

IBM NAAN MUDHALVAN PROJECT

**COURSE NAME:Internet of things(IOT)**

**TITLE :smart public restrooms**

**GROUP :07**

**PROJECT NUM :08**

**YEAR :03**

**DEPARTMENT :electronics and communication engineering**

[Type the abstract of the document here. The abstract is typically a short summhe abstract of the documenere.]

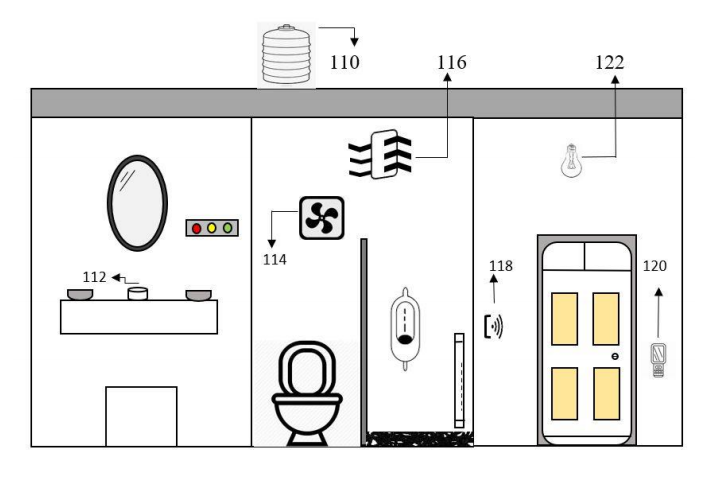
OBJECTIVES:

In this section continue building the project by performing different activities like feature engineering, model training, evaluation on smart public restrooms using IOT.

FEATURED ENGINEERING:

By developing this smart public restrooms in IOT

We have to install an proposed system on it.



1. Water Tank.

120-Odour detection MQ-135.

1. Soap detector.

112 Automatic lights.

116- Ventilation fan

124- Frequency counter.

126- Cleaners attendance.

**CALIBRATION OF SENSORS:**

On the market, there are many distinct types of ammonia gas sensors, each with its own size, sensitivity, and price. We intend to employ the MQ135 gas sensor, which can detect NH3, C2H6O, and even CO. These MQ135 sensors are commonly utilized in factories and houses due to their compact size and lightweight. Tin Oxide (SnO2) is an element that is present inside the sensor and is used to detect various gases. With the use of a potentiometer, the sensor's sensitivity can be changed to meet the needs. When gases in the air come into contact with SnO2, the resistance changes, and the output voltage changes as well, indicating which gas

has been identified.

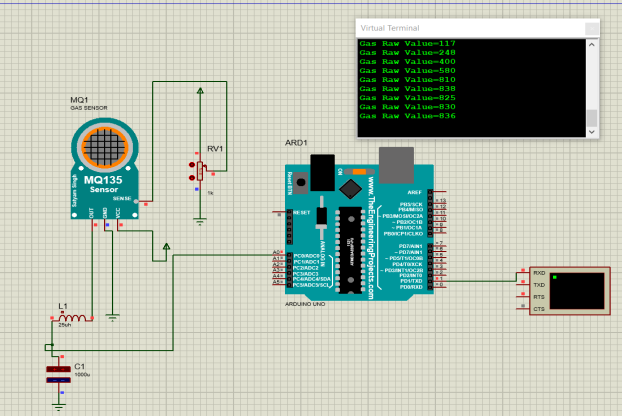
***Data Processing is done over the cloud :***

To begin, data from IOT devices is pre-processed depending on the sensor parameters' values. There might be gaps in this data, as well as false measurements. Erroneous readings are removed from the data, which is then normalized and recorded with a time stamp in Thing speak database storage. This Thing speak is a suitable choice for a system that requires real-time data updates.

MODEL TRAINING:

*A. Simulation of Gas sensor*

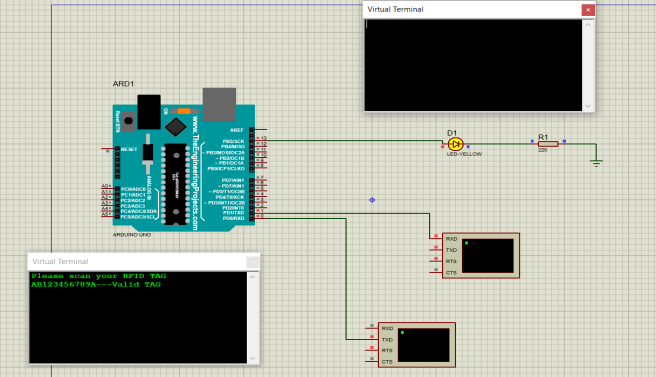
Within the toilet, the MQ-135 sensor detects smells. This MQ-135 sensor measures the amount of ammonia gas present in the toilet's air.



The ammonia inside the toilet produces a nasty odour. It has a strong odour that can be detected at concentrations more than 5 parts per million (parts per million).The ventilation fans are automatically switched on when the ammonia concentration in the toilet surpasses the predetermined threshold of 5 ppm; however, if the concentration falls below 5 ppm, no action is performed, as shown in Fig. 2.

1. *Simulation of RFID Sensor*

The RFID (Radio Frequency Identification) technology, is a key enabling technology of the IoT sensing layer. As shown in Fig. 3, when the cleaner enters the toilet and when he scans his ID in the RFID scanner, It displays his name with the valid tag, and in another case when he scans other than his ID it shows that the ID is invalid. This helps to monitor the cleaner’s attendance.



*C. Real time interface*

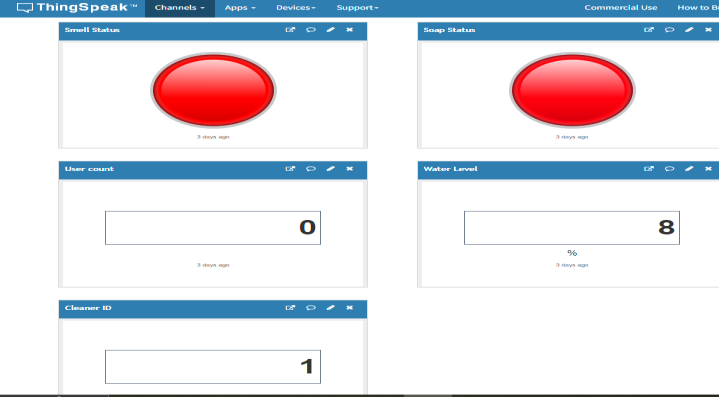
We employed five sensors in our project. The first is an RFID sensor, which is used to track the cleaner's attendance. The MQ-135 sensor was the second sensor we utilized to assess air quality. For soap detection, we employed an IR sensor as the third sensor. Ultrasonic sensor for tank water level measurement is the fourth sensor. The fifth sensor is an infrared sensor that counts how many people enter the

restroom. These sensors are all connected to Arduino, which collects all sensor data and sends it to the cloud via the ESP8266 Wi-Fi module. Here we have used Thingspeakopen-source website for storing the collected sensors data. All the sensors’ data is even displayed on LCD screen as well. Fig. 4 Shows the Thingspeak website where all the data are stored. It shows Smell status, Soap status, user count,

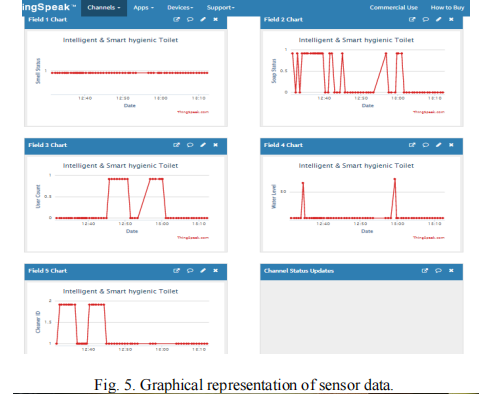
water level, cleaner id. Soap and smell status is determined

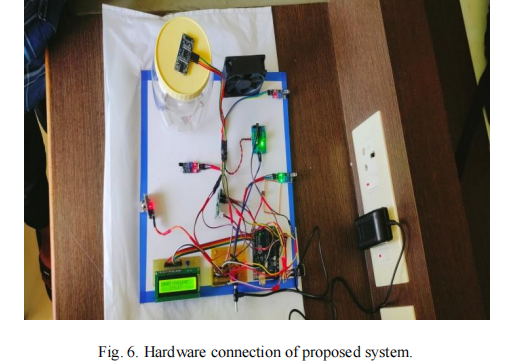
using red light as shown in Fig. 4. If the soap is present in the container then the red LED will turn on, similarly if there is smell in the toilet then the red LED for smell status will turn on. Water level is determined using percentages in water level block. It also counts number of people entering the

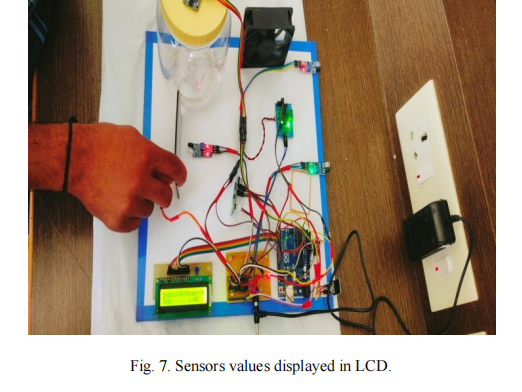
toilet in user count block.



Public view in Thingspeak software.







the sensor output on the LCD screen. As displayed here it shows the smell status, if there is smell in

the room or threshold value of the MQ-135 sensor is reached then the ventilation fan is turned on. The fan is turned on by the L293 motor drive module. Here the smell is in ok condition, so fan will be in off state. similarly goes with the soap detection, if soap is present then the status of the soap will be OK as shown in Fig. 7. If the soap is missing then N.A will be displayed. It also counts number of people entering the toilet. As you can see in Fig. 7, the ‘C’ value in the LCD display is zero. Whenever person enters the toilet, it goes on increasing the count value. Similarly, whenever the person exits the toilet, the count is decremented from one to zero. For water level detection we have used ultrasonic sensor to measure the level of the water in the tank. In our hardware we have used small container to store the water so based on the distance, the ultrasonic sensor can detect the water level which in turn displays the results on the LCD display as well as in Thing speak in percentage format.

**SOFTWARE SETUP:**

Implementing a smart public restroom system using JavaScript for the Internet of Things (IoT) typically involves developing the server and user interface components. Below, I'll provide an outline of how you can structure the JavaScript code for both the server and a simple web-based user interface. Please note that this is a simplified example for illustration purposes.

### Server-Side (Node.js)

1. Set up a Node.js server to handle IoT device communication and data processing.

const express = require('express');

const bodyParser = require('body-parser');

const app = express();

const port = 3000;

app.use(bodyParser.json());

// Implement routes for IoT device data and control

app.post('/sensor-data', (req, res) => {

// Handle incoming sensor data, e.g., occupancy, temperature, water quality

const sensorData = req.body;

// Store or process the data as needed

console.log('Received sensor data:', sensorData);

res.status(200).send('Data received');

});

// Implement routes for IoT device control, e.g., faucet control, smart locks

app.post('/control-faucet', (req, res) => {

// Control faucet based on the request

const { faucetId, action } = req.body;

// Implement control logic here

res.status(200).send('Faucet control request processed');

});

app.listen(port, () => {

console.log(`Server is running on port ${port}`);

});

### User Interface (Web-Based)

Develop a web-based user interface for restroom users and administrators using HTML, CSS, and JavaScript. You can use popular front-end libraries and frameworks like React, Angular, or Vue.js.

Create a simple HTML structure and include JavaScript to interact with the server and display data.

<!DOCTYPE html>

<html>

<head>

<title>Smart Public Restroom</title>

</head>

<body>

<h1>Smart Public Restroom</h1>

<!-- Display restroom occupancy -->

<p>Restroom Status: <span id="restroomStatus">Loading...</span></p>

<!-- Control faucet -->

<button onclick="controlFaucet('faucet1', 'on')">Turn On Faucet</button>

<button onclick="controlFaucet('faucet1', 'off')">Turn Off Faucet</button>

<script>

// Function to fetch and update restroom occupancy

function updateRestroomStatus() {

// Fetch data from the server and update the UI

fetch('/get-restroom-status')

.then((response) => response.json())

.then((data) => {

document.getElementById('restroomStatus').textContent = data.status;

});

}

// Function to control a faucet

function controlFaucet(faucetId, action) {

// Send a request to control the faucet to the server

fetch('/control-faucet', {

method: 'POST',

headers: {

'Content-Type': 'application/json',

},

body: JSON.stringify({ faucetId, action }),

})

.then(() => {

// Handle the response if needed

console.log(`Faucet control request sent: ${action}`);

});

}

// Periodically update restroom occupancy

setInterval(updateRestroomStatus, 5000); // Update every 5 seconds

updateRestroomStatus(); // Initial update

</script>

</body>

</html>

CONCLUSION:

The initiative met its goal of creating an economical, user-friendly interface between the cleaning company and public restrooms, allowing for more effective staffing.

This program's installation is straightforward. Time series forecasts can be utilized to conduct simultaneous toilet research. If this toilet condition is utilized in the toilet, it assists in keeping the toilet clean before it becomes unclean. The mobile app is considerably easier to use now

that the data display has been updated. The Internet of Things device is both inexpensive and portable.

In future, this study might be enhanced by employing sensitive and modern sensors to generate more precise data. By learning from sample data and increasing the amount of test data, machine learning increases prediction accuracy. When this approach is used on a large scale, better storage systems and cloud servers can be used.