SMART PUBLIC RESTROOM

To deploy an IoT-enabled Smart Public Restrooms system, you'll need to set up IoT sensors, collect data, and send it to a central platform for monitoring and analysis. Here's a high-level guide on how to get started with this project:

Hardware and Sensors Setup

1. Occupancy Sensor: For occupancy detection, you can use PIR (Passive Infrared) or ultrasonic sensors, or even a combination of both. Place these sensors near the restroom entrance to detect if someone is inside.

2. Cleanliness Sensor: You can use a simple button or a more sophisticated sensor that can detect cleanliness. For example, a pressure or weight sensor can be placed on the restroom seat to monitor cleanliness. When pressed, it can indicate that the seat needs cleaning.

3. Microcontroller: Use a microcontroller like Raspberry Pi, Arduino, or ESP8266 to interface with the sensors and send data to the central platform.

Central Restroom Information Platform

1. **Database:** Set up a database to store sensor data. You can use databases like MySQL, PostgreSQL, or a NoSQL database like MongoDB, depending on your requirements.

2**. Web Application or API:** Create a web application or API to handle data reception and visualization. You can use Flask, Django, or any other web framework for Python.

Python Script for IoT Sensors

1. Occupancy Sensor Script:

- Use a Python script on the IoT sensors to read data from the occupancy sensor.

- When occupancy is detected, send a message with occupancy data to the central platform. You can use MQTT, HTTP, or other communication protocols to send data.

Here's a simplified Python script for an occupancy sensor using MQTT:

```python
import paho.mqtt.client as mqtt
import RPi.GPIO as GPIO
import time

def on_connect(client, userdata, flags, rc):
    print("Connected with result code " + str(rc))

GPIO.setmode(GPIO.BCM)
GPIO.setup(17, GPIO.IN)  # Example GPIO pin for occupancy sensor

client = mqtt.Client()
client.on_connect = on_connect
client.connect("your-mqtt-broker", 1883, 60)

try:
```

```
    while True:
        if GPIO.input(17):
            print("Occupancy detected")
            client.publish("restroom/occupancy", "occupied")
        else:
            print("Restroom is vacant")
            client.publish("restroom/occupancy", "vacant")
        time.sleep(1)
except KeyboardInterrupt:
    GPIO.cleanup()
```

2. **Cleanliness Sensor Script:**

- Similar to the occupancy sensor, create a Python script to read data from the cleanliness sensor.

- When the sensor indicates that cleaning is needed, send a message to the central platform.

3. Data Sending:

- Use a suitable protocol (e.g., MQTT, HTTP, or WebSocket) to send data to the central platform.

- Include the sensor data, timestamp, and any additional relevant information.

1. Data Reception: Develop a service on your central platform to receive data from the IoT sensors. You can use libraries like Flask for HTTP endpoints or a MQTT broker for MQTT communication.

2. Data Storage: Store the received data in a database for historical analysis.

3. Data Visualization: Create a dashboard or user interface to visualize occupancy and cleanliness data in real-time. You can use libraries like Plotly, Matplotlib, or data visualization tools like Grafana.

4. Alerts: Implement alerting mechanisms to notify maintenance staff when cleaning is needed or when restrooms are occupied for an extended period.

5. Analytics: Use the collected data for analytics and insights to improve restroom management.

Keep in mind that this is a simplified example, and a production-ready system would require additional features, security measures, and scalability considerations. It's also important to consider privacy and data protection regulations, especially when dealing with occupancy data from public restrooms.