(6115)MAHENDRA INSTITUTE OF ENGINEERING AND TECHNOLOGY

## SMART PUBLIC RESTROOM

## TEAMCODE;Proj\_223280\_Tem\_1,TEAM ID :555

**DOMAIN: INTERNET OF THINGS (IOT)**

## TEAM MEMBERS:

AJIN PIYATO JEN.P (611521104005)

AJAY.K (611521104003)

AJAY.R (611521104004)

ABINESH.N (611521104002)

AKILAN.J (611521104007)

## Faculty Mentor Name:

## SANTHANARAJ.M

**PHASE-3**

# Process and data:

## Hardware Setup:

Install sensors like occupancy sensors, motion sensors, temperature sensors, humidity sensors, etc., in the restroom.

Connect these sensors to an Arduino Uno board.

## Arduino Programming:

Write an Arduino sketch to read data from the sensors. Process and format the sensor data.

Use a serial connection to send the data to a connected computer.

## Python Script:

Develop a Python script to run on a computer (or a Raspberry Pi) connected to the Arduino Uno. Conﬁgure the script to read data from the Arduino Uno via a serial connection.

## ThingSpeak Integration:

Create a ThingSpeak channel to receive and store the sensor data. Obtain an API key for your ThingSpeak channel.

\*\*Python Script for ThingSpeak:\*\*

Modify the Python script to format the sensor data

Send an HTTP POST request to ThingSpeak with the formatted data, using the ThingSpeak API key.

## ThingSpeak Data Storage:

ThingSpeak will store the data sent by your Python script.

## Data Analysis and Visualization:

ThingSpeak provides built-in tools for data visualization and analysis.

You can create charts, graphs, and triggers based on the data to monitor restroom usage and conditions.

## Alerts and Notiﬁcations (Optional):

Conﬁgure ThingSpeak to send alerts or notiﬁcations when certain conditions are met, like low soap levels or high restroom occupancy.

## Document Creation:

Document your project, including hardware setup, Arduino code, Python script, and ThingSpeak conﬁguration.

Explain how the system works, the sensors used, and the beneﬁts of having a Smart Public Restroom.

Share this document for assessment as mentioned in your original request.

## Maintenance and Monitoring:

Regularly monitor the system and perform maintenance on sensors and hardware as needed. Review and analyze data to make improvements in restroom management.

# Used Sensors:

Occupancy Sensors: These can detect if someone is inside the restroom and help manage lighting and ventilation based on occupancy.

Motion Sensors: Useful for detecting movement, ensuring lights and water ﬁxtures are activated when someone enters.

Ultrasonic Sensors: They can measure water levels in toilets and urinals, helping to monitor usage and maintenance needs.

Temperature and Humidity Sensors: These sensors help control the climate within the restroom for user comfort.

CO2 Sensors: To monitor air quality and trigger ventilation systems when needed for odor and health reasons.

Door Sensors: Indicate when restroom doors are opened or closed, useful for occupancy tracking.

Water Quality Sensors: To monitor the quality of water in sinks and toilets, ensuring cleanliness and detecting issues.

Toilet Paper Dispenser Sensors: To monitor and report on the availability of essential supplies.

Soap Dispenser Sensors: To keep track of soap levels and reﬁll requirements.

Hand Dryer Sensors: To monitor usage and maintenance needs for hand dryers.

Waste Bin Sensors: Indicate when the trash bins need emptying.

# Python script for smart public restroom:

class Restroom: def init (self):

self.occupancy = False

self.cleaning\_schedule = 0

def enter(self):

if not self.occupancy: self.occupancy = True

print("Restroom is now occupied.") else:

print("Restroom is already occupied.")

def exit(self):

if self.occupancy: self.occupancy = False

print("Restroom is now vacant.") else:

print("Restroom is already vacant.")

def clean(self):

if self.cleaning\_schedule > 0: self.cleaning\_schedule -= 1

print(f"Restroom cleaned. Next cleaning in {self.cleaning\_schedule} hours.") else:

print("No cleaning needed right now.")

def set\_cleaning\_schedule(self, hours): self.cleaning\_schedule = hours

print(f"Cleaning scheduled every {hours} hours.")

# Example Usage: restroom = Restroom()

restroom.set\_cleaning\_schedule(4)

restroom.enter() # Occupied restroom.enter() # Already occupied restroom.exit() # Vacant restroom.exit() # Already vacant

restroom.clean() # No cleaning needed right now

**Output :**

