NAME: Aaditya Bhatt

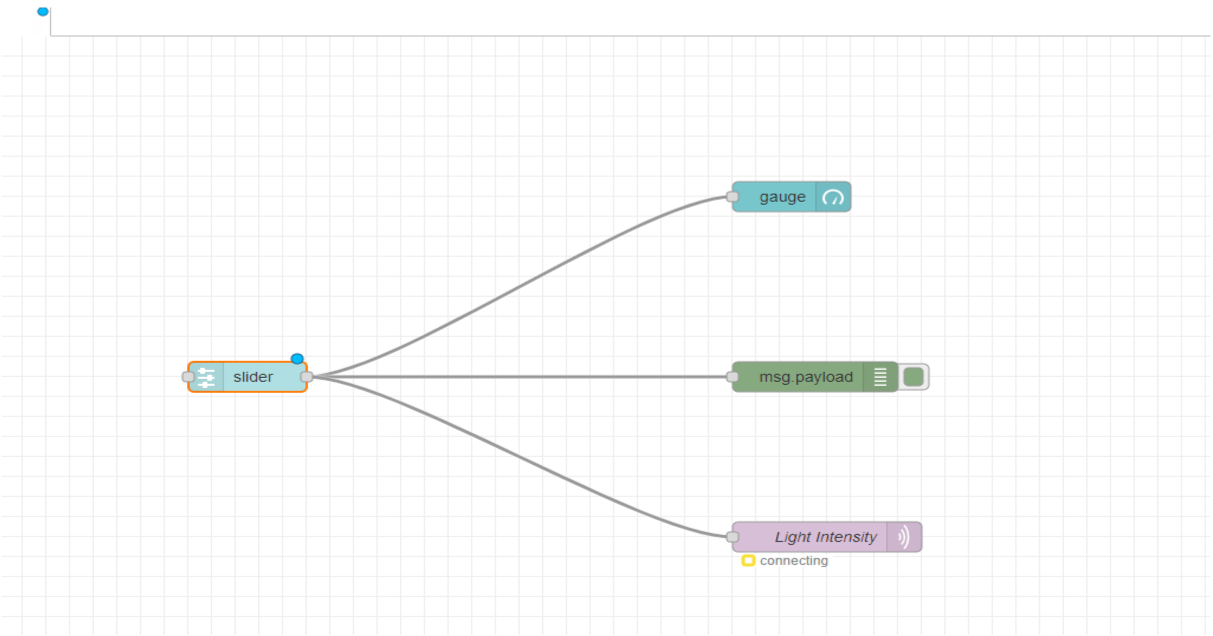
REG NO: 21BEC1531

EXP-1: Display Light Intensity using Node-Red and MQTT

AIM:

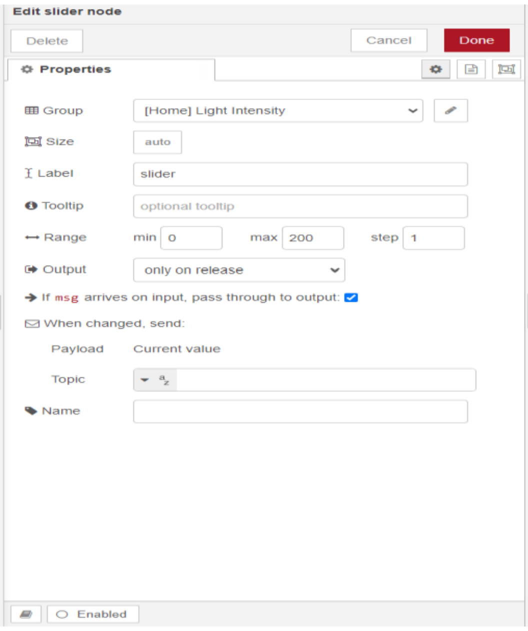
A real-time light intensity monitoring system was built using Node-RED and MQTT, showcasing IoT integration for environmental monitoring. This system measures light levels and displays them in real-time, illustrating the seamless connection between IoT devices and data visualization platforms.

FLOWCHART:



Use MQTT out Node in Node Red and change Label to “Light Intensity”

PROPERTIES:



Group: [Home] Light Intensity

Size: auto

Label: slider

Tooltip: optional tooltip

Range: min 0 max 200 step 1

Output: only on release

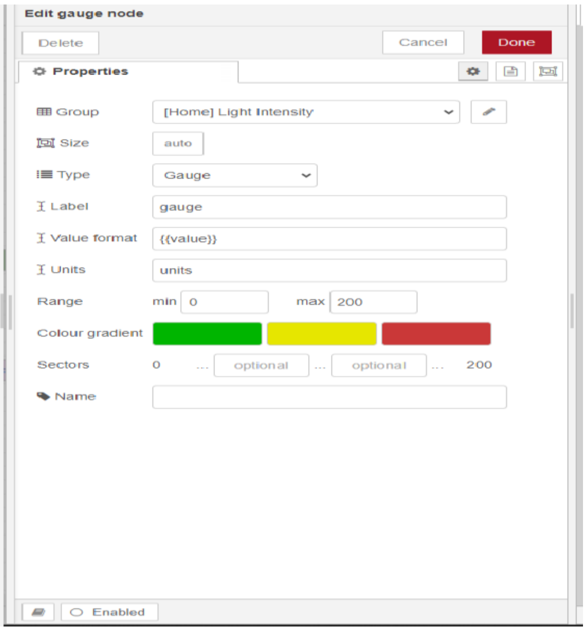
If msg arrives on input, pass through to output: ☑

When changed, send:

Payload: Current value

Topic:

Name:



Group: [Home] Light Intensity

Size: auto

Type: Gauge

Label: gauge

Value format: {{value}}

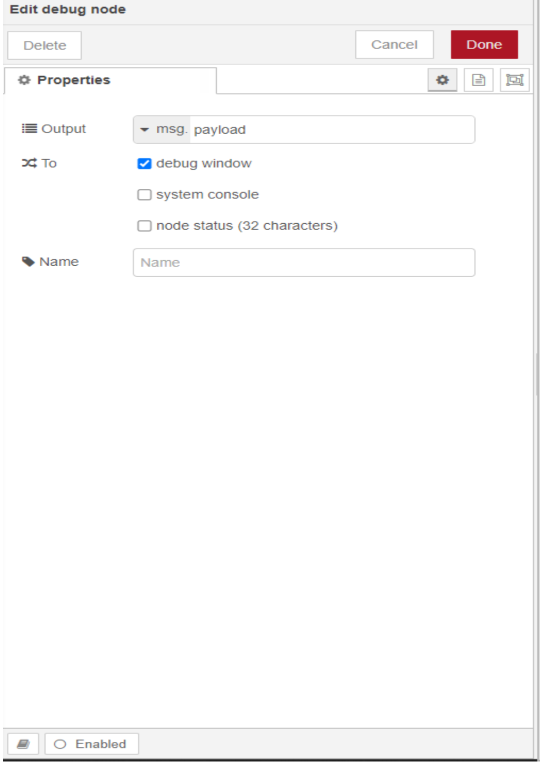
Units: units

Range: min 0 max 200

Colour gradient: 🟩 🟨 🟥

Sectors: 0 ... optional ... optional ... 200

Name:



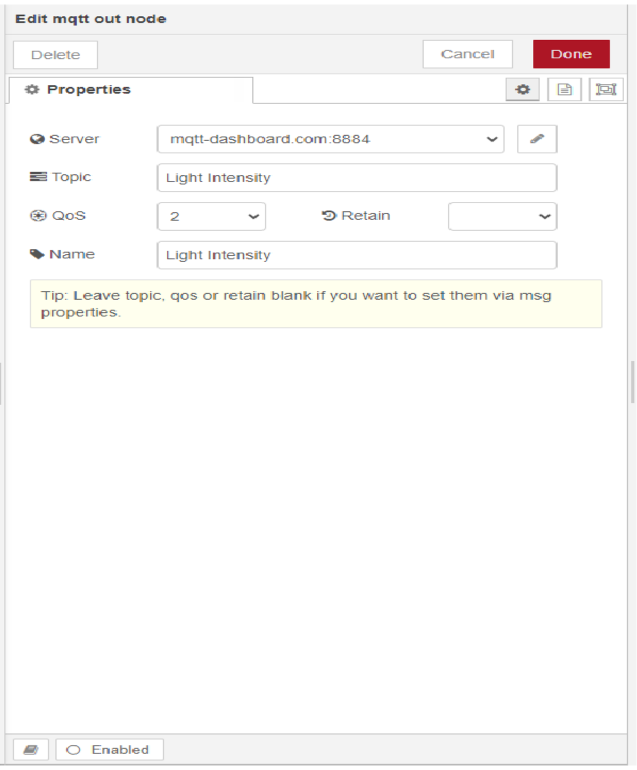
Output msg. payload

To ☑ debug window

☐ system console

☐ node status (32 characters)

Name Name



Server mqtt-dashboard.com:8884

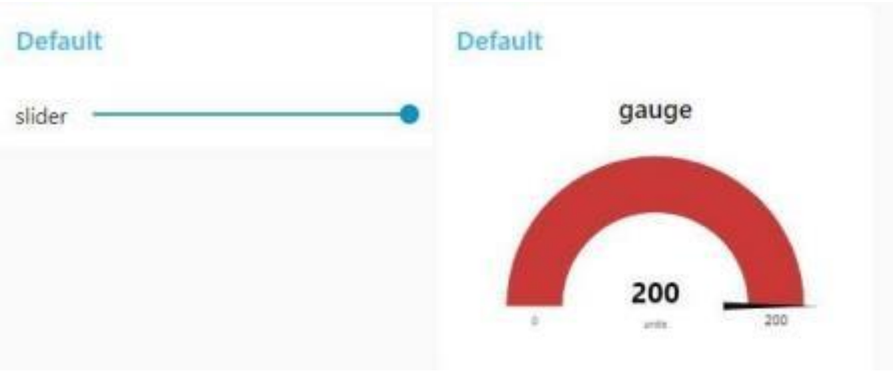
Topic Light Intensity

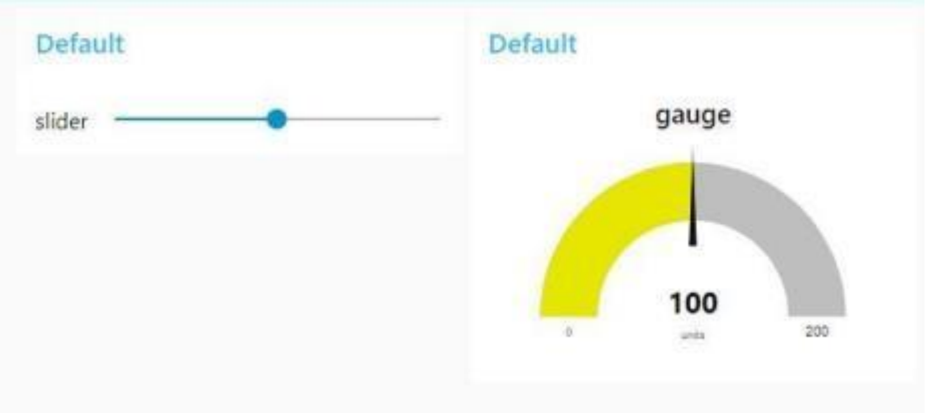
QoS 2 Retain

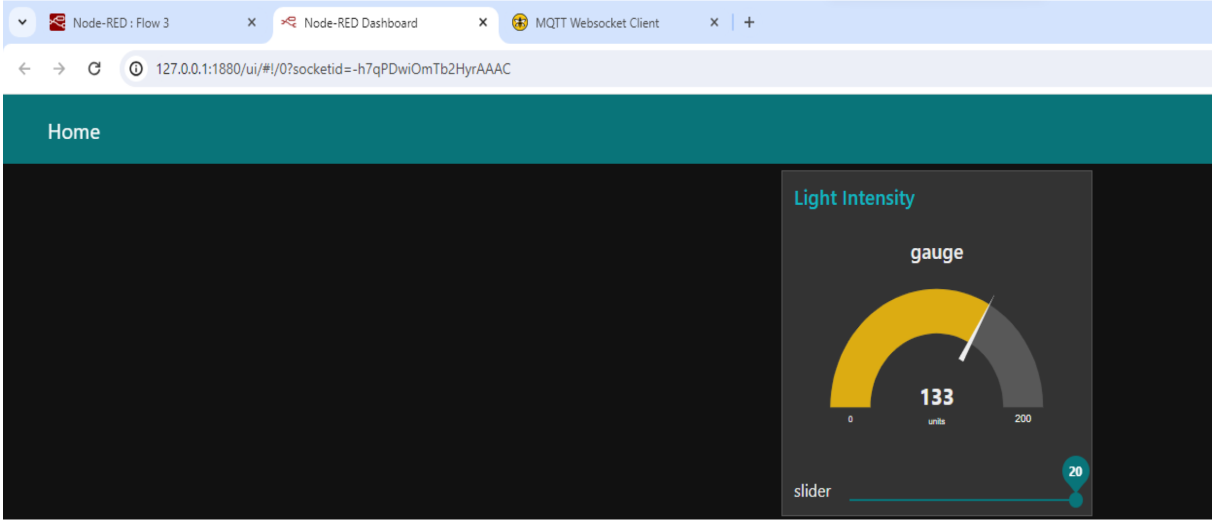
Name Light Intensity

Tip: Leave topic, qos or retain blank if you want to set them via msg properties.

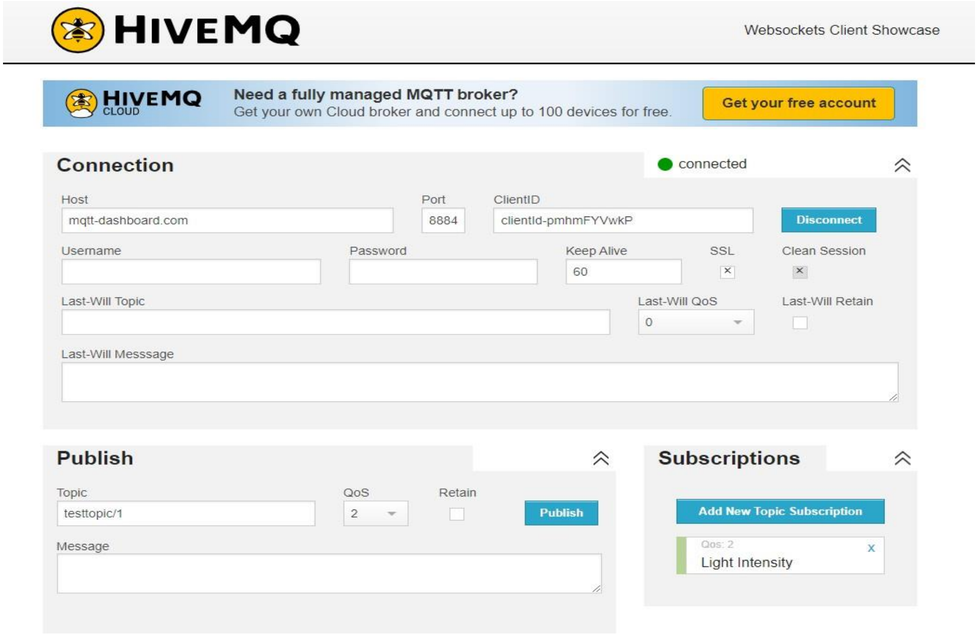
OUTPUT:







127.0.0.1:1880/ui/#!/0?socketid=-h7qPDwiOmTb2HyrAAAC


HIVEMQ
Websockets Client Showcase
HIVEMQ CLOUD
Need a fully managed MQTT broker?
Get your own Cloud broker and connect up to 100 devices for free.
Get your free account

Connection ● connected

Host: mqtt-dashboard.com
Port: 8884
ClientID: clientId-pmhmFYVwkP
Username:
Password:
Keep Alive: 60
SSL: ☒
Clean Session: ☒
Last-Will Topic:
Last-Will QoS: 0
Last-Will Retain: □
Last-Will Messsage:

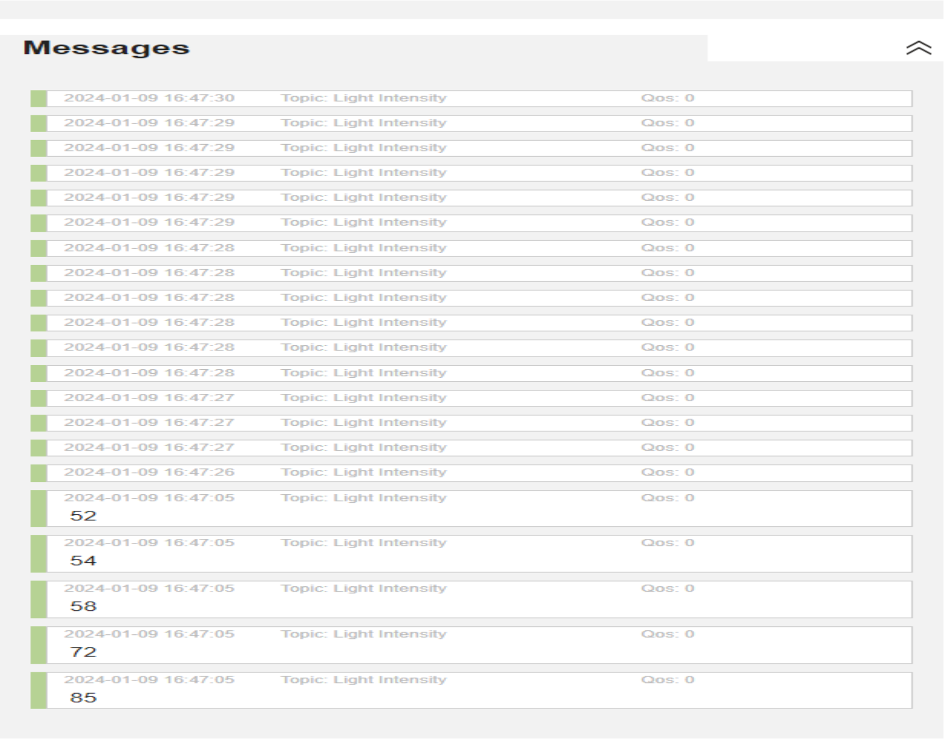
Publish

Topic: testtopic/1
QoS: 2
Retain: □
Message:

Subscriptions

Add New Topic Subscription

Qos: 2
Light Intensity



Messages

```
2024-01-09 16:47:30    Topic: Light Intensity    Qos: 0
2024-01-09 16:47:29    Topic: Light Intensity    Qos: 0
2024-01-09 16:47:29    Topic: Light Intensity    Qos: 0
2024-01-09 16:47:29    Topic: Light Intensity    Qos: 0
2024-01-09 16:47:29    Topic: Light Intensity    Qos: 0
2024-01-09 16:47:29    Topic: Light Intensity    Qos: 0
2024-01-09 16:47:28    Topic: Light Intensity    Qos: 0
2024-01-09 16:47:28    Topic: Light Intensity    Qos: 0
2024-01-09 16:47:28    Topic: Light Intensity    Qos: 0
2024-01-09 16:47:28    Topic: Light Intensity    Qos: 0
2024-01-09 16:47:28    Topic: Light Intensity    Qos: 0
2024-01-09 16:47:28    Topic: Light Intensity    Qos: 0
2024-01-09 16:47:27    Topic: Light Intensity    Qos: 0
2024-01-09 16:47:27    Topic: Light Intensity    Qos: 0
2024-01-09 16:47:27    Topic: Light Intensity    Qos: 0
2024-01-09 16:47:26    Topic: Light Intensity    Qos: 0
2024-01-09 16:47:05    Topic: Light Intensity    Qos: 0
 52
2024-01-09 16:47:05    Topic: Light Intensity    Qos: 0
 54
2024-01-09 16:47:05    Topic: Light Intensity    Qos: 0
 58
2024-01-09 16:47:05    Topic: Light Intensity    Qos: 0
 72
2024-01-09 16:47:05    Topic: Light Intensity    Qos: 0
 85
```

RESULT:

Light intensity was measured, displayed, and published using MQTT and a dashboard, demonstrating the effective integration of IoT technologies for environmental monitoring and data visualization.

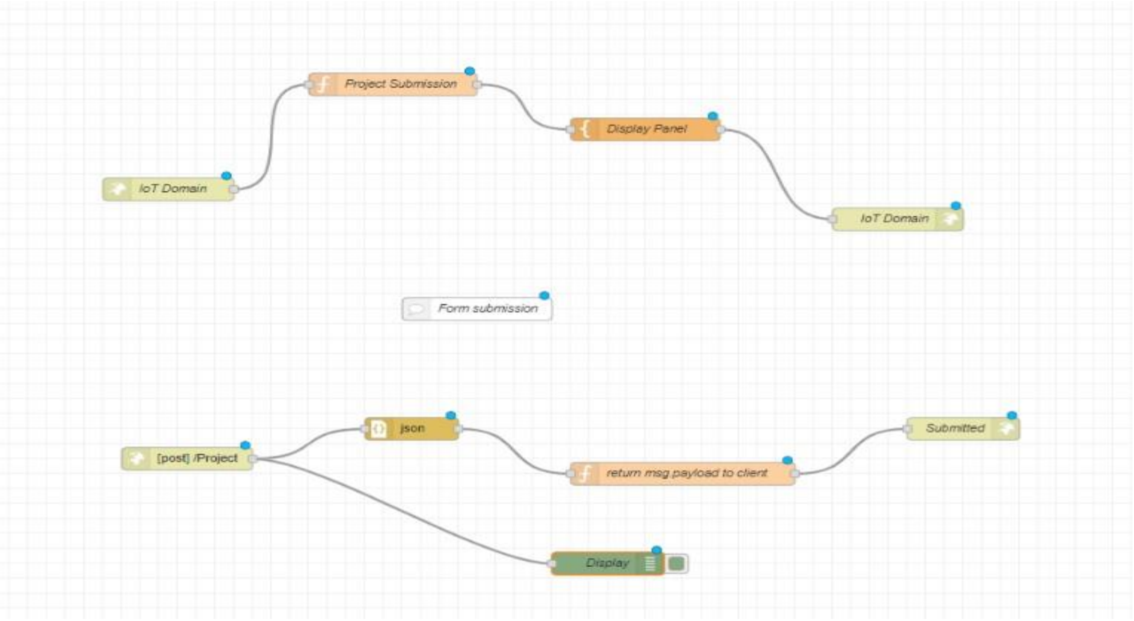
NAME: Aaditya Bhatt
REG NO: 21BEC1531

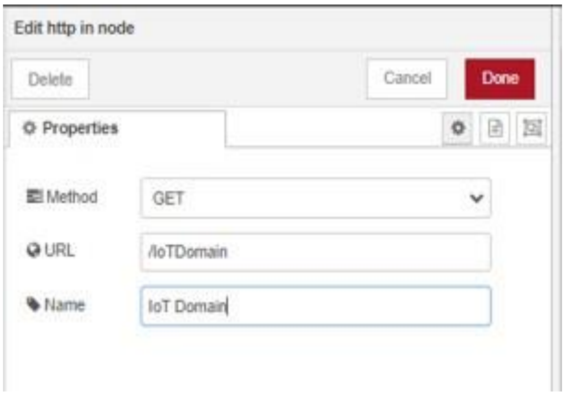
EXP-2:
Web Based Application (HTML) Form Creation & Submission in Node-Red

AIM:

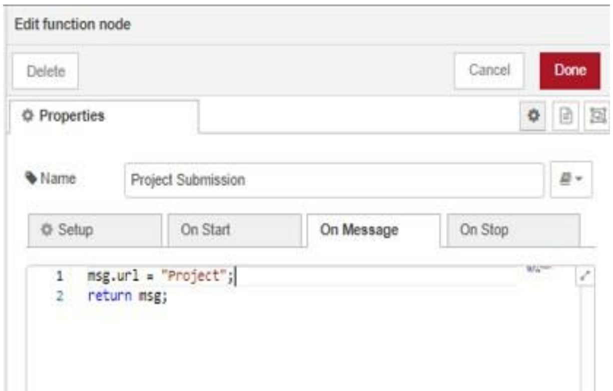
A web-based application was developed in Node-RED, leveraging HTML for form creation and submission. This enables users to interact with the application through forms, showcasing the integration of web technologies for streamlined data input and processing.

Flowchart:



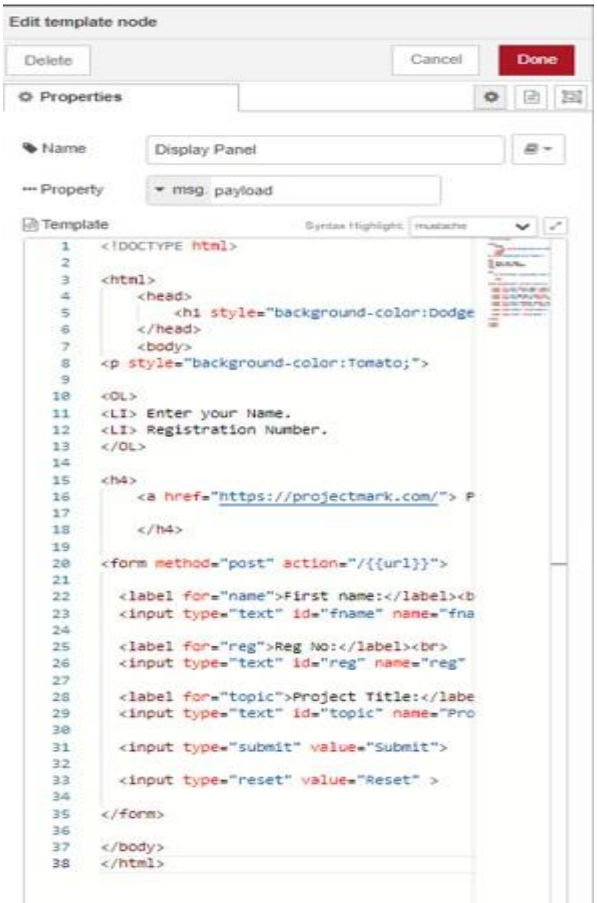


Method: GET
URL: /IoTDomain
Name: IoT Domain



Name: Project Submission

```
msg.url = "Project";
return msg;
```


Edit template node

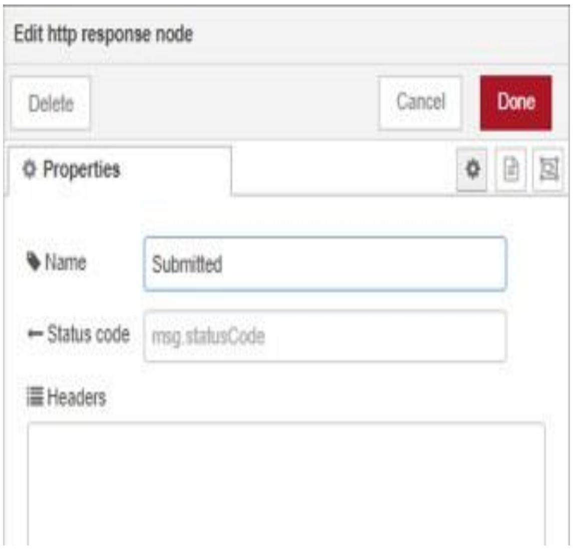
Name: Display Panel

Property: msg. payload

Template

Syntax Highlight: mustache

```
<!DOCTYPE html>

<html>
      <head>
            <h1 style="background-color:Dodge
      </head>
      <body>
<p style="background-color:Tomato;">

<OL>
<LI> Enter your Name.
<LI> Registration Number.
</OL>

<h4>
      <a href="https://projectmark.com/"> P

      </h4>

<form method="post" action="/{{url}}">

   <label for="name">First name:</label><b
   <input type="text" id="fname" name="fna

   <label for="reg">Reg No:</label><br>
   <input type="text" id="reg" name="reg"

   <label for="topic">Project Title:</labe
   <input type="text" id="topic" name="Pro

   <input type="submit" value="Submit">

   <input type="reset" value="Reset" >

</form>

</body>
</html>
```


Edit http response node

Name: Submitted

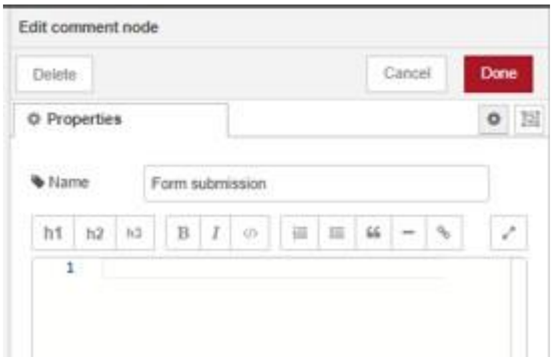
Status code: msg.statusCode

Headers

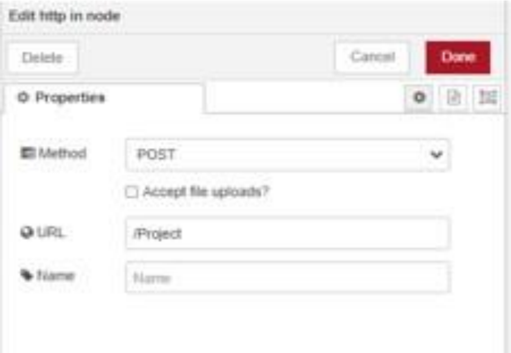
CODE:

```
<!DOCTYPE html>

<html>
      <head>
            <h1 style="background-color:DodgerBlue;">IoT Domain</h1>
      </head>
      <body>
<p style="background-color:Tomato;">

<OL>
<LI> Enter your Name.
<LI> Registration Number.
</OL>
<h4>
      <a href="https://projectmark.com/"> Project Mark</a>

      </h4>
<form method="post" action="/{{url}}">
   <label for="name">First name:</label><br>
   <input type="text" id="fname" name="fname"><br>

   <label for="reg">Reg No:</label><br>
   <input type="text" id="reg" name="reg" ><br><br>

   <label for="topic">Project Title:</label><br>
   <input type="text" id="topic" name="Project Topic" ><br><br>
```

```
    <input type="submit" value="Submit">

    <input type="reset" value="Reset" >
  </form>

  </body>
  </html>
```



Name: Form submission

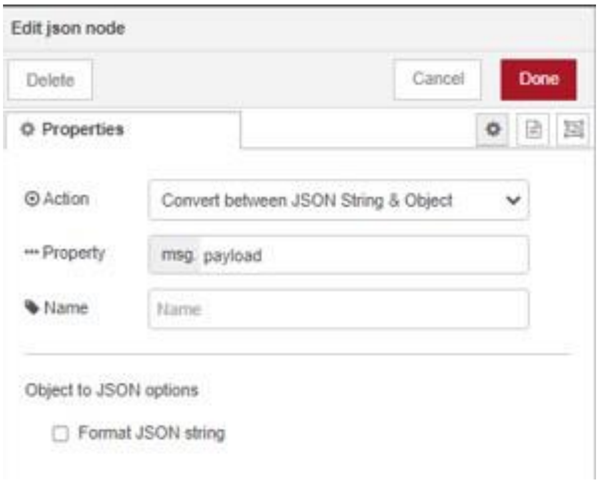


Method: POST

□ Accept file uploads?

URL: /Project

Name: Name



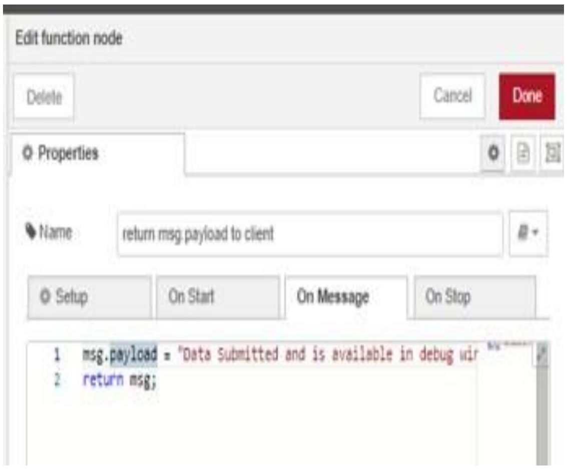
Action: Convert between JSON String & Object

Property: msg. payload

Name: Name

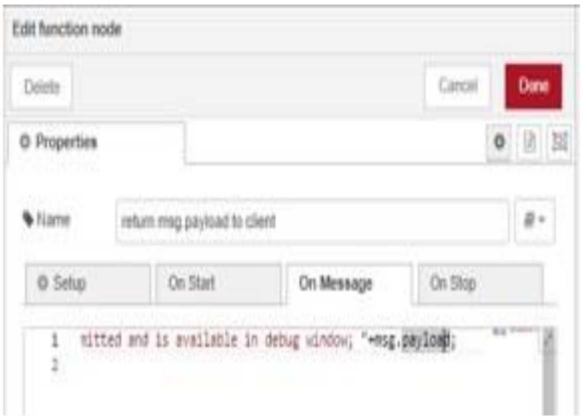
Object to JSON options

□ Format JSON string



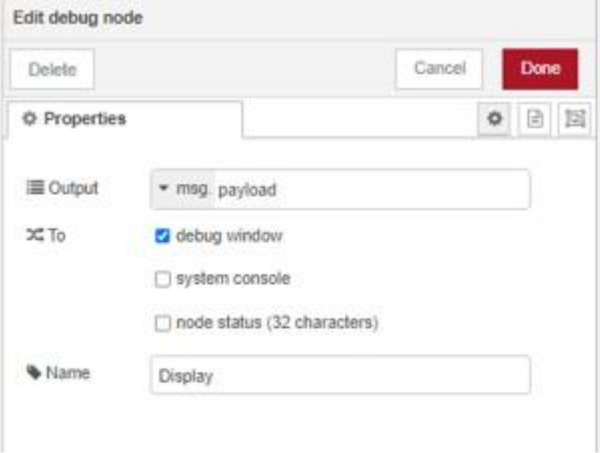
Name: return msg payload to client

```
msg.payload = "Data Submitted and is available in debug wir
return msg;
```



Name: return msg payload to client

```
nitted and is available in debug window; "+msg.payload;
```



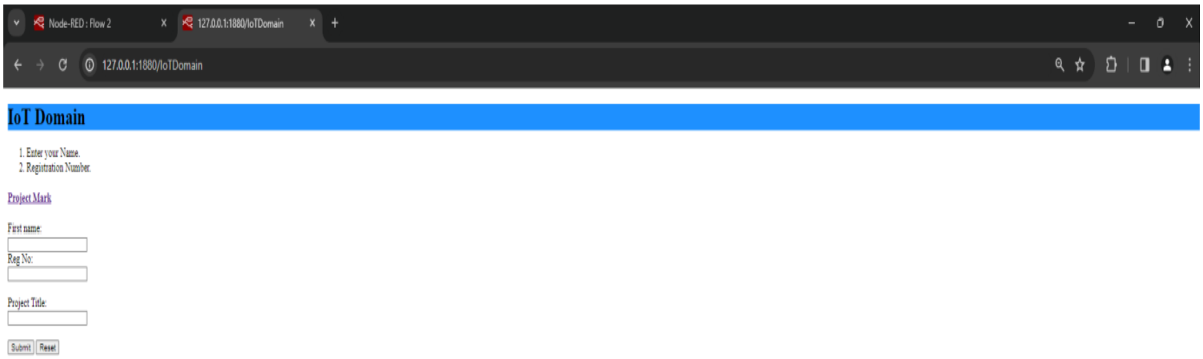
Output: msg. payload

To: ☑ debug window

□ system console

□ node status (32 characters)

Name: Display



IoT Domain

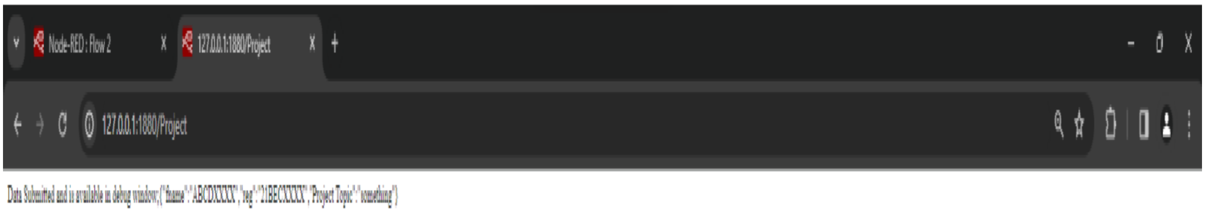
1. Enter your Name.
2. Registration Number.

Project Mark

First name:

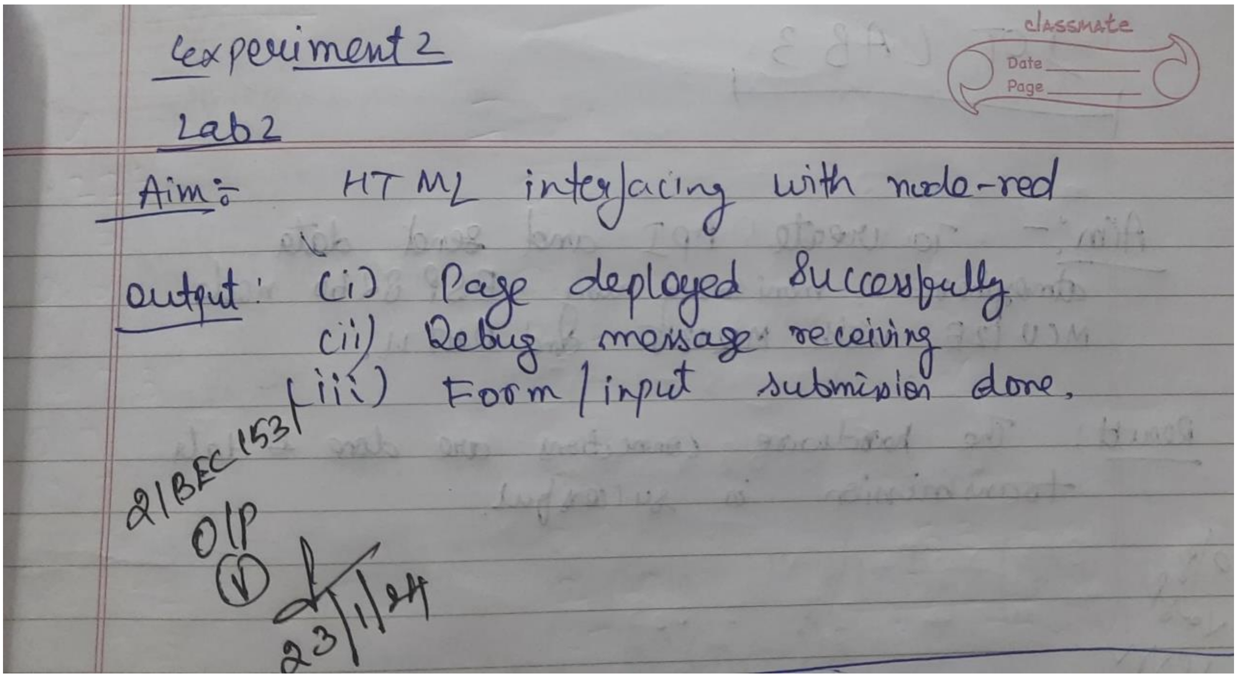
Reg No:

Project Title:



Data Submitted and is available in debug window;{"fname":"ABCDXXXX","reg":"21BECXXXXX","Project Topic":"something"}

Verification sign:


classmate
Date
Page

experiment 2

Lab 2

Aim:- HTML interfacing with node-red

output: (i) Page deployed Successfully
(ii) Debug message receiving
(iii) Form / input submission done.

21BEC1531

O/P

(V)

23/1/24

RESULT:

We conducted Web-Based Application Form Creation and Submission for Project Title Submission and implemented a Smart Parking System in Node-Red, successfully obtaining desired outputs for both tasks.

NAME: AADITYA BHATT

REG NO: 21BEC1531

EXP-3: Atmospheric monitoring using ESP8266 – NODEMCU 12Ewith MQ135 and DHT 11and Thingspeak Cloud Computing

AIM:

To find out the atmospheric parameters using ESP8266 and NODEMCU with MQ135 and DHT11 and visualizing it in Thingspeak.

CODES:

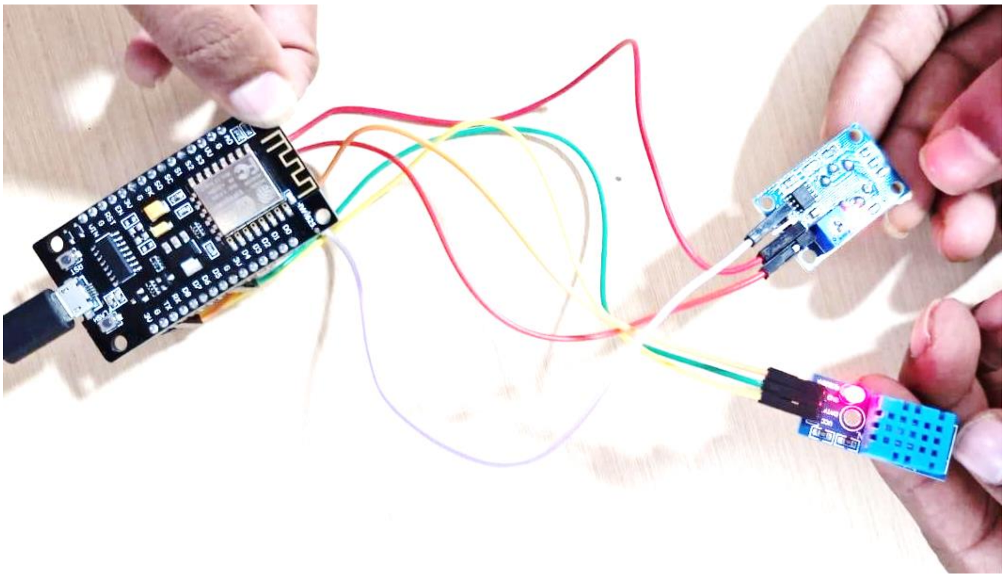
ARDIUNO CODE:

```
#include <ESP8266WiFi.h>
#include  <DHT.h>
#include <ThingSpeak.h>
#include "MQ2.h"
int Analog_Input = A0;
int lpg, co, smoke, smoke1;
 float h, t;
MQ2 mq2(Analog_Input);
const char *ssid = "Galaxy A22 5G7259";
const char *pass =  "tflp4120";
DHT dht(D5, DHT11);
WiFiClient client;
long myChannelNumber = 1498653;
const char myWriteAPIKey[] = "WIMEYLX14IIGH3I3";
// The load resistance on the board
//#define RLOAD 22.0
// Calibration resistance at atmospheric CO2 level
//#define RZERO 879.13
int val;
void setup() {
   mq2.begin();
  // put your setup code here, to run once:
  Serial.begin(115200);
  WiFi.begin(ssid, pass);
  while(WiFi.status() != WL_CONNECTED)
  {
    delay(200);
    Serial.print("..");
  }
  Serial.println();
  Serial.println("NodeMCU is connected!");
  Serial.println(WiFi.localIP());
  dht.begin();
  ThingSpeak.begin(client);
}
void loop() {
  // put your main code here, to run repeatedly:
  float h = dht.readHumidity();
  float t =
  dht.readTemperature();
  float* values= mq2.read(true); //set it false if you don't want to print the values in the Serial
  //lpg = values[0];
  lpg = mq2.readLPG();
  //co = values[1];
  co = mq2.readCO();
  //smoke =  values[2];
  smoke =
  mq2.readSmoke();
  smoke1=(smoke*100)/100000
 0;Serial.println("LPG:");
 Serial.println(lpg);
 Serial.println(" CO:");
 Serial.println(co);
 Serial.println("SMOKE:");
 Serial.println(smoke1);
 Serial.println(" %");
  delay(1000);
  ThingSpeak.writeField(myChannelNumber, 1, lpg, myWriteAPIKey);
  ThingSpeak.writeField(myChannelNumber, 2, co, myWriteAPIKey);
  ThingSpeak.writeField(myChannelNumber, 3, smoke1, myWriteAPIKey);
  Serial.println("Temperature: " + (String) t);
  Serial.println("Humidity: " + (String) h);
  ThingSpeak.writeField(myChannelNumber, 4, t,
  myWriteAPIKey);ThingSpeak.writeField(myChannelNumber,
  5, h, myWriteAPIKey);delay(100);
}
```

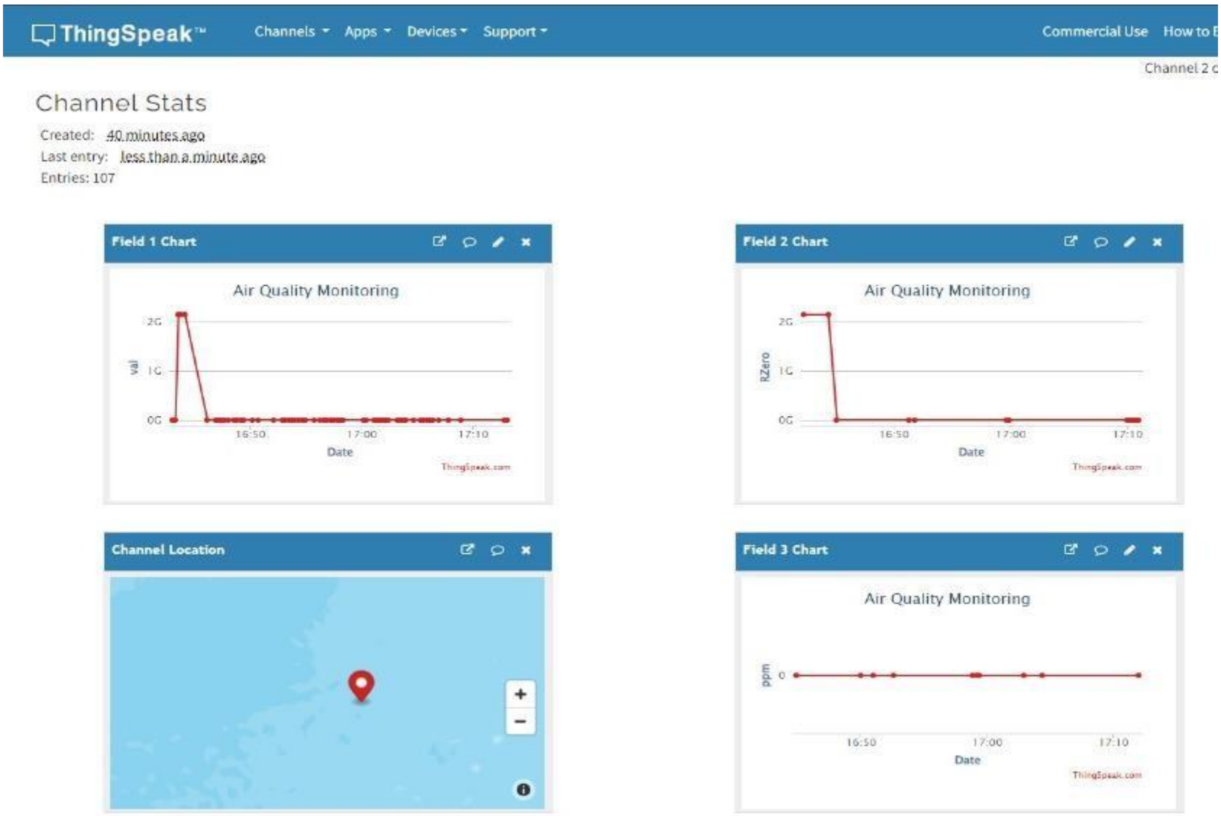
MQ135 CONFIG:

```
#include <ESP8266WiFi.h>
//#define RLOAD 10.0
// Calibration resistance at atmospheric CO2 level
//#define RZERO 59
#include "MQ135.h"
MQ135 gasSensor = MQ135(A0);
int val;
int sensorPin = A0;
int sensorValue = 0;
void setup() {
Serial.begin(9600);
  pinMode(sensorPin, INPUT);
}
void loop() {
  val = analogRead(A0);
  Serial.print ("raw = ");
  Serial.println (val);
  float zero = gasSensor.getRZero();
  Serial.print ("rzero: ");
  Serial.println (zero);
  float ppm = gasSensor.getPPM();
  Serial.print ("ppm: ");
  Serial.println  (ppm);
  delay(5000);
}
```

CONNECTIONS:

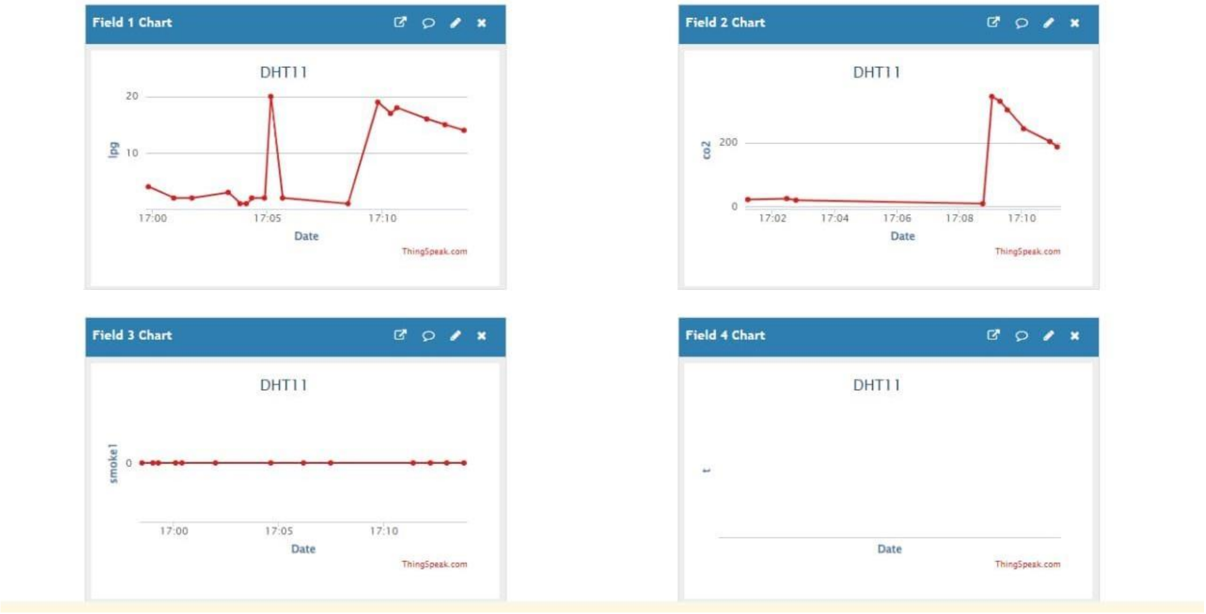


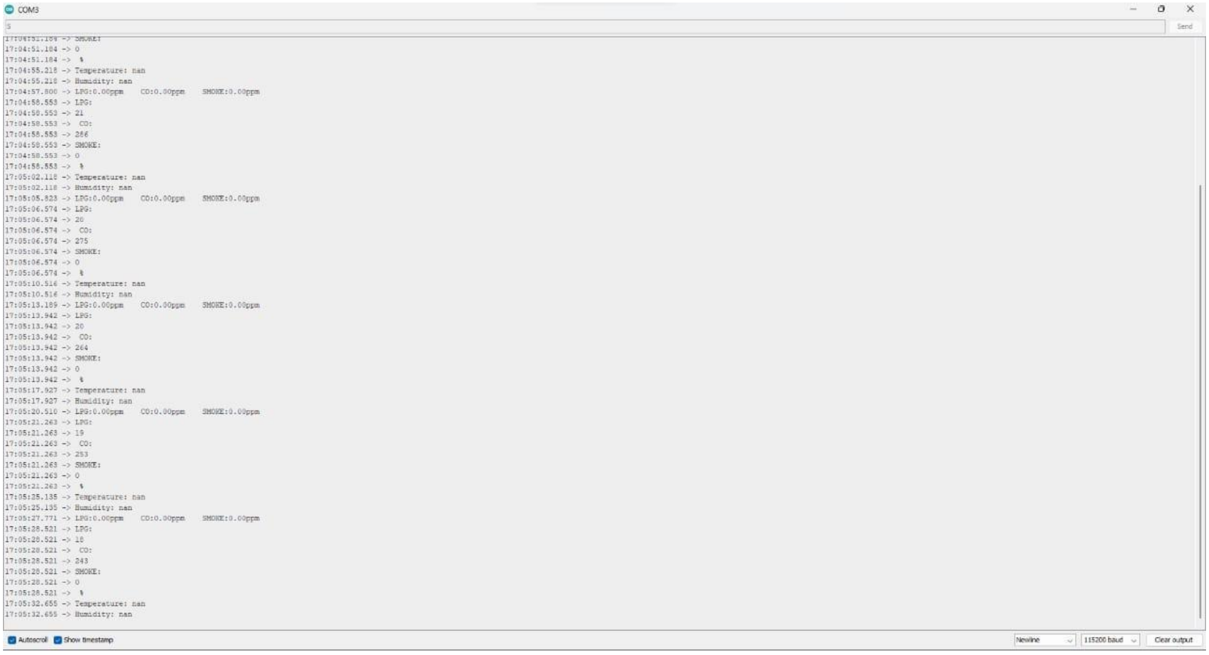
OUTPUT:



Channel Stats

Created: 40 minutes ago
Last entry: less than a minute ago
Entries: 107

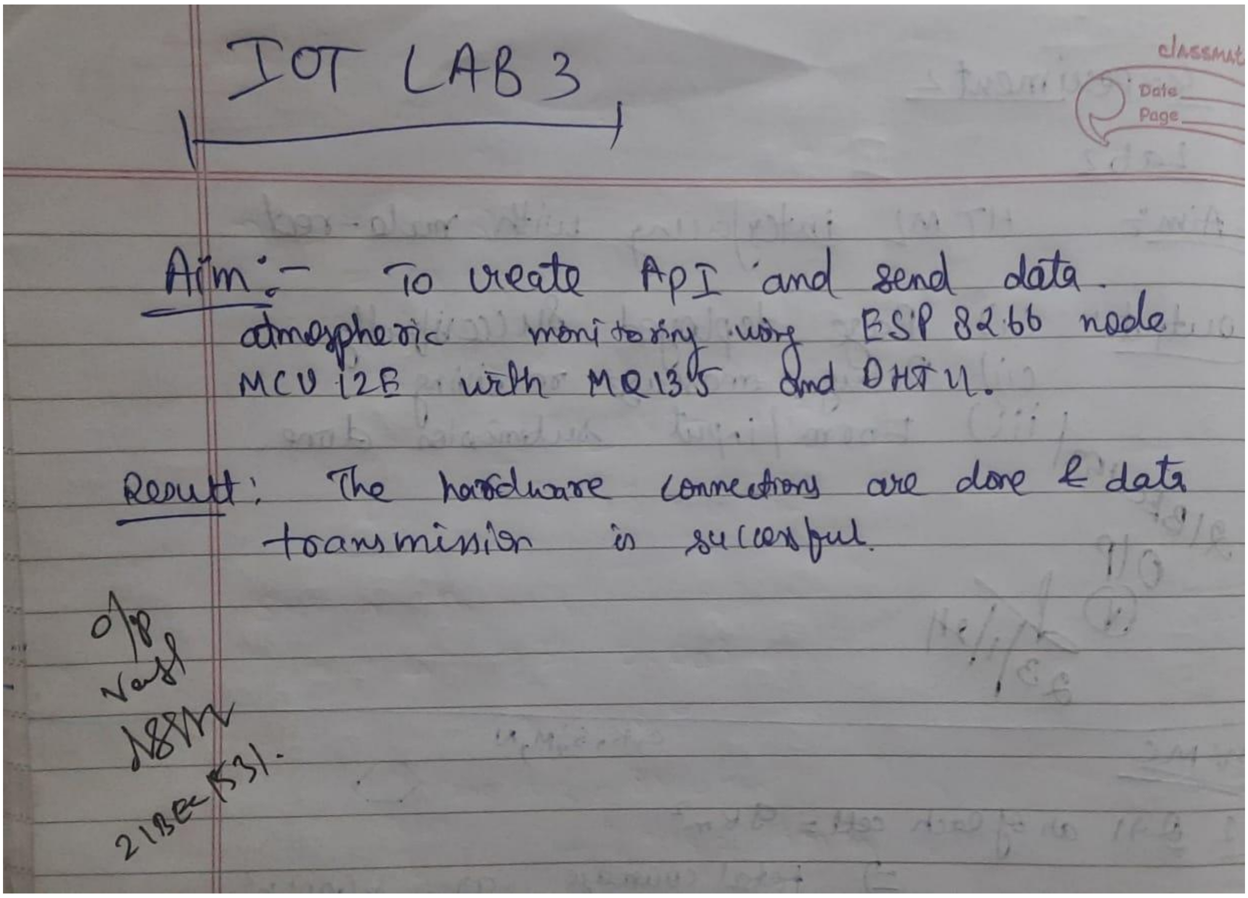




```
17:04:51.184 -> SMOKE:
17:04:51.184 -> 0
17:04:51.184 ->  %
17:04:55.218 -> Temperature: nan
17:04:55.218 -> Humidity: nan
17:04:57.800 -> LPG:0.00ppm    CO:0.00ppm    SMOKE:0.00ppm
17:04:58.553 -> LPG:
17:04:58.553 -> 21
17:04:58.553 ->  CO:
17:04:58.553 -> 266
17:04:58.553 -> SMOKE:
17:04:58.553 -> 0
17:04:58.553 ->  %
17:05:02.118 -> Temperature: nan
17:05:02.118 -> Humidity: nan
17:05:05.823 -> LPG:0.00ppm    CO:0.00ppm    SMOKE:0.00ppm
17:05:06.574 -> LPG:
17:05:06.574 -> 20
17:05:06.574 ->  CO:
17:05:06.574 -> 275
17:05:06.574 -> SMOKE:
17:05:06.574 -> 0
17:05:06.574 ->  %
17:05:10.516 -> Temperature: nan
17:05:10.516 -> Humidity: nan
17:05:13.189 -> LPG:0.00ppm    CO:0.00ppm    SMOKE:0.00ppm
17:05:13.942 -> LPG:
17:05:13.942 -> 20
17:05:13.942 ->  CO:
17:05:13.942 -> 264
17:05:13.942 -> SMOKE:
17:05:13.942 -> 0
17:05:13.942 ->  %
17:05:17.927 -> Temperature: nan
17:05:17.927 -> Humidity: nan
17:05:20.510 -> LPG:0.00ppm    CO:0.00ppm    SMOKE:0.00ppm
17:05:21.263 -> LPG:
17:05:21.263 -> 19
17:05:21.263 ->  CO:
17:05:21.263 -> 253
17:05:21.263 -> SMOKE:
17:05:21.263 -> 0
17:05:21.263 ->  %
17:05:25.135 -> Temperature: nan
17:05:25.135 -> Humidity: nan
17:05:27.771 -> LPG:0.00ppm    CO:0.00ppm    SMOKE:0.00ppm
17:05:28.521 -> LPG:
17:05:28.521 -> 18
17:05:28.521 ->  CO:
17:05:28.521 -> 243
17:05:28.521 -> SMOKE:
17:05:28.521 -> 0
17:05:28.521 ->  %
17:05:32.655 -> Temperature: nan
17:05:32.655 -> Humidity: nan
```

☑ Autoscroll ☑ Show timestamp

Verification sign:



IOT LAB 3

Aim:- To create ApI and send data atmospheric monitoring using ESP 8266 node MCU 12E with MQ135 and DHT 11.

Result: The hardware connections are done & data transmission is succesful.

o/p Verfd

21BEC1531.

RESULT:

Atmospheric parameters are monitored using ESP8266 and NodeMCU equipped with MQ135 and DHT11 sensors, with data visualization achieved through integration with ThingSpeak platform.

Name: Aaditya Bhatt

Reg.No: 21BEC1531

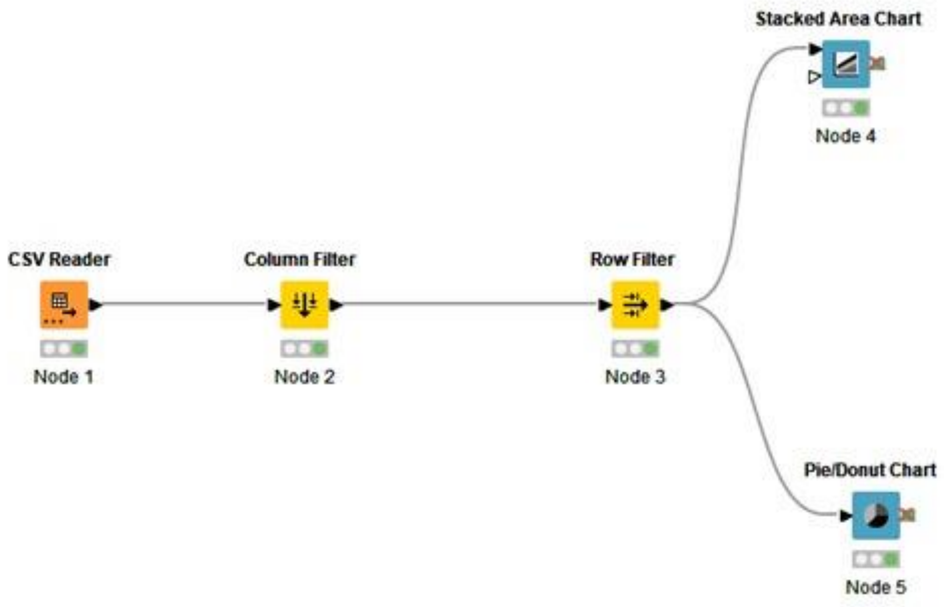
<u>EXP-4:</u> Representation of data using KNIME analytic software.

<u>Aim:</u>

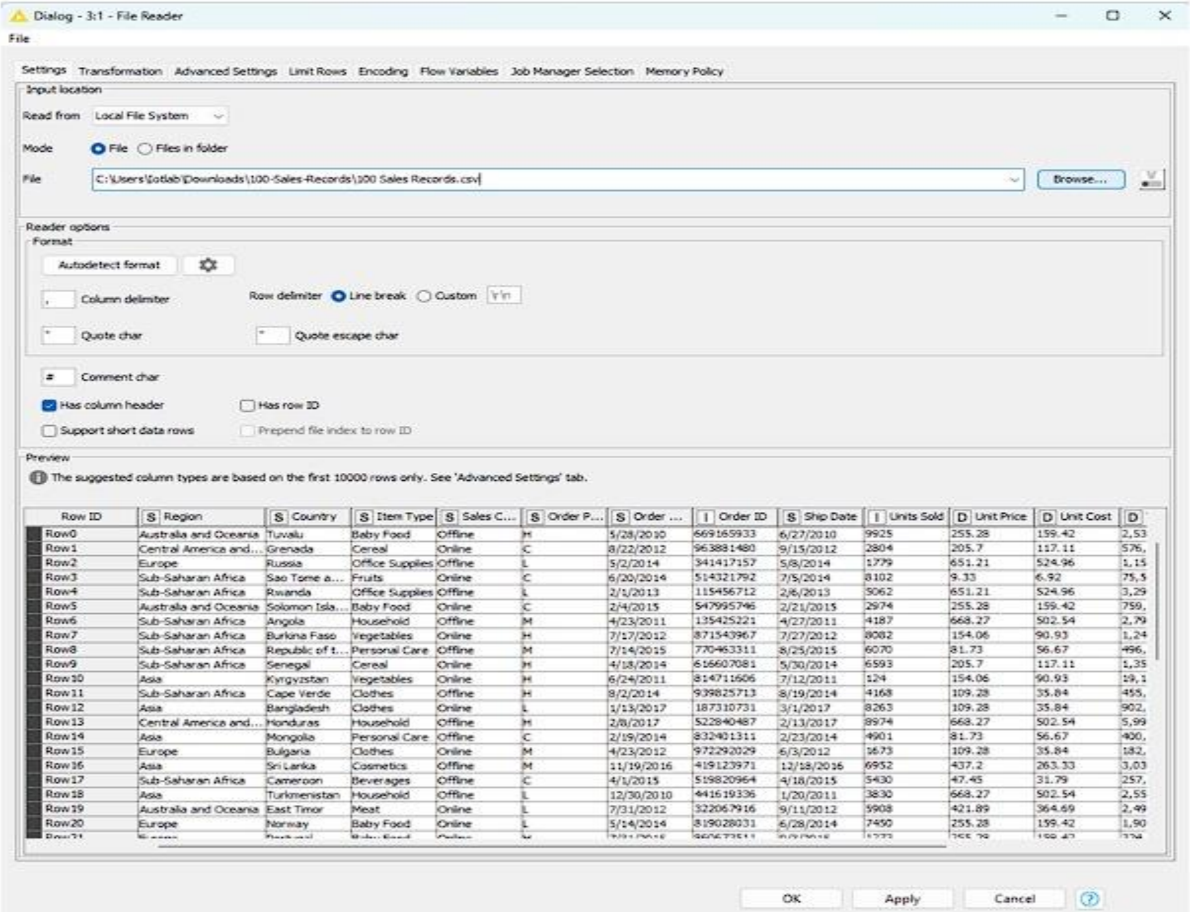
To show our data stored in the form of csv file to visual representation in form of stack area and pie chart using KNIME software.

<u>Software used:</u> KNIME

Circuit diagram:



Dataset:



Dialog - 3:1 - File Reader

File

Settings | Transformation | Advanced Settings | Limit Rows | Encoding | Flow Variables | Job Manager Selection | Memory Policy

Input location

Read from: Local File System

Mode: ● File ○ Files in folder

File: C:\Users\Iotlab\Downloads\100-Sales-Records\100 Sales Records.csv

Reader options

Format

Column delimiter: ,

Row delimiter: ● Line break ○ Custom \r\n

Quote char: "

Quote escape char: "

Comment char: #

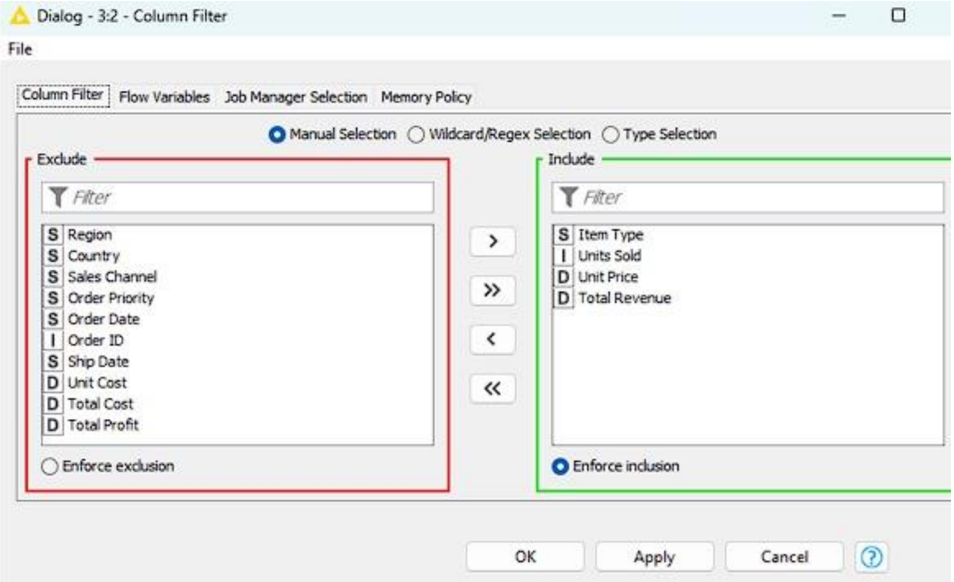
☑ Has column header □ Has row ID

□ Support short data rows □ Prepend file index to row ID

Preview

The suggested column types are based on the first 10000 rows only. See 'Advanced Settings' tab.

Row ID	S Region	S Country	S Item Type	S Sales C...	S Order P...	S Order ...	I Order ID	S Ship Date	I Units Sold	D Unit Price	D Unit Cost	D
Row0	Australia and Oceania	Tuvalu	Baby Food	Offline	H	5/28/2010	669165933	6/27/2010	9925	255.28	159.42	2,53
Row1	Central America and...	Grenada	Cereal	Online	C	8/22/2012	963881480	9/15/2012	2804	205.7	117.11	576,
Row2	Europe	Russia	Office Supplies	Offline	L	5/2/2014	341417157	5/8/2014	1779	651.21	524.96	1,15
Row3	Sub-Saharan Africa	Sao Tome a...	Fruits	Online	C	6/20/2014	514321792	7/5/2014	8102	9.33	6.92	75,5
Row4	Sub-Saharan Africa	Rwanda	Office Supplies	Offline	L	2/1/2013	115456712	2/6/2013	5062	651.21	524.96	3,29
Row5	Australia and Oceania	Solomon Isla...	Baby Food	Online	C	2/4/2015	547995746	2/21/2015	2974	255.28	159.42	759,
Row6	Sub-Saharan Africa	Angola	Household	Offline	M	4/23/2011	135425221	4/27/2011	4187	668.27	502.54	2,79
Row7	Sub-Saharan Africa	Burkina Faso	Vegetables	Online	H	7/17/2012	871543967	7/27/2012	8082	154.06	90.93	1,24
Row8	Sub-Saharan Africa	Republic of t...	Personal Care	Offline	M	7/14/2015	770463311	8/25/2015	6070	81.73	56.67	496,
Row9	Sub-Saharan Africa	Senegal	Cereal	Online	H	4/18/2014	616607081	5/30/2014	6593	205.7	117.11	1,35
Row10	Asia	Kyrgyzstan	Vegetables	Online	H	6/24/2011	814711606	7/12/2011	124	154.06	90.93	19,1
Row11	Sub-Saharan Africa	Cape Verde	Clothes	Offline	H	8/2/2014	939825713	8/19/2014	4168	109.28	35.84	455,
Row12	Asia	Bangladesh	Clothes	Online	L	1/13/2017	187310731	3/1/2017	8263	109.28	35.84	902,
Row13	Central America and...	Honduras	Household	Offline	H	2/8/2017	522840487	2/13/2017	8974	668.27	502.54	5,99
Row14	Asia	Mongolia	Personal Care	Offline	C	2/19/2014	832401311	2/23/2014	4901	81.73	56.67	400,
Row15	Europe	Bulgaria	Clothes	Online	M	4/23/2012	972292029	6/3/2012	1673	109.28	35.84	182,
Row16	Asia	Sri Lanka	Cosmetics	Offline	M	11/19/2016	419123971	12/18/2016	6952	437.2	263.33	3,03
Row17	Sub-Saharan Africa	Cameroon	Beverages	Offline	C	4/1/2015	519820964	4/18/2015	5430	47.45	31.79	257,
Row18	Asia	Turkmenistan	Household	Offline	L	12/30/2010	441619336	1/20/2011	3830	668.27	502.54	2,55
Row19	Australia and Oceania	East Timor	Meat	Online	L	7/31/2012	322067916	9/11/2012	5908	421.89	364.69	2,49
Row20	Europe	Norway	Baby Food	Online	L	5/14/2014	819028031	6/28/2014	7450	255.28	159.42	1,90



Dialog - 3:2 - Column Filter

File

Column Filter | Flow Variables | Job Manager Selection | Memory Policy

● Manual Selection ○ Wildcard/Regex Selection ○ Type Selection

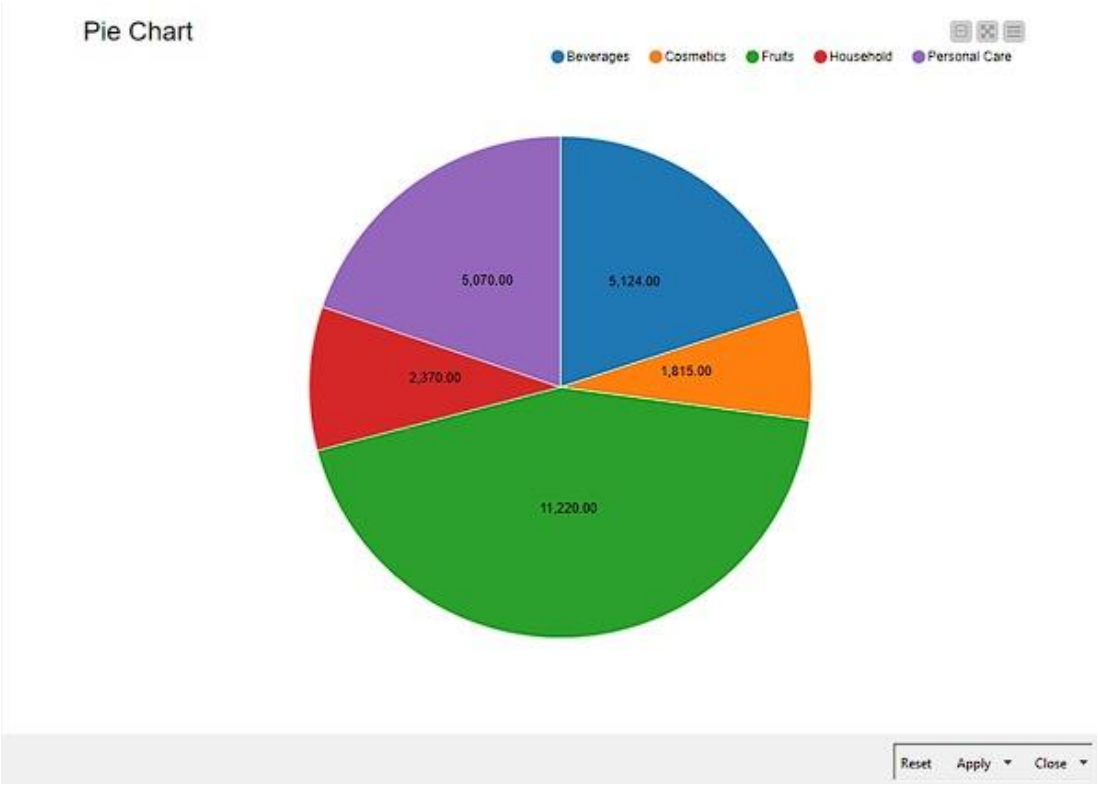
Exclude:
- S Region
- S Country
- S Sales Channel
- S Order Priority
- S Order Date
- I Order ID
- S Ship Date
- D Unit Cost
- D Total Cost
- D Total Profit

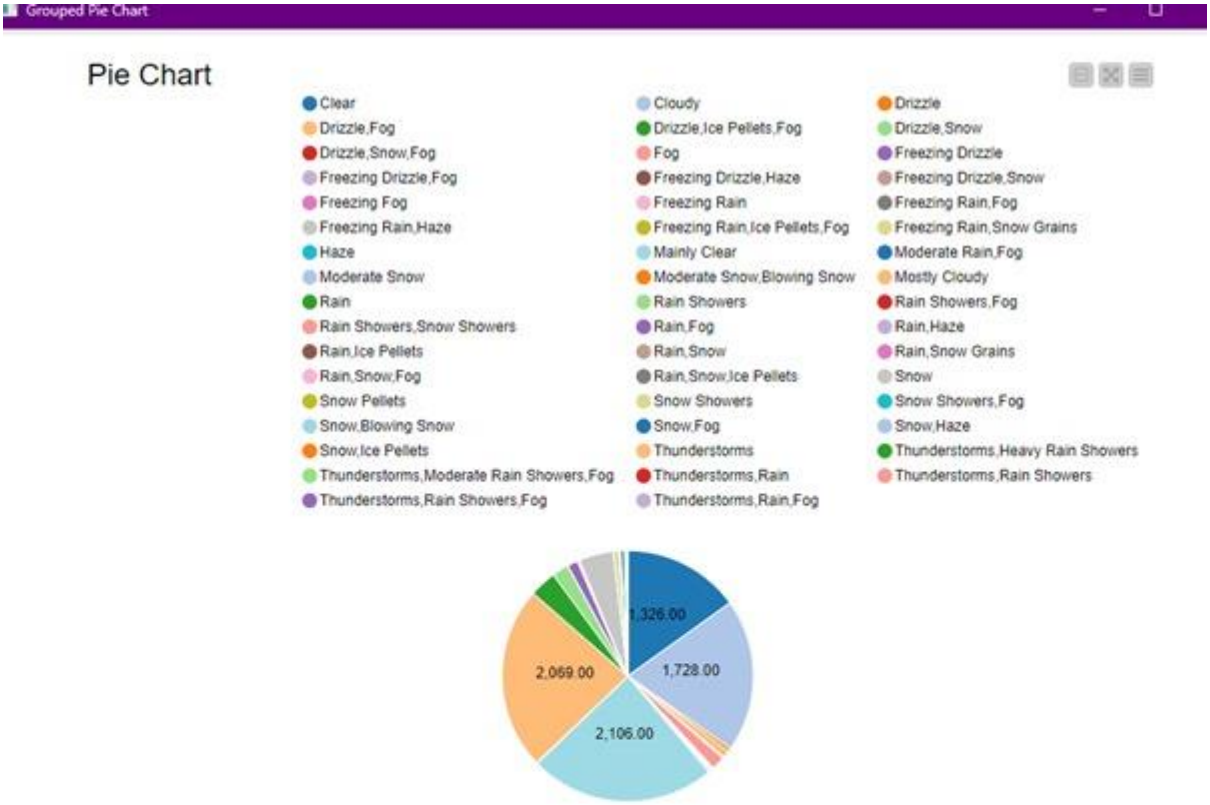
○ Enforce exclusion

Include:
- S Item Type
- I Units Sold
- D Unit Price
- D Total Revenue

● Enforce inclusion

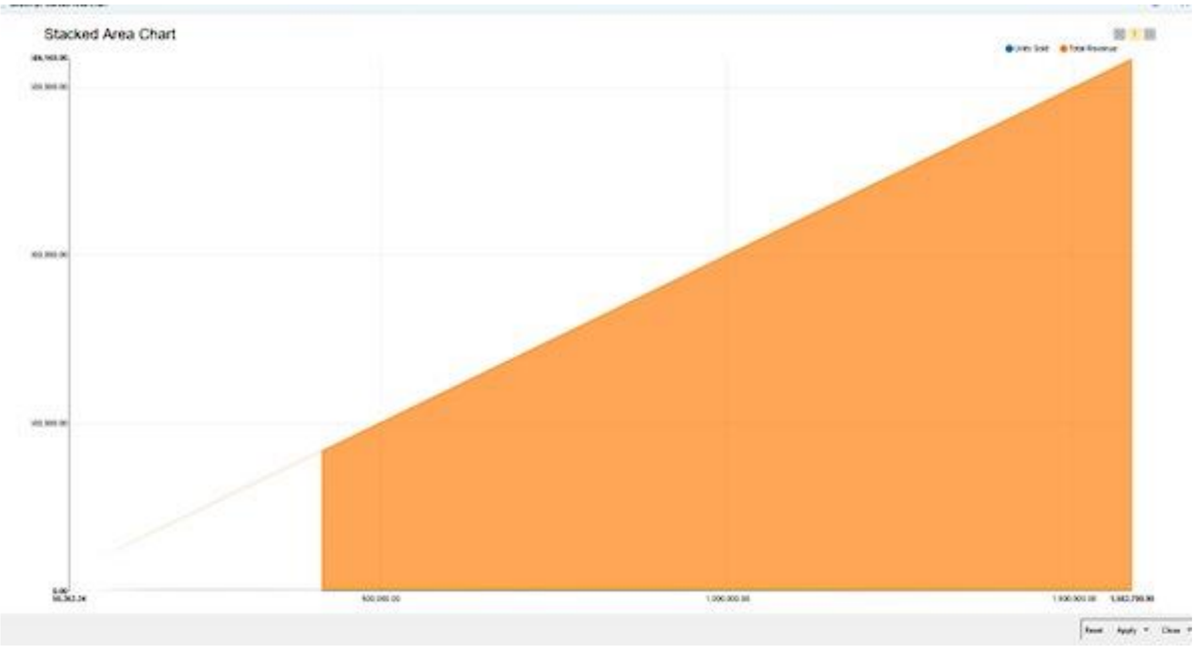
Pie chart:

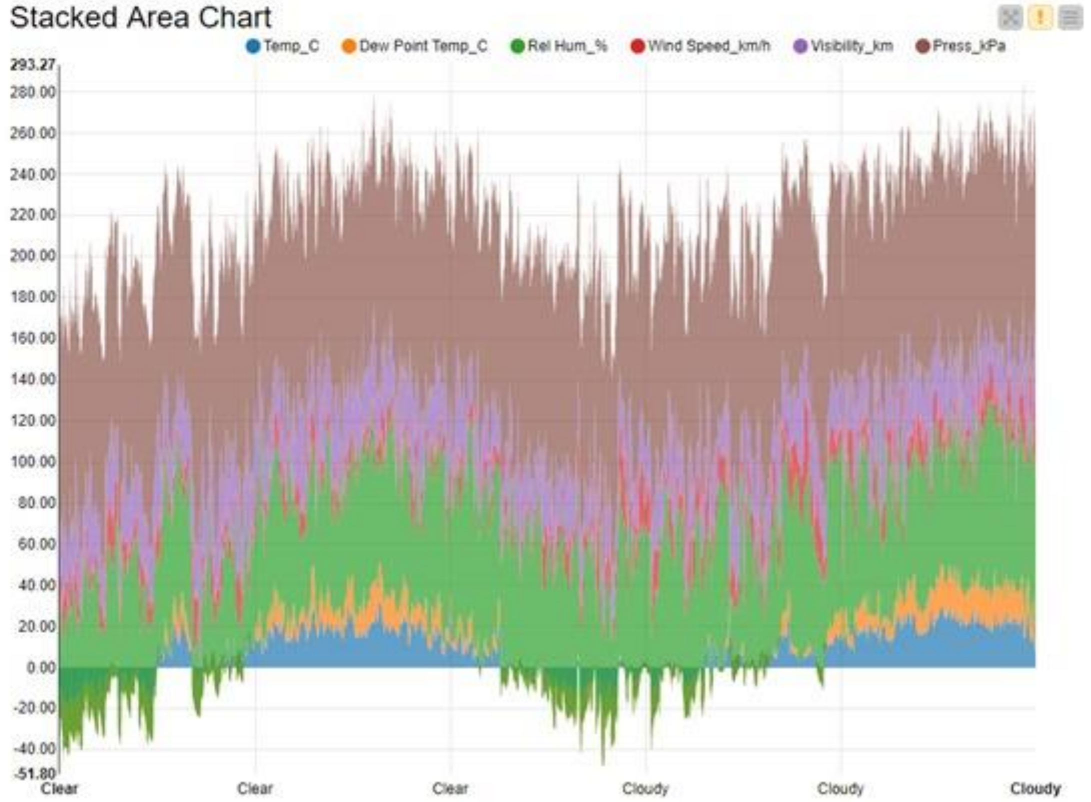

Pie Chart
Beverages
Cosmetics
Fruits
Household
Personal Care
5,070.00
5,124.00
1,815.00
2,370.00
11,220.00
Reset
Apply
Close



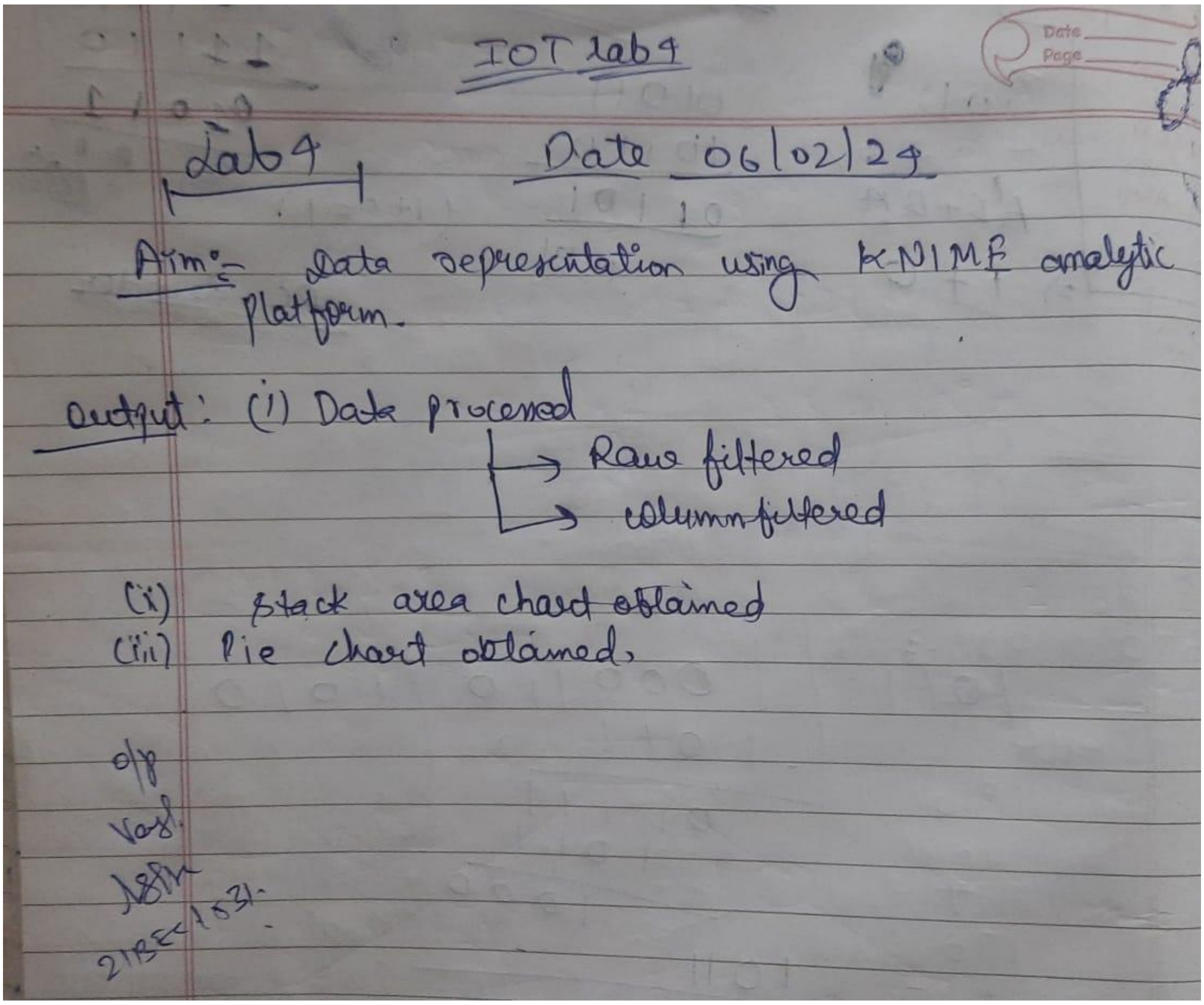
It shows weather of a particular area and they are depicted in the form of different colours in the pie chart.

Stacked area chart:





Verification sign:



IOT lab4

Date
Page 8

Lab4 Date 06/02/24

Aim:- Data representation using KNIME analytic platform.

Output: (i) Data processed
→ Raw filtered
→ column filtered

(ii) stack area chart obtained
(iii) Pie chart obtained,

o/p
Verf.
NSM
21BEC1631.

Result: The dataset was visualized using KNIME, revealing its composition through a pie chart and illustrating trends over time with a stacked area chart. The pie chart showcased categorical distribution, while the stacked area chart depicted numerical trends.

Name: Aaditya Bhatt
Reg.No: 21BEC1531

EXP 5:

Smart Home automation using CISCO-Packet tracer

Software used: Cisco-Packet tracer.

AIM:

To create a IoT home automation environment in cisco packet tracer.

SOFTWARE REQUIRED:

CISCO PACKET TRACER

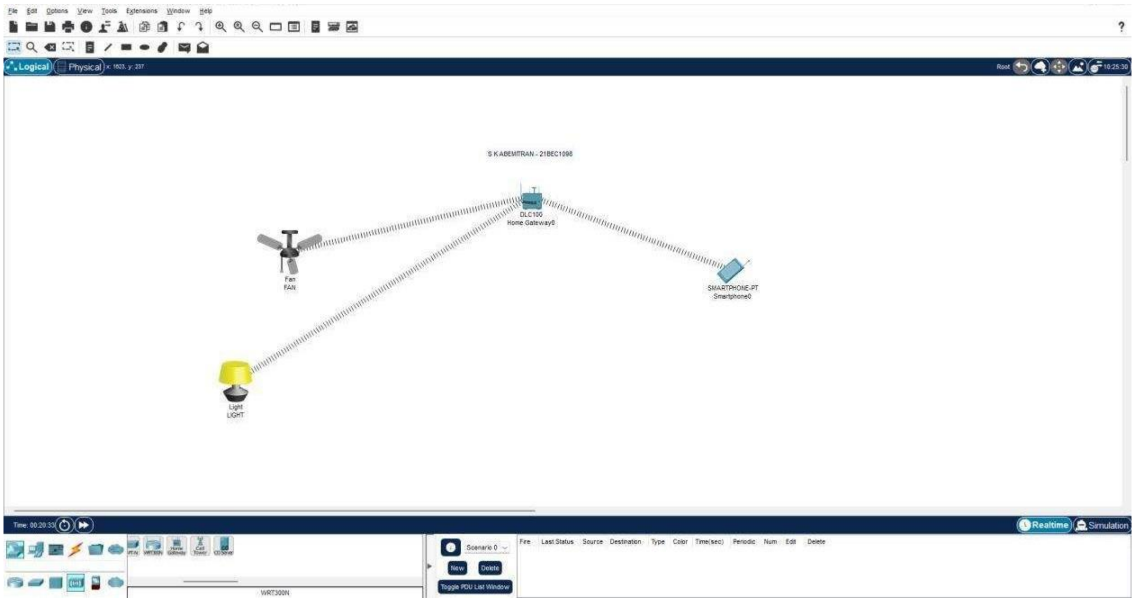
THEORY:

- Cisco Packet Tracer is a network simulation tool.
- It enables users to design, configure, and troubleshoot network setups.
- It supports various networking devices like routers, switches, and end devices.
- Packet Tracer facilitates learning by simulating real-world network scenarios.
- It's widely used in educational settings for teaching networking concepts.
- Users can experiment with different configurations without affecting real networks.

PROCEDURE:

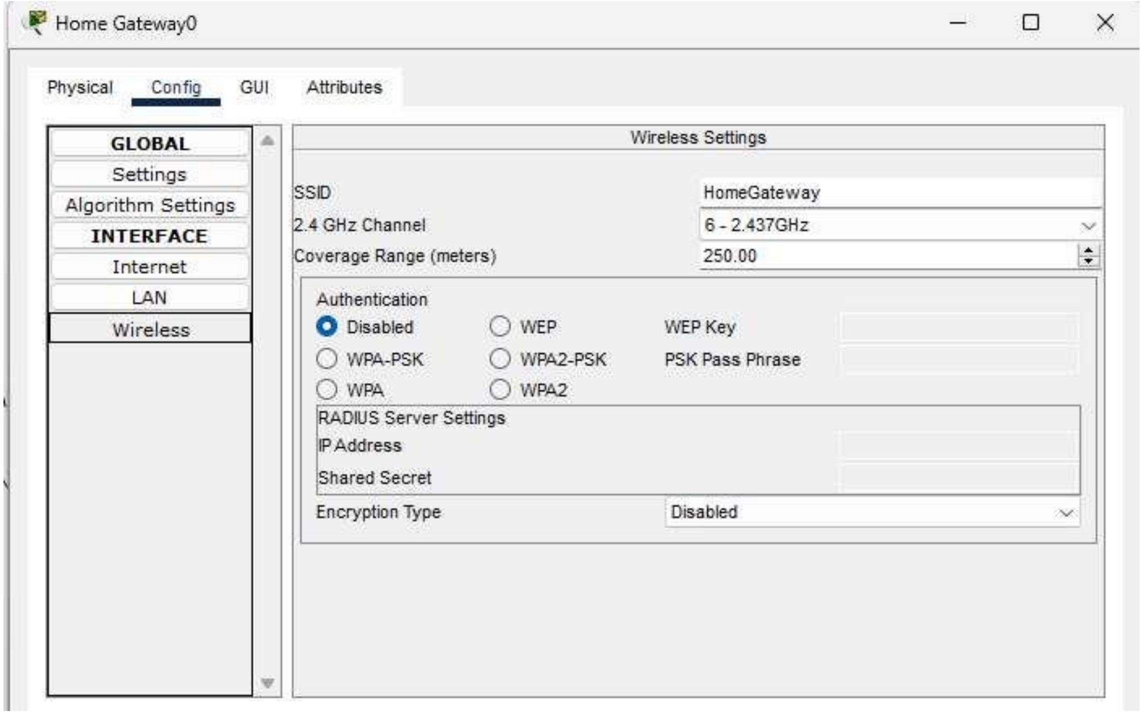
- Open Cisco Packet Tracer application and set up the devices as shown below
- Set up the SSID in the smart phone same as in the home gateway device
- Ensure that the IoT server of the home device is selected as home gateway.
- In the smart phone option, select desktop and select IoT monitor app.
- Login with the present credentials and the options to control the home devices are seen.

LAYOUT:


File Edit Options View Tools Extensions Window Help
Logical
Physical
S K ABEMITRAN - 21BEC1098
DLC100
Home Gateway0
Fan
FAN
SMARTPHONE-PT
Smartphone0
Light
LIGHT
Realtime
Simulation
Scenario 0
New
Delete
Toggle PDU List Window
Fire Last Status Source Destination Type Color Time(sec) Periodic Num Edit Delete
WRT300N

SET UP:

HOME- GATEWAY:



Wireless Settings

SSID: HomeGateway

2.4 GHz Channel: 6 - 2.437GHz

Coverage Range (meters): 250.00

Authentication: ● Disabled ○ WEP ○ WPA-PSK ○ WPA2-PSK ○ WPA ○ WPA2

WEP Key

PSK Pass Phrase

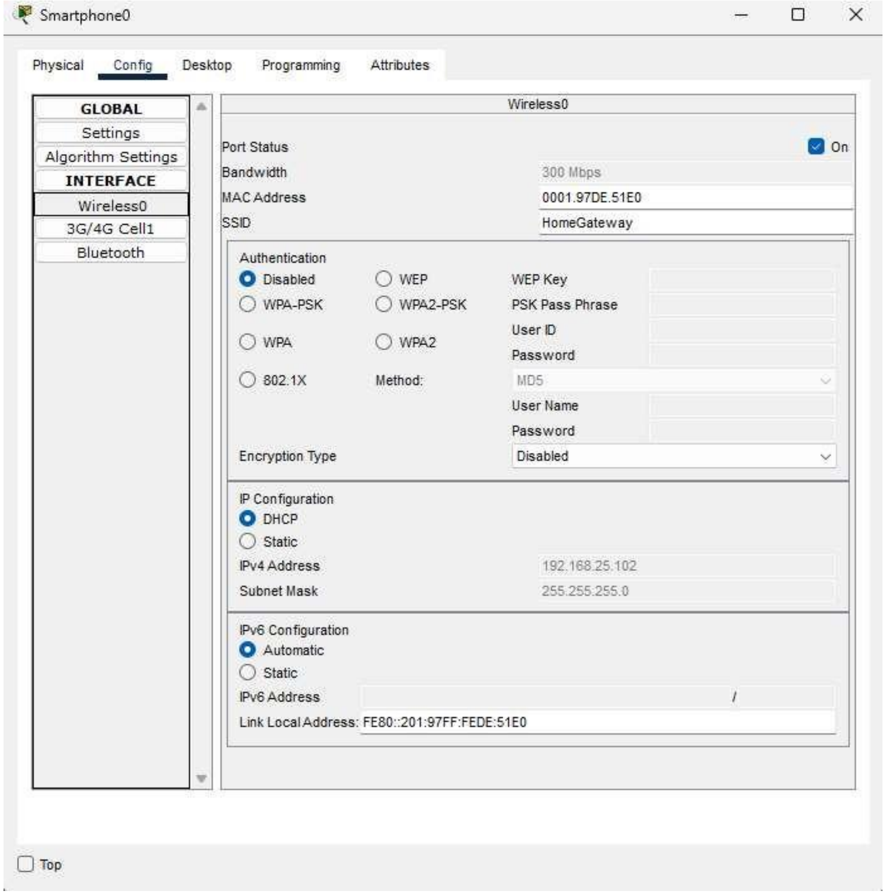
RADIUS Server Settings

IP Address

Shared Secret

Encryption Type: Disabled

SMART-PHONE:



Wireless0

Port Status: ☑ On

Bandwidth: 300 Mbps

MAC Address: 0001.97DE.51E0

SSID: HomeGateway

Authentication: ● Disabled ○ WEP ○ WPA-PSK ○ WPA2-PSK ○ WPA ○ WPA2 ○ 802.1X

WEP Key

PSK Pass Phrase

User ID

Password

Method: MD5

User Name

Password

Encryption Type: Disabled

IP Configuration: ● DHCP ○ Static

IPv4 Address: 192.168.25.102

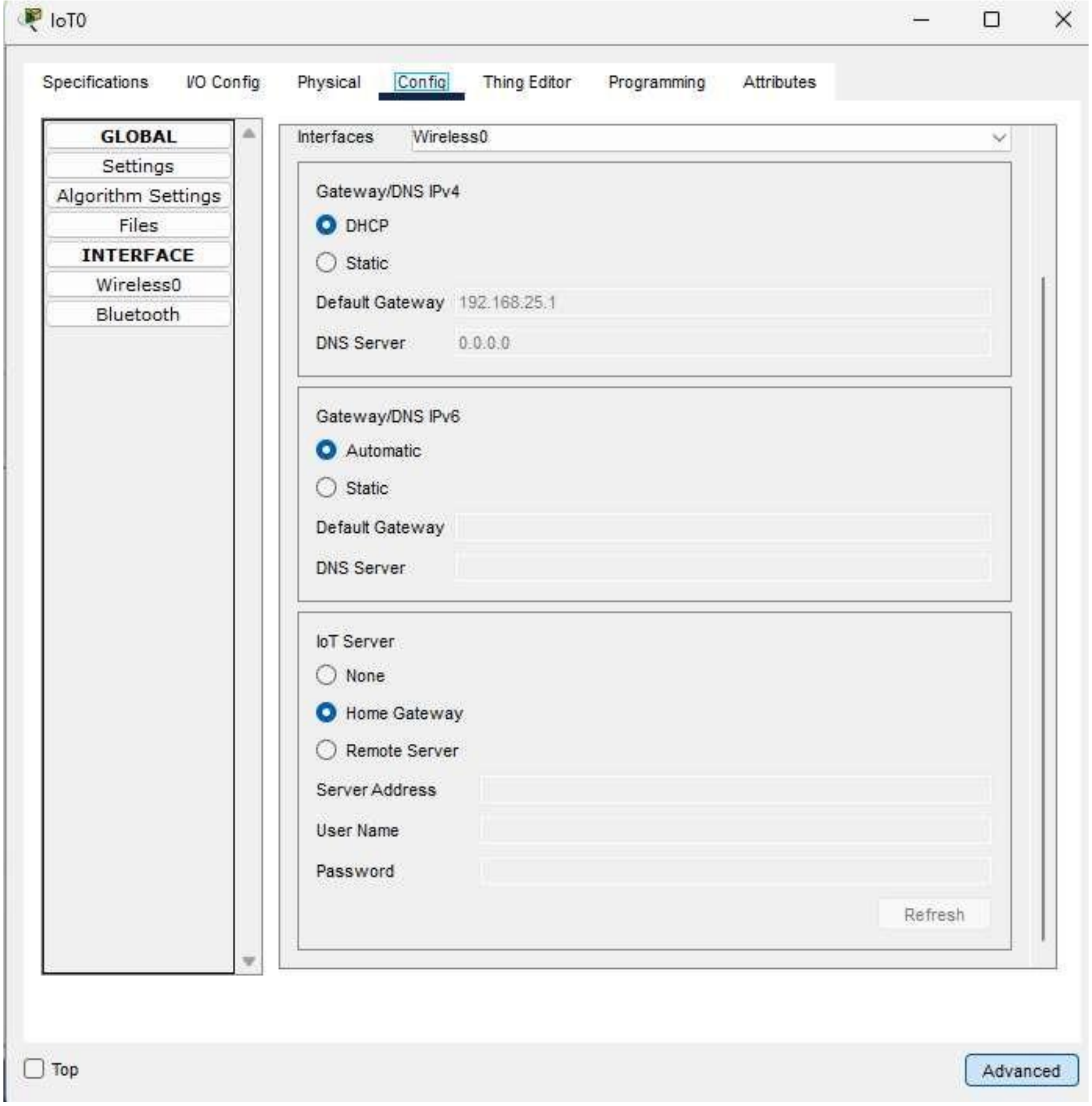
Subnet Mask: 255.255.255.0

IPv6 Configuration: ● Automatic ○ Static

IPv6 Address: /

Link Local Address: FE80::201:97FF:FEDE:51E0

HOME DEVICE (fan and light):



Interfaces: Wireless0

Gateway/DNS IPv4

● DHCP

○ Static

Default Gateway: 192.168.25.1

DNS Server: 0.0.0.0

Gateway/DNS IPv6

● Automatic

○ Static

Default Gateway

DNS Server

IoT Server

○ None

● Home Gateway

○ Remote Server

Server Address

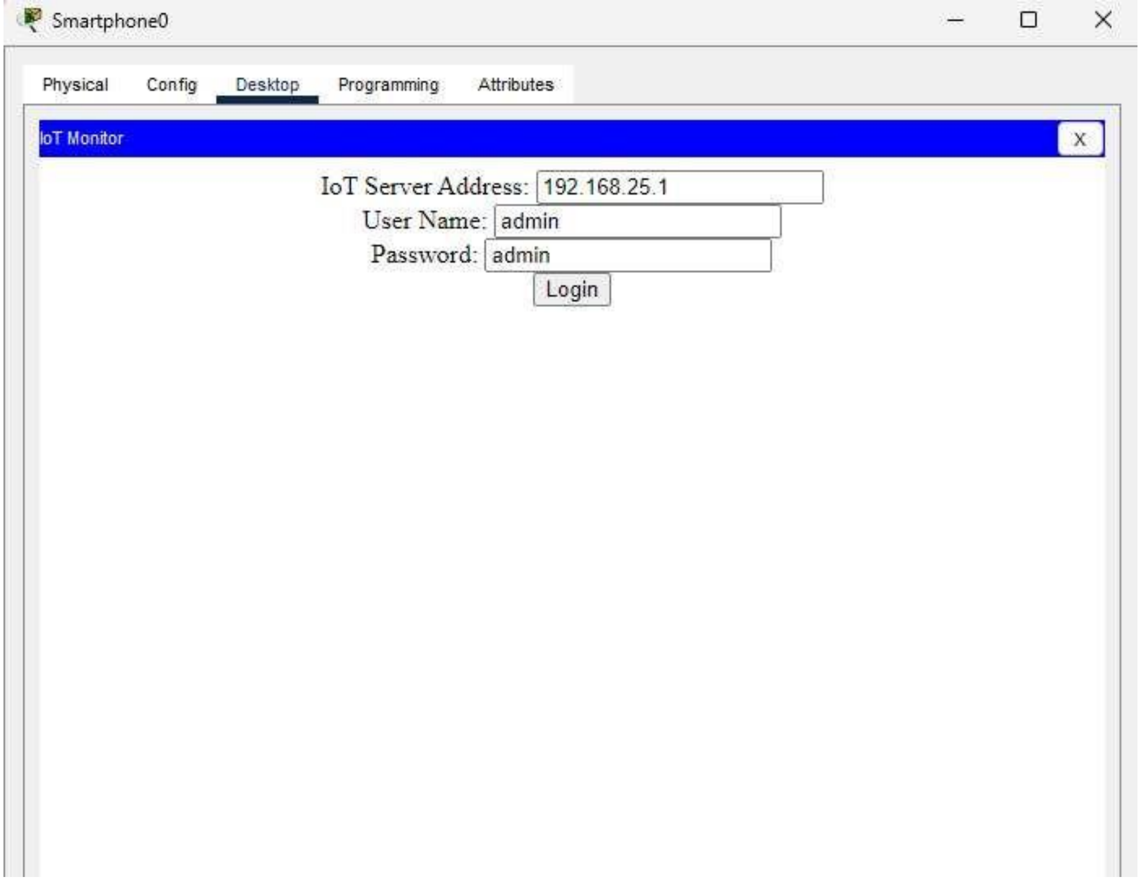
User Name

Password

□ Top

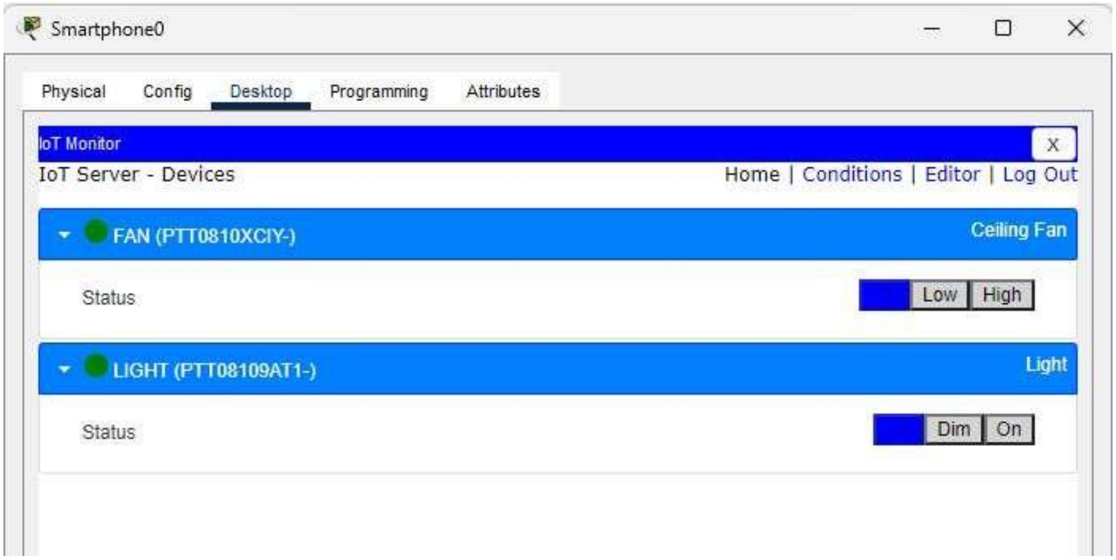
OUTPUT:

DEVICE MONITORING APP LOGIN:



IoT Server Address: 192.168.25.1
User Name: admin
Password: admin

DEVICE LIST:

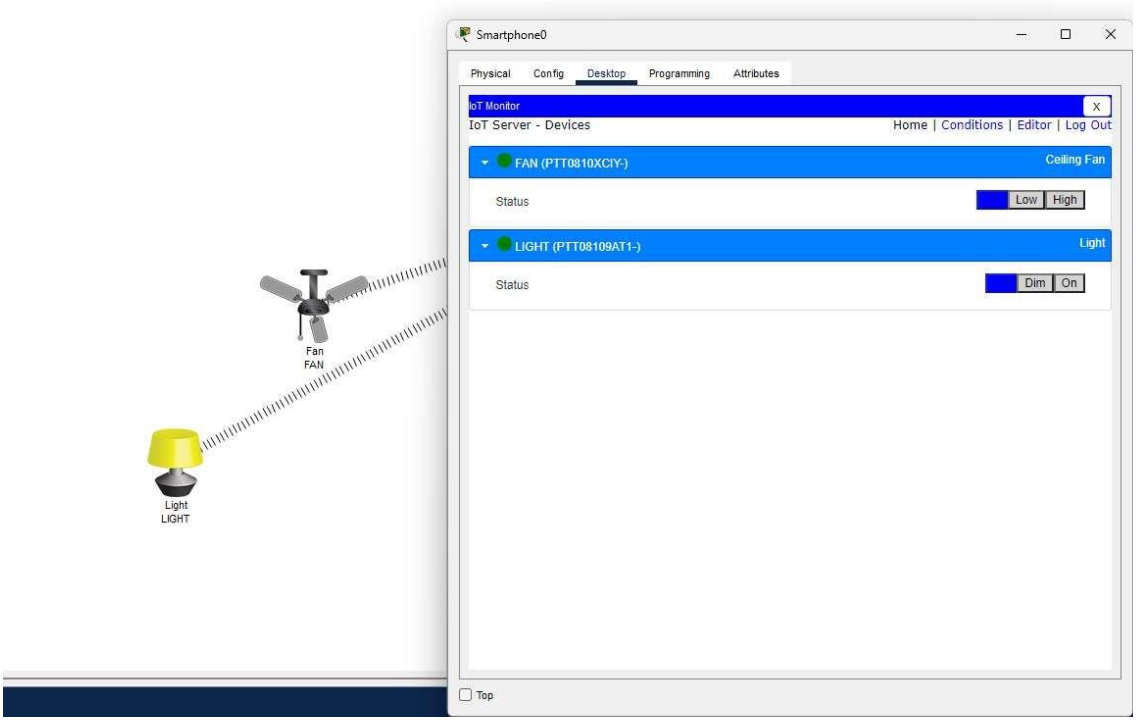


IoT Server - Devices

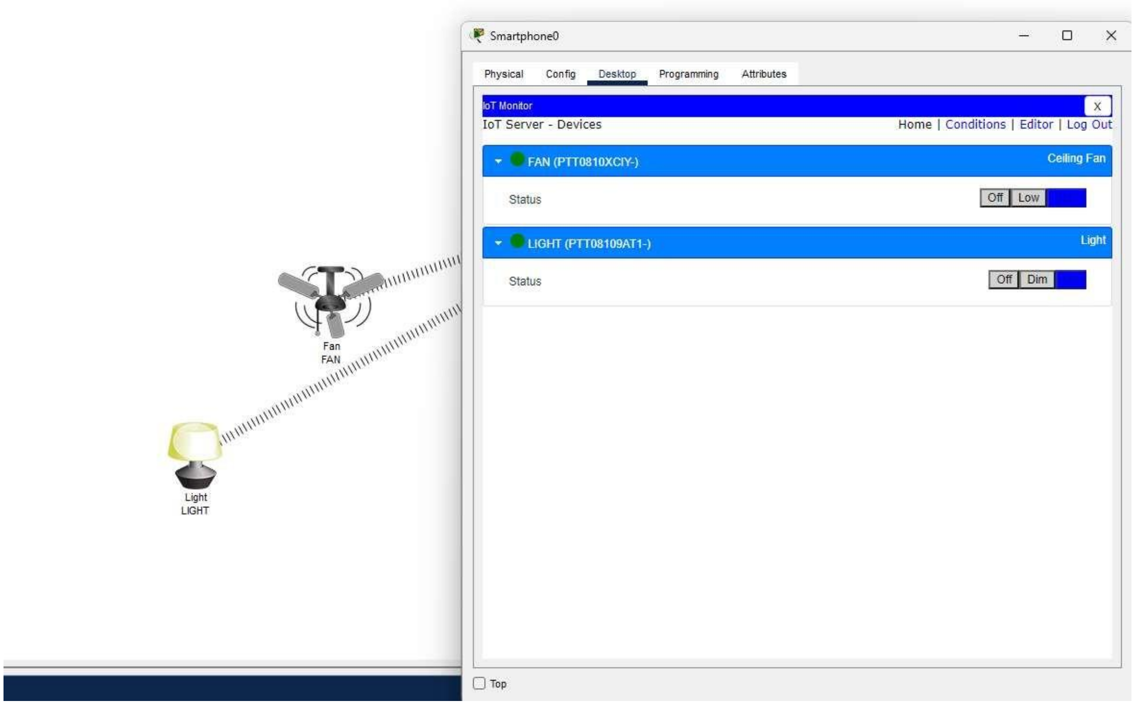
● FAN (PTT0810XCIY-) Ceiling Fan
Status

● LIGHT (PTT08109AT1-) Light
Status

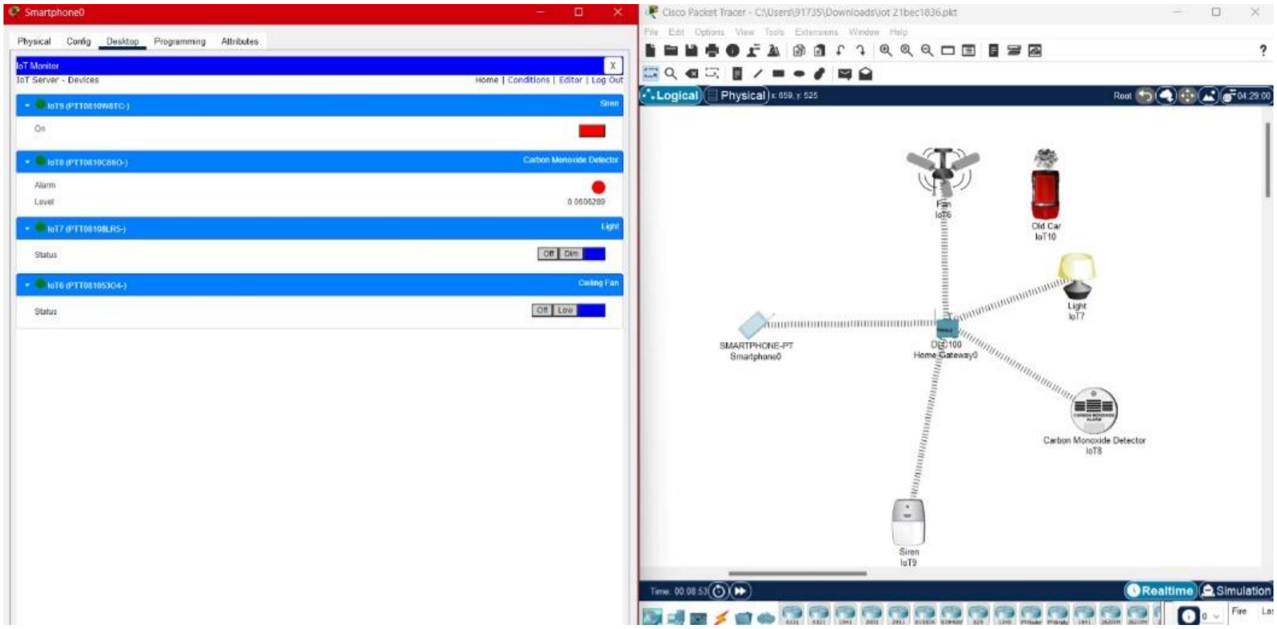
DEVICES IN TURNED OFF CONDITION:



DEVICES IN TURNED ON CONDITION:



RESULT:



Smartphone0

Physical | Config | Desktop | Programming | Attributes

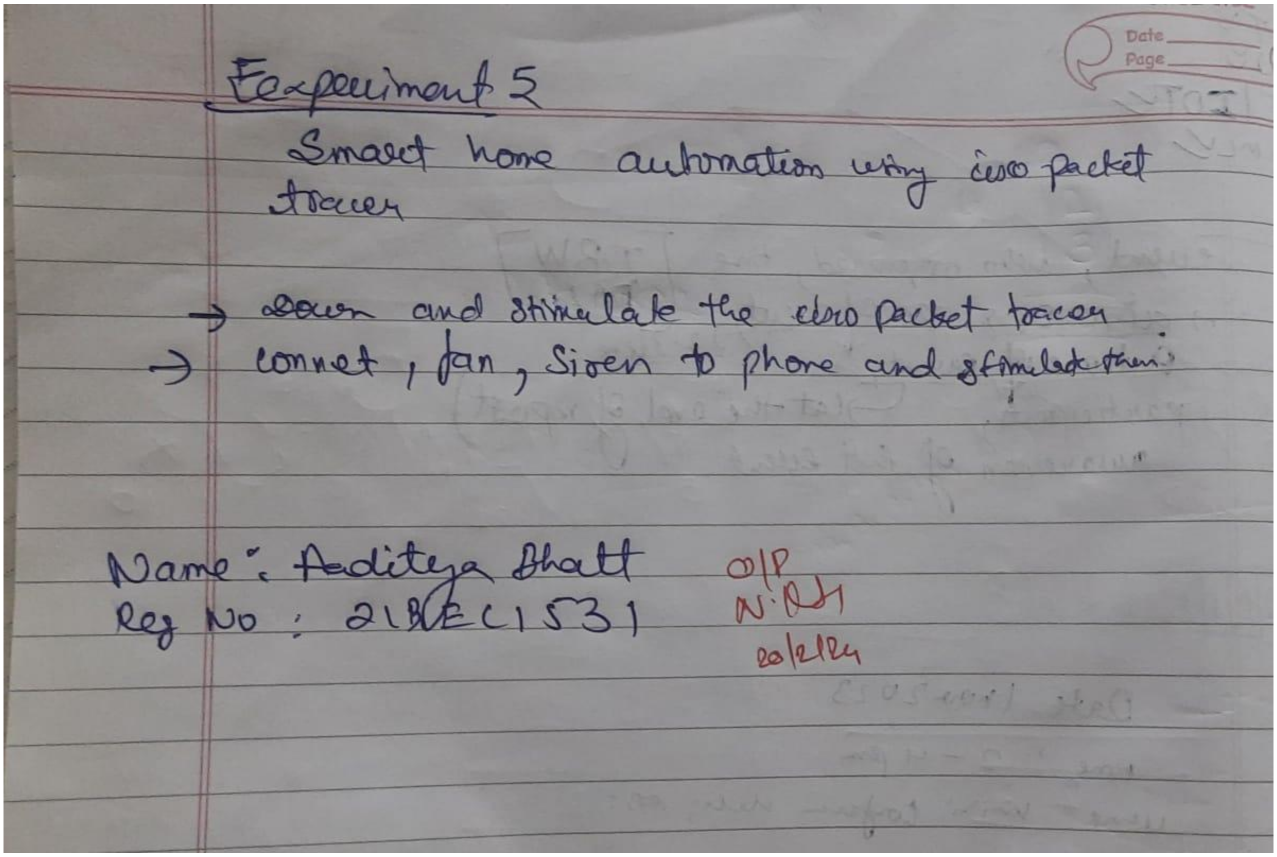
IoT Monitor

IoT Server - Devices — Home | Conditions | Editor | Log Out

- IoT9 (PTT0810W8TC-) — Siren
  - On: 🟥
- IoT8 (PTT0810C88O-) — Carbon Monoxide Detector
  - Alarm: ●
  - Level: 0.0606289
- IoT7 (PTT08108LR5-) — Light
  - Status: Off | Dim
- IoT6 (PTT08105304-) — Ceiling Fan
  - Status: Off | Low

Cisco Packet Tracer - C:\Users\91735\Downloads\iot 21bec1836.pkt

Verification sign:



Experiment 5

Smart home automation using cisco packet tracer

→ Learn and stimulate the cisco packet tracer.

→ connet, fan, Siren to phone and stimulate them

Name: Aditya Bhatt

Reg No: 21BEC1531

O/P

20/2/24

Result:

In Cisco Packet Tracer, a home automation system was implemented, enabling remote control of household devices such as lights, thermostats, fan, siren. The system offers seamless integration with mobile devices for convenient management and monitoring of home functionalities.