**🛠️ Project Document: Smart Fuel Tracker (Royal Enfield & General Use)**

**📌 Objective**

Build a real-time fuel and trip tracking system tailored for Indian vehicle users (starting with Royal Enfield), with support for tracking odometer, fueling, reserve behavior, trip segmentation, and cost insights. The system should allow for advanced insights while maintaining a minimal and focused core.

**📁 Project Modules Overview**

**1. 🔧 Vehicle Management**

* Add & manage **vehicles**
* Store essential bike/car details:
  + Name / Model
  + Fuel Type: Petrol / Diesel
  + Tank Capacity
  + Odometer at time of adding (e.g. 1000 km)
  + Initial fuel spent: 0 L
  + Initial amount spent: 0 Rs
  + Trip feature available or not
    - If not available, use odometer readings for trip tracking (start, mid, end)
    - In such cases, **Trip A Start/Mid/End should be replaced by Odometer Start/Mid/End directly**

**2. ⛽ Fuel Entry Recording**

**Triggered on refueling**

**Required Inputs:**

* Odometer Reading at fueling
* Is Reserve On? → Yes/No
  + If Yes: Trip F Reading (e.g. 11.2 KM)
* Fuel by:
  + Amount (Rs) → system fetches petrol cost using bunk type/date API
  + or Litres → calculates total cost
* Fuel Station Type (private, PSU, highway)
* Fueling Date

**3. 🧬 Trip Tracker**

**Optional but critical for high-quality insights**

If the bike doesn't support Trip A/B meters, fallback to odometer-based logging.

* Trip A Start:
  + Odometer reading (optional, used for syncing with Trip A)
  + Was Trip A reset before start?
    - Yes → Trip A Start = 0 KM
    - No → Provide current Trip A reading
* Trip A Mid:
  + One or more intermediate points (record Trip A reading at those points)
* Trip A End:
  + Final Trip A reading (A\_End)
  + Optional: Odometer at trip end (used for validation or fallback)

**Note:**

* If **Trip Meter = false**, then:
  + Use direct fields: Odometer Start, Odometer Mid, and Odometer End in place of Trip A tracking.
  + Trip entries should only log odometer readings, and mileage/analysis will be based directly on them.

**🔧 Odometer Sync Logic:**

* If both odometer and Trip A are provided, compute:
  + Distance covered = Odo\_End - Odo\_Start = A\_End - A\_Start (used for verification)
  + Infer Odo\_Start = Odo\_End - (A\_End - A\_Start) if only Odo\_End is available

**🔖 Labels:**

* Any segment (start/mid/end) can be tagged with **labels**
* Helps categorize and extract grouped insights
* E.g., "Work Ride", "Weekend Trip", "Commute", etc.
* Label-wise summaries are shown in the insights dashboard

**📊 Data Insights**

**1. ⚙️ Mileage Tracking**

* **Per Trip**: Trip A distance ÷ Litres
* **Overall Average**
* **Reserve Mileage** (Trip F-based)
* **Mileage over time** trend line

**2. 💰 Fuel Spending**

* Total fuel **cost**
* **Cost per KM**
* **Monthly/weekly spend**
* **Fuel price** variations over time

**3. 📏 Distance Coverage**

* Total distance across:
  + Trips
  + Labels
  + Reserve segments
  + Calendar-based ranges
* **Average daily KM**
* **Distance between refuels**

**4. ⚠️ Reserve Usage**

* **Total count** of times reserve was used
* **Avg KM covered on reserve**
* **Total KM covered on reserve**

**5. 🔄 Trip-Odometer Accuracy**

* Compare expected odometer end vs actual
* Validate correct resets of Trip A
* Useful for detecting entry errors

**6. 🎼 Trip/Label Based Analysis**

* Distance grouped by labels
* Fuel consumption by ride type
* Reserve hits by journey category

**🧱 Core Features to Implement (First Stage)**

| **Feature** | **Status** |
| --- | --- |
| Add new vehicle | ✅ |
| Add fuel entry | ✅ |
| Add trip data | ✅ |
| Calculate mileage | ✅ |
| Show spending summary | ✅ |
| Show total distance | ✅ |
| Group by labels | ✅ |
| Reserve analysis | ✅ |
| Insights by trip type | ✅ |
| Data validation & sync | ✅ |