SMART WATER MANAGEMENT

USING IOT

S.GOWRISHANKAR
au812921106014
gowripubg1718@gmail.com

DEVELOPMENT PART 2 :

Introduction:

Smart water management involves the intelligent planning, monitoring, and optimization of water-related processes to ensure the efficient and sustainable use of this precious resource. It plays a pivotal role in environmental conservation, addressing water scarcity issues, and enhancing overall water quality. The integration of technology, data analytics, and proactive strategies is key to achieving smart water management goal
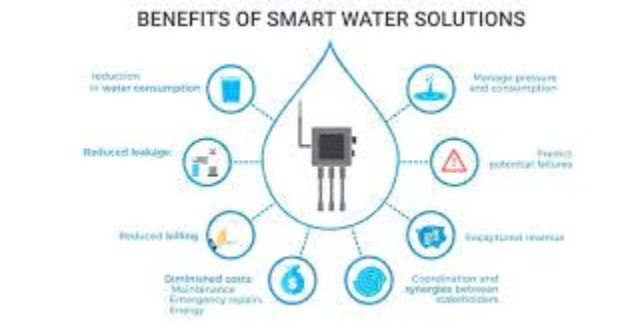
BLOCK   DIAGRAM :
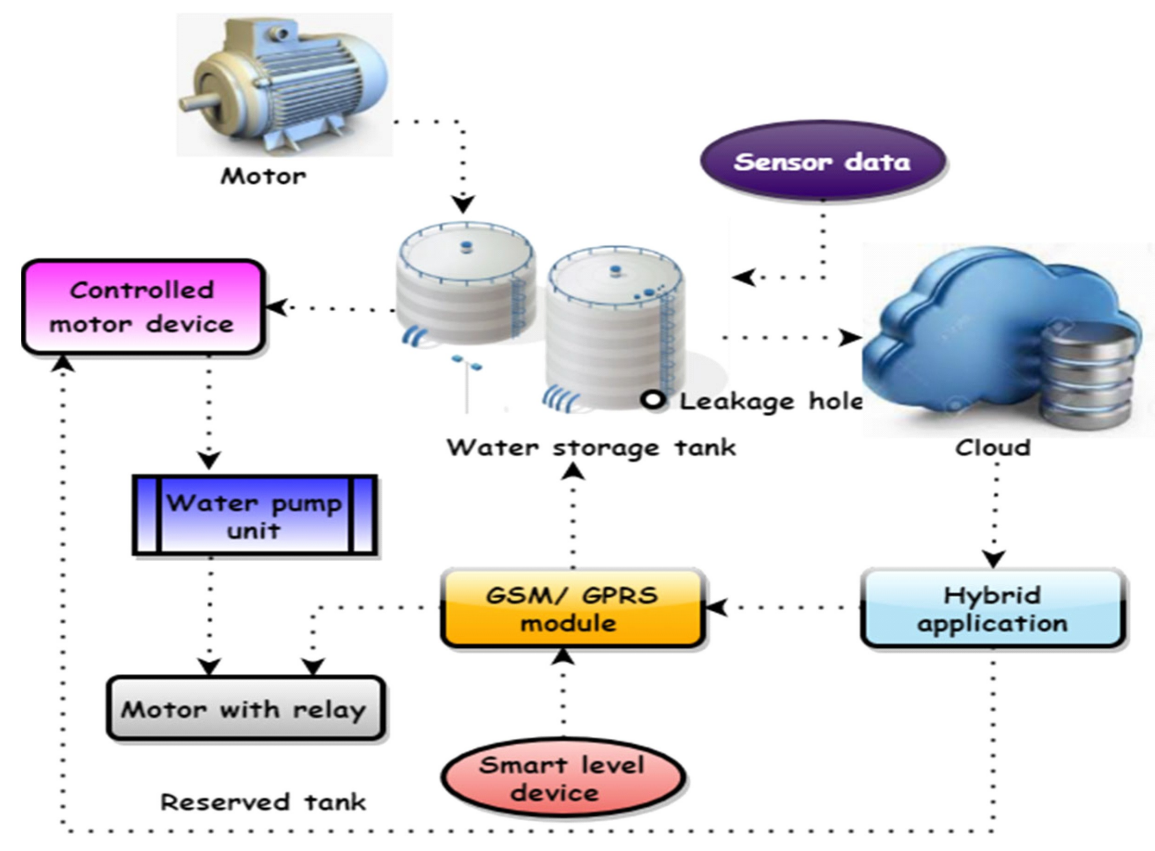
Defining Project Requirements:

Begin by clearly defining the objectives of your Smart Water

Management project. What specific data are you looking to collect and analyze? What problems are you trying to solve? Understanding your project's goals is essential


IoT Device Deployment:

Choose the appropriate IoT devices for your project. These devices should be capable of collecting, processing, and transmitting data to a central server or cloud platform.

Data Analysis and Visualization:

Set up a data analysis platform that can receive data from your IoT devices and perform relevant analyses. Tools like Python libraries (e.g., Pandas, Matplotlib, Seaborn) can be used for data analysis and visualization.
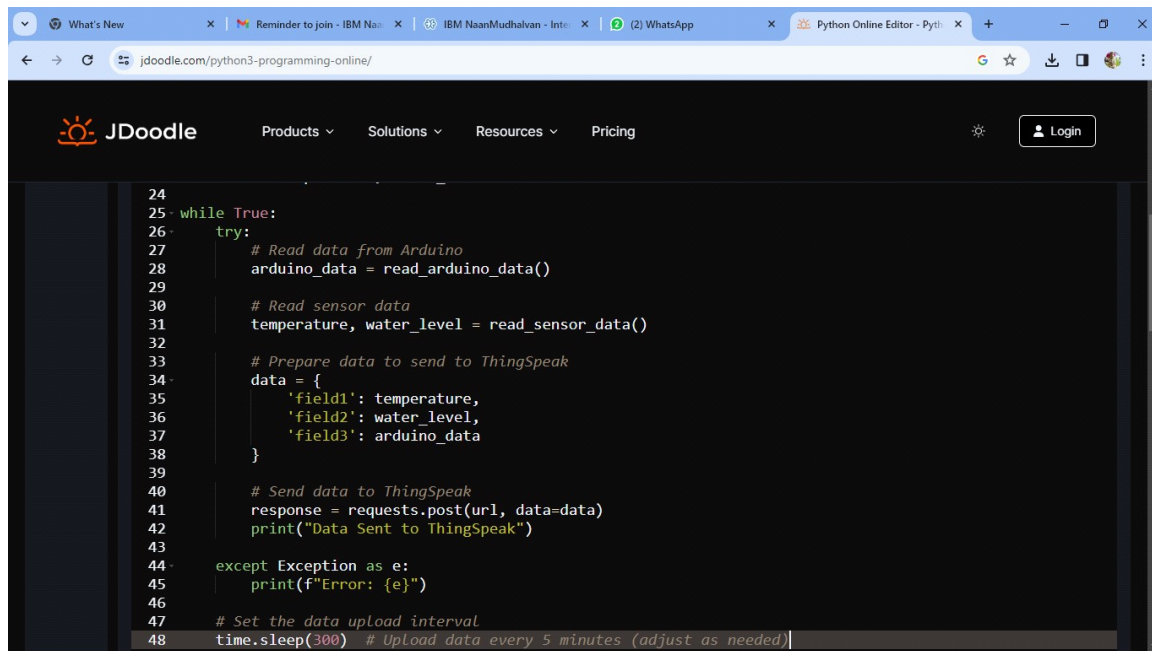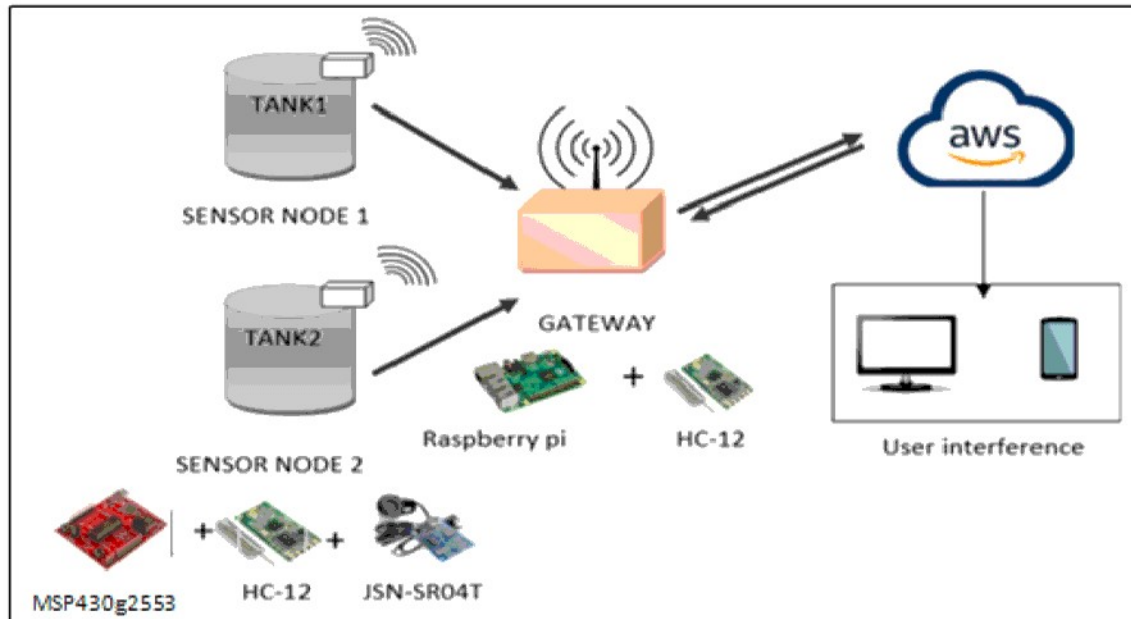

Remote Monitoring and Alerts:

Implement remote monitoring and alerting systems to notify relevant stakeholders when specific water parameters go beyond predefined thresholds. This can be done through emails, SMS, or mobile apps.
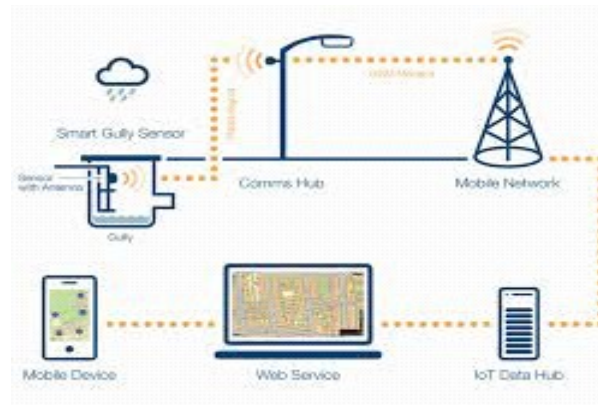
PYTHON PROGRAM :


```python
import serial
import time
import requests
import random

# Define your ThingSpeak API key and channel URL
api_key = "G1RSE98QHVUFT626"
url = f"https://api.thingspeak.com/update?api_key={api_key}"

# Initialize the serial connection to Arduino
arduino_port = '/dev/ttyACM0'   # Update with your Arduino's port
ser = serial.Serial(arduino_port, 9600)

# Function to read data from Arduino
def read_arduino_data():
    data = ser.readline().decode().strip()
    return data

# Simulated sensor data (replace with actual sensor readings)
def read_sensor_data():
    temperature = random.uniform(20.0, 30.0)
    water_level = random.uniform(0, 100)
    return temperature, water_level
```

```
24
25  while True:
26      try:
27          # Read data from Arduino
28          arduino_data = read_arduino_data()
29
30          # Read sensor data
31          temperature, water_level = read_sensor_data()
32
33          # Prepare data to send to ThingSpeak
34          data = {
35              'field1': temperature,
36              'field2': water_level,
37              'field3': arduino_data
38          }
39
40          # Send data to ThingSpeak
41          response = requests.post(url, data=data)
42          print("Data Sent to ThingSpeak")
43
44      except Exception as e:
45          print(f"Error: {e}")
46
47      # Set the data upload interval
48      time.sleep(300)  # Upload data every 5 minutes (adjust as needed)
```

OUTPUT :

Dear user, need your attention here. The water tanks will be filled in 5450 seconds

📢📢 You can turn off the water motor by the following ways:
👉 Direct API Request:
https://bit.ly/3ghdLbl
👉 Cloud Dashboard:
https://bit.ly/3AWkBLI
👉 Dedicated Web Application:
https://wheels4water.me

You can use any of the methods as per your convenience and in case you forgot to switch it off, it will be done on my own.

Save Water & i 🙏🙏 2:36 PM