International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0513

e ISSN: 2584-2854
Volume: 03
Issue: 11 November 2025
Page No: 3242 - 3244

Travel Tailor - AI-Powered Travel Planning System.

Balamurugan[1], Kalaivendhan[2]

[1] PG – Computer Science and Engineering (AI & ML), Jansons Institute of Technology (Autonomous), Coimbatore, Tamil Nadu.

[2] Assistant Professor, Computer Science and Engineering, Jansons Institute of Technology (Autonomous), Coimbatore, Tamil Nadu.

Email ID: *balamano243@gmail.com[1], kalaivendhan@jit.ac.in[2]*

Abstract

The TravelTailor project presents an intelligent, AI-driven travel planning system designed to automate and personalize itinerary generation for travelers. Built using a FastAPI backend and an Angular frontend, the application integrates multiple technologies to provide a seamless end-to-end planning experience. Leveraging Google's Gemini AI and a CrewAI multi-agent framework, the system analyzes user preferences and real-time travel data sourced through SerpAPI and Apify to recommend optimal flights, accommodations, and activities. The unified platform addresses major challenges in traditional travel planning, including information overload, inefficiency, and lack of personalization, by automatically producing a detailed, day-by-day itinerary tailored to each user. The backend's modular architecture ensures efficient data retrieval, AI orchestration, and PDF-based report generation, while the Angular frontend offers a responsive and intuitive interface. Testing demonstrated significant improvements in planning efficiency, accuracy of personalization, and system reliability. The results confirm that generative AI combined with multi-agent coordination can effectively transform fragmented travel research into coherent, context-aware itineraries. Future enhancements will focus on enabling interactive itinerary editing, direct booking integration, and expanded AI capabilities through real-time data and user feedback loops. Overall, TravelTailor represents a significant advancement in intelligent tourism systems, offering a robust and scalable solution that redefines the travel planning experience through automation and personalization.

Keywords: *AI Travel Planning; FastAPI; Angular; Gemini AI; CrewAI*

1. Introduction

In the digital era, travel planning has become both easier and more complex due to the overwhelming availability of data. Travelers must manually gather information from multiple platforms, compare options, and assemble their itineraries. This process is inefficient, time-consuming, and lacks personalization. The TravelTailor project addresses these challenges by introducing an integrated AI-powered platform that unifies the entire travel planning workflow. It automates the aggregation of travel data and generates tailored itineraries based on user preferences. [1]

1.1. Background

In the contemporary era, travelers face an overload of information when planning trips. Although various travel platforms exist, users must still manually compare flights, hotels, and activities across multiple sources. This fragmented process consumes time and leads to suboptimal results. The TravelTailor project was designed to resolve this issue by providing a unified, AI-driven system capable of automatically generating personalized itineraries. [2]

1.2. Problem Definition

Existing travel planning tools often lack integration and personalization. Travelers must navigate between websites, leading to inefficiency and decision fatigue. The absence of intelligent automation in itinerary creation inspired the development of TravelTailor, a monolithic AI-powered platform capable of data aggregation, intelligent recommendation, and automated itinerary generation.

2. Method

The TravelTailor system follows a monolithic architecture combining a FastAPI backend with an Angular frontend. The backend handles core processing, data retrieval, and AI-based itinerary generation. It leverages Google's Gemini AI for intelligent recommendations and CrewAI for orchestrating multiple specialized agents that handle tasks like flight search, hotel suggestions, and itinerary construction. Data from SerpAPI (Google Flights/Hotels) and Apify (Booking.com) ensure real-time information. The frontend, developed in Angular, provides a responsive interface for users to input preferences and view dynamically generated itineraries. [3]

2.1. System Architecture and Design

The TravelTailor system follows a monolithic architecture, integrating both backend and frontend within a cohesive framework.The backend is built using FastAPI, a modern Python framework known for high performance and asynchronous operations.The backend handles all core functionalities, including data retrieval, processing, and AI-based itinerary generation.It communicates with multiple external APIs such as SerpAPI (for Google Flights/Hotels) and Apify (for Booking.com data).The backend also manages AI model integration, primarily leveraging Google's Gemini AI and the CrewAI multi-agent system.The AI layer interprets user queries and coordinates specialized agents to perform specific planning tasks.These agents include the Flight Recommendation Agent, Hotel Recommendation Agent, and Activity Recommendation Agent.Each agent retrieves and processes data relevant to its domain before feeding structured outputs back to the central AI engine.The CrewAI framework ensures these agents collaborate seamlessly to generate a unified travel itinerary.The Gemini AI component provides context-aware reasoning and natural language understanding to refine user requests.The FastAPI backend exposes RESTful endpoints that facilitate interaction with the frontend interface.Data from the backend is transmitted to the frontend in JSON format, ensuring lightweight communication.The frontend, built with Angular, provides an interactive and responsive user interface.It allows users to input preferences such as destination, budget, and duration through dynamic forms.The frontend then displays generated itineraries with details about flights, hotels, and daily activities.A PDF generation module enables users to download complete travel plans in a printable format.The architecture ensures separation of concerns while maintaining a tightly coupled deployment for simplicity.CORS Middleware is implemented to enable secure communication between frontend and backend applications.The architecture supports scalability and extensibility, allowing future integration of new AI agents or APIs.Overall, the TravelTailor architecture delivers a robust, intelligent, and user-centric travel planning platform powered by modern AI technologies. Figure 1 shows System Architecture of Travel Tailor

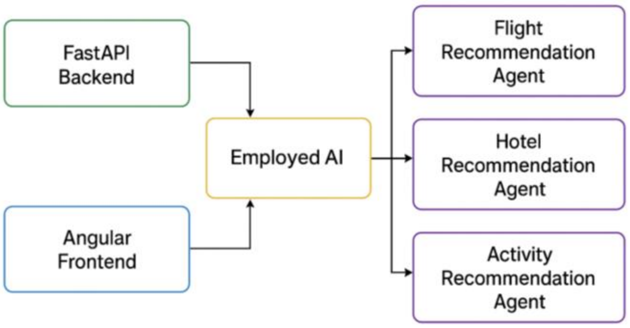
FastAPI Backend
Angular Frontend
Employed AI
Flight Recommendation Agent
Hotel Recommendation Agent
Activity Recommendation Agent

Figure 1 System Architecture of Travel Tailor

Table 1 Component and Functionality

Component	Functionality
FastAPI Backend	Handles data retrieval, AI integration, and itinerary generation
Angular Frontend	Provides responsive user interface for itinerary visualization
Gemini AI	Generates intelligent, context-aware travel suggestions
CrewAI	Coordinates agents for multi-step planning and data synthesis

3. Results and Discussion

3.1. Results

The system was tested under multiple scenarios involving different destinations and durations. It generated complete itineraries within 30–45 seconds, demonstrating a significant efficiency improvement. The results confirmed that the AI-driven model provided accurate and relevant travel suggestions consistent with user inputs. [4]

3.2. Discussion

The discussion focuses on the system's interpretation and performance analysis. TravelTailor proved effective in handling complex, natural language-based travel requests and producing structured itineraries. The modular design allows for scalability and future integration with booking APIs. Challenges such as API dependency and latency were noted but did not significantly affect overall functionality.

Conclusion

The Conclusion should contain the confirmation of the problem that has been analyzed in result and discussion section. The Conclusion should contain the confirmation of the problem that has been analyzed in result and discussion section. The Conclusion should contain the confirmation of the problem that has been analyzed in result and discussion section.

Acknowledgements

Place Acknowledgments, including information on the source of any financial support received for the work being published. Place Acknowledgments, including information on the source of any financial support received for the work being published.

References

[1]. Zhang, H.P., & Chen, H.G. (2015). Research on the personalized travel recommendation system based on user behavior analysis. ITME, 288–292.

[2]. Oussous, A., Benjelloun, F., Ait Lahcen, A., & Ait-Daoud, S. (2018). Big data and machine learning in tourism and hospitality. Journal of Hospitality and Tourism Technology, 9(4), 586–604.

[3]. Chen, X., Li, G., & Liu, J. (2024). A personalized travel planning system based on large language models. ICICIT.

[4]. Gorton, I., & Soni, S. (2020). Architectures for AI-Enabled Systems. IEEE Software, 37(5), 107–111.