IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Iot Based Weather Monitoring System

Mrs. Ashwini Kuradagi[1], Darshan Byadagi[2], Souvik Majumdar[3], Suraj Mahadik[4] and Chaitanyagouda Patil [5]

[1] Asst.Prof Department of Electronics and Communications, Rural Engineering College, Hulkoti, Karnataka

[2,3,4,5] Department of Electronics and Communications, Rural Engineering College, Hulkoti, Karnataka,

ABSTRACT:

The proposed system offers an efficient solution for real-time weather monitoring in the Karnataka region by leveraging IoT technology. It utilizes multiple sensors to collect data on key environmental parameters such as temperature, humidity, rainfall, and atmospheric pressure, which is then processed by a NodeMCU controller. The system operates on a client-server architecture, enabling the transmission of sensor data via a serial monitor to a cloud platform, making the data publicly accessible. Users can monitor weather conditions remotely from any location, without reliance on specific applications or websites. The system demonstrates improved accuracy and reliability over traditional weather monitoring methods, providing valuable, real-time insights into environmental conditions for better decision-making and analysis.

***Index Terms* -** IoT, Weather Monitoring, Sensors, NodeMCU, Temperature, Humidity, Pressure, Rainfall, Cloud Computing.

I INTRODUCTION:

The Internet of Things (IoT) has emerged as a transformative technology, redefining traditional systems by introducing advanced methods for collecting, analyzing, and utilizing data in real time. According to Al-Fuqaha et al. (2015) [1], IoT creates a seamless network of interconnected devices capable of communicating with one another and making decisions based on real-time data. This paradigm shift has paved the way for smarter and more efficient systems across a wide range of applications. Among these, weather monitoring systems have gained significant attention due to their ability to provide accurate and timely environmental data, which is essential for tackling critical challenges related to climate change, disaster management, and resource optimization.

IoT-based weather monitoring systems utilize a combination of sensors to measure parameters such as temperature, humidity, atmospheric pressure, rainfall, and light intensity. These sensors, such as the DHT22 for temperature and humidity and the BMP280 for pressure, are integrated into a network that transmits data to centralized systems for processing and analysis. As Mishra and Singh (2021) [2] noted, this real-time data acquisition enables a proactive approach to weather forecasting and environmental monitoring. Unlike traditional weather systems, which often involve delayed updates and limited reach, IoT-powered solutions provide instantaneous insights, making them invaluable in critical scenarios such as storm warnings and flood predictions.

The impact of such systems extends to numerous sectors. In agriculture, precise weather data helps farmers determine the optimal times for planting, irrigation, and harvesting, ultimately increasing crop yields and reducing waste. Transportation networks benefit from IoT-based weather systems by improving traffic flow and minimizing disruptions caused by adverse weather conditions. Urban planning and disaster management also leverage these systems to create resilient infrastructure, mitigate risks, and enhance preparedness for natural calamities. Gupta and Pal (2020) [4] highlighted that the integration of IoT with existing weather monitoring frameworks allows for enhanced scalability, reliability, and coverage, ensuring that even remote areas can access accurate environmental data.

Moreover, the scalability and flexibility of IoT solutions make them adaptable to different environments and needs. Advanced communication protocols and cloud computing capabilities enable seamless data transfer and storage, while real-time analytics provide actionable insights. By fostering a deeper understanding of environmental patterns and trends, IoT weather monitoring systems are helping societies adapt to changing climatic conditions and make informed decisions. This evolution signifies not just technological advancement but also a critical step toward sustainable development and global resilience in the face of environmental uncertainties.

II. LITERATURE REVIEW

Mishra and Singh (2021) investigated the implementation of IoT-based weather monitoring systems, highlighting their ability to acquire and transmit data in real time. Their study emphasized the significance of these systems in accurately forecasting weather conditions and analyzing environmental data, which are crucial for applications like agriculture and disaster preparedness [2]. In a similar vein, Patil and Pawar (2020) explored the use of cost-effective IoT solutions for weather monitoring, emphasizing the integration of low-energy sensors and efficient data transmission methods to create scalable and adaptable systems suitable for various environments [3].

Gupta and Pal (2020) delved into the technological advancements in IoT-based weather monitoring systems, focusing on how modern sensors are integrated into pre-existing infrastructures to enhance data precision and reliability. They highlighted the capacity of IoT systems to extend coverage to remote and underdeveloped areas, ensuring broader accessibility to accurate environmental data [4]. Verma and Agrawal (2018) showcased the architecture of real-time IoT-enabled weather systems, providing insights into their data handling processes, including collection, processing, and analysis, which significantly contribute to their efficiency [8].

Chaudhary and Gupta (2020) emphasized the role of advanced sensors, such as the DHT22 and BMP280, in improving the precision of environmental data. They elaborated on how integrating these sensors into IoT systems elevates the overall monitoring accuracy, underlining the growing dependence on sophisticated sensor technologies in this domain [12]. Similarly, Kumar and Singh (2019) examined the development of energy-efficient weather monitoring systems using low-power sensors, emphasizing the need for solutions that balance energy consumption and system performance, making them sustainable for long-term deployment [9].

Reddy and Reddy (2022) further expanded on the capabilities of IoT weather monitoring systems by exploring data analytics and visualization techniques. Their study demonstrated how real-time data analysis can provide actionable insights, enabling better decision-making across sectors like urban planning and environmental management [7]. Additionally, Verma and Agrawal's (2018) work highlighted the robustness of IoT weather systems, showcasing how their design supports large-scale deployment and reliability [8].

Collectively, these studies illustrate the evolution of IoT in weather monitoring, emphasizing its transformative potential to address global challenges. From real-time data acquisition to energy-efficient design and advanced analytics, IoT systems continue to play a pivotal role in delivering accurate and actionable environmental insights. These systems are not only enhancing technological capabilities but are also creating sustainable solutions for diverse applications worldwide.

III. RESEARCH METHODOLOGY

The research methodology of this project builds upon a robust foundation of previous scholarly works and practical implementations to ensure the development of an efficient, reliable, and scalable IoT-based weather monitoring system. Kumar and Singh (2019) highlighted the advantages of utilizing low-power sensors in IoT systems, paving the way for energy-efficient designs that are crucial for long-term environmental monitoring [9]. Taking inspiration from this, the project integrates cutting-edge sensors like DHT22 for accurate temperature and humidity data [5], BMP280 for high-precision atmospheric pressure measurements [6], LDR for light intensity monitoring [10], and the Rain Sensor YL-83 for capturing real-time rainfall metrics [11]. These sensors are selected for their high sensitivity, low power consumption, and compatibility with IoT frameworks.

The use of the Arduino IDE for programming and sensor integration ensures seamless data acquisition and transmission. As per the comprehensive guide provided by Arduino [13], the IDE supports robust coding practices and offers real-time debugging tools, making it an ideal choice for managing complex sensor interactions. This methodology aligns with the recommendations of Chaudhary and Gupta (2020), who emphasized the importance of integrating advanced sensors and efficient data processing systems to enhance IoT-based weather monitoring capabilities [12].

To ensure real-time functionality, the design framework incorporates principles from Mishra and Singh (2021), who stressed the critical role of real-time data acquisition and communication in weather monitoring systems [2]. The system's architecture is designed to process data at the edge, reducing latency and ensuring immediate access to critical weather metrics. This real-time approach is vital for applications in agriculture, disaster management, and urban planning, where timely data can inform crucial decisions.

Additionally, insights from Patil and Pawar (2020) informed the project's focus on scalability and adaptability, ensuring that the system can accommodate additional sensors or expand to cover larger geographical areas with minimal modifications [3]. Their work underscored the importance of designing systems that are not only cost-effective but also versatile in handling diverse environmental variables.

Further inspiration is drawn from Verma and Agrawal (2018), who highlighted the need for robust data management and processing capabilities in IoT systems [8]. By leveraging cloud-based storage solutions and data analytics platforms, this project ensures that the collected data can be visualized, analyzed, and shared seamlessly, supporting advanced weather prediction models and long-term climate studies.
Energy efficiency, a critical aspect of IoT systems, is addressed by incorporating sustainable design practices as advocated by Reddy and Reddy (2022) [7]. Their research emphasized the need for optimizing power usage in IoT systems, especially in remote or off-grid locations. This project incorporates power management strategies such as dynamic sleep modes for sensors and efficient data transmission protocols to extend operational lifespans.

Lastly, the methodology considers the challenges of sensor calibration, data accuracy, and environmental variability. Drawing on the sensor-specific datasheets and recommendations [5, 6, 10, 11], the system includes automated calibration routines and error-checking algorithms to ensure consistent performance across varying environmental conditions. By synthesizing these diverse methodologies and innovations, the project delivers a comprehensive IoT-based weather monitoring system tailored to modern needs and future challenges.

IV. SYSTEM DESIGN AND ARCHITECTURE

The **IoT-based Weather Monitoring System** uses an array of sensors and the **NodeMCU ESP8266** to gather, process, and transmit data about key environmental factors such as temperature, humidity, light levels, pressure, and rainfall. The system ensures that data can be monitored both locally and remotely in real-time.

1. Architecture Breakdown

The system is composed of several layers that each serve a unique function:

- **Data Collection Layer**: Collects environmental data through various sensors.
- **Data Processing Layer**: The NodeMCU ESP8266 microcontroller processes the incoming data.
- **Data Communication Layer**: Sends processed data to a local interface or cloud platform.
- **User Interface Layer**: Displays data in a user-friendly format via dashboards or apps for easy access.

2. Core Components

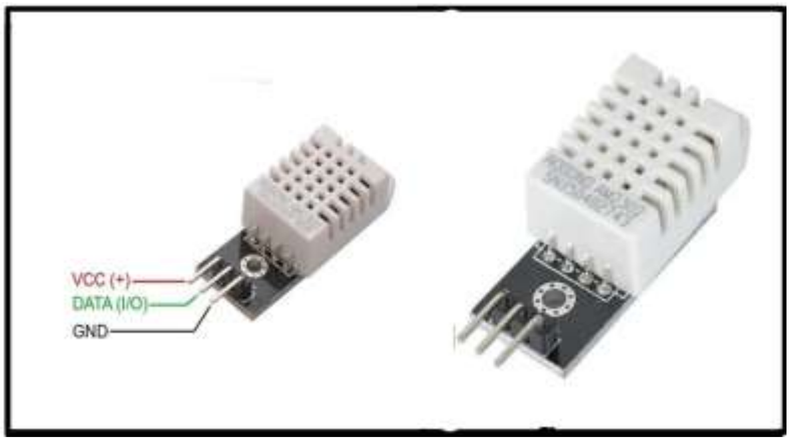
a. NodeMCU ESP8266 Microcontroller

- **Function**: Acts as the central processor for the system, interpreting sensor data and managing data transmission.
- **Key Features**:
  - **Built-in Wi-Fi**: Facilitates wireless communication and remote data access.
  - **Flexible Protocol Support**: Compatible with I2C, SPI, and UART for connecting to a range of sensors.
  - **Compact and Power-Efficient**: Designed for IoT applications where space and energy are limited.

- **Technical Specifications**:
  - Equipped with an 80 MHz 32-bit processor.
  - Supports GPIO pins for connecting sensors and devices.
  - Up to 4 MB of flash memory for program storage.
- **Operation**:
  - Receives and processes data from various sensors and then transmits it to local displays or cloud platforms.

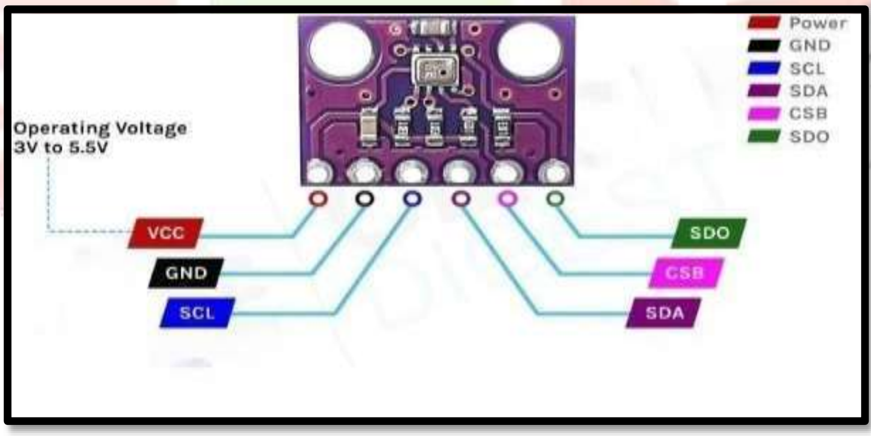
- Optimized for low energy usage, enabling long-duration operation in remote setups.

b. DHT22 Sensor (Temperature and Humidity)



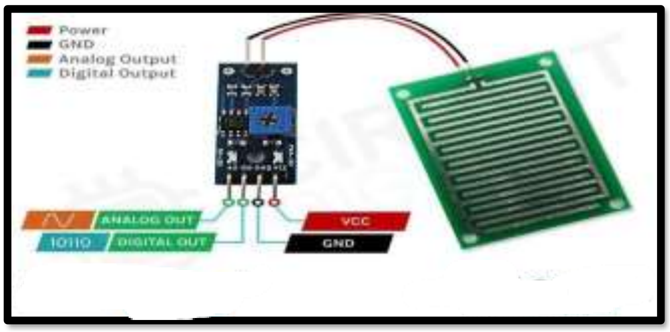
- **Purpose**: Measures temperature and humidity in the environment with high accuracy.
- **Specifications**:
  - Temperature: -40°C to 80°C, with a precision of ± 0.5°C.
  - Humidity: 0% to 100%, with an accuracy of ±2%.
- **Functionality**:
  - Outputs digital data representing temperature and humidity readings.
  - Requires a pull-up resistor for stable data transmission

c. BMP280 Sensor (Atmospheric Pressure)



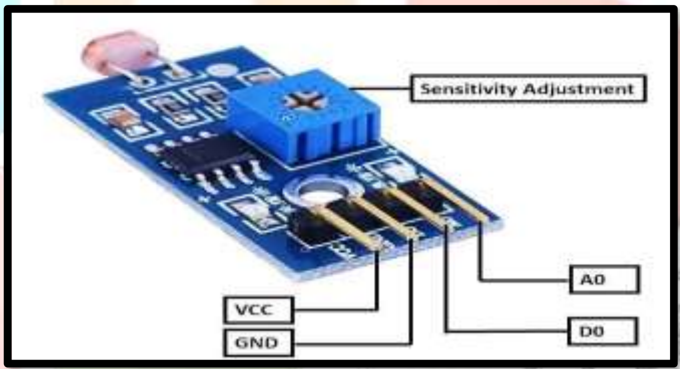
- **Purpose**: Measures atmospheric pressure and temperature, important for weather-related applications.
- **Specifications**:
  - Pressure: 300–1100 hPa with an accuracy of ±1 hPa.
  - Temperature: -40°C to 85°C with an accuracy of ±1°C.
- **Functionality**:
  - Communicates with the NodeMCU through I2C or SPI.

d. Rain Sensor (Precipitation Detection)



- **Purpose**: Detects the presence of rain and its intensity.
- **Specifications**:
  - Provides both analogy and digital signals.
  - Sensitivity is adjustable to different levels of rainfall.
- **How It Works**:
  - Measures changes in electrical resistance as rainwater interacts with the sensor's surface.
  - This change in resistance generates an electrical signal corresponding to the amount of rainfall.

e. LDR (Light Dependent Resistor)

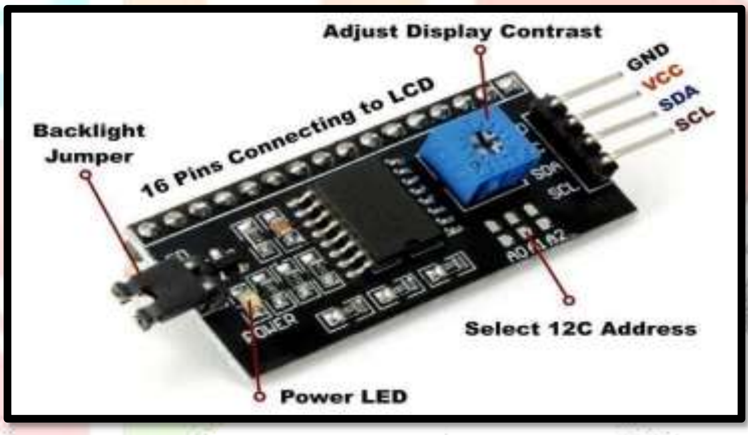


- **Purpose**: Monitors the level of ambient light to detect the time of day or light conditions.
- **Specifications**:
  - Highly sensitive to light intensity.
  - Outputs an analog
  - signal based on light levels.
- **How It Works**:
  - The resistance of the LDR decreases with increased light intensity, producing a measurable output that is processed by the NodeMCU.

f. LCD Display (16x2 with I2C Module)

- **Purpose**: Provides a visual representation of sensor data for local monitoring.
- **Specifications**:
  - Display capability: 16 characters per row on 2 rows.
  - Uses I2C communication to simplify wiring and reduce GPIO pin usage.
- **How It Works**:
  - Data is transmitted from the NodeMCU to the LCD, displaying environmental information like temperature, humidity, and rainfall in real time.

g. Pinout of I2C Module



Purpose: The I2C module simplifies communication between the **NodeMCU** and peripherals like an LCD display, reducing the number of required connections.

Pinout:

1. **VCC**: Supplies power to the module (usually 5V or 3.3V depending on the device).
2. **GND**: Ground pin, connected to the system ground.
3. **SDA (Serial Data)**: Transmits data between devices.
4. **SCL (Serial Clock)**: Carries the clock signal for synchronization.

V. HARDWARE IMPLEMENTATION

Integration of Sensors with NodeMCU ESP8266

The NodeMCU ESP8266 is the primary microcontroller, which is responsible for collecting data from a variety of environmental sensors. The following explains the connection and integration of each sensor:

- **DHT22 (Temperature and Humidity Sensor):**
    - Connection: The data output from the DHT22 is routed to a GPIO pin (e.g., GPIO2) on the NodeMCU.
    - Power Source: The sensor is powered using the 3.3V pin of the NodeMCU.
- **BMP280 (Pressure and Temperature Sensor):**
    - Connection: The SDA and SCL pins of the BMP280 are connected to GPIO4 (SDA) and GPIO5 (SCL) pins on the NodeMCU, using I2C communication.
    - Power Source: This sensor operates at 3.3V, powered directly by the NodeMCU.
- **Rain Sensor:**
    - Connection: The digital output pin is linked to a GPIO pin (e.g., GPIO12) on the NodeMCU. The analog output is connected to the ADC pin for measuring rainfall intensity.
    - Power Source: The rain sensor is powered by the 5V pin of the NodeMCU.
- **LDR (Light Dependent Resistor):**
    - Connection: The LDR is part of a voltage divider circuit. The output from this circuit is fed to the ADC pin on the NodeMCU to measure light intensity.
    - Power Source: The LDR is powered from the 3.3V pin on the NodeMCU.
- **LCD Display (16x2 with I2C Module):**
    - Connection: The SDA and SCL pins of the LCD are connected to GPIO4 (SDA) and GPIO5 (SCL) on the NodeMCU, allowing for I2C communication.
    - Power Source: The LCD module receives 5V from the NodeMCU's 5V pin.

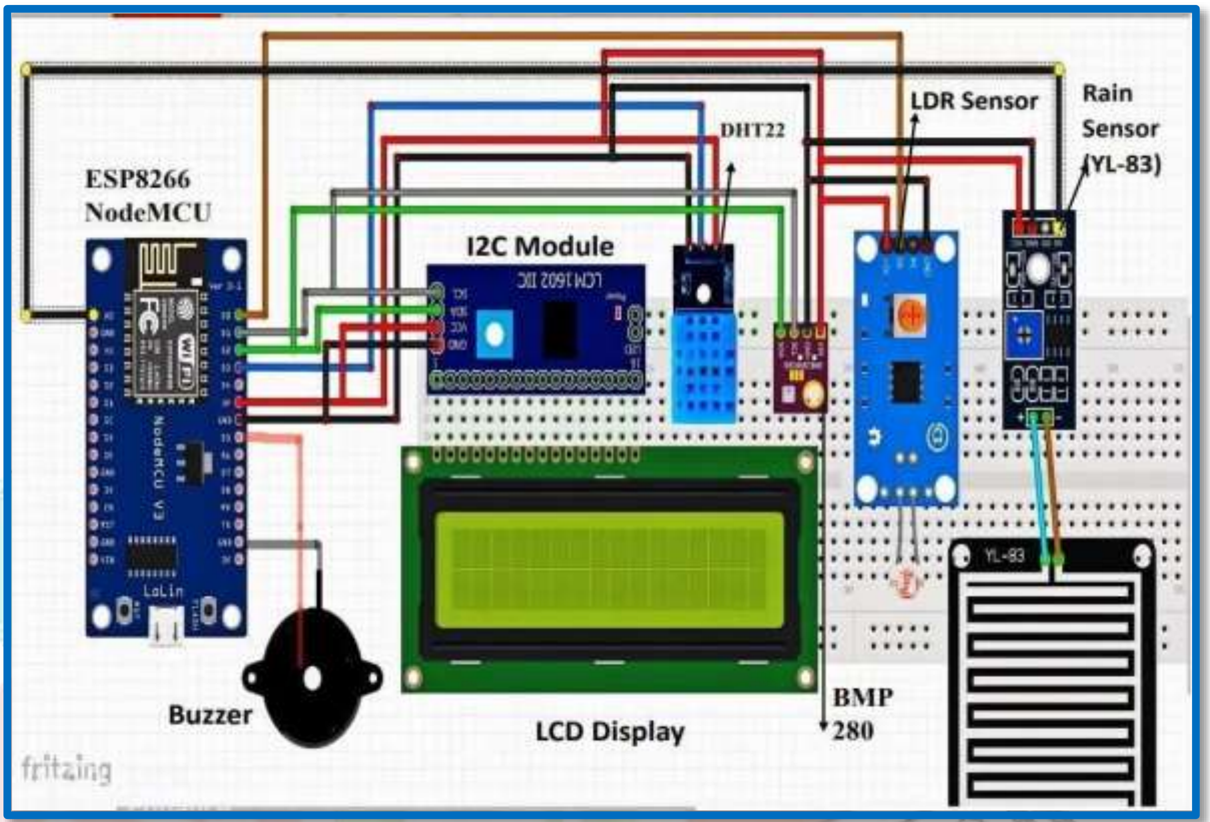
SENSOR	VCC	GND	DATA/SDA/SCL	ANALOG INPUT
DHT22	3.3	GND	GPIO2	-
BMP280	3.3V	GND	GPIO4 (SDA), GPIO5 (SCL)	-
Rain Sensor	5V	GND	GPIO12	ADC pin
LDR	3.3 V	GND	-	ADC pin
LCD (16x2)	5V	GND	GPIO4 (SDA), GPIO5 (SCL)	-

Power Supply and Circuit Design

Power Supply:

- The NodeMCU is powered via a 5V USB connection, which is converted internally to 3.3V to power both the microcontroller and the sensors.
- Sensors receive power from the 3.3V or 5V pins on the NodeMCU, depending on their individual voltage needs.
- In battery-operated systems, a DC-DC converter can be used to maintain a stable voltage supply.

Circuit Design:



- **Grounding:** All components share a common GND pin, ensuring a uniform reference voltage across the system.
- **Signal Routing:** The I2C protocol is used for the BMP280 sensor and the LCD display, reducing the number of required data lines by utilizing just the SDA and SCL connections.
- **Sensor Integration:** The rain sensor provides digital and analog outputs, which are routed to the appropriate GPIO pins and ADC pin, respectively, to measure rainfall intensity.
- **Light Measurement:** The LDR is part of a voltage divider circuit, which outputs an analog signal proportional to light levels, and this signal is sent to the ADC pin for processing.

VI. SOFTWARE DEVELOPMENT

Use of Arduino IDE for Programming and Deployment:

The Arduino Integrated Development Environment (IDE) is a robust platform designed for creating and deploying code on microcontroller boards. Key features include:

- **Coding Interface**: Supports programming in C/C++, featuring tools like syntax highlighting, error detection, and auto-indentation to enhance code quality.
- **Extensive Library Support**: Provides built-in libraries that simplify interaction with various sensors and hardware modules, saving development time.
- **Program Upload**: Enables straightforward deployment of code to devices such as NodeMCU ESP8266 via a USB connection.
- **Debugging Tools**: Includes a Serial Monitor for tracking and analyzing sensor data during development, allowing real-time debugging and adjustments.

Code Modules for Sensor Integration and Data Visualization:

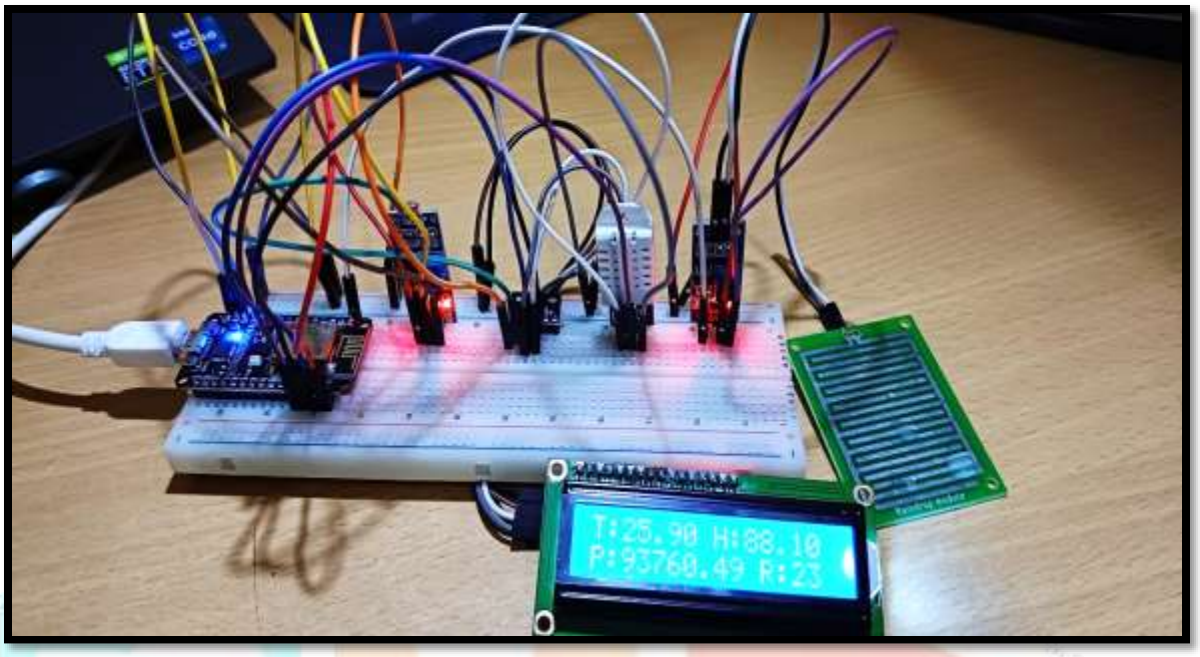
- **Sensor Integration**:
  - **Environmental Sensors**: Devices like DHT11 or DHT22 capture temperature and humidity data with high precision.
  - **Pressure Measurement**: The BMP280 sensor records atmospheric pressure and altitude using I2C or SPI protocols.
  - **Rain Detection**: Rain sensors identify water presence and quantify rainfall by detecting resistance changes.
- **Visualization**:
  - Data from sensors can be displayed locally on LCD screens with integrated I2C modules or shared remotely via web or mobile applications.
  - Libraries such as those for graphical interfaces streamline the visualization process, ensuring users can easily interpret data.

Implementation of Data Transmission Using Wi-Fi for Remote Monitoring:

- **NodeMCU ESP8266**:
  - Acts as a central hub for processing and transmitting sensor data over Wi-Fi.
  - Sends information to cloud services or local servers using standard communication protocols such as HTTP or MQTT.
- **Wireless Connectivity**:
  - Configured to connect with wireless networks, enabling seamless real-time data updates for remote monitoring.
  - Ensures secure and efficient data transmission through industry-standard encryption methods.
- **Remote Access**:
  - Allows sensor data to be viewed on web interfaces or mobile dashboards, accessible from anywhere.

VII. RESULTS AND ANALYSIS

In an IoT-based weather monitoring system, real-time environmental data is shown on an LCD screen, which displays several parameters such as temperature, humidity, pressure, rainfall, and light intensity. Here's an in-depth breakdown of each parameter and its presentation:



1) **Temperature:**
- **Display Format**: Typically shown in Celsius (°C).
- **Sensor Used**: DHT22 sensor.
- **Representation**: The screen will display a line like "Temp: 25.3°C", updating constantly based on the sensor's measurement of the surrounding temperature.

2) **Humidity**:
- **Display Format**: Displayed as a percentage (%).
- **Sensor Used**: DHT22 sensor.
- **Representation**: The screen will show "Humidity: 60%". This represents the amount of water vapor in the air relative to the maximum it could hold at the current temperature, helping to assess comfort and environmental conditions.

3) **Pressure**:
- **Display Format**: Shown in hectopascals (hPa).
- **Sensor Used**: BMP280 sensor.
- **Representation**: A line like "Pressure: 1013 hPa" appears, indicating the weight of the atmosphere pressing down at that location. Changes in pressure are valuable for forecasting weather patterns.

4) **Rainfall**:
- **Display Format**: Presented as either "Rain: Yes" or "Rain: No", and sometimes includes the intensity of rainfall.
- **Sensor Used**: Rain Sensor.
- **Representation**: If rain is detected, the display shows "Rain: Yes"; otherwise, it shows "Rain: No". Some systems might also provide the rainfall intensity, for example, "Rain: 5 mm/hr", which gives more precise details on precipitation.

5) **Light Intensity**:

- **Display Format**: Displayed as a numerical value, often in lumens or relative units.
- **Sensor Used**: Light Dependent Resistor (LDR).
- **Representation**: A line like "Light: 300" shows the current ambient light level. A higher number indicates stronger light, with lower numbers corresponding to dimmer conditions.

VIII. CONCLUSION

The IoT-powered weather monitoring system effectively combines cutting-edge technology to deliver real-time environmental data through an intuitive and accessible interface. By integrating various sensors such as the DHT22 for temperature and humidity, the BMP280 for atmospheric pressure, a Rain Sensor for detecting precipitation, and an LDR for measuring light intensity, the system ensures precise and continuous tracking of essential weather variables.

The heart of the system lies in the ESP8266 microcontroller, which manages data acquisition, processing, and communication. It takes the raw data from the sensors, transforms it into useful metrics, and sends it to a cloud server for remote monitoring. Meanwhile, the 16x2 LCD display, powered by the I2C interface, shows users the current conditions with real-time updates on temperature, humidity, pressure, rainfall, and light intensity.

Designed for both local and remote monitoring, this system enhances decision-making and operational productivity across various fields. Whether it's for agriculture, environmental studies, or personal weather tracking, it offers crucial insights and practical advantages. The system's ability to present accurate and timely data in a user-friendly manner highlights its versatility and effectiveness in diverse weather monitoring applications.

In summary, the IoT-driven weather monitoring system exemplifies the potential of combining IoT technology with environmental sensing to provide reliable, accurate, and easily accessible weather information.

IX. REFERENCE

1. **Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M.** (2015). *Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications*. IEEE Communications Surveys & Tutorials, 17(4), 2347-2376.
2. **Mishra, D., & Singh, A.** (2021). *Design and Implementation of IoT-Based Weather Monitoring System*. International Journal of Advanced Research in Computer Science, 12(1), 100-105.
3. **Patil, A., & Pawar, D.** (2020). *IoT-Based Weather Monitoring System*. International Journal of Innovative Research in Technology, 7(4), 56-62.
4. **Gupta, P., & Pal, S. K.** (2020). *A Study of Weather Monitoring Using IoT Systems*. Journal of Emerging Technologies and Innovative Research, 7(8), 102-110.
5. **DHT22 Sensor Datasheet.** (n.d.). Retrieved from https://www.adafruit.com/dht22
6. **BMP280 Sensor Datasheet.** (n.d.). Retrieved from https://www.bosch-sensortec.com/products/environmental-sensors/pressure-sensors/bmp280/

7. **Reddy, K. S., & Reddy, B. C.** (2022). *IoT-Based Weather Monitoring and Analysis System*. International Journal of Advanced Science and Technology, 31(5), 12-18.
8. **Verma, P., & Agrawal, S.** (2018). *Design and Implementation of a Real-Time Weather Monitoring System Using IoT*. Proceedings of the 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), 527-530.
9. **Kumar, P., & Singh, R.** (2019). *IoT-Based Weather Monitoring System Using Low-Power Sensors*. International Journal of Computer Applications, 178(12), 25-30.
10. **LDR Sensor Technical Specifications.** (n.d.). Retrieved from https://components101.com/sensors/ldr
11. **Rain Sensor YL-83 Datasheet.** (n.d.). Retrieved from https://lastminuteengineers.com/rain-sensor-arduino-tutorial/
12. **Chaudhary, P., & Gupta, A.** (2020). *Integration of Sensors in IoT-Based Weather Monitoring Systems*. Journal of Environmental Monitoring Systems, 45(3), 112-118.
13. **Arduino IDE: Comprehensive Guide.** (n.d.). Retrieved from https://www.arduino.cc/en/Guide