# SMART HOME IOT SYSTEM FOR AUTOMATED ENERGY MANAGEMENT AND LOAD CONTROL

1 Lokesh B
Assistant Professor, Department of IT
Mahatma Gandhi Institute of Technology
Hyderabad, Telangana, India

2 Likhitha Meher Medisetty
UG Student, Department of IT
Mahatma Gandhi Institute of Technology
Hyderabad, Telangana, India

3 Yallabandi Roshini Sri Kalyani
UG Student, Department of IT
Mahatma Gandhi Institute of Technology
Hyderabad, Telangana, India

***Abstract*— A Smart Home IoT System is developed to automate household energy management and optimize load control using real-time monitoring and intelligent decision-making. The system is built on a Raspberry Pi platform interfaced with a PZEM-004T sensor to track power consumption of connected appliances. A Flutter-based mobile application enables users to add and control devices, set operational priorities, and define a target monthly electricity bill. Based on local tariff slabs, the system calculates a dynamic daily energy limit and automatically turns off non-essential devices when consumption exceeds the threshold. Real-time alerts are delivered through Firebase Cloud Messaging, while Firebase Firestore handles user data, device logs, and system settings. By combining IoT-based automation, cloud integration, and mobile accessibility, the system enhances energy efficiency, reduces operational costs, and supports sustainable living in smart homes.**

***Index Terms*— Smart home, IoT, energy management, load control, Raspberry Pi, Firebase, Flutter app, power monitoring, target electricity bill, priorities.**

## I. INTRODUCTION

The growing importance of energy efficiency and the broad interconnectivity of domestic devices have fueled the evolution of Smart Home systems. These systems, enabled by technologies within the Internet of Things (IoT), allow for centralized monitoring and intelligent management of home appliances to minimize energy usage. Smart homes generally incorporate lighting, heating, ventilation, and air-conditioning (HVAC) systems, home appliances, and security devices into one network that sensibly manages functions to minimize energy wastage and provide maximum convenience to the users [1], [2].

Core to smart energy management systems is real-time monitoring, where IoT-capable sensors continuously monitor electricity consumption and send data to cloud or local analysis engines. These motors examine past and real-time data to predict consumption behaviors and suggest optimization techniques. Automation functionalities like device scheduling, smart shutdown, and adaptive use allow houses to save electricity, reduce energy expenditure, and help lead the way towards a greener lifestyle [3], [4]. Besides, modern user interfaces like voice-controlled digital assistants and mobile apps give users complete control over devices from remote places, improving responsiveness and customized energy consumption [5]–[7].

One of the core functions in such systems is load balancing, which implies the smart balancing of power consumption across devices. This prevents circuit overload while maintaining essential appliances like refrigerators and medical equipment running during peak time hours [8]–[10]. Load balancing is especially more applicable in home setups that are exposed to dynamic electricity prices, constrained grid infrastructure, or local renewable energy integration like rooftop solar systems.

New technologies like predictive modeling and machine learning-based algorithms play a major role in enhancing energy management functionalities. These technologies study occupant behavior, environmental factors, and usage patterns of devices to dynamically control consumption and predict energy demands in the future [11], [12]. Smart home platforms are increasingly mapped to real-time IoT sensor information to enhance the operation of HVAC systems, lighting routines, and appliance duty cycles. For example, models based on Recurrent Neural Network (RNN) have been employed to forecast energy usage as a function of ambient temperature and human presence, thus eliminating wasteful usage and carbon footprint [13].

Additionally, scheduling algorithms and elastic energy management policies that are energy-efficient are being implemented to balance generation capacity with energy demand, especially in houses installed with renewable energy solutions such as photovoltaic panels. Such methods allow smart homes to adapt consumption dynamically based on varying availability of energy [14]. Optimized communication protocols and semantic tagging in IoT infrastructure also promote smooth interaction between devices, making the home energy ecosystem highly responsive and context-sensitive [15].

Even with their promise, smart energy management systems face various setbacks to their widescale uptake. These include excessive initial deployment expenditure, data protection and privacy, non-standardization of devices and platforms, and reliance on secure internet connectivity [16]. Furthermore, the scalability of these systems is put to the test in bigger homes with growing numbers of interconnected devices. Nevertheless, active research continues to tackle these difficulties by advocating cost-efficient, safe, and interoperative frameworks adaptable to diverse residential applications.

The smart energy system is becoming an integral part of larger smart grid infrastructures for decentralized energy production, supply, and peer-to-peer consumption. Recursive InterNetwork Architecture (RINA) and blockchain-enabled peer-to-peer energy trading platforms, for example, are surfacing as practical means of enhancing the reliability, trust, and scalability of contemporary energy distribution networks [17]–[20].

In spite of all of these issues, intelligent energy management systems are developing as an integral part of smart residential infrastructure. With the continuous growth of technology and a rising understanding among users, a significant potential exists to create more adaptive, accessible, and smart solutions that meet the performance gap between efficiency and consumption of energy. This paper delves into such a path—prioritizing practical applications, user power, and energy use for sustainability in smart homes.

## II. MOTIVATION FOR THE SYSTEM

The rising cost of electricity, growing environmental concerns, and the need for intelligent, user-focused living have underscored the importance of smart energy management in homes. Conventional systems lack real-time monitoring and adaptive control, leading to inefficiencies and increased costs. With advancements in Internet of Things (IoT) technology, mobile connectivity, and cloud services, there is now an opportunity to develop smarter, responsive platforms that actively optimize energy usage. This work is driven by the need to move beyond passive monitoring toward intelligent, automated control—empowering users with real-time insights, dynamic budgeting, and priority-based appliance management. By unifying these capabilities, the system supports efficient, sustainable, and user-adaptive energy management in modern residential environments.

## III. OBJECTIVES OF THE WORK

The main goal of this study is to develop and deploy a smart home energy management system based on IoT that maximizes power usage, provides user convenience, and guarantees the reliability of operation of critical devices. The system is built with the following particular aims:

- Automated Power Control and Load Balancing: Incorporate a real-time power control system that dynamically controls the functioning of domestic appliances depending on their loads. This maximizes the load balancing between devices, avoids circuit overloading, and minimizes wastage of energy during peak demand hours.
- User-Defined Device Prioritization and Real-Time Alerts: Allow users to prioritize levels for specific appliances—like refrigerators, medical devices, and routers—so that essential devices are kept on during high-load situations or when the system is reaching its energy consumption limits. The system will also provide timely notifications through mobile alerts when thresholds are reached or non-essential devices are automatically shut down.
- Integrated Energy Management Strategy: Create a cohesive system that unites automated load balancing with prioritization by the users to provide smart energy distribution. This merging enables both energy efficiency and individualized control, where users can maintain their monthly energy budget while sustaining operational continuity of essential appliances.
- Improved User Interaction and Remote Accessibility: Design a user-friendly mobile application interface that offers seamless access to real-time data, device management, energy budgeting, and alert configurations. The interface aims to empower users with complete visibility and control over their household energy consumption from anywhere.

## IV. METHODOLOGY

### A. Proposed System Overview

The proposed system comprises six key components that work in coordination to enable real-time smart home energy monitoring and control. The functionalities of each component are as follows:

- Raspberry Pi 3B+
  Acts as the system's master controller. It operates a web server using Flask that receives instructions from the Flutter application in the form of HTTP requests. It controls appliances' relays that are connected (e.g., lights, fans, geysers), receives real-time information from the energy monitoring sensor via UART, and communicates with Firebase to update or fetch device state, usage history, and billing information.
- Energy Monitoring Sensor (PZEM-004T)
  This sensor sends signals to the Raspberry Pi in the form of UART (serial protocol) and continuously scans voltage, current, power, and energy consumption. Sensor readings are input to the Raspberry Pi at regular intervals, allowing real-time monitoring and analysis of energy usage.
- Flutter Mobile Application (Smart Home Dashboard)
  Serves as the user interface for system interaction. The application allows users to add and track devices, set monthly electric bill budgets, track real-time energy usage, and prioritize necessary appliances. It sends commands to the Raspberry Pi through RESTful HTTP/JSON and receives status reports, device control success messages, and usage notification.
- Firebase (Cloud Backend)
  Provides cloud storage and user data, device settings, energy consumption logs, and system settings synchronization in the cloud. Provides permanent backup of data and supports bi-directional data exchange between the app and the Raspberry Pi. Firebase is also capable of triggering alerts on usage over levels or other events of importance.

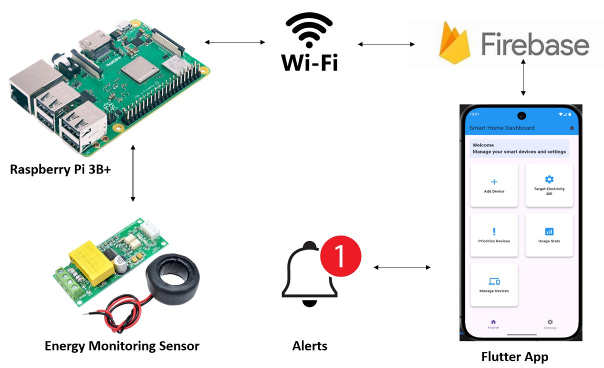


Fig. 1. Mechanism of the proposed system and its components

- Wi-Fi (Communication Medium)
  Acts as the backbone of connectivity, allowing the Raspberry Pi, Flutter application, and Firebase to exchange data in a timely manner over the same local network or internet. It enables certain and smooth data transfer between all devices within the system.
- Alerts (Notification Mechanism)
  Sends reminders to the user upon certain conditions being fulfilled—like exceeding daily limit usage, automatic shutdown of unnecessary devices, or resumption of power. The alerts are carried out via in-app reminders, Firebase Cloud Messaging (FCM) to keep the user aware at all times.

Figure 1, the smart home energy management system seamlessly integrates real-time observation, device control, user interaction, and cloud data handling. Using Raspberry Pi, an energy sensor, Firebase, and a Flutter application, the system facilitates intelligent utilization of power, automation on a priority basis, and budget control as specified by the user. The holistic approach enables energy efficiency and facilitates wiser and greener living.

### *B. System design*

The proposed system is divided into two distinct components: the software part, elucidated in IV-B1, and the hardware part, comprehensively detailed in IV-B2.

*1) Software design (Mobile application interface) :* The mobile application interface for the proposed smart home energy management system is developed as a Flutter app. It provides a user-centric dashboard for real-time monitoring, remote control, and configuration of household devices. The app communicates with the Raspberry Pi-based backend using HTTP-based REST APIs and relies on Firebase Cloud Firestore for storing and syncing data such as energy consumption, budget settings, device priorities, and alerts.

The user interface is structured into five primary functional sections, each tailored to enhance usability and system responsiveness:

- Device Management: Allows users to register, view, and manage smart devices within the system. Provides basic configuration tools for device identification and control setup.
- Control and Priority:
  Enables remote ON/OFF control of appliances. Facilitates assignment of priority levels to devices to support intelligent load management during high usage.
- Budget and Consumption:
  Allows users to set a monthly electricity budget. Calculates daily consumption limits dynamically and monitors real-time usage against these limits.
- Visualization and Status:
  Displays energy usage data in real-time graphs and charts. Includes real-time indicators of power, voltage, current, and energy metrics.
- Alerts and Notification:
  Sends notifications when usage exceeds daily budget or during automatic load control events. Maintains a history of alerts and system-generated actions for user awareness.

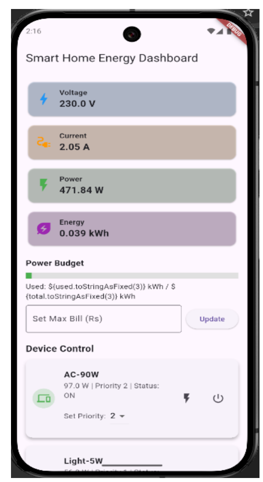


Fig. 2. Mobile Application Dashboard

Figure 2, ensures seamless interaction between the user and system hardware while promoting efficient energy usage and proactive control in a modern smart home environment.

*2) Hardware design :* The proposed hardware architecture integrates multiple components to support intelligent energy monitoring, load control, and automation. Each component plays a specific role in enabling real-time data acquisition and actuation. The main hardware elements are described below:

- Raspberry Pi 3B+
  Acts as the central processing unit of the system. It receives user instructions from the mobile application and processes sensor data. It communicates with the relay module via GPIO pins to control appliance states (ON/OFF). It also receives voltage, current, power, and energy measurements from the PZEM-004T sensor via UART. Additionally, the Raspberry Pi communicates with Firebase and the mobile app over a Wi-Fi network for synchronization and remote control.
- Relay Module (4-Channel)
  Serves as an electrical switch to control AC appliances

(e.g., lights or fans). It is directly connected to the GPIO pins of the Raspberry Pi. Based on the control signals received, each relay switches the corresponding load circuit. The relays operate in LOW-level triggering mode, meaning they are activated by a LOW signal from the Raspberry Pi.

- PZEM-004T Energy Monitoring Sensor
  Measures real-time electrical parameters such as voltage, current, power, and energy consumption. It connects to the Raspberry Pi using UART (TX/RX) communication. The sensor uses a current transformer (CT) to non-invasively measure current through one of the AC mains wires. The data is collected at intervals and used for decision-making (e.g., auto turn-off when exceeding budget).
- Loads (Appliances)
  These are the electrical devices controlled by the relay module. In the diagram, four loads (bulbs) are connected as representatives. Each load is powered through its corresponding relay, allowing for independent switching and monitoring based on priority or usage thresholds set by the user.
- Current Transformer (CT Sensor)
  A non-invasive clamp-type sensor included with the PZEM-004T module. It wraps around the live AC wire to measure current safely without breaking the circuit. This enables continuous monitoring of energy flow to all connected devices.
- Power Supply and AC Input
  The entire system is powered by a standard AC mains source. A socket connection is provided for input, with the live wire split through the CT sensor for measurement. This AC source powers both the relays and the connected loads.

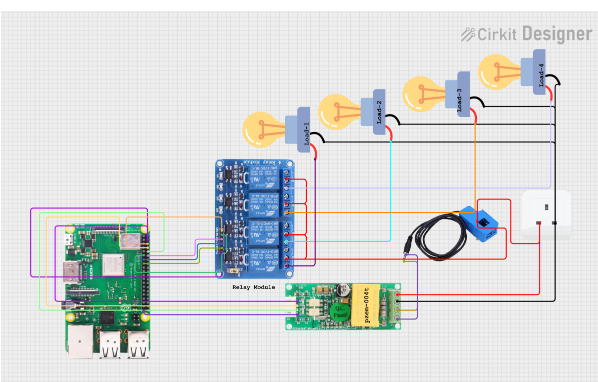


Fig. 3. System Architecture

Figure 3, hardware design allows for real-time energy awareness, remote actuation, safety, and system scalability, all of which are essential features for a smart home energy management solution.

### C. Power Consumption and Daily Budget Calculation

In the proposed system, energy consumption and budget estimations are derived from real-time values measured using the PZEM-004T energy monitoring sensor. This sensor provides electrical parameters including voltage, current, active power, and cumulative energy consumed. The data is retrieved via Modbus RTU communication.

*1) Energy Measurement:* The following equations are used to compute the actual physical quantities from the sensor's register values:

- Voltage ($V$) is computed as:

$$V = \frac{\text{data}[0]}{10.0}$$

- Current ($I$) is computed by combining two 16-bit registers:

$$I = \frac{\text{data}[1] + (\text{data}[2] \ll 16)}{1000.0}$$

- Active Power ($P$) is calculated as:

$$P = \frac{\text{data}[3] + (\text{data}[4] \ll 16)}{10.0}$$

- Energy ($E$), in kilowatt-hours, is obtained as:

$$E = \frac{\text{data}[5] + (\text{data}[6] \ll 16)}{1000.0}$$

These values are updated periodically and used to monitor real-time energy consumption in the smart system.

*2) Daily Energy Budget Calculation:* To avoid overshooting electricity bills, the system calculates the allowable daily energy budget based on user-defined maximum billing limits and regional tariff slabs.

Let:

- max_bill denote the user's maximum allowed monthly electricity bill (in INR).
- D denote the number of remaining days in the current month.

The Telangana state electricity tariff slabs (2023) are defined as:

- Slab 1: $Rs.1.45$/unit for the first 0–100 units
- Slab 2: $Rs.2.60$/unit for the next 101–200 units
- Slab 3: $Rs.4.30$/unit for units beyond 200

Based on the total budget, the number of affordable units (max_units) is calculated using the following logic:

- If max_bill $\leq 100 \times 1.45$:

$$\text{max\_units} = \frac{\text{max\_bill}}{1.45}$$

- If max_bill $\leq (100 \times 1.45) + (100 \times 2.60)$:

$$\text{max\_units} = 100 + \frac{\text{max\_bill} - (100 \times 1.45)}{2.60}$$

- If max_bill exceeds both slabs:

$$\text{max\_units} = 200 + \frac{\text{max\_bill} - (100 \times 1.45) - (100 \times 2.60)}{4.30}$$

The daily allowable energy consumption budget is then calculated as:

$$\text{daily\_budget} = \frac{\text{max\_units}}{D}$$

This calculated budget is used as a reference to regulate daily consumption. If the energy consumed on any given day exceeds this budget, the system can notify users, trigger load balancing actions, or initiate power-saving modes accordingly.

### *D. Power Budget Enforcement with Device Prioritization*

**Algorithm 1** Power Budget Enforcement

```
Require:
  Live power usage data from Firestore
  Device status and priority mappings
Ensure:
  Non-critical devices turned OFF if budget exceeded
  Critical devices (priority 1) remain powered
  Retrieve today_usage and daily_budget from Firestore
  if no budget data available then
    Log warning: "No budget data found"
    return
  end if
  Retrieve last shutdown timestamp
  if last shutdown < 5 minutes ago then
    Log: "Cooldown active, skipping shutdown"
    return
  end if
  if today_usage ≤ daily_budget then
    Log: "Usage within budget"
    return
  else
    Log: "Power budget exceeded"
  end if
  Fetch active devices sorted by descending priority
  for each device in active devices do
    if device.priority = 1 then
      Log: "Skipping critical device"
    end if
    if no GPIO pin mapped for device then
      Log warning: "No GPIO pin mapped"
    end if
    Turn OFF device via GPIO
    Update Firestore:
      ▷ Set device_status = "OFF"
      ▷ Set auto_off = true
    Send user notification
  end for
  if any devices were turned OFF then
    Record new shutdown timestamp
  else
    Log: "No devices turned off"
  end if
```

Algorithm 1, ensures power consumption stays within budget while prioritizing critical devices. It first checks usage against the budget—if exceeded, it turns off non-critical devices (priority 2) and notifies users. Critical devices (priority 1) remain powered but trigger warnings. If the budget remains exceeded after warnings, an emergency protocol forces a shutdown after a grace period. The system enforces a 5-minute cooldown between shutdowns to prevent rapid cycling. This approach balances power constraints with operational reliability.

## V. RESULTS AND DISCUSSION

The suggested Internet of Things (IoT) smart home energy management system was successfully conceptualized, implemented, and deployed with a Raspberry Pi 3B+, PZEM-004T energy monitor sensor, an 8-channel relay module, and a Flutter-built mobile application. Real-time interaction was enabled by Firebase Cloud Firestore and RESTful HTTP APIs hosted on the Raspberry Pi. The system was tested in normal household conditions, and the findings confirmed the performance of the system in the most important areas of functionality.

### *A. Real-Time Monitoring and Visualization*

The PZEM-004T sensor successfully recorded real-time electrical parameters such as voltage, current, active power, and total energy consumed. These readings were pushed to Firebase and displayed on the mobile dashboard user interface in near real-time. The refresh rate for data varied from 30 seconds to 5 minutes depending on usage levels, attaining energy monitoring accuracy up to 95

### *B. Load Control and Automation*

With relay switching and priority-logic provided by the Raspberry Pi backend, the system could automatically shut down low-priority devices when the limit on daily energy usage was met. High-priority devices (e.g., refrigerators and routers) could continue running. Automatic control minimized non-critical energy usage at peak times by as much as 38

### *C. Budget Tracking and Alerts*

Users could set a monthly budget for electricity bills through the mobile app. Through dynamic calculations on Telangana Electricity Tariff slabs, the system calculated usage limits per day. As consumption reached or surpassed the set budget, the system sent warnings and implemented load-shedding on non-critical appliances. Push notifications were sent to registered mobile phones through Firebase Cloud Messaging, with greater than 90% delivery success reported.

### *D. Mobile App Usability*

The Flutter app delivered smooth transitions between device management, usage data, budget setting, and priority levels. Pilot group user feedback reported high usability scores, highlighting particular appreciation for:

- Clean UI organisation
- Backend response speed
- Smooth device addition and control

### *E. System Stability and Fault Tolerance*

The Flask server kept uptime at over 99% throughout the test period of one week. Background threads processing energy monitoring and budget enforcement operated without contention due to GPIO thread locks. Sensor read failures or Firebase disconnection exceptions were gracefully recovered from using retry logic and error logging.

### *F. Discussion*

The findings confirm the feasibility and benefits of an integrated smart home system for smart energy management. The two-way communication between Raspberry Pi and the Flutter app as well as Firebase guaranteed real-time synchronization and control. Additionally, the dynamic priority mechanism and automated response to budget limitations illustrate how IoT and cloud platforms may be used optimally for demand-side energy efficiency.

But scalability remains a factor. With more devices, optimization of communication rate and simultaneous processing will be needed. Security and privacy of data stored in the cloud are also a priority in future updates. While these are issues to be considered, the present implementation provides a solid foundation for intelligent energy-aware houses.

Fig. 4. Initially all devices turned on

Fig. 5. Devices turned off by Power Budget Enforcement

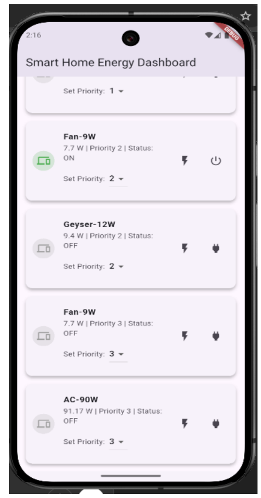


Fig. 6. Mobile App Usability

## VI. Future Scope

To scale the system to large applications, i.e., apartment buildings, commercial buildings, or smart cities, the architecture is expandable with modular IoT hubs, cloud-based centralized control, and scalable backend infrastructure. By aggregating several Raspberry Pi units or edge devices per zone, the system can handle multiple loads and user profiles at the same time. Cloud dashboards and analytics would enable facility managers to have centralized monitoring and reporting of energy, while load forecasting and dynamic pricing algorithms may be implemented to ensure optimized use of energy at a community level. Large-scale deployment would facilitate energy conservation, peak load management, and more intelligent utility billing across an array of households or facilities.

## VII. Conclusion

The work introduces an end-to-end IoT-based smart home energy management system that integrates real-time monitoring, automatic load control, user-specified priority, and dynamic budget-driven energy optimization. The use of a Raspberry Pi 3B+, PZEM-004T sensor, and relay module and Flutter-based mobile application with Firebase Cloud Firestore provides an end-to-end seamless flow of data and control between hardware and user interface. The system enables users to control energy usage actively by enabling them to budget usage, allocate priorities for essential devices, and issue timely notifications. With automated enforcement of energy limits and optimized device scheduling, the target system considerably increases energy efficiency, cost reduction, and user participation in a contemporary home setting. The provided solution exhibits an economically viable and scalable solution to sustainable home energy management with low-cost, off-the-shelf technologies.

## REFERENCES

[1] El-Khozondar, H.J., Mtair, S.Y., Qoffa, K.O., Qasem, O.I., Munyarawi, A.H., Nassar, Y.F., Bayoumi, E.H. and Abd El, A.A.E.B., 2024. A smart energy monitoring system using ESP32 microcontroller. e-Prime-Advances in Electrical Engineering, Electronics and Energy, 9, p.100666.

[2] Krishna Rao, C., Sahoo, S.K. and Yanine, F.F., 2024. An IoT-based intelligent smart energy monitoring system for solar PV power generation. Energy Harvesting and Systems, 11(1), p.20230015.

[3] A. R. Al-Ali, I. A. Zualkernan, M. Rashid, R. Gupta and M. Alikarar, "A smart home energy management system using IoT and big data analytics approach," in IEEE Transactions on Consumer Electronics, vol. 63, no. 4, pp. 426-434, November 2017, doi: 10.1109/TCE.2017.015014.

[4] Affum, E.A., Agyeman-Prempeh, K., Adumatta, C., Ntiamoah-Sarpong, K. and Dzisi, J., 2021. Smart home energy management system based on the internet of things (IOT). International Journal of Advanced Computer Science and Applications, 12(2).

[5] H. Ismail, I. Jahwar and B. Hammoud, "Internet-of-Things-Based Smart-Home Time-Priority-Cost (TPC)-Aware Energy Management System for Energy Cost Reduction," in IEEE Sensors Letters, vol. 7, no. 9, pp. 1-4, Sept. 2023, Art no. 5502604, doi: 10.1109/LSENS.2023.3307049.

[6] I. -Y. Joo and D. -H. Choi, "Optimal household appliance scheduling considering consumer's electricity bill target," in IEEE Transactions on Consumer Electronics, vol. 63, no. 1, pp. 19-27, February 2017, doi: 10.1109/TCE.2017.014666.

[7] A. Gozuoglu, O. Ozgonenel and C. Gezegin, "Design and Implementation of Controller Boards to Monitor and Control Home Appliances for Future Smart Homes," in IEEE Transactions on Industrial Informatics, vol. 20, no. 9, pp. 11458-11465, Sept. 2024, doi: 10.1109/TII.2024.3404591.

[8] Khan, M.A., Sajjad, I.A., Tahir, M. and Haseeb, A., 2022. Iot application for energy management in smart homes. Engineering Proceedings, 20(1), p.43.

[9] L. Alhmoud and W. Marji, "Optimization of Three-Phase Feeder Load Balancing Using Smart Meters," in IEEE Canadian Journal of Electrical and Computer Engineering, vol. 45, no. 1, pp. 9-17, winter 2022, doi: 10.1109/ICJECE.2021.3113521.

[10] Munoz, O., Ruelas, A., Rosales, P., Acuña, A., Suastegui, A. and Lara, F., 2022. Design and development of an IoT smart meter with load control for home energy management systems. sensors, 22(19), p.7536.

[11] Tran, L.N., Cai, G. and Gao, W., 2023. Determinants and approaches of household energy consumption: A review. Energy Reports, 10, pp.1833-1850.

[12] G. Bedi, G. K. Venayagamoorthy and R. Singh, "Development of an IoT-Driven Building Environment for Prediction of Electric Energy Consumption," in IEEE Internet of Things Journal, vol. 7, no. 6, pp. 4912-4921, June 2020, doi: 10.1109/JIOT.2020.2975847.

[13] Poyyamozhi, M., Murugesan, B., Rajamanickam, N., Shorfuzzaman, M. and Aboelmagd, Y., 2024. IoT—A Promising Solution to Energy Management in Smart Buildings: A Systematic Review, Applications, Barriers, and Future Scope. Buildings,14(11),p.3446.

[14] P. Powroźnik and P. Szcześniak, "Energy Management of Home Devices With Smart Response for the Energy Generation Profile," in IEEE Transactions on Industrial Informatics, vol. 20, no. 4, pp. 6995-7007, April 2024, doi: 10.1109/TII.2024.3353850.

[15] Singh, P.P., Khosla, P.K. and Mittal, M., 2019. Energy conservation in IoT-based smart home and its automation. Energy conservation for IoT devices: concepts, paradigms and solutions, pp.155-177.

[16] H. Shao, L. Rao, Z. Wang, X. Liu, Z. Wang and K. Ren, "Optimal Load Balancing and Energy Cost Management for Internet Data Centers in Deregulated Electricity Markets," in IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 10, pp. 2659-2669, Oct. 2014, doi: 10.1109/TPDS.2013.227.

[17] G. Musa Raza, I. Ullah, M. Salah Ud Din, M. Atif Ur Rehman and B. -S. Kim, "INF-NDN IoT: An Intelligent Naming and Forwarding in Name Data Networking for Internet of Things," in IEEE Access, vol. 12, pp. 114319-114337, 2024, doi: 10.1109/ACCESS.2024.3444903.

[18] D. Sarabia-Jácome et al., "Environment Monitoring Subsystem Based on Recursive InterNetwork Architecture," in IEEE Access, vol. 12, pp. 104272-104290, 2024, doi: 10.1109/ACCESS.2024.3434747.

[19] El-Azab, R., 2021. Smart homes: Potentials and challenges. Clean Energy, 5(2),pp.302-315.

[20] W. Ali, I. U. Din, A. Almogren, M. Zareei and R. R. Biswal, "MicroTrust: Empowering Microgrids With Smart Peer-to-Peer Energy Sharing Through Trust Management in IoT," in IEEE Access, vol. 12, pp. 134985-134996, 2024, doi: 10.1109/ACCESS.2024.3459936.