TravelPilot: A LLM-Based Agent for Customized Traveling

1 Team Information

Yu Bohan 3036657328 Ao Jiaqi 3036654833

Su Yongchao 3036659352 Chen Zhenyan 3036655526

Yu Haoguang 3036659974

2 Introduction

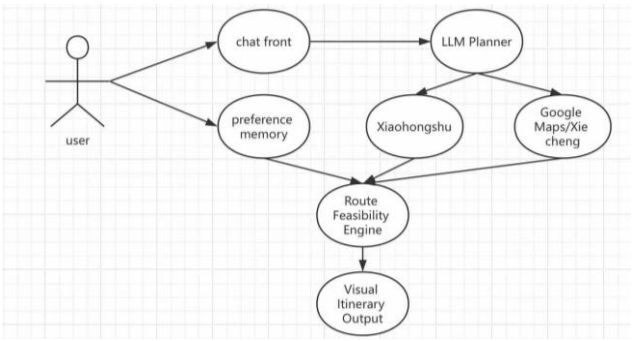
We present TravelPilot, an agentic travel planning system that combines Large Language Models (LLMs) with Model Context Protocol (MCP) tools to generate reliable, budget-aware, and day-by-day travel itineraries. The system integrates live data from Airbnb (via MCP), Google Maps (distance, travel time, place photos), and real-time web search. It augments itineraries with social media content discovery (YouTube and Google Custom Search, TikTok/Instagram filtered) and supports ICS calendar export. A React frontend interfaces with a FastAPI backend and a Supabase persistence layer. Quantitatively, the system returns structured JSON itineraries with costs, addresses, hours, travel durations, and accommodation options; qualitatively, users reported improved trust and usability thanks to grounded data and transparent cost breakdowns.

3 Background and Motivation

Most existing travel planning tools only provide generic and standardized travel routes that are designed for mass tourism. They fail to reflect individual travel styles and personal needs like:

- local gourmet experience
- natural scenery exploration
- urban sightseeing
- shopping-focused routes

To address this gap, we leverage Xiaohongshu (小红书) to extract real, user-generated travel demands and preferences from posts. We will apply sentiment analysis to identify authenticity and adjust recommendation weights accordingly—ensuring personalized, trustable travel guidance.



4 Implemented Functions

This set of features integrates multi-tool-driven planning leveraging platforms like Airbnb, Google Maps MCP, and RedNote MCP to generate travel itineraries; enables structured JSON output for seamless direct rendering in frontends; incorporates social media discovery from YouTube, TikTok, and Rednote to capture travel trends and authentic local experiences; supports ICS calendar export for easy schedule synchronization; and ensures plan persistence and retrieval for authenticated users, allowing them to save and access their travel plans anytime.The actual process of each function will be introduced in detail below.

4.1 LLM Agent System

4.1.1 Travel Route Generation

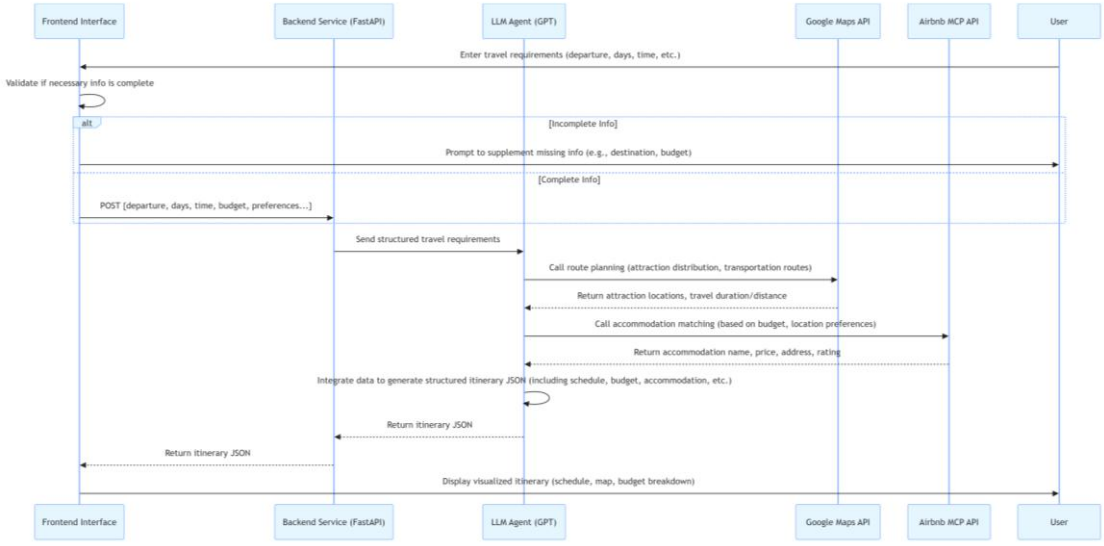
Tools: GPT 4o, Google Maps API, Airbnb MCP

Input: Departure, destination,start/end date of the trip, travel duration, number of travelers, total budget, travel preference.

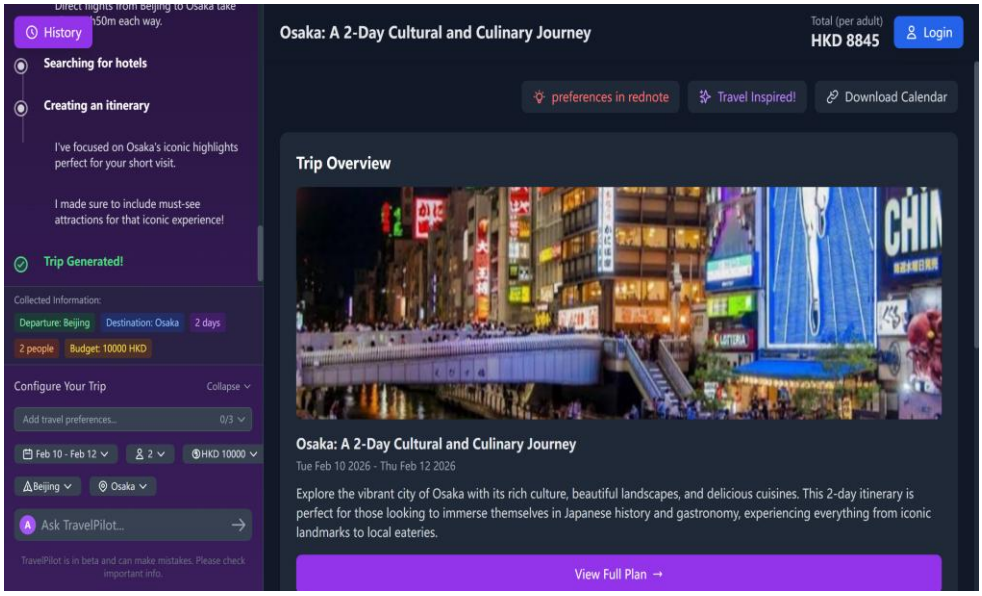
Output: A JSON follows the structure with "trip_overview", "accommodation", "daily_itinerary", and "budget_breakdown".

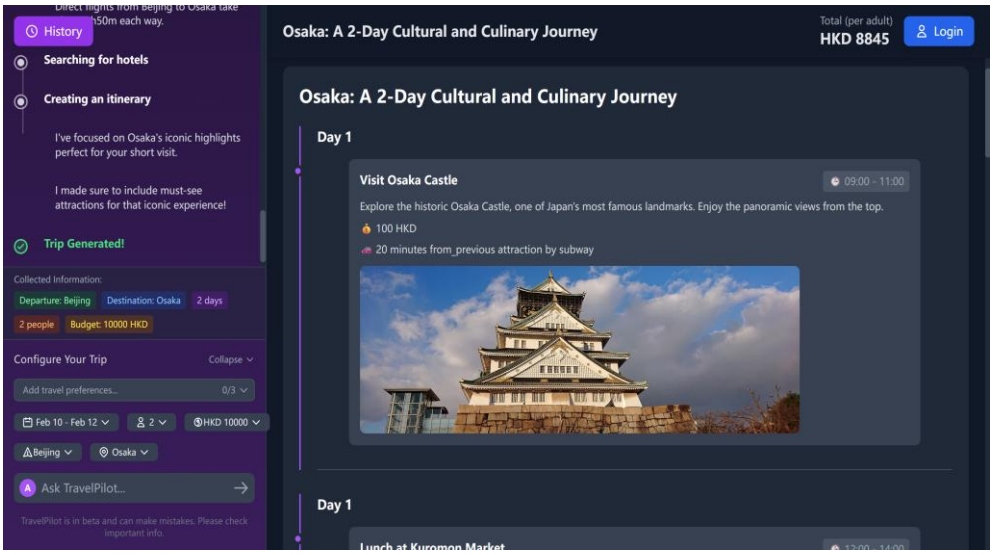
Implementation Procedure:

1. Receive user input and check if mandatory information (departure location, travel dates, travel duration) is complete; prompt the user to supplement if incomplete.
2. Call GPT model with complete information, which invokes Google Maps API (for route planning) and Airbnb MCP API (for accommodation matching) to generate the structured JSON itinerary as specified.



3. Display the query progress, detailed schedule information and scenic spot pictures on the front end





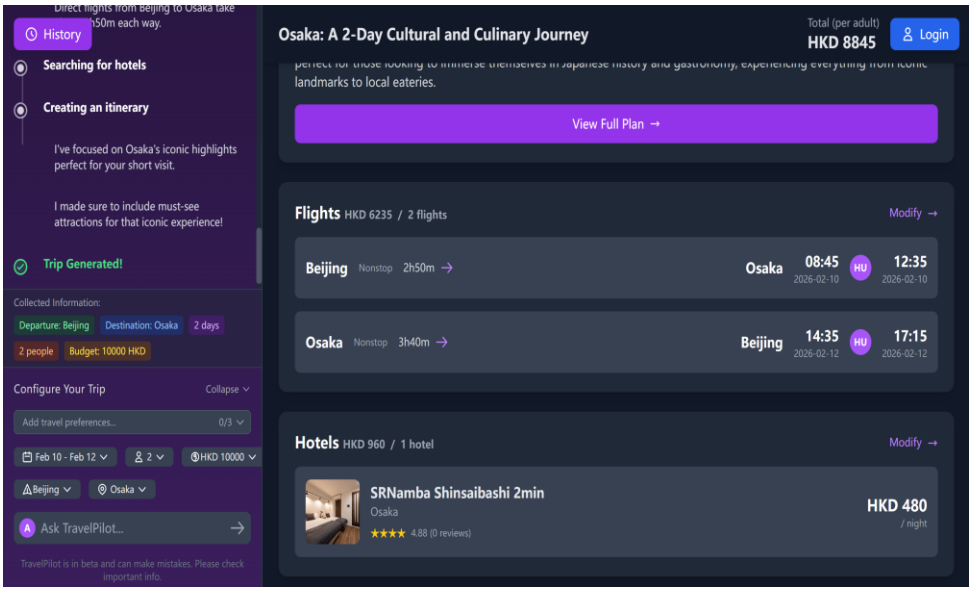
4. After users read the front-end travel plan, they can modify the basic travel information or put forward opinions according to their own ideas (such as replacing hotels or certain tourist attractions). The system will optimize the travel plan based on the user's personalized requirements.

4.1.2 Real-time flight ticket inquiry

Tool: Amadeus API

Input: Departure, destination,start/end date of the trip, number of travelers, total budget

Output: Display flights that fit the budget on the front end, including key information such as departure/arrival time, airline, duration.



4.2 Memory Augmentation Module

We provide persistent travel preference modeling:

Function	Implementation
Short-term memory	Temporary trip-related context
Long-term preference	Structured trip constraints stored in DB
Auto-application	Memory → constraints → itinerary regeneration

The project integrates Google OAuth 2.0 for user authentication, allowing users to log in via Google. It uses Supabase (PostgreSQL-based) to store travel plan histories. Authenticated users can save, retrieve, and manage their travel plans, which include details like destinations, itineraries, flights, and costs. The database structure balances structured fields for easy querying and flexible JSON storage for complex plan data. RESTful APIs ensure only authenticated users access their own travel plan data.

4.3 Social Content Integration

- Scrape relevant UGC posts based on user travel requirements
- Perform sentiment analysis to filter non-authentic content
- Assign confidence weighting before feeding into itinerary planning
- Align itinerary with actual travel recommendations from local users.

4.3.1 Preference-Aware Multi-Agent Recommendation

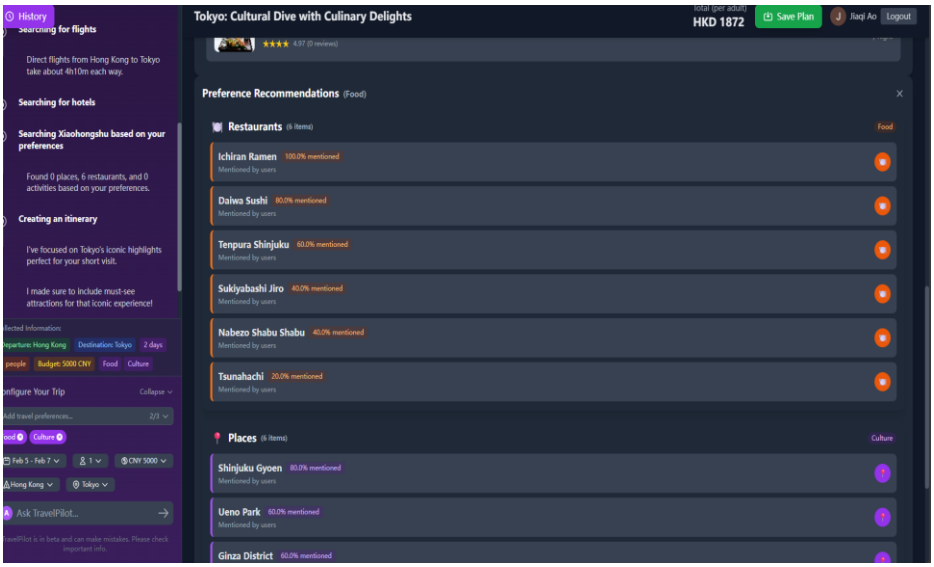
Main Agent (backend.py → run_mcp_travel_planner):

- Tools: Airbnb MCP, Google Maps MCP, Google Search Tools.

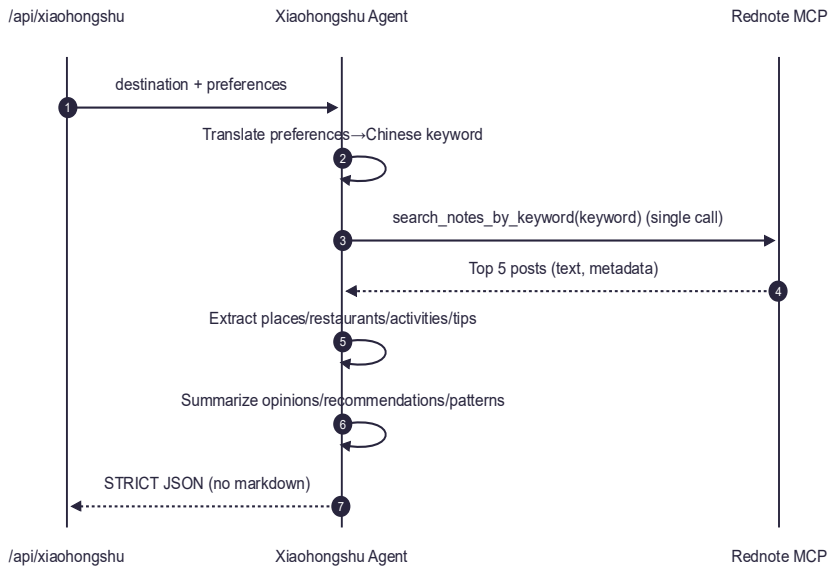
- Input: Departure , destination , budget , duration and so on ....
- Output: Full structured itinerary JSON (trip overview, daily schedule, budget breakdown).

Xiaohongshu Agent (xhs.py → run_mcp_xiaohongshu):

- Tools: Rednote MCP (rednote-mind-mcp).
- Input: Trip information, user preference
- Output: Preference-driven, structured recommendations from top-5 posts (places, restaurants, activities)



Prompted Workflow:



4.3.2 Location and Catergory Based Media Retrieval Feature

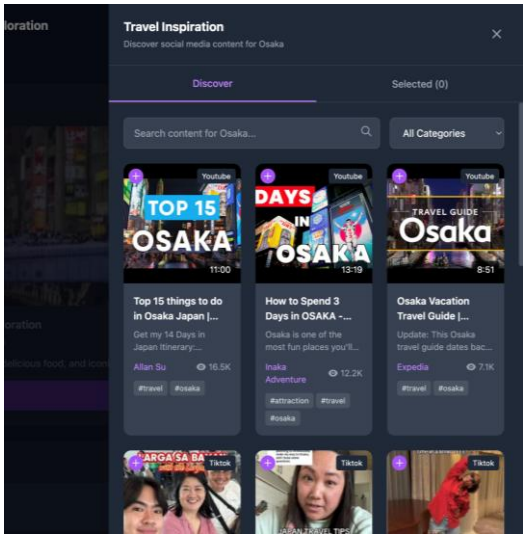
Route: POST /api/social-media-content

Tools: YouTube Data API v3: For fetching YouTube video metadata (title, description, thumbnail, etc.), Google Custom Search JSON API: For discovering travel-related content from blogs and websites.

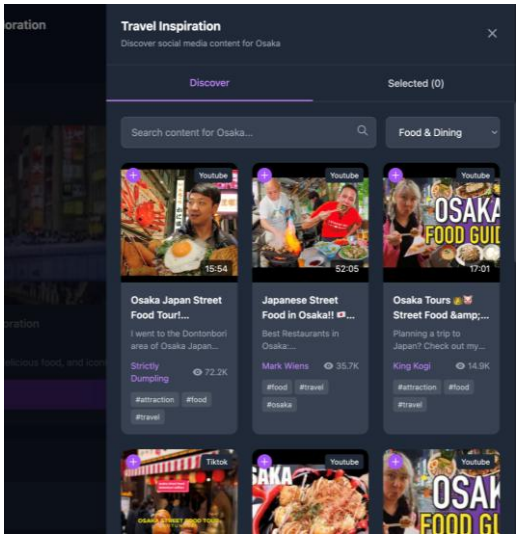
Input: API Request Body (SocialMediaRequest Model)

Output: API Response (SocialMediaResponse Model)

All Categories



Food & Dining



YouTubeService: Fetches high-quality travel videos from YouTube.

Process:

Search: Uses the YouTube Data API to search for videos with a query like "{destination} travel guide {tags}".

Detail Enrichment: For each video found, a second API call fetches detailed information, including contentDetails (for video duration) and statistics(for like count).

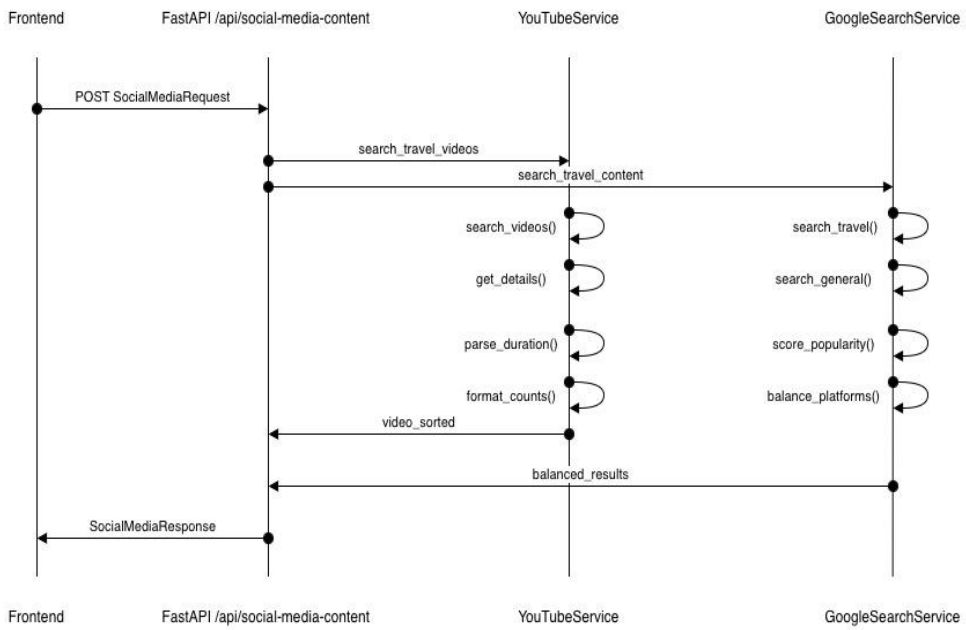
Data Processing:

Duration Parsing: Uses the isodate library to convert the ISO 8601 duration format (e.g., PT15M30S) into a human-readable string (e.g., 15:30).

Count Formatting: Converts raw like counts into abbreviated, user-friendly strings (e.g., 155000 becomes 155K).

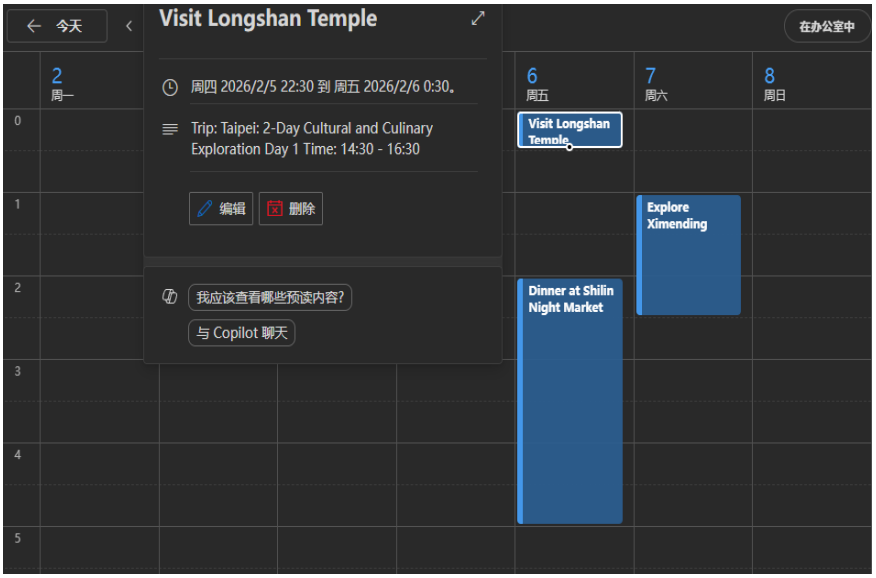
Ranking: Sorts the final list of videos by view_count in descending order to present the most popular content first.

Workflow:

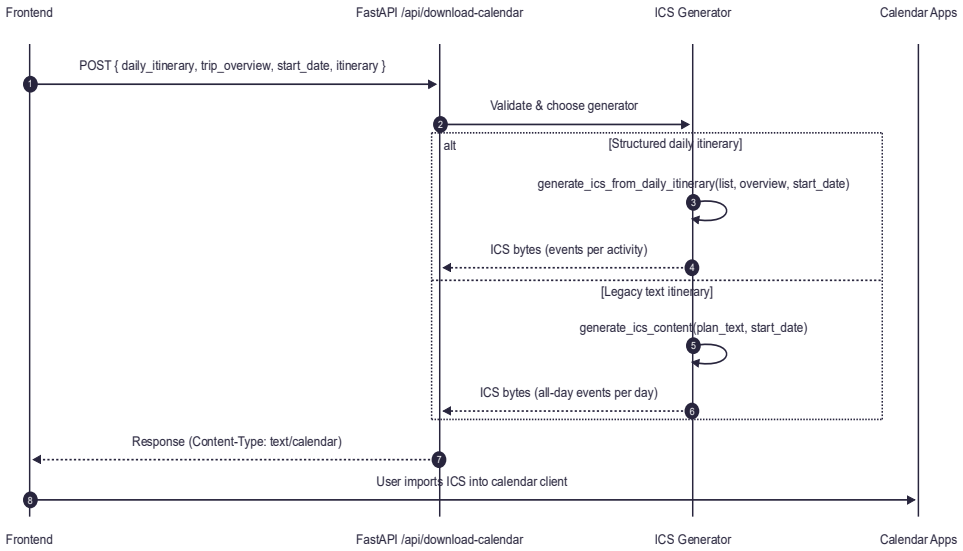


4.4 Exportability (ICS calendar)

We provide seamless calendar export of generated itineraries to .ics files compatible with Google Calendar, Apple Calendar, and Outlook. Frontends can trigger “Download as Calendar” with minimal coupling.



Prompted Workflow:



5 Technical route

We adopt a front-end & back-end decoupled architecture to ensure flexibility and modularity.

Layer	Technology stack	Purpose
Frontend Web UI	React.js	Responsive interactive interface, itinerary visualization, real-time collaboration components, data visualization charts
Backend Service	FastAPI (Python)	Agent logic layer, tool orchestration, session storage
Intelligent Planning （LLM）	GPT	LLM scheduling, tool call chain management, and memory module interface
External Service Tools	Xiaohongshu MCP, Google Maps MCP, Airbnb MCP	Xiaohongshu comment crawling, Google Maps for map information, and Ctrip for flight and hotel information
Data Storage	PostgreSQL / JSON cache	Memory, preference persistence, caching external data