

Date	28 January2026
Team ID	LTVIP2026TMIDS74153
Project Name	Explore With Ai: Custom Itineraries For Your Next Journey
Maximum Marks	3 Marks

## 1. INTRODUCTION

### 1.1 Project Overview

Explore with AI: Custom Itineraries for Your Next Journey is a generative AI-based web application designed to simplify and automate travel planning. The project allows users to generate personalized travel itineraries by providing basic inputs such as destination, number of days, and number of nights through an interactive web interface.

The system leverages a pre-trained generative AI model to create detailed and structured travel itineraries that include daily activities, nearby attractions, food recommendations, and useful travel tips. Instead of relying on manual research or static datasets, the application processes real-time user inputs and dynamically generates customized travel plans.

The application is developed using Python and Streamlit, providing a user-friendly interface for seamless interaction. By integrating generative AI with a lightweight web framework, the project demonstrates the practical application of AI in real-world travel planning scenarios. This solution is beneficial for individual travelers, travel agencies, and travel content creators by reducing planning effort, saving time, and improving the overall travel planning experience.

---

## 1.2 Purpose

- To develop an AI-based application for generating personalized travel itineraries.
- To use a pre-trained generative AI model for automated itinerary creation.
- To design a simple user interface using Streamlit for easy interaction.
- To validate user inputs for accurate and meaningful itinerary generation.
- To deploy the application as a working travel planning solution.

## 2. IDEATION PHASE

### 2.1. Problem Statements

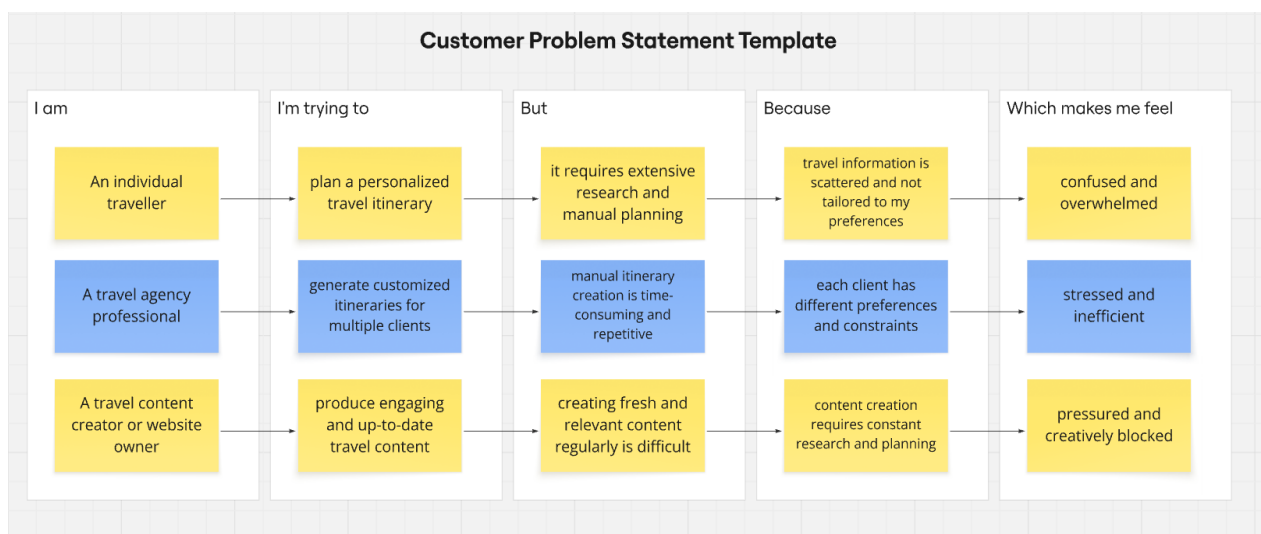
#### Define Problem Statements (Customer Problem Statement Template):

Travel planning and content creation often take a lot of time and effort because they require continuous research and personalization. Many individual travellers find it difficult to create customized travel itineraries that match their interests and trip duration, which can be confusing and frustrating. In the same way, travel agencies struggle to quickly prepare personalized itineraries for multiple clients due to repetitive manual work. Travel websites and content creators also face challenges in regularly producing engaging and updated travel content. These difficulties create a need for an intelligent and automated solution. The **Explore with AI** system addresses these challenges by using generative AI to automatically generate personalized travel itineraries and relevant travel content with minimal manual effort.

---

# GOOGLE CLOUD GENERATIVE AI

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	An individual traveller	plan a personalized travel itinerary	it requires extensive research and manual planning	travel information is scattered and not tailored to my preferences	confused and overwhelmed
PS-2	A travel agency professional	generate customized itineraries for multiple clients	manual itinerary creation is time-consuming and repetitive	each client has different preferences and constraints	stressed and inefficient
PS-3	A travel content creator or website owner	produce engaging and up-to-date travel content	creating fresh and relevant content regularly is difficult	content creation requires constant research and planning	pressured and creatively blocked



Reference: <https://miro.com/templates/customer-problem-statement/>

## 2.2. Empathy Map Canvas

### Project Proposal (Proposed Solution) template

Explore with AI: Custom Itineraries For Your Next Journey is a web-based application that helps users plan their trips easily. The system uses generative AI to create personalized travel itineraries based on user inputs such as destination, number of days, and nights. The application is built using Streamlit and provides a simple interface for users to generate travel plans quickly.

Project Overview	
Objective	The main objective of this project is to develop an AI-based application that automatically generates customized travel itineraries, reducing the time and effort required for manual travel planning.
Scope	<p>The project focuses on:</p> <ul style="list-style-type: none"><li>• Taking user inputs through a Streamlit web interface</li><li>• Using a generative AI model to create travel itineraries</li><li>• Displaying the generated itineraries to users</li></ul> <p>The project does not include booking services or payment features.</p>
Problem Statement	
Description	Creating personalized travel itineraries requires a lot of time and research. Travelers, travel agencies, and content creators often find it difficult to manually prepare customized and engaging travel plans.
Impact	An automated travel itinerary generator helps save time, improves efficiency, and makes travel planning easier for users by providing personalized results quickly.
Proposed Solution	
Approach	The system collects travel details from users through a Streamlit interface. These details are sent to a generative AI model, which creates a detailed travel itinerary. The generated itinerary is then displayed to the user.

---

# GOOGLE CLOUD GENERATIVE AI

Key Features	<ul style="list-style-type: none"><li>Personalized travel itinerary generation using AI</li><li>Easy-to-use web interface</li><li>Fast and automated content generation</li><li>Input validation and error handling</li></ul>
--------------	---

## Resource Requirements

Resource Type	Description	Specification/Allocation
Hardware		
Computing Resources	Standard personal computer	Windows 11 Home Single Language
Memory	Minimum 4 GB	16 GB
Storage	Minimum 20 GB free space	256 GB SSD
Software		
Frameworks	Python, Streamlit frameworks	Streamlit
Libraries	google-generativeai, streamlit	google-generativeai, streamlit
Development Environment	IDE, version control	Visual Studio Code, Git
Data		
Data	Input Data: Destination, days, nights	User Input

## 2.3. Brainstorming

### Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create a product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members	Sprint Start Date	Sprint End Date (Planned)
--------	-------------------------------	-------------------	-------------------	--------------	----------	--------------	-------------------	---------------------------

## GOOGLE CLOUD GENERATIVE AI

Sprint-1	Project Setup	USN-1	As a developer, I want to set up the project environment and install required libraries so that development can start smoothly.	2	High	K.Tejaswi	28 January 2026	31 January 2026
Sprint-1	Model Initialization	USN-2	As a developer, I want to generate and configure the Gemini API key to enable interaction with the generative AI model.	2	High	M.Vasanthi	28 January 2026	31 January 2026
Sprint-2	AI Integration	USN-3	As a user, I want to generate a personalized travel itinerary by providing destination, days, and nights so that I can plan my trip easily.	3	High	P.Jyothi	02 February 2026	09 February 2026
Sprint-2	Prompt Handling	USN-4	As a system, I want to send structured prompts to the Gemini model to generate meaningful and detailed itineraries.	2	Medium	N.Geetha	05 February 2026	09 February 2026
Sprint-3	User Interface	USN-5	As a user, I want a simple and interactive Streamlit interface to enter travel details and view	2	High	K.Tejaswi M.Vasanthi P.Jyothi	10 February 2026	12 February 2026

GOOGLE CLOUD GENERATIVE AI

			generated itineraries.					
Sprint-3	Error Handling	USN-6	As a user, I want the system to validate inputs and display error messages so that incorrect inputs are handled properly.	1	Medium	K.Tejaswi, P.Jyothi, Vasanthi, Geetha	12 February 2026	18 February 2026
Sprint-4	Deployment	USN-7	As a developer, I want to deploy the application using Streamlit so that users can access it through a web interface.	2	High	K.Tejaswi, P.Jyothi, M.Vasanthi, N.Geetha	14 February 2026	18 February 2026

3. REQUIREMENT ANALYSIS

3.1. Customer Journey Map

Data Collection Plan & Raw Data Sources Identification

This phase describes how data is collected and managed for the project. Since the application is based on a generative AI model, the data is collected dynamically from users rather than from static datasets.

Data Collection Plan

Section	Description
Project Overview	Explore with AI is a generative AI-based travel itinerary generation system. The project aims to generate personalized travel itineraries

## GOOGLE CLOUD GENERATIVE AI

	based on real-time user inputs such as destination, number of days, and number of nights using a pre-trained AI model.
Data Collection Plan	The data for this project is collected directly from users through a Streamlit-based web interface. The inputs are provided in real time and are processed immediately to generate customized travel itineraries. No external datasets are downloaded or stored as part of this project.
Raw Data Sources Identified	The raw data used in the system consists of user-provided input data entered through the application interface. This data is temporary and session-based, used only for generating the itinerary and not stored permanently.

### Raw Data Sources

Source Name	Description	Location / URL	Format	Size	Access Permissions
User Input – Destination	Travel destination entered by the user	Streamlit Web Interface	Text	Small (per request)	Public (User Provided)
User Input – Trip Duration	Number of days and nights entered by the user	Streamlit Web Interface	Numeric	Small (per request)	Public (User Provided)
AI Model Response	Generated travel itinerary content	Generated at runtime	Text	Variable	Public (User Generated)

## 3.2. Solution Requirement

### Data Quality Report



# GOOGLE CLOUD GENERATIVE AI

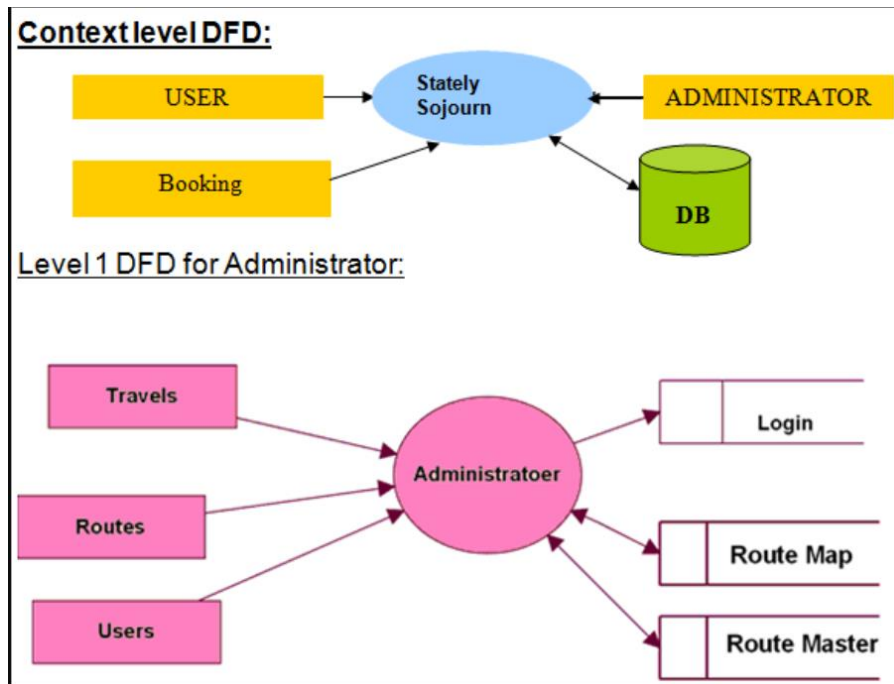
The data used in this project is collected dynamically from users through a web-based interface. Since no static dataset is used, data quality is ensured by validating user inputs before sending them to the generative AI model.

Data Source	Data Quality Issue	Severity	Resolution Plan
User Input (Destination)	Empty or unclear destination input	High	Input validation is applied to ensure the destination field is not empty before processing.
User Input (Days)	Invalid or zero number of days	High	Numeric validation ensures the number of days is greater than zero.
User Input (Nights)	Illogical or negative number of nights	Moderate	Input constraints ensure nights are zero or positive values.
Prompt Input	Incomplete or vague prompt generation	Moderate	Structured prompt formatting is used to ensure meaningful AI responses.
User Interaction	Repeated or incorrect submissions	Low	Error messages guide users to correct inputs before resubmission.

### 3.3. Data Flow Diagram

#### Preprocessing

In the Explore with AI project, data preprocessing focuses on **user-provided textual and numerical input** rather than traditional datasets. Since the application uses a **pre-trained generative AI model**, no external dataset collection or complex preprocessing is required. Instead, preprocessing ensures that user inputs such as destination, number of days, and number of nights are **clean, valid, and meaningful**, enabling the generation of accurate and personalized travel itineraries.



## 3.4 Technology Stack

Section	Description
<b>Data Overview</b>	The data used in this project consists of real-time user inputs such as travel destination, number of days, and number of nights. No external dataset is used.
<b>Text Cleaning</b>	User input text (destination) is cleaned by removing unnecessary spaces and checking for empty or invalid entries.
<b>Input Validation</b>	Ensures that the destination field is not empty, the number of days is greater than zero, and the number of nights is a valid non-negative value.
<b>Token Handling</b>	The validated input is passed to the Gemini generative AI model, which internally handles tokenization and text processing.
<b>Prompt Formatting</b>	The user inputs are formatted into a structured prompt before being sent to the AI model for travel itinerary generation.
<b>Error Handling</b>	The system handles invalid inputs and API-related issues gracefully. If incorrect or missing inputs are detected, appropriate error messages are displayed to guide the user. Any runtime or API errors are also caught and reported without crashing the application.
<b>Data Preprocessing Templates</b>	

# GOOGLE CLOUD GENERATIVE AI

<b>Loading Data</b>	User inputs are collected directly through Streamlit input fields, including text and numeric inputs.
<b>Input Validation</b>	Validation logic ensures that all required inputs are provided and fall within acceptable ranges before further processing.
<b>Prompt Creation</b>	The validated user inputs are combined into a structured prompt that clearly specifies the destination and trip duration.
<b>Model Invocation</b>	The formatted prompt is sent to the pre-trained generative AI model to generate a personalized travel itinerary.
<b>Output Handling</b>	The generated itinerary is received as text output and displayed on the Streamlit interface for user review.

## 4. PROJECT DESIGN

### 4.1. Problem Solution Fit

Problem–Solution Fit is a critical aspect of the Explore with AI: Custom Itineraries for Your Next Journey project because it demonstrates that the proposed system effectively addresses real-world travel planning challenges. Many travelers struggle with organizing trips due to the time-intensive research required, the difficulty of finding reliable information, and the lack of personalized recommendations. These issues often lead to stress, poor planning, and inefficient use of time and budget.

Identified Problem	Proposed Solution	Expected Benefit
Travel planning is time-consuming and requires extensive research across multiple platforms.	AI automates itinerary creation by generating travel plans instantly based on user inputs.	Saves time and reduces manual effort.
Lack of personalized travel recommendations.	The system analyzes preferences such as budget, interests, and trip duration to create customized itineraries.	Provides highly relevant and user-focused travel plans.
Information overload from websites, blogs, and travel guides.	A centralized platform delivers structured travel details in one place.	Simplifies decision-making and avoids confusion.
Difficulty in managing travel budget and schedule.	AI suggests optimized plans that balance cost, time, and activities.	Improves budget control and trip efficiency.

# GOOGLE CLOUD GENERATIVE AI

Identified Problem	Proposed Solution	Expected Benefit
Uncertainty about choosing the best attractions and routes.	Integration with travel APIs provides reliable data on destinations and nearby attractions.	Enhances travel confidence and planning accuracy.
Stress and frustration during trip preparation.	A quick, intelligent planning tool reduces complexity and planning anxiety.	Creates a smoother and more enjoyable travel experience.

The AI-powered travel itinerary generator directly responds to these challenges by automating the planning process. By analyzing user preferences such as destination, budget, duration, and interests, the system generates customized travel plans within seconds. This eliminates the need to browse multiple platforms and ensures that travelers receive structured, relevant, and practical recommendations tailored to their needs.

## 4.2. Proposed Solution

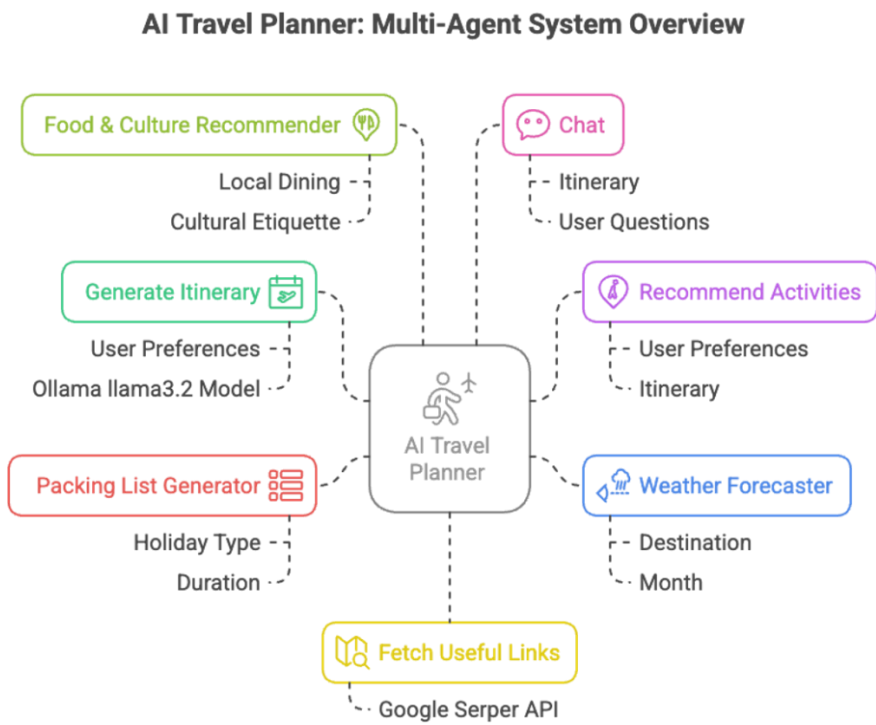
The proposed solution for Explore with AI: Custom Itineraries for Your Next Journey is an intelligent travel planning system that leverages Artificial Intelligence to simplify and enhance the trip preparation process. The application is designed to automatically generate personalized travel itineraries based on user inputs such as destination, budget, travel duration, and individual interests. By replacing manual research with smart automation, the system minimizes planning time and effort while improving the overall travel experience.

Component	Description	Purpose	User Benefit
Intelligent Travel Planner	An AI-based system that generates customized travel itineraries based on user preferences.	Automate the trip planning process.	Saves time and reduces manual research.
User Input Interface	Interactive platform where users enter details such as destination, budget, duration, and interests.	Collect accurate traveler requirements.	Easy and user-friendly experience.
AI Processing Engine	Uses Generative AI and Natural Language Processing to analyze inputs and create structured travel plans.	Deliver personalized recommendations.	Highly relevant and tailored itineraries.

# GOOGLE CLOUD GENERATIVE AI

Component	Description	Purpose	User Benefit
Data Integration	Connects with travel data sources to gather information about attractions, routes, and travel details.	Ensure reliable and practical recommendations.	Improves planning accuracy and confidence.
Itinerary Generator	Organizes travel suggestions into a clear day-by-day schedule.	Provide a structured travel plan.	Helps users stay organized throughout the trip.
Centralized Platform	Combines all travel information into a single system.	Eliminate the need to search multiple websites.	Simplifies decision-making.

## 4.3 Solution Architecture



Layer	Component	Function
Presentation Layer	Web Interface (Streamlit)	Allows users to input travel details and view generated itineraries.
Application Layer	Backend Logic	Processes requests, manages workflows, and connects system components.
Intelligence Layer	AI Processing Engine	Analyzes user preferences and generates personalized travel recommendations.
Data Layer	External Travel APIs	Provides real-time data about destinations, routes, and attractions.
Output Layer	Itinerary Generator	Formats AI results into a structured, day-by-day travel plan.

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

Project Planning is a crucial phase that establishes a clear roadmap for developing the AI-powered travel itinerary system. It involves defining project goals, identifying required technologies, organizing tasks, and setting timelines to ensure smooth execution. Effective planning helps minimize risks, optimize resources, and maintain focus throughout the development lifecycle.

Planning Aspect	Description	Outcome
Project Objective	Develop an AI-based system to generate customized travel itineraries.	Clear vision and direction.
Problem Identification	Recognize challenges such as time-consuming research and lack of personalization in travel planning.	Strong foundation for the solution.

# GOOGLE CLOUD GENERATIVE AI

Planning Aspect	Description	Outcome
Technology Selection	Choose Python, Streamlit, and Generative AI for development.	Efficient and scalable system.
Requirement Analysis	Determine user needs including budget planning, destination suggestions, and schedule creation.	User-centered design.
Module Design	Divide the system into components like User Interface, AI Engine, and Data Integration.	Easier development and maintenance.
Resource Planning	Allocate tools, development environment, and testing methods.	Smooth workflow.
Timeline Creation	Define milestones for design, development, testing, and deployment.	On-time project completion.

## 6. FUNCTIONAL AND PERFORMANCE TESTING

### 6.1. Performance Testing

Performance Testing evaluates how efficiently the system operates under different conditions. It ensures that the AI-powered travel planner delivers fast responses, handles multiple user requests smoothly, and maintains reliability without performance degradation. This testing phase is essential to guarantee a seamless user experience, especially when generating personalized itineraries in real time.

Testing Parameter	Description	Result / Expected Outcome
Response Time	Measures how quickly the system generates an itinerary after user input.	The system responds within a few seconds, ensuring minimal waiting time.

Testing Parameter	Description	Result / Expected Outcome
Load Handling	Evaluates the system’s ability to manage multiple users simultaneously.	Maintains stable performance without crashes.
Scalability	Tests whether the application can support increased user demand.	System can be expanded with cloud support when needed.
Accuracy Under Load	Checks if recommendation quality remains high during heavy usage.	Consistent and relevant itineraries generated.
System Stability	Monitors performance during prolonged operation.	No major failures or interruptions observed.
Resource Utilization	Analyzes CPU and memory usage during execution.	Optimized resource consumption.

7. RESULTS

7.1. Output Screenshots

The complete execution of Explore with AI: Custom Itineraries for Your Next Journey application is represented step by step in the following screenshots.

**Step 1:** To run the Streamlit Application we have to use the command `streamlit run travel.py` in the terminal in path where the `travel.py` file is located.



The image shows a VS Code editor window with a Python file named `travel.py` open. The file is part of a project named "Explore with AI Custom Itineraries". The code defines a function `generate_itinerary` that takes `destination`, `days`, and `nights` as arguments. It uses the `google-generativeai` library to call the Gemini API and generate a detailed travel itinerary. The output is printed to the console.

```
1 # Import the Gemini API
2 from google.generativeai import GenerativeModel
3
4 # API key (replace with your actual API key)
5 client = google.generativeai.Client(api_key="AIzaSyB840xol2pWl2bu3Ms3lFmfYqDn2Ico")
6
7 def generate_itinerary(destination, days, nights):
8     # Create model configuration
9     generation_config = {
10         "temperature": 0.7,
11         "top_p": 0.9,
12         "top_k": 50,
13         "max_output_tokens": 2048,
14         "response_mime_type": "text/plain"
15     }
16
17     response = client.models.generate_content(
18         model="gemini-2.5-flash",
19         contents=f"""
20             Create a detailed travel itinerary for {destination}
21             for {days} days and {nights} nights.
22         """)
23
24     print(response.text)
```

The terminal output shows the command `streamlit run travel.py` being executed. The output displays a sample itinerary for a 7-day trip to Paris.

```
PS E:\Smarbridge 4-2 project\Explore with AI Custom Itineraries> E:\Anaconda\Scripts\activate
PS E:\Smarbridge 4-2 project\Explore with AI Custom Itineraries> conda activate myenv
conda : the term 'conda' is not recognized as the name of a cmdlet, function, script file, or operable program. Check the spelling of the
name, or if a path was included, verify that the path is correct and try again.
At line:1 char:1
+ conda activate myenv
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (conda:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

PS E:\Smarbridge 4-2 project\Explore with AI Custom Itineraries> streamlit run travel.py
```

```
PROBLEMS OUTPUT TERMINAL PORTS
```

```
PS E:\Smartbridge 4-2 project\Explore with AI Custom Itineraries> conda activate myenv
conda : The term 'conda' is not recognized as the name of a cmdlet, function, script file, or operable program. Check the spelling of the
name, or if a path was included, verify that the path is correct and try again.
At line:1 char:1
+ conda activate myenv
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (conda:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

PS E:\Smartbridge 4-2 project\Explore with AI Custom Itineraries> streamlit run travel.py

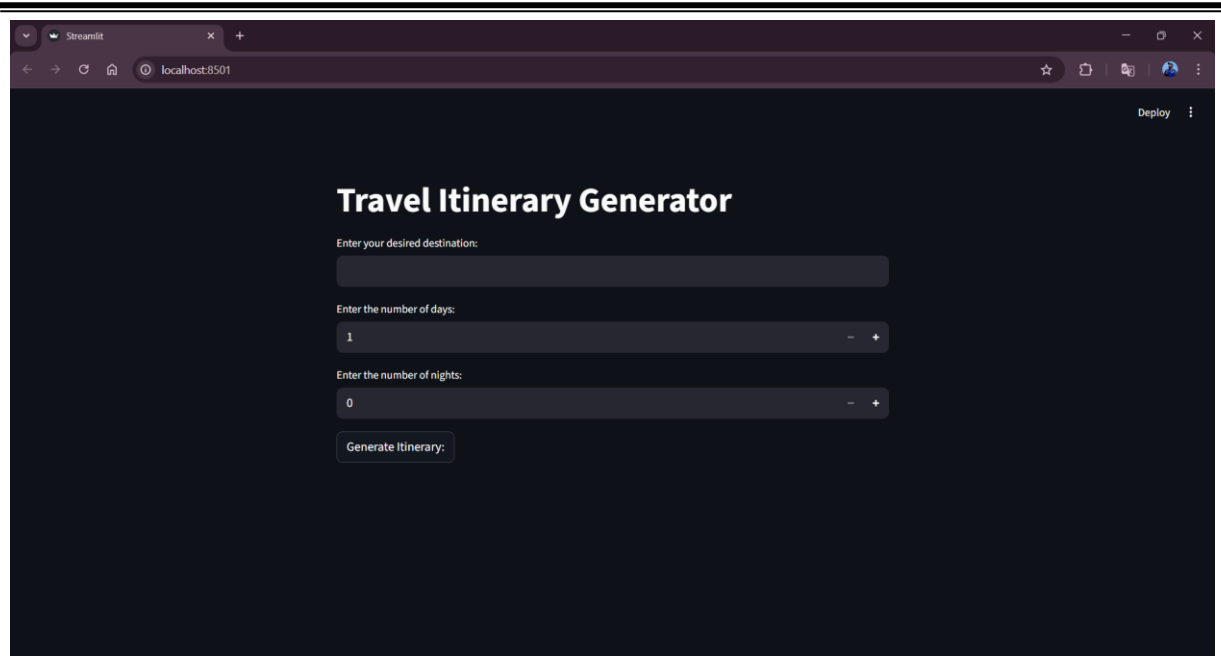
You can now view your Streamlit app in the browser.

Local URL: http://localhost:8501
Network URL: http://192.168.29.223:8501
```

Ln 5, Col 73 Spaces: 4 UTF-8 CRLF Python 3.11.13 (myenv) Go Live Prettier

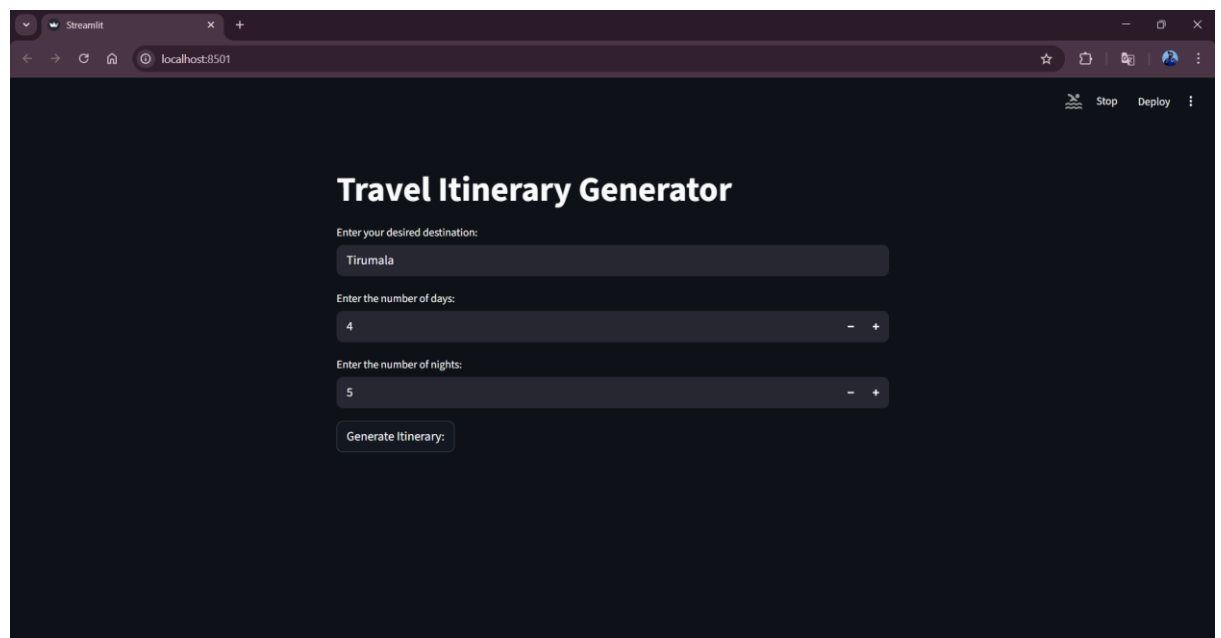
**Step 3:** The Streamlit webpage opens as shown in the figure given below. This is an automated webpage. No secondary HTML codes required to build this webpage. Python code itself consists the webpage building code.

# GOOGLE CLOUD GENERATIVE AI



The screenshot shows a web browser window with the address bar displaying 'localhost:8501'. The page title is 'Streamlit'. The main heading is 'Travel Itinerary Generator'. Below the heading, there are three input fields: 'Enter your desired destination:', 'Enter the number of days:', and 'Enter the number of nights:'. The first field is empty. The second field contains the value '1'. The third field contains the value '0'. Below these fields is a button labeled 'Generate Itinerary:'. In the top right corner, there is a 'Deploy' button.

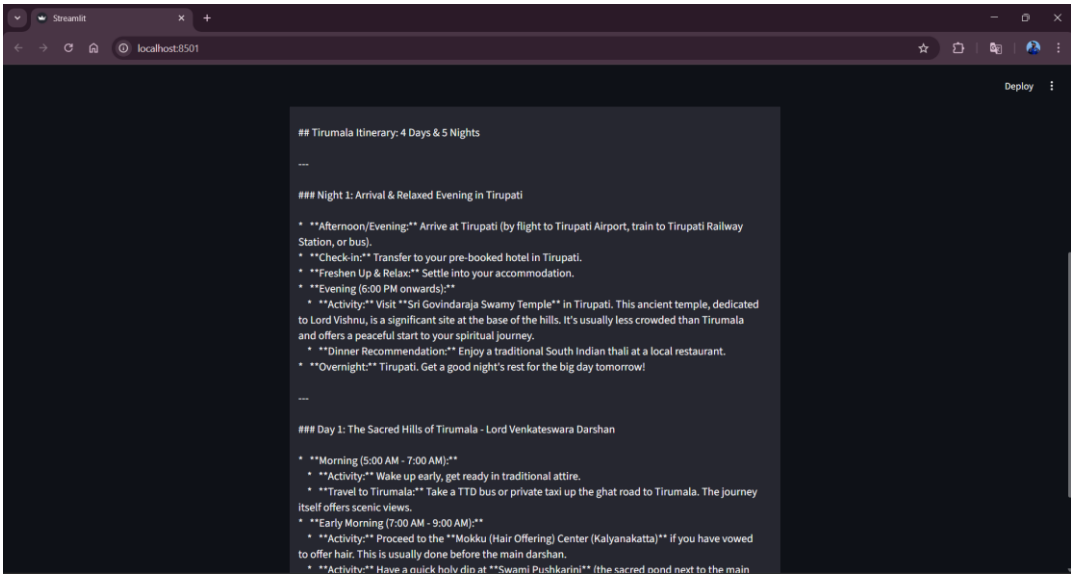
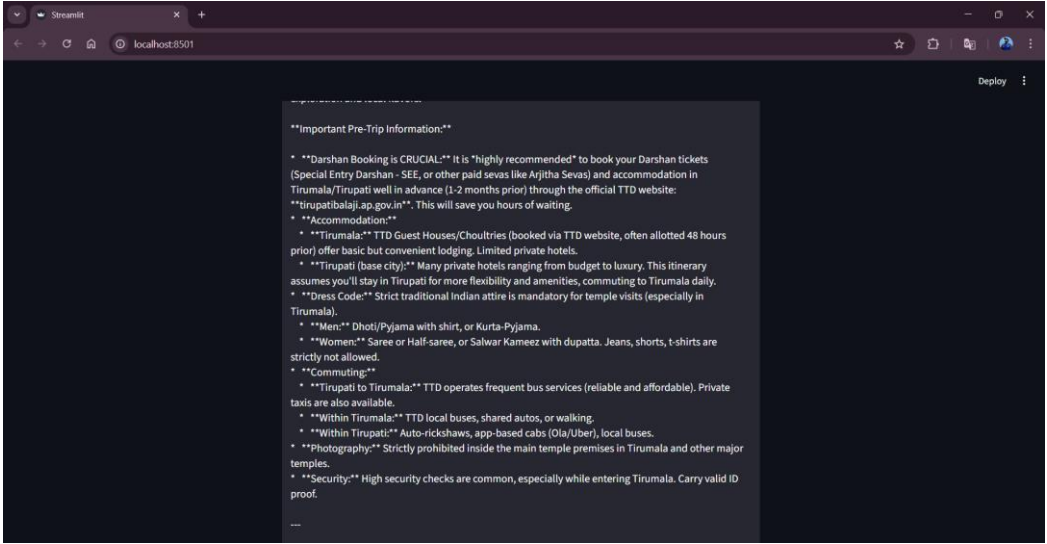
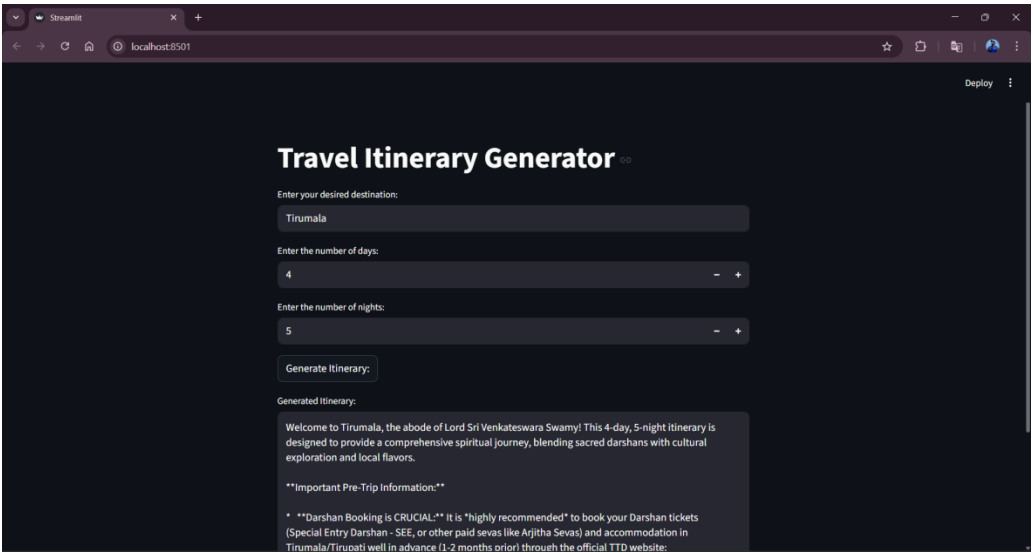
**Step 3:** The user has to give inputs in the website such as desired Destination Name, Number of Days and Number of Nights. The Number of Days and Nights mean in how many days and nights, the user wants their travel planning to be made. After entering the required details, the user should click on Generate Itinerary button to generate the travel plan. Here I chose the destination as Tirumala, days as 4 and nights as 5.



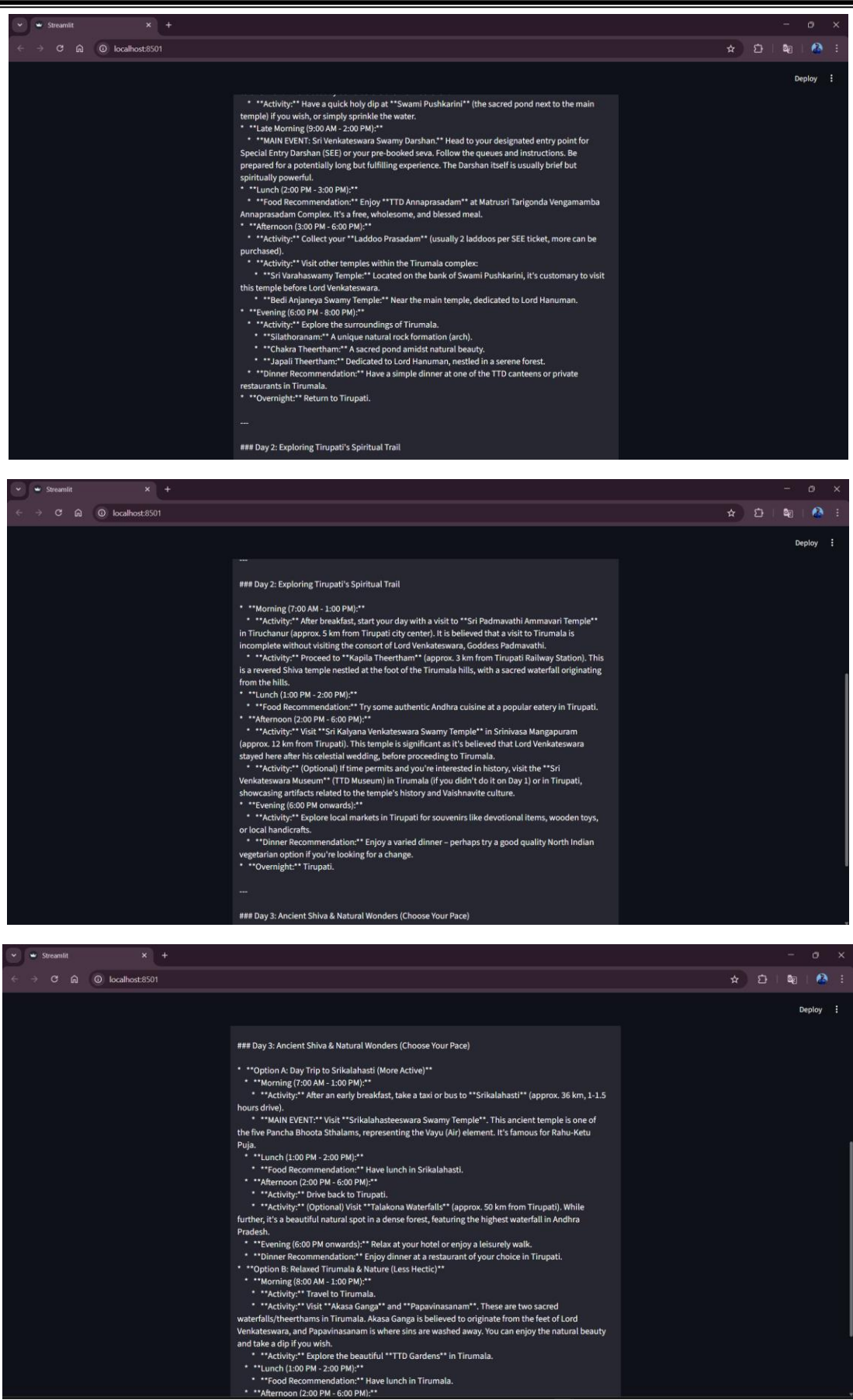
The screenshot shows the same web browser window as before, but with user input. The 'Enter your desired destination:' field now contains 'Tirumala'. The 'Enter the number of days:' field now contains '4'. The 'Enter the number of nights:' field now contains '5'. The 'Generate Itinerary:' button is still visible. In the top right corner, there are now 'Stop' and 'Deploy' buttons.

**Step 4:** After clicking the Generate Itinerary button, in fraction of seconds the itinerary will be generated based on the user input as shown in the following images.

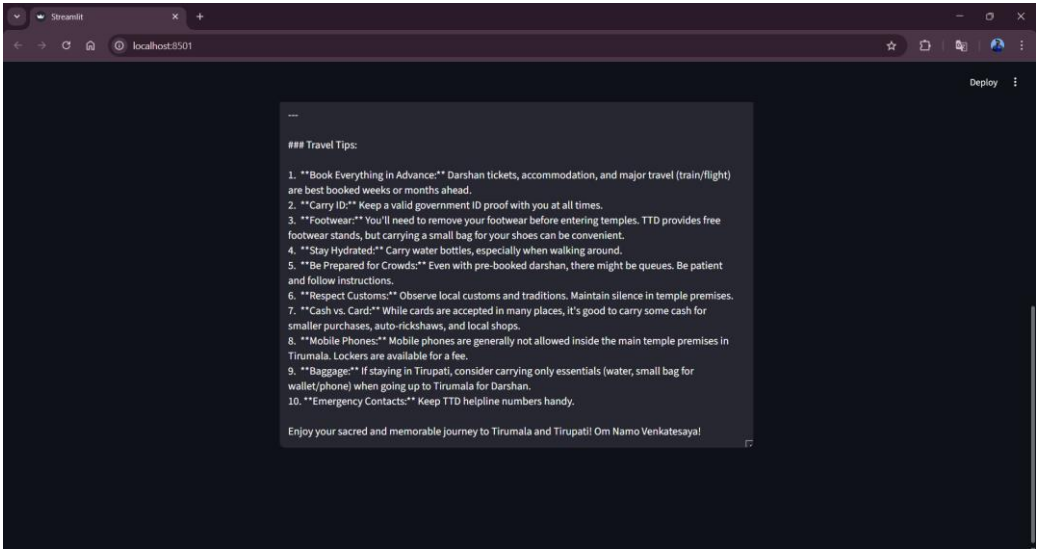
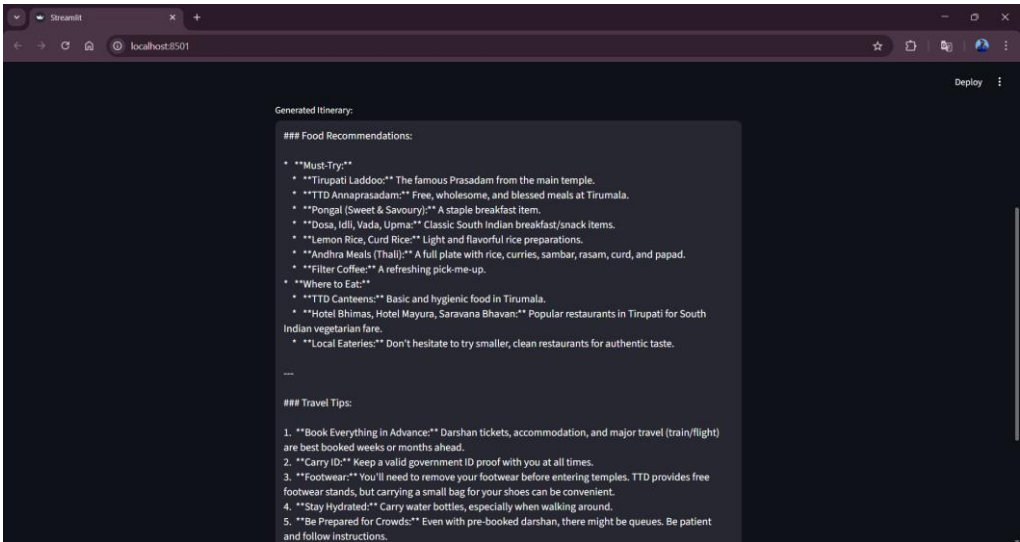
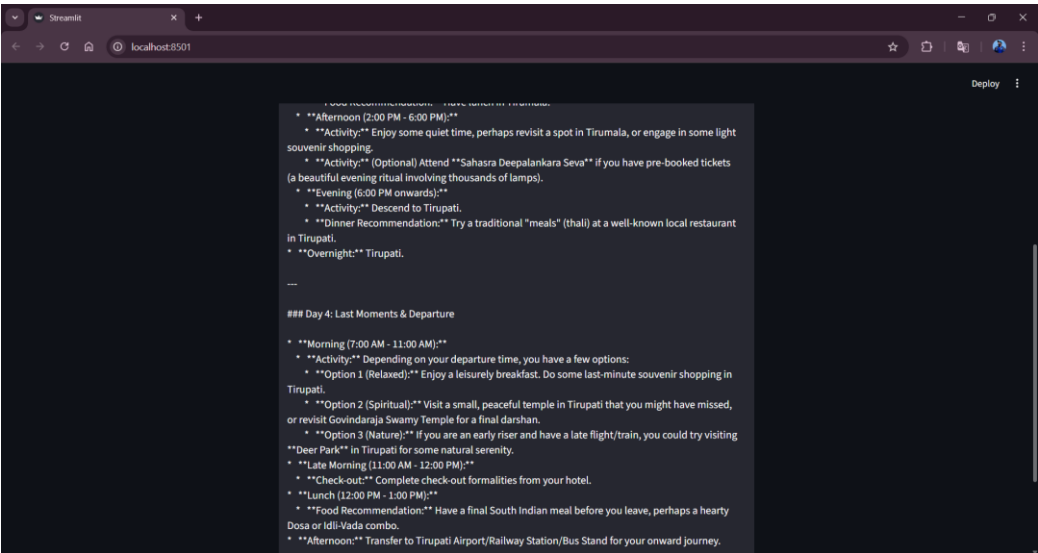
# GOOGLE CLOUD GENERATIVE AI



# GOOGLE CLOUD GENERATIVE AI



# GOOGLE CLOUD GENERATIVE AI



### 8. ADVANTAGES AND DISADVANTAGES

#### Advantages

- Automates travel itinerary creation, saving time and effort for users.
- Generates personalized travel plans based on user preferences and trip duration.
- Easy-to-use interface built with Streamlit, suitable for non-technical users.
- Uses a pre-trained generative AI model, eliminating the need for model training.
- Can be extended to support travel agencies and content creators.

#### Disadvantages

- Depends on internet connectivity and availability of the AI API.
  - Generated itineraries may not always reflect real-time travel conditions.
  - Limited customization options in the current version of the application.
  - Does not include booking, cost estimation, or real-time availability features.
  - API usage may incur cost limitations based on usage quotas.
-

### 9. CONCLUSION

The Explore with AI project demonstrates how generative AI can be used to simplify travel planning by automatically generating personalized travel itineraries. The application integrates a pre-trained AI model with a Streamlit interface to provide quick and user-friendly itinerary generation. This project highlights the practical use of AI in real-world applications and serves as a foundation for future enhancements in intelligent travel planning. The developed application serves as a functional prototype that can be further expanded with additional features, making it a valuable tool for travelers, travel agencies, and content creators.

### 10. FUTURE SCOPE

The application can be enhanced by allowing users to specify additional preferences such as budget, travel style, and accommodation type. Real-time data such as weather updates, local events, and travel advisories can be integrated for more accurate itineraries. Booking features for hotels, transportation, and tourist attractions can be added to make the system more comprehensive. Multi-language support can be introduced to serve users from different regions. The system can be deployed on cloud platforms and scaled to support a larger number of users.



## 11. APPENDIX

### Source Code

The source code for the Explore with AI: Custom Itineraries for Your Next Journey project includes the implementation of the Streamlit user interface, integration of the Gemini Flash model using the Google Generative AI API, travel plan/ itinerary generation logic, and plan upgrading based on the number of days and nights given by the user feature. The code is written in Python and follows a modular and readable structure.

#### Code:

```
import streamlit as st

import os

from dotenv import load_dotenv

import google.generativeai as genai

# -----

# Load API Key from .env

# -----

load_dotenv()

API_KEY = os.getenv("GOOGLE_API_KEY")

if not API_KEY:

    st.error(" ✕ API Key not found. Please check your .env file.")

    st.stop()
```

---

# GOOGLE CLOUD GENERATIVE AI

---

```
# -----
```

```
# Configure Gemini
```

```
# -----
```

```
genai.configure(api_key=API_KEY)
```

```
model = genai.GenerativeModel(  
    model_name="models/gemini-flash-latest",  
    generation_config={  
        "temperature": 0.4,  
        "top_p": 0.95,  
        "top_k": 64,  
        "max_output_tokens": 2048  
    }  
)
```

```
# -----
```

```
# Background Styling (CLEAN + SAFE)
```

```
# -----
```

```
def set_background():  
    st.markdown(  
        """  
        <style>  
        .stApp {  
            background: radial-gradient(circle at top left, #1f2933, #020617);
```

# GOOGLE CLOUD GENERATIVE AI

---

```
        color: #ffffff;

    }

    h1, h2, h3, label {

        color: #e5e7eb !important;

    }

    input, textarea {

        background-color: #020617 !important;

        color: #ffffff !important;

        border-radius: 10px !important;

    }

    button {

        background: linear-gradient(135deg, #6366f1, #8b5cf6) !important;

        color: white !important;

        border-radius: 12px !important;

        padding: 0.6rem 1.4rem !important;

        font-weight: 600 !important;

    }

    /* 🔄 REMOVE "Press Enter to apply" (Streamlit caption/help text) */

    div[data-testid="stTextInput"] small,

    div[data-testid="stTextInput"] div[data-testid="stCaption"],
```

---

# GOOGLE CLOUD GENERATIVE AI

---

```
div[data-testid="stNumberInput"] small,

div[data-testid="stNumberInput"] div[data-testid="stCaption"] {

    display: none !important;

}

</style>

"""

unsafe_allow_html=True

)


# -----

# Generate Travel Itinerary

# -----

def generate_itinerary(destination, days, nights):

    prompt = f"""

    Create a detailed and clean travel itinerary.


    Destination: {destination}

    Duration: {days} days and {nights} nights

    Include:

    - Day-wise plan

    - Attractions

    - Food recommendations

    - Travel tips

    """
```

---

# GOOGLE CLOUD GENERATIVE AI

---

```
response = model.generate_content(prompt)

return response.text


# -----

# Streamlit UI

# -----

st.set_page_config(

    page_title="Travel Itinerary Generator",

    page_icon="🌐",

    layout="centered"

)


set_background()


st.title("🌐 Travel Itinerary Generator")

st.divider()


destination = st.text_input("📍 Enter your destination")

days_input = st.text_input("📅 Number of days", value="1")

nights_input = st.text_input("🌙 Number of nights", value="0")

st.divider()


if st.button("🚀 Generate Itinerary"):
```

---

## GOOGLE CLOUD GENERATIVE AI

---

try:

```
days = int(days_input)
```

```
nights = int(nights_input)
```

```
if days <= 0 or nights < 0:
```

```
    st.warning("Days must be > 0 and nights ≥ 0.")
```

```
elif not destination.strip():
```

```
    st.warning("Please enter a destination.")
```

```
else:
```

```
    with st.spinner("🔄 Planning your journey..."):
```

```
        itinerary = generate_itinerary(destination, days, nights,)
```

```
    st.text_area(
```

```
        "📖 Generated Travel Itinerary",
```

```
        itinerary,
```

```
        height=300
```

```
    )
```

```
except ValueError:
```

```
    st.error("Please enter valid numeric values for days and nights.")
```

---

## Github& Project Demo Link

### ❖ Demo Video Link:

<https://drive.google.com/file/d/1COsoDT3oATVqB6-n6zOv6vdPS-YulZUS/view?usp=sharing>

### ❖ Github Repository:

<https://github.com/JahnaviBurri/Explore-with-AL-Custom-Itineraries-for-your-Next-Journey-.git>

---