

Model Development Phase

Date	15 February 2026
Team ID	LTVIP2026TMIDS65532
Project Title	Explore With Ai: Custom Itineraries For Your Next Journey
Maximum Marks	10 Marks

Initial Model Training Code, Model Validation and Evaluation Report

This project uses a **pre-trained generative AI model** and does not involve custom model training. Therefore, this phase focuses on model initialization, prompt-based interaction, and output evaluation instead of traditional training and validation.

Model Initialization and Integration (5 marks):

The generative AI model used in this project is a **pre-trained Gemini Flash model** provided through the Google Generative AI API. The model is initialized using a secure API key configuration and integrated into the application backend.

The model receives structured textual prompts containing user inputs such as destination, number of days, and number of nights. Based on this input, the model generates a detailed and personalized travel itinerary.

```
def generate_itinerary(destination, days, nights):
    #Create model configuration
    generation_config = {
        "temperature": 0.7,
        "top_p": 0.9,
        "top_k": 50,
        "max_output_tokens": 2048,
        "response_mime_type": "text/plain"
    }

    response = client.models.generate_content(
        model="gemini-2.5-flash",
        contents=f"""
            Create a detailed travel itinerary for {destination}
            for {days} days and {nights} nights.
            Include daily activities, nearby attractions,
            food recommendations, and travel tips.
        """
    )

    return response.text
```

Model Validation and Evaluation Report (5 marks):

Since the model is pre-trained, validation is performed through functional testing and output quality assessment rather than numerical performance metrics.

Evaluation Criteria Used:

- Relevance of the generated itinerary to the selected destination
- Logical distribution of activities across days
- Completeness of itinerary (activities, attractions, food suggestions)
- Response consistency for different inputs

Multiple test inputs were provided to the system, and the generated outputs were reviewed to ensure accuracy, coherence, and usefulness.

Travel Itinerary Generator

Enter your desired destination:

Enter the number of days:
 - +

Enter the number of nights:
 - +

Generate Itinerary:

Generated Itinerary:

Dwarka (also spelled Dwaraka) is one of the seven holiest cities in Hinduism, a revered pilgrimage site, and believed to be the ancient kingdom of Lord Krishna. This 5-day, 5-night itinerary is designed to immerse you in its spiritual aura, explore its historical significance, and enjoy the coastal charm of Gujarat.

--

Dwaraka: 5 Days & 5 Nights Itinerary

Best Time to Visit: October to March (pleasant weather). Avoid monsoon (July-September) due to heavy rains and summer (April-June) due to intense heat.

Getting There:

- * **By Air:** The nearest airports are Jamnagar (approx. 130 km), Porbandar (approx. 110 km), and Rajkot (approx. 225 km). From there, you can hire a taxi or take a bus.
- * **By Train:** Dwarka has its own railway station, well-connected to major cities like Ahmedabad, Mumbai, Delhi, etc.
- * **By Road:** Dwarka is well-connected by state and national highways. Regular bus services operate from various cities in Gujarat.

Accommodation: Dwarka offers a range of options from budget hotels and guesthouses to mid-range hotels and dharamshalas (pilgrim rest houses). It's advisable to book in advance, especially during peak season or festivals.

--

Day 1: Arrival & First Glimpse of Dwarkadish

- * **Morning/Afternoon:** Arrive in Dwarka. Check into your hotel.
- * **Lunch:** Enjoy a traditional Gujarati Thali at a local restaurant.
- * **Late Afternoon (4:00 PM - 6:00 PM):**
- * **Gomti Ghat:** Head to Gomti Ghat, the sacred confluence of the River Gomti with the Arabian Sea. Take a holy dip (optional) in the sacred waters. Spend some time watching the ritualistic

Model	Summary	Evaluation Method
Gemini Flash Lite	Large Language Model used via API	Output quality validation through multiple test cases