EXPLORE WITH AI: CUSTOM ITINERARIES FOR YOUR NEXT JOURNEY (TravelGuideAI)

1. INTRODUCTION

1.1 Project Overview

TravelGuideAI is an AI-powered web application designed to generate personalized travel itineraries automatically. The system takes user inputs such as destination, number of travel days, nights, and preferences, and processes them using a generative AI model to create structured, day-wise travel plans. The application uses Streamlit for the frontend to provide a simple and interactive user interface, while Python handles backend processing. By integrating generative AI, the project demonstrates how modern AI technologies can be applied to solve real-world problems in the travel domain.

1.2 Purpose

The primary purpose of this project is to reduce the effort and time involved in planning travel itineraries. It aims to provide users with personalized and meaningful travel recommendations while minimizing manual research. For travel agencies, the system can automate itinerary creation, improving efficiency and consistency. Additionally, the project serves as a practical demonstration of generative AI integration in web applications, showcasing its potential for automation and personalization.

2. IDEATION PHASE

2.1 Problem Statement

Travel planning often requires extensive research across multiple platforms to identify destinations, attractions, food options, and accommodations. This process can be overwhelming for individual travelers and inefficient for travel agencies that must repeatedly create customized itineraries for different clients. The lack of automation and personalization highlights the need for an AI-based solution that can generate accurate and customized itineraries quickly and effectively.

2.2 Empathy Map Canvas

User: Individual Traveler

- Thinks: I want a perfect trip but don't know where to start.

- Feels: Confused and overwhelmed.

- Needs: Quick and personalized travel plan.

User: Travel Agency

- Needs automation for faster itinerary generation.

- Wants efficient and high-quality output.

2.3 Brainstorming

Possible solutions included static guides, predefined templates, and AI-based dynamic generation. The final solution selected was an AI-powered itinerary generator using Gemini Pro LLM.

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

1. User opens Streamlit app
2. Enters destination, days, nights, preferences
3. Clicks Generate
4. Backend sends data to Gemini Pro
5. AI generates itinerary
6. Output displayed to user

3.2 Solution Requirement

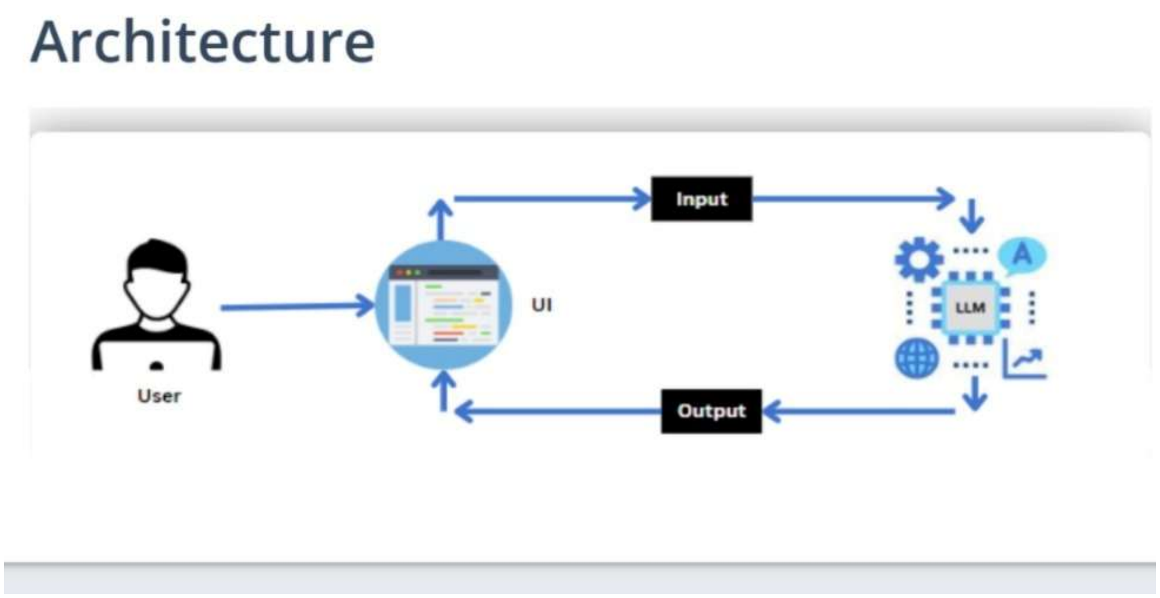
Functional Requirements:

- Accept destination and duration
- Generate structured itinerary
- Display output clearly

Non-Functional Requirements:

- Fast response time
- User-friendly interface
- Secure API integration

3.3 Data Flow Diagram



3.4 Technology Stack

The technology stack was selected to ensure simplicity, scalability, and effectiveness. Streamlit is used for the frontend due to its ease of development and rapid deployment capabilities. Python serves as the backend language because of its strong support for AI integration. A generative AI model is used to produce intelligent itineraries, and deployment is handled through cloud-based platforms for accessibility.

- **Frontend**: Streamlit
- **Backend**: Python
- **AI Model**: Google Gemini Pro
- **Deployment**: Streamlit Cloud

4. PROJECT DESIGN

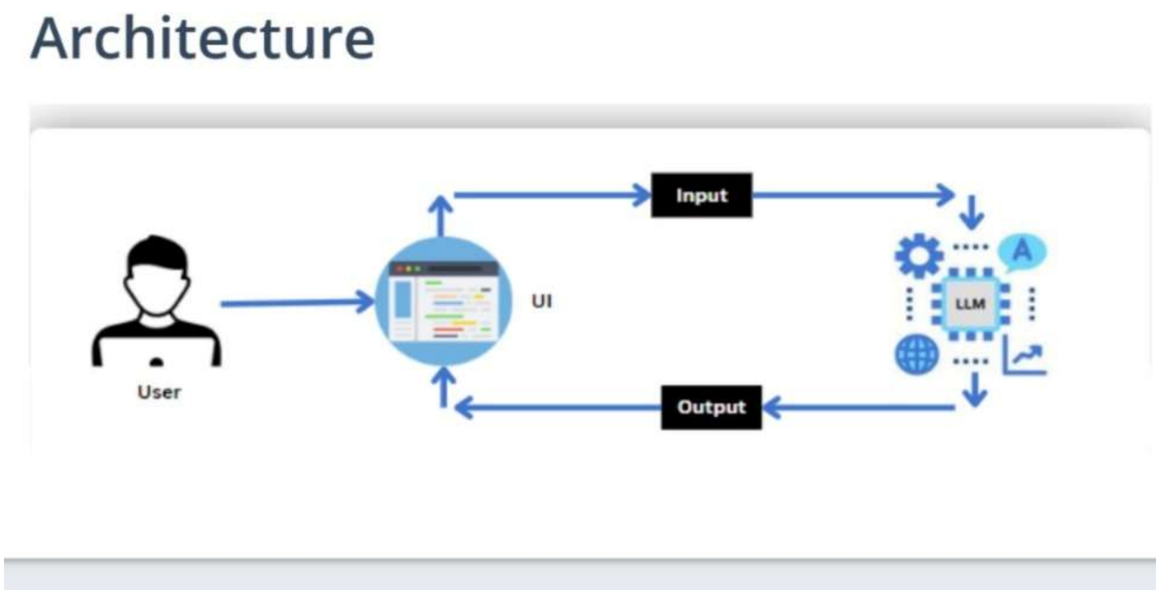
4.1 Problem-Solution Fit

Manual travel planning requires significant time and effort, often resulting in generic or incomplete itineraries. TravelGuideAI provides a strong problem-solution fit by automating the itinerary generation process using AI. This approach not only saves time but also enhances personalization, making travel planning more efficient and user-centric.

4.2 Proposed Solution

The proposed solution is an AI-driven system that collects user travel details and generates a comprehensive, day-wise itinerary. The system includes recommendations for attractions, activities, dining options, and travel tips. By leveraging generative AI, the solution adapts to different destinations and preferences, providing unique itineraries for each user.

4.3 Solution Architecture



5. PROJECT PLANNING & SCHEDULING

Phases:

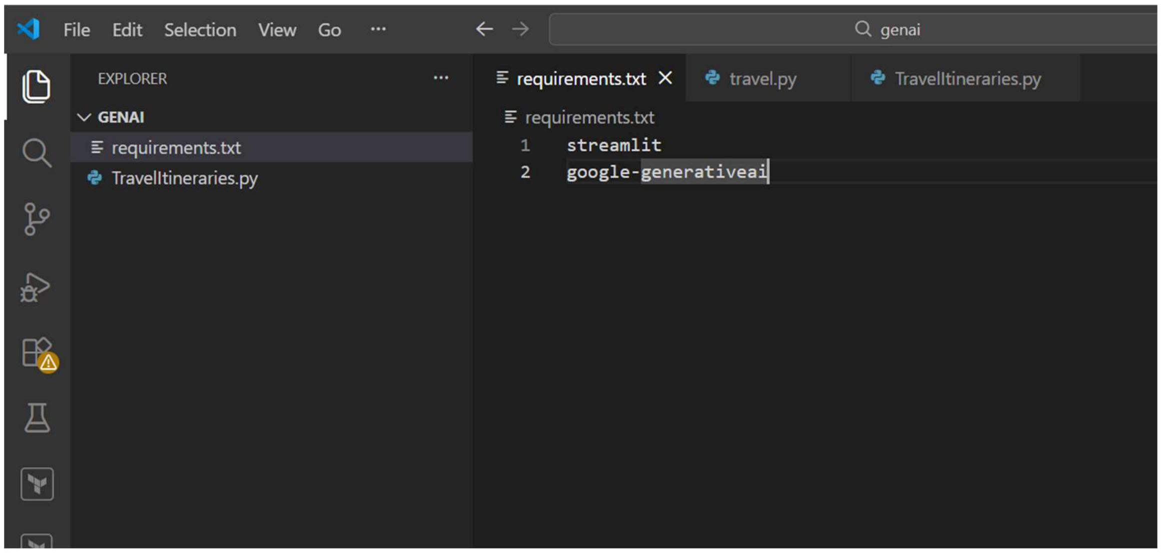
Research → Design → Development → Integration → Testing → Deployment

6. FUNCTIONAL AND PERFORMANCE TESTING

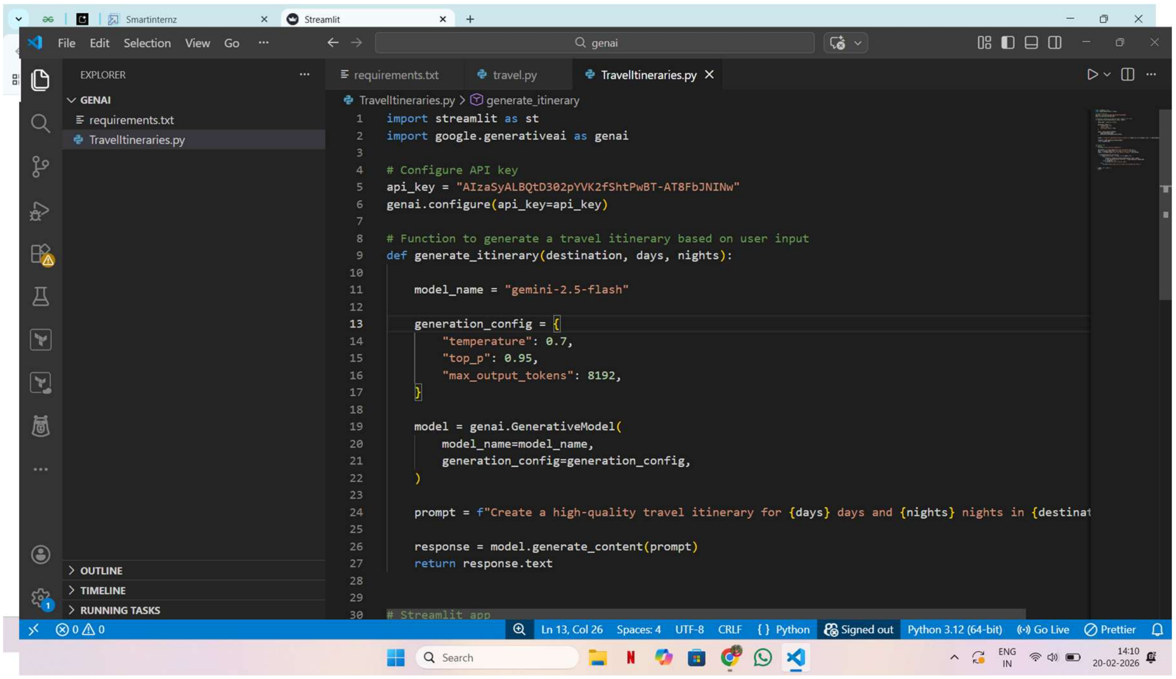
The application was tested with multiple destinations and varying travel durations to ensure reliability and accuracy. Functional testing verified correct input handling and itinerary generation, while performance testing ensured fast response times. The testing phase confirmed that the system produces structured and meaningful itineraries under different scenarios.

7. RESULTS

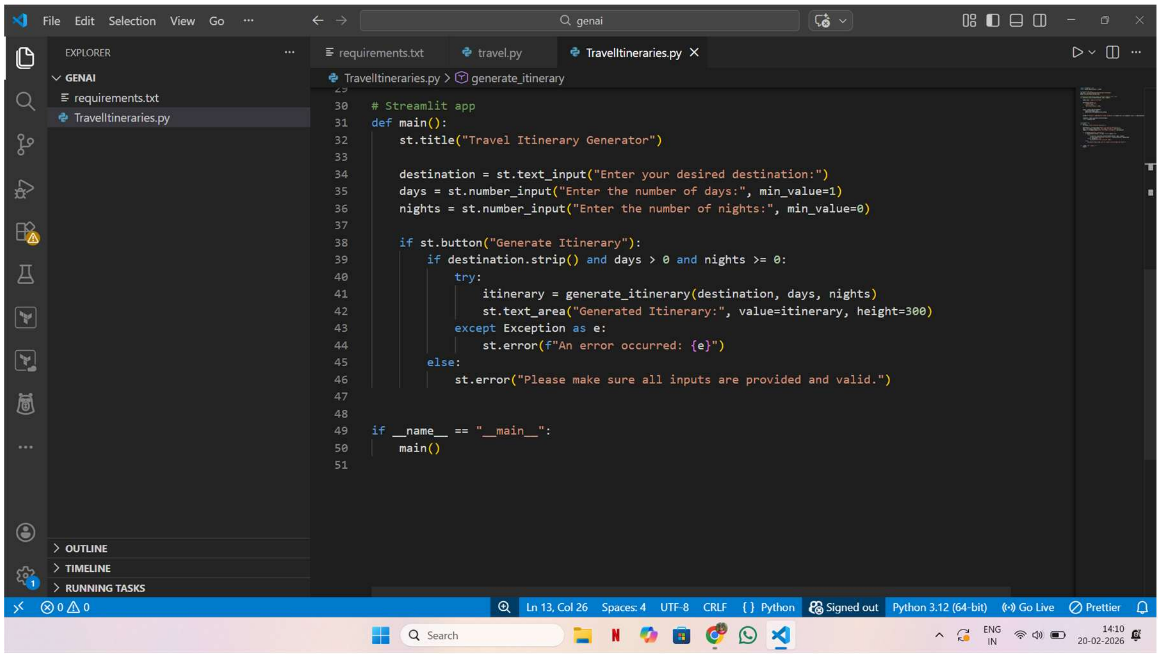
The application successfully generates detailed, personalized day-wise travel itineraries.



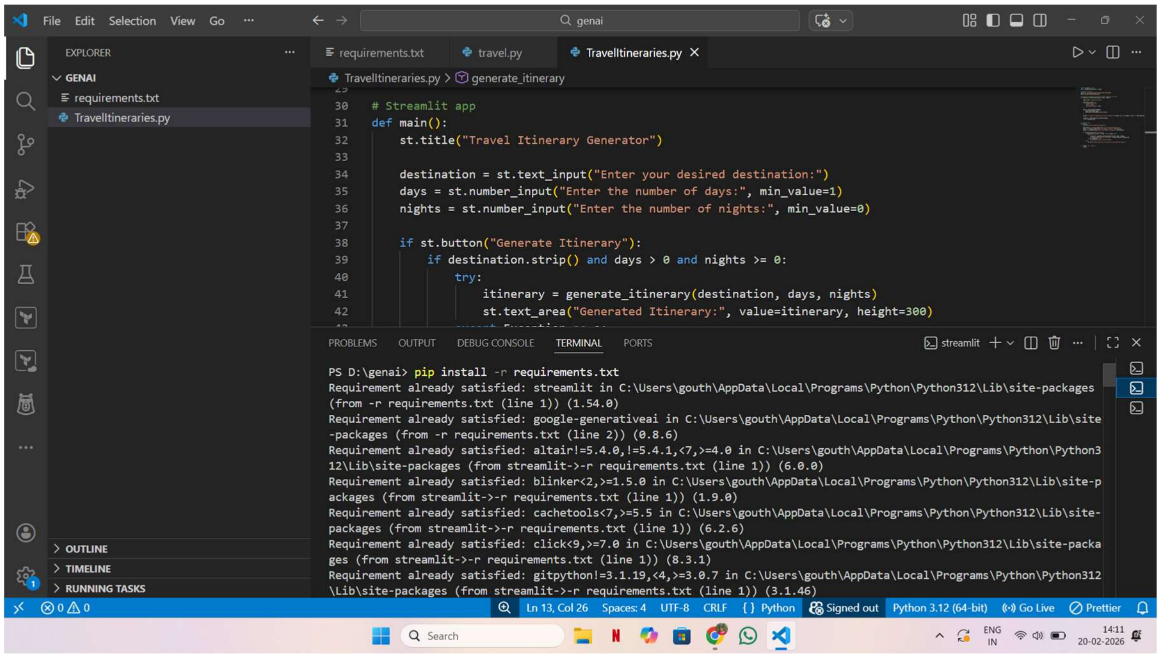
```
streamlit
google-generativeai
```



```python
import streamlit as st
import google.generativeai as genai

# Configure API key
api_key = "AIzaSyALBQtD302pYVK2fShtPwBT-AT8FbJNINw"
genai.configure(api_key=api_key)

# Function to generate a travel itinerary based on user input
def generate_itinerary(destination, days, nights):

    model_name = "gemini-2.5-flash"

    generation_config = {
        "temperature": 0.7,
        "top_p": 0.95,
        "max_output_tokens": 8192,
    }

    model = genai.GenerativeModel(
        model_name=model_name,
        generation_config=generation_config,
    )

    prompt = f"Create a high-quality travel itinerary for {days} days and {nights} nights in {destinat

    response = model.generate_content(prompt)
    return response.text


# Streamlit app
```



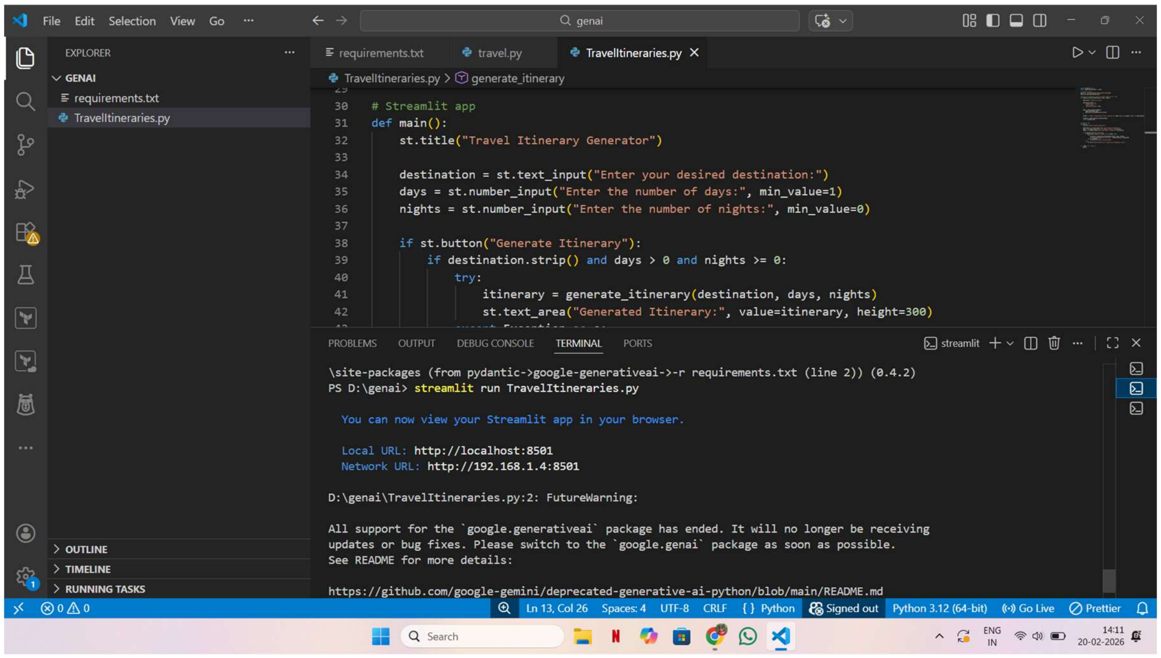
```python
# Streamlit app
def main():
    st.title("Travel Itinerary Generator")

    destination = st.text_input("Enter your desired destination:")
    days = st.number_input("Enter the number of days:", min_value=1)
    nights = st.number_input("Enter the number of nights:", min_value=0)

    if st.button("Generate Itinerary"):
        if destination.strip() and days > 0 and nights >= 0:
            try:
                itinerary = generate_itinerary(destination, days, nights)
                st.text_area("Generated Itinerary:", value=itinerary, height=300)
            except Exception as e:
                st.error(f"An error occurred: {e}")
        else:
            st.error("Please make sure all inputs are provided and valid.")


if __name__ == "__main__":
    main()
```



```python
    # Streamlit app
    def main():
        st.title("Travel Itinerary Generator")

        destination = st.text_input("Enter your desired destination:")
        days = st.number_input("Enter the number of days:", min_value=1)
        nights = st.number_input("Enter the number of nights:", min_value=0)

        if st.button("Generate Itinerary"):
            if destination.strip() and days > 0 and nights >= 0:
                try:
                    itinerary = generate_itinerary(destination, days, nights)
                    st.text_area("Generated Itinerary:", value=itinerary, height=300)
```

```
PS D:\genai> pip install -r requirements.txt
Requirement already satisfied: streamlit in C:\Users\gouth\AppData\Local\Programs\Python\Python312\Lib\site-packages
(from -r requirements.txt (line 1)) (1.54.0)
Requirement already satisfied: google-generativeai in C:\Users\gouth\AppData\Local\Programs\Python\Python312\Lib\site
-packages (from -r requirements.txt (line 2)) (0.8.6)
Requirement already satisfied: altair!=5.4.0,!=5.4.1,<7,>=4.0 in C:\Users\gouth\AppData\Local\Programs\Python\Python3
12\Lib\site-packages (from streamlit->-r requirements.txt (line 1)) (6.0.0)
Requirement already satisfied: blinker<2,>=1.5.0 in C:\Users\gouth\AppData\Local\Programs\Python\Python312\Lib\site-p
ackages (from streamlit->-r requirements.txt (line 1)) (1.9.0)
Requirement already satisfied: cachetools<7,>=5.5 in C:\Users\gouth\AppData\Local\Programs\Python\Python312\Lib\site-
packages (from streamlit->-r requirements.txt (line 1)) (6.2.6)
Requirement already satisfied: click<9,>=7.0 in C:\Users\gouth\AppData\Local\Programs\Python\Python312\Lib\site-packa
ges (from streamlit->-r requirements.txt (line 1)) (8.3.1)
Requirement already satisfied: gitpython!=3.1.19,<4,>=3.0.7 in C:\Users\gouth\AppData\Local\Programs\Python\Python312
\Lib\site-packages (from streamlit->-r requirements.txt (line 1)) (3.1.46)
```

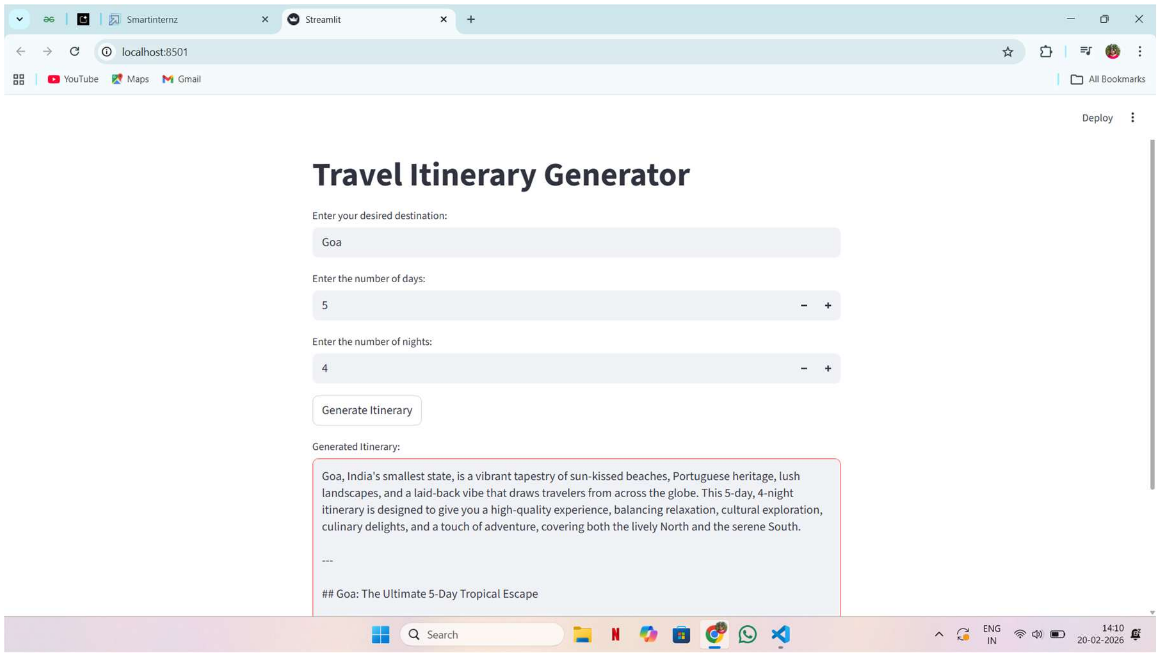
File Edit Selection View Go
genai
EXPLORER
GENAI
requirements.txt
TravelItineraries.py
travel.py
TravelItineraries.py > generate_itinerary
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
streamlit
OUTLINE
TIMELINE
RUNNING TASKS
Ln 13, Col 26 Spaces: 4 UTF-8 CRLF { } Python Signed out Python 3.12 (64-bit) Go Live Prettier
Search
ENG IN
14:11
20-02-2026



```python
    # Streamlit app
    def main():
        st.title("Travel Itinerary Generator")

        destination = st.text_input("Enter your desired destination:")
        days = st.number_input("Enter the number of days:", min_value=1)
        nights = st.number_input("Enter the number of nights:", min_value=0)

        if st.button("Generate Itinerary"):
            if destination.strip() and days > 0 and nights >= 0:
                try:
                    itinerary = generate_itinerary(destination, days, nights)
                    st.text_area("Generated Itinerary:", value=itinerary, height=300)
```

```
\site-packages (from pydantic->google-generativeai->-r requirements.txt (line 2)) (0.4.2)
PS D:\genai> streamlit run TravelItineraries.py

  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://192.168.1.4:8501

D:\genai\TravelItineraries.py:2: FutureWarning:

All support for the `google.generativeai` package has ended. It will no longer be receiving
updates or bug fixes. Please switch to the `google.genai` package as soon as possible.
See README for more details:

https://github.com/google-gemini/deprecated-generative-ai-python/blob/main/README.md
```

File Edit Selection View Go
genai
EXPLORER
GENAI
requirements.txt
TravelItineraries.py
travel.py
TravelItineraries.py > generate_itinerary
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
streamlit
OUTLINE
TIMELINE
RUNNING TASKS
Ln 13, Col 26 Spaces: 4 UTF-8 CRLF { } Python Signed out Python 3.12 (64-bit) Go Live Prettier
Search
ENG IN
14:11
20-02-2026



Travel Itinerary Generator

Enter your desired destination:

Goa

Enter the number of days:

5

Enter the number of nights:

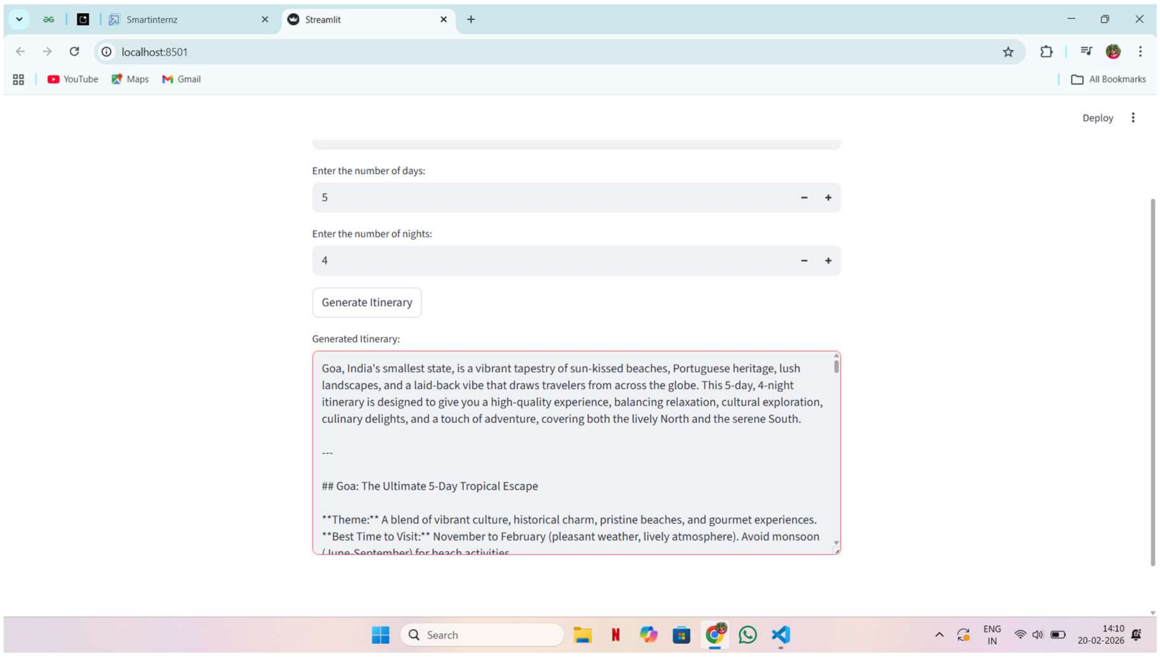
4

Generate Itinerary

Generated Itinerary:

Goa, India's smallest state, is a vibrant tapestry of sun-kissed beaches, Portuguese heritage, lush landscapes, and a laid-back vibe that draws travelers from across the globe. This 5-day, 4-night itinerary is designed to give you a high-quality experience, balancing relaxation, cultural exploration, culinary delights, and a touch of adventure, covering both the lively North and the serene South.

Goa: The Ultimate 5-Day Tropical Escape



Enter the number of days:

5

Enter the number of nights:

4

Generate Itinerary

Generated Itinerary:

Goa, India's smallest state, is a vibrant tapestry of sun-kissed beaches, Portuguese heritage, lush landscapes, and a laid-back vibe that draws travelers from across the globe. This 5-day, 4-night itinerary is designed to give you a high-quality experience, balancing relaxation, cultural exploration, culinary delights, and a touch of adventure, covering both the lively North and the serene South.

Goa: The Ultimate 5-Day Tropical Escape

Theme: A blend of vibrant culture, historical charm, pristine beaches, and gourmet experiences.
Best Time to Visit: November to February (pleasant weather, lively atmosphere). Avoid monsoon (June-September) for beach activities.

8. ADVANTAGES & DISA

DVANTAGES

Advantages:

- Saves time
- Personalized results
- Easy to use

Disadvantages:

- Requires internet connection
- Depends on API availability

9. CONCLUSION

The TravelGuideAI project demonstrates a practical and effective application of generative AI in the travel domain. By integrating AI with a user-friendly web interface, the system delivers customized travel itineraries efficiently. The project highlights how AI can enhance user experiences and automate complex planning tasks.

10. FUTURE SCOPE

Future enhancements can significantly expand the capabilities of TravelGuideAI. Planned improvements include integration with hotel and flight booking systems, budget-based itinerary filtering, multi-language support, and mobile application development. These enhancements will further increase usability and reach.

11. APPENDIX

GitHub Link:

https://github.com/priyanka-1624/Explore-With-AI-Custom-Itineraries-For-Your-Next-Journey.git

Demo Link:

https://drive.google.com/file/d/11Spj91YNeVl068pWv_02ac2wPH0B7V_1/view?usp=drive_link