

TravelGuideAI: Custom Itineraries for Your Next Journey

TEAM DETAILS:

Team ID : LTVIP2026TMIDS90611

Team Lead: Kusuma Boddeti

Team Member: Mukesh Chandra Rama Ganesh Manepalli

Team Member: Mounika Gandharapu

Team Member: Satya Supraja

1. INTRODUCTION

1.1 Project Overview:

Travel planning often requires extensive research to identify destinations, attractions, accommodations, and activities that match a traveler's interests. This process can be time-consuming and sometimes confusing due to the large amount of information available online. **TravelGuideAI** is an AI-powered travel planning system that automatically generates personalized travel itineraries based on user inputs such as destination, duration of stay, and interests. By integrating generative AI models and an interactive web interface, the system simplifies the travel planning process and provides structured, day-wise travel recommendations.

Purpose:

The main purpose of this project is to:

Automate the creation of personalized travel itineraries.

Reduce manual research effort for travelers.

Provide structured travel plans including attractions, activities, and food recommendations.

Demonstrate the use of Large Language Models (LLMs) in real-world applications.

2. IDEATION PHASE

2.1 PROBLEM STATEMENT:

Travelers often face the following challenges:

- Difficulty in planning day-wise travel schedules.
- Lack of personalized recommendations based on interests.
- Time-consuming manual research across multiple websites.

Travel agencies also need faster ways to generate customized travel plans for clients.

Therefore, there is a need for an intelligent system that automatically generates personalized travel itineraries based on user input.

2.2 Empathy Map Canvas:

Users

- Individual Travelers
- Travel Agencies
- Travel Content Creators

User Needs

- Quick and reliable travel planning
- Personalized recommendations
- Easy-to-use interface

User Problems

- Time-consuming research
- Lack of structured travel schedules
- Difficulty choosing attractions and activities

2.3 Brainstorming:

To address travel planning challenges, we designed a system that combines:

- Generative AI for content generation
- Web-based interface for user interaction
- Automated itinerary creation logic

3. REQUIREMENTS:

● 3.1 Functional Requirements:

The system should allow users to perform the following operations:

- Enter the travel destination.
- Enter the number of days and nights.
- Click a button to generate the itinerary.
- View the generated travel plan.
- Copy or export the itinerary for future use.

● 3.2 Non-Functional Requirements:

Usability

- The interface must be simple and easy to understand.
- Users should generate itineraries without technical knowledge.

Performance

- The system should generate responses within a few seconds.

Reliability

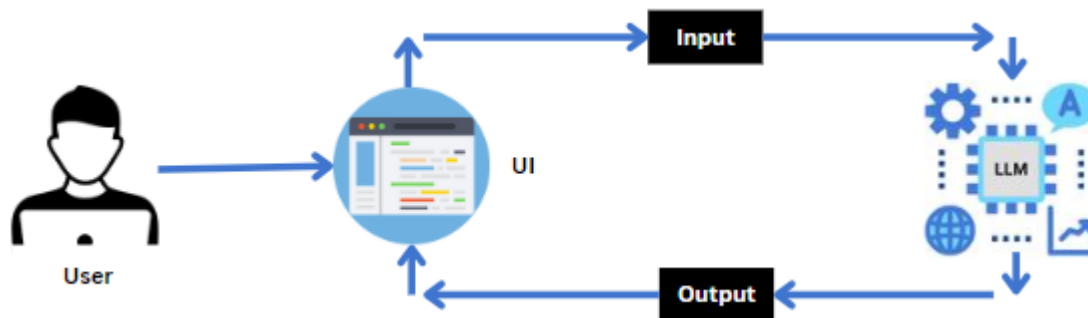
- The system should handle invalid inputs properly.

Security

- API keys should be stored securely using environment variables.

3.3 Data Flow Diagram:

The diagram represents the overall workflow of the **AI-based Travel Itinerary Generator**, showing how user input is processed through the interface and the AI model to produce the final output.



Detailed Flow Breakdown:

Step 1: User Interaction

The user opens the application and enters travel details such as destination, number of days, and nights.

These inputs are provided through the web interface.

Step 2: User Interface Processing

The UI collects and validates the entered data.

It ensures all required fields are correctly filled.

Step 3: Input Transmission

The validated input is sent from the UI to the backend system.

The data is structured for processing by the AI model.

Step 4: AI Model Processing

The backend forwards the input to the Large Language Model (LLM).

The AI analyzes the travel details to generate recommendations.

Step 5: Itinerary Generation

The AI creates a personalized day-wise travel itinerary. It includes attractions, activities, and travel suggestions.

Step 6: Output Display

The formatted itinerary is sent back to the UI. The results are displayed to the user on the screen.

Step 7: User Review

The user reviews the generated itinerary. The user can modify inputs and regenerate the plan if needed.

3.4 Technology Stack:

- **Frontend**
- **Streamlit** – Used to build the interactive web interface for entering travel details and displaying the generated itinerary. Provides input fields, buttons, and output text areas for user interaction.
- **Backend**
- **Python** – Core programming language used for processing user input and integrating the model. Handles application logic and communication with the API.
- **AI Model**
- **Gemini Pro (Generative AI Model)** – Used to generate personalized travel itineraries based on user inputs. Integrated using APIs provided by Google.

4 PROJECT DESIGN:

The project folder includes the following files:

4.1. System Architecture

The project follows a simple three-layer architecture:

- User Layer: Accepts travel details from the user.
- Application Layer: Processes input using Python logic.
- AI Layer: Generates the travel itinerary using the AI model.

4.2. Modules Description

User Input Module

This module collects travel information such as destination, number of days, and nights from the user through the Streamlit interface.

Processing Module

This module validates user input and sends the structured data to the AI model for itinerary generation.

AI Generation Module

The generative AI model processes the input and creates a personalized travel itinerary.

Output Module

The generated itinerary is formatted and displayed on the web interface for the user.

4.3. Project Folder Structure

GuideAI1/

```
|  
├── app.py  
├── requirements.txt  
└── .env
```

app.py – Contains the main application code and AI integration.

requirements.txt – Contains required Python libraries.

.env – Stores API keys securely.

4.4 CODE:

- Requirements.txt

```
streamlit
google-genai
python-dotenv
```

- travel.py

```
travel.py > ...
1  import streamlit as st
2  from google import genai
3  from dotenv import load_dotenv
4  import os
5
6  # Load environment variables from .env file
7  load_dotenv()
8
9  # Configure Streamlit page
10 st.set_page_config(page_title="Travel Itinerary Generator")
11 st.title("🌍 Travel Itinerary Generator")
12
13 # Get API key securely from environment variable
14 api_key = os.getenv("GOOGLE_API_KEY")
15
16 # Check if API key exists
17 if not api_key:
18     st.error("❌ GOOGLE_API_KEY not found. Please check your .env file.")
19     st.stop()
20
```



```

21 # Create Gemini client
22 client = genai.Client(api_key=api_key)
23
24
25 def generate_itinerary(destination, days, nights):
26     prompt = f"""
27     Create a detailed travel itinerary for {destination}
28     for {days} days and {nights} nights.
29
30     Include:
31     - Day-wise plan
32     - Tourist attractions
33     - Activities
34     - Local food suggestions
35     - Travel tips
36     """
37
38     response = client.models.generate_content(
39         model="gemini-2.5-flash",
40         contents=prompt,
41     )

```

```

42
43     return response.text
44
45
46 # User Inputs
47 destination = st.text_input("Enter Destination")
48 days = st.number_input("Number of Days", min_value=1, step=1)
49 nights = st.number_input("Number of Nights", min_value=1, step=1)
50
51 # Button Action
52 if st.button("Generate Itinerary"):
53     if destination:
54         with st.spinner("Generating your itinerary... ✈️"):
55             result = generate_itinerary(destination, days, nights)
56             st.write(result)
57     else:
58         st.warning("⚠ Please enter a destination.")

```

4.5 CODE OVERVIEW:

- Required libraries like streamlit and google-genai are included in the requirements.txt file.
- The API key is securely stored using Streamlit Secrets and accessed safely in the application.
- The generate_itinerary() function creates a structured prompt using user inputs (destination, days, nights).
- The function sends the prompt to the Gemini API and returns the generated travel itinerary.
- Streamlit provides an interactive interface to accept user inputs and display the generated itinerary.
- Input validation ensures a warning message is shown if the destination field is empty.

5 PROJECT PLANNING & SCHEDULING

5.1 Project Planning:

Week 1: Requirement Analysis and Research

- Understanding travel planning challenges
- Studying Large Language Models and generative AI
- Finalizing project idea and tools

Week 2: UI Design and Development

- Designing the Streamlit interface
- Creating input fields for destination, days, and nights
- Implementing basic layout and structure

Week 3: AI Integration and Backend Development

- Connecting the Gemini API with Python
- Creating the itinerary generation function
- Processing user inputs and generating outputs

Week 4: Testing and Deployment

- Testing the application with different inputs
- Fixing errors and improving output format
- Deploying the project using Streamlit and preparing documentation

6 FUNCTIONAL AND PERFORMANCE TESTING:

Case1: Valid input



Travel Itinerary Generator

Enter Destination

kedarnath

Number of Days

3

- +

Number of Nights

2

- +

Generate Itinerary

Kedarnath, one of the most revered Jyotirlingas of Lord Shiva, is nestled in the Garhwal Himalayas. The journey is challenging yet incredibly rewarding, offering breathtaking views and a profound spiritual experience.

This 3-day, 2-night itinerary is designed for a focused trek to Kedarnath, assuming you are starting your journey to the base camp (Sonprayag/Sitapur) on Day 1. Please note that this is a **fast-paced itinerary** and requires good physical fitness.

The description is successfully generated in Case 1.

Objective: Verify itinerary is generated for valid inputs.

Input: Destination = "Paris", Days = 3, Nights = 2

Expected Result:

- System generates a day-wise itinerary.
- Includes attractions, activities, food suggestions, and travel tips.
- Output is displayed correctly on the Streamlit page.

Case2: Empty Destination Test.



Enter Destination

Number of Days

 - +

Number of Nights

 - +

Generate Itinerary

⚠ Please enter a destination.

Objective: Validate input field error handling.

Input: Destination left empty, Days = 3, Nights = 2

Expected Result:

- System displays an error/warning message.
- API is not called.
- No itinerary is generated.

Case3 :Minimum Value Test.

- The description is successfully generated in Case 3.

Travel Itinerary Generator

Enter Destination

kerala

Number of Days

1

- +

Number of Nights

1

- +

Generate Itinerary

A 1-day, 1-night trip to Kerala requires a very focused itinerary to truly experience a slice of the state. Given the time constraint, we'll concentrate on **Fort Kochi**, a charming historical precinct that offers a rich blend of culture, history, art, and delicious food, all within a compact, walkable area, and easily accessible from Cochin International Airport (COK).

Kerala in a Day: Fort Kochi Immersion (1 Day,

Objective: Check API error handling.

Condition: Remove or provide incorrect API key.

Expected Result:

- System shows an appropriate error message.
- Application does not crash.
- User is informed about configuration issue.

7 RESULTS

7.1 Output Screenshots:



Travel Itinerary Generator

Enter Destination

kerala

Number of Days

6

- +

Number of Nights

5

- +

Generate Itinerary

Kerala, also known as "God's Own Country," offers a diverse range of experiences from serene backwaters and pristine beaches to misty mountains and vibrant culture. This 6-day, 5-night itinerary focuses on the most iconic and accessible regions: Kochi, Munnar, and Alleppey (Alappuzha), offering a perfect blend of culture, nature, and relaxation.

Kerala: God's Own Country - 6 Days / 5 Nights

...

Regions Covered: Kochi (Fort Kochi), Munnar, Alleppey (Alappuzha)

Best Time to Visit: October to March (pleasant weather)

Day 1: Arrival in Kochi & Fort Kochi Exploration (Cultural Immersion)

- **Morning (Arrival & Transfer):**
 - Arrive at Cochin International Airport (COK).
 - Transfer to your hotel in **Fort Kochi** (approx. 1-1.5 hours drive). Check-in and freshen up.
 - **Afternoon (Historic Walk):**
 - Begin your exploration with a walk through the charming streets of Fort Kochi, known for its colonial architecture.
 - **Chinese Fishing Nets:** Witness these iconic cantilevered nets in action, a legacy of Chinese traders.
 - **St. Francis Church:** The oldest European church in India, where Vasco da Gama was initially buried.
 - **Santa Cruz Basilica:** A magnificent Gothic-style church.
 - **Evening (Cultural Extravaganza & Dinner):**
 - **Mattancherry Palace (Dutch Palace):** Explore its exquisite murals depicting scenes from Hindu epics (closed on Fridays).
 - **Paradesi Synagogue (Jew Synagogue):** The oldest active synagogue in the Commonwealth, located in Jew Town (closed on Fridays & Saturdays). Browse the antique shops in Jew Town.
-

Day 6: Departure from Kochi

- **Morning (Departure):**
 - Enjoy breakfast at your hotel.
 - Depending on your flight schedule, you might have time for some last-minute souvenir shopping or a relaxing morning.
 - Transfer to Cochin International Airport (COK) for your onward journey, carrying beautiful memories of Kerala.
-

Travel Tips for Kerala:

1. Transportation:

- **Private Car with Driver:** Highly recommended for convenience and comfort, especially for inter-city travel (Kochi-Munnar-Alleppey). You can book this through your tour operator or hotel.
- **Cabs/Auto-rickshaws:** Readily available for local commuting within cities.
- **Ferry:** Used extensively in Kochi for inter-island travel.

2. Accommodation:

- **Book in Advance:** Especially during peak season (Oct-Mar) and for houseboats in Alleppey.
- **Options:** Choose from luxury resorts, boutique hotels, charming homestays, and unique treehouses in certain areas.

8 ADVANTAGES & DISADVANTAGES

ADVANTAGES:

- Provides personalized travel itineraries automatically.
- Saves time by reducing manual travel research.
- User-friendly interface that is easy to operate.
- Demonstrates practical use of Generative AI in real-world applications.
- Quick generation of structured day-wise travel plans.

DISADVANTAGES:

- Requires a stable internet connection for AI processing.
- Output accuracy depends on the quality of user input.
- Dependency on external AI API services.
- Generated content may sometimes be generic or repetitive.

9 CONCLUSION:

The **Travel Itinerary Generator** project successfully demonstrates the practical application of Generative Artificial Intelligence in solving real-world problems related to travel planning. Planning a trip traditionally requires extensive research across multiple websites, blogs, and travel platforms. This process can be time-consuming, confusing, and inefficient. The developed system simplifies this process by automatically generating a structured, day-wise travel itinerary based on user inputs such as destination, number of days, and nights.

By integrating **Python**, **Streamlit**, and the **Gemini AI API**, the application provides a user-friendly web interface that allows users to quickly receive personalized travel plans. The system dynamically constructs prompts and interacts with the AI model to generate meaningful and organized travel suggestions that include tourist attractions, activities, food recommendations, and travel tips.

The project highlights how AI-powered automation can significantly reduce manual effort while enhancing productivity and user experience. It also demonstrates the importance of prompt engineering, API integration, and responsive UI design in building intelligent web applications.

Furthermore, the system ensures proper error handling, input validation, and secure API key management, making it reliable and efficient. Overall, the Travel Itinerary Generator serves as a strong foundation for scalable, AI-driven travel planning solutions and showcases the growing impact of artificial intelligence in everyday applications.

10 FUTURE SCOPE:

Although the current system effectively generates AI-based travel itineraries, there are several enhancements that can further improve its functionality and scalability.

One major future enhancement is the integration of **hotel and transportation booking features**, allowing users to directly book accommodations and travel tickets within the application. This would transform the system into a complete travel management platform.

Another improvement could include **budget-based travel recommendations**, where users can specify their spending limit and receive customized plans accordingly. This would make the system more practical and user-centered.

The application can also be enhanced with **interactive map integration** using mapping services to provide route visualization and distance calculations between tourist spots. This would improve usability and navigation planning.

Adding **multi-language support** would make the platform accessible to global users and expand its reach internationally.

Developing a **mobile application version** would increase accessibility and allow users to generate itineraries on the go.

Additionally, incorporating **real-time weather updates and location-based suggestions** could provide dynamic travel recommendations based on current conditions.

In the future, advanced AI personalization features such as remembering user preferences, suggesting repeat destinations, and integrating user reviews could make the system even smarter and more adaptive.

11 APPENDIX:

- Source Code: Hosted on GitHub.
- GitHub Link:
<https://github.com/KusumaBoddeti/Travel-Itinerary-Generator>
- Project Demo Link:<https://travel-itinerary-generator-kxdh8kn7ijfgpmautwtdjr.streamlit.app/>