FACULTY OF ELECTRICAL AND ELECTRONIC ENGINEERING TECHNOLOGY

BVI 3114 TECHNOLOGY SYSTEM OPTIMIZATION II

SEMESTER 5 2025/2026

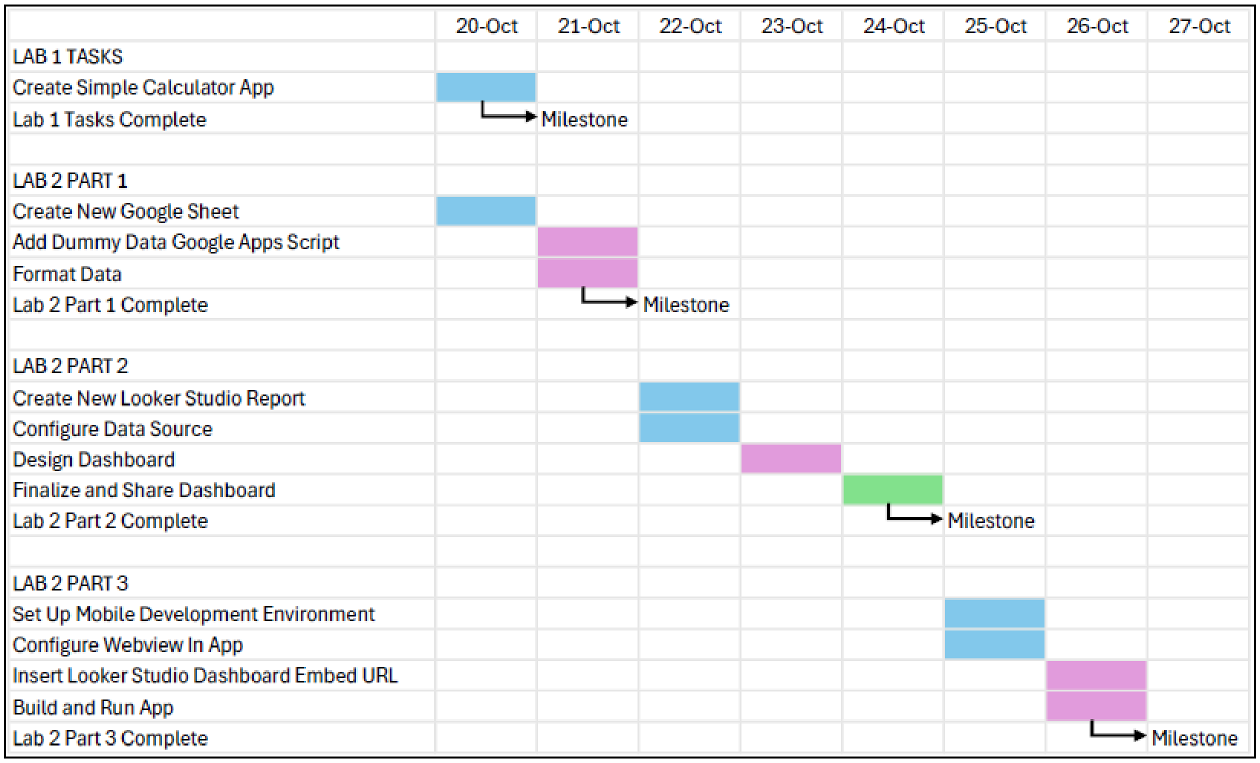
Lecturer:
Dr. Mohd Zamri Bin Ibrahim

Report:
Temperature and Humidity Dashboard with Looker Studio and integration with Android Studio

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

Table of Content

Timeline and Milestones



	20-Oct	21-Oct	22-Oct	23-Oct	24-Oct	25-Oct	26-Oct	27-Oct
LAB 1 TASKS								
Create Simple Calculator App	🟦							
Lab 1 Tasks Complete	↳	Milestone						
LAB 2 PART 1								
Create New Google Sheet	🟦							
Add Dummy Data Google Apps Script		🟪						
Format Data		🟪						
Lab 2 Part 1 Complete		↳	Milestone					
LAB 2 PART 2								
Create New Looker Studio Report			🟦					
Configure Data Source			🟦					
Design Dashboard				🟪				
Finalize and Share Dashboard					🟩			
Lab 2 Part 2 Complete					↳	Milestone		
LAB 2 PART 3								
Set Up Mobile Development Environment						🟦		
Configure Webview In App						🟦		
Insert Looker Studio Dashboard Embed URL							🟪	
Build and Run App							🟪	
Lab 2 Part 3 Complete							↳	Milestone

Figure 1: Timeline and Milestones.

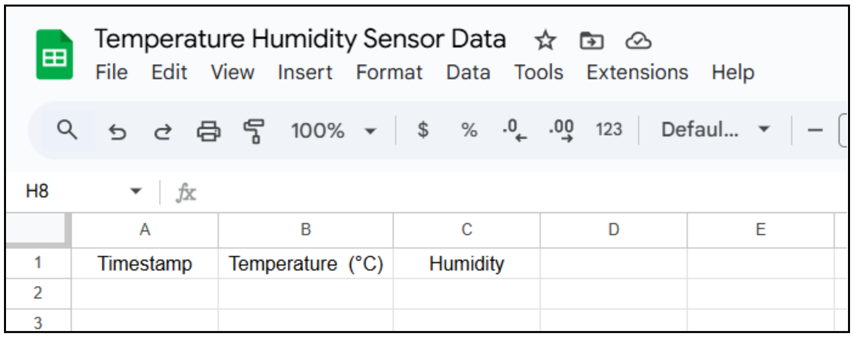
Implementation Process

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 "Temperature Humidity Sensor Data". Then create the three column headers in row 1 which is:

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


Temperature Humidity Sensor Data
File Edit View Insert Format Data Tools Extensions Help
100%
Defaul...
H8

	A	B	C	D	E
1	Timestamp	Temperature (°C)	Humidity		
2					
3					

Figure 2: Created a Sheet with Timestamp, Temperature (°C) and Humidity (%) columns.

Step 1.2: Add Google Apps Script to Generate Dummy Data

Open the Google Sheet Apps Script in the Extensions menu above. Clear out the existing code and insert the given following Google Apps Script code.

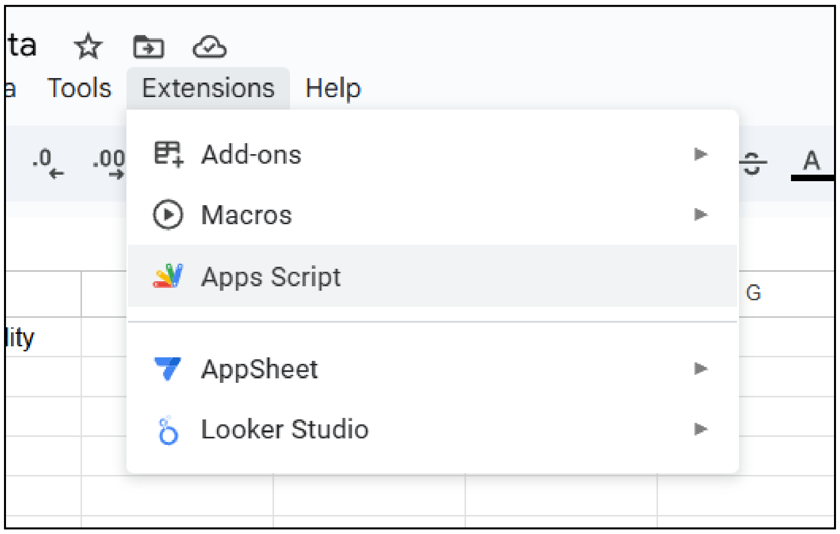
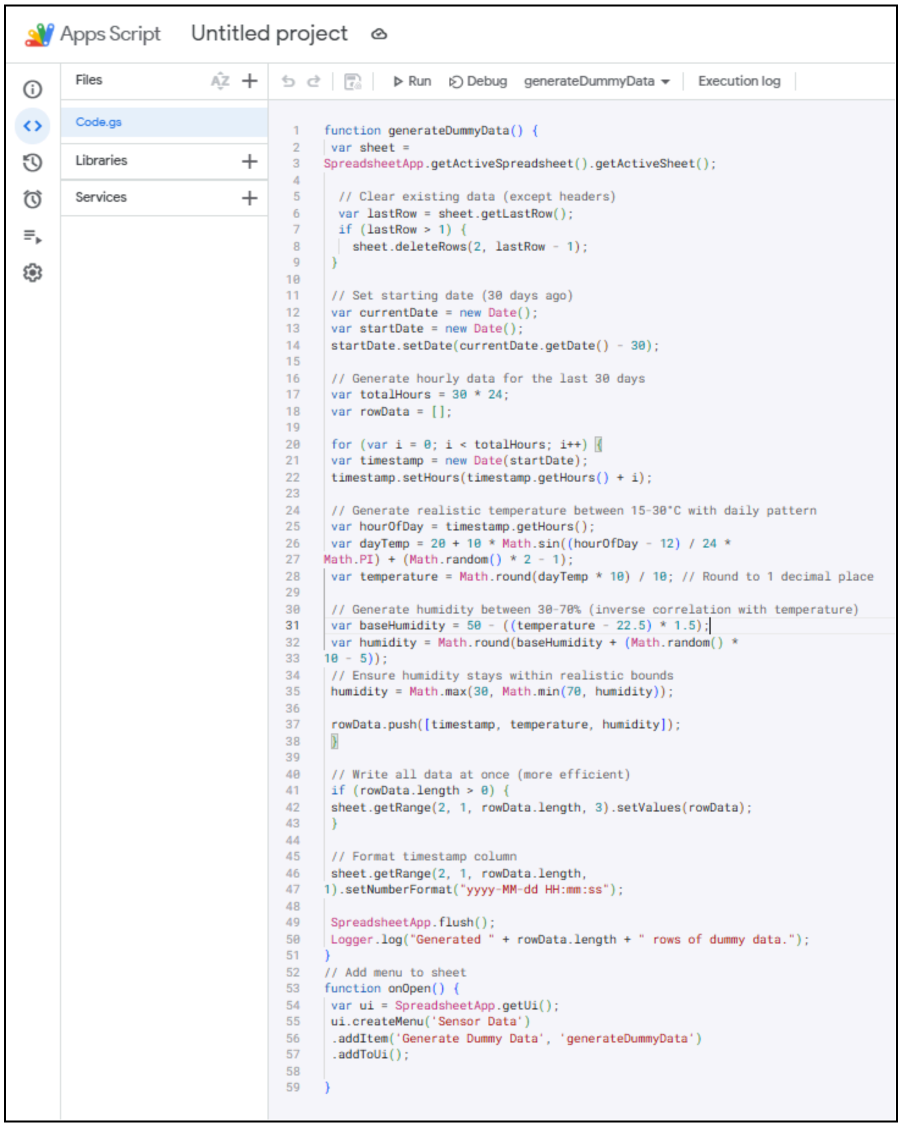


Figure 3: Apps Script Tool Option in Extensions Menu.



```javascript
function generateDummyData() {
 var sheet =
SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();

  // Clear existing data (except headers)
  var lastRow = sheet.getLastRow();
  if (lastRow > 1) {
    sheet.deleteRows(2, lastRow - 1);
 }

 // Set starting date (30 days ago)
 var currentDate = new Date();
 var startDate = new Date();
 startDate.setDate(currentDate.getDate() - 30);

 // Generate hourly data for the last 30 days
 var totalHours = 30 * 24;
 var rowData = [];

 for (var i = 0; i < totalHours; i++) {
 var timestamp = new Date(startDate);
 timestamp.setHours(timestamp.getHours() + i);

 // Generate realistic temperature between 15-30°C with daily pattern
 var hourOfDay = timestamp.getHours();
 var dayTemp = 20 + 10 * Math.sin((hourOfDay - 12) / 24 *
Math.PI) + (Math.random() * 2 - 1);
 var temperature = Math.round(dayTemp * 10) / 10; // Round to 1 decimal place

 // Generate humidity between 30-70% (inverse correlation with temperature)
 var baseHumidity = 50 - ((temperature - 22.5) * 1.5);
 var humidity = Math.round(baseHumidity + (Math.random() *
10 - 5));
 // Ensure humidity stays within realistic bounds
 humidity = Math.max(30, Math.min(70, humidity));

 rowData.push([timestamp, temperature, humidity]);
 }

 // Write all data at once (more efficient)
 if (rowData.length > 0) {
 sheet.getRange(2, 1, rowData.length, 3).setValues(rowData);
 }

 // Format timestamp column
 sheet.getRange(2, 1, rowData.length,
1).setNumberFormat("yyyy-MM-dd HH:mm:ss");

 SpreadsheetApp.flush();
 Logger.log("Generated " + rowData.length + " rows of dummy data.");
}
// Add menu to sheet
function onOpen() {
 var ui = SpreadsheetApp.getUi();
 ui.createMenu('Sensor Data')
 .addItem('Generate Dummy Data', 'generateDummyData')
 .addToUi();

}
```

Figure 4: Given Code Inserted in Google Apps Script.

After saving the inserted code, name the Apps Script title to Temperature Humidity Data Generator.

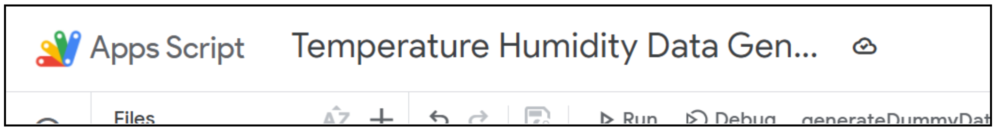


Figure 5: Apps Script Titled Temperature Humidity Data Generator.

Return to the Google Sheet page, refresh the page and a new menu item called Sensor Data will appear.

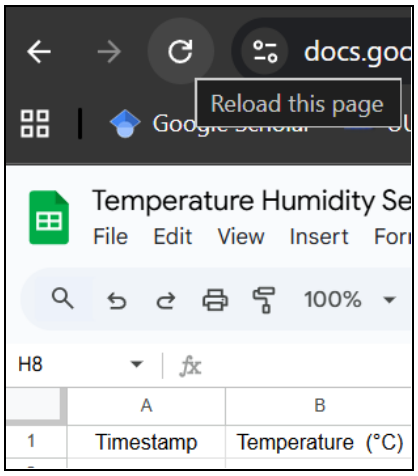


Figure 6: Refreshing the Google Sheets Page.

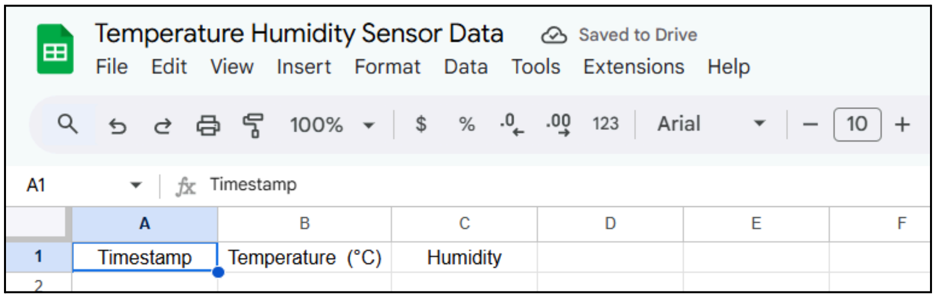


Figure 7: Before Refresh Google Sheets page.

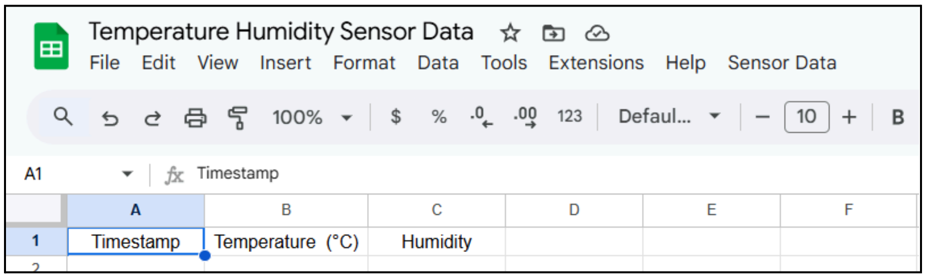


Figure 8: After Refresh Google Sheets page.

Click the Sensor Data menu and select Generate Dummy Data in order to fill the sheet with dummy sensor data. Click OK when prompted Authorization Required and allow the script to run. After the script completes, it will fill one row of dummy sensor data hourly for 30 days.

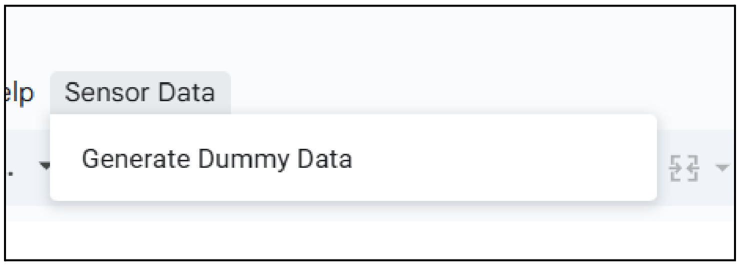
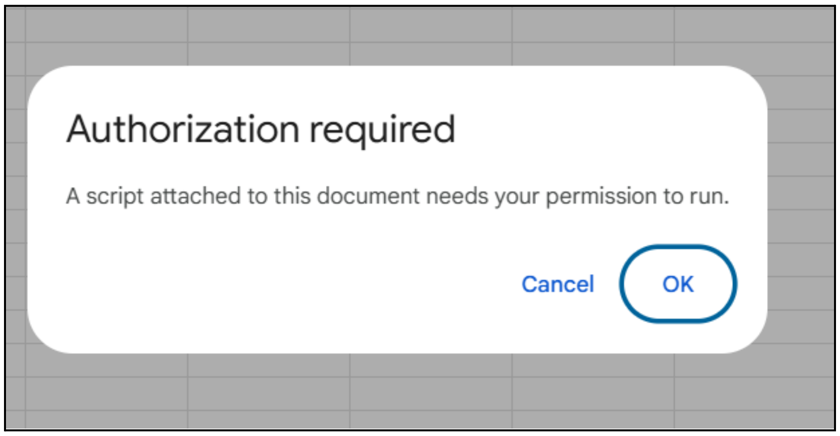
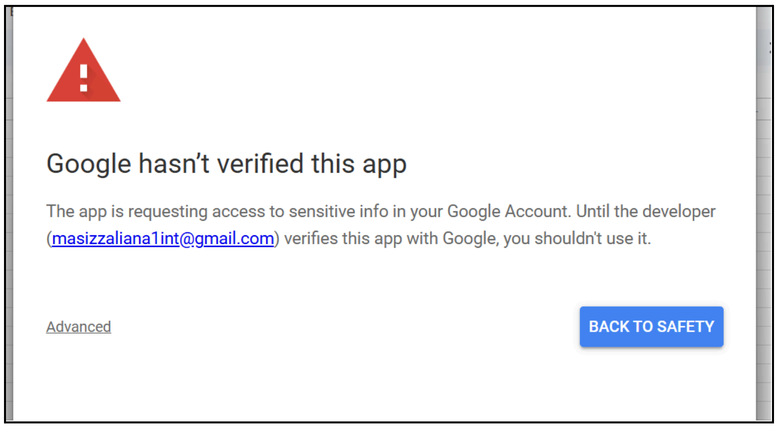


Figure 9: Generate a Dummy Data Tool.


Authorization required

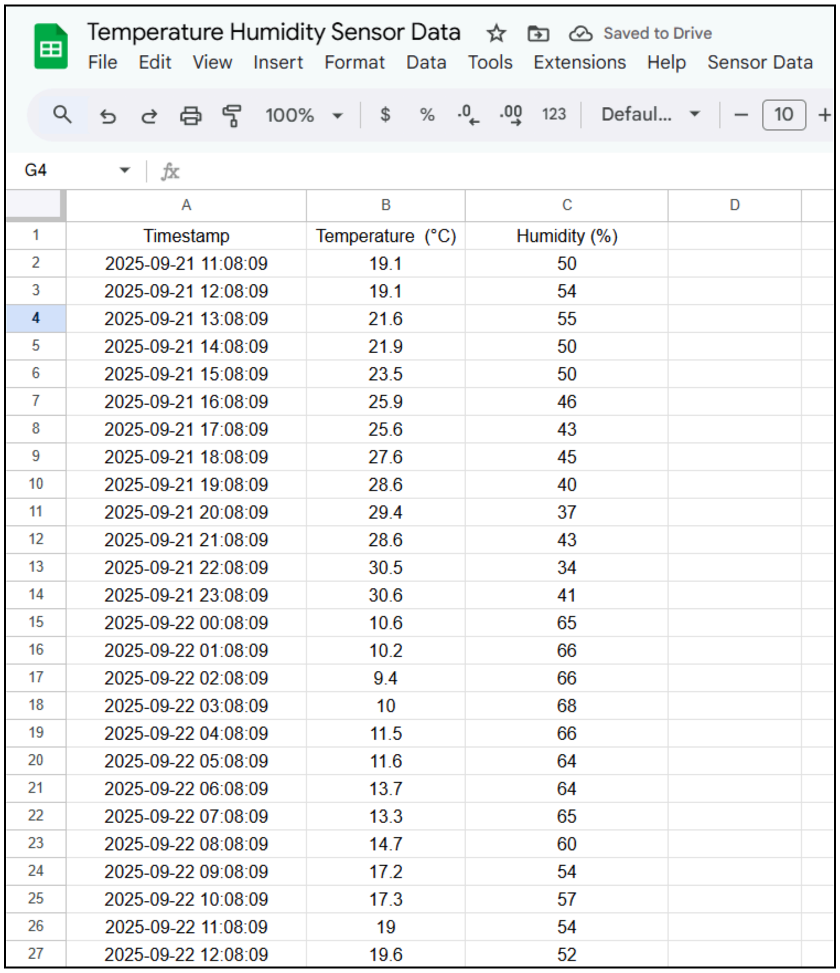
A script attached to this document needs your permission to run.

Figure 10: Required Authorization Prompted.


Google hasn't verified this app

The app is requesting access to sensitive info in your Google Account. Until the developer (masizzaliana1int@gmail.com) verifies this app with Google, you shouldn't use it.

Figure 11: Allowing the Script to Run.

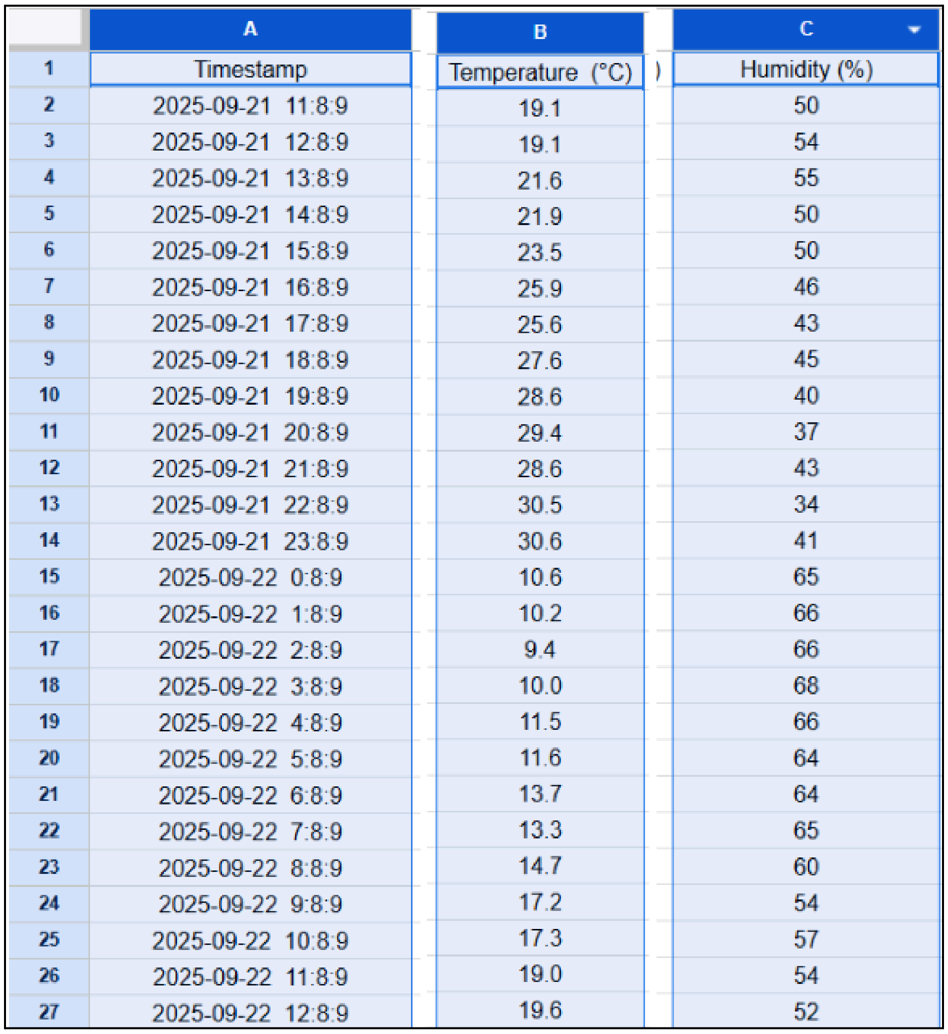

Temperature Humidity Sensor Data
Saved to Drive
File Edit View Insert Format Data Tools Extensions Help Sensor Data
100%
Defaul...
10
G4

	A	B	C	D
1	Timestamp	Temperature (°C)	Humidity (%)	
2	2025-09-21 11:08:09	19.1	50	
3	2025-09-21 12:08:09	19.1	54	
4	2025-09-21 13:08:09	21.6	55	
5	2025-09-21 14:08:09	21.9	50	
6	2025-09-21 15:08:09	23.5	50	
7	2025-09-21 16:08:09	25.9	46	
8	2025-09-21 17:08:09	25.6	43	
9	2025-09-21 18:08:09	27.6	45	
10	2025-09-21 19:08:09	28.6	40	
11	2025-09-21 20:08:09	29.4	37	
12	2025-09-21 21:08:09	28.6	43	
13	2025-09-21 22:08:09	30.5	34	
14	2025-09-21 23:08:09	30.6	41	
15	2025-09-22 00:08:09	10.6	65	
16	2025-09-22 01:08:09	10.2	66	
17	2025-09-22 02:08:09	9.4	66	
18	2025-09-22 03:08:09	10	68	
19	2025-09-22 04:08:09	11.5	66	
20	2025-09-22 05:08:09	11.6	64	
21	2025-09-22 06:08:09	13.7	64	
22	2025-09-22 07:08:09	13.3	65	
23	2025-09-22 08:08:09	14.7	60	
24	2025-09-22 09:08:09	17.2	54	
25	2025-09-22 10:08:09	17.3	57	
26	2025-09-22 11:08:09	19	54	
27	2025-09-22 12:08:09	19.6	52	

Figure 12: Timestamp, Temperature and Humidity Dummy Data Generated for 720 Rows.

Step 1.3: Format Data
Select all three 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.



	A	B	C
1	Timestamp	Temperature (°C)	Humidity (%)
2	2025-09-21 11:8:9	19.1	50
3	2025-09-21 12:8:9	19.1	54
4	2025-09-21 13:8:9	21.6	55
5	2025-09-21 14:8:9	21.9	50
6	2025-09-21 15:8:9	23.5	50
7	2025-09-21 16:8:9	25.9	46
8	2025-09-21 17:8:9	25.6	43
9	2025-09-21 18:8:9	27.6	45
10	2025-09-21 19:8:9	28.6	40
11	2025-09-21 20:8:9	29.4	37
12	2025-09-21 21:8:9	28.6	43
13	2025-09-21 22:8:9	30.5	34
14	2025-09-21 23:8:9	30.6	41
15	2025-09-22 0:8:9	10.6	65
16	2025-09-22 1:8:9	10.2	66
17	2025-09-22 2:8:9	9.4	66
18	2025-09-22 3:8:9	10.0	68
19	2025-09-22 4:8:9	11.5	66
20	2025-09-22 5:8:9	11.6	64
21	2025-09-22 6:8:9	13.7	64
22	2025-09-22 7:8:9	13.3	65
23	2025-09-22 8:8:9	14.7	60
24	2025-09-22 9:8:9	17.2	54
25	2025-09-22 10:8:9	17.3	57
26	2025-09-22 11:8:9	19.0	54
27	2025-09-22 12:8:9	19.6	52

Figure 13: Timestamp Column Data Formatted to Date Time, Temperature Column Data Formatted to 1 Decimal Place Number and Humidity 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.

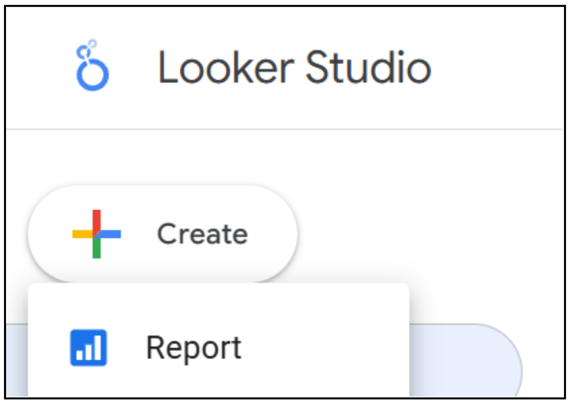
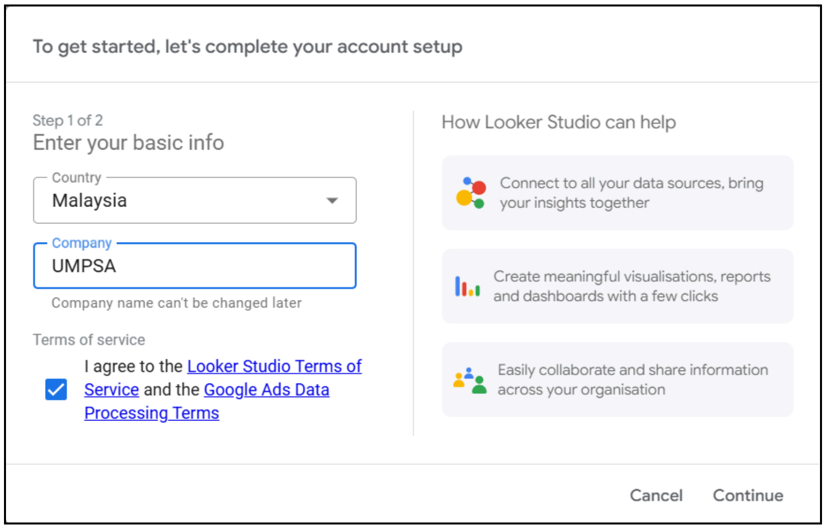


Figure 14: Create a New Report by Clicking the Create Button.



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

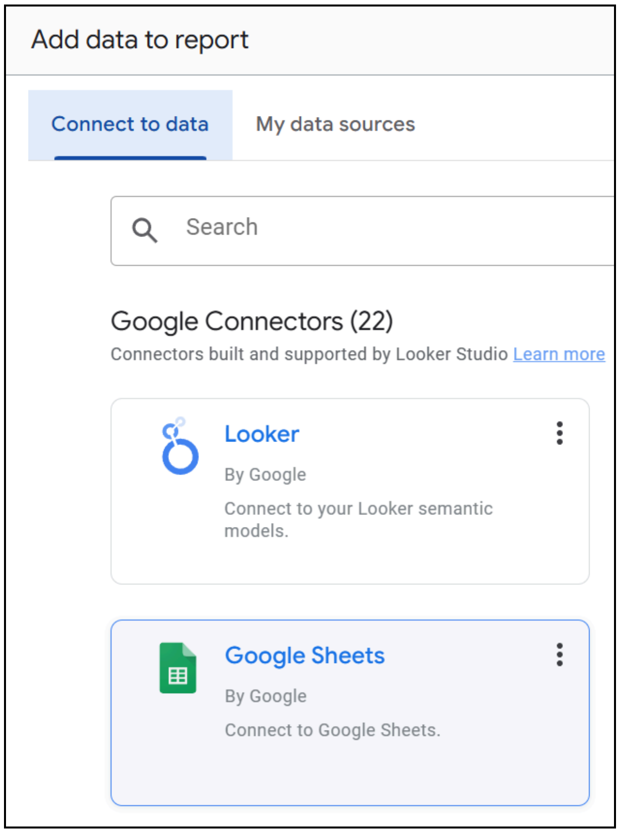
How Looker Studio can help

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 15: Enter the Basic Info Before Continuing.


Add data to report

Connect to data | My data sources

Search

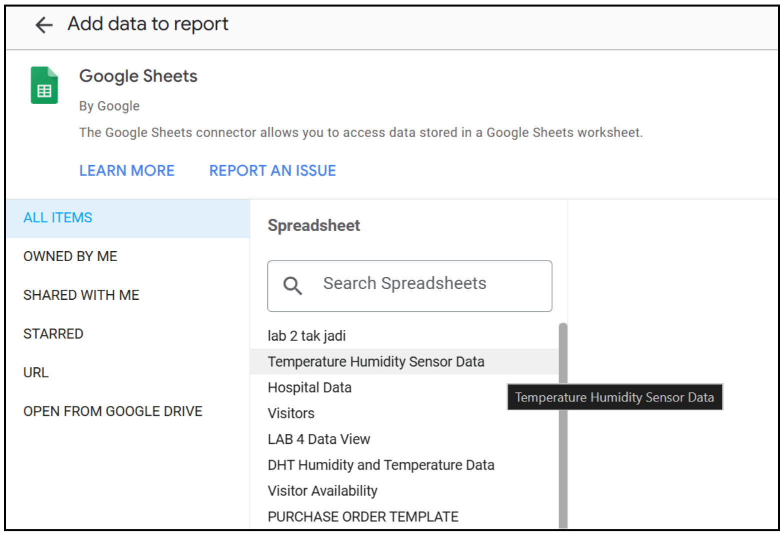
Google Connectors (22)

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 16: Select Google Sheets to Connect and Add Data to Report.


Add data to report

Google Sheets
By Google
The Google Sheets connector allows you to access data stored in a Google Sheets worksheet.

LEARN MORE   REPORT AN ISSUE

ALL ITEMS
OWNED BY ME
SHARED WITH ME
STARRED
URL
OPEN FROM GOOGLE DRIVE

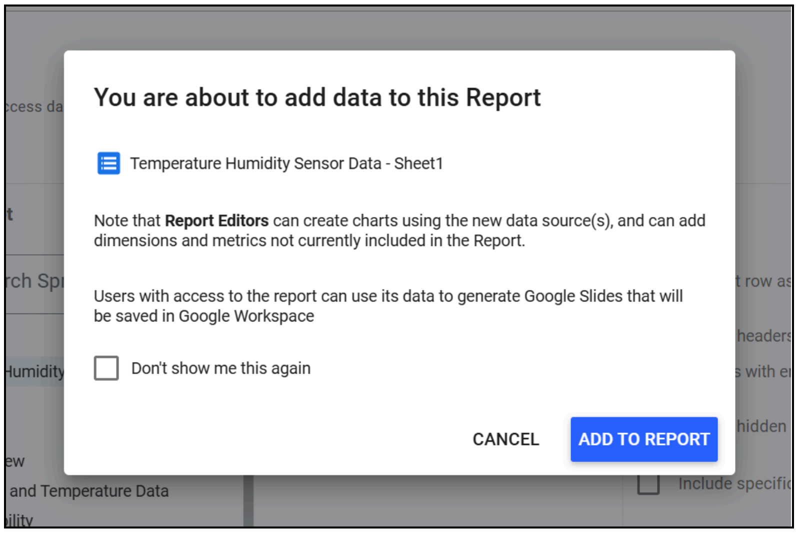
Spreadsheet

Search Spreadsheets

lab 2 tak jadi
Temperature Humidity Sensor Data
Hospital Data
Visitors
LAB 4 Data View
DHT Humidity and Temperature Data
Visitor Availability
PURCHASE ORDER TEMPLATE

Temperature Humidity Sensor Data

Figure 17: Select the Temperature Humidity Sensor Data Spreadsheet.


You are about to add data to this Report

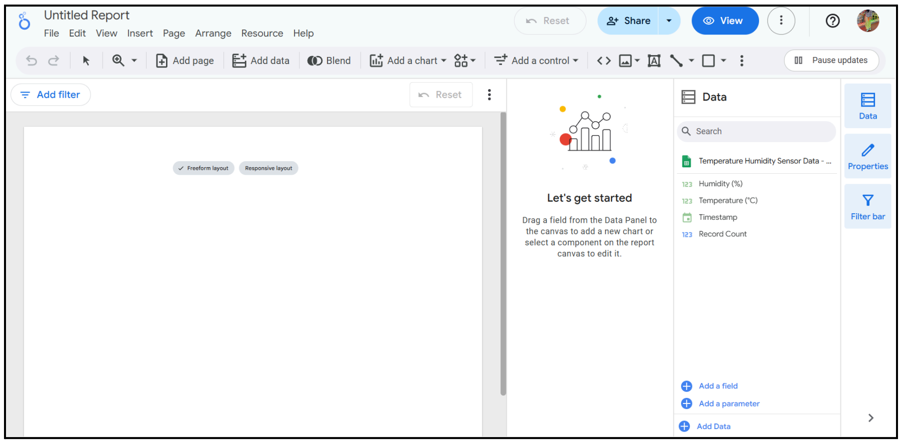
Temperature Humidity Sensor Data - Sheet1

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 18: Confirm the Selected Data to Be Added to Report.



Let's get started

Drag a field from the Data Panel to the canvas to add a new chart or select a component on the report canvas to edit it.

Figure 19: 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 and Humidity should be a numeric metric.

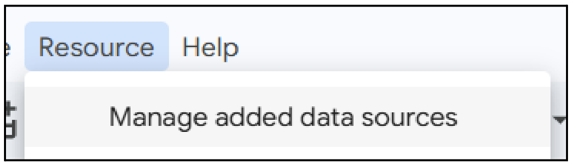
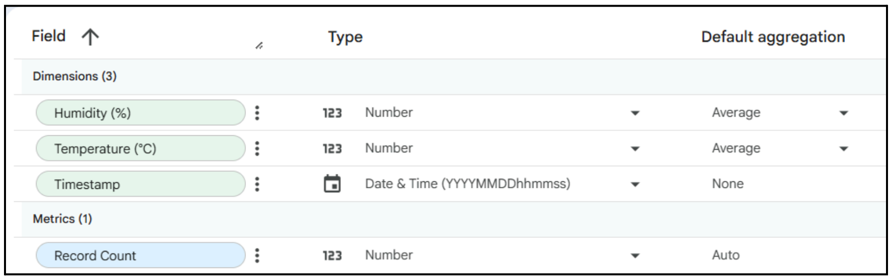


Figure 20: Select Manage added data sources to View Data Types.



Field	Type	Default aggregation
Dimensions (3)		
Humidity (%)	Number	Average
Temperature (°C)	Number	Average
Timestamp	Date & Time (YYYYMMDDhhmmss)	None
Metrics (1)		
Record Count	Number	Auto

Figure 21: Humidity and Temperature are Number Data Type With Average Default Aggregation and Timestamp are Date & Time 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 and separate metrics scorecard for average and current values of temperature and humidity.

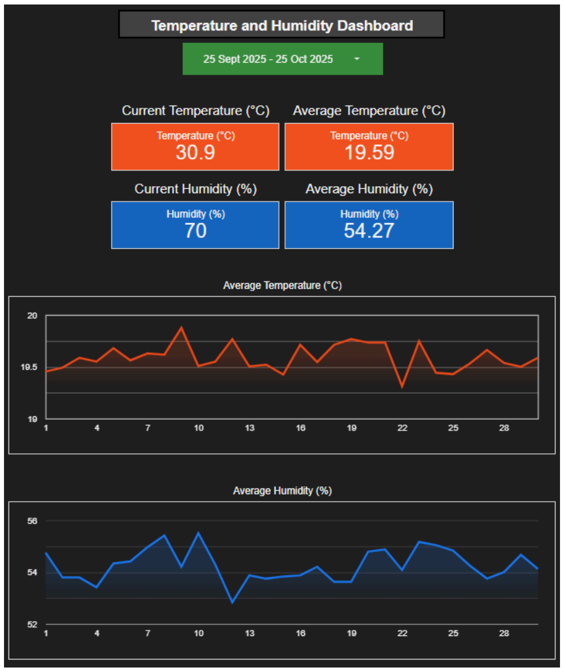


Figure 22: Dashboard Design Included With Title, Data Range Control, Time Series Chart and Separate Metrics Scorecard of Average and Current Values of Temperature and Humidity Data.

Step 2.4: Finalize and Share Dashboard

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

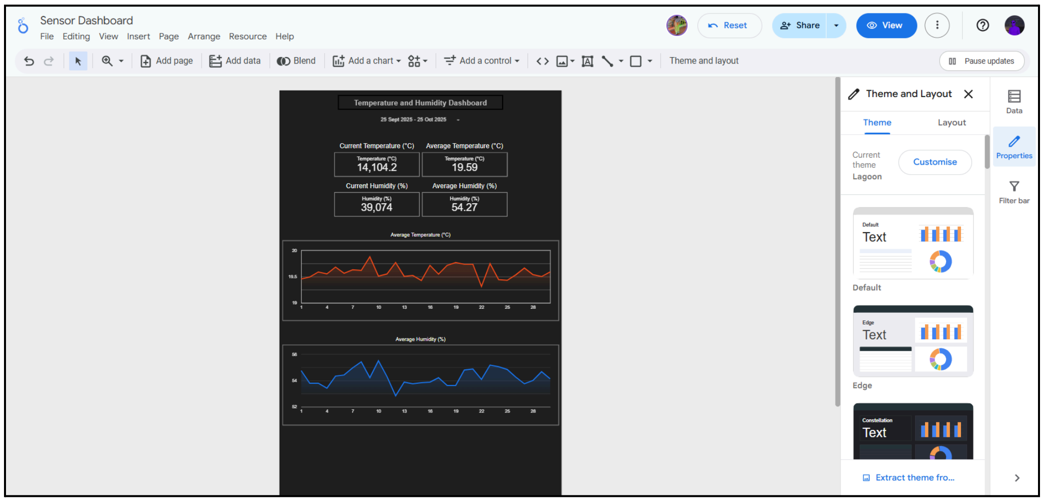


Figure 23: Looker Studio Edit Mode.

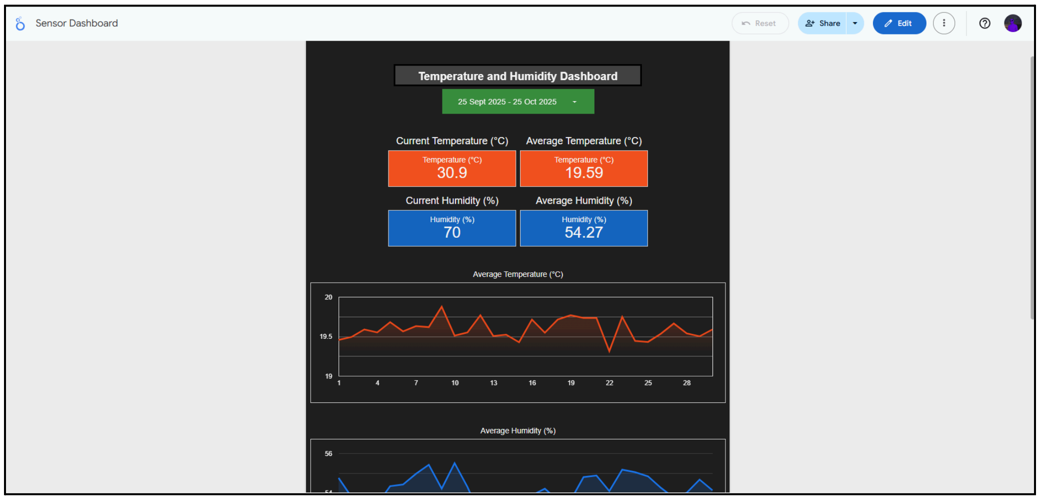
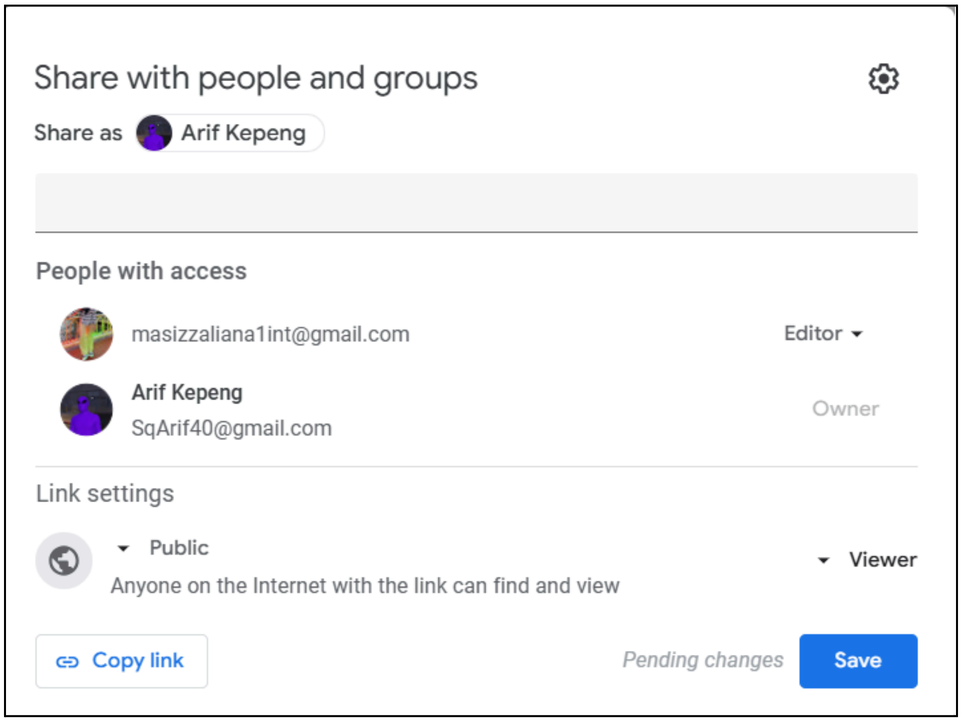


Figure 24: Looker Studio View Mode.

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

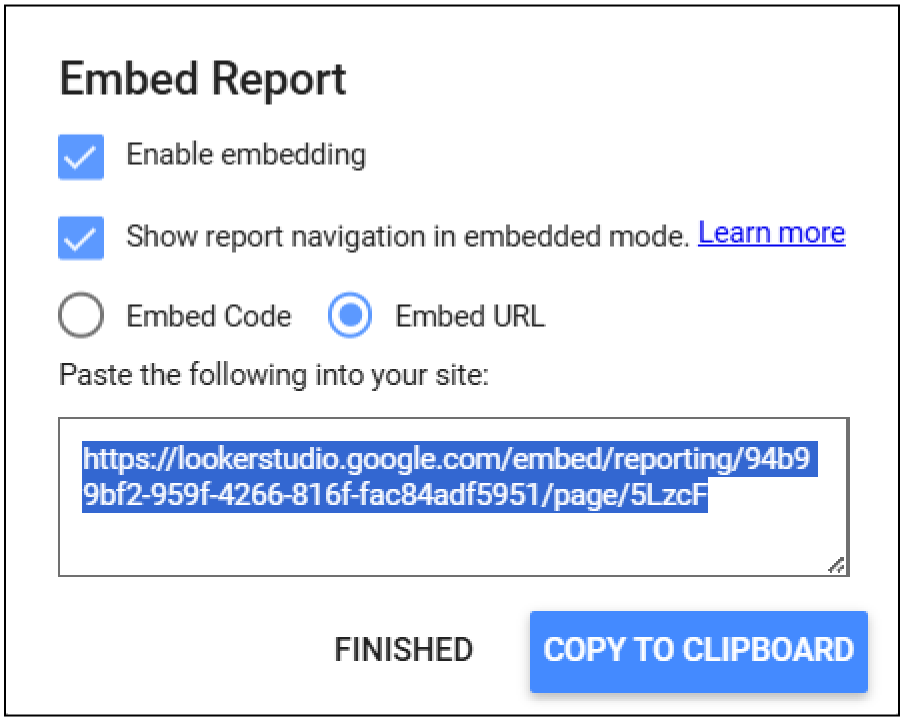
Public

Anyone on the Internet with the link can find and view

Viewer

Figure 25: 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 26: 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.

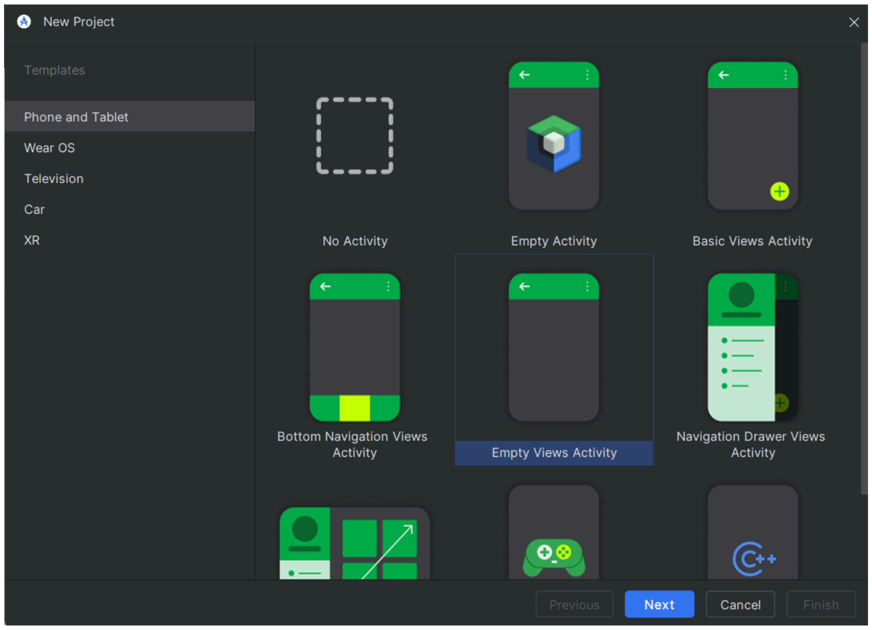
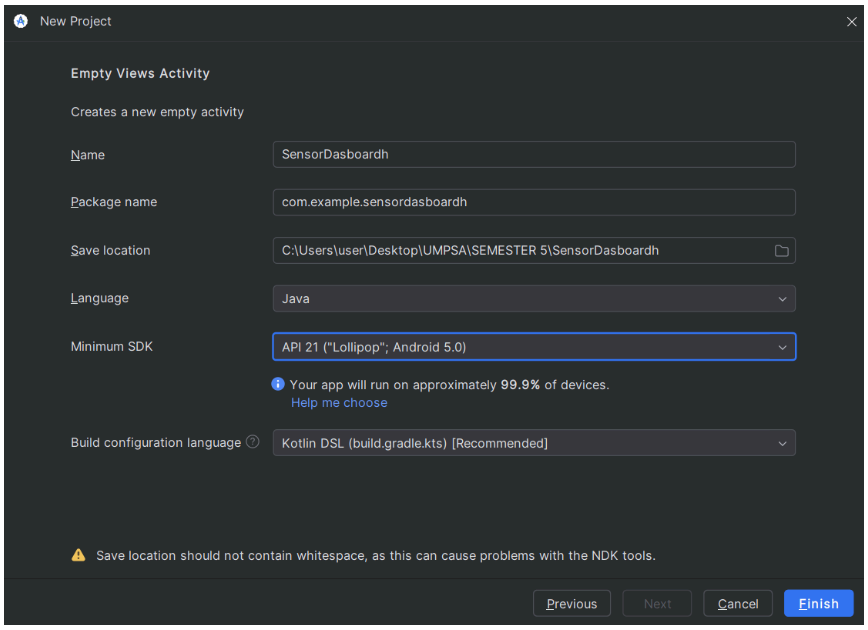


Figure 27: Select Empty Views Activity and Click Next.

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 28: 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>
```

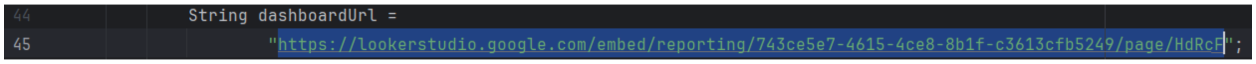
Update the MainActivity.java file code with this code:

```
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 29: 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.

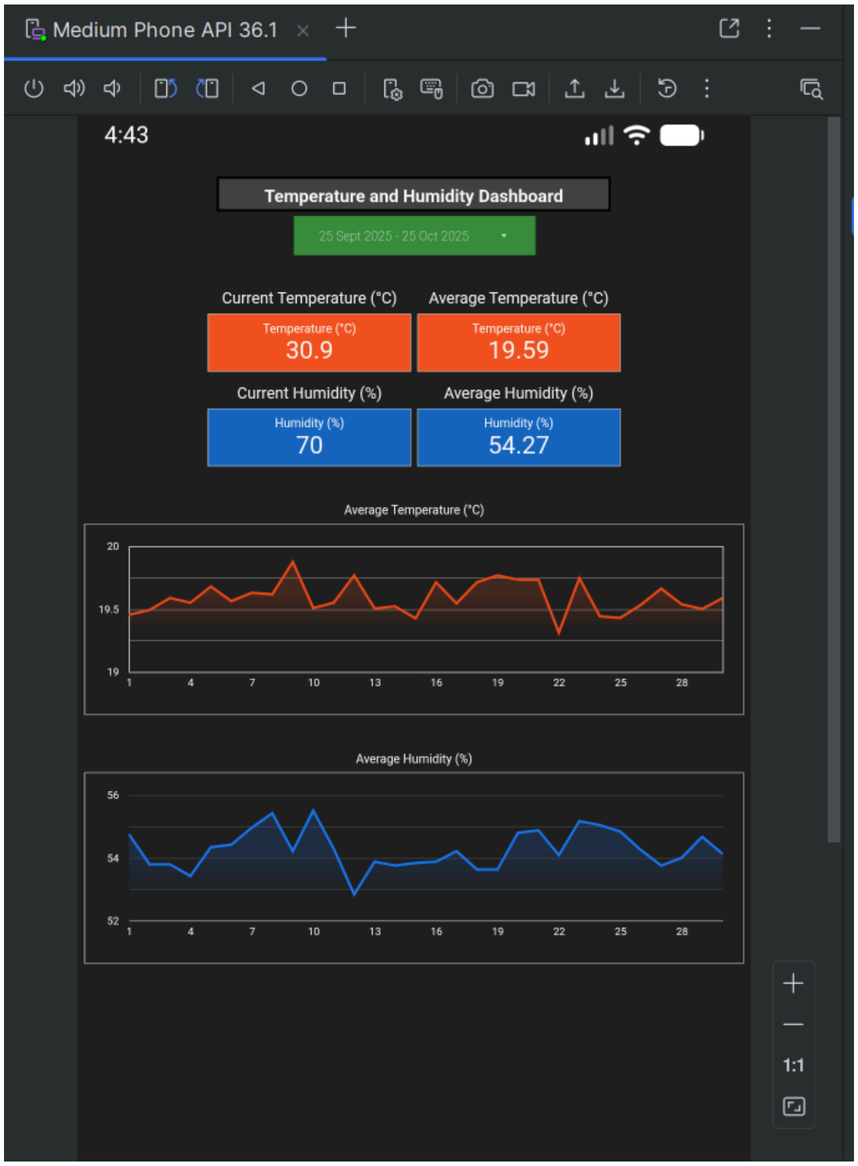
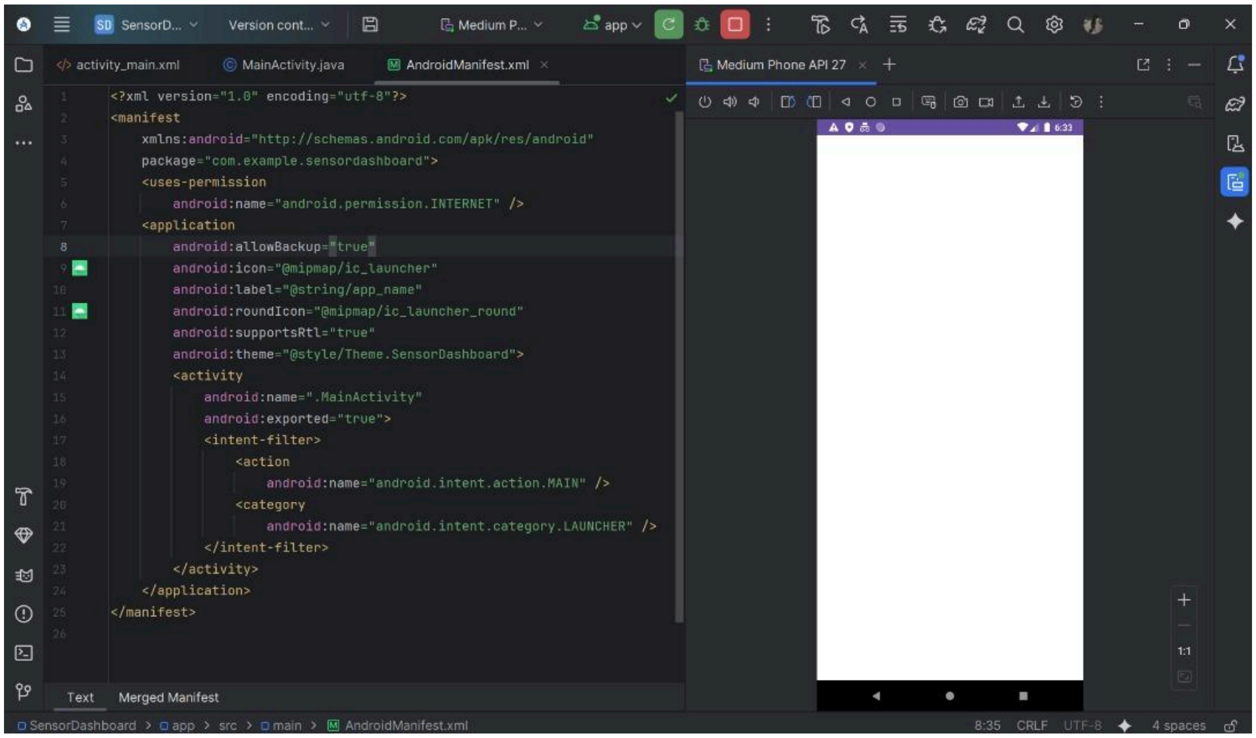


Figure 30: Created Looker Studio Dashboard Successfully Run in App Using Android Studio.

Challenges

The challenge was while running the Looker Studio Dashboard app in Android Studio where the screen appeared completely blank white even after refreshing.



```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 31: SensorDashboard App Appear Blank White.

Solutions

The solution to this challenge was to repeat back step 2 which recreate back the Looker Studio Dashboard. Then reuse the same activity_main.xml, MainActivity.java and AndroidManifest.xml codes in the same project file. After running the app, the outcome are as shown in Figure 29.