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

Date	31 January 3035
Team ID	LTVIP2026TMIDS36980
Project Name	Gemini Pro Financial Decoder
Maximum Marks	4 Marks

Technical Architecture:

The Gemini Pro Financial Decoder follows a web-based, AI-driven architecture where users upload financial statements that are processed using application logic and analyzed using Google Gemini Generative AI. The system generates simplified summaries and visual insights to assist decision-making for non-finance users.

The architecture clearly separates:

- User Interface
- Application Logic
- External AI Services
- Data Processing & Visualization

Table-1: Components & Technologies

S.No	Component	Description	Technology
1	User Interface	Web interface where users upload financial statements and view summaries & charts	Streamlit (Python Web Framework), HTML, CSS
2	Application Logic-1	Handles file upload, validation, and preprocessing of CSV/XLSX files	Python
3	Application Logic-2	Processes financial data and prepares prompts for AI analysis	Python, Pandas

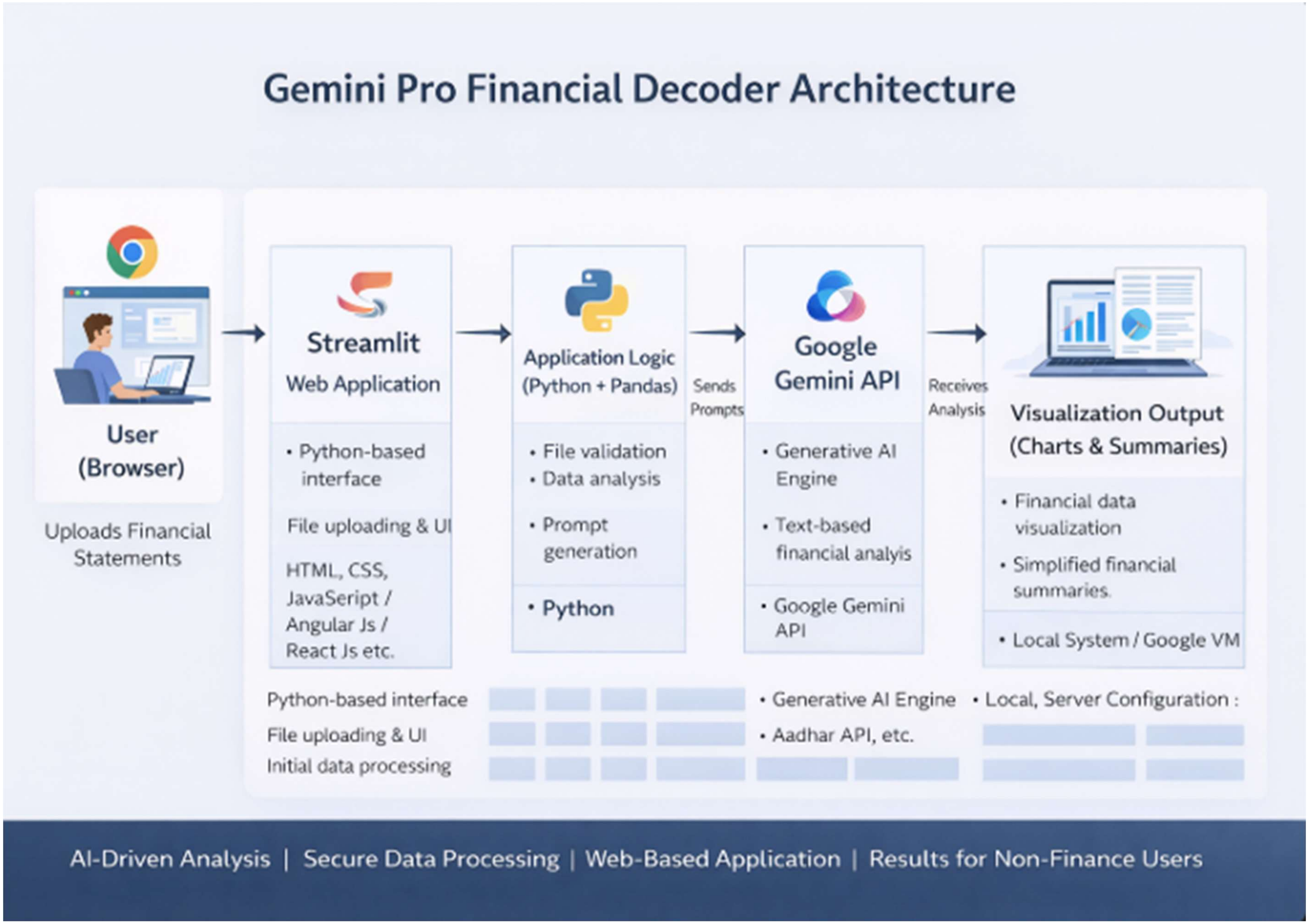
S.No	Component	Description	Technology
4	Application Logic-3	Controls workflow between UI, AI analysis, and visualization	Python
5	Database	Temporary in-memory data handling for uploaded financial files	Pandas DataFrames
6	Cloud Database	Not used; data processed locally for privacy	N/A
7	File Storage	Temporary storage of uploaded financial statements during runtime	Local File System
8	External API-1	Generates financial summaries using Generative AI	Google Gemini API
9	External API-2	Not applicable	N/A
10	Machine Learning Model	Natural Language Generation for financial summaries	Gemini Pro / Gemini Flash
11	Infrastructure (Server / Cloud)	Application runs locally or can be deployed on cloud	Local System / Google Cloud

---

Table-2: Application Characteristics

S.No	Characteristic	Description	Technology Used
1	Open-Source Frameworks	Uses open-source tools for UI and data processing	Python, Streamlit, Pandas
2	Security Implementations	API key-based access control and secure data handling	Google API Key, HTTPS
3	Scalable Architecture	Modular design allows easy scaling of AI and UI components	Python Modular Architecture
4	Availability	Application is available as long as local server or cloud instance is running	Localhost / Cloud VM
5	Performance	Efficient data handling with limited rows sent to AI; visualizations rendered locally	Pandas, Streamlit

▯



Gemini Pro Financial Decoder Architecture

User
(Browser)

Uploads Financial
Statements

Streamlit
Web Application

- Python-based interface

File uploading & UI

HTML, CSS,
JavaSeript /
Angular Js /
React Js etc.

Application Logic
(Python + Pandas)

- File validation
- Data analysis
- Prompt generation

- Python

Sends
Prompts

Google
Gemini API

- Generative AI Engine
- Text-based financial analyis

- Google Gemini API

Receives
Analysis

Visualization Output
(Charts & Summaries)

- Financial data visualization
- Simplified financial summaries.

- Local System / Google VM

Python-based interface

File uploading & UI

Initial data processing

- Generative AI Engine
- Aadhar API, etc.

- Local, Server Configuration :

AI-Driven Analysis | Secure Data Processing | Web-Based Application | Results for Non-Finance Users