

SMART PUBLIC RESTROOM

-A Comprehensive solution

TEAMNAME : Proj_224780_Team_8

Team Members:

MINUKU SIDHARDHA	(113321205028)
DABBUGOTTU MANEESH	(113321205010)
CHEMUKULA SRILESH	(113321205009)
AAKASH.B	(113321205001)

Phase2 : Innovation

Smart public restrooms can greatly enhance user experience and hygiene. Here are some innovative ideas:

- **Automated Cleaning:**

Incorporate sensors and robotics to automatically clean and disinfect restroom fixtures and floors, reducing the need for human intervention.

- **Touchless Fixtures:**

Use touchless faucets, soap dispensers, and flush mechanisms to minimize germ transmission.

- **Real-time Availability Monitoring:**

Implement sensors to track restroom occupancy and display real-time availability on a mobile app, helping users find the nearest open restroom.

- **Energy Efficiency:**

Utilize smart lighting and HVAC systems that adjust based on occupancy to save energy.

- **Water Conservation:**

Install water-saving technologies like low-flow toilets and urinals, coupled with leak detection sensors.

- **Hygiene Stations:**

Provide touchless hand sanitizing stations at restroom entrances and exits.

- **Odor Control:**

Use smart air fresheners and ventilation systems to maintain pleasant restroom air quality.

- **Accessible Design:**

Ensure accessibility for people with disabilities with features like accessible stalls, Braille signage, and audio assistance.

- **Sustainability:**

Incorporate sustainable materials and practices, such as solar panels for energy or rainwater harvesting for flushing.

- **Security:**

Implement security cameras and emergency buttons for user safety.

- **Gender-Neutral Facilities:**

Design inclusive spaces that cater to all genders.

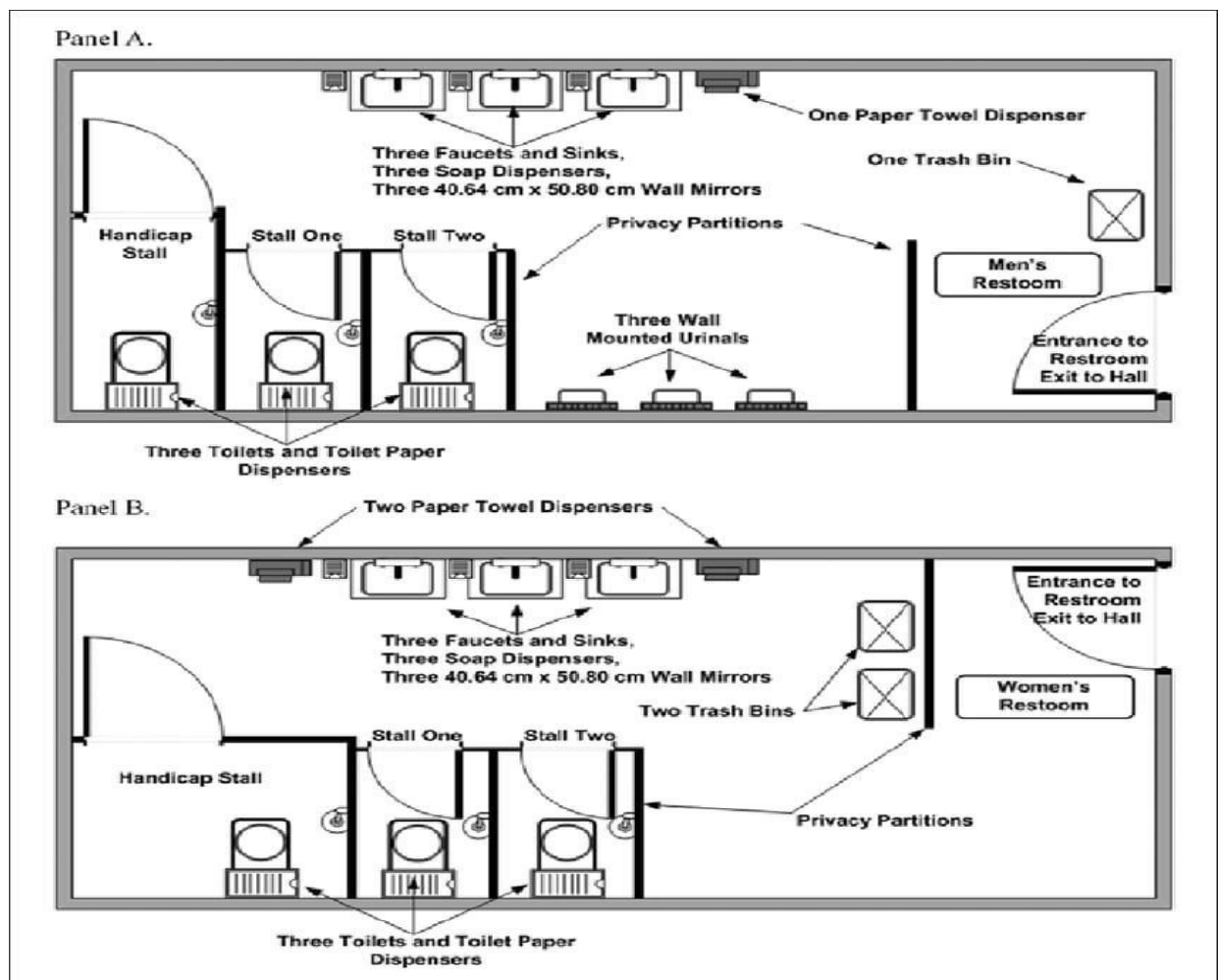
- **Augmented Reality Navigation:**

Develop AR apps that guide users to the nearest restroom within a complex facility like a mall.

- **Health Monitoring:**

Integrate health monitoring sensors for early detection of health issues, like blood pressure or glucose levels.

IoT SENSOR DESIGN:



Solution:-

Creating a smart public restroom involves integrating various technologies for efficiency, cleanliness, and user experience. Here's a high-level solution:

Automated Entry and Exit: Implement touchless access control using RFID cards, QR codes, or motion sensors to enter and exit the restroom.

Occupancy Monitoring: Install sensors to track restroom occupancy and display availability status outside. This reduces wait times.

Smart Toilets and Urinals: Utilize water-saving, self-cleaning toilets and urinals that can detect usage and adjust flushing accordingly.

Hygiene Stations: Provide touchless soap dispensers, faucets, and hand dryers to minimize contact and encourage proper handwashing.

Odor Control: Use air quality sensors and air purifiers to maintain a fresh atmosphere. Automated fragrance dispensers can be used as well.

Cleaning Robots: Employ autonomous robots to clean and sanitize the restroom during low-traffic periods.

Smart Maintenance: Use IoT sensors to monitor restroom supplies (toilet paper, soap, etc.) and alert maintenance staff when refills are needed.

Energy Efficiency: Optimize lighting and HVAC systems to conserve energy. Use motion sensors to control lighting.

User Feedback: Implement a system for users to report issues or provide feedback on restroom cleanliness and functionality.

Accessibility: Ensure the restroom is accessible to all, with features like grab bars, accessible stalls, and audio cues for visually impaired users.

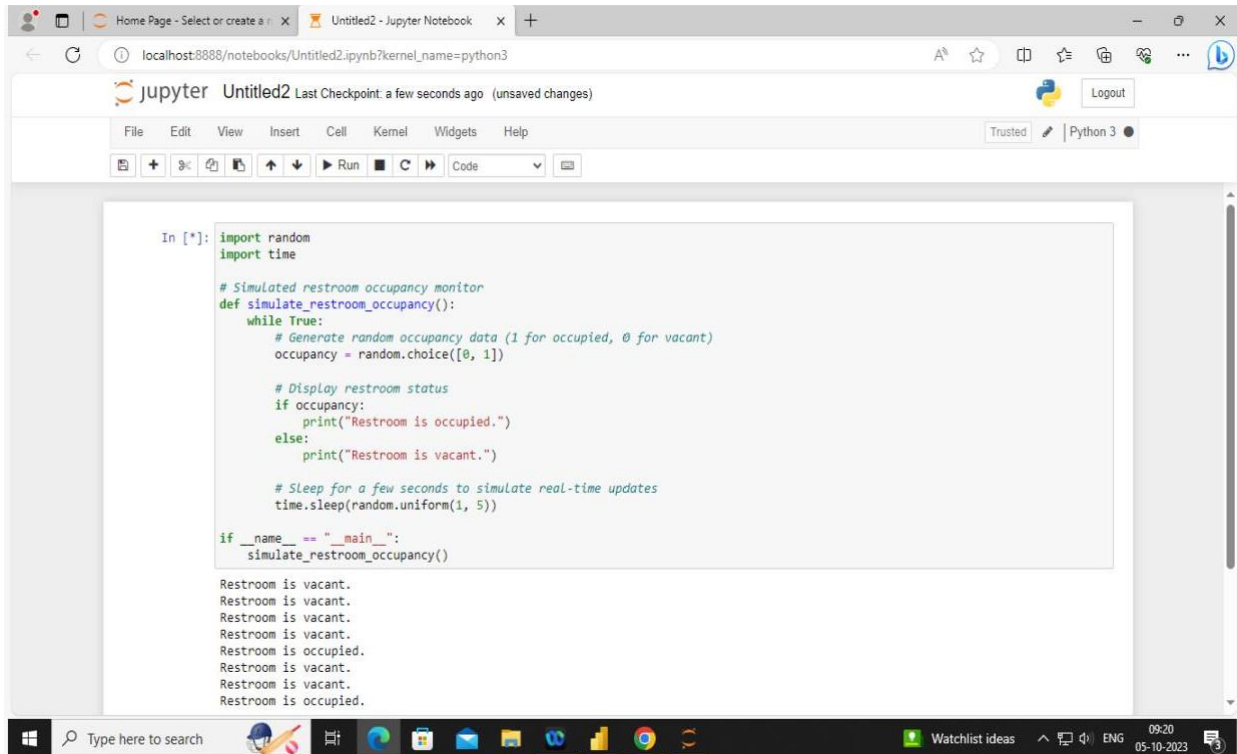
Data Analytics: Collect data on restroom usage, traffic patterns, and maintenance needs to improve efficiency and user satisfaction.

Security and Privacy: Protect user data and ensure restroom security to prevent misuse of smart features.

Maintenance Schedule: Create a regular maintenance schedule to ensure all smart components are functioning correctly.

Emergency Features: Install panic buttons or emergency alarms for user safety.

Code to find whether the restroom is occupied or vacant:



The screenshot shows a Jupyter Notebook interface with a single code cell. The code defines a function `simulate_restroom_occupancy()` that runs in an infinite loop. Inside the loop, it generates a random occupancy value (0 for vacant, 1 for occupied) and prints the status. It also includes a sleep function to simulate real-time updates. The code is executed, and the output shows a sequence of "Restroom is vacant." and "Restroom is occupied." messages.

```
In [*]: import random
import time

# Simulated restroom occupancy monitor
def simulate_restroom_occupancy():
    while True:
        # Generate random occupancy data (1 for occupied, 0 for vacant)
        occupancy = random.choice([0, 1])

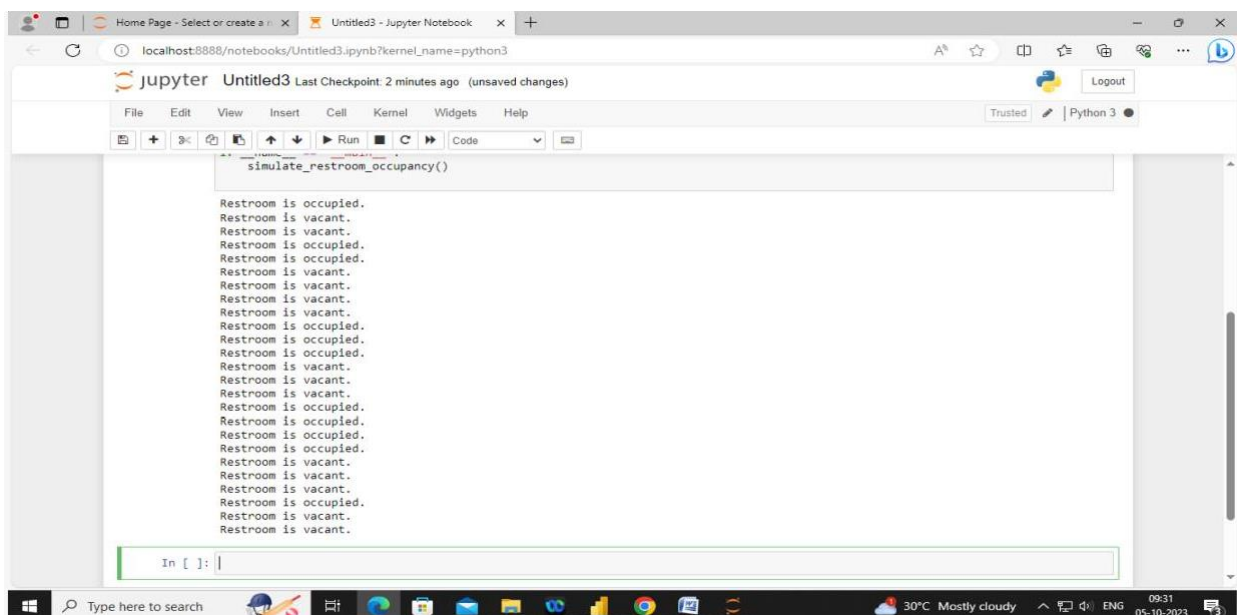
        # Display restroom status
        if occupancy:
            print("Restroom is occupied.")
        else:
            print("Restroom is vacant.")

        # Sleep for a few seconds to simulate real-time updates
        time.sleep(random.uniform(1, 5))

if __name__ == "__main__":
    simulate_restroom_occupancy()

Restroom is vacant.
Restroom is vacant.
Restroom is vacant.
Restroom is vacant.
Restroom is occupied.
Restroom is vacant.
Restroom is vacant.
Restroom is occupied.
```

Sample output:



The screenshot shows the same Jupyter Notebook interface, but now the code cell is collapsed, and the output is visible. The output consists of a long list of "Restroom is occupied." and "Restroom is vacant." messages, demonstrating the random nature of the simulation. The code cell is titled `simulate_restroom_occupancy()`.

```
simulate_restroom_occupancy()

Restroom is occupied.
Restroom is vacant.
Restroom is vacant.
Restroom is occupied.
Restroom is vacant.
Restroom is vacant.
Restroom is vacant.
Restroom is occupied.
Restroom is occupied.
Restroom is occupied.
Restroom is vacant.
Restroom is vacant.
Restroom is occupied.
Restroom is occupied.
Restroom is occupied.
Restroom is vacant.
Restroom is vacant.
Restroom is vacant.
Restroom is occupied.
Restroom is vacant.
Restroom is vacant.

In [ ]: |
```

When you run this code, it will continuously print either "Restroom is occupied." or "Restroom is vacant." with random time intervals in between to simulate occupancy changes. You can stop the program manually when you're done observing the simulated output.