RESEARCH ARTICLE OPEN ACCESS

Campus Buddy – AI Powered Bot for College Information

Pallavi[1], Sanjana S[2], Shreya H S[3], Radhika Dattatraya Balemane[4],
Prof. Navile Nageshwara Naveen[5]
[1,2,3,4] Department of Computer Science & Engineering,
[5] Assistant Professor, Department of Computer Science & Engineering,
Jyothy Institute of Technology, Bangalore

1. Abstract

This project presents the design and implementation of a local, secure, speech-enabled campus enquiry and navigation assistant. The system integrates a custom knowledge base built for our institutional environment and offers students an intuitive interface for obtaining college information, navigating to departments, and accessing academic details. By combining speech input, structured database retrieval, and natural conversational output generation, the system aims to reduce onboarding friction for newly admitted students. The backend is designed for local deployment due to institutional data security considerations, with modularity and scalability prioritized to allow future expansion.

Keywords: *Chatbot, AI Assistant, RAG, Avatar Video Bot, NLP, Virtual Campus Guide, Deep Learning, Voice Interaction.*

2. Introduction

Newly admitted students frequently struggle to locate departments, facilities, and important offices within the campus. Traditional navigation methods such as printed maps or administrative inquiries lead to inefficiency, especially during admission periods. To address this, we developed a conversational campus assistant that provides speech-based query handling, database-driven responses, and intuitive guidance. Unlike generic chatbot systems, this solution uses a custom dataset curated specifically for our institution, ensuring relevance, accuracy, and privacy.

This work builds upon concepts from conversational AI, campus navigation systems, and database-driven dialogue agents. The system is implemented using a React-based frontend, an Express/MongoDB backend, and a flexible retrieval layer that supports both direct database querying and optional local LLM assistance.

3. Literature Survey

Early enquiry chatbots for college systems primarily utilized rule-based structures and AIML-driven templates to interpret student queries. These systems relied on pattern matching and keyword triggers, offering limited flexibility and struggling with new or ambiguous questions. Though functional, they lacked adaptability and provided no emotional or personalized interaction.

Later studies introduced web-based chatbot interfaces incorporating lightweight NLP and database retrieval techniques. These systems automated common queries related to admissions, academic timetables, fees, and examination schedules. They reduced manual workload but operated on static datasets, offering no dynamic response generation or multimodal interaction.

Machine learning enhanced these systems as neural networks and NLP pipelines enabled improved intent detection and context understanding. Models trained on categorized student queries achieved higher accuracy and consistency. Preprocessing techniques such as stemming, tokenization, and bag-of-words representations contributed to more robust classification of inputs. These neural chatbots were deployed in universities to deliver timely information, yet they remained restricted to text responses and offered minimal engagement.

Recent advancements shifted toward Retrieval-Augmented Generation (RAG) and large language model (LLM)–powered systems. These models integrate embedding-based retrieval and generative reasoning to produce contextually

 ©IJSRED: All Rights are Reserved

accurate answers derived from institutional knowledge bases. RAG architectures demonstrate significantly higher performance on complex and multi-turn queries, and their ability to update dynamically aligns well with evolving academic information.

Parallel research in multimodal AI introduced audio-visual chatbots that integrate speech recognition, text-to-speech synthesis, and animated facial expressions. Systems like AVIN-Chat exemplify how emotional modulation and 3D avatar animation enhance user experience by mirroring human conversational cues. These improvements make interaction more natural and accessible, particularly for first-year students or individuals uncomfortable with formal administrative communication.

Neural Radiance Fields (NeRFs) have further enabled realistic video-based conversational agents capable of generating synchronized lip movements, head gestures, and emotive expressions. Such systems provide face-to-face virtual assistance suited for counselling sessions, academic guidance, and personalized query resolution.

Immersive virtual navigation tools developed using spherical panorama imaging offer remote campus exploration through 360-degree visuals and interactive hotspots. These systems improve accessibility for prospective students and complement conversational agents by visually supporting campus-related enquiries.

Finally, AI-driven academic and career guidance chatbots using SVM, TF-IDF, and structured knowledge bases demonstrate how automated assistants can scale to manage large volumes of queries consistently.

Across all studies, a clear trend emerges: the transition from static, text-based chatbots to deeply personalized, intelligent, and multimodal AI companions capable of natural communication. This evolution

Figure 1: Flow Diagram

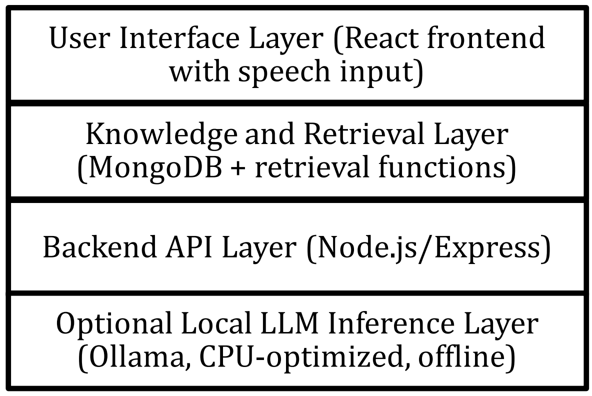
underscores the motivation for Campus Buddy—a video-based, character-driven academic assistant designed to embody friendliness, clarity, and emotional presence.

4. System Overview

The system follows a modular client–server architecture consisting of for primary layers:

Figure 2: System Architecture

The design supports local execution for data security, avoids cloud dependencies, and allows easy dataset expansion through structured model files.



4.1 System Architecture

User interactions begin with speech input captured at the frontend. The system transcribes the audio to text (using browser speech APIs). The backend receives the query, identifies intent, fetches corresponding information from the database, optionally enhances the response using a local LLM, and returns a conversational output to the user.

The entire architecture is optimized for portability, simplicity, and institutional data safety.

 ©IJSRED: All Rights are Reserved

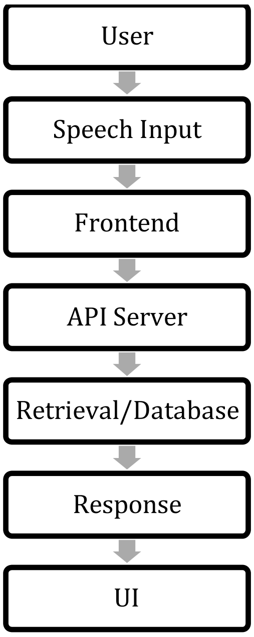


Figure 3: Block Diagram

4.2 Design Choices

- Local deployment for data confidentiality
- Modular models for easy extension when new departments/facilities are added
- CPU-level LLM inference to avoid GPU constraints
- Simple, maintainable directory structure
- React-based UI for rapid updates and clean design
- JWT-secured authentication for accessing personalized features

5. Implementation

5.1 Frontend (React)

The frontend serves as the primary interaction point. It includes:

- Speech input using browser APIs
- Dynamic chat UI
- Quick actions for frequently asked queries
- Avatar-based visual interface (optional enhancements)
- Pages for chat, college info, and user profile

Source reference: frontend/src/ directory including components, pages, and contexts.

5.2 Backend (Node.js + Express)

The backend is responsible for:

- API routing for authentication, college info, and chat history
- JWT-based authentication
- Database queries to retrieve structured information
- Handling the /ask endpoint for query processing

Key modules include:

- index.js (server entry)
- authRoutes.js, chatHistoryRoutes.js, chatRoutes.js, collegeRoutes.js
- auth middleware for token verification

The backend is intentionally lightweight, enabling deployment on campus machines without cloud dependencies.

5.3 Knowledge Base (MongoDB Models)

The system uses structured data stored across multiple collections such as:

- Departments
- Courses
- Navigation paths
- Facilities
- Events
- Rankings
- Scholarships
- Placements

These models allow fine-grained retrieval and updating as the college evolves.

5.4 Local LLM Integration

The system optionally integrates with a local Ollama-based LLM for:

- Making responses more conversational
- Summarizing database text
- Handling fuzzy queries

The LLM runs entirely on CPU to comply with institutional constraints. The retrieval system ensures that the response remains grounded in the verified database content.

6. Data Flow

1. User speaks the query in the UI.
2. Speech is converted to text.
3. Text is sent to the backend via API call.
4. Backend processes the intent and retrieves corresponding data from MongoDB.
5. (Optional) Local LLM refines the response into conversational form.
6. Backend returns the final message.
7. UI displays the answer to the user.

This flow maintains both speed and correctness while always prioritizing data security.

 ©IJSRED: All Rights are Reserved

7. Discussion

The system addresses several key challenges:

- New students struggle with campus orientation
- Large campuses require scalable navigation systems
- Sensitive institutional data cannot be handled through cloud LLMs
- Staff workload increases when new batches arrive

Our implementation directly reduces onboarding workload, provides an accessible interface, and allows flexible data expansion. The architecture also enables future additions such as:

- Image-based navigation
- AR campus tours
- Multilingual query support

The design intentionally keeps the computational footprint light so it can run inside campus networks with modest hardware.

8. Conclusion

A fully local, speech-driven, flexible chatbot system for campus navigation and enquiries was implemented successfully. The modular architecture allows seamless future scaling, and the custom-built knowledge base ensures domain accuracy. The system is tailored to institutional needs, secure by design, and highly extensible.

9. References

[1] T. P. Mothankar, P. S. Maski, and P. D. Dhote, “AI-powered Chatbot for College Enquiry Using NLP and Machine Learning,” International Journal of Innovative Research in Computer and Communication Engineering.

[2] C. Park, “AVIN-Chat: Audio-Visual Interactive Chatbot with Emotive Avatars,” Proceedings of the ACM Conference on Human Factors in Computing Systems.

[3] Y. Du et al., “UniPi: Visual Planning Agents via Text-Conditioned Video Generation,” Conference on Neural Information Processing Systems (NeurIPS).

[4] Y. Yan, “DialogueNeRF: Realistic Face-to-Face Avatar Conversations with Responsive Behaviors,” IEEE Transactions on Multimedia.

[5] S. Neupane, “BARKPLUG V.2: A Retrieval-Augmented LLM-based Chatbot for Personalized University Query Handling,” Mississippi State University.

[6] W. Hassan et al., “Improving Accuracy of College Info Systems using Ensemble Learning Techniques,” Conference on Intelligent Systems.

[7] Y. Dan et al., “Addressing Limitations in Educational Chatbots with Domain Understanding,” Proceedings of the EdTech AI Conference.

[8] L. Chopra and S. Chakraborty, “PARIMA: Predictive Viewport Allocation for Efficient 360° Video Streaming,” Multimedia Systems Journal.

[9] L. T. Lin, “Interactive Minecraft-based Campus Tours for Enhancing Student Engagement,” International Journal of Virtual Learning.

 ©IJSRED: All Rights are Reserved