

**THEME : SMART WATER FOUNTAINS  
PHASE III  
DEVELOPMENT PART 1**

SUBMITTED BY: P.NAVEEN

G.THAMARAISELVAN  
M.PERUMAL  
B.K.KARTHIKEYAN

S.no	CONTENT	Pg.no
1.	INTRODUCTION	3
2.	DRAWBACKS OF WATER FOUNTAIN	4
3.	HARDWARE COMPONENTS	5
4.	SOFTWARE COMPONENTS	6
5.	STEPS TO CREATE THE SMART WATER FOUNTAIN	7
6.	CODING	8
7.	DESIGN	9
8.	DEVELOPMENT	10
9.	ACKNOWLEDGEMENT	11

# Introduction

- Water is an essential part in our day to day life.
- smart water fountain is an automated system that provides a continuous flow of water in a controlled manner. It incorporates various technologies and features to enhance functionality, efficiency, and convenience.

# Drawbacks of water fountains

Waste of water resources

Man power is need to operate

Difficult to find errors

Different style

Purpose of fountain to flow

# Hardware Components

1. Raspberry Pi 3 Model B or later
2. Water pump
3. Relay module
4. Water level sensor (optional)
5. Power supply for the pump
6. Tubing and fountain nozzle
7. Waterproof container for the water reservoir
8. Various cables, connectors, and a breadboard

# Software Components

1. Raspbian OS (or a suitable Raspberry Pi OS)
2. Python for programming
3. IoT platform (e.g., MQTT, AWS IoT, or Google Cloud IoT Core)
4. Libraries for GPIO control (e.g., RPi.GPIO)
5. Optional: Web server and HTML/CSS/JavaScript for a web-based user interface

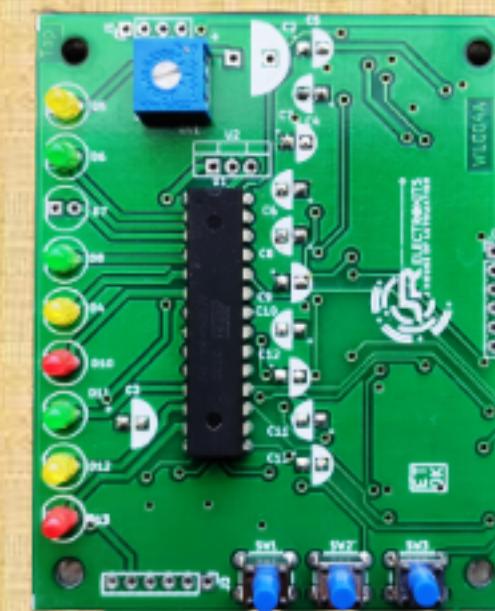
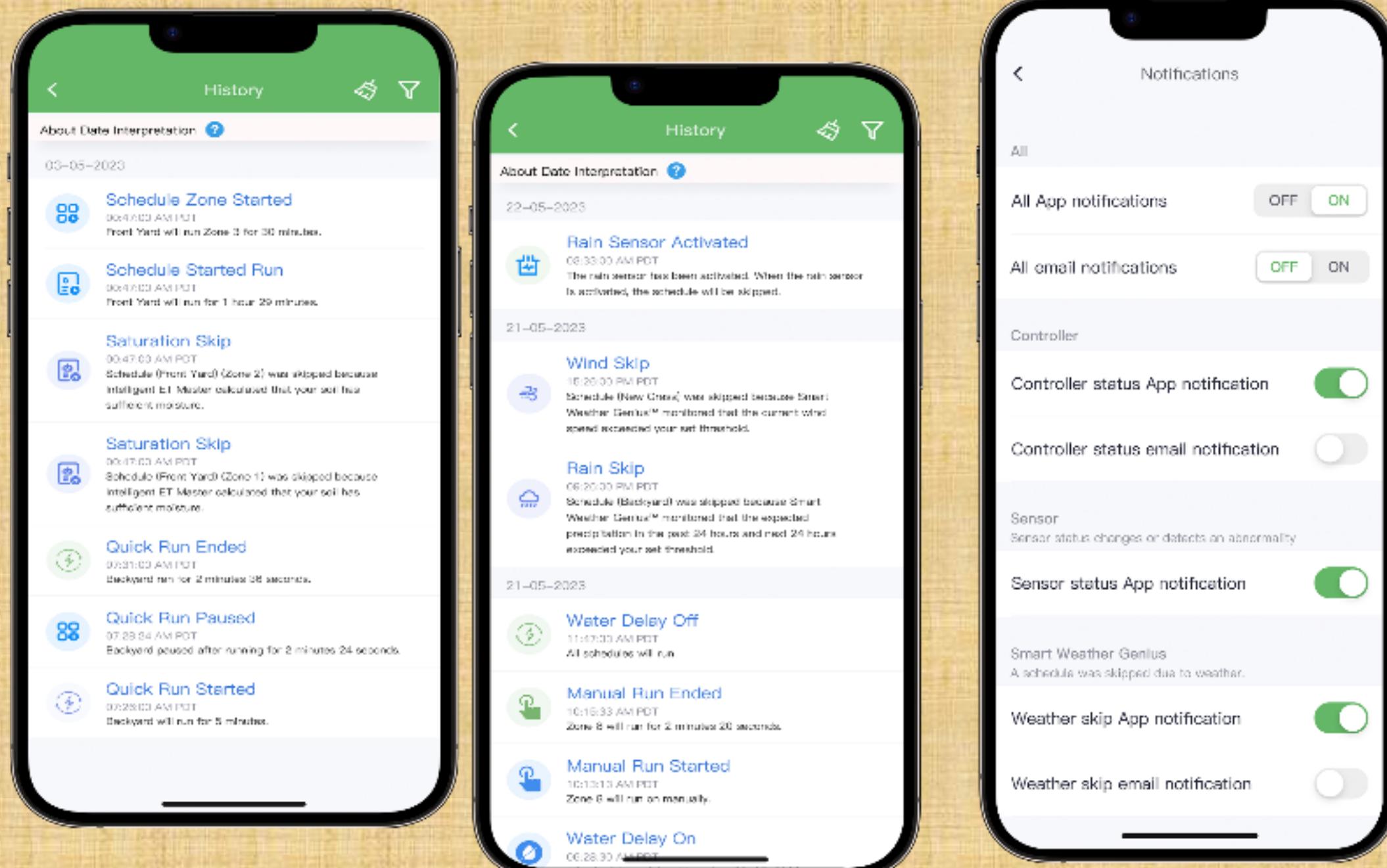
# Steps to Create the Smart Water Fountain

- Set Up Raspberry Pi
- Hardware Setup
- Install Required Libraries
- Code the Fountain Control
- IoT Integration
- Web Interface
- Assemble and Test
- Finalize and Deploy

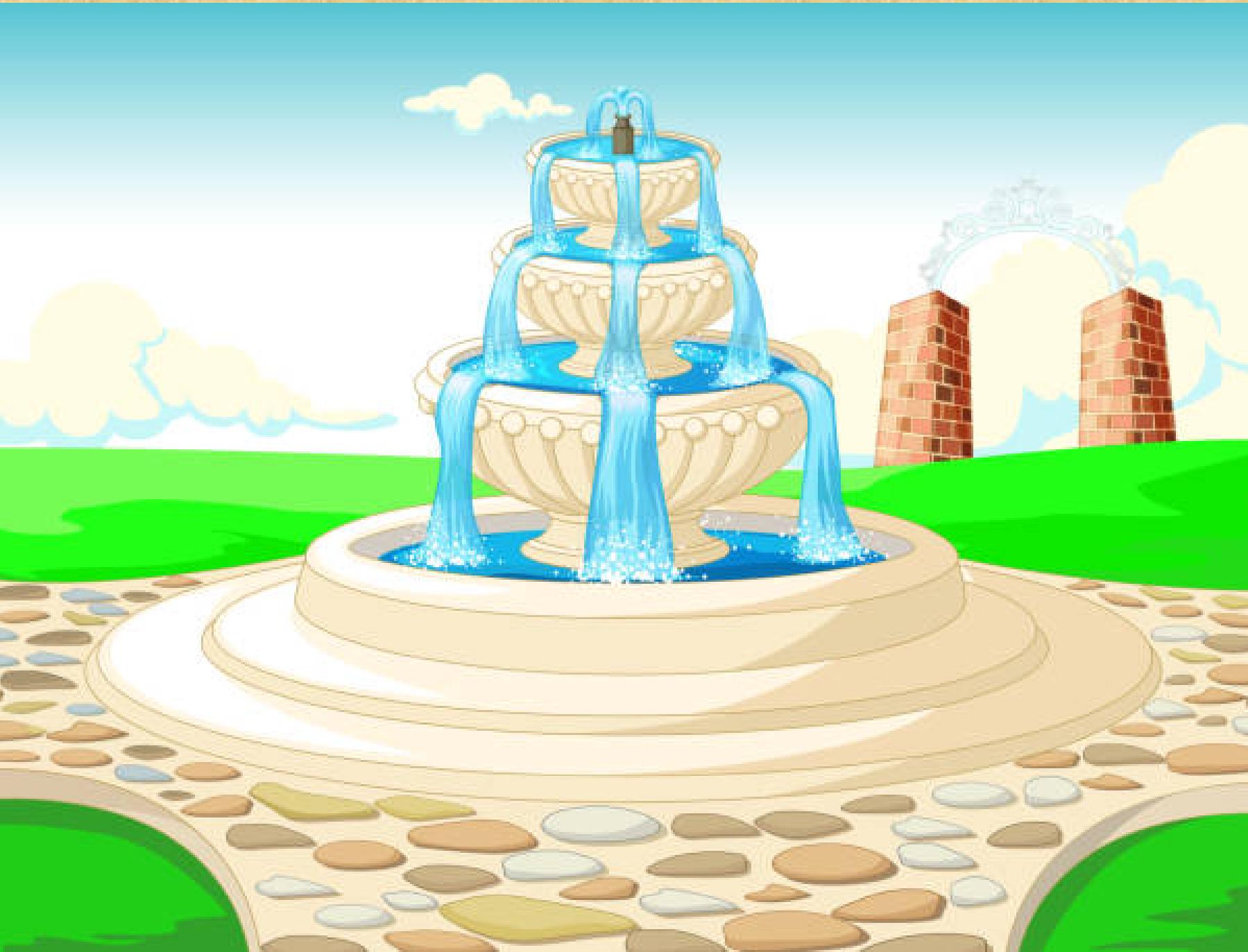
# coding

```
#import RPi.GPIO as GPIO
    import time
# Set the GPIO mode and pin numbers
    GPIO.setmode(GPIO.BCM)
    water_pump_pin = 17
    status_led_pin = 18
# Setup GPIO pins
    GPIO.setup(water_pump_pin, GPIO.OUT)
    GPIO.setup(status_led_pin, GPIO.OUT)
    def water_fountain_on():
        print("Water fountain turned on")
        GPIO.output(water_pump_pin, GPIO.HIGH)
        GPIO.output(status_led_pin, GPIO.HIGH)
    def water_fountain_off():
        print("Water fountain turned off")
        GPIO.output(water_pump_pin, GPIO.LOW)
        GPIO.output(status_led_pin, GPIO.LOW)
# Main loop
try:
    while True:
# Check if the water level is low
        if is_water_level_low():
            water_fountain_on()
else:
            water_fountain_off()
# Delay for 1 second
        time.sleep(1)
        except KeyboardInterrupt:
# Clean up GPIO on keyboard interrupt
        GPIO.cleanup()
```

# Design



# Development



# Thank you

- Our acknowledgement to our mentors, naan mudhalvan team, IBM team, and also our management.