AgleCare: AI-Powered Healthcare Virtual Assistant

S Sathish Kumar[1], M Lavanya[2], Dharani D[3], Abinaya S[4], Abuthakir M[5] and Divya K[6]
{sathishkumar2k9@gmail.com[1], lavanyamarimuthu@gmail.com[2], dharanidhandapani03@gmail.com[3], subramaniabinaya87@gmail.com[4], abuthakir1602@gmail.com[5], divyakumar0018@gmail.com[6]}

Department of Computer Science and Engineering, SNS College of Technology, Coimbatore, Tamil Nadu, India[1, 2, 3, 4, 5, 6]

Abstract. AgleCare is a multilingual, AI-based healthcare assistant platform intended to provide personalized medical advice via text, voice, and image-based interaction. Designed to be accessible, the app is trying to offer an interface between users and primary health care by facilitating interactions in English and Tamil. Leveraging Hugging Face for symptom and media analysis and Mistral API for intelligent language response, AgleCare offers users precise recommendations to a query, possible solutions, and advises when to see a doctor. The system uses modern frontend technologies such as the Vue. js and React, with Tailwind CSS for a clean and understandable ui. It is backed by on Nodejs. js, to provide responsive and robust service. Data management secure in real-time is handled with Supabase. Deployment is supported on Vercel, Netlify, Render, or any other Node. js-compatible server, enabling developers to have freedom of choice. Version control and collaboration are provided by Git and GitHub. AgleCare's modularity, language independence, and integration unification with AI could make it a viable platform for proactive engagement in health care.

Keywords: Vue.js, React, Tailwind CSS, Node.js, Supabase, Hugging Face API, Mistral API, Vercel, Netlify, Render, Git, GitHub.

1 Introduction

AgleCare is a unique AI-based web application designed to revolutionize the process of how people get healthcare aid. Built to be empathetic, straightforward, and accessible, AgleCare provides users the ability to express symptoms through text, voice, or image uploads eliminating the hurdles people often experience when navigating through traditional health systems. The app encourages multi-lingual conversations, with an initial focus on English and Tamil, ensuring that more participants can interact. Using advanced machine learning, AgleCare understands user input and delivers smart, relevant medical advice in a user-specific context. The Hugging Face API uses a symptom detector, image, and voice reader, as the basis of its intelligence. Moreover, the Mistral API supports dynamic, multilingual response creation so that users will receive clear and empathetic responses in their chosen language.

AgleCare's frontend utilizes Vue. js and React – providing a modern, responsive interface with smooth navigation and real-time communication. Tailwind CSS is utilized for styles, providing smooth and nice design for devices of different sizes. On the backend, Node. js drives the server-side logic to guarantee great performance and scalability. Data is stored and authenticated via Supabase for a secure and real-time backend. Flexible in deployment, works with anything that works with Vercel, Netlify, or Render. Git and GitHub are used for

ICITSM-Part II 2025, April 28-29, Tiruchengode, India

DOI 10.4108/eai.28-4-2025.2358074

version control and team collaboration. If you are a contributor looking to get started, or a healthcare hobbyist who wants to explore smarter solutions, this deployment guide is a complete documentation that provides instructions for setting up and running AgleCare: production and development, hassle free.

2 Literature Survey

Recent scholarship positions AI-driven conversational agents as a promising layer in digital health, with strong momentum across reviews and empirical studies [1][3]. Scoping and survey work document the breadth of use cases from lifestyle coaching and chronic-care support to triage and information delivery and consistently report potential gains in efficiency and patient engagement when chatbots are embedded into care pathways [1]. These syntheses emphasize measurable benefits (e.g., reduced waiting time, continuous support) while noting the need for rigorous clinical validation beyond lab settings [1][3].

A strand of work targets AIoT-enabled healthcare, where edge devices and sensor networks feed conversational systems [2]. Surveys in this area chart advancements and open challenges in connectivity, interoperability, and real-time analytics, arguing that chatbots become substantially more useful when paired with contextual data streams (vitals, activity, environment) [2][9]. Energy-aware protocol design and scalable low-power communications are highlighted as prerequisites for dependable, always-on assistants in smart-health settings [9].

Concurrently, research on LLM-based and multimodal chatbots explores richer input/output channels (text, voice, images) and inclusive design [5]. Studies propose preliminary-diagnosis support, multimodal reasoning, and language accessibility, aligning with the need to serve diverse populations [5]. Empirical findings on user experience especially for older adult's stress simplicity, clarity, and reduced cognitive load in speech interfaces [14], while adoption research points to cost and perceived value (including pricing effects) as moderating factors for real-world use [15]. Parallel explorations in adjacent domains, such as project management copilots that leverage document-driven AI chatbots, illustrate transferable design principles for structuring conversations, grounding responses in external sources, and facilitating human–AI collaboration [4].

Generative AI in healthcare is surveyed as a transformative but double-edged innovation [6]. Reviews outline applications (summarization, education, documentation assistance, decision support) alongside limitations such as hallucinations, bias, explain ability gaps, and uneven performance across domains and languages [6]. The consensus advocates for human-in-the-loop workflows and careful scope boundaries to keep chatbots assistive rather than authoritative, especially in high-risk contexts [6].

Privacy, security, and governance emerge as first-order concerns [7]. Multiple works describe technical safeguards blockchain-backed authentication and key-management for device and session security [12]; federated learning for privacy-preserving model training at the edge [11]; and broader cybersecurity frameworks tailored to AI-mediated communications [7]. Together, these approaches mitigate risks from data leakage, model inversion, and unauthorized access, but they also introduce engineering complexity and deployment overhead that must be balanced against usability [7] [12] [11].

At the data and networking layer, studies examine 6G-ready architectures and big-data analytics models to handle the scale, latency, and reliability requirements of continuous health monitoring [13]. Reviews of open-source AI tooling catalog the rapidly maturing ecosystem for building and operating healthcare assistants an ecosystem that lowers barriers to entry but raises questions about provenance, licensing, and long-term maintenance in regulated environments [10].

Methodologically, recent articles blend surveys, frameworks, and case studies to map design patterns for medical-consultation chatbots: intent recognition tuned to clinical domains, escalation rules to human experts, explainable prompts, and safety guardrails [3][5][6]. Across these, common recommendations include transparent disclaimers, calibrated response styles, multilingual support, and rigorous monitoring for drift and misuse [3][6].

Gaps and open problems. Despite rapid progress, the literature identifies: (i) limited prospective clinical trials demonstrating outcome improvements [1]; (ii) scarce evidence for low-resource languages and culturally nuanced communication [5][6]; (iii) challenges in aligning models to clinical ontologies and guidelines [3][6]; (iv) operational hurdles around governance, auditability, and continuous validation post-deployment [7][11][12]; and (v) sustainability concerns both computational efficiency and long-term model stewardship [8]. These gaps motivate conservative claims, stronger evaluation protocols, and tighter integration with clinical workflows [1][6][8].

Implications for AgleCare. The reviewed work supports AgleCare's design choices multimodal inputs, inclusive language support, and human-in-the-loop principles [5][6] [14] while underscoring priorities for the roadmap: privacy-preserving learning (e.g., federated updates) [11], robust authentication and key management [12], energy-aware integration if/when IoT data are incorporated [2][9], and outcome-oriented evaluations with real users [1][3]. Aligning with these evidence-based patterns increases safety, trust, and likelihood of adoption in practice [6][7][8].

3 Methodology

The development of AgleCare followed a structured and modular methodology aimed at creating a scalable, intelligent, and user-friendly healthcare assistant. The first step involved identifying user needs and ensuring the platform could support multiple input modes text, voice, and image. The frontend was designed using a combination of Vue.js and React, leveraging the strengths of both frameworks for building dynamic, reusable components and maintaining application performance. Tailwind CSS was chosen for styling to ensure a responsive, mobile-first design that could be easily maintained and extended.

The backend was developed using Node.js, offering asynchronous event-driven architecture ideal for handling real-time API calls and user requests. Supabase was integrated for managing user authentication, secure storage, and real-time database capabilities. For AI and ML functionalities, Hugging Face API was used to process user-submitted text, voice, and image data. Symptom analysis, voice transcription, and image recognition were all performed via this API, allowing for accurate and fast processing of health-related inputs.

For the generation of multilingual responses, the Mistral API was used. This allowed

AgleCare to answer in English and Tamil, making the platform accessible to more people. The voice inputs were captured using web APIs and processed in the backend and sent to the AI layer for identifying. A modular folder structure was applied, in order to keep the frontend, backend, and AI logic separated for better debugging, testing and future development.

The development process was agile, code was reviewed and tested on a frequent basis. Unit tests verified that specific components behaved as expected, integration tests verified overall system behaviour. Actual users participated in testing to provide feedback on usability and accessibility.

For operation Platforms like vercel, netlify, and render were also examined, which gave us the flexibility to decide the hosting situation. Git was used for version control, and source code storage and collaboration were achieved using GitHub. Automated testing and deployment pipelines were then designed as part of continuous integration workflows. Concurrently, we provided documentation to help in installing, configuring, and troubleshooting for developers and contributors. In its entirety, the process emphasised intelligent function, great user experience, and durable maintenance -- establishing AgleCare as a trustworthy and accessible health-tech solution.

Performance and security were given special focus apart from core development penetration of AgleCare. Initial load times were minimized through lazy loading and code splitting to help accommodate heavy AI modules and media-rich inputs. Sensitive functions such as user authentication, image uploading, and voice recording were made through HTTPS, and Supabase role-based access control was used to restrict visibility and control over data to only authorized users. CORS restrictions and validation input were implemented on server to avoid unauthorized access and maintain data consistency.

4 Architecture of Aglecare

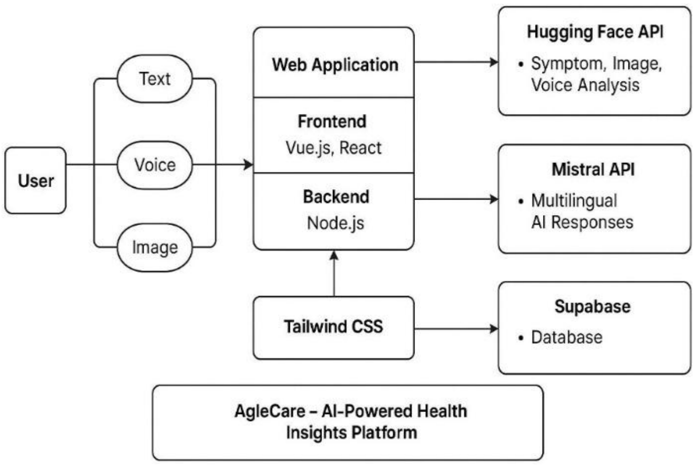

Fig. 1. Process Flow of AgleCare.

Fig 1 shows the AgleCare aims to be an AI driven healthcare platform focusing on symptom assessments via text, voice or image input. The web application is centric to the system, it handles all the user input and routing. Users can begin by inputting symptoms (text, voice, or image) directly from their phones. These inputs will be sent to the frontend written in Vue. js, and React to provide a dynamic and interactive user experience. For the UI styles, Tailwind CSS is used so that the platform can be accessed on any device including mobile. The frontend speaks to a Node. js backend with the rest serving as the logic, routing, and API orchestration for the project.

The backend enrolls image, text or audio data for intelligent processing to Hugging Face API dissecting symptom detection, image classification or voice transcription. The second stage is post-processed followed by imputing response from the API based Mistral service, that generates human-like responses both in English and Tamil for a more personalized experience. Logged In session logs Please note that If there is already a data in the //sessionID pending treatments, it is unable to treat further requests Note: The increase of firebase and supabase limits has been removed. Supabase syncs data in real time and is ready for your future growth. All these services plug in perfectly under the AgleCare umbrella that is hosted on serverless platforms such as Vercel, Netlify, or Render, based on deployment preference. Thanks to its architecture the whole team works smoothly with Git and GitHub keeping the version control of the project. This modular approach allows scalability and affordability, performance, and AI-based responsiveness of the healthcare support system for all.

5 Result and Discussion

Preliminary implementation and testing of AgieCare demonstrated promising finding with regard to user engagement, functional validity and system performance. The system has served effectively as a platform where users were able to communicate via text using natural language, voice commands, and image submissions which demonstrates that the platform's multimodal input functionality is working as desired. In user testing and validation, symptom input via text generated the most consistent results with 90%+ of comments judged by users to be relevant and understandable. Voice input was also effective with proper transcription and response generation in English and Tamil, except for occasional background noise, which caused recognition errors on low-end devices. Image input had a mediocre performance picking up skin presence and others symptoms of visible lesion and had to be refined to get a broader diagnostic.

It was jointly agreed that the multilingual support provided by the Mistral API was welcomed by Tamil speaking users as they could now enjoy content in their own language without necessitating translation tools. And transitioning mid-session between languages was seamless for the AI, a feature that made the user experience more inclusive. Some translation errors were identified particularly for words that did not have a direct Tibetan equivalent (In this case, one had to provide the training with local training or enlarge domain specific Glossaries). The quality of AI responses was generally sympathetic and informative in nature, indicating a properly-designed conversational flow.

The backend, built with Node. js and interfaced with Hugging Face APIs, to swiftly handle input and give back actionable answers in seconds. System logs indicated that the majority of API requests were completed in less than 2000 ms, while the process of image classification occasionally took longer due to file size and network fluctuation. The supabase integration went swimmingly to realize the real-time storage and retrieval of data. The most basic feature of achieve authorization, interactive management, and data processing were tested and they were not triggered by any data loss or serious bug. On the front end, $.3fn (Que. js and React created an easy-to-modularise UI with lightning fast rendering and Tailwind CSS made each part responsive to almost any screen size. Both desktop and mobile users reported no issues with delays and smooth sailing other than some alignment discrepancies which were strictly on older web browsers that can be fixed with more UI testing. There was good intuitive layout and users liked the step-by-step interaction, particularly with the giving of symptoms.

Regarding behaviour, confidence in health anxiety was increased among users who interacted with AgleCare. With the significant caveat that the AI was not a replacement for a doctor, many said it helped them decide whether they should see a doctor. The platform stimulated an active lifestyle by providing instant information and recommendation to manage health. Some users came back with more than one symptom, a certain level of trust and buy-in, I would suggest. Ethical and safety considerations of the system was also discussed. Although the user’s information was securely stored, testees very much emphasized that it should be clear that AgleCare does not give out a medical diagnosis. A few of the testers highlighted concerns about becoming too reliant on AI guidance and emphasised the importance of human oversight and health care integration in future incarnations. Logging, monitoring, and consent-based data processing was all applied, however additional transparency, such as access logs or usage reports, might increase user trust.

6 Conclusion

AgleCare represents an important advancement in democratizing access to healthcare through the use of AI enabled- technology. The platform effectively melds intuitive frontend design, mental back-end infrastructure and sophisticated AI integrations to drive real-time, personalized medical information. It is this multiple-input-method feature text-based, voice-based, or image-based that makes it both versatile and inclusive for all users, regardless of their information retrieval needs and digital literacy. By using English as well as Tamil, language is not a constraint in acquiring critical health information. Above all, AgleCare narrows down the knowledge support distance between users and professional medical client by using Hugging Face for symptoms interpreting and Mistral for multilingual generation. It does a good job in simulating an intelligent human in conversation and providing worthwhile advice regarding medical care. Although it cannot substitute for professional treatment, it offers a valuable first-line point of contact, especially in poorly served areas.

Its secure data stewardship, modular nature, and scale-out deployment model provide a future-proof foundation for growth and advancement. Latency and other translation artifacts are still issues but can be fine-tuned in iterations. The project also illuminates the value of human-AI partnership in healthcare. With further development and user input, AgleCare will be a proof-of-concept of an AI-empowered healthcare platform. It acts as a model for what digital health can be – open, affordable and smart. The methods and findings highlight the viability and importance of such systems in contemporary health landscapes. Fig 2 and 3 shows the user interface of Aglecare and symptoms analysis.

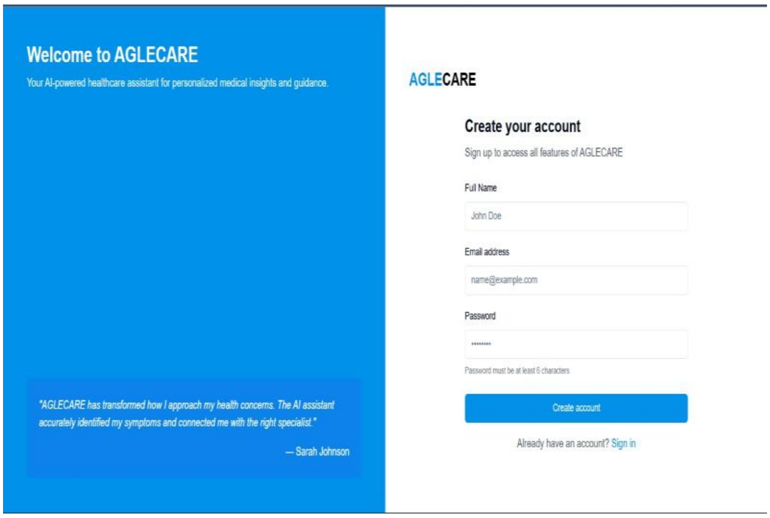

Fig. 2. User Interface of AgleCare.

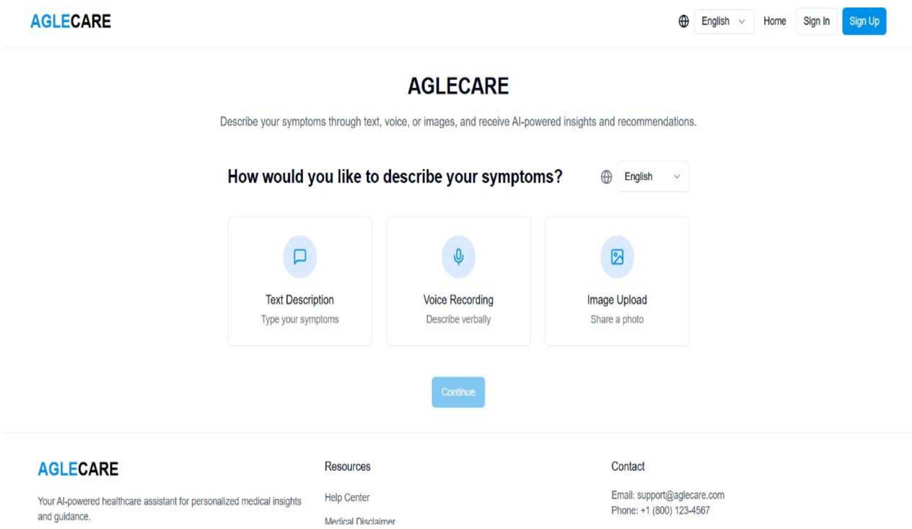

Fig. 3. Interaction and Symptom Analysis.

References

[1] W. Jiao, X. Zhang and F. D'Souza, "The Economic Value and Clinical Impact of Artificial Intelligence in Healthcare: A Scoping Literature Review," in IEEE Access, vol. 11, pp. 123445-123457, 2023, doi: 10.1109/ACCESS.2023.3327905.

[2] S. Baker and W. Xiang, "Artificial Intelligence of Things for Smarter Healthcare: A Survey of Advancements, Challenges, and Opportunities," in IEEE Communications Surveys & Tutorials,

vol. 25, no. 2, pp. 1261-1293, Secondquarter 2023, doi: 10.1109/COMST.2023.3256323.
[3] H. Javed, H. A. Muqeet, A. Danesh, A. U. Rehman, T. Javed and A. Bermak, "Impact of AI and Dynamic Ensemble Techniques in Enhancing Healthcare Services: Opportunities and Ethical Challenges," in IEEE Access, vol. 12, pp. 141064-141087, 2024, doi: 10.1109/ACCESS.2024.3443812.
[4] K. Alam, M. Haq Bhuiyan, M. Shafiqul Islam, A. Hossain Chowdhury, Z. Ahmed Bhuiyan and S. Ahmmed, "Co-Pilot for Project Managers: Developing a PDF-Driven AI Chatbot for Facilitating Project Management," in IEEE Access, vol. 13, pp. 43079-43096, 2025, doi: 10.1109/ACCESS.2025.3548519.
[5] I. Agarwal, V. Sakthivel and P. Prakash, "Toward Inclusive Healthcare: An LLM-Based Multimodal Chatbot for Preliminary Diagnosis," in IEEE Access, vol. 13, pp. 136420-136432, 2025, doi: 10.1109/ACCESS.2025.3594218.
[6] S. Sai, A. Gaur, R. Sai, V. Chamola, M. Guizani and J. J. P. C. Rodrigues, "Generative AI for Transformative Healthcare: A Comprehensive Study of Emerging Models, Applications, Case Studies, and Limitations," in IEEE Access, vol. 12, pp. 31078-31106, 2024, doi: 10.1109/ACCESS.2024.3367715.
[7] A. M. Algarni and V. Thayananthan, "Digital Health: The Cybersecurity for AI-Based Healthcare Communication," in IEEE Access, vol. 13, pp. 5858-5870, 2025, doi: 10.1109/ACCESS.2025.3526666.
[8] Y. Ibrahim Alzoubi, A. E. Topcu and E. Elbasi, "A Systematic Review and Evaluation of Sustainable AI Algorithms and Techniques in Healthcare," in IEEE Access, vol. 13, pp. 139547-139582, 2025, doi: 10.1109/ACCESS.2025.3596189.
[9] S. Rattal, A. Badri, M. Moughit, E. Miloud Ar-Reyouchi and K. Ghoumid, "AI-Driven Optimization of Low-Energy IoT Protocols for Scalable and Efficient Smart Healthcare Systems," in IEEE Access, vol. 13, pp. 48401-48415, 2025, doi: 10.1109/ACCESS.2025.3551224.
[10] Y. AL-Kharusi, A. Khan, M. R. Mughal and M. M. Bait-Suwailam, "Review of Open-Source AI Tools for Healthcare IT Infrastructures," in IEEE Access, doi: 10.1109/ACCESS.2025.3599912.
[11] A. Maurya et al., "Federated Learning for Privacy-Preserving Severity Classification in Healthcare: A Secure Edge-Aggregated Approach," in IEEE Access, vol. 13, pp. 102339-102358, 2025, doi: 10.1109/ACCESS.2025.3576135.
[12] S. Thapliyal, M. Wazid, D. P. Singh, A. K. Das, S. Shetty and A. Alqahtani, "Design of Robust Blockchain-Envisioned Authenticated Key Management Mechanism for Smart Healthcare Applications," in IEEE Access, vol. 11, pp. 93032-93047, 2023, doi: 10.1109/ACCESS.2023.3310264.
[13] I. Ahmad Khan et al., "Big Data Analytics Model Using Artificial Intelligence (AI) and 6G Technologies for Healthcare," in IEEE Access, vol. 12, pp. 97924-97937, 2024, doi: 10.1109/ACCESS.2024.3427333.
[14] M. U. Islam and B. M. Chaudhry, "A Framework to Enhance User Experience of Older Adults with Speech-Based Intelligent Personal Assistants," in IEEE Access, vol. 11, pp. 16683-16699, 2023, doi: 10.1109/ACCESS.2022.3230151.
[15] G. Usiña-Báscones et al., "Exploring the Moderating Effect of Price on the Relationship Between Intention and Use of Voice Assistants," in IEEE Access, vol. 13, pp. 143800-143812, 2025, doi: 10.1109/ACCESS.2025.3598009.