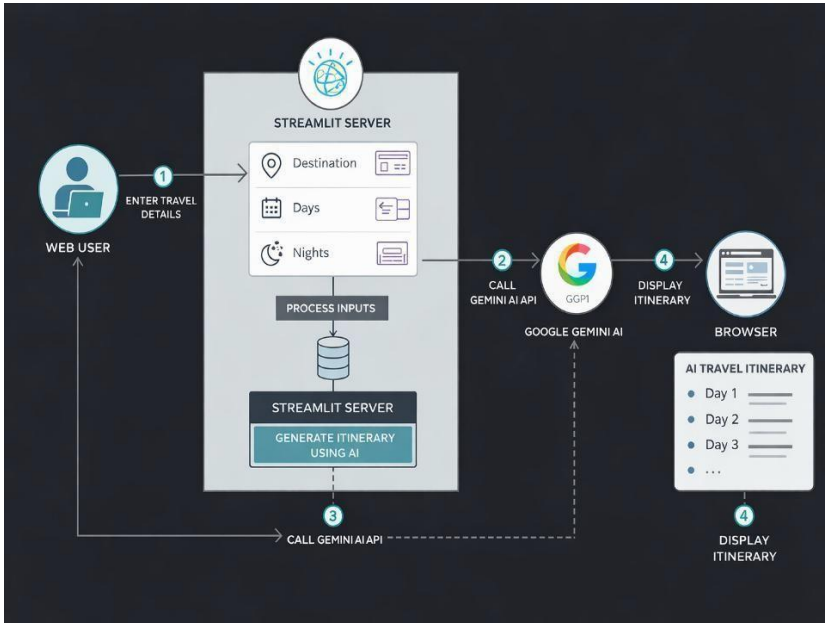
Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	31 January 3035
Team ID	LTVIP2026TMIDS79632
Project Name	Explore with AI: Custom Itineraries for Your Next Journey
Maximum Marks	4 Marks

Technical Architecture:

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

2 **Example: Explore with AI: Custom Itineraries for Your Next Journey**



Guidelines:

- Show the user accessing the Streamlit web application through a browser.
- Include input fields for destination, number of days, and number of nights.
- Show Streamlit application sending the input to the AI model (Gemini API).
- Show the AI model processing the request and generating the itinerary.
- Show the generated itinerary sent back to the Streamlit application.
- Show the itinerary displayed to the user in the browser interface.
- Use arrows to clearly represent the data flow between user, Streamlit app, AI model, and output display.

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Interface where user enters destination, days, and nights	Streamlit
2.	Streamlit Application Server	Runs the web application and handles user interaction	Streamlit (Python)
3.	Input Processing Module	Processes user input and prepares data for AI	Python

4.	Gemini AI Integration Module	Connects application with Gemini AI API	Google Generative AI (Gemini API), Python
5.	Itinerary Generation Module	Generates travel itinerary using AI response.	Gemini Generative AI Model
6.	Output Display Module	Displays generated itinerary to the user	Streamlit
7.	Browser Client	Allows user to access the application	Web Browser (Chrome, Edge, etc.)
8.	Gemini AI API Service	External AI service used to generate itinerary	Google Gemini API
9.	Prompt Processing Logic	Creates prompt from destination, days, nights	Python
10.	Local Runtime Environment	Runs the application locally	Python Environment
11.	Internet Connectivity Module	Enables communication between app and Gemini API	Internet Connection

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology

1.	Open-Source Frameworks	The application is built using an open-source framework for creating the web interface and running the application	Streamlit, Python
2.	Security Implementations	Secure communication between application and Gemini AI using API key authentication and HTTPS	Gemini API Key, HTTPS
3.	Scalable Architecture	The application uses modular architecture with separate user interface, processing logic, and external AI service	Streamlit, Python, Gemini AI API
4.	Availability	The application is available when running on the local system with internet connectivity and accessible Gemini AI service	Streamlit Local Server, Internet Connection
5.	Performance	The application provides fast itinerary generation using efficient input processing and cloud-based AI response	Python, Streamlit, Gemini Generative AI