

# **Industry Project Report On VehicleLens: The Most Comprehensive Vehicle Lookup Tool**

**Developed By: -**

Krishiv Patel (21162171036)

**Guided By:-**

Prof. Pritesh Andharia (Internal)

**Submitted to  
Faculty of Engineering and Technology  
Institute of Computer Technology  
Ganpat University**



**Ganpat  
University**

॥ विद्यया समाजोत्कर्षः ॥

**Institute of  
Computer  
Technology**



**Year - 2025**



## CERTIFICATE

This is to certify that the **Industry** Project work entitled “**VehicleLens: The Most Comprehensive Vehicle Lookup Tool**” by Krishiv Patel (Enrolment No. 21162171036) of Ganpat University, towards the partial fulfillment of requirements of the degree of Bachelor of Technology – Computer Science and Engineering, carried out by them in the CSE(CS) Department at ZeroRisk Labs Pvt. Ltd.. The findings presented in this project have not been submitted, either in whole or in part, to any other university or institute for the award of any degree or diploma.

Prof. Pritesh Andharia  
Internal Guide

Prof. Dharmesh Darji  
Head , CSE Department

**PLACE : ICT - GUNI**  
**DATE OF EXAMINATION : 10/05/2025**

## **ACKNOWLEDGEMENT**

Undertaking the IBM/Industry Internship project has been a golden opportunity for learning and self-development. I consider myself truly fortunate and honored to have received guidance and support from so many dedicated individuals throughout the course of this project. First and foremost, I would like to express my sincere gratitude to Dr. Rohit Patel, Principal, ICT, and Prof. Dharmesh Darji, Head of the ICT Department, for granting us the opportunity to carry out this project. I am deeply thankful to Prof. Pritesh Andharia (Internal Guide) and Mr. Juhef Ahmed (External Guide) for their constant support, invaluable guidance, and insightful feedback on our project, VehicleLens: The Most Comprehensive Vehicle Lookup Tool. Despite their demanding schedules, they always made time to mentor us and steer us in the right direction. This project would not have reached its current form without their help. I also extend my heartfelt thanks to the CSE Department for consistently monitoring our progress and providing all necessary resources and facilities, which made our journey smoother and more manageable. With deepest appreciation, I acknowledge the contribution of each individual who played a role in the successful completion of this project.

**KRISHIV PATEL (Enrollment No:21162171036)**

## **ABSTRACT**

VehicleLens is an innovative vehicle lookup platform designed to revolutionize access to comprehensive vehicular information. Unlike traditional tools that offer limited data, VehicleLens aggregates extensive details—from vehicle registration, insurance, fitness, and tax status to technical specifications and, where available, owner contact information such as full name, phone number, and address. This project report explores the platform's technical architecture, data integration methodologies, and the unique value proposition it offers to researchers, cybersecurity experts, and businesses. By leveraging real-time data sources and advanced aggregation techniques, VehicleLens addresses significant gaps in the existing vehicle information retrieval landscape. The report also examines the ethical, legal, and privacy implications of disseminating sensitive personal data, providing a balanced discussion on the responsibilities of data providers and the safeguards necessary to protect individual privacy. Ultimately, VehicleLens exemplifies a forward-thinking approach to data accessibility in the automotive sector, setting a new standard for transparency and utility while navigating the complexities of modern data ethics.

# INDEX

Certificate	i
Acknowledgement	ii
Abstract	iii
Table of Contents	iv
CHAPTER 1: INTRODUCTION	2
CHAPTER 2: PROJECT SCOPE	4
CHAPTER 3: SOFTWARE AND HARDWARE REQUIREMENTS	7
CHAPTER 4: IMPLEMENTATION DETAILS	10
CHAPTER 5: INTERFACE	15
CHAPTER 6: REFERENCES	23

## **CHAPTER: 1 INTRODUCTION**

## **CHAPTER 1 INTRODUCTION**

In today's modern digital era, immediate and detailed access to information is paramount across all sectors, including the automotive industry. Traditional vehicle lookup tools typically provide only basic registration and technical details, leaving a significant gap for users who require more comprehensive data. VehicleLens emerges as a pioneering solution designed to bridge this gap by offering an unprecedented level of detail—including vehicle registration specifics, insurance, fitness, tax status, technical specifications, and, when available, owner contact details such as full name, phone number, and address.

VehicleLens leverages advanced data integration techniques to consolidate information from multiple sources, ensuring that users receive a holistic and real-time snapshot of a vehicle's status and history. This innovative platform caters to a diverse audience ranging from researchers and cybersecurity professionals to businesses looking for reliable and detailed vehicle information. By pushing the boundaries of conventional vehicle lookup systems, VehicleLens not only enhances user convenience but also sets a new benchmark in the accessibility and transparency of vehicular data.

This report aims to provide a comprehensive analysis of VehicleLens, detailing its technical architecture, data acquisition methodologies, and the distinctive features that differentiate it from existing tools. Additionally, the report explores the broader implications of the platform, including the ethical, legal, and privacy considerations associated with handling sensitive personal information. Through this analysis, we seek to understand both the transformative potential of VehicleLens and the challenges that accompany its deployment in a data-driven world.

## **CHAPTER: 2 PROJECT SCOPE**



## CHAPTER 2 PROJECT SCOPE

VehicleLens aims to provide a one-stop platform for retrieving and displaying comprehensive vehicle information, including sensitive owner details when available. The scope of this project includes the end-to-end process of data collection, validation, integration, and presentation in a user-friendly interface. The following points outline the core areas covered by the project:

---

### 1. Data Acquisition & Aggregation

- **Source Identification:** Identify and integrate with reliable government, and third-party databases to fetch data such as RC status, insurance validity, fitness, tax, PUC, and technical specifications.
- **Data Mapping & Normalization:** Ensure that all data fields (e.g., engine number, chassis number, manufacturing date, owner name, address, etc.) are standardized and mapped accurately across different data sources.
- **Real-time Updates:** Implement mechanisms to refresh and validate data at regular intervals, ensuring that displayed information (e.g., policy expiration dates) is up-to-date.

### 2. Data Processing & Validation

- **Data Cleaning:** Establish protocols to filter out incomplete, inaccurate, or duplicated records before displaying them to the end user.
- **Quality Control:** Develop validation rules to detect inconsistencies (e.g., mismatched engine and chassis numbers, incorrect manufacturing dates) and flag or correct them where possible.
- **Error Handling:** Implement robust error-handling procedures to manage scenarios where data sources are temporarily unavailable or return invalid responses.

### 3. System Architecture & Implementation

- **Backend Services:**
  - **API Layer:** Create a secure, scalable API layer that interfaces with various external data sources and processes requests from the frontend.
- **Frontend Interface:**
  - **User-Friendly Search:** Provide a straightforward search bar where users can enter a vehicle registration number (e.g., "GJ01KU0001") and receive detailed results.
  - **Responsive Design:** Optimized interface supported various devices (desktop, mobile, tablet) to enhance accessibility and usability.

### 4. Key Functionalities

- **Vehicle Information Overview:** Display essential vehicle details—manufacturer, model, fuel type, color, vehicle class, body type, seating capacity, etc.
- **Status & Validity Checks:** Provide real-time information on RC status, insurance validity, fitness certificate, tax status, and PUC expiration.
- **Technical Specifications:** Show engine and chassis numbers, manufacturing date, cubic capacity, number of cylinders, and weight/dimensions.
- **Owner Details (If Available):**
  - **Personal Information:** Owner's name, father/husband's name.
  - **Contact Information:** Phone number and address.
- **Last Updated Timestamp:** Clearly indicate the most recent update time to inform users of data freshness.
- **Forensic Evidence:** Real captured vehicle images, Exact image captured location, Driver • Suspect • Accused person images.

### 5. Privacy & Ethical Considerations

- **Data Protection:**
  - **Encryption & Security:** Use strong encryption standards (such as SSL/TLS) to secure data during transmission, and apply encryption at rest when appropriate to ensure end-to-end data protection.
- **User Consent & Transparency:**
  - **Terms of Use & Privacy Policy:** Clearly outline the scope of data usage, storage, and

sharing.

- **Ethical Boundaries:**
  - **Responsible Data Handling:** Avoid misuse of sensitive personal information and ensure lawful usage aligned with the project's objectives.
- 6. **Testing, Quality Assurance & Documentation**
  - **Functional Testing:** Validate each feature (search, data retrieval, display of vehicle details, error handling) to ensure reliable performance.
  - **Performance & Load Testing:** Simulate high-traffic conditions to confirm the system can handle large volumes of requests without degradation in speed or reliability.
  - **Security Testing:** Conduct vulnerability assessments to identify and mitigate potential security risks, including unauthorized data access.
  - **User Documentation & Training:**
    - **User Guide:** Provide instructions on how to use the platform effectively.
    - **Developer Documentation:** Maintain clear technical documentation outlining data flow, API endpoints, and integration details.
- 7. **Project Deliverables**
  - **Functional Prototype:** A live website or application demonstrating real-time vehicle data retrieval and display.
  - **Documentation:**
    - **Technical Architecture:** Diagrams and explanations of system components and data flow.
    - **Integration & API References:** Detailed guides on how external systems and data sources are connected.
    - **User Manuals & Tutorials:** Step-by-step instructions for users and administrators.
  - **Compliance & Ethical Framework:** Policies and procedures addressing data privacy, user consent, and compliance with relevant regulations.
- 8. **Constraints & Assumptions**
  - **Data Availability:** Owner contact details (phone number, address) may not always be present or accessible due to regional laws and data provider limitations.
  - **Regulatory Compliance:** The project assumes that required legal permissions for data retrieval and display have been obtained or will be pursued.

---

By defining these parameters, the VehicleLens project establishes a clear and comprehensive roadmap for building and deploying an advanced vehicle lookup system. The scope ensures that all functional, technical, and ethical requirements are met, paving the way for a secure, reliable, and user-centric solution.

## **CHAPTER: 3 SOFTWARE AND HARDWARE REQUIREMENTS**

## CHAPTER 3 SOFTWARE AND HARDWARE REQUIREMENTS

### 1. Software Requirements

#### 1.1 Operating Systems

- **Development Environment:**
  - Windows 11 or later
  - macOS 10.14 (Mojave) or later
  - Linux (Ubuntu 18.04+ or similar)
- **Production Environment:**
  - Linux-based servers for backend deployments
  - Cloud-based environments (AWS, Vercel) for hosting and serverless functions

#### 1.2 Programming Languages & Frameworks

- **Frontend Development:**
  - **Vue.js and React.js:**
    - For building dynamic, responsive user interfaces
    - Use modern JavaScript (ES6+) along with HTML5 and CSS3
- **Backend Development & Serverless Architecture:**
  - **JavaScript (Node.js):**
    - Core language for both frontend and backend development
  - **API Development:**
    - RESTful services using Node.js frameworks (e.g., Express) deployed via Vercel Serverless Functions

#### 1.3 Cloud Services & Deployment Platforms

- **AWS Services:**
  - **AWS S3:**
    - For hosting static assets and files
  - **Additional AWS Services (as needed):**
    - AWS API Gateway for managing API endpoints
- **Vercel:**
  - For frontend deployment and hosting, enabling smooth CI/CD workflows and rapid deployments

#### 1.4 Database & Data Storage

- **In-Memory Caching (Optional):**
  - **Redis:** For caching frequently accessed data to boost performance

#### 1.5 Development & Collaboration Tools

- **Version Control:**
  - **Git:** Managed through platforms like GitHub, GitLab, or Bitbucket
- **Integrated Development Environments (IDEs):**
  - Visual Studio Code (recommended), WebStorm, or Sublime Text
- **CI/CD Pipelines:**
  - Using Vercel's built-in deployment system, GitHub Actions, or AWS CodePipeline for continuous integration and deployment
- **API Testing & Documentation:**
  - Tools like Postman or Insomnia, and Swagger/OpenAPI for API documentation

#### 1.6 Security & Monitoring

- **Security:**
  - SSL/TLS certificates for secure HTTPS communication
  - Use security middleware (e.g., Helmet for Node.js) to protect against common vulnerabilities
- **Monitoring & Logging:**
  - AWS CloudWatch, Datadog, or similar tools for performance monitoring and logging

---

### 2. Hardware Requirements

#### 2.1 Development Workstation

- **Processor:** Intel Core i3/i5/i7/i9 or higher
- **RAM:** Minimum 8 GB (16 GB recommended for handling multiple local development containers and IDEs)
- **Storage:**
  - 256 GB SSD or more for faster read/write operations and efficient local development
- **Graphics:**

- Standard integrated graphics are sufficient unless performing heavy graphics rendering

## 2.2 Server & Cloud Deployment Environment

### Entry-Level / Development Servers

- **Cloud Instances (for prototyping and testing):**
  - **CPU:** 2 vCPUs (e.g., AWS t3.medium or similar)
  - **RAM:** 8-16 GB
  - **Storage:** 100-150 GB SSD
  - Ideal for low-traffic, development, or staging environments

### Production-Level Servers (for moderate to high traffic)

- **Cloud-Based Production Environment:**
  - **CPU:** 4+ vCPUs (e.g., AWS t3.large or m5.large)
  - **RAM:** 8-16 GB
  - **Storage:** 100-200 GB SSD or scalable storage via AWS S3 and EBS
  - **Scaling & Load Balancing:**
    - Use AWS Auto Scaling and Elastic Load Balancing for managing traffic surges
    - Deploy via Vercel for the frontend to benefit from global CDN distribution

### High-Availability / Enterprise Deployments

- **Scalable Architecture:**
  - Container orchestration (e.g., Kubernetes on AWS EKS or Docker Swarm) for microservices
  - Multiple instances across various availability zones for redundancy and fault tolerance
  - Dedicated database instances with replication and read replicas for load distribution

## 2.3 Network & Connectivity

- **Internet Bandwidth:**
  - Adequate to support concurrent requests and data transfer, especially for serverless functions and content delivery
- **Security Appliances:**
  - Firewalls and DDoS protection integrated within your cloud provider (AWS Shield, Vercel's built-in security measures)

---

### Summary:

This setup leverages modern web development technologies (Vue, React, JavaScript) combined with a robust serverless backend and reliable cloud services (AWS S3, Vercel) to create a scalable, secure, and efficient vehicle lookup platform. The hardware requirements range from a capable local development environment to a fully scalable cloud deployment architecture, ensuring high performance and availability as the application grows.

## **CHAPTER: 4 IMPLEMENTATION DETAILS**

## CHAPTER 4 IMPLEMENTATION DETAIL

The implementation of VehicleLens is structured around several core components, each designed to ensure a robust, scalable, and secure platform for retrieving comprehensive vehicle information. Below is an in-depth look at how the various elements of the system are implemented:

---

### 1. Frontend Implementation

- **Frameworks & Libraries:**
    - **Vue.js & React.js:**
      - **Vue.js** is used for building certain dynamic components and interactive views, while **React.js** is leveraged for other parts of the UI, enabling modular and reusable components.
    - **JavaScript (ES6+), HTML5, and CSS3:**
      - These form the backbone of the application, ensuring compatibility and responsiveness across devices.
  - **User Interface & Experience:**
    - **Search Bar & Input Validation:**
      - Users input a vehicle registration number which is validated on the client-side before sending requests.
    - **Dynamic Data Rendering:**
      - The UI dynamically renders vehicle details, status updates, and owner information (when available) using component-based architecture.
    - **Responsive Design:**
      - The interface is designed to work seamlessly across desktops, tablets, and mobile devices.
  - **Deployment:**
    - **Vercel:**
      - The frontend is deployed on Vercel, which offers a global CDN, fast static file hosting, and easy integration with Git-based workflows for continuous deployment.
- 

### 2. Backend Implementation

- **Serverless Architecture:**
  - **API Gateway:**
    - AWS API Gateway provides a unified interface to trigger serverless functions., managing HTTP requests, routing them appropriately, and providing features like request throttling and caching.
- **Application Logic & Data Processing:**

- **API Endpoints:**
    - RESTful endpoints are designed for vehicle lookup operations, handling requests to retrieve registration, technical, insurance, fitness, and owner data.
  - **Data Validation & Sanitization:**
    - Incoming data from various sources is validated and sanitized to ensure consistency and security before being processed and stored.
  - **Integration with External Data Sources:**
    - **Data Fetching & Aggregation:**
      - Serverless functions are used to call external APIs (government, insurance databases, etc.) to fetch the latest vehicle data.
    - **ETL Processes:**
      - Lightweight Extract, Transform, Load (ETL) processes run within serverless functions to clean and normalize data from diverse sources.
- 

### 3. Data Storage & Management

- **Static File Storage:**
    - **AWS S3:**
      - Static assets, logs, and backups are stored in AWS S3, ensuring durability and easy retrieval.
  - **Caching:**
    - **Redis (Optional):**
      - In-memory caching using Redis is considered to improve response times for frequently accessed data.
- 

### 4. Deployment & Infrastructure

- **Cloud Infrastructure:**
  - **AWS Cloud Services:**
    - AWS services (API Gateway, S3, RDS/DynamoDB) form the backbone of the backend infrastructure.
  - **Vercel:**
    - The frontend is deployed on Vercel, benefiting from its streamlined CI/CD integration and global content delivery.
- **CI/CD Pipelines:**
  - **Git-Based Workflows:**
    - Code is maintained in a Git repository (GitHub/GitLab) with automated testing and deployment pipelines using Vercel for the frontend and AWS CodePipeline or GitHub Actions for backend services.
- **Scalability & Load Balancing:**
  - **Auto Scaling:**



- AWS automatically scales with incoming requests, and load balancing is managed through AWS API Gateway.
- **Monitoring & Logging:**
  - AWS CloudWatch is integrated to monitor performance, log errors, and track usage metrics.

Lam

## 5. Security

- **Data Security:**
  - **Encryption:**
    - All data in transit is secured via SSL/TLS. Sensitive data at rest is encrypted using AWS encryption services.
  - **Authentication & Authorization:**
    - If user authentication is required, JWT (JSON Web Tokens) or OAuth mechanisms are implemented to secure API endpoints.

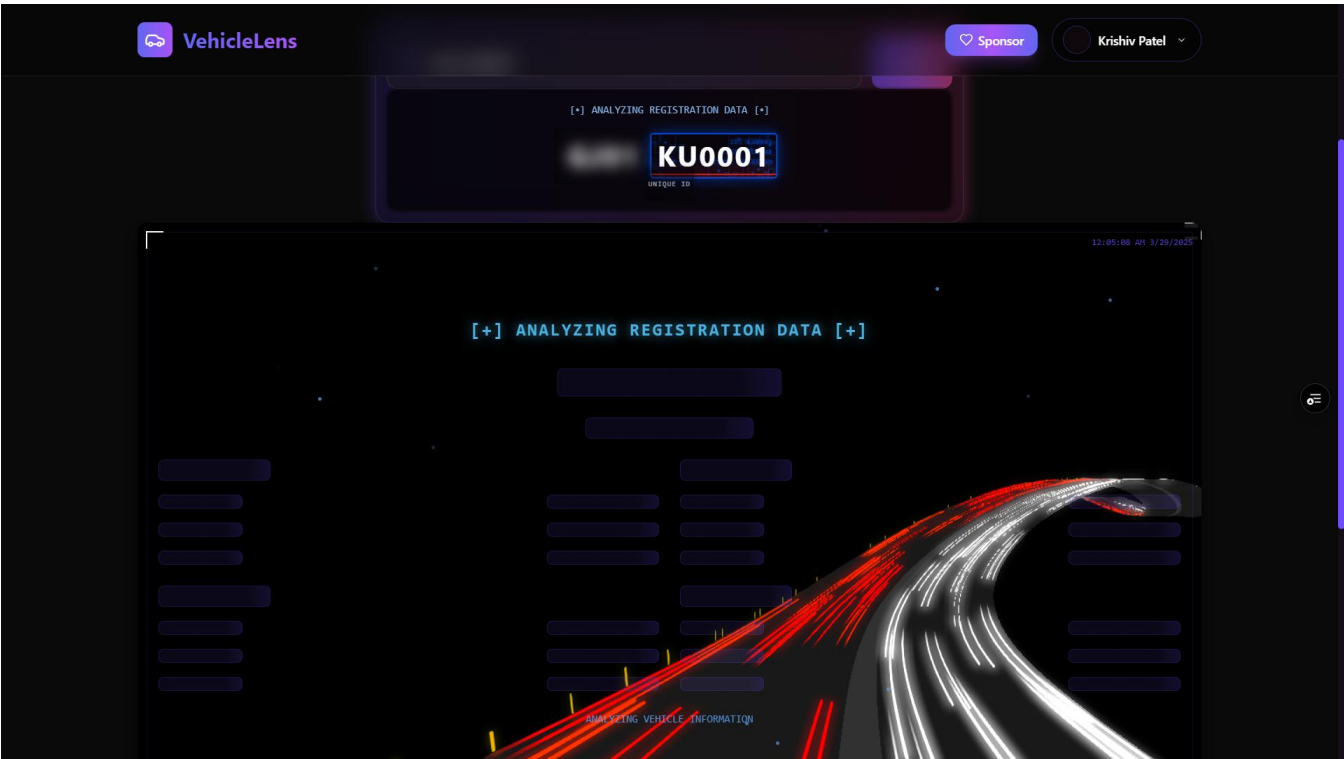
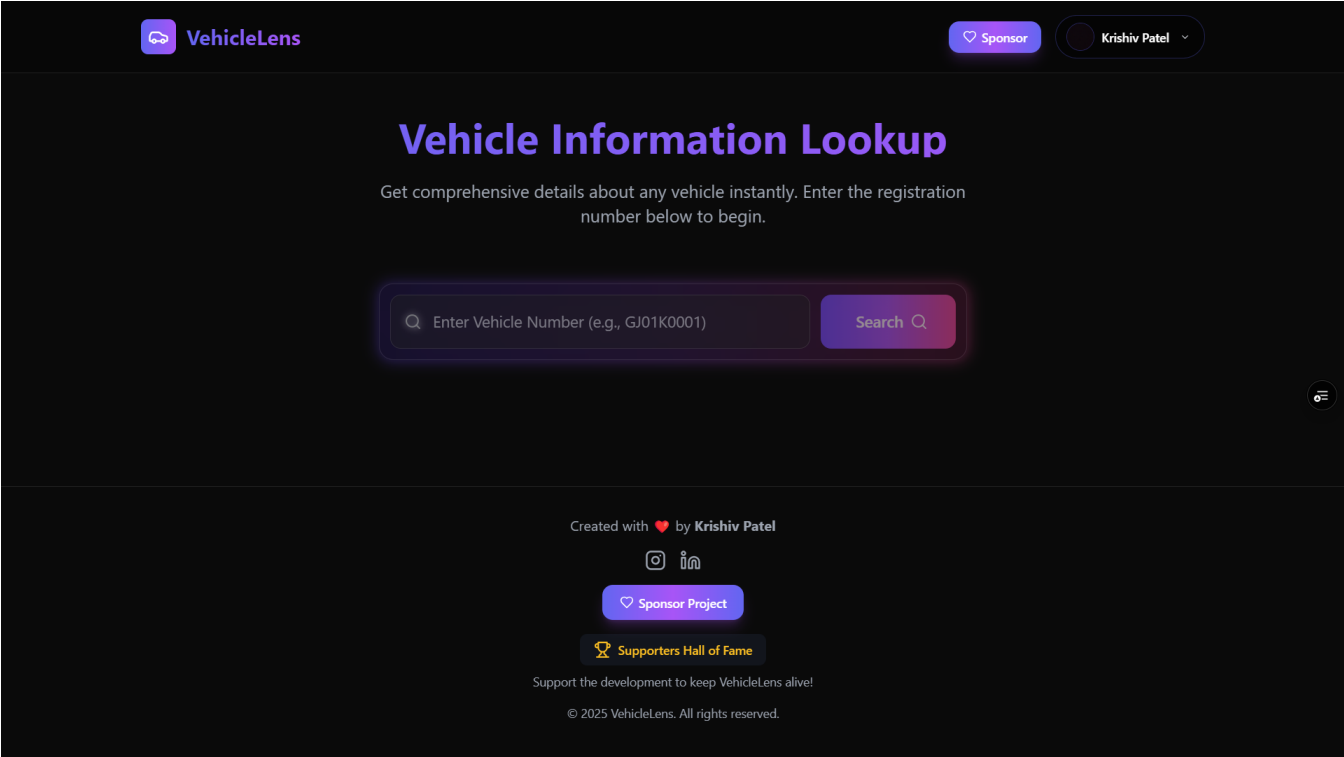
## 6. Error Handling, Testing & Documentation

- **Error Handling:**
  - **Resilient Code Design:**
    - Comprehensive error handling within serverless functions ensures graceful degradation and clear error messages.
  - **Retry Mechanisms:**
    - Built-in retries and fallback procedures are implemented to handle transient failures in data fetching or API calls.
- **Testing Strategy:**
  - **Unit & Integration Testing:**
    - Both frontend and backend components are rigorously tested using frameworks like Jest (for JavaScript) and Mocha/Chai (for Node.js).
  - **End-to-End Testing:**
    - Automated E2E testing tools (Cypress) are used to simulate real user interactions and validate the entire system flow.
- **Documentation:**
  - **Technical Documentation:**
    - Detailed documentation is maintained for API endpoints (using Swagger/OpenAPI), data flow diagrams, and architectural overviews.
  - **User Guides:**
    - Comprehensive user manuals and developer guides are provided to assist with deployment, maintenance, and future enhancements.

## Summary:

The implementation of VehicleLens leverages a hybrid frontend framework (Vue and React) alongside a serverless backend (AWS API Gateway) to provide a scalable, secure, and efficient platform. By integrating AWS and Vercel, the system benefits from modern CI/CD practices, global scalability, and robust data management, all while ensuring strict adherence to security and privacy standards.

## **CHAPTER: 5 INTERFACE**



VehicleLens

Sponsor

Krishiv Patel

Vehicle Information

Registration Number: GJ01KU0001

RC Status

Active

Insurance

Valid

Fitness


Valid

Tax

Paid

PUC

Valid



Registration Details

Registration Date

2019-01-24

RTO

RAJKOT, Gujarat

Status

ACTIVE

Status As On

2025-03-22

NOC Status

Not Issued

Basic Vehicle Details

Manufacturer

PORSCHE AG GERMANY

Insurance & Validity

Insurance Company

HDFC ERGO General Insurance Company Ltd

VehicleLens

Sponsor

Krishiv Patel

Vehicle Information

Registration Number: GJ01KU0015

RC Status

Active

Vehicle Age


6Y 5M 6D

Owner Info

First Owner

Price Info

₹47,80,000



Registration Details

Registration Date

21/12/2018

RTO

AHMEDABAD

Registration Location

AHMEDABAD

Status

ACTIVE

RC Status As On

29/03/2025

Owner Serial Number

1

Basic Vehicle Details

Manufacturer

BMW INDIA PVT LTD

Insurance & Validity

Insurance Company

HDFC ERGO General Insurance Company Ltd

Sponsor

Krishiv Patel

Basic Vehicle Details

Manufacturer

BMW INDIA PVT LTD

Model

3 SERIES 320D GRAN TURISMO 1995 CC

Fuel Type

DIESEL

Color

IMPERIAL BLUE BRILLI

Vehicle Class

LMV

Seating Capacity

5

Technical Specifications

Engine Number

0295Y192

Chassis Number

WBA8T370XJBN97307

Manufacturing Year

12/2018

Manufacturing Month

9

Purchase Date

21/12/2018

Cubic Capacity

1995.0 cc

Ex-Showroom Price

₹47,80,000

Insurance Company

HDFC ERGO General Insurance Company Ltd

Policy Number

2302205119721702000

Previous Insurer

HDFC ERGO General Insurance Company Ltd

Previous Policy Number

2302205119721702000

Owner Details

Owner Name

PAULOMI

Present Address

7 AASOPALAV BUNGLOWS,NR THALTEJ  
GRAM PANCHAYAT  
THALTEJ,AHMEDABAD -380059

Sponsor

Krishiv Patel

Traffic Challans

Pending

0

Disposed

1

Payable

0

Total Amount

₹2,000

GJ4160604240806050280

CH-00031296418

06 Aug 2024

9 months ago

Disposed

Paid

₹2,000

Offences

Driving at excessive speed - LMV/અતિશય ઝડપે વાહન ચલાવવું - એલવએમવી

Challan Details

Challan ID:

CH-00031296418

State:

Gujarat (GJ)

Status:

Disposed

Challan State:

Offline

Provider Type:

E CHALLAN

Court Case Details

Challan No:

GJ4160604240806050280

Court Status

Received Date:

Page 18

VehicleLens

Sponsor

Krishiv Patel

Court Case Details

Challan No:  
GJ4160604240806050280

Challan Date:  
Aug 6, 2024, 01:39:10 PM

Name:  
PAULOMI

Place of Offence:  
tapovnit3, tapovan sarcal amdavad city Gujarat, Navrangpura, 00246

District:  
Ahmadabad

Circle:  
G-Traffic(Airport, Krushna Nagar, Meghani Nagar, Naroda, Saradar Nagar)

Court Status

Received Date:  
07-11-2024

Verification Status:  
Challan is yet to be verified in the Virtual Court

Allocation Status:  
Challan is yet to be allocated in the Virtual Court.

Proceeding Status:  
Proceedings of Challan is yet to be completed.

Offline Challan

You may see some Challan as unpaid even if it is paid at police station in offline mode.

FastTag Information

HDFC Bank FastTag

Status: Active

Balance

₹0

Unknown

FastTag Details

VehicleLens

Sponsor

Krishiv Patel

RAJ08501-19072900334

Pay Now

Pending

Unpaid - Payable

₹100

FastTag Information

ICICI Bank FastTag

Status: Active

Balance

₹3817

Last updated today

FastTag Details

Owner Name

Umang Garg

Bank

ICICI Bank

Status

Active

Call Bank ( 18002100104 )

Call

Last Updated


2025-03-29 00:05

Page 19

VehicleLens

SponsorKrishiv Patel

FastTag Information



HDFC Bank FastTag

Status: Active

Balance

₹0

Unknown

FastTag Details

Owner Name

Paulomi

Bank

HDFC Bank

Status

Active

Call Bank ( 18002100104 )

Call

FORENSIC EVIDENCE

TARGET VEHICLE CAPTURE

REF: GJ01KU0015

CASE #641490

TIMESTAMP: 05/04/2025

VehicleLens

SponsorKrishiv Patel

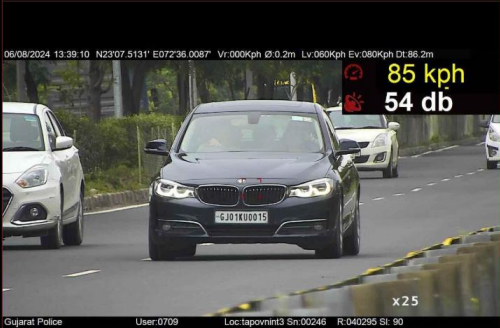
TARGET VEHICLE CAPTURE

REF: GJ01KU0015

CASE #641490

TIMESTAMP: 05/04/2025

06/08/2024 13:39:10 N23°07'5131" E072°36'0087" Vr:000Kph Ø:0.2m Lv:060Kph Ev:080Kph Dt:86.2m



85 kph

54 db

x25

Gujarat Police

User:0709

Loc:lapovint3 Sn:00248 R:040295 Sl: 90

EVIDENCE TYPE: VEHICULAR | HIGH-RESOLUTION CAPTURE #1

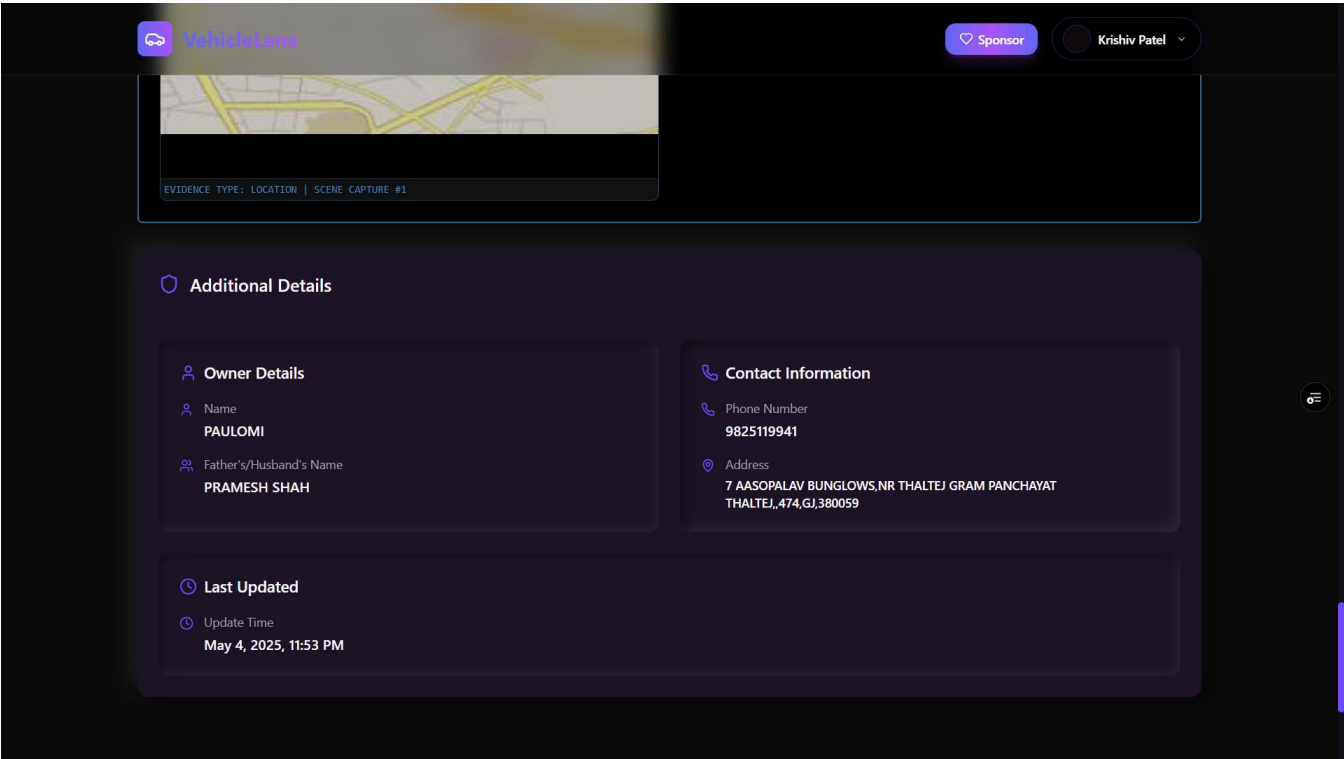
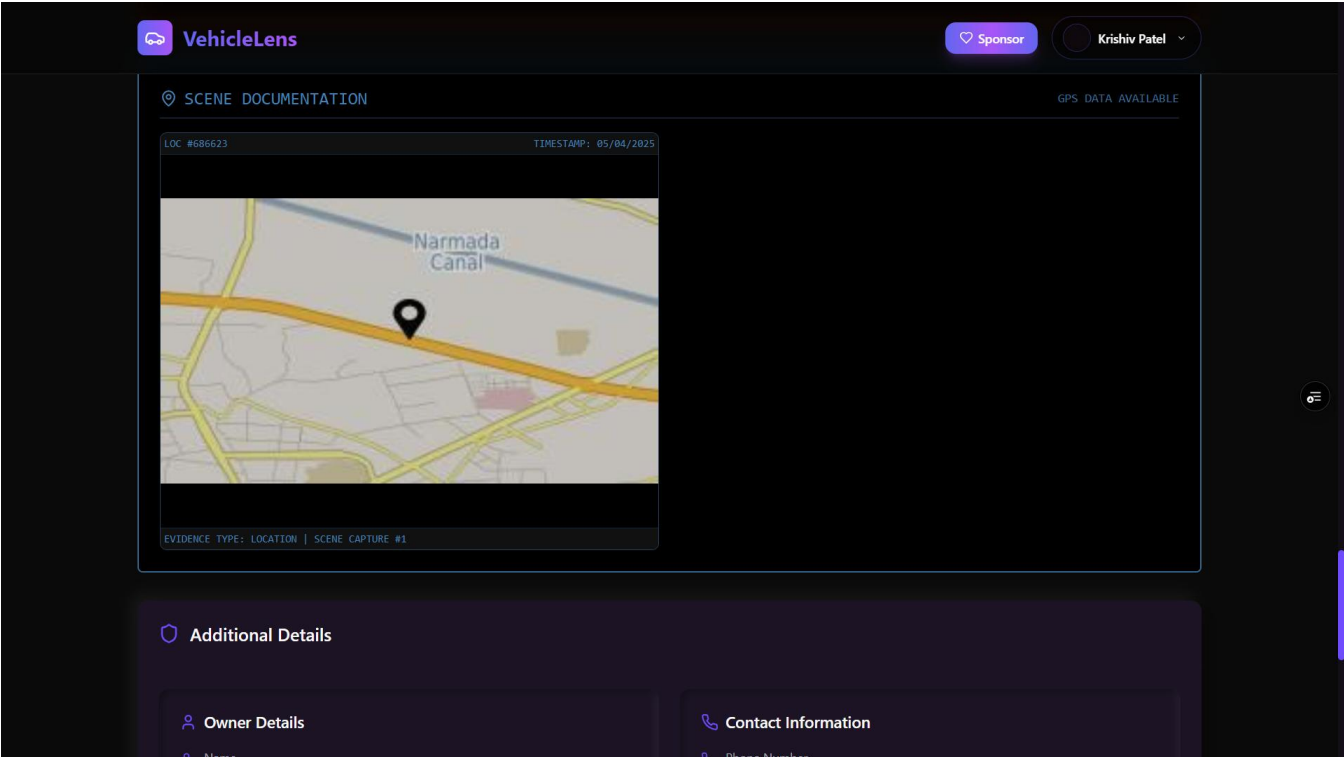
SCENE DOCUMENTATION

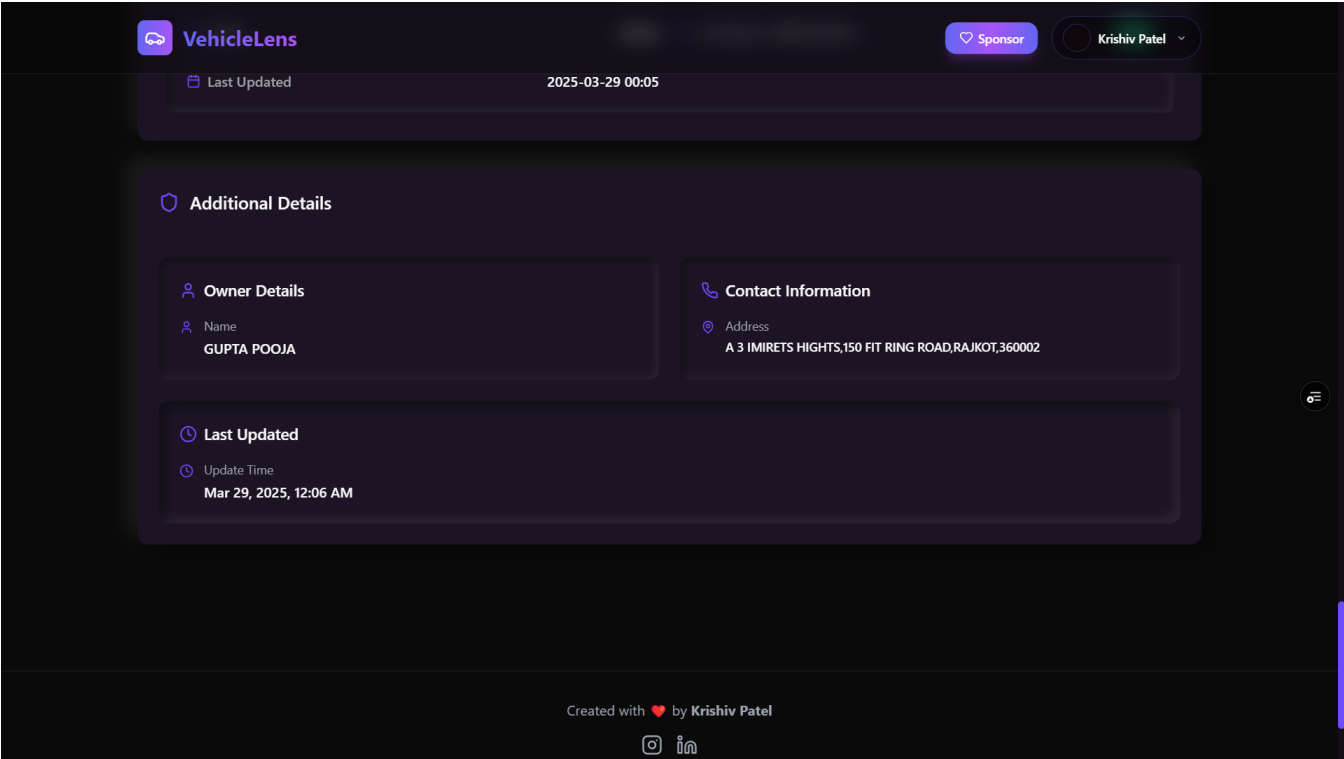
GPS DATA AVAILABLE

CASE #606623

TIMESTAMP: 05/04/2025







## **CHAPTER 6 REFERENCES**

## CHAPTER 6 REFERENCES

1. VehicleLens Website Link: <https://vehiclelens.vercel.app/>
2. POC Demo YouTube Video: <https://youtube.com/shorts/Pj3dBm-ISAQ>
3. Vercel Documentation.  
Vercel. Retrieved from <https://vercel.com/docs>