Development Part1

## Phase 3

Start building the IoT-enabled Environmental Monitoring in Parks system.

Deploy IoT devices (e.g., temperature and humidity sensors) in various locations within public parks to measure environmental conditions.

* **Plan and Design:**

Define the scope and objectives of the system.

Identify the number and types of sensors needed (e.g., temperature, humidity, air quality).

Determine the locations within the parks to deploy sensors.

Plan the data collection frequency and method.

Consider power sources and communication methods for IoT devices (e.g., battery-powered, Wi-Fi, LoRa, cellular).

Create a system architecture and data flow diagram.

* **Select IoT Hardware:**

Choose IoT sensors and devices compatible with your requirements.

Ensure they have the necessary communication capabilities and power management features.

Procure the hardware components.

Sensor Deployment:

Install sensors at selected locations within the parks.

Ensure they are protected from environmental conditions.

Set up a power source if needed (e.g., solar panels or batteries).

Calibrate sensors for accurate data collection.

IoT Gateway and Connectivity:

Set up a central IoT gateway to collect data from sensors.

Establish a secure and reliable communication network (e.g., IoT protocols, Wi-Fi, LoRaWAN).

Ensure data transmission is encrypted and secure.

* **Data Storage and Processing:**

Choose a cloud or on-premises solution for data storage.

Implement data processing and analysis algorithms to derive meaningful insights.

Create data visualization tools or dashboards for real-time monitoring.

Data Analysis and Alerts:

Define threshold values for environmental parameters.

Implement alerting mechanisms for when conditions exceed specified limits.

User Interface:

Develop a user-friendly interface for park staff or visitors to access environmental data.

Consider mobile apps or web-based dashboards.

* **Maintenance and Monitoring:**

Regularly check and maintain sensors and hardware.

Monitor data quality and system performance.

Address issues promptly.

Scaling and Expansion:

Plan for future expansion or additional sensors if needed.

Continuously improve the system based on user feedback and changing requirements.

* **Compliance and Privacy:**

Ensure compliance with data privacy regulations.

Secure the system against cyber threats and unauthorized access.

Remember to involve experts in IoT, data analysis, and environmental monitoring to ensure a successful deployment.

**Develop a Python script on the IoT devices to send real-time environmental data to the monitoring platform.**

To send real-time environmental data from IoT devices to a monitoring platform using Python, you'll need to write a script that reads data from the sensors and transmits it to the platform. Below is a simplified example using the MQTT protocol to send data to an MQTT broker, which can be part of your monitoring platform:

import paho.mqtt.client as mqtt

import random

import time

# Configuration for MQTT broker

broker\_address = "your\_mqtt\_broker\_address"

port = 1883

topic = "environmental\_data"

# Simulated sensor data (replace with real sensor readings)

def read\_sensor\_data():

temperature = random.uniform(15, 35)

humidity = random.uniform(30, 80)

return {"temperature": temperature, "humidity": humidity}

# Callback when the client is connected to the broker

def on\_connect(client, userdata, flags, rc):

print("Connected with result code " + str(rc))

client.subscribe(topic)

# Initialize MQTT client

client = mqtt.Client()

client.on\_connect = on\_connect

# Connect to the MQTT broker

client.connect(broker\_address, port, 60)

while True:

# Read sensor data

sensor\_data = read\_sensor\_data()

# Publish data to the specified topic

client.publish(topic, str(sensor\_data))

print("Published data:", sensor\_data)

# Adjust the interval based on your data collection needs

time.sleep(60) # Publish data every 60 seconds

Make sure to replace "your\_mqtt\_broker\_address" with the actual address of your MQTT broker, and adjust the read\_sensor\_data() function to read data from your sensors. Additionally, you can modify the publishing frequency (time.sleep()) to match your data collection interval.

This script uses the Paho MQTT client library, so you may need to install it using pip install paho-mqtt if you haven't already. You should also ensure that your IoT device has network connectivity and the necessary credentials to connect to your MQTT broker securely.