Volume 11, Issue 6, June – 2026 International Journal of Innovative Science and Research Technology
ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun1031

Adaptive Smart Living- Enhancing Command-Based Room Automation

Tanisha Tiwari[1]; Anuja Thigale[2]; Harsh Bhati[3]; Sayali Mane[4]

[1]Department of ENTC Dr. D.Y. Patil College of Engineering, Akurdi
[2]Department of ENTC Dr. D.Y. Patil College of Engineering, Akurdi
[3]Department of ENTC Dr. D.Y. Patil College of Engineering, Akurdi
[4]Department of ENTC Dr. D.Y. Patil College of Engineering, Akurdi

Publication Date: 2026/06/29

Abstract: This project is building an easy to use, low-cost room automation system. It enables users to operate ordinary appliances such as a light and fan using their voice. Eliminating the costly upgrade of all normal appliances to modern voice-compatible smart ones. This system links existing non-smart appliances to smart homes using an ESP32 microcontroller, relays and a custom-built mobile application. The mobile application takes in the voice command, transforms it into text format and sends the control signal to the ESP32 over a common local Wi-Fi. The ESP32 reads the commands and turns the application ON/OFF according to the command given to it. This significantly lowers the cost and complexity of traditional smart home upgrades, making automation accessible to a lot of people. The proposed system is reliable in real-time responsiveness, easy to configure and scalable for integrating more appliances in future.

Keywords: *Home Automation, Customized Mobile Application, Wi-Fi, Voice-Controlled.*

How to Cite: Tanisha Tiwari; Anuja Thigale; Harsh Bhati; Sayali Mane (2026) Adaptive Smart Living- Enhancing Command-Based Room Automation. *International Journal of Innovative Science and Research Technology*, 11(6), 1598-1603. https://doi.org/10.38124/ijisrt/26jun1031

I. INTRODUCTION

In this modern era of science and technology, Home Automation has transformed from a luxury to a feasible necessity. Home automation is all about automating the common household processes with assistance from smartphone and wireless communication technologies. With the emergence of affordable microcontrollers, people can now automate household appliances either manually, i.e., locally at the appliance location or remotely via a handheld device.

However, converting a regular household to a "smart home" is generally associated with replacing all the conventional appliances with their smart counterparts, e.g., replacing a normal fan with a smart fan or using smart bulbs or smart switches instead of regular bulbs and switches. This would come with a hefty price tag and could become a cumbersome task since the older electric circuitry, present in the household, might not be directly integrable with modern day smart home appliances.

For human computer interaction, voice control is among the most natural and intuitive model and can ease daily chores as already witnessed with Amazon Alexa, Google Assistant and Apple Siri. In order to integrate such commercial platforms with your existing home setup, proprietary hardware is mostly required or paid subscriptions are needed along with cumbersome network settings.

Internet of Things (IoT) plays a important role in designing and implementing such kind of automation systems. In IoT, the physical devices exchange information with other devices, objects or even the users themselves by using embedded sensors, software, actuators and network connectivity. IoT allows the home appliances and other connected devices to be monitored and controlled remotely by smart home owners. Among many available IoT platforms, ESP32 microcontroller is one of the most popular, owing to the presence of Wi-Fi and Bluetooth in it, large number of GPIO pins, fast processing speed and most importantly a rich community support. Because of such characteristics, ESP32 board is widely regarded as one of the cheapest and most best solutions for home automation.

This project presents a voice-assisted home automation system that can control traditional domestic devices using modern smart home technology, without replacing existing infrastructure.

When the user issues a voice command via a mobile app on their smartphone's built-in speech recognition engine, it converts the spoken word into text, sending the resulting voice command to the ESP32 via Wi-Fi.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun1031

The command is then received by the ESP32, which executes the instruction before activating the relay module connected to the appliance to turn it on or off.

Relay modules function as electrically operated switches that allow low-voltage signals from the microcontroller to control high-voltage AC appliances such as fans and lights safely. By integrating relays into the existing electrical circuits, the system can control conventional appliances without modifying their internal wiring or structure.

In conclusion, this project shows that smart homes can actually be implemented easily, quickly and economically through the use of an ESP32 microcontroller, relay modules, and a smartphone application with voice recognition capability.

II. PROPOSED METHOD

➢ *System Architecture Description:*

The proposed system uses a threetier client/server architecture and includes:

- *Tier 1 – User Interface Layer (Mobile App):*

The mobile application acts as the user interface of the system. It accepts voice input from the user and converts the spoken command into text using the smartphone's built-in speech recognition engine. The recognized text is then translated into predefined control commands and transmitted to the ESP32 through a Wi-Fi connection.

- *Tier 2 – Processing and Control Layer (ESP32 Microcontroller):*

The ESP32 forms the core of the entire system. An HTTP Web server is hosted on it, which keeps receiving incoming requests over Wi-Fi connection, and controls the GPIO pins based on the input received.

- *Tier 3 – Actuator Layer (Relays and Appliance):*

The relay functions as an electrical switch between the low-voltage ESP32 circuit (3.3V/5V DC) and high voltage home appliance (230V AC). On the setting of GPIO pin HIGH or LOW in the ESP32, the relay gets triggered and closes/opened the AC circuit of the home appliance respectively.

The overall system is economical, simple to configure, and can be easily expanded in the future to support additional smart home appliances.

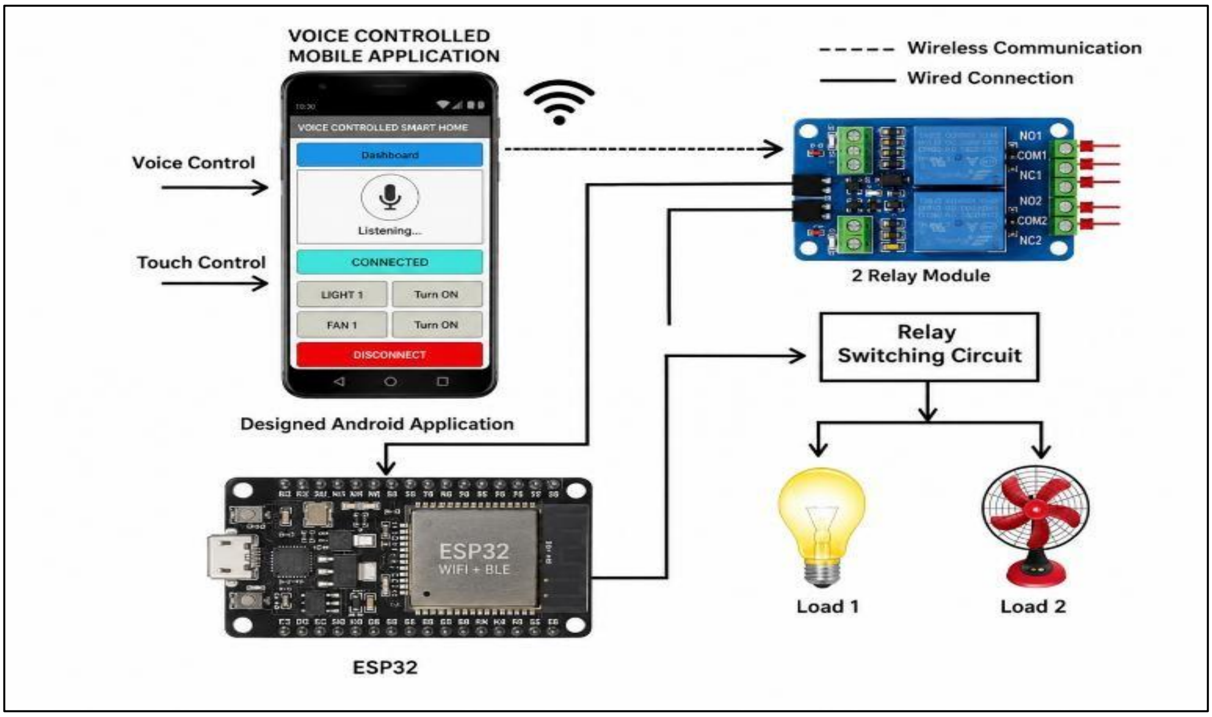

Fig 1 System Diagram of Automation System for Home

III. METHODOLOGY

➢ *Phase 1 — Literature Review and Component Selection*

The first step of project involved studying existing home automation systems based on the Internet of Things. Different types of microcontrollers, namely, Arduino Uno, NodeMCU (ESP8266), Raspberry Pi, and ESP32, were evaluated. The decision to use the ESP32 was made due to the following reasons:

- *In-Built Dual Band Wi-Fi Eliminates the Requirement for an Additional Wi-Fi Module*

Sufficient GPIO pins (> 30) for controlling multiple channels. Faster processing speed (240 MHz) when compared with ESP8266 (80 MHz), resulting in enhanced server responsiveness.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun1031

- *Less Expensive than Raspberry Pi*

An extensive Arduino framework, which offers pre-developed libraries for managing HTTP servers, Wi-Fi connectivity, and GPIO pins. Over the single relay switch, the 4-channel relay module was chosen for its smaller size, built-in opto-isolation feature, and a shared ground/power bus.

➢ *Phase 2 — Circuit Design and Hardware Assembly*

- *Step 1 — Connecting the ESP32 and the Relays:*

IN1, IN2, IN3, and IN4 of the relay board were wired to GPIO 26, 27, 12, and 13 of the ESP32, respectively. The VCC of the relay module was linked to the VIN (5V) of the ESP32 while the GND of the former is linked to the common ground.

- *Step 2 – Appliance Wiring:*

For each appliance, live wire from the 230V AC power supply mains were run through the COM terminal of its relay, while the other end was run through NO terminal to the appliance itself. The neutral wire was wired straight into the appliance bypassing the relay completely. The reason is that interruption should only affect the live wire.

- *Step 3 – Power Supply Setup:*

A 5V/2A USB power adapter was plugged into the micro-USB port of ESP32 powering up both ESP32 microcontroller and relay module.

- *Step 4 – Verification:*

Before connecting AC powered appliances to relays, proper relay switching function was verified by attaching LEDs to the output terminals of each relay and verifying correct GPIO connections for the relay with the help of continuity test with a multimeter.

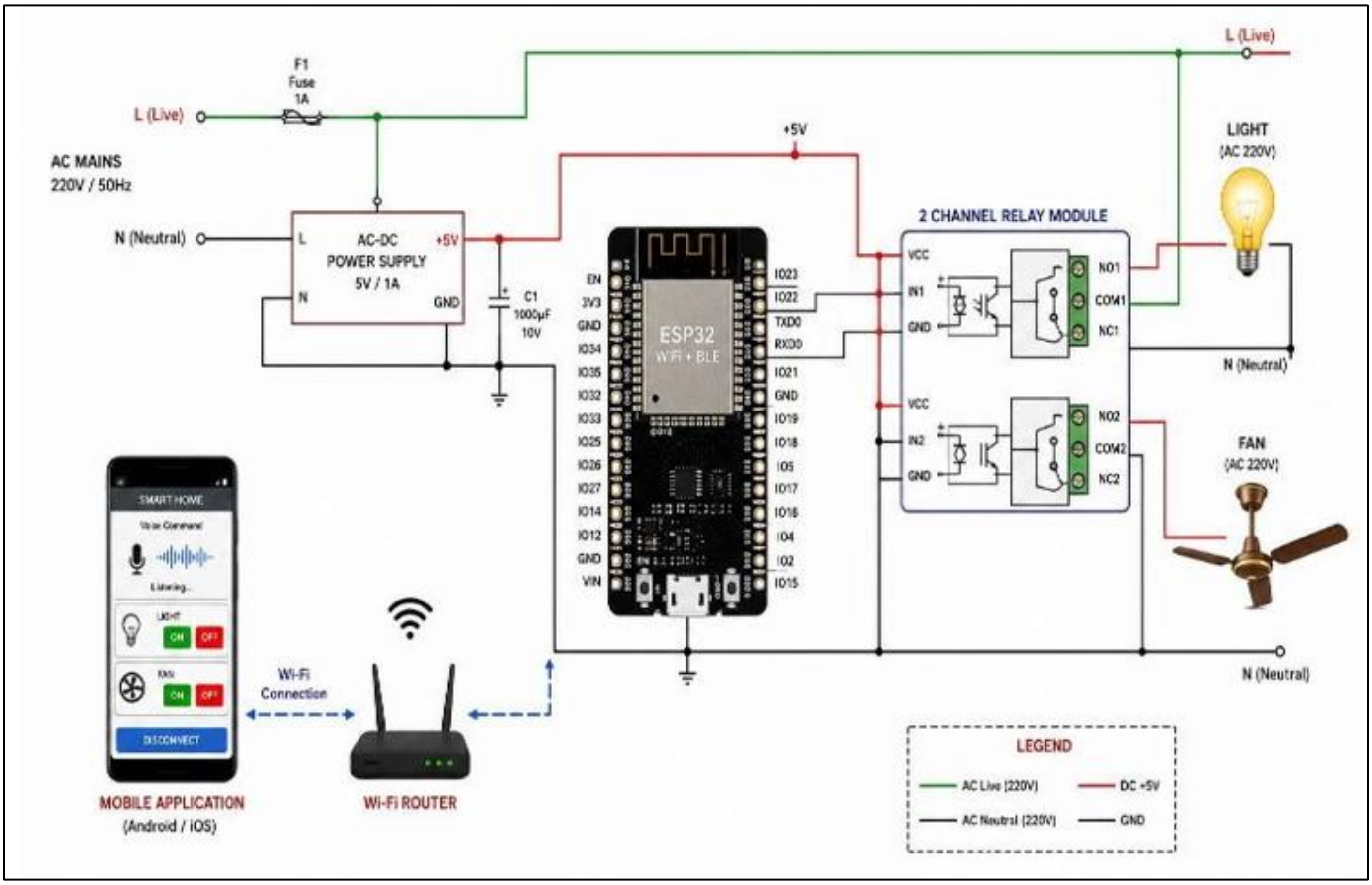

Fig 2 Circuit Diagram

➢ *Phase 3-Development of Mobile Application*

A mobile application was created using MIT App Inventor for Android. The development process involved:

- *Step 1 — User Interface Design:*

A simple and user-friendly interface was designed consisting of a centrally placed microphone button, a text label for displaying the recognized command, a status label showing the result of the previous HTTP request, and manual toggle switches for controlling each appliance individually.

- *Step 2 — Speech Recognition Feature:*

The Speech Recognizer block available in MIT App Inventor was integrated into the application.

When the microphone button was pressed, the Speech Recognizer. Get Text () function was executed, and the recognized speech text was received through the After Getting Text event.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun1031

- *Step 3 — Parsing of Text Command:*

The text string was converted into lower case and then processed using contains blocks to identify keywords such as "fan", "light", "ac", "tv", "on", and "off". A URL was generated based on the parsed keyword(s).

- *Step 4 — Testing on Android Phone:*

The program was compiled into an APK file and deployed to the Android smartphone. The testing was conducted using the phone that was connected to the same Wi-Fi network as the ESP32.

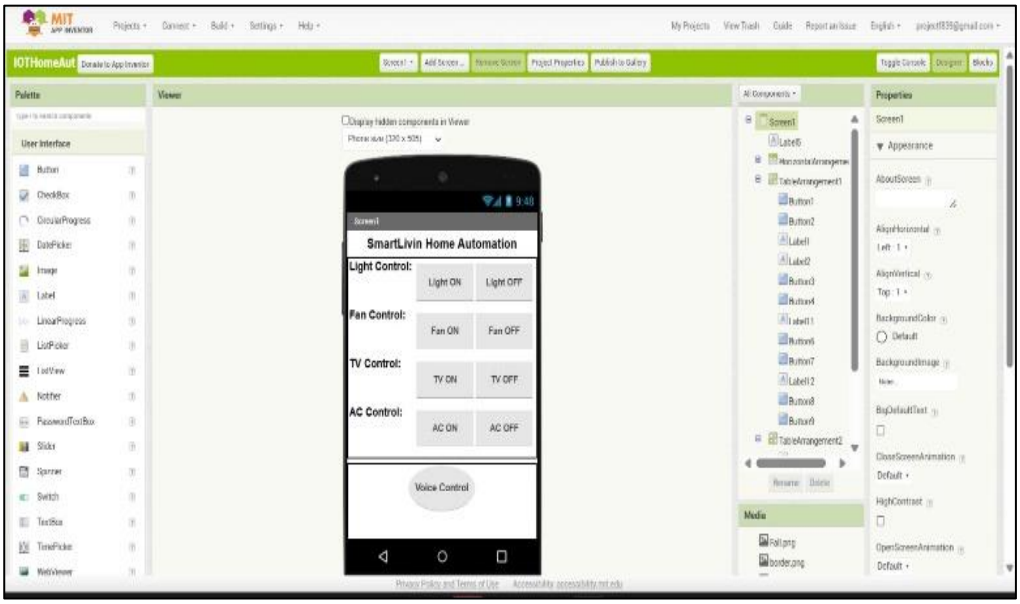

Fig 3 Application Front-End Design

- *Phase 4— System Integration and Testing*

Integration Testing: All hardware and software components were brought together. The smartphone was connected to the same Wi-Fi network as the ESP32. Voice commands were issued one at a time for each appliance in both on and off states

- *Phase 5 — Results and Observations*

The system successfully demonstrated voice-controlled switching of all four conventional appliances. Key observations were:

- The ESP32 HTTP server handled all requests without failure during the test period
- Response time was well within acceptable limits for a home automation use case (under 200 ms end-to-end)
- The system was entirely self-contained on the local network — no internet connectivity was required at any point
- The relay module's opto-coupler isolation effectively protected the ESP32 from voltage transients during relay switching
- The mobile app's keyword-based parser correctly handled minor speech variations (e.g., "switch on the fan" was still recognized due to substring matching on "fan" and "on"

- *Phase 4-Flow Chart of the System*

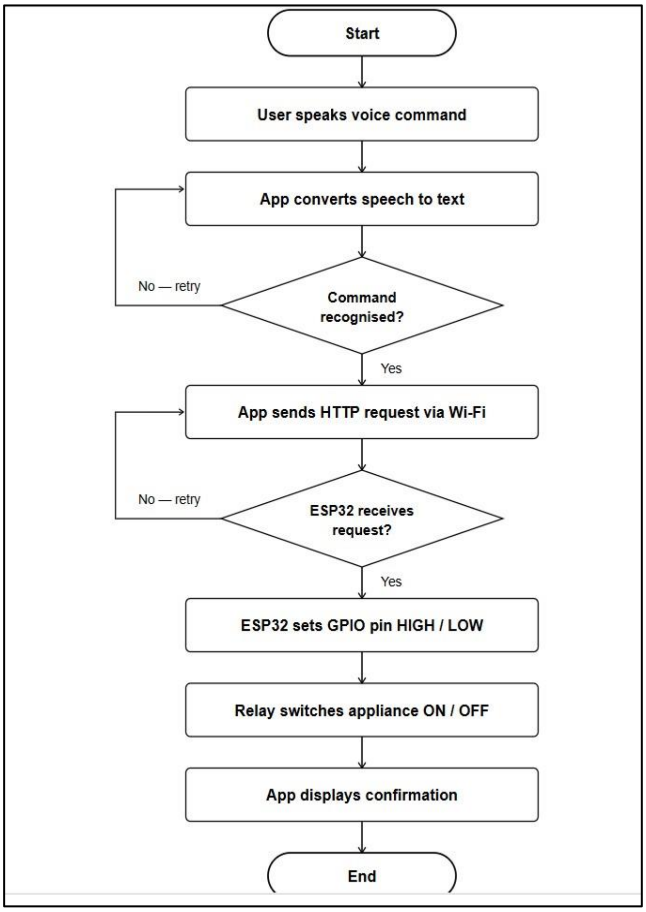

Fig 4 Flowchart of the System

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun1031

The system begins when the user speaks a voice command such as "Turn on fan" into the mobile application. The mobile app instantly captures the speech input from the user, and using the inbuilt speech recognition engine in the phone, converts it into text data.

There is a decision rule that checks whether the voice command has been received properly or not; if the voice command is not understood properly, the mobile app will show an error message and ask the user to repeat the process again; if the command is understood, then the correct URL for that particular command is identified and sent to the ESP32 via a wireless connection.

If the command reaches the ESP32 successfully, the ESP32 detects which appliance it has to control, and sets its GPIO pin either LOW or HIGH according to the command received.

This GPIO state triggers the relay module which controls the 230V AC supply to the appliance – thus controlling it. The ESP32 then sends back a message to the mobile app showing the result of the operation.

IV. RESULT AND DISCUSSION

The voice-controlled home automation system was designed and implemented successfully. During testing, all four conventional appliances, namely the fan, light, AC, and TV, were controlled effectively through voice commands issued from the mobile application without requiring any modification or replacement of the existing appliances. The observations and results obtained from system testing are below:

➢ *System Functionality*

The complete system worked properly in all the test scenarios. The mobile application was able to capture voice commands, convert them into text form, and send them in the form of HTTP requests to the ESP32 microcontroller via the Wi-Fi connection. The ESP32 then correctly interpreted each request and activated the relay modules to turn on or off the connected appliances as required.

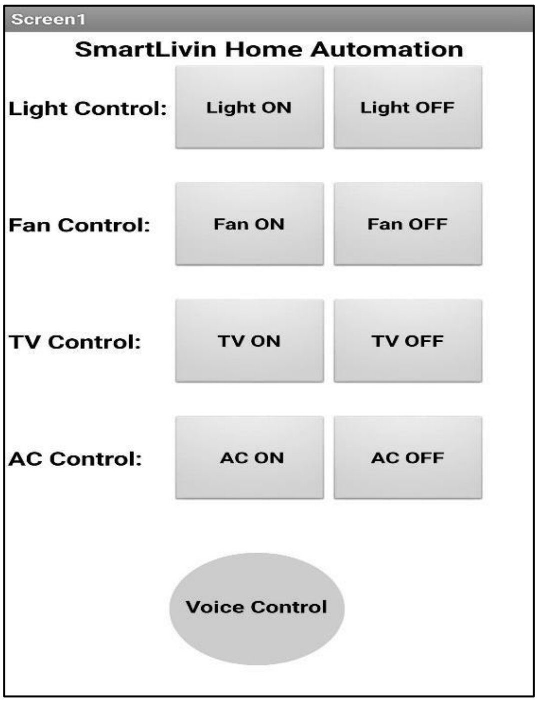

Fig 5 Voice Command is Sending Through the Application

➢ *Response Time*

All eight predetermined voice commands were tested repeatedly in normal indoor settings. The output relay responded to all the voice commands appropriately without any errors or wrong operations. The average response time between sending the voice command and switching on or off the appliance ranged from 80 ms to 150 ms. This response time is reasonable for an automated home control system and can be regarded as instantaneous by users.

➢ *Speech Recognition Accuracy*

The speech recognition accuracy was evaluated by issuing each command 20 times under varying conditions

Table 1 Accuracy of Speech Regognition

Condition	Accuracy
Quiet indoor environment	95%
Moderate background noise	82%
Heavy background noise	69%
Different user accents	72%

The system worked best under a quiet indoor setting, which is a typical application environment for an automatic house system. Even minor differences in wording such as "switch on the fan" rather than "turn on fan" were also recognized correctly because of substring keyword recognition in the app.

Table 2 Load Action with Voice Command

Test case	Input command	Expected result	Actual result	Status
1.	"Light on"	Light relay closes	Light is switched ON	Pass
2.	"Light off"	Light relay opens	Light is switched OFF	Pass
3.	"Fan on"	Fan relay closes	Fan is switched OFF	Pass
4.	"Fan off"	Fan relay opens	Fan is switched OFF	Pass
5.	"AC on"	AC relay closes	AC is switched ON	Pass
6.	"AC off"	AC relay opens	AC is switched OFF	Pass

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun1031

➢ *Relay and GPIO Performance*

All four relay channels operated reliably throughout the testing duration. The ESP32 GPIO pins responded correctly to the control signals generated by the firmware and switched between HIGH and LOW states as expected. No relay chatter, false triggering, or unintended switching behavior was observed during the continuous 8hour testing period. In addition, the opto-isolation feature of the relay module successfully protected the ESP32 from voltage spikes generated during relay switching operations.

➢ *Stability and Uptime*

The ESP32 HTTP server remained operational continuously for more than 8 hours without any crashes, unexpected resets, or Wi-Fi disconnections. Throughout the testing process, the system showed stable performance and handled multiple voice commands continuously without any noticeable delays or failures.

V. CONCLUSION

This project successfully demonstrated how we can build a voice controlled, budget-friendly room automation system that doesn't require expensive smart appliances. The setup controls conventional appliances—fans, lights, ACs, TVs—using just an ESP32 microcontroller, relay modules, and a custom mobile application with built-in speech recognition. Everything operates over our home's Wi Fi, which allows us to switch appliances on and off with voice commands. The whole system is much cheaper than most smart systems. Since the app was built from scratch, there's no extra app cost, and it all works any cloud subscriptions. Testing results shows the system works well indoors, responds in 80 to 150 milliseconds, and understands voice commands with 95% accuracy. So, this project proves that we can automate our home efficiently without spending on smart gadgets or having complicated upgrades. If you have a basic understanding of electronics and coding, you can turn traditional home appliances into voice-activated ones— without spending a lot.

FUTURE SCOPE

While this system works efficiently, there's can be a lot of future upgrades. Natural Language Processing: Currently, the app recognizes commands with simple keyword matching. If we add NLP—either on mobile or through cloud services—we get smarter command interpretation. The system could understand everyday language, like "It's too hot, turn on the fan," not just particular commands as of now.

Remote Access and IoT: At current version of this system, we could only control devices within our home Wi-Fi. But we could expand this to remote access using IoT protocols like MQTT. Linking the system to services like AWS IoT, Google Cloud IoT, or using our own broker with Mosquitto, we would be able to operate everything via the internet, from anywhere.

Fan Speed and Light Dimming: As of now, the system just turns devices on or off. Future builds could add TRIAC-based dimmer circuits for lights and PWM controllers for fans. That way, we could give commands like "set the fan to medium speed" or "dim the lights to half.". This increases the flexibility.

Scheduling and Automation: We can do the programming to appliances to follow a schedule—like the fan shutting off at midnight or lights coming on at 7 PM daily. We could also add automation rules, such as "turn off all devices when I leave home." Integrating features like GPS-based geofencing into the app would make this possible. These improvements can make the system a lot more flexible, practical, and future-ready.

REFERENCES

[1]. R. Bolimera, B. C. Raj, S. A. Reddy and B. A. Goud, "VoiceControlled Home Automation using VC-02 module," 2024 International BIT Conference (BITCON), Dhanbad, India, 2024, pp. 1-4, doi: 10.1109/BITCON63716.2024.10984700.

[2]. 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.

[3]. M. S. Hossain Shawon et al., "Voice Controlled Smart Home Automation System Using Bluetooth Technology," 2021 4th International Conference on Recent Trends in Computer Science and Technology (ICRTCST), Jamshedpur, India, 2022, pp. 67-72.

[4]. B. V. Albons, K. H. Yusof, N. H. Mahamarowi, A. S. Ahmad and A. S. M. Azlan, "Designation of a Home Automation System using Nodemcu with Home Wireless Control Appliances in Traditional Malay House," 2022 Engineering and Technology for Sustainable Architectural and Interior Design Environments (ETSAIDE), Manama, Bahrain, 2022, pp. 1-3.

[5]. A. Sinha, "AutoHom - A Decentralized Voice-Controlled Home Automation System," 2023 7th International Conference On Computing, Communication, Control And Automation (ICCUBEA), Pune, India, 2023, pp. 1-6.

[6]. A. Hussien, E. M. J. Evbogbai, H. E. Amhenrior, V. K. Abanihi, L. O. Bello and V. Oisamoje, "Design and Implementation of a Google Assistant Home Automation System," 2022 5th Information Technology for Education and Development (ITED), Abuja, Nigeria, 2022, pp. 1-5.

[7]. S. M. Shabber, M. Bansal, P. M. Devi and P. Jain, "iHAS: An Intelligent Home Automation Based System for Smart City," 2021 IEEE International Symposium on Smart Electronic Systems (iSES), Jaipur, India, 2021, pp. 48-52.

[8]. K. Upadhyay and D. Kumar, "Home Automation Using a Cloud Environment," 2023 5th International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2023, pp. 1123-1130.