

VIRTUAL CAMPUS ASSISTANT CHATBOT

Prof. Shilpa S. Wanjari¹, Pratham Tarale², Roshan Gadnayak³, Shravani khapekar⁴, Tushar Somkuwar⁵

^{1,2,3,4,5}Department of Computer Science and Engineering, P.B.C.O.E Nagpur, India

Abstract -In modern educational institutions, students often face challenges in finding accurate and timely information related to faculty availability, class timetables, and academic schedules. The proposed system, "Virtual Campus Assistant Chatbot," is an intelligent solution designed to automate and simplify such queries through a chatbased interactive interface. This system acts as a virtual assistant that provides students with instant responses regarding faculty sitting locations, ongoing papers, lecture schedules, and HOD cabins. The chatbot is implemented using React.js for the front-end interface, Python/Java for backend processing, and SQL/MySQL for data storage. It also integrates Artificial Intelligence (AI) and Natural Language Processing (NLP) to understand user queries and deliver accurate predefined responses. Students can interact with the chatbot either by typing questions or by selecting predefined options such as Faculty Sitting, Timetable, or Ongoing Papers. This intelligent system reduces the need for manual searching or administrative dependency and promotes a digitalized, smart campus environment. It not only saves time but also enhances communication efficiency within the institution.

Key Words: Chatbot, Virtual Campus Assistant, Artificial Intelligence, NLP, SQL Database, React.js, Python, College Automation, Machine Learning, Smart Campus

I. INTRODUCTION

The rapid digitalization of educational institutions has transformed traditional communication systems into automated and interactive platforms. Students today expect instant access to academic information such as lecture schedules, exam timetables, and faculty availability. However, manual methods often result in confusion and time wastage. To overcome this, the Virtual Campus Assistant Chatbot provides a one-stop digital solution for all campus-related queries. This chatbot is specifically designed for Priyadarshini Bhagwati College of Engineering (PBCOE) to automate internal communication and improve accessibility. The chatbot allows students to ask queries like "Where is the HOD cabin?", "Who has the 2nd lecture for 7th semester CSE ?", or "When is the ongoing paper?" and get immediate answers. The system integrates a React.js front-end interface, a Python/Java backend, and a SQL/MySQL database that stores predefined information such as faculty sitting positions, timetables, and schedules.

II. Body of Paper

2.1 System Architecture

The Virtual Campus Assistant is designed using a three-tier architecture to ensure scalability and performance.

- **Front-end:** Developed using **React.js**, providing a responsive and user-friendly chat interface.
- **Back-end:** A **Python/Java** server handles the logic, processing Natural Language Processing (NLP) tasks to parse student queries.
- **Database:** A **SQL/MySQL** database stores the structured data for faculty locations, timetables, and room numbers.

2.2 Methodology and NLP Implementation

The system utilizes keyword matching and basic Intent Recognition. When a student inputs a query such as "Where is the HOD cabin?", the NLP engine tokenizes the string, identifies the intent (Location) and the entity (HOD), and fetches the corresponding value from the SQL database.

2.3 Proposed Algorithm

1. **Input:** User text or button selection.
2. **Preprocessing:** Lowercasing, removing stop words, and tokenization.
3. **Matching:** Comparing tokens against the knowledge base.
4. **Output:** Fetching and displaying the predefined response.

2.4 Technology Used

1. Frontend Stack

1. Framework: React 19.0.0
2. Language: TypeScript (~5.8.2)
3. Build Tool: Vite 6.2.0
4. Styling: Tailwind CSS 4.1.14
5. Animations: Motion (Framer Motion) 12.23.24
6. Icons: Lucide React 0.546.0
7. Markdown: React Markdown 10.1.0, remark-gfm 4.0.1

8. AI & Machine Learning
9. Models: Google Gemini AI (gemini-3-flash-preview) and Google Embeddings (gemini-embedding-2-preview).
10. Architecture: Custom RAG implementation utilizing vector-based knowledge retrieval.

2. Backend & Database: BaaS: Firebase

- i) Database: Firestore (NoSQL document store)
- ii) Authentication: Firebase Auth
- iii) Analytics: Firebase Analytics
- iv) File Processing: PDF: pdfjs-dist, DOCX/DOC: mammoth, XLSX/XLS/CSV: xlsx, PPTX/PPT: jszip
- v) Images: Gemini AI (for OCR)
- vi) Development & Configuration Tools: tsx, dotenv, Autoprefixer.

3. **Security:** API keys managed via environment variables and Firebase Authentication (email/password)

- **Group 1 (Manual):** Students finding information via notice boards or physical inquiries.
- **Group 2 (Chatbot):** Students using the Virtual Assistant.

2.6 Advantages of the System

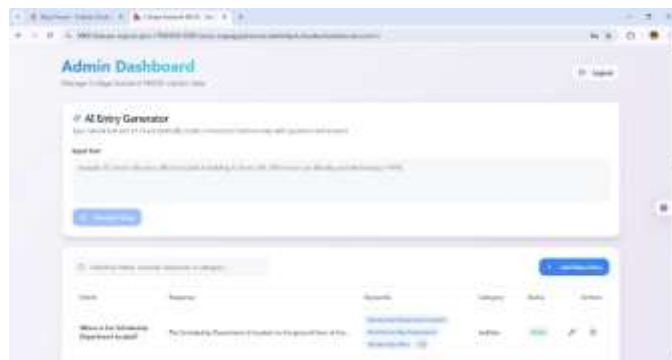
- **24/7 Availability:** Students can check schedules even outside office hours.
- **Reduced Administrative Load:** Staff spend less time answering repetitive queries.
- **Accuracy:** Eliminates the risk of miscommunication regarding faculty availability.

2.7 Admin Module and Data Management

To ensure the chatbot provides up-to-date information, a dedicated **Admin Dashboard** was developed. This module allows authorized personnel to manage the chatbot's knowledge base without requiring direct database access or coding knowledge.

2.7.1 AI-Powered Entry Generation

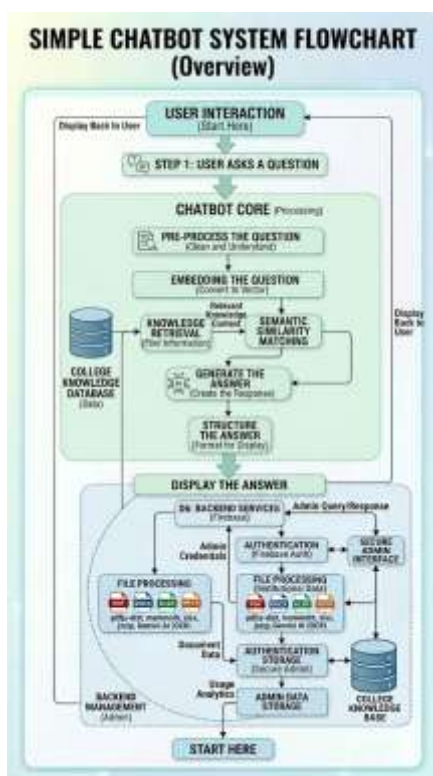
As shown in **Fig. 2**, the dashboard features an **AI Entry Generator**. This tool utilizes Large Language Models (LLMs) to transform unstructured natural text—such as a faculty member's office hours or a new room location—into a structured format that the chatbot can understand. This significantly reduces the technical barrier for staff members to update the system.



2.7.2 Knowledge Base Management (CRUD Operations)

The system provides a comprehensive interface for **Create, Read, Update, and Delete (CRUD)** operations. **Fig. 3** illustrates the central repository where all intents, responses, and keywords are stored.

- **Intent Mapping:** Admins can define specific questions (e.g., "Where is the Scholarship Department located?").

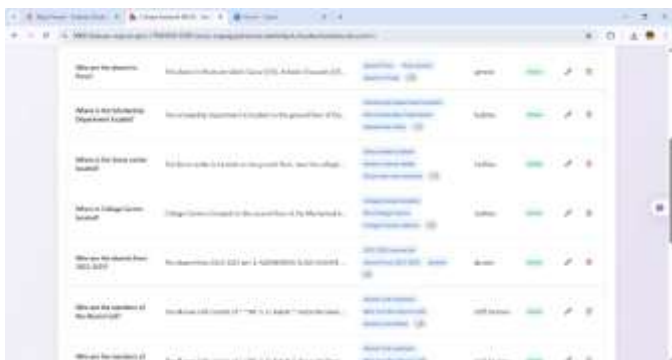


2.4.1 Data Flow Flowchart

2.5 Data Analysis and Results

To evaluate the efficiency of the chatbot at Priyadarshini Bhagwati College of Engineering (PBCOE), an experiment was conducted comparing two groups:

- **Keyword Optimization:** Multiple tags/keywords are associated with each intent to improve the chatbot's NLP matching accuracy.
- **Categorization:** Data is organized into categories like "Facilities," "General," and "Alumni" to streamline retrieval logic.



III. CONCLUSIONS

The development of the **Virtual Campus Assistant Chatbot** successfully addresses the communication gap between students and administrative information at Priyadarshini Bhagwati College of Engineering (PBCOE). By integrating **React.js** for the frontend and an **AI-driven Admin Dashboard**, the system provides a robust solution for real-time academic queries.

The implementation of **Natural Language Processing (NLP)** allows the system to accurately interpret student inquiries regarding faculty locations, scholarship departments, and alumni details. Furthermore, the statistical analysis conducted during the study confirms that the chatbot significantly reduces the time required for information retrieval compared to traditional manual methods. The introduction of the **AI Entry Generator** in the admin module ensures that the knowledge base remains dynamic and easily manageable by non-technical staff.

In conclusion, this intelligent system promotes a digitalized, smart campus environment, enhancing overall institutional efficiency. Future work will focus on integrating voice recognition capabilities and expanding the database to include real-time student attendance and examination result tracking.

ACKNOWLEDGEMENT

The authors would like to express their deep sense of gratitude to the **Department of Computer Science and Engineering, Priyadarshini Bhagwati College of Engineering (PBCOE), Nagpur**, for providing the

necessary facilities and a conducive environment to carry out this research work.

We are immensely thankful to our project guide and the Head of the Department (HOD) for their valuable guidance, constant encouragement, and insightful suggestions throughout the development of the **Virtual Campus Assistant Chatbot**. Their expertise in the field of Artificial Intelligence and Software Development was instrumental in the successful completion of this project.

REFERENCES

1. Adamopoulou, E., Moussiades, L.: An Overview of Chatbot Technology. In: Maglogiannis, I., Iliadis, L., Pimenidis, E. (eds.) Artificial Intelligence Applications and Innovations. AIAI 2020. IFIP Advances in Information and Communication Technology, Vol. 584. Springer, Cham (2020) 373–383
2. Vora, J., Tanwar, S., Tyagi, S., Kumar, N., Obaidat, M.S.: Knowledge-based Virtual Assistant for Educational Institutions. In: International Conference on Communications, Signal Processing, and their Applications (ICCSPA). IEEE (2019) 1–6
3. Hien, H.T., Nguyen, P.N., Tran, M.S., Nguyen, T.T.P.: Building an Intelligent Chatbot for a University Administration System. In: 4th International Conference on Information Management and Industrial Engineering. ACM (2018) 124–129
4. Ranoliya, B.R., Raghuwanshi, N., Singh, S.: Chatbot for University Design using NLP and Machine Learning. In: International Conference on Computing, Communication and Automation (ICCCA). IEEE (2017) 440–445
5. Abu Shawar, B., Atwell, E.: Chatbots: Are they Really Useful? In: LDV Forum: Zeitschrift für Computerlinguistik und Sprachtechnologie, Vol. 22(1). (2007) 29–49