Project Progress Report: Plant Monitor IoT Dashboard

Date: August 25, 2025
Prepared for: Dr. Acharyya
Prepared by: Sandhya Patel

1. Project Objective

The goal of this project is to develop a fully functional Plant Monitoring IoT system that uses sensors (DHT22 and MP503) connected to an ESP32 microcontroller. The device transmits environmental data such as Temperature, Humidity, and VOC levels to an Android application via BLE (Bluetooth Low Energy), enabling real-time monitoring and alert management.

2. Hardware Setup

- Microcontroller: ESP32 DevKit v1 (connected and programmed via Arduino IDE)
 -Sensors:
• DHT22 (Temperature and Humidity Sensor)
 - connected to GPIO15 with 10k pull-up resistor between VCC and DATA.
 • MP503 VOC Sensor
– analog AOUT connected to GPIO34.
 - Breadboard and jumper wires used for prototyping.
- Wiring verified with reference diagrams and functional circuit assembled.
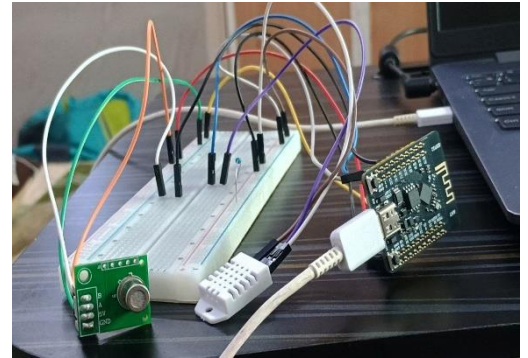
3. Firmware Status (Arduino)

- Successfully written and uploaded BLE server firmware on ESP32.
 - Data packet format: voc_raw,temp_celsius,humidity_percent
- Verified on Serial Monitor: Temperature , humidity and VOC data Transmitted correctly.
 - BLE advertising working with device name 'PlantSensor_01'.
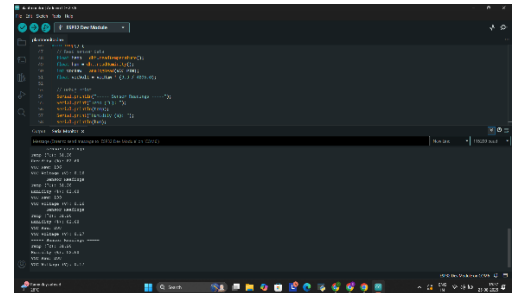
4. Android App Progress

- Developed PlantMonitor Android App using Kotlin and XML layouts. - Screens completed: Sensor Connection, Dashboard, Alerts, Trends.
 - Implemented BLE scanning and auto-pairing to ESP32 device.
 - Dashboard simulates live data; Alerts triggered using thresholds (VOC > 50 ppm, Temp > 35°C, Humidity < 40%).
 - BLEManager.kt supports GATT connection and notification listener.
 - Next step: finalize BLE integration with real data.
- Complete stable BLE data reception on Android using BLEManager
- Display real sensor data on charts (MPAndroidChart integration).

5. Challenges Faced

- BLE library conflict (ArduinoBLE vs esp32 BLE): resolved by removing ArduinoBLE. - DHT22 humidity readings missing: under investigation (wiring verified, resistor added). - Android BLE connection occasionally unstable: workaround using fallback debug path (Skip button). - Data persistence and historical export still to be implemented.

6. Next Steps

- Add SQLite or CSV export functionality for offline storage.
 - Polish UI, icons, and transitions for demo readiness.
-Prepare demo video and project documentation by next review.

Thank you for your continued guidance and support. We are now in the functional integration phase and expect a fully working prototype soon.