AI-Powered Travel Planner

Ananya Makwana
Faye Yang
Jianen Peng
Rosnita Dey

Abstract - The Travel Planner is designed to streamline collaborative trip planning for young adults by centralizing group coordination and automating itinerary creation. By utilizing a multi-agent AI system with cloud-based technologies, the platform analyzes user preferences, availability, and budget constraints to generate personalized and optimized travel plans. The solution aggregates data through user surveys, collects destination research, and dynamically adjusts itineraries using integrated AI agents. This paper provides a detailed overview of the system architecture, core functionalities, user personas, testing methodologies, security considerations, and the technologies utilized in building the solution.

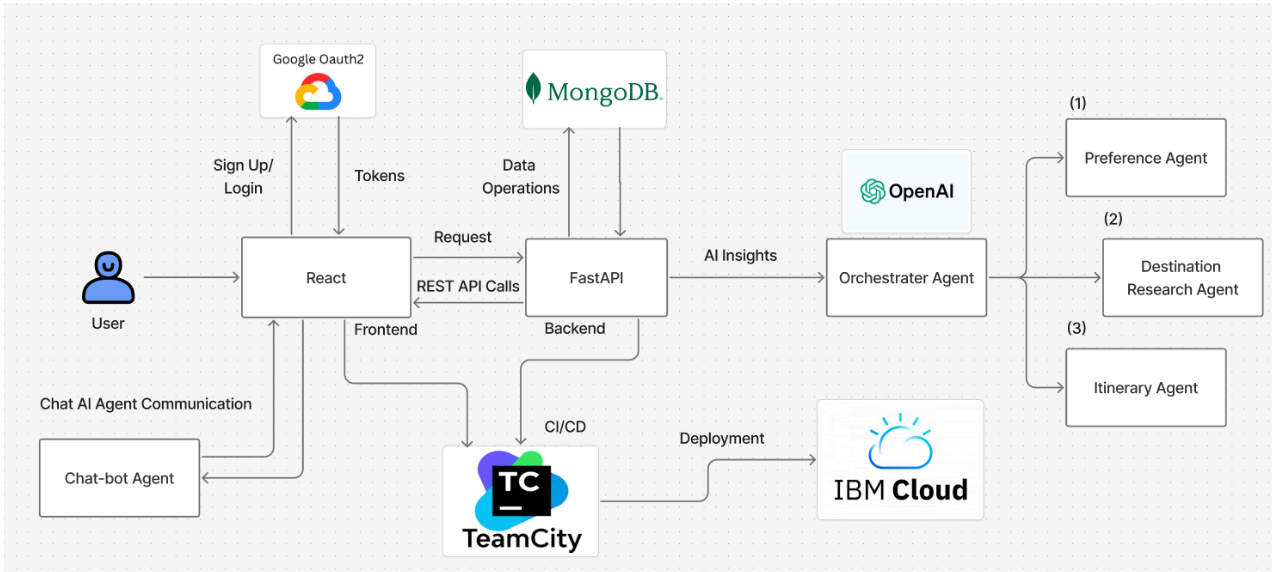
I. INTRODUCTION

In today's fast-paced and efficiency-driven society, organizing group events or trips can be a time-consuming and complex task, particularly for young people with busy schedules. Planning is often designated towards an individual in the friend group who must coordinate the groupmates' time availability, interests, or preferences, as well as maintain a feasible budget. Even after gathering the group's information, they also have to gather ideas and information about the destination for building the travel plan. This process can be both tedious and inefficient, often leading to incomplete, disorganized plans or arguments within friend groups.

The AI-Powered Travel Planner eliminates the chaos of travel planning through an agentic system of centralizing planning that allows the friend group to avoid many of the problems of individually creating a travel plan. The Travel Planner has integrated AI Agents that ensure each part of the final plan is as close to the group's vision of their ideal group vacation as possible.

II. SYSTEM ARCHITECTURE

This system is designed to ensure minimal effort and interaction required in the application, where the agents are tasked to do all the hard thinking for the users. The architecture is split into several components, each serving a specific function:



1. Frontend (React + TypeScript + Tailwind + Socket.io)

The frontend of the system is built with React and TypeScript to ensure a modular, maintainable, and type-safe code. Tailwind CSS is utilized for responsive and consistent user interface design throughout the website, which allows rapid development of visually cohesive components without requiring manual implementation of CSS on each page of the website. The real-time interactions between users and the system are facilitated through Socket.io, which provides a "low-overhead communication channel between the server and the client" (Socket.IO). It enables dynamic updates such as collaborative trip planning and instant synchronization of itinerary changes.

2. Hosting (IBM Cloud)

The website is deployed on IBM Cloud, which provides a scalable and reliable hosting environment for both frontend and backend services. This also allows users to access the website without having to locally host the website on their personal devices. The cloud-based deployment ensures high availability, efficient resource management, and seamless scalability to handle varying user loads.

3. OAuth (Google OAuth)

For secure user authentication, the login and registration are managed through Google OAuth, enabling secure and convenient access using existing Google credentials. This enhances data protection by leveraging industry-standard authentication protocols while minimizing the interactions required to create account information and passwords. By linking the Google account to the website, it also allows future developments to interact with users' Google Calendar and maps.

4. Backend API Service (FastAPI)

The backend is developed with the FastAPI framework, chosen for its high performance and asynchronous capabilities. The backend handles API endpoints, communication between the frontend and external data sources, and connection to the AI agents in the system. The API Service manages the AI-driven itinerary generation, data processing, and integration with third-party APIs, ensuring efficient and secure client-server interactions.

5. Agent API (OpenAI)

   The website utilizes OpenAI to power its AI-driven travel planning features. The orchestrator agent coordinates workflow and routes between agents. The preference agent tracks all of the users' preferences, such as time availability, budgets, interested activities, and any deal breakers. The Destination Research Agent generates a destination-specific activity catalog aligned to the group's preferences. The Itinerary agent organizes the results from the Preferences and Destination Research Agent to create an itinerary plan that fits the time available for the whole group and organizes events that match the group's main preferences for vacation activities. There is also a chatbot AI, "the AI Travel Assistant," that the user is able to interact with to help navigate and interact with the website.

6. CI/CD (TeamCity)

   The team utilizes TeamCity to manage the system for continuous integration and deployment processes. It automates building, testing, and deployment for both the front and backend. It also ensures that each code change is validated before release. The main usage was to allow simultaneous deployment of frontend and backend to the server from updates in the code published on the team's github.

III. FUNCTIONALITIES

The system provides a range of functionalities for improving the process of travel planning for young adults. Each feature is designed to minimize manual coordination, promote collaboration, and enable intelligent itinerary generation through AI automation.

A. Create group rooms to collaborate and interact with the travel plan.

   The system allows users to create dedicated group rooms that serve as shared spaces for planning trips collaboratively. Each room can host multiple participants, enabling friends or acquaintances to contribute to the same plane. The others can join the room through a unique Trip code that is generated while creating the group rooms. This ensures a secure and convenient access to the planning sessions. There is also a chatbox for users to have real-time messaging. Within the room, members can view, discuss, and modify itinerary details in real time.

B. Track users preferences

   Each user has to fill out a preference survey that gathers personal information. The Preference agent processes and aggregates this data utilizing a semantic search and embedding techniques to identify patterns and similarities across all participants. The agent aggregates this data to form a unified preference profile for the group, allowing the system to generate travel suggestions that balance all members' priorities. Examples of tracked preferences include:

- Time availability and preferred Travel Duration
- Activity Interest (E.g., outdoor adventures, cultural attractions, relaxation)
- Budget limitation
- Non-negotiable Constraints

C. Generate Vacation Itinerary

Using the OpenAI API, the system automatically generates a comprehensive and personalized travel itinerary based on the aggregated group preferences. This itinerary includes destination suggestions, accommodation options, and day-by-day activity schedules. The specialized agents work collaboratively to ensure that the generated plan aligns with user expectations. The AI continuously refines results through contextual reasoning, producing an itinerary that is coherent, time-efficient, and budget-conscious.

D. Automatically adjust the Itinerary with real-time status updates.

The system supports dynamic updates to the itinerary based on user feedback or changes in group preferences. When any participant modifies the survey inputs or availability, the AI agent automatically adjusts the itinerary to address the feedback while comparing it to previous preferences and activity data. The Chatbot AI also addresses and accepts changes through the messages to update the itinerary. The travel plans remain synchronized with the group's evolving circumstances.

IV. PERSONA

The system is designed with several young adults within a friend group, each with different opinions regarding their ideal vacation:

A. The Organizer

Organizers are the responsible, detail-oriented, proactive leaders of the friend group. In typical group planning events, they handle the majority of the effort towards developing the itinerary and information tracking. With the AI-Powered Travel Planner, the Organizer can efficiently create and manage travel planning rooms, monitor participation, and use the integrated chatbot to adjust itinerary details. This reduces the manual workload and allows for more structured, collaborative, and transparent coordination.

B. The Easygoing Friend

The Easygoing Friends are the social, spontaneous users who prefer simplicity over detailed planning. They only engage with the system as much as necessary, typically joining the room and completing the initial preferences survey. The platform benefits this group by generating itineraries and presenting them in an accessible format without requiring extensive input. This functionally allows the user to stay informed and involved while avoiding the complexity and time demands of traditional planning processes.

C. The Budget Watcher

Budget Watchers are financially conscious individuals who may face economic limitations, a common concern among young travelers. Their main challenge is finding cost-effective activities and accommodations while ensuring that the rest of the group remains satisfied with the plans. The Travel Planner's AI System helps address this issue by monitoring expenses and generating itineraries that stay within budget constraints, even as activities or preferences change. This ensures transparency, fairness, and financial balance throughout the planning process.

D. The Picky Friend

The Picky Friends represent selective travelers who frequently disagree with group decisions or shift their interests through the planning process. They typically have an ideal vacation in their minds, but struggle to translate it into a feasible plan. These users frequently adjust their preferences, propose changes to the itinerary, and suggest alternatives through the platform. The system accommodates these behaviors using AI-based recommendation filtering and transparent preference tracking, ensuring that their input is valued without disrupting group cohesion. The AI Agentic system continuously processes new inputs, refining the itinerary to balance differing opinions and minimize conflicts that arise during coordination.

V. TECHNOLOGIES USED

A. React and TypeScript (Frontend Development)

React was selected as the frontend development library due to its component-based architecture, reusability, and the team's familiarity with development in it. Combined with TypeScript, the frontend benefits from static typing and improved maintainability, reducing runtime errors during development. The system's interface allows users to create group rooms, input preferences, and view AI-generated itineraries with seamless navigation and real-time updates.

B. FastAPI (Backend Development)

FastAPI was chosen for backend development because of its asynchronous performance, automatic documentation generation, and ease of integration with modern Python libraries. It enables management of API endpoints, user requisitions and allows connections to external AI services and databases. The backend ensured efficient communication between the frontend and the AI agent layer while maintaining low latency and scalability.

C. MongoDB (Database Management)

MongoDB serves as the system's primary database, chosen for its flexible NoSQL schema that efficiently stores unstructured travel data such as user preferences, itineraries, and activity lists. Motor, the asynchronous driver, allows seamless interactions between MongoDB and the FastAPI backend, enabling real-time read/write operations. Its scalability and high availability make it suitable for handling growing user data as usage of the application increases.

D. Google OAuth 2.0 (Authentication and Security)

Google OAuth 2.0 provides secure authentication and authorization for the system, allowing users to log in using existing Google credentials. This eliminates the need for manual password management while ensuring data security through industry-standard OAuth 2.0 protocols. The integration also allows future enhancements to the application, such as linking travel plans for Google Calendar or Maps services for a more connected experience.

E. OpenAI (AI Agent Integration)

OpenAIi powers the system's AI-driven planning agents responsible for the generation of personalized travel recommendations and itineraries. This is integrated through the LangChain and LangGraph frameworks. OpenAIhandles the natural language queries, interprets user preferences, and produces optimized travel plans. The multiple specialized agents, such as the Preference Agent, Destination Research Agent, and Itinerary Agent, collaborate to refine outputs dynamically, ensuring that the generated trip aligns with user goals, time availability, and budget.

F. IBM Cloud and Docker (Deployment and Hosting)

The application is hosted on IBM Cloud, which provides a scalable and reliable infrastructure for both frontend and backend services. Docker was used for containerization, ensuring consistent environments across development and production. The frontend is served through an Nginx Alpine image, which the backend runs on a Python 3.13-slim image using Uvicorn and the ASGI server. This setup enhances portability, scalability, and ease of maintenance.

G. Tailwind CSS (User Interface Styling)

Tailwind CSS was adopted for its utility-first approach to styling, enabling rapid development of a responsive and visually cohesive interface. This also allows group development of the application to flow more seamlessly without having to develop a style guide that each developer must follow. It allows consistent design patterns across all components without requiring repetitive CSS code, contributing to both performance-efficient and maintainable UI design.

VI. TESTING

The AI-Powered Travel Planner underwent comprehensive testing to ensure accuracy, reliability, and overall performance of both the frontend interface and the AI-driven backend components. Two primary technologies were utilized: Testim for frontend and interface validation, and Pytest for backend agent functionality. The testing process included unit tests, integration tests, and automated UI tests to verify that all modules operated according to the design requirements. The following subsections detail the testing strategies applied to each part of the system:

A. Frontend Testing (Testim.io)

For the frontend, Testim was employed to validate the performance, functionality, and general usability of the web interface developed with React and TypeScript. Testim's automated testing framework enabled efficient execution of end-to-end tests across the platform's interactive components.

- User Interface Testing: Automated UI tests were created to ensure that all interface elements (forms, navigation, menus, and real-time updates) function as intended. These tests confirm that the user is able to create planning rooms, generate trip codes, join sessions with the trip code, and update preferences without errors.
- Error Handling and Edge Cases: Tetim scripts simulate user actions such as incorrect form entries, invalid trip codes, or connectivity disruptions to verify that the system can provide appropriate feedback and maintain stability under extreme conditions.
- Cross-Browser and Responsiveness Testing: The application was tested across multiple browsers and screen sizes to ensure consistency in user experience and responsive design behavior, validating the effectiveness of Tailwind CSS implementation.

B. Backend Agent Testing (Pytest)

The backend AI component, developed using FastAPI and integrated with OpenAI, was tested using Pytest to ensure accurate performance and agent reliability. Pytest was selected for its simplicity, modularity, and compatibility with asynchronous API testing.

- Unit Testing of AI Agents: all of the agent modules were tested in isolation to validate their logic, data handling, and output consistency. These tests ensure that each agent responds appropriately to different user preferences and constraints.
  - E.g., all conflicting times, a wide range of favorite activities, etc.
- Integration Testing: End-to-end integration tests were performed to assess the communication between FastAPI backend, the OpenAI API, and the MongoDB database. Mock responses were used to simulate external API calls, verifying that the system could handle real-time agent responses and error conditions effectively.
- Performance and Reliability Testing: Pytest was also used to monitor the agents' processing times and error rates under multiple concurrent user requests. This confirmed that the AI-driven itinerary generation maintained performance stability even under increased workloads.

The combination of Testim and Pytest, the system to achieve robust validation of both the user-facing and the AI-driven components. Automated testing ensured functional correctness, improved reliability, and reduced regression errors during iterative development, resulting in a stable and high-performing application in deployment.

VII. API SECURITY

To ensure secure access to the AI-Powered Travel Planner, the system employs Google OAuth 2.0 v2 with OpenID Connect for user authentication and authorization. This integration provides a reliable and industry-standard mechanism for protecting user credentials, managing access to APIs, and safeguarding sensitive travel-related data. The following are the key aspects of the system's API security strategy:

A. Authentication with Google OAuth 2.0

Google OAuth 2.0 is used to authenticate users through their existing Google accounts, offering a secure and familiar login experience. This approach eliminates the need for storing passwords within the system and leverages Google's identity platform to handle authentication.

- Secure Token Exchange:
  Upon successful authentication, Google issues an OAuth 2.0 access token that is transmitted to the backend API. This token allows the backend to verify the user's identity and confirm that requests originate from a legitimate, authenticated source.
- Access and Refresh Tokens:
  The system uses access tokens to authorize API requests. In the case of a token expiration, a refresh token is used to obtain a new one without requiring the user to log in again. This process ensures session continuity while maintaining a high level of security.

B. Authorization

After successful authentication, the backend validates user authorization by verifying token scopes and claims provided by Google's identity service. These claims determine the level of access a user has within the system.

- Scope-Based Access Control:
  The system enforces scope-based authorization, where only users with valid tokens and proper scopes can access their respective planning rooms and itineraries, while unauthorized users do not have access to those rooms.

C. API Security Best Practices

- Token Validation: Each API request is checked for a valid Google-issued access token. The backend verifies the token's signature, expiration time, and audience to ensure its authenticity and integrity.
- Encryption: all communication between the frontend, backend, and Google's OAuth servers occurs over HTTPS, ensuring that sensitive data is encrypted in transit and protected from interception.
- Token Expiry and Rotation: Access tokens are short-lived to minimize potential misuse. Refresh tokens are securely stored and rotated periodically to further enhance session security and mitigate the risk of token compromise.

D. Logging and Monitoring

The system maintains detailed logs of authentication and API access events in the database to detect potential anomalies or security threats. Suspicious activities, such as repeated failed login attempts or invalid token usage, are flagged for administrative review. Google OAuth's built-in monitoring tools and API activity reports assist in maintaining accountability and compliance.

By integrating Google OAuth 2.0, the AI-Powered Travel Planner ensures that only verified users can access and modify travel plans. This security model upholds confidentiality, integrity, and trust to provide users with a safe and seamless authentication experience while protecting all sensitive data and integrations within the platform.

VIII. CONCLUSION

The AI-powered Travel Planner is an intelligent and collaborative platform designed to simplify and enhance group vacation planning for young adults. Built on a robust system architecture utilizing React and TypeScript for the frontend, FastAPI for the backend, and MongoDB for data management, the system integrates multi-agent AI powered by OpenAI to generate personalized, well-structured travel plans. The platform offers a comprehensive range of functionalities, such as collaborative trip rooms, automated preference aggregation, dynamic itinerary generations with real-time adjustments. These functions ensure users receive tailored and coherent travel recommendations that reflect their collective interests, budgets, and availability. It supports multiple use cases by accommodating diverse user profiles, from organizers to picky travelers, while maintaining clarity, fairness, and convenience throughout the planning process.

The technologies used includes: React, FastAPI, MongoDB, Google OAuth, Tailwind CSS, IBM Cloud, and Docker. They collectively ensure scalability, usability, and secure deployment. The system underwent extensive testing through Tetim for frontend validation and Pytest for backend AI agent verification, achieving reliable performance across user interactions and AI-driven workflow. For security, the website integrates Google OAuth 2.0 and secure token management to ensure strong authentication and authorization, protecting sensitive data and preserving system integrity. In conclusion, the AI-Powered Travel Planner delivers a cohesive, automated, and secure solution that leverages advanced AI and modern web technologies to transform the complexity of group travel planning into an efficient, engaging, and fast user experience.