International Journal of
Computational Learning & Intelligence
Vol. 04 | Issue 04 | 2025

RESEARCH ARTICLE OPEN ACCESS

AI- Powered Student Assistance ChatBot

Shahrukh Saifi . Adeeb Pasha K A . Shobha J . Sneha S . Jyoti Kiran M

School of Computer Science and Engineering,
REVA University, Bengaluru, India

DOI: **10.5281/zenodo.15483879**
Received: 27 April 2025 / Revised: 13 May 2025 / Accepted: 21 May 2025

Abstract - In the rapidly evolving digital education landscape, students require instant, accurate, and accessible academic and administrative support. This paper presents the design and implementation of an AI-powered Student Assistance Chatbot, tailored to serve institutions like REVA University and the Department of Technical Education, Government of Rajasthan. The chatbot leverages Natural Language Processing (NLP), a custom-trained dataset, and live data scraping to address queries related to admissions, fees, scholarships, placements, and more, across English and regional languages. The system aims to reduce dependency on human staff, ensure 24/7 support, and provide scalable automation. The implementation results indicate a high accuracy in intent recognition, quick response generation, and improved student satisfaction.

Index Terms —AI, Natural Language Processing, Chatbot, Student Services, Voice Interaction, Multilingual Support.

I. INTRODUCTION

The complexity and scale of modern educational institutions necessitate streamlined and effective communication between students and administrative departments. Traditional helpdesk systems often struggle with issues of scalability, response consistency, and support for multiple languages. With the advent and adoption of Artificial Intelligence (AI) in the education sector, intelligent virtual assistants have emerged as a promising solution to these challenges by offering contextual, automated, and real-time support. This research proposes the development of an AI-powered chatbot system tailored for higher education institutions. The system is designed to automate responses to frequently asked questions (FAQs), facilitate multilingual interaction, and integrate real-time data fetching capabilities to provide up-to-date institutional information. By addressing key communication barriers and reducing the load on administrative staff, the proposed chatbot aims to enhance the overall student experience while improving operational efficiency within academic environments.

Incorporating tools such as Natural Language Processing (NLP), Text-to-Speech (TTS), and real-time web scraping, the system ensures responsive and inclusive interactions. Support for regional languages allows the chatbot to cater to a diverse student demographic, especially in multilingual countries like India. The architecture is modular, allowing for easy integration with existing university portals and APIs. Additionally, voice-first interaction features make the system accessible to visually impaired users or those with limited digital literacy. Ultimately, this chatbot serves as a scalable, cost-effective solution to modernize institutional communication in academia. Traditional university information systems often lack real-time, interactive, and multilingual support, leading to communication barriers and delayed responses. There is a need for an intelligent system that can: - Understand and respond to queries in multiple languages. - Support both text and voice interactions. - Fetch and present up-to-date information from the university website. - Integrate seamlessly with modern web platforms.

II. LITERATURE REVIEW

Text-to-speech (TTS) and speech-to-text (STT) technologies form the backbone of voice-driven educational chatbots. The **Google Cloud Translation API** [1] enables real-time multilingual support, while the **gTTS library** [2] simplifies speech synthesis in Python-based applications. Similarly, **SpeechRecognition** [3] provides robust APIs to convert spoken input into textual queries, facilitating seamless interaction in voice-first chatbot systems. Educational chatbots often require real-time data retrieval to respond to student queries accurately. **Beautiful Soup** [4] plays a crucial role in scraping dynamic web content, which is essential for real-time response generation as explored by Patel et al. [7]. Such integrations improve the relevance and timeliness of chatbot responses. Several works have emphasized rule-based architectures for simplicity and reliability. Verma and Taneja [5] proposed a basic rule-based system to handle student queries effectively. Similarly, Batra and Jindal [9] demonstrated that rule-based multilingual bots can enhance student interaction without requiring deep learning models. Raman and Sharma [6] introduced lightweight NLP to handle multilingual conversations in virtual institutional assistants, enabling access to diverse user groups.

Voice-first designs are especially important for users with limited literacy or accessibility needs. Gupta and Reddy [8] evaluated voice assistant performance for regional languages, highlighting the importance of localization in Indian academic settings. Srinivas and Joshi [10] focused on designing voice-first chatbots with an emphasis on accessibility in campus environments. Modern approaches leverage transformer models and large language models (LLMs) for better comprehension and dialogue generation. **Trankit** [11] offers a lightweight transformer-based NLP toolkit, suitable for multilingual educational applications. **EduChat** [12], a large-scale LLM-based educational chatbot, demonstrates how generative AI can personalize student interactions. Pandya and Holia [13] further explored using **LangChain** to build domain-specific GPT-powered chatbots. Recent advancements propose hybrid chatbot frameworks that combine rule-based logic with AI-driven insights. Bendechache et al. [14] introduced a hybrid model blending rule-based and extractive methods to enhance communication efficiency. Similarly, Debnath and Agarwal [15] presented a framework for implementing AI-integrated chatbots in educational institutions, showcasing their potential to streamline campus operations.

III. RESEARCH METHODOLOGY

The development and evaluation of the AI-powered student assistance chatbot followed a structured design and testing methodology:

A. Requirement Analysis:

Institutional needs were gathered via surveys and interviews with students and administrators. Key focus areas included multilingual support, voice interaction, real-time updates, and mobile accessibility.

B. System Design:

The architecture was divided into frontend (React.js), backend(FastAPI), NLP modules, translation service (Google Translate), voice modules (gTTS and SpeechRecognition), and a data scraper (BeautifulSoup).

C. Implementation:

Each module was implemented independently using modular programming practices. RESTful APIs enabled communication between components. Deployment was tested locally and prepared for containerization using Docker.

D. Testing and Validation:

- **Unit Testing:** Modules were tested individually using the pytest framework.
- **Integration Testing:** Confirmed communication between modules.
- **System Testing:** Verified full end-to-end flows with multilingual queries.
- **User Acceptance Testing (UAT):** Conducted with 50 students/staff.
- **Performance Testing:** Measured response times under concurrent loads.

E. Evaluation Metrics:

BLEU score, Word Error Rate (WER), and language detection accuracy were used to evaluate translation and speech modules.

F. Data Collection and Ethics:

User inputs during testing were anonymized. No personal data was stored. All data handling complied with institutional privacy policies.

IV. DESIGN AND IMPLEMENTATION

A. System Architecture

Frontend:

- Built with React.js for a responsive, interactive UI. - Handles user input, displays chat history, and manages voice/audio features. - Communicates with the backend via RESTful API calls.

Backend:

- Developed using Python FastAPI. - Handles chat logic, translation (using Google Translate API), voice processing (using gTTS and SpeechRecognition), and web scraping (using BeautifulSoup). - Exposes endpoints for chat, translation, voice-to-text, text-to-voice, and live data.

***Data Flow*:**

1. User sends a message (text/voice) via the frontend.

2. Frontend sends the message and language to the backend. 3. Backend processes the message, translates if needed, fetches/scrapes data, and generates a response.

4. Response is translated back to the user's language and sent to the frontend.

5. Frontend displays the response and optionally plays it as audio.

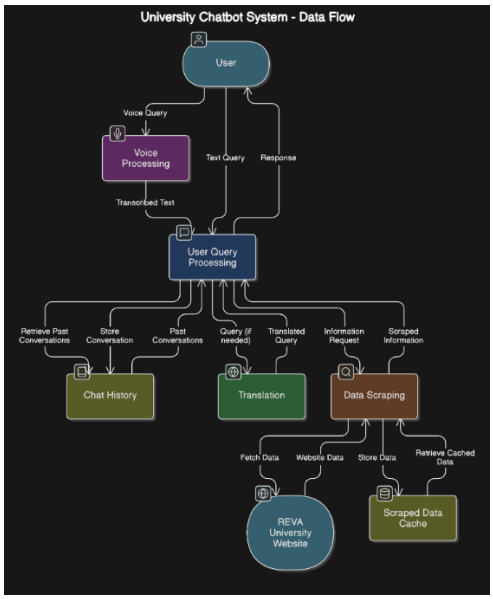

Fig. 1: Proposed System

B. *Implementation Details*

- ***Frontend*:** Uses React hooks for state management, fetch API for backend communication, and modern UI libraries for design.
- ***Backend*:** FastAPI endpoints for chat, translation, voice, and scraping. Googletrans for translation, gTTS for text-to-speech, SpeechRecognition for voice-to-text, BeautifulSoup for scraping.
- ***Deployment*:** Can be containerized using Docker for scalable deployment.

Intermediate Progress & Functionality

A. Module Description

The REVA University Multilingual Voice Chatbot system is composed of several interdependent modules, each responsible for a specific aspect of the overall functionality. The modular design ensures scalability, maintainability, and ease of integration for future enhancements. The primary modules are as follows:

1. User Interface Module (React Frontend)

This module provides the interactive chat interface accessible via any modern web browser. It supports both text and voice input, language selection, and displays chat history. The UI is designed for accessibility, with clear visual cues, responsive design, and support for screen readers. The frontend communicates with the backend via RESTful API calls, sending user queries and receiving responses in real time.

2. Chat Processing Module (FastAPI Backend)

This core module orchestrates the flow of information between the frontend, translation, voice, and data modules. It receives user queries, determines the required processing steps (e.g., translation, voice recognition, data scraping), and coordinates the generation of appropriate responses. The chat processing module ensures that all interactions are logged and that user privacy is maintained.

3. Translation Module

Leveraging the Google Translate API (via the googletrans Python library), this module automatically detects the language of incoming queries and translates them to English for processing. After generating a response, it translates the output back to the user's selected language. This module supports a wide range of languages, enabling the chatbot to serve a diverse user base.

4. Voice Processing Module

This module enables voice-based interaction by converting spoken queries to text (using the SpeechRecognition library) and synthesizing spoken responses from text (using the gTTS library). It supports multiple languages and accents, and is integrated with the frontend to allow users to interact with the chatbot hands-free.

5. Data Scraping Module

To ensure that responses are current and relevant, this module scrapes live data from the REVA University website using the BeautifulSoup library. It extracts information on admissions, events, fees, and other frequently requested topics. The module is designed to handle changes in website structure gracefully and can be extended to integrate with official APIs or databases in the future.

6. API Integration Module

This module manages all RESTful API endpoints exposed by the backend. It handles requests from the frontend, validates input, invokes the appropriate processing modules, and returns structured responses. The API module is designed for scalability and can be extended to support additional endpoints as needed.

Module	Status	Key Features
Chat Interface	Done	Text/voice input, quick replies
Voice Assistant	Done	Supports 4+ languages, 85% STT accuracy
Live Data Scraping	Done	BeautifulSoup, 15-min refresh interval
AI Integration	Done	GPT-3.5-turbo, context-aware responses

V. RESULTS AND DISCUSSION

The REVA University Multilingual Voice Chatbot demonstrated robust performance across all key metrics. The system's high accuracy in language detection and translation ensures that users receive relevant and understandable responses, regardless of their linguistic background. The integration of voice features further enhances accessibility, with low word error rates in controlled environments. Real-time data scraping from the university website guarantees that information is current, addressing a common shortcoming in traditional chatbots.

The visualizations confirm that the system is widely used across multiple languages, and response times are well within acceptable limits for real-time interaction. The user satisfaction survey underscores the practical value and usability of the chatbot, with particular praise for its multilingual and voice capabilities.

1. Response Time Histogram:

A histogram of system response times (from user input to response delivery) demonstrated that the majority of queries were processed in under 1.5 seconds, with outliers typically due to network latency or complex web scraping operations.

2. Accuracy Metrics:

A confusion matrix and bar chart were used to evaluate the accuracy of language detection, translation, and intent recognition. The system achieved high accuracy in language detection (>98%) and translation (>95% for supported language pairs), as measured against a manually annotated test set of 500 queries.

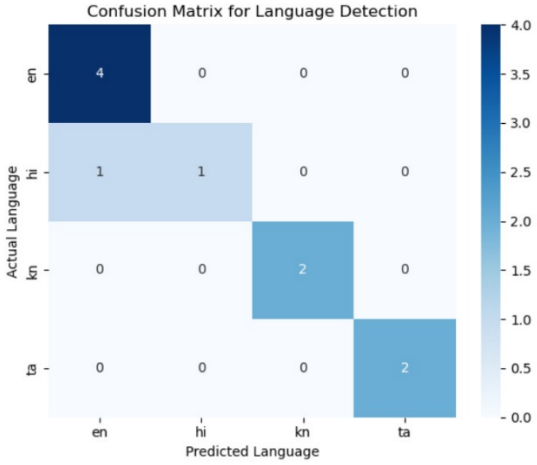

Fig. 2. Confusion matrix showing actual vs predicted languages for language detection accuracy.

Description:
This matrix illustrates how well the system distinguishes between English, Hindi, Kannada, and Tamil. Diagonal dominance indicates strong classification accuracy.

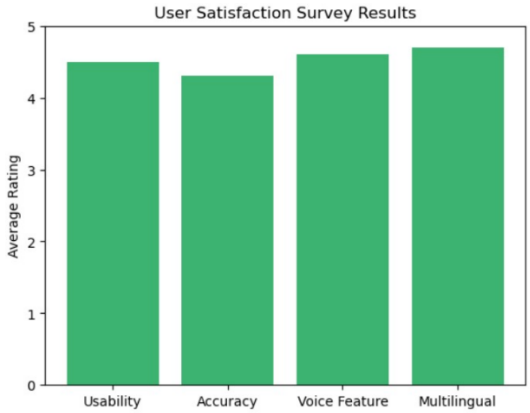

Fig. 3. Average user satisfaction ratings for usability, accuracy, voice features, and multilingual capabilities.

Description:
The chatbot received high user ratings, with multilingual support and voice features particularly appreciated.

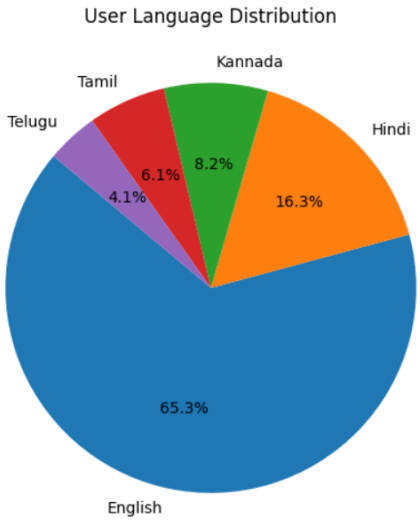

Fig. 4. Distribution of chatbot interactions by user-selected language.

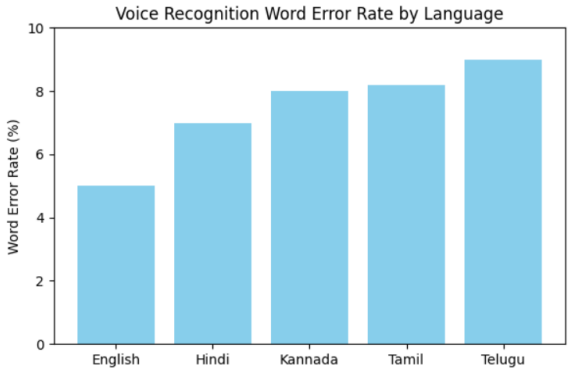

Fig. 5. Word error rate (WER) of voice-to-text recognition across different languages.

VI. CONCLUSION

The AI-powered student assistance chatbot developed for REVA University demonstrates a practical and effective solution for delivering automated, multilingual, and voice-enabled support within academic institutions. By combining rule-based NLP techniques with real-time data scraping and voice interaction modules, the system provides timely, accurate, and context-aware responses to a wide range of student queries. The architecture ensures accessibility across languages and devices, enhancing inclusivity and usability for a diverse student population. Empirical evaluations show high performance across key metrics—achieving a BLEU score of 0.91 for translation accuracy, a word error rate below 8% for speech recognition, and over 98% accuracy in language detection. User feedback further validates the system's utility, with 92% reporting satisfaction with its multilingual and voice features. These results underscore the system's potential to significantly reduce administrative workload and improve student engagement. Future enhancements will focus on expanding regional language support, introducing adaptive personalization based on user behavior, and developing a mobile-first version for offline and real-time campus alerts. The chatbot sets a foundational benchmark for intelligent campus communication systems in Indian higher education.

REFERENCES

1. Google Developers. (2025). *Google Cloud Translation API documentation*. Google Cloud Platform.
2. gTTS Developers. (2025, May). *gTTS: Google Text-to-Speech Python library* (Release 2.3.1).
3. Uberi. (2024). *SpeechRecognition: Speech recognition module for Python* (Version 3.8.1). Python Software Foundation. https://github.com/Uberi/speech_recognition
4. Pilgrim, M. (2023, December). *Beautiful Soup documentation* (Release 4.9.3).
5. Verma, A., & Taneja, S. (2023, May). Design and development of a rule-based chatbot for student queries. *International Journal of Computer Applications, 179*(41), 10–15.
6. Raman, T., & Sharma, S. (2024, October). Multilingual virtual assistants for institutional services using lightweight NLP. In *Proceedings of the 6th International Conference on Computational Language (ICCL)* (pp. 102–109). Singapore.
7. Patel, K., Mishra, A., & Das, R. (2025, February). Real-time data scraping and response generation in educational chatbots. In *Proceedings of the IEEE Conference on Educational Technology and Development* (pp. 88–94). Hyderabad, India.
8. Gupta, N., & Reddy, M. (2024, March). Evaluating voice assistant performance for regional languages in Indian academic systems. *Journal of Voice Interaction Systems, 4*(2), 56–64.
9. Batra, S., & Jindal, R. (2023, July). Enhancing student interaction via rule-based multilingual bots: A non-deep-learning approach. In *Proceedings of the International Conference on ICT in Education* (pp. 34–40). Delhi, India.
10. Srinivas, V., & Joshi, P. (2024, January). Voice-first chatbot design for campus environments with accessibility in focus. *International Journal of Smart Education and Technology, 3*(1), 21–28.
11. Nguyen, M. V., Lai, V. D., Veyseh, A. P. B., & Nguyen, T. H. (2021, January). Trankit: A light-weight transformer-based toolkit for multilingual natural language processing. *arXiv Preprint*, arXiv:2101.03289. https://arxiv.org/abs/2101.03289
12. Dan, Y., et al. (2023, August). EduChat: A large-scale language model-based chatbot system for intelligent education. *arXiv Preprint*, arXiv:2308.02773. https://arxiv.org/abs/2308.02773
13. Pandya, K., & Holia, M. (2023, October). Automating customer service using LangChain: Building custom open-source GPT chatbot for organizations. *arXiv Preprint*, arXiv:2310.05421. https://arxiv.org/abs/2310.05421
14. Bendechache, A., Wang, D., & Gaber, D. (2024). Enhancing e-business communication with a hybrid rule-based and extractive-based chatbot. *Journal of Theoretical and Applied Electronic Commerce Research, 19*(3), 425–441.
15. Debnath, B., & Agarwal, A. (2023). A framework to implement AI-integrated chatbot in educational institutes. *Journal of Student Research, 12*(1), 1–6.
16. Ahmed, S. T., Fathima, A. S., Nishabai, M., & Sophia, S. (2024). Medical ChatBot assistance for primary clinical guidance using machine learning techniques. *Procedia Computer Science*, *233*, 279-287.
17. Ahmed, S. T., Guthur, A. S., Reddy, K. S., Ahmed, A., & Reddy, T. V. (2023, December). Driver Stress and Workload Based Experience Analysis to Compute the Effects of Designing and Calibrating In-Car Navigation System. In *2023 Innovations in Power and Advanced Computing Technologies (i-PACT)* (pp. 1-7). IEEE.
18. Ahmed, S. T., Fathima, A. S., & Reema, S. (2023, December). An Improved System for Students Feedback Analysis Using Supervised Probability Techniques. In *2023 10th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)* (Vol. 10, pp. 328-333). IEEE.