# Vehicular Sensor Data Monitoring and Analysis Webapp

## Overview:

The Vehicular Sensor Data Monitoring and Analysis Platform is a web application built with Flask, designed to collect, store and analyze real-time data from various sensors. The platform provides a user-friendly interface for monitoring sensor readings, visualizing, access of historical data and downloading CSV files for in-depth analysis. Here, the sensor values are depicted from dummy data.

The webapp is functioned to feed us live data from various sensors in a vehicle. The sensors provide data such as Oil temperature, Intake temperature, Coolant temperature, Rpm signals, Speed signals, Throttle positions. Although the data being depicted from dummy variables, these are crucial readings and are being used now a days, as the automobile industry is getting automated year by year.

## Key Features of Webapp:

### 1. Real-Time Dashboard:

A dynamic dashboard displays real-time sensor data, allowing users to monitor live updates from connected sensors.

### 2. Data Visualization:

Graphical representation of sensor data using interactive charts and graphs for easy analysis and trend identification.

### 3. Historical Data Analysis:

Access to historical sensor data for in-depth analysis. Users can view trends, patterns, and anomalies over specific time ranges.

### 4. CSV File Export:

Users can download CSV files containing sensor data for offline analysis or reporting purposes. The platform provides a list of available files for download.

### 5. Responsive Web Interface:

A responsive and intuitive web interface that adapts to different screen sizes, ensuring a seamless user experience across devices.

### 6. Background Data Processing:

Continuous background processes handle data updates and synchronization with the database(local storage), ensuring the platform reflects the most recent sensor information.

## Technical Stack:

* Backend: Flask, SQLAlchemy, Flask-SocketIO, Flask-WTF, other in-built modules
* Frontend: HTML templates with Jinja templating, Chart.js for data visualization.
* Database: SQLite3

## Further Enhancements:

* Extend the functionality of database by live storage and retrieval of data
* Adding more visualization techniques from the data uploaded by user
* More comparison techniques with live data and stored data