multilingual functionality and produced near real time results. Additionally, because of the use of a structured, menu driven approach to designing the user interface, users are better able to navigate through the application and avoid confusion which can be a challenge for many who have limited digital experience. Although there has been some evidence of success identified as part of this research, there are limitations related to this study. Due to the nature of this being an internal pilot level assessment with a relatively small number of participants, the researchers did not include a larger scale or external quantitative analysis. Therefore, future research will need to investigate how to deploy the system at a wider scale, integrate the system into multiple platforms and create performance metrics that allow researchers to assess the effectiveness of the system. In general, this research provides evidence that using structured chat bots may be one way to increase access to healthcare information in communities with limited availability of traditional sources of information.

CRediT Author Contribution Statement

Mohammed Abbas Sayyed: Conceptualization, Methodology, Software, Validation, Investigation, Data curation, Visualization, Writing – Original draft. **Amaan Shaikh:** Conceptualization, Methodology, Software, Validation, Investigation, Data curation, Visualization, Writing – Original draft. **Shrushti Sapte:** Conceptualization, Methodology, Software, Validation, Investigation, Data curation, Visualization, Writing – Review & editing. **Zara Shaikh:** Conceptualization, Methodology, Software, Validation, Investigation, Data curation, Visualization, Writing – Review & editing. **Yumna Qureshi:** Conceptualization, Methodology, Software development, System implementation, Validation, Investigation, Data curation, Visualization, Writing – Original draft. **Awij Shaikh:** Conceptualization, Methodology, Software, Validation, Investigation, Data curation, Visualization, Writing – Review & editing. All authors participated in the design and development of the SehatBot prototype, reviewed the manuscript critically for important intellectual content, approved the final manuscript, and agree to be accountable for all aspects of the work. All authors have read and agreed to the published version of the manuscript.

Data Availability Statement

The datasets and implementation details supporting the findings of this study are available from the corresponding author upon reasonable request.

Conflict of Interest
There is no conflict of interest.

Artificial Intelligence (AI) Use Disclosure
The authors declare that artificial intelligence (AI)-assisted tools were used only for language refinement, grammar improvement, and manuscript structuring purposes during the preparation of this work. All technical content, experimental implementation, results, and interpretations were independently developed and verified by the authors.

Supporting Information
Not applicable.

References

[1] L. Laranjo, A. G. Dunn, H. L. Tong, A. B. Kocaballi, J. Chen, R. Bashir, D. Surian, B. Gallego, F. Magrabi, A. Y. Lau, E. Coiera, Conversational agents in healthcare: a systematic review, *Journal of the American Medical Informatics Association*, 2018, **25**, 1248-1258, doi: 10.1093/jamia/ocy072.

[2] B. A. Shawar, E. Atwell, Chatbots: are they really useful?, *Journal for Language Technology and Computational Linguistics*, 2007, **22**, 29-49.

[3] A. M. Turing, Computing Machinery and Intelligence (1950), *Mind*, 1987, **59**, 33-60, doi: 10.1093/oso/9780198250791.003.0017.

[4] J. Weizenbaum, ELIZA—A Computer Program for the Study of Natural Language Communication Between Man and Machine, *Communications of the ACM*, 1983, **26**, 23-28, doi: 10.1145/357980.357991.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems, 2017, **30**, doi: 10.48550/arXiv.1706.03762.

[6] T. Bickmore, D. Schulman, L. Yin, Maintaining engagement in long-term interventions with relational agents, *Applied Artificial Intelligence*, 2010, **24**, 648-666, doi: 10.1080/08839514.2010.492259.

[7] Pallets Projects, *Flask: A Lightweight WSGI Web Application Framework*, 2023, https://flask.palletsprojects.com/en/stable/, Accessed 15 April 2026.

[8] Meta Platforms, Inc., *WhatsApp Business Platform Documentation*, 2023, https://developers.facebook.com/documentation/business-messaging/whatsapp/overview, Accessed 15 April 2026.

[9] Twilio Inc., *Twilio API for WhatsApp*, 2023, https://www.twilio.com/en-us/messaging/channels/whatsapp, Accessed 15 April 2026.

[10] NGINX, Inc., *NGINX Documentation*, 2023, https://nginx.org/en/docs/, Accessed 17 April 2026.

[11] *Gunicorn: Python WSGI HTTP Server*, 2023, https://gunicorn.org/, Accessed 17 April 2026.

[12] Docker Inc., *Docker Documentation*, 2023, https://docs.docker.com/, Accessed 19 April 2026.

[13] Rasa Technologies GmbH, *Rasa: Open-Source Conversational AI Framework*, 2023, https://rasa.com/, Accessed 20 April 2026.

[14] SQLite Consortium, *SQLite Documentation*, 2023, https://sqlite.org/docs.html, Accessed 18 April 2026.

[15] PostgreSQL Global Development Group, *PostgreSQL Documentation*, 2023, https://www.postgresql.org/docs/, Accessed 20 April 2026.

[16] Google LLC, *Google Cloud Translation API Documentation*, 2023, https://docs.cloud.google.com/translate/docs/reference/rest, Accessed 21 April 2026.

[17] M. Honnibal, spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsin, 2017, https://sentometrics-research.com/publication/72/, Accessed 20 April 2026.