Faculty of Engineering Helwan University

Electronics, Communications and Computer Engineering Department

Computer and Software Engineering Section

Project Name: AirAware

Supervised By Dr: Mahmoud Zaki

Prepared By:

Hosam Mohamed Ahmed

John Salama Beshay

Rania Mustafa Ibrahim

Helal Omar Helal

Abstract

Air pollution poses a critical environmental concern, impacting the well-being of individuals and communities worldwide. To effectively tackle this issue, it becomes crucial to ensure reliable and accurate monitoring of air quality. With this objective in mind, we present an innovative gas sensor-based Air Quality Index (AQI) device that senses and analyzes various gases present in the atmosphere.

Within the pages of this book, we strongly emphasize the significance of monitoring air quality, considering its profound effects on human health and the environment. We address the urgent need for precise and accessible air quality monitoring solutions that can raise awareness and enable informed decision-making.

The primary aim of the AQI device project is to develop an affordable system that offers efficient gas detection capabilities. The book thoroughly explores diverse approaches and methodologies employed to attain this objective. It highlights the utilization of cost-effective yet reliable sensors, optimization of data acquisition and processing techniques, and the implementation of economically viable hardware solutions.

Readers of this book will acquire a comprehensive understanding of the AQI device project, its underlying technology, and its potential applications. The primary intention is to inspire researchers, engineers, and enthusiasts in the field of air quality monitoring by providing a detailed account of the device's developmental journey and the challenges encountered along the way. Ultimately, this book serves as a valuable resource for those interested in developing cost-effective gas detection and monitoring systems, thus contributing to the broader objective of promoting healthier and more sustainable environments.

Table of Contents

List of Figures

List of Acronyms and Definitions

ADC – Analog-Digital Converter
AR – Autoregression
ARIMA – Autoregressive Integrated Moving Average
ANN – Artificial Neural Network
AVR – Advanced Virtual RISC
AQI – Air Quality Index
AWS SDK – Amazon Web Services Software Development Kit

BLE – Bluetooth Low Energy

CO – Carbon Monoxide

DIO – Digital Input Output

EPA – Environmental Protection Agency
EEPROM – Electrically Erasable Programmable Read-Only Memory

HAL – Hardware Abstract Layer
HTTP – Hypertext Transfer Protocol

I2C – Inter-Integrated Circuit
IDE – Integrated Development Environment
IoT – Internet of Things
IR – Infrared

LED – Light-Emitting Diode
LSTM – Long Short-Term Memory

MCAL – Micro-controller Abstract Layer
MPPT – Maximum Power Point Tracking
MQTT – Message Queuing Telemetry Transport
ML – Machine Learning

NO2 – Nitrogen Dioxide

O3 – Ozone Gas

PM2.5 – Particulate Matter 2.5 micrometers or smaller
PM10 – Particulate Matter 10 micrometers or smaller
PPB – parts per billion
PPM – parts per million
PWM – Pulse Width Modulation

RISC – Reduced Instruction Set Computer
RNN – Recurrent Neural Network

SnO2 – Tin oxide
SRAM – Static Random Access Memory

UART – Universal Asynchronous Receiver Transmitter

WDT – Watch-Dog Timer
WSN – Wireless Sensor Nodes
WHO - World Health Organization

SO2 – Sulfur Dioxide
SPI – Serial Peripheral Interface

VOCs – volatile organic compounds

Literature Review

We find related prior work regarding Air Quality Monitoring systems existworldwidee by companies like:

- **BreezoMeter:** company that provides real-time air quality data and insights. It utilizes advanced algorithms and data analytics to gather information from various sources, such as government monitoring stations, weather patterns, traffic, and satellite imagery.
    - **Founded in:** 2014
    - **Scale:** Global
    - **Features:** BreezoMeter offers its services through various channels, including a mobile app, API integration for businesses and smart devices, and partnerships with companies that incorporate air quality data into their products or services.

- **IQAir:** company that specializes in air purification technology and solutions**.** Their primary focus is on developing and manufacturing high-performance air purifiers for residential, commercial, and industrial use. Their air purifiers are designed to remove a wide range of pollutants from the air, but additionally, IQAir offers air quality monitoring solutions. Their monitoring devices provide real-time information about indoor and outdoor air quality, allowing individuals and organizations to track pollutant levels and make informed decisions to improve the air they breathe.
    - **Founded in:** 1963
    - **Scale:** Global
    - **Features:** IQAir's main focus is air purifiers but additionally, they offer air quality monitoring device and platform software for visualization and monitoring.

- **AirNow:** government-operated air quality monitoring and reporting system in the United States. It is a collaborative effort between the U.S. Environmental Protection Agency (EPA), the National Oceanic and Atmospheric Administration (NOAA), the National Park Service (NPS), tribal organizations, and state and local air quality agencies, The AirNow program provides real-time air quality information to the public through its website (airnow.gov), mobile apps, and various data dissemination tools.
    - **Founded in:** –
    - **Scale:** Global (but particularly US)

    - **Features:** AirNow collects air quality data from a network of monitoring stations located across the United States. These monitoring stations are operated by federal, state, tribal, and local agencies.

- **Clarioty.io:** company that provides a turn-key air quality monitoring solution, their mission is to empower the world to reduce air pollution by transforming how governments, communities, and businesses understand and respond to air pollution with IoT-based air quality monitoring technology.
    - **Founded in:** 2014
    - **Scale:** Global
    - **Features:** Clarity offers a variety of other air quality monitoring products and services, including Clarity Modules and Clarity OpenMap.

1. Introduction

1.1 Problem Statement:

Air pollution is the contamination of the air by harmful substances, resulting in adverse effects on human health, the environment, and overall quality of life. It occurs when pollutants, such as gases, particles, and biological materials, are released into the atmosphere through natural processes or human activities. This widespread issue affects individuals of all ages and backgrounds, as well as wildlife, plants, and the climate.

Humans are particularly susceptible to the detrimental effects of air pollution. Those exposed to high levels of pollutants are at risk of developing various respiratory and cardiovascular diseases, including asthma, bronchitis, lung cancer, and heart disease. Vulnerable populations such as children, the elderly, and individuals with pre-existing respiratory conditions are especially impacted, experiencing severe health consequences due to polluted air.

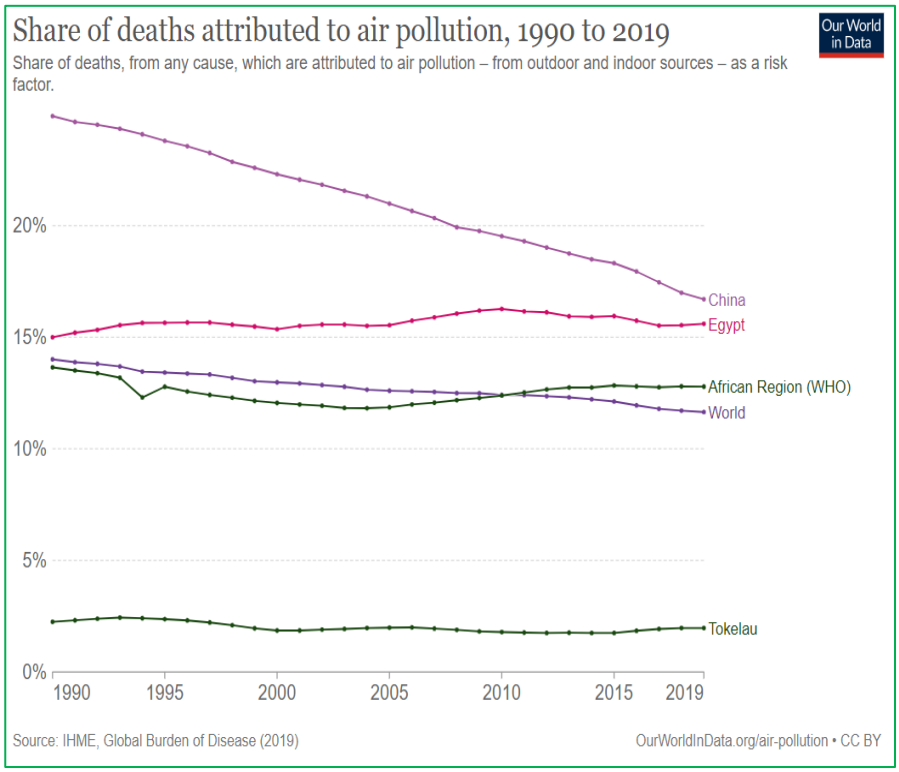


FIGURE 1: DEATHS FROM AIR POLLUTION

Not only do humans suffer, but wildlife and ecosystems are also victims of air pollution. Birds, mammals, and aquatic life can face respiratory problems, organ damage, impaired reproduction, and even death because of exposure to pollutants. Plants are not spared either, as air pollution damages their leaves, reduces photosynthesis, stunts growth, and leads to the acidification of soil and water bodies, negatively impacting their health and biodiversity.

1.2 Air Quality Index:

The Air Quality Index (AQI) is a standardized metric used to measure and communicate the level of air pollution in a specific location. It provides an easy-to-understand numerical value that represents the overall air quality and the potential health risks associated with pollution levels.

The AQI typically ranges from 0 to 500, with higher values indicating poorer air quality. The index is divided into different categories, each corresponding to a specific range of AQI values and associated health effects. The categories are often color-coded to enhance clarity and ease of interpretation.

FIGURE 2: AIR QUALITY INDEX

The AQI considers several pollutants, including particulate matter (PM2.5 and PM10), ground-level ozone (O3), nitrogen dioxide (NO2), sulfur dioxide (SO2), and carbon monoxide (CO). These pollutants are known to have adverse effects on human health and the environment.

By monitoring the concentrations of these pollutants and comparing them to established air quality standards, the AQI provides a measure of the air quality level. This information allows individuals, communities, and authorities to gauge the current air pollution situation and take appropriate actions to protect public health.

For example, an AQI value of 0 to 50 is typically associated with "Good" air quality, indicating minimal health risks. As the AQI increases, the air quality deteriorates, and

health effects may become more significant. AQI values in the higher ranges, such as 151 to 200 (Unhealthy) or 301 to 500 (Hazardous) Figure 2, suggest severe pollution levels with potential health hazards, particularly for sensitive individuals.

2. Solution

2.1 AirAware

Our device, "AirAware," offers a promising solution to reduce the impact of pollution by informing people about the state of their environment. Our project revolves around a system that monitors environmental changes, pollution levels, and air quality in specific regions. We achieve this through the implementation of a Wireless Sensor Network (WSN) that measures the Air Quality Index (AQI).

By deploying a network of sensors strategically across desired areas, AirAware continuously gathers real-time data on various pollutants present in the air. These sensors detect and measure concentrations of harmful substances, including particulate matter (PM2.5), ozone (O3), carbon monoxide (CO), in addition to temperature and humidity.

By raising awareness, providing valuable information, and facilitating data-driven decision-making, AirAware plays a vital role in reducing the impact of pollution on human health and the environment. By actively monitoring air quality and taking appropriate actions, we can work together towards cleaner and healthier surroundings for ourselves and future generations.

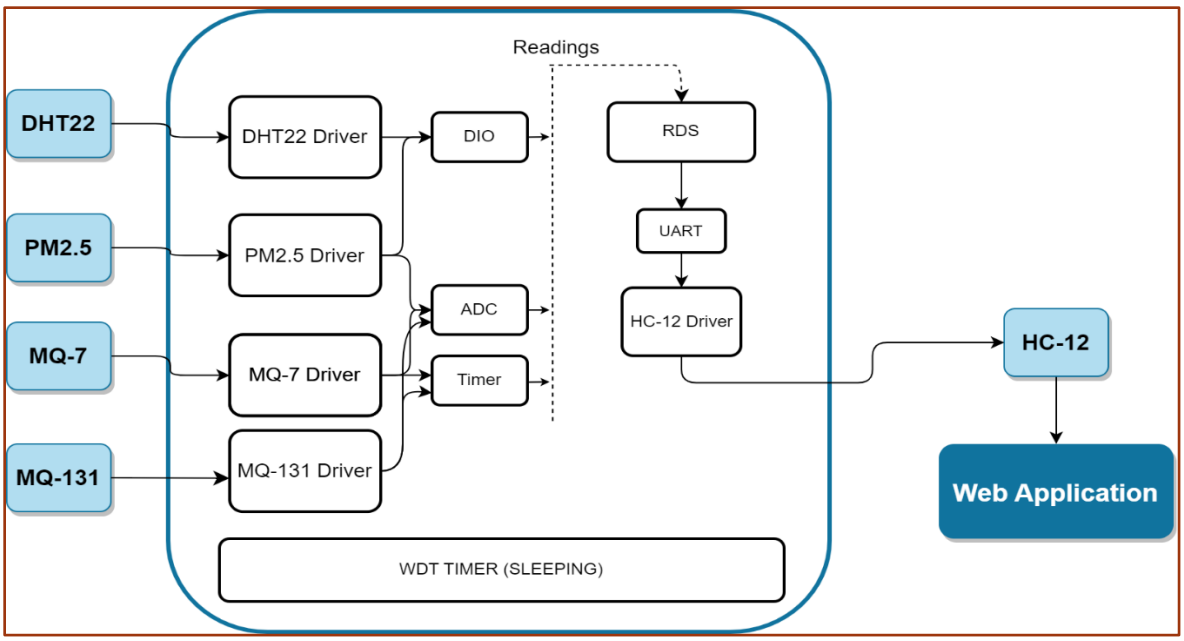


FIGURE 3: SYSTEM ARCHITECTURE

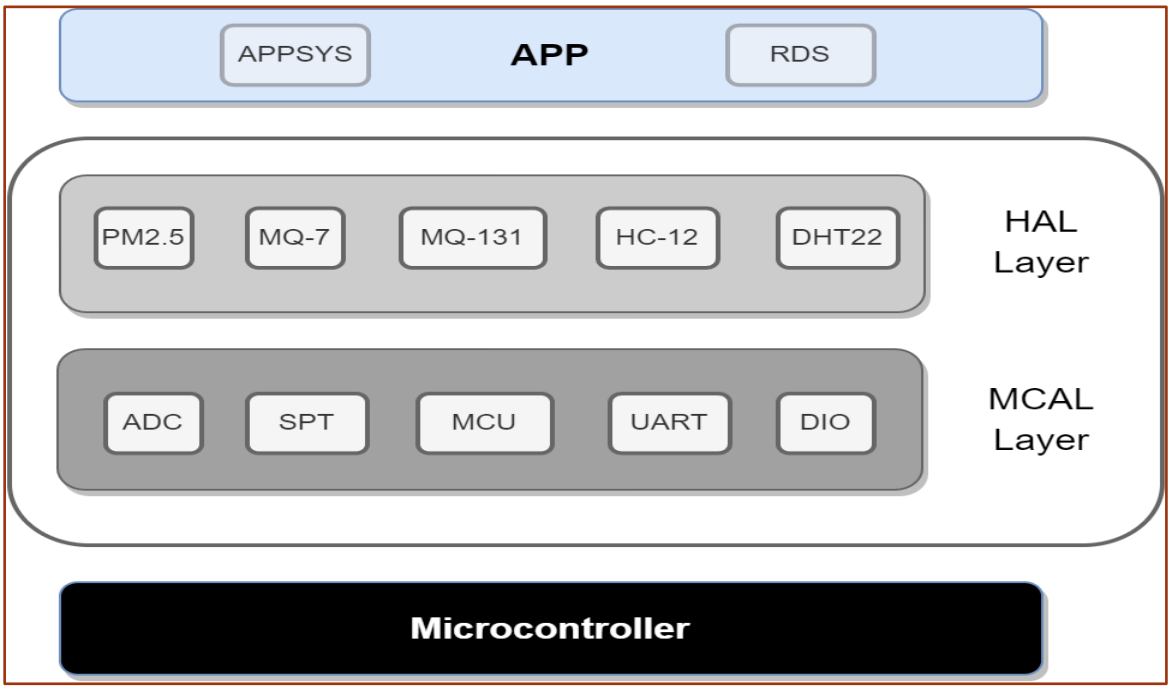


FIGURE 4: SYSTEM HARDWARE LAYERS

After explaining the main purpose of your project and the architecture, it is indeed a good idea to elaborate on each aspect of your device, starting with the hardware and then the software, in the following pages.

3. Hardware

We are using the Wireless sensor nodes technology to achieve our objectives, Wireless sensor nodes are small, self-contained devices equipped with sensors, processors, and wireless communication capabilities. These nodes are deployed in a wireless sensor network (WSN) to monitor and collect data from the physical environment. They play a vital role in gathering information, processing it, and wirelessly transmitting it to a central location or other nodes within the network.

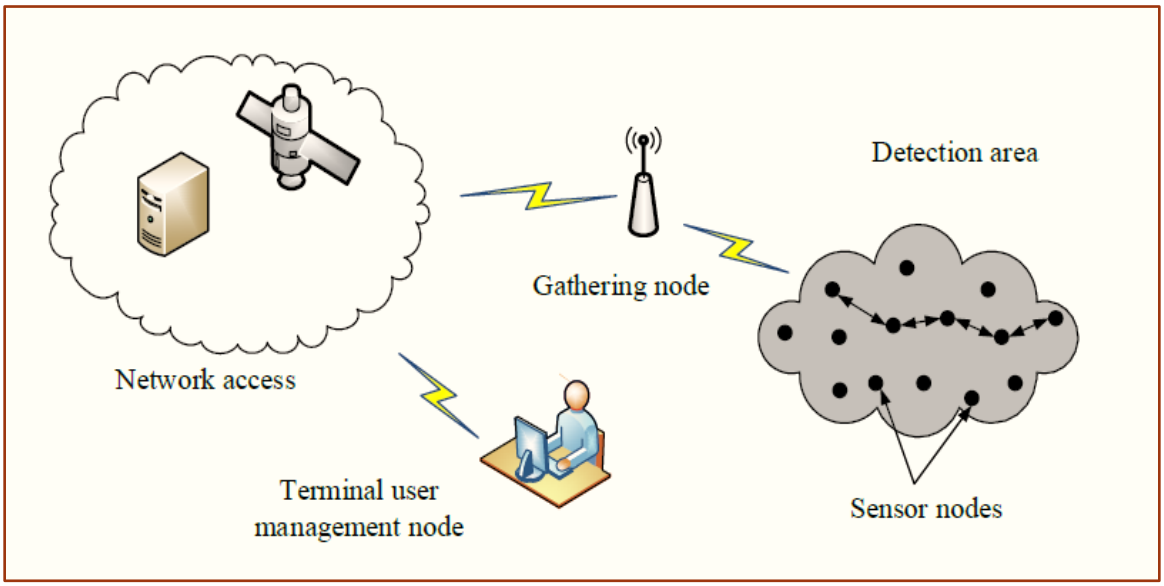


FIGURE 5: ILLUSTRATES THE COMPONENTS OF WSN

3.1 Wireless Sensor Nodes Components:

1-Sensors: Wireless sensor nodes are equipped with various sensors that capture data from the surrounding environment. These sensors can measure parameters such as temperature, humidity, light intensity, pressure, or pollution levels. The selection of sensors depends on the specific application.

2-Microcontroller/Microprocessor: The microcontroller or microprocessor serves as the brain of the sensor node. It processes the data collected by the sensors, performs computations, and controls the overall operation of the node. It may include memory, input/output interfaces, and power management capabilities.

3-Transceiver: The transceiver enables wireless communication between sensor nodes and other nodes within the network. It allows the nodes to transmit and receive data wirelessly, facilitating seamless communication and data exchange.

4-Power Source: Sensor nodes require a power source to operate. They can be powered by batteries, energy harvesting techniques (such as solar panels or kinetic energy harvesters), or a combination of both. Power management techniques are employed to optimize energy consumption and extend the lifespan of the nodes.

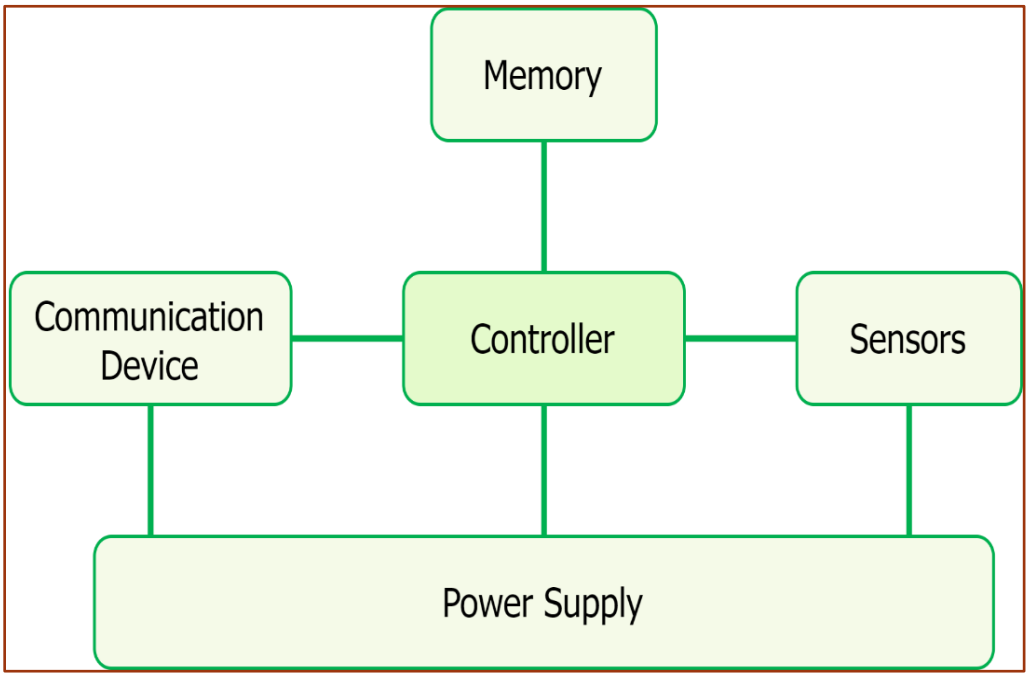


FIGURE 6: ARCHITECTURE OF HARDWARE

5-Memory: Wireless sensor nodes include memory components to store data temporarily or to log historical data for later retrieval and analysis. Memory capacity varies depending on the application requirements and the duration of data storage needed.

3.1.1 Microcontroller:

The Arduino Uno

is a popular microcontroller board based on the ATmega328P microcontroller. It is widely used in various electronics projects and prototyping due to its simplicity, versatility, and ease of use. Here's a brief description of the Arduino Uno:

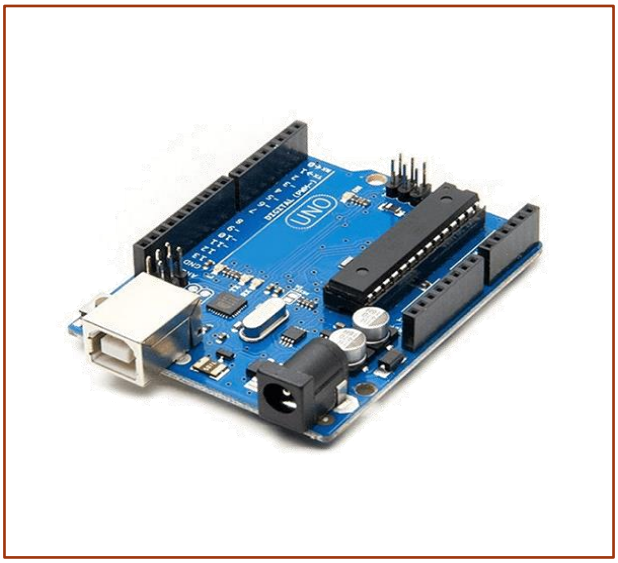


FIGURE 7: ARDUINO UNO

1. Microcontroller: The Arduino Uno, which is constructed by the ATmega328P microcontroller, which is an 8-bit AVR microcontroller with built-in flash memory, EEPROM, SRAM, and various I/O interfaces.

2. I/O Pins: The board features a total of 14 digital input/output pins, of which 6 can be used as PWM (Pulse Width Modulation) output pins. It also has 6 analog input pins.

3. Clock Speed: The ATmega328P microcontroller on the Arduino Uno operates at a clock speed of 16 MHz, allowing for fast execution of instructions and precise timing.

4. Programming: The Arduino Uno can be programmed using the Arduino programming language, which is based on C/C++. The Arduino IDE (Integrated Development Environment) provides a user-friendly interface for writing, compiling, and uploading code to the board.

5. Power Options: The Arduino Uno can be powered via a USB connection or an external power supply. It has a built-in voltage regulator that allows it to be powered with a range of voltages (typically 7-12V).

6. Communication: The board supports serial communication through USB, enabling easy connection to a computer for programming and serial communication with other devices. It also has dedicated pins for I2C and SPI communication.

7. Shields and Expansion: The Arduino Uno is compatible with a wide range of shields, which are additional circuit boards that can be stacked on top of the Uno, providing additional functionality such as WiFi, Ethernet, motor control, and more.

8. Community and Documentation: The Arduino Uno has a large and active community of users, providing extensive documentation, tutorials, and examples that make it easy for beginners to get started and for experienced users to develop complex projects.

The Arduino Uno is a versatile and beginner-friendly microcontroller board that has found applications in various fields, including robotics, home automation, Internet of Things (IoT), and educational projects. Its simplicity, extensive community support, and wide range of compatible shields make it a popular choice among hobbyists, students, and professionals alike.

The ESP32

is a powerful microcontroller module that combines Wi-Fi connectivity with the processing capabilities of a microcontroller. It is widely used in Internet of Things (IoT) applications where wireless communication and advanced processing are required. Here's a brief description of the ESP32 with Wi-Fi:

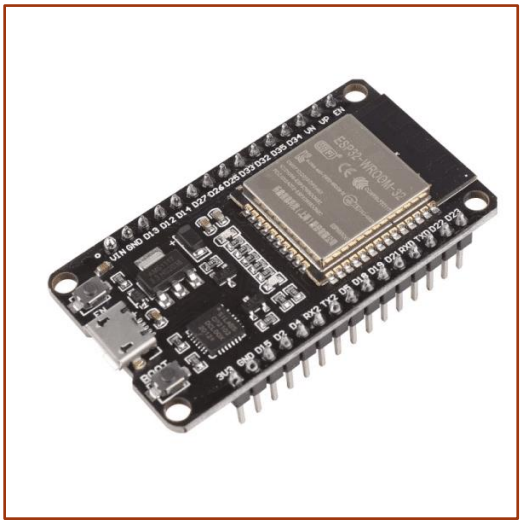


FIGURE 8: ESP32

1. Microcontroller and Wi-Fi: The ESP32 module integrates a dual-core Tensilica Xtensa LX6 microcontroller with Wi-Fi connectivity. This combination allows for both data processing and wireless communication capabilities in a single module.

2. Wi-Fi Connectivity: The ESP32 supports both 2.4 GHz and 5 GHz Wi-Fi bands, providing flexibility in connecting to different Wi-Fi networks. It supports various Wi-Fi standards, including 802.11b/g/n, and offers features such as station mode (connecting to an existing Wi-Fi network) and access point mode (creating its own Wi-Fi network).

3. Processing Power: The ESP32 features a dual-core microcontroller with clock speeds up to 240 MHz. This enables efficient multitasking and handling of complex tasks, making it suitable for applications that require data processing and real-time operation.

4. GPIO Pins and Interfaces: The ESP32 module provides a range of general-purpose input/output (GPIO) pins, which can be used for digital input/output, analog input, PWM output, and other purposes. It also supports interfaces such as SPI, I2C, UART, and ADC, allowing for easy integration with various sensors, displays, and other peripheral devices.

5. Memory and Storage: The ESP32 module typically comes with a generous amount of flash memory for storing program code and data. It also has built-in RAM for variable storage and buffer usage.

6. Programming: The ESP32 can be programmed using the Arduino IDE or other programming environments such as Espressif's ESP-IDF (IoT Development Framework). It offers a wide range of libraries and examples that simplify development and enable rapid prototyping.

7. Extensibility: The ESP32 module supports additional features through its GPIO pins, allowing for the connection of external modules or sensors. It also has support for Bluetooth, making it capable of Bluetooth Low Energy (BLE) communication.

8. Applications: The ESP32 with Wi-Fi is suitable for a wide range of IoT applications, including home automation, smart devices, sensor networks, industrial monitoring, and more. Its combination of Wi-Fi connectivity and processing power makes it an ideal choice for projects that require wireless communication and data processing capabilities.

The ESP32 with Wi-Fi has gained popularity among developers and hobbyists due to its versatility, low cost, and wide range of applications. Its combination of Wi-Fi connectivity, processing power, and extensive development resources make it an excellent choice for IoT projects that require wireless communication and advanced functionality.

3.1.2 Transceiver:

HC-12 Wireless Transceiver

The HC-12 is a wireless transceiver module that provides a simple and cost-effective solution for long-range communication between microcontrollers or other electronic devices. It operates on the 433MHz frequency band and offers reliable and stable wireless communication over distances of up to several kilometers in open environments. In this section, we will explore the working principle, features, and applications of the HC-12 wireless transceiver.

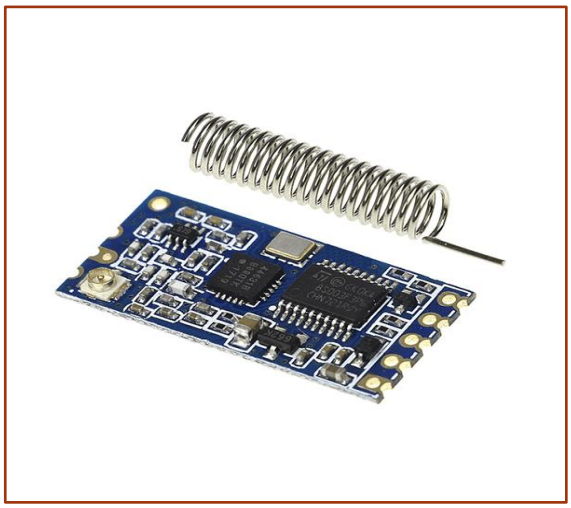


FIGURE 9: HC-12 SI4463 WIRELESS

1. Working Principle:

The HC-12 wireless transceiver module utilizes frequency-shift keying (FSK) modulation to transmit and receive data. It employs a serial communication interface (UART) to connect with microcontrollers or other devices. The module converts the data into wireless signals, which are transmitted through an antenna. On the receiving end, another HC-12 module receives the signals, decodes them, and provides the data to the receiving device.

2. Features and Specifications:

The HC-12 wireless transceiver module offers several features and specifications that make it suitable for long-range communication:

a. Long Communication Range: The module can achieve reliable communication over distances of up to several kilometers in open areas, depending on the environment and antenna configuration.

b. Low Power Consumption: The HC-12 module operates on low power, making it suitable for battery-powered applications and energy-efficient systems.

c. Adjustable Communication Parameters: The module allows for adjustable baud rate, frequency, and power level settings, providing flexibility to optimize communication performance based on the specific requirements of the application.

d. Error Correction Mechanism: The HC-12 module incorporates error correction techniques to improve data reliability, reducing the chances of data corruption during transmission.

e. Simple Serial Interface: The module communicates with microcontrollers or other devices using a straightforward UART interface, enabling easy integration into existing projects.

3. Applications:

The HC-12 wireless transceiver module finds applications in various fields where long-range wireless communication is required. Some common applications include:

a. Remote Sensing and Telemetry: The HC-12 module is used to establish communication links between remote sensors or telemetry systems and a central monitoring station. This enables data collection and monitoring in applications such as environmental monitoring, weather stations, and industrial automation.

b. Robotics and Drone Control: The module is utilized in robotics and drone applications to enable wireless control and communication between the control unit and the robotic or drone platform. It allows for remote control and data transmission in real-time.

c. Home Automation: The HC-12 module can be employed in home automation systems to enable wireless communication between different devices, such as sensors, actuators, and control panels. It facilitates remote control and monitoring of home automation functions.

d. Wireless Sensor Networks: The module is used in wireless sensor networks to establish communication between sensor nodes and a central gateway. It enables data transmission and coordination among multiple sensors in applications like smart agriculture, smart cities, and building automation.

4. Precautions and Limitations:

When using the HC-12 wireless transceiver module, it is important to consider the following precautions and limitations:

a. Regulatory Compliance: Ensure that the frequency band and power level used comply with local regulations governing wireless communication. Different regions may have specific restrictions and guidelines.

b. Interference: The HC-12 module operates in the 433MHz frequency band, which is shared with other devices. It is susceptible to interference from other devices operating in the same frequency range. Care should be taken to minimize potential interference sources.

c. Data Rate and Latency: The HC-12 module has a limited data rate and may introduce some latency due to the FSK modulation scheme. Consider the data transfer requirements of the application and select appropriate settings accordingly.

3.1.3 Sensors:

MQ-7 Sensor:

The MQ-7 sensor is a gas sensor widely used in various applications for detecting the presence of carbon monoxide (CO) gas. It belongs to a family of gas sensors known as Metal Oxide Semiconductor (MOS) sensors. The MQ-7 sensor is highly sensitive to carbon monoxide and can provide accurate measurements in real-time, making it an essential component in gas detection systems and environmental monitoring devices.

FIGURE 10: MQ-7 SENSOR

1. Working Principle:

The MQ-7 sensor operates based on the principle of chemical reaction and resistance changes. It consists of a tin dioxide (SnO2) sensing element that detects the presence of carbon monoxide in the surrounding environment. When exposed to carbon monoxide gas, a chemical reaction occurs on the surface of the sensing element, causing a change in its electrical conductivity.

2. Structure and Composition:

The MQ-7 sensor typically consists of a small ceramic substrate upon which the tin dioxide sensing element is deposited. This sensing element is equipped with electrodes that facilitate the measurement of electrical resistance. Additionally, the sensor is integrated with a heater element that ensures the sensing element operates at an optimal temperature for accurate detection.

3. Sensitivity and Detection Range:

The MQ-7 sensor is highly sensitive to carbon monoxide gas, capable of detecting concentrations in the range of 20 to 2000 parts per million (ppm). It is important to note that the sensitivity of the sensor may vary depending on factors such as temperature, humidity, and the presence of other gases.

4. Circuitry and Signal Processing:

To obtain reliable measurements, the MQ-7 sensor is usually connected to a microcontroller or an analog-to-digital converter (ADC). The sensor's resistance changes are converted into corresponding electrical signals, which can then be processed and analyzed by the connected electronic circuitry. The output signal from the sensor can be calibrated to provide accurate carbon monoxide gas concentration readings.

5. Applications:

The MQ-7 sensor finds applications in various fields due to its ability to detect carbon monoxide gas. Some common applications include:

a. Domestic Safety Systems: The sensor is used in residential and commercial settings to detect carbon monoxide leaks from gas heaters, stoves, or other fuel-burning appliances, ensuring the safety of occupants.

b. Industrial Safety: The sensor is employed in industrial environments to monitor carbon monoxide levels and provide early warning in areas where the gas may be present, such as factories, refineries, and warehouses.

c. Automotive Systems: The MQ-7 sensor is integrated into automotive exhaust systems to measure carbon monoxide emissions, helping to ensure compliance with environmental regulations and monitor engine performance.

d. Environmental Monitoring: It is used in environmental monitoring systems to measure carbon monoxide concentrations in the atmosphere and assess air quality in urban areas.

6. Precautions and Limitations:

While the MQ-7 sensor is a valuable tool for carbon monoxide detection, there are some precautions and limitations to consider:

a. Cross-Sensitivity: The sensor may exhibit cross-sensitivity to other gases, such as alcohol, methane, and hydrogen. Care should be taken to account for potential interference from these gases during measurements.

b. Calibration and Maintenance: Regular calibration and maintenance are necessary to ensure accurate and reliable readings over time. Sensor drift and aging can affect its performance, requiring periodic calibration checks.

c. Environmental Conditions: The sensor's sensitivity and response time may be influenced by environmental factors such as temperature, humidity, and air flow. These conditions should be considered during installation and usage.

GP2Y1014FU Sensor:

The GP2Y1014FU sensor is an optical dust sensor that is commonly used for measuring the concentration of fine particles in the air. It is an effective tool for monitoring air quality and is widely employed in applications such as air purifiers, HVAC systems and environmental monitoring devices. In this section, we will explore the working principle, features, and applications of the GP2Y1014FU sensor.

FIGURE 11: GP2Y1014FU SENSOR

1. Working Principle:

The GP2Y1014FU sensor operates on the principle of light scattering. It consists of an infrared (IR) LED emitter and a photodetector positioned at a specific angle. When particles pass through the sensor's detection area, they scatter the emitted infrared light. The photodetector captures the scattered light, and based on the intensity of the scattered light, the sensor calculates the concentration of particles present in the air.

2. Features and Specifications:

The GP2Y1014FU sensor offers several features and specifications that make it suitable for air quality monitoring applications:

a. Particle Size Detection Range: The sensor is capable of detecting particles within the range of 0.5 to 5.0 micrometers, which covers a broad spectrum of fine particles, including dust, smoke, pollen, and other airborne contaminants.

b. Output Signal: The sensor provides an analog output voltage that is proportional to the concentration of particles. This voltage can be easily interfaced with microcontrollers or other electronic systems for further processing and analysis.

c. Low Power Consumption: The sensor operates on low power, making it energy-efficient and suitable for applications that require long-term continuous monitoring.

d. Compact Size: The GP2Y1014FU sensor has a compact form factor, allowing for easy integration into various devices and systems.

3. Applications:

The GP2Y1014FU sensor finds applications in diverse areas where monitoring and control of air quality are essential. Some common applications include:

a. Air Purifiers: The sensor is commonly used in air purifiers to measure the concentration of fine particles in the air and activate the purification system accordingly.

b. HVAC Systems: Heating, ventilation, and air conditioning (HVAC) systems utilize the GP2Y1014FU sensor to monitor the air quality within buildings. This information helps optimize the system's performance and maintain a healthy indoor environment.

c. Environmental Monitoring: The sensor is employed in environmental monitoring systems to measure and analyze the concentration of particulate matter in outdoor air. This data aids in assessing air pollution levels and implementing appropriate mitigation strategies.

d. Industrial Processes: The GP2Y1014FU sensor is utilized in industrial settings to monitor and control dust emissions generated by manufacturing processes, ensuring compliance with environmental regulations and maintaining a safe working environment.

e. Research and Development: Researchers and scientists use the sensor for air quality studies, studying the impact of airborne particles on human health and the environment.

4. Precautions and Limitations:

When using the GP2Y1014FU sensor, it is important to consider the following precautions and limitations:

a. Calibration: The sensor may require periodic calibration to ensure accurate and reliable measurements. Calibration procedures provided by the manufacturer should be followed for optimal performance.

b. Sensor Placement: Proper positioning of the sensor is crucial for obtaining accurate readings. The sensor should be placed in an area representative of the air quality being monitored, away from direct sources of particles or obstructions.

c. Sensitivity to Environmental Factors: The sensor's performance may be influenced by environmental factors such as temperature, humidity, and airflow. These factors should be considered during sensor installation and operation.

MQ-131 Sensor:

The MQ-131 sensor is a gas sensor specifically designed for detecting ozone (O3) gas in the air. Ozone is a highly reactive and toxic gas that can be found in various environments,

including industrial settings, laboratories, and outdoor air. The MQ-131 sensor provides a means to measure ozone concentrations accurately and is widely used in ozone monitoring systems and environmental monitoring devices. In this section, we will explore the working principle,

features, and applications of the MQ-131 sensor.

1. Working Principle:

The MQ-131 sensor operates based on the principle of the electrochemical reaction. It consists of a sensing element composed of a tin dioxide (SnO2) layer and a heated electrode. When ozone gas comes into contact with the sensing element, it reacts with the tin dioxide layer, causing a change in the sensor's electrical conductivity. This change in conductivity is then converted into an output voltage signal, which can be measured and analyzed to determine the concentration of ozone gas in the surrounding environment.

2. Features and Specifications:

The MQ-131 sensor offers several features and specifications that make it suitable for ozone detection:

a. High Sensitivity: The sensor is highly sensitive to ozone gas, allowing for accurate detection and measurement of ozone concentrations in the range of several parts per billion (ppb) to parts per million (ppm).

b. Fast Response Time: The MQ-131 sensor provides a fast response time, enabling real-time monitoring of ozone levels.

c. Low Power Consumption: The sensor operates on low power, making it energy-efficient and suitable for applications that require continuous monitoring.

FIGURE 12: MQ-131 SENSOR

d. Analog Output: The sensor provides an analog voltage output that is proportional to the concentration of ozone gas. This output can be easily interfaced with microcontrollers or other electronic systems for further processing and analysis.

3. Applications:

The MQ-131 sensor finds applications in various fields where monitoring and control of ozone levels are essential. Some common applications include:

a. Environmental Monitoring: The sensor is used in environmental monitoring systems to measure and assess ozone levels in the atmosphere. This data is valuable for studying air quality, understanding the impact of ozone on human health and the environment, and implementing appropriate measures for pollution control.

b. Industrial Processes: The MQ-131 sensor is employed in industrial settings, particularly in industries where ozone is used or generated, such as water treatment plants, semiconductor manufacturing, and chemical processing facilities. It helps ensure compliance with safety regulations and facilitates the monitoring and control of ozone emissions.

c. Occupational Safety: The sensor is utilized in occupational safety systems to monitor ozone levels in work environments where ozone exposure is a concern, such as laboratories, research facilities, and healthcare settings. It helps protect workers from potential health risks associated with ozone exposure.

d. Ozone Generators: The MQ-131 sensor is integrated into ozone generators to monitor and control the output of ozone gas. This ensures the desired concentration of ozone is maintained for various applications, such as water treatment, disinfection, and sterilization processes.

4. Precautions and Limitations:

When using the MQ-131 sensor, it is important to consider the following precautions and limitations:

a. Calibration: Regular calibration of the sensor is necessary to ensure accurate and reliable measurements. Calibration procedures provided by the manufacturer should be followed to maintain optimal sensor performance.

b. Cross-Sensitivity: The sensor may exhibit cross-sensitivity to other gases, such as nitrogen dioxide (NO2) and sulfur dioxide (SO2). This cross-sensitivity should be taken into account when interpreting the sensor's output and distinguishing ozone gas from other gases.

c. Environmental Factors: The sensor's sensitivity and response time may be influenced by factors such as temperature, humidity, and air flow. These factors should be considered during installation and usage to obtain accurate ozone concentration readings.

DHT22 Sensor:

The DHT22 sensor, also known as the AM2302 sensor, is a digital temperature and humidity sensor widely used for environmental monitoring and control applications. It provides accurate and reliable measurements of temperature and humidity, making it a popular choice for both hobbyist projects and professional systems. In this section, we will explore the working principle, features, and applications of the DHT22 sensor.

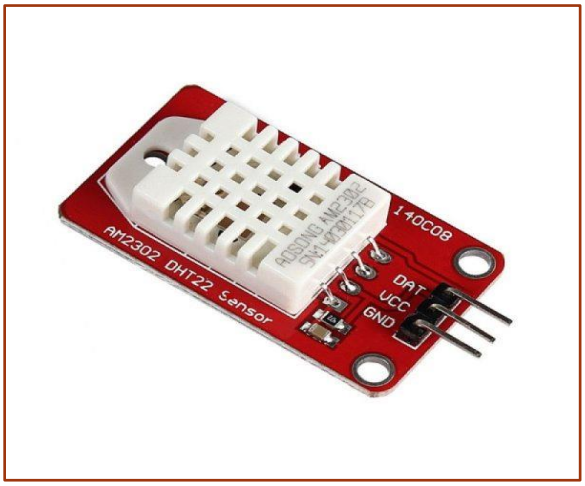


FIGURE 13: DHT-22 SENSOR

1. Working Principle:

The DHT22 sensor utilizes a capacitive humidity sensor and a thermistor for temperature measurement. The capacitive humidity sensor detects changes in the surrounding air's relative humidity, while the thermistor measures temperature based on the thermal characteristics of its materials. The sensor combines these measurements and provides a digital output that can be read and processed by microcontrollers or other electronic devices.

2. Features and Specifications:

The DHT22 sensor offers several features and specifications that make it suitable for temperature and humidity monitoring:

a. Temperature Range: The sensor can measure temperatures ranging from -40°C to 80°C with an accuracy of ±0.5°C.

b. Humidity Range: It can measure relative humidity ranging from 0% to 100% with an accuracy of ±2%.

c. Digital Output: The DHT22 sensor provides a digital output using a single-wire protocol, making it easy to interface with microcontrollers and other digital systems.

d. Low Power Consumption: The sensor operates on low power, making it suitable for battery-powered applications and energy-efficient systems

e. Long-Range Capability: The DHT22 sensor can transmit data reliably over long distances, allowing flexibility in sensor placement.

3. Applications:

The DHT22 sensor finds applications in various fields where temperature and humidity monitoring are crucial. Some common applications include:

a. Weather Stations: The sensor is commonly used in weather stations to measure temperature and humidity, providing valuable data for weather forecasting and monitoring.

b. Indoor Climate Control: The DHT22 sensor is employed in HVAC systems to monitor and control indoor temperature and humidity levels, ensuring comfortable and healthy environments in homes, offices, and other buildings.

c. Agricultural Monitoring: It is used in agriculture to monitor temperature and humidity in greenhouses, helping to optimize growing conditions for plants and prevent crop damage from extreme environmental conditions.

d. Data Logging: The DHT22 sensor is utilized in data logging systems to record temperature and humidity variations over time. This data can be analyzed to identify trends, patterns, and potential environmental issues.

e. Industrial Automation: The sensor is integrated into industrial automation systems to monitor and control temperature and humidity in manufacturing processes, storage facilities, and cleanrooms.

4. Precautions and Limitations:

When using the DHT22 sensor, it is important to consider the following precautions and limitations:

a. Accuracy: While the DHT22 sensor provides reasonably accurate measurements, it is not suitable for applications that require high precision. For critical applications, calibration and periodic verification may be necessary.

b. Response Time: The DHT22 sensor has a moderate response time, typically around 2 seconds for temperature and 2-5 seconds for humidity measurements. This response time should be considered when real-time monitoring is required.

c. Operating Conditions: The sensor's performance may be affected by factors such as ambient temperature, humidity, and airflow. It is essential to operate the sensor within its specified operating conditions for accurate measurements.

MT3608 Boost Converter:

The MT3608 is a boost converter module used for increasing voltage levels in electronic circuits. It operates on the principle of switching regulation, efficiently stepping up lower input voltages to higher output voltages. The module features a wide input voltage range, adjustable output voltage, high efficiency, and overcurrent protection. It finds applications in battery-powered devices, LED lighting, power supplies, Arduino and Raspberry Pi projects, and audio amplifiers. Users should

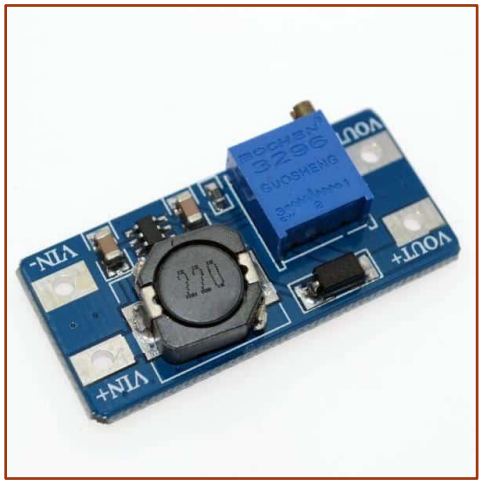


FIGURE 14: MT3608 BOOST CONVERTER

consider input/output voltage limits, current limits and heat dissipation for proper operation.

3.1.4 Power Supply

CN3065 Solar controller

The CN3065 is a compact and versatile solar power management chip designed for small-scale solar energy harvesting applications. It integrates several essential functions to efficiently charge and regulate power from solar panels to power a connected load or charge a battery. Here's a brief overview of the CN3065:

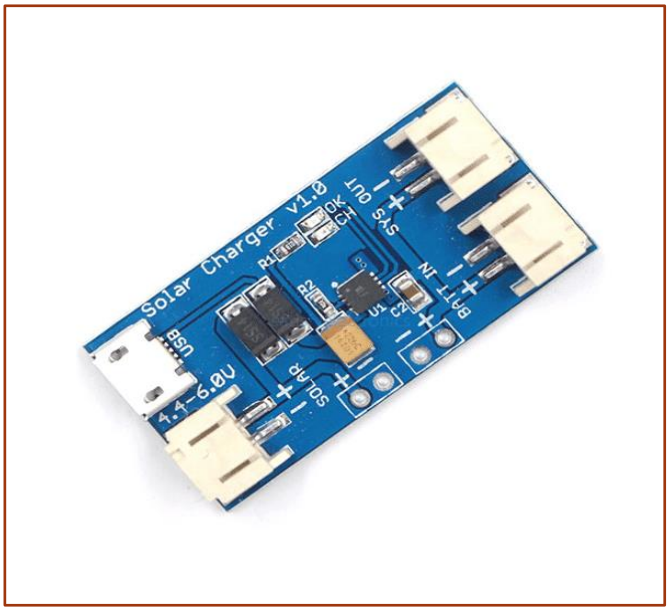


FIGURE 15: CN3065 SOLAR CONTROLLER

Solar Power Management:

The CN3065 is specifically designed for solar power applications, enabling efficient energy harvesting from solar panels.

Power Conversion and Regulation:

It incorporates a built-in boost converter that efficiently steps up the low input voltage from the solar panel to a higher usable voltage level.

The chip also includes a voltage regulator to provide a stable output voltage for powering the connected load or charging a battery.

Maximum Power Point Tracking (MPPT):

The CN3065 features MPPT functionality, allowing it to track the maximum power point of the connected solar panel.

MPPT helps optimize power extraction from the solar panel, ensuring maximum energy conversion efficiency.

Battery Charging:

The chip includes battery charging circuitry, allowing it to charge rechargeable batteries from the solar panel's output.

It supports various types of rechargeable batteries, including lithium-ion, lithium-polymer, and other commonly used battery chemistries.

Low Power Consumption:

The CN3065 is designed with low power consumption in mind, making it suitable for energy-efficient and battery-powered applications.

Protection and Safety Features:

The chip includes various protection features, such as overvoltage protection, overcurrent protection, and thermal shutdown, ensuring safe operation and preventing damage to connected components.

Compact Size and Easy Integration:

The CN3065 is available in a small form factor, making it easy to integrate into space-constrained applications.

It typically comes in a surface-mount package, allowing for convenient PCB assembly.

The CN3065 is commonly used in small-scale solar-powered devices, including solar chargers, portable solar power banks, solar garden lights, and other low-power solar energy harvesting applications. Its integrated power management functions make it a suitable choice for maximizing energy efficiency and optimizing power conversion from solar panels.

The Solar Panel

The 5.5V, 2.5W polycrystalline silicon epoxy solar panel with dimensions of 160x116mm. This type of solar panel is commonly used in small-scale solar power applications, such as portable solar chargers, outdoor lighting systems, and small electronic devices. Here's a brief description:

FIGURE 16: SOLAR PANEL 5.5V, 2.5W

The solar panel is made from polycrystalline silicon, which is a cost-effective and widely used material in solar cell production. The panel is designed to convert sunlight into electrical energy through the photovoltaic effect. When exposed to sunlight, the photons in the sunlight interact with the polycrystalline silicon material, generating an electric current.

The solar panel has a rated output voltage of 5.5V and a power output of 2.5W. The power output indicates the maximum power the panel can produce under ideal conditions. The dimensions of 160x116mm make it compact and suitable for applications where space is limited.

To calculate the time required to recharge a 3300mAh lithium-ion battery using this solar panel, we need to consider the power output of the panel and the charging efficiency. Assuming an 80% charging efficiency, the equation to calculate the charging time is as follows:

Charging time (hours) = Battery capacity (Ah) / (Charging current (A) * Charging efficiency)

Given:

Battery capacity: 3300mAh / 1000 = 3.3Ah

Charging current: 2.5W / 5.5V = 0.4545A

Using the equation:

Charging time = 3.3Ah / (0.4545A * 0.8) = 9.146 hours (approximately)

Therefore, it would take approximately 9.146 hours to recharge a 3300mAh lithium-ion battery using the 5.5V, 2.5W solar panel with an assumed charging efficiency of 80%. It's important to note that this is an estimated value and actual charging time may vary based on environmental conditions and the efficiency of the charging system.

Based on the power consumption rates of the components, we can calculate the total current draw in different time intervals. Here is the formal representation of the calculations:

1. Power Consumption Rates:

- MQ7: 0.045 mAh/s
- GPY2: 0.0056 mAh/s
- DHT22: 0.00070 mAh/s
- HC-12: 0.0045 mAh/s
- ATmega 328p (Idle): 0.0000018 mAh/s
- ATmega 328p (Active): 0.023 mAh/s

2. Current Consumption in 60 seconds:

Total current consumption = (0.045 + 0.0056 + 0.00070 + 0.0045 + 0.023) * 60 = 3.715 mA

3. Current Consumption in 14 minutes:

ATmega 328p (Idle) current consumption = 0.0000018 * 14 * 60 = 0.001512 mA

4. Total Current Consumption in 15 minutes:

Total current consumption = 3.715 mA + 0.001512 mA = 4.716 mA

5. Total Current Consumption in 1 hour:

Total current consumption = 4.716 mA * 4 = 18.864 mA

6. Battery Life Calculation:

Assuming a battery capacity of 3300 mAh:

Battery life = Battery capacity / Total current consumption

= 3300 mAh / 18.864 mA

≈ 174.89 hours (or approximately 220 hours)

Therefore, based on the given power consumption rates, the system is expected to have a battery life of approximately 220 hours (or 9.17 days) with a 3300 mAh battery.

---

4. Software

We developed a powerful and secure server using Express.js and MongoDB. This server acts as the backbone of my project, handling various functionalities and serving as the central hub for data storage and retrieval. Leveraging a range of libraries and packages, the server ensures robustness, efficiency, and reliability.

4.1 Middleware

To build the server, I utilized a wide array of technologies and libraries, each serving a specific purpose in enhancing the server's functionality and security. Here are some of the key dependencies and their roles:

1. Express.js (v4.18.1): Express.js is a fast and minimalist web application framework for Node.js. It provides a robust set of features for building web servers and APIs, making it an ideal choice for this project.

2. MongoDB (v5.13.14): MongoDB is a highly flexible and scalable NoSQL database system. It offers a document-based data model, allowing for efficient storage and retrieval of data. MongoDB integrates seamlessly with Node.js, making it an excellent choice for this server.

3. AWS SDK (v2.1313.0): The AWS SDK provides a comprehensive set of tools and services for interacting with various Amazon Web Services. This library enables seamless integration with AWS services, allowing the server to leverage cloud-based functionalities.

4. Axios (v1.3.2): Axios is a popular HTTP client for making requests from the server to external APIs or services. It simplifies the process of sending HTTP requests and handling responses, enhancing the server's ability to communicate with external systems.

5. Bcrypt.js (v2.4.3): Bcrypt.js is a library used for password hashing and encryption. It ensures the security of sensitive user information by securely storing passwords in the database.

6. Compression (v1.7.4): Compression is a middleware that enables gzip compression of HTTP responses, reducing the size of transmitted data and improving server performance.

7. Cookie-parser (v1.4.6): Cookie-parser is a middleware used for parsing cookies attached to incoming requests. It simplifies cookie handling and enables the server to read and set cookies as needed.

8. CORS (v2.8.5): CORS (Cross-Origin Resource Sharing) is a mechanism that allows restricted resources on a web page to be requested from another domain outside the domain from which the resource originated. This package enables cross-origin communication between the server and client-side applications.

9. Dotenv (v16.0.1): Dotenv is a zero-dependency module used for loading environment variables from a .env file into process.env. It helps manage sensitive information and configuration details securely.

10. Express-mongo-sanitize (v2.2.0): Express-mongo-sanitize is a middleware that sanitizes user-supplied data to prevent MongoDB query injection attacks. It removes any potential malicious characters or operators from user input, ensuring data security.

11. Express-rate-limit (v6.6.0): Express-rate-limit is a middleware that limits the number of requests a client can make within a specified time frame. It helps protect the server from abusive or malicious activities, such as brute-force attacks or excessive API requests.

12. Helmet (v6.0.0): Helmet is a collection of middleware functions that enhance the security of Express.js applications. It helps set various HTTP headers to mitigate common

security vulnerabilities, such as cross-site scripting (XSS) and cross-site request forgery (CSRF) attacks.

13. Hpp (v0.2.3): Hpp (HTTP Parameter Pollution) is a middleware that prevents HTTP parameter pollution attacks. It ensures that only a single value is accepted for a given parameter, protecting against potential security risks.

14. Html-to-text (v9.0.3): Html-to-text

is a library used to convert HTML content to plain text? It is particularly useful when processing user-generated content or parsing HTML emails, as it helps sanitize and extract readable text from HTML markup.

15. Jsonwebtoken (v9.0.0): Jsonwebtoken, also known as JWT, is a library used for creating, signing, and verifying JSON Web Tokens. JWTs are commonly used for authentication and authorization purposes, allowing secure transmission of claims between parties.

16. Mongoose (v5.13.14): Mongoose is an Object Data Modeling (ODM) library for MongoDB and Node.js. It provides a straightforward way to interact with MongoDB, defining schemas, models, and performing database operations.

17. Morgan (v1.10.0): Morgan is a middleware used for logging HTTP requests in the server's console or log files. It provides detailed information about each request, including the request method, URL, response status, and response time.

18. Mqtt (v4.3.7): MQTT (Message Queuing Telemetry Transport) is a lightweight messaging protocol used for IoT (Internet of Things) communication. The MQTT package enables the server to establish MQTT connections and publish/subscribe to MQTT topics.

19. Multer (v1.4.5-lts.1): Multer is a middleware used for handling multipart/form-data, primarily used for file uploads. It simplifies the process of handling file uploads, including storing files on the server and managing file metadata.

20. Nodemailer (v6.7.8): Nodemailer is a module used for sending emails from Node.js applications. It supports various email services and protocols, allowing the server to send transactional emails, notifications, or password reset emails.

21. Pug (v3.0.2): Pug, formerly known as Jade, is a high-performance template engine for Node.js. It provides a concise and expressive syntax for generating HTML templates, making it easier to dynamically render HTML content on the server.

22. Sharp (v0.31.3): Sharp is a powerful image-processing library for Node.js. It allows the server to manipulate images, perform resizing, cropping, compression, and apply various filters or transformations.

23. Slugify (v1.6.5): Slugify is a library used to convert strings into URL-friendly slugs. It removes special characters, replaces spaces with hyphens, and ensures that the resulting string is suitable for use in URLs.

24. Validator (v13.7.0): Validator is a library used for data validation and sanitization. It provides various validation methods for checking the format of strings, numbers, emails, URLs, and more, ensuring the integrity and validity of user input.

25. Xss-clean (v0.1.1): Xss-clean is a middleware used to sanitize user input and prevent cross-site scripting (XSS) attacks. It removes or escapes potentially malicious scripts or HTML tags from user-supplied data, ensuring secure handling of user input.

This middleware and packages significantly enhance the functionality, security, and performance of the Express.js and MongoDB server in my graduation project. By leveraging these dependencies, I have developed a robust and secure server capable of handling various tasks, including authentication, data storage, API communication, file handling, and email functionality.

4.2 Schemas

We have developed a server that incorporates several schemas to handle different aspects of the application. These schemas serve as the foundation for organizing and managing data within the system. In this section, I will provide an overview of each schema, including the User Feedback Schema, Gas Effects Schema, User Schema, Node Schema, and MQTT Schema (for readings from a hardware device). Each schema plays a crucial role in defining the structure and behavior of the corresponding data entities, allowing for efficient data storage, retrieval, and manipulation.

4.2.1 Feedback Schema

The Feedback Schema is designed to store feedback information provided by users. It captures details such as the user's email, name, subject, feedback details, seen status, and timestamp.

- **email** (String):
  - Type: String
  - Required: Yes
  - Validation: The email field should be a valid email address.
  - Description: Represents the email address of the user who provided the feedback.
- **name** (String):
  - Type: String
  - Required: Yes
  - Description: Represents the name of the user who provided the feedback.
- **subject** (String):
  - Type: String
  - Required: Yes
  - Description: Represents the subject of the feedback.
- **details** (String):
  - Type: String

    - Required: Yes
    - Description: Represents the details or content of the feedback provided by the user.
- **seen** (Boolean):
    - Type: Boolean
    - Default: False
    - Description: Represents whether the feedback has been seen or not. It is set to False by default.
- **sendedAt** (Date):
    - Type: Date
    - Default: Current date and time
    - Description: Represents the timestamp when the feedback was submitted. It is set to the current date and time by default.

The Feedback Schema enables the storage and retrieval of feedback information from users. It captures essential details such as the user's email, name, subject, feedback details, seen status, and submission timestamp. This schema provides a structured approach to managing feedback data within the application.

4.2.2 Gas Effects Schema:

The Gas Effects Schema is designed to store information about the effects of various gases on human health and the environment. This schema provides a comprehensive overview of each gas, including its name, information, short-term exposure effects, long-term exposure effects, sources, prevention measures, risk reduction strategies, and factors that may affect its behavior.

Field Descriptions:

1. **Name**:
    - Type: String
    - Required: Yes
    - Unique: Yes
    - Trim: Yes
    - Description: Represents the name of the gas. It provides a unique identifier for each gas.
2. **Information**:
    - Type: String
    - Required: Yes
    - Description: Contains information about the gas, including its characteristics, properties, and potential effects.
3. **Short-Term Exposure**:
    - Type: Array of Strings
    - Required: Yes
    - Description: Lists the short-term health effects associated with exposure to the gas. It includes symptoms or conditions that may arise after brief exposure.
4. **Long-Term Exposure**:
    - Type: Array of Strings
    - Required: Yes

- Description: Specifies the long-term health effects resulting from prolonged exposure to the gas. It includes chronic conditions or diseases that may develop over time.

5. **Sources**:
    - Type: Array of Strings
    - Required: Yes
    - Description: Identifies the sources or origins of the gas. It includes natural processes, human activities, or industrial emissions that contribute to its presence in the environment.
6. **Prevention**:
    - Type: Array of Strings
    - Required: Yes
    - Description: Outlines preventive measures or actions that can be taken to minimize exposure to the gas. It includes strategies for individuals, communities, or industries to reduce its impact.
7. **Risk Reduction**:
    - Type: Array of Strings
    - Required: Yes
    - Description: Provides recommendations or methods to mitigate the risks associated with the gas. It includes steps to minimize its harmful effects on human health and the environment.
8. **Affected By**:
    - Type: Array of Strings
    - Required: Yes
    - Description: Specifies factors or conditions that may affect the behavior or impact of the gas. It includes environmental factors, interactions with other substances, or specific scenarios that influence its behavior.

The Gas Effects Schema enables the storage and retrieval of comprehensive information about various gases and their effects. It allows users to access detailed insights into the health and environmental implications of different gases, facilitating informed decision-making and proactive measures for gas management and risk reduction.

4.2.3 MQTT Schema (Readings from Hardware Device Schema):

The MQTT Schema, also known as the Readings from Hardware Device Schema, is designed to store and manage the readings obtained from a hardware device through an MQTT broker. This schema captures various sensor data related to carbon monoxide (CO) level, CO Air Quality Index (AQI), particulate matter 2.5 (PM2.5) level, PM2.5 AQI, ozone (O3) level, O3 AQI, temperature, humidity, overall Air Quality Index (AQI), reading timestamp, and geolocation.

Here is a detailed description of each field within the MQTT Schema:

1. **node** (mongoose.Schema.ObjectId):
    - Data Type: mongoose.Schema.ObjectId
    - Reference: 'Node'
    - Requirement: Mandatory
    - Description: This field represents the associated node for which the MQTT readings have been received. It references the Node model using the ObjectId.
2. **co** (Number):
    - Data Type: Number
    - Minimum Value: 0
    - Maximum Value: 1000
    - Description: The 'co' field stores the carbon monoxide (CO) level reading obtained from the hardware device.
3. **coAQI** (Number):
    - Data Type: Number
    - Minimum Value: 0
    - Maximum Value: 500
    - Description: The 'coAQI' field stores the Air Quality Index (AQI) corresponding to the carbon monoxide (CO) level obtained from the hardware device.

4. **pm2_5** (Number):
    - Data Type: Number
    - Minimum Value: 0
    - Maximum Value: 1000
    - Description: The 'pm2_5' field stores the particulate matter 2.5 (PM2.5) level reading obtained from the hardware device. PM2.5 refers to the concentration of airborne particles with a diameter of 2.5 micrometers or smaller.

5. **pm2_5AQI** (Number):
    - Data Type: Number
    - Minimum Value: 0
    - Maximum Value: 500
    - Description: The 'pm2_5AQI' field stores the Air Quality Index (AQI) corresponding to the particulate matter 2.5 (PM2.5) level obtained from the hardware device.
6. **o3** (Number):
    - Data Type: Number
    - Minimum Value: 0
    - Maximum Value: 1000
    - Description: The 'o3' field stores the ozone (O3) level reading obtained from the hardware device.
7. **o3AQI** (Number):
    - Data Type: Number
    - Minimum Value: 0
    - Maximum Value: 500
    - Description: The 'o3AQI' field stores the Air Quality Index (AQI) corresponding to the ozone (O3) level obtained from the hardware device.

8. **temp** (Number):
    - Data Type: Number
    - Minimum Value: -100
    - Maximum Value: 100
    - Description: The 'temp' field stores the temperature reading obtained from the hardware device.
9. **hum** (Number):
    - Data Type: Number
    - Minimum Value: 0
    - Maximum Value: 100
    - Description: The 'hum' field stores the humidity reading obtained from the hardware device.
10. **AQI** (Number):
    - Data Type: Number
    - Minimum Value: 0
    - Maximum Value: 500
    - Description: The 'AQI' field stores the overall Air Quality Index (AQI) obtained from the hardware device. It represents the comprehensive assessment of the air quality based on multiple factors and pollutant levels.
11. **readTime** (Date):
    - Data Type: Date
    - Description: The 'readTime' field stores the timestamp when the MQTT readings were received. It captures the date and time of the reading, enabling chronological analysis and tracking of the sensor data.
12. **Geolocation** (Object):
    - Data Type: Object
    - Description: The 'Geolocation' field represents the geolocation information of the node associated with the MQTT readings. It includes the type, which is set to 'Point' by default, and the coordinates (latitude and longitude) of the node's location. This information facilitates spatial representation and mapping of the sensor data.

The MQTT Schema provides a structured and organized approach to store and retrieve readings obtained from a hardware device through an MQTT broker. By utilizing this schema, applications can effectively manage and analyze sensor data, enabling comprehensive monitoring and assessment of environmental conditions.

4.2.4 Node Schema:

The Node Schema is designed to store and manage information related to nodes within the system. A node can refer to a physical device or a logical entity associated with data collection and monitoring. This schema encompasses multiple fields that capture various details about a node, including its name, location, node ID, status, geolocation, battery level, link status, last active time, owner, sensor readings, and predictions.

Below is a detailed description of each field within the Node Schema:

1. **name** (String):
    - Data Type: String
    - Trimming: Enabled
    - Maximum Length: 20 characters
    - Minimum Length: 3 characters
    - Description: The 'name' field represents the name of the node. It should be a concise description or identifier for the node.
2. **location** (String):
    - Data Type: String
    - Trimming: Enabled
    - Description: The 'location' field represents the location of the node. It can be a textual representation of the node's physical or logical position.
3. **nodeId** (String):
    - Data Type: String
    - Requirement: Mandatory
    - Trimming: Enabled

- Uniqueness: Unique
- Description: The 'nodeId' field represents the unique identifier of the node. It is required and should be unique to identify each node uniquely.

4. **status** (String):
    - Data Type: String
    - Trimming: Enabled
    - Default Value: 'active'
    - Enumeration: ['active', 'inactive']
    - Description: The 'status' field represents the status of the node. It can be either 'active' or 'inactive', and it defaults to 'active' if not provided.

5. **Geolocation** (Object):
    - Data Type: Object
    - Description: The 'Geolocation' field represents the geolocation information of the node. It includes the type (default: 'Point') and the coordinates (latitude and longitude) of the node's location.
6. **battery** (Number):
    - Data Type: Number
    - Minimum Value: 0
    - Maximum Value: 100
    - Description: The 'battery' field represents the battery level of the node. It should be a number between 0 and 100, indicating the percentage of battery remaining.
7. **linkStatus** (String):
    - Data Type: String
    - Trimming: Enabled
    - Default Value: 'not Linked'
    - Enumeration: ['Linked', 'not Linked']
    - Description: The 'linkStatus' field represents the link status of the node. It can be either 'Linked' or 'not Linked', and it defaults to 'not Linked' if not provided.
8. **lastActiveTime** (Date):
    - Data Type: Date

- Description: The 'lastActiveTime' field represents the timestamp of the node's last activity. It stores the date and time when the node was last active.

9. **owner** (mongoose.Schema.ObjectId):
    - Data Type: mongoose.Schema.ObjectId
    - Reference: 'User'
    - Description: The 'owner' field represents the owner of the node. It references the associated User document using the ObjectId and refers to the 'User' model.

10. **reads** (Object):
    - Data Type: Object
    - Description: The 'reads' field represents the sensor readings of the node. It includes various fields such as CO level, CO AQI, PM2.5 level, PM2.5 AQI, O3 level, O3 AQI, temperature, humidity, and overall AQI.

11. **predictions** (Array of Objects):
    - Data Type: Array of Objects
    - Description: The 'predictions' field represents the predictions for the node's air quality. Each prediction object contains fields for the date, time, AQI, and PM2.5 level.

The Node Schema provides a structured and organized approach to storing and managing information related to nodes within the system. By utilizing this schema, applications can effectively track and analyze data from nodes, including their identification, status, location, sensor readings, and predictions. This schema facilitates efficient data management and enables comprehensive monitoring and assessment of environmental conditions.

4.2.5 User Schema:

The User Schema is designed to store and manage user information within the system. It encompasses multiple fields that capture details about a user, including their name, email,

photo, role, password, email verification status, account activation status, associated nodes, gender, age, password change history, password reset token, password reset expiration, email verification token, and email verification expiration.

Below is a detailed description of each field within the User Schema:

1. **name** (String):
    - Data Type: String
    - Requirement: Mandatory
    - Uniqueness: Unique
    - Trimming: Enabled
    - Maximum Length: 40 characters
    - Minimum Length: 5 characters
    - Description: The 'name' field represents the user's name. It is a required field and should be unique. It should have a length of at least 5 characters and not exceed 40 characters.
2. **email** (String):
    - Data Type: String
    - Requirement: Mandatory
    - Uniqueness: Unique
    - Trimming: Enabled
    - Lowercasing: Enabled
    - Email Validation: Enabled
    - Description: The 'email' field represents the user's email address. It is a required field, should be unique, and undergoes email validation to ensure a valid email format.
3. **photo** (String):
    - Data Type: String
    - Default Value: URL to the default profile picture
    - Description: The 'photo' field represents the URL of the user's profile picture. It has a default value pointing to a default profile picture if no custom picture is provided.
    -

4. **role** (String):
    - Data Type: String
    - Enumeration: ['admin', 'user', 'guide', 'lead-guide']
    - Default Value: 'user'
    - Description: The 'role' field represents the user's role within the system. It can have one of the specified roles and defaults to 'user' if not provided.
5. **password** (String):
    - Data Type: String
    - Requirement: Mandatory
    - Minimum Length: 8 characters
    - Select: Disabled
    - Description: The 'password' field represents the user's password. It is a required field and should have a minimum length of 8 characters. The 'select' option is disabled, ensuring the password is not returned in query results.
6. **passwordConfirm** (String):
    - Data Type: String
    - Requirement: Mandatory
    - Minimum Length: 8 characters
    - Select: Disabled
    - Description: The 'passwordConfirm' field represents the user's password confirmation. It is a required field and should match the 'password' field. The 'select' option is disabled, ensuring it is not returned in query results.
7. **emailVerified** (Boolean):
    - Data Type: Boolean
    - Default Value: false
    - Description: The 'emailVerified' field indicates whether the user's email has been verified. It has a default value of 'false' and is set to 'true' once the email verification process is completed.
8. **active** (Boolean):
    - Data Type: Boolean
    - Default Value: true

- Description: The 'active' field indicates whether the user's account is active. It has a default value of 'true' and can be set to 'false' to deactivate the user's account.

9. **NodesId** (Array of mongoose.Schema.ObjectId):
    - Data Type: Array of mongoose.Schema.ObjectId
    - Reference: 'Node'
    - Description: The 'NodesId' field represents the associated nodes for the user. It stores an array of ObjectIds referencing the Node model, indicating the nodes associated with the user.

10. **gender** (String):
    - Data Type: String
    - Requirement: Mandatory
    - Enumeration: ['Male', 'Female']
    - Description: The 'gender' field represents the user's gender. It is a required field and can have one of the specified genders.
11. **age** (Number):
    - Data Type: Number
    - Requirement: Mandatory
    - Validation: Minimum age of 18
    - Description: The 'age' field represents the user's age. It is a required field and should be a number greater than or equal to 18.
12. **passwordChangedAt** (Date):
    - Data Type: Date
    - Description: The 'passwordChangedAt' field stores the timestamp when the user's password was last changed.
13. **passwordResetToken** (String):
    - Data Type: String
    - Description: The 'passwordResetToken' field stores the token generated for password reset functionality.
14. **passwordResetExpires** (Date):

- Data Type: Date
- Description: The 'passwordResetExpires' field stores the expiration time for the password reset token.

15. **emailVerificationToken** (String):

- Data Type: String
- Description: The 'emailVerificationToken' field stores the token generated for email verification.

16. **emailVerificationExpires** (Date):

- Data Type: Date
- Description: The 'emailVerificationExpires' field stores the expiration time for the email verification token.

The User Schema enables the storage and management of user information within the system. It provides a structured representation of user data, including their personal details, authentication-related fields, and associated nodes. By utilizing this schema, applications can effectively handle user registration, authentication, role management, and account-related functionalities.

4.2.6 Summary

These schemas collectively contribute to the overall functionality, organization, and efficiency of the server in my graduation project. By utilizing each schema appropriately, the server can handle different data entities and provide a robust foundation for data management, analysis, and interaction within the application.

4.3 Routes

Feedback router

The "/api/v1/feedback" router is a crucial component of our application's API structure. It handles operations related to user feedback, including retrieving all feedbacks, submitting new feedback, updating feedback status, and deleting feedback entries. The router incorporates middleware functions for authentication and access restriction, ensuring that only authorized users can interact with the routes. Admin users have additional privileges to retrieve all feedbacks, while any user can submit feedback. Specific feedback can be retrieved, updated, or deleted based on its ID. This router provides a structured and secure approach to manage user feedback, facilitating effective communication and continuous improvement within our application.

GasEffect router

The "/api/v1/gassesEffect" router is an essential part of our application's API structure. It handles operations related to gases effect, including retrieving all gases, adding new gases, updating gas information, and deleting gases. The router incorporates middleware functions for authentication and access restriction, ensuring that only authorized users can interact with the routes. Admin users have exclusive access to retrieve all gases and add new ones. Specific gas information can be retrieved, updated, or deleted based on its ID. This router provides a reliable and secure mechanism for managing gases effect data within our application, contributing to improved analysis and decision-making processes.

Node router

The "/api/v1/node" router is a fundamental part of our application's API structure. It handles operations related to nodes, including retrieving all nodes, adding new nodes, updating node information, and deleting nodes. The router incorporates middleware functions for authentication and authorization, ensuring secure access to the routes. Admin users have privileged access to perform operations on nodes, such as deactivating or activating nodes. Additionally, users can retrieve specific node information, update nodes for themselves, retrieve readings for a specific node, and find nodes within a certain distance from a given center point. This router provides a comprehensive set of functionalities to manage and interact with node data within our application.

User router

The "/api/v1/users" router is an essential part of our application's API structure. It handles user-related operations such as user registration, login, profile management, password reset, and user data manipulation. The router provides routes for user authentication, updating user information, deleting user data, and performing admin-specific actions. It ensures secure access and authentication using middleware functions. This router plays a crucial role in managing user accounts and facilitating user interactions within our application.

4.4 Functional Requirements Specification & Diagrams:

1. Functional Requirements:

REQ - 1	User & Admin can login from home page, users will redirect to Dashboard page while admins will redirect to Nodes page.
REQ - 2	User & Admin can sign up a new user from home page.
REQ - 3	View of readings and predictions: User can view readings of sensors & can also view the predictions of the following 12 hours at Dashboard page.
REQ - 4	Add node: User & Admin can add node from manage page.
REQ - 5	Delete node: User & Admin can delete node from manage page.
REQ - 6	Link node: User & Admin can link node to get data from it.
REQ - 7	Unlink node: User & Admin can unlink node to prevent getting data from it.
REQ - 8	View Nodes' Places: User & Admin can view nodes on map in a range of 10km.

REQ - 9	View Information of Gas pollutants: User can view information about gas pollutants and the effects of each one of them.
REQ - 10	Add Feedback: User & Admin can add feedback.
REQ - 11	Delete User: Admin can delete user.
REQ - 12	Delete Node: Admin can delete node from system if node is not linked by a user.
REQ - 13	Create Node: Admin can create node in nodes page so users can use it.
REQ - 14	Admin can see and delete all feedbacks in feedback page.
REQ - 15	Admin & User can update their profile's data.
REQ - 16	ML model take readings of AQI from last 25 hours and predict the following 12 hours of AQI readings.

2. Non-Functional Requirements:

REQ-17	Security is paramount in an Air Quality Monitoring System, employing robust measures like encryption, authentication, and access control to safeguard sensitive data from unauthorized access or breaches, ensuring the privacy and integrity of user's information. Regular security audits and updates help to identify and address vulnerabilities, providing a safe and secure environment for data management and user confidentiality.
REQ-18	Availability ensures that the site runs constantly in all browsers. The web application could be running at every time.
REQ-19	In an Air Quality Monitoring System ensures that the software is designed and implemented in a way that allows for easy updates, bug fixes, and modifications as needed, ensuring smooth operation and adaptability to changing

	requirements. By prioritizing maintainability, the system can be efficiently managed, reducing downtime and enhancing overall user experience for users & admins.
REQ-20	The system shall utilize a non-relational database and interact with it through APIs. Queries upon the database shall be performed in less than 3 seconds, ensuring efficient data retrieval and processing. The system shall be designed to be fast and responsive, providing users with a seamless and smooth experience.
REQ-21	A user & admin shall be able to log into the system at any instance of time.

3. **System Stakeholders:**

   The system Stakeholders are classified into 3 groups, there are,

   - **User**
   - **Admin**
   - **ML model**

4. **Actors & Goals:**

   - **User**: is the main actor & has multiple functions but the main part is that they can see & read data coming from sensors and the predictions coming from ML model
   - **ML model**: has only one function which is getting last 25 readings and predicting another 12 readings for the following 12 hours.
   - **Admin**: can see the feedback of users and admins and see the status of all nodes and users.

4.4.1 Software Diagrams:

4.4.2 Use Case Diagram:

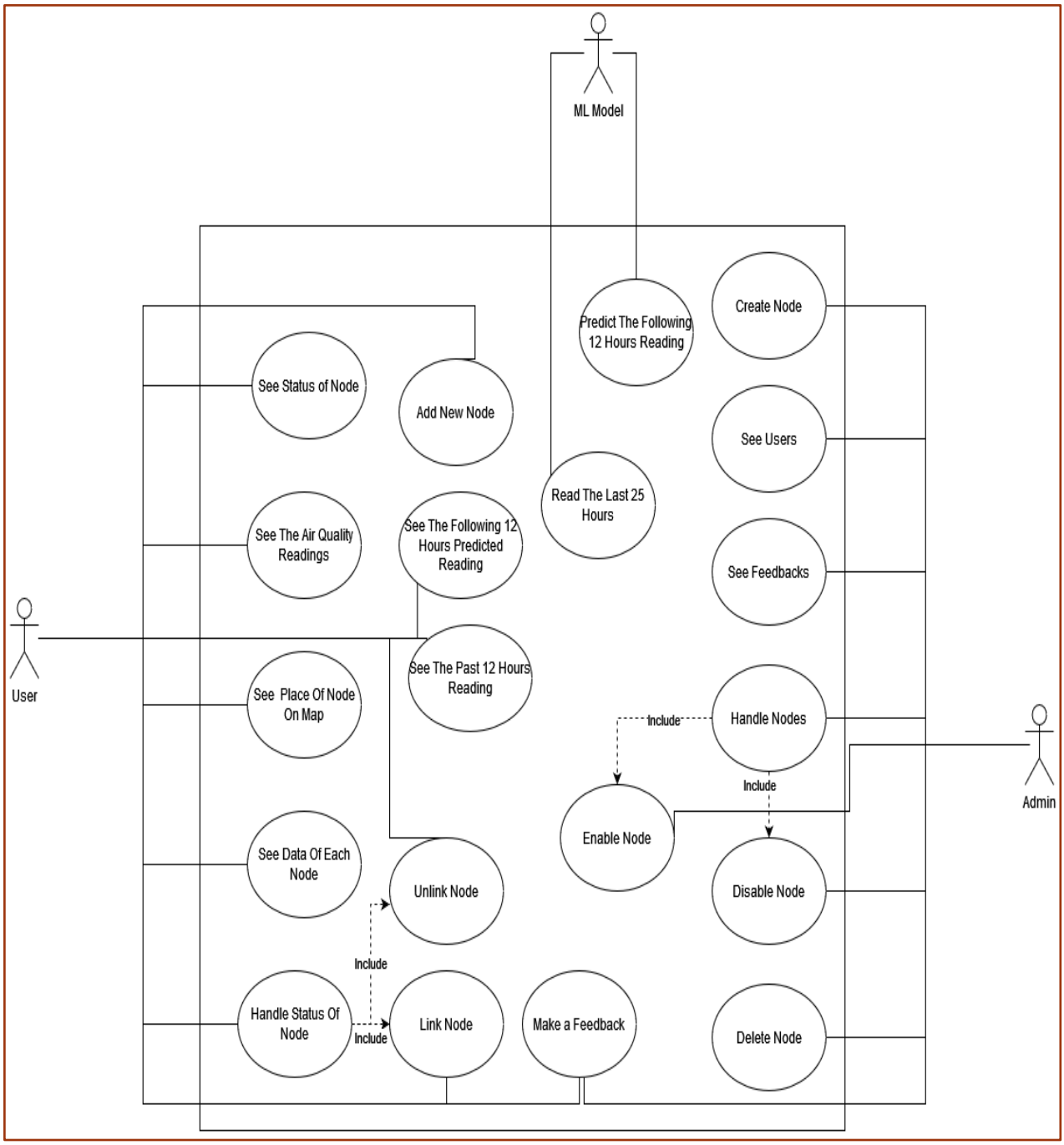


FIGURE 17: USE CASE DIAGRAM

4.4.3 Sequence Diagram:

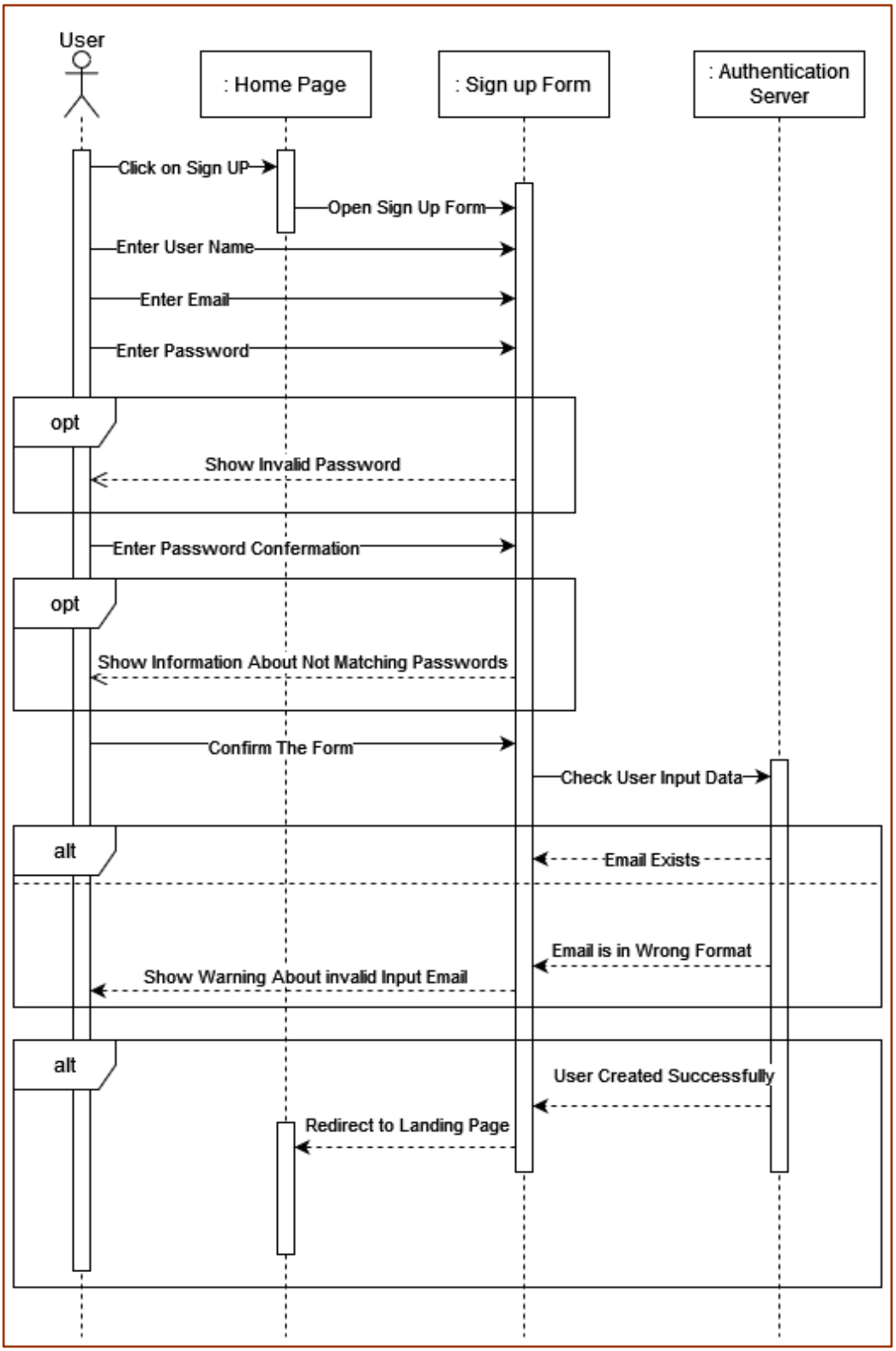


FIGURE 18: SIGN UP SEQUENCE DIAGRAM

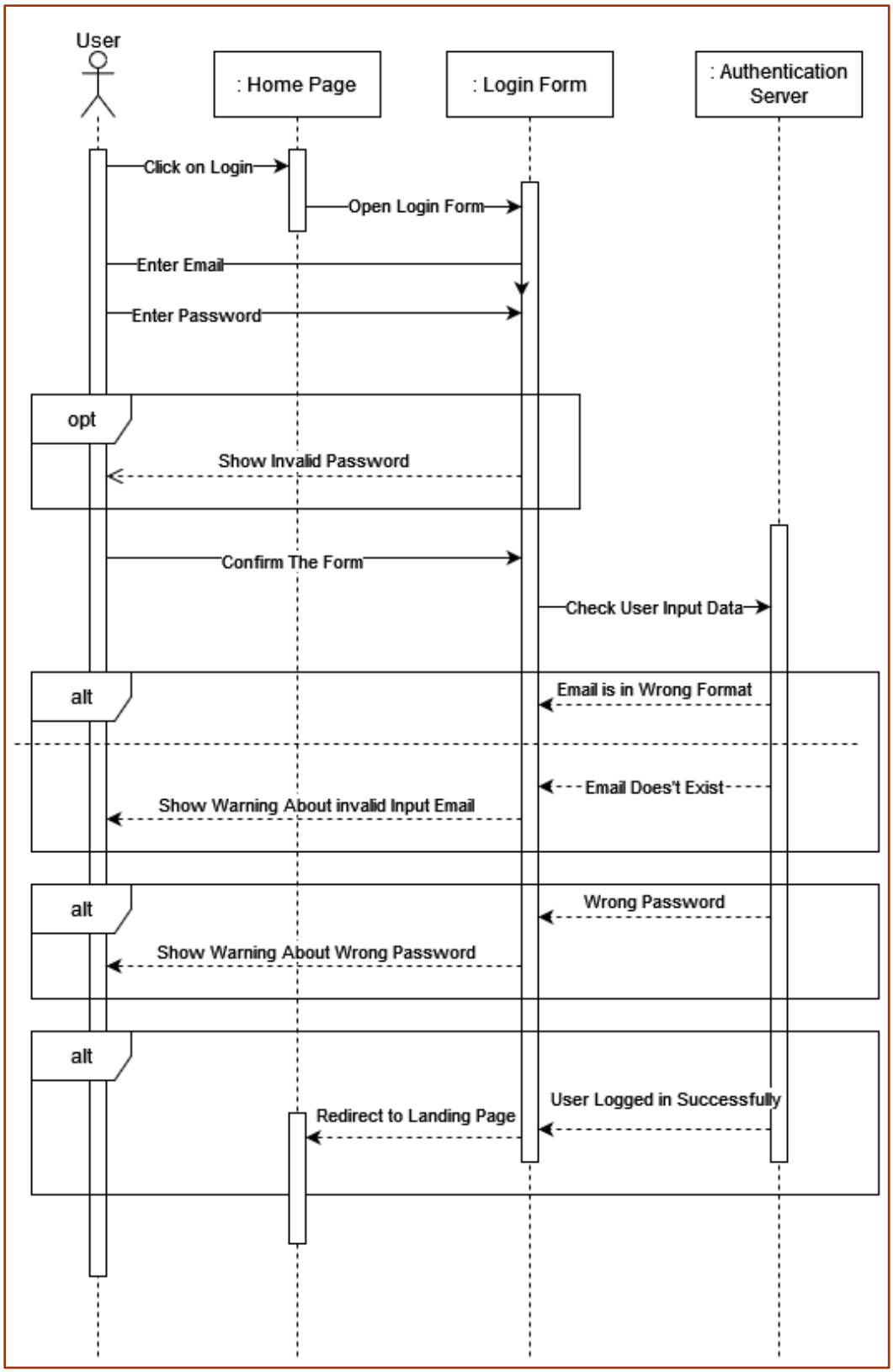


FIGURE 19: LOG IN SEQUENCE DIAGRAM

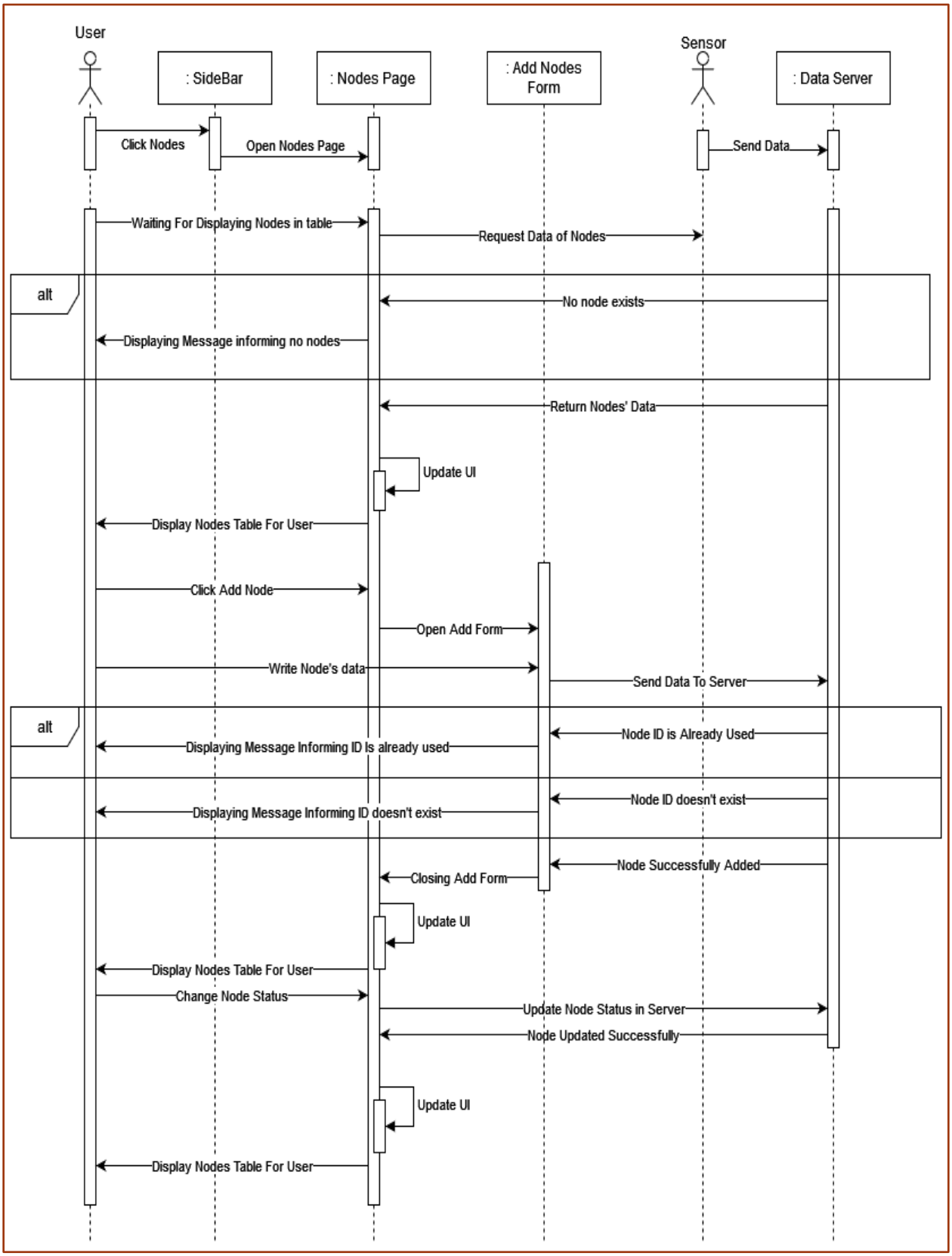


FIGURE 20: NODES SEQUENCE DIAGRAM

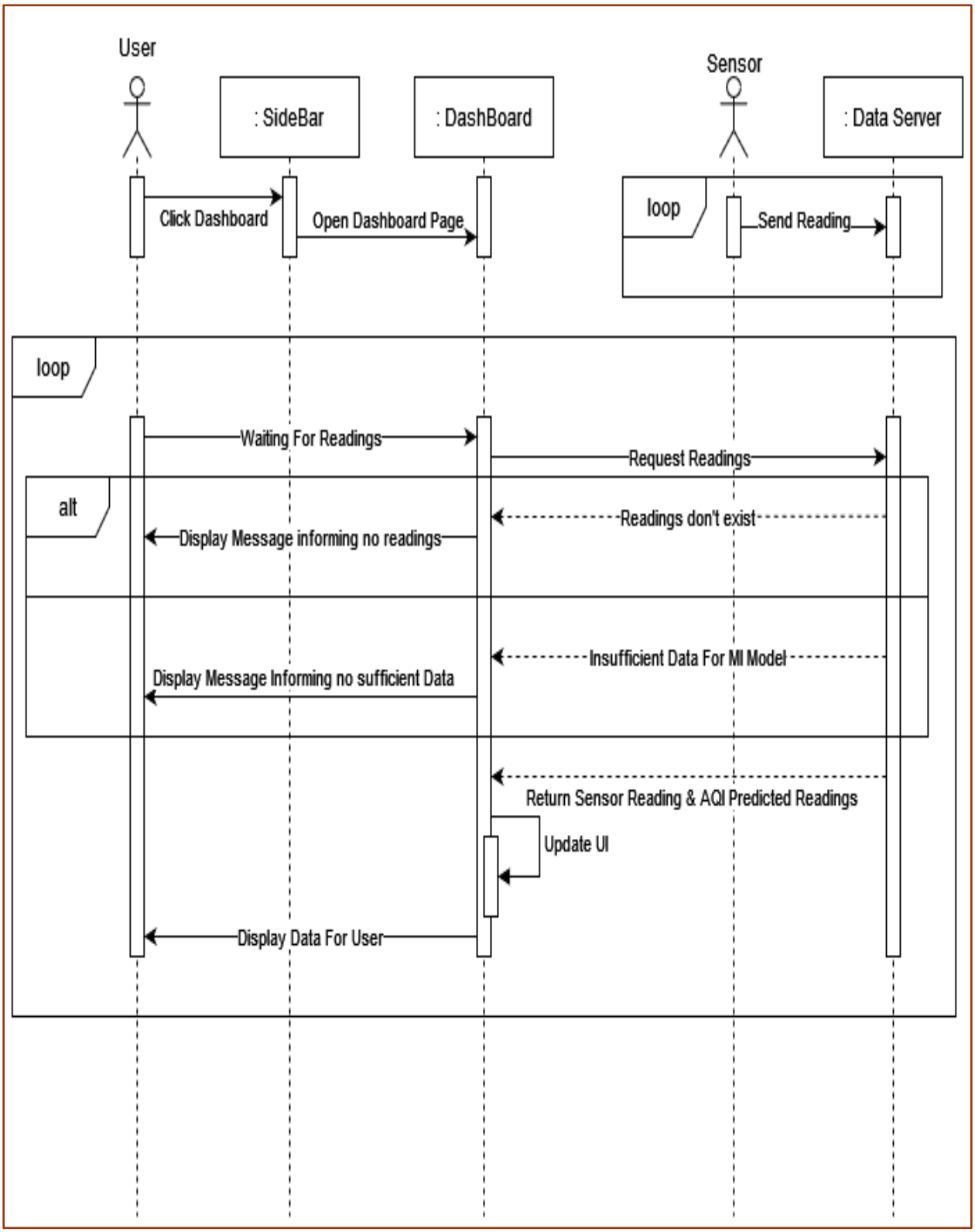


FIGURE 21: DASHBOARD SEQUENCE DIAGRAM

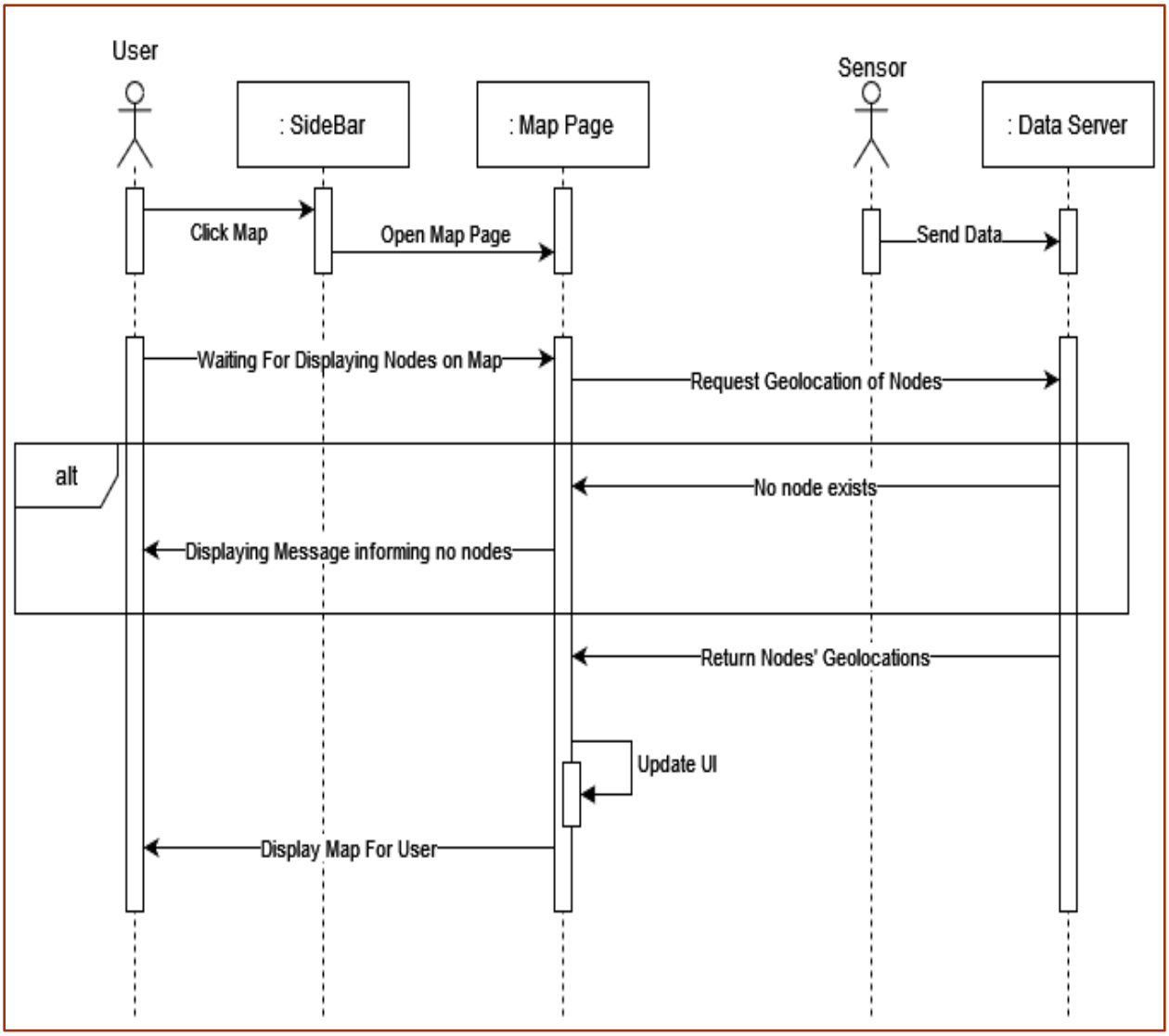


FIGURE 22: MAP SEQUENCE DIAGRAM

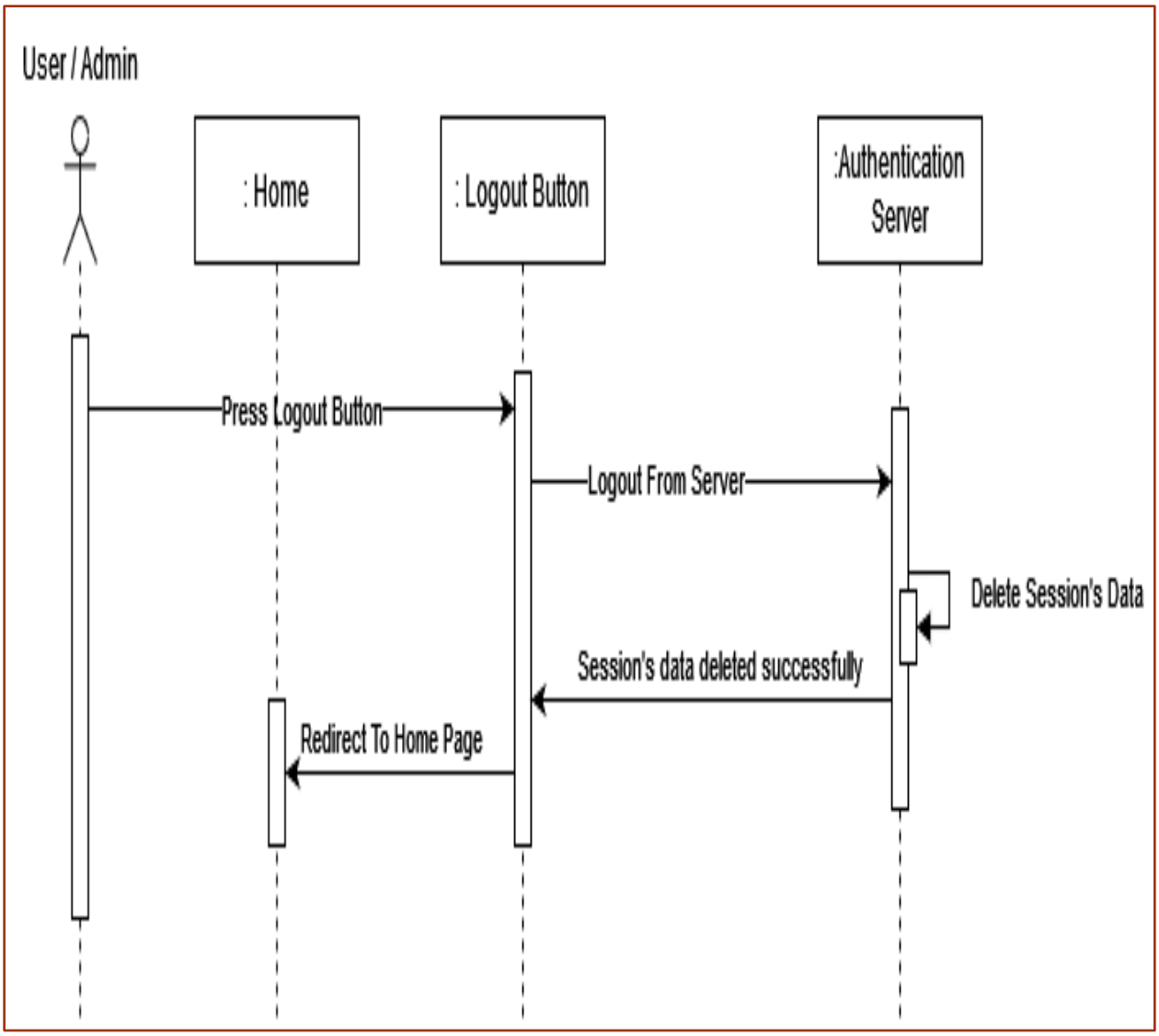


FIGURE 23: LOGOUT SEQUENCE DIAGRAM

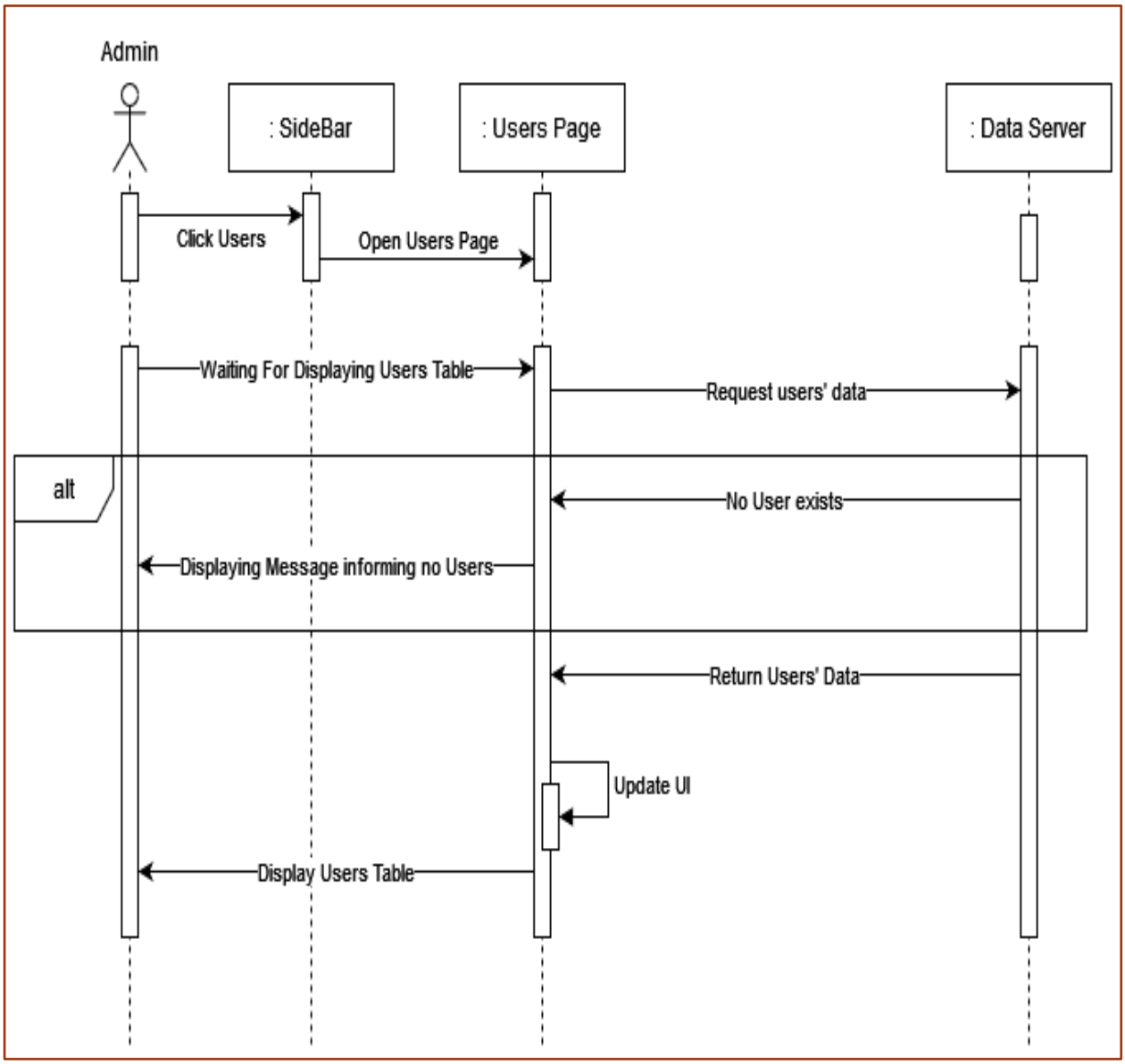


FIGURE 24: USERS PAGE SEQUENCE DIAGRAM

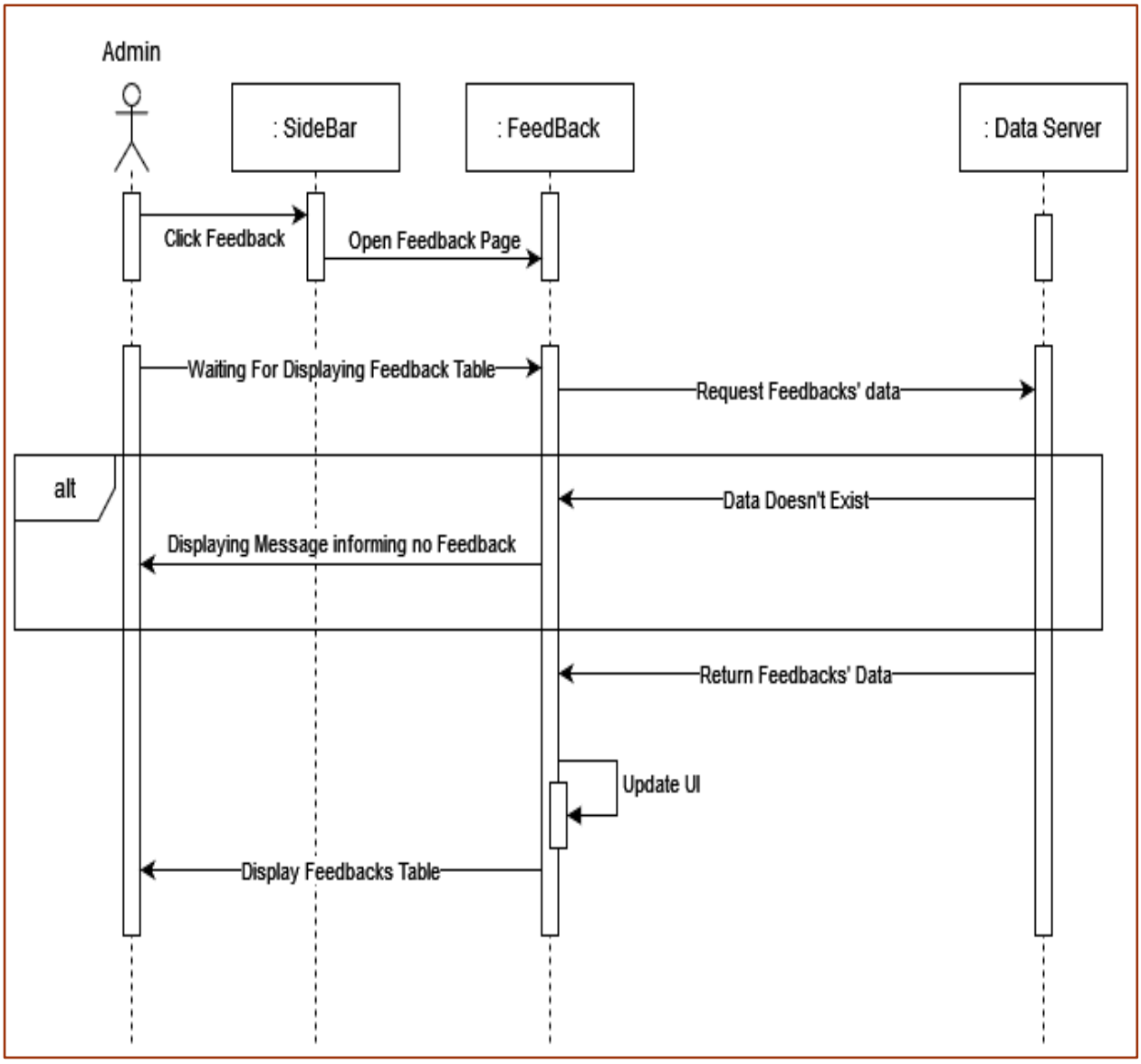


FIGURE 25: FEEDBACK SEQUENCE DIAGRAM

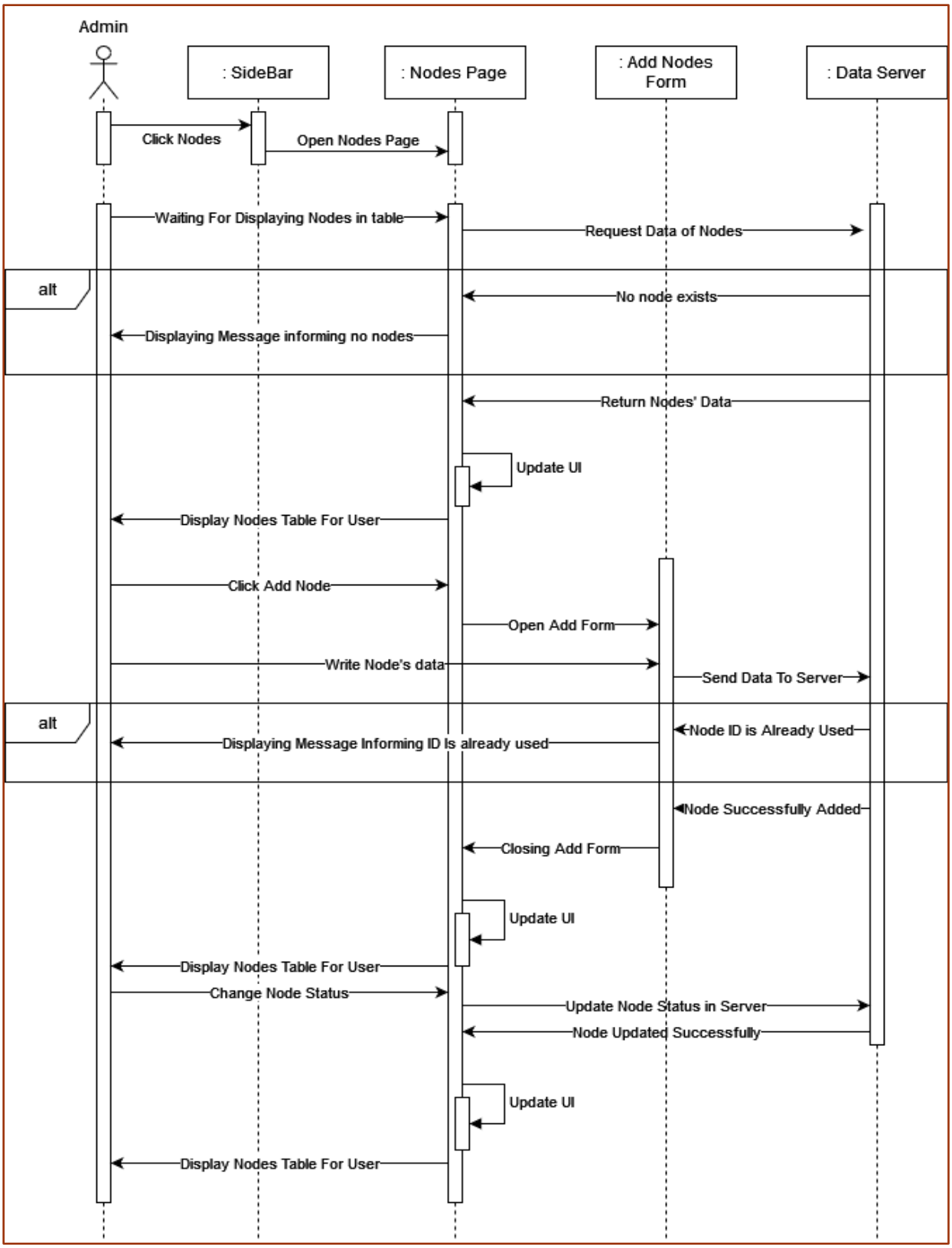


FIGURE 26: ADMIN NODES PAGE SEQUENCE DIAGRAM

4.5 Web Application

4.5.1 Home Page:

FIGURE 27: HOME PAGE

4.5.2 Sign up Modal:

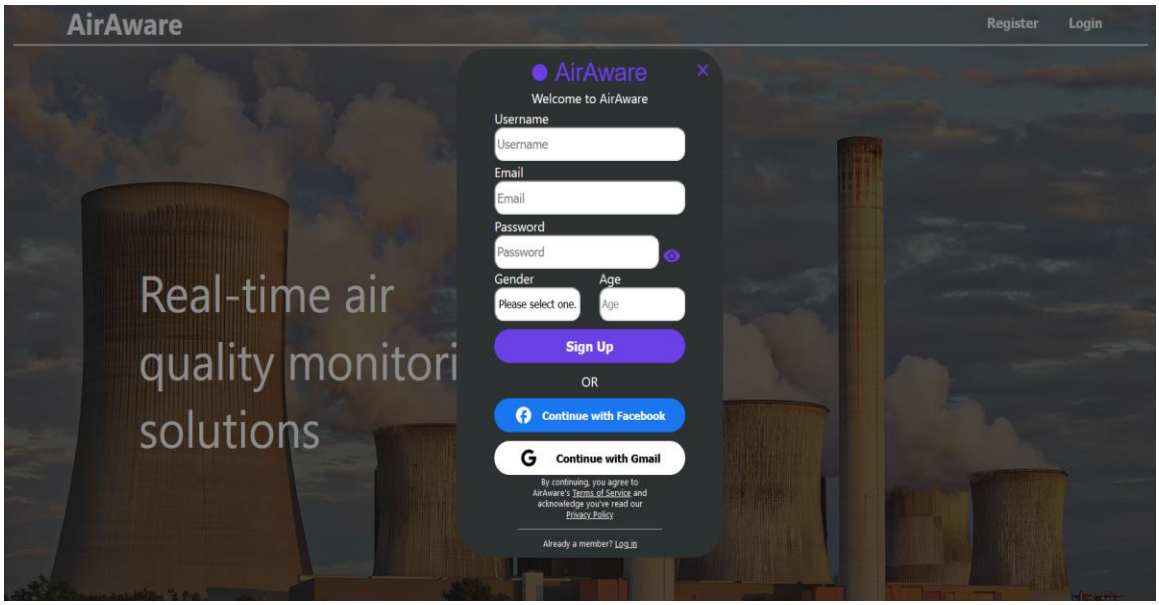


FIGURE 28: SIGN UP MODAL

4.5.3 Login Modal:

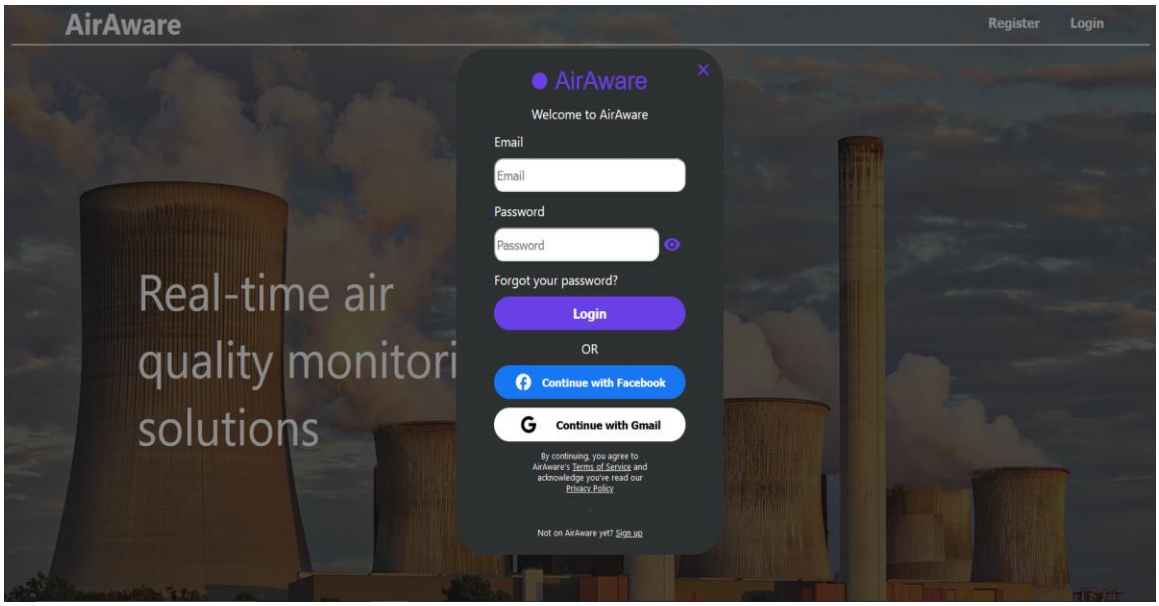


FIGURE 29: LOGIN MODAL

4.5.4 Dashboard page:

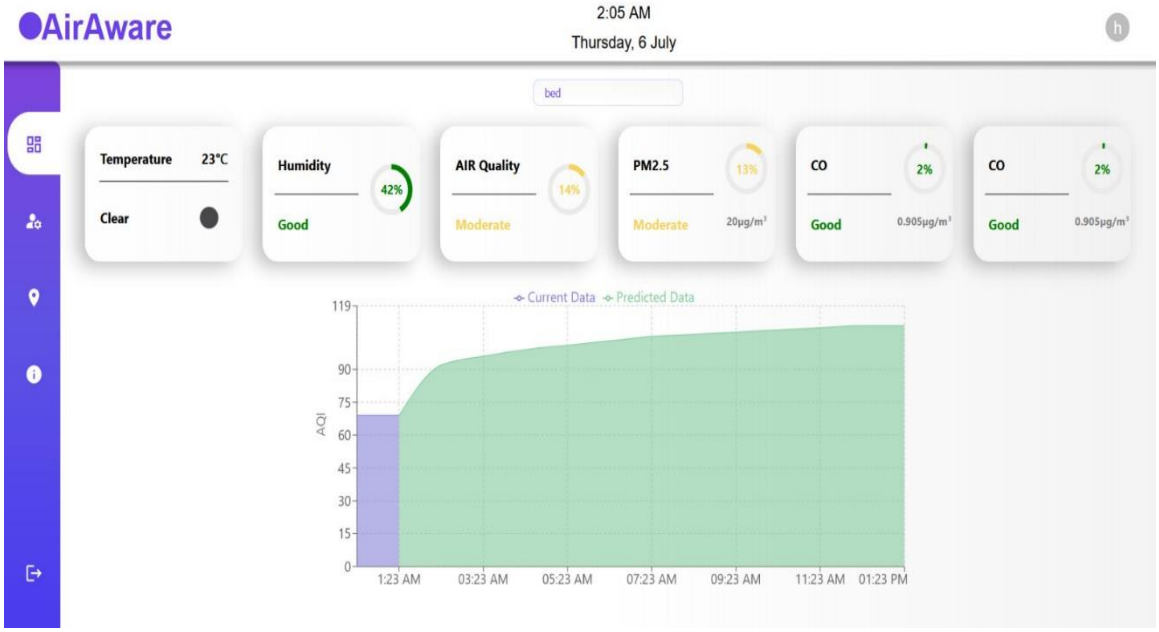


FIGURE 30: DASHBOARD PAGE

4.5.5 Manage page:

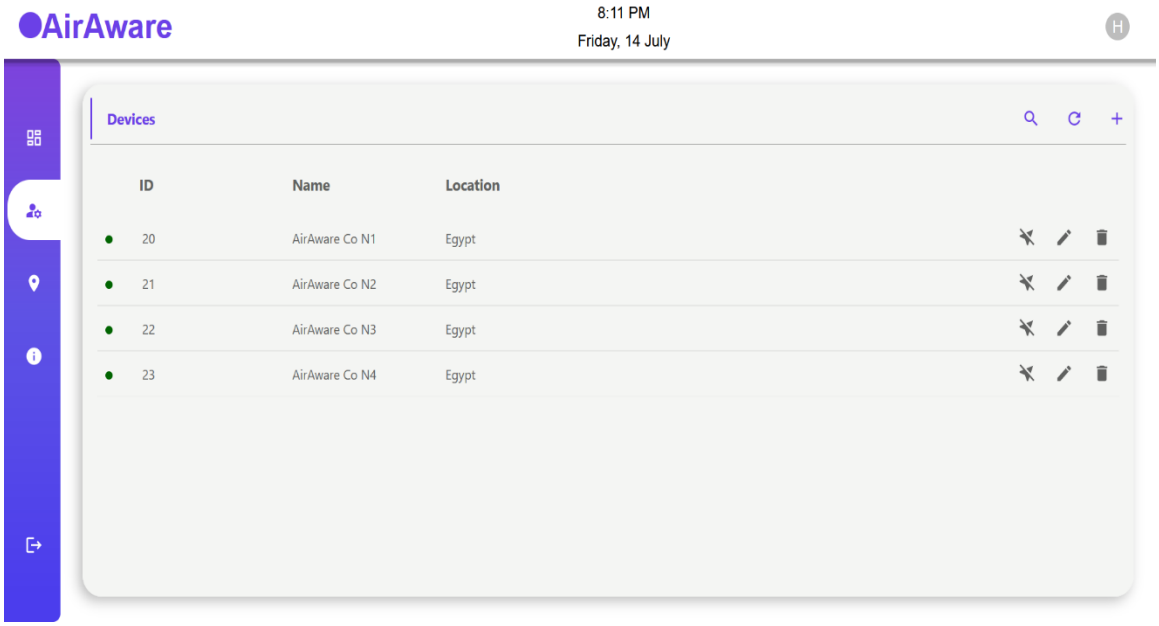


FIGURE 31: MANAGE PAGE

4.5.6 Map page:

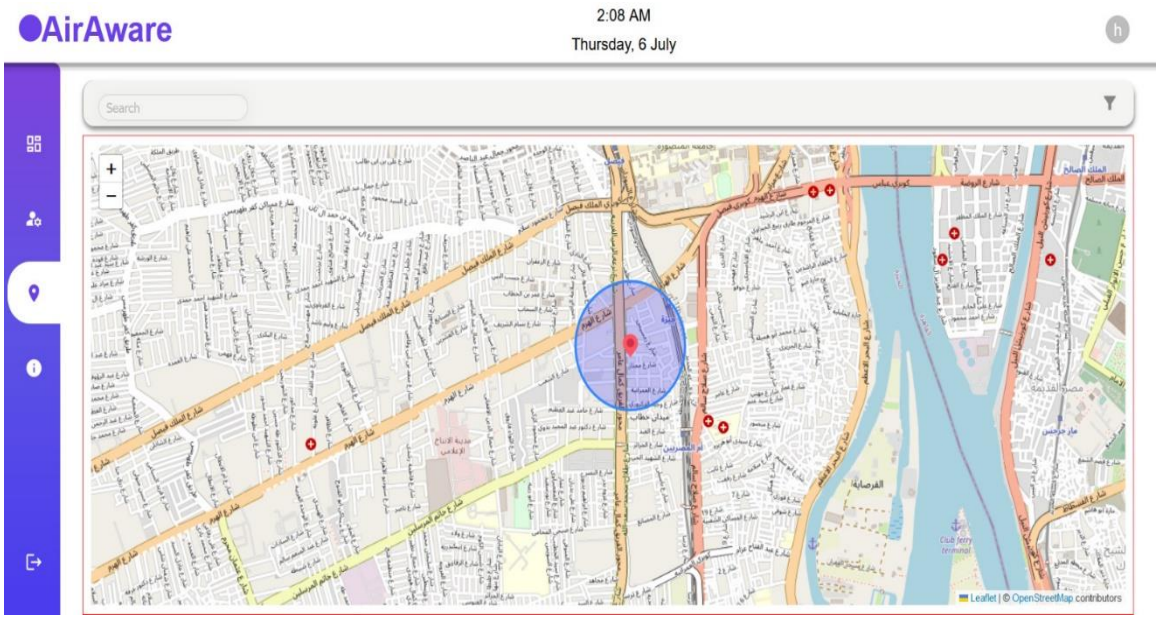


FIGURE 32: MAP PAGE

4.5.7 Info Pages:

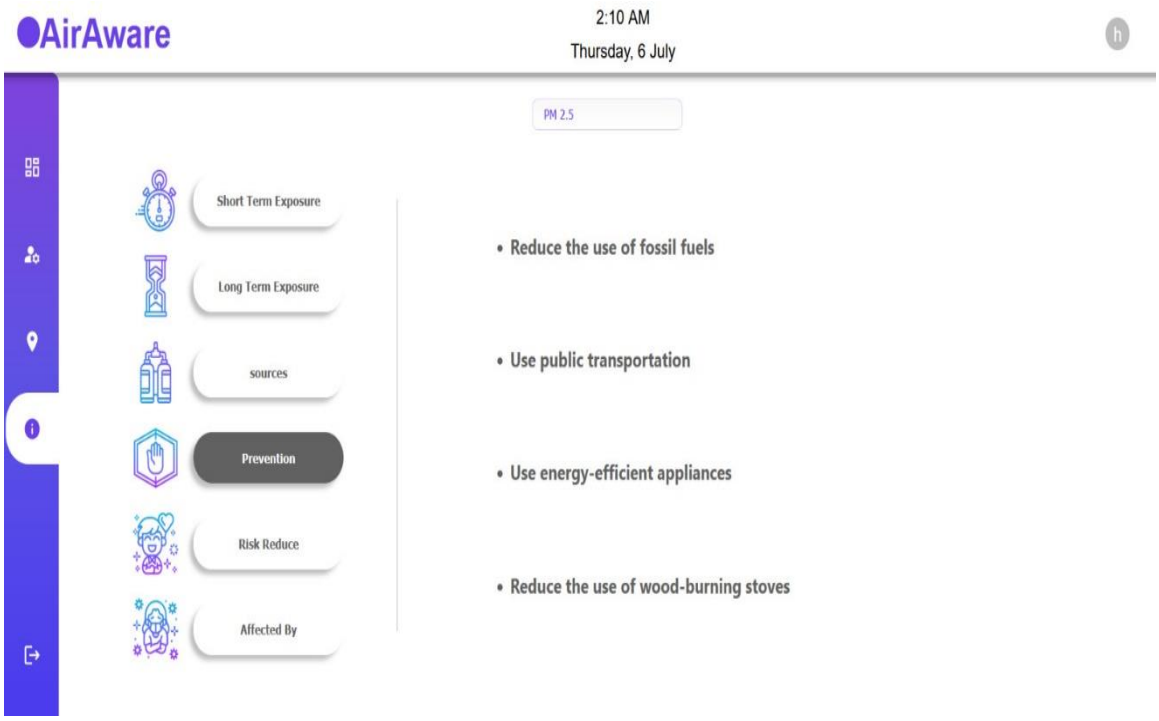


FIGURE 33: INFO PAGE SPECIFYING A GAS POLLUTANT

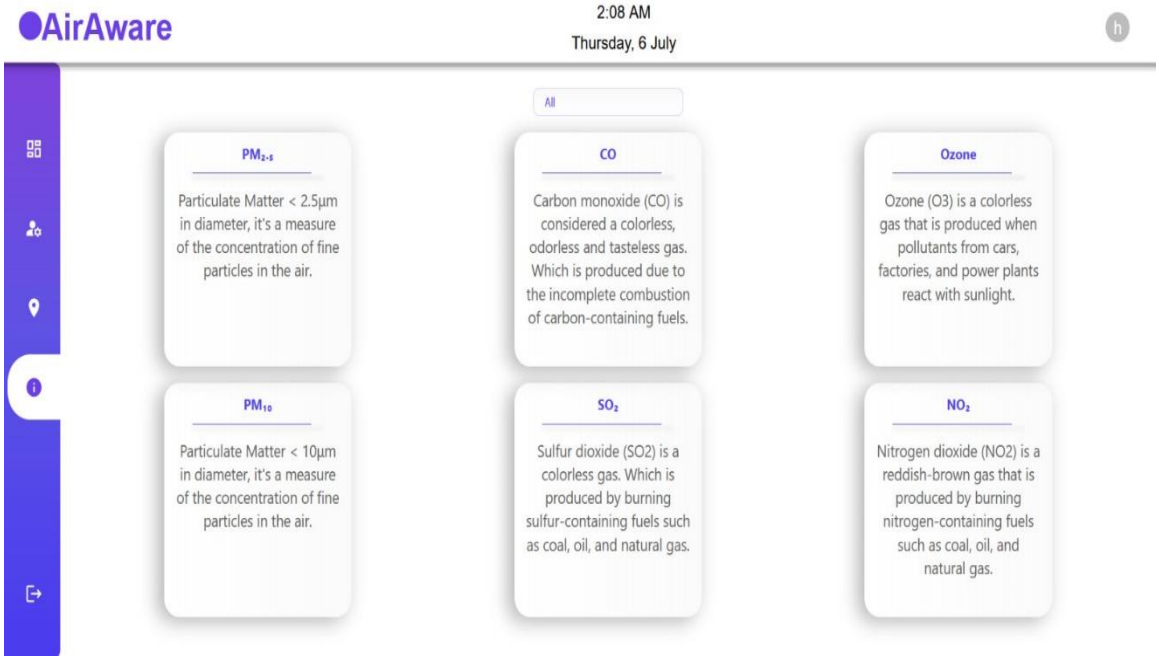


FIGURE 34: INFO PAGE

5. Machine Learning

5.1 Introduction:

Machine learning algorithms are powerful tools that contribute to addressing environmental challenges and reducing pollution. From environmental monitoring and data analysis to pollution prediction and prevention, energy optimization, waste management, and decision support systems, algorithms play a vital role in creating a more sustainable and environmentally friendly future. By harnessing the capabilities of machine learning, we can make informed decisions, implement effective strategies, and foster a cleaner and healthier environment for current and future generations.

Forecasting time series data is an important subject in economics, business, and finance. Traditionally, there are several techniques to effectively forecast the next lag of time series data such as univariate **Autoregressive (AR),** and more notably **Autoregressive Integrated Moving Average (ARIMA),** with its many variations. In particular, the ARIMA model has demonstrated its outperformance in precision and accuracy of predicting the next lags of time series. With the recent advancement in computational power of computers and more importantly developing more advanced machine learning algorithms and approaches such as deep learning, new algorithms are developed to forecast time series data. such as **"Long Short-Term Memory (LSTM)"**

5.2 Time Series Forecasting:

Time series analysis aims to study the path observations of time series and build a model to describe the structure of data and predict the future values of time series. Due to the importance of time series forecasting in many branches of applied sciences, it is essential to build an effective model with the aim of improving the forecasting accuracy. So we have employed multiple algorithms to compare their performance in terms of errors. However, it is worth noting that all of these algorithms have undergone the same data processing steps.

5.3 Data:

We've collected our data set from https://www.clarity.io/ from a wireless sensor node fixed on a highway in Alexandria. The node is designed to measure PM2.5 and output the AQI based on this and it outputs a reading every single hour, *figure 35*.

	A	B	C	D
1	Date	Time	PM2.5	AQI
2	12/1/2022	11:00:00 PM	164	214
3	12/1/2022	10:00:00 PM	167	217
4	12/1/2022	9:00:00 PM	173	223
5	12/1/2022	8:00:00 PM	151	201
6	12/1/2022	7:00:00 PM	115	182
7	12/1/2022	6:00:00 PM	83	165
8	12/1/2022	5:00:00 PM	45	124
9	12/1/2022	4:00:00 PM	51	139
10	12/1/2022	3:00:00 PM	59	153
11	12/1/2022	2:00:00 PM	78	163
12	12/1/2022	1:00:00 PM	93	170

FIGURE 35: PREPARING DATA FOR MACHINE MODELS

5.3.1 Data preprocessing:

We've cleaned the data, transferred it into the right date-time formula in order to accurately represent and analyze the data over time. The correct date-time formula will ensure that the data is properly ordered and that time intervals are consistent, which is essential for making accurate predictions and identifying trends.

datetime	PM2.5	AQI
2022-08-04 10:00:00	53	144
2022-08-04 11:00:00	60	153
2022-08-04 12:00:00	54	147
2022-08-04 13:00:00	43	119
2022-08-04 14:00:00	37	105
2022-08-04 15:00:00	40	112
2022-08-04 16:00:00	42	117
2022-08-04 17:00:00	43	119

FIGURE 36: DATA AFTER PREPARING

Gaps in time series data refer to periods of time for which no data is available. These gaps can occur for a variety of reasons, such as equipment failure, data transmission errors, or missing data due to holidays or weekends. Gaps in time series data can be problematic because they can affect the accuracy and reliability of statistical analyses and models. so with the help of matplotlib, we can notice them figure3.

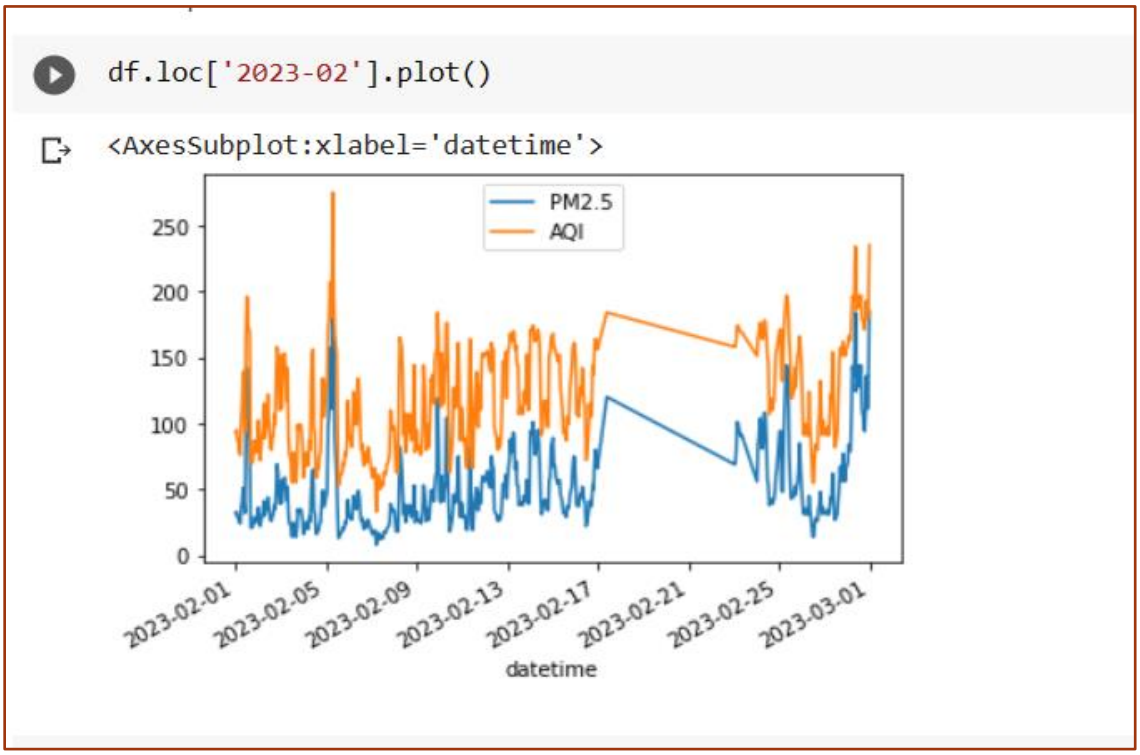


FIGURE 37: PLOTTING AQI AGAINST TIME

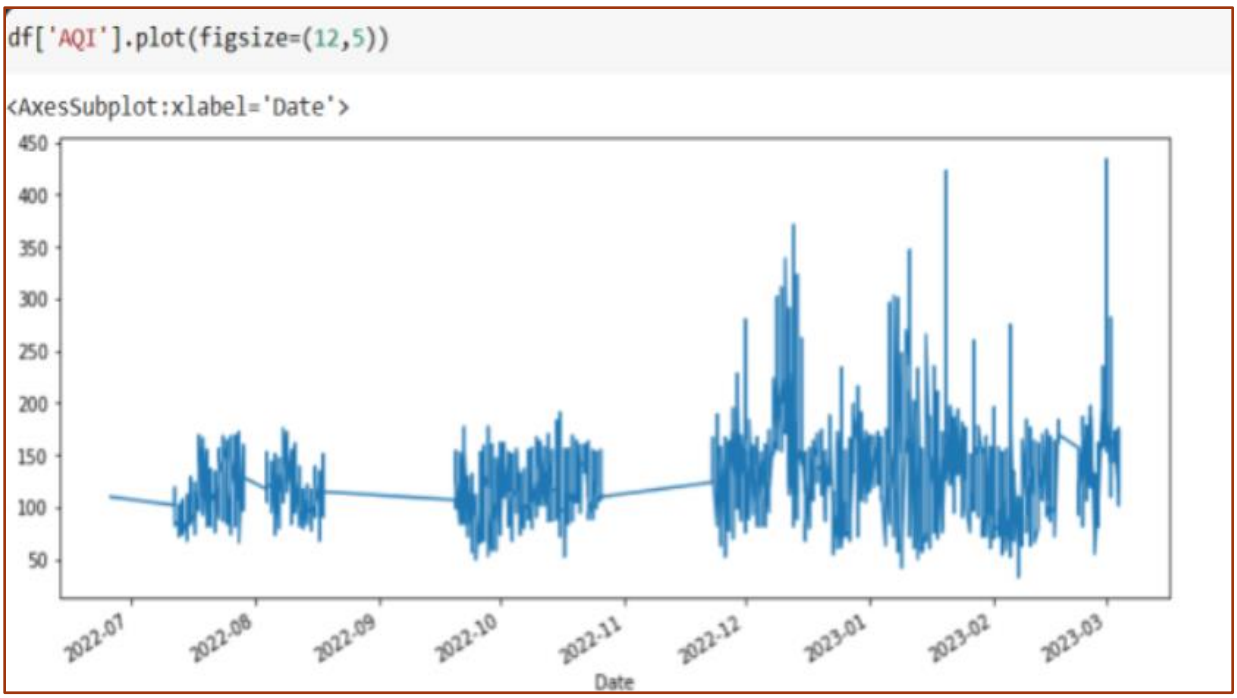


FIGURE 38: AQI PLOT

we can fill these gaps using one of these methods:

-Forward Fill

-Backward Fill

-Mean or Median Imputation

the choice of method for filling gaps depends on the specific characteristics of the data, the duration and extent of the gaps.

the choice of method for filling gaps depends on the specific characteristics of the data, the duration and extent of the gaps.

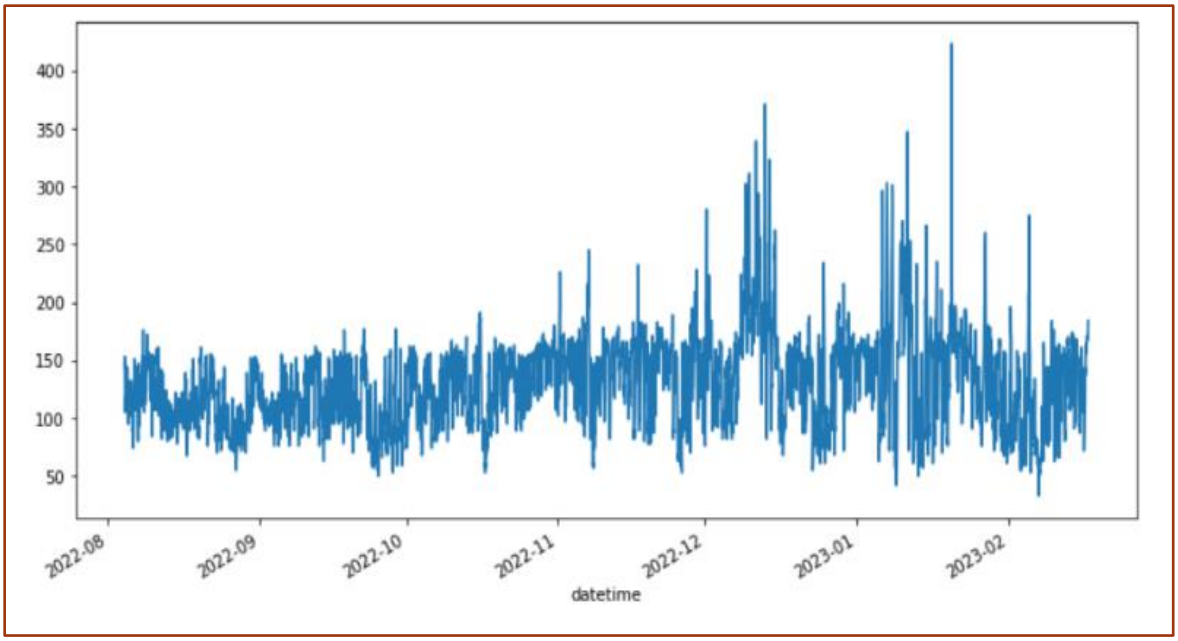


FIGURE 39: DATA WITH OUTLIERS

The third step is getting rid of outliers. We used the Moving average technique. Moving average is a commonly used method for smoothing time series data, and it can also be used to detect outliers. One limitation of moving average is that it is sensitive to the choice of window size. If the window size is too small, the moving average will be too sensitive to short-term fluctuations in the data, which can lead to false positives (i.e., identifying normal data points as outliers). On the other hand, if the window size is too large, the moving average may not capture short-term fluctuations in the data, which can cause it to miss outliers. In the following figure, we show the data after dealing with outliers.

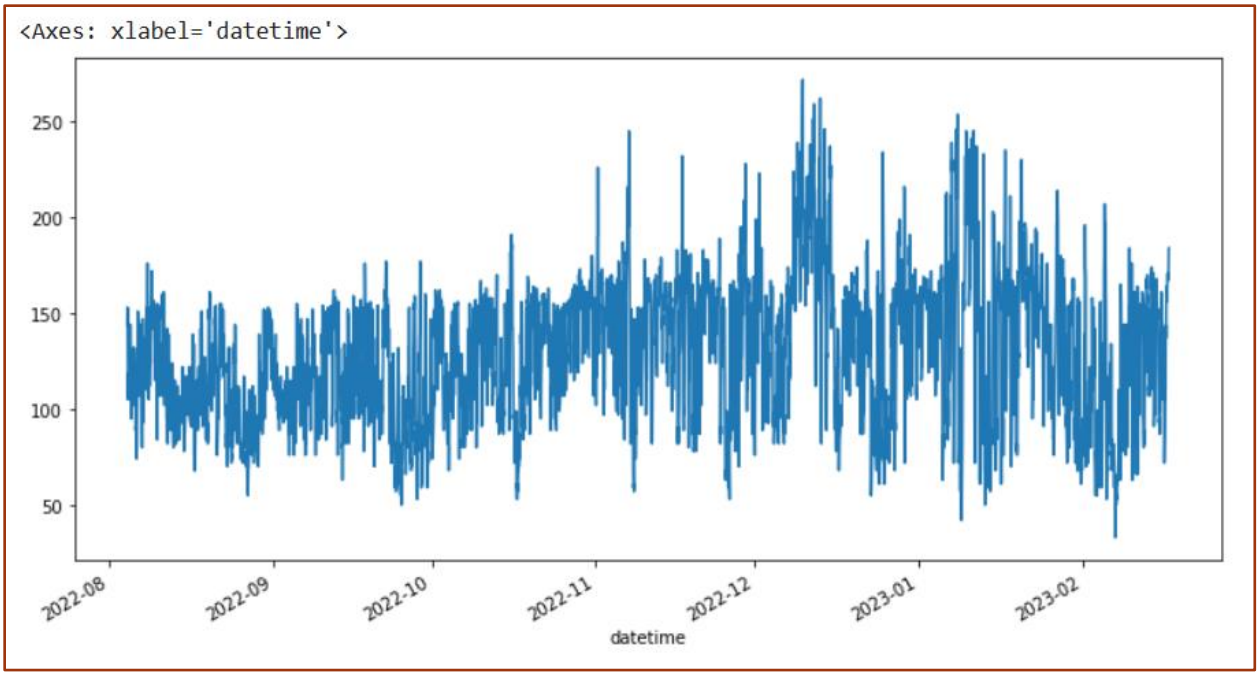


FIGURE 40: AFTER GETTING RID OF OUTLIERS

5.4 Training and Testing the Model

From here after putting the data into the right formula we can start splitting our data and building our model. We have tried three models in order to find the best one based on chosen performance metrics for our data. Then data is split into two parameters which contain air particles composition and time series in one parameter and AQI data in other parameters. We've considered PM 2.5 variables and time series for predicting the AQI variable.

We've tried many machine learning algorithms on our cleaned data to get the best accuracy. In the following section, we will provide you with a comparison of the result of each one and the best of them.

6. Models

6.1 Autoregressive Integrated Moving Average Model (ARIMA)

The first one is ARIMA (Autoregressive Integrated Moving Average) is a popular time series modelling technique used in statistics and econometrics. It is a generalization of the simpler ARMA (Autoregressive Moving Average) model and is capable of handling non-stationary time series data.

ARIMA models consist of three components:

Autoregression (AR): This component models the relationship between an observation and several lagged observations of the same variable.

Integrated (I): This component deals with the differencing of the time series to make it stationary.

Moving Average (MA): This component models the relationship between an observation and a residual error from a moving average model applied to lagged observations of the same variable.

Output graph:

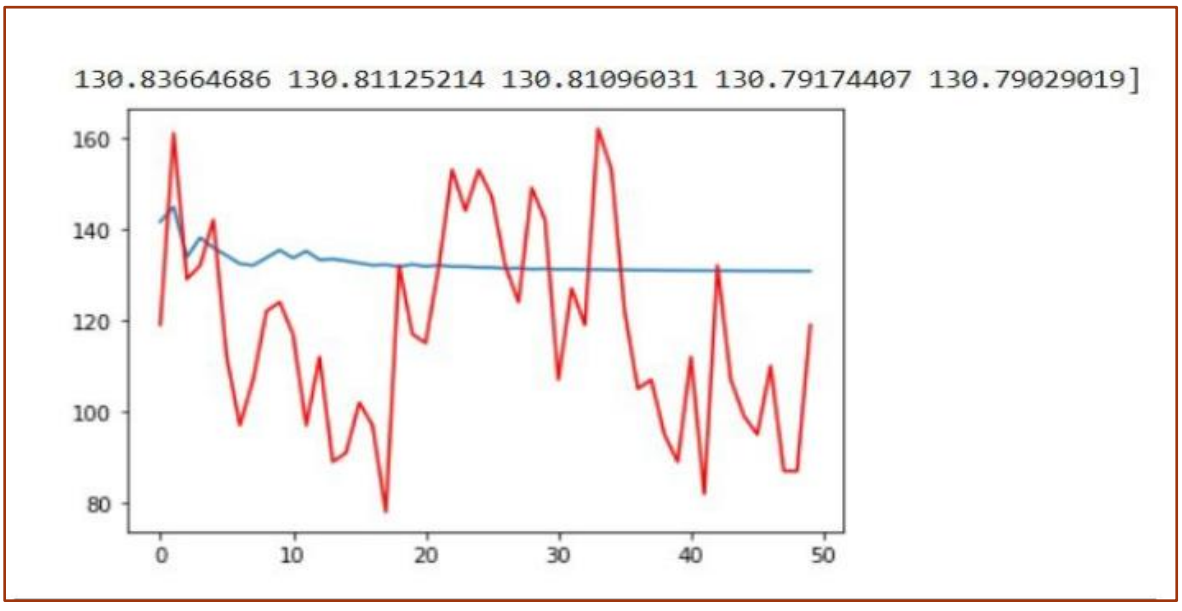


FIGURE 41: OUTPUT GRAPH OF ARIMA MODEL

6.2 Autoregression

The second one is **AR**, an AR (Autoregressive) model is a statistical model used in time series analysis to explain the behavior of a variable over time based on its past values.

In the AR model, the value of the variable at any given time is assumed to be a linear combination of its past values, with a random error term added to account for any unexplained variation. The order of the AR model, denoted by p, determines the number of past values that are used to predict the current value.

The equation for an AR(p) model is:

$$Y_t = c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + ... + \phi_p Y_{t-p} + \varepsilon_t$$

where Y_t is the value of the variable at time t, c is a constant, ϕ_1, ϕ_2, ..., ϕ_p are the coefficients for the past values of the variable, and ε_t is the random error term.

AR models are widely used in fields such as finance, economics, and engineering to analyze time series data and make predictions about future values of a variable.

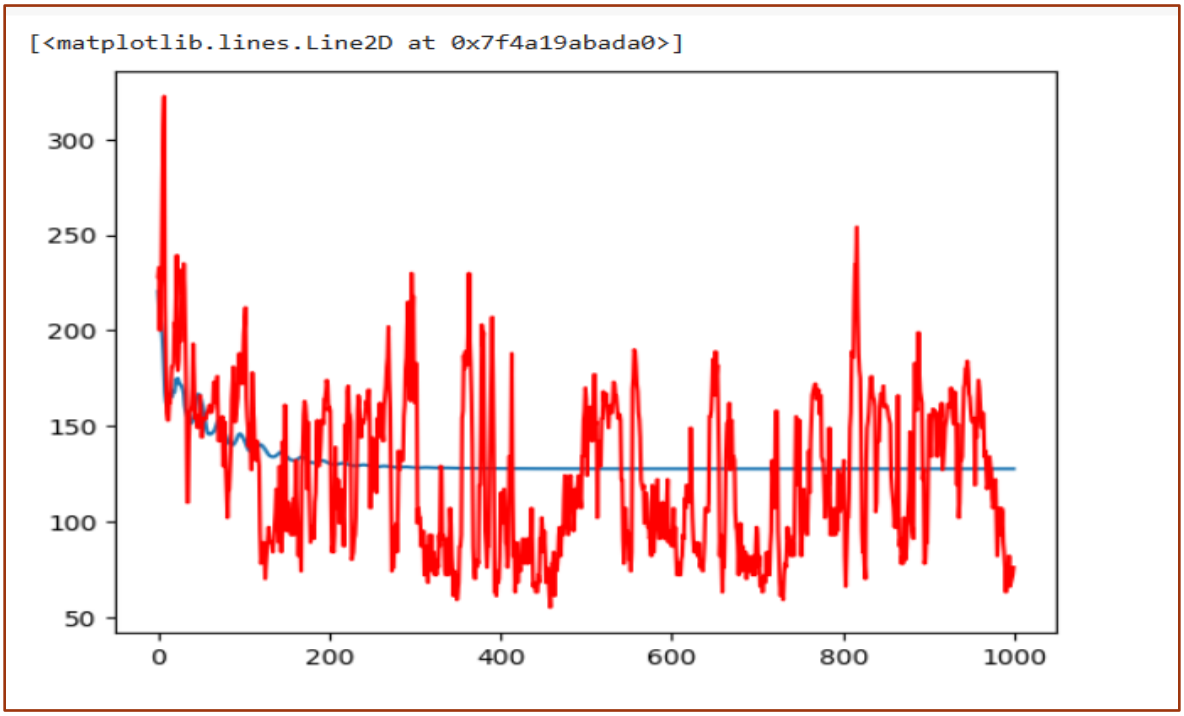


FIGURE 42: OUTPUT GRAPH OF AR MODEL

6.3 Long Short-Term Memory (LSTM)

LSTM is a special kind of RNNs with additional features to memorize the sequence of data. The memorization of the earlier trend of the data is possible through some gates along with a memory line incorporated in a typical LSTM. The internal structure of an LSTM cell is demonstrated in Diagram 1

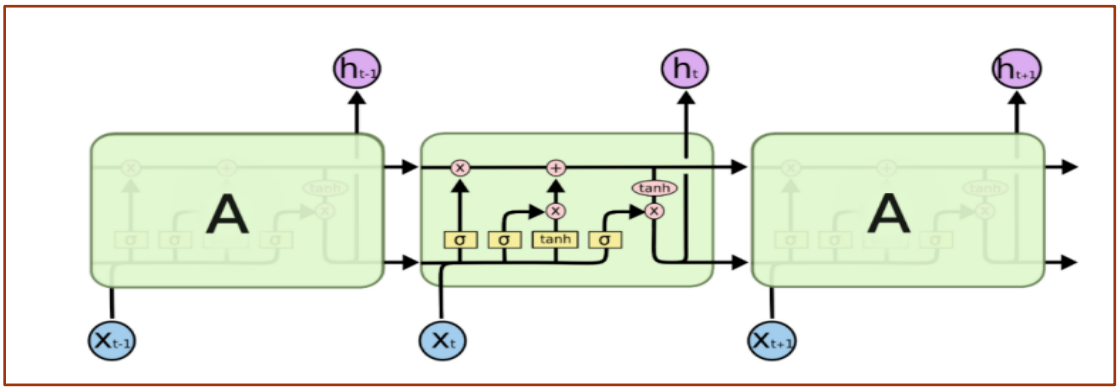


FIGURE 43: LSTM ARCHITECTURE

LSTM is a special kind of RNNs with additional features to memorize the sequence of data. Each LSTM is a set of cells, or system modules, where the data streams are captured and stored. The cells resemble a transport line (the upper line in each cell) that connects one module to another one conveying data from the past and gathering them for the present one. Due to the use of some gates in each cell, data in each cell can be disposed, filtered, or added for the next cells. Hence, the gates, which are based on sigmoidal neural network layer, enable the cells to optionally let data pass through or disposed. Each sigmoid layer yields numbers in the range of zero and one, depicting the amount of every segment of data that ought to be let through in each cell. More precisely, an estimation of zero value implies that "let nothing pass through"; whereas; an estimation of one indicates that "let everything pass through." Three types of gates are involved in each LSTM with the goal of controlling the state of each cell:

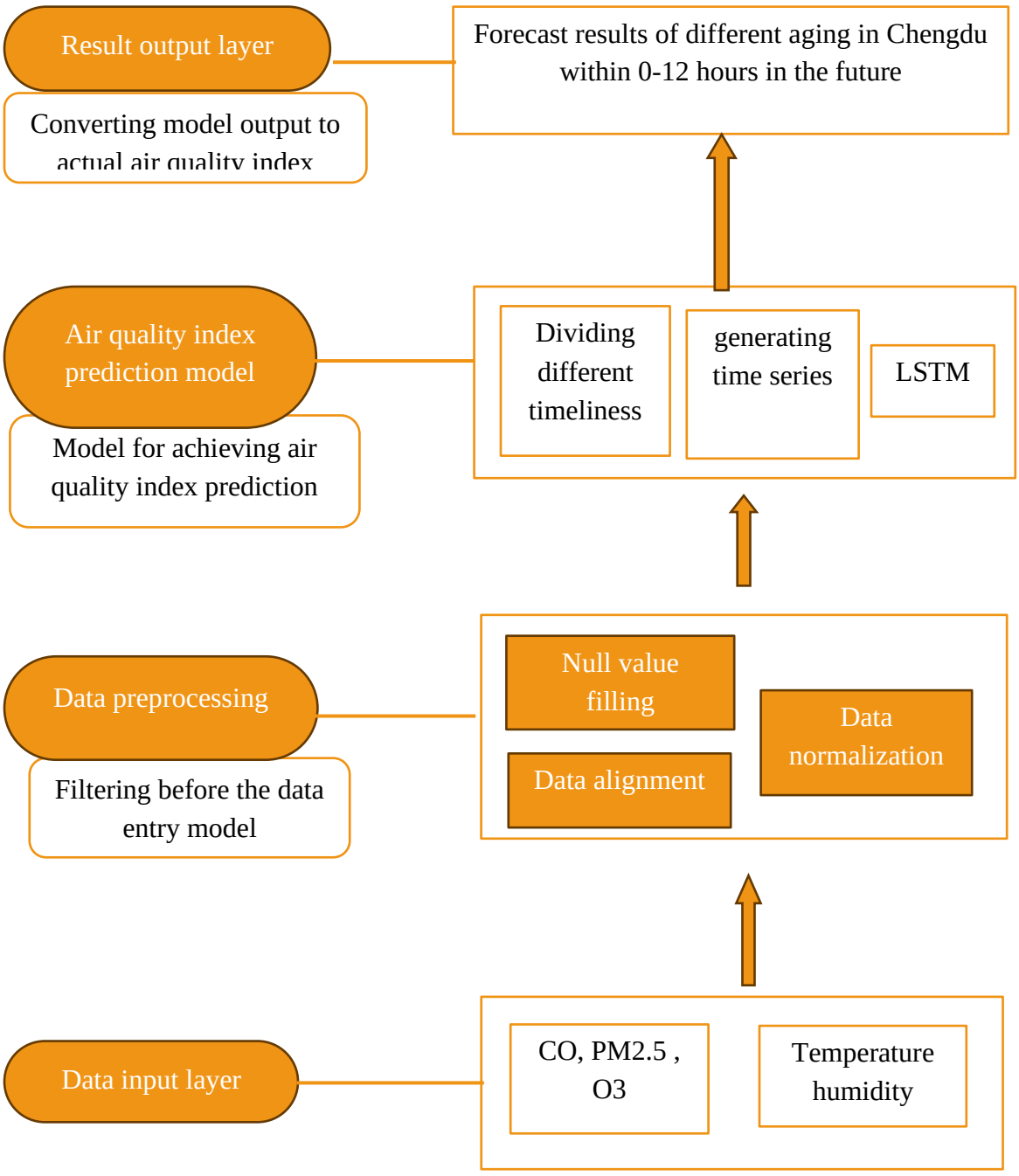
- **Forget Gate** outputs a number between 0 and 1, where 1 shows "completely keep this"; whereas, 0 implies "completely ignore this."

- **Memory Gate** chooses which new data need to be stored in the cell. First, a sigmoid layer, called the "input door layer" chooses which values will be modified. Next, a tanh layer makes a vector of new candidate values that could be added to the state.

- **Output Gate** decides what will be yielded out of each cell. The yielded value will be based on the cell state along with the filtered and newly added data.

6.4 Building of Model

LSTM-based AQI prediction model overall architecture



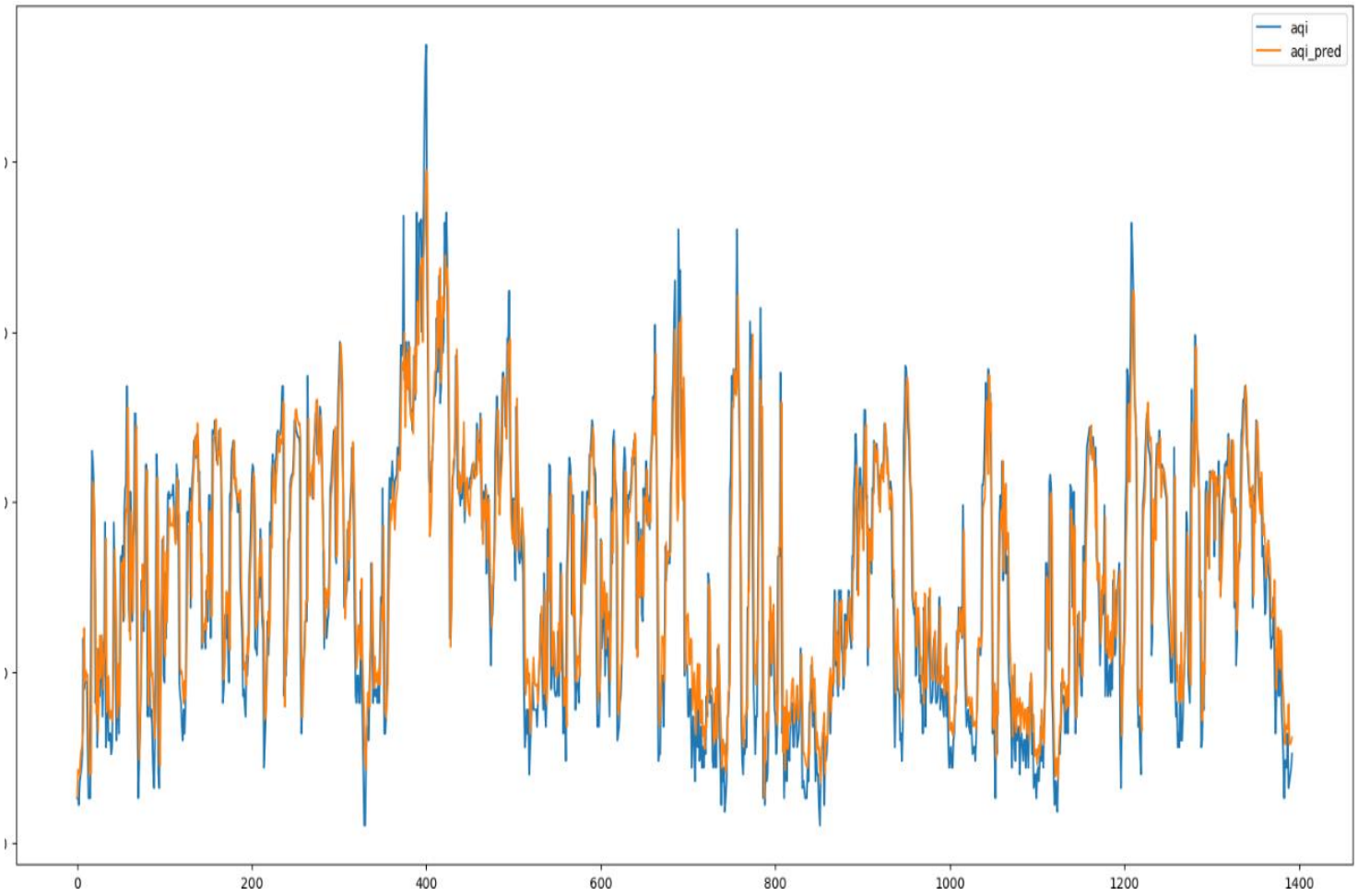


FIGURE 44: OUTPUT GRAPH FOR LSTM MODEL

```
df_final[['aqi','aqi_pred']].head(10)
```

	aqi	aqi_pred
0	63.0	63.362267
1	63.0	71.357780
2	61.0	71.283440
3	68.0	69.825912
4	70.0	75.588280
5	72.0	77.368340
6	76.0	79.342239
7	110.0	82.778008

FIGURE 45: OUTPUT OF LSTM MODEL

6.5 Results

Model	ARIMA	LSTM	AR
RMSE(root mean square error)	71.2029550438	18.068070645	44.498091163
MAE (Mean absolute error)		13.024864	32.75749066705875

7. Future Work

7.1 Upgrading Microcontroller:

To use more efficient and advanced microcontrollers to enhance the device's performance, power efficiency, and processing capabilities.

7.2 Enhanced Gas Sensors:

incorporate advanced gas sensors capable of detecting a wider range of pollutants with higher accuracy and sensitivity.

7.3 Security Measures:

Strengthen the security of your device's nodes to protect against unauthorized access and data compromise. Implement encryption protocols, secure data transmission channels, and authentication mechanisms to ensure the integrity and privacy of the collected data.

7.4 Data Validation and Calibration:

Develop robust data validation algorithms to identify and discard erroneous or misleading data. Regularly calibrate the sensors and validate the accuracy of measurements to ensure reliable and trustworthy air quality data.

7.5 Connectivity and Data Sharing:

Improve connectivity options by incorporating various wireless protocols (such as Wi-Fi, Bluetooth, or cellular) to enable seamless data transfer and remote monitoring.

7.6 User-Friendly Interface:

Enhance the device's user interface and make it more intuitive and user-friendly.

7.7 Power Optimization:

Focus on optimizing power consumption to extend the device's battery life or reduce energy requirements when connected to a power source. Efficient power management techniques can ensure long-term and reliable operation.

8. Testing:

8.1. Unit Testing:

A Type of testing used to test separate functions and components black box testing included equivalence partitioning and determining set boundaries.

8.2. Test Cases:

- **Unit Testing of Deleting Node:**

Test Case Number	Action	Expected output
TC_#1	Deleting Linked Node	Deleted successfully
TC_#2	Deleting Unlinked Node	Error message

- **Unit Testing of Linking Node:**

Test Case Number	Action	Expected output
TC_#1	Linking Unlinked Node	Linked successfully
TC_#2	Linking Linked Node	Error message

- **Unit Testing of Unlinking Node:**

Test Case Number	Action	Expected output
TC_#1	Unlinking Linked Node	Unlinked successfully
TC_#2	Unlinking Unlinked Node	Error message

- **Unit Testing of Activate Node:**

Test Case Number	Action	Expected output
TC_#1	Activate Node	Activated successfully

- **Unit Testing of Deactivate Node:**

Test Case Number	Action	Expected output
TC_#1	Deactivate Node	Deactivated successfully

- **Unit Testing of Deleting Feedback:**

Test Case Number	Action	Expected output
TC_#1	Delete Feedback	Deleted successfully

- **Unit Testing of Deleting User:**

Test Case Number	Action	Expected output
TC_#1	Delete User	Deleted successfully

- **Unit Testing of Creating new node:**

Equivalence Classes:

EC for ID:

EC1: empty -> invalid (0 Character)
EC2: positive numbers -> valid
EC3: alphanumeric -> invalid
EC4: alphabetic -> invalid
EC5: Symbols -> invalid
EC6: negative numbers -> invalid

Test Case Number	Input data	Expected output
TC_#1	Valid ID	Created successfully
TC_#2	Invalid ID	Error message

- **Unit Testing of adding Feedback:**

  Equivalence Classes:

  EC for Subject:

  EC1: empty -> invalid (0 Character)
  EC2: numbers -> valid
  EC3: alphanumeric -> valid
  EC4: alphabetic -> valid

  EC for Details:

  EC1: empty -> invalid (0 Character)
  EC2: numbers -> valid
  EC3: alphanumeric -> valid
  EC4: alphabetic -> valid

Test Case Number	Input data	Expected output
TC_#1	Valid Subject Valid Details	Added successfully
TC_#2	Valid Subject Invalid Details	Error message
TC_#3	Invalid Subject Valid Details	Error message
TC_#4	Invalid Subject Invalid Details	Error message

- **Unit Testing of Login:**

  Equivalence Classes:

  EC for Email:

  EC1: empty -> Invalid (0 Character)
  EC2: numbers -> Invalid
  EC3: alphanumeric -> Invalid
  EC4: email structure -> valid (1 ->50 Character)
  EC5: email structure invalid length -> invalid (>50 Character)
  EC6: Symbols -> invalid

  EC for Password:

  EC1: empty -> invalid
  EC2: alphanumeric but not correct -> invalid.
  EC3: Correct Password -> valid.

Test Case Number	**Email**	**Password**	**Output**
TC_#1	Empty	Empty	Error
TC_#2		Incorrect Alphanumeric	Error
TC_#3		Correct Password	Error
TC_#4	Numbers	Empty	Error
TC_#5		Incorrect Alphanumeric	Error
TC_#6		Correct Password	Error
TC_#7	Alphanumeric	Empty	Error
TC_#8		Incorrect Alphanumeric	Error
TC_#9		Correct Password	Error
TC_#10	Email structure (Correct)	Empty	Error
TC_#11		Incorrect Alphanumeric	Error
TC_#12		Correct Password	**Login**
TC_#13	Email	Empty	Error
TC_#14		Incorrect Alphanumeric	Error

TC_#15	structure invalid length	Correct Password	Error
TC_#16	Symbols	Empty	Error
TC_#17		Incorrect Alphanumeric	Error
TC_#18		Correct Password	Error

- **Unit Testing of Sign up:**

  EC for Name:

  EC1: empty -> Invalid (0 Character)
  EC2: numbers -> Invalid
  EC3: alphanumeric -> Invalid
  EC4: Symbols -> Invalid
  EC5: Alphabetic -> Valid

  EC for Email:

  EC1: empty -> Invalid (0 Character)
  EC2: numbers -> Invalid
  EC3: alphanumeric -> Invalid
  EC4: email structure -> valid (1 ->50 Character)
  EC5: email structure invalid length -> invalid (>50 Character)
  EC6: Symbols -> invalid

  EC for Password:

  EC1: empty -> invalid
  EC2: alphanumeric but not correct -> invalid.
  EC3: Correct Password -> valid.

  Test Cases Considering Name is Valid:

Test Case Number	**Email**	**Password**	**Output**
TC_#1	Empty	Empty	Error
TC_#2		Incorrect Alphanumeric	Error
TC_#3		Correct Password	Error
TC_#4	Numbers	Empty	Error
TC_#5		Incorrect Alphanumeric	Error
TC_#6		Correct Password	Error
TC_#7	Alphanumeric	Empty	Error
TC_#8		Incorrect Alphanumeric	Error
TC_#9		Correct Password	Error
TC_#10	Email structure (Correct)	Empty	Error
TC_#11		Incorrect Alphanumeric	Error
TC_#12		Correct Password	sign up
TC_#13	Email structure invalid length	Empty	Error
TC_#14		Incorrect Alphanumeric	Error
TC_#15		Correct Password	Error
TC_#16	Symbols	Empty	Error
TC_#17		Incorrect Alphanumeric	Error
TC_#18		Correct Password	Error

9. Conclusion

In conclusion, the development of the sensor package, known as "Node," for air pollution monitoring and prediction represents a significant step towards creating a more environmentally friendly and sustainable future. By harnessing the power of technology, we have successfully designed a system that not only measures air quality but also communicates this crucial information to a dedicated website.

Throughout this project, we focused on addressing the pressing issue of air pollution, recognizing its detrimental effects on human health and the environment. The Node sensors offer a practical and efficient solution for monitoring air quality in real-time, providing valuable data for individuals, communities, and policymakers to make informed decisions.

One of the notable achievements of this project is the ability to predict the Air Quality Index (AQI) for the next 12 hours. By analyzing the collected sensor data and leveraging advanced algorithms, we can anticipate changes in air pollution levels and provide early warnings to help mitigate the potential risks. This predictive capability empowers individuals and authorities to take proactive measures and implement timely interventions, thereby protecting public health and minimizing the environmental impact of air pollution.

Moreover, the development of the Node sensors aligns with the growing demand for sustainable technologies that promote environmental stewardship. By utilizing these sensors, we contribute to a comprehensive air quality monitoring network that fosters awareness and accountability. This collective effort can drive positive change by raising public consciousness, encouraging responsible practices, and supporting evidence-based policymaking.

In conclusion, the Node sensor package serves as a testament to the power of innovation and collaboration in addressing critical environmental challenges. By combining cutting-edge sensor technology, data analysis, and predictive capabilities, we have created a practical and scalable solution for air quality monitoring. With the potential to improve public health, inform decision-making, and promote sustainable practices, the Node sensors are a valuable tool in safeguarding our environment and creating a healthier future for all.

10.References

[1] Ullo, S.L.; Sinha, G.R. Advances in Smart Environment Monitoring Systems Using IoT and Sensors. Sensors 2020, 20, 3113. https://doi.org/10.3390/s20113113

[2] Dimosthenis, Sarigiannis (2015). Chapter 46 Indoor Air Quality Indicators. In Environmental indicators (pp. 827–841). Dordrecht, New York; Springer.

[3] Paul, Scheepers (2015). Chapter 50 Health-Related Indicators of Outdoor Air Quality. In Environmental indicators (pp. 883–898). Dordrecht, New York; Springer.

[4] IQAir. (2021, January 11). What is AQI? IQAir Newsroom Article on Air Quality Index. [Online]. https://www.iqair.com/newsroom/what-is-aqi

[5] Li, X., Huang, S., Jiao, A., Yang, X., Yun, J., Wang, Y., ... & Zhang, Y. (2019). Exposure to ambient fine particulate matter and premature mortality due to all causes and cardiovascular disease in China. EClinicalMedicine, 11, 68-79.

[6] Zhang, J., Zhu, T., Kipen, H., Wang, G., Huang, W., Rich, D., ... & Wang, Y. (2017). Cardiorespiratory biomarker responses in healthy young adults to drastic air quality changes surrounding the 2016 Beijing Olympics. Atmospheric Environment, 161, 1-10.

[7] Ouyang, M., Cao, G., He, M., Wang, Y., Zhang, S., & Chen, R. (2021). How to reduce the risk of outdoor exposure to air pollution. Journal of thoracic disease, 13(2), 1196.

[8] United States Environmental Protection Agency. (2015, June 14). Ground-level Ozone Basics. [Online]. https://www.epa.gov/ground-level-ozone-pollution/ground-level-ozone-basics#effects

[9] California Air Resources Board. Carbon Monoxide & Health. ——. Carbon Monoxide and Health. [Online]. Available: https://ww2.arb.ca.gov/resources/carbon-monoxide-and-health

[10] United States Environmental Protection Agency. (2022, August 2). Basic Information about Carbon Monoxide (CO) Outdoor Air Pollution. [Online]. Available: https://www.epa.gov/co-pollution/basic-information-about-carbon-monoxide-co-outdoor-air-pollution

[11] United States Environmental Protection Agency. (2023, February 16). Sulfur Dioxide Basics. [Online]. Available: https://www.epa.gov/so2-pollution/sulfur-dioxide-basics#what%20is%20so2

[12] United States Environmental Protection Agency. (2022, July 18). Particulate Matter (PM) Basics. [Online]. https://www.epa.gov/so2-pollution/sulfur-dioxide-basics#what%20is%20so2

[13] United States Environmental Protection Agency. (2022, August 30). Health and Environmental Effects of Particulate Matter (PM). [Online]. https://www.epa.gov/pm-pollution/health-and-environmental-effects-particulate-matter-pm

[14] United States Environmental Protection Agency. (2022, August 2). Basic Information about NO2. [Online]. https://www.epa.gov/no2-pollution/basic-information-about-no2#What%20is%20NO2

[15] Wikipedia. (2023, March 21). Air Pollution. [Online]. https://en.wikipedia.org/wiki/Air_pollution

[16] [Online] https://www.airnow.gov/

[17] [Online] https://www.teachmemicro.com/use-mq-7-carbon-monoxide-sensor/

[18] [Online] https://www.stateofglobalair.org/air/pm

[19] [Online] https://simple-circuit.com/pic16f877a-dht22-sensor-proteus-simulation/

[20] [Online] https://github.com/sharpsensoruser/sharp-sensor-demos/wiki/Application-Guide-for-Sharp-GP2Y1014AU0F-Dust-Sensor