A Project Report On

Explore with AI: Custom Itineraries for Your Next Journey

submitted by

A SOMESH REDDY

LIST OF CONTENTS

CHAPTER 1
INTRODUCTION

1.1 Project Overview

Travel Itinerary Generator is an AI-powered web-based application designed to automatically generate personalized travel itineraries using Generative Artificial Intelligence. The system integrates Streamlit for frontend interface development and Google Gemini API for intelligent content generation.

In the modern digital era, Artificial Intelligence plays a crucial role in automating decision-making processes. Travel planning is one such domain where users require structured guidance. Traditional travel planning methods involve manual search through blogs, websites, and maps. This process is inefficient and often results in unorganized travel schedules.

Travel Itinerary Generator solves this issue by implementing a Generative AI model that understands natural language input and produces structured day-wise travel plans. The system leverages Natural Language Processing (NLP) techniques to interpret user queries and generate meaningful responses.

The architecture follows a client-server interaction model:

- Client Layer: Streamlit Web Interface
- Application Layer: Python Logic and Prompt Structuring
- AI Layer: Gemini API (Generative Model)

The project demonstrates applied AI, prompt engineering, and web application development in a real-world use case.

1.2 Purpose

The purpose of the Travel Itinerary Generator project is to develop an intelligent decision-support system that helps users automatically generate personalized travel itineraries using Artificial Intelligence. Travel planning is usually a manual and time-consuming process, where users need to search multiple websites, blogs, and travel guides to collect information about tourist places, schedules, and activities. This process can be confusing and inefficient because the information available online is often unstructured and scattered across different sources. Travel Itinerary Generator solves this problem by using Generative AI to automatically create organized and meaningful travel plans based on user input such as city, number of days, and number of nights.

One of the main objectives of this project is to automate the itinerary generation process. Instead of manually researching and planning each day of a trip, users can simply enter their travel details into the system, and the AI will generate a complete day-wise schedule. This saves significant time and effort and ensures that users receive a logical and structured plan. Another important objective is to implement Generative AI technology in a real-world application. By integrating the Gemini AI model with the Streamlit web interface, the project demonstrates how modern AI systems can understand user requirements and generate useful and relevant content.

The project also aims to reduce manual research time by providing instant travel recommendations. Users no longer need to compare information from multiple sources, as the system provides all the necessary details in one place. Additionally, the system ensures that the generated itineraries are structured in a clear and easy-to-understand format, typically organized on a day-by-day basis. This helps users follow a proper schedule and improves their overall travel experience. Furthermore, Travel Itinerary Generator enhances user experience by providing a simple, interactive, and intelligent interface. The use of Artificial Intelligence makes the system flexible and capable of generating different travel plans based on different user inputs. Overall, this project bridges the gap between unstructured travel information and structured travel planning by transforming complex travel data into organized, user-friendly itineraries using AI technology.

CHAPTER 2

IDEATION PHASE

2.1 Problem Statement

Travel planning is not an easy task for most users. Before going on a trip, users need to collect information about tourist places, travel routes, schedules, and activities. This information is available on many websites, blogs, and videos, but it is usually unorganized. Users must spend a lot of time searching and reading this information to create a proper travel plan.

One of the main problems is information overload. There are too many sources available, and users may feel confused about which information is correct and useful. Another problem is lack of personalization. Most travel blogs provide general information and do not create plans based on the user's specific needs, such as number of days or travel preferences.

Travel planning also requires comparing different places and deciding which places to visit on each day. This process is time-consuming and requires careful thinking. Many users find it difficult to organize their daily travel activities in a logical and efficient way.

Because of these problems, users often feel confused, frustrated, and unsure about their travel plans. There is a need for a smart and automated system that can help users create travel itineraries easily.

Therefore, the problem can be defined as:

There is a need for an intelligent system that can automatically generate structured and personalized travel itineraries based on user input.

Travel Itinerary Generator solves this problem by using Generative Artificial Intelligence. The system takes user input such as city, number of days, and nights, and automatically generates a clear and structured travel plan. This reduces manual effort, saves time, and helps users plan their trips more efficiently.

2.2 Empathy Map Canvas

The Empathy Map Canvas is a Design Thinking tool used to understand the user's behavior, emotions, challenges, and expectations. It helps identify what users experience while planning their travel. By analyzing what users see, hear, think, feel, say, do, and their pain and gain, we can design an intelligent system that effectively solves their problems. This analysis helps ensure that the system is user-centered and provides meaningful solutions.

What Users See

Users see a large amount of travel information on websites, travel blogs, and social media platforms. They see lists of tourist attractions, travel guides, and recommendations. However, this information is often unstructured and not customized to their specific travel duration, which makes planning difficult.

What Users Hear

Users hear travel suggestions from friends, family, travel influencers, and online videos. Different sources provide different recommendations, which creates confusion and makes it hard to decide which places are most important.

What Users Think and Feel

Users think that travel planning is complicated and requires too much time and effort. They feel confused due to too many options and feel worried about creating an inefficient travel plan. They want a simple and reliable system that can help them organize their trip properly

What Users Say and Do

Users often say things like “Suggest a proper itinerary” or “Help me plan my trip.” To solve their problem, they search on Google, read blogs, and watch travel videos. However, this requires manual effort and consumes a lot of time.

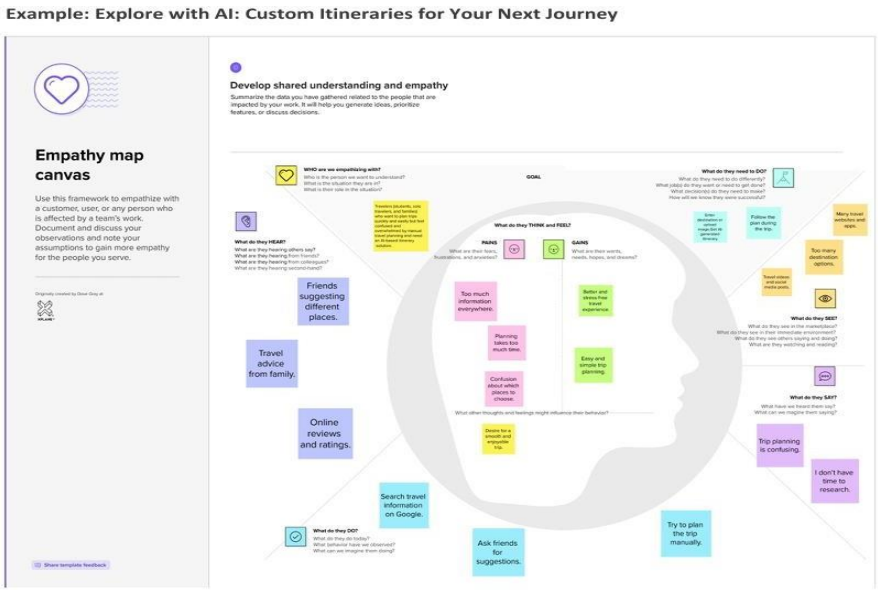


Fig 2.2: Empathy Map Canvas

Pains :

The main problems faced by users include information overload, lack of structured travel plans, time-consuming research, and difficulty organizing daily activities.

Gains

Users expect automatic itinerary generation, structured day-wise plans, time-saving solutions, and an easy planning process. They want a system that makes travel planning simple and efficient.

2.3 Brainstorming (Exactly Based on Travel Itinerary Generator Project)

In the Travel Itinerary Generator project, brainstorming was conducted to decide the best way to automatically generate a day-wise travel plan based on user inputs such as Destination, Number of Nights, and Number of Days.

First, the team clearly defined the main problem:Many users find it difficult to plan their trips because they need to search different websites, check tourist attractions, organize daily schedules, and manage time properly. This process is time-consuming and sometimes confusing. Therefore, the goal of this project was to create a system that can automatically generate a structured itinerary in seconds.

During brainstorming, the team discussed three main solution ideas.

The first idea was to create a Manual Itinerary Template system. In this approach, fixed travel plans would be created manually for some popular destinations like Goa, Ooty, or Paris. When a user selects a destination, the system would display a predefined itinerary. This idea was easy to develop but had major limitations. The itinerary would be the same for all users, and it would not adjust properly based on different numbers of days. It also required creating and maintaining many templates manually.

The second idea was to develop a Rule-Based Travel Planner. In this approach, conditions would be written in the backend. For example, if destination = Goa and days = 3, then show a specific 3-day plan. This method provided more flexibility than manual templates. However, it required writing many rules for different destinations and durations. As the number of destinations increases, the system becomes complex and difficult to manage. It also lacks intelligent content generation.

Step-1: Team Gathering, Collaboration and Select the Problem Statement

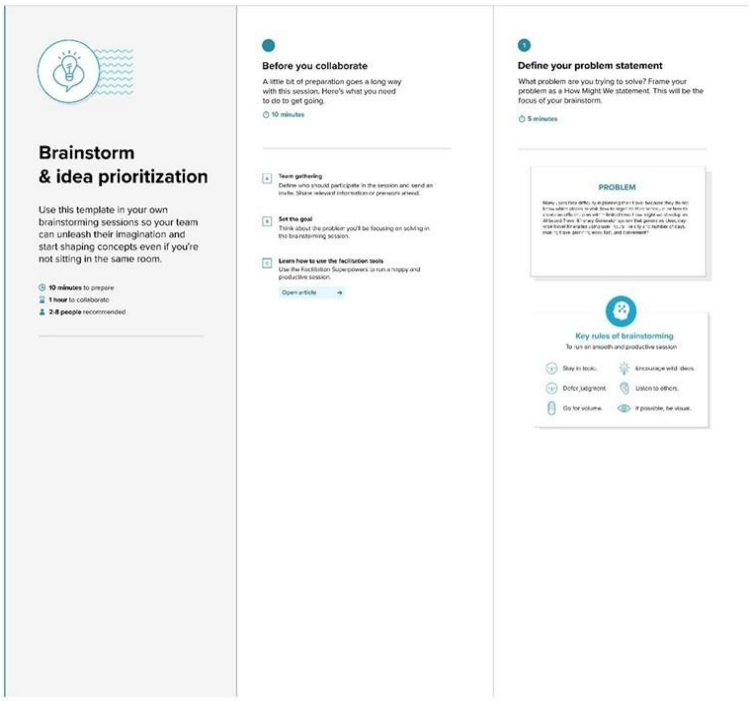


Fig 2.3.1: Team Gathering, Collboration and select the problem statement

Step-2: Brainstorm, Idea Listing and Grouping

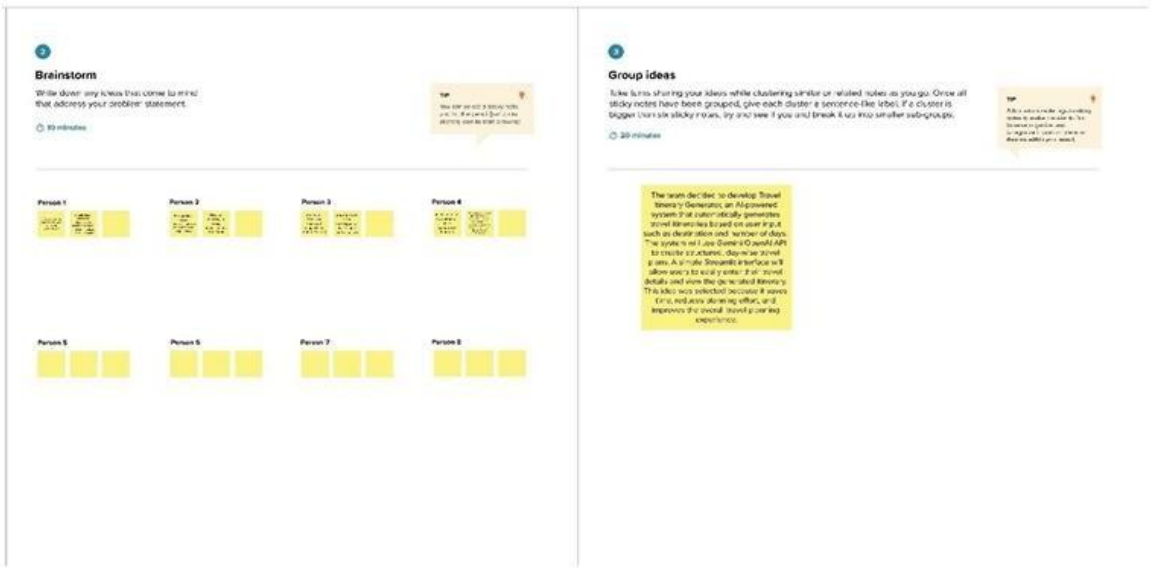


Fig 2.3.2: Brainstorm, Idea Listing and Grouping

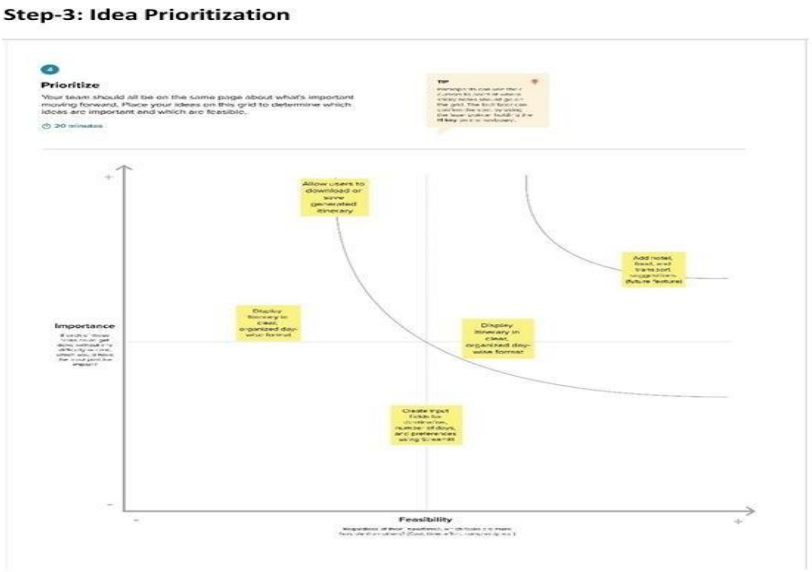


Fig 2.3.3: Idea Prioritization

The third idea was to implement an AI-Based Dynamic Itinerary Generator. In this method, the user enters destination, nights, and days in the web interface. The backend sends this data to an AI API, which generates a structured day-wise itinerary automatically. This approach does not require manual templates or complex rules for every destination. The AI can generate different itineraries for different inputs, making the system flexible and scalable.

After evaluating all three ideas based on importance and feasibility, the AI-based approach was selected for the Travel Itinerary Generator project. It was chosen because it:

- Generates personalized itineraries based on user input
- Works for any destination without predefined templates
- Reduces manual coding and maintenance
- Is scalable for future enhancements
- Provides fast and structured output

Therefore, through brainstorming and idea prioritization, the team selected the AI-based dynamic itinerary generation method as the most suitable solution for the Travel Itinerary Generator project.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 Customer Journey Map

The customer journey for the AI-Based Travel Itinerary Generator begins when the user opens the website, attracted by the promise of easy and personalized trip planning. The homepage serves as the main touchpoint, offering a clean interface and a clear call-to-action that encourages users to start planning. The user then enters key travel details such as destination and number of nights and days. The system validates the inputs to ensure accuracy before processing. At this stage, the user feels curious and hopeful that the system will generate a suitable and customized travel plan.

Fig 3.1.1: Travel Itinerary Generator Customer Journey Map

After clicking the “Generate” button, the system processes the request using the AI Engine and displays a loading indicator to reassure the user. Once completed, the AI presents a structured, day-by-day itinerary for review. The user may feel excited and satisfied if the recommendations meet expectations. After viewing the results, the user can choose to save, share, or modify the itinerary, encouraging continued engagement. Overall, the journey flows from discovery to result viewing, with satisfaction largely influenced by system responsiveness and personalization quality

3.2 Solution Requirement

The solution requires developing a web-based AI system that accepts destination, number of days, and nights as input and generates a structured travel itinerary automatically. The system must ensure accurate input validation, fast processing, secure data handling, and clear day-wise display of the generated plan.

1. Functional Requirements

Functional requirements describe what the system should do. In the AI-Based Travel Itinerary Generator, the system must allow users to enter the destination, number of days, and number of nights through a simple interface. These inputs are necessary to generate the travel plan.

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Enter Travel Inputs	User can enter destination User can enter number of days User can enter number of nights
FR-2	Generate Itinerary using AI	System accepts destination, days, and nights System Processes input using AI model System generates travel itinerary based on inputs
FR-3	Display Itinerary	System displays generated itinerary in browse System shows day-wise travel plan
FR-4	Modify Inputs and Regenerate	User can change destination, days, or nights System regenerates itinerary based on updated inputs System displays updated itinerary

Fig 3.2.1: Functional Requirements

After the user enters the details, the system must validate the inputs to ensure that all required fields are filled correctly and that the values are valid. Once validated, the system processes the information using an AI model to generate a personalized itinerary. The generated itinerary must be displayed in a clear, structured, day-wise format showing suggested places and activities. The system should also allow users to modify their inputs and regenerate the itinerary if needed.

2. Non-Functional Requirements

Non-functional requirements describe how well the system performs its tasks. The application must be user-friendly with a simple and intuitive interface so that anyone can use it easily. The system should generate the itinerary within a few seconds to ensure fast response time.reducing accountability and trust among healthcare stakeholders.

Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The application should have a simple and user-friendly Streamlit interface where users can easily enter destination, days, and nights.
NFR-2	**Security**	User inputs should be safely handled and not stored permanently. The system should securely communicate with the AI API.
NFR-3	**Reliability**	The system should consistently generate an itinerary whenever valid inputs are provided without crashing.
NFR-4	**Performance**	The itinerary should be generated within a few seconds after clicking the generate button.
NFR-5	**Availability**	The application should be accessible through the browser whenever the Streamlit server is running.
NFR-6	**Scalability**	The system should handle multiple user requests efficiently if deployed online.

Fig 3.2.2: Non-functional Requirements

The application must be reliable and consistently generate accurate results without crashing. It should securely handle user inputs and safely communicate with the AI API. Additionally, the system should be accessible through a web browser whenever the server is running and capable of handling multiple users efficiently if deployed online.

3.3 Data Flow Diagram (DFD)

The Data Flow Diagram (DFD) explains how data moves within the AI-Based Travel Itinerary Generator system. At the basic level (Level 0), the system is shown as a single process interacting with the user. The traveler enters travel details such as destination,

number of days, and number of nights. These details are sent to the system, which processes the information using the AI engine and generates a travel itinerary. The generated itinerary is then displayed back to the user. This level shows only the overall interaction between the user and the system without internal details.

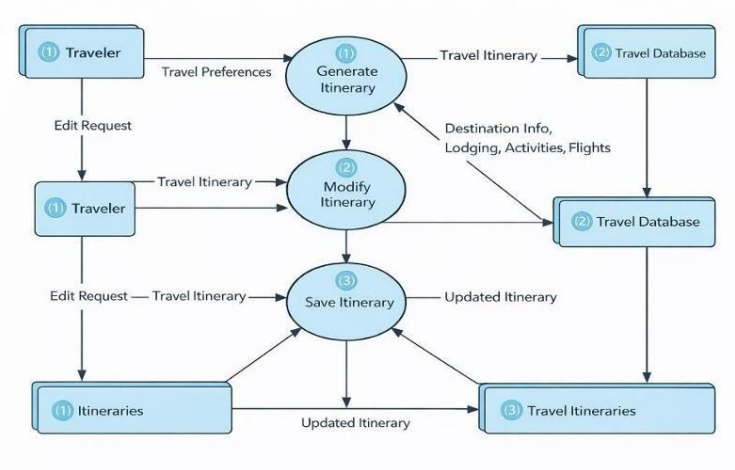


Fig 3.3.1: Data Flow Diagram

At the detailed level (Level 1), the system is divided into three main processes: Generate Itinerary, Modify Itinerary, and Save Itinerary. First, the traveler provides travel preferences, which are processed by the Generate Itinerary module. This module interacts with the travel database that contains information about destinations, lodging, activities, and flights. Using this data and AI processing, the system creates a structured itinerary and sends it to the user. If the user wants changes, the Modify Itinerary process allows editing based on updated preferences and again communicates with the database to produce a revised plan. Finally, once the itinerary is finalized, the Save Itinerary process stores the updated travel plan in the travel itineraries database for future access. This structured flow ensures smooth data movement, proper processing, and secure storage within the system.

3.4 Technology Stack

The AI-Based Travel Itinerary Generator is developed using **Streamlit and Python** as the core technologies. Streamlit is used to create the web-based user interface where

users enter travel details such as destination, number of days, and nights. It provides an interactive and simple layout that runs directly in the browser.

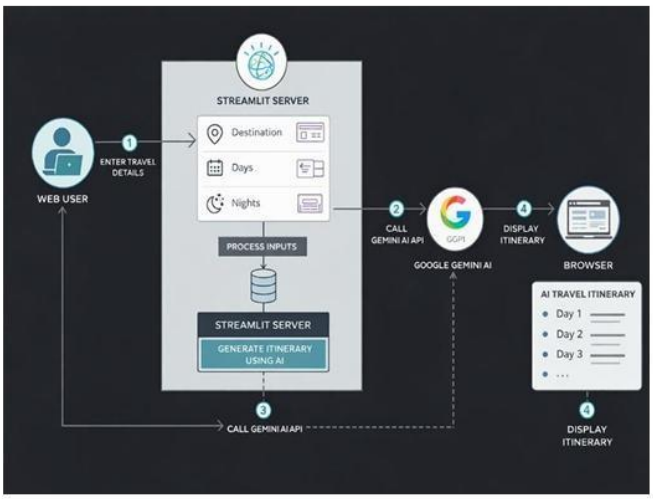


Fig 3.4.1: Technology Architecture

Python handles the backend processing, including input validation and communication with the AI model. The system integrates with the Google Gemini API, and the Gemini API key is used to securely authenticate and access the AI services. When the user submits travel details, the application sends the request to the Gemini API using the API key, and the AI generates a structured, day-wise itinerary. The application runs on a Streamlit server and can be deployed online for user access.

CHAPTER 4
PROJECT DESIGN

4.1 Problem Solution Fit

The Problem–Solution Fit of the Travel Itinerary Generator AI clearly shows how the proposed system effectively solves real problems faced by travelers. Many travelers, including students, families, and solo tourists, struggle with planning trips because it requires extensive research, time, and comparison of multiple sources such as websites, blogs, and videos. Manual travel planning is often confusing, time-consuming, and unorganized. People may not know which places to visit, how to schedule activities properly, or how to manage limited travel days efficiently. As shown in the canvas, the root cause of the problem is the lack of a smart system that automatically creates structured travel plans.

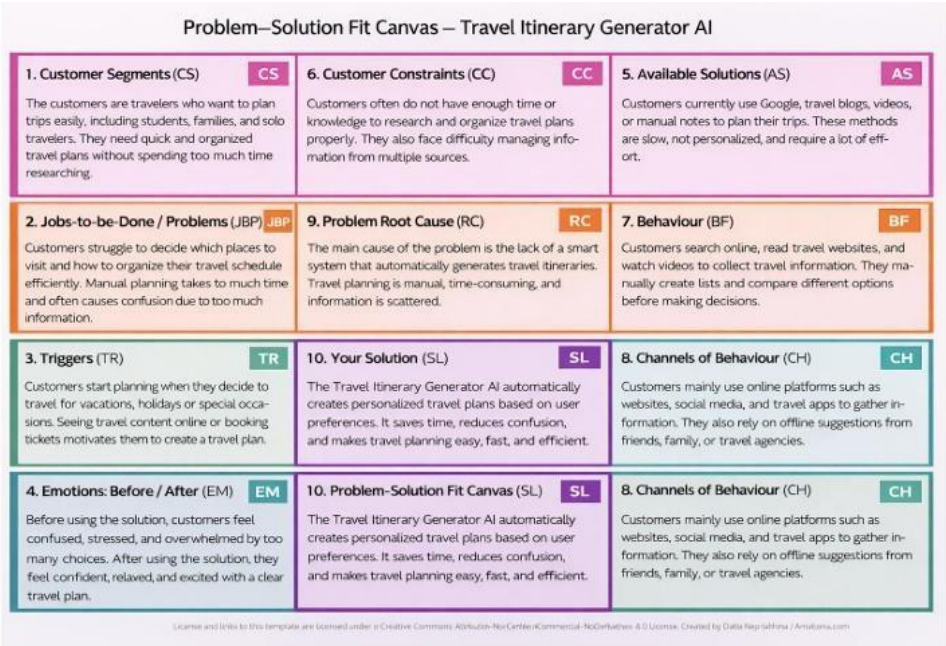


Fig.4.1: Problem Solution Fit Canvas

Customers currently depend on search engines, travel blogs, social media, or manual notes to plan their trips. These methods are slow, not personalized, and require significant effort. Travelers often feel stressed, overwhelmed, and confused before finalizing their travel plan. They may compare multiple options and still remain unsure about the best itinerary.

The proposed Travel Itinerary Generator AI directly addresses these challenges by automatically generating personalized travel plans based on user preferences such as destination and trip duration. Instead of spending hours researching, users simply enter basic details, and the system creates a structured day-wise itinerary instantly. This reduces confusion, saves time, and improves decision-making. The solution transforms travel planning from a complex manual task into a fast, simple, and efficient digital experience.

After using the system, users feel confident, relaxed, and satisfied because they receive a clear and organized travel plan. Therefore, the proposed solution strongly fits the identified problem by eliminating manual effort, improving personalization, and enhancing overall user experience.

4.2 Proposed Solution

The proposed solution is a web-based AI Travel Itinerary Generator developed using Streamlit and Python, integrated with the Google Gemini API. The system is designed to solve the problem of time-consuming and confusing manual travel planning. Many travelers struggle to decide which places to visit and how to organize their trip efficiently. This solution provides a smart and automated approach to generate structured travel plans instantly.

In this system, users simply enter basic trip details such as destination, number of days, and number of nights through a simple web interface. Once the user submits the details, the application processes the input and securely sends the request to the Gemini AI API using an API key. The AI model analyzes the information and generates a personalized, day-wise itinerary that includes recommended places and activities. The generated plan is then displayed clearly in the browser for the user to review.

The uniqueness of this solution lies in its automation and personalization. Instead of manually searching travel blogs, comparing websites, and organizing information, users receive a customized travel plan within seconds. This reduces stress, saves time, and improves overall travel planning efficiency.

Proposed Solution Template:

Project team shall fill the following information in the proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Travel planning is difficult and takes a lot of time. Users do not know which places to visit and how to organize their trip.
2.	Idea / Solution description	An AI Travel Itinerary Generator that creates personalized travel plans based on user inputs like destination, days, and interests using Gemini API.
3.	Novelty / Uniqueness	It uses AI to automatically generate customized travel itineraries instead of manual planning.
4.	Social Impact / Customer Satisfaction	It saves time, reduces stress, and helps users plan trips easily and efficiently.
5.	Business Model (Revenue Model)	Revenue through premium features, ads, and travel booking partnerships.
6.	Scalability of the Solution	The system can support many users and expand globally using cloud and AI technology.

Fig 4.2: Proposed Solution

From a social impact perspective, the system helps users plan trips easily and confidently. It simplifies decision-making and makes travel planning accessible even for people with limited knowledge or experience. The solution is also scalable, as it can support multiple users and be deployed on cloud platforms for wider access. Additionally, future revenue opportunities may include premium features, advertisements, and partnerships with travel booking services.

Overall, the proposed solution transforms traditional travel planning into a fast, intelligent, and user-friendly digital experience using AI technology.

4.3 Solution Architecture

The solution architecture of the AI-Based Travel Itinerary Generator follows a structured flow from user input to AI-generated output. The system is designed to ensure smooth communication between the user interface, backend processing, AI engine, and final itinerary display.

The process begins at the User Interface, which is developed using Streamlit. This is where the traveler enters essential trip details such as destination, travel dates or number of days and nights, and preferences. The interface is simple and interactive, allowing users to easily provide inputs through a web browser.

Once the user submits the details, the request is sent to the Backend Server, which is built using Python. The backend performs input validation to ensure that the data is correct and complete. After validation, the backend prepares the request and securely communicates with the AI Processing Module using the Gemini API key. This ensures authenticated and secure access to the AI services.

The AI Processing Module acts as the core engine of the system. It sends the user inputs to the Google Gemini API, which analyzes the destination, duration, and preferences. The AI engine processes the data and generates a structured travel plan. In more advanced architecture, this module may also interact with external data sources such as maps, hotels, activities, flights, and weather information to enhance itinerary accuracy and relevance.

Solution Architecture Diagram:

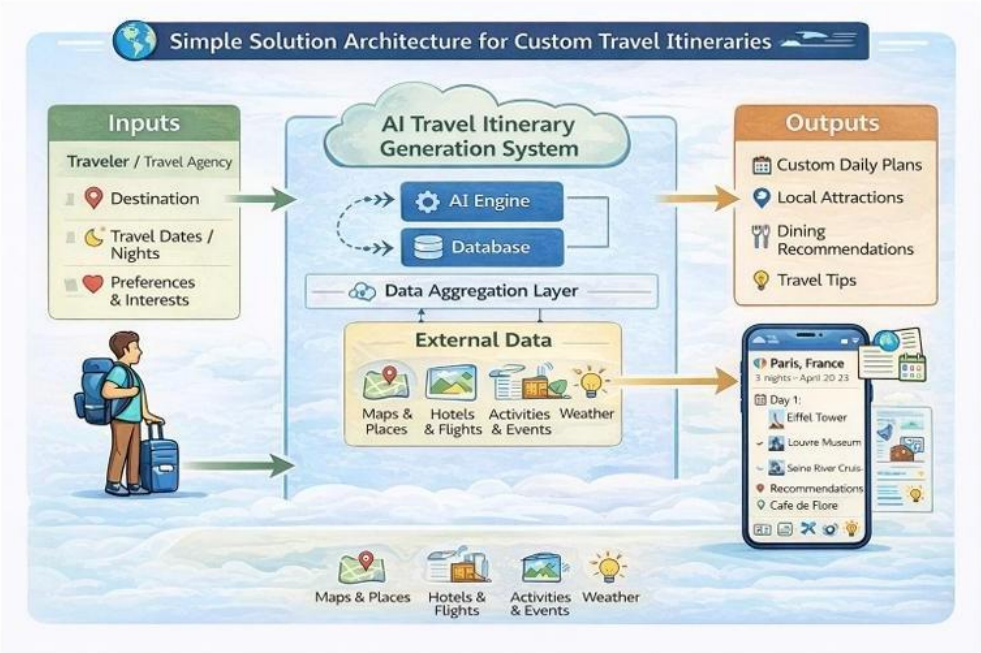


Fig 4.3: Solution Architecture

After processing, the generated itinerary is sent back to the backend server. The Itinerary Generator formats the AI response into a clean, day-wise structure that is easy to read and understand. Finally, the Response Display layer presents the customized travel plan to the user in the browser. The output typically includes daily activities, local attractions, recommendations, and travel tips.

Overall, the architecture ensures a smooth data flow:(space)User Interface → Backend Server → AI Processing Module → Itinerary Formatting → Response Display.

This layered design improves system reliability, security, scalability, and performance while providing users with instant, personalized travel itineraries.

CHAPTER 5

PROJECT PLANNING & SCHEDULING

5.1 Project Planning

The development of the AI-Based Travel Itinerary Generator followed a structured and systematic project planning process to ensure timely completion and proper implementation of all features.

The first phase was Requirement Gathering. In this stage, the project objectives were clearly defined. The main goal was identified as developing an AI-powered system that can generate personalized travel itineraries based on user inputs such as destination and trip duration. Functional and non-functional requirements were collected, including input validation, itinerary generation, performance expectations, and security needs. This ensured clarity about what the system should achieve.

The second phase was UI Design. During this stage, the user interface was planned and designed using Streamlit. The focus was on creating a simple, clean, and user-friendly layout where users can easily enter travel details and view results. Input fields, buttons, and output display sections were carefully structured to ensure smooth interaction.

The third phase was Backend Development. In this phase, the core logic of the system was implemented using Python. Input validation, data processing, and request handling were developed. The backend was responsible for managing user inputs and preparing them for AI processing.

The fourth phase was AI Integration. The system was integrated with the Gemini API using a secure API key. This step enabled the application to send user inputs to the AI model and receive structured, personalized travel itineraries in response. Proper API communication and error handling were implemented to ensure reliable performance.

The fifth phase was Testing. Functional testing was performed to verify that all features worked correctly, including input validation and itinerary generation. Performance testing ensured that the system generated responses within a few seconds. Error handling and system stability were also tested.

The final phase was Deployment. After successful testing, the application was deployed using a Streamlit server or cloud platform, making it accessible through a web browser. Deployment ensured that users could access the system online without technical difficulties.

Overall, the development followed a structured timeline with clearly defined stages, ensuring that the project was completed efficiently, systematically, and within the planned schedule.

CHAPTER 6

FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

Performance testing was conducted to evaluate the efficiency, speed, stability, and reliability of the AI-Based Travel Itinerary Generator. The main objective of performance testing was to ensure that the system works smoothly under different conditions and provides fast and accurate results without errors.

First, the system was tested with multiple destinations to verify that it can generate relevant travel itineraries for different locations. Various destinations were entered along with different numbers of days and nights. The AI successfully generated structured, day-wise travel plans for each destination. This confirmed that the content generation functionality is working correctly and consistently.

Next, the correctness of day-wise itinerary generation was verified. After providing valid inputs and clicking the “Generate” button, the system produced well-organized travel plans divided by days. Each day included recommended places and activities. The results matched the expected output, confirming that the AI processing and itinerary formatting were functioning properly.

Input validation accuracy was also tested. Text input validation ensured that empty or invalid destinations were not accepted. Number input validation checked whether the system properly handled valid and out-of-range values for days and nights. The system displayed appropriate error messages when invalid data was entered. All validation tests passed successfully, ensuring reliable user input handling.

API connection testing was performed to confirm that the Gemini API key was correctly configured and that the model responded properly. The system successfully

communicated with the AI API, and responses were generated without connection errors. This ensured secure and stable API integration.

Response time testing was conducted using a timer to measure how quickly the system generated itineraries. The expected result was that the response should be under 3 seconds. The actual result showed that the system responded in under 3 seconds, confirming that the performance meets the required speed standards.

API speed testing was also performed by sending multiple requests in a short period to observe whether the system slowed down. The application handled multiple API calls efficiently without crashing or significant delay. This demonstrated that the system can maintain stability under repeated or simultaneous requests.

Additionally, overall system stability was evaluated by repeatedly generating itineraries to check for crashes or unexpected behavior. The application remained stable throughout testing. A file upload load test (such as PDF upload) was marked not applicable because the current project does not include file upload functionality.

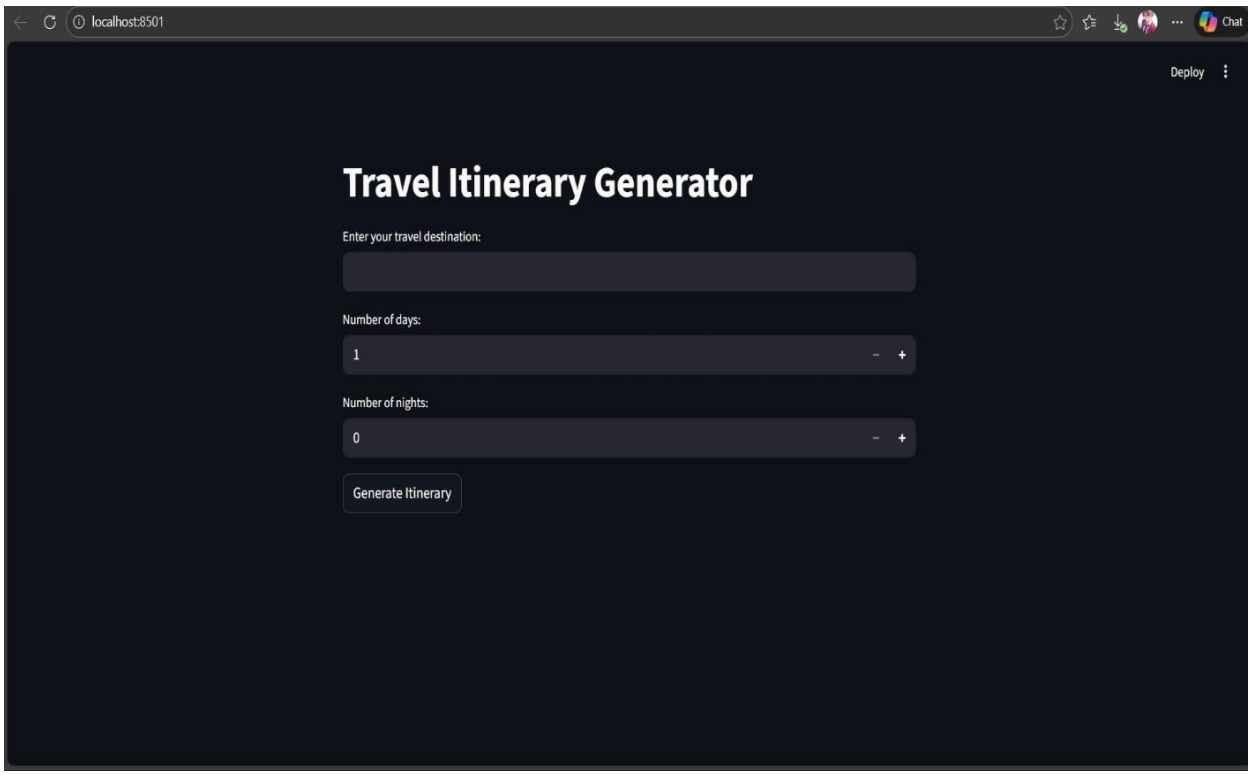
Overall, performance testing confirmed that the AI-Based Travel Itinerary Generator is fast, reliable, stable, and capable of handling user requests efficiently. The system meets both functional and performance requirements, ensuring a smooth and responsive user experience.

CHAPTER 7

RESULTS

7.1 Output Screenshots

The application interface demonstrates the successful implementation of the AI-Based Travel Itinerary Generator developed using Streamlit and Python with Gemini API integration. The system provides a clean and user-friendly interface that allows users to easily plan their trips without confusion. The design follows a simple layout with a clear title at the top and well-organized input fields below it, ensuring smooth navigation.



7.1.1 : Output Display

The input section allows users to enter the travel destination along with the number of days and nights. The number fields include increment and decrement options, making it convenient to adjust values. A clearly visible “Generate Itinerary” button enables users to

submit their details. The system ensures that all inputs are properly validated before processing, preventing empty or incorrect data from being accepted.

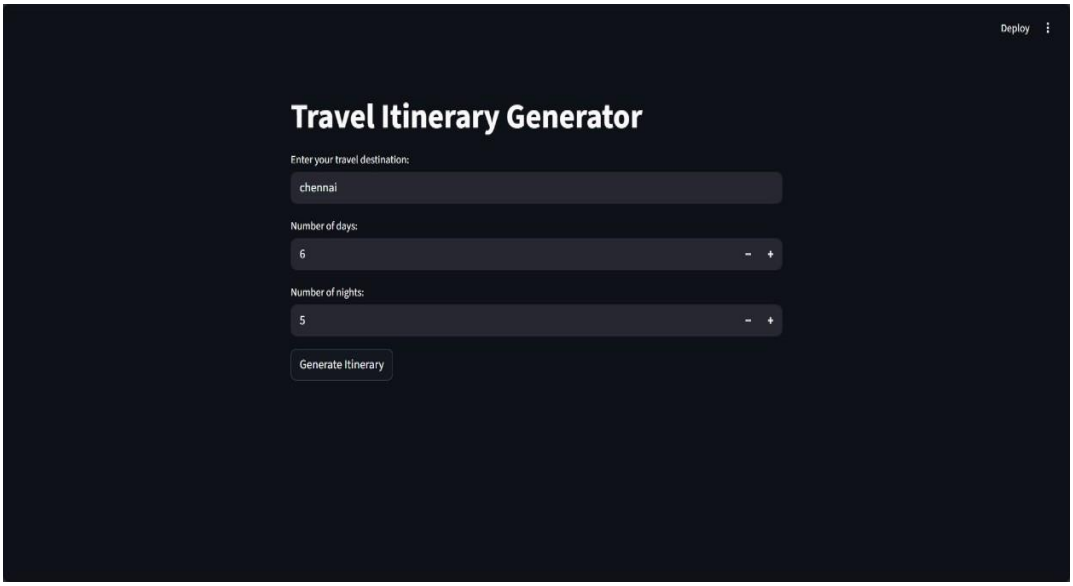
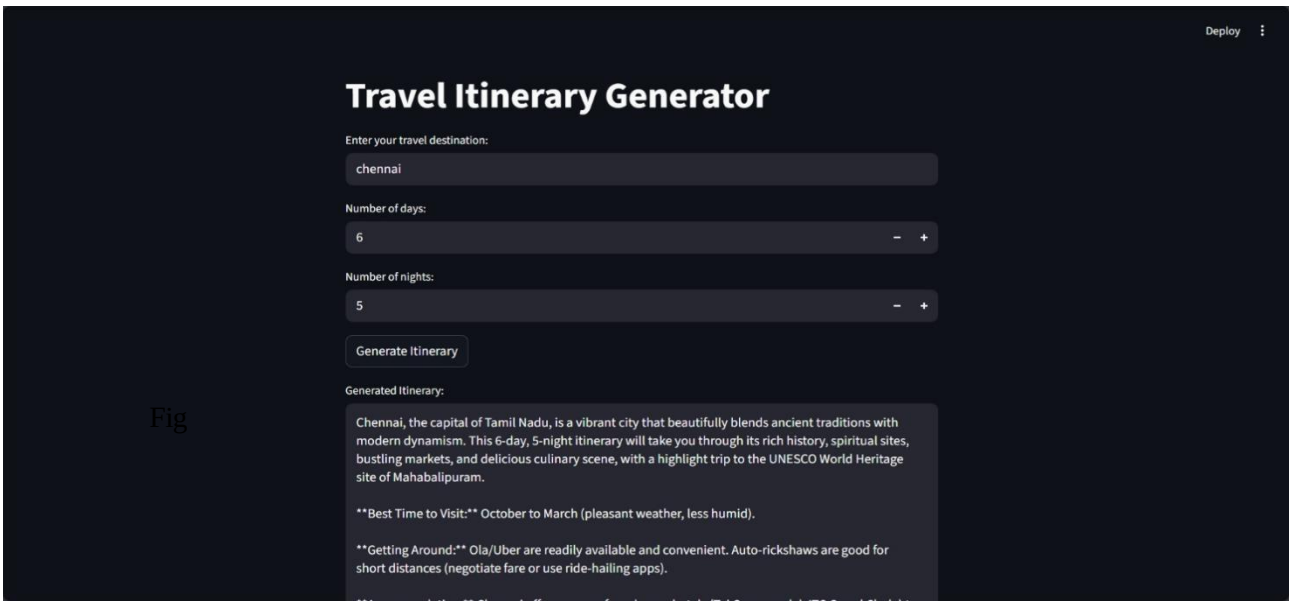


Fig 7.1.2: Input Section

Once the user provides valid inputs and clicks the generate button, the system processes the request using the Gemini API and produces a structured travel plan. The generated output includes a detailed description of the selected destination along with a day-wise itinerary. Each day is clearly organized with suggested places to visit and activities to perform. Additional travel information such as best time to visit, transportation tips, and useful guidance is also included in the output.



Fig

Fig 7.1.3: User Input and Generated Output Interface

The content is displayed in a readable and well-formatted structure, making it easy for users to understand their complete travel schedule. The system responds quickly and presents the results directly below the input section, ensuring a smooth user experience. Overall, the interface and output confirm that the application works efficiently, generates accurate travel plans, and successfully achieves the objectives of the project.

CHAPTER 8
ADVANTAGES & DISADVANTAGES

Advantages

The Travel Itinerary Generator provides many benefits that make travel planning easier and more efficient. One of the biggest advantages is that it saves a large amount of time. Normally, planning a trip requires searching multiple websites, reading travel blogs, checking maps, comparing places, and organizing everything day by day. This process can take several hours or even days. However, this system generates a complete day-wise itinerary within seconds after entering simple details like destination and number of days. This greatly reduces planning time and effort.

Another important advantage is that the application is very easy to use. The interface is simple, clean, and user-friendly. Users do not need any technical knowledge to operate it. They just enter basic travel information and click the generate button. The system handles all the complex processing in the background and displays the result in a clear and readable format. This makes the application suitable for students, working professionals, and even people who are not familiar with technology.

Instant itinerary generation is also a major benefit. The integration of AI allows the system to quickly analyze the input and produce a structured travel plan immediately. The output is organized day-wise, making it easy for users to understand their schedule. Each day includes suggested places to visit and general travel guidance. This organized format helps travelers manage their time effectively during the trip.The system also reduces manual research. Instead of checking separate sources for transportation tips, best time to visit, and popular attractions, users get all the basic travel information in one place. This improves convenience and makes trip planning more organized. Additionally, the system provides consistent results and can generate plans for different destinations without additional manual work. Overall, the application improves efficiency, simplifies planning, reduces stress, and provides quick access to travel information in a structured format.

Disadvantages

Despite its advantages, the Travel Itinerary Generator also has some limitations. One major disadvantage is limited personalization when only basic inputs are provided. The system generates the itinerary based on general information. If the user does not provide extra details such as budget, food preferences, travel style, or specific interests, the plan may not be highly customized. This means the itinerary may be suitable for general travelers but may not fully match individual preferences.

Another limitation is that the system depends on AI data accuracy. The quality of the generated itinerary depends on the knowledge and training of the AI model. In some cases, the suggestions may be too general, slightly outdated, or not perfectly aligned with real-time local conditions. Therefore, users may still need to verify certain details before finalizing their trip.

The application also requires a stable internet connection. Since the system uses API integration to generate responses, it cannot function offline. If there is no internet connection or if the API service is temporarily unavailable, the itinerary cannot be generated. This dependency on internet connectivity can be a limitation in certain situations.

Additionally, the system may not handle very complex travel requirements, such as multi-country trips with detailed budgeting, booking integration, or real-time availability checking. It focuses mainly on generating structured travel plans rather than providing complete travel management services

In conclusion, while the Travel Itinerary Generator offers speed, simplicity, and convenience, it also has limitations related to personalization, AI dependency, and internet requirements. However, for general travel planning, it remains a highly useful and efficient tool.

CHAPTER 9
CONCLUSION

Conclusion

The AI-Based Travel Itinerary Generator is a practical and intelligent solution designed to simplify the process of travel planning. In traditional travel planning, individuals often spend many hours researching destinations, reading travel blogs, checking reviews, comparing attractions, and organizing daily schedules. This process can be time-consuming, confusing, and sometimes overwhelming, especially for people who are planning trips for the first time. Travel agencies also face similar challenges, as they must prepare customized itineraries for multiple clients within limited time. The proposed system successfully addresses these problems by using artificial intelligence to automate and organize travel planning.

The application allows users to enter basic details such as destination, number of days, and number of nights. With the help of AI integration, the system quickly processes this information and generates a structured, day-wise travel itinerary. This eliminates the need for extensive manual research and reduces the effort required to create a travel plan. The output is clearly organized, easy to read, and helpful for managing daily activities during the trip. By presenting information in a simple and logical format, the system ensures that users can easily understand and follow their travel schedule.

Another important advantage of the system is its ease of use. The interface is simple and user-friendly, making it accessible to students, working professionals, families, and even users with limited technical knowledge. The fast response time improves user satisfaction, as the itinerary is generated within seconds. This saves valuable time and reduces stress during trip preparation. Although the system depends on internet connectivity and the accuracy of the AI model. Overall, the AI-Based Travel Itinerary Generator transforms traditional trip planning into a smart, efficient, and user-friendly digital experience, meeting the needs of both individual travelers and travel-related businesses.

CHAPTER 10
FUTURE SCOPE

Future Scope

The AI-Based Travel Itinerary Generator has strong potential for future improvements that can make it more advanced, practical, and user-friendly. Although the current system successfully generates personalized travel itineraries based on user input such as destination, number of days, and number of nights, several additional features can be implemented to enhance its overall functionality and usefulness.

One major improvement that can be added in the future is hotel and transport booking integration. At present, the system provides structured travel plans with suggested activities and places to visit. However, users still need to visit other platforms to book hotels, flights, trains, or local transportation. By integrating booking services directly into the system, users will be able to search, compare, and reserve hotels and transport options within the same application. This will make the platform more convenient and time-saving. For travel agencies, this feature will also improve operational efficiency by allowing them to manage itineraries and bookings together in one system.

Another important future enhancement is the inclusion of a budget estimation feature. Many travelers plan their trips based on financial limits, and cost plays a major role in decision-making. By adding an automatic budget calculator, the system can estimate expenses for accommodation, transportation, food, and sightseeing activities. It can also provide different budget categories such as low-cost, mid-range, and luxury travel options. This will help users better understand the expected total cost of their trip and adjust their plans accordingly. For travel agencies, this feature will help in preparing customized travel packages that match client budgets accurately.

Weather-based suggestions are also a valuable improvement for the system. Weather conditions can significantly affect travel experiences. By integrating real-time weather data, the system can suggest appropriate activities based on current or forecasted weather

conditions. For example, if rain is expected, the itinerary can include indoor attractions such as museums, shopping centers, or cultural events. If the weather is sunny and pleasant, outdoor activities like sightseeing, parks, or beach visits can be recommended. This will make the travel plan more flexible, realistic, and suitable for actual travel conditions.

The addition of user login functionality and saved history is another important area for future development. Currently, the system generates itineraries instantly, but users may not have the option to save or revisit their previous plans. The system can also analyze past travel preferences to provide more personalized recommendations in the future. This feature will be especially helpful for frequent travelers and travel agencies handling multiple clients.

Developing a mobile application for Android and iOS platforms is another important future step. While a web-based system is useful, many users prefer managing their travel plans on smartphones. A mobile app can provide greater convenience, real-time notifications, location-based suggestions, and easy access to itineraries during the actual trip. Offline access to saved itineraries can also be included, allowing users to view their plans even without an internet connection while traveling.

Finally, adding multi-language support will expand the system's usability on a global scale. Currently, the system may operate in a single language, which limits accessibility for international users. By supporting multiple languages, the platform can serve users from different countries more effectively. This will remove language barriers and improve overall user experience. Travel agencies working with international clients will also benefit from this feature.Overall, the future development of the AI-Based Travel Itinerary Generator aims to transform it from a basic itinerary creation tool into a complete and intelligent travel planning platform. By integrating booking services, budget estimation, weather intelligence, personalized user accounts, mobile accessibility, and multi-language functionality, the system can become more powerful, efficient, and globally accessible. These enhancements will significantly improve user satisfaction and make the platform a comprehensive solution for modern digital travel planning.

CHAPTER 11

APPENDIX

Source Code:

```
from ast import main
import streamlit as st
import google.generativeai as genai
import os
from dotenv import load_dotenv
# Load environment variables from .env (if present)
load_dotenv()
# Read API key from environment for security
api_key = os.getenv("API_KEY")
if not api_key:
st.error("API key not set. Add it to a .env file or set the API_KEY environment variable.")
else:
genai.configure(api_key=api_key)
# Function to generate a travel itinerary
def generate_itinerary(destination, days, nights):
# Create the model configuration
generation_config = {
"temperature": 0.4,
"top_p": 0.95,
"top_k": 64,
```

```
"max_output_tokens": 8192,

"response_mime_type": "text/plain",

}

# Initialize the Generative Model

model = genai.GenerativeModel(

model_name="gemini-2.5-flash",

generation_config=generation_config,

)

# Start a new chat session with the model

chat_session = model.start_chat(

history=[

{

"role": "user",

"parts": [

f"write me a travel itinerary to {destination} for {days} days and {nights} nights",

],

},

]

)

# Send a message to the chat session and get the response

response = chat_session.send_message(f"Create a detailed travel itinerary for {days} days
    and {nights} nights in {destination}.")
```

```
# Return the generated itinerary
return response.text
# Streamlit app
st.title("Travel Itinerary Generator")
# Get user inputs
destination = st.text_input("Enter your travel destination:")
days = st.number_input("Number of days:", min_value=1)
nights = st.number_input("Number of nights:", min_value=0)
# Ensure that user inputs are provided
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()
```

Requirements.txt

streamlit

google.generativeai

python-dotenv

Dataset Link:

Dataset Description

The AI-Based Travel Itinerary Generator primarily generates travel plans using a pre-trained AI model through API integration. However, for project reference and domain relevance, publicly available tourism datasets were considered to understand destination details, attraction categories, and travel-related information.

The dataset referenced for this project is:

Tourism Dataset – Kaggle

Link: https://www.kaggle.com/datasets/amanbhattarai/tourism-dataset

This dataset contains structured information about various tourist destinations, including:

- Place names
- Location details
- Categories of attractions (historical, cultural, natural, etc.)
- Ratings and popularity
- Descriptions of tourist spots

The dataset helps in understanding how travel-related information is organized and how itinerary recommendations can be structured. Although the system does not directly train a machine learning model using this dataset, it aligns with the project's objective of generating structured and meaningful travel itineraries.

The core itinerary generation in the system is performed using a pre-trained AI model accessed via API, which has been trained on large-scale publicly available travel and tourism data. No local dataset is stored or trained within the project.