PREDICTION OF TOXIC LEVEL IN LAVATORY AND ALERT GENERATING SYSTEM USING ANDROID APPLICATION

A DESIGN PROJECT REPORT

submitted by

SHARU RUBA S

SUBITHA G

VARSHAA S R

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

K RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai, Approved by AICTE, New Delhi)

Samayapuram – 621 112

NOVEMBER , 2024

PREDICTION OF TOXIC LEVEL IN LAVATORY AND ALERT GENERATING SYSTEM USING ANDROID APPLICATION

A DESIGN PROJECT REPORT

submitted by

SHARU RUBA S (811722104144)

SUBITHA G (811722104159)

VARSHAA S R (811722104175)

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

K RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai, Approved by AICTE, New Delhi)

Samayapuram – 621 112

NOVEMBER , 2024

K RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(AUTONOMOUS)

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report titled **"PREDICTION OF TOXIC LEVEL IN LAVATORY AND ALERT GENERATING SYSTEM USING ANDROID APPLICATION"** is bonafide work of **SHARU RUBA S (811722104144), SUBITHA G (811722104159), VARSHAA S R (811722104175)** who carried out the project under my supervision. Certified further, that to the best of my knowledge the work reported here in does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr. A Delphin Carolina Rani M.E.,Ph.D.,

HEAD OF THE DEPARTMENT

PROFESSOR

Department of CSE

K Ramakrishnan College of Technology

(Autonomous)

Samayapuram – 621 112

SIGNATURE

Mrs. A.Themozhi, M.E.,

SUPERVISOR

Assistant Professor

Department of CSE

K Ramakrishnan College of Technology

(Autonomous)

Samayapuram – 621 112

Submitted for the viva-voice examination held on .....

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

We jointly declare that the project report on **"PREDICTION OF TOXIC LEVEL IN LAVATORY AND ALERT GENERATING SYSTEM USING ANDROID APPLICATION"** is the result of original work done by us and best of our knowledge, similar work has not been submitted to **"ANNA UNIVERSITY CHENNAI"** for the requirement of Degree of Bachelor Of Engineering. This project report is submitted on the partial fulfilment of the requirement of the award of Degree of Bachelor Of Engineering.

Signature

SHARU RUBA S

SUBITHA G

VARSHAA S R

Place: Samayapuram

Date:

ACKNOWLEDGEMENT

It is with great pride that we express our gratitude and indebtness to our institution **"K RAMAKRISHNAN COLLEGE OF TECHNOLOGY"**, for providing us with the opportunity to do this project.

We are glad to credit honorable chairman **Dr. K RAMAKRISHNAN, B.E.,** for having provided for the facilities during the course of our study in college.

We would like to express our sincere thanks to our beloved Executive Director **Dr. S KUPPUSAMY, MBA, Ph.D.,** for forwarding our project and offering adequate duration to complete it.

We would like to thank **Dr. N VASUDEVAN, M.Tech., Ph.D.,** Principal, who gave opportunity to frame the project with full satisfaction.

We whole heartily thank **Dr. A DELPHIN CAROLINA RANI, M.E., Ph.D.,** Head of the Department, **COMPUTER SCIENCE AND ENGINEERING** for providing her support to pursue this project.

We express our deep and sincere gratitude and thanks to our project guide **Mrs. A THEMOZHI, M.E.,** Department of **COMPUTER SCIENCE AND ENGINEERING,** for his incalculable suggestions, creativity, assistance and patience which motivated us to carry our this project.

We render our sincere thanks to Course Coordinator and other staff members for providing valuable information during the course. We wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

ABSTRACT

The system is an advanced, automated solution designed to monitor and maintain toilet facilities efficiently by leveraging real-time data collection, cloud-based processing, and predictive analytics. Sensors track occupancy and gas levels, with the data processed by a microcontroller and transmitted to a cloud platform like Firebase for storage and further analysis. A linear regression model evaluates this data to predict maintenance needs, such as cleaning or addressing high gas levels, triggering automated alerts to maintenance personnel for timely responses. A dynamic feedback loop continuously refines the model's predictions, improving accuracy as the system adapts to changing conditions. Future enhancements could include integrating advanced machine learning models for better predictions, adding sensors for comprehensive environmental monitoring, incorporating IoT devices for automated ventilation adjustments, and optimizing the system for sustainability and scalability. This intelligent system offers a proactive, efficient, and user-focused approach to maintaining hygiene and safety in both public and private restroom facilities.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATION	FULL FORM
IoT	Internet of Things
ML	Machine Learning
AI	Artifical intelligence
HVAC	Heating Ventilation And Air Conditioning
SMS	Short Message Service
CO2	Carbon Dioxide
MQ	Gas Sensors
LED	Light Emitting Diode
API	Application Programming Interface
UI	User Interface
UX	User Experience
PIR	Passive Infrared Sensor

CHAPTER 1
INTRODUCTION

1.1 BACKGROUND

The smart toilet alert management and prediction system is an innovative approach to health monitoring that uses sensor technology and data analytics to deliver valuable insights into a person's health status. This system operates by integrating various sensors within the toilet to track and analyze a range of health indicators such as hydration levels, glucose and protein presence, blood in urine, and stool characteristics. The data collected through these sensors provides a continuous stream of information that can reveal important health patterns and potential risks.Central to this system is the capability to process this data in real-time and identify any anomalies that may require immediate attention. For example, if the system detects an abnormal concentration of glucose, it can alert the user to the possibility of blood sugar irregularities, potentially flagging early signs of diabetes. Similarly, changes in stool consistency or frequency can be tracked over time, providing insights into digestive health or signaling the onset of gastrointestinal issues. By issuing alerts, the system allows users to respond proactively, potentially reducing the need for frequent manual health checks or visits to a healthcare facility.

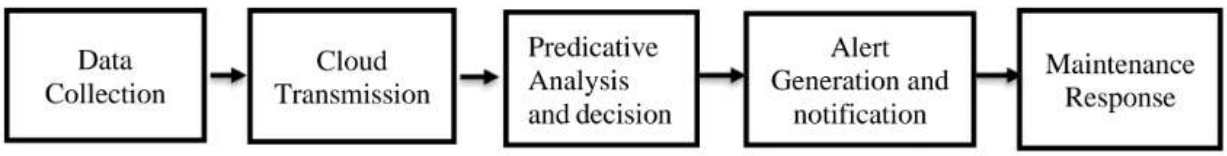


Figure 1.1 Flow control

1.2 OVERVIEW

The core function of the smart toilet system begins with data collection, facilitated by sensors embedded in restroom components. These sensors gather data on key metrics, including toilet and urinal usage, handwashing frequency, supply levels for consumables like soap and paper towels, and environmental factors like humidity and odor levels. Data on these factors is continuously transmitted to a centralized platform that analyzes it in real-time. Using this data, the system can detect when supplies are low, when a facility requires cleaning, or when maintenance issues are likely to occur. For instance, if a sensor detects that a soap dispenser is nearly empty, it automatically triggers an alert to the maintenance staff, allowing them to restock the supplies before they run out. Beyond real-time alerts, the smart toilet system also utilizes historical data for predictive analytics, allowing it to forecast future maintenance needs based on usage patterns. Over time, the system can identify patterns in restroom traffic and peak usage times. This predictive capability is beneficial for facilities that experience seasonal variations, such as airports, stadiums, and shopping malls, where demand can fluctuate dramatically. For instance, a stadium restroom might experience significant increases in usage during game days, which would be reflected in the system's data. The system would adjust its cleaning and restocking alerts accordingly, ensuring that the facility remains clean and well-stocked during periods of high demand. Furthermore, data collected by the smart toilet system can guide decision-making on resource allocation and maintenance schedules. Instead of a traditional fixed cleaning schedule, facility managers can adopt a demand-driven approach, dispatching cleaning staff and supplies based on real-time usage data. This approach not only enhances operational efficiency but also ensures that restrooms are consistently maintained to a high standard, improving the overall user experience.

1.3 PROBLEM STATEMENT

Public toilets need to be hygienic or they would affect the health of people because hygiene is directly connected to the badly maintained toilets. In an existing system, the focus is towards reducing water wastage on toilets, by the implementation of automatic flusher.

1.4 OBJECTIVE

To provide a technical solution to monitor the quality of the lavatories, so as to maintain it as and when required. Our aim is to provide an IoT based solution towards monitoring the quality of the lavatories, on the go while alerting the concerned one to carry out the necessary action using android application. Data analysis of lavatory gives us the solution to prevent the future causes. The system is tested for its working and it is observed that implementing them in real time would help in building a healthy chain of lavatories.

1.5 IMPLICATION

A smart toilet system is envisioned as a sophisticated device that seamlessly integrates advanced technology to enhance user experience, optimize resource consumption, and proactively address potential issues. By harnessing the power of data analysis, this system aims to revolutionize bathroom hygiene and efficiency. Additionally, the system can generate scheduled notifications for routine maintenance reminders and usage reports. It optimizes resource consumption, reduces maintenance costs, and promotes a healthier and more sustainable bathroom experience.

CHAPTER 2

LITERATURE SURVEY

TITLE : IoT smart restroom solution for your convenience and well-being

AUTHORS : Andriy Kornatskyy, Myroslav Opyr, Volodymyr

YEAR 2023

This paper discusses the implementation of an IoT-based smart restroom solution aimed at enhancing convenience and well-being. Through sensors and real-time monitoring, the system tracks restroom occupancy, cleanliness, supply levels, and overall user experience, optimizing restroom management and user satisfaction. Odor and air quality sensors help monitor the restroom environment to ensure users are not exposed to harmful substances, thus maintaining a healthy atmosphere.

TITLE : IoT in facility management

AUTHORS : Nazly Atta

YEAR : 2021

The paper likely starts by defining IoT and its role in facility management. IoT refers to the network of interconnected devices that can collect, share, and analyze data. The paper may conclude with insights on the future of IoT in facility management, possibly touching on AI integration, more advanced predictive analytics, and the role of 5G in enhancing IoT connectivity. This blogspot focus on building management systems. New breed of sensors is helping to monitor various existing systems like Fire Fighting & Alarm Systems, Intrusion Detection Systems. It can also monitor Occupancy of office desks, meeting rooms, cabins etc, Indoor Air Quality (parameters like Temp, Humidity, VOC, CO2 gasses), Vibes of the place (Lux level, Noise level).

TITLE : Exploratory Data Analysis(beginner), Univariate, Bivariateand Multivariate-Habberman database

AUTHORS :John Tukey,Habberman, purnasai gudikandula

YEAR :1977

https://medium.com/swlh/an-open-architecture-iot-washroom-e71af755f96d

This paper focus on EDA using Univariate visualizations, bivariate visualization, and multivariate visualization and some plots. This blog was helpful to get an idea of using EDAYou want to perform Exploratory Data Analysis (EDA) on the Haberman dataset, but with a focus on IoT toilet data. You want to perform Exploratory Data Analysis (EDA) on the Haberman dataset, but with a focus on IoT toilet data.
Principal Component Analysis (PCA) to reduce dimensionality and identify patterns in the data .Cluster analysis (k-means or hierarchical clustering) to identify groups of toilets with similar usage patterns. Heatmaps or correlation matrices to visualize relationships between vavariables

TITLE : Exploratory Data Analysis (EDA) in Python

AUTHORS : Sami D. Alaruri

YEAR :2022

This paper gave an idea about how to explore a Data set and perform Exploratory data analysis in python.Exploratory Data Analysis (EDA) is a method for inspecting, visualizing, investigating, modifying and analyzing an obesity dataset before performing detailed analysis and modeling to the obesity dataset. In this tutorial some of the EDA techniques using Python are given. These techniques include summary statistics, missing data analysis, outliers' detection, duplicate data analysis, correlation analysis, and data visualization using plots like box plots, scatter plots, bar plots and histograms.

TITLE : A Beginner's Guide to Exploratory Data Analysis with Linear Regression.
AUTHOR : Kan Nishida
YEAR 2018

The paper, A Beginner's Guide to Exploratory Data Analysis with Linear Regression, serves as an introductory guide for readers interested in understanding how to explore and interpret data using linear regression as a foundational statistical method. It combines concepts from Exploratory Data Analysis (EDA) with practical insights on linear regression, creating a clear and accessible path for beginners to start analyzing data and extracting meaningful insights.

TITLE : Scheduling and Predictive Maintenance for Smart Toilet
AUTHOR: Amar Lokman, R Kanesaraj Ramasamy
YEAR 2023

Modern society needs bathrooms. Poor sanitation is caused by worn-out appliances and expensive cleaning. The technique also requires an inexpensive, dependable sensor. This study had three goals. Creating an IoT administration platform is the main goal. Literature evaluations assess the merits and downsides of existing systems. Second, we suggest predictive maintenance to assist predict bathroom equipment breakdowns. Finally, a scheduling algorithm was used to determine how many janitors to hire. We'll measure the model's effectiveness and make future recommendations. Infrared, temperature and humidity sensors create an IoT bathroom.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

There are several existing systems and technologies for smart toilet alert generation

which are implemented with multiple algorithm

WATER FLOW SENSOR:

- A water flow sensor in a toilet alert system can be designed to monitor and detect abnormal water flow, which may indicate leaks, overflows, or other plumbing issues.

AUTOMATIC FLUSHER:

- An automatic flusher with an alert-generating system can be designed for restrooms to enhance hygiene, convenience, and maintenance.

MOTION SENSOR:

- A motion sensor-based alert system in a toilet can be used to notify relevant personnel (such as cleaning staff, management, or maintenance) about activity or other specific events within the restroom. Such a system can improve hygiene management, detect prolonged occupancy, or alert staff for issues requiring attention.

BASIC CONTROL LOGIC:

- A basic toilet alert system, control logic can be implemented to monitor the occupancy, cleanliness, or other conditions of the restroom. Here's a general breakdown of how a basic control logic could work for a toilet alert system
    - No data analysis
    - No toxic value prediction
    - No mobile app

FLUSH OPERATION:

- A flush operation in a toilet alert generating system would typically refer to a sequence where a sensor or series of components detect that the toilet has been flushed.

3.2 PROPOSED SYSTEM

A proposed toilet alert system that automates notifications for maintenance based on gas levels and occupancy detection. Here's a breakdown of each component and process:

TOILET SENSOR

- **PIR Sensor:** Detects human presence in the toilet.
- **MQ135 Sensor:** Measures gas levels (presumably for detecting harmful or unpleasant gases that might indicate the need for cleaning).

MICROCONTROLLER

- Collects data from the sensors. It monitors the toilet status (occupancy and gas level) and sends it to the cloud.

OLED DISPLAY

- Located near the toilet, possibly displaying occupancy or maintenance status.

FIREBASE (Cloud Service):

- Receives the toilet status data from the microcontroller, allowing for remote monitoring and storage.

DECISION PROCESS

- The system checks if the gas level exceeds a certain threshold.
- **If the gas level is above the threshold**: An alert is generated, and a task is deployed to notify the concerned personnel for cleaning or maintenance.
- **If the gas level is below the threshold**: No action is taken.

3.3 DIAGRAM FOR PROPOSED SYSTEM

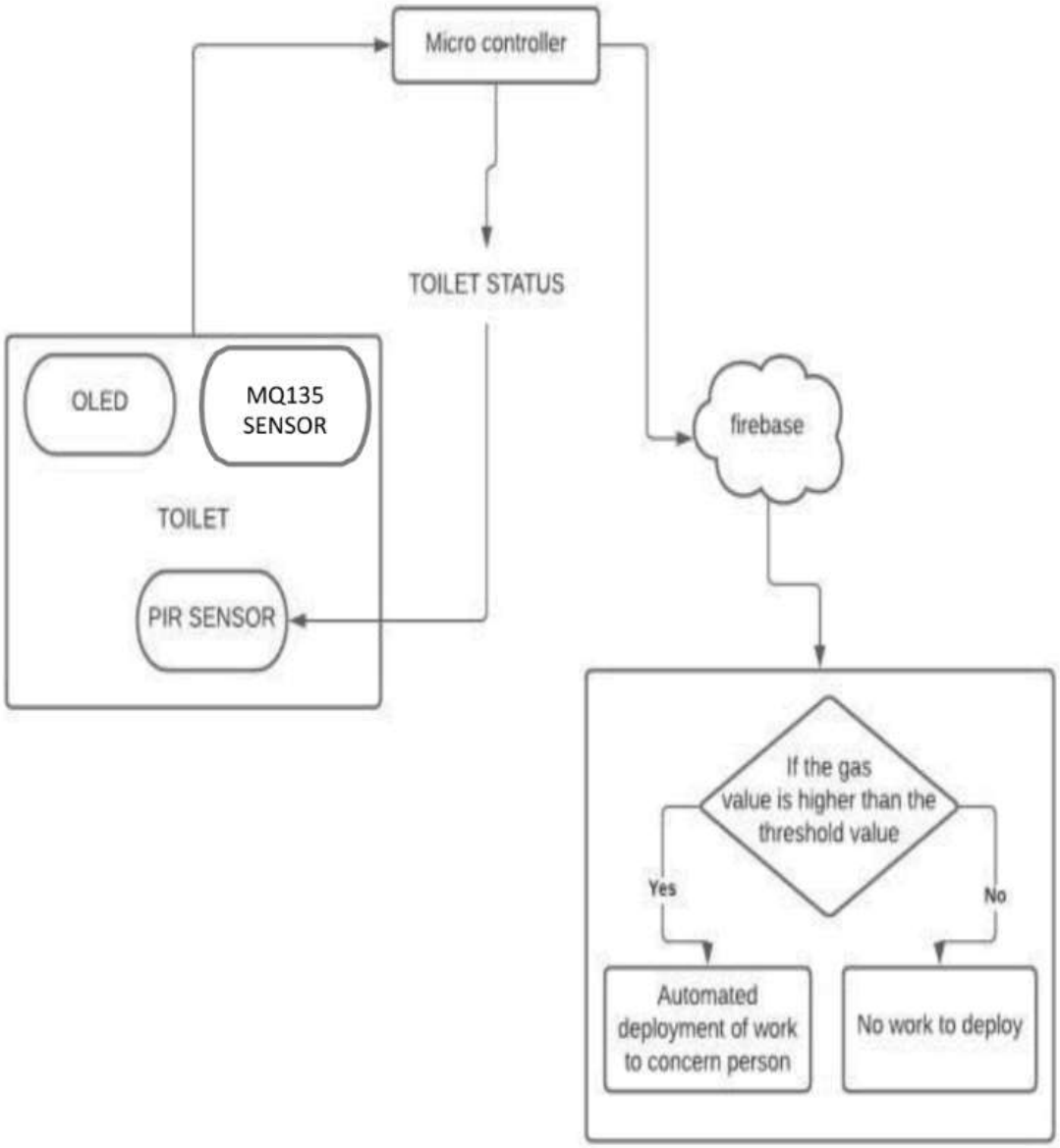


Figure 3.1: Work Flow Diagram

3.4 FLOWCHART

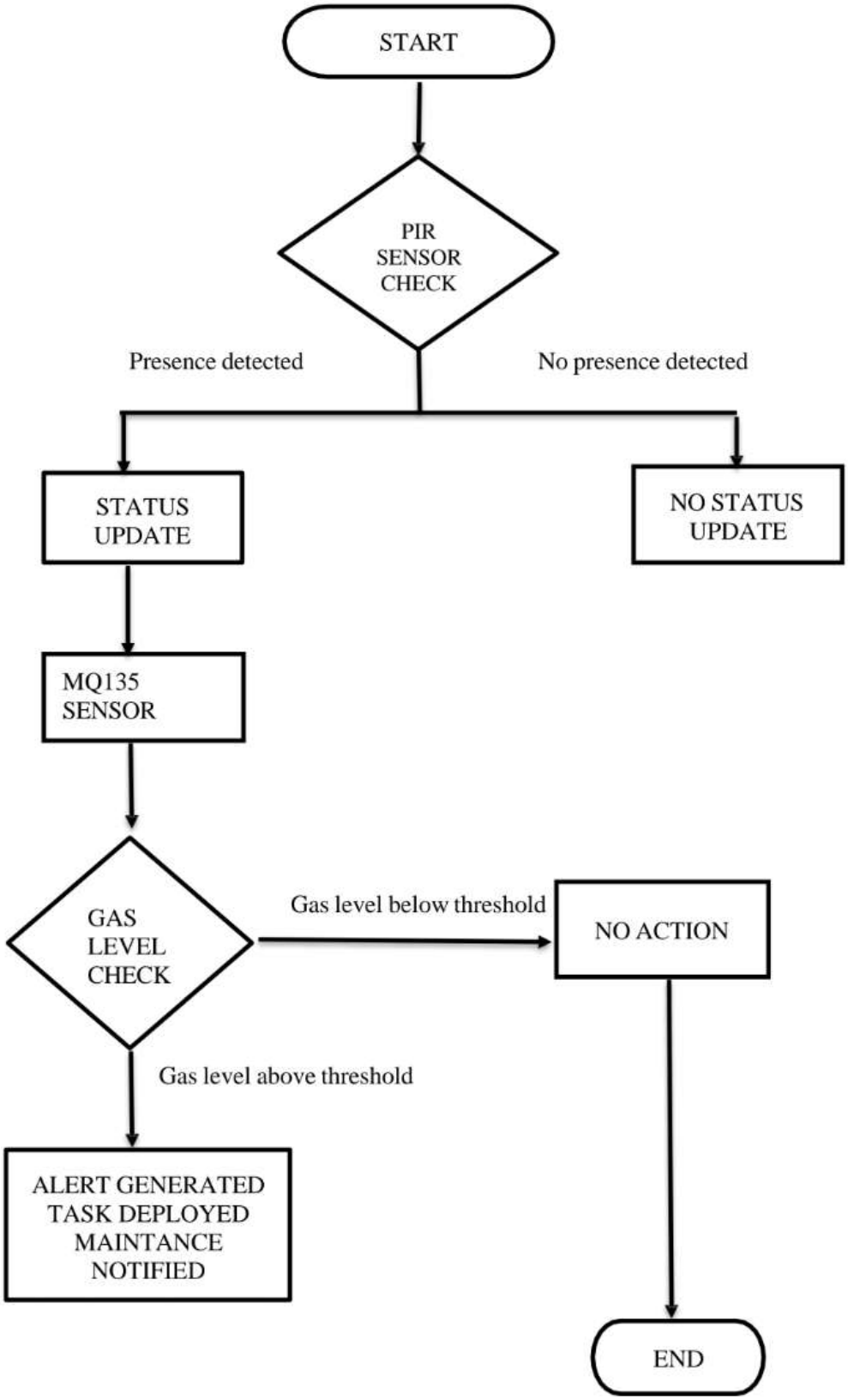


Figure 3.2: Flow Chart

3.5 PROCESS CYCLE

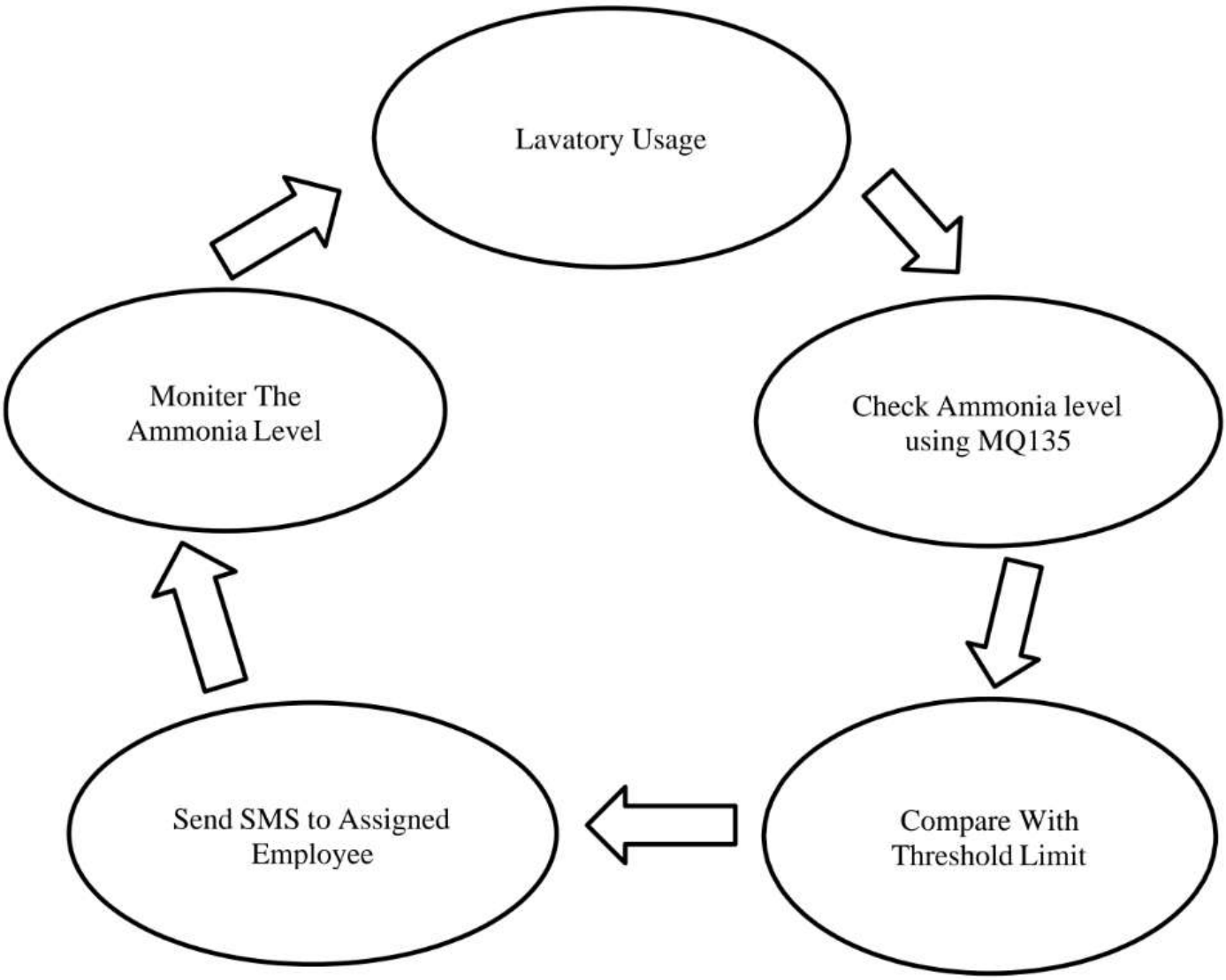


Figure 3.3: Life Cycle of the Process

3.6 ACTIVITY DIAGRAM

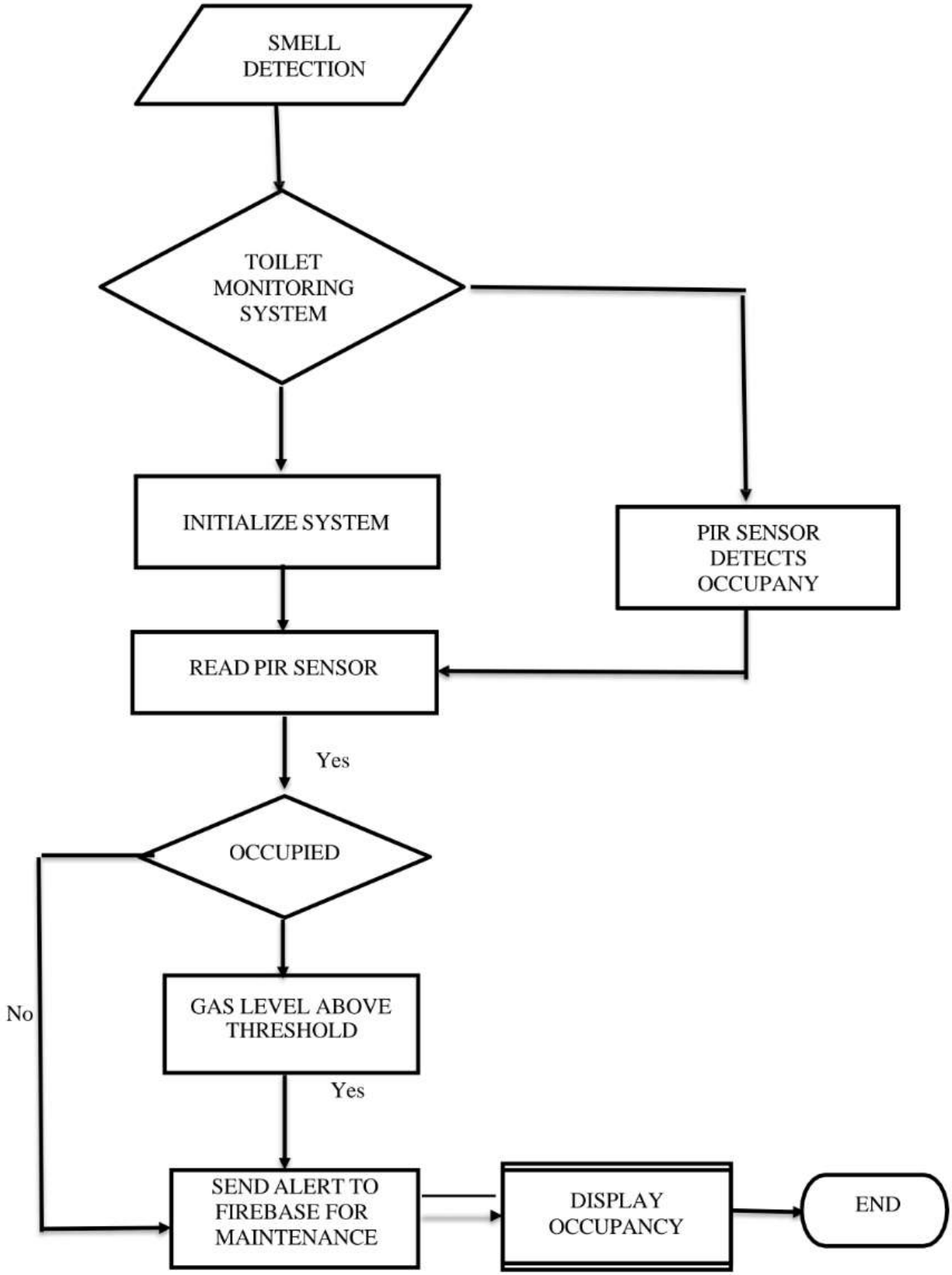


Figure 3.4: Action Sequence Structure for alert generating system

CHAPTER 4
MODULES

4.1 MODULE DESCRIPTION

- Sensor Module
- DataAnalysis Module
- Linear regression Algorithm
- Splitting Date into month,year,day,time,season,time
- Application module
- Thershold analyzing Module

4.1.1 SENSOR MODULE

The Sensor Module in a toilet alert generating system is responsible for monitoring the occupancy and air quality of the restroom. It consists of a PIR (Passive Infrared) sensor, which detects human presence to determine if the toilet is occupied, and an MQ135 gas sensor, which measures gas levels in the environment. The MQ135 sensor is particularly useful for detecting the presence of unpleasant or potentially harmful gases, providing an indication of the restroom's air quality.

The data collected by these sensors is sent to a microcontroller for processing. When the gas level exceeds a pre-set threshold, the microcontroller triggers an alert, indicating the need for cleaning or maintenance. The sensor module operates continuously, ensuring real-time monitoring of the restroom's conditions and enabling prompt responses based on occupancy and air quality status.

- PIR SENSOR
- MQ135 SENSOR

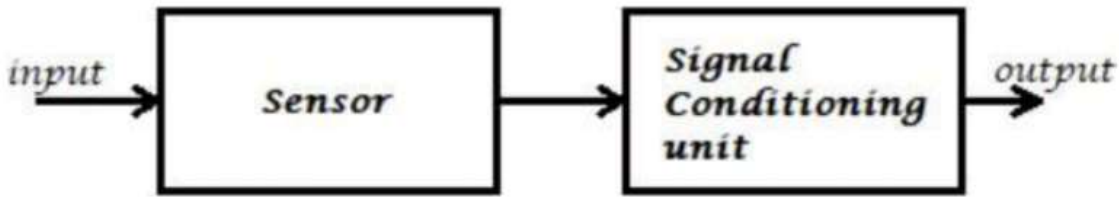


Figure 4.1: Flow of Sensor

4.1.1.1 PIR SENSOR

In an alert-generating system, the PIR (Passive Infrared) sensor is used to detect human presence within the restroom. It senses infrared radiation emitted by humans, identifying when someone enters or occupies the toilet area. When motion is detected, the PIR sensor sends a signal to the system's microcontroller, confirming occupancy.The microcontroller uses this occupancy data along with air quality measurements from other sensors (like the MQ135 gas sensor) to assess the restroom's conditions. If the PIR sensor detects that the toilet is in use, the system monitors air quality more closely to determine if maintenance is needed. This sensor ensures the system operates efficiently, responding only when the restroom is occupied, which conserves energy and optimizes monitoring.

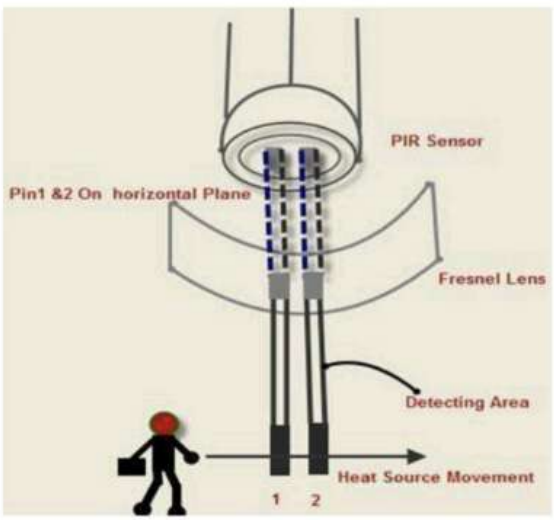


Figure 4.2:PIR Sensor

The PIR (Passive Infrared) Sensor is a widely used electronic component designed to detect motion or presence of humans, animals, or objects. This sensor operates on the principle of infrared radiation detection, where the sensing element, typically a pyroelectric sensor, detects changes in infrared radiation levels emitted by objects within its detection range. The PIR Sensor is extensively used in various applications, including security systems, automation, lighting control, and public toilet hygiene monitoring.

The PIR Sensor features a high sensitivity and fast response time, detecting motion within a range of 6-7 meters (20-23 feet) and responding within 100-300 milliseconds. This sensor operates at a voltage of 5V and consumes a relatively low current of 1-2 mA. The operating temperature range is -20°C to 80°C (-4°F to 176°F), and it can withstand humidity levels between 20% and 90% RH.

In terms of pin configuration, the PIR Sensor typically has three pins: VCC (Power), GND (Ground), and OUT (Digital output signal). The OUT pin provides a digital signal (0 or 1) indicating the presence or absence of motion. This digital output allows for easy integration with microcontrollers like Arduino, ESP32, or Raspberry Pi.To integrate the PIR Sensor with a microcontroller, simply connect the OUT pin to a digital input pin and use the microcontroller's digital input/output (GPIO) functions to read the sensor output. This digital output enables straightforward motion detection and monitoring.

The PIR Sensor offers several advantages, including high sensitivity and fast response time, low power consumption, compact size, and easy integration. These features make it an ideal choice for various applications, particularly in public toilet hygiene monitoring.In public toilet hygiene monitoring, the PIR Sensor detects occupancy and triggers automated flushing, lighting, or ventilation systems. This information can be used to maintain a clean and healthy environment, reducing the risk of diseases and unpleasant odors.Furthermore, the PIR Sensor's compact size and low power consumption make it suitable for battery-powered applications or remote monitoring systems.The PIR Sensor features a simple pin configuration, consisting of VCC (Power), GND (Ground), and OUT (Digital output signal), producing a digital signal (0 or 1) indicating presence or absence of motion. Interfacing is straightforward, requiring connection of the

OUT pin to a microcontroller's digital input pin and utilizing the microcontroller's GPIO functions to read sensor output.

The PIR Sensor boasts several advantages, including high sensitivity and fast response time, low power consumption, compact size, and easy integration, as well as digital output for straightforward motion detection. Ideal applications include security systems, automation, lighting control, public toilet hygiene monitoring, industrial safety monitoring, and environmental monitoring. By leveraging the PIR Sensor's capabilities, developers can create efficient and effective motion detection systems for various applications.

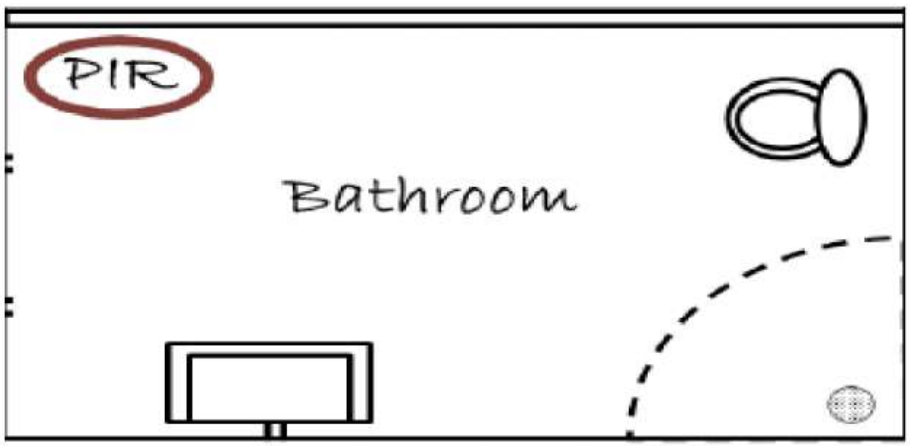


Figure 4.3: Location of PIR

PIR SENSOR PINOUT:

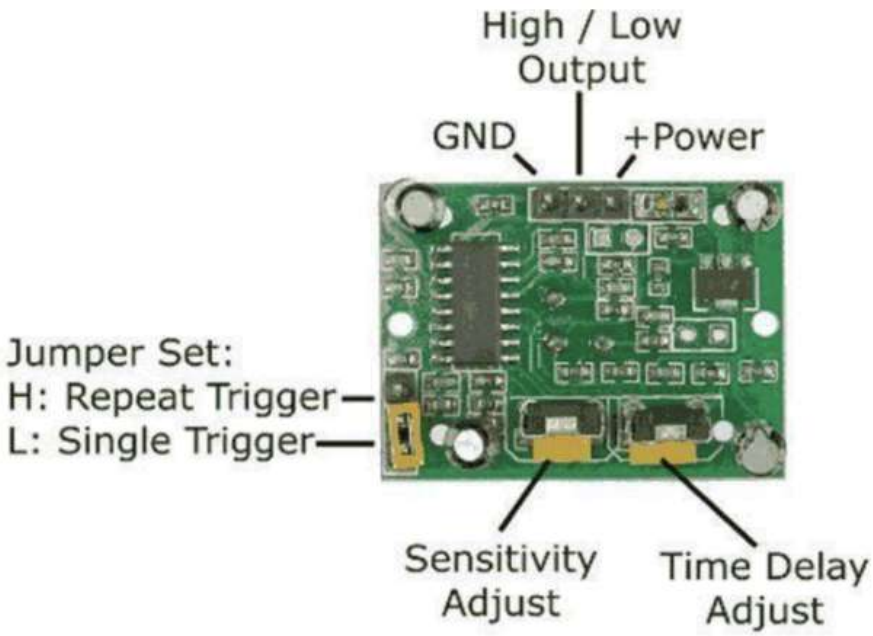


Figure 4.4: Sensor pinout

4.1.1.2 MQ135 SENSOR:

The MQ135 gas sensor is commonly used for air quality monitoring, as it can detect various gases such as ammonia, sulfur, benzene, smoke, and other harmful gases. In a toilet alert system, the MQ135 could be set up to detect high levels of ammonia (NH3) or hydrogen sulfide (H2S), which are common in waste environments, and trigger an alert when concentrations exceed a specific threshold. Here's a general overview of how you could implement a toilet alert system with the MQ135 sensor.

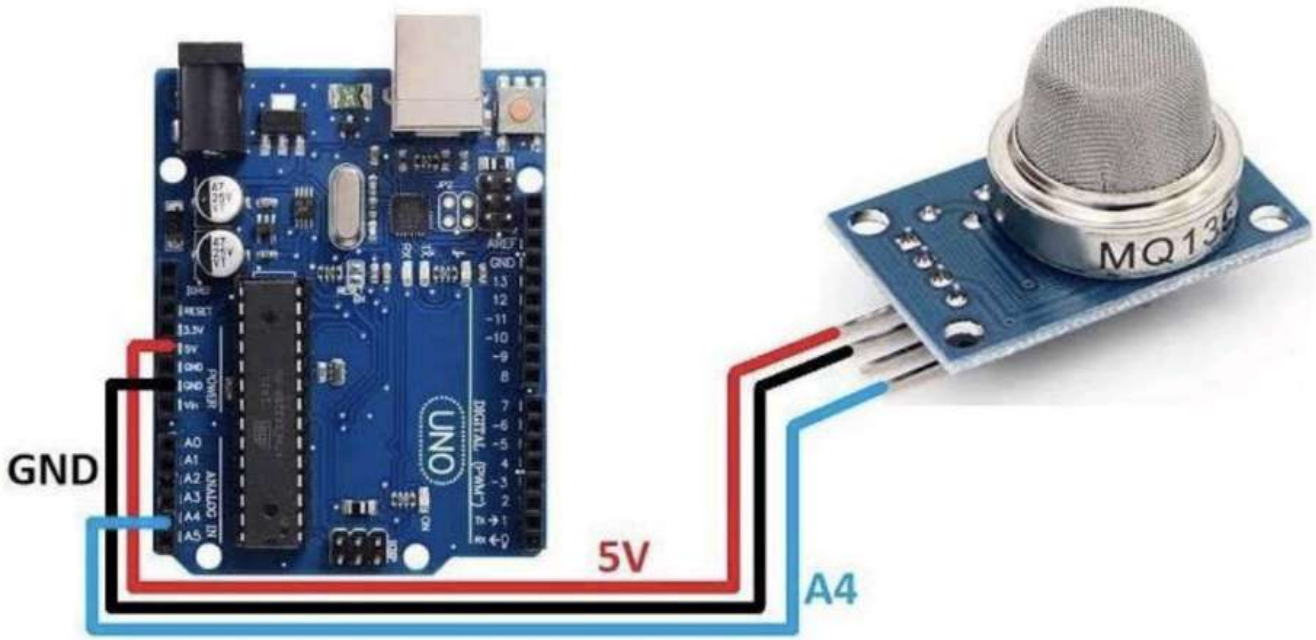


Figure 4.5: MQ135 Sensor

The MQ135 sensor is a highly sensitive and reliable gas sensor designed to detect harmful gases in public toilets, ensuring timely maintenance and minimizing health risks. Maintaining hygiene in public toilets is crucial for preventing the spread of diseases.The MQ135 sensor is a metal oxide semiconductor (MOS) gas sensor that detects gases such as ammonia (NH3), nitrogen oxide (NOx), carbon monoxide (CO), hydrogen sulfide (H2S), methane (CH4), and liquefied petroleum gas (LPG). Its key features include high sensitivity and fast response time, low power consumption, compact size, easy integration, analog output for precise measurement, and simple pin configuration.

The technical specifications of the MQ135 sensor include a detection range of 100-1000 ppm for ammonia, 10-100 ppm for nitrogen oxide, 10-1000 ppm for carbon monoxide, 1-100 ppm for hydrogen sulfide, 300-10,000 ppm for methane, and 100-10,000 ppm for liquefied petroleum gas. Its sensitivity ranges from 0.1-1.0 V/ppm for ammonia, 0.01-0.1 V/ppm for nitrogen oxide, 0.01-1.0 V/ppm for carbon monoxide, 0.001-0.1 V/ppm for hydrogen sulfide, 0.1-1.0 V/ppm for methane, and 0.05-0.5 V/ppm for liquefied petroleum gas.

The MQ135 sensor has a response time of ≤10 seconds and a recovery time of ≤30 seconds. It operates at 5V and consumes 50-100 mA. Its operating temperature range is -20°C to 50°C, and it can withstand humidity levels between 20% and 90% RH.The MQ135 sensor's pin configuration consists of VCC (Power), GND (Ground), and OUT (Analog output signal). Interfacing with the sensor involves connecting the OUT pin to a microcontroller's analog input pin and utilizing the microcontroller's ADC to read sensor output.

The MQ135 sensor is suitable for various applications, including gas leak detection systems, industrial safety monitoring, environmental monitoring, and public toilet hygiene monitoring. Its advantages include high sensitivity and fast response time, low power consumption, compact size, easy integration, and analog output for precise measurement.However, the MQ135 sensor requires calibration and is sensitive to temperature and humidity. It may also detect false positives. Calibration involves using known gas concentrations and adjusting sensor sensitivity using a potentiometer.

The MQ135 sensor can be integrated with microcontrollers like Arduino or Raspberry Pi. Connection involves linking the MQ135 sensor to the microcontroller's analog input pin and using the microcontroller's programming language to process sensor data.In conclusion, the MQ135 sensor is a reliable and efficient gas sensor designed for detecting harmful gases in public toilets. Its high sensitivity, fast response time, and low power consumption make it an ideal choice for public toilet hygiene monitoring and various other gas detection applications.

4.1.2 DATA ANALYSIS MODULE

The Data Analysis Module in a toilet alert-generating system is designed to process and analyze data collected from various sensors, delivering insights that optimize facility management, improve cleanliness, and enhance the user experience. This module aggregates data from all connected sensors, including occupancy counters, odor and air quality sensors (such as the MQ135), supply level sensors, and usage pattern trackers, to provide a comprehensive overview of lavatory conditions over time. By analyzing trends and patterns, the module enables facility managers to make informed, data-driven decisions on cleaning schedules, supply replenishment, and maintenance needs.

As data flows in from multiple sources, the module performs both real-time and historical analysis. In real-time, it monitors incoming data to identify unusual spikes or conditions that might warrant immediate attention, such as a sudden drop in supply levels or an unexpected increase in odor intensity. This capability allows the system to adapt dynamically, issuing immediate alerts if conditions exceed defined thresholds. Beyond real-time monitoring, the Data Analysis Module also looks at long-term patterns to assess usage trends, identifying peak times, high-traffic periods, and recurring issues.

Using these insights, the module can help optimize staffing and scheduling. For instance, if analysis shows that certain restrooms experience high occupancy during specific hours, cleaning schedules can be adjusted accordingly to ensure facilities are well-maintained when most needed. Additionally, data analysis reveals patterns in supply usage, enabling more accurate predictions for replenishment and reducing the risk of supply shortages. This predictive capability also improves resource allocation, ensuring that staff and supplies are directed to high-need areas.

Furthermore, the Data Analysis Module supports predictive maintenance by identifying trends in equipment wear and performance. For example, if sensor data reveals a gradual increase in air quality issues despite regular cleaning, the module may suggest a deeper inspection of ventilation systems. This proactive approach helps prevent larger issues, reducing downtime and repair costs while maintaining a pleasant environment for users.

- Univariate analysis
- Multivariate analysis

4.1.2.1 UNIVARIATE ANALYSIS

Univariate analysis is a statistical method that focuses on the examination of a single variable within a dataset, providing insights into its individual behavior, distribution, and patterns. In the context of a toilet alert-generating system, where both the MQ135 sensor (for air quality) and the PIR sensor (for occupancy detection) are used, univariate analysis can offer valuable insights into each sensor's data independently, helping facility managers optimize their responses and maintenance schedules. By analyzing each variable separately, univariate analysis allows us to understand the trends, variations, and anomalies that exist within each sensor's data, making it easier to set appropriate thresholds, detect issues early, and improve the overall management of restroom conditions.

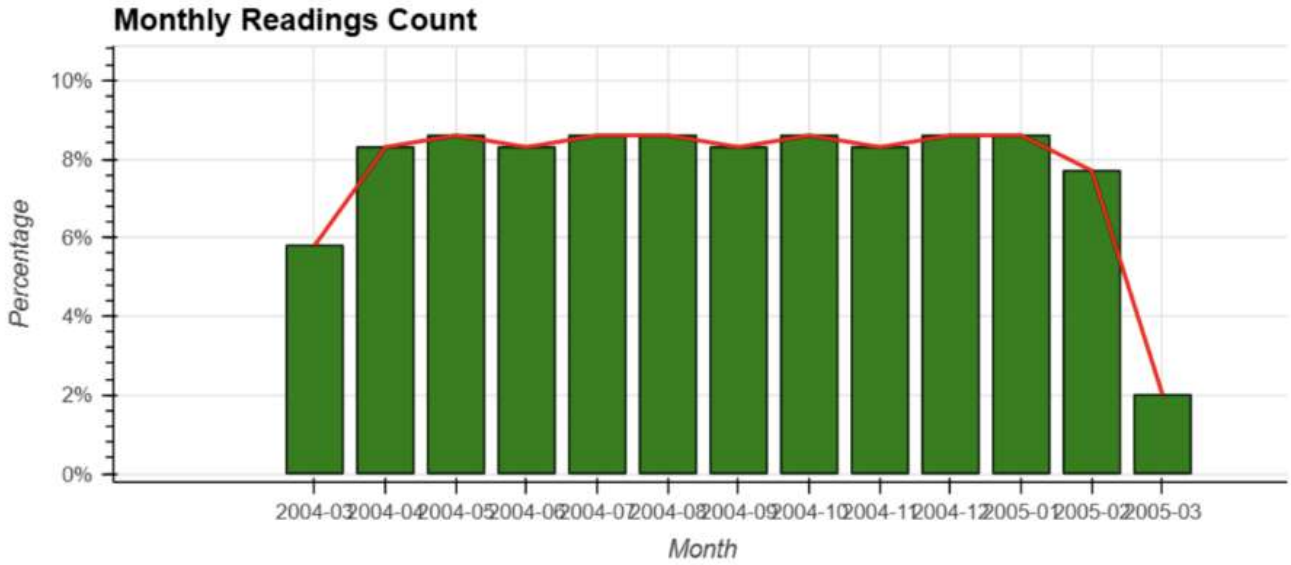


Figure 4.6: Count Analysis

When applied to the MQ135 sensor, which monitors air quality by detecting gases like ammonia, carbon dioxide, and other pollutants, univariate analysis provides a comprehensive view of the air quality levels over time. This analysis typically focuses on measuring the central tendency and dispersion of the sensor data, helping to identify the typical air quality levels and how they fluctuate. For example, the mean value gives the average concentration of pollutants detected by the sensor during a specific time frame, indicating the typical air quality in the restroom. Additionally, the variance or standard deviation helps assess how stable the air quality is, revealing if there are frequent spikes or drops in sensor readings that may require attention. Through univariate analysis, it becomes possible to determine if the air quality exceeds predefined thresholds, triggering alerts to the maintenance staff.

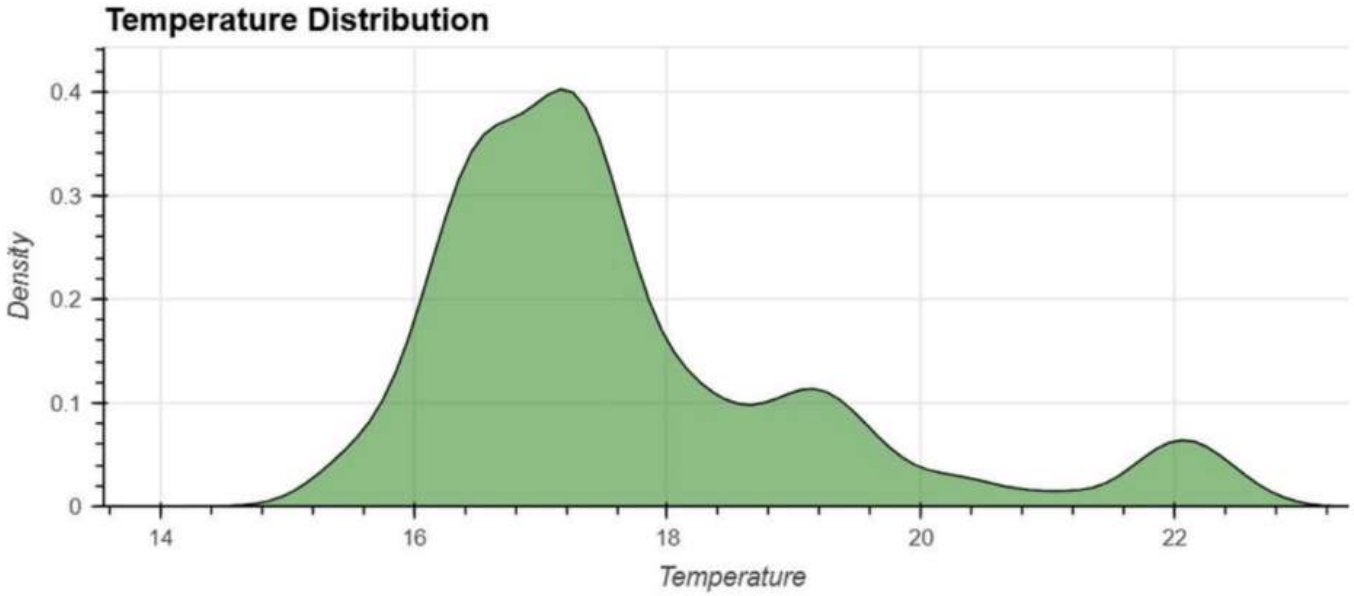


Figure 4.7: Temperature Distribution

Through univariate analysis, the behavior of each sensor is isolated and analyzed in detail, providing a deeper understanding of how individual factors—such as air quality or occupancy—affect the overall restroom environment. This approach allows for the fine-tuning of sensor thresholds to ensure that alerts are triggered only when necessary, reducing unnecessary notifications and ensuring that maintenance staff focus on critical issues. Additionally, univariate analysis helps optimize resource allocation by identifying peak usage times and recurring issues, enabling facility managers to adjust cleaning and maintenance schedules more effectively.Ultimately, univariate analysis serves as a foundational tool for improving the performance of the toilet alert-generating system, helping to maintain a clean, well-managed environment that meets the needs of users while enhancing operational efficiency.

However, univariate analysis has limitations, including oversimplification of complex relationships, ignoring interactions between variables, and limited generalizability. To mitigate these limitations, researchers should explore data visually, check assumptions like normality and homoscedasticity, handle missing values, and validate results with additional analysis.

4.1.2.2 MULTIVARIATE ANALYSIS

Multivariate analysis is a statistical technique used to examine the relationships and interactions between multiple variables within a dataset. Unlike univariate analysis, which focuses on a single variable, multivariate analysis allows for the simultaneous study of multiple factors and how they affect each other. In the context of a toilet alert-generating system, multivariate analysis is particularly useful because it can consider data from multiple sensors, such as the MQ135 air quality sensor and the PIR occupancy sensor, at once. By examining how these variables work together, multivariate analysis provides a more comprehensive understanding of lavatory conditions and enables more informed decision-making for maintenance and management.

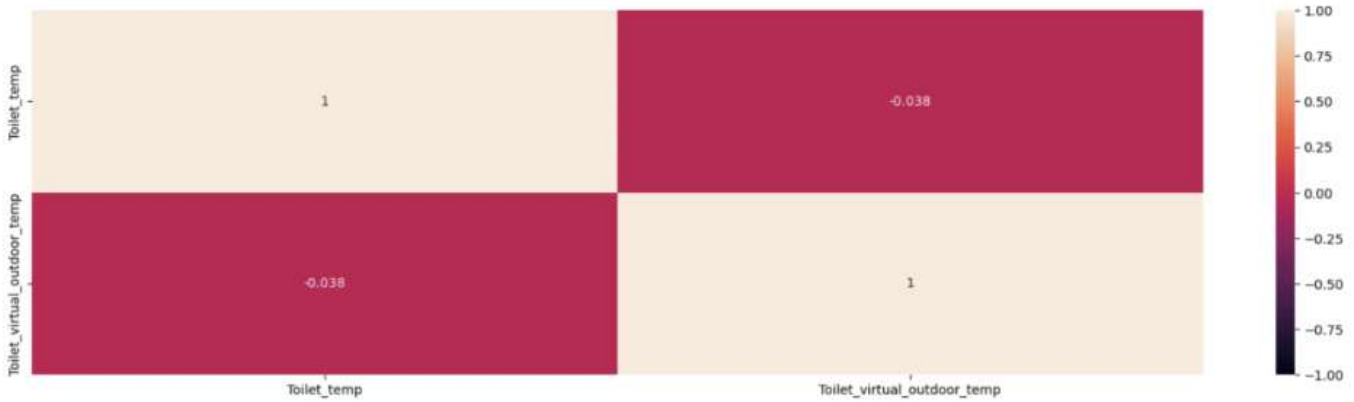


Figure 4.8: Temperature Multivariate Analysis

Moreover, multivariate analysis can help optimize the response times for alerts in a toilet management system. For instance, instead of analyzing air quality or occupancy levels independently, the system can consider how the two variables together indicate when maintenance or cleaning is necessary. If the MQ135 sensor detects a sharp rise in pollution levels while the PIR sensor records an increase in occupancy, the system can trigger an alert that is more reliable and relevant, ensuring that maintenance staff are notified only when both conditions align. This kind of integrated analysis makes the system more efficient, as it reduces false alarms triggered by a single variable and provides more accurate, context-specific information for management.By looking at multiple variables together, the system can account for complex patterns, such as how the increase in occupancy might directly impact the air quality, or how other factors like temperature or time of day influence these readings. Multivariate analysis is especially useful in situations where the relationships between variables are not immediately obvious, such as when external factors, like ventilation systems or restroom layout, may affect the readings of the sensors.

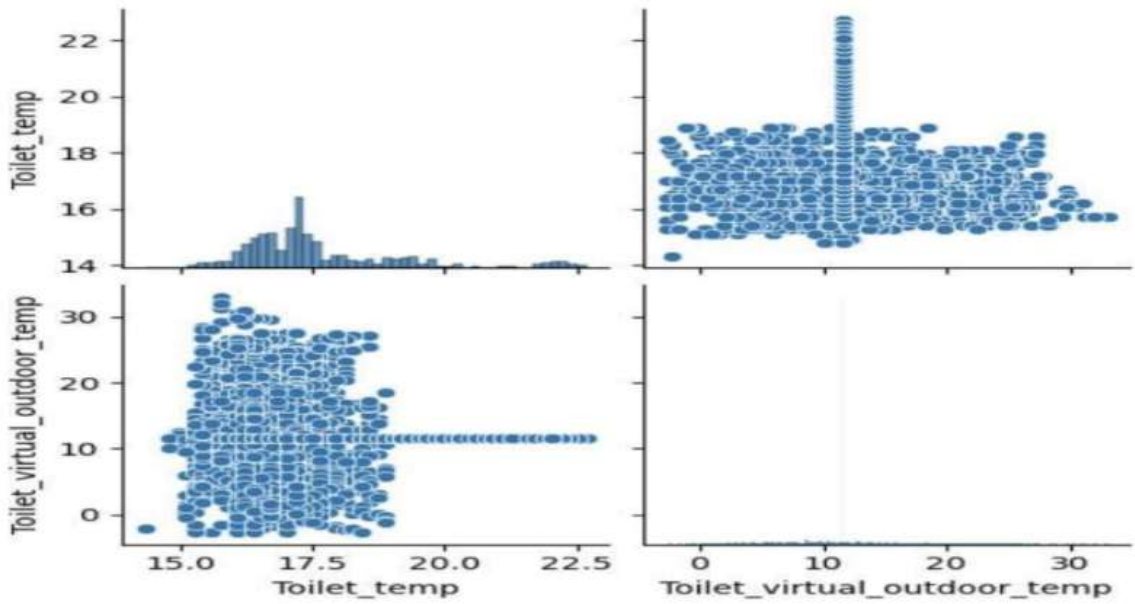


Figure 4.9: Virtual outdoor Temp

Additionally, multivariate analysis can be used to predict future conditions by examining historical data. By analyzing patterns across multiple variables over time, the system can make predictions about when certain thresholds are likely to be exceeded, such as when air quality will degrade or occupancy levels will spike. This predictive capability allows the system to act proactively, alerting staff ahead of time to prepare for increased demand or necessary maintenance actions. In a larger context, multivariate analysis also helps facility managers identify trends that might not be evident when considering variables in isolation. For example, it could uncover correlations between environmental conditions, sensor data, and user behavior, offering valuable insights for improving resource allocation, such as determining optimal cleaning times, adjusting air filtration systems, or optimizing supply management.

Multivariate analysis provides a deeper and more integrated understanding of restroom conditions by examining how multiple variables interact with each other. It enhances the toilet alert-generating system by improving the accuracy of alerts, offering predictive insights, and supporting more efficient and effective decision-making. By analyzing multiple sensor readings together, multivariate analysis enables a more nuanced approach to facility management, leading to better maintenance scheduling, improved air quality, and a more responsive system overall. This integrated approach makes it possible to address issues more holistically, improving the user experience and optimizing operational efficiency.

4.1.3 LINEAR REGRESSION ALGORITHM

The Linear Regression model predicts unknown gas sensor values in public toilets, ensuring timely maintenance and minimizing health risks. Maintaining hygiene in public toilets is crucial for preventing the spread of diseases. Poorly maintained toilets can lead to unpleasant odors, unhealthy environments, and compromised air quality.The Linear Regression model uses the equation $y = \beta0 + \beta1x + \varepsilon$, where y is the predicted gas sensor value, x is the input feature (time), β0 is the intercept, β1 is the slope coefficient, and ε is the error term. To obtain the slope (m) and intercept © values, the model utilizes get_variance, get_covariance, and get_mean functions.

The model workflow involves data collection, preprocessing, training, testing, and prediction. The input consists of test and train data, including historical gas sensor values. The output includes month, year, date, day, time, season, and predicted time.The Linear Regression model offers several advantages, including accurate predictions, real-time monitoring, efficient maintenance scheduling, and improved hygiene and health safety. However, it also has limitations, such as assuming linearity, sensitivity to outliers, and multicollinearity issues.

The model has various real-world applications, including public toilet maintenance, air quality monitoring, industrial process control, and environmental monitoring. By implementing this model, public toilet maintenance can be optimized, reducing health risks and promoting hygiene.The technical specifications include inputting gas sensor values (test and train data), outputting predicted gas sensor values (month, year, date, day, time, season, time), using the Linear Regression algorithm, evaluating metrics (MSE, MAE, R-squared), and utilizing the scikit-learn library.

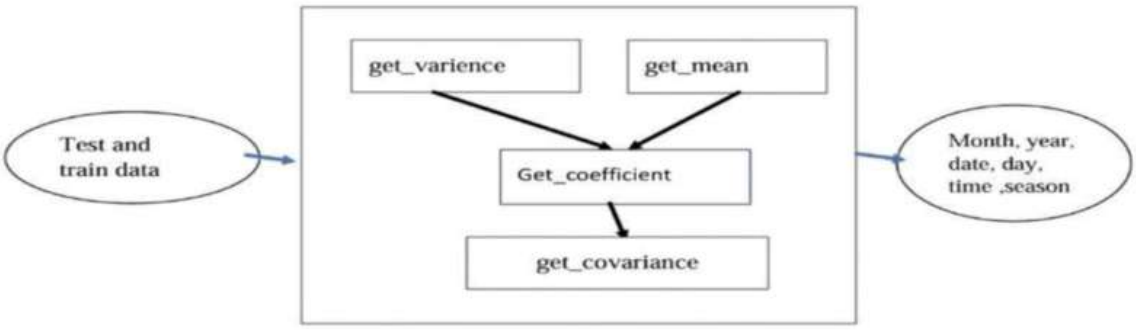


Figure 4.10:Linear Regression

Linear Regression model is compatible with various gas sensor modules and requires Python 3.x, scikit-learn library, and NumPy library.The Linear Regression model predicts unknown gas sensor values in public toilets, ensuring timely maintenance and minimizing health risks. Maintaining hygiene in public toilets is crucial for preventing the spread of diseases. Poorly maintained toilets can lead to unpleasant odors, unhealthy environments, and compromised air quality. The Linear Regression model uses the equation $y = \beta0 + \beta1x + \varepsilon$, where y is the predicted gas sensor value, x is the input feature (time), $\beta0$ is the intercept, $\beta1$ is the slope coefficient, and ε is the error term.To obtain the slope (m) and intercept (c) values, the model utilizes get_variance, get_covariance, and get_mean functions. The model workflow involves data collection, preprocessing, training, testing, and prediction. The input consists of test and train data, including historical gas sensor values. The output includes month, year, date, day, time, season, and predicted time.

The Linear Regression model offers several advantages, including accurate predictions, real-time monitoring, efficient maintenance scheduling, and improved hygiene and health safety. However, it also has limitations, such as assuming linearity, sensitivity to outliers, and multicollinearity issues. The model has various real-world applications, including public toilet maintenance, air quality monitoring, industrial process control, and environmental monitoring. By implementing this model, public toilet maintenance can be optimized, reducing health risks and promoting hygiene.

The technical specifications include inputting gas sensor values (test and train data), outputting predicted gas sensor values (month, year, date, day, time, season, time), using the Linear Regression algorithm, evaluating metrics (MSE, MAE, R-squared), and utilizing the scikit-learn library. The Linear Regression model is compatible with various gas sensor modules and requires Python 3.x, scikit-learn library, and NumPy library. A Python implementation example includes importing necessary libraries, loading data, splitting data, creating and training the model, making predictions, and evaluating the model using mean squared error (MSE).

4.1.4 SPLITTING DATE INTO MONTH, YEAR ,TIME

In an alert-generating system, splitting date information into its individual components—such as month, year, and time—is a crucial process. This breakdown allows the system to manage, 26 analyse, and trigger alerts based on specific conditions, such as detecting anomalies in a particular month or comparing data across years. By separating these elements, the system gains flexibility in sorting, filtering, and setting up conditional triggers based on different temporal criteria.The date information typically comes in a standard format (e.g., "YYYY-MM-DD HH:MM:SS"), and the process begins by parsing this format. Programming languages like Python, Java, or SQL often have libraries and functions that can handle these date manipulations efficiently. For instance, in Python, the datetime module can parse a date string and directly access individual date components, while in SQL, functions like YEAR(), MONTH(), and HOUR() can isolate these parts for database records.

	Date	Time	Methane	Toilet_temp	Toilet_virtual_outdoor_temp	Toilet_thermostat_temp	Toilet_brightness	Toilet_humidity	IR
0	2004-03-10	18:00:00	674.0	15.91	6.4	15.53	3.66	50.0	52.0
1	2004-03-10	19:00:00	646.0	15.75	7.3	15.37	14.65	49.0	53.0
2	2004-03-10	20:00:00	590.0	15.59	7.9	15.22	33.88	48.0	51.0
3	2004-03-10	21:00:00	627.0	16.06	8.1	15.06	54.02	49.0	55.0
4	2004-03-10	22:00:00	896.0	16.38	8.2	15.37	75.08	50.0	57.0

After the date components are split, they can be stored in separate variables or columns in a database. For example, in a table tracking system events, the fields EventYear, EventMonth, and EventTime might store each part separately. Storing dates this way enhances the system's ability to generate alerts, as it can quickly retrieve relevant records and apply conditions specifically to the time component (such as sending hourly alerts) or to the month and year (for periodic reports or detecting seasonal patterns).

In real-time alert scenarios, splitting the date is also helpful for time-based triggers. An alert could be set to activate only during specific hours or months. For example, a retail system might generate alerts for high traffic only in December or during peak hours, improving operational efficiency by reducing the likelihood of unnecessary alerts.Furthermore, splitting dates is advantageous for historical analysis and reporting. An alert-generating system often needs to compare current data with historical data to detect trends or outliers. By maintaining separated date components, the system can compare records more effectively.

Separating date components in an alert-generating system allows for precise threshold management, essential for reducing false positives and optimizing alert relevance. By isolating month, year, and time elements, the system can account for predictable variations such as seasonal demand cycles or hourly traffic fluctuations. For instance, isolating the month component enables setting higher alert thresholds during expected high-activity months, such as retail peaks in November and December. This segmentation also enhances the system's ability to differentiate between peak and off-peak hours, ensuring that alerts align with relevant activity levels and reducing unnecessary alerts during low-traffic times. Such time-specific precision enhances both the reliability and efficiency of the alerting process, making it responsive to real-world patterns.

Furthermore, breaking down dates into individual components facilitates advanced forecasting and trend analysis, allowing for more informed, data-driven alert settings. Analyzing historical data by month or year provides insights into recurring patterns, enabling the system to anticipate future trends and set alerts proactively based on these forecasts. For example, in cybersecurity, the system might identify high-risk periods based on historical time-based attack data, setting preemptive alerts during these intervals. This time-aware approach improves the system's responsiveness to emerging risks while providing decision-makers with actionable insights for long-term strategic planning and risk management.

The Date-Time Splitting Module is a software component designed to extract and separate date and time information from a given input string or dataset. This module provides a flexible and efficient way to parse and split date-time data into its constituent parts.Functional Requirements.The module accepts date-time data in various formats, identifies and extracts date and time components, separates date and time into individual elements, and provides the extracted date and time components in a structured format.

4.1.5 APPLICATION MODULE

The Application Module for a lavatory alert-generating system is a comprehensive solution designed to maintain high hygiene standards, manage resources efficiently, and streamline restroom upkeep. It serves as the central component in a networked system that collects data from various IoT sensors within lavatories, such as occupancy, odor, and supply levels, to detect conditions that require maintenance or cleaning attention. Based on real-time sensor data, the module automatically triggers alerts for issues like high occupancy, unpleasant odors, or low supplies, delivering these notifications directly to maintenance or cleaning staff to ensure swift responses and maintain cleanliness in high-traffic areas.

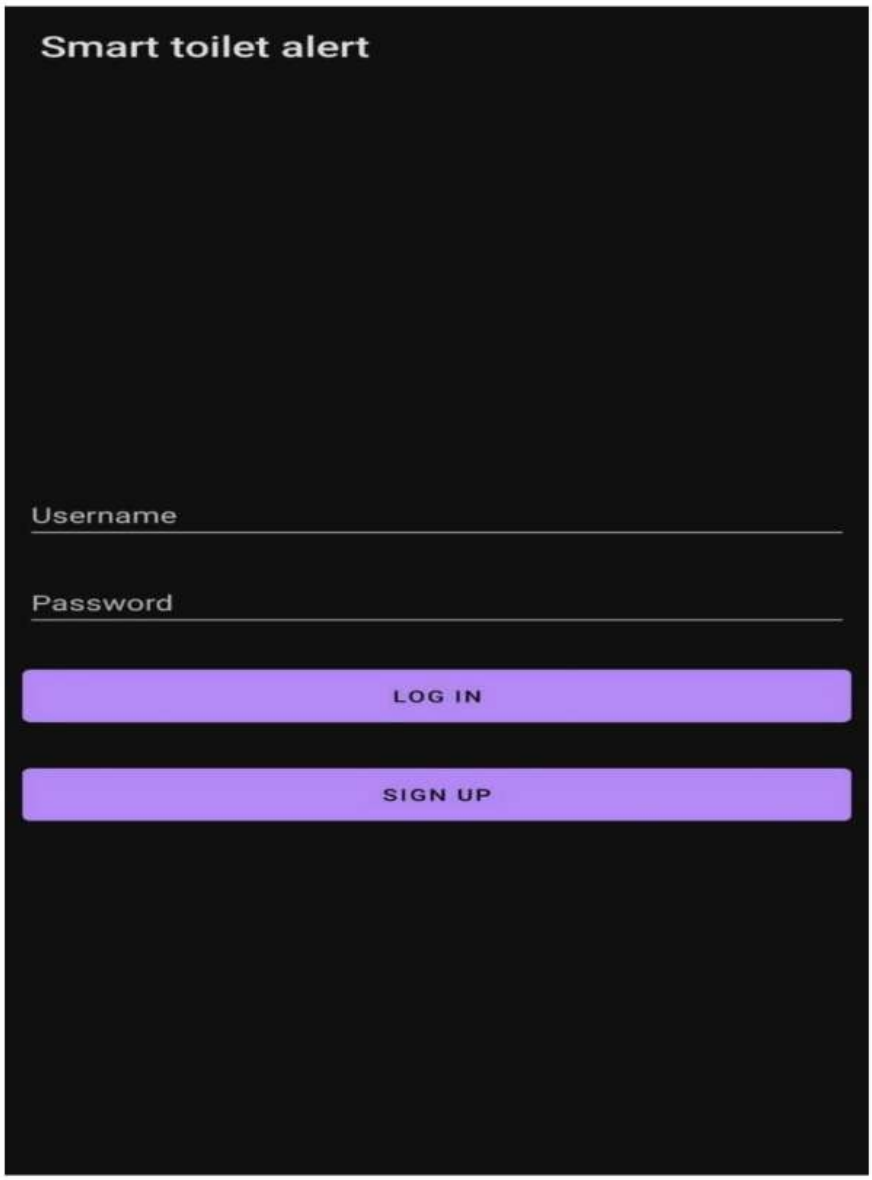


Figure 4.11: Login Page

The module operates through an intuitive dashboard accessible via both web and mobile devices. This dashboard provides a live view of lavatory status, active alerts, and historical data on issues and usage patterns. Maintenance personnel can update alert statuses, mark issues as resolved, and track response times, while administrators can access usage trends and generate reports tooptimize cleaning schedules and resource planning. Alerts are assigned priority levels, allowing staff to quickly recognize and respond to critical issues that impact user experience. Visual indicators, such as color-coded alert statuses, further enhance usability, helping maintenance teams to prioritize tasks efficiently.

The module is also highly customizable, allowing administrators to adjust alert thresholds, notification settings, and access permissions. This flexibility enables facility managers to configure the system according to the specific needs of each lavatory, adapting to factors like high or low traffic periods. Additionally, the module provides a robust incident tracking system, recording each alert and its resolution. This feature allows management to assess response times, track recurring issues, and monitor maintenance quality, creating accountability and ensuring high service standards.

As a data-driven tool, the module aggregates and logs all activity for analysis. It generates detailed reports on lavatory usage, common issues, and supply consumption, which can inform strategic decisions on resource allocation and supply ordering. Trend analysis within the module identifies peak usage times, helping managers optimize staffing and maintenance schedules for maximum efficiency. Ultimately, this Application Module supports a proactive approach to lavatory management, promoting a cleaner, well-maintained environment that enhances user satisfaction and operational efficiency across various facilities.

4.1.6 THRESHOLD ANALYZING MODULE

In an alert-generating system, the threshold-analyzing module plays a critical role by establishing, monitoring, and adjusting alert levels based on predefined or dynamically calculated criteria. This module operates by setting boundary values or thresholds, which, when crossed, trigger alerts. These thresholds are designed to distinguish between normal and abnormal values in the monitored data, such as traffic volume, system load, transaction rates, or environmental metrics, depending on the application. By accurately defining thresholds, the system can detect unusual behavior or potential issues in real-time, ensuring rapid response and minimizing downtime or other adverse impacts.

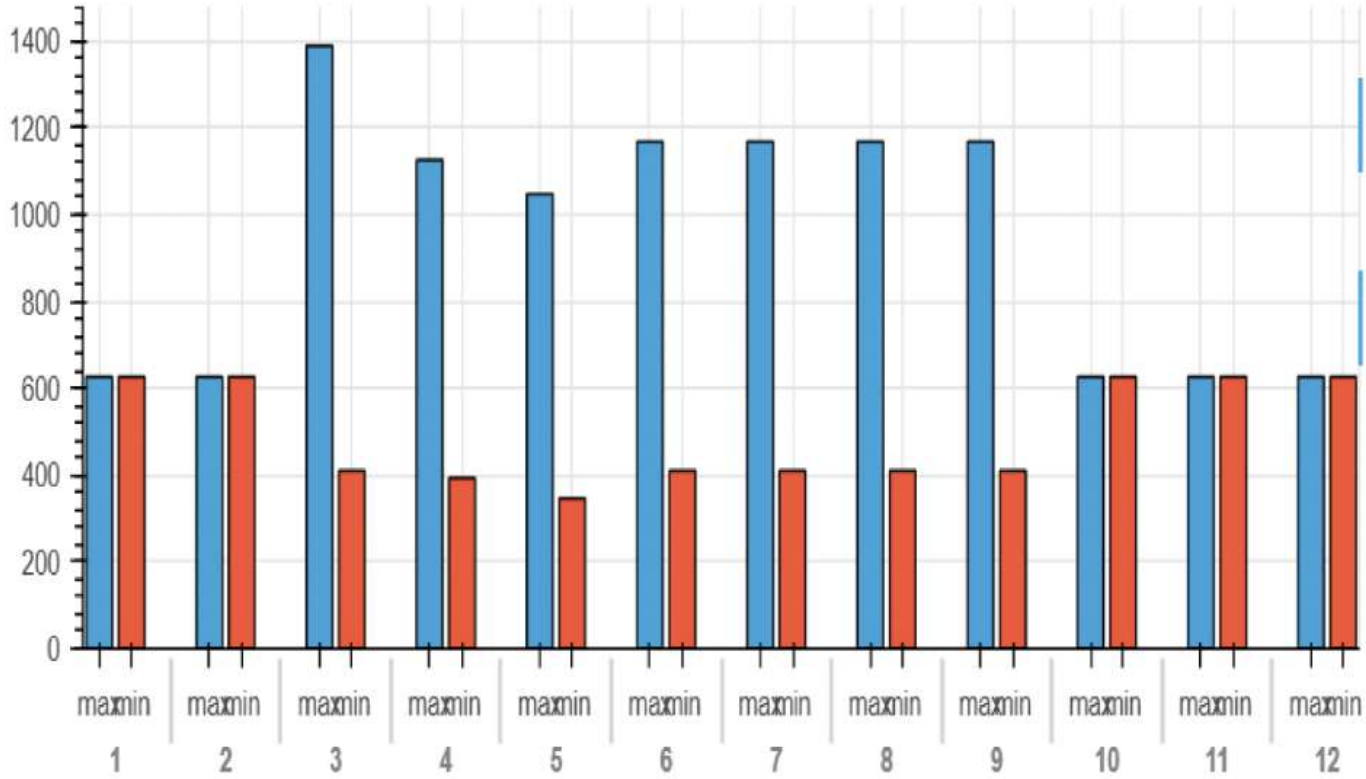


Figure 4.12: Threshold Analyzing

A threshold-analyzing module typically begins by gathering data inputs relevant to the system's objectives. For example, in a network monitoring system, the module might collect data on packet loss, response times, and connection errors. In this context, thresholds might be set based on acceptable performance ranges for each metric. The module could have fixed thresholds, such as a maximum allowed response time of 200 milliseconds, or adaptive thresholds that adjust based

on historical or real-time analysis. Adaptive thresholds are particularly effective for systems where normal values fluctuate significantly over time or are influenced by factors such as time of day or seasonal demand. In these cases, the module might employ statistical techniques, such as moving averages, standard deviations, or machine learning models, to establish dynamic baselines and automatically adjust alert levels accordingly.

Once thresholds are established, the module continuously monitors incoming data to detect deviations. When data exceeds a threshold, the system generates an alert, indicating a potential issue that requires attention. The alert is then typically prioritized based on the degree to which the data deviates from the threshold. In some systems, multi-level thresholds are used to create severity-based alerts. For example, a network monitoring system might have a warning level for minor packet loss, an error level for moderate packet loss, and a critical level for severe or continuous packet loss. This layered approach allows for prioritization in responding to alerts, focusing resources on more severe issues first while keeping lower-level issues on a watchlist.

The threshold-analyzing module also plays an important role in alert optimization and feedback adjustment. After an alert is generated, the system can record response times and outcomes to evaluate the threshold's accuracy. If a threshold is set too low, it may cause frequent false positives, leading to alert fatigue and potentially causing important alerts to be overlooked. Conversely, if thresholds are set too high, critical events might go undetected until they cause significant impact. The module may incorporate feedback mechanisms that use this performance data to adjust thresholds over time, either manually through administrator input or automatically through an adaptive learning model.

Threshold analysis also enables the module to perform trend and anomaly detection. By comparing current data to historical patterns, the system can detect subtle deviations from typical trends, generating early alerts even if values are within normal ranges but show an unusual pattern. This functionality is especially useful for applications like fraud detection, where sudden spikes or dips in activity, even within expected ranges, could signal malicious behavior. In such cases, the threshold-analyzing module might rely on anomaly detection algorithms, which can spot irregularities in the data, identify them as potential risks, and trigger alerts without relying solely on fixed thresholds.

Overall, the threshold-analyzing module is the foundation of effective alert generation, as it balances the need for accuracy with the flexibility to adapt to changing data patterns. Through continual monitoring, dynamic threshold adjustment, and a structured alert prioritization, this module ensures that alerts are both relevant and timely, providing system operators with actionable insights to maintain smooth and secure operations.

With the MQ135 sensor set to a threshold level of 600, system is configured to monitor air quality specifically for gas concentrations that may indicate unpleasant or harmful conditions, such as odors or pollutants in the lavatory. This setup allows the Threshold Analyzing Module to receive continuous data from the MQ135 sensor, which detects various gases like ammonia and carbon dioxide, commonly associated with poor air quality. When the sensor's output reading approaches or exceeds 600, the module interprets this as a potential issue that may require immediate attention, such as ventilation or cleaning.

The system's threshold of 600 acts as a trigger point for generating alerts. When this level is reached, an alert is sent to maintenance staff, notifying them of deteriorating air quality in the lavatory. This approach enables early intervention, which can be especially useful in high-traffic facilities where poor air quality could quickly affect user comfort. This real-time threshold setting allows the lavatory management team to address air quality issues promptly, preventing unpleasant conditions from persisting and impacting users.

If the system supports dynamic threshold adjustments, the Threshold Analyzing Module may adapt the threshold slightly during peak or low usage times. For example, the module might increase the threshold slightly when data indicates frequent but transient rises in gas levels, to prevent redundant alerts, while still keeping the focus on actual issues. This capability relies on historical data trends, which provide context on typical variations in air quality throughout the day, enabling smarter and more targeted monitoring.

Each alert generated by the Threshold Analyzing Module is logged, allowing for ongoing analysis of air quality data. Over time, this data offers insights into air quality patterns, helping facility managers understand how often alerts are triggered and during which times. Such information can support decisions to adjust cleaning or ventilation routines, reducing the likelihood of future issues. This historical data is also useful for trend analysis, showing when specific conditions (like high occupancy) may lead to poorer air quality, guiding strategic adjustments in facility management Finally, to ensure the MQ135 sensor continues to provide reliable readings, periodic recalibration may be necessary, as sensor sensitivity can vary with environmental conditions such as temperature or humidity.

The Threshold Analyzing Module can facilitate this process, ensuring that the threshold remains accurate over time and that alerts maintain their intended precision. By using the MQ135 sensor with a 600 threshold, the system provides effective, real-time air quality monitoring, ensuring a healthier and more comfortable environment for lavatory users while supporting efficient and proactive management.

Each alert generated by the Threshold Analyzing Module is logged, allowing for ongoing analysis of air quality data. Over time, this data offers insights into air quality patterns, helping facility managers understand how often alerts are triggered and during which times. Such information can support decisions to adjust cleaning or ventilation routines, reducing the likelihood of future issues. This historical data is also useful for trend analysis, showing when specific conditions (like high occupancy) may lead to poorer air quality, guiding strategic adjustments in facility management.

CHAPTER 5
SYSTEM SPECIFICATION

5.1 SOFTWARE REQUIREMENTS

- Android Studio
- Anaconda
- Jupiter Notebook
- Fire Base
- Arduino IDE
- Python
- Java
- **C**

5.2 HARDWARE REQUIREMENTS

- Nodemcu Board
- PIR Sensor
- MQ135 Sensor
- Bread Board

5.1.1 ANDROID STUDIO

Android Studio is the official integrated development environment (IDE) for building Android applications, created by Google and based on JetBrains' IntelliJ IDEA. It offers a comprehensive suite of tools for creating, debugging, and testing Android apps, making it the primary tool for developers working on the Android platform. Since its release, Android Studio has been optimized for the unique needs of Android development, providing features like a powerful code editor, real-time code analysis, emulator support, and extensive integration with Android-specific SDKs (software development kits) and tools.

One of the standout features of Android Studio is its robust code editor, which includes smart code completion, syntax highlighting, and a rich library of templates that speed up common development tasks. The editor provides a real-time preview of UI layouts, making it easy to build responsive designs that work well across various screen sizes and device types. This feature is further enhanced by Android Studio's Layout Editor, which allows developers to drag and drop UI components onto a visual canvas, preview changes immediately, and test layouts on different virtual devices without needing to compile the app each time.

The environment also integrates deeply with Gradle, a build automation system, which simplifies dependency management, code building, and packaging for different versions of an app (e.g., production vs. testing builds). This makes it straightforward to handle complex project configurations and dependencies, enabling developers to customize builds for specific scenarios like testing or release distribution.

5.1.2 ANACONDA

Anaconda is a popular open-source distribution primarily focused on Python and R programming languages, designed for data science, machine learning, and scientific computing. It simplifies package management and deployment, providing users with a comprehensive platform that includes a collection of libraries, tools, and an environment manager. Created by Anaconda, Inc., this platform is widely adopted in academia, industry, and research due to its ease of use, robust package management, and ability to handle complex dependencies, making it particularly suited for large-scale data science projects and cross-platform compatibility.

One of the main features of Anaconda is its package manager, conda, which allows users to easily install, update, and manage packages and their dependencies across multiple environments. By using conda environments, developers can create isolated workspaces with specific versions of packages, ensuring consistency across projects and avoiding conflicts between libraries. This is essential for data science, where different projects often rely on specific versions of libraries, and avoiding "dependency hell" is crucial. Additionally, Anaconda includes a wide range of pre-installed libraries, including NumPy, pandas, SciPy, scikit-learn, Matplotlib, and TensorFlow, enabling users to get started on data analysis and machine learning tasks without needing extensive setup

5.1.3 JUPITER NOTEBOOK

Jupyter Notebook is an open-source, interactive web application that allows users to create and share documents containing live code, visualizations, narrative text, equations, and multimedia elements. It's a powerful tool widely used in data science, machine learning, scientific research, and education for tasks that range from simple data analysis to complex deep learning model development.

The core of Jupyter Notebook is the "notebook" itself, a document that combines code cells with markdown cells. Code cells allow users to write and execute code in various programming languages—most commonly Python, but also R, Julia, and more—while markdown cells enable users to write formatted text, include headings, lists, images, links, and equations (using LaTeX syntax) for documentation or explanation. This structure makes Jupyter Notebook ideal for exploratory data analysis (EDA), as users can interactively test different methods, visualize data in real-time, and document their findings all in one place.

5.1.4 FIRE BASE

Firebase is a comprehensive platform developed by Google to support the development of web and mobile applications, offering a suite of tools that streamline backend services, improve app quality, and facilitate user engagement. Designed to reduce the complexity of managing backend infrastructure, Firebase provides services like real-time databases, authentication, cloud storage, hosting, and cloud functions, all integrated within a single, unified platform. One of its core features is the Firebase Realtime Database, which enables apps to sync data instantly across all clients in real-time, making it ideal for applications requiring collaborative or live-update functionalities, such as chat apps, collaborative work tools, or multiplayer games.

Firebase Authentication simplifies the process of user authentication by supporting sign-in methods like email/password, Google, Facebook, and other providers, enhancing user experience with easy integration and security. Another valuable component is Firebase Analytics, which gives developers a deeper understanding of user behavior, tracking engagement, retention, and conversion across platforms, thereby guiding data-driven enhancements to the app. Cloud Firestore, a newer database solution within Firebase, expands capabilities and offline functionality, making it especially suitable for applications requiring complex data management. Additionally, Firebase provides tools for A/B testing, crash reporting, and remote configuration.

5.1.5 ARDUINO IDE

The Arduino IDE (Integrated Development Environment) is a user-friendly platform that enables developers, hobbyists, and engineers to write, compile, and upload code to Arduino microcontroller boards. Designed to simplify the process of programming microcontrollers, the Arduino IDE supports the C and C++ programming languages, with a library of built-in functions and simplified syntax to make hardware programming accessible. The IDE includes an editor where users can write code, a compiler to translate the code into machine language, and an uploader to transfer the compiled code to the Arduino board via USB. One of the Arduino IDE's standout features is its pre-configured libraries, which allow users to easily add functions for controlling sensors, motors, LEDs, and other hardware components without extensive coding. It also has a built-in serial monitor that displays real-time data from the Arduino board, which is invaluable for debugging and monitoring applications. Available for Windows, macOS, and Linux, the Arduino IDE is open-source and has a large, active community that contributes libraries, tutorials, and code samples, making it a powerful yet approachable tool for developing embedded systems and IoT projects.

5.1.6 JAVA

In Java, a threshold-analyzing module within an alert-generating system leverages Java's powerful data-handling, concurrency, and modular capabilities to monitor key metrics and trigger alerts based on predefined or adaptive thresholds. Java’s built-in libraries, such as java.time for handling date and time, java.util for managing collections, and java.math for handling complex calculations, make it highly suited for processing real-time data inputs and calculating thresholds in various domains. The threshold values can be configured as static constants or dynamically calculated using algorithms or data models, and Java's Object-Oriented Programming (OOP) structure makes it possible to encapsulate these thresholds within specific classes, ensuring modularity and ease of updates.

Java's ExecutorService framework can efficiently handle multi-threaded monitoring, allowing the system to process high volumes of data concurrently, which is crucial for real-time alerting systems. Additionally, Java’s robust exception handling ensures that the system can manage

unexpected values or inputs gracefully, maintaining stability even under fluctuating data loads. In scenarios that require dynamic thresholds, Java integrates seamlessly with machine learning frameworks such as Weka or Deeplearning4j, enabling real-time, adaptive threshold adjustments based on predictive models. This adaptability allows the module to adjust alert thresholds based on historical data patterns or anticipated seasonal trends, using Java-based machine learning models to detect anomalies or potential risks proactively.

5.1.7 PYTHON

Python is a versatile, high-level programming language known for its simplicity and readability, making it a popular choice for beginners and experienced developers alike. Its syntax is designed to be easy to understand and write, reducing the amount of code needed to accomplish tasks compared to many other programming languages. This clarity has led to Python's widespread use in web development, data science, artificial intelligence, automation, and many other fields. Python's extensive libraries and frameworks, like Django for web development, Pandas and NumPy for data manipulation, and TensorFlow and PyTorch for machine learning, make it particularly powerful for specialized applications. Additionally, Python supports multiple programming paradigms, including procedural, object-oriented, and functional programming, giving developers flexibility in how they approach problems. Its large, supportive community also means there is an abundance of tutorials, resources, and tools available, which contributes to Python's ongoing evolution and adaptability to new challenges in technology.

5.1.8 C PROGRAMMING

C is a powerful and versatile programming language developed by Dennis Ritchie in the 1970s. Known for its efficiency, C is widely used for system-level programming, including operating systems, embedded systems, and high-performance applications. Its simplicity and ability to provide low-level access to memory make it a preferred choice for tasks requiring direct hardware interaction. The language is portable, allowing developers to write code that runs on multiple platforms with minimal changes.

CHAPTER 6

METHODOLOGY

6.1 SENSOR DATA COLLECTION

The methodology for sensor data collection in an alert generation system involves several interconnected stages to ensure the reliability and efficiency of the monitoring and alerting process. The first step involves identifying the purpose of the system, such as environmental monitoring, industrial safety, or healthcare alerts. This ensures the system meets specific operational goals. Based on the defined objectives, appropriate sensors are selected to measure relevant parameters .

Factors such as accuracy, sensitivity, range, power consumption, and cost are considered to ensure optimal performance. For example, in a temperature monitoring system, sensors might be placed in high-risk areas prone to overheating. Sensors are connected to a data acquisition system using appropriate communication. The choice depends on factors like range, bandwidth, and power availability. Before operation, sensors are calibrated to ensure they provide accurate measurements, accounting for environmental conditions. Sensors capture raw data continuously or at defined intervals, depending on system requirements. This data often includes time-stamped readings to provide contextual insights. The collected data is transmitted in real-time or near-real-time to a central processing system. This transmission relies on reliable protocols to prevent data loss or corruption. Raw sensor data often contains noise or irrelevant variations. Techniques like moving averages or advanced filtering algorithms are applied to enhance signal quality. The data is normalized to bring it within a standard range and formatted for compatibility with processing algorithms. Missing or inc Predefined thresholds are set for parameters. For example, a temperature sensor may trigger an alert if readings exceed 50°C.omplete data is addressed using interpolation, imputation, or other error-handling methods. Machine learning or statistical methods are applied to identify patterns or trends indicating potential issues. Advanced algorithms detect deviations from normal behavior, which may indicate system faults or risks. When thresholds are breached or anomalies are detected, the system generates alerts. These triggers are designed to be fast and reliable to ensure timely notifications.

Alerts are categorized based on severity to enable appropriate response levels. Alerts are sent to users through multiple channels, such as SMS, email, mobile apps, or on-site alarms, ensuring the message is received promptly. User feedback and historical data are used to refine the alerting thresholds and detection algorithms, improving accuracy over time.

6.2 DATA TRANSMISSION AND PROCESSING

The sensor data is collected by a microcontroller, which functions as the core of the system, responsible for processing inputs in real time. Sensors embedded within the system continuously monitor parameters such as occupancy and gas levels within the toilet environment. The microcontroller receives these raw inputs and processes them using pre-programmed algorithms, enabling it to interpret the data accurately. For instance, occupancy sensors detect whether the toilet is in use, while gas sensors monitor the concentration of specific gases to assess air quality and safety.

Once the microcontroller processes this data, it is transmitted to a cloud-based service, such as Firebase, using wireless communication protocols like Wi-Fi or Bluetooth. Firebase serves as a centralized platform for storing the processed data securely and efficiently. Beyond storage, Firebase also provides tools for further analysis, allowing the system to generate insights or notifications based on trends or anomalies in the data. This cloud integration enables remote access to toilet status information, which can be viewed or managed through web or mobile applications. By leveraging the cloud, the system ensures scalability, real-time monitoring, and ease of access, enhancing both user convenience and administrative control over the toilet facilities. The system begins with sensor integration, where occupancy sensors and gas level detectors work in tandem to gather raw environmental data from the toilet. Occupancy sensors, such as infrared or ultrasonic devices, identify whether the toilet is currently in use, while gas sensors detect harmful or unpleasant gases like ammonia, methane, or carbon dioxide. This dual-layer monitoring ensures comprehensive data collection, enabling the system to capture both usage status and air quality conditions. These sensors are strategically positioned to maximize coverage and accuracy while minimizing false readings. The remote access capability provided by Firebase enhances user and administrative convenience. Through web or mobile applications, users can check the status of toilet availability or air quality before use, while facility managers can monitor multiple toilets across different locations. This not only improves user experience but also optimizes resource allocation for maintenance and cleaning services. By integrating real-time processing, cloud storage, and remote access, the system offers a modern, scalable solution for managing toilet facilities efficiently and effectively.

6.3 DECISION MAKING PROCESS

The decision-making process within the system is primarily guided by a linear regression model, which is used to analyze gas level data collected from the sensors in the toilet. The model is designed to predict and evaluate gas concentrations in real-time, based on historical data and patterns. By analyzing this data, the linear regression model can detect trends and establish thresholds for various conditions, including safe gas levels. For example, the system is programmed to recognize when the concentration of gases, such as methane or ammonia, exceeds a predefined safety threshold that could indicate a potential hazard or poor air quality.

When gas levels surpass this threshold, the linear regression model triggers an alert in the system. This alert is not only based on raw sensor data but also considers contextual factors such as historical gas level patterns and occupancy times. The model's predictive capabilities ensure that alerts are generated accurately and promptly when a potential issue is identified. The alert serves as a signal that the air quality has deteriorated beyond acceptable levels, potentially affecting user safety and comfort.

Once the alert is generated, the system automatically sends a notification to the maintenance staff or relevant personnel through the connected system, such as a mobile app, email, or internal communication platform. This alert is designed to be clear and actionable, detailing the specific issue (e.g., high gas concentration) and the required response, such as cleaning or ventilation adjustments. The use of automated notifications streamlines the process of addressing issues in real-time, ensuring that maintenance teams are immediately informed without delay, which can be crucial in maintaining safe and hygienic conditions.

By incorporating the linear regression model into the decision-making process, the system not only automates the detection of gas level anomalies but also ensures that maintenance staff is alerted based on reliable, data-driven predictions. This approach reduces the likelihood of human error and enhances the speed and accuracy of responses. Moreover, the decision-making process ensures that the necessary steps are taken before the situation escalates into a more serious problem, such as a hazardous gas build-up or a severe decline in air quality, thus safeguarding both users and the facility.

6.4 ALERT GENERATION AND NOTIFICATION

The alert generation and notification process is a critical component of the system, designed to ensure quick action when issues arise with the toilet environment, particularly concerning gas levels. This process begins when the linear regression model predicts that the concentration of gases in the toilet exceeds safe thresholds. The model uses real-time sensor data, along with historical patterns and predictive analytics, to determine whether the gas levels have reached a point where maintenance or cleaning is required. If gas levels surpass the predefined threshold, the model triggers an automated alert, indicating that immediate attention is needed.Once the alert is triggered, the system automatically generates a notification that is transmitted through the cloud-based Firebase platform. Firebase serves as the communication backbone for the entire system, providing a seamless way to send real-time alerts to the appropriate maintenance personnel. The notifications can be configured to be sent via various channels, such as push notifications through a mobile app, emails, or even SMS messages, depending on the setup and preferences of the maintenance team. This ensures that the right person is promptly notified regardless of their location or preferred communication method.

The notification itself is carefully crafted to be both informative and actionable. It typically includes details such as the specific issue (e.g., "high gas concentration detected"), the location of the toilet that requires attention, and any additional context, such as the severity of the problem. This clear and concise communication ensures that maintenance staff can quickly assess the situation and respond appropriately. In the case of high gas levels, the maintenance staff is alerted that cleaning or ventilation intervention is needed to restore the environment to a safe and hygienic state.Because the Firebase system operates in real-time, the notifications are sent immediately after the model's prediction indicates a problem, ensuring that maintenance personnel can act swiftly. This reduces the risk of prolonged exposure to poor air quality or the possibility of other issues, such as a buildup of hazardous gases, which could lead to health concerns or discomfort for users. Additionally, by relying on automated alerts, the system eliminates the need for manual monitoring, ensuring that maintenance staff is always promptly informed when action is necessary.Overall, the alert generation and notification system powered by Firebase enhances operational efficiency by enabling quick, data-driven responses to potential issues. It empowers maintenance teams to address problems in real-time, ensuring that toilets remain clean, safe, and comfortable for users. This proactive approach reduces downtime, enhances user experience, and contributes to a more effective and reliable facility management system.

6.5 SYSTEM FEEDBACK LOOP AND MODEL ADJUSTMENT

The system incorporates a feedback loop that allows the model to continuously improve and adjust its predictions based on real-time data, ensuring long-term accuracy and reliability. This feedback loop operates by regularly feeding new sensor data back into the linear regression model. As the model processes new information about gas levels, occupancy status, and other environmental factors, it is able to fine-tune its predictions and adapt to any changes in the toilet environment.

Each time the sensors collect data, whether it's a gas concentration reading or an occupancy signal, this data is transmitted to the cloud service, where it is processed and used to update the model's understanding of the environment. The model compares the newly acquired data against historical data and evaluates the accuracy of its previous predictions. For example, if the model previously predicted that gas levels would remain within a safe range but new data shows a spike in gas concentration, the model can adjust its thresholds and predictive parameters to better align with the changing conditions.

This continuous flow of data allows the model to refine its predictions over time, reducing the risk of errors and improving its ability to anticipate future conditions. By incorporating real-time data, the system becomes increasingly effective at forecasting when maintenance will be required or when thresholds will be exceeded. The adjustment process ensures that the system stays relevant and responsive to evolving conditions, such as seasonal changes in gas levels or patterns in toilet usage.

The feedback loop also allows the system to detect anomalies or unexpected patterns that may have been previously overlooked. If, for instance, the model identifies a recurring pattern of high gas levels at certain times of the day or after specific events, it can modify its predictive algorithms to account for these nuances. This helps ensure that the system can handle a wide range of scenarios, even as user behavior or environmental conditions change over time.By continuously learning from new data, the system not only improves the accuracy of its alerts and maintenance predictions but also optimizes the overall management of the toilet facilities. The dynamic nature of the feedback loop ensures that the model remains effective in delivering timely, actionable insights and helps maintain the safety and comfort of users in the long run.

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

7.1 CONCLUSION

In conclusion, the system discussed integrates cutting-edge technology to streamline the management and maintenance of toilet facilities. At its core, the system combines real-time data collection, processing, and analysis to provide an efficient solution for monitoring occupancy and gas levels in restrooms. Sensors embedded within the facilities continuously gather data on critical parameters, such as whether the toilet is occupied and the concentration of harmful gases like methane, ammonia, or carbon dioxide..The processed data is then transmitted to a cloud-based platform, such as Firebase, which serves as the central hub for data storage, analysis, and remote access. Firebase not only securely stores the collected data but also enables further historical and real-time data, providing insights into usage patterns and air quality trends. This cloud integration allows facility managers and users to remotely monitor the status of the toilet, making it easy to track important metrics such as gas levels and occupancy in real time.The linear regression model employed by the system plays a crucial role in decision-making. It uses historical and real-time data to predict and assess gas levels in the toilet environment. When the model detects that gas concentrations have exceeded predefined safety thresholds, an automated alert is triggered. This alert, generated by the system, is instantly communicated to maintenance personnel through Firebase, ensuring they are informed promptly and can take action, such as cleaning or improving ventilation. This real-time notification system eliminates the need for manual monitoring, reduces response times, and ensures that necessary maintenance is performed without delay, enhancing the safety and hygiene of the facility.Furthermore, the system is designed with a dynamic feedback loop, where the linear regression model continuously receives new sensor data. This feedback loop allows the model to adjust its predictions over time, refining its understanding of normal gas levels and occupancy patterns. As the system processes more data, it learns from past experiences, becoming increasingly accurate in forecasting when maintenance will be required and adjusting its parameters to reflect changing environmental conditions. The ability to continuously refine the model's predictions based on real-time data makes the system adaptable to a wide range of scenarios, ensuring its long-term effectiveness and providing ongoing improvements in facility maintenance practices

7.2 FUTURE ENHANCEMENT

Future enhancements to the system could significantly improve its functionality and adaptability. One potential enhancement would involve the integration of more advanced machine learning models, such as deep learning algorithms, to refine the system's ability to predict and assess gas levels, occupancy patterns, and other environmental factors. By leveraging these more sophisticated models, the system could handle more complex patterns in the data, leading to improved accuracy in predicting maintenance needs and identifying potential issues earlier.Another area of improvement would be the addition of more environmental sensors. For instance, incorporating sensors to measure humidity, temperature, and air quality would provide a more comprehensive view of the toilet environment. This added data would allow the system to not only monitor gas levels and occupancy but also adjust predictions based on factors such as high humidity or temperature, which could affect air quality and cleanliness.

Allowing users to report issues via a mobile app could provide real-time input on toilet conditions, which the system could use to trigger alerts or adjust its predictions. For example, if users report discomfort due to poor air quality or cleanliness, the system could generate immediate alerts for maintenance teams, improving the overall user experience and system accuracy.In addition, real-time analytics dashboards would give administrators a more comprehensive view of the toilet network, allowing them to track status, gas levels, occupancy, and maintenance needs across multiple locations. These dashboards could also include predictive insights, helping managers identify potential problems before they arise, enabling preemptive actionsExpanding the communication options within the system could further enhance its usability. Notifications could be sent through various channels, such as email, SMS, or messaging platforms like Slack or Microsoft Teams, ensuring that maintenance staff receive alerts in their preferred formats. This flexibility would improve the system's efficiency in larger, more diverse teams.The mobile app could also be improved with more features, such as the ability for users to check toilet availability, air quality, and cleanliness in real time. Features like location tracking could guide users to the nearest available toilet, enhancing user experience, especially in large public areas like shopping malls or airports. These future enhancements would make the system more intelligent, adaptive, and user-centered, leading to a more efficient, responsive, and sustainable solution for managing toilet facilities.

APPENDIX – 1

SOURCE CODE

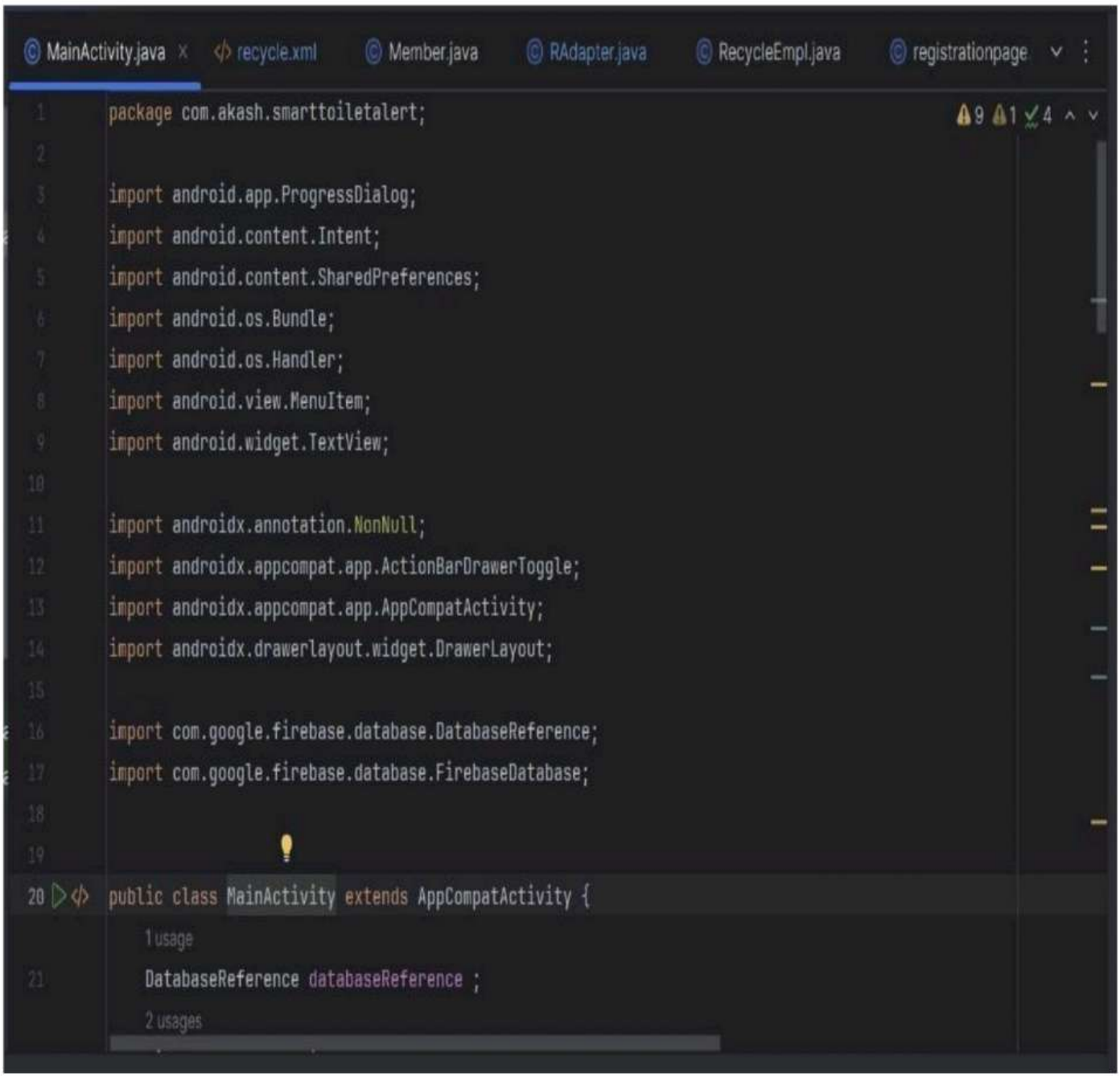
MainActivity.java

```
package com.akash.smarttoiletalert;
import  android.app.ProgressDialog;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.os.Handler;
import android.view.MenuItem;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.appcompat.app.ActionBarDrawerToggle;
import androidx.appcompat.app.AppCompatActivity;
import androidx.drawerlayout.widget.DrawerLayout;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
public class MainActivity extends AppCompatActivity {
   DatabaseReference databaseReference ;
   FirebaseDatabase firebaseDatabase;
   Member check;
   private TextView name;
   private static int SPLASH_TIME_OUT=3000;
   public DrawerLayout drawerLayout;
   public ActionBarDrawerToggle actionBarDrawerToggle;
   private ProgressDialog loadingbar;
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      loadingbar=new ProgressDialog(this);
      loadingbar.setTitle("    Loading Your HomePage  ");
```

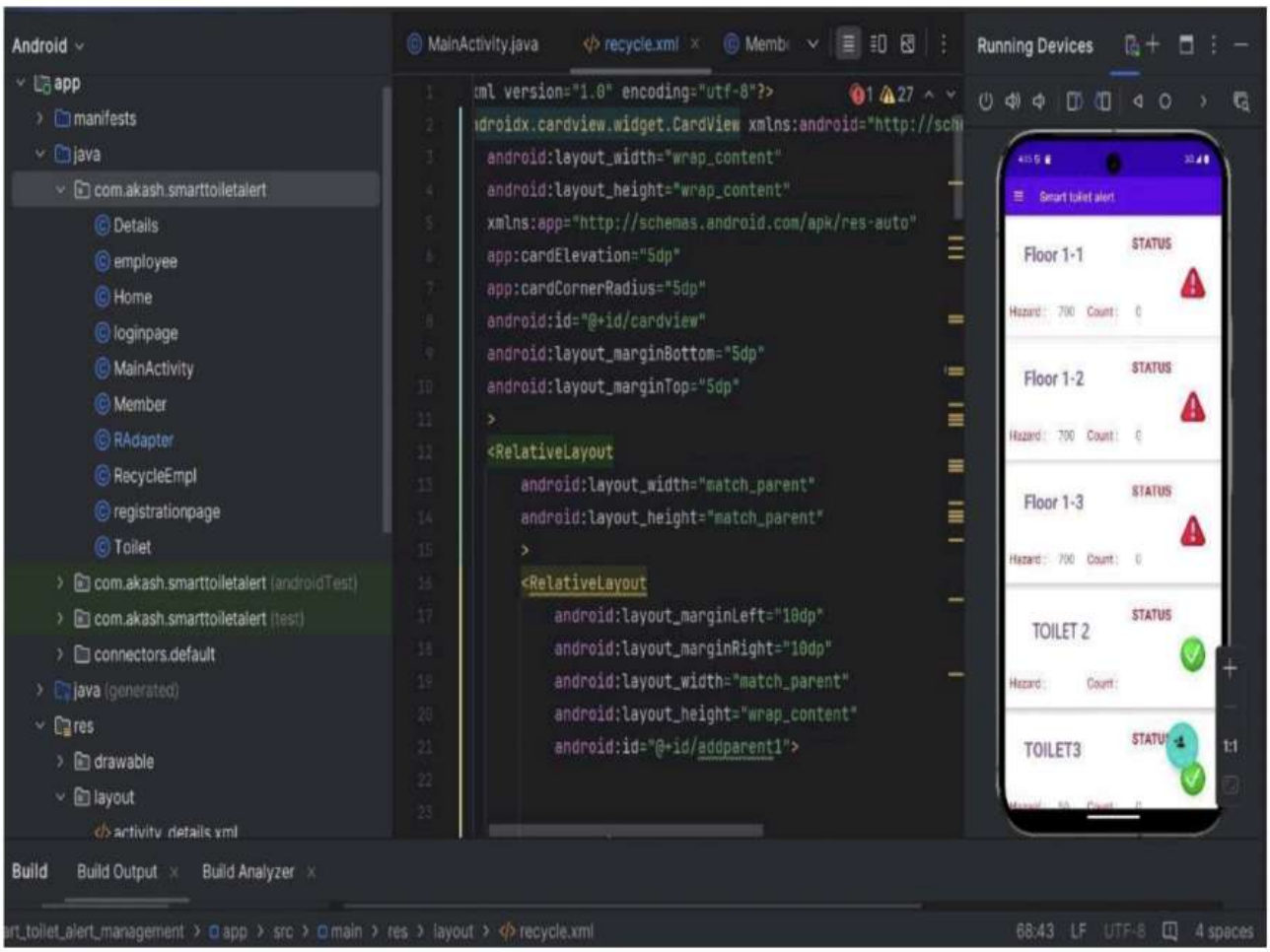
```
        loadingbar.setMessage("please wait, while we are get into your account..");
        loadingbar.show();
        loadingbar.setCanceledOnTouchOutside(true);
        firebaseDatabase= FirebaseDatabase.getInstance();
        databaseReference = firebaseDatabase.getInstance("https://smart-alert-toilet-default-rtdb.asia-
southeast1.firebasedatabase.app/").getReference("RAMANI");
    new Handler().postDelayed(new Runnable()
        public void run() {
            SharedPreferences sharedPreferences=getSharedPreferences(loginpage.PREFS_NAME,0);
            boolean  hasloggedin=sharedPreferences.getBoolean("hasloggedin",false);
            if(hasloggedin){
                startActivity(new Intent(MainActivity.this, Home.class));
                finish();
                loadingbar.dismiss();}
            else
            {    startActivity(new Intent(MainActivity.this,loginpage.class));
                finish();
                loadingbar.dismiss();
            }
        } },SPLASH_TIME_OUT
        drawerLayout = findViewById(R.id.my_drawer_layout);
        actionBarDrawerToggle=newActionBarDrawerToggle(this, drawerLayout, R.string.nav_open,
R.string.nav_close);
        drawerLayout.addDrawerListener(actionBarDrawerToggle);
        actionBarDrawerToggle.syncState();
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    } public boolean onOptionsItemSelected(@NonNull MenuItem item) {
        if (actionBarDrawerToggle.onOptionsItemSelected(item)) {
            return true;
        }return super.onOptionsItemSelected(item);
    }
}
```

APPENDIX – 2
SCREENSHOTS

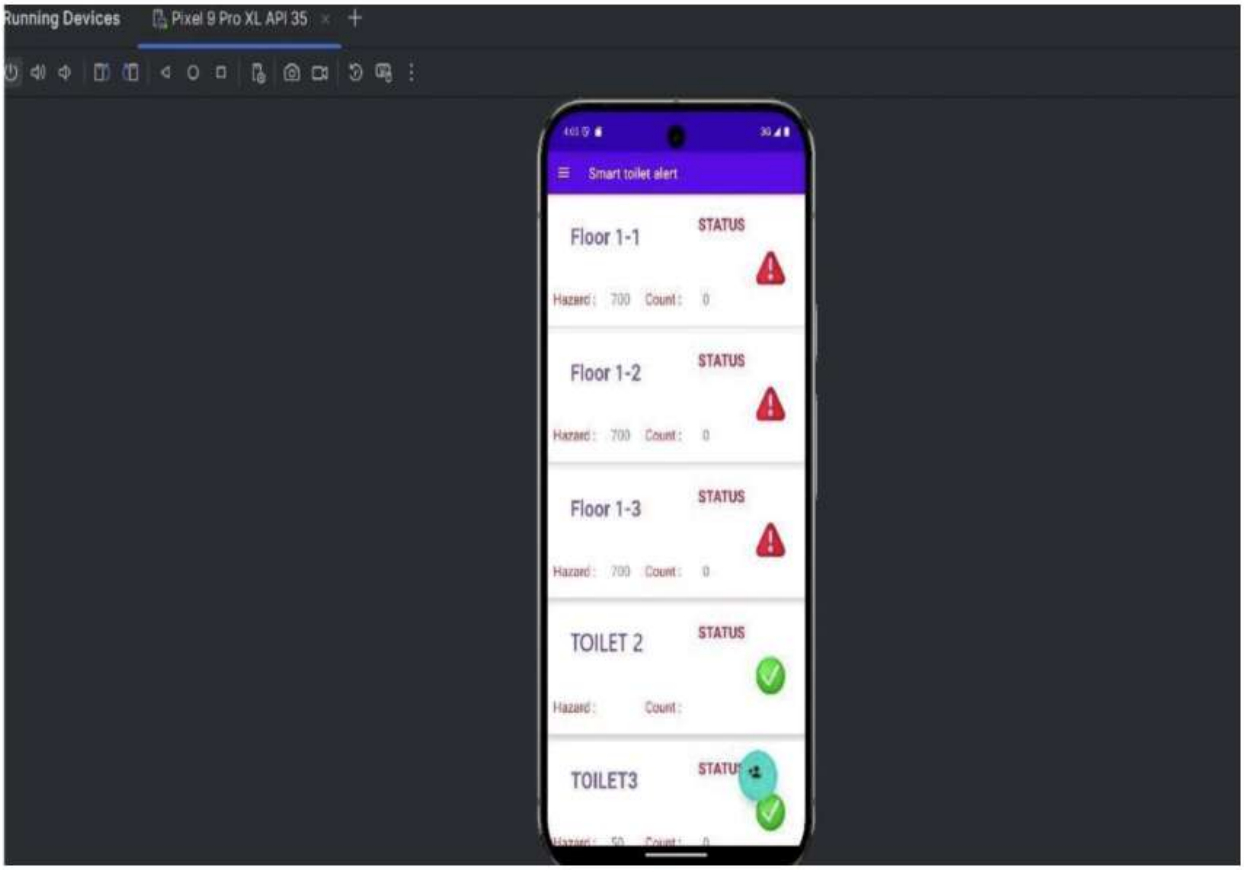
Sample Output



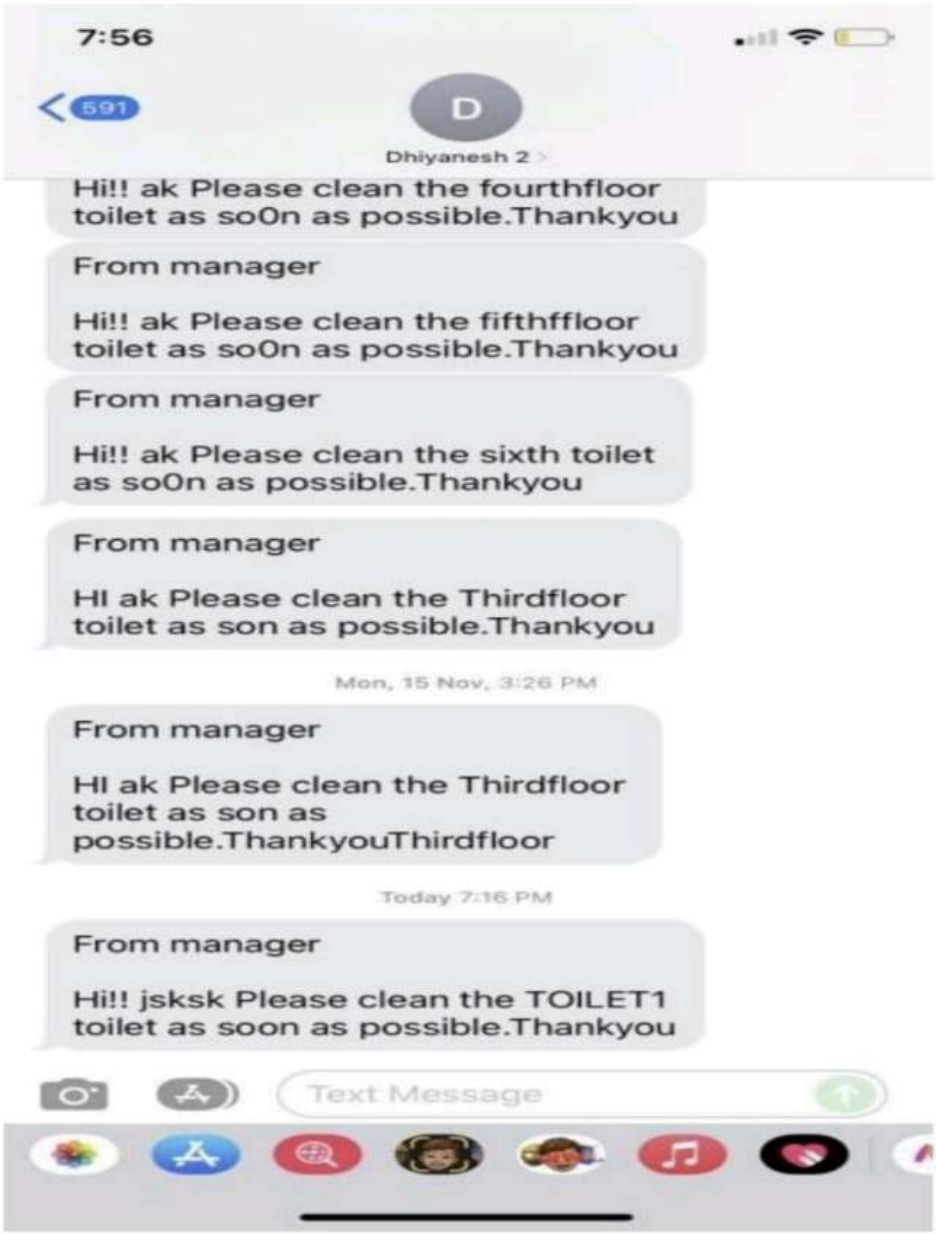
Execution of Code


Android
app
manifests
java
com.akash.smarttoiletalert
Details
employee
Home
loginpage
MainActivity
Member
RAdapter
RecycleEmpl
registrationpage
Toilet
com.akash.smarttoiletalert (androidTest)
com.akash.smarttoiletalert (test)
connectors.default
java (generated)
res
drawable
layout
activity_details.xml
MainActivity.java
recycle.xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView xmlns:android="http://sch
android:layout_width="wrap_content"
android:layout_height="wrap_content"
xmlns:app="http://schemas.android.com/apk/res-auto"
app:cardElevation="5dp"
app:cardCornerRadius="5dp"
android:id="@+id/cardview"
android:layout_marginBottom="5dp"
android:layout_marginTop="5dp"
>
<RelativeLayout
android:layout_width="match_parent"
android:layout_height="match_parent"
>
<RelativeLayout
android:layout_marginLeft="10dp"
android:layout_marginRight="10dp"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:id="@+id/addparent1">
Running Devices
Smart toilet alert
Floor 1-1
STATUS
Floor 1-2
STATUS
Floor 1-3
STATUS
TOILET 2
STATUS
TOILET3
Build
Build Output
Build Analyzer
68:43
LF
UTF-8
4 spaces

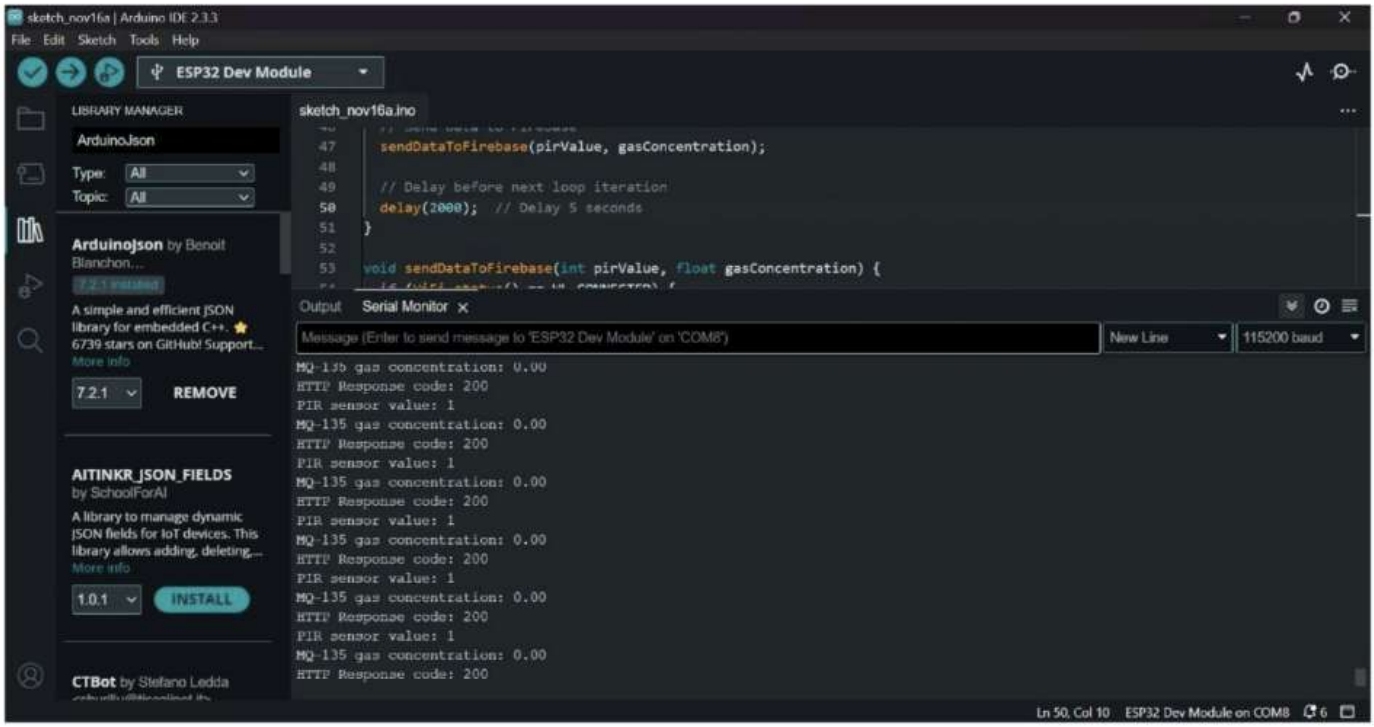
Android app



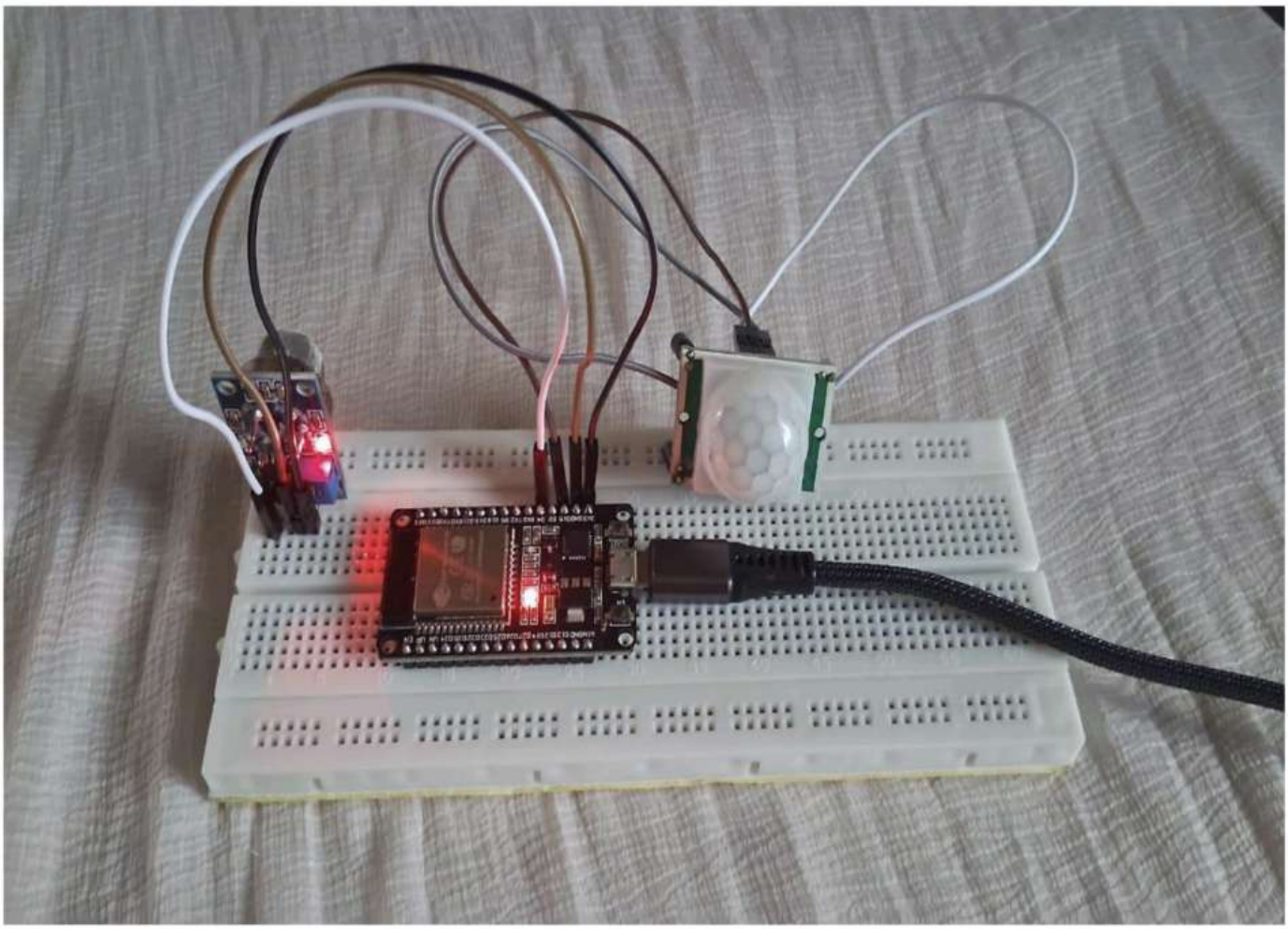
Running of android application



Alert Message For Employee



Sensor Output



ESP32 Connection

REFERENCES

[1] Basics about android studio, https://www.javatpoint.com/android- tutorial

[2]Android Development for Beginners - Full Course - YouTube
https://www.youtube.com/watch?v=fis26HvvDII

[3] Exploratory Data Analysis (EDA) Using Python | Python Data Analysis | Python Training | Edureka https://youtu.be/-o3AxdVcUtQ

[4]Data analysis of toilet management https://thesai.org/Downloads/Volume11No5/Paper_47-IoT_Technology_for_Facilities_Management.pdf

[5]Toilet-cleanliness-indicator
https://github.com/knot10120/toilet-cleanliness-indicator

[6] JJ K.Elavarasi, V.Suganthi International Journal of Pure and Applied Mathematics 119 (14), 2018 , 611-618

[7] An Internet-of-Things (IoT) system development and implementation for bathroom safety enhancement DD Koo, JJLee, A Sebastiani, J Kim Procedia Engineering 145, 2016 , Elsevier Ltd.

[8] An Internet of Things based solution for the open lavatory problem in developing countries V.Sudha , D.Aswini 2020 Fourth International Conference on Inventive Systems and Control (ICISC).

[9]Indian Women Afraid Of Using Public Washrooms: Survey
http://www.businessworld.in/article/90-Of-Indian-Women-Afraid-Of-Using-Public-Washrooms-Survey/08-08-2019-174544