**AIR QUALITY** **MONITERING**

**TEAM MEMBERS**

**1)B VIGNESH**

**2)R SHOBAN BABU**

**3)K SANJIV KANTH**

**4)M VIMAL BABU**

**5)M YUGANDHAR**

# PYTHON SCRIPT FOR AIR QUALITY MONITERING

Before you run the script, make sure to install the **requests** library if you haven't already. You can install it using pip: **pip install requests**

## PYTHON SCRIPT

import requests

def get\_air\_quality(location, parameter='pm25'):

base\_url = "https://api.openaq.org/v1/measurements"

# Define the parameters for the API request

params = {

'location': location,

'parameter': parameter,

'limit': 1, # You can adjust the limit to get more data points

'order\_by': 'datetime',

'sort': 'desc',

}

try:

response = requests.get(base\_url, params=params)

response.raise\_for\_status()

data = response.json()

if data['results']:

measurement = data['results'][0]

value = measurement['value']

unit = measurement['unit']

timestamp = measurement['date']['utc']

print(f"Air quality in {location}: {value} {unit} at {timestamp}")

else:

print(f"No air quality data available for {location}.")

except requests.exceptions.RequestException as e:

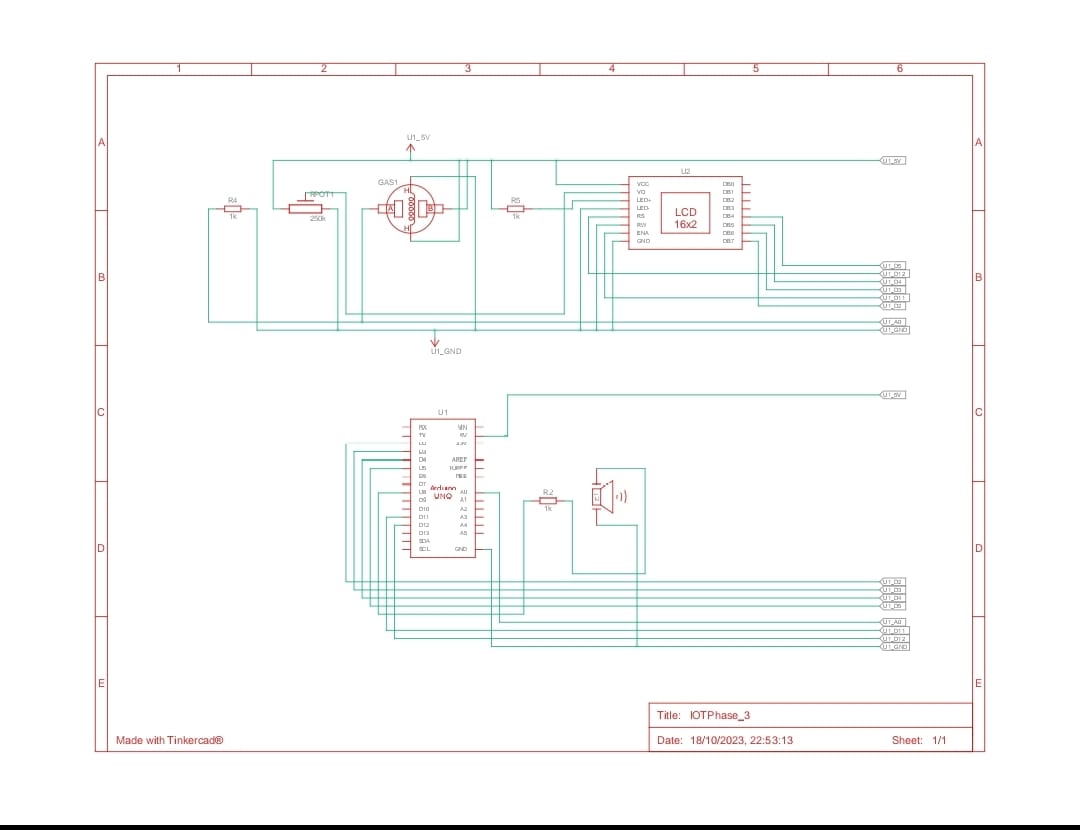
print(f"An error occurred while fetching data: {e}")

if \_\_name\_\_ == "\_\_main\_\_":

location = "chennai" # Replace with the desired location (e.g., a city or specific site)

parameter = "pm10" # You can change this to other parameters like "pm10" or "o3"

get\_air\_quality(location, parameter)



### DOCUMENT FOR AIR QUALITY MONITERING

Import Necessary Libraries:

i)import requests # for making HTTP requests to fetch data

ii)import json # for parsing JSON data

iii)import time # for scheduling data collection

Define API Endpoint (if applicable):

If you are fetching air quality data from an external API, define the API endpoint and any required API keys.

Set Up Data Collection:

Define the location or coordinates for which you want to monitor air quality.

Specify a data collection interval (e.g., every 15 minutes).

Create a Data Retrieval Function:

Create a function to fetch air quality data from a sensor or an API. This function should make HTTP requests, parse the response, and return relevant air quality parameters.

Data Analysis and Processing:

Analyze the retrieved data, including parameters like PM2.5, PM10, AQI (Air Quality Index), temperature, humidity, etc.

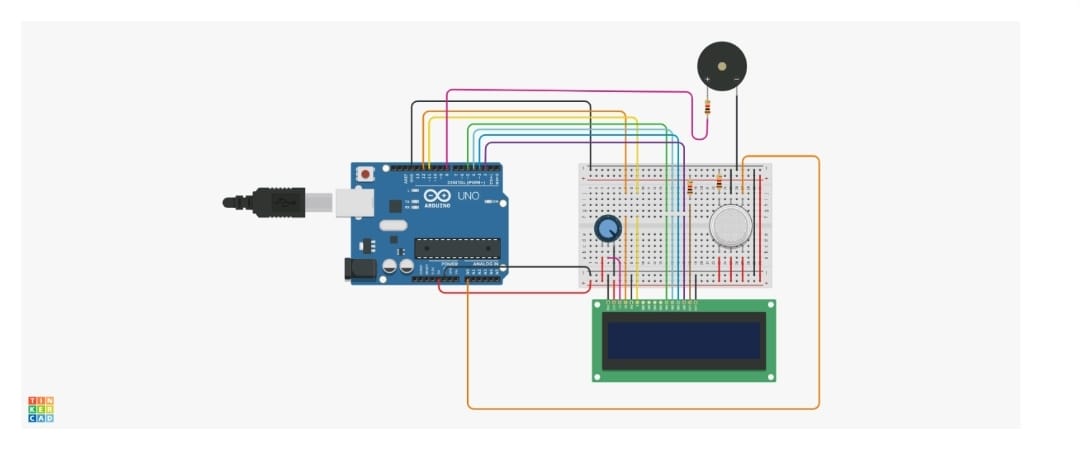
Calculate and display statistics or trends in air quality over time.

Data Storage (Optional):

Store collected data in a local database or cloud storage for historical analysis.

Visualization (Optional):

Use libraries like Matplotlib or Plotly to create graphs and charts for visualizing air quality trends.



Alerts and Notifications (Optional):

Set up notifications or alerts based on predefined air quality thresholds. Send notifications via email, SMS, or other communication channels when air quality deteriorates.

Main Loop:

Continuously run the data collection and analysis process within a loop, respecting the defined data collection interval.

Error Handling:

Implement error handling to manage issues like network connectivity problems, API rate limits, and data parsing errors.

Logging:

Log relevant information and errors for troubleshooting and historical reference.

Run the Script:

Execute the script to start the air quality monitoring process.

