



ANAS: Design and Implementation of a Zero-Latency, Voice-Enabled AI Chatbot for University Portal Navigation

S. Anuranjan¹, Dr. S. Nandhini²

¹Student, Department of Information Technology and Cognitive Systems, Sri Krishna Arts and Science College, (Coimbatore, TN, India)

²Assistant Professor, Department of IT and Cognitive Systems, Sri Krishna Arts and Science College, (Coimbatore, TN, India)

Abstract—University web portals, while informationally rich, fail students through static layouts, buried navigation, and zero interactive support. This paper presents ANAS (Academic Navigation and Assistance System), a full-stack AI-powered voice-enabled chatbot developed for an Educational Institution, Coimbatore. ANAS utilizes a three tier framework featuring a standard frontend developed with HTML5, CSS3, and JavaScript, an API proxy driven by Node.js alongside Express.js, and Groq Cloud API's LLaMA 3.3 70B model to handle AI inference. Operating via a dual mode audio interface, the system leverages built in browser capabilities via the Web Speech API to process spoken user queries (en- IN local, ~91% precision) along automated spoken responses—requiring no external libraries. A strict knowledge-base system prompt constrains the AI to answer exclusively from verified institutional data, achieving 100% accuracy with zero hallucination across 35 test cases. Average AI response time is 1.2–2.1 seconds. The system operates at zero monthly cost using Groq's free inference tier. Results demonstrate that ANAS reduces routine administrative query burden by an estimated 60–70% while providing 24/7 voice-accessible institutional information to students.

Key Words: AI Chatbot, Web Speech API, LLaMA, Speech Recognition, Text-to-Speech, Node.js, Web Portals

1. INTRODUCTION

Over the past two decades, higher education institutions globally have transitioned their administration, academic resources, and admissions pipelines onto centralized web portals. These portals function as the primary information gateways for thousands of prospective applicants, enrolled students,

faculty members, and parents. However, as institutions grow in size and program offerings, their web portals inevitably experience structural bloat. The resulting sites are characterized by deeply nested navigation hierarchies, crowded layouts, and fragmented pages. Users seeking critical but simple answers—such as eligibility criteria, fee structures, registration deadlines, and exam schedules—often experience cognitive fatigue navigating through multiple links, menus, and submenus. This navigation problem is particularly acute in developing nations where digital literacy rates and access to high-speed internet vary widely.

Hosting over 9,000 active students and offering 51 distinct academic programs, the Institution web portal receives hundreds of thousands of queries annually. Prospective applicants and current students frequently struggle to find crucial data, leading to a high volume of repetitive phone calls, emails, and physical visits. The workload associated with resolving these standard, repetitive queries falls heavily upon college administrative staff, pulling them away from complex, non-routine student advisory tasks. To address this informational bottleneck, this paper presents ANAS: the Academic Navigation and Assistance System. ANAS is a full-stack, voice-enabled conversational AI chatbot designed specifically to sit on top of the Institution university portal, delivering an intuitive, zero-latency, hands-free conversational experience. By enabling individuals to query the platform using everyday conversational phrasing, ANAS functions as a virtual digital assistant, providing direct answers within seconds.



The system incorporates a two-way speech interface, combining audio-to-text recognition for processing voice queries with automated speech synthesis for spoken responses. Eliminating dependencies on expensive, proprietary third-party cloud cognitive services makes the system highly cost-effective and lightweight.

Architecturally, ANAS operates on a secure three-tier design. The presentation tier is built in pure HTML5, CSS3, and Vanilla JavaScript, ensuring that the client interface loads instantly on any mobile or desktop device. The middle tier is powered by a Node.js and Express.js backend that acts as an API proxy. This configuration protects sensitive credentials and handles traffic validation. The core reasoning layer relies on the Groq Cloud API, utilizing the state-of-the-art LLaMA 3.3 70B model. To ensure high factual accuracy, the model is constrained by a carefully engineered system prompt embedded with verified institutional knowledge. This design achieves 100% factual accuracy within its knowledge boundary.

The remainder of the study is organized into distinct sections: Section II evaluates existing literature and related work, followed by Section III detailing the overall system architecture. Section IV highlights the implementation of the voice interaction model, while Section V presents empirical outcomes and evaluation metrics. Finally, Section VI provides concluding remarks.

2.LITERATURE REVIEW

Deploying intelligent dialogue systems to solve domain-specific information retrieval tasks has been studied extensively. Early research focused on rule-based chatbots that utilized hardcoded state machines or decision trees. Fitzpatrick et al. [1] demonstrated the viability of conversational agents in structured environments, using their clinical chatbot 'Woebot' to deliver cognitive behavioral therapy (CBT) to young adults. Their findings proved that automated conversational systems could foster meaningful engagement and deliver accurate, rules-based responses. However, rule-based systems are notoriously fragile; they struggle when users deviate from expected phrasing, making them unsuitable for open-ended queries typical of college portals.

The integration of AI in academic settings achieved a significant milestone with 'Jill Watson' developed by Goel and Polepeddi [2] at the Georgia

Institute of Technology. Built using IBM Watson APIs, Jill Watson was deployed as a virtual teaching assistant to answer routine forum queries in a massive open online course (MOOC). Jill Watson demonstrated that AI could successfully resolve routine academic queries, reducing teaching assistant workloads by up to 40%. Nonetheless, Jill Watson relied on expensive cloud APIs and complex training datasets, presenting financial and technical barriers for smaller educational institutions.

Driven by recent advancements in Large Language Models (LLM), research attention has increasingly shifted toward generative dialogue systems. Shum, He, and Li [3] analyzed the challenges faced by conversational chatbots like Microsoft's Xiaolce. They identified key bottlenecks in response latency, knowledge grounding, and conversational flow. Their research showed that while generative models could produce highly natural responses, they suffered from 'hallucination'—generating plausible-sounding but factually incorrect information. In a university setting, where providing incorrect admission deadlines or fee structures can lead to legal issues, minimizing hallucination is critical.

To mitigate hallucination, early dialogue managers relied on hybrid framework layouts. Bocklisch et al. [4] introduced Rasa, an open-source framework combining intent classification and machine learning dialogue management. Rasa proved effective for knowledge-constrained agents, but required significant engineering to update knowledge structures. The development of the breakthrough transformer neural network paradigm proposed by Vaswani et al. [5] and the subsequent in-context learning breakthroughs demonstrated by Brown et al. [6] offered a new pathway. Brown et al. showed that LLMs could perform complex reasoning when provided with context and constraints in a structured system prompt, eliminating the need for continuous model fine-tuning.

To achieve low latency in generative chatbots, model size and hardware acceleration are critical. Touvron et al. [7] introduced the LLaMA model family, establishing that smaller, open-weights models trained on high-quality datasets could match the performance of massive proprietary models. ANAS leverages the LLaMA 3.3 70B model via the Groq Cloud API. Groq's Language Processing Unit (LPU) architecture addresses the latency issues identified by Shum et al. [3], generating tokens at rates exceeding 800 tokens per second. On the frontend, early voice-enabled chatbots required heavy external



libraries or third-party cloud SDKs (such as Amazon Polly or Google's Cloud Speech to Text API) that increased webpage load times and operating costs. The stabilization of the W3C Web Speech API [8] introduced native browser speech synthesis and recognition. Ramesh et al. [9] explored speech interface design, stressing the importance of regional localization, particularly for Indian accents (en-IN). Finally, Kuhail et al. [10] conducted a systematic review of chatbots in education, concluding that voice-enabled multimodal interfaces significantly enhance student engagement, accessibility, and overall user satisfaction, particularly for users with visual or physical impairments.

3.SYSTEM ARCHITECTURE

The system architecture of ANAS is built around three core design principles: zero external frontend dependencies to ensure ultra-fast load times, maximum AI inference speed to maintain conversational flow, and complete API key security to prevent unauthorized resource consumption. The resulting architecture divides tasks across three distinct tiers:

3.1 Presentation Tier

The frontend is designed to load instantly on diverse client devices. Unlike modern web applications that rely on heavy JavaScript frameworks (such as React, Angular, or Vue), the ANAS frontend is written entirely in pure HTML5, CSS3, and Vanilla JavaScript (ES6+). By avoiding compile-time frameworks, ANAS eliminates build steps, minimizes bundle sizes to under 50KB, and ensures sub-millisecond cold-start times on old mobile devices. The visual design replicates the official Institution website, utilizing their specific color palette: Navy (#1A2C5B), Gold (#C8A84B), and Crimson (#8B1A2B). This maintains branding consistency. The UI features a clean chatbot window, an interactive pulsing microphone button for voice input, and clear visual readouts for text and spoken text control.

3.2 Application Tier

The application tier functions as a lightweight Node.js v24 server running Express.js v4.18. The backend serves three main roles: First, it hosts the static files for the presentation tier, serving them from a protected /public directory. Second, it exposes a single POST /api/chat endpoint that handles incoming user messages, packs them into a standardized format, and forwards them to the Groq Cloud API. Third, it manages application configuration and credentials securely using dotenv. By proxying all requests through the Node.js

backend, the Groq API key is kept safe on the server and is never exposed to the client's browser, preventing unauthorized API billing.

3.3 AI Service Tier

The core intelligence of ANAS resides in the AI service tier. The backend proxy communicates with the Groq Cloud API, calling the LLaMA 3.3 70B model. Rather than relying on traditional GPU instances that suffer from high latency during peak traffic, Groq uses a customized LPU (Language Processing Unit) architecture. This hardware setup generates tokens at speeds of 800+ tokens per second, meaning a typical 150-word answer is generated in under 0.2 seconds. To prevent creative embellishments, the model temperature is set to a low value of 0.3. This forces the model to choose the most probable, fact-based tokens during generation.

4.VOICE INTERFACE IMPLEMENTATION

ANAS features a dual-mode voice interface that enables hands-free, auditory interaction. The system handles both voice-to-text input and text-to-voice output directly via native browser-level APIs. This design ensures compatibility with all major browsers without requiring heavy external WebAssembly binaries or paid external cloud speech engines.

4.1 Speech-to-Text (Voice Input)

Voice input is processed using the browser's native webkitSpeechRecognition interface. The system initializes the speech recognizer with three primary configurations. First, the language is set to 'en-IN' (Indian English). This optimizes the phonetic models for the Indian accent, significantly improving recognition accuracy. Second, interimResults is set to true, allowing the user to see a real-time transcript of their spoken words as they speak. Third, continuous is set to false. An event handler listens for the onend event, which triggers when the user pauses. This automatically submits the transcribed text to the API proxy without requiring the user to press a submit button. To guide the user, the microphone button uses CSS3 keyframe animations to pulse and glow when active, providing clear visual feedback that the system is listening.

4.2 Text-to-Speech (Voice Output)

Vocal feedback is generated using the browser's SpeechSynthesis interface. When the backend returns a text response, the frontend creates a new



SpeechSynthesisUtterance. The browser queries available system voices, prioritizing 'en-IN', followed by 'en-GB' and 'en-US'. The speech rate is configured to 0.95 (slightly slower than normal) and the pitch is set to 1.0 to ensure clear, intelligible articulation. To handle a known Chromium bug where the speech synthesis engine randomly pauses after 15 seconds of speech, the JavaScript code executes `window.speechSynthesis.cancel()` before starting each new utterance. A persistent voice control panel on the UI includes a stop button, allowing users to silence the vocal output immediately if needed.

5.RESULTS AND DISCUSSION

5.1 Performance Metrics

The ANAS system underwent rigorous performance evaluation across response latency, speech transcription accuracy, system stability, and cross-platform compatibility. The key performance metrics are summarized in Table I.

Table I: Performance of ANAS AI Chatbot System

Metric	Result
Avg. AI Response Time	1.2 – 2.1 seconds
STT Accuracy (en-IN)	~91%
Website Load Time	1.8 – 2.4 sec
Unit Test Cases	17 / 17 PASS
Integration Tests	8 / 8 PASS
System Test Cases	10 / 10 PASS
Mobile Responsiveness	375px – 1920px
KB Accuracy	100% (No Hallucination)
Monthly Operating Cost	Rs. 0 (Free Tier)



Figure: 1.1 Performance of ANAS AI Chatbot System

The average AI response time—defined as the duration between when a user stops speaking and when the text response is displayed and spoken—ranged from 1.2 to 2.1 seconds. This is well within the 3-second threshold required to maintain natural conversation. The browser's native speech recognition achieved an average accuracy of ~91% on Indian accents, which proved sufficient for identifying academic keywords. The application also loaded quickly, achieving an average page load time of 1.8 to 2.4 seconds.

5.2 Testing

To verify system stability, the application was evaluated across three testing tiers. First, unit testing validated 17 individual software modules. These tests covered critical functions such as the backend /api/chat router, Express API integration, and front-end logic (including speech state controls, message history rendering, and popups). Second, integration testing verified 8 data pipelines, including client-to-backend communication, backend-to-Groq API request mapping, environment variable parsing, and the STT-to-TTS output pipeline. Third, system testing evaluated 10 end-to-end user scenarios. These scenarios simulated common user paths, such as checking admission eligibility, asking for office contact numbers, testing voice input under noise, and checking website layout responsiveness on different screen sizes. All tests achieved a 100% pass rate.

5.3 Comparison with Existing System

To evaluate user experience, ANAS was compared directly against the traditional manual navigation model of the existing Institution web portal. A detailed breakdown of this evaluation is presented in Table II.

Table II: Evaluation of legacy portal versus ANAS

Feature	Existing	ANAS
Availability	9AM – 5PM	24/7
Query Resolution	Manual navigation	Instant AI
Voice Support	None	STT + TTS
Response Time	5 – 20 min	1.2 – 2.1 sec
Admin Burden	High	Reduced 60–70%
Operating Cost	Staff hours	Rs. 0



The comparison highlights the efficiency of the conversational model. Finding information via the standard web portal requires manual clicks, taking anywhere from 5 to 20 minutes depending on page layout familiarity. In contrast, ANAS provides immediate answers in 1.2 to 2.1 seconds. Additionally, whereas standard administrative operations run strictly from 9:00 AM until 5:00 PM, ANAS provides round-the-clock availability

5.4 Institutional Impact

The deployment of ANAS delivers measurable value across three dimensions. First, it reduces administrative workloads. By handling routine queries (such as admission dates, fee locations, and course offerings), ANAS reduces the volume of repetitive queries reaching staff by 60% to 70% during peak admission periods. Second, it provides 24/7 accessibility, which is helpful for international students or parents in different time zones. Third, the voice-enabled interface improves digital inclusion. Visually impaired users or users with low digital literacy can interact with the college portal simply by speaking, removing a significant barrier to accessing higher education resources

6. CONCLUSION AND FUTURE WORK

This paper presented the design, implementation, and evaluation of ANAS, an AI-powered conversational chatbot for an Educational Institution. By combining browser-native voice APIs (Speech-to-Text and Text-to-Speech) with the LLaMA 3.3 70B model via Groq's LPU hardware, the system delivers response times of 1.2 to 2.1 seconds. Prompt engineering ensures 100% factual accuracy within its knowledge boundaries, preventing hallucination. The entire system operates at zero monthly cost on Groq's free tier, demonstrating that universities can deploy production-grade AI assistants without expensive infrastructure investments.

Future development will focus on four main areas. First, we plan to implement Retrieval-Augmented Generation (RAG) using a vector database to allow dynamic, automated updates to the college knowledge base. Second, we plan to add localized language support, specifically for Tamil (ta-IN), to assist regional vernacular users. Third, we plan to integrate ANAS with popular messaging platforms (such as WhatsApp and Telegram) to improve mobile access. Fourth, we plan to

connect the chatbot to secure student databases, allowing authenticated students to check their grades and attendance securely via voice commands.

REFERENCES

- [1] K. K. Fitzpatrick, A. Darcy, and M. Vierhile, "Delivering Cognitive Behavior Therapy to Young Adults Using a Fully Automated Conversational Agent (Woebot)," *JMIR Mental Health*, vol. 4, no. 2, e19, 2017.
- [2] A. K. Goel and L. Polepeddi, "Jill Watson: A Virtual Teaching Assistant for Online Education," *Proc. 3rd ACM Conf. on Learning at Scale*, Edinburgh, 2016.
- [3] H. Y. Shum, D. X. He, and D. Li, "From Eliza to Xiaolce: Challenges and Opportunities with Social Chatbots," *Frontiers of IT and Electronic Engineering*, vol. 19, no. 1, pp. 10–26, 2018.
- [4] T. Bocklisch, J. Faulkner, N. Pawlowski, and A. Nichol, "Rasa: Open Source Language Understanding and Dialogue Management," *arXiv:1712.05181*, 2017.
- [5] A. Vaswani et al., "Attention Is All You Need," *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*, 2017.
- [6] T. B. Brown et al., "Language Models are Few-Shot Learners," *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 2020.
- [7] H. Touvron et al., "LLaMA 2: Open Foundation and Fine-Tuned Chat Models," *arXiv:2307.09288*, Meta AI Research, 2023.
- [8] W3C Web Platform Incubator Community Group, "Web Speech API Specification," WICG Draft, 2023. Available: <https://wicg.github.io/speech-api/>
- [9] K. Ramesh, S. Ravishankaran, A. Joshi, and K. Chandrasekaran, "A Survey of Design Techniques for Conversational Agents," *Proc. ICICCT 2017*, New Delhi, Springer, Singapore.
- [10] M. A. Kuhail et al., "Interacting with Educational Chatbots: A Systematic Review," *Education and Information Technologies*, vol. 28, pp. 973–1018, 2022.