



VELAMMAL
INSTITUTE OF TECHNOLOGY

Approved by AICTE - New Delhi
Affiliated to Anna University - Chennai
Accredited by NBA & NAAC

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PROJECT NAME: SMART PUBLIC RESTROOM

TEAM NAME: proj_224780_Team_4

TEAM MEMBERS:

ABINAYA.P(113321104001)

ABHIRAMI.T(113321104002)

ALAGANENI SUMI(113321104003)

GEETHA.S.V

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.
- ❑ **Energy Efficiency:** Utilize smart lighting and HVAC systems that adjust based on occupancy to save energy

❑ **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.

❑ **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:**

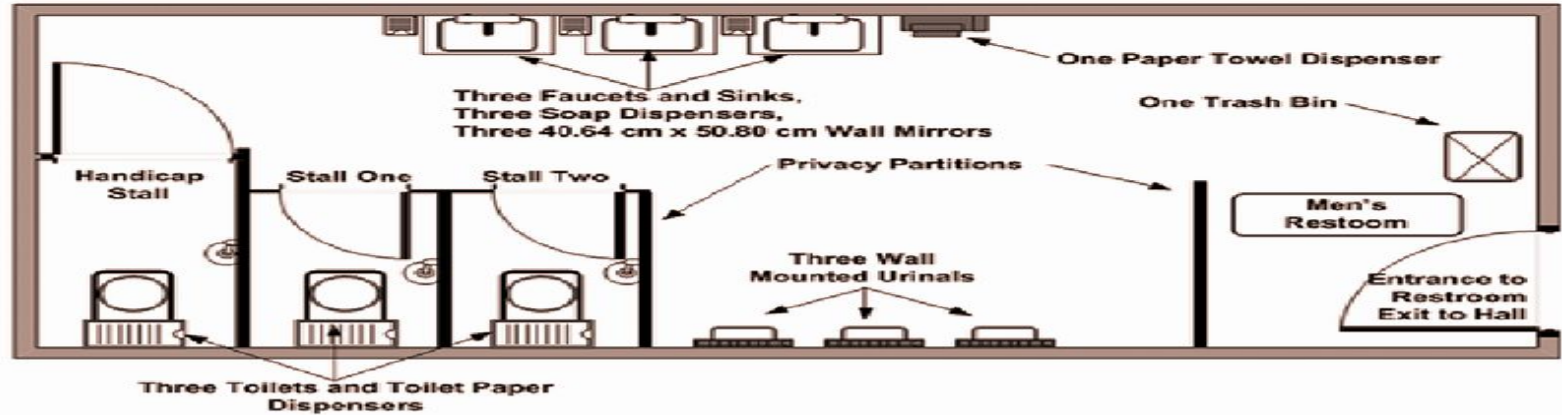
Implement security cameras and emergency buttons for user safety.

❑ **Gender-Neutral Facilities:**

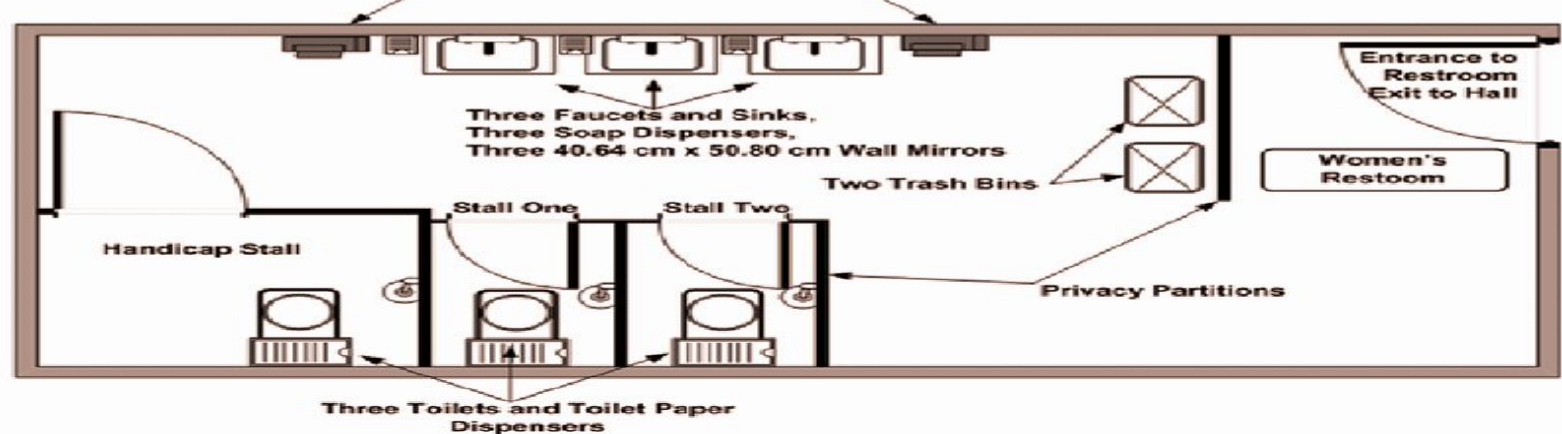
Design inclusive spaces that cater to all genders.

IoT SENSOR DESIGN:

Panel A.



Panel B.



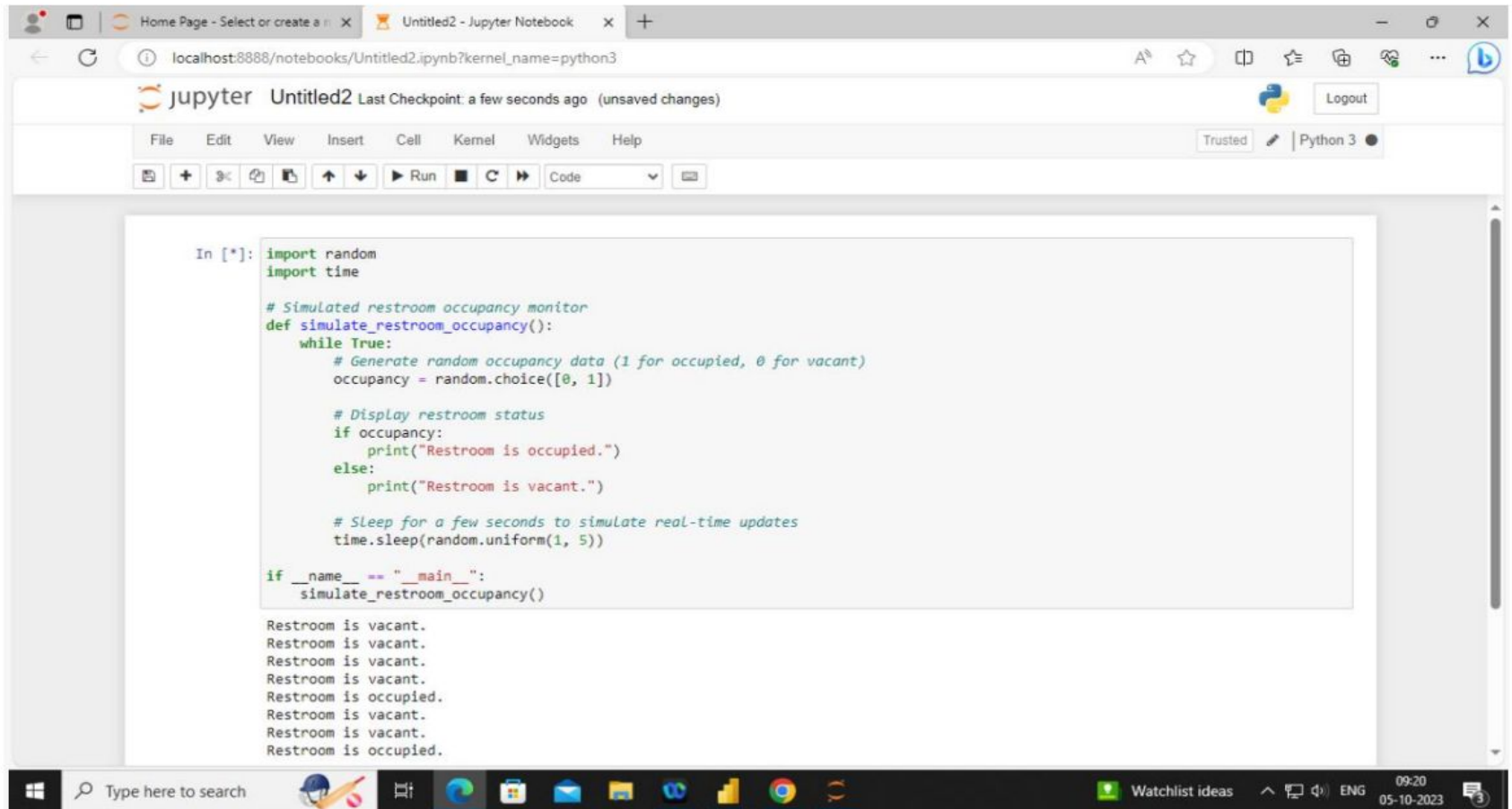
Solution:

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

- ❑ **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.

Code to find whether the restroom is occupied or vacant:



The screenshot shows a web browser window displaying a Jupyter Notebook. The browser's address bar shows the URL `localhost:8888/notebooks/Untitled2.ipynb?kernel_name=python3`. The Jupyter Notebook interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations and execution. The notebook's title bar reads "jupyter Untitled2 Last Checkpoint: a few seconds ago (unsaved changes)".

The code in the notebook cell is as follows:

```
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))

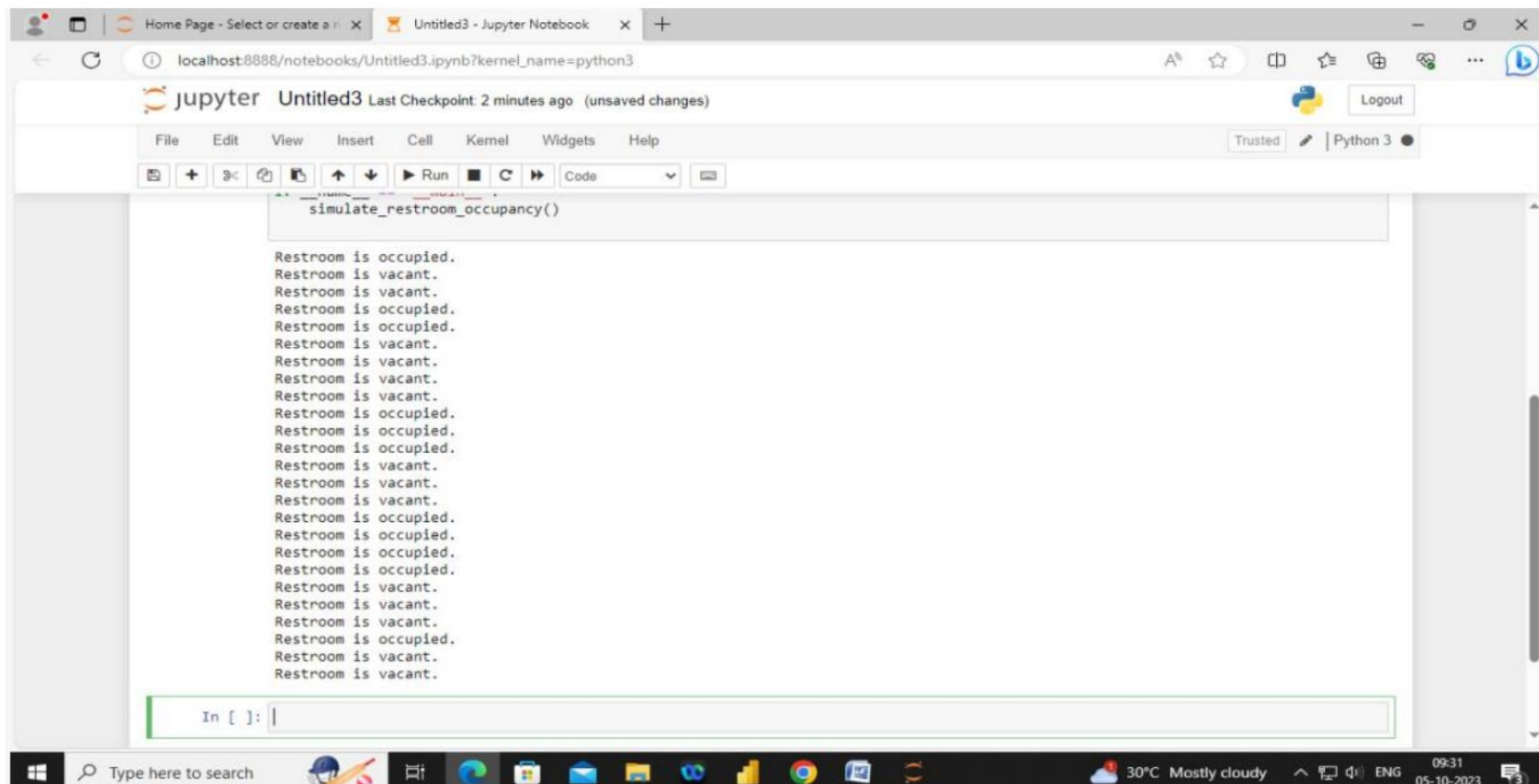
if __name__ == "__main__":
    simulate_restroom_occupancy()
```

The output of the code, displayed below the cell, shows a sequence of status messages:

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

The Windows taskbar at the bottom of the screen shows the system clock as 09:20 on 05-10-2023, along with various application icons and a search bar.

Sample Output:



The screenshot shows a Jupyter Notebook interface in a web browser. The browser's address bar displays `localhost:8888/notebooks/Untitled3.ipynb?kernel_name=python3`. The notebook's title bar indicates it is 'Untitled3' with 'Last Checkpoint: 2 minutes ago' and '(unsaved changes)'. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running, and other functions. The kernel is identified as 'Python 3'. The code cell contains the function `simulate_restroom_occupancy()`. The output of this function is a list of 20 status messages: 'Restroom is occupied.' followed by 'Restroom is vacant.', and then a sequence of alternating 'occupied' and 'vacant' states. The input prompt 'In []:' is visible at the bottom of the cell.

```
simulate_restroom_occupancy()
```

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



THANK YOU