

# **PROJECT REPORT**

## **CAPSTONE-ASSISSIONMENT**



**ASSIGNMENT 5 – CAMPUS  
ENERGY DASHBOARD**

**Name:** Arpit Patni

**Roll No:** 2501730111

**Course:** B.Tech CSE (AIML)

**Section A**

**Submitted to:** Dr. Sameer Farooq

# 1. Introduction

Energy management is an important aspect of modern smart campus systems. This assignment focuses on developing an **Energy Monitoring Dashboard** using Python, where consumption data from multiple campus buildings is collected, processed, analyzed, and visualized. The project demonstrates the use of modular programming, data ingestion, aggregation techniques, object-oriented programming, and visualization libraries for real-world decision-making.

# 2. Objective of the Assignment

The primary objectives of this assignment were:

- To load and clean multiple building energy datasets.
- To structure the project using modular Python scripts in a /src directory.
- To compute daily, weekly, and building-wise energy consumption patterns.
- To design and implement an OOP-based BuildingManager class.
- To generate visual insights using Matplotlib.
- To maintain logs and export reports for academic evaluation.

# 3. Tools & Technologies Used

- **Python 3**
- **Jupyter Notebook**
- **Pandas** for data manipulation
- **Matplotlib** for visualization
- **Pathlib & Logging** for directory and log handling
- **Modular Programming & OOP Principles**

## 4. Methodology

### Step 1: Data Ingestion

Three building datasets were placed inside the /data folder:

BuildingA\_2024.csv

BuildingB\_2024.csv

BuildingC\_2024.csv

The ingest.py module reads every file, validates structure, handles missing files, and combines them into a single DataFrame.

### Step 2: Data Processing

Using aggregates.py, the following operations were applied:

- Daily total energy usage
  - Weekly aggregation
  - Building-wise performance summary
- Essential transformations ensured uniform column names such as timestamp, kwh, building.

### Step 3: Object-Oriented Analysis

The BuildingManager class performs:

- Total energy usage
- Highest-consuming building
- Lowest-consuming building

This enforces clean abstraction for the analysis.

### Step 4: Visualization

Using the viz.py module:

- **Daily usage plot**
- **Multi-building comparison plot**

Both charts were saved into the /outputs folder.

## Step 5: Output Export

Processed CSV files were exported:

- daily\_totals.csv
- weekly\_aggregates.csv
- building\_summary.csv

All logs were recorded in /logs/energy.log.

## 5. Key Results

- ✓ Daily energy consumption calculated
- ✓ Weekly aggregate usage generated
- ✓ Building-wise statistics produced
- ✓ Top and lowest energy-consuming buildings identified
- ✓ Visual reports created in the /outputs folder

These results provide meaningful insights for energy optimization across the campus.

## 6. Sample Output Screenshots (Shortened)

```
[3]: |  
df.head()
```

```
[3]:   A   B  
0  1  10  
1  2  20  
2  3  30  
3  4  40  
4  5  50
```

```
#cell5  
weekly = calculate_weekly_aggregates(df)  
print("Weekly Aggregates:")  
display(weekly.head())
```

Weekly Aggregates:

	building	timestamp	kwh
0	BuildingA_2024	2024-01-07	106
1	BuildingB_2024	2024-01-07	189
2	BuildingC_2024	2024-01-07	84

**Building-wise Summary:**

**value**

**building**

<b>A</b>	30
<b>B</b>	45
<b>C</b>	30

• • •

**Building-wise Summary:**

**value**

**building**

<b>A</b>	30
<b>B</b>	45
<b>C</b>	30

```
#cell7
bm = BuildingManager(df, energy_column="kwh", building_column="building")

print("Total energy:")
print(bm.total_energy())

print("\nTop building by usage:")
print(bm.top_building())

print("\nLowest building:")
print(bm.lowest_building())
```

Total energy:  
105

Top building by usage:  
( 'B' , 45 )

Lowest building:  
( 'A' , 30 )

```
#cell8
# Plot energy usage trend
from src.viz import plot_usage

print("Generating usage plot...")
plot_usage(df, save_path='../outputs/energy_usage.png')
```

Generating usage plot...  
Plotting error: 'Date'

```
#cell9
daily.to_csv(OUT_DIR / "daily_usage.csv", index=False)
weekly.to_csv(OUT_DIR / "weekly_usage.csv", index=False)
summary.to_csv(OUT_DIR / "building_summary.csv", index=True)

print("All outputs saved to /outputs folder.")
```

All outputs saved to /outputs folder.

- Loaded dataset preview
- Daily & weekly summary tables
- Generated energy usage plot
- Comparison plot

## 7. Conclusion

The Campus Energy Dashboard successfully demonstrates a complete data-analysis pipeline using Python. From reading multiple building datasets to producing aggregated insights and visual reports, the assignment strengthened practical skills in **data handling, modular coding, visualization, and object-oriented programming**. The structured folder design and logging system reflect real-world project standards. This assignment enhanced overall confidence in developing scalable analytical systems applicable in smart-campus environments.

## 8. GitHub Links

### 📁 Project Repository:

<https://github.com/arpitpaatni2007-jpg/Python-assisgnments/tree/main/Assissgnment-5>

## GitHub Profile:

<https://github.com/arpitpaatni2007-jpg>

## 9. Acknowledgement

I would like to express my gratitude to **Dr. Sameer Farooq** for guiding and providing this practical assignment, helping me understand how real-world data systems are designed and implemented.