Research Trends International Journal of Technology and Innovation
An International Peer-Reviewed, Refereed & Open-Access Journal
ISSN : 3144-3273
Available Online at : https://rtijti.com/
doi : https://doi.org/10.5281/zenodo.23720410

SmartHome: An IoT-Based Home Automation and Control System with Remote Monitoring and Energy Management

C. P. Divate[1], N. R. Bhokare[2], S. S. Chougule[3], Sabina Jamadar[4], Mohammad Rakheeb Inamdar[5], Akanksha Sungar[6], Sudarshan Kharat[7], Vishnu Parit[8]
[1] Academics Dean, [2] Head of Department, [3] Project Guide, [4, 5, 6, 7, 8] Students
Department of Computer Engineering, ATS Sanjay Bhokare Group of Institute, Miraj, Maharashtra, India
Corresponding author: sabinajamadar795@gmail.com

ARTICLE INFO

Article History:
Received : 24-10-2026
Accepted : 25-10-2026
Published : 26-10-2026

Publication Issue
Volume 1, Issue 4
October – December 2026

Page Number
28-33

ABSTRACT

Conventional homes depend on manual switching of lights, fans, and air conditioners. Appliances are often left running when nobody is in the room, and the owner has no way to control them or receive security alerts when away from home. Commercial smart home products solve part of this problem, but they are often costly, tied to one brand, and difficult to extend.

This paper proposes SmartHome, a low-cost home automation and control system through which users can monitor and control household appliances from a web or mobile application. ESP32 or Arduino microcontrollers switch appliances through relays and read motion, temperature, light, gas, and door sensors. Data is exchanged with a Node.js or Python (Flask/Django) back end through a REST API and lightweight protocols such as MQTT and WebSockets. A MySQL or PostgreSQL database stores users, devices, automation requests, ratings, and energy records, and cloud services such as Firebase or AWS provide remote access. Users can create time schedules and sensor-based rules, for example switching a light on when motion is detected in a dark room. Access is protected by JWT-based authentication and bcrypt password hashing.

The system is organised into four modules: homepage, admin, end-user, and database. The paper presents the architecture, methodology, energy measures, and planned evaluation of the system.

Index Terms - Smart Home, Home Automation, Internet of Things, ESP32, MQTT, Energy Efficiency, Remote Monitoring, Sensors

I. INTRODUCTION

The demand for automated and energy-efficient homes is growing quickly. People want to manage household appliances and check on their homes without depending on wall switches. Appliances left on by mistake waste electricity, and homeowners usually have no idea what is happening at home while they are away.

The Internet of Things (IoT) makes it possible to connect appliances and sensors to a central application. Low-cost microcontrollers such as the ESP32 and Arduino, combined with Wi-Fi or Bluetooth, can switch devices and send sensor readings to a server [2], [4]. However, many commercial smart home products are expensive, work only with one brand, and offer little room for customisation by students and small households.

1.1 Background and Motivation

This paper proposes SmartHome, a home automation and control system through which users can switch lights, fans, air conditioners, and other appliances ON or OFF, view their current status, and set automated

Copyright © 2026 The Author(s) : This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/)

schedules from a web or mobile interface. Sensors for motion, temperature, light, gas, and doors let the system react to the home environment on its own. The motivation is to bring remote control, safety alerts, and energy saving together in a design that a student team can build with parts that cost a few thousand rupees.

1.2 Contributions

The main contributions of this work are as follows:

- A low-cost IoT architecture connecting microcontrollers, sensors, a cloud back end, and a web or mobile application.
- A rule-based and schedule-based automation scheme aimed at energy saving.
- A relational database design that stores devices, automation requests, ratings, and energy records.

If implemented as designed, the system can improve convenience, home security, and energy efficiency.

1.3 Organisation of the Paper

Section II reviews the literature and Section III analyses the main problems of smart home systems. Section IV describes the methodology as four modules together with the IoT layer, database, and energy measures. Section V discusses the expected benefits and the evaluation plan, Section VI concludes the paper, and Section VII lists future work.

II. LITERATURE REVIEW

A. qToggle Home Automation Platform: Stolojescu-Crișan et al. presented qToggle, an IoT-based smart home automation system that connects sensors, actuators, and other data sources through a common application programming interface (API). The API makes it simple to bring devices from different makers into one platform. However, the work centres on the API and does not cover energy analysis or user feedback [1].

B. Review of Smart Home Architecture and Security: Mocrii et al. reviewed IoT-based smart homes in terms of system architecture, software, communication technologies, privacy, and security. The review explains how home devices talk to each other over networks and points out the security and privacy risks that appear once household devices are connected to the internet [2].

C. Energy-Efficient Smart Home Systems: Rocha Filho et al. studied energy-efficient smart home systems built on wireless sensor and actuator networks with computational intelligence for decision making. The work shows how intelligent control can cut unnecessary electricity use, but it focuses on the infrastructure and decision process rather than a complete user application [3].

D. Survey of Tools and Communication Interfaces: A survey of smart home automation systems covered the tools, technologies, architectures, and communication interfaces such as Wi-Fi, Bluetooth, and ZigBee. It also discussed the cost and technical trade-offs involved in selecting components [4].

E. Raspberry Pi Based Energy-Efficient Prototype: Ahmed et al. built a prototype on a Raspberry Pi 5 with sensors for motion, temperature, humidity, light, and smoke. A web dashboard allows manual control and monitoring, appliances are automated from sensor data, and energy data is recorded for analysis. The Raspberry Pi based design, however, costs more per node than a microcontroller based one [5].

F. IoT for Smart Homes, Challenges and Solutions: Risteska Stojkoska and Trivodaliev reviewed IoT for smart homes and proposed a holistic framework that links home devices, the smart grid, and cloud services. They identified interoperability, data volume, and energy management as open problems [6].

G. Home Automation with Weather Monitoring: Willoughby et al. combined appliance control with monitoring of weather parameters in a low-cost IoT system. The design shows that a single microcontroller can handle switching and sensing together, but it does not include user roles or a stored history of energy use [7].

A comparison of the reviewed studies is given in Table 1, arranged from the most recent study to the oldest.

Table 1. Comparative analysis of reviewed studies

Ref.	Year	Technique	Key Finding	Limitation
[5]	2026	Raspberry Pi 5 + sensors + web dashboard	Sensor-based automation with energy logging	Higher cost per node
[4]	2022	Survey of tools and interfaces	Compares Wi-Fi, Bluetooth, ZigBee	No system built
[7]	2022	IoT control + weather sensing	Switching and sensing on one board	No user roles or energy history
[1]	2021	qToggle API platform	Common API joins many devices	No energy analysis or feedback
[3]	2019	WSAN + computational intelligence	Smart control cuts waste	Not a user application
[2]	2018	Review of architecture and security	Maps privacy and security risks	Review only
[6]	2017	Review and framework	Links home, grid, and cloud	Conceptual framework

2.1 Research Gap

The reviewed works address APIs, surveys, energy methods, or single prototypes. Few of them offer a low-cost, student-friendly system that combines ESP32 or Arduino control, role-based web and mobile access, schedules and sensor rules, energy records, and user ratings in one design. The proposed system is intended to fill this gap.

III. PROBLEM ANALYSIS

The main problems of smart home systems, and how the proposed design responds to each, are discussed below.

3.1 Usability

Complicated controls and unclear navigation make many systems hard to use, especially for elderly family members. A simple, responsive dashboard with one tap per appliance is proposed.

3.2 Connectivity and Compatibility

Devices from different makers use different protocols. A common architecture based on Wi-Fi, MQTT, and REST gives a single path for all commands and readings.

3.3 Security and Privacy

Unauthorised access could let an attacker control appliances or read sensor data. JWT authentication, bcrypt password hashing, and encrypted communication are used to reduce this risk [2].

3.4 Energy Management

Appliances left running waste power. Schedules and sensor rules switch them off automatically when they are not needed [3], [5].

3.5 Reliability

Network failures can delay control commands. Lightweight protocols and regular device status checks support near real-time operation.

3.6 Scalability

New devices must be added without changing the rest of the system. A modular architecture allows the home to grow one device at a time.

IV. METHODOLOGY

The system has two types of users and follows the layered architecture shown in Figure 1. A home user registers, logs in, adds devices, controls appliances, monitors sensors, and sets schedules. An administrator manages users, devices, permissions, and system settings. The modules are summarised in Table 2.

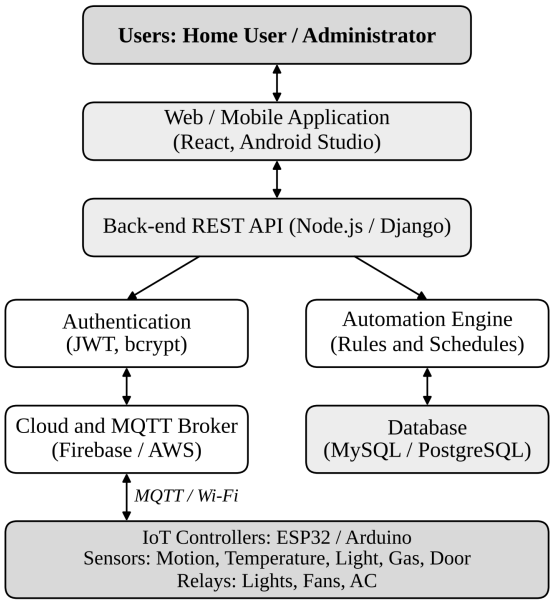


Figure 1. Block diagram of the proposed SmartHome system.

Table 2. Summary of the modules of SmartHome

Module	Function	Key Tools
1. Homepage	Home, About, Features, Devices, FAQs, Contact, Login	HTML5, CSS3, React
2. Admin	Users, devices, monitoring, permissions, activity logs	Node.js / Django
3. End user	Login, appliance control, status, schedules, profile	Web / Android app
4. Database	Users, devices, automation, ratings, energy	MySQL / PostgreSQL
IoT layer	Switching and sensing, wireless link	ESP32, Arduino, MQTT

4.1 Module 1: Homepage

The public homepage introduces the system and contains the Home, About Us, Smart Home Features, Devices and Appliances, FAQs, and Contact Us sections. A secure login portal gives registered users and administrators access to their own features.

4.2 Module 2: Admin Panel

The admin panel is used for supervision. After authentication the admin reaches a dashboard with user management, device management, device monitoring (status and connectivity), access permission management, activity log review, and logout.

4.3 Module 3: End-User Panel

The end-user panel offers registration, login, a dashboard, appliance control (for example switching a light ON or OFF), device status monitoring (ON, OFF, or disconnected), profile management, logout, and remote access from outside the home. Passwords are stored as hashes and never as plain text.

4.4 IoT Layer and Automation

ESP32 or Arduino controllers drive relays connected to appliances and read the motion (PIR), temperature, light (LDR), gas, and door sensors. They communicate over Wi-Fi using MQTT or WebSockets for low delay. The automation engine supports two types of rules: time schedules and sensor conditions. The working of a typical sensor rule is shown in Figure 2. If motion is detected and the ambient light is below a threshold, the light is switched ON. If no motion is seen for a set period T, the light is switched OFF and the ON time is logged for energy calculation.

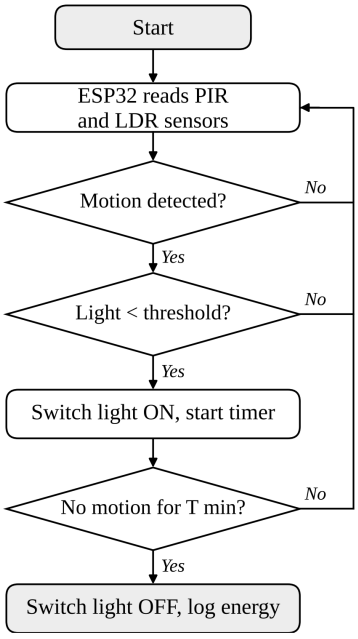


Figure 2. Working flow of a motion and light based automation rule.

4.5 Module 4: Database Design

A relational database stores all records. The proposed tables are listed in Table 3. A user can manage many devices and create many automation requests and ratings. Each energy record belongs to one device, which allows consumption to be tracked per appliance.

Table 3. Proposed database tables

Table	Main Fields
Users	user_id (PK), username, hashed password, email, role (Admin, User), created_at
Customers	customer_id (PK), user_id (FK), account_id, phone_number, address
Smart Devices	device_id (PK), user_id (FK), device_type, location, status
Automation Requests	automation_id (PK), user_id (FK), device_id (FK), description, status (Pending, Active, Completed), created_at
Ratings	rating_id (PK), device_id (FK), user_id (FK), score, feedback, created_at
Energy Consumption	energy_id (PK), device_id (FK), user_id (FK), amount, energy_status, record_date

4.6 Energy and Performance Measures

The energy consumed by a device over a period is estimated from its rated power and its ON time as

$$E = (P \times t) / 1000 \quad (1)$$

where E is the energy in kWh, P is the power rating in watts, and t is the ON time in hours. The energy saving achieved by automation is

$$S = ((E_{before} - E_{after}) / E_{before}) \times 100\ \% \quad (2)$$

where E_{before} and E_{after} are the energy used before and after automation over the same period. The automation success rate is

$$R = (N_{success} / N_{total}) \times 100\ \% \quad (3)$$

where $N_{success}$ is the number of scheduled or rule-based tasks that were executed correctly and N_{total} is the total number of tasks triggered.

4.7 Technology Stack and Security

The front end uses HTML5, CSS3, JavaScript, and React or Angular, with Android Studio or Flutter for the mobile app. The back end uses Node.js or Python (Flask/Django) with a REST API. Data is stored in MySQL or PostgreSQL, and cloud services such as Firebase, AWS, or Azure support remote access and synchronisation. Security measures include JWT authentication, bcrypt hashing, SSL/TLS, role-based access control, and regular backups. Development follows Agile practice with GitHub for version control.

V. EXPECTED BENEFITS AND EVALUATION PLAN

5.1 Expected Benefits

Based on the design, the system is expected to give convenient remote control of appliances, real-time monitoring, lower electricity use through schedules and sensor rules, better home security through motion, door, and gas alerts, and a base to which new devices can be added. These are design goals and not measured results.

5.2 Planned Evaluation

After implementation, the system will be evaluated on the following points:

- device response time and system uptime;
- device connectivity and availability;
- error rate and mobile responsiveness;
- sensor accuracy for the motion, temperature, light, gas, and door sensors;
- automation success rate R from Eq. (3) and energy saving S from Eq. (2);
- security checks for authentication and unauthorised access;
- user engagement and satisfaction from user acceptance surveys.

5.3 Feasibility

Technical: ESP32/Arduino, Node.js or Python, MySQL or PostgreSQL, and Android Studio or React are mature and well documented.

Economic: Microcontrollers, relays, and sensors are inexpensive, and the energy saved gives a long-term return.

Operational: The interface is simple enough for users with different levels of technical skill.

Legal: The system must follow data protection, privacy, and IoT security requirements, with authentication and access control for all user data.

VI. CONCLUSION

This paper proposed SmartHome, an IoT-based home automation and control system that lets users control and monitor household appliances in real time from a web or mobile application. ESP32 or Arduino controllers, sensors, a REST back end, a relational database, cloud services, and secure authentication are combined in a modular design. The system aims to improve convenience, home security, and energy efficiency at low cost, and the measures in Section IV give a clear basis for testing these aims once the system is built.

VII. FUTURE ENHANCEMENT AND DIRECTIONS

- Completing the implementation and carrying out the evaluation planned in Section V.
- Voice assistant integration for hands-free control.
- Machine-learning based prediction of usage patterns to create schedules automatically.
- ZigBee support for battery-powered sensors.
- Integration with solar panels and smart energy meters.

ACKNOWLEDGMENT

The authors thank the Department of Computer Engineering, ATS Sanjay Bhokare Group of Institute, Miraj, for its support.

REFERENCES

[1] C. Stolojescu-Crișan, C. Crișan, and B.-P. Butunoi, "An IoT-based smart home automation system," *Sensors*, vol. 21, no. 11, Art. no. 3784, 2021, doi: 10.3390/s21113784.

[2] D. Mocrii, Y. Chen, and P. Musilek, "IoT-based smart homes: A review of system architecture, software, communications, privacy and security," *Internet of Things*, vol. 1-2, pp. 81-98,

2018, doi: 10.1016/j.iot.2018.08.009.

[3] G. P. Rocha Filho, L. A. Villas, V. P. Gonçalves, G. Pessin, A. A. F. Loureiro, and J. Ueyama, "Energy-efficient smart home systems: Infrastructure and decision-making process," *Internet of Things*, vol. 5, pp. 153-167, 2019, doi: 10.1016/j.iot.2018.12.004.

[4] K. Sharma and S. R. N. Reddy, "Survey on smart home automation systems: Tools, technology, architecture and communication interfaces," in *Proc. 2022 Int. Conf. Computing, Communication, and Intelligent Systems (ICCCIS)*, Greater Noida, India, 2022, pp. 406-413, doi: 10.1109/ICCCIS56430.2022.10037661.

[5] A. Ahmed, M. M. Rahman, S. Memon, and T. Ahmed, "An IoT-enabled smart home automation system for energy efficiency with web-based control," arXiv:2605.20981, 2026.

[6] B. L. Risteska Stojkoska and K. V. Trivodaliev, "A review of Internet of Things for smart home: Challenges and solutions," *Journal of Cleaner Production*, vol. 140, pp. 1454-1464, 2017, doi: 10.1016/j.jclepro.2016.10.006.

[7] A. A. Willoughby, A. O. Soge, M. A. Adeleke, and A. O. Ilori, "An IoT-based home automation and weather monitoring system," *International Journal of Research and Innovation in Applied Science*, vol. 7, no. 3, pp. 26-29, 2022, doi: 10.51584/IJRIAS.2022.7302.