Covid 19 Case Analysis

phase 5

INTRODUCTION:

Creating a real-time noise level monitoring system with IoT sensors, a noise pollution information platform, and a mobile app involves multiple components. While I can't provide actual diagrams, schematics, screenshots, or code implementation due to the limitations of this text-based format, I can describe each component and its purpose.

Project Objectives:

1. Real-time Noise Monitoring: Develop a system that can continuously monitor noise levels in specific areas.

2. Data Collection: Gather and store noise data from IoT sensors for analysis.

3. Data Visualization: Create a user-friendly platform and mobile app to display noise pollution data.

4. Public Awareness: Promote public awareness of noise pollution and encourage responsible behavior.

5. Noise Pollution Mitigation: Contribute to reducing noise pollution through informed decision-making.

IoT Sensor Deployment:

The IoT sensors are deployed in strategic locations throughout the target area. These sensors consist of:

1. Microphones: To capture ambient noise levels.

2. Data Processors: To process and filter the noise data.

3. Connectivity Modules: To transmit data to the central platform.

Sensors are powered by battery or solar panels, and they communicate via Wi-Fi or cellular networks. They collect data in real-time and send it to the central server for processing and analysis.

Platform Development:

The Noise Pollution Information Platform serves as a central hub for data management and visualization. It includes:

1. Data Storage: A database to store historical noise data.

2. Data Analysis: Algorithms to process, filter, and analyze incoming noise data.

3. Data Visualization: Charts, graphs, and maps to display real-time and historical noise levels.

4. User Management: Authentication and authorization for users, including administrators, researchers, and the general public.

5. Alerting System: Automated alerts for excessive noise levels.

Mobile App Development:

The mobile app is designed for both iOS and Android devices and provides access to the noise data. It features:

1. Real-time Noise Monitoring: Displays noise levels from nearby sensors in real-time.

2. Noise History: Allows users to access historical noise data.

3. Maps: Displays sensor locations and their noise levels on an interactive map.

4. User Profiles: Allows users to customize alerts and preferences.

5. Feedback System: Enables users to report noise disturbances and submit feedback.

Code Implementation:

The code base for this project involves multiple components:

- Sensor Firmware: Code to capture, process, and transmit noise data.

- Platform Backend: Server-side code to handle data storage, analysis, and user management.

- Platform Frontend: Code for the web-based platform's user interface.

- Mobile App: Code for the mobile app, developed separately for iOS and Android.

- API Integration: Code to enable communication between the sensors, platform, and mobile app.

Promoting Public Awareness and Mitigation:

The real-time noise level monitoring system promotes public awareness and contributes to noise pollution mitigation in several ways:

1. Data Accessibility: It makes noise data easily accessible to the public, allowing individuals to see the noise levels in their area.

2. Education: Users can learn about the impacts of noise pollution and make informed decisions to reduce their own noise emissions.

3. Community Engagement: By reporting noise disturbances and submitting feedback through the app, users can actively participate in addressing noise pollution.

4. Regulatory Compliance: Authorities can use this data to enforce noise regulations and take action against excessive noise.

5. Preventative Measures: Businesses and individuals can take preventative measures to reduce noise pollution, leading to a quieter and more peaceful environment.

Overall, the system empowers communities to take actions to mitigate noise pollution, ultimately improving the quality of life for residents and contributing to a quieter and more sustainable urban environment.

Python script for simulating an IoT noise sensor:

```
import random
import time
import requests

# Simulated IoT sensor parameters
sensor_id = "sensor001"
api_url = "http://your-server-url/api/noise"

def simulate_noise_sensor():
    while True:
        # Simulate noise level (in dB) in the range of 40 to 90 dB
        noise_level = random.uniform(40, 90)

        # Simulate a timestamp (you may want to use a real-time clock here)
        timestamp = time.strftime("%Y-%m-%d %H:%M:%S")

        # Prepare the data to send to the server
        data = {
            "sensor_id": sensor_id,
            "noise_level": noise_level,
            "timestamp": timestamp
        }

        # Send data to the server
        try:
            response = requests.post(api_url, json=data)
            if response.status_code == 200:
                print(f"Data sent: {data}")
            else:
                print(f"Failed to send data. Status code: {response.status_code}")
        except Exception as e:
            print(f"Error: {str(e)}")

        # Send data every 5 minutes (adjust as needed)
        time.sleep(300)

if __name__ == "__main__":
    simulate_noise_sensor()
```

Additionally, you should replace **"http://your-server-url/api/noise"** with the actual URL of your server where the data should be sent.

THANK YOU…