Smart Public Restroom

Phase 3

Creating a smart public restroom using IoT technology involves integrating various sensors and devices to enhance user experience and optimize restroom management. Here’s a list of materials you might need and a basic outline for an IoT-based Python script:

\*\*Materials:\*\*

1. \*\*Motion Sensors:\*\* To detect the presence of users.

2. \*\*Ultrasonic Sensors:\*\* To measure the distance of objects, useful for occupancy detection.

3. \*\*RFID/NFC Readers:\*\* For secure access control.

4. \*\*Smart Locks:\*\* To secure the restroom when it’s occupied.

5. \*\*Temperature and Humidity Sensors:\*\* To monitor the restroom environment.

6. \*\*Water Flow Sensors:\*\* To monitor water usage.

7. \*\*IoT Development Board (e.g., Raspberry Pi, Arduino):\*\* To connect and control sensors.

8. \*\*Relays:\*\* To control devices such as water supply and lights.

9. \*\*Internet Connectivity:\*\* Wi-Fi module or Ethernet shield.

10. \*\*Cloud Platform:\*\* Like AWS IoT, Azure IoT, or Google Cloud IoT for data storage and analysis.

\*\*Python Script:\*\*

Here’s a basic outline for your Python script to handle the IoT functionalities:

```python

# Import necessary libraries

Import RPi.GPIO as GPIO ln pHImport time

Import requests

# Set up GPIO pins for sensors and actuators

GPIO.setmode(GPIO.BOARD)

Motion\_sensor\_pin = 11

Ultrasonic\_trig\_pin = 13

Ultrasonic\_echo\_pin = 15

Lock\_pin = 19

# … Other sensor pins …

# Initialize sensors and actuators

GPIO.setup(motion\_sensor\_pin, GPIO.IN)

GPIO.setup(ultrasonic\_trig\_pin, GPIO.OUT)

GPIO.setup(ultrasonic\_echo\_pin, GPIO.IN)

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

# … Initialize other sensors and actuators …

# Main function to monitor restroom status

Def monitor\_restroom():

While True:

# Motion detection

If GPIO.input(motion\_sensor\_pin):

Print(“Motion detected!”)

# Check occupancy using ultrasonic sensor

If check\_occupancy():

Unlock\_restroom()

# Send occupancy data to the cloud platform

Send\_data\_to\_cloud(“Occupied”)

Else:

Print(“Restroom is occupied. Please wait.”)

Send\_data\_to\_cloud(“Occupied”)

Else:

Lock\_restroom()

Print(“No motion detected. Restroom is vacant.”)

Send\_data\_to\_cloud(“Vacant”)

Time.sleep(2)

# Function to check restroom occupancy using ultrasonic sensor

Def check\_occupancy():

# … Code to measure distance using ultrasonic sensor and determine occupancy …

Pass

# Function to unlock the restroom

Def unlock\_restroom():

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

Print(“Restroom unlocked.”)

# Function to lock the restroom

Def lock\_restroom():

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

Print(“Restroom locked.”)

# Function to send data to the cloud platform

Def send\_data\_to\_cloud(status):

# … Code to send restroom status data (occupied/vacant) to the cloud platform …

Pass

# Run the monitoring function

If \_\_name\_\_ == “\_\_main\_\_”:

Try:

Monitor\_restroom()

Except KeyboardInterrupt:

Print(“Monitoring stopped.”)

GPIO.cleanup()

```

Please note that this is a basic outline, and you’ll need to implement the `check\_occupancy()` function based on your ultrasonic sensor data. Also, you’ll need to integrate RFID/NFC readers and other sensors according to your project requirements. Additionally, set up your cloud platform to receive and store the restroom status data sent by the device.

By. SASIDHARAN R