IOT_Phase3

Development Part 1

Configure IoT sensors to measure water consumption in public places.

1. Sensor Setup:

- Install the flow meters at the appropriate locations in public places where you want to measure water consumption.

2. Data Collection:

- Set up IoT devices to collect data from the flow meters. These devices should be capable of processing and transmitting data to a central server or cloud platform.

3. Data Transmission:

- Implement a communication protocol (e.g., MQTT, HTTP, or CoAP) to transmit data from the sensors to a central server or cloud platform. Ensure data security and encryption for sensitive information.

4. Central Data Processing:

- Use a central server or cloud platform to receive and store the data from all sensors. Popular options include AWS IoT, Azure IoT, Google Cloud IoT, or dedicated IoT platforms like ThingSpeak, Ubidots, or Particle.

5. Data Analysis:

- Process and analyze the data to extract meaningful insights. Calculate water consumption patterns, trends, and other relevant metrics.

6. Data Visualization:

- To visualize the water consumption data, you can use various tools and platforms:
    - **Custom Dashboard**: Create a custom web-based or mobile dashboard using HTML, CSS, and JavaScript. You can use libraries like D3.js or Chart.js to create interactive charts and graphs.
    - **IoT Platforms**: Many IoT platforms offer built-in data visualization tools. For example, ThingSpeak provides a simple charting feature that allows you to visualize your data.
    - **Business Intelligence Tools**: Tools like Tableau, Power BI, or Google Data Studio can connect to your data source and create visualizations and dashboards.
    - **Open-Source Tools**: Tools like Grafana and Kibana are popular for real-time data visualization and monitoring. They can integrate with various data sources, including IoT platforms and databases.

7. Real-Time Updates:

- Ensure that your visualization system can update in real-time as new data arrives from the sensors. This is especially important for monitoring water consumption as it happens.

8. User Access:

- Depending on your use case, provide access to the data and visualizations to relevant stakeholders, such as water utility companies, government agencies, or the public. Implement user authentication and access control as needed.

9. Alerts and Notifications:

- Implement alerting and notification mechanisms within your visualization system to detect abnormal water consumption patterns or sensor malfunctions and alert administrators.

10. Historical Data:

- Store historical data for future analysis and reference. You can archive the data in a database or a cloud storage solution.

11. Scaling and Expansion: Plan for scalability and expansion as you may want to add more sensors or extend your monitoring to additional public places.

PYTHON SCRIPT

```
import time

import random

import requests


# ThingSpeak API endpoint and API key

THINGSPEAK_API_ENDPOINT =
"https://api.thingspeak.com/update"

THINGSPEAK_API_KEY = "YOUR_THINGSPEAK_API_KEY"


# Simulated water consumption function (you should replace
this with your actual sensor data)

def simulate_water_consumption():

    return random.uniform(0.1, 5.0)  # Simulate consumption
between 0.1 and 5.0 liters


while True:

    # Simulate water consumption

    water_consumption = simulate_water_consumption()


    # Create a dictionary with your data fields (field1, field2,
etc.)
```

```
    data = {'api_key': THINGSPEAK_API_KEY, 'field1': water_consumption}

    try:
        # Send a POST request to ThingSpeak
        response = requests.post(THINGSPEAK_API_ENDPOINT, data=data)

        if response.status_code == 200:
            print(f"Data sent successfully: {water_consumption} liters")
        else:
            print(f"Failed to send data. Status code: {response.status_code}")

    except Exception as e:
        print(f"An error occurred: {str(e)}")

    # Set the update interval (in seconds)
    time.sleep(300)  # Update every 5 minutes (adjust as needed)
```