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

Date	31 January 2026
Team ID	LTVIP2026TMIDS71395
Project Name	AutoSage App Using Gemini Flash
Maximum Marks	4 Marks

Technical Architecture:

The AutoSage application follows a simple 3-layer architecture:

1. User Interface Layer (Streamlit Web App)
2. Application Logic Layer (Python Backend + Gemini Flash API Integration)
3. Data Layer (Vehicle Data & System Storage)

The system allows users to upload vehicle images, which are validated and processed using the Gemini Flash model to generate structured vehicle details. The results are displayed through a web-based interface.

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1	User Interface	Web-based interface for users to upload vehicle images and view results	Streamlit (Python)
2	Application Logic-1	Image validation and processing logic	Python
3	Application Logic-2	AI-based vehicle image analysis	Gemini Flash API
4	Application Logic-3	Structured response generation (brand, mileage, price extraction)	Prompt Engineering (Python + Gemini)
5	Database	Storage of vehicle-related information (if applicable)	SQLite / Local Storage
6	File Storage	Temporary storage of uploaded vehicle images	Local File System
7	External API-1	AI model integration for vehicle analysis	Google Gemini Flash API
8	Machine Learning Model	Image-based vehicle understanding and detail generation	Gemini Flash Multimodal Model
9	Infrastructure (Server / Cloud)	Application deployment and hosting	Streamlit Cloud / Local System

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1	Open-Source Frameworks	Frameworks used for building application	Streamlit, Python
2	Security Implementations	Secure API key storage and input validation	Environment Variables, HTTPS
3	Scalable Architecture	Modular design separating UI, Logic, and AI model integration	Layered Architecture
4	Availability	Application accessible via browser once deployed	Streamlit Cloud
5	Performance	Optimized image validation and fast AI response handling	Python Optimization + API Response Handling