

# Project Report: Travel Planner AI

## 1. Introduction

### 1.1 Overview

The **Travel Planner AI** project is a generative AI-powered chatbot designed to help travellers efficiently plan their trips. It provides intelligent travel recommendations such as **top 3 hotel suggestions** and **personalized itineraries** based on user-specified destinations.

The system integrates **Generative AI**, **Function Calling APIs**, and **CSV-based data lookup** to produce context-aware, conversational responses. It also includes **content moderation and safety mechanisms** to handle inappropriate user inputs gracefully.

### 1.2 Motivation

Travel planning often requires browsing multiple platforms for accommodations, destinations, and itineraries. This project aims to simplify that experience by creating an **AI travel assistant** that provides all essential recommendations through a single conversational interface.

---

## 2. Project Objectives

1. **Develop an intelligent chatbot** that assists users in planning their travel using natural conversation.
  2. **Recommend top 3 hotels** from CSV datasets based on user-provided destinations.
  3. **Suggest daily itineraries** aligned with user preferences (location, duration, budget).
  4. **Implement Function Calling API** for structured data retrieval and dynamic responses.
  5. **Integrate safety mechanisms** such as content moderation to terminate harmful conversations.
  6. **Ensure smooth conversational flow** and consistent output formatting.
  7. **Provide an end-to-end system design**, implementation, and evaluation.
-

### 3. System Design

#### 3.1 Architecture Overview

The architecture consists of five primary components:

Component	Description
User Interface	Chat interface (CLI or web) through which the user interacts with the AI.
LLM Engine	Handles user inputs, performs reasoning, and generates responses via OpenAI’s Function Calling API.
Function Handler Layer	Interprets LLM requests, fetches CSV data, and executes defined actions (e.g., hotel lookup).
Data Source (CSV)	Stores structured travel data — destinations, hotel names, ratings, prices, and itinerary templates.
Moderation Layer	Scans inputs for unsafe or harmful content; ends the session if flagged.

#### 3.2 Workflow Diagram

User → LLM → Function Calling API → Function Handler → CSV Lookup

    ↳ Moderation Check → (Safe?) → Response / Session Termination

#### 3.3 Data Flow

1. User enters destination query (e.g., “Plan my trip to Goa”).
  2. LLM interprets request .
  3. Function retrieves top 3 hotels from CSV file.
  4. AI generates day-by-day itinerary.
  5. Response formatted with hotel list + itinerary + travel tips.
  6. Moderation ensures no policy-violating input.
-

## 4. Implementation

### 4.1 Technologies Used

- **Python 3.10**
- **OpenAI API (Function Calling & Moderation)**
- **Pandas** for CSV data handling

### 4.2 Key Functions

- **CSV-based Data Source**  
The chatbot reads directly from the CSV file, making it modular and scalable for real datasets.
- **Top Hotel Recommendations**  
Retrieves the top 3 hotels for each destination based on rating and pricing.
- **Dynamic Itinerary Generation**  
Suggests 3-day itineraries from attractions sorted by popularity.
- **LLM-driven Response Formatting**  
Combines data-driven results with natural language responses via `compose_response_llm()`.
- **Safety & Moderation**  
Any harmful query (e.g., violent or illegal content) triggers session termination.
- **Error Handling**  
Handles unknown destinations gracefully by responding:  
*"No itinerary found as per your requirement."*

### 4.3 Sample CSV Structure

Destination Hotel Name		Rating	Price	Description
Goa	Beach Paradise Resort	4.7	9000	Seaside resort with ocean view
Goa	Coconut Grove Inn	4.5	7500	Family-friendly resort near beach
Goa	Urban Retreat	4.3	8000	Modern stay close to nightlife

---

## 5. Challenges and Solutions

Challenge	Solution
Handling irrelevant or malicious input	Integrated OpenAI moderation to end session when flagged
Maintaining output consistency	Defined strict response schema in function calling
Limited data diversity	Used public hotel datasets and sample CSV augmentation
Conversational context loss	Implemented state memory to preserve preferences
Model hallucination	Restricted model to retrieve from CSV-based factual data

---

## 6. Results

### 6.1 Example Interaction

**User:** “Plan a trip to Goa for 3 days.”

**AI:**

**Top 3 Hotels:**

1. Beach Paradise Resort (4.7★) – Seaside view, ₹9000/night
2. Coconut Grove Inn (4.5★) – Family-friendly, ₹7500/night
3. Urban Retreat (4.3★) – Near nightlife, ₹8000/night

**Suggested Itinerary:**

- **Day 1:** Explore beaches (Baga, Calangute) and try Goan cuisine
  - **Day 2:** Visit Fort Aguada, Dolphin Cruise
  - **Day 3:** Shopping at Anjuna Flea Market, evening at beach shack
-

## 7. Lessons Learned

1. **Function Calling simplifies tool integration** — structured and reliable.
  2. **Moderation APIs are essential** for user safety in open-domain chatbots.
  3. **Data quality directly impacts recommendation quality.**
  4. **Prompt engineering** improves accuracy and reduces irrelevant answers.
  5. Building conversational memory enhances personalization significantly.
- 

## 8. Future Enhancements

- Integrate **real-time hotel APIs** (e.g., Booking.com, TripAdvisor).
  - Add **multi-language support** for global travellers.
  - Include **flight and weather data integration.**
  - Deploy as a **web-based GenAI travel assistant** using Streamlit or React frontend.
- 

## 9. Conclusion

The **Travel Planner AI** successfully demonstrates how generative AI can transform travel planning into a seamless, interactive experience. The system effectively combines LLM capabilities with structured data, moderation, and clear design principles — ensuring both usability and safety.