**SMART PUBLIC RESTROOM:**

* **Process and data:**

1. Hardware Setup:
   * Install sensors like occupancy sensors, motion sensors, temperature sensors, humidity sensors, etc., in the restroom.
   * Connect these sensors to an Arduino Uno board.
2. Arduino Programming:
   * Write an Arduino sketch to read data from the sensors.
   * Process and format the sensor data.
   * Use a serial connection to send the data to a connected computer.
3. Python Script:
   * Develop a Python script to run on a computer (or a Raspberry Pi) connected to the Arduino Uno.
   * Configure the script to read data from the Arduino Uno via a serial connection.
4. ThingSpeak Integration:
   * Create a ThingSpeak channel to receive and store the sensor data.
   * Obtain an API key for your ThingSpeak channel.
5. \*\*Python Script for ThingSpeak:\*\*
   * Modify the Python script to format the sensor data
   * Send an HTTP POST request to ThingSpeak with the formatted data, using the ThingSpeak API key.
6. ThingSpeak Data Storage:
   * ThingSpeak will store the data sent by your Python script.
7. Data Analysis and Visualization:
   * ThingSpeak provides built-in tools for data visualization and analysis.
   * You can create charts, graphs, and triggers based on the data to monitor restroom usage and conditions.
8. Alerts and Notifications (Optional):
   * Configure ThingSpeak to send alerts or notifications when certain conditions are met, like low soap levels or high restroom occupancy.
9. Document Creation:
   * Document your project, including hardware setup, Arduino code, Python script, and ThingSpeak configuration.
   * Explain how the system works, the sensors used, and the benefits of having a Smart Public Restroom.
   * Share this document for assessment as mentioned in your original request.
10. Maintenance and Monitoring:
    * Regularly monitor the system and perform maintenance on sensors and hardware as needed.
    * Review and analyze data to make improvements in restroom management.

* **Used Sensors:**

1. Occupancy Sensors: These can detect if someone is inside the restroom and help manage lighting and ventilation based on occupancy.
2. Motion Sensors: Useful for detecting movement, ensuring lights and water fixtures are activated when someone enters.
3. Ultrasonic Sensors: They can measure water levels in toilets and urinals, helping to monitor usage and maintenance needs.
4. Temperature and Humidity Sensors: These sensors help control the climate within the restroom for user comfort.
5. CO2 Sensors: To monitor air quality and trigger ventilation systems when needed for odor and health reasons.
6. Door Sensors: Indicate when restroom doors are opened or closed, useful for occupancy tracking.
7. Water Quality Sensors: To monitor the quality of water in sinks and toilets, ensuring cleanliness and detecting issues.
8. Toilet Paper Dispenser Sensors: To monitor and report on the availability of essential supplies.
9. Soap Dispenser Sensors: To keep track of soap levels and refill requirements.
10. Hand Dryer Sensors: To monitor usage and maintenance needs for hand dryers.
11. Waste Bin Sensors: Indicate when the trash bins need emptying.

* **Python script for smart public restroom:**

Import serial

Import requests

Import time

# ThingSpeak settings

Api\_key = ‘NZTQ0TSB4KARPAZZ’

Write\_url = f’https://api.thingspeak.com/update?api\_key={api\_key}’

# Arduino serial connection

Arduino\_port = ‘COM3’

# Adjust this to your Arduino’s port

Baud\_rate = 9600

Ser = serial.Serial(arduino\_port, baud\_rate)

While True:

Try:

# Read data from Arduino

Arduino\_data = ser.readline().decode().strip()

Sensor\_data = arduino\_data.split(‘,’)

# Extract sensor values (modify as per your Arduino code)

Occupancy = sensor\_data[0]

Temperature = sensor\_data[1]

Humidity = sensor\_data[2]

# Send data to ThingSpeak

Data = {

‘field1’: occupancy,

‘field2’: temperature,

‘field3’: humidity,

}

Response = requests.post(write\_url, data)

If response.status\_code == 200:

Print(‘Data sent to ThingSpeak successfully.’)

Else:

Print(‘Failed to send data to ThingSpeak.’)

Time.sleep(15) # Send data every 15 seconds (adjust as needed)

Except Exception as e:

Print(f’Error: {str€}’)

Ser.close()