Institute of Engineering and

Technology (IET)

JK Lakshmipat University, Jaipur

EE1222: INTERNET OF THINGS PROJECT REPORT

Title: INTELLIHOUSE

Under the Supervision of:

Dr. Hanuman Prasad Agrawal

Divanshu Jain

Submitted by:

Pranay Kumar Jha (2021BTECH086)

Kushal Kumawat (2021BTECH066)

Vinit Nihala (2021BTECH119)

TABLE OF CONTENTS

ABSTRACT

Smart homes are becoming increasingly popular as people embrace the convenience and efficiency that they offer. Some of the key features of a smart home include:

Remote control: Smart homes allow you to remotely control various devices and appliances, such as lighting, thermostat, security systems, and home entertainment systems, using your smartphone, tablet, or computer.

Voice control: Many smart home devices come with voice control options that allow you to use voice commands to control your home's various systems and devices.

Automated routines: Smart home systems can be programmed to perform various tasks automatically, such as turning off lights and locking doors when you leave the house, or adjusting the thermostat based on your schedule and preferences.

Energy efficiency: Smart home devices can help you save on energy costs by automatically adjusting temperature settings, turning off lights when not in use, and monitoring energy usage.

Security: Smart homes offer advanced security features such as motion detectors, security cameras, and smart locks that can help keep your home safe and secure.

Entertainment: Smart homes can provide a personalized entertainment experience with features like streaming music, movies, and TV shows on demand, and integrating home theater systems.

Health and wellness: Smart homes can help improve your health and wellness by monitoring your sleep patterns, fitness levels, and nutrition, and providing personalized recommendations based on your data.

Overall, smart homes offer a range of benefits, from increased convenience and energy efficiency to enhanced security and entertainment options.

PROBLEM STATEMENT

Title: INTELLIHOUSE

Problem Statement:

As technology continues to advance, there is a growing demand for more efficient and intelligent solutions to enhance the quality of daily life. The conventional home management systems often lack integration and fail to adapt to the dynamic needs of modern households. The emergence of the Internet of Things (IoT) provides an opportunity to revolutionize home automation, and the project INTELLIHOUSE aims to address the following challenges:

Limited Interconnectivity: Existing home automation systems often struggle with seamless communication and integration of various smart devices. The lack of a unified platform hampers the potential for holistic control and monitoring.

User-Centric Adaptability: Many smart home systems lack the ability to learn and adapt to the preferences and habits of the residents. Customization is often a complex and time-consuming process, hindering the user experience.

Energy Inefficiency: Traditional home management systems may not effectively optimize energy usage, leading to unnecessary consumption and increased utility costs. There is a need for intelligent energy management to promote sustainability.

Security Concerns: As the number of connected devices increases, so does the vulnerability to security threats. Existing systems may not provide robust security measures, putting users at risk of unauthorized access and data breaches.

Complex Installation and Configuration: Setting up smart home devices can be challenging for users who are not tech-savvy. Simplifying the installation and configuration process is crucial to encourage widespread adoption.

Scalability Issues: Many current systems may struggle to scale effectively as new devices are added to the network. Ensuring a scalable infrastructure is vital for accommodating future advancements in smart home technology.

Data Privacy: With the increasing amount of personal data generated by smart homes, ensuring robust data privacy measures is imperative. Users should have confidence that their personal information is secure and not susceptible to misuse.

The INTELLIHOUSE project seeks to address these challenges by developing an innovative, user-friendly, and intelligent home automation system leveraging IoT technologies. The goal is to create a seamless and adaptive environment that enhances convenience, energy efficiency, and security while prioritizing user privacy. Through the integration of machine learning algorithms, intuitive interfaces, and comprehensive device compatibility, INTELLIHOUSE aims to redefine the smart home experience.

METHODOLOGY

Design and Implementation

Designing and implementing a **IntelliHouse** requires careful planning and consideration of various factors. Here are some steps to follow when designing and implementing a smart house:

Choose the right devices and systems: Choose the right devices and systems that meet your requirements and goals. Consider factors such as compatibility, reliability, and ease of use.

Plan the layout and installation: Plan the layout of the smart devices and systems, including their installation and configuration. Consider factors such as accessibility, aesthetics, and functionality.

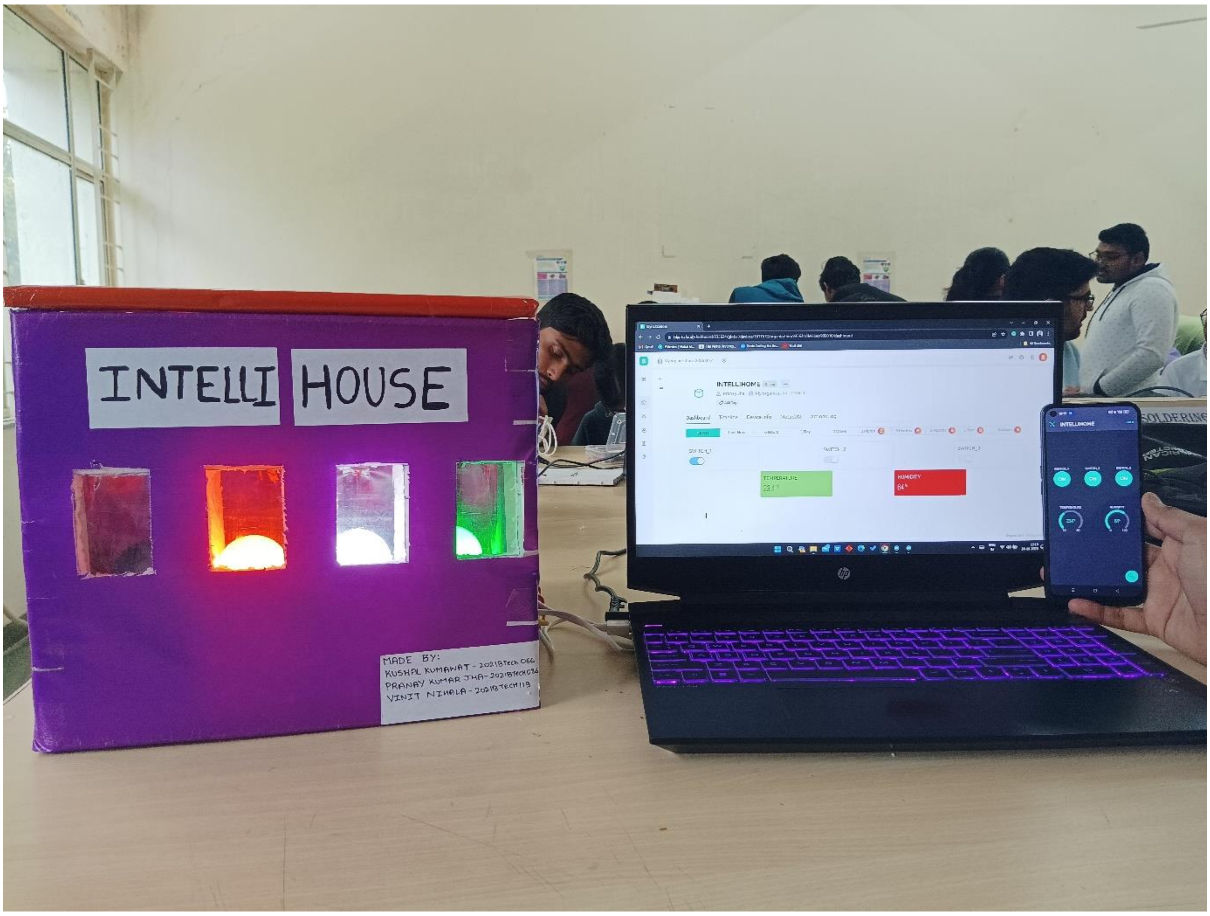
Integrate the devices and systems: Integrate the devices and systems to work together seamlessly. Ensure that all devices are compatible and can communicate with each other.

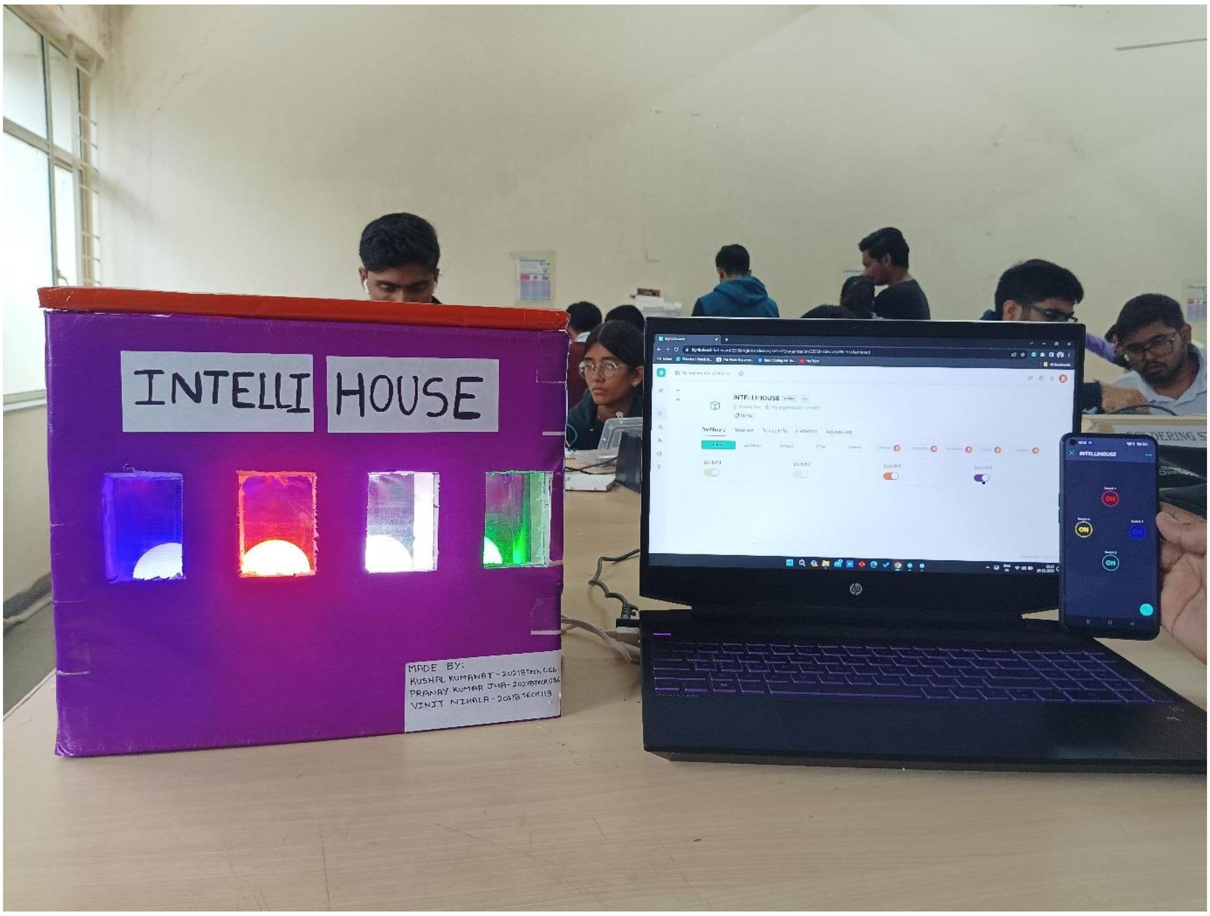
Configure the settings: Configure the settings for the smart devices and systems according to your preferences. Consider factors such as energy efficiency, security, and comfort.

Test and troubleshoot: Test the smart devices and systems and troubleshoot any issues that arise. Ensure that everything is working correctly and meets your requirements.

Maintain and upgrade: Maintain and upgrade the smart devices and systems regularly to ensure that they continue to function correctly and meet your changing requirements.

Overall, designing and implementing a **IntelliHouse** requires careful planning, selection of the right devices and systems, integration, configuration, testing, and maintenance. By following these steps, homeowners can create a **IntelliHouse** that meets their requirements and enhances their lifestyle.


INTELLI HOUSE
MADE BY:
KUSHAL KUMAWAT - 2021BTech066
PRANAY KUMAR JHA - 2021BTech086
VINIT NIHALA - 2021BTECH119
TEMPERATURE
HUMIDITY
INTELLIHOME


INTELLI HOUSE
MADE BY:
KUSHAL KUMAWAT - 2021BTech066
PRANAY KUMAR JHA - 2021BTech086
VINIT NIHALA - 2021BTECH119
INTELLIHOUSE
ON
ON
ON

Circuit design

The circuit design of a **IntelliHouse** will vary depending on the specific features and systems being installed. Here are some key considerations when designing the circuits for a **IntelliHouse**:

Power supply: The power supply for the smart devices and systems needs to be reliable and efficient. Consider installing backup power sources, such as batteries or generators, to ensure uninterrupted power supply.

Wiring: The wiring for the smart devices and systems needs to be installed properly to ensure safety and reliability. Consider using high-quality wiring materials that can handle the load and provide good conductivity.

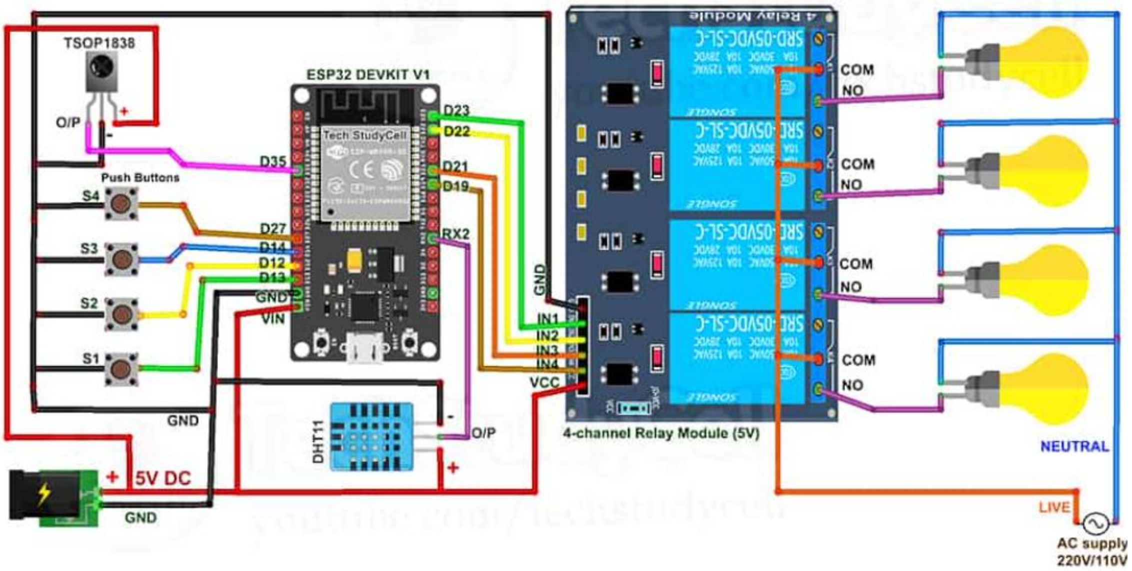
Control systems: IntelliHouse require control systems to automate the various functions. Consider using microcontrollers, sensors, and actuators to control and monitor the devices and systems.

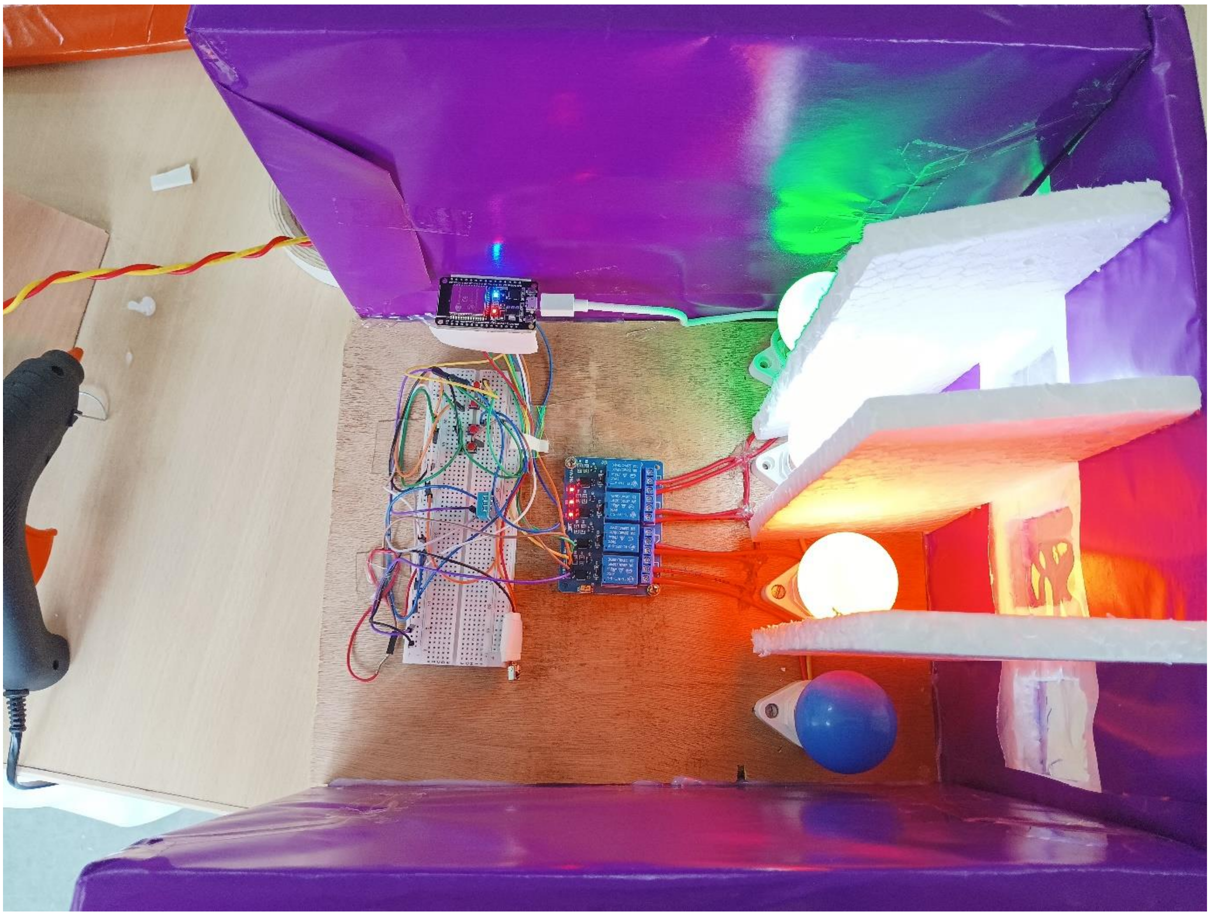
Communication protocols: The smart devices and systems need to communicate with each other, requiring the use of communication protocols such as Wi-Fi, Bluetooth, or Zigbee. Ensure that the protocols used are compatible with the devices and provide reliable communication.

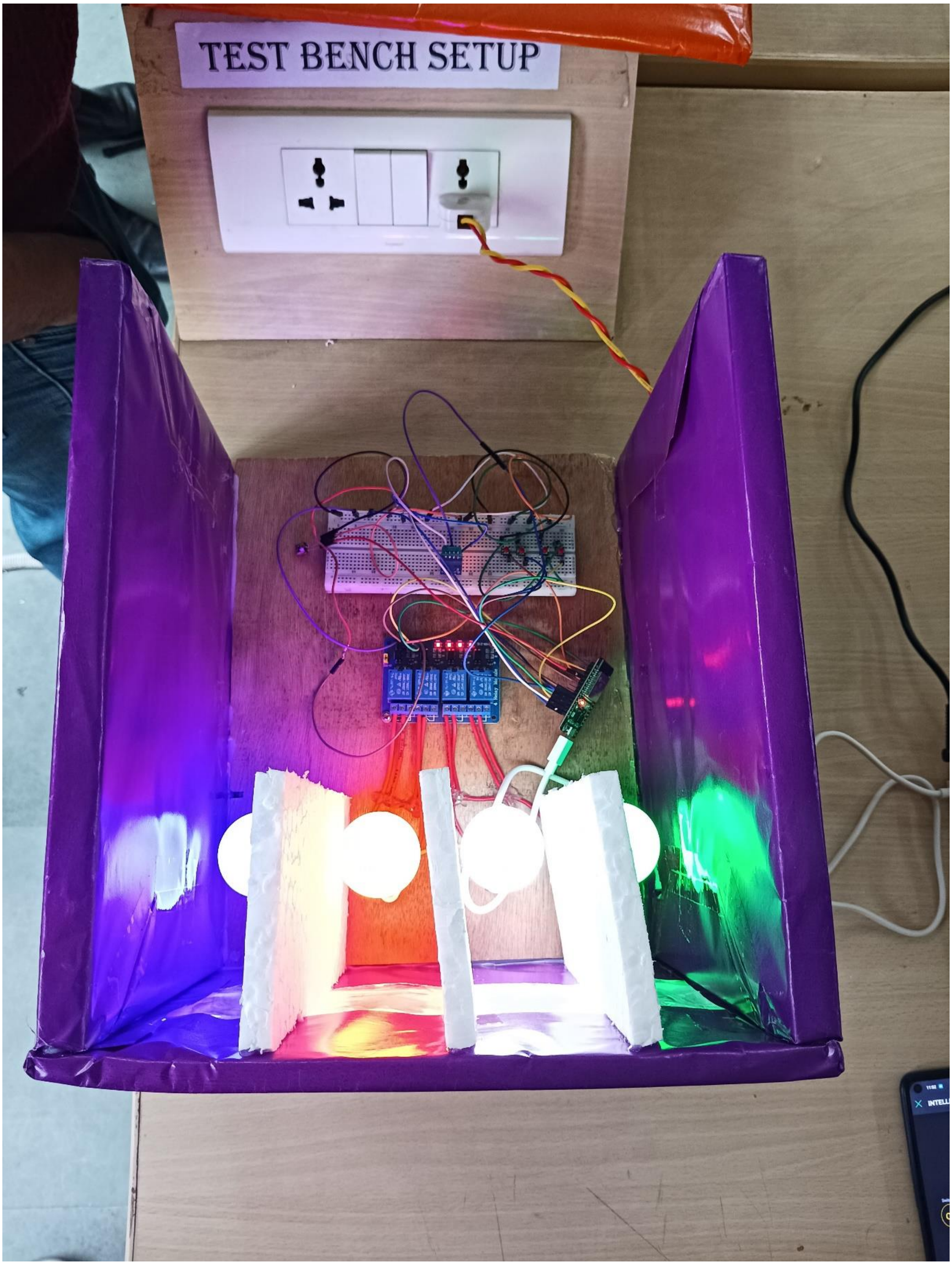
Security: The **IntelliHouse** circuits need to be designed with security in mind to prevent unauthorized access and hacking. Consider using encryption, firewalls, and other security measures to protect the circuits.

Integration: The circuits for the smart devices and systems need to be integrated and configured to work together seamlessly. Consider using software platforms and APIs to ensure compatibility and efficient communication.

Overall, the circuit design of a **IntelliHouse** requires careful consideration of power supply, wiring, control systems, communication protocols, security, and integration. By designing the circuits properly, homeowners can ensure that their smart home functions reliably, safely, and efficiently.


TSOP1838
O/P
+
-
Push Buttons
S4
S3
S2
S1
ESP32 DEVKIT V1
D23
D22
D35
D21
D19
D27
D14
D12
D13
GND
VIN
RX2
GND
IN1
IN2
IN3
IN4
VCC
4 Relay Module
4-channel Relay Module (5V)
COM
NO
COM
NO
COM
NO
COM
NO
NEUTRAL
LIVE
AC supply
220V/110V
DHT11
O/P
+
-
GND
+
5V DC
GND




TEST BENCH SETUP

Hardware Requirements

The hardware requirements for a **IntelliHouse** will depend on the specific features and systems being installed. However, here are some common hardware requirements for a basic **IntelliHouse** setup:

Smart devices: Smart devices are the core hardware components of a **IntelliHouse**. These devices include smart thermostats, smart lights, smart locks, smart cameras, and smart entertainment systems.

Sensors: Sensors are used to detect changes in the environment, such as motion, temperature, and humidity. These sensors provide input to the smart devices and systems, allowing them to adjust their functions accordingly.

Microcontrollers: Microcontrollers are used to control the various functions of the smart devices and systems. These controllers provide the logic for automation and can be programmed to respond to specific inputs and triggers.

Wi-Fi and Internet connectivity: **IntelliHouse** require Wi-Fi and internet connectivity to allow communication between the various devices and systems. Ensure that the internet connection is reliable and can handle the load of the smart devices.

Power supply: Smart devices and systems require a reliable and efficient power supply. Consider using backup power sources such as batteries or generators to ensure uninterrupted power supply.

Mounting hardware: Smart devices need to be mounted properly to ensure safety and functionality. Consider using mounting hardware such as brackets, screws, and anchors to secure the devices.

Overall, the hardware requirements for a **IntelliHouse** include smart devices, sensors, microcontrollers, Wi-Fi and internet connectivity, power supply, and mounting hardware. By ensuring that these hardware components are installed properly, homeowners can enjoy the benefits of a **IntelliHouse**.

Software Requirements

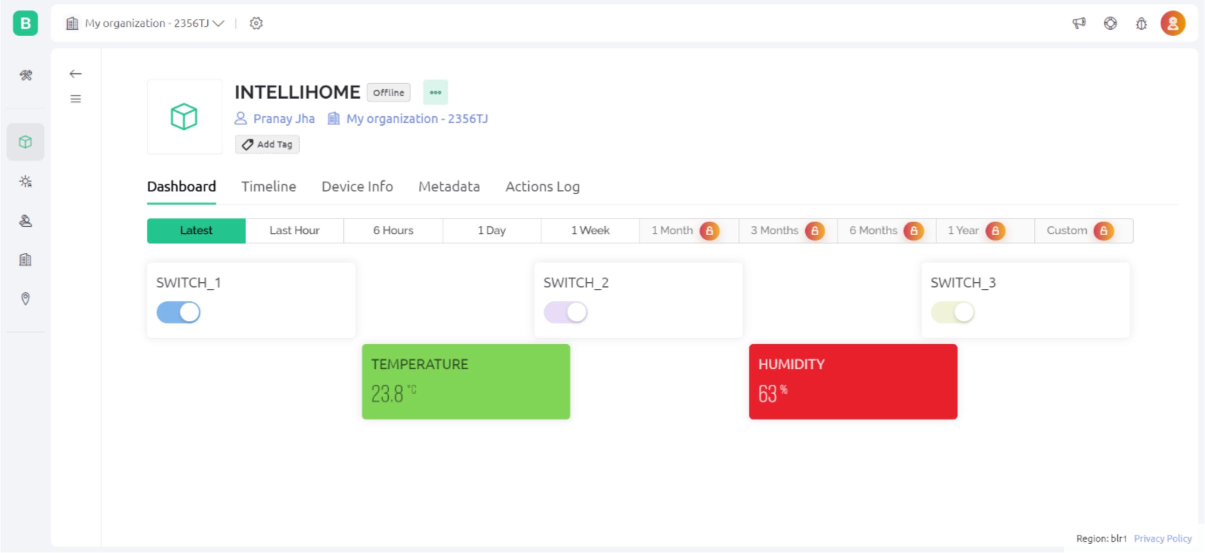
The software requirements for a **IntelliHouse** will depend on the specific features and systems being installed. However, here are some common software requirements for a basic smart house setup:

Mobile apps: Mobile apps are essential for controlling and monitoring smart devices and systems from a smartphone or tablet. Ensure that the mobile apps are compatible with the smart devices and systems being used.

Automation software: Automation software is used to program the smart devices and systems to perform specific functions. These programs can be set up to respond to triggers such as time of day, temperature changes, or motion detection.

Overall, the software requirements for a **IntelliHouse** include mobile apps, automation software, virtual assistants, Arduino software, security software, and cloud services. By ensuring that the software components are installed and configured properly, homeowners can enjoy the benefits of a smart house.

User Interface



INTELLIHOME Offline

Pranay Jha · My organization - 2356TJ

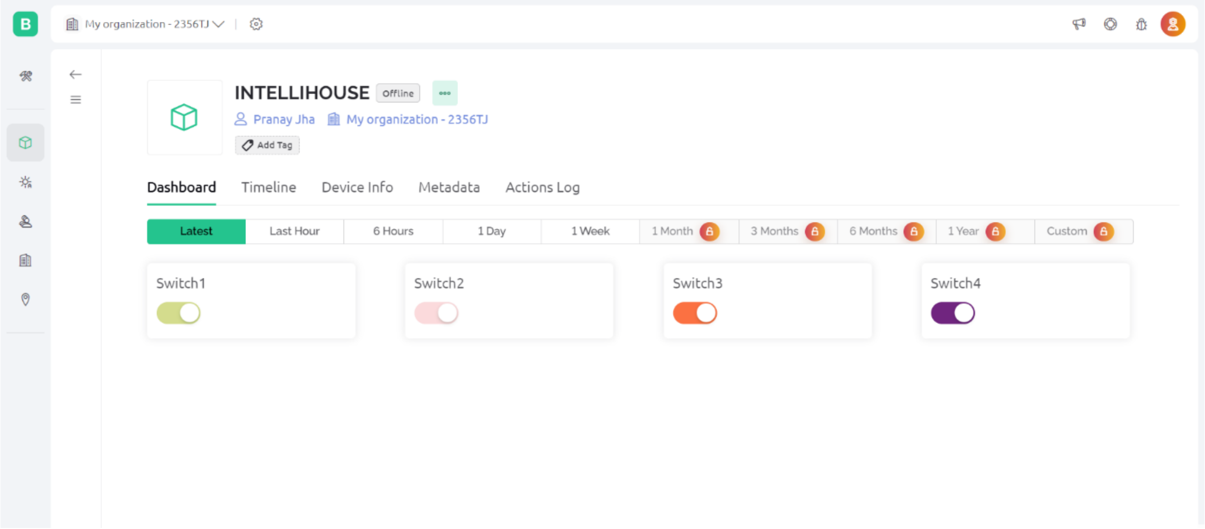
SWITCH_1

SWITCH_2

SWITCH_3

TEMPERATURE 23.8 °C

HUMIDITY 63 %



INTELLIHOUSE Offline

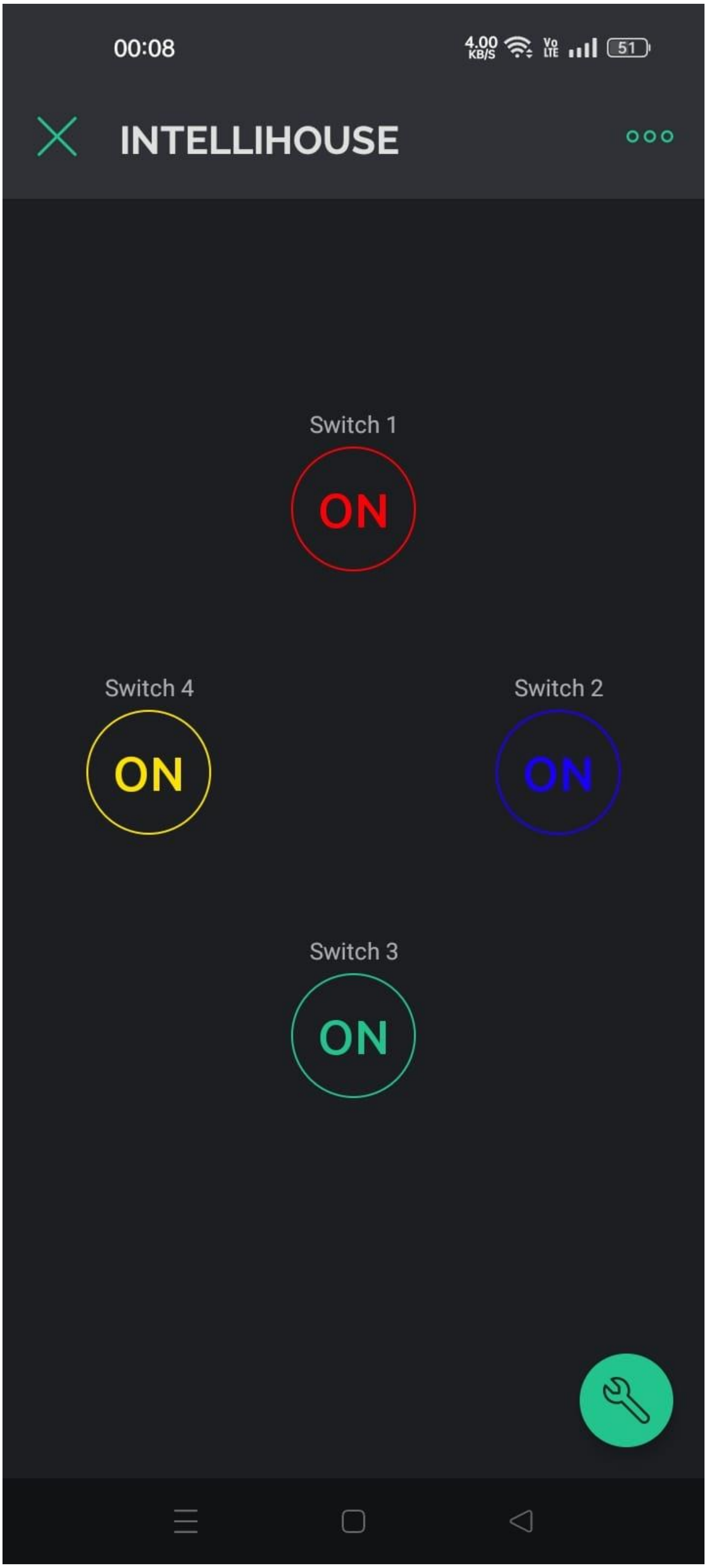
Pranay Jha · My organization - 2356TJ

Switch1

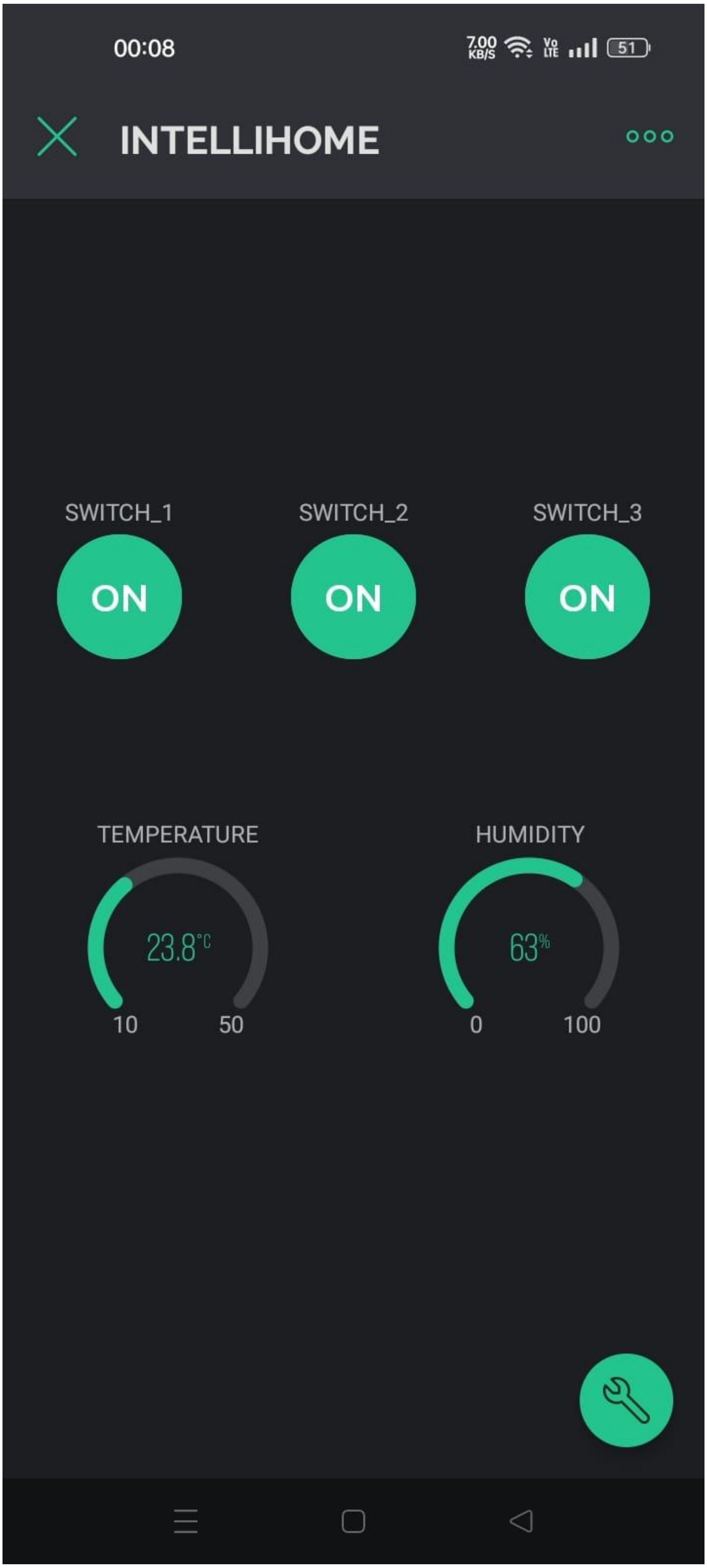
Switch2

Switch3

Switch4



00:08
4.00 KB/S
51
INTELLIHOUSE
Switch 1
ON
Switch 4
ON
Switch 2
ON
Switch 3
ON


00:08
7.00 KB/S
INTELLIHOME
SWITCH_1
ON
SWITCH_2
ON
SWITCH_3
ON
TEMPERATURE
23.8°C
10
50
HUMIDITY
63%
0
100

RESULTS AND DISCUSSION

The results and discussion of a **IntelliHouse** will depend on the specific features and systems being installed and the metrics being measured. Here are some examples of metrics that can be analyzed to evaluate the performance of a smart house:

Energy consumption: **IntelliHouse** can be designed to optimize energy consumption by using sensors, automation, and smart devices. Energy consumption can be measured and analyzed to identify areas for improvement and cost savings.

Temperature and climate control: **IntelliHouse** can be designed to control the temperature and climate based on occupancy, time of day, and weather conditions. Temperature and climate control can be measured and analyzed to ensure comfort and energy efficiency.

Security: **IntelliHouse** can be designed to provide enhanced security through the use of smart locks, cameras, and sensors. Security can be measured and analyzed to ensure that the smart huse is protected from unauthorized access and intrusions.

Convenience and comfort: **IntelliHouse** can be designed to provide convenience and comfort through the use of automation and smart devices. Convenience and comfort can be measured and analyzed to ensure that the smart house meets the needs and preferences of the homeowners.

Maintenance and repair: **IntelliHouse** can be designed to provide real-time monitoring and alerts for maintenance and repair needs. Maintenance and repair data can be analyzed to identify areas for improvement and cost savings.

Overall, the results and analysis of a **IntelliHouse** can provide insights into energy consumption, temperature and climate control, security, convenience and comfort, and maintenance and repair. By analyzing these metrics, homeowners can optimize the performance of their smart house and enjoy the benefits of a more efficient, secure, and comfortable living space.

CONCLUSIONS

In conclusion, **IntelliHouse** offer a wide range of benefits and advantages for homeowners. They provide automation, convenience, energy efficiency, security, and comfort through the use of smart devices, sensors, and software. With the rapid advancements in technology, **IntelliHouse** are becoming more affordable and accessible for homeowners.

However, there are also some limitations and challenges associated with **IntelliHouse**, including the need for reliable internet connectivity, potential security risks, and the complexity of installation and maintenance. These challenges can be addressed through proper planning, installation, and maintenance, as well as the use of professional services when needed.

Overall, **IntelliHouse** represent a promising direction for the future of residential living. As technology continues to evolve, it is likely that **IntelliHouse** will become even more advanced, efficient, and convenient for homeowners. By embracing the benefits of **IntelliHouse** , homeowners can enjoy a more comfortable, secure, and connected living space.

REFERENCES

https://www.electronicsforu.com/iot-projects-ideas

https://www.rcciit.org/students_projects/projects/ee/2020/GR3.pdf

https://sih.gov.in/sih2023PS

APPENDICES (CODE)

IntelliHouse:

```
//pip install esptool --upgrade
//esptool.py --chip esp32 --port COM5 erase_flash

#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

char auth[] = "ARbgnjKZdhZEu-hE3PHRnx5so0b0fKP4";
char ssid[] = "Pranay";
char pass[] = "12345678";

// Pins connected to the relay module
int relayPins[] = {23, 22, 21, 19}; // Adjust the pin numbers based on your ESP32 board

void setup() {
  Serial.begin(9600);
  Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);

  // Set relay pins as output
  for (int i = 0; i < 4; i++) {
    pinMode(relayPins[i], OUTPUT);
    digitalWrite(relayPins[i], HIGH); // Turn off all relays initially
  }
}

void loop() {
  Blynk.run();
}

// Blynk virtual pin handler
BLYNK_WRITE(V1) {
  int state = param.asInt();
  digitalWrite(relayPins[0], state);
}

BLYNK_WRITE(V2) {
  int state = param.asInt();
  digitalWrite(relayPins[1], state);
}

BLYNK_WRITE(V3) {
  int state = param.asInt();
  digitalWrite(relayPins[2], state);
}
```

```
BLYNK_WRITE(V4) {
  int state = param.asInt();
  digitalWrite(relayPins[3], state);
}
```

IntelliHome:

```
#include <WiFi.h>
#include <BlynkSimpleEsp32.h>
#include <DHT.h>

char ssid[] = "Pranay";
char pass[] = "12345678";

bool fetch_blynk_state = true;

#define DHTPIN            16 // D16 pin connected with DHT

// Uncomment whatever type you're using!
#define DHTTYPE DHT11     // DHT 11
//#define DHTTYPE DHT22   // DHT 22, AM2302, AM2321
//#define DHTTYPE DHT21   // DHT 21, AM2301

#define RelayPin1 23  // D23
#define RelayPin2 22  // D22
```

```
#define RelayPin3 21  // D21
#define RelayPin4 19  // D19

#define SwitchPin1 13  // D13
#define SwitchPin2 12  // D12
#define SwitchPin3 14  // D14
// #define SwitchPin4 27  // D27 (commented out as it is not used)

#define wifiLed   2   // D2

#define VPIN_BUTTON_1    V1
#define VPIN_BUTTON_2    V2
#define VPIN_BUTTON_3    V3
#define VPIN_TEMPERATURE V4
#define VPIN_HUMIDITY    V5

bool toggleState_1 = LOW;
bool toggleState_2 = LOW;
bool toggleState_3 = LOW;

int wifiFlag = 0;
float temperature1 = 0;
float humidity1 = 0;

char auth[] = "fjgqhbLxJPaop0BqR2YRAcWEAemKjVcp";
```

```
BlynkTimer timer;
DHT dht(DHTPIN, DHTTYPE);

// Additional setup for relay pins
int relayPins[] = {RelayPin1, RelayPin2, RelayPin3, RelayPin4};

void toggleRelay(int virtualPin, int relayPin, BlynkParam param) {
  bool toggleState = param.asInt();
  digitalWrite(relayPin, !toggleState);
  Blynk.virtualWrite(virtualPin, toggleState);
}

BLYNK_WRITE(VPIN_BUTTON_1) {
  toggleRelay(VPIN_BUTTON_1, RelayPin1, param);
}

BLYNK_WRITE(VPIN_BUTTON_2) {
  toggleRelay(VPIN_BUTTON_2, RelayPin2, param);
}

BLYNK_WRITE(VPIN_BUTTON_3) {
  toggleRelay(VPIN_BUTTON_3, RelayPin3, param);
}
```

```
// BLYNK_WRITE(VPIN_BUTTON_4) {
//   toggleRelay(VPIN_BUTTON_4, RelayPin4, param);
// }

void checkBlynkStatus() {
  bool isconnected = Blynk.connected();
  if (isconnected == false) {
    wifiFlag = 1;
    Serial.println("Blynk Not Connected");
    digitalWrite(wifiLed, LOW);
  }
  if (isconnected == true) {
    wifiFlag = 0;
    if (!fetch_blynk_state) {
      Blynk.virtualWrite(VPIN_BUTTON_1, toggleState_1);
      Blynk.virtualWrite(VPIN_BUTTON_2, toggleState_2);
      Blynk.virtualWrite(VPIN_BUTTON_3, toggleState_3);
    }
    digitalWrite(wifiLed, HIGH);
  }
}

BLYNK_CONNECTED() {
  if (fetch_blynk_state) {
    Blynk.syncVirtual(VPIN_BUTTON_1);
    Blynk.syncVirtual(VPIN_BUTTON_2);
    Blynk.syncVirtual(VPIN_BUTTON_3);
  }
  Blynk.syncVirtual(VPIN_TEMPERATURE);
  Blynk.syncVirtual(VPIN_HUMIDITY);
}

void readSensor() {
  delay(3000); // Increase delay before reading the sensor
  float h = dht.readHumidity();
  float t = dht.readTemperature();

  if (isnan(h) || isnan(t)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  else {
    humidity1 = h;
    temperature1 = t;
  }
}

void sendSensor() {
  readSensor();
  Blynk.virtualWrite(VPIN_HUMIDITY, humidity1);
  Blynk.virtualWrite(VPIN_TEMPERATURE, temperature1);
}

void setup() {
  Serial.begin(9600);

  Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);

  // Set relay pins as output
  for (int i = 0; i < 4; i++) {
    pinMode(relayPins[i], OUTPUT);
    digitalWrite(relayPins[i], HIGH); // Turn off all relays initially
  }

  pinMode(wifiLed, OUTPUT);

  pinMode(SwitchPin1, INPUT_PULLUP);
  pinMode(SwitchPin2, INPUT_PULLUP);
  pinMode(SwitchPin3, INPUT_PULLUP);
  // pinMode(SwitchPin4, INPUT_PULLUP);

  digitalWrite(RelayPin1, !toggleState_1);
  digitalWrite(RelayPin2, !toggleState_2);
  digitalWrite(RelayPin3, !toggleState_3);
  // digitalWrite(RelayPin4, !toggleState_4);
  dht.begin();
  digitalWrite(wifiLed, LOW);

  timer.setInterval(2000L, checkBlynkStatus);
  timer.setInterval(1000L, sendSensor);
  delay(1000);

  if (!fetch_blynk_state) {
    Blynk.virtualWrite(VPIN_BUTTON_1, toggleState_1);
    Blynk.virtualWrite(VPIN_BUTTON_2, toggleState_2);
    Blynk.virtualWrite(VPIN_BUTTON_3, toggleState_3);
  }
}

void loop() {
  Blynk.run();
  timer.run();
}
```