Name: G.Karthick Raj
Register No. : 203002045

IOT BASED WEATHER REPORTING SYSTEM

AIM:

The main aim of the project is to develop a IoT based weather reporting system.
The main objectives of the weather reporting system are as follows:

1. To design a system that collects data about weather in real time and is highly reliable and accurate.
2. To develop an alerting system using some threshold values for different weather parameters.
3. To develop a report that is useful in sports for athletes, coaches, and spectators. With this project, real-time weather data can be gathered and analyzed to provide valuable insights for various outdoor sports activities.

HARDWARE REQUIREMENTS:

Raspberry Pi 4B, power cables, jumper wires, HDMI cable, BMP 180 sensor, DHT11 sensor, Rain sensor, pendrive.

SOFTWARE REQUIREMENTS:

Raspbian OS, Python IDLE v3.7, Google locker studio, Integromat

THEORY:

DHT11 SENSOR:

The DHT11 sensor is a low-cost, digital humidity and temperature sensor that can be used in a wide range of applications. It consists of a capacitive humidity sensor and a thermistor for measuring temperature, and it communicates with microcontrollers via a single-wire digital interface. One of the main advantages of the DHT11 sensor is its affordability, making it an ideal choice for projects where cost is a primary consideration. It is also relatively easy to use and comes in a small, compact package. However, the DHT11 sensor has some limitations. Its accuracy can be affected by factors such as temperature changes, humidity changes, and the location of the sensor. It also has a relatively slow response time and is not suitable for applications that require high levels of precision. Overall, the DHT11 sensor is a popular choice for hobbyists and makers who want to add temperature and humidity sensing capabilities to their projects without breaking the bank.

Fig.6.1 DHT11 Temperature and Humidity Sensor

Name: G.Karthick Raj
Register No. : 203002045

BMP 180 SENSOR:

The BMP180 sensor is a high-precision, low-power digital barometric pressure and temperature sensor. It is designed to measure atmospheric pressure and temperature, making it useful for a variety of applications including weather monitoring, altitude measurement, and navigation. One of the key advantages of the BMP180 sensor is its high level of accuracy. It can measure pressure with an accuracy of up to ±1 hPa, and temperature with an accuracy of up to ±1°C. It also has a very low power consumption, making it suitable for use in battery-powered devices. The BMP180 sensor communicates with microcontrollers via a standard I2C interface, and comes in a small, surface-mount package. It can be easily integrated into a variety of projects, and there are a number of libraries available for different microcontroller platforms to simplify the programming. Overall, the BMP180 sensor is a versatile and reliable choice for projects that require accurate pressure and temperature sensing. Its low power consumption and small form factor make it particularly well-suited for use in portable or battery-powered applications.

Fig.6.2 BMP 180 Pressure Sensor

RAIN SENSOR:

A rain sensor is a device that detects the presence and/or intensity of rain. There are several types of rain sensors, but one of the most common is the optical rain sensor, which works by measuring changes in the amount of light reflected or absorbed by a surface. Optical rain sensors typically consist of a transmitter and a receiver that are placed a short distance apart, and a surface that is designed to collect raindrops. When rain falls on the surface, it causes the amount of light that reaches the receiver to decrease, which can be detected and measured by the sensor. Rain sensors are commonly used in a variety of applications, including weather monitoring, irrigation systems, and automotive safety systems. For example, they can be used to automatically adjust the speed of windshield wipers in a car, or to shut off irrigation systems when it is raining. Overall, rain sensors are an important tool for monitoring and responding to weather conditions, and can help to conserve resources and improve safety in a variety of settings.

Fig.6.3 Rain Sensor

Name: G.Karthick Raj
Register No. : 203002045

GOOGLE LOCKER STUDIO:

Google Locker Studio is a cloud-based storage and management platform developed by Google. It is designed to provide a secure and flexible way for organizations to store and manage their digital assets, such as images, videos, and other media files. With Google Locker Studio, users can easily upload and organize their files, and control who has access to them. The platform also provides powerful search and filtering capabilities, making it easy to find and retrieve files quickly. One of the key benefits of Google Locker Studio is its integration with other Google tools and services, such as Google Drive and Google Workspace. This allows users to seamlessly collaborate on projects and share files with others, without having to switch between different platforms or applications. Google Locker Studio is particularly well-suited for organizations that need to manage large volumes of digital assets, or that require a high degree of control over who can access and use their files. It is also a cost-effective solution, as it eliminates the need for on-premise storage and reduces the risk of data loss or theft.

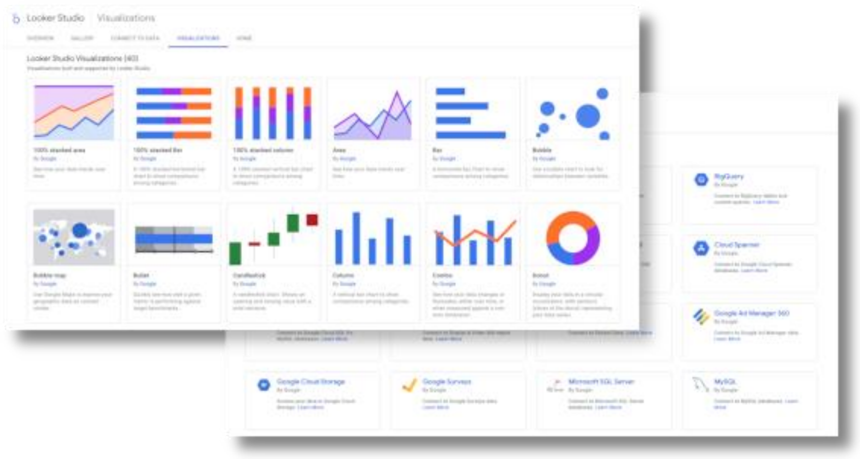


Fig.6.4 Google Locker Studio Interface

INTEGROMAT:

Integromat is a cloud-based automation platform that enables users to connect different apps and services and automate workflows between them. It provides a graphical user interface for creating complex automation workflows, without the need for coding or technical expertise. With Integromat, users can create automation scenarios that include a variety of actions, such as data extraction, filtering, transformation, and sharing. For example, they can create workflows that automatically post new social media content to a blog, or that send email notifications when specific events occur in other apps. Integromat integrates with over 500 different apps and services, including popular tools like Google Drive, Slack, Trello, and Shopify. It also includes a range of advanced features, such as error handling, loops, and conditional branching, which allow for even greater flexibility and control over automation workflows. Overall, Integromat is a powerful tool for streamlining and automating repetitive tasks, and can help users save time and increase productivity. Its ease of use and broad range of integrations make it accessible to users of all technical levels, from beginners to advanced users.

Name: G.Karthick Raj
Register No. : 203002045

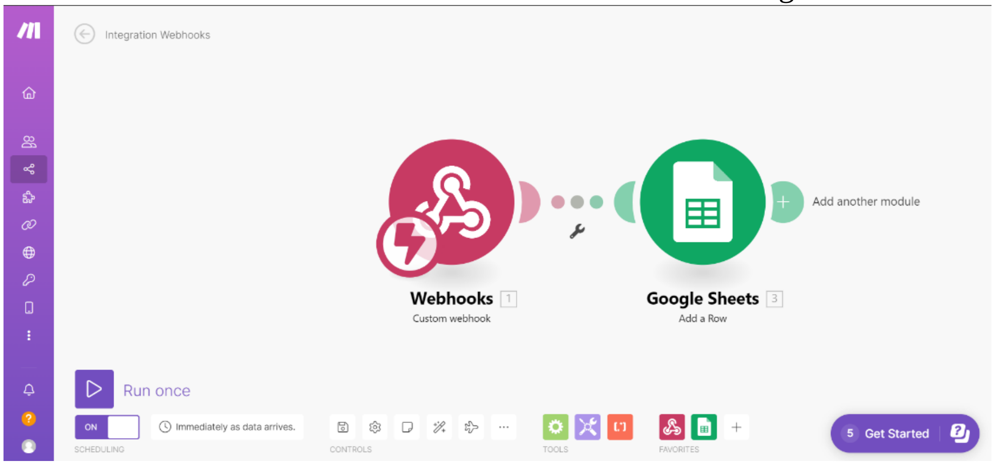


Fig.6.5 Integromat Interface

ALGORITHM:

1. Power up the Pi module and make the necessary connections.
2. Using the VNC viewer, connect the Raspberry pi via PC remotely.
3. Connect the different sensors according to the given circuit diagram.
4. Collect the data from the sensors and transfer it into Google Sheets using Integromat.
5. Open Google Locker Studio , log in to your account and create a new project.
6. Using the data collected , we visualize the data and make it into a report for analysis.
7. The report is useful for people involved in sports like athletes, coaches, and spectators.

Name: G.Karthick Raj
Register No. : 203002045

FLOW CHART:

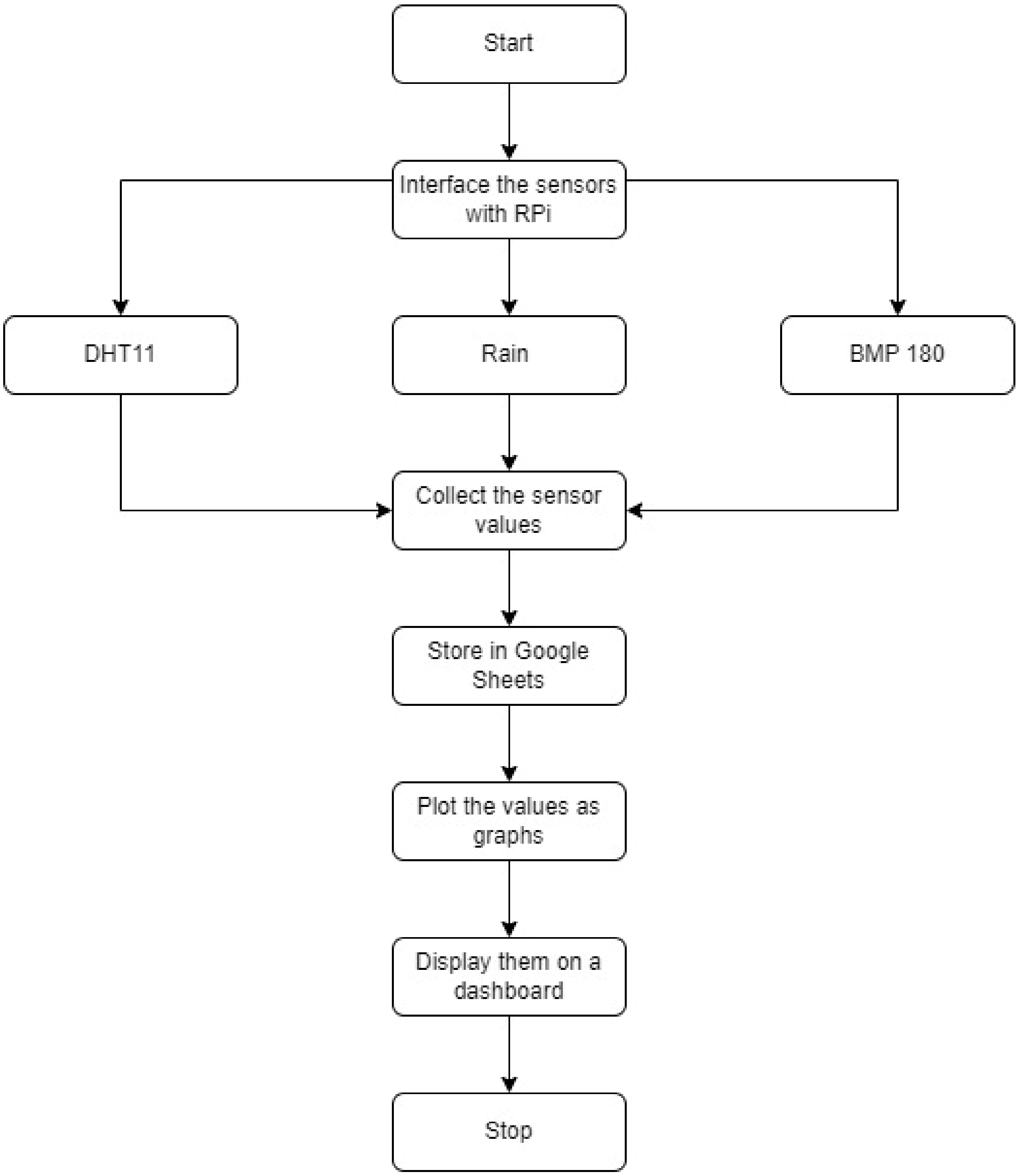

Start
Interface the sensors with RPi
DHT11
Rain
BMP 180
Collect the sensor values
Store in Google Sheets
Plot the values as graphs
Display them on a dashboard
Stop

Figure 6.6 Working Flowchart

Name: G.Karthick Raj
Register No. : 203002045

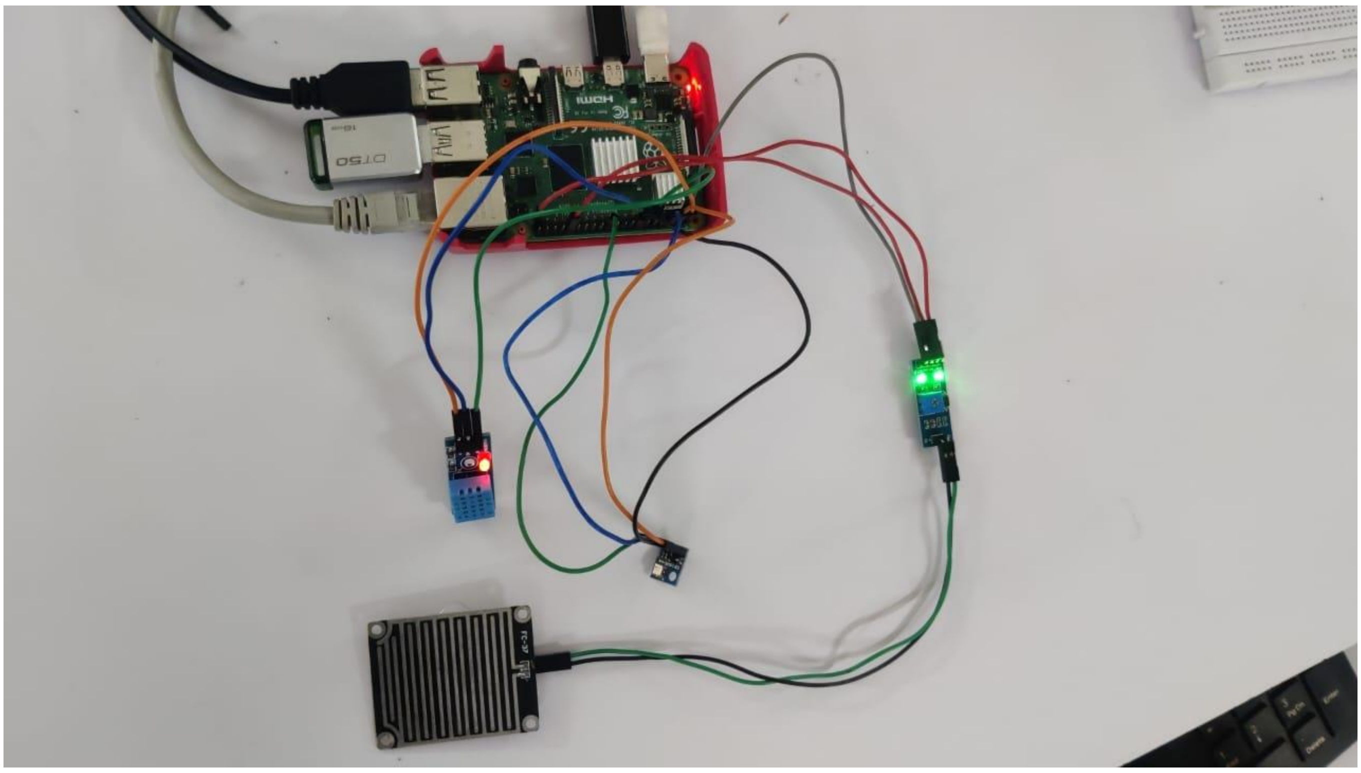
R-PI CODE:

```
import RPi.GPIO as GPIO
import datetime
import math
import requests
import time
import dht11
from twilio.rest import Client

from gpiozero import InputDevice, AnalogInputDevice
from time import sleep

GPIO.setmode(GPIO.BCM)
myDHT=dht11.DHT11(pin=17)

data_to_send={}
result=myDHT.read()

rain_pin = 12
rain_count=0.6475
rain=InputDevice(rain_pin)
# Set GPIO pin as input and enable pull-down resistor
GPIO.setup(rain_pin, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
Rain_percentage=format((rain_count / 60.0)*100)
time.sleep(1)

a = 17.27
b = 237.7
def dew_point(temp, hum):
    alpha = ((a * temp) / (b + temp)) + math.log(hum/100.0)
    return (b * alpha) / (a - alpha)
def relative_humidity(temp, dp):
    alpha = ((a * temp) / (b + temp)) + math.log(dp/100.0)
    return math.exp(alpha) * 100.0
def pressure_estimate(temp, rh):
    # Convert temperature to Kelvin
    temp = result.temperature + 273.15

    # Calculate the saturation vapor pressure
    es = 6.112 * math.exp((17.67 * temp) / (temp + 243.5))

    # Calculate the actual vapor pressure
    e = (rh / 100.0) * es

    # Calculate the pressure estimate
    p = ((e / 0.378) / (result.temperature / 216.6) ** 5.212/10000000)
    return p
# Example usage
dp = dew_point(result.temperature, 50.0) # Dew point at 50% relative humidity
```

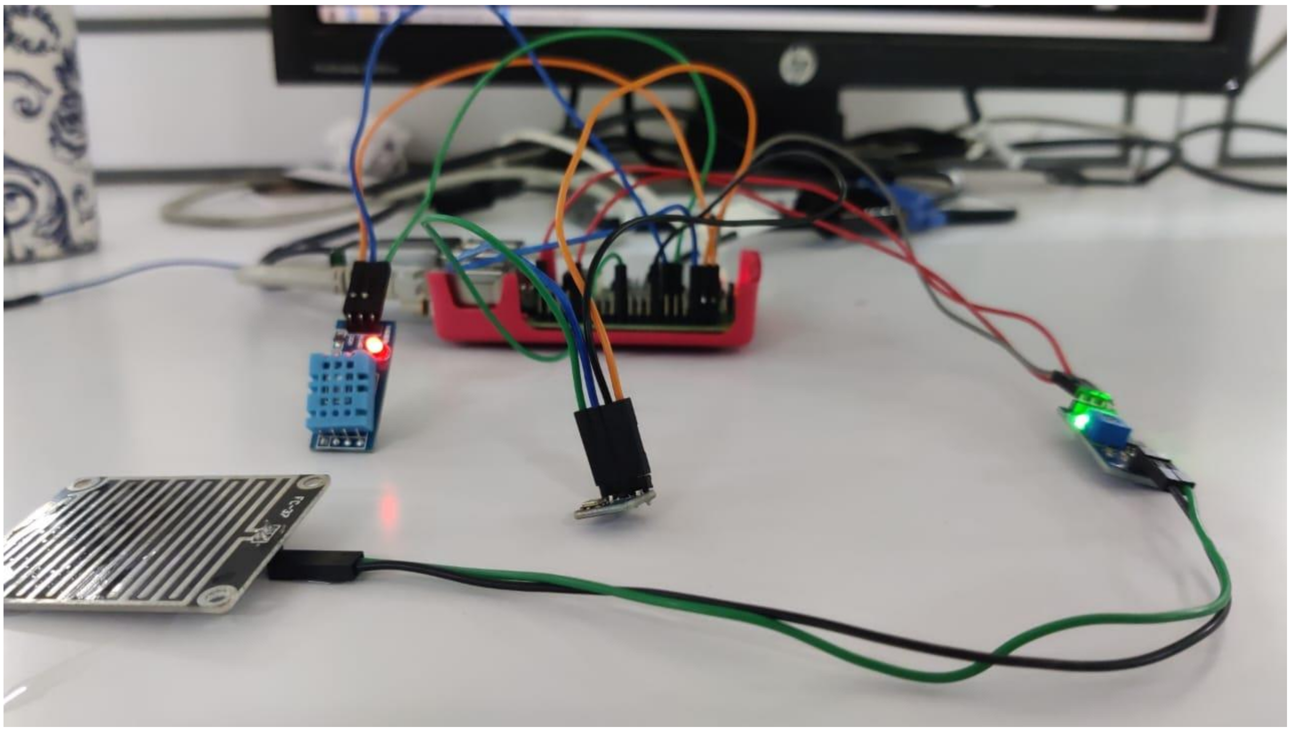
```
rh = relative_humidity(result.temperature, dp)
pressure = pressure_estimate(result.temperature, rh)
#print("Temperature: {:.1f} degrees Celsius".format(result.temperature))
#print("Dew point: {:.1f} degrees Celsius".format(dp))
#print("Relative humidity: {:.1f}%".format(rh))
#print("Pressure: {:.1f} hPa".format(pressure))
#print("Rain percentage: {:.1f}%".format((rain_count / 60.0) * 100.0))

data_to_send["Date"]= str(datetime.datetime.now())
data_to_send["Temperature"]=result.temperature
data_to_send["Humidity"]=result.humidity
data_to_send["Rain"]=rain.is_active
data_to_send["Pressure"]=pressure
if rain.is_active == False:
        data_to_send["Rain Percentage"]=Rain_percentage
else:
        data_to_send["Rain Percentage"]=0
print(data_to_send)
r=requests.post("https://hook.eu1.make.com/pwgt47l8l8ldy1uew1ocjsng2awbrs1q",json=data_to_sen
d)
print(r.status_code)

account_id='AC49a217050e42aa0184123e0f3c67a96f'
auth_token='d63d927c355b251c102e272e2b98cd8d'
client=Client(account_id,auth_token)
if (result.temperature>=20.00 and result.temperature<=32.00):
        print("Ideal Weather Conditions : Enjoy the pleasant day ")

message=client.api.account.messages.create(to='+919361795817',from_='+12765985345',body='Ide
al Weather Conditi>

elif (result.temperature>32.00 and result.temperature<=36.00):
        print("Hot Day!! Caution ")

message=client.api.account.messages.create(to='+919361795817',from_='+12765985345',body='Cau
tion: Chances of H>
else:
        print("Extremely Hot Day!! Caution  ")

message=client.api.account.messages.create(to='+919361795817',from_='+12765985345',body='Cau
tion : High Chance>
if (result.humidity>=60 and result.humidity<=70.00):
        print("Ideal Weather Conditions : Enjoy the pleasant day ")

message=client.api.account.messages.create(to='+919361795817',from_='+12765985345',body="Ide
al Weather Conditi>

elif (result.humidity>70.00 and result.humidity<=80.00):
        print("Humid Day!! Caution ")
```

```
message=client.api.account.messages.create(to='+919361795817',from_='+12765985345',body="Ca
ution: Chances of  >
else:
        print("Extremely Humid Day!! Caution  ")

 message=client.api.account.messages.create(to='+919361795817',from_='+12765985345',body="Ca
ution : High  chanc>

if (rain.is_active == False):
        print("Drizzling !! ")

message=client.api.account.messages.create(to='+919361795817',from_='+12765985345',body=
'Drizzling')
```

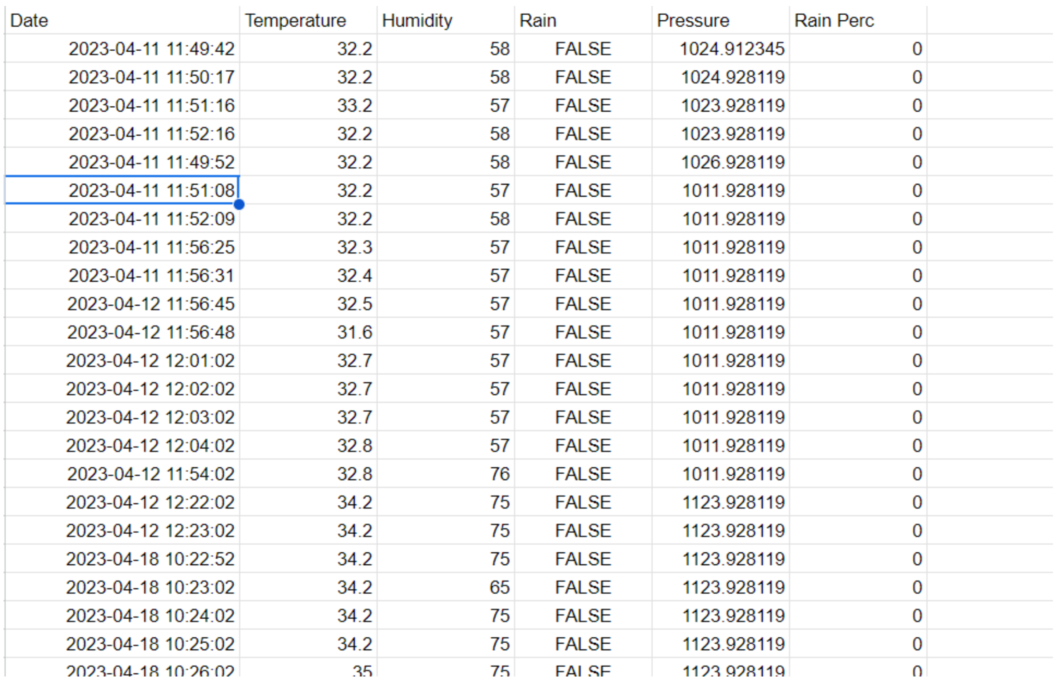
<u>Connection setup</u>



Name: G.Karthick Raj
Register No. : 203002045



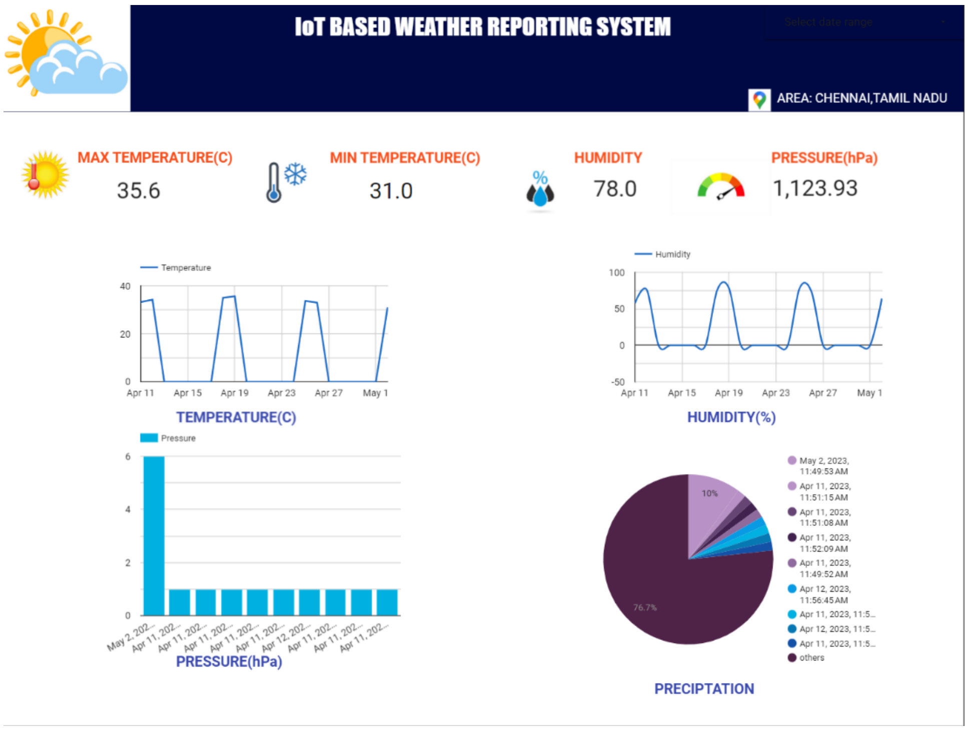
OUTPUT:

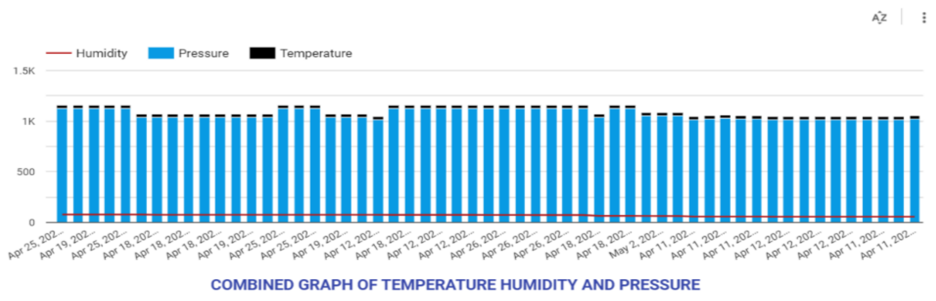


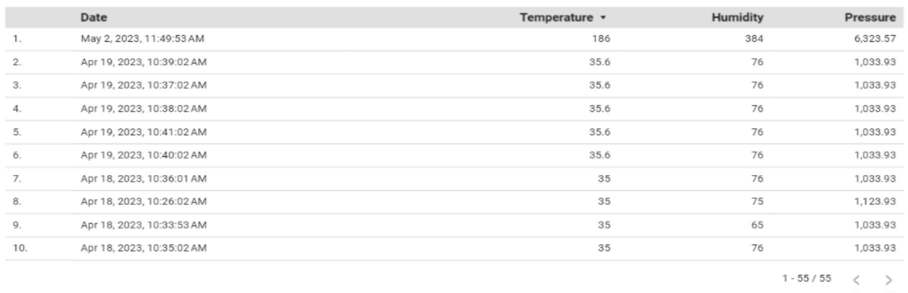
Date	Temperature	Humidity	Rain	Pressure	Rain Perc
2023-04-11 11:49:42	32.2	58	FALSE	1024.912345	0
2023-04-11 11:50:17	32.2	58	FALSE	1024.928119	0
2023-04-11 11:51:16	33.2	57	FALSE	1023.928119	0
2023-04-11 11:52:16	32.2	58	FALSE	1023.928119	0
2023-04-11 11:49:52	32.2	58	FALSE	1026.928119	0
2023-04-11 11:51:08	32.2	57	FALSE	1011.928119	0
2023-04-11 11:52:09	32.2	58	FALSE	1011.928119	0
2023-04-11 11:56:25	32.3	57	FALSE	1011.928119	0
2023-04-11 11:56:31	32.4	57	FALSE	1011.928119	0
2023-04-12 11:56:45	32.5	57	FALSE	1011.928119	0
2023-04-12 11:56:48	31.6	57	FALSE	1011.928119	0
2023-04-12 12:01:02	32.7	57	FALSE	1011.928119	0
2023-04-12 12:02:02	32.7	57	FALSE	1011.928119	0
2023-04-12 12:03:02	32.7	57	FALSE	1011.928119	0
2023-04-12 12:04:02	32.8	57	FALSE	1011.928119	0
2023-04-12 11:54:02	32.8	76	FALSE	1011.928119	0
2023-04-12 12:22:02	34.2	75	FALSE	1123.928119	0
2023-04-12 12:23:02	34.2	75	FALSE	1123.928119	0
2023-04-18 10:22:52	34.2	75	FALSE	1123.928119	0
2023-04-18 10:23:02	34.2	65	FALSE	1123.928119	0
2023-04-18 10:24:02	34.2	75	FALSE	1123.928119	0
2023-04-18 10:25:02	34.2	75	FALSE	1123.928119	0
2023-04-18 10:26:02	35	75	FALSE	1123.928119	0

Figure 6.7 Data collected in Google Sheets

Name: G.Karthick Raj
Register No. : 203002045







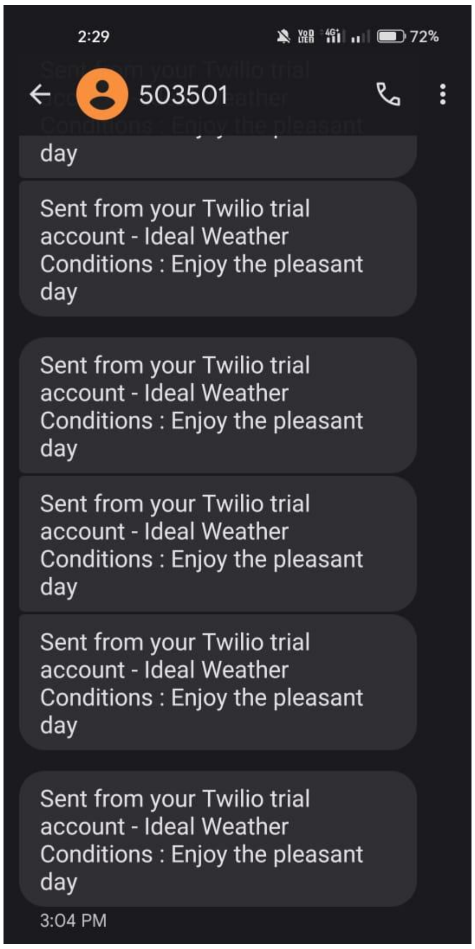
	Date	Temperature ▾	Humidity	Pressure
1.	May 2, 2023, 11:49:53 AM	186	384	6,323.57
2.	Apr 19, 2023, 10:39:02 AM	35.6	76	1,033.93
3.	Apr 19, 2023, 10:37:02 AM	35.6	76	1,033.93
4.	Apr 19, 2023, 10:38:02 AM	35.6	76	1,033.93
5.	Apr 19, 2023, 10:41:02 AM	35.6	76	1,033.93
6.	Apr 19, 2023, 10:40:02 AM	35.6	76	1,033.93
7.	Apr 18, 2023, 10:36:01 AM	35	76	1,033.93
8.	Apr 18, 2023, 10:26:02 AM	35	75	1,123.93
9.	Apr 18, 2023, 10:33:53 AM	35	65	1,033.93
10.	Apr 18, 2023, 10:35:02 AM	35	76	1,033.93

1 - 55 / 55

Fig 6.8 Visual Representation of data collected from sensors using Google Locker Studio

Name: G.Karthick Raj
Register No. : 203002045

Alerts :



2:29 72%

503501

day

Sent from your Twilio trial account - Ideal Weather Conditions : Enjoy the pleasant day

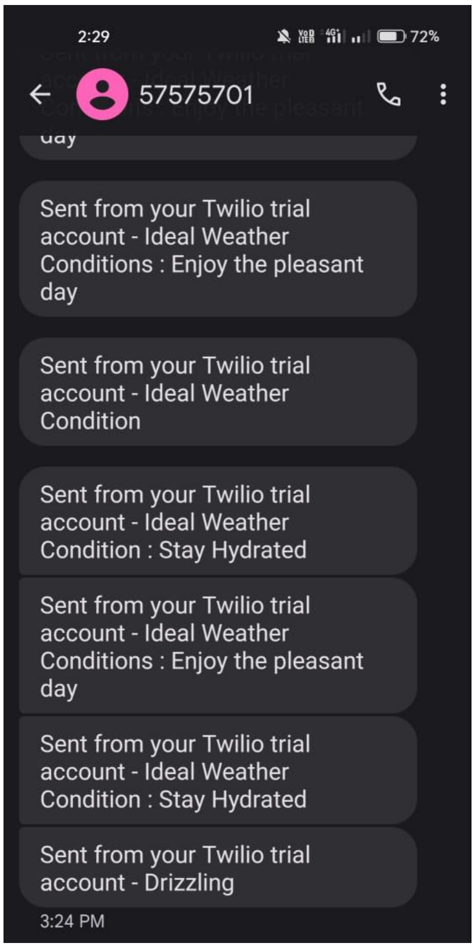
Sent from your Twilio trial account - Ideal Weather Conditions : Enjoy the pleasant day

Sent from your Twilio trial account - Ideal Weather Conditions : Enjoy the pleasant day

Sent from your Twilio trial account - Ideal Weather Conditions : Enjoy the pleasant day

Sent from your Twilio trial account - Ideal Weather Conditions : Enjoy the pleasant day

3:04 PM



2:29 72%

57575701

day

Sent from your Twilio trial account - Ideal Weather Conditions : Enjoy the pleasant day

Sent from your Twilio trial account - Ideal Weather Condition

Sent from your Twilio trial account - Ideal Weather Condition : Stay Hydrated

Sent from your Twilio trial account - Ideal Weather Conditions : Enjoy the pleasant day

Sent from your Twilio trial account - Ideal Weather Condition : Stay Hydrated

Sent from your Twilio trial account - Drizzling

3:24 PM

Fig 6.9 SMS Alert for different weather scenarios

Name: G.Karthick Raj
Register No. : 203002045

INFERENCE:

An IoT-based weather reporting system is a network of interconnected devices that collect and transmit weather data in real-time. These devices include sensors that measure various weather parameters such as temperature, humidity, pressure and precipitation. The collected data is then sent to a cloud-based platform where it is processed, analyzed and made available to end-users through various channels such as web-based dashboards, mobile applications, or SMS.

One of the primary benefits of an IoT-based weather reporting system is that it enables accurate and timely weather forecasts that are essential for various industries such as agriculture, transportation, and construction.

An IoT-based weather reporting system has immense potential to revolutionize the way we monitor and forecast weather conditions. With the growing demand for accurate and timely weather information, this technology can play a crucial role in enabling various industries to make informed decisions and improve their operations.

RESULT:

Thus an IoT based weather reporting system, which collects the data about temperature, humidity, pressure and precipitation in a particular area and displays it in the form of a dashboard and also displays the various weather parameters of a city using Google Locker Studio was developed.