**SMART PUBLIC RESTROOM**

**PROBLEM DEFINITION:**

The aim of this project is to address these issues and develop a "Smart Public Restroom" solution that leverages technology to enhance the accessibility, cleanliness, and overall usability of public restrooms.

**MALFUNCTIONS:**

* Sensor Failures
* Water Supply Issues
* Soap and Paper Dispenser Jamming
* Waste Management
* Power Outages
* Vandalism and Damage

**ALGORITHMS:**

* + Use presence sensors (e.g., motion sensors or infrared sensors) to detect occupancy. Analyze sensor data to identify patterns and optimize cleaning schedules.
  + Utilize queue prediction models to estimate wait times based on historical data and current occupancy. Display estimated wait times to users and staff for better planning.
  + Implement sensors and inventory tracking systems to monitor supply levels. Automatically trigger replenishment orders when supplies are low.
  + Use occupancy data to determine when to activate flush mechanisms, reducing unnecessary flushes. Implement water-saving flush modes based on usage patterns.
  + Incorporate occupancy sensors to control lighting and ventilation. Adjust climate control based on occupancy and external weather conditions.
  + Analyze restroom usage data to identify low-traffic periods for cleaning. Implement an algorithm to prioritize cleaning based on occupancy and usage patterns.
  + Utilize sensor data and automated diagnostics to detect issues such as clogs, leaks, or malfunctioning fixtures. Send alerts to maintenance staff for prompt resolution.
  + Utilize sensor data and automated diagnostics to detect issues such as clogs, leaks, or malfunctioning fixtures. Send alerts to maintenance staff for prompt resolution.
  + Use access control algorithms to manage electronic locks and entry permissions. Implement intrusion detection algorithms to alert security personnel in case of unauthorized access attempts.
  + Develop a system for users to provide feedback on cleanliness, maintenance, and overall satisfaction. Analyze feedback data to identify areas for improvement.
  + Implement algorithms for emergency communication and response. This includes notifying emergency services, providing guidance to users, and activating emergency protocols.

**PYTHON CODE:**

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:

print("Toilet is already vacant.")

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()

**FLOW CHAT:**

Start the program

Initialize Smart Restroom with default values (occupancy, toilet paper, and soap levels)

Main Program Loop

**Display Menu**

1. Detect Occupancy
2. Flush Toilet
3. Refill TP
4. Dispense Soap
5. Monitor Levels
6. Exit

Toggle occupancy status

(vacant/occupied)

**1. Detect**

User chooses an option from the menu (e.g., "1" for detecting occupancy)

Option: Detect Occupancy

Toggle occupancy status (occupied / vacant)

Back to the Main Program Loop

Print occupancy status (E.g. "Toilet is now occupied /vacant.")

(other options)

Loop continues until the user chooses to exit the program. (e.g., "6" for Exit)

Option: Exit

Print "Exiting the Smart Restroom. Goodbye!"

End of the Program

**s**