**Phase 3 project**

**Project Title: SMART PUBLIC RESTROOM**

**Project ID:** proj\_223730\_Team\_1

**College code:** 6208

**College:** Gnanamani College of Technology

**Year:** IIIrd year

**Branch:** B.Tech-Information Technology

Team Members:

Boopathi.V(620821205009)

Pandiyaraj.S.L(620821205039)

Saranraj.P(620821205049)

Sathishkumar.V(620821205050)

Muralitharan.R(620821205301)

**SMART PUBLIC RESTROOM**

**Definition:**

A smart public restroom, also known as an intelligent or connected restroom, refers to a technologically advanced facility designed to enhance user experience, hygiene, and resource efficiency in public restroom environments. It leverages a range of Internet of Things (IoT) technologies and other smart features to provide a more convenient, safe, and environmentally-friendly experience for users.

**PHASE 3**

Deploying IoT sensors in public restrooms to collect data on occupancy and cleanliness is a great idea to improve the user experience and maintenance efficiency. Here's a high-level overview of how you can develop a Python script for IoT sensors to send real-time data to a restroom information platform.

**Hardware and Sensor Setup**

1. **Select IoT Hardware:** Choose appropriate IoT hardware for your project. This might include microcontrollers (e.g., Raspberry Pi, ESP8266, Arduino) or dedicated IoT devices.
2. **Occupancy Sensors:** Install occupancy sensors (e.g., motion detectors or people counters) in strategic locations within the restroom to detect occupancy.
3. **Cleanliness Sensors**: Install cleanliness sensors (e.g., air quality sensors, touchless soap dispensers) to monitor cleanliness.

**Python Script for IoT Sensors:**

For the Python script running on the IoT sensors, you'll need to collect data from the sensors and send it to your restroom information platform. Here's a simplified example for a Raspberry Pi using the requests library to make HTTP POST requests to your platform:

import requests

import time

import random

# Sensor Configuration

sensor\_id = 1 # Unique identifier for this sensor

restroom\_id = 'A101' # Unique identifier for the restroom

api\_url = 'https://your-restroom-platform.com/api/data' # The endpoint to send data

# Simulate sensor data (replace with actual sensor readings)

def read\_occupancy\_sensor():

return random.choice([0, 1])

def read\_cleanliness\_sensor():

return random.uniform(0, 1)

while True:

occupancy = read\_occupancy\_sensor()

cleanliness = read\_cleanliness\_sensor()

# Create a JSON payload with sensor data

data = {

'sensor\_id': sensor\_id,

'restroom\_id': restroom\_id,

'occupancy': occupancy,

'cleanliness': cleanliness

}

try:

# Send data to the restroom information platform

response = requests.post(api\_url, json=data)

if response.status\_code == 200:

print("Data sent successfully.")

else:

print("Failed to send data. Status code:", response.status\_code)

except Exception as e:

print("An error occurred:", e)

time.sleep(60) # Adjust the interval as needed

This script simulates sensor data, but in a real-world scenario, you would replace the **read\_occupancy\_sensor** and **read\_cleanliness\_sensor** functions with code to read data from your actual sensors.

**Restroom Information Platform**

You'll need to set up a restroom information platform to receive, process, and display the data sent by the IoT sensors. This platform should have APIs to receive the data and store it for analysis.

Remember to handle authentication, security, and error handling in both the IoT sensor script and the platform to ensure the reliability and security of your system. Also, consider how you will visualize and analyze the collected data for restroom management.