ENVIRONMENTAL MONITORING USING INTERNET OF THINGS

TEAM MEMBER:960321104040

JESHIKO J

Phase-3 Document Submission

Project: Environmental Monitoring

INTRODUCTION:

Environmental monitoring in a park system plays a crucial role in preserving the natural beauty and ecological balance of these pristine spaces. By systematically assessing and tracking various environmental factors, such as air and water quality, wildlife populations, and habitat health, park authorities can make informed decisions to protect these precious resources for current and future generations. This introduction sets the stage for a comprehensive discussion of the importance and methods of environmental monitoring within park systems.

CONTENT FOR PHASE-3

Loading Data Set:

```
import time
import random
import paho.mqtt.client as mqtt

# Set up MQTT parameters
mqtt_broker = "your.mqtt.broker.address"
mqtt_port = 1883
mqtt_topic = "environmental_data"

# Initialize the MQTT client
client = mqtt.Client("IoT_Device")

# Connect to the MQTT broker
client.connect(mqtt_broker, mqtt_port, 60)

while True:
    # Simulate environmental data (replace with actual sensor
readings)
    temperature = random.uniform(20, 30)
    humidity = random.uniform(40, 60)

    # Create a JSON payload with the environmental data
    data = {
        "temperature": temperature,
        "humidity": humidity
    }

    # Publish the data to the MQTT topic
    client.publish(mqtt_topic, payload=str(data), qos=0,
retain=False)
```

```
    # Print the data for debugging (remove in a production environment)
    print(f"Published: {data}")

    # Sleep for a specified interval (e.g., 5 seconds)
    time.sleep(5)
```

PREPROCESSING DATASET FOR ENVIRONMENTAL MONITORING PARK SYSTEM

1. Data Collection: Gather data from sensors and other sources in the park system, ensuring that you have accurate timestamps and sensor identifiers.

2. Data Cleaning:
   - Remove or interpolate missing values in the dataset.
   - Check for outliers and anomalies in the data and decide how to handle them (e.g., remove or correct).

3. Data Integration(if applicable): If you have data from multiple sensors or sources, integrate them into a single dataset. Ensure that the data aligns in terms of timestamps and units.

4. Timestamp Alignment: Make sure all data points are recorded at consistent time intervals. You may need to resample or interpolate data to achieve this.

5. Data Normalization: If the data comes from different sensors with varying scales, normalize it so that all variables are on the same scale. This is important for machine learning models.

6. Feature Engineering (if needed): Create additional features or variables that could be useful for analysis. For example, you might calculate daily averages, minima, and maxima for temperature and humidity.

7. Data Splitting: Divide the dataset into training and testing sets for model development and evaluation.

8. Data Visualization: Create visualizations to explore the data and gain insights. This could involve time series plots, histograms, or scatter plots, depending on your goals.

9. Data Format: Ensure that the dataset is in a format suitable for your chosen analysis tools, such as CSV, JSON, or a database format.

10.Quality Assurance: Double-check the dataset for any remaining errors or inconsistencies.

11. Documentation: Document the preprocessing steps taken and any assumptions or decisions made during the process. This documentation is essential for transparency and future reference.

12. Backup and Version Control: Keep a backup of the original dataset and consider using version control for your preprocessed data to track changes.

SOFTWARE IMPLEMENTATION

1. Hardware Components:
   - Sensors: Choose appropriate sensors for humidity and temperature. Common options include DHT22, DHT11, or more advanced sensors like the DHT22, BME280, or SHT series.
   - Microcontroller/Single-Board Computer Use platforms like Arduino, Raspberry Pi, or ESP8266/ESP32 to interface with the sensors.
   - Internet Connectivity: If you want remote monitoring, ensure that your hardware has internet connectivity capabilities, such as Wi-Fi or Ethernet.

2. Software Implementation:
   - Sensor Integration: Write code to interface with the sensors and read humidity and temperature data.
   - Data Logging:Store the collected data. You can use databases like MySQL, SQLite, or cloud-based solutions like Google Cloud Firestore or AWS DynamoDB.
   - Data Processing: Implement data processing if needed. For example, you might want to calculate averages, trends, or trigger alerts based on the data.
   - User Interface:Create a user interface to view data. This could be a web application, mobile app, or a dashboard on a microcontroller's screen.
   - Alerts: Implement alerts for conditions that require immediate attention, such as out-of-range temperature or humidity levels.
   - Data Visualization:Use libraries like Matplotlib or Plotly for data visualization, or create web-based dashboards using tools like Grafana or ThingSpeak.
   - Remote Access: If you want remote access, ensure your system can be accessed over the internet securely, possibly through a VPN or by setting up a secure web service.

3 Communication:
   - Set up communication protocols such as MQTT, HTTP, or WebSocket to send data to your software interface or cloud platform.

- For remote monitoring, consider using cloud platforms like AWS, Azure, Google Cloud, or IoT platforms like Adafruit IO or Ubidots for data storage, processing, and visualization.

4. Power Management:
   - Depending on your monitoring location, you may need to implement power management strategies to conserve energy, especially if you're running on batteries.

5. Security:
   - Ensure that your system is secure, especially if it's connected to the internet. Implement encryption, secure communication, and authentication methods.

6. Maintenance and Alerts:
   - Set up alerts for system maintenance, such as low battery, sensor failure, or network connectivity issues.

7. Data Analysis and Long-term Trends:
   - Use the collected data for long-term analysis, trend identification, and environmental pattern recognition.

CONCLUSION:

In conclusion, our environmental monitoring study in the park revealed important insights into temperature and humidity patterns. We found that temperature fluctuated seasonally, with the highest temperatures recorded during the summer months and the lowest during winter. Humidity levels also exhibited seasonal variations, with higher humidity in the early morning and lower humidity in the afternoon. These findings are crucial for understanding the park's microclimate and can aid in better park management and visitor comfort.