Project Status Report of Group 5

Week 2: Web Application Foundation and Mathematical Implementation

Project: Lock-Hub: IoT-Based Centralized Control System Development
Members:
Escaña, Jylie Anne R.
Lagunilla, Genesis L.
Llabores, Rodrigo III D.

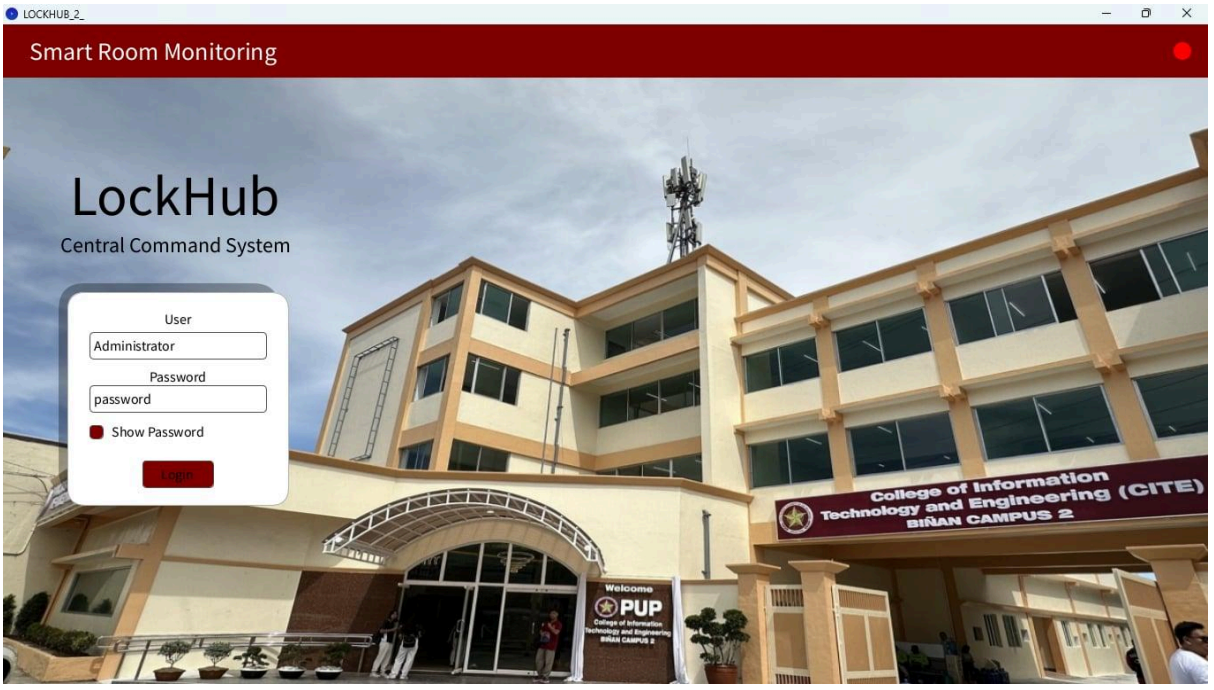
Web Application Framework

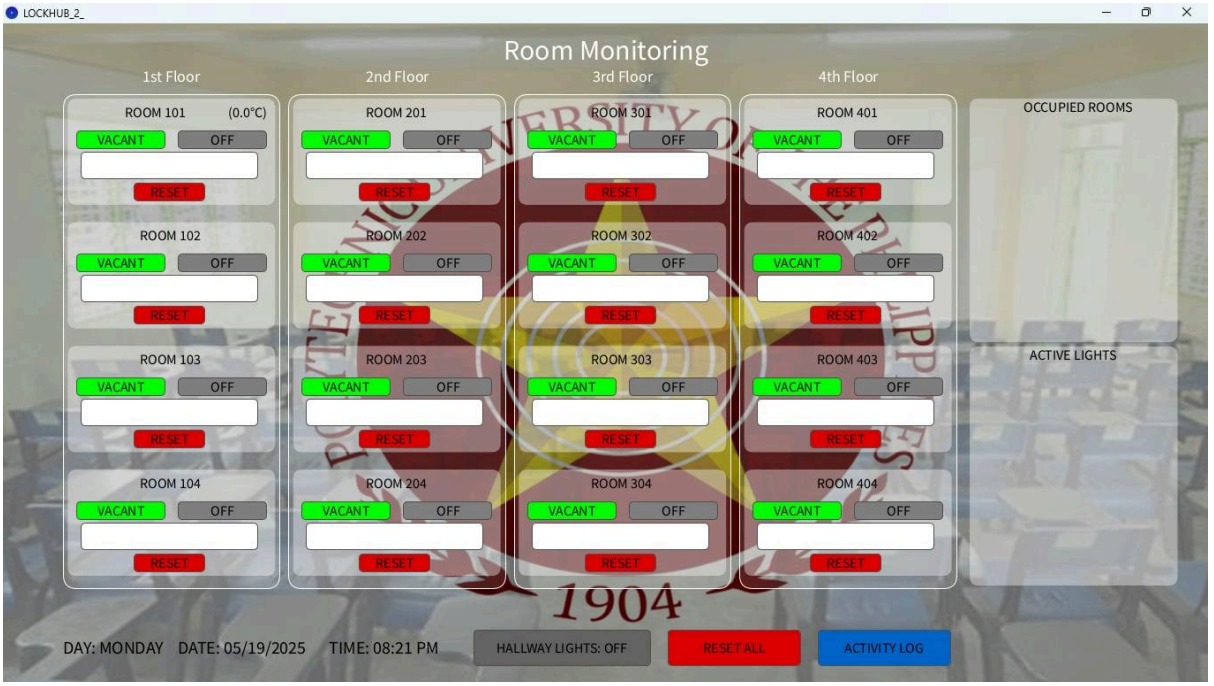
1. Backend API Design and Implementation for Sensor Data Access

The backend system is developed using the **Arduino IDE**, where the ESP32 microcontroller is programmed to collect and process environmental data from the **DHT22 sensor** (temperature). The data is refined using numerical method, the **Curve Fitting- Linear Regression** to provide smooth and trend-aware readings. These processed values are then sent via serial communication to a connected host system. This setup acts as a data service, continuously providing real-time sensor information to the application.

2. Frontend Layout and Component Structure

The frontend interface is built using **Processing**, a Java-based graphical programming environment ideal for interactive visualization. The layout is organized into functional panels, including real-time numerical displays of **temperature**, and **trend values**, along with color indicators to reflect system status. Visual components are updated based on the incoming serial data from the ESP32, allowing users to monitor the environment dynamically and clearly.





3. Data Visualization Components for At Least One Sensor Type

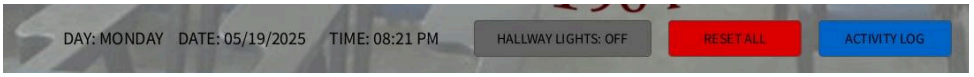
Temperature data is visualized through graphical elements like **line charts** showing historical EMA values, **bar meters** representing current readings, and **rate indicators** reflecting how quickly the temperature is changing. The system also computes and displays the **slope of the temperature trend** using linear regression, helping users interpret whether the temperature is rising or falling. These visuals provide intuitive insights into the behavior of the environment over time.

OCCUPIED/VACANT ----- Area status
ON/OFF ------ Lights Indicator
RESET ----- Set to orignal state of a certain area
RESET ALLL ---- All and all

4. Control Interface for System Actuators

The system includes a **physical button** connected to the ESP32, used to cycle display modes: **temperature**, This allows users to switch what information is shown or logged without needing a computer interface.

On the **user interface built with Processing**, each monitored **room or zone** is represented with its own panel that includes a **"Reset" button**. When clicked, this button clears the historical sensor data specific to that room, effectively restarting data tracking from zero. Additionally, a **"Reset All" button** is available in the main interface to reset sensor data for **all rooms simultaneously**. These buttons enhance user control, allowing for quick reinitialization of monitoring sessions, especially after significant environmental changes or system adjustments.



5. Deployment Plan for Web Application

The application is designed to run on a local computer, where **Processing** serves as the visualization frontend and receives data from the **ESP32 via USB serial connection**. The Arduino sketch uploaded to the ESP32 handles all sensor reading and data processing logic. For portable or embedded deployment, the system can be hosted on a device such as a **Raspberry Pi**, with Processing set to auto-launch on boot. This makes the project suitable for use as a **standalone environmental monitoring station**, without needing continuous development environment access.