School of Science and Engineering

Integrating Teltonika for Real-Time Fleet Management and Driver Monitoring

Capstone Design Report

FALL 2024

Ali Lazraq

Supervised by:

Dr. Amine Abouaomar

Student conduct and copyright

The supervisor and the student (the Capstoner) agree that:

1. Permission has been obtained for any third party content (eg Data, illustrations, photographs, charts or maps).

2. The results described in this report have not previously been published

Student's name and signature:

Ali Lazraq:

[illegible]

Supervisor's name and signature:

Dr. Amine Abouaomar:

[illegible]

Integrating Teltonika for Real-Time Fleet Management and Driver Monitoring

Capstone Report

Student Statement:

I, Ali Lazraq, have applied ethics to the design process and in the selection of the final proposed design. And I have held the safety of the public to be paramount and have addressed this in the presented design wherever may be applicable.

s

Ali Lazraq

Approved by the Supervisor

Dr. Amine Abouaomar

ACKNOWLEDGEMENTS

I would like to thank all those who have given me very strong support in this project journey from beginning to end. First, I would like to express my profound gratitude for everything to my parents, whose love, encouragement, and sacrifices have been the anchor of all my academic success. Their support kept me held strong.

My appreciation extends to my heart for my supervisor, Dr. Amine Abouaomar, for the invaluable guidance, constructive feedback, and unconditional support. His mentorship has largely influenced the direction and success of this project.

Sincere appreciation touches my heart also to my friends who have not remained at any given time without uttering words of encouragement or showing signs of positivity for this challenging yet wonderful journey. I must appreciate El Mehdi Ziate , Haytam Mostaine, Amine Elarabi, El Mehdi El Asli, and Hanie Boukmakh for their insightful opinions, guidance, and moral support, which I believe helped keep me focused and motivated.

This very great accomplishment wouldn't have been possible without the touch of all these fantastic people, and for that, I am most grateful to them for having touched my life.

CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS AND ABBREVIATIONS

AVL - Advanced Vehicle Location

CAN - Controller Area Network

CSV - Comma-Separated Values

ERD - Entity-Relationship Diagram

GPS - Global Positioning System

GPRS - General Packet Radio Service

HTTP - Hypertext Transfer Protocol

IoT - Internet of Things

JWT - JSON Web Token

MAD - Moroccan Dirham

MVC - Model-View-Controller

SQL - Structured Query Language

TCP/IP - Transmission Control Protocol/Internet Protocol

URL - Uniform Resource Locator

VIN - Vehicle Identification Number

3G - Third Generation Mobile Telecommunications Technology

ABSTRACT (ENGLISH)

This project presents a comprehensive solution for vehicle tracking and fleet management by integrating IoT technology with an intuitive dashboard system. The Teltonika FM5300 device was configured to collect real-time data, including GPS locations, speed, fuel logs, and maintenance metrics, which were securely transmitted via GPRS/3G to a centralized server. The backend, developed with Spring Boot, efficiently processed and stored this data in a MySQL database, while the frontend, built using React and React Native, provided user-friendly interfaces tailored for administrators and drivers.

The system features live GPS tracking, geofencing alerts, maintenance tracking, and fuel log visualizations, offering actionable insights to improve operational efficiency. Additionally, a price prediction tool was implemented to estimate the resale value of vehicles, leveraging external APIs for real-time currency conversion and market data. Secure user authentication was achieved using JWT-based authentication, ensuring reliability and scalability.

This project significantly enhances fleet management by optimizing operations, reducing costs, and improving decision-making. Future enhancements, such as transitioning to a microservices architecture, integrating predictive maintenance with machine learning, and expanding driver-centric features, will further increase the system’s capabilities and adaptability. This solution demonstrates the potential of IoT in revolutionizing transportation management for industries reliant on efficient fleet operations.

RESUME (FRENCH)

Ce projet propose une solution complète pour le suivi des véhicules et la gestion de flotte en intégrant la technologie IoT à un système de tableau de bord intuitif. Le dispositif Teltonika FM5300 a été configuré pour collecter des données en temps réel, notamment la localisation GPS, la vitesse, les journaux de carburant et les indicateurs de maintenance, transmises en toute sécurité via GPRS/3G à un serveur centralisé. Le backend, développé avec Spring Boot, a permis de traiter et de stocker efficacement ces données dans une base de données MySQL, tandis que le frontend, conçu avec React et React Native, offre des interfaces conviviales adaptées aux administrateurs et aux conducteurs.

Le système inclut des fonctionnalités telles que le suivi GPS en temps réel, les alertes de géorepérage, le suivi de maintenance, la visualisation des journaux de carburant, ainsi qu'un outil de prédiction des prix pour estimer la valeur de revente des véhicules. L'authentification sécurisée des utilisateurs a été assurée via des jetons JWT, garantissant fiabilité et évolutivité.

Ce projet améliore considérablement la gestion de flotte en optimisant les opérations, en réduisant les coûts et en renforçant la prise de décision. Les améliorations futures, comme l'adoption d'une architecture microservices, l'intégration de la maintenance prédictive via le machine learning et l'expansion des fonctionnalités destinées aux conducteurs, renforceront davantage les capacités et l'adaptabilité du système.

Mots clés:

Suivi des véhicules - Gestion de flotte - Technologie IoT - Tableau de bord intuitif -

Maintenance prédictive - Architecture microservices.

1 Introduction

The integration of IoT in vehicle tracking and fleet management is changing the way institutions track their operations and optimize them. The project addresses the real-time vehicle tracking and managing challenges through the Teltonika FM5300 IoT device with a well-thought-out dashboard application. Consequently, the decision-making of both vehicle administrators and drivers can be done through the most actionable insights, including mechanisms such as GPS tracking and data collection from CAN bus systems, along with modern visualization means.

The prime objective of the project is to construct a system capable of collecting, processing, and subsequently making available the information on a user-friendly dashboard. The system's functionality will include but not be limited to the following- real-time GPS tracking, fuel and maintenance management, geofencing alerts, and historical data visualization. This will not only enhance the efficiency of operations but also promote safety, reduce costs, and improve adherence to the maintenance schedule.

In terms of importance, IoT modular vehicle tracking systems are the backbone throughout the logistics, transportation, and vehicle rentals industries, where fleet management is key to the growth of the business. The project brings together hardware configuration, backend engineering, and frontend development to demonstrate how the Internet of Things can be integrated into real-life applications.

It means that this serves as a high-level overview of the project as far as modern fleet management goes, and as a result provides an appropriate precursor for the technical, operational, and functional insights presented here.

2 Problem Statement:

Efficient vehicle tracking and fleet management are critical in several industries including logistics, transport, and vehicle rentals. However, traditional systems are not highly effective to meet the demands for precision, revamping, and real-time capabilities for ensuring efficiency in operations, reducing costs, and improving safety. These problems are hardly being captured and visualized properly, like fuel consumption, maintenance status, and vehicle location, for actioning in an integrated way.

Basically, a system is absent that could easily blend real-time data transmission and comprehensive analysis with user-friendly visualizations. Current solutions mostly fail to offer reliable geofencing alerts, proactive maintenance tracking, and customized dashboards according to different user requirements like admin or driver. Furthermore, there are the gaps of data accuracy and usability that have been created due to the lack of synchronization of IoT devices, backend processing, and frontend visualization.

Besides that, scalability difficulty is experienced in current systems when handling numerous vehicles in diverse-nation regions. Such systems almost always flounder in dealing with dynamic fleet operations, including monitoring of driving behaviors, route optimization, and vehicle compliance.

The Teltonika FM5300 IoT will be integrated with a dashboard application that will collect and process real-time data besides visualizing it for effective decision-making. Therefore, the project aims to resolve these problems into a robust, scalable, and user-friendly solution for modern fleet management.

3 Project Specifications:

Integrating real-time data visualization and fleet management challenges, this project includes IoT-based vehicle tracking with an advanced dashboard system. Its two major components are Real-Time Data through Teltonika FM5300 IoT-equipped devices and subsequently, an integrated dashboard application for data process visualization and decision-making. Together, both components strive towards further simplification of the vehicle monitoring system while enhancing operational efficiency.

3.1 System Components

Teltonika FM5300 IoT device is an intelligent device capturing relevant vehicle data such as GPS location coordinates, speed, etc., along with data from the engine's CAN bus. At the same time, it transmits this to the backend server through the mobile GPRS network 3G in real time. The device has a communication towards a static IP address, offering a reliable passage of data for processing and storage. This is a strong interface for visualization and decision-making over the data with the IoT device. It uses a MySQL database to structure data, Spring Boot as its backend, and React for front end. The dashboard thus converts raw data into actionable intelligence. It allows the roles of the administrator and driver and their roles have been customized according to their own responsibilities.

3.2 Features Overview

Presently, the sensor data collection system of which Teltonika FM5300 becomes an integral part. The critical parameters captured would include location from GPS, reason, and consumption, and maintenance checks, among others. Data are thus encoded into hexadecimal, giving it a smaller form factor that is easier to transmit via a GPRS network. After transmission, data processing begins. The incoming data are then converted through the use of a gateway based on Python into a more user-friendly interpretation. This will then be stored into a MySQL database for a longer store and for quicker retrieval for subsequently further analysis and visualization.

The dashboard is certainly data visualization; it has interactive tools that improve fleet management. Real-time vehicle tracking is shown on an interactive map based on Leaflet libraries while bar and line charts by Material-UI and Nivo are provided to give insights to fuel logs, distance traveled, and levels of maintenance. It also has a feature in maintaining that makes use of visual progress bars to show due or overdue tasks to keep fleet managers informed about the health of vehicles.

An even more important feature of the setup geofencing, that is, defining geographic boundaries using latitude, longitude, and radius values to the administrator. It means that if a vehicle goes beyond these mentioned boundaries, an alert will be raised from the system and will show an alert message on the geofencing page with important information such as time and location of the breach.

The system also defines role-specific views for the administrator and the driver. Admin dashboard features include vehicle selection, geofence management, real-time GPS tracking, and thorough maintenance tracking. For drivers, it contains a simplified mobile application used to log fuel transactions, ensuring reliable entry of data.

3.3 Technical Specifications

Spring Boot is the framework that has been used to develop the backend system based on the Model-View-Controller (MVC) design pattern. This paradigm is effective by distinguishing patterns into different domains, thus improving the maintainability and scalability of the software. Therefore, a secured channel between the backend and the frontend can be set up via user authentication based on JWT tokens.

The front end is React-based for web applications and React Native for the mobile counterpart. Hence, each user interface is user-friendly and features modes of operation that could be offered in dark or light, depending on the user's preferences. The frontend application can be implemented in conjunction with the backend to keep a smooth interaction for the responsive, intuitive experience.

To manage data efficiently, this database system is based on MySQL. Through it, data coming from the Teltonika FM5300 device will be classified and stored in an organized manner for rapid visualization and analysis.

This transmits data using GPRS technology over 3G networks. Data is sent in compact hexadecimal format, and then converted into TCP/IP for reliable communication between the device and server. This guarantees the most accurate data transfer in a safe manner, and is a requirement in real-time operations.

4 Steeple Analysis

Social

This system promotes safety and efficiency in a fleet operation by providing real-time vehicle tracking and timely maintenance alerts that lead to driver accountability and transparency in operation. It promotes informed decision-making, thus boosting trust and collaboration on the part of stakeholders.

Technology

This project integrates the use of IoTs, GPRS communication, and modern frameworks such as Spring Boot and React to showcase the prowess of technology in collecting, processing, and visualizing data and providing a true scalable and reliable approach for future extensions.

Economic

Operational costs fall due to better fuel performance, proactive maintenance tracking, and vehicle downtime minimization. Its scalability allows better long-term savings for businesses that manage large fleets, making it an economically viable system.

Environmental

The project encourage green procedures by route optimization which gets reductions in fuel consumption and reduces unnecessary trips based on better planning which goes a long way in promoting sustainable fleet operations that reduce carbon footprints.

Legal

The project fulfills the data security standard requirements by enabling the security through JWT-based authentication and transmission encryption of data for the privacy of user activities according to standards outlined legally in IoT and data management.

5 Engineering Standards:

5.1 ISO 26262 – Functional Safety for Road Vehicles

This standard ensures the safety of automotive electronics. It helps in designing systems that won't cause harm if they fail. Which means making sure that the data collection and transmission from the vehicle are safe and reliable, preventing any issues that could affect vehicle operation.

5.2 IEEE 802.15.4 – Low-Rate Wireless Personal Area Networks

Low-power, low-bandwidth communication, similar to the data transmission challenges in IoT systems. The standard helps ensure that data transmission is efficient and reliable, which is crucial for handling data from the telematics device.

6 LOGIC MODEL FRAMEWORK

6.1 Target Population

An organization relying on vehicle-based operations falls under the primary target population for this work. Such companies can be found in various sectors - logistics, delivery services, or among transportation providers - where tracking a vehicle and data stream in real-time helps them ensure smooth operations. Also, implementing the IoT-based Teltonika FM5300 device with a dashboard application can allow these companies to smoothen the management of their fleet; make it cost-effective, and better decision-making through data analysis. Rental companies form another segment of users of this system because such enterprises often find it difficult tracking their vehicles, putting up scheduled maintenance, monitoring fuel use-all essential in ensuring that the fleet remains fit for the needs of customers. Such a dashboard can monitor such aspects as status and location of vehicles, send alerts on maintenance needs, and produce reports on fuels consumed and distances traveled by rental companies. Data from those vehicles can be used by the administrator for different analysis such as geofencing alerts, maintenance schedules, and real-time GPS tracking. This will allow companies to optimize their fleet operations and longevity and efficiency for the vehicles within. This is especially important as the project adapts to different business needs, making it a suitable tool for any company willing to enhance its capabilities on fleet management.

6.2 Underlying Assumptions

The most important aspect of this project is that the clarity of real-time data collected from the Teltonika FM5300 device is assumed to improve fleet management through actionable insights to be provided for administrators or fleet managers. Companies are supposed to empower their decisions toward improved operational efficiency, vehicle safety, and reduced overall operating costs through a user-friendly dashboard integrated with advanced IoT technology.

Real-time vehicle tracking provides an opportunity for route optimization that will reduce fuel consumption due to less engine running time and vehicle downtime. It ensures geofencing-the option of operating vehicles only within predefined geofenced areas-and prevents unauthorized usage. Besides this, logistical planning also improves on the basis of real-time location tracking. Maintenance alerts and trends in fuel consumption are expected to assist administrators in the proactive maintenance of their vehicles, thus increasing the lifespan of the fleet and avoiding costly repairs.

Also, by monitoring vehicle performance and the maintenance schedule, the system assumes the presence of data-driven strategies in the organization to manage its assets. It will include data visualization - fuel logs, distance tracking, and maintenance progress indicators - which are expected to smoothly integrate all these data, making it simple for administrators to interpret and act upon large datasets about problems and solutions. These assumptions provide the theoretical framework for designing and developing the project in line with improving fleet management practices.

6.3 Resources/Challenges

Several technical resources are indeed required for the success of this project, at the forefront of which would be the Teltonika FM5300 devices to act as IoT sensors for taking real-time data. Another thing will be a joined implementation of a last connection that will ensure the data transmission by 3G GPRS mobile network is effective. The dashboard developed in React at the front end and Spring Boot at the back end is also important data visualization and user interaction components. Similarly, a MySQL database is essential to store and retrieve the data coming from the vehicles for analysis and tracking in a detailed manner. Trained personnel would, however, be needed for the installation and configuration of the devices and maintenance to ensure consistent performance.

Of these certain challenges is faced by the project. One of the critical issues is data security during transmission, so no attempts can be made to access it illegally. The integrity for data accuracy and reliability must be enforced on the system to avoid differences which can affect decision-making. Continuous monitoring would also create more discomfort in the users due to privacy concerns hence the need for transparency and compliance with regular regulations on personal data. Scalability is yet another major challenge; as the fleet expands, it must increase the capacity of the system to accommodate more devices, more data points, and users. Probably most likely to be constrained by upfront purchasing costs for hardware and network subscriptions, as well as ongoing operational costs. Therefore, ensuring effective allocation of resources and budget is crucial to making the system cost-effective and scalable for larger fleets. This is one of the key challenges associated with achieving the objective of this project, which is providing a seamless and reliable fleet management solution.

6.4 Activities

These were adequately planned to achieve the goal of the project. The first of these activities entailed fitting and configuring the Teltonika FM5300 on the vehicles. The whole process involves setting the devices to acquire data from sensors, GPS, and CAN bus and thus getting it to transmit data in real time over the 3G GPRS mobile network onto a server. Special care was taken to ensure that the devices were well linked to each vehicle and calibrated for more accurate data collection.

The other activity of utmost importance was in the establishment of real-time tracking of data. This set up the whole system for learning on the location of vehicles, performance parameters such as speed and fuel consumption, and states of maintenance; protocols on data transmission were set for reliability and security-including, but not limited to, encoding data into hybrid hexadecimal format for efficient transmission and decoding on the server side. Building user-friendly dashboards was one of the project's critical outputs. Thus, the development of a responsive web application under the umbrella of React in the frontend and Spring for backend to integrate onto the dashboard. The Dashboard contains valuable information on fuel logs, maintenance status, kilometers traveled, and live GPS. More administrative features include geofence alerts and price prediction functionalities. Regular accuracy checks on the information were made for the validation of data sent by these devices. Security measures were including securing measures for data transmission purposes, which cover confidentiality and integrity assurance, such as encryption of data and JWT token-based secure user authentication. Setting thresholds and alerts were also done for certain maintenance needs and driving behaviors such as breach of geofence or overdue engine oil change. These are critical in enabling proactively manage and ensure a system designed to cater for the diverse needs of its users. The backbone of this project thus incorporates everything into one single entry and serves this purpose of providing full fleet management solution.

6.5 Outputs of the Project

Several tangible results from the project ensure the intense solution provided for real-time fleet management. The output of this results is an uninterrupted GPS tracking system giving live data on the location, routing, and performance of vehicles. Integrated with a user-friendly dashboard, this tracking system makes it possible for administrators to follow vehicle positions and movements over an interactive map interface, improving operational visibility.

Speed, fuel consumption, readings from the odometer, and maintenance status are also summarized in the dashboard. Along with the speed and fuel consumption, maintenance status is also summarized in the dashboard. Bar charts visualize fuel logs and transaction histories. Line charts show the distance traveled by fleet managers for analyzing performance trends and decision making based on data. Maintenance alerts and geofence breach notifications are also available to help facilitate active vehicle management that keeps vehicles maintained at optimum and reduces potential costs of repairs or delays.

The system also produces compliance and maintenance records. It accounts for driver behavior towards the vehicle and how the vehicle is used. It gives data into the habits of the drivers, for example, speed patterns and geofencing compliance, which can play an instrumental role in improving driver safety, optimizing routes, and ensuring compliance with operational policies. The provision of a price prediction module has further increased the functionality of the system through its analysis of the resale value of vehicles for better financial decisions by companies on their fleet. Therefore, this collective output will empower enterprises with increased efficiency and reduced operational costs and provide better utilization of assets.

6.6 Outcomes

Expectedly, the project will churn out several instantly and longer-term results in fleet management with sustainable solutions. In addition, direct, real-time tracking of vehicles visibility and safety of drivers through the monitoring of their location against dangerous driving behavior and alerts on geofence intrusions and maintenance will bring immeasurable long-term benefits. In this way, timely corrective measures are enforced, preventing accidents and misused vehicles. These analytics and monitoring features of the dashboard usher in operational efficiencies in the intermediate term.

Scheduling of maintenance is being looked at proactively to minimize unforeseen vehicle downtime in addition to repair costs for the organization. Fuel consumption analysis facilitates routing better for less fuel consumption and cost savings. Detailed records of usages enable administrators to refine fleet operations for adequate resource utilization.

Ultimately, the project aims at fostering a culture of safety and compliance in organizations. The dashboard will allow fleet managers to base their tactical decision-making prowess on data. It will help us understand the requirement for continuous improvement. Fuel efficiency and ideal vehicle maintenance make up a part of an environmental-friendly fleet operation taking sustainable approaches. So, the project appropriately meets the urgent calls of organizations into operational efficiency, cost savings, and sustainability for the long term, thereby making it one of the relevant components in modern fleet management.

7 Literature Review

Having viewed IoT technology in vehicle tracking systems, it reminds that this field is so promising for change and improved activities in fleet management. Real-time data are collected from IoT systems on cloud computing and analytics to bring real-time insights into action for vehicle performance, driver behavior, and operations efficiency analytics. Studies show IO systems can improve route optimization significantly, reduce fuel consumption, and increase safety by monitoring driver behavior contrary to speeding or harsh braking techniques.

It, however, has systems in the market much like that shown by GPS tracking providers, the consideration being just limited functions like finding out where the assets are and fuel use monitoring. Although such devices come with highly critical functionalities, they lack a very strong combination of many others such as maintenance tracking, geofencing alerts, or even dedicated dashboards customized to a specific user role. Besides, most existing systems cannot scale to make the systems inappropriate in such situations of management where many fleets are in the same company but with different operational needs.

That, therefore, would be the vision to bridge the gaps defined: establish real-time data transmission with advanced visualization tools for user-specific interfaces between administrators and drivers. With that combination employing the IoT capabilities of Teltonika FM5300 and customized dashboard solution, one would make an even stronger, scalable, and friendly alternative to the traditional systems. Other additional features such as geofencing, maintenance alerts, or fuel transaction logging all serve to future-proof the system even further.

8 Methodology & Capstone Design

It goes without saying that this section will explain in much detail everything throughout design, methodology, and technologies used to develop the project. It will cover the configuration of the Teltonika FM5300 device, construction of a secure and scalable backend, development of an interactive frontend, and finally comprehensive visualizations.

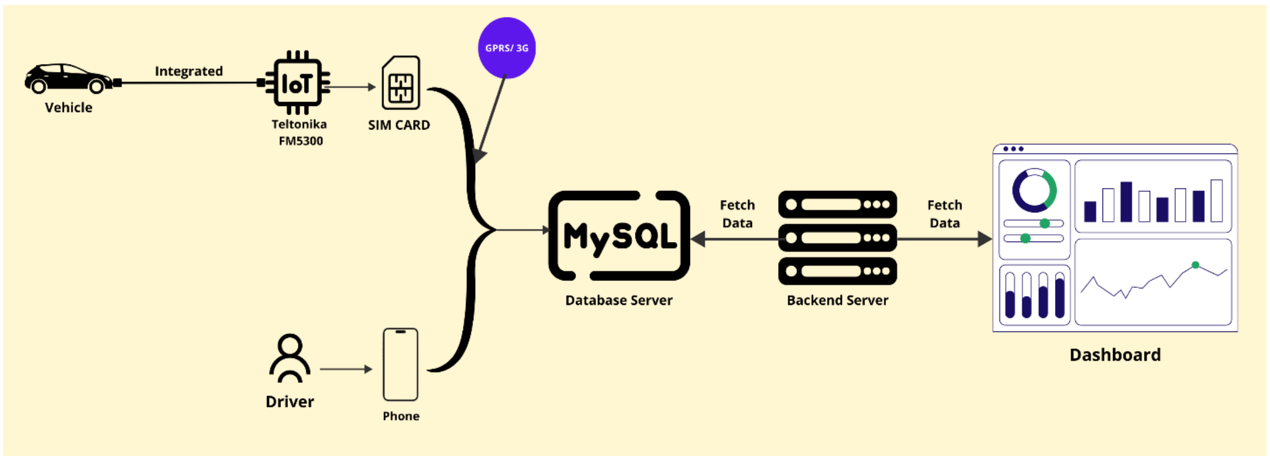


Figure 1: General architecture

8.1 Device Configuration: Teltonika FM5300 and Data Flow

The Teltonika FM5300 device is specifically developed for this project. This is to collect real-time data from vehicles and store it under secured networks for further analysis. The process is achieved by the configuration of the device so that it collected information through sensors, GPS, and the CAN bus. All this data is compactly encoded in hex to enhance transmission load but not change the accuracy of the measurements. The encoded data will travel through the GPRS mobile network using 3G protocol and will be converted into TCP/IP packets for further processing.

On the server side, the data are converted back to the binary format using a Python-based gateway for ensuring that they are in usable formats. The converted data are then stored into MySQL Database, carefully structured for its efficient retrieval and exposure.

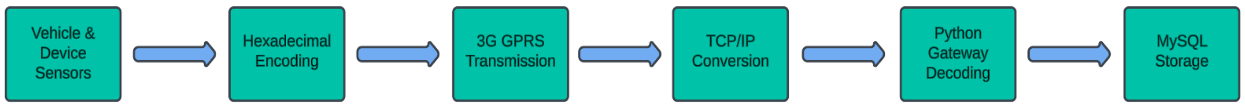


Figure 2: Data Flow Process

8.2 Technology Stack

The entire project capitalizes on a new-age technology stack to encapsulate the entire scenario leading to a reliable and secure mode of data processing and communication along with easy human interface design. The technology components consist of backend development through Spring Boot and frontend development through React and React Native.

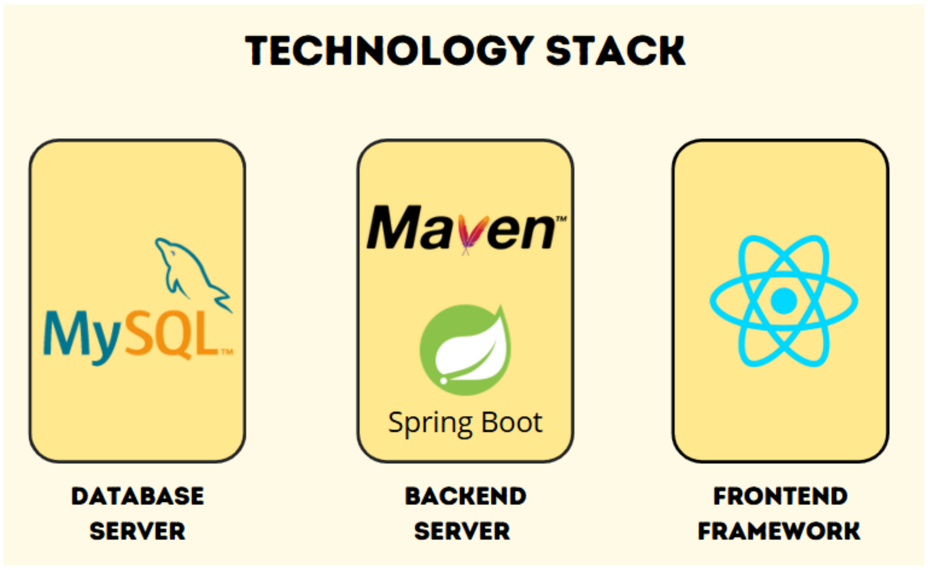


Figure 3: Technology Stack

8.2.1 Backend Development with Spring Boot

The backend implementation of the project is purely using Spring Boot with MVC architecture. This type of architecture will keep distinct responsibilities and hence, modular and maintainable systems. The Models represent the domain entities in the database; Controllers capture HTTP requests; and Services handle the core operations of the business logic.

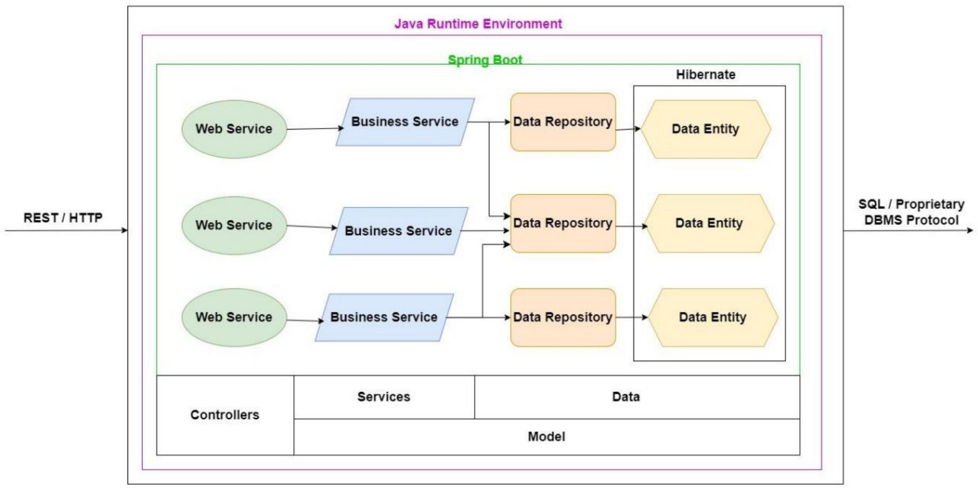


Figure 4: Backend MVC architecture

To secure the web application, it uses the syntax of JSON web tokens (JWT) for authentication and authorization purposes. Instead of maintaining session states in memory, JWT tokens are generated when a user successfully logs into the app (and carry subsequent requests using the same token). Moreover, it does prove secure during the communication between frontend and backend and for securing sensitive data.
An API has been developed in order to manage any interactions required between frontend and backend, from authentication to data retrieval and storage. All these APIs created a real-time synchronization between the dashboard and the mobile app which is connected to the database. Besides documentation and testing of the APIs performed through Postman, this tool aids in building a properly defined collection of endpoints with clear instructions on how to use them, including the expected request formats and responses. The Postman Collection comprises of all critical commands necessary for data fetching and manipulation via the API.
***Postman Collection Link*:** Postman API Documentation

8.2.2 Login and Signup

It allows entering the system with login and signup by ensuring secure and easy access by the end-user. This is present in the backend using Spring Boot because this allows it to manage authentication and authorization through JSON Web Tokens.

The signup takes personal information from the user using their credentials, then it is hashed and saved into the MySQL database. This ensures that data is intact. It protects sensitive information against unauthorized access. The entered data would be cross-checked with existing records in the database during the login stage. If successfully validated, a JWT token will be generated, and it will be sent back to the client.

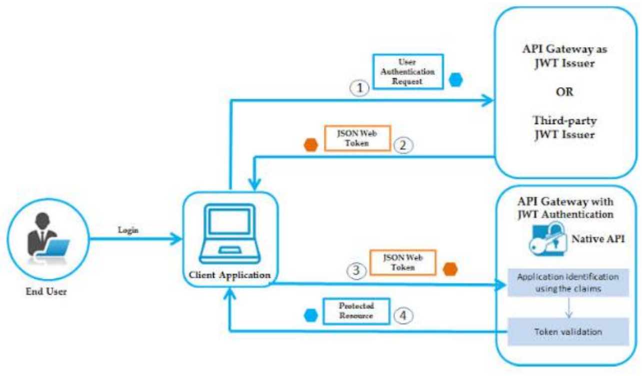


Figure 5: JWT Representation

It plays the role of continuity in communication between the frontend and backend systems. It is very much secured inside the headers of requests made to any APIs within the backend, allowing the server to validate whom to provide access to any protected resources with regards to the user. Thereby not bearing the overhead of storing sessions on the server side, scalability isn't much of an issue related to security.

The whole login and signup system depends on where the user is accessing the platform (through website or mobile app), making distinctions between an administrator and a driver. Within this scheme, users can access features and data based on their roles.

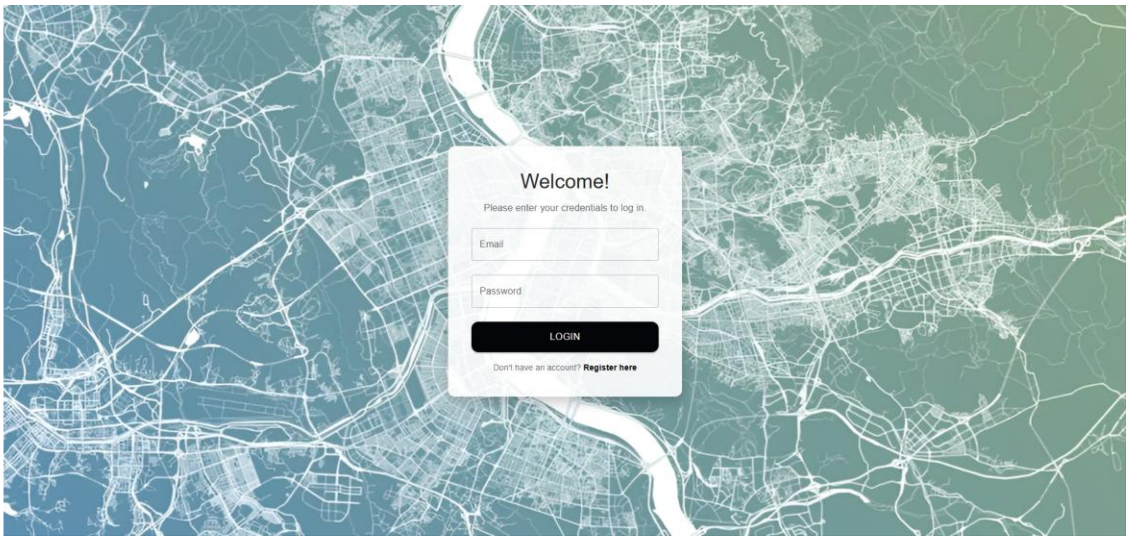

Welcome!
Please enter your credentials to log in.
Email
Password
LOGIN
Don't have an account? Register here

Figure 6: Login for Admin's view

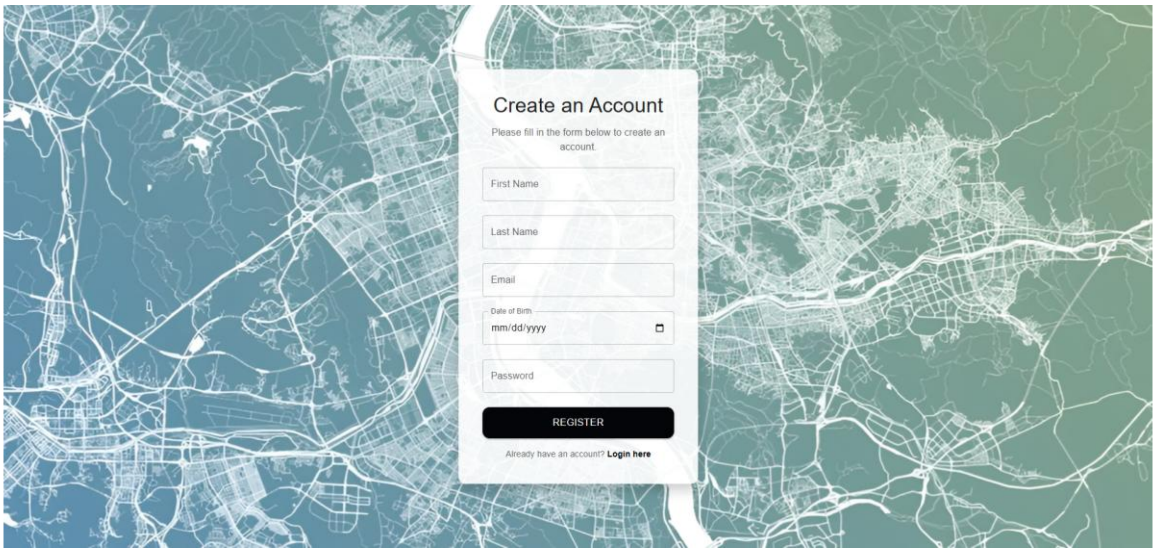

Create an Account
Please fill in the form below to create an account.
First Name
Last Name
Email
Date of Birth
mm/dd/yyyy
Password
REGISTER
Already have an account? Login here

Figure 7: Registration Admin's view

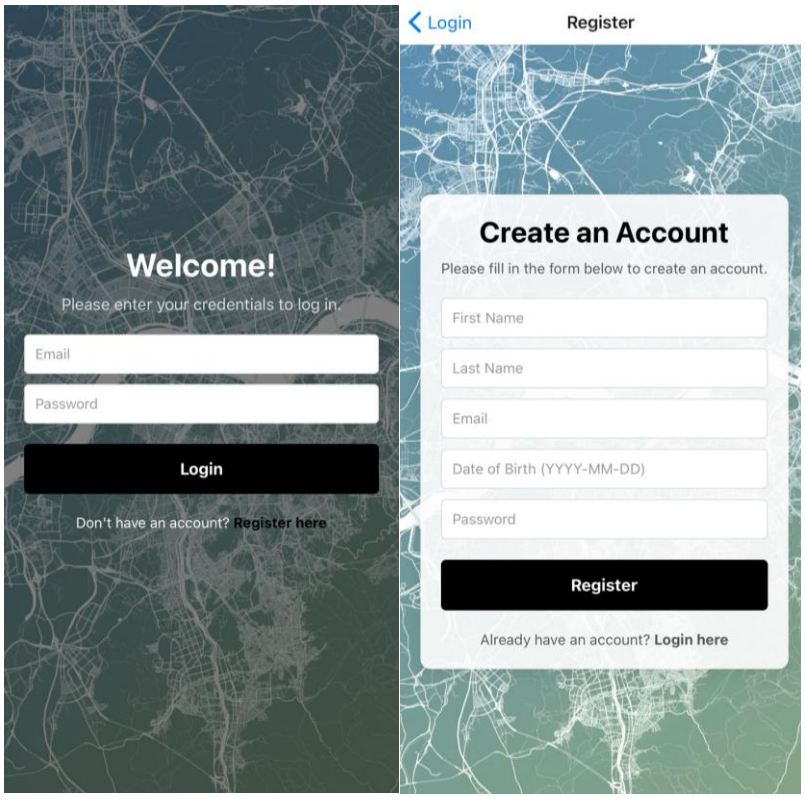

Welcome!
Please enter your credentials to log in.
Email
Password
Login
Don't have an account? Register here
Login
Register
Create an Account
Please fill in the form below to create an account.
First Name
Last Name
Email
Date of Birth (YYYY-MM-DD)
Password
Register
Already have an account? Login here

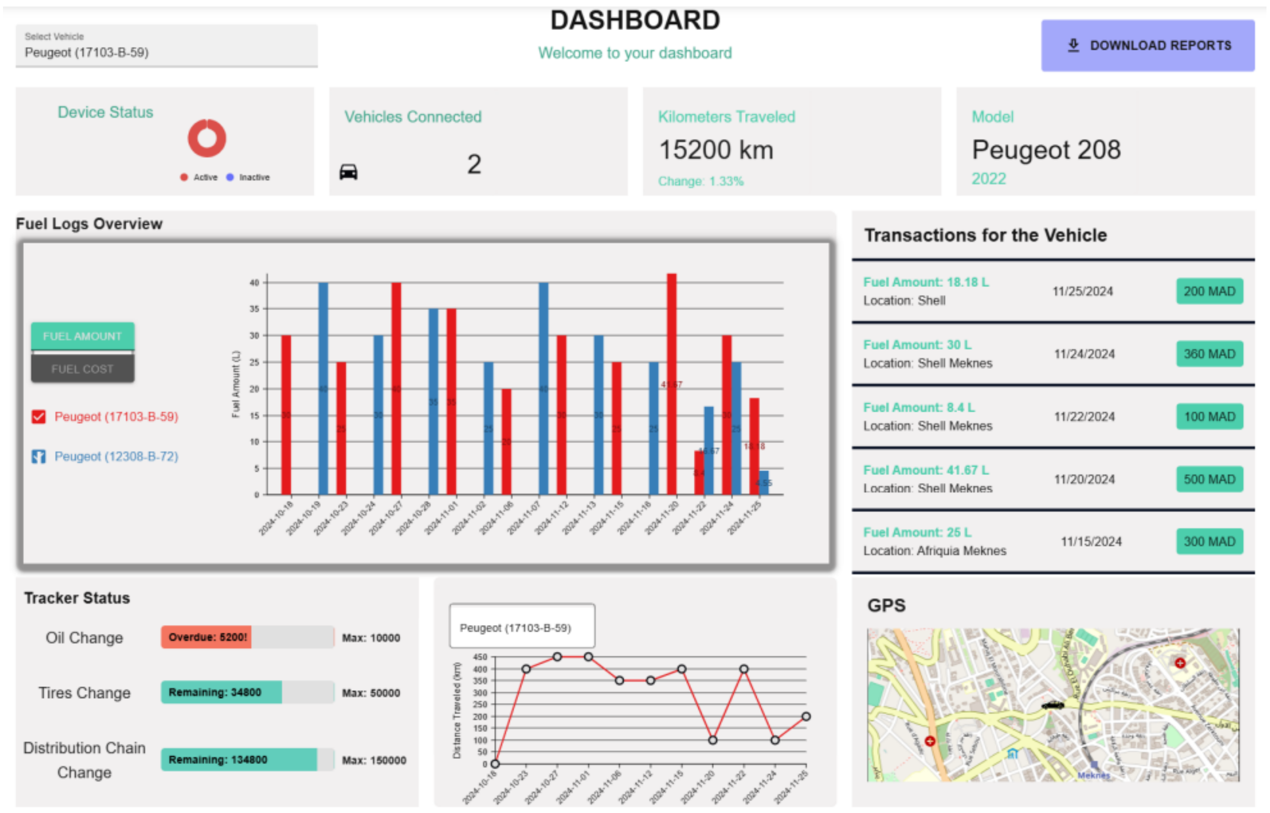
Figure 8: Login and Registration for Driver's view

8.2.3 Frontend Development with React

The frontend is developed using React to create an interactive and user-friendly interface. Two views were implemented to cater to different user roles: the admin dashboard and the driver's mobile application.

The admin dashboard includes all features such as an option to choose the vehicle, real-time GPS tracking, visualization of fuel logs, alerting on maintenance dates, and geofence management. The design brings in a modern look with light and dark modes for better use.

The drivers' application is designed with the help of React Native, which emphasizes easier entry of fuel logs. Here the drivers can choose the vehicles along with details of transaction and ensure real-time updating of these to the system.


Select Vehicle
Peugeot (17103-B-59)

DASHBOARD

Welcome to your dashboard

DOWNLOAD REPORTS

Device Status
● Active ● Inactive

Vehicles Connected
2

Kilometers Traveled
15200 km
Change: 1.33%

Model
Peugeot 208
2022

Fuel Logs Overview

FUEL AMOUNT
FUEL COST
Peugeot (17103-B-59)
Peugeot (12308-B-72)

Transactions for the Vehicle

Fuel Amount: 18.18 L Location: Shell	11/25/2024	200 MAD
Fuel Amount: 30 L Location: Shell Meknes	11/24/2024	360 MAD
Fuel Amount: 8.4 L Location: Shell Meknes	11/22/2024	100 MAD
Fuel Amount: 41.67 L Location: Shell Meknes	11/20/2024	500 MAD
Fuel Amount: 25 L Location: Afriquia Meknes	11/15/2024	300 MAD

Tracker Status

Oil Change	Overdue: 5200!	Max: 10000
Tires Change	Remaining: 34800	Max: 50000
Distribution Chain Change	Remaining: 134800	Max: 150000

Peugeot (17103-B-59)

GPS

Figure 9: User-friendly Dashboard

The Admin Interface contains a fairly well laid out sidebar navigation panel which directs one easily into key functionalities. The whole panel is broadly divided besides the Dashboard into three segments; Pages, Charts, and Data.

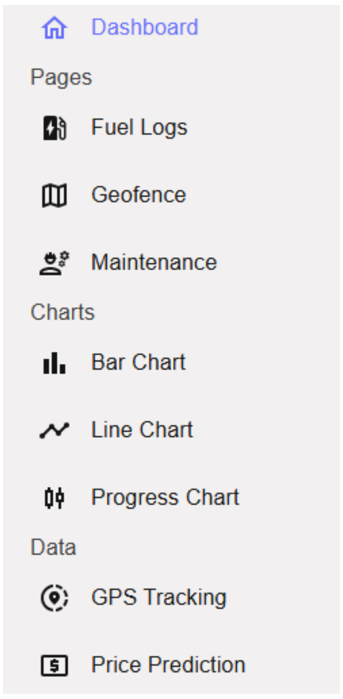


Figure 10: Admin Dashboard Sidebar Navigation

The Pages section of this panel directs users to the Fuel Logs, Geofence Management, and Maintenance Tracking for an administrator to oversee, manage, and report performance aspects of the fleet. In Charts, through the availability of bar charts, line charts, and maintenance progress visuals, the collection shows an overview of fleet operations and performance. Lastly, the Data includes GPS tracking and Price Prediction. GPS Tracking maps out a proper road view of vehicle locations and uses a Price Prediction tool which estimates resale value based on attributes of the vehicles. Such a structure provides an easy path of administration without difficulties.

8.2.4 Frontend Development with React-native

The mobile application for drivers, designed in React Native, has a neat interface for fuel log entries. The app enables drivers accurately to make fuel entries with minimum errors through validation.

Functionality

The mobile phone application allows drivers to quickly perform all the tasks that are priority-oriented to the users of this application, regarding fuel log entry. For example, here a driver selects a vehicle from the dropdown list, so that while making this entry it is associated with the right vehicle. Each vehicle entry is recognized by its plate number and model, thus providing clarity at data input. Other details required to be done by the driver include total fuel expense, cost per liter, the place where the transaction is done, and odometer reading at the time of refuel. The mobile application also enforces validation rules for ensuring data consistency and accuracy. For instance, the odometer reading ought to always be more than the last recorded value. The blank entries or erroneous inputs trigger error messages that guide the user in correcting their data before submission.

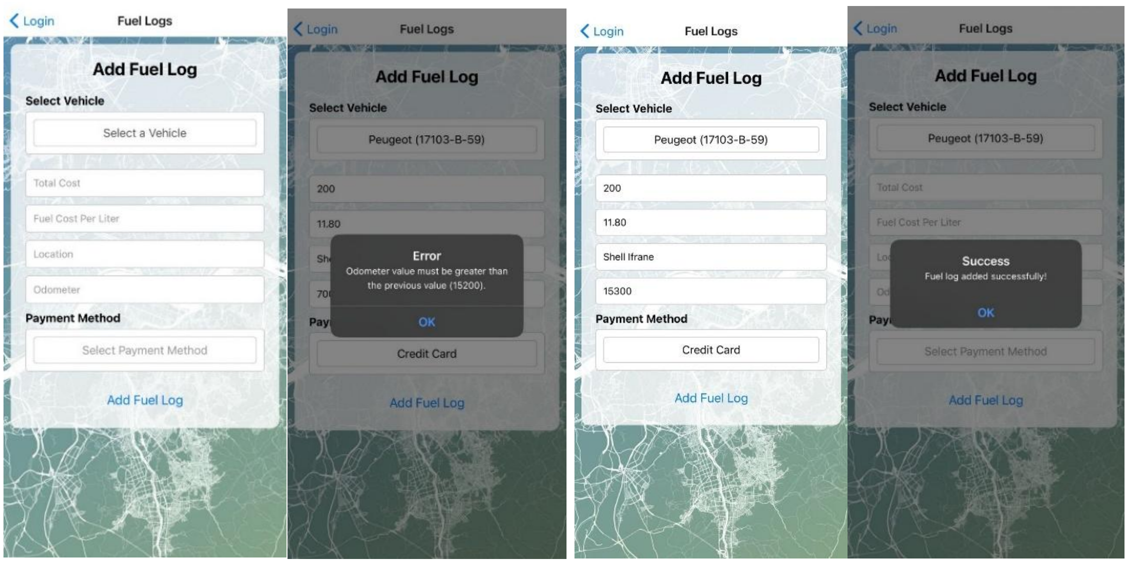
Technical Implementation

Using React Native, the mobile application for drivers was developed and can be used by any other user, whether it is on iOS or Android. It is an app that connects to the backend API based in Spring Boot in order to log fuel entries real-time for synchronization. So, the data entered by the drivers is stored in a MySQL database with an automatic update onto the admin dashboard. One major challenge in development was always dynamically validating the odometer readings without failure during data entry. This was solved by fetching the last recorded value from the back end at the time of fuel log submission and checking it against the new input. Other issues also revolving around the use of synchronization due to some disconnections in the network were resolved by the retry mechanisms and the status indicators which enable the users to understand if their submissions went through or not.

Design Considerations

Intuitive and accessible, the app's interface is for users who might not have solid technical grounding, for instance-the drivers themselves. Layouts have clean and very well labeled input fields, thus enabling users to navigate their way through the app and input whatever is needed easily. Also, error messages and tips to guide the users on how to rectify those issues such as invalid data entry and blanks are incorporated within the app. The app was also meant for use in devices with poorer processing capabilities an aspect that allows for its smooth running and usability under a wide range of mobile devices. That was to be an approach toward providing a better user experience as well as the reliability and functionality of that app.

Driver's View



Peugeot (17103-B-59)
200
11.80
Shell Ifrane
15300
Credit Card

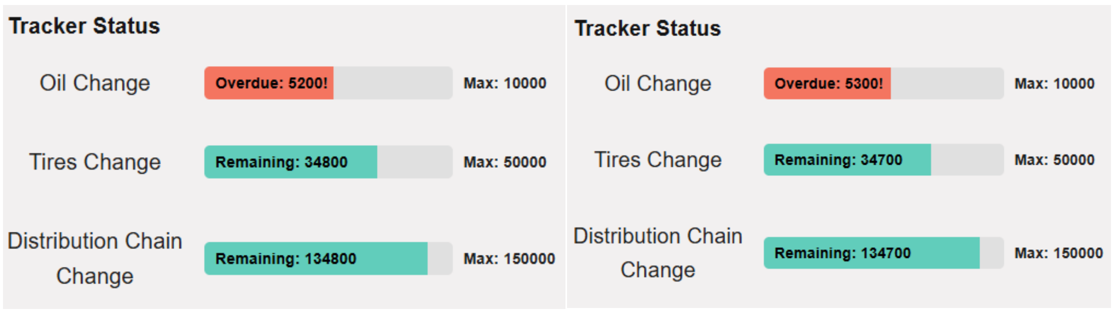
Error
Odometer value must be greater than the previous value (15200).

Success
Fuel log added successfully!

Figure 11: Fuel log entry page in the driver's mobile app

As shown in Figure 9 the fuel log entry page includes dropdown menus and validation rules to ensure accurate data submission. The example of 'Error' is shown through entering an odometer value smaller than the previous one. Then 15,300 km was entered and we can see the 'Success' message.

Admin's View



Tracker Status

Oil Change — Overdue: 5200! — Max: 10000

Tires Change — Remaining: 34800 — Max: 50000

Distribution Chain Change — Remaining: 134800 — Max: 150000

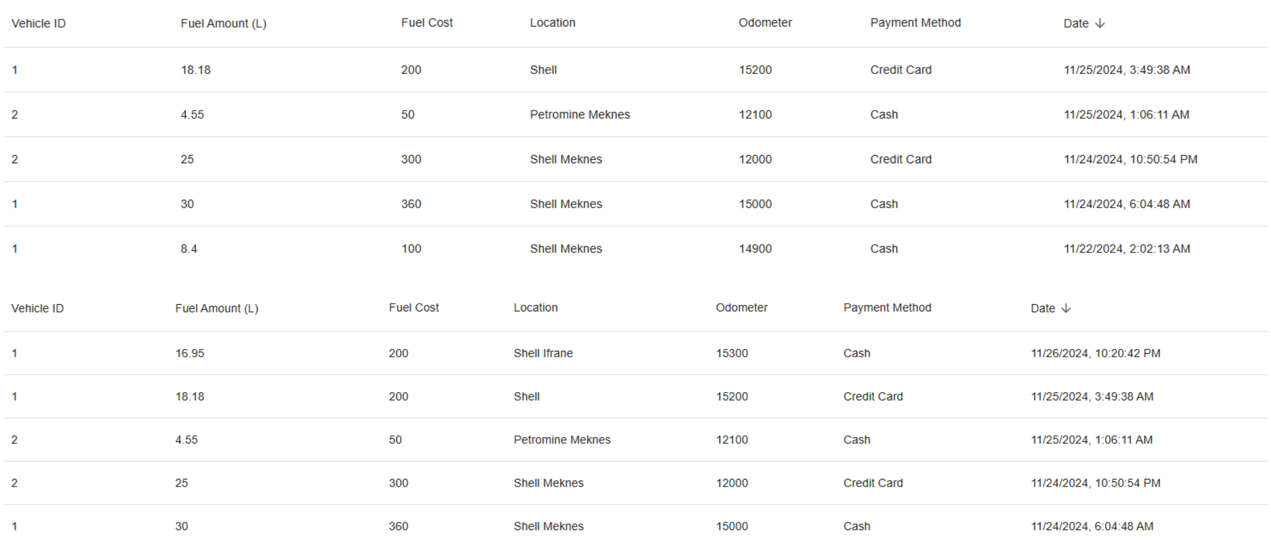
Tracker Status

Oil Change — Overdue: 5300! — Max: 10000

Tires Change — Remaining: 34700 — Max: 50000

Distribution Chain Change — Remaining: 134700 — Max: 150000

Figure 12: The maintenance tacker before and after fuel log



Vehicle ID	Fuel Amount (L)	Fuel Cost	Location	Odometer	Payment Method	Date ↓
1	18.18	200	Shell	15200	Credit Card	11/25/2024, 3:49:38 AM
2	4.55	50	Petromine Meknes	12100	Cash	11/25/2024, 1:06:11 AM
2	25	300	Shell Meknes	12000	Credit Card	11/24/2024, 10:50:54 PM
1	30	360	Shell Meknes	15000	Cash	11/24/2024, 6:04:48 AM
1	8.4	100	Shell Meknes	14900	Cash	11/22/2024, 2:02:13 AM

Vehicle ID	Fuel Amount (L)	Fuel Cost	Location	Odometer	Payment Method	Date ↓
1	16.95	200	Shell Ifrane	15300	Cash	11/26/2024, 10:20:42 PM
1	18.18	200	Shell	15200	Credit Card	11/25/2024, 3:49:38 AM
2	4.55	50	Petromine Meknes	12100	Cash	11/25/2024, 1:06:11 AM
2	25	300	Shell Meknes	12000	Credit Card	11/24/2024, 10:50:54 PM
1	30	360	Shell Meknes	15000	Cash	11/24/2024, 6:04:48 AM

Figure 13: Fuel log table with the new input

Figures 10 and 11 illustrate the change made when entering the new fuel log by the driver as seen in Figure 9. The inputs are now visible on the table in the fuel log page for the admin's view as well as on the Trackers Status which the value of current odometer becomes 15300 which explains the remaining showing 'Overdue by 5300 km'.

8.2.5 Visualization Tools

To enhance the interpretation of data, several visualization libraries were used. Material-UI was utilized for building the user interface, providing a polished and professional look. For creating dynamic and interactive charts, the Nivo library was employed, enabling clear visual representation of fuel logs, maintenance statuses, and distance traveled. Additionally, Leaflet was used to integrate real-time maps, allowing administrators to track vehicle locations and trajectories seamlessly.

These tools ensure that the dashboard is both visually appealing and functionally rich, offering users the insights they need at a glance.

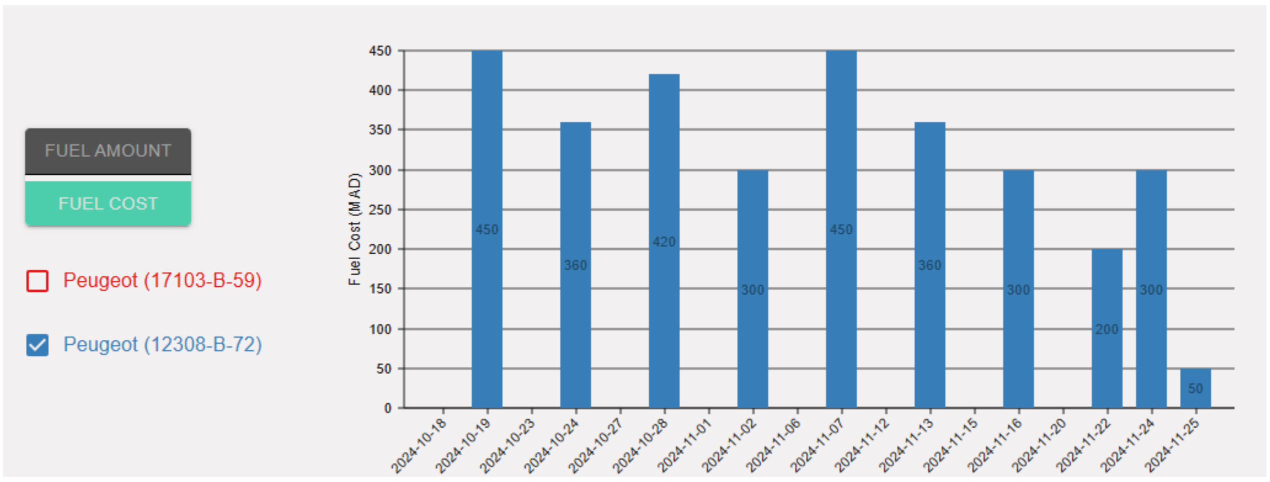


Figure 14: Bar Chart of Fuel logs (Nivo visualization)

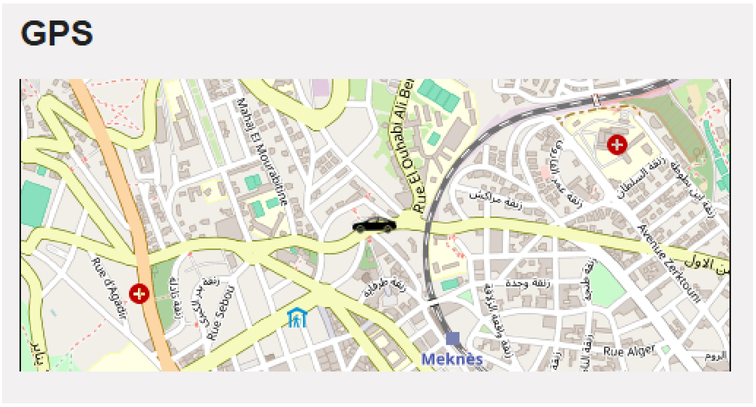
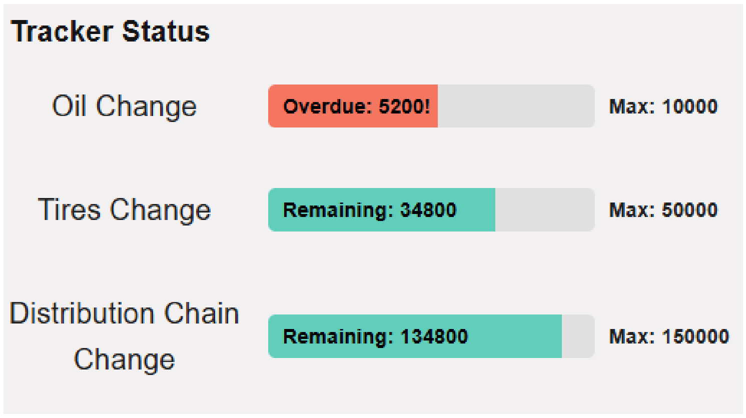


Figure 15: Real-time GPS map (Leaflet integration)



Tracker Status

Oil Change	Overdue: 5200!	Max: 10000
Tires Change	Remaining: 34800	Max: 50000
Distribution Chain Change	Remaining: 134800	Max: 150000

Figure 16: Maintenance progress visual (Material-UI progress bar)

9 Feasibility Study

Furthermore, this part contains evaluation of three important dimensions, namely technical feasibility, operational feasibility and economic feasibility is a scalable project to accommodate several vehicles and multiple stakeholders.

9.1 Technical Feasibility

The project employs technologies that are well established historical and therefore, it is technically feasible. The Teltonika FM5300 is a reliable IoT device that collects real-time data and transmits it via GPRS/3G networks. Integration with a solid backend, created with Spring Boot, ensures smooth processing of the data through the database, which is scalable and structured for data storage using MySQL.

The front end using React provides a responsive and user-friendly frontend that can manage very complex interactions and visualizations. The different tools like Material-UI, Nivo, and Leaflet provide such technical depth as creating professional-grade visualization and even mapping capabilities. The use of JWT tokens is a promise for secured communication, thus technical soundness for real world applications.

All of that can be said because the selected technology stack is itself scalable and adaptable, and in synch with nowadays software development methods. Compared to that, however, regular testing and bug fixing during development proved and validated the ability of the system to handle real-time data, offering an explicit guarantee for the technical feasibility of the project.

9.2 Operational Feasibility

It is the ability of the system to meet the demands of its users effectively, which makes it operationally feasible. Administrators would have specific data inclusive of vehicle performance, maintenance schedules, and geofencing, which would help them make informed decisions. For drivers, the mobile can simplify the process of entering fuel log, thus ensuring accurate information entry and minimal manual errors.

The system's modular design allows it to easily integrate into existing operations, resulting in minimal disruptions during the implementations. The project is equipped with intuitive user interfaces and real-time data updates, ensuring seamless day-to-day operation. Maintenance of infrastructure at Teltonika FM5300 devices and software supporting operational performance is considered.

9.3 Economic Feasibility

From an economic standpoint, the initiative offers the implementation of a relatively favourable fleet-management technology. It entails the investment for upfront costs including the purchase of Teltonika FM5300 devices and server setup, and recurring costs account for data subscriptions and periodic maintenance costs. However, those cost outlays will be eventually outweighed by long-term savings as a result of fuel utilization optimization, vehicle downtime reduction, and proactive maintenance.

This indeed walks the line towards operational cost reduction directly accounted in terms of having a dashboard capable of identifying inefficiencies and tracking vehicle performance. Its scalability allows the incorporation of more vehicles and users without heavy and significant increases in cost, essentially making this solution available to small and large fleets alike.

9.4 Scalability

The entire project has been designed so far as a monolithic system, wherein all components—data processing and storage, visualization and user interfaces—are integrated into a unified architecture. Such a design ensures simplicity in both development and deployment, making it very appropriate for the early stages of a project and manageable fleet sizes.

However, the limitations of monolithic structures begin to manifest in scaling for larger fleets or more users. With such expansion comes performance bottlenecks and issues of maintainability in the code base. For future improvement, a transition to microservices would be implemented to address these issues. This would mean decoupling the components of the system as independent services, therefore improving scalability and maintainability for the seamless injection of new vehicles, users, and capabilities.

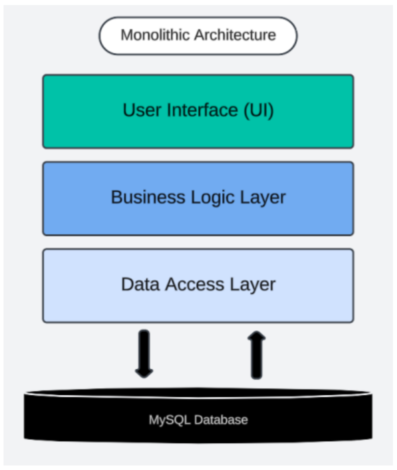


Figure 17: Monolithic Architecture Visual

10 Data Presentation

This section provides an overview of how data is structured and visualized within the system. It covers the database schema design in MySQL and examples of data visualization implemented in the dashboard.

10.1 Database Schema (MySQL)

The system is designed with a well-structured MySQL database to effectively handle data collected from the Teltonika FM5300 device. This schema ensures the smooth management of real-time data transmission and maintains scalability for multiple vehicles and stakeholders.

Table Name	**Description**	**Key Fields**
User Table	Stores user details and credentials for administrators and drivers.	id (Primary Key), first_name, last_name, email, dob (date of birth), password
Vehicles Table	Contains vehicle-specific data, including make, model, year, VIN, and odometer readings.	vehicle_id (Primary Key), plate_number, make, model, year, vin
Fuel Logs Table	Tracks fuel transactions, including amount, cost, location, and odometer readings at the time of refueling.	fuel_log_id (Primary Key), date, fuel_amount, fuel_cost, location, odometer, payment_method
Maintenance Table	Maintains records of vehicle maintenance activities, such as operation (oil changes, tires change, and distribution chain change) and associated costs.	maintenance_id (Primary Key), vehicle_id (Foreign Key), operation_type, price, maintenance_date, alert
Device Table	Stores data about the Teltonika FM5300 devices, such as the device ID, status, and associated vehicle.	device_id (Primary Key), imei, vehicle_id (Foreign Key), is_active
Geofence Table	Manages geofencing data, including city name, center latitude and longitude, and radius for creating geofenced areas.	geofence_id (Primary Key), name, center_latitude, center_longitude, radius

Table Name	Description	Key Fields
Alerts Table	Logs geofence breaches and other system alerts with associated vehicle and timestamp details.	alert_id (Primary Key), vehicle_id (Foreign Key), geofence_id (Foreign Key), latitude, longitude, timestamp, breach_status
Tracker Table	Keeps track of the values that result from the subtraction of the current odometer value and the threshold set the for the specified operation.	Id (Primary Key), operation_type, vehicle_id (Foreign Key), value, created_at ()sets automatically the date when the operation was performed.
AVL Data Table	Tracks advanced vehicle location (AVL) data, such as GPS coordinates, speed, angle, and satellites.	avl_data_id (Primary Key), device_id (Foreign Key), latitude, longitude, speed, angle, timestamp

Table 1: Tables in Database

User Table:

Stores credentials for all users, such as administrators and drivers, including their login details and profile information.

Vehicles Table:

Keeps detailed information about each vehicle, including its make, model, and unique VIN for identification. It also tracks odometer readings to monitor mileage and maintenance needs.

Fuel Logs Table:

Logs fuel transaction details, such as fuel quantity, cost, location, and odometer values. This data is crucial for analyzing fuel consumption and operational costs.

Maintenance Table:

Tracks vehicle maintenance records, including operation type, costs, and thresholds for activities such as oil changes or tire replacements.

Device Table:

Manages data about Teltonika FM5300 devices, including their unique IDs, IMEI numbers, and the vehicles they are attached to. It also tracks whether the devices are active or inactive.

Geofence Table:

Stores geofencing data, including city names and geographic boundaries, defined by latitude, longitude, and radius. This enables geofencing alerts.

Alerts Table:

Logs real-time alerts, such as geofence breaches, with details like time, location, and associated vehicle.

Tracker Table:

Keeps track of the increase in the odometer value to detect if the thresholds im km for the operation were reached to make them as an alert in red.

AVL Data Table:

Tracks real-time location data, such as GPS coordinates, speed, and angle, allowing for live vehicle monitoring.

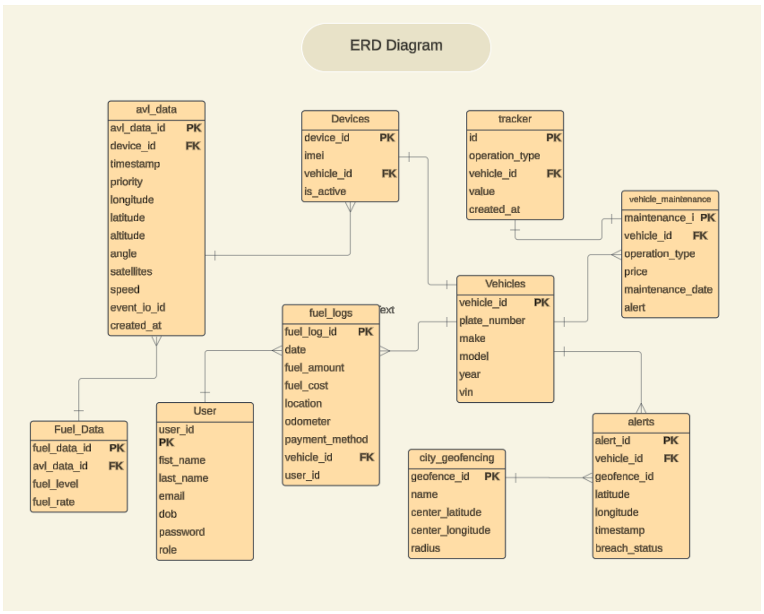


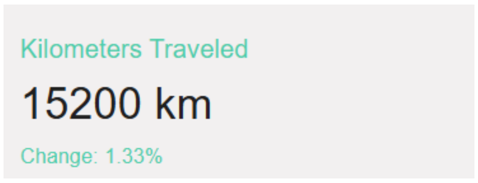
Figure 18: ERD Diagram of MySQL

10.2 Data Visualization Examples from the Dashboard

The dashboard transforms raw data into actionable insights using a variety of visualizations.

Visualization	Description	Example
Fuel Logs Overview	Bar charts display daily fuel consumption and costs. Administrators can filter data to view specific vehicles or time periods.	A bar chart showing fuel consumption (liters) and cost (currency) for the past month.
Distance Traveled	Line charts represent the distance traveled by vehicles over time, highlighting trends and inefficiencies.	A line chart showing kilometers traveled daily for the past month.
Maintenance Progress	Progress bars visualize the status of maintenance tasks (oil changes, tires change, and distribution chain change) with color-coded thresholds for overdue items.	A progress bar showing "Remaining: 2,000 km" for an oil change, with overdue items in the red, means that the operation should have been done 2,000 km before.
Real-Time GPS Tracking	Interactive maps show current vehicle locations and trajectories. Clicking on a vehicle icon displays details such as speed and timestamp.	A live map displaying vehicle icons at their current locations, along with recent route history.
Geofencing Alerts	Alerts pop up when a vehicle breaches a geofenced area, displaying breach time and location.	A notification stating: "Vehicle Peugeot 208 exited Casablanca geofence at 12:30 PM."
Vehicle Timelines	Feature provides a chronological representation of the key points visited by vehicles, offering a clear visualization of vehicle movement and activities.	"Boulevard Amir Moulay Abdallah, Meknes, Morocco 11/10/2024, 3:35:43 PM 21 km/h 245°"Visual with five different locations that the vehicle crossed with speed, date, angle.
Device Status	Displays the status of the devices connected to vehicles. A pie chart visualizes active vs inactive devices.	Device Status showing the proportion of active and inactive devices in the fleet.
Vehicles Connected	Shows the total number of vehicles currently connected to the system.	Total number of vehicles connected to the system displayed as a counter (e.g., 2 vehicles)
Kilometers Traveled	Indicates the total distance covered by a vehicle, extracted from odometer readings, and displays percentage change compared to the previous value.	Total kilometers traveled by the fleet, with percentage change indicating progress.
Model	Displays the make and model of the currently selected vehicle along with its manufacturing year.	Vehicle model and year of manufacturing for the selected vehicle (e.g., Peugeot 208, 2022).

Table 2: Visualizations and Descriptions



Kilometers Traveled

15200 km

Change: 1.33%

Figure 19: Kilometers Travelled

Displayed in Figure 19 is the total kilometers covered by a selected vehicle derived from its odometer data, including a percentage change that refers to the difference from its last odometer reading- which then provides an overview of how used the vehicle has been recently.

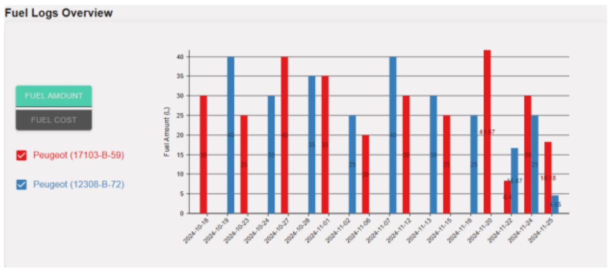
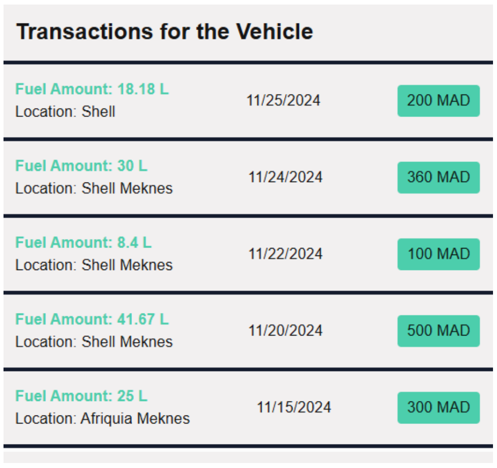


Figure 20: Fuel logs Overview

The bar chart in Figure 20 presents fuel consumption in liters and its cost by vehicle type, against time. The left axis is dedicated to the fuel amounts in liters and the right axis depicts the local currency cost amounts; it has similar colors for different vehicles for easy trend analysis by the administrator.



Transactions for the Vehicle

Fuel Amount: 18.18 L Location: Shell	11/25/2024	200 MAD
Fuel Amount: 30 L Location: Shell Meknes	11/24/2024	360 MAD
Fuel Amount: 8.4 L Location: Shell Meknes	11/22/2024	100 MAD
Fuel Amount: 41.67 L Location: Shell Meknes	11/20/2024	500 MAD
Fuel Amount: 25 L Location: Afriquia Meknes	11/15/2024	300 MAD

The selected vehicle's most current fuel transaction for stipulation of the amount of fuel, its total cost, the location, and transaction date includes fuel transactions, which helps administrator easily access and check transaction records for operational transparency.

Figure 21: Transactions for the Vehicle

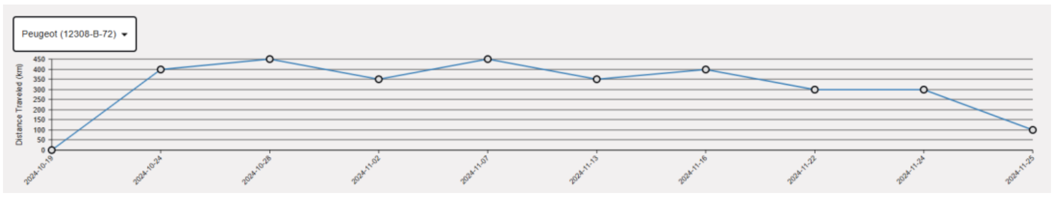


Figure 22: Distance Travelled

The line graph in Figure 22 shows the distance a vehicle covers each fuel log operation. This is presented during a defined period. It extracted from the odometer value by checking the value from which it increases after the driver enters it.

11 Simulations, Results, and Interpretation

This section focuses on the results and insights derived from the system, showcasing the effectiveness of real-time data transmission, data visualization, maintenance tracking, and geofencing alerts.

11.1 Real-Time Data Transmission and Visualization Results

The system was successfully tested to ensure seamless real-time data transmission from the Teltonika FM5300 device to the server. Data collected from vehicle sensors, including GPS coordinates, speed, and fuel logs, were encoded, transmitted via GPRS, and decoded at the server. The decoded data was stored in the MySQL database and visualized in real-time on the dashboard.

Results found:

Data latency was minimal, ensuring timely updates on the dashboard.

Accurate representation of vehicle locations on interactive maps.

Real-time bar and line charts dynamically updated with new data, enhancing decision-making.

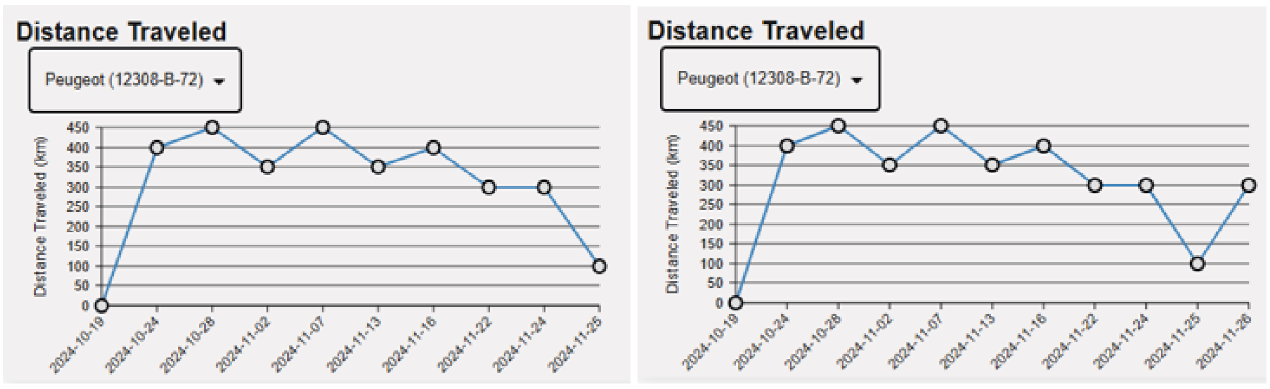


Figure 23: Distance travelled after change

In the figure we can see the change that happened when adding new fuel log inputs in the distance travelled chart as it records the km travelled between each new fuel transaction. So, we can deduce that the driver has made a distance of 300 km between the 25th and 26th.

11.2 Maintenance Tracking

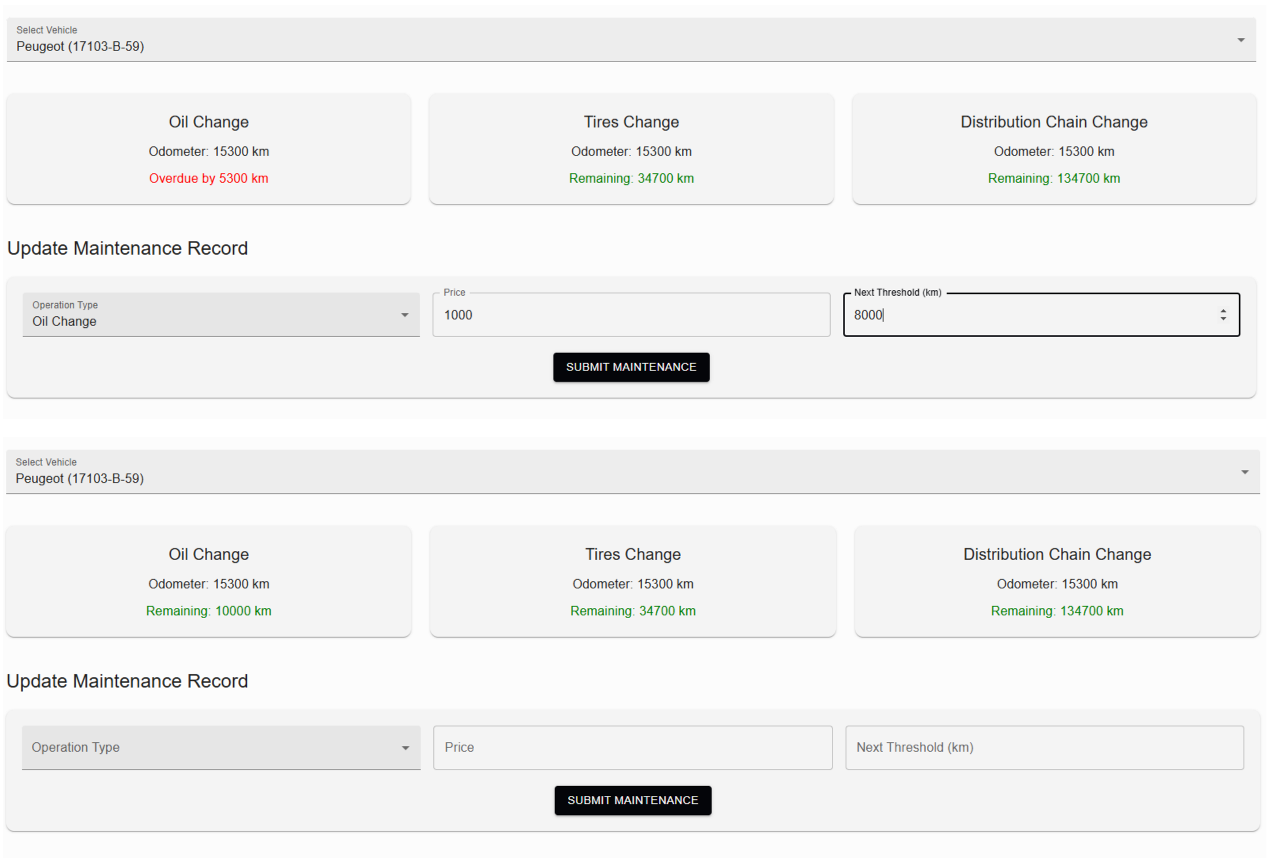
The system effectively monitors vehicle maintenance requirements using thresholds for oil changes, tire replacements, and distribution chain changes. Progress bars provide a clear overview of remaining kilometers until the next maintenance operation, with overdue maintenance flagged in red.

Results found:

Overdue maintenance alerts were successfully triggered based on odometer readings.

Users were able to update maintenance records via the page provided from the side bar, resetting thresholds and maintaining accuracy.

Visual progress bars provided an easy-to-understand summary of maintenance needs.


Select Vehicle
Peugeot (17103-B-59)

Oil Change	Tires Change	Distribution Chain Change
Odometer: 15300 km	Odometer: 15300 km	Odometer: 15300 km
Overdue by 5300 km	Remaining: 34700 km	Remaining: 134700 km

Update Maintenance Record

Operation Type: Oil Change
Price: 1000
Next Threshold (km): 8000

Select Vehicle
Peugeot (17103-B-59)

Oil Change	Tires Change	Distribution Chain Change
Odometer: 15300 km	Odometer: 15300 km	Odometer: 15300 km
Remaining: 10000 km	Remaining: 34700 km	Remaining: 134700 km

Update Maintenance Record

Figure 24: Maintenance after Change

In this figure we can see the change happening on the maintenance page when the admin records that the oil change where there was an overdue. After clicking on submit maintenance we can clearly see that the remaining is set to the new threshold entered by the admin.

11.3 Geofencing Alerts

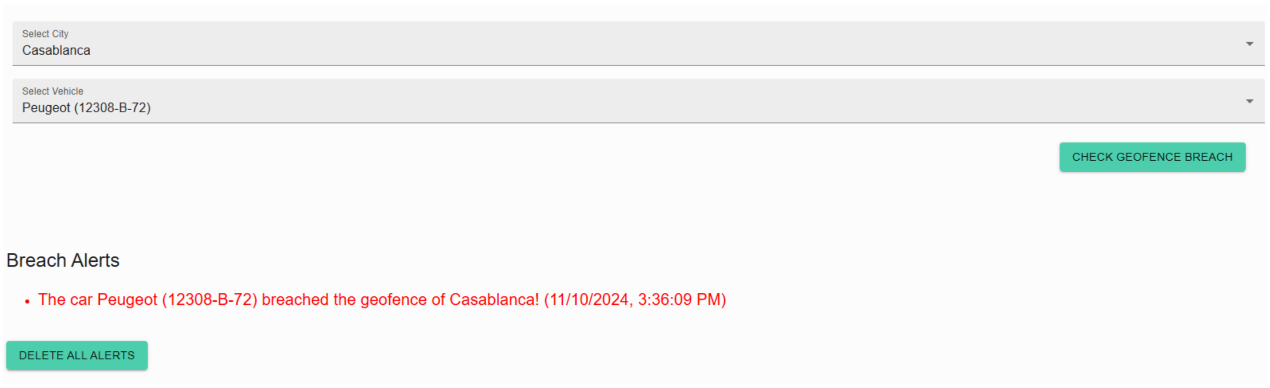
The geofencing feature demonstrated its ability to monitor vehicle movements within predefined boundaries. Alerts were generated when vehicles exited geofenced areas, displaying details such as vehicle ID, location, and breach time. These alerts were essential for enforcing compliance and enhancing security.

Results found:

Geofence boundaries accurately triggered alerts when breached.

Real-time alert notifications were displayed on the dashboard.

Admins could manage and delete alerts, maintaining a clean interface.


Select City: Casablanca

Select Vehicle: Peugeot (12308-B-72)

Breach Alerts

- The car Peugeot (12308-B-72) breached the geofence of Casablanca! (11/10/2024, 3:36:09 PM)

Figure 25: Geofencing alert

11.4 Timeline Tracking

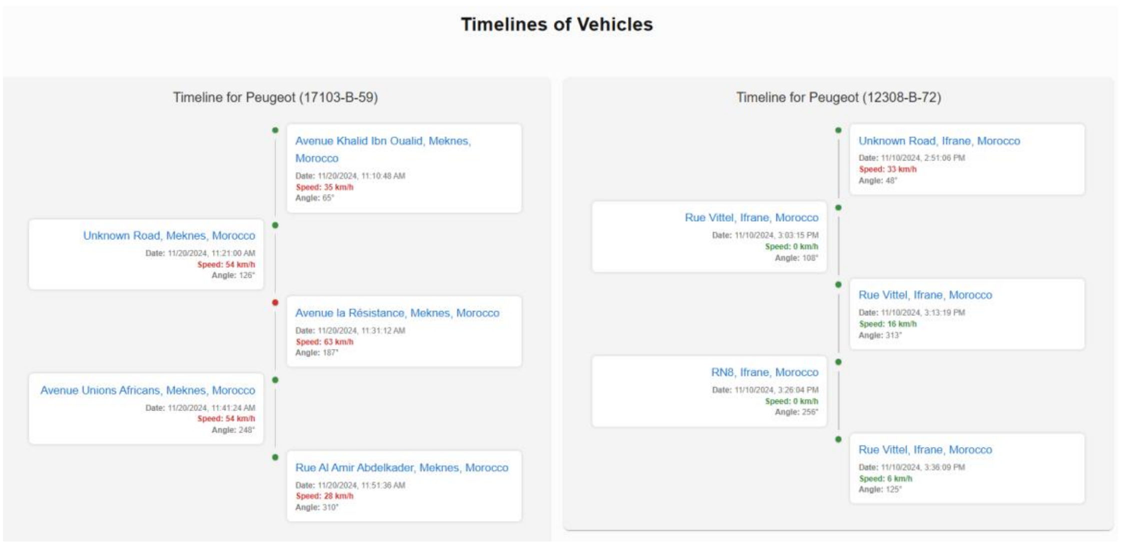
The timeline facility in Figure 23 gives a linearly arranged visual display of the journeys done by a vehicle, stressing particular sites along the route. These include the date, speed, and angle for each of the stops along the route. This feature provides route tracking in real time with an analysis of driving behavior for optimization purposes to the administrator. The timeline data is dynamically updated in real-time, ensuring accurate visualization of vehicle trajectories from point A to point B.

Results found:

Route efficiency as the data highlighted deviations from planned routes, enabling the administrator to investigate causes, such as traffic or driver decisions.

Driving behavior as speed and angular data helped identify instances of sharp turns or sudden stops, which may indicate unsafe driving practices.

Real-Time Monitoring which provides the ability to track vehicles in real-time ensured swift responses to potential geofencing breaches or route anomalies.



Timelines of Vehicles

Timeline for Peugeot (17103-B-59)

Avenue Khalid Ibn Oualid, Meknes, Morocco
Date: 11/20/2024, 11:10:48 AM
Speed: 35 km/h
Angle: 65°

Unknown Road, Meknes, Morocco
Date: 11/20/2024, 11:21:00 AM
Speed: 54 km/h
Angle: 126°

Avenue la Résistance, Meknes, Morocco
Date: 11/20/2024, 11:31:12 AM
Speed: 63 km/h
Angle: 187°

Avenue Unions Africans, Meknes, Morocco
Date: 11/20/2024, 11:41:24 AM
Speed: 54 km/h
Angle: 248°

Rue Al Amir Abdelkader, Meknes, Morocco
Date: 11/20/2024, 11:51:36 AM
Speed: 28 km/h
Angle: 310°

Timeline for Peugeot (12308-B-72)

Unknown Road, Ifrane, Morocco
Date: 11/10/2024, 2:51:06 PM
Speed: 33 km/h
Angle: 48°

Rue Vittel, Ifrane, Morocco
Date: 11/10/2024, 3:03:15 PM
Speed: 0 km/h
Angle: 108°

Rue Vittel, Ifrane, Morocco
Date: 11/10/2024, 3:13:19 PM
Speed: 16 km/h
Angle: 313°

RN8, Ifrane, Morocco
Date: 11/10/2024, 3:26:04 PM
Speed: 0 km/h
Angle: 256°

Rue Vittel, Ifrane, Morocco
Date: 11/10/2024, 3:36:09 PM
Speed: 6 km/h
Angle: 125°

Figure 26: Timeline Tracking for Vehicle Movements

11.5 Price Prediction Feature

The price prediction feature boosts the capability of the dashboard by allowing admin to forecast the resale price of vehicles. A friendly interface for capturing such attributes is developed so that the admin can input vehicle types, year, make, model, trim, color, body style, and mileage in kilometers. The system then predicts using an external API and returns the price estimation along with a price range.

Implementation Details:

The feature has been implemented using React for the frontend with API communication through Axios. The form is designed to capture all the entries and validate them for completeness and correctness. Once submitted, it passes the input to the API and reflects the price estimation in USD. A conversion step directly translates this into MAD for its significance in local context. It also has all the error handling on an incomplete input or connection to the API.

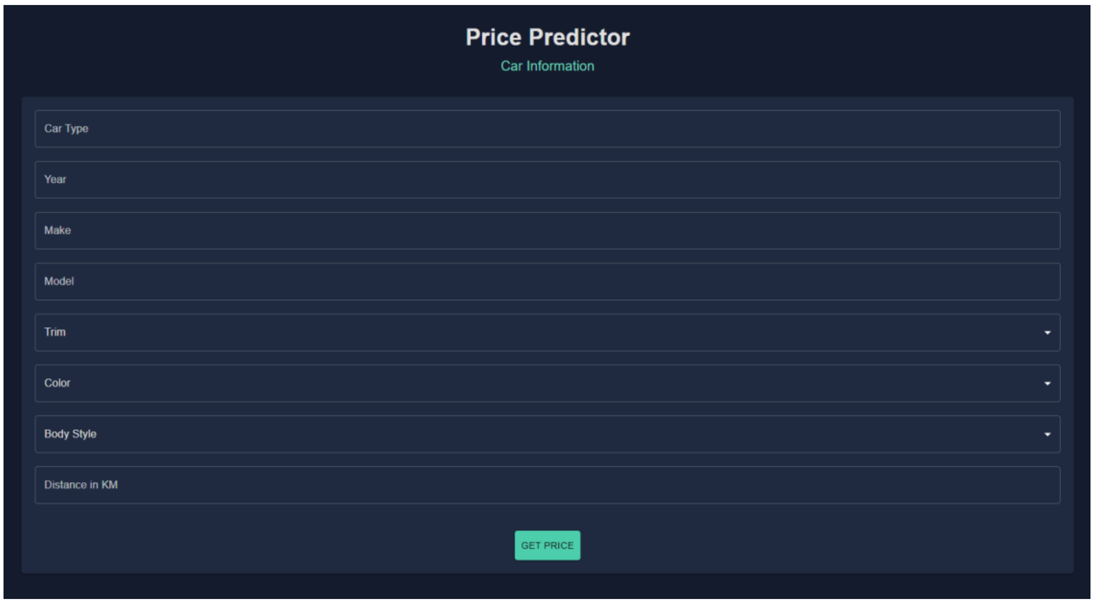


Figure 27: Input form for vehicle details

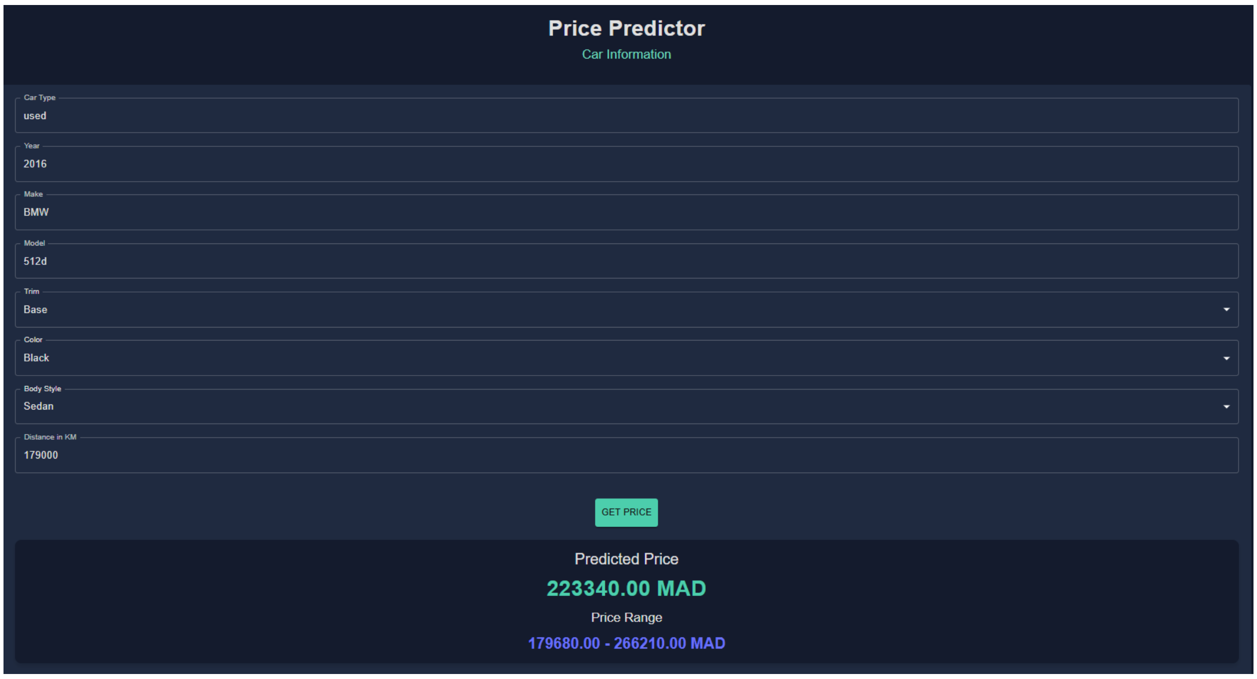
The admin enters all relevant details about the vehicle into the form.

The system converts its mileage from kilometers to miles for API compatibility.

The request is sent to the API, then the predicted price and range are retrieved.

The next step converts everything into MAD, using the actual and real-time exchange rates.

Finally, the results are displayed on the dashboard, together with the estimated price and price range.

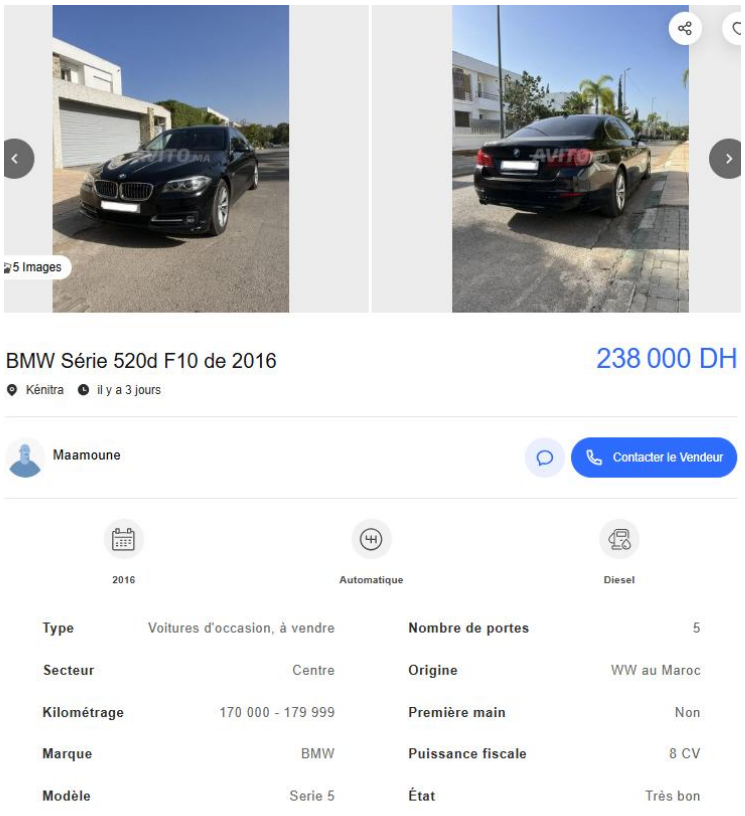


Car Type	used
Year	2016
Make	BMW
Model	512d
Trim	Base
Color	Black
Body Style	Sedan
Distance in KM	179000

Predicted Price
223340.00 MAD
Price Range
179680.00 - 266210.00 MAD

Figure 28: Prediction results, including the estimated price and range

For instance, as illustrated in the Figure 28 when predicting the price for a 2016 BMW 512d, the system returned an estimated value of 223,340 MAD, with a range between 179,680 MAD and 266,210 MAD. This range aligns closely with an actual market offer of 238,000 MAD, validating the feature's accuracy and utility as observed on the Figure 29 brought from an ‘Avito’ post.



BMW Série 520d F10 de 2016

238 000 DH

Kénitra · il y a 3 jours

Type	Voitures d'occasion, à vendre	Nombre de portes	5
Secteur	Centre	Origine	WW au Maroc
Kilométrage	170 000 - 179 999	Première main	Non
Marque	BMW	Puissance fiscale	8 CV
Modèle	Serie 5	État	Très bon

Figure 29: Market offer example validating the prediction accuracy

12 Learning Strategies

In this figure the last data entered for the car selected was in Ifrane which means not in Casablanca. Since I do not have a vehicle, I could not test the geofencing dynamically, so I just click on the check geofence breach button I get the alert that shows which car is not in the geofence and in which time the geofence was breached.

12.1 Technical Skills Acquired

During this project, several technical skills were developed and upscaled, including IoT configuration, backend development, and frontend implementation.

IoT Configuration

Configuring an actual Teltonika FM5300 IoT device has led to practical involvement in real-time data collection and transmission where a setup device was configured to capture GPS data, CAN bus metrics as well as other vehicle parameters. Coding data in hexadecimal and transmitting it through GPRS/3G was mastered alongside server configuration that decodes received packets, thereby safe-keeping data in structured databases. Besides, encryption being incorporated for safe communication provided real-life experience of IoT security protocols.

Backend Development

The creating of backend applications with Spring Boot improved the technical capabilities in which have been gained in server-side programming and database integration. It has effectively brought knowledge especially into the modular and maintainable design of a system through the implementation of Model-View-Controller (MVC) architecture. It was one hands-on experience in security on authentication and authorization methods by working with user authentication using the JWT token. Further encouraged proficiency in back-end development includes the interaction that brought MySQL as the storage mechanism and ensured that it worked well with the IoT device to server and frontend interaction.

Frontend Implementation

The making of the dashboard and mobile application through React and React Native sharpened frontend development skills. Usage of libraries such as Material-UI and Nivo for creating compelling visualizations and thus enriching skills to create intuitive user interfaces. Such functionalities as real-time GPS tracking, Geofencing alerts, as well as maintenance progress tracking stand testament to having had practical experience bringing dynamic and interactive web applications into life.

12.2 Project Management Skills

The deployment of this project had the effect of developing competence in vital project management skills, like planning, teamwork, and problem-solving strategies.

Planning

For instance, effective planning was the key to managing the overall complexity posed by the integration of IoT hardware, backend services, and frontend interfaces. There was systematic completion of tasks by breaking the project down into phases that were manageable, like device configuration, backend development, and frontend implementation. Gantt charts and task management software were used to track the project progress, prioritize tasks, and meet deadlines. Planning also involved stating clear goals for the system, such as dashboards according to roles, maintenance tracking, and geofencing alerts. This was all done to ensure that development efforts remain aligned toward the project goals.

Teamwork

Though it was an individual project, I would like to appreciate my supervisor, Dr. Amine Abouaomar, for giving me guidance, advice, and support throughout the project duration. With regular meetings with my supervisor, I would remind him of the progress I was making, raise issues I was facing during the project, and obtain constructive criticism on my work. This would thereby help me bring the project in line with what was earlier defined and what was best practice for such a project.

Working alone on such a major project really helped me learn independence as well as instill some good skills in the management of time. The contributions made by my supervisor ensured that I stayed on track while delivering high standards in both technical implementation and project management. This was an indicator that seeking expert advice while carrying out responsibility in the development process would help in balancing autonomy and mentorship.

Problem-Solving Strategies

Challenges were faced during the project, such as configuring the Teltonika FM5300 device, real-time data synchronization, and scalability of the system. A systematic approach was taken to each of these through problem-solving techniques such as root-cause analysis and iterative testing. For example, debugging issues in the data transmission involved analyzing the packet structures and examining server configurations. In addition, feedback loops established around features such as geofencing alerts and JWT-based authentication make it possible to build robust functionalities.

These project management skills were not only useful for the successful delivery of the system but also for effective transformation to more complex real-life projects in the world of professional environments.

13 Conclusion

Accomplishment summary

The project, which connected IoT technology to a very user-friendly dashboard, eventually provided a completely embedded, fully functional vehicle tracking and fleet management solution. The Teltonika FM5300 device was configured to collect real-time data, such as GPS, speed, and maintenance metrics, and to send it securely via a GPRS/3G network to a centralized server. This back-end was developed using Spring Boot and efficiently processes and stores data into a MySQL database, while the frontend, built with React and React Native, provided intuitive interfaces tailored to administrators and drivers.

It implemented some important features like live GPS tracking, geofencing alerts, maintenance tracking, and a visualization of fuel logs, which eventually give actionable insights and improve operational efficiency. Moreover, the system used JWT-based authentication so that… Secure user access it can ensure. This offers a sustainable solution for the fleet of all sizes and provides a significant effect on the industries which mostly depend on transportation.

Future Work Recommendations

Although the project has met the desired objectives, there are still a number of fine-tuning points that can be added to enhance and scale the systems towards improvement. A gradual shift from a monolithic architecture to microservice systems can accommodate scalability and more effective handling of larger fleets and operations' complexity.

Predictive maintenance can effectively be incorporated with modern-day machine learning algorithms, thus enabling proactive identification of potentially faulty vehicles before they cause any down time in operations, and optimizing fleet operation. Moreover, introducing real-time analytics for factors such as driving behavior-hard braking, speeding, etc., would allow greater visibility in terms of fleet safety and compliance in operations.

Further functionalities in the mobile application are route optimization, real-time alerts, and others to offer more usability and efficiency in operations to drivers. Adopting these enhancements would transform the project into a more scalable, intelligent, and user-oriented system that would remain relevant even to the changing needs of the organization in Fleet Management.

REFERENCES

Teltonika. (n.d.). Teltonika FM5300 User Manual. Retrieved from https://www.getic.lt/files/catalogue/teltonika/fm5300-user-manual-v3-57.pdf

Oracle. (2023). MySQL Documentation. Oracle Corporation. Retrieved from https://dev.mysql.com/doc/

Pivotal Software, Inc. (n.d.). Spring Boot Documentation. Retrieved from https://spring.io/projects/spring-boot

Meta Platforms, Inc. (n.d.). React Native Documentation. Retrieved from https://reactnative.dev/docs/getting-started

Teltonika. (n.d.). Teltonika FM5300 User Manual. Retrieved from https://teltonika-gps.com/product/fm5300/

Leaflet. (n.d.). Leaflet: Interactive Maps Documentation. Retrieved from https://leafletjs.com/

Nivo. (n.d.). Nivo: React Data Visualization Components. Retrieved from https://nivo.rocks/

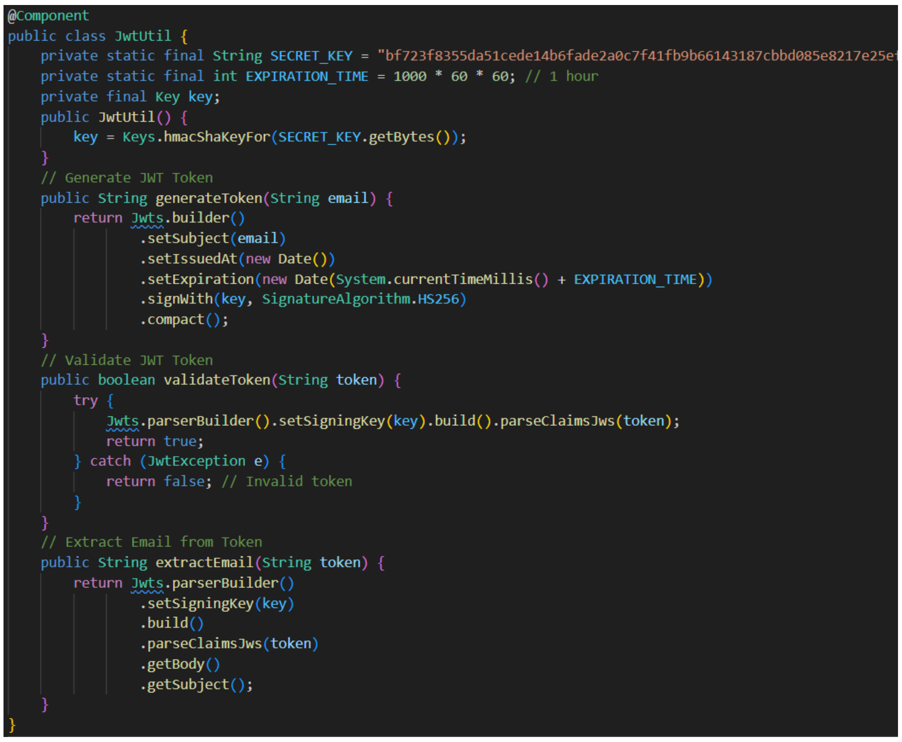
Material-UI. (n.d.). Material-UI Documentation. Retrieved from https://mui.com/material-ui/getting-started/overview/

ExchangeRate-API. (n.d.). Exchange Rate API Documentation. Retrieved from https://www.exchangerate-api.com/

Marketcheck. (n.d.). Marketcheck API Documentation. Retrieved from https://marketcheck.com/api/

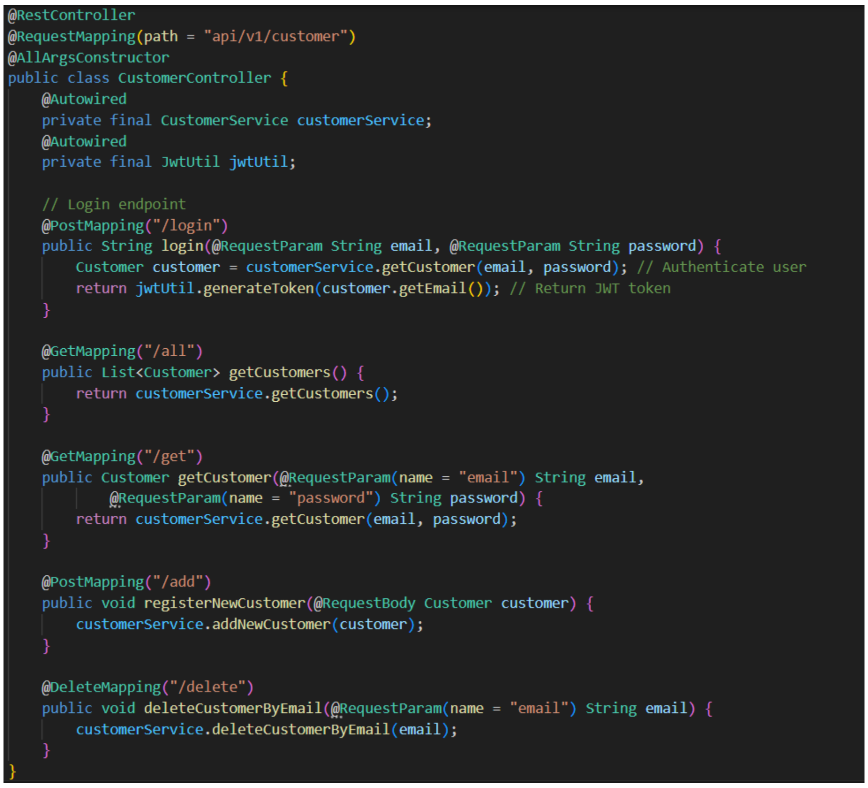
APPENDIX A: CODE

Backend Code



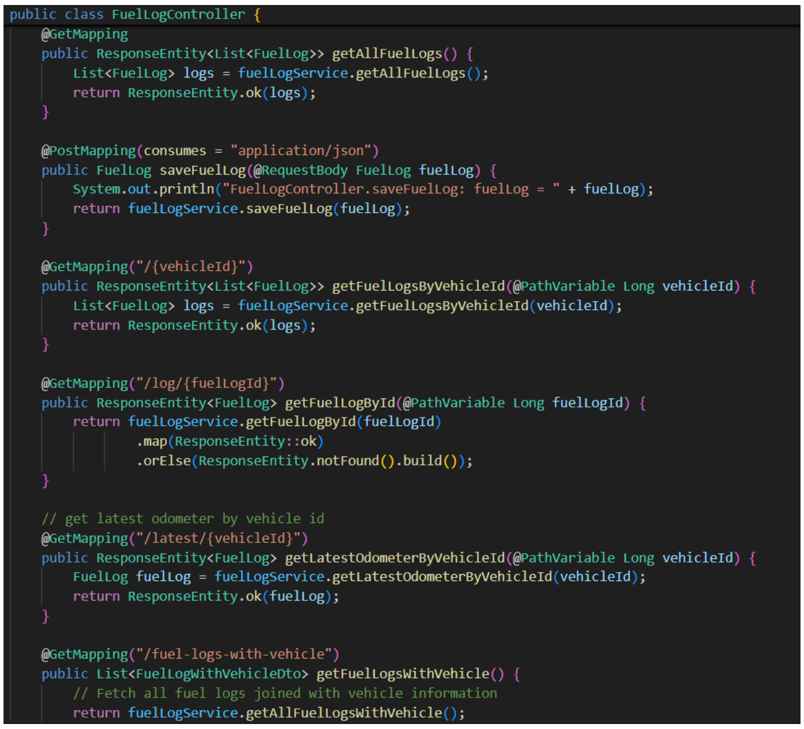
```java
@Component
public class JwtUtil {
    private static final String SECRET_KEY = "bf723f8355da51cede14b6fade2a0c7f41fb9b66143187cbbd085e8217e25e
    private static final int EXPIRATION_TIME = 1000 * 60 * 60; // 1 hour
    private final Key key;
    public JwtUtil() {
        key = Keys.hmacShaKeyFor(SECRET_KEY.getBytes());
    }
    // Generate JWT Token
    public String generateToken(String email) {
        return Jwts.builder()
                .setSubject(email)
                .setIssuedAt(new Date())
                .setExpiration(new Date(System.currentTimeMillis() + EXPIRATION_TIME))
                .signWith(key, SignatureAlgorithm.HS256)
                .compact();
    }
    // Validate JWT Token
    public boolean validateToken(String token) {
        try {
            Jwts.parserBuilder().setSigningKey(key).build().parseClaimsJws(token);
            return true;
        } catch (JwtException e) {
            return false; // Invalid token
        }
    }
    // Extract Email from Token
    public String extractEmail(String token) {
        return Jwts.parserBuilder()
                .setSigningKey(key)
                .build()
                .parseClaimsJws(token)
                .getBody()
                .getSubject();
    }
}
```

Figure 30: Utility class for generating, validating, and parsing JWT tokens



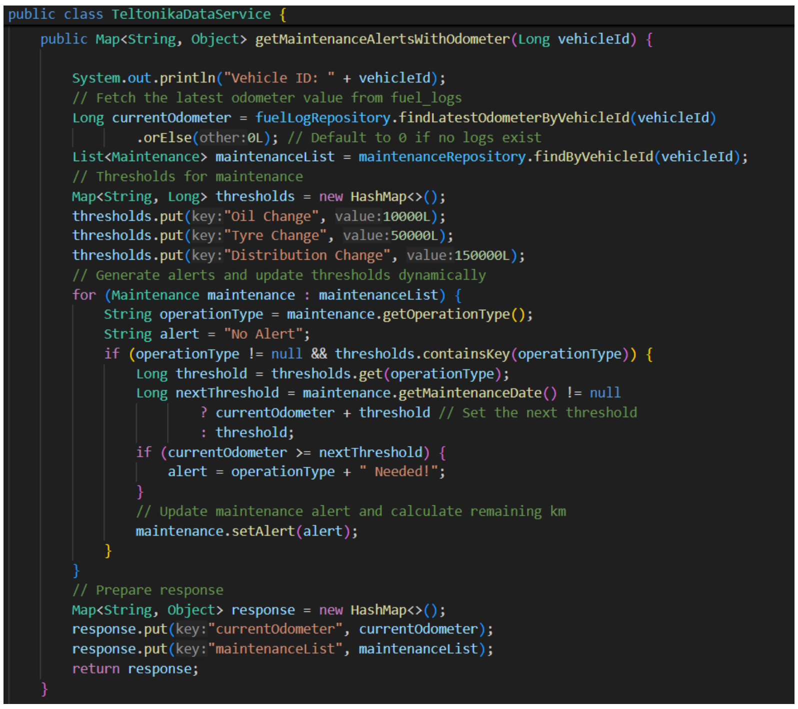
```java
@RestController
@RequestMapping(path = "api/v1/customer")
@AllArgsConstructor
public class CustomerController {
    @Autowired
    private final CustomerService customerService;
    @Autowired
    private final JwtUtil jwtUtil;

    // Login endpoint
    @PostMapping("/login")
    public String login(@RequestParam String email, @RequestParam String password) {
        Customer customer = customerService.getCustomer(email, password); // Authenticate user
        return jwtUtil.generateToken(customer.getEmail()); // Return JWT token
    }

    @GetMapping("/all")
    public List<Customer> getCustomers() {
        return customerService.getCustomers();
    }

    @GetMapping("/get")
    public Customer getCustomer(@RequestParam(name = "email") String email,
            @RequestParam(name = "password") String password) {
        return customerService.getCustomer(email, password);
    }

    @PostMapping("/add")
    public void registerNewCustomer(@RequestBody Customer customer) {
        customerService.addNewCustomer(customer);
    }

    @DeleteMapping("/delete")
    public void deleteCustomerByEmail(@RequestParam(name = "email") String email) {
        customerService.deleteCustomerByEmail(email);
    }
}
```

Figure 31: CustomerController class



```java
public class FuelLogController {
    @GetMapping
    public ResponseEntity<List<FuelLog>> getAllFuelLogs() {
        List<FuelLog> logs = fuelLogService.getAllFuelLogs();
        return ResponseEntity.ok(logs);
    }

    @PostMapping(consumes = "application/json")
    public FuelLog saveFuelLog(@RequestBody FuelLog fuelLog) {
        System.out.println("FuelLogController.saveFuelLog: fuelLog = " + fuelLog);
        return fuelLogService.saveFuelLog(fuelLog);
    }

    @GetMapping("/{vehicleId}")
    public ResponseEntity<List<FuelLog>> getFuelLogsByVehicleId(@PathVariable Long vehicleId) {
        List<FuelLog> logs = fuelLogService.getFuelLogsByVehicleId(vehicleId);
        return ResponseEntity.ok(logs);
    }

    @GetMapping("/log/{fuelLogId}")
    public ResponseEntity<FuelLog> getFuelLogById(@PathVariable Long fuelLogId) {
        return fuelLogService.getFuelLogById(fuelLogId)
                .map(ResponseEntity::ok)
                .orElse(ResponseEntity.notFound().build());
    }

    // get latest odometer by vehicle id
    @GetMapping("/latest/{vehicleId}")
    public ResponseEntity<FuelLog> getLatestOdometerByVehicleId(@PathVariable Long vehicleId) {
        FuelLog fuelLog = fuelLogService.getLatestOdometerByVehicleId(vehicleId);
        return ResponseEntity.ok(fuelLog);
    }

    @GetMapping("/fuel-logs-with-vehicle")
    public List<FuelLogWithVehicleDto> getFuelLogsWithVehicle() {
        // Fetch all fuel logs joined with vehicle information
        return fuelLogService.getAllFuelLogsWithVehicle();
```

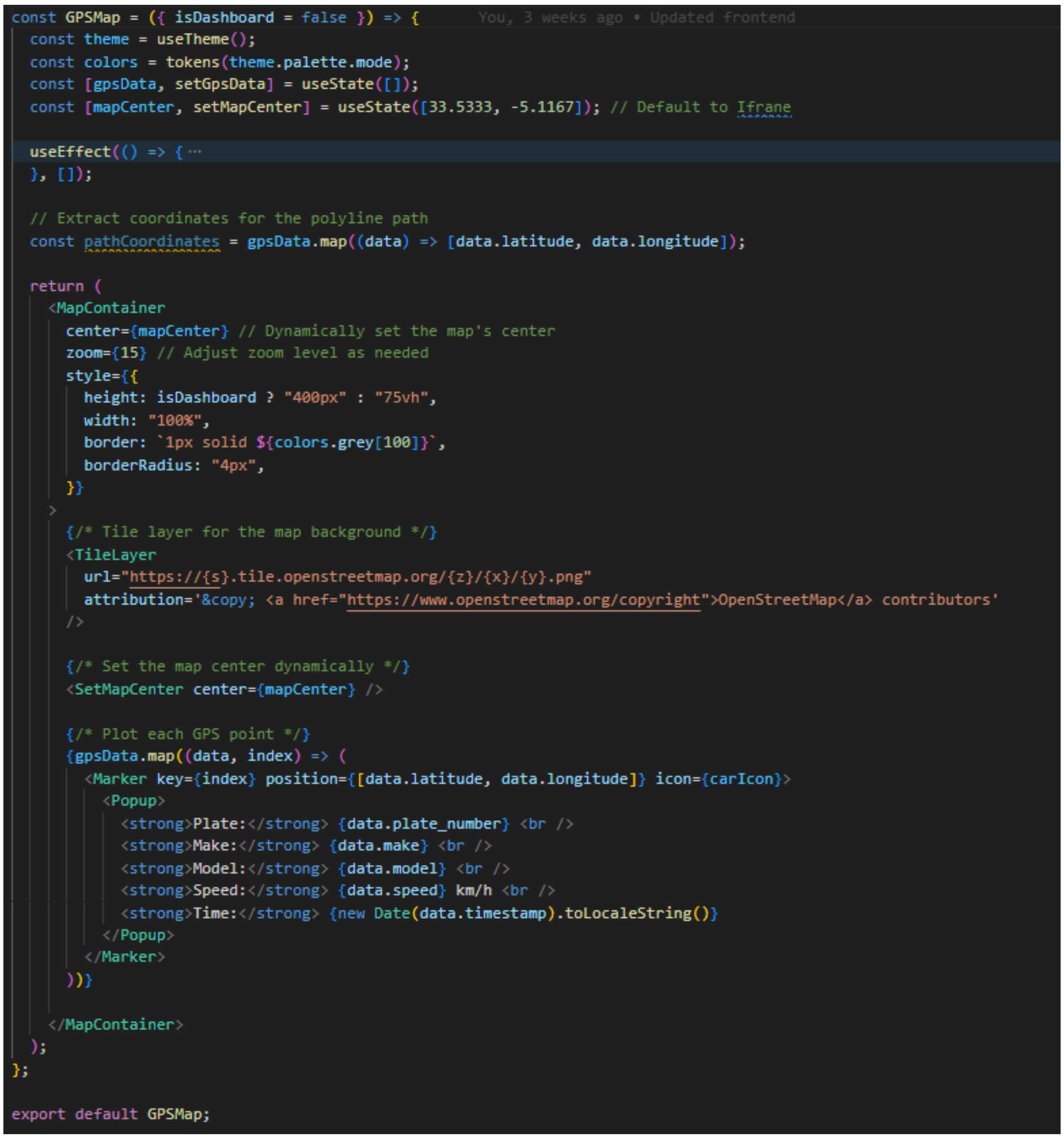
Figure 32: FuelLogController class



```java
public class TeltonikaDataService {
    public Map<String, Object> getMaintenanceAlertsWithOdometer(Long vehicleId) {

        System.out.println("Vehicle ID: " + vehicleId);
        // Fetch the latest odometer value from fuel_logs
        Long currentOdometer = fuelLogRepository.findLatestOdometerByVehicleId(vehicleId)
                .orElse(other:0L); // Default to 0 if no logs exist
        List<Maintenance> maintenanceList = maintenanceRepository.findByVehicleId(vehicleId);
        // Thresholds for maintenance
        Map<String, Long> thresholds = new HashMap<>();
        thresholds.put(key:"Oil Change", value:10000L);
        thresholds.put(key:"Tyre Change", value:50000L);
        thresholds.put(key:"Distribution Change", value:150000L);
        // Generate alerts and update thresholds dynamically
        for (Maintenance maintenance : maintenanceList) {
            String operationType = maintenance.getOperationType();
            String alert = "No Alert";
            if (operationType != null && thresholds.containsKey(operationType)) {
                Long threshold = thresholds.get(operationType);
                Long nextThreshold = maintenance.getMaintenanceDate() != null
                        ? currentOdometer + threshold // Set the next threshold
                        : threshold;
                if (currentOdometer >= nextThreshold) {
                    alert = operationType + " Needed!";
                }
                // Update maintenance alert and calculate remaining km
                maintenance.setAlert(alert);
            }
        }
        // Prepare response
        Map<String, Object> response = new HashMap<>();
        response.put(key:"currentOdometer", currentOdometer);
        response.put(key:"maintenanceList", maintenanceList);
        return response;
    }
```

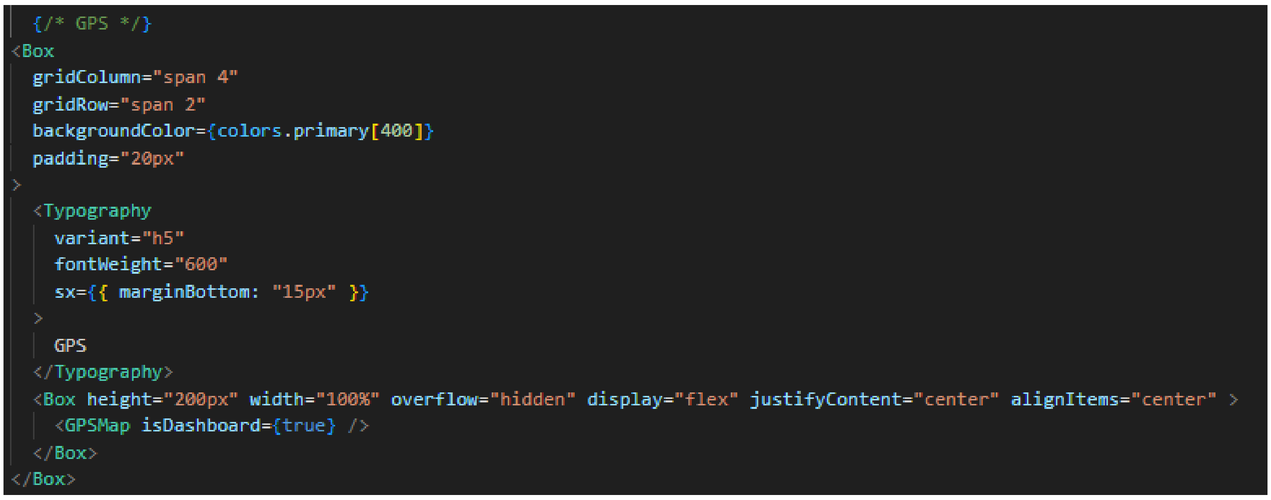
Figure 33: Fetch the Odometer Maintnance Altert

Frontend Code



```
const GPSMap = ({ isDashboard = false }) => {        You, 3 weeks ago • Updated frontend
  const theme = useTheme();
  const colors = tokens(theme.palette.mode);
  const [gpsData, setGpsData] = useState([]);
  const [mapCenter, setMapCenter] = useState([33.5333, -5.1167]); // Default to Ifrane

  useEffect(() => {…
  }, []);

  // Extract coordinates for the polyline path
  const pathCoordinates = gpsData.map((data) => [data.latitude, data.longitude]);

  return (
    <MapContainer
      center={mapCenter} // Dynamically set the map's center
      zoom={15} // Adjust zoom level as needed
      style={{
        height: isDashboard ? "400px" : "75vh",
        width: "100%",
        border: `1px solid ${colors.grey[100]}`,
        borderRadius: "4px",
      }}
    >
      {/* Tile layer for the map background */}
      <TileLayer
        url="https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png"
        attribution='&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors'
      />

      {/* Set the map center dynamically */}
      <SetMapCenter center={mapCenter} />

      {/* Plot each GPS point */}
      {gpsData.map((data, index) => (
        <Marker key={index} position={[data.latitude, data.longitude]} icon={carIcon}>
          <Popup>
            <strong>Plate:</strong> {data.plate_number} <br />
            <strong>Make:</strong> {data.make} <br />
            <strong>Model:</strong> {data.model} <br />
            <strong>Speed:</strong> {data.speed} km/h <br />
            <strong>Time:</strong> {new Date(data.timestamp).toLocaleString()}
          </Popup>
        </Marker>
      ))}

    </MapContainer>
  );
};

export default GPSMap;
```

Figure 34: GPS Component



```
  {/* GPS */}
<Box
  gridColumn="span 4"
  gridRow="span 2"
  backgroundColor={colors.primary[400]}
  padding="20px"
>
  <Typography
    variant="h5"
    fontWeight="600"
    sx={{ marginBottom: "15px" }}
  >
    GPS
  </Typography>
  <Box height="200px" width="100%" overflow="hidden" display="flex" justifyContent="center" alignItems="center" >
    <GPSMap isDashboard={true} />
  </Box>
</Box>
```

Figure 35: GPS Representation on the Dashboard

APPENDIX C: CONFERENCE PROPOSAL

Abstract--- Nowadays, integration of the Internet of Things (IoT) devices into the automotive industry is a game changer especially when it comes to fleet management solutions, providing new opportunities for vehicle monitoring and management. This paper primarily emphasizes on how a secure dashboard is designed for data acquisition and analysis of vehicles fitted with Teltonika FM5300. The FM5300 is a GPRS based GPS tracking device installed on vehicles which access sensors which will be shown on a dashboard aimed at consolidating, decoding and displaying information on real-time parameters such as location, speed, fuel level, and maintenance statuses in a user-friendly format. Primarily, the purpose of the dashboard is to offer fleet management tracking in a real-time fashion so as to provide operators insight into the monitoring of their vehicles. For that, the system contains a Python designed gateway that translates absolute telemetry data encoded by FM5300 in its communication over TCP/IP. The data backed up by this communication once compressed is sent out and uploaded to a Postgres database making it easy to retrieve and query the information. The interface has connected a map that indicates whereabouts of the fleets for easier planning while also windows that give views of detail reports on the vehicle statistics. This project will also be examination the communication architecture with regards to the data transfer and exhibition from FM5300 through GPRS along the interpretation of data encoding security measures that are integrated within the structure.