FAKULTI TEKNOLOGI KEJURUTERAAN ELEKTRIK DAN ELEKTRONIK

PROJECT:

AI-INTEGRATION SENSOR FORECASTING AND SUMMARIZATION

BVI 3114

TECHNOLOGY SYSTEM OPTIMIZATION II

BIL	NO ID	NAME
1	VC22030	RAZMIN BIN ABDUL RAHIM
2	VC22015	ISYRAF HILMI BIN IMRAN

TABLE OF CONTENTS

1. EXECUTIVE SUMMARY

This project showcases a smart, end-to-end solution for real-time sensor data monitoring, forecasting, and analysis using Google Apps Script. Light intensity readings are continuously collected and stored in Google Sheets, enabling centralized data management. Advanced forecasting techniques, including Exponential Moving Average (EMA) and Holt-Winters, are applied to predict future trends accurately. To enhance data interpretation, Gemini AI is integrated to automatically generate insightful summaries, offering a clear understanding of patterns and anomalies directly within the spreadsheet.

2. SYSTEM ARCHITECTURE DIAGRAM & WORKFLOW

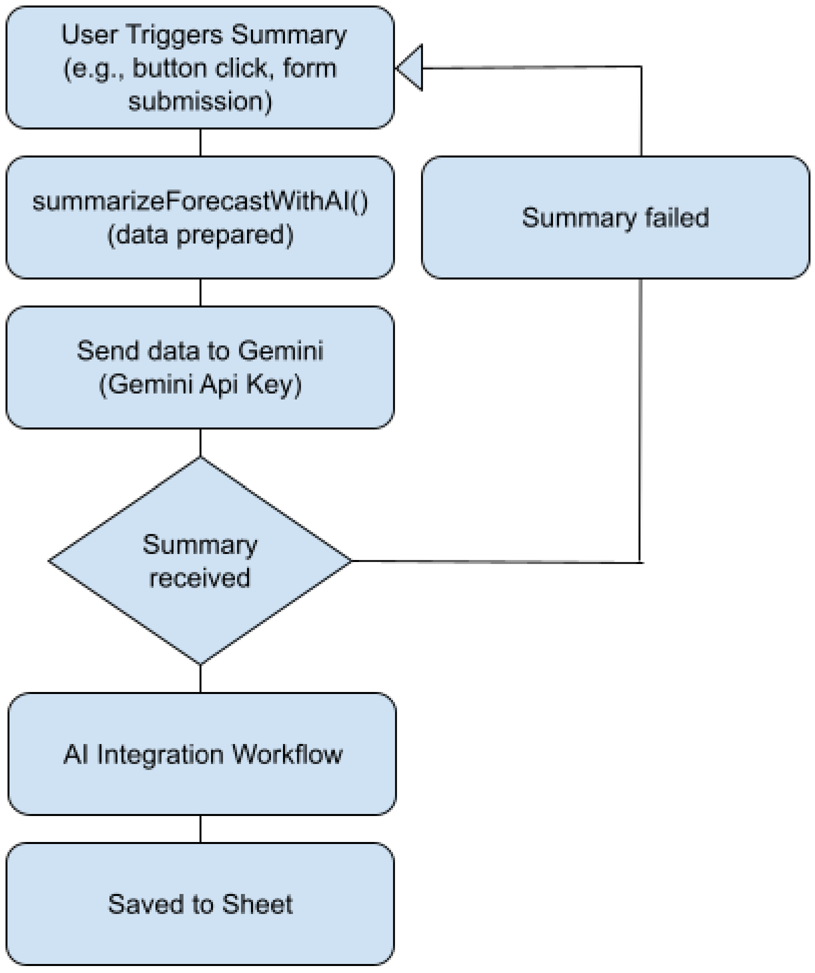

User Triggers Summary (e.g., button click, form submission)
summarizeForecastWithAI() (data prepared)
Send data to Gemini (Gemini Api Key)
Summary received
AI Integration Workflow
Saved to Sheet
Summary failed

Figure 1: System Architecture Diagram

<u>Here's the sequence of operations for the "System Architecture Diagram " in a list:</u>

- <u>User Starts Summary</u>
  User clicks a button or submits a form to generate a summary.

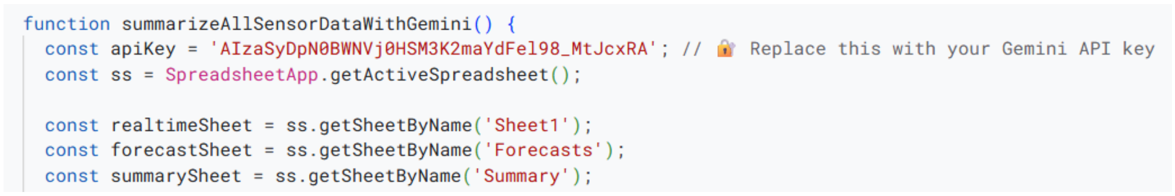
- <u>Text is Prepared</u>
  The system gets the text from a spreadsheet, cleans it, and breaks it into parts if it's too long.

- <u>Send to Gemini AI</u>
  The cleaned text is sent to the Gemini API with instructions to summarize.

- <u>Summary is Received</u>
  Gemini AI returns a short summary based on the input text.

- <u>Save to Spreadsheet</u>
  The summary is saved back to the spreadsheet along with the original text.

2.1 **User Starts Summary**

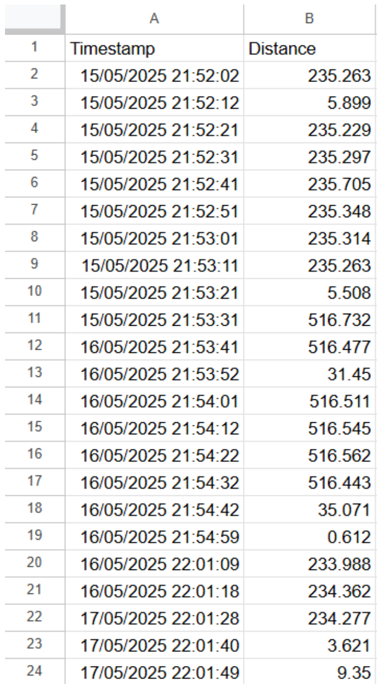


```
function summarizeAllSensorDataWithGemini() {
  const apiKey = 'AIzaSyDpN0BWNVj0HSM3K2maYdFel98_MtJcxRA'; // 🔐 Replace this with your Gemini API key
  const ss = SpreadsheetApp.getActiveSpreadsheet();

  const realtimeSheet = ss.getSheetByName('Sheet1');
  const forecastSheet = ss.getSheetByName('Forecasts');
  const summarySheet = ss.getSheetByName('Summary');
```

- This action starts the process to generate a summary from existing text data.
- The text to be summarized is typically stored in Google Sheets

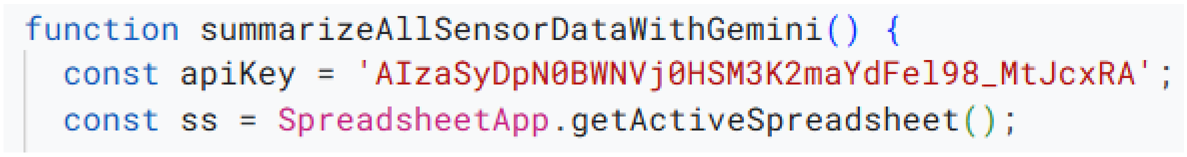
2.2 **Text is Prepared**



	A	B
1	Timestamp	Distance
2	15/05/2025 21:52:02	235.263
3	15/05/2025 21:52:12	5.899
4	15/05/2025 21:52:21	235.229
5	15/05/2025 21:52:31	235.297
6	15/05/2025 21:52:41	235.705
7	15/05/2025 21:52:51	235.348
8	15/05/2025 21:53:01	235.314
9	15/05/2025 21:53:11	235.263
10	15/05/2025 21:53:21	5.508
11	15/05/2025 21:53:31	516.732
12	16/05/2025 21:53:41	516.477
13	16/05/2025 21:53:52	31.45
14	16/05/2025 21:54:01	516.511
15	16/05/2025 21:54:12	516.545
16	16/05/2025 21:54:22	516.562
17	16/05/2025 21:54:32	516.443
18	16/05/2025 21:54:42	35.071
19	16/05/2025 21:54:59	0.612
20	16/05/2025 22:01:09	233.988
21	16/05/2025 22:01:18	234.362
22	17/05/2025 22:01:28	234.277
23	17/05/2025 22:01:40	3.621
24	17/05/2025 22:01:49	9.35

- The system retrieves the original text (e.g., sensor logs, notes, or descriptions) from a specific sheet.
- Basic cleaning and formatting are done to ensure the text is suitable for AI processing

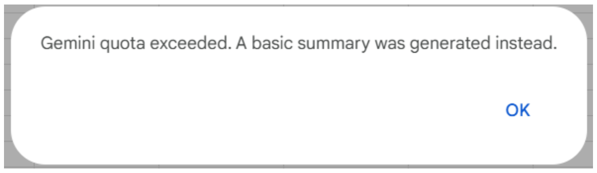
2.3 **Send To Gemini AI**



```
function summarizeAllSensorDataWithGemini() {
  const apiKey = 'AIzaSyDpN0BWNVj0HSM3K2maYdFel98_MtJcxRA';
  const ss = SpreadsheetApp.getActiveSpreadsheet();
```

- This step includes authentication using an API key and model selection (e.g., `gemini-1.5-pro`)
- The cleaned text is sent over the internet to the **Gemini AI API**

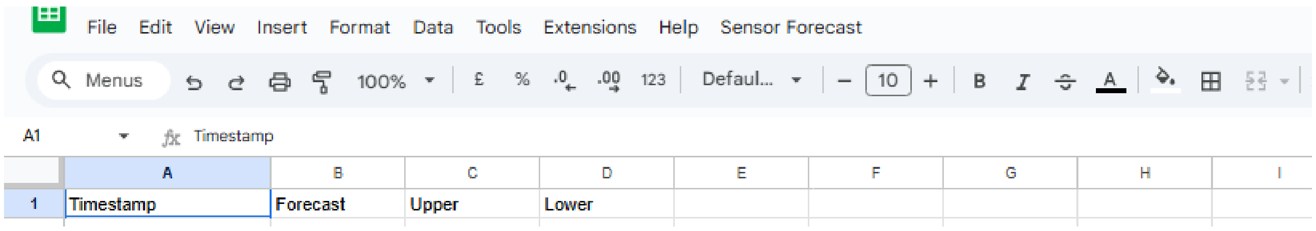
2.4 **Summary Is Received**



Gemini quota exceeded. A basic summary was generated instead.

- Gemini processes the prompt and generates a summarized version of the original text.
- This response is returned as part of the API call output.

2.5 **Save To Spreadsheet**



File Edit View Insert Format Data Tools Extensions Help Sensor Forecast
Menus 100% £ % .0 .00 123 Defaul... 10 B I A
A1 fx Timestamp

	A	B	C	D	E	F	G	H	I
1	Timestamp	Forecast	Upper	Lower					

- The returned summary is saved into the Google Sheet, alongside the original text.
- A new row is added for each processed entry with timestamps

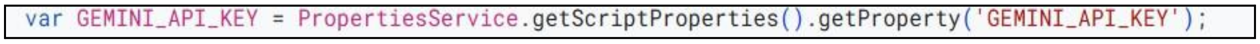
3. GOOGLE APPS SCRIPT BACKEND

The system's backend is entirely developed using Google Apps Script within a single .gs file, enabling seamless integration of data collection, forecasting, AI summarization, and spreadsheet interactivity.

- **Data Logging**
  Incoming light sensor readings are captured via doPost(e) requests and stored with timestamps in Sheet1, enabling real-time data collection.
- **Forecast Pipeline**
  The generateCombinedForecasts() function processes the logged data to generate predictions using two key algorithms:
  - *Exponential Moving Average (EMA)*
  - *Holt-Winters Seasonal Forecasting*
- **AI Summarization**
  The summarizeForecastWithAI() function communicates with the Gemini API to transform forecast data into natural-language insights, providing automatic summaries within the spreadsheet.
- **Dynamic Charting**
  Forecast visualization is handled directly through the Sheets interface using SpreadsheetApp.getActiveSpreadsheet().getSheetByName(...), ensuring charts are updated in real-time.
- **Forecast Reset**
  The clearForecasts() function removes previously generated forecast results to prevent confusion and maintain data clarity.

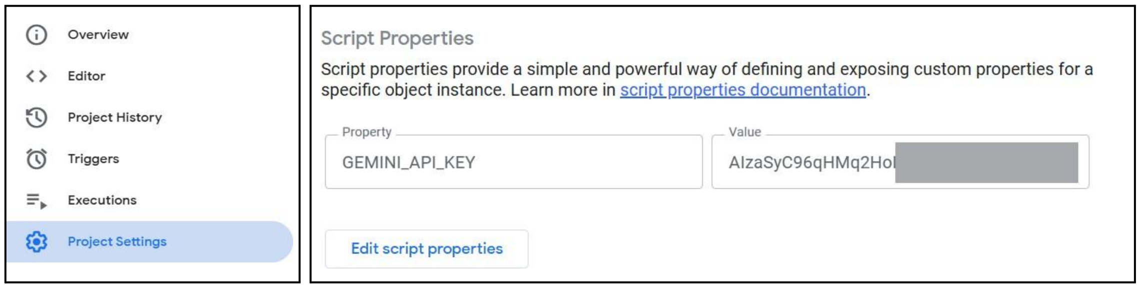
Security & API Integration:

- Uses `PropertiesService.getScriptProperties().getProperty("GEMINI_API_KEY")` to securely manage API keys, which are stored in **Project Settings > Script Properties** for centralized and hidden access



```
var GEMINI_API_KEY = PropertiesService.getScriptProperties().getProperty('GEMINI_API_KEY');
```

- All AI API interactions and exceptions are logged via `Logger.log()` to aid debugging.



Script Properties

Script properties provide a simple and powerful way of defining and exposing custom properties for a specific object instance. Learn more in script properties documentation.

Property	Value
GEMINI_API_KEY	AIzaSyC96qHMq2Hol[redacted]

Best Practices Implemented:

- Minimal hardcoded values uses constants and environment-based configurations.
- Error handling for API timeouts, missing values, and malformed inputs.
- Modularized forecast and summary functions for reusability.

```
function doPost(e) {
  try {
    var sheet =
SpreadsheetApp.openByUrl("https://docs.google.com/spreadsheets/d/1i_UG
_e6P4jl7CRTIaKHLuHTGTJW9Z1IRs1d-
MNkviP4/edit?gid=0#gid=0").getSheetByName("Sheet1");
    if (!sheet) {
      throw new Error("Sheet1 not found.");
    }

    var data = JSON.parse(e.postData.contents);
    var distance = data.distance;
    var timestamp = new Date();

    sheet.appendRow([timestamp, distance]);
    return ContentService.createTextOutput("Success");
  } catch (error) {
    return ContentService.createTextOutput("Error: " + error.message);
  }
}
```

4. SENSOR DATA ACQUISITION

- **Receives** a POST request from a device (e.g., ESP32). // Get the Sheet1
- **Parses** the JSON body to extract the `light` value. // Extract the 'light' field
- **Appends** the timestamp and light value as a new row in **Sheet1**. // Add a new row with timestamp and light value
- **Returns** a JSON response indicating success or failure. // If something goes wrong, return error response

5. **FORECASTING ALGORITHM SELECTION (EMA)**

This project uses two time-series forecasting methods: Exponential Moving Average (EMA) and Holt-Winters (Holt's Linear Smoothing). These were chosen because they are both accurate and efficient for handling real-time sensor data.

5.1 Exponential Moving Average (EMA)

EMA is a forecasting technique that gives more weight to recent data points, making it sensitive to short-term changes while smoothing out random noise.

- Formula: `EMA[i] = alpha * value[i] + (1 - alpha) * EMA[i-1]`
- Captures short-term trends

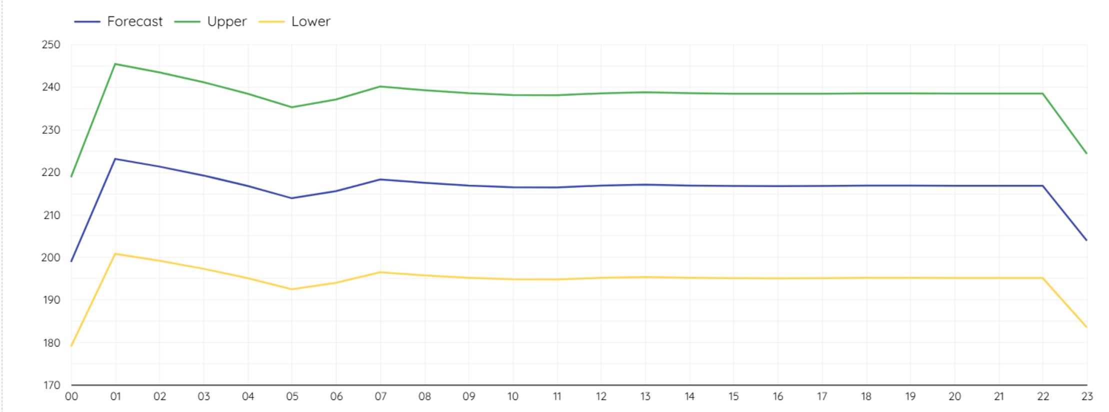
5.2 Justification for Choice of Algorithms

Algorithm	Reason for use
EMA (Exponential Moving Average)	Fast, simple and efficient at identifying short-term patterns and noise reduction

Table 1: Forecasting Data and Upper, Lower Data

Timestamp	Forecast	Upper	Lower
18/05/2025 23:05:18	203.84	224.23	183.46
19/05/2025 00:05:18	198.84	218.73	178.96
19/05/2025 01:05:18	223.04	245.35	200.74
19/05/2025 02:05:18	221.25	243.37	199.12
19/05/2025 03:05:18	219.15	241.06	197.23
19/05/2025 04:05:18	216.69	238.36	195.02
19/05/2025 05:05:18	213.80	235.18	192.42

6. **CHART & VISUALIZATION**



Average Upper Bound	Average Forecast	Average Lower Bound
Upper 237.49	Forecast 215.9	Lower 194.31

	Timestamp	Forecast ▾	Upper	Lower
1.	19 May 2025, 01:05:18	223.04	245.35	200.74
2.	19 May 2025, 02:05:18	221.25	243.37	199.12
3.	19 May 2025, 03:05:18	219.15	241.06	197.23

The chart is generated dynamically within the `Forecasts` sheet.

Line charts show:

- Forecast and distance
- Upper/lower bounds for future estimates

7. **IMPLEMENTATION CHALLENGES AND SOLUTION**

- **Handling API Quota Limits** – Mitigated with fallback summaries.
- **Timestamp Parsing Issues** – Handled by normalizing date input.
- **Visualization Accuracy**– Ensured through in-sample vs out-of-sample validation.

8. **CHALLENGE**

- API Limit

  – Too many requests caused Gemini API to stop.
  → Fixed by reducing how often it sends.

- Date Format Error

  – Time format from sensor caused problems.

  → Fixed by standardizing the format.

- Wrong Forecast During Spikes

  – Sudden changes made forecast not accurate.

  → Tuned the forecast settings.

- Chart Confusing

  – Forecast and real data mixed in chart.

  → Used colors and labels to make it clear.

- AI Summary Too General

  – Gemini gave boring or repeated summaries.

  → Gave better prompts with more info.

9. FUTURE IMPROVEMENTS

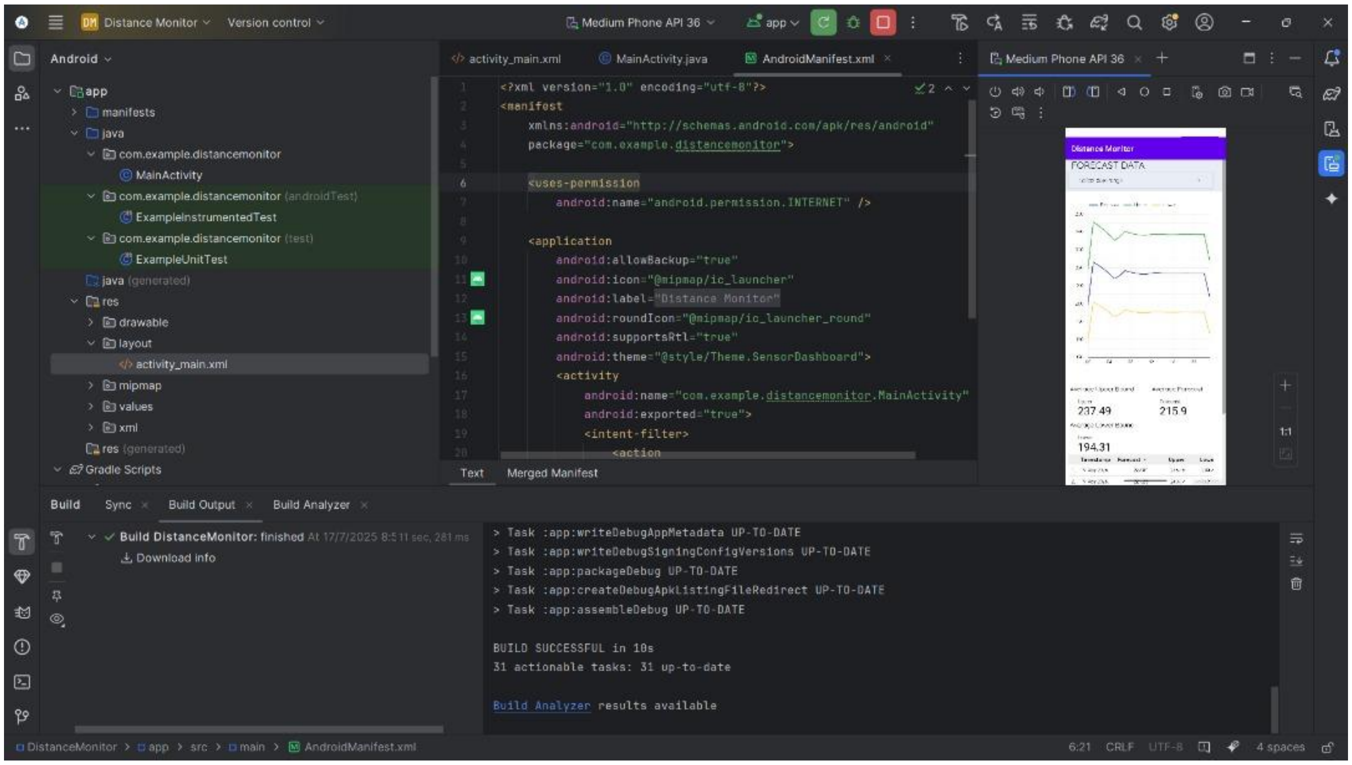
- Integrate email/PDF export of summaries
- Add ML model comparison (e.g., LSTM)
- Schedule daily summary generation
- Add authentication/token rotation for secure access

10. CONCLUSION

This project effectively integrates real-time sensor data acquisition, time series forecasting, and AI-powered summarization into a unified, cloud-based platform. By leveraging Google Apps Script, it showcases a lightweight yet powerful solution for bridging IoT data with intelligent analytics. The system not only automates data logging and forecasting but also enhances human interpretation through natural language summaries, demonstrating a practical approach to smart monitoring and data-driven decision-making.

11. **REFERENCES**

IMAGES OF ANDROID STUDIO APP



```
<?xml version="1.0" encoding="utf-8"?>
<manifest
    xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.distancemonitor">

    <uses-permission
        android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Distance Monitor"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.SensorDashboard">
        <activity
            android:name="com.example.distancemonitor.MainActivity"
            android:exported="true">
            <intent-filter>
                <action
```

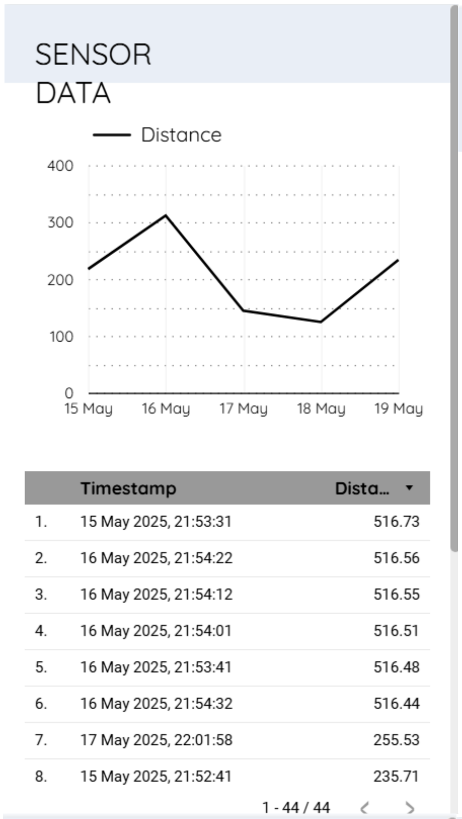
```
> Task :app:writeDebugAppMetadata UP-TO-DATE
> Task :app:writeDebugSigningConfigVersions UP-TO-DATE
> Task :app:packageDebug UP-TO-DATE
> Task :app:createDebugApkListingFileRedirect UP-TO-DATE
> Task :app:assembleDebug UP-TO-DATE

BUILD SUCCESSFUL in 10s
31 actionable tasks: 31 up-to-date

Build Analyzer results available
```

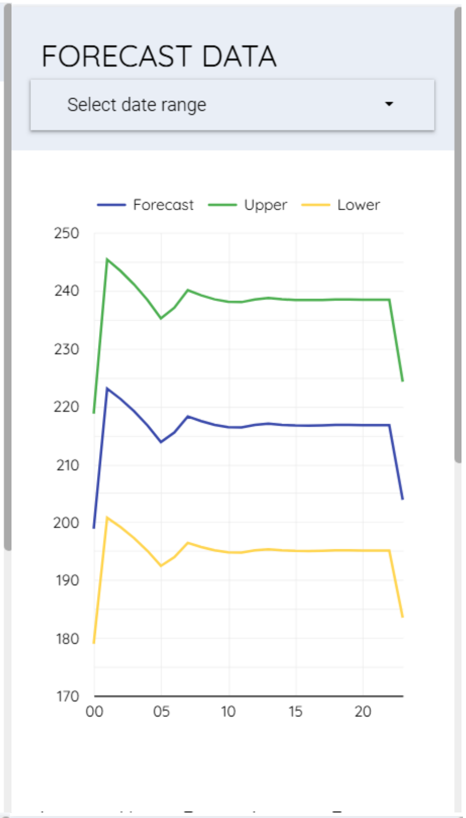
Android Studio provides an integrated environment where you can write code, design layouts, and run your app on an emulator to test and debug in real time.

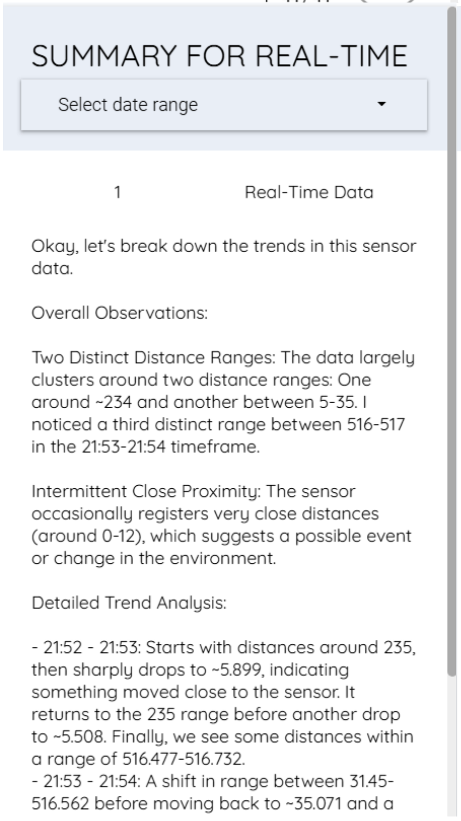
IMAGES OF ANDROID APP



	Timestamp	Dista...
1.	15 May 2025, 21:53:31	516.73
2.	16 May 2025, 21:54:22	516.56
3.	16 May 2025, 21:54:12	516.55
4.	16 May 2025, 21:54:01	516.51
5.	16 May 2025, 21:53:41	516.48
6.	16 May 2025, 21:54:32	516.44
7.	17 May 2025, 22:01:58	255.53
8.	15 May 2025, 21:52:41	235.71

1 - 44 / 44





1 Real-Time Data

Okay, let's break down the trends in this sensor data.

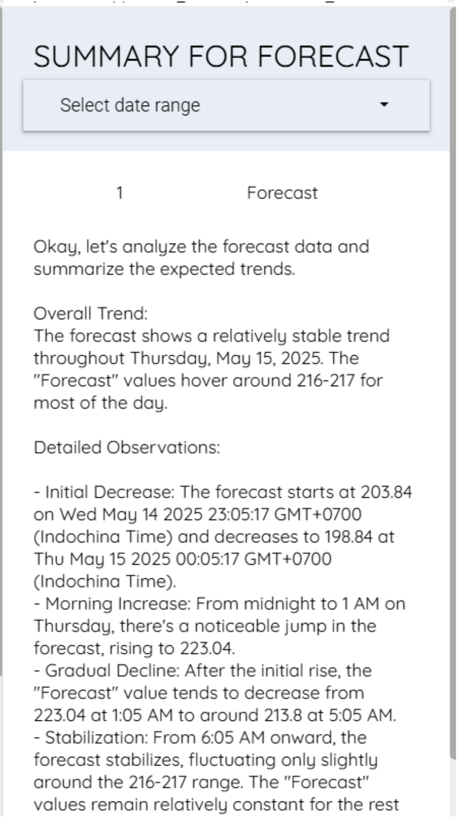
Overall Observations:

Two Distinct Distance Ranges: The data largely clusters around two distance ranges: One around ~234 and another between 5-35. I noticed a third distinct range between 516-517 in the 21:53-21:54 timeframe.

Intermittent Close Proximity: The sensor occasionally registers very close distances (around 0-12), which suggests a possible event or change in the environment.

Detailed Trend Analysis:

- 21:52 - 21:53: Starts with distances around 235, then sharply drops to ~5.899, indicating something moved close to the sensor. It returns to the 235 range before another drop to ~5.508. Finally, we see some distances within a range of 516.477-516.732.
- 21:53 - 21:54: A shift in range between 31.45-516.562 before moving back to ~35.071 and a



1 Forecast

Okay, let's analyze the forecast data and summarize the expected trends.

Overall Trend:
The forecast shows a relatively stable trend throughout Thursday, May 15, 2025. The "Forecast" values hover around 216-217 for most of the day.

Detailed Observations:

- Initial Decrease: The forecast starts at 203.84 on Wed May 14 2025 23:05:17 GMT+0700 (Indochina Time) and decreases to 198.84 at Thu May 15 2025 00:05:17 GMT+0700 (Indochina Time).
- Morning Increase: From midnight to 1 AM on Thursday, there's a noticeable jump in the forecast, rising to 223.04.
- Gradual Decline: After the initial rise, the "Forecast" value tends to decrease from 223.04 at 1:05 AM to around 213.8 at 5:05 AM.
- Stabilization: From 6:05 AM onward, the forecast stabilizes, fluctuating only slightly around the 216-217 range. The "Forecast" values remain relatively constant for the rest

Link of Google Sheet

https://docs.google.com/spreadsheets/d/1i_UG_e6P4jl7CRTIaKHLuHTGTJW9Z1IRs1d-MNkviP4/edit?gid=0#gid=0

Link of Looker Studio Dashboard

https://lookerstudio.google.com/embed/reporting/f33214b6-2a00-4b58-99f8-512dea7a9f58/page/2EZRF

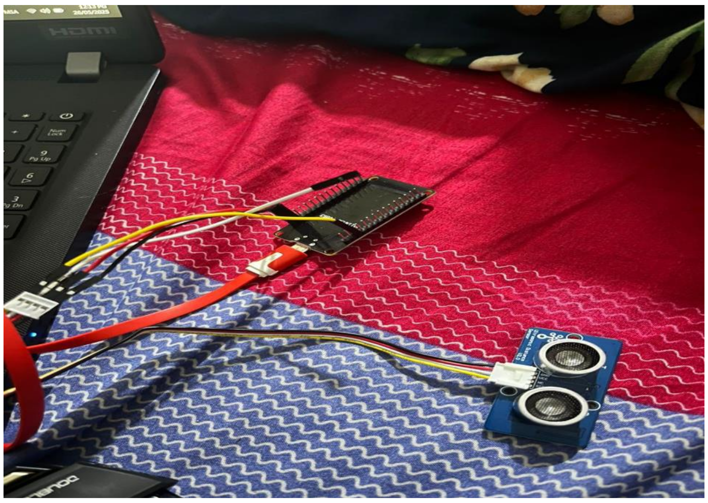
Full code snippet (with commented):

https://lookerstudio.google.com/reporting/f33214b6-2a00-4b58-99f8-512dea7a9f58/page/2EZRF/edit

Link of GitHub

https://github.com/RAZSTAR01/MiniProjectTSO2

12. **HARDWARE PHOTO AND ASSEMBLY GUIDE**



ULTRASONIC RANGE:

- Connect VCC to 5V on ESP32
- Connect GND to GND on ESP32
- Connect SIG to a digital GPIO