

Environmental Monitoring

Phase 3: Development Part 1

In this part you will begin building your project.

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.

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

Solution:

Creating an IoT-enabled Environmental Monitoring in Parks system involves deploying IoT devices, such as temperature and humidity sensors, and developing a Python script to collect and transmit real-time environmental data to a monitoring platform. Here's a step-by-step guide to help you get started:

Hardware Setup:

1. **Select IoT Devices:** Choose suitable IoT devices and sensors for monitoring temperature and humidity. There are various options available in the market that can connect to IoT platforms. Make sure they are designed for outdoor use if they will be deployed in public parks.
2. **Deploy Sensors:** Place the sensors in various locations within public parks where you want to monitor environmental conditions. Ensure that they are securely mounted and protected from the elements, vandalism, and theft.
3. **Power Supply:** Provide a reliable power source for the IoT devices. Depending on the devices used, this could be batteries, solar panels, or a wired power supply.
4. **Connectivity:** Ensure that the IoT devices have a reliable network connection. You can use Wi-Fi, cellular, LoRaWAN, or other connectivity options, depending on the park's infrastructure.

Software Development:

1. **Install Necessary Libraries:** Make sure you have the required libraries and dependencies installed on your IoT devices. You might need libraries for sensor data processing and communication.
2. **Develop Python Script:**
 - a). Create a Python script on each IoT device that reads data from the environmental sensors. Below is a basic script outline:

PROGRAM:

```
# Import necessary libraries
```

```
import time
```

```
import sensor_library # Replace with the actual library for your sensors
```

```

import data_sender # Library for sending data to the monitoring platform

# Initialize the sensors

sensors = sensor_library.initialize_sensors() # Replace with your sensor initialization code

while True:

    # Read data from the sensors

    environmental_data = {}

    for sensor in sensors:

        sensor_data = sensor.read_data() # Replace with your sensor's data retrieval code

        environmental_data[sensor.name] = sensor_data

    # Send data to the monitoring platform

    data_sender.send_data(environmental_data)

    # Wait for a specified interval before the next reading

    time.sleep(300) # Adjust the interval as per your requirements (e.g., every 5 minutes)

```

b). Replace "sensor_library" and "data_sender" with the actual libraries or functions needed for your specific sensors and monitoring platform.

3. **Data Processing:** Depending on your requirements, you might need to process the data within the script, such as calculating averages, handling outliers, or converting units.
4. **Data Transmission:** Implement the data_sender module to send data to your environmental monitoring platform. This could involve using APIs, MQTT, HTTP, or other communication methods.

Testing and Deployment:

1. **Testing:** Test the setup by running the Python script on your IoT devices and monitor the data sent to your monitoring platform. Ensure that the data is accurate and that the script functions as expected.
2. **Deployment:** Install the IoT devices with sensors in various locations within the public parks.
3. **Monitoring and Maintenance:** Regularly monitor the system to ensure it's working correctly and perform maintenance as needed, including changing batteries or cleaning sensors.

Ensure that the monitoring platform can receive and display data from these IoT devices and that you have proper data visualization and alerting mechanisms in place to make the most of the collected environmental data.