Journal of Engineering Science and Technology
Vol. 21, No. 4 (2026) 1072 - 1088

DEVELOPMENT OF A WEB-BASED REMOTE MONITORING SYSTEM FOR SEWER GAS EMISSIONS

SHERLYE DONOL[1], JANICE LYNN AYOG[1],
ZYKAMILIA KAMIN[1,*], AWANG BONO[2]

[1]Faculty of Engineering, University Malaysia Sabah, Jalan UMS,
88400 Kota Kinabalu, Sabah, Malaysia
[2]Faculty of Business Management and Information Technology, Universiti
Muhammadiyah Malaysia, Padang Besar, 02100 Perlis, Malaysia
*Corresponding Author: zykamilia@ums.edu.my

Abstract

Monitoring sewer gas is important, as it poses significant risks to public health, the environment, and infrastructure. This is especially critical for sewer workers who are exposed to these gases on a daily basis. These gases are flammable and can cause explosions. In addition to physical hazards, they can lead to serious health issues such as eye irritation, respiratory failure and even death. Manual sewer gas monitoring requires a sewer worker to be physically present at the site to collect data. Since sewer flow fluctuates throughout the day, a larger dataset is needed for meaningful analysis, which means the sewer worker must remain on-site for extended periods. As a result, this increases their duration of exposure and the associated health risks. In this study, a web-based remote monitoring system was designed to provide users with continuous updates on gas concentration levels through an accessible web interface. First, a gas sensor prototype was developed to detect sewer gases, process the data and transmit it to a database. Then, a web interface was designed using Django, a Python-based web framework, to communicate with the database for data visualization. The database system was developed using MySQL for data management. The web-based system consists of a login page, a homepage and graph plotting features for trend analysis. Through the homepage, users can monitor gas composition in real time without the need to be on-site continuously. The web interface is accessible anytime and anywhere using a computing device, such as a computer or mobile phone, with an internet connection. This remote monitoring system offers a scalable, efficient and safe solution for sewer gas monitoring. Future work may include enhancements to gas sensor calibration and integration with an alert system.

Keywords: Chemical engineering, Gas monitoring, Internet of Things (IoT), Remote monitoring, Sewer gas, Web-based system.

1. Introduction

Sewer workers face hazardous working conditions daily due to the nature of their job, which involves the operation and maintenance of sewerage facilities. Sewers produce hazardous gases such as hydrogen sulphide, ammonia and methane, which can cause serious health problems [1]Click or tap here to enter text.. Exposure to hydrogen sulphide can cause irritation and nausea, while at higher concentrations it can be fatal [2].Click or tap here to enter text. Ammonia gas affects the respiratory system, causing symptoms such as coughing, shortness of breath and respiratory failure at higher concentrations, which can ultimately lead to death [3, 4]. Methane gas is highly flammable and is one of the major contributors to greenhouse gases (GHGs), playing a significant role in global warming and climate change [5]. Click or tap here to enter text.In China, a methane gas explosion resulted in loss of life and damage to surrounding infrastructure [6].

The characteristics of these gases, such as being colourless and odourless, make them even more difficult to detect without proper equipment, thereby increasing the risk of accidents. Even hydrogen sulphide, which has a distinct rotten egg smell, can cause olfactory fatigue at higher concentrations, a condition in which a person can no longer detect the smell. The Occupational Safety and Health Administration (OSHA) has set the Permissible Exposure Limit (PEL) for hydrogen sulphide at 10 ppm over an 8-hour work shift, while ammonia has a slightly higher PEL of 25 ppm. This PEL indicates the duration of exposure considered safe for workers without causing any adverse health effects [7].

In Sabah, the government agency responsible for managing sewerage in the state, the Sabah Sewerage Services Department (SSD), relies on manual monitoring using portable gas detectors. Currently, there is no online monitoring system for sewer gas emissions implemented by the department. Although no sewer gas-related accidents involving sewer workers have been reported to date, the department has previously received public complaints regarding pungent odour near sewerage assets. In one case, a sewer worker was dispatched to measure hydrogen sulphide concentrations, spending two hours in the morning and another two hours in the afternoon collecting data. From the data analysis, there was a period during which hydrogen sulphide concentrations at the affected site exceeded 10 ppm. The source of the odour could only be confirmed after a site investigation, and necessary actions were taken following the completion of data analysis. This reactive approach delays the response time to public complaints and increases the risk of prolonged exposure for sewer workers.

In terms of gas sensors, metal oxide semiconductor (MQ) sensors are among the preferred choices due to their low cost, fast response and high sensitivity [8-10]. When exposed to gases, an interaction occurs between the target gas and the sensing layer of the MQ sensor. This interaction results in a change in electrical resistance, which can be measured to determine the gas composition [11]. MQ sensors have certain limitations, as their sensitivity can be affected by factors such as temperature and humidity [12]. Water molecules act as a barrier that reduces the interaction between gas and the sensing layer, resulting in lower sensitivity under high humidity conditions. Additionally, the response time of the MQ sensor tends to increase with rising temperature but decreases rapidly once a certain threshold is reached [13]. Therefore, in this study, in addition to the MQ gas sensor, a weather condition sensor is implemented to provide data on temperature, humidity and air pressure.

The advancement of device control aligns with the development of gas sensors, especially open-source microcontrollers such as Arduino Uno [14, 15]. Click or tap here to enter text.Microcontrollers serve as the brain of the system, where raw gas data (analogue) is processed, converted into digital format and then transmitted to the database. The use of Arduino Uno as a microcontroller is widely used in gas monitoring [16-18]. Click or tap here to enter text.Although the Arduino Uno can process gas data, its lack of an integrated Wi-Fi module makes it unable to transmit data. As a result, these monitoring systems are often paired with ESP8266 or Digi XBee modules specifically for data transmission purposes, as reported in previous studies by Fahim et al. [19]. Ramesh and Saran [20] and Abraham and Lee [21]. The ESP8266 was further developed into a microcontroller with built-in Wi-Fi functionality, making it more advanced than the Arduino Uno. However, it features the same 10-bit Analog-to-Digital Converter (ADC) as the latter. The ADC is one of the key components in processing raw gas data; the higher the ADC resolution, the more accurate the representation of gas composition. This study used the ESP32 as a microcontroller, which has a 12-bit ADC and a built-in Wi-Fi module. Therefore, by utilizing ESP32, it provides more accurate readings and simplifies the network connection to the database [22].

A systematic database management system is crucial as it bridges the communication between the microcontroller and the web interface (Graphical User Interface, GUI). One of the database systems commonly used to store and manage sensor data is MySQL, an open-source Relational Database Management System (RDBMS) [23, 24]. This database system is useful when handling large volumes of data, especially those generated from continuous monitoring. In a web-based monitoring system, the MySQL database receives gas data from microcontrollers and is used to store, retrieve or update the data based on the system setup [25, 26]. Click or tap here to enter text.However, due to the communication protocol, the microcontroller cannot send data directly to the MySQL database. It requires an intermediary, which is the PHP interpreter [19]. PHP scripts are needed to translate requests or queries from the microcontroller before submitting them to the MySQL database.

Setting up a database system requires some level of computer programming knowledge; therefore, the use of commercial GUI platforms that can offer both database and web interface functionality has been on the rise in recent years [27-29]. These commercial platforms allow users to simply input information related to the monitored asset, while the platform automatically manages data storage and system configuration. The collected data are stored in the platform's cloud database, enabling users to remotely monitor and manage their assets through a ready-made interface. Blynk is a mobile-based application that provides ready-made templates, allowing users to easily create an account, log in and set up a monitoring system. Studies conducted by Arul et al. [30] and Jumaa et al. [31] used Blynk, which includes an alert system that notifies users when a predefined threshold is reached, a feature that is particularly useful in gas monitoring. Other studies done by Anusha et al. [32] and Vidhya et al. [33] used ThingSpeak, which also offers ready-made cloud-based database storage, allowing users to easily visualize and analyse data remotely.

Although these ready-made platforms are user-friendly and easy to set up, they present several limitations for large-scale monitoring applications. In most cases, the database belongs to the commercial platform provider, which raises concerns regarding data ownership and security, especially for government agencies. Furthermore, the system design is often rigid, and customization typically incurs additional costs. The free usage of these commercial platforms is usually limited to a single asset, often intended for testing or development purposes. As more assets are added, the cost increases accordingly, eventually making large-scale asset monitoring significantly more costly [34].

This limitation becomes particularly significant when managing a large number of sewerage assets, such as those operated by the Sabah Sewerage Services Department (SSD), which manages more than a thousand sewerage facilities across the state. Under such conditions, relying on commercial monitoring platforms becomes impractical and economically inefficient. In addition, manual monitoring methods are time-consuming and make continuous on-site monitoring difficult to implement. Therefore, a more efficient monitoring approach is required. To address this challenge, the development of a web-based monitoring system is necessary to enable remote and continuous monitoring of sewer gas emissions across multiple sewerage assets.

In this study, a customized web-based monitoring system was developed to provide full control over database management and system configuration. A gas sensor prototype was integrated with the system to transmit gas concentration data to a centralized database, where the information can be visualized through a web interface for real-time monitoring and analysis. Compared with existing commercial monitoring platforms, this approach offers greater flexibility, scalability and data ownership, making it more suitable for large-scale sewer gas monitoring applications.

The aim of this study is to develop a web-based remote monitoring system utilizing MySQL as the database management system and a custom-built web interface developed using Django, a Python-based web framework. The web interface was designed to visualize not only sewer gas composition but also environmental parameters such as temperature, humidity and air pressure, providing a more comprehensive overview of the sewer environment. To achieve this, two sensor prototypes were developed: a gas sensing unit and a weather sensing unit, which measure gas composition, temperature, humidity and air pressure. The system enables real-time visualization of gas composition data and supports historical data analysis through trend graphs. By enabling continuous remote monitoring through the developed web-based system, this study provides a practical approach for real-time monitoring of sewer gas emissions, improving response time, reducing reliance on manual monitoring and enhancing the safety of sewer workers.

2. Proposed System Requirement

2.1. Hardware requirement

The sewer gas detection prototype, referred to as the Gas Sensing Unit (GSU), consists of MQ gas sensors, an ESP32 microcontroller and an OLED display. Each GSU is dedicated to a specific gas type and is connected to its respective microcontroller and OLED display, all assembled on individual breadboards. A weather condition sensor named Weather Sensing Unit (WSU) is also incorporated to provide temperature, humidity and pressure readings. The main hardware components used in the proposed sewer gas monitoring system are shown in Fig. 1.

(a) Gas sensor

The sensors used in this study are metal-oxide semiconductor (MQ) type gas sensors. MQ-4, MQ-136, and MQ-137 were selected to detect methane, hydrogen sulphide, and ammonia, respectively. All these MQ sensors use the same principle for detecting gas. They are readily available both locally and through international online stores. Their affordability makes them one of the top choices for gas monitoring. It has four pins: VCC (power input), GND, DO (digital output) and AO (analogue output). The AO pin sends an analogue output (voltage output) to the ESP32 for further processing.

(b) ESP32 microcontroller

The system uses the ESP32-WROOM-32 microcontroller (30-Pin) with an integrated Wi-Fi system, enabling it to connect to other Wi-Fi networks for data transmission. It operates at 3.3V to 5V and is powered via a Type-C connection, allowing it to connect directly to a power supply. It features a 12-bit Analog-to-Digital Converter (ADC), meaning it can represent analogue values as digital values from 0 to 4095 (2^{12} levels). The ESP32 is equipped with 25 GPIO pins that can be connected to other devices and serve various functions. GPIO36 is the pin that receives the analogue signal from the MQ gas sensor for ADC conversion. The ESP32 also has two I2C (Inter-Integrated Circuit) channels, which are the communication protocols that allow it to communicate with other devices. The system supports communication over both local (LAN) and wide-area (WAN) networks via Wi-Fi connectivity.

(c) Organic light emitting diode (OLED) display

The system uses a 0.96-inch OLED display (4-pin) for real-time data visualization of gas concentrations. OLED displays do not require a backlight, unlike traditional LCDs. Therefore, they consume less power while offering better contrast and visibility, even in low-light environments. The OLED display used for this study has a dual-colour scheme (yellow and blue) and a 128x64 pixel resolution, ensuring sharp and clear graphical output. It operates using the I2C communication protocol, which utilises two data pins, thereby reducing wiring complexity and simplifying integration with the ESP32 microcontroller. These pins are the Serial Data Line (SDA) and the Serial Clock Line (SCL). The SDA pin carries data, while the SCL pin provides the clock signal. The program is set to display gas concentration values in parts per million (ppm). With the display, it provides instant data to users on-site.

(d) Weather Condition sensor (AHT20+BMP280)

The weather condition sensor consists of two integrated components: AHT20 and BMP280. The AHT20 measures both temperature (°C) and relative humidity (%), while the BMP280 provides air pressure data as well as temperature. However, for accuracy and consistency, temperature data from the AHT20 sensor are used for display on the OLED and web interface, as it offers higher precision compared to the BMP280. The sensor also operates via the I2C communication protocol, making it easily integrable with the ESP32 microcontroller.

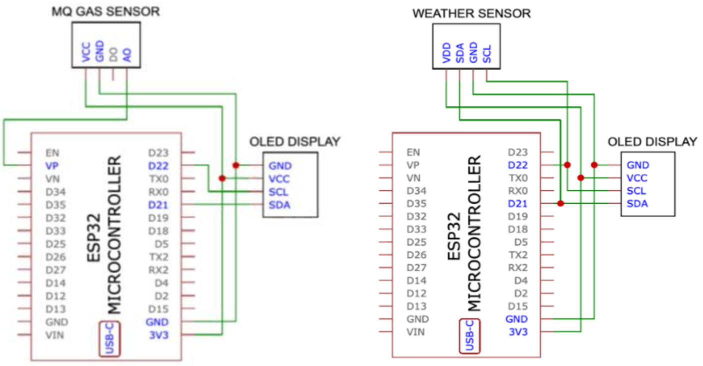
(a) (b) (c) (d)

Fig. 1. Hardware components used in the proposed sewer gas monitoring system. (a) MQ series gas sensor used for detecting methane (MQ-4), hydrogen sulphide (MQ-136) and ammonia (MQ-137) in sewer gas monitoring system, (b) ESP32 microcontroller as the core processing system, handling gas data processing and data transmission, (c) OLED display for data visualization on-site, and (d) Weather condition sensor (AHT20+BMP280) to monitor temperature, humidity and air pressure.

2.2. Software specifications

2.2.1. Microcontroller processing and sensor connectivity

The pin connection between the sensors, OLED display and the ESP32 is shown in Fig. 2. The power supply is provided through the 3V3 pin of the ESP32, which is connected to the VCC/VDD pins of the sensors and OLED display. The analogue output (AO) of the gas sensor is connected to GPIO36 (VP) on the ESP32, where ADC conversion is performed. The converted values are used to calculate gas concentration, which is then stored in a MySQL database. The DO pin is not used in this analogue configuration.



(a) (b)

Fig. 2. Circuit diagram of (a) GSU and (b) WSU.

MicroPython is a lightweight version of Python designed for embedded systems and is used to program the ESP32. The MicroPython scripts were developed through the Thonny IDE. Two software components were installed on the ESP32 microcontroller: the CP2102 driver and the SSD1306 library. The CP2102 driver enables USB-to-Serial (UART) communication between the microcontroller and the computer, while the SSD1306 library allows data to be displayed on the OLED screen. Once these steps were completed, the MicroPython code was ready to be uploaded to the ESP32 microcontroller, as shown in Fig. 3.

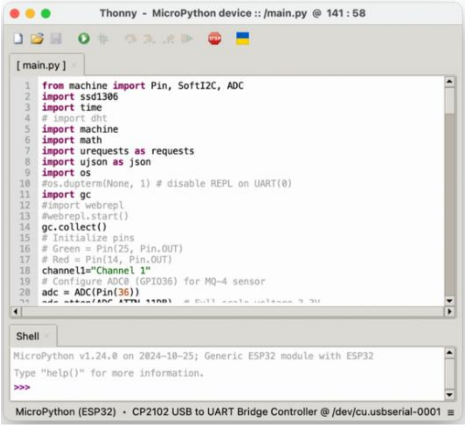


Fig. 3. Serial monitor of program uploaded in ESP32 microcontroller in Thonny IDE, for GSU.

The MQ sensors need to be calibrated, beginning with a preheating process that involves supplying power via the Type-C connection on the microcontroller for at least 48 hours. The MQ sensor relies on its internal heating element to function properly and provide accurate readings. Therefore, all GSUs need to be preheated, especially during initial use to allow for stabilization. After this period, the calibrated resistance value of the sensor, known as RO, is recorded under clean-room air conditions.

The gas value in parts per million (ppm) is calculated based on the sensitivity curve (log RS/RO ratio vs. ppm). The sensitivity curve was obtained from the datasheet provided by Zhengzhou Winsen Electronics Technology Co., Ltd. Once the ppm values are calculated, the base reading is taken and stored in the database as BaseComp. This completes the calibration process for the GSU.

2.2.2. Web-based Monitoring system programming

The back end consisted of the creation of the database and the web interface. A database was created using MySQL to store all sensor data. Separate tables were created in the database for methane, hydrogen sulphide, ammonia and weather conditions. Each table includes fields for sensor ID and composition, allowing both real-time monitoring and historical trend analysis. Additional tables were also designed to support website functionality, including user management, device setting and system configuration. The ESP32 transmits sensor data to the database over the HTTP protocol via POST method. However, MySQL is not capable of directly processing HTTP requests, thus an intermediary layer is required. To address this, PHP scripts are used to receive the incoming HTTP requests from the ESP32, parse the data and insert it into the database.

These PHP scripts are hosted on an Apache web server, which handles the server-side HTTP communication. The web interface was programmed in Python using Visual Studio Code as the development environment. Django, a Python-based web framework, was utilized to structure and manage the web application. Django simplifies the web development process by providing built-in templating and routing features, which handle URLs and connect users to specific functions such as the login page. It is also compatible with MySQL databases, making it easier to store and retrieve data.

The front end was built using integrated HyperText Markup Language (HTML), Cascading Style Sheets (CSS) and JavaScript. HTML is used to structure the content of the website, such as defining headings, menu buttons and graph plot buttons. CSS provides styling to the webpage, including fonts, spacing and colours. Lastly, JavaScript adds interactivity to the website. Real-time gas composition updates and interactive graphs are made possible through JavaScript. Figure 4 shows the HTML code for the homepage designed in Visual Studio Code.

The homepage of the system is titled Sewer Gas Emission Monitoring System. The website is designed with a login interface and a homepage view. The homepage displays information about the sewer assets or the locations of the sensing units. Alongside the sewer facility information, the types of gases being monitored and their real-time composition levels are also shown. For each gas, a 'Plot Composition' button is provided to view the gas concentration data in real-time graph format. Weather conditions are also shown to provide temperature, humidity and air pressure data.

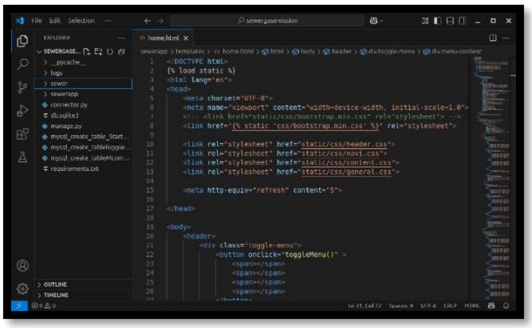


Fig. 4. Serial monitor of the HTML code for the web-based homepage view in Visual Studio Code.

A menu button is created on the navigation bar as a dropdown menu consisting of options such as Logger Setup, Device Setup, Download Logged Data, View Gas Composition, and other system-related settings. The summary of the menu button is as shown in Table 1. Figure 5 shows the complete hardware and software setup of the remote monitoring system.

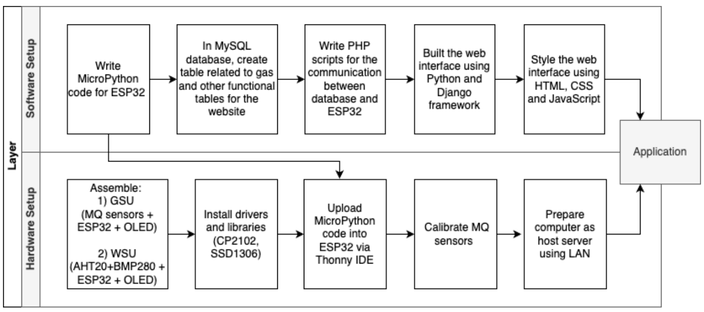


Fig. 5. Integration of the hardware and software setup for the system.

Table 1. Summary of Menu button functions.

Menu Button	Function
Device Setup	Form for users to input calibration data for each gas and its corresponding calibration date.
Logger Setup	Form that allows users to configure the graph settings: • *Total Plot Points:* number of data points (x,y) displayed on a graph • *Total Log Points:* number of records stored in the database, depending on the interval setting • *Interval:* time interval between each recorded data point
Download Logged Data	Download the recorded data (based on Total Log Points) in CSV format.
View Gas Compositions	Generate a graph based on the data from the uploaded CSV file.
Change Password	Allows users to change their password.
Logout	Log out of the system.
Download User-Log	Download records of user login activity.
Clear User-Log	Clear all user login records.
About	Display information about the developer of the monitoring system.

3.System Integration

The overall remote monitoring system uses a Wide Area Network (WAN), allowing any computing device connected to the WAN to access the monitoring system's website. As shown in Fig. 6, each sewer facility equipped with the remote monitoring system can send gas composition data to the cloud database (MySQL). Clients can access the website via a computer, laptop or mobile phone to view gas composition easily anytime and anywhere in real time. This remote monitoring system enables the headquarters to monitor sewer facilities located in different districts where sensing units are installed.

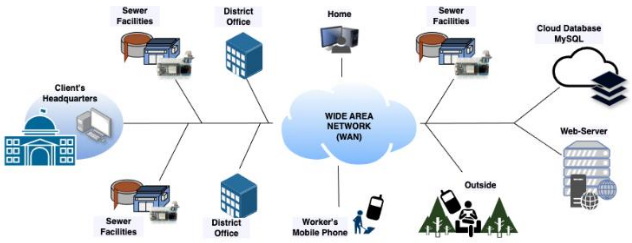


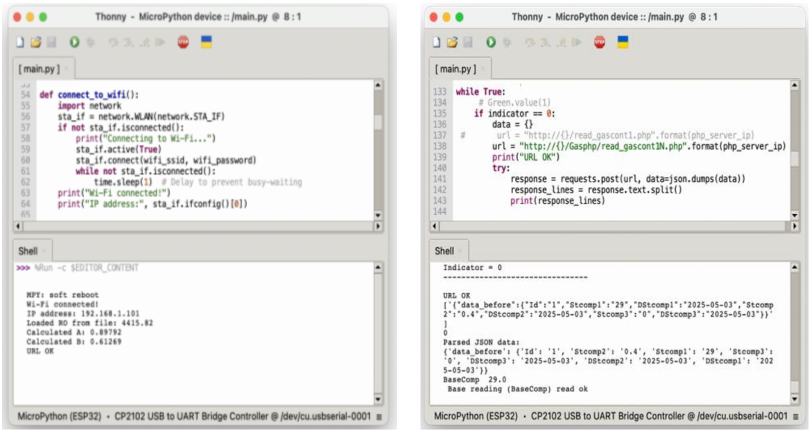
Fig. 6. General architecture of the overall monitoring system.

For the development stage, the monitoring system is tested within a Local Area Network (LAN) before transitioning to a cloud hosting server. The program, database and PHP scripts are stored on a laptop, and all sensing units are powered via USB-C cables, as shown in Fig. 7.

Fig. 7. The experimental setup of the GSU (MQ-4, MQ-136, MQ-137) and WSU, powered via a Type-C connection.

Figure 8 illustrates sections of the code for (a) establishing the Wi-Fi connection and (b) retrieving the BaseComp data. The code is structured to first establish a Wi-Fi connection. If the connection is successful, the ESP32 sends a POST request to a PHP server that hosts the necessary scripts to retrieve the base reading (BaseComp). The calculation of gas concentration in ppm consists of two steps: performing ADC conversion and then subtracting the result by the BaseComp value.

The final result represents the gas composition in ppm. Once the gas composition is calculated, the data is stored in the MySQL database. Essentially, there are two PHP scripts for each gas sensor - one to retrieve the base reading and another to update the gas composition.



(a) **(b)**

Fig. 8. Serial monitor output in Thonny IDE showing part of (a) the Wi-Fi connection code and (b) the base reading retrieval code.

4. Results and Discussion

The user management login system is to ensure appropriate security levels for administrators, supervisors and operators. Administrators are responsible for managing system settings and configuring the overall behaviour of the website. Supervisors, typically engineers, have access to edit device configurations, such as the calibration settings of the MQ gas sensors. Operators, who are sewer workers handling tasks on-site, are limited to accessing and viewing gas composition data and real-time trend graphs.

From the homepage view, users can monitor the current gas composition levels and determine whether they have exceeded the allowable permissible limits, as shown in Fig. 9. This enables early preparation and mitigation efforts, especially in the event of complaints. A home dashboard specifically designed for guest users can be implemented to allow them to view selected information, depending on the permissions defined by the system administrator. Figure 10 shows the website homepage viewed via mobile phone.

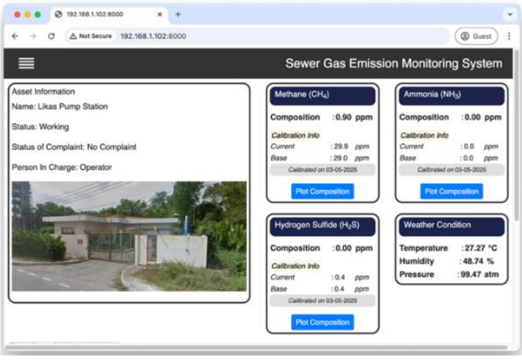
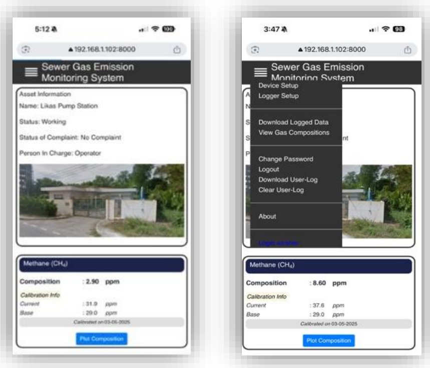


Fig. 9. Website homepage viewed on a computer.



(a) **(b)**

Fig. 10. Website homepage viewed on (a) mobile phone and (b) dropdown menu button.

The Device Setup is used for gas sensor calibration settings, where users can input the calibration data (BaseComp), as shown in Fig. 11. Based on these base readings, the backend code calculates the gas composition for each gas, which is then displayed on the homepage.

The Logger Setup allows users to configure the data logging interval for plotting gas composition graphs. In this system, the interval is set to 4.0 seconds, meaning the graph is updated with new data points in real time every 4.0 seconds, as shown in Fig. 12. The graph serves as a valuable tool for identifying gas emission trends and correlating them with peak and off-peak usage of bathroom or toilet facilities. Any abnormal spike in gas emissions outside the expected trend may indicate potential issues at the site, such as a sewer pump malfunction, manhole blockage or network failure.

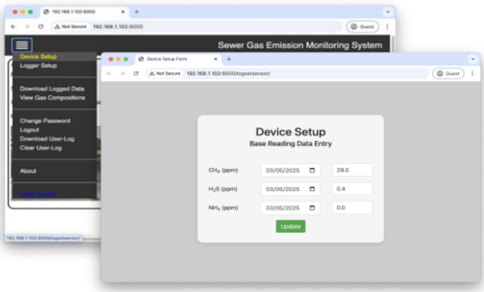
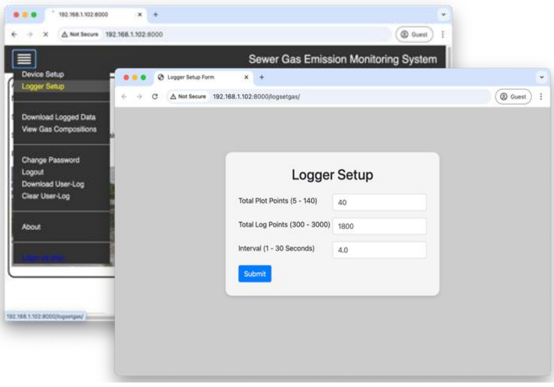
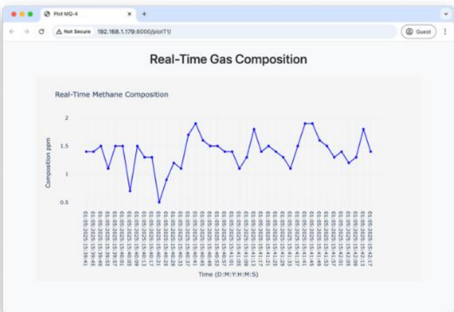


Fig. 11. Dropdown menu button showing the Device Setup form which is used for entering base reading data.



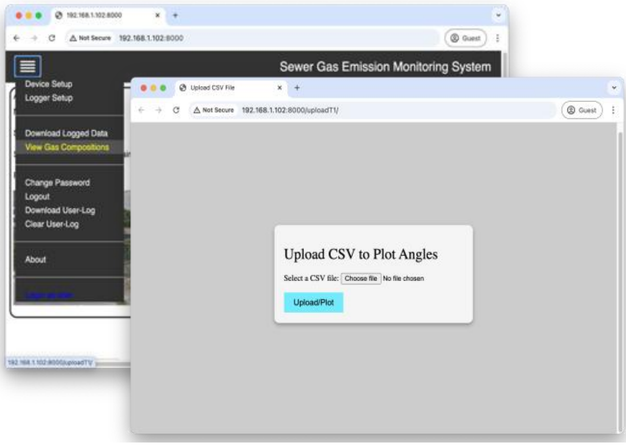
(a)



(b)

Fig. 12. (a) Logger Setup form for configuring graph settings.
(b) Real-time graph plotted based on the Logger Setup configuration.

For trend analysis, users can download the data in a CSV file by clicking on Download Logged Data and view the recorded values by selecting the View Gas Composition option, as shown in Fig. 13. Other items in the dropdown menu include Change Password, Login/ Logout, Download User-Log, Clear User-Log and About.



(a)

(b)

Fig. 13. View Gas Composition page showing (a) the CSV upload section, and (b) the graph generated from the uploaded historical data.

The overall development of the system was carried out using Linux, although developers have the flexibility to use Windows or other operating systems. The system is designed to be universal and with a custom-built IoT platform, users can add and display any type of information without limitations. Additionally, the system can be easily migrated to cloud hosting services, offering greater scalability for data management, particularly when monitoring a larger number of sewer assets.

One of the key factors contributing to the successful implementation of the proposed monitoring system is the effective integration between the sensing hardware and the database system. The ESP32 microcontroller, equipped with built-in Wi-Fi capability, enables stable communication with the database through HTTP requests. In addition, the use of a structured MySQL database allows efficient storage and retrieval of sensor data, supporting both real-time monitoring and historical trend analysis. Furthermore, the custom-designed web interface developed using Django provides flexible data visualization and system configuration compared with ready-made commercial platforms. Collectively, these components enhance the overall effectiveness of the monitoring system by enabling continuous monitoring of sewer gas emissions.

Despite the successful development and integration of the monitoring system, several limitations should be acknowledged. The gas sensing units developed in this study primarily serve as prototypes to demonstrate the functionality of data acquisition and transmission to the database and web interface. Therefore, the sensor readings are intended to verify the successful operation of the web-based monitoring system rather than to provide highly accurate representations of gas compositions. In addition, the current system relies on a continuous power supply and Wi-Fi connectivity for the sensing units. This requirement may pose challenges for installations in remote sewer facilities where a stable power source is not readily available or where internet connectivity is limited.

5. Conclusions

This study successfully developed a web-based remote monitoring system with a custom-designed database system. The system allows users to remotely monitor gas composition and store sewer data in the MySQL database. Users can download the historical data and perform trend analysis, which aids in decision-making and in convincing top management. The monitoring system is always accessible via any computing device, provided there is internet connectivity in the area, which is particularly beneficial for monitoring in urban areas.

Future work should therefore include further calibration using known reference gas concentrations to improve measurement accuracy for practical deployment. Real-world deployment at actual sewerage facilities should also be conducted to evaluate the system performance under varying operational and environmental conditions. Such deployment would allow further validation of sensor reliability, network stability and system scalability when multiple sensing units are installed across different locations. In addition, an alert system, such as sending messages to sewer workers for emergency evacuation, can also be added to the monitoring system. The web interface is a continuous work in progress and can be enhanced with more comprehensive features and informational content based on user requirements. This research shows the potential of IoT in modernizing urban sewerage systems and the possibilities for seamless integration with smart city concepts.

References

1. Kumar, L.V.S.; and Ayyarao, T.S.L.V. (2024). Real time environmental monitoring with Raspberry Pi 3B+ and MATLAB for enhanced worker safety through IoT. *Proceedings of the 2024 2nd International Conference on Cyber*

Physical Systems, Power Electronics and Electric Vehicles (ICPEEV). Hyderabad, India, 1-6.

2. Park, K.; Lee, H.; Phelan, S.; Liyanaarachchi, S.; Marleni, N.; Navaratna, D.; Jegatheesan, V.; and Shu, L. (2014). Mitigation strategies of hydrogen sulphide emission in sewer networks - a review. *International Biodeterioration & Biodegradation*, 95(Part A), 251-261.
3. U.S. Environmental Protection Agency (EPA). Ammonia. Retrieved April 14, 2025, from https://www.epa.gov/caddis/ammonia
4. Government of UK. Ammonia: general information. Retrieved May 17, 2025, from https://www.gov.uk/government/publications/ammonia-properties-incident-management-and-toxicology/ammonia-general-information
5. Li, H.; You, L.; Du, H.; Yu, B.; Lu, L.; Zheng, B.; Zhang, Q.; He, K.; and Ren, N. (2024). Methane and nitrous oxide emissions from municipal wastewater treatment plants in China: A plant-level and technology-specific study. *Environmental Science and Ecotechnology*, 20, 100345.
6. He, S.; Su, L.; Fan, H.; and Ren, R. (2019). Methane explosion accidents of tunnels in SW China. *Geomatics, Natural Hazards and Risk*, 10(1), 667-677.
7. Barsan, M.E. (2007). NIOSH Pocket Guide to Chemical Hazards. Retrieved May 17, 2025, from http://www.cdc.gov/niosh/npg/npg.html
8. Ajiboye, A.T.; Opadiji, J.F.; Yusuf, A.O.; and Popoola, J.O. (2021). Analytical determination of load resistance value for MQ-series gas sensors: MQ-6 as case study. *Telkomnika Telecommunication Computing Electronics and Control*, 19(2), 575-582.
9. Fine, G.F.; Cavanagh, L.M.; Afonja, A.; and Binions, R. (2010). Metal oxide semi-conductor gas sensors in environmental monitoring. *Sensors*, 10(6), 5469-5502.
10. Huang, C.W.; Wu, C.Y.; and Hsueh, T.J. (2025). A room temperature ZnO:Ga NWs&NSs/MEMS H2S gas sensor. *Materials Science in Semiconductor Processing*, 187, 109149.
11. Fakra, D.A.H.; Andriatoavina, D.A.S.; Razafindralambo, N.A.M.N.; Amarillis, K.A.; and Andriamampianina, J.M.M. (2020). A simple and low-cost integrative sensor system for methane and hydrogen measurement. *Sensors International*, 1, 100032.
12. Nagahage, I.S.P.; Nagahage, E.A.A.D.; and Fujino, T. (2021). Assessment of the applicability of a low-cost sensor–based methane monitoring system for continuous multi-channel sampling. *Environ Monit Assess*, 193, 509.
13. Wang, C.; Yin, L.; Zhang, L.; Xiang, D.; and Gao, R. (2010). Metal oxide gas sensors: Sensitivity and influencing factors. *Sensors*, 10(3), 2088-2106.
14. Subbarayudu, P.; Asha Latha, B.; Bhavya Sree, D.; Pavan Raj, S.; and Tejesh, H. (2019). Automatic Gas leak detection and prevention using Arduino and GSM Module. *Journal of Emerging Technologies and Innovative Research (JETIR)*, 6(3), 45-49.
15. Song Z.; and Abu Ebayyeh, A.A.R.M. (2025). Learning from synthesized data for quality assurance in open-source microcontroller manufacturing. *Measurement,* 253, 117490.

16. Sreerama Murty, M.; Rao Dhanikonda, S.; Sowjanya, P.; Jagdish Kumar, P.; Chandra, N.S.; and Dastagiraiah, C. (2024). Smart gas leakage detector using IoT sensors. *International Journal of Intelligent Systems and Applications in Engineering (IJISAE)*, 12(4), 2613-2620.
17. Banupriya, N.; Gokul, G.; Naveen, K.S.; Narayani, S.; and Afsar, M.A. (2023). Wearable IoT gas monitoring for sewers safety. *Proceedings of the 2023 2nd International Conference on Automation, Computing and Renewable Systems (ICACRS).* Pudukkottai, India, 268-272.
18. Mudholkar, P.; Mudholkar, M.; Prabhu, S.; B., S.S.; Rahaman, S.M.A.; and Shanil, A. (2021). Environmental monitoring system based on Internet of Things. *Design Engineering*, 8, 425-438.
19. Fahim, M.I.; Tabassum, N.; Habibullah, A.A.; Sarker, A.; Nahid, S.I.; and Monirujjaman Khan, M. (2021). Design of an IoT based gas wastage monitoring, leakage detecting and alerting system. *Proceedings of the 2021 IEEE 12th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*. New York, USA, 591-597.
20. Ramesh, K.B.; and Saran, S. (2025). Drone-based aerial surveillance and real-time hazardous gas leakage detection system with Power BI dashboard integration for enhanced environmental safety. *Proceedings of the 2025 6th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI)*. Goathgaun, Nepal, 541-547.
21. Abraham, S.; and Li, X. (2014). A cost-effective wireless sensor network system for indoor air quality monitoring applications. *Procedia Computer Science*, 34, 165-171.
22. Hercog, D.; Lerher, T.; Truntič, M.; and Težak, O. (2023). Design and implementation of ESP32-based IoT devices. *Sensors*, 23(15), 6739.
23. Bell, C.A. (2013). *Beginning sensor networks with Arduino and Raspberry Pi*, Apress.
24. Xie, Y.; Gao, M.; Luo, F.; Zhou, A.; Yang, Y.; Hu, J.; Jiang, W.; and Ye, Y. (2025). Django-based framework database for leakage detection using machine learning for water distribution networks. *Engineering Applications of Artificial Intelligence*, 149, 110525.
25. Darjat; Ayub, A.Z.; Sudjadi; and Harum, A.S. (2020). Design of multisensor data acquisition system (oxygen, hydrogen, temperature and pressure) through android-based website. *Proceedings of the 2020 7th International Conference on Information Technology, Computer and Electrical Engineering (ICITACEE)*. Semarang, Indonesia, 121-125.
26. Shah, J.; and Mishra, B. (2016). IoT enabled environmental monitoring system for smart cities. *Proceedings of the 2016 International Conference on Internet of Things and Applications (IOTA).* Pune, India, 383-388.
27. Asthana N.; and Bahl, R. (2025). IoT device for sewage gas monitoring and alert system. *Proceedings of the 2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT)*. Chennai, India, 1-7.
28. Ali, A.; Kazmi, S.A.A.; and Khan, M.Z. (2024). Real-time gas and air quality monitoring and controlling by IoT solution. *Proceedings of the 2024 19th International Conference on Emerging Technologies (ICET)*. Topi, Pakistan, 1-6.

29. Gandi, N.; Karri, J.S.; Nune, S.S.S.; Ravada, L.; and Darapu, D. (2024). Enhanced urban infrastructure: IoT-based sewer vault monitoring and cautioning system. *Proceedings of the 2024 2nd International Conference on Inventive Computing and Informatics (ICICI)*. Bangalore, India, 552-555.
30. Arul, A.; Subathra; Rahin, R.; Ramalingam, V.; and ArunKumar, G. (2023). Wireless IoT gas and smoke detection system with ESP8266 and Blynk integration. *Proceedings of the 2023 1st International Conference on Optimization Techniques for Learning (ICOTL)*. Bengaluru, India, 1-6.
31. Jumaa, N.K.; Abdulkhaleq, Y.M.; Nadhim, M.A.; and Abbas, T.A. (2022). IoT based gas leakage detection and alarming system using Blynk platforms. *Iraqi Journal for Electrical and Electronic Engineering*, 18(1), 64-70.
32. Anusha, P.; Balaji, A.; Nithyasundari, B.; Patil, Y.D.; and Ravi, S. (2024). Experimental evaluation of an Internet of Things powered sewage gas monitoring and reporting system using artificial intelligence strategy. *Proceedings of the 2024 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)*. Chennai, India, 1-6.
33. Vidhya, A.; Sudarmani, R.; Vaisali, S.; and Aishwarya, K. (2022). Development of manhole cover detection and continuous monitoring of hazardous gases using WSN and IoT. *Proceedings of the 2022 6th International Conference on Computing Methodologies and Communication (ICCMC)*. Erode, India, 202-206.
34. El-Khozondar, H.J.; Mtair, S.Y.; Qoffa, K.O.; Qasem, O.I.; Munyarawi, A.H.; Nassar, Y.F.; Bayoumi, E.H.E.; Abd El Halim, A.A.E.B. (2024). A smart energy monitoring system using ESP32 microcontroller. *E-Prime - Advances in Electrical Engineering, Electronics and Energy*, 9, 100666.