

## Project Design Phase-II

### Technology Stack (Architecture & Stack)

Date	31 January 3035
Team ID	LTVIP2026TMIDS77573
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

#### **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.

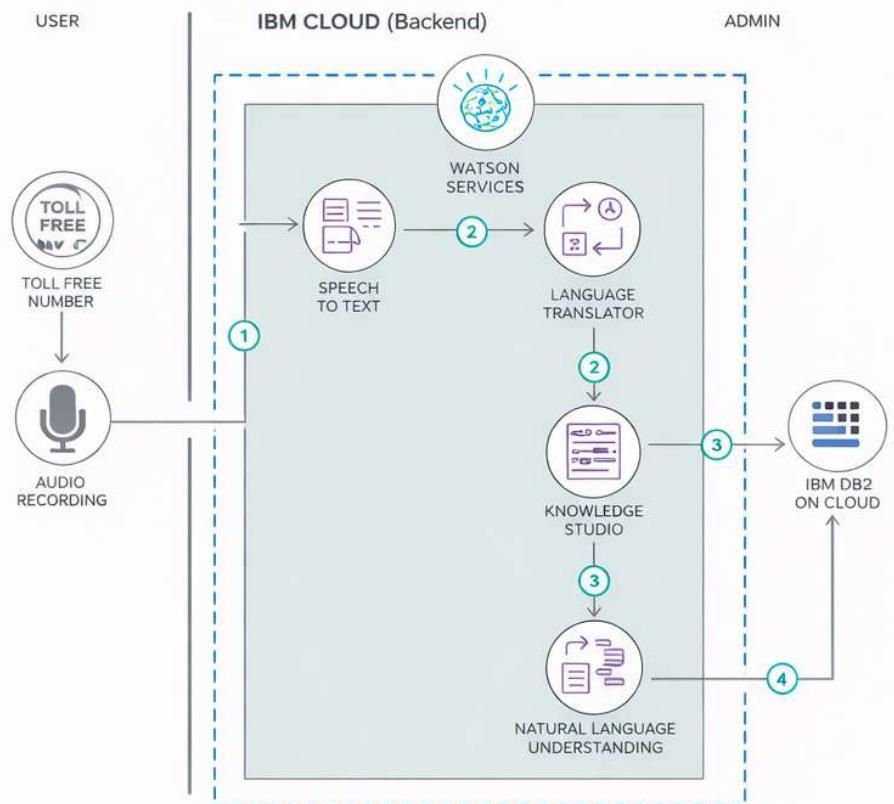
The generated itinerary is then sent back to the Streamlit interface for display to the user.

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

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

## 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-2d20c9fda90d>