

Intelligent Travel Assistant AI – Task Report

1. Task Overview

This task focuses on developing an **AI-powered Travel Assistant** using LangChain, integrated with:

- **Google Gemini** for reasoning and response generation
- **WeatherAPI** for live weather updates
- **DuckDuckGo Search** for tourist attraction discovery

The assistant accepts a destination from the user and provides:

1. Real-time weather information
2. Top local tourist attractions

The core system uses **LangChain's agent architecture** to dynamically decide which tools to invoke based on user queries.

2. LLM Reasoning and Agent Design

Purpose of the LLM (Gemini)

The LLM is responsible for:

- Understanding user prompts
- Reasoning about which tools to invoke (weather or search)
- Summarizing tool results into coherent responses

How Reasoning Works

- A structured prompt guides the LLM to act as a travel assistant.

- LangChain's `create_tool_calling_agent` binds the LLM to the tools.
- As the LLM processes the input, it determines whether to fetch weather data, attraction information, or both.
- Tool responses are recorded in an **agent scratchpad**, which allows the LLM to maintain memory of what has been done and build a final response based on the complete context.

This looped interaction allows the LLM to chain thoughts and actions, effectively using a "Reasoning + Acting" (ReAct) pattern.

3. System Components and Flow

3.1 Tools

Weather Tool

A custom function fetches weather data from **WeatherAPI** using HTTP requests. The tool accepts a city name and returns the current temperature and condition (e.g., clear, cloudy).

Tourist Attractions Tool

This tool uses the **DuckDuckGo Search API** to gather information about popular attractions in a given city. Search results are parsed and summarized into concise bullet points for the user.

3.2 Agent Setup

The agent is constructed using LangChain's `create_tool_calling_agent` method, which combines:

- A **prompt template** including `agent_scratchpad` for iterative reasoning
- The **Gemini model** as the LLM
- The **weather and attraction tools**

This setup enables the LLM to plan, call tools, remember intermediate steps, and finally summarize outputs into a complete response.

3.3 Execution Flow

1. The user provides a destination (e.g., “Tokyo”).
 2. A structured query is built to request both weather and tourist data.
 3. The agent:
 - Decides which tools are needed
 - Executes each tool independently
 - Stores results in a scratchpad
 4. The LLM summarizes both results into one human-readable response.
 5. The final result is printed for the user.
-

4. Final Output Example

For input:

“Tell me the weather and top places to visit in Paris.”

The assistant responds with:

- Current temperature and weather conditions in Paris
 - A summarized list of top attractions, such as the Eiffel Tower, Louvre Museum, and Notre-Dame Cathedral
-

5. Technologies Used

Component	Technology
Language Model	Google Gemini via LangChain
Weather Data	WeatherAPI.com
Tourist Information	DuckDuckGo Search
Agent Framework	LangChain Tool-Calling Agent

6. Summary

This Travel Assistant demonstrates how **large language models** can be enhanced through **external tool integration** to provide context-aware, real-time answers. The use of LangChain agents enables structured reasoning and dynamic tool orchestration, making the assistant versatile and scalable for real-world use cases.