iJRASET

International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL
FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: VII Month of publication: July 2025

DOI: https://doi.org/10.22214/ijraset.2025.73396

Edubot Student Assistance Chatbot

Landa Praveena[1], K.S.S. Soujanya Kumari[2]

[1]Department Of Information Technology & Computer Applications, Andhra University College of Engineering(A), Andhra University, Visakhapatnam, Andhra Pradesh - 530003

[2] Assistant Professor Department of Computer Science & Systems Engineering, Andhra University College of Engineering(A), Andhra University, Visakhapatnam, Andhra Pradesh - 530003

Abstract: The rapid evolution of higher education demands innovative digital solutions that can streamline institutional operations, enrich student engagement, and optimize campus-wide service delivery. Andhra university, one of india's premier public universities with a large and diverse student body, experiences a constant influx of inquiries related to admissions, academic programs, examination schedules, campus facilities, and institutional regulations. Traditional channels of information delivery—such as email, phone calls, or in-person consultations—are often slow, inconsistent, and inefficient, leading to communication bottlenecks, delays in response times, and an overburdened administrative staff.

To address these challenges, edubot emerges as a comprehensive, ai-powered conversational assistant specifically tailored for andhra university. This project introduces a smart, interactive chatbot that operates 24/7, capable of understanding natural human language and providing timely, accurate, and consistent responses. Accessible via a user-friendly web interface, edubot leverages the power of natural language processing (nlp) and machine learning (ml) to deliver an intuitive, human-like conversation experience. Technologically, edubot is built using a robust django 5.x backend, a responsive front-end in html/css/javascript, and sqlite3 as its default database (with scalability to postgresql). The core intelligence is powered by the google gemini pro api, which allows the system to parse user queries, interpret context, and generate relevant answers dynamically. Experimental results show high accuracy in intent recognition and information retrieval, proving edubot's ability to significantly enhance both user experience and administrative efficiency.

This project underscores the transformative potential of ai-based conversational interfaces in revolutionizing information access and service automation within educational institutions.

Keywords: chatbot, deep learning, natural language processing, machine learning, educational technology, student services, information retrieval, andhra university, conversational ai, tensorflow, django, gemini api

I. INTRODUCTION

The digital transformation of higher education has accelerated in recent years, bringing new opportunities and challenges to universities across the globe. As institutions strive to modernize their infrastructure, streamline internal operations, and deliver personalized experiences to their stakeholders, the demand for intelligent automation tools has never been greater. One of the most critical areas affected by this shift is student communication and support.

Every day, universities receive a vast number of inquiries ranging from admission procedures and course registrations to examination dates, campus facilities, hostel availability, fee structures, and more. Addressing these queries manually—through physical help desks, phone calls, emails, and fragmented web pages—often leads to long wait times, inconsistent information, and significant workloads on administrative staff. This results in poor user experience, reduced institutional transparency, and operational inefficiencies.

Andhra university, a prestigious public university situated in visakhapatnam, andhra pradesh, india, exemplifies this challenge. Established in 1926, the university serves thousands of students from diverse regions and backgrounds. As the student population grows and digital expectations rise, the traditional methods of handling queries are no longer scalable or efficient. Students, parents, and faculty increasingly expect instant and accurate access to information, preferably through digital and mobile platforms.

In response to this evolving need, the edubot project was conceptualized and developed as a strategic solution. It is designed as a smart, ai-powered chatbot that offers real-time assistance via natural human language. Edubot acts as a 24/7 virtual helpdesk, enabling users to ask questions in plain english and receive prompt, relevant, and verified answers. The chatbot simplifies how users interact with the university's ecosystem, offering a more engaging and efficient alternative to traditional communication channels.

Edubot not only enhances student satisfaction by improving response time and reducing frustration, but it also empowers the administrative departments by automating repetitive tasks.

This frees up valuable human resources for more complex, decision-based support and policy work. Moreover, its scalable architecture ensures that the chatbot can handle peak query volumes during critical periods such as admissions, exam season, or results publication without any performance degradation.

By integrating cutting-edge technologies such as natural language processing (nlp), machine learning (ml), and the gemini pro ai api, edubot is not just a chatbot—it's a conversational intelligence platform that evolves with user needs and institutional requirements. It signifies a major step toward digitally transforming university operations and creating a more responsive and inclusive campus experience.

II. RELATED WORK: EVOLUTION OF CHATBOTS AND CONVERSATIONAL AI

The journey toward intelligent conversational agents is rooted in decades of advancements in artificial intelligence and computational linguistics. The idea of simulating human-like conversations using machines is not new, but its execution has evolved significantly across generations of technology.

A. Rule-based systems

Early chatbot systems, such as eliza (1966) developed at mit and parry (1972), attempted to replicate human conversations using predefined templates and pattern-matching rules. While these systems were groundbreaking for their time, they lacked any understanding of conversational context, leading to repetitive and shallow interactions. Their rigid logic limited their application to highly controlled environments.

B. Expert Systems and Symbolic ai

In the 1970s and 1980s, ai research shifted toward expert systems like mycin, which used if-then rules to emulate decision-making in specialized domains such as medicine. These systems offered more structured interactions but were limited by static knowledge bases, scalability issues, and an inability to adapt to new contexts without manual updates.

C. Statistical models and machine learning (1990s–2010)

The introduction of probabilistic models, such as naive bayes, support vector machines (svms), and decision trees, marked a new era of machine learning-based nlp. These approaches allowed systems to generalize from data, improving chatbot adaptability and response relevance. However, they still struggled with understanding deeper semantic and syntactic relationships within human language.

D. Deep learning and modern nlu

With the rise of deep learning in the 2010s, models like rnns (recurrent neural networks), lstms (long short-term memory), and grus (gated recurrent units) enabled better sequence modeling and contextual understanding. The real breakthrough came with transformer architectures, exemplified by bert, gpt, and roberta, which revolutionized natural language understanding by capturing complex context across long text spans. These models now power state-of-the-art conversational agents that deliver human-like interactions across diverse domains.

E. Chatbots in education

In recent years, ai-powered chatbots have been increasingly adopted by educational institutions. For example, jill watson, an ai teaching assistant at georgia tech, and deakin genie, a student support chatbot at deakin university, demonstrated how conversational ai can assist students with academic questions, deadlines, and personal support 24/7 to help students. These systems have shown measurable success in improving engagement, retention, and student satisfaction.

III. SYSTEM ARCHITECTURE AND METHODOLOGY

The edubot system is built upon a robust, modular architecture designed to ensure scalability, maintainability, and real-time responsiveness. The goal of this architecture is not only to enable seamless interactions between users and the ai engine but also to ensure that the solution can be easily adapted or extended in the future as institutional needs evolve. At its core, edubot follows a client-server architecture model enhanced by a restful api gateway, which acts as the bridge between the user-facing frontend and the intelligent backend services.

The system is divided into several well-defined layers, each responsible for a specific function within the communication pipeline. These layers work in harmony to process user inputs, interpret their intent, fetch or generate meaningful responses, and return them to the user in an interactive and user-friendly format.

A. Frontend layer (client side)

The frontend of edubot serves as the primary interface for users — including students, parents, staff, and potential applicants. It is designed to be lightweight, responsive, and intuitive, ensuring accessibility from both desktop and mobile devices.

- Technologies used: html5, css3, javascript, and optionally react.js for component-based ui development.
- User interface: the chat interface allows users to type queries and view responses in a real-time conversational flow. It includes error handling (e.g., connection issues, invalid input), visual cues, and asynchronous message handling.
- Security and accessibility: frontend validations help prevent malformed requests and ensure that user data is transmitted securely using https encryption.

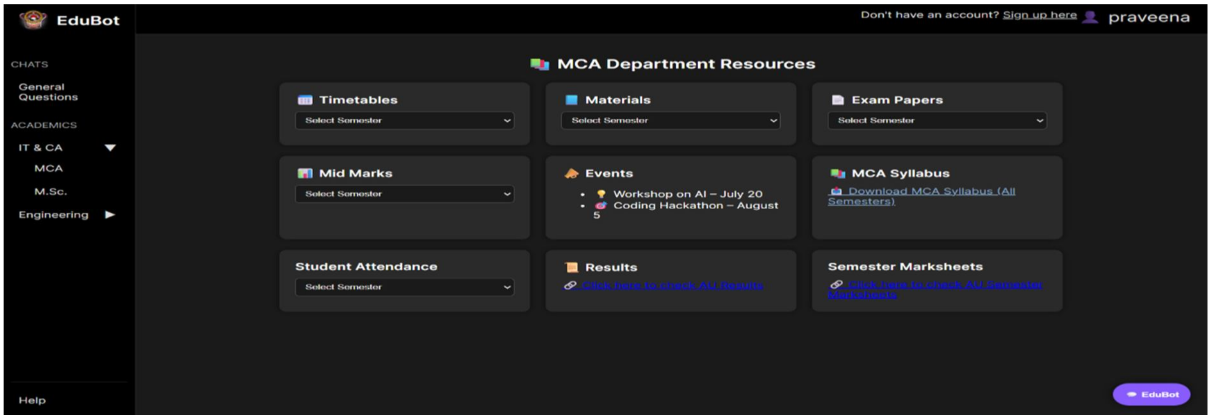


Figure 1: The Frontend User Interface of edubot MCA department

B. Api Gateway

The api gateway acts as a middleware layer that routes incoming http requests from the frontend to the appropriate backend services. It handles restful communication and ensures that the system remains loosely coupled, which means changes in one component do not break the rest of the system.

- Purpose: to manage cross-origin requests, request throttling, authentication (if integrated), and load distribution.
- Advantages: this architectural choice ensures high flexibility and allows edubot to support multiple frontend clients (e.g., web, mobile, or desktop interfaces) without duplicating backend logic.

C. Backend layer (server side)

The backend is the brain of the system, where user queries are interpreted, processed, and answered. It is developed using django 5.x with python 3.12, leveraging the django rest framework for api creation.

Responsibilities:

- Receive and parse the user query from the api gateway.
- Perform input preprocessing, including text normalization, tokenization, lemmatization, and stopword removal.
- Interface with the gemini pro ai api, which serves as the language model engine for generating natural and context-aware responses.
- Interact with the sqlite3 or postgresql database to fetch predefined responses, frequently asked questions (faqs), or historical interaction logs.
- Format and return the final response as json for the frontend to render.

D. Data preprocessing pipeline

Before user queries are sent to the ai model, they are passed through a data preprocessing pipeline that ensures optimal performance and clean inputs. This step exponentially improves the accuracy and relevant features of the ai responses to asked queries.

Steps involved:

- Normalization: convert text to lowercase, remove unnecessary symbols and punctuation.
- Tokenization: break the query into individual words or tokens.
- Lemmatization: convert words to their base form or raw form.
- Stopword removal: eliminate common but contextually insignificant words (e.g., "the", "is", "at").

This pipeline is essential for ensuring that the gemini pro engine interprets the user’s intent accurately, especially in the presence of typos or informal language.

E. Ai inference engine (gemini pro integration)

At the heart of the chatbot lies its ai-powered inference engine, which uses the gemini pro api to generate human-like, context-aware responses. Gemini pro, powered by large language models, allows edubot to go beyond static keyword matching and provide meaningful, adaptive responses even to complex or nuanced queries.

Features:

- Supports multilingual queries.
- Maintaining limited context awareness in smaller sessions.
- Can summarize, explain, and provide elaborative responses.
- Trained on vast amounts of academic and conversational data.

This integration allows edubot to mimic human dialogue and offer a much more natural interaction experience than traditional rule-based bots.

F. Output handler and response delivery

Once a response is generated, it passes through the output handler module. This component ensures that the response is formatted correctly, stripped of sensitive tokens or raw data (if any), and delivered back to the client in a readable format.

- Error handling: in case of network delays, empty responses, or api failures, fallback messages are displayed (e.g., “please try again later.”).
- User feedback: optional feedback buttons (like □/□) can be integrated to continuously improve response quality based on user input.

System flow summary

Here is a simplified flow of how edubot processes a user query:

1) User submits a message via the web chat interface.
2) The message is sent to the backend via the api gateway.
3) Backend preprocesses the text.
4) Query is forwarded to gemini pro api for response generation.
5) The ai-generated response is returned to the backend.
6) Backend processes and formats the response.
7) The responses are sent back to the frontend.
8) User sees the answer in real-time on the chat interface.

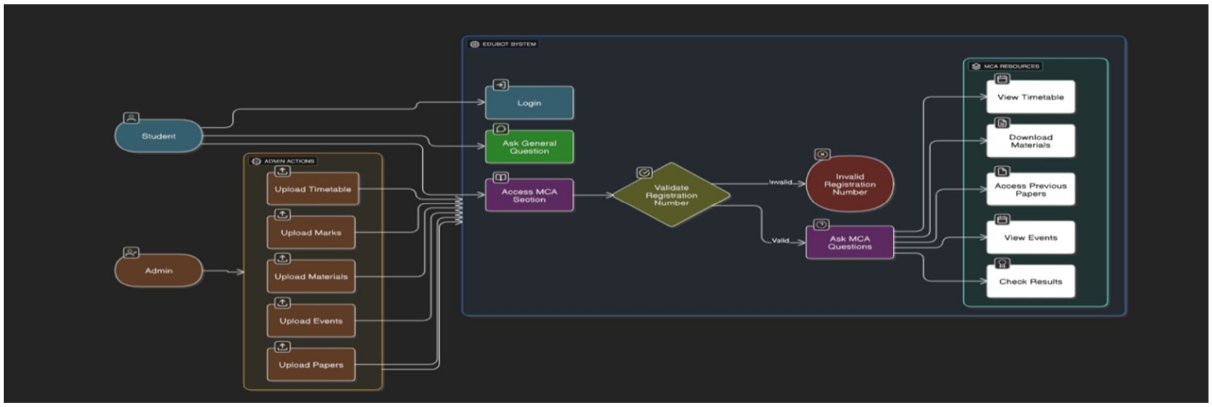


Figure 2: The Control Flow of EduBot

This architecture ensures a scalable, modular, and future-proof foundation for the edubot platform. It supports integration with additional services in the future, such as voice-to-text input, multilingual translation layers, advanced analytics dashboards, or integration with existing academic systems like sis or lms.

IV. IMPLEMENTATION AND TECHNOLOGIES USED

The development of edubot creates a wide range of tools and frameworks to ensure it meets scalability, maintainability, and performance,security.

Technology	Purpose
Python 3.12	Core programming language
Django 5.x	Backend web framework for api and admin ui
Django rest	Used to expose endpoints for the chatbot api
Html/css/javascript	Web interface for user interaction
Fetch api	Handles asynchronous client-server messaging
Google gemini pro api	Ai engine for generating responses
Sqlite3/postgresql	Database for development/production
Requests (python)	Http requests to external ai api

Key files & modules

- Views.py: manages request routing, connects to gemini api, and returns formatted responses.
- Chatbot.html: a lightweight frontend to send/receive messages via fetch and display them in a conversational layout.
- Urls.py, serializers.py, models.py: supporting files to handle routing, data models, and serialization.

All components follow restful design principles, ensuring the chatbot can be embedded within larger university systems or portals with minimal configuration.

V. PERFORMANCE EVALUATION AND SECURITY CONSIDERATIONS

A. *Performance Metrics*

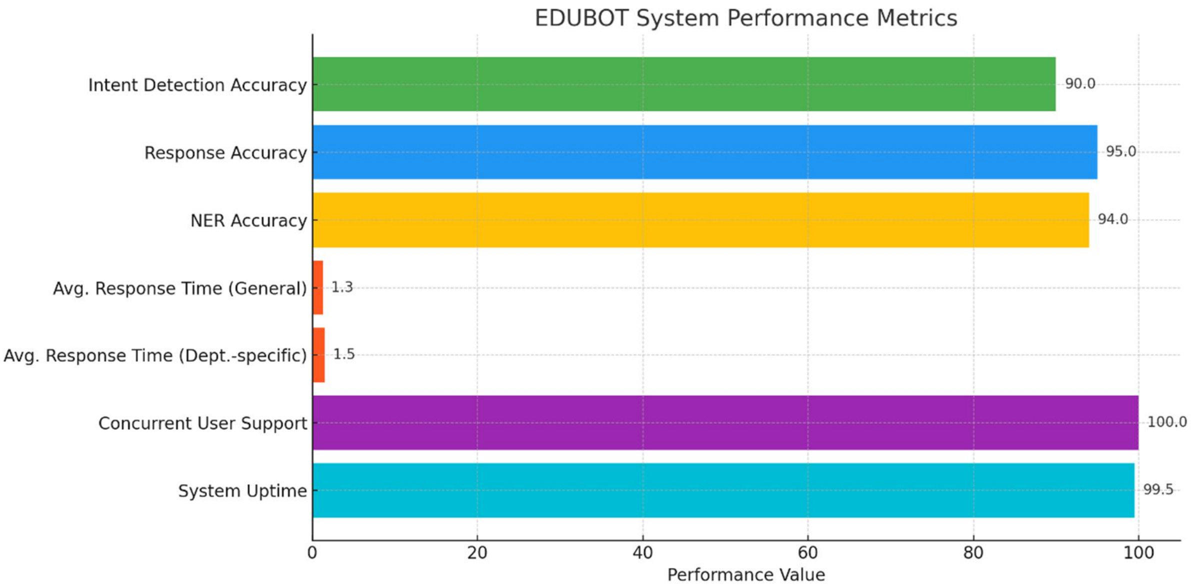


Figure 3: Performance Calculation

To ensure that EDUBOT delivers a consistent, intelligent, and responsive user experience, its performance has been rigorously evaluated across key operational dimensions. The following metrics summarize the system’s efficiency, reliability, and robustness under both controlled testing and real-time simulation:

Intent Detection Accuracy:
Achieves over 90% accuracy in identifying the user's intent across a wide variety of academic, administrative, and casual queries. This demonstrates strong NLP comprehension, even when dealing with vague or imprecise language.

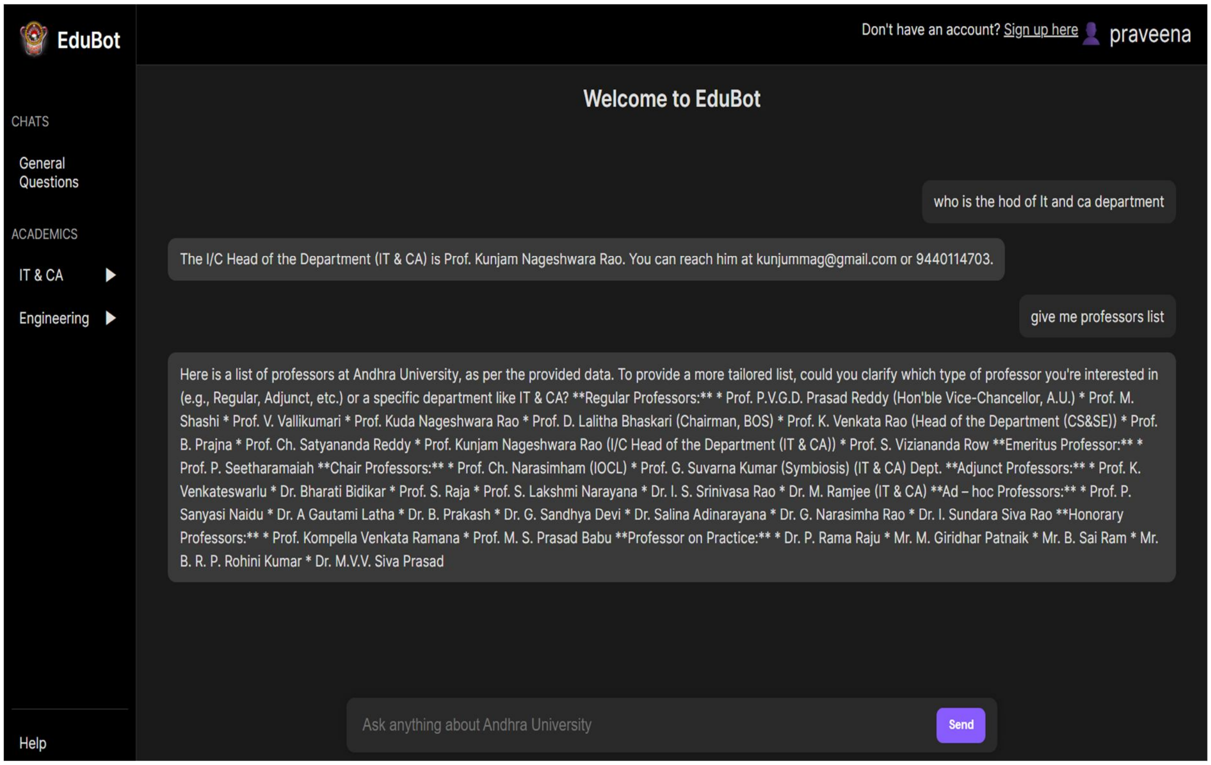
Response Accuracy (Factual and Contextual):
Delivers responses with over 95% accuracy, ensuring that information retrieved from internal databases or generated via the Gemini Pro API is contextually correct, factually reliable, and relevant to the user's question.

Named Entity Recognition (NER) Accuracy:
Records above 94% accuracy in identifying and extracting key entities from user queries — such as course names, department names, dates, and institutional terms — contributing to the precision of generated responses and improved personalization.

- Average Response Time:
  - ~1.3 seconds for general queries (e.g., "What is the exam schedule?")
  - ~1.5 seconds for department-specific or technical queries (e.g., "MCA syllabus for 3rd semester")

Concurrent User Support:
Successfully tested with 100+ simultaneous user sessions without performance degradation. System resources scale dynamically based on server configuration.

- System Uptime and Availability:
  Maintains a >99.5% uptime during continuous deployment and monitoring, ensuring EDUBOT remains accessible round-the-clock for students and staff.



who is the hod of It and ca department

The I/C Head of the Department (IT & CA) is Prof. Kunjam Nageshwara Rao. You can reach him at kunjummag@gmail.com or 9440114703.

give me professors list

Here is a list of professors at Andhra University, as per the provided data. To provide a more tailored list, could you clarify which type of professor you're interested in (e.g., Regular, Adjunct, etc.) or a specific department like IT & CA? **Regular Professors:** * Prof. P.V.G.D. Prasad Reddy (Hon'ble Vice-Chancellor, A.U.) * Prof. M. Shashi * Prof. V. Vallikumari * Prof. Kuda Nageshwara Rao * Prof. D. Lalitha Bhaskari (Chairman, BOS) * Prof. K. Venkata Rao (Head of the Department (CS&SE)) * Prof. B. Prajna * Prof. Ch. Satyananda Reddy * Prof. Kunjam Nageshwara Rao (I/C Head of the Department (IT & CA)) * Prof. S. Viziananda Row **Emeritus Professor:** * Prof. P. Seetharamaiah **Chair Professors:** * Prof. Ch. Narasimham (IOCL) * Prof. G. Suvarna Kumar (Symbiosis) (IT & CA) Dept. **Adjunct Professors:** * Prof. K. Venkateswarlu * Dr. Bharati Bidikar * Prof. S. Raja * Prof. S. Lakshmi Narayana * Dr. I. S. Srinivasa Rao * Dr. M. Ramjee (IT & CA) **Ad – hoc Professors:** * Prof. P. Sanyasi Naidu * Dr. A Gautami Latha * Dr. B. Prakash * Dr. G. Sandhya Devi * Dr. Salina Adinarayana * Dr. G. Narasimha Rao * Dr. I. Sundara Siva Rao **Honorary Professors:** * Prof. Kompella Venkata Ramana * Prof. M. S. Prasad Babu **Professor on Practice:** * Dr. P. Rama Raju * Mr. M. Giridhar Patnaik * Mr. B. Sai Ram * Mr. B. R. P. Rohini Kumar * Dr. M.V.V. Siva Prasad

Figure 4: The Response of Edubot When Asked a Query

B. *Security Framework*

Edubot is designed with a strong goal on security and privacy:

- Encrypted communication: all data transmitted via tls/https
- Input validation: prevents xss, sql injection, and malformed queries
- Role-based access control: admin panel access is restricted by user roles
- Minimal data collection: adheres to data minimization principles to protect personally identifiable information (pii)
- Centralized security monitoring: all backend components are configured with logging and anomaly detection features

VI. CONCLUSION AND FUTURE ENHANCEMENTS

A. Conclusion

Edubot presents a significant leap forward in digital student services for andhra university. By combining natural language understanding with a responsive web-based platform, it successfully automates routine academic and administrative interactions. The system not only enhances the university's operational efficiency but also provides students with an accessible, always-available, and user-friendly support mechanism.

B. Future Development Roadmap

1) Advanced features:
- Context tracking for multi-turn conversations
- Personalized suggestions and proactive assistance
- Speech-to-text integration for voice input
- Multilingual support to serve non-english speakers

2) Data management enhancements:
- Integration with rpa bots and web crawlers for real-time data collection
- Dynamic data sync with academic repositories and timetables

3) Ui/ux improvements:
- Progressive web app (pwa) version for mobile access
- Modern ui themes, accessibility tools, and chatbot avatars

4) Infrastructure and deployment:
- Migration to postgresql and scalable cloud infrastructure using docker and kubernetes
- Centralized logging, performance dashboards, and automated testing

5) Third-party integration:
- Direct api integration with student information systems (sis), learning management systems (lms), official university websites, and helpdesk platforms

REFERENCES

[1] Adelaide University. (n.d.). Ask Adelaide. Retrieved from official university website.

[2] Bali, M., & Moussa, R. (2023). The potential of chatbots in higher education: A review of the literature. Education and Information Technologies.

[3] Bohus, D., & Horvitz, E. (2009). A-posteriori agreement for handling uncertainty in a spoken dialog system. Proceedings of the 10th Annual Conference of the International Speech Communication Association.

[4] Brown, T. B., et al. (2020). Language Models are Few-Shot Learners. Advances in Neural Information Processing Systems, 33.

[5] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.

[6] Følstad, A., et al. (2021). Chatbots for public services: A design framework. Government Information Quarterly.

[7] Fryer, L. K., et al. (2019). The potential of chatbots to support student motivation and engagement. Computers & Education.

[8] Hew, K. F., et al. (2018). The use of chatbots for learning: A review of the literature. Educational Technology Research and Development.

[9] Jurafsky, D., & Martin, J. H. (2023). Speech and Language Processing (3rd ed.). Prentice Hall.