

**LICENCE PROFESSIONNELLE**

**Projet de Fin d'Études**

**Filière :** Intelligence Artificielle

**Thème :** LOCALISER ET CONTROLER LE DEMARRAGE ET L'ARRET DE SON VEHICULE A DISTANCE

**Rédigé et présenté par :**

*Hamza El ASLY | Ghita ESSALEK*

**Matricule :**

*758 | 735*

**Jury :**

**Encadré par :**

*Mr. Adnane TALHA*

**Année Universitaire 2024/2025**

# 

# Remerciements

Je remercie mon encadreur

**Mr. Adnane TAHA** pour son aide, son orientation et ses conseils durant l’accomplissement du projet. Tous mes infinis remerciements aux membres du jury d’avoir accepté de juger ce travail de fin d’études. Je remercie également tous les enseignants qui ont contribué à notre formation durant notre cursus SUP'MANAGEMENT.

Mes remerciements les plus chaleureux vont à mes chers parents pour leurs encouragements, leur patience et leur grand soutien durant toutes ces années d’études. Enfin, je remercie tous ceux qui ont participé de près ou de loin à la réalisation de ce mémoire.

SUMMER

1. [Remerciements 1](#_Toc202886843)
2. [Introduction 3](#_Toc202886844)
3. [Contexte et problématique 3](#_Toc202886845)
4. [Cahier des charges 4](#_Toc202886846)
5. [Architecture du système 4](#_Toc202886847)
6. [Simulation GPS et interface utilisateur 5](#_Toc202886848)
7. [Diagramme de Cas 6](#_Toc202886849)
8. [Stockage et analyse des données 8](#_Toc202886850)
9. [Réalisation technique 9](#_Toc202886851)
10. [Tests et résultats 12](#_Toc202886852)
11. [Conclusion 13](#_Toc202886853)

# Introduction

Dans un monde en constante évolution technologique, la gestion intelligente des véhicules devient une priorité. Notre projet s'inscrit dans cette dynamique en proposant une application logicielle capable de localiser un véhicule et de contrôler son démarrage et arrêt à distance. Ce projet est conçu dans un cadre purement logiciel, sans intégration de matériel physique, et s’appuie sur les compétences acquises en Intelligence Artificielle, traitement de données et interfaces utilisateur.

# Contexte et problématique

Dans un monde en constante évolution où la mobilité joue un rôle clé dans les activités quotidiennes et professionnelles, la sécurité et le contrôle des véhicules deviennent des enjeux majeurs. Le développement rapide des technologies connectées et de l’intelligence artificielle a ouvert la voie à des solutions novatrices dans le domaine de la gestion de flotte, du suivi GPS et du contrôle à distance.

Cependant, les solutions actuellement disponibles sur le marché sont souvent propriétaires, complexes à intégrer et financièrement inaccessibles pour les particuliers ou les petites entreprises. Ces outils sont majoritairement conçus pour des environnements professionnels avec des contraintes d’infrastructure matérielle (capteurs, modules GSM/GPS physiques), ce qui limite leur adoption à grande échelle.

La problématique que nous abordons dans ce projet est donc double :

1. **Comment offrir une solution logicielle légère, accessible et sans matériel pour le suivi et le contrôle d’un véhicule ?**
2. **Comment assurer l’interaction utilisateur dans un environnement multilingue et intuitif ?**

Notre objectif est de proposer une solution logicielle simulée, mais fonctionnelle, qui offre les fonctionnalités de base d’un système de contrôle de véhicule, notamment :

* Localisation en temps réel via coordonnées GPS simulées.
* Contrôle de démarrage et d’arrêt du véhicule.
* Historique des activités (logs).
* Interface bilingue (français/anglais).

Ce projet permet également d’illustrer comment des technologies telles que Flask, Streamlit, SQLite et des librairies de visualisation peuvent être combinées dans un contexte d’Intelligence Artificielle pour offrir une expérience utilisateur cohérente et évolutive.

# Cahier des charges

- Interface web responsive  
- Suivi GPS simulé  
- Commandes à distance (démarrer, arrêter, verrouiller...)  
- Interface bilingue (FR / EN)  
- Tableau de bord en temps réel  
- Historique d'activité (logs)  
- Authentification utilisateur (version étendue)

# Architecture du système

L’architecture générale du système repose sur trois couches principales :

**Présentation (Frontend), Traitement (Backend) et Stockage (Base de données)**.

1. **Couche de présentation : Interface Web**

* Développée avec **HTML**, **CSS (via Tailwind CSS)**, et **JavaScript**.
* Interface interactive utilisant **Streamlit** pour la carte GPS simulée.
* Affichage en temps réel des véhicules, positions, et boutons de contrôle.
* Composants de tableau de bord : boutons d’action, cartes, graphiques, etc.

1. **Couche de traitement : Backend**

* Utilisation du Framework **Flask (Python)**.
* Mise en place de **routes API** pour :
  + Démarrer/arrêter un véhicule.
  + Récupérer les logs.
  + Changer de langue.
  + Fournir les données GPS simulées.
* Gestion de l’internationalisation via des dictionnaires dynamiques Python (FR/EN).

1. **Couche de stockage : Base de données**

* Système de gestion de base **SQLite** (léger et embarqué).
* Enregistrement des points GPS simulés avec horodatage.
* Historique des actions utilisateurs (logs) : démarrage, arrêt, connexion, etc.
* Stockage des identifiants des véhicules et des informations associées.

# Simulation GPS et interface utilisateur

Une interface a été conçue en Streamlit avec une carte interactive où un véhicule virtuel se déplace automatiquement. Les boutons permettent à l’utilisateur de démarrer ou arrêter la simulation. Le système simule le comportement d’un véhicule en déplacement dans une ville fictive ou réelle selon des coordonnées GPS générées dynamiquement.

# Diagramme de Cas

Ce diagramme représente les cas d’usage de l’application de contrôle de véhicule à distance, du point de vue de l’utilisateur principal : l’Admin ou le Propriétaire du véhicule.

**Acteur principal :**

* Admin & Propriétaire de véhicule

**Cas d’utilisation principaux (fonctionnalités) :**

* Connexion à l’interface

Authentification de l’utilisateur pour accéder aux fonctionnalités de gestion du véhicule.

* Verrouillage & Déverrouillage

Action de sécuriser ou ouvrir le véhicule à distance.

* Démarrage & Arrêt moteur

Permet d’allumer ou d’éteindre le moteur via l’interface, en cas de besoin ou pour la mise en route à distance.

* Visualisation position

Affiche la position actuelle du véhicule sur une carte (via GPS/Folium).

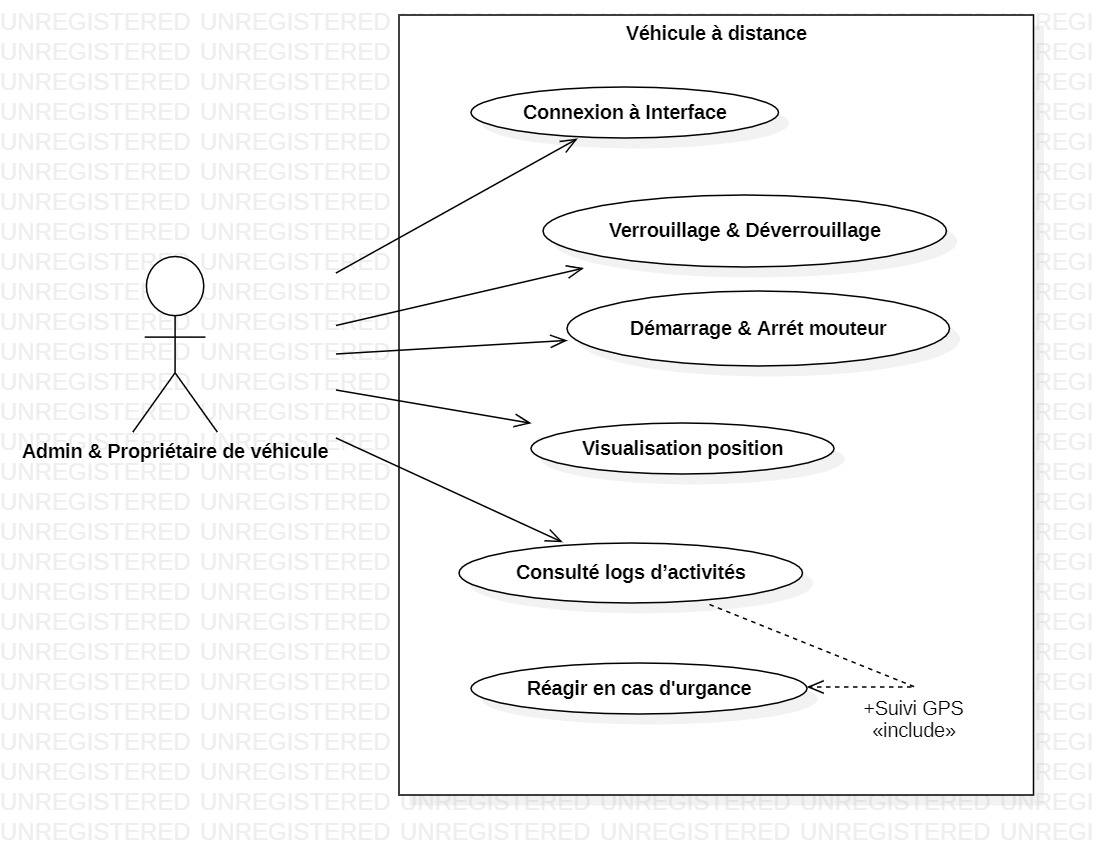
* Consulter les logs d’activités

Permet de visualiser l’historique des actions (verrouillage, démarrage, arrêts, etc.), enregistré automatiquement.

* Réagir en cas d’urgence

Active un arrêt d’urgence du moteur.

+ Ce cas inclut automatiquement un suivi GPS pour localiser précisément le véhicule.

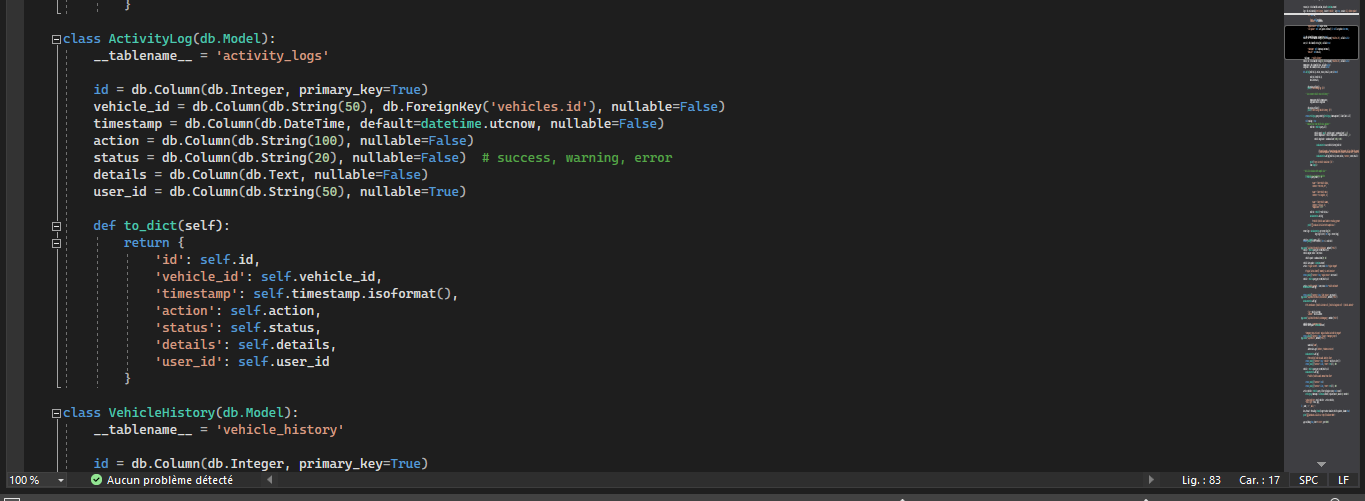


# Stockage et analyse des données

Chaque point GPS est stocké avec un horodatage dans une base de données. À partir de ces données, nous analysons :

* La distance totale parcourue
* Les heures d’activité les plus fréquentes
* Les zones les plus visitées

Les résultats sont présentés sous forme de graphiques pour faciliter l’interprétation. Ces visualisations permettent aussi d’identifier les habitudes de déplacement et d’anticiper des comportements futurs.

Figure : Extrait du modèle ORM pour les logs

Planification du projet

Planification hebdomadaire :

* **Semaine 1 :** Analyse fonctionnelle et technique
* **Semaine 2 :** Création des maquettes et interface
* **Semaine 3 :** Backend Flask (routes et sécurité)
* **Semaine 4 :** Gestion base de données et logs
* **Semaine 5 :** Tests, i18n et simulation GPS
* **Semaine 6 :** Rédaction rapport et soutenance

# Réalisation technique

Nous avons conçu une interface utilisateur affichant tous les véhicules et leur état en temps réel. Chaque commande effectuée (ex. démarrer moteur) est enregistrée avec son statut et sa date. Un système de traduction par dictionnaire Python permet de changer dynamiquement la langue du tableau de bord.

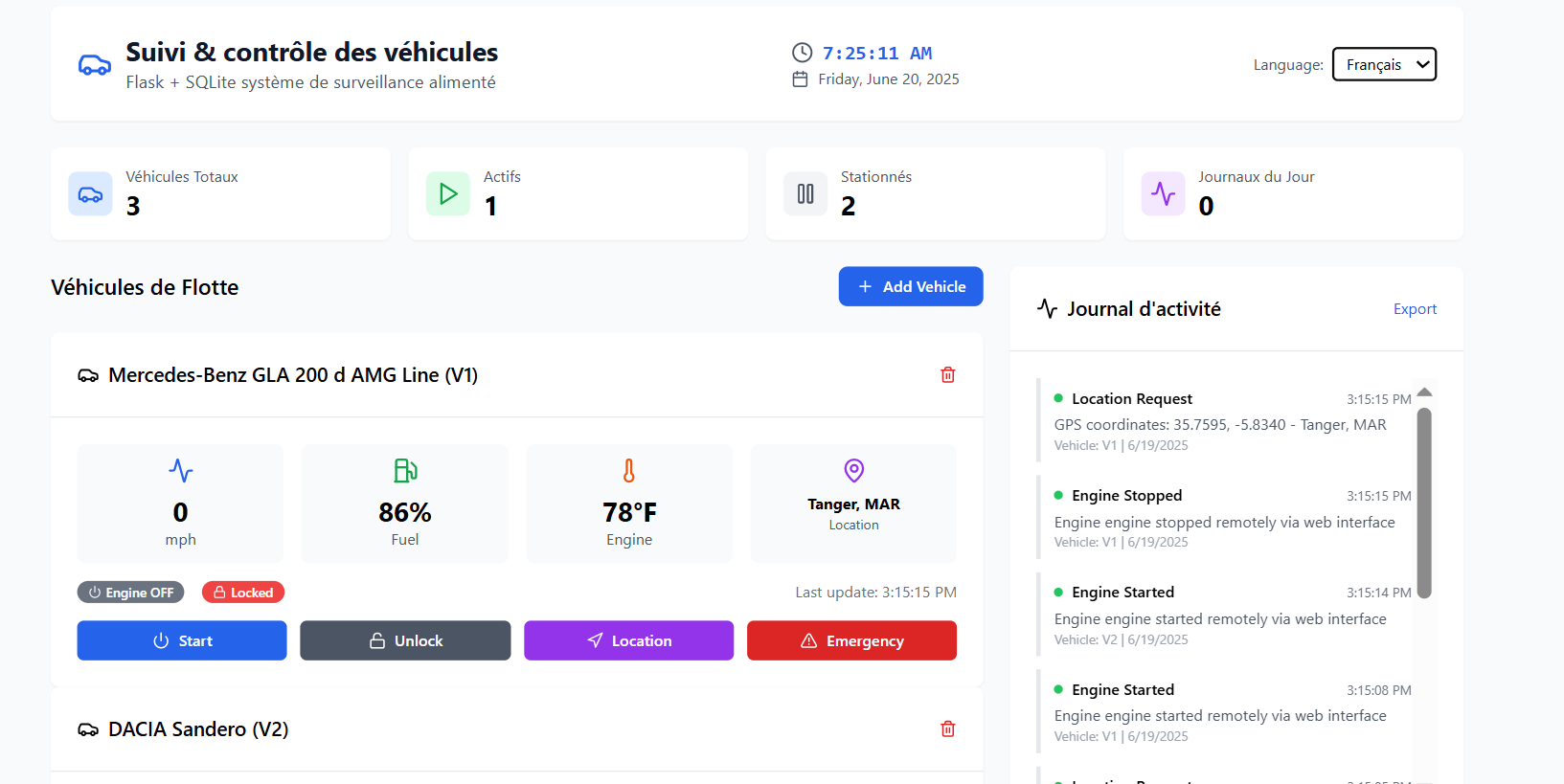
*Vue du tableau de bord (Dashboard) :*

Figure : Interface principale (Dashboard)

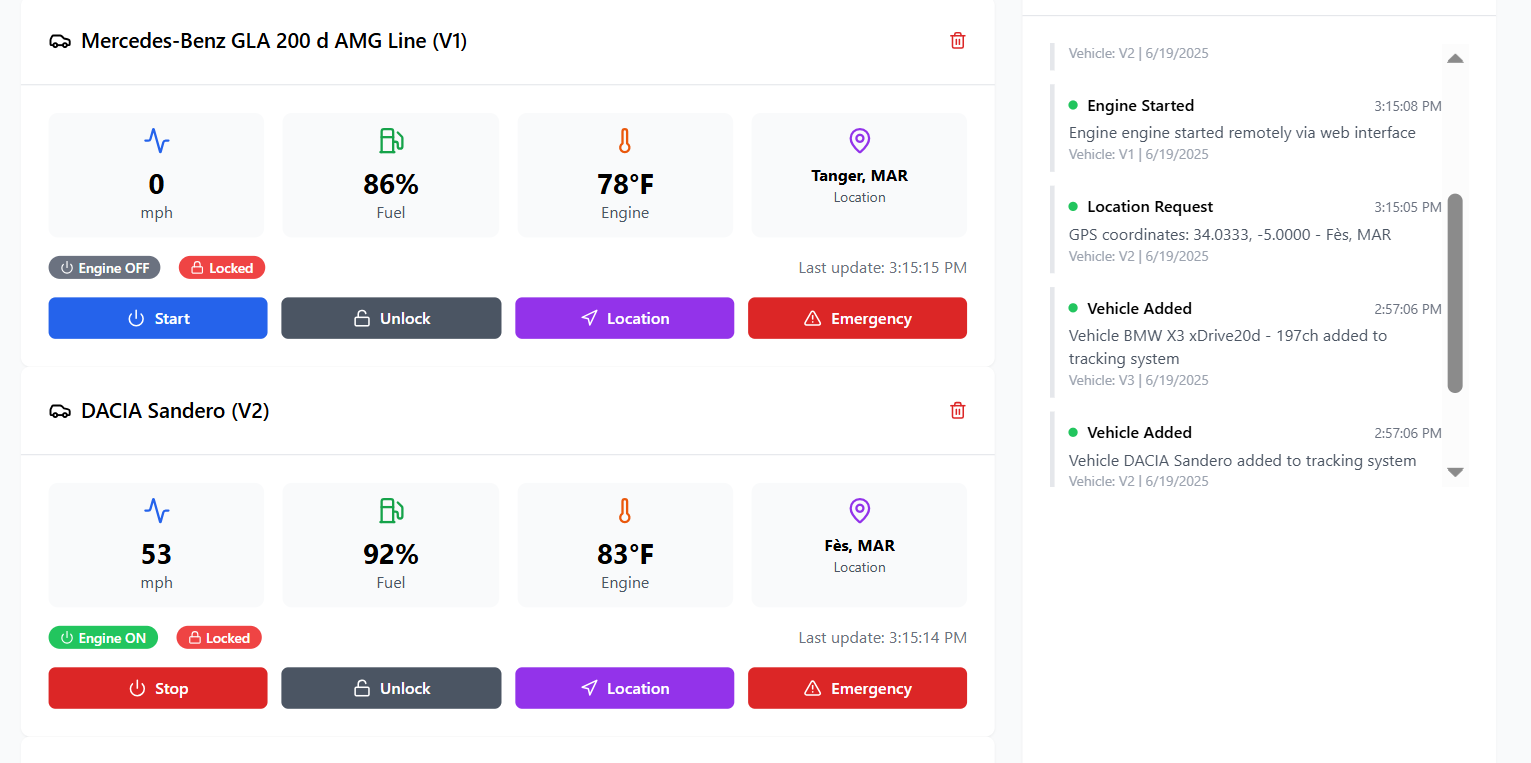
L’interface est fluide, multilingue, et fonctionnelle avec plusieurs véhicules.

Figure : Vue multi-véhicules dans l'application

*Formulaire d’ajout de véhicule :*

Capture d’écran illustrant le formulaire d’ajout de nouveaux véhicules dans l’interface.

Champs : Identifiant du véhicule, nom, position initiale, état par défaut.

Exemple :

* ID : VH005
* Nom : Dacia Ligan
* Position : 67,7639 / -52,0010 initiales
* État : Arrêté

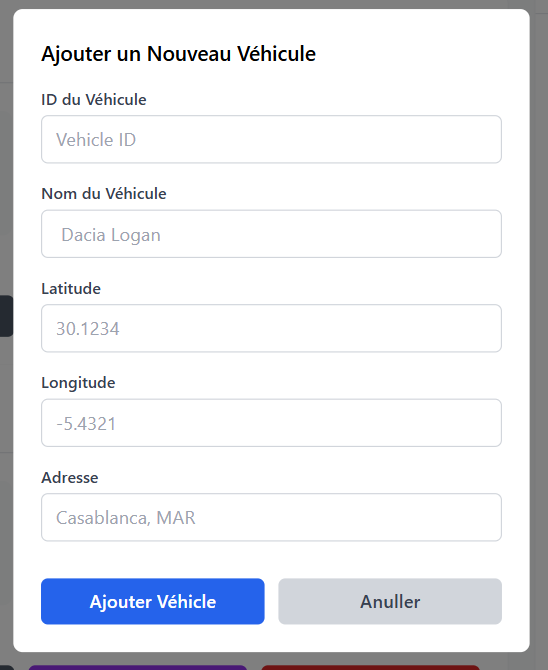


Figure : Formulaire d’ajout de véhicule

*Historique des logs :*

Exemple de table de logs :

|  |  |  |  |
| --- | --- | --- | --- |
| Date/Heure | Action | Véhicule | Résultat |
| 2025-04-11 14:05 | Start | VH001 | Succès |
| 2025-04-12 10:06 | Stop | VH001 | Succès |
| 2025-04-12 13:10 | Simulation GPS Position | VH002 | En cours |

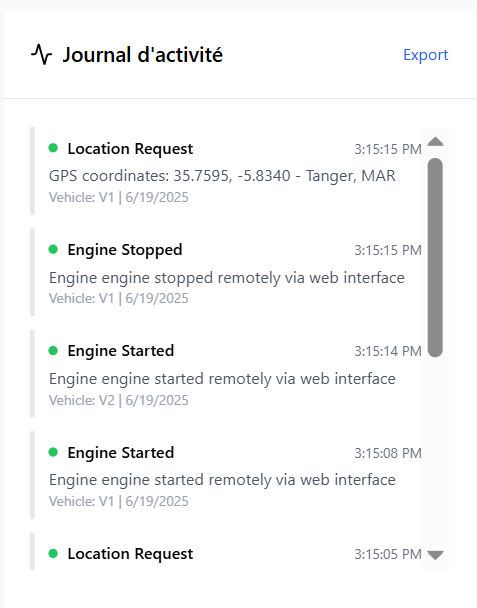


Figure : Historique des logs

# Tests et résultats

Les tests effectués montrent une réactivité satisfaisante du système, notamment pour :

* Le démarrage/arrêt du moteur
* Le verrouillage/déverrouillage du véhicule
* L’affichage des logs

# Conclusion

Le présent projet s’inscrit dans une démarche de transformation numérique appliquée à la gestion et au contrôle des véhicules à distance. En mobilisant des technologies modernes telles que Python/Flask pour le backend, Streamlit pour l’interface, ainsi qu’une base de données SQLite, nous avons pu concevoir une solution logicielle complète, modulaire et orientée utilisateur.

La spécificité du projet réside dans son approche purement logicielle, simulant les aspects essentiels du suivi et du contrôle d’un véhicule sans recourir à du matériel physique. Cette orientation a permis de se concentrer sur l’architecture logicielle, la structuration des données, la réactivité de l’interface, ainsi que l’adaptabilité multilingue du système.

La solution développée se distingue par son accessibilité, son interface intuitive et son approche entièrement logicielle, évitant les contraintes matérielles habituelles. Grâce à l'utilisation de technologies modernes telles que Python, Flask, Streamlit et SQLite, nous avons pu créer un environnement fonctionnel, modulable, et apte à évoluer vers des versions plus avancées, notamment avec des applications mobiles, des notifications temps réel ou une extension vers des usages professionnels.

En perspective, cette application pourrait être enrichie par une intégration matérielle (capteurs réels, modules GPS/GSM), un système d’alertes en temps réel, ou encore un déploiement mobile multiplateforme. Ce projet constitue ainsi une base solide pour des développements futurs dans le domaine des véhicules intelligents et de l’Internet des Objets (IoT).