**Smart water fountain**

**INTRODUCTION:**

Welcome to the future of hydration – Smart Water Fountains. These cutting-edge marvels seamlessly blend technology with the essential need for clean, accessible water. Designed to revolutionize the way we quench our thirst, these fountains go beyond the conventional, offering a fusion of innovation and sustainability. Get ready to experience a refreshing leap into the world of intelligent hydration with Smart Water Fountains.



**Smart Water Fountain Python**

Creating a smart water fountain using Python would typically involve a combination of hardware and software components. Here's a simplified example using a Raspberry Pi and some basic Python code:

**Hardware Components:**

Raspberry Pi (or similar microcontroller)

Water pump

Water level sensor

Relay module

Tubing and fountain structure

Power supply

**Software Components:**

Raspbian OS (or any other suitable OS for Raspberry Pi)

Python script for controlling the fountain

Python Code:

python

Copy code

import RPi.GPIO as GPIO

import time

# GPIO pins

PUMP\_PIN = 17

WATER\_SENSOR\_PIN = 18

# Set up GPIO

GPIO.setmode(GPIO.BCM)

GPIO.setup(PUMP\_PIN, GPIO.OUT)

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

def turn\_on\_pump():

GPIO.output(PUMP\_PIN, GPIO.HIGH)

def turn\_off\_pump():

GPIO.output(PUMP\_PIN, GPIO.LOW)

def is\_water\_level\_low():

return GPIO.input(WATER\_SENSOR\_PIN) == GPIO.LOW

try:

while True:

if is\_water\_level\_low():

turn\_on\_pump()

print("Water level low - Pumping water.")

else:

turn\_off\_pump()

print("Water level normal - Pump turned off.")

time.sleep(5) # Check water level every 5 seconds

except KeyboardInterrupt:

print("Program terminated by user.")

GPIO.cleanup()

**Explanation:**

The code uses the GPIO library to control the Raspberry Pi's GPIO pins.



The water pump is connected to a relay module, controlled by the PUMP\_PIN.

The water level sensor is connected to the WATER\_SENSOR\_PIN.

The main loop continuously checks the water level, and if it's low, it turns on the pump.

**PROGRAM:**

Water quality:

Import plotly.express as px

Data = data

Figure = px.histogram(data, x = “ph”,

Color = “Potability”,

Title= “Factors Affecting

Water Quality: PH”)

Figure.show()

Temperature:

Import glob

Import time

RATE = 30

Sensor\_dirs =

glob.glob(“/sys/bus/w1/devices/28\*”)

If len(sensor\_dirs) != 0:

While True:

Time.sleep(RATE)

For directories in sensor\_dirs:

Temperature\_file =

open(directories + “/w1\_slave”)

# Reading the files

Text = temperature\_file.read()

Temperature\_file.close()

# Split the text with new lines (\n)

and select the second line.

Second\_line = text.split(“\n”)[1]

# Split the line into words, and

select the 10th word

Temperature\_data =

second\_line.split(“ “)[9]

# We will read after ignoring first

two character.

Temperature =

float(temperature\_data[2:])

# Now normalise the temperature

by dividing 1000.

Temperature = temperature /

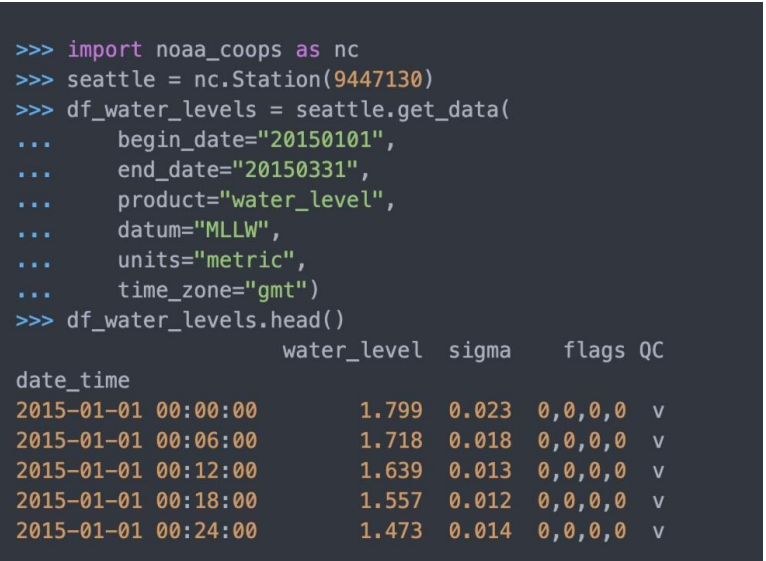
1000

Print ‘Address :

‘+str(directories.split(‘/’)[-1])+’,

Temperature

Water level:



**CONCULUSION:**

In conclusion, smart water fountains represent a promising intersection of technology and sustainability. With features like sensor-based usage tracking, efficient water consumption, and connectivity for data analysis, these fountains not only offer convenience but also contribute to conservation efforts. The integration of smart technologies ensures a more mindful and resource-efficient approach to water consumption in public spaces, aligning with the growing emphasis on environmental responsibility. As technology continues to advance, smart water fountains have the potential to play a significant role in promoting sustainable practices and enhancing user experiences.