ATMOSAWARE

Mini Project submitted in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING (INTERNET OF THINGS)

Under the esteemed guidance of

Dr K. Srinivas

HOD-IOT, Professor

By

SAKETH B	**21R11A6907**
TEJASWI GVNS	**21R11A6926**
SHIVA TEJA K	**21R11A6931**
SAI VARSHITH Ch	**22R15A6902**

DEPARTMENT OF INTERNET OF THINGS
Accredited by NBA

Geethanjali College of Engineering and Technology
(UGC Autonomous)
(Affiliated to J.N.T.U.H, Approved by AICTE, New Delhi)
Cheeryal (V), Keesara (M), Medchal.Dist.-501 301.
November-2024

Geethanjali College of Engineering and Technology

(UGC Autonomous)

(Affiliated to JNTUH, Approved by AICTE, New Delhi)
Cheeryal (V), Keesara(M), Medchal Dist.-501 301.

DEPARTMENT OF INTERNET OF THINGS

Accredited by NBA

CERTIFICATE

This is to certify that the B Tech Mini Project report entitled **"ATMOSAWARE"** is a Bonafide work done by **Saketh B (21R11A6907), GVNS Tejaswi (21R11A6926), K Shiva Teja(21R11A6931), Ch Sai Varshith(22R15A6902)**, in partial fulfillment of the requirement of the award for the degree of Bachelor of Technology in "**COMPUTER SCIENCE AND ENGINEERING (INTERNET OF THINGS)**" from Jawaharlal Nehru Technological University, Hyderabad during the year 2024-2025.

Internal Guide	**Project Coordinator**	**HOD-IOT**
Dr. K. Srinivas	**Mr. Manohar P**	**Dr. K. Srinivas**
HOD, Professor	**Associate Professor**	**Professor**

External Examiner

Geethanjali College of Engineering & Technology

(UGC Autonomous)

(Affiliated to JNTUH Approved by AICTE, New Delhi)
Cheeryal (V), Keesara(M), Medchal Dist.-501 301.

DEPARTMENT OF INTERNET OF THINGS

Accredited by NBA

DECLARATION BY THE CANDIDATE

We, **Saketh B, GVNS Tejaswi, K Shiva Teja, Ch. Sai Varshith bearing Roll Nos. 21R11A6907, 21R11A6926, 21R11A6931,22R15A6902** hereby declare that the project report entitled **"ATMOSAWARE"** is done under the guidance of **Dr K. SRINIVAS, HOD, Professor,** internal guide, Department of Internet of Things, Geethanjali College of Engineering and Technology, is submitted in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in CSE (INTERNET OF THINGS)**

This is a record of Bonafide work carried out by us in **Geethanjali College of Engineering and Technology** and the results embodied in this project have not been reproduced or copied from any source. The results embodied in this project report have not been submitted to any other University or Institute for the award of any other degree or diploma.

Saketh B(21R11A6907)
GVNS Tejaswi(21R11A6926)
K Shiva Teja(21R11A6931)
Ch Sai Varshith(22R15A6902)
Department Of CSE(IOT),
Geethanjali College of Engineering and Technology

Atmosaware.pdf

ORIGINALITY REPORT

1%	1%	0%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Technological University Of The Philippines Student Paper	<1%
2	Submitted to Notre Dame of Marbel University Student Paper	<1%
3	webthesis.biblio.polito.it Internet Source	<1%
4	Submitted to Savonia Ammattikorkeakoulu Student Paper	<1%
5	Submitted to Sim University Student Paper	<1%
6	Submitted to Universiti Teknologi Petronas Student Paper	<1%
7	www.cisco.com Internet Source	<1%
8	www.process.st Internet Source	<1%
9	www.sidestone.com	

	Internet Source	<1%
10	"International Conference on Signal, Machines, Automation, and Algorithm", Springer Science and Business Media LLC, 2024 Publication	<1%
11	hvac-eng.com Internet Source	<1%
12	repository.ubharajaya.ac.id Internet Source	<1%
13	studyres.com Internet Source	<1%
14	www.x-mol.com Internet Source	<1%
15	www.pcb-hero.com Internet Source	<1%
16	Thara Seesaard, Kamonrat Kamjornkittikoon, Chatchawal Wongchoosuk. "A comprehensive review on advancements in sensors for air pollution applications", Science of The Total Environment, 2024 Publication	<1%

ACKNOWLEDGEMENT

We are greatly indebted to the authorities of Geethanjali College of Engineering and Technology, Cheeryal, Medchal District, for providing us the necessary facilities to successfully carry out this mini project work titled “**ATMOSAWARE**.”.

Firstly, we thank and express our solicit gratitude to our HOD, **Dr K. SRINIVAS**, Professor, IoT department, Geethanjali College of Engineering and Technology, for his invaluable help and support which helped us a lot in successfully completing our mini project.

Secondly, we express our gratitude to **Dr K. SRINIVAS**, HOD, Professor, internal guide, IoT department, Geethanjali College of Engineering and Technology, for his suggestion and encouragement which helped us in the successful completion of our mini project.

We express our heartfelt gratitude to **Mr. P Manohar**, Associate Professor, Project Coordinator IT Department, Geethanjali College of Engineering and Technology, for his valuable guidance and constant support, which greatly contributed to the successful completion of our mini project.

We would like to express our sincere gratitude to our Principal **Dr. S. UDAYA KUMAR** for providing the necessary infrastructure to complete our project.

We convey our gratitude to our Chairman, **Mr. G. RAVINDER REDDY**, for his invaluable support and encouragement for propelling our innovation forward.

Finally, we would like to express our heartfelt thanks to our parents who were verysupportive both financially and mentally and for their encouragement to achieve our set goals.

ABSTRACT

Atmosaware- a fusion of the terms atmosphere and awareness is a project designed to raise awareness of atmospheric conditions using real-world data. The project leverages an IoT- based weather data monitoring system built on the NodeMCU microcontroller and the Blynk platform. This model includes sensors to measure temperature, humidity, rainfall and light intensity along with a Wi-Fi module for seamless connectivity to the Blynk platform. The Blynk app enables users to remotely access and manage their IoT devices, allowing them to monitor weather data from anywhere. The system gathers atmospheric information and sends it to the Blynk platform, providing on-the-spot updates through a mobile app interface. With the NodeMCU microcontroller, sensors are easily connected via a breadboard, making the setup straightforward. The Blynk app also allows users to set custom notifications based on specific weather conditions and view sensor data as it is collected. The project emphasizes ease of use and affordability, enabling individuals and communities to deploy their own weather monitoring solutions. Atmosaware aims to enhance public safety by providing early weather warnings, assisting emergency responses, and supporting disaster preparedness. Additionally, it contributes to agricultural planning and management by optimizing crop-related decisions and pest control strategies. Accurate weather data also improves efficiency in transport logistics and energy management, helping stakeholders make informed choices for resourceutilization. By empowering users with real-time data, Atmosaware fosters a deeper understanding of local atmospheric conditions and promotes proactive decision-making across various sectors.

Keywords: Atmosaware, IoT (Internet of Things), Node MCU, Micro controller, Blynk Platform, Sensors, Atmospheric Conditions.

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

CHAPTER 1: OVERVIEW

1.1 PROBLEM STATEMENT

Weather forecasting remains one of the most complex challenges in meteorology due to the highly dynamic nature of the atmosphere and the multitude of variables influencing both localand global weather patterns. Traditional satellite-based weather observation systems primarily provide generalized atmospheric data over extensive regions, often lacking the resolution needed to capture localized weather variations accurately. This limitation is particularly significant for applications requiring precise meteorological data, such as precision agriculture, disaster risk management, and environmental monitoring.

Conventional weather monitoring systems face significant challenges, primarily their high cost. They rely on advanced sensors and technology, making them expensive to install and maintain. Moreover, many lack real-time data visualization, complicating user interpretation andtimely responses. Automated alert mechanisms for hazardous conditions, such as sudden rainfallor temperature drops, are often missing. Modern localized weather monitoring systems overcomethese issues. They provide real-time data on parameters like temperature, humidity, and soil moisture, offering actionable insights.

Localized weather monitoring systems also promote resilience in communities vulnerableto climate change by providing data that helps in planning and responding to extreme events. Byintegrating machine learning algorithms, these systems can predict future weather patterns with greater accuracy, further enhancing their value. The scalability of such systems ensures that theycan be implemented in various regions, from rural areas to urban centers. Moreover, with the riseof IoT technology, these systems can be easily integrated with other smart devices, creating an interconnected network of environmental sensors. Ultimately, localized weather monitoring contributes to the broader goal of building more adaptive, sustainable communities globally.

Such systems are invaluable in agriculture, helping farmers optimize irrigation, predict yields, and address weather-related challenges. Additionally, they support informed decisions about fertilization and pest control. Beyond agriculture, these systems benefit urban planning, disaster management, and environmental conservation.

1.2 INTRODUCTION

Recent technological innovations have progressively focused on the wireless control and tracking of devices through the internet, enabling continuous communication between connected systems. In this context, an efficient environmental monitoring system is essential for tracking and assessing weather conditions. A smart system is defined by its ability to self-monitor and protect itself, made possible by incorporating sensors, microcontrollers, and various software applications.

Weather monitoring systems play a crucial role in tracking dynamic weather patterns. The data gathered by sensors is used not only for weather reporting but also for observing environmental changes in specific locations. This information proves invaluable in studying the Earth and analyzing how climate and environmental conditions evolve over time. Additionally, the data and insights collected can be applied across various fields such as agriculture, geology, mining, and the development of weather forecasting models. In this project, we have developed a simple weather tracking device that can measure temperature, humidity, light intensity, and rainfall in a specific area. The system is based on NodeMCU, an IoT device that is compatible with Arduino. To program the NodeMCU board, either the Arduino IDE can be used, or an account can be created on the Arduino Cloud Platform. The program code can be written using the Web IDE on the Arduino website and uploaded wirelessly to the registered IoT board. Once the NodeMCU board is connected to the Arduino Cloud service, the code is transmitted over the internet, and the board starts functioning based on the instructions in the code.

The system offers several long-term benefits such as:

- Helping to maintain optimal indoor temperatures.
- Providing early warnings for severe weather, allowing for timely precautions.
- Educating the public on climate change by making weather data accessible.
- Assisting communities in preparing for and adapting to shifting climate conditions.
- Monitoring air quality and tracking pollutant dispersal.
- Ensuring flight safety and operational efficiency by supplying real- time weather data.
- Enhancing route planning and safety for maritime operations.

1.3 OBJECTIVE

The weather monitoring system project is designed as a comprehensive and user-friendly IoT solution that brings advanced weather tracking to everyone. This low-cost device, built using the NodeMCU microcontroller, integrates various sensors to collect critical environmental data such as temperature, humidity, rainfall andlight intensity. By connecting to a cloud service, it uploads the collected information, makingit accessible for real-time monitoring, data analysis, and pattern recognition.

1.3.1 KEY FEATURES

- Cost-Effective and Scalable: The system is purposefully designed to be affordable, using readily available components, which opens up possibilities for widespread use, including educational applications and small-scale agricultural needs. Its modular build allows users to add or remove sensors, making it a highly adaptable platform. Users can start with basic features and gradually expand based on their specific requirements, making this device suitable for both beginners and advanced users.
- Real-Time Data Monitoring and Analysis: By leveraging the cloud, the device offers real-time weather data tracking. Users can view live updates from anywhere, whether they are on-site or remote. The cloud integration enables efficient data storage and analysis, helping users identify long-term patterns and trends in weather conditions, which can be invaluable in sectors like agriculture, disaster response, and urban planning.
- Customizability and Personalization: This project's design is not just about monitoring but also about user empowerment. The device can be easily customized to suit personal needs or specific environmental factors in different locations. For users interested in making further modifications, the system is open to changes in both hardware and software, fostering a learning experience for those new to IoT and electronics.
- User-Friendly and Energy Efficient: With a focus on simplicity, the project ensures that users can assemble, install, and maintain their devices with minimal technical expertise making it accessible for a broad audience. Energy-efficient components like the NodeMCU and low-power sensors ensure that the system can operate for extended periods on minimal power, reducing the need for frequent maintenance.

This weather monitoring system stands out as an innovative, customizable, and low-cost solution that makes weather data collection and analysis accessible for all. By empowering users to build and personalize their devices, it promotes environmental awareness, supports educational initiatives and enables communities to track and respond to climate changes in real time. This innovative approach not only fosters a sense of ownership among users but also encourages collaboration and knowledge sharing. As a result, communities can work together more effectively to address pressing environmental challenges and create sustainable solutions.

1.4 UNIQUENESS OF THE DEVICE

- Affordable and Compact: NodeMCU is budget-friendly with a small form factor, ideal for weather monitoring setups.
- Built-in Wi-Fi for IoT: Easily connects to cloud platforms like Blynk for real-time data access on mobile devices.
- Low Power and Flexible: Supports multiple sensors, uses minimal power, and is compatible with the Arduino IDE for easy programming.
- Portable Design: Its small size allows for deployment in various locations, including remote areas with limited space.
- Scalable and Customizable: Additional sensors (e.g., light, rainfall) can be added, making it suitable for simple to advanced monitoring.
- Real-Time Monitoring: With cloud integration, environmental data can be viewed instantly from anywhere, enhancing accessibility.
- Open-Source and Community Support: NodeMCU is an open-source platform with a large community of developers. This means there are plenty of resources, tutorials, and libraries available, making it easy to troubleshoot and expand your project.
- High Processing Speed: The ESP8266 microcontroller in NodeMCU operates at a clock speed of 80 MHz, providing fast processing power, which is essential for handling multiple sensors and real-time data transmission efficiently.
- Cost-Effective Cloud Options: With compatibility for cloud platforms like ThingSpeak and Firebase, NodeMCU offers affordable ways to store and analyze data, ideal for personal and small-scale projects.

- Easy to Code: NodeMCU supports the Lua scripting language in addition to the Arduino IDE, providing flexibility for developers to choose a language they're comfortable with, whether they're experienced programmers or beginners.
- Low Latency Data Transmission: The built-in Wi-Fi ensures that data can be transmitted to the cloud almost instantaneously, making it suitable for applications that require near real-time updates, such as emergency weather alerts.
- Data Logging and Analysis: With integration into cloud services, NodeMCU can store historical data, allowing users to analyze trends over time, such as temperature changes, humidity patterns, and other environmental metrics.
- Compact and Lightweight: Weighing only a few grams, NodeMCU is highly portable and can be easily mounted or embedded in various enclosures. Its lightweight design is especially advantageous for drone or UAV-based monitoring applications.
- Battery Operable: NodeMCU can run on battery power, making it ideal for remote monitoring systems. With its low-power mode, it can operate for extended periods, especially when paired with energy-efficient sensors and a rechargeable battery.
- Customizable Alerts and Automation: Through cloud platforms, you can set up custom alerts (e.g., SMS, email) based on predefined sensor thresholds, which is useful for weather-critical applications like early warnings for frost, rainfall, or high winds.
- Rapid Prototyping: NodeMCU’s low cost and ease of use make it suitable for iterative design and testing. Users can quickly prototype different weather-monitoring configurations, making it ideal for educational purposes and research.
- Durability and Reliability: Despite its low cost, NodeMCU is highly reliable, capable of running continuously in monitoring setups, especially when placed in weather-resistant enclosures. Its durable design allows it to function in various environmental conditions.
- Integrates with Home Automation Systems: For users interested in smart home applications, NodeMCU can be connected to platforms like Home Assistant and MQTT, enabling it to act as a weather station within a larger automated home setup.
- Compatible with Solar Power: For remote areas, NodeMCU can easily be powered by solar panels due to its low power requirements, ensuring consistent operation even in off-grid locations.

CHAPTER 2: SYSTEM ANALYSIS

2.1 LITERARTURE REVIEW

Weather forecasting is one of the most challenging areas in meteorology due to the highly dynamic nature of the atmosphere and the numerous factors influencing local and global weather patterns. Traditional satellite weather reporting systems primarily provide general atmospheric conditions over large areas, which often lack the specificity required to capture localized variations in weather. This limitation can be particularly impactful in applications requiring precise weather data, such as agriculture, disaster management, and environmental monitoring. One significant drawback of conventional weather monitoring systems is their cost. These systems often require high-end sensors and advanced technology,which can be expensive to purchase, set up, and maintain. Furthermore, many of these systems lack accessible, real-time data visualization, making it difficult for users to interpret and respond to the information effectively. Another limitation is that most conventional weather systems do not include an automated alert mechanism, which could provide immediate warnings in case of abnormal or potentially hazardous weather conditions, such as sudden temperature drops or unexpected rainfall.

In contrast, a modern, localized weather monitoring system can address these gaps effectively. For instance, real-time data on weather characteristics such as temperature, humidity and light intensity can be collected and displayed, offering users a clear and immediate understanding of current environmental conditions. This kind of system is particularly useful in agriculture, where weather monitoring plays a crucial role.

With up-to-date information about rainfall, evaporation rates, and soil moisture levels, farmers can make informed decisions about irrigation timing and water usage. This information can helpin predicting crop yields, optimizing irrigation schedules, and planning for weather-related challenges. A low-cost, customizable weather monitoring system equipped with automatedalerting features and real-time data visualization not only offers a practical solution but alsodemocratizes access to essential environmental information, empowering individuals and communities to make more informed, proactive decisions. This approach contributes to increased environmental awareness and promotes sustainable practices, allowing users to respond more effectively to changes in their local environment.

2.2 EXISTING SYSTEM

Conventional weather stations largely depend on standalone sensors to collect key meteorological data, such as temperature, humidity, wind speed, and atmospheric pressure, sometimes also monitoring additional factors like rainfall or solar radiation. These sensors are usually connected to a centralized data logger or collection system that aggregates information for later analysis. However, these traditional setups often lack real-time remote monitoring capabilities, which restricts data accessibility and limits timely decision-making. Data retrieval typically relies on manual access to physical memory, such as SD cards or internal storage in the data logger, which requires personnel to visit the station periodically to collect and transfer data. This dependence on physical access can lead to data gaps, particularly if weather, logistical challenges, or the station's remote location impede retrieval. By the time data is gathered, critical environmental changes may have already occurred, making it challenging for sectors like agriculture, aviation, or emergency services that depend on real-time data for responsive action.

Additionally, manual data handling increases the potential for human error, such as mistakes during recording or file transfer, which can compromise the accuracy of analysis and lead to misinformed decisions. The cost and logistics of maintaining these weather stations can be considerable, with regular calibration, battery replacements, and upkeep of data loggers required. In remote locations, this high maintenance burden can add up, especially since trained personnel are typically needed for these tasks. Conventional setups also lack scalability and customization, making it difficult to modify or expand the sensor arrays to monitor new environmental parameters without substantial effort and expense. Most rely on fixed power sources, limiting deployment flexibility, particularly in remote areas without stable power options.

Furthermore, conventional weather stations lack integration with automation systems, so users must manually check data, and there are few, if any, options for automated alerts or threshold-based reporting. Minimal connectivity with IoT and cloud platforms further limits these stations, as they can’t leverage advanced digital capabilities for real-time monitoring, automated data processing, or dynamic analysis. Overall, the manual processes and limited accessibility inherent in conventional weather stations reduce the efficiency and utility of the data they provide.

2.3 LIMITATIONS

Satellite and radar technologies, commonly employed by national meteorological services, facilitate large-scale data collection. However, these systems often necessitate specialized and costly equipment, rendering them less accessible for smaller applications or localized weather monitoring efforts. Although some contemporary systems have begun to integrate IoT technology, they are often characterized by high costs and complexity. These advanced systems utilize a diverse array of sensors and proprietary software for data collection, analysis, and reporting. Furthermore, they may not provide the flexibility required for specific applications or seamless integration with cloud-based services, which constrains their usability in sectors that demand customizable and scalable solutions.

- Lack of Real-Time Data: Traditional systems frequently fail to deliver real-time weather information, relying on manual data collection processes that can lead to delayed updates and slower responses to environmental changes.
- Limited Remote Access: Many conventional systems do not support remote monitoring, necessitating physical presence at the station to access data.
- High Cost and Complexity: Modern systems incorporating satellite and radar technologies, as well as advanced IoT setups, can be prohibitively expensive and technically intricate, limiting accessibility for smaller organizations or specific applications.
- Manual Intervention: Traditional weather stations require manual input and local storage, increasing the risk of data errors and constraining the potential for system automation.
- Difficult Integration: These systems often struggle to integrate effectively with cloud-based platforms, complicating the use of collected data in contemporary applications, such as AI-driven forecasting models or large-scale data analytics.

In addition to traditional weather monitoring systems, services such as Google offer weather updates via online platforms and mobile applications. While these services provide general weather updates via online platforms and mobile applications.

2.4 PROPOSED SYSTEM

Traditional setups typically require extensive infrastructure, incorporating specialized sensors for measuring atmospheric pressure, wind speed, and other precise meteorological parameters. This reliance on advanced equipment not only increases costs but also necessitates regular maintenance and skilled personnel to ensure accurate data collection. Additionally, many conventional systems depend on complex wired data transmission methods or intricate RF-based communication setups, complicating operational deployment and limiting flexibility. These limitations render traditional weather stations prohibitively expensive and impractical for localized or community-based monitoring, especially in remote or resource-constrained areas. This has led to a growing interest in alternative solutions, such as low-cost sensor networks and mobile weather stations. These innovative approaches can provide real-time data while being more accessible and adaptable to the needs of various communities.

In contrast, the proposed weather monitoring solution, built upon the NodeMCU platform and integrated withDHT11, rainfall, and LDR sensors, offers an optimized and cost-effective alternative.The NodeMCU microcontroller, featuring built-in Wi-Fi capabilities, facilitates real-time data transfer without the need for additional network infrastructure. The DHT11 sensor provides reliable measurements of temperature and humidity, which, although less precise than high-end meteorological instruments, sufficiently meet the requirements for applications in agriculture, environmental monitoring, and educational initiatives. Additionally, the rainfall sensor offers vital precipitation data, delivering real-time insightsinto current weather conditions without the costs associated with sophisticated rain- gaugingequipment. The incorporation of an LDR sensor further enhances the system by measuring light intensity, a crucial parameter often neglected in conventional setups but essential for solar energy monitoring, urban planning, and agricultural forecasting. By tracking daylight levels and cloud cover, the LDR sensor contributes to a comprehensive understanding of environmental conditions with minimal hardware requirements. This straightforwardness renders it a compelling choice for a range of applications, spanning from minor projects to extensive industrial implementations. Furthermore, the gathered data can be seamlessly incorporated into current systems to improve decision-making and operational efficiency.

2.5 ADVANTAGES OF PROPOSED SYSTEM

Weather monitoring systems play a critical role in understanding and responding to environmental changes, but traditional systems often come with high costs, complexity and limited accessibility. By leveraging IoT technology, the integration of affordable components like low-cost sensors and microcontrollers has revolutionized the approach to environmental monitoring. These modern systems provide a more efficient, scalable, and user-friendly solution, bridging the gap between advanced weathr monitoring and widespread accessibility for various users, including small-scale farmers, educational institutions, and community organizations**.**

- Cost-Effectiveness: The combination of low-cost sensors and a microcontroller significantly reduces the overall investment compared to traditional systems making itaccessible for educational institutions, small farms, and community organizations.
- Real-Time Data Access: The IoT capabilities of the NodeMCU enable real-time monitoring and data transmission, allowing users to access current information from anywhere, facilitating timely decision-making.
- Ease of Deployment: With minimal setup requirements and a wireless data transmission approach, the system can be quickly deployed across various environments, including urban, rural, and remote locations.
- User-Friendly Interface: Data can be easily integrated into mobile and web applications, offering users intuitive interfaces for monitoring and analyzing weather conditions.
- Environmental Sustainability: The low power consumption of the sensors and the ability to operate on renewable energy sources, such as solar power, promote sustainability inmonitoring operations.
- Data Analytics and Predictive Insights: With cloud integration, data from the NodeMCU-based system can be stored, analyzed, and visualized over time, enabling users to identify trends, make predictive assessments, and optimize responses to changing environmental conditions, such as early warnings for adverse weather.

CHAPTER 3: REQUIREMENT ANALYSIS

3.1 FEASIBLITY STUDY

Atmosaware is a NodeMCU-based weather monitoring system provides a practical, low-cost solution for capturing localized weather data, useful in applications like agriculture, urban planning, and environmental monitoring. The NodeMCU microcontroller, equipped with built-in Wi-Fi, connects to affordable sensors like DHT11 for temperature and humidity, an LDR for light intensity, and a rainfall sensor. This combination of sensors offers a comprehensive view of environmental conditions, and data is uploaded to Blynk Cloud, where users can access real-time information via a mobile app or web interface. This system's simplicity allows for easy setup, with minimal maintenance required apart from occasional sensor calibration. Compared to traditional weather stations, this setup is highly cost-effective, typically costing around $10 to $20, making it accessible for individual or community use. Furthermore, the system's reliance on Wi-Fi ensures efficient data transmission, though alternative power options, such as solar, can increase deployment flexibility for remote locations. The system has potential for further customization, allowing users to add sensors like soil moisture or wind speed to tailor the setup to specific needs. Alerts for abnormal conditions, such as sudden temperature changes or heavy rainfall, can also be integrated, improving response times to environmental changes. This flexibility and expandability make device not only feasible but also highly adaptable to diverse needs.

3.2 FUNCTIONAL REQUIREMENTS

These functional requirements aim to ensure the NodeMCU weather monitoring system is accurate, reliable and easy to use with provisions of customization and future expandability.

- Data Collection: The system will collect data on humidity, light intensity and rainfall using connected sensors (DHT11, LDR, and rainfall sensor). The system should support additional sensors if required, like soil moisture and air quality sensors, for expanded functionality.
- Data Processing: NodeMCU will read sensor values at regular intervals and process this data for upload. The system will convert sensor data into readable values (e.g., temperature in Celsius, light in lumens) for user interpretation.

- Data Transmission: The NodeMCU will transmit collected data over Wi-Fi to a cloud platform, such as Blynk Cloud. It should automatically reconnect to the network if Wi-Fi connectivity is lost.
- Data Storage: The cloud platform will store historical data for analysis and visualization. The system should allow users to access historical weather data for specific time periods.
- Real-Time Monitoring: The system enables real-time monitoring, where users can view live data on the Blynk app or web dashboard.
- Alerts and Notifications: The system will send notifications or alerts in cases of abnormal readings, such as high temperatures or heavy rainfall. Users should be able to customize alert thresholds according to specific needs.
- Data Visualization: The cloud platform will provide data visualization tools like graphs and charts for easy data interpretation. The interface will be user-friendly, displaying real-time and historical data clearly.
- Power Management: The system will function efficiently with low power requirements, with options to use battery or solar power for remote installations.
- User Control and Configuration: Users will be able to configure the sampling intervals, sensor thresholds, and data display settings on the app. The system should support remote configuration for easy customization without physical access.
- System Calibration and Maintenance: The system allows periodic calibration to maintain sensor accuracy. Notifications will be available for maintenance needs, such as sensor replacement or recalibration.

3.3 NON-FUNCTIONAL REQUIREMENTS

The non-functional requirements for a weather monitoring system using NodeMCU define the system's quality attributes and constraints

- Performance: The system will capture and upload sensor data at regular intervals (e.g., every minute), without significant delays. Data will be processed and displayedon the cloud platform within seconds for near real-time updates.
- Reliability: The system also operates reliably in varying environmental conditions (e.g., temperature changes, humidity, dust) with minimal downtime. It will automatically reconnect to Wi-Fi if the connection is interrupted and continue transmitting data without manual intervention.
- Scalability: This system will support additional sensors if more environmental data is needed, without major reconfiguration. The cloud platform can handle large data load if additional systems are deployed in multiple locations.
- Availability: The system will maintain high availability, ideally 24/7 operation, with minimal maintenance. The cloud platform will be continuously available to ensure data access at any time.
- Usability: This system's user interface (on Blynk or other platforms) will be intuitive, allowing users to easily navigate, view data, and set alerts. It will be accessible for users with basic technical knowledge.
- Security: Data transmitted from the NodeMCU to the cloud will be encrypted to prevent unauthorized access. The system implements secure authentication to ensure only authorized users can access or configure the device.
- Maintainability: The system design allows easy replacement or calibration of sensors as needed. The software and firmware will be easily updatable for bug fixesand improvements.
- Portability: This system will be lightweight and compact for ease of installation in different locations. It is adaptable for both indoor and outdoor use with minimal modification.
- Energy Efficiency: This system consumes minimal power, with the potential for solar power or battery operation in remote locations.

3.4 SOFTWARE REQUIREMENTS

The software requirements for a Atmosaware include essential tools, libraries, and platforms to ensure data collection, processing, and visualization.

1. **Arduino IDE**: It is a free, open-source software platform designed for programming Arduino boards and compatible microcontrollers like NodeMCU and ESP32. With a user-friendly interface, it allows users to write, edit, and upload code in the C/C++ language directly to microcontroller boards. It provides a code editor with syntax highlighting, a library manager to add functionality for various sensors and modules, and a Serial Monitor for real-time debugging and data monitoring. It supports a wide range of boards, the IDE makes it easy to select the correct board and port, compile code, and upload it via USB for immediate execution. This platform also supports third-party libraries, making it easy to extend functionality and integrate advanced features like cloud connectivity and machine learning. With regular updates,the IDE ensures compatibility with the latest hardware and provides enhanced features for improved programming efficiency. Its drag-and-drop programming tools and support for plug-and-play sensors further simplify the development process. Combined with its extensive online community, the Arduino IDE continues to drive innovation and creativity in the maker and engineering communities worldwide. It is available for Windows, macOS, and Linux, the Arduino IDE is accessible to a broad audience, making it popular in IoT, robotics, and other DIY electronics projects. Its extensive online community and resources, including tutorials and example code, making it particularly useful for both beginners and experienced developers looking to experiment and innovate.
2. **Firmware:** The ESP8266/ESP32 core for Arduino. It is essential firmware that integratesESP8266 and ESP32 microcontrollers with the Arduino IDE, allowing users to program these boards as if they were standard Arduino devices. This core includes boarddefinitions, libraries, and example sketches specific to ESP8266 and ESP32, enabling seamless communication between the Arduino IDE and the microcontroller. By installing this core, the IDE can recognize the unique architecture of ESP8266 and ESP32, which are Wi-Fi-enabled microcontrollers widely used in IoT applications. This firmware core provides essential functions like Wi-Fi libraries (ESP8266WiFi.h or WIFI for ESP32), enabling easy network connections and IoT functionality.

It also includes libraries for GPIO control, analog and digital pin operations, and communication protocols like SPI and I2C. This core makes it easy to use advanced ESP8266/ESP32 features such as deep sleep mode for energy efficiency, over-the-air (OTA) updates for remote firmware uploading, and various encryption protocols for secure data transfer. Additionally, users can select the appropriate ESP board from the Arduino IDE’s Tools menu, configure settings like flash memory size, and customize upload speed. By installing the ESP8266/ESP32 core, developers can leverage the full power of these microcontrollers while maintaining the simplicity of the Arduino programming environment, making it ideal for building scalable, Wi-Fi enabled projects in a user-friendly way.

3. **Libraries**: For successful execution we need to install the below mentioned libraries:
   - **DHT Library**: The DHT library is a powerful and versatile tool designed to interface with DHT11 and DHT22 sensors, both of them measures temperature and humidity in the surrounding environment. These sensors are widely used in a variety of applications due to their affordability, ease of use, and reliability. The library abstracts the complexities of communication with the sensor by handling low-level details such as timing and data parsing, allowing users to interact with the sensor using simple and easy-to-understand code.With the DHT library, developers can efficiently obtain temperature and humidity values with just a few lines of code, streamlining the process of integrating environmental data into projects. The library allows users to set up functions for periodic data readings, ensuring consistent updates for real-time monitoring. It also includes robust error-handling mechanisms, allowing the system to handle scenarios such as sensor disconnections or inaccurate readings, which helps improve reliability and user experience. The library supports multiple units of measurement, including Celsius and Fahrenheit for temperature and percentage for humidity, making it flexible for different use cases. The DHT library is widely used in Internet of Things (IoT) projects, home automation systems, and environmental monitoring applications.
   - **Blynk Library**: The Blynk library is an essential tool that enables seamless connectivity between NodeMCU and Blynk Cloud, making it ideal for Internet of Things (IoT) applications. This library simplifies the process of data transmission and remote monitoring by providing an easy-to-use platform for developers to connect their devices to the cloud. Once the NodeMCU is connected to Blynk

Cloud, real-time data can be displayed on the Blynk app or web dashboard, which features interactive widgets like gauges, graphs, and buttons. These widgets allow users to visualize sensor data in an intuitive manner, making it especially useful for applications that require continuous monitoring and data interpretation, such as weather stations, smart homes, and industrial IoT systems.The Blynk library utilizes virtual pins, a feature that greatly enhances its flexibility and ease of use. Virtual pins enable users to send data from sensors (such as temperature and humidity readings) and receive commands to control actuators (like LEDs, motors, or relays) without the need to hardcode specific physical pins or engage in complex data handling. This abstraction layer simplifies code and allows for more efficient device management, making it possible to control multiple devices simultaneously with minimal programming effort.

4. **Blynk App:** It is a Cloud application and IoT-focused platform that provides a user-friendly interface for real-time data visualization, storage, and device control, accessible through both the Blynk mobile app and web dashboard. It offers dynamic widgets such as graphs, gauges, and buttons for displaying data like temperature and humidity, and supports customizable layouts for a personalized user experience. Blynk Cloud also enables remote device control, allowing users to manage and automate connected devices from anywhere, making it ideal for monitoring and controlling systems like weather stations. Historical data is automatically logged, which allows for trend analysis and exporting for further insights. Each Blynk project is secured with a unique authentication token, ensuring only authorized devices can connect, while data encryption protects the integrity of transmitted information. Additionally, Blynk is highly scalable, supporting multi-device integration, cross-platform access on iOS, Android, and web, and integration with protocols like HTTP and MQTT for flexibility. The platform supports event-based triggers and automated responses, allowing users to set specific actions based on sensor readings, such as sending alerts during critical weather conditions. With its offline-to-online synchronization, Blynk ensures data reliability even during intermittent internet connectivity. The platform integrates seamlessly with popular microcontrollers like NodeMCU and ESP32, making it accessible for both beginners and advanced developers. Regular updates and an active online community enhance its usability by providing support and new features. Blynk's ability to handle diverse IoT applications, from smart homes to industrial automation, makes it a versatile and comprehensive solution for modern IoT needs.

3.5 HARDWARE REQUIREMENTS

The hardware requirements include essential components like DHT11 sensor, Rainfall sensor, LDR (Light Dependent Resistor) and NodeMCU (ESP2866) for sensing, processing, and transmitting environmental data.

1. **DHT11 sensor:** The DHT11 sensor is a low-cost digital sensor used to measure temperature and humidity in various environments. It is popular for weather monitoring and IoT applications due to its simplicity, small size, and ease of use. The sensor contains a thermistor to measure temperature and a capacitive humidity sensor, both integrated with an 8-bit microcontroller that processes and outputs the data as a digital signal, making it straightforward to read with microcontrollers like NodeMCU. The DHT11 sensor has a 3-pin layout, although often only three of these pins are necessary for operation, as it typically comes with VCC, Data, and Ground pins.

- **VCC (Pin 1):** This is the power supply pin and should be connected to a voltage source between 3.3V and 5.5V, compatible with common microcontrollers like the NodeMCU and Arduino.
- **Data (Pin 2):** This pin carries the sensor's digital output signal. It sends temperature and humidity data to the microcontroller using a single-wire protocol. The data pin requires a 10kΩ pull-up resistor connected to VCC to ensure stable data transmission, particularly in longer data lines.
- **Ground (Pin 3):** The ground pin connects to the ground (GND) of the power source, completing the circuit and enabling stable operation.

The DHT11's pin arrangement is designed to facilitate straightforward connections to microcontrollers, with its single-wire communication from the Data pin simplifying data handling. Each reading session sends data as a 40-bit digital output, which contains humidity and temperature data along with a checksum to verify data integrity. This layout and communication protocol make it user-friendly for integration into IoT and DIY projects. This sensor operates in a temperature range of 0–50°C with a tolerance of ±2°C and a humidity range of 20–90% with ±5% accuracy.

It sends data in a 40-bit format that includes both temperature and humidity values. Operating at 3.3V to 5.5V, it can be directly powered by microcontrollers like the NodeMCU or Arduino, with minimal power consumption around 2.5mA when active and just 100-150µA in idle mode. One limitation is its sampling rate, which isabout once per second (1Hz), making it unsuitable for high-speed or rapid environmental changes. However, its simplicity and low-cost appeal to hobbyists, educators, and beginners. It can also be used effectively indoors or in sheltered outdoor environments, as long as it is kept dry and within its operating range.The DHT11's reliable performance, affordability, and compatibility with various microcontroller platforms make it a staple sensor for learning and prototyping in IoT and environmental projects.

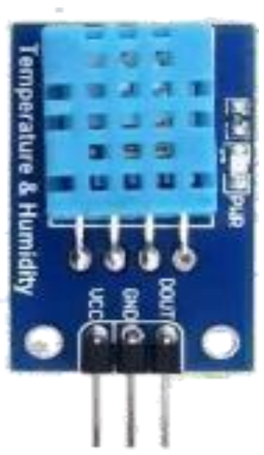


Fig 3.5.1 DHT11 SENSOR

2. **Rain Sensor:** A rainfall sensor is a device used to detect and measure the presence of rain, making it useful in weather monitoring systems, irrigation management, and environmental sensing projects. The most commonly used rainfall sensor module includes a rain sensor board and a control module. The rain sensor board has a series of conductive traces that form a grid; when raindrops fall on the board, they create connections between the traces, altering the resistance and allowing the sensor to detect moisture levels. The control module translates this into a digital or analog signal that can be read by microcontrollers like NodeMCU or Arduino.

The rain sensor module typically has three pins:

- **VCC:** This pin is connected to a power source, generally requiring 3.3V to 5V, making it compatible with most microcontrollers.
- **GND:** The ground pin connects to the GND of the power source, ensuring acomplete and stable circuit.
- **D0 (Digital Output):** This pin outputs a high (1) or low (0) signal based on rain presence. When rain is detected, the sensor outputs a low signal, otherwise, it outputs a high signal. This digital output allows for simple rain detection.

Some rainfall sensors also have an A0 (Analog Output) pin, which provides a variable voltage corresponding to the intensity of the rainfall. This analog output can be useful for applications that need to measure the amount of rain rather than simply detecting its presence. The module often includes a potentiometer to adjust the sensitivity, enabling fine-tuning based on environmental conditions. The rainfall sensor is ideal for applications like automated irrigation, where it can signal when to stop watering based on natural rainfall, or for weather monitoring systems that log rainfall data for analysis.

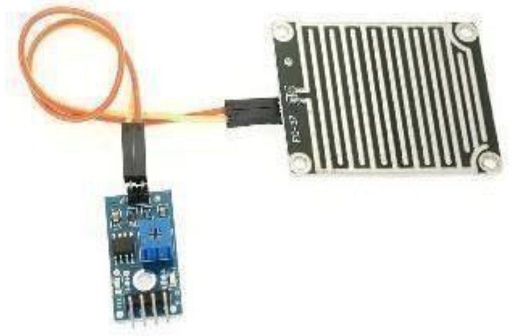


Fig 3.5.2 RAIN SENSOR

3. **LDR (Light Dependent Resistor) Sensor:** It is also known as a photoresistor, is used to detect and measure the intensity of light in the surrounding environment. It is widely utilized in applications like automatic lighting systems, ambient light monitoring, and weather stations to track day/night cycles or light levels. The LDR’s resistance changes inversely with light intensity, its resistance decreases as light intensity increases and increases as light intensity decreases.

This characteristic makes it ideal for monitoring varying light conditions. An LDRsensor module typically includes the following pins:

- **VCC (Power):** This pin connects to the power source, usually 3.3V to 5V, makingit compatible with microcontrollers such as NodeMCU and Arduino.
- **GND (Ground):** The ground pin connects to the GND of the power supply,completing the circuit for stable operation.
- **D0 (Digital Output):** This pin provides a high (1) or low (0) digital output depending on light intensity. When light intensity is below a set threshold, the sensor outputs a high signal, and when it is above the threshold, it outputs a low signal. The threshold can be adjusted using an onboard potentiometer, which allows users to control the light level at which the sensor triggers.

The LDR sensor's versatility, simple design, and responsiveness make it suitable for various projects. For example, it can trigger lights to turn on in low-light conditions or log light intensity data for environmental monitoring purposes. Its straightforward interface and adjustable sensitivity allow easy integration and
fine-tuning in IoT and automation projects. When combined with other sensors, such as a rainfall sensor, LDRs can enhance the functionality of automated systems enabling them to respond not only to rain but also to varying light conditions.

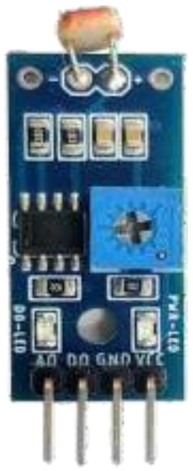


Fig 3.5.3 LDR SENSOR

4. **NodeMCU (ESP2866):** The NodeMCU is a popular, low-cost, open-source IoT development board based on the ESP8266 or ESP32 Wi-Fi-enabled microcontroller. It combines the features of the microcontroller and Wi-Fi module, allowing for seamless connection to the internet, making it ideal for IoT projects such as weather monitoring, home automation, and data logging. The board is programmable via the Arduino IDE or Lua scripting language, with extensive community support for IoT applications. The NodeMCU has multiple pins that support various functionalities, including GPIO, ADC, PWM. Here is an overview of the essential pins and their functions:

Power Pins:

- **VIN:** This pin accepts an input voltage of 5V (e.g., from USB or an external powersource) to power the board.
- **3V3:** Provides a 3.3V output from the onboard voltage regulator, which can be usedto power sensors or other low-voltage components.
- **GND:** Ground pins for completing circuits; multiple GND pins are available for easeof connectivity.

Digital GPIO Pins: The NodeMCU has several General-Purpose Input/Output (GPIO) pins, labeled D0 to D8, which are used to interface with various digital devices like LEDs, relays, and sensors. GPIO pins can be programmed for input or output, making them versatile for handling various tasks. Some GPIOs (D1 and D2) can also be configured for I2C communication.

Analog Pin (A0): A0 is the analog-to-digital conversion (ADC) pin, which reads analog signals, such as from sensors like LDRs, providing a value between 0 and 1023 corresponding to voltage levels between 0V and 1V. Since the input voltage range for A0 is limited, voltage dividers are often used if higher voltages need to be read.

Communication Pins:

- **UART (RX, TX):** These pins are used for serial communication, allowing the NodeMCU to communicate with computers, other microcontrollers, or serial devices. RX (D9) is for receiving data, while TX (D10) is for transmitting data.

- **SPI:** Pins such as D5, D6, D7, and D8 can be configured for SPI communication, useful for devices requiring fast data exchange, like certain displays or SD card modules.
- **I2C (SDA, SCL):** GPIO pins D1 (SCL) and D2 (SDA) can be used for I2C communication, enabling data exchange with I2C-compatible devices like LCDs, accelerometers, or temperature sensors.
- **PWM Pins:** WM (Pulse Width Modulation) can be applied to most GPIO pins on the NodeMCU, allowing for functions like LED brightness control or servo motor position adjustment. This compact, feature-rich board provides the functionality required for connected applications, with easy programming via the Arduino IDE or Lua scripting, appealing to both beginners and experienced developers in IoT.

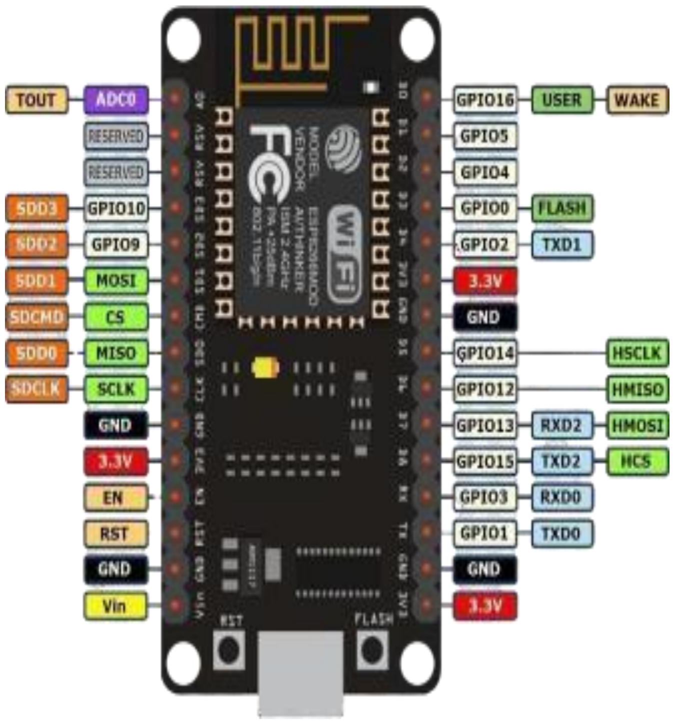


Fig 3.5.4 NODEMCU (ESP2866)

5. **Jumper Wires:** These are small electrical wires with connectors on each end, used to make temporary or semi-permanent connections on breadboards and between electronic components. They are essential tools for prototyping and testing circuits, as they allow quick and flexible connections without the need for soldering, which is particularly useful in experimental setups, hobby projects, and educational contexts. Jumper wires come in three main types based on their connector ends:

- **Male-to-Male:** These have exposed metal pins on each end, making them ideal for connecting components on a breadboard or plugging into female headers on development boards like Arduino or NodeMCU.
- **Female-to-Female:** These have female connectors (or sockets) on both ends and are useful for connecting pins on different modules or development boards that have male header pins.
- **Male-to-Female:** One end has a male pin, and the other end has a female socket, making them versatile for connecting boards to modules, sensors, or other components.

Jumper wires are usually available in different lengths and colors, allowing for organized, color-coded connections in complex circuits. They are commonly used with breadboards, which have rows and columns of contacts that allow multiple jumper wires to be connected, creating easy-to-change circuits. Jumper wires are also insulated with plastic to prevent accidental short circuits and ensure safety while handling circuits. Overall, jumper wires are fundamental for rapid prototyping in electronics, making them highly valued in learning environments and DIY electronics projects.In addition to their use in prototyping, jumper wires are often employed in debugging circuits, allowing for quick adjustments and testing of different configurations. Their flexible nature also makes them ideal for connecting components that need to be moved or adjusted frequently. Jumper wires are compatible with a wide range of electronic components, including sensors, motors, LEDs, and more, making them an essential tool for experimenting with new ideas. Their simplicity and effectiveness have made them a staple in educational kits for teaching basic electronics. As electronic projects become increasingly complex, jumper wires remain a reliable and indispensable tool in the development process.

Fig 3.5.5 JUMPER WIRES

6. **BREADBOARD:** A breadboard is a reusable and versatile tool used in electronics for prototyping and testing circuit designs without the need for soldering. Typically made from plastic, a breadboard consists of a grid of interconnected holes that facilitate the easy arrangement of electronic components like resistors, capacitors, integrated circuits (ICs), and various sensors. The main features of a breadboard include:

- **Interconnected Strips:** The breadboard has horizontal and vertical strips of conductive metal (usually copper) that run beneath the plastic surface. These strips allow for the connection of multiple components across the board. The rows are generally used for power and ground connections, while the columns are used for signal connections.
- **Power Rails:** Most breadboards have two long vertical rows on either side, known as power rails, typically marked with red (for positive voltage) and blue (for ground). These rails make it easy to distribute power to various components without requiring multiple connections to a single power source.
- **Modular Design:** The breadboard’s design allows for easy insertion and removal of components, making it ideal for experimentation. Users can quickly change circuit configurations, test different setups, and troubleshoot without the permanence of soldering.

- **Multiple Sizes:** Breadboards come in various sizes, from small, compact boards suitable for simple circuits to larger ones that can accommodate more complex designs. This adaptability allows users to work on projects of different scales and complexities.
- **No Soldering Required:** One of the most significant advantages of using a breadboard is that it eliminates the need for soldering. This feature is particularly beneficial for beginners and hobbyists, as it simplifies the learning process and allows for quick iterations in circuit design.

Breadboards are essential tools in educational settings, allowing students and hobbyists to experiment with electronics and learn about circuit design and functionality. They are widely used in prototyping phases of product development by engineers and inventors, facilitating a hands-on approach to testing and refining electronic concepts. Overall, the breadboard's ease of use, flexibility, and reusability make it an invaluable asset in the field of electronics.

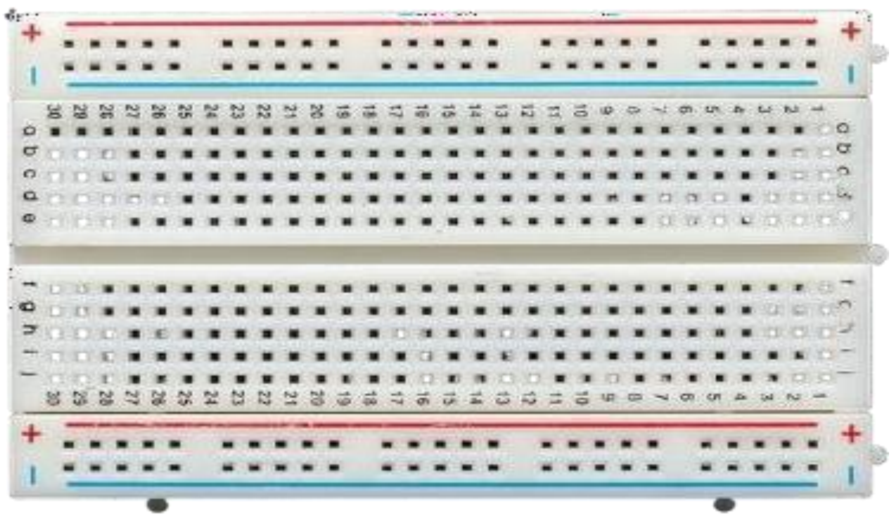


Fig 3.5.6 BREADBOARD

3.6 EXPECTED HURDLES

Implementing a weather monitoring system with NodeMCU offers numerous benefits but also poses significant challenges in power management, connectivity, and environmental protection. The NodeMCU’s reliance on Wi-Fi for data transmission limits its range and requiresa stable internet connection, which can be unreliable in remote or rural areas. This dependency may lead to delays or loss of real-time data, reducing system efficiency. Additionally, managingpower consumption is critical, as continuous operation of sensors and the NodeMCU can quicklydeplete power sources, especially in off-grid settings. Efficient power solutions like solar panelsor low-power modes are essential to maintain uninterrupted performance. Outdoor deploymentsfurther face challenges like waterproofing and protecting sensors from harsh weather conditionssuch as rain, dust, and extreme temperatures, which can degrade the sensors and impact data accuracy.

Security and data handling also present challenges. The NodeMCU's limited processing and storage capabilities make it difficult to manage large datasets from multiple sensors,especially without reliable cloud integration. While platforms like Blynk can address this, they depend on consistent internet access. Furthermore, the NodeMCU's constrained resources for encryption leave the system vulnerable to data breaches and unauthorized access if not properlysecured. Addressing these issues requires a robust system design that incorporates durable components, efficient power management, and secure communication protocols. With proper planning, calibration, and maintenance, a NodeMCU-based weather monitoring system can provide an effective, scalable, and cost-efficient solution for a variety of applications.

CHAPTER 4: SYSTEM DESIGN

4.1 DESIGN APPROACH

To design a weather monitoring system using NodeMCU, start by defining the system requirements, including the weather parameters you wish to monitor, such as temperature, humidity, rainfall, and light intensity. Select hardware components, including the NodeMCU microcontroller, DHT11 sensor for temperature and humidity, a rainfall sensor, and an LDR (Light Dependent Resistor) for measuring light intensity. Create a circuit by connecting the sensors to the appropriate pins on the NodeMCU, ensuring proper wiring according to the specifications of each component. Use the Arduino IDE or Platform IOT for software development, incorporating necessary libraries like DHT11, ESP8266WiFi, and Blynk. Write code to read sensor data at regular intervals, sending it to the Blynk Cloud for real-time monitoring via the Blynk app. After testing and calibrating the system for accurate readings, deploy it in a suitable enclosure at an optimal location. Regular maintenance and firmware updates will ensure the system remains functional and accurate.

4.2 BLOCK DIAGRAM

The below block diagram will define the input and output devices of the system

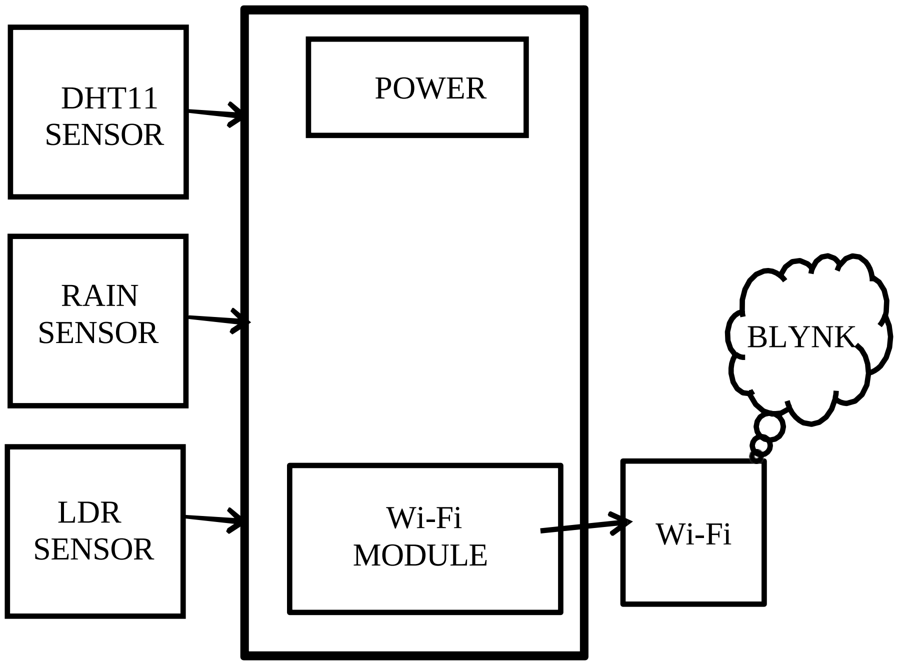


Fig 4.2.1 BLOCK DIAGRAM OF ATMOSAWARE

4.2 SYSTEM ARCHITECTURE

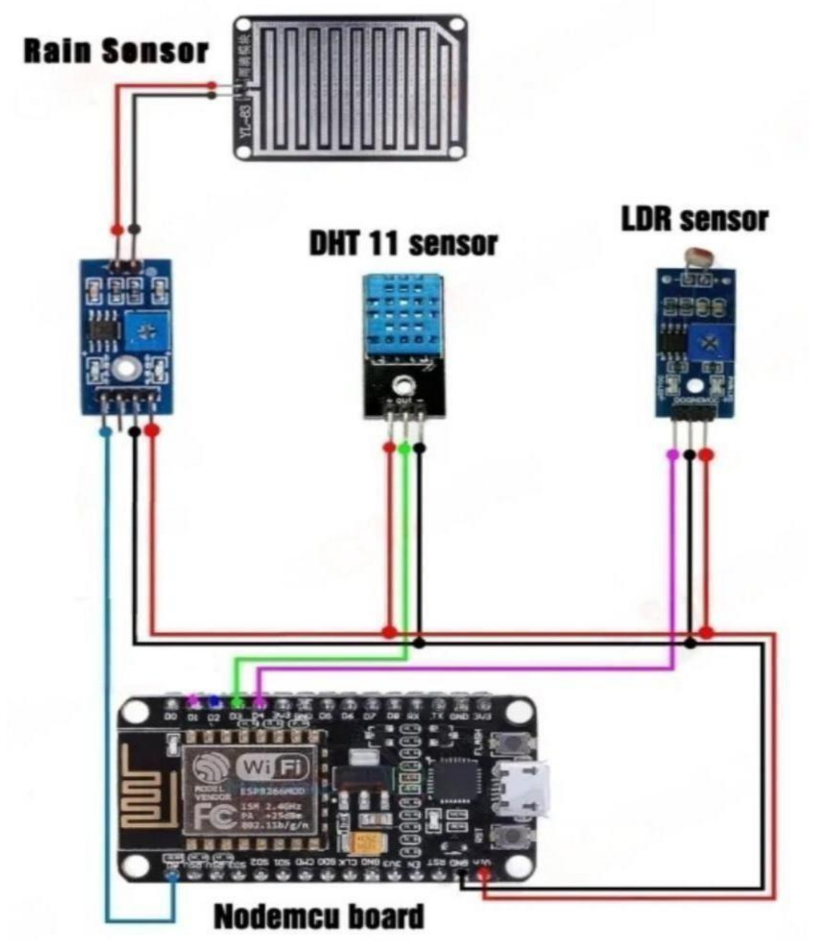


Fig 4.3.1 CIRCUIT OF ATMOSAWARE

DATA PIN	A0
VCC	Vin
GND	GND

Table 4.3.1 CONNECTIONS OF RAIN SENSOR

DATA PIN	D3
VCC	Vin
GND	GND

Table 4.3.2 CONNECTIONS OF DHT11

DATA PIN	D4
VCC	Vin
GND	GND

Table 4.3.3 CONNECTIONS OF LDR

4.3 DETAIL DESIGN

The ATMOSAWARE is a weather monitoring system using NodeMCU is designed as a low-cost, efficient, and scalable solution for monitoring environmental changes in real time. This system uses a combination of sensors, including the DHT11 for temperature and humidity, a rainfall sensor for precipitation data, and an LDR (Light Dependent Resistor) for measuring light intensity. By leveraging the NodeMCU’s Wi-Fi capabilities, the systemis able to send collected data to the Blynk Cloud platform for remote monitoring and data analysis. This design allows users to monitor multiple weather parameters without requiring advanced hardware or technical knowledge, making it both user-friendly and accessible.

In this setup, the NodeMCU serves as the main controller. It connects to the DHT11 sensor through the D3 pin, which reads the temperature and humidity levels. The DHT11 is known for its ease of use and reasonable accuracy, suitable for non-commercial applications. The rainfall sensor is connected to the A0 pin and measures rainfall intensity by detecting changes in resistance as water comes into contact with the sensor’s surface. Similarly, the LDR is connected to the D4 pin to detect variations in light intensity, which gives an indication of the ambient lighting conditions. Power connections are straightforward,each sensor’s VCC and GND are connected to a shared power source and ground, making the circuit layout compact and reducing wiring complexity.

Once the data is gathered, the NodeMCU sends it to the Blynk Cloud using Wi-Fi. The integration with Blynk provides a graphical interface through a mobile app, where users can view real-time data or analyze historical trends in each parameter. The Blynk platform also allows for easy customization, so users can set thresholds and receive alerts when certain conditions are met, such as high humidity or low light. This capability enhances thesystem’s functionality, particularly for applications in remote weather stations, agriculture,or home-based environmental monitoring. The collected data from the weather monitoringsystem is presented on a mobile screen using the Blynk platform, which provides a user- friendly interface for real- time monitoring. The Blynk app displays temperature, humidity, rainfall, and light intensity readings, making it easy to view and analyze the data from anywhere with an internet connection.

The mobile display is customizable, allowing users to set up notifications or alerts for specific conditions, such as high humidity or low light levels, ensuring that they stay informed about environmental changes without needing additional hardware. The system's modularity and open-source nature make it ideal for experimentation and customization. Additionally, the NodeMCU’s small size and low power requirements make it suitable for deployment in various environments. To keep the cost low and make it DIY-friendly, off- the-shelf components and widely available sensors are used, allowing users to assemble thedevice with minimal technical skills. This weather monitoring system provides a foundationthat can be modified and scaled according to user needs, balancing simplicity with functional depth. Its reliance on Wi-Fi and Blynk for data management not only allows remote monitoring but also paves the way for potential expansion, such as integration with smart home systems or advanced data analytics. This design meets essential criteria as it is cost-effective, user-friendly, customizable, and capable of providing reliable environmental data, making it a practical choice for individuals, hobbyists, and small-scalemonitoring projects. Ultimately, this design fosters a deeper connection to environmental awareness by empowering users to understand their immediate surroundings better, supporting both individual and community efforts to track and respond to environmental changes. Its robust yet flexible structure makes it suitable for a wide range of applications, from educational projects to sustainable living practices, proving that effective environmental monitoring is achievable for everyone.

Furthermore, the Atmosware weather monitoring system embodies a forward-thinking approach to technology by promoting environmental consciousness and sustainability. Its affordability and ease of use encourage widespread adoption, making it accessible to students, hobbyists, and researchers alike. By equipping users with the tools to monitor andunderstand their local weather patterns, this system has the potential to inspire community- driven initiatives aimed at addressing climate challenges. As technology continues to evolve, Atmosaware lays the groundwork for future innovations, serving as a stepping stonetoward smarter, more interconnected environmental monitoring solutions.

4.4 FLOWCHART FOR PROPOSED SYSTEM

The flowchart provided outlines the steps to establish a functional connection between multiple sensors and the NodeMCU microcontroller, enabling continuous environmental data collection, processing, and wireless transmission to the Blynk Cloud platform. This efficient integration allows users to monitor real-time atmospheric conditions remotely and interactively through the Blynk mobile application. In this setup, each sensor plays a crucial role in capturing specific environmental parameters. The DHT11 sensor measures temperature and humidity, providing critical information on atmospheric comfort levels and weather conditions. The rainfall sensor gauges precipitation levels, aiding in detecting rainfall intensity and patterns, which is particularly valuable for agricultural and disaster management purposes. Meanwhile, the Light Dependent Resistor (LDR) assesses ambient light intensity, offering insights into solar exposure or detecting changes in day-night cycles. The NodeMCU microcontroller serves as the central processing unit for this system. It initializes each sensor, collects raw data, processes it, and uses its inbuilt Wi-Fi module to establish a reliable internet connection. Once connected, the NodeMCU securely transmits the real-time environmental data to the Blynk Cloud platform, where it is stored and processed for visualization. Users can access this data on the Blynk mobile app, which features an intuitive and interactive interface. This interface enables users to monitor various parameters such as temperature, humidity, rainfall, and light levels, providing a comprehensive view of the environmental conditions. The system operates in a continuous loop, ensuring real-time data updates for seamless monitoring.

By repeatedly collecting and transmitting sensor data, the mobile app display remains consistently refreshed with the latest readings. This ensures that users receive accurate and up-to-date information, enhancing their ability to respond proactively to environmental changes. Additionally, the Blynk app allows users to customize alerts for specific conditions, such as temperature thresholds, heavy rainfall warnings, or changes in light intensity, further increasing the system's functionality. This adaptability makes the solution suitable for a wide range of applications, including agricultural planning, urban monitoring, environmental research, and personal weather tracking. By combining cost-effectiveness, user-friendliness, and real-time capabilities, the system offers a powerful tool for understanding and responding to atmospheric changes.

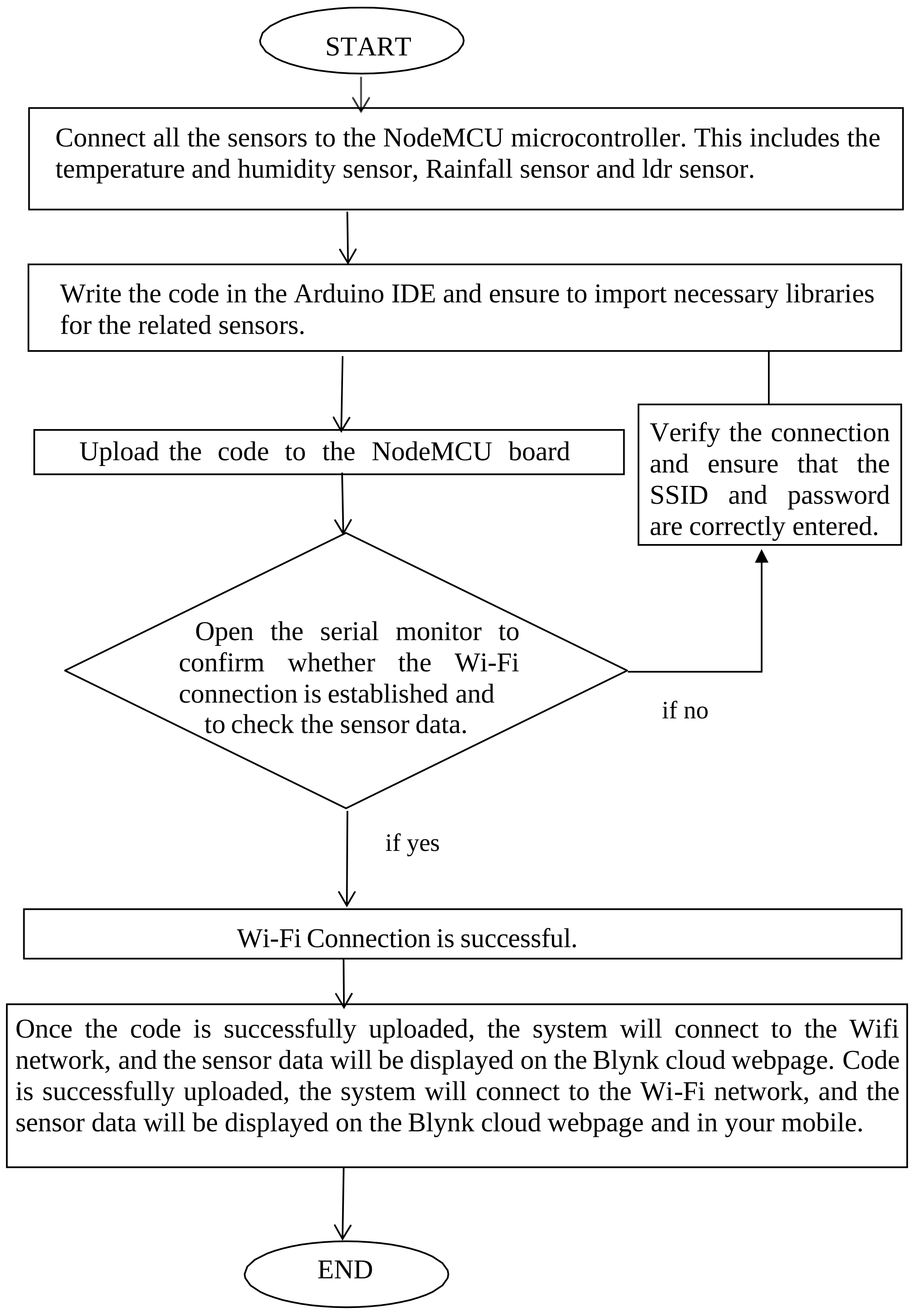

START
Connect all the sensors to the NodeMCU microcontroller. This includes the temperature and humidity sensor, Rainfall sensor and ldr sensor.
Write the code in the Arduino IDE and ensure to import necessary libraries for the related sensors.
Upload the code to the NodeMCU board
Verify the connection and ensure that the SSID and password are correctly entered.
Open the serial monitor to confirm whether the Wi-Fi connection is established and to check the sensor data.
if no
if yes
Wi-Fi Connection is successful.
Once the code is successfully uploaded, the system will connect to the Wifi network, and the sensor data will be displayed on the Blynk cloud webpage. Code is successfully uploaded, the system will connect to the Wi-Fi network, and the sensor data will be displayed on the Blynk cloud webpage and in your mobile.
END

Fig 4.5.1 FLOWCHART

4.5 METHODOLOGY

The Agile methodology is highly suitable for developing a Atmosaware due to its iterative, flexible, and user-centric approach. In Agile, the project is broken down into sprints, each focused on specific features such as integrating sensors, establishing Wi-Fi connectivity, and implementing data visualization. This allows developers to test and refine each component continuously, ensuring that issues like sensor accuracy, power consumption, and connectivity are addressed early. The iterative nature of Agile also supports frequent feedback and adaptation, which is essential for IoT projects where real-world conditions may reveal unforeseen challenges. As a result, developers can adjust the system as needed, adding features or modifying components based on testing or user feedback. Agile's adaptability enables the weather monitoring system to evolve over time, with phased deployment allowing for incremental improvements and maintenance, such as firmware updates and sensor calibration. Overall, Agile promotes collaboration, reduces risk, and ensures that the system meets both technical and user requirements while staying responsive to emerging needs or changes.

Agile methodology offers significant advantages in the development Atmosaware a weather monitoring device using NodeMCU by incorporating early testing, it helps identifypotential risks such as unreliable Wi-Fi, sensor inaccuracies, or excessive power consumption, significantly reducing the chances of failure in the final product. The iterative nature of Agile allows for continuous improvements, enabling developers to refine and optimize system performance based on testing feedback and evolving needs. Furthermore, Agile focuses on prioritizing high-value features in early sprints, ensuring efficient resource allocation, which is particularly important for hardware-constrained projects like those using NodeMCU. This approach not only optimizes time and costs but also ensures that the system evolves in line with real-world conditions and user demands.

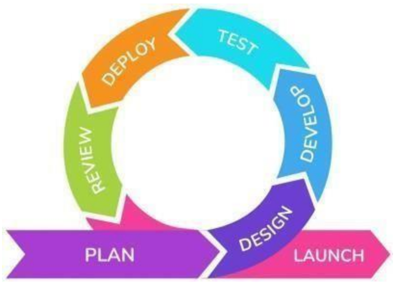


Fig 4.6.1 AGILE PROCESS

CHAPTER 5: IMPLEMENTATION AND TESTING

5.1 SOURCE CODE

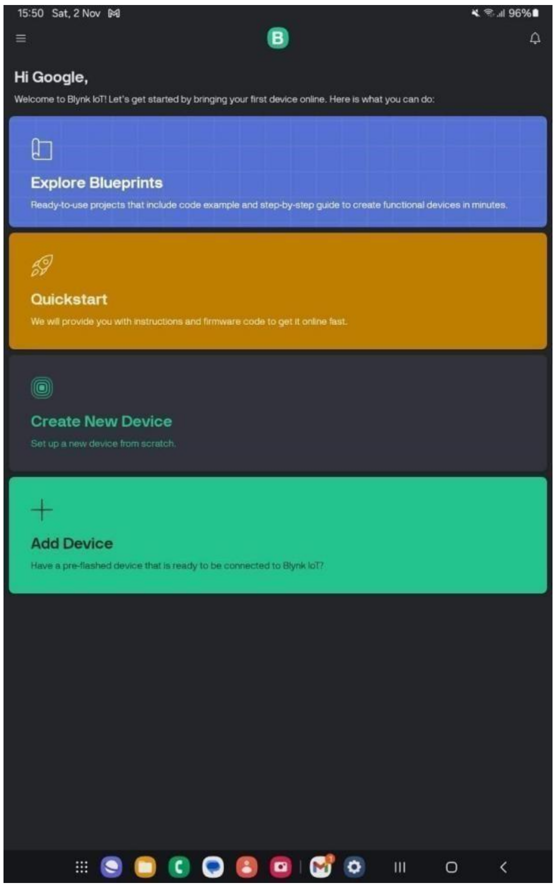
```
#define BLYNK_TEMPLATE_ID "TMPL3ppFS6u8p"
#define BLYNK_TEMPLATE_NAME "Weather
Monitoring"#include <LiquidCrystal_I2C.h>
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include<BlynkSimpleEsp8266.h
>#include <DHT.h>
#include <stdlib.h>
LiquidCrystal_I2C lcd (0x27,
16,2);
DHT dht(D3, DHT11); //(sensor pin, sensor type) BlynkTimer timer;
char auth[] = "L2dkreOMbmPyOMoCX3od3keANl2G-xtQ"; //Enter Auth code send by
Blynkchar ssid[] = "OPPO A16"; //Enter your WIFI Name
char pass[] = "12345678"; //Enter your WIFI
Passwordvoid weather()
{
   float h,t;
   h=dht.readHumidity();
   t=dht.readTemperature();
   int r =analogRead(A0);
   Serial.println(r);
   bool l;
   l=digitalRead(D4
   );
   if (isnan(h) || isnan(t))
   {
        Serial.println("Failed to read from DHT
        sensor!");return;
   }
```

```
    Blynk.virtualWrite(V0, t); //V0 is for
    TemperatureBlynk.virtualWrite(V1, h); //V1 is
    for Humidity Blynk.virtualWrite(V2, r);  //V2 is
    for Rainfall
    if (l == 0)
    {
      WidgetLED led1(V4); led1.on();
      lcd.setCursor(9, 1); lcd.print("L
      :");lcd.print("High");
      lcd.print(" ");
    }
    else if (l == 1)
    {
      WidgetLED
      led1(V4);led1.off();
      lcd.setCursor(9, 1);
      lcd.print("L :");
      lcd.print("Low");
      lcd.print(" ");
    }
    lcd.setCursor(0,
    0);lcd.print("T :");
    lcd.print(t);
    lcd.setCursor(0,
    1);lcd.print("H
    :"); lcd.print(h);
    lcd.setCursor(9,
    0);lcd.print("R
    :"); lcd.print(r);
    lcd.print(" ");
  }
```

```
 void setup()
{
    Serial.begin(115200); // See the connection status in Serial Monitor
    lcd.init();
    lcd.backlight();
    Blynk.begin(auth, ssid, pass);
    dht.begin();
    timer.setInterval(10L,
    weather);
}
void loop()
{
    Blynk.run(); // Initiates Blynk
    timer.run(); // Initiates
    SimpleTimer
}
```

5.2 BLYNK SETUP

1. Download and install the Blynk app on your phone. After, sign up for this app using your email address. Then, click the “Quicktart” button



15:50 Sat, 2 Nov

96%

Hi Google,

Welcome to Blynk IoT! Let's get started by bringing your first device online. Here is what you can do:

Explore Blueprints

Ready-to-use projects that include code example and step-by-step guide to create functional devices in minutes.

Quickstart

We will provide you with instructions and firmware code to get it online fast.

Create New Device

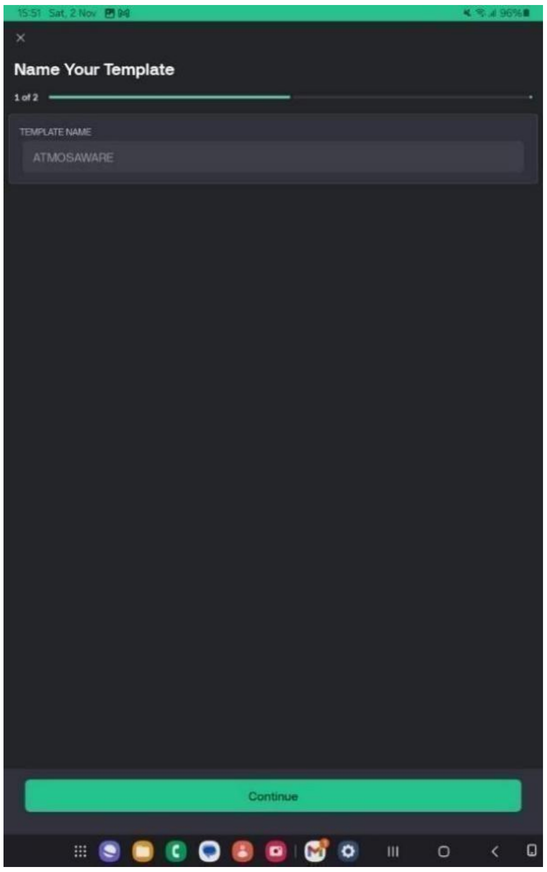
Set up a new device from scratch.

Add Device

Have a pre-flashed device that is ready to be connected to Blynk IoT?

Fig 5.2.1 CREATE TEMPLET

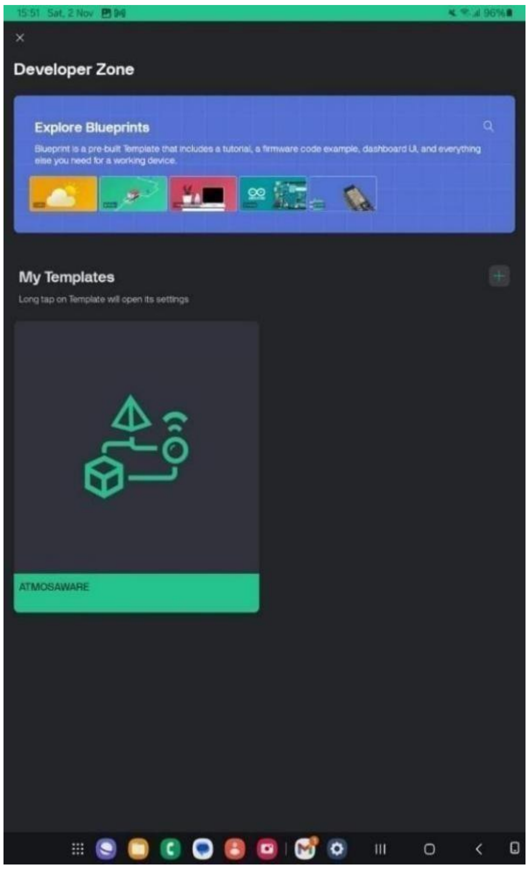
2. Assign a Project Name. I have entered “ATMOSAWARE” as the project name, then click on the "Continue" button to proceed.



TEMPLATE NAME
ATMOSAWARE

Fig 5.2.2 NAME THE TEMPLET

3. The project template will now be displayed on the screen



Blueprint is a pre-built Template that includes a tutorial, a firmware code example, dashboard UI, and everything else you need for a working device.

Long tap on Template will open its settings

Fig 5.2.3 TEMPLET VIEW

4. To add widgets to the interface, click the "+" icon at the bottom of the screen. Then, add three Gauge widgets and one LED widget.

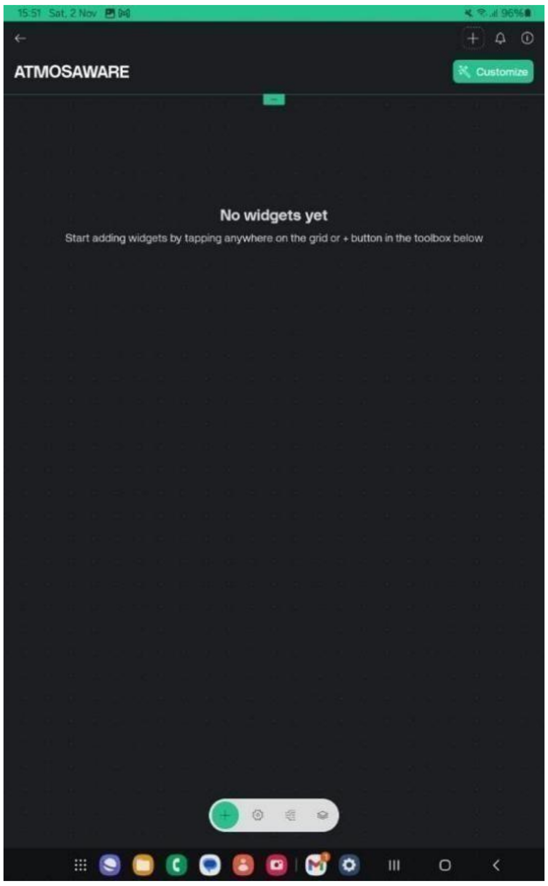
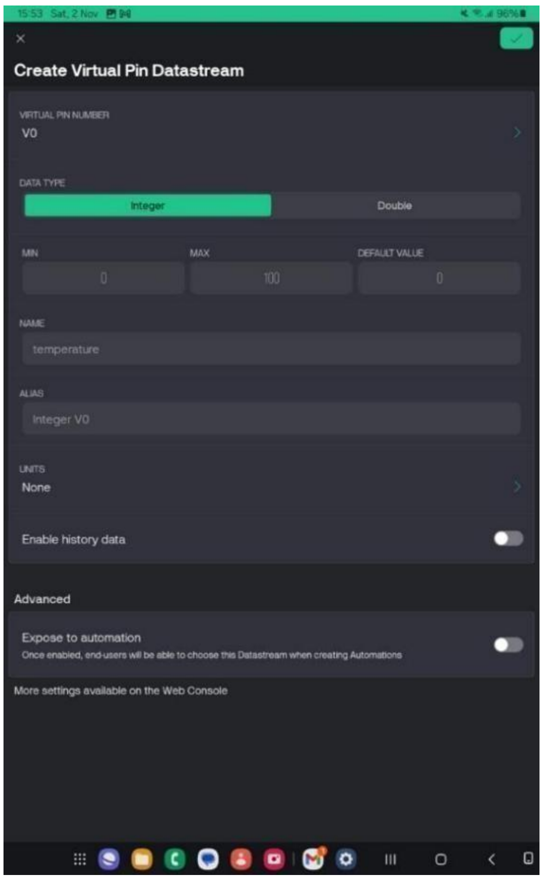

ATMOSAWARE
Customize
No widgets yet
Start adding widgets by tapping anywhere on the grid or + button in the toolbox below

Fig 5.2.4. WIDGETS CREATIONS

5. Set up each widget individually by naming the Gauge widgets as Temperature, Humidity, and Rainfall, respectively. Assign the PINs to V0, V1, and V2, and configure the display values to range from 0 to 100.



15:53 Sat, 2 Nov

96%

Create Virtual Pin Datastream

VIRTUAL PIN NUMBER
V0

DATA TYPE
Integer | Double

MIN	MAX	DEFAULT VALUE
0	100	0

NAME
temperature

ALIAS
Integer V0

UNITS
None

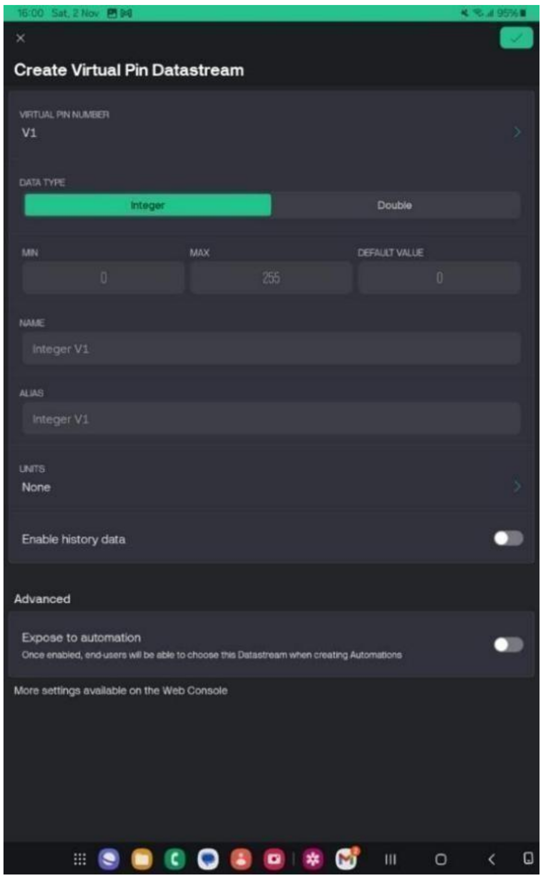
Enable history data

Advanced

Expose to automation
Once enabled, end-users will be able to choose this Datastream when creating Automations

More settings available on the Web Console

Fig 5.2.5 TEMPERATURE WIDGET



16:00 Sat, 2 Nov

95%

Create Virtual Pin Datastream

VIRTUAL PIN NUMBER
V1

DATA TYPE
Integer | Double

MIN	MAX	DEFAULT VALUE
0	255	0

NAME
Integer V1

ALIAS
Integer V1

UNITS
None

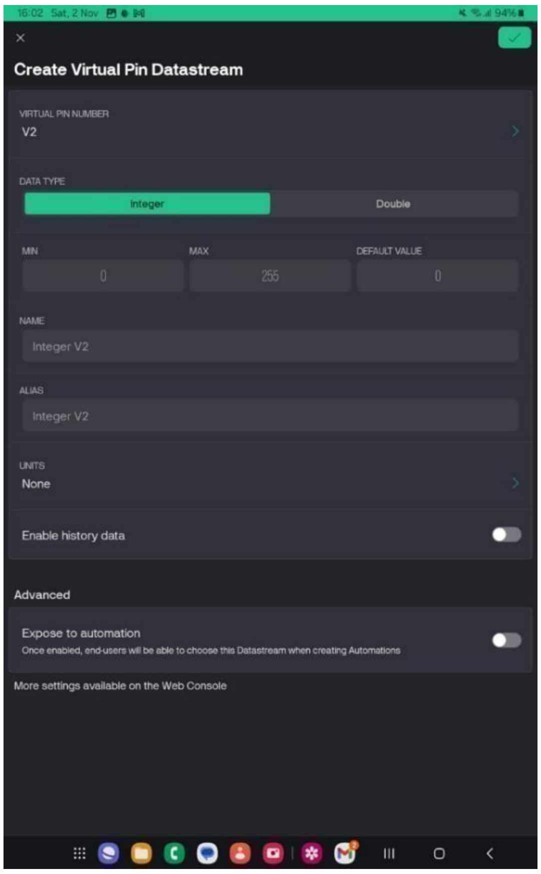
Enable history data

Advanced

Expose to automation
Once enabled, end-users will be able to choose this Datastream when creating Automations

More settings available on the Web Console

Fig 5.2.6 HUMIDITY WIDGET



16:02 Sat, 2 Nov 94%

Create Virtual Pin Datastream

VIRTUAL PIN NUMBER
V2

DATA TYPE
Integer | Double

MIN	MAX	DEFAULT VALUE
0	255	0

NAME
Integer V2

ALIAS
Integer V2

UNITS
None

Enable history data

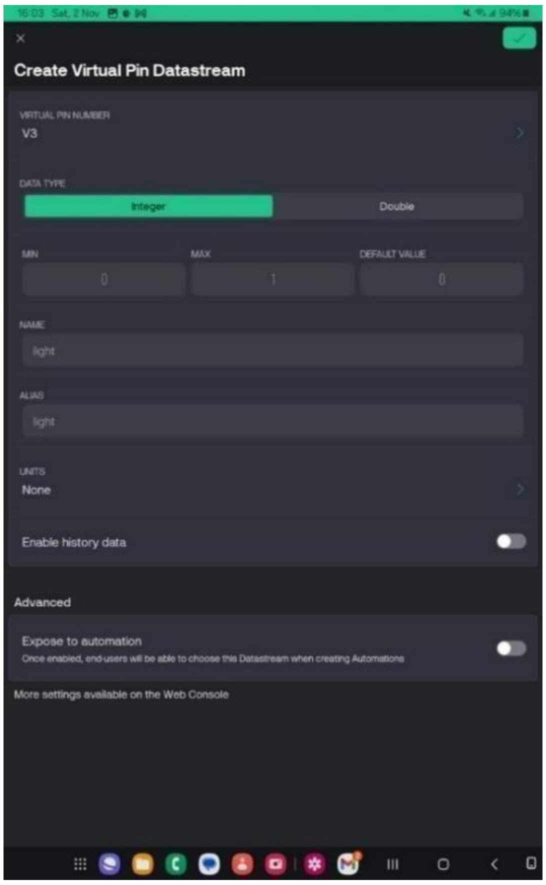
Advanced

Expose to automation
Once enabled, end-users will be able to choose this Datastream when creating Automations

More settings available on the Web Console

Fig 5.2.5 RAINFALL WIDGET CREATION

6. Select the LED widget and assign the PIN to V3.



Create Virtual Pin Datastream

VIRTUAL PIN NUMBER
V3

DATA TYPE
[✓ Integer] Double

MIN	MAX	DEFAULT VALUE
0	1	0

NAME
light

ALIAS
light

UNITS
None

Enable history data

Advanced

Expose to automation
Once enabled, end-users will be able to choose this Datastream when creating Automations

More settings available on the Web Console

Fig 5.2.6 LDR WIDGET DISPLAY

7. Lastly, customize the widget as desired. Your Blynk app is now ready for the project.

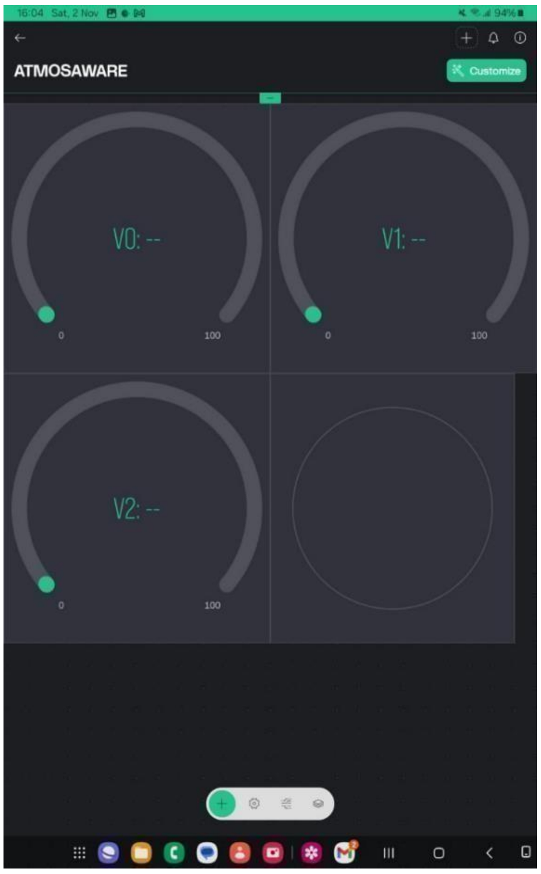


Fig 5.2.7 OVERALL DISPLAY OF WIDGETS

5.3 CODE MODULE

1. **Firstly, library files are included.**

```
#include <LiquidCrystal_I2C.h>
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include
<BlynkSimpleEsp8266.h>
#include <DHT.h>
#include <stdlib.h>
```

2. **Next, objects are created for these libraries**

```
LiquidCrystal_I2C lcd (0x27, 16, 2);
DHT dht(D3, DHT11); //(sensor pin,sensor
type)BlynkTimer timer;
```

3. **Now let’s include the Blynk app Auth token and Wi-Fi connection details.**

```
#define BLYNK_TEMPLATE_ID "TMPL3ppFS6u8p"
#define BLYNK_TEMPLATE_NAME "Weather
Monitoring"char auth[] =
"L2dkreOMbmPyOMoCX3od3keANl2G-xtQ"; char ssid[] =
"OPPO A16";
char pass[] = "12345678";
```

4. **This is the main function of this program**

```
void weather()
{
   float h = dht.readHumidity();
   float t =
   dht.readTemperature();int
   r=analogRead(A0);
   Serial.println(r);
   bool l = digitalRead(D4);
```

```
 if (isnan(h) || isnan(t))
 {
      Serial.println("Failed to read from DHT
      sensor!");return;
   }
 Blynk.virtualWrite(V0, t); //V0 is for
 TemperatureBlynk.virtualWrite(V1, h); //V1 is
 for Humidity Blynk.virtualWrite(V2, r); //V2 is
 for Rainfall
 if (l == 0)
 {
    WidgetLED
    led1(V4);led1.on();
    lcd.setCursor(9, 1);
    lcd.print("L :");
      lcd.print("High");
      lcd.print(" ");
 }
 else if (l == 1)
{
 WidgetLED
 led1(V4);led1.off();
 lcd.setCursor(9, 1);
 lcd.print("L :");
 lcd.print("Low");
 lcd.print(" ");
}
 lcd.setCursor(0,
 0);lcd.print("T :");
 lcd.print(t);
 lcd.setCursor(0,
 1);lcd.print("H
 :");
```

```
    lcd.print(h);
    lcd.setCursor(9,
    0);lcd.print("R
    :"); lcd.print(r);
    lcd.print(" ");
}
```

5. In the setup function,

```
void setup()
{
     Serial.begin(115200);
     lcd.init();
     lcd.backlight();
     Blynk.begin(auth, ssidpass);
     dht.begin();
     timer.setInterval(10L, weather);
}
```

6. In the lo op function, the Blynk library is run

```
void loop()
{
    Blynk.run(); // Initiates Blynk
    timer.run();  // Initiates
    SimpleTimer
}
```

5.4 LANGUAGE, TOOLS AND TECHNOLOGIES

For developing Atmosaware, we used Embedded C Arduino IDE, and IoT technology to enable the creation of connected systems that collect, process, and communicate data in realtime, making them essential for building efficient and user-friendly IoT applications. In a weather monitoring system, these technologies work in tandem to create a reliable solutionthat delivers real-time environmental insights and provides remote data access, ultimately enhancing decision-making and responsiveness to weather conditions.

- **Embedded C language:** It is a specialized extension of the C programming language designed for programming embedded systems. It is widely used in applications that require direct interaction with hardware, such as microcontrollers, sensors, and actuators. Unlike standard C, Embedded C provides additional features that allow developers to interface directly with hardware components, access low-level memory, and manipulate registers. This makes it suitable for resource-constrained devices, whereefficiency in terms of memory usage and processing power is crucial. Embedded C is known for its simplicity, portability, and ability to generate highly optimized code, making it ideal for time-critical tasks. It also supports real-time operations, which is essential in embedded systems that need to respond to external events promptly.Embedded C is often used in applications like robotics, IoT devices, automotivesystems, and weather monitoring, where interacting with sensors, collecting data, and communicating with other systems are common requirements.
- **IOT Technology:** IoT (Internet of Things) technology refers to the network of interconnected physical devices that communicate and exchange data with each other over the internet or other communication networks. These devices, often embedded with sensors, actuators, and software, can collect data from their environment, processit, and share it in real time. The core of IoT technology is its ability to enable automationand remote control, allowing devices to perform tasks without human intervention. IoTdevices can range from simple sensors (e.g., temperature, humidity, motion sensors) tocomplex systems like smart home appliances, industrial machines, and vehicles.

- **Arduino IDE:** Arduino IDE (Integrated Development Environment) is an open-source platform used for programming and uploading code to Arduino microcontroller boards.It is widely used by hobbyists, engineers, and educators to create embedded systems andIoT projects due to its simplicity and versatility. The Arduino IDE allows users to write code in Arduino C/C++ and provides a user-friendly interface for compiling and uploading programs to the board. The key features of the Arduino IDE include cross- platform compatibility, as it runs on Windows, macOS, and Linux, making it accessible to a wide range of users. It offers a simplified coding environment, providing a straightforward text editor with syntax highlighting and auto-completion, which helps beginners easily get started with programming.

The IDE also supports library integration, offering a rich collection of pre-built libraries that simplify the process of interfacing with various sensors, modules, and communication protocols. This feature enables users to quickly connect hardware like sensors, motors, and displays. Additionally, the Serial Monitor tool allows users to monitor real-time data sent from the Arduino board, making debugging and testing much easier. Finally, the IDE supports code compilation and uploading, enabling users to compile their programs and upload them directly to the Arduino board via USB, streamlining the development process. These features collectively make the Arduino IDE a powerful and accessible tool for both beginners and advanced developers working on embedded systems and IoT projects.

The Arduino IDE supports a variety of microcontroller boards, including popular ones like Arduino Uno, Arduino Nano, ESP8266, ESP32, and many others. This flexibility allows it to be used in a wide range of applications, including automation, robotics, and IoT, where devices need to collect data, process it, and communicate with other systems. The simplicity and community-driven support make Arduino IDE an excellent tool for both beginners and experienced developers working on embedded systems and IoT projects.

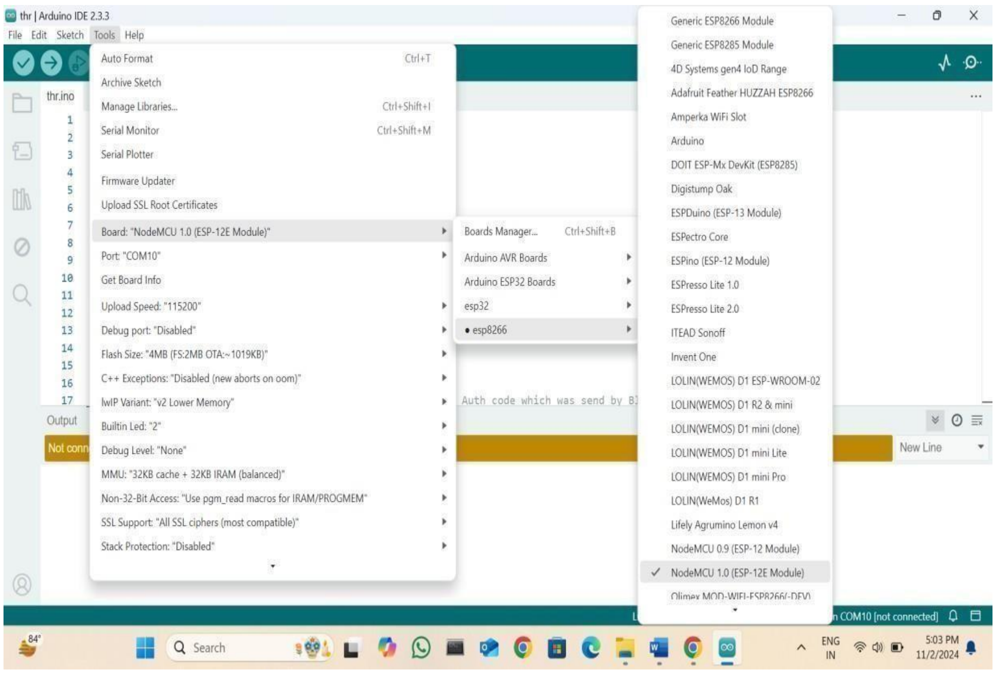
5.4.1 STEPS FOR ARUDINO IDE

To write and upload the embedded code to the NodeMCU board using the Arduino IDE, follow these steps:

1. **Write the Embedded Code:** Begin by writing the required code in the Arduino IDE, ensuring it is properly structured to interface with the sensors and components connected to the NodeMCU board.

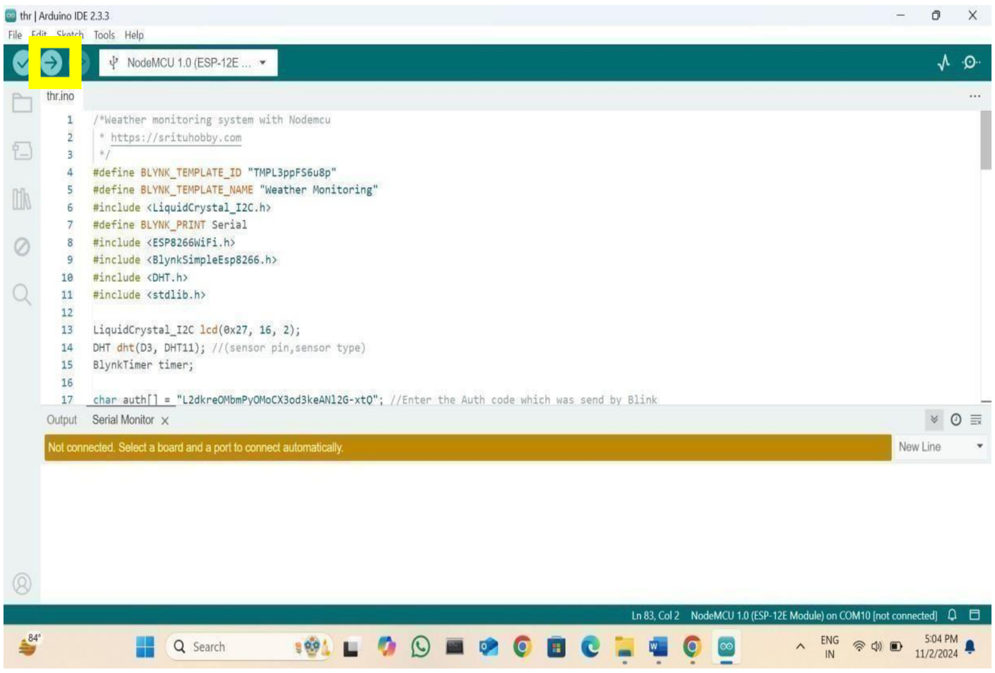
2. **Select the Board and Port:** In the Arduino IDE, navigate to the Tools menu. Under the Board option, select the appropriate board (NodeMCU 1.0 (ESP-12E Module) or the corresponding NodeMCU variant). Next, under the Port option, choose the correct port to which your NodeMCU board is connected.

3. **Upload the Code:** Once the board and port are selected, click the Upload button in the Arduino IDE. The code will be compiled and uploaded to the NodeMCU board via the USB connection. This process ensures that the NodeMCU board is programmed with the embedded code and is ready to execute the intended functionality.



Board: "NodeMCU 1.0 (ESP-12E Module)"
Port: "COM10"
Upload Speed: "115200"
Debug port: "Disabled"
Flash Size: "4MB (FS:2MB OTA:~1019KB)"
C++ Exceptions: "Disabled (new aborts on oom)"
lwIP Variant: "v2 Lower Memory"
Builtin Led: "2"
Debug Level: "None"
MMU: "32KB cache + 32KB IRAM (balanced)"
Non-32-Bit Access: "Use pgm_read macros for IRAM/PROGMEM"
SSL Support: "All SSL ciphers (most compatible)"
Stack Protection: "Disabled"

Fig 5.5.1 SETTING PORT AND BOARD



```
/*Weather monitoring system with Nodemcu
 * https://srituhobby.com
 */
#define BLYNK_TEMPLATE_ID "TMPL3ppFS6u8p"
#define BLYNK_TEMPLATE_NAME "Weather Monitoring"
#include <LiquidCrystal_I2C.h>
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <DHT.h>
#include <stdlib.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);
DHT dht(D3, DHT11); //(sensor pin,sensor type)
BlynkTimer timer;

char auth[] = "L2dkreOMbmPyOMoCX3od3keANl2G-xtQ"; //Enter the Auth code which was send by Blink
```

Not connected. Select a board and a port to connect automatically.

Ln 83, Col 2 NodeMCU 1.0 (ESP-12E Module) on COM10 [not connected]

Fig 5.5.2 UPLOADING THE CODE

5.5 TESTING

Testing is a vital phase in the development of any project, as it ensures the reliability, accuracy, and durability of the device under diverse environmental conditions. This process encompasses a series of checks for each component, beginning with sensor accuracy testing, where each sensor, such as the DHT11 for temperature and humidity, rainfall sensor, and LDR for light intensity, is calibrated and tested against known standards to confirm accurate readings. For example, the temperature and humidity readings from the DHT11 sensor are compared with a standard thermometer and hygrometer to verify precision within acceptable error margins. Testing in various temperature and humidity ranges is essential to confirm the sensors' responsiveness and consistency, ensuring reliable data collection across different weather scenarios.

Following sensor accuracy testing, data transmission testing is conducted to validate the NodeMCU's Wi-Fi connectivity and data integrity when communicating with the Blynk Cloud platform. This step involves evaluating the stability and reliability of the Wi-Fi connection, especially under conditions of weak signal strength or interference, and testing the system's ability to resume data transmission automatically after connection interruptions. Since the weather monitoring system relies on real-time updates, minimizing delay between sensor readings and data display on the Blynk app is crucial, allowing users to monitor environmental conditions promptly.

The environmental condition testing module tests the entire system's resilience to different weather conditions, such as varying temperatures, humidity, light levels, and rainfall, to confirm it performs consistently and withstands real-world weather challenges. Power consumption is also evaluated in cases where the device is battery- powered to ensure efficient energy usage, as frequent battery changes or unexpected shutdowns would reduce the system's practicality. Furthermore, user interface testing on the Blynk app ensures that data display is clear and that users can interact with controls, like setting threshold alerts for specific environmental conditions. This test guarantees that the app reflects real-time data accurately and allows users to customize the monitoring experience. Lastly, error handling and system stability are tested to ensure that the system remains operational even when certain components fail or unexpected reboots occur.

5.5.1 TEST CASES

1. Sensor Accuracy Testing

This step ensures that each sensor in the system accurately captures environmental data.Test Cases:

Temperature Sensor (DHT11)

TC1.1: Verify if the DHT11 sensor accurately measures the ambient temperature within ±2°Cof a calibrated thermometer reading.

TC1.2: Check sensor accuracy across a range of temperatures (e.g., 10°C, 25°C, 35°C).TC1.3: Test response time for temperature changes (e.g., placing the sensor in a warmer/cooler environment and measuring reaction time).

Humidity Sensor (DHT11)

TC1.4: Verify if humidity readings are within ±5% of a standard hygrometer reading.TC1.5: Test sensor accuracy at low and high humidity levels (e.g., 30%, 80%).

Rainfall Sensor

TC1.6: Simulate varying rainfall levels (light rain, moderate rain, heavy rain) and check if thesensor outputs correspond to expected values.

TC1.7: Verify if the sensor detects no rainfall in a dry environment.TC1.8: Test sensor responsiveness to short bursts of rain.

Light Sensor (LDR)

TC1.9: Test the LDR’s response to different lighting conditions (e.g., bright sunlight, shade,darkness).

TC1.10: Check if the sensor detects sudden changes in light intensity accurately.

2. Data Transmission Testing

This ensures the NodeMCU’s Wi-Fi module reliably transmits data to the Blynk Cloud,maintaining connection and data accuracy.

Test Cases:

Wi-Fi Connectivity

TC2.1: Test if NodeMCU connects successfully to Wi- Fi within 10 seconds.

TC2.2: Simulate weak Wi-Fi signals and check if the NodeMCU maintains a stableconnection.

TC2.3: Disconnect and reconnect Wi-Fi to test if the NodeMCU automatically reconnects.

Data Integrity

TC2.4: Confirm data consistency by comparing values transmitted to Blynk with actual sensorreadings.

TC2.5: Test for packet loss by disconnecting and reconnecting Wi-Fi and checking for anymissing data points on Blynk.

Real-Time Updates

TC2.6: Measure the delay between a change in environmental condition and its appearanceon the Blynk app.

TC2.7: Simulate rapid environmental changes to test the update frequency and speed onthe Blynk platform.

3. Power Consumption Testing

Ensures the device operates efficiently with minimal power consumption, crucial if usinga battery-powered setup.

Test Cases:

Power Usage

TC3.1: Measure NodeMCU's power draw in normal operation to ensure it stays withinacceptable limits.

TC3.2: Test the device's power consumption under peak data transmission.

Voltage Stability

TC3.3: Test for stable power output by monitoring sensor readings for inconsistencies causedby voltage drops.

TC3.4: Simulate low battery power and check for potential malfunctions.

4. Environmental Condition Testing

Tests the system's ability to handle different environmental conditions.Test Cases:

Temperature Variations

TC4.1: Place the device in a high-temperature environment and observe stability.TC4.2: Place the device in a low-temperature environment and observe stability.

Humidity Variations

TC4.3: Test performance under high humidity (above 90%) and check if all componentsoperate normally.

TC4.4: Test under low humidity conditions to ensure sensor consistency.

Light Variations

TC4.5: Expose the device to sudden changes in light and observe if the LDR accuratelycaptures variations.

TC4.6: Test the LDR in complete darkness and in full daylight.

Rain Simulation

TC4.7: Simulate rainfall on the sensor to check accurate response.

TC4.8: Test system behavior during prolonged exposure to water to ensure the rainfallsensor's durability.

5. Data Display and User Interface Testing

This ensures that the user can view and interact with data smoothly on the Blynk app.Test Cases:

Data Display

TC5.1: Check if all sensor data is correctly displayed on the Blynk app interface.

TC5.2: Test app responsiveness when data updates frequently.

User Controls

TC5.3: Check if the user can adjust alert thresholds on the app for temperature, humidityTC5.4: Test custom notifications for exceeding preset environmental limits.

6. Error Handling and System Stability Testing

Tests the system’s stability under failures and its ability to recover or provide feedbackon errors.

Test Cases:

Sensor Failures

TC6.1: Disconnect the DHT11 sensor and observe if the system reports an error.

TC6.2: Repeat for each sensor (rainfall and LDR) and check for alerts or notifications.**System Reboot**

TC6.3: Manually reboot the NodeMCU to check if it resumes data transmission seamlessly.TC6.4: Test if the system automatically reconnects to Wi- Fi and Blynk after a reboot.

Network Failures

TC6.5: Disconnect Wi-Fi for a few minutes and reconnect, checking if data updates resume onBlynk.

TC6.6: Check for data storage during network outages if the system has offline data-savingcapabilities.

7. Data Logging and Historical Data Testing

If the system stores data, this ensures it logs data correctly for historical analysis.Test Cases:

Data Logging

TC7.1: Verify if data logs are accurate with proper timestamps for temperature, humidity, andother sensor readings.

TC7.2: Check data accuracy over an extended period to confirm consistent logging.

Historical Analysis

TC7.3: Verify that the Blynk app allows easy access to historical data.

TC7.4: Test if historical data displays correctly and is free of gaps or inconsistencies.

8. Customization and Flexibility Testing:

Ensures the system can handle additional sensors or configuration changes.Test Cases:

Sensor Replacements

TC8.1: Replace the DHT11 with a new one and verify the system's ability to operate withoutreprogramming.

TC8.2: Test adding another sensor to the system and check if the data integrates with Blynk.

Configuration Testing

TC8.3: Change threshold values (e.g., temperature alerts) in the app and confirm newsettings are saved.

TC8.4: Test if customizations are retained after rebooting the system.

CHAPTER 6: OUTPUT

6.1 OUTPUT SCREENS

6.1.1 WEBSITE VIEW

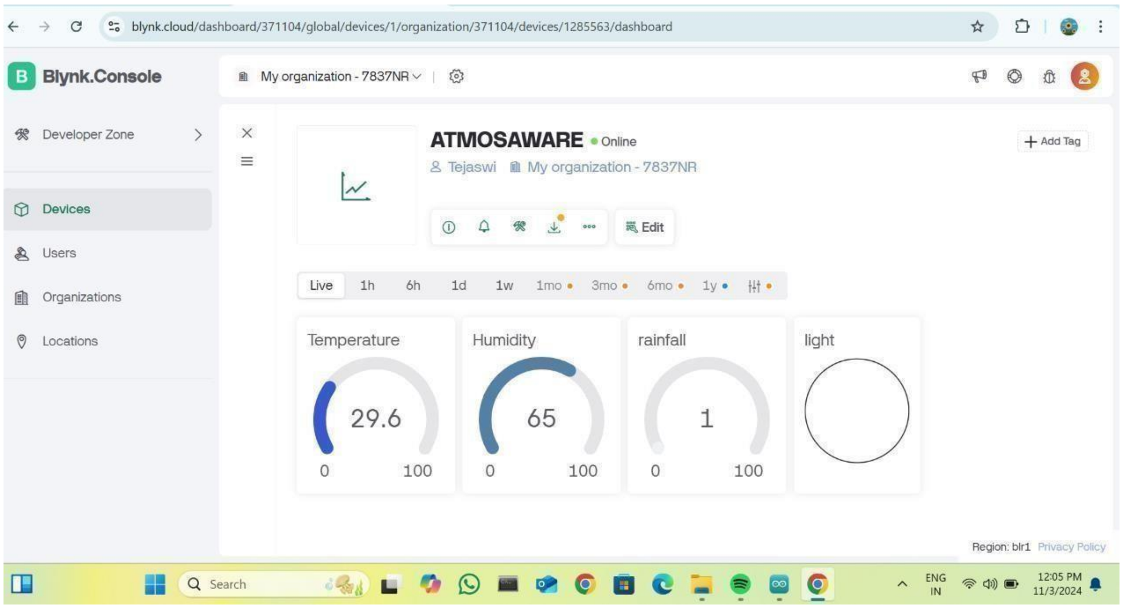


Fig 6.1.1.1 Clear Beginnings

Fig 6.1.1.1 Clear Beginnings: A Snapshot of Initial Environmental Conditions the Clear Beginnings stage depicts an environment characterized by mild and steady weather conditions. This serene setting can be observed through several key parameters:

- Temperature: The ambient temperature is recorded at 29.6°C, which suggests a warm atmosphere. This level of warmth can be considered comfortable for many outdoor activities, but it may also indicate the onset of heat, particularly in regions prone to higher temperatures. Such conditions often encourage the growth of various plants and can affect local wildlife behaviors, as many species thrive in warmer climates.
- Humidity: The humidity level stands at 65%, reflecting a moderately humid environment. This level of humidity can contribute to a feeling of warmth and can be conducive to the growth of certain plant species that require moisture. However, it may also lead to a slight discomfort for individuals sensitive to humidity, as higher humidity levels can impede the body’s ability to cool itself through sweat evaporation.

- Rainfall: The rainfall measurement is recorded at 1 unit, indicating a very low amount of precipitation. Depending on the calibration of the sensor, this could reflect a recent light rain or simply a threshold reading indicating minimal moisture accumulation. Such low rainfall is characteristic of dry spells, which can have significant implications for agriculture, water supply, and the local ecosystem.
- Light Intensity: The Light Dependent Resistor (LDR) sensor displays an unfilled circle, indicating low or no light intensity detected in the environment. This could imply that the area is shaded, possibly due to overcast skies or obstructions such as buildings or trees. Low light intensity can impact plant growth and behavior, as many species rely on sunlight for photosynthesis. Additionally, reduced light levels can influence the activity patterns of nocturnal and diurnal animals, affecting their feeding and breeding behaviors.

The conditions characterized by low temperature, humidity, and light intensity suggest a calm and stable environment, potentially ideal for relaxation or certain outdoor activities. However, the mild conditions may also indicate a need for careful monitoring, particularly in agricultural settings, to ensure that the plants receive adequate care during this dry spell. Furthermore, understanding these initial environmental parameters is crucial for assessing trends over time, as fluctuations in these readings can signal changes in weather patterns or ecological conditions, prompting necessary interventions or adjustments in management practices. In conclusion, "Clear Beginnings" serves as a vital reference point for analyzing the evolving dynamics of the weather and its broader implications on both the environment and human activities.

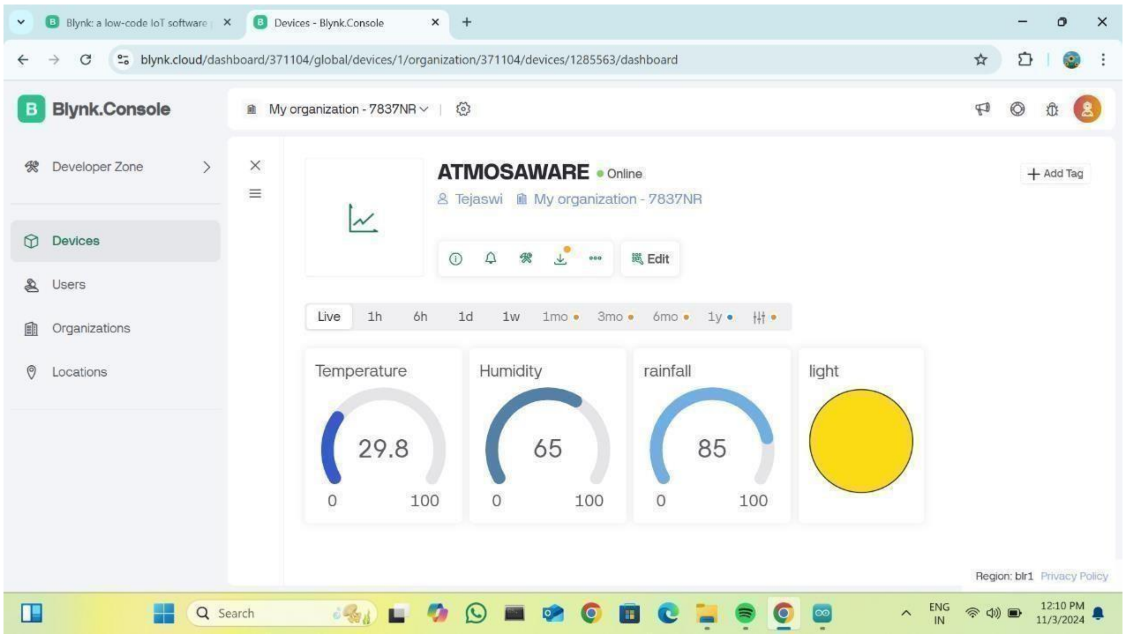


Fig 6.1.1.2 Dynamic Change

Fig 6.1.1.2 Dynamic Change: A Transition to More Active Weather the Dynamic Change phase reflects a noticeable shift in environmental conditions, characterized by a balance of increasing light and rainfall. This transition signals a more active and potentially volatile weather pattern, marked by several key parameters:

- Temperature: The recorded temperature has risen slightly to 29.8°C, indicating a minor yet significant increase from the previous reading. This change, although subtle, can influence local ecosystems and human activities. A slight increase in temperature often correlates with heightened metabolic rates in various organisms and can encourage increased plant growth and activity. However, it may also lead to discomfort for individuals sensitive to heat, particularly in combination with humidity.
- Humidity: Humidity remains stable at 65%, suggesting consistent moisture levels in the environment. This stability indicates that despite changes in other parameters, the air's moisture content is holding steady. Stable humidity can be beneficial for certain crops and plants, as it reduces stress and promotes growth. However, if rainfall increases significantly, as indicated in this phase, the interaction between humidity and precipitation may alter the moisture availability in the soil, impacting agriculture and water drainage.

- Rainfall: The rainfall measurement has increased dramatically to 85 units, indicating a sudden onset or detection of rain. This significant uptick can imply a shift toward more active weather conditions, possibly due to the arrival of a weather front or storm system. Increased rainfall may affect local wildlife, prompting changes in behavior as animals seek shelter or adapt to the availability of water resources. Aquatic ecosystems may experience a boost in nutrient flow, potentially leading to algal blooms or shifts in fish populations.
- Light Intensity: The light sensor displays a filled yellow circle, indicating a higher light intensity. This increase in light can result from several factors, such as the emergence of sunlight through clearing clouds or enhanced indoor lighting conditions. For plants, increased light intensity often leads to enhanced photosynthesis rates, which can promote growth and flowering. This can be particularly significant for crops and gardens that rely on optimal sunlight for yield. Higher light levels can influence the activity patterns of diurnal animals, which may become more active during periods of increased sunlight, impacting foraging and mating behaviors.

The "Dynamic Change" phase signifies an important transition toward more active weather conditions, with interrelated impacts on temperature, humidity, rainfall, and light intensity. These changes can influence agricultural practices, local ecosystems, and human comfort levels. Monitoring these shifts is essential for effective environmental management and decision- making, particularly in the context of climate variability and potential extreme weather events. In conclusion, this stage not only highlights the immediate changes in the weather but also serves as a reminder of the interconnectedness of various environmental factors and their cumulative impact on the ecosystem and human activities.

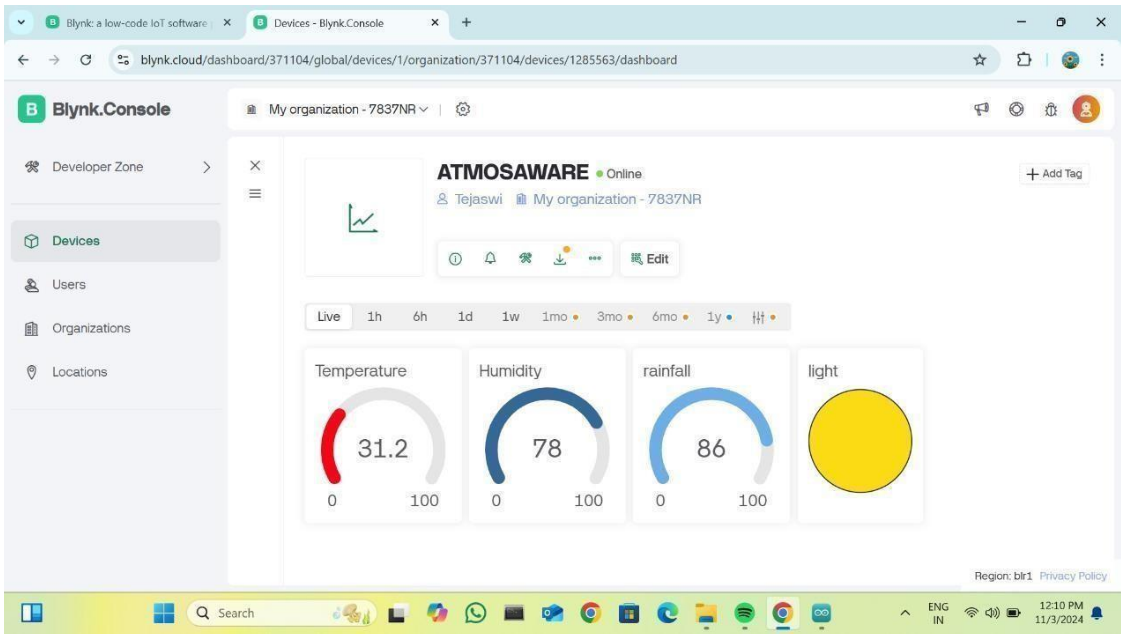


Fig 6.1.1.3 High Humidity Zone

Fig 6.1.1.3: High Humidity Zone: It is a Peak in Environmental Parameters. This phase is characterized by elevated readings across several critical environmental parameters, highlighting a peak in humidity and temperature. This stage reflects the interplay between ongoing rainfall and increased sunlight, resulting in significant atmospheric changes. Below are the detailed observations for each parameter:

- Temperature: The temperature has risen further to 31.2°C, indicating a consistent warming trend. As temperatures rise, there is an increased risk of heat stress for both humans and wildlife. For humans, prolonged exposure to higher temperatures can lead to discomfort and health risks such as heat exhaustion. For wildlife, especially ectothermic species (cold-blooded), the elevated temperatures caninfluence their activity levels and metabolic rates. Crops may respond positively to the increased warmth; however, extreme temperatures can also adversely affect yields, particularly if combined with water stress or insufficient light conditions.
- Humidity: Humidity levels have surged to 78%, indicating a significant increase in moisture content in the air. High humidity can lead to discomfort for humans as it impairs the body's ability to cool itself through sweating. This may necessitate the use of air conditioning or other cooling measures in residential and commercial spaces. Elevated humidity can benefit certain crops, particularly those that thrive in moist conditions.

However, excessive humidity may also lead to the development of fungal diseases and pests that thrive in damp environments. High humidity can result in increased condensation, leading to potential visibility issues and the formation of fog, especially during nighttime or early morning hours.

- Rainfall: The rainfall measurement has slightly increased to 86 units, indicating ongoing or increased precipitation. Continuous rainfall can saturate the soil, enhancing water availability for plants but also increasing the risk of waterlogging and root rot in susceptible species. Excessive moisture can lead to runoff and soil erosion, impacting agricultural productivity. Prolonged periods of rainfall can elevate the risk of flooding, particularly in low-lying areas, necessitating adequate drainage and water management strategies to mitigate damage.
- Light Intensity: The light sensor continues to display a filled yellow circle, indicating that light levels remain high. With higher light levels persisting, plants can maximize photosynthesis, promoting growth and development. This can be especially beneficial for agricultural crops during the growing season. High light intensity may encourage the activity of various animal species, including pollinators, which can enhance ecosystem productivity. However, it may also lead to increased competition among plants for light, especially in dense vegetation areas.

The High Humidity Zone represents a critical phase in environmental monitoring, highlighting a peak in temperature and humidity driven by ongoing weather patterns. The interactions among temperature, humidity, rainfall, and light intensity underline the complexities of the ecosystem and the potential challenges faced by both natural and human systems. For urban planning and infrastructure, recognizing the implications of high humidity and rainfall can guide the design of resilient systems capable of managing potential flooding and ensuring comfort for inhabitants.

In summary, the High Humidity Zone serves as an essential reference point for monitoring environmental conditions, facilitating informed decision-making in agriculture, ecology, and urban management as climate variability continues to present challenges and opportunities.

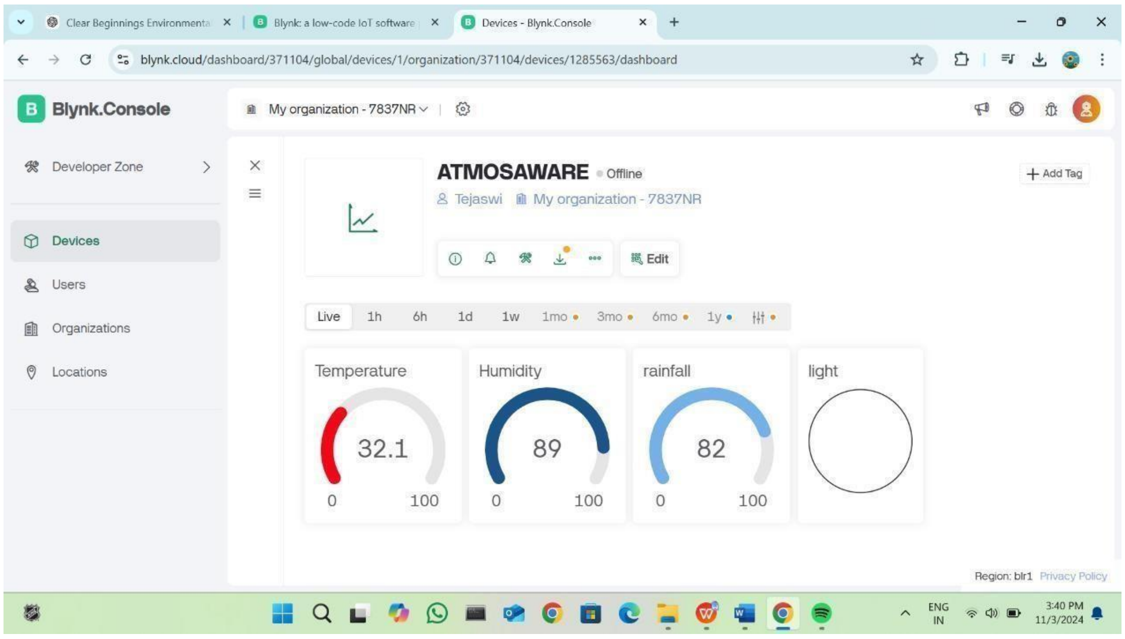
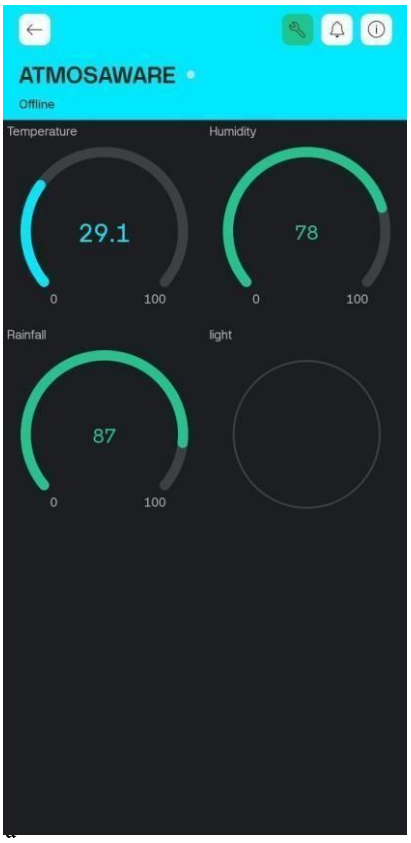


Fig 6.1.1.4: Inactive

Fig 6.1.1.4 Inactive: It shows the offline status of the "ATOMSAWARE" device indicates thatthere is no established connection between the device and the Blynk Cloud. As a result, the dashboard displays previously recorded data, allowing users to review historical performance metrics despite the current lack of connectivity. This feature ensures that users can still access valuable insights into the device's operational history, facilitating informed decision-making and troubleshooting.

6.1.2 MOBILE VIEW

The mobile view of the Blynk Console dashboard for the "ATOMSAWARE" device offers significant advantages by enabling users to access and monitor their device from smartphones or tablets, ensuring real-time updates and notifications wherever they are. This streamlined interface is optimized for smaller screens, presenting critical information clearly and facilitating quick control access for actions such as restarting the device or adjusting settings. Users benefit from simplified data visualizations that allow them to track performance metrics at a glance, leading to improved decision-making and responsiveness to changes in environmental conditions. Furthermore, integration with push notifications keeps users informed of important events, enhancing overall engagement and management of their IoT device.



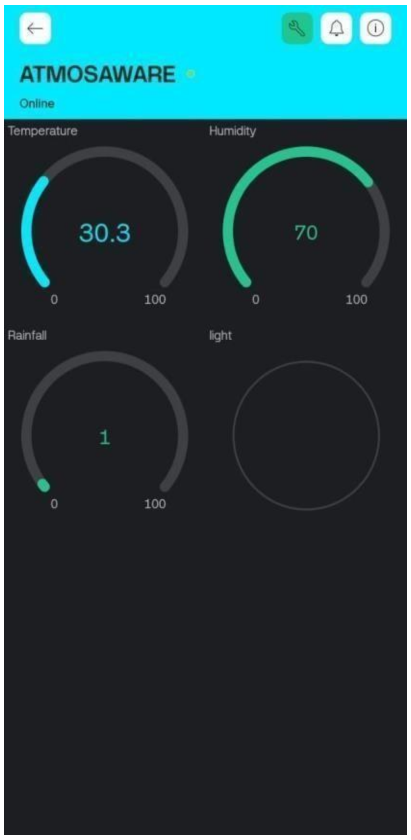


Fig 6.1.2.1 Offline Data Viewer   Fig 6.1.2.2 Stable Atmosphere

The Fig 6.1.2.1 Offline Data Viewer Conveys that the dashboard provides an overview of the last status of the device.

The Fig 6.1.2.2 Stable Atmosphere is the initial image depicts a serene environment with a comfortable temperature of 30.3°C and moderate humidity at 70%, indicating balanced conditions. With minimal rainfall of 1 unit and low light intensity, it highlights a stable atmosphere conducive to both flora and fauna.

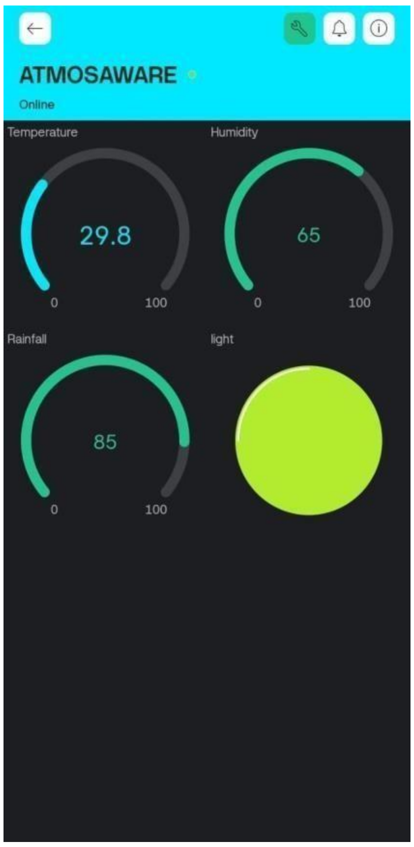


Fig 6.1.2.3 Rising Intensity

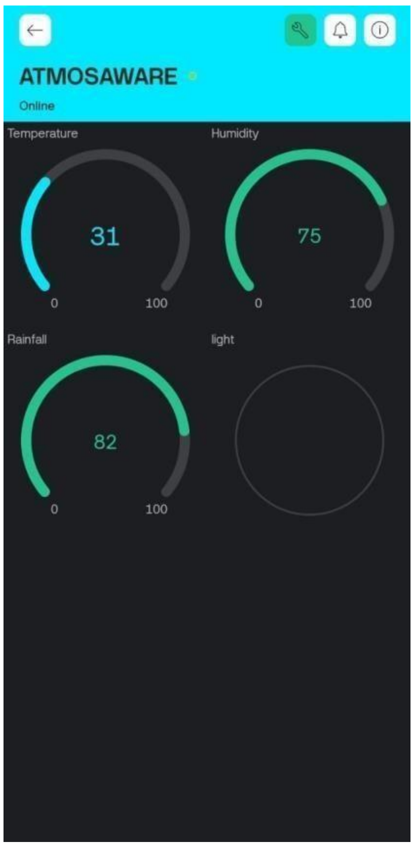


Fig 6.1.2.4 Humidity Surge

In fig 6.1.2.3 Rising Intensity the data indicates a slight increase in temperature, accompanied by a significant spike in both rainfall and light intensity, suggesting a shift towards more dynamic weather conditions.

In fig 6.1.2.4 Humidity Surge this phase emphasizes the peak levels of humidity and temperature, indicating a critical point in the environmental conditions that may affect both ecological systems and human activities.

6.2 GRAPHS

The fig 6.1.3.1 and fig 6.1.3.2 shows the performance of a device tested on Nov 1, 2024, over a specific time range (from 11:00 AM to 2:00 PM). The purpose of the test appears to be to monitor how temperature and humidity levels change in the environment where the device is operating. This analysis is useful for assessing how well the device performs under different environmental conditions and might also help in identifying any fluctuations that could impact its functionality.

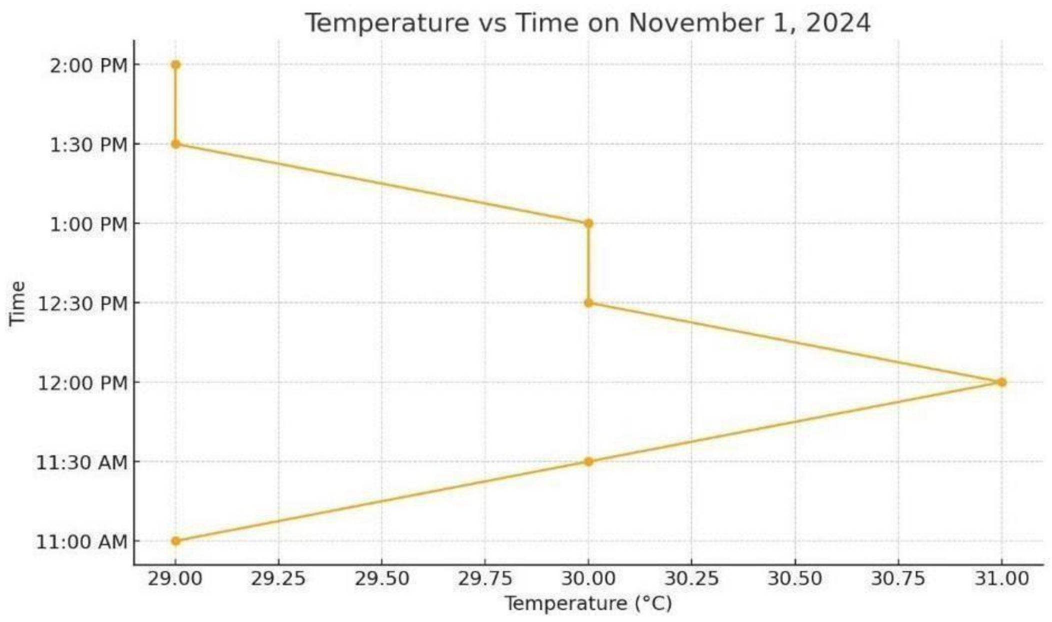


Fig 6.2.1 Temperature vs Time

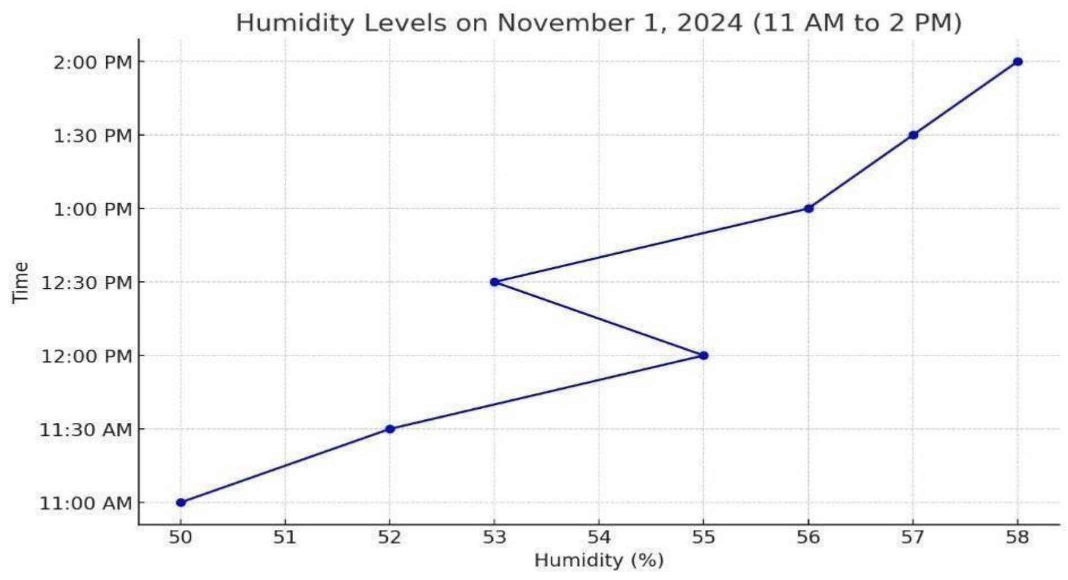


Fig 6.2.2 Humidity vs Time

6.3 WHY BLYNK CLOUD?

Using the Blynk app for weather monitoring with NodeMCU provides a multitude of advantages that significantly enhance both functionality and user experience. One of the primary benefits is its intuitive and user-friendly interface, which allows users to easily monitor real-time weather data without needing extensive technical knowledge. The app's drag-and-drop builder facilitates seamless customization of the dashboard, ensuring that users can design an interface that meets their specific needs and preferences.

Blynk excels in real-time data visualization, enabling users to receive live updates directly from their NodeMCU device. This capability is essential for timely decision-making, particularly in scenarios where weather conditions can change rapidly, such as in agricultural management or event planning. Furthermore, the app supports remote access, allowing users to monitor their weather station from anywhere using their smartphones or tablets. For example, a farmer could check soil moisture levels and weather forecasts while away from the farm, enabling them to make informed irrigation decisions without needing to be physically present.

The Blynk app also offers robust data logging features, allowing users to track historical weather data over time. This function is vital for analyzing trends and patterns in environmental conditions, which can inform future planning and strategy. For instance, a researcher studying climate change can utilize the data collected over months or years to draw meaningful conclusions about local weather patterns.

Customizable alerts and notifications are another critical feature of Blynk, enabling users to receive real-time updates on specific weather conditions, such as high humidity, temperature spikes, or rainfall thresholds. This proactive approach ensures that users can respond swiftly to significant changes, potentially mitigating adverse impacts on crops or outdoor activities. Moreover, Blynk allows integration with various sensors and IoT devices, enhancing the versatility of the weather monitoring system. Users can easily add additional sensors to measure wind speed, UV index, or atmospheric pressure, providing a comprehensive overview of their local weather conditions. Cost-effectiveness is also a significant advantage, especially for DIY projects. Blynk offers a free tier with essential functionalities, while premium features can be unlocked at reasonable prices, making it accessible for hobbyists and professional developers alike. Its cross-platform compatibility ensures that users can access the app on both iOS and Android devices, broadening its usability.

Additionally, Blynk benefits from an active user community and extensive documentation, which provide valuable support for troubleshooting and optimizing projects. This resource is especially beneficial for both beginners and experienced users looking to enhance their weather monitoring systems.

In summary, the Blynk app, when paired with NodeMCU for weather monitoring, creates a powerful and flexible system that enhances user engagement, provides real-time insights, and supports effective environmental management. The ability to monitor and access data from anywhere, coupled with the ease of integration with additional sensors, makes Blynk an ideal choice for anyone looking to establish a comprehensive weather monitoring solution.

6.4 APPLICATIONS OF ATOMSAWARE

The Atomsaware weather monitoring system leverages IoT to provide real-time, remote, and cost-effective environmental data collection across a broad range of applications:

1. **Agriculture and Farming:** This system helps farmers optimize irrigation by tracking temperature, humidity, and rainfall, supports crop health monitoring, and enables pest and disease management by analyzing weather conditions that promote pest growth.
2. **Environmental Research and Climate Studies:** It aids climate research by providing long-term data for tracking climate change, assessing biodiversity impacts, and monitoring air pollution levels to understand environmental health risks.
3. **Disaster Management and Preparedness:** The system can forecast floods, droughts, and extreme weather events by tracking rainfall, humidity, and temperature, offering timely alerts to communities for better preparedness.
4. **Smart Cities and Urban Planning:** It assists cities in managing urban heat, monitoring air quality, and improving public facilities by providing data that support sustainable urban design and air pollution control.
5. **Energy Management:** The system optimizes renewable energy resources like solar and wind, helps forecast energy consumption, and supports energy-saving initiatives.
6. **Home Automation and Smart Homes:** The system enhances indoor climate control, automates lawn irrigation, and alerts residents about allergens and air quality, making homes healthier and more efficient.
7. **Educational Applications and STEM Projects:** It offers students hands-on experience in IoT and data analysis and supports environmental awareness campaigns, helping schools and communities learn about local climate patterns.
8. **Transportation and Infrastructure:** Weather monitoring supports road and traffic safety, helps manage schedules for airports and railways, and ensures optimal conditions for construction projects.

In summary, Atomsaware offers versatile applications in agriculture, environmental research, disaster preparedness, smart cities, and beyond, promoting sustainability, safety, and efficient resource management across diverse fields.

CHAPTER 7: CONCLUSION

ATMOSAWARE is a weather monitoring device that utilizes IoT technology to offer a highly efficient, scalable, and cost-effective approach to gathering and analyzing environmental data. By leveraging the NodeMCU microcontroller, this system is designed to capture critical parameters such as temperature, humidity, rainfall, and light intensity, making it highly valuable across multiple sectors including agriculture, urban planning, disaster management, energy optimization, and education. It has Real-Time Data Collection and Remote Access. The system provides continuous, real-time weather data, which users can access remotely from mobile devices, ensuring that critical information is always available, regardless of location. This feature is especially beneficial for sectors that rely on timely weather data for operational decisions, such as farmers who need accurate information to optimize irrigation or disaster response teams requiring immediate updates during severe weather events. With accurate and up-to-date weather insights, users in agriculture can manage resources efficiently, adjusting irrigation based on actual soil moisture and precipitation, which helps conserve water and reduce costs. In urban settings, weather data from ATMOSAWARE can help city planners design more resilient infrastructure and mitigate urban heat island effects by identifying hotspots and promoting green spaces. It provides Environmental Awareness and Sustainability. The system empowers users to track climate- related changes and make more sustainable choices based on reliable data. For example, in educational settings, ATMOSAWARE can be integrated into STEM curriculums, giving students hands-on experience with IoT technology while promoting environmental literacy. Communities, in turn, can leverage this data to better understand local climate trends, improving resilience against climate-related challenges.

CHAPTER 8: FUTURE SCOPE

The future scope of **ATOMSAWARE** includes significant enhancements to its monitoring capabilities through the integration of additional sensors. These improvements will not only expand the range of measurable parameters but also bolster its application in various domains such as environmental monitoring, disaster preparedness, and smart city planning.
Here are some sensors that hold great potential for enhancing future applications:

- **Barometric Pressure Sensor**: This sensor would improve weather prediction accuracy by tracking changes in atmospheric pressure, which are critical indicators of rain, storms, or other weather events. Applications include more precise short-term and long-term weather forecasting, enabling better preparation for adverse weather conditions.
- **Gas Sensors (e.g., CO2, NO2, Methane)**: Integration of gas sensors would provide real-time air quality data, crucial for pollution monitoring and health advisory systems. Potential use cases include urban air quality monitoring, early warning systems for hazardous gas leaks, and contributing to climate change research.
- **Wind Speed and Direction Sensors**: These sensors would be invaluable for renewable energy projects by aiding in the optimal placement and operation of wind turbines. Additionally, they could enhance disaster preparedness efforts by alerting authorities to high wind events, such as storms or cyclones, reducing potential damages.
- **Soil Moisture Sensor**: By measuring soil moisture levels, this sensor would be a boon for agriculture, helping farmers make informed irrigation decisions and conserve water resources. It would also aid in land management practices and improve crop yield predictions.
- **Ultraviolet (UV) Sensor**: A UV sensor could monitor UV radiation levels, providing valuable data for public health advisories related to sun exposure and skin protection. This feature would also benefit research into the effects of UV radiation on ecosystems and materials.

- **Particulate Matter (PM) Sensor**: By measuring fine particles (PM2.5 and PM10), this sensor would enhance air quality monitoring, helping authorities and communities combat pollution. It is particularly useful for monitoring urban environments and areas affected by wildfires.
- **Rain Gauge Sensor**: Adding a rain gauge sensor would enable the system to measure the amount and intensity of rainfall directly, providing more granular precipitation data for weather analysis. This data could help in water resource management and flood forecasting.
- **Sound Level Sensor**: Monitoring environmental noise pollution can provide insights into urban noise levels, aiding smart city planning and public health studies related to noise-induced stress.
- **Solar Radiation Sensor**: A solar radiation sensor would track sunlight intensity, benefiting solar energy projects by optimizing panel placement and efficiency. It can also support agricultural applications by analyzing light availability for crop growth.
- **Infrared (IR) Thermal Sensor**: By monitoring surface temperatures, this sensor could detect heat islands in urban areas or assess thermal characteristics in agricultural fields. It can also support energy efficiency studies in building management.

CHAPTER 9: BIBILOGRAPHY

[1]. IoT Weather Monitoring System – Techatronic https://techatronic.com/iot-weather-monitoring-system/

[2]. Journal by Kalyan Kumar, P. Ravi Shankar, G. Harika, Thirumal, S. Yuvaraj1 Assistant Professor, UG scholars,EEE department, KSRMCE(A), Kadapa, AP http://www.journal-iiie-india.com/1_may_24/127_online_may.pdf

[3]. 127 Online - Journal PDF http://www.journal-iiie-india.com/online_may.pdf

[4]. IoT Live Weather Station Monitoring Using NodeMCU ESP8266 - How2Electronics https://how2electronics.com/iot-live-weather-stationmonitoring-using-nodemcu-esp8266/

[5]. IoT Weather Monitoring System – Hackster https://www.hackster.io/Techatronic/iot-weather monitoring-system-ffbf89

[6]. IoTWeather Station Project Using NodeMCU - Learn Electronics India https://www.learnelectronicsindia.com/post/iotweather-station-project-using-nodemcu

[7]. IoTWeather Station Using ESP8266 Board – Instructables https://www.instructables.com/IoT-Weather-StationUsing-ESP8266-Board/

[8]. SSRNPaper https://papers.ssrn.com/sol3/papers.cfm?abstract_id=448

[9]. Weather Reporting System Using IoT – WebbyLab https://webbylab.com/blog/weather-reporting-systemusing-iot-benefits-use-cases/

[10]. RCCIITStudentProject-GR9.pdf https://www.rcciit.org.in/students_projects/projects/ee/2 022/GR9.pdf

[11]. Weather Monitoring Paper – WJARR https://wjarr.com/sites/default/files/WJARR-2024-0679.pdf

[12]. Blynk IoT - Low Code Software Platform https://blynk.io/blynk-iot-low-code-software-platform

[13]. DHT11TemperatureSensor-Components101 https://components101.com/sensors/dht11-temperaturesensor

[14]. Arduino Software https://www.arduino.cc/en/software

[15]. NodeMCUSpecificationsMake-ithttps://www.make-it.ca/nodemcu-details-specifications/

[16]. Rain Sensor Working - Elprocus https://www.elprocus.com/rain-sensor-working-and-itsapplications

[17]. IntroductiontoNodeMCU-ElectronicWings https://www.electronicwings.com/nodemcu/introduction-to-nodemcu

[18]. Weather Monitoring Benefits Sigfox https://sigfox.com/weather-conditions-monitoringbenefits/

[19]. A Brief on DHT11 Sensor - Elprocus https://www.elprocus.com/a-brief-on-dht11-sensor/

[20]. Working of DHT Sensors - Nerdy Electronics https://nerdyelectronics.com/working-of-dht-sensor-dht11-an dht22/