اونيۏرسيتي مليسيا ڤهڠ السلطان عبد الله

UNIVERSITI MALAYSIA PAHANG
AL-SULTAN ABDULLAH

FACULTY OF ELECTRICAL AND ELECTRONIC ENGINEERING TECHNOLOGY

BVI 3114 TECHNOLOGY SYSTEM OPTIMIZATION II

SEMESTER 5 2025/2026

Lecturer:
Dr. Mohd Zamri Bin Ibrahim

Report:
Real-time Sensor Data Collection with ESP32 and Google Sheets Integration

Prepared By	MUHAMMAD ARIF BIN AMRAN (VC23004)
	MAS IZZALIANA BINTI MOHAMAD MOKHTAR (VC23035)

Table of Content

Timeline and Milestones

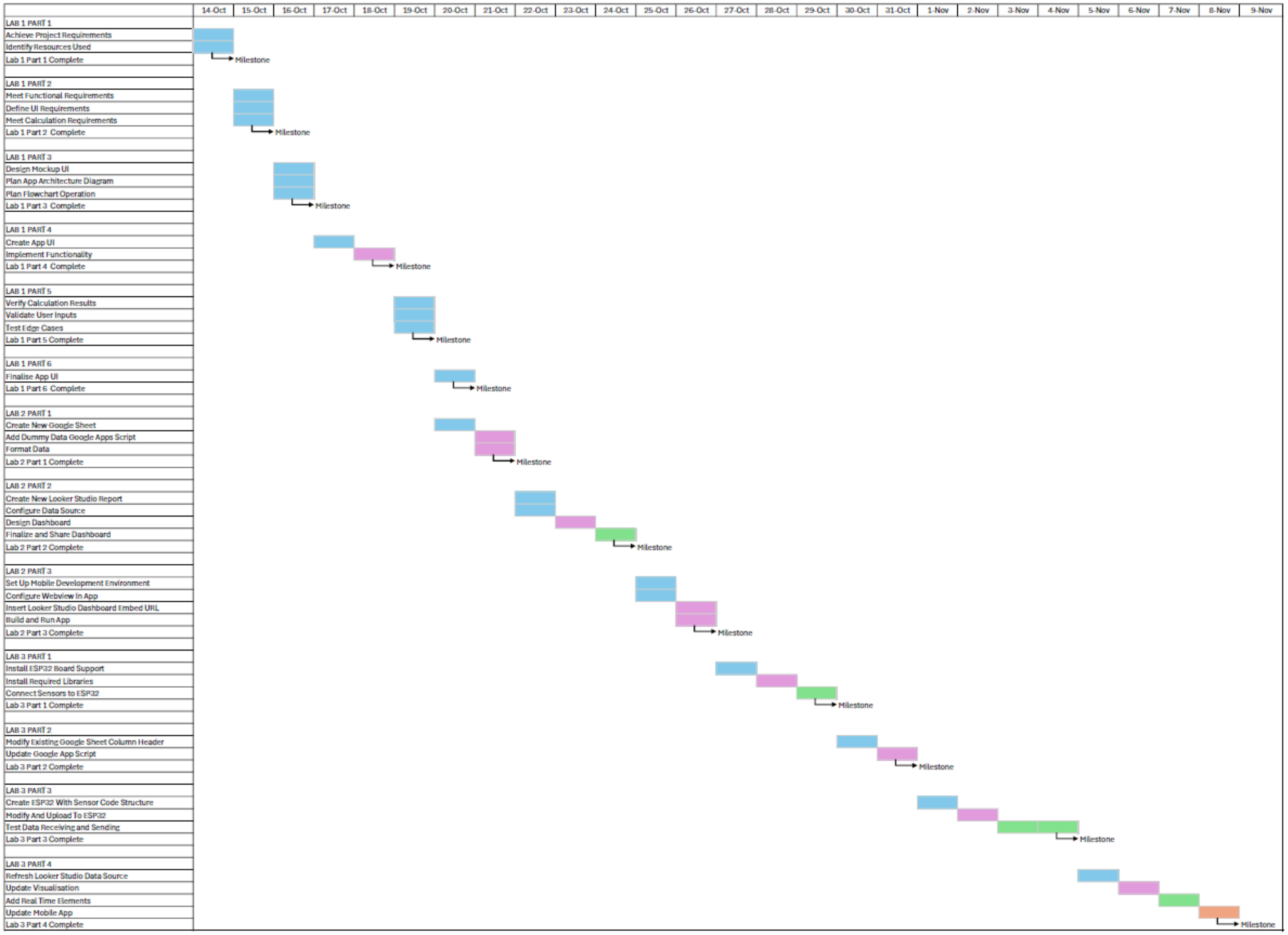


Figure 1: Timeline and Milestones.

Description of Hardware Setup

For this next lab exercise, the previous lab will be extended to replacing the dummy data with real sensor readings from the assigned ESP32 microcontroller, HC-SR04 Ultrasonic sensor and BME280 Temperature and Humidity Sensor.

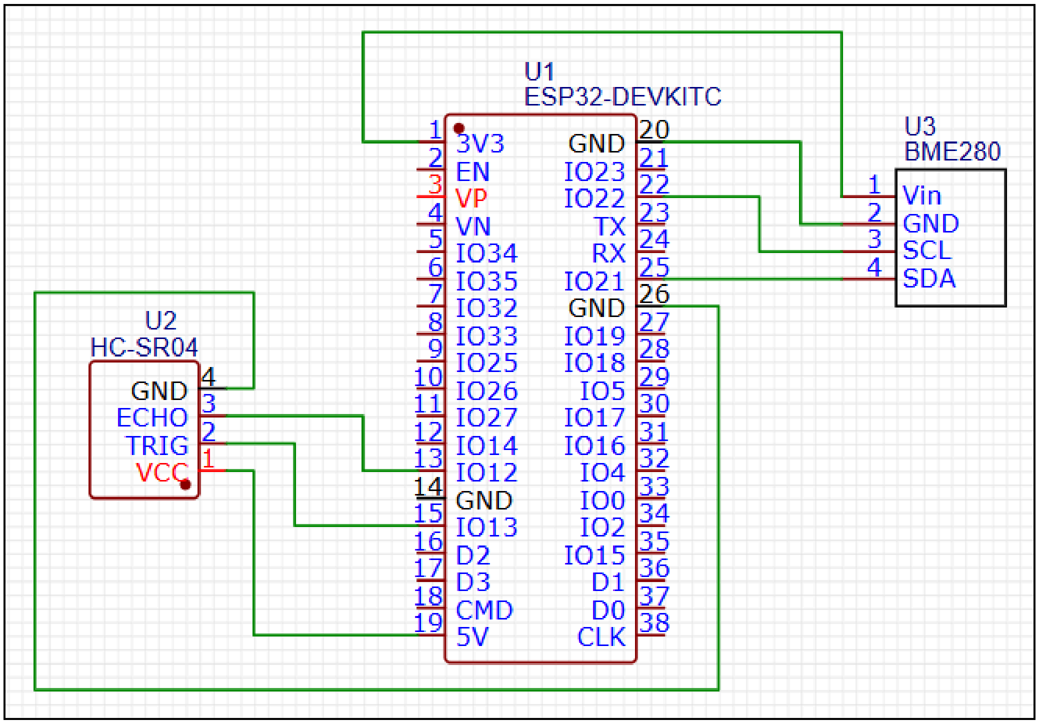


Figure 2: ESP32 Connection With HC-SR04 Ultrasonic sensor and BME280 Temperature and Humidity Sensor Schematic Diagram.

As shown in Figure 2, the wiring for ESP32 with BME280 Temperature and Humidity sensor and HC-SR04 Ultrasonic sensor are as follow;

a. BME280 Temperature and Humidity Sensor Wiring:
   - Vin Pin will connect to 3.3V ESP32.
   - GND Pin will connect to GND ESP32.
   - SCL Pin will connect to I2C Pin GPIO22 ESP32.
   - SDA Pin will connect to I2C Pin GPIO 21 ESP32.

b. HC-SR04 Ultrasonic Sensor Wiring:
   - VCC Pin will connect to 5V ESP32.
   - GND Pin will connect to GND ESP32.
   - ECHO Pin will connect to GPIO12 ESP32.
   - TRIG Pin will connect to GPIO13 ESP32.

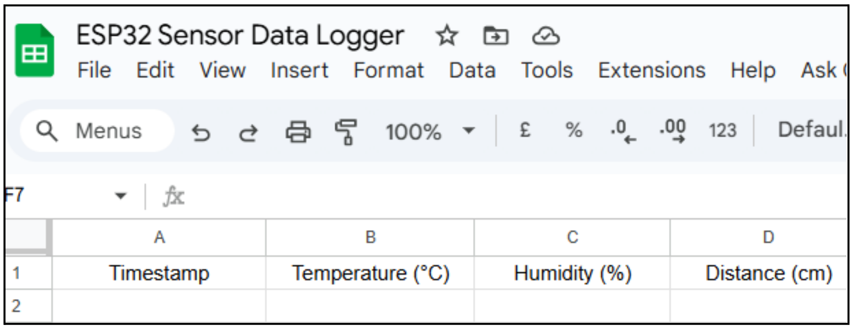
Implementation

Part 1: Creating a Data Source with Google Sheets

Step 1.1: Create a New Google Sheet

Create a new spreadsheet in Google Sheets and rename it to “ESP32 Sensor Data Logger”. Then create the four column headers in row 1 which is:

- A1: Timestamp.
- B1: Temperature (°C).
- C1: Humidity (%).
- D1: Distance (cm).



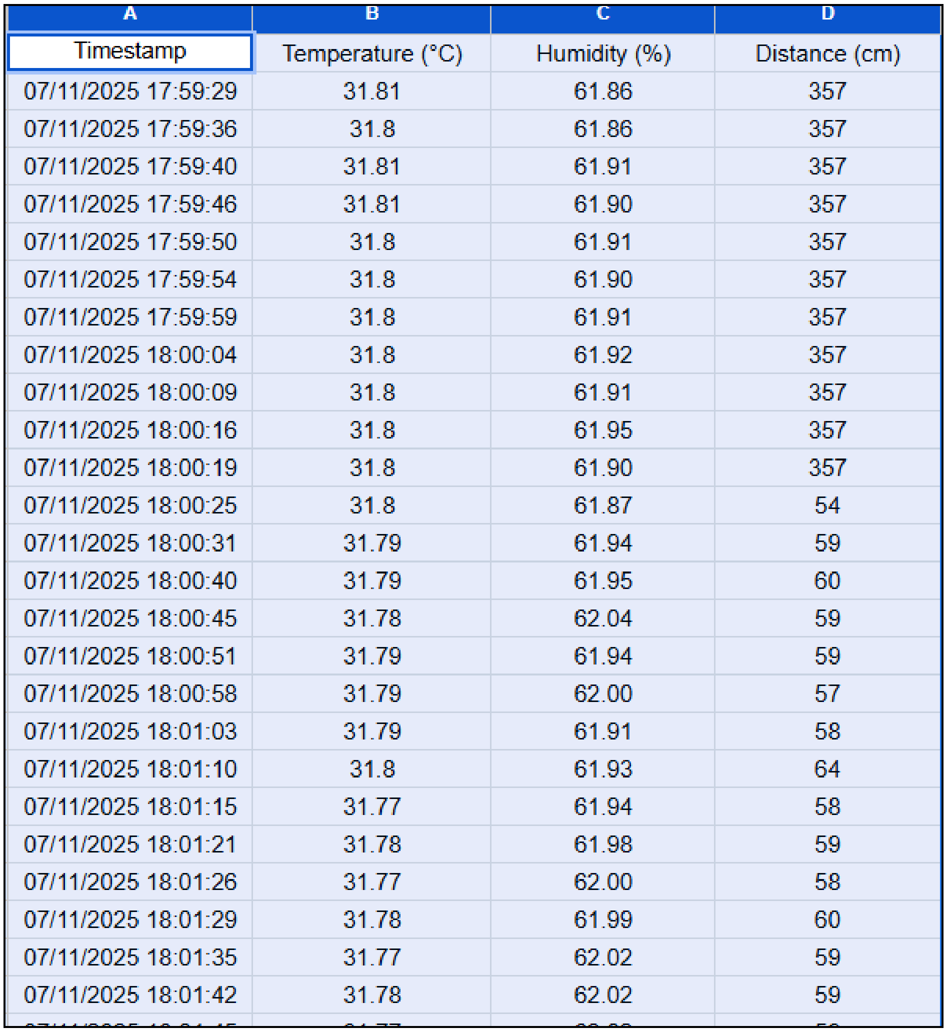
	A	B	C	D
1	Timestamp	Temperature (°C)	Humidity (%)	Distance (cm)
2				

Figure 3: Created a Sheet with Timestamp, Temperature (°C), Humidity (%) and Distance (cm) columns.

Step 1.2: Format Data

Select all four columns filled with data and click Number on the Format menu. Then choose the appropriate formats for each columns:

- Timestamp column: Date time format.
- Temperature column: Number with 1 decimal place.
- Humidity column: Number with no decimal places.
- Distance column: Number with no decimal places.



A	B	C	D
Timestamp	Temperature (°C)	Humidity (%)	Distance (cm)
07/11/2025 17:59:29	31.81	61.86	357
07/11/2025 17:59:36	31.8	61.86	357
07/11/2025 17:59:40	31.81	61.91	357
07/11/2025 17:59:46	31.81	61.90	357
07/11/2025 17:59:50	31.8	61.91	357
07/11/2025 17:59:54	31.8	61.90	357
07/11/2025 17:59:59	31.8	61.91	357
07/11/2025 18:00:04	31.8	61.92	357
07/11/2025 18:00:09	31.8	61.91	357
07/11/2025 18:00:16	31.8	61.95	357
07/11/2025 18:00:19	31.8	61.90	357
07/11/2025 18:00:25	31.8	61.87	54
07/11/2025 18:00:31	31.79	61.94	59
07/11/2025 18:00:40	31.79	61.95	60
07/11/2025 18:00:45	31.78	62.04	59
07/11/2025 18:00:51	31.79	61.94	59
07/11/2025 18:00:58	31.79	62.00	57
07/11/2025 18:01:03	31.79	61.91	58
07/11/2025 18:01:10	31.8	61.93	64
07/11/2025 18:01:15	31.77	61.94	58
07/11/2025 18:01:21	31.78	61.98	59
07/11/2025 18:01:26	31.77	62.00	58
07/11/2025 18:01:29	31.78	61.99	60
07/11/2025 18:01:35	31.77	62.02	59
07/11/2025 18:01:42	31.78	62.02	59

Figure 4: Timestamp Column Data Formatted to Date Time, Temperature Column Data Formatted to 1 Decimal Place Number, Humidity and Distance Column Data Formatted to No Decimal Place Number.

Part 2: Creating a Looker Studio Dashboard

Step 2.1: Create New Looker Studio Report

Go to Looker Studio, click the Create button and click Report. Choose Google Sheets at the data source selection and select where Temperature Humidity Sensor Data sheet are located. Then click ADD TO REPORT.

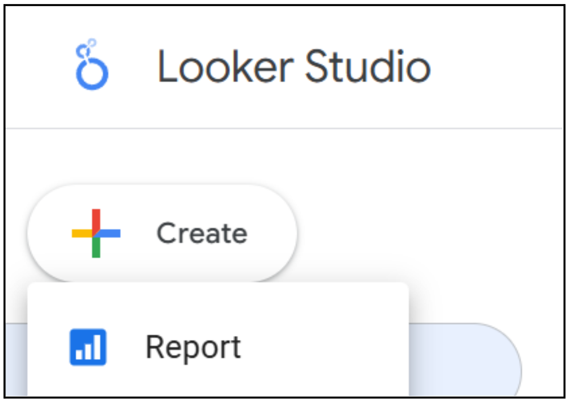
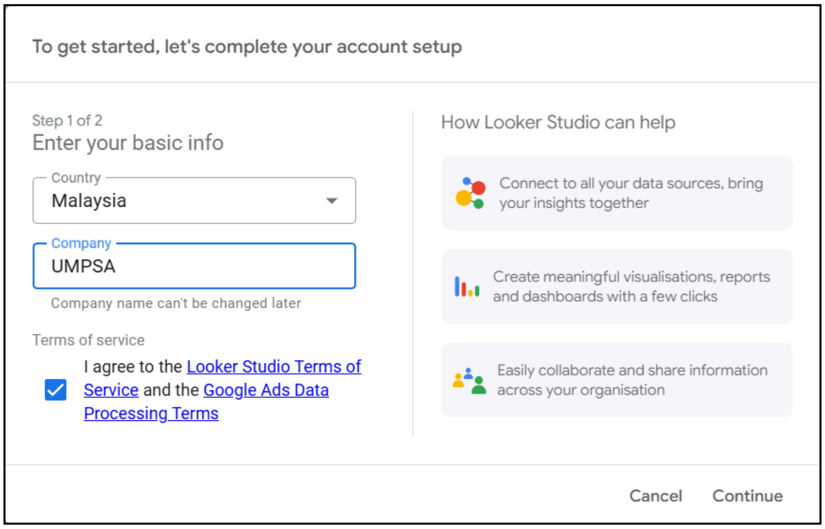


Figure 5: Create a New Report by Clicking the Create Button.



To get started, let's complete your account setup

Step 1 of 2

Enter your basic info

Country: Malaysia

Company: UMPSA

Company name can't be changed later

Terms of service

☑ I agree to the Looker Studio Terms of Service and the Google Ads Data Processing Terms

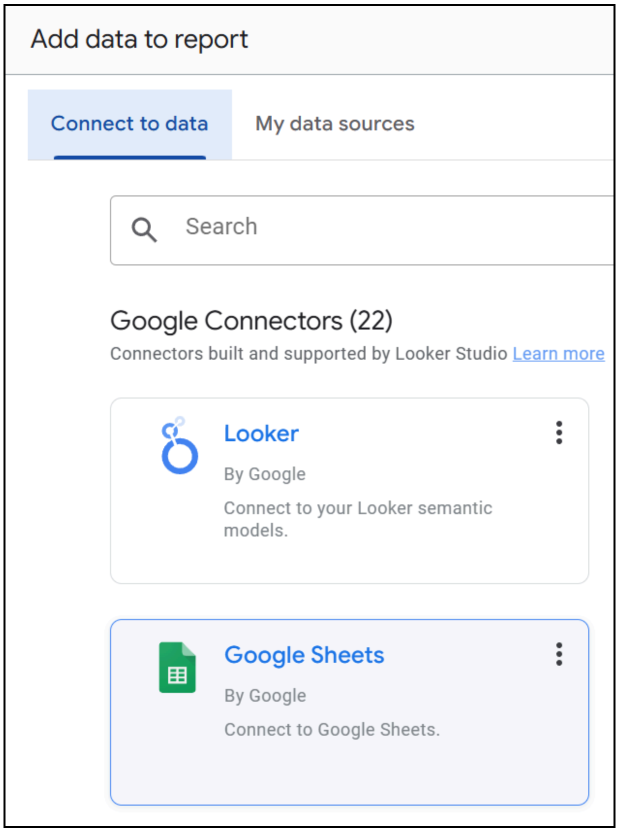
How Looker Studio can help

Connect to all your data sources, bring your insights together

Create meaningful visualisations, reports and dashboards with a few clicks

Easily collaborate and share information across your organisation

Figure 6: Enter the Basic Info Before Continuing.



Add data to report

Connect to data

My data sources

Search

Google Connectors (22)

Connectors built and supported by Looker Studio Learn more

Looker

By Google

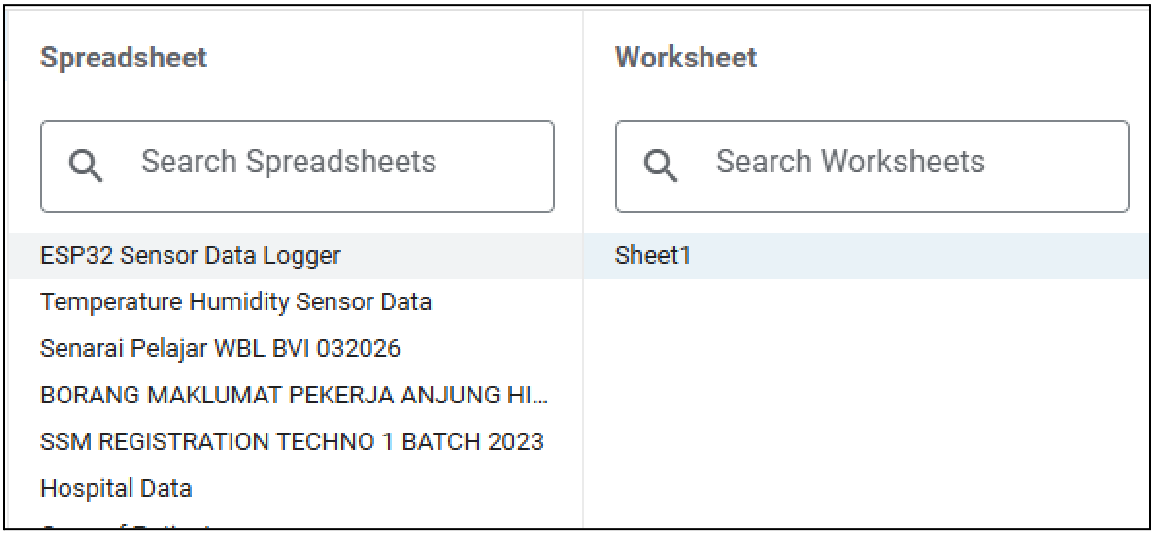
Connect to your Looker semantic models.

Google Sheets

By Google

Connect to Google Sheets.

Figure 7: Select Google Sheets to Connect and Add Data to Report.



Spreadsheet

Search Spreadsheets

ESP32 Sensor Data Logger

Temperature Humidity Sensor Data

Senarai Pelajar WBL BVI 032026

BORANG MAKLUMAT PEKERJA ANJUNG HI...

SSM REGISTRATION TECHNO 1 BATCH 2023

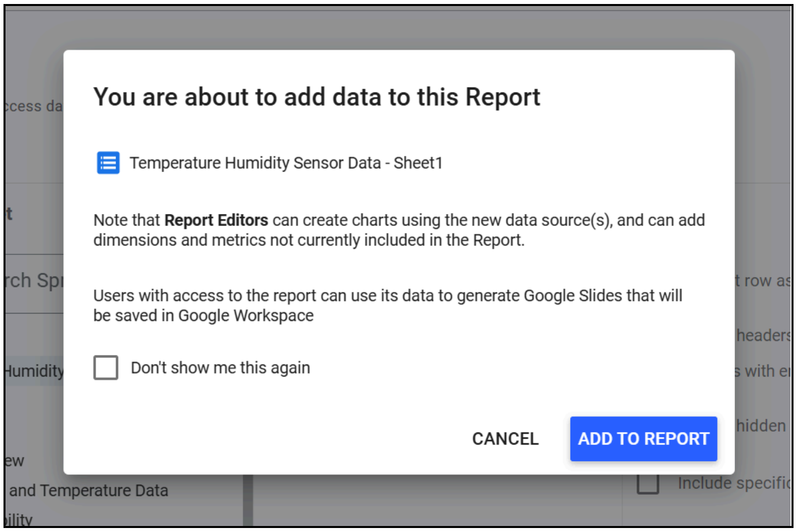
Hospital Data

Worksheet

Search Worksheets

Sheet1

Figure 8: Select the ESP32 Sensor Data Logger Spreadsheet.



You are about to add data to this Report

Temperature Humidity Sensor Data - Sheet1

Note that **Report Editors** can create charts using the new data source(s), and can add dimensions and metrics not currently included in the Report.

Users with access to the report can use its data to generate Google Slides that will be saved in Google Workspace

□ Don't show me this again

Figure 9: Confirm the Selected Data to Be Added to Report.

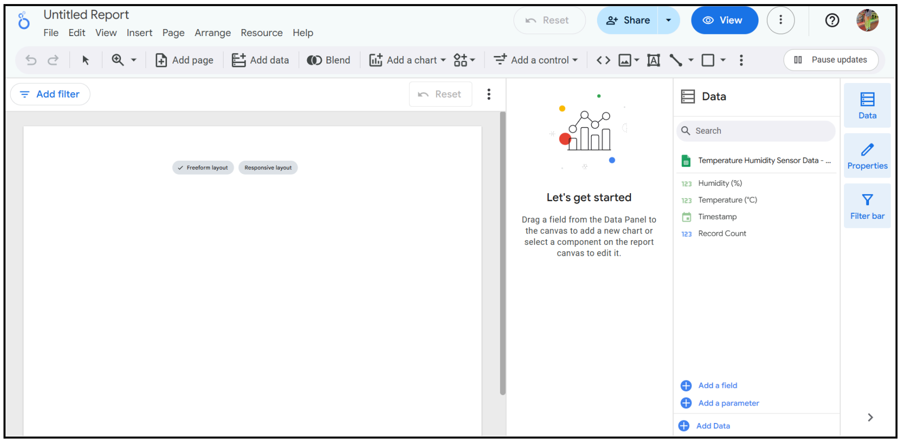


Figure 10: Blank Dashboard Report With Temperature Humidity Sensor Data Spreadsheet Resources.

Step 2.2: Configure Data Source

The data types in Looker Studio is correctly identified where Timestamp should be a Date & Time, Temperature, Humidity and Distance should be a numeric metric.

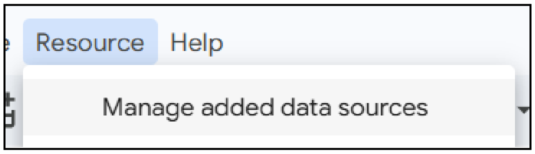
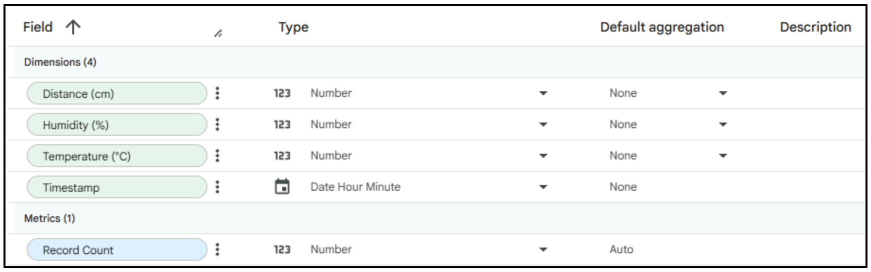


Figure 11: Select Manage added data sources to View Data Types.

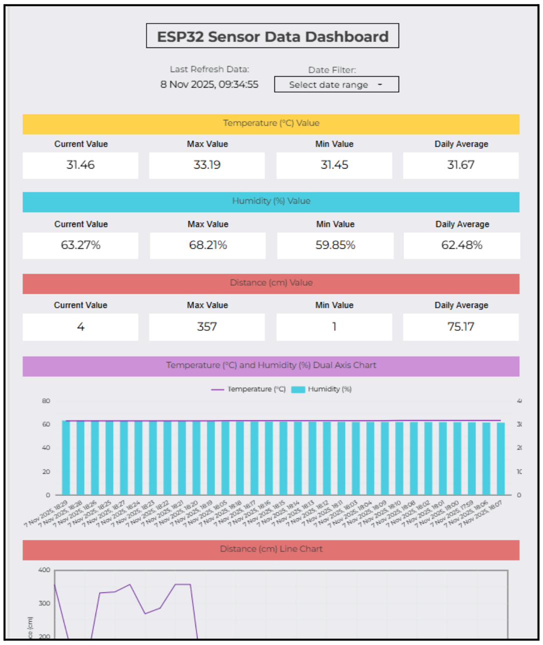


Field ↑	Type	Default aggregation	Description
Dimensions (4)			
Distance (cm)	123 Number	None	
Humidity (%)	123 Number	None	
Temperature (°C)	123 Number	None	
Timestamp	Date Hour Minute	None	
Metrics (1)			
Record Count	123 Number	Auto	

Figure 12: Temperature, Humidity and Distance are Number Data Type With Average Default Aggregation and Timestamp are Date Hour Minute Data Type With None Default Aggregation.

Step 2.3: Design Dashboard

In the new report, the dashboard will be added title, data range control, time series chart, dual axis chart and separate metrics scorecard for average, min, max and current values of temperature, humidity and distance.



ESP32 Sensor Data Dashboard

Last Refresh Data: 8 Nov 2025, 09:34:55

Date Filter: Select date range

Temperature (°C) Value

Current Value	Max Value	Min Value	Daily Average
31.46	33.19	31.45	31.67

Humidity (%) Value

Current Value	Max Value	Min Value	Daily Average
63.27%	68.21%	59.85%	62.48%

Distance (cm) Value

Current Value	Max Value	Min Value	Daily Average
4	357	1	75.17

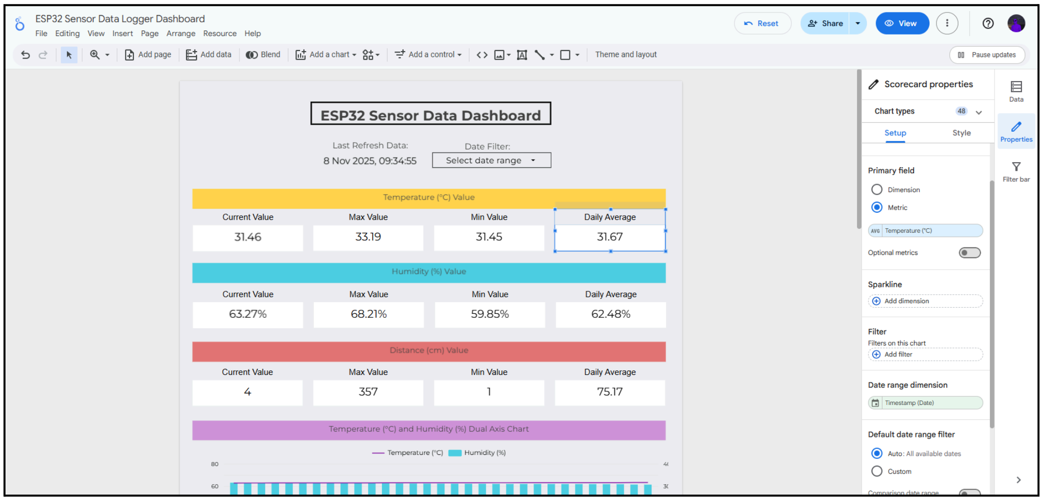
Temperature (°C) and Humidity (%) Dual Axis Chart

Distance (cm) Line Chart

Figure 13: Dashboard Design Included With Title, Date Range Control, Time Series Chart, Dual Axis Chart and Separate Metrics Scorecard of Average, Min, Max and Current Values of Temperature, Humidity and Distance Data.

Step 2.4: Finalize and Share Dashboard

Click View mode to test dashboard functionality and Edit mode for any adjustments needed.



ESP32 Sensor Data Dashboard

Last Refresh Data: 8 Nov 2025, 09:34:55

Date Filter: Select date range

Temperature (°C) Value

Current Value	Max Value	Min Value	Daily Average
31.46	33.19	31.45	31.67

Humidity (%) Value

Current Value	Max Value	Min Value	Daily Average
63.27%	68.21%	59.85%	62.48%

Distance (cm) Value

Current Value	Max Value	Min Value	Daily Average
4	357	1	75.17

Temperature (°C) and Humidity (%) Dual Axis Chart

Temperature (°C) 🟦 Humidity (%)

Scorecard properties

Chart types 48

Setup Style

Primary field
○ Dimension
● Metric

AVG Temperature (°C)

Optional metrics

Sparkline
Add dimension

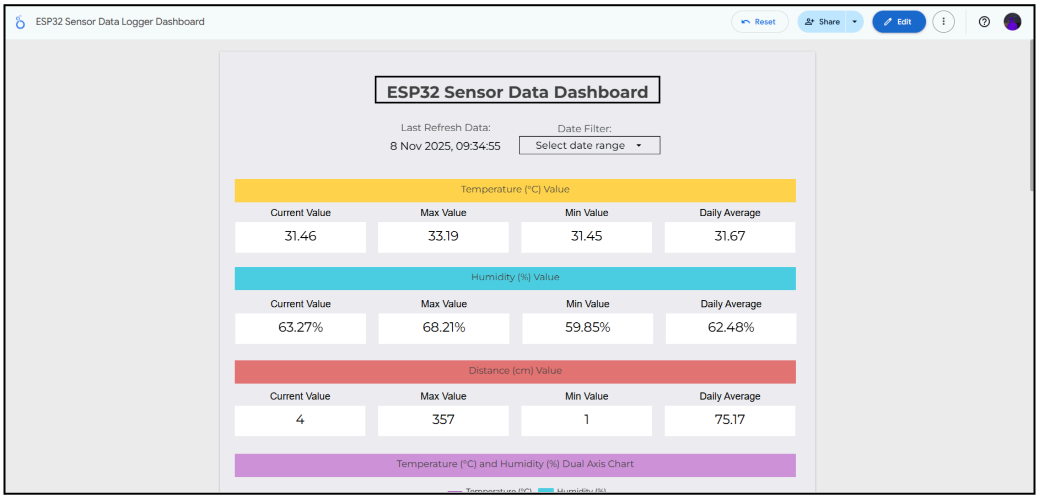
Filter
Filters on this chart
Add filter

Date range dimension
Timestamp (Date)

Default date range filter
● Auto: All available dates
○ Custom

Comparison date range

Figure 14: Looker Studio Edit Mode.



ESP32 Sensor Data Dashboard

Last Refresh Data: 8 Nov 2025, 09:34:55

Date Filter: Select date range

Temperature (°C) Value

Current Value	Max Value	Min Value	Daily Average
31.46	33.19	31.45	31.67

Humidity (%) Value

Current Value	Max Value	Min Value	Daily Average
63.27%	68.21%	59.85%	62.48%

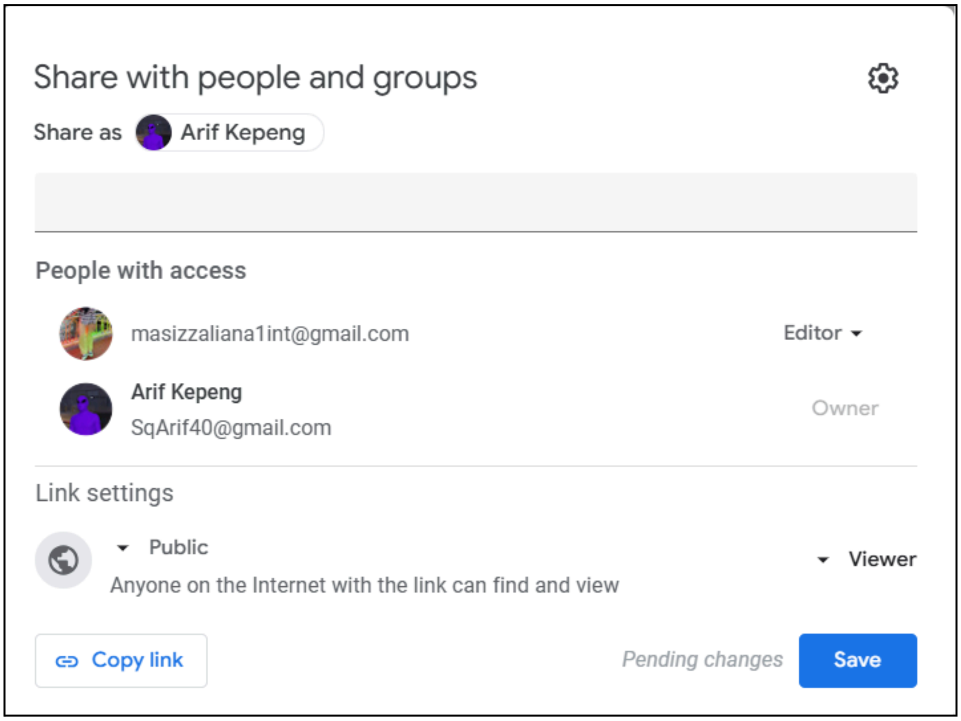
Distance (cm) Value

Current Value	Max Value	Min Value	Daily Average
4	357	1	75.17

Temperature (°C) and Humidity (%) Dual Axis Chart

Figure 15: Looker Studio View Mode.

Then click Share in top right corner and set the access of the dashboard from Restricted to Anyone on the Internet with the link can find and view.



Share with people and groups

Share as Arif Kepeng

People with access

masizzaliana1int@gmail.com Editor

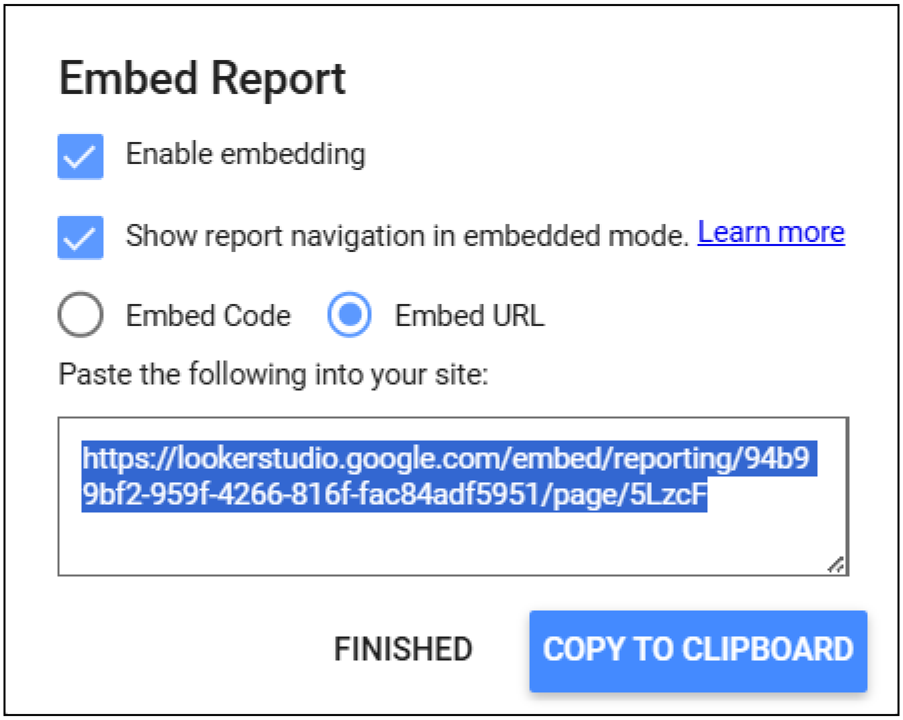
Arif Kepeng
SqArif40@gmail.com Owner

Link settings

Public
Anyone on the Internet with the link can find and view
Viewer

Figure 16: Changed the Access of Dashboard to Public but Viewer Only.

Get the dashboard embed URL by opening Embed Report from the File menu.



Embed Report

☑ Enable embedding

☑ Show report navigation in embedded mode. Learn more

○ Embed Code ● Embed URL

Paste the following into your site:

```
https://lookerstudio.google.com/embed/reporting/94b9
9bf2-959f-4266-816f-fac84adf5951/page/5LzcF
```

Figure 17: Click Both Check Boxes, Select Embed URL and COPY TO CLIPBOARD.

Part 3: Creating Mobile App With Embedded Dashboard

Step 3.1: Set Up Mobile Development Environment

Open Android Studio application and create a new project using Empty Views Activity template.

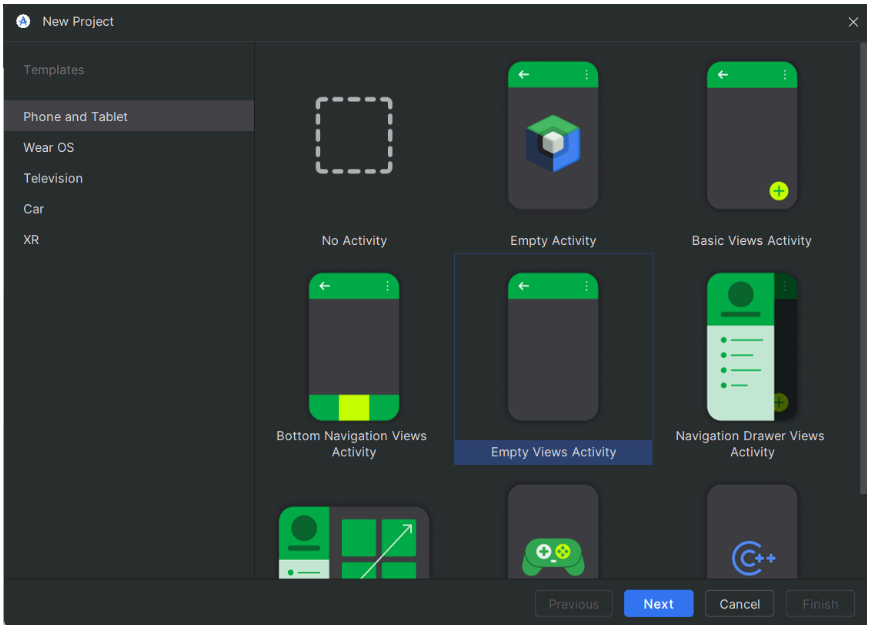
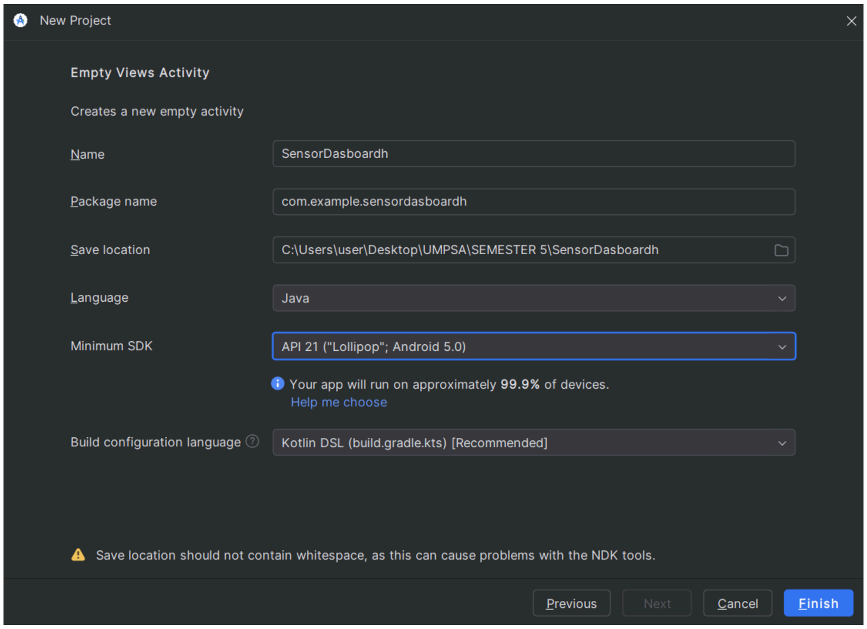


Figure 18: Select Empty Views Activity and Click Next.

Change the project name to SensorDashboard and use Android 5.0 Lollipop.



Empty Views Activity

Creates a new empty activity

Name: SensorDasboardh

Package name: com.example.sensordasboardh

Save location: C:\Users\user\Desktop\UMPSA\SEMESTER 5\SensorDasboardh

Language: Java

Minimum SDK: API 21 ("Lollipop"; Android 5.0)

Your app will run on approximately **99.9%** of devices.
Help me choose

Build configuration language: Kotlin DSL (build.gradle.kts) [Recommended]

Save location should not contain whitespace, as this can cause problems with the NDK tools.

Figure 19: New Empty Views Activity Project Named SensorDashboard Using Java Language With Android 5.0 at Minimum SDK.

Step 3.2: Configure the Webview in App

Replace the existing activity_main.xml file code with this code:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <WebView
        android:id="@+id/dashboardWebView"
        android:layout_width="0dp"
        android:layout_height="0dp"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent" />

    <ProgressBar
        android:id="@+id/progressBar"
        style="?android:attr/progressBarStyleLarge"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:visibility="gone"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Update the MainActivity.java file code with this code:

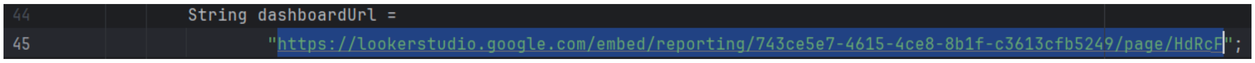
```
package com.example.sensordashboard;

import androidx.activity.OnBackPressedCallback;
import androidx.appcompat.app.AppCompatActivity;

import android.annotation.SuppressLint;
import android.os.Bundle;
import android.view.View;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.ProgressBar;

public class MainActivity extends AppCompatActivity {
    private WebView dashboardWebView;
    private ProgressBar progressBar;

    @SuppressLint("SetJavaScriptEnabled")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        dashboardWebView = findViewById(R.id.dashboardWebView);
        progressBar = findViewById(R.id.progressBar);

        WebSettings webSettings = dashboardWebView.getSettings();
        webSettings.setJavaScriptEnabled(true);
        webSettings.setDomStorageEnabled(true);
        webSettings.setLoadWithOverviewMode(true);
        webSettings.setUseWideViewPort(true);
        webSettings.setSupportZoom(true);
        webSettings.setBuiltInZoomControls(true);
        webSettings.setDisplayZoomControls(false);

        dashboardWebView.setWebViewClient(new WebViewClient() {
            @Override
            public void onPageFinished(WebView view, String url) {
                progressBar.setVisibility(View.GONE);
                super.onPageFinished(view, url);
            }
        });

        String dashboardUrl =

"https://lookerstudio.google.com/embed/reporting/94b99bf2-959f-4266
-816f-fac84adf5951/page/5LzcF";
        progressBar.setVisibility(View.VISIBLE);
        dashboardWebView.loadUrl(dashboardUrl);

        getOnBackPressedDispatcher().addCallback(this, new
OnBackPressedCallback(true) {
            @Override
            public void handleOnBackPressed() {
                if (dashboardWebView.canGoBack()) {
                    dashboardWebView.goBack();
                } else {
                    finish();
                }
            }
        });
    }
}
```

Update the AndroidManifest.xml file code to add internet permission with this code:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
    xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.sensordashboard">
    <uses-permission
        android:name="android.permission.INTERNET" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.SensorDashboard">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action
                    android:name="android.intent.action.MAIN" />
                <category
                    android:name="android.intent.category.LAUNCHER"
/>
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Step 3.3: Insert Looker Studio Dashboard Embed URL

Open the MainActivity.java code and replace the existing URL with the created Looker Studio dashboard at the line containing `dashboardURL`.

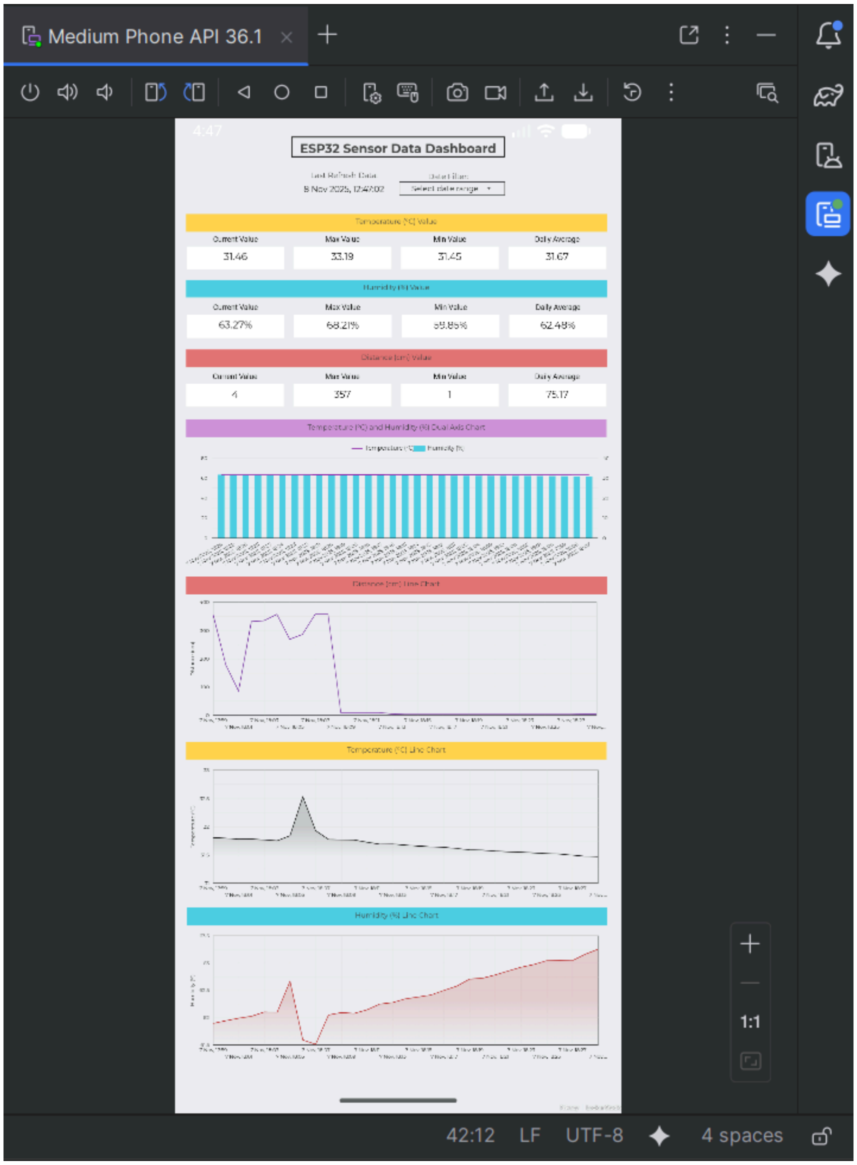


```
        String dashboardUrl =
                "https://lookerstudio.google.com/embed/reporting/743ce5e7-4615-4ce8-8b1f-c3613cfb5249/page/HdRcF";
```

Figure 20: Changed Existing URL With Created Dashboard Copied Embed URL.

Step 3.4: Build and Run App

Run the built app by clicking the Run button in Android Studio and wait for the app to install and launch.



ESP32 Sensor Data Dashboard

Last Refresh Date
8 Nov 2025, 12:47:02

Date Filter:
Select date range ▾

Temperature (°C) Value

Current Value	Max Value	Min Value	Daily Average
31.46	33.19	31.45	31.67

Humidity (%) Value

Current Value	Max Value	Min Value	Daily Average
63.27%	68.21%	59.85%	62.48%

Distance (cm) Value

Current Value	Max Value	Min Value	Daily Average
4	357	1	75.17

Figure 21: Created Looker Studio Dashboard Successfully Run in App Using Android Studio.

Base Code Modification

The original Ultrasonic code was modified by adding the BME280 sensor, adjusting the interval and updating the data sensor validation.

1. Added BME280 Sensor Support.
   - Libraries for BME280 Sensor.

```
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BME280.h>
```

   - BM280 Sensor Global Object.

```
Adafruit_BME280 bme;
```

   - BME280 Sensor Initialization Setup.

```
if (!bme.begin(0x76)) {
  Serial.println("Could not find BME280 sensor!");
  while (1);
}
```

   - BME280 Sensor Readings in Loop.

```
float temperature = bme.readTemperature();
float humidity = bme.readHumidity();
```

2. Fixed Ultrasonic Pin Usage.
   - In the original code, the pin for ultrasonic was unclear and the Ultrasonic library required two pins which are ECHO and TRIG. Then it was replaced with this code:

```
#define TRIG_PIN 13
#define ECHO_PIN 12
Ultrasonic ultrasonic(TRIG_PIN, ECHO_PIN);
```

3. Added Temperature and Humidity with Existing Distance to the JSON.
   - In the original code, only distance data sent to Google Sheets and with BME280 sensor being added, the temperature and humidity data are added to the existing code:

```
doc["temperature"] = temperature;
doc["humidity"] = humidity;
doc["distance"] = distance;
```

4. Added Sensor Data Validation.
   - Improved the original data validation code to prevent bad or incomplete readings from being sent to Google Sheets:

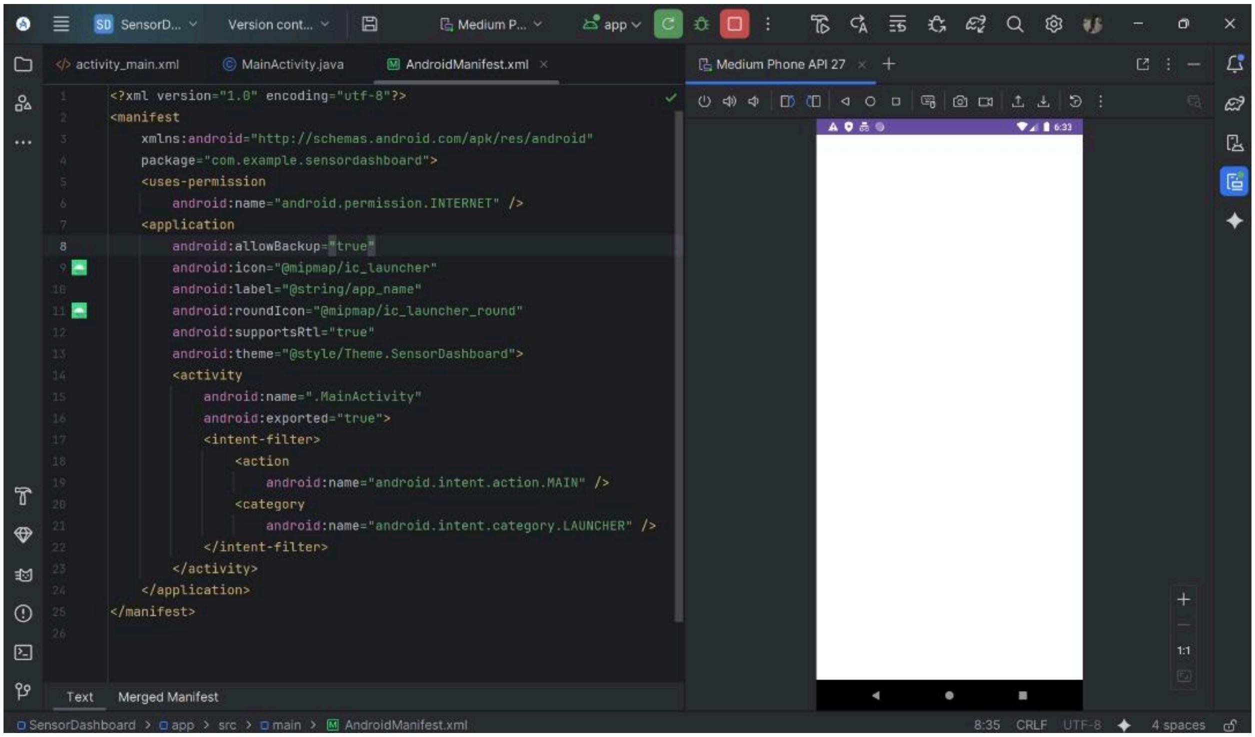
```
long distance = ultrasonic.read();
if (isnan(temperature) || isnan(humidity) || distance <= 0) {
  Serial.println("Failed to read from sensors!");
  return;
}
```

5. Updated Serial Output for Debugging.
   - Added a print statement to see all three sensor readings in the Serial Monitor.

```
Serial.print("Temperature: ");
Serial.print(temperature);
Serial.print("°C, Humidity: ");
Serial.print(humidity);
Serial.print("%, Distance: ");
Serial.print(distance);
Serial.println(" cm");
```

Challenges

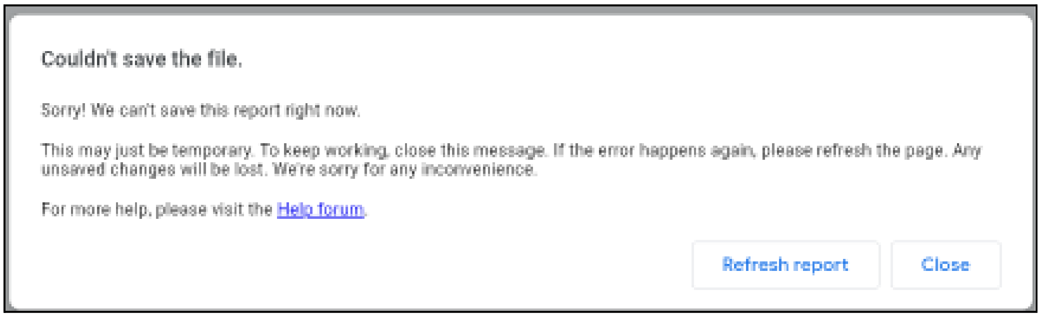
1. The screen appeared completely blank white even after refreshing while running the app in Android Studio



```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest
    xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.sensordashboard">
    <uses-permission
        android:name="android.permission.INTERNET" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.SensorDashboard">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action
                    android:name="android.intent.action.MAIN" />
                <category
                    android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Figure 22: App Appear Blank White.

2. An error message pop-up saying "Couldn't save the file. Sorry! We can't save this report right now." after the app was running.



Couldn't save the file.

Sorry! We can't save this report right now.

This may just be temporary. To keep working, close this message. If the error happens again, please refresh the page. Any unsaved changes will be lost. We're sorry for any inconvenience.

For more help, please visit the Help forum.

Figure 23: Error Message Pop-up in App.

Solutions

1. The solution to first challenge was to recreate back the Looker Studio Dashboard. Then reuse the same activity_main.xml, MainActivity.java and AndroidManifest.xml codes in the same project file.

2. The solution to the second challenge was refreshing the Looker Studio page and changing manage access from public to restricted and rechange it back to public access that allow people with link to edit and view.

Collected Data Analysis

The sensors data reading collected from the ESP32 includes temperature (°C), humidity (%), and distance (cm) that were then logged into Google Sheets and visualized through Looker Studio dashboards. The recording period on 7 November 2025 shows stable sensor behavior with small variations across all three parameters.

1. Temperature Data Analysis and Insight:

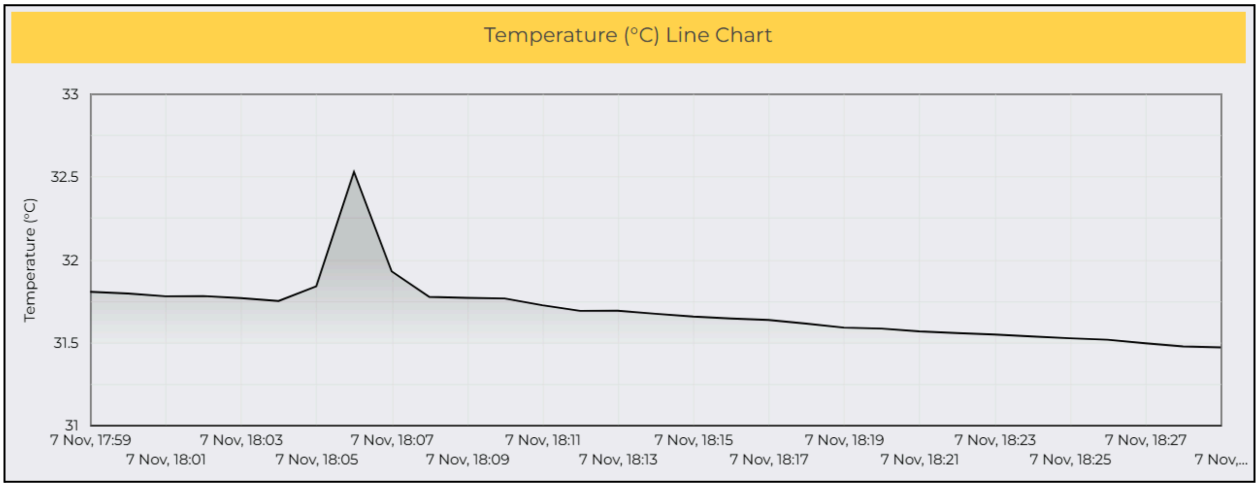


Figure 24: Temperature Data Line Chart.

In Figure 24, the temperature ranged from 31.0°C to 32.8°C throughout the period. A small peak was observed around 6.05 p.m., due to a momentary change in surrounding airflow or heat from nearby components. After that, the temperature gradually decreased and stabilized at around 31.3°C.

2. Humidity Data Analysis and Insight:

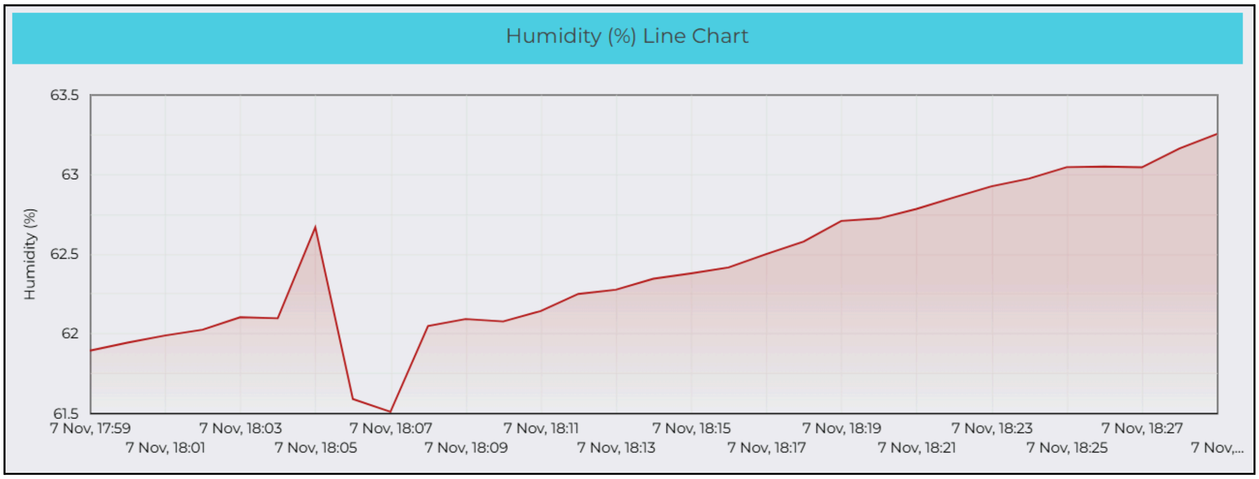


Figure 25: Humidity Data Line Chart.

In Figure 25, humidity readings were ranged between 61.5% and 63.5%. A minor drop corresponding to the brief temperature rise at 6.05 p.m., but after that the humidity steadily increased. This chart shows the expected inverse relationship between temperature and humidity which was temperature decreased, humidity increased slightly.

3.Distance Data Analysis and Insight:

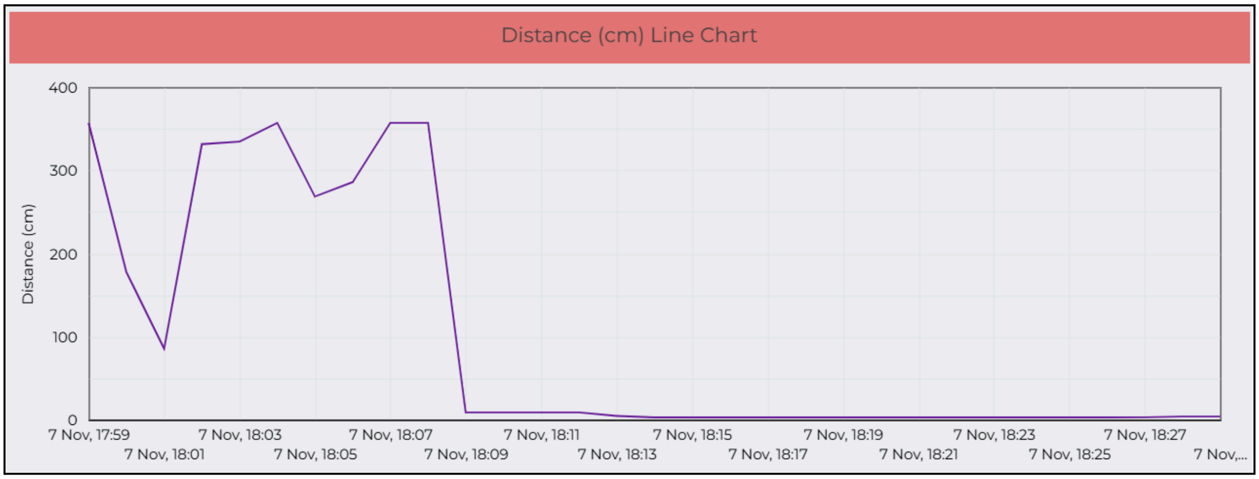


Figure 26: Distance Data Line Chart.

Readings were fluctuated between 100 cm to 350 cm in Figure 26 due to the object was far from the sensor detection range. But after 6.09 p.m., the readings dropped and stabilized near 0 cm, which indicate that the measured object was moved very close to the sensor almost as if the sensor field of view was blocked.

Screenshots of Working System

1. Hardware Setup.

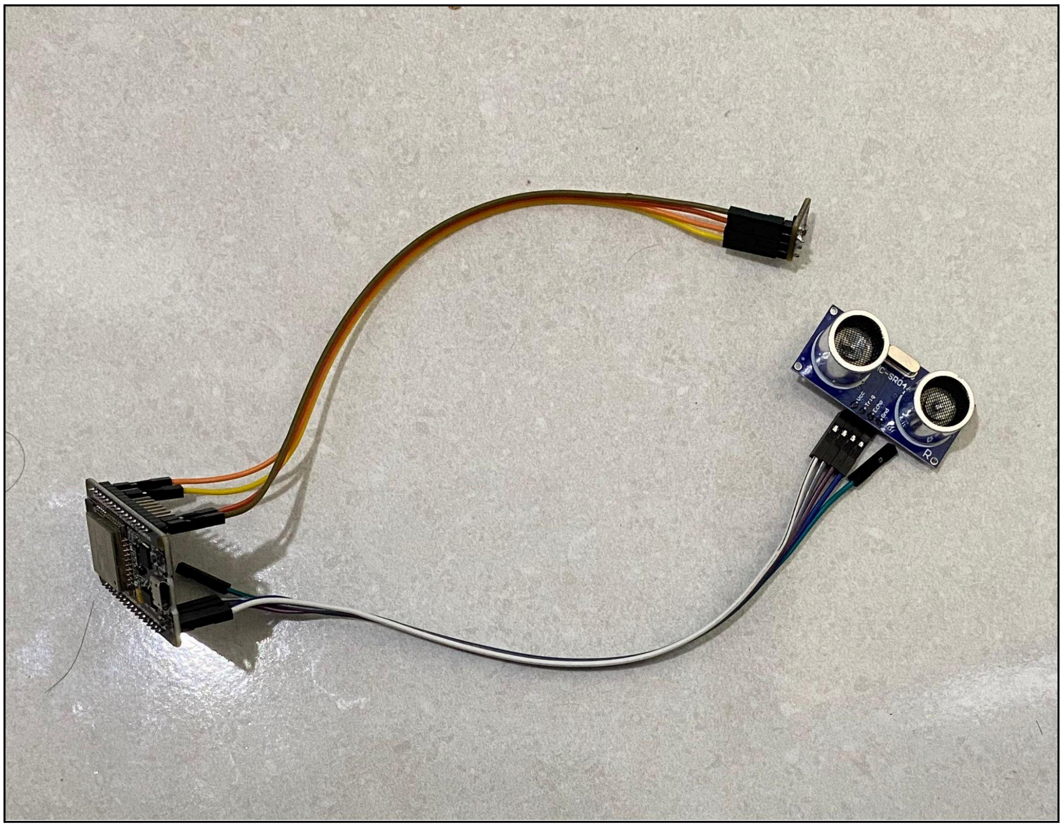
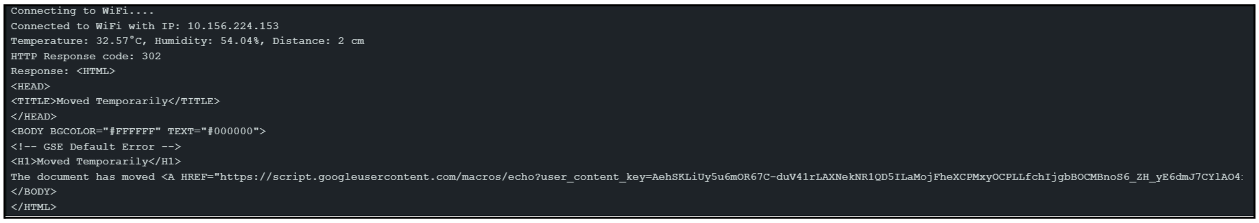


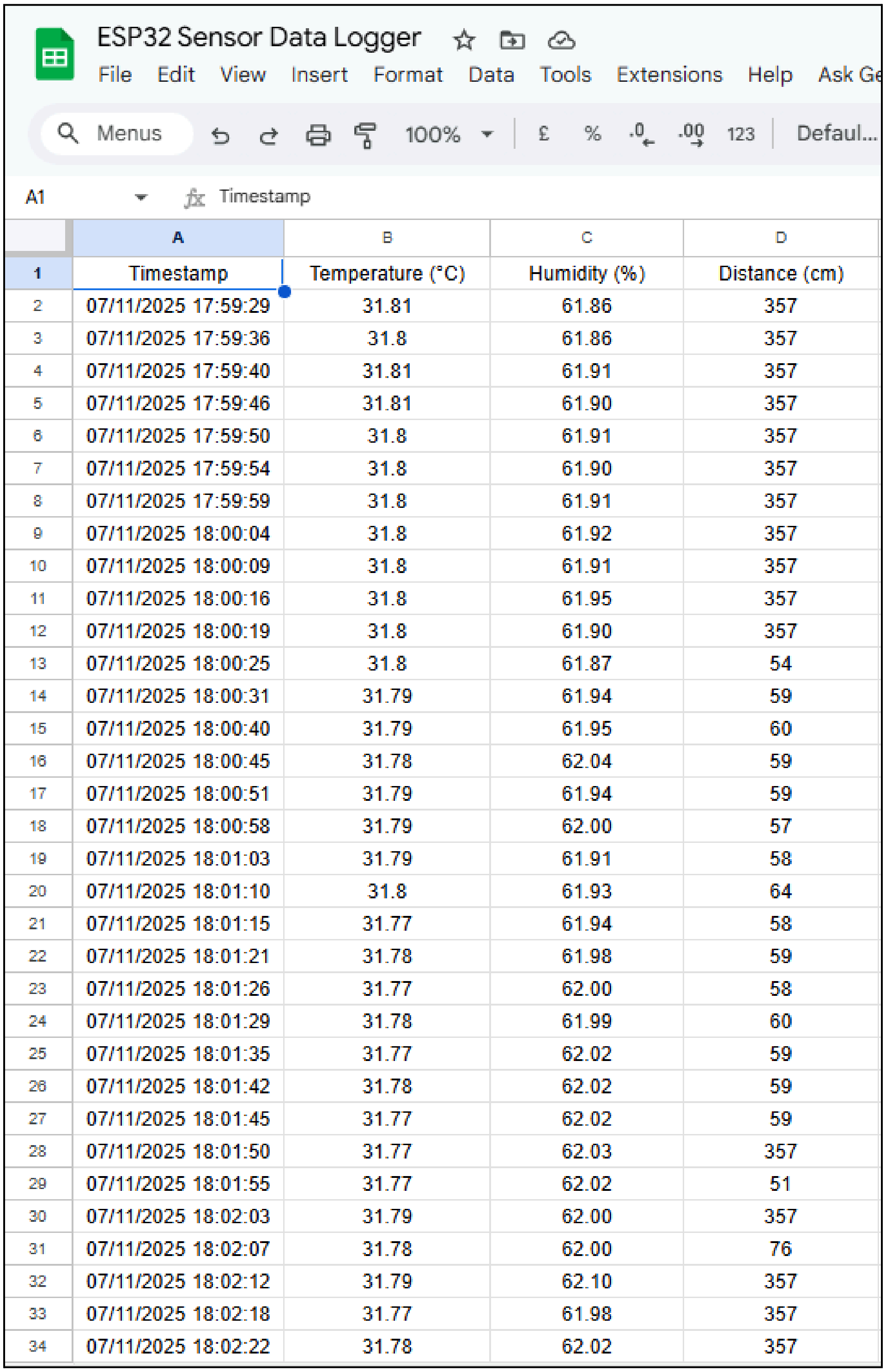
Figure 27: BME280 Temperature and Humidity sensor and HC-SR04 Ultrasonic sensor Hardware Connection to ESP32.



```
Connecting to WiFi....
Connected to WiFi with IP: 10.156.224.153
Temperature: 32.57°C, Humidity: 54.04%, Distance: 2 cm
HTTP Response code: 302
Response: <HTML>
<HEAD>
<TITLE>Moved Temporarily</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="#000000">
<!-- GSE Default Error -->
<H1>Moved Temporarily</H1>
The document has moved <A HREF="https://script.googleusercontent.com/macros/echo?user_content_key=AehSKLiUy5u6mOR67C-duV41rLAXNekNR1QD5ILaMojFheXCPMxyOCPLLfchIjgbBOCMBnoS6_ZH_yE6dmJ7CYlAO4:
</BODY>
</HTML>
```

Figure 28: ESP32 Connected to Wi-Fi, Received Temperature, Humidity and Distance Data And Successfully Send Data To Google Sheet.

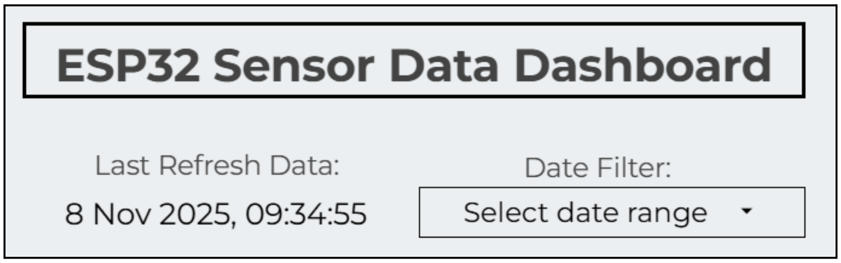
2. Google Sheet.



	Timestamp	Temperature (°C)	Humidity (%)	Distance (cm)
2	07/11/2025 17:59:29	31.81	61.86	357
3	07/11/2025 17:59:36	31.8	61.86	357
4	07/11/2025 17:59:40	31.81	61.91	357
5	07/11/2025 17:59:46	31.81	61.90	357
6	07/11/2025 17:59:50	31.8	61.91	357
7	07/11/2025 17:59:54	31.8	61.90	357
8	07/11/2025 17:59:59	31.8	61.91	357
9	07/11/2025 18:00:04	31.8	61.92	357
10	07/11/2025 18:00:09	31.8	61.91	357
11	07/11/2025 18:00:16	31.8	61.95	357
12	07/11/2025 18:00:19	31.8	61.90	357
13	07/11/2025 18:00:25	31.8	61.87	54
14	07/11/2025 18:00:31	31.79	61.94	59
15	07/11/2025 18:00:40	31.79	61.95	60
16	07/11/2025 18:00:45	31.78	62.04	59
17	07/11/2025 18:00:51	31.79	61.94	59
18	07/11/2025 18:00:58	31.79	62.00	57
19	07/11/2025 18:01:03	31.79	61.91	58
20	07/11/2025 18:01:10	31.8	61.93	64
21	07/11/2025 18:01:15	31.77	61.94	58
22	07/11/2025 18:01:21	31.78	61.98	59
23	07/11/2025 18:01:26	31.77	62.00	58
24	07/11/2025 18:01:29	31.78	61.99	60
25	07/11/2025 18:01:35	31.77	62.02	59
26	07/11/2025 18:01:42	31.78	62.02	59
27	07/11/2025 18:01:45	31.77	62.02	59
28	07/11/2025 18:01:50	31.77	62.03	357
29	07/11/2025 18:01:55	31.77	62.02	51
30	07/11/2025 18:02:03	31.79	62.00	357
31	07/11/2025 18:02:07	31.78	62.00	76
32	07/11/2025 18:02:12	31.79	62.10	357
33	07/11/2025 18:02:18	31.77	61.98	357
34	07/11/2025 18:02:22	31.78	62.02	357

Figure 29: Acquired Timestamp, Temperature, Humidity and Distance Data Sent By ESP32 Microcontroller.

3. Looker Studio Dashboard.

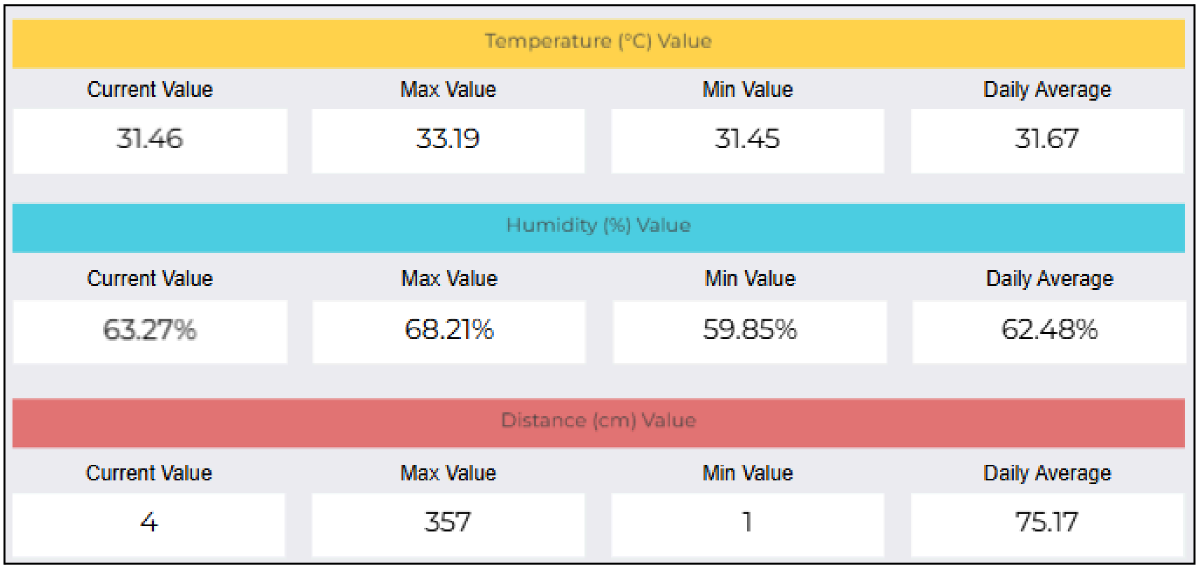


ESP32 Sensor Data Dashboard

Last Refresh Data:
8 Nov 2025, 09:34:55

Date Filter:
Select date range ▾

Figure 30: Text Element Dashboard Title, Table Element Last Refresh and Date Range Control Element.



Temperature (°C) Value

Current Value	Max Value	Min Value	Daily Average
31.46	33.19	31.45	31.67

Humidity (%) Value

Current Value	Max Value	Min Value	Daily Average
63.27%	68.21%	59.85%	62.48%

Distance (cm) Value

Current Value	Max Value	Min Value	Daily Average
4	357	1	75.17

Figure 31: Temperature, Humidity and Distance Data Value with 4 Different Aggregation Setup Alongside Text Element Separating 3 Data Source Rows.

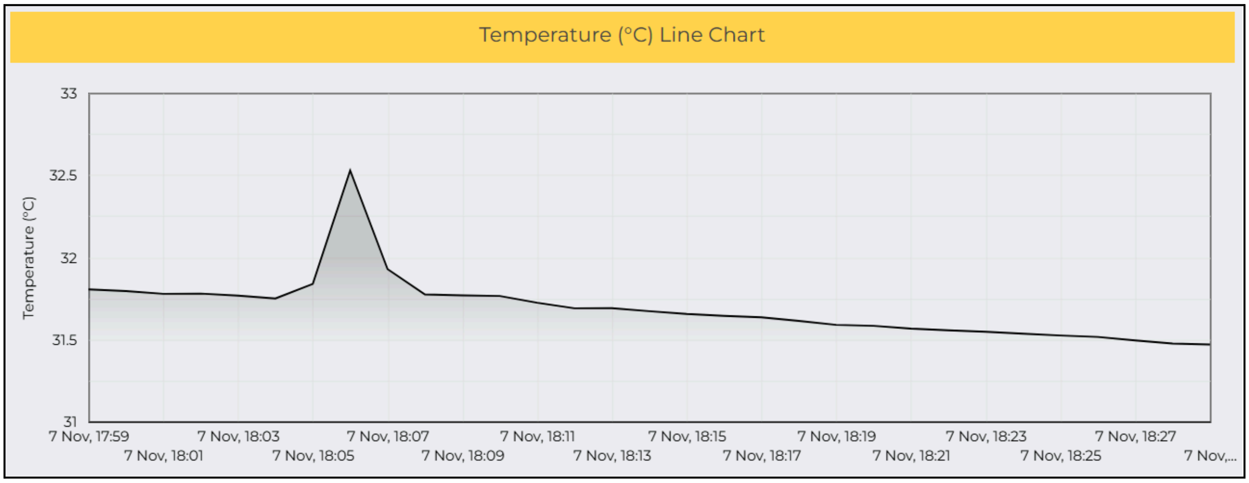


Figure 32: Temperature Data with Average Aggregation Using Line Chart.

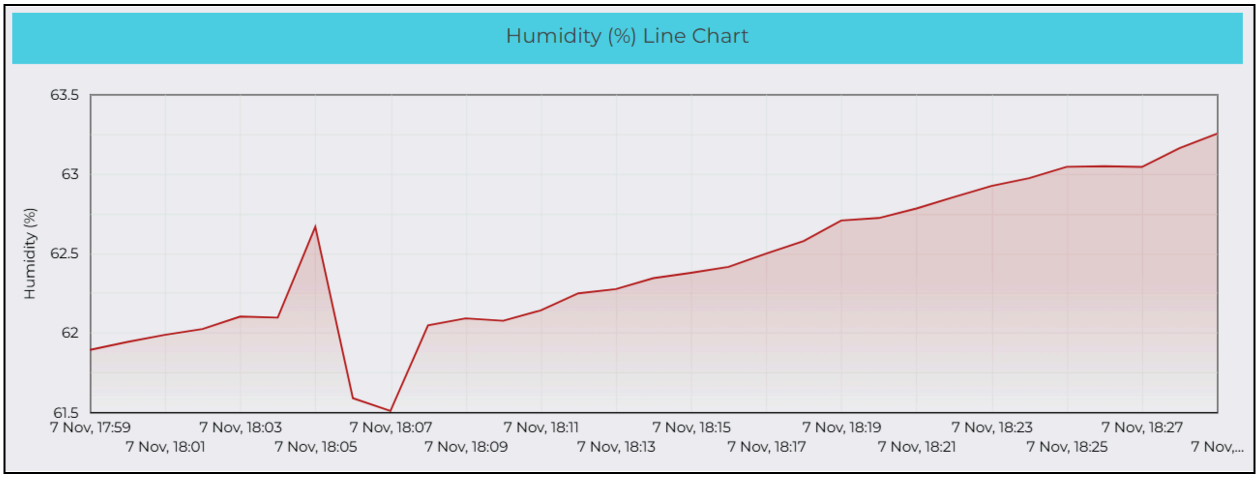


Figure 33: Humidity Data with Average Aggregation Using Line Chart.

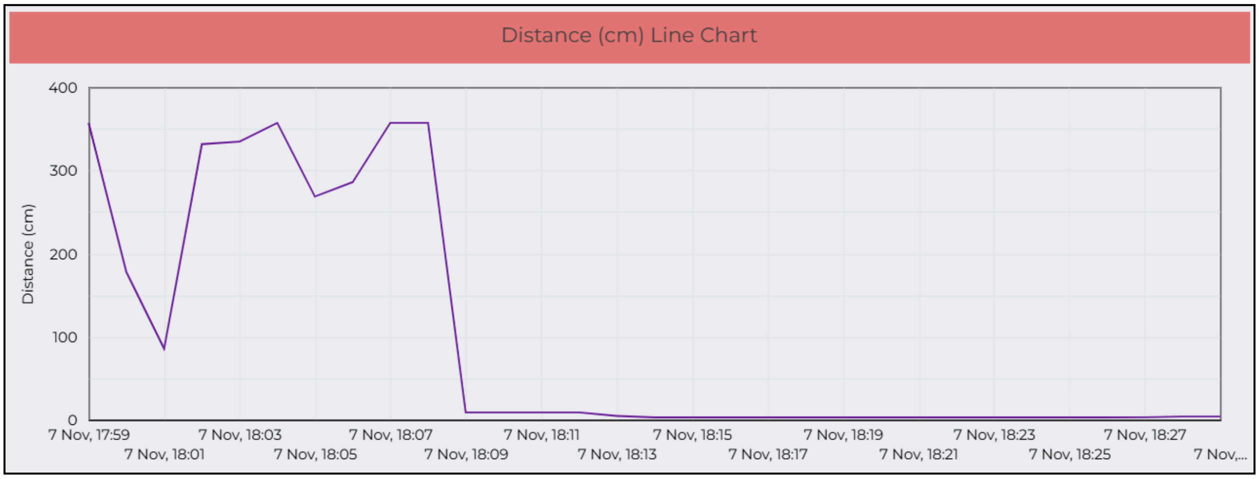
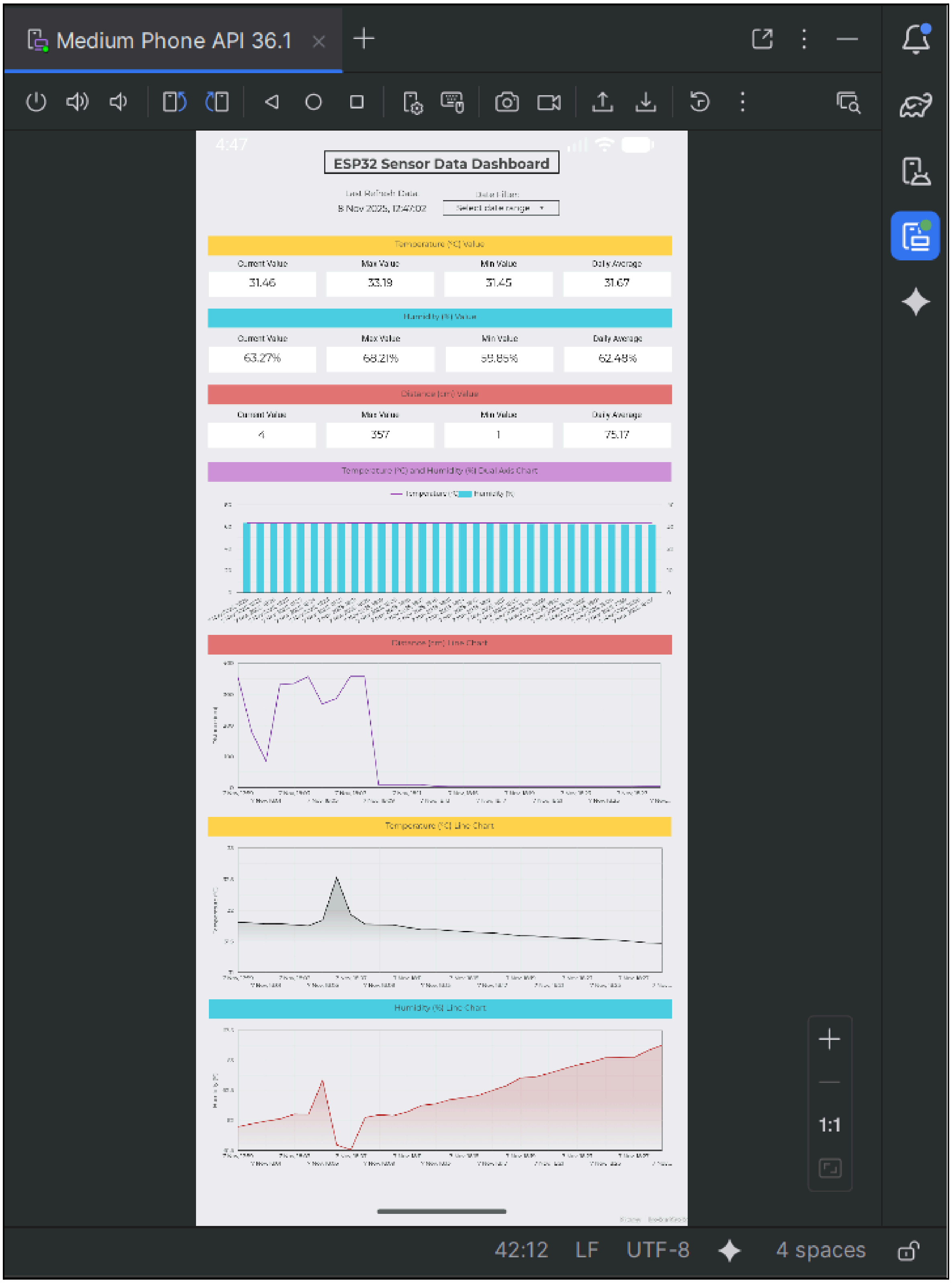


Figure 34: Distance Data with Average Aggregation Data Using Line Chart.

4. Android Studio Dashboard App.



ESP32 Sensor Data Dashboard

Last Refresh Date: 8 Nov 2025, 12:47:02

Data Filter: Select date range ▾

Temperature (°C) Value

Current Value	Max Value	Min Value	Daily Average
31.46	33.19	31.45	31.67

Humidity (%) Value

Current Value	Max Value	Min Value	Daily Average
63.27%	68.21%	59.85%	62.48%

Distance (cm) Value

Current Value	Max Value	Min Value	Daily Average
4	357	1	75.17

Temperature (°C) and Humidity (%) Dual Axis Chart

Distance (cm) Line Chart

Temperature (°C) Line Chart

Humidity (%) Line Chart

Figure 35: ESP32 Sensor Data Logger Dashboard App Running in Android Studio.

GitHub Repositories Link

https://github.com/Kepengan/TSO2_LAB3