CHAPTER 1

INTRODUCTION

Gas leak refers to a leak of natural gas or another gaseous product from a pipeline or other containment into any area where the gas should not be present. Gas leaks can be hazardous to health as well as the environment. Even a small leak into a building or other confined space may gradually build up an explosive or lethal concentration of gas. Leaksof natural gas and refrigerant gas into the atmosphere are especially harmful due to their global warming potential and ozone depletion potential.

Leaks of gases associated with industrial operations and equipment are also generally known as fugitive emissions. Natural gas leaks from fossil fuel extraction and use are known as fugitive gas emissions. Such unintended leaks should not be confused with similar intentional types of gas release, such as: Every year, numerous people lose their lives to gas-related incidents. It is important to take necessary precautions to avoid such accidents. Alcohol can be dangerous and should be handled with care and proper precautions. It is a flammable liquid that can easily catch fire if it comes into contact with an ignition source. Thus, it is important to store alcohol in a cool, dry and well-ventilated area, away from heat sources and open flames.

In addition to its use as a source of energy, methane also has an important role in the natural gas industry. It is the primary component of natural gas, which is used for heating and cooking in homes and businesses, as well as in transportation and industrial operations. Methane is also used as feed stock in the production of chemicals such as methanol, ammonia, and hydrogen. However, the release of methane into the atmosphere is a major contributor to climate change. Methane is a potent greenhouse gas, and its warming effect is estimated to be up to 84 times stronger than carbon dioxide over a 20-year time frame. Methane emission come from a variety of sources, including the oil and gas industry, agriculture, and waste management.

To address this issue, efforts are being made to develop technologies and practices that can reduce methane emissions in various industries. In our project when this alcohol or methane gases are released and detected by the sensor automatically solutions are pumped by the pump from the container and reduces the content ration of those gases.

1.1 What is IoT?

The Internet of Things (IoT) describes the network of physical objects “things” that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet. These devices range from ordinary household objects to sophisticated industrial tools.

Over the past few years, IoT has become one of the most important technologies of the 21st century. Now that we can connect everyday objects—kitchen appliances, cars, thermostats, baby monitors—to the internetvia embedded devices, seamless communication is possible between people, processes, and things. By meansof low-cost computing, the cloud, big data, analytics, and mobile technologies, physical things can share andcollect data with minimal human intervention. In this hyperconnected world, digital systems can record, monitor, and adjust each interaction between connected things. The physical world meets the digital world—and they cooperate.

The Internet of Things (IoT) describes the network of physical objects - "things"- that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet. These devices range from ordinary household objects to sophisticatedindustrial tools. With more than 7 billion connected IoT devices today, experts are expecting this number togrow to 10 billion by 2020 and 22 billion by 2025. Oracle has a network of device partners.

1.1.1 Why is Internet of Things (IoT) so important?

Over the past few years, IoT has become one of the most important technologies of the 21st century. Now that we can connect everyday objects—kitchen appliances, cars, thermostats, baby monitors—to the internet via embedded devices, seamless communication is possible between people, processes, and things. By means of low-cost computing, the cloud, big data, analytics, and mobile technologies, physical things can share and collect data with minimal human intervention. In this hyperconnected world, digital systems can record, monitor, and adjust each interaction between connected things. The physical world meets the digital world—and they cooperate.

1.1.2 What technologies have made IoT possible?

While the idea of IoT has been in existence for a long time, a collection of recent advances in a number of different technologies has made it practical. Access to low-cost, low-power sensor technology.Affordable and reliable sensors are making IoT technology possible for more manufacturers.

- Connectivity: A host of network protocols for the internet has made it easy to connect sensors to the cloud and to other "things" for efficient data transfer.
- Cloud computing platforms: The increase in the availability of cloud platforms enables both businessesand consumers to access the infrastructure they need to scale up without actually having to manage it all.
- Machine learning and analytics: With advances in machine learning and analytics, along with access

and vast amounts of data stored in the cloud, businesses can gather insights faster and more easily. The emergence of these allied technologies continues to push the boundaries of IoT and the data produced by IoT also feeds these technologies.

- Conversational artificial intelligence (AI): Advances in neural networks have brought natural-language processing (NLP) to IoT devices (such as digital personal assistants Alexa, Cortana, and Siri) and made them appealing, affordable, and viable for home use.

1.1.3 What is industrial IoT?

Industrial IoT (IIoT) refers to the application of IoT technology in industrial settings, especially with respect to instrumentation and control of sensors and devices that engage cloud technologies. Recently, industries have used machine-to-machine communication (M2M) to achieve wireless automation and control. But with the emergence of cloud and allied technologies (such as analytics and machine learning), industries can achieve a new automation layer and with it create new revenue and business models. IIoT is sometimes called the fourth wave of the industrial revolution, or Industry 4.0.

The following are some common uses for IIoT:

- Smart manufacturing
- Connected assets and preventive and predictive maintenance
- Smart power grids
- Smart cities
- Connected logistics
- Smart digital supply chains

1.1.4 How Does an IoT Network Work?

With the basics covered, let’s explore how exactly IoT networks work, what sensors do, and how administrators manage them.

IoT Sensors

IoT networks rely on small inexpensive sensors to collect information about the environment. For example, farmers use IoT sensors to monitor moisture levels while industrial plants use similar sensors to monitor pipe pressure. IoT sensors are highly configurable and can monitor hundreds of different changes. A few examples of what IoT sensors can monitor include the following:

- Geolocation
- Fluid levels

- Temperature, humidity, and other atmospheric conditions
- Electrical currents
- Data packets
- The presence of particular gases or chemicals

IoT Connectivity

IoT sensors continuously send data back to the cloud or an edge computing device for processing. IoT sensors typically use little power and send small amounts of data rather than large streams of information. Businesses that require the lowest latency and fastest response time often opt for edge computing as it shortens the distance between the sensor and the server. Depending on the technology and use case, businesses can choose from various IoT networks to meet their goals. The two most common ways sensors send their data are through Wi-Fi or cellular connection. We'll touch more on the different types of networks and their advantages later on.

IoT Processing

Once the data is collected, software processes and records that data in the cloud or on an edge server. Many platforms use artificial intelligence and machine learning to take action when specific data is sent from a sensor. Enterprises pair IoT networks with automation to orchestrate device management in a way that's affordable, predictable, and highly scalable. IoT management systems can process data from various systems allowing enterprises to monitor everything from machine maintenance to the weather outside.

IoT Interface

Administrators simply set rules for the software to follow and teach the software what actions to trigger when certain conditions are met. In some cases where automation isn't appropriate, the software can automatically alert a human when a specific event occurs. For instance, in an industrial IoT setting sensors can automatically create a maintenance request to change a machine's oil when it reaches a certain level. If that request isn't fulfilled and the device is in danger of overheating, the sensors can send a more urgent alert and shut down the machine to prevent significant damage if needed.

The back-end interface allows administrators to set conditional rules and service levels to shape how monitoring and automation occur. These interfaces have come a long way since the early days and are much easier to use and navigate.

Big Data: It refers to the massive amount of structured and unstructured data that can be accessed and analyzed to extract insights and make informed decisions. Bigdata technologies enable the processing, storage, and analysis of this data in real time or near-real time, using advanced algorithms and statistical models.

Embedded Systems: An embedded system is a computer system designed to perform specific tasks with a dedicated function within a larger mechanical or electrical system. It is a combination of hardware and software that is designed to perform a specific function often with real-time computing constraints.

Security Protocols and Architectures: Security protocols and architectures are designed to secure IoT systems from different vulnerabilities, such as unauthorized access, theft tampering, and hacking. These include encryption, authentication, and access control mechanisms to ensure data privacy, integrity, and confidentiality.

Protocols enabling Communication: Communication protocols are necessary for IoT devices to be able to communicate with each other to perform specific tasks. These include wireless protocols such as Bluetooth, Zigbee, and Wi-Fi, as well as wired protocols such as Ethernet and RS- 485.Web services are software systems designed to support interoperable machine-to- machine interaction over a network. They enable different IoT devices and systems to access and share data with each other using web-based technologies. Internet and Search Engines: The Internet and search engines facilitate the connectivity and communication between different IoT systems, devices, and users. They allow the devices to access the vast amounts of data available on the web and provide users with the ability to find and access specific information relevant to their needs.

1.2 Aim and Objective

The main objective of this project is to build a existing IoT device is related to kitchen, room vehicles but not related to industry. Mainly for hazardous gases or it's components. Sending message using NodeMCU module.

Our proposed system detects the smoke, alcohol, and automatically pumps the solution resided in the container and reduce the content of gases in the atmosphere. These applets consist of a trigger (the “if this” part) and an action (the “then that” part). For example, if the temperature sensor detects a certain temperature (trigger), then the air conditioning turns on (action). IFTTT can be used with the Arduino IoT Cloud to integrate IoT projects with other services and devices, such as Twitter, Instagram, and Philips Hue light. This allows for greater automation and customization of IoT projects. Our proposed system detects the smoke, alcohol, and automatically pumps the solution resided in the container and reduce the content of gases in the atmosphere. The objectives for developing the project are as follows:

- To improve the existing system using IoT technology.
- To reduce the workload of industries to detect the fault.
- We are supposed to learn the concept of IoT and how it can be utilized to work on different sectors.

1.3 Applications

An IoT powered gas detection solution uses gas sensors to identify the presence of toxic gases such as CO2, CO, NOx in the industrial facilities. Especially, in the oil and gas industry where many gaseous products like propane, butane, and hydrogen are manufactured at a greater level. Hence, the chances of gas explosions are higher as these gases are easily combustible in the oxygen-rich environment. Apart from these, toxic gases like hydrogen sulfide (H2S) is produced during refining processes that might harm the workers' health. Thus, it becomes a necessity to keep a real-time check on gas production. If these toxic gases are released untreated, their harmful contaminants result in air pollution and acid rains.

Harmful Gas Detection

The sensing of toxic gases such as H2S, Methane, and CO is of great importance in any industry to avoid unwanted leakage and consequences like poisoning or explosions. The presence of these gases can be easily detected in the industrial facilities and commercial buildings with the help of IoT-powered gas monitoring solution. Moreover, a gas detector or sensor device is a crucial part to carry out safe industrial operations. The sensor-enabled solution helps prevent the high risk of gas explosions and affecting any casualties within and outside the premises.

- Get real-time alerts about the gaseous presence in the atmosphere
- Supervise gas concentration levels
- Ensure worker's health
- Real-time updates about leakages
- Cost-effective installation
- Data analytics for improved decisions
- Get immediate gas leak alerts

CHAPTER 2

LITERATURE SURVEY

2.1 Existing System

A great deal of work having been gotten really intended to date the territory of IoT systems. A concise audit of the important work in the field of Gas detection System is depicted in this part. Numerous other Measurable methodologies, for example

[1] Mohammad Monirujjaman Khan, Sensor-Based Gas Leakage Detector System, MDPI, 14 November 2020.

Additionally, the system includes a relay and a pump controller. When gas is detected, the relay will be triggered to activate the pump and start pumping the gas out of the atmosphere. The pump controller adjusts the speed of the pump based on input from various sensors such as flow meters and level sensors. This helps ensure the gas is quickly and efficiently removed from the atmosphere. Once the gas is removed, the system will turn off the pump and stop the alarm. The Node MCU is used for connecting the system to the internet and enabling remote monitoring and control. It includes firmware that runs on the ESP8266 Wi-Fi SoC from Expressive Systems, allowing the system to connect to the internet and send notifications or alerts. The user can monitor the system status and control the pump or alarm remotely through the IoT cloud platform's open editor. This provides greater flexibility and convenience, allowing the user to manage the system from anywhere with an internet connection. Overall, this system provides a reliable and efficient solution for gas detection and removal, with the added convenience and flexibility of remote monitoring and control. It can be used in a variety of settings, including homes, offices, and industrial facilities.

[2] Afsana Mim Anika Dept. of Computer Science and Engineering Daffodil International University Dhaka, Bangladesh, Gas Leakage with Auto Ventilation and Smart Management System Using IoT, IEEE 2021.

The gas sensor (MQ2 Sensor) is used to detect the presence of flammable gases such as LPG, propane, and methane. Once the gas is detected, the sensor sends a signal to the microcontroller to activate the alarm and display the gas level on the LCD screen.

The fire sensor is used to detect the presence of fire or high temperatures. When the sensor detects heat or fire, it sends a signal to the microcontroller to activate the alarm and display the fire level on the LCD screen. The proposed system also includes an auto air ventilation system that detects the quality of air inside a room and automatically switches on the ventilation system to provide fresh air.

Similarly, the water flow system detects the water level in a tank and automatically controls the water pump to maintain a certain level. All of these systems are controlled by an Arduino microcontroller, which is programmed to monitor and control the different sensors and systems. The GSM module is used to send a message to the owner's phone if there is any emergency detected by the sensors. This allows the owner to take action immediately and prevent any accidents.

In conclusion, the proposed system is an advanced and innovative way to ensure the safety of a home or building. In conclusion, the proposed system is an advanced and innovative way to ensure the safety of a home or building.

[3]Zaw Lin Oo Ministry of Science and Technology, Myanmar Theint Win Lai KBTC University, IoT Based LPG Gas Level Detection & Gas Leakage Accident Prevention with Alert System , Journal Vol. 9, No. 4, October 2021.

Liquefied Petroleum Gas (LPG) is widely used for cooking fuel in developing countries for economic reasons, for energy-rich fuel source that contains the higher calorific value, for clean fuel with low carbon emission and for a portable that is available in even the faraway of areas. Therefore the proposed gas leakage detection and monitoring system for the gasoline content present in household LPG cylinder is developed. Usually, the capacity of LPG in cylinder is not determined in an exact manner and a cylinder when the gas is about to empty will be a difficult situation for the one who uses LPG gas for cooking continuously.

This paper will solve the problem for not only detects any leakage of the LPG gas but removes automatically the leaked gas to prevent from fire accidents and it also alerts the user by creating alarm song and warning signal. This proposed method consists of gas leakage detection system, gas cylinder weight measurement module, alert system, microcontroller, IoT system for gas consumption record and cloud communications platform for SMS.

[4]Dr. Chalasani Srinivas ,Mohan Kumar.Ch , Toxic Gas Detection and Monitoring and utilizing Internet of Things International Journal of Civil Engineering and Technology (IJCIET) Volume 8, Issue 12, December 2017.

Harmful gas leakage accidents are the main reason for workers death in industries which work mainly using chemicals. Gas leakage can be easily detected and controlled by using latest trends in information technology by applying internet of things. This project intended to avoid industrial accidents and to monitor harmful gases and to intimate alert message to safety control board of industry using Arduino Uno R3 and internet of things. Arduino Uno R3 board is used as central microcontroller which is connected with sensor. Such as temperature, gas sensor, alcohol sensor which can continuously monitor respective environmental parameters. Hence this device may be used as multi gases detection apparatus more over the rate of response is high. An alarm is produced instantly if the level of the gases goes above the normal level means indication through the internet specific receiver section. Data received by sensor is stored in internet which can be used for further processing and it can be analyzed for improving safety regulations. This model can be future extended for providing better living environment for people in and around industries with a pollution controlled environment.

[5] Rohan Chandra Pandey1, Manish Verma2, Lumesh Kumar Sahu3,1Kalinga University, PG Student, Vlsi and Embedded System, Internet of Things (IOT) Based Gas Leakage Monitoring and Alerting System with MQ-2 Sensor, IJEDR | Volume 5, Issue 2, 2017.

Safety plays a major role in today's world and it is necessary that good safety systems are to be implemented in places of education and work. This work modifies the existing safety model installed in industries and this system also be used in homes and offices. The main objective of the work is designing microcontroller based toxic gas detecting and alerting system. The hazardous gases like LPG and propane were sensed and displayed and notify each and every second in the LCD display. If these gases exceed the normal level then an alarm is generated immediately and also an alert message (Email) is sent to the authorized person through the INTERNET and used ARM development board. The advantage of this automated detection and alerting system over the manual method is that it offers quick response time and accurate detection of an emergency and in turn leading faster diffusion of the critical situation.

[6] Noor Kareem Jumaa*, Younus Mohammed Abdulkhaleq, Muntadher Asaad Nadhim, Tariq Aziz Abbas, Al-Mansour University College, Computer Technology Eng. Dept., Baghdad, Iraq, IoT Based Gas Leakage Detection and Alarming System using Blynk platforms, Iraqi Journal for Electrical and Electronic Engineering, Vol. 18| Issue 1| June 2022.

Gas or liquefied petroleum gas (LPG) is a chemical substance resultant from petroleum and could be dangerous in industrial places or those that deal with this substance. Gas leakage causes many health issues. So, to prevent such catastrophes and in order to maintain a clean air environment, the workspace atmosphere should be frequently monitored and controlled.

The proposed monitoring gas leakage detector system is based on Internet of Things (IoT)technology. NodeMCU ESP8266 Wi-Fi is used to be the microcontroller for the whole system. The combustible gas sensor (MQ2) is used in order to detect the presence of methane (CH4) and carbon monoxide gas (CO). MQ2 sensor will detect the concentration of the gas according to the voltage output of the sensor and the ESP8266 will send the data reading from the gas sensor to Blynk IoT platform over an IOS phone; data visualization is done using Thing speak IoT Platform. Besides, a fan will immediately work upon the leakage occurs along with an alarming buzzer.

[7] Prof. Pranay Meshram*1, Nancy Shukla, Stuti Mendhekar ,Renuka Gadge, Shivani Kanaskar, IoT Based LPG Gas Leakage Detector, Research Article, IJESC Volume 7 Issue No.4, 2017.

Gas leakage is a major problem in the industrial sector, in residential locations etc. One of the preventive methods to stop the incident related to the gas leak is to install a gas leak detection kit at vulnerable locations.

The goal of this paper is to propose a system that can detect, alert and automatically control gas leaks. In particular, a gas sensor has been used which has a high sensitivity to gases such as propane and butane together with LPG. There is an alarm that is triggered once the LPG has been detected. The gas leakage system consists of a Wi-Fi module that alerts the user by sending an SMS message.

2.2 Gaps Identified

Only after the recognition of need, then the proposed system is compared and then further analysis is possible. At this stage, we had to perceive the problem and opportunities, the existing system is studied and found out that there were few areas where we can integrate with other technology to make the system better than the existing system. It was analyzed that such proposed system would be possible to develop with given and it might turn out to be the feasible solution. In this project, the biggest challenge was to integrate the existing online voting system with the designed blockchain framework and on further development levels we encountered various unit level problems such as the model for the Iot gas detection and providing solution for the gas leakage.

2.3 Proposed System

Our proposed system detects the smoke, alcohol, and automatically pumps the solution resided in the container and reduce the content of gases in the atmosphere.

These applets consist of a trigger (the “if this” part) and an action (the “then that” part). For example, if the temperature sensor detects a certain temperature (trigger), then the air conditioning turns on (action). IFTTT can be used with the Arduino IoT Cloud to integrate IoT projects with other services and devices, such as Twitter, Instagram, and Philips Hue lights. Arduino Nano This allows for greater automation and customization of IoT projects. Our proposed system detects the smoke, alcohol, and automatically pumps the solution resided in the container and reduce the content of gases in the atmosphere.

CHAPTER 3

REQUIREMENT SPECIFICATION

Framework prerequisites are the setup that a framework should have for an equipment or programming application to run as expected and proficiently. Inability to meet these necessities can bring about establishment issues or execution issues. The previous may keep a gadget or application from getting introduced, while the last option might make an item breakdown or perform underneath assumption or even to hang or crash.

3.1 Hardware Requirements

- NodeMCU
- Arduino nano 3.0
- IR Sensor
- Relay
- MQ 2 Sensor

3.2 Software Requirements

- Arduino IOT Cloud Platform
- IFTTT Platform
- Cadio Application

3.3 Non-Functional Requirements

1. Ease of use: The application created ought to be easy to use. There should not be any disarrays in choosing the choices.
2. Execution: The reaction time should be pretty much as little as could really be expected and there ought not be many postponements.
3. Interoperability: Various methods ought to mix and cooperate to make the framework work appropriately.

Android Studio

Android Studio is the authority Coordinated Improvement Climate (IDE) for Google's Android Working Framework, based on Jet Brain's IntelliJ Thought programming and planned explicitly for Android advancement. It is accessible for download on Windows MacOS and Linux based Working Framework. It is a substitution for the Obscuration Android Improvement Tools (ADT) as the essential IDE for local Android application improvement.

Arduino Cloud Platform

Arduino IoT Cloud is a service that tries to make it seamless to convert your Arduino devices into IoT devices. Any IoT device typically gathers data from sensors, does some processing onboard, and transmits either the raw or the processed data to a server. Arduino IoT Cloud allows you to generate a digital twin of your device (called as a thing), add variables and settings to that digital twin, and then generates the Arduino Sketch automatically which you can upload to the device. Thus, you essentially don’t need to write the Arduino Sketch yourself. What’s more, it also provides dashboards and widgets to monitor your devices, and also tracks events (i.e. special application messages sent by devices, indicating that something has happened) as and when they are received. It is important to note that not all Arduino devices are compatible with Arduino IoT cloud. Only a select few with WiFi or GSM/NB-IoT provisions are compatible. Some third-party boards are also compatible.

IFTTT

IFTTT derives its name from the programming conditional statement “if this, then that.” What the company provides is a software platform that connects apps, devices and services from different developers in order to trigger one or more automations involving those apps, devices and services.The automations are accomplished via applets — which are sort of like macros that connect multiple apps to run automated tasks. You can turn on or off an applet using IFTTT’s website or mobile apps (and/or the mobile apps’ IFTTT widgets). You can also create your own applets or make variations of existing ones via IFTTT’s user-friendly, straightforward interface.

CADIO

CADIO is complete home automation platform allows you to build and control smart home devices, with many new features developed to give you the best smart home experience. By this application if the people gets registered by the master details will get notified by the message where ever the person is present.

Description of Components

1. IR Sensor

IR sensor is an electronic device, that emits the light in order to sense some object of the surroundings. An IR sensor can measure the heat of an object as well as detects the motion. Usually, in the infrared spectrum, all the objects radiate some form of thermal radiation. These types of radiations are invisible to our eyes, but infrared sensor can detect these radiations.The emitter is simply an IR LED (Light Emitting Diode) and the detector is simply an IR photodiode . Photodiode is sensitive to IR light of the same wavelength which is emitted by the IR LED. When IR light falls on the photodiode, the resistances and the output voltages will change in proportion to the magnitude of the IR light received. There are five basic elements used in a typical infrared detection system: an infrared source, a transmission medium, optical component, infrared detectors or receivers and signal processing. Infrared lasers and Infrared LED's of specific wavelength used as infrared sources. In this Project IR Sensor used for fire detection.

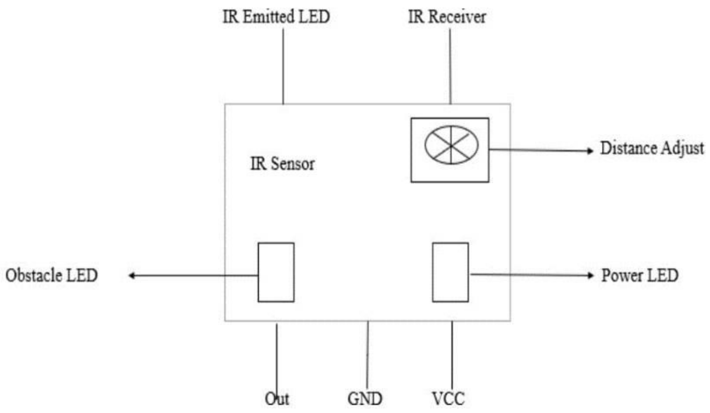


FIGURE 3.1: IR Sensor

2. Relay

A relay is an electronic component that uses an electro-magnet to act as a mechanical switch. The main purpose of a relay is to switch on and off a high powered circuit from a low powered circuit. A relay switch can be seen in many different shapes, sizes, colors, electrical ratings and capabilities. It is embedded in electrical products work in a similar way, they receive an electrical signal and send the signal it Supports 5V.

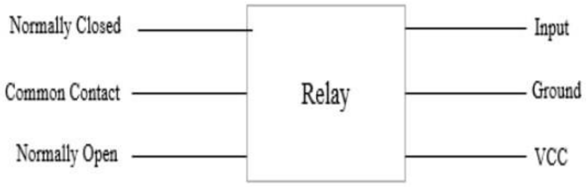


FIGURE 3.2: Relay

3. MQ2 Sensor

The MQ2 family of sensors has a tiny heater inside along with an electro-chemical sensor. They react for range of gases at the room temperature. An analog signal and can be read with an analog input of the Arduino as the output of each sensor. The main objective of the overall system is to detect poisonous gas and radiation leakage. In case any toxin gases or radiation present in industries areas that gases or radiation mainly affected by the industries surrounding areas living peoples. Some gases continuous breathing means kill the human begin and environment mixed this gas or radiation polluted environment condition. If the gases are odorless will be exposed to it for a long time which may cause serious health problems. Gases like CO (carbon monoxide) are odourless which with concentration above surrounding areas living peoples. Some gases continuous breathing means kill the human gases are odorless will be exposed to it for a long time which may cause serious health problems. Gases like CO (carbon monoxide) are odourless which with concentration above system is to detect poisonous gas and radiation leakage. In case any toxin gases or radiation present in industries areas that gases or radiation mainly affected by the industries surrounding areas living peoples. Some gases continuous breathing means kill the human begin and environment mixed this gas or radiation polluted environment condition. If the gases are odorless will be exposed to it for a long time which may cause serious health problems. Gases like CO (carbon monoxide) are odourless which with concentration above 350ppm cause confusion and fainting, above it will surely kill individual. Each gas has its own physical and chemical properties, which make them difficult to analyze without any instrument. Toxic gases present in various levels depending on the concentration and density of it.

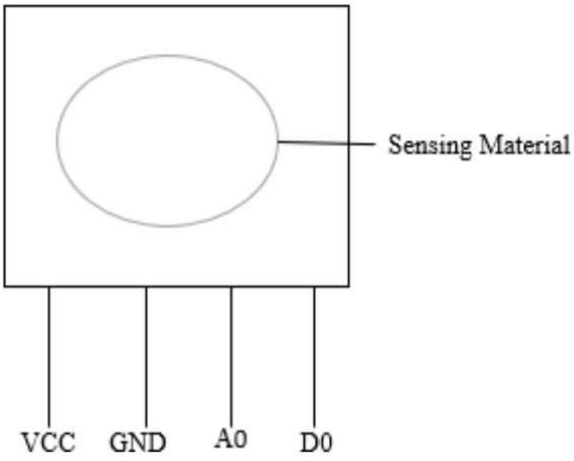


FIGURE 3.3: MQ 2 Sensor

4. Buzzer

Buzzers are electric sounding devices that generate sounds. Typically powered by DC voltage, they can be categorised as Piezo buzzer and magnetic buzzer. They come in different designs and uses as well, and based on that, they can produce different sounds. An audio signaling device like a beeper or buzzer may be electromechanical or piezoelectric or mechanical type. The main function of this is to convert the signal from audio to sound. Generally, it is powered through DC voltage and used in timers, alarm devices, printers, alarms, computers, etc. Based on the various designs, it can generate different sounds like alarm, music, bell & siren.When value goes beyond threshold it make sound and Alter the near by neighbours.

.

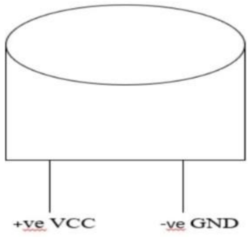


FIGURE 3.4: Buzzer

5. NodeMCU

The NodeMCU (Node Micro Controller Unit) is an open-source software and hardware development environment built around an inexpensive System-on-a-Chip (SoC) called the ESP8266. The ESP8266, designed and manufactured by Espressif Systems, contains the crucial elements of a computer: CPU, RAM, networking (WiFi), and even a modern operating system and SDK. That makes it an excellent choice for Internet of Things (IoT) projects of all kinds. However, as a chip, the ESP8266 is also hard to access and use. You must solder wires, with the appropriate analog voltage, to its pins for the simplest tasks such as powering it on or sending a keystroke to the “computer” on the chip. You also have to program it in low-level machine instructions that can be interpreted by the chip hardware. This level of integration is not a problem using the ESP8 266 as an embedded controller chip in mass-produced electronics. It is a huge burden for hobbyists, hackers, or students who want to experiment with it in their own IoT projects.

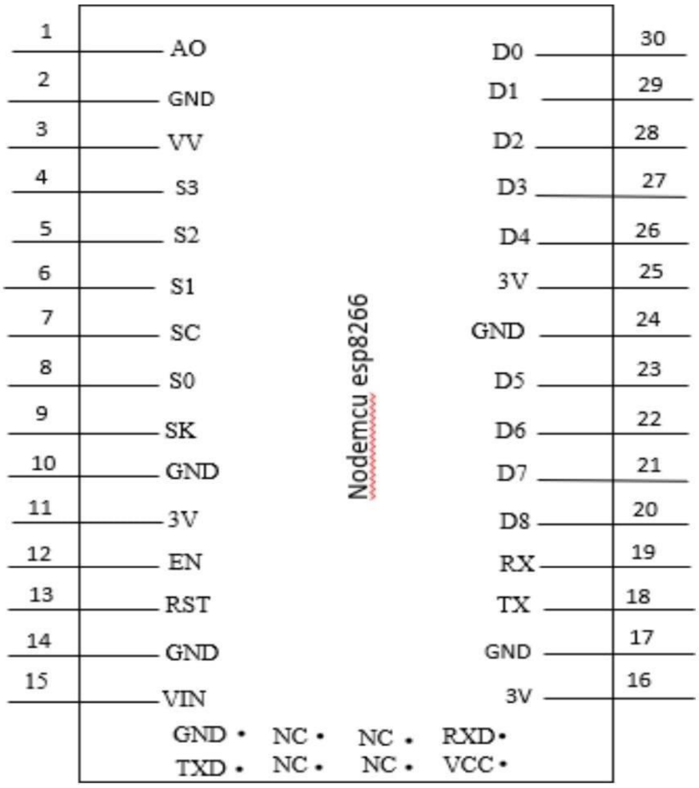


FIGURE 3.5: NodeMCU

3.4 Cost Estimation of the Project

Project cost assessment is the strategy for assessing the expense of the undertaking in computer programming. The assessment utilizes measurable displaying to foster a quote. It utilizes verifiable information of key expense drivers to compute a gauge for various boundaries like expense, exertion and term.

Table 3.1 Model Cost

Components	Price/unit	No. of Units	Total Price
Arduino Nano 3.0	650	1	650
NodeMCU	550	1	550
LED	10	5	50
MQ 2 Sensor	260	1	260
MQ 3 Sensor	260	1	260

MQ 5 Sensor	260	1	260
MQ 7 Sensor	260	1	260
IR Sensor	260	1	260
PCB	250	1	250
Relay	280	1	280
Control Pump	180	1	180
Battery	60	4	240
Buzzer	250	1	250
Wire bunch	150	1	150
Others	300		300
Total			4150

CHAPTER 4

METHODOLOGY

Framework configuration is the method involved with characterizing the components of a framework, for example, the engineering, modules and parts, the various connection point of those parts and the information that goes through that framework. Fulfilling explicit necessities and prerequisites of a business or association through the designing of a cognizant and well-running system is implied.

4.1 Conceptual Diagram

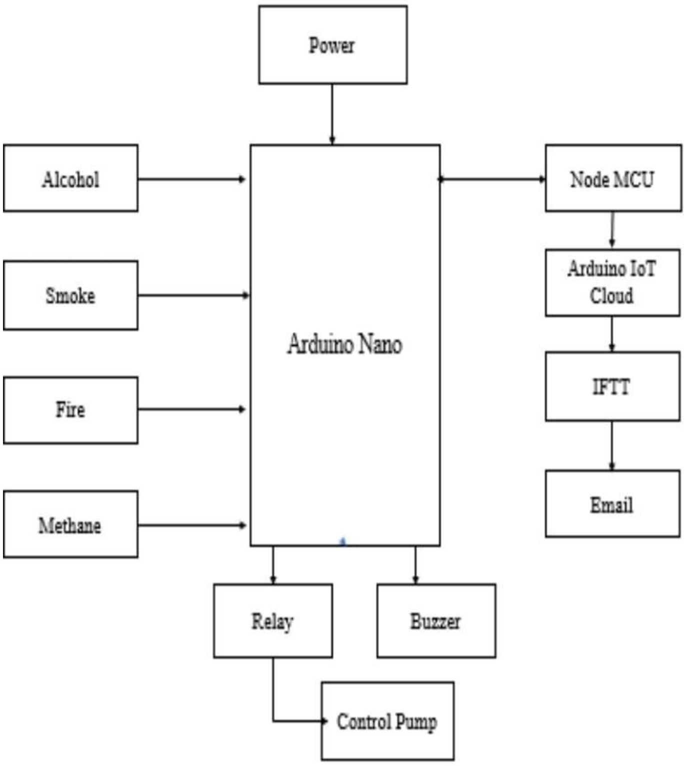


FIGURE 4.1: Conceptual Design

The Arduino Nano is a small, compact board based on the ATmega328P microcontroller, and is suitable for small, low-power projects. It includes features such as a USB interface for programming and power, and can be programmed using the Arduino IDE. The Nano also includes digital and analog pins for interfacing with sensors and actuators, and can be powered by a 5V supply. It is commonly used in IoT projects for its small size, low power consumption, and ease of use.

4.1 Flow Chart

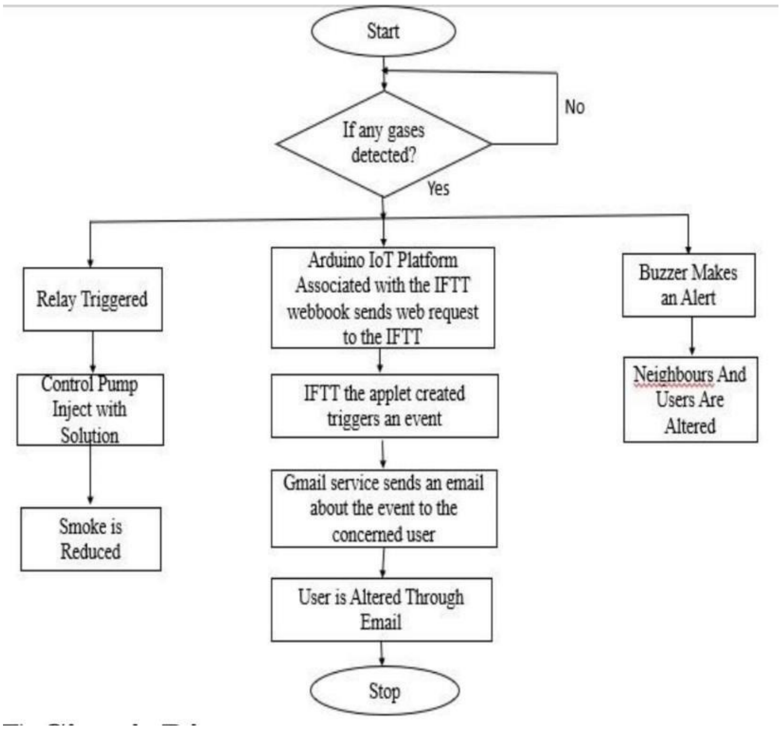

Start
If any gases detected?
No
Yes
Relay Triggered
Control Pump Inject with Solution
Smoke is Reduced
Arduino IoT Platform Associated with the IFTT webbook sends web request to the IFTT
IFTT the applet created triggers an event
Gmail service sends an email about the event to the concerned user
User is Altered Through Email
Stop
Buzzer Makes an Alert
Neighbours And Users Are Altered

FIGURE 4.2: Flow Chart

If the gas is detected by the MQ sensor the data is recorded in the arduino iot cloud platform then if it is above the given threshold then the relay is triggered in turn control pump pumps out the solution from the container and email notification is sent from the the IFTTT through the webhook which is connected to the arduino iot cloud. IFTTT stands for “If This Then That.” It’s a free web service that helps users automate web-based tasks and improve productivity. IFTTT connects various developers’ devices, services and apps to create “applets” that perform automations. The service is incredibly easy to use and includes guides for setting up myriad specific automations. A single web-based tool called IFTTT can serve as an automation hub that helps to integrate internet-connected apps. We’ll explore IFTTT and how business owners can use the service to drive traffic and engagement, track social media mentions, save files to the cloud and much more. The Arduino IoT Cloud is a platform that allows anyone to create IoT projects, with a user friendly interface, and an all in one solution for configuration, writing code, uploading and visualization.

4.3 Activity Diagram

In UML, the activity diagram is used to demonstrate the flow of control within the system rather than the implementation. It models the concurrent and sequential activities. The activity diagram helps in envisioning the workflow from one activity to another. It put emphasis on the condition of flow and the order in which it occurs. The flow can be sequential, branched, or concurrent, and to deal with such kinds of flows, the activity diagram has come up with a fork, join, etc. It is also termed as an object-oriented flowchart. It encompasses activities composed of a set of actions or operations that are applied to model the behavioral diagram.

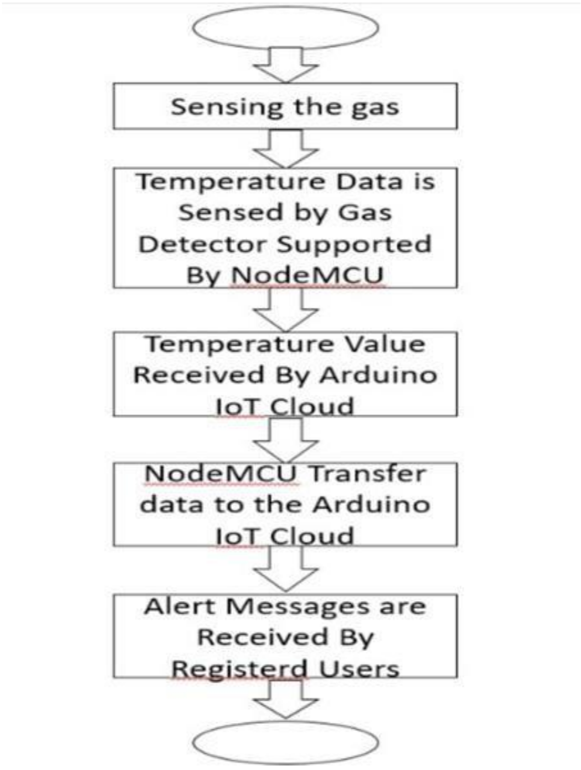

Sensing the gas
Temperature Data is Sensed by Gas Detector Supported By NodeMCU
Temperature Value Received By Arduino IoT Cloud
NodeMCU Transfer data to the Arduino IoT Cloud
Alert Messages are Received By Registerd Users

FIGURE 4.3: Activity Diagram

4.2 Sketch

The first code section is used to include the libraries required by our application.

The thing Properties.h includes all the WiFi connection functionalities and cloud connection management; the only information required from the user are the credentials necessary to establish a connection with a WiFi network (SSID and PASSWORD). Arduino IoT Cloud authenticates the connected device using a key stored inside the board's crypto-chip during the Device Setup (under Devices section - ADD DEVICE), guaranteeing data transmission over a secure channel.

4.3 Features

Below is a list of Arduino IoT Cloud features:

- Data Monitoring - learn how to easily monitor your Arduino's sensor values through a dashboard.
- Variable Synchronisation - variable synchronisation allows you to sync variables across devices, enabling communication between devices with minimal coding.
- Scheduler - schedule jobs to go on/off for a specific amount of time (seconds, minutes, hours).
- Over-The-Air (OTA) Uploads - upload code to devices not connected to your computer.
- Webhooks - integrate your project with another service, such as IFTTT.
- Amazon Alexa Support - make your project voice controlled with the Amazon Alexa integration.
- Dashboard Sharing - share your data with other people around the world

The Arduino IoT Cloud supports a wide range of third party boards based on the ESP32 and ESP8266 microcontrollers with support for Wi-Fi. To set them up, simply choose the third party option in the device setup.

Setting up the Arduino IoT Cloud and accessing the different features available involves a few simple steps. So let's take a look at how to go from start to finish:

1. Creating an Arduino Account

To starting using the Arduino IoT cloud, we first need to log in or sign up to Arduino.

2. Go to the Arduino IoT Cloud

After we have signed up, you can access the Arduino IoT Cloud from any page on arduino.cc by clicking on the four dots menu in the top right corner. You can also go directly to the Arduino IoT Cloud.

3. Creating a Thing

The journey always begin by creating a new Thing. In the Thing overview, we can choose what device to use, what Wi-Fi network we want to connect to, and create variables that we can monitor and control. This is the main configuration space, where all changes we make are automatically generated into a special sketch file.

4. Configuring a Device

Devices can easily be added and linked to a Thing. The Arduino IoT Cloud requires your computer to have the Arduino Create Agent installed. The configuration process is quick and easy, and can be done by clicking on the "Select device" button in the Thing overview. Here, we can choose from any board that has been configured, or select the "Configure new device" option. We can also get a complete overview of our devices by clicking the “Devices" tab at the top of the Arduino IoT Cloud interface. Here we can manage and add new devices.

5. Creating Variables

The variables we create are automatically generated into a sketch file. There are several data types we can choose from, such as int, float, boolean, long, char. There’s also special variables, such as Temperature, Velocity, Luminance that can be used. When clicking on the “Add variable” button, we can choose name, data type, update setting and interaction mode.

6. Connecting to a Network

To connect to a Wi-Fi network, simply click the “Configure” button in the network section. Enter the credentials and click “Save”. This information is also generated into your sketch file.

7. Editing the Sketch

Now that we have configured variables, devices and network settings, we can get to programming our devices.An automatically generated sketch file can be found in the “Sketch” tab. It has the same structure as a typical .ino file, but with some additional code to make the connection to your network and to the cloud.A sketch that, for example, reads an analog sensor, and use the cloud variable to store it. When the sketch has been uploaded, it will work as a regular sketch, but it will also update the cloud variables that we use. Additionally, each time we create a variable that has the Read & Write permission enabled, a function is also generated, at the bottom of your sketch file. Each time this variable changes, it will execute the code within this function! This means that we can leave most of the code out of the loop() and only run code when needed. To upload the program to our board, simply click the "Upload" button.

8. Creating a Dashboard

Now that we have configured the device & network, created variables, completed the sketch and successfully uploaded the code, we can move on to the fun part: creating dashboards. Dashboards are visual user interface for interacting with your boards over the cloud, and we can setup many different setups depending on what your IoT project needs. We can access our dashboards by clicking on the “Dashboards” tab at the top of the Arduino IoT Cloud interface, where we can create new dashboards, and see a list of dashboards created for other Things. If we click on “Create new dashboard”, we enter a dashboard editor. Here, we can create something called widgets. Widgets are the visual representation of our variables we create, and there are many different to choose from. Below is an example using several types of widgets.

CHAPTER 5

IMPLEMENTATION

Execution is the period of venture where the hypothetical layout is changed into a functioning structure. Project execution is the stage where dreams and plans become reality. In the event that execution isn't deliberately organized and controlled, it can achieve disarray and bewilderment.

5.1 Circuit Diagram

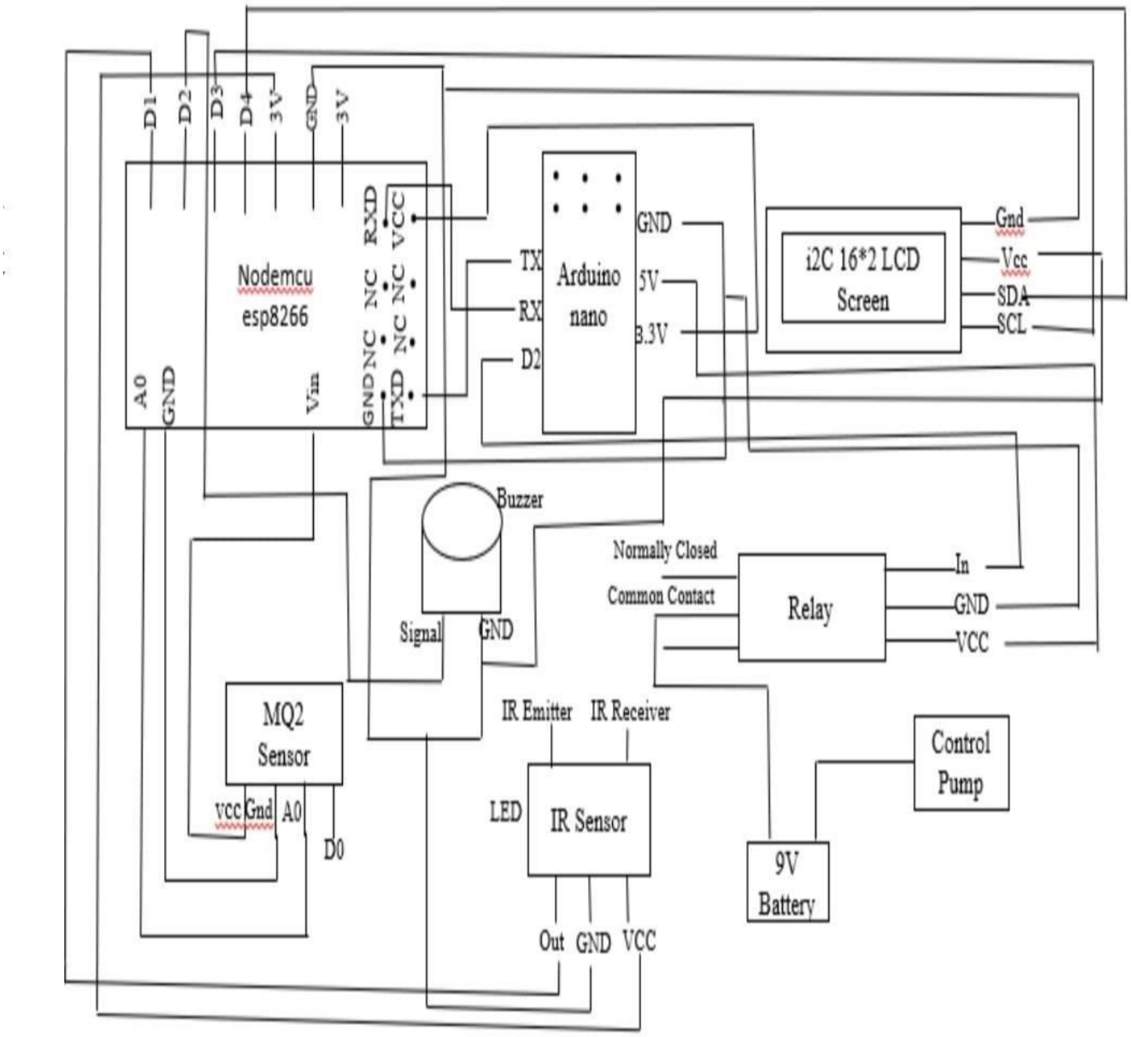


FIGURE 5.1: Circuit Diagram

NodeMCU is a low-cost open source IoT platform. It initially included firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which was based on the ESP-12 module. Node MCU is connected to the arduino iot cloud in turn which is connected to IFTTT platform when MQ2 sensor defects the gas the value is recorded in the iot cloud it is shown in the widgets form then when it reaches some value automatically the email is sent to the registered user. Automatically the solution is pumped by the container to reduce the gas concentration.

5.2 IoT Security

IoT Security is the act of securing Internet devices and the networks they're connected to from threats and breaches by protecting, identifying, and monitoring risks all while helping fix vulnerabilities from a range of devices that can pose security risks to your business.

- Learn: With complete network visibility, security solutions can authenticate and classify IoT devices to build a risk profile and assign them to IoT device groups.
- Segment: Once the enterprise understands its IoT attack surface, IoT devices can be segmented into policy-driven groups based on their risk profiles.
- Protect: The policy-driven IoT groups and internal network segmentation enable monitoring, inspection, and policy enforcement based on the activity at various points within the infrastructure.

5.3 IoT Cloud Security

Cloud computing offers several advantages to businesses, including greater technological flexibility, reduced operational costs and easy scalability. When cloud computing is implemented in an IoT network, the cloud platform and connected applications become highly vulnerable to cyber threats. Here are some ways in which you can ensure IoT cloud security using holistic security principles:

1) Encryption of data at rest

Businesses embracing IoT for the first time lay a lot of focus on the security of the cloud infrastructure. So, it is crucial to deploy encryption technologies to secure the cloud. Encryption is a process in which legible data (plaintext) is converted into an output (ciphertext) that does not reveal any information about the input plaintext. An encryption algorithm is employed for this conversion. Encryption ensures that even if an attacker obtains access to storage devices with sensitive data, they would not be able to decipher it Encryption of data at rest implies that an encryption algorithm is used to safeguard data that is stored on any kind of disk, including backup devices and solid-state drives. Several layers of encryption can be used to protect data at rest. An example of this is the encryption of sensitive information prior to storage along with the encryption of the storage drive itself.

2) Encryption of data in transit

Data in transit is considered to be at higher risk for security breaches. So, whether the data is being communicated over the internet or between data centers, it is crucial to ensure that an end-to-end security strategy is in place. In order to protect data in transit, encryption is enabled prior to moving the data.

Encrypted connections such as HTTPS, FTPS, SSL, TLS, etc. can also be used.

3) Device identity

Each device in an IoT implementation should have a unique device identity. When a device comes online, this identity is used to authenticate it and authorize secure communication with other components of the IoT ecosystem. Device authentication using OAuth 2.0 is a powerful open standard that can be used by API developers to protect an IoT ecosystem. It is a token-based authentication and authorization solution that also offers a framework for the decisions associated with authentication.

4) User role and policy

As part of access management, a privileged user management system can be deployed to ensure that stringent authentication processes are followed for user access to IoT data. It is also possible to create policies that can be attached to identities/resources to define their permissions. The administrator defines the policies and specifies the access level of resources.

5) Certificate based authentication

A certificate is essentially a signed digital document that includes attributes identifying its issuer and owner (also referred to as subject). It contains two important fields – a public key that belongs to the owner/subject and a digital signature from the issuer. The issuer is usually a Certificate Authority (CA) and X.509 is a popular digital certificate standard.The public key can establish a secure communication channel with the subject. The signature is proof that the subject's identity is verified by the issuer. The subject also possesses a private key that matches the public key, but this is not a part of the certificate.

The private key is used for proving the identity of the subject once a communication session is established.

5.4 Alarm

The caution or signal utilizing this framework primary reason is demonstrated to ready working individuals and staying individuals moved wellbeing place spare the general population life and condition.Sensors continuously monitors the gases in the surrounding and posts in to the server for storing and usage of data futurly. While continuously monitoring if any gas level exceed range to that of normal range in air the alert will be enhanced and a email notification will be posted to safety control board of organization and even to the workers mobile station only if required.

5.5 MQ2 Sensor

The MQ2 family of sensors has a tiny heater inside along with an electro-chemical sensor. They react for range of gases at the room temperature. An analog signal and can be read with an analog input of the Arduino as the output of each sensor. The main objective of the overall system is to detect poisonous gas and radiation leakage. In case any toxin gases or radiation present in industries areas that gases or radiation mainly affected by the industries surrounding areas living peoples. Some gases continuous breathing means kill the human begin and environment mixed this gas or radiation polluted environment condition. If the gases are odorless will be exposed to it for a long time which may cause serious health problems. Gases like CO (carbon monoxide) are odourless which with concentration above 350ppm cause confusion and fainting, above it will surely kill individual. Each gas has its own physical and chemical properties, which make them difficult to analyze without any instrument. Toxic gases present in various levels depending on the concentration and density of it.

Every year, numerous people lose their lives to gas-related incidents. It is important to take necessary precautions to avoid such accidents. Alcohol can be dangerous and should be handled with care and proper precautions. It is a flammable liquid that can easily catch fire if it comes into contact with an ignition source. Thus, it is important to store alcohol in a cool, dry and well-ventilated area, away from heat sources and open flames. In addition to its uses a source of energy, methane also has an important role in the natural gas industry. It is the primary component of natural gas, which is used for heating and cooking in homes and businesses, as well as in transportation and industrial operations. Methane is also used as feedstock in the production of chemicals such as methanol, ammonia, and hydrogen. However, the release of methane into the atmosphere is a major contributor to climate change. Methane isa potent greenhouse gas, and its warming effect is estimated to be up to 84 times stronger than carbon dioxide over a 20-year time frame. Methane emissions come from a variety of sources, including the oil and gas industry, agriculture, and waste management.

Arduino Nano is one type of microcontroller board, and it is designed by Arduino.cc. It can be built with a microcontroller like Atmega328. This microcontroller is also used in Arduino UNO. It is a small size board and also flexible with a wide variety of applications. Other Arduino boards mainly include Arduino Mega, Arduino Pro Mini, Arduino UNO, Arduino YUN, Arduino Lilypad, Arduino Leonardo, and Arduino Due. And other development boards are AVR Development Board, PIC Development Board, Raspberry Pi, Intel Edison, MSP430 Launchpad, and ESP32 board.

5.6 Arduino-nano

1) Arduino Nano Communication

The communication of an Arduino Nano board can be done using different sources like using an additional Arduino board, a computer, otherwise using microcontrollers. The microcontroller using in Nano board (ATmega328) offers serial communication (UART TTL). This can be accessible at digital pins like TX, and RX. The Arduino software comprises of a serial monitor to allow easy textual information to transmit and receive from the board. The TX & RX LEDs on the Nano board will blink whenever information is being sent out through the FTDI & USB link in the direction of the computer. The library-like Software Serial allows serial communication on any of the digital pins on the board. The microcontroller also supports SPI & I2C (TWI) communication.

2) Arduino Nano Programming

The programming of an Arduino nano can be done using the Arduino software. Click the Tools option and select the nano board. Microcontroller ATmega328 over the Nano board comes with preprogrammed with a boot loader. This boot loader lets to upload new code without using an exterior hardware programmer. The communication of this can be done with the STK500 protocol. Here the boot loader can also be avoided & the microcontroller program can be done using the header of in-circuit serial programming or ICSP with an Arduino ISP.

5.7 Source Code:

```
package com.example.registeractivity;
import android.content.ContentValues;
import  android.content.Context;
import android.database.Cursor;
            import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
public class DBHelper extends SQLiteOpenHelper {
public static final String DBNAME = "login.db";
            public DBHelper(Context context) {
super(context, "Login.db", null,1);
            }
@Override
public void onCreate(SQLiteDatabase MyDB) {
MyDB.execSQL("create Table users(username TEXT primary key, password TEXT, email TEXT, ssid
TEXT, wifi TEXT)");
}
@Override
 public void onUpgrade(SQLiteDatabase MyDB, int i, int i1) {
 MyDB.execSQL("drop Table if exists users");
 }
public Boolean insertData(String username, String password,String email, String ssid,String wifi) {
SQLiteDatabase MyDB = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
contentValues.put("username", username);
        contentValues.put("password", password);
contentValues.put("email",email);
contentValues.put("ssid",ssid);
contentValues.put("wifi",wifi);
long result = MyDB.insert("users", null, contentValues);
if (result == -1) return false;
else
return true;



}
        public Boolean checkusername(String username) {
        SQLiteDatabase MyDB = this.getWritableDatabase();
Cursor cursor = MyDB.rawQuery("Select * from users where username = ?",
new String[]{username});
if (cursor.getCount() > 0)
return true;
else
return false;
}
public Boolean ccheckusernamepassword(String username, String password) {
        SQLiteDatabase MyDB = this.getWritableDatabase();
        Cursor cursor = MyDB.rawQuery("Select * from users where username = ? and         password
= ?", new String[] {username,password});
if (cursor.getCount() > 0)
        return true;
        else
return false;
}
}
```

```
              package com.example.registeractivity;
              import androidx.appcompat.app.AppCompatActivity;
import android.annotation.SuppressLint;
              import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class HomeActivity extends AppCompatActivity {
              EditText username1,password1,password2;
              Button btnlogin1;

private String name;
private int password;
              DBHelper DB;
             @SuppressLint("MissingInflatedId")
             @Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_home);

username1 = (EditText) findViewById(R.id.email);
               password1 = (EditText)  findViewById(R.id.ssid);
 password2 = (EditText)  findViewById(R.id.password2);

 btnlogin1 = (Button) findViewById(R.id.button1);
 DB = new DBHelper(this);

               btnlogin1.setOnClickListener(new View.OnClickListener() {
 @Override
 public void onClick(View v) {
```

```
 String user1 = username1.getText().toString();
 String pass1 = password1.getText().toString();
 String pass2 = password2.getText().toString();

 String username = getIntent().getStringExtra("keyuser");
 String password = getIntent().getStringExtra("keypassword");

 if(user1.equals("")||pass1.equals("")||pass2.equals(""))
Toast.makeText(HomeActivity.this, "Please enter all the fields",
Toast.LENGTH_SHORT).show(); Boolean insert = DB.insertData(username,
String.valueOf(password),user1,pass1, pass2);
   if(insert==true){
Toast.makeText(HomeActivity.this, "Submitted Successfully ",     Toast.LENGTH_SHORT).show();
              }else{
Toast.makeText(HomeActivity.this, "Invalid Credentials",        Toast.LENGTH_SHORT).show();
              }
          }
       });
    }
}
package com.example.registeractivity;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class LoginActivity extends AppCompatActivity {
    EditText username, password;
    Button btnlogin;
    DBHelper DB;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        username = (EditText) findViewById(R.id.username1);
        password = (EditText) findViewById(R.id.password1);
        btnlogin = (Button) findViewById(R.id.btnsignin1
 DB = new DBHelper(this);
btnlogin.setOnClickListener(new View.OnClickListener() {
               @Override
                public void onClick(View view) {

                String user = username.getText().toString();
                String pass = password.getText().toString();

                if(user.equals("")||pass.equals(""))
                 Toast.makeText(LoginActivity.this, "Please enter all the fields",
Toast.LENGTH_SHORT).show();
                else{
                Boolean checkuserpass = DB.ccheckusernamepassword(user, pass);
                if(checkuserpass==true){
                Toast.makeText(LoginActivity.this, "Sign in successfull", Toast.LENGTH_SHORT).show();
                Intent intent = new Intent(getApplicationContext(), HomeActivity.class);
                startActivity(intent);
                }else{
               Toast.makeText(LoginActivity.this, "Invalid Credentials", Toast.LENGTH_SHORT).show();
                   }
                }
            }
        });
    }}
```

CHAPTER 6

TESTING AND RESULTS

6.1 Testing

In the world today, technology is used to create several machines and make life easier. The software could have multiple bugs and might not be working as it is intended to. Hence testing is required to control and make sure that the software is error-free. It identifies all the defects that the software and makes sure the software is meeting the required specifications. Testing is also very cost-effective. It prevents failure from occurring in the future. It will also be cheaper to fix the bugs when it is in an earlier stage.It also improves the quality of the products after it is tested. This project uses *Mocha* as the testing framework to unit test and integration test all of our test cases for the application. Following strategies are used:

(i) Unit Testing: This is the first and the most important level of testing. Its need begins from the moment a programmer develops a unit of code. Every unit is tested for various scenarios. Detecting and fixing bugs during early stages of the Software Lifecycle helps reduce costly fixes later on. It is much more economical to find and eliminate the bugs during early stages of application building process. Hence, Unit Testing is the most important of all the testing levels. As the software project progresses ahead it becomes more and more costly to find and fix the bugs. Steps for Unit Testing are:

Step 1: Creation of a Test Plan

Step 2: Creation of Test Cases and the Test Data

Step 3: Creation of scripts to run the test cases wherever applicable

Step 4: Execution of the test cases, once the code is ready

Step 5: Fixing of the bugs if present and re testing of the code

Step 6: Repetition of the test cycle until the Unit is free from all types of bugs.

(ii) Integration Testing: Integration strategy stands for how individual modules will be combined during Integration testing. The individual modules can be combined in one go,

or they can be joined one by one. A decision on how to put the pieces together is called the Integration Strategy. We have used bottom-up integration approach to integrate test our application. In Bottom-Up

Integration, we move from the bottom to top i.e. the components below are first written and these are integrated first. The integration happens from bottom to top. If the calling component is yet to be developed, it is replaced by aspecially written component called a Driver.

6.2 Cost Estimation and Project Planning

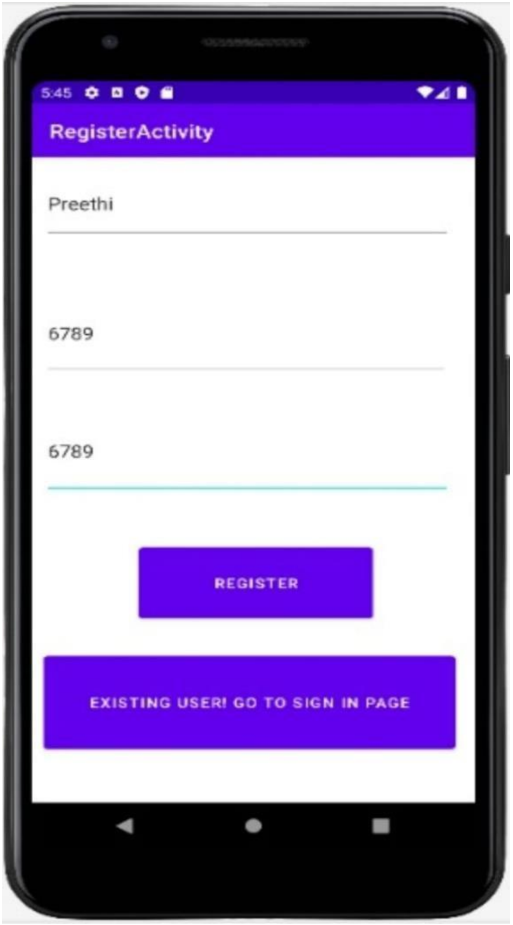
The people who do the cost estimates could be either directly or indirectly responsible for the implementation for a project, such as a developer or manager, respectively. Someone who has knowledge of the organization and previous projectscould use an analogy-based approach to compare the current project with previous projects, which is a common method of estimation for small organizations and small projects. The historical data is often limited to the memory of the estimator. In this case, the estimator would need to be experienced and would likely have been with the companyfor a while. Some people believe it is better if the estimates are done by outsiders so that there is less chance of bias. It is true that people outside an organization will likely haveto deal with fewer company politics than people within the organization. For example, thedeveloper for a company may want to please the manager and so give an estimate that is overly-optimistic. The disadvantage of having an outside estimate is that the person would have less knowledge of the development environment, especially if the person is from outside the company. An empirical method of estimation would then be required, such as the Constructive Cost Model (COCOMO). Empirical methods of estimation can be used by all types of estimators. There may be some resistance to using an empirical method of estimation because there may be some question on whether a model could outperform an expert. People who are accurate estimators are rare in our experience, and so it is best to get the opinion of several people or tools.

In the actual cost estimation process, there are other inputs and constraints that needed to be considered besides the cost drivers. One of the primary constraints of the software cost estimate is the financial constraint, which are the amount of the money that can be budgeted or allocated to the project. There are other constraints such as manpower constraints, and date constraints. Other input such as architecture, which defines the components that made up the system and the interrelationships between these components. Some company will have certain software process or an existing architecturein place; hence for these companies the software cost estimation must base their estimateson these criteria. There are only very few cases where the software requirements stay fixed. Hence, how do we deal with software requirement changes, ambiguities or inconsistencies. During the estimation process, an experienced estimator will detect the ambiguities and inconsistency in the requirements. As part of the estimation process, the estimator will try to solve all these ambiguities by modifying the requirements. If the ambiguities or inconsistent requirements stay unsolved, which will correspondingly affect the estimation accuracy.

Cost estimation is an important tool that can affect the planning and budgeting ofa project. Because there are a finite number of resources for a project, all of the featuresof a requirements document can often not all be included in the final product. A cost estimate done at the beginning of a project will help determine which features can be included within the resource constraints of the project (e.g., time). Requirements can be prioritized to ensure that the most important features are included in the product. The risk of a project is reduced when the most important features are included at the beginning because the complexity of a project increases with its size, which means there is more opportunity for mistakes as development progresses. Thus, cost estimation can have a big impact on the life cycle and schedule for a project.

CHAPTER 7

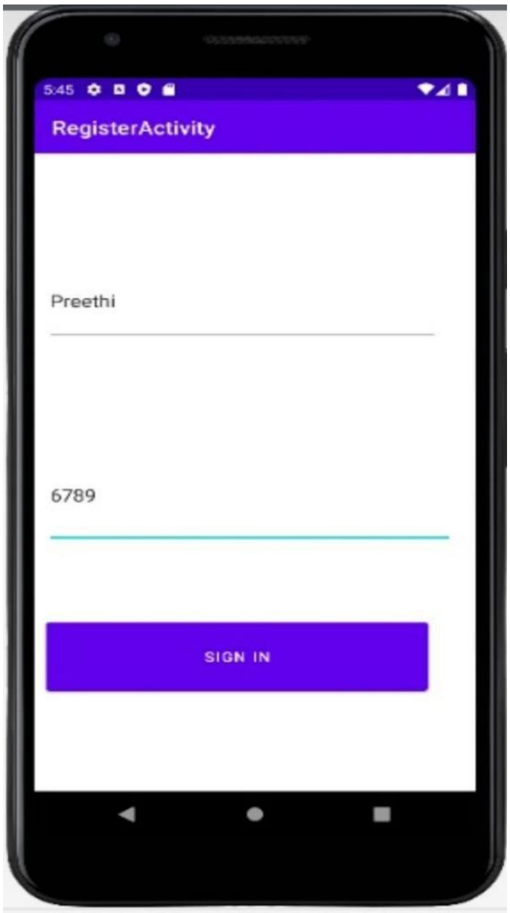
SNAPSHOTS



Preethi

6789

6789

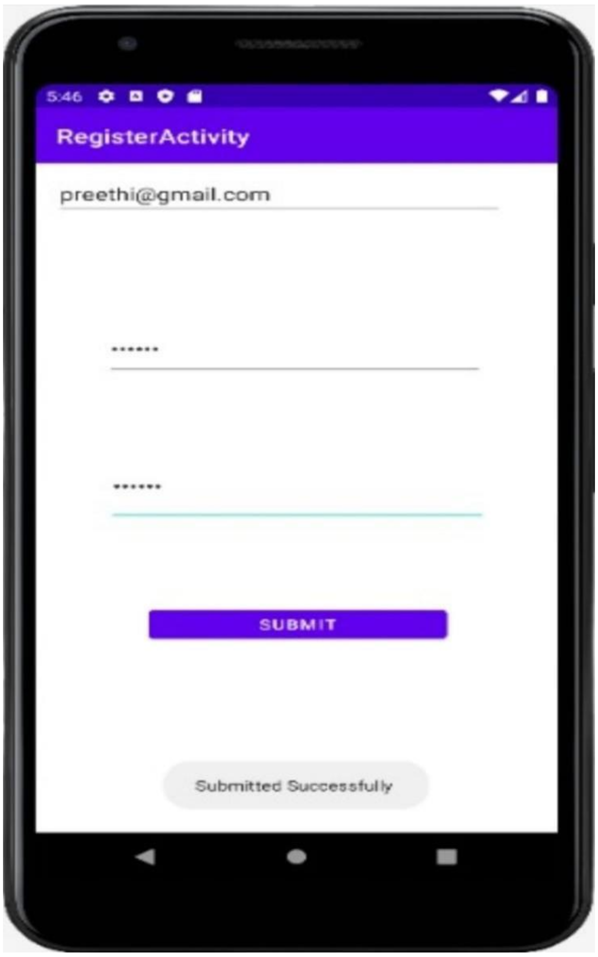


Preethi

6789

Figure 7.1: User Registration

Figure 7.2: User Sign-in



RegisterActivity

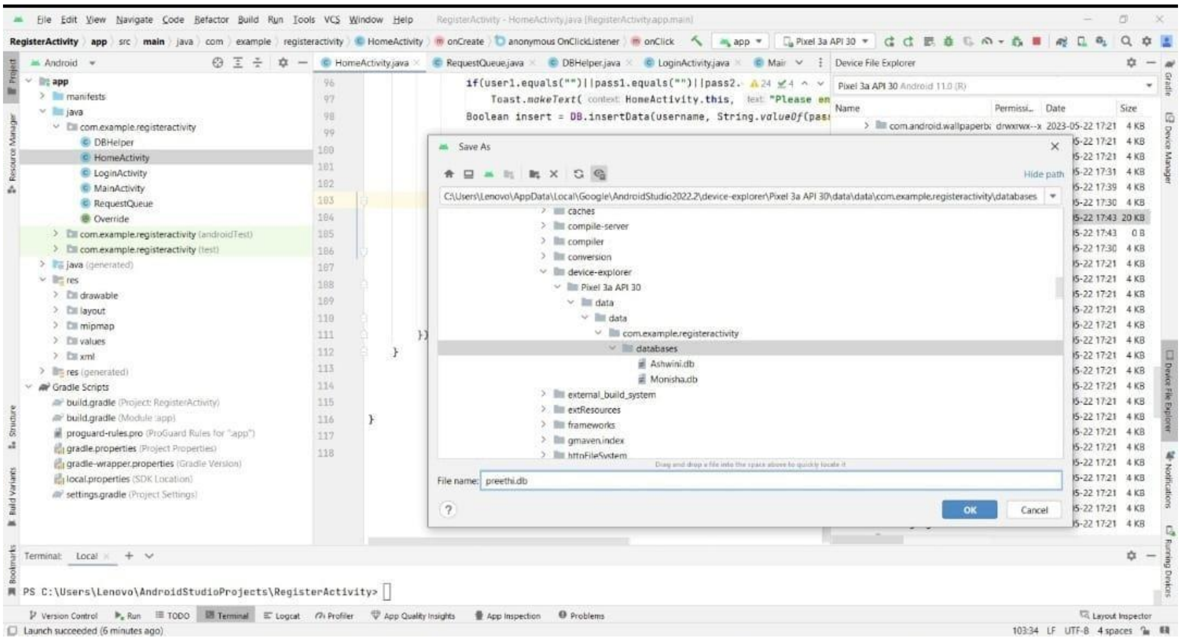
preethi@gmail.com

******

******

Submitted Successfully

Figure 7.3: Collecting User Device SSID and Password



```
if(user1.equals("")||pass1.equals("")||pass2.
    Toast.makeText( context: HomeActivity.this,  text: "Please en
Boolean insert = DB.insertData(username, String.valueOf(pas
```

Save As

C:\Users\Lenovo\AppData\Local\Google\AndroidStudio2022.2\device-explorer\Pixel 3a API 30\data\data\com.example.registeractivity\databases

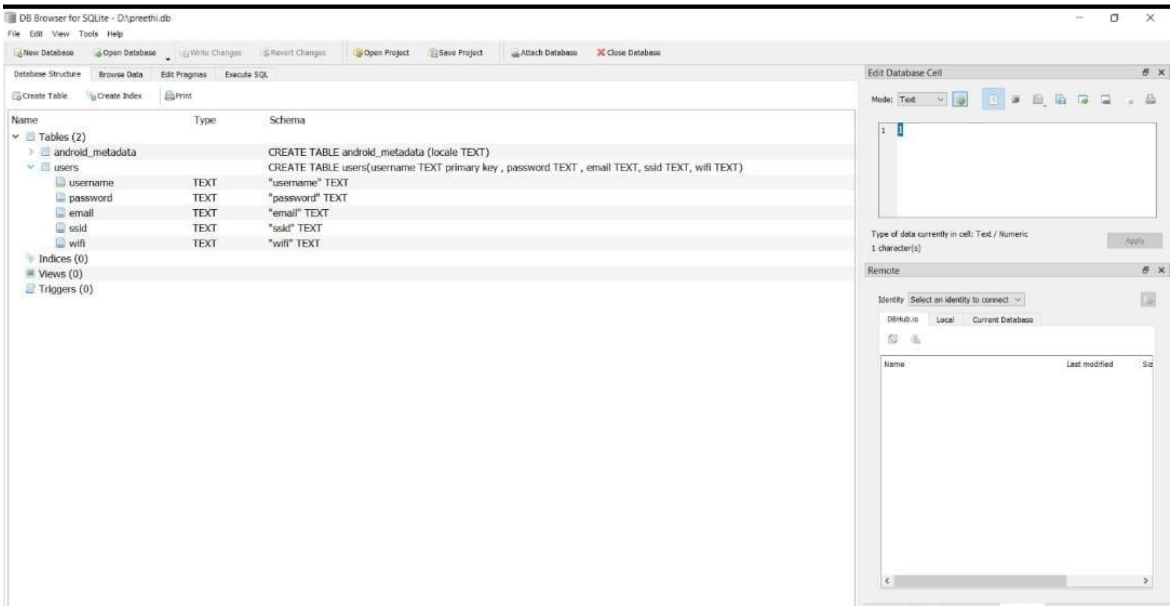
File name: preethi.db

```
PS C:\Users\Lenovo\AndroidStudioProjects\RegisterActivity>
```

Launch succeeded (6 minutes ago)

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
RegisterActivity - HomeActivity.java [RegisterActivity.app.main]
RegisterActivity app src main java com example registeractivity HomeActivity onCreate anonymous OnClickListener onClick
app
Pixel 3a API 30
HomeActivity.java
RequestQueue.java
DBHelper.java
LoginActivity.java
Device File Explorer
Pixel 3a API 30 Android 11.0 (R)
Name
Permissi...
Date
Size
com.android.wallpaperb
Android
app
manifests
java
com.example.registeractivity
DBHelper
HomeActivity
LoginActivity
MainActivity
RequestQueue
Override
com.example.registeractivity (androidTest)
com.example.registeractivity (test)
java (generated)
res
drawable
layout
mipmap
values
xml
res (generated)
Gradle Scripts
build.gradle (Project: RegisterActivity)
build.gradle (Module :app)
proguard-rules.pro (ProGuard Rules for "app")
gradle.properties (Project Properties)
gradle-wrapper.properties (Gradle Version)
local.properties (SDK Location)
settings.gradle (Project Settings)
Hide path
caches
compile-server
compiler
conversion
device-explorer
Pixel 3a API 30
data
data
com.example.registeractivity
databases
Ashwini.db
Monisha.db
external_build_system
extResources
frameworks
gmaven.index
httpFileSystem
Drag and drop a file into the space above to quickly locate it
OK
Cancel
Terminal: Local
Version Control
Run
TODO
Terminal
Logcat
Profiler
App Quality Insights
App Inspection
Problems
Layout Inspector
103:34 LF UTF-8 4 spaces

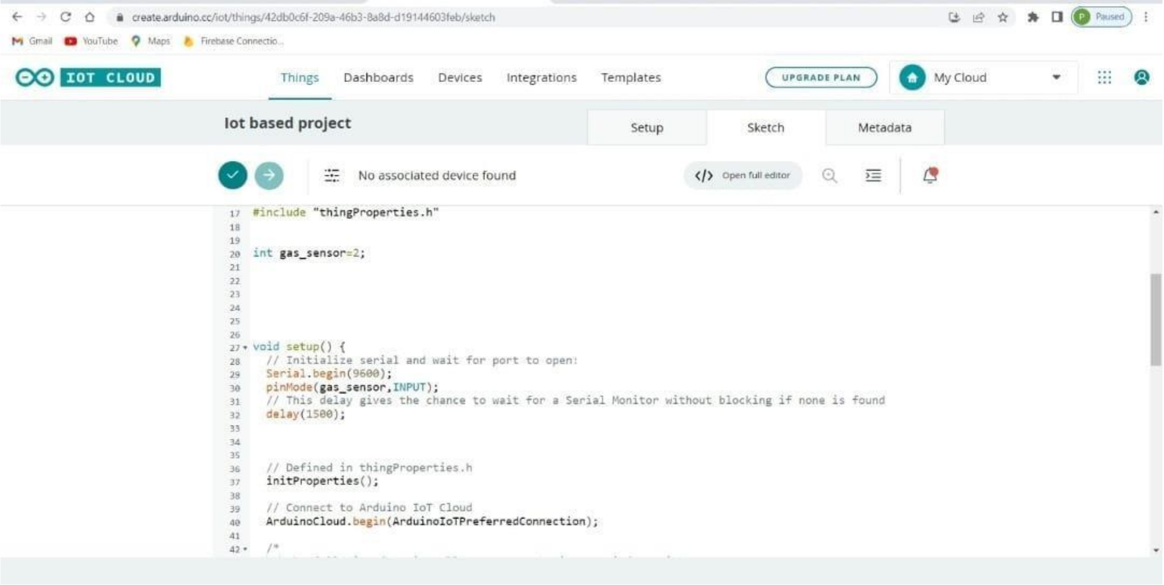
Figure 7.4: Saving the file as .db



Name	Type	Schema
Tables (2)		
android_metadata		CREATE TABLE android_metadata (locale TEXT)
users		CREATE TABLE users(username TEXT primary key , password TEXT , email TEXT, ssid TEXT, wifi TEXT)
username	TEXT	"username" TEXT
password	TEXT	"password" TEXT
email	TEXT	"email" TEXT
ssid	TEXT	"ssid" TEXT
wifi	TEXT	"wifi" TEXT
Indices (0)		
Views (0)		
Triggers (0)		

Type of data currently in cell: Text / Numeric
1 character(s)

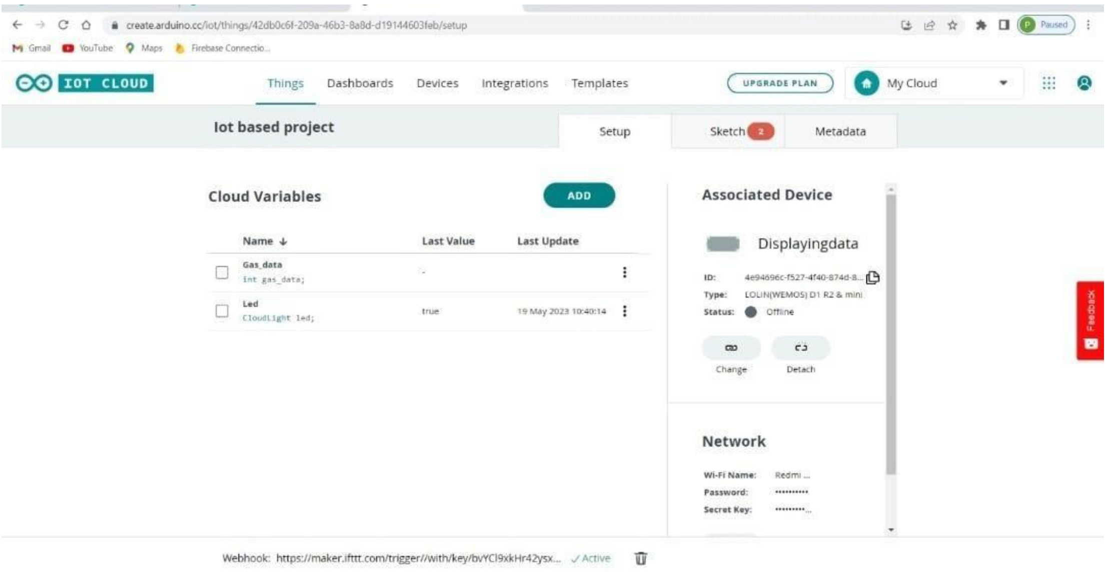
Figure 7.5: VIEW of database tables



No associated device found

```
#include "thingProperties.h"


int gas_sensor=2;






void setup() {
  // Initialize serial and wait for port to open:
  Serial.begin(9600);
  pinMode(gas_sensor,INPUT);
  // This delay gives the chance to wait for a Serial Monitor without blocking if none is found
  delay(1500);



  // Defined in thingProperties.h
  initProperties();

  // Connect to Arduino IoT Cloud
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);

  /*
```

Figure 7.6: Editing the code in arduino iot cloud



create.arduino.cc/iot/things/42db0c6f-209a-46b3-8a8d-d19144603feb/setup

Iot based project

Cloud Variables

	Name	Last Value	Last Update
□	Gas_data `int gas_data;`	-	
□	Led `CloudLight led;`	true	19 May 2023 10:40:14

Associated Device

Displayingdata

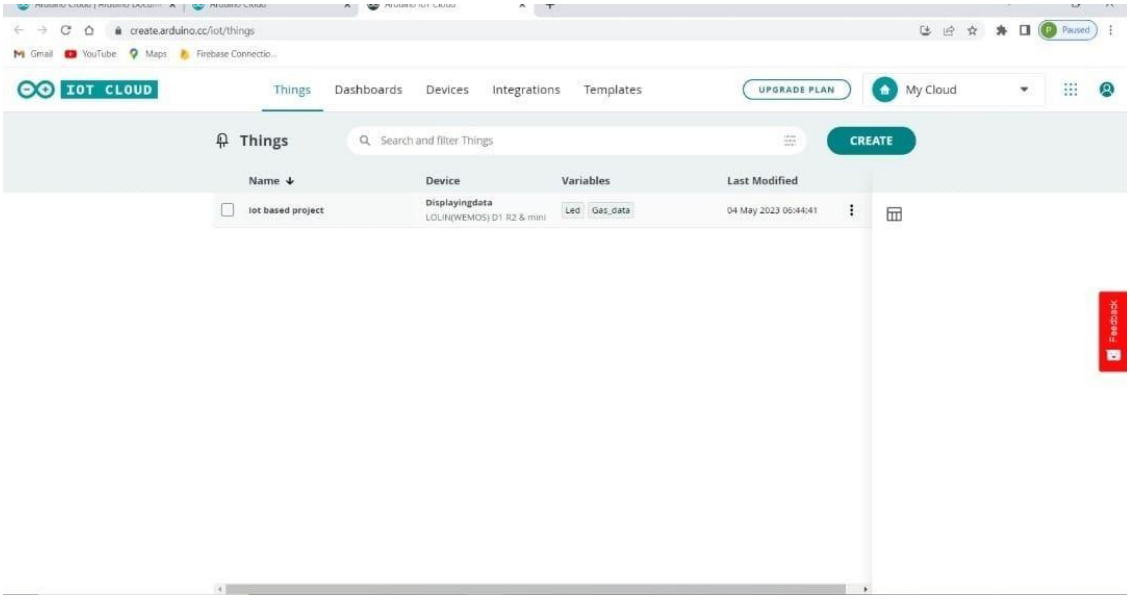
ID: 4e94696c-f527-4f40-874d-8...
Type: LOLIN(WEMOS) D1 R2 & mini
Status: ● Offline

Network

Wi-Fi Name: Redmi ...
Password: ••••••••••
Secret Key: •••••••••...

Webhook: https://maker.ifttt.com/trigger//with/key/bvYCl9xkHr42ysx... ✓ Active

Figure 7.7: Displaying created variable

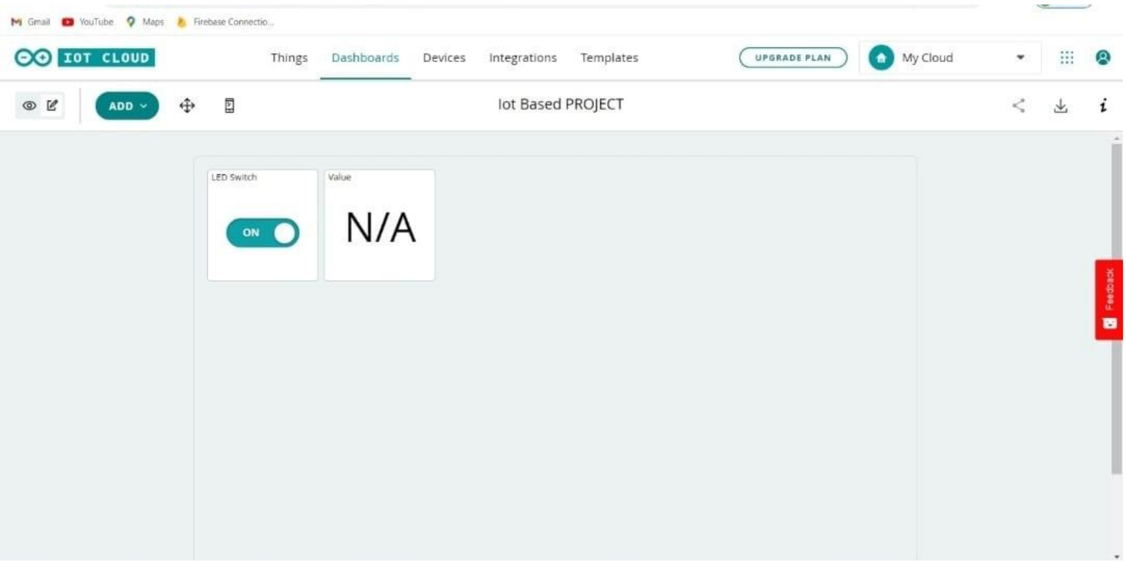


create.arduino.cc/iot/things

Things

	Name	Device	Variables	Last Modified
□	Iot based project	Displayingdata LOLIN(WEMOS) D1 R2 & mini	Led Gas_data	04 May 2023 06:44:41

Figure 7.8:Displaying the device created

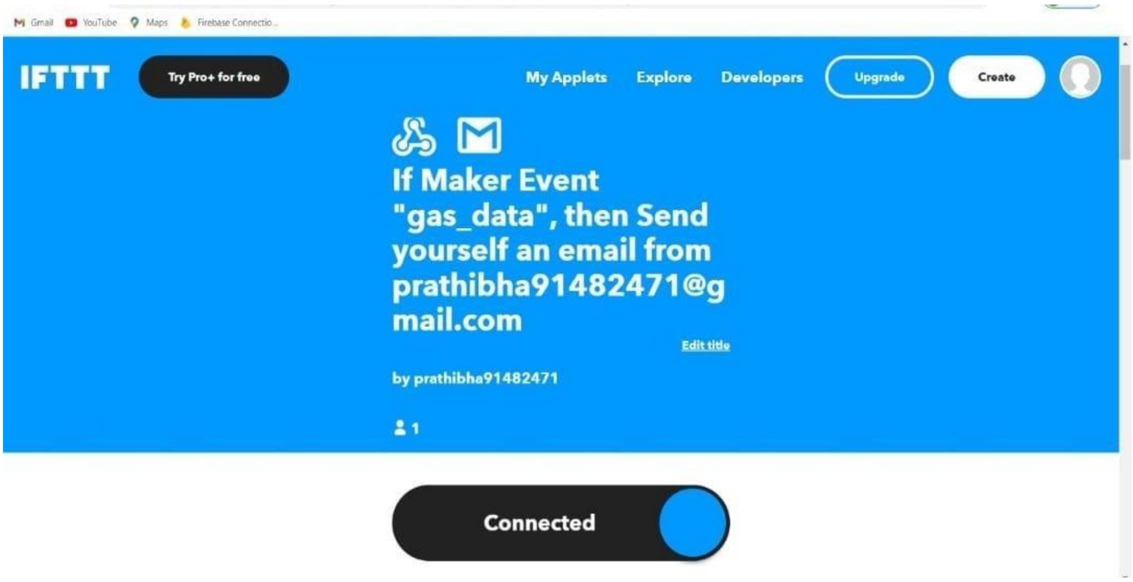


Iot Based PROJECT

LED Switch: ON

Value: N/A

Figure 7.9: Widgets display the values



If Maker Event "gas_data", then Send yourself an email from prathibha91482471@gmail.com

by prathibha91482471

Connected

Figure 7.10: IFTTT

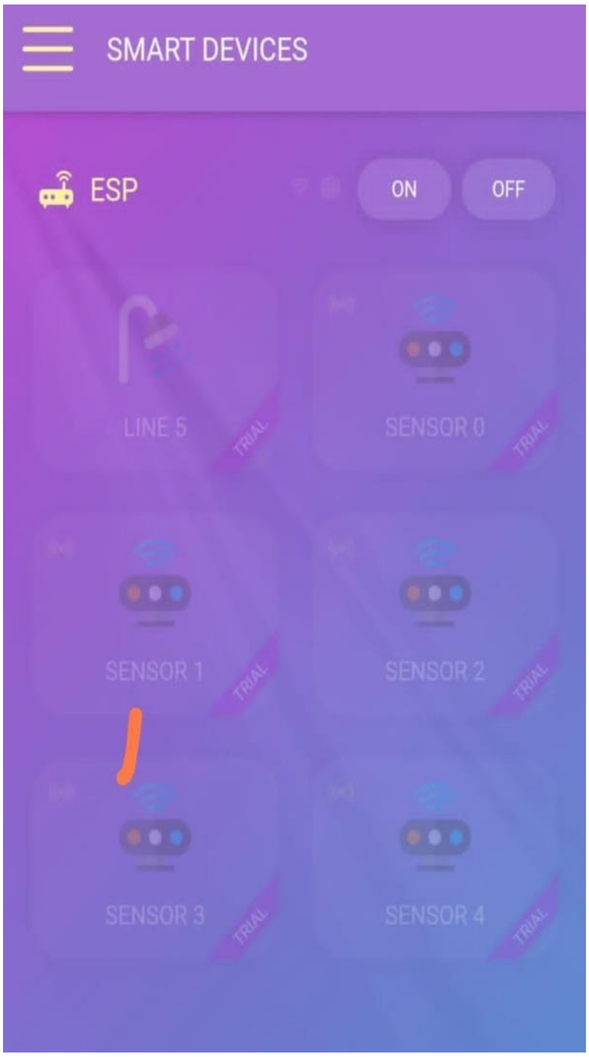


Figure 7.11: Cadio App Display

CHAPTER 8

CONCLUSION AND FUTURE ENHANCEMENTS

8.1 Conclusion

This is a great proposal for a NodeMCU based detection and management system to improve safety and reduce the risk of fire deaths and infrastructure damages caused by gas leakages. The proposal highlights the need for an automatic detection and management system to ensure the safety of people who may be unconscious during an emergency.The proposed system is low-cost and incorporates an automatic detection system to detect gas leakages. Additionally, the system includes a solution container that is automatically pumped to reduce there leased gases and prevent accidents from occurring.The system offers an innovative solution to gas leakages, which can be a significant safety issue.

In addition, the project can be incorporated into various systems to make them safe and efficient. Overall, the proposal is well-thought-out, and the proposed system can be beneficial to many people. The low-cost and efficient nature of the system makes it accessible to a wide range of people, thereby helping to reduce the risk of gas leakage and
prevent accidents from occurring. Arduino IoT cloud no need of importing any of the libraries like Arduino IDE or any need of the Blynk app installation.

8.2 Future Enhancements

This system is providing its requirement but adding some more features in this project wish to make the assignment more convenient in future. This Project will add real time camera sensor to provide the situations of the infrastructure.The future enhancement can be made by industries which are releasing hazards gas like styrene,Carbon monoxide etc. They can use this iot based detection project for detecting the styrene by using styrene detector as styrene detector costs much we haven't used here to suppress the styrene content stored in storage tank we can use the polymerization inhibitors as a solution in the container as inhibitors breaks the chain of the styrene polymer and converts styrene polymer to styrene monomer so by this we can use this to decrease the content.

REFERENCES

[1] Mohammad Monirujjaman Khan, Sensor-Based Gas Leakage Detector System,MDPI, 14 November 2020.

[2] Afsana Mim Anika Dept. of Computer Science and Engineering Daffodil International University Dhaka, Bangladesh, Gas Leakage with Auto Ventilation and Smart Management System Using IoT, IEEE 2021.

[3] Zaw Lin Oo Ministry of Science and Technology, Myanmar Theint Win Lai KBTC University, IoT Based LPG Gas Level Detection & Gas Leakage Accident Prevention with Alert System , Journal Vol. 9, No. 4, October 2021.

[4] Dr. Chalasani Srinivas ,Mohan Kumar.Ch , Toxic Gas Detection and Monitoring and utilizing Internet of Things International Journal of Civil Engineering and Technology (IJCIET) Volume 8, Issue 12, December 2017.

[5] Rohan Chandra Pandey1, Manish Verma2, Lumesh Kumar Sahu3,1Kalinga University, PG Student, Vlsi and Embedded System, Internet of Things (IOT) Based Gas Leakage Monitoring and Alerting System with MQ-2 Sensor, IJEDR | Volume 5, Issue 2, 2017.

[6] Noor Kareem Jumaa*, Younus Mohammed Abdulkhaleq, Muntadher Asaad Nadhim, Tariq Aziz Abbas, Al-Mansour University College, Computer Technology Eng. Dept., Baghdad, Iraq, IoT Based Gas Leakage Detection and Alarming System using Blynk platforms, Iraqi Journal for Electrical and Electronic Engineering, Vol. 18| Issue 1| June 2022.

[7] Prof. Pranay Meshram*1, Nancy Shukla, Stuti Mendhekar ,Renuka Gadge, Shivani Kanaskar, IoT Based LPG Gas Leakage Detector, Research Article, IJESC Volume 7 Issue No.4, 2017.

[8] Zaw Lin Oo,Pheint Win Lai and Aung Moe,IoT Based LPG Gas Level Detection and Gas Leakage Accident Prevention With Alert System,Bankan Journal, Oct 2022

[9] V. P. K. A. R. C. Sony Shrestha, "IOT BASED SMART GAS MANAGEMENT," in Electronics and Informatics (ICOEI 2019), India, 2019.

[10] S.D. P. P. K. A. H. Sachin Malviya*, "LPG Gas Leakage Detector System using IOT," International Journal of Scientific Research and Engineering Development, vol. 2, no. 6, 2019.

[11] Q. I. Sarhan, "Arduino Based Smart Home Warning System," in IEEE 6th International Conference on Control Science and Systems Engineering, Iraq , 2020.

[12] Bader Farhan Alshammari Electrical Engineering Department, Muhammad Tajammal Chughta

Electrical Engineering Department,Technology & Applied Science Research, Vol. 10, No. 4, 2020.

[13] P.Nagendra Siva Kumar1 Dept of Instrumentation Technology, Dr. A. Kamala Kumari2 Asst Prof, Department of Instrumentation Technology, International Journal of Engineering Research & Technology (IJERT), Vol. 8 Issue 11, November2019.

[14] Ravi Kishore Kodali Department of ECE, Greeshma, R.N.V. Department of ECE, IOT Based Industrial Plant Safety Gas Leakage Detection System, 4th International Conference on Computing Communication and Automation (ICCCA), IEEE 2018.

[15] Kiran Patil V 1, Kiran B.N2, Rampur Srinath3, Internet of Things for Green House Gas Detection System, International Research Journal of Engineering and Technology (IRJET), Volume: 03 Issue: 05 May-2016.

[16] Shaheduzzaman Chowdury, Design and Implementation of an IoT Based Air Pollution Detection and Monitoring System, IEEE 2019.

[17] Tamil Selvi a/p Sitan and Aimi Syamimi Ab Ghafar, Liquefied Petroleum Gas (LPG) Leakage Detection and Monitoring System, Journal of Science and Technology, Vol. 10 No. 3 2018.

[18] 1st stMethila Farzana Woishe, 2nd Nila Sultana, 3rdTamanna Zaman Bristy, 4thNafiza Anjum Khan, A Secured Model of IoT-based Smart Gas Detecting and Automatic Alarm System, International Journal of Computer and Information System (IJCIS), Vol. 03, Issue 02, April 2022.