

INTRODUCTION

AUTOSAGE APP USING GEMINI FLASH is an AI-powered Vehicle Intelligence and Advisory System designed to deliver structured, data-driven automotive insights using Google Gemini Flash.

The application enables users to:

- Submit vehicle-related queries (text-based analysis)
- Upload vehicle images for visual identification and technical breakdown
- Combine image + prompt for advanced multimodal intelligence

The system analyzes user input and generates a highly structured automotive intelligence report covering:

- Vehicle identity and specifications
- Engine & performance analysis
- Efficiency and cost insights
- Safety and feature breakdown
- Market positioning and pricing (Indian context)
- Ownership, depreciation, and resale analysis
- Expert recommendation summary

The primary objective of the project is to bridge the gap between general automotive search results and professional-grade vehicle intelligence by delivering consistent, structured, and market-relevant insights.

The application leverages:

- Gemini 2.5 Flash for AI-driven content generation
- Streamlit for the interactive user interface
- Python-based backend logic
- Image processing for multimodal analysis

Project Overview

AUTOSAGE APP USING GEMINI FLASH is an AI-driven Vehicle Intelligence & Advisory System developed to provide structured, analytical, and market-focused automotive insights using Google Gemini Flash.

The application supports three core interaction modes:

- Text-Based Query Analysis – Users can ask vehicle-related questions (buying decisions, maintenance, comparisons, eco-friendly options, etc.).
- Image-Based Vehicle Analysis – Users can upload vehicle images for AI-powered identification and structured specification breakdown.
- Multimodal Analysis (Image + Prompt) – Combines user query with visual input for deeper and more contextual intelligence generation.

The system is designed specifically for the Indian automobile market, ensuring:

- INR-based pricing
- Indian variant alignment
- Market-relevant competitor comparison
- Realistic efficiency and ownership cost estimation

Purpose

The purpose of the AUTOSAGE APP USING GEMINI FLASH project is to design and develop an AI-powered Vehicle Intelligence & Advisory System that delivers structured, data-driven, and market-relevant automotive insights for users in the Indian automobile market.

This system aims to:

- Provide accurate and structured vehicle analysis using text, image, or multimodal inputs.
- Assist users in making informed buying decisions through pricing, competitor comparison, and value analysis.
- Support ownership planning by offering insights on maintenance, efficiency, reliability, and depreciation.
- Enable visual vehicle recognition and intelligent validation using AI-based image analysis.
- Deliver consistent, professional-grade reports using controlled prompt engineering and structured output formatting.

IDEATION PHASE

Problem Statement

The Indian automobile market offers a wide range of vehicles across segments, fuel types, and price categories. However, customers often struggle to obtain structured, reliable, and comparative vehicle intelligence in a single place.

Users face the following challenges:

- Scattered and inconsistent vehicle information across multiple websites.
- Difficulty comparing specifications, pricing, resale value, and ownership cost.
- Limited ability to validate vehicle details from images alone.
- Lack of personalized insights based on purpose (buying decision, maintenance, eco-friendly choice).
- Overexposure to marketing-driven content instead of analytical evaluation.

As a result, customers experience confusion, decision fatigue, and reduced confidence when selecting or evaluating a vehicle.

The problem is to develop an AI-powered system that can:

- Analyze vehicle information using text, image, or both.
- Generate structured, market-aligned intelligence reports.
- Provide data-driven insights for buying, ownership, and resale decisions.
- Ensure clarity, consistency, and reliability tailored to the Indian automobile market.

Empathy Map Canvas

Project: AUTOSAGE APP USING GEMINI FLASH

User Persona: Indian Vehicle Buyer / Vehicle Research User

Says

- “Which vehicle is best under my budget?”
- “Is this car worth buying?”
- “What is the real mileage?”
- “Is this EV better than petrol?”
- “What are the maintenance costs?”

Thinks

- “Am I making the right decision?”
 - “Will resale value be good after 5 years?”
 - “Is this information reliable?”
 - “Is EV really cost-effective long term?”
 - “What hidden costs am I missing?”
 - “Is this vehicle suitable for city use?”
-

Feels

- Confused due to too many options.
 - Concerned about financial commitment.
 - Excited about owning a new vehicle.
 - Overwhelmed by technical specifications.
 - Skeptical about marketing claims.
 - Cautious about long-term reliability.
-

Hears

- Friends recommending specific brands.
 - Dealers promoting higher variants.
 - Online influencers giving mixed reviews.
 - News about fuel price increases.
 - Discussions about EV adoption in India.
 - Social media comparisons.
-

Sees

- Advertisements highlighting features.
- Multiple comparison websites with conflicting data.

- YouTube review videos.
 - Rising EV options in the market.
 - Price fluctuations and new launches.
 - Competitive models in the same segment.
-

Does

- Searches vehicle reviews online.
 - Compares price and features.
 - Checks mileage and maintenance cost.
 - Visits dealership showrooms.
 - Reads user reviews and ratings.
 - Asks friends and family for advice.
-

Pain Points

- Information overload.
 - Lack of structured comparison.
 - Unclear variant-level differences.
 - Hidden ownership costs.
 - Conflicting data sources.
-

Goals

- Make a confident buying decision.
- Understand true ownership cost.
- Get structured and reliable analysis.
- Compare ICE vs EV effectively.
- Save time in research.

Brainstorming

Brainstorming was conducted to explore multiple solution approaches for addressing the problem of unstructured, unreliable, and overwhelming vehicle information available to Indian users.

The session focused on identifying how AI can simplify vehicle research, improve decision-making, and provide structured intelligence rather than generic reviews.

Identified Ideas

1. AI-Based Vehicle Intelligence Engine

Use Gemini Flash to generate structured, analytical vehicle reports instead of conversational responses.

2. Multimodal Analysis (Image + Text)

Allow users to upload a vehicle image and receive detailed insights based on visual identification.

3. Purpose-Based Analysis Mode

Adjust output dynamically for:

- Buying Decision
- Maintenance Guidance
- Eco-Friendly Search

4. ICE vs EV Intelligent Detection

Automatically determine powertrain type using:

- User query
- Visual cues
- Market data

5. Cost Transparency Module

Provide realistic:

- Cost per 1,000 km
- Maintenance estimates
- Depreciation analysis

6. Structured Market Intelligence Report Format

Replace scattered content with:

- Technical specs
- Ownership insights
- Resale strength
- Value score

7. Confidence-Level Output

Display identification confidence to increase trust.

8. Session-Based Context Controls

Allow users to define vehicle type and purpose via control panel.

REQUIREMENT ANALYSIS

Customer Journey map

The customer journey map outlines the step-by-step experience of a user interacting with the AutoSage application, from awareness to decision-making.

Stage 1: Awareness

User Action:

User realizes difficulty in finding structured and reliable vehicle information online.

Touchpoints:

- Online search
- Word of mouth
- Social media

User Goal:

Find a trustworthy and analytical vehicle advisor.

Pain Points:

- Too many biased reviews
- Unclear technical specifications
- Confusing price variations

Emotional State:

Curious but uncertain

Stage 2: Exploration

User Action:

User opens AUTOSAGE APP USING GEMINI FLASH and explores features.

Touchpoints:

- Prompt tab
- Image analysis tab
- Control panel (Vehicle type & Purpose selection)

User Goal:

Understand how the system works.

Pain Points:

- Unsure if AI output will be accurate
- Wants structured information

Emotional State:

Interested and testing reliability

Stage 3: Interaction**User Action:**

- Enters vehicle query OR
- Uploads vehicle image OR
- Uses Image + Prompt together

System Response:

Generates structured vehicle intelligence report.

User Goal:

Get clear technical, financial, and ownership insights.

Pain Points:

- Wants realistic pricing
- Needs resale and maintenance clarity

Emotional State:

Engaged and analytical

Stage 4: Evaluation**User Action:**

Reviews generated intelligence report.

User Checks:

- Fuel efficiency
- Maintenance cost
- Depreciation
- Value score

- Competitors

User Goal:

Compare options and reduce risk.

Pain Points:

- Concern about hidden costs
- Long-term reliability doubts

Emotional State:

Confident if report is structured and logical

Stage 5: Decision

User Action:

Uses insights to:

- Shortlist vehicle
- Make buying decision
- Plan maintenance
- Explore EV alternative

Outcome:

Improved clarity and informed decision-making.

Emotional State:

Confident and satisfied

Stage 6: Retention

User Action:

Returns to platform for:

- Maintenance guidance

Comparison reports

EV cost analysis

Outcome:

Builds trust with AutoSage as long-term vehicle advisor.

Solution Requirement

1. Functional Requirements

The system must provide the following core functionalities:

FR-1: User Query Processing

- Accept text-based vehicle queries.
- Interpret user intent (Buying, Maintenance, Eco-Friendly, etc.).
- Generate structured vehicle intelligence reports.

FR-2: Image-Based Vehicle Analysis

- Accept vehicle image uploads (JPG, PNG).
- Identify vehicle brand/model using visual cues.
- Detect powertrain type (ICE / EV) using exhaust, badging, charging port visibility.

FR-3: Multimodal Processing (Image + Prompt)

- Combine user text query and image input.
- Use text for primary identification and image for validation.
- Generate a consolidated structured report.

FR-4: Structured Output Generation

- Produce reports in predefined structured format.
- Ensure Indian market pricing in INR (₹).
- Include only relevant efficiency section (ICE or EV).

FR-5: Control Panel Personalization

- Allow user to select vehicle type (Car, Bike, EV, etc.).
- Allow user to select purpose (Buying Decision, Maintenance, Eco-Friendly).
- Adjust output depth based on selected purpose.

FR-6: AI Integration

- Integrate Gemini Flash model API.
 - Handle API requests and responses efficiently.
 - Display generation errors gracefully.
-

2. Non-Functional Requirements

NFR-1: Performance

- Response generation should occur within 3–5 seconds under normal load.

NFR-2: Reliability

- System must handle invalid inputs without crashing.
- Graceful failure handling for API errors.

NFR-3: Scalability

- Architecture should support increasing number of concurrent users.
- Modular structure for future feature expansion.

NFR-4: Usability

- Clean tab-based UI (Prompt / Image / Multimodal).
- Clear warnings for missing inputs.

NFR-5: Security

- Secure storage of API keys using environment variables.
- Prevent exposure of backend credentials.

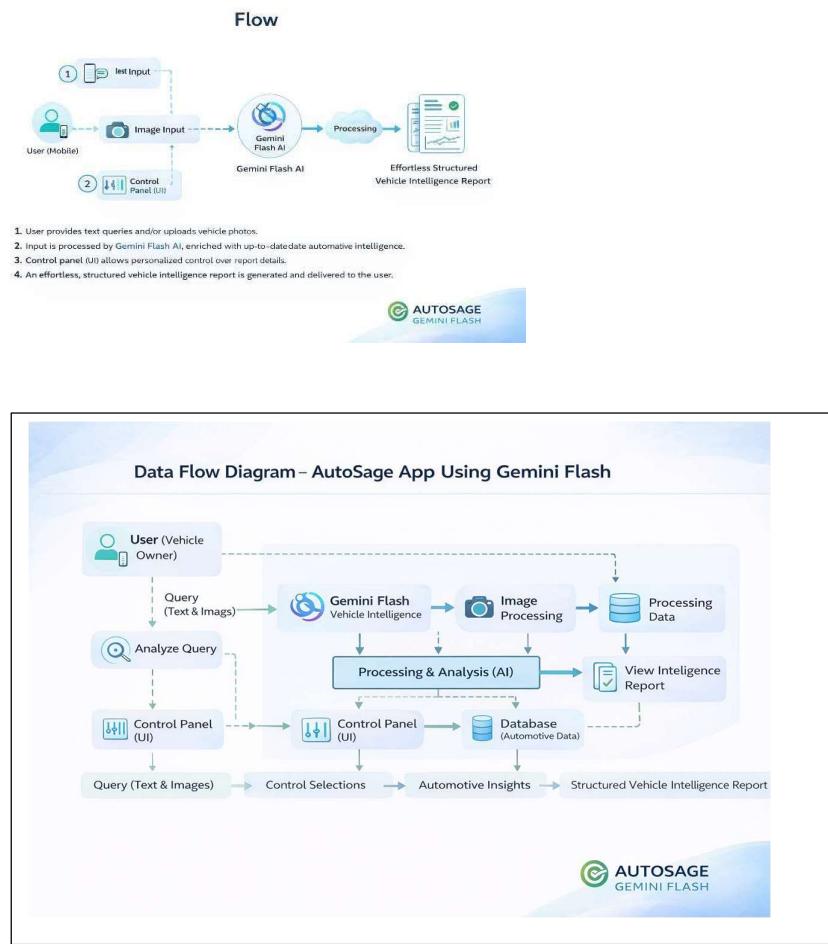
NFR-6: Maintainability

- Modular code structure (separate functions for prompt & image handling).
- Easy update of AI prompts without affecting UI logic.

3. Technical Requirements

- Frontend: Streamlit
- Backend Logic: Python
- AI Model: Gemini 2.5 Flash
- Image Processing: PIL
- Environment Management: dotenv
- Deployment: Web-based interface

Data Flow Diagram



Technology Stack

1. Frontend Layer

Responsible for user interaction and UI rendering.

- Framework: Streamlit
- UI Components: Tabs, Sidebar Controls, File Upload, Text Area
- Languages Used:
 - Python (UI scripting via Streamlit)
- Design Approach: Lightweight web-based interactive dashboard

2. Backend / Application Logic

Handles business logic, prompt engineering, and system validation.

- **Programming Language: Python**
- **Core Logic:**
 - Prompt Engineering
 - Multimodal Input Handling (Text + Image)
 - Validation & Error Handling

Libraries Used:

- PIL (Image Processing)
- dotenv (Environment Variable Management)
- os (System Integration)

3. AI & Machine Learning Layer

Responsible for intelligent vehicle analysis and content generation.

- AI Model: Gemini 2.5 Flash
- Integration Library: google-generativeai
- Capabilities Used:
 - Text Generation
 - Multimodal Processing (Image + Text)
 - Structured Report Generation

4. Data Handling Layer

- No traditional database (current version)
- Real-time inference-based architecture
- Stateless request-response AI pipeline

5. Security Layer

- API Key stored securely using .env file
 - No hardcoded credentials
 - Environment-based configuration
-

6. Deployment Environment

- Local Development Environment
- Web-based Interface via Streamlit
- Scalable for cloud deployment (Streamlit Cloud / AWS / Azure)

PROJECT DESIGN

Problem Solution Fit

Core Problem

Vehicle buyers and owners in India face:

- Fragmented information across multiple platforms
- Lack of structured vehicle comparison insights
- Confusion between ICE and EV ownership economics
- Limited ability to analyze vehicles from images
- Generic advice not aligned with user intent (buying / maintenance / eco-search)

This leads to:

- Poor purchase decisions
- Misinformation
- Time loss
- Reduced confidence in vehicle selection

Target Users

- Urban & semi-urban vehicle buyers
- First-time car/bike buyers
- EV comparison seekers
- Existing vehicle owners needing maintenance insights
- Users who only have a vehicle image but not model details

Proposed Solution

AUTOSAGE APP USING GEMINI FLASH provides:

- AI-powered structured vehicle intelligence reports
- Multimodal analysis (Text + Image)
- Indian market-specific pricing & specifications

- ICE vs EV powertrain determination logic
- Context-aware analysis based on user purpose
- Clean, professional report formatting

Problem	Solution by AutoSage
Scattered data	Unified structured AI-generated report
Hard to identify vehicle from image	Visual recognition + inference
EV vs ICE confusion	Automated powertrain logic
Unstructured online reviews	Analytical bullet-based format
No personalization	Sidebar-driven context (purpose + vehicle type)

Why the Fit is Strong

- Directly solves real buyer pain points
- Aligns with Indian automobile market context
- Reduces decision-making friction
- Uses AI where it adds real analytical value
- Scalable to dealerships, EV startups, or marketplaces

Validation Indicators

- Users get complete structured reports in seconds
- Supports multimodal inputs (modern UX expectation)
- Adapts output based on buying/maintenance/eco intent
- Eliminates need to browse 5–6 different websites

Proposed Solution

Problem to be Solved

Vehicle buyers and owners in India struggle with:

- Unstructured and scattered vehicle information
- Confusion between ICE and EV ownership economics
- Lack of reliable structured comparison data
- Difficulty identifying vehicles from images
- Generic advice not aligned with their purpose (buying, maintenance, eco-search)

This results in inefficient decision-making, misinformation, and reduced confidence.

Proposed Idea / Solution Description

AUTOSAGE APP USING GEMINI FLASH is an AI-powered vehicle intelligence and advisory system that:

- Generates structured vehicle analysis reports
- Supports text-only, image-only, and multimodal inputs
- Identifies vehicles using visual cues
- Determines powertrain type (ICE / EV) logically
- Provides Indian market pricing and specification insights
- Adapts analysis based on user-selected purpose
- Maintains professional, structured output formatting

The system leverages Gemini Flash for fast, scalable AI inference.

Novelty / Uniqueness

- Multimodal vehicle intelligence (Image + Prompt combined analysis)
- Automated ICE vs EV determination logic
- Context-aware analysis using sidebar controls
- Strictly structured professional automotive reports
- Indian market-focused pricing & ownership insights

Unlike generic chatbots, AutoSage provides analytical-grade automotive intelligence.

Social Impact / Customer Satisfaction

- Improves transparency in vehicle buying decisions
- Helps users compare EV vs ICE ownership realistically
- Reduces misinformation from informal sources
- Assists first-time buyers in understanding market positioning
- Promotes eco-conscious vehicle decisions

Enhances trust, clarity, and user confidence.

Business Model (Revenue Model)

Potential revenue streams:

- Freemium model (basic reports free, premium detailed insights paid)
 - Subscription for dealerships and automobile consultants
 - API integration for automotive platforms
 - Affiliate commissions from vehicle marketplaces
 - Sponsored analytics from EV brands
-

Scalability of the Solution

- Built on cloud-based AI (Gemini Flash)
- Modular architecture (UI + Logic + AI layer)
- Expandable to:
 - Used car valuation
 - Insurance estimation
 - Loan eligibility assessment
 - EV charging station recommendations

The solution is scalable across web platforms, mobile apps, and enterprise integrations.

Solution Architecture

Architecture Overview

The solution architecture bridges user interaction and AI-powered vehicle intelligence generation.

The system follows a modular layered architecture to ensure scalability, maintainability, and performance.

It consists of:

- Presentation Layer (User Interface)
 - Application Layer (Business Logic)
 - AI Processing Layer (Gemini Flash Integration)
 - Data & Configuration Layer
-

Architectural Layers

Presentation Layer (Frontend)

Purpose:

Handles user interaction and input collection.

Components:

- Smart Query Tab (Text Input)
- Smart Vision Tab (Image Input)
- Multimodal Analysis Tab (Image + Prompt)
- Sidebar Controls (Vehicle Type, Purpose Selection)

Technology Used:

- Streamlit (Web Interface)
 - HTML / CSS (UI Rendering)
-

B. Application Layer (Business Logic)

Purpose:

Processes user inputs and constructs structured prompts.

Responsibilities:

- Input validation
- Prompt construction logic
- Powertrain determination rules
- Context injection (Vehicle Type & Purpose)
- Output formatting enforcement
- Error handling

Technology Used:

- Python
-

C. AI Processing Layer

Purpose:

Generates structured vehicle intelligence reports.

Component:

- Gemini Flash Model (Google Generative AI)

Responsibilities:

- Text analysis
- Image understanding
- Multimodal reasoning
- Structured report generation

Model Used:

- gemini-2.5-flash
-

D. Data & Configuration Layer

Purpose:

Manages configuration and environment settings.

Components:

- Environment variables (.env file)
- API Key Management
- Session State Storage

Technology Used:

- Python dotenv
 - Streamlit session state
-

Data Flow

User provides:

- Text / Image / Both

System validates input

Prompt is dynamically generated

Image (if available) converted to byte format

Structured prompt + image sent to Gemini Flash

Model returns structured intelligence report

Output rendered on UI

Architectural Characteristics

- Modular Design
- Cloud AI Integration
- Multimodal Capability
- Stateless Processing (per request)

- Easily scalable to microservices
-

Deployment Model

- Web-based deployment
 - API-based AI inference
 - Ready for cloud hosting (GCP / AWS / Azure)
-

Future Scalability

The architecture allows integration of:

- Database layer for storing vehicle reports
- User authentication module
- Payment gateway
- Vehicle comparison engine
- Real-time price API integration

PROJECT PLANNING & SCHEDULING

Project Planning

Project Objective

To design and develop an AI-powered vehicle intelligence and advisory system that provides structured, market-relevant automotive analysis using Gemini Flash (multimodal capabilities).

Project Scope

In Scope:

- Smart Query (Text-based vehicle analysis)
- Smart Vision (Image-based vehicle identification)
- Multimodal Analysis (Image + Prompt)
- Structured intelligence report generation
- Indian automobile market focus
- Sidebar context controls (Vehicle Type & Purpose)

Out of Scope:

- User authentication system
- Database persistence
- Payment integration
- External pricing APIs
- Mobile app deployment

Phase	Activity	Duration	Dates
Phase 1	Requirement Analysis & Planning	1 Day	12 Feb
Phase 2	UI Design (Streamlit Layout & Tabs)	1 Day	13 Feb
Phase 3	Prompt Engineering & AI Integration	1 Day	14 Feb
Phase 4	Multimodal Processing & Validation Logic	1 Day	15 Feb
Phase 5	Testing, Debugging & Optimization	1 Day	16 Feb

Development Approach

- Agile-inspired short sprint model
- Iterative prompt refinement
- Continuous testing after each module
- Feature validation before moving to next stage

Key Deliverables

- Functional Web Application (Streamlit-based)
- Structured Automotive Intelligence Reports
- Multimodal Image + Text Analysis
- Clean UI with intelligent controls
- Documentation (SRS, Architecture, Planning)

Risk	Impact	Mitigation
Token Limit Exceeded	Medium	Prompt compression strategy
AI Hallucination	Medium	Strict formatting constraints
Image Misidentification	Medium	Confidence scoring logic
API Failure	High	Error handling & fallback messages

Resource	Role
Developer	UI, Backend, Prompt Engineering
Gemini Flash API	AI Intelligence Engine
Streamlit	Frontend Framework
Python	Core Logic Implementation

Success Criteria

- Structured output without formatting errors
- Accurate ICE/EV detection logic
- Context-aware analysis (Purpose-based adjustment)
- Stable AI response time
- Clean and professional UI

Performance Testing

Objective

To evaluate the system's responsiveness, stability, scalability, and API efficiency under normal and peak usage conditions.

Performance Test Areas

1. Response Time Testing

Objective: Measure time taken to generate AI reports.

- Test Scenario: Submit text query (Smart Query tab)
 - Expected Response Time: \leq 5 seconds (average)
 - Multimodal (Image + Prompt): \leq 8 seconds (average)
 - Result Criteria: System must respond without timeout or crash.
-

2. Load Testing

Objective: Validate system behavior under concurrent users.

- Scenario: Multiple users sending queries simultaneously.
 - Target Load: 10–20 concurrent requests.
 - Expected Result:
 1. No API failure
 2. No UI freezing
 3. Stable memory usage
-

3. Stress Testing

Objective: Identify system breaking point.

- Scenario: Rapid consecutive API calls.
- Expected Behavior:
 - Graceful error handling
 - No application crash

- Clear error message if rate-limited
-

4. Image Processing Performance

Objective: Measure efficiency of image-based analysis.

- Scenario: Upload high-resolution vehicle images (2–5MB).
 - Expected Result:
 - Image processing without delay (>10 seconds unacceptable)
 - No memory overflow
 - Accurate analysis output
-

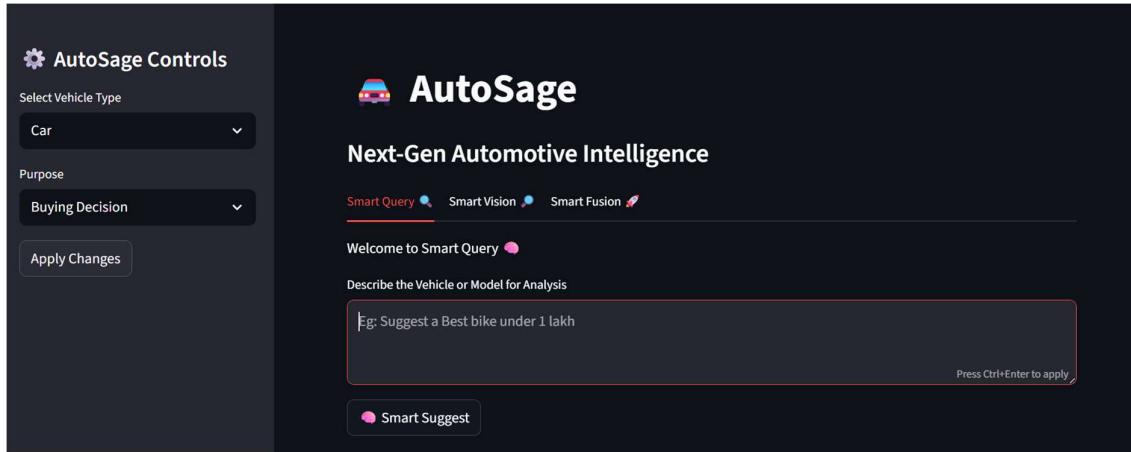
5. API Performance Test

- Objective: Validate Gemini Flash integration stability.
- **Validate:**
 - Correct API key authentication
 - Proper response formatting
 - Handling of token limits
- **Expected:**
 - No malformed responses
 - Structured report generation maintained

RESULTS

Output Screenshots

Home Page:



AutoSage Controls

Select Vehicle Type: Car

Purpose: Buying Decision

Smart Suggest

AutoSage

Next-Gen Automotive Intelligence

Smart Query Smart Vision Smart Fusion

Welcome to Smart Query

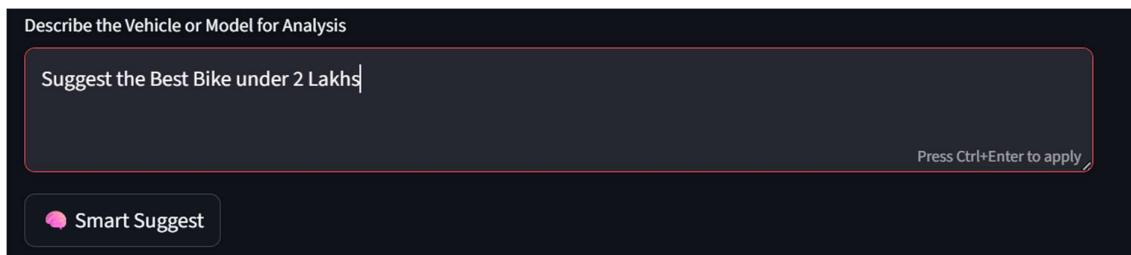
Describe the Vehicle or Model for Analysis

Eg: Suggest a Best bike under 1 lakh

Press Ctrl+Enter to apply

Smart Suggest

Prompt:



Describe the Vehicle or Model for Analysis

Suggest the Best Bike under 2 Lakhs

Press Ctrl+Enter to apply

Smart Suggest

Prompt Output:



STRUCTURED VEHICLE INTELLIGENCE REPORT

VEHICLE OVERVIEW

- Brand: TVS
- Model: Apache RTR 160 4V
- Variant: Dual Channel ABS (Assumed)
- Vehicle Type: Motorcycle
- Segment: 160cc Sport Commuter
- Launch Year (India): 2018 (4V series, with regular updates)
- Current Status: Active

ENGINE & PERFORMANCE

- Engine Options: Single 159.7cc, 4-stroke, 4-valve, Oil-cooled, Fuel Injection
- Engine Capacity: 159.7 cc
- Fuel Type: Petrol
- Power (bhp): 17.3-17.5 bhp (Sport Mode)
- Torque (Nm): 14.7-14.8 Nm (Sport Mode)
- Transmission: 5-speed manual
- Drivetrain: Chain Drive
- Performance Character: Sporty, agile, responsive

EFFICIENCY ANALYSIS

ICE:

- ARAI Mileage: 45-47 kmpl (Data may vary by variant - Approximate Indian specification.)
- Real-world Mileage: 40-42 kmpl
- Fuel Tank Capacity: 12 L
- Cost per 1,000 km: ₹2,500 - ₹2,700 (Approx. @ ₹105/L petrol)

Image Input:

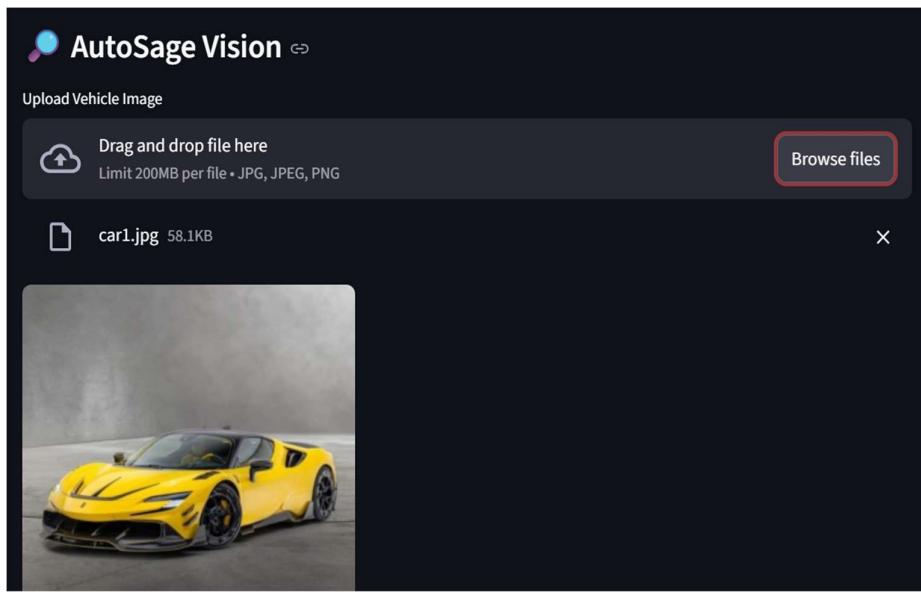


Image Output:

◆ **VEHICLE IDENTITY**

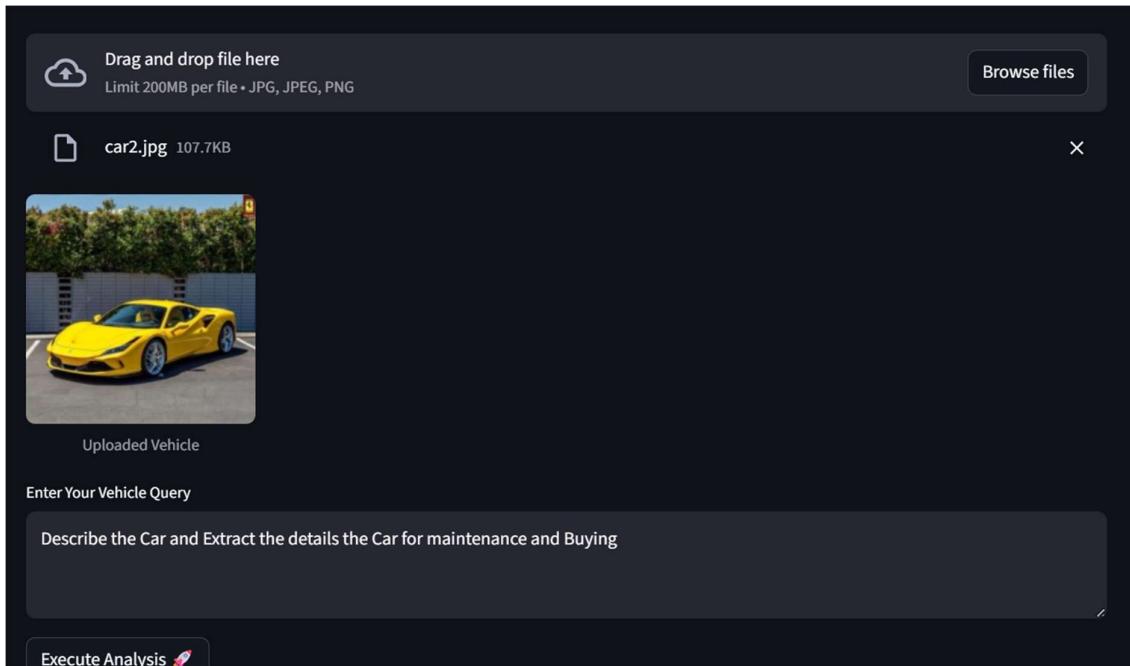
- Brand: Ferrari
- Model: SF90 Stradale
- Variant: Assetto Fiorano (Estimated, based on aggressive aerodynamic elements)
- Vehicle Type: Supercar
- Segment: Ultra-Luxury Performance
- Launch Year (India): 2020 (Estimated from market data)

◆ **ENGINE & PERFORMANCE**

- Engine Capacity: 4.0L Twin-Turbo V8 + 3 Electric Motors (Estimated from Indian market data)
- Fuel Type: Petrol + Electric (PHEV)
- Power Output (bhp): ~986 bhp (Combined) (Estimated from Indian market data)
- Torque (Nm): ~800 Nm (V8 only, combined higher) (Estimated from Indian market data)
- Transmission: 8-speed Dual-Clutch (Estimated from Indian market data)
- Drivetrain: All-Wheel Drive (AWD) (Estimated from Indian market data)
- 0-100 km/h: ~2.5 seconds (Estimated from Indian market data)
- Top Speed: ~340 km/h (Estimated from Indian market data)

◆ **MILEAGE & EFFICIENCY**

Prompt and Image input:



Output:

AUTOSAGE STRUCTURED VEHICLE INTELLIGENCE REPORT

↔

◆ INPUT ANALYSIS

- Input Type: Image
- Identification Confidence: High
- Identification Notes: Vehicle identified as a Ferrari F8 Tributo based on distinct design cues, including headlight air intakes, and overall body silhouette, validated by the Ferrari logo in the image.
- Powertrain Determination Basis: Visual identification of a high-performance sports car (Ferrari), known to be a Internal Combustion Engine (ICE) model, absence of EV charging port, and typical supercar design.

◆ VEHICLE IDENTITY

- Brand: Ferrari
- Model: F8 Tributo
- Variant: Coupe (Estimated - Based on available input)
- Vehicle Type: Sports Car / Supercar
- Segment: Luxury Performance

ADVANTAGES & DISADVANTAGES

ADVANTAGES

Multimodal Intelligence

Supports:

- Text-based vehicle queries
- Image-based vehicle identification
- Combined Image + Prompt analysis

This increases analytical depth and validation accuracy.

Structured Professional Reports

- Generates highly organized automotive intelligence reports
- Maintains Indian market relevance
- Eliminates unstructured AI output

Useful for academic, research, and advisory purposes.

Indian Market Specialization

- Focused pricing in INR (₹)
 - Indian segment classification
 - Local competitor comparison
 - Realistic mileage & ownership estimates
-

AI-Powered Automation

- Reduces manual vehicle research effort
 - Instant insights (buying, maintenance, resale)
 - Automated powertrain detection (ICE vs EV)
-

Scalable Architecture

- Built using Gemini Flash API
 - Easily extendable to mobile/web platforms
 - Can integrate databases and vehicle APIs in future
-

User-Centric Design

- Simple UI (Prompt / Image / Multimodal tabs)
- Context-aware output using sidebar controls
- Analytical customization based on purpose

DISADVANTAGES

Dependency on External API

- Requires active internet connection
 - Subject to API rate limits
 - Cost implications for large-scale usage
-

AI Estimation Limitations

- Some specifications are estimated
 - Exact variant-level accuracy not always guaranteed
 - Image clarity affects identification confidence
-

Performance Variability

- Large prompts may increase response time
 - High-resolution images can delay processing
 - Token limit constraints
-

No Real-Time Dealer Integration

- Does not fetch live dealership inventory
 - Pricing is market-estimated, not dynamic
-

Limited Offline Functionality

- Cannot function without API access
- No offline data model implemented

CONCLUSION

AUTOSAGE APP USING GEMINI FLASH is an AI-powered vehicle intelligence and advisory system designed to deliver structured, data-driven automotive insights for the Indian market.

By integrating text analysis, image recognition, and multimodal reasoning, the system enables users to obtain professional-grade vehicle evaluations covering performance, pricing, ownership, resale, and market positioning.

The platform reduces manual research effort, improves decision clarity, and enhances user experience through standardized analytical reporting.

While the system depends on external AI APIs and may rely on estimated data in certain cases, it demonstrates strong scalability potential and practical applicability in automotive advisory, research, and consumer decision-support environments.

Future Scope

The future scope of AUTOSAGE APP USING GEMINI FLASH includes technological enhancement, feature expansion, and large-scale deployment capabilities.

1. Advanced AI Enhancements

- Integration of fine-tuned automotive domain models for higher identification accuracy.
- Real-time vehicle recognition using live camera feed.
- Predictive analytics for resale value forecasting using historical datasets.
- AI-based comparison engine for multi-vehicle benchmarking.

2. Data Integration & Automation

- Integration with official automotive APIs for real-time pricing updates.
- Direct integration with dealership inventory systems.
- Automated insurance and loan eligibility estimation module.
- Real-time fuel price and electricity tariff integration.

3. Personalization & Smart Recommendations

- User profiling and behavioral-based vehicle suggestions.
- Budget-based intelligent filtering with financing simulations.
- Maintenance reminder and ownership tracking dashboard.

4. Enterprise & Commercial Expansion

- SaaS model for dealerships and automotive consultants.
- API-based vehicle intelligence service for third-party platforms.
- Integration with e-commerce automotive marketplaces.

5. Mobile & Deployment Enhancements

- Dedicated Android and iOS application.
- Progressive Web App (PWA) support.
- Multi-language support for regional Indian users.

APPENDIX

The appendix section provides supporting references, technical details, and supplementary materials relevant to the AUTOSAGE APP USING GEMINI FLASH project.

A. Abbreviations

- **AI** – Artificial Intelligence
- **ML** – Machine Learning
- **ICE** – Internal Combustion Engine
- **EV** – Electric Vehicle
- **ARAI** – Automotive Research Association of India
- **API** – Application Programming Interface
- **UI** – User Interface
- **UAT** – User Acceptance Testing
- **DFD** – Data Flow Diagram
- **NFR** – Non-Functional Requirement
- **FR** – Functional Requirement

B. Tools & Technologies Used

- **Frontend** – Streamlit (Python-based UI Framework)
 - **Backend Logic** – Python
 - **AI Engine** – Google Gemini 2.5 Flash Model
 - **Image Processing** – PIL (Python Imaging Library)
 - **Environment Management** – dotenv
 - **Deployment (Optional)** – Streamlit Cloud / Local Server
-

C. Key Assumptions

- Indian automobile market context only.
 - Prices represented in INR (₹).
 - Technical specifications may use estimated values when not publicly disclosed.
 - Identification confidence depends on clarity of user input (text/image).
-

D. Limitations

- Real-time dealership inventory integration not implemented.
 - No direct integration with official OEM databases.
 - EV charging cost estimation may vary by region.
 - Performance dependent on API response time.
-

E. References

- Official manufacturer websites (Indian market versions)
 - Automotive industry reports
 - ARAI published data
 - Public automotive comparison platforms
-

F. Supporting Documents

- Empathy Map Canvas
- Problem Statement Template
- Data Flow Diagram (DFD)
- Solution Architecture Diagram
- Test Case Report
- Product Backlog & Sprint Plan
- Performance Testing Report

Source Code

```
import time

from PIL import Image

import streamlit as st

import pandas as pd

from dotenv import load_dotenv

import os

load_dotenv()

api = os.getenv("GOOGLE_API_KEY")

import google.generativeai as genai

genai.configure(api_key=api)

# Model

model = genai GenerativeModel(


    model_name="models/gemini-2.5-flash",

    generation_config={

        "temperature": 0.3,

        "max_output_tokens": 4096

    }

)

# Page config

st.set_page_config(


    page_title="AutoSage",

    page_icon="📝",

    layout="wide"

)

# Sidebar
```

```
st.sidebar.title("⚙️ AutoSage Controls")

vehicle_type = st.sidebar.selectbox(
    "Select Vehicle Type",
    ["Car", "Bike", "Electric Vehicle", "Other"]
)

if vehicle_type == "Other":
    custom_type = st.sidebar.text_input(
        "Enter Vehicle Type"
    )
else:
    custom_type = vehicle_type

purpose = st.sidebar.selectbox(
    "Purpose",
    ["Buying Decision", "Maintenance Tips", "Eco-Friendly Search", "Other"]
)

if purpose == "Other":
    custom_purpose = st.sidebar.text_input(
        "Enter Purpose"
    )
else:
    custom_purpose = purpose

# st.sidebar.markdown("---")
# st.sidebar.info("AI features will be enabled soon")

# Initialize session state once
if "vehicle_type" not in st.session_state:
    st.session_state.vehicle_type = None
```

```

st.session_state.purpose = None

# Apply button
apply_btn = st.sidebar.button("Apply Changes")
# Apply changes only when button is clicked
if apply_btn:
    st.session_state.vehicle_type = custom_type
    st.session_state.purpose = custom_purpose
    st.success("Changes applied successfully ✅")

```

Tab 1:

```

def get_prompt_response(prompt):
    try:
        response = model.generate_content(prompt)
        return response.text
    except Exception as ex:
        return f"Error Generating Response {str(ex)}"

```

Tab 2

```

def input_image_setup(uploaded_file):
    if uploaded_file is not None:
        bytes_data = uploaded_file.getvalue()
        return [
            "mime_type": uploaded_file.type,
            "data": bytes_data
        ]
    return None

# # Tab 3
# def input_prompt_image(prompt1, uploaded_image):

```

```

# Main Title

st.title("🚗 AutoSage")
st.subheader("Next-Gen Automotive Intelligence")

tab1, tab2, tab3 = st.tabs(["Smart Query 🔎", "Smart Vision 📸", "Smart Fusion 🚀"])

with tab1:
    st.write("Welcome to Smart Query 💡")
    user_input = st.text_area(
        "Describe the Vehicle or Model for Analysis",
        placeholder = "Eg: Suggest a Best bike under 1 lakh",
        key = "prompt_tab"
    )
    analyze_btn_tab1 = st.button("💡 Smart Suggest", key = "prompt")

if analyze_btn_tab1 :
    if not user_input.strip():
        st.warning("Please provide vehicle details to proceed.")
    else:
        with st.spinner("Analyzing Vehicle Data..."):
            vehicle_context = st.session_state.vehicle_type or "Not Specified"
            purpose_context = st.session_state.purpose or "General analysis"

            prompt = f"""
                You are AutoSage AI — senior automotive market intelligence analyst for the Indian automobile
                industry.
            """

```

USER CONTEXT:

- Vehicle Type: {vehicle_context}
- Purpose: {purpose_context}

- Query: {user_input}

PURPOSE ADJUSTMENT:

- Buying Decision → Emphasize pricing, competitors, resale, value score.
- Maintenance Tips → Emphasize reliability, service cost, ownership risk.
- Eco-Friendly Search → Emphasize efficiency, emissions, cost per km, EV alternatives.

CORE RULES:

- Indian market only.
- Use latest generation sold in India.
- If variant unclear → "Most Common Variant (Assumed)".
- If uncertain → "Data may vary by variant - Approximate Indian specification."
- No filler text. No marketing tone.
- INR (₹) pricing only.
- Use realistic rounded ranges.
- Maintain strict structure.
- Do not output both ICE and EV sections.
- If data unavailable → "Information not publicly disclosed."

STRUCTURED VEHICLE INTELLIGENCE REPORT

◆ **VEHICLE OVERVIEW**

- Brand:
- Model:
- Variant:
- Vehicle Type:
- Segment:
- Launch Year (India):
- Current Status:

◆ ENGINE & PERFORMANCE

- Engine Options:
- Engine Capacity:
- Fuel Type:
- Power (bhp):
- Torque (Nm):
- Transmission:
- Drivetrain:
- Performance Character:

◆ EFFICIENCY ANALYSIS

(Include only relevant section)

ICE:

- ARAI Mileage:
- Real-world Mileage:
- Fuel Tank Capacity:
- Cost per 1,000 km:

EV:

- Battery Capacity:
- Claimed Range:
- Real-world Range:
- Charging Time:
- Charging Cost per Full Charge:

◆ DIMENSIONS & PRACTICALITY

- Boot Space:
- Seating Capacity:
- Ground Clearance:

- Practicality Score (1-10):

◆ KEY FEATURES (Top 7)

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.
- 7.

◆ SAFETY & TECHNOLOGY

- Airbags:
- ADAS Level:
- NCAP Rating:
- Safety Score (1-10):

◆ PRICE & POSITIONING (India)

- Ex-Showroom Range:
- On-Road Range:
- Top 4 Competitors:
- Value Score (1-10):

◆ OWNERSHIP

- Service Interval:
- Annual Maintenance Cost:
- Warranty:
- Reliability Score (1-10):

◆ DEPRECIATION

- 3-Year:

- 5-Year:
- Resale Strength (1-10):

◆ FINAL VERDICT

- Ideal Buyer:
- Pros:
- Cons:
- 3-Line Executive Summary:

""

```
try:
    response = get_prompt_response(prompt)
    st.markdown(response)
except Exception as e:
    st.error(f"AI Generation Failed: {str(e)}")
```

Tab 2:

with tab2:

```
st.subheader("🌐 AutoSage Vision")
# Image Upload Section
uploaded_image_tab2 = st.file_uploader(
    "Upload Vehicle Image",
    type=["jpg", "jpeg", "png"])
if uploaded_image_tab2:
    image = Image.open(uploaded_image_tab2)
    st.image(image, caption="Uploaded Vehicle", width=300)
# Action Button
analyze_btn_tab2 = st.button("🌐 Unlock Insights", key = "image_tab")
```

```
# Output Sections (Placeholders)

if analyze_btn_tab2:
    if uploaded_image_tab2 is None:
        st.warning("Please provide a vehicle image for processing.")
    else:
        with st.spinner("Processing Automotive Intelligence..."):
            image_prompt = f"""

                You are AutoSage AI — an expert automotive analyst specializing in the Indian automobile
                market.
```

TASK:

Analyze the provided vehicle image and generate a structured, professional, Indian-market vehicle intelligence report.

STRICT RULES:

1. Identify using visual cues only (logo, design, body type, badging, styling).
2. If variant unclear → mark "Estimated".
3. If not visually determinable → state "Not Visible - Estimated from market data".
4. All prices in INR (₹).
5. Professional tone. No filler text.
6. Use visual evidence first for identification.
7. If confidence low → state "Estimated Identification".
8. All non-visible technical data → label "Estimated from Indian market data".
9. Avoid exact fabricated numbers; use realistic Indian market ranges.
10. Do NOT break format.

RESPONSE FORMAT (STRICT)

◆ **VEHICLE IDENTITY**

- Brand:

- Model:
- Variant:
- Vehicle Type:
- Segment:
- Launch Year (India):

◆ ENGINE & PERFORMANCE

- Engine Capacity:
- Fuel Type:
- Power Output (bhp):
- Torque (Nm):
- Transmission:
- Drivetrain:
- 0-100 km/h:
- Top Speed:
- If powertrain unclear → "Estimated - Based on Market Variant"

◆ MILEAGE & EFFICIENCY

Determine ICE or Electric using visual cues (EV badge, charging port, exhaust absence).

Include ONLY relevant section.

(If ICE)

- ARAI Mileage:
- Real-world Mileage:
- Fuel Tank Capacity:
- Range:

(If Electric)

- Battery Capacity:
- Claimed Range:
- Charging Time:

- Cost per Full Charge:

◆ KEY FEATURES (Top 5)

- 1.
- 2.
- 3.
- 4.
- 5.

◆ SAFETY PACKAGE

- Airbags:
- ABS / EBD:
- ADAS:
- NCAP Rating:
- Key Safety Highlight:

◆ INTERIOR & COMFORT

- Infotainment:
- Connectivity:
- Seating Capacity:
- Boot Space:
- Premium Elements:

◆ PRICE ANALYSIS (India)

- Ex-Showroom Range:
- On-Road Estimate:
- Competitors:
- Value-for-Money (1-10):

◆ MAINTENANCE & OWNERSHIP

- Avg Annual Maintenance:

- Service Interval:

- Warranty:

- Spare Parts Cost Level:

◆ RESALE & LONG TERM VALUE

- 5-Year Depreciation:

- 10-Year Resale Estimate:

- Reliability (1-10):

◆ UNIQUE SELLING PROPOSITION

- Main USP:

- Ideal Buyer Profile:

◆ IDENTIFICATION CONFIDENCE

- Confidence Level (1-10):

- Reasoning Basis:

◆ FINAL VERDICT

4-5 line expert summary.

If image quality is poor, infer logically using visible design cues.

Maintain clarity and structured bullet format.

...

try:

```
input_image_data = input_image_setup(uploaded_image_tab2)
```

```
if input_image_data:
```

```
    response = model.generate_content([image_prompt, *input_image_data])
```

```
    st.markdown(response.text)
```

```
else:
```

```
    st.warning("Image Processing Failed")
```

```
except Exception as exe:  
    st.error(f"Error Generating Content: {str(exe)}")
```

Tab 3:

with tab3:

```
    st.subheader("Multimodal Analysis ⚡")  
  
    # Image Upload Section  
  
    uploaded_image_tab3 = st.file_uploader(  
        "Upload Vehicle Image",  
        type=["jpg", "jpeg", "png"],  
        key = "image_prompt"  
    )
```

```
if uploaded_image_tab3:  
    image = Image.open(uploaded_image_tab3)  
    st.image(image, caption="Uploaded Vehicle", width = 200)
```

```
user_prompt = st.text_area(  
    "Enter Your Vehicle Query",  
    placeholder = "Eg: Suggest a Best bike under 1 lakh",  
    key = "Prompt_image_tab"  
)
```

```
# Action Button  
  
analyze_btn_tab3 = st.button("Execute Analysis 🚀", key = "prompt_image_tab")
```

```
# Output Sections (Placeholders)
```

```
if analyze_btn_tab3:  
    vehicle_context = st.session_state.vehicle_type or "Not Specified"  
    purpose_context = st.session_state.purpose or "General Analysis"
```

```
prompt_and_image = f""
```

You are AutoSage AI — a senior automotive intelligence analyst with expertise in visual vehicle recognition and Indian automobile market analytics.

USER CONTEXT:

- Selected Vehicle Type: {vehicle_context}
- Selected Purpose: {purpose_context}
- User Query: {user_prompt}

INPUT TYPES YOU MAY RECEIVE:

1. Text only (vehicle name, model, variant, or description)
2. Image only (vehicle photo)
3. Both text + image

YOUR TASK:

Generate a highly structured, professional-grade automotive intelligence report using all available inputs.

PRIORITY LOGIC:

- If both image and text are provided → Use text for primary identification and image for validation.
 - If only image is provided → Identify vehicle using design cues, logos, badging, body type.
 - If only text is provided → Use Indian market knowledge.
 - If unsure → Clearly mark as "Estimated based on available input".
 - Never fabricate highly specific variant-level data without confidence.
 - If vehicle is discontinued, explicitly mention status.
- If exact variant cannot be confidently identified:
Set Variant as: "Most Common Variant (Estimated)"

POWERTRAIN DETERMINATION RULE:

Determine whether the vehicle is ICE or EV using:

- User text
- Visible exhaust presence
- EV badging
- Charging port visibility

If powertrain type cannot be confidently determined:

Set Fuel Type as: "Estimated - Based on Market Variant"

If EV → Fuel Type must be set as: Electric

If ICE → Specify Petrol / Diesel / CNG / Hybrid as applicable

STRICT RULES:

- All prices in INR (₹)
- Professional tone only
- No conversational filler
- No generic descriptions
- Clean structured bullet format
- Do NOT break format

Ensure internal consistency between:

- Fuel Type
- Engine specifications
- Efficiency section
- Pricing range

If input includes image:

All technical specifications not directly visible in the image must be labeled:

"Estimated - Based on Indian market data"

If input is text-only:

Use market data normally without over-labeling.

Ensure numerical values remain within realistic Indian market ranges.

Avoid unrealistic pricing or performance figures.

Do not contradict previously stated values across sections.

Do not omit any mandatory field in the defined structure.

AUTOSAGE STRUCTURED VEHICLE INTELLIGENCE REPORT

◆ INPUT ANALYSIS

- Input Type: (Text / Image / Both)
- Identification Confidence: (High / Medium / Low)
- Identification Notes:
- Powertrain Determination Basis:

◆ VEHICLE IDENTITY

- Brand:
- Model:
- Variant:
- Vehicle Type:
- Segment:
- Fuel Type:
- Launch Year (India):
- Current Status: (Active / Discontinued)

◆ ENGINE & PERFORMANCE

- Engine Options:
- Engine Capacity:
- Power Output (bhp):
- Torque (Nm):
- Transmission:
- Drivetrain:
- Performance Character:

◆ EFFICIENCY & RUNNING COST

Include ONLY the relevant subsection (ICE or EV).

Do NOT output both.

Do not leave the selected subsection empty.

(If ICE Vehicle)

- ARAI Mileage:

- Real-world Mileage:

- Fuel Tank Capacity:

- Estimated Cost per 1,000 km:

(If EV)

- Battery Capacity:

- Claimed Range:

- Real-world Range:

- Charging Time:

- Estimated Charging Cost per Full Charge:

◆ KEY FEATURES (Top 7)

1.

2.

3.

4.

5.

6.

7.

◆ SAFETY & TECHNOLOGY

- Airbags:

- ABS / EBD:

- ADAS Level:
- NCAP Rating:
- Advanced Safety Highlights:

◆ INTERIOR & PRACTICALITY

- Seating Capacity:
- Boot Space:
- Infotainment System:
- Connectivity Features:
- Premium Highlights:

◆ PRICE & MARKET POSITION (India)

- Ex-Showroom Price Range:
- On-Road Price Range:
- Primary Competitors:
- Market Positioning:
- Value Score (1-10):
- If numerical data is approximate, use rounded values instead of precise decimals.

◆ OWNERSHIP EXPERIENCE

- Service Interval:
- Estimated Annual Maintenance Cost:
- Warranty:
- Spare Parts Cost Level:
- Reliability Score (1-10):

◆ DEPRECIATION & RESALE

- 3-Year Depreciation Estimate:
- 5-Year Depreciation Estimate:
- 10-Year Resale Value Estimate:
- Resale Strength Score (1-10):

◆ BUYER FIT ANALYSIS

- Ideal Buyer Profile:
- Use Case Suitability:
- Pros:
- Cons:

◆ FINAL EXPERT VERDICT

Provide a 4-line executive summary.

ERROR HANDLING:

If identification confidence is Low and model cannot be reasonably inferred:

Return:

"Vehicle identification insufficient. Please provide clearer image or full model name."

Maintain strict formatting consistency.

Ensure logical coherence across sections.

Avoid speculative exaggeration.

...

```
image_input_data = input_image_setup(uploaded_image_tab3)
if not image_input_data:
    st.warning("Please upload an Image")
elif not user_prompt or not user_prompt.strip():
    st.warning("Please Enter Your Vehicle Query")
else:
    with st.spinner("Generating intelligent Report"):
        final_prompt = prompt_and_image + f"\n\nUSER QUERY: \n{user_prompt}"
```

```
try:  
    response = model.generate_content([final_prompt, *image_input_data])  
    st.markdown(response.text)  
  
except Exception as Exe:  
    st.error(f"AI Generation Error: {str(Exe)}")
```

GitHub & Project Demo Link

GitHub: <https://github.com/M-Sudheer18/autosage-intelligence.git>

Demo Link: https://drive.google.com/file/d/1xiKq-VoFinL_anu4hW68vsQvyX-jTZZ/view?usp=drive_link