

Electricity Sector Data Streaming and Analysis

- COMP5339 Project Assignment 2 -

In this data engineering assignment, your team will retrieve, process and visualize real-time electricity generation data. The project aims to provide practical experience in core data engineering skills such as data retrieval, stream processing, integration, cleaning, transformation, and visualisation.

This is a **group assignment**, comprising teams of 2 to 3 students. By default, you will remain in **the same group as formed for Assignment 1**, but you are free to switch groups (in assignment 2 groups setting) if you prefer.

This assignment builds on Assignment 1.

Submit your Assignment 2 report & code by **23:59, Friday 7 November (Week 13)**. All submissions will undergo plagiarism checks.

Assignment 2 Tasks

This Assignment 2 concentrates on acquiring and visualizing real-time data about electricity generation and emissions in Australia for historic data and trend analysis. Your team needs to develop Python programs to complete the following tasks:

1. Data Retrieval

Retrieve the following data from the Open Electricity platform using its REST API (<https://docs.openelectricity.org.au/api-reference/overview>):

- per-facility power generated (5 min interval)
- per-facility CO2 emissions (5 min interval)
- optional: per-market price and demand (5 min interval)

You can concentrate on the NEM network (which covers Queensland, NSW, Victoria, South Australia and Tasmania). Get the data for at least 1 week in October 2025, but notice that with a free account at OpenElectricity, you have an allowance of only 500 requests per day (<https://docs.openelectricity.org.au/api-reference/data-limits>).

2. Data Integration and Materialization/Caching

Combine the retrieved data into a single, consolidated dataset of generated power and CO2 emissions per facility. During this process, you may need to clean and pre-process the data to ensure consistency. Tasks may include handling missing values, converting data formats, and filtering out irrelevant or inconsistent data. Store the consolidated data into a CSV file.

The goal of this is to avoid continuous external API calls in the next tasks.

3. Data Publishing via MQTT

For each retrieved **power** and **emissions** record of a facility, publish a combined message power+emissions to a MQTT server containing all the necessary information required for subsequent tasks, simulating an ordered data stream (event time order). Make sure, there is at least a **0.1**-second delay between each published message.

4. Data Subscription and Visualisation

Create a map-based dashboard using a Python framework of your choice (similar to <https://explore.openelectricity.org.au/facilities/nem/>).

Specifically, subscribe to the MQTT messages published in the previous task 3. For each received message, dynamically add a marker to the map at the corresponding electricity facility's location (as collected in Assignment 1). Each marker should display the power station's name and current power output or emissions (ideally switchable). When a user clicks a marker, it should display a pop-up with the station's name, type, and the latest power production and emissions data.

Optional: Add filtering by network region and fuel technology, and in case you did the optional Task 1, display the corresponding electricity market price and demand too.

5. Continuous Execution

Ensure your program for Task 1—3 runs continuously to simulate an unbounded data stream. Implement a 60-second delay between each new round of API data retrieval; this delay is in addition to the 0.1-second delay between publishing messages.

6. Documentation and Reporting

Maintain detailed documentation throughout your project workflow. Your final report should clearly summaries your methods for data retrieval, integration, cleaning, augmentation, and transformation. Specifically:

- Highlight key insights and findings.
- Describe challenges encountered and how you overcame them.
- Provide recommendations for future improvements.

Deliverables

Submit the following three files via Canvas:

1. Python files (.py or .ipynb)

- Consolidate your Python scripts or Jupyter notebooks into **as few files as possible** (ideally two: one for Tasks 1—3 and one for Task 4).

2. requirements.txt file

- Include a requirements.txt file listing all Python packages required to run your program within a clean Python virtual environment.

3. Project Report (.pdf)

- Provide a clearly structured report of **up to 4 pages** (not counting title page and with

optional additional appendix).

- Recommended sections includes, but are not limited to:
 - **System Description:** Explain the system components and their roles for your dashboard that displays the continuous power generation data.
 - **Data Acquisition / Cleaning / Transformation:** Explain how you acquired the original dataset from OpenElectricity and any steps required to prepare the timeseries data for publishing as individual MQTT messages. How did you e.g. handle that some power facilities consist of several facility units?
 - **Data Integration:** Explain how you integrate the MQTT messages with your existing data from Assignment 1.
 - **Data Visualisation:** Show some example visualization or dashboard screenshot.
- You report must include a short paragraph not exceeding 100 words, detailing the contribution of individual team members, example:
 - AXX: Led data collection and preprocessing, ...
 - BXX: Developed the dashboard ...
 - ...

Submission Deadline

All Assignment 2 deliverables are due by **23:59, Friday 7 November** (Week 13, extended from original plan). Late submissions will incur penalties according to university policy.

Group Member Participation and Marking

This is a group assignment. Please declare the level of participation of each team member as part of your report (see 'Deliverables' above), but also note that the mark awarded for your assignment is conditional on you being able to explain any of your answers to your tutor or the lecturer if asked.

If your group is experiencing difficulties with the content, you should ask on Ed (use a private post if you need to discuss code or report writing directly).

Level of contribution	Proportion of group mark received
No participation.	0%
Minor contributor with at least some understanding of the group submission.	50%
Major contributor to the group's submission.	100%