CASE STUDY SUBMISSION
(Generative AI & Large Language Models – UG Winning Camp)

1. Cover Page

Title of the Case Study: GenAI-Powered Smart Campus Assistant

Problem Statement Title: GenAI-Powered Smart Campus Assistant

Team Name: Team Innovators

Team Members (Name, Roll No., Branch, Year):

- Arnav Mishra 23BAI70585- AIT CSE(3rd year)
- Adarsh Deep 23BAI70047- AIT CSE(3rd year)
- Sikander 23BAI70600- AIT CSE(3rd year)

Institute Name: CHANDIGARH UNIVERSITY
Mentor / Faculty Coordinator: Dr. Prabhjot Kaur
Date of Submission: 15-12-2025

2. Abstract (150–200 words)

Modern educational campuses generate vast amounts of information related to academics, administration, events, and student services. However, students often struggle to access the right information quickly through traditional websites or helpdesks. This case study proposes a **GenAI-Powered Smart Campus Assistant** that leverages Large Language Models (LLMs) to provide instant, accurate, and context-aware responses to student queries. The assistant understands natural language inputs, retrieves relevant institutional data, and generates concise answers while supporting multi-turn conversations. Unlike rule-based chatbots, the proposed system demonstrates true Generative AI capabilities through reasoning, contextual understanding, and adaptive responses. The solution is designed to operate ethically by incorporating responsible AI practices such as hallucination control, data grounding, and privacy preservation. The Smart Campus Assistant aims to improve student experience, reduce administrative workload, and enable scalable information access across the campus. This case study outlines the problem motivation, objectives, proposed solution, GenAI methodology, system architecture, and future enhancements, highlighting how Generative AI can transform digital campus support systems.

3. Problem Understanding & Motivation

Students frequently ask repetitive questions regarding class schedules, exam dates, attendance rules, fee deadlines, events, and campus facilities. Existing systems such as static websites or rule-based chatbots are difficult to navigate and fail to handle follow-up questions. The motivation behind this project is to create an intelligent assistant that can converse naturally with students, understand intent, and provide accurate institutional information in real time. The target users include undergraduate students, new admissions, and campus visitors.

4. Objectives

- To build a conversational AI system capable of understanding natural language queries
- To provide accurate, concise, and context-aware campus-related information
- To support multi-turn conversations using Generative AI principles

5. Proposed Solution

The proposed solution is a **Smart Campus Assistant** powered by a Large Language Model integrated with institutional data sources. Users interact with the system via text-based chat. The assistant interprets queries, retrieves relevant information from a knowledge base, and generates human-like responses. It supports follow-up questions, summarizes lengthy information, and adapts responses based on user context.

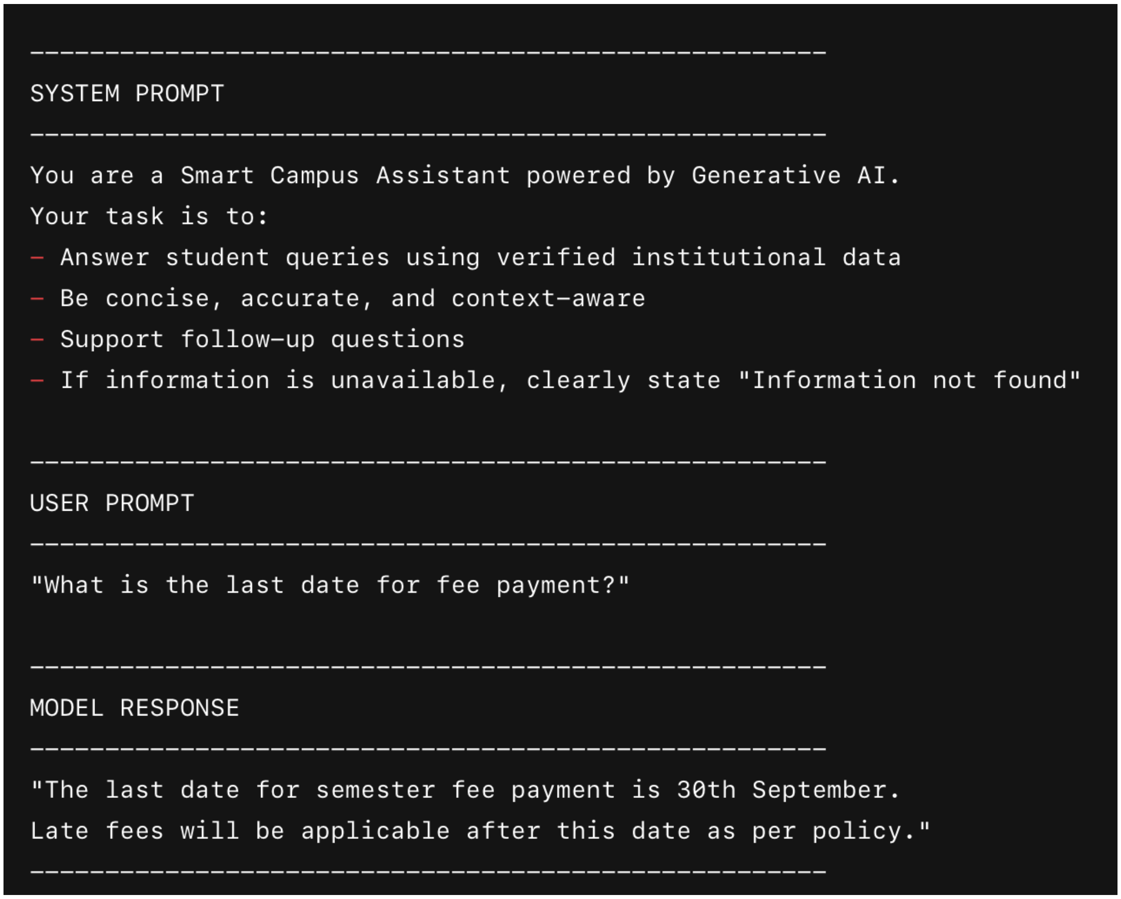
6. GenAI & LLM Methodology

LLM Used: GPT-based Large Language Model
Prompt Engineering Strategy: System prompts defining assistant role, tone, and boundaries; user prompts structured for clarity
Retrieval / Grounding Approach: Retrieval-Augmented Generation (RAG) using campus documents, FAQs, and databases

7. System Architecture

The system consists of a user interface, backend server, LLM API, and institutional knowledge base. User queries are sent to the backend, relevant data is retrieved, and the LLM generates grounded responses. This architecture ensures accuracy and scalability.

8. Prompt Design

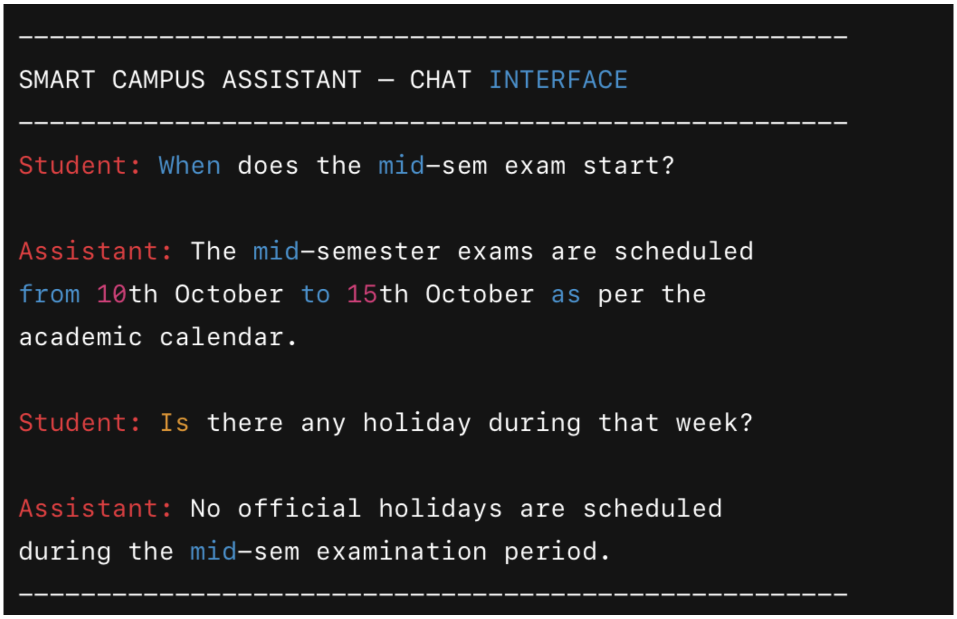


```
-----------------------------------------------------
SYSTEM PROMPT
-----------------------------------------------------
You are a Smart Campus Assistant powered by Generative AI.
Your task is to:
- Answer student queries using verified institutional data
- Be concise, accurate, and context-aware
- Support follow-up questions
- If information is unavailable, clearly state "Information not found"

-----------------------------------------------------
USER PROMPT
-----------------------------------------------------
"What is the last date for fee payment?"

-----------------------------------------------------
MODEL RESPONSE
-----------------------------------------------------
"The last date for semester fee payment is 30th September.
Late fees will be applicable after this date as per policy."
-----------------------------------------------------
```

9. Prototype / Implementation Details

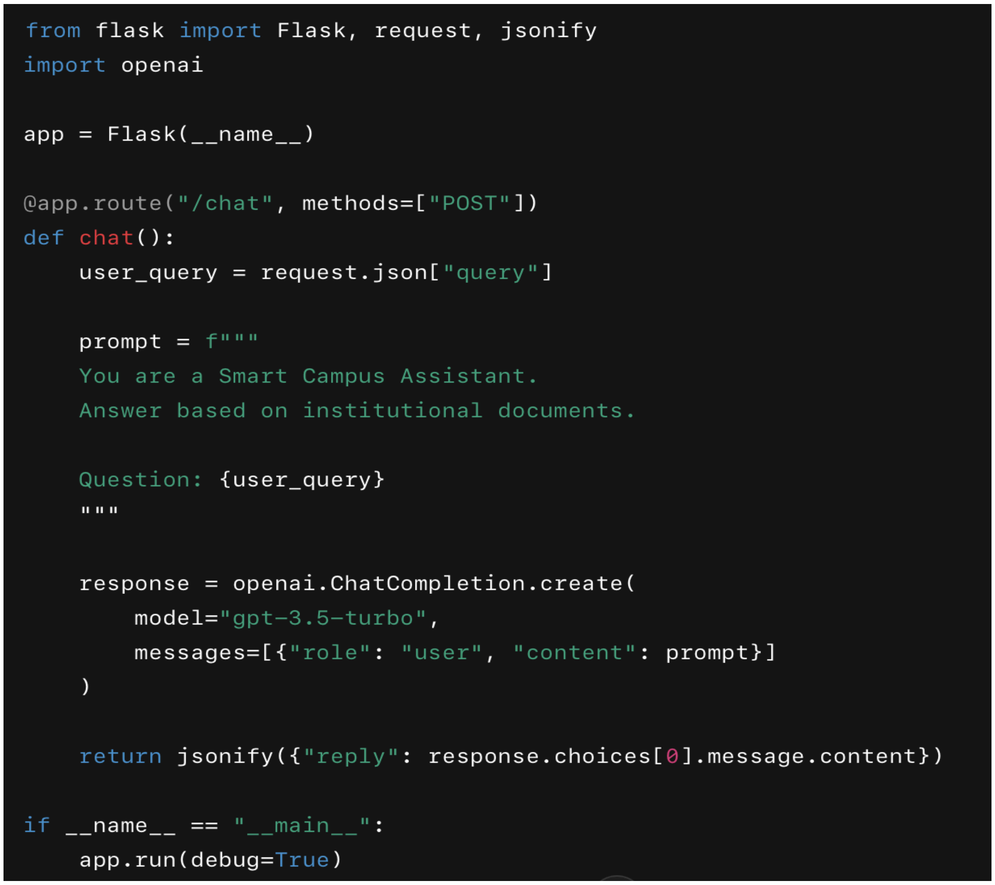
Tools & Technologies: Python, Flask, OpenAI API, Vector Database
Dataset / Documents Used: Academic calendar, campus FAQs, policy documents

Prototype Description:



```
----------------------------------------------------
SMART CAMPUS ASSISTANT — CHAT INTERFACE
----------------------------------------------------
Student: When does the mid-sem exam start?

Assistant: The mid-semester exams are scheduled
from 10th October to 15th October as per the
academic calendar.

Student: Is there any holiday during that week?

Assistant: No official holidays are scheduled
during the mid-sem examination period.
----------------------------------------------------
```

Backend Prototype Code Snippet (Flask + LLM)



```python
from flask import Flask, request, jsonify
import openai

app = Flask(__name__)

@app.route("/chat", methods=["POST"])
def chat():
    user_query = request.json["query"]

    prompt = f"""
    You are a Smart Campus Assistant.
    Answer based on institutional documents.

    Question: {user_query}
    """

    response = openai.ChatCompletion.create(
        model="gpt-3.5-turbo",
        messages=[{"role": "user", "content": prompt}]
    )

    return jsonify({"reply": response.choices[0].message.content})

if __name__ == "__main__":
    app.run(debug=True)
```

10. Sample Outputs

```
INPUT:
"Where can I download my attendance report?"

OUTPUT:
"You can download your attendance report
from the student portal under the 'Academics' section."

EXPLANATION:
The response is generated by retrieving attendance
information from the campus knowledge base.
```

11. Ethical Considerations & Responsible AI

The system minimizes hallucinations through data grounding, avoids biased language, protects student privacy, and includes disclaimers for uncertain responses.

12. Results & Impact

The Smart Campus Assistant improves information accessibility, reduces administrative workload, and enhances student satisfaction. It can scale across departments and institutions.

13. Limitations

- Dependence on data quality
- Limited support for voice-based interaction

14. Future Enhancements

- Voice assistant integration
- Multilingual support
- Mobile application deployment

15. Conclusion

This case study demonstrates how Generative AI can be effectively applied to build an intelligent, ethical, and scalable Smart Campus Assistant that enhances the overall campus experience.

16. References

- OpenAI Documentation
- Research papers on Conversational AI
- Institutional policy documents

Declaration

We hereby declare that this submission is our original work and all references have been cited.

Team Leader Signature: _____________
Date: _____________