UNIVERSITY OF MISSOURI
KANSAS CITY

Mini-project #4: Secure IoT Monitoring System with ESP32 and Web Dashboard

Objectives

- (mini-project1) Design and implement an IoT-based temperature and humidity monitoring system using ESP32 and DHT11.
- (mini-project2) Encrypt the sensor data before transmission and securely send it to a web server for processing.
- Store the received data in a database and visualize real-time temperature and humidity readings on a web interface.
- Implement a search function to filter historical temperature data efficiently and display past records as needed.
- **(Optional)** Implement a secure login system.

Project Requirements (1/3)

1. ESP32 (Client-Side):
   - Read DHT11 sensor data (temperature & humidity).
   - Encrypt the data before transmission.
   - Transmit data via HTTP POST to the web server every 10 seconds.

2. UI (Dashboard) Display:
   - Current temperature & humidity (real-time updates).
   - Search feature: Users can input a threshold to filter all past temperature readings above or below that value.
   - Real-time dynamic graph to visualize temperature trends.

Project Requirements (2/3)

3. Database (Encrypted Storage) in Web Server:
   - Store all received temperature & humidity data in a database.
   - Use AES encryption (or any secure method) before storing the data.

4. Search & Visualization:
   - Search function: Users can input a threshold value to retrieve past temperature readings above or below the specified value.
   - Real-time graph: Dynamically plot temperature data as it is received.

Project Requirements (3/3)

5. (Optional for extra credit: 3 points) Implement a Secure Login System:

- Login Page UI: A good-looking login page with username & password fields.
- User Authentication:
  - Users must log in before accessing the dashboard.
  - Credentials (username + password) must be hashed (e.g., bcrypt, Argon2) and stored in the database.
  - Session-based authentication (or JWT) to maintain login state.
- Logout Function: Users must be able to log out securely.

Debug Guide

- Web Server Issues:
  - Make sure the server is running if no data is received.
  - Verify the database connection and table structure if data is not stored.
  - Implement WebSockets or AJAX polling if the graph does not update in real-time.
  - Test the SQL query and ensure there is matching data if the search function is not working.

- Authentication Issues:
  - Enforce authentication middleware if users can access the dashboard without logging in.
  - Check password hashing and validation if login attempts fail.
  - Ensure the session or JWT is properly cleared if logout does not work.

Submissions

- Source Code: Provide well-documented code (Arduino code *.ino) with comments explaining each part of the code. **(Deadline: 11: 59 PM on March 30, 2025)**

- Project Report: A concise report describing the project's implementation and any challenges faced. **(Deadline: 11: 59 PM on March 30, 2025)**

- In-class Demo: Showcase the working project in real-time. **(Deadline: 5:30 PM – 6:45 PM on March 31, 2025)**

Evaluation Criteria

- Designing a secure IoT-based temperature and humidity monitoring system using ESP32 and DHT11, where encrypted sensor data is transmitted to a web server for real-time visualization, storage, and historical search functionality.

- **(Optional for extra credit)** Implement a secure login system requiring users to authenticate before accessing the dashboard, using hashed credentials and session-based authentication or JWT.

- Code quality (readability, comments, and structure).

- Clarity and thoroughness of the project report and demonstration.