WWW.IJRPR.COM

International Journal of Research Publication and Reviews

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

Intelligent College Enquiry Chatbot using Google Gemini, Django, and MongoDB

Mrs. V Vanaja[1], P Shivakoti[2], Vishnubhatla Archana[3], Pabbathi Praveen[4]

[1]Assistant Professor, III B. Tech, Department of CSE (Data Science), ACE Engineering College, Hyderabad, India
[2,3,4]UG Student, III B. Tech, Department of CSE (Data Science), ACE Engineering College, Hyderabad, India

ABSTRACT:

In the modern educational environment, there is a growing need for efficient and automated systems that provide quick access to college-related information. This project presents a smart College Enquiry Chatbot designed for ACE Engineering College, powered by Google Gemini, to deliver accurate and instant responses to student and visitor queries. The chatbot is built using the Django web framework for backend integration and MongoDB as the database for storing and retrieving college data. Google Gemini is used to directly understand user questions, convert them into appropriate MongoDB queries, and generate clear and relevant responses. The system covers information such as academic timetables, faculty details, admission procedures, placement data, library books, notes, textbooks, and more. It also provides graphical insights into placement statistics and college performance. By integrating this chatbot into the college website, the project ensures 24/7 access to information, reduces administrative workload, and enhances the overall user experience. This solution demonstrates how advanced AI models and modern web technologies can be combined to build an effective and intelligent college Enquiry system

Keywords: College Inquiry Chatbot, Google Gemini, AI-powered assistant, Django, MongoDB, timetables, downloadable resources, Dual-mode Chatbot, Web-based Chatbot

INTRODUCTION:

In the current digital era, students, parents, and visitors expect fast and reliable access to college information. To address this need, the College Inquiry Chatbot for has been developed as an AI-powered virtual assistant. This chatbot uses Google Gemini to directly understand user queries and provide instant, accurate responses related to academic courses, faculty details, timetables, placement statistics, and available study materials. The backend is built with the Django framework, while all college data is securely stored in a MongoDB database. Users can easily access downloadable resources such as textbooks and notebooks through the chatbot interface.

Apart from handling college-specific queries, the chatbot is also capable of answering general questions about current events, general knowledge, and personal assistance, making it a versatile tool for users. By integrating this system into the college website, the chatbot ensures 24/7 availability, reducing the workload on administrative staff and enhancing the overall user experience. This project demonstrates how AI technology and modern web development can be combined to create an efficient, interactive, and user-friendly solution for college information dissemination.

The use of Google Gemini in this project allows for seamless query interpretation without relying on separate natural language processing modules, simplifying the architecture while maintaining high accuracy in understanding user intent. By converting user inputs directly into database queries, the chatbot efficiently retrieves relevant information and presents it in an easy-to-understand format. This approach not only improves response speed but also ensures that the chatbot can handle a wide variety of questions effectively. Overall, this project highlights the potential of combining AI-driven language models with full-stack web technologies to transform how educational institutions interact with their communities.

EXISTING SYSTEM

Currently, most colleges rely on manual administrative processes or static websites to provide information to students. Students must visit the college office, contact faculty, or browse the college website to get details about admissions, timetables, placements, and other academic resources. Some colleges provide inquiry services via email or phone calls, which can be slow and inefficient. Additionally, traditional FAQ pages on college websites offer limited, predefined answers that may not cover all student queries.

1. In most engineering colleges, student queries are handled through traditional channels such as email, phone calls, and in-person visits to administrative offices.

2. While these methods serve their basic purpose, they have several limitations that reduce the efficiency and quality of student support services.
3. Email communication, though convenient, often leads to delayed responses. Administrative staff, overwhelmed with large volumes of emails, may take hours or even days to reply to queries.
4. Phone calls pose challenges as well; staff availability is restricted to office hours, limiting access for students who need help during evenings, weekends, or holidays.
5. During peak times, phone lines can become congested, leading to long wait times and further delays in addressing student concerns.
6. Additionally, phone conversations do not create a permanent record of communication, making it difficult to track or follow up on unresolved issues.
7. In-person visits, while allowing face-to-face interaction, are not always practical. Students off-campus or with busy schedules may find it hard to visit the office. These visits can also lead to long queues and waiting periods, especially during high-demand times.

PROPOSED SYSTEM

The proposed system is an intelligent College Enquiry Chatbot designed to assist users in obtaining information related to the college without visiting the administration physically. This system uses Django as the backend framework, MongoDB as the database, and HTML, CSS, and JavaScript for the frontend interface. It ensures users can access structured data like faculty details, timetables, placement statistics, and study materials efficiently.

Unlike traditional chatbots, this system integrates Google Gemini to convert natural language queries into MongoDB queries, enabling users to retrieve specific college-related data stored in the database. This ensures that the responses are accurate and relevant to the institution's data. When a user asks a query related to the college, the chatbot interprets it, fetches the corresponding information from MongoDB, and presents it in a user-friendly format.

Additionally, the chatbot is also capable of handling general queries unrelated to the college. In such cases, the request is passed directly to Google Gemini, which generates appropriate responses using its language understanding capabilities. This dual-function approach makes the chatbot not only a tool for college enquiries but also a general-purpose assistant for broader queries, enhancing its usefulness and user experience.

ADVANTAGES OF THE PROPOSED SYSTEM

1. **AI-Powered Query Handling:**

   The chatbot uses a combination of predefined responses and plans for integration with advanced NLP models (OpenAI/Gemini) to process and respond to user queries in real time.

2. **Real-Time Data Retrieval:**

   User queries will be converted into MongoDB queries using Google Gemini, enabling the chatbot to fetch live information from the college’s database.

3. **Multi-Platform Access:**

   The chatbot will be embedded in the college website and accessible from both desktop and mobile devices, ensuring students can use it anytime, anywhere.

4. **User-Friendly Interface:**

   The frontend, built using HTML, CSS, and JavaScript, provides an intuitive chat widget where students can type or select their questions easily.

5. **Secure Backend:**

   Flask is used for backend logic to ensure secure query processing. The chatbot's data handling strictly follows privacy guidelines to protect student information.

6. **Modular Architecture:**

   The system is designed with a clear separation of concerns: frontend, backend processing, and data storage. This modularity allows for easy updates and future scalability.

METHODOLOGY

1. **Data Collection and Storage:**

All relevant college information such as faculty details, academic timetables, placement statistics, and study resources (textbooks, notebooks, etc.) are collected and stored in a MongoDB database. This database supports flexible storage of structured and unstructured data.

2. Backend Development:

The backend of the chatbot is developed using the Django web framework. Django manages user requests, communicates with the database, and handles the integration with Google Gemini.

3. User Query Processing:

When a user submits a query through the chatbot interface, the input text is sent to Google Gemini. Gemini processes the query by understanding its intent and converts it directly into a MongoDB database query.

4. Data Retrieval:

The MongoDB query generated by Gemini is executed on the database to fetch relevant college information or resources.

5. Response Generation:

Google Gemini then takes the retrieved data and generates a clear, concise, and user-friendly response, which is sent back to the user via the chatbot interface.

6. General Query Handling

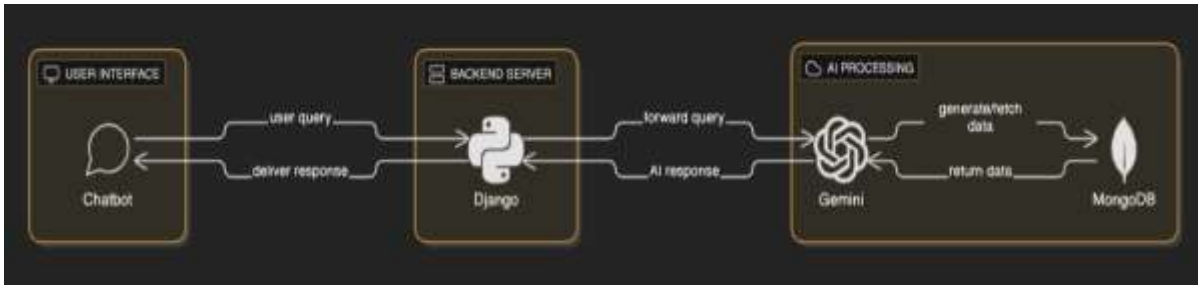
For questions unrelated to the college (e.g., general knowledge or personal assistance), Gemini processes and responds to them directly..

7. User Interface:

The chatbot is embedded into the college website, providing users with 24/7 access to information and downloadable materials, ensuring a seamless and interactive user experience.

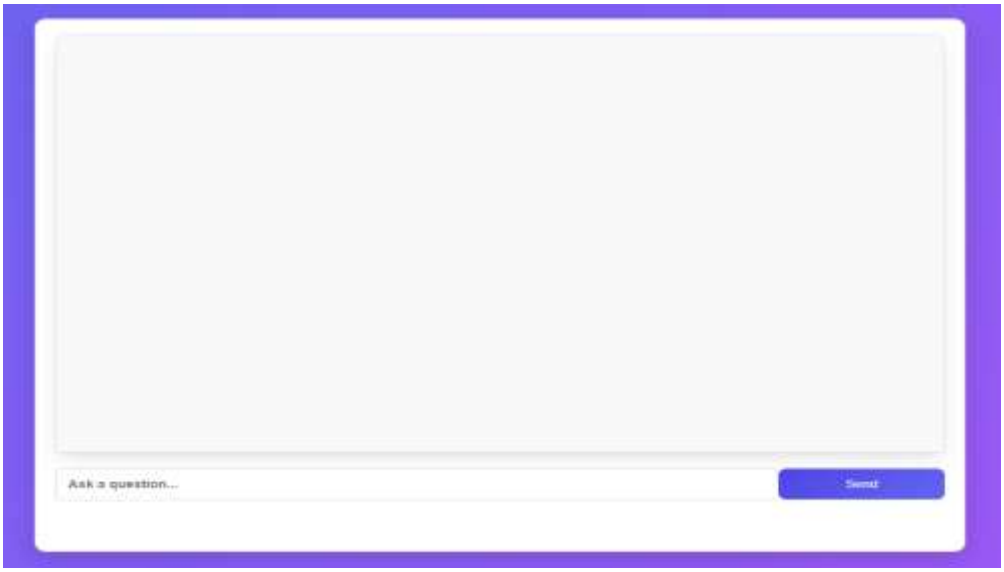
SYSTEM ARCHITECTURE

System architecture is a comprehensive blueprint that defines the structure, behavior, and interactions of various components within a system, whether it's a software application, a computer system, or a complex network of systems. It provides a high-level view of how the system is organized and how different parts, such as hardware, software, data storage, processing units, communication protocols, and user interfaces interact to perform specific functions. In software systems, architecture describes how modules or services are divided, how they communicate (e.g., via APIs or message queues), and how data flows through the system. Hardware systems include the design of processors, memory units, input/output devices, and how they are connected. System architecture also includes considerations for scalability (handling growth in users or data), security (protecting data and operations), maintainability (ease of updates and debugging), and performance (speed and efficiency).

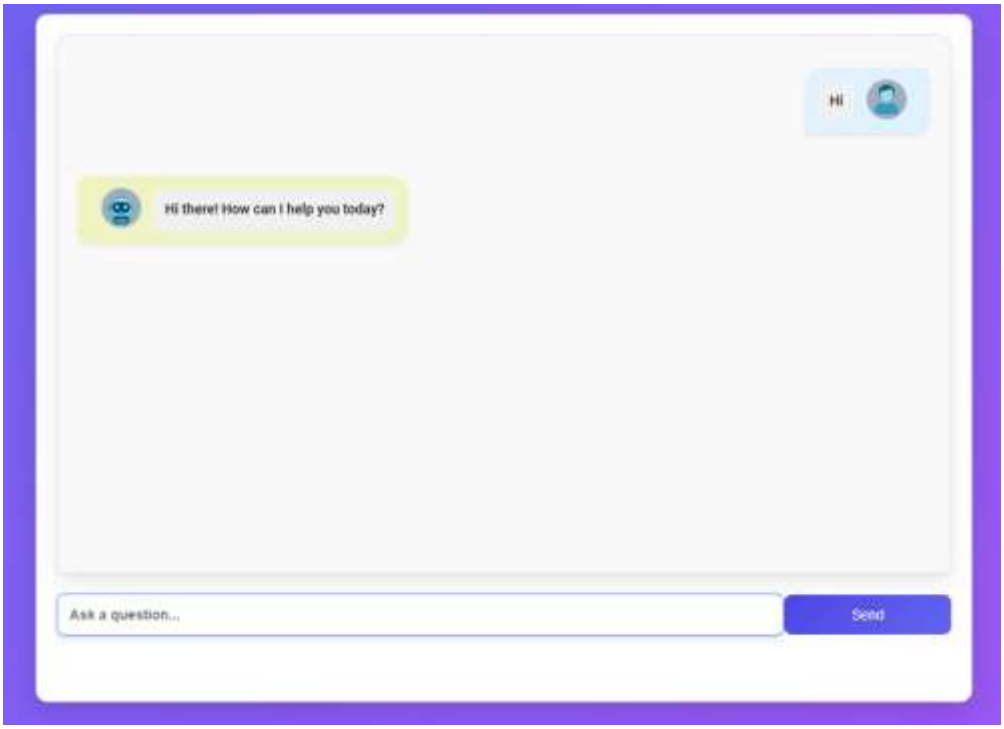


RESULTS AND OUTPUT

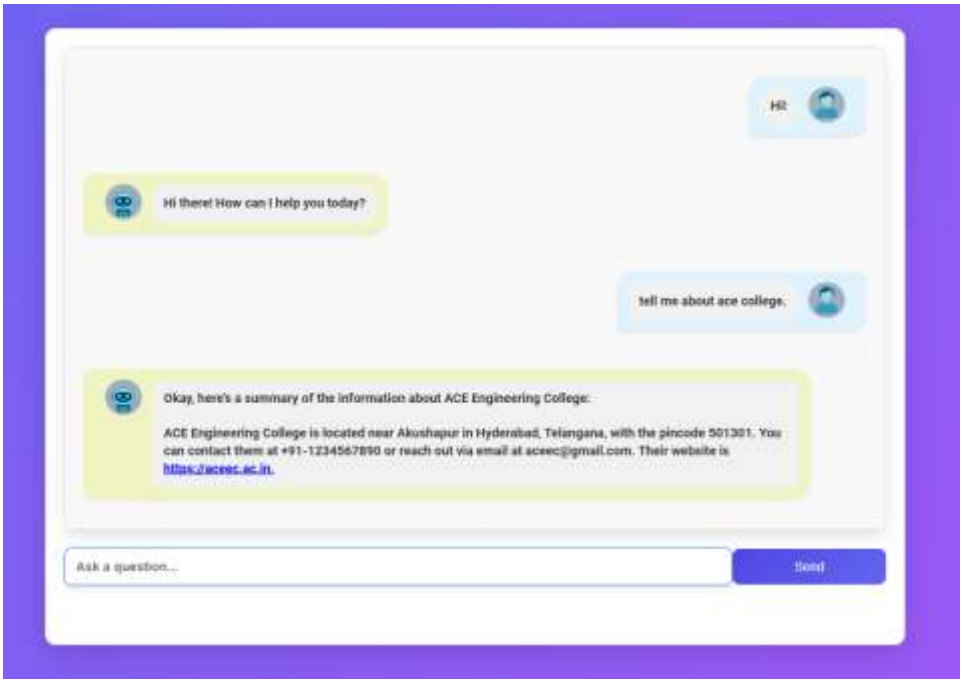
User Interface :



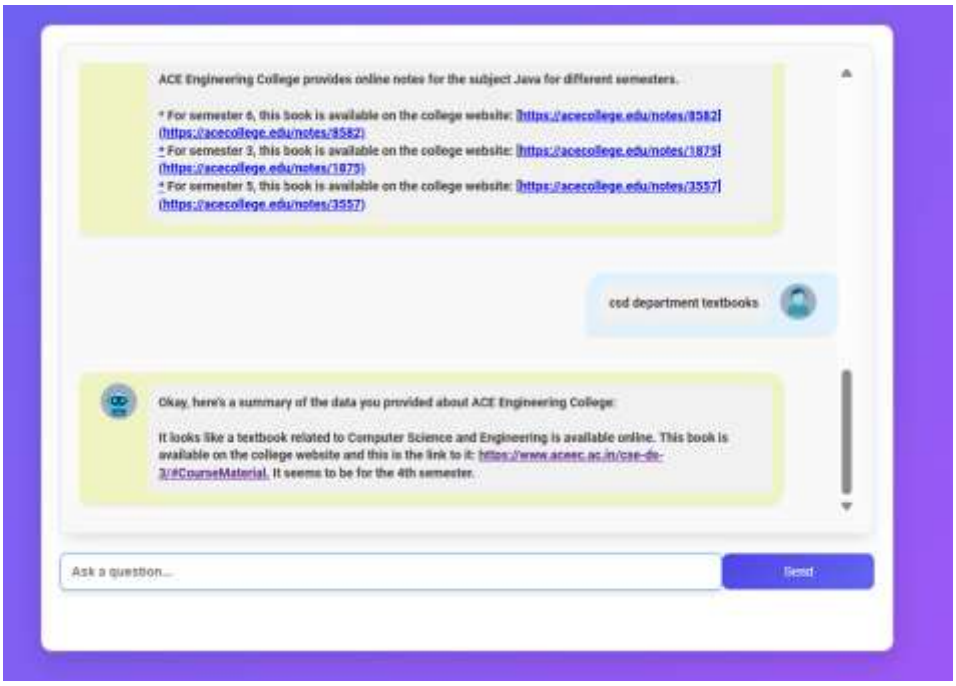
QUERY : Hi!



QUERY: Tell me about ACE ENGINEERING COLLEGE



QUERY: CSD Department Textbooks



Output: Redirect link for CSD Department textbooks

QUERY: Write The Factorial Function in Python

CONCLUSION

The development of a College Enquiry Chatbot using Django, MongoDB, and Google Gemini presents a modern solution for streamlining student and parent communication with college administration. By automating responses to frequently asked questions, the chatbot reduces the workload on staff while providing instant and accurate information to users. It supports both college-related queries through database integration and general/non-college questions using Gemini's advanced language capabilities.

The system architecture leverages Google Gemini to convert user queries into MongoDB queries for relevant data retrieval. This approach ensures that even complex and dynamic queries are understood and processed effectively. Additionally, the chatbot enhances the user experience by delivering data in user-friendly formats such as text, tables, and visual charts.

Though effective, challenges remain in ensuring up-to-date data in the database, maintaining query accuracy for varied input types, and enhancing multilingual support. Overall, this chatbot acts as a smart, scalable solution for educational institutions aiming to provide tech-driven communication.

FUTURE SCOPE

Voice Assistant Integration:

Adding speech-to-text and voice response functionality will make the chatbot more accessible, especially for visually impaired users or those more comfortable with voice-based interaction.

Mobile Application Version:

Developing an Android/iOS version using frameworks like Flutter can help students access the chatbot easily on mobile devices, increasing usability and engagement.

Real-Time Admin Panel for Data Update:

Introducing a dashboard for admin/faculty to update timetables, faculty info, and documents directly into MongoDB will ensure that the chatbot always returns the latest information.

Student Feedback & Analytics Module:

Collecting user feedback and tracking query trends can help improve the bot's performance and guide future improvements. It also helps college management understand the common concerns or interests of students.

Multilingual Support with NLP:

Adding support for Telugu, Hindi, or other regional languages can make the bot more inclusive for users who are not fluent in English, leveraging advanced NLP models.

REFERENCES

Websites:

- **ACE Engineering College official website – *https://www.aceec.ac.in/***
- **Django documentation – *https://docs.djangoproject.com/***
- **MongoDB documentation – *https://www.mongodb.com/docs/***
- **Google Gemini API – *https://ai.google.dev/***

Tools/Libraries:

- **Django**
- **MongoDB**
- **Gemini API**
- **Python**
- **HTML/CSS (for frontend)**
- **GitHub**

Paper:

International Journal Of Creative Research Thoughts (IJCRT)