Case Study 3: Indoor Climate Monitoring and Visualization with Raspberry Pi

DAT-230 Data Visualization & Storytelling with AI

Instructor: Dr. Vikas Thammanna Gowda Semester: Fall 2025
Contact: vthammannagowda@champlain.edu Office Location: West Hall 100
Office Hours: TBD

Case Study 3: Weather and Climate Patterns
Assigned Date: (Insert date here) **Due Date:** (Insert date here)

LLM Prompts for Raspberry Pi Setup

Prompt 1: Dependency Installation

```
You are a Raspberry Pi expert.
Please guide me step-by-step to update the package index and install Python3, pip, venv,
and the sensor libraries (Adafruit_DHT, adafruit-circuitpython-tsl2561,
adafruit-circuitpython-ads1×15) on Raspberry Pi OS. Explain each command's purpose
and how to verify successful installation.
```

Prompt 2: Enable I^{2}C

```
You are a Raspberry Pi configuration assistant.
Explain how to enable I2C on Raspberry Pi OS using raspi-config, including menu navigation
and verification of I2C status.
```

Prompt 3: MQ135 Instantiation

```
 You are an Adafruit sensor library tutor.  Show me how to correctly instantiate an
AnalogIn channel for the MQ135 sensor using adafruit-ads1x15 in Python, including import
statements and usage.
```

Prompt 4: LDR Integration

```
 You are a hardware integration instructor.  Describe how to wire an LDR in a resistor
divider to an ADS1115 channel, and update the read_light_lux() function to read and
return raw or normalized LDR values in Python.
```

Prompt 5: MQ135 Calibration

```
 You are a data calibration guide.  Explain how to record a clean-air baseline for
MQ135 at startup, implement normalization (normalized = (raw - baseline)/(max_expected
- baseline)), and log both raw and normalized readings.
```

Prompt 6: Time Synchronization

`You are a system administrator. Instruct me on installing and configuring NTP or systemd-timesyncd on Raspberry Pi and how to verify the clock synchronization status.`

Prompt 7: Deployment Metadata

`You are a deployment documentation assistant. Show me how to structure a JSON metadata section to record placement details, intentional bias notes (e.g., \sunlight exposure 14:00{16:00"), and calibration offsets before a data run.`

Immediate Next Steps on the Raspberry Pi

- **Install dependencies** (for Debian-based Raspberry Pi OS):

  ```
  sudo apt update
  sudo apt install python3-pip python3-venv
  pip install Adafruit_DHT adafruit-circuitpython-tsl2561 adafruit-circuitpython-ads1x15
  ```

- **Enable I^{2}C:** In `sudo raspi-config` under "Interface Options," turn on I^{2}C if using ADS1115 or TSL2561.
- **Fix the MQ135 channel instantiation:** Replace the placeholder in `read_mq135_raw()` with a real AnalogIn. For example:

  ```
  from adafruit_ads1x15.analog_in import AnalogIn
  chan = AnalogIn(ads, ADS.P0)   % or AnalogIn(ads, 0) as needed
  return chan.value
  ```

- **Swap in an LDR (optional):**
  - Wire the LDR in a resistor divider to an ADS1115 channel.
  - Update `read_light_lux()` to read from that channel and log raw or normalized values (note: it won't be true lux without calibration).
- **MQ135 calibration / mapping:**
  - Read a clean-air baseline at startup.
  - Optionally normalize:

    ```
    normalized = (raw - baseline) / (max_expected - baseline)
    ```

  - Always also record the raw value for later drift correction.
- **Ensure time synchronization:**
  - Install and enable `ntp` or `systemd-timesyncd`.
  - The script should timestamp readings immediately upon sensor acquisition.
  - For multi-Pi deployments, point all devices at the same NTP server.
- **Deployment descriptor / bias annotation:** Before starting a data run, edit your JSON (or script) metadata to record:
  - Placement details (height, distance to window/vent).
  - Any intentional bias (e.g., "sensor exposed to direct sunlight 14:00–16:00 to illustrate radiative heating").
  - Calibration offsets or notes.