



**Group Members**

**Submitted To**

**Dr. Rafia Mumtaz (Associate Professor)**

**Head of Department - Software Engineering**

**Office No. A-203, Faculity Block SEECS, NUST, Islamabad**

Saad Mahmud Mirza

400432 - MSCS 2022

SEECS, NUST H12

smirza.mscs22seecs@seecs.edu.pk

Tabinda Ghazal

402420 - MSCS 2022

SEECS, NUST H12

tghazal.mscs22seecs@seecs.edu.pk

Ayesha Sana

402433 - MSCS 2022

SEECS, NUST H12

ayeshasana.mscs22seecs@seecs.edu.pk

Maasooma

429876 - MSCS 2022

SEECS, NUST H12

maasooma.mscs22seecs@seecs.edu.pk

**Real Time Weather Station Monitoring System using NodeMCU ESP8266**

# Contents

[1 Contents 2](#_Toc135582602)

[**2** **Project Description** 3](#_Toc135582603)

[**2.1** **Objectives:** 3](#_Toc135582604)

[**2.2** **Introduction** 4](#_Toc135582605)

[**2.3** **Expected Outcomes** 4](#_Toc135582606)

[**2.4** **Significance of Monitoring Weather Conditions Using IoT Technology** 4](#_Toc135582607)

[**3** **Application Requirements** 5](#_Toc135582608)

[**3.1** **Hardware Components** 5](#_Toc135582609)

[**3.2** **Description of Hardware Components Used** 6](#_Toc135582610)

[**3.2.1** **NodeMCU ESP8266 Board** 6](#_Toc135582611)

[**3.2.2** **BMP180 - Pressure Sensor (Barometer)** 9](#_Toc135582612)

[**3.2.3** **DHT11 Sensor** 10](#_Toc135582613)

[**3.2.4** **Rain Sensor FC-37** 11](#_Toc135582614)

[**3.2.5** **Connecting Wires** 13](#_Toc135582615)

[**3.2.6** **Breadboard** 13](#_Toc135582616)

[**3.2.7** **System Architecture** 13](#_Toc135582617)

[**3.2.8** **Circuit Diagram** 14](#_Toc135582618)

[**3.2.9** **Implemented System** 14](#_Toc135582619)

[**3.2.10** **Workflow Diagram** 14](#_Toc135582620)

[**3.2.11** **Software Components – Tools and Technologies Used** 15](#_Toc135582621)

[**3.2.12** **Application Interface** 15](#_Toc135582622)

[**3.2.13** **ThingSpeak IoT Platform** 16](#_Toc135582623)

[**3.2.14** **Communication Protocols and Technologies Used** 17](#_Toc135582624)

[**3.2.15** **Proposed Topologies** 17](#_Toc135582625)

[**4** **Testing and Visualisation** 19](#_Toc135582626)

[**4.1** **Testing** 19](#_Toc135582627)

[**4.2** **Visualization** 19](#_Toc135582628)

[**4.2.1** **ThingSpeak Dashboard** 19](#_Toc135582629)

[**4.2.2** **Observations and Discussion** 20](#_Toc135582630)

[**4.2.3** **Graphical Representation** 21](#_Toc135582631)

[**4.3** **Data analysis using descriptive, diagnostic, and predictive analytics** 22](#_Toc135582632)

[**4.3.1** **Descriptive Analysis** 22](#_Toc135582633)

[**4.3.2** **Predictive Analysis** 22](#_Toc135582634)

[**4.3.3** **Prescriptive Analysis** 23](#_Toc135582635)

[**5** **Conclusion** 23](#_Toc135582636)

[**6** **GIT Repository** 23](#_Toc135582637)

[**7** **Source Code** 24](#_Toc135582638)

[**7.1** **WebServer Integeration** 24](#_Toc135582639)

[**7.1.1** **Index.h** 24](#_Toc135582640)

[**7.1.2** **Main.ino File** 26](#_Toc135582641)

[**7.1.3** **IoT Cloud Integeration** 29](#_Toc135582642)

[**8** **References** 32](#_Toc135582643)

# **Project Description**

## **Objectives:**

* Develop a weather station system that collects data on temperature, humidity, barometric pressure, and rainfall.
* Implement IoT connectivity to transmit sensor data to a central monitoring system in real-time.
* Analyze the collected weather data for insights, trends, and patterns.
* Visualize the data through a user-friendly interface, providing infographics and graphical representations.
* Optimize power consumption for efficient and sustainable operation.
* Facilitate team collaboration using GIT version control for seamless project development and updates.

## **Introduction**

The IoT-Based Weather Station project aims to leverage Internet of Things (IoT) technology to develop a system for monitoring and analyzing weather conditions. This project recognizes the significance of accurate and real-time weather data in various domains such as agriculture, transportation, disaster management, and urban planning. By implementing an IoT-based weather station, we can gather comprehensive weather information and enable data-driven decision-making.

The primary objective of a real-time weather station monitoring system is to provide up-to-date and accurate weather information. It enables users to monitor and understand current weather conditions in a particular location or region. This information is crucial for various applications such as agriculture, aviation, disaster management, and urban planning. The system typically consists of weather sensors that are strategically placed to capture weather data. These sensors are connected to a microcontroller or a gateway device, which collects the data and sends it to a centralized server or cloud platform using Internet of Things (IoT) technology. The cloud platform stores and processes the data, allowing users to access and analyze it remotely.

## **Expected Outcomes**

A real-time weather station monitoring system can provide several expected outcomes, including:

* **Accurate and Up-to-Date Weather Information:** The system collects weather data in real-time, ensuring that the information provided is current and accurate. Users can access real-time weather updates and make informed decisions based on the most recent data.
* **Enhanced Weather Forecasting:** By continuously monitoring weather conditions and collecting historical data, the system can improve the accuracy of weather forecasting. Advanced algorithms and analytics can be applied to the collected data to generate more precise and reliable weather predictions.
* **Early Warning and Disaster Management:** Real-time weather station monitoring systems can help in detecting and predicting severe weather events such as storms, hurricanes, or floods. By providing early warnings, emergency response teams and individuals can take necessary precautions and mitigate potential risks.
* **Optimization of Resource Allocation:** Access to real-time weather data enables various industries to optimize their resource allocation. For example, agriculture can adjust irrigation schedules based on rainfall data, energy providers can optimize energy generation based on weather conditions, and transportation companies can plan routes considering weather patterns.

## **Significance of Monitoring Weather Conditions Using IoT Technology**

Monitoring weather conditions using IoT technology offers numerous benefits and applications:

* Accurate Data Collection: IoT-based weather stations allow for precise and continuous monitoring of weather parameters, reducing reliance on manual data collection methods.
* Real-time Information: IoT connectivity enables the instantaneous transmission of weather data, providing timely and up-to-date information for decision-making.
* Decision Support Systems: Weather data collected through IoT can be used to develop decision support systems for agriculture, transportation, and emergency response, enhancing operational efficiency and safety.
* Resource Optimization: By understanding weather patterns, organizations can optimize resource allocation, such as irrigation in agriculture, energy management, and transportation planning.
* Early Warning Systems: IoT-based weather stations can contribute to the development of early warning systems for severe weather events, enabling timely alerts and mitigation measures.

# **Application Requirements**

The application requirements are divided into two main parts, given as:

1. Hardware Components
2. Software Software Components

## **Hardware Components**

Hardware part of the project consists of the following components, as shown.

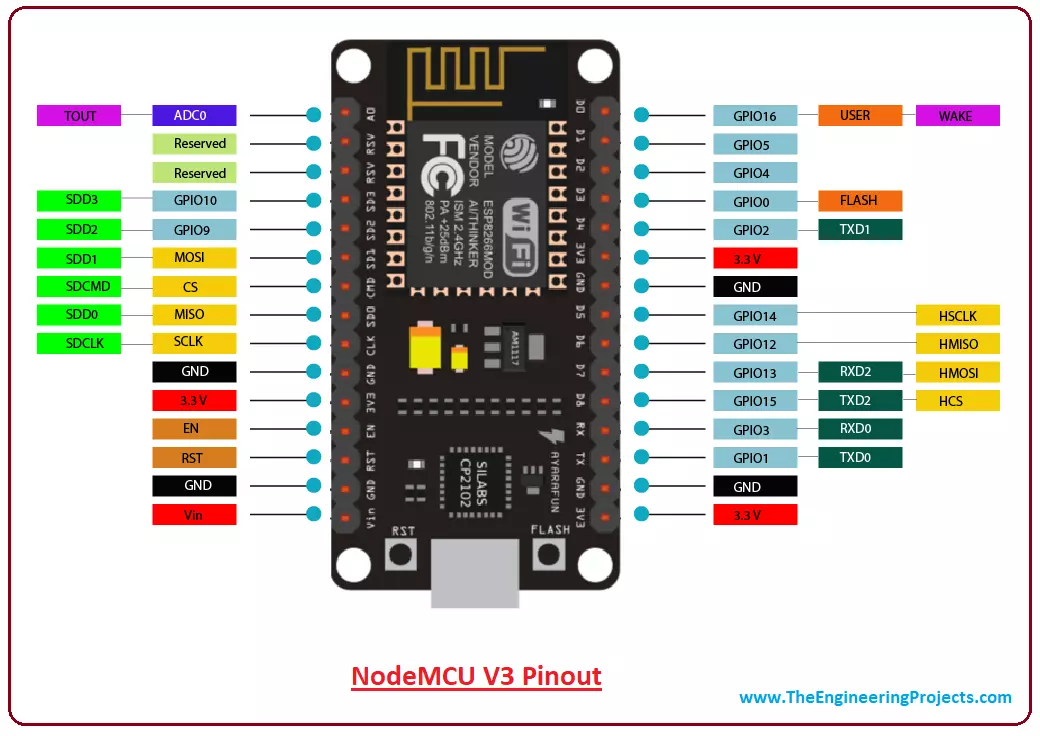
|  |  |  |  |
| --- | --- | --- | --- |
| **Hardware** | **Details** | **Hardware** | **Details** |
| NodeMCU ESP8266 Board |  | BMP180 Sensor |  |
| Rain Sensor FC-37 |  | DHT11 Sensor |  |
| Resister 4.7k |  | Connecting Wires |  |
| Breadboard |  |  |

## **Description of Hardware Components Used**

### **NodeMCU ESP8266 Board**

The objective of the NodeMCU ESP8266 board is to provide an affordable, versatile, and user-friendly platform for IoT development. It enables connectivity, sensor integration, and microcontroller functionality, making it an ideal choice for building IoT prototypes and applications. The NodeMCU (Node MicroController Unit) is an open-source software and hardware development environment built around an inexpensive System-on-a-Chip (SoC) called the ESP8266.

The NodeMCU ESP8266 board provides built-in Wi-Fi connectivity, allowing it to connect to the internet and communicate with other devices and services. This enables seamless integration with IoT ecosystems and cloud platforms. The ESP8266 chip used in the NodeMCU board is known for its low power consumption. This makes it suitable for battery-powered or energy-efficient IoT applications where power consumption is a critical factor. For this project we are using Official Amica NodeMCU. NodeMCU V3 comes with a number of GPIO Pins. Following figure shows the Pinout of the board Amica NodeMCU measures 49mm x 26mm with a standard pin space of 0.1″ between pins and 0.9″ between rows.



#### **Technical Specifications:**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Specification** | **Detail** |
|  | **Microcontroller** | ESP-8266 32-bit |
|  | **NodeMCU Model** | Amica |
|  | **NodeMCU Size** | 49mm x 26mm |
|  | **Pin Spacing** | **0.9" (22.86mm)** |
|  | **Clock Speed** | 80 MHz |
|  | USB**to Serial** | CP2102 |
|  | **USB Connector** | Micro USB |
|  | **Operating Voltage** | 3.3V |
|  | **Input Voltage** | 4.5V-10V |
|  | **Flash Memory/SRAM** | 4 MB / 64 KB |
|  | **Digital I/O Pins** | 11 |
|  | **Analog In Pins** | 1 |
|  | **ADC Range** | 0-3.3V |
|  | **UART/SPI/I2C** | 1 / 1 / 1 |
|  | **WiFi Built-In** | 802.11 b/g/n |
|  | **Temperature Range** | -40C - 125C |

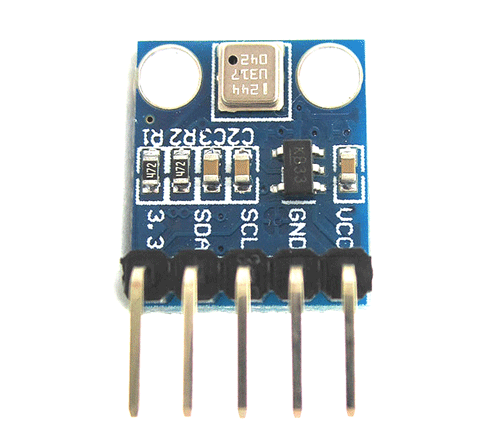
#### **Power Consumption & Optimization**

The detailed description of the power consumption of the NodeMCU ESP8266 and approaches for optimal power consumption are given as:

* **Operating Voltage:** Provide a stable 3.3V supply for proper functioning and reduced power consumption.
* **Sleep Modes:** Utilize Modem Sleep, Light Sleep, and Deep Sleep modes to significantly reduce power consumption during idle periods.
* **Clock Frequency:** Configure the clock frequency to balance power savings and desired performance.
* **Wi-Fi Operation:** Optimize Wi-Fi usage, minimize communication duration and frequency, and periodically enter deep sleep mode to reduce power consumption.
* **GPIO Pins:** Configure unused GPIO pins as inputs in low-power mode to minimize power consumption.
* **Power Supply Efficiency: Use** an efficient and stable power supply to minimize power losses and improve overall efficiency.
* **Code Optimization**: Minimize unnecessary operations and delays in the software code to reduce power consumption.
* **Power Management ICs:** Consider using external power management ICs for regulated and optimized power supply, voltage regulation, power sequencing, and power gating

### **BMP180 - Pressure Sensor (Barometer)**

The BMP180 sensor, manufactured by Bosch Sensortec, is a widely used digital barometric pressure and temperature sensor. It utilizes MEMS (Micro-Electro-Mechanical Systems) technology to provide precise pressure readings with up to 0.01 hPa resolution and temperature measurements with 0.1°C resolution. Operating via I2C communication, the sensor offers different measurement modes and incorporates built-in calibration data for accuracy. With its compact size, low power consumption, and high precision, the BMP180 sensor is popular in weather monitoring, and IoT applications requiring reliable barometric pressure and temperature measurements.



#### **Pins Detail**

|  |  |
| --- | --- |
| **Pin Name** | **Description** |
| VCC | Connected to +5V |
| GND | Connected to ground. |
| SDA | Serial Data pin (I2C interface) |
| SCL | Serial Clock pin (I2C interface) |
| 3.3V | 5 V or 3.3 C pin |

#### **Technical Specifications**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Component** | **Specs** |
|  | Operating  voltage of | 1.3V – 3.6V |
|  | Input voltage | :3.3V to 5.5V |
|  | Peak current | 1000uA |
|  | Consumes | 0.1uA standby |
|  | Maximum voltage at SDA , SCL : VCC + | 0.3V |
|  | Operating temperature | -40ºC to +80ºC |

#### **Power Consumption & Optimization:**

The detailed description of the power consumption of the BMP180 sensor and approaches for optimal power consumption:

* **Power Supply Voltage:** Use the lowest acceptable voltage within the range of 1.3V to 3.6V to optimize power consumption.
* **Measurement Modes**: Choose the appropriate mode (ultra-low power, standard, or high-resolution) to balance power consumption and measurement accuracy.
* **Measurement Frequency:** Adjust the frequency of measurements according to the application needs to reduce power consumption.
* **Sleep Mode:** Utilize the sleep or low-power modes of the microcontroller/system to put the sensor in a low-power state during idle periods.
* **I2C Communication:** Minimize duration and frequency of I2C communication to conserve power.
* **Power Management Techniques**: Implement strategies like turning off the sensor when not in use or adjusting power supply dynamically to optimize power consumption.
* **Calibration:** Properly calibrate the BMP180 sensor for accurate readings and reduced power usage.

### **DHT11 Sensor**

The DHT11 is a commonly used temperature and humidity sensor. The **DHT11 Temperature and Humidity Sensor**has a**temperature and humidity sensor complex**with a calibrated digital signal output. It offers high dependability and outstanding long-term stability by employing an innovative digital-signal-acquisition technique as well as temperature and humidity sensing technologies. This sensor consists of**a resistive-type humidity measurement component**and an**NTC temperature measurement component.**



**Vcc:** Need to provide +5V at this pinout.

**Data**: It's the digital output pin, gives either 0V or 5V.

**NC:** Not Connected. (It's left open for future design)

**GND:** Need to provide Ground at this pinout

#### **Technical Features**

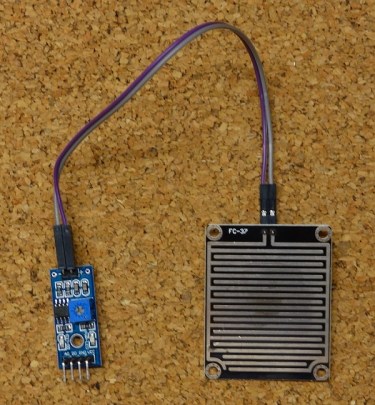
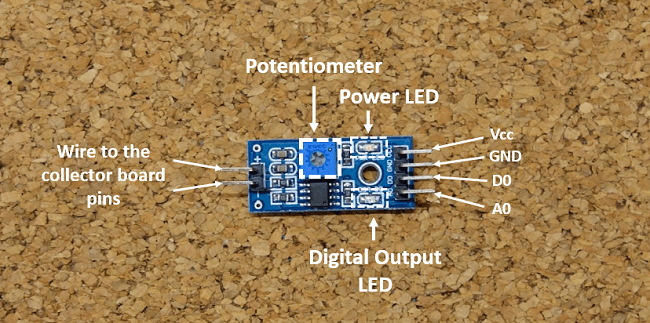
|  |  |  |
| --- | --- | --- |
| **S.No** | **Specs** | **Value** |
|  | Voltage range: | **3.5V** to**5.5V** |
|  | Current | 0.3mA (measuring) 60uA operating current (standby) |
|  | Output | Serail Data |
|  | Temperature Range | **0°C**to**50°C** |
|  | Humidity levels | Rranges from**20%** to**90%** |
|  | Temperature and humidity are both 16-bit resolutions | |
|  | Accuracy | 1°C and 1 per cent |

**Power Consumption & Optimization:** Approaches for optimal power consumption with the DHT11 sensor include:

* **Measurement Frequency:** Adjust measurement frequency to match application needs and minimize unnecessary measurements.
* **Sleep Mode:** Utilize sleep or low-power modes of the microcontroller/system to reduce power consumption during idle periods.
* **Wake-up and Measurement Synchronization**: Coordinate sensor wake-up and measurement cycles with system operation to minimize active state time.
* **Power Supply Optimization:** Ensure stable and efficient power supply to the DHT11 sensor, utilizing power management techniques.
* **Data Handling Efficiency:** Optimize data processing and minimize unnecessary computations or operations to reduce power consumption.

### **Rain Sensor FC-37**

The FC-37 rain sensor is set up by two pieces: the electronic board (at the left) and the collector board (at the right) that collects the water drops, as you can see in the following figure. The rain sensor has a built-in potentiometer for sensitivity adjustment of the digital output (D0). It also has a power LED that lights up when the sensor is turned on and a digital output LED.

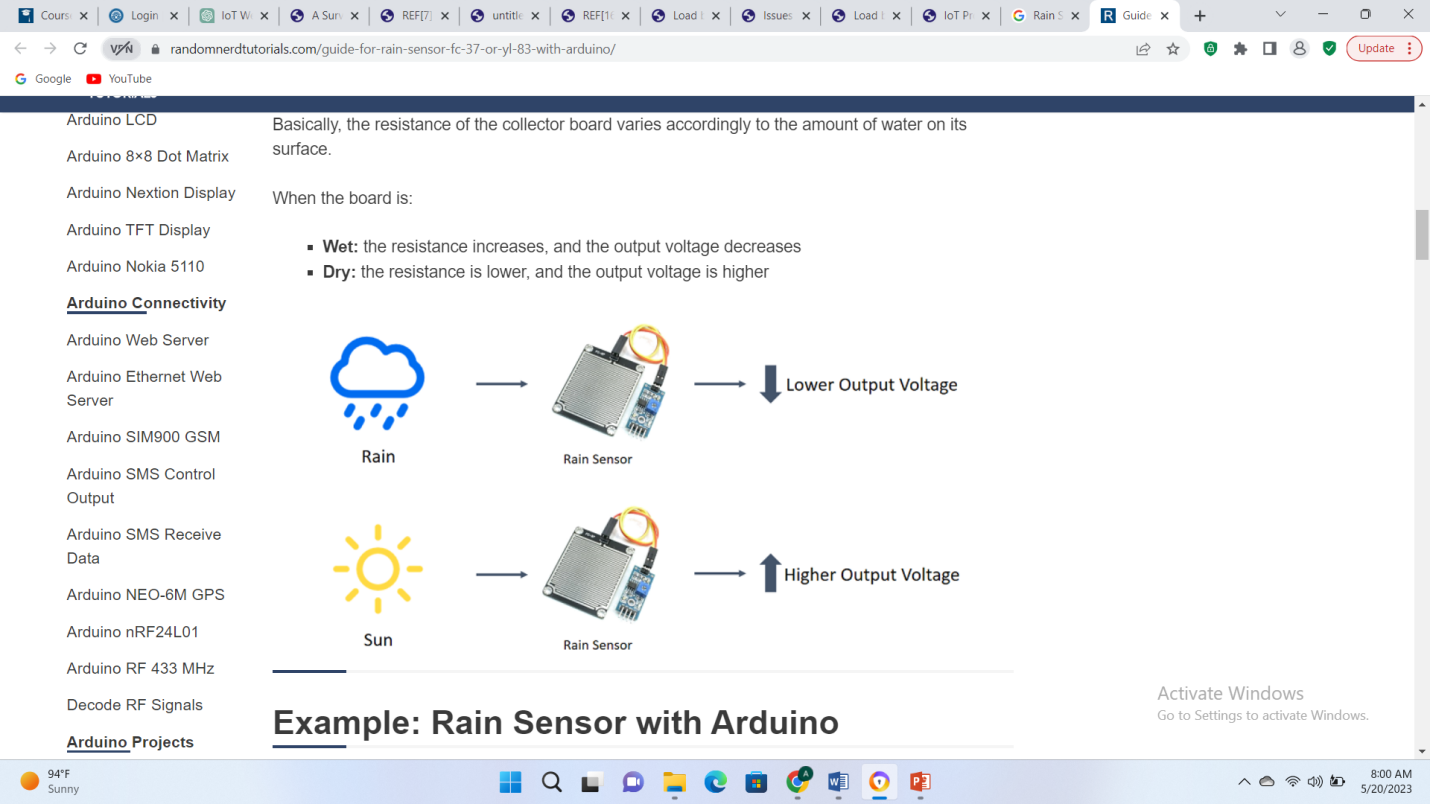


#### **Pin Details**

|  |  |
| --- | --- |
| **SENSOR PIN** | **CONNECTION** |
| A0 | Any Analog Pin |
| D0 | Digital Pins |
| GND | Ground of board |
| VCC | 5V (Regulated Type) |

The resistance of the collector board varies accordingly to the amount of water on its surface. When the board is:

* **Wet**: the resistance increases, and the output voltage decreases
* **Dry:** the resistance is lower, and the output voltage is higher



#### **Technical specifications**

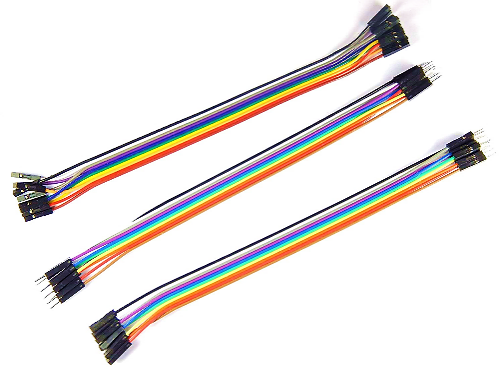
|  |  |  |
| --- | --- | --- |
| **S No** | **Component** | **Specs** |
|  | Operating Voltage | 5V |
|  | Output | Digital and Analog |
|  | Sensitivity | Adjustable |
|  | Output LED Indication |  |
|  | Bolt Holes | For easy installation |
|  | TTL Compatible |  |

#### **Power Consumption & Optimization:**

The detailed description of the power consumption of the FC-37 Rain Sensor and approaches for optimal power consumption:

* **Power Supply Voltage:** Provide a stable and regulated 5V power supply to ensure proper sensor operation.
* **Operating Current:** The sensor typically operates at low currents, ranging from mill amperes to tens of mill amperes during active sensing.
* **Sensing Duration**: Minimize the sensing duration to conserve power, especially if continuous monitoring is not required.
* **Trigger Mechanism:** Use a trigger mechanism to activate the sensor only when necessary, based on weather conditions or external triggers, reducing overall power consumption.
* **Power Management:** Implement power management techniques such as powering off the sensor when not in use, utilizing sleep modes, or using external power management.

### **Connecting Wires**

Jumper wires are a type of wire used in electronics and prototyping to connect components on a breadboard or other circuit board. They are typically made from flexible stranded wire with a plastic insulation and have pre-stripped ends that can be easily inserted into breadboard or connector pins.

### **Breadboard**

A breadboard (also known as a prototyping board) is a tool used for creating temporary electronic circuits by inserting and connecting components without the need for soldering.



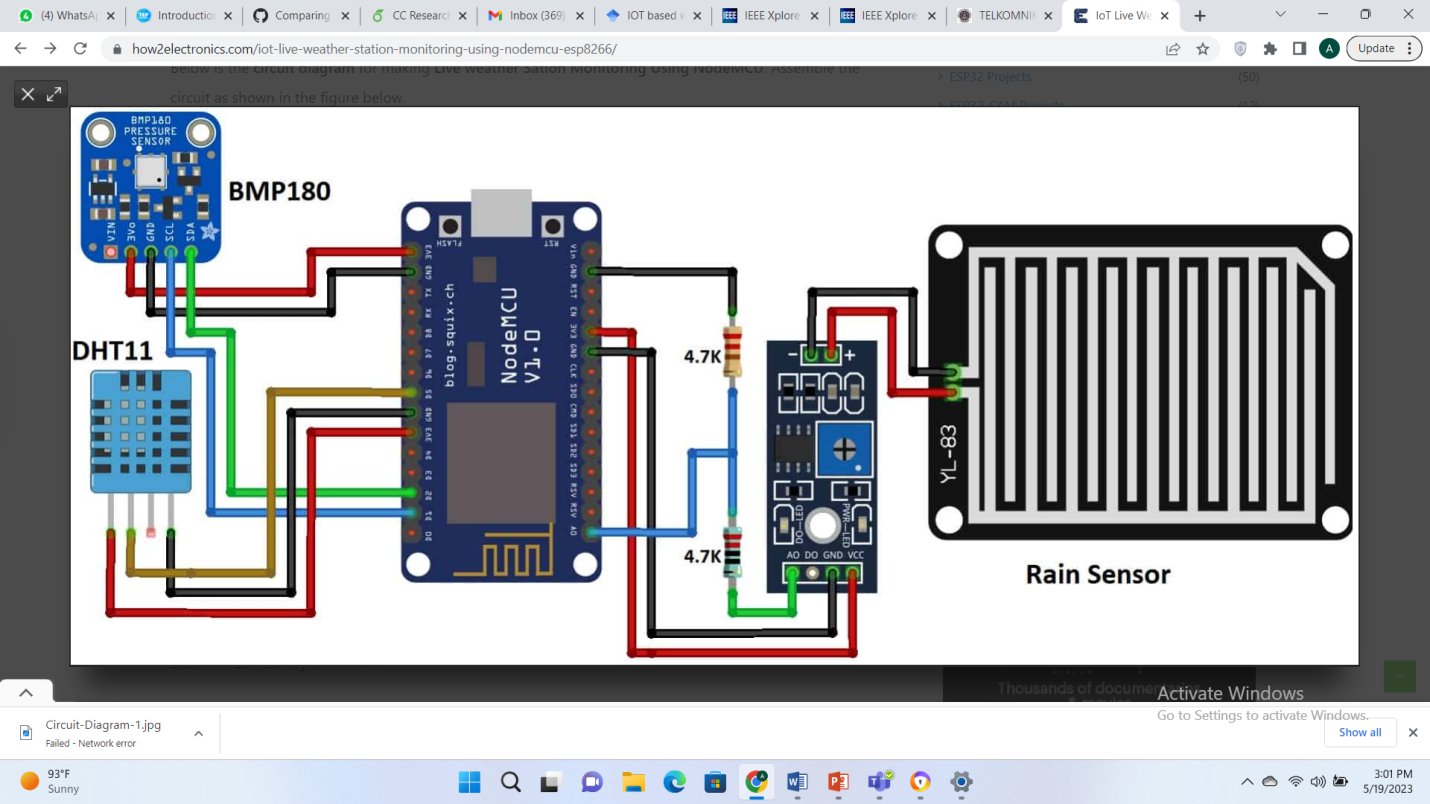
### **System Architecture**

The system typically consists of various components as discussed above and it works in layers, together to collect, transmit, process, and analyze weather data. The sensors are deployed to capture real time date of weather. The interface is web based dashboard, which shows the reading in real time. System is connected to ThingSpeak server for analysis of the gathered data, which is shown in graphical format.

The NodeMCU ESP8266 acts as the brains of the system, providing the processing power and connectivity, necessary to control the temperature, pressure and rain sensor. It is responsible for controlling the connection to the application, and connecting to the cloud based ThingSpeak server.ThingSpeak is an IoT analytics platform service that allows you to aggregate, visualize, and analyze live data streams in the cloud.

After Interfacing the sensors with nodemcu both DHT11 and BMP180 are connected directly without using any external resistance but for rain sensor we are using 4.7K ohm resistors. The Wi-Fi connection in the above project is established using the ESP8266 Wi-Fi module. The ESP8266 module is connected via hardware serial communication, allowing the board to connect to a wireless network and transmit data over the internet.

### **Circuit Diagram**



### **Implemented System**



### **Workflow Diagram**

GUI for real time data visualization and analysis, using webserver

Upload the code to NodeMCU via Aurdino IDE

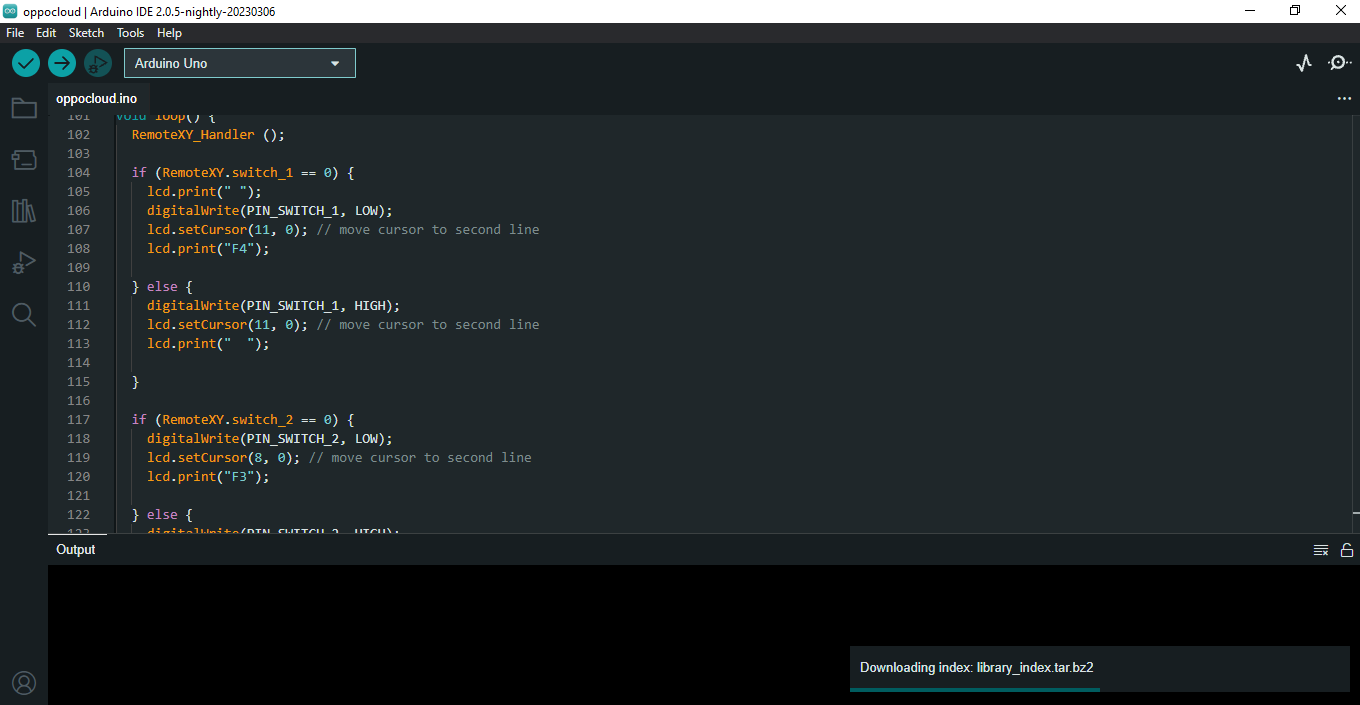
Hardware including NodeMcu and other sensors

Live weather monitoring parameters using Thingspeak

### **Software Components – Tools and Technologies Used**

#### **Arduino IDE**

The Arduino IDE (Integrated Development Environment) is a software application used for programming and developing applications for Arduino boards and compatible microcontrollers. It provides an easy-to-use interface and a set of tools for writing, compiling, and uploading code to Arduino boards.

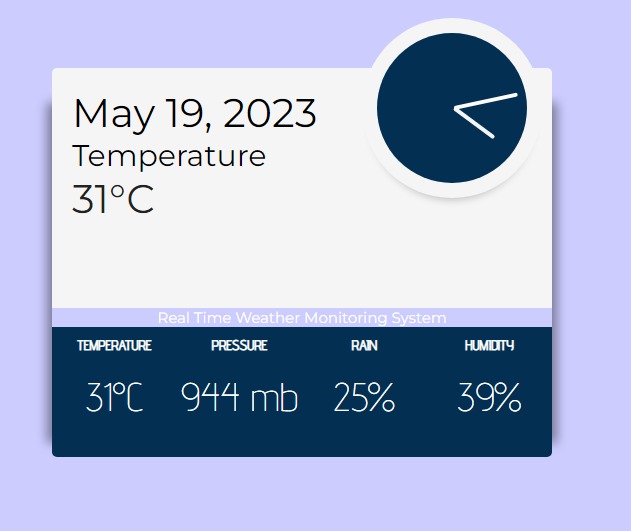


### **Application Interface**

To analyze the real time monitoring of weather station based on sensors, we have created a application Interface. We have developed GUI by following steps:

* **Set up a Web Server:** Start by setting up a web server on NodeMCU ESP8266 board. We can use libraries like ESP8266WebServer to handle web server functionalities.
* **Create HTML/CSS/JavaScript Files(index.h):** Design and create the HTML, CSS, and JavaScript files that help to make up GUI. HTML defines the structure of the web page, CSS handles the styling, and JavaScript adds interactivity.
* **Serve Web Pages**: Store the HTML, CSS, and JavaScript files on the NodeMCU ESP8266 board and serve them via the web server.
* **Connect to Hardware:** Interface the GUI with the hardware components of project. For example, if you have sensors or actuators connected to your NodeMCU board, you can use JavaScript to send AJAX requests to the server and update the GUI with real-time data or control the hardware based on user interactions.

After compilation, once the code is uploaded to the NodeMCU, the serial monitor gives the ip address of the web browser. After going to any browser and entering that specific ip address, it directs us to our application Interface page where we can observe the changes in sensors readings in real time for better decision making. It is shown below:

****

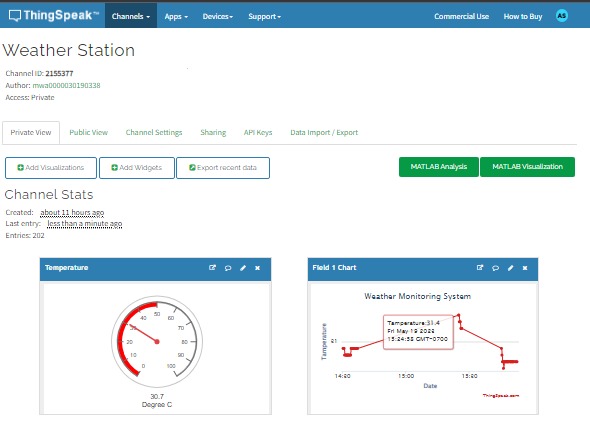
This interface provides the the weather monitoring necessary parameters including temperature, pressure, rain and humidity based on current time and date.

### **ThingSpeak IoT Platform**

ThingSpeak is a versatile IoT platform that simplifies the process of collecting, analyzing, and visualizing sensor data. It is widely used in IoT projects for monitoring, data logging, analytics, and automation. The Key features of ThingSpeak include:

* **Data Collection:** ThingSpeak allows us to collect data from various sources, such as sensors, devices, and external APIs. You can send data to ThingSpeak using HTTP, MQTT, or other protocols.
* **Channel and Field Management:** In ThingSpeak, we can create channels to organize our data. Each channel can have multiple fields representing different data parameters. we can define metadata for each field, such as data type, units, and descriptions.
* **Data Visualization:** ThingSpeak provides built-in tools to visualize our data in the form of charts, gauges, and maps. We can configure the visualizations to display real-time or historical data and customize the appearance and layout.

We have created a new channel named, *Weather Station* and in the next step, a dashboard is created including four gauges with their respective ranges. In order to visualise the results in real time, we have plotted graphs based on the live data coming from the sensors, as well.



### **Communication Protocols and Technologies Used**

The communication technology used for a web server typically involves the Hypertext Transfer Protocol (HTTP) as the primary protocol for data transmission between the server and client devices (web browsers). HTTP is an application-layer protocol that enables the transfer of various types of data, including web pages, images, videos, and other resources, over the internet. When a client device (web browser) wants to retrieve a web page or resource from the server, it sends an HTTP request. The request includes a URL (Uniform Resource Locator) specifying the desired resource and other optional parameters. Upon receiving an HTTP request, the web server processes it and generates an HTTP response. The response contains the requested resource or information about the requested action (e.g., status codes, headers, and content).

ThingSpeak, being a cloud-based IoT platform, utilizes various communication technologies to facilitate data exchange between connected devices and the platform itself. The primary communication technology used by ThingSpeak is the Hypertext Transfer Protocol (HTTP), which enables the transfer of data over the internet. ThingSpeak also supports the Message Queuing Telemetry Transport (MQTT) protocol, which is a lightweight publish-subscribe messaging protocol for constrained devices and low-bandwidth networks. MQTT provides efficient and

reliable communication between IoT devices and the ThingSpeak platform, allowing for real-time data streaming and control.

### **Proposed Topologies**

When scaling up the IoT-based Weather Station system, different network topologies can be considered based on the requirements and constraints of the deployment environment. Here are a few possible network topologies:

#### **Star Topology**

* Devices connected directly to a central hub or gateway device.
* Centralized control and management system.
* Easier monitoring and control of the weather station network.
* Requires longer cable lengths or wireless connectivity for devices located far from the hub.

#### **Mesh Topology**

* All devices interconnected, providing multiple paths for data transmission.
* Robustness and redundancy, as data can be routed through alternative paths if one device fails.
* Suitable for large-scale deployments and scattered environments.
* Ensures reliable communication but may require more complex routing algorithms.

#### **Hybrid Topology**

* Combination of different topologies to meet specific requirements.

For example, a central hub connecting devices in a star configuration, while devices interconnect in a mesh configuration.

* Provides benefits of both centralized control and robust communication paths.
* Customizable and adaptable to specific deployment needs.

#### **Cellular or Internet-based Topology**

* Devices connect to the internet via cellular networks or other internet connectivity options.
* Data transmission to a central server or cloud platform for processing and analysis.
* Enables remote monitoring and management of the weather station system.
* Suitable for wide geographical deployments but requires internet connectivity and may involve additional costs.

# **Testing and Visualisation**

## **Testing**

We have tested our Project in real conditions with following methodology:

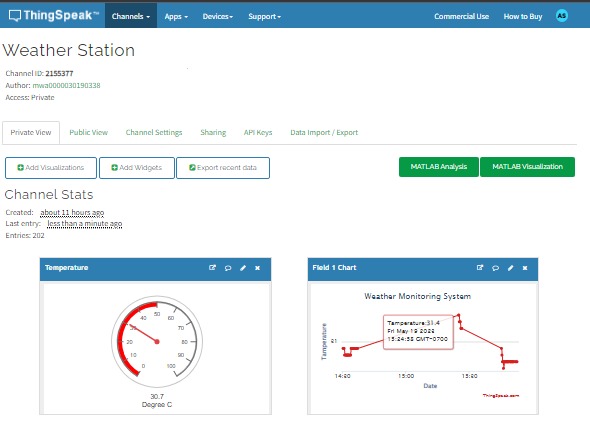
* Hardware Testing: The hardware components of the system, including the NodeMCU and various sensors and breadboard tested for functionality and proper connections. This includes verifying that they are powered on and connected to the appropriate modules and components.
* Reliability Testing: The system is tested over same time for a number of days to ensure that it is reliable and functioning. The system was tested in open air, closed quarters and at day and night time. It was ensured that with change in weather conditions the readings also change.
* Energy Efficiency Testing: The system is also tested for energy efficiency to ensure that it is saving energy and reducing electricity consumption. The idea behind is that energy efficiency is basically a design problem, and efficiency in usage of energy can be achieved by fine tuning the system scheduling and transferring filtered data.
* Sometimes some parts of the system went unresponsive due to poor contact on the breadboard. If the ESP8266 becomes unresponsive out of a sudden, first thing we suggest doing would be to check the connections once again.
* If you see a lot of garbage in the Serial Monitor console make sure the baud rate in the console is set accordingly with the ESP8266 settings.

## **Visualization**

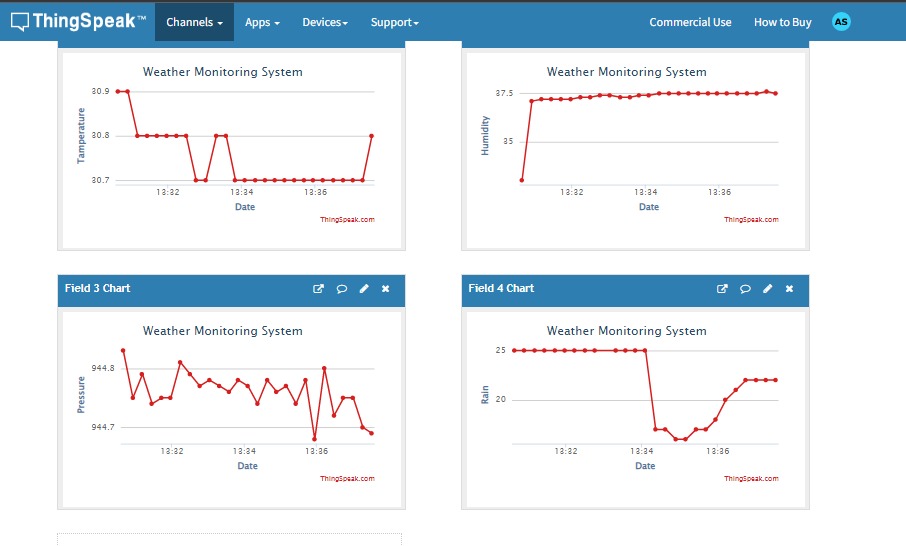
The results and analysis of the gathered data will be discussed in this section.

### **ThingSpeak Dashboard**

After the integeration of the weather station hardware module with ThingSpeak, the analysis is performed on the data as shown below:



After the connection, data read by the sensors was fed into the ThingSpeak application to generate the analysis charts, as shows below.



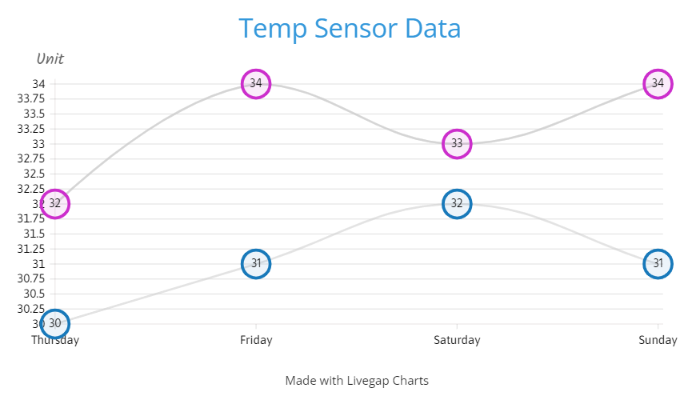
### **Observations and Discussion**

Data values were gathered from the proposed weather system. The results were obtained from the work station located in NUST university IoT lab for three days and in Hostel area. The temperature, humidity, pressure and rain readings were taken at 0700 hrs, 1600 hrs and 1900 hrs for three consecutive days. While the proposed system shows weather details for particular time and location, for analysis purposes the same data was compared with accuweather.com on those particular days. The difference in temperature is about 2 - 3 C, the difference in humidity is about 4 -5 %, no difference in rain status, and the difference in pressure is about 5 /10 MB.

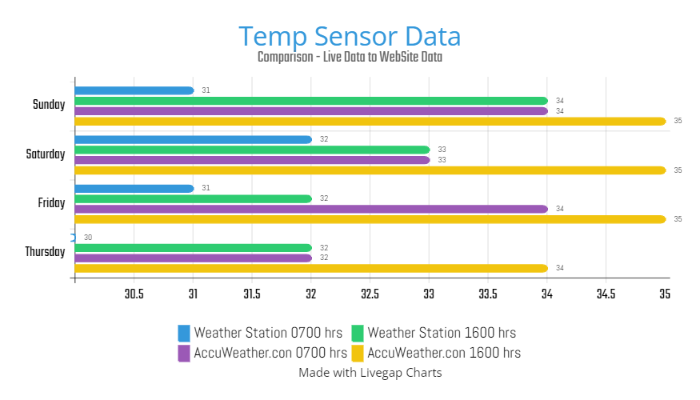
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Proposed Station | Accuweather | Proposed Station | Accuweather | Proposed Station | Accuweather |
| Sensor | **Time** | **Thurs** | | **Fri** | | **Sat** | |
| Temp | 0700 hrs | 30 C | 32 C | 30 C | 31 C | 31 C | 33 C |
|  | 1600hrs | 32 C | 28 C | 33 C | 35 C | 38 C | 40C |
| Pressure | 0700 hrs | 998 mb | 1013 mb | 989 mb | 1013 mb | 987 mb | 1001 mb |
|  | 1600 hrs | 985 mb | 1009 mb | 976 mb | 1009 Mb | 936 mb | 944 mb |
| Humidity | 0700 hrs | 39% | 33% | 43% | 47% | 27% | 30% |
|  | 1600hrs | 40% | 36% | 38% | 40% | 10% | 13% |
| Rain | 0700 hrs | 18% | 8% | 15% | 10% | 10% | 5% |
|  | 1600hrs | 20% | 15% | 16% | 13% | 6% | 2% |

### **Graphical Representation**

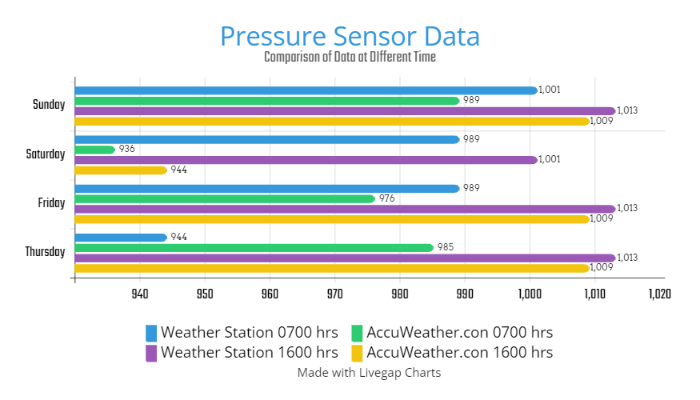
The gathered data can be stored and further used for descriptive analysis. By plotting the data on graphs, it shows us a trend of temperature, currently for three days with a 12 Hr gap. However, it can be plotted with more details to give trend in temperature of weather and predict rain, pressure and humidity for future planning allowing meteorologists and individuals to monitor and analyze the current weather conditions accurately.

****

This data is crucial for understanding how comfortable or warm or humid the environment is and can also be used to assess the likelihood of precipitation.

****

The figure shows the data collected by Live Weather Station (from pressure sensor) at Morning and Evening Time and compares it with data from Internet website AccuWeather at the same time for approx. same location. Changes in barometric pressure can indicate approaching weather systems or changes in weather conditions.

****

## **Data analysis using descriptive, diagnostic, and predictive analytics**

Descriptive, predictive, and prescriptive analysis can be applied to live weather station data in IoT to gain insights, make predictions, and provide actionable recommendations. Here's how each type of analysis can be utilized:

### **Descriptive Analysis**

Descriptive analysis focuses on summarizing and describing historical data. In the context of a live weather station, descriptive analysis involves examining past weather patterns, trends, and statistical characteristics of the collected data. This analysis provides an understanding of the current weather conditions, such as temperature ranges, humidity levels, wind patterns, and precipitation frequencies. Descriptive analysis can help in identifying long-term climate patterns, seasonal variations, and historical weather events.

### **Predictive Analysis**

Predictive analysis aims to forecast future weather conditions based on historical data patterns. By applying statistical and machine learning algorithms to the live weather station data, predictive analysis can generate predictions for variables such as temperature, rainfall, wind speed, and atmospheric pressure. These forecasts can range from short-term predictions, such as hourly or daily forecasts, to long-term forecasts that span weeks or months. Predictive analysis enables early warning systems, decision support for agriculture, and assists in planning outdoor activities.

### **Prescriptive Analysis**

Prescriptive analysis goes beyond predicting future weather conditions and provides actionable recommendations. It utilizes advanced algorithms and optimization techniques to suggest optimal actions based on predicted weather conditions. For example, prescriptive analysis can recommend adjusting irrigation schedules based on forecasted rainfall, optimizing energy consumption based on temperature predictions, or suggesting appropriate clothing based on expected temperature and humidity levels. Prescriptive analysis helps in making informed decisions and taking proactive measures based on anticipated weather conditions.

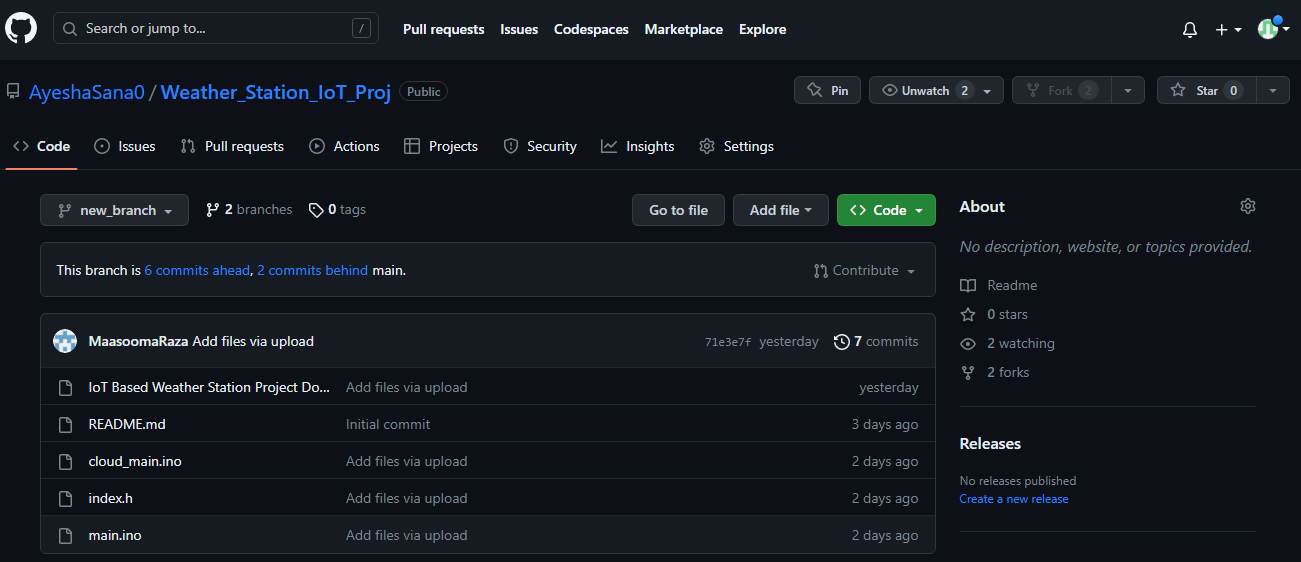
By combining descriptive, predictive, and prescriptive analysis, the live weather station in IoT can provide valuable insights into current weather patterns, enable accurate forecasts, and guide decision-making for various industries and applications, including agriculture, transportation, energy management, and disaster preparedness.

# **Conclusion**

To conclude, the IoT based weather station is an excellent example of how the Internet of Things (IoT) technology can be leveraged to improve energy efficiency, convenience, and overall user experience. By utilizing a network of interconnected sensors and devices, this project has achieved real-time data collection and analysis. The collected data provides valuable insights into the current weather conditions and serves as a foundation for accurate weather forecasting and prediction.

# **GIT Repository**

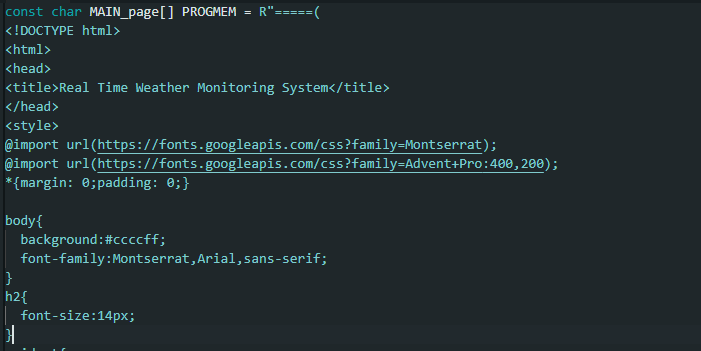
**URL Link:** <https://github.com/AyeshaSana0/Weather_Station_IoT_Proj/tree/new_branch>

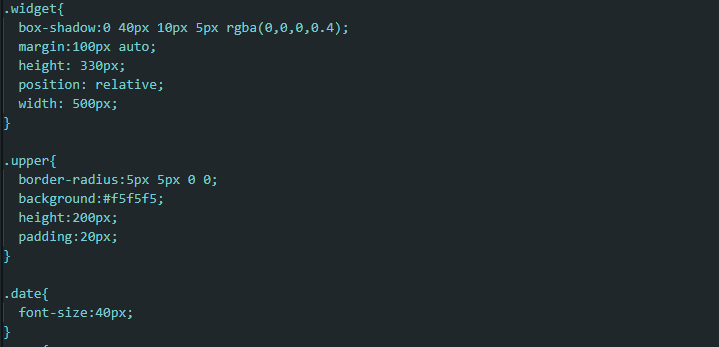


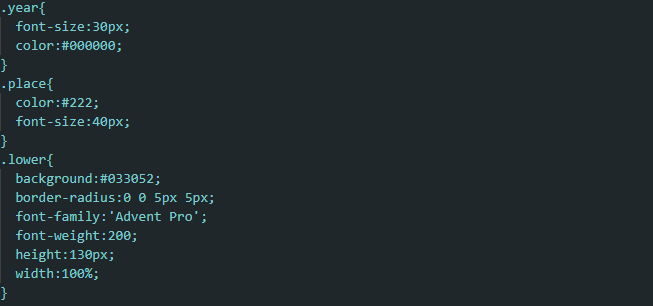
# **Source Code**

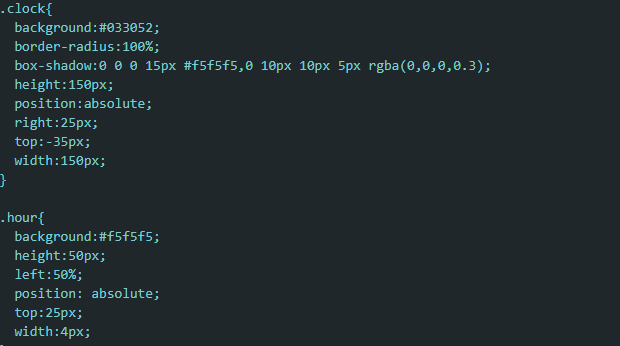
## **WebServer Integeration**

### **Index.h**

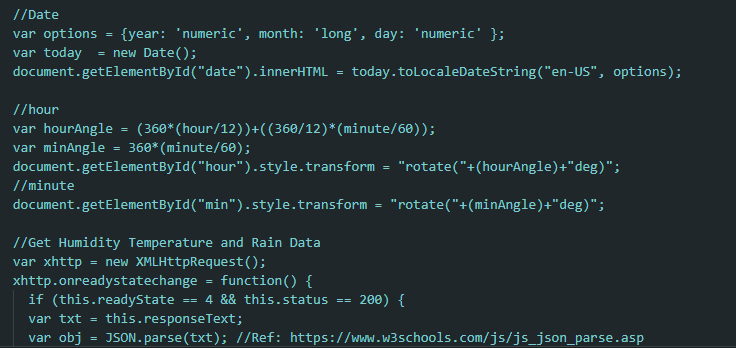


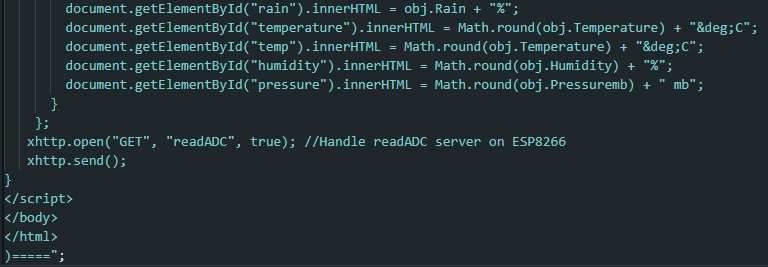




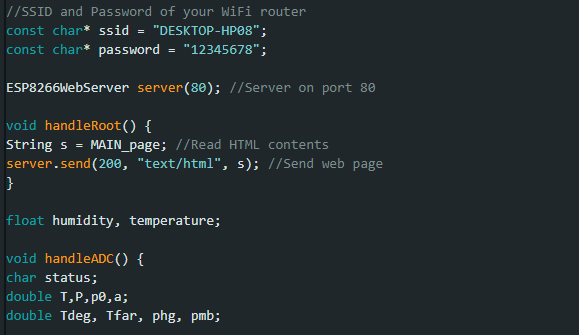


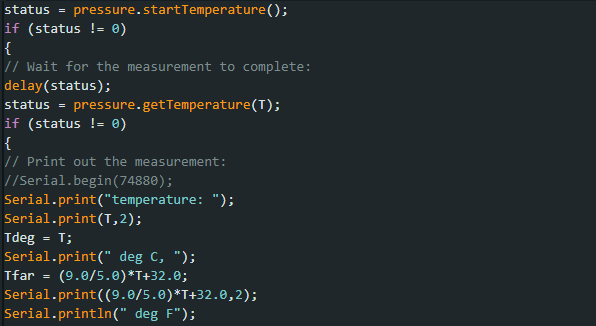


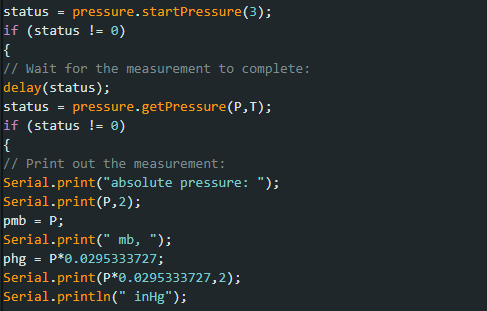


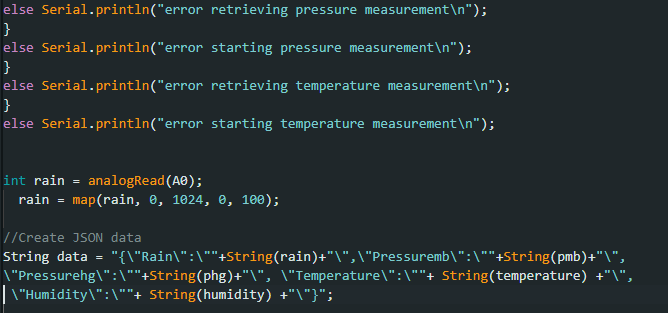


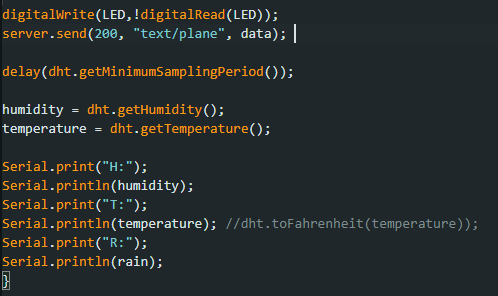
### **Main.ino File**



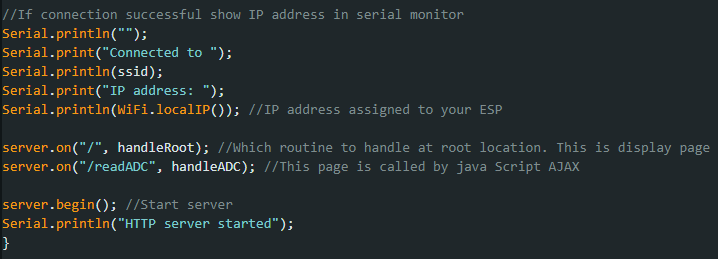
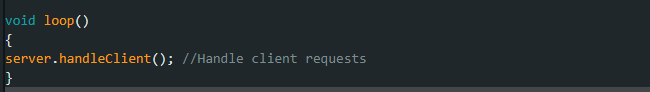






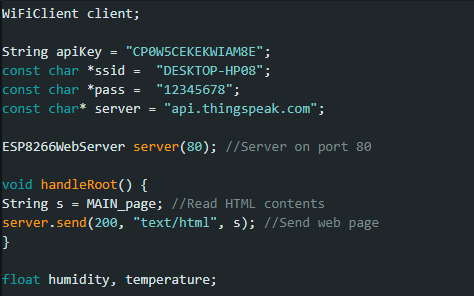




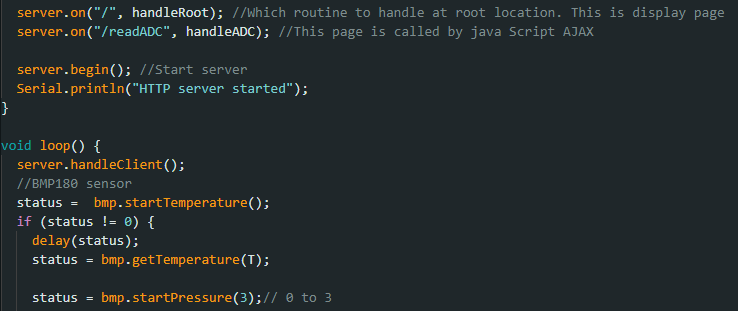


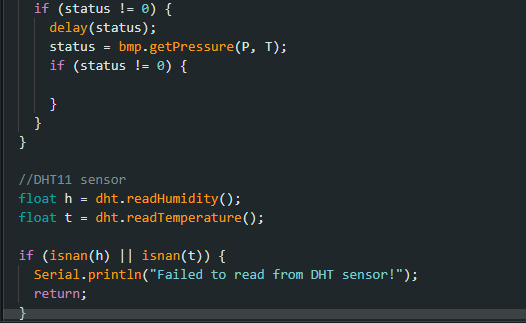
### **IoT Cloud Integeration**

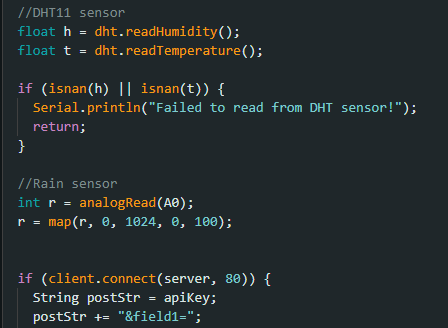
#### **Main.io File**

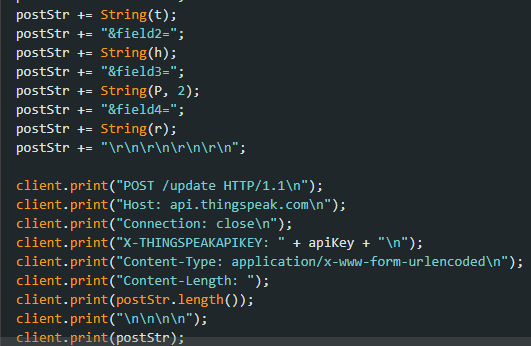


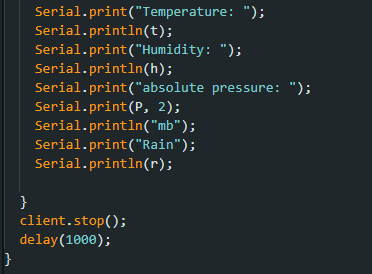












# **References**

* 2016 International Conference on Control, Instrumentation, Communication and Computational Technologies , IoT Based Weather Station, Ravi Kishore Kodali and Snehashish Mandal , National Institute of Technology, Warangal(ICCICCT)
* <http://telkomnika.uad.ac.id/index.php/TELKOMNIKA/article/viewFile/12784/7508>
* <https://how2electronics.com/iot-live-weather-station-monitoring-using-nodemcu-esp8266/>
* <https://components101.com/sensors/bmp180-atmospheric-pressure-sensor>
* <https://www.circuitschools.com/live-weather-station-using-nodemcu-esp8266-with-dashboard-iot/>
* <https://circuitdigest.com/microcontroller-projects/esp12-nodemcu-based-iot-weather-station>
* H. Saini, A. Thakur, S. Ahuja, N. Sabharwal, and N. Kumar, “Arduino based automatic wireless weather station with remote graphical application and alerts,” in 2016 3rd International Conference on Signal Processing and Integrated Networks (SPIN), Feb 2016, pp. 605–609.