# Automated Helpdesk Support for Student Admission Department

# 1. Introduction

The Student Admission process involves multiple repetitive queries from prospective students regarding eligibility, documentation, deadlines, and procedures. Automating these queries through a chatbot provides a faster, more efficient experience. This project demonstrates a GenAI-powered helpdesk that can answer student queries using a simple web-based interface.

# 2. Objective

To develop an intelligent chatbot using Generative AI that can automate responses to frequently asked questions related to student admissions, thus reducing manual workload and enhancing student experience.

# 3. Technologies Used

- Python  
- Streamlit (Frontend)  
- Gemini API (Mocked for demo)  
- VS Code (Development Environment)

# 4. System Architecture

The system is built using a modular approach with two main components: a frontend (user interface) built with Streamlit, and a backend written in Python that handles the logic and integrates with the Gemini API (or mock responses). This ensures scalability and ease of integration with future systems.

# 5. Implementation

## *5.1 Backend Logic (chatbot\_backend.py)*

This file contains the function `get\_ai\_response()` which takes user queries as input and returns a mock AI-generated response. This is helpful for testing and demonstration without requiring an actual API key or billing setup.

## *5.2 Frontend Interface (app.py)*

Built using Streamlit, the app allows users to enter queries and receive answers. It has input handling, validation, and a loading spinner for a better user experience.

## 6. Flow Chart Diagram:

* ***User Agent***: Initiates queries through the Streamlit interface, acting as the input source for the system.
* ***Frontend Agent (Streamlit App)****:* Captures user input, triggers backend calls, and displays AI-generated responses, ensuring user interaction is smooth.
* ***Backend Agent (Python Script)***: Processes the query and communicates with the AI model (OpenAI API or mock function), ensuring appropriate responses are generated.
* ***AI Agent (Gemini AI or Mock Response)****:* Simulates or provides intelligent answers based on the query content, forming the core of the helpdesk logic.
* ***Synchronization****:* Streamlit handles real-time communication between frontend and backend, ensuring that user queries are sent, processed, and responses returned without delay or conflict.

## 6. Flow Chart Diagram:

## 

# 7. Code Example

## *6.1 Backend Code*

import google.generativeai as genai

genai.configure(api\_key="Give your API key")

model = genai.GenerativeModel('gemini-1.5-pro')

chat\_session = model.start\_chat()

def get\_ai\_response(user\_query):

    """

    Function to get a response from the Gemini model using chat session

    """

    try:

        response = chat\_session.send\_message(user\_query)

        return response.text

    except Exception as e:

        return f"Error occurred: {str(e)}"

if \_\_name\_\_ == "\_\_main\_\_":

    sample\_query = "What are the admission requirements for your university?"

    print(f"User query: {sample\_query}")

    response = get\_ai\_response(sample\_query)

    print(f"Gemini Response: {response}")

## *6.2 Frontend Code*

# app.py

import streamlit as st

from chatbot\_backend import get\_ai\_response

st.set\_page\_config(page\_title="Student Admission Helpdesk", layout="centered")

st.title("🎓 Student Admission Helpdesk")

st.markdown("Ask any question related to admissions!")

user\_query = st.text\_input("Enter your query here:")

if st.button("Get Response"):

    if user\_query:

        with st.spinner("Thinking..."):

            response = get\_ai\_response(user\_query)

        st.success("Response:")

        st.write(response)

    else:

        st.warning("Please enter a question.")

# 8. Conclusion

This project demonstrates a simple yet effective application of Generative AI in educational institutions. By automating common student queries, institutions can ensure better engagement and quicker support without increasing manpower.

# 9. Research Papers Aligning with the Project

* [https://arxiv.org/abs/2303.12712](https://arxiv.org/abs/2303.12712%20) (ChatGPT and Education)
* <https://arxiv.org/abs/2001.09977> (Conversational AI systems overview)
* <https://arxiv.org/abs/2206.14435> (Real-World QA systems using LLMs)

# 10. Acknowledgement

I would like to express my heartfelt gratitude to everyone who supported me in the successful completion of this project. Special thanks to my teachers, and my team, for their invaluable guidance, constructive feedback, and encouragement throughout the development of the **Automated helpdesk support for Student Admission Department**.

Their insights and expertise were instrumental in overcoming challenges and achieving the objectives of this work. I am also thankful for the resources and tools that made the implementation of this project possible.Thank you all for your unwavering support and belief in my efforts.

## *Our Team:*

* Semanti Dutta (CSE , 12022002002193)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* Srijita Mitra (CSE , 12022002002194)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* Panchajanya Paul (CSE , 12022002002195)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_