

AI Travel Assistant

Upgrad
Prepared by:
Priyanuj Misra

Content

1. Understanding System Design
2. Technical Implementation
3. Challenges Faced and
Recommendations

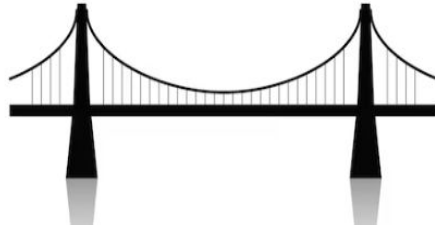
Intelligent Travel Planner using Generative AI

Prompt Templates

- Initialized initiation and travel recommendation prompt templates to define the assistant's role and personality
- Set tone and task expectations clearly
- Guide the app through step by step reasoning

API Integration

- Weather API - To get weather forecast
- Flight API - To get flight timings to recommend based on user requirement



Dialogue Management System

- Conversation Initialization
- User input loop with Moderation Check, LLM Call, Intent Confirmation, API triggers
- Travel Itinerary and Flight Suggestion Generation

Function Calling

- LLM Chat Completion with `function_call = "auto"`
- Functions registered with function description
- LLM auto-decides when to call a tool

Technical Implementation

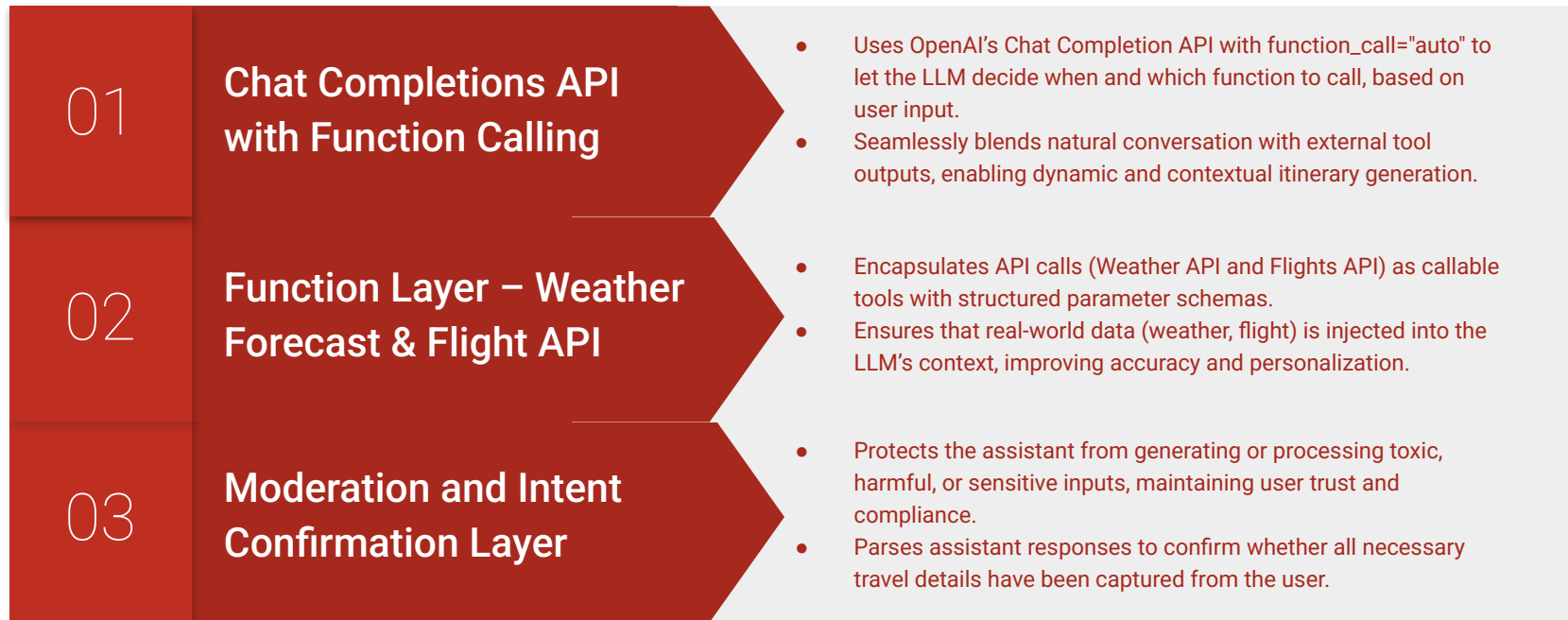
Technical Steps Involved



Key Steps

1. Chat Completions API with function calling
2. Function Layer - Weather Forecast and Flight API
3. Moderation Layer
4. Intent Confirmation Layer

Understanding major technical processes



Challenges Faced and Future Scope

Challenges Faced and Future Scope of Work

Challenges

1. OpenAI SDK Version Incompatibility: Needed to downgrade to openai==0.28 to support function calling, which led to breaking changes in moderation and chat completion code (e.g., `openai.moderations.create()` not available).
2. Limited Weather API Forecast Window: Weather API only supports forecasts for the next 5 days, requiring logic to gracefully fallback or inform users when their travel dates fall outside this window.
3. Flight API has only 20 Free Trials therefore had to limit the code run for that function to eventually have the answers in the final output
4. Handling API Failures and Rate Limits: External APIs like FlightAPI and WeatherAPI occasionally return 401/403 errors, needing retry logic, validation layers, and error-safe fallback messages for a smooth user experience.

Future Course of Action

1. Integration of Hotel Search APIs
2. Local Experience and Restaurant Recommendations
3. Frontend Development - (Fast API or Flask/ Streamlit) - Web or App
4. Downloadable itinerary via PDF or Email

Thank
You