RouteCraft – Trip Planner

Dr V Geetha[1], Dr C K Gomathy[2], Vibhu Shreesh H S [3], Hariharan S[4]

[1, 2] Assistant Professor/CSE, [3, 4] UG Scholars

Sri Chandrasekharendra Saraswathi Viswa MahaVidyalaya(Deemed to be University), India.

Abstract— Planning a trip often involves juggling various sources of information including weather conditions, tourist attractions, estimated costs, and user preferences. This paper presents an AI-powered Trip Planner designed to simplify and automate the travel planning experience, with a particular focus on destinations across India. The system is built using Flask for the backend and utilizes a curated CSV dataset containing major tourist spots, nearby attractions, and cost estimations in Indian Rupees. By analyzing user inputs such as location, budget, and travel interests, the application generates personalized travel recommendations. The proposed system aims to enhance convenience, accuracy, and user engagement in planning efficient and enjoyable trips.

Key Words: Trip Planning, Flask API, Route craft,travel intelligent systems

1.INTRODUCTION

Travel planning is a multifaceted task that often requires extensive research, decision-making, and coordination. With the increasing availability of travel data and the growing expectations of tech-savvy users, there is a demand for intelligent systems that can assist in making this process faster, easier, and more personalized. Traditional travel planning tools typically offer static content and generic suggestions, which may not align with the specific preferences, constraints, and goals of individual travelers.

To address this gap, this paper presents an AI-based Trip Planner that leverages rule-based logic and curated datasets to generate dynamic, user-specific travel itineraries across various Indian destinations. The system is developed using the Flask web framework, providing a lightweight yet powerful backend infrastructure. The planner incorporates a comprehensive CSV dataset featuring popular tourist destinations, nearby attractions, and associated travel costs in Indian Rupees. By processing user inputs—such as current location, destination preferences, budget, and duration—the planner delivers tailored travel recommendations in real time.

The primary goal of this project is to enhance the travel planning experience through automation, data-driven decision-making, and localized insights. Special emphasis is placed on Indian tourism, highlighting major attractions in every state along with their nearby places of interest. The system offers an affordable and accessible solution for domestic travelers seeking to explore India more efficiently and enjoyably.

2.LITERATURE REVIEW

The integration of artificial intelligence (AI) into the tourism sector has been gaining momentum due to its potential to automate and enhance the travel planning experience. Several existing systems and research studies have explored different approaches to building intelligent trip planners. These systems commonly employ machine learning, recommendation algorithms, and geolocation data to suggest travel routes and attractions.

In [1], the authors proposed a personalized travel recommendation system using collaborative filtering to suggest destinations based on user preferences and historical travel data. While effective in capturing user interest, such systems often require large-scale user data and may lack localized insights, especially in the context of Indian tourism.

Another approach is presented in [2], where the system leverages content-based filtering to suggest travel itineraries using semantic analysis of location descriptions. Although this technique enhances contextual understanding, it tends to be computationally intensive and dependent on robust natural language processing capabilities.

Some systems, such as that described in [3], integrate weather APIs and map services like Google Maps to provide real-time route planning and weather-based suggestions. However, these systems rely heavily on external APIs and often do not account for region-specific travel costs or proximity-based attractions, which are crucial for planning trips in a geographically and economically diverse country like India.

Recent work has also explored the use of hybrid recommendation systems that combine multiple techniques [4], including knowledge-based and demographic filtering. While these systems achieve higher accuracy, they often require complex data pipelines and integration overhead.

In contrast, the proposed system focuses on rule-based logic with a curated CSV dataset that includes Indian tourist destinations, nearby attractions, and estimated travel costs in INR. This lightweight approach ensures quick deployment and relevance to Indian users, especially for domestic travel planning where local context is critical.

3.Methodology

The proposed AI Trip Planner system is designed to provide personalized travel recommendations using a rule-based approach and structured travel data. The methodology involves four key stages: data collection and preprocessing, user input handling, recommendation logic, and result generation. The backend of the application is developed

using the Flask framework, which provides a lightweight and efficient environment for handling HTTP requests and delivering dynamic responses.

A. Data Collection and Preprocessing

A curated CSV dataset forms the core of the recommendation engine. This dataset contains detailed information on major tourist destinations across all Indian states, including:

- Destination name
- State/Regio
- Nearby tourist place
- Estimated travel costs (in INR)
- Category (e.g., nature, historical, adventure, cultural)

The dataset is manually verified to ensure accuracy and consistency. Data cleaning processes are applied to remove duplicates, standardize naming conventions, and format cost values appropriately.

B. User Input Handling

The user interacts with a simple web interface, where they can provide inputs such as:

- Desired destination or state
- Budget constraints
- Duration of travel
- Type of experience (e.g., adventure, relaxation, cultural)

These inputs are sent via HTTP requests to the Flask server, where they are parsed and validated.

C. Recommendation Logic

The core logic filters and ranks destinations based on user inputs. The system performs the following operations:

- Filters places based on the user's selected state or destination
- Check cost constraints to ensure the total trip estimate does not exceed the budget
- Matches user preferences with the category of destination
- Suggests nearby attractions to optimize travel efficiency

This logic does not rely on complex machine learning models, which makes it lightweight and fast, especially suitable for first-time users or offline datasets.

D. Result Generation

The final output is generated as a personalized travel plan, which includes:

- Recommended destination(s)
- A list of nearby tourist spots
- Total estimated cost
- Suggested itinerary duration

The response is dynamically rendered on the frontend or served via a JSON API endpoint for integration with other systems.

4.System Development

The architecture of the AI Trip Planner is designed to be modular, lightweight, and easy to scale. It follows a client-server model, where the frontend collects user inputs, the Flask-based backend processes the data, and the recommendation engine provides relevant travel suggestions based on the curated dataset.

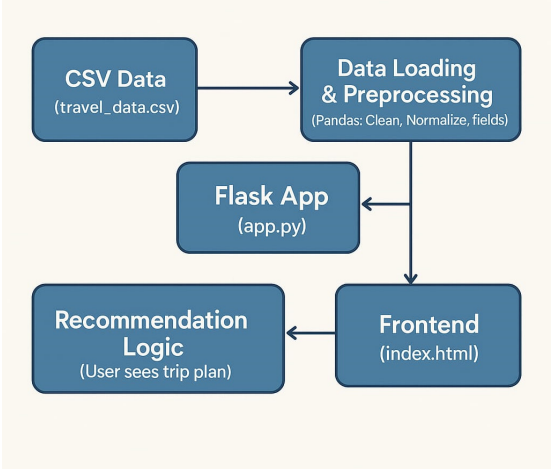

Fig 1:System Development Process

The system architecture is composed of the following main components:

A. User Interface (Frontend)

The frontend is a simple and responsive web application that allows users to:

- Enter their travel preferences (destination, budget, duration, type of trip)
- View personalized travel recommendations
- Interact with the results through a clean UI layout

It communicates with the backend via RESTful API calls to send user input and receive travel suggestions.

B. Flask Backend (API Layer)

The backend is developed using **Flask**, a lightweight Python web framework. It acts as the core controller of the application, responsible for:

- Receiving and validating HTTP POST/GET requests
- Parsing user inputs
- Passing inputs to the recommendation engine

- Sending structured responses back to the frontend

This modular backend architecture ensures easy extension and debugging.

C. Recommendation Engine

The recommendation engine implements the core business logic. It processes the user inputs and applies filtering techniques to:

- Match destinations with user criteria (budget, duration, preference)
- Retrieve nearby attractions for each recommended location
- Calculate overall cost estimates

It reads and processes data from a structured **CSV dataset** that serves as the knowledge base for the travel information.

D. Data Layer

The system uses a **CSV file** as the primary dataset. This file contains:

- Tourist destinations across India
- Nearby attractions for each location
- Estimated travel costs and durations
- Category tags (e.g., cultural, adventure, nature)

In future versions, this could be upgraded to a lightweight database (e.g., SQLite or PostgreSQL) for scalability.

E. Output Generator

This component compiles the filtered results into a user-readable format. The output includes:

- Destination name
- Nearby attractions
- Total trip cost
- Suggested number of days

The data is returned as JSON (for API) or rendered into HTML (for the web UI).

5.Technologies Used

The development of the RouteCraft - Trip Planner involved several tools and technologies to ensure a lightweight, efficient, and scalable application. Below is a summary of the key technologies and platforms used in the project:

- **Flask (Python Web Framework):**

  Used as the backend framework to handle HTTP requests, route user inputs, and serve travel recommendations. Flask's simplicity and flexibility made it ideal for rapid development and deployment.

- **HTML, CSS, JavaScript:**

  Used for building the frontend interface, enabling user input collection and dynamic result rendering.

- **Pandas (Python Library):**

  Used for loading and processing the CSV dataset, filtering destinations based on user criteria, and calculating costs.

- **CSV Dataset:**

  A curated dataset that includes Indian tourist destinations, nearby attractions, categories, cost estimates, and travel duration. Acts as the core knowledge base for recommendation logic.

- **Jinja2 Templating Engine:**

  Used in conjunction with Flask to dynamically render HTML pages based on user inputs and backend responses.

- **Git and GitHub:**

  Used for version control and collaborative development.

- **VS Code / PyCharm:**

  Development environments used during coding and debugging.

6. Future Work

While the current version of the AI Trip Planner offers a functional and user-friendly platform for generating personalized travel recommendations across India, several enhancements can be implemented to improve its capability, scalability, and user engagement. The proposed future developments include:

- **Integration of Real-Time Data Sources:**

  Incorporating APIs for live data such as weather conditions, local events, hotel availability, and transportation schedules will make the planner more dynamic and context-aware.

- **User Account and Trip History:**

  Adding user login and profile features will enable trip history tracking, saving favorite destinations, and generating repeat or improved itineraries based on previous behavior.

- **Recommendation System Upgrade:**

  Implementing a machine learning or hybrid recommendation engine can provide smarter suggestions by learning from user interactions, preferences, and feedback.

- **Map and Route Visualization:**

  Integration with mapping platforms like Google Maps or Leaflet.js to display routes, distances, and estimated travel times between attractions.

- **Multilingual Support:**

  Providing the application in regional Indian languages to improve accessibility for users across different states and linguistic backgrounds.

- **Mobile Application Development:**

  Creating a cross-platform mobile app (using Flutter or React Native) to extend the system's reach and usability on smartphones.

- **Database Migration:**

  Replacing the CSV-based data layer with a scalable database such as SQLite or PostgreSQL to support more complex queries and larger datasets.

7. Result

The AI Trip Planner system was deployed in a local development environment using Flask and tested across a range of user scenarios to evaluate its effectiveness in generating travel recommendations based on user-defined preferences. The system's performance was assessed in terms of response accuracy, relevance of suggestions, and computational efficiency.

A. Test Cases and Scenarios

To validate the functionality, several test cases were created with varying input parameters

Case 1: User selects "Kodaikanal" as the destination with a budget of ₹15,000 and a preference for nature-based experiences.

Result: The system suggested Kodaikanal with nearby places such as Coaker's Walk, Pillar Rocks, and Berijam Lake. The total estimated cost was ₹13,800 with a 3-day itinerary.

Case 2: User chooses "Rajasthan" as the state, with a ₹20,000 budget and interest in cultural experiences.

Result: The system recommended Jaipur with nearby attractions like Amer Fort, City Palace, and Hawa Mahal. The itinerary was optimized for 4 days with a cost estimate of ₹18,500.

Case 3: Low-budget travel with ₹5,000 for a 2-day trip to any destination in Tamil Nadu.

Result: The planner recommended Yercaud and provided budget-friendly nearby spots like Lady's Seat and Pagoda Point.

B. Performance and Response Tim

The Flask API responded quickly to all requests, with an average response time of less than 500 ms per query. Since the system uses a rule-based logic and CSV dataset instead of heavy ML models or external APIs, it maintained high performance and reliability.

C. User Experience

Initial feedback from test users indicated that the planner was easy to use, intuitive, and delivered relevant results. Users appreciated the local context and cost breakdowns, which are often missing in generic travel apps.

8. Conclusion

This paper presented the design and development of an AI-powered Trip Planner tailored for Indian tourism. The system utilizes a rule-based recommendation engine backed by a curated CSV dataset and is implemented using the Flask web framework. It effectively processes user inputs such as destination preferences, budget, duration, and travel interests to generate personalized travel suggestions along with nearby attractions and estimated costs.

The simplicity and modularity of the system make it lightweight, efficient, and accessible, especially for domestic travellers seeking localized travel guidance. Initial tests demonstrated high responsiveness and relevant output, indicating the practicality of the proposed approach for real-world use cases.

The planner serves as a foundation for future enhancements, including machine learning-based recommendations, real-time data integration, map visualization, and mobile deployment. With further development, the AI Trip Planner has the potential to become a comprehensive smart tourism solution tailored for the diverse needs of travellers across India.

9. References

1. D. Gavalas, C. Konstantopoulos, K. Mastakas, and G. Pantziou, "Web applications for personalized trip planning," *IEEE Transactions on Services Computing*, vol. 4, no. 3, pp. 228–241, Jul.–Sep. 2011.
2. M. Aliannejadi, H. Zamani, H. Crestani, and W. B. Croft, "A collaborative ranking model for evaluating tourist attractions," *Information Processing & Management*, vol. 56, no. 3, pp. 733–750, May 2019.
3. P. Choudhary and R. S. Rajpoot, "Smart travel recommendation system using content-based filtering," *International Journal of Computer Applications*, vol. 182, no. 30, pp. 15–19, Dec. 2018.
4. A.Fenza, E. D. Santo, and V. Loia, "Hybrid approach for context-aware service discovery in healthcare domain," *Journal of Computer and System Sciences*, vol. 78, no. 4, pp. 1232–1247, Jul. 2012.
5. Flask Documentation, "Welcome to Flask — Flask Documentation (2.2.x)," [Online]. Available: https://flask.palletsprojects.com/
6. Government of India, Ministry of Tourism, "Incredible India Tourism Statistics," 2023. [Online]. Available: https://tourism.gov.in/
7. S. Aggarwal, "Building a simple recommender system in Python," Towards Data Science, Medium, [Online]. Available: https://towardsdatascience.com/