Smart Water System

Phase 3: Development Part 1

In this part you will begin building your project.

Start building the IoT water consumption monitoring system.

Configure IoT sensors (e.g., flow meters) to measure water consumption in public places.

Develop a Python script on the IoT sensors to send real-time water consumption data to the data-sharing platform.

Solution:

Building an IoT water consumption monitoring system involves several steps, including configuring IoT sensors, developing a Python script to gather data, and sending that data to a data-sharing platform. Below, I'll outline the initial steps for setting up the IoT sensors and creating a Python script to collect and transmit water consumption data.

Hardware Setup:

1. **Select IoT Sensors:** Choose appropriate IoT sensors for measuring water consumption. Flow meters or water usage sensors are common choices for this purpose.
2. **Connect Sensors:** Connect the selected sensors to your IoT hardware. Make sure they are properly wired and powered.
3. **Connect to IoT Device:** If your IoT sensors use a specific communication protocol (e.g., Modbus, MQTT, or HTTP), connect them to your IoT device (e.g., Raspberry Pi, Arduino, ESP8266/ESP32). You might need analog/digital pins or communication modules, depending on your sensor's interface.

Software Development:

1. **Install Necessary Libraries:** Ensure that you have the required libraries and dependencies installed on your IoT device. You might need libraries for sensor data processing, communication, and any other specific requirements.

2. **Develop Python Script:**

   a). Start by creating a Python script that reads data from the IoT sensors. Here's a basic outline of what this script might look like:

   <u>PROGRAM:</u>

```
# Import necessary libraries
import time
import sensor_library  # Replace with the actual library for your sensor
import data_sender  # Library for sending data to the data-sharing platform

# Initialize the sensor
sensor = sensor_library.initialize_sensor()  # Replace with your sensor's initialization code

while True:
    # Read data from the sensor
    data = sensor.read_data()  # Replace with your sensor's data retrieval code

    # Process the data if needed
    processed_data = process_data(data)

    # Send data to the data-sharing platform
    data_sender.send_data(processed_data)

    # Wait for a specified interval before the next reading
    time.sleep(60)  # Adjust the interval as per your requirements
```

   b). Replace "sensor_library" and "data_sender" with the actual libraries or functions needed for your specific sensor and data-sharing platform.

3. **Data Processing:** Depending on your requirements, you might need to perform data processing within the script, such as calculating water consumption over time, averaging readings, or filtering outliers.
4. **Data Transmission:** Implement the data_sender module to send data to your data-sharing platform. This could involve using APIs, MQTT, HTTP, or other communication methods.

Testing and Deployment:

1. **Testing:** Test your setup by running the Python script and monitoring the data sent to your data-sharing platform. Ensure that the data is accurate and that the script functions as expected.
2. **Deployment:** Install and securely mount the IoT device with sensors in the public places where you want to monitor water consumption.
3. **Monitoring and Maintenance:** Regularly monitor the system to ensure it's working correctly and perform maintenance as needed.

Remember to consider security, power management, and data backup in your IoT system, especially if it's deployed in public places. Additionally, adapt the script and hardware to the specific requirements of your project and the data-sharing platform you're using.