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

| | |
|---|---|
| Date | 07 February 2026 |
| Team ID | LTVIP2026TMIDS88090 |
| Project Name | AUTOSAGE APP USING GEMINI FLASH |
| 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: Order processing during pandemics for offline mode**

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

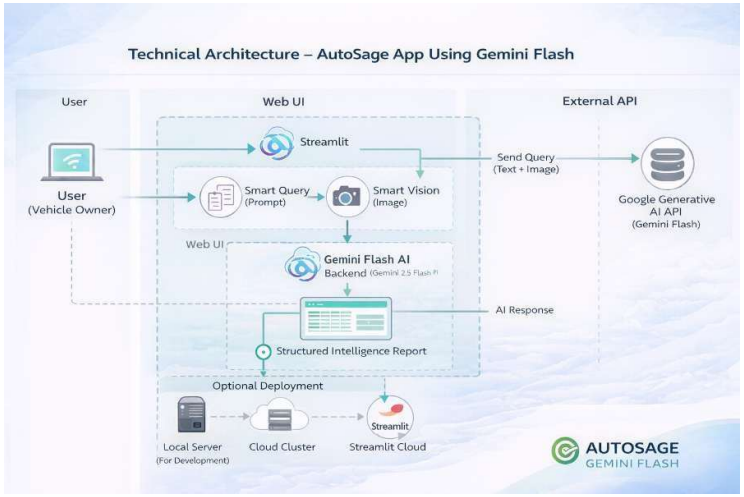

**Table-1 : Components & Technologies:**

| S.No | Component | Description | Technology |
|---|---|---|---|
| 1. | User Interface | Web-based interface where users submit vehicle queries (text/image) and view structured intelligence reports. | Streamlit (Python), HTML, CSS |
| 2. | Application Logic-1 | Handles user inputs, prompt construction, validation, and report formatting logic | Python |
| 3. | Application Logic-2 | Multimodal AI processing for vehicle analysis (text + image understanding) | Google Gemini 2.5 Flash (Generative AI API) |
| 4. | Application Logic-3 | Vehicle intelligence structuring, powertrain determination rules, Indian market contextualization | Custom Prompt Engineering + Python Processing |
| 5. | Database | (Optional) Store user queries, analysis history, logs. | SQLite / PostgreSQL (Optional) |
| 6. | Cloud Database | Deployment and hosting environment | Streamlit Cloud / Local Server. |
| 7. | File Storage | Temporary storage of uploaded vehicle images for processing | Local Filesystem (Streamlit Upload Handler) |
| 8. | External API-1 | AI model API for generating structured vehicle intelligence reports | Google Generative AI API (Gemini Flash) |

**Table-2: Application Characteristics:**

| S.No | Characteristics | Description | Technology |
|---|---|---|---|
| 1. | Open-Source Frameworks | Web application framework and supporting libraries used for building UI and backend logic | Streamlit, Python, Pillow, dotenv |
| 2. | Security Implementations | Secure API key management, environment variable protection, HTTPS-based API communication, input validation | Environment Variables (.env), HTTPS (TLS), Google API Security, Basic Input Validation |
| 3. | Scalable Architecture | Modular architecture separating UI, Prompt Engine, and AI Processing Layer. Can scale horizontally when deployed on cloud platforms | Streamlit Deployment + Google Gemini Cloud API (Cloud-based scalable inference) |
| 4. | Availability | Modular architecture separating UI, Prompt Engine, and AI Processing Layer. Can scale horizontally when deployed on cloud platforms | Streamlit Cloud / Cloud Hosting + Google Cloud Infrastructure |
| 5. | Performance | Optimized prompt engineering to reduce token usage; lightweight UI; cloud-based AI processing; controlled temperature & max token configuration | Gemini 2.5 Flash (Low Latency Model), Python Optimization, Configured Token Limits |