NAME – DRISHTI

ROLL NO. – 2501730232

SUBJECT – PROGRAMMING FOR PROBLEM SOLVING USING PYTHON

SUBMITTED TO – MR. SAMEER FAROOQ

PROJECT REPORT 5

Real-Time Campus Energy Monitoring Dashboard using Python OOP, Pandas & Matplotlib

Abstract

This project develops a complete end-to-end energy analytics pipeline that automatically ingests hourly electricity consumption data from multiple campus buildings, validates and cleans the data, models it using Object-Oriented Programming, performs campus-wide and building-wise aggregation, generates professional visualizations, and produces an executive summary with actionable insights.

Key Objectives Achieved

Task	Description	Status
1	Automated ingestion of multiple CSV files	Completed
2	Campus-wide & building-wise aggregation	Completed
3	Object-Oriented Modeling (MeterReading & Building classes)	Completed
4	Multi-panel dashboard visualization	Completed
5	Logging, persistence & executive report	Completed

Technology Stack

- **Core Libraries**: pandas, matplotlib, pathlib, logging
- **Design Pattern**: Object-Oriented Programming with separation of concerns
- **Output**: CSV exports, PNG dashboard, text summary, detailed log file

---

Page 2 – System Architecture & OOP Design

Overall Architecture Diagram (Conceptual)

text

```
[data/*.csv]
    ↓ (ingest_and_validate_data)
Combined DataFrame → BuildingManager
                          ↓
               Individual Building Objects
                          ↓
             Aggregation + Visualization + Reports
```

```
                ↓
        [output/] folder (CSV, PNG, TXT, LOG)
```

Core OOP Classes Designed

Class	Responsibility
MeterReading	Encapsulates single timestamp + kWh reading with validation
Building	Stores all readings of one building • Total, daily, weekly calculations • Summary statistics • Text report generation
BuildingManager	Central controller • Holds all Building instances • Distributes data • Generates campus-level summary table

Key Features of OOP Implementation

- Strict data validation (negative kWh not allowed)
- Internal pandas DataFrame for fast calculations
- Clean separation between data storage and analytics
- Reusable methods (calculate_daily_totals, generate_report)

Page 3 – Features & Output Description

1. Automated Data Ingestion (Task 1)

- Supports any number of CSV files in data/ folder
- Filename format: BuildingName_MonthYear.csv → extracts building name automatically
- Robust error handling: skips bad lines, logs errors, converts types safely
- Adds Building_Name column for later grouping

2. Aggregation & Analytics (Task 2 & 3)

Level	Calculations Performed
Building	Total kWh, Mean/Min/Max hourly, Daily totals, Weekly sum & mean
Campus	Daily total consumption across all buildings Weekly total consumption Peak load detection

3. Professional Dashboard Visualization (Task 4)

File: output/dashboard.png (18×5 inches – 3 subplots)

| Subplot | Chart Type | Insight Provided | | |--------|-----------------|---| | Left | Line Chart | Daily consumption trend for each building | | Middle | Horizontal

Bar | Average weekly hourly usage comparison | | Right | Scatter Plot | Hourly load profile (peak usage hours visible) |

4. Executive Summary Report (Task 5)

File: output/summary.txt

Sample Output (Actual values depend on data):

text

```
================================================
   Campus Energy-Use Executive Summary
================================================
Analysis Period: 2024-10-01 to 2024-11-09


1. Total Consumption: 28,451.73 kWh


2. Highest-Consuming Building:
   - Labs_C with a total of 12,847.21 kWh.


3. Peak Load Time:
   - The highest recorded load was 587.40 kWh at 2024-10-15 14:00


4. Consumption Trends:
   - The average daily campus consumption was 712.14 kWh.
   - The average weekly consumption was approximately 4,985 kWh.
================================================
```

5. Exported Files (All in output/ folder)

File	Content
cleaned_energy_data.csv	Final combined & cleaned dataset
building_summary.csv	One-row-per-building statistical summary
dashboard.png	3-panel professional dashboard
summary.txt	Human-readable executive report
processing.log	Complete log of operations and warnings

Page 4 – Conclusion & Learning Outcomes

Key Achievements & Innovations

- Fully automated pipeline – just drop CSV files and run script
- Production-grade logging for debugging and audit trail
- Defensive programming with validation at every layer
- Realistic mock data generation (uncomment one line to test without real files)
- Clean, maintainable OOP structure ready for future extensions (e.g., database, web API)

Skills Demonstrated

Domain	Concepts Applied
OUTPUT SCR	pathlib, glob, automatic directory creation
Data Cleaning & Validation	Type coercion, bad line skipping, logging
Advanced Pandas	resample(), multi-level grouping, pivot/unstack
Object-Oriented Design	Encapsulation, single responsibility, composition
Visualization	Professional visualization & reporting

Future Enhancements (Possible Extensions)

- Add SQLite/MySQL export
- Web dashboard using Streamlit/Plotly Dash
- Anomaly detection (sudden spikes)
- Cost calculation using tariff rates
- Email/PDF report automation

Final Statement

This project successfully delivers a robust, scalable, and professional-grade energy monitoring system entirely in Python. It goes beyond basic requirements by incorporating logging, OOP best practices, automated reporting, and a polished multi-chart dashboard — making it suitable for real-world campus facility management use.

OUTPUT SCREEN