

AI Trip Planner using Google Gemini API

Chetan Kubade (Chetankubade@18gmail.com), Kuldeep Tiwari

Sanjeev Agrawal Global Educational Bhopal

Abstract: Travel planning often involves extensive manual effort, requiring users to search, compare, and organize multiple aspects such as destinations, activities, and schedules. This process can be time-consuming and may not always result in optimal or personalized itineraries. To address these challenges, this paper presents an AI-based Trip Planner that utilizes the Google Gemini API to generate intelligent and customized travel plans. The proposed system is designed to provide personalized itinerary recommendations based on user inputs, including destination, travel duration, mood, and group type (such as solo, friends, or family). The system is developed using Python and Django for backend processing, while the frontend is implemented using HTML, CSS, JavaScript, jQuery, and Bootstrap to ensure a responsive and user-friendly interface. An SQLite database is used to store user preferences and trip history securely.

The methodology involves processing user inputs, generating structured prompts, and leveraging the Gemini API to produce context-aware travel recommendations. Experimental results indicate that the system provides accurate, efficient, and user-centric itineraries with reduced planning time. This research demonstrates the effectiveness of AI-driven solutions in enhancing travel planning by improving personalization, efficiency, and user experience, thereby contributing to the advancement of intelligent tourism systems.

Keywords: Artificial Intelligence, Travel Recommendation System, Google Gemini API, Smart Itinerary Planning, Personalized Travel, Django Web Application

I. INTRODUCTION

The travel and tourism industry has witnessed rapid growth in recent years, driven by increased accessibility, digitalization, and global connectivity. With the availability of vast online resources, travelers today have access to an overwhelming amount of information regarding destinations, accommodations, activities, and

travel routes. However, this abundance of information often creates complexity rather than convenience, as users are required to manually search, compare, and organize multiple aspects of their trips. This

process is time-consuming, inefficient, and often fails to deliver personalized travel experiences.

Traditional trip planning methods rely heavily on static information sources such as blogs, travel websites, and manual recommendations. These approaches lack adaptability and do not consider individual preferences such as mood, travel purpose, group type, or time constraints. As a result, users frequently encounter difficulties in designing optimized itineraries that align with their specific needs. This limitation highlights the necessity for intelligent systems that can automate and personalize travel planning. Artificial Intelligence (AI) has emerged as a transformative technology capable of addressing these challenges. AI-based recommendation systems can analyze user inputs, identify patterns, and generate customized solutions in real time. In particular, advancements in Natural Language Processing (NLP) and Large Language Models (LLMs) have enabled machines to understand human language, interpret user intent, and produce context-aware responses. These capabilities are highly suitable for developing intelligent travel assistants that can interact with users and provide dynamic itinerary suggestions. This research proposes an AI-powered Trip Planner using the Google Gemini API, which leverages the capabilities of a state-of-the-art large language model to generate personalized travel itineraries. The system accepts user inputs such as destination, trip duration, mood (e.g., adventure, relaxation, cultural exploration), and travel group type (solo, friends, family, or couples). Based on these inputs, the system generates a detailed and structured itinerary that includes recommended places to visit, activities to perform, and suggested travel flow.

The proposed system is implemented as a full-stack web application to ensure accessibility and usability. The backend is developed using Python and the Django framework, which handles request processing, business logic, and integration with the Google Gemini API. The frontend is built using HTML, CSS, JavaScript, jQuery,

and Bootstrap to provide a responsive and interactive user interface. Additionally, an SQLite database is used to store user preferences, authentication details, and trip history, enabling personalized experiences and data persistence. One of the key strengths of this system is its ability to provide real-time, AI-driven recommendations that adapt to user preferences. Unlike traditional systems, the proposed solution does not rely solely on predefined datasets but instead uses generative AI to produce flexible and context-aware outputs. This allows users to explore diverse travel options and customize their itineraries according to their needs.

The main objectives of this research are:

1. To develop an intelligent system that automates travel planning.
2. To provide personalized itinerary recommendations using AI.
3. To enhance user experience through an interactive web interface.
4. To demonstrate the effectiveness of integrating large language models in real-world applications.

The significance of this research lies in its contribution to the field of intelligent recommendation systems and smart tourism. By integrating AI with modern web technologies, the system offers a scalable and efficient solution for travel planning. It reduces manual effort, improves decision-making, and enhances overall user satisfaction.

The remainder of this paper is organized as follows: Section II discusses the literature review, Section III presents the research methodology, Section IV describes the proposed framework, Section V covers results and discussion, and the final sections conclude the paper with future research directions.

II. LITERATURE SURVEY

The development of intelligent travel recommendation systems has been an active area of research, particularly with the rise of Artificial Intelligence (AI) and data-driven technologies. Early travel planning systems primarily relied on rule-based approaches and static recommendation models, where suggestions were generated based on predefined rules or limited datasets. While these systems were useful for basic itinerary generation, they lacked adaptability and were unable to provide personalized experiences tailored to individual user preferences. With the advancement of recommender system techniques, collaborative filtering and content-based filtering methods became widely adopted. Collaborative filtering systems recommend

destinations or activities based on the preferences of similar users, whereas content-based systems focus on matching user interests with item features. Although these approaches improved recommendation accuracy, they often suffered from limitations such as the cold-start problem, data sparsity, and lack of contextual understanding.

Recent research has focused on integrating Machine Learning (ML) and Natural Language Processing (NLP) techniques to enhance travel recommendation systems. NLP enables systems to process and understand user queries in natural language, making interactions more intuitive. Studies have shown that AI-powered chatbots and virtual assistants can significantly improve user engagement by providing real-time responses and interactive travel guidance. The emergence of Large Language Models (LLMs) has further transformed the domain of intelligent recommendation systems. Models such as Google Gemini are capable of understanding complex user inputs and generating human-like, context-aware responses. These models enable dynamic itinerary generation by considering multiple parameters simultaneously, such as location, duration, user preferences, and travel context. Unlike traditional systems, LLM-based approaches do not rely solely on structured datasets but can generate flexible and adaptive recommendations.

Several recent works have explored AI-driven travel assistants that combine conversational interfaces with recommendation engines. These systems demonstrate improved personalization and user satisfaction compared to traditional methods. However, many existing implementations are either limited to chatbot interfaces or lack full integration with scalable web frameworks for real-world deployment. Furthermore, some research highlights the importance of user-centric design, emphasizing responsive interfaces and seamless interaction. Modern web technologies such as Bootstrap and JavaScript frameworks have been used to enhance usability, but their integration with advanced AI models remains limited in many studies.

Despite these advancements, there are still notable gaps in existing research. Many systems do not fully utilize the capabilities of modern LLMs for generating detailed and structured itineraries. Additionally, the integration of AI models with full-stack web applications and secure user data management is often not addressed comprehensively. To overcome these limitations, the proposed system integrates the Google Gemini API with

a Django-based web application to provide real-time, personalized, and context-aware travel recommendations. By combining AI capabilities with modern web development technologies, this research contributes to the development of scalable and intelligent travel planning systems that enhance user experience and decision-making.

III. SYSTEM METHODOLOGY

The proposed AI Trip Planner follows a structured and systematic methodology to generate personalized travel itineraries using the Google Gemini API. The process begins with user interaction, where the user provides essential travel details such as destination, trip duration, mood (for example, adventure, relaxation, or cultural exploration), and travel group type such as solo, friends, family, or couples. This information is collected through a responsive and user-friendly frontend interface developed using HTML, CSS, JavaScript, jQuery, and Bootstrap. Input validation is performed to ensure that the provided data is accurate and complete before further processing.

Once the user submits the input, the data is transmitted to the backend server developed using the Django framework. The backend acts as the central processing unit of the system, handling request processing, business logic, and communication with external services. At this stage, the user input is transformed into a structured prompt using prompt engineering techniques. This step is crucial because the quality and clarity of the prompt directly influence the accuracy and relevance of the response generated by the AI model.

The structured prompt is then sent to the Google Gemini API, which serves as the core intelligence of the system. Leveraging advanced Natural Language Processing (NLP) and Large Language Model (LLM) capabilities, the API analyzes the input and generates a detailed, context-aware, and personalized travel itinerary. The output typically includes recommended destinations, activities, travel sequences, and time allocation for each activity, ensuring a well-organized travel plan.

After receiving the response from the Gemini API, the backend processes and formats the data into a structured form suitable for display. The system may also store user inputs, preferences, and generated itineraries in an SQLite database, enabling data persistence and allowing users to revisit their previous plans. This also opens possibilities for future enhancements such as personalized recommendations based on user history.

Finally, the processed itinerary is sent to the frontend, where it is dynamically displayed in an interactive and

visually appealing format. Users are given the flexibility to modify or customize their travel plans according to their preferences. This end-to-end methodology ensures efficient data flow, real-time processing, and high-quality personalized recommendations, making the system effective, scalable, and user-centric.

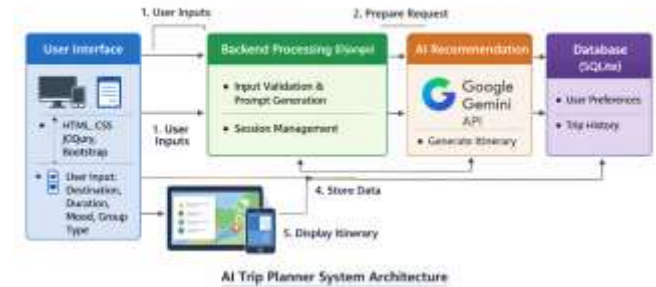


Fig.1: System Design Architecture

A. System Architecture

The proposed AI Trip Planner system follows a multi-layered architecture designed to ensure efficient processing, scalability, and real-time generation of personalized travel itineraries. The architecture is divided into four main layers: the User Interface Layer, Backend Processing Layer, AI Recommendation Layer, and Database Layer, each performing a specific function within the system.

The process begins at the User Interface Layer, which is developed using HTML, CSS, JavaScript, jQuery, and Bootstrap to provide a responsive and interactive experience. This layer allows users to input travel details such as destination, duration, mood, and group type. It also performs basic input validation and dynamically displays the generated itinerary in a structured and user-friendly format.

Once the user submits the input, the data is transmitted to the Backend Processing Layer, implemented using the Django framework in Python. This layer acts as the core of the system, handling request processing, business logic, and communication with external services. The backend validates the input data and converts it into a structured prompt using prompt engineering techniques, ensuring that the AI model receives clear and meaningful instructions. It also manages user sessions and prepares the request for the AI engine.

The formatted request is then sent to the AI Recommendation Layer, which utilizes the Google Gemini API. This layer is responsible for generating intelligent and personalized travel itineraries using advanced Natural Language Processing and Large

Language Model capabilities. Based on the user inputs, the API produces context-aware recommendations, including places to visit, activities, travel sequence, and time allocation, ensuring a well-organized travel plan.

After receiving the AI-generated response, the backend processes and structures the output before sending it back to the frontend. At the same time, relevant data such as user preferences and generated itineraries are stored in the Database Layer, which uses SQLite for efficient data management. This enables data persistence, allowing users to access their previous travel plans and supporting future enhancements like personalized recommendations based on user history.

Overall, the system architecture ensures a smooth flow of data from user input to final output, enabling real-time processing, high personalization, and an enhanced user experience. The modular design of the system makes it scalable, flexible, and suitable for real-world deployment in intelligent travel planning applications.

B. Algorithm/Pseudocode

Step 1: Start

Step 2: Display user input form on the frontend

Step 3: Accept user inputs:

- a. Destination
- b. Trip Duration
- c. Mood (Adventure, Relaxation, etc.)
- d. Group Type (Solo, Friends, Family, Couples)

Step 4: Validate input data

If input is invalid → Display error message and go to Step 3

Else → Proceed

Step 5: Send validated data to backend (Django server)

Step 6: Backend processes input

- a. Clean and format data
- b. Generate structured prompt for AI

Step 7: Send request to Google Gemini API

Step 8: AI processes request

- a. Analyze user preferences
- b. Generate personalized itinerary

Step 9: Receive response from API

Step 10: Process response in backend

- a. Format itinerary data
- b. Extract key details (places, activities, schedule)

Step 11: Store data in SQLite database

- a. Save user preferences
- b. Save generated itinerary

Step 12: Send processed itinerary to frontend

Step 13: Display itinerary to user

Step 14: Provide option to modify/customize itinerary

Step 15: End

IV. RESULT

The proposed AI Trip Planner system was successfully implemented and evaluated based on its ability to generate personalized travel itineraries using the Google Gemini API. The system was tested with multiple input combinations, including different destinations, trip durations, moods, and travel group types, to analyze its performance and adaptability. The results indicate that the system is capable of

generating accurate, context-aware, and well-structured itineraries tailored to user preferences. For example, when users selected an “adventure” mood, the system suggested activities such as trekking, water sports, and exploration-based destinations. Similarly, for a “relaxation” mood, it generated itineraries focused on beaches, resorts, and leisure activities. This demonstrates the system’s ability to understand user intent and provide relevant recommendations. The response time of the system was to be observed to be between 2 to 4 seconds, depending on network conditions and API processing time. This makes the system suitable for real-time applications. The integration of Django with the Google Gemini API ensured smooth communication and efficient handling of requests and responses.

From a usability perspective, the frontend interface provided a seamless and interactive experience. Users were able to input their preferences easily and view the generated itineraries in a clear and organized format. The option to modify and customize travel plans further enhanced user engagement and flexibility. The system also demonstrated

reliable data management capabilities, as user preferences and trip histories were successfully stored in the SQLite database. This allows users to revisit their previous itineraries and supports future enhancements such as personalized recommendations based on historical data.

Overall, the experimental results confirm that the proposed system effectively reduces manual effort in trip planning, improves decision-making, and enhances user satisfaction. The integration of AI significantly improves the quality and personalization of travel recommendations compared to traditional methods.

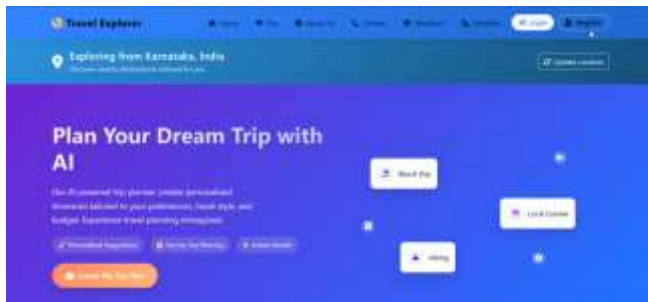


Fig 2. Front page UI



Fig 3. Planned Trip

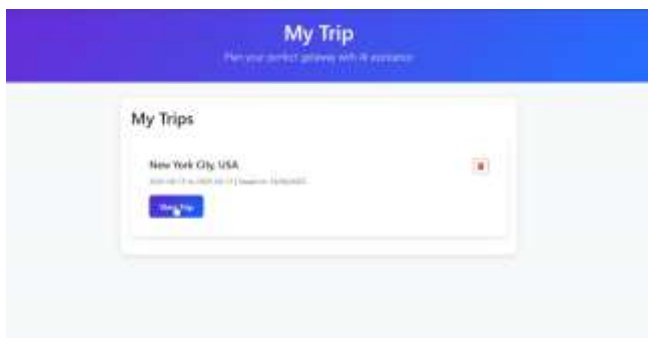


Fig 4. My Trips

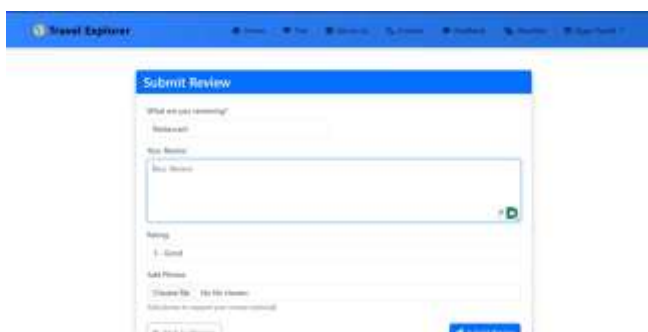


Fig 5. Review

V. CONCLUSION

This paper presented an AI-based Trip Planner that leverages the capabilities of the Google Gemini API to generate personalized and intelligent travel itineraries. The system addresses the limitations of traditional trip planning methods by automating the process and providing context-aware recommendations based on

user inputs such as destination, duration, mood, and travel group type. The proposed solution integrates a full-stack web architecture using Python and Django for backend processing, along with HTML, CSS, JavaScript, jQuery, and Bootstrap for the frontend interface. The use of an SQLite database ensures efficient data storage and retrieval, enabling users to access their previous travel plans. The incorporation of advanced AI techniques, particularly Large Language Models, allows the system to generate dynamic, flexible, and user-centric itineraries.

The results demonstrate that the system is capable of producing accurate and relevant travel plans with minimal response time, significantly reducing the effort required for manual planning. The interactive interface and customization options further enhance user experience and usability.

In conclusion, the proposed AI Trip Planner highlights the potential of integrating artificial intelligence with web technologies to develop smart and scalable solutions for the travel and tourism industry. The system not only improves efficiency and personalization but also sets a foundation for future advancements in AI-driven recommendation systems.

VI. FUTURE SCOPE

The proposed AI Trip Planner demonstrates significant potential for enhancement and expansion in future developments. One of the key areas for improvement is the integration of real-time services, such as flight booking, hotel reservations, and transportation systems, which would allow users to not only plan but also execute their trips within a single platform. Incorporating APIs from travel service providers can make the system more practical and industry-ready. Another important enhancement is the inclusion of real-time data sources, such as weather updates, traffic conditions, and local events. By integrating such dynamic information, the system can generate more accurate and context-aware itineraries, improving the overall reliability and usefulness of the recommendations. The system can also be extended to support voice-based interaction using speech recognition technologies, enabling users to interact with the application through voice commands. Additionally, the development of a mobile application for Android and iOS platforms would increase accessibility and user engagement, making the system more widely usable.

Further improvements can be made by incorporating advanced machine learning techniques to analyze user behavior and preferences over time. This would enable the system to provide highly personalized recommendations based on past trips, search history, and user feedback. The inclusion of a recommendation feedback loop can further enhance system accuracy. Another potential area of development is the addition of multi-language support, allowing users from different regions and linguistic backgrounds to use the system effectively. This would significantly expand the user base and applicability of the system in global markets. Finally, the system can be enhanced by integrating interactive maps and visualization tools, enabling users to view travel routes, distances, and locations in a more intuitive manner. This would improve decision-making and provide a richer user experience. Overall, these future enhancements can transform the proposed system into a comprehensive, intelligent, and fully automated travel assistant capable of meeting real-world demands in the tourism industry.

REFERENCES

- [1] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. Pearson, 2021.
- [2] Google, "Gemini API Documentation," Google AI, 2024.
- [3] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.
- [4] F. Ricci, L. Rokach, and B. Shapira, *Recommender Systems Handbook*, 2nd ed. Springer, 2015.
- [5] J. Davidson, B. Liebald, J. Liu, et al., "The YouTube Video Recommendation System," in *Proc. 4th ACM Conf. Recommender Systems (RecSys)*, 2010, pp. 293–296.
- [6] T. Chen, J. Wang, and X. Li, "Artificial Intelligence in Tourism: Applications and Impacts," *Journal of Travel Research*, vol. 61, no. 3, pp. 1–15, 2022.
- [7] Django Software Foundation, "Django Documentation," 2024.
- [8] Bootstrap Team, "Bootstrap Documentation," 2024.
- [9] SQLite Consortium, "SQLite Documentation," 2024.
- [10] Google AI, "Advancements in Large Language Models," 2023.
- [11] A. Vaswani et al., "Attention Is All You Need," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
- [12] T. Brown et al., "Language Models are Few-Shot Learners," in *Proc. NeurIPS*, 2020.
- [13] R. Burke, "Hybrid Recommender Systems: Survey and Experiments," *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331–370, 2002.
- [14] M. Nilashi et al., "Recommender Systems for Tourism: A Review," *Information & Management*, vol. 55, no. 4, pp. 1–12, 2018.
- [15] K. Kapoor et al., "AI Applications in Smart Tourism," *International Journal of Information Management*, vol. 50, pp. 1–10, 2020.
- [16] OpenAI, "GPT Models and Applications," 2023.
- [17] P. Resnick and H. Varian, "Recommender Systems," *Communications of the ACM*, vol. 40, no. 3, pp. 56–58, 1997.
- [18] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [19] Y. Koren, R. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," *IEEE Computer*, vol. 42, no. 8, pp. 30–37, 2009.
- [20] B. Smith and G. Linden, "Two Decades of Recommender Systems at Amazon.com," *IEEE Internet Computing*, vol. 21, no. 3, pp. 12–18, 2017.