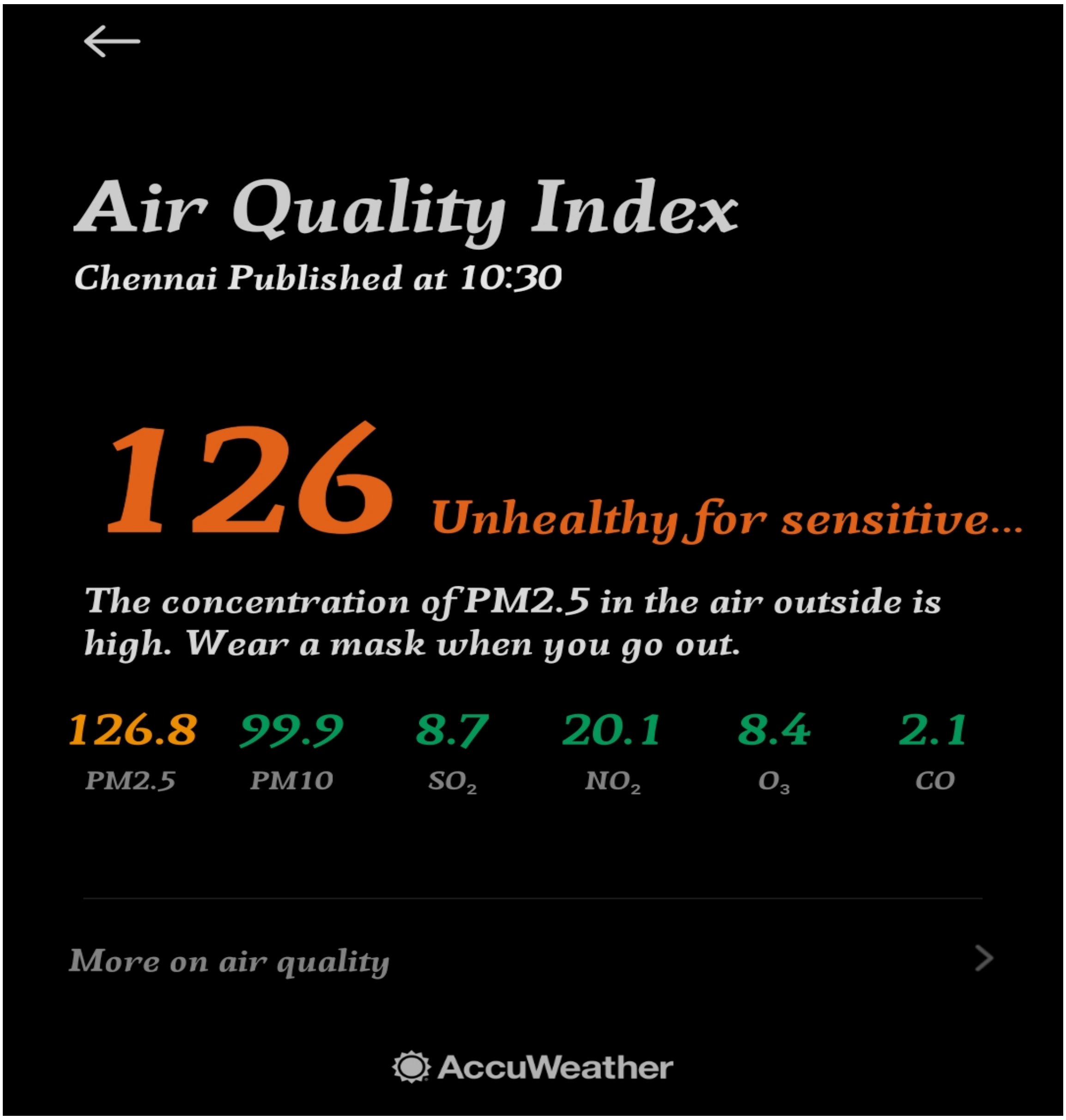
PROJECT NAME: AIR QUALITY MONITORING
PHASE 5: PROJECT DOCUMENTATION AND SUBMISSION

INTRODUCTION:

- Air quality monitoring refers to the systematic, long-term assessment of pollutant levels by measuring the quantity and types of certain pollutants in the surrounding, outdoor air. Pollutants tied to human and environmental health impacts include PM2.5, PM10, ground-level ozone, nitrogen dioxide and sulphur dioxide.
- In this phase, we have created a document combining every phases. We have included the objectives, IoT setup, platform development and code implementation.

OBJECTIVES:

- Determine Air Quality: The general objective of an ambient AQM is to determine quality of air in a region and assess its effects on human health.
- Measure Against Standards: Monitoring helps in assessing the level of pollution in relation to the ambient air quality standards. Standards are a regulatory measure to set the target for pollution reduction and achieve clean air.



- Data Collection and Measurements: Data from the samples are compared to clean air standards and historical information regarding air quality levels, along with data reflecting its health and environmental impacts to determine the state of the air.
- Early Warning: Develop an early warning system to alert the public and authorities about hazardous air quality conditions, enabling timely preventive measures.
- Public Awareness: Raise awareness among the public about air quality issues, its impact on health and ways individuals can contribute to improving air quality.

Problem Definition:

Developing an effective Air Quality Monitoring system that can accurately measure and analyse air pollution could be difficult. Air quality remains a significant environmental health challenge in India and large sections of the population remains in areas with higher population. Efforts are necessary to improve measurement coverage and quality including the use of Air Quality Index Meter to measure the particulate matter present in the air.

- *Accuracy:* The developing model should be accurate as it has various air pollutants to take into account, such as Particulate Matter (PM2.5 and PM10), Nitrogen Dioxide (NO2), Sulphur Dioxide (SO2), Carbon Monoxide (CO) and Ozone(O3) as they are the major particles for measuring air pollutants.

- *Data Accessibility:* To make the model successful, data which has been measured has to accessible to the public for research and awareness of the surroundings.

- *Real Time Monitoring:* The AQM system should monitor continuous and real-time data to allow immediate responses on deteriorating air quality. It has to measure PM for two size ranges,

i)PM2.5: PM2.5 consists of "fine particles" with aerodynamic diameters less than or equal to 2.5 microns.

ii)PM10: PM10 includes both "fine particles" and "coarse particles" with aerodynamic diameters greater than 2.5 microns and less than or equal to 10 microns.

➢ *Scalability*: A system should be designed so that it can be scaled up to cover larger geographical areas.

➢ *Environmental Factors*: The system should measure environmental factors like weather conditions, temperature and humidity which can influence air quality.

Design Thinking:

The AQM system includes setting up IoT devices to measure air quality parameters like pollution levels and particulate matter. This data can be made publicly available to raise awareness about air quality and its impact on public health.

The monitoring system is developed for transmission and reception of information received from various data-sources with micro-controllers. The wireless sensing real-time data are transmitted into desired form across network through internet connection.

Front end application can be developed and hosted on the cloud platform. Additionally, an App can be developed on Android platform to visualize real-time data, which are uploaded in the designed web server, displayed data on the smart phone.

The system can also alert the users if the air quality reaches a dangerous level, allowing them to take precautions to protect themselves.

I. Air Quality Index Calculations:

The AQI in India is calculated based on the average concentration of a particular pollutant measured over a standard time interval (24 hours for most pollutants, 8 hours for carbon monoxide and ozone). The AQI for PM2.5 and PM10 is based on 24-hour average concentration and AQI for Carbon Monoxide is based on 8-hour average concentration. The AQI calculations include the eight pollutants that are

- PM10
- PM2.5
- Nitrogen Dioxide (NO2)
- Sulphur Dioxide (SO2)
- Carbon Monoxide (CO)
- Ground-level ozone (O3)
- Ammonia (NH3)
- Lead (Pb)

However, all of the pollutants are not measured at every location. Based on the measured 24-hour ambient concentrations of a pollutant, a sub-index is calculated, which is a linear function of concentration (e.g., the sub-index for PM2.5 will be 51 at concentration 31 µg/m3, 100 at concentration 60 µg/m3, and 75 at a concentration of 45 µg/m3). The worst sub-index (or maximum of all parameters) determines the overall AQI.

II. Components of IoT Monitoring System:

1)*Sensors*:

Sensors are the primary components of IoT-based air pollution monitoring systems. They measure various air quality parameters such as particulate matter, carbon monoxide, sulphur dioxide, and nitrogen oxides. The sensors can be classified into two categories: physical and chemical sensors. Physical sensors measure parameters such as temperature, humidity, and pressure, while chemical sensors measure air pollutants. E.g.: VOC Sensors

2)*Microcontrollers*:

The microcontroller is the brain of IoT-based air pollution monitoring systems. It receives data from the sensors, processes it, and sends it to the cloud server. The microcontroller is usually a microprocessor such as Arduino, Raspberry Pi, or similar devices.

3)*Communication Module:*

The communication module is responsible for transmitting data from the microcontroller to the cloud server. Communication modules can use various wireless technologies such as Wi-Fi, Bluetooth, or cellular networks.

4)*Cloud Server:*

The cloud server is a centralized platform for storing, analysing, and sharing air quality data. It collects data from the communication module and stores it in a database. The cloud server also provides web and mobile applications for users to access the data. E.g.: esp8266 module can be connected to upload this data to cloud.

5)*Power Supply:*

IoT-based air pollution monitoring systems require a power supply to operate. In case of permanent installations external power supply is provided and batteries are provided for portable devices.

6)*Enclosure:*

The enclosure is the outer covering that protects the components from environmental factors such as dust, water, and temperature.

III. Blog Creation:

IoT can be used to create blogs for better understanding of the data received. A blog can be used to raise awareness about the use of IoT in Air Quality Monitoring and its benefits. It can be used to enhance research and innovation in the field of air quality and environmental science.

Creation of blog on AQM includes the steps such as

- Choose a specific topic or theme of the blog, such as the effects of air pollution on health, the challenges and opportunities of IoT for air quality monitoring.

- Conduct research on the topic using reliable sources, such as scientific journals, reports, websites, etc.
- Write an outline the blog post, including a catchy title, an introduction that summarizes the main idea and purpose of your post.
- The blog post using clear and concise language, avoiding jargon and technical terms that might confuse your readers.
- Use headings, subheadings, bullet points, tables, charts, images, etc., to organize your content and make it more visually appealing.

Visualizing the data of air quality monitoring is a useful way to understand and communicate the levels and trends of air pollutants in the atmosphere. There are different methods and tools that can help to create interactive and informative graphs and charts to display air quality data.

- *Heatmaps:* Heatmaps are plots that use colours to represent the values of a variable across a grid. For example, you can use heatmaps to compare the PM2.5 levels in different cities during a certain period of time, as shown in this blog post. Heatmaps can help you see the patterns and variations of air quality across space and time.
- *Line charts:* Line charts are plots that use lines to connect the values of a variable over time. For example, you can use line charts to show the changes of air quality over time in a single location. Line charts can help see the trends and fluctuations of air quality over time.

- *Pie charts:* Pie charts are plots that use sectors of a circle to represent the proportions of different categories of a variable. For example, you can use pie charts to show the composition of air pollutants in a certain location. Pie charts can help see the relative contributions of different sources or types of air pollutants.
- *Scatter plots:* Scatter plots are plots that use points to represent the values of two variables on a coordinate plane. For example, you can use scatter plots to show the relationship between two air quality parameters, such as temperature and humidity . Scatter plots can help you see the correlation and distribution of air quality data.

To create these kinds of graphs and charts, various tools and software can be used such as Python, R, Excel, Tableau, Power BI, etc.

IOT DEVICE SETUP

I. MICROCONTROLLERS:

i. Arduino UNO R3

- It does not have Operating System on its own.
- Arduino just run C/C++ code that is stored in their firmware.
- It is well-suited for simple, real-time data collection and control tasks.
- It is ideal for small-scale, localized air quality monitoring projects.

ii. Raspberry Pi

- It is a small, low-cost computer that connects to a computer monitor.
- Raspberry Pi is a single-board computer with more processing power and capabilities.
- It can handle complex data processing, web connectivity and storage.
- Raspberry Pi is commonly not used as non-local processing of data is required.

iii. ESP8266:

- ESP8266 offers Wi-Fi connectivity but lacks Bluetooth.
- ESP8266 has a single-core processor, which may limit its performance in more demanding applications.
- ESP8266 consumes less power compared to ESP32.

iv. **ESP32:**

- ESP32 has both Wi-Fi and Bluetooth capabilities, making it versatile for data transmission and communication.
- ESP32 is more powerful in terms of processing and has dual-core processors.
- It can used for complex data and simultaneous process of multiple processes.
- ESP32 can be configured for low-power modes, but it may consume more power compared to ESP8266 in certain scenarios.
- ESP32 may be the better choice for air quality monitoring.

II. SENSORS

i. **Particulate Matter Sensor:**

- Particulate matter sensor measures the quantity of (fine) dust particles in the air, expressed in PM (Particulate Matter).
- PM2.5 and PM10 Sensors: These sensors measure the concentration of fine particulate matter in the air, which includes particles with diameters of 2.5 micrometers or smaller (PM2.5) and 10 micrometers or smaller (PM10).

ii. **Digital Humidity and Temperature Sensors:**

- These sensors provide precise measurements of humidity and temperature, which are crucial for assessing indoor and outdoor air quality.

- There are various types of digital humidity and temperature sensors available, including capacitive, resistive and thermal sensors.
- Digital humidity and temperature sensors offer high accuracy and reliability, making them suitable for applications where precise measurements are essential.
- Proper calibration and maintenance are essential to ensure reliable and accurate measurements in your monitoring system.

iii. <u>Air Quality Index (AQI) Sensors:</u>

- AQI sensors typically measure multiple air pollutants, including:
  - Particulate Matter (PM2.5 and PM10)
  - Ground-level Ozone (O3)
  - Nitrogen Dioxide (NO2)
  - Sulfur Dioxide (SO2)
  - Carbon Monoxide (CO)
- The AQI is calculated based on the concentration of these pollutants and is usually expressed as a numerical value on a scale with different color-coded categories, such as "Good," "Moderate," "Unhealthy," etc.

III. CONNECTIVITY

Air quality monitoring systems use various modes of connectivity to transmit data and communicate with central databases or platforms.

i. <u>Fiber Optic:</u>

In high-density urban areas or research facilities, high-speed fiber optic connections may be used for data transmission.

ii. Ethernet

Some monitoring stations are directly connected to the internet via Ethernet cables, which is common for stationary installations in urban areas or research facilities.

iii. LPWAN (Low Power Wide Area Networking):

LPWAN technologies like Sigfox and NB-IoT are used for low-power, long-range connectivity in IoT-based air quality monitoring.

iv. Wi-Fi:

- Wi-Fi connectivity eliminates the need for physical cables, allowing air quality sensors to transmit data wirelessly to a designated network.
- Researchers and environmental agencies can access and control the equipment from a central location, reducing the need for physical site visits.
- Air quality data collected through Wi-Fi-connected sensors is often made accessible to the public through websites or mobile apps.

IV. CLOUD SERVER

The cloud server is a centralized platform for storing, analyzing, and sharing air quality data. It collects data from the communication module and stores it in a database.

The information gathered by each sensor node is uploaded to a cloud server, where it is stored and can be viewed through a web browser at any time and from any location

i. Beeceptor

Beeceptor is a **powerful tool** for testing, mocking, and intercepting HTTP requests. It provides a **dashboard** to intercept and inspect all HTTP requests in real-time.

ii. **<u>Alibaba Cloud:</u>**

Alibaba Cloud provides cloud computing services with a presence in various regions. It can be used for data storage and processing in air quality monitoring applications, particularly in Asia.

iii. **<u>IBM Cloud:</u>**

IBM Cloud provides services for data storage and analytics, including tools for data warehousing and machine learning. These services can be employed in air quality monitoring projects.

V. PROTOCOL

i. **<u>Environmental Protection Agency:</u>**

These are standardized methods for measuring specific air pollutants, such as PM2.5 and criteria pollutants like sulphur dioxide (SO_2), nitrogen dioxide (NO_2), carbon monoxide (CO) and ozone (O_3).

ii. **<u>Gas Chromatography:</u>**

GC is used to analyse volatile organic compounds (VOCs) and other gaseous pollutants. It separates and quantifies different components in a gas sample.

iii. **<u>HTTP</u>:**

- HTTP (Hypertext Transfer Protocol) is used in AQM to transfer data between the client and server. It is a protocol that defines how messages are formatted and transmitted and what actions Web servers and browsers should take in response to various commands.
- In AQM, HTTP is used to send data from the sensors to the server, where it can be analysed and processed. The server can then send back information to the client, such as alerts or notifications, using HTTP.

iv. MQTT:

- MQTT (Message Queuing Telemetry Transport) is a lightweight messaging protocol that is widely used in Air Quality Monitoring (AQM) systems.
- It helps in efficient data transfer.
- It reduces network bandwidth consumption dramatically.
- Suitable for remote sensing and control.

v. AMQP (Advanced Message Queuing Protocol)

AQM is a messaging protocol that is widely used in AQM systems. It is a reliable and efficient way to transfer data between systems, and supports a wide range of messaging patterns, including point-to-point and publish-subscribe models.

VI. FEATURES

- Real-time monitoring: AQM systems provide real-time monitoring of air quality, which helps in identifying and addressing air pollution issues quickly.
- Data collection: AQM systems collect data from various sources, such as sensors, and transmit it to a central server for analysis and processing.
- Data analysis: AQM systems analyse the collected data to identify patterns and trends in air quality.
- Alerts and notifications: AQM systems can send alerts and notifications to users when air quality levels exceed safe limits.

PLATFORM DEVELOPMENT AND CODE IMPLEMENTATION:

Simulation using Wokwi for ESP32 of Micropython

COMPONENTS USED IN THE PROJECT

- DHT22
- ESP32
- Ultrasonic Distance Sensor

1.DHT22

- DHT22 is commonly used to measure temperature and humidity. It is based on a digital sensor output and can provide high accuracy measurements with a resolution of 0.1-degree Celsius temperature and 0.1% for humidity.
- The sensor can measure temperature from -40 degree Celsius to 80 degree Celsius and humidity from 0% to 100% with an accuracy of ±1°C and ±1%.

2.ESP32

- ESP32 not only has Built in Wi-Fi but also has Bluetooth and Bluetooth Low Energy[1]
- The ESP32 can be used as the main microcontroller to gather data from various sensors (like the

DHT22 for temperature and humidity) and send it to a server for further processing. This can be done through Wi-Fi on the AQM system.

3.ULTRASONIC DISTANCE SENSOR

- Its main function is to measure the distance between the sensor and a target object by sending and receiving ultrasonic waves.
- Incorporating ultrasonic distance sensors to an ESP32, it enhances the system's ability to provide precise and location-optimized measurements, ensuring that the data collected accurately reflects the air quality near the source of interest.

CODE

```
import time

import machine

import dht

import ujson

from umqtt.simple import MQTTClient

try:

    dht_sensor = dht.DHT22(machine.Pin(4))

    mqtt_broker = "your_mqtt_broker_address"
```

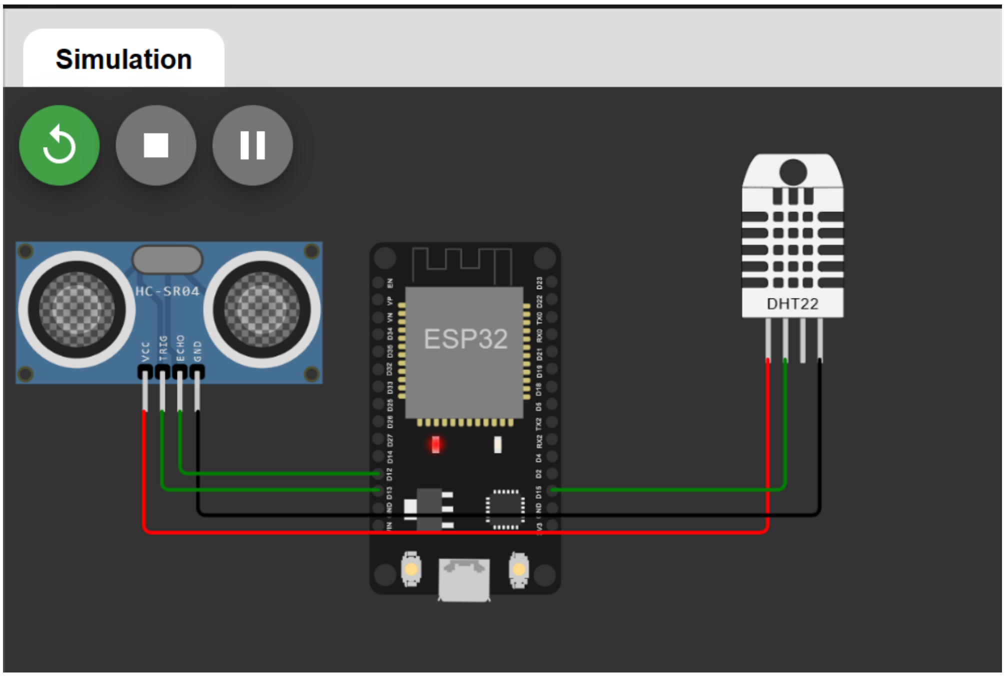
```
mqtt_topic = "your_topic"
mqtt_username = "your_username"
mqtt_password = "your_password"
mqtt_client_id = "your_client_id"
client =MQTTClient(mqtt_client_id,mqtt_broker,
user=mqtt_username, password=mqtt_password)
client.connect()
try:
    while True:
        try:
            dht_sensor.measure()
            temperature_c = dht_sensor.temperature()
            humidity = dht_sensor.humidity()
            data = {
                "temperature": temperature_c,
                "humidity": humidity
                    }
            client.publish(mqtt_topic, ujson.dumps(data))
```

```
            except Exception as e:
                print("Error:", e)
        time.sleep(60)
    except KeyboardInterrupt:
        pass
    client.disconnect()

except OSError as e:
  print(“OSError”,e)
```

OUTPUT

<u>Simulation</u>:



Output:

```
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:2
load:0x3fff0030,len:4728
load:0x40078000,len:14876
ho 0 tail 12 room 4
load:0x40080400,len:3368
entry 0x400805cc
connected to wifi....
   data sent successfully
MicroPython v1.21.0 on 2023-10-05; Generic ESP32 module with ESP32
Type "help()" for more information.
>>>
```

Webpage using HTML, CSS and JavaScript

CODE:

index.html:

```html
<html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>AQM</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <div id="hero">
    <div class="navbar">
      <img src="C:\Users\SANTHA KUMAR\Desktop\sara\example\brown.png" class="logo">
      <div class="user">
        <a href="index2.html" target="_blank">
         <button type="button">AQM</button>
        </a>
      </div>
```

```html
        </div>

        <div class="container">

          <div class="info">

              <h1>AQM</h1>

              <p> Air Quality Monitoring (AQM) is the systematic, long-term  assessment of pollutant levels by measuring the quantity and types  of certain pollutants in the surrounding, outdoor air.It is an integral  part of an effective air quality management system.The locations for  monitoring stations depend on the purpose of the monitoring. <br><br> Most air quality monitoring networks are designed to support  human health objectives, and monitoring stations are established in  population centers. They may be near busy roads, in city centers, or  at locations of particular concern (e.g., a school, hospital, particular  emissions sources). Monitoring stations also may be established to  determine background pollution levels, away from urban areas and  emissions sources</p>

        </div>

    </div>

</body>

</html>
```

style.css:

```css
*{

  margin:0;

  padding:0;

  font-family: timesnewroman;

 }


#hero{

  width:100%;

  height:100vh;

  background-image:url(air6.jpg);

  background-size:cover;

  background-position:center;

  position:relative;

}
```

```css
.navbar{
  width:90%;
  margin:auto;
  display:flex;
  align-items:center;
  justify-content:space-between;
  position:relative;
  z-index:10;
}

.logo{
  top:11%;
  left:90%;
  position:relative;
}

.navbar .logo{
  width:60px;
  margin:30px 0;
  cursor:pointer;
}

.user{
  display:flex;
  align-items:center;

}

.user img{
  width:30px;
  margin-left:50px;
```

```css
  cursor:pointer;
}

button{
  padding:10px 25px;
  background:transparent;
  outline:none;
  border: 5px solid #fff;
  border-radius:20px;
  color:#fff;
  font-size:50px;
  font-weight:bold;
  cursor:pointer;
  top:325%;
  right:75%;
  position:absolute;
}

.container{
   width:45%;
   height:100vh;
   background: rgba(88,86,86,0.3);
   backdrop-filter:blur(10px);
   position:absolute;
   left:0;
   top:0;
}

.info{
   width:550px;
   color:#fff;
```

```
  position:relative;
  right:0%;
  top:0%;

}

.info h1{
  font-size:50px;
  color:#fff;
  letter-spacing:2px;
}

.info p{
  color:#fff;
  font-size:20px;
  margin:20px 0;
  line-height:20px;
}

.info input{
  width:50%;
  padding:8px 10px;
  outline:none;
  border:2px solid #fff;
  border-radius:26px;
  background:transparent;
  color:#420F11;
  font-size:12px;
}
```

index2.html:

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>AQM</title>

  <link rel="stylesheet" href="./style5.css" />

  <link rel="preconnect" href="https://fonts.gstatic.com">

  <link
href="https://fonts.googleapis.com/css2?family=Fira+Code:wght@700&family=Montserrat:wght@700&f
amily=Poppins&display=swap" rel="stylesheet">

</head>

<body>

  <div class="root">

    <h1 class="heading">AQM</h1>

    <div class="location-container">

      <h2 class="sub-heading">Your Location :</h2>

      <label for="error-msg" style="color: rgb(130, 0, 0);"></label><br />

      <input

        type="number"

        name="lat"

        placeholder="Latitude"

        id="latitude"

        step="0.0001"

        class="loc-input" />


      <input

        type="number"

        name="lon"

        placeholder="Longitude"
```

```
        id="longitude"
        step="0.0001"
        class="loc-input" />
    <button class="search-btn">Search</button>
</div>
<div class="air-info">
    <h2 class="sub-heading">Air Quality Index :</h2>
    <span class="air-quality">...</span>
    <span class="arr"> &rarr; </span>
    <span class="air-quality-status">...</span>
    <div class="component-container">
        <h2 class="sub-heading">
            Concentration of pollutants in air :
        </h2>

        <div>
            <h3 class="component-names">Carbon Monoxide :</h3>
            <span class="component-val" data-comp="co"></span>
        </div>
        <div>
            <h3 class="component-names">Nitrogen Monoxide :</h3>
            <span class="component-val" data-comp="no"></span>
        </div>
        <div>
            <h3 class="component-names">Nitrogen Dioxide :</h3>
            <span class="component-val" data-comp="no2"></span>
        </div>
        <div>
            <h3 class="component-names">Ozone :</h3>
            <span class="component-val" data-comp="o3"></span>
        </div>
```

```html
            <div>
               <h3 class="component-names">Sulphur Dioxide :</h3>
               <span class="component-val" data-comp="so2"></span>
            </div>
            <div>
               <h3 class="component-names">Fine particles matter :</h3>
               <span class="component-val" data-comp="pm2_5"></span>
            </div>
            <div>
               <h3 class="component-names">Coarse particulate matter :</h3>
               <span class="component-val" data-comp="pm10"></span>
            </div>
            <div>
               <h3 class="component-names">Ammonia :</h3>
               <span class="component-val" data-comp="nh3"></span>
            </div>
            <!--  micrograms per cubic meter of air (µg/m3)  -->
         </div>
      </div>
   </div>
   <script src="./script1.js" defer></script>
</body>
</html>
```

style5.css:

```
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

.root {
    padding: 2vw;
    background-image: url(air4.jpg);
    width: 100%;
    height: 100%;
    font-family: timesnewroman;
    min-height: 100vh;
}

.heading {
    text-align: center;
    font-family: 'timesnewroman';
    font-size: 3rem;
    margin-bottom: 60px;
}

.sub-heading {
    margin: 20px 0;
    font-family: 'timesnewroman';
    font-size: 2rem;
}

.location-container {
    margin: 40px 0;
```

```
}

.loc-input {
    border: 3px solid #269fe6;
    font-size: 1rem;
    padding: 8px;
    border-radius: 6px;
    margin: 10px;
    width: 270px;
}

.loc-input:invalid {
    border: 3px solid #d34545;
}

.search-btn {
    padding:10px 25px;
  background:transparent;
  outline:none;
  border: 5px solid #fff;
  border-radius:20px;
  color:#fff;
  font-size:50px;
  font-family:"timesnewroman";
  font-weight:bold;
  cursor:pointer;
  top:5%;
  right:5%;
  position:absolute;
}
```

```
.air-quality, .arr, .air-quality-status {
    font-size: 4rem;
    font-weight: 700;
}

.component-names {
    display: inline-block;
}

.component-container div {
    margin: 10px 0;
}

.component-val {
    font-size: 1.2rem;
    margin-left: 20px;
}
```

script1.js:

```
const errorLabel = document.querySelector("label[for='error-msg']")
const latInp = document.querySelector("#latitude")
const lonInp = document.querySelector("#longitude")
const airQuality = document.querySelector(".air-quality")
const airQualityStat = document.querySelector(".air-quality-status")
const srchBtn = document.querySelector(".search-btn")
const componentsEle = document.querySelectorAll(".component-val")

const appId = "a9c6b366e58335198d4d23162b5ff2b4"
```

```
const link = "https://api.openweathermap.org/data/2.5/air_pollution"    // API end point

const getUserLocation = () => {
    if (navigator.geolocation) {
        navigator.geolocation.getCurrentPosition(onPositionGathered, onPositionGatherError);
    } else {
        onPositionGatherError({ message: "Can't Access your location. Please enter your co-ordinates" });
    }
}

const onPositionGathered = (pos) => {
    let lat = pos.coords.latitude.toFixed(4), lon = pos.coords.longitude.toFixed(4);

    latInp.value = lat;
    lonInp.value = lon;

    getAirQuality(lat, lon);
}

const getAirQuality = async (lat, lon) => {
    const rawData = await fetch(`${link}?lat=${lat}&lon=${lon}&appid=${appId}`).catch(err => {
        onPositionGatherError({ message: "Something went wrong. Check your internet conection." });
        console.log(err);
    })
    const airData = await rawData.json();
    setValuesOfAir(airData);
    setComponentsOfAir(airData);
}

const setValuesOfAir = airData => {
    const aqi = airData.list[0].main.aqi;
```

```
let airStat = "", color = "";

airQuality.innerText = aqi;

switch (aqi) {
    case 1:
        airStat = "Good";
        color = "rgb(19, 201, 28)";
        break;
    case 2:
            airStat = "Fair";
            color = "rgb(15, 134, 25)";
            break;
    case 3:
            airStat = "Moderate";
            color = "rgb(201, 204, 13)";
            break;
    case 4:
            airStat = "Poor";
            color = "rgb(204, 83, 13)";
            break;
    case 5:
        airStat = "Very Poor";
        color = "rgb(204, 13, 13)";
        break;
    default:
        airStat = "Unknown";
}

airQualityStat.innerText = airStat;
airQualityStat.style.color = color;
```

```
}

const setComponentsOfAir = airData => {
   let components = {...airData.list[0].components};
   componentsEle.forEach(ele => {
      const attr = ele.getAttribute('data-comp');
      ele.innerText = components[attr] += " µg/m³";
   })
}

const onPositionGatherError = e => {
   errorLabel.innerText = e.message;
}

srchBtn.addEventListener("click", () => {
   getAirQuality(parseFloat(latInp.value).toFixed(4), parseFloat(lonInp.value).toFixed(4));
})

getUserLocation()
```

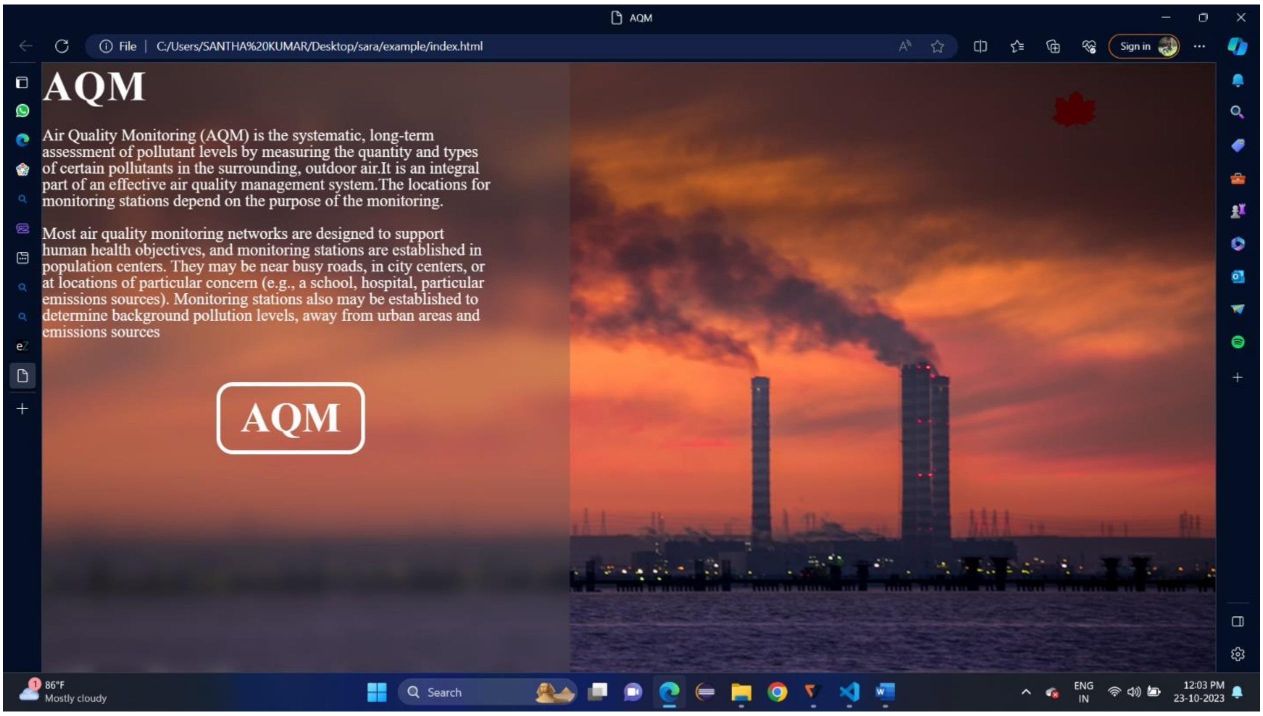
- In this program, we have linked an Application Program Interface (API) called "OpenWeatherMap" which helps to provide a way for different software applications to communicate with each other and exchange data.
- A "key" is created in the OpenWeatherMap API to access the data from the website.
- Use of API in AQM includes,
    - Real-time Data Access

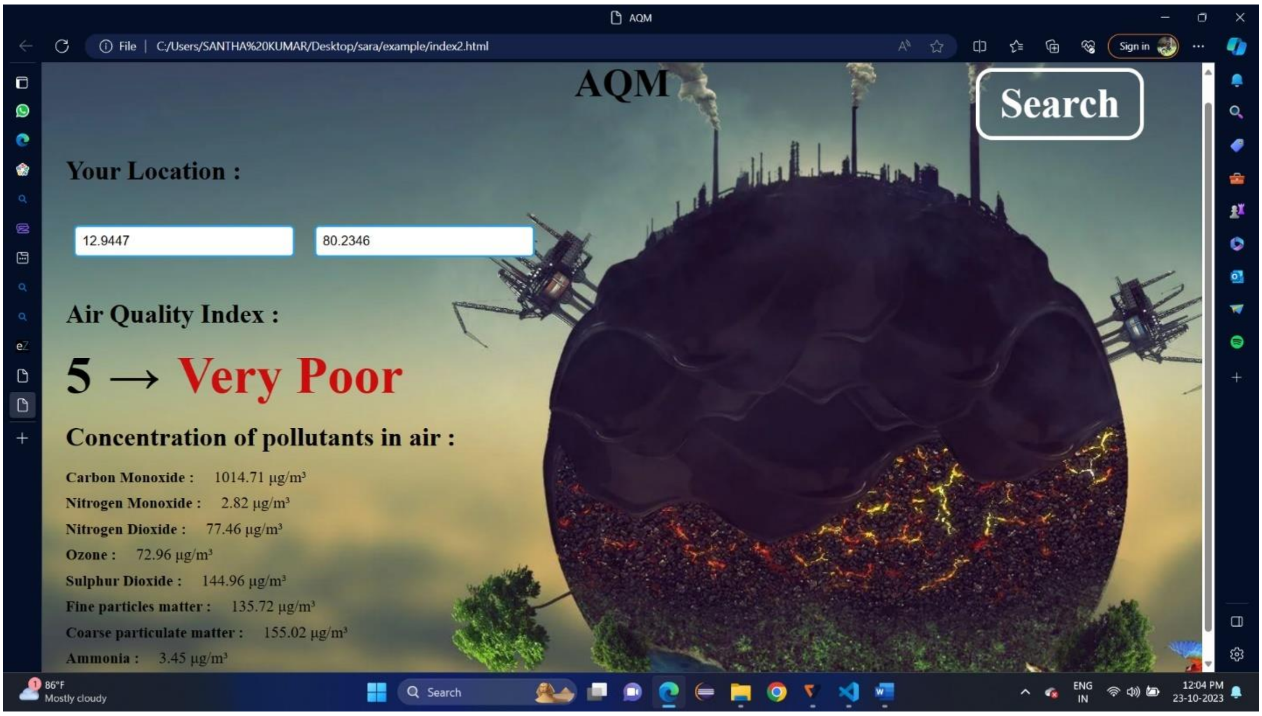
- ❖ Current Air Pollution Data
- ❖ Data Visualization
- ❖ Historical Data Access
- ❖ Pollutant Information
- ❖ Forecasting
- ❖ Integration with other services
- ❖ Development of user-friendly website

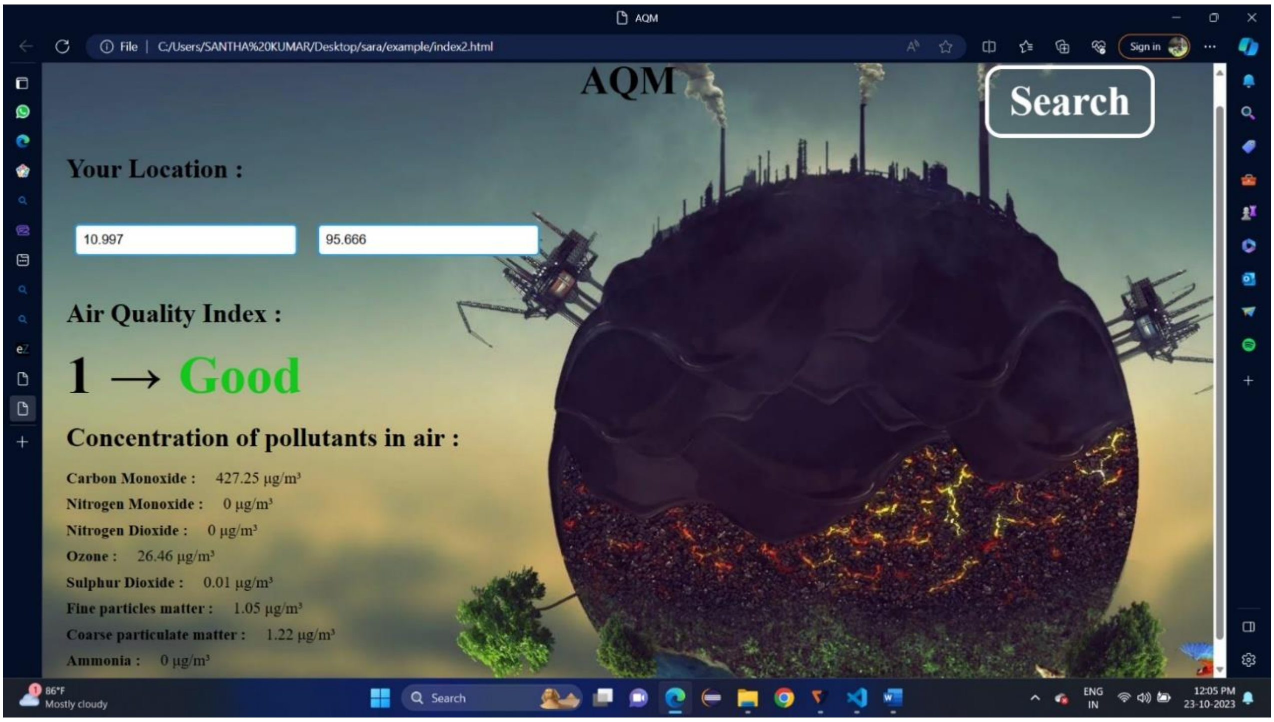
➢ Here, geolocation is received as input such as latitude and longitude in order to check the air quality.

➢ Pollutant Information such as “Carbon Monoxide”, “Nitrogen Monoxide”, “Nitrogen Dioxide”, “Ozone”, “Sulphur Dioxide”, “PM2.5”, “PM10” and “Ammonia” can be received to monitor the Air Quality.

OUTPUT




AQM
C:/Users/SANTHA%20KUMAR/Desktop/sara/example/index2.html
AQM
Search
Your Location :
12.9447
80.2346
Air Quality Index :
5 → Very Poor
Concentration of pollutants in air :
Carbon Monoxide : 1014.71 µg/m³
Nitrogen Monoxide : 2.82 µg/m³
Nitrogen Dioxide : 77.46 µg/m³
Ozone : 72.96 µg/m³
Sulphur Dioxide : 144.96 µg/m³
Fine particles matter : 135.72 µg/m³
Coarse particulate matter : 155.02 µg/m³
Ammonia : 3.45 µg/m³
86°F
Mostly cloudy
12:04 PM
23-10-2023


AQM
C:/Users/SANTHA%20KUMAR/Desktop/sara/example/index2.html
AQM
Search
Your Location :
10.997
95.666
Air Quality Index :
1 → Good
Concentration of pollutants in air :
Carbon Monoxide : 427.25 µg/m³
Nitrogen Monoxide : 0 µg/m³
Nitrogen Dioxide : 0 µg/m³
Ozone : 26.46 µg/m³
Sulphur Dioxide : 0.01 µg/m³
Fine particles matter : 1.05 µg/m³
Coarse particulate matter : 1.22 µg/m³
Ammonia : 0 µg/m³
86°F
Mostly cloudy
12:05 PM
23-10-2023

CONCLUSION:

In conclusion, air quality monitoring is a critical aspect of environmental management and public health. It provides valuable data that can be used to assess the state of the air, measure it against established standards, and establish long-term trends. This information is essential for making informed decisions about environmental policies and interventions.

Moreover, air quality monitoring plays a significant role in protecting public health by identifying potential risks and informing strategies to mitigate them. It also contributes to environmental protection by helping to identify sources of pollution and areas of concern.