

Drive Ledger: a MERN-Powered Vehicle Management Solution

Dr. Daneshwari Mulimani¹, Navya Uttareshwar²

¹Assistant Professor Department of M.Sc.(CS), BLDEA's S.B. Arts and K.C.P.Sci college
Vijayapur.Krnataka,

²M. Sc (CS) Studen Department of M.Sc.(CS), BLDEA's S.B. Arts and K.C.P.Sci college
Vijayapur.Krnataka t

Abstract—The rapid growth of urbanization and vehicle ownership has led to an increased need for efficient, streamlined Drive Ledger Model (DLM). This paper explores the development of Drive Ledger, a modern, full-stack vehicle management solution powered by the MERN stack (MongoDB, Express.js, React.js, Node.js). DLM integrates various vehicle management functionalities, such as vehicle tracking, maintenance scheduling, user profiles, and real-time data analytics, fuel usage monitoring along with log maintenance and optimal vehicle related expenses providing a centralized platform for fleet operators, owners, and service providers. The methodology provides the detailed performance analysis of one-to-one vehicle and respective driver. So that enables smooth handling of the agent along with risks in fleet management process.

Index Terms—JWT, RESTful API, Node.js & Express.js, Git & GitHub, MongoDB, React.js.

- MongoDB: A NoSQL database designed for high performance and scalability.
- Express.js: A web application framework for Node.js that simplifies backend development.
- React.js: A JavaScript library for building interactive user interfaces.
- Node.js: A JavaScript runtime that allows backend code to be executed outside the browser.

The MERN stack allows developers to write both frontend and backend code in JavaScript, leading to faster development and easier maintenance. The combination of these technologies provides a high degree of flexibility and scalability, making it ideal for building dynamic applications like Drive Ledger Model (DLM).

I INTRODUCTION

As transportation technologies evolve, the complexity of vehicle management increases, requiring robust, scalable, and user-friendly solutions. The advent of cloud-based systems and the MERN stack offers significant opportunities to build such systems, providing scalability, efficiency, and ease of development.

Drive Ledger utilizes the MERN stack, which stands for MongoDB, Express.js, React.js, and Node.js, to create a unified web application capable of managing vehicle-related tasks in real-time. This system allows users to manage vehicle data, track real-time locations, schedule services, and analyze performance metrics efficiently.

The MERN stack is a combination of four powerful JavaScript technologies:

II EXISTING SYSTEM AND CHALLENGES

Through the literature study, Traditional vehicle management systems, however, often lack real-time data synchronization, fail to scale effectively, and require significant infrastructure investments. Furthermore, many systems are not designed to integrate the vast array of services that modern vehicles require, such as integration with IoT devices and third-party APIs.

Proposed system has motivated by the various challenges and user requirements in the aim of data management and integration, scalability, majorly in user authentication and security, data analytics and Reporting raised the major issue to handle in huge content. The DLM is an enhanced adversarial system approaches to numerous current issues.

III METHODOLOGY AND IMPLEMENTATION

A. High-Level Architecture

The architecture of the Drive Ledger system is divided into three main layers:

1. Frontend (Client-Side): Built using React.js, the frontend is a dynamic, responsive user interface that allows users to interact with the system.
2. Backend (Server-Side): Express.js and Node.js form the backend, handling business logic, API endpoints, and user requests.
3. Database Layer: MongoDB is used to store vehicle data, user profiles, maintenance logs, and real-time GPS information.

B. Components:

- User Dashboard: A React-based interface that provides a detailed overview of vehicle information and operations.
- Admin Dashboard: A backend management panel for administrators to oversee vehicle status, reports, and user management.
- Real-Time GPS Tracker: Integrated into the system using third-party GPS API services to track the real-time location of vehicles.
- Maintenance Scheduler: Users can input maintenance tasks and the system will provide automated alerts based on pre-set time intervals or mileage limits.

C. Data Flow

1. Vehicle Registration: When a user registers a vehicle, the details (e.g., model, make, year, VIN) are stored in MongoDB.
2. Maintenance Tracking: The backend keeps track of scheduled services and past service records. When a service is due, the system notifies the user.
3. Real-Time Data: GPS data is fetched periodically from external services and updated in the MongoDB database.
4. User Interaction: All actions are performed through the React-based frontend. User inputs (such as service scheduling or mileage tracking) are processed by the backend via API calls, and the data is reflected in real-time

The detailed methodology has shown in the figure long with working model with respect to the tools and their performance

D. Security Considerations

DLM employs several security mechanisms:

- JWT (JSON Web Tokens): For user authentication and session management.
- HTTPS: All communication is encrypted using SSL/TLS.
- Data Validation and Sanitization: To prevent injection attacks and ensure the integrity of the data stored in MongoDB.
- Role-Based Access Control (RBAC): Different user roles (e.g., admin, fleet manager, individual user) are assigned specific permissions.

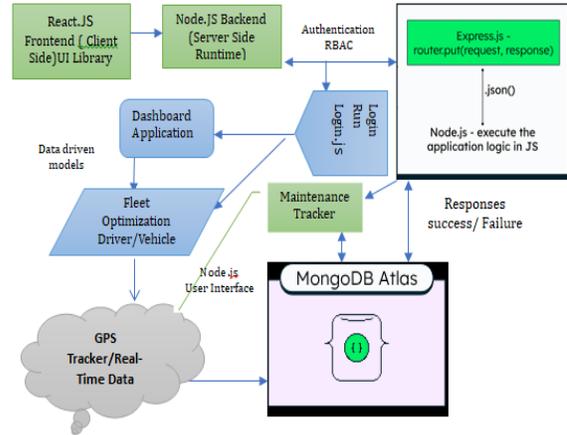


Figure 1: Methodology on Drive Ledger Model (DLM) Using MERN Stack

IV. PROPOSED SYSTEM ENACTMENT:

Proposed model provides version control, collaboration, and continuous integration/deployment by using Git & GitHub. Postman used as an API testing tool that allows developers to send, test, and debug RESTful API requests efficiently.

- Fleets Optimization: The need for real-time insights to manage both personal and commercial multiple vehicles efficiently.
- Lack of state of art updates, comprehensive maintenance tracking, and driver behavior analytics are handled. Inefficiencies in communication between vehicle owners, operators, and service providers.
- Efficiency & Automation – Reduces manual effort by automating trip logging, fuel tracking, and maintenance scheduling.
- Cost Savings – Helps users monitor expenses, analyze fuel efficiency, and reduce unnecessary vehicle costs.
- Integration with Third-Party Services: DLM supports integration with external services such as

weather APIs, traffic condition APIs, and vehicle insurance systems,

- Data Analytics: Reports on fuel efficiency, mileage, and other critical metrics.

Along with these DLM has following features

- Vehicle Tracking:** DLM allows users to track the real-time location of their vehicles using GPS. The system integrates with third-party GPS APIs to provide updates every few seconds. This feature is particularly useful for fleet managers to monitor vehicle locations and optimize routes in real-time.
- Maintenance and Service Scheduling:** The system supports the creation of maintenance schedules based on predefined intervals (e.g., mileage or time) or triggered by the vehicle's status. Notifications are sent to users when a vehicle requires maintenance, and users can book services directly through the platform.
- Driver Behavior Analytics:** The system tracks key driving metrics, such as speed, fuel consumption, and braking patterns. By analyzing this data, DLM helps identify inefficient driving behaviors and suggests improvements, potentially reducing maintenance costs and improving fuel efficiency.
- Reporting and Analytics:** Comprehensive analytics dashboards offer insights into fleet performance, including: Vehicle utilization and downtime. Fuel consumption and efficiency, Maintenance and repair costs Vehicle history and service records.

These insights assist fleet managers in making data-driven decisions for maintenance, replacements, and driver training.

V PERFORMANCE EVALUATION

A. Load Testing: Load testing was performed to evaluate the system's performance under different user loads. The system

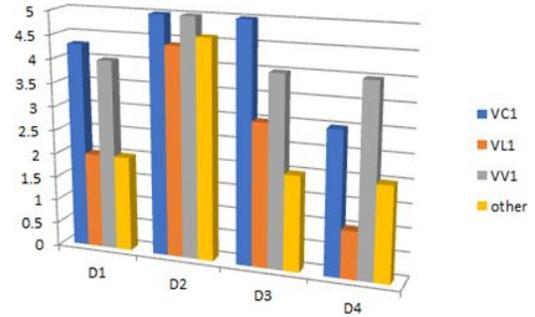


Figure 2: Performance of Each Driver With respect to the Different Vehicles

demonstrated good scalability, with response times under

200ms for API calls even with 100+ concurrent users. Figure 2 shows the registered driver performance with respect different vehicles (Car, Large-Vehicle, Van) they give service with. The data is fetched by the cloud system on their dynamic state along with mapping between customer feedbacks. With this we can analyze the driver performance for future reference, interpret and smooth handling the owner/customer relationship.

B. User Feedback: Beta testing with a group of fleet operators and vehicle owners provided positive feedback, with users highlighting the ease of use, real-time tracking, and maintenance scheduling features as the system's key strengths. We can set different measures for performance tracking so that it provides the stake-holders to track each vehicle status, reduce maintenance tractability and current performance analysis with future prediction. Figure 3 represents the performance of Car on different measures. Key performance measures can be Average speed, Number of incidents/accidents, Fuel efficiency, On-time delivery Rate, Customer satisfaction score. These can better use to analyze and performance tracking for better fleet management process.

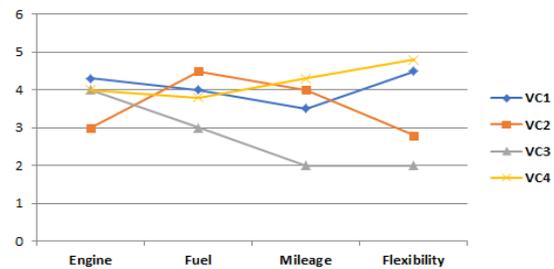


Figure 3: Individual Vehicle Performance with respect to Engine, Fuel, and Mileage And it's Travelling Flexibility

VI CONCLUSION AND FUTURE WORK

DLM offers a powerful, MERN-based solution for vehicle management, combining the strengths system's real-time data updates, maintenance tracking, and analytics features make it an invaluable tool for fleet management and individual vehicle owners. However, future enhancements could include: 1. Docker & Kubernetes for containerization and scalable deployment of the application. 2. GraphQL: An advanced API query language for more efficient and flexible data fetching.

By leveraging the flexibility and scalability of the MERN stack, DLM provides a foundation for developing a modern, cloud-based vehicle management platform.

And Analytical Reviews · August 2023, <https://www.researchgate.net/publication/386515863>

- [8] Prof Yogesh Kadam et al, "Introduction to MERN Stack & Comparison with Previous Technologies", *European Chemical Bulletin* 12(Special Issue 4):14382-14386, June 2023.
- [9] Sourabh Malewade, Archana Ekbote, "Performance Optimization using MERN stack on Web Application", *International Journal of Engineering Research & Technology (IJERT)* Vol. 10 Issue 06, June-2021, ISSN: 2278-018, <http://www.ijert.org> IJERTV10IS060239 www.ijert.org.

REFERENCES

- [1] Chamalla Sravani et al, "Constructing a Study Buddy Using MERN (MongoDB, Express.js, React, Node.js) Stack Technologies" *Eng. Proc.* 2024, 66, 27. <https://doi.org/10.3390/engproc2024066027>, <https://www.mdpi.com/journal/engproc>
- [2] V. S. Thokala, "A Comparative Study of Data Integrity and Redundancy in Distributed Databases for Web Applications," *IJRAR*, vol. 8, no. 4, pp. 383–389, 2021.
- [3] S. Raju, S. Soundararajan, and V. Loganathan, "MERN Stack Web Application" *R.S.C.B.*, vol. 25, no. 6, pp. 6325–6332, 2021.
- [4] M. Nabeel, B. Ali, and A. Hamdan, "Real-time feedback on consumer's behavior: Literature review", *Int. J. Energy Econ. Policy*, 2019, doi:10.32479/ijeep.8353.
- [5] M. Bawane, "A Review on Technologies used in MERN stack", *Int. J. Res. Appl. Sci. Eng. Technol.*, 2022, doi: 10.22214/ijraset.2022.39868.
- [6] Vhandale, S. Gandhak, S. Karhale, S. Prasad, and P.S. A. Bachwani, "MERN Stack: Technologies Used for Web Development", *Int. J. Res. Appl. Sci. Eng. Technol.*, 2022, doi: 10.22214/ijraset.2022.40247.
- [7] Vasudhar Sai Thokala, "Enhancing User Experience with Dynamic Forms and Real-time Feedback in Web Applications Using MERN and Rails", *International Journal Of Research*