Zenfinity Energy - Frontend Intern Assignment

Overview

Welcome to the Zenfinity Energy internship assessment! Your task is to build a comprehensive battery analytics dashboard using our Battery Snapshots API. We want to see your ability to visualize complex data, create intuitive user interfaces, and derive meaningful insights from battery telemetry.

Goal: In a snapshot, we want to show how a battery was used/ran in a particular cycle and also try to capture its variations over cycles.

API Access

Base URL: `https://zenfinity-intern-api-104290304048.europe-west1.run.app`

For security reasons, this API is restricted. You are authorized to access data **ONLY** for the following battery IMEIs:

- `865044073967657`
- `865044073949366`

API Endpoints

The API provides the following endpoints to access battery cycle data:

1. **GET /api/snapshots/summary**
    - **Description**: Returns a summary list of all accessible batteries.
    - **Usage**: Use this to identify the available IMEIs and get high-level stats (total cycles, avg health, etc.).
    - **Parameters**: `imei` (optional filter).
2. **GET /api/snapshots**
    - **Description**: Retrieve a list of cycle snapshots for a specific battery.
    - **Usage**: Fetch detailed cycle data for a chosen IMEI. Supports pagination.
    - **Parameters**: `imei` (required), `limit` (default 100), `offset` (default 0).
3. **GET /api/snapshots/{imei}/latest**
    - **Description**: Get the single most recent cycle snapshot for a battery.
4. **GET /api/snapshots/{imei}/cycles/{cycle_number}**
    - **Description**: Get detailed analytics for a specific cycle number.

Data Dictionary

Here is a breakdown of what the data fields in a cycle snapshot mean:

```
{
  "imei": "Unique Identifier for the battery pack",
  "cycle_number": "Sequential number of the charge/discharge cycle",
  "cycle_start_time": "Timestamp when the cycle began (ISO 8601)",
  "cycle_end_time": "Timestamp when the cycle ended (ISO 8601)",
  "cycle_duration_hours": "Total duration of the cycle in hours",

  "soh_drop": "Decrease in State of Health during this cycle (%)",
  "average_soc": "Average State of Charge maintained during the cycl
  "min_soc": "Lowest SOC recorded in this cycle",
  "max_soc": "Highest SOC recorded in this cycle",

  "average_temperature": "Average temperature of the pack (°C)",
  "temperature_dist_5deg": {
     "20-25": 10.5  // Means 10.5 minutes were spent between 20°C ar
  },
  // Note: temperature_dist_10deg, _15deg, _20deg follow similar log

  "total_distance": "Distance traveled by the vehicle in this cycle
  "average_speed": "Average speed during motion (km/h)",
  "max_speed": "Maximum speed recorded (km/h)",

  "charging_instances_count": "Number of times the battery was charg
  "average_charge_start_soc": "Average SOC at which charging began (

  "voltage_avg": "Average pack voltage (V)",
  "voltage_min": "Minimum pack voltage (V)",
  "voltage_max": "Maximum pack voltage (V)",

  "alert_details": {
      "warnings": ["List of warning codes/messages"],
      "protections": ["List of protection events triggered"]
  }
}
```

Assignment Requirements

Part 1: Data Retrieval

1. Access the API and fetch the summary data.

2. Select one of the allowed IMEIs.
3. Fetch the detailed cycle snapshots for the selected battery.

Part 2: Dashboard Implementation

Create a web-based dashboard (using React, Vue, Svelte, or any modern framework) that visualizes the fetched data. The dashboard must clearly showcase:

1. **Cycle Navigation**: A responsive way to navigate through different cycles (slider, dropdown, etc.).
2. **Cycle Statistics**: Key metrics: Cycle Number, Start/End DateTime, Duration, etc.
3. **Performance Metrics**: Visualize Average Speed, Total Distance, etc.
   - *Note: Distance/GPS calculations might have gaps; focus on visualizing the available data.*
4. **Temperature Distribution**: Histogram/chart of time spent in temperature ranges.
   - **Requirement**: Toggle between sampling rates (5°C, 10°C, 15°C, 20°C).
5. **Battery Health (SOC & SOH)**: Visualizations for SOC and SOH trends.
6. **Alerts & Safety**: Display Protections/Warnings clearly.
7. **Charging Insights**: Charging event stats.
8. **Additional Insights**: Any other patterns you find.

Part 3: Advanced Analysis (Bonus)

9. **Long-term Trends**: Analyze how values change **across all cycles** (e.g., SOH degradation curve).

Evaluation Criteria

- **Dashboard Quality**: Simple, functional, clean, and intuitive designs will be prioritized.
- **Meaningful Analysis**: Ability to derive insights from the data.
- **Code Quality**: Clean, maintainable code.
- **Comprehension**: Basic understanding of Li-ion battery metrics (LFP chemistry) is a plus.

Submission

1. **GitHub Repository**: Public repo with source code and instructions to run locally.
2. **Live Demo**: Host your dashboard on Vercel, Netlify, or similar.
3. **Submit Here**: Google Form Link

Deadline: 14th December, 2025 (Sunday)