Name	**Nivetha.S**
Reg. No.	**621421106027**
Department	**ECE**
Year	**III**
College Name	**Maha Barathi Engineering College**
Group	**IOT – Smart water fountains**

PHASE - 3

INTRODUCTION

An Internet of Things (IoT)-enabled in the ultimate tabletop fountain that combines convenience, versatility, and style. Designed to sit on any table, this smart water fountain connects seamlessly to a refillable water jug, eliminating the need for constant refills and ensuring uninterrupted enjoyment. With its rechargeable feature and the ability to fill different cup sizes, this fountain offers a truly customizable experience for every user.

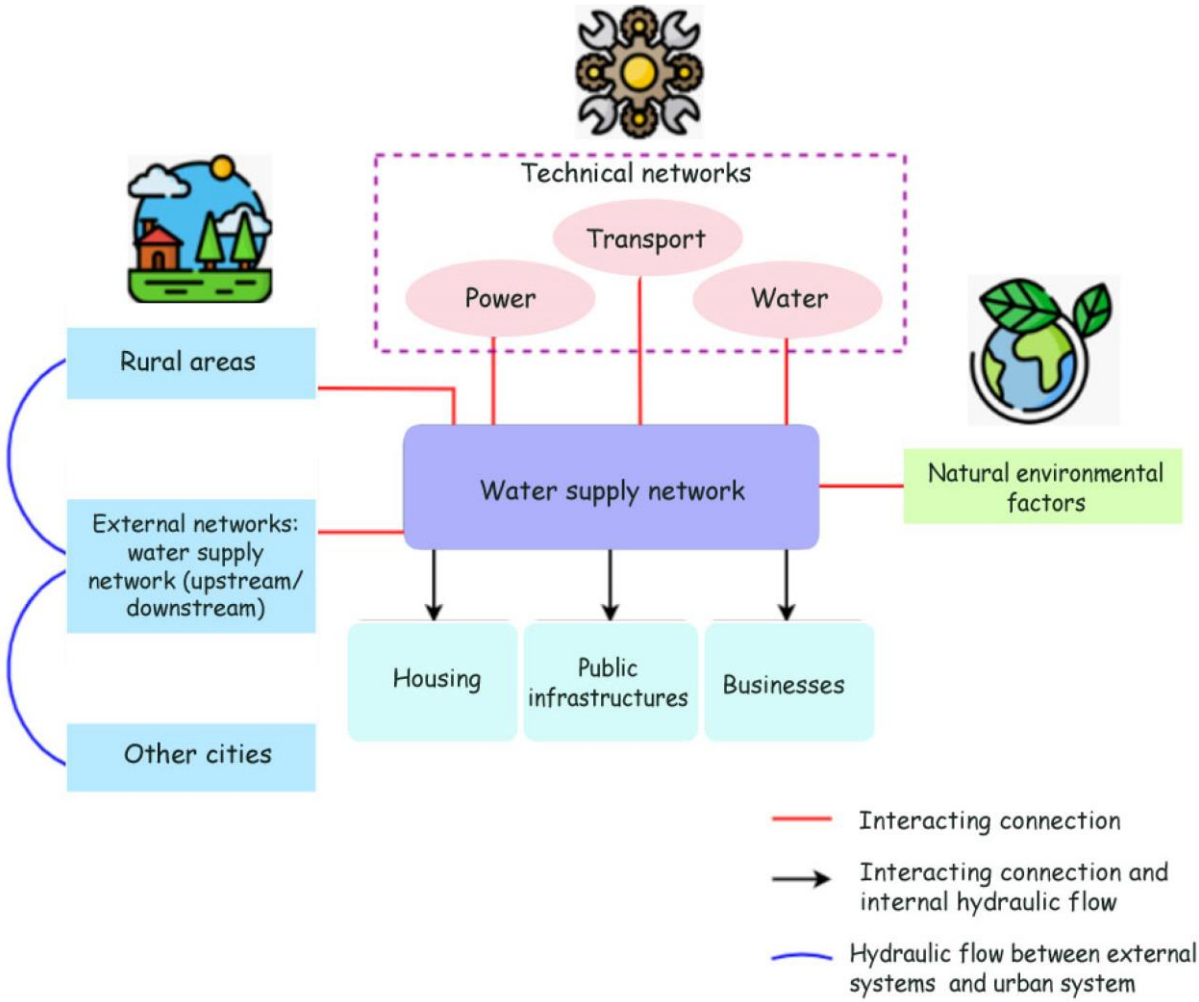
Easy Setup: Connect the smart water fountain to a refillable water jug effortlessly, without the need for complicated plumbing or constant monitoring. Enjoy a continuous flow of refreshing water, making it perfect for tabletops, desks, or even outdoor gatherings.

Functioning of smart water Monitoring System Using IoT Capabilities

The smart water applications include water pipeline monitoring, water quality in open water source, smart water meter reading, IoT security for smart water system(SWS) etc.

In order to ensure the integrity of customer information,secy of the devices and data passed through the network, a framework and methodology were developed [37]. Water quality monitoring of open water sources was another application that was implemented using

IoT devices.



The observation that a fountain of water is created at the leaking joint of pipes of the main water supply line that the pressure exerted on the small hole of the pipe of main water supply that makes the water move out of the pipe through the pressure and hence form a fountain.

Python program in smart water fountains:

Creating a smart water fountain program typically involves combining hardware and software components. Here's a simplified Python program outline for a smart water fountain using a Raspberry Pi and a water pump. This program will turn on the water pump when triggered by a sensor (e.g., a motion sensor or a proximity sensor). You'll need to adapt and extend this basic code to your specific setup and requirements:

```python
import RPi.GPIO as GPIO
import time

# GPIO pin for controlling the water pump
water_pump_pin = 17

# Initialize GPIO
```

```
GPIO.setmode(GPIO.BCM)
GPIO.setup(water_pump_pin, GPIO.OUT)

# Function to turn on the water pump
def turn_on_water_pump():
    GPIO.output(water_pump_pin,
GPIO.HIGH)
    print("Water pump is ON")

# Function to turn off the water pump
def turn_off_water_pump():
    GPIO.output(water_pump_pin,
GPIO.LOW)
    print("Water pump is OFF")

try:
    while True:
        # You can replace this with your own
sensor input logic
        # For example, using a PIR motion
sensor
        # if motion_detected():
```

```
        #     turn_on_water_pump()
        # else:
        #     turn_off_water_pump()

        # Simulate the presence of motion
        motion_detected = True

        if motion_detected:
            turn_on_water_pump()
            # Run the pump for a specified
duration
            time.sleep(10)  # Adjust this
duration as needed
            turn_off_water_pump()

except KeyboardInterrupt:
    GPIO.cleanup()
```

In this example, you'll need to replace the `motion_detected` logic with your actual

sensor input. You might also want to integrate other features like water level monitoring, scheduling, and remote control based on your project's requirements.

Additionally, remember to install the required libraries and set up your Raspberry Pi with the necessary hardware components (e.g., a relay module to control the water pump).