PHASE 5:FINAL SUBMISSION

Creating project documentation and preparing a submission for a smart public restroom project in IoT is essential to ensure that your project is well-documented and can be effectively communicated to stakeholders and evaluators. Below is a step-by-step guide on how to go about it:

ABSTRACT AND TITLE:

The abstract of project is to provide clean and hygienic toilets or washrooms. All the public toilets should be clean and hygienic means disease free. In our country, government has introduced the scheme called "Swachh Bharat" (Clean India) so keeping the toilets clean is the one of the objective of Clean India Scheme.

Title for the project is "SMART PUBLIC RESTROOM".

INTRODUCTION:

It is introduce to use and maintain the toilets in the clean and hygienic way. The project is based on IOT concepts using different sensors like smell sensor, dirt sensor, sonic sensor, RFID reader, Database. Using these materials we are trying to provide the clean toilets and create the awareness among the people.

Faucets, flush valves and other fixtures integrated with IoT technology can monitor water flow and immediately alert maintenance workers if a plumbing issue arises. Smart toilets help conserve water by using sensors that automatically flush when a user moves away from the toilet.

OBJECTIVES:

The main objective is to maintain the level of cleanliness in the city and form an environment which is better for living. By using this system we can constantly check the level of the garbage in the dustbins which are placed in various parts of the city.

SYSTEM ARCHITECTURE:

This smart toilet uses intelligent sensors (MQ-137) that communicate with a microcontroller (NodeMCU). The gathered data is sent to be visualized in a dashboard made using ReactJS. A mobile application is also developed to allow the toilet's cleanliness status in real time.

SCOPE OF THE PROJECT:

In this paper we are going to provide the clean toilet. This paper can create the awareness among the people about the clean and hygienic toilets. This paper can ensure the responsibilities of the sweeper. Finally, this concept is the one of the stepping stone to the "Clean and disease free India".Let us see about all phases which we have completed for the smart public restroom project.

PHASE 1: PROJECT DEFINITION AND DESIGN THINKING

The Smart Public Restroom using IoT project aims to design and implement
an innovative and efficient public restroom system that leverages Internet of Things (IoT)
technology to enhance user experience, reduce operational costs, and promote
sustainability. This project focuses on creating a restroom facility that offers real-time
monitoring, smart maintenance, and improved hygiene while conserving resources.

PROJECT DEFINTION:

ENHANCE USER EXPERIENCE:

• Provide a clean and well-maintained restroom environment.

• Reduce wait times and queues through real-time occupancy tracking.

• Offer touchless and convenient access to restroom facilities.

Improve Operational Efficiency:

• Implement predictive maintenance to reduce downtime and operational
disruptions.

• Optimize resource consumption, such as water and energy, to reduce costs and
environmental impact.

• Monitor and manage restroom supplies (e.g., toilet paper, soap) efficiently.

Promote Sustainability:

• Implement water-saving features like smart flushing and faucets.

• Monitor and control energy consumption through efficient lighting and HVAC
systems.

• Minimize waste by optimizing trash disposal and recycling.

DESIGN THINKING:

EMPATHIZE:

• Understand the needs and pain points of restroom users, facility managers, and cleaning staff.

• Conduct surveys, interviews, and observations to gather insights into user experiences and challenges.

DEFINE:

• Define specific problems and opportunities based on the information collected.

• Identify key goals and requirements for the smart public restroom, considering user comfort, hygiene, and sustainability.

IDEATE:

• Brainstorm innovative solutions to address the defined problems and meet the

PROJECT GOALS:

• Explore IoT technologies and devices that can be integrated into the restroom design.

PROTOTYPE:

• Create a prototype or mockup of the smart public restroom design, including IoT components and user interfaces.

• Test the prototype with stakeholders to gather feedback and make necessary adjustments.

TEST:

• Conduct real-world testing of IoT sensors, devices, and systems within the restroom environment.

• Evaluate the functionality, reliability, and usability of the smart restroom features.

Implement:

• Develop and deploy the final IoT-based smart public restroom system,

incorporating the tested components and design improvements.

• Ensure seamless integration with existing infrastructure.

ITERATIVE:

• Continuously collect feedback from users, facility managers, and maintenance staff to identify areas for improvement.

• Iterate on the design and functionality of the restroom system to enhance its performance and user satisfaction.

LAUNCH:

• Officially launch the smart public restroom, making it available to the public.

• Promote the restroom's features and benefits to encourage usage and gain user acceptance.

MONITOR AND MAINTAIN:

• Implement ongoing monitoring and maintenance routines to ensure the system's reliability and performance.

• Collect data on resource consumption, user satisfaction, and system health to make data-driven improvements.

SCALE AND EXPAND:

• Consider expanding the deployment of smart public restrooms to other locations or

FACILITIES.

• Explore opportunities for further innovation and integration with smart city initiatives.

• By following the design thinking process and project definition outlined above, you can create a Smart Public Restroom using IoT that not only addresses user needs but also contributes to sustainability and operational efficiency.

PHASE 2: INNOVATION

Creating a Smart Public Restroom using IoT involves innovative solutions to enhance user experience, increase operational efficiency, and promote sustainability. Here are some key innovations that can be implemented in such a restroom.

IOT SENSORS FOR OCCUPANCY MONITORING:

🎬 Utilize occupancy sensors to monitor restroom usage in real time.

🎬 Implement a user-friendly app or signage to display restroom availability.

TOUCHLESS ACCESS CONTROL:

🎬 Use RFID cards, smartphone apps, or biometric authentication for touchless access.

🎬 Incorporate automatic doors and sanitization systems triggered by user presence.

PREDICTIVE MAINTENANCE:

🎬 Deploy IoT sensors to monitor restroom equipment (e.g., toilets, faucets) for usage and wear.

🎬 Implement predictive maintenance to schedule repairs and replacements proactively.

WATER AND ENERGY CONSERVATION:

🎬 Install smart faucets and toilets that adjust water flow based on user needs.

🎬 Use IoT to regulate lighting, HVAC systems, and ventilation according to occupancy.

SUPPLIES MANAGEMENT:

🎬 Implement IoT-connected dispensers for soap, paper towels, and toilet paper.

🎬 Automatically reorder supplies when they are running low.

REAL-TIME FEEDBACK AND REPORTING:

🎬 Provide a feedback mechanism for users to report issues like cleanliness or maintenance needs.

🎬 Utilize IoT to create a maintenance ticketing system that notifies staff of reported issues.

HYGIENE MONITORING:

🎬 Employ sensors to monitor hygiene-related factors, such as soap and paper towel usage.

🎬 Implement UV-C or other sanitation technologies for continuous disinfection.

SUSTAINABILITY FEATURES:

🎬 Incorporate rainwater harvesting for flushing toilets or solar panels for energy generation.

🎬 Promote water recycling and waste reduction strategies.

DATA ANALYTICS AND OPTIMIZATION:

🎬 Collect and analyze data on restroom usage, resource consumption, and user feedback.

🎬 Use insights to continuously optimize operations, reduce costs, and enhance user satisfaction.

USER ENGAGEMENT AND EXPERIENCE:

🎬 Develop a user-friendly mobile app or kiosk interface for accessing and rating the restroom.

🎬 Offer entertainment options (e.g., music displays) to improve the user experience.

ACCESSIBILITY FEATURES:

🎬 Integrate IoT technologies to assist people with disabilities, such as voice-activated controls and adaptive lighting.

SECURITY AND SAFETY MEASURES:

🎬 Implement IoT-enabled security cameras and alarms to ensure user safety.

🎬 Offer emergency call buttons with automatic location reporting.

User Privacy Considerations:

🎬 Ensure that data collected from IoT devices respects user privacy and complies with data protection regulations.

SMART WASTE MANAGEMENT:

🎬 Use IoT sensors to monitor waste bin levels and optimize trash collection routes.

🎬 Implement recycling separation systems to reduce landfill waste.

SCALABILITY AND INTEGRATION:

🎬 Design the system with scalability in mind to deploy in various locations.

🎬 Integrate the smart restroom system with broader smart city initiatives for efficiency.

SENSORY COMFORT:

🎬 Implement air quality sensors to ensure a comfortable environment.

🎬 Adjust music or lighting based on user preferences or time of day.

The key to a successful Smart Public Restroom using IoT is a
user-centered design that prioritizes convenience, cleanliness, and sustainability, while leveraging the power of data and automation to optimize operations.
Innovations in IoT technologies can significantly enhance public restroom facilities, making them more efficient, hygienic, and enjoyable for users.

PHASE 3 : IOT DEVELOPMENT PART 1

To send real-time occupancy and cleanliness data to a restroom information platform

involves a combination of hardware and software components. I'll provide you with a high-level example of how you might structure such a project. Keep in mind that the specific

implementation will depend on the hardware and communication protocols used in your IoT sensors

Using MQTT for communication between the IoT sensors and the platform. MQTT is a lightweight messaging protocol commonly used in IoT applications.

Step 1: Setting up the IoT Sensors
You'll need IoT sensors capable of detecting restroom occupancy and cleanliness. These sensors should be programmed to send data to a central server. Below is a simplified

example for an occupancy sensor.

PROGRAM:

```
import paho.mqtt.client as mqtt
import time
import random
def simulate_occupancy_sensor():


client = mqtt.Client()
client.connect("mqtt.broker.com", 1883)
while True:
occupancy = random.choice([0, 1]) # 0 for empty, 1 for occupied
client.publish("restroom/occupancy", payload=occupancy, qos=1)
```

```
time.sleep(5) # Send data every 5 seconds
if __name__ == "__main__":
simulate_occupancy_sensor()
```

You would have a similar script for cleanliness data.

Step 2: Setting up the Restroom Information Platform

Your platform should have an MQTT broker and a backend to process incoming data. You can use a library like Eclipse Paho to create an MQTT broker.

Step 3: Receive and Process Data on the Restroom Platform

On your platform, you need to create a Python script to receive data from the sensors, process it, and update the restroom information accordingly.

PROGRAM:

```
import paho.mqtt.client as mqtt
def on_message(client, userdata, msg):
topic = msg.topic
payload = msg.payload.decode("utf-8")
if topic == "restroom/occupancy":
process_occupancy_data(payload)
elif topic == "restroom/cleanliness":

process_cleanliness_data(payload)

def process_occupancy_data(data):
print(f"Occupancy data received: {data}")
def process_cleanliness_data(data):
print(f"Cleanliness data received: {data}")
client = mqtt.Client()
```

```
client.connect("mqtt.broker.com", 1883)
client.on_message = on_message
client.subscribe("restroom/occupancy", qos=1)
client.subscribe("restroom/cleanliness", qos=1)
client.loop_forever()
```

This script subscribes to MQTT topics for occupancy and cleanliness data, processes the data, and updates the restroom information accordingly.

Step 4: Run the Scripts

Now you can run both the sensor and platform scripts. Make sure to replace "mqtt.broker.com" with the actual MQTT broker address we are using.

This is a basic example, and in a real-world scenario, we would need to handle more complex tasks such as data persistence, security, and potentially use a proper IoT framework depending on the scale of your project.

PHASE 4:IOT DEVELOPMENT PART 2

Designing a mobile app for iOS and Android using HTML, CSS, and JavaScript typically involves creating a web app and packaging it within a WebView component for each platform. Here's a simplified example of how you can structure the HTML code for a basic restroom information app. Please note that this is a minimal example and does not cover all aspects of app development.

HTML (index.html):

```
<!DOCTYPE html>
<html>
<head>
<title>Restroom Finder</title>
<link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
<header>
<h1>Restroom Finder</h1>
</header>
<main>
<div id="map"></div>
<div id="restroom-list">
<!-- Restroom listings will be dynamically generated here -->
</div>
</main>
<footer>
<button id="refresh-button">Refresh</button>
</footer>

<script src="app.js"></script>
</body>
```

```
</html>

CSS (styles.css):
body {
font-family: Arial, sans-serif;
}
header {
background-color: #3498db;
color: #fff;
text-align: center;
padding: 10px;
}
main {
padding: 20px;
}
#map {
width: 100%;
height: 300px;
}
#restroom-list {
/* Styling for the restroom listings */
}
footer {
text-align: center;
padding: 10px;
}

button {
background-color: #3498db;
color: #fff;
padding: 10px 20px;

border: none;

cursor: pointer;

}
```

JavaScript (app.js):

```
const restroomData = [

{

name: "Restroom 1",

location: "123 Main St",

cleanliness: "Clean",

availability: "Available",

},

{

name: "Restroom 2",

location: "456 Elm St",

cleanliness: "Average",

availability: "In Use",

},

];

function renderRestroomList() {

const restroomList = document.getElementById("restroom-list");

restroomList.innerHTML = "";

for (const restroom of restroomData) {

const listing = document.createElement("div");
listing.classList.add("restroom-listing");

listing.innerHTML = `
<h2>${restroom.name}</h2>
<p><strong>Location:</strong> ${restroom.location}</p>
<p><strong>Cleanliness:</strong> ${restroom.cleanliness}</p>
<p><strong>Availability:</strong> ${restroom.availability}</p>
` restroomList.appendChild(listing);
}
}
const refreshButton = document.getElementById("refresh-button");
refreshButton.addEventListener("click", renderRestroomList);
renderRestroomList();
```

IN THIS SIMPLIFIED EXAMPLE:

• The HTML file (index.html) defines the structure of the app, including a header, a map section, a restroom listing section, and a refresh button in the footer.

• The CSS file (styles.css) provides basic styling for the app's elements.

• The JavaScript file (app.js) contains sample data, a function to render the restroom listings, and an event listener for the refresh button.

Let us again come to phase 5 and see other topics for the documentation and submissison of the project.

PHASE 5:

KEY FEATURES:

Smart Lighting - automatic turn ON/OFF light and exhaust fan. Smart Hygiene Maintenance - auto flush and floor cleaning. Water Conservation - Decide optimum water to dispense based on time spent. Entry Management - inform users if the toilet is occupied.

WASTE REMOVAL IN SMART PUBLIC RESTROOM USING PYTHON :

Creating a system for automatic waste removal in a public restroom using IoT involves a combination of sensors, actuators, and software to detect and manage waste disposal. Below is a simplified example of a Python program for such a system using a Raspberry Pi, an ultrasonic sensor for detecting waste levels, and a motor for removing waste. Please note that this is a basic example, and a real-world system would be more complex and use appropriate safety mechanisms.

HARDWARE SETUP:

Raspberry Pi (with Raspbian OS)

Ultrasonic sensor (for waste level detection)

Motor or actuator (for waste removal)

PYTHON PROGRAM:

```
import RPi.GPIO as GPIO

import time


# Define GPIO pins


TRIG_PIN = 23  # Ultrasonic sensor trigger pin

ECHO_PIN = 24  # Ultrasonic sensor echo pin

MOTOR_PIN = 18  # Motor control pin


# Initialize GPIO settings

GPIO.setmode(GPIO.BCM)

GPIO.setup(TRIG_PIN, GPIO.OUT)

GPIO.setup(ECHO_PIN, GPIO.IN)

GPIO.setup(MOTOR_PIN, GPIO.OUT)
GPIO.output(MOTOR_PIN, GPIO.LOW)

# Function to measure distance using the ultrasonic sensor

def measure_distance():
    GPIO.output(TRIG_PIN, True)
    time.sleep(0.00001)
    GPIO.output(TRIG_PIN, False)

    while GPIO.input(ECHO_PIN) == 0:
        pulse_start = time.time()
    while GPIO.input(ECHO_PIN) == 1:
        pulse_end = time.time()

    pulse_duration = pulse_end - pulse_start
    distance = pulse_duration * 17150
    distance = round(distance, 2)
    return distance


try:
    while True:
        # Measure the waste level
        distance = measure_distance()

        if distance < 10:  # Replace with an appropriate threshold for your setup
            print("Waste level is high. Removing waste...")
            GPIO.output(MOTOR_PIN, GPIO.HIGH)
            time.sleep(5)  # Run the motor for 5 seconds (adjust as needed)
            GPIO.output(MOTOR_PIN, GPIO.LOW)
        else:
            print("Waste level is within acceptable range.")



    time.sleep(10)  # Check waste level every 10 seconds (adjust as needed)


except KeyboardInterrupt:
    print("Program terminated by user")
    GPIO.cleanup()
```

Creating a physical output for an automatic waste removal system in a public restroom using IoT typically involves controlling a motor or actuator to physically remove waste when certain conditions are met. In the code example provided earlier, when the waste level exceeds a certain threshold, a motor is activated. The physical output would be the motor's movement. Here's what you can expect as the output when the system detects high waste levels:

HIGH WASTE LEVEL DETECTED:

The ultrasonic sensor measures the waste level and finds it to be above the defined threshold (e.g., less than 10 cm in the code example).

The program outputs a message to indicate that the waste level is high.

MOTOR ACTIVATION:

Upon detecting a high waste level, the program activates the motor (or actuator) responsible for waste removal.

The motor physically moves or engages to start the waste removal process.

WASTE REMOVAL:

The motor runs for a specified period (e.g., 5 seconds in the code example) to remove waste.

The waste, which may be in a receptacle, is transported or emptied into a waste storage area.

MOTOR DEACTIVATION:

After the motor has run for the specified time, the program deactivates the motor.

The motor returns to its idle state.

REGULAR MONITORING:

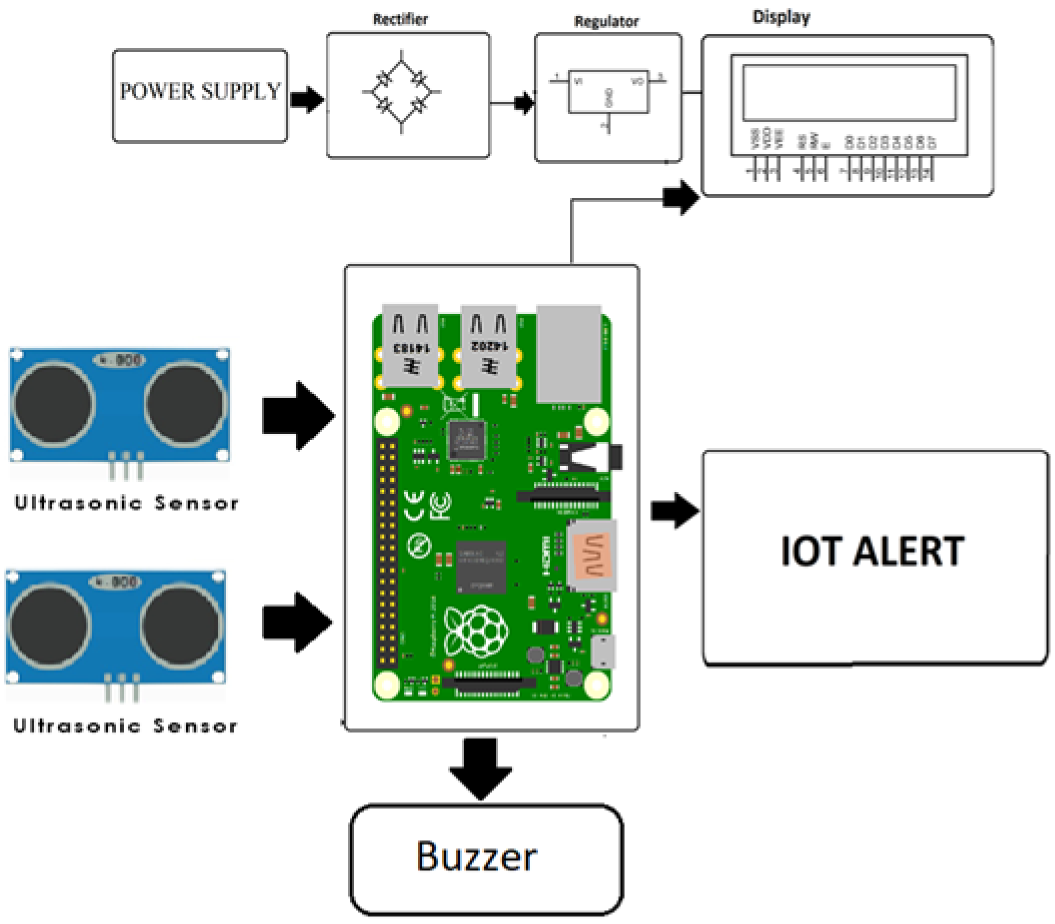
The system continues to regularly monitor the waste level at defined intervals (e.g., every 10 seconds in the code example).

OUTPUT:

The key output of this system is the physical movement of the motor to remove waste when necessary. This automation minimizes the need for manual intervention and helps maintain cleanliness and hygiene in the public restroom.

Keep in mind that in a real-world implementation, the motor's movement and waste removal process should be well-designed and safety measures should be in place to prevent accidents or mishaps. Additionally, the system may log data, send alerts, or communicate with a central management system as part of its operation, depending on the specific requirements of the IoT application.

DIAGRAM:



HARDWARE SPECIFICATIONS:

- Raspberry Pi 3
- Ultrasonic Sensor
- LCD Display
- Resistors
- Capacitors
- Transistors

- Cables and Connectors
- Diodes
- PCB and Breadboards
- LED
- Transformer/Adapter
- Push Buttons
- Switch
- IC
- IC Sockets

SOFTWARE SPECIFICATIONS:

- Raspberry Pi Compiler
- MC Programming Language: Python
- IOT Gecko

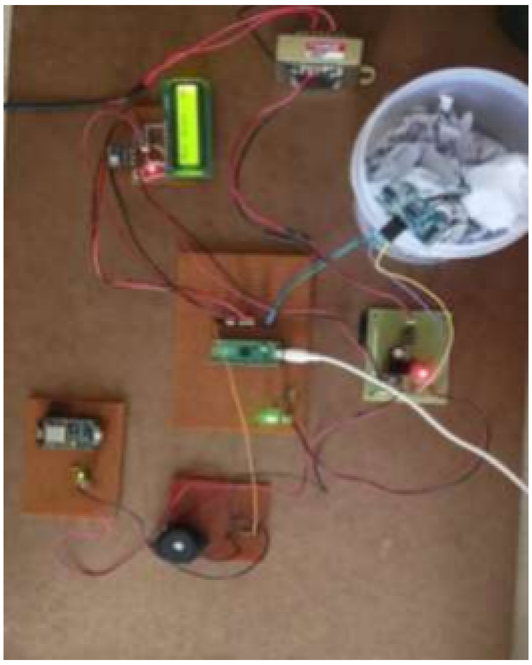
IOT GARBAGE MONITORING USING RASPBERRY PI:

The IOT Garbage Monitoring System is designed by ease the problems people or organisation face while managing their waste. The system allow the user to keep watch on the garbage bins by utilising buzzer and IoT service. The system has a buzzer on it which sets off an alarm on fulfillment of the garbage bin, other than this the user can also watch over the bins from anywhere using IoT service.

The system is constructed using a Model B of Raspberry Pi 3, which works as the brain of the system. It is packed with many features like, a quicker 64-bit 1.4GHz quad core chip, 1GB of RAM, quicker dual-band 802.11 b/g/n/ac wireless LAN, Bluetooth 4.2, and much quicker 300 Mbit/s ethernet. It has Improved thermals on the Pi 3 B+ means that the CPU on the BCM2837 SoC can now run at 1.4GHz. To keep watch on level of garbage in the bins, the system consists of a couple of HC-SR04 Ultrasonic Range Finder Distance Sensor Module. The Ultrasonic sensor works on the principle of SONAR and is designed to measure the distance using ultrasonic wave to determine the distance of an object from the sensor. The sensor helps the system to sense the level of garbage in the bins. The

Raspberry Pi is equipped with WiFi connectivity, thus making it suitable to watch the system using IoT, from anywhere. As so the system works towards ease the garbage management.

SCREENSHOTS:



CONCLUSION:

An IoT based intelligent garbage bin is formulated for the accurate monitoring and managing of waste management in the Municipality office. This system alerts the Municipality office for the particular garbage bin throughout the street to help the sweepers to collect the garbage from bin and clean it respectively, so that garbage won't spill out of the bin and we can control the spread of diseases by improving the manual system to the IoT based system. Thus, this system becomes useful as an efficient solution in improving the street environmental maintenance.