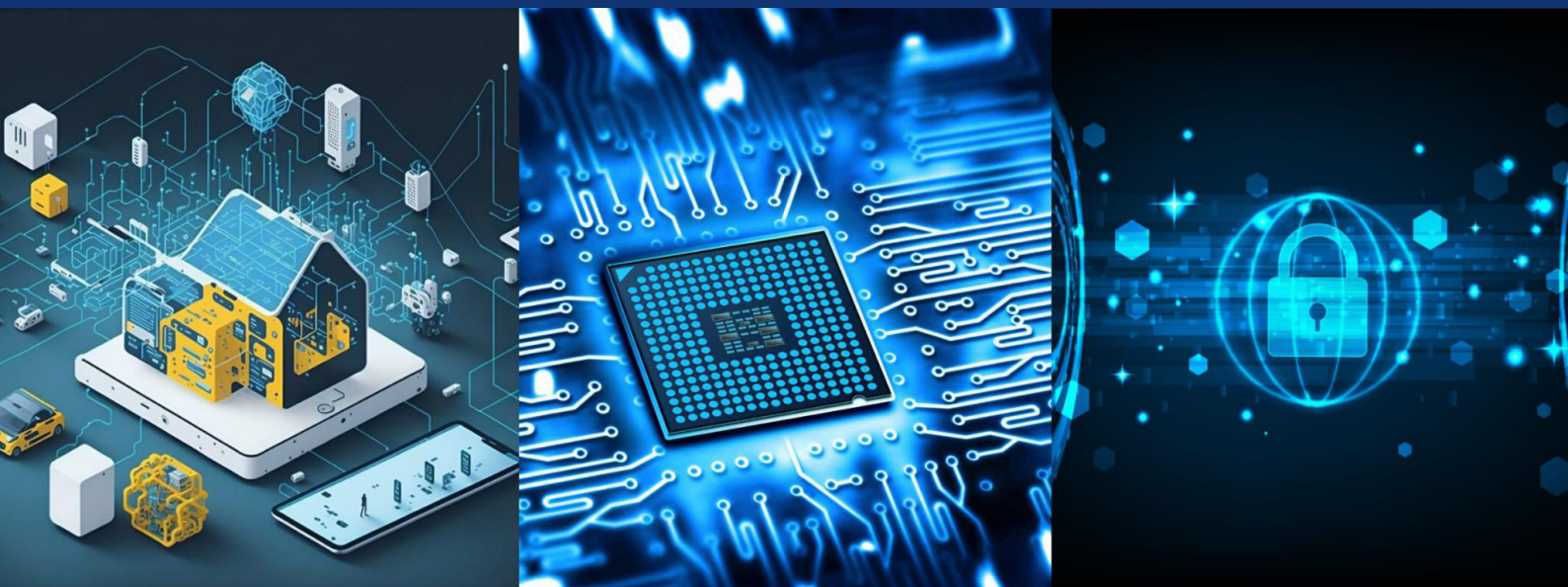
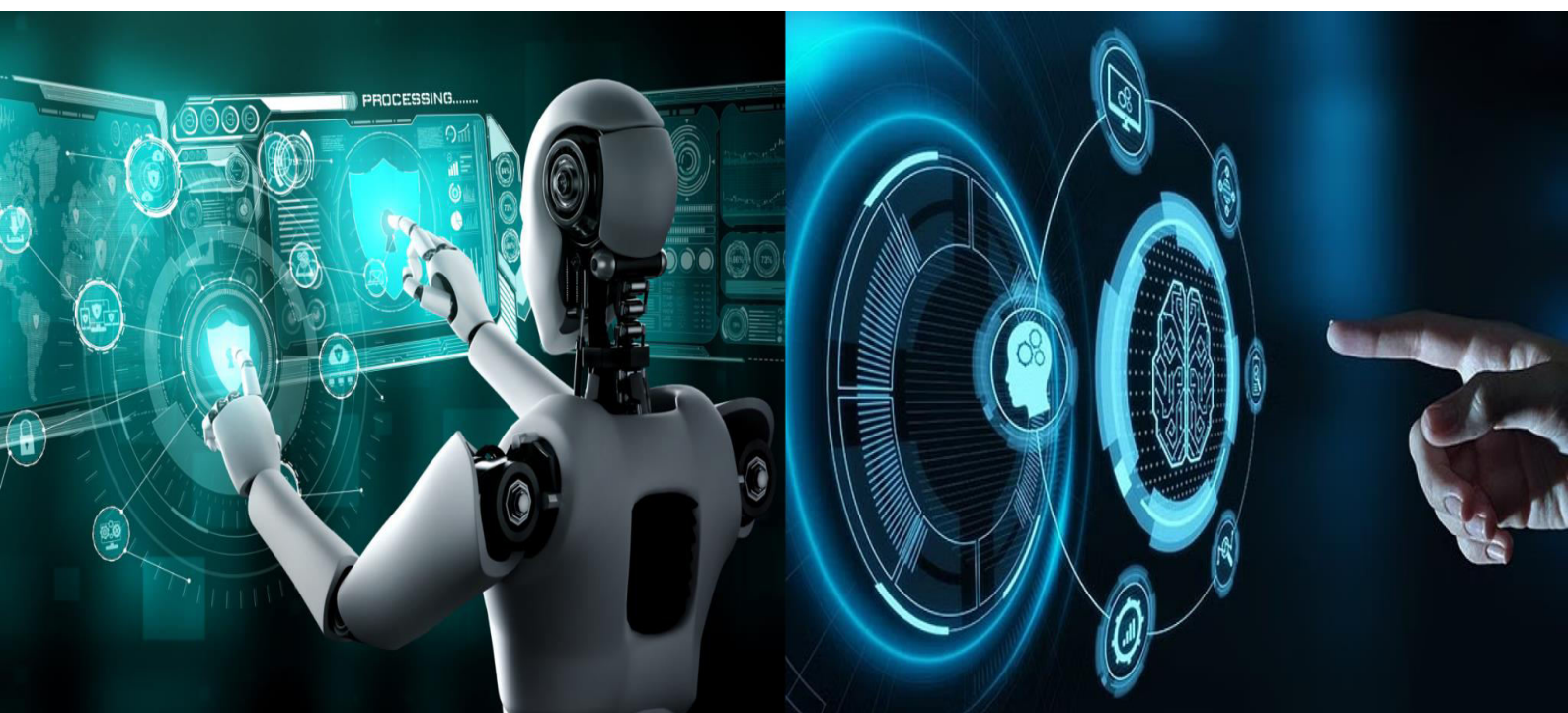




International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)





International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

AI Based College Enquiry Chatbot

Taniya Anil Bhise¹, Anisha Atmaram Gawade², Pooja Rupesh Kesarkar³, Advita Tushar Shinde⁴,
Prof.P. H. Dongare⁵

Department of Computer Engineering, Yashwantrao Bhonsale Institute of Technology, Sawantwadi, MH, India¹⁻⁴

Guide, Department of Computer Engineering, Yashwantrao Bhonsale Institute of Technology, Sawantwadi, MH, India⁵

ABSTRACT: The AI-Based College Enquiry Chatbot is an intelligent and automated system developed to simplify the process of handling enquiries related to educational institutions. Students, parents, and other users often require information regarding admissions, courses, fee structures, scholarships, placements, academic calendars, examination schedules, and campus facilities. Traditional enquiry methods, such as phone calls, emails, and in-person visits, are time-consuming and require significant manual effort from administrative staff. These methods may also lead to delays, inconsistencies, and limited accessibility outside working hours.

To overcome these challenges, this project implements an AI-powered chatbot based on the Retrieval-Augmented Generation (RAG) architecture. The system collects information from official college documents, including PDF files, and processes the content by extracting text and dividing it into smaller sections. These sections are then transformed into vector embeddings and stored in a vector database to enable efficient information retrieval. Whenever a user submits a query, the chatbot identifies the most relevant information using semantic search techniques and generates an accurate and context-aware response through a Large Language Model (LLM).

The backend of the system is developed using Python and FastAPI, while the frontend is built using React.js (Next.js) and Tailwind CSS to create a responsive and user-friendly interface. PostgreSQL with the pgvector extension is used for efficient data storage and retrieval. Additional development tools, such as GitHub, Postman, Netlify, and Railway, are used to support deployment, testing, and maintenance of the system.

The proposed system significantly reduces the workload of administrative staff, provides consistent and reliable information, and offers twenty-four-hour accessibility to users. The chatbot improves communication efficiency, enhances user satisfaction, and provides a scalable solution for modern educational institutions. Future improvements may include Optical Character Recognition (OCR), conversation memory, hybrid search methods, role-based authentication, and analytical dashboards to further improve the overall performance of the system.

I. INTRODUCTION

In today's digital era, educational institutions handle a large number of enquiries from prospective students, parents, and current students regarding admissions, courses offered, fee structure, eligibility criteria, placements, scholarships, facilities, academic calendar, and various institutional policies. Traditionally, these enquiries are handled manually through phone calls, emails, or in-person visits. This approach often leads to delayed responses, repetitive workload for administrative staff, limited availability during working hours, and inconsistent communication.

The College Enquiry Chatbot developed in this project overcomes these limitations by using a Large Language Model (LLM) integrated with document-based retrieval techniques. The system is designed to extract information directly from official college documents such as PDFs. These documents are processed by extracting text, dividing it into meaningful chunks, and converting them into vector embeddings. When a user submits a query through the React JS-based web interface, the system retrieves the most relevant document chunks using semantic similarity and generates an accurate response using the LLM.

The backend of the system is implemented using Python, which handles document processing, embedding generation, query handling, and communication with the language model. The frontend is developed using React JS to provide an interactive and user-friendly chat interface. The integration of frontend and backend ensures seamless communication between users and the AI system.



International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

This solution provides a scalable and intelligent enquiry management system that delivers quick, consistent, and document-based responses. It significantly reduces manual intervention, improves efficiency, and enhances user experience by making institutional information accessible anytime and anywhere.

II. LITERATURE SURVEY

Paper / Source	Key Finding	Relevance to My Project
Paper 1: NLP-based Chatbot Systems	Natural Language Processing improves understanding of user queries in conversational systems.	Used for designing intelligent query handling and response generation module.
Paper 2: Machine Learning in Chatbots	ML models enhance intent recognition and accuracy over rule-based systems.	Helps in selecting MLbased intent classification for enquiry handling.
Paper 3: Cloud-based Deployment of AI Systems	Cloud platforms improve scalability, availability, and real-time access.	Supports decision to deploy chatbot on cloud for 24/7 accessibility.
Paper 4: Secure Authentication Mechanisms	Secure login and encryption techniques protect user data.	Used for implementing secure admin login and data protection.
Paper 5: Real-Time Data Processing in AI Applications	Real-time processing improves response speed and system efficiency.	Applied in designing instant response system for student enquiries.

III. PROBLEM STATEMENT & MOTIVATION

3.1 Problem Statement

Educational institutions face many challenges in managing student inquiries and information. They receive a large number of repetitive questions about admissions, fees, exams, timetables, and results. Staff members often spend a lot of time answering the same queries again and again. In addition, staff availability is limited after office hours, on weekends, or during holidays, which causes delays in responding to emails and phone calls. Sometimes, students receive different information from different staff members, which creates confusion.

Another major issue is that important information is stored in multiple documents and files, so staff need to manually search for the required details. This process is time-consuming and inefficient. Due to these problems, students may feel frustrated and dissatisfied. Therefore, there is a need for a smart and automated system that can understand questions in simple language and provide quick, accurate, and consistent answers at any time.

3.2 Proposed Solution

The proposed solution is an AI-based chatbot designed using a Retrieval-Augmented Generation (RAG) architecture to provide accurate and reliable information. The system collects data from multiple sources such as PDFs, Word documents, and databases, and processes this information by converting it into embeddings. These embeddings are securely stored in a vector database for efficient searching. When a user asks a question, the chatbot retrieves the most relevant information from the stored data and uses a Large Language Model (LLM) to generate clear, accurate, and context-based responses. This solution minimizes manual effort, ensures consistent information delivery, saves time, and significantly improves efficiency and user satisfaction by providing quick and reliable answers anytime.



International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

VI. TECHNOLOGY STACK SELECTION

4.1 Programming Language

The project uses multiple programming languages to ensure smooth development and efficient performance. Python is used for backend programming because it is simple, powerful, and widely used for building APIs and handling data processing tasks. It helps in managing server-side logic and integrating AI functionalities. JavaScript, specifically with Next.js, is used for frontend development to create an interactive and dynamic user interface. SQL is used for database management, allowing structured storage, retrieval, and manipulation of data. HTML is used to design the structure of web pages, while CSS with Tailwind is used to style and enhance the appearance of the user interface. Together, these programming languages work efficiently to build a complete, responsive, and scalable system.

4.2 Framework

The project uses FastAPI as the backend framework to build and manage APIs efficiently, allowing smooth communication between the frontend and the database. For the frontend, Next.js (React) is used to create a responsive and user-friendly interface with better performance and fast loading. Additionally, pgvector is integrated as a vector search extension for PostgreSQL, enabling efficient storage and retrieval of embedding vectors for accurate and context-based information search.

4.3 Database

The project uses Supabase PostgreSQL as the main database to store and manage all system data efficiently. PostgreSQL is a powerful and reliable relational database that ensures secure and structured data storage. With the integration of pgvector, the database can store embedding vectors generated from document content. It stores uploaded PDF metadata such as file names and details, as well as text chunks extracted from PDFs for processing. The embedding vectors are saved to enable fast and accurate similarity search. The database also stores admin credentials for secure access control. Additionally, it can store chat responses for optional logging and future reference. Overall, the database ensures efficient data management, quick retrieval, and secure storage for the entire system.

4.4 Frontend

Next.js with Tailwind CSS is used to develop the frontend of the system. It helps in creating an interactive, responsive, and user-friendly chatbot interface. The design ensures smooth navigation, fast performance, and compatibility across different devices.

4.5 Development Tools

The development tools used in this project include Visual Studio Code as the integrated development environment (IDE) for writing and managing code efficiently. GitHub is used for version control to track changes and manage project collaboration. Postman is used for testing APIs to ensure proper communication between the frontend and backend. Railway or Render is used for backend hosting to deploy the server online, while Netlify is used for frontend hosting to make the chatbot interface accessible on the web.



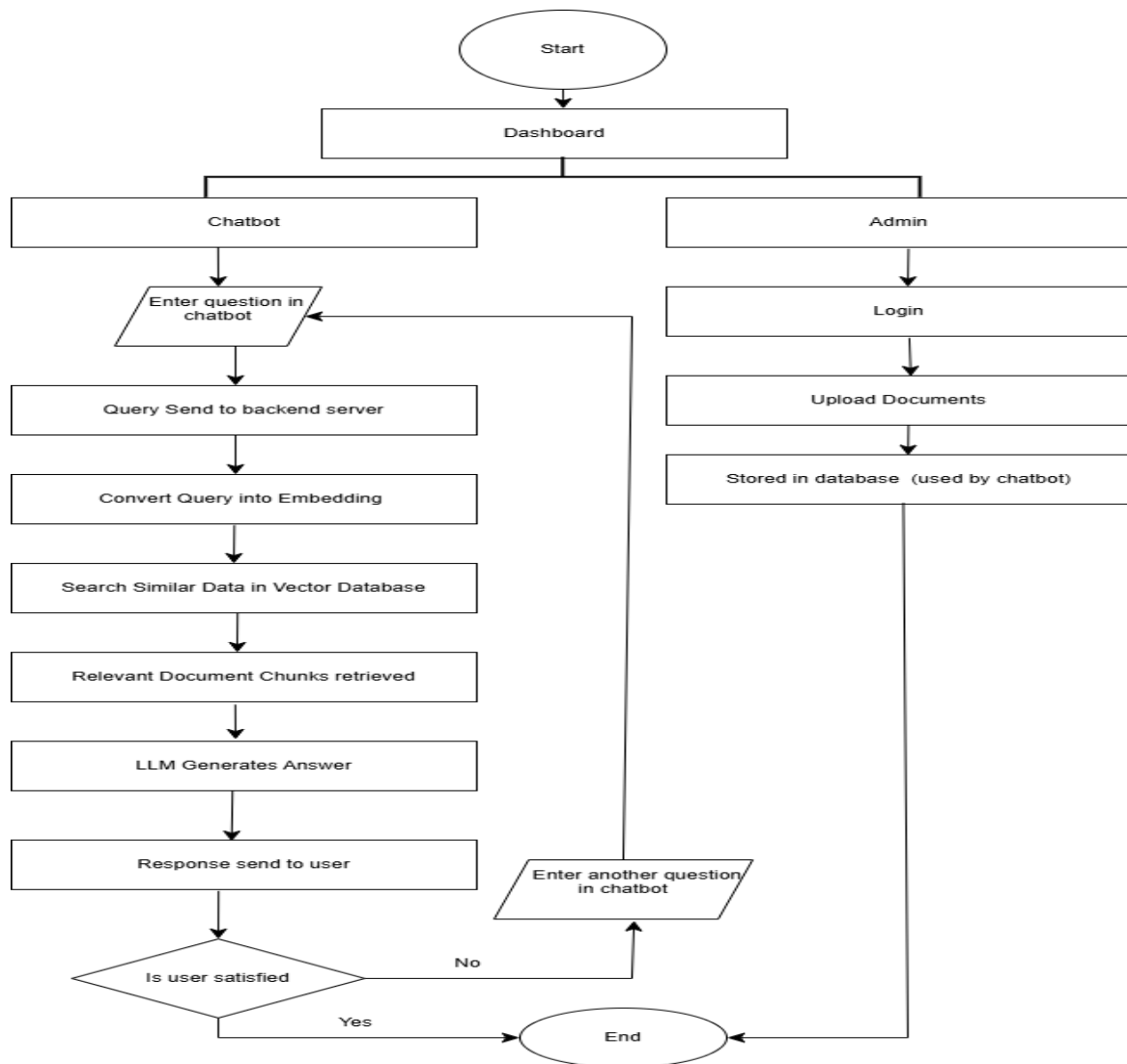
International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

V. SYSTEM MODELING

System modeling represents the structural and functional design of the AI Based College Enquiry Chatbot. It helps in understanding the system workflow, data flow, and interaction between different components before implementation. This section uses diagrams and models to clearly explain how user inputs are processed and how outputs are generated.

5.1 Flowchart



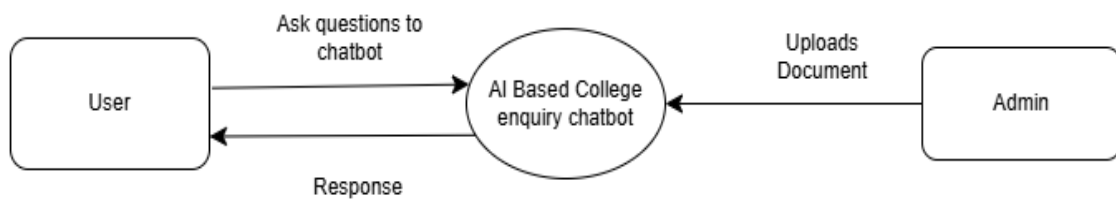


International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

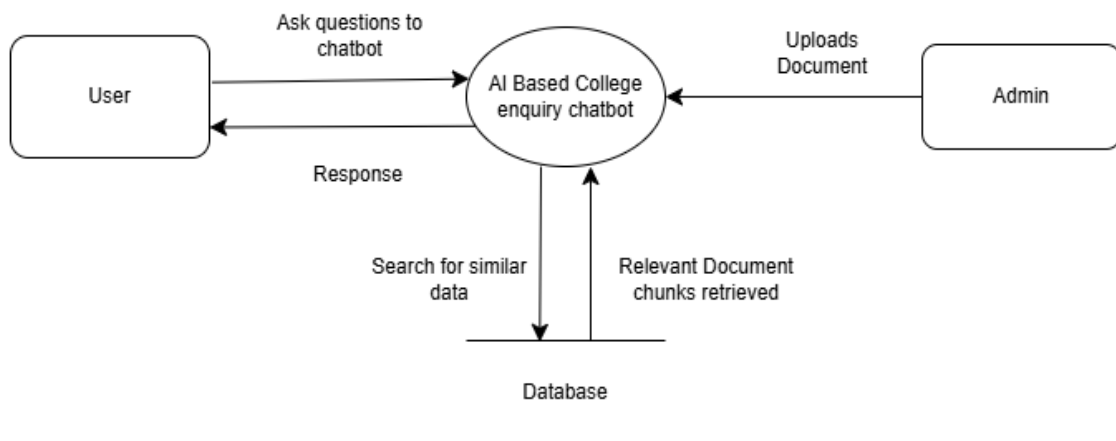
(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

5.2 Data Flow Diagram

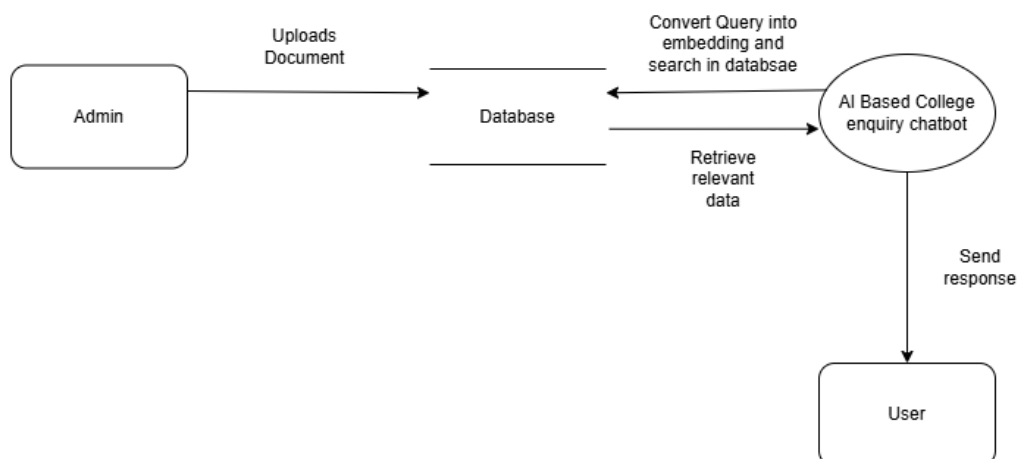
Level 0:



Level 1:



Level 2:





International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

VI. CONCLUSION AND FUTURE WORK

6.1 Conclusion

The College Enquiry Chatbot successfully demonstrates the application of Artificial Intelligence in automating institutional enquiry systems. By integrating a Python-based backend with FastAPI, Sentence Transformers for embeddings, FAISS for semantic retrieval, and an open-source LLM within a RAG architecture, the system delivers accurate and context-aware responses based on official college documents.

The use of a Gradio-based chat interface ensures simple and interactive user communication, while deployment on free cloud platforms makes the solution cost-effective and accessible. Overall, the system improves response efficiency, reduces administrative workload, and enhances user experience by providing reliable, document-driven information in real time. It serves as a scalable and practical solution for modern educational institutions.

6.2 Future Work

In future work, several improvements can be added to enhance the system's performance and functionality. Optical Character Recognition (OCR) can be integrated to support scanned PDFs and extract text from images. Conversation memory can be added to allow the chatbot to remember previous interactions and provide more context-aware responses. Role-based authentication can be implemented to ensure secure access control for different types of users. Hybrid search combining BM25 and vector search can be introduced to improve the accuracy of information retrieval. Additionally, an analytics dashboard can be developed to monitor user queries, system performance, and overall usage statistics.

REFERENCES

- [1] E. Rich and K. Knight, *Artificial Intelligence*, 3rd Edition, Tata McGraw Hill, 2017.
- [2] Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach*, 3rd Edition, Pearson, 2010.
- [3] J. Weizenbaum, "ELIZA—A Computer Program for the Study of Natural Language Communication," *Communications of the ACM*, Vol. 9, Issue 1, pp. 36–45, 1966.
- [4] A. Shawar and E. Atwell, "Chatbots: Are They Really Useful?", *Journal for Language Technology and Computational Linguistics*, 2007
- [5] FastAPI. (n.d.). *FastAPI Official Documentation*. <https://fastapi.tiangolo.com/>
- [6] Hugging Face. (n.d.). *Hugging Face Transformers Documentation*. <https://huggingface.co/docs/transformers>
- [7] Gradio. (n.d.). *Gradio User Documentation*. <https://gradio.app/>
- [8] Facebook AI Research. (n.d.). *FAISS: Library for Efficient Similarity Search*. <https://github.com/facebookresearch/faiss>
- [9] OpenAI. (n.d.). *Introducing GPT Models and Usage*. <https://openai.com/research>



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

 9940 572 462  6381 907 438  ijircce@gmail.com



www.ijircce.com

Scan to save the contact details