Software Task 2

Report Name: Steny Thankkam Raju
College: Saintgits College Of Engineering

Introduction

In this project, we developed a web-based IoT data visualization dashboard that displays sensor data for multiple verticals such as Air Quality, Water Quality, and Solar Light Systems.

The system combines a FastAPI backend that provides API endpoints for processed sensor data with a React.js (Vite) frontend for visual, interactive dashboards using charts. The platform helps users understand sensor trends, identify anomalies, and make data-driven decisions.

Dataset Overview

Air Quality (aq):
calibrated_pm25, calibrated_pm10, calibrated_temperature, calibrated_relative_humidity, calibrated_noise

(sl):
active_power, voltage_rs, frequency, power_factor, pv1_power, pv2_power, pv3_power

Water Flow (wf):
flowrate, pressure, pressure_voltage, flow_volume, total_flow

Steps Followed

1. Created PostgreSQL database **iot_db**.

2. Loaded the dataset **iot_dataset.csv** and **iot_dataset_mapping.csv** into Python using Pandas.

3. Processed and inserted records into PostgreSQL tables (aq , sl and wf).

4. Developed FastAPI backend with endpoints /api/aq, /api/sl, and /api/wf to serve data in JSON format.

5. Handled missing (NaN) values by converting them to None before sending responses.

6. Built React frontend using **Vite** to visualize real-time IoT data trends.

7. Line graph visualization of each vertical (Air Quality, Solar Light, Water Flow).

Backend Implementation (FastAPI)

The **FastAPI** backend is responsible for serving sensor data.
When an API request (e.g., /api/aq) is made:

1. It connects to the **PostgreSQL** database.

2. Retrieves relevant vertical data from its respective table.

3. Converts NaN → None to ensure valid JSON serialization.

4. Returns the data as a structured JSON response to the frontend.

Frontend Implementation

The **frontend** was built using **React (Vite)**.
It communicates with the FastAPI backend through axios and displays sensor trends using **ECharts**.
Users can switch between dashboards for:

- **Air Quality (AQ)** – shows PM2.5, PM10, Temperature, Humidity, Noise

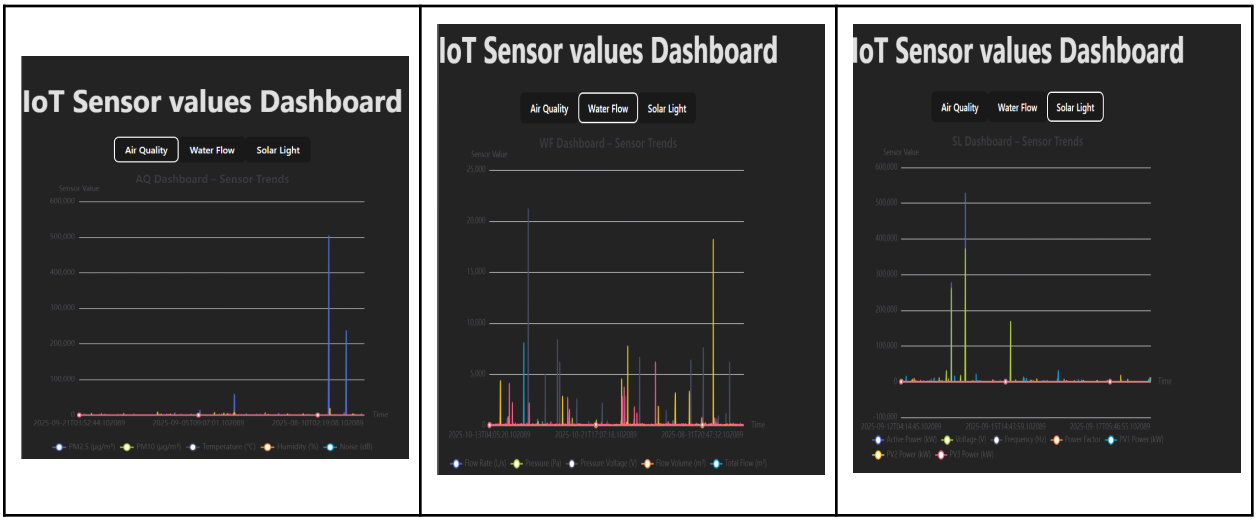
- **Solar (SL)** – shows Active Power, Voltage, Frequency, Power Factor, PV1–PV3 Power

- **Water Flow (WF)** – shows Flowrate, Pressure, Pressure Voltage, Flow Volume, Total Flow

Visualization

Each dashboard presents **time-series line charts** generated with ECharts.

- X-axis → Timestamp (created_at)
- Y-axis → Corresponding sensor metric values
- Multiple lines display different sensor parameters for the same vertical.
  This provides a clear visual comparison of sensor behavior and performance trends.



Conclusion

The project successfully integrates PostgreSQL, FastAPI, and React (ECharts) for IoT data visualization.
It provides dynamic dashboards for different verticals, allowing users to analyze trends across air quality, solar light, and water flow systems in real-time.