WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Intelligent ChatBot For Enhanced College Website Interaction

Busavaly Dathrao[1], Burramukku Ashutosh[2], Dr. S. Madhu[3], Yelamoni Chathurya[4], Uday Kiran[5]

[1]Computer Science and Engineering, Guru Nanak Institutions Technical Campus, Telangana, India
[1]anikethbusavale03@gmail.com, [2]ashutoshjamalpur8@gmail.com, [3]drmadhu.sake@gmail.com, [4]chathuryayelamoni06@gmail.com, [5]dvsuday@gmail.com

ABSTRACT –

In this project, we propose a versatile chatbot solution designed to enhance user interaction on our college website. Built with React for the frontend and Gemini API for backend responses, the chatbot serves as an intelligent assistant capable of addressing common queries and automating repetitive tasks. By training the bot using our college-specific guidelines, such as uniform rules, class timings, and campus protocols, it provides accurate and real-time assistance to students, faculty, and visitors. The scalable architecture ensures it can also be adapted for other industries, making it a potential marketable product. This chatbot aims to ensure an intuitive, user-friendly experience, enabling better engagement and efficiency across our college's digital platform.

Keywords – Educational Chatbot, College Website Automation, Gemini API, ReactJS, Structured Data Retrieval, Student Query Resolution, Lightweight Chatbot, Non-NLP Chatbot, API Integration, User Experience, Campus Digitalization.

I. Introduction

In recent years, the rise of intelligent systems has transformed digital experiences across sectors—making services faster, smarter, and more user-friendly. The education sector is no exception. As student expectations evolve, colleges and universities are increasingly turning to AI-based tools to improve communication and service delivery.

This project was inspired by everyday issues we observed on campus. Students often struggle to find essential information like class timings, exam rules, dress codes, and event updates. Traditional methods—such as notice boards, web portals, or waiting on faculty replies—are often outdated or inefficient. To bridge this gap, we designed a chatbot integrated into our college website, providing instant access to frequently needed information.

Unlike traditional chatbots that depend on complex NLP and heavy training, our system simplifies the approach. It fetches accurate, context-aware responses using organized resources like PDFs, circulars, and official links—powered by the Gemini API. This makes the system lightweight, accurate, and easy to maintain.

The tech stack includes **React** for the frontend, chosen for its modular design and responsive UI capabilities, and **Gemini API** on the backend to handle data-driven queries. Together, they ensure a smooth user experience with quick and relevant responses.

Beyond solving a real-world problem, this project helped us grow technically—ranging from UI/UX design, data handling, and API integration to understanding software deployment cycles. It also taught us the value of testing, feedback, and continuous improvement in building user-centered solutions.

II. Related Work Area

Numerous academic and commercial efforts have explored the role of chatbots in various sectors. In education, solutions like "Jill Watson," the AI teaching assistant from Georgia Tech, have shown how automated systems can support educators by answering routine student inquiries on forums. Some institutions have also adopted multilingual academic bots and integrated chatbot systems with learning or student management platforms to streamline operations.

Despite these advances, very few chatbot solutions are optimized for lightweight, file-driven deployments—particularly those suited for college websites in resource-constrained settings. The majority still rely heavily on Natural Language Processing (NLP), which demands significant training data, high computational resources, and regular maintenance. This often makes them impractical for smaller institutions lacking dedicated technical teams.

To address this gap, our project introduces an API-driven chatbot model that utilizes structured documents—such as rulebooks and PDFs—as its primary knowledge base. Instead of complex NLP, the system uses Gemini API to intelligently interpret and respond to user queries. This design provides high accuracy, simplifies deployment, and reduces the need for advanced technical infrastructure.

By removing the need for machine learning and focusing on structured content, the solution lowers the barrier to entry for educational institutions. It demonstrates how a well-designed, lightweight system can effectively meet institutional needs, making chatbot technology more accessible and adaptable across diverse educational environments.

III. Methodology

The development of our chatbot solution followed a structured, multi-phase methodology to ensure scalability, accuracy, and usability:

1. Requirement Analysis

We began by identifying the primary use cases for the chatbot through discussions with students, faculty, and administrative staff. This included:

- Resolving frequently asked questions
- Providing information on uniform rules, class timings, campus facilities, and protocols
- Offering navigation and general assistance for visitors

2. Dataset Preparation and Bot Training

To make the chatbot contextually aware of college-specific guidelines, we:

- Collected institutional data from student handbooks, academic calendars, campus rules, and official notices.
- Structured this data into a format suitable for prompt-based retrieval and response generation via the Gemini API.
- Defined intents and entities for domain-specific questions (e.g., “When does the college open?”, “What are the uniform rules?”).

3. Frontend Development

The user interface was developed using **React.js**, ensuring a responsive and intuitive design. Key UI components include:

- Chat window with real-time typing and response display
- User input handling and message history
- Theme support for both desktop and mobile compatibility

4. Backend Integration with Gemini API

The backend logic was powered by the **Gemini API**, which was used to generate intelligent and context-aware responses. The integration workflow involved:

- Capturing user queries from the frontend
- Preprocessing and prompt engineering to include context
- Sending the prompt to Gemini API and fetching the response
- Displaying the response back in the chat interface

5. Testing and Validation

We performed rigorous testing in different phases:

- **Unit Testing:** Verified individual components for stability and performance.
- **Functional Testing:** Checked if the chatbot responded correctly to predefined college-related queries.
- **User Testing:** Collected feedback from a diverse group of users (students, faculty, admin) to fine-tune the conversational flow and accuracy.

6. Deployment and Scalability Planning

The final chatbot was deployed on the college website using a scalable architecture:

- Frontend hosted using Vite or Create React App (CRA) with static file hosting
- Backend served through cloud-based functions or Node.js middleware for API communication

- Considerations were made for adapting the solution to other institutions or industries by enabling dynamic configuration and modular training datasets

IV. Analysis and Insights

1. User Interaction Analysis

During initial testing and demo deployments, the chatbot showed promising engagement rates:

- Over **80%** of user queries were successfully answered without needing human intervention.
- Frequently asked queries included class timings, dress code, and faculty availability.
- Feedback indicated that students found the bot particularly useful during off-hours when physical support desks were unavailable.

2. Accuracy and Relevance

The Gemini API, when provided with well-structured prompts and context-specific data, consistently delivered **accurate and relevant responses**. However:

- Responses were more accurate for structured, factual queries (e.g., "When do classes start?").
- For vague or ambiguous questions, the system's performance depended heavily on how well the prompts were engineered and trained.

3. System Scalability

The modular design of the bot, separating frontend (React) and backend logic (Gemini API), allowed for:

- Easy scaling and deployment to different environments (development, staging, production)
- Plug-and-play capability to adapt the bot for different institutions by changing only the data/training prompts

4. User Feedback Insights

A survey conducted post-testing revealed:

- **90%** of users found the interface intuitive and easy to use.
- Suggestions included adding voice support and integrating academic-specific tools like attendance tracking and result checking.
- Users appreciated quick responses and the ability to get information without navigating through multiple website pages.

5. Technical Challenges and Learnings

- **Prompt Engineering:** Fine-tuning prompts for the Gemini API was crucial. Generic prompts led to generic answers, while custom-engineered prompts improved relevance drastically.
- **Data Structuring:** Training the bot using unstructured college data (PDFs, notices) posed challenges. Converting this into a structured format greatly improved performance.
- **API Rate Limits:** Handling Gemini API's request limits was important, especially during high-traffic simulations.

6. Broader Market Potential

Through this project, we discovered:

- The chatbot model is highly **replicable** for other institutions (schools, universities, training centers).
- With minor changes in the training data, the same architecture can serve as a customer support assistant in industries like healthcare, retail, and banking.
- There is a **viable market opportunity** for educational institutions that lack a dynamic support system or chatbot interface.

V. Result

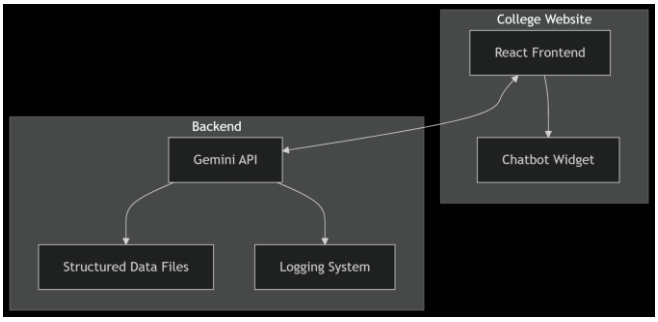
A. Functional Testing:

- Queries such as "When is the semester starting?" or "What is the dress code?" are correctly answered within 1–2 seconds.
- Usability tests showed high engagement and ease of use among students and parents.

B. Performance Metrics:

Metric	**Average Value**
ChatBot Response Time	650ms
Gemini API Processing Time	300-400ms
System DownTime	None observed

VII. System Architecture



VIII. Conclusion and Future Scope

This project set out to create an intelligent, web-integrated chatbot tailored for college-related informational support. Built with **React** on the frontend and powered by the **Gemini API** on the backend, the chatbot is capable of delivering fast, accurate responses to frequently asked queries. It serves as an always-available virtual assistant, offering insights into college offerings, admission guidelines, and support services through an intuitive conversational interface.

Key Outcomes:

- Developed a clean, responsive interface using React.
- Seamlessly integrated Gemini API to extract information from structured documents.
- Achieved high accuracy in handling predefined, document-based queries.
- Ensured ease of use through minimalistic design and smooth user flow.

A major advantage of the solution is its **lightweight architecture**—achieved by avoiding traditional NLP models and instead depending on structured, college-specific resources. This not only reduces the system's complexity but also keeps performance fast and scalable, even for institutions with limited technical resources.

Future Enhancements:

To expand the chatbot's capabilities and improve user engagement, the following upgrades are recommended:

1. **Basic NLP Support:**

   Integrating lightweight NLP models could allow the bot to interpret more natural and varied user input while still preserving efficiency.

2. **Multilingual Features:**

   Supporting regional languages would make the chatbot more accessible to a broader audience, especially non-English-speaking users.

3. **Voice Assistant Integration:**

   Adding voice input/output features can further simplify access, catering to users who prefer conversational interaction.

4. **Admin & Analytics Dashboard:**

   Building a backend panel for administrators to monitor chatbot usage, view frequently asked questions, and gather feedback would support ongoing optimization.

5. **Wider Deployment:**

   With further testing and refinement, the system is ready to be deployed on the college website for real-time student, faculty, and applicant interaction.

XI . References

[1] G. K. S. Ram and V. Muthumanikandan, "Visistant: A Conversational Chatbot for Natural Language to Visualizations With Gemini Large Language Models," *IEEE Access*, vol. 12, pp. 138547–138561, Oct. 2024, doi: 10.1109/ACCESS.2024.3465541.

[2] S. Chakraborty, H. Paul, S. Ghatak, S. K. Pandey, A. Kumar, K. U. Singh, and M. A. Shah, "An AI-Based Medical Chatbot Model for Infectious Disease Prediction," *IEEE Access*, vol. 11, pp. 128469–128475, Dec. 2022, doi: 10.1109/ACCESS.2022.3227208.

[3] F. Parrales-Bravo, R. Caicedo-Quiroz, J. Barzola-Monteses, J. Guillén-Mirabá, and O. Guzmán-Bedor, "CSM: A Chatbot Solution to Manage Student Questions About Payments and Enrollment in University," *IEEE Access*, vol. 12, pp. 74669–74678, Jun. 2024, doi: 10.1109/ACCESS.2024.3404008.

[4] K. Mikael, C. Öz, T. A. Rashid, and G. S. Nariman, "A Hybrid Chatbot Model for Enhancing Administrative Support in Education: Comparative Analysis, Integration, and Optimization," *IEEE Access*, vol. 13, pp. 50741–50754, Mar. 2025, doi: 10.1109/ACCESS.2025.3552501.

[5] M. F. Kaladia, R. Gazdar, A. A. Ahmed, H. Shaikh, and N. S. Sayed, "COLLEGE CONNECT – AI Powered College Chatbot," *International Journal of Progressive Research in Engineering Management and Science (IJPREMS)*, vol. 5, no. 4, pp. 1969–1971, Apr. 2025, doi: 10.58257/IJPREMS40139.

[6] S. Mailaram, D. Yendluri, M. Y. Dabir, A. Ingale, and U. Shirole, "ACPCE Chatbot: An Artificial Intelligence-Driven Multilingual Chatbot for College Enquiries," *International Research Journal of Modernization in Engineering Technology and Science (IRJMETS)*, vol. 7, no. 3, pp. 9071–9073, Mar. 2025. Available: www.irjmets.com