**Documentation Technique**

Fonctionnalités de l'Application

* Authentification:

Permet l'inscription (/api/auth/register) et la connexion (/api/auth/login) des utilisateurs.

Utilisation de JWT (JSON Web Token) pour l'authentification et l'autorisation.

* Gestion des Véhicules:

Exposition des positions des véhicules en temps réel via WebSocket (/api/vehicles).

Utilisation de Socket.IO pour la communication en temps réel entre le serveur et les clients.

* Simulateur de Véhicules:

Utilisation d'une classe VehicleSimulator pour simuler les positions et les alertes des véhicules.

Émission des événements position et alert aux clients connectés via Socket.IO.

* Base de Données:

Utilisation de MongoDB avec Mongoose pour la persistance des données des utilisateurs et des véhicules.

Connexion à MongoDB via mongoose.connect().

* Technologies Utilisées

Backend:

* Node.js
* Express.js pour la gestion des routes et middleware
* Socket.IO pour la communication en temps réel
* MongoDB avec Mongoose pour la base de données
* JSON Web Tokens (JWT) pour l'authentification
* Dépendances Principales:
* express: Framework web pour Node.js
  + socket.io: Bibliothèque pour la communication en temps réel
  + mongoose: ODM (Object-Document Mapper) pour MongoDB
  + bcrypt: Pour le hachage sécurisé des mots de passe
  + jsonwebtoken: Pour la création et la vérification des JWT
  + Outils:

Postman pour tester les endpoints API ( POST,GET)