**SMART PUBLIC RESTROOM**

1. **Feature Engineering:**

Feature engineering is crucial in any machine learning project. For a smart public restroom in IoT, some potential features to consider are:

- Occupancy sensor data: Measure the number of people in the restroom using motion sensors or infrared sensors.

- Temperature and humidity: Monitor the climate inside the restroom for user comfort.

- Air quality: Utilize sensors to assess air quality, especially for gas leakage or foul odors.

- Toilet paper and soap dispensers: Install sensors to monitor refill levels.

- Noise level: Use sound sensors to detect noise disturbances.

- Door status: Track if the restroom is open or closed.

2. **Data Collection and Preprocessing:**

Collect data from the sensors installed in the restroom. Ensure that the data is cleaned, normalized, and transformed into a suitable format for modeling.

3. **Model Training:**

Depending on the specific objectives of your smart restroom project, you might need different machine learning models. Here are some potential use cases and models:

**- Occupancy prediction**: Use time series models or deep learning models to predict restroom occupancy at different times of the day.

- **Anomaly detection**: Train anomaly detection models to identify unusual conditions like high noise levels, poor air quality, or low stock levels in dispensers.

- **Resource optimization**: Implement optimization algorithms to manage resource replenishment efficiently.

4. **Model Evaluation:**

Assess the performance of your models using appropriate metrics. For example:

- For occupancy prediction, you can use metrics like Mean Absolute Error (MAE) or Mean Squared Error (MSE).

- For anomaly detection, you may use precision, recall, and F1-score.

- For resource optimization, measure how well the system minimizes resource wastage while ensuring availability.

5. **User Interface:**

Design a user-friendly interface for both restroom users and administrators. This could be a mobile app or a web dashboard. The interface can display occupancy information, restroom conditions, and alerts.

6. **Connectivity and IoT Integration:**

Ensure that the restroom's sensors are connected to the internet for real-time data transmission. Use IoT protocols and platforms to manage these devices and gather data.

7. **Alerting and Automation:**

Set up alerting mechanisms to notify maintenance staff or administrators when anomalies are detected. Implement automated systems for resource replenishment, such as ordering more toilet paper or soap when levels are low.

8. **Testing and Deployment:**

Before deploying your system in a real public restroom, thoroughly test it in a controlled environment to identify and fix any issues. After successful testing, deploy the system in a real-world restroom.

9. **Maintenance and Updates:**

Regularly maintain and update the IoT system and machine learning models to ensure they continue to function optimally and adapt to changing conditions.

10. **Data Privacy and Security:**

Implement strong security measures to protect the data collected from the restroom sensors and ensure that privacy concerns are addressed.

11. **User Education**:

If necessary, provide information and instructions to restroom users on how to interact with the smart features to improve their experience.

Building a smart public restroom in IoT is a complex project that combines hardware, software, data science, and user experience design. It's essential to consider all these aspects to create a reliable and user-friendly system that enhances the restroom experience.

12. **Program**

import time

class SmartRestroom:

def \_\_init\_\_(self):

self.occupied = False

self.toilet\_paper\_level = 100 # Initial toilet paper level (%)

self.soap\_level = 100 # Initial soap level (%)

def detect\_occupancy(self):

return self.occupied

def flush\_toilet(self):

if self.occupied:

print("Flushing the toilet...")

time.sleep(2) # Simulating flushing time

print("Toilet is now empty.")

self.occupied = False

else:

def refill\_toilet\_paper(self):

print("Refilling toilet paper...")

time.sleep(2) # Simulating refill time

self.toilet\_paper\_level = 100

print("Toilet paper has been refilled to 100%.")

def dispense\_soap(self):

print("Dispensing soap...")

time.sleep(2) # Simulating soap dispensing time

self.soap\_level -= 10 # Simulate soap consumption

if self.soap\_level < 0:

self.soap\_level = 0

print("Soap has been dispensed.")

def monitor\_resource\_levels(self):

print(f"Toilet Paper Level: {self.toilet\_paper\_level}%")

print(f"Soap Level: {self.soap\_level}%")

def run(self):

while True:

print("\nSmart Restroom Menu:")

print("1. Detect Occupancy")

print("2. Flush Toilet")

print("3. Refill Toilet Paper")

print("4. Dispense Soap")

print("5. Monitor Resource Levels")

print("6. Exit")

choice = input("Enter your choice: ")

if choice == '1':

self.occupied = not self.occupied

status = "occupied" if self.occupied else "vacant"

print(f"Toilet is now {status}.")

elif choice == '2':

self.flush\_toilet()

elif choice == '3':

self.refill\_toilet\_paper()

elif choice == '4':

self.dispense\_soap()

elif choice == '5':

self.monitor\_resource\_levels()

elif choice == '6':

print("Exiting the Smart Restroom. Goodbye!")

break

else:

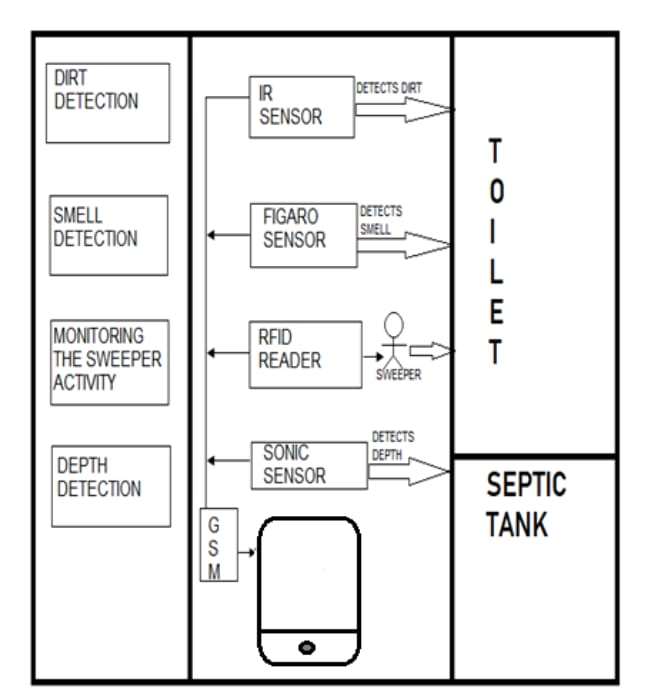
print("Invalid choice. Please try again.")

if \_\_name\_\_ == "\_\_main\_\_":

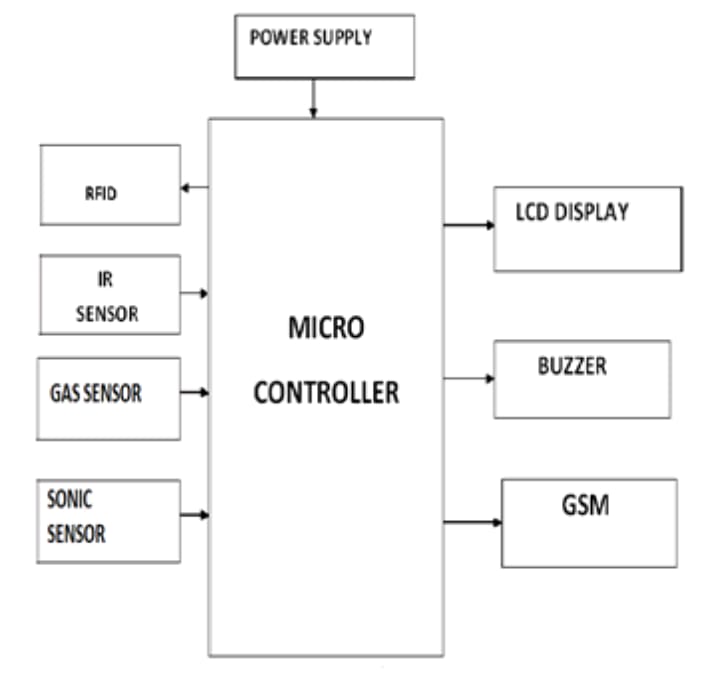
smart\_restroom = SmartRestroom()

smart\_restroom.run()

13**. Architecture of the proposed system:**



14. **Block diagram of the proposed system:**



15. **Working Model:**

This is the module of the proposed system. Here the sensors are connected with the microcontroller.

