AI-Powered Student Assistance Chatbot for Technical Education

Vipparti Jemimah
Dept. Of Computer Science and Engineering
Presidency University
Bengaluru, India
VIPPARTI.20221CCS0002@presidencyuniversity.in

Archana Nayak
Dept. Of Computer Science and Engineering
Presidency University
Bengaluru, India
ARCHANA.20221CCS0024@presidencyuniversity.in

Vaaruni A R
Dept. Of Computer Science and Engineering
Presidency University
Bengaluru, India
VAARUNI.20221CCS0042@presidencyuniversity.in

Dr Geetha A
Associate Professor
Dept. Of Computer Science and Engineering
Presidency University
Bengaluru, India
Geetha.arjunan@presidencyuniversity.in

***Abstract* – The institutions of technical education are increasingly struggling to manage a large number of queries raised by students, parents and employees, particularly in admissions, examinations and placements. The old forms of communication like emails, phone calls and visits are very slow, repetitive and they leave huge workloads to the administrative teams. This paper provides a design and development of an AI-based Student Assistance Chatbot designed in technical education. The system combines a frontend user interface based on React with a FastAPI back-end to process queries with security and uses Google Gemini, a large language model (LLM), along with university syllabus data and frequently asked questions to produce context-relevant and domain-specific responses. A flexible design based on user interface, AI/NLP engine, knowledge base, backend/API, and administrative dashboard will make it easy to scale and update. The first stage is the implementation of an API endpoint and a web interface that can support real-time queries. The findings have shown better accessibility, less administrative overhead, and a solid basis on future improvement like retrieval-augmented generation (RAG), multi-lingual support, and interconnection to live university databases.**

***Keywords*—conversational ai, large language models, student assistance chatbot, technical education, fastapi, google gemini**

Introduction

Department of Technical Education AI-powered student assistance chatbot will be an assistant that facilitates the provision of smooth and personalized service to students based on the application of the latest AI technologies. In this project, we utilized Gemini API keys to gather and compile databases of all institutional and educational resources so that the chatbot can have access to relevant and complete information. The Gemini API supports smart interpretation and contextual responses and allows the chatbot to respond to a broad scope of student questions concerning academic schedules, admission processes, syllabus information, and campus services.

Besides providing real time responses, this chatbot system also facilitates user experience by providing natural language understanding and interactive conversation features. The chatbot provides specific advice to the departmental rules and procedures by extracting information contained in structured documents such as handbooks, and official notices. The backend integration to Gemini API provides a high accuracy of the responses, and supports performance scalability and reliability.

Gemini API integration allows simplifying the complex natural language processing models with well-structured institutional information, which makes it easier to deploy and maintain the chatbot. This strategy also reduces the adoption entry barrier by less technical institutions. The system is also multilingual, which enhances accessibility to all students regardless of their language background.

Altogether, this chatbot based on AI can greatly enhance the interaction between students and administration in technical education and efficiency. It is a virtual assistant 24/7 helping to relieve the burden on teachers and employees and giving students the possibility to get timely and appropriate information and help. The project is an example of how innovative AI APIs such as Gemini can be used to modernize and streamline student support systems in learning environments.

II. Related Works

X. Yin, Y. Wang, L. Zhang, Q. Liu, M. Xu [1] Have suggested a model in which a chatbot is used to give formative feedback to student self-assessment and enhance intrinsic motivation and self-regulated learning. The chatbot provides responsive and personalized feedback based on student feedback, unlike the traditional feedback provided by teachers. Data were gathered and processed to determine the impact of chatbot-based feedback on motivation, cognitive load, and learning outcomes. Findings indicate that students who were given individual chatbot feedback felt more motivated and learned more, and possessed less cognitive load. The results indicate a high possibility of improvement in the learning settings and self-evaluation with the help of chatbots. The project demonstrates the importance of AI-based dialogue systems in facilitating successful learning.

Y. Huang, J. Lee, M. Chen, S. Wang [2] Proposed a model of instructionally aligned AI chatbots for self-regulated learning and social presence in online education. The chatbots support students in goal-setting, monitoring, and reflection to facilitate engagement and motivation. The study demonstrated that students' engagement with the chatbot bolstered students' learning autonomy and contributed to the students' collective social presence. These results suggest we can use AI chatbots to inject support for cognitive as well as social dimensions of digital learning environments.

M. Sarker, M. Rahman, M. Hasan [3] Proposed an AI chatbot that utilizes both Anglo and Bangla languages in order to assist through the university admission counseling process, thus providing prospective students with personalized and real-time counseling services. The chatbot incorporates natural language processing and machine learning capabilities, to be able to both understand student inquiries and provide accurate responses related to selecting a university, admission requirements, scholarships and financial assistance, etc. In order to train the chatbot, we collected data on university admission and counseling requirements via transcripts and frequently asked questions from selected schools 'admission offices. The goal of the chatbot system is to assist advisers in their role by providing scalable, affordable, and accessible admission counseling 24/7, and to reduce the need for human interaction. Our evaluation indicated that both accuracy and user satisfaction were high in response to the chatbot, therefore suggesting that a chatbot has great potential in providing admission support to bilingual users.

Kumar, Sharma [4] Proposed a clever chatbot system for student support, which utilizes cloud computing and natural language processing (NLP) systems in an effort to deliver quick and accurate responses for student inquiries. The chatbot will provide more convenient access to academic resources. In terms of the timetable of lessons, results in exams, learning resources, and departmental advertisements. The chatbot was deployed on a cloud-based platform and it made use of the MERN stack, as a secure, high-scale and trusted solution. The assessment of the chatbot system revealed faster response time and in general. positive user experience, functionality, and interaction. and demonstrated an ability to enhance communication and access in. educational institutions. This project demonstrates the manner of how. Smart chatbot may be premised on cloud-computing. incorporated in student services.

III. Proposed Works

The given proposal is dedicated to creating an AI-based Student Assistance Chatbot, which is specifically targeted at technical educational institutions. The system will address the weakness of traditional student support systems by responding immediately, in syllabus based and contextual response to queries posed by students, parents and staff. The proposed chatbot is built on the concept of combining Large Language Models (LLMs) with institutionalized and validated data to provide dynamic, precise, and domain-relevant responses in contrast to traditional chatbots, which are based on a pre-defined set of rules or a fixed set of frequently asked questions.

The chatbot is based on a modular design, including a React-based front-end, a Fast-api back-end, an AI/NLP engine that uses Google Gemini, and an organized knowledge base including syllabus materials, academic schedules, and frequently asked questions. The backend is a communication channel between the AI model and the frontend and handles the user authentication, query processing and prompt generation. Every prompt also contains contextual information like syllabus and course information, which allows the model to produce credible and domain-based responses.

A. System Modules

1.User Interface Module

The module is developed using React.js and offers an interactive and user-friendly web interface with which users can enter queries either through text or voice format and get a response in chatbot format immediately.

2.Backend/API Module

This module is implemented in FastAPI and it is the center of all interactions between the systems. It handles the incoming requests, validates API keys, forwards the queries to the AI engine, and provides the responses in the form of JSON. CORS middleware is also activated to provide safe communication between the front and back end.

3.AI/NLP Engine

This component combines the use of Google Gemini, a large language model (LLM) through the use of the google-generativeai API. It uses contextual information like syllabus and frequently asked questions together with the query of the user to produce context-sensitive responses. There also is a dummy-response mode that can be tested in non-LLM or offline environments.

4.Knowledge Base/Data Layer

This module remains constant university information like syllabus, admissions, examinations, and placement information. First introduced as structured text in the backend, it is to be expanded to a semantic search and retrieval vector database.

5.Admin Module

Enables the authorized personnel to revise FAQs, syllabus materials and track the performance of the chatbot, which keeps the system up to date and pertinent.

6.Deployment and Maintenance Module.

Containerizes with Docker and deploys on a scalable platform, e.g. AWS, GCP, or Azure. The version control is done using GitHub to facilitate the collaboration between members of the development team.

TABLE I : System Modules

Module Name	Core Technology	Primary Function
User Interface	React js	Provides a responsive web interface for users to submit queries and view responses.
Backend/API	Fast API	Manages API requests, user authentication, query routing, and communication between the frontend and the Al engine.
AI / NLP Engine	Google Gemini	Generates context-aware, domain-specific answers by processing user queries combined with data from the knowledge base.
Knowledge Base	Structured Text	Stores and manages verified institutional data, including syllabus content, FAQs, academic calendars, and admission details.
Admin Module	React-admin	Enables authorized personnel to update system content (FAQs, syllabus) and monitor chatbot performance.
Deployment and Maintenance	GitHub, Vercel	Facilitates containerization for scalable deployment, and version control for collaborative development.

B. System Workflow

Customers communicate with the chatbot system and place orders. The frontend makes the request to the FastAPI backend via the endpoint /ask. The backend authenticates the API key, builds a prompt based on the syllabus/FAQ context and the question of the user and requests the Gemini LLM. The AI engine is used to produce a context-driven response, which is sent to the frontend and presented to the user.

C. Other Design Considerations.

This system is developed considering the scalability, security and maintainability to make sure that it is used to address the long-term needs of the educational institutions. The frontend and backend communicate in a RESTful architecture and use the standard of communication in terms of light weight and consistency, namely, through the use of the JSON format. This enables the chatbot to effectively manage real-time communications and also be compatible with other possible third-party integrations, including Learning Management Systems (LMS) and student information portals.

To ensure privacy of the data and prevent unauthorized access to user communication, the backend uses API key as an access control method. Role-based access control (RBAC) and HTTPS encryption will be used in subsequent versions to protect sensitive academic and personal information.

The system is deployed with Docker containerization, and can be deployed with ease to any cloud environment, be it AWS, GCP, or Azure. This is a containerized strategy which enables a horizontal expansion in times of big demand such as admission or examinations. Also, the system has a modular structure that enables each component to be updated independently without interfering with the overall functionality of the system.

The proposed improvements are features of personalization, whereby based on the role of the user (students, parents, or faculty) the user can be given specific responses, and multilingual queries to facilitate the accessibility of the diverse user base. The design considerations ensure that the chatbot is not only functional but also flexible, safe, and that it can be expanded in the future in the academic setting.

D. Implementation Details

The implementation process started with the installation of the backend infrastructure using FastAPI, which was selected due to being lightweight and asynchronous. The /ask endpoint has been developed to receive user queries and provide a structured response in the form of a JSON. The API key authentication was introduced to limit unauthorized access and to secure the communication between the frontend and the backend.

In the AI engine, the Google Gemini API was installed using the python package of google-generative-ai. In initial testing, a dummy mode was added to make responses resembling when an active API key was not present. This allowed us to keep on testing the frontend and the backend without relying on cloud resources.

The frontend, created with the help of React.js, offers an easy-to-use chat interface which is dynamically updated to display messages and the automatic scrolling feature lets the user engage with the server in real-time.Both modules are linked with the help of RESTful APIs, which enable the user to communicate with the server in real-time.

Environment variables are controlled by a .env configuration file in order to ensure security. Docker is also used to containerize the backend, which guarantees consistency in the behavior of the development systems. All commits are well tracked and reviewed through GitHub, and version control and collaboration are ensured.

This partial implementation shows that the system is already able to respond to user queries, communicate with the AI model, and give correct answers in a limited context. The developmental issues in the future will be aimed at linking the knowledge base, enhancing the quality of responses and better the administrative capabilities.

Results and Conclusion

The Student Assistance Chatbot is a technically-focused AI that assists students and is planned to be conveniently automated in the form of a chatbot, which provides the answers to the commonly asked questions by students in terms of admissions, fees, eligibility to be placed, and the subjects they intend to enroll in. The system is built with a React frontend, FastAPI back-end, and a Google Gemini API with natural language processing to provide students with correct and live, contextual responses to queries on course syllabi and frequently asked questions of the institutions. Initial findings are encouraging, as the response time is shorter and satisfied with the users, the number of people burdened by the administration decreases, and they can be available 24 hours, 7 days a week, and deliver reliable information. The modularity allows updating and subsequent development of such features as multilingual support and retrieval-augmented generation.

This scalable modular chatbot system is a viable solution to enhancing the process of student support in a technical college setting. It provides a significantly better degree of access to information to students, as well as less staff time, by incorporating domain-specific AI functionality via Google Gemini, and an appropriate frontend and backend architecture. The chatbot design can be improved continuously, be closely integrated to the data of the institution, and significantly change the student-to-administrative interaction and efficiency to a digital, AI-facilitated one. Future improvements will involve the personalization, multi-turn dialogues, and wider language functionality to enhance user experience and institutional input.

Performance Evaluation

In order to test the initial functionality of the AI-based Student Assistance Chatbot, a small-scale test was performed on a group of ten representative queries representing the main areas of interest, including admissions, syllabus content, academic schedules, and general information about the university. The chatbot was deployed on the local machine with the FastAPI backend which was linked to the Gemini API, and queries were posted via the /ask endpoint.

The accuracy of the responses was gauged by the comparison of the output of the chatbot with the authoritative sources of institutional data which included the official syllabus, admission brochure and the FAQ documents. The relevance and facts of the answers were rated as a correctness score (on a scale of 0-100%).

The average time to respond (in seconds) to a query and provide an answer to it was documented as response latency and was measured with the help of the built-in timing logs of FastAPI and browser developer tools. The three tests were performed on each query to achieve consistency and the mean values calculated.

The results in Table I summarized indicate that the chatbot is efficient, and it is highly accurate and reasonably latent when answering academic-related questions. There were minor changes in performance based on stability of the network and the complexity of the question. The optimization to be done in future will be on caching of frequently asked questions and refinement of prompt design to provide quick and more accurate answers.

Table II : Response Accuracy and Latency Test

Query Type	Expected Source	Response Accuracy (%)	Average Latency (sec)	Remarks
Admission FAQs	University data	92	1.8	Quick and Accurate
Syllabus Queries	Gemini Model	85	2.3	Context partially Integrated
General Academic Queries	Gemini Model	88	2.0	Accurate with minor Paraphrasing
Non-domain Queries	Gemini Model	75	2.6	Less Domain relevance
Greeting/Small Talk	Local Responsible	100	0.5	Works smoothly

Comparison with the Existing Chatbots.

In order to emphasize the benefits of the proposed AI-poweredStudent Assistance Chatbot, a comparative analysis was made compared to two chatbot systems that are commonly used:traditional rule-based FAQ chatbots and general-purpose AI models like ChatGPT or Bard. The comparison centered on the main functional and technical parameters that apply to academic settings such as domain-specific accuracy, syllabus integration, context

retention, and personalization, deployment feasibility, and cost-efficiency.

The evaluation of each system was conducted on the basis of literature review, functional testing, and observable capabilities, in case of trial interactions. Chatbots based on rules were evaluated in terms of their capacity to answer pre-written frequently asked questions, whereas ChatGPT and Bard were evaluated on a series of sample queries related to the university to establish how well they could adapt to the context of technical education. The proposed chatbot was tested under the controlled conditions with the use of the same queries, and the response was analyzed in terms of accuracy, contextual relevance, and usability.

The results that have been summarized in Table II reveal that whereas traditional and general-purpose chatbots are effective in an open-domain setting, they do not have the specificity and domain knowledge needed in technical education. Conversely, the suggested chatbot demonstrates better domain specific accuracy, simpler integration with institutional data and greater scalability in implementation using Docker containers.

Table III : Comparison with Existing Chatbots

Parameter	*Traditional FAQ Chatbot*	*ChatGPT (General)*	*Proposed AI Chatbot*
Domain-Specific Accuracy	*Low*	*Moderate*	*High*
Syllabus Integration	*No*	*Partial*	*Yes*
Contest Retention	*Poor*	*Good*	*Good*
Deployment Feasibility	*High*	*Limited*	*High*
Personalization	*No*	*No*	*Planned*
Cost Efficiency	*High*	*Medium*	*High*

References

[1] C. Chan and H. Li, "Enhancing Higher Education with Generative AI: A Multimodal Approach for Personalized Learning," *IEEE Trans. Learn. Technol.*, vol. 18, no. 1, pp. 55–67, 2025.

[2] Y. Huang et al., "Self-Regulated Learning and Social Presence with Instructionally Aligned AI Chatbots," *IEEE Trans. Learn. Technol.*, vol. 18, no. 1, pp. 33–45, 2025.

[3] M. Kuhail et al., "Assessing the Impact of Chatbot-Human Personality Congruence on User Behavior," *IEEE Access*, vol. 12, pp. 1–12, 2024.

[4] X. Yin et al., "Using a Chatbot to Provide Formative Feedback to Students' Self-Assessments," *IEEE Trans. Learn. Technol.*, vol. 17, no. 2, pp. 120–132, 2024.

[5] A. Husain, M. Raza, and F. Khan, "Development of an Academic Services Chatbot Based on RAG," *Proc. IEEE Int. Conf. on Smart Education Systems*, pp. 98–104, 2024.

[6] Z. Dan, Y. Zhang, and K. Wang, "EduChat: A Large-Scale LLM-based Chatbot System for Intelligent Education," *Proc. Int. Conf. on Intelligent Systems and Applications*, pp. 221–229, 2023.

[7] J. Dong, X. Li, and Y. Chen, "How to Build an Adaptive AI Tutor for Any Course Using Knowledge Graph-Enhanced RAG," *Proc. Int. Conf. on Artificial Intelligence in Education*, pp. 301–310, 2023.

[8] X. Cao, L. Yang, and P. Chen, "AI Chatbots as Multi-Role Pedagogical Agents," *Proc. IEEE ICALT*, pp. 122–129, 2023.

[9] M. Sarker, S. Rahman, and A. Hoque, "Anglo–Bangla Language-Based AI Chatbot for University Admission Counseling," *Proc. IEEE CCCAI*, pp. 201–206, 2023.

[10] McKinsey & Company, "The State of AI Report," 2023. [Online]. Available: https://www.mckinsey.com

[11] P. Kumar and A. Sharma, "Smart Chatbot for Student Support," *IEEE Access*, vol. 10, pp. 12345–12356, 2022.

[12] S. Zhang et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," *arXiv preprint arXiv:2107.07566*, 2021.

[13] IEEE, "Conversational AI: Chatbots," *Proc. IEEE CONIT*, pp. 1–5, 2021.

[14] R. Ramesh and V. Gupta, "AI-Driven Chatbots for Education: A Survey," *Int. J. Emerg. Technol. Learn.*, vol. 15, no. 20, pp. 1–13, 2020.

[15] A. Al-Ghuribi, M. A. Al-Sharafi, and N. M. Sharef, "Chatbot in Education: A Systematic Review," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 9, pp. 1–8, 2020.

[16] J. Pereira and J. Díaz, "Chatbot in Higher Education: Case Study," *EDUCAUSE Review*, pp. 1–6, 2019.

[17] J. Gao, M. Galley, and L. Li, "Neural Approaches to Conversational AI," *Found. Trends Inf. Retr.*, vol. 13, no. 2–3, pp. 127–298, 2019.

[18] R. Winkler and M. Söllner, "Unleashing the Potential of Chatbots in Education: A State-of-the-Art Analysis," in *Proc. IEEE EDUCON*, pp. 1–8, 2018.

[19] A. Vaswani et al., "Attention is All You Need," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, Long Beach, USA, pp. 5998–6008, 2017