

**Global AI Agents League**

**Fetch.ai Innovation Lab Hackathon 2025**

**Team Members**

1. Rohit Kosamkar (Northeastern University, Boston, MA)
2. Sapna Chavan (Northeastern University, Boston, MA)

Table of Contents

[Introduction 2](#_Toc194372553)

[Inspiration 2](#_Toc194372554)

[What it does 4](#_Toc194372555)

[How we built it 5](#_Toc194372556)

[Challenges we ran into 5](#_Toc194372557)

[Accomplishments that we're proud of 6](#_Toc194372558)

[What we learned 6](#_Toc194372559)

[What's next for TravelGenie AI-Agent 6](#_Toc194372560)

[Conclusion 6](#_Toc194372561)

[References 8](#_Toc194372562)

# Introduction

TravelGenie is an AI-powered multi-agent travel planner built to transform the conventional experience of travel itinerary generation. Unlike existing large language model (LLM) chatbots such as ChatGPT or Gemini that deliver plain-text travel suggestions, TravelGenie offers a highly interactive and engaging experience. The platform collects minimal input from the user such as source, destination, and travel dates—and dynamically generates a detailed travel itinerary enriched with:

* Real-time weather conditions
* Route and travel mode suggestions
* Top flight options
* Popular tourist attractions
* Recommended restaurants
* Local events and happenings
* A concise itinerary summary

Built using technologies like Langchain and Gemini 2.0 Flash, it orchestrates multiple intelligent agents to autonomously gather and deliver contextual travel information in one sleek, unified dashboard.

What sets TravelGenie apart is its modular agent-based architecture and visually rich UI that goes beyond mere conversation to deliver a truly immersive planning experience.

# Inspiration

While using various AI chatbots to plan trips, we noticed that most provide long, text-heavy responses which are difficult to digest and often lack the integration of real-time data. We envisioned a platform that feels like true travel assistant—visually intuitive, engaging, and precise. TravelGenie was born out of the desire to make travel planning fun, interactive, and intelligent.

Architecture

At the heart of TravelGenie lies a modular multi-agent system orchestrated via a supervisor agent. Here's how the architecture flows:

A diagram of a company

AI-generated content may be incorrect.

Figure 1. Architecture

1. **User Input via Web UI**  
   Users enter natural language queries (e.g., "Plan a trip from Boston to NYC on April 5").
2. **Supervisor Agent**  
   This is the brain of the system, powered by:

* **Intent Classifier**: Detects whether the user query is travel-related using Gemini 2.0 Flash. If not travel-related, the user gets a friendly response and no agents are triggered.
* **Session Memory**: Stores all trip-related details across multiple interactions.
* **Extractor Module**: Parses travel fields like source, destination, and dates from the message.

**Validator**: Ensures source ≠ destination, valid dates, and trip length ≤ 14 days.

If all required details are captured, it returns "ready": True, triggering the core agents.

1. **TravelGenie Core**  
   Once the trip is validated, this component intelligently coordinates with:

* **Weather Agent**: Destination and Source City weather forecast
* **Route Agent**: Driving or transit routes
* **Flight Agent**: Real-time flights via Amadeus API
* **Places Agent**: Must-visit attractions
* **Restaurant Agent**: Top-rated restaurants
* **Event Agent**: Local events via Ticketmaster

1. **Gemini 2.0 Flash & LangChain**  
   Used within the classifier and extractor for intent parsing, LLM-powered memory completion, and ReAct-style orchestration.
2. **Dashboard**  
   After processing, all agent outputs are sent to the UI as rich visual cards on the TravelGenie dashboard.

# What it does

TravelGenie uses a multi-agent system orchestrated by a Supervisor Agent to:

1. Collect user inputs via a conversational chatbot.
2. Dispatch sub-agents to perform specialized tasks:
   * **Weather Agent**: Fetches weather forecasts using OpenWeatherMap API.
   * **Route Agent**: Computes route and transport suggestions between cities.
   * **Flight Finder Agent**: Searches top flight options using the Amadeus API.
   * **Food Explorer Agent**: Retrieves top-rated restaurants using Google Maps API.
   * **Event Agent**: Lists upcoming events using the Ticketmaster API.
   * **Explorer Agent**: Recommends popular places via Google Places API.
   * **Summary Agent**: Summarizes the itinerary with useful tips and prep advice using Gemini.
3. Display the full itinerary on a dynamic TravelGenie dashboard.

All sub-agents are registered and live on Agentverse, ensuring modularity and extensibility.

* [Google API Route Agent] [Link](https://agentverse.ai/agents/details/agent1qfetzztsmf373m7qa4x6l5mjveykecm87cu9m7myaxd7ukp0rwrjkkax9wt/profile)
* [Weather Agent] [Link](https://agentverse.ai/agents/details/agent1q2zc0ctnxwyzd7xxgafwzwcd7tkzg7w790zjret9vwm2f38gmm5dv7wm22j/profile)
* [Google Places Explorer Agent] [Link](https://agentverse.ai/agents/details/agent1q0vj2jq0xe05zdukwnesaqvey0sve99gahvfp452yrlemtng8axzvcf8hq0/profile)
* [Flight Search Agent] [Link](https://agentverse.ai/agents/details/agent1qd3qj9wsjwdasgqu2w3gzwjgzc3cjx7pwdl26th3r67y46umgulwxu4emrg/profile)
* [Restaurant Explorer Agent] [Link](https://agentverse.ai/agents/details/agent1qtcaxwkqgs0kgfmtsr90j9lcpjk3a7t3mtkf9sty8a07gdr0el6sjgkzy60/profile)
* [Event Explorer Agent] [Link](https://agentverse.ai/agents/details/agent1qvmkshe4kn3ucfsecd9d703jqzqy858z5l2unn9qye3y6dyvc5ph20zmqch/profile)

# How we built it

* **Frontend**: Developed in React and Angular with Tailwind CSS for styling, providing a modern, responsive UI.
* **Backend**: Built with FastAPI to handle message routing and agent orchestration.
* **Multi-Agent System**: Each function is handled by an autonomous agent built using the uagents framework and registered on Agentverse.
* **Agent Integration**: APIs like Amadeus, Google Maps, Weather API, and Ticketmaster were integrated into agents with secure key management.
* **Agentverse**: All agents were deployed, tested, and hosted on Agentverse for public discovery and collaboration.
* **Orchestration & Planning:** Langchain’s ReAct paradigm and Gemini 2.0 Flash were used for LLM reasoning, intent parsing, and response generation.
* **UI Experience**: Users chat via a bot interface; once details are complete, the dashboard auto-generates with all trip components visually presented.

# Challenges we ran into

* **Data Orchestration**: Syncing multiple agents to work asynchronously while maintaining context requires meticulous design.
* **Flight Data Scrapper**: We built a flight data scrapper using selenium for kayak.com which was taking time to display information on dashboard then we pivot our plan to Amadeus API.
* **Frontend/Backend Decoupling**: Making the UI react to backend messages in real-time without blocking UX.
* **Agent Registration**: Debugging message model issues while registering on Agentverse.

# Accomplishments that we're proud of

* Successfully deployed and registered **6+ intelligent agents** on Agentverse.
* Built a fully functional and visually appealing **TravelGenie dashboard** that compiles and presents trip insights.
* Created reusable agent frameworks for food, weather, events, and flights that can be integrated into other systems.
* Made the UX seamless—from chat interaction to dashboard transition—with minimal user inputs.

# What we learned

* Deepened our understanding of agentic architecture and their role in orchestrating intelligent workflows.
* Gained hands-on experience in integrating and handling real-time data from multiple APIs.
* Understood the intricacies of multi-agent coordination, error handling, and scalability.
* Leveraged Agentverse to expose and deploy public agents effectively.

# What's next for TravelGenie AI-Agent

* **User Login & Personalization**: Save past itineraries, preferences, and history.
* **Cost Optimization Agent**: Estimate trip cost and provide budgeting tips.
* **Hotel Finder Agent**: Integrate hotel booking APIs for lodging suggestions.
* **Voice-based Input**: Make the interaction hands-free for travelers on-the-go.
* **Collaborative Planning**: Allow multiple users to co-plan trips in real-time.
* **Mobile App Version**: Extend TravelGenie to a native mobile experience.

# Conclusion

Building TravelGenie during this hackathon has been an incredible journey—equal parts challenging and rewarding. From experimenting with autonomous agents to integrating real-time APIs and reasoning through Langchain + Gemini, we pushed ourselves beyond comfort zones. This project not only deepened our understanding of AI-agent systems but also sparked a passion for creating human-centric, intelligent applications. TravelGenie is more than just a hackathon project—it’s a glimpse into the future of how we plan and experience travel.

*TravelGenie is more than a travel planner—it's an intelligent agent ecosystem designed to make your journey planning seamless, enjoyable, and informative.*

# References

* Agentverse. (n.d.). *Discover AI Agents for every use case*. Retrieved April 1, 2025, from <https://agentverse.ai/agents>
* Amadeus. (n.d.). *Amadeus for Developers – Travel APIs*. Retrieved April 1, 2025, from <https://developers.amadeus.com/>
* Google Cloud. (n.d.). *Cloud APIs documentation*. Retrieved April 1, 2025, from <https://cloud.google.com/apis?hl=en>
* LangChain. (n.d.). *Build context-aware applications with language models*. Retrieved April 1, 2025, from <https://www.langchain.com/>
* OpenWeatherMap. (n.d.). *Weather API – OpenWeather*. Retrieved April 1, 2025, from <https://openweathermap.org/api>
* Render. (n.d.). *Cloud Hosting for Developers*. Retrieved April 1, 2025, from <https://render.com/>
* Ticketmaster. (n.d.). *Discovery API Documentation*. Retrieved April 1, 2025, from <https://developer.ticketmaster.com/products-and-docs/apis/discovery-api/v2/>
* Vercel. (n.d.). *Vercel – Frontend Cloud Platform*. Retrieved April 1, 2025, from <https://vercel.com/>
* Fetch.AI. (2025). *Fetch.ai Global AI Agents League Hackathon*. Retrieved from <https://fetch-ai-hackathon.devpost.com/>
* Gemini. (n.d.). *Gemini 2.0 Flash by Google DeepMind*. Retrieved April 1, 2025, from https://deepmind.google/technologies/gemini/