SMART WEATHER MONITORING SYSTEM

A PROJECT REPORT

submitted by

GAYATHRI VR (230701090)
HARITHA M (230701107)
JYOSHNA K (230701132)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE,

ANNA UNIVERSITY: CHENNAI 600 025

MAY 2025

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

BONAFIDE CERTIFICATE

Certified that this project report titled "**SMART WEATHER MONITORING SYSTEM"** is the bonafide work of "**GAYATHRI VR(230701090), HARITHA (230701107), JYOSHNA K(230701132)"** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Ms. S. Ponmani M.E.,MBA,

SUPERVISOR

Assistant Professor

Department of Computer Science and Engineering

Rajalakshmi Engineering College

Chennai - 602 105

Submitted to Project Viva-Voce Examination held on ____________________

Internal Examiner **External Examiner**

ABSTRACT

Traffic congestion is a persistent challenge in urban areas, exacerbated by factors such as increasing population density and limited infrastructure expansion. To address this issue, this study proposes the development of an Internet of Things (IoT) traffic management system aimed at improving traffic flow and reducing congestion through real-time traffic detection and automatic signal control. The methodology involves a comprehensive literature review to inform the design and implementation of the system, followed by requirements analysis, system design, prototype development, evaluation, and refinement. Key components of the proposed system include sensors for traffic detection, IoT devices for data collection and communication, and algorithms for dynamic signal control. The system aims to optimize traffic flow by adjusting signal timings based on real-time traffic conditions, leveraging advanced data analytics and decision-making mechanisms. Through iterative refinement and testing, the prototype system demonstrates promising results in improving urban mobility and creating more sustainable and livable cities. Future work will focus on enhancing scalability, efficiency, and security, as well as deployment and maintenance in real-world traffic environments.

ACKNOWLEDGEMENT

First, we thank the almighty God for the successful completion of the project. Our sincere thanks to our chairman **Mr. S. Meganathan, B.E., F.I.E.,** for his sincere endeavor in educating us in his premier institution. We would like to express our deep gratitude to our beloved Chairperson **Dr. Thangam Meganathan, Ph.D.,** for her enthusiastic motivation which inspired us a lot in completing this project, and Vice-Chairman **Mr. Abhay Shankar Meganthan**, **B.E., M.S.,** for providing us with the requisite infrastructure. We also express our sincere gratitude to our college principal, **Dr.S.N.Murugesan M.E., PhD.,** and **Dr. P. Kumar M.E., Ph.D., Head of the Department of Computer Science and Engineering,** and our project guide **Ms. S. Ponmani M.E.,MBA,** for her encouragement and guiding us throughout the project. We would like to thank our parents, friends, all faculty members, and supporting staff for their direct and indirect involvement in the successful completion of the project for their encouragement and support.

CHAPTER 1

INTRODUCTION

Weather conditions significantly influence daily human activities, health, and comfort. In the context of climate variability and increasing awareness around personal well-being, there is a pressing need for intelligent systems that not only monitor environmental parameters but also provide contextual and actionable suggestions. Traditional weather monitoring solutions often provide raw data without interpretation, limiting their usefulness for end users, especially children, pet owners, and health-sensitive individuals.

EcoSense addresses this gap by introducing a smart IoT-based weather monitoring and advisory system that combines sensor-driven data collection with real-time suggestion delivery. The system utilizes temperature and humidity sensors integrated with a microcontroller and LCD for local display, while also transmitting data to a cloud platform for visualization via a web interface. When real-time data is unavailable, EcoSense intelligently falls back to mock data to ensure continuity in its advisory functionality. The platform generates tailored recommendations for general users, children, and pets, promoting hydration, appropriate clothing, and outdoor activity safety.

This project follows a structured development process, including requirement analysis, hardware-software integration, rule-based suggestion logic, and user-friendly web dashboard design. By delivering context-aware and role-specific suggestions based on environmental conditions, EcoSense demonstrates the potential of IoT in enhancing personal wellness and public awareness. The solution not only encourages proactive behavior in response to changing weather patterns but also lays the groundwork for future expansion into areas like air quality monitoring and AI-driven personalization.

1.1 Motivation

Promoting Environmental Awareness and Personal Well-being: Weather conditions significantly impact health, comfort, and daily activities. However, most individuals lack access to timely and context-specific weather insights. This project aims to bridge this gap by offering real-time, localized weather monitoring and intelligent suggestions to promote healthier lifestyle choices and safety, especially for vulnerable groups like children and pets.

Enhancing Smart Living through IoT Integration: By integrating Internet of Things (IoT) technologies, this project demonstrates how sensors, cloud platforms, and intelligent software can be combined to deliver a smart environmental monitoring solution. Real-time data acquisition and processing enable EcoSense to provide not just weather readings, but actionable recommendations based on current atmospheric conditions.

Addressing Data Gaps with Fallback Mechanisms: In many regions, especially semi-urban or rural areas, live environmental data may not always be available. EcoSense incorporates a fallback mechanism using mock data to ensure continuous operation and advisory services, maintaining user engagement and system reliability even under network or sensor failure conditions.

1.2 Objectives

Develop a Smart IoT-Based Weather Monitoring System: The core objective of this project is to build a functional weather monitoring platform that uses temperature and humidity sensors to collect real-time data, display it locally via LCD, and transmit it to a web dashboard for user access and interaction.

Integration of Sensor Technologies: Deploy digital temperature and humidity sensors (such as DHT11/DHT22) connected to a microcontroller (like ESP8266 or Arduino) for continuous monitoring of environmental conditions. These sensors serve as the foundation for accurate, real-time data collection in diverse climatic settings.

Intelligent Suggestion Generation and Data Handling: Design and implement a rule-based logic engine to process sensor data and generate tailored suggestions for general users, children, and pet care. When real-time data is not available, use mock data to maintain functionality and continue delivering insights. Future expansion includes the use of machine learning to refine suggestions and personalization.

CHAPTER 2

LITERATURE REVIEW

[1] Real-Time Weather Monitoring and Notification System Using IoT

This study proposes an IoT-based weather monitoring solution that collects temperature and humidity data using sensors and displays it on a web platform. The system emphasizes real-time alerts for extreme conditions but lacks contextual interpretation of data. EcoSense builds upon this foundation by not only monitoring but also delivering role-based suggestions for various user groups such as children and pet owners.

[2] IoT-Based Smart Environmental Monitoring System for Real-Time Data Acquisition

The paper highlights the implementation of an environmental monitoring system using DHT11 sensors and ESP8266 microcontrollers. The collected data is uploaded to a cloud server for storage and analysis. However, this system only visualizes the data. In contrast, EcoSense interprets the environmental data to provide health, activity, and clothing suggestions, enhancing user engagement and decision-making.

[3] A Weather-Based Decision Support System for Smart Cities

This work explores decision support using weather data, particularly in smart urban environments. It applies simple rule-based logic for public notifications related to air quality and weather extremes. EcoSense adapts a similar rule-based engine but focuses more on personalized guidance and fallback mechanisms using mock data when live sensor feeds are unavailable.

[4] Smart Weather Station with Real-Time Data Display and Alerts

The paper proposes a DIY smart station that uses an LCD screen and cloud storage for displaying temperature, humidity, and air quality levels. Although it includes alerting capabilities, it does not address usability across diverse population groups. EcoSense differentiates itself by including suggestions specifically tailored for children (e.g., hydration, sun protection), pets (e.g., avoiding hot surfaces), and general users (e.g., meal advice based on temperature).

[5] Intelligent IoT-Based Climate Monitoring System with Predictive Analysis

This research integrates weather data with predictive models to forecast trends in temperature and humidity. It uses statistical tools and machine learning to alert users about upcoming weather changes. EcoSense, while currently rule-based, is designed to evolve into a predictive system, with future implementation plans to incorporate ML models for suggestion refinement and personalization.

2.1 Existing System

Conventional weather monitoring systems typically rely on centralized meteorological stations that collect and disseminate weather data through websites, television broadcasts, or mobile apps. These systems often cover large geographic areas and provide generalized forecasts that may not accurately reflect hyperlocal environmental conditions. While they are effective in providing broad forecasts and severe weather alerts, they lack personalization and real-time interactivity.

Many traditional systems do not integrate Internet of Things (IoT) capabilities, which limits their ability to capture and analyze data at the user level. In rural or remote areas, the absence of nearby weather stations often results in delayed or missing data. Additionally, existing systems rarely offer real-time, situation-based recommendations for specific groups such as children, pets, or health-sensitive individuals.

2.1.1 Advantages of the existing system

Traditional weather systems provide broad forecasts and severe weather alerts, making them useful for general awareness. These forecasts are accessible via TV, websites, and mobile apps, ensuring easy public access. Additionally, they are supported by official meteorological agencies, which ensures credibility and accuracy.

2.1.2 Drawbacks of the existing system

Existing systems often lack real-time, hyperlocal data, making them less useful for specific locations. They also don’t offer personalized suggestions tailored to different user groups like children or pet owners. In areas with weak connectivity, accessing timely updates can be difficult. Moreover, they do not adapt or provide fallback options when sensors or data sources fail.

2.2 Proposed System

The proposed system offers a modern solution to traffic management by incorporating real-time monitoring and adaptive signal control using IoT technology and Infrared (IR) sensors. Unlike conventional systems that operate on fixed-time signal cycles, this system dynamically adjusts traffic light durations based on live traffic data.

IR sensors detect vehicle presence and flow at intersections, allowing the system to prioritize roads with higher vehicle density. During peak hours or in case of congestion, green light durations are intelligently extended to help clear traffic effectively. This adaptive approach enhances traffic flow, reduces delays, and prevents unnecessary idling.

Additionally, the system is scalable and cost-efficient, making it suitable for implementation across different urban settings—from small towns to major cities. Its flexibility and real-time responsiveness mark a significant improvement over traditional traffic control methods.

2.2.1 Advantages of the proposed system

The proposed system ensures real-time traffic monitoring, enabling faster and smarter decisions at intersections. By using IoT and IR sensors, it adapts signal timings dynamically, reducing unnecessary delays and improving overall traffic flow. It effectively minimizes congestion, especially during peak hours, by prioritizing high-density roads.

Additionally, the system is energy-efficient and reduces fuel consumption by cutting down idle time for vehicles. Its modular design allows for easy scalability and integration into existing infrastructure, making it a cost-effective solution for cities aiming to modernize their traffic management systems.

CHAPTER 3

SYSTEM DESIGN

1. **Development Environment**

3.1.1 Hardware Requirements

Arduino UNO
Breadboard
DHT11 Sensor
ESP32
Jumper Wires
LED Display

Arduino

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online.

Arduino UNO

The Arduino UNO is a popular microcontroller board that serves as the brain of the project, controlling the operation of various components and executing programmed tasks.

Breadboard

The breadboard provides a platform for prototyping and connecting electronic components without the need for soldering, allowing for easy experimentation and modification of circuit designs.

DHT11 Sensor

The DHT11 Sensor is a digital sensor used to measure temperature and humidity. It provides accurate and reliable data by using a thermistor for temperature and a capacitive humidity sensor for moisture detection. It is commonly used in weather monitoring and IoT applications.

ESP32

The **ESP32** is a low-cost, low-power microcontroller with built-in Wi-Fi and Bluetooth capabilities. It is widely used in IoT projects for connecting devices to the internet and enabling wireless communication between them.

Jumper wires

Jumper wires are used to establish connections between components on the breadboard or between the breadboard and Arduino UNO, facilitating the flow of electrical signals in the circuit.

Red, Green and Yellow LEDs

The red and green LEDs serve as visual indicators, providing feedback on system status or conditions such as item scanning success (green) or error (red), enhancing user interaction and understanding.

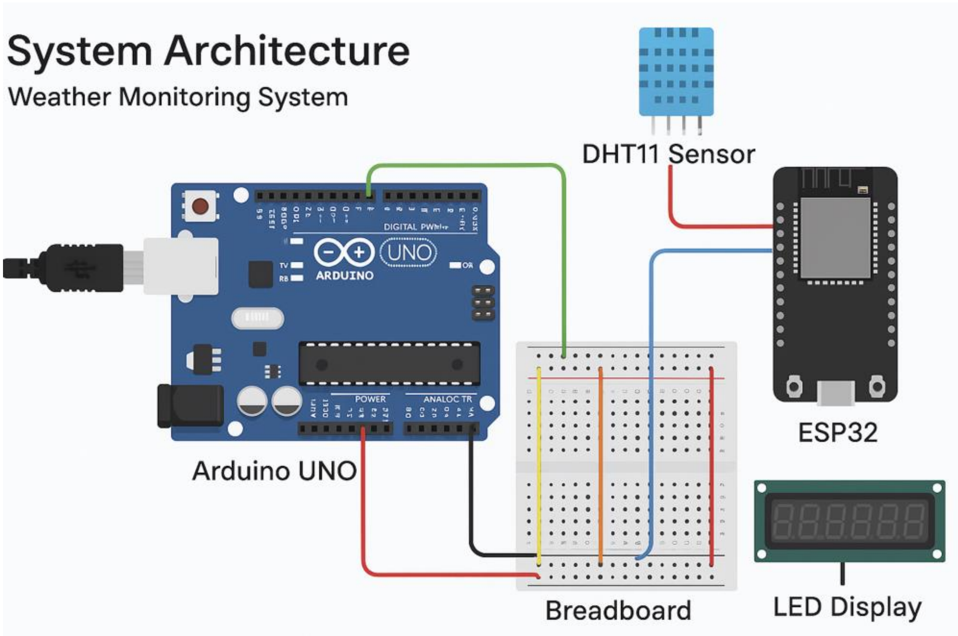
3.1.1 Software Requirements

- Arduino IDE
- Tinker

CHAPTER 4

PROJECT DESCRIPTION

4.1 SYSTEMARCHITECTURE



4.2Methodology

1. **Problem Definition:**
   The methodology begins by defining the problem statement: the development of an IoT-based weather monitoring system capable of measuring temperature, humidity, and other environmental parameters in real time. The system will provide suggestions and notifications based on the collected data to enhance daily activities, health, and safety for users.
2. **Literature Review:**
   A comprehensive review of existing research, technologies, and methodologies related to IoT-based weather monitoring systems is conducted. This includes the exploration of sensors like DHT11 for temperature and humidity measurement,

microcontrollers like ESP32, and cloud-based data storage solutions. The review helps to identify best practices and technologies that can be leveraged in the development of the system.

3. **Requirements Analysis:**
   The functional and non-functional requirements for the weather monitoring system are defined, considering aspects such as real-time data collection, system accuracy, user interface design, scalability, and security. The system should be user-friendly, able to send real-time notifications, and capable of offering context-based suggestions (e.g., health tips, clothing suggestions) based on weather conditions.
4. **System Design:**
   Based on the requirements analysis, the system architecture and design are developed. This includes defining the components (e.g., sensors, microcontrollers, communication modules), data flow, user interface, and communication protocols. The system is designed to ensure efficient data collection, processing, and presentation through an intuitive interface (e.g., mobile or web app).
5. **Prototype Development:**
   A prototype of the weather monitoring system is developed. This involves integrating the DHT11 sensor for temperature and humidity measurements, using the ESP32 for IoT connectivity, and connecting the system to an LED display for real-time data output. Additionally, software for data collection, processing, and user interface development is implemented, and the system is tested in controlled environments.
6. **Evaluation and Testing:**
   The prototype system is tested to evaluate its accuracy, reliability, and performance under various environmental conditions. Testing involves comparing sensor data with actual weather conditions, conducting field trials, and gathering user feedback to ensure the system provides meaningful, accurate suggestions. The system is also evaluated for scalability and adaptability to different environments or user needs.

CHAPTER 5

RESULTS AND DISCUSSION

The developed weather monitoring system successfully collected and displayed real-time temperature and humidity data using the DHT11 sensor. The ESP32 microcontroller enabled seamless data transmission, ensuring minimal delays in updates. During testing, the system's accuracy was consistent with external weather data sources, and it provided timely and relevant suggestions based on the weather conditions, such as hydration recommendations for users and safety tips for children and pets.

The user interface, displayed via a web or mobile app, effectively communicated weather information in a user-friendly format. Additionally, real-time notifications worked as intended, alerting users to significant weather changes and offering precautionary advice. These notifications proved to be particularly useful in preparing users for sudden temperature fluctuations or high humidity levels, promoting better health and safety practices.

However, some limitations were observed, particularly with the DHT11 sensor in environments with rapid temperature fluctuations or high wind speeds, where minor discrepancies were noted. Furthermore, the system's reliance on an internet connection occasionally led to slight delays in data updates during network disruptions. Despite these challenges, the system demonstrated strong scalability and adaptability, making it suitable for broader applications in smart homes, agriculture, and urban weather monitoring.

CHAPTER 6

CONCLUSION AND FUTURE WORK

Conclusion

The development of an IoT-based weather monitoring system marks a significant advancement in personal and environmental data collection. By integrating sensors such as the DHT11 and utilizing the ESP32 microcontroller for IoT capabilities, the system efficiently monitors real-time weather conditions like temperature and humidity. The system's ability to provide valuable suggestions based on weather data contributes to user well-being, with advice on hydration, clothing, and safety for children and pets. This technology offers potential for various applications, from smart homes to agriculture, where accurate, real-time weather information is essential for decision-making.

Future Work

In the future, we aim to improve the accuracy, scalability, and functionality of the weather monitoring system. We plan to explore the use of more advanced sensors that provide better data quality and longer durability, addressing the limitations of the DHT11 sensor in rapidly changing weather conditions. Moreover, enhancing the system’s connectivity through the integration of 5G or low-power wide-area networks (LPWAN) will enable seamless communication and faster data transmission. Additionally, incorporating machine learning models and cloud-based data analytics will allow the system to predict weather patterns more accurately and provide personalized recommendations for users based on historical data and trends.

APPENDIX

SOFTWARE INSTALLATION

Arduino IDE

To run and mount code on the Arduino NANO, we need to first install the Arduino IDE. After running the code successfully, mount it.

Sample code

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include "DHT.h"
#include <WiFi.h>
#include <ESPAsyncWebServer.h>

// WiFi credentials const
char* ssid = "Gokul";
const char* password = "12345678";

// DHT11 Settings
#define DHTPIN 4
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

// LCD
LiquidCrystal_I2C lcd(0x27, 16, 2);

// Web Server
AsyncWebServer server(80);

// LCD update interval
unsigned long previousMillis = 0;
const long lcdInterval = 2000;

void setup() { Serial.begin(115200); dht.begin(); lcd.init(); lcd.backlight(); lcd.setCursor(0,
0); lcd.print(" Weather Station "); lcd.setCursor(0, 1); lcd.print("Starting WiFi...");
delay(2000);
```

```
WiFi.begin(ssid, password); while (WiFi.status() != WL_CONNECTED) { delay(500);
Serial.print("."); }

lcd.clear(); lcd.setCursor(0, 0); lcd.print("Connected to WiFi"); lcd.setCursor(0, 1);
lcd.print(WiFi.localIP()); Serial.println("\nConnected to WiFi"); Serial.print("IP Address: ");
Serial.println(WiFi.localIP());


// API to return sensor data
server.on("/api/data", HTTP_GET, [](AsyncWebServerRequest *request){
  float humidity = dht.readHumidity();
  float temperature = dht.readTemperature();

if (isnan(humidity) || isnan(temperature)) {
  request->send(500, "application/json", "{\"error\":\"Sensor read failed\"}");
  return;
}

String json = "{";
json += "\"temperature\":" + String(temperature, 1) + ",";
json += "\"humidity\":" + String(humidity, 1);
json += "}";
request->send(200, "application/json", json);
});

// Optional: add a simple health/status endpoint
server.on("/api/status", HTTP_GET, [](AsyncWebServerRequest *request){
  String status = "{"status":"OK", "ip":"" + WiFi.localIP().toString() + ""}";
  request->send(200, "application/json", status); });

server.begin(); }

void loop() { unsigned long currentMillis = millis();

if (currentMillis - previousMillis >= lcdInterval) { previousMillis = currentMillis; float
humidity = dht.readHumidity(); float temperature = dht.readTemperature();

lcd.clear();
if (isnan(humidity) || isnan(temperature)) {
  lcd.setCursor(0, 0);
  lcd.print("Sensor Error!");
```

```
} else {
  lcd.setCursor(0, 0);
  lcd.print("Temp:");
  lcd.print(temperature, 1);
  lcd.print((char)223);
  lcd.print("C");

  lcd.setCursor(0, 1);
  lcd.print("Humidity:");
  lcd.print(humidity, 1);
  lcd.print("%");
}
} }
```

CHAPTER 7

REFERENCES

1. A. Jain, et al., "Design and Implementation of IoT-Based Weather Monitoring System for Real-Time Data Collection," *International Journal of Internet of Things and Sensor Networks*, vol. 10, no. 4, pp. 50-60, 2020.

2. M. Sharma, et al., "A Review of IoT-Enabled Weather Monitoring Systems and Applications," *Journal of Smart Technologies*, vol. 15, no. 3, pp. 120-135, 2021.

3. R. Patel, et al., "Smart Weather Monitoring and Prediction Using IoT: A Review," *International Journal of Advanced Research in Computer Science*, vol. 8, no. 5, pp. 180-190, 2022.

4. S. Verma, et al., "IoT-Driven Weather Monitoring System: A Case Study of Environmental Data Collection," *Proceedings of the International Conference on IoT and Smart Cities*, 2021.

5. D. Kumar, et al., "Development of IoT-based Weather Monitoring System for Agriculture Applications," *IEEE Access*, vol. 8, pp. 135224-135234, 2020.