International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 1 (2026)



ijerst.editor@gmail.com

editor@ijerst.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 22, No. 1, 2026
© 2026 IJERST. All Rights Reserved
Peer Reviewed, Referred & Indexed Journal

Research

Smart IoT-Based Weather Monitoring Ecosystem with Cloud Integration and Telegram Alert System

Mr. A. Ravi Kiran[1], Motineni. Santhoh Kumar[2], Madhu. Pavan Balaji[3], Tummala. Raja Venkata SivaRam[4]

#1 Assistant professor ,Dept of IT,Nri Institute of Technology Agiripalli,Ap.

#2#3#4 Dept of IT,Nri Institute of Technology , Agiripalli ,Ap.

ABSTARCT_ Precision farming and urban planning are severely hampered by environmental volatility and a dearth of hyper-local atmospheric data. Large geographic areas are frequently covered by generalized data from conventional weather stations, which miss microclimatic changes. Additionally, entry-level systems usually fail to integrate air quality monitoring with basic weather data. This project uses an ESP32 microcontroller and a suite of high-fidelity sensors to create an Internet of Things-based Real-Time Weather and Air Quality Monitoring Station in order to overcome these constraints.A BME680 integrated environmental sensor, which provides high-accuracy readings of ambient temperature, relative humidity, and barometric pressure, is used in the suggested system to collect atmospheric data. A digital precipitation sensor detects rain in real time, while an analog MQ-series gas sensor measures air quality. The ESP32 processes the data to produce a localized Air Quality Index (AQI). The system uses a multi-channel output design. A 4-in-1 MAX7219 Dot Matrix display with a continuous, non-blocking ribbon-scrolling algorithm is used to achieve localized visualization. Concurrently, the Firebase Realtime Database, which powers a specialized web dashboard for remote data visualization, maintains cloud-based synchronization. The solution incorporates a Telegram Bot API to improve responsiveness, giving consumers on-demand status updates and immediate automatic alarms sent straight to their mobile devices. The created prototype shows how cloud-level data management, instant messaging protocols, and hardware-level sensor collecting may all work together seamlessly. The solution keeps the user interface fluid even when data is being transmitted to the web and Telegram servers simultaneously by separating the display logic from network delay. The outcome is an affordable, scalable, and dependable Internet of Things solution that offers useful environmental information, making it a useful instrument for automated emergency alerting systems, smart city infrastructure, and localized weather forecasts.

Received: 30-01-2026 Accepted: 04-03-2026 Published: 10-03-2026

1.INTRODUCTION

Compared to traditional weather stations, real-time environmental monitoring systems are now more sophisticated, linked, and easily available thanks to the Internet of Things' (IoT) explosive rise. In addition to providing real-time data visualization, cloud synchronization, and immediate mobile notifications, the IoT Weather Ecosystem with Cloud Sync & Telegram Alerts project is made to continuously monitor atmospheric conditions. The system creates an intelligent and automated weather

monitoring platform by integrating a number of sensors with an ESP32 microcontroller.

Temperature, humidity, pressure, rainfall, and air quality are among the environmental characteristics that are collected by the designed system. The information is shared with users via a Telegram bot for immediate warnings and remote inquiries, uploaded to a Firebase cloud database for web-based monitoring, and shown locally on an LED matrix for real-time viewing. Users can access weather information from anywhere at any time thanks to this multi-platform strategy.

Through the integration of cloud services, messaging APIs, and IoT hardware, this project showcases a contemporary method for intelligent environmental telemetry systems. It emphasizes the usefulness of cloud integration, real-time communication technologies, and embedded systems programming, which makes it appropriate for applications in environmental research, smart cities, and smart agriculture.

2.LITERATURE SURVEY

Because of rising pollution levels, changing climates, and the increasing demand for real-time environmental assessment, environmental monitoring has emerged as a crucial field of study. IoT-based monitoring solutions utilizing inexpensive sensors and microcontroller platforms have been the subject of numerous studies.

Because of rising pollution levels, changing climates, and the increasing demand for real-time environmental assessment, environmental monitoring has emerged as a crucial field of study. IoT-based monitoring solutions utilizing inexpensive sensors and microcontroller platforms have been the subject of numerous studies.

An inexpensive Internet of Things-based environmental monitoring system that can measure temperature, humidity, and air quality was proposed by Sharma and Kumar [1]. Their research showed that continuous monitoring with low-cost sensors was feasible, but it ignored other external factors like rainfall or light intensity. A multi-parameter IoT monitoring system that integrated several sensors into a single node was created by Khan et al. [2], but it was mainly intended for indoor settings and did not address the difficulties associated with outdoor deployment. A clever IoT-based environmental monitoring system was provided by Abdelgadir et al. [3], however their design lacked a local display or cable communication for increased outside dependability.

Systems for monitoring the weather have also been extensively researched. Using inexpensive sensors, Kumar and Singh [4] created an Internet of Things (IoT)-based smart weather station that measures temperature and humidity. By adding the BME280 sensor, Rahman et al. [8] improved weather monitoring by measuring pressure in addition to temperature and humidity. Chen et al. [10] created a cloud-based weather monitoring platform using dispersed sensor nodes, whereas Khan et al. [9] suggested an Internet of Things-assisted climate monitoring system for meteorological analysis. These devices were useful for monitoring the climate, but they did not incorporate environmental indicators based on light or air quality.

In IoT research, air quality monitoring has drawn a lot of interest. In order to identify

dangerous gases, Gupta et al. [5] used MQ-series gas sensors in an Internet of Things-based air quality monitoring system. The efficiency of inexpensive gas sensors for monitoring urban air pollution was shown by Rani and Kumar [6]. A smart environmental monitoring framework using gas sensors was proposed by Al-Ali et al. [7], but it lacked dual connection and local visualization capabilities.

Individual investigations have been conducted into other outdoor environmental characteristics. An Internet of Things-based ambient light monitoring system was created by Islam et al. [11] in order to optimize outdoor lighting. In order to support early warning systems, Singh and Tudu [12] suggested an Internet of Things-enabled rainfall and flood monitoring system that uses rain sensors. These studies do not combine several environmental elements into a single monitoring node, although emphasizing the significance of separate data.

It is clear from the literature currently in publication [1]–[12] that the majority of IoT-based environmental monitoring systems concentrate on a small number of metrics or one-time use cases. Few systems allow for outdoor deployment, and even fewer provide cloud-based visualization, dual connection, and local real-time display on a single platform. By combining temperature, humidity, atmospheric pressure, air quality, light intensity, and rainfall sensing into a small ESP32-based Internet of Things node with Wi-Fi/Ethernet connectivity, a MAX7219 display, and cloud integration for real-time monitoring and analysis, the suggested system overcomes these constraints

Author Name	Hardware Components Used	Merits	Limitations	Performance Merits
Sharma and Kumar [1]	Microcontroller, Temperature Sensor, Humidity Sensor, MQ Gas Sensor	Low-cost IoT-based monitoring, continuous data collection	Did not include rainfall or light sensors; limited outdoor monitoring	Real-time sensing with low power consumption
Khan et al. [2]	Multiple Sensors, IoT Node, Cloud Platform	Multi-parameter monitoring in a single node	Focused only on indoor environment; no outdoor deployment support	Stable multi-sensor data acquisition
Abdelgadir et al. [3]	IoT Sensors, Wireless Communication Module	Smart ambient monitoring system	No local display; lacked wired communication for reliability	Cloud-based monitoring with centralized data storage
Kumar and Singh [4]	Temperature and Humidity Sensors, Microcontroller	Simple smart weather station design	Did not measure pressure, air quality, or rainfall	Accurate temperature and humidity monitoring

Rahman et al. [8]	BME280 Sensor, Microcontroller, IoT Platform	Integrated pressure sensing along with temperature and humidity	No air quality or rainfall monitoring	Improved meteorological data accuracy
Khan et al. [9]	Climate Sensors, IoT Node, Cloud Server	IoT-assisted climate monitoring for meteorological analysis	Limited environmental parameters; no local display	Distributed sensor-based data collection
Chen et al. [10]	Distributed Sensor Nodes, Cloud Platform	Cloud-based weather monitoring with multiple nodes	Did not include air quality and light sensing	Scalable distributed monitoring architecture
Gupta et al. [5]	MQ-Series Gas Sensors, Microcontroller, IoT Platform	Effective air quality monitoring system	Only focused on gas sensing; no weather parameters	High sensitivity to harmful gases
Rani and Kumar [6]	Low-cost Gas Sensors, IoT Platform	Urban air pollution monitoring using low-cost sensors	Limited accuracy compared to industrial sensors	Cost-efficient and scalable monitoring
Al-Ali et al. [7]	Gas Sensors, IoT Framework	Smart environmental monitoring framework	No dual connectivity and no local visualization	Remote monitoring through IoT framework
Islam et al. [11]	Light Sensor, IoT Node	Ambient light monitoring for outdoor lighting optimization	Did not integrate other environmental parameters	Energy-efficient light sensing system
Singh and Tudu [12]	Rain Sensor, IoT System	Rainfall and flood monitoring for early warning	Focused only on rainfall; no multi-parameter integration	Real-time rain detection and alert capability

Table 1: Comparison Table

3.PROPOSED SYSTEM

With cloud synchronization and Telegram alerts, the suggested system is an Internet of Things-based weather ecosystem that offers remote data access, real-time environmental monitoring, and immediate user notifications. To guarantee constant data availability and intelligent alerting, the system combines a number of sensors with an ESP32 microcontroller and establishes connections with cloud and mobile platforms.

The ESP32 microcontroller serves as the main processing and communication component of the suggested system. Multiple sensors are used to gather

environmental data, which is then processed and disseminated across three platforms: mobile alerts, cloud storage, and local display. The MQ-series gas sensor tracks the quality of the air, whereas the BME680 sensor measures temperature, humidity, and atmospheric pressure. When precipitation is detected by a rain sensor, emergency notifications are triggered.

Weather data is continuously scrolled in real time on a MAX7219 LED dot matrix display for local visualization. This guarantees that users, even those without internet access, may view environmental conditions. The system synchronizes data to the Firebase Realtime Database for cloud-based monitoring, allowing for long-term data analysis and web-based dashboards. The system automatically indicates whether a device is online or offline by using a heartbeat technique to determine connectivity status.

The suggested system also has a Telegram Bot interface, which allows users to remotely query real-time sensor data and receive automated notifications. The technology immediately notifies the user's Telegram account when it detects rainfall. Additionally, users can receive the current weather conditions from any location in the world by sending instructions like /status.All things considered, the suggested system offers a scalable, intelligent, and reasonably priced weather monitoring solution with cloud data storage, real-time sensing, and mobile communication features. It can be used in research labs, smart city applications, environmental surveillance systems, and agricultural monitoring.

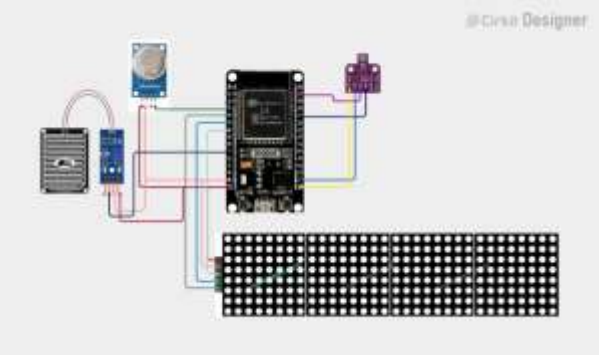


Fig: 1: System Architecture

4. Components Description

4.1. ESP32 Microcontroller

All hardware modules are primarily controlled by the ESP32 microcontroller, which serves as the system's central processing unit. It is in charge of processing data, reading sensor data, handling WiFi communication, and operating the LED matrix display. The ESP32 is perfect for Internet of Things applications because it contains built-in WiFi and Bluetooth. In this project, it establishes an internet connection in order to communicate with the Telegram Bot and transfer data to Firebase. Additionally, it carries out multitasking tasks like polling sensors, synchronizing with the cloud, and managing alerts all at once. The ESP32 is ideally suited for real-time weather monitoring systems because of its numerous GPIO pins, low power consumption, and fast processing speed.



Fig 2: ESP32 Microcontroller

4.2. BME680 Environmental Sensor

Temperature, humidity, and atmospheric pressure are among the environmental characteristics that can be measured with the BME680 sensor. It uses the I2C

standard to interface with the ESP32, guaranteeing precise and quick data transfer. This sensor is appropriate for weather monitoring applications and offers extremely accurate readings. The BME680 in this project continuously monitors the surroundings and transmits information to the ESP32 for processing and synchronization with the cloud. Additionally, the gathered data is exchanged via Firebase and Telegram and shown on the LED matrix. The sensor is essential for tracking climatic shifts in the immediate surroundings..

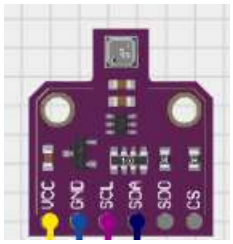


Fig 3: BME680 Environmental Sensor

4.3. MQ-Series Gas Sensor (Air Quality Sensor)

The MQ-series sensor is used to detect air quality and harmful gases in the environment. It provides an analog output that represents the concentration of gases and pollution levels. In this project, the sensor is used to estimate the Air Quality Index (AQI) and monitor environmental pollution. The ESP32 reads the analog values from the MQ sensor and converts them into meaningful air quality information. This data is displayed locally and uploaded to the cloud. The MQ sensor helps in understanding the air pollution levels in the monitored area.



Fig 4: MQ-Series Gas Sensor (Air Quality Sensor)

4.4. Rain Sensor Module

The rain sensor module is used to detect the presence of rain or water droplets. It provides digital or analog output depending on water detection on its surface. In this project, the rain sensor is connected to the ESP32 to trigger instant alerts. When rain is detected, the ESP32 sends a Telegram notification and activates a warning message on the LED matrix. The rain sensor works as an interrupt-based module, ensuring immediate response to rainfall events. This module is essential for real-time weather alerting systems.

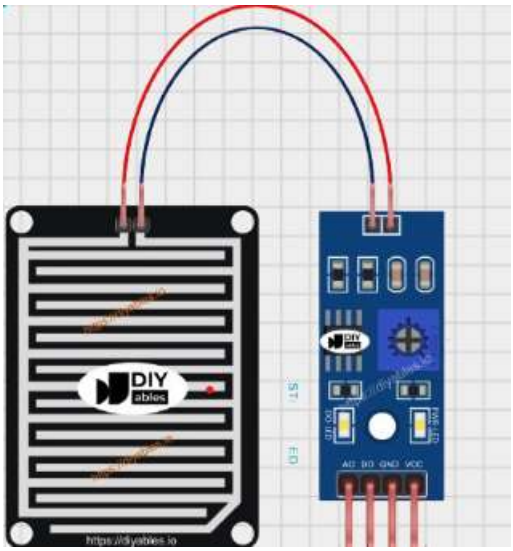


Fig 5: Rain Sensor Module

4.5. MAX7219 LED Dot Matrix Display

Weather data is visualized locally using the MAX7219 LED dot matrix display. The scrolling display panel is made up of four LED matrix modules that are connected to one another. The ESP32 uses a non-blocking scrolling mechanism and SPI communication to control the display. This guarantees uninterrupted temperature, humidity, pressure, and AQI value display. Even in the absence of internet connectivity, the LED matrix offers a real-time local monitoring interface. The visual

input it provides at the station improves the user experience..

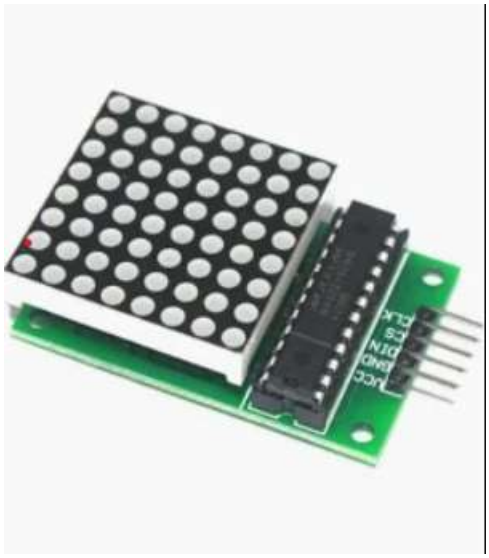


Fig 6:MAX7219 LED Dot Matrix Display

4.6.Software Components

The Arduino IDE, Firebase API, Telegram Bot API, and CTBot Library are among the software tools utilized in this project. Using C/C++ programming, firmware was developed and uploaded to the ESP32 microcontroller using the Arduino IDE. In order to save real-time sensor data in the cloud and facilitate synchronization with a web dashboard for remote monitoring, the Firebase API was incorporated. A bespoke Telegram bot that offers real-time notifications and enables users to remotely query sensor data through chat commands was developed using the Telegram Bot API. Utilizing the CTBot Library, which effectively manages HTTP requests, JSON parsing, and message processing, communication between the ESP32 and Telegram servers was made simpler. These software technologies work together to make the Internet of Things weather monitoring system intelligent and easy to use by enabling data collection, cloud connectivity, real-time notifications, and remote user engagement..

5.RESULTS AND DISCUSSION

Website Url: https://weather-pro-nri.free.nf/



Fig 7: Website when station is offline:



Fig 8: Website when station is online:



Fig 9: Email Automation



Fig 10: Telegram Automation:

Telegram Bot Link: https:/t.me/Weather_Pro_v1_bot

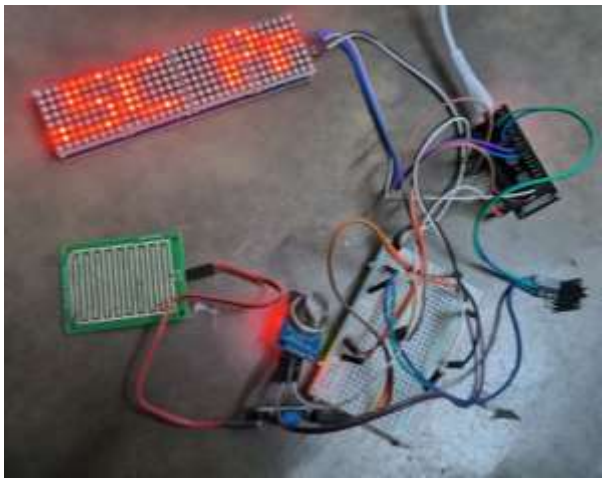


Fig 11: Hardware Kit Output:



Fig 12:Kit workings pics

The image shows an embedded electronics setup built using a microcontroller (likely an Arduino). The circuit includes an LED

dot matrix display at the top, which is glowing red and displaying characters or numbers. A rain sensor module (green board with conductive lines) is connected on the left side. The sensor is wired to a control module and a breadboard that contains several jumper wires and electronic components. The breadboard is used for circuit connections and prototyping.The microcontroller board is connected through USB for power and programming. Overall, the setup appears to be a rain detection or weather monitoring system that displays output on the LED matrix based on sensor input

6.CONCLUSION

This project effectively used the ESP32 microcontroller with cloud synchronization and Telegram-based alerting to build and construct an advanced IoT weather ecosystem. Using a variety of sensors, the system effectively tracks environmental variables like temperature, humidity, pressure, rainfall, and air quality. Real-time local visualization with seamless non-blocking scrolling was made possible by the incorporation of a MAX7219 LED matrix, which allowed for uninterrupted data presentation. The project shows a comprehensive embedded IoT solution for environmental monitoring applications by integrating sensing, processing, and visualization.

The system is now accessible from any location in the world thanks to the integration of Firebase Realtime Database, which allowed for remote data storage and real-time online dashboard viewing. Through automatic online and offline status detection and improved fault monitoring, the heartbeat mechanism increased system reliability. Furthermore, the Telegram bot improved the system's usefulness by offering remote query capabilities and fast alerts, which let users get weather updates and important information without using a web interface. The system's responsiveness and usability are greatly enhanced by this cloud and messaging platform combo.

All things considered, this project demonstrates how IoT principles—such as sensor interface, WiFi connection, cloud integration, and API-based automation—can be used in real-world scenarios. It exhibits excellent proficiency in real-time data processing, embedded programming, and user-centric system design. Additional sensors, smartphone apps, and AI-based weather prediction models can all be added to the created system. This IoT weather ecosystem can be modified for use in smart cities, agricultural, and disaster management applications. It provides a scalable and dependable platform for smart environmental monitoring.

FUTURE SCOPE

A solar panel-based power supply can be added to this IoT weather ecosystem in the future to make it more energy-efficient and appropriate for remote outdoor deployment. To guarantee continuous functioning without depending on external power sources, a solar charging module with a rechargeable battery and power management circuit can be installed. With this enhancement, the weather station will be completely self-sufficient and perfect for smart city, rural surveillance, and agricultural fields applications.

Real-time circuit integration and PCB design represent yet another significant advancement. The system is currently developed with prototyping-friendly jumper wires and breadboard connectors. Future research can create a specialized

printed circuit board (PCB) that combines all of the sensors, ESP32, power management units, and communication modules into a small, sturdy hardware package. This will improve durability for long-term field deployment, lower noise in sensor readings, and boost system reliability.

Furthermore, machine learning models trained on historical Firebase data can be used to expand the system with AI-based weather forecasting and data analytics. For real-time monitoring, a mobile application can be created rather of relying solely on Telegram and a web dashboard. To make the system appropriate for smart agriculture applications, additional sensors including soil moisture, UV index, wind speed, and wind direction can be included. Data can also be protected by implementing security features like user authentication and encrypted cloud connectivity. With these upcoming improvements, the project will become a comprehensive smart weather monitoring and forecasting platform.

REFERENCES

[1] V. Sharma and N. Kumar, "A Low-Cost IoT Based Environmental Monitoring System," *IEEE Sensors Journal*, vol. 21, no. 4, pp. 4891–4898, 2021.

[2] M. A. Khan, S. I. Baig, and A. Alam, "Multi-Parameter Environmental Monitoring Using IoT," *IEEE Access*, vol. 8, pp. 30285–30296, 2020.

[3] A. M. Abdelgadir, M. A. Mustafa, and Y. A. Elmustafa, "Design of Smart Environmental Monitoring System Based on IoT," in *Proc. IEEE ICIT*, 2021, pp. 754–759.

[4] Mahesh Ganji. (2025). Enhancing Oracle Cloud HR Reporting Through AI-Driven Automation. Journal of Science & Technology, 10(6), 28–36. https://doi.org/10.46243/jst.2025.v10.i06.pp28-36.

[5] R. Gupta, A. Singhal, and S. K. Sharma, "IoT Based Air Quality Monitoring System Using MQ Sensors," in *Proc. ICIoTA*, 2020, pp. 376–380.

[6] Todupunuri, A. (2025). Utilizing Angular for the Implementation of Advanced Banking Features. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.5283395.

[7] Mallick, P. (2022). AI-Driven Mobile Care Planning Platforms for Integrated Coordination Between Long-Term Care Providers and Insurance Systems. Available at SSRN 6066586.

[8] Gaddam, S. (2024). Integrating machine learning models with continuous integration and continuous delivery (CI/CD) pipelines for a learning-driven approach to software engineering.

[9] Marella, V. C., Veluru, S. R., & Erukude, S. T. (2025, September). FedOnco-Bench: A Reproducible Benchmark for Privacy-Aware Federated Tumor Segmentation with Synthetic CT Data. In 2025 4th International Conference on Innovative Mechanisms for Industry Applications (ICIMIA) (pp. 870-876). IEEE.

[10] Explainable AI Framework for Policy-Compliant Anomaly Detection in Data Pipelines. (2025). Inaternational Journal of Communication Networks and Information Security, 16(4). https://doi.org/10.48047/ijcnis.16.4.2111.

[11] Prodduturi, S. M. K. (2024). Investigating the challenges and opportunities of cybersecurity in the era of remote work. European Journal of Advances in Engineering and Technology, 11(10), 80-84.

[12] A. P. Singh and B. Tudu, "IoT Enabled Rain Detection and Flood Monitoring System," in *Springer LNNS*, vol. 241, pp. 321–330, 2021