AI-Based College Information Assistance Chatbot

Vidyashree N
Assistant Professor, Department of ECE
Dr. Ambedkar Institute of Technology
Bangalore, India
vidyashreen.ec@drait.edu.in

Sohan Kumar A
Department of ECE
Dr. Ambedkar Institute of Technology
Bangalore, India
sohankumar2580369@gmail.com

Mohan D
Department of ECE
Dr. Ambedkar Institute of Technology
Bangalore, India
mohandvg100@gmail.com

Naveen Araganji
Department of ECE
Dr. Ambedkar Institute of Technology
Bangalore, India
naveenaraganji2003@gmail.com

Nithin N J
Department of ECE
Dr. Ambedkar Institute of Technology
Bangalore, India
njnithin6@gmail.com

***Abstract*—The quick and real-time accessing academic and administrative information is necessary in educational institutes, which can be done through our project 'AI Based College Information Assistance Chatbot' helps to reduce asking for manual queries, waiting for emails and relaying on notice boards, Chatbot integrated with Natural language processing (NLP) and the structured college database aims to build dual platform architecture, Helpdesk – is made of web based interface integrating with Raspberry Pi, touched screen display and microphone supporting with speech to text text to speech for both input query and output response respectively, This Helpdesk easier to accessible users and helps in used for impaired persons. The Chatbot is tested for reliability, usability, response accuracy. The results of the Chatbot are significant: response accuracy, information access speed, to reduce the traditional communication for quick and real-time access to academic and administrative information. Chatbot demonstrates the ability to modernize campus by addressing academic and administrative information and supports future integration with institutional services.**

I. Introduction

Many educational institutions depend on timely and accurate information about academic and administrative information. However, traditional communication, like relaying on noticeboards, emails and manual queries, fails to obtain instant and real-time academic or administrative information. These traditional types of communication lead to delays and a lack of real-time access. In peak times, like admissions and examinations, it's hard to find specific information. To overcome these challenges, our project aims to create real-time and quick information accessibility.

As educational institutions increase, their scale and complexity are needed for an intelligent, centralized and user-friendly information retrieving system. Emerging technologies like Artificial Intelligence (AI) and Natural language processing (NLP) provide an opportunity to automate educational institution communication.

Our project 'AI Based College Information Assistance Chatbot' desired to resolve these challenges like real-time, quick access to information. Chatbot is implemented with a dual-platform solution: integrated with the college website, which can be accessed remotely and Helpdesk using Raspberry Pi for on-campus use, enabling access for students, faculty, staff and visitors. The chatbot, including NLP techniques, college database and speech to text and text to speech modules, handles queries related to faculty information, timetables, syllabus, exam schedules, announcements and other educational services.

The key contributions of the project:

1. Development of college-specific AI Chatbot for college-related queries
2. Help desk: on-campus access environment
3. Integrated with college website to access remotely

II. Literature Review

A lot of research explored the use of AI and NLP to automate retrieving information in educational institutions. Previous works are based on rule-based and pattern matching systems. Recent research integrates machine learning and deep learning models for quick and real-time access to information.

Sonawane and Shanmughasudaram(1) introduced 'ChatBot for College Website' based on the LSTM model which is a sequence-to-sequence model. Their system, which takes user queries and converts them into vector word embedding and response as an encoder and decoder architecture. Rashinkar et al.(2) proposed a 'Design of Chatbot System for College Website' which uses Natural language processing (NLP) and Navive Bayes classifier for user queries. Their system includes NLP, database lookup for a rule-based approach. Karunamurthy et al.(3) introduced an 'intelligent college-enquiry chatbot uses NLP' using NLP and Machine learning models supported in feedback loops for continuous improvements. Tiwari et al.(4) introduced an 'AI Chatbot for College Enquiry'. This system includes neural networks and NLP to classify user queries related to departments, admissions, academic schedules, fee structure and campus facilities like our project database, but the system emphasizes tokanization and word embedding to improve response accuracy. Recently, Kumar et al.(5) presented 'StudentEase: An AI-Powered College

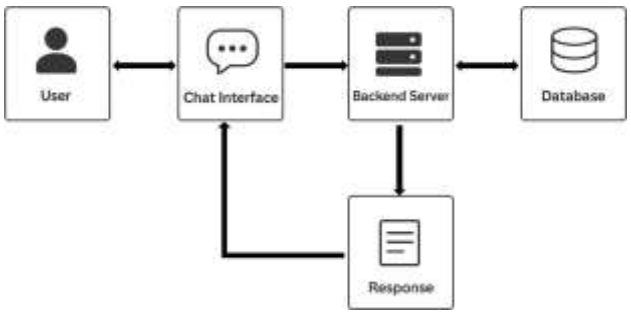

Fig. 1. Enter Caption

Helpdesk Chatbot', an advanced college helpdesk chatbot that includes semantic matching and ollama-based generative models for natural response generation. Their system includes word embedding and interfaces for campus information retrieval.

These studies highlight the growing adoption of AI-driven educational environments, where earlier systems were mainly based on rule-based approaches, where modern solutions increasingly include machine learning, semantic embedding, and NLP techniques. The gaps that are identified include limited multi-platform availability, inadequate voice-based support, and a lack of reading the schema of the database. Our project objects aims to fill these gaps by working on a dual platform with text and voice interaction capabilities and reading the schema of the database can be generated SQL queries using Artificial. Intelligence.

III. Methodology

The system is designed for students to access real time data about their college and academics. It combines the power of AI and SQL to understand and give accurate results. The workflow inside the chatbot system is divided into several blocks to understand, process, retrieve and respond to the user. Let us look at the system architecture.

The system is based on client-server architecture. From fig. 1, we can see that the user interacts with the chatbot using a simple web page. The user message or queries is forwarded to the backend of the server where it processes the query using AI and generates an SQL query. That query is used to fetch the data in the database. Finally the raw data is processed to give a final response through the same chat interface.

Workflow: The workflow inside the chatbot is represented in the flowgraph. Fig. 2 shows the backend system processes the user's query in the following stages.

1) User query: the chatbot takes the user's query using a webpage interface and forwards it backend server.
2) NLU: it's where the AI processes the user query and understands what the user is looking for and based on that it generates an SQL query to search Database. It ensures proper generation of SQL queries which don't affect the database.

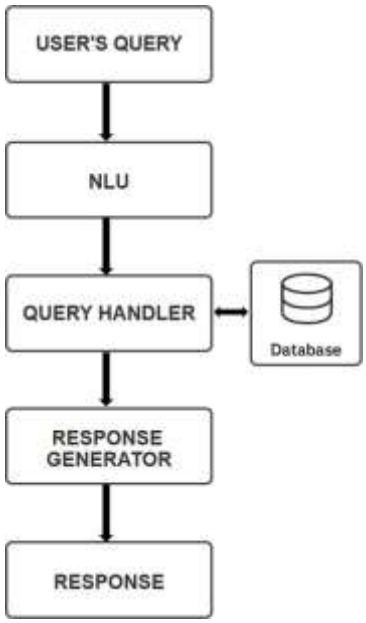

Fig. 2. Backend Dataflow

3) Query handler: the generated SQL query is checked for syntax errors and validation. After that the query is executed in the Database. After that it returns the result of that query and passes it to the next stage to process this raw data.
4) Response generator: after receiving the raw data, it is passed to an LLM with user query. The LLM compares the query and DB data, processes it and generates a meaningful response. If they are not related then it ask for rephrasing of the user's question.
5) Response: finally the generated response is given back to the user through the web interface.

Implementation

A. Hardware Tools

- Raspberry Pi 4B: It is a single-board computer that is powerful with a CPU to process and execute the backend chatbot, data communication, and text and voice communication input/output.
- Display module: The display we are using is the touch screen display having a size of 7 inch allowing users to have a visual interaction with the chatbot. It will display the webpage of the chatbot, answers user questions.
- Microphone: The microphone is used to engage the users orally. It receives voice input on the part of the user, the voice data is translated to text, in order that the chatbot can process spoken queries. It enhances the access of chatbot particularly to those clients who feel more comfortable using their voices.
- Speaker: The speaker will be applied to audio output of the chatbot responses. It will read out the answers given by the chatbot and thus make the system more user-friendly.

B. Software frameworks

- Flask: is a lightweight Python web application which is applied to create the backend of the chatbot system. Flask in this project does the routing of the chatbot interface and server logic, API calls (OpenAI or Rasa), user queries, and routes 12 to the database to retrieve information like class schedules or faculty. It is the heart of movement of communication between the frontend and the backend.
- MySQL: this is the structured database system that is used to store all the college specific data. This contains such information as the schedule of classes, the contacts of the faculty, announcements, and exams. The chatbot uses Flask to query this database in real time and get information related to what the user needs and get them in the form of a conversation.
- Web Speech API: is applied to the web front end to respond to voice input. It allows the users to talk their queries into a microphone and the browser converts them to written text and forwards it to the chatbot. This makes the system more user friendly and easy to use among the students who are used to talking than typing.
- Google TTS (Text-to-Speech): is a web-based application that transforms the text messages that are sent to the chatbot into natural speech. In your project, it is applied to voice output, so the chatbot can say out loud its response, which would be practical with the Raspberry Pi kiosk, which would be more interactive and inclusive.
- Render: is a cloud computing service, where your chatbot Flask back-end and Rasa server can be hosted and made publicly available using the internet. It guarantees that web interface and the Raspberry Pi kiosk is able to communicate with the same server and therefore makes the system scalable, reliable and easy to update remotely.
- GitHub: is utilized in the version control and collaboration in development. It keeps your source code, monitors changes and also lets you roll back in case of necessity. It also allows collaborating with the team and automatization of deployments, such as updating GitHub to Render.

This system contains a multi user category login system for different users like guest, students, faculties and admin. Based on the user they can login with their college domain mail IDs. Each category has different options and use cases. In fig. 3 we can see that we have login options at top right corner. The user will select and login with mail IDs. After that based on the category, the UI and options will change.

1) Guest mode: It is the basic and by default mode where the user only has basic collage information like college infrastructure, facilities, courses, placements etc. The user may be a student or their parents trying to know about college. This mode will help them to get basic information like before joining college without logging in.
2) Student mode: The student mode is more privileged than guest mode. Once students get admitted, they get their college domain email ID. They can use it for logging in and access more features like study materials, academic details, syllabus, timetables, results, etc.
3) Faculty mode: In faculty mode, the faculty of the college can log in with their college domain mail IDs and have features like adding study materials, take attendance, and generates attendance reports directly from the webpage, as shown in fig. 5. There is No need for traditional method of taking attendance in a logbook and calculating attendance percentages. They take, check, and generate attendance reports and send it to connected student mail IDs.
4) Admin mode: In admin mode the college admin can manage and update academic data like they can manage and update academic timetables, syllabus, add the latest news.

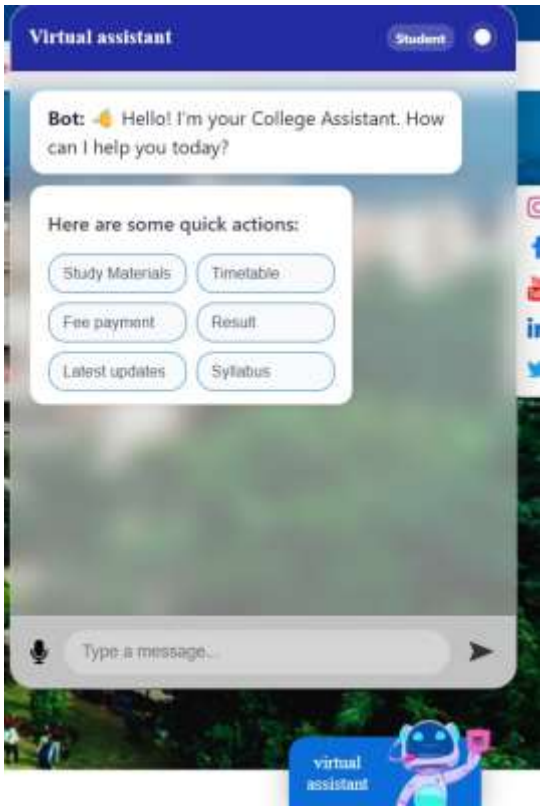

Fig. 3. Chatbot interface

Results and Discussion

The system was tested for accuracy, speed, usability, and robustness on both the web version and the Raspberry Pi kiosk, and the outcomes are described below. In terms of response accuracy, tests on 120 real student and faculty queries showed that queries made by faculty were about 93% correct, exam timetable questions were answered correctly 92% of the time, class time schedule queries achieved about 95% accuracy, general college information queries had 96% accuracy, department-specific questions were about 90% accurate, and the overall accuracy level was approximately 93%. For system

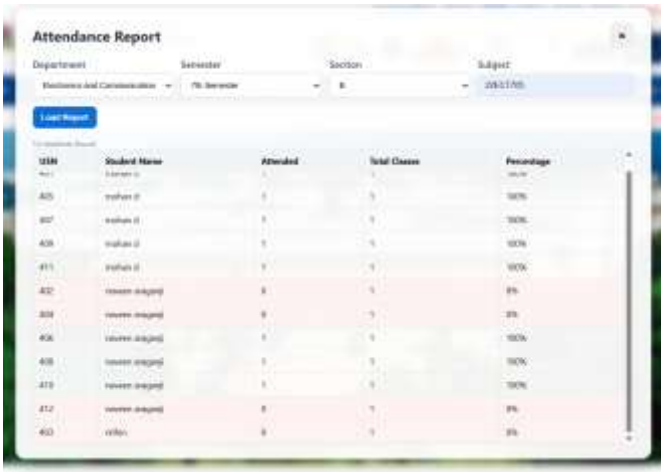

Fig. 4. Attendance report page (faculty mode)

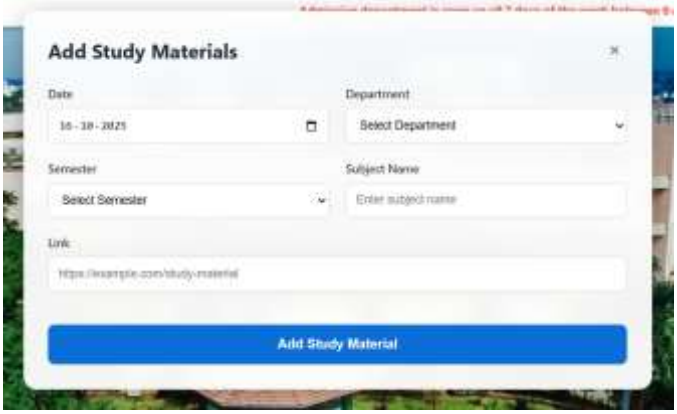

Fig. 5. Adding study materials (faculty mode)

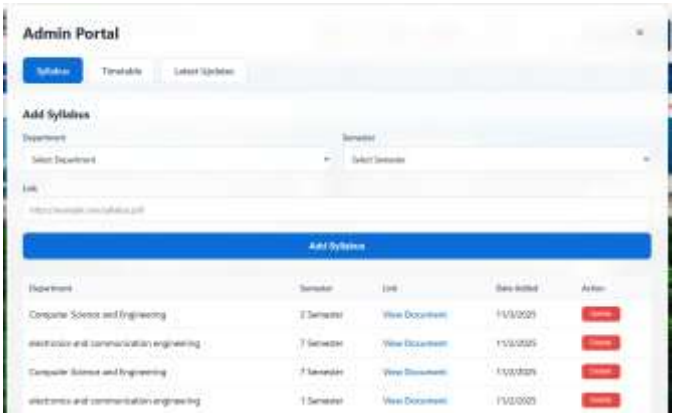

Fig. 6. Admin portal

latency and performance, text queries on the web application took less than 1 second to answer, while on the Raspberry Pi they took an average of 1.3 seconds; voice queries, including speech-to-text and response generation, needed about 1.8 seconds on the web and 2.5 seconds on the Raspberry Pi, and database retrieval was consistently fast with an average of 0.4 seconds per query.

Voice interaction quality was evaluated in three noise conditions: in a quiet indoor room, the recognition accuracy was about 96%, in a normal classroom it was about 89%, and in a busy corridor it was about 74%, showing that performance was strong in controlled conditions but degraded where there was significant background interference. User experience feedback was collected from 50 students and 10 faculty members; around 92% of students found the system easy to use, about 88% felt it saved time compared to manual searches, about 84% liked the voice interaction, and about 90% said it was helpful for exam and timetable announcements, while faculty members reported fewer repeated student queries, valued centrally delivered information, and preferred the kiosk to be placed in high-traffic locations.

Overall, the discussion indicates that the chatbot greatly increases campus information availability compared to traditional channels such as notice boards or face-to-face inquiries, due to its high accuracy, low response time, and support for both text and voice interaction on web and kiosk platforms, which makes it suitable for real-world deployment, although challenges remain in terms of reduced voice accuracy in noisy environments and the lack of ERP integration for personalized data, which are suggested as directions for future improvement.

CONCLUSION

This work successfully provides its objective of accessing college-related information quickly and accurately. A website and Raspberry Pi-based kiosk are built to form a system where the staff and students can access the information instantly and accurately through text or voice from these two developed platforms. The real-time responses are obtained by integrating these two platforms. This system is organized by developing the backend and structured data, where it handles the common queries such as faculty details, timetables, etc. Also, the college ERP system is integrated with this system, where it provides handling of queries such as student attendance, student records and many other such queries. The addition of both text and voice support has enhanced the use of this system. This system makes an AI-based college information assistance chatbot. This system provides a good impact on communication within the college. This minimizes manual inquiries and saves time for both the faculty and students. Since this system provides a voice-enabled interface, it helps in the improvisation of accessibility. Overall, this system provides a good impact on accuracy and quick interactions between the user and chatbot. Though this system performs well, there are certain limitations. The responses of the user queries heavily depend on the completeness of the stored

database; therefore, the database should be updated regularly. Accuracy of voice recognition may vary in noisy areas such as in busy campus environments. The smooth performance of this system depends on the internet connectivity, which can be an issue during network issues. In future works, these limitations can be improved.

FUTURE WORK

1) Improvement of system scalability and performance: In the case of limited users based on the system, the system responds quickly and accurately. But, in the case of many users, the optimization of the backend must be done for the system to respond quickly and accurately to the user queries. In future work, the backend can be optimized by reducing the delay of the responses; there are a few techniques to reduce the response delay. The techniques include indexing of the database, caching of queries, and smarter handling of API. These improvements help in providing scalability and performance.
2) Enhancing the accuracy of voice interaction: Though this system's voice interaction performs well, its accuracy may vary in noisy areas. So, to come across this issue, advanced speech-to-text models and techniques for filtering background noise to improve voice clarity have to be used in future works. These improvements enhance the accuracy of voice interaction.
3) Multi-language support: To provide access to a wider audience, also to visitors and parents, the multi-language support must be integrated to this system. The use of multilingual models of NLP in future works makes the chatbot to understand and respond in multi-languages. So, this helps the needs of educational institutions that require diverse languages. These improvements help in providing multilingual support.
4) Development of mobile application: To provide faster access to the services of the chatbot and also to add features such as exam alerts, timetable changes, etc. A dedicated mobile application must be implemented in the future works. The application can be developed such that the common FAQ's can be accessed offline. This makes the system accessible from anywhere easily. These improvements help in the development of mobile applications.
5) Development of OCR technology: New announcements and circulars can be extracted automatically from the scanned images or PDF's using OCR technology in the future works. This helps in the elimination of the manual updates of the database, minimizes human errors, and also improves the reliability of the responses. These improvements help in the development of OCR technology.
6) Predictive analytics to support students: To analyze the internal marks, attendance and many more such areas of students, machine learning models must be added in future work. It predicts the areas of students where the students may need external support, warns the students about low performance and also helps the students to stay on track in academia. These improvements make the chatbot an academic advisor rather than just a tool for information.

REFERENCES

[1] S. Sonawane and R. Shanmughasundaram, "ChatBot for College Website", International Journal of Innovative Technology and Exploring Engineering, vol. 8, no. 10, pp. 566–569, 2019.
[2] S. P. Rashinkar, N. D. Wanjol, S. N. Rane, P. P. Shinde, and S. A. Talekar, "Design of Chatbot System for College Website", International Journal of Computer Sciences and Engineering, vol. 9, no. 7, pp. 41–45, 2021.
[3] A. Karunamurthy, R. Ramakrishnan, S. Jothsana, and M. Roshni, "Intelligent college enquiry chatbot using NLP for enhanced student interaction", Journal of Advanced Research Engineering and Technology, vol. 3, no. 2, pp. 30–37, 2024.
[4] A. G. Kumar, K. L. Lakshmi, K. S. Priya, G. Devi, and G. R. V. Teja, "StudentEase: An AI-Powered College Helpdesk Chatbot", International Journal of Scientific Research in Engineering and Management, vol. 9, no. 3, pp. 1–9, 2025.
[5] R. Tiwari, R. Khandelwal, Y. Agrawal, V. Tiwari, and W. H. Bisen, "AI Chatbot for College Enquiry", International Journal of Engineering and Management Research, vol. 13, no. 2, pp. 90–93, 2023.
[6] Harith L. K, K. V. Raja, and G. S. Mamatha, "University chatbot with database integration using RASA", International Journal of Advance Research, Ideas and Innovations in Technology, vol. 7, no. 3, pp. 2365–2368, 2021.
[7] J. Singh, M. H. Joesph, and K. A. Jabbar, "Rule-based chatbot for student enquiries", Journal of Physics: Conference Series, vol. 1228, pp. 1–5, 2019.
[8] V. Y. Shetty, N. U. Polekar, S. U. Das, and S. Pansambal, "Artificially Intelligent College Oriented Virtual Assistant", International Journal of Technical Research and Applications, vol. 4, no. 2, pp. 305–310, 2016.