Journal of Edge Computing, 2025, Vol. **4**, Iss. 1, pp. 17–34 https://doi.org/10.55056/jec.814

IoT-based home control system using NodeMCU and Firebase

Paul O. Ayeni[1], Olayiwola C. Adesoba[1]

[1] *Federal University of Technology Akure, P.M.B. 704, 340110, Akure, Ondo State, Nigeria*

Abstract. Integrating Internet of Things (IoT) technology in home control systems has led to a significant advancement in the control and monitoring of home appliances and environmental conditions. This study presents the development of an IoT-based home control system that uses the NodeMCU ESP8266 microcontroller to manage several home appliances. This system connects to a Firebase real-time database to store or retrieve data in real-time. The user interacts with the IoT system through a mobile application named My Home, developed using Java programming language, and this mobile application is also interfaced with the Firebase cloud server. The developed IoT system allows users to control home appliances such as lights, sockets, fans, and cookers, and it also provides real-time updates of environmental parameters, such as temperature and humidity, which are measured by a DHT11 sensor. In addition, a PIR motion sensor was integrated into the system to enhance the home's security by detecting intrusion. The system's hardware functionality is based on a written Arduino code, which establishes Wi-Fi connectivity to a predefined network, communicates with the Firebase database, handles appliance control and manages sensor data. This IoT-based home control system shows the potential of integrating microcontrollers with cloud services to create a smart, responsive and user-friendly home control platform. The developed IoT system offers a foundation for advancements, thereby making it a valuable contribution to the growing field of smart home technology.

Keywords: IoT, Firebase, NodeMCU, mobile app, firebase realtime database, control, radio-frequency identification, Arduino, cloud, proteus, application programming interfaces, relay module, automation

1. Introduction

Advancements in technology have been rapid in recent times. One of the primary drivers of this advancement is to increase the quality of the standard of living, which aids the comfort of man. In other words, this drive for comfort births creativity, and as such, technological advancement is inevitable. This advancement cuts across all life sectors, from healthcare to education to agriculture to entertainment to energy. Almost all sectors are interconnected because as a specific sector is being developed, others must also be stepped up to catch up with the pace of development. One of the most important inventions and products of the fast-paced development of technology is the Internet. Although there are numerous Internet applications, the Internet of Things (IoT) is a recent and emerging application. It is a network based on the Internet that aims to enable real-time communication among devices and humans using various technological methods, allowing devices to be connected to the Internet [16, 19].

The phenomenon of IoT was well-favoured by the quick growth and advancement of different areas of wireless sensor networks, radio-frequency identification (RFID), and mobile communication [9]. This quick growth of IoT happened less than 50 years

0009-0009-7255-6343 (P. O. Ayeni); 0009-0007-1238-2287 (O. C. Adesoba)
ayenipaultobi1234@gmail.com (P. O. Ayeni); ocadesoba@futa.edu.ng (O. C. Adesoba)
https://linkedin.com/in/paul-ayeni-188423252 (P. O. Ayeni);
https://linkedin.com/in/engr-olayiwola-adesoba-aa049b63 (O. C. Adesoba)

after its first experimentation, which occurred in 1982 when a vending machine was connected to the Internet. Later, in 1999, Kevin Ashton formed the term "Internet of Things", which has been widely accepted since then [11].

The basic principle of IoT is the advanced interconnection of many units or devices inside a network that is powered by sensors, actuators and other components that operate wirelessly or wirelessly [9]. In essence, IoT has numerous devices connected through a platform, and this platform has a number of roles, including device/unit management, network security, data analysis, data processing and storage, and device response management [5]. IoT has numerous advantages and applications, such as IoT in home control and automation, IoT for aiding the aged and people with disabilities, IoT in advanced data gathering, IoT in industrial manufacturing, IoT in smart cities, and many more [11, 13, 15].

IoT, incorporated into home control and automation systems, rapidly increases human life quality [17]. Control is the ability to exercise influence over something and dictate its behaviour. At the same time, automation is the ability of a system or machine to operate and function appropriately with very minimal or no human intervention. Home control and automation, on the other hand, involves the management of home appliances automatically and remotely using a particular type of technology which could be connected through the Internet. In other words, smart home technologies are used to manage electrical home appliances and can also include a security system [4]. There are several benefits of the integration of IoT into home control systems. These benefits include enhanced home security and safety, energy management, and many more [1, 20]. IoT enables users to live comfortably and resourcefully manage energy at home [17].

In a study on an IoT-based home automation system, the IoT system integrates a NodeMCU ESP8266 microcontroller with several sensors, a multi-plug, and the Blynk app to monitor power usage and environmental data [5]. Key functionalities of the system in this study include overload and over-temperature protection, where the IoT system can switch off devices if the power consumption exceeds 200W or if the temperature crosses a threshold, thereby preventing damage [5]. Additionally, the system allows real-time data monitoring and sends notification updates on the Blynk app and email notifications [5]. However, this system does not include a global or central mechanism to turn off devices, which the user can only manually control. In other words, although the system can automatically switch off devices, there is no provision for users to deactivate all devices with the tap of a button manually. As a result, in cases where the system's sensors fail and an emergency arises, users would have to deactivate each connected device individually. The implication is that, as the system scales up and more devices are connected, individually deactivating devices may become unfeasible in a short time. Additionally, this system only monitors and manages energy consumption, not including security monitoring functionalities.

In another study on an IoT-based environmental monitoring system, a NodeMCU microcontroller was utilized and incorporated with the Firebase cloud server [12]. The system was developed to predict rain, take temperature and humidity readings, and detect the presence of CO gas [12]. With the aid of a mobile application, Firebase cloud server, and three Wireless Sensor Network (WSN) nodes, of which each consists of a DHT11 temperature and humidity sensor, a NodeMCU board, a MQ7 gas sensor, a FC-37 rain sensor, and a battery, this aim was achieved [13]. This system can be implemented in buildings and small areas. However, the functionality of this system was streamlined to environmental monitoring. This implies that although this system can be implemented in a building, it cannot control and manage other appliances or devices in that building.

Furthermore, in a paper titled "Automated Home Using Firebase and Google Assistant", an automatic IoT-based home system was developed which utilized both Firebase (which served as the cloud server) and Google Assistant (which served as a voice interface for user commands) [18]. This system allows Google Assistant to read device statuses from Firebase and give commands to control the devices; that is, the system allows for natural language commands to turn on/off devices [18]. This system leverages natural language processing to decode voice commands. However, the system also fails to include a global or central mechanism to turn off devices that the user can control manually. In addition, although the system incorporates an Artificial Intelligence (AI) tool (Google Assistant) and relies on it for voice commands, no device in the system operates autonomously. In other words, there is no automation feature like sensor-triggered responses. Inclusively, the system does not incorporate dedicated security monitoring functionality, preventing unauthorized access or potential breaches in the home.

Another study on home security and automation using NodeMCU shows a lack of a central mechanism to shut down the system [7]. Therefore, it can be deduced that the lack of a global or central mechanism to shut down all appliances, excluding the sensors, is a significant limitation in IoT systems incorporating NodeMCU and Firebase. As such, this study aims to develop an IoT system that addresses this limitation and other identified limitations. By overcoming these challenges, the IoT-based home control and automation system will provide users with easy control, comfort, improved security, convenience and energy management in their daily lives.

2. Materials and methods

2.1. System architecture

A system architecture is a model that shows the structure and blueprint of a system. The system architecture for the developed IoT system includes the units of the IoT system and shows the flow of communication between each unit, from the mobile application through the Firebase cloud server to the hardware unit of the IoT system and vice versa.

The architecture of the developed IoT system, as shown in figure 1, illustrates the system's three units: the mobile application (software unit), the Firebase cloud server, and the hardware unit. The architecture of the IoT system shows that in the hardware unit, relay modules serve as the actuators in the system. Additionally, sensors were utilized in the system. The sensors used are the DHT11 humidity and temperature sensor and the PIR motion sensor. DHT11 reads the ambient temperature and humidity values while the PIR sensor detects motion. The Firebase Realtime Database relays these detected and read parameters to the mobile app.

Moreover, NodeMCU, the microcontroller in this system, serves as the brain of the system, which processes the received data from the cloud server and also sends data to the cloud server. In addition, the appliances used in this system are a light bulb, which represents the home's lighting system, a multipurpose socket, a dedicated socket for the fan and a dedicated socket for the electric cooker. These appliances can be controlled remotely from the mobile application.

2.2. Methodology

A detailed explanation of the systematic approach and the methods that were employed in the development of the IoT-based home control system is given in this section. Additionally, this section explains the critical stages of the system's development and details the configuration of Firebase Realtime Database as a cloud-based backend. Figure 2 visualizes the methodology employed in developing the system.

The methodology, as shown in figure 2, illustrates that the development of the system originated from a circuit design. The designed circuit schematics determine

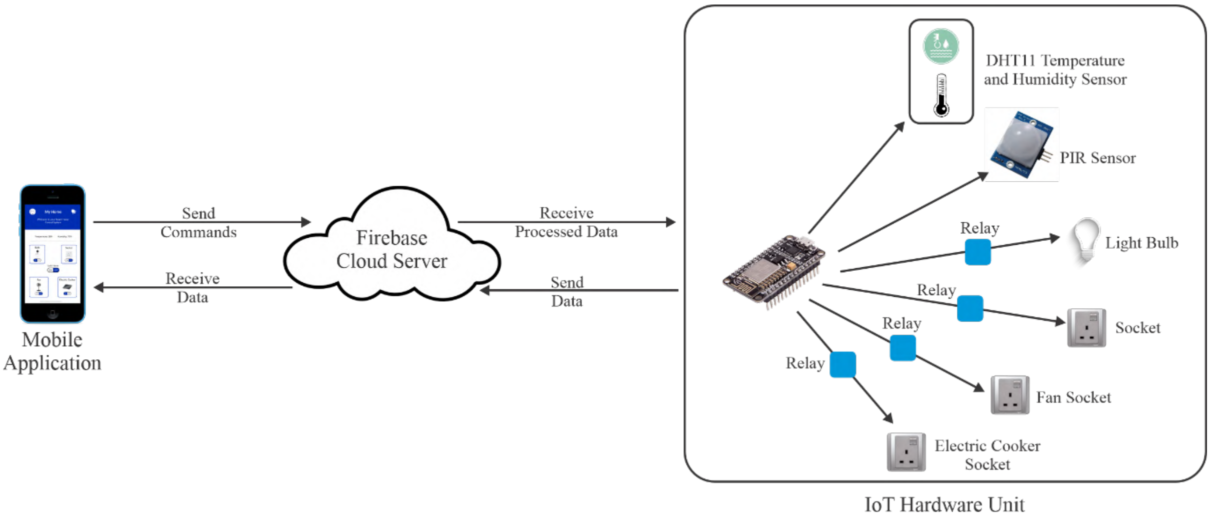

Figure 1: The architecture of the IoT system.

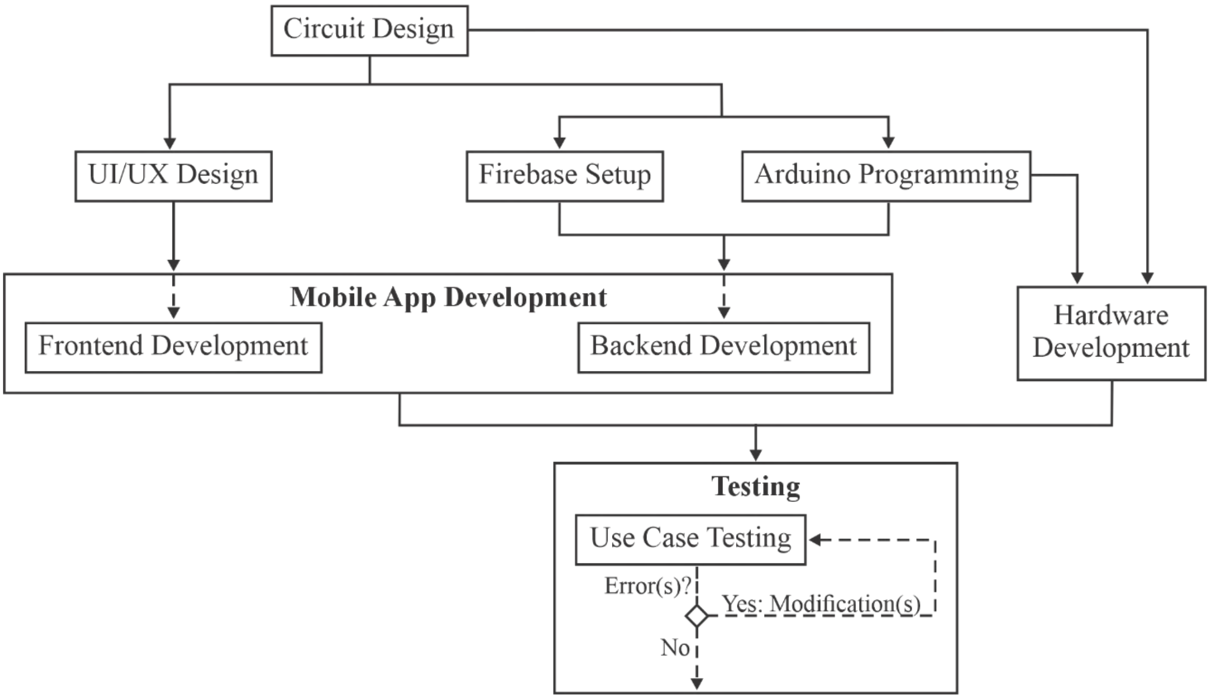

Figure 2: Study methodology diagram.

which features and components are needed in this study to develop the IoT system. In the same way, the User Interface and User Experience (UI/UX) design aid the front-end development of the mobile app, while the Firebase Setup and Arduino code snippet aid the back-end development of the mobile app. Figure 2 also depicts that the circuit design and Arduino code snippet are used for the hardware development phase. Upon the completion of the development of both the mobile app and the hardware unit, the system is tested and modified if necessary.

2.2.1. Circuit schematics design

Circuit schematic design is the process whereby an electrical or electronic circuit is visually illustrated using symbols that are connected and represent the components in the circuit. Although several software tools can be used in designing circuit schematics diagrams, Proteus was chosen for the circuit schematics design of this IoT system. Proteus is software used to design and simulate circuits and printed circuit boards (PCBs) [14]. Using Proteus software enhances learners' mathematical and critical thinking skills [10]. Factors influencing Proteus's choice include the availability of an extensive component library and the availability of downloadable libraries, a user-friendly interface, and a strong user community. Figure 3 shows the circuit schematics design of the IoT system in the Proteus interface.

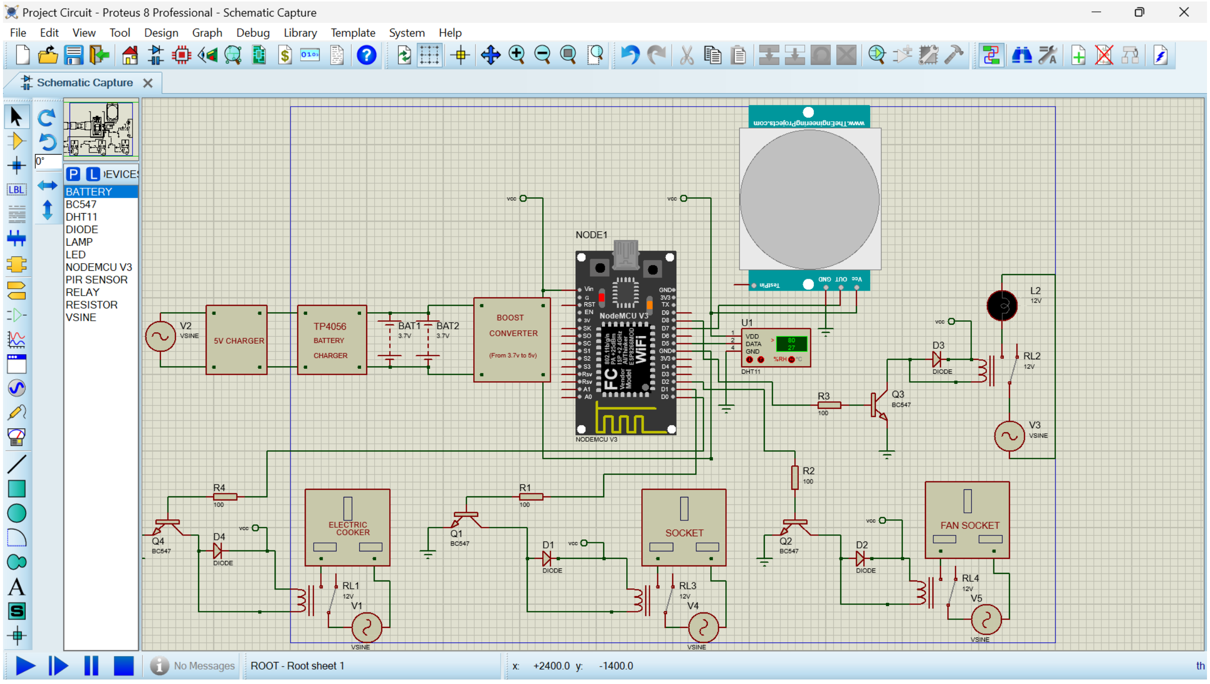

Figure 3: Circuit schematics diagram.

Figure 3 shows that the components which were used are a 5V charger, which is used for converting 220V AC power to 5V DC power; a TP4056 battery charger, which is used for charging the batteries; two 3.7V batteries, which are connected in parallel, boost converter which is used for boosting 3.7V output of the batteries to 5V, NodeMCU which is embedded with an ESP8266 chip and serves as the brain of the hardware unit, DHT11, Passive Infrared (PIR) sensor, light bulb, sockets, relays which are the actuators to switch the states of appliances, resistors, transistors which serve as the drivers for the relays, and diodes which serve as protection for the relay modules. Shapes were used for components that had no online or pre-installed libraries.

2.2.2. UI design

User Interface (UI) design is essential in developing software applications and other digital products because it shows how users interact with the system [3]. A good UI design makes applications look good and easy to navigate [3]. In other words, it could be stated that UI design is beneficial in developing the frontend of mobile applications. Figma was utilized for the UI design of the mobile application of the IoT system. Factors influencing this choice include accessibility from any device as it is cloud-based, easy to use, and has an intuitive interface. This study designed two user interfaces: the splash screen and the homepage.

2.2.3. Arduino code development

Arduino Integrated Development Environment (IDE) is an open-source software application that can run on different operating systems. It is written in C programming language [2, 8]. The Arduino programming language is a simplified version of C++ and was developed to program microcontrollers used on Arduino boards. Arduino IDE has numerous libraries which can be downloaded to improve its functionality. Although NodeMCU was designed to be programmed using the Lua scripting language, in this study, it was programmed with the Arduino programming language using the Arduino IDE because of its compatibility with C/C++. The reasons for choosing NodeMCU over other microcontrollers in this study are its Wi-Fi connectivity feature, low cost, and ease of use. The circuit design was helpful in the programming of the microcontroller because the General Purpose Input/Output (GPIO) pins to be used were indicated in the design. However, the NodeMCU pin has a mapping to the GPIO pins of the ESP8266 chip. This was taken in the pin definition of the Arduiof code to avoid compilation errors.

2.2.4. Mobile app development

The mobile application, My Home, was developed using Java programming on Android Studio to create a native Android application. The frontend consisted of a dashboard, icons, control buttons, and notifications. It allows users to interact with the cloud server and IoT devices. It fetches data from the Firebase real-time Database and updates the UI in real-time.

On the contrary, the backend of the mobile application is the server-side components that handle data processing, storage, authentication, and communication with other services. While the Android app runs on the user's device, the backend typically runs on remote servers and interacts with the app via APIs (Application Programming Interfaces). In the mobile app's backend, the Java program constituted of Firebase's (Software Development Kit) SDK to interact with the Firebase Real-time Database.

2.2.5. Hardware unit development

The hardware unit of the IoT system consists of several key components that work together and synchronise to control home appliances and monitor the home environment. From figure 3, the 220V AC power source is converted to 5V DC power using a 5V phone charger. A 5V charger was preferred over a step-down transformer to convert 220V AC power to 5V AC power and a rectifier to convert AC to DC because the 5V charger reduces the complexity of the circuit and is cost-effective. This 5V DC power was inputted into the TP4056 battery charger to charge two 3.7V batteries. These batteries were connected in parallel to improve their lifespan. Giving an output of 3.7V, this output voltage was boosted by a 5V boost converter. This boosts the output of 3.7V to 5V to power the NodeMCU board. This terminal for the +5V is also the VCC point of the circuit, and other components of VCC terminals were connected to this point in the circuit. Though the NodeMCU typically requires a 3.3V power supply, the 5V power supply is still in a safe range of power supply for the board.

Moreover, each appliance is connected to a relay that acts as the hardware unit's actuator. A transistor drives each relay. The NodeMCU controls these relays through its GPIO pins. The relay activates when a pin is set high, allowing current to flow to the connected appliance.

Two sensors were connected to the NodeMCU board: the DHT11 and PIR sensor. The DHT11 sensor has some limitations in accuracy and response time, so it is only suitable for basic applications rather than high-precision tasks. In terms of temperature accuracy, DHT11 has an accuracy range of $\pm 2°$C when the temperature falls between $0°$C and $50°$C. Inclusively, regarding humidity accuracy, the sensor has an accuracy range of $\pm 5\%$ when the relative humidity value falls between 20% and 90%. Additionally, the sensor's response time can be up to 6 seconds for adjustments to sudden environmental changes.

In summary, the DHT11 sensor is best suited for indoor applications. On the other hand, PIR sensors detect changes in infrared radiation from moving objects, and they can detect motion up to 12 meters. However, this range may vary depending on the specific model. In addition, most PIR sensors can detect motion within $110°$ to $180°$. As a result, PIR sensors are always focused on a specific area. However, PIR sensors are susceptible to false positives due to environmental factors and electromagnetic interference, so indoor usage will also be most appropriate for the sensor.

The VCC pin of the DHT11 is connected to the +5V power supply (VCC), and the Ground (GND) pin is connected to the GND of the NodeMCU, while the data pin of the DHT11 is connected to one of the GPIO pins. The VCC pin of the PIR sensor is also connected to the +5V power supply (VCC), and the GND pin is connected to the GND pin of the NodeMCU, while the output pin is connected to D7 on the NodeMCU board. Furthermore, four appliances were connected to the NodeMCU board via relays serving as switches. These four appliances are bulb, fan, socket and cooker. Each relay module has a control pin connected to a GPIO pin on the NodeMCU. These relay modules were mounted on the Veroboard, and the 100 Ohms resistors and the BC547 transistors were also arranged as shown in the circuit diagram in figure 3. Figure 4 shows the circuit connection on the Veroboard. The final IoT system was thoroughly tested.

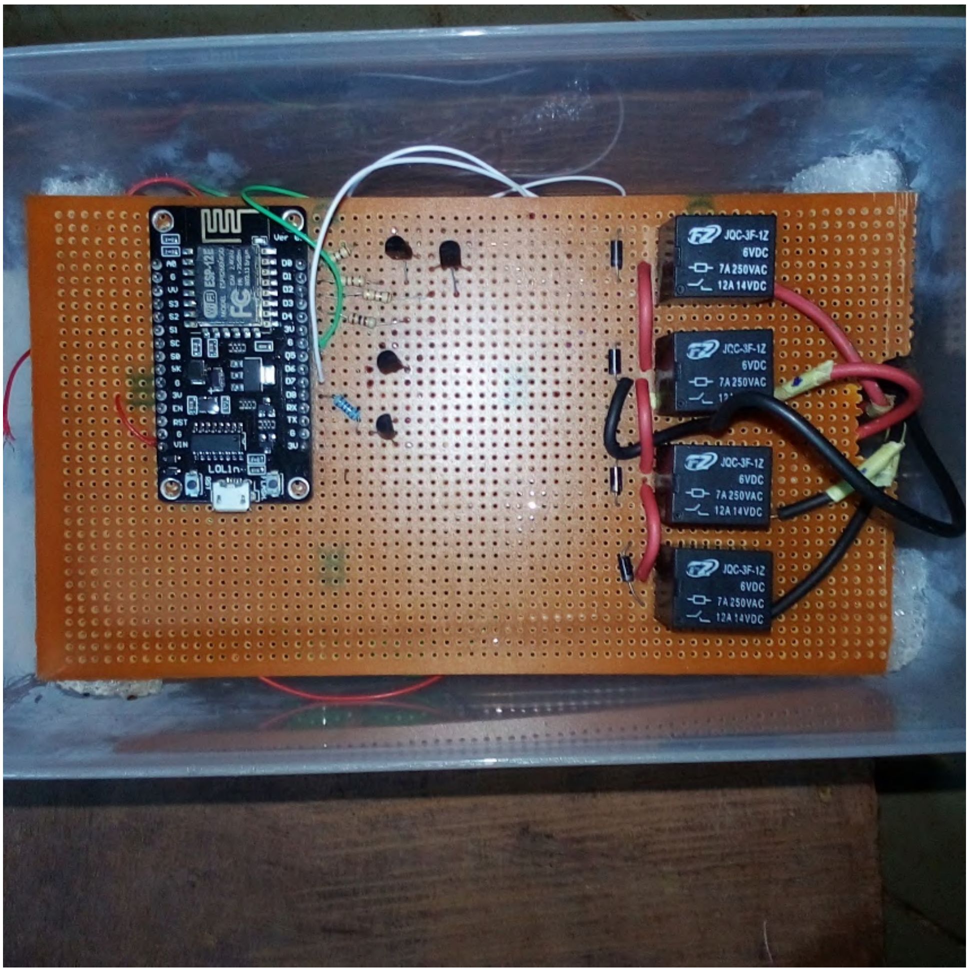

Figure 4: Circuit layout on Veroboard.

3. Results

3.1. System testing

The tests conducted on the IoT system can be classified into four major parts: software tests (mobile app tests), hardware tests, cloud server tests, and tests combining all three units. Software testing is the primary way of verifying that a software product meets its requirements, and about half of the software development and half of the development costs are allocated to software testing [6]. It ensures the completeness of the software product in terms of features, the correctness of the system in terms of functionalities and the overall quality of the software system. Testing was done at every phase of the software development process of this study to ensure it meets its specified functionalities. Figures 5, 6, and 7 are the results obtained after testing the mobile app. Figure 5 shows the mobile app's interface when the bulb is activated. Figure 6 shows the mobile app's interface when the "Lock Home" button is activated. Figure 7 shows the mobile app's interface when motion is detected while the home is locked.

Figure 7 illustrates that while a user is on the app, a notification interface appears when motion is detected. In addition to this feature, another pop-up notification displays on the notification tab of the mobile phone. This pop-up notification indicates that even when the user is not using the My Home app, they will receive notifications of any home movements. At the same time, their phone is connected to the internet. Furthermore, several other test cases were also used in the testing process of the My Home mobile app. These are discussed in table 1.

In addition to these, the system's responsiveness was also tested. The system's responsiveness was done in two phases: the time taken for the app's commands to reflect on the Firebase Realtime Database and for the hardware to execute commands from the Firebase Realtime Database. These two phases of responsiveness testing were conducted with internet speed considered. Figure 8 illustrates the speed of sending commands from the mobile app to Firebase. Figure 9 illustrates the hardware unit's responsiveness to the Firebase command. The vertical axis represents the time in seconds. Each coloured line represented the internet upload speed in figure 8 and download speed in figure 9 when three tests were carried out for the app's responsiveness and the hardware responsiveness, respectively. These internet speed tests were conducted online via https://www.speedtest.net.

The system was also tested under unstable communication conditions. The network bandwidth was limited on the router's admin page to implement this, and network connectivity disconnections were made frequently. In the case where the bandwidth of the network was limited, the command sent from the mobile app was processed and executed accurately but with delays. On the other hand, when there is a network connectivity disconnect, commands are not executed, as the hardware system relies on internet connectivity to write data and read commands from the Firebase cloud server. When communication is lost between the hardware system and the Firebase cloud server, the devices in the hardware system retain their respective last-known states until the connection is restored. Additionally, during this period of lost communication, commands sent from the mobile app are stored and updated on Firebase.

Moreover, the developed IoT-based home control system presents a reliable performance in its various features and functionalities, ensuring smooth and seamless control of household appliances and monitoring environmental conditions through a mobile app over the internet. This system integrates hardware components coupled into a hardware unit, Firebase Realtime Database serving as the cloud server, and a mobile application that can be used to manage the home locally and remotely. The system performance overview is presented in table 2.

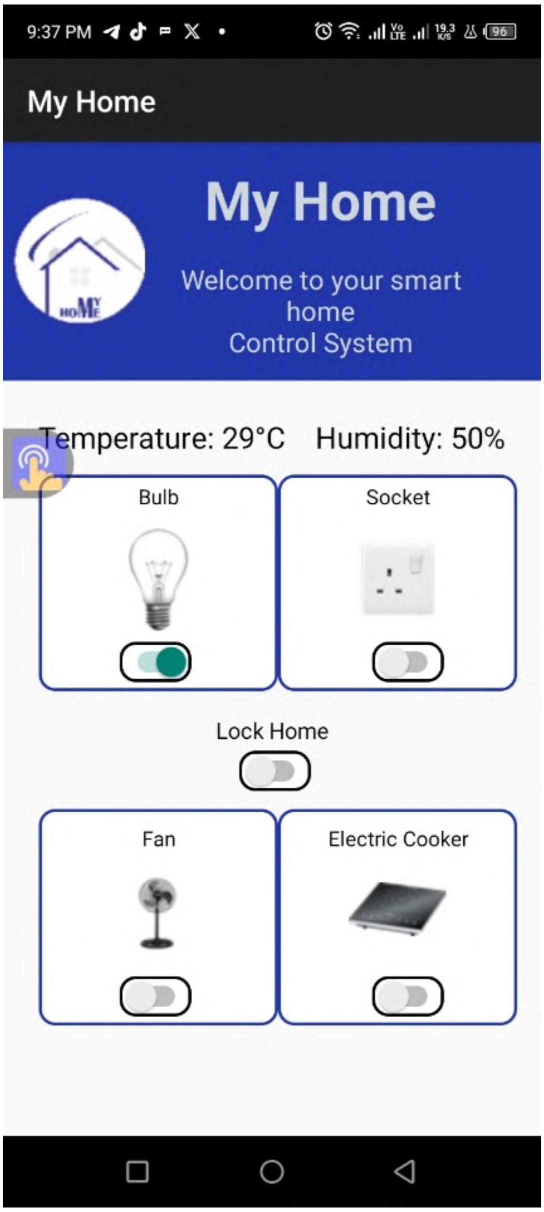

Figure 5: App interface display showing bulb activation.

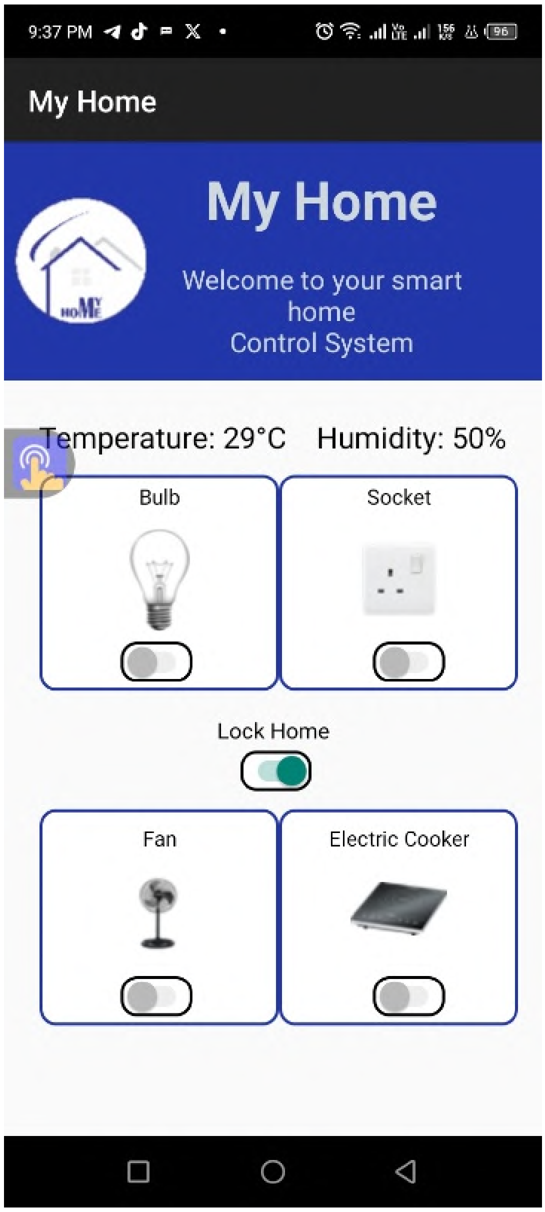

Figure 6: App interface display when Lock Home is activated.

In summary, it can be inferred from table 2 that the IoT-based home control system has a strong performance across various functionalities. The system's ability to control individual and multiple appliances in the case of Lock Home, manage security through motion detection and monitor ambient environmental parameters shows that it can be utilized in real-world scenarios. In addition, integrating the Firebase cloud server into the system improves its versatility and scalability, making it an effective solution in smart home management systems. Figure 10 shows the complete hardware unit of the IoT system with the fan and cooker sockets being activated on the mobile app.

4. Discussion

Table 1 shows that the mobile app meets most of its requirements, except for responsiveness in slow and unstable network situations, simultaneous multi-device usage due to inadequate user authentication, and adaptive orientation. In addition, knowing that the system experiences minimal delay, this little delay in communication

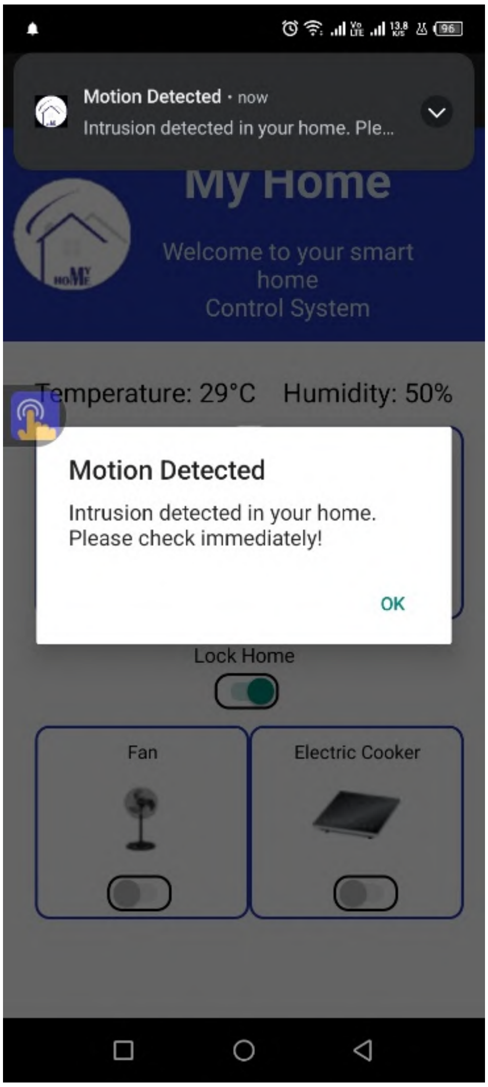

Figure 7: Motion detection display on application interface.

must also be discussed. Several reasons could have caused this delay. One of these reasons is Wi-Fi connection latency. This refers to the delay in data transmission over a wireless network. NodeMCU heavily depends on a Wi-Fi connection to communicate with Firebase. Therefore, weak signals and network congestion can cause delays in sending or receiving data. Wi-Fi connection latency can be a result of network traffic whereby multiple devices are connected to the network simultaneously or a considerable distance between the router and the NodeMCU board, which leads to weaker signal or interference from nearby devices that emit radio frequencies (such as microwaves and Bluetooth devices); or low network hardware quality which may not be able to handle communication effectively. In this study, the first two cases of probable causes for latency in Wi-Fi connection are dismissed. This is because the Wi-Fi router was dedicated to the NodeMCU board only during testing, and as such, there is insignificant network traffic, and the distance between the router and NodeMCU board was less than a meter during testing, with a recommendation for maximum connection quality falling in the range of 20 meters for a 2.4 GHz band router. This implies that the most likely causes of Wi-Fi latency are interferences or network hardware quality. In this study, a 2.4 GHz band router was used, which is more susceptible to interferences because it has a broader wavelength, and the router

Table 1
Outcomes of mobile app test cases.

№	Test cases	Results
1	Verification that the mobile app installs successfully on an Android device with an Android version of Android 11 and above.	This test matched the expected results; the mobile app was successfully installed on an Android device with an Android 11 and another device with a higher Android version.
2	Verification that the mobile app launches seamlessly on devices with different screen sizes.	The mobile app launched without glitches, smoothly transitioning from the splash screen to the home page on devices with different screen sizes.
3	Verification that the contents on the home page are well arranged and organized.	The contents on the page were well organized.
4	Verification that all buttons function correctly.	The five buttons were tested and it was found that they all functioned properly. Figures 5 and 6 show two of the visual results.
5	Verification that the app is responsive to both portrait and landscape orientations and adjusts correctly without losing functionality.	However, the app was not responsive to landscape orientation. The app did not adjust well when switching orientation from portrait to landscape.
6	Verification that the app does not send data to the cloud server while the mobile device is not connected to the internet.	The app could not communicate with the cloud server without internet connectivity, which was the expected result.
7	Verification that the IoT devices respond to commands from the app.	The result proved positive, implying that the state of IoT devices was switched when a command was sent from the app.
8	Verification that the sensor data are displayed correctly on the app and in real-time.	The sensor data values were displayed in the right position and updated in real-time as the environment parameters were altered.
9	Verification that notification is displayed on the app and the mobile device when any motion is detected while the home is locked.	The app's notification interface and the mobile device's notification tab were displayed when motion was detected. This result is shown visually in figure 7. This notification pops up only when the home is locked.
10	Verification that multiple users can simultaneously update the states of IoT devices using different mobile devices.	However, this yielded a negative result because there was no authentication for different users, so the app on both devices crashed repeatedly.
11	Verification that the app communicates with the cloud server while using the device's mobile data and while connected to a network with internet connectivity.	In both scenarios, the app communicated successfully with the cloud server, reading data and writing commands to the cloud server.
12	Verification that the app is very responsive during slow and unstable networks.	However, during unstable network conditions, the app delays sending commands to the cloud server and reading data from the cloud server.
13	Verification that the notification interface on the app can be cleared off by tapping the "ok" button.	This also yielded a positive result; the notification interface was cleared off when the "ok" button was pressed.

model used was a 2022 model. For optimisation of the connectivity of the IoT system: a 5 GHz band router can be utilised, or a newer model of router can be employed; or the 2.4 GHz router can be placed away from devices that emit electromagnetic signals

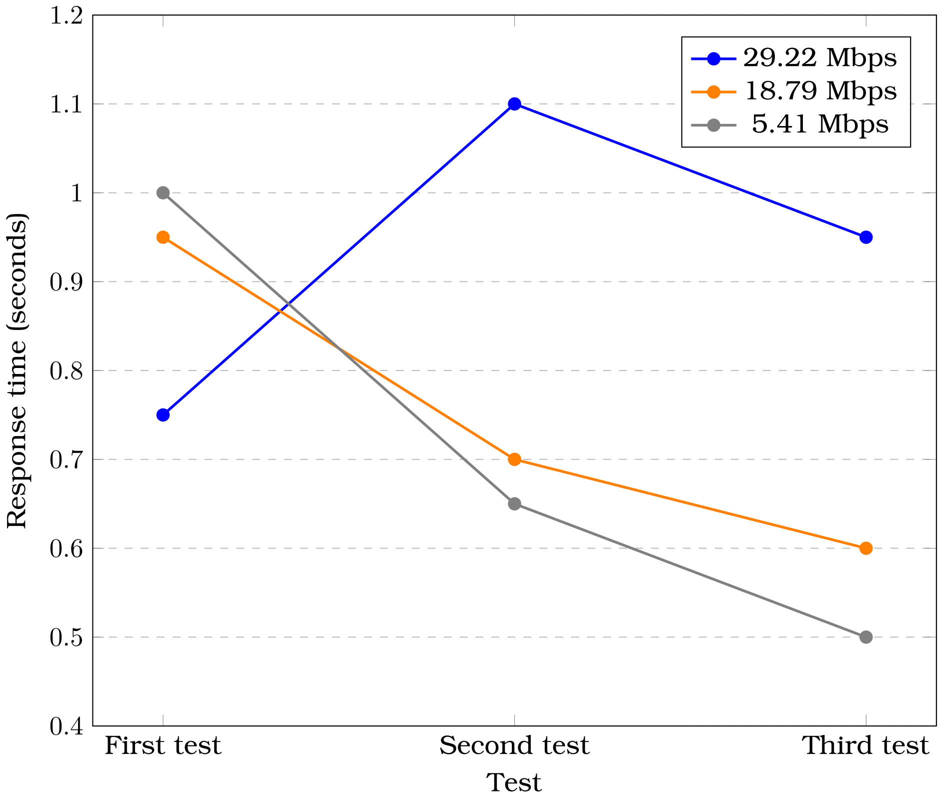

Figure 8: Command transmission speed from app to Firebase.

or radio waves, or legacy 802.11b mode, which can introduce interference, is disabled in the router's settings, and the router is restricted to either 802.11g or 802.11n modes which has higher speeds and is supported by NodeMCU microcontroller.

Another possible reason for the delay in executing commands is Firebase communication overhead. Firebase is a cloud-based platform. This implies that commands from the mobile app must first travel to the Firebase server and then to the NodeMCU microcontroller and vice versa for environmental parameters. The Firebase remote server selected for this study is in the United States. This round-trip time to and from the Firebase remote server also introduces delay. Additionally, knowing that the processes involved in authenticating with Firebase and exchanging data involve multiple steps and data is constantly being read and written to Firebase, these can also introduce delays. To optimise the IoT system, local data caching on the NodeMCU can aid in quicker access to frequently used data. Significant data, like commands, can be sent to Firebase immediately, while non-critical data, such as environmental parameters, can be sent at scheduled intervals. Additionally, the Firebase remote server, which is closest to the location of the IoT system, can be selected.

Additionally, a probable reason for the delay is the limited processing power of NodeMCU. NodeMCU is a low-power microcontroller with limited processing capability, and when it handles multiple tasks such as connecting to Wi-Fi, reading sensor data, and interacting with Firebase, the device might have issues processing everything quickly, which may eventually lead to delays. To optimise the system in this case, scheduled time intervals could be allocated for operations of the NodeMCU microcontroller to avoid multiple simultaneous operations being done by the microcontroller.

The reliability of a system is an essential factor in assessing its overall performance. Reliability is a system's ability to consistently perform its intended functions under specified conditions for a particular period without failure. For this IoT system,

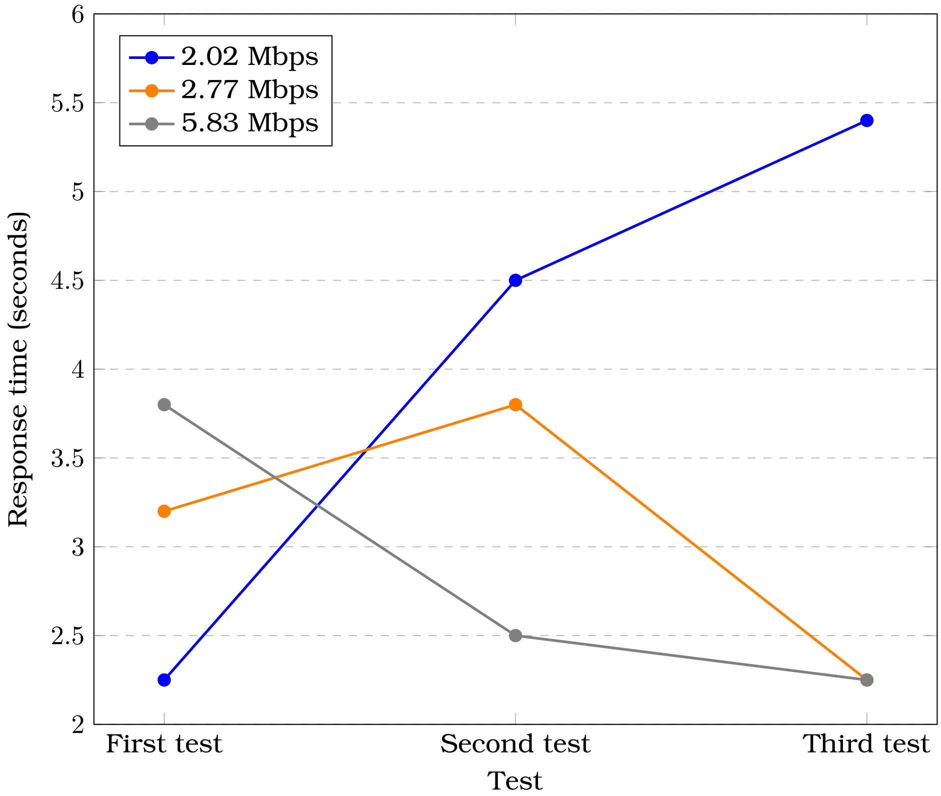

Figure 9: Command execution speed on hardware from Firebase.

Figure 10: Fully assembled hardware unit with activated fan and cooker sockets.

reliability covers several sections, such as the stability of the hardware components, the software's robustness, the dependability of network connectivity, etc. To ensure the reliability of the hardware components, the components were tested under various load conditions to ensure they could handle the power requirements of the connected appliances. It was discovered that the relay module would not be able to drive any appliance with a rating above seven amps to avoid malfunction or failure of the system. However, this system is a prototype to illustrate the functionality of an IoT-based home control system. Therefore, any appliance above a rating of 7 amps, like the

Table 2
System performance overview.

Features	Functionality	Performance
Individual appliance control	The system allows users to control individual appliances through the mobile app. These appliances are connected to relays which switch the states of the appliances.	Each relay responds to a command from the app, with each appliance turning on or off as expected. However, the system experiences minimal delay in the execution of their commands.
Global appliance shutdown (Lock Home)	The "Lock Home" feature on the app allows users to shut down all appliances simultaneously. When this button is pressed, the system sets all relays to the off position, ensuring that all connected appliances are turned off.	The system promptly turns off all appliances when receiving the lock command. This shows the system's ability to handle simultaneous relay operations without issues. This feature is handy in cases where the user needs to secure the home quickly.
Motion sensor activation	The motion sensor is programmed only to send signals when the "Lock Home" button is activated. This ensures that motion detection is only active when the home is secured.	The motion sensor successfully sends signals only when the home is locked. This activation method helps in reducing false alarms.
Intrusion detection	When the motion sensor detects movement while the home is locked, the system sends a notification to the Firebase database. This notification is then relayed to the mobile application, notifying the user.	The motion sensor detects movement and triggers the intrusion notification without delay. This specific feature improves the security aspect of the system, thereby ensuring that users are informed of any unauthorized movement.
Temperature and humidity monitoring	The DHT11 sensor, which is integrated into the system, monitors ambient parameters, which are temperature and humidity, and it sends these data to the Firebase database. The mobile app then retrieves these data and displays them to the user on the mobile app.	The sensor provides readings, which are updated regularly on the app. The system maintains a consistent data flow from the sensor to the app.

electric cooker, will not be connected to the system. The reliability of the sensors was also examined. DHT11 was tested under a range of temperature conditions ranging from 20°C to 45°C, including the proper room temperature range. It was discovered that the DHT11 sensor is reliable and effective within this temperature range. Also, DHT11 was tested under different humidity environments ranging from 20% to 75%, encompassing the appropriate indoor humidity range. It was discovered that the sensor functioned accurately and efficiently in these conditions. This implies that DHT11 is reliable in indoor environmental conditions. Inclusively, the motion sensor was tested, and it reliably detected motion when the home was locked and accurately triggered the notification. In addition, two 3.7V batteries were added to the system to ensure the system functions in cases of power outages. These batteries also ensure the system's reliability in non-predictive electricity situations, making the IoT system suitable for environments with unstable power supply.

In addition to the reliability of the hardware components, the system's software reliability was also examined. Knowing that the system's functionality is heavily dependent on the Arduino code that controls the hardware components and manages communication with Firebase, it was ensured that the code was free from bugs and errors. This is to optimise the system's performance and prevent crashes or unexpected behaviour during operation. Inclusively, the reliability tests of the mobile app were carried out. Furthermore, network connectivity reliability was observed. In cases where the connection is lost, the Arduino code includes a mechanism which allows reconnection and ensures continuous operation.

Several existing smart home systems exist, including Google Nest and Amazon Alexa. These are commercial IoT platforms. A comparison of the features of the developed IoT system and these commercial IoT platforms is given in table 3. It can be inferred from table 3 that the developed IoT system, built using NodeMCU and integrated with Firebase for real-time database management, offers essential features in the home control system, providing features that are also similar to the features in commercial IoT platforms. Additionally, the developed IoT system integrates some specific features that some commercial IoT platforms do not have. Some commercial IoT platforms are exclusively integrated with specific features and can only work with compatible devices. However, the developed IoT system can be expanded, scaled and modified to suit individual requirements.

From table 3, it can be deduced that although commercial IoT platforms are more sophisticated than the developed IoT system, the developed system has some functionalities and features not incorporated into the commercial platform. Both Google Nest and Amazon Alexa can only function in their ecosystem. However, the developed system is versatile and can be customized for several other devices, and as such, it is scalable. This scalability can be achieved by additional coding. Additionally, there are variations in security features. Both Google Nest and Amazon Alexa can control door locks. However, they do not include a feature to centrally deactivate all connected devices to the IoT systems, excluding the sensors. This implies that in cases of utmost urgency, the user will have to shut down the entire system or deactivate each device one after the other.

Although the developed IoT system has essential features for home control and management, there are possible technical issues that can be encountered if the system is implemented in real-world conditions. One of these possible technical issues is power fluctuation and outages. Frequent power interruptions can lead to a disruption of the connection between the NodeMCU and Firebase. An Uninterruptible Power Supply (UPS) can be implemented into the system to avoid this. Additionally, due to the limited processing power of NodeMCU, the system may struggle with complex operations and loads of operations in cases where the system has been scaled up. To avoid this issue, complex operations can be offloaded to the Firebase cloud server for processing, and only the results are retrieved by the NodeMCU. High Wi-Fi latency can also be a possible technical issue in real-world scenarios, and as such, routers should be optimized for maximum network connectivity quality.

5. Conclusion

This study successfully demonstrates the development of an IoT-based home control system that integrates a NodeMCU board with Firebase Real-time Database and a mobile application. The system was designed to provide real-time data monitoring and control of household appliances to ensure users' convenience and the home's security. Furthermore, the Firebase Real-time Database was the cloud server for storing and retrieving data. It can be deduced from this that using Firebase as a cloud server is a reliable technology for real-time data management. The developed mobile application also served as a homeowner's user interface to control home appliances locally or remotely. It was portrayed that users can lock and unlock their homes, turn appliances on or off and monitor environmental conditions. This smooth integration with the

Table 3
Comparison with commercial IoT platforms.

Features	**The developed system**	**Google Nest**	**Amazon Alexa**
Platform integration	It is custom-built with Firebase integration for real-time data management.	It works within Google's ecosystem and is integrated with Google Assistant and other Google services.	It only works within Amazon's ecosystem, and Alexa voice control is integrated.
Device control	It controls specific home appliances, such as bulbs, sockets, fans, and electric cookers, through a mobile app.	It controls Google Nest thermostats, cameras, doorbells, etc., through the app and voice command.	It controls smart home devices like lights, thermostats, and locks through Alexa-enabled devices.
Lock Home Feature	It provides a "Lock Home" button to turn off all appliances centrally.	It can control door locks and set even home/away modes. However, it is not specific to all appliances at once.	It can also lock doors and control security systems, but not all appliances simultaneously.
Motion detection	Motion sensor becomes active only when "Lock Home" is activated and sends an "Intrusion detected" notification.	Nest cameras can detect motion and send notifications to the app.	Alexa-compatible cameras and sensors can also detect motion and send notifications.
Temperature and humidity monitoring	It uses a DHT11 sensor to monitor temperature and humidity readings.	Nest Thermostat monitors and controls temperature, but no humidity sensor.	It is compatible with several third-party sensors for temperature and humidity.
Mobile app development	It consists of a mobile app developed using Java on Android Studio.	Google Nest ecosystem consists of an app developed by Google for Android and iOS.	Amazon Alexa app is also available for download for Android and iOS devices.
Cloud integration	It uses Firebase Realtime Database for data storage and real-time updates.	It uses Google Cloud for data storage and device management.	It uses Amazon Web Services (AWS) for cloud-based functions.
Customization and scalability	It is a custom-built system specific to the provided hardware and code, but it can be expanded with additional coding.	It is limited to Nest and compatible third-party devices.	It supports various compatible devices, with Alexa skills for custom actions.

Firebase cloud server ensured that any changes made in the app were reflected in the hardware system. Therefore, it can be inferred that the IoT system manages data efficiently and has minimal delay in executing commands. Also, the system included the "Lock Home" feature, which turns all appliances off when activated, and the motion sensor becomes activated to detect intrusion. Therefore, it is concluded that the IoT system enhances the security of the home environment by detecting intrusions and notifying homeowners in real time. Conclusively, the developed IoT system successfully integrates hardware components with cloud services and a mobile application, allowing users to manage their home environment from anywhere. It offers primary and essential functionalities in the home, provides a simple mobile app with an intuitive interface that allows easy interaction between users and the system, and communicates data between the mobile app and the hardware system in real time over the Firebase cloud server. Therefore, this study shows the benefits of IoT in home management.

Although the developed IoT system is fully functional, several improvements could be considered for future work. These include less dependence on internet connectivity, implementation of advanced security features, integration of artificial intelligence and machine learning, and integration of voice control features. Additionally, future work can explore other means of communication, such as satellite communication. Inclusively, future work should aim to significantly reduce the delay in the execution of commands.

References

[1] Aliero, M.S., Qureshi, K.N., Pasha, M.F. and Jeon, G., 2021. Smart Home Energy Management Systems in Internet of Things networks for green cities demands and services. *Environmental Technology & Innovation*, 22, p.101443. Available from: https://doi.org/10.1016/j.eti.2021.101443.

[2] García-Tudela, P.A. and Marín-Marín, J.A., 2023. Use of Arduino in Primary Education: A Systematic Review. *Education Sciences*, 13(2), p.134. Available from: https://doi.org/10.3390/educsci13020134.

[3] Hamidli, N., 2023. Introduction to UI/UX Design: Key Concepts and Principles. Available from: https://www.academia.edu/98036432/Introduction_to_UI_UX_Design_Key_Concepts_and_Principles.

[4] Islam, R., Rahman, M.W., Rubaiat, R., Hasan, M.M., Reza, M.M. and Rahman, M.M., 2022. LoRa and server-based home automation using the internet of things (IoT). *Journal of King Saud University - Computer and Information Sciences*, 34(6, Part B), pp.3703–3712. Available from: https://doi.org/10.1016/j.jksuci.2020.12.020.

[5] Joha, M.I. and Islam, M.S., 2021. IoT-Based Smart Home Automation Using NodeMCU: A Smart Multi-Plug with Overload and Over Temperature Protection. *2021 24th International Conference on Computer and Information Technology (ICCIT)*. pp.1–6. Available from: https://doi.org/10.1109/ICCIT54785.2021.9689913.

[6] Lima, R., Cruz, A.M.R. da and Ribeiro, J., 2020. Artificial Intelligence Applied to Software Testing: A Literature Review. *2020 15th Iberian Conference on Information Systems and Technologies (CISTI)*. pp.1–6. Available from: https://doi.org/10.23919/CISTI49556.2020.9141124.

[7] Lokesh, S., Patil, S.B. and Gugawad, A., 2020. Home Security And Automation Using NodeMCU-ESP8266. *2020 IEEE Bangalore Humanitarian Technology Conference (B-HTC)*. pp.1–6. Available from: https://doi.org/10.1109/B-HTC50970.2020.9297917.

[8] Lu, C., Hong, J., Chen, F. and Ma, S., 2020. Elementary school students learn Arduino programming to assemble Sensory-Controlled works. *International Journal of Information and Education Technology*, 10(4), pp.265–270. Available from: https://doi.org/10.18178/ijiet.2020.10.4.1374.

[9] Malhotra, P., Singh, Y., Anand, P., Bangotra, D.K., Singh, P.K. and Hong, W.C., 2021. Internet of Things: Evolution, Concerns and Security Challenges. *Sensors*, 21(5), p.1809. Available from: https://doi.org/10.3390/s21051809.

[10] Matsun, M., Boisandi, B., Sari, I.N., Hadiati, S. and Hakim, S.L., 2021. Use of Arduino Microcontroller and Proteus Software in Physics Lesson in Review of Mathematics Ability and Critical Thinking Skills. *Jurnal Penelitian Pendidikan IPA*, 7(SpecialIssue), p.20–27. Available from: https://doi.org/10.29303/jppipa.v7iSpecialIssue.916.

[11] Mouha, R., 2021. Internet of Things (IoT). *Journal of Data Analysis and Information Processing*, 9(2), p.77–101. Available from: https://doi.org/10.4236/jdaip.2021.92006.

[12] Nikam, S. and Kulkarni, S., 2021. IOT Based Environmental Monitoring System. *International Journal of Engineering Research & Technology (IJERT)*, 10(6), pp.845–848. Available from: https://doi.org/10.17577/IJERTV10IS060401.

[13] Nižetić, S., Šolić, P., López-de-Ipiña González-de-Artaza, D. and Patrono, L., 2020. Internet of Things (IoT): Opportunities, issues and challenges towards a smart and sustainable future. *Journal of Cleaner Production*, 274, p.122877. Available from: https://doi.org/10.1016/j.jclepro.2020.122877.

[14] Saleh, A.L., Obed, A.A., Hassoun, Z.A. and Yaqoob, S.J., 2020. Modeling and Simulation of A Low Cost Perturb & Observe and Incremental Conductance MPPT Techniques In Proteus Software Based on Flyback Converter. *IOP Conference Series: Materials Science and Engineering*, 881(1), p.012152. Available from: https://doi.org/10.1088/1757-899X/881/1/012152.

[15] Salih, K.O.M., Rashid, T.A., Radovanovic, D. and Bacanin, N., 2022. A Comprehensive Survey on the Internet of Things with the Industrial Marketplace. *Sensors*, 22(3), p.730. Available from: https://doi.org/10.3390/s22030730.

[16] Shehu Yalli, J., Hilmi Hasan, M. and Abubakar Badawi, A., 2024. Internet of Things (IoT): Origins, Embedded Technologies, Smart Applications, and Its Growth in the Last Decade. *IEEE Access*, 12, pp.91357–91382. Available from: https://doi.org/10.1109/ACCESS.2024.3418995.

[17] Sivapriyan, R., Rao, K.M. and Harijyothi, M., 2020. Literature Review of IoT based Home Automation System. *2020 Fourth International Conference on Inventive Systems and Control (ICISC)*. pp.101–105. Available from: https://doi.org/10.1109/ICISC47916.2020.9171149.

[18] Tripathy, S. and Das, S.S., 2022. Automated home using firebase and Google assistant. *Journal of Information and Optimization Sciences*, 43(5), pp.1021–1028. Available from: https://doi.org/10.1080/02522667.2022.2093437.

[19] Wang, J., Lim, M.K., Wang, C. and Tseng, M.L., 2021. The evolution of the Internet of Things (IoT) over the past 20 years. *Computers & Industrial Engineering*, 155, p.107174. Available from: https://doi.org/10.1016/j.cie.2021.107174.

[20] Yar, H., Imran, A.S., Khan, Z.A., Sajjad, M. and Kastrati, Z., 2021. Towards Smart Home Automation Using IoT-Enabled Edge-Computing Paradigm. *Sensors*, 21(14), p.4932. Available from: https://doi.org/10.3390/s21144932.