**Project Report: Campus Query Chatbot**

**Candidate: Vibhav Anand**

**Project: AI-Powered Campus Query Chatbot**

**Duration: 2 Days**

**Tools & Technologies: Python, Streamlit, Groq API, JSON, Requests**

**1. Introduction**

The Campus Query Chatbot is an AI-powered assistant designed to answer common student queries related to campus facilities, academic schedules, hostel accommodations, Wi-Fi access, and other university-related information. It simplifies access to information for students by providing instant, conversational responses.

**2. Objective**

To build a chatbot that uses a custom FAQ dataset combined with Groq’s AI language model to deliver relevant and context-aware answers to user questions via an easy-to-use Streamlit web interface.

**3. Key Features**

* User-friendly web interface built with Streamlit
* Custom FAQ dataset with typical campus-related questions and answers
* Fuzzy matching to find closest relevant FAQ question
* Integration with Groq’s AI API for enhanced context-aware responses
* Displays helpful messages when questions are not found or unclear
* Easily extendable with more FAQs or additional features

**4. Tools and Technologies Used**

* **Python:** Main programming language
* **Streamlit:** For rapid development of the web UI
* **Groq API:** For AI-powered response generation
* **Requests:** HTTP library for API calls
* **JSON:** To store and manage FAQ dataset
* **Difflib:** To perform fuzzy string matching for user questions

**5. Dataset**

The FAQ dataset is a JSON file containing key-value pairs of commonly asked campus questions and their respective answers. It covers topics such as library hours, semester start dates, course registration, hostel accommodation, placement cell contacts, and more.

Example:

json

CopyEdit

{

"What are the library hours?": "The library is open from 8 AM to 8 PM on weekdays.",

"When does the semester start?": "The semester begins on July 10, 2025."

}

**6. Workflow and Architecture**

1. User inputs a question in the Streamlit interface.
2. The chatbot uses fuzzy matching (difflib.get\_close\_matches) to find the best matching FAQ question.
3. If a match is found, the question and answer context are sent to the Groq API.
4. Groq’s AI model generates a refined and natural language response.
5. The chatbot displays the AI-generated answer to the user.
6. If no match is found, the chatbot politely informs the user to rephrase the question.

**7. Challenges Faced**

* Ensuring the correct Groq API endpoint and proper API key authentication.
* Managing environment variables for sensitive API keys.
* Designing fuzzy matching thresholds to balance recall and precision of FAQ retrieval.
* Handling cases where no relevant FAQ matches user queries.

**8. Future Enhancements**

* Expand FAQ dataset with more questions and categories.
* Add NLP-based intent recognition to improve matching accuracy.
* Implement user feedback collection to improve answers over time.
* Deploy on cloud platforms like Render or Heroku for broader access.
* Integrate voice input and output for accessibility.

**9. Conclusion**

This project successfully demonstrates the development of a simple yet effective AI chatbot for campus queries by combining rule-based matching with powerful AI language understanding through Groq. It enhances student experience by providing quick access to vital campus information and serves as a foundation for more complex conversational AI applications.