Chandigarh Engineering College Jhanjeri
Mohali-140307
Department of Artificial Intelligence (AI) and Data Sciences

Engineering Clinics Project
Mid-Term Report
IoT Based Weather Monitoring System

BACHELOR OF TECHNOLOGY

(Artificial Intelligence and Data Science.)

SUBMITTED BY :

Ayush Gupta, Ayush Sharma, Ayush Shukla, Bablesh Kumar
2420690, 240691, 2420692, 2420692, 2420693 October
2025

Under the Guidance of :
Er. Shubham Sharma (J4224)
Assistant Professor

Department of Artificial Intelligence and Data Science Chandigarh Engineering College Jhanjeri Mohali – 140307

Table of Contents

Chapter 1 : Abstract and Acknowledgement

1.1 Abstract

Weather monitoring is a critical field that impacts vital sectors like agriculture, aviation, and disaster management. However, traditional meteorological systems are often expensive, sparsely distributed, and provide data with significant delays. This creates a significant information gap, as they fail to provide the real-time, hyperlocal data needed for efficient decision-making.

1.2 Acknowledgement

We wish to express our sincere gratitude to our project guide, Er. Shubham Sharma, Assistant Professor, Department of AI&DS, for his invaluable guidance, constant encouragement, and insightful feedback throughout the duration of this project.

We are also thankful to the **Department of Artificial Intelligence and Data Science, Chandigarh Engineering College, Jhanjeri**, for providing us with the necessary resources, infrastructure, and a conducive environment that made this project possible.

Finally, we extend our heartfelt thanks to our friends, peers, and families for their unwavering support, valuable discussions, and encouragement, which motivated us to bring this project to a successful conclusion.

Chapter 2 : Declaration

We, **Ayush Gupta, Ayush Sharma, Ayush Shukla,** and **Bablesh Kumar,** hereby declare that the work presented in this project report entitled "**IoT Based Weather Monitoring System**" is the outcome of our own original work. This work has been carried out under the guidance of Er. Shubham Sharma (Assistant Professor) at Chandigarh Engineering College, Jhanjeri (Mohali).

This report, or any part thereof, has not been submitted, either in part or in full, for the award of any other degree, diploma, or certificate at this or any other university or institution. All sources of information and data have been duly acknowledged and cited.

Chapter 3 : Introduction

3.1 Background and Motivation

In an era increasingly affected by climatic volatility, the demand for precise and timely weather information has never been higher. Weather monitoring is a field of immense critical importance, directly influencing the safety, efficiency, and profitability of numerous sectors. From a farmer planning, irrigation schedules and protecting crops, to an airline plotting safe flight paths, to a city's disaster management team preparing for an impending storm, access to accurate weather data is fundamental.

3.2 The Importance of Real-Time Weather Data

The value of weather data is directly proportional to its timeliness and granularity. Real-time data allows for proactive and immediate decisionmaking, which can be the deciding factor in outcomes across various domains:

- **Agriculture:** Farmers can optimize irrigation, pesticide application, and harvesting schedules based on immediate local conditions, directly impacting crop yields and resource conservation.
- **Aviation:** Real-time data on barometric pressure, temperature, and humidity is crucial for flight planning and ensuring safety, especially during take-off and landing.
- **Disaster Management:** The ability to monitor sudden changes in atmospheric conditions in real-time is essential for issuing timely warnings for flash floods, storms, or other weather-related

Chapter 4 : Literature Survey

4.1 Conventional Weather Monitoring Techniques

The field of meteorology has traditionally been the domain of government agencies and large research institutions. This conventional approach relies on a global network of highly sophisticated, expensive, and nonautomated weather stations.

These stations, while providing highly accurate and reliable data, suffer from several drawbacks as noted in Chapter 1. The primary issue is **spatial density**. Due to the high deployment and maintenance costs, stations are placed many kilometres apart. This sparsity means the data collected is an average over a large area and cannot capture localized, rapidly changing weather events, which are often the most dangerous. Furthermore, the data dissemination process is often slow. Data is collected, processed centrally, and then broadcast, leading to significant delays.

4.2 Advancements with IoT and Embedded Systems

The last decade has seen a revolution in electronics, catalysed by the emergence of **low-cost microcontrollers and sensors**. This has enabled a paradigm shift from centralized, expensive systems to decentralized, affordable, and "smart" devices.

The "Internet of Things" (IoT) describes this network of physical devices embedded with sensors, software, and connectivity, allowing them to exchange data over the internet. This technology is perfectly suited to address the shortcomings of traditional weather monitoring.

Chapter 5 : System Requirements

The following software tools and platforms are required for development:

Component	Specification	Purpose
Firmware IDE	Arduino IDE	Used to program the ESP32 microcontroller in C++.
App Dev IDE	Android Studio	The official IDE for developing the native Android application.
Cloud Platform	ThingSpeak IoT Platform Account	The cloud backend for data aggregation, storage, visualization, and API access.
Libraries	ESP32 Core for Arduino	Provides board definitions and core functions.
	Adafruit DHT Sensor Library	To interface with the DHT22 sensor.
	Adafruit BMP280 Library	To interface with the BMP280 sensor via I2C/SPI.
	ThingSpeak MQTT/HTTP Library	To simplify data transmission to the cloud.

Chapter 6 : System Design

6.1 System Architecture

The system is designed using a **three-layer architecture**, which separates the concerns of data sensing, data processing, and data presentation. This modular design is robust, scalable, and easy to maintain. The three layers are:

1. **Hardware (Sensing) Layer:** The physical layer responsible for sensing the environment and transmitting data.
2. **Communication & Cloud Layer:** The middleware responsible for data transport, storage, and processing.
3. **Application (Presentation) Layer:** The user-facing layer that visualizes the data and provides insights.

The following sections detail the components and technologies used in each layer.

6.3 Communication & Cloud Layer

This layer acts as the bridge between the physical hardware and the enduser application. It handles data transmission, storage, and API access.

- **Lightweight:** Has a very small packet header, minimizing network bandwidth.
- **Communication Protocol: Wi-Fi & MQTT** The ESP32 collects sensor data and transmits it over a standard **Wi-Fi** (802.11b/g/n) network. For data transmission, the **MQTT** protocol is used. MQTT is an extremely lightweight, publish-subscribe messaging protocol. It is the industry standard for IoT because it is:

Chapter 7 : Implementation

7.1 Overview of Technology Stack

The implementation of this project relies on a stack of technologies spanning hardware, firmware, cloud services, and mobile development. • **Hardware:** ESP32 Development Board , DHT22 Temperature/Humidity Sensor , BMP280 Barometric Pressure Sensor.

- **Firmware:** C/C++ via the Arduino IDE. Key libraries include WiFi.h (for connectivity), PubSubClient (for MQTT), and sensorspecific libraries (e.g., Adafruit_BMP280).
- **Cloud Platform:** ThingSpeak IoT Platform. Used for data ingestion, storage, and API provision.
- **Communication Protocol:** MQTT for publishing data from the ESP32, and REST/HTTP for data retrieval by the Android app.
- **Application:** Native Android (Java/Kotlin) developed in Android Studio. Key libraries include Retrofit (for API calls) and MPAndroidChart (for data visualization).

7.2 Microcontroller Programming (Firmware)

The firmware was developed in C/C++ using the Arduino IDE. The code is structured into several key logical blocks:

1. **Initialization (setup()):**

   ○ Initializes the serial monitor for debugging. ○ Initializes the DHT22 and BMP280 sensors. ○ Calls a setup_wifi() function, which connects the ESP32 to the predefined Wi-Fi SSID and password. ○ Sets up the MQTT client with the ThingSpeak server address, channel ID, and API key.

Chapter 8 : Results and Discussion

8.1 Project Results Overview

The project was successful in achieving all its primary objectives. A functional, end-to-end prototype of an IoT-based weather monitoring system was designed, built, and validated.

The prototype **successfully operated continuously**, validating the core design principles of using an ESP32, ThingSpeak, and a custom Android application. The system reliably collected data from the DHT22 and BMP280 sensors, transmitted it to the cloud, and displayed it on the mobile dashboard.

The following sections detail the specific outcomes of the functional and performance testing.

8.2 Challenges Faced & Solutions Implemented

No project is without its challenges. The following three issues were identified during testing and successfully resolved.

- **Challenge: Wi-Fi Flakiness & Connection Drops** ○ **Problem:** The initial firmware was "optimistic," assuming the Wi-Fi would always be available. During testing, if the router was reset, the ESP32 would crash or hang, failing to send any more data.
    - **Solution:** We implemented **robust re-connection logic** in the firmware's main loop. Before any attempt to publish, the code now checks the WiFi.status() and mqtt.connected(). If either check fails, it triggers a non-blocking reconnect() function that continuously attempts to re-establish the connection before proceeding.

      .

Chapter 9 : Conclusion and Future Work

9.1 Conclusion

This project has been an invaluable experience in applying the principles of Artificial Intelligence and Data Science to a tangible, real-world hardware environment. We successfully designed, built, and validated a complete, end-to-end **IoT Based Weather Monitoring System**.

The project achieved its core objective of creating a low-cost, scalable, and autonomous system that overcomes the primary limitations of traditional weather monitoring. By integrating a custom-built hardware node with a robust cloud platform and a user-friendly mobile application, we have demonstrated a practical solution for providing the **real-time, hyperlocal weather data** that modern applications in agriculture, logistics, and public safety demand.

9.2 Future Enhancements

While the current prototype is a complete success, this project lays the groundwork for numerous exciting expansions. We aim to expand the project's capabilities in the next phase:

1. **Add More Sensors:** The most immediate enhancement is to integrate an **anemometer and a wind vane**. This would add crucial data on wind speed and direction, making the station much more comprehensive. Sensors for UV index, air quality (PM2.5), and rainfall (tipping bucket) could also be added.

Chapter 10 : References

1. Adithan, M. and Gupta, A.B. (1996), "Manufacturing Technology", New Age, International Publishers, New Delhi.
2. Kumar, N., Kumar, S. and Prasad, R., 2019. "IoT-Based Smart Weather Monitoring System using NodeMCU", Proceedings of Second International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, pp. 245-249.
3. Singh, S. and Shan, H. S. (2002) "Development of Magneto Abrasive Flow Machining Process", International Journal of Machine Tools & Manufacturing, vol.42, 2, 2002, pp. 953-959.
4. *Arduino IDE.* (n.d.). Retrieved from https://www.arduino.cc/en/main/software
5. *ThingSpeak IoT Analytics.* (n.d.). Retrieved from https://thingspeak.com
6. *ESP32 Datasheet.* (n.d.). Espressif Systems. Retrieved from https://www.espressif.com/en/products/socs/esp32
7. *DHT22 (AM2302) Datasheet.* (n.d.). Aosong.
8. *BMP280 Datasheet.* (n.d.). Bosch Sensortec.