Jurnal Informasi dan Teknologi

https://jidt.org/jidt

2025 Vol. 7 No. 1 Page: 27-34 e-ISSN: 2714-9730

IoT-Enabled Smart Home Energy Monitoring System Using Web Server-Based Control Logic

Tubagus Luthfi Almas'ud[1✉], Dendy K Pramudito[2], Aceng Badruzzaman[3]

[123]Universitas Pelita bangsa, Indonesia

jurnal.jidt@gmail.com

Abstract

In this era of development, applications for Controlling and monitoring household devices remotely are increasingly becoming a significant concern. In this context, this research aims to implement an IoT-based Smarthome Control and Monitoring System using ESP32 as a sensor node and WEB SERVER. The ESP32 was chosen for its strong wifi capabilities and sufficient Processing capabilities. The proposed system allows users to control and monitor home appliances through an intuitive web interface accessed from any device connected to the Internet. The research methodology includes the system design, implementation, and testing stages. System design involves selecting hardware and software components that suit system requirements. Implementation is carried out by building ESP32-based hardware and developing a WEB SERVER to communicate with the device. Tests were conducted to verify system functionality and measure Controlling and Monitoring Smarthome performance. Test results show that the proposed system can Control home equipment well and accurately monitor device conditions. Integration with a WEB SERVER allows users to access the system remotely and provides ease of use. In addition, using ESP32 as a sensor node provides flexibility in developing IoT systems. This research contributes to developing IoT applications in the context of Smart Homes and energy management. The developed system can help improve energy efficiency and provide greater comfort for users in Controlling household devices. Apart from that, this research also opens up opportunities for further development in IoT and related technologies.

Keywords: *Internet of Things, ESP32, Smarthome, Web Server.*

1. Introduction

Smart home technology is a solution to support renewable energy and overcome air pollution. This system utilizes solar panels, inverters, and electric current controllers with wifi connectivity to monitor power production and consumption. With sensor integration, this system optimizes energy efficiency and environmental sustainability [1]. This research develops an IoT-based Smart Park system with NodeMCU ESP8266 to overcome resource limitations and inefficient park monitoring. This system allows real-time and remote monitoring and control of park conditions, including soil moisture, air, temperature, and watering [2]. The Internet of Things, as is bright rooms, is vital in community life. The Internet of Things, often abbreviated as IoT, can connect various electronic devices that can be controlled indoors and outdoors in a way that allows them to be controlled from anywhere using the Internet [3]. Internet of Things (IoT) is a structure in which people are provided with exclusive identities and the ability to move data over a network without requiring two-way human-to-human, i.e., source-to-destination or human-to-computer interaction [4]—The Influence of Sensors on the Internet of Things (IoT). Sensors differentiate IoT from other sophisticated machines—the Influence of Connectivity on the Internet of Things (IoT). The Internet of Things (IoT) is a concept that aims to expand the benefits of internet connectivity that is continuously connected [5]. Cyber Attack Threats: IoT devices in Smart Homes are vulnerable to attacks from outside parties, such as Denial of Service (DoS) attacks, malware, and hacking.User Privacy: With so many devices interconnected [6]. This study develops a PROTOTYPE of a Smart Home control system based on the Internet of Things (IoT) using the MQTT (Message Queuing Telemetry Transport) method. This system integrates Google Assistant as a publisher, IFTTT as a subscriber, and NodeMcu esp8266 as an MQTT broker. Testing is carried out to measure the system's effectiveness in controlling devices such as lights and fans through voice commands [7]. PROTOTYPE development uses Raspberry Pi 3 as a server, OpenHAB as an automation platform, and Telegram as a communication medium. This system utilizes a DHT-11 sensor for temperature and humidity and a magnetic sensor to detect door conditions [8]. PROTOTYPE development uses NodeMCU as a Microcontroller, Blynk application for device control, and sensors such as flame sensor, MQ-6, PZEM 004T, REED Switch sensor, and DHT 11 [9]. PROTOTYPE development uses Raspberry Pi 3 Model B as the control center, a PIR sensor for motion detection, temperature, and humidity

Receipt: 20-01-2025 | Revision: 15-02-2025 | Publish: 27-04-2025 | doi: 10.60083/jidt.vi0.611

sensors, and a camera for monitoring. The system is controlled through a browser-based web application [10]. System development using Arduino Uno as a Microcontroller, PIR sensor for motion detection, door, and window sensors, and GSM module for notification sending. The system is equipped with an alarm as an intrusion indicator [11]. Although smart home technology and Internet of Things (IoT)-based systems have been widely developed, there are still several challenges that have not been fully resolved, such as limited resources in public areas such as city parks, lack of efficient monitoring systems, and security and privacy issues on interconnected IoT devices. In addition, the integration between various sensors and control platforms is often still separate and not optimal. Therefore, a system that can integrate multiple devices and sensors in one platform that is efficient, safe, and easily accessible in real-time is needed to support energy efficiency and environmental sustainability [12]. How to design and develop a prototype of an Internet of Things (IoT)-based smart home system capable of integrating various sensors and control platforms to support energy efficiency, real-time environmental monitoring, and ease of remote control [13].

2. Research Methods

explains the research method to develop a smart home system based on the Internet of Things (IoT). The stages carried out to build the system are the Waterfall Model using the System Development Life Cycle (SDLC) method presented in the Explanation covering requirements gathering and analysis, design, implementation, integration and testing, verification, operation & maintenance.

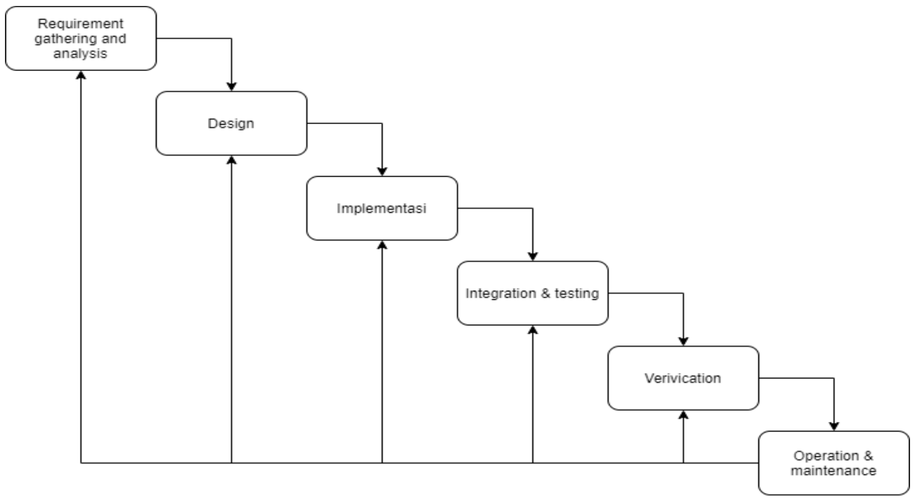

Figure 1. Waterfall

2.1. Requirements Gathering and Analysis

Literature Study: Review previous research related to IoT technology, smart home systems, communication protocols, and relevant devices
Needs Analysis: Identify user needs for a smart home system, including required features, devices used, and user scenarios.

2.2. Design

The design of this smart home system is based on the Internet of Things (IoT) framework, aiming to provide an intelligent, efficient, and remotely accessible home environment. The system integrates various sensors—such as temperature, motion, and light—with a microcontroller connected to a wifi network, enabling real-time data transmission to and from a web server. Users can monitor environmental conditions and control electronic devices like lights and fans through a web-based dashboard. This architecture supports automation, high responsiveness, and scalability for future development.

a. Control Circuit Design
   The control circuit in a smart home system functions as a link between the microcontroller and the controlled household electrical devices. This circuit generally consists of a relay or transistor module as an electronic switch activated based on commands from a web server or programmed automation logic. With a design that considers electrical safety and operational stability, this control circuit allows the system to activate or deactivate devices such as lights, fans, and other electronic equipment efficiently and responsively.

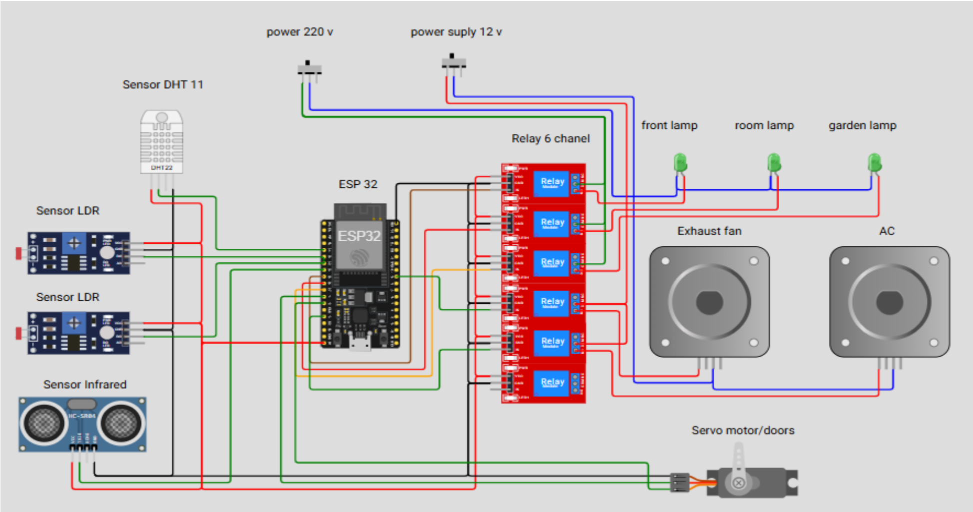

Figure 2. Circuit diagram

b. WorkFlows

"The workflow on an IoT-based smart home system includes a series of integrated processes, starting from data acquisition by sensors, data processing by a microcontroller, to sending data to the server to be displayed via the user interface. This system also receives user input via a web-based dashboard, which is then translated into control commands for actuator devices such as lights, fans, servomotors, air conditioners, and exhaust fans. This workflow is designed to support real-time two-way communication, ensuring high responsiveness, energy efficiency, and ease of remote control.

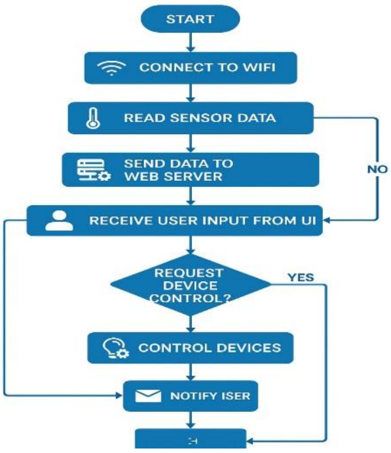

Figure 3. Workflow

2.3. Implementation Hardwere

The hardware implementation on the smart home system prototype is done by designing and assembling a circuit consisting of a microcontroller (ESP32), various sensors (such as temperature, humidity, and motion sensors), and actuators in the form of relay modules to control electronic devices. All components are systematically connected on a breadboard or PCB as an initial testing stage. This hardware is integrated with a wifi network to support data communication between the system and the server, facilitating real-time monitoring and control of devices via a web interface. This implementation is the basis for testing the functionality and reliability of the system before moving on to further development stages. The algorithm design must be written using Lucida Console 8 pt font and the title in 9 pt. Each algorithm display is given a title identity in the first line. Symbols

Table 1. Supporting Hardware Table

part	Amount	Function
Esp 32	1	Microcontroller
Relay	6	Conduct electric
Power Supply	1	Voltage source 12

Lamp	3	illuminator
Fan	3	cooling
Servo motor	1	Open the door
Sensor Dht 11	1	Temperature detection
Sensor Infrared	1	Object detection
Sensor LDR	2	Light detection

Figure 4. Implementation Hardware

2.4. Implementation Software

The software implementation on this smart home system involves the development of a microcontroller program to read sensor data, manage control logic, and send or receive data via the HTTP communication protocol. In addition, a web-based user interface is developed that is connected to a server to enable real-time monitoring and control of devices. This software is designed to run efficiently with a lightweight data processing system responsive to user input and supports integration with automation and notification systems. The software is a liaison between users, hardware, and communication networks in the smart home system architecture. This design uses Arduino Ide software and C language.

Table 2. Supporting Software and Hardware Table

Software	Function
Arduino Ide	Pemrogaman Esp32
Wifi	Connecting the microcontroller
Google/Web	Access Entry
Android/Pc	User Application

Algorithm 1. Program

```
#include <WiFi.h>
#include <DHTesp.h>
#include <ESP32Servo.h>
// Replace with your network credentials
const char* ssid = "A";
const char* password = "12345678";
// Set web server port number to 80
WiFiServer server(80);
// Variable to store the HTTP request
String header; void setup() {
Serial.begin(115200);
// Inisialisasi sensor DHT
dht.setup(DHT_PIN, DHTesp::DHT22);
// Inisialisasi pin
pinMode(LDR_DEPAN, INPUT);
pinMode(LDR_TAMAN, INPUT);
pinMode(IR_SENSOR, INPUT);
pinMode(LAMPU_DEPAN, OUTPUT);
pinMode(LAMPU_RUANG, OUTPUT);
pinMode(AC_RUANG, OUTPUT);
  pinMode(EXHAUST, OUTPUT);
  pinMode(LAMPU_TAMAN, OUTPUT);
  pinMode(PINTU, OUTPUT);
```

Figure 5. web view

2.5. Verification

The verification process is carried out to ensure that all components of the smart home system have functioned according to the specified functional requirements. Testing is done by observing the system's response to input from sensors and control commands given via the web interface. The verification results show that the system can read sensor data in real time, send data to the server with high accuracy, and activate or deactivate electronic devices with a fast and stable response. Thus, the hardware and software implementation on the smart home prototype can be stated to have successfully met the designed specifications.

The first step is to call the IP that is read on the esp32 to find the IP address; the steps we need to do are connecting the NodeMCU port to the laptop by opening the Arduino IDE application by paying attention to the associated PORT, then by selecting the tool and selecting the serial monitor. After finding the IP Address, the next step is to enter the IP Address that is read on the serial monitor into the Android browser page, namely http://192.168.126.175.

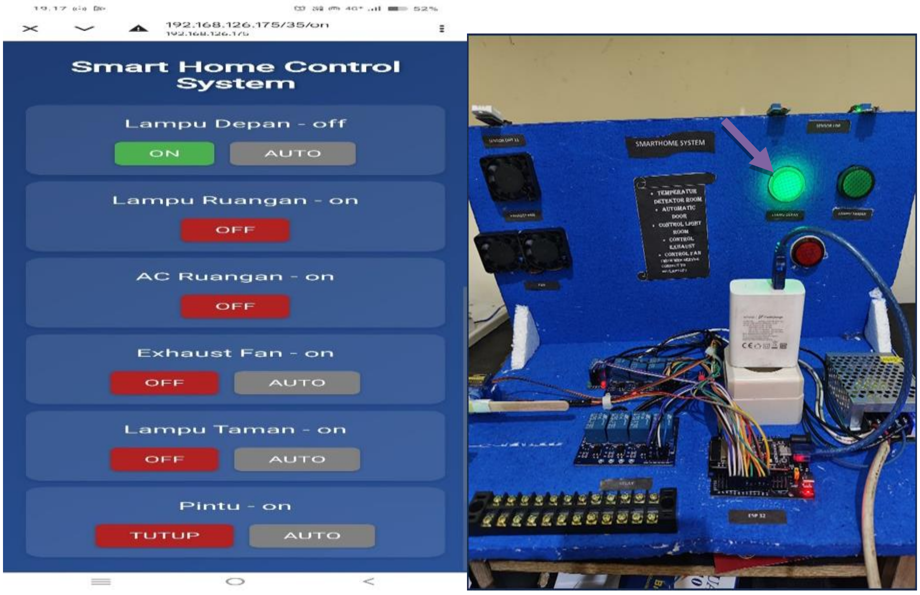

Figure 6. Test & Verification

After connecting to the menu, it will look like the image above, then press ON on button 1 for the headlights to turn on, the headlight status changes from off to on, and the button that was originally green for the on switch changes to red for the off switch, which means the lights are on. The standby switch is on to turn off. When pressed, the auto mode automatically turns on or off according to the light intensity obtained by the LDR sensor, while the relay circuit no. 1 for the headlights does not turn on because it uses a usually close (NC) system.

```
Client disconnected.

GET /35/auto HTTP/1.1
Host: 192.168.115.175
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Linux; Android 10; K) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/135.0.0.0 Mobile Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://192.168.115.175/34/auto
Accept-Encoding: gzip, deflate
Accept-Language: id-ID,id;q=0.9,en-US;q=0.8,en;q=0.7
```

Figure 7. Database

Table 3. Result Verification

Equipment	Status	Result Information
Headlamps	Auto/On/Off	Headlights on when there is light/manual on/off
Taillight	ON/Off	Manual on/off
GardenLights	Auto/On/Off	Garden lights on when there is light/manual on/off
AC	ON/Off	Manual on/of
Exhaust Fan	Auto/On/Off	exhaust fan on when the temperature reaches 36'/manual on/off
Doors	Auto/On/Off	pintu terbuka ketika ada object dan menutup sendiri setelah 5 detik tidak ada objek/manual on/off

2.6. Operation & Maintenance

This IoT-based Smart Home system is designed to be easy to operate by users with the following features:

Figure 8. Operation and maintenance

3. Results and Discussion

3.1. Device Control via Web Server

One of the essential features of the developed system is the ability to control home devices remotely via a browser-based web server interface. This system uses a NodeMCU ESP32 programmed to host a simple web page that can be accessed via a local or public IP address (with port forwarding).

Test Results:

a. The web page displays the real-time status of devices such as lights, fans, and security systems (on/off).
b. Users can turn devices on or off by pressing the control button directly from the web page.
c. The average response time of the device to commands from the web is less than 2 seconds in a local network (LAN) and around 3–5 seconds for access from outside the network (WAN).

3.2. Temperature and Humidity Monitoring

The DHT-11 sensor can measure room temperature and humidity in real-time and send data to a web server. The data displayed has a time lag of less than 2 seconds from the measurement time to activate the exhaust fab.

3.3. Light Automation with LDR (Light Dependent Resistor)

LDR sensor is used to detect light intensity in the surrounding environment. The system is designed so that the lights automatically turn on when the ambient light is low (at night) and turn off when it is bright (during the day).

a. The light threshold is set at a specific value (e.g., <200 lux).
b. The system response time to turn the lights on or off is <1 second.
c. Tests show that detection accuracy reaches 95% under normal lighting conditions.

3.4. Automatic Door Control with Infrared (IR) Sensor

An infrared sensor is used to detect the presence of objects or movement in front of the door. When an object is detected, the system automatically activates the servo motor to open the door.

a. The effective detection distance of the IR sensor is 0–2 cm.
b. The servo motor opens the door while there is still an object; if there is no object, the door will close automatically after 5 seconds.
c. The system can also be added with a manual override button for security or emergency conditions.

4. Conclusion

This research has successfully developed a Smart Home system based on the Internet of Things (IoT) that integrates various sensors and automatic and manual control methods to improve residents' comfort, energy efficiency, and safety. This system uses NodeMCU ESP8266 as the main microcontroller. It is supported by sensors such as LDR for lighting automation, infrared sensors for automatic door control, and a web server platform for real-time and remote device control.

References

[1] S. Fuady, U. Khaira, Y. R. Hais, and R. H. Sitanggang, "Penerapan Model Terdistribusi untuk Sistem Smarthome Menggunakan Multi-Sensor Berbasis Internet of Things (IoT)," *J. Electr. Power Control Autom.*, vol. 4, no. 2, p. 68, 2021, doi: 10.33087/jepca.v4i2.57.

[2] I. W. B. Darmawan, I. N. S. Kumara, and D. C. Khrisne, "Smart Garden Sebagai Implementasi Sistem Kontrol Dan Monitoring Tanaman Berbasis Teknologi Cerdas," *J. SPEKTRUM*, vol. 8, no. 4, p. 161, 2022, doi: 10.24843/spektrum.2021.v08.i04.p19.

[3] S. Sumartono and K. Khoeri, "Rancang Bangun Sistem Smart Room Dengan Kombinasi Sensor Photodioda Dan Sensor Pir Sebagai Upaya Penghematan Energi Listrik Dan Monitoring Ruangan," *Media Elektr.*, vol. 11, no. 1, p. 20, 2019, doi: 10.26714/me.11.1.2018.20-26.

[4] A. Prihanto, N. Rachmawati, and A. Prapanca, "Smart Garden Automation Dengan Memanfaatkan Teknologi Berbasis Internet Of Things (IoT)," *J. Inf. Eng. Educ. Technol.*, vol. 5, no. 2, pp. 55–60, 2021, doi: 10.26740/jieet.v5n2.p55-60.

[5] B. F. Alwaaritsi, T. Tohir, and D. Aming, "Rancang Bangun Sistem Kontrol dan Pemantauan Serikultur Menggunakan Mikrokontroler Berbasis Internet-of-Things," *Pros. Ind. Res. Work. Natl. Semin.*, vol. 12, pp. 41–46, 2021, doi: 10.35313/irwns.v12i0.2654.

[6] Hasri Awal, "Perancangan Prototype Smart Home Dengan Konsep Internet of Thing (IoT) Berbasis Web Server," *Maj. Ilm. UPI YPTK*, no. 26, pp. 65–79, 2019, doi: 10.35134/jmi.v26i2.53.

[7] V. Simadiputra and N. Surantha, "Rasefiberry: Secure and efficient raspberry-pi based gateway for smarthome iot architecture," *Bull. Electr. Eng. Informatics*, vol. 10, no. 2, pp. 1035–1045, 2021, doi: 10.11591/eei.v10i2.2741.

[8] D. A. Zubov, M. S. Qureshi, U. Köse, and A. I. Kupin, "Prototyping Smart Home for Immobilized People: Eeg/Mqtt-Based Brain-To-Thing Communication," *Radio Electron. Comput. Sci. Control*, no. 2, p. 90, 2022, doi: 10.15588/1607-3274-2022-2-9.

[9] Y. Hermanto, "Prototype Monitoring Electricity System 220V of Wind Power Plant (Pltb) Based on the Internet of Things," *Internet Things Artif. Intell. J.*, vol. 1, no. 3, pp. 133–145, 2021, doi: 10.31763/iota.v1i3.469.

[10] S. I. Pella and H. F. L, "Disain Sistem Otomasi Suhu Ruangan Pertemuan Dengan Penerapan Teknik Machine Learning," *J. Media Elektro*, vol. VIII, no. 2, pp. 117–120, 2019, doi: 10.35508/jme.v0i0.1781.

[11] I. W. K. Bima, V. Suryani, and A. A. Wardana, "A Performance Analysis of General Packet Radio Service (GPRS) and Narrowband Internet of Things (NB-IoT) in Indonesia," *Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, vol. 4, pp. 11–20, 2020, doi: 10.22219/kinetik.v5i1.947.

[12] A. Nababan and T. K. WIjaya, "Perancangan Kontrol Alat-Alat Elektronik Rumah Berbasis Iot Menggunakan Voice Command," *Sigma Tek.*, vol. 4, no. 2, pp. 277–284, 2021, doi: 10.33373/sigmateknika.v4i2.3621.

[13] A. F. Isnanto, A. Surriani, S. Lestari, and U. Y. Oktiawati, "Prototype of Smart Home and Monitoring Application Based On Internet of Things (IoT) Using Android," *J. List. Instrumentasi dan Elektron. Terap.*,

vol. 1, no. 1, pp. 6–9, 2021, doi: 10.22146/juliet.v1i1.53939.

[14] M. Kasendra, D. K. Pramudito, and D. Afandi, "The Soil Moisture Monitoring Utilize IoT System To Improve Urban Farming Productivity and Enhance Food Security ", *jidt*, pp. 1-12, Apr. 2025.

[15] D. K. Pramudito, S. Arijanti, A. Yanto RUkmana, D. S. Oetomo, and K. Kraugusteeliana, "The Implementation of End User Computing Satisfaction and Delone & Mclean Model to Analyze User Satisfaction of M.TIX Application", *jidt*, vol. 5, no. 3, pp. 7-12, Aug. 2023.

[16] D. K. Pramudito, A. Nuryana, S. Assery, H. Purnomo, and A. Ady Bakri, "Application of Unified Theory of Acceptance, Use of Technology Model and Delone & Mclean Success Model to Analyze Use Behavior in Mobile Commerce Applications", *jidt*, vol. 5, no. 3, pp. 1-6, Aug. 2023.

[17] N. Somu, and N. S. Dasappa, "An edge-cloud IIoT framework for predictive maintenance in manufacturing systems", *Advanced Engineering Informatics*, Vol. 65, Part D, 2025, https://doi.org/10.1016/j.aei.2025.103388.

[18] S. K. Vishwakarma, P. Upadhyaya, B. Kumari and A. K. Mishra, "Smart Energy Efficient Home Automation System Using IoT," *2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)*, Ghaziabad, India, 2019, pp. 1-4, doi: 10.1109/IoT-SIU.2019.8777607

[19] P. M R and B. Bhowmik, "Development of IoT-Based Smart Home Application with Energy Management," *2023 International Conference on Sustainable Communication Networks and Application (ICSCNA)*, Theni, India, 2023, pp. 367-373, doi: 10.1109/ICSCNA58489.2023.10370276.

[20] M. Kumar, R. K. Gupta, G. Jain and P. sharma, "Next-Generation Smart Homes: An IoT Approach to Home Automation and User Experience," *2024 International Conference on Artificial Intelligence and Quantum Computation-Based Sensor Application (ICAIQSA)*, Nagpur, India, 2024, pp. 1-6, doi: 10.1109/ICAIQSA64000.2024.10882400.