RESEARCH PROJECT REPORT

ON

“Energy Efficient Lighting and Safe Environment through IoT-Enabled System”

Fergusson College (Autonomous)

AFFILIATED TO

SAVITRIBAI PHULE PUNE UNIVERSITY

FOR THE DEGREE OF

M.Sc.

ELECTRONIC SCIENCE

BY

Isha Daine (236901)

Department of Electronics Science,

Fergusson College (Autonomous), Pune-04

APRIL 2024-25

Acknowledgement:

We want to express our sincere gratitude to all those who provided their support and encouragement throughout this research project. Their invaluable assistance played a significant role in the successful completion of our work. We extend our heartfelt thanks to our project guides, Ms. Ranjana Ubale and Ms. Panchsheela Kamble, Assistant Professors in the Department of Electronic Science, Fergusson College (Autonomous), Pune. Their constant guidance, expert advice, and encouragement throughout the development of our project entitled ***"Energy-Efficient Lighting and Safe Environment through IoT Enabled system"*** have been instrumental in shaping our research and enriching our learning experience.

We would also like to extend our heartfelt gratitude to the Head of the Department and offer special thanks to our Postgraduate Staff, M.Sc. Coordinator Dr. Kalpana Kulkarni, Ms. Yogita Khandge, and the lab instructors of the *Department of Electronic Science, Fergusson College*, for their unwavering academic support and for providing the necessary infrastructure to carry out this project successfully.

We are especially thankful to the non-teaching staff of the department and the technical lab assistants for their kind cooperation, timely assistance, and valuable help in managing the logistics and resources during the practical phase of this work.

We are equally grateful to our classmates and friends for their constant encouragement, insightful discussions, and moral support throughout the course of this project.

This project has been a deeply enriching learning experience, and We are sincerely thankful to everyone who contributed to its successful completion.

Table of Content:

Sr.No	Details	Page No
Chapter 1		5
1.1	Importance of project	5
1.2	Problem definition	5
1.3	Aim and Objective	6
1.4	Introduction	7
Chapter 2	**Fundamentals of____**	8
2.1	Literature survey	8
2.2	Basic Theory	9
Chapter3	**Project proposal**	10
3.1	Component Specifications	10
3.2	System Block Diagram	14
3.3	Method of Implementation	14
3.4	Task vs Time schedule	15
Chapter 4	**Planning Resources**	16
4.1	Hardware	16
4.2	Software	17
4.3	Miscellaneous	17
4.4	BOM	18
Chapter 5	**Experimentation & Result**	19
5.1	Experimental setup	19
5.2	Lighting System Test	20
5.3	Gas Leak Detection Test	25
5.4	Door Monitoring Test	26
Chapter 6	**Summary & Future Scope**	27
6.1	Summary	27
6.2	Future Scope	27
6.3	References	28
6.4	System Code	29

Table of Figures:

No.	Name of Figures	Page No.
1	Fig 1: ESP32 Microcontroller	10
2	Fig 2: MQ-6 Gas Sensor	10
3	Fig 3: Ultrasonic Sensor (HC-SR04)	11
4	Fig 4: Reed Switch	11
5	Fig 5.1: Digital Dimmer Module	12
6	Fig 5.2: Pin Connections	12
7	Fig 7: Block diagram of IoT based prototype system	14
8	Fig 8: Hardware Set up	16
9	Fig 9.1: Hardware Set up image 1	19
10	Fig 9.2: Hardware Set up image 2	19
11	Fig 10.1: Filament partially on	20
12	Fig 10.2: lux of partially light on	20
13	Fig 10.3: Filament fully on	20
14	Fig 10.4: lux of fully light on	20
15	Fig 11.1: LED partially on, person is not detected	21
16	Fig 11.2: Lux of partially on light	21
17	Fig 11.3: LED fully on, Perso is detected	21
18	Fig 11.4: Lux of fully on light	21
19	Fig 12: Comparison of Power consumption: led vs filament bulb	22
20	Fig 13: Hourly energy usage comparison	25
21	Fig 14.1: leakage of gas	25
22	Fig 14.2: E-mail notification	25
23	Fig 15.1: Door open condition	26
24	Fig 15.2: Door Close condition	26
25	Fig 15.3: Door open Alert notification	26

Index of Tables:

No.	Table Name	Page No.
1	Table 1: Observation of Filament bulb	22
2	Table 2: Observation of Led bulb	23
3	Table 3: Observation table of energy saving of LED bulb	24

Chapter 1: Introduction

1.1 Importance of the project:

This research addresses two critical challenges in modern infrastructure: energy efficiency and occupant safety. Traditional systems often lead to excessive energy consumption and delayed responses to hazards like gas leaks and unauthorized access. This study introduces a low-cost, IoT-enabled smart building solution using the ESP32 microcontroller, integrating intelligent lighting control and real-time safety alerts. Presence-based lighting modulation using an ultrasonic sensor reduces energy use, with empirical data showing significant savings compared to conventional systems. The system also enhances safety through a gas leak detection mechanism using the MQ-6 sensor, which sends immediate email alerts, and a door monitoring subsystem that provides real-time notifications to the user's smartphone via the Blynk IoT platform. The ESP32 platform ensures low power consumption and real-time processing, making the system scalable for residential and commercial environments. This research demonstrates how IoT-based technologies can reduce energy usage and improve safety, supporting broader objectives in sustainable building practices and responsive environmental control.

1.2 Problem definition

In many buildings, electricity is wasted due to lights remaining on in unoccupied areas, and there is often a lack of real-time systems to detect safety hazards such as gas leaks or unauthorized door access. Traditional systems are not equipped to respond intelligently or promptly to such situations, leading to increased energy consumption and compromised safety. To address these challenges, this project proposes the development of an IoT-based solution using the ESP32 microcontroller. The system will automate lighting based on occupancy, detect gas leaks, and monitor door status, while providing real-time alerts to enhance both energy efficiency and safety.

1.3 Aim and Objective:

To design and implement a cost-effective, IoT-enabled smart building system using ESP32 for enhanced energy efficiency and safety. The system includes intelligent lighting, gas leakage alerts, and door access monitoring, offering a modular and scalable automation solution for modern buildings.

To achieve this aim, the following specific objectives have been established:

- To develop an energy-efficient smart lighting system that utilizes occupancy detection to automatically adjust light intensity, thereby minimizing unnecessary energy consumption in building spaces.
- To implement a real-time gas leakage detection mechanism using the MQ-6 sensor, with immediate alert notifications sent via email through the SMTP protocol for a timely response.
- To design a door intrusion monitoring system using a magnetic reed switch for enhanced security.
- To integrate all system components using the ESP32 microcontroller, enabling wireless data transmission and remote alert capabilities.

1.4 Introduction:

Today, as we look for ways to make our homes and buildings smarter, energy efficiency and safety have become top priorities. Most traditional systems like lights that stay on all the time or gas detectors that aren't connected to any alerts end to waste energy or react too slowly during emergencies. With technology advancing so rapidly, there's a growing need to make our living and working spaces more intelligent and responsive. affordable and powerful microcontrollers like the ESP32, it's now possible to build smart systems that not only save electricity but also help keep us safe. The ESP32, with its built-in Wi-Fi and real-time capabilities, makes it easy to connect sensors and send alerts without needing expensive infrastructure. In this project, we've designed a smart building system that tackles two key things: reducing electricity usage through smart lighting and improving safety through real-time alerts. For energy saving, we've set up an ultrasonic sensor that can detect when someone enters a building lobby and automatically turn the lights on to full brightness. When no one is around, the system dims the lights to save power. We tracked energy use for a month and found a noticeable drop in electricity consumption compared to regular lighting systems. On the safety side, we created two additional features. One detects gas leaks using an MQ-6 sensor and sends an alert to the homeowner's email. The other monitors door activity using a magnetic reed switch—if the door opens unexpectedly, the system sends a notification to the user's phone through the Blynk IoT platform. This keeps users informed even when they're not at home. Overall, this project shows how simple and cost-effective technology can be used to make homes smarter, safer, and more energy-efficient. The system is flexible and can be adapted for different types of buildings, making it a practical solution for modern living.

Chapter 2: Fundamentals of energy-efficient lighting and safety monitoring system

2.1 Literature Survey:

[1] Development of a microcontroller-based digital ac dimmer for light intensity control

Neeraj Khera and his team designed a smart AC light dimmer using an Arduino that lets users control brightness either by touch or through Bluetooth from a smartphone. Their system improved convenience over traditional manual dimmers but didn't include internet connectivity or any safety features. Building on their work, our project uses the ESP32 to enable Wi-Fi-based control and adds smart safety features like gas leak detection and door monitoring**,** making it more practical and reliable for today's smart homes and buildings.

[2] A smart energy monitoring using esp32 microcontroller

The increasing adoption of IoT and microcontroller-based systems, such as ESP32, for real-time energy monitoring and management. Studies have shown that platforms like Blynk enhance user interaction by providing mobile-based visualization and control. Prior research by Babiuch et al. demonstrated ESP32's efficiency in data processing and energy monitoring applications. Other works emphasized the integration of smart meters and wireless communication for improved energy usage insights. Moreover, incorporating ICT and AI has been proven to enhance energy efficiency, especially in smart homes and industrial setups. This study builds on these foundations by offering a low-cost, real-time monitoring solution tailored to regions with limited energy access.

[3] IoT based smart home automation system

The literature review explores multiple IoT-based home automation systems using platforms like Raspberry Pi, Arduino, and NodeMCU. These systems are designed to enhance user convenience, improve energy efficiency, and provide better home security. Control methods include Android apps, voice commands, and desktop applications. Sensors like PIR, LDR, and IR are commonly used for automatic control of appliances based on environmental conditions. Some systems also incorporate biometric and motion detection for enhanced safety. Most models aim to be cost-effective, scalable, and easy to use. However, connectivity limitations and interface complexity still present challenges.

Conclusion The reviewed papers provide a detailed understanding how smart home and energy systems have come a long way with the help of microcontrollers and IoT. The first paper focused on a simple light dimmer using Arduino, which made light control easier but didn't have internet features or safety measures. The second paper took things further by using the ESP32 to monitor energy use in real time and let users track everything through their phones, making it more practical and affordable. The third paper looked at full home automation using sensors and voice or app control, improving comfort and security. While these systems are getting smarter and more user-friendly, they still face issues like tricky interfaces and internet dependency. Overall, they reflect how tech is making homes more efficient, connected, and safer.

2.2 Basic theory

This project integrates core principles of embedded systems, sensor networks, and Internet of Things (IoT) to enhance energy efficiency and safety in smart buildings. The system is powered by the ESP32 microcontroller, a dual-core 32-bit SoC with built-in Wi-Fi and Bluetooth, capable of handling real-time data processing and wireless communication.

For lighting automation, an **HC-SR04 ultrasonic sensor** measures distance using time-of-flight calculations based on ultrasonic pulse reflection. The ESP32 uses this data to detect occupancy and adjust lighting via an **AC dimmer module**, conserving energy by dimming lights during inactivity.

The safety subsystem employs two sensors:

- An **MQ-6 gas sensor**, which detects combustible gases by measuring changes in resistance of a tin dioxide (SnO_2) layer. Upon detecting a leak, the ESP32 sends an email alert using SMTP.
- A **magnetic reed switch** for intrusion detection, which triggers a signal when a door is opened. The ESP32 sends a real-time push notification through the **Blynk IoT platform**.

This architecture demonstrates how low-cost microcontrollers and sensors can be integrated to build scalable, responsive smart building solutions.

Chapter 3: Project Proposal- Energy efficient lighting and safe environment through IoT enabled system

3.1 Component Specifications:

3.1.1 ESP32 Microcontroller:

Fig 1: ESP32 Microcontroller

The ESP32 is a powerful, low-cost microcontroller developed by Espressif Systems, widely used in IoT (Internet of Things) applications. It is an advanced version of the ESP8266 and is known for its excellent performance, low power consumption, and integrated wireless communication features, it is a very versatile, low-cost, robust, and low-power-based soc Chip general-purpose microcontroller with a dual-core processor.

This device provides a wide range of features like: a large memory capacity, a dual-core processor & a rich set of peripherals. As compared to other types of microcontrollers, these devices offer different features, a superior power balance, and flexibility.

3.1.2 MQ-6 Gas Sensor

Fig 2: MQ-6 Gas Sensor

The MQ-6 is a low-cost gas sensor designed to detect flammable gases such as LPG, propane, butane, and, to a lesser extent, methane and hydrogen. It uses a tin dioxide (SnO_2) sensing layer, whose resistance decreases in the presence of gas, producing an analog voltage output proportional to gas concentration. The sensor also provides a digital output that triggers when a set threshold is exceeded.

Operating at 5V with a current draw of ~150 mA, the MQ-6 has a detection range of 200–10,000 ppm and a typical response time under 10 seconds. It requires a short preheat period for optimal

sensitivity. In this project, the ESP32 microcontroller continuously reads the analog output of the MQ-6. When the gas concentration surpasses a safe limit, the ESP32 sends an email alert using SMTP, enabling real-time remote gas leak notifications to enhance building safety.

3.1.3 Ultrasonic Sensor

Fig 3: Ultrasonic Sensor (HC-SR04)

An ultrasonic sensor is a widely used electronic device that measures the distance to an object by using ultrasonic sound waves**.** It works by emitting a high-frequency sound pulse (usually around 40 kHz) from its transmitter. This sound travels through the air, and when it hits an object, it reflects to the sensor's receiver. This sensor is perfect for measuring distances without touching anything.

It can detect objects between 2 cm and 400 cm away (about 13 feet) with impressive accuracy of about 3 millimeters. The sensor then calculates the distance to the object based on the time taken for the echo to return using the formula: distance = (speed of sound × time) / 2. In this research project, the ultrasonic sensor is used to detect human presence in building lobbies. When a person enters the sensor's range, the ESP32 microcontroller increases the lighting to full brightness. If no presence is detected for a set time, the system dims the light to save energy.

3.1.4 Reed switch

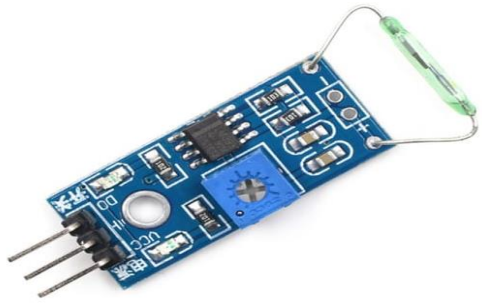

Fig 4: Reed Switch

A reed switch is a compact electromagnetic sensor widely used in security systems to detect the opening and closing of doors and windows. It consists of two thin, flexible metal reeds enclosed in a small glass tube filled with inert gas. When a magnetic field is brought close—typically by placing a permanent magnet nearby the reeds attract and close the circuit.

Once the magnet is removed, the reeds separate, breaking the circuit. This simple mechanism enables the reed switch to act as a reliable binary sensor with minimal power consumption and long operational life. In this project, the reed switch is used to keep an eye on the main door. It's connected to the ESP32 microcontroller, which constantly checks the switch’s status. If the door is opened when it shouldn’t be, the ESP32 quickly sends a real-time alert to the user’s phone using the Blynk app. This makes it easy for homeowners to stay informed and feel secure, even when they're not at home.

3.1.5 Digital Dimmer Module

Fig 5.1: Digital Dimmer Module

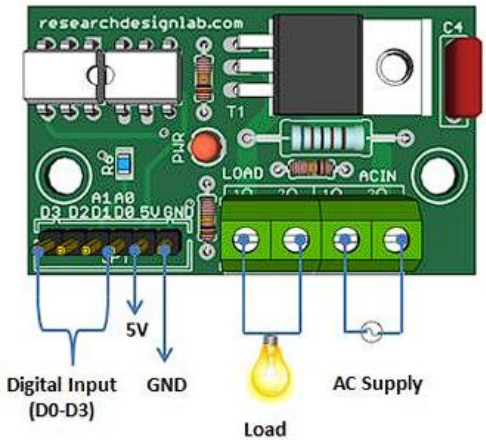

Fig 5.2: Pin Connections

A DC dimmer module is an electronic circuit designed to control the intensity of DC loads. Instead of just turning the light on or off, this module allows you to adjust the brightness smoothly, depending on what’s needed. It does this using a technique called PWM., which means it sends very fast on-and-off signals to the light, and by changing how long it stays “on” during each cycle, the light gets brighter or dimmer. This module has easy screw terminals to connect your power supply and light, and it also has pins that you can connect to a microcontroller like the ESP32. In your smart lighting system, the ESP32 can send signals to this module based on whether a person is nearby or not (using an ultrasonic sensor). If someone is present, the ESP32 tells the dimmer to make the light bright. If no one is there, it dims the light to save power. It’s a simple, The board uses BTA12-600B, Triac suitable for general purpose mains power AC switching.

Features:

- Dimming method: Phase dimming
- On/Off and dimming control
- Circuit enabled with snubber
- Auto Zero cross detection
- Accurate firing angle control and smooth dimming
- Supply voltage +5 DC
- Isolated power section from the input
- Load Capacity 6 Amp AC (Up to 1500 Watt)
- Works from any microcontroller input.

Application:

- LED light dimming using phase control
- Motor/Fan speed control
- High speed ON/OFF control application
- Heater/Temperature control application
- PID Temperature Control System
- PID AC single-phase speed control system

3.2 System Block Diagram:

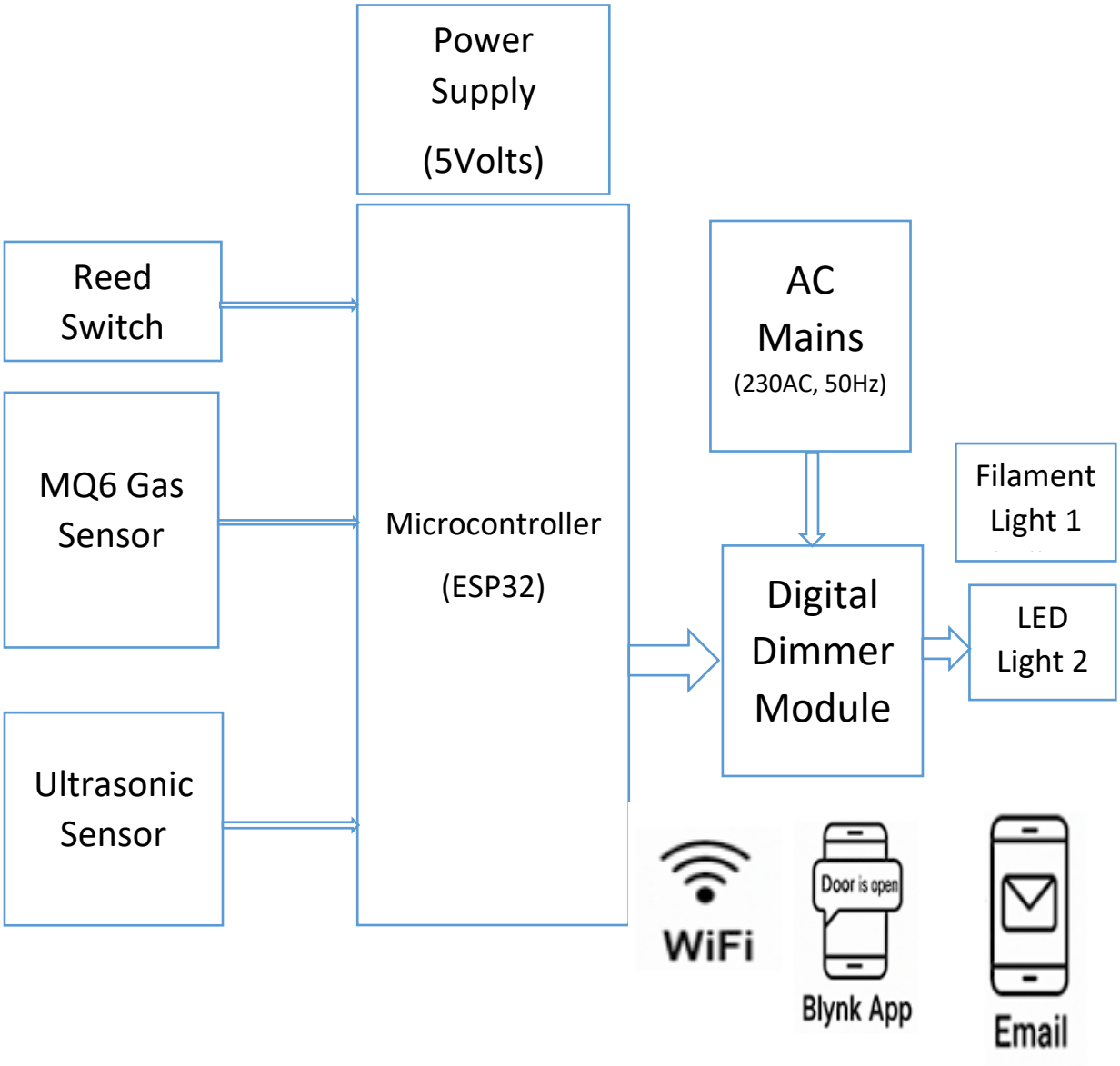

Fig 7: Block diagram of the proposed IoT-based prototype system.

3.3 Method of Implementation:

This smart building system was developed using the ESP32 microcontroller due to its integrated Wi-Fi, real-time performance, and energy-efficient design. The system integrates three key modules: presence-based lighting control, gas leak detection, and door activity monitoring. Each module was designed, tested individually, and later combined into a unified system to ensure functionality and stability. The intelligent lighting subsystem uses the HC-SR04 ultrasonic sensor

to detect human presence within the monitored area. When movement is detected within a specified range, the ESP32 activates a DC dimmer module to switch the lighting to full brightness. If no motion is detected, a five-second delay is introduced before dimming the lights to a partial brightness level. Electrical parameters such as voltage, current, power, and energy consumption were recorded continuously for one month in both lighting states. This data was analyzed to assess energy savings compared to traditional lighting systems. For gas leak detection, the MQ-6 sensor was connected to the ESP32 to monitor concentrations of combustible gases such as LPG and butane. When levels exceeded a defined threshold, the ESP32 triggered an email alert using the SMTP protocol, allowing immediate remote notification to the homeowner. The door monitoring system employed a magnetic reed switch that outputs a HIGH signal when the door is opened and a LOW signal when closed. The ESP32 processed this signal and used the Blynk IoT platform to send real-time notifications to the user's smartphone, ensuring continuous awareness of door status. All modules communicated over a common Wi-Fi network, with Blynk handling mobile alerts and SMTP managing email communication. The overall system was designed to be modular and scalable, enabling future expansion with additional sensors or smart automation features.

3.4 Task Vs Time Schedule

Month	January				February				March				April			
Week	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4
Finalization of Research Problem	✓	✓	✓	✓												
System Design & Block Diagram		✓	✓	✓	✓	✓										
Hardware Setup (ESP32, Sensors, etc.)				✓	✓	✓	✓									
Software Development (Arduino/ESP32 Code)								✓	✓	✓	✓					
Integration & Testing of Modules											✓	✓	✓	✓		
Data Collection & Analysis												✓	✓	✓		
Report Writing & Final Research Proposal													✓	✓	✓	✓

Chapter 4: Planning Resources

4.1 Hardware

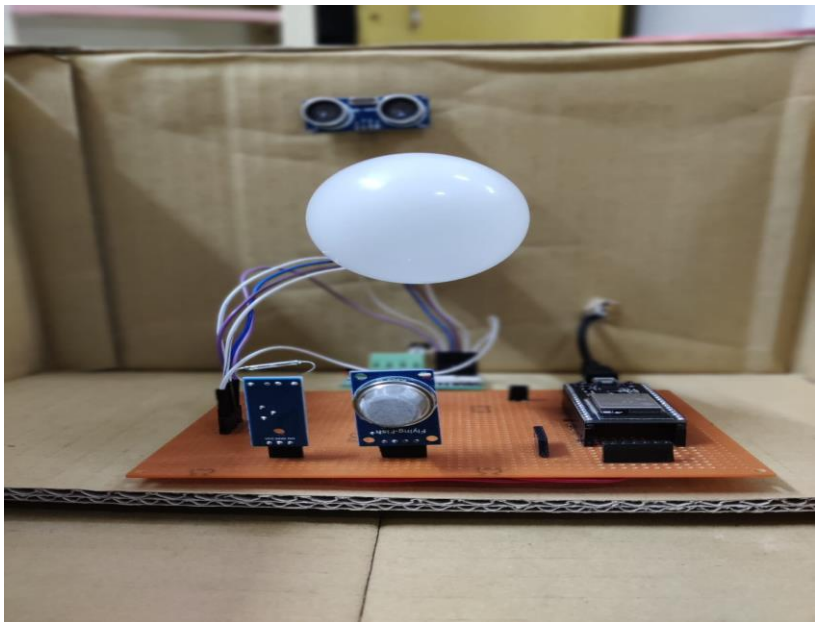

Fig 8: Hardware Set up

This smart system is built using various electronic components integrated with a microcontroller to achieve efficient lighting control and real-time safety alerts.

1 ESP32 microcontroller: Acts as the main controller of the system, manages sensor data, lighting control, and network communication. It is Connected to the Blynk IoT platform and SMTP server for mobile and email alerts.

2 HC-SR04 Ultrasonic sensor: Used for detecting human presence based on distance measurement. Sends real-time proximity data to the ESP32 to trigger lighting behavior.

3 DC Dimmer Module: Controls the brightness of connected DC-powered lights, helps in reducing energy consumption during inactivity.

4 MQ-6 Gas Sensor: Detects flammable gases such as LPG, propane, and butane, Provides analog and digital output to the ESP32 for continuous monitoring.

5 Magnetic Reed Switch: Mounted on a door in the prototype to detect opening and closing events, sends a signal to the ESP32 to trigger a Blynk notification when the door is accessed.

6 LED bulb and Filament bulb: Connected to the dimmer module to demonstrate real-world lighting control, used for testing brightness adjustments and energy efficiency in the prototype.

Also used external hardware like jumper wires for the connection of circuit, Zero PCB, cardboard for prototype demonstration, external ac power supply for the ac load .

4.2 Software:

- The ESP32 was programmed using the **Arduino IDE,** with necessary libraries for Wi-Fi, Blynk, sensor input, PWM dimming, and email alerts.
- The **Blynk IoT platform** was used to display real-time door notifications on the user's mobile phone when the reed switch was triggered.
- For gas leak detection, the ESP32 sends an email alert via SMTP email Service when the MQ-6 sensor detects dangerous gas levels.
- A **Light Meter mobile app** was used to measure light intensity during testing to verify the dimmer module's performance under different conditions.

4.3 Miscellaneous

During the development of this project, various components were individually tested to ensure reliability and compatibility. Sensors such as the IR sensor, MQ-6 gas sensor, reed switch, and ultrasonic sensor were all evaluated through standalone code before being integrated into the final system.

Initially, a PIR sensor was used for motion detection to implement the smart lighting feature. However, it introduced some second delay and lacked responsiveness due to its non-repeatable mode, making it unsuitable for real-time occupancy detection. As a result, the ultrasonic sensor was adopted, which provided faster and more accurate distance-based detection, making the system more responsive and practical. The IR sensor was also tested as an alternative but was found to have limitations in terms of angle and distance detection, especially in larger spaces or areas with varying light conditions. Additionally, the code was developed modularly—each sensor was first tested independently, and later the code was merged to ensure smooth coordination between lighting control, gas detection, and door monitoring.

To visualize the system setup for demonstration purposes, a cardboard prototype display was created, helping to neatly arrange components and simulate a real-world application

4.4 Bill of Materials:

Sr No.	Component	Quantity	Remarks
1	ESP32 uc	1	Main controller for the system
2	Reed switch module	1	door/opening sensor
3	Ultrasonic (HC-SR04)	1	person detection
4	MQ6 gas sensor	1	Detects LPG, butane, methane
5	DC dimmer	1	Controls brightness of led/ filament bulb
6	LED bulb	1	Energy-efficient lighting
7	Filament Bulb	1	traditional lighting example
8	Zero PCB	2	for assembling components
9	Wires, Connectors	1	internal connection
10	Bulb Holder	1	hold the bulb

Chapter 5: Experiments and Results

5.1 Experimental setup:

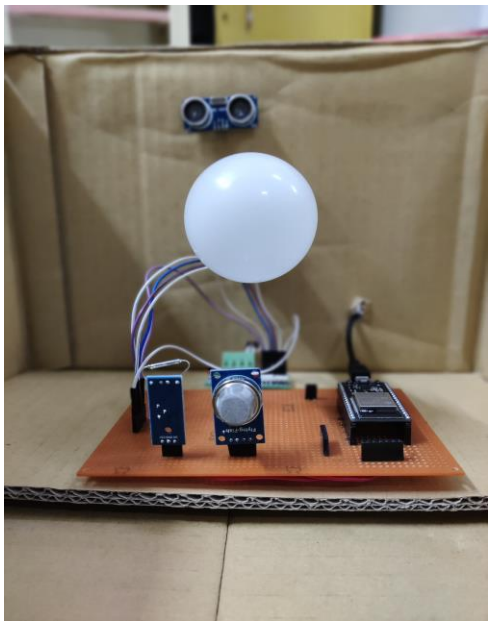

Fig 9.1: Hardware Set up image 1

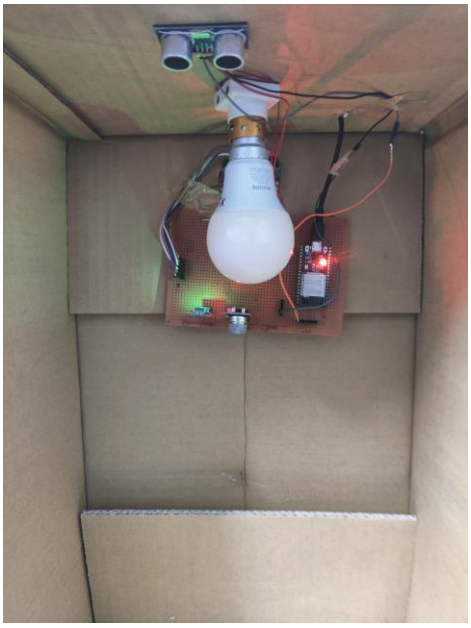

Fig 9.2: Hardware Set up image 2

Above figure 9.1 and 9.2 shows the experimental set up 57 cm high cardboard enclosure to simulate a controlled indoor environment. An HC-SR04 ultrasonic sensor and a bulb holder were mounted on the top surface, allowing for the testing of both LED and filament bulb under different lighting conditions. A custom zero PCB was designed and mounted on the side wall of the enclosure. It hosted key components including the ESP32 microcontroller, MQ-6 gas sensor, magnetic reed switch, and a DC dimmer module. Electrical connections were made using standard jumper wires. The DC dimmer module was powered via AC mains, with its output connected to the bulbs to control brightness levels. For evaluation, A power meter measured energy consumption in both partial and full brightness modes. A Light Meter mobile app was used to record lux levels at a consistent 50 cm distance from the light source. This setup enabled accurate assessment of energy efficiency, sensor response, and lighting performance within a compact prototype environment.

5.2 Lighting system test:

5.2.1 Experiment of filament Bulb:

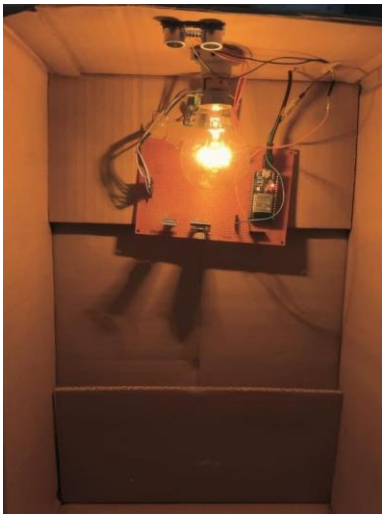

Fig 10.1: Filament partially on

Fig 10.2: lux of partially on Light

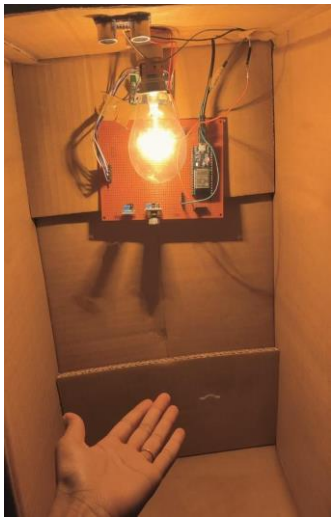

Fig 10.3: Filament fully on

Fig 10.4: lux of fully on Light

The lighting control system was tested using a filament bulb connected through a DC dimmer module which shown in above figure 10.1, 10.2, 10.3 and 10.4. When no object was detected within range by the HC-SR04 ultrasonic sensor, the bulb operated in partial brightness mode, with a measured illuminance of 51 lux (measured at 50 cm using a mobile light meter application). Upon presence detection, the ESP32 triggered the bulb to full brightness, reaching an intensity of 571 lux. The sensor continuously monitored the distance; once the object was no longer detected, the system initiated a 5-second delay, after which the bulb automatically returned to partial brightness mode. This behavior demonstrates adaptive lighting control for energy efficiency based on real-time occupancy sensing.

Condition	Power Consume (W)	Intensity of Light(Lux)
Partially ON	35.82	51
Fully ON	89	372

Table 1: Observation of Filament Bulb

Power saved= fully on- Partially on

=89-35.82

=**53.18 W**, Power saved by the filament bulb during the partially on time period

Percentage of power saved=power saved/power consumption when fully on

= (53.18/ 89) *100

= 59.75%

5.2.2 Experiment of LED Bulb:

Fig 11.1: LED partially on, person is not present

Fig 11.2: lux of partially on Light

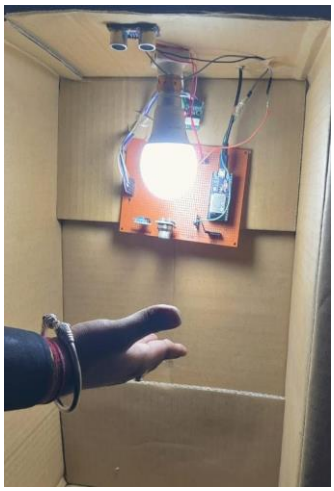

Fig 11.3: LED fully on, person is detected

Fig 11.4: lux of fully on Light

Above figure 11.1,11.2,11.3,11.4 shows that when no presence was detected by the HC-SR04 sensor, the LED bulb operated in partial brightness mode with an intensity of 861 lux measured at 50 cm. Upon detecting a person, the system increased brightness to full, reaching 2699 lux. After the person left, the light returned to partial mode following a 5-second delay for energy efficiency.

Condition	Power Consume (W)	Intensity of Light(Lux)
Partially ON	2.64	861
Fully ON	7.8	2699

Table 2: Observation of LED Bulb

Power saved= fully on- Partially on

=7.8-2.64

=**5.16 W,** Power saved by the LED bulb during the partially on time period

Percentage of power saved=power saved/power consumption when fully on

= (5.16/7.8) *100%

= 66.15%

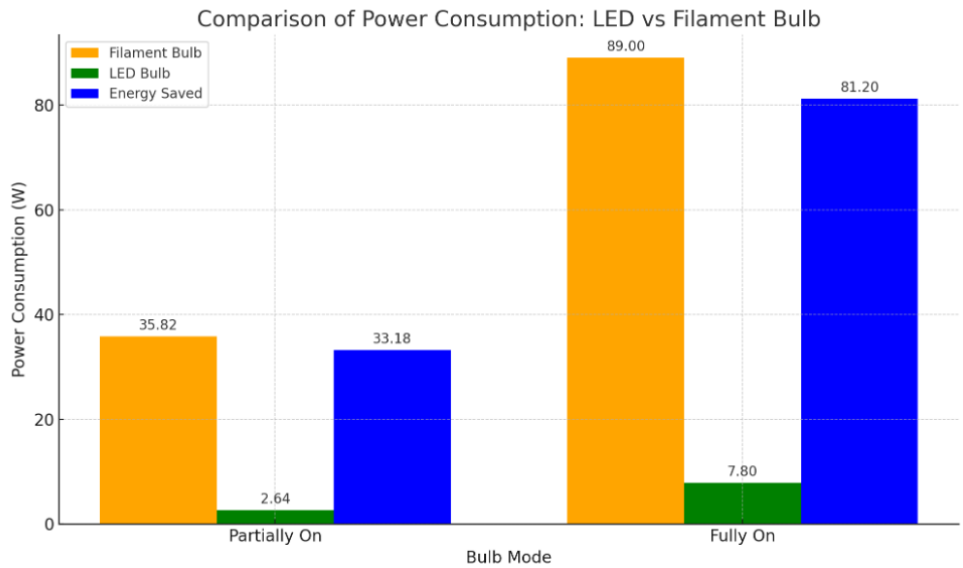

Fig 12: Comparison of Power consumption: LED vs Filament bulb

The power consumption of both filament and LED bulbs was measured in partial and full brightness modes. The filament bulb consumed 35.82 W when dimmed and 89 W at full brightness, indicating high energy usage in both states. In contrast, the LED bulb was significantly more efficient, drawing only 2.64 W in partial mode and 7.8 W at full brightness. The above graph (Fig 12) clearly demonstrates the superior energy efficiency of LED lighting, making it a more sustainable choice for smart building applications.

5.2.3 To calculate the energy saved during the partially on time for Led bulb:

Fully ON power = 7.8 W

Partially ON power = 2.64 W

Fully ON time = 5 minutes = 5/60 = 1/12 hour

Partially ON time = 55 minutes = 55/60 = 11/12 hour

Energy consumed in 1 hour:

If the bulb was fully ON for the whole hour:

Energy = Power × Time = 7.8 W × 1 hr = 7.8 Wh

Actual energy used:

= (7.8 × 1/12) + (2.64 × 11/12)

= 0.65 Wh + 2.42 Wh

= 3.07 Wh

Energy saved:

= 7.8 Wh (if fully on) - 3.07 Wh (actual)

= 4.73 Wh energy is saved for 55 min of time.

According to above calculation, took the data of led bulb for 24 hours which shown below the Table 3

Hour	Motion Detected?	Fully ON Time (min)	Partially ON Time (min)	Energy - Fully ON (Wh)	Energy - Par ON (Wh)	Total Energy Used (Wh)	Fully ON Energy (Wh)	Energy Saved (Wh)
00–01	Rare	5	55	0.65	2.42	3.07	7.8	4.73
01–02	No	0	60	0	2.64	2.64	7.8	5.16
02–03	No	0	60	0	2.64	2.64	7.8	5.16
03–04	No	0	60	0	2.64	2.64	7.8	5.16
04–05	No	0	60	0	2.64	2.64	7.8	5.16
05–06	Rare	5	55	0.65	2.42	3.07	7.8	4.73
06–07	Low	10	50	1.3	2.2	3.5	7.8	4.3
07–08	Moderate	15	45	1.95	1.98	3.93	7.8	3.87
08–09	High	30	30	3.9	1.32	5.22	7.8	2.58
09–10	High	35	25	4.55	1.1	5.65	7.8	2.15
10–11	High	40	20	5.2	0.88	6.08	7.8	1.72
11–12	Moderate	20	40	2.6	1.76	4.36	7.8	3.44
12–13	Moderate	25	35	3.25	1.54	4.79	7.8	3.01
13–14	High	35	25	4.55	1.1	5.65	7.8	2.15
14–15	Moderate	20	40	2.6	1.76	4.36	7.8	3.44
15–16	Low	10	50	1.3	2.2	3.5	7.8	4.3
16–17	Moderate	25	35	3.25	1.54	4.79	7.8	3.01
17–18	High	35	25	4.55	1.1	5.65	7.8	2.15
18–19	High	40	20	5.2	0.88	6.08	7.8	1.72
19–20	Moderate	20	40	2.6	1.76	4.36	7.8	3.44
20–21	Low	10	50	1.3	2.2	3.5	7.8	4.3
21–22	Rare	5	55	0.65	2.42	3.07	7.8	4.73
22–23	low	10	50	1.3	2.2	3.5	7.8	4.3
23–24	No	0	60	0	2.64	2.64	7.8	5.16
Total		**395**	**1045**	**51.35**	**45.98**	**97.33**	**187.2**	**89.87**

Table 3: Observation table of energy saving of LED bulb

Conclusion:

- Total Energy Used (Smart Mode): 97.33 Wh
- Energy Used if Always Fully ON: 187.2 Wh (7.8 W × 24 h)
- Total Energy Saved: 89.87 Wh, or about 48% per day

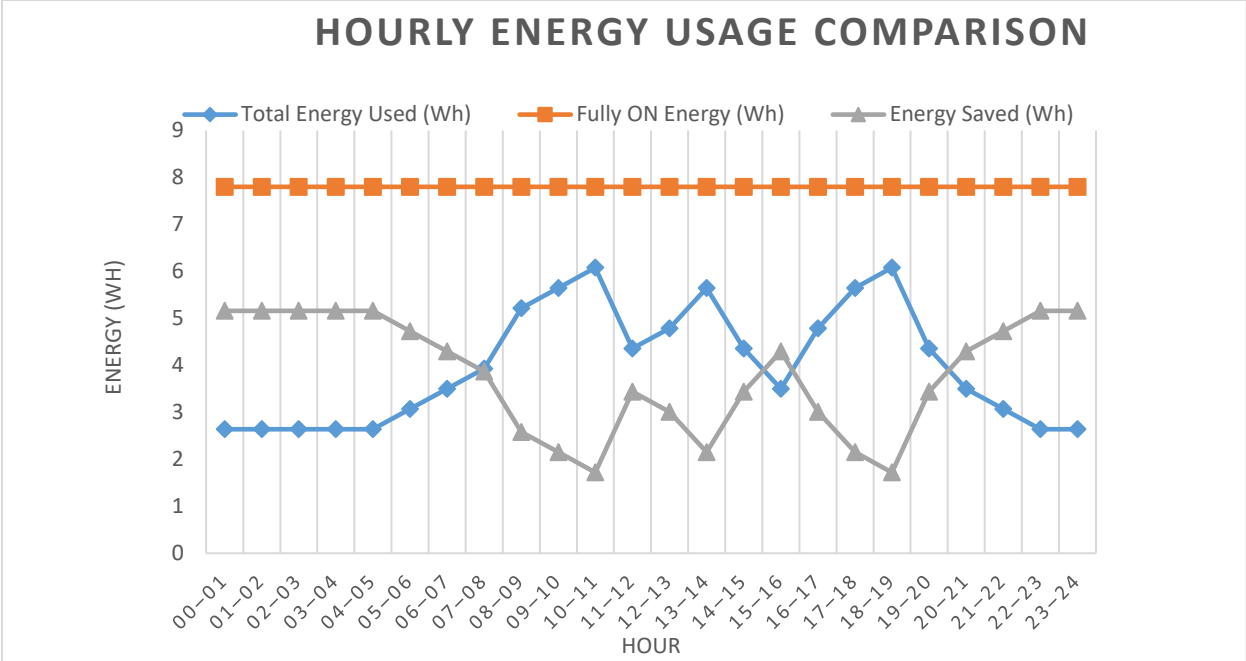

Fig 13: Hourly energy usage comparison

The above graph (Fig 13) compares total energy consumption over a 24-hour period in two scenarios: Smart Mode (with automatic dimming and occupancy detection) and Always ON mode. In Smart Mode, the LED lighting system used 97.33 Wh, while keeping the light fully ON for 24 hours would have consumed 187.2 Wh (7.8 W × 24 h). This results in an energy savings of 89.87 Wh, which is approximately 48% reduction in daily energy usage. The graph highlights the effectiveness of smart control in optimizing power consumption and improving energy efficiency**.**

5.3 Gas Leak Detection Test:

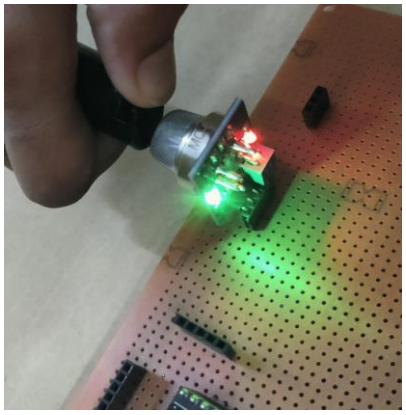

Fig 14.1: Leakage of gas

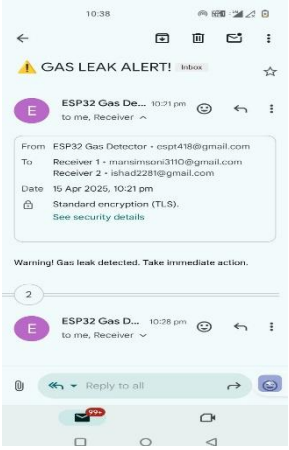

Fig 14.2: E-mail Notification

From the above figure 14.1, 14.2 shows that to validate the gas leak alert system, a controlled leak was simulated using a gas lighter near the MQ-6 sensor. Upon detection of flammable gas, the ESP32 immediately triggered an email alert to the user using the SMTP protocol. The test confirmed reliable notification delivery, with a screenshot of the received email shown in the report. Notably, the ESP32 and mobile device were connected to different networks, yet the alert was successfully transmitted without error, demonstrating the system's robustness and cloud-based accessibility. This ensures users receive real-time warnings from any location, enabling prompt response to hazardous situations.

5.4 Door Monitoring System:

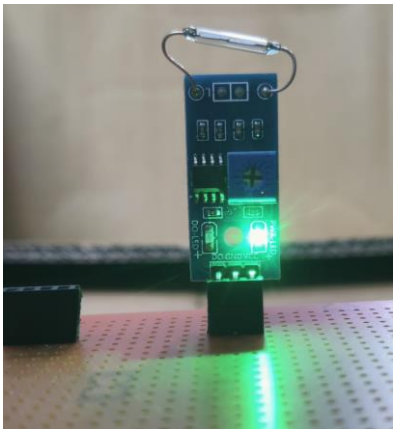

Fig 15.1: Door open condition

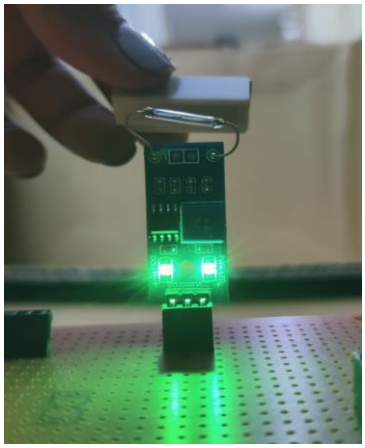

Fig 15.2: Door Closed condition

Fig 15.3: Door open Alert notification

The above figure 15.1,15.2,15.3 shows that the door monitoring system effectively detects the door's status based on the proximity of the magnet to the reed switch. When the door is closed, the magnet aligns with the reed sensor, maintaining a closed circuit. Conversely, when the door is opened, the magnet is displaced, causing the switch to open and triggering the ESP32 to send a real-time notification via the Blynk IoT application. This demonstrates the system’s ability to remotely monitor door activity and immediately alert the user to any unauthorized access or movement.

Chapter 6: Summary and Future Scope

6.1 Summary:

This study presents the design and implementation of an IoT-based smart building system using the ESP32 microcontroller. An ultrasonic sensor controls LED lighting based on occupancy, dynamically adjusting brightness to conserve energy. Experimental results over a 24-hour cycle show a reduction in energy consumption from 187.2 Wh to 97.33 Wh, indicating 48% energy savings. A comparative analysis showed that partially ON LED and filament bulbs achieved power savings of 66.15% and 59.75%, respectively. The system also integrates an MQ-6 gas sensor for LPG leak detection, sending real-time alerts via SMTP. Additionally, a magnetic reed switch is used for door status monitoring, with notifications delivered through the Blynk IoT platform. All modules communicate wirelessly using the ESP32's onboard Wi-Fi. The architecture is modular, cost-effective, and scalable for residential and commercial deployment. This research demonstrates a practical approach to real-time, energy-efficient automation with enhanced safety monitoring.

6.2 Future Scope:

- This system can become even smarter by using AI to learn user habits and automate lighting or alerts more intelligently. A dedicated mobile app could make it easier to control and monitor everything in one place.
- By saving data to the cloud, users could track energy usage over time and make better decisions. Adding voice control through Alexa or Google Assistant would make the system more accessible.
- It could also include more sensors like for smoke or temperature to boost safety. With battery backup and GSM alerts, it would work even during power or internet outages. Plus, its flexible design means it can be scaled up for larger buildings or workplaces in the future.

6.3 References:

[1] H. J. El-Khozondar et al., "A smart energy monitoring system using ESP32 microcontroller," *e-Prime – Advances in Electrical Engineering, Electronics and Energy*, vol. 9, 2024. [Online]. Available: https://doi.org/10.1016/j.prime.2024.100666

[2] S. Aashik, P. Jayarama, and N. Abishlal, "Smart energy management and load monitoring of individual loads," *E3S Web of Conferences*, vol. 405, 2023. https://www.e3s-conferences.org/articles/e3sconf/abs/2023/76/contents/contents.html

[3] C. H. Lien, H. C. Chen, Y. W. Bai, and M. B. Lin, "Power monitoring and control for electric home appliances based on power line communication," in *2008 IEEE Instrumentation and Measurement Technology Conference*, IEEE, 2008, pp. 2170–2175. [Online]. Available: https://ieeexplore.ieee.org/document/4547020

[4] Aashik, S., Jayarama, P., Abishlal, N. S., & Benedict, A. (2023). Smart energy management and load monitoring of individual loads. *E3S Web of Conferences*, 405, 02013. https://doi.org/10.1051/e3sconf/202340502013

6.4 Code:

```
#define BLYNK_TEMPLATE_ID "TMPL3UKpzjiL4"

#define BLYNK_TEMPLATE_NAME "Door Alert"

#define BLYNK_AUTH_TOKEN "XjTfMwIMJPfTChbM6TvTNeK18SF1AkHI"

#define BLYNK_PRINT Serial

#include <WiFi.h>

#include <BlynkSimpleEsp32.h>

#include <ESP_Mail_Client.h>

#include <time.h>


// Wi-Fi Credentials

#define WIFI_SSID "OnePlus Nord"

#define WIFI_PASSWORD "100ni12345"

// SMTP Server Details (For Gmail)

#define SMTP_HOST "smtp.gmail.com"

#define SMTP_PORT 465  // Use 465 for SSL, or 587 for TLS


// Sender Email Details

#define AUTHOR_EMAIL "espt418@gmail.com"

#define AUTHOR_PASSWORD "yowk xeol funq auki"


// Recipient Emails (Multiple)

#define RECIPIENT1 "mansimsoni3110@gmail.com"
```

```
#define RECIPIENT2 "ishad2281@gmail.com"

//#define RECIPIENT3 "shailendra310677zz@gmail.com"


// Sensor & Output Pins

#define Reed_Switch  23  // Door Sensor Pin

#define MQ6_PIN 16       // MQ-6 Digital Output Pin

#define LED_PIN 17       // LED for Gas Alert


const int D3 = 18;  // MSB

const int D2 = 19;

const int D1 = 21;

const int D0 = 22;  // LSB

const int Pins[4] = {D3, D2, D1, D0};

const int trigPin = 5;    // HC-SR04 Trig pin

const int echoPin = 4;    // HC-SR04 Echo pin

int level = 11;

long lastChangeTime = 0;

bool isFullOn = false;

char auth[] = BLYNK_AUTH_TOKEN;

BlynkTimer timer;

SMTPSession smtp;

bool emailSent = false;  // Prevent multiple emails for same gas event

void setBrightness(int level) {
```

```
  digitalWrite(Pins[3], (level & 0b1000) >> 3);
  digitalWrite(Pins[2], (level & 0b0100) >> 2);
  digitalWrite(Pins[1], (level & 0b0010) >> 1);
  digitalWrite(Pins[0], (level & 0b0001));
}

long measureDistance() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
 digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  long duration = pulseIn(echoPin, HIGH);
  long distanceCm = duration * 0.034 / 2;
return distanceCm;
}
// ✅ Sync Time with NTP Server (Required for ESP-Mail-Client)
void setTimeUsingNTP() {
    configTime(0, 0, "pool.ntp.org", "time.nist.gov");
    Serial.print("Syncing time");
    time_t now = time(nullptr);
    while (now < 8 * 3600 * 2) {
        Serial.print(".");
```

```
        delay(500);
        now = time(nullptr);
    }
    Serial.println("\nTime synchronized!");
}


// ✅ Connect to Wi-Fi
void connectWiFi() {
    Serial.print("Connecting to Wi-Fi");
    WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
    while (WiFi.status() != WL_CONNECTED) {
        Serial.print(".");
        delay(1000);
    }
    Serial.println("\nConnected to Wi-Fi!");
}
// ✅ Send Email Alert for Gas Leak
void sendEmailAlert() {
    smtp.debug(1);
    ESP_Mail_Session session;
    session.server.host_name = SMTP_HOST;
    session.server.port = SMTP_PORT;
    session.login.email = AUTHOR_EMAIL;
```

```
  session.login.password = AUTHOR_PASSWORD;
  session.login.user_domain = "";
SMTP_Message message;
  message.sender.name = "ESP32 Gas Detector";
  message.sender.email = AUTHOR_EMAIL;
  message.subject = "⚠️ GAS LEAK ALERT!";
  message.text.content = "Warning! Gas leak detected. Take immediate action.";

  // Add Multiple Recipients
  message.addRecipient("Receiver 1", RECIPIENT1);
  message.addRecipient("Receiver 2", RECIPIENT2);
  //message.addRecipient("Receiver 3", RECIPIENT3);

  if (!smtp.connect(&session)) {
    Serial.println("❌ Failed to connect to mail server");
    return;
  }
if (MailClient.sendMail(&smtp, &message)) {
    Serial.println("✅ Email alert sent successfully!");
  } else {
    Serial.println("❌ Failed to send email");
  }
```

```
}
// ✅ Door Alert Function (Blynk Notification)
void notifyOnTheft() {
    int isTheftAlert = digitalRead(Reed_Switch);
    Serial.println(isTheftAlert);
    if (isTheftAlert == 1) {
        Serial.println("🚨 Door Open Alert!");
        Blynk.logEvent("door_Alert", "Your door is open, Please close the door");
 }
}
// ✅ Gas Detection Function
void checkGasLeak() {
    int gasDetected = digitalRead(MQ6_PIN);  // Read MQ-6 Digital Output
 if (gasDetected == LOW) {  // Gas detected (LOW means active)
        Serial.println("⚠️ GAS LEAK DETECTED!");
        digitalWrite(LED_PIN, HIGH);  // Turn ON LED
        if (!emailSent) {
            sendEmailAlert();
            emailSent = true;  // Prevent multiple emails for same event
        }
    } else {
        Serial.println("✅ No Gas Detected.");
```

```
        digitalWrite(LED_PIN, LOW);  // Turn OFF LED
        emailSent = false;  // Reset email trigger when gas is gone
    }
}
void setup() {
    Serial.begin(115200);
 for (int i = 0; i < 4; i++) {
    pinMode(Pins[i], OUTPUT);
    //digitalWrite(Pins[i], LOW);
  }
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
    pinMode(Reed_Switch, INPUT);
    pinMode(MQ6_PIN, INPUT);
    pinMode(LED_PIN, OUTPUT);
    delay(100);
    for (int i = 0; i < 3; i++) {
    setBrightness(level);
    delay(100); // Delay to ensure dimmer sees it
  }
    connectWiFi();        // Connect to Wi-Fi
    setTimeUsingNTP();    // Sync time using NTP
    Blynk.begin(auth, WIFI_SSID, WIFI_PASSWORD);
```

```
    Serial.println("[illegible] Door & Gas Monitoring System Ready...");
  timer.setInterval(5000L, notifyOnTheft);  // Check Door every 5 seconds
    timer.setInterval(2000L, checkGasLeak);   // Check Gas every 2 seconds
}

void loop() {
  long distance = measureDistance();
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");
  unsigned long currentTime = millis();
 if ((distance < 38 || distance > 40) && !isFullOn) {
    level = 0;  // Fully ON
    setBrightness(level);
    isFullOn = true;
    Serial.println("Distance changed -> Light FULL ON");
  }
  // If distance is back to 39 cm, and it's currently full on
  if ((distance >= 38 && distance <= 40) && isFullOn) {
    if (currentTime - lastChangeTime > 1000) {
      delay(2000);
      level = 11;  // Back to dim
      setBrightness(level);
```

```
      isFullOn = false;
      Serial.println("Back to 38 cm -> Light DIM again");
    }
  } else {
    lastChangeTime = currentTime;
  }


  delay(200);  // Slow down loop for stability
    Blynk.run();
    timer.run();
}
```