Gas Monitor: Real-Time SQL Storage

Project Overview:

This project involves sending sensor data from an Arduino UNO to a MySQL server using Node-RED. The MQ-2 Gas Sensor is used to detect gas levels, and the Arduino sends the sensor data to a MySQL database for monitoring. This allows for real-time tracking of gas levels and can be expanded for use in various applications such as safety monitoring in industrial or home environments.

Technology Used:

- Arduino UNO: A microcontroller board used to interface with the MQ-2 Gas Sensor and read analog values.
- MQ-2 Gas Sensor: A sensor capable of detecting a wide range of gases, including carbon monoxide, methane, and smoke.
- Node-RED: A flow-based development tool for wiring together hardware devices, APIs, and online services. It helps to easily integrate sensor data with the MySQL database.
- MySQL: A relational database management system used to store and retrieve sensor data for monitoring and analysis.
- JavaScript (Node.js): Used to install and run Node-RED and manage the flow of data from the Arduino to the database.

Objective of the Project:

The primary objective of this project is to collect real-time gas sensor data from the MQ-2 sensor, process it using Node-RED, and store the data in a MySQL database. This enables easy monitoring of gas levels over time, with potential applications in safety, health monitoring, and environmental control systems. The goal is to create an efficient and automated way of monitoring sensor data remotely.

Challenges and Solutions:

1. Data Communication Between Arduino and Node-RED:
    - Challenge: Ensuring seamless communication between Arduino UNO and Node-RED.
    - Solution: Use the Serial In node in Node-RED to receive data from the Arduino's serial output. This allows the data sent from the Arduino to be read in Node-RED and processed.
2. Data Formatting and Transfer:
    - Challenge: The data from the sensor needs to be correctly formatted before sending to the MySQL database.
    - Solution: Use the JSON node to parse the data from Arduino, ensuring that it is properly formatted and readable before sending it to the database.
3. Database Integration:

- Challenge: Storing and retrieving real-time sensor data in MySQL.
- Solution: Use the MySQL node in Node-RED to insert the sensor data into a MySQL database. This allows data to be stored in real-time and queried for analysis later.

4. Sensor Calibration and Accuracy:
   - Challenge: Ensuring the gas sensor provides accurate readings.
   - Solution: Proper calibration of the MQ-2 sensor and testing under various conditions to ensure accurate gas concentration levels.
   -

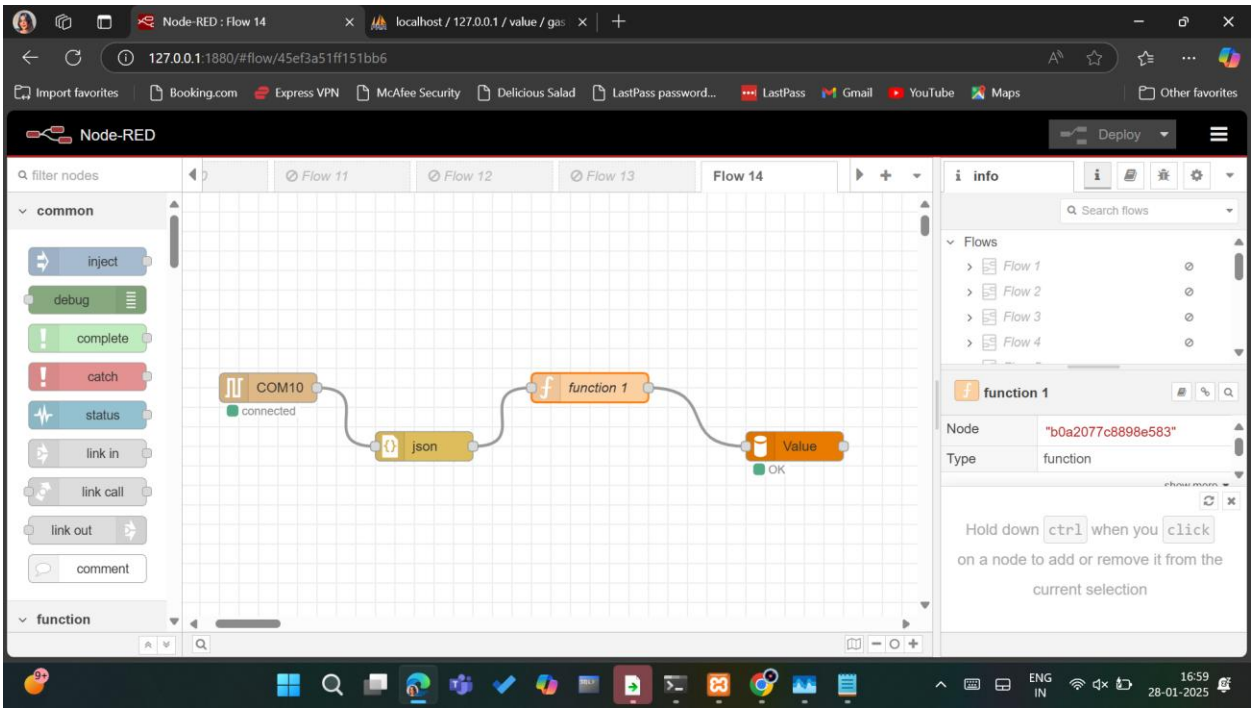
Key Features:

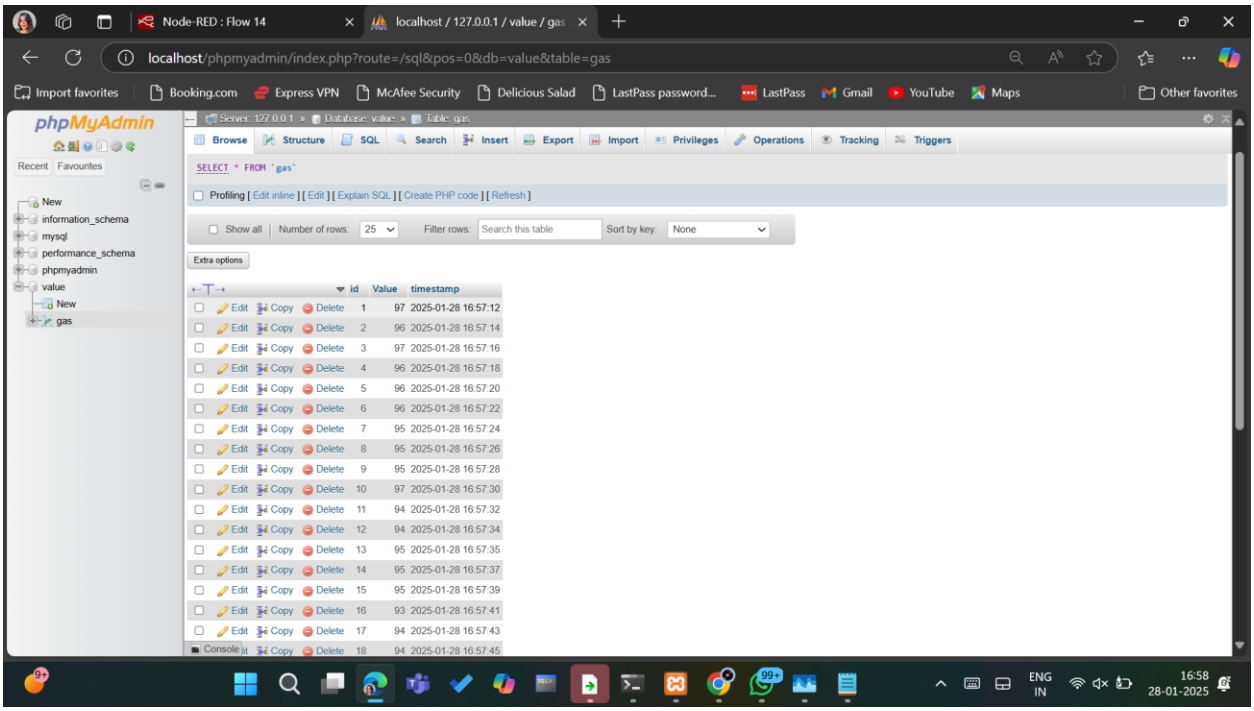
- Real-time Data Collection: Continuously collects gas sensor data from the MQ-2 Gas Sensor and sends it to the MySQL database for real-time monitoring.
- Automated Data Entry: Node-RED automates the process of reading from the Arduino and storing the data in MySQL, reducing the need for manual intervention.
- Scalability: The system is scalable, meaning additional sensors or other data types can be added easily to the database.
- Remote Monitoring: With the sensor data stored in MySQL, users can query the database remotely for analysis and monitoring purposes.

- Integration Flexibility: Node-RED allows easy integration with other devices, APIs, or web interfaces, enabling future expansion of the project.

Project Link and Screenshot :

GitHub Repository: https://github.com/Aakanksssha/Gas-Monitor-Real-Time-SQL-Storage.git




Node-RED : Flow 14
localhost / 127.0.0.1 / value / gas
localhost/phpmyadmin/index.php?route=/sql&pos=0&db=value&table=gas
Import favorites
Booking.com
Express VPN
McAfee Security
Delicious Salad
LastPass password...
LastPass
Gmail
YouTube
Maps
Other favorites
phpMyAdmin
Recent
Favourites
New
information_schema
mysql
performance_schema
phpmyadmin
value
New
gas
Server: 127.0.0.1 » Database: value » Table: gas
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Tracking
Triggers
SELECT * FROM `gas`
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all
Number of rows: 25
Filter rows: Search this table
Sort by key: None
Extra options
id Value timestamp
1 97 2025-01-28 16:57:12
2 96 2025-01-28 16:57:14
3 97 2025-01-28 16:57:16
4 96 2025-01-28 16:57:18
5 96 2025-01-28 16:57:20
6 96 2025-01-28 16:57:22
7 95 2025-01-28 16:57:24
8 95 2025-01-28 16:57:26
9 95 2025-01-28 16:57:28
10 97 2025-01-28 16:57:30
11 94 2025-01-28 16:57:32
12 94 2025-01-28 16:57:34
13 95 2025-01-28 16:57:35
14 95 2025-01-28 16:57:37
15 95 2025-01-28 16:57:39
16 93 2025-01-28 16:57:41
17 94 2025-01-28 16:57:43
18 94 2025-01-28 16:57:45
Console
ENG
IN
16:58
28-01-2025