**Safety and Energy Efficiency**

Discuss the implementation of safety features, like motion sensors and automatic shut-off.Highlight the importance of energy-efficient components and scheduling features.

**Testing and Quality Assurance**

Emphasize the need for thorough testing to ensure reliability and performance.Mention quality control procedures.

**Safety and Energy Efficiency**

Discuss the implementation of safety features, like motion sensors and automatic shut-off.Highlight the importance of energy-efficient components and scheduling features.

**Connectivity**

Explain how smart water fountains can connect to home automation systems or the internet. Mention remote control options and real-time status updates.

**Automation and Control**

Highlight the importance of automation for convenience and user experience.Discuss features like timers and motion sensors.

python

import RPi.GPIO as GPIO

import time

# GPIO pins for water pump and LED lights

pump\_pin = 17

light\_pin = 18

# Initialize GPIO

GPIO.setmode(GPIO.BCM)

GPIO.setup(pump\_pin, GPIO.OUT)

GPIO.setup(light\_pin, GPIO.OUT)

# Function to control the water pump

def control\_pump(state):

if state == "on":

GPIO.output(pump\_pin, GPIO.HIGH)

else:

GPIO.output(pump\_pin, GPIO.LOW)

# Function to control LED lights

def control\_lights(state):

if state == "on":

GPIO.output(light\_pin, GPIO.HIGH)

else:

GPIO.output(light\_pin, GPIO.LOW)

# Main program

try:

while True:

# Read sensor data and make decisions based on it.

# For example, turn on the water pump and lights if water level is low.

control\_pump("on")

control\_lights("on")

time.sleep(5) # Run the fountain for 5 seconds

control\_pump("off")

control\_lights("off")

time.sleep(60) # Wait for a minute before running again

except KeyboardInterrupt:

GPIO.cleanup() # Cleanup GPIO on program exit