

# Intelligent Assistant — Prompt Engineering, API Integration & Hallucination Management

## Project Overview

The aim of this assignment is to design and implement a conversational assistant that leverages a Large Language Model (LLM) to deliver coherent, engaging, and context-aware interactions. The assistant's domain and purpose are entirely up to you (e.g., productivity, tutoring, information lookup, entertainment, sports, etc.). A core requirement is the enrichment of responses through seamless integration with at least two external APIs.

## Core Requirements

### Conversation-Oriented Design

- **Assistant Purpose:**  
Design an assistant in any domain of your choice. It should handle at least three different types of queries or tasks relevant to its intended purpose.
- **Context & Continuity:**  
The assistant must track and manage conversational context, including follow-up and clarifying questions, for natural multi-turn interactions.
- **Interaction Flow:**  
Ensure conversation feels natural and user-friendly, prioritizing clarity, helpfulness, and accuracy in each exchange.

### Advanced Prompt Engineering

- **Prompt Crafting:**  
Demonstrate thoughtful prompt design to elicit accurate, concise, and context-relevant answers from the LLM.
- **Multi-Step Reasoning:**  
Include at least one prompt or structured method that guides the LLM through a chain-of-thought process.
- **Control Strategies:**  
Implement prompt techniques or instructions that help reduce the likelihood of hallucinations or off-topic responses.

### Technical Implementation

- **Programming Language:**  
Free choice of language.
- **LLM Model:**  
Integrate with any LLM API or mode
- **Interface:**  
A CLI is required; a web UI is optional but encouraged for enhanced usability.

## External Data Integration

- **API Connections:**  
The assistant must use at least two external APIs (your choice).
- **Data Fusion:**  
Carefully combine external API data with LLM knowledge in your responses, ensuring the data presented is accurate and reliable.
- **Decision Logic:**  
Develop a clear strategy for deciding when to use external data versus relying on the LLM, prioritizing accuracy and minimizing hallucinations

## Hallucination Detection & Management

- Include methods or heuristics to detect potential hallucinations and misinformation in the assistant's responses.

## Evaluation Criteria

1. **Conversation Quality:**  
Are interactions helpful, logical, accurate, and engaging?
2. **Hallucination Handling:**  
How well does the system detect, mitigate, and recover from hallucinations or LLM mistakes? Is the approach robust and user-friendly?
3. **Context Management:**  
Is conversational history followed and context maintained throughout multi-turn exchanges?
4. **External Data Integration:**  
Are your chosen APIs meaningfully and accurately blended with LLM outputs?

## Submission Guidelines

- **Source Code:**  
Submit all code on GitHub.
- **Conversation Transcripts:**  
Provide sample transcripts/screen captures that demonstrate system functionality.

## Additional Notes

This assignment is above all a demonstration of your ability to design effective, reliable, and trustworthy LLM-powered conversational systems. Highlight not only your technical skills but also your judgment in managing the limitations and risks of generative models. Prioritize creativity, conversation quality, user trust, and clarity.