Travel Assistant

Project Focus

Build a travel assistant that demonstrates your ability to create natural, effective conversations with LLMs. The technical implementation should be simple, allowing you to focus primarily on prompt engineering and conversational quality.

Core Requirements

Conversation-First Design

- Create a travel planning assistant that can handle at least 3 different types of travel queries (e.g., destination recommendations, packing suggestions, local attractions)
- Design the system to handle follow-up questions and maintain context across the conversation
- Implement a conversation flow that feels natural and helpful to users

Enhanced Prompt Engineering

- Design your system prompts to extract maximum value from the LLM
- Include at least one "chain of thought" prompt that guides the LLM through a multi-step reasoning process
- Implement prompt techniques that help the LLM provide concise, relevant responses

Simple Technical Implementation

- Use any programming language you're comfortable with
- Integrate with a free LLM API (Ollama, DeepSeek, etc.)
- Create a simple interface (can be CLI, basic web UI, or API endpoints) that allows testing the conversation

Data Augmentation

- Integrate with at least one external API (weather, country info, etc.)
- Design prompts that effectively blend external data with LLM knowledge
- Implement a decision method for when to use external data vs. LLM knowledge

What to Prioritize

1. Conversation Quality: Focus on making the interactions feel natural and helpful
2. Prompt Design: Show your skill in crafting effective prompts that get the best results
3. Error Handling: Demonstrate how you recover from confused responses or hallucinations
4. Context Management: Show how you maintain conversation history and context

Evaluation Focus

- How well the assistant maintains coherent, helpful conversations

- Quality and creativity of your prompt engineering approach
- How you handle edge cases and potential LLM limitations
- Your approach to blending external data with LLM capabilities

Submission

Submit your code via GitHub or zip file, including:

- All source code
- Sample conversation transcripts showing your system in action
- Brief notes on key prompt engineering decisions

This assignment is designed to showcase your ability to think about and implement effective LLM conversations, rather than focusing on documentation or complex backend systems.