**LLM DRIVEN TRAVEL AUTOMATION:**

**SCALABLE AGENT DESIGN FOR PERSONALIZED PLANNING**

By **Dhriti Malik, Iris Luo, Leo Zhou, Quan Nguyen**,

sponsored by **TravEXP**, and advised by professor **Caliendo**

1. **ABSTRACT**

This project aims to enhance the travel planning experience through the integration of Large Language Models (LLMs) into a scalable agent framework. Our core objective was to develop and optimize APIs that improve the relevance and accuracy of search engine results within travel platforms.

By leveraging the power of LLMs, our system generates precise, context-aware, and user-friendly responses to complex travel queries. The proposed solution assists travelers with various components of trip planning—including visa requirements, accommodation, and activity recommendations—through a personalized and interactive experience. This initiative supports the growing demand for intelligent travel assistants that can adapt to individual user needs in real-time.

1. **INTRODUCTION.**

In today’s digital landscape, travel planning is fragmented, with users navigating between maps, blogs, forums, and booking platforms. Finding itineraries tailored to specific needs or personas is often overwhelming, and coordinating group trips adds to the complexity.

TravEXP addresses these challenges by streamlining the planning process through:

* AI-driven, persona-based experience browsing
* Collaborative itinerary planning
* Community sharing and reward-based engagement

This project demonstrates how AI can transform scattered travel data into curated, interactive, and scalable trip-planning tools.

1. **CHALLENGES**

Developing the AI-powered deals agent for TravExp presented a series of complex challenges that spanned data integration, system design, API limitations, and AI model behavior. While several of these were anticipated in our initial project plan, others emerged during the iterative development process.

**Handling Large-Scale and Unstructured Data**

One of the most significant technical challenges was handling large-scale data across multiple third-party APIs. Each provider, Viator, TripAdvisor etc., offered data in different formats, with varying levels of completeness and reliability. This required extensive data preprocessing to ensure consistency across sources. We implemented robust data filtering, normalization procedures for fields like price and rating, and created a modular JSON structure that could accommodate multiple deal types (stay, car, and activity). These steps enabled a streamlined data pipeline that was compatible with both LLM-based personalization and frontend display requirements.

**Limited and Restricted API Access**

Another major challenge was the issue of limited and restricted access to certain APIs. Some platforms required formal partnerships or offered only sandbox environments, while others provided insufficient documentation or limited query parameters. For instance, the TripAdvisor API did not support comprehensive car rental listings. To address this, we prioritized the use of publicly accessible endpoints and adopted flexible response handling strategies.

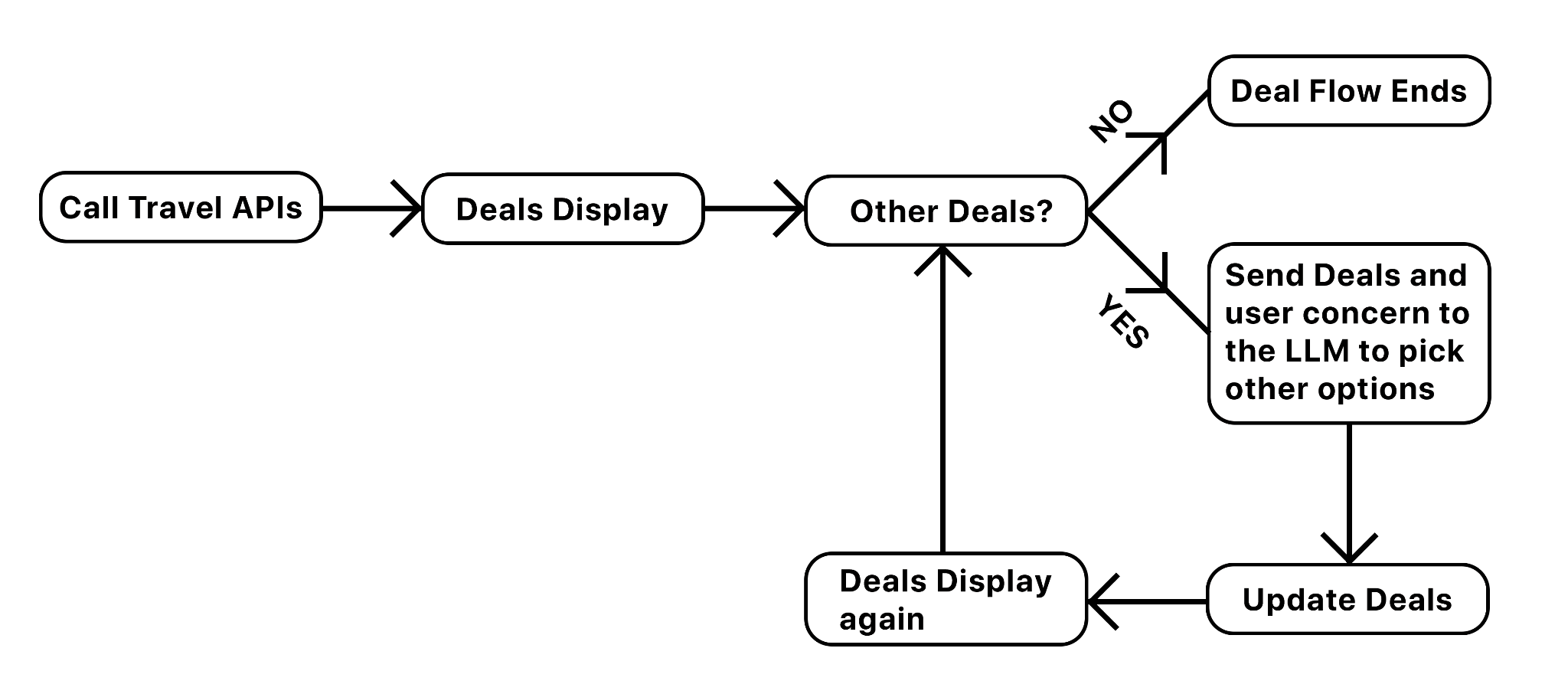
**LLM Prompt Tuning and Consistency**

Challenges also arose from our use of large language models (LLMs) for ranking deals and providing visa-related responses. While these models were powerful, they often produced inconsistent or verbose outputs depending on how prompts were structured. We iteratively refined our prompt templates, balancing clarity, conciseness, and context awareness. Additionally, we tested both OpenAI’s GPT-4 Turbo and Google’s Gemini 1.5 Pro to identify trade-offs in accuracy, official source referencing, and engagement. To enhance reliability, we also implemented short-term conversation memory for visa queries and structured the JSON responses with clear rationale fields.

Finally, with multiple features being developed simultaneously, deal sourcing, visa chatbot, LLM pipelines, maintaining alignment across the system was a coordination challenge. To address this, we organized weekly team syncs, followed a modular development approach, and conducted early integration testing using mock data. This allowed us to detect misalignments and ensure that APIs, prompts, and UI-ready outputs could communicate smoothly across the system. These challenges ultimately strengthened the robustness and adaptability of our solution, and the lessons learned will continue to inform the future development and deployment of the TravExp platform.

These challenges ultimately strengthened the robustness and adaptability of our solution, and the lessons learned will continue to inform the future development and deployment of the TravExp platform.

1. **PROJECT DEVELOPMENT**
2. **Project development**

****

1. **Work distribution**

**Dhriti Malik:**

As part of the TravExp Capstone project, I led the end-to-end development of the Visa Advisory Chatbot and played a key role in enhancing the Hotel Recommendation System's performance and usability.

* **Visa Chatbot Development**
  + Designed the dual-LLM architecture where:
    - Gemini 1.5 Pro generates trusted, source-backed visa responses.
    - GPT-4 suggests relevant follow-up questions to improve user experience.
  + Crafted context-aware prompt templates that adapt to user-provided nationality and destination.
  + Built and tested a conversational framework with memory tracking (deque) to simulate a real advisor.
  + Implemented model output evaluation using similarity score visualizations to benchmark Gemini vs GPT.
* **Hotel Stay Optimization:**
  + Rebuilt the backend flow to support parallel API calls via ThreadPoolExecutor, improving first-run response time by 30–40%.
  + Introduced a local caching system that stores hotel and LLM outputs as .json, reducing second-run latency by up to 3×.
  + Conducted extensive time benchmarking across cities and created visual comparisons that clearly illustrate improvements.
  + Helped format output into clean, JSON-ready responses that integrate seamlessly with the front-end website.

**Iris Luo:**

As part of the TravExp Capstone project, I led the development of the Activity Deals module and supported project-wide coordination including poster content, meeting documentation, and presentation support.

* **Activity Deals System**
  + Built the end-to-end pipeline to fetch and format activity recommendations from the Viator API.
  + Developed reverse geocoding and fuzzy matching to convert user coordinates into Viator destination IDs.
  + Applied multi-criteria filtering (rating, reviews, cost) and output structured JSON including activity types, costs, duration, booking links etc. for LLM processing.
  + Conducted integration testing to ensure compatibility with UI and LLM modules.
* **Teamwide Contributions**
  + Helped prepare and deliver content for presentations.
  + Drafted and refined several poster sections and coordinated layout inputs.
  + Took some detailed meeting minutes to support team alignment and project tracking.

**Leo Zhou:**

As a team member on this travel recommendation project, I took on two significant areas of responsibility:

* **API Integration and Testing:**
  + Initially tested Amadeus API for hotel data integration, conducting comprehensive evaluations of data quality, coverage, and response times.
  + After comparative analysis, it was determined that TripAdvisor's API provided more comprehensive hotel information with better ratings consistency and geographic coverage.
  + Made the data-driven decision to pivot from Amadeus to TripAdvisor, improving our recommendation quality significantly.
* **LLM Enhancement Development:**

Designed and implemented two core LLM-powered modules that transformed raw API data into personalized recommendations:

* + **Hotel LLM Module**
    - Developed the get\_better\_deals function that analyzes user preferences against TripAdvisor hotel data.
    - Created hotel comparison functionality that generates detailed strength/weakness analysis.
    - Implemented booking consideration generation to highlight important factors before reservation.
    - Built a nearby attraction recommendation system based on selected hotel locations.
  + **Activity LLM Module**
    - Engineered the activity categorization system that intelligently organizes Viator activities into three intuitive groups.
    - Developed preference-based ordering that prioritizes activity categories according to user interests.
    - Created personalized recommendation rationales for each suggested activity.
    - Implemented robust JSON parsing and error handling to ensure consistent performance.

**Quan Nguyen:**

* **Leadership:**

As the project lead, I led a team of four engineers and collaborated closely with TravEXP founder and professor/advisor Caliendo to deploy TravExp.live, a personalized, AI-powered travel planning platform. Our solution integrated Large Language Models (LLMs) with real-time data APIs to deliver intelligent recommendations for hotels, activities, and visa requirements through a user-friendly interface. I oversaw the end-to-end development lifecycle — from designing the architecture to refining API functionality and optimizing LLM outputs for context-aware, concise travel recommendations.

* **Hotel Data Extraction + LLM integration:**

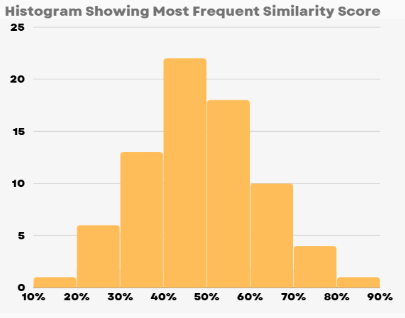
My contributions focused on driving the personalized hotel recommendation engine, where I implemented filtering logic, ranking functions, and LLM-generated summaries that adapt to user preferences. I coordinated data extraction from sources like TripAdvisor, structured it into dynamic JSON outputs, and ensured a seamless connection between backend data processing and interactive front-end web components. Through a blend of engineering leadership and applied machine learning, I helped shape a scalable, modular system that bridges natural language intelligence with real-world travel planning needs.

1. **VISA ADVISORY CHATBOT**

The Visa Advisory Chatbot is designed to simplify the complex and often ambiguous process of understanding visa requirements for international travelers. Travelers frequently struggle with navigating fragmented information spread across embassy websites, blogs, and outdated forums. This chatbot aims to bridge that gap by providing instant, accurate, and context-aware visa guidance using Large Language Models (LLMs).

**Key Features**

* **Dual-Model LLM Architecture**
  + We implemented a hybrid model approach, combining Google’s Gemini 1.5 Pro and OpenAI’s GPT-4 Turbo.
  + Gemini is used to generate the core visa advisory answer, citing official sources and giving up-to-date requirements for nationality and destination-specific queries.
  + GPT-4 is used to generate 3–4 relevant follow-up questions that guide the user toward further clarification (e.g., questions about visa extensions, study/work permits, vaccination requirements).
* **Context-Aware Prompt Engineering**
  + Prompts include structured formatting (e.g., “Question → Answer → Suggested Questions”).
  + User nationality and destination are dynamically embedded to tailor responses.
  + A conversation memory buffer (deque) stores the last 5 queries to simulate continuity in multi-turn conversations.
* **User-Centric Output Design**
  + Responses are structured, concise, and free from LLM hallucinations.
  + Gemini outputs are encouraged to cite source links (e.g., immigration portals, embassy pages).
  + Suggested questions from GPT offer logical next steps without overwhelming the user.
* **Evaluation via Similarity Scoring**
  + We conducted an inter-model comparison using a similarity scoring histogram, evaluating response consistency between GPT and Gemini for the same queries.
  + The most frequent similarity scores ranged between 40–60%, reflecting complementary rather than redundant strengths between the two models.



1. **STAY DEALS**

A major component of our project is a personalized hotel recommendation engine built to help users identify ideal accommodation options based on location, price, and user demographics.

**Key Features:**

* Location-Based Search: Automatically fetches the top 10 hotels near a city or address specified by the user.
* Filtering Mechanism: Hotels can be filtered using price levels (e.g., $, $$, $$$), enabling tailored searches based on individual budgets.
* LLM Integration (Gemini): The model generates concise, natural language summaries of hotels. It also offers personalized guidance such as nearby activity suggestions and follow-up tips depending on user needs.
* Rich Output Format: The system presents structured hotel details—including names, pricing, ratings, reviews, addresses, and TripAdvisor links—in a clean, tabular JSON-ready format. This makes it suitable for integration with front-end platforms or chatbot-based assistants.

**Workflow Breakdown:**

Using TripAdvisor search data, we constructed a pipeline consisting of the following steps:

1. **Basic Hotel Details Extraction:** A list of hotels is extracted including name, price category, rating, number of reviews, and TripAdvisor URL.
2. **Function Definitions for Filtering:** Functions are implemented to filter by attributes such as price tier, enabling dynamic searches (e.g., only showing $$ or $$$ results).
3. **Detailed Hotel Information Structuring:** The raw data is structured to allow rich representation of all key details: Name, Price, Rating, Ranking, Reviews, Address, and Links.
4. **Hotel Ranking Logic:** Hotels are ranked based on multiple factors including reviews and ranking scores, helping users quickly find the best-rated, best-value options.

**Conclusion**

By integrating real-time data with LLM intelligence, this system creates a foundation for AI-powered travel assistants. The personalization capabilities ensure that users receive recommendations that are relevant, insightful, and streamlined, reducing the cognitive load involved in planning a trip.

1. **ACTIVITY DEALS**

The activity deals is a core component of the TravExp Deals Agent, designed to recommend personalized, real-time experiences such as tours, events, and local attractions based on a user’s location and preferences. This feature focused on integrating the Viator API to fetch curated activity options, applying filters and logic to ensure that the results were not only relevant but also high in quality and diversity.

To enable location-aware search, we began by implementing a reverse geocoding pipeline using the OpenStreetMap API. Once the destination ID was identified, we used Viator’s product search endpoint to pull detailed activity data. The results included attributes such as activity type, duration, price, traveler rating, review count, and booking link etc. We applied filtering rules to prioritize activities with high ratings and a sufficient number of reviews, considering rating first, then review volume, followed by cost in particular. It ensures that the top-ranked results were well-reviewed, popular, and suitable for users across budget levels.

In summary, the activity deals system successfully bridges geographic input with specific experience data by combining geolocation processing, API integration, filtering logic, and structured formatting.

1. **LARGE LANGUAGE MODELS**

Our primary goal was to transform standard travel search results into personalized, intelligent recommendations by integrating large language models with traditional API data. We aimed to create a system that understands nuanced user preferences and provides tailored suggestions with clear rationales, mimicking the expertise of a knowledgeable travel advisor. By combining the comprehensive data from TripAdvisor and Viator APIs with the analytical capabilities of Gemini 1.5 Pro, we sought to solve the common problem of information overload in travel planning and help users discover options that truly match their unique preferences.

**Key Features**

* **Personalized Hotel Recommendations**

Our Hotel LLM module analyzes user preferences and TripAdvisor data to generate targeted hotel suggestions. When users specify criteria like "low price" or "family-friendly," the system identifies the most suitable options and explains why each recommendation matches their needs. For example, when a user requests budget-friendly accommodations in Boston, the system not only identifies The Revolution Hotel as an affordable option but explains its value proposition compared to other properties. The system also provides detailed hotel comparisons highlighting strengths, potential drawbacks, and target traveler types, alongside generating relevant booking considerations to help users make informed decisions.

* **Categorical Activity Classification**

The Activity LLM module intelligently categorizes travel experiences into three intuitive groups: Outdoors & Activity, Art, Cultural & Historical, and Laid-back. This classification allows users to navigate options based on their interests rather than sifting through hundreds of unorganized listings. The system applies sophisticated prompting techniques to ensure accurate categorization, even for activities that could potentially span multiple categories. Users can prioritize these categories according to their preferences, ensuring they see the most relevant suggestions first. This approach creates a more natural discovery process that aligns with how travelers actually think about their interests.

* **Robust Error Handling and Structured Outputs**

We developed sophisticated JSON parsing mechanisms to ensure reliable performance even when LLM responses vary. Our system includes fallback strategies for handling unexpected output formats, ensuring consistent user experiences without technical failures. The structured output format presents key information clearly, making complex travel decisions more manageable. We designed the display components using pandas and HTML to create intuitive, visually organized recommendations that highlight the most relevant details for each option. This robust architecture maintains a consistent user experience while extracting maximum value from the AI's analytical capabilities.

**Conclusion**

By leveraging Gemini 1.5 Pro's capabilities, we've created an intelligent recommendation system that bridges the gap between overwhelming data and personalized travel planning. Our LLM-enhanced approach not only saves users time but provides the context and explanations behind each recommendation, empowering more confident travel decisions. The system demonstrates how AI can transform travel planning from a tedious filtering process into an intuitive, personalized experience.

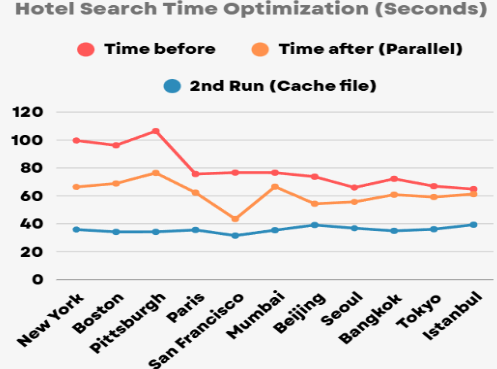
The true innovation of our system lies in its ability to understand the "why" behind user preferences and connect them to the right options with clear explanations. Rather than simply matching keywords or applying filters, our LLM-enhanced system recognizes nuanced concepts like "stylish but affordable" or "family-friendly but still sophisticated" and translates these into meaningful recommendations. As travel planning continues to evolve, this approach represents a significant step toward more intelligent, human-centered digital travel assistance that combines the breadth of online options with the personalized touch of an expert advisor.

**Speed Optimization**

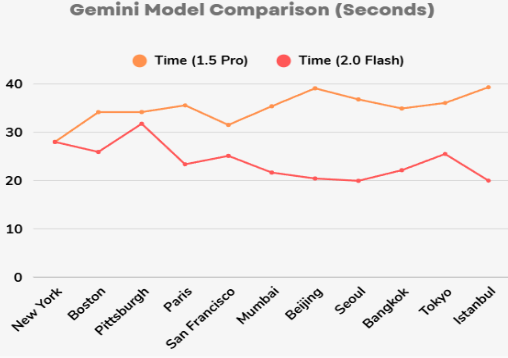
As we scaled the Hotel Recommendation Chatbot, runtime performance emerged as a critical concern. Initially, the system fetched details (description, reviews, ratings, and photos) for up to 10 hotels sequentially from the TripAdvisor API, resulting in slow response times (~100 seconds in some cases). We undertook a systematic optimization process that involved both backend logic improvements and caching strategies.

**Optimization Techniques**

* **Parallel API Calls with ThreadPoolExecutor**
  + We wrapped API calls (e.g., hotel details and images) in a ThreadPoolExecutor, enabling concurrent execution across hotel locations.
  + This reduced the first-run fetch time by up to 30–40%, depending on network latency and API response consistency.
  + A chart was created to showcase performance improvements across major cities (New York, Tokyo, Paris, etc.), comparing before vs. after parallel execution.
* **Local Caching with JSON Files**
  + To eliminate redundant network calls, hotel data and LLM-generated summaries were stored in .json cache files locally.
  + On subsequent runs, if the same hotel was queried again, data was retrieved instantly from cache, reducing average response time to under 40 seconds on second runs.
  + This not only improved speed but reduced API token usage and costs, which is critical for scalable deployment.
* **Time Benchmarking and Visualization**
  + We measured processing times for:
    - Before optimization
    - After parallelization
    - Cached second runs
  + Results were visualized in a multi-line chart titled "Hotel Search Time Optimization", providing a clear view of speed gains across various scenarios.



* **Gemini Model Swap and Testing**
  + We also tested Gemini 2.0 Flash vs 1.5 Pro, plotting latency results across cities.
  + Gemini Flash consistently outperformed 1.5 Pro in inference speed by 5–15 seconds depending on complexity, while maintaining acceptable quality.



1. **ACCOMPLISHMENT**

Over the course of the project, our team successfully designed and implemented an AI-powered travel planning system that combines real-time data APIs with large language models to deliver personalized, explainable, and scalable recommendations for travelers.

**We developed a complete multi-component pipeline that:**

* Integrated data from Viator, TripAdvisor, and Amadeus APIs to retrieve real-time travel deals for activities and hotel deals.
* Engineered a robust Visa Advisory Chatbot using Gemini 1.5 Pro and GPT-4, offering up-to-date, source-backed answers with follow-up suggestions.
* Structured all results in modular JSON outputs, enabling clarity in downstream LLM processing and potential UI integration.
* Built LLM modules that ranked and explained recommendations using user preferences, enhancing transparency in travel planning.
* Optimized system performance using parallelization and local caching, reducing hotel search latency by up to 3×.

In addition to our technical development, we created a clean and extensible system architecture, a visualized poster summarizing our framework, and a functional backend that demonstrates the potential of combining LLM reasoning with structured travel data. Our solution supports the growing need for intelligent digital travel assistants and lays the foundation for future UI deployment and user-facing applications.

We look forward to expanding this system through frontend integration, broader API coverage, and continued refinement of our personalization and performance capabilities.