IOT-BASED REAL-ESTATE MANAGEMENT APPLICATION WITH AIR QUALITY MONITORING

By

GAURAV VINOD BHAMBHANI
18BCE072

NIRMA UNIVERSITY
INSTITUTE OF TECHNOLOGY
NAAC ACCREDITED 'A' GRADE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
Ahmedabad 382481

IOT-BASED REAL-ESTATE MANAGEMENT APPLICATION WITH AIR QUALITY MONITORING

Minor Project Report

Submitted in partial fulfillment of the requirements

For the degree of

Bachelor of Technology in Computer Science & Engineering

By

GAURAV VINOD BHAMBHANI
18BCE072

Guided By
DR. MOHD ZUHAIR
[DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING]

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
Ahmedabad 382481

CERTIFICATE

This is to certify that the minor project entitled "IOT-based Real-Estate Management Application with Air Quality Monitoring" submitted by Gaurav Vinod Bhambhani (18BCE072), towards the partial fulfillment of the requirements for the degree of Bachelor of Technology in Computer Science and Engineering of Nirma University is the record of work carried out by him/her under my supervision and guidance. In my opinion, the submitted work has reached the level required for being accepted for examination.

Dr. Mohd Zuhair,
Assistant Professor,
Computer Science and Engineering Dept.,
Institute of Technology,
Nirma University,
Ahmedabad

Dr. Madhuri Bhavsar,
Professor and HOD,
Computer Science and Engineering Dept.,
Institute of Technology,
Nirma University,
Ahmedabad

ACKNOWLEDGEMENT

I would like to start by acknowledging the strength, energy, and patience the almighty GOD bestowed upon me to start and accomplish this work with the support of all concerned, a few of them I am trying to name hereunder.

I would like to express my gratitude and gratefully acknowledge the help of Dr. Mohd Zuhair, under whose guidance I worked on this minor project 'IoT-based Real-Estate Management Application with Air Quality Monitoring'.
He was always available for consultation and the successful completion of this research would not have been possible without him.

I would also like to thank all my friends who have, directly or indirectly, helped me with the completion of this research.

No words are adequate to express my indebtedness to my parents and for their blessings and good wishes. To them, I bow in the deepest reverence.

- GAURAV BHAMBHANI (18BCE072)

ABSTRACT

The objective of this minor project entitled “IOT-based Real-Estate Management Application with Air Quality Monitoring” was to develop a mobile application that would make finding healthy homes easier.

For this project, I worked with devices like ESP8266, which helps us connect to the WiFi and the temperature sensor DHT11, to acquire real-time air temperature and humidity data.

I then connected my equipment to the cloud, for which I used ThingSpeak, using the ESP8266 WiFi module, and collected and stored the sensor data on the cloud.

Next, I developed the mobile application using Flutter, which displays homes for sale in different areas with the information of the temperature and humidity of that home.

CONTENTS

1 INTRODUCTION

1.1 Prologue

In today's world, where pollution has become a major problem, especially for patients with terminal illnesses like asthama and even elderly folks, the need for a solution that facilitates the searching of healthy living areas has become a necessity. It is with this mission, I have tried to develop a solution that uses IoT and mobile application development.
For demonstration purposes, I have used a DHT11 sensor that collects temperature and humidity information of the particular area, and a WiFi module ESP8266 that helps connects the sensor to the cloud where the current data is transmitted every 15 seconds.
I then developed a mobile application that enables its users to find healthy living homes by displaying the air quality data, area-wise.

2 TOOLS AND TECHNOLOGIES

2.1 HARDWARE

2.1.1 ESP8266

The ESP8266, produced by Espressif Systems, is a highly-integrated WiFi microcontroller unit, with an integrated TCP/IP protocol stack that enables any microcontroller to access any WiFi network.
These microcontroller chips have been succeeded by the ESP32 microcontroller.

I installed the libraries for ESP8266 by adding the json file for ESP8266 into the Additional Boards Manager URLs field in the Preference window in Arduino, and installed the ESP8266 board in the Board Manager.

2.1.2 DHT11

The DHT11 is a basic, low-cost digital temperature and humidity sensor.

It uses a capacitive humidity sensor and a thermistor to measure the surrounding air and spits out a digital signal on the data pin.

2.2 SOFTWARE

2.2.1 ARDUINO IDE

The Arduino Integrated Development Environment is a cross-platform application.
It is used to write and upload programs to Arduino compatible boards, but also, with the help of third-party cores, other vendor development boards.

2.2.2 ThingSpeak

ThingSpeak is an IoT analytics platform service that allows you to aggregate, visualize, and analyze live data streams in the cloud.
It is an open-source software written in Ruby which allows users to communicate with internet enabled devices. It facilitates data access, retrieval and logging of data by providing an API to both the devices and social network websites.

2.2.3 Flutter

Flutter is an open-source UI software development kit created by Google. It is used to develop cross platform applications for Android, iOS, Linux, Mac, Windows, etc. from a single codebase.

3 BACKEND

3.1 ARDUINO

After connecting the ESP8266 with the DHT11 sensor and running the following code:

connect | Arduino 1.8.16

File Edit Sketch Tools Help

connect

```
#include <DHT.h>  // Including library for dht

#include <ESP8266WiFi.h>

String apiKey = "65Z381PVBK123AMS";     //  Enter your Write API key from ThingSpeak

const char *ssid =  "GDMH";     // replace with your wifi ssid and wpa2 key
const char *pass =  "hitansh26";
const char* server = "api.thingspeak.com";

#define DHTPIN 0          //pin where the dht11 is connected

DHT dht(DHTPIN, DHT11);

WiFiClient client;

void setup()
{
        Serial.begin(9600);
        delay(10);
        dht.begin();

        Serial.println("Connecting to ");
        Serial.println(ssid);


        WiFi.begin(ssid, pass);

       while (WiFi.status() != WL_CONNECTED)
      {
             delay(500);
             Serial.print(".");
      }
       Serial.println("");
```

```
IROM   : 247924          - code in flash         (default or ICACHE_FLASH_ATTR)
IRAM   : 27337   / 32768 - code in IRAM          (IRAM_ATTR, ISRs...)
```

Type here to search 30°C ENG 12:19 PM 28-11-2021

connect | Arduino 1.8.16

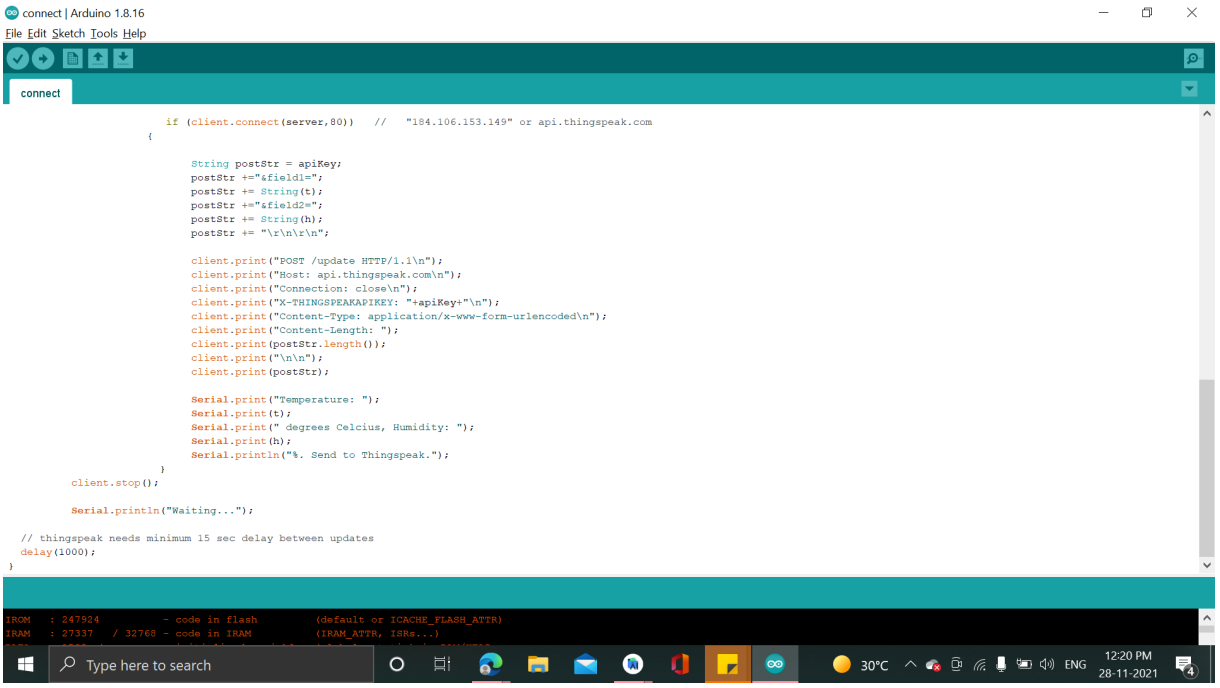
File Edit Sketch Tools Help

connect

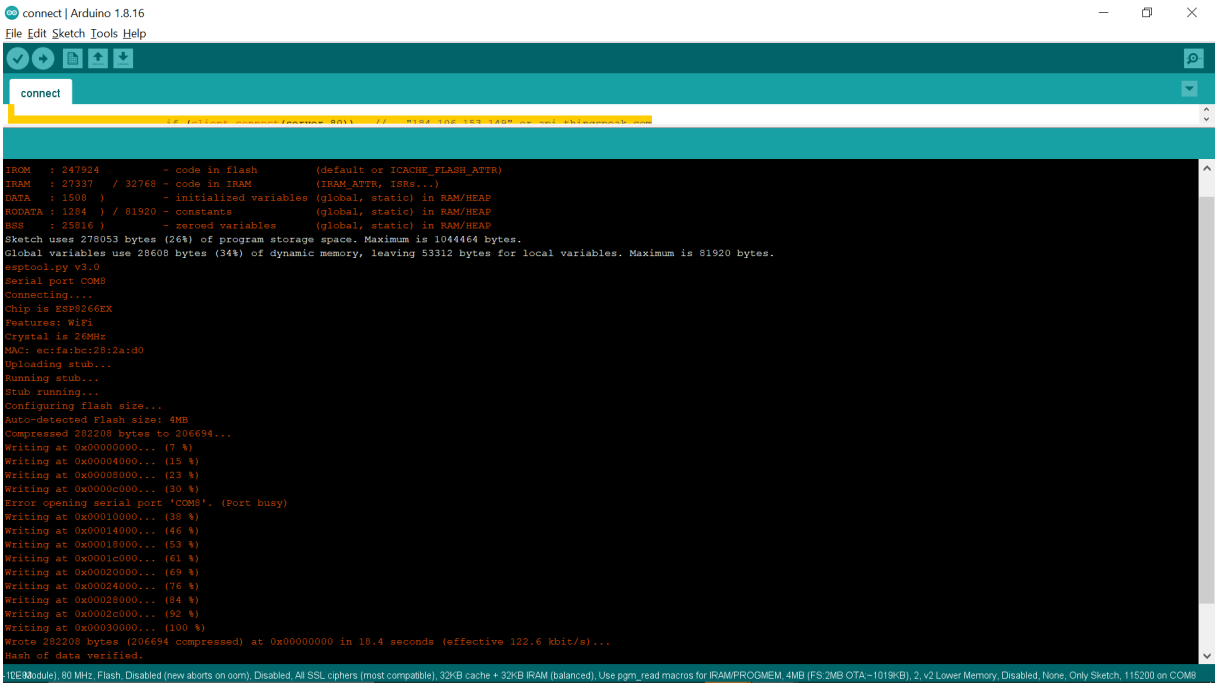
```
       while (WiFi.status() != WL_CONNECTED)
      {
             delay(500);
             Serial.print(".");
      }
       Serial.println("");
       Serial.println("WiFi connected");

}

void loop()
{

       float h = dht.readHumidity();
       float t = dht.readTemperature();

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

                     if (client.connect(server,80))   //   "184.106.153.149" or api.thingspeak.com
                  {

                          String postStr = apiKey;
                          postStr +="&field1=";
                          postStr += String(t);
                          postStr +="&field2=";
                          postStr += String(h);
                          postStr += "\r\n\r\n";

                          client.print("POST /update HTTP/1.1\n");
```

```
IROM   : 247924          - code in flash         (default or ICACHE_FLASH_ATTR)
IRAM   : 27337   / 32768 - code in IRAM          (IRAM_ATTR, ISRs...)
```

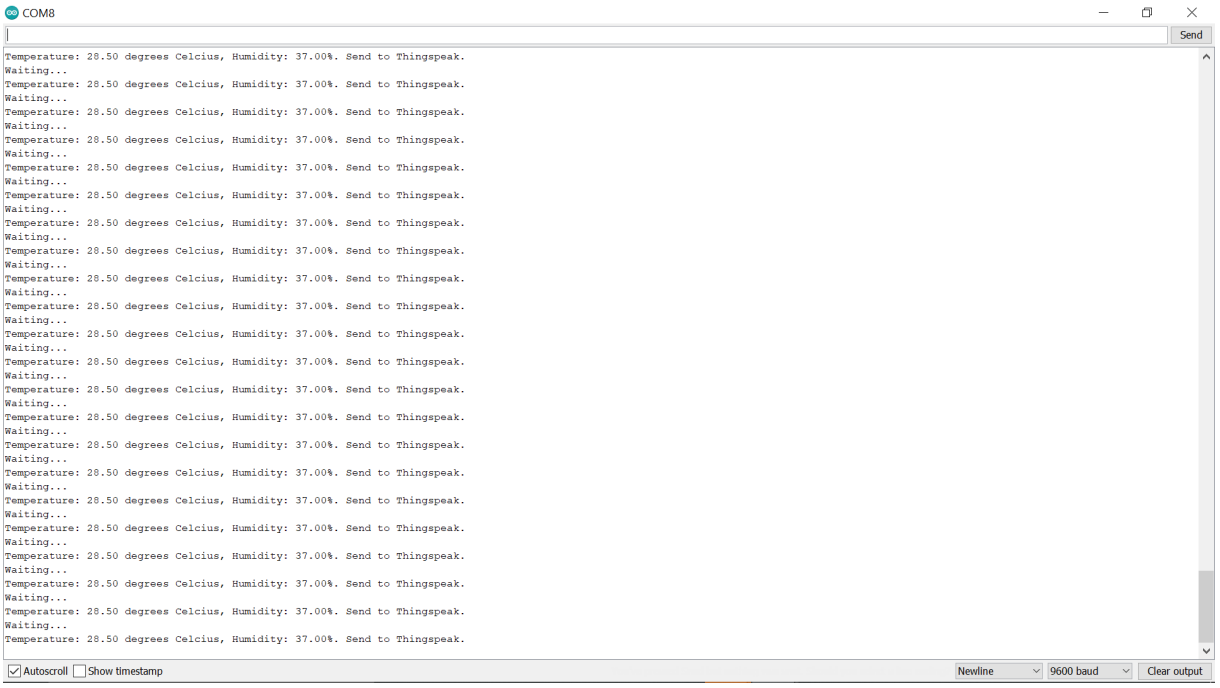
Type here to search 30°C ENG 12:20 PM 28-11-2021



we get the following output:

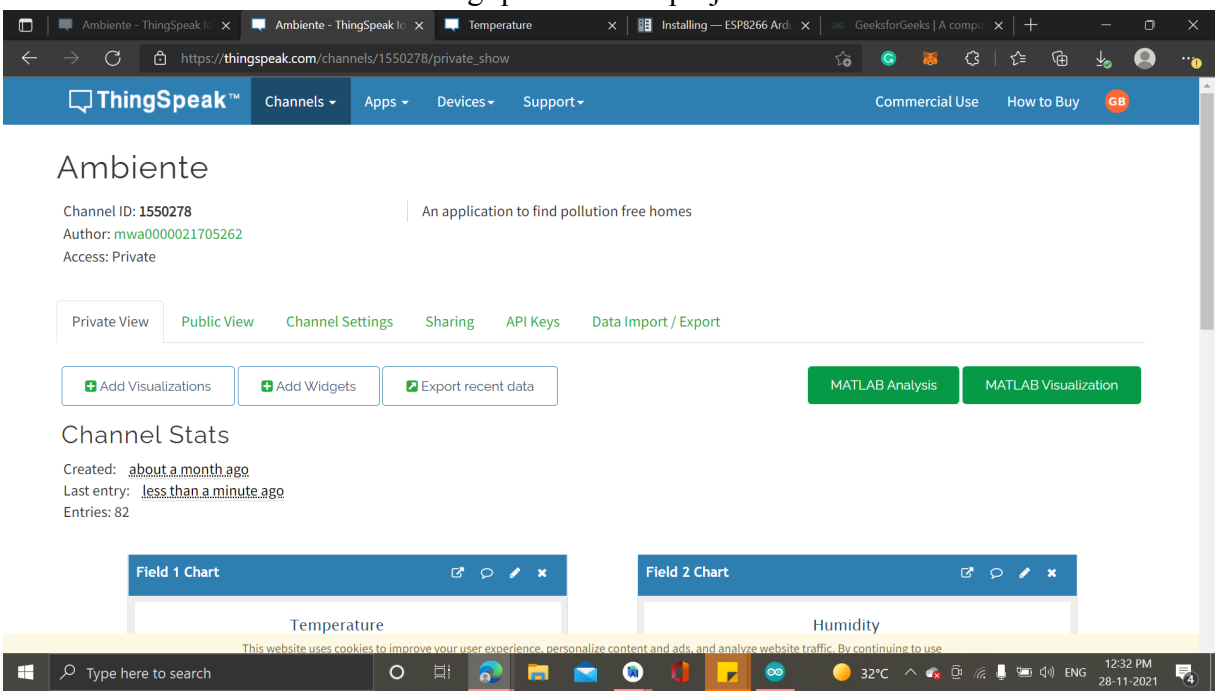


Hence, on the serial monitor, we get the following readings for temperature and humidity from the DHT11 sensor, which gets sent to ThingSpeak:

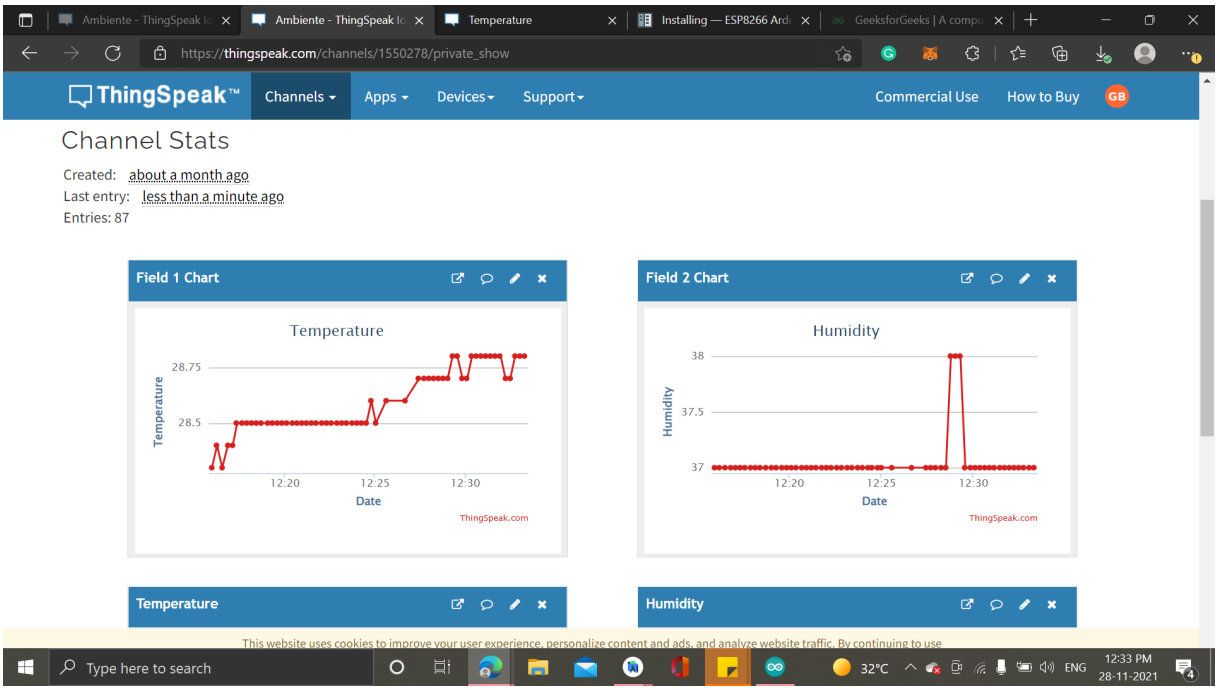


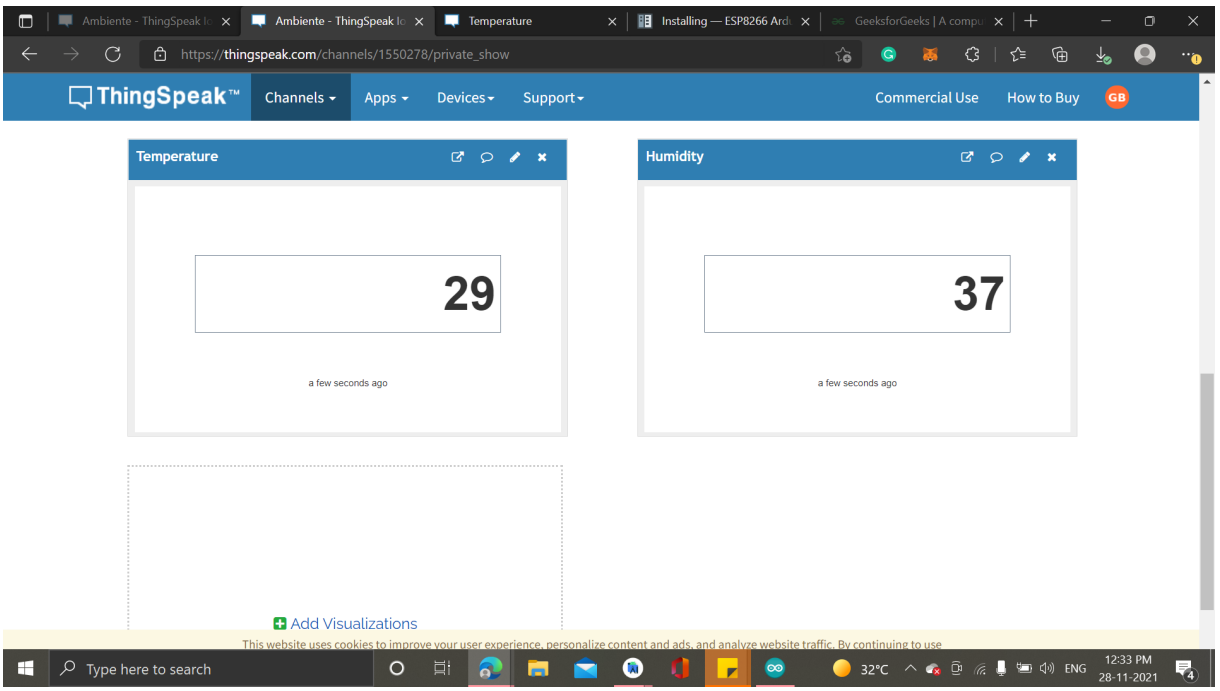
3.2 ThingSpeak

This is the channel I created on ThingSpeak for this project.



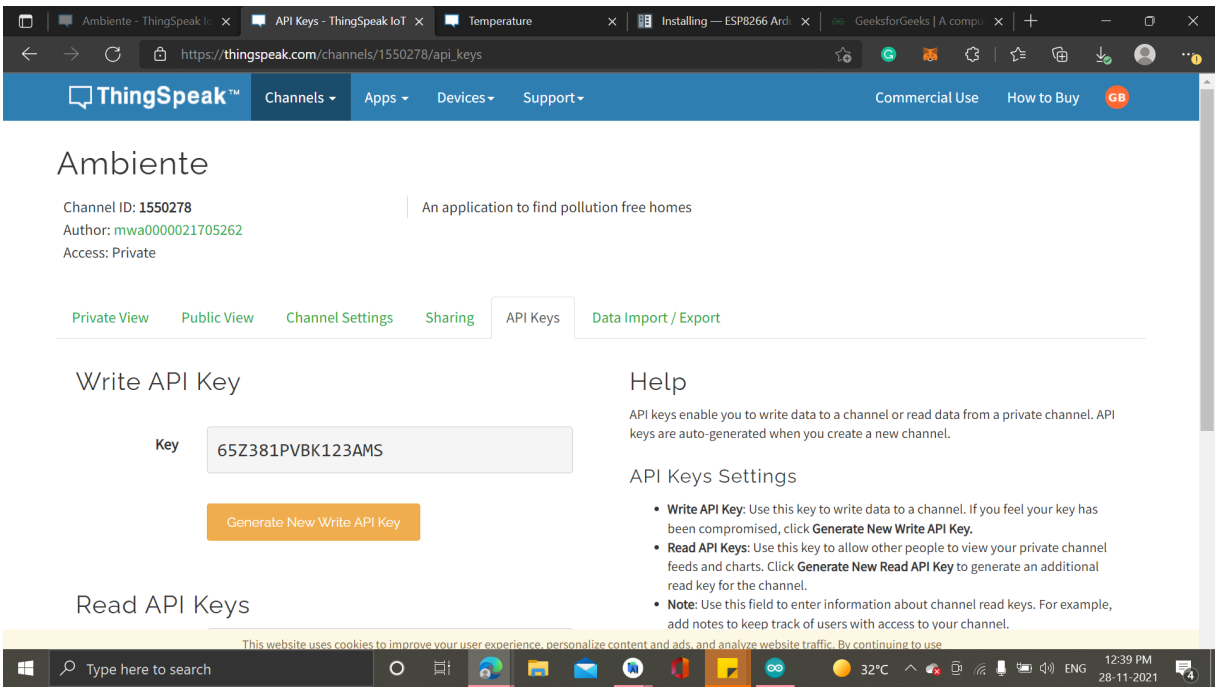
I have used 4 visualization tools for this project, which involves graphical representations as well as textual representation of data.





The values are updated every 15 seconds on ThingSpeak.

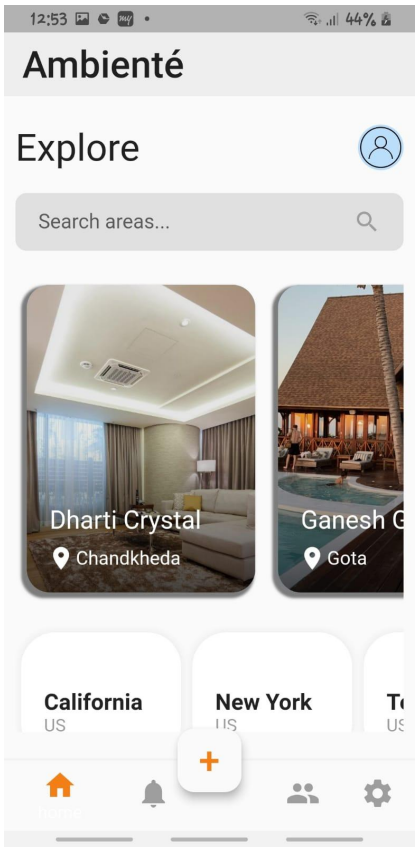
To connect the Arduino code with ThingSpeak, I used the API key generated by ThingSpeak, which enables us to write the data generated by the sensor onto the ThingSpeak platform.

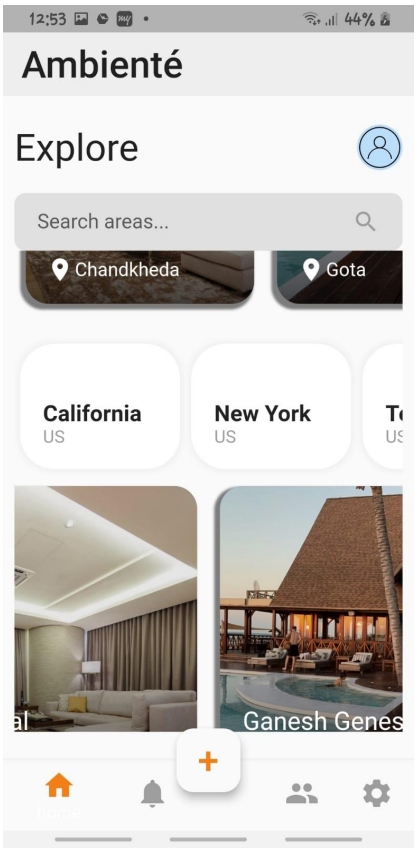


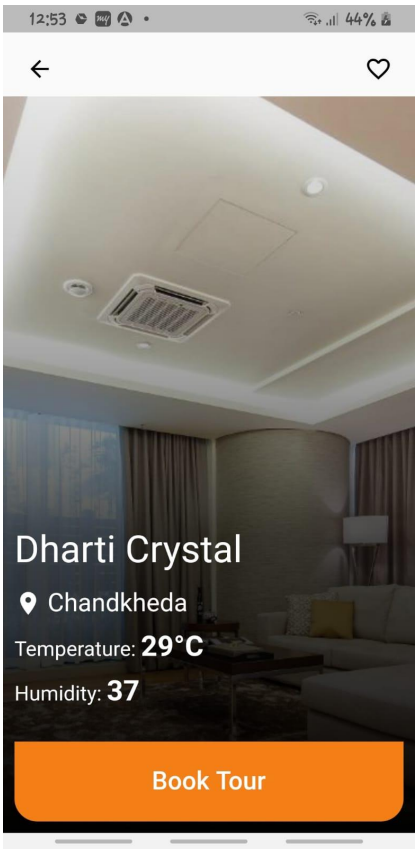
4 FRONTEND

4.1 FLUTTER APPLICATION

I developed the real-estate mobile application using flutter, on android studio.







5 FUTURE WORKS

Since I have used only a DHT11 sensor for demonstration purposes, ahead in time, we can use various air pollution measuring sensors like FLOW^{EVO} sesnors to measure the amount of Sulphur Dioxide present in the air, NDIR and various other chemical gas sensors, and even sensors to measure the amount of oxygen in the air.

The mobile application can be upgraded by adding graphical representation of the collected data as well.

I also intend on using web scraping/harvesting to gather information of air quality of various other locations where the sensors might not be installed.

6 APPENDIX

1. Installing — ESP8266 Arduino Core 3.0.2-32-g076a4edf documentation (arduino-esp8266.readthedocs.io)
2. ThingSpeak Documentation - MathWorks India
3. ESP8266 - Setup and First WiFi Connection - Arduino Project Hub
4. ESP8266: DHT11 Temperature and Humidity Web Server - Hackster.io
5. DHT11 Humidity Sensor with ESP8266 and ThingSpeak (electronicshub.org)
6. DHT11 Humidity Temperature Monitor with NodeMCU on ThingSpeak (how2electronics.com)