ALAGAPPA CHETTIAR GOVERNMENT COLLEGE OF ENGINEERING AND TECHNOLOGY
KARAIKUDI – 630 003

SMART WATER FOUNTAIN

Objective:

The objective of this IoT water fountain system is to provide an efficient, reliable, and user-friendly way to access and monitor public water fountains while conserving water resources and ensuring they are in working condition.

Development part – 01 :

Creating a smart water fountain using Python typically involves integrating various hardwarecomponents and sensors for control and automation. Here's a basic example of Python code for a smart water fountain using a Raspberry Pi and some additional hardware components:

Hardware Components:

- Raspberry Pi (or any other microcontroller)
- Water pump
- Water level sensor (e.g., ultrasonic sensor or float switch)
- Relay module for controlling the pump
- RGB LED (optional for lighting effects)
- Tubing and fountain head
- Power source for the pump and LED (if used)

Python Code:

1. First, you'll need to install the necessary libraries. You can use the `RPi.GPIO` libraryfor Raspberry Pi GPIO control.

```
bash
pip install RPi.GPIO
```

2. Here's a basic Python script for a smart water fountain:

```
python
import RPi.GPIO as GPIOimport
time

# Define GPIO pin numbers
water_level_pin = 17 # GPIO pin for the water level sensor relay_pin = 18  #
GPIO pin for the relay module (pump control)
led_red_pin = 22 # GPIO pin for the red channel of an RGB LED (optional)led_green_pin
= 23 # GPIO pin for the green channel of an RGB LED (optional)
led_blue_pin = 24  # GPIO pin for the blue channel of an RGB LED (optional)

# Set  up  GPIO GPIO.setmode(GPIO.BCM)
GPIO.setup(water_level_pin, GPIO.IN)
GPIO.setup(relay_pin, GPIO.OUT)
GPIO.setup(led_red_pin, GPIO.OUT)
GPIO.setup(led_green_pin, GPIO.OUT)
GPIO.setup(led_blue_pin, GPIO.OUT)

# Function to check water leveldef
is_water_low():
    return GPIO.input(water_level_pin) == GPIO.LOW
```

```
# Function to control the pumpdef
control_pump(state):
    GPIO.output(relay_pin, state)

# Function to control the RGB LED (optional)
def control_led(red, green, blue): GPIO.output(led_red_pin, red)
GPIO.output(led_green_pin, green)GPIO.output(led_blue_pin, blue)

try:
          while True:
              if is_water_low():
                  print("Water level is low. Turning on the pump.")
                  control_pump(GPIO.HIGH) # Turn on the pump control_led(0, 1,
                  0)  # Optional: Set the LED to green
              else:
                  print("Water level is sufficient. Turning off the pump.")
                  control_pump(GPIO.LOW)  # Turn off the pump control_led(0, 0,
                  1)  # Optional: Set the LED to blue
              time.sleep(5)  # Check the water level every 5 seconds

except KeyboardInterrupt:
    GPIO.cleanup()
```

Make sure to connect the hardware components to the specified GPIO pins on your Raspberry Pi. This code checks the water level and controls the pump accordingly. You canexpand this project by adding more features, such as remote control via a web interface or integrating other sensors and actuators for advanced functionality.