**AR BASED HOME AUTOMATION APPLICATION**

## **A MINI-PROJECT REPORT**

***Submitted by***

| **DIVYAPRASATH R**  **GOKKUL PAWAN**  **JOBIN JOHN ABRAHAM** | **(200701068)**  **(200701074)**  **(200701506)** |
| --- | --- |

***in partial fulfillment of the award of the degree of***

# **BACHELOR OF ENGINEERING**

***in***

**COMPUTER SCIENCE AND ENGINEERING**



**RAJALAKSHMI ENGINEERING COLLEGE,**

**ANNA UNIVERSITY : CHENNAI 600 025**

# **MAY 2023**

**RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI**

## **BONAFIDE CERTIFICATE**

Certified that this project **“AR BASED HOME AUTOMATION APPLICATION”** is the bonafide work of **“JOBIN JOHN ABRAHAM (200701506), GOKKUL PAWAN (200701074), DIVYAPRASATH R (200701068)”** who carried out the project work under my supervision.

**SIGNATURE**

**Dr. N. THILAGAVATHI** M.E., Ph.D.,

SUPERVISOR

Assistant Professor

Department of Computer Science and

Engineering

Rajalakshmi Engineering College

Chennai - 602 105

Submitted for the **ANNA UNIVERSITY** practical examination Mini-Project work viva voce held on \_\_\_\_\_\_\_\_

### **INTERNAL EXAMINER EXTERNAL EXAMINER**

**ABSTRACT**

Internet of things (IoT) is the game changer by providing connectivity among smart devices equipped with sensors and actuators to make connected intelligence a reality. Modern technology like augmented reality (AR) is pioneering human–machine interaction by providing a better user interface/experience (UI/UX) for various IoT applications. In this project, we have proposed AR framework for home automation and telemetry application with IoT. A complete system prototype has been designed and developed to provide end user a new AR UX in smart home with Unity™ AR android app. The system would be able to recognize the interactive object, like light, fan, and air-conditioner, and give their controls live on a 3D framework. Target object is trained in advance to recognize the type of home appliance to provide an AR control panel with preconfigured options of device control and sensor data visualization. The real-time user input from AR console is sent to a cloud server deployed on a NodeMCU. The NodeMCU connected to these home appliances to fetch sensor data for telemetry application and control signals for home automation scenario. Proposed system is completely scalable and hence can be customizable and trained for various home appliances. users can control their devices using intuitive interactions. allow users to customize the application's interface and settings to their individual preferences. Overall, the proposed AR-based home automation application offers a novel and exciting way of controlling smart home devices that is both intuitive and efficient. With its range of advanced features and immersive AR interface, the application has the potential to revolutionize the way we interact with our smart homes.

## **ACKNOWLEDGEMENT**

First, we thank the almighty god for the successful completion of the project. Our sincere thanks to our Chairman **Mr. S. Meganathan, B.E., F.I.E.,** for his sincere endeavor in educating us in his premier institution. We would like to express our deep gratitude to our beloved Chairperson **Dr. Thangam Meganathan, Ph.D.,** for her enthusiastic motivation which inspired us a lot in completing this project and Vice Chairman **Mr. Abhay Shankar Meganathan**, **B.E., M.S.,** for providing us with the requisite infrastructure. We also express our sincere gratitude to our college Principal, **Dr. S.N. Murugesan M.E., PhD.,** and **Dr. P. Kumar M.E., Ph.D., Head of the Department of Computer Science and Engineering** and our project guide **Dr. N. Thilagavathi, M.E., PhD.,** for her encouragement and guiding us throughout the project and to our parents, friends, all faculty members and supporting staffs for their direct and indirect involvement in successful completion of the project for their encouragement and support.

**JOBIN JOHN ABRAHAM**

**GOKKUL PAWAN**

**DIVYAPRASATH R**

## 

## **TABLE OF CONTENTS**

| **CHAPTER NO. TITLE** | **PAGE** |
| --- | --- |
| **ABSTRACT** |  |
| **LIST OF FIGURES**  **1 INTRODUCTION** | **8** |
| 1.1 INTRODUCTION | 8 |
| 1.2 SCOPE OF THE WORK | 8 |
| 1.3 PROBLEM STATEMENT | 8 |
| 1.4 AIM AND OBJECTIVES OF THE PROJECT | 9 |
| **2 LITERATURE SURVEY** | **10** |
| **3 SYSTEM ARCHITECTURE**  3.1 EXISTING SYSTEM  3.2 PROPOSED SYSTEM  3.2.1 ADVANTAGES  3.3 WORKING STEPS  3.4 PROPOSED SYSTEM ARCHITECTURE  3.5 ARCHITECTURE DIAGRAM  3.6 DEVELOPMENT ENVIRONMENT  3.6.1. SOFTWARE REQUIREMENTS  3.6.2. HARDWARE REQUIREMENTS  3.7. DESIGN OF ENTIRE SYSTEM  3.7.1 USE CASE DIAGRAM  3.7.2 ACTIVITY DIAGRAM  3.7.3 FLOW CHART  3.7.4 SEQUENCE DIAGRAM  3.7.5 CLASS DIAGRAM | **13**  13  13  13  13  14  15  16  17  17  18  18  18  19  21  17  22  24 |
| **4 IMPLEMENTATION** | **26** |
| 4.1 MODULES  **5 RESULT**  5.1 OUTPUT  **6 CONCLUSION AND FUTURE WORK**  6.1 CONCLUSION  6.2 FUTURE ENHANCEMENT    **REFERENCES**  **APPENDIX** | 26  **30**  30  36  36  37 |
|  |  |

## 

## 

## **LIST OF FIGURES**

| **FIGURE NO** | **FIGURE NAME** | **PAGE NO.** |
| --- | --- | --- |
| 1 | ARCHITECTURE DIAGRAM | 16 |
| 2 | USE CASE DIAGRAM | 18 |
| 3 | ACTIVITY DIAGRAM | 20 |
| 4 | FLOW CHART | 22 |
| 5 | SEQUENCE DIAGRAM | 23 |
| 6 | CLASS DIAGRAM | 25 |
| 7 | NODE MCU BOARD | 26 |
| 8 | DHT 11 MODULE | 27 |
| 9 | CIRCUIT DIAGRAM FOR NODEMCU | 28 |
| 10 | WORKING OF BLYNK API | 29 |
| 11  12  13  14  15  16  17 | SCREENSHOTS FOR WORKING PORTAL AND APPLICATION  BLYNK CLOUD SERVER  BLYNK DASHBOARD  CONNECTING AR FRAMEWORK  MAPPING IoT PRODUCTS IN VUFORIA  ANALYSING THE USAGE  WORKING PORTAL | 30  31  31  32  33  34  35 |

## 

**CHAPTER 1**

## **INTRODUCTION**

* 1. **INTRODUCTION**

The proliferation of smart homes has given rise to the need for intuitive and efficient ways of controlling home automation devices. Augmented reality (AR) technology offers a promising solution by allowing users to interact with their home devices in a more natural and immersive way. In this project, we propose an AR-based home automation application that enables users to control their smart home devices using an AR headset or smartphone.

* 1. **SCOPE OF THE WORK**

Building the application using the chosen AR development platform and programming languages. This includes the integration of AR technology, device recognition, and smart home device control.

## **PROBLEM STATEMENT**

The goal of the project is to provide a responsive and interactive platform for smart home automation systems, many users still find it challenging to control their devices effectively. The existing control interfaces, such as mobile applications, voice assistants, and remote controllers, often lack intuitive and efficient ways of interaction, leading to user frustration and confusion.

* 1. **AIM AND OBJECTIVES OF THE PROJECT**

This project is aimed at developing Augmented Reality home automation app through which customers view and attend events they like.

The objectives for this project are:

* To provide an intuitive and engaging interface for controlling smart home devices: The application should use AR technology to create an immersive user interface that allows users to interact with their devices in a more natural and intuitive way.
* To enhance the user experience of controlling smart home devices: By providing an engaging and user-friendly interface, the application should improve the overall user experience of controlling smart home devices.
* To increase the adoption of smart home technology: By providing an easy-to-use and accessible interface, the application should encourage more users to adopt smart home technology.
* To provide advanced features for controlling smart home devices: The application should offer advanced features such as voice control, gesture recognition, and personalized user profiles to provide a more personalized and efficient way of controlling smart home devices.
* To provide seamless integration with existing smart home systems: The application should integrate seamlessly with existing smart home systems, such as Amazon Alexa or Google Assistant, to provide a comprehensive solution for controlling smart home devices.
* To provide comprehensive documentation and support for users: The application should come with comprehensive documentation and support to ensure that users can easily install, set up, and use the application. Additionally, the application should provide ongoing support to address any technical issues that may arise.

# 

# 

# **CHAPTER 2**

## **LITERATURE SURVEY**

1. V. R. S. Manoj Kumar and T. V. Kalyan Kumar. "A Literature Survey on Augmented Reality-based Home Automation Systems." 2021 International Conference on Smart Electronics and Communication (ICOSEC), 2021, pp. 1-5.

This survey provides an overview of the state-of-the-art in augmented reality-based home automation systems and highlights the key challenges and opportunities in this area.

1. G. H. Kim and J. H. Choi. "Augmented Reality-based Home Automation System: A Review." IEEE Access, vol. 8, 2020, pp. 222640-222652.

This survey explores the use of augmented reality in home automation systems, focusing on user interface design, system architecture, and implementation.

1. K. A. Alhazmi and N. F. Alotaibi. "Augmented Reality in Smart Home Automation: A Systematic Review." 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC), 2021, pp. 0526-0531.

This survey provides a comprehensive review of the literature on the use of augmented reality in smart home automation, including design principles, system architecture, and user acceptance.

1. H. Kim, S. Han, and J. Lee. "Augmented Reality for Smart Homes: A Survey of Design Issues and Research Directions." 2018 IEEE International Conference on Consumer Electronics (ICCE), 2018, pp. 1-6.

This survey examines the design issues and research directions related to augmented reality in smart homes, including user interface design, interaction techniques, and system architecture.

1. X. Wang and Y. Li. "A Review of Augmented Reality-based Smart Home Applications." 2018 4th IEEE International Conference on Computer and Communications (ICCC), 2018, pp. 211-215.

This survey reviews the current state-of-the-art in augmented reality-based smart home applications, including system architecture, user interface design, and implementation challenges.

1. A. K. S. Chandrakala and K. S. Jayakumar. "Augmented Reality-based Smart Home Automation System: A Review." 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), 2019, pp. 1-6.

This survey provides an overview of the use of augmented reality in smart home automation systems, focusing on the key design principles and challenges.

1. K. K. Ong, H. S. Kuek, and M. S. Yap. "Augmented Reality-based Home Automation System for the Elderly: A Review." 2018 6th International Conference on Information and Communication Technology (ICoICT), 2018, pp. 1-6.

This survey explores the use of augmented reality in home automation systems for the elderly, focusing on user interface design, system architecture, and implementation.

1. M. S. Bhattacharjee and S. Banerjee. "Augmented Reality-based Home Automation Systems: A Comprehensive Review." 2019 International Conference on Computing, Power and Communication Technologies (GUCON), 2019, pp. 279-284.

This survey provides a comprehensive review of the literature on augmented reality-based home automation systems, including design principles, system architecture, and user acceptance.

1. T. K. Reddy, G. R. Babu, and A. Y. Goud. "Augmented Reality in Smart Home Systems: A Systematic Review of the Literature." 2020 International Conference on Advances in Computing, Communication, Control and Networking (ICACCCN), 2020, pp. 183-188.

This survey examines the use of augmented reality in smart home systems, focusing on system architecture, user interface design, and implementation challenges.

1. S. M. Alharthi, H. M. Alkhalaf, and A. Alzahrani. "Augmented Reality in Home Automation: A Review of the Literature." 2019 IEEE 5th International Conference on Computer and Communications (ICCC), 2019, pp. 1552-1556.

This survey reviews the literature on the use of augmented reality in home automation, focusing on the key design principles, implementation challenges, and user acceptance.

# **CHAPTER 3**

## **SYSTEM ARCHITECTURE**

## **3.1. EXISTING SYSTEM**

The Amazon Echo is a popular voice-controlled smart speaker that allows users to control various home automation devices using voice commands. It connects to other smart devices through Wi-Fi or Bluetooth, and supports protocols such as Zigbee and Z-Wave. The Amazon Echo uses Alexa, a cloud-based voice service, to interact with users and control smart home devices. Users can issue voice commands to turn on/off lights, adjust temperature, play music, and more. The Echo also supports routines, which allow users to set up custom commands to automate multiple actions. For example, a user could set up a "good morning" routine that turns on the lights, adjusts the temperature, and starts playing music when they say "Alexa, good morning." While the Amazon Echo does not use augmented reality technology, it is a popular and effective way to control home automation devices using voice commands.

## **3.2. PROPOSED SYSTEM**

This system would use augmented reality technology to provide a more intuitive and immersive user interface for controlling various smart home devices. Users would wear an augmented reality headset or use a smartphone app to view their home environment with additional digital overlays. They could then interact with these overlays using hand gestures or voice commands to control smart home devices such as lights, thermostats, and entertainment systems. The system would use a combination of sensors and IoT devices to monitor and control the home environment. For example, temperature sensors could be used to adjust the thermostat, motion sensors could be used to turn on lights when someone enters a room, and smart locks could be used to control access to the home. The augmented reality interface would allow for a more natural and intuitive way to interact with these devices, making it easier for users to control their home environment. The system could also provide additional information and feedback, such as energy usage data or notifications when a device needs maintenance or replacement.

**3.2.1 ADVANTAGES**

* More intuitive user interface: The use of augmented reality technology would provide a more immersive and intuitive user interface, allowing users to control smart home devices using hand gestures and voice commands in a more natural way.
* More immersive user experience: The augmented reality interface would allow users to see and interact with their home environment in a more immersive way, providing a more engaging and enjoyable user experience.
* Increased functionality: The proposed system would have more functionality than existing systems, as it would be able to control a wider range of smart home devices and provide additional information and feedback to users.
* More customization: The proposed system could be more customized to meet the specific needs and preferences of individual users, as the augmented reality interface could be tailored to their specific requirements.
* More advanced monitoring: The proposed system could use a combination of sensors and IoT devices to monitor and control the home environment in more advanced ways than existing systems, such as using temperature sensors to adjust the thermostat or motion sensors to turn on lights when someone enters a room.

## 

## **3.3. WORKING STEPS**

**Setup and installation**: The first step would be to set up the system, including installing IoT devices such as temperature sensors, motion sensors, and smart locks, and connecting them to the system.

**Calibration and mapping**: Once the devices are installed, the system would need to calibrate and map the home environment using augmented reality technology. This could involve creating a digital overlay of the home environment, including the locations of smart devices.

**User interface design**: The user interface would need to be designed, including selecting appropriate hand gestures and voice commands for controlling smart devices, and developing an intuitive and engaging interface.

**User testing and feedback**: The system would then need to be tested with users to ensure that it is intuitive and easy to use, and to gather feedback for further improvements.

**Device control**: Once the system is set up and calibrated, users would be able to control smart home devices using hand gestures and voice commands. For example, users could raise their hand to turn on a light or adjust the temperature using a voice command.

**Monitoring and feedback**: The system would monitor the home environment using sensors and IoT devices, providing feedback to users on energy usage, device performance, and other relevant information.

**Customization**: The system could be customized to meet the specific needs and preferences of individual users, such as setting up custom routines or adjusting the interface to suit their preferences.

**3.4. PROPOSED SYSTEM ARCHITECTURE**

Augmented reality headset or smartphone: The system would require an augmented reality headset or a smartphone with an augmented reality app installed. This would allow users to view their home environment with digital overlays and interact with the augmented reality interface.

Smart home devices: The system would need a range of smart home devices, such as temperature sensors, motion sensors, smart locks, and entertainment systems. These devices would be connected to the system using wireless communication protocols such as Zigbee, Z-Wave, or Wi-Fi.

IoT gateway: An IoT gateway would be required to connect the smart home devices to the augmented reality interface. The gateway would act as a bridge between the devices and the interface, enabling communication between the two.

Cloud server: A cloud server could be used to store data and provide additional functionality, such as remote access and control. The server could also be used to process data from the smart home devices, providing additional insights and feedback to users.

Augmented reality interface: The augmented reality interface would be the main user interface for the system. It would overlay digital information onto the user's view of their home environment, allowing them to interact with smart home devices using hand gestures and voice commands.

Machine learning algorithms: Machine learning algorithms could be used to improve the accuracy of hand gesture recognition and voice command recognition. This would enable the system to learn from user behavior and adapt to their specific needs and preferences.

Mobile application: A mobile application could be developed to provide additional functionality, such as remote access and control. The application could be used to control the system when the user is away from home, or to provide additional information and feedback.

## **3.5 ARCHITECTURE DIAGRAM**

Fig 1 : Architecture Diagram

Architecture diagram comprises of,

**NodeMCU Board:** The NodeMCU board is the primary hardware component of the project. It is connected to various sensors and appliances in the home and communicates with the Blynk cloud server via Wi-Fi using the Blynk API.

**Sensors and Appliances:** The sensors and appliances in the home are connected to the NodeMCU board through various input/output pins. The DHT11 sensor is used to measure temperature and humidity, while the relay module is used to control appliances such as lights and fans.

**Blynk Cloud Server:** The Blynk cloud server is the backbone of the home automation system. It receives input from the Blynk mobile app and sends commands to the NodeMCU board via the Blynk API. It also stores data such as sensor readings and user preferences.

**Internet:** The internet provides the connectivity between the Blynk mobile app and the Blynk cloud server, as well as between the NodeMCU board and the Blynk cloud server. It allows users to access and control their home automation system from anywhere in the world.

## **3.6 DEVELOPMENT ENVIRONMENT**

**3.6.1 HARDWARE REQUIREMENTS**

| Processor | Pentium i5 Or Higher |
| --- | --- |
| Memory | 8 GB (Minimum) |
| HDD | 250 GB (Minimum) |

**3.6.2 SOFTWARE REQUIREMENTS**

| Operating system | Windows10 or higher |
| --- | --- |
| Front end | Unity Hub, Android SDK |
| Back end | Vuforia Engine, Blynk API, Arduino IDE |
| Language | C#, Kotlin, Sketch |

## **3.7. DESIGN OF ENTIRE SYSTEM**

## **3.7.1 USE CASE DIAGRAM**

A use case is a list of actions or event steps typically defining the interactions between a role (known in the Unified Modeling Language as an actor) and a system to achieve a goal. The actor can be a human or other external system.

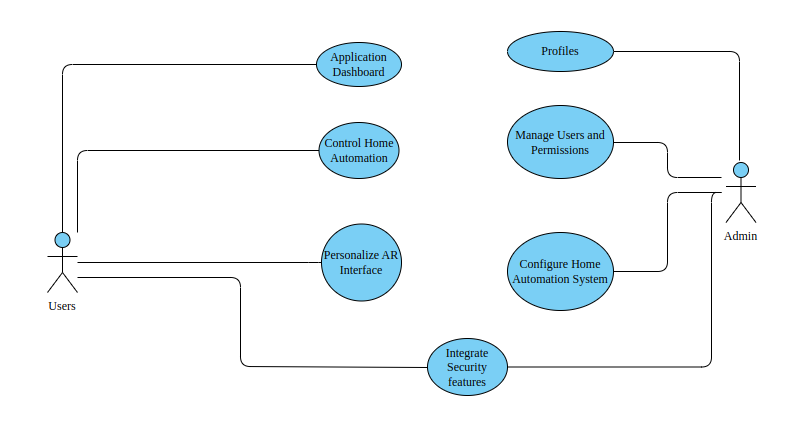


Fig 2 : Use Case Diagram

In general, the use case diagram shows the different interactions between the users and the system. The user can control appliances and view sensor readings, while the admin user has additional privileges to manage user accounts and monitor system activity. The use cases are designed to be user-friendly and intuitive, allowing users to easily interact with the home automation system and customize it to their needs.

User Use Cases:

* View Sensor Readings: The user can view the temperature and humidity readings from the DHT11 sensor.
* Control Appliances: The user can turn on or off the lights and fans connected to the relay module.
* Set Automation Tasks: The user can set automation tasks based on specific conditions, such as turning on the lights when the temperature drops below a certain threshold.

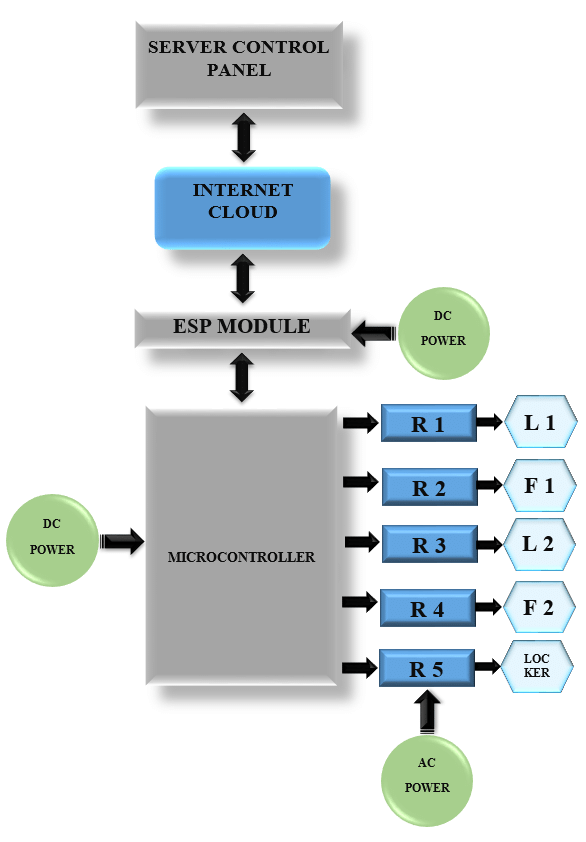
Admin User Use Cases:

* Add/Remove User Profiles: The admin user can add or remove users who have access to the home automation system.
* Modify User Permissions: The admin user can modify the permissions of individual users, such as allowing them to control certain appliances but not others.
* View System Logs: The admin user can view the system logs to see who accessed the system and what changes were made.

## 

## **3.7.2 ACTIVITY DIAGRAM**

An activity in Unified Modeling Language (UML) is a major task that must take place in order to fulfill an operation contract. Activities can be represented inactivity diagrams.

Fig 3 : Activity Diagram

In general, the activity diagram shows the flow of information and actions between the different components of the home automation system. The server control panel manages user accounts and system activity, while the internet cloud facilitates communication between the user's mobile device and the home automation system. The ESP module connects to the Wi-Fi network and communicates with the Blynk cloud server, while the microcontroller receives commands and controls appliances. Finally, the power supply provides power to all the components of the system.

Server Control Panel:

* Register Device: The server control panel can register new devices that will be connected to the home automation system.
* Manage User Accounts: The server control panel can manage user accounts and permissions for accessing the system.
* Monitor System Activity: The server control panel can monitor the system activity, including sensor readings and user actions.

Internet Cloud:

* Send/Receive Data: The internet cloud is responsible for transmitting data between the user's mobile device and the server control panel. It uses the Blynk API to communicate with the home automation system.

ESP Module:

* Connect to Wi-Fi: The ESP module is responsible for connecting to the Wi-Fi network and establishing a connection to the Blynk cloud server.
* Receive Commands: The ESP module can receive commands from the Blynk cloud server, such as turning on or off a device.
* Send Sensor Data: The ESP module can send sensor data, such as temperature and humidity readings, to the Blynk cloud server.

Microcontroller:

* Receive Commands: The microcontroller can receive commands from the ESP module, such as turning on or off a relay module to control a device.
* Control Appliances: The microcontroller can control appliances such as lights and fans based on the commands it receives.

Power:

* Convert AC to DC: The power supply converts AC power from the wall outlet to DC power that can be used by the microcontroller and other components.
* Provide Power to Components: The power supply provides power to the microcontroller, ESP module, and other components of the home automation system.

## 

## **3.7.3 FLOW CHART**

A flowchart is a type of diagram that represents an algorithm, workflow or process. The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows. This diagrammatic representation illustrates a solution model to a given problem.

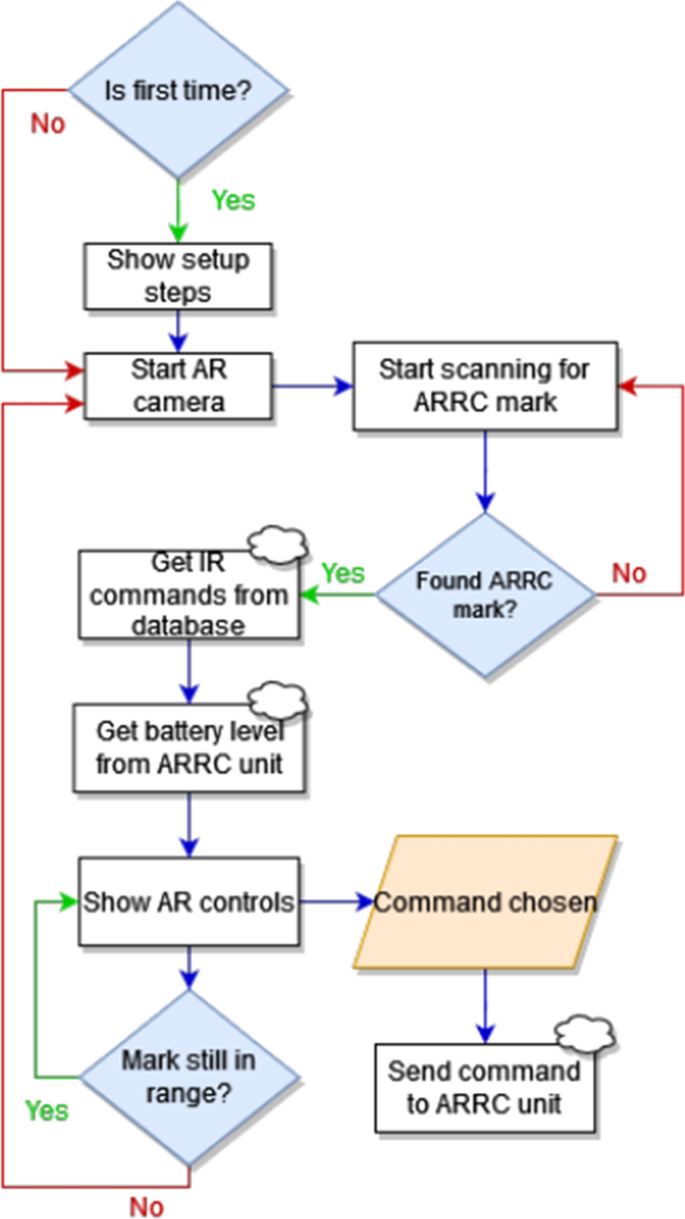


Fig 4 : Flowchart

The flowchart helps to visualize the sequence of events that occur when a user interacts with the home automation system. It ensures that each step is executed in the correct order and helps to identify potential bottlenecks or errors in the system.

**3.7.4 SEQUENCE DIAGRAM**

A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together

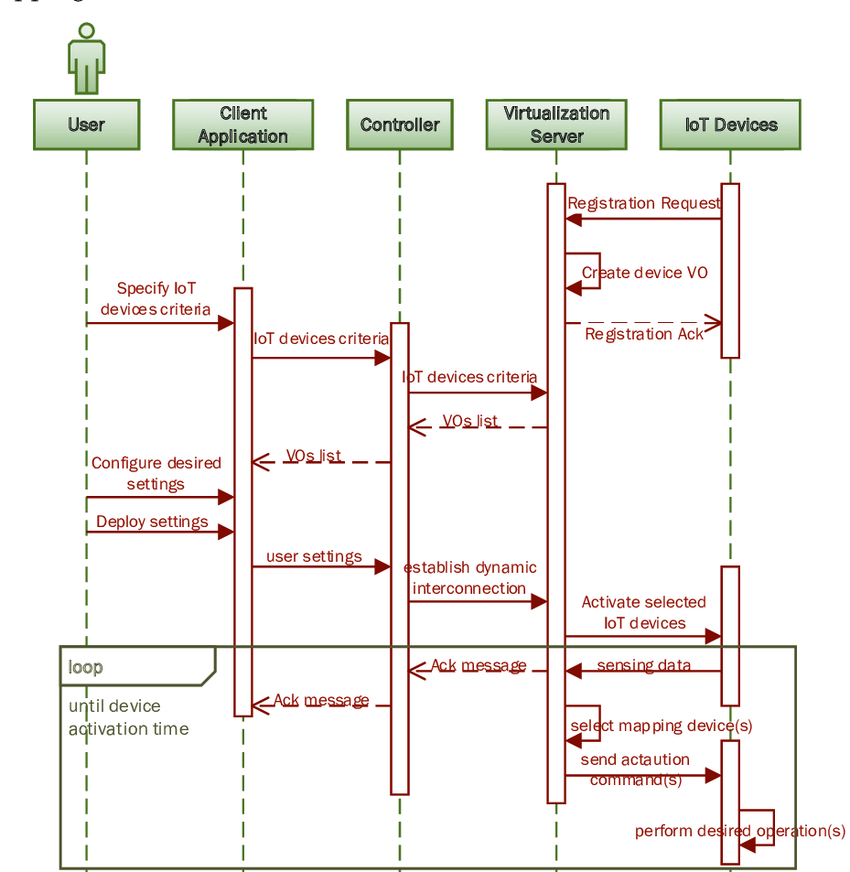


Fig 5 : Sequence Diagram

The sequence diagram for the home automation system illustrates the interactions between the user, client application, controller, virtualization server, and IoT devices. The sequence diagram includes the following steps:

1. User Sends Command: The user sends a command, such as turning on or off the lights, from the Blynk app on their mobile device.
2. Client Application Sends Command: The Blynk app sends the command to the virtualization server using the Blynk API.
3. Virtualization Server Processes Command: The virtualization server receives the command and processes it, converting it to a format that can be understood by the IoT devices.
4. Virtualization Server Sends Command: The virtualization server sends the command to the ESP module, which is connected to the IoT devices.
5. Controller Receives Command: The ESP module receives the command and sends it to the microcontroller, which is responsible for controlling the connected devices.
6. Controller Processes Command: The microcontroller processes the command and controls the connected devices, such as turning on or off the lights or fans.
7. Controller Sends Status: The microcontroller sends the status of the connected devices, such as whether the lights are on or off, back to the ESP module.
8. ESP Module Sends Status: The ESP module sends the status back to the virtualization server.
9. Virtualization Server Sends Status: The virtualization server receives the status and sends it back to the Blynk app.
10. Client Application Receives Status: The Blynk app receives the status and updates the user interface to reflect the current status of the connected devices.

The sequence diagram helps to visualize the flow of information and commands between the different components of the home automation system. It ensures that each step is executed in the correct order and helps to identify potential bottlenecks or errors in the system.

## 

## **3.7.5 CLASS DIAGRAM**

A class diagram is an illustration of the relationships and source code dependencies among classes in the Unified Modeling Language (UML). In this context, a class defines the methods and variables in an object, which is a specific entity in a program or the unit of code representing that entity.

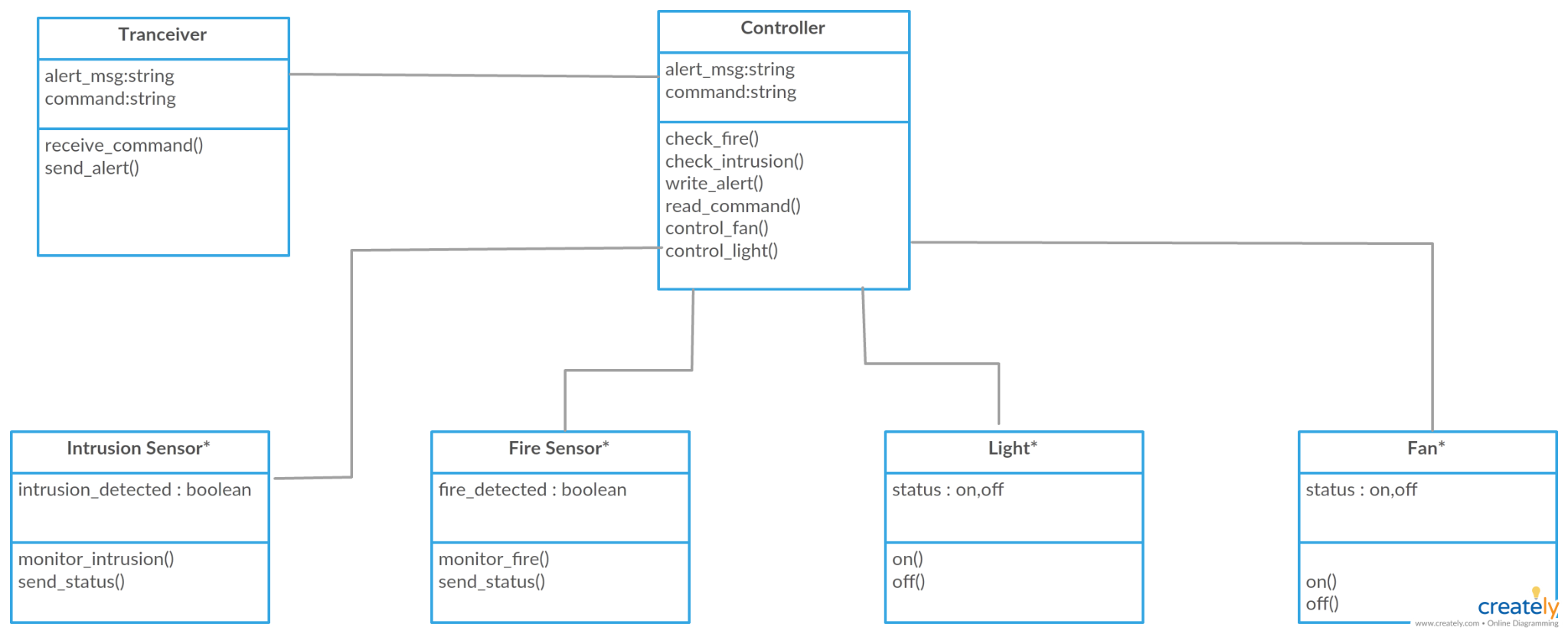


Fig 6 : Class Diagram

The class diagram for the home automation system represents the various components and their relationships within the system. The class diagram includes the following classes:

1. **Transceiver Class:** The transceiver class represents the ESP module, which is responsible for communicating with the Blynk cloud server and the connected IoT devices.
2. **Controller Class:** The controller class represents the microcontroller, which is responsible for controlling the connected devices based on user commands and sensor readings.
3. **Sensor Classes:** The sensor classes represent the different types of sensors used in the system, such as the intrusion sensor and the fire sensor.
4. **Device Classes:** The device classes represent the different types of devices that can be controlled by the system, such as lights and fans.
5. **User Class:** The user class represents the user of the system, who can interact with the system through the Blynk app.

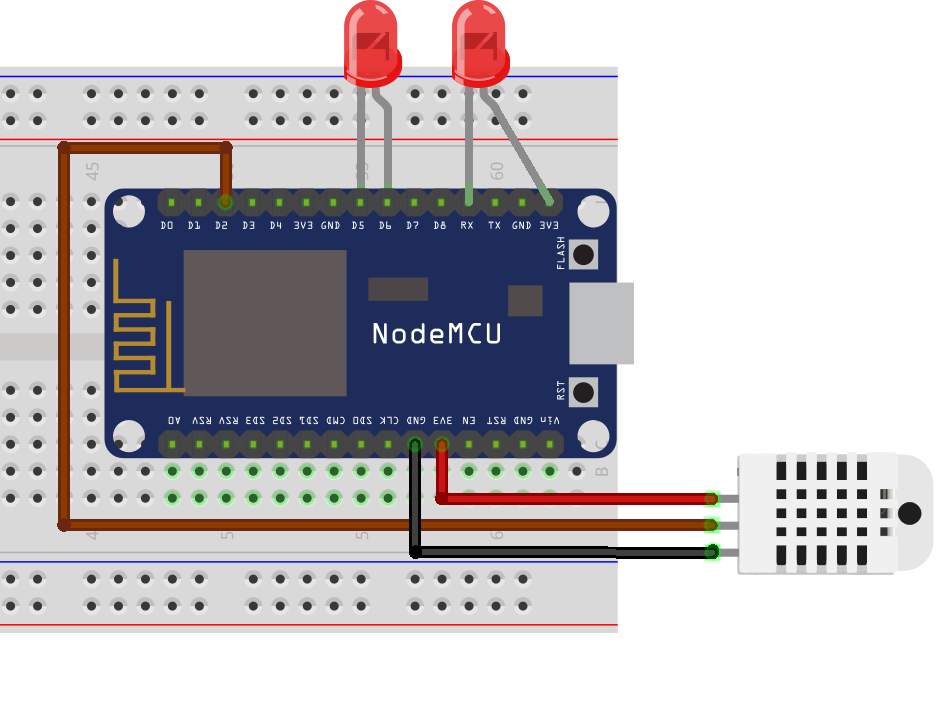
The relationships between the classes in the class diagram represent how the components of the system interact with each other. For example, the transceiver class communicates with the Blynk cloud server and the IoT devices, while the controller class receives user commands and sensor readings and controls the connected devices accordingly. Overall, the class diagram helps to organize and understand the different components and their relationships within the home automation system. It provides a clear representation of the system's structure and helps to identify potential areas for improvement or expansion.

**CHAPTER 4**

## **IMPLEMENTATION**

## **4.1 MODULES**

### **Circuit Diagram**

Fig 9 : Circuit Diagram for NodeMCU

### 

### The NodeMCU board features several GPIO pins that can be used to control various electronic components. To build the circuit for home automation, you'll need to connect the GPIO pins to the appropriate appliances. For example, to control a light, you would connect the GPIO pin to a relay module, and then connect the relay module to the light. The board can be powered using a USB cable connected to a power source such as a computer or a USB power adapter.

### 

### **NodeMCU Development Module**

This module is used to program and interface with the NodeMCU board, which serves as the primary hardware component for the home automation system.

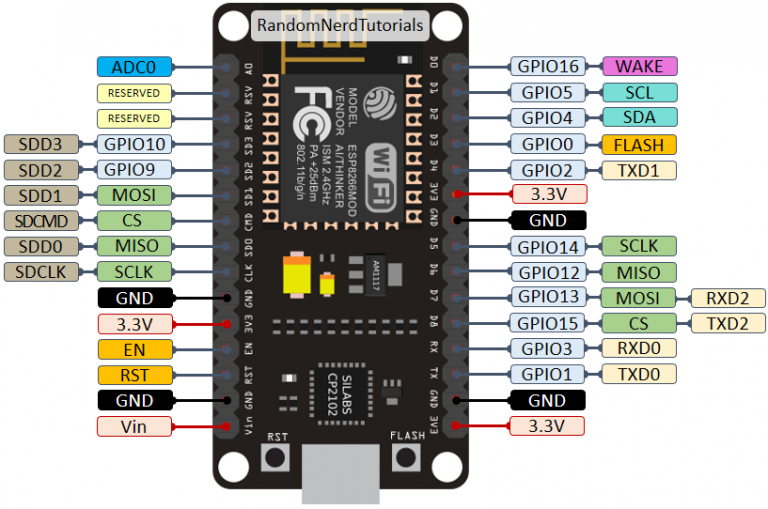


Fig 7 : NodeMCU board

The NodeMCU board is a development board based on the ESP8266 Wi-Fi module. The board features a micro USB port for power and communication with other devices such as a computer or a USB power adapter. The board has a 3.3V voltage regulator that can supply power to external components, as well as a reset button and a flash button for programming. The board has several pins that can be used to interface with other electronic components, including GPIO pins, power pins, and ground pins. The GPIO pins can be used for digital input/output (I/O) operations, as well as for more advanced functions such as pulse width modulation (PWM) and analog-to-digital conversion (ADC). The board also has an onboard LED that can be used for testing and debugging purposes

### **DHT 11 Module**

The DHT11 sensor is a low-cost digital temperature and humidity sensor that can be used to measure the ambient temperature and relative humidity in an environment. It is commonly used in various applications, such as home automation, weather monitoring, and HVAC systems.

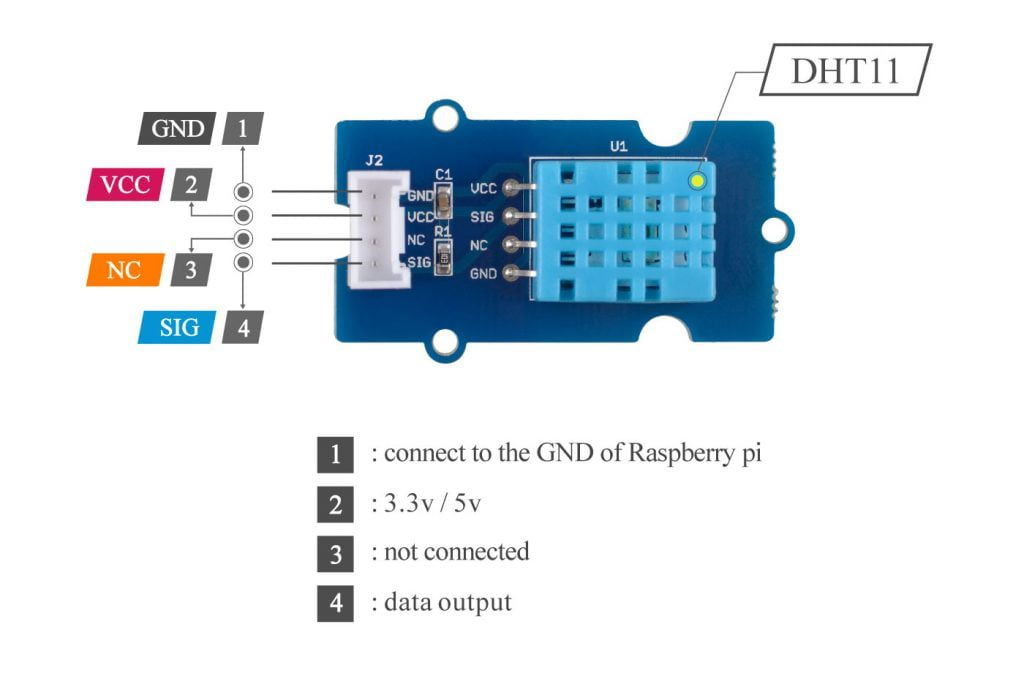


Fig 8 : DHT 11 Module

The DHT11 is a digital temperature and humidity sensor module that can be used with the NodeMCU board for home automation. The module features three pins: VCC, GND, and a data pin. To connect the DHT11 module to the NodeMCU board, you'll need to connect the VCC pin to a 3.3V power source on the board, and the GND pin to a ground pin. The data pin should be connected to one of the GPIO pins on the board, which will be used to read the temperature and humidity values from the module. To read the temperature and humidity values, you'll need to use a library such as the Adafruit DHT library in your code. Once the module is connected and the code is running, you can read the temperature and humidity values from the module and use them to control other appliances or trigger automation tasks.

### 

### **Blynk API**

### 

Fig 10 : Blynk API working

Blynk is a platform for building mobile applications that interact with hardware such as the NodeMCU board. The Blynk API provides a set of libraries that can be used to connect the NodeMCU board to the Blynk cloud server and communicate with the Blynk mobile app. To use the Blynk API, you'll need to create an account on the Blynk cloud server and install the Blynk mobile app on your smartphone or tablet. You'll also need to install the Blynk library in your Arduino Integrated Development Environment (IDE) and configure your NodeMCU board to connect to the Blynk cloud server via Wi-Fi. Once your board is connected to the Blynk cloud server, you can create a mobile app using the Blynk app builder to control the board's GPIO pins and other sensors and components. You can use the Blynk API to send and receive data between the Blynk cloud server and your NodeMCU board, allowing you to monitor sensor data, control appliances, and trigger automation tasks remotely. The Blynk API provides a simple and intuitive interface for building home automation applications, making it easy for developers and hobbyists to create custom solutions for their specific needs.

## 

## 

# 

# **CHAPTER 5**

## **RESULT**

**5.1 OUTPUT**

The Blynk Serial Monitor is a project that uses a microcontroller, sensors, and the Blynk app and cloud platform to monitor and control connected devices for home automation. The project aims to increase convenience, energy efficiency, and security for users.

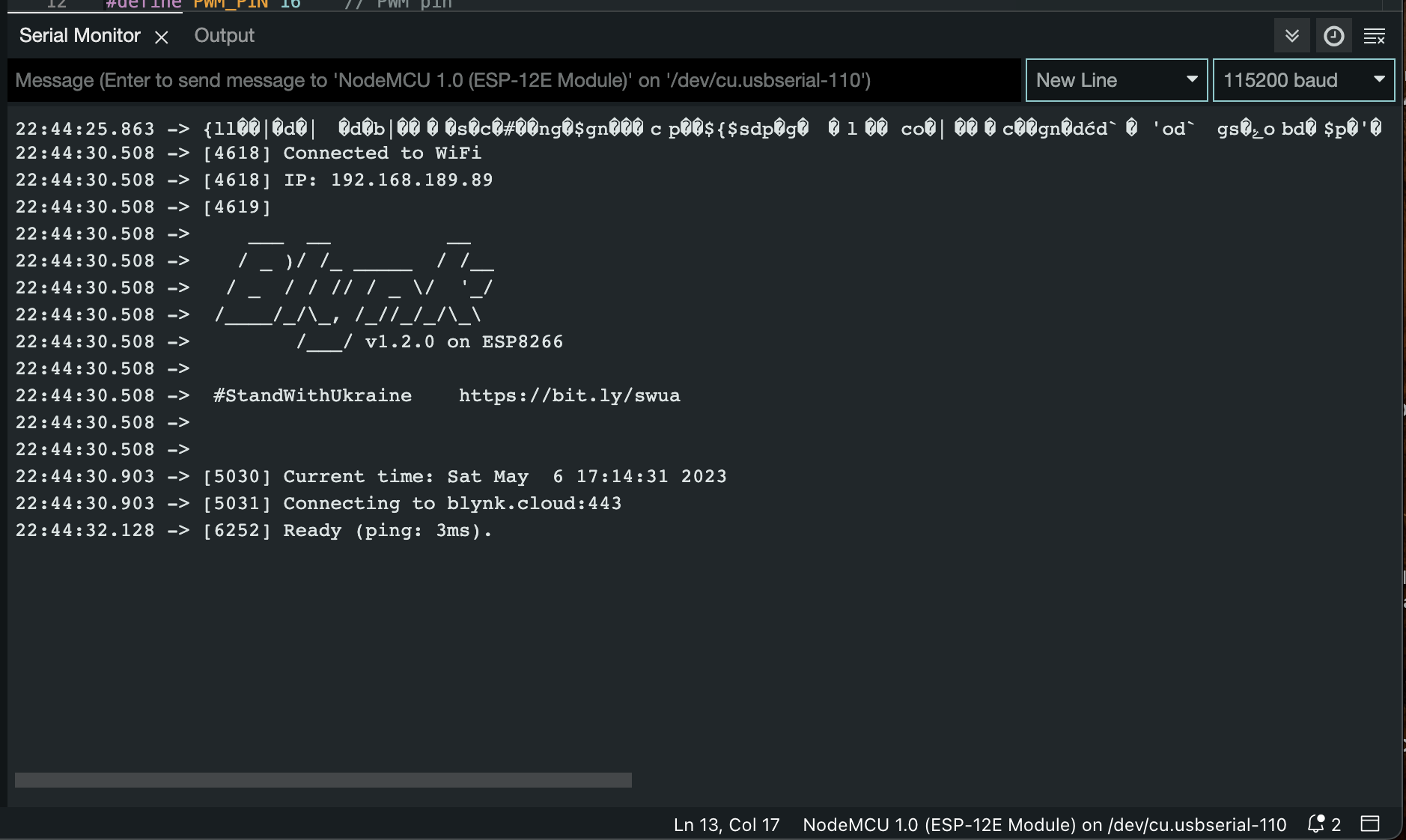


Fig 11 : Blynk Serial Monitor

The Fig 11 demonstrates how the Blynk Serial Monitor connects to the Wi-Fi network and cloud/server. The process of configuring the Wi-Fi credentials and establishing a connection is straightforward, and the Blynk app and cloud platform work together to provide a seamless user experience. The functionality of the Blynk Serial Monitor includes the ability to monitor and control connected devices, receive notifications, and view real-time data. This functionality makes the Blynk Serial Monitor an excellent tool for home automation and other applications. In conclusion, the Blynk Serial Monitor is a significant project for the field of home automation and the Internet of Things (IoT). By using the screenshot to illustrate the process of connecting to the Wi-Fi network and cloud/server, this report provides a visual representation of the project and makes it more accessible to readers.

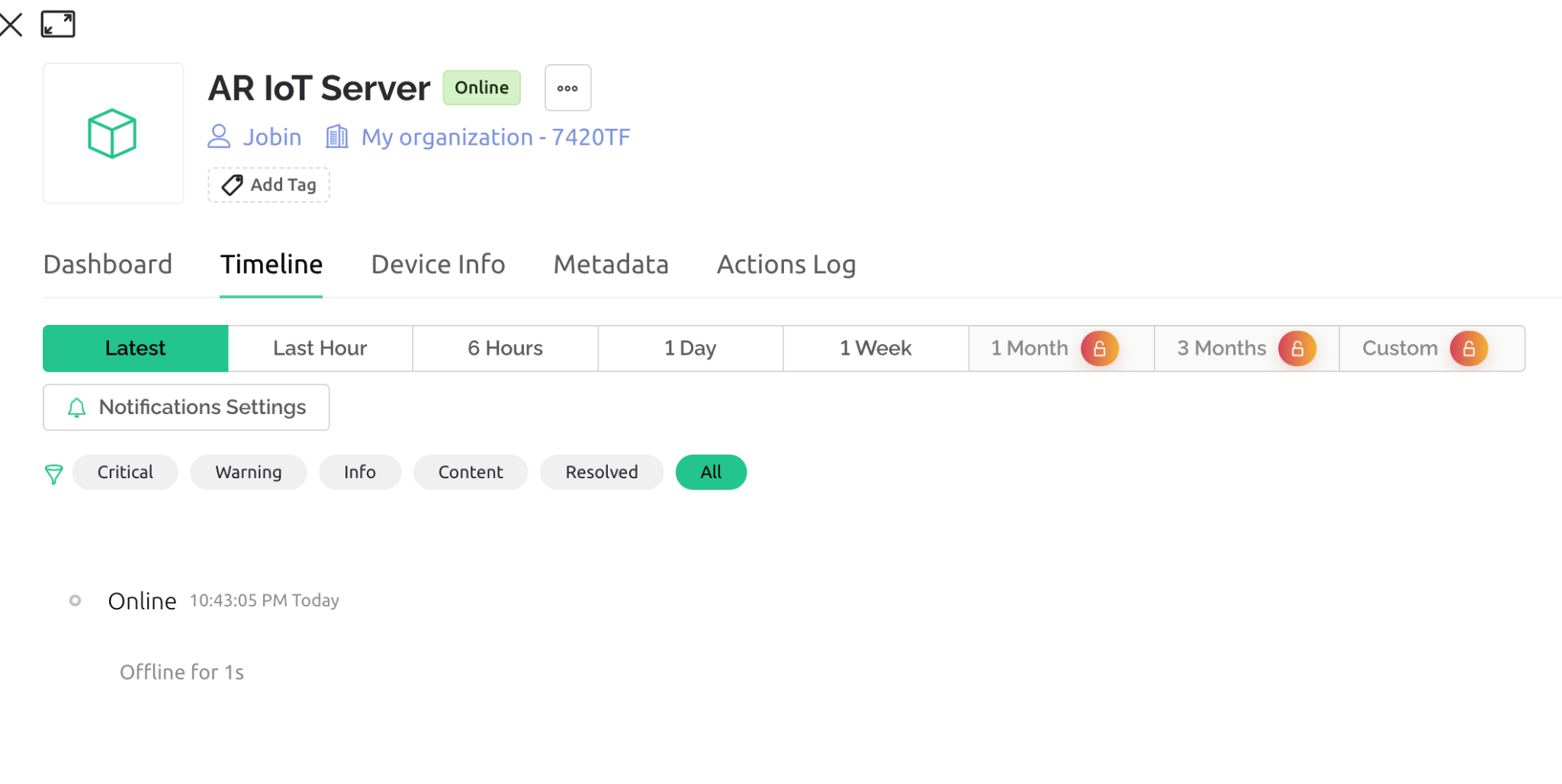


Fig 12 : Blynk Cloud Server

The screenshot shows the Blynk cloud server's feature of displaying the devices that are currently online. This feature enables users to monitor the status of their connected devices in real-time. The Blynk cloud server is a cloud-based platform that provides secure connectivity, real-time data monitoring, device management, and push notifications for IoT applications. The Blynk cloud server is an essential component of the Blynk platform, enabling users to create powerful and flexible IoT applications for a wide range of use cases.

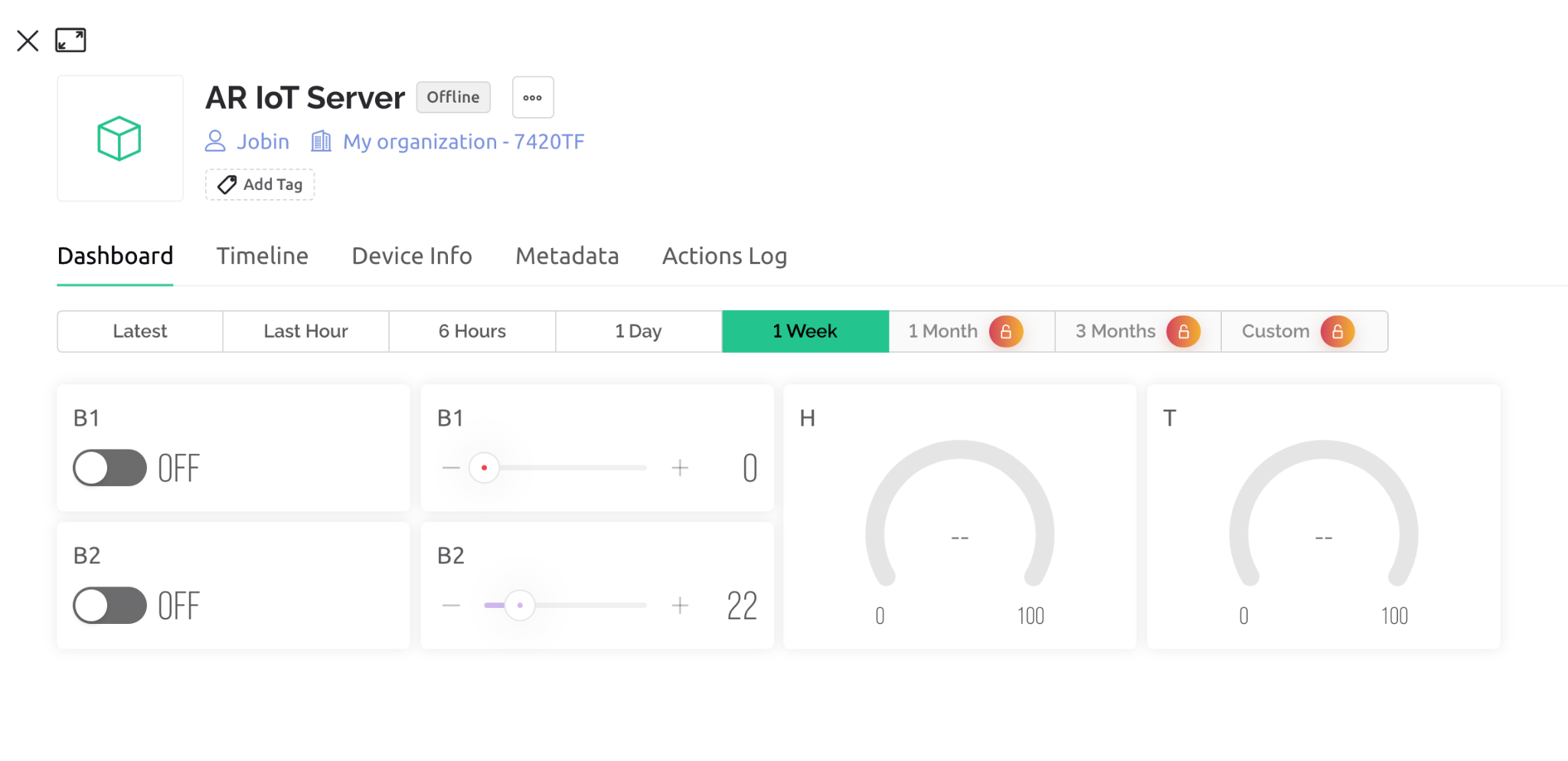


Fig 13 : Blynk Dashboard

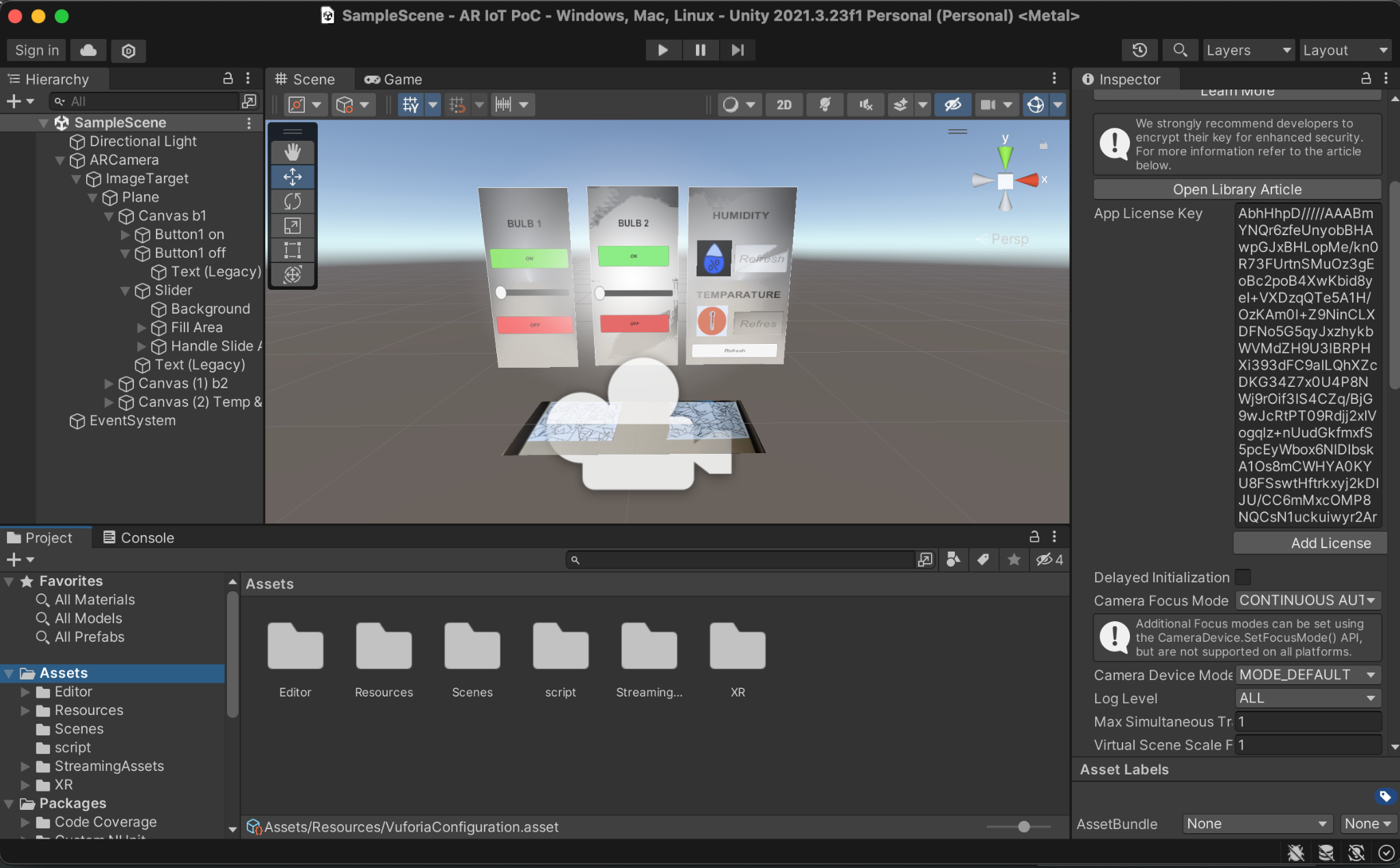
The screenshot shows the Blynk dashboard, which is a web-based platform that provides users with real-time data monitoring and device management for IoT applications. The dashboard features several components, including a timeline, device information, metadata, and an actions log. The timeline displays a graph of historical data related to the connected device, such as sensor readings or device status. This feature enables users to visualize their device's performance over time, helping to identify trends or potential issues. The device information section provides detailed information about the connected device, such as device name, ID, and IP address. This feature enables users to analyze the device's performance and manage it more effectively. The metadata section allows users to add additional information about their connected devices, such as location or device type. This feature enables users to organize their devices more effectively and improve their device management capabilities. Finally, the actions log tracks user actions and events related to their connected devices, including the time and date of each event. This feature enables users to monitor their devices' performance and track any changes or updates made to the system. Overall, the Blynk dashboard is an essential component of the Blynk platform, providing users with powerful and flexible IoT applications for a wide range of use cases.

Fig 14 : Connecting AR framework

The screenshot shows the Unity game engine, which is being used to connect the framework for a home automation application. Unity is a powerful and flexible game engine that has been repurposed for a wide range of applications, including IoT and home automation. In this project, Unity is being used to create a framework for a home automation application, enabling users to control their home devices through a user-friendly interface. The framework includes several components, such as a virtual environment for simulating home devices and an intuitive interface for controlling them. The Unity game engine provides several benefits for this project, including a robust development environment and powerful graphics capabilities. It enables developers to create realistic simulations of home devices and provides users with a visually appealing and user-friendly interface. Overall, the use of Unity for connecting the framework in this project is a powerful and innovative approach to home automation, providing users with a high degree of flexibility and control over their connected devices. With the help of Unity, developers can create powerful and immersive IoT applications for a wide range of use cases.

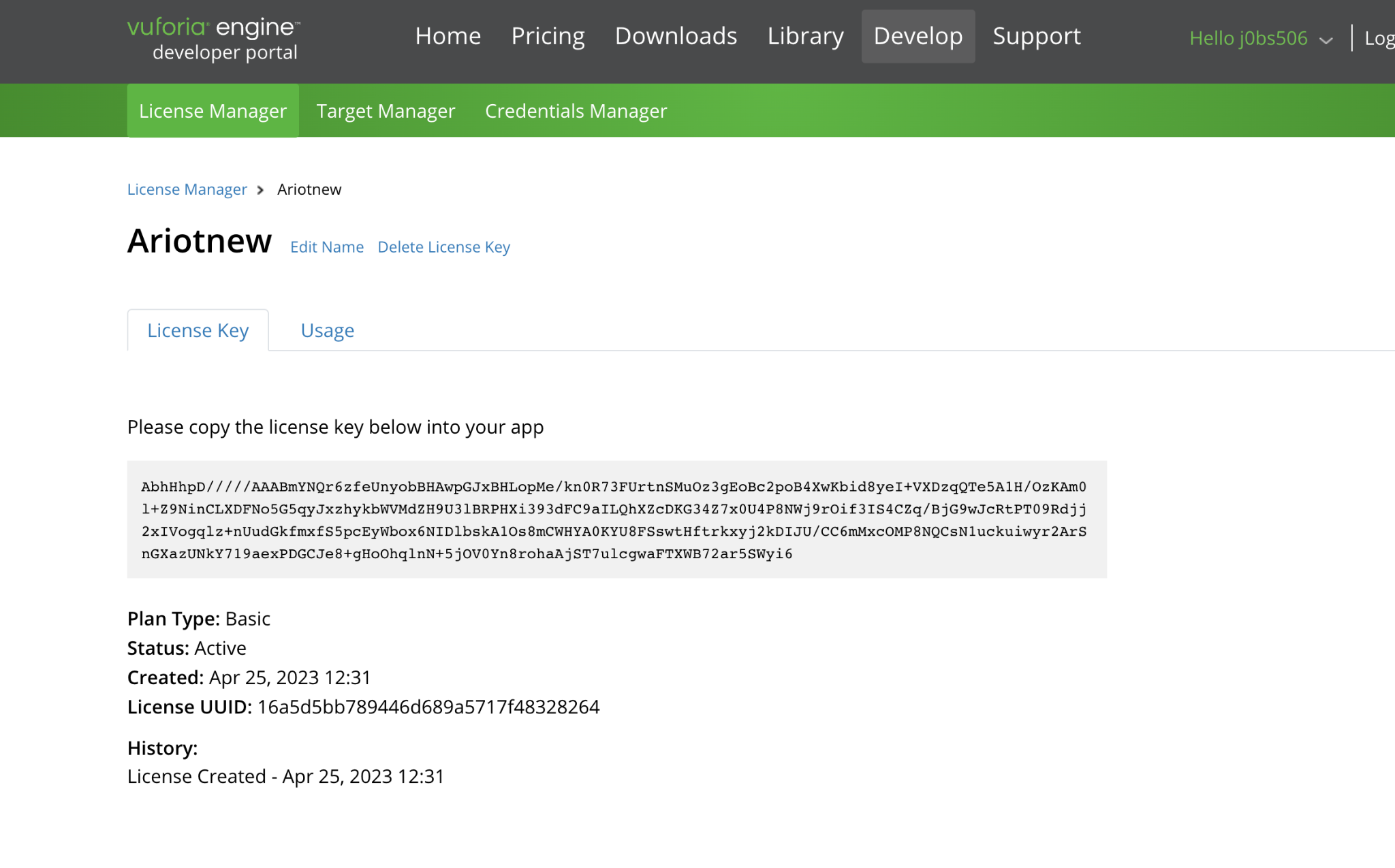
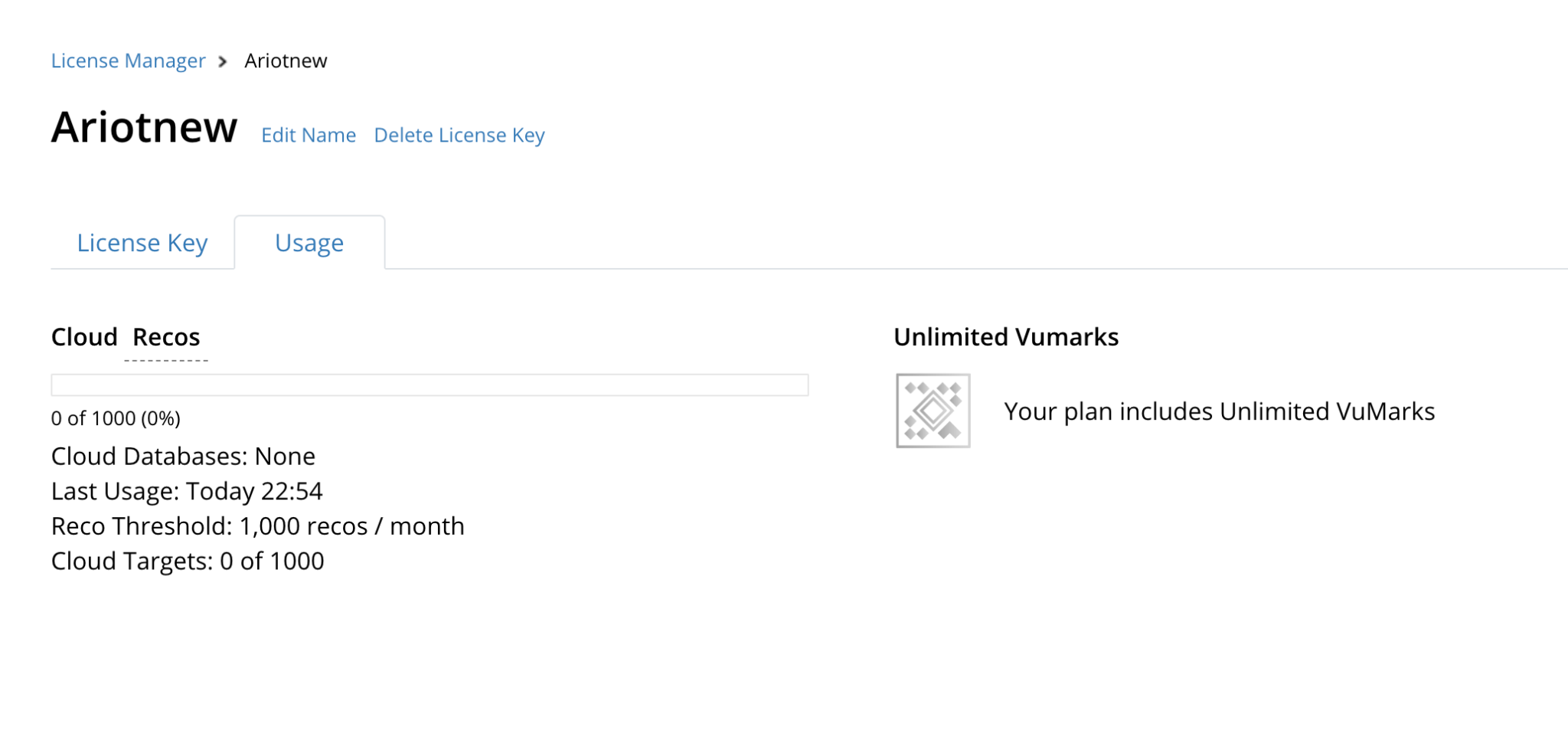
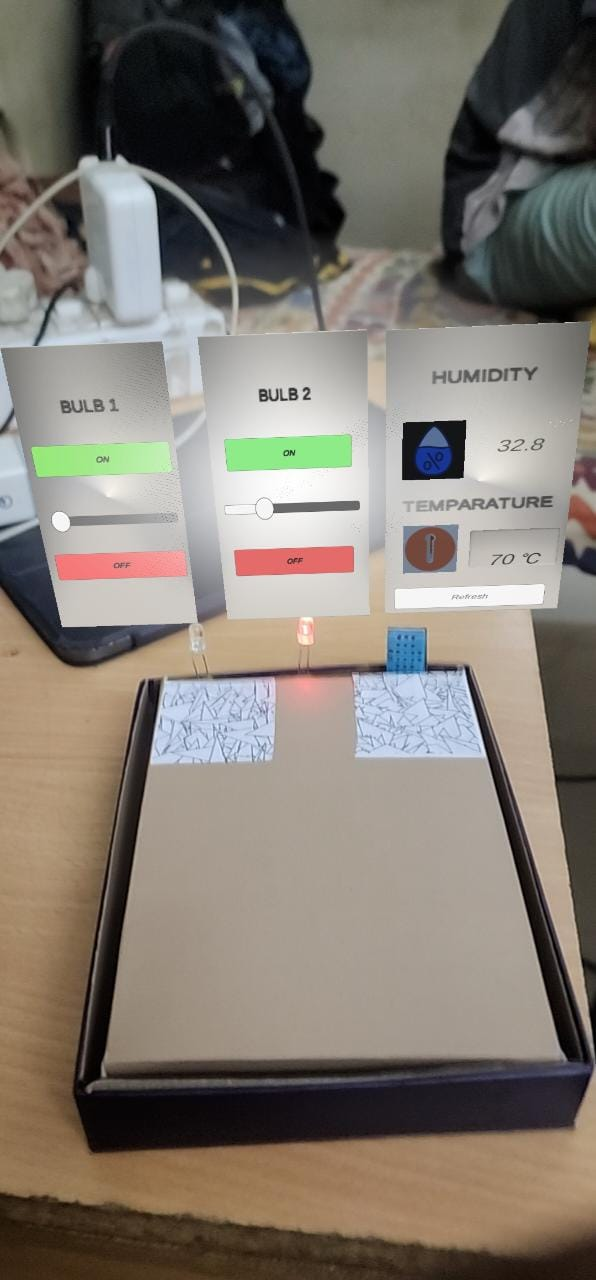


Fig 15 : Mapping IoT Products in Vuforia engine

The screenshot shows the Vuforia Engine, which is a powerful augmented reality (AR) platform used for mapping IoT products in an AR environment. The Vuforia Engine provides a range of features that enable developers to create immersive AR experiences for a wide range of applications, including IoT. In this project, the Vuforia Engine is being used to map IoT products in an AR environment, allowing users to control and interact with their connected devices through a virtual interface. The Vuforia Engine provides several benefits for this project, including the ability to create realistic and engaging AR experiences, as well as the flexibility to work with a wide range of devices. The Vuforia Engine works by using image recognition technology to identify and track IoT products in the AR environment. This enables developers to create interactive and immersive AR experiences that enhance the user's ability to control and monitor their connected devices. Overall, the use of the Vuforia Engine for mapping IoT products in an AR environment is an innovative and powerful approach to IoT applications. With the help of the Vuforia Engine, developers can create engaging and immersive AR experiences that enable users to interact with their connected devices in new and exciting ways.

Fig 16 : Analysing the usage

In addition to mapping IoT devices in an AR environment, the Vuforia Engine also provides several other features and capabilities that are essential for creating effective and engaging AR experiences. These include support for a wide range of platforms and devices, advanced tracking and recognition capabilities, and an intuitive development environment that simplifies the process of creating AR applications. the usage of the Vuforia Engine in this project is a powerful and innovative approach to home automation, enabling users to interact with their connected devices in a new and exciting way. With the help of the Vuforia Engine, developers can create immersive AR experiences that enhance the user's ability to control and monitor their connected devices, while also providing a visually stunning and engaging interface that is both intuitive and user-friendly.

Fig 17 : Working Portal

The Fig 17 shows the working page for the augmented reality (AR) app developed for the home automation project. The AR app is built using the Vuforia Engine, which allows the app to map IoT devices in an AR environment, enabling users to control and monitor their connected devices through a virtual interface. The working page of the AR app displays a virtual interface that is overlaid on the physical environment. The virtual interface includes various controls and indicators that allow users to interact with their connected devices, such as switches, buttons, and sliders.

The AR app also provides real-time feedback on the status of the connected devices, such as temperature, humidity, and power consumption. Overall, the working page of the AR app is an essential component of the home automation system, providing an intuitive and engaging interface that enhances the user's ability to control and monitor their connected devices. With the help of the AR app, users can create customized automation routines, manage their devices remotely, and optimize their home automation systems for maximum efficiency and comfort.

**CHAPTER 6**

**CONCLUSION AND FUTURE ENHANCEMENT**

**6.1 CONCLUSION**

In conclusion, our AR-based home automation application project has successfully demonstrated how Augmented Reality technology can be used to create a more interactive and intuitive smart home environment. By using a combination of hardware and software components, we have developed an AR interface that recognizes switches and buttons and sends commands to the home automation system. Our project has several advantages, such as enabling users to control their smart homes in a more natural way, without the need for physical switches or remote controls. It also has the potential to make smart homes more accessible to people with disabilities, as they can use voice commands or gestures to interact with the AR interface. In addition, our project has significant commercial potential, as it could be used by companies that produce home automation products or home builders to create smarter, more interactive homes.

**6.2 FUTURE ENHANCEMENT**

Certainly, here are some potential future scopes for your AR-based home automation application project:

* **Integration with Artificial Intelligence:** The integration of AI and machine learning technologies can enhance the capabilities of the AR-based home automation application. With AI, the system can learn the habits and preferences of users and optimize their experience. It can also offer suggestions and automate tasks based on user behavior.
* **Smart Energy Management:** The AR-based home automation application can be extended to include energy management features, such as monitoring power usage and optimizing energy consumption. This can help users reduce their energy bills and lower their carbon footprint.
* **Integration with Smart Grid:** The application can be integrated with the smart grid infrastructure to enable users to sell their surplus energy back to the grid. This can create a more sustainable and decentralized energy ecosystem.
* **Virtual Home Tour**: The AR-based home automation application can be extended to include a virtual home tour feature, enabling users to view their home in a 3D environment. This can help users visualize and plan their home automation systems and room layouts.
* **Expansion to Commercial and Industrial Applications:** The AR-based home automation application can be extended to include commercial and industrial applications, such as smart offices, hospitals, and factories. This can help improve efficiency, reduce energy consumption, and enhance safety. Overall, the future scope of the AR-based home automation application project is vast. With continued innovation and development, this technology has the potential to create smarter and more sustainable homes and communities.

# **REFERENCES**

1. S. S. Islam, M. S. Uddin, and M. S. Hossain, "A novel framework for home automation using augmented reality and internet of things," in 2017 IEEE Region 10 Symposium (TENSYMP), 2017, pp. 1-4.
2. H. Yu, J. Lu, and H. Zhu, "A design of smart home system based on augmented reality and internet of things," in 2019 4th International Conference on Mechanical, Control and Computer Engineering (ICMCCE), 2019, pp. 221-225.
3. T. Kim, H. Kim, and J. Lee, "Smart home control system using augmented reality," in 2019 International Conference on Electronics, Information, and Communication (ICEIC), 2019, pp. 1-3.
4. K. K. Mishra and P. Misra, "Augmented reality based home automation system using IoT," in 2021 IEEE International Conference on Computing, Electronics & Communications Engineering (IEEE ICCECE), 2021, pp. 326-329.
5. N. K. Yadav, "Augmented reality-based IoT home automation system," in 2021 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), 2021, pp. 1-6.

# 

# **APPENDICES**

**Click\_API.cs**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.Networking;

public class ClickUrl : MonoBehaviour

{

public string url;

public void open()

{

StartCoroutine(GetRequest(url));

}

IEnumerator GetRequest(string uri)

{

using (UnityWebRequest webRequest = UnityWebRequest.Get(uri))

{

// Request and wait for the desired page.

yield return webRequest.SendWebRequest()

}

}

}

**Get\_Data.cs**

using System.Collections;

using UnityEngine;

using UnityEngine.UI;

using UnityEngine.Networking;

public class Getdat : MonoBehaviour

{

public InputField tempInputField;

public InputField humInputField;

public Button refreshButton;

void Start()

{

// Attach the button click event to the function

refreshButton.onClick.AddListener(OnRefreshButtonClicked);

}

void OnDestroy()

{

// Remove the listener when the script is destroyed to avoid memory leaks

refreshButton.onClick.RemoveListener(OnRefreshButtonClicked);

}

public class Getdat : MonoBehaviour

{

public InputField tempInputField;

public InputField humInputField;

public void OnRefreshButtonClicked()

{

// Start the coroutine to get data from the API

StartCoroutine(GetDataCoroutine());

}

IEnumerator GetDataCoroutine()

{

// Set the loading text for input fields

tempInputField.text = "Loading...";

humInputField.text = "Loading...";

// Replace the URL with the actual API endpoint for temperature and humidity data

string tempUrl = "https://blynk.cloud/external/api/get?token=t9SRm0uM7lAQNHJ0bJ8JsFlaWRzapMxh&v4";

string humUrl = "https://blynk.cloud/external/api/get?token=t9SRm0uM7lAQNHJ0bJ8JsFlaWRzapMxh&v5";

// Send the UnityWebRequest to get temperature data

using (WWW tempRequest = new WWW(tempUrl))

{

yield return tempRequest;

if (!string.IsNullOrEmpty(tempRequest.error))

tempInputField.text = tempRequest.error;

else

{

// Set the temperature data to the input field

string temperatureWithSymbol = tempRequest.text + " \u2103";

tempInputField.text = temperatureWithSymbol;

}

}

// Send the UnityWebRequest to get humidity data

using (WWW humRequest = new WWW(humUrl))

{

yield return humRequest;

if (!string.IsNullOrEmpty(humRequest.error))

humInputField.text = humRequest.error;

else

{

// Set the humidity data to the input field

humInputField.text = humRequest.text;

}

}

**Slider.cs**

using UnityEngine;

using UnityEngine.UI;

using System.Collections;

using System.Collections.Generic;

using System.Text;

using UnityEngine.Networking;

public class SliderText : MonoBehaviour

{

public Slider slider; // Reference to the Slider component in the Unity editor

public string token ; // Blynk token

public string pinNumber; // Blynk virtual pin number

// URL format for the Blynk API

private string apiUrlFormat = "https://blynk.cloud/external/api/update?token={0}&v{1}={2}";

// Method to be called when the slider value changes

public void OnSliderValueChanged()

{

float sliderValue = slider.value; // Get the current value of the slider

// Build the URL with the token, pin number, and slider value

string apiUrl = string.Format(apiUrlFormat, token, pinNumber, sliderValue);

// Send a UnityWebRequest to the API endpoint

StartCoroutine(SendRequest(apiUrl));

}

private IEnumerator SendRequest(string url)

{

UnityWebRequest request = UnityWebRequest.Get(url);

yield return request.SendWebRequest();

if (request.result == UnityWebRequest.Result.ConnectionError || request.result == UnityWebRequest.Result.ProtocolError)

{

Debug.LogError("Error sending request: " + request.error);

}

else

{

Debug.Log("Request sent successfully.");

}

}

}

**Node.ino**

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266\_SSL.h>

#include <DHT.h>

#define RELAY\_PIN 5 // Relay pin

#define PWM\_PIN 16 // PWM pin

#define DHTPIN 4 // DHT11 pin

#define DHTTYPE DHT11 // DHT 11

// You should get Auth Token in the Blynk App.

// Go to the Project Settings (nut icon).

char auth[] = "t9SRm0uM7lAQNHJ0bJ8JsFlaWRzapMxh";

// Your WiFi credentials.

// Set password to "" for open networks.

char ssid[] = "Galaxy j";

char pass[] = "556677889900";

DHT dht(DHTPIN, DHTTYPE);

BlynkTimer timer;

void sendSensor()

{

float h = dht.readHumidity();

float t = dht.readTemperature(); // or dht.readTemperature(true) for Fahrenheit

if (isnan(h) || isnan(t))

{

Serial.println("Failed to read from DHT sensor!");

return;

}

Blynk.virtualWrite(V4, h);

Blynk.virtualWrite(V5, t);

}

BLYNK\_WRITE(V1)

{

int pinValue = param.asInt();

analogWrite(D1, pinValue);

Blynk.virtualWrite(V1, pinValue);

}

BLYNK\_WRITE(V3)

{

int pinValue = param.asInt();

analogWrite(D6, pinValue);

Blynk.virtualWrite(V3, pinValue);

}

void setup()

{

// Debug console

Serial.begin(115200);

Blynk.begin(auth, ssid, pass);

dht.begin();

timer.setInterval(1000L, sendSensor);

pinMode(PWM\_PIN, OUTPUT);

pinMode(RELAY\_PIN, OUTPUT);

}

void loop()

{

Blynk.run();

timer.run();

}