Developing an IoT smart water system involves integrating sensors, communication devices, and a central platform. Start by selecting water quality and quantity sensors compatible with IoT. Connect these sensors to microcontrollers or single-board computers like Arduino or Raspberry Pi. Implement communication protocols such as MQTT or HTTP for data transmission.

Set up a cloud platform to collect, store, and analyze sensor data. Use services like AWS IoT, Azure IoT, or Google Cloud IoT. Develop a user interface, either a web or mobile app, to visualize the water-related data for users.

Incorporate actuators for remote control, such as valves or pumps, to manage water flow based on the system's analysis. Implement security measures to protect data and devices from potential threats.

Regularly update and maintain the system to ensure its effectiveness and security. Consider scalability and flexibility in the design to accommodate future enhancements or additions to the system.

PROGRAM :

import Adafruit\_DHT

import RPi.GPIO as GPIO

import requests

import json

import time

# Pin configuration for the DHT sensor and water level sensor

DHT\_PIN = 4

WATER\_LEVEL\_PIN = 17

GPIO.setmode(GPIO.BCM)

GPIO.setup(WATER\_LEVEL\_PIN, GPIO.IN)

# Replace 'YOUR\_CLOUD\_API\_ENDPOINT' with the actual endpoint provided by your cloud platform

CLOUD\_API\_ENDPOINT = 'https://your-cloud-api-endpoint/data'

while True:

humidity, temperature = Adafruit\_DHT.read\_retry(Adafruit\_DHT.DHT22, DHT\_PIN)

water\_level = GPIO.input(WATER\_LEVEL\_PIN)

data = {

'temperature': temperature,

'humidity': humidity,

'water\_level': water\_level

}

headers = {'Content-Type': 'application/json'}

try:

response = requests.post(CLOUD\_API\_ENDPOINT, data=json.dumps(data), headers=headers)

print('Data sent to cloud. Response:', response.status\_code)

except Exception as e:

print('Error sending data:', e)

time.sleep(60) # Adjust the time interval based on your requirements