**HARDWARE COMPONENTS;**

* Hardware Specifications
* Arduino Uno
* Wifi Module
* Temperature Humidity Sensor
* Ultrasonic Sensor
* **Piezoelectric Sensors**
* LCD Display
* Resistors
* Capacitors
* Transistors
* Cables and Connectors
* Diodes
* PCB and Breadboards
* LED
* Transformer/Adapter
* Push Buttons
* Switch
* IC
* IC Sockets

**SOFTWARE COMPONENTS ;**

* Wokwi
* Arduino code
* ESP32 DevKit
* Visual code studio

**DEVLOPMENT OF NOISE POLLUTION MONITERING;**

• Sends alert in the event of sound levels and classification such as warning ,critical and high critical.

. • Noise sensors collect data and transmit it to a central hub or gateway. The data can be transmitted via wired or wireless connections, such as Wi-Fi, LoRa, NB-IoT, or cellular networks, depending on the deployment location and connectivity option.

The data received from the sensors is processed and stored in a cloud-based platform or a local server. Cloud platforms like AWS, Azure, Google Cloud, or dedicated IoT platforms like ThingSpeak or Adafruit IO can be used.

* By implementing IoT-based noise pollution monitoring, you can gather detailed and real-time data to assess the impact of noise pollution, identify sources, and take corrective actions when necessary. This approach is particularly useful for smart cities, industrial areas, construction sites, and other locations where noise pollution is a concern.
* The collected data can be used for regulatory compliance reporting or for research purposes, helping to address noise pollution issues at both local and regional levels
* Creating a full-fledged IoT-based noise pollution monitoring system in Python is a complex task that involves hardware integration, data transmission, and data analysis. However, I can provide you with a simplified Python script that simulates the collection of noise data and sends it to a hypothetical cloud. Please note that this script serves as a basic example and does not include actual IoT sensor integration. In a real-world scenario, you would need to use compatible noise sensors and IoT
* PYTHON SCRIPT

import time

import random

import paho.mqtt.client as mqtt

# Simulated noise sensor data

def generate\_noise\_data():

return round(random.uniform(40, 90), 2) # Simulating noise levels between 40 and 90 dB

# MQTT settings

mqtt\_broker = "mqtt.eclipse.org" # Replace with your MQTT broker URL

mqtt\_port = 1883

mqtt\_topic = "noise\_data"

# Create an MQTT client

client = mqtt.Client("NoiseSensor")

# Connect to the MQTT broker

client.connect(mqtt\_broker, mqtt\_port)

try:

while True:

noise\_level = generate\_noise\_data()

# Simulate sending noise data to the MQTT broker

client.publish(mqtt\_topic, noise\_level)

print(f"Published noise level: {noise\_level} dB")

time.sleep(10) # Simulate data transmission every 10 seconds

except KeyboardInterrupt:

pass

# Disconnect from the MQTT broker

client.disconnect()

PROJECT SUBMITTED BY:

Name: Sana prudvinath

Reg no:713921106046

Nm ID:au713921106046

Mail id: nprudvi25@gmail.com

College code:7139