TITLE: Smart Home Automation System with Remote Access

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

Submitted by

A.VAMSI VISWESWARA RAO	**(A22126511002)**
B.LIKHITHA	**(A22126511006)**
B.GOWRI AMRUTHA	**(A22126511007)**
D.VISHNUVANDANA	**(A22126511016)**
MD.AARIF SHARIEF	**(A22126511034)**

Under The Guidance

Of

MR DEEPA, PHD

ASS PROFESSOR

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES (A)

(Affiliated to AU, Approved by AICTE and Accredited by NBA &NAAC with A$^+$)

SANGIVALASA, BHEEMILI MANDAL, VISAKHAPATNAM DIST, A.P.

2022-2026

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES

(***Affiliated to AU, Approved by AICTE and Accredited by NBA&NAAC with A^{+}***)

SANGIVALASA, BHEEMILI MANDAL, VISAKHAPATNAM DIST, A.P.

DEPARTMENT OF INFORMATION TECHNOLOGY

PRAGNANAM BRAHMA
ANITS

CERTIFICATE

This is to certify that this project work entitled **"SAMRT HOME AUTOMATION WITH REMOTE ACCESS"** is the bonafide work being submitted by **A.VAMSI VISWESWARA RAO (A22126511002), B.LIKHITHA (A22126511006), B.GOWIRI AMURUTHA (A22126511007), D.VISHNUVANDANA (A22126511016),MD. AARIF SHARIEF (A22126511034)** of third year in Bachelor of Technology in Information Technology in ANITS (Approved by AICTE, affiliated to Andhra University and accredited by NBA, NAAC). It is a record of bonafide work carried out under the esteemed guidance of **Deepa, Assistant Professor**, Dept. of EEE during academic year 2024-2025.

Internal Examiners

Mrs. B. Deepa, Asst. Professor, Dept of ECE

Mr. N. Chaitanya, Dept. of IT

Mr. G. Ravindranath, Dept. of IT

Dr.M.RekhaSundari

Professor and Head of the Dept,

Departmentof IT

ANITS

ACKNOWLEDGEMENT

It is with immense pleasure that we express my sincere sum of gratitude and humble appreciation to Asst. Professor **Deepa**, Department of Electrical and Electronics Engineering, for his invaluable guidance, whole-hearted co-operation, constructive criticism and continuous encouragement in the preparation of thesis. Without his support and guidance, the present work wouldn't be happening.

We would like to extend our gratitude for dr. **Rekha Sundari**, Head of the Department of information technology for providing us with all the facilities and required areas of improvement. We sincerely thank for his exemplary guidance and encouragement. We are very much thankful to the Principal **krishna** and Management of Anil Neerukonda Institute of Technology & Sciences, Sangivalasa, for their encouragement and cooperation to carry out this work.

With sincere regards,

A.VAMSI VISWESWARA RAO	**(A22126511002)**
B.LIKHITHA	**(A22126511006)**
B.GOWRI AMRUTHA	**(A22126511007)**
D.VISHNUVANDANA	**(A22126511016)**
MD.AARIF SHARIEF	**(A22126511034)**

Table of Contents

Smart Home Automation System with Remote Access

1.Problem statement:

"Smart Home Automation System with Remote Access"

• Description: Use NodeMCU to control appliances such as lights, fans, and ACs remotely via a mobile app or web interface. Integrate motion sensors and a temperature sensor to automate the system.

• Application: Homes, apartments, and small offices(server rooms),ware houses,factories.

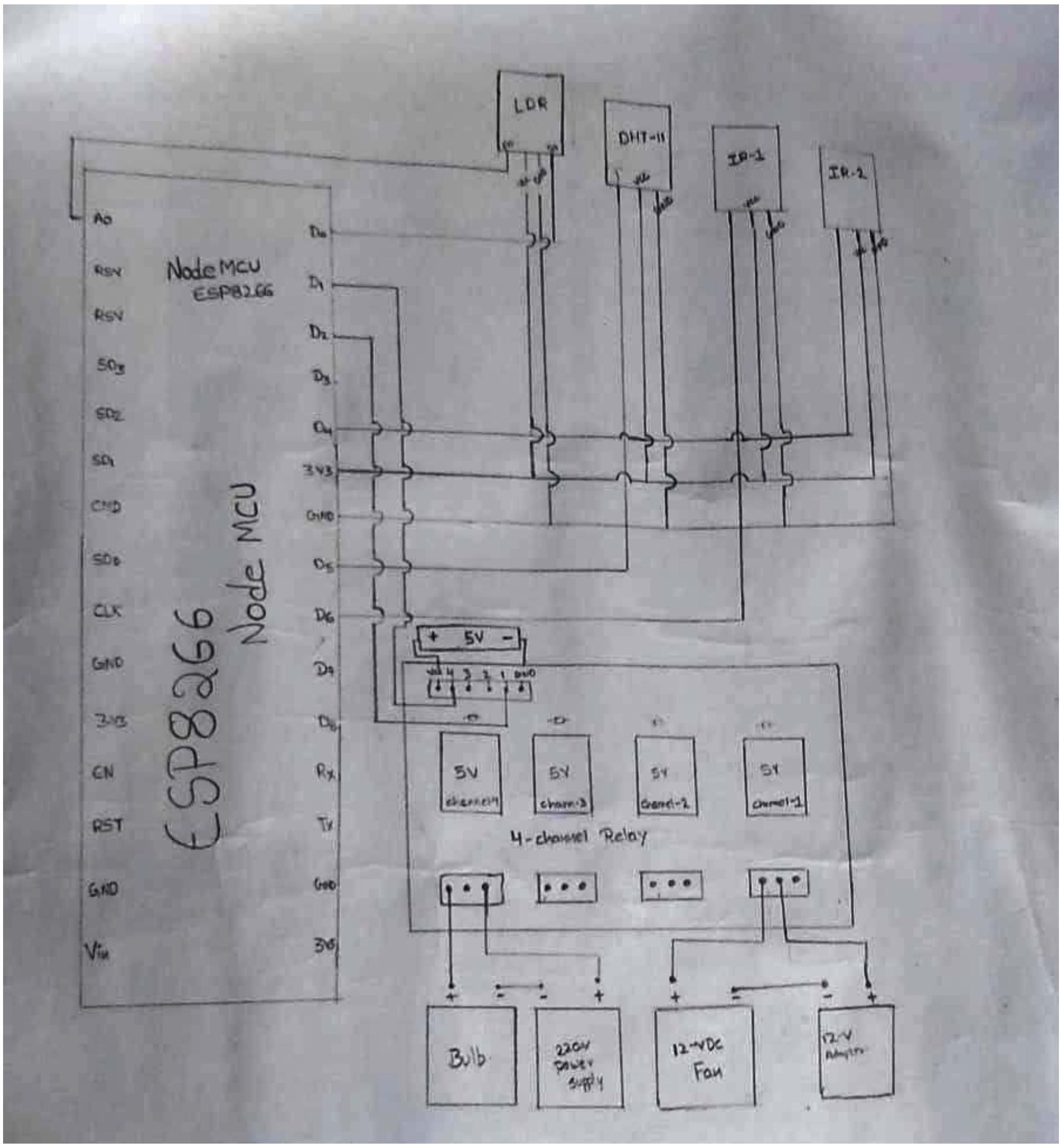
2. Abstract:

This project presents the design and implementation of a **Smart Home Automation System with Remote Access**, enabling users to control household appliances such as bulbs and fans using a mobile application through **Blynk**. The core of the system is built around the **NodeMCU (ESP8266)** microcontroller, which connects to the internet and facilitates communication between sensors and actuators.

The system integrates various sensors including the **DHT11 temperature and humidity sensor**, **LDR (Light Dependent Resistor)** for ambient light detection, and **IR sensors** for object detection. Additionally, a **relay module** is used to control high-voltage appliances like lights and fans. The system supports both **manual control** (through the Blynk app) and **automatic control**, where appliances respond to sensor inputs — for example, turning on the fan if the temperature exceeds a certain threshold, or switching off lights based on ambient light detected by the LDR.

This smart automation not only enhances **user convenience and comfort**, but also contributes to **energy efficiency and home security**. Real-time data monitoring and control from anywhere make the system scalable and practical for modern home environments.

3. Circuit Diagram :



4.Apparatus Required:

Components List

- NodeMCU (ESP8266)
- DHT11 Sensor
- LDR
- IR Sensor (2x)
- 4-Channel Relay Module
- Bulb
- 12V DC Fan
- Jumper Wires
- Breadboard
- 5V Power Supply
- Mobile Phone with Blynk App
- 12V adapter

5. Methodology Followed

System Design

The system is designed to automate home appliances using NodeMCU (ESP8266) as the central controller. The design integrates sensors such as LDR, DHT11 (for temperature and humidity), and IR sensors for automation based on environmental conditions. Outputs are controlled through a 4-channel relay which operates appliances like a bulb and fan. Manual control is enabled via the Blynk mobile application, while automatic control depends on real-time sensor data.

Hardware Connections

The smart home automation system is built using the NodeMCU ESP8266 microcontroller, multiple sensors, and a relay module to control home appliances like a bulb and a fan. The components are connected as follows:

1. NodeMCU (ESP8266) – Central Controller

- Acts as the main processing unit.
- Provides Wi-Fi connectivity and controls all inputs and outputs.
- Powered via USB (5V supply).

2. Sensor Connections

Sensor	**Pin on NodeMCU**	**Function**
LDR	A0	Detects ambient light intensity
DHT11	D5	Measures temperature and humidity

IR Sensor 1	D6	Detects motion (zone 1)
IR Sensor 2	D7	Detects motion (zone 2)

- All sensors are connected to **5V and GND** from the NodeMCU or external 5V source.

3. Relay Module (4-Channel)

Used to control high-voltage devices like bulbs and DC fans. Relays are triggered by digital pins on the NodeMCU.

Relay Channel	**NodeMCU Pin**	**Connected Appliance**
Relay 1	D1	Bulb (220V AC)
Relay 2	D2	12V DC Fan
Relay 3	D3	(Optional future use)
Relay 4	D0	(Optional future use)

Relay VCC connected to 5V supply.

- **Relay GND** connected to NodeMCU GND.
- **Relay IN1–IN4** connected to digital output pins D1–D0.

4. Appliances

- **Bulb (220V AC):** Connected to Relay 1 output terminals, powered by the main AC supply.
- **Fan (12V DC):** Connected to Relay 2 output and powered via 12V adapter.

- Appliances are isolated from the microcontroller using relay channels for safety.

5. Power Supply

- **NodeMCU** is powered via USB (5V).
- **Relay module and sensors** share a regulated **5V external power source**.
- **Fan** uses a separate **12V adapter**.
- Proper grounding is maintained across all components.

Blynk App:

The **Blynk App** is used to remotely control and monitor appliances like lights, fans, and ACs through a smartphone. It connects with the **NodeMCU** via the internet, allowing real-time data display (e.g., temperature, motion) and remote switching using buttons, sliders, and other widgets. Blynk also supports automation rules and push notifications, making it ideal for both home and light industrial automation systems.

HOW TO USE :

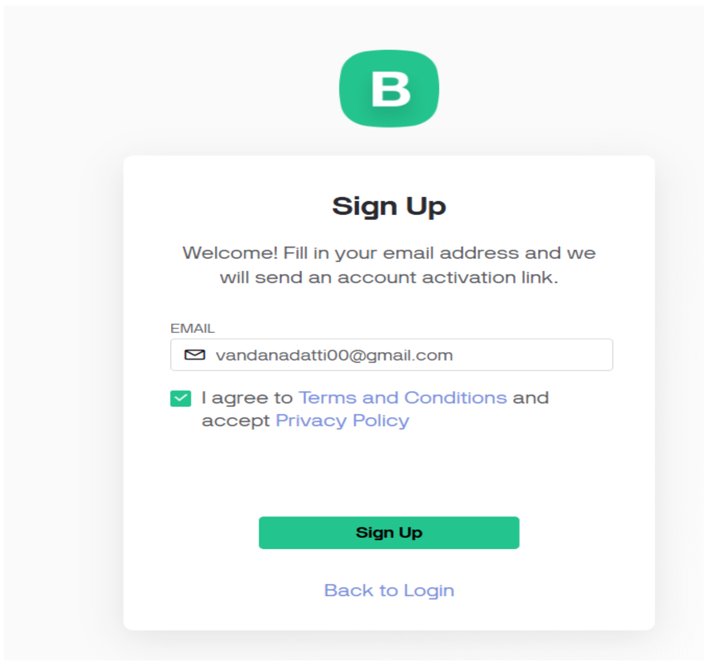
The Blynk App is used to remotely control and monitor devices through a smartphone. To use it, download the Blynk IoT app, create an account, and start a new project by selecting your hardware (like NodeMCU/ESP8266) and Wi-Fi connection. You'll get an Auth Token, which is added to your Arduino code. Inside the app, you can add widgets (like buttons or gauges) by tapping the “+” icon, then assign virtual pins (e.g., V0, V1) to each widget. These virtual pins allow your NodeMCU to send or receive data from the app, making it easy to control appliances or display sensor readings in real-time.

Blynk App Setup

- **Blynk Application** is used for remote monitoring and manual control.

  Intial setup:

  1.login/signup



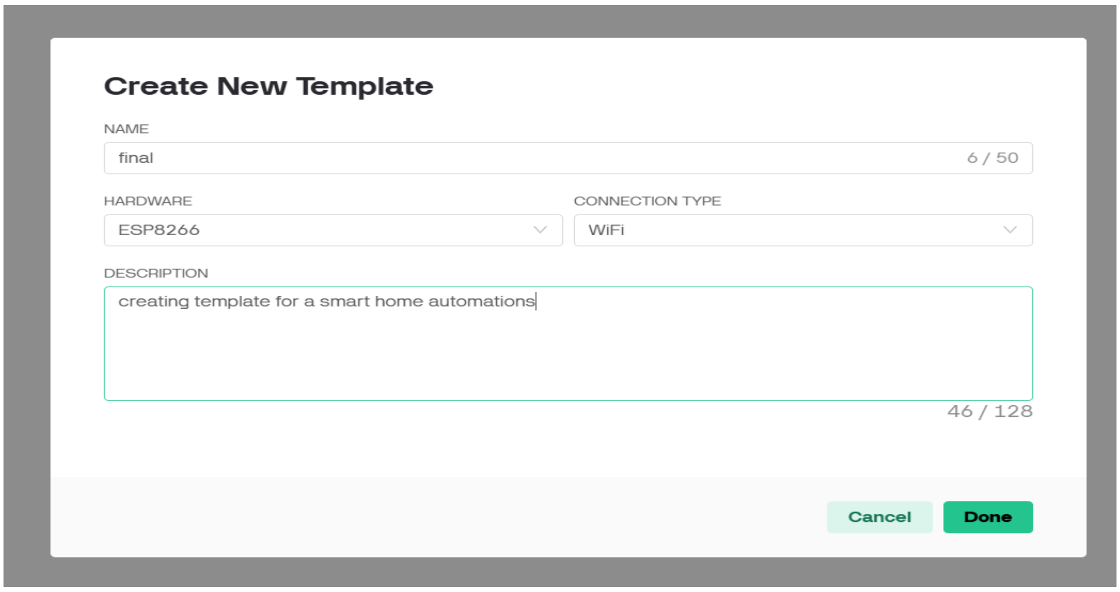
Sign Up

Welcome! Fill in your email address and we will send an account activation link.

EMAIL
vandanadatti00@gmail.com

✓ I agree to Terms and Conditions and accept Privacy Policy

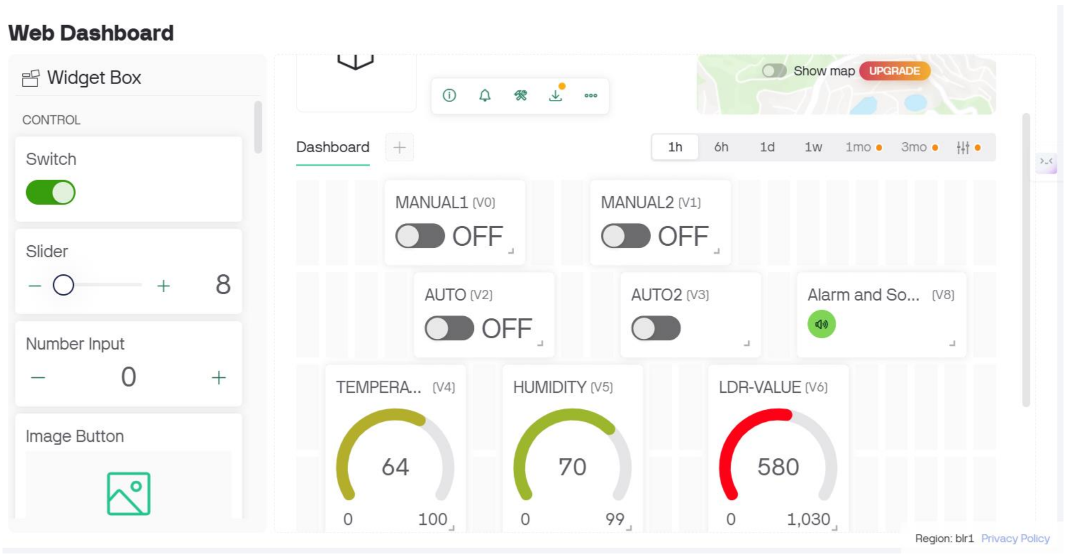
2.create template:



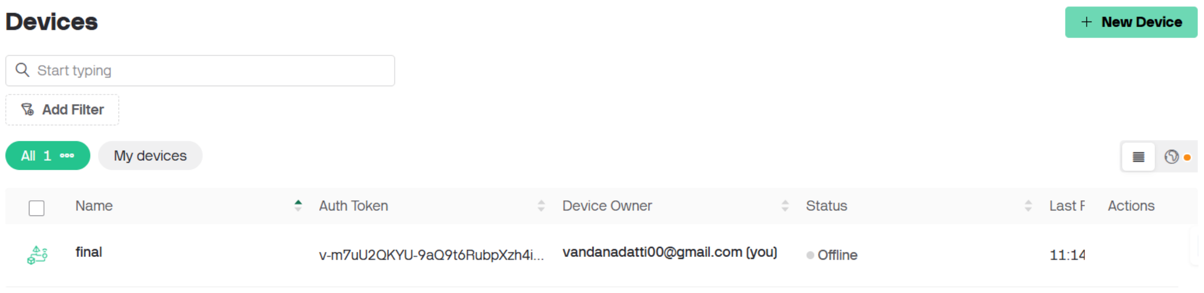
Create New Template

NAME	final (6 / 50)
HARDWARE	ESP8266
CONNECTION TYPE	WiFi
DESCRIPTION	creating template for a smart home automations (46 / 128)

3.go to web dashboard and add the widgets and give virtual pins



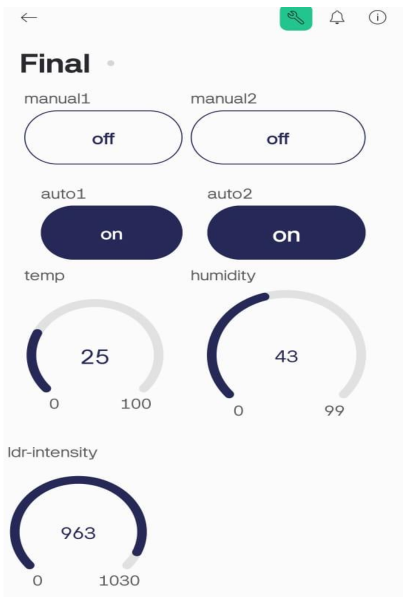
4.go to devices and add new device



Name	Auth Token	Device Owner	Status	Last F	Actions
final	v-m7uU2QKYU-9aQ9t6RubpXzh4i...	vandanadatti00@gmail.com (you)	Offline	11:14	

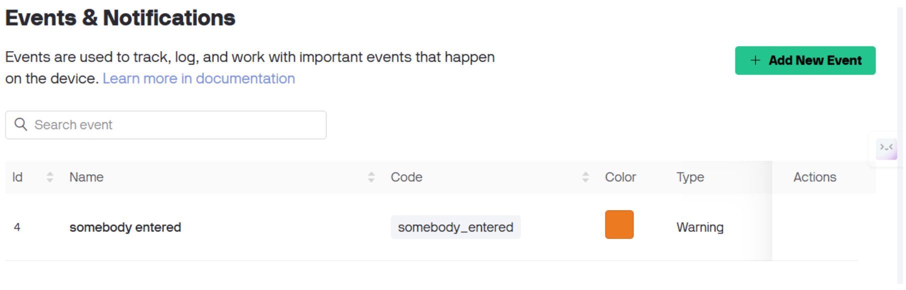
5.open your mobile and add widgets

-login your mail and edit mobile dashboard



- Virtual pins:
  - **V0** – Manual control for fan
  - **V1** – Manual control for bulb
  - **V2** – auto control for fan
  - **V3** – auto control for bulb
  - **V4** – Display Temperature
  - **V5**- Display Humidity
  - **V6**- Display LDR value
  - **V7**-alarm and sound

6. Event and notifications:



Events & Notifications

Events are used to track, log, and work with important events that happen on the device. Learn more in documentation

Id	Name	Code	Color	Type	Actions
4	somebody entered	somebody_entered	🟧	Warning	

- The app is linked to the NodeMCU using Auth Token, Wi-Fi credentials, and appropriate widgets (Buttons, Gauge, Value Display).

Working and Automation Logic

- **LDR** checks brightness levels: If it's dark and manual override is off, the bulb turns ON.
- **IR Sensors** detect motion: Can be used for triggering lights or alerts.
- **DHT11** monitors room temperature: If temperature exceeds threshold and motion is detected, the fan is automatically turned ON.
- **Relay module** acts as a switch between NodeMCU and appliances.
- **Manual override via Blynk** allows users to control appliances from anywhere.
- Sensor values are updated to **ThingSpeak**, providing visual plots for analysis and record.

Blynk Auth Token & Virtual Pin Setup

When you create a new project in the Blynk app, it generates a unique **Auth Token** and sends it to your email. This token connects your **NodeMCU (or other hardware)** to the Blynk cloud.

To use virtual pins with the token:

1. **Copy the Auth Token** from your Blynk app.
2. **Paste it into your Arduino code**, like this:

```
#define BLYNK_TEMPLATE_ID "TMPL35u8RrJBJ"

#define BLYNK_TEMPLATE_NAME "final2"

#define BLYNK_AUTH_TOKEN "q77QBX6SoF00ZwJh67XiCBeJb5xxh6i8”
```

6.Source code:

```
#define BLYNK_TEMPLATE_ID "TMPL35u8RrJBJ"

#define BLYNK_TEMPLATE_NAME "final2"

#define BLYNK_AUTH_TOKEN "q77QBX6SoF00ZwJh67XiCBeJb5xxh6i8"


#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

#include <DHT.h>

#include <ESP8266HTTPClient.h>


// WiFi credentials

char ssid[] = "Aarif";

char pass[] = "05220406";


// ThingSpeak API Key

String writeAPIKey = "IV32ZE7W4H74FT0I";


// Sensor Pins

#define DHTPIN 14        // D5 = GPIO14

#define DHTTYPE DHT11

#define LDRPIN A0        // Analog pin A0

#define IR1_PIN 12       // D6 = GPIO12 (used with DHT to glow LED1)

#define IR2_PIN 2        // D4 = GPIO2 (used for notification + alarm)


#define LED1_PIN 4       // D2 = GPIO4 (DHT + IR1 control)

#define LED2_PIN 5       // D1 = GPIO5 (LDR control)


// Virtual Pins

#define VPIN_TEMP V4
```

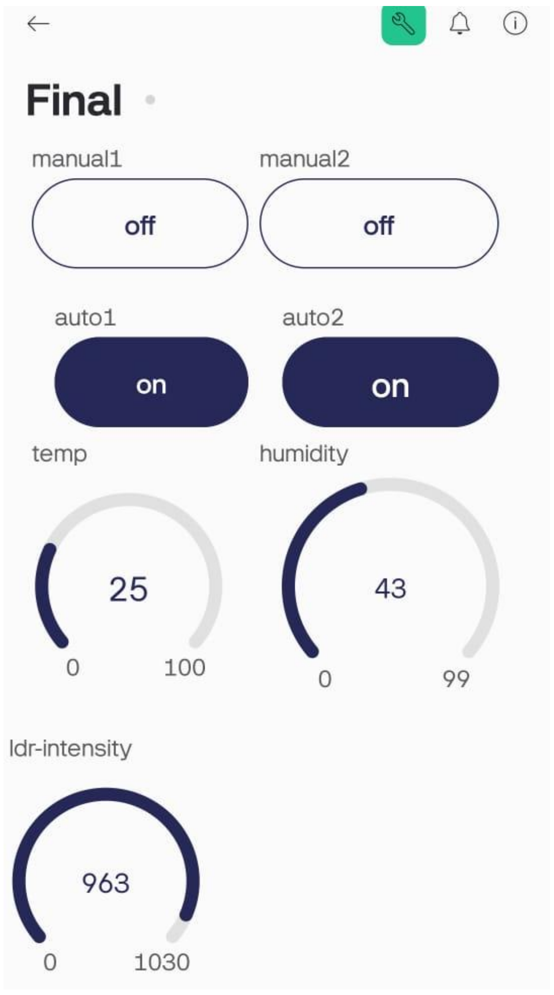
```
#define VPIN_HUM V5
#define VPIN_LDR V6
#define MANUAL1_PIN V0
#define AUTO1_PIN V2
#define MANUAL2_PIN V1
#define AUTO2_PIN V3
#define ALARM_VPIN V7

DHT dht(DHTPIN, DHTTYPE);
BlynkTimer timer;

// Control flags
bool manual1 = false;
bool auto1 = true;
bool manual2 = false;
bool auto2 = true;

int motionFlag = 0;  // To avoid repeated alerts from IR2

// Blynk virtual pin write handlers
BLYNK_WRITE(MANUAL1_PIN) { manual1 = param.asInt(); }
BLYNK_WRITE(AUTO1_PIN)   { auto1   = param.asInt(); }
BLYNK_WRITE(MANUAL2_PIN) { manual2 = param.asInt(); }
BLYNK_WRITE(AUTO2_PIN)   { auto2   = param.asInt(); }

// Sensor reading and device control
void sendSensor() {
  float temp = dht.readTemperature();
  float hum = dht.readHumidity();
```

```
int ldrValue = analogRead(LDRPIN);
int ir1Motion = digitalRead(IR1_PIN);  // LOW = motion
int ir2Motion = digitalRead(IR2_PIN);  // LOW = motion

// Send to Blynk
if (!isnan(temp) && !isnan(hum)) {
  Blynk.virtualWrite(VPIN_TEMP, temp);
  Blynk.virtualWrite(VPIN_HUM, hum);
}
Blynk.virtualWrite(VPIN_LDR, ldrValue);

// Debug print
Serial.print("🌡 Temp: "); Serial.print(temp);
Serial.print(" °C | 💧 Hum: "); Serial.print(hum);
Serial.print(" % | IR1: "); Serial.print(ir1Motion == LOW ? "Motion" : "No");
Serial.print(" | 💡 LDR: "); Serial.print(ldrValue);

// ---------- LED1 Control (DHT + IR1) ----------
bool led1State = false;
if (auto1) {
  if (temp > 20.0 && ir1Motion == LOW) {
    led1State = false;
    Serial.print(" | 🔁 AUTO1: LED1 ON");
  } else {
    led1State = true;
    Serial.print(" | 🔁 AUTO1: LED1 OFF");
  }
}
if (manual1) {
  led1State = !led1State;
  Serial.print(" | ✋ MANUAL1 Override: LED1 ");
  Serial.print(led1State ? "FORCED ON" : "FORCED OFF");
}
digitalWrite(LED1_PIN, led1State ? HIGH : LOW);

// ---------- LED2 Control (LDR) ----------
bool led2State = false;
int ldrThreshold = 500;
if (auto2) {
  if (ldrValue < ldrThreshold) {
    led2State = true;
    Serial.print(" | 🔁 AUTO2: LED2 ON");
  } else {
    Serial.print(" | 🔁 AUTO2: LED2 OFF");
  }
}
if (manual2) {
  led2State = !led2State;
  Serial.print(" | ✋ MANUAL2 Override: LED2 ");
  Serial.print(led2State ? "FORCED ON" : "FORCED OFF");
}
digitalWrite(LED2_PIN, led2State ? HIGH : LOW);
Serial.println();

// ---------- Send to ThingSpeak ----------
if (WiFi.status() == WL_CONNECTED) {
  WiFiClient client;
  HTTPClient http;
```

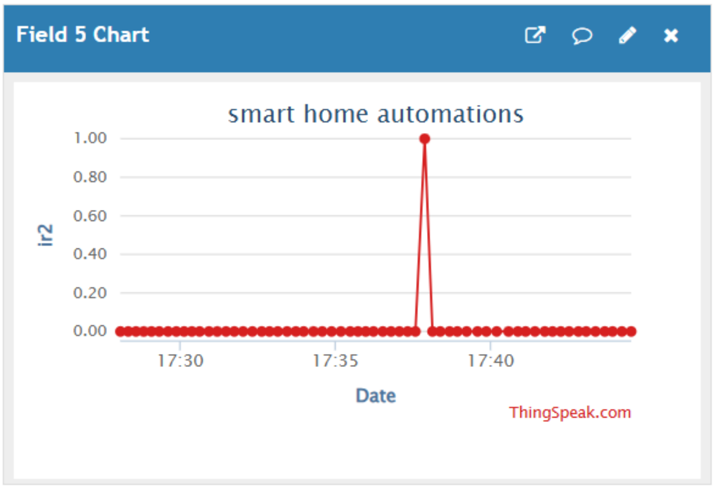
```
    String url = "http://api.thingspeak.com/update?api_key=" + writeAPIKey +
                 "&field1=" + String(temp) +
                 "&field2=" + String(hum) +
                 "&field3=" + String(ldrValue) +
                 "&field4=" + String(ir1Motion == LOW ? 1 : 0) +
                 "&field5=" + String(ir2Motion == LOW ? 1 : 0);

    http.begin(client, url);
    int httpResponseCode = http.GET();
    http.end();

    Serial.print("📶 ThingSpeak: ");
    Serial.println(httpResponseCode == 200 ? "Updated" : "Failed");
  }
}

// Alert via IR2
void notifyMotion() {
  int motion = digitalRead(IR2_PIN);  // LOW = motion
  if (motion == LOW && motionFlag == 0) {
    Serial.println("🚨 IR2 Motion Detected!");
    Blynk.virtualWrite(ALARM_VPIN, 1);
    Blynk.logEvent("evaroo_vacharaa_babuu", "🚨 evaroo vacharuuu raa babbuuuu!");
    motionFlag = 1;
  }
  else if (motion == HIGH && motionFlag == 1) {
    Serial.println("✅ IR2: No Motion.");
    Blynk.virtualWrite(ALARM_VPIN, 0);
    motionFlag = 0;
  }
}
void setup() {
  Serial.begin(115200);
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
  dht.begin();
  pinMode(IR1_PIN, INPUT);
  pinMode(IR2_PIN, INPUT);
  pinMode(LED1_PIN, OUTPUT);
  pinMode(LED2_PIN, OUTPUT);
  digitalWrite(LED1_PIN, LOW);
  digitalWrite(LED2_PIN, LOW);
  timer.setInterval(3000L, sendSensor);     // every 3 sec
  timer.setInterval(1000L, notifyMotion);   // every 1 sec
  Serial.println("🚀 System Ready: All sensors active.");
}

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

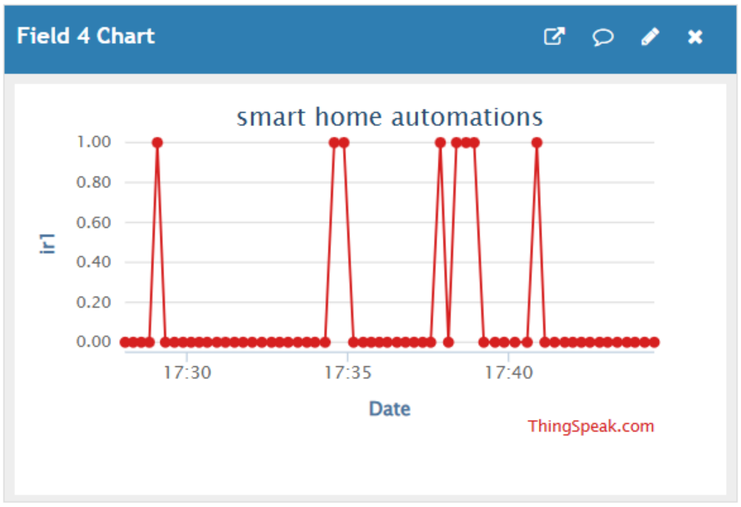
7.Images of Working Model

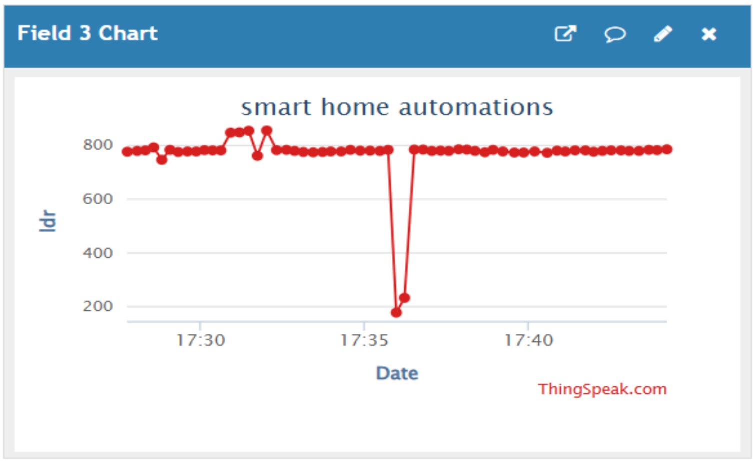
Blynk Interface Screenshots

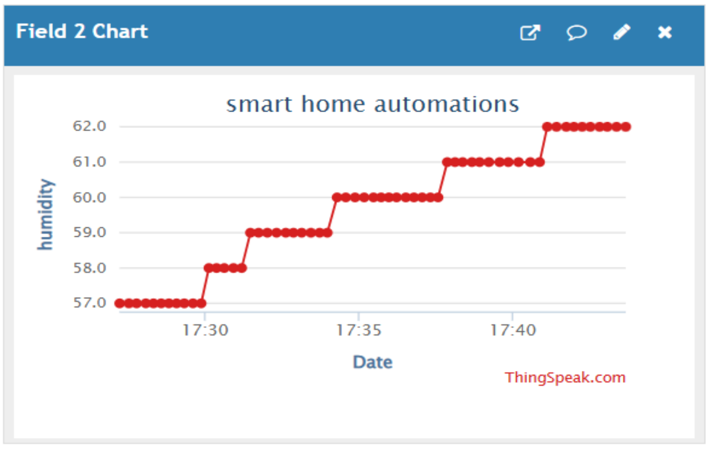


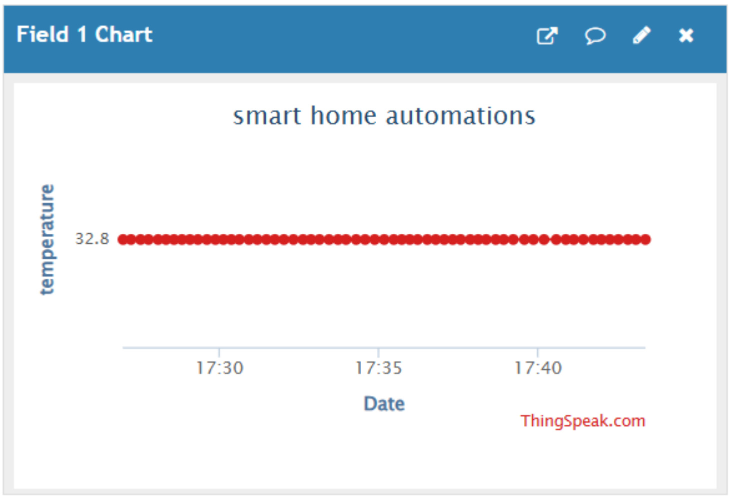
ThingSpeak Graphs


Field 5 Chart
smart home automations
1.00
0.80
0.60
0.40
0.20
0.00
ir2
17:30
17:35
17:40
Date
ThingSpeak.com


Field 4 Chart
smart home automations
1.00
0.80
0.60
0.40
0.20
0.00
ir1
17:30
17:35
17:40
Date
ThingSpeak.com


Field 3 Chart
smart home automations
800
600
400
200
ldr
17:30
17:35
17:40
Date
ThingSpeak.com





Thingspeak links:

https://thingspeak.mathworks.com/channels/2916666/private_show

Read api-

api:https://api.thingspeak.com/channels/2916666/feeds.json?api_key=XRZ1WN 6MGLB7DEO1

8. Conclusion

The smart home automation system was successfully implemented using NodeMCU and Blynk. The system provides real-time control and monitoring of appliances and sensors. With motion detection and temperature-based automation, it enhances home security and energy efficiency. The integration with Blynk makes the solution cost-effective and easy to use for smart home applications.

Smart automation systems offer a significant improvement in the way we interact with our homes, offices, and even light industrial setups. By integrating technologies like **NodeMCU**, **sensors**, and **Blynk**, appliances such as lights, fans, air conditioners, and even industrial equipment can be controlled remotely with ease. The use of **virtual pins** allows for seamless communication between the hardware and mobile apps, providing real-time control and monitoring.

This system not only enhances **convenience** but also optimizes **energy efficiency** by automating operations based on environmental data, such as temperature, humidity, or motion. The ability to set up **automated rules** and **receive notifications** further strengthens the system's functionality, making it perfect for both **personal use** and **small-scale industrial applications**.

Ultimately, the flexibility, scalability, and ease of use of smart automation systems make them an essential part of creating smarter, more efficient environments, offering benefits ranging from energy savings to enhanced security and operational control.