Building an IoT air quality monitoring system involves multiple steps, and I can provide you with an outline for Phase 3, Development Part 1. Keep in mind that the specifics may vary depending on the hardware and data-sharing platform you are using. Here's a general guideline:

1. Select IoT Devices and Sensors:Choose the appropriate sensors for measuring air quality parameters like pollution levels, particulate matter (PM2.5, PM10), temperature, humidity, and more. Common sensors used for this purpose include gas sensors, dust sensors, temperature and humidity sensors, and possibly a GPS module.

2. Set Up IoT Hardware:Assemble and configure the IoT hardware components. This typically involves connecting sensors to your IoT device (e.g., Raspberry Pi, Arduino, ESP8266/ESP32), providing power, and setting up any required circuits.

3. Install Required Software and Libraries:Ensure that your IoT device is equipped with the necessary software and libraries to interface with the sensors. For example, you might need to install Python and libraries like Adafruit\_DHT for temperature and humidity sensors and libraries specific to your gas and particulate matter sensors.

4. Collect Data from Sensors:Write a Python script on your IoT device to collect data from the sensors. The script should read data from each sensor periodically and store it in variables.

pythonCopy code

Example code for reading data from a DHT22 sensor (temperature and humidity) importAdafruit\_DHTsensor = Adafruit\_DHT.DHT22pin = 4# GPIO pin where the sensor is connected humidity, temperature = Adafruit\_DHT. read\_retry(sensor, pin) if humidity is not none and temperature is not none: print(f'Temperature: {temperature}C') print(f'Humidity: {humidity}%') else:print('Failed to retrieve data from the sensor')

5. Data Formatting and Transmission:Format the collected data into a suitable format (e.g., JSON) and send it to your chosen data-sharing platform. You can use MQTT, HTTP, or other communication protocols, depending on your platform. Ensure that you have the necessary credentials and endpoints for data transmission.

pythonCopy code

# Example code for sending data over MQTTimportpaho.mqtt.client asmqttclient = mqtt. Client()client.connect("mqtt.broker.com", 1883)data = {"temperature": temperature,"humidity": humidity, # Add more data for other sensors}client.publish("air\_quality\_data", json.dumps(data))

6. Test Data Transmission:Verify that your IoT device successfully sends data to the data-sharing platform. You can monitor the data using the platform's tools or services to ensure that it's being received correctly.

This outline should help you get started with building your IoT air quality monitoring system. Please adapt the steps and code to your specific sensors, hardware, and data-sharing platform. Additionally, it's crucial to consider security and error handling in your IoT device and data transmission code.