CECS 327 Assignment 8 Report: End-to-End IoT System

Group Members: Matthew Marietta, Matthew Carranza

Group 14
Course: CECS 327 - Spring 2025

System Architecture and Approach

Our project implements a complete IoT system using a TCP client-server model integrated with a PostgreSQL database hosted on NeonDB. The client allows the user to select from three predefined IoT-related queries. These are sent to the server via TCP, which we configured from a previous assignment. The server handles incoming queries by connecting to the database, retrieving relevant sensor data, using device metadata from a separate metadata table, and performing necessary calculations before returning a human-readable and meaningful result to the client.

Components and Metadata not being used

The system components include the client, which is used to connect to the server and send one of the queries to the server. The server listens on the same socket as the client and then fetches data from NeonDB, applying logic using metadata from the virtual table and other use cases for it. We did not explicitly use the metadata table in Dataniz because it was not useful in our case for just grabbing information; the virtual table provided us with everything.

IoT Sensor Research

We simulated three smart home IoT devices:

- **Fridge**: Reports moisture levels inside the fridge. Data is recorded as RH% (relative humidity) as we configured on Dataniz.

- **Smart Dishwasher**: Reports water usage per wash cycle and is recorded in liters.

- **Fridge #2**: Same thing as the previous fridge, but set different generation limits for testing purposes.

These results were stored in tables respectively labelled as virtual and metadata; however, we only used virtual because we could grab our numbers there with our respective queries. The metadata did not offer anything useful to use for deciding the device type because this was already provided by the virtual table.

Algorithms, Calculations, and Conversions

The server performs the following operations by using helper functions within python and our SQL queries:

Moisture: Average RH% readings from the past 3 hours

Water Usage: Averages the water usage values per dishwasher cycle and converts from liters to gallons.

Electricity: Sums electricity values per device and converts joules to kilowatt-hours (kWh) using the formula: kWh = joules / 3,600,000.

Timestamps: Converted from UTC to PST using Python's pytz library.

Challenges and How We Addressed Them

- **Timezone conversion**: Handled with the library pytz to ensure accurate time filtering.
- **Using metadata**: Only used the virtual table, didn't find anything meaningful to use within the actual metadata table.
- **Unit conversions**: Had to ensure consistent interpretation and use outside resources to check the conversion of raw sensor values.

Feedback on Dataniz

Dataniz was helpful for simulating real IoT devices and generating realistic test data. We appreciated the ability to toggle devices and monitor real-time values.

Suggestions for improvement:

- Add an option to export sensor and metadata as CSV within the site, such as for machine learning.
- Improve visibility of units and time zones in the interface, or add an extra site that has access to common devices that cannot be found on the website.
- Include a short tutorial or walkthrough for new users, hard for users who did not have access to our assignment PDF.