School of Science and Engineering

"FINEK": SMART VEHICLE TRACKING SYSTEM FOR FLEET MANAGEMENT

Capstone Design Report

FALL 2024

Alaaeddine Elhaoua

Supervised by:

Dr. Oumaima Hourrane

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: **Alaaeddine Elhaoua**

Supervisor's name and signature: **Oumaima Hourrane**

"FINEK": SMART VEHICLE TRACKING SYSTEM FOR FLEET MANAGEMENT

Capstone Report

Student Statement:

This project has been an amazing learning experience for me. It gave me the chance to explore the development of a vehicle tracking system in a way that combined everything I've learned over the past four years at Al Akhawayn University. It really helped me put the theory into practice.

Throughout this journey, I got to try out new tools, frameworks, and technologies, which was both exciting and challenging. Beyond the technical skills, this project pushed me to improve my problem-solving abilities and sharpen my time management skills, things that turned out to be just as important as the coding itself.

Looking back, this project feels like a big step forward in both my academic and personal growth. I'm grateful for the opportunity, and I know the lessons I've learned here will guide me in my career in the future.

Alaaeddine Elhaoua

Approved by the Supervisor

Dr. Oumaima Hourrane

ACKNOWLEDGEMENTS

I am truly grateful for the incredible journey I have had at AUI, a journey that has shaped me both academically and personally. Working on this project has been an exciting challenge, one that pushed me to apply everything I have learned, stay curious, and constantly seek out new knowledge. It has been a rewarding experience, and I am so proud of how far I have come.

I want to sincerely thank my capstone supervisor, Dr. Oumaima Hourrane, for her support, thoughtful advice, and guidance through every step of this process. Her encouragement made all the difference. I am also incredibly thankful to my professors, who gave me the tools and knowledge to bring this project to life.

A huge thank you goes to Dr. Amine Abouaomar for inspiring this project and for answering my many questions along the way. His input was invaluable. I am also endlessly grateful to my family, who stood by me every step of the way, cheering me on and showing so much patience as I navigated the ups and downs of this journey.

And, of course, to my amazing friends, thank you for keeping me motivated, for listening to me talk about this project non-stop, and for always encouraging me, even when you did not fully understand what I was saying. A special shoutout to Mohammed Moubaraki for letting me use his home router to setup my project. Thank you all for making this experience unforgettable.

CONTENT

LIST OF FIGURES

ABSTRACT (ENGLISH)

This project suggests the development of a vehicle tracking system using Teltonika FMB130 devices integrated with a web application for real-time data visualization. The system will be designed to improve fleet management for companies by tracking speed, location, and some other vehicle metrics. The project aims to enhance efficiency, reduce maintenance costs, and improve safety through intelligent data analysis and reporting.

Keywords: Vehicle Tracking, Fleet Management, Data Analysis, Teltonika FMB130, MySQL.

RESUME (FRENCH)

Ce projet propose le développement d'un système de suivi des véhicules utilisant des appareils Teltonika FMB130 intégrés à une application web pour la visualisation des données en temps réel. Le système sera conçu pour améliorer la gestion de la flotte des entreprises en suivant la vitesse, la localisation et d'autres paramètres du véhicule. Le projet vise à améliorer l'efficacité, à réduire les coûts de maintenance et à améliorer la sécurité grâce à une analyse intelligente des données et à l'établissement de rapports.

Mots clés: Suivi de véhicules, gestion de flotte, analyse de données, Teltonika FMB130, MySQL.

1. Introduction

The demand for advanced vehicle tracking applications and solutions has increased, particularly in car rental services and logistics companies, where operational efficiency is necessary. This project focuses on leveraging the Teltonika FMB130, a GPS tracker with rich features, to enhance vehicle monitoring for companies. The aim is to not only track location but also driving behavior and vehicle condition, such as speed, acceleration, and detecting accidents.

2. Problem Statement

Managing a fleet of vehicles efficiently is a challenge for many businesses. Fleet managers need real-time information on vehicle locations, fuel levels, speeds, and engine status to make quick decisions, reduce costs, and keep their operations running smoothly. Traditional tracking systems often fall short, providing only basic information and lacking the reliability needed for real-time monitoring. This project, "Finek: Smart Vehicle Tracking System for Fleet Management," seeks to bridge that gap by using the Teltonika FMB130 device to deliver detailed, continuous vehicle data through a user-friendly web interface, empowering fleet managers to track and manage their vehicles more effectively.

3. Project Specifications

3.1. Hardware: Teltonika FMB130

A web application setup with a backend that receives GPS and other data from devices connected to vehicles. It will allow the user to track their own vehicles in real time. The project will consist of the following components:

The FMB130 will serve as the GPS tracking device, collecting data such as longitude, latitude, altitude, speed, acceleration, and total distance travelled. It will send this data over GSM to a central gateway.

3.2. Software Architecture

The system will include a gateway server developed with Python sockets, which will manage data received from the Teltonika devices. Data will be processed, stored in a MySQL database, and made available for the backend developed with Django, which would serve it to the ReactJS front end.

3.3. Data Visualization and User Interface

A web application will be developed to provide an interactive map that displays the location and metrics of the entire vehicle fleet. The map will allow users to track multiple cars simultaneously and view detailed reports on driving behavior and trip summaries.

4. STEEPLE Analysis

Social: (impact of the project) It would help promote trust and transparency between the vehicle owners and the drivers.

Technological: (impact on the project) Rapidly evolving GPS and IoT technologies may require continuous updates to keep the project system efficient.

Environment: (impact of the project) by monitoring and analyzing driving behavior and fuel consumption across many vehicles, we can find and promote the most sustainable driving practices which would lead to reduced emissions.

Economic: (impact of the project) The car owners will be able to improve maintenance scheduling based on the gathered data, which ultimately will reduce repair costs.

Political: (impact on the project) The government can change import laws and procedures, which can impact the availability of the Teltonika devices used for vehicle tracking.

Legal: (impact on the project) Laws related to data storage and collection, especially for location data, should be followed which would increase the complexity of the project.

Ethical: (impact on the project) The project should follow ethical standards especially for collecting location data, otherwise it would damage the reputation of the companies that would use it.

5. Engineering Standards

ISO/IEC 23026:2023; This standard is related to the engineering and management of websites for systems and software, and since I will be developing a web application in my capstone, it will be relevant.

ISO 26000; This ISO standard gives guidance on how to act and operate ethically, specifically in my case to collect and use data ethically.

6. Logic Model Framework

6.1. Target Population

The target population for this project is fleet managers, car rental companies and logistics businesses that require real time data about their vehicles' locations, fuel consumption and driver behavior. This population can also be extended to government departments such as emergency services (ambulances, fire trucks, police cars…) for faster and more efficient dispatch. Moreover, it can also work on governmental and municipal fleets to make sure government resources are not used for personal purposes.

6.2. Underlying Assumptions

While planning for this project and going through its development I have several assumptions. The transmitted data is assumed to be accurate and consistent, which is essential for this project. Additionally, the devices used are assumed to be durable and require minimal maintenance minimizing downtime. Also, better fleet management is assumed to lead to reduced fuel consumption and emissions contributing to environmental and financial goals.

6.3. Resources/Challenges

6.3.1. Resources

The resources needed to complete this project are categorized into hardware and software. For the hardware, I need Teltonika FMB devices and a server with static external IP. As for software, I need to use Python3, Django, MySQL, React.

6.3.2. Challenges

The project has some challenges to overcome, such as being able to scale to accommodate additional vehicles and data as the fleet grows. Another challenge is the cost of the implementation being tightly related to the prices of the Teltonika devices, which can be impacted by import fees.

6.4. Activities

These are the main activities to be done to fulfill the project goals. These activities include setting up a server with a static external IP to receive data from the Teltonika device, and setting up the TCP socket gateway for data transmission and preprocessing in the server. It is also needed to set up the database to store the data within the server, implement the backend to serve APIs with the stored data, and develop the front end interface/dashboard to visualize vehicle information.

6.5. Outputs of the Project

The output of the project would be a web application where users can monitor vehicle data. It would also have a dashboard with vehicle status and metrics.

Another output is a database containing historical data on vehicle movements, fuel usage, and other data that can be used to create a driver behavior scoring model using machine learning.

6.6. Outcomes

The benefits that would occur as a result of this project can be split into short term, intermediate, and longs term outcomes:

6.6.1. Short Term Outcomes

On the short term, fleet managers will gain real-time insights into vehicle locations, speed, and fuel levels enabling quicker responses to issues.

6.6.2. Intermediate Outcomes

With access to real time data, companies can streamline scheduling, reduce idle times, and optimize routes saving time and fuel. By reducing fuel wasting, unnecessary maintenance, and preventing excessive wear on vehicles, companies can cut operational costs. Also, monitoring driver behavior can help identify risky practices, leading to better enforcement of safety policies and training for drivers..

6.6.3. Long Term Outcomes

On the long run, access to historical data allows companies to make more informed strategic decisions about their fleets, staffing and resources. By optimizing routes and reducing fuel usage, the company can minimize their carbon emissions. Additionally, improved operational efficiency would translate to faster deliveries and better service, enhancing customer trust and loyalty.

7. Existing Solutions

Existing fleet management solutions using GPS tracking devices in Morocco commonly focus on providing basic vehicle location and speed data. While these features are essential for tracking fleet movement, they often lack deeper insights into driver behavior, such as harsh braking, rapid acceleration, or idle time. Without this information, fleet managers miss out on valuable metrics that could help improve driver safety, reduce fuel consumption, and lower maintenance costs. This project aims to fill that gap by integrating driver behavior analysis, offering a more comprehensive solution for fleet monitoring.

As for the technologies behind them, many don’t share those details publicly. But it’s likely they relational databases to manage and store location data. These databases are usually located in physical server racks they offer with their solutions. On the front end, they might use tools like React or Angular to create dashboards that make it easy for users to view maps and track vehicles, since they usually have web applications.

8. Methodology & Capstone Design

8.1.Capstone Design & Application Development

For this capstone project, I am working in an Agile setup to keep things flexible, efficient, and responsive to any challenges that come up. By breaking the project into smaller, manageable pieces and working in short, focused durations, I was able to make steady progress, adapt to new information, and get quick feedback along the way.

In my Agile process, I start each sprint with a clear set of goals to make sure I know exactly what I am working towards. The goals that were achieved so far are:

8.1.1. Initial Planning and Setup

In the first sprint, I focused on laying the groundwork. I made a component diagram of how the project would be set up (figure 1), and an ERD (Entity Relationship Diagram) for the database (figure 2).

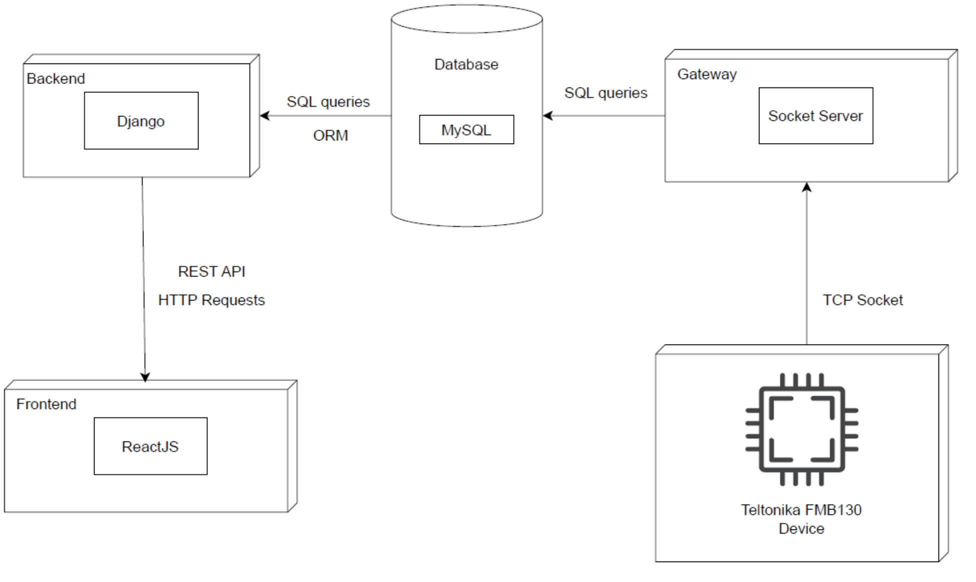

figure 1: Component Diagram

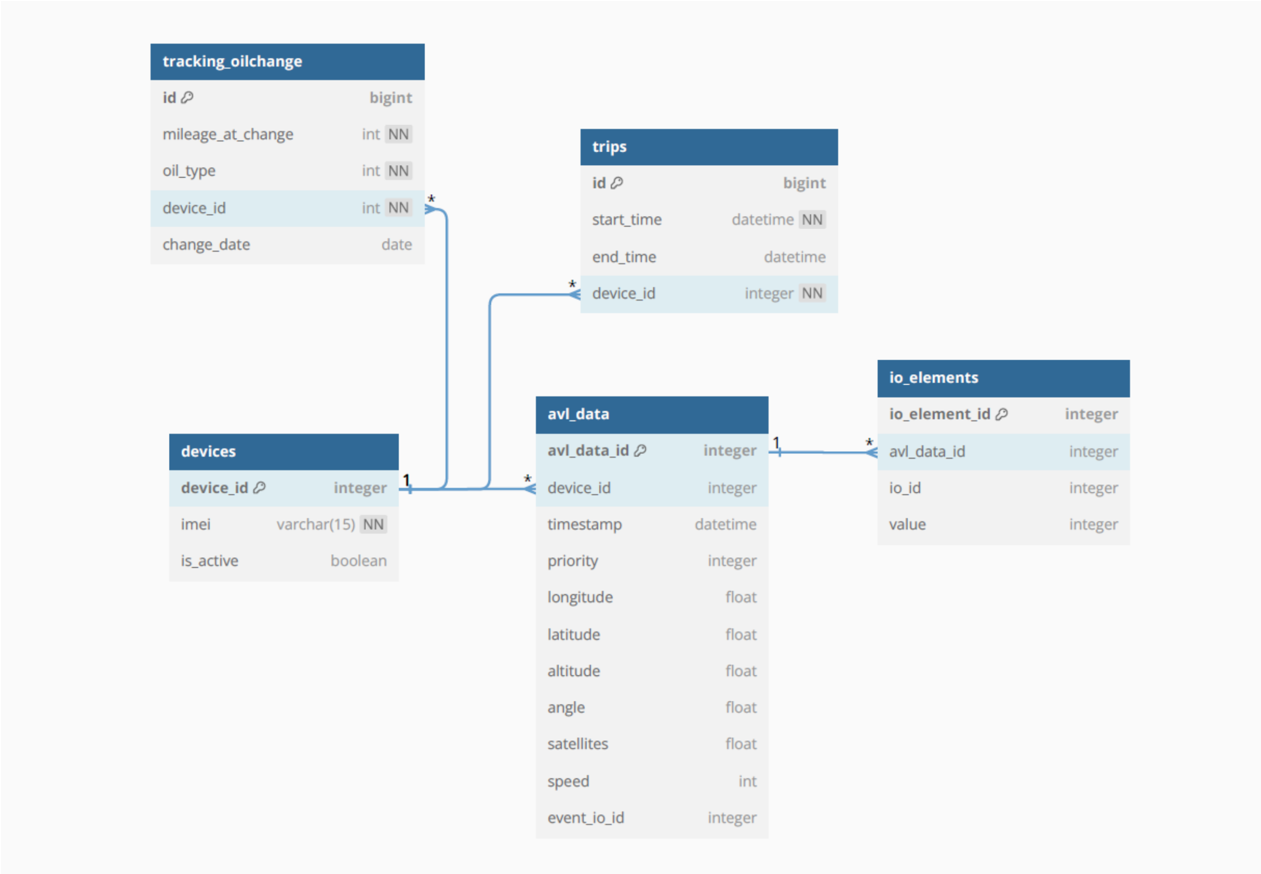

figure 2: Entity Relationship Diagram

8.1.2. Server Setup

In the second sprint, I tackled the foundation of the project by setting up a reliable server environment. Using a Raspberry Pi, I installed the Ubuntu Server operating system, which provided a stable platform for all the project components. To ensure consistent access within my local network, I assigned a static internal IP to the Raspberry Pi. However, since obtaining a static external IP wasn't feasible, I turned to NoIP.com, a dynamic DNS service, which gave me a free domain name that automatically updates whenever my external IP changes. This solution ensured that the server could consistently communicate with external devices like the Teltonika FMB130, making it accessible from anywhere.

8.1.3. Gateway Setup

After setting up the server, I worked on the gateway application, which acts as the middleman between the hardware and the database. This code, written in Python and running on the server, listens for data sent by the Teltonika devices. When data packets arrive, the gateway parses them to extract meaningful information such as GPS coordinates, speed, and timestamps. Once the data is structured, it's stored in the MySQL database. This gateway plays a vital role in ensuring smooth and efficient data handling, connecting the physical device to the backend systems.

8.1.4. Database Creation

To manage the incoming data effectively, I designed a MySQL database tailored to the project's needs. This involved creating tables for key entities like devices, trip data, and vehicle metrics, and defining their relationships. For example, each trip is tied to a specific device, and vehicle metrics are recorded and stored in relation to trips. Structuring the database this way made it easy to retrieve and analyze data, whether it's for real-time use on the dashboard or for historical records.

8.1.5. Hardware Configuration

I installed the Teltonika FMB130 device in my car and configured it to send data to the server. Using Teltonika's configurator app, I set the device to transmit data to the server's domain name on a specified port, allowing it to share real-time information like location, speed, and vehicle status. This wasn't a smooth process; I faced several challenges, including misconfigurations and data transmission issues. But in the end, I resolved these bugs and ensured the device communicated seamlessly with the server.

8.1.6. Backend Development

The next step was building the backend using Django, which handles data requests and responses. This is where all the raw data from the database gets transformed into usable insights. For instance, APIs fetch daily mileage data, oil change history, and vehicle status to serve the frontend. Hosting the backend directly on the Raspberry Pi alongside the database minimizes delays and ensures smooth data flow. Building the backend not only involved handling data but also formatting it in a way that made it easier to visualize on the dashboard.

8.1.7. Frontend Development

Finally, I focused on creating a user-friendly dashboard using ReactJS. This is where all the data comes to life. The dashboard includes sections like vehicle status, mileage, oil change history, and a map that displays trips and the car's last known location. I used the Leaflet library for the map, which allowed me to dynamically render GPS data with interactive markers and paths. For charts, I turned to Chart.js, which provided nice interactive graphs to show trends like daily mileage. Developing the frontend was especially rewarding, as it made the data not just accessible but also visually engaging for the user.

9. Feasibility Study

9.1. Technical Feasibility

This project combines hardware and software components to create a reliable vehicle tracking system. It required configuring the Teltonika FMB130 device, setting up a Raspberry Pi server, and developing a backend with Django and a frontend using ReactJS. The technical challenges, like real-time data parsing and ensuring smooth communication between the device and server, were resolved with consistent testing and debugging. Using well-established tools and technologies ensures the system is not only functional but also scalable for future improvements.

9.2. Financial Feasibility

The project was designed to be budget-friendly. By leveraging a Raspberry Pi as a server and relying on free or open-source tools like Django, ReactJS, Leaflet, and Chart.js, costs were kept low. Services like NoIP.com eliminated the need for expensive static external IPs, further reducing expenses. This approach makes the system accessible for small to medium-sized operations, offering significant value in terms of efficiency and vehicle management without requiring a major financial investment.

9.3. Time Feasibility

The available timeline was sufficient to develop a prototype, from hardware setup to user interface design. The milestones were achievable within the timeframe, ensuring a functional system was delivered. With additional time and resources, the project can be expanded to include advanced features or support larger-scale deployments.

10. Data Presentation

The data collected and displayed by the system plays a vital role in understanding how the vehicle is being used, ensuring timely maintenance, and making informed decisions about its operations. The dashboard provides a user-friendly interface to access this information in a way that feels intuitive.

10.1. Collected Data:

The system collects and organizes various types of data that provide insights into the vehicle's performance, usage, and maintenance needs.

10.1.1. Trips Data

Captures the start and end times of trips.

Example:


```json
{
    "id": 32,
    "device": 2,
    "start_time": "2024-11-18T10:30:40Z",
    "end_time": "2024-11-18T11:15:00Z"
}
```


10.1.2. IO Data

The IO-Elements table stores different elements like the mileage of vehicles, the vehicle's ignition and movement status. Each element is identifiable by its io_id; for example an io_id of 16 indicates mileage.

Example:


```json
{
    "io_element_id": 2001,
    "io_id": 16,
    "value": 50158054,
    "avl_data": 292
}
```

10.1.3. AVL Data

The AVL table captures the location and speed of vehicles.

Example:

```
{
    "avl_data_id": 2091,
    "device": {
      "device_id": 2,
      "imei": "352016704202811",
      "is_active": true
    },
    "timestamp": "2024-11-08T14:39:18Z ",
    "priority": 0,
    "longitude": -5.130935,
    "latitude": 33.5219933,
    "altitude": 1676,
    "angle": 10,
    "satellites": 14,
    "speed": 43,
    "event_io_id": 0
  },
```

10.1.4. Oil Change Data

Saves a history of oil change dates and the latest change with how many kilometers are left until an oil change should be scheduled.

Example:

```
{
  "latest": {
    "last_change_date": "2024-11-27",
    "mileage_at_change": 50000,
```

```
        "oil_type": 5000,
        "device": 2,
        "kilometers_left": 4071.944000000003,
        "warning": false
    },
    "all_changes": [
        {
            "id": 10,
            "device": 2,
            "last_change_date": "2024-11-27",
            "mileage_at_change": 50000,
            "oil_type": 5000
        },
        {
            "id": 13,
            "device": 2,
            "last_change_date": "2024-06-08",
            "mileage_at_change": 40000,
            "oil_type": 10000
        }
    ]
}
```


10.1.5. Daily Mileage Data

Keeps track of daily mileage of vehicles by programmatically calculating it from the vehicle mileage at the end of each day.

Example:


```json
{
    "date": "2024-11-12",
    "device": 2,
    "distance": 27.917
}
```


10.2. Displayed Data:

10.2.1. Vehicle Status

The system provides real-time insights into the vehicle's condition, showing whether it's currently on or off, and whether it's moving or stationary. Simple visual indicators, like green and red dots, make it easy to grasp the vehicle's status at a glance. This feature is especially helpful for quickly checking on the vehicle without needing to sift through data.

10.2.2. Mileage Tracking

Mileage is presented in an easy-to-read format, showing how many kilometers the vehicle has traveled. This data is not only useful for monitoring usage but also for scheduling maintenance or estimating operational costs. Over time, patterns and trends in mileage can also be analyzed to better understand how the vehicle is being utilized.

10.2.3. Oil Change Records

Keeping up with oil changes is essential for the health of any vehicle. The dashboard tracks when the last oil change occurred, the mileage at the time, and the type of oil used. It even includes alerts to remind users when the next oil change is approaching. This feature ensures the vehicle stays in top condition and avoids unnecessary damage.

10.2.4. Daily Mileage Insights

The system breaks down how much the vehicle travels each day and displays this information in a clean and interactive bar chart using Chart.js. This visualization helps spot trends, such as high-usage days or periods of inactivity, making it easier to optimize vehicle scheduling and operations.

10.2.5. Trip Visualization

Using Leaflet, the dashboard provides a dynamic map that shows the routes the vehicle has taken and its last known location. Markers indicate start and end points, and users can view details like timestamps and speeds along the way. This feature is great for visualizing trip patterns and ensuring everything is on track.

10.2.6. Interactive Design

All of this data is brought together in a user-friendly dashboard built with ReactJS. The interface is designed to be simple and responsive, allowing users to quickly filter, view, and analyze the information they need. It’s built to feel natural and intuitive, even for someone who isn’t deeply familiar with technical systems.

By combining real-time data with intuitive visuals, the system transforms raw information into something meaningful and actionable.

11. Results

This project resulted in a fully functional vehicle tracking and monitoring system that combines real-time data collection, a powerful backend, and an intuitive dashboard. Below are some key features of the application, highlighted with screenshots to give a clearer picture of its functionality.

11.1. Dashboard Overview

The dashboard is designed as the central hub for all key vehicle metrics, such as mileage, vehicle status, and maintenance history. It provides a clean and user-friendly interface for monitoring everything at a glance.

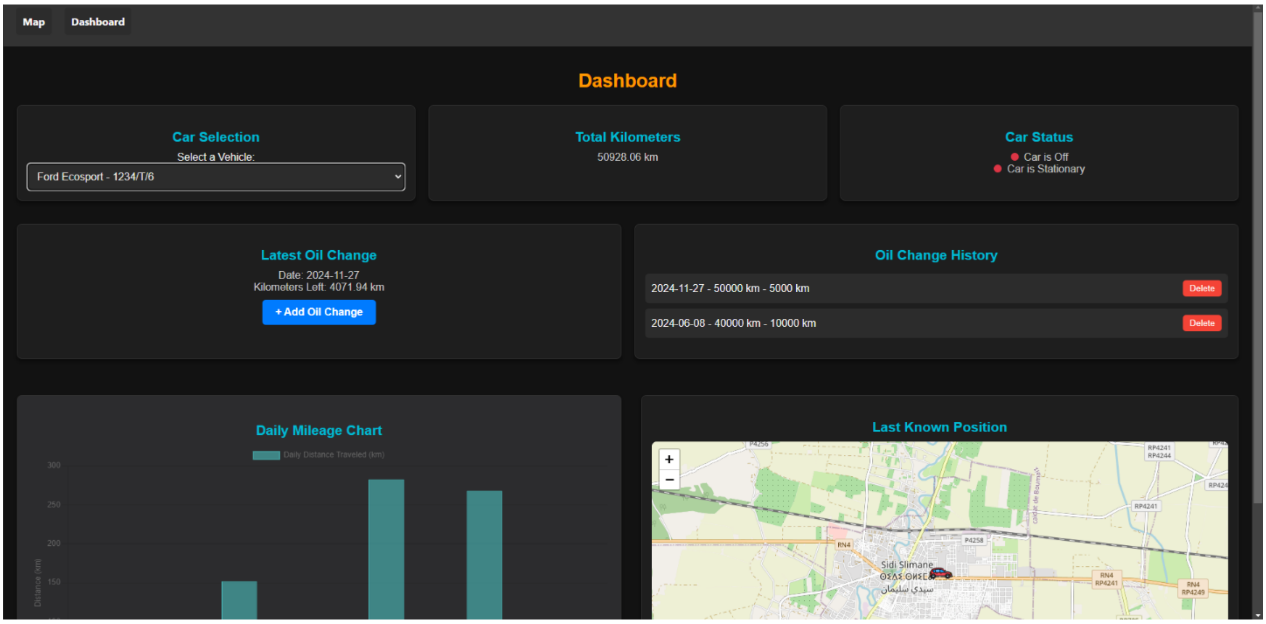

figure 3: Dashboard View

11.2. Daily Mileage Chart

The application features a bar chart that visualizes the vehicle's daily mileage. This makes it easy to spot usage patterns and understand how far the vehicle travels on different days.

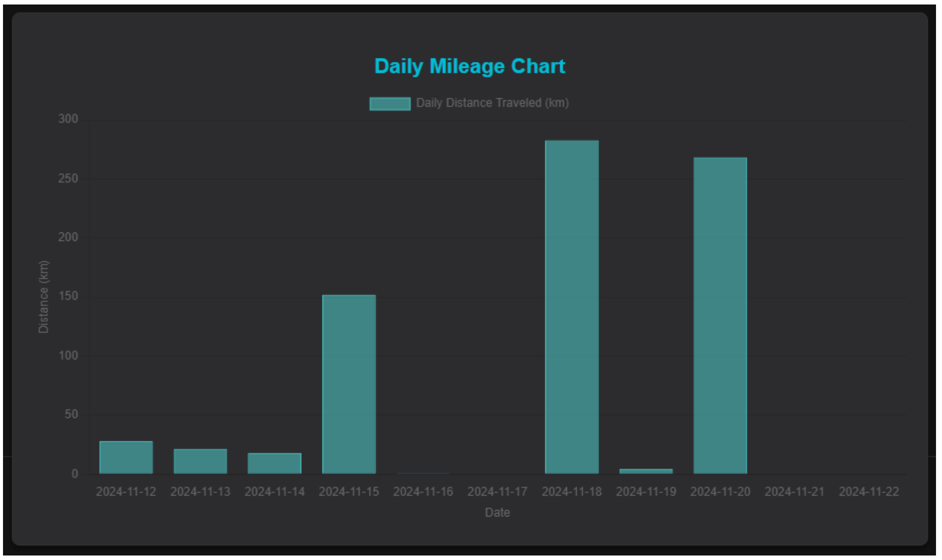

figure 4: Daily Mileage Chart

11.3. Map Integration

One of the most exciting features is the live map that shows the vehicle's last known location. This map is integrated directly into the dashboard, offering an intuitive way to check where the car is and track recent movements.

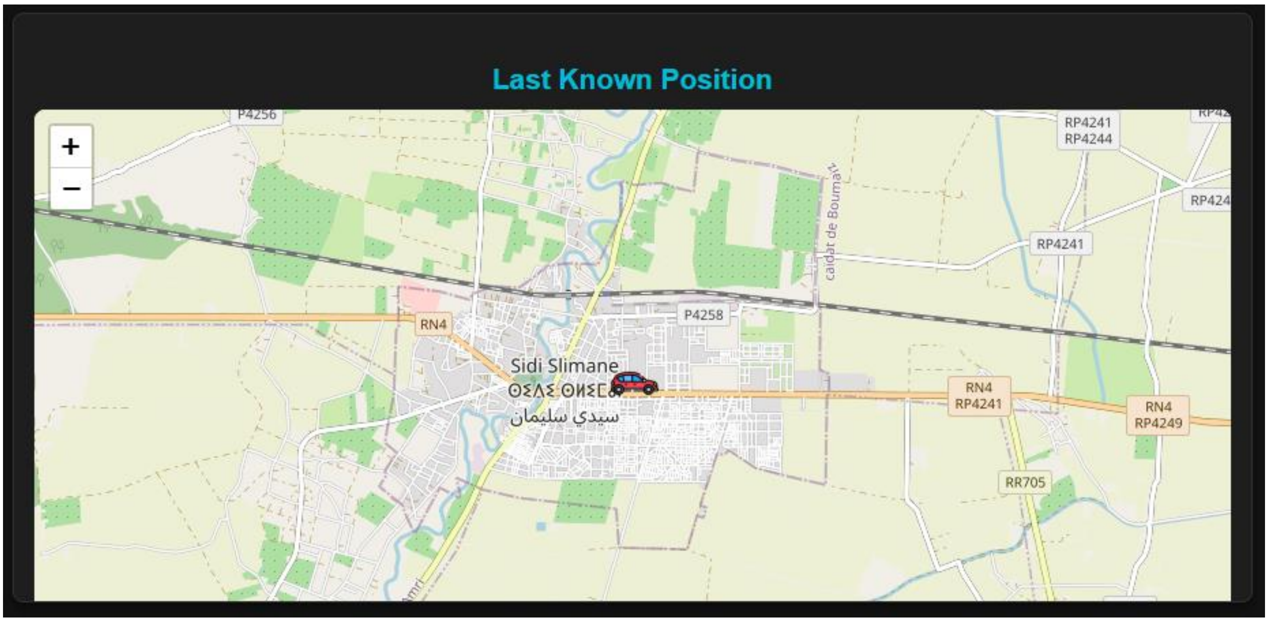

figure 5: Last Known Position / Current Position

11.4. Oil Change Management

With the oil change management feature, users can see the last oil change details, a history of previous changes, and even add new entries directly from the dashboard.

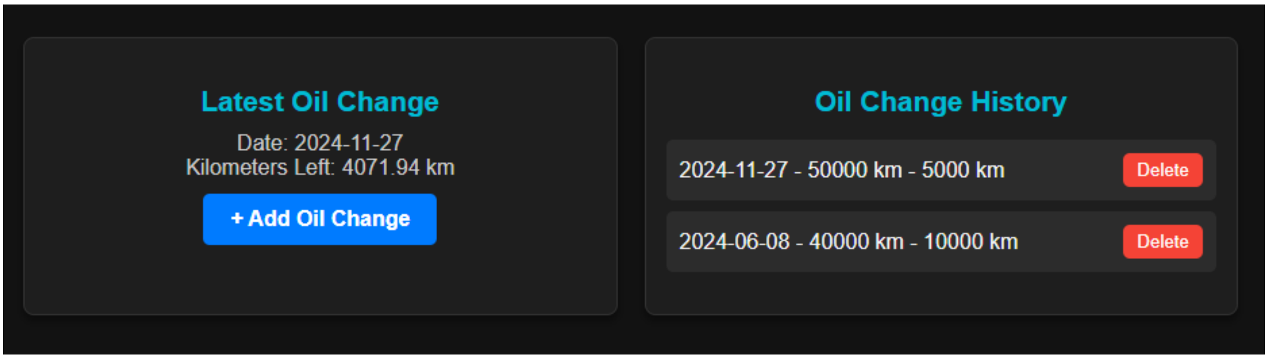

figure 6: Oil Change Interface

11.5. Map Page

The map view allows the user to first select the car, which shows its last known location/current location. It also allows to search for trips by date.

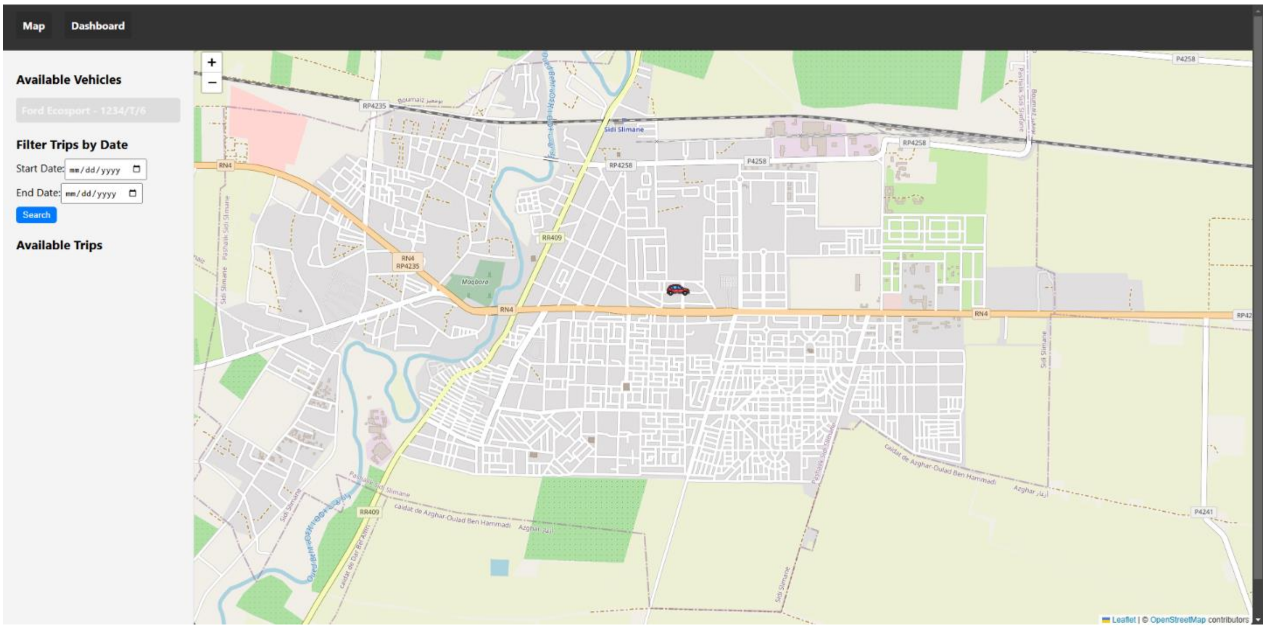

figure 7: Map Page

When a trip is selected, the map will display the starting and ending point of the trip. The user can also hover over the path at any point to see what the speed of the car in that place was.

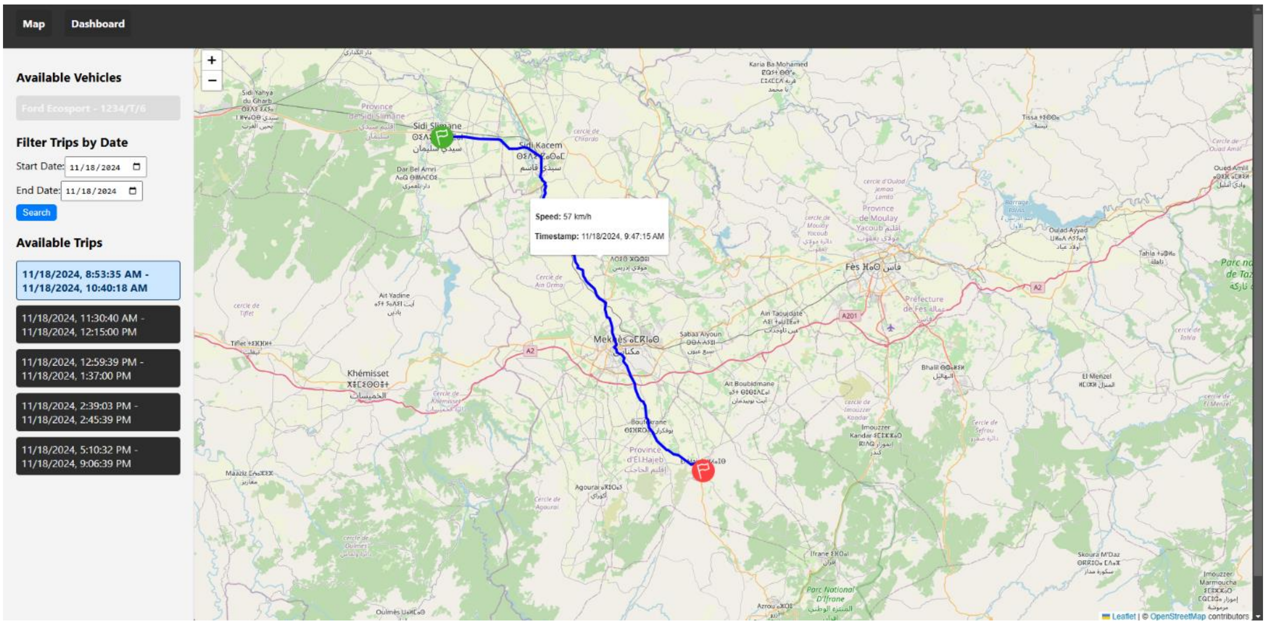

figure 8: Trip Information

12. Learning Strategies

12.1. Future Work

One of the most exciting directions for this project is leveraging the collected data to analyze driver behavior. By studying patterns such as abrupt accelerations, frequent braking, or high-speed trends, the system can provide valuable insights into driving habits. Here are some of the ideas I would like to explore in the future:

Driver Behavior Analysis:

Using the data collected from vehicles, I aim to develop modules that can identify risky driving behaviors. These insights could be used to provide feedback to drivers or even suggest safer driving practices.

Scoring System for Drivers:

Implementing a scoring system based on driving habits, such as smoothness, following speed limits, and efficiency, could help drivers improve their behavior over time. This feature could be particularly useful for businesses managing larger fleets, like Jumia and Amana.

Incident Detection and Alerts:

Enhancing the system to detect events like hard braking, collisions, or excessive speeding in real-time and sending immediate alerts to relevant parties could make the system more dynamic and safety-focused.

Driver Training Recommendations:

Based on behavior analysis, the system could recommend personalized training programs or tips to drivers to encourage better practices.

Data-Driven Reports for Insurance or Fleet Management:

Compiling data into reports that highlight driving trends, incidents, and improvements could be useful for insurance companies or fleet managers looking to monitor and improve driver safety.

12.2. Growing My Skills

To achieve these goals, I plan to focus on several key areas for personal and professional growth:

Deepening Knowledge in Data Science and Machine Learning:

I will explore advanced data analysis techniques and machine learning models that can process large datasets and extract meaningful insights. Learning libraries like TensorFlow, Scikit-learn, and Pandas will be crucial for this.

Exploring Advanced IoT Applications:

I want to explore how IoT systems can be optimized for real-time event detection and how they can integrate seamlessly with machine learning pipelines.

Improving Visualization Techniques:

Presenting driver behavior data in a clear and intuitive manner is just as important as collecting it. I will focus on improving my skills in data visualization libraries and tools to create impactful dashboards.

Experimenting with Predictive Analytics:

Applying predictive analytics to anticipate risky behaviors or possible incidents could add another layer to the system. For this, I will need to learn more about predictive modeling.

By focusing on these areas, I can ensure that the system not only tracks vehicle data but also evolves into a comprehensive platform for promoting safer driving practices and enhancing road safety.

13. Conclusion

This project has been an incredible learning experience, combining what I have studied over the years with hands-on challenges to create a fully functional vehicle tracking and monitoring system. From setting up the hardware to designing and building a user-friendly application, each step pushed me to think critically and apply my skills in a real-world context.

The system has successfully achieved its goals, allowing for real-time tracking of vehicle data such as mileage, status, and history. The integration of maps and visual data representations, like charts, makes it intuitive and easy to use. These features not only provide insights into vehicle performance but also lay the groundwork for exploring driver behavior and safety enhancements.

Moving forward, this project offers so many exciting possibilities. From using the collected data to analyze driver habits to introducing advanced fleet management capabilities. It feels like just the beginning of what could be achieved.

Overall, this journey has been both challenging and rewarding. It’s not just about meeting the project objectives but also about the knowledge gained along the way, which I’m sure will guide me in future projects.

14. References

Efficient Fuel Management Solutions. (n.d.-a). Retrieved September 10, 2024, from https://teltonika-gps.com/use-cases/telematics/efficient-fuel-management-solutions

FMB130 first start - wiki knowledge base | Teltonika GPS. (n.d.-b). Retrieved October 2, 2024, from https://wiki.teltonika-gps.com/view/FMB130_First_Start

ISO 26000 - Social Responsibility. ISO. (2020, November 30). Retrieved September 15, 2024, from https://www.iso.org/iso-26000-social-responsibility.html

ISO/IEC/IEEE 23026:2023. ISO. (2023, September 1). Retrieved October 5, 2024, from https://www.iso.org/standard/81896.html

Jennings, N. (2023, November 16). Socket programming in python (guide). Real Python. Retrieved October 2, 2024, from https://realpython.com/python-sockets/

Making queries: Django documentation. Django Project. (n.d.). Retrieved September 29, 2024, from https://docs.djangoproject.com/en/5.1/topics/db/queries/