Project 1 – Hardware Abstraction with Adapter Pattern

1. Objective

Design and implement a hardware abstraction layer using the **Adapter Design Pattern**. The goal is to create a unified interface to read temperature data from two different sensors (Analog LM34/35 via ADS1110 and Digital DHT11) using the lgpio library on Raspberry Pi.

2. Teams and Hardware

- **Groups:** Two teams (one 2-person, one 3-person).
- **Requirement:** Each group must include at least one student who has completed **CSCI 255**.
- **Equipment:** One Raspberry Pi 3, breadboard, ADS1110 ADC, LM34/35 sensor, and DHT11 sensor per group.

3. Technical Requirements

- **Library:** Use exclusively the lgpio Python library for hardware interaction.
- **Connectivity:**
    - **ADS1110 (Analog to $I^{2}C$):** Connect to Raspberry Pi pins (SDA/SCL).
    - **DHT11 (Digital):** Connect to a standard GPIO pin.
- **Environment:** Development will be done via **SSH** to the Raspberry Pi. Note: The Pi does not have internet access. You must manage your code locally and push to GitHub from your host machine.

4. Task Steps

1. **Research:** Read about the **Adapter Pattern** in the textbook and watch a tutorial online.
2. **Circuitry:** Build the circuit on the breadboard.
3. **Low-Level Drivers:** Write two "Black Box" classes that interact directly with the sensors using lgpio:
    - One to handle the communication with ADS1110 (to get LM34/35 values).
    - One to handle the timing-sensitive digital signal from DHT11.

4. **The Adapter:** Create a unified interface (Base Class) and two Adapters so that the main application can call .get_temperature() regardless of which sensor is plugged in.
5. **GitHub:** Create a repository and add the instructor as a collaborator.

5. Documentation Requirements (README.md):

Your repository must include a professional README.md with the following Mermaid diagrams:

1. **Class Diagram:** Representing the Adapter Pattern structure.
2. **Sequence Diagram:** Showing the flow of data from the physical sensor to the main application.
3. **Component Diagram:** Visualizing the interaction between the Python application, the lgpio library, and the hardware.

6. Deliverables & Grading

- **Sequential Commits:** Every team member must contribute. We expect a clear commit history showing the evolution of the project.
- **Live Demo:** Be prepared to show the system switching between sensors in the terminal.
- **Diagram:** A simple Mermaid class diagram in the README.md showing your Adapter implementation.