

# **LONG TERM INTERNSHIP**

**Team ID: LTVIP2026TMIDS81651**

## **1.INTRODUCTION**

The rapid growth of digital technology and artificial intelligence has significantly transformed how people plan and experience travel. Modern travelers seek personalized, quick, and reliable travel recommendations without spending hours researching destinations, accommodations, and activities. Traditional travel planning methods often involve browsing multiple websites, reading reviews, comparing itineraries, and manually organizing plans, which can be time-consuming and overwhelming.

To address these challenges, the TravelGuideAI project leverages the power of Generative Artificial Intelligence to automate and personalize travel planning. This project focuses on developing an intelligent travel assistant capable of generating customized travel itineraries based on user inputs such as destination, number of days, and nights. By integrating Generative AI models, the system can produce human-like, context-aware travel plans that include attractions, daily schedules, and food suggestions.

This project demonstrates how AI can be used not only as a technical tool but as a practical solution to real-world problems. TravelGuideAI enhances user convenience, improves decision-making, and reduces planning effort. It represents the future of smart travel planning where AI acts as a virtual travel advisor.

### **1.1 Project Overview**

TravelGuideAI is a Generative AI-based travel itinerary generator designed to help users create personalized travel plans quickly and efficiently. The application allows users to enter a destination and trip duration, and then automatically generates a structured itinerary with recommended places to visit, activities, and food suggestions.

The system is built using:

- Python for backend logic
- Streamlit for the user interface
- Google Gemini Generative AI API for AI-powered content generation
- Cloud-based AI services for scalable and intelligent responses

The application workflow is simple and user-friendly:

1. User enters travel details (destination, days, nights)
2. The system sends a structured prompt to the Generative AI model
3. The AI processes the request and generates a detailed itinerary
4. The itinerary is displayed in an organized format for the user

This project not only provides functional value but also serves as a learning platform for applying Generative AI in real-world scenarios. It demonstrates integration of APIs, prompt engineering, user interface design, and agile development practices.

TravelGuideAI is scalable and can be expanded in the future to include features such as hotel recommendations, budget planning, maps integration, and real-time weather insights.

## 1.2 Purpose of the Project

The primary purpose of the TravelGuideAI project is to simplify and enhance the travel planning experience using Generative AI. Many travelers struggle with planning due to lack of time, information overload, and difficulty in organizing schedules. This project aims to solve these issues by providing instant, personalized, and intelligent travel recommendations.

Key purposes include:

### 1 Simplify Travel Planning

Enable users to generate travel itineraries within seconds instead of hours of manual research.

### 2 Provide Personalization

Offer customized plans based on user inputs, making each itinerary unique and relevant.

### 3 Demonstrate Generative AI Capabilities

Showcase how modern AI models can generate meaningful, human-like content for practical applications.

### 4 Improve User Experience

Deliver a smooth and intuitive interface that requires minimal effort from users.

### 5 Academic & Skill Development

Serve as a hands-on project for learning:

- Generative AI integration
- API usage
- Prompt engineering
- Agile project management
- Software documentation and testing

### 6 Real-World Impact

Help travelers make better decisions, save time, and explore destinations more effectively

## 2. IDEATION PHASE

### 2.1 Problem Statement

Travel planning is an essential part of any trip, yet it remains a time-consuming and often stressful process for many people. Travelers typically spend hours researching destinations, browsing blogs, watching videos, comparing travel guides, and organizing daily schedules. This manual planning process can lead to information overload, confusion, and poorly structured itineraries.

Many existing travel platforms provide generic recommendations that do not fully match individual preferences, trip durations, or user needs. As a result, travelers may miss important attractions, waste time on inefficient schedules, or struggle to balance activities, food, and relaxation during their trip.

Additionally, not all travelers have prior knowledge about destinations, local culture, or must-visit places. First-time travelers, in particular, face difficulty in deciding what to prioritize within limited travel days. This creates a gap between available travel information and personalized travel planning.

With the rapid advancement of Artificial Intelligence, especially Generative AI, there is an opportunity to automate and personalize travel planning. However, many users still lack access to simple AI-powered tools that can instantly generate structured and customized itineraries.

Therefore, the core problem this project addresses is:

How can we simplify and personalize travel planning using Generative AI so that users can quickly generate reliable, customized itineraries without spending excessive time on research? TravelGuideAI aims to solve this problem by providing an AI-powered travel itinerary generator that produces personalized, structured, and easy-to-follow travel plans based on user inputs such as destination and trip duration.

### 2.2 Empathize Map

The empathy map for TravelGuideAI was created to understand travelers' behaviors, needs, and emotions during trip planning. The team analyzed how users search for travel information, what they expect from a planning tool, and what challenges they commonly face.

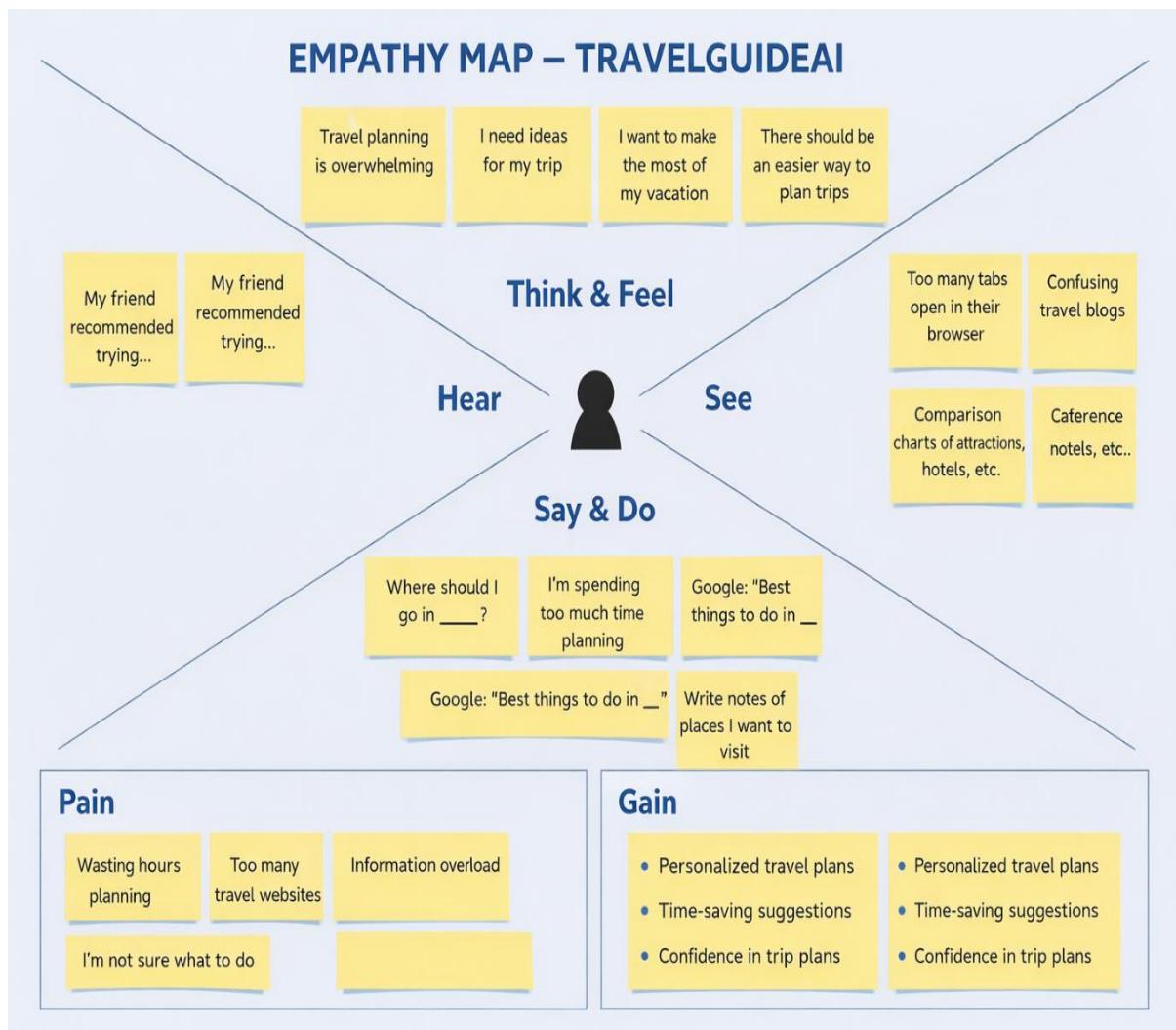
Travelers often want a smooth and enjoyable trip, but planning requires time, research, and decision-making. By mapping what users think, feel, say, and do, the team gained insights into their real needs and frustrations.

This empathy exercise helped the team view the problem from the traveler's perspective. It guided the design of TravelGuideAI to ensure the solution is user-friendly, time-saving, and personalized.

Understanding the user deeply allows TravelGuideAI to deliver meaningful and practical travel itineraries.

Reference: <https://www.mural.co/templates/empathy-map-canvas>

## Empathy Map – TravelGuideAI



## 2.3 Brainstorm & Idea Prioritization Template

Brainstorming was conducted by the team to identify innovative project ideas suitable for a Generative AI internship. The objective was to explore real-world problems where AI could provide practical and scalable solutions. All team members actively contributed ideas related to AI, automation, and user convenience.

During the session, multiple ideas were discussed, evaluated, and expanded collaboratively. The team focused on selecting a project that demonstrates Generative AI capabilities, solves a real problem, and is feasible within the internship timeline.

Reference: <https://www.mural.co/templates/brainstorm-and-idea-prioritization>

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

## Step-2: Brainstorm, Idea Listing and Grouping

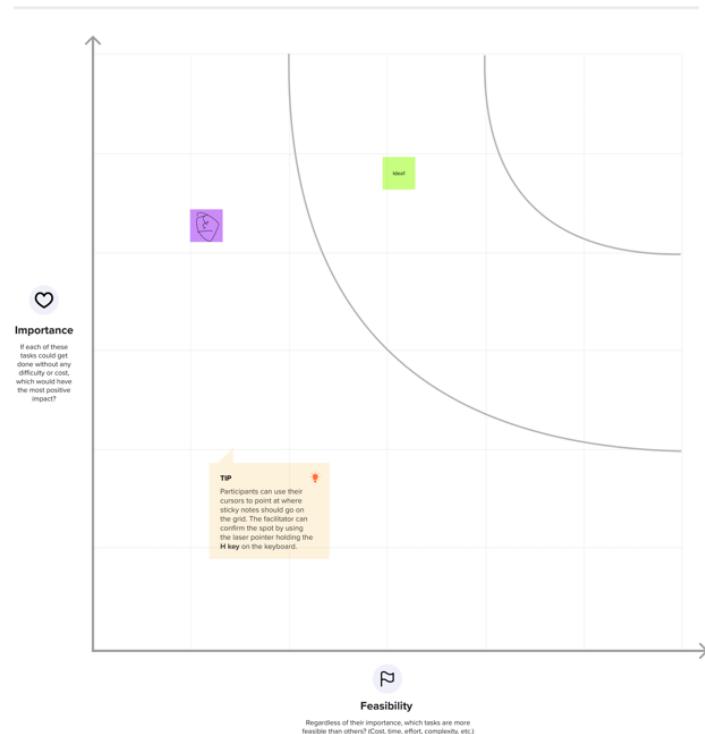
## Step-3: Idea Prioritization

4

### Prioritize

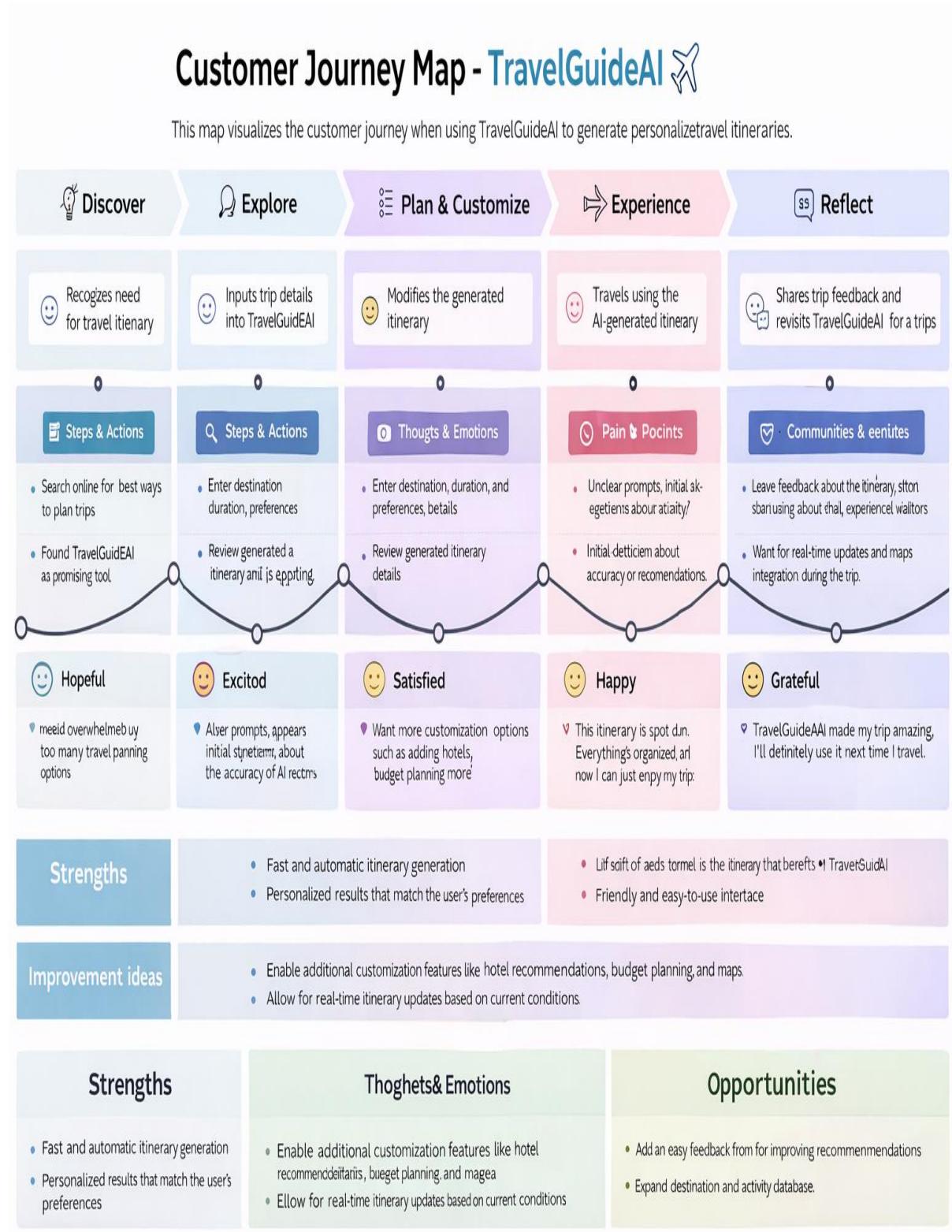
Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

⌚ 20 minutes



### 3. REQUIREMENT ANALYSIS

#### 3.1 Customer Journey map



### 3.2 Solution Requirements (Functional & Non-functional)

#### Functional Requirements

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIn
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3		
FR-4		

#### Non-Functional Requirements

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	
NFR-2	Security	
NFR-3	Reliability	
NFR-4	Performance	
NFR-5	Availability	
NFR-6	Scalability	

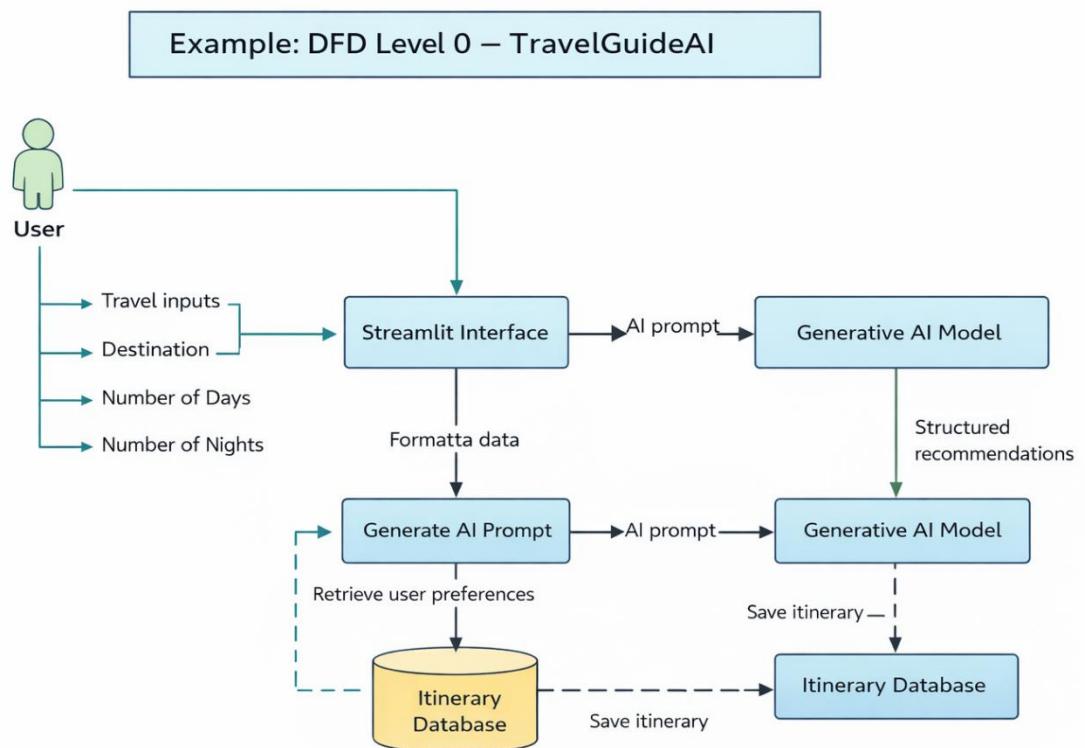
### 3.3 Data Flow Diagram & User Stories

The Data Flow Diagram (DFD) for TravelGuideAI illustrates how user inputs are processed to generate a personalized travel itinerary. The system is designed to capture user requirements, process them through a Generative AI model, and return structured travel recommendations.

In TravelGuideAI, the user provides inputs such as destination, number of days, and number of nights through the Streamlit interface. This data is sent to the backend system where it is formatted into a prompt for the Generative AI model.

The AI model processes the request and generates a detailed itinerary. The output is then sent back to the Streamlit application and displayed to the user.

The DFD helps visualize how information flows between the user, the application interface, and the AI model, ensuring a smooth and efficient itinerary generation process.



## User Stories – TravelGuideAI

User Type	Functional Requirement	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account/ dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can register & access with Gmail Login	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can access my account/ dashboard	High	Sprint-1
Customer (Web user)	Dashboard					Sprint-1
Customer Care Executive						
Administrator						

### 3.4 Technology Stack (Architecture & Stack)

#### Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

#### Technical Architecture – TravelGuideAI

The technical architecture of TravelGuideAI is designed to provide a smooth and intelligent travel itinerary generation experience. The system integrates a user-friendly frontend with a powerful Generative AI backend.

The architecture consists of three main layers:

## 1 Frontend Layer (User Interface)

The frontend is developed using **Streamlit**, which allows users to enter travel details such as destination, number of days, and number of nights. The interface is simple, interactive, and accessible through a web browser.

## 2 Application Logic Layer

This layer processes user inputs and formats them into structured prompts. It handles validation, request formatting, and communication between the frontend and the AI model.

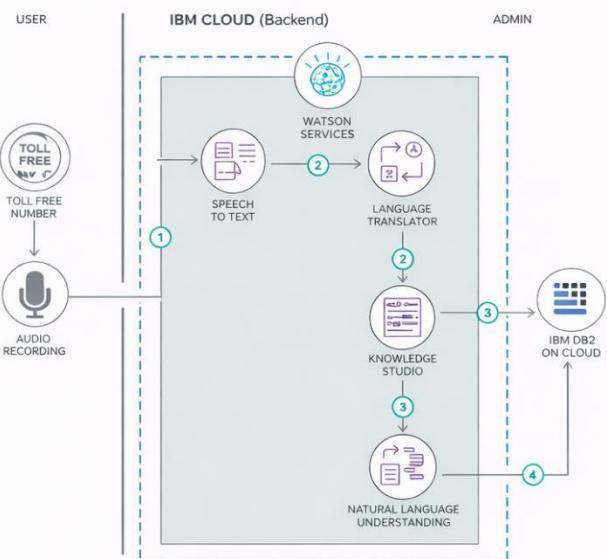
## 3 AI Processing Layer

The Generative AI model (Gemini) analyzes the user's inputs and generates a personalized travel itinerary. The model creates structured outputs including daily plans, attractions, and dining suggestions.

The generated itinerary is then sent back to the Streamlit interface for display to the user.

**Example:** Order processing during pandemics for offline mode

Reference: <https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/during-pandemics/>



### Guidelines:

- Include all the processes (As an application logic / Technology Block)
- Provide infrastructural demarcation (Local / Cloud)
- Indicate external interfaces (third party API's etc.)
- Indicate Data Storage components / services
- Indicate interface to machine learning models (if applicable)

**Table-1 : Components & Technologies:**

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript / Angular JS / React Js etc.
2.	Application Logic-1	Logic for a process in the application	
3.	Application Logic-2	Logic for a process in the application	Java / Python
4.	Application Logic-3	Logic for a process in the application	Generative AI Model (Gemini)
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
5.	Cloud Database	Database Service on Cloud	Cloud Firestore / Cloud MySQL etc.
6.	Cloud Storage	Cloud file storage requirements	Google Cloud Storage
7.	External API	Purpose of External API used in the application	Travel API, Weather API etc.
8.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud	Google Cloud / Kubernetes etc.
9.			Google Cloud / Kubernetes etc.

**Table-2: Application Characteristics:**

### Characteristics for Cloud or Local Architecture

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Streamlit, Jenkins, Docker, Kubernetes, TensorFlow etc.
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	SHA-256, Encryptions, IAM Controls, OWASP etc.
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Containerization, Load Balancers, Microservices etc.
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	Google Kubernetes Engine, Cloud Load Balancer etc.
5.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	Google Kubernetes Engine, Cloud Load Balancer etc.
5.	Performance	Design consideration for the performance of the application (number of requests per s, use of Cache, use of CDN's) etc.	Caching, Content Delivery Networks (CDN's) etc.
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	Caching, Content Delivery Networks (CDN's) etc.

**References:**

<https://c4model.com/>

<https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/>

<https://www.ibm.com/cloud/architecture>

<https://aws.amazon.com/architecture>

<https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fd90d>

## 4. PROJECT DESIGN

### 4.1 Problem – Solution Fit Template

In the context of *TravelGuideAI* means identifying real travel planning difficulties faced by users and providing an AI-powered itinerary solution that genuinely solves those issues. It ensures that the system is not just innovative, but also practical, useful, and aligned with user needs.

*TravelGuideAI* focuses on simplifying trip planning by generating personalized travel itineraries based on user inputs like destination, number of days, and preferences. The solution fits the problem by reducing planning time, confusion, and information overload.

#### Purpose

- Solve real travel planning challenges

*TravelGuideAI* helps users who struggle with deciding where to go, what to do, and how to organize their trips efficiently.

- Improve solution adoption

By providing instant AI-generated itineraries, users can quickly see value, increasing usage and trust in the platform.

- Enhance communication & engagement

Clear recommendations, structured daily plans, and user-friendly design make the experience simple and engaging.

- Increase user touchpoints

Frequent travelers can reuse the platform for multiple trips, creating continuous engagement and loyalty.

- Understand and improve user experience

By analyzing user inputs and preferences, *TravelGuideAI* continuously provides more relevant and personalized suggestions.

## Template:

<b>Define CS, CC, AS</b>	<p><b>1. CUSTOMER SEGMENT(S)</b> Who are your customers? i.e. working parents of 6-8 yo kids</p>	<p><b>CS</b></p> <p><b>6. CUSTOMER CONSTRAINTS</b> What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices</p>	<p><b>CC</b></p> <p><b>5. AVAILABLE SOLUTIONS</b> Which solutions are available to the customers when they face the problem as defined in point 2? What would they say if this problem is a pain &amp; do ho-sae! solutions feel luxurious and are in alternative to sight reducing trannics.</p>	<b>Explore AS, differentiatie</b>
<b>Focus on R, Hg into DE, undervarianten DC</b>	<p><b>2. JOBS-TO-BE-DONE / PROBLEMS</b> What jobs need to be done/pain/dolem, will for your customers? There could be more than one, encourage different ideas.</p>	<p><b>J&amp;P</b></p> <p><b>9. PROBLEM ROOT CAUSE</b> What is the real reason that this problem exists? Are customers have kids of 6-8yo mang in the woc job? i.e. customers have kids &amp; no more sports time due to le regulations.</p>	<p><b>RC</b></p> <p><b>7. BEHAVIOUR</b> What do they do now in extandje to address the problem and get the job done? i.e. indirectly sourced second hand sports time in avolueering work &amp; Greenspace.</p>	<b>BE</b>
<b>Identify a po enge TR BN</b>	<p><b>3. TRIGGERS</b> What triggers customers to act? i.e. seeing their neighbour installing smart signers, reading about a more efficient solution.</p>	<p><b>TR</b></p> <p><b>10. YOUR SOLUTION</b> As a new business of an extelling business, witten down your current soluton fist, what then óne the surmat contritoation, tot thunfer stiglacion. Show also as coices in miergs to extante of then to Geton your soluton. #2. focbenness of them erentie streeet to dophos in reute in netter firbrons. Eaten to batone! and constraints costined the customers antilization eats.</p>	<p><b>SL</b></p> <p><b>8. CHANNELS of BEHAVIOUR</b></p>	<b>CH</b>
	<p><b>3. TRIGGERS</b> What triggers customers to act? i.e. seeing their neighbour installing smart signers, reading about a more efficient solution.</p>	<p><b>EM</b></p>	<p><b>6.1. ONLINE</b> What kind of actions do customers take online? Extract online channels from it and use then for customer aharaparent.</p> <p><b>6.3. OFFLINE</b> What kind of actions do customers take offline? Extract offline channels from it and use then for customer aharaparent.</p>	<b>Extract online daaren of RE</b>

## References:

1. <https://www.ideahackers.network/problem-solution-fit-canvas/>
2. <https://medium.com/@epicantus/problem-solution-fit-canvas-aa3dd59cb4fe>

## 4.2 Proposed Solution Template

### Proposed Solution Template: TravelGuideAI

Project team shall fill the following information for *TravelGuideAI* in the provided template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Users face difficulties in efficiently planning personalized travel itineraries.
2.	Idea / Solution description	An AI-powered platform that generates customized travel itineraries based on user inputs like destination, days, and preferences.
3.	Novelty / Uniqueness	Instant, personalized itineraries generated by generative AI (Gemini).
4.	Social Impact / Customer Satisfaction	Improves user experience by reducing planning effort and offering tailored recommendations.
5.	Business Model (Revenue Model)	Freemium: Basic services free, premium features available through subscription.
6.	Scalability of the Solution	Built to handle increasing user demand and adaptable for expanded services.

## 4.3 Solution Architecture

### Solution Architecture:

Solution architecture for *TravelGuideAI* defines how the system converts user travel needs into intelligent, AI-generated itineraries. It connects user requirements, application logic, and generative AI services into a smooth, scalable system.

#### Goals of the TravelGuideAI Solution Architecture

- Deliver an AI-powered travel planning solution that generates personalized itineraries based on destination, days, and nights.
- Clearly define system structure and components (UI, backend, AI model, and APIs) so stakeholders understand how the system works.
- Outline features and development phases such as itinerary generation, user input handling, and result presentation.
- Specify technical requirements for deployment, API integration, and data handling.
- Ensure scalability and reliability so multiple users can generate itineraries simultaneously.

## Solution Architecture Diagram:

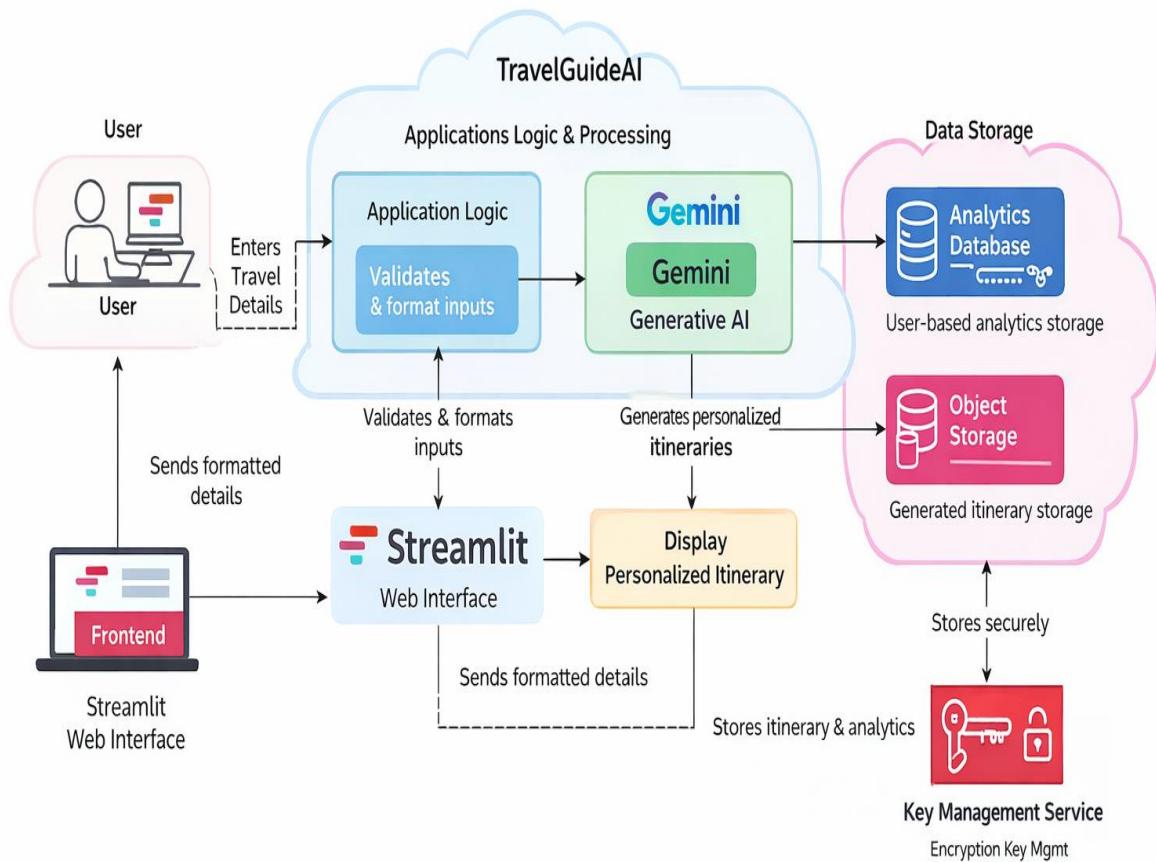


Figure : Architecture and data flow of the TravelGuideAI application

Reference: <https://aws.amazon.com/blogs/industries/voice-applications-in-clinical-research-powered-by-ai-on-aws-part-1-architecture-and-design-considerations/>

## 5. PROJECT PLANNING & SCHEDULING

### 5.1 Project Planning

**Product Backlog, Sprint Schedule, and Estimation (TravelGuideAI)**

Sprint	Functional Requirement (Epic)	Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	
Sprint-2		USN-3	As a user, I can register for the application through Facebook	2	Low	
Sprint-1	Dashboard	USN-4	As a user, I can register for the application through Gmail	2	Medium	
Sprint-1		USN-5	As a user, I can log into the application by entering email & password	1	High	
Sprint-1	Login	USN-5		1	High	

**Project Tracker, Velocity & Burndown Chart (TravelGuideAI)**

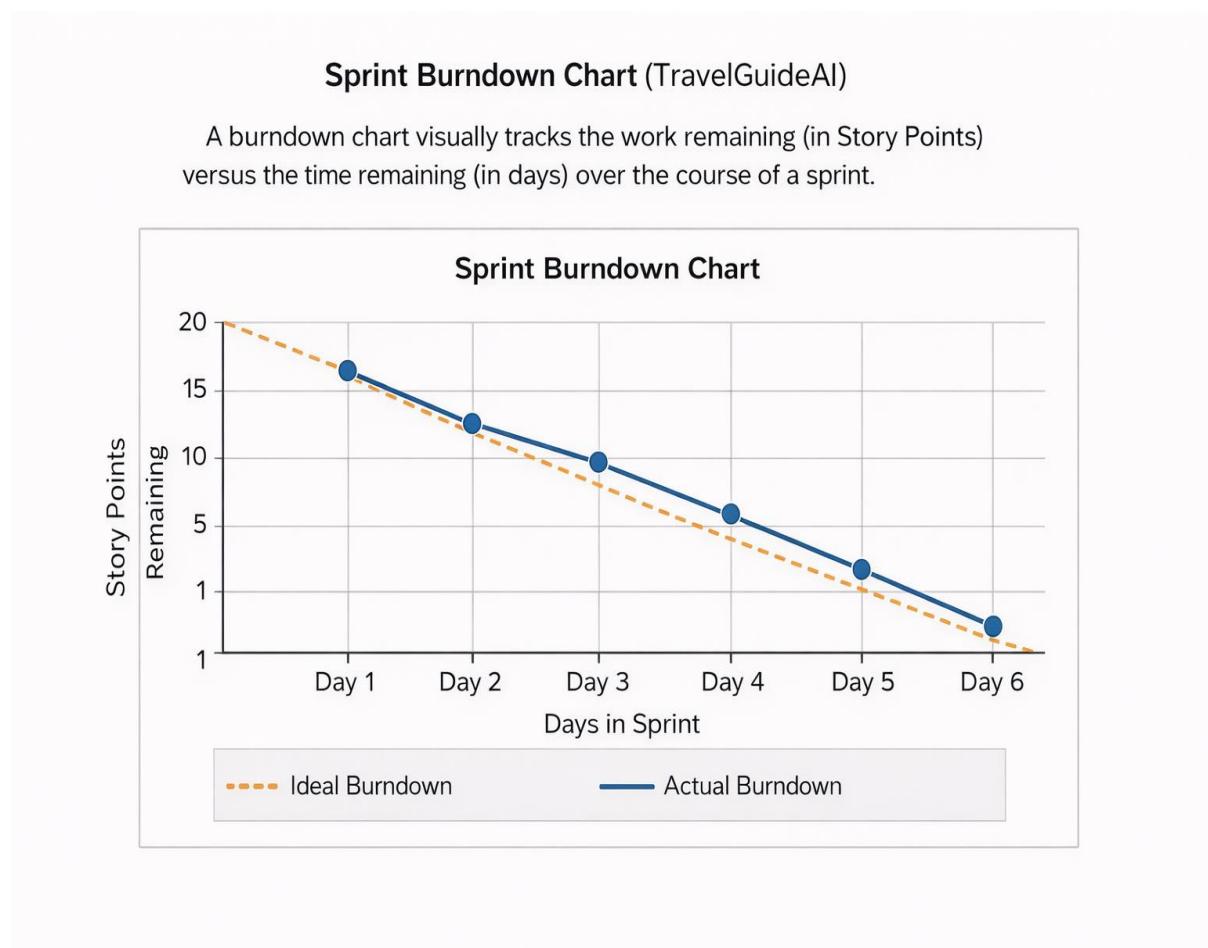
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

#### Velocity:

Each sprint has a 6-day duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day).

$$AV = \frac{\text{sprint duration}}{\text{velocity}} = \frac{20}{6} \approx 3,33$$

## Burndown Chart:



<https://www.visual-paradigm.com/scrum/scrum-burndown-chart/>

<https://www.atlassian.com/agile/tutorials/burndown-charts>

## Reference:

<https://www.atlassian.com/agile/project-management>

<https://www.atlassian.com/agile/tutorials/how-to-do-scrum-with-jira-software>

<https://www.atlassian.com/agile/tutorials/epics>

<https://www.atlassian.com/agile/tutorials/sprints>

<https://www.atlassian.com/agile/project-management/estimation>

<https://www.atlassian.com/agile/tutorials/burndown-charts>

## 6. FUNCTIONAL AND PERFORMANCE TESTING

### 6.1 Functional & Performance Testing Template

#### Test Scenarios & Results

Test Case ID	Scenario (What to test)	Test Steps (How to test)	Expected Result	Actual Result	Result	Pass/Fail
FT-01	Text Input Validation (e.g., topic, job title)	Enter valid and invalid text in input fields	Valid inputs accepted, errors for invalid inputs	✓ Valid inputs	✓ Valid inputs	✓ Pass
FT-02	Number Input Validation (e.g., word count, size, rooms)	Enter numbers within and outside the valid range	Accepts valid values, shows error for out-of-range	✓ Valid inputs	✓ Valid inputs	✓ Pass
FT-03	Content Generation (e.g., blog, resume, design idea)	Provide complete inputs and click "Generate"	Correct content is generated based on input	Correct content	Correct content	✓ Pass
FT-04	API Connection Check	Check if API key is correct and model responds	API responds successfully	API responds	API responds	✓ Pass
PT-01	Response Time Test	Use a timer to check content generation time	Should be under 3 seconds	Under 3 seconds	Under 3 seconds	✓ Pass
PT-01	API Speed Test	Send multiple API calls at the same time	API should not slow down	API did not slow down		✓ Pass
PT-02	API Speed Test	Upload multiple API calls at the same time	API did not slow down	Worked smoothly without crashing		✓ Pass
PT-03	File Upload Load Test (e.g., PDFs)	Upload multiple PDFs and check processing	Worked smoothly without crashing			✓ Pass

# Acceptance Testing

## UAT Execution & Report Submission

### 1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing (UAT).

### 2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal	Total
By Design	10	4	2	3	20	20
Duplicate	1	0	3	0	4	4
External	2	3	0	1	6	6
Fixed	11	2	4	20	37	37
Not Reproduced	0	0	1	0	1	1
Skipped	0	0	1	1	2	2
Won't Fix	0	5	2	1	8	8
Totals	24	14	13	26	77	77

### 3. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

## 7. RESULTS

Output Screenshot:

# Travel Itinerary Generator ✈️

Enter Destination

varanashi

Number of Days

5

Press Enter to apply

- +

Number of Nights

0

I

- +

Generate Itinerary

Generate Itinerary

Your Travel Plan:

---

### Day 1: Arrival, Assi Ghat & Evening Aarti Spectacle

\*\*Morning (Arrival & Check-in):\*\*

- \* \*\*Arrival:\*\* Arrive at Lal Bahadur Shastri International Airport (VNS), Varanasi Junction Railway Station (BSB), or Manduadih Railway Station (MUV).
- \* \*\*Transfer:\*\* Take a pre-booked taxi or auto-rickshaw to your hotel. Depending on your hotel's location, you might have to walk the last stretch from the main road if it's in the labyrinthine lanes near the ghats.
- \* \*\*Check-in & Refresh:\*\* Settle into your accommodation.
- \* \*\*Lunch:\*\* Have a light lunch at your hotel or a nearby café.

\*\*Afternoon (Assi Ghat Serenity):\*\*

- \* \*\*1:00 PM - 3:00 PM: Explore Assi Ghat:\*\* Start your Varanasi journey at Assi Ghat, the southernmost and one of the most vibrant ghats. It's relatively less crowded than Dashashwamedh

[Generate Itinerary](#)

Your Travel Plan:

- proclaimed guides.
- \* **Photography:** Be respectful. Ask permission before photographing people. Avoid taking pictures at cremation ghats.
- \* **Bargaining:** It's common practice in markets.
- \* **Beware of Scams:** Be cautious of individuals offering unsolicited services, especially near major temples and ghats.
- \* **Stay Hydrated:** Carry a water bottle, especially if traveling outside winter months.
- \* **Food & Water:** Stick to bottled water. Eat at reputable places or where you see a good turnover of locals.
- \* **Flexibility:** Varanasi can be overwhelming. Don't try to cram too much in. Allow yourself time to just sit and observe. The charm of Varanasi is often in its unplanned moments.
- \* **Connectivity:** Wi-Fi is available in most hotels and some cafes, but be prepared for occasional power cuts or network fluctuations.

This itinerary provides a solid framework, but feel free to adjust it based on your interests and pace. Enjoy your spiritual journey through the mystical lanes of Varanasi!

## Travel Itinerary Generator

Enter Destination

hyderabad

Number of Days

3

- +

Number of Nights

4

- +

[Generate Itinerary](#)



 Generating itinerary...

## **8. ADVANTAGES & DISADVANTAGES**

### **Advantages**

#### **1. Personalized Trip Planning**

TravelGuideAI generates customized itineraries based on user preferences such as destination, duration, budget, and interests. This ensures each travel plan is unique and relevant.

#### **2. Time-Saving**

Instead of spending hours researching destinations, hotels, and activities, users receive a complete plan in seconds, significantly reducing planning time.

#### **3. User-Friendly Interface**

The Streamlit-based interface is simple and intuitive, allowing even non-technical users to generate travel plans easily.

#### **4. AI-Powered Recommendations**

Using Generative AI (Gemini), the system provides smart suggestions for attractions, food, and daily schedules that feel human-like and creative.

#### **5. Flexible & Editable Plans**

Users can regenerate or modify itineraries, making the system adaptable to changing needs.

#### **6. Scalable Solution**

The system can be expanded to include hotel booking, maps, weather integration, and real-time travel updates.

#### **7. 24/7 Availability**

Unlike human travel agents, the AI system is available anytime, providing instant assistance.

### **Disadvantages**

#### **1. Dependency on API & Internet**

The system requires a stable internet connection and API access. If the API quota is exceeded or the service is down, functionality is affected.

#### **2. Limited Real-Time Data**

The itinerary may not always reflect real-time changes like weather, traffic, or sudden closures.

### **3. Possible AI Inaccuracies**

AI-generated suggestions might sometimes be generic or slightly inaccurate compared to local expert advice.

### **4. No Direct Booking Integration**

Currently, users cannot directly book hotels, flights, or tickets through the system.

### **5. Cost at Scale**

Heavy usage may incur API costs when scaling beyond free tiers.

### **6. Less Human Touch**

Some travelers prefer human travel agents for emotional assurance and deep local insights.

## **9.CONCLUSION**

TravelGuideAI demonstrates how Generative AI can significantly transform the way people plan their travel. By combining user inputs such as destination, number of days, and nights with the power of large language models, the system is able to generate structured, personalized, and meaningful travel itineraries within seconds. This project highlights the practical application of AI in solving real-world problems and improving user convenience.

Through this project, we successfully designed and developed an AI-powered itinerary generator using Streamlit for the frontend and Google's Gemini model for intelligent content generation. The system simplifies travel planning by reducing manual research, offering creative suggestions, and presenting organized day-wise plans. It proves that AI can act as a smart assistant in decision-making and planning tasks.

The development process also provided valuable learning in areas such as API integration, prompt design, user experience design, and AI-based application deployment. Challenges like API limits, model selection, and response handling helped us understand real-world constraints in AI solutions.

Overall, TravelGuideAI serves as a strong foundation for future expansion. With additional features such as real-time data integration, booking systems, maps, and budget optimization, the system can evolve into a complete travel companion. This project not only meets its objective of generating travel itineraries but also showcases the potential of Generative AI in building intelligent, user-centric applications.

In conclusion, TravelGuideAI is a successful implementation of AI-driven travel assistance, demonstrating innovation, practicality, and scalability, while opening opportunities for further enhancement and real-world adoption.

## **10.FUTURE SCOPE**

TravelGuideAI has strong potential for enhancement and real-world scalability. While the current version focuses on AI-generated travel itineraries, several advanced features can be integrated to make it a complete smart travel assistant. The future scope of this project includes technical improvements, feature expansion, and real-time service integration.

### **1. Real-Time Data Integration**

Future versions can connect to live APIs for:

- Weather forecasts
- Flight and hotel prices
- Local events and festivals
- Traffic and travel time updates

This would allow the itinerary to be dynamic and context-aware.

### **2. Booking & Reservation Integration**

The system can be extended to support:

- Hotel booking
- Flight/train reservations
- Cab and local transport booking
- Activity and tour reservations

Users could plan and book their entire trip within one platform.

### **3. Personalized Recommendation System**

By using user history and preferences:

- AI can learn favorite destinations, food choices, and budgets
- Provide smarter, repeat-user personalization
- Suggest hidden gems and non-touristy experiences

This would improve user satisfaction and retention.

#### **4. Budget Optimization Feature**

Future upgrades can include:

- Budget-based itinerary generation
- Cost breakdown per day
- Affordable alternatives for activities and stays

This makes the tool more practical for students and budget travelers.

#### **5. Multi-Language Support**

Adding multilingual capabilities would:

- Help international users
- Increase accessibility
- Expand global usability

#### **6. Mobile Application Development**

Developing Android and iOS apps would:

- Improve accessibility
- Provide on-the-go planning
- Enable offline itinerary access

#### **7. Map & Navigation Integration**

Integration with map services can:

- Show routes between attractions
- Provide navigation support
- Optimize daily travel paths

## **8. Voice Assistant Integration**

Users could plan trips using voice commands via:

- Voice chatbots
- Smart assistants
- Hands-free interaction

## **9. AI Chat Travel Assistant**

A conversational AI could:

- Answer travel questions
- Suggest places in real-time
- Modify plans instantly

## **10. Social & Community Features**

Future versions may allow:

- User reviews and ratings
- Shared itineraries
- Community travel tips

## 11. APPENDIX

### Source Code

```
import streamlit as st

import google.generativeai as genai

# Configure API key
genai.configure(api_key="*****")

# Function to generate itinerary

def generate_itinerary(destination, days, nights):

    model = genai.GenerativeModel("gemini-2.5-flash")

    prompt = f"""

        Create a detailed travel itinerary for {destination}

        for {days} days and {nights} nights.

        Include daily activities, attractions, and food suggestions.

    """

    response = model.generate_content(prompt)

    return response.text

# Streamlit UI

st.title("Travel Itinerary Generator ✈️")

destination = st.text_input("Enter Destination")

days = st.number_input("Number of Days", min_value=1)

nights = st.number_input("Number of Nights", min_value=0)

if st.button("Generate Itinerary"):

    if destination.strip() == "":

        st.error("Please enter a destination")

    else:

        with st.spinner("Generating itinerary..."):

            result = generate_itinerary(destination, days, nights)

            st.text_area("Your Travel Plan:", result, height=400)
```

Github: <https://github.com/SaiBhaskar2004/TravelGuide/tree/main>

Project Demo :

<https://drive.google.com/file/d/1Qxd9dKDBFqzemDAz44VGQA0fbkEsbjiX/view?usp=sharing>