BUILD YOUR OWN PROJECT: TRAVEL PLANNER AI

1. Project Overview

The **Travel Planner AI** is a multi-stage, AI-assisted chatbot designed to help users plan trips by recommending suitable hotels and tourist attractions based on their preferences.
The system follows a **structured GenAI architecture**, inspired by the ShopAssist AI reference, to ensure reliable, explainable, and user-centric recommendations.

Instead of relying solely on a language model, the chatbot combines:

- **LLM-based intent understanding and response generation**
- **Rule-based data processing and validation**
- **Publicly available datasets (Kaggle)** as the source of truth

This hybrid approach addresses real-world challenges such as hallucination, inconsistent recommendations, and lack of explainability.

2. System Design

2.1 Overall Architecture

The system is designed as a **three-stage pipeline**:

1. **Stage 1 – Intent Clarity & Confirmation**
2. **Stage 2 – Data Mapping, Scoring & Validation**
3. **Stage 3 – Recommendation & Itinerary Generation**

Each stage has a **clear responsibility**, ensuring separation of concerns and modularity.

User Input

↓

Stage 1: Intent Understanding (LLM)

↓

Destination Enrichment (City → Country)

↓

Stage 2: Dataset-driven Scoring (Rule-based)

↓

Stage 3: Natural Language Recommendation (LLM / Mock)

2.2 Innovation and Creativity

The innovation in this system lies in:

- **Hybrid GenAI Design**
  The LLM is used only where it adds value (conversation and explanation), while deterministic logic is used for comparisons and scoring.
- **Destination Enrichment Layer**
  A lightweight rule-based enrichment step maps user-provided cities to countries, resolving dataset granularity mismatches without requiring additional datasets or APIs.
- Recommendations are only generated after passing **validation thresholds**, preventing weak or irrelevant suggestions.
- **Explainable Decision Making**
  Every hotel and attraction is scored using transparent rules (budget, rating, popularity, preferences), making the system auditable and explainable.

2.3 Real-World Problem Addressed

In real-world travel planning:

- Users rarely provide complete information upfront
- Data sources differ in granularity (city vs country)
- LLM-only systems can hallucinate hotels or attractions

This architecture directly addresses these issues by:

- Iteratively collecting intent
- Using real datasets as ground truth
- Separating reasoning from data processing

3. Technical Implementation

3.1 AI Model Usage

The system uses an OpenAI language model for:

- **Stage 1**: Extracting structured user intent from natural language
- **Stage 3**: Generating human-readable recommendations and itineraries

To ensure robustness:

- A **mock LLM mode** is provided when API access or quota is unavailable
- The system architecture remains unchanged between mock and live modes

This design makes the system suitable for both academic evaluation and production deployment.

3.2 Dataset Usage (External Tools)

Two publicly available Kaggle datasets are used:

1. **Hotels Dataset**
   - Hotel name
   - City
   - Country
   - Price
   - Rating
   - Reviews count
   - Amenities
2. **Tourist Attractions Dataset**
   - Destination name (attraction)
   - Country
   - Continent
   - Type
   - Average rating
   - Annual visitors
   - UNESCO site flag

The datasets are loaded using **pandas**, inspected, normalized, and converted into internal feature dictionaries before processing.

3.3 Efficiency of Code

- **Deterministic scoring logic** ensures fast and repeatable execution
- Dataset preprocessing is done once and reused
- Filtering reduces candidate size early (city/country match)
- Modular functions allow easy testing and maintenance

No unnecessary API calls are made during scoring or validation.

3.4 Effective Interaction with External Resources

- Kaggle datasets act as a reliable knowledge base
- Pandas is used for efficient data loading and preprocessing
- OpenAI API (or mock) is used only at defined interaction points

This controlled interaction avoids over-reliance on the LLM and reduces cost and latency.

4. User Experience

4.1 Conversational Flow

The chatbot follows a natural, guided interaction:

1. User expresses travel intent in free text
2. The system asks clarifying questions if needed
3. Once intent is confirmed, recommendations are generated

This mirrors how a human travel agent would interact with a customer.

4.2 Clarity of Responses

The final response includes:

- Clearly listed hotel recommendations with reasons
- Clearly listed attractions with ratings and popularity
- A simple, easy-to-follow multi-day itinerary
- A follow-up prompt for refinement or exit

This ensures responses are **concise, readable, and actionable**

Input:

```
# 1) Stage 1: collect user requirements
user_input = "I want a trip to Beijing, budget 200 per night, prefer free wifi and pool, I like museums and food."
```

Output:

```
=== Recommended Hotels ===
- Shangri-La Hotel, Beijing (rating=5.0, price=156.6666667, score=3)
- Holiday Inn Express Beijing Yizhuang (rating=5.0, price=68.0, score=3)

=== Recommended Attractions ===
- Hidden Ruins (Historical, rating=5.0, visitors=NoneM, score=1)
- Crystal Plaza (Adventure, rating=4.4, visitors=NoneM, score=1)
- Ancient Pagoda (Historical, rating=4.1, visitors=NoneM, score=1)
- Sacred Pagoda (Nature, rating=4.9, visitors=NoneM, score=1)
- Serene Park (Adventure, rating=4.4, visitors=NoneM, score=1)
- Lush Plaza (City, rating=4.9, visitors=NoneM, score=1)

=== Sample 3-Day Itinerary ===
Day 1: Visit Hidden Ruins and explore local culture
Day 2: Explore Crystal Plaza and nearby sites
Day 3: Leisure activities, shopping, and relaxation

Type 'exit' if satisfied, or tell me what you want to refine.
```

4.3 Seamless Interaction

Because each stage has a defined role:

- The user is never overwhelmed with questions
- Recommendations are not shown prematurely
- Refinement is easy and intuitive

5. Documentation

5.1 Design Choices

Key design decisions include:

- Using a **three-stage architecture** for clarity and reliability
- Separating LLM reasoning from rule-based scoring
- Using Kaggle datasets instead of live APIs for reproducibility
- Adding destination enrichment to resolve dataset mismatches

Each decision improves robustness and explainability.

5.2 Challenges Faced

Challenge	Solution
Different dataset granularity	Destination enrichment layer
OpenAI API quota limitations	Mock LLM implementation
LLM hallucination risk	Dataset-driven validation
Inconsistent user input	Intent confirmation layer

5.3 Limitations and Future Enhancements

Current limitations

- City-to-country mapping is rule-based and limited
- No real-time pricing or availability
- No map or distance-based optimization

Future enhancements

- Geocoding APIs for global city-country mapping
- Live hotel and attraction APIs
- Weather- and season-aware itinerary planning
- Multi-city trip support

6. Conclusion

The Travel Planner AI demonstrates a **well-structured, explainable, and scalable GenAI system**.
By combining conversational AI with deterministic data processing, the project avoids common pitfalls of LLM-only solutions and delivers reliable, user-friendly travel recommendations.