SCHOOL OF COMPUTING

COLLEGE OF ARTS AND SCIENCES

SEMESTER A242 SESSION 2024/2025

STTHK3113 SENSOR-BASED SYSTEMS

MID TERM ASSIGNMENT

PREPARED FOR

DR. AHMAD HANIS BIN MOHD SHABIL

PREPARED BY

NAME	MATRIC
WAN MUHAMAD IKMAL BIN WAN MARZUKI	295154

GITHUB LINK:https://github.com/Podolskyy/SensorMidTem

YOUTUBE VIDEO LINK: https://youtu.be/p-t6AWef348

1.0 SYSTEM ARCHITECTURE DIAGRAM

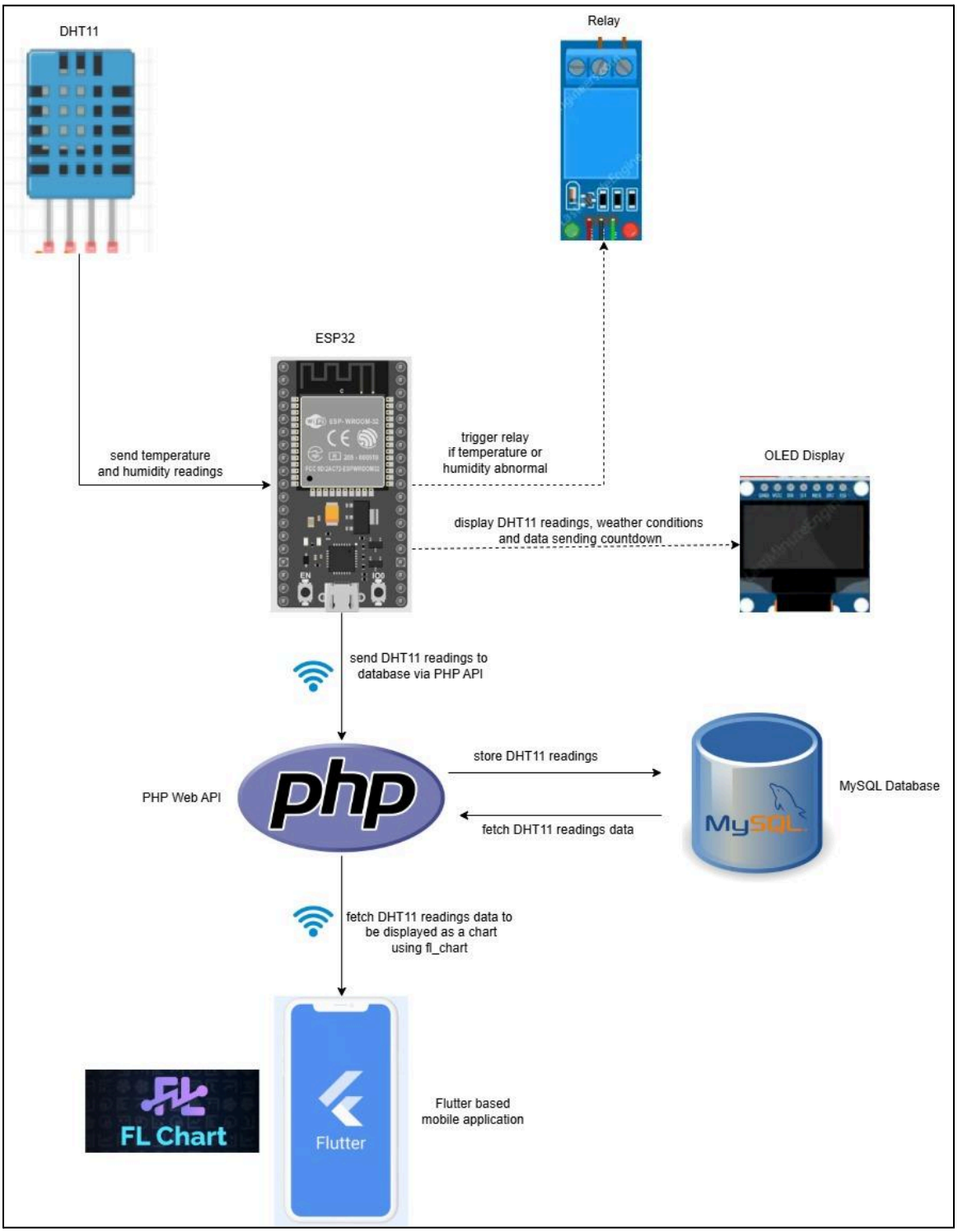


Figure 1.0: System Architecture Diagram

2.0 SETUP STEPS

2.1 Hardware

1. Connect DHT11:-

- + to 3.3v
- - to GND
- out to GPIO27 (D27)

2. Connect relay:-

   DC+ to 3.3v

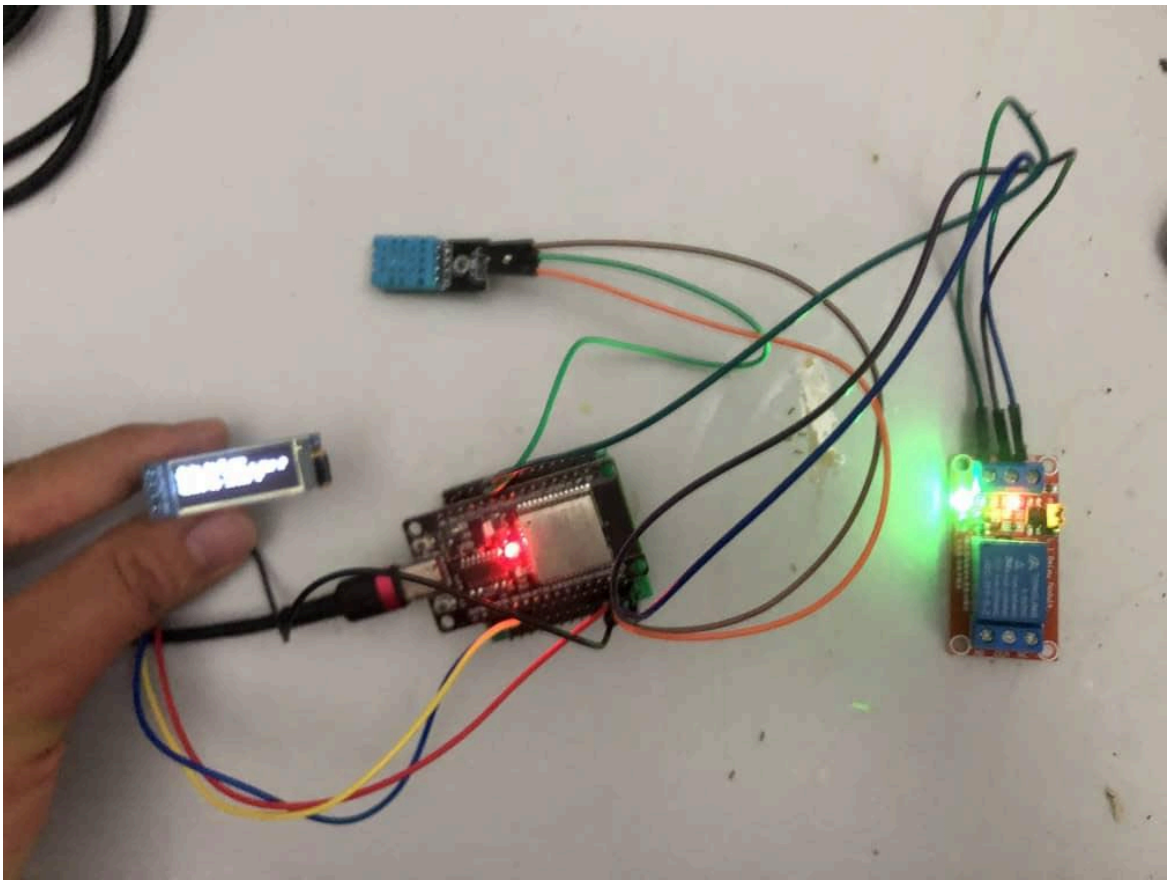
   DC- to GND

   out to GPIO26 (D26)

3. Connect OLED Display:-

- GND to GND
- VCC to 3.3v
- SCK to GPIO15 (D15)
- SDA to GPIO4 (D4)

The final hardware setup should look like this



2.2 Software

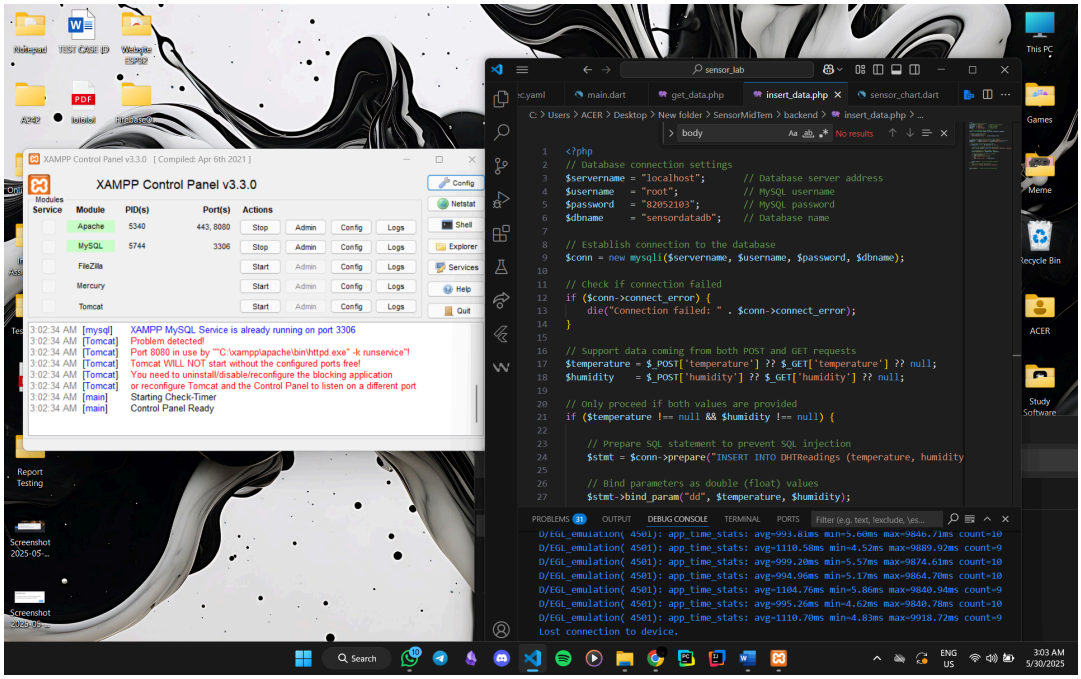
1. Create MySQL database and table readings with columns: id (PK), temperature, humidity, timestamp.
2. Upload PHP files (insert_data.php, get_data.php) to local host or web hosting server.
3. Configure PHP scripts with database credentials.
4. Test API endpoints with Postman or browser.

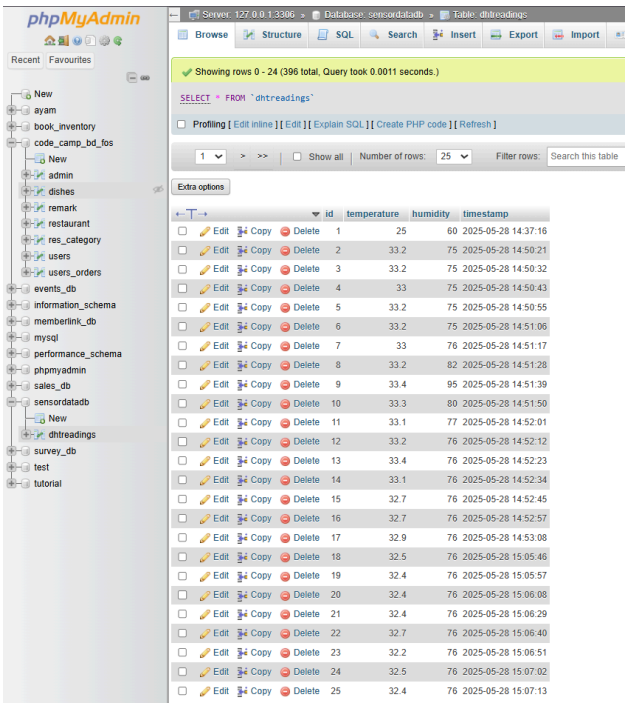
2.3 Server

1. Open Flutter project in VSCode.
2. Update API URL to point to your hosted backend.
3. Run flutter pub to install dependencies.
4. Build and run the app on a device or emulator.
5. Verify real-time data fetching and graph rendering.

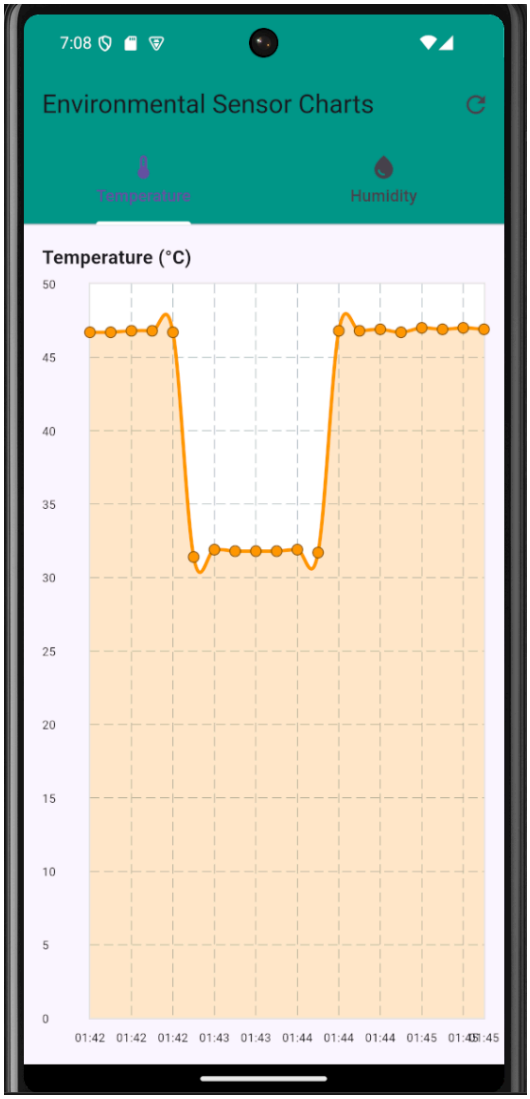
3.0 SCREENSHOTS

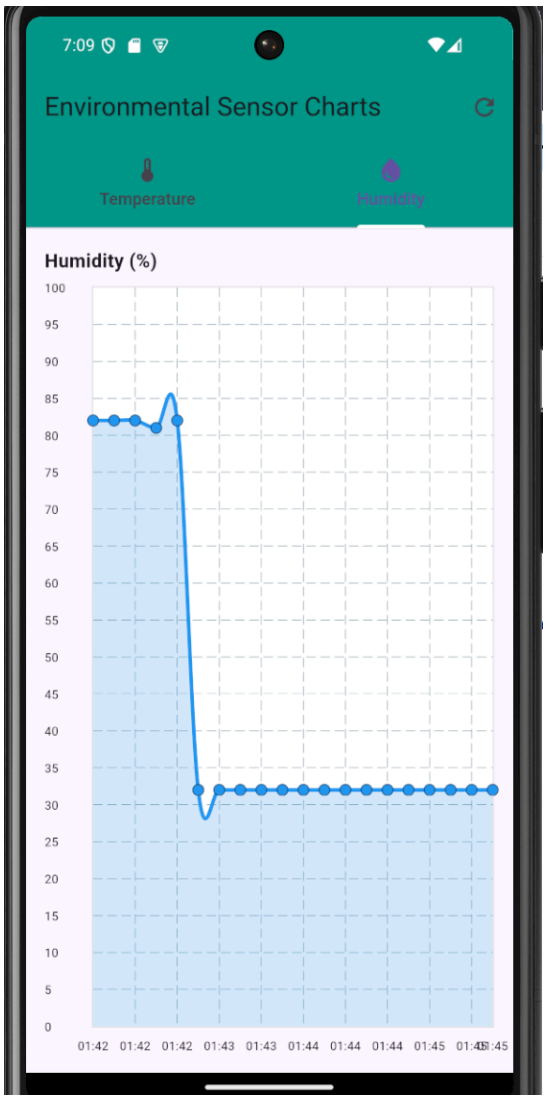
3.1 Backend



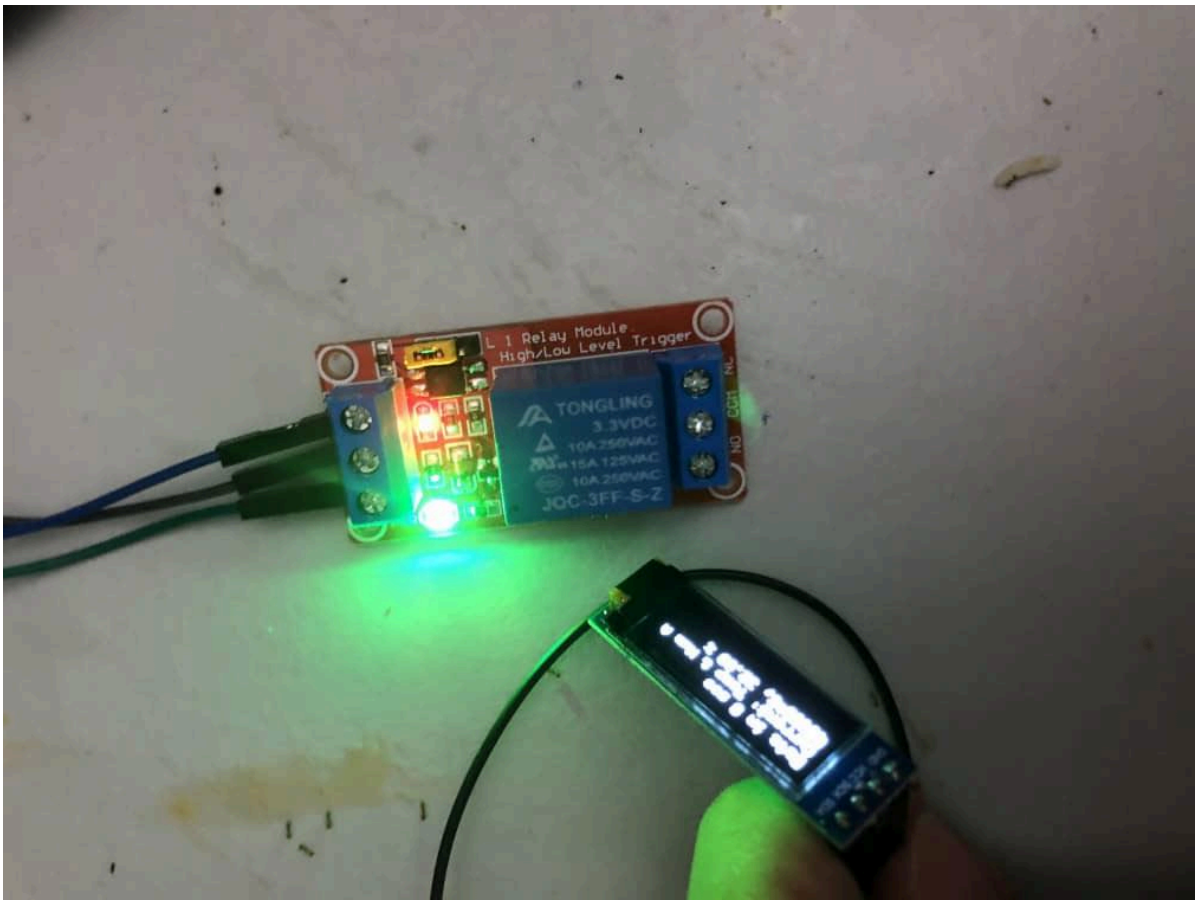


3.2 Mobile Application





3.3 Relay


1 Relay Module
High/Low Level Trigger
TONGLING
3.3VDC
10A 250VAC
15A 125VAC
10A 250VAC
JQC-3FF-S-Z
NO
COM
NC

4.0 CHALLENGES AND IMPROVEMENT

This project involved several key technical challenges and areas for improvement. One major task was managing the OLED display layout within the limited screen resolution to ensure clarity and usability. Another critical area was maintaining WiFi connection stability on the ESP32, including implementing robust reconnect logic. Additionally, issues such as HTTP 500 errors, which were caused by special characters in database credentials, needed to be resolved. On the frontend, efforts were made to fix Flutter package version incompatibilities and address chart rendering issues. Timing coordination was also essential, particularly in managing the fl_chart to update the readings consistently.

Several improvements have been proposed to enhance the system. These include adding a user interface in the app to allow dynamic threshold value settings, implementing authentication and data security measures for backend APIs, and expanding the system to support multiple sensors and devices. Further enhancements involve introducing notification alerts via email or SMS for threshold breaches and optimizing ESP32 power consumption to support battery-powered operations.