IOT SENSOR DATA FORECASTING SYSTEM

GROUP MEMBER :

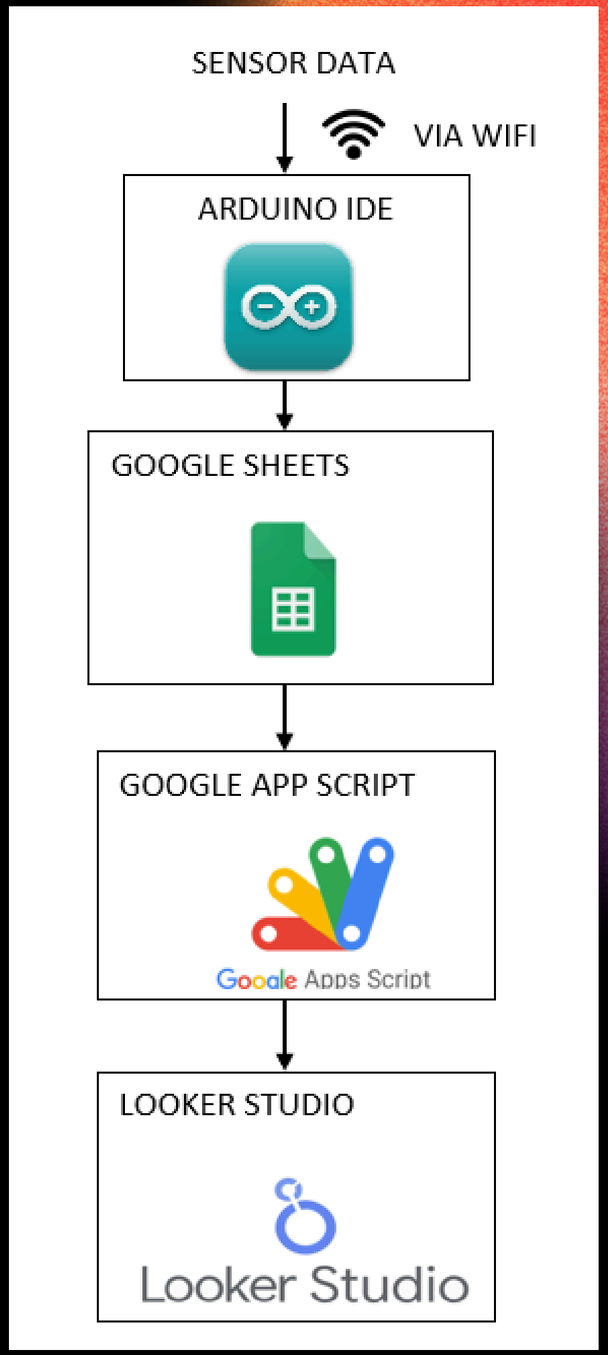
NUR FADHLIN IRDINA BINTI MOHD NISAM (VC21023)

MUHAMMAD ZAINI BIN ZAINAL ABIDIN (VC22009)

PROJECT OVERVIEW

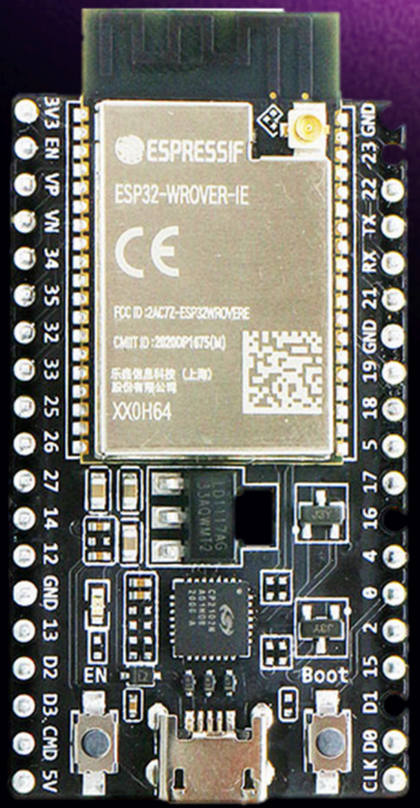
- To develop an IoT system using ESP32 and DHT11 to monitor temperature and humidity.
- To transmit real-time sensor data to Google Sheets via Google Apps Script.
- To implement data forecasting using time-series analysis methods in Apps Script.
- To visualize both live and forecasted data using Google Looker Studio.
- To demonstrate the integration of hardware, cloud services, and data analytics in a single IoT solution.

SYSTEM ARCHITECTURE DIAGRAM



HARDWARE

ESP32



DHT11

USB CABLE

SOFTWARE

ARDUINO IDE

GOOGLE SPREADSHEET

GOOGLE APP SCRIPT

LOCKER STUDIO

ESP32 CODE HIGHLIGHT

(FULL CODE AVAILABLE IN GITHUB)

✅ Sensor Initialization:

```
// DHT setup
#define DHTPIN 26
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
```

✅ HTTP POST Setup

```
HTTPClient http;
http.begin(scriptURL);
http.addHeader("Content-Type", "application/json");

String jsonData = "{\"temperature\": " + String(temperature) + ", \"humidity\": " + String(humidity) + "}";

int httpResponseCode = http.POST(jsonData);
```

ESP32 CODE HIGHLIGHT

✅ Validation Logic

```
if (isnan(humidity) || isnan(temperature)) {
  Serial.println("Failed to read from DHT sensor!");
  return;
}
```

✅ Additional Validation Function

```
// Data validation function
bool validateSensorReading(float reading) {
  if (isnan(reading)) {
    return false;
  }

  if (reading < -40.0 || reading > 80.0) {
    return false;
```

APPS SCRIPT LOGIC

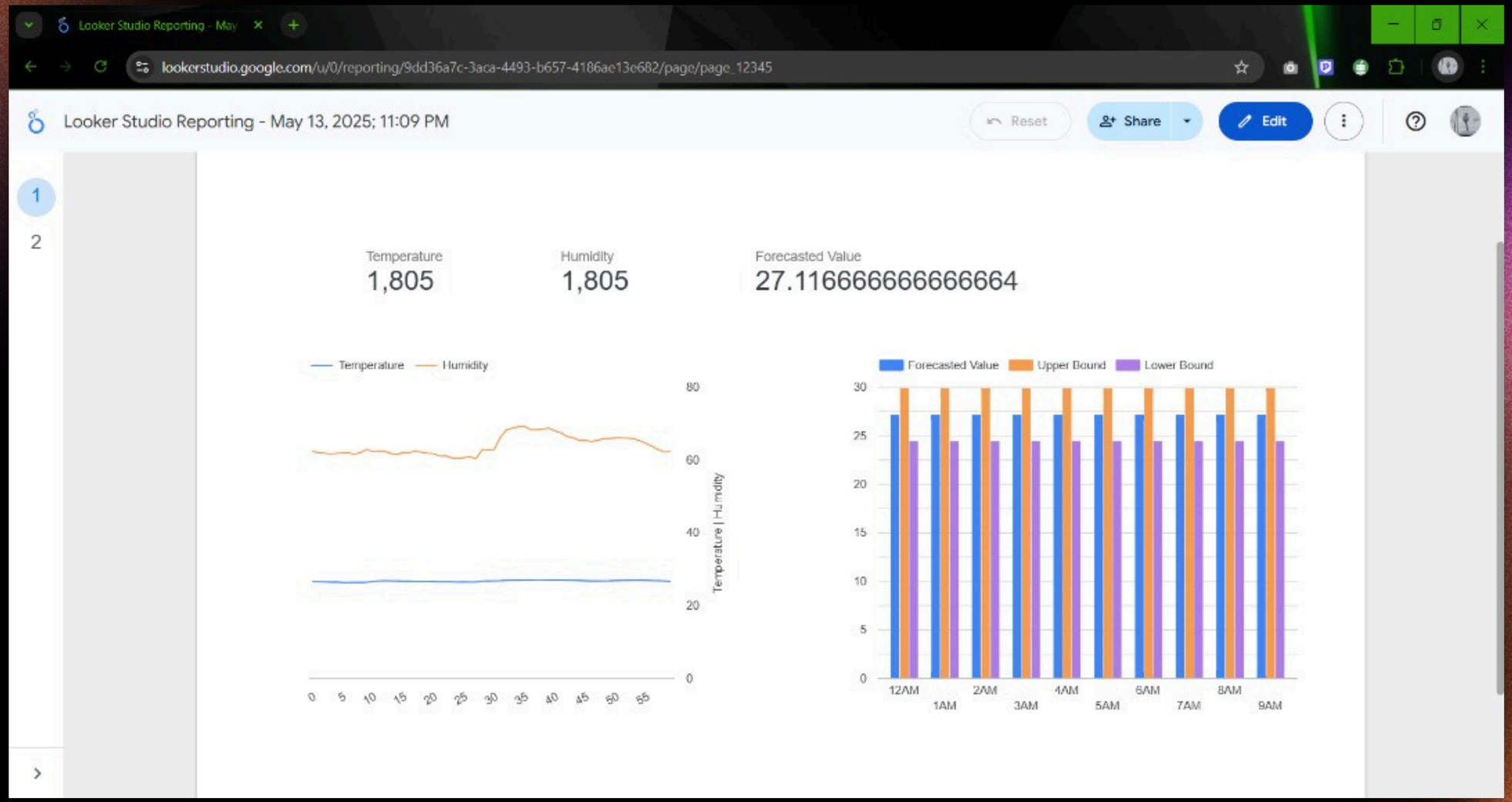
(FULL CODE IN GITHUB)

- Script receives sensor data in a “SensorData” sheet
- Function generateForecasts() runs to compute 24-hour predictions
- Chosen algorithm: Simple Moving Average (SMA) using past 6 data points
- Results saved in a separate “Forecasts” sheet
- Includes confidence bounds (±10%)

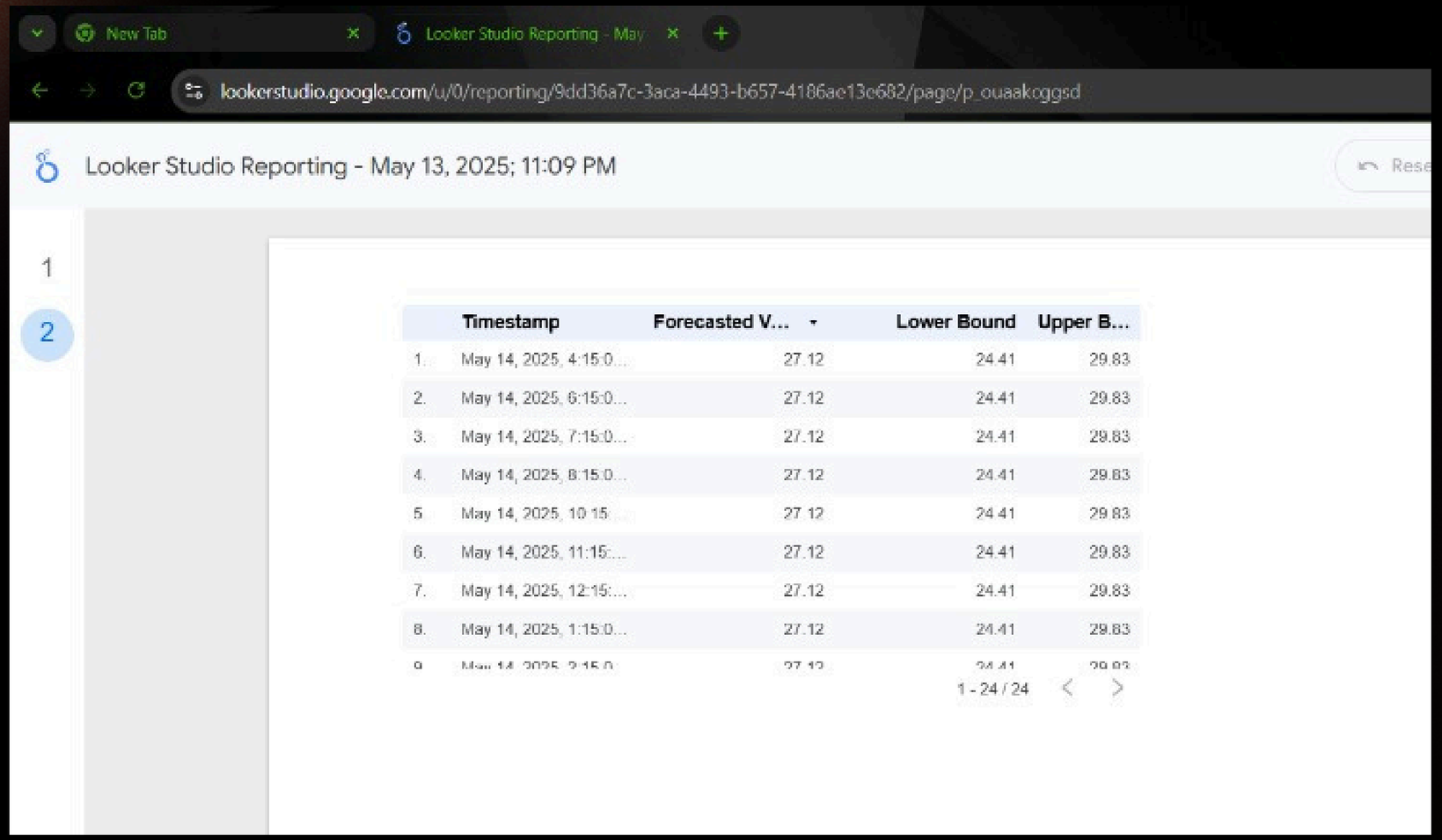
FORECAST SHEET STRUCTURE

	A	B	C	D
1	Timestamp	Forecasted Value	Upper Bound	Lower Bound
2	5/14/2025 1:35:15	27.16666667	29.88333333	24.45
3	5/14/2025 2:35:15	27.16666667	29.88333333	24.45
4	5/14/2025 3:35:15	27.16666667	29.88333333	24.45
5	5/14/2025 4:35:15	27.16666667	29.88333333	24.45
6	5/14/2025 5:35:15	27.16666667	29.88333333	24.45
7	5/14/2025 6:35:15	27.16666667	29.88333333	24.45
8	5/14/2025 7:35:15	27.16666667	29.88333333	24.45
9	5/14/2025 8:35:15	27.16666667	29.88333333	24.45
10	5/14/2025 9:35:15	27.16666667	29.88333333	24.45
11	5/14/2025 10:35:15	27.16666667	29.88333333	24.45
12	5/14/2025 11:35:15	27.16666667	29.88333333	24.45
13	5/14/2025 12:35:15	27.16666667	29.88333333	24.45
14	5/14/2025 13:35:15	27.16666667	29.88333333	24.45
15	5/14/2025 14:35:15	27.16666667	29.88333333	24.45
16	5/14/2025 15:35:15	27.16666667	29.88333333	24.45
17	5/14/2025 16:35:15	27.16666667	29.88333333	24.45
18	5/14/2025 17:35:15	27.16666667	29.88333333	24.45

DASHBOARD VISUALIZATION



DASHBOARD VISUALIZATION



	Timestamp	Forecasted V...	Lower Bound	Upper B...
1.	May 14, 2025, 4:15:0...	27.12	24.41	29.83
2.	May 14, 2025, 6:15:0...	27.12	24.41	29.83
3.	May 14, 2025, 7:15:0...	27.12	24.41	29.83
4.	May 14, 2025, 8:15:0...	27.12	24.41	29.83
5.	May 14, 2025, 10:15:...	27.12	24.41	29.83
6.	May 14, 2025, 11:15:...	27.12	24.41	29.83
7.	May 14, 2025, 12:15:...	27.12	24.41	29.83
8.	May 14, 2025, 1:15:0...	27.12	24.41	29.83
9.	May 14, 2025, 2:15:0...	27.12	24.41	29.83

1 - 24 / 24

IMPLEMENTATION CHALLENGES AND SOLUTIONS

- WiFi instability → added reconnection logic
- Sensor NaN readings → implemented data validation
- Apps Script execution limits → optimized data size and forecast interval
- ESP32 timing → used fixed 2-second interval

FUTURE IMPROVEMENTS

- Use EMA or Holt-Winters for better responsiveness
- Add mobile notification alerts
- Deploy in real life not dummy data

THANK YOU