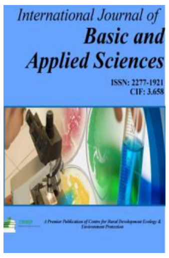
Content is available at: CRDEEP Journals
Journal homepage: http://www.crdeepjournal.org/category/journals/ijbas/

International Journal of Basic and Applied Sciences
(ISSN: 2277-1921) (Scientific Journal Impact Factor: 6.188)

UGC Approved-A Peer Reviewed Quarterly Journal



Research Paper

A College-Centric Virtual Assistant with Smart FAQ and File Management

Anand Kumar, Mohini Khattri, Madhav Gupta, Divyanshu Snehi and Dr. Shikha Tayal Aeron[1]
PG Scholar, Department of Computer Application Tula's Institute, Dehradun
[1]-*Associate Professor, Department of Computer Application Tula's Institute, Dehradun*

ARTICLE DETAILS

Corresponding Author:
Dr Shikha T. Aeron

Key words:
College Centric, Management

ABSTRACT

This paper presents a college-centric virtual assistant designed to streamline academic and administrative interactions within a college environment. The system integrates a smart FAQ feature powered by Natural Language Processing (NLP), enabling students and staff to obtain real-time answers to common queries. Additionally, the virtual assistant offers file management capabilities, allowing users to easily upload, retrieve, and manage academic materials like syllabi, notes, and assignments. Built using modern web technologies, this solution enhances communication efficiency, improves accessibility, and reduces the administrative burden, ultimately fostering a more productive and user-friendly campus experience.

1. Introduction:

In the dynamic environment of a college campus, students, faculty, and administrative staff often find themselves entangled in a web of scattered information. Despite the sincere efforts of the college administration, heads of departments, and faculty members to keep students informed, crucial updates and information sometimes fail to reach the right people at the right time. Students frequently encounter delays in getting answers to their questions, often relying on visits to the admin office or their mentors—who may be preoccupied with their own responsibilities. This process can be time-consuming, frustrating, and inefficient. To address this recurring issue, we envisioned and developed the College Query Chatbot, a smart solution designed to provide immediate, relevant, and clear responses to queries related to academic procedures, course materials, and administrative tasks. Built using the Flask framework and advanced Natural Language Processing (NLP), this chatbot is capable of understanding context rather than just matching keywords. It utilizes semantic search through pre-trained Sentence Transformer models, allowing it to retrieve the most accurate information. Moreover, it offers features like downloadable study materials and syllabus access through the chat interface, all managed via an intuitive admin panel. Our aim is to streamline the flow of information and significantly enhance the user experience within the college ecosystem. Accessing timely and accurate information within a college environment often becomes a challenge for students and staff alike. Traditional approaches like navigating static websites, relying on busy mentors, or contacting the administrative office frequently lead to inconsistent or delayed responses. Moreover, existing chatbot systems typically lack the capability to understand complex queries and often fail to deliver contextually relevant answers. This results in user frustration and wasted time. Our project, College Query Chatbot, addresses these issues by offering a responsive, AI-driven solution that bridges the gap between users and the information they seek, improving efficiency and user satisfaction.

2. Methodology:

The College Query Chatbot is developed using the Flask web framework, serving as the foundation for both the user interface and backend logic. When a user submits a query, the system first checks for the presence of keywords related to downloadable resources such as "syllabus" or "notes." If detected, it performs a keyword-based search across a structured file database containing metadata like display name, keywords, and file type. Matching files are returned as downloadable links. If the query is not file-related, it is processed using semantic FAQ matching. FAQs are stored in a relational database and are pre-processed using the Sentence Transformer model (all-MiniLM-L6-v2) to generate vector embeddings at application startup or upon updates. The user's query is similarly converted into an embedding, and cosine similarity is

[1]Corresponding Author can be contacted at *Department of Computer Application Tula's Institute, Dehradun*

Received: 18-May-2025; Sent for Review on: 22-May-2025; Draft sent to Author for corrections: 28-May-2025; Accepted on: 12-June-2025; Online Available from 16-June-2025
DOI: 10.13140/RG.2.2.25470.50247

used to identify the most relevant FAQ. If the similarity score exceeds a threshold, the matching response is returned. An admin panel supports full CRUD operations on both FAQs and files.

3. Workflow:

The College Query Chatbot follows a structured methodology encompassing several key phases:

1. **User Submits Query:** This initial step involves the user interacting with the chatbot's front-end. They can either type their question into a text input field or, using the voice input feature, speak their query. This query then serves as the input to the system.
2. **Convert Query to Embedding:** Once received, the user's raw text query is converted into a high-dimensional numerical vector, known as an embedding. This process is performed by the Sentence Transformer model, which captures the semantic meaning and contextual nuances of the query, allowing for more sophisticated comparisons than simple keyword matching.
3. **Check for Keywords:** The system's intelligence begins by performing an initial scan of the user's query. It looks for specific keywords or phrases that are commonly associated with requests for documents, such as "syllabus," "notes," "timetable," or "question paper."
4. **Keyword-Based Search:** If the keyword check identifies terms suggesting a file request, the system then initiates a search within its dedicated "Managed File" database. This search focuses on the metadata associated with stored documents (like display name, keywords, file type, or course code) to find direct matches.
5. **Process Non-file Query:** If the initial keyword scan does not indicate a file request, or if a file request was made but no matching documents were found in the database, the system then proceeds to treat the query as a general information request.
6. **Return Downloadable Links:** When the keyword-based file search successfully identifies relevant documents, the chatbot generates and presents direct HTML download links to these files within the chat interface, enabling the user to easily access the requested material.
7. **Return Matching Response:** For general queries, the system utilizes the previously generated query embedding. It then compares this embedding to pre-computed embeddings of all existing FAQ questions using cosine similarity. The FAQ with the highest similarity score (above a set threshold) is deemed the most relevant, and its corresponding answer is retrieved and provided to the user.
8. **Admin Panel Operations:** This refers to the separate, secure web interface accessible only to authorized administrators. Through this panel, staff can perform crucial content management operations, including adding new FAQs, editing existing answers, deleting outdated entries, and uploading.

As Figure illustrates, the system architecture of the chatbot, where the Flask backend connects the NLP module, database, and file management system to handle user queries and provide admin-level control for FAQ and document management.

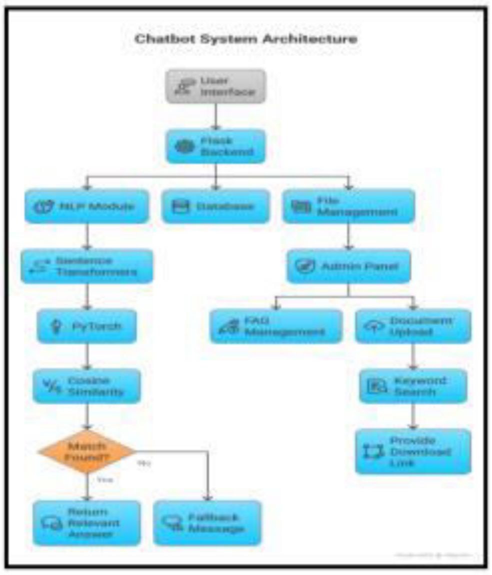
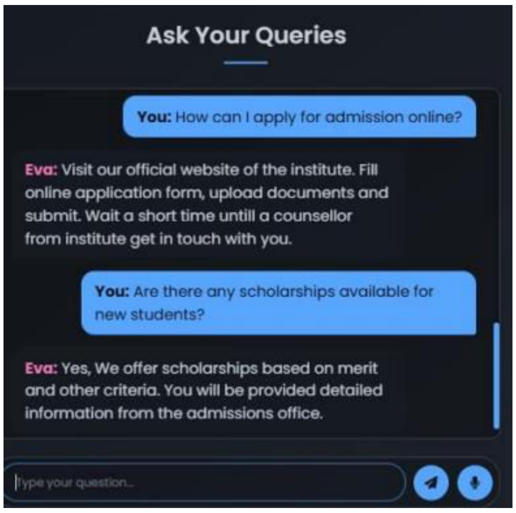


Fig 1: System architecture

3. Results

The implemented system's user interface (Fig. 3) demonstrates the chatbot, "Eva," successfully providing a contextually relevant FAQ answer regarding admission and scholarship.



Ask Your Queries

You: How can I apply for admission online?

Eva: Visit our official website of the institute. Fill online application form, upload documents and submit. Wait a short time untill a counsellor from institute get in touch with you.

You: Are there any scholarships available for new students?

Eva: Yes, We offer scholarships based on merit and other criteria. You will be provided detailed information from the admissions office.

Type your question...

Fig3: ChatbotInterface

The admin panel interface (Fig. 4) showcases the dashboard for managing FAQs, including options for adding new entries and uploading CSV files, illustrating the system's content management capabilities.

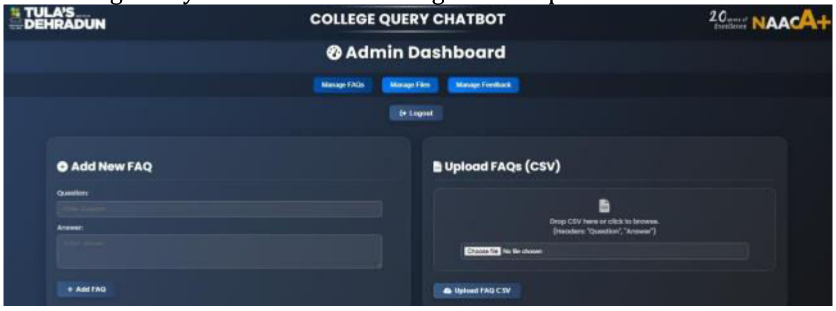


Fig 4: AdminPanel Interface

4.Conclusion

This project successfully implements a Flask-based intelligent chatbot that integrates semantic search for context-aware FAQ handling and an efficient document retrieval system. It demonstrates a significant improvement over basic chatbot designs by delivering more relevant responses, enhancing the user experience, and serving as a reliable information medium for college-related queries.

5. Future Scope

Looking ahead, the chatbot could evolve to better understand ongoing dialogues, remembering previous questions to answer follow-ups more intelligently. We're also keen to explore how newer AI like large language models might help generate even more natural-sounding replies or even find answers directly within uploaded documents. Right now, finding files relies on keywords, but a smarter, meaning-based search for documents would be a big step up. Gathering user opinions on how helpful the answers are could also help us fine-tune the system. Imagine if the chatbot could automatically know you're an MCA student in your final year and offer exactly the right project guidelines or syllabus without you even asking precisely! Finally, if the bot notices many people asking similar things, it could perhaps start offering that information upfront.

Reference

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). *BERT: A deep learning approach for context-aware language modeling.* Preprint retrieved from arXiv:1810.04805.

Flask Documentation Team. (n.d.). *Official guide to web development with Flask, including routing, templates, and server configuration.* Retrieved from Flask's official documentation portal.

Grinberg, M. (2018). *Hands-on guide to developing Python web applications using Flask* (2nd ed.). O'Reilly Media.

Jurafsky, D., & Martin, J. H. (2023). *A comprehensive overview of natural language processing and speech technology* (3rd ed., draft).

Reimers, N., & Gurevych, I. (2019). *Development of sentence-level semantic representations using Siamese architecture with BERT.* In *Proceedings of EMNLP-IJCNLP 2019*.

SQLAlchemy Developers. (n.d.). *SQLAlchemy ORM: A toolkit for database schema modeling and object-relational mapping in Python.* Available from SQLAlchemy.org documentation.