

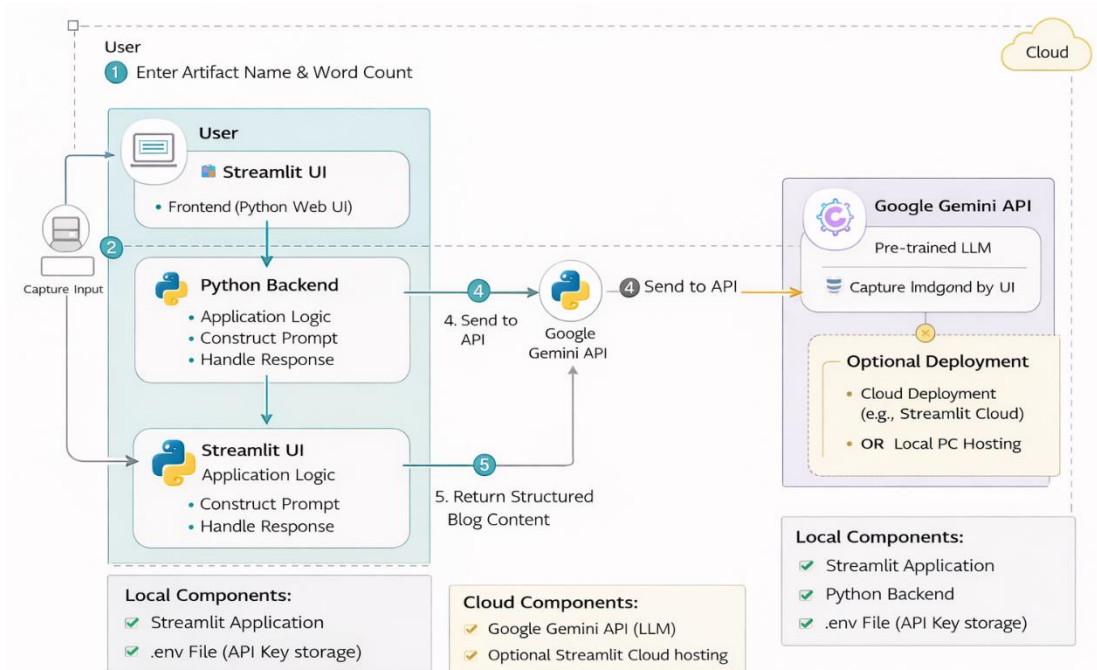
## Project Design Phase-II Technology Stack (Architecture & Stack)

Date	19 February 2026
Team ID	LTVIP2026TMIDS34956
Project Name	Gemini Historical Artifact Description App
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: Gemini Historical Artifact Description App



### 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	Allows user to input artifact name and word count	Streamlit (Python Web UI)
2	Application Logic-1	Prompt construction and response handling	Python
3	Application Logic-2	AI content generation processing	Google Gemini API
4	Machine Learning Model	Pre-trained LLM for text generation	Gemini-Pro Model
5	Database	Not mandatory (stateless app)	Not Used
6	Cloud Deployment	Hosting of application	Streamlit Cloud / Localhost
7	File Storage	Download generated blog as .txt file	Local File System
8	External API-1	AI text generation service	Google Generative AI API
9	Infrastructure	Local system or Cloud hosting	Local PC / Cloud Deployment

**Table-2: Application Characteristics:**

S.No	Characteristics	Description	Technology
1	Open-Source Frameworks	Web UI and backend framework	Streamlit, Python
2	Security Implementations	API key stored securely in .env file	Environment Variables
3	Scalable Architecture	Can scale by deploying on cloud	Streamlit Cloud
4	Availability	Available via web browser anytime	Web-based deployment
5	Performance	Lightweight app, fast API response	Gemini API Optimization

**References:**

<https://ai.google.dev/docs>

[https://ai.google.dev/tutorials/python\\_quickstart](https://ai.google.dev/tutorials/python_quickstart)

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

<https://owasp.org/www-project-top-ten/>

<https://docs.streamlit.io/>