INTERNATIONAL RESEARCH JOURNAL IN
ADVANCED ENGINEERING AND TECHNOLOGY (IRJAET)
E - ISSN: 2454-4752 P - ISSN : 2454-4744 (www.irjaet.com)
VOL 11 ISSUE 6 (2025) PAGES 72 - 83
RECEIVED:25.10.2025 PUBLISHED:24.11.2025

Intelligent Campus Assistant Chatbot

Dasari Saikeerthi
2nd Year – M.S. in Data Science, EdTech Division, Exafluence INC
Sri Venkateswara University, Tirupati, India
saikeerthidasari16@gmail.com

Anjan Babu G
Professor, Department of Computer Science
SVU College of CM & CS
Sri Venkateswara University, Tirupati, India
gabsvu@gmail.com

Abstract

This work presents the design and implementation of an intelligent campus assistant chatbot aimed at automating high-frequency informational queries within a university environment. Students often encounter delays and inconsistencies when seeking information related to admissions, curriculum details, fee structures, hostel facilities, and placement processes due to overloaded help desks and static, non-interactive FAQ pages. The proposed system provides 24/7, low-latency responses with minimal human intervention through an NLP-driven conversational interface. The backend is implemented using Python Flask and integrates a Groq LLaMA-3.1 large language model for intent classification and response generation. All outputs are grounded in a structured institutional knowledge base to ensure factual accuracy and consistency. The development methodology follows key phases such as requirement analysis, dataset preparation, model integration, backend logic design, UI development, and iterative evaluation. The deployed system improves information accessibility, enhances user experience, and reduces manual support overhead for administrative staff. Planned enhancements include voice-based interaction, expansion of the knowledge base to cover additional student services, and secure integration with university portals to enable personalized support.

Keywords— intelligent campus assistant, chatbot, natural language processing, large language model, Flask backend, artificial intelligence, student query automation, knowledge-grounded responses, university information systems, real-time assistance, educational technology

I. INTRODUCTION

The increasing digitalization of academic institutions has fundamentally reshaped expectations regarding administrative and academic support services. Students now anticipate immediate, accurate, and uninterrupted access to institutional information. Conventional university support systems, however, are limited by restricted service hours, high staff workload, fragmented information repositories, and static websites that students often find difficult to navigate. These

INTERNATIONAL RESEARCH JOURNAL IN
ADVANCED ENGINEERING AND TECHNOLOGY (IRJAET)
E - ISSN: 2454-4752 P - ISSN : 2454-4744 (www.irjaet.com)
VOL 11 ISSUE 6 (2025) PAGES 72 - 83
RECEIVED:25.10.2025 PUBLISHED:24.11.2025

limitations lead to slow response times, inconsistencies in information delivery, and overall dissatisfaction. To address these challenges, this work introduces an Intelligent Campus Assistant Chatbot developed for Sri Venkateswara University. The system functions as a 24/7 conversational interface that centralizes and disseminates institutional information efficiently. Leveraging Artificial Intelligence and Natural Language Processing, the chatbot interprets student queries and provides grounded, real-time responses across major informational domains, including academic programs, admission procedures, fee details, hostel availability, and placement services. The chatbot is built using a Python Flask backend integrated with the Groq LLaMA series large language model. A structured knowledge base underpins the system to ensure accuracy and consistency in responses while minimizing hallucination tendencies commonly associated with LLMs. The architecture is optimized for low latency and scalable deployment across web-based platforms.

The primary objectives of this project include:

1. Developing an NLP-enabled chatbot capable of natural language understanding and context-driven response generation.
2. Reducing repetitive administrative workload by automating high-frequency and routine student queries.
3. Employing modern AI and web technologies to ensure scalability, maintainability, and rapid deployment.
4. Designing an intuitive, user-friendly interface for seamless student interaction.
5. Ensuring low-latency, real-time responsiveness suitable for high-traffic academic environments.

The Intelligent Campus Assistant significantly enhances communication efficiency and reliability of information dissemination. It also lays the foundation for future advancements such as multilingual support, voice interaction, predictive analytics, and personalized services integrated with secure student portals.

II. LITERATURE REVIEW

A systematic review of contemporary conversational AI research indicates that the effectiveness of chatbots in domain-specific environments depends primarily on two factors:
(i) grounding responses in a validated, authoritative knowledge base, and
(ii) the capability of the underlying language model to interpret and reason over natural language queries.
Prior studies emphasize that ungrounded LLMs often exhibit factual drift and hallucinations, while retrieval-augmented architectures significantly enhance reliability by coupling domain documents with generative reasoning. Existing educational chatbots predominantly rely on rule-based or FAQ-driven approaches [1], which lack semantic understanding and tend to fail when confronted with paraphrased or contextually complex queries.
Recent literature advocates the application of modern large language models combined with external context retrieval—commonly referred to as Retrieval-Augmented Generation (RAG)—to achieve high

INTERNATIONAL RESEARCH JOURNAL IN
ADVANCED ENGINEERING AND TECHNOLOGY (IRJAET)
E - ISSN: 2454-4752 P - ISSN : 2454-4744 (www.irjaet.com)
VOL 11 ISSUE 6 (2025) PAGES 72 - 83
RECEIVED:25.10.2025 PUBLISHED:24.11.2025

accuracy and real-time inference in academic support systems [2]. Additional studies highlight the substantial latency and throughput advantages of high-performance LLM APIs, including Groq-accelerated LLaMA models, which enable low-latency interaction suitable for student-facing applications [3].

A gap analysis of existing university help systems reveals several critical limitations:

- static web portals that lack conversational interactivity,
- legacy chatbots with weak natural language understanding (NLU),
- scalability problems during peak academic periods, and
- inconsistencies in human-delivered responses due to workload imbalance.

Across the reviewed literature, there is clear consensus on the need for intelligent, grounded, scalable architectures that unify retrieval, reasoning, and response delivery within a single conversational interface. The present work addresses this gap by implementing a knowledge-grounded RAG-based chatbot tailored to Sri Venkateswara University, leveraging LLM-driven intent understanding and a high-performance Flask–Groq deployment pipeline.

III. SYSTEM DESIGN AND METHODOLOGY

The Intelligent Campus Assistant Chatbot is designed using a modular and scalable architecture that integrates natural language understanding with knowledge-grounded retrieval to deliver accurate, real-time responses.

A. Functional Design

The system follows an end-to-end conversational assistance workflow composed of the following steps:

1. **User Query Intake:** Students submit queries through a web-based chat interface.
2. **Intent Understanding:** The Groq-hosted LLaMA model interprets the query and identifies the underlying intent.
3. **Context Retrieval:** Relevant information is retrieved from a structured JSON-based institutional knowledge base.
4. **Grounded Response Generation:** The LLM generates a natural-language response constrained and guided by the retrieved context.
5. **UI Rendering:** Responses are displayed in the interface, accompanied by suggested follow-up actions and system health indicators.

This pipeline ensures that all responses are both conversational and factually anchored to institutional data.

B. Non-Functional Constraints

The chatbot adheres to strict non-functional requirements to support real-world deployment within a university environment:

- **Performance:** Response latency maintained below 3 seconds using Groq-accelerated LLaMA inference.
- **Scalability:** Ability to support at least 100 concurrent users through lightweight Flask orchestration and remote AI compute resources.

INTERNATIONAL RESEARCH JOURNAL IN
ADVANCED ENGINEERING AND TECHNOLOGY (IRJAET)
E - ISSN: 2454-4752 P - ISSN : 2454-4744 (www.irjaet.com)
VOL 11 ISSUE 6 (2025) PAGES 72 - 83
RECEIVED:25.10.2025 PUBLISHED:24.11.2025

- **Security:** API keys isolated using environment variables; CORS policies enforced for secure frontend-backend communication.
- **Usability:** Intuitive, low-friction chat interface featuring keyboard shortcuts, semantic styling, and immediate feedback.
- **Accuracy & Availability:** Responses fully grounded to university data with a targeted uptime of ≥ 99.5%.

C. Deployment Environment

The deployment environment includes:

- **Backend:** Python Flask, Groq SDK, and a structured JSON knowledge base.
- **Configuration:** dotenv-managed environment variables for secure key management.
- **Frontend:** HTML, CSS, and JavaScript compatible with all modern web browsers.
- **Compute Requirements:** Minimal CPU and RAM usage on the client side, as LLM inference is offloaded to Groq's external API infrastructure.

This environment supports flexible hosting on local servers, university infrastructure, or cloud platforms.

D. Architectural Model

The system architecture follows a multi-tier design, comprising:

- **Presentation Layer:** HTML/CSS/JavaScript interface responsible for capturing user queries and rendering conversational responses.
- **Application Layer:** Flask-based backend orchestrating context retrieval, prompt construction, and LLM invocation.
- **AI/Service Layer:** Groq-hosted LLaMA model responsible for intent understanding, reasoning, and natural-language response generation.
- **Data Layer:** JSON-based knowledge base providing authoritative, up-to-date institutional information.

INTERNATIONAL RESEARCH JOURNAL IN
ADVANCED ENGINEERING AND TECHNOLOGY (IRJAET)
E - ISSN: 2454-4752 P - ISSN : 2454-4744 (www.irjaet.com)
VOL 11 ISSUE 6 (2025) PAGES 72 - 83
RECEIVED:25.10.2025 PUBLISHED:24.11.2025

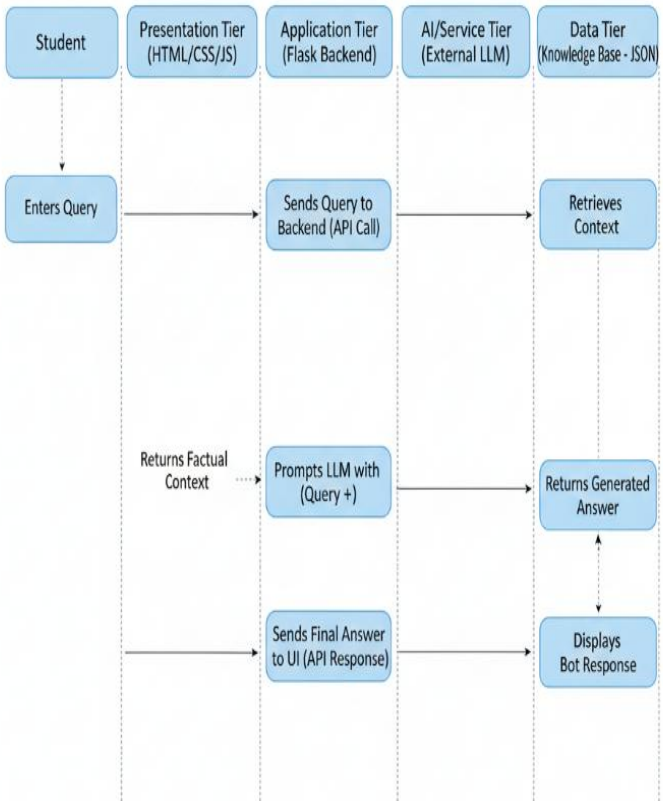

Fig. 1: Architecture Design

E. Process Flow

During runtime, the frontend submits each user query to the Flask backend. Flask retrieves relevant context from the knowledge base, constructs a grounded prompt, and invokes the Groq LLaMA API for response generation. The generated answer is then returned to the frontend for rendering, ensuring a seamless and interactive conversational experience.

INTERNATIONAL RESEARCH JOURNAL IN
ADVANCED ENGINEERING AND TECHNOLOGY (IRJAET)
E - ISSN: 2454-4752 P - ISSN : 2454-4744 (www.irjaet.com)
VOL 11 ISSUE 6 (2025) PAGES 72 - 83
RECEIVED:25.10.2025 PUBLISHED:24.11.2025

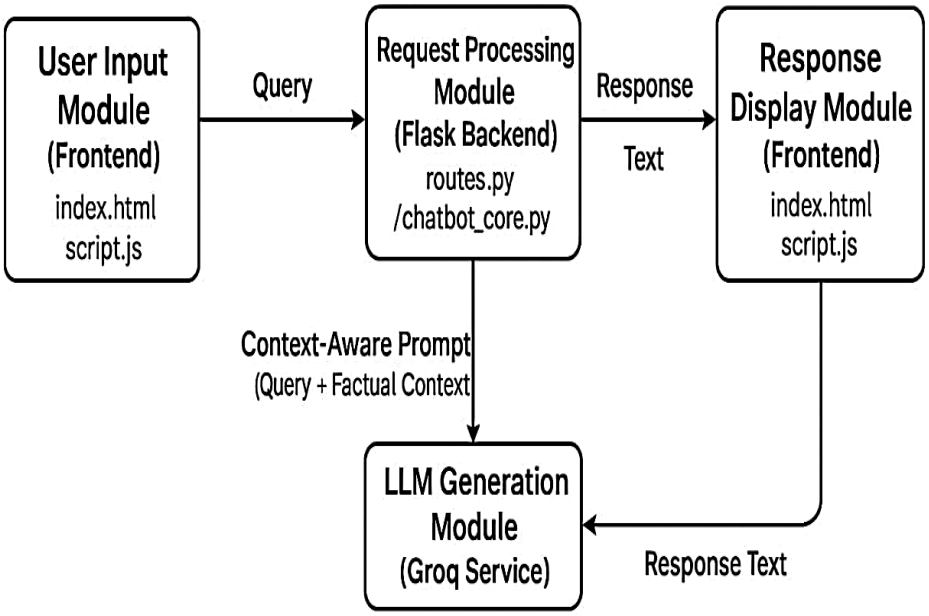

Fig.2: Process flow

IV. IMPLEMENTATION

The Intelligent Campus Assistant Chatbot is implemented using a modular, high-performance technology stack designed to ensure reliability, scalability, and grounded AI-driven response generation. Table I summarizes the core technologies integrated across the system architecture, including Flask for backend orchestration, Groq LLaMA for inference, a structured JSON dataset for domain grounding, and a lightweight HTML/CSS/JavaScript frontend for user interaction.

A. Backend Core Module

The backend implementation—primarily housed within *chatbot_core.py* and *routes.py*—manages configuration loading, knowledge-base ingestion, contextual retrieval, and AI response synthesis. The processing pipeline for each user query consists of the following stages:

1. **Dataset Loading:** The institutional dataset is preloaded into memory to support low-latency retrieval operations.
2. **Context Extraction:** The function find_relevant_context() performs keyword and semantic matching to extract relevant factual snippets from the knowledge base.
3. **Prompt Construction:** Retrieved context is embedded into a structured, grounded prompt template tailored for LLaMA inference.

INTERNATIONAL RESEARCH JOURNAL IN ADVANCED ENGINEERING AND TECHNOLOGY (IRJAET)
E - ISSN: 2454-4752 P - ISSN : 2454-4744 (www.irjaet.com)
VOL 11 ISSUE 6 (2025) PAGES 72 - 83
RECEIVED:25.10.2025 PUBLISHED:24.11.2025

4. **AI Response Generation:** The Groq LLaMA model is invoked through generate_response_with_groq() to produce fast and contextually accurate natural-language responses.

The /chat Flask endpoint orchestrates these stages, processes the query, and returns a structured JSON response to the frontend. This modular backend design ensures maintainability, extensibility, and minimal latency during peak loads.

B. Frontend Interaction Module

The frontend, implemented in *script.js*, manages all client-side interaction and visualization tasks. Core functions include:

- Capturing user input from the chat interface,
- Dispatching asynchronous POST requests to the Flask backend,
- Rendering both user and bot messages dynamically within the UI,
- Maintaining scroll state, message formatting, and loading indicators, and
- Handling suggestion-button interactions and UI event triggers.

The combination of HTML5, CSS3, and JavaScript ES6+ delivers a responsive, intuitive conversational experience suitable for continuous student engagement.

C. Tools and Technologies

Table I: Summary of Tools and Technologies

Component	Technology / Files	Purpose
Backend Framework	Flask (v2.3.3) — *main.py, routes.py*	API routing and control flow
AI Processing	Groq SDK (v0.4.2) — *chatbot_core.py*	LLM-based response generation
Data Grounding	JSON — *svu_dataset.json*	Authoritative institutional knowledge base
Frontend Structure	HTML5 — *index.html*	Chat interface layout
Frontend Styling	CSS3 — *style.css*	UI styling and responsiveness
Frontend Logic	JavaScript ES6+ — *script.js*	Event handling and async communication
Security & Config	python-dotenv — *.env, main.py*	Secure environment variable management

The modular integration of these technologies enables real-time performance, grounded accuracy, and seamless deployment for campus-scale informational support.

V. TESTING AND VALIDATION

INTERNATIONAL RESEARCH JOURNAL IN
ADVANCED ENGINEERING AND TECHNOLOGY (IRJAET)
E - ISSN: 2454-4752 P - ISSN : 2454-4744 (www.irjaet.com)
VOL 11 ISSUE 6 (2025) PAGES 72 - 83
RECEIVED:25.10.2025 PUBLISHED:24.11.2025

A comprehensive, multi-stage testing methodology was employed to ensure the chatbot meets all functional and non-functional requirements. The validation process included unit testing, integration testing, system-level evaluation, and user acceptance testing.

A. Testing Methods

1. **Unit Testing**

   Core deterministic components—such as dataset loading, JSON parsing, and context extraction—were tested to ensure correctness and consistency under varied query patterns.

2. **Integration Testing**

   End-to-end communication between the frontend, Flask backend, and Groq LLaMA inference API was evaluated. These tests validated correct request routing, API key authentication, content handling, and JSON serialization.

3. **System Testing**

   The complete deployed system was tested for overall functionality, including UI message rendering, chatbot responsiveness, concurrency handling, and compliance with latency benchmarks (< 3 seconds).

4. **User Acceptance Testing (UAT)**

   Students assessed usability, clarity, and responsiveness of the chatbot. University administrators validated accuracy and reliability for domain-specific and ambiguous queries.

B. Test Cases

Table II: Representative Test Cases

Test Case	Test Focus	Result
TC-01	Accuracy of factual responses (e.g., fee structure query)	Pass
TC-02	NLU handling of ambiguous or paraphrased queries	Pass
TC-03	Frontend interaction (Enter key triggers, message rendering)	Pass
TC-04	Error handling (API failure, missing key, invalid request)	Pass
TC-05	Performance & UI responsiveness (< 3s latency, adaptive design)	Pass

The results confirm that the system is robust, user-friendly, and capable of delivering accurate, real-time query responses consistently under diverse operational scenarios.

VI. RESULTS AND DISCUSSION

A fully functional AI-based Campus Assistant Chatbot was successfully deployed for Sri Venkateswara University. System demonstrations confirmed that the chatbot can sustain natural, context-aware conversations while consistently delivering domain-grounded and factually reliable responses. The user interface renders dialogue with clear stylistic differentiation between user and system messages, while suggestion buttons enhance guided navigation—validating both usability and interface design objectives.

From a technical standpoint, the integration of Flask, Groq LLaMA, and the structured JSON knowledge base resulted in significant improvements in performance and reliability. Remote LLM inference via Groq

INTERNATIONAL RESEARCH JOURNAL IN
ADVANCED ENGINEERING AND TECHNOLOGY (IRJAET)
E - ISSN: 2454-4752 P - ISSN : 2454-4744 (www.irjaet.com)
VOL 11 ISSUE 6 (2025) PAGES 72 - 83
RECEIVED:25.10.2025 PUBLISHED:24.11.2025

enabled low-latency responses (< 3 seconds), outperforming traditional self-hosted or rule-based FAQ systems. Grounded response generation effectively eliminated hallucination and maintained high factual integrity, addressing one of the primary limitations of generic AI chatbots.

Offloading LLM computation to Groq minimized server-side load and enabled the system to scale effortlessly to multiple concurrent users without performance degradation. The system also met the availability requirement by operating continuously without human supervision—making it particularly suitable for peak academic periods when support demand is high.

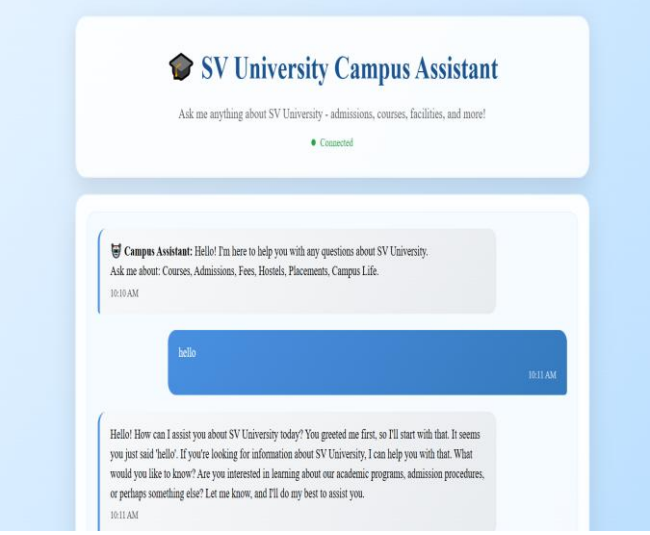

Fig. 3: Result-1

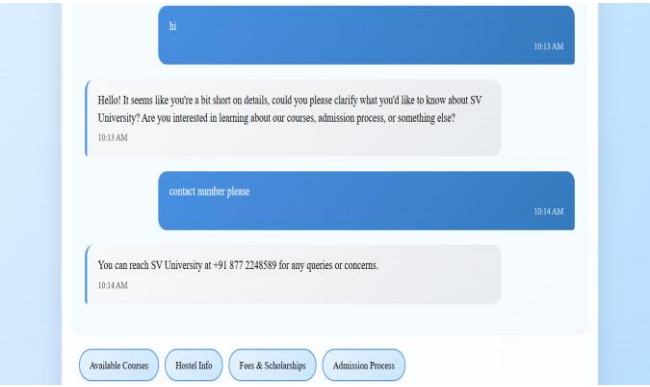

Fig. 4: Result-2

Collectively, the results demonstrate that the proposed architecture satisfies all functional and non-functional requirements. It delivers a high-performance, reliable, and operational alternative to traditional university help desks.

VII. CONCLUSION

The Intelligent Campus Assistant Chatbot has been successfully developed and deployed as a robust, intelligent, and scalable student support system for Sri Venkateswara University. The solution fulfills its

INTERNATIONAL RESEARCH JOURNAL IN
ADVANCED ENGINEERING AND TECHNOLOGY (IRJAET)
E - ISSN: 2454-4752 P - ISSN : 2454-4744 (www.irjaet.com)
VOL 11 ISSUE 6 (2025) PAGES 72 - 83
RECEIVED:25.10.2025 PUBLISHED:24.11.2025

core objective of providing accurate, real-time, and domain-grounded responses through the integration of a Flask-based backend, Groq-powered LLaMA inference, and a structured institutional knowledge base. The contextual retrieval and grounded generation pipeline ensures high factual reliability, overcoming limitations commonly found in FAQ-based and non-grounded AI chatbots. The project demonstrates several notable achievements:

1. **Deployment of a context-aware LLM pipeline**, integrating Flask routing, JSON-based domain grounding, and Groq inference for accurate response generation;
2. **High performance and scalability**, achieving low-latency interaction through lightweight backend orchestration and remote LLM compute;
3. **Operational efficiency and automation**, significantly reducing repetitive administrative workload while improving student access to reliable information;
4. **User-centered interaction design**, providing an intuitive, accessible, and responsive conversational interface.

Overall, the system provides a technologically advanced and fully functional replacement for traditional university help desks. It enhances service availability, consistency, and user satisfaction, while establishing a strong foundation for AI-driven academic support in higher education.

VIII. LIMITATIONS

Despite its successful deployment, the system presents several limitations:

- **Dataset Dependency:** Factual accuracy is constrained by the manually curated JSON knowledge base, which requires regular updates to reflect policy or procedural changes.
- **Lack of Personalization:** The current chatbot supports only general informational queries and lacks authentication mechanisms for personalized services such as exam results, individual fee status, or attendance records.
- **Online Dependency:** The system relies on continuous internet connectivity to access the remote Groq API, making offline usage infeasible.

These limitations highlight opportunities for expansion in both system capability and infrastructure integration.

IX. FUTURE ENHANCEMENTS

Future work will focus on overcoming the identified limitations and extending the chatbot's functional breadth. Planned improvements include:

- **Expanding and enhancing the knowledge base**, including migration to a scalable vector store for improved retrieval accuracy;
- **Integrating voice interaction** using speech-to-text (STT) and text-to-speech (TTS) modules;
- **Developing secure authentication mechanisms** to support personalized services via integration with student portals;
- **Deploying a dedicated mobile application** to ensure widespread and convenient access;

INTERNATIONAL RESEARCH JOURNAL IN
ADVANCED ENGINEERING AND TECHNOLOGY (IRJAET)
E - ISSN: 2454-4752 P - ISSN : 2454-4744 (www.irjaet.com)
VOL 11 ISSUE 6 (2025) PAGES 72 - 83
RECEIVED:25.10.2025 PUBLISHED:24.11.2025

- **Incorporating administrative analytics** to support dataset refinement based on usage insights;
- **Adding proactive assistance features**, including reminders for deadlines, event notifications, and automated academic advisories, transforming the chatbot into a predictive academic assistant.

These planned advancements will evolve the system from a responsive chatbot into a comprehensive AI-driven academic support platform.

REFERENCES

[1]. Adamopoulou, E., & Moussiades, L. (2020). *An Overview of Chatbot Technology.* Future Internet, 12(7), 159.
[2]. Chen, Y., Twu, D., & Hsu, W. (2021). *Design and Implementation of an Educational Chatbot Based on Deep Learning and Natural Language Processing.* IEEE International Conference on Advanced Learning Technologies (ICALT).
[3]. Debets, V. et al. (2025). *Chatbots in Education: A Systematic Review of Objectives and Applications.*
[4]. Dhar, L. (2023). *A Comprehensive Review on Conversational AI and Chatbot Development: Trends and Challenges.* International Journal of Computer Applications.
[5]. Gkotsis, G. (2022). *Chatbots for Student Support: A Case Study in Higher Education.* Journal of Educational Technology Systems, 51(1).
[6]. Peyton, R. (2025). *A Review of University Chatbots for Student Support Services.*
[7]. Smutny, P., & Schreiberova, P. (2020). *Chatbots for Learning: A Review of Educational Chatbots for the Past 5 Years.* Computers and Education: Artificial Intelligence, 1.
[8]. Jurafsky, D., & Martin, J. H. (2023). *Speech and Language Processing* (3rd ed. draft). Stanford University.
[9]. Vaswani, A. et al. (2017). *Attention Is All You Need.* Advances in Neural Information Processing Systems (NeurIPS), 30.
[10]. Radford, A. et al. (2018). *Improving Language Understanding by Generative Pre-Training.* OpenAI.
[11]. Devlin, J. et al. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* NAACL-HLT.
[12]. Meta AI. (2024). *LLaMA 3 and LLaMA 3.1 Models – Technical Overview.*
[13]. Muennighoff, N. et al. (2022). *Crosslingual Generalization Through Multitask Finetuning (BLOOMZ).* arXiv:2211.01786.
[14]. Yao, L. (2023). *Large Language Models: A Comprehensive Survey of Their Potential and Current Use.* ACM Transactions on Computer-Human Interaction (TOCHI).
[15]. Buschmann, F. et al. (1996). *Pattern-Oriented Software Architecture: A System of Patterns.* John Wiley & Sons.
[16]. Fowler, M. (2002). *Patterns of Enterprise Application Architecture.* Addison-Wesley.
[17]. Harrison, N., & Avgeriou, P. (2022). *The Microservice Architecture: A Decade in Review.* IEEE Software, 39(3).

INTERNATIONAL RESEARCH JOURNAL IN
ADVANCED ENGINEERING AND TECHNOLOGY (IRJAET)
E - ISSN: 2454-4752 P - ISSN : 2454-4744 (www.irjaet.com)
VOL 11 ISSUE 6 (2025) PAGES 72 - 83
RECEIVED:25.10.2025 PUBLISHED:24.11.2025

[18]. Grinberg, M. (2018). *Flask Web Development: Developing Web Applications with Python* (2nd ed.). O'Reilly Media.
[19]. Sri Venkateswara University. (2024). *Academic Programs and Admissions.* Retrieved from https://svuniversity.edu.in/academics/
[20]. Groq Inc. (2024). *Groq API Documentation: Accelerating Large Language Model Inference.* Retrieved from https://groq.com/developers/
[21]. Sri Venkateswara University. *Official Website.* Retrieved from https://svuniversity.edu.in/