1. INTRODUCTION

1.1 Project Overview

AutoSage is an AI-powered application that provides detailed information about two-wheeler and four-wheeler vehicles using image-based analysis. The system uses Google's Gemini Flash model to analyze vehicle images and generate structured details such as brand, model, mileage, price range, and resale value. AutoSage simplifies the vehicle selection process by delivering real-time insights through an easy-to-use Streamlit interface.

- **Scenario 1: Buying a New Motorcycle**
  Sarah uses AutoSage to compare motorcycle specifications, features, and prices within her budget, helping her make an informed purchase decision.
- **Scenario 2: Vehicle Maintenance Tips**
  AutoSage provides seasonal maintenance suggestions, such as checking tire pressure and battery health, to help users maintain vehicle safety and performance.
- **Scenario 3: Finding Eco-Friendly Vehicles**
  Emma uses AutoSage to explore electric and hybrid vehicles by understanding efficiency, environmental impact, and available incentives.

1.2 Objectives

The primary purpose of the **AutoSage project** is to develop an intelligent Generative AI–based system that can automatically analyse vehicle images and provide detailed vehicle information. It aims to assist users such as vehicle buyers, owners, and eco-conscious consumers by enabling quick and accurate understanding of two-wheeler and four-wheeler details reducing manual research and improving decision-making efficiency.

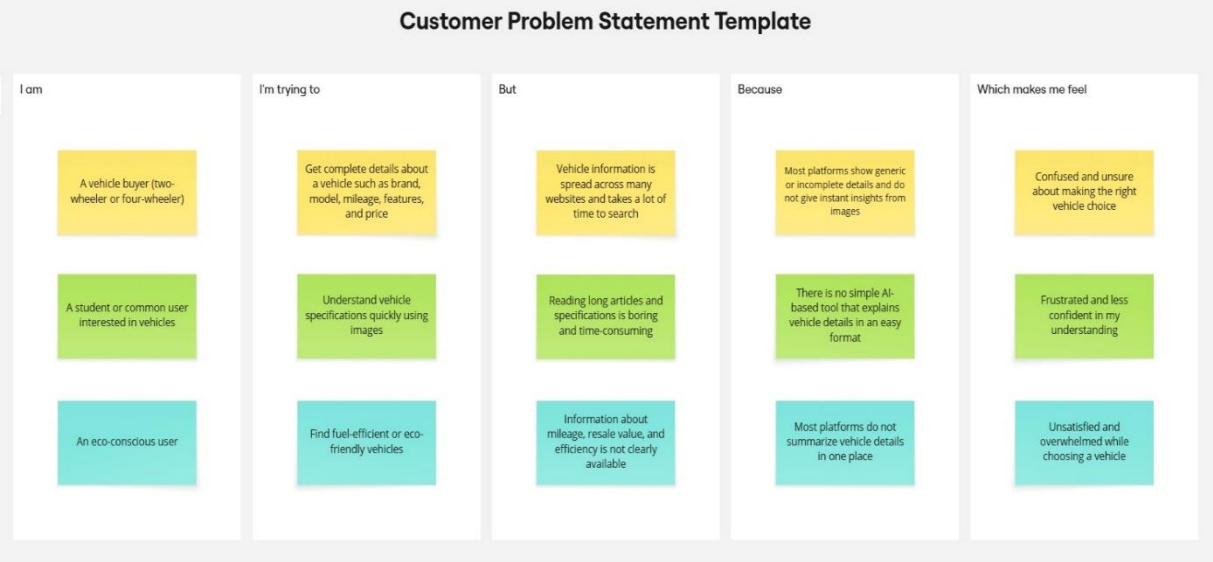
Key Objectives:

- To analyse vehicle images using a multimodal Generative AI model.
- To generate structured vehicle details such as brand, model, mileage, price and resale value.
- To build a simple user-friendly web application using Streamlit.
- To enable real-time vehicle comparison and information retrieval for practical use cases.

2. PROJECT INITIALIZATION AND PLANNING PHASE

2.1 Defining Problem Statement

Customers who want to buy or compare two-wheelers and four-wheelers often find it difficult to get clear and complete vehicle information in one place. They try to understand details such as brand, model, mileage, features, price, and resale value, but the information is scattered across multiple websites. Most platforms require manual searching and reading long, complex specifications, which is time-consuming and confusing. Existing systems do not use image-based AI to provide instant and structured insights. This makes customers feel frustrated, unsure, and less confident while making vehicle-related decisions.



Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	A vehicle buyer	Understand vehicle details before buying a bike or car	Searching for vehicle information takes too much time	Vehicle details are spread across many websites and need manual comparison	Confused and unsure about choosing right vehicle

PS-2	A student or general user	Identify and understand a vehicle using its image	Readin g ling specific ations and reviews is difficult	Most platforms do not support image-based vehicle analysis	Frustrated and less confident
PS-3	An eco-conscious user	Find fuel-efficiency or eco-friendly vehicles	It is hard to get clear informa tion about mileage and efficien cy	Vehicle informatio n is not summarize d in one place	Dissatisfied and overwhelmed

2.2 Project Proposal (Proposed Solution)

Project Overview	
Objective	The main objective of the AutoSage project is to build an application that gives vehicle details such as brand, model, mileage, price, and resale value by analyzing vehicle images using Gemini Flash.
Scope	The project allows users to upload images of two-wheelers and four-wheelers and receive vehicle information through an AI model. The project focuses only on providing information and does not include vehicle booking or payment features.
Problem Statement	

Description	Users find it difficult to collect complete vehicle information from one place. Vehicle details are available on many websites, which makes searching and comparing vehicles slow and confusing.
Impact	This problem affects users by wasting time and making it harder to choose the right vehicle. A simple and quick solution is needed to reduce confusion.
Proposed Solution	
Approach	AutoSage uses the Gemini Flash AI model to analyze vehicle images. A Streamlit web application is used to upload images and show vehicle details instantly.
Key Features	Upload vehicle images AI-based vehicle analysis Clear and structured output Easy-to-use interface Secure API key usage

Resource Requirements

Resource Type	**Description**	**Specification/Allocation**
Hardware		
Computing Resources	Laptop or Desktop	Standard System
Memory	RAM	Minimum 8 GB RAM
Storage	Disk space	10 – 20 GB
Software		
Frameworks	Python frameworks	Streamlit
Libraries	Additional Python libraries	Google-generativeai, python-dotenv, Pillow

Development Environment	IDE, version control	VS Code, Github
Data		
Data	User-uploaded vehicle images	Small to medium-sized images (JPG, JPEG, PNG)

2.3 Initial Project Planning

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

The following table represents the product backlog and sprint-wise planning for the **AutoSage** project.

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members	Sprint Start Date	Sprint End Date (Planned)
Sprint-1	User Interface Setup	USN-1	As a user, I want a Streamlit-based interface to upload vehicle images easily.	2	High	All Team members	28 January 2026	31 January 2026
Sprint-1	Image Upload Validation	USN-2	As a user, I want the system to validate the uploaded image format (jpg, jpeg, png).	1	High	All Team members	28 January 2026	31 January 2026
Sprint-2	AI Model Integration	USN-3	As a user, I want the system to analyze vehicle images using the Gemini Flash model.	3	High	All Team members	02 February 2026	09 February 2026
Sprint-2	Prompt Engineering	USN-4	As a user, I want the system to generate structured vehicle details like brand, mileage, and price.	2	High	All Team members	02 February 2026	09 February 2026

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members	Sprint Start Date	Sprint End Date (Planned)
Sprint-3	Output Display	USN-5	As a user, I want to view the generated vehicle details clearly on the screen.	2	High	All Team members	12 February 2026	18 February 2026
Sprint-3	Deployment	USN-6	As a user, I want the AutoSage application to be deployed and accessible through the browser.	2	Medium	All Team members	12 February 2026	18 February 2026

3. DATA COLLECTION AND PREPROCESSING PHASE

3.1 Data Collection Plan and Raw Data Sources Identified

Data Collection Plan Template

Section	Description
Project Overview	AutoSage is an AI-based application that analyzes vehicle images to provide structured vehicle information such as brand, model, mileage, price, and resale value.
Data Collection Plan	Data is collected directly from users in the form of uploaded vehicle images through the Streamlit web interface.
Raw Data Sources Identified	The raw data source consists of images of two-wheelers and four-wheelers uploaded by users during application usage.

Raw Data Sources Template

Source Name	Description	Location/URL	Format	Size	Access Permissions
User-uploaded vehicle images	Images of two-wheelers and four-wheelers uploaded by users	Local system via Streamlit app	JPG, JPEG, PNG	Small to Medium	Private(User provided)

3.2 Data Quality Report

Data Quality Report Template

Data Source	Data Quality Issue	Severity	Resolution Plan
User-uploaded vehicle images	Images may be blurry or low resolution	Moderate	Prompt users to upload clear images and rely on Gemini Flash's robustness
User-uploaded vehicle images	Different image sizes and orientations	Low	Images are resized automatically during display and preprocessing
User-uploaded vehicle images	Incorrect or unrelated images	Moderate	Application validates image format and processes only supported formats.

3.3 Data Preprocessing

Preprocessing Template

In the **AutoSage** project, vehicle images uploaded by users are preprocessed before analysis. The images are resized and normalized to maintain consistency. Basic noise handling and color processing are applied to improve image clarity. The complete image is used without heavy modifications to retain important vehicle details. These preprocessing steps help the AI model generate accurate and reliable vehicle information.

Section	Description
Data Overview	The data used in this project consists of vehicle images uploaded by users through the Streamlit interface. These

	images include two-wheelers and four-wheelers. No external dataset is used.
Image Validation	Uploaded images are checked to ensure they are in supported formats such as JPG, JPEG, or PNG before processing.
Image Handling	The uploaded image is loaded using the Pillow library and passed directly to the Gemini Flash model for analysis.
Prompt Formatting	A structured prompt is combined with the uploaded image to guide the AI model in generating organized vehicle details.
Model Input Processing	The image and prompt are sent to the Gemini Flash model, which internally handles feature extraction and image understanding.
Data Preprocessing Code:	
Loading Data	Vehicle images are collected directly from users through the Streamlit file uploader.
Input Validation	The application ensures an image is uploaded before sending the request to the AI model.
Prompt Creation	The validated image is paired with a predefined automobile-specific prompt.
Model Invocation	The combined image and prompt are sent to the Gemini 2.5 Flash model for content generation.
Output Handling	The generated vehicle details are received from the model and displayed clearly on the Streamlit interface.

4. MODEL DEVELOPMENT PHASE

4.1 Model Selection Report

Model Selection Report

In the AutoSage project, the focus is on selecting a **pre-trained multimodal generative AI model** capable of analyzing vehicle images and generating structured vehicle information in real time. Unlike traditional deep learning projects that require training convolutional neural networks (CNNs), this application leverages an existing large-scale generative model.

The model is selected based on its ability to handle **image and text inputs**, fast response time, ease of integration, and suitability for real-time vehicle information generation.

Model Selection Report:

Model	Description
Gemini 2.5 Flash (models/gemini-2.5-flash)	A pre-trained multimodal generative AI model designed for fast and efficient processing of image and text inputs. It provides real-time analysis of vehicle images and generates structured information such as brand, model, mileage, price, and resale value. Its low latency and high-quality responses make it ideal for the AutoSage application.

4.2 Initial Model Training Code, Model Validation and Evaluation Report

In this project, no custom model training is performed. Instead, a pre-trained generative AI model is integrated and used for recipe blog generation. The focus of this phase is on model selection, configuration, prompt design, and output evaluation, rather than training from scratch.

Initial Model Training Code (5 marks):

Model Selection and Initialization

The **Gemini Flash (models/gemini-2.5-flash)** model is selected for the **AutoSage** project due to its ability to process both image and text inputs, fast response time, and suitability for real-time applications.

The model is accessed using the Google GenAI SDK, and it is initialized securely using an API key stored in environment variables. The application sends a vehicle image along with a structured prompt to the model, which generates vehicle-related insights.

Since the model is already pre-trained, no training loop, loss function, or optimizer is required.

```
# Configure Gemini API Key

genai.configure(api_key=os.getenv("GOOGLE_API_KEY"))



# Load Google Gemini API and get response

def get_gemini_response(input_prompt, image):

    model = genai.GenerativeModel('models/gemini-2.5-flash')
    response = model.generate_content([input_prompt, image[0]])
    return response.text
```

Model Validation and Evaluation Report (5 marks):

Model	Summary	Training and Validation Performance Metrics
Gemini 2.5-flash	Pre-trained generative language model optimized for fast text generation	Pre-trained multimodal generative AI model capable of understanding images and generating structured text responses

5. MODEL OPTIMIZATION AND TUNING PHASE

5.1 Tuning Documentation

Model Optimization and Tuning Phase

The Model Optimization and Tuning Phase in the **AutoSage** project focuses on improving the quality, clarity, and reliability of AI-generated vehicle information. Since **AutoSage** uses a pre-trained Generative AI model, optimization is achieved through prompt refinement, response structuring, and inference configuration rather than traditional neural network training.

Hyperparameter Tuning Documentation (8 Marks):

Model	Tuned Hyperparameters
Gemini 2.5 Flash	**Temperature:** Set to a balanced value to ensure the generated vehicle details are clear and factual without unnecessary creativity. **Top-p:** Used to control token selection so that the responses remain relevant and consistent. **Top-k:** Helps limit token choices to avoid irrelevant or incorrect vehicle information. **Max Output Tokens:** Configured to ensure all required vehicle details such as brand, mileage, price, and resale value are generated completely. **Response Format:** Set to structured plain text for easy display in the Streamlit user interface.

5.2 Final Model Selection Justification

Final Model	Reasoning
Gemini 2.5 Flash *(models/gemini-2.5-flash)*	The Gemini 2.5 Flash model is selected due to its fast response time, strong multimodal capabilities, and efficient handling of image and text inputs. It provides accurate and structured vehicle information in real time and integrates seamlessly with the Streamlit application, making it well-suited for the AutoSage project.

6. RESULTS

6.1 Output Screenshots

The complete execution of the AutoSage application is shown in the images step by step as shown below.

Step 1: Run the app.py code using **streamlit run app.py** and you will get a link in terminal http://localhost:8501 which opens automatically in the browser.

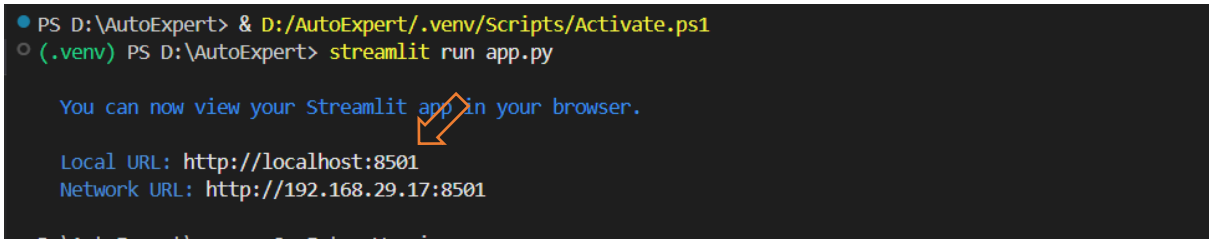


Fig 6.1.1: Code running in Terminal

Step 2: The link automatically opens to display the application.

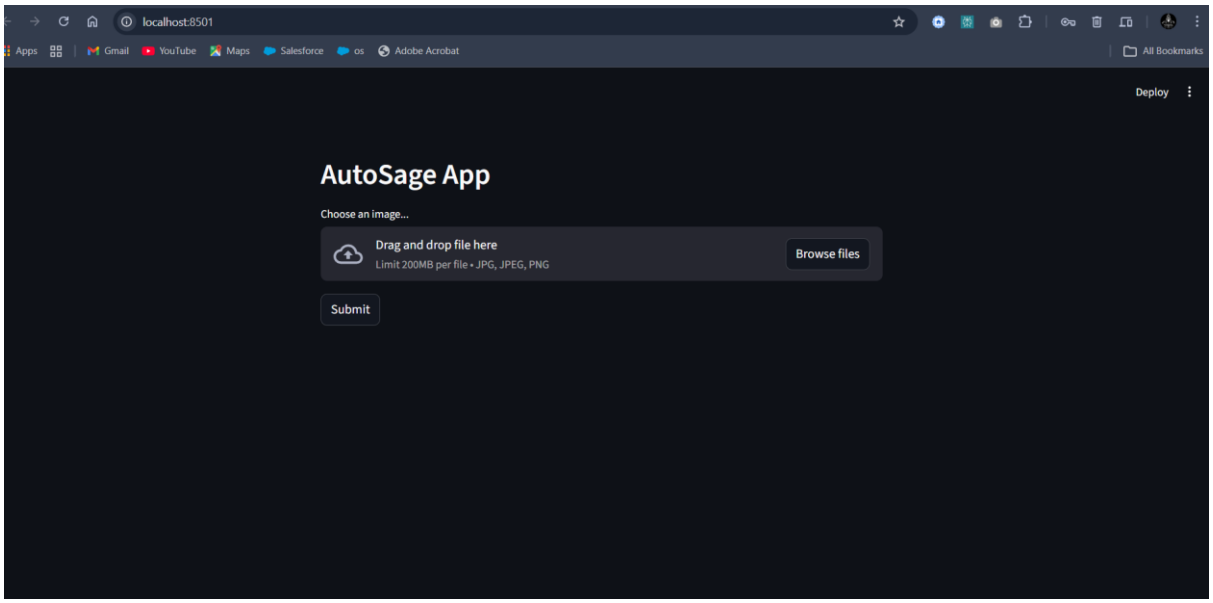


Fig 6.1.2: AutoSage App Page

Step 3: Click on **Browse Files** option to predict the vehicle details.

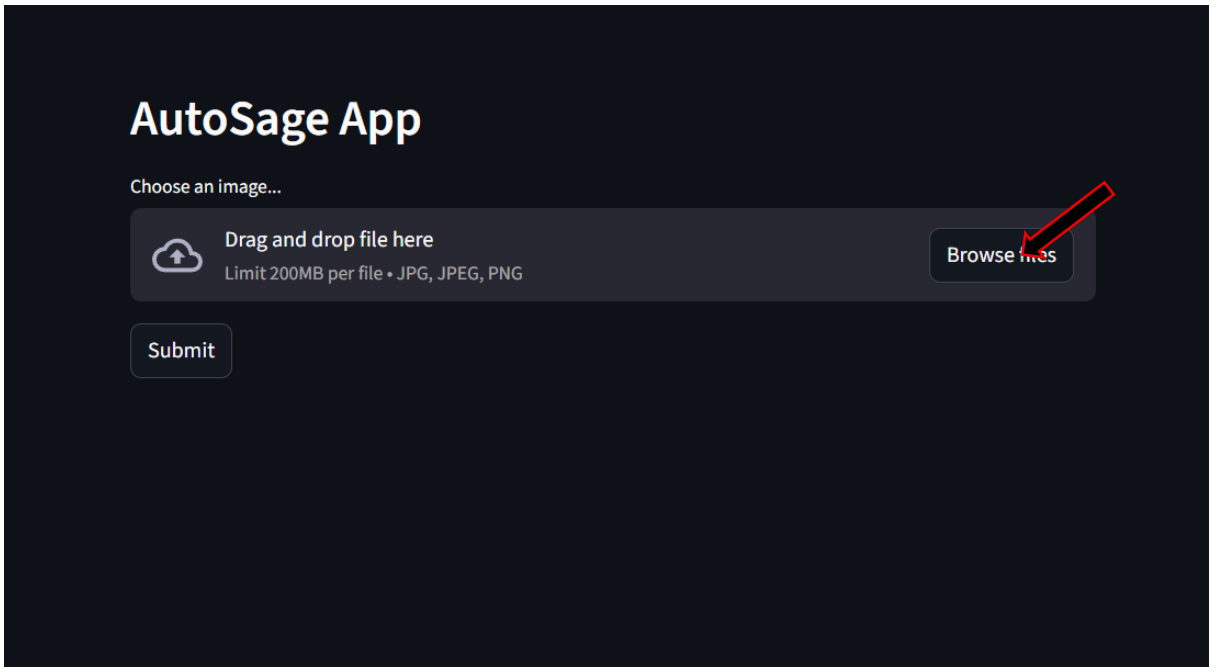


Fig 6.1.3: Prediction Page of AutoSage App

Step 4: Click on choose file option to choose the images that need to predict.

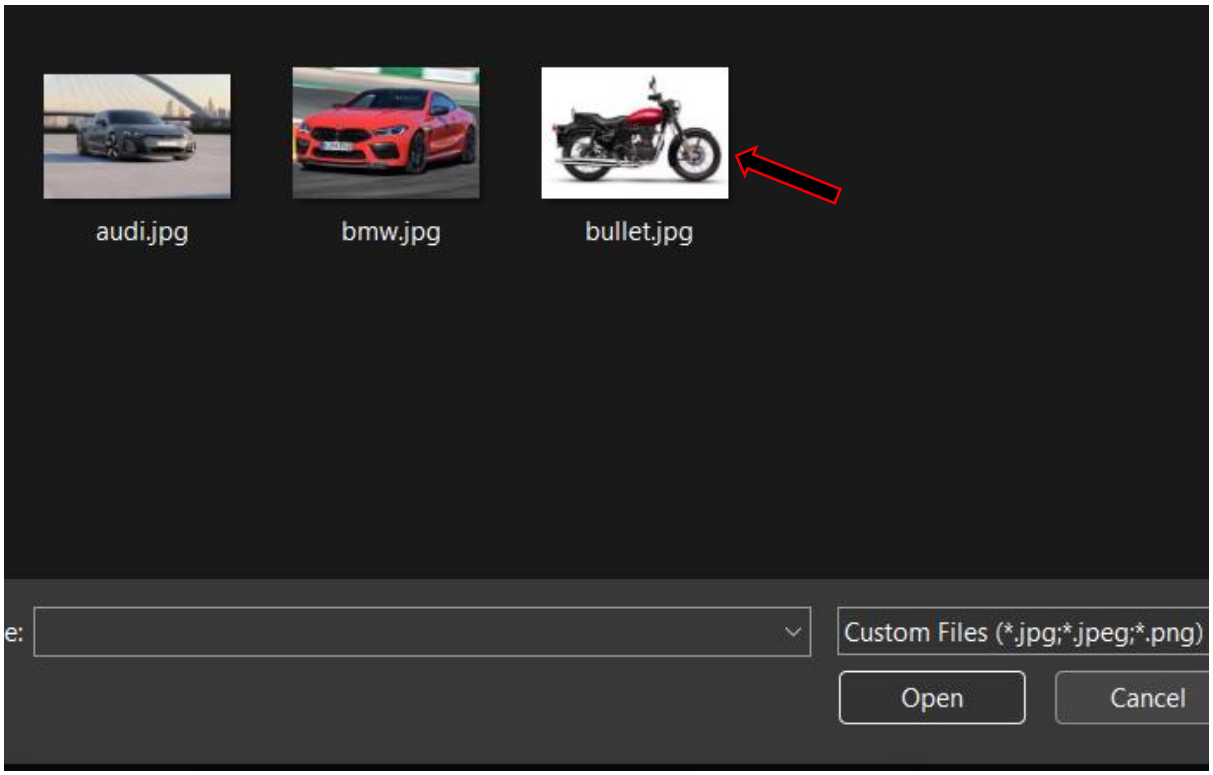


Fig 6.1.4: Window to choose image for prediction

Select any image for prediction and click on Open.

Step 5: The image loads into the application.

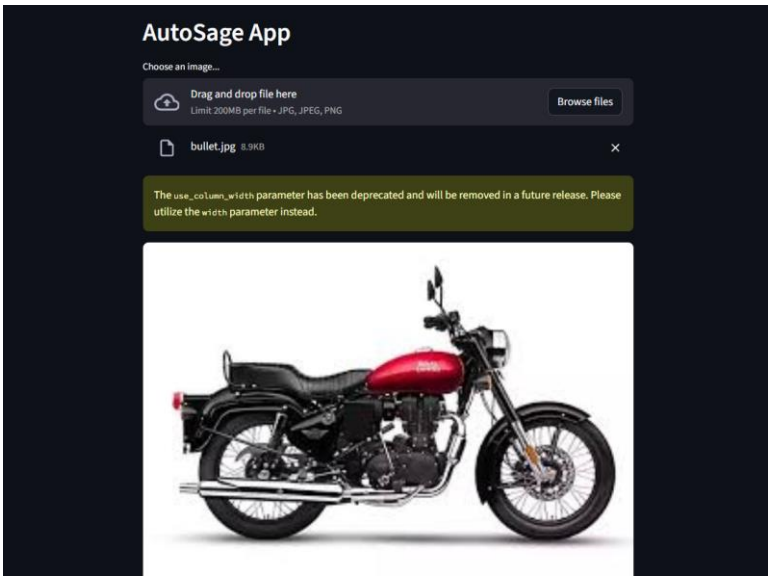


Fig 6.1.5: Image loads into the app.

Step 6: Now click on the Submit button to detect the type of vehicle and its details.

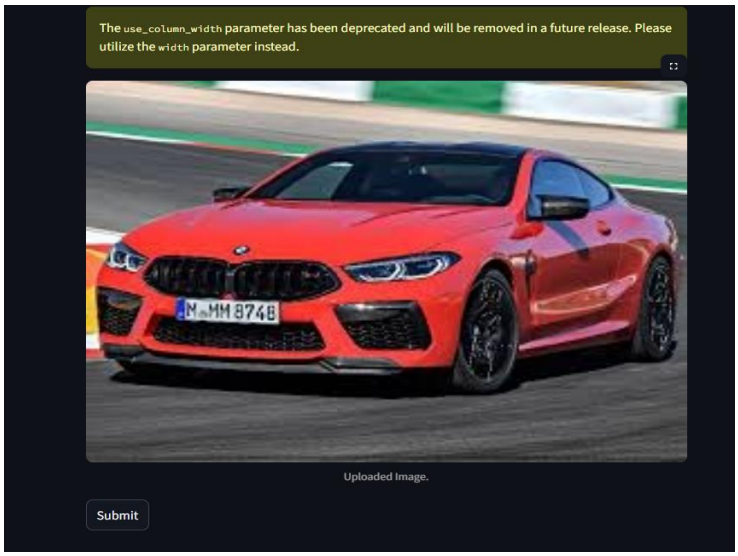
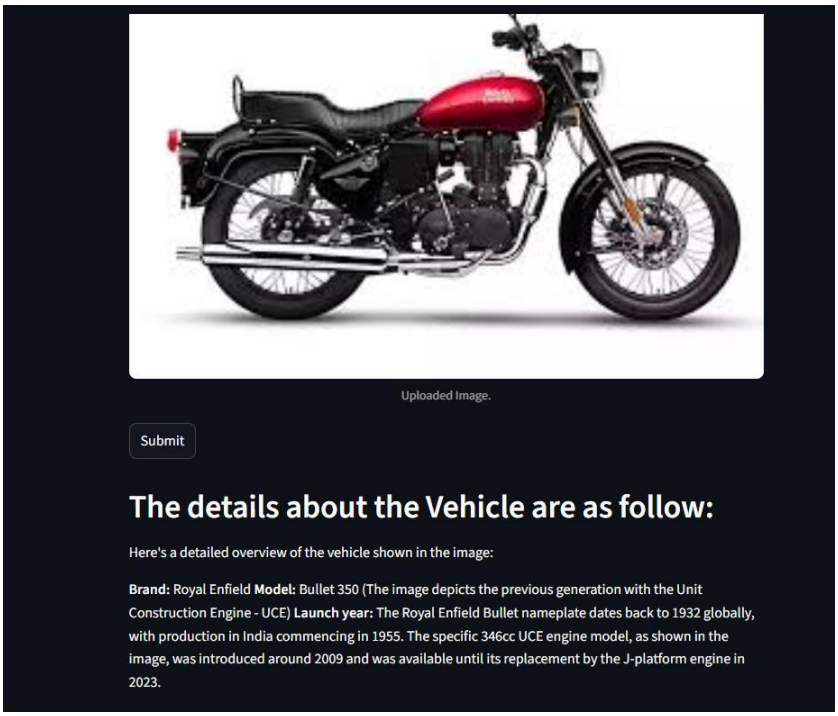


Fig 6.1.6: Click on submit to predict

Step 7: The model then predicts the type of the vehicle and provides its specifications as given below.



Uploaded Image.

Submit

The details about the Vehicle are as follow:

Here's a detailed overview of the vehicle shown in the image:

Brand: Royal Enfield **Model:** Bullet 350 (The image depicts the previous generation with the Unit Construction Engine - UCE) **Launch year:** The Royal Enfield Bullet nameplate dates back to 1932 globally, with production in India commencing in 1955. The specific 346cc UCE engine model, as shown in the image, was introduced around 2009 and was available until its replacement by the J-platform engine in 2023.

Key Features:

- **Engine Capacity:** 346cc, Single-cylinder, 4-stroke, Air-cooled, UCE engine.
- **Type:** Classic Cruiser Motorcycle.
- **Special Features:**
  1. **Iconic Retro Styling:** Features a timeless design with a distinctive, large teardrop fuel tank, traditional headlamp, comfortable spring-loaded seat, and an overall vintage aesthetic that has remained largely unchanged for decades.
  2. **Distinctive "Thump" Exhaust Note:** Known for its unique, deep, and rhythmic exhaust note, often referred to as the "thump," which is characteristic of Royal Enfield's long-stroke engine architecture.
  3. **Robust, All-Metal Construction:** Built with heavy-duty metal components throughout, giving it a solid, durable feel and strong road presence, which contributes to its perceived reliability and longevity.

Mileage: Approximately 30-35 km/l (depending on riding style, terrain, and maintenance).

Average Price in INR: For the UCE Bullet 350, when it was last available, the ex-showroom price ranged from **₹1,40,000 to ₹1,65,000**. (The newer J-platform Bullet 350 starts from around ₹1,73,000 ex-showroom).

Other Details:

- **Maintenance Costs:** Moderate. While spare parts are widely available and generally affordable, the traditional engine design requires regular and proper servicing (typically every 3000-5000 km) to ensure optimal performance and prevent issues. Mechanics familiar with Royal Enfield are plentiful across India.
- **Benefits:** Offers a comfortable and upright riding posture suitable for both city commutes and long-distance touring. It commands a strong road presence and is part of India's rich motorcycling heritage. The Bullet 350 provides a raw, mechanical, and engaging riding experience. It is also highly customizable, with a vast aftermarket accessory market.
- **Unique Selling Points:** Its unparalleled heritage and enduring legacy, the signature "thump" that resonates deeply with enthusiasts, its robust and sturdy build quality, and the emotional connection it

- **Unique Selling Points:** Its unparalleled heritage and enduring legacy, the signature "thump" that resonates deeply with enthusiasts, its robust and sturdy build quality, and the emotional connection it fosters with its riders, making it more than just a mode of transport – it's a lifestyle statement.

Approximate Resale Value: Royal Enfield motorcycles, especially the Bullet 350, tend to hold their value exceptionally well, provided they are maintained properly. After 10 years, a well-maintained Royal Enfield Bullet 350 (UCE model) could still fetch an approximate resale value of **₹50,000 to ₹90,000 INR**, depending heavily on its overall condition, mileage, service history, and current market demand for classic motorcycles. Some exceptionally well-preserved or custom models might even command higher prices.

Fig 6.1.7: Output (Details of the vechicle)

7. ADVANTAGES & DISADVANTAGES

7.1 Advantages of AutoSage Project

Advantage	Description
Automated Vehicle Analysis	Automatically analyzes vehicle images and generates detailed information, reducing the need for manual research.
Fast & Real-Time Results	Gemini Flash provides quick responses, allowing users to get vehicle details instantly.
User-Friendly Interface	The Streamlit-based web application allows users to upload images and view results easily without technical knowledge.
Multimodal AI Capability	Supports both image and text input, enabling accurate understanding of vehicle visuals and descriptions.
Scalability	The application can be extended to support more vehicle types, brands, and additional features in the future.
Data-Driven Decisions	Helps users make informed vehicle purchase and maintenance decisions based on structured AI-generated insights.
Cost & Time Saving	Eliminates the need to visit multiple websites or consult experts, saving time and effort for users.

Table 7.1: Advantages

7.2 Disadvantages / Limitations

Disadvantage	Description
Dependence on Image Quality	Low-quality or unclear vehicle images may reduce the accuracy of generated information.
Internet Dependency	The application requires an active internet connection to access the Gemini API.
API Usage Limits	Free-tier API usage is limited and may cause temporary restrictions under heavy usage.
No Physical Inspection	The system cannot detect hidden mechanical issues or internal vehicle defects.
Requires API Key Security	Improper handling of API keys may lead to security or quota issues.

Table 7.2: Disadvantages

8. CONCLUSION

The AutoSage project successfully demonstrates the use of Generative AI to analyze vehicle images and generate structured vehicle information using the Gemini Flash model. By integrating a user-friendly Streamlit interface, the system allows users to upload vehicle images and instantly receive details such as brand, model, features, mileage, price range, and resale value. This approach reduces manual research, saves time, and supports informed decision-making for vehicle buyers and owners. Overall, AutoSage highlights the practical application of AI in the automotive domain and provides a strong foundation for future enhancements.

9. FUTURE SCOPE

The **AutoSage** project can be further enhanced by integrating additional features and technologies. In the future, the application can support video-based vehicle analysis to provide more accurate insights. Integration with official vehicle databases can improve data accuracy and reliability. The system can also be extended to include a mobile application for wider accessibility. Additionally, multi-language support and cloud deployment can be implemented to improve scalability and reach a larger user base.

10. APPENDIX

10.1 Source Code

The complete source code for the **AutoSage** project is implemented using Python, Streamlit, and the Google Gemini API. The code includes modules for image upload, AI model integration, prompt handling, and output display.

10.2 GitHub & Project Demo Link

The source code and project demonstration video are hosted on GitHub. The repository contains all required files, including the application code, documentation, and setup instructions.

- **GitHub Repository:** https://github.com/skjaffareykareem/Autosage-using-gemini-flash.git

- **Project Demo Video:** *(Paste your demo video link here)*