APSCHE Long Term VIP Program

PROJECT REPORT
ON
TRAVEL GUIDE AI
CUSTOM ITINERARIES FOR YOUR NEXT JOURNEY

TEAM ID : LTVIP2026TMIDS79519
NAME : INNI TANINKI (Team Leader)
Email : innitaninki@gmail.com

CONTENTS

1. INTRODUCTION

1.1 Project Overview

TravelGuideAI is an AI-powered web application designed to generate personalized travel itineraries using Google's Gemini Generative AI model. The system enables users to enter key travel details such as destination, number of days and nights, and personal interests. Based on these inputs, the application dynamically generates a structured, day-wise itinerary that includes recommended tourist attractions, local food suggestions, and practical travel tips.

The application integrates Streamlit as the user interface framework and Google's Gemini API for AI-based content generation. By combining an intuitive frontend with advanced generative AI capabilities, the system provides users with customized and detailed travel plans in real time.

1.2 Purpose

The primary purpose of this project is to develop an intelligent travel planning assistant that simplifies and enhances the trip planning process. The objectives of the project include:

- To simplify travel planning through automation
- To provide personalized and dynamic travel itineraries
- To reduce the time and effort required for manual research
- To demonstrate real-world integration of Generative AI in web applications
- To showcase the practical implementation of AI-powered solutions for solving everyday problems

The project aims to bridge the gap between traditional travel planning methods and modern AI-driven solutions, making itinerary creation faster, smarter, and more user-centric.

2. IDEATION PHASE

2.1 Problem Statement

Travel planning is a complex and time-consuming process. Travelers often need to explore multiple websites, blogs, and review platforms to gather information about tourist attractions, accommodation options, local cuisine, transportation, and travel tips. This scattered information makes the planning process overwhelming and inefficient.

Similarly, travel agencies spend considerable time designing customized itineraries for clients with diverse preferences and constraints. Preparing detailed and personalized travel plans manually can reduce productivity and delay service delivery.

Therefore, there is a need for an intelligent, automated system that can generate personalized travel itineraries quickly and efficiently using Artificial Intelligence. Such a system should reduce manual effort, enhance user experience, and provide customized travel recommendations based on user preferences.

2.2 Empathy Map Canvas

To better understand user needs, an empathy map was created to analyze user thoughts, feelings, actions, and expectations during travel planning.

User Thinks:

- “How can I plan everything within limited time?”
- “What are the must-visit attractions?”
- “Am I missing something important?”

User Feels:

- Confused due to excessive information
- Overwhelmed by planning details
- Excited about the trip but stressed about organizing it

User Says:

- “I want a customized travel plan.”
- “I don’t want generic or repetitive suggestions.”
- “I need a plan that matches my interests.”

User Does:

- Searches Google for travel blogs and guides
- Watches YouTube travel vlogs
- Reads reviews on travel websites
- Asks friends and family for recommendations

This analysis helped identify the need for a simplified and intelligent travel planning solution.

2.3 Brainstorming

During the ideation phase, multiple possible solutions were explored:

1. Static Travel Blog Website

A website providing pre-written travel guides was considered. However, this approach lacks personalization and cannot adapt to individual user preferences.

2. Rule-Based Itinerary Generator

A system based on predefined rules and templates was evaluated. While structured, this approach offers limited flexibility and cannot dynamically adjust to diverse user interests.

3. AI-Powered Itinerary Generator

An AI-driven system using generative models was identified as the most effective solution. This approach enables dynamic, personalized, and context-aware itinerary generation based on user inputs.

After evaluating these alternatives, the final decision was to develop an AI-powered Travel Itinerary Generator using Google's Gemini Generative AI model to create smart and customized travel plans.

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

The Customer Journey Map outlines the step-by-step interaction of the user with the TravelGuideAI application.

1. **User Opens the Application**
   The user launches the web application through a browser.
2. **User Enters Destination**
   The user specifies the travel location they wish to visit.
3. **User Enters Number of Days and Nights**
   The user provides trip duration details.
4. **User Enters Interests**
   The user mentions preferences such as beaches, adventure, food, culture, historical sites, etc.
5. **User Clicks "Generate" Button**
   The system processes the input and sends a request to the Gemini API.
6. **User Receives Personalized Itinerary**
   The AI-generated itinerary is displayed in a structured format, including:
   - Day-wise plan
   - Tourist attractions
   - Food recommendations
   - Travel tips

This journey ensures a smooth, intuitive, and interactive user experience.

3.2 Solution Requirements

Functional Requirements

Functional requirements describe what the system should do.

- The system shall provide input fields for destination, days, nights, and interests.
- The system shall integrate with Google Gemini API for itinerary generation.
- The system shall generate a personalized itinerary based on user input.
- The system shall validate user inputs before processing.
- The system shall display the generated itinerary on the web interface.

Non-Functional Requirements

Non-functional requirements describe system quality and performance.

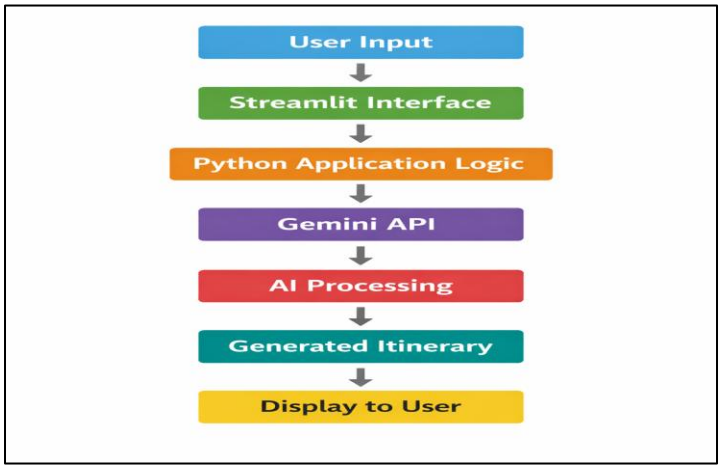
- The system should generate responses within a few seconds.
- The API key must be handled securely using environment variables.
- The interface should be simple and user-friendly.
- The system ‘s performance should be stable under repeated usage.
- The application should be scalable for future feature enhancements.

3.3 Data Flow Diagram

The Data Flow Diagram describes how information moves through the system.

1. The user provides input through the Streamlit interface.
2. The Python backend collects and validates the input data.
3. A structured prompt is generated based on the input.
4. The prompt is sent to the Google Gemini API.
5. The Gemini model processes the request and generates a response.
6. The generated itinerary is returned to the application.

7. The final output is displayed to the user.



3.4 Technology Stack

The project uses the following technologies:

Layer	Technology Used
Frontend	Streamlit
Backend	Python
AI Model	Google Gemini (gemini-2.5-flash)
Development Environment	VS Code
Deployment	Localhost (Streamlit server)

Technology Justification

- **Streamlit** was chosen for rapid UI development and easy deployment.
- **Python** was selected due to strong AI and API support.
- **Google Gemini API** enables dynamic, context-aware itinerary generation.
- **VS Code** provides a lightweight and efficient development environment.

4. PROJECT DESIGN

4.1 Problem–Solution Fit

Identified Problem

Manual travel planning is complex, time-consuming, and often overwhelming. Users must search across multiple platforms to gather information about attractions, food options, accommodations, and travel schedules. This scattered process reduces efficiency and increases planning stress.

Proposed Solution

To address this issue, an AI-based automated itinerary generator was developed. The system uses Generative AI to dynamically create personalized travel plans based on user inputs.

Fit Justification

The solution directly aligns with the identified problem by:

- Eliminating manual research efforts
- Providing instant, structured travel plans
- Personalizing recommendations based on user interests
- Reducing planning time significantly

Thus, the system effectively bridges the gap between traditional travel planning and intelligent automation.

4.2 Proposed Solution

The proposed system leverages Google's Gemini Generative AI model to dynamically generate travel itineraries tailored to user preferences.

System Input Parameters

The application collects the following information from users:

- Destination
- Number of days
- Number of nights
- Interests (e.g., beaches, food, adventure, culture)

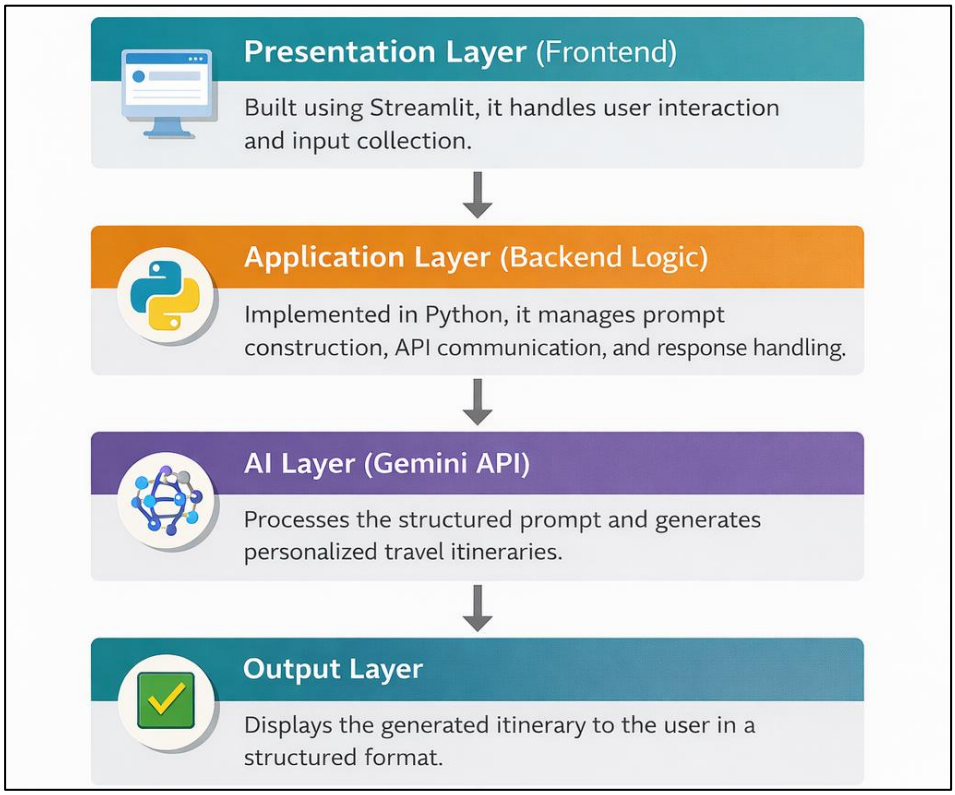
Working Mechanism

1. The user provides travel details through the Streamlit interface.
2. The backend constructs a structured prompt using the provided inputs.
3. The prompt is sent to the Gemini API.
4. The AI model processes the request and generates a customized itinerary.
5. The response is displayed to the user in a readable format.

The solution ensures flexibility, scalability, and real-time personalization.

4.3 Solution Architecture

The architecture of the system follows a simple and modular design to ensure clarity and maintainability.



Architectural Explanation

- **Presentation Layer (Frontend)**
  Built using Streamlit, it handles user interaction and input collection.
- **Application Layer (Backend Logic)**
  Implemented in Python, it manages prompt construction, API communication, and response handling.
- **AI Layer (Gemini API)**
  Processes the structured prompt and generates personalized travel itineraries.
- **Output Layer**
  Displays the generated itinerary to the user in a structured format.

This layered architecture ensures separation of concerns, making the system scalable and easy to enhance in the future.

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

The project was completed over a total duration of 20 days. The development process was divided into multiple phases to ensure systematic execution and timely completion. Each phase focused on specific objectives to maintain clarity and efficiency.

Phase	Duration (Days)
Ideation	3 Days
Design	4 Days
Development	7 Days
Testing	3 Days
Documentation	3 Days
Total	20 Days

Phase Description

Ideation Phase (3 Days):
During this phase, the problem statement was finalized, objectives were defined, and possible solutions were analyzed. The decision to implement a Generative AI-based solution using the Gemini API was made.

Design Phase (4 Days):
The system architecture, data flow, and user interface structure were designed. Input fields and overall workflow of the application were planned.

Development Phase (7 Days):
The core application was developed using Python and Streamlit. Integration with the Gemini API was implemented, and prompt engineering logic was created to generate personalized itineraries.

Testing Phase (3 Days):
The application was tested for input validation, API response handling, performance efficiency, and overall functionality. Errors were identified and corrected.

Documentation Phase (3 Days):
The project report, diagrams, and presentation materials were prepared to clearly explain the system design, implementation, and results.

6. FUNCTIONAL AND PERFORMANCE TESTING

Testing was conducted to ensure that the TravelGuideAI application performs accurately, efficiently, and reliably under different user inputs. Both functional and performance aspects of the system were evaluated.

6.1 Functional Testing

Functional testing was performed to verify that all features of the system work according to the specified requirements.

Functional Test Cases

Test Case	Description	Expected Result	Status
Valid Input Test	Enter valid destination, days, nights, and interests	Personalized itinerary generated successfully	Passed
Empty Field Test	Leave required fields empty	Warning message displayed	Passed
Multiple Destinations	Test with different cities (e.g., Paris, Tokyo, Goa)	Accurate and relevant itinerary generated	Passed
Different Interests	Enter interests like adventure, food, beaches, culture	Customized output based on interests	Passed
Large Input Test	Enter long interest description	System handles input without crash	Passed

All functional requirements were verified and confirmed to be working correctly.

6.2 Performance Testing

Performance testing was conducted to evaluate the system's speed, stability, and responsiveness.

Performance Evaluation

- The application was tested with multiple destinations.
- Different combinations of interests were provided.
- Response time was measured.
- Input validation was verified.
- Empty input scenarios were handled properly.

Performance Results

- Average response time: **2–5 seconds**
- No application crashes observed.
- Stable API communication.
- Smooth user interface performance.

The system demonstrated stable and reliable performance under normal usage conditions.

7. RESULT

7.1 Output Screenshots

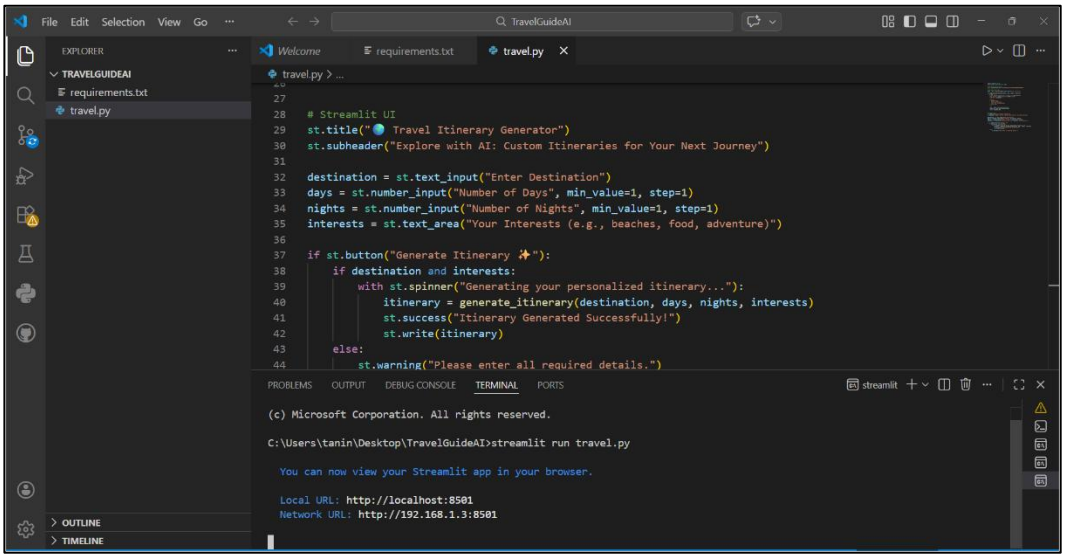


Fig 7.1.1 Source Code with Running Application

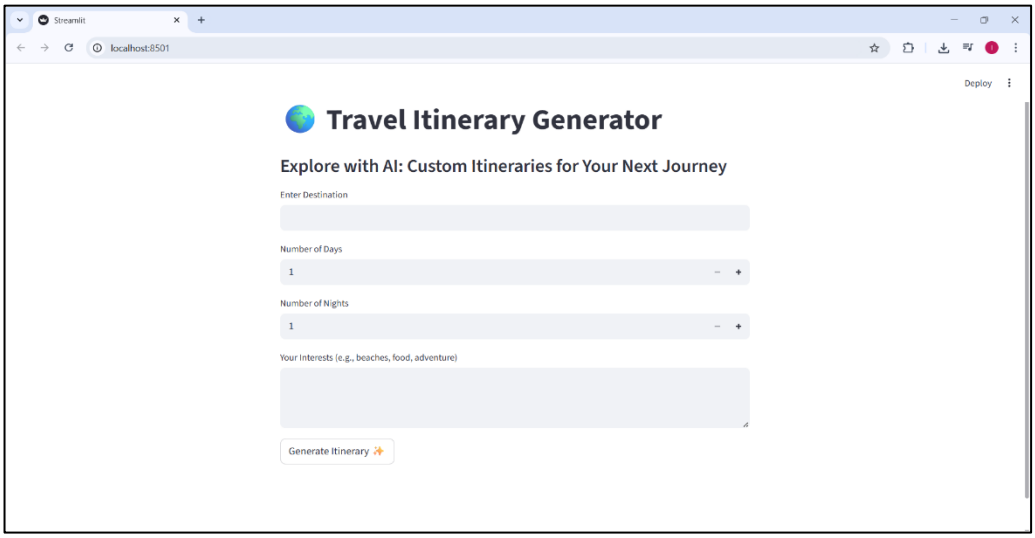


Fig 7.1.2 Application User Interface

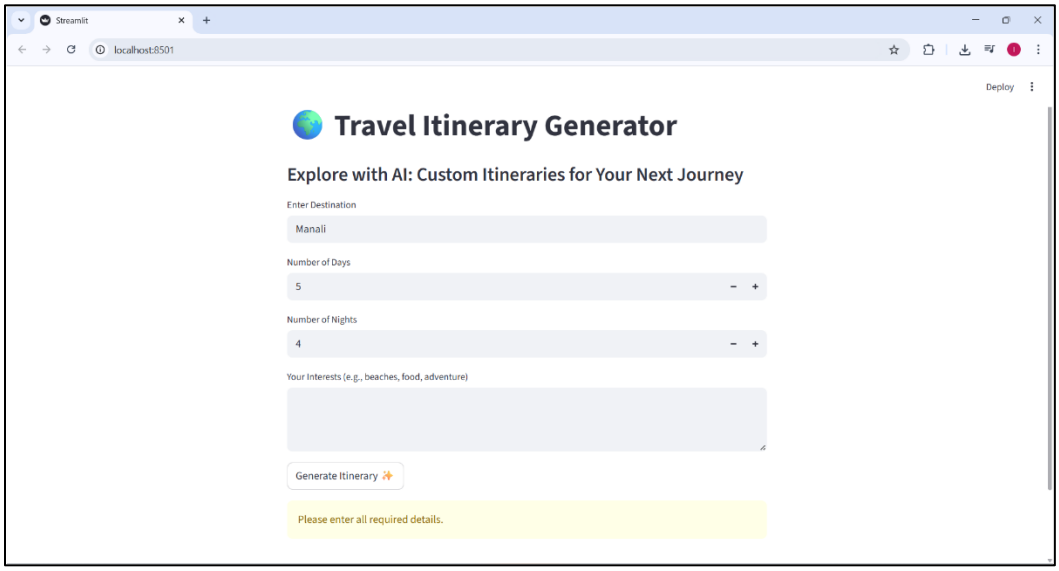


Fig 7.1.3 Input Validation and Error Handling

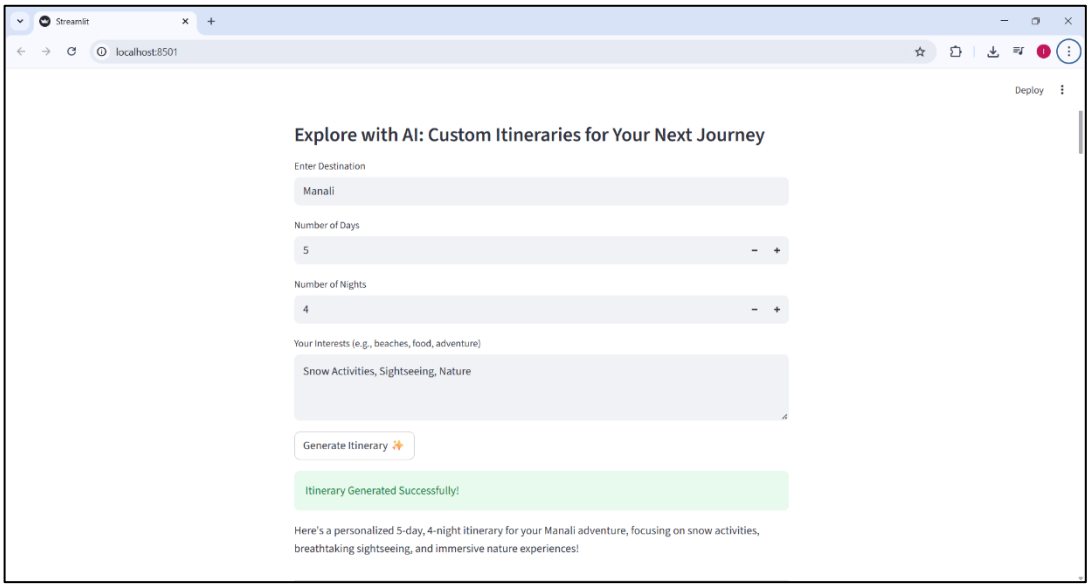


Fig 7.1.4 Generated Itinerary Planning Output

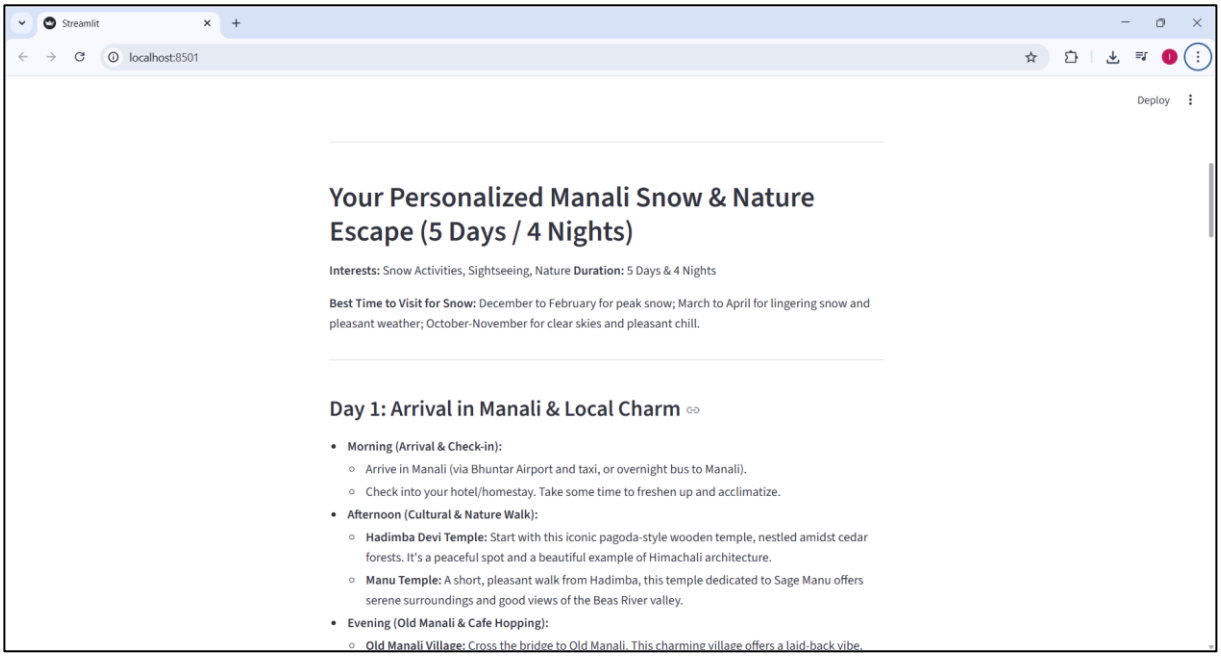


Fig 7.1.4.1 Generated Itinerary Planning Output

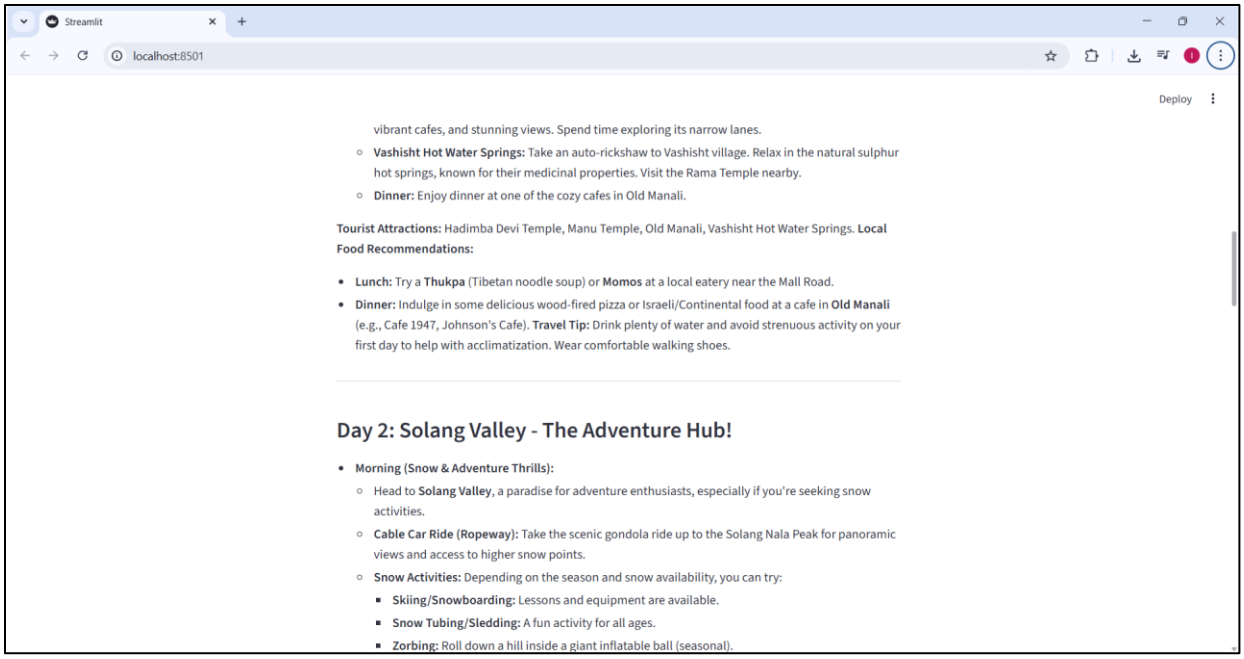


Fig 7.1.4.2 Generated Itinerary Planning Output

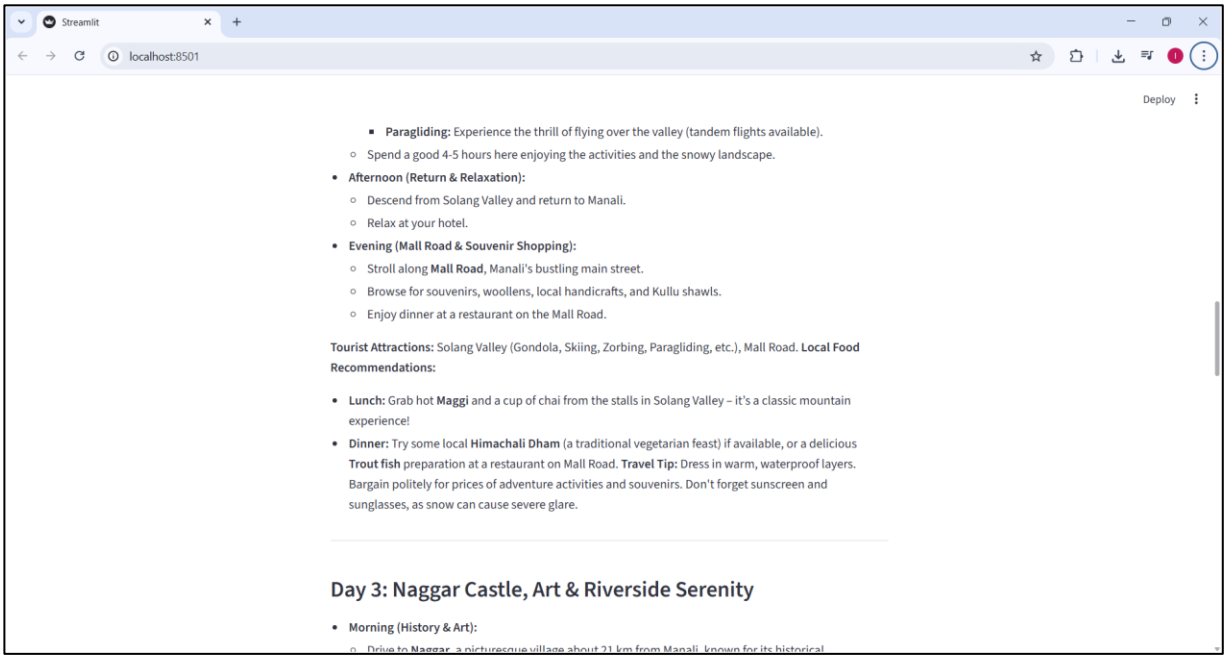


Fig 7.1.4.3 Generated Itinerary Planning Output

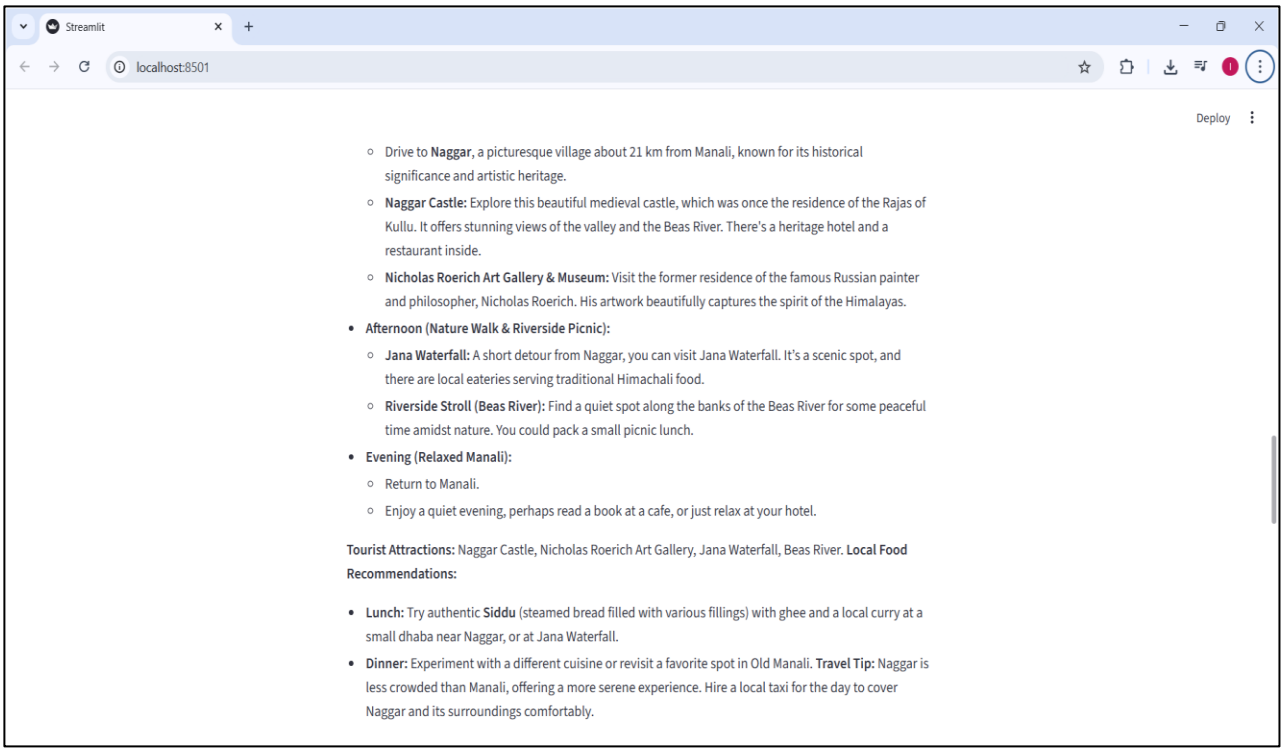


Fig 7.1.4.4 Generated Itinerary Planning Output

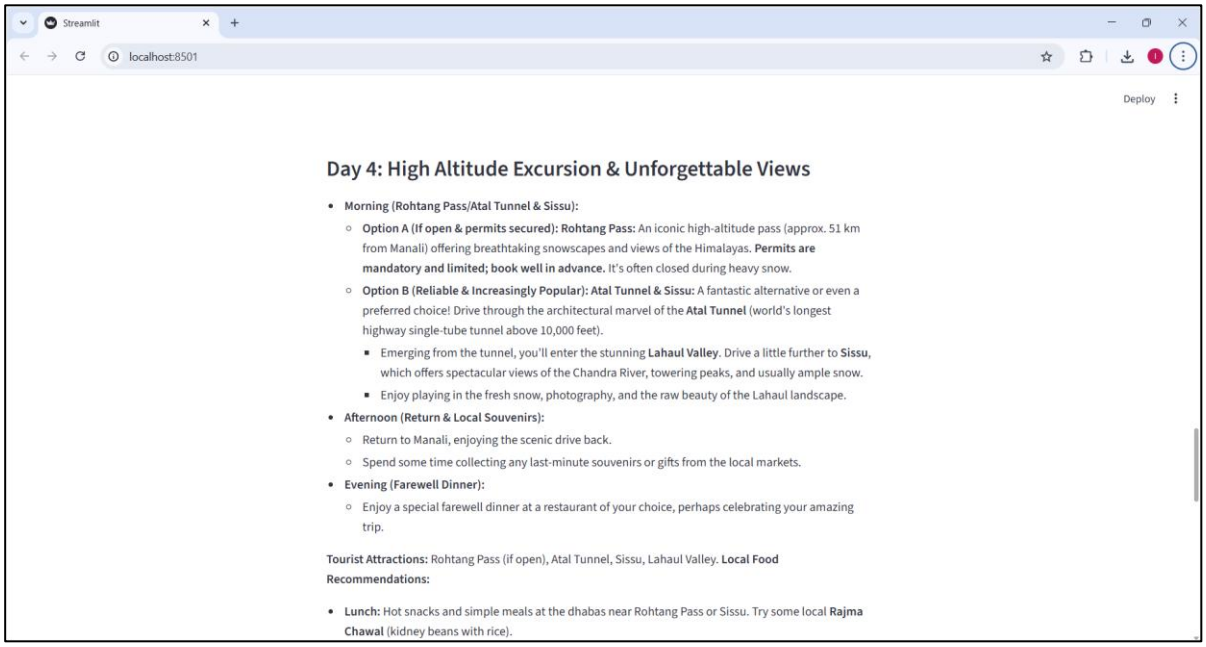


Fig 7.1.4.5 Generated Itinerary Planning Output

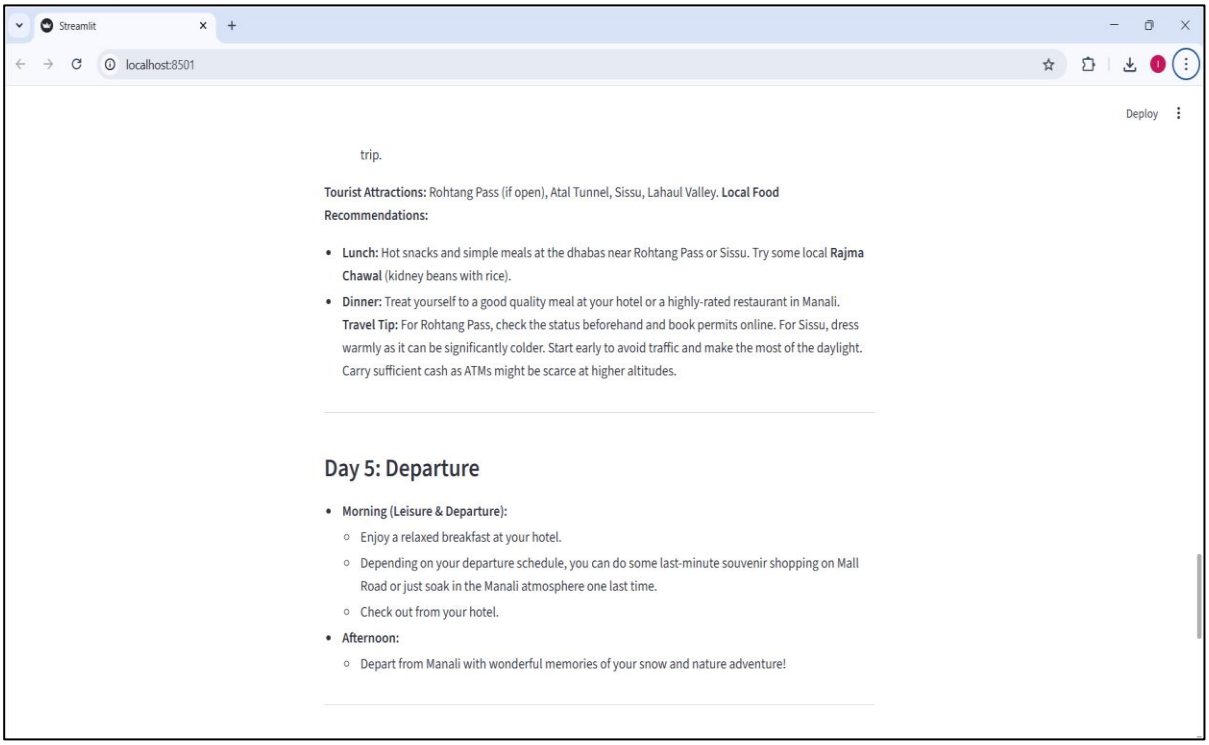


Fig 7.1.4.6 Generated Itinerary Planning Output

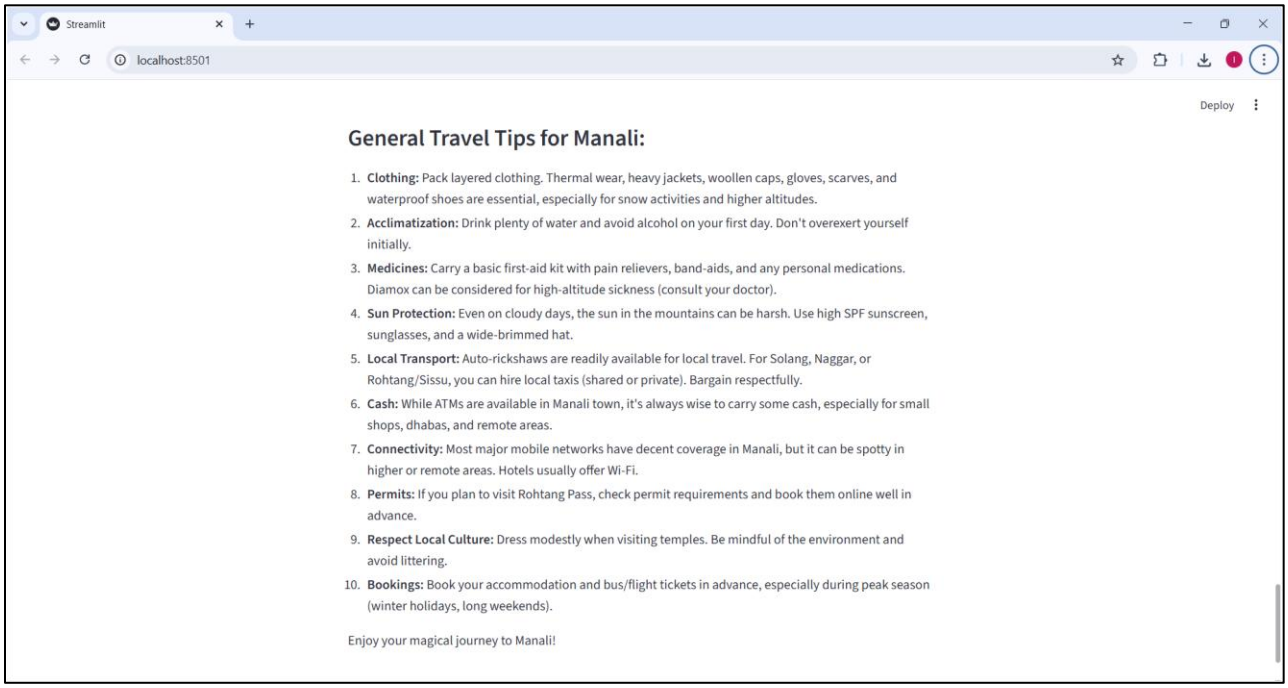


Fig 7.1.4.7 Generated Itinerary Planning Output

8. ADVANTAGES & DISADVANTAGES

8.1 Advantages

The TravelGuideAI system offers several benefits:

- **Personalized Travel Planning:** Generates customized itineraries based on user preferences.
- **Time-Saving:** Eliminates the need for extensive manual research.
- **User-Friendly Interface:** Simple and intuitive design using Streamlit.
- **AI-Powered Dynamic Responses:** Provides intelligent and context-aware travel suggestions.
- **Scalability:** Can be extended for travel agencies and large-scale applications.
- **Automation:** Reduces human effort in itinerary creation.

8.2 Disadvantages

Despite its benefits, the system has certain limitations:

- **Internet Dependency:** Requires an active internet connection to access the Gemini API.

- **API Availability:** Performance depends on external API service uptime.
- **Response Variability:** AI-generated responses may slightly vary for similar inputs.
- **API Usage Limits:** Free-tier API keys may have request limits.
- **No Offline Support:** The system cannot function without cloud-based AI processing.

9. CONCLUSION

The TravelGuideAI project successfully demonstrates the practical integration of Generative AI with web application development. By combining Streamlit and Google's Gemini API, the system provides personalized, dynamic travel itineraries based on user inputs such as destination, duration, and interests.

The project simplifies travel planning, reduces manual effort, and enhances user experience through intelligent automation. It effectively showcases how AI can be applied to solve real-world problems in the travel and tourism domain.

Overall, TravelGuideAI highlights the growing importance of Generative AI in building smart, interactive, and user-centric applications.

10. FUTURE SCOPE

The system can be further enhanced with the following features:

- Integration of **budget planning and cost estimation**
- Hotel and accommodation recommendations
- Integration with **Google Maps API** for route visualization
- PDF download option for generated itineraries
- Cloud deployment (AWS, Azure, or GCP)
- Multilingual support for global users
- User login and profile management system
- Weather-based travel suggestions
- Real-time event recommendations

These improvements can make the system more powerful, scalable, and commercially viable.

11. APPENDIX

The appendix section includes the following:

- **requirements.txt** file
- Complete **travel.py** source code
- Step-by-step installation procedure
- Gemini API configuration method
- Screenshots of application output
- Test case documentation