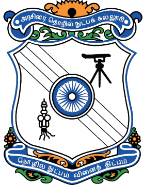


****

**GOVERNMENT COLLEGE OF TECHNOLOGY COIMBATORE-641013**

**DEPARTMENT OF ECE**

**SMART WATER FOUNTAINS**

**PHASE III**

**TEAM MEMBERS**

1. SATHIYA PRAKASH B L -71772114138

2. KRISHNA PRAVEEN S -71772114302

3. MONISHA V - 71772114304

4. SUDHARANI K – 71772114L02

**SMART WATER FOUNTAINS**

**PHASE III: DEVELOPMENT PART-I**

**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
* Manpower needs
* 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**

