

**SMART MASK**

by

JERLYN.S

(1741126)

TITUS.P.MATHEWS

(1741160)

Under the guidance of

Dr. Nizar Banu P K

&

Dr. Sivakumar R

A project report submitted in partial fulfillment of the requirements for the award of degree of Bachelors of Computer Applications of CHRIST (Deemed to be University), March – 2020.



CERTIFICATE

*This is to certify that the report titled Smart Mask is a bonafide record of work done by* ***Titus.P.Mathews(1741160)*** *and* ***Jerlyn.S(1741126)*** *of CHRIST (Deemed to be University), Bangalore, in partial fulfillment of the requirements of VI Semester BCA Major Project during the year 2019-20.*

**Head of the Department Project Guide**

Name :

Register Number :

Examination Centre : CHRIST (Deemed to be University)

Date of Exam :

Valued by:

1.

2.

**ACKNOWLEDGEMENTS**

Firstly, I would like to thank the Almighty God, for keeping us well during all these times and providing us with all the necessities.

I would like to express my special thanks of gratitude to our guides Dr.Nizar Banu.PK and Dr.Sivakumar.R as well as our Head of Department, Prof. Joy Paulose who gave us the golden opportunity to do this wonderful project, which also helped me in doing a lot of research and I came to know about a lot of new things. I am really thankful to them.

Secondly, I would also like to thank our family and friends who helped us a lot in finalizing this project within the limited time frame.

**Jerlyn.S**

**Titus.P.Mathews**

**ABSTRACT**

This project aims at developing a smart mask that is IOT based product that is capable of sensing the harmful gases present around while inhaling through the mask and also detect the temperature and humidity present around .This Project analyses and visualizes the amount of gases along with the current temperature and humidity present in terms of units and graphs. This project was proposed because air pollution is one among the most common problems faced by the entire globe. Since IOT can help in making devices more smarter and easier to use,this project would be a great contribution to the society . Even with respect to the current issue of Delhi pollution where the pollution level is in the state of crossing the severe category ,has also been one of the motivating factor for us to take up this project which can help an individual in determining the amount of harmful gases he/she takes in and brings an awareness among them.

This system will facilitate in keeping complete track of the air quality ppm levels(Parts Per Million) and temperature humidity track on day to day basis and would provide comparisons.

Future scope includes developing a wireless mask that could be easily carried by the users and would give detailed analysis of gases and temperature

**TABLE OF CONTENTS**

Acknowledgements…………………………………………………………………………...iii

Abstract…………………………………………………………………………………...…...iv

List of Tables………………………………………………………………………...……… vii

List of Figures……………………………………………………………………………..... viii

**[1](#_Toc27342316)****[INTRODUCTION](#_Toc27342316)** [1](#_Toc27342316)

**[1.1](#_Toc27342317)****[BACKGROUND OF THE PROJECT](#_Toc27342317)** [1](#_Toc27342317)

**[1.2](#_Toc27342318)****[OBJECTIVES](#_Toc27342318)** [1](#_Toc27342318)

**[1.3](#_Toc27342319)****[PURPOSE & SCOPE](#_Toc27342319)** [2](#_Toc27342319)

[1.3.1 PURPOSE 2](#_Toc27342320)

[1.3.2 SCOPE 2](#_Toc27342321)

**[1.4](#_Toc27342322)****[OVERVIEW OF THE PROJECT](#_Toc27342322)** [2](#_Toc27342322)

**[2](#_Toc27342323)****[SYSTEM ANALYSIS](#_Toc27342323)** [3](#_Toc27342323)

**[2.1](#_Toc27342324)****[EXISTING SYSTEMS](#_Toc27342324)** [3](#_Toc27342324)

**[2.2](#_Toc27342325)****[LIMITATIONS OF EXISTING SYSTEM](#_Toc27342325)** [3](#_Toc27342325)

**[2.3](#_Toc27342326)****[PROPOSED SYSTEM](#_Toc27342326)** [3](#_Toc27342326)

**[2.4](#_Toc27342327)****[BENEFITS OF PROPOSED SYSTEM](#_Toc27342327)** [4](#_Toc27342327)

**[2.5](#_Toc27342328)****[PRODUCT FEATURES](#_Toc27342328)** [5](#_Toc27342328)

**[2.6](#_Toc27342329)****[SOFTWARE AND HARDWARE REQUIREMENTS](#_Toc27342329)** [5](#_Toc27342329)

**[2.7](#_Toc27342330)****[DESIGN AND IMPLEMENTATION CONSTRAINTS](#_Toc27342330)** [7](#_Toc27342330)

**[2.8](#_Toc27342331)****[ASSUMPTIONS AND DEPENDENCIES](#_Toc27342331)** [7](#_Toc27342331)

[2.8.1 ASSUMPTIONS 7](#_Toc27342332)

[2.8.2 DEPENDENCIES 7](#_Toc27342333)

**[2.9](#_Toc27342334)****[FUNCTIONAL REQUIREMENTS](#_Toc27342334)** [7](#_Toc27342334)

**[2.10](#_Toc27342335)****[EXTERNAL INTERFACE REQUIREMENTS](#_Toc27342335)** [10](#_Toc27342335)

[2.10.1 USER INTERFACES 10](#_Toc27342336)

[2.10.2 HARDWARE INTERFACES 11](#_Toc27342337)

[2.10.3 SOFTWARE INTERFACES 11](#_Toc27342338)

[2.10.4 COMMUNICATION INTERFACES 12](#_Toc27342339)

**[2.11](#_Toc27342340)****[NON-FUNCTIONAL REQUIREMENTS](#_Toc27342340)** [12](#_Toc27342340)

[2.11.1 SAFETY REQUIREMENTS 12](#_Toc27342342)

[2.11.2 SECURITY REQUIREMENTS 12](#_Toc27342343)

[2.11.3 SOFTWARE QUALITY ATTRIBUTES 13](#_Toc27342344)

**[3](#_Toc27342345)****[SYSTEM DESIGN](#_Toc27342345)** [14](#_Toc27342345)

**[3.1](#_Toc27342346)****[SYSTEM ARCHITECTURE](#_Toc27342346)** [14](#_Toc27342346)

**[3.2](#_Toc27342347)****[MODULE DIAGRAM/ BLOCK DIAGRAM/ FUNCTIONAL DIAGRAM](#_Toc27342347)** [15](#_Toc27342347)

**[3.3 CIRCUIT DIAGRAM 1](#_Toc27342349)6**

**[3.4](#_Toc27342350)****[PIN DIAGRAM OF NODE MCU](#_Toc27342350)** [1](#_Toc27342350)6

**[3.5](#_Toc27342352)****[INTERFACE DESIGN AND PROCEDURAL DESIGN](#_Toc27342352)** [1](#_Toc27342352)7

[3.5.1 USER INTERFACE DESIGN 1](#_Toc27342353)7

[3.5.2 APPLICATION FLOW/ CLASS DIAGRAM 1](#_Toc27342354)8

**[4](#_Toc27342345)****[IMPLEMENTATION](#_Toc27342345)** [2](#_Toc27342345)4

**[4.1](#_Toc27342346)****[IMPLEMENATION APPROACHES](#_Toc27342346)** [2](#_Toc27342346)5

**[4.2](#_Toc27342347)****[CODING STANDARD](#_Toc27342347)** [2](#_Toc27342347)8

**[4.3](#_Toc27342348)****[CODING DETAILS](#_Toc27342348)** [3](#_Toc27342348)4

**[4.4](#_Toc27342349)****[SCREENSHOTS](#_Toc27342349)** [4](#_Toc27342349)0

**[5](#_Toc27342345)****[TESTING](#_Toc27342345)** [5](#_Toc27342345)0

**[5.1](#_Toc27342346)****[TEST CASES](#_Toc27342346)** [5](#_Toc27342346)1

**[5.2](#_Toc27342347)****[TEST APPROACHES](#_Toc27342347)** [5](#_Toc27342347)4

**[5.3](#_Toc27342347)****[TEST REPORTS](#_Toc27342347)** [5](#_Toc27342347)9

**[6](#_Toc27342345)****[CONCLUSION](#_Toc27342345)** [6](#_Toc27342345)1

**[6.1](#_Toc27342346)****[DESIGN AND IMPLEMENTATION](#_Toc27342346)** [6](#_Toc27342346)2

**[6.2](#_Toc27342347)****[ADVANTAGES AND LIMITATIONS](#_Toc27342347)** [6](#_Toc27342347)2

**[6.3](#_Toc27342347)****[FUTURE SCOPE OF THE PROJECT](#_Toc27342347)** [6](#_Toc27342347)3

**[APPENDIX](#_Toc27342357)** [6](#_Toc27342357)3

**[REFERENCES](#_Toc27342356)**

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Title** | **Page No.** |
| 2 | Different gas sensors | 11 |
| 2 | DHT11 sensor values | 11 |
| 3 | Software interfaces | 11 |
| 3 | Air quality ppm values | 16 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Fig No.** | **Figure Name** | **Page No.** |
| 1.1 | NODEMCU ESP8266 | 11 |
| 1.2 | Arduino IDE | 16 |
| 1.3 | DHT11 Sensor | 18 |
| 1.4 | MQ135 Sensor | 18 |
| 1.5 | Thingspeak | 18 |
| 32.1 | Temperature and humidity sensors | 18 |
| 3.1 | System Architecture | 19 |
| 3.2 | Block Diagram | 20 |
| 3.3 | Data Flow Diagram | 21 |
| 3.4 | Data Flow | 22 |
| 3.5 | Circuit Diagram | 23 |
| 3.6 | Pin Description | 24 |
| 3.7 | Widgets | 26 |
| 3.8 | Graphs | 26 |
| 4.1 | Main Page | 37 |
| 4.2 | SignUp Page | 38 |
| 4.3 | Login Page | 39 |
| 4.4 | AboutUs Page | 40 |
| 4.5 | Values Page | 41 |
| 4.6 | Channel Settings | 42 |
| 4.7 | API Keys | 43 |
| 4.8 | My Channels | 43 |
| 4.9 | Export Formats | 44 |
| 4.10 | CSV Formats | 46 |
| 4.11 | JSON Formats | 46 |
| 4.12 | Widgets ThingSpeak | 47 |
| 4.13 | Matlab Analysis ThingSpeak | 48 |
| 4.14 | Export Code | 49 |
| 4.15 | MyAccount | 50 |
| 4.16 | MyAccount ThingSpeak | 50 |
| 4.17 | MainPage Code Block | 51 |
| 4.18 | HomePage Code Block | 51 |
| 4.19 | InfoPage Code Block | 52 |
| 4.20 | LoginPage Code Block | 52 |
| 4.21 | HomePage | 53 |
| 4.22 | SignUpPage Code Block | 53 |
| 4.23 | MyProjects | 53 |
| 4.24 | UserInterface Component | 54 |
| 4.25 | Palette Description | 55 |
| 4.26 | Error Compiling Board | 56 |
| 4.27 | Port Not Detected | 57 |
| 4.28 | Library NotFound | 57 |
| 4.29 | Failed To Read DHT Sensor | 58 |
| 4.30 | DHT Pin Not Declared | 58 |
| 4.31 | Not a Number | 58 |
| 4.32 | WIFI Not Available | 59 |

# 1. INTRODUCTION

# 1.1. ABSTRACT

The capital city of India,New Delhi has been recognised as the most polluted city in the world.World Health Organization(WHO) has reported as per their findings in 2014.Continued exposure to environments with poor air quality is a major public health concern in developed and developing countries.It is indeed an alarming issue for the health of our future generation in India.Yet people are unaware of the ill effects of environmental pollutants and toxicants on health status of human as well as other living organisms and environment.

# 1.2. OBJECTIVES

The objectives of this project include:

1) To detect the harmful gases that an individual is inhaling.

2) To predict possible solution to reduce the intake of these gases

3) To provide a proper data analyzation of the gases inhaled by the user in the form of

graphs and numeric data

4) Easy view of the data on their smart devices (Laptops, Mobile phones)

5) To show the harmful gases inhaled on daily basis and to show the increase

or decrease in the intake level

# 1.3. PURPOSE & SCOPE

# 1.3.1. PURPOSE

This project aims at developing a SMART MASK which is IOT based that is capable of sensing the harmful gases present around that is inhaled through the mask, analyses it and also visualizes the amount of those gases present in terms of units and graphs. It was decided to propose this project because air pollution is among the most common problem faced by the entire globe. Since IOT is the current need of the hour and is making devices more smarter and easier to use, this project would be a great contribution to the society. Even with respect to the current issue of Delhi pollution where the pollution level is in the state of crossing the severe category, has also been one of the motivating factor to take up this project which can help an individual in determining the amount of harmful gases he/she takes in and brings an awareness among them.

# 1.3.2. SCOPE

The purpose of the SMART MASK is to make it easy to detect the harmful pollutants in the most subtle way by wearing a Smart Face Mask. The system is a IOT based project using air quality sensors and circuits to create a prototype of Mask that is capable to capture gases present around and provide proper data visualization of the pollutants inhaled. The project proposed would have a database to store the data of the gases inhaled and a data visualization tool that could manipulate the raw data into a organized data with graphical and numerical representations.

# 2.SYSTEM ANALYSIS

**2.1. EXISTING SYSTEM**

The existing air quality monitoring system is efficient for checking the ppm values of the air and determining its quality [1]. It can detect the various gases present in the air give accurate ppm values and based on the values determine if the air is good for the public to consume. The system uses Mobile application (SMART MASK) to show the values. It can only be used to determine the air quality, but this proposed project includes temperature and humidity sensors to increase the features of the device.[1]

**2.2. LIMITATIONS OF THE EXISTING SYSTEM**

* The existing system only determines the ppm values and does not provide proper data visualization.
* The system does not provide health hazards based on the ppm values and therefore is not of much use to the customers
* The existing system cannot be carried by users and has to be placed in one particular place.
* Existing system can only detect the air quality and does not include other features like detecting temperature, humidity.

**2.3. PROPOSED SYSTEM**

The system that is proposed overcomes the limitations mentioned thus providing a more functionally stable device in the form of a smart mask that can’t detect pollution level along with temperature and humidity present around us. It provides proper comparisons of the air quality, temperature and humidity and provides a interactive way in which the users can view the data of these gases, temperature and humidity levels on their android devices (Mobiles). The system also contains temperature and humidity sensors that adds more functionality to the mask. It can help in determining temperature details and humidity percentage along with various health concerns associated with the change in the air, temperature and humidity.

**2.4. BENEFITS OF PROPOSED SYSTEM**

* It helps in determining the increase in pollution level on a day to day basis which is stored in Thing Speak and visualized using mobile app (SMARTMASK)
* It provides users an interactive user interface to view the data in ppm range with numerical representations
* It could help users to carry the device along with them that increases the portability of the device
* Providing comparisons of air quality, temperature and humidity on day to day basis with their health hazards.
* Temperature and humidity detection is another feature we have added along with the air quality sensor detection

**2.5. SOFTWARE FEATURES**

The product will operate with the following software components and applications:

The product being developed will be running under Windows operating system using Arduino IDE as a coding platform.

The code would be uploaded to the controller which would further control the sensor. ThingSpeak would be used as a platform for

the visualization of the data.

**2.6. HARDWARE REQUIREMENTS**

The hardware that will be running these programs is being developed in Arduino uno R3 board with a Microprocessor which will contain the code. The gas sensor would be used for detecting air quality.

**2.7. DESIGN AND IMPLEMENTATION CONSTRAINTS**

* + Wireless integration of IOT components
  + Only specific gases can be detected
  + Fitting sensors to the mask
  + Expenses of the IOT devices
  + Real time database management

**2.8. ASSUMPTIONS AND DEPENDENCIES**

Every user who with the SMART MASK has a unique API Key coded on to the Node MCU board present inside the mask that specifies the particular ThingSpeak channel.Some of the most common sensor problems are down to the initial choice of sensor – without knowing what else is available on the market, many people choose the same sensors over and over again, replacing faulty or worn out ones with the same model. The material that a sensor needs to detect also plays a part in the right sensor choice, and is a factor that is often overlooked.Detecting different gases with a sensor is only applicable within short distances. The materials around the sensor, as well as those it is expected to detect, is a vital factor to think about when choosing the right sensor, otherwise the sensor may end up detecting things it is not supposed to [3].

**2.10. EXTERNAL INTERFACE REQUIREMENTS**

**2.10.1. USER INTERFACES :**

**Front-end software**: Arduino IDE and MIT app Inventor (SMART MASK)

**Back-end software**: ThingSpeak (Database)

**Notifications**: SMART MASK app

**2.10.2. HARDWARE INTERFACES :**

**USB:** Universal Serial Bus is a technology that allows a person to connect an electronic device to a micro controller. It is a fast serial bus.

**GPIO:** General-purpose input/output pins area generic pin on an integrated circuit or computer board whose behavior — including whether it is an input or output pin — is controllable by the user at run time.

**IC:** Integrated Circuit’s serial bus uses a protocol that enables multiple modules to be assigned a discrete address on the bus.

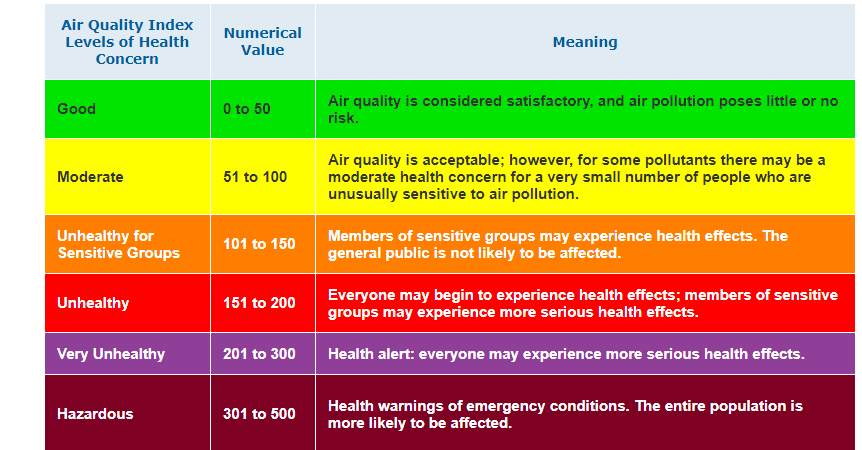
**2.10.3. SOFTWARE INTERFACES :**

Table 2.1 Softwarre interfaces used in our project

|  |  |
| --- | --- |
| **1.Operating system** | The project needs Windows operating system for its best support and user-friendliness |
| **2.Database** | ThingSpeak to store the data in the cloud and return the data for visualizing |
| **3.Arduino software** | It allows users to write and upload code to the NODEMCU ESP8266 within a real time work environment. |
| **4.Mobile Application** | SMART MASK application is developed to view the data in the simplest form in which users can view the data. |

**4.3. Air quality PPM values:[2]**

Table 2.2 Air quality PPM values



**2.10.4. COMMUNICATION INTERFACES :**

The communication between the different parts of the system is important since they depend on each other. However, in what way the communication is achieved is not important for the system and is therefore handled by the Wi-Fi module for both hardware devices such as NODEMCU ESP8266, sensor, etc. and the web portal.

**Mobile Application:** By using MIT app inventor SMART MASK app is created that can visualize the data from cloud database ThingSpeak from where the data is sent in JSON format to the application.

**ThingSpeak**: Things peak’s “Read URL” interacts directly with the mobile application and sends the data to the mobile.

**NodeMCU**: The NodeMCU interacts with the sensors MQ135 and DHT11 which in turn interacts with the atmosphere to collect the values.

**2.11. NON-FUNCTIONAL REQUIREMENTS**

**2.11.1. SAFETY REQUIREMENTS**

The power to the Arduino board is directly given from the desktop or laptop instead of a power supply whenever there is power failures as Arduino just requires a voltage of 5V.In case of disk crash or catastrophic failure the recovery method restores a past copy of the database that was backed up to archive storage.[4]

**2.11.3. SECURITY REQUIREMENTS**

1. Device and data security: Authentication of devices, its confidentiality and integrity of data is necessary.
2. Meeting all compliance requirements and requests
3. Meeting performance requirements as per use cases

**2.11.4. SOFTWARE QUALITY ATTRIBUTES**

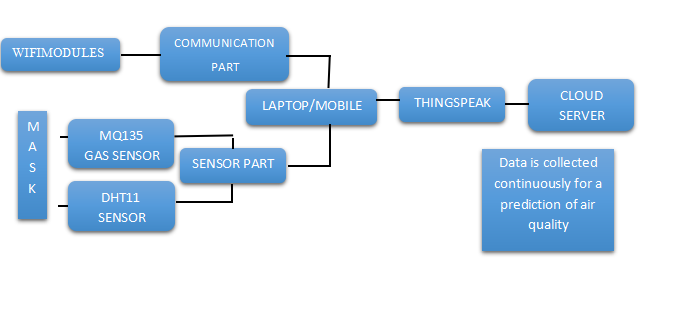
1. **Functionality:** The proper working of the sensor, NODEMCU ESP8266 and the entire circuit is necessary in determining the précised ppm values of different gases such as CO2,NH3,NO2,alcohol, benzene,smoke,CO2,Ammonia,Sulfide and other harmful gases which could further be used for data visualization and their comparisons.
2. **Maintainability:** The database must maintain the day to day ppm values of different gases and also must make appropriate comparisons. Recovery methods are necessary to recover from any failures by keeping the copy of the database that was backed up to archive storage.

**3.SYSTEM DESIGN**

**3.1. SYSTEM ARCHITECTURE**

The system/product under consideration consists of four sections.They are

* **Smart mask :** The smart mask is cable of detecting Air quality ,Temperature and humidity and these captured data would be used for visualization
* **Circuit Board :** The board used is Node MCU that would contain the code and perform the various processes needed for implementation of the software.The bread board would serve as a platform for providing connections to various sensors
* **Database:** The database that would be used is a Cloud based Data Visualization database that could store data like the ppm values,temperature values and humidity on a day to day basis.
* **Data Visualization tool :**The Smart mask would consist of proposed MQ135 gas sensor to detect the harmful gases inhaled and DHT11 temperature and humidity sensor.The Gas sensor and other sensors would be connected to the circuit board that consists of Node MCU,ESP2866 Wi-Fi module. The arduino board would be further connected to the database(Thingspeak) to store the information regarding the gases and their percentage.Thingspeak will be further connected to the SMART MASK app for data visualization (Numeric values and graphs) and notifications .



**Fig 3.1.SYSTEM ARCHITECTURE**

## 3.2. MODULE DIAGRAM/ BLOCK DIAGRAM/ FUNCTIONAL DIAGRAM

The project follows a divide and conquer approach wherein the overall problem is divided into more structure and manageable parts. Each one of the parts are developed separately.

**MASK**

**ESP8266 WIFI**

**MODULE**

**Node MCU**

**MQ3 SERIES**

**GAS SENSOR**

**DHT11 TEMPERATUREAND HUMIDITY SENSOR**

**DHT 11 TEMPERATURE AND HUMIDITY SENSOR**

**WIRELESS LAN**

**MACHINE**

**LEARNING**

**ALGORITHM**

**DATABASE**

**-+**

**THING**

**SPEAK**

**CLOUD**

**SMART MASK APP**

## 

**FIG 3.2.BLOCK DIAGRAM**

## 3.3. DATA FLOW DIAGRAM

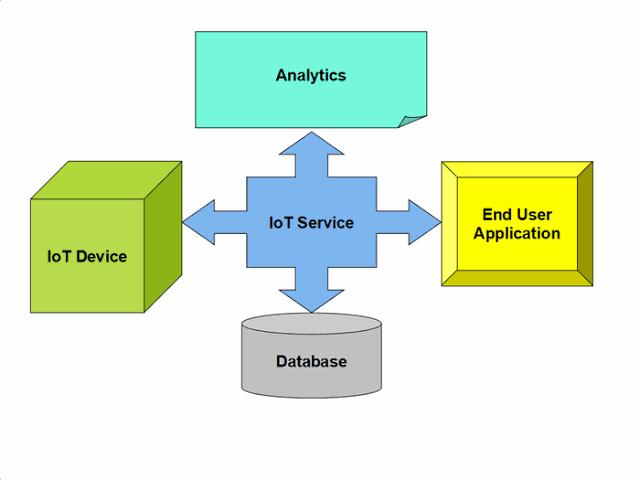
A Data Flow Diagrams is a structured analysis and design tool that can be used for creating the flow chart in place of, or in association with, information-oriented and process-oriented systems flowcharts. A DFD is a network that describes the flow of data and the processes that change, or transform, data throughout a system.

Table 1.3: DFD Diagram Symbols

|  |  |
| --- | --- |
| **Symbols** | **Description** |
|  | **Data Flow:** Data flow are pipelines through the packets of information flow. |
|  | **Process:** A Process or task performed by the system |
|  | **Entity:** Entity are object of the system. A source or destination data of a system |
|  | **Data Store:** A place where data to be stored |

**Context Level DFD – Level 0**

The context level data flow diagram (DFD) describes the whole system. The 0-level DFD describes all the sections which operate on the system.



**Fig 3.3.DFD of SMART MASK**

DATA FROM MQ135,DHT11 SENSOR

ARDUINO UNO R3

PPM VALUES AND DIFFERENT SENSOR VALUES

DATA SENT TO CLOUD THROUGH INTERNET

GETTING NORMALISE VALUES

**YES**

**NO**

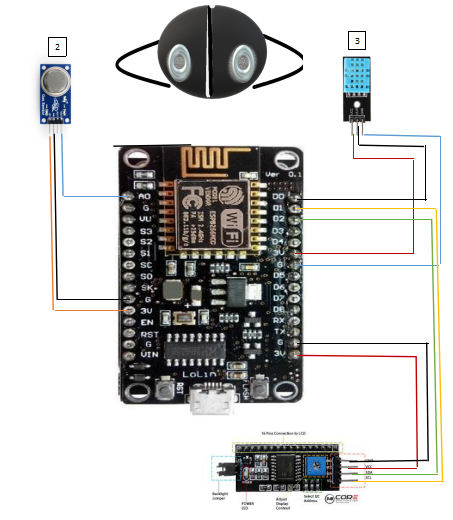
CHECK FOR ENVIRONMENTAL CONDITION (GOOD OR BAD AIR QUALITY)

DISPLAY ON THINGSPEAK WEBSITE

**Fig 3.4.DATA FLOW OF SMART MASK WORKING**

## 3.4. CIRCUIT DIAGRAM

A circuit diagram is a visual display of an electrical circuit using either basic images of parts or industry standard symbols. Symbol usage depends on the audience viewing the diagram.



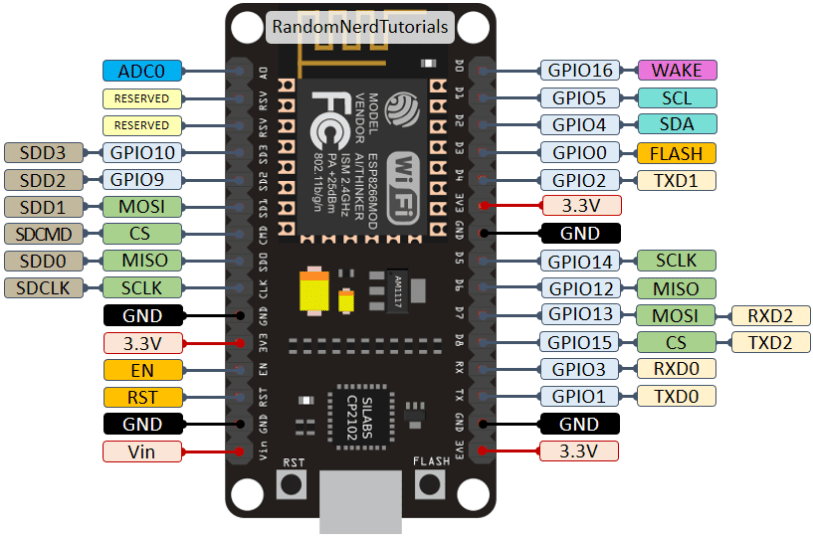
1

**FIG 3.5.CIRCUIT DIAGRAM**

**COMPONENTS:**

* Mask
* MQ135 Sensor
* DHT 11 Sensor
* LCD
* NODEMCU ESP8266
* Power Bank

**4 .PIN DESCRIPTION OF NODE MCU**



**FIG 3.6.PIN DESCRIPTION**

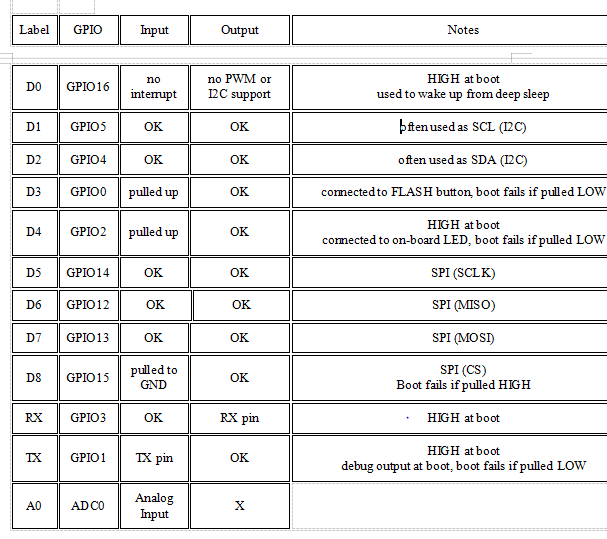
**PINS USED:**

1. MQ135 sensor and NodeMCU :

* A0 of NodeMCU to A0 of MQ135
* G of NodeMCU to Gnd of MQ135
* 3V3 of NodeMCU to Vcc of MQ135

1. DHT11 sensor and NodeMCU :

* D0 of NodeMCU to Data pin of DHT11
* G of NodeMCU to Gnd of DHT11
* 3V3 of NodeMCU to Vcc of DHT11



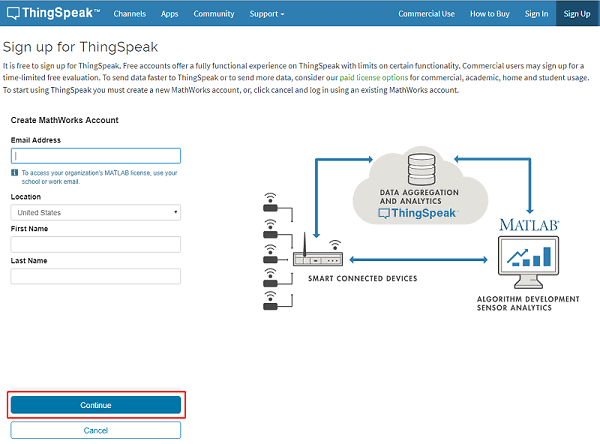
**FIG 3.7.PIN DESCRIPTION**

**3.6. INTERFACE DESIGN AND PROCEDURAL DESIGN**

**3.6.1. USER INTERFACE DESIGN**

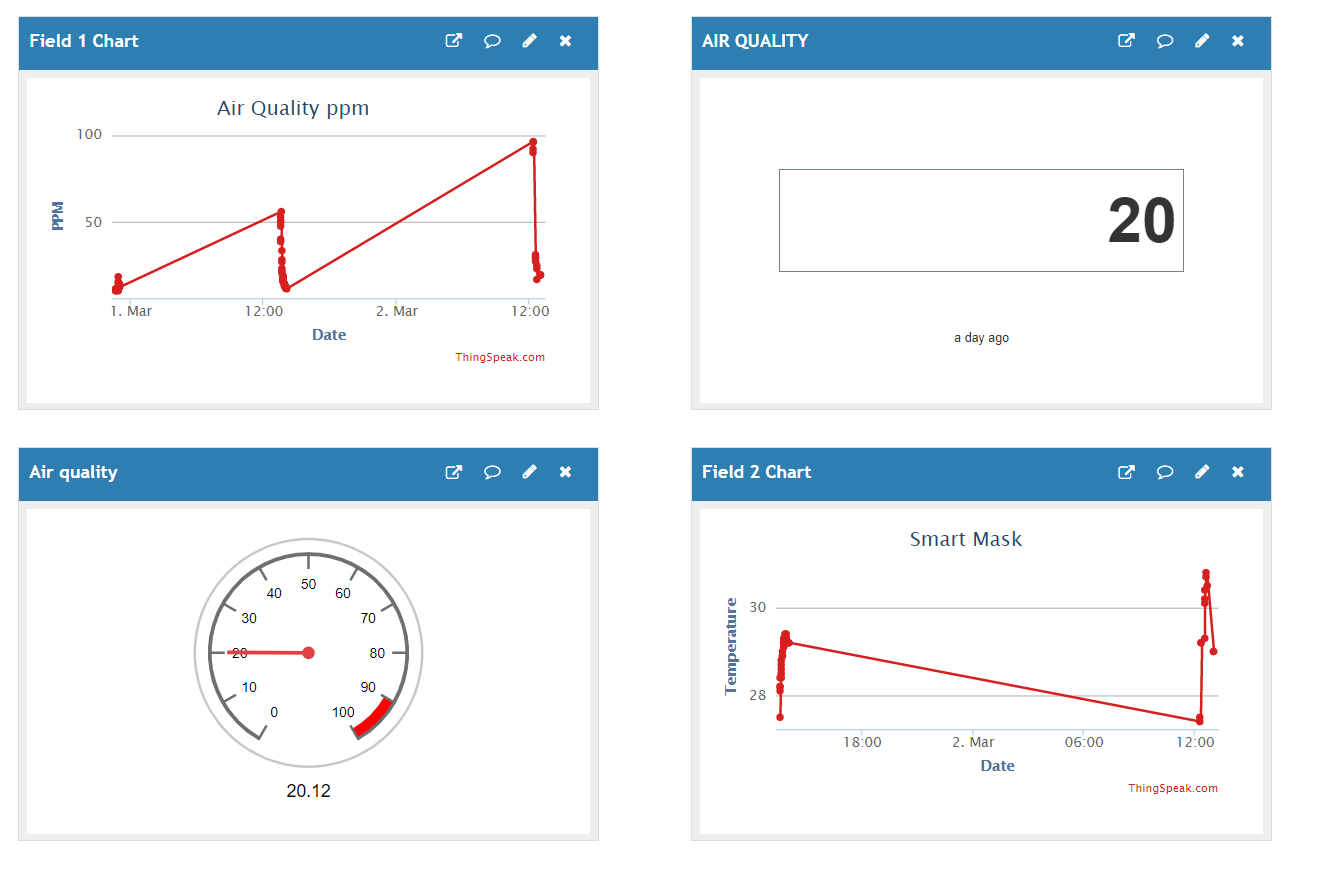
1. SIGN UP FOR THINGSPEAK :

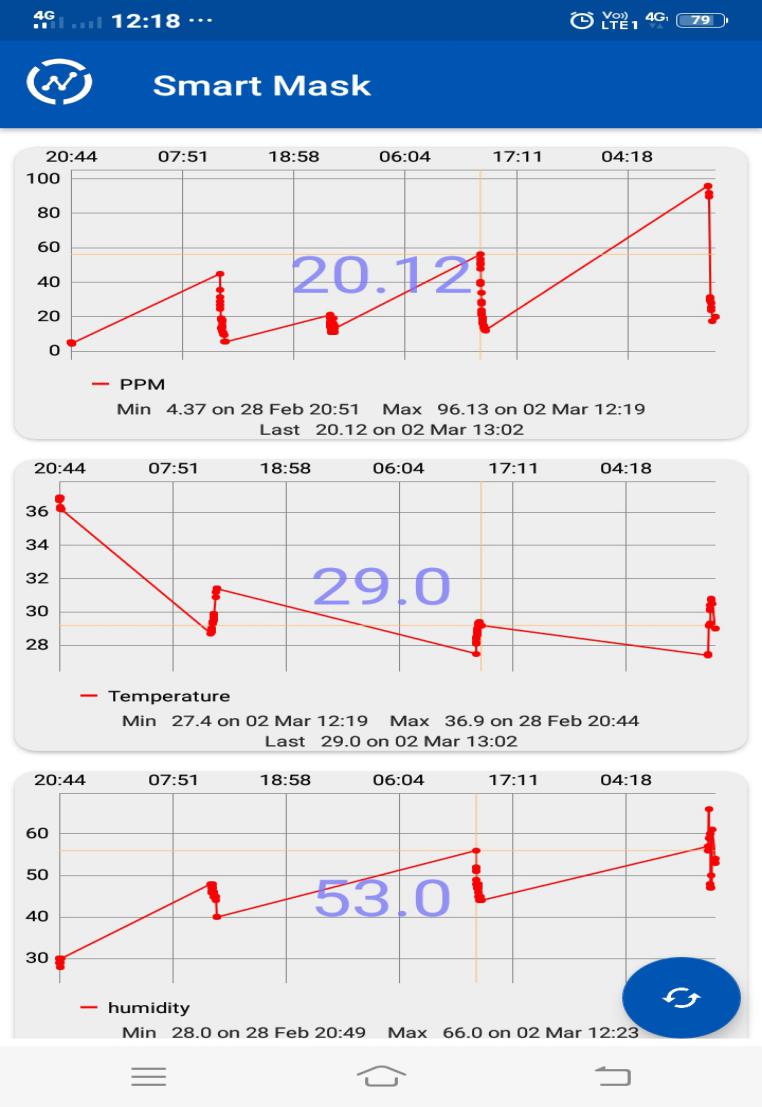
Sign In to ThingSpeak using your MathWorks  Account, or create a new MathWorks account.[6]



1. DATA VISUALIZATION

Thingspeak uses MATLAB Visualizations app to visualize data present in the channel. You can view and explore data using interactive visualizations such as an area plot, line plot, or scatter plot in static visualizations using other MATLAB plots.





**FIG 3.8.Graphs and Widgets ThingSpeak**

**3.6.2. CLASS DIAGRAM**

DHT11 SENSOR

-temperature:Int

-humidity:Int

\*getTemp()

\*get\_Humidity()

NodeMCU ESP8266

-result 1

-result 2

-result 3 \*get\_info()

\*send\_info()

MQ135 GAS SENSOR

-smoke\_level:Int

get\_smokelevel()

USER

-final\_results:Int

## 4. IMPLEMENTATION

**4.1. IMPLEMENTATION APPROACH**

It is important to ensure that how a particular functionality could be implemented. This way it could be ensured that IOT will improve your business in a way that will not only repay the investment but open the possibility for higher cost-efficiency of your existing service, product or production line.

### STEP 1: OBJECTIVES

* The problem statement : Air pollution is one of the serious problems faced by people all around the world .Especially in India, Delhi's pollution is in the state of crossing the severe category levels where the major cause as been urban air pollution and Indoor air pollution. People are still unaware of the various pollutants present around and various chronic ailments cause by them.
* By solving this we achieve
* In the short term :

This project aims at developing a SMART MASK which is IOT based that is capable of sensing the harmful gases present around that is inhaled through the mask, analyses it and also visualizes the amount of those gases present in terms of units and graphs.

-In the long term :

It aims at developing a fully fledged product that is capable of sensing air quality and helping users to decide whether the filter mask is necessary in that environment.

* What is the best way to solve it?

Since IOT is the current need of the hour and is making devices more smarter and easier to use, we feel this project would be a great contribution to the society. Even with respect to the current issue of Delhi pollution where the pollution level is in the state of crossing the severe category, has also been one of the motivating factor for us to take up this project

### STEP 2: TESTED IOT USE CASES

**Preventive maintenance** : Keeping track of the status of the amount of wear, damage of parts, oxidation as a way to prevent irreversible or expensive damage

**Environmental Monitoring :** Getting data on the environment, such as humidity, pollution, water quality, soil humidity, temperature and such

**Design for IoT :** A new product aimed at consumers, business or industries designed to have built-in integration to IoT networks

**Process Control & Optimization** : Getting data on process efficiency and applying it to increase efficiency

## STEP 3 : DECIDING ON CORRECT HARDWARE

This hardware consists in its most basic form of sensors, that is, devices used to provide data on air quality, temperature, humidity, etc. The data gathered by the sensors are then sent over the internet, to the Node MCU board where the data is stored there. Also the values sensed are sent to ThingSpeak which consists of a real time database and to Smart Mask app that helps in data visualization.

In general, the sensors need to be low energy devices in order to maintain operable for a long period without having to replace energy sources.

## STEP 4 : SELECTING IOT TOOLS

The internet is the foundation of IoT, therefore making sure proper internet connection with high speed is necessary. An IoT device is any device that is able to be connected to an internet network in order to :

* Gather sensor data, process it, and send it through the Internet to its designated endpoints
* Receive commands through the Internet in order to control actuators and do a task or a set of tasks

### STEP 5: SELECTING AN IOT PLATFORM

An IoT platform is the software used to control and centralize every aspect of the IoT network and its connected devices, including sending commands and gathering data, commonly by means of cloud integration .IoT platform defines how everything communicates and how data is handled. Changing it is a very costly process.On the other hand, as soon as it is ready and set up, implementing IoT itself gets much easier: it’s a matter of setting up connections and protocols for communication throughout the facility, following the platform’s guidelines.[7]

### STEP 6 : GATHER USEFUL DATA

The sensors MQ135 and DHT11 must be continuously kept in track as it is generating gigabytes of data every hour, every minute, or even every second, its usual tendency when it faces large volumes of data is that it discards everything as soon as it uses it. After all, you would need a lot of storage space to store all of that, which would add to the cost of implementing IoT.

Ideally, the data gathered should be processed, interpreted and sent to a database, where it would remain for a finite (or infinite) amount of time. From there, applications may be used to gather it, compile it and analyzed by data analysts, to ensure quality and efficiency of the processes, detect possible problems, and, of course, find ways to optimize them.

### STEP 7 : IMPLEMENT MACHINE LEARNING

Machine learning consists of an AI (that is, an extremely complex and adaptable group of algorithms) that is programmed to review data in real-time, identify patterns in them, and, depending on implementation, act on it which helps in analysing. A commercial ThingSpeak license also enables users to use MATLAB Toolboxes for machine learning, signal processing, system identification, and more with ThingSpeak, provided you have a license for the toolbox.

**4.2. CODING STANDARD**

* **C Language:**

The syntax is cluttered with punctuation, and there are a million different little mistakes you can make, but the language is still the first choice for many programmers who write for the lowest layer of software, the one closest to the hardware. The language hides nothing from you, and that means you can fiddle with every part of the code to squeeze out the best performance from an underpowered device. Every bit can be flipped. Every value on the stack is available. C remains the language of choice for constrained devices.

1. **Used for including library and header files of different sensors**

#include <ESP8266\_Lib.h> //esp8266 library

#include <DHT.h> //dht11 library

1. **WIFI credentials**

#define WIFI\_SSID "titu" // wifi name

#define WIFI\_PASSWORD "titu1998" // wifi password

1. **Thing Speak connection**

String apiKey = "MBQNXE14H4FDMN9Q"; Write API key here

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

client.print("Host: api.thingspeak.com\n");

client.print("Connection: close\n");

client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");

client.print("Content-Type: application/x-www-form-urlencoded\n");

client.print("Content-Length: ");

client.print(postStr.length());

1. **OTA(OverTheAir) connection**

ArduinoOTA.onStart([]() {

Serial.println("Start");

});

ArduinoOTA.onEnd([]() {

Serial.println("\nEnd");

});

ArduinoOTA.onProgress([](unsigned int progress, unsigned int total) {

Serial.printf("Progress: %u%%\r", (progress / (total / 100)));

});

ArduinoOTA.onError([](ota\_error\_t error) {

Serial.printf("Error[%u]: ", error);

if (error == OTA\_AUTH\_ERROR) Serial.println("Auth Failed");

else if (error == OTA\_BEGIN\_ERROR) Serial.println("Begin Failed");

else if (error == OTA\_CONNECT\_ERROR) Serial.println("Connect Failed");

else if (error == OTA\_RECEIVE\_ERROR) Serial.println("Receive Failed");

else if (error == OTA\_END\_ERROR) Serial.println("End Failed");

});

ArduinoOTA.begin();

**4.3. CODING DETAILS**

#include <ESP8266WiFi.h>

#include <DHT.h>

#include <ESP8266mDNS.h>

#include <WiFiUdp.h>

#include <ArduinoOTA.h>

#include "MQ135.h"

#include <BlynkSimpleEsp8266.h>

char auth[] = "t5GWG\_hZKygPN8tCL9zegjsVPv5tuA3l";

String apiKey = "MBQNXE14H4FDMN9Q"; // Write API key here

const char \*ssid = "vivo 1908" ; // Enter your WiFi Name,,,,,7u'

const char \*password = "elsajerlyn"; // Enter your WiFi Password

const char\* server = "api.thingspeak.com";

const int sensorPin= 4;

int air\_quality;

#define DHTPIN 0

DHT dht(DHTPIN, DHT11);

WiFiClient client;

// Replace with your network credentials

const int ESP\_BUILTIN\_LED = 2;

void setup() {

Serial.begin(115200);

Serial.println("Booting");

WiFi.mode(WIFI\_STA);

WiFi.begin(ssid, password);

Blynk.begin(auth, ssid, password);

while (WiFi.waitForConnectResult() != WL\_CONNECTED) {

Serial.println("Connection Failed! Rebooting...");

delay(5000);

ESP.restart();

}

// Port defaults to 8266

// ArduinoOTA.setPort(8266);

// Hostname defaults to esp8266-[ChipID]

// ArduinoOTA.setHostname("myesp8266");

// No authentication by default

// ArduinoOTA.setPassword((const char \*)"123");

ArduinoOTA.onStart([]() {

Serial.println("Start");

});

ArduinoOTA.onEnd([]() {

Serial.println("\nEnd");

});

ArduinoOTA.onProgress([](unsigned int progress, unsigned int total) {

Serial.printf("Progress: %u%%\r", (progress / (total / 100)));

});

ArduinoOTA.onError([](ota\_error\_t error) {

Serial.printf("Error[%u]: ", error);

if (error == OTA\_AUTH\_ERROR) Serial.println("Auth Failed");

else if (error == OTA\_BEGIN\_ERROR) Serial.println("Begin Failed");

else if (error == OTA\_CONNECT\_ERROR) Serial.println("Connect Failed");

else if (error == OTA\_RECEIVE\_ERROR) Serial.println("Receive Failed");

else if (error == OTA\_END\_ERROR) Serial.println("End Failed");

});

ArduinoOTA.begin();

Serial.println("Ready");

Serial.print("IP address: ");

Serial.println(WiFi.localIP());

pinMode(ESP\_BUILTIN\_LED, OUTPUT);

}

void loop() {

ArduinoOTA.handle();

//float h = dht.readHumidity();

float t = dht.readTemperature();

MQ135 gasSensor = MQ135(A0);

air\_quality = gasSensor.getPPM();

float RL = 10;

float R0 = 76.63;

float Rs,ratio,CO2\_PPM;

dht.begin();

if (client.connect(server,80))

{

String postStr = apiKey;

postStr +="&field1=";

postStr += String(air\_quality);

postStr +="&field2=";

postStr += String(t);

postStr +="&field3=";

postStr += String(h);

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

client.print("Host: api.thingspeak.com\n");

client.print("Connection: close\n");

client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");

client.print("Content-Type: application/x-www-form-urlencoded\n");

client.print("Content-Length: ");

client.print(postStr.length());

client.print("\n\n");

Serial.print("Air Quality: ");

Serial.print(air\_quality);

client.print(postStr);

Rs=( ( 5.0 \* RL ) - ( RL \* air\_quality ) );

ratio=Rs/R0;

ratio=ratio\*0.3611;

CO2\_PPM = 14.15\*2.868-ratio;

Serial.print("CO2:");

Serial.print(CO2\_PPM);

Rs=( ( 5.0 \* RL ) - ( RL \* air\_quality ) );

ratio=Rs/R0;

ratio=ratio\*0.015 ;

CO = 14.15\*2.868-ratio;

Serial.print("CO:");

Serial.print(CO);

if(air\_quality<15)

{

Serial.print("Air quality is good");

}

else if(air\_quality<50)

{

Serial.print("Health alert");

}

else if (air\_quality<150)

{

Serial.print("Not suitable environment");

}

else{

Serial.print("Dangerous condition");

}

Serial.print("Temp: ");

Serial.print(t);

Serial.print(" degrees Celcius, Humidity: ");

Serial.print(h);

Serial.println("%. Send to Thingspeak.");

}

client.stop();

Serial.println("Waiting...");

Blynk.run(); //Run the Blynk Read temperature and put it in t variable

float q = analogRead(A0);

Blynk.virtualWrite(V0, t); //Send t value to blynk in

Blynk.virtualWrite(V5, q);

if(q<15)

{

Blynk.notify("Air quality is good");

}

else if(q<50)

{

Blynk.notify("Health alert");

}

else if (51<q && q<150)

{

Blynk.notify("Not suitable environment");

}

else{

Blynk.notify("Dangerous condition");

}

if (t>40) {

Blynk.notify("The Temperature is too high"); //Blynk will send notify if t>40

}

delay(1000);

}

**4.4. SCREENSHOTS**

**SMART MASK APP**

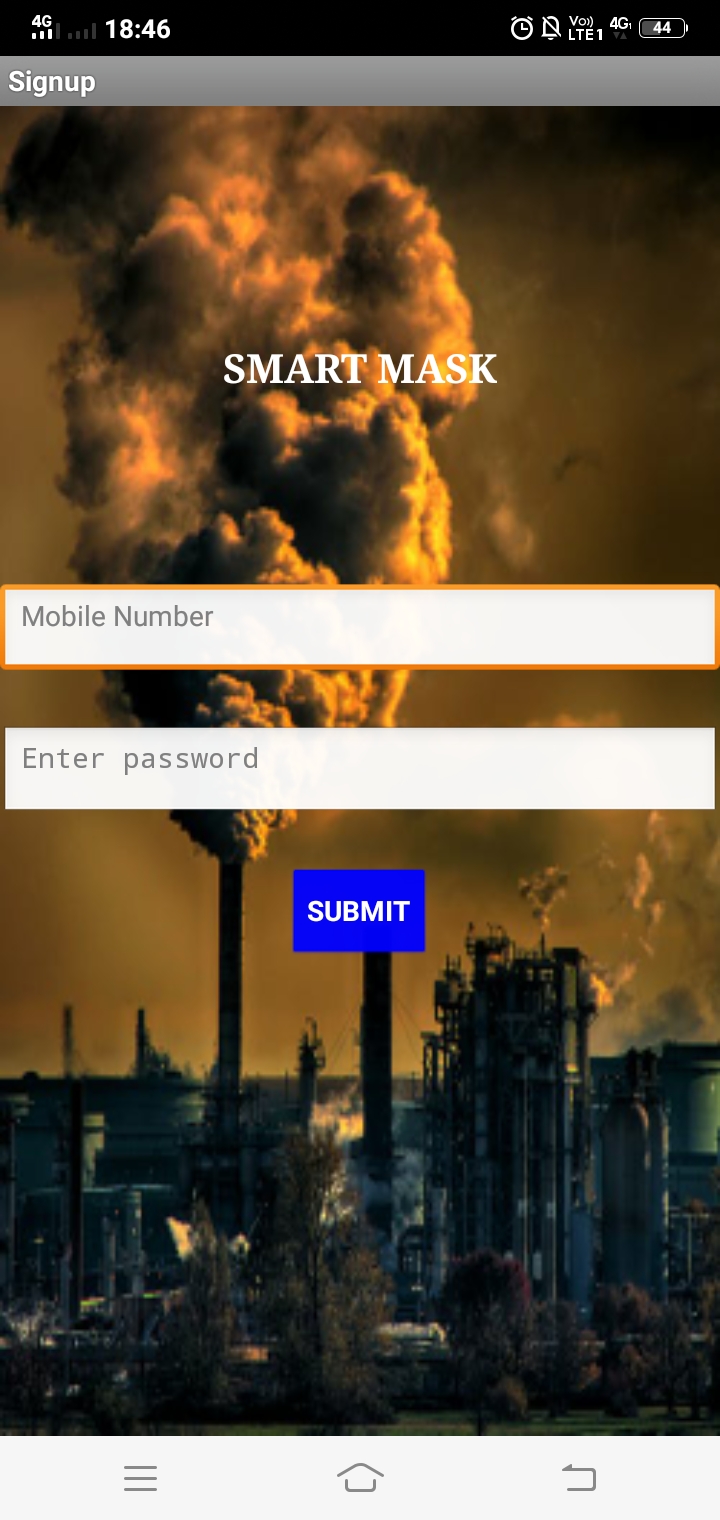
1. Home page:

The User Homepage module lets users choose a specific page of the site as their homepage. Users with a homepage will be redirected to this page upon successful login on the site.

****

**FIG 4.1.Home Page**

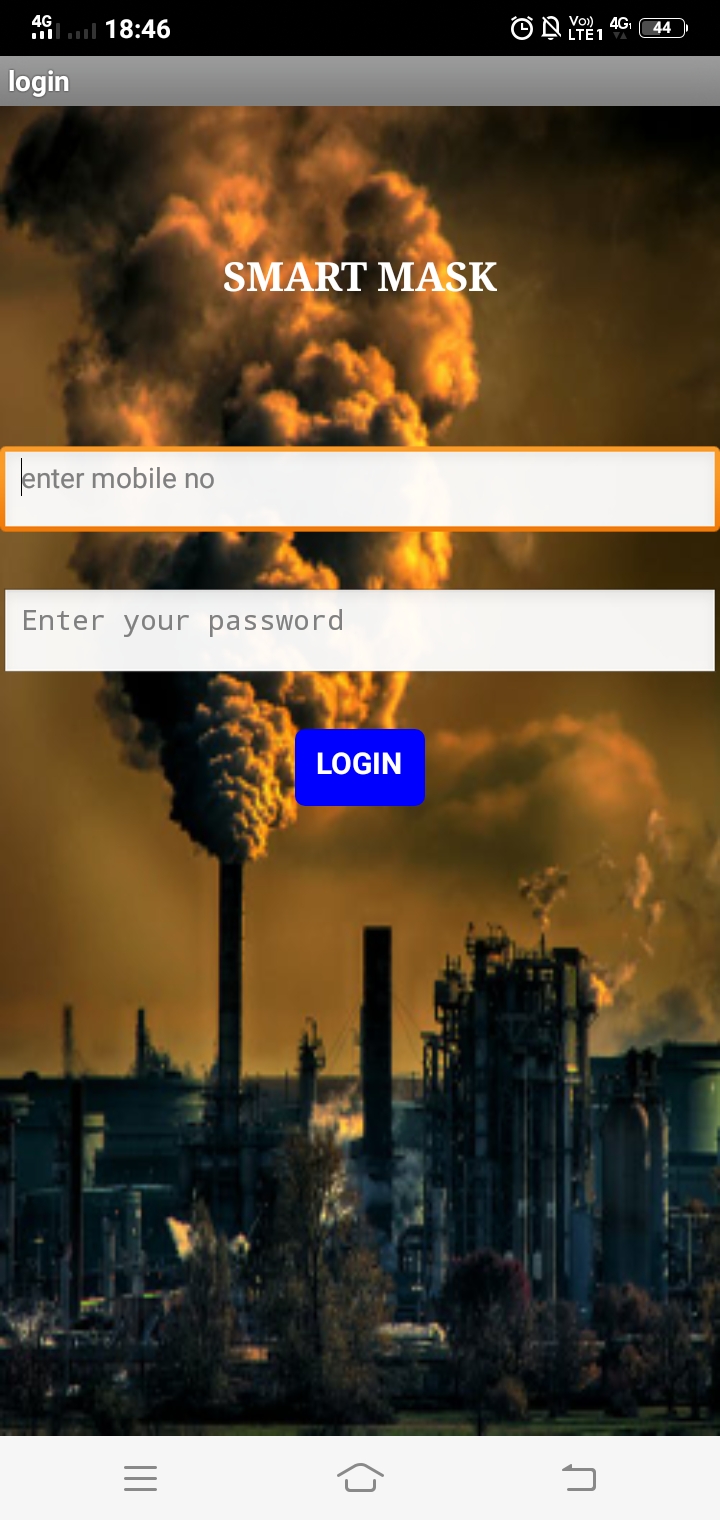
2.SignUp page :

****

**FIG 4.2.SignUp Page**

3.Login page :

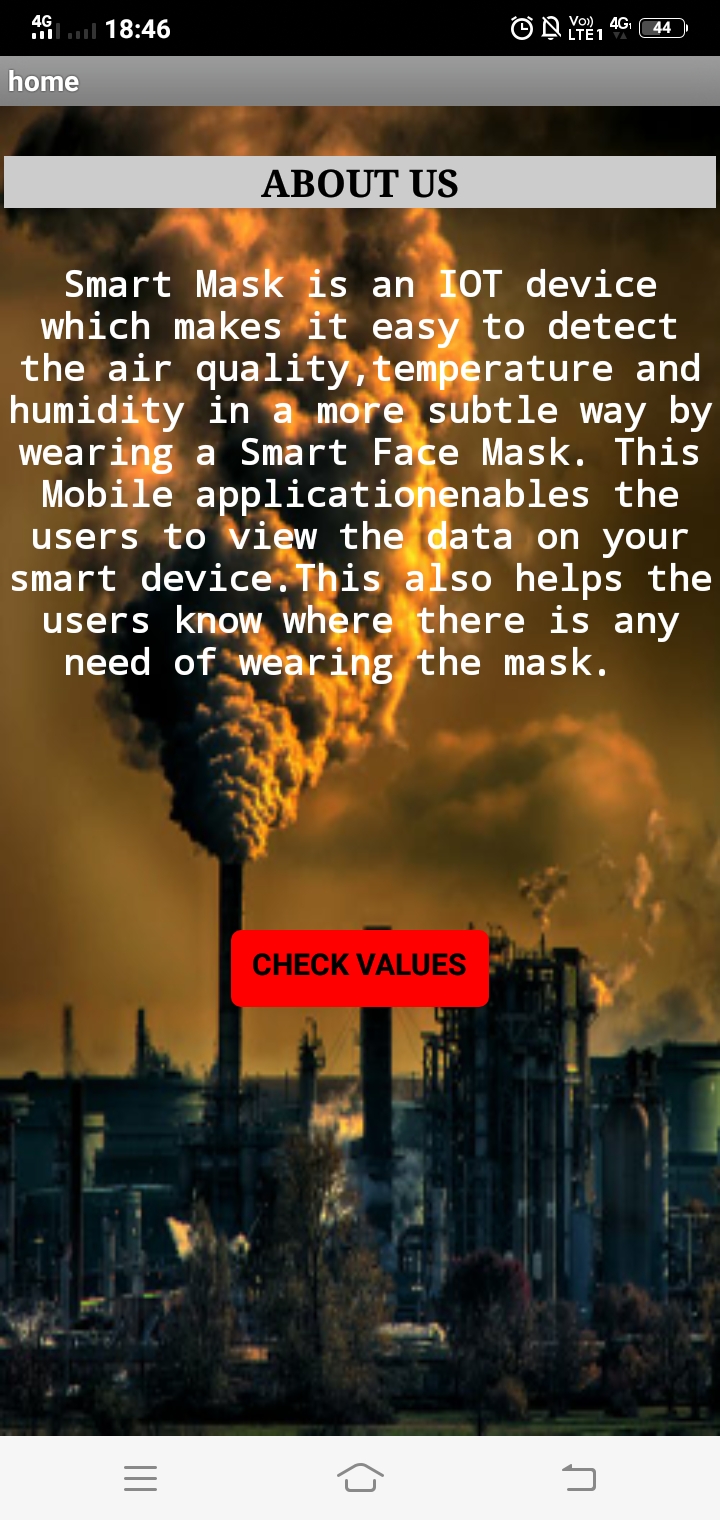
The Login Module is a portal module that allows users to type a user name and password to log in. You can add this module on any module tab to allow users to log in to the system. More on creating module tabs.

****

**FIG 4.3.Login Page**

1. About Us page :

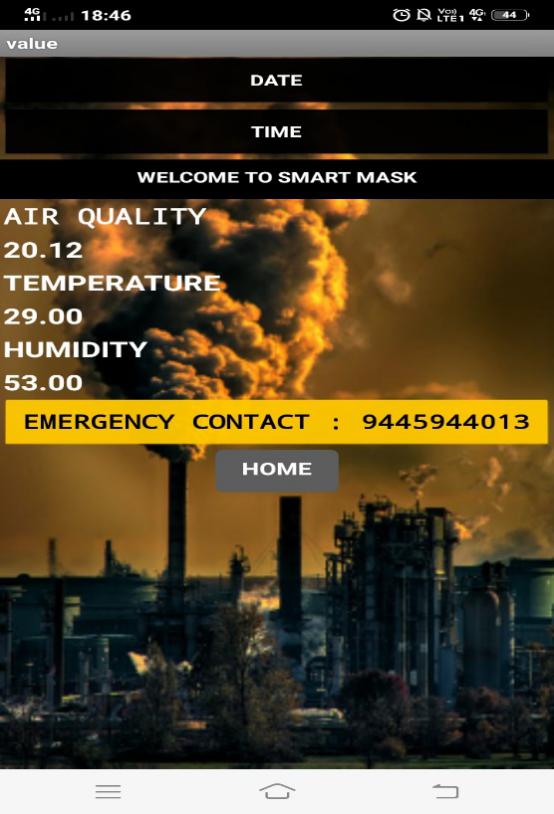
About Us page gives the overall description of what exactly the app or a website is about.

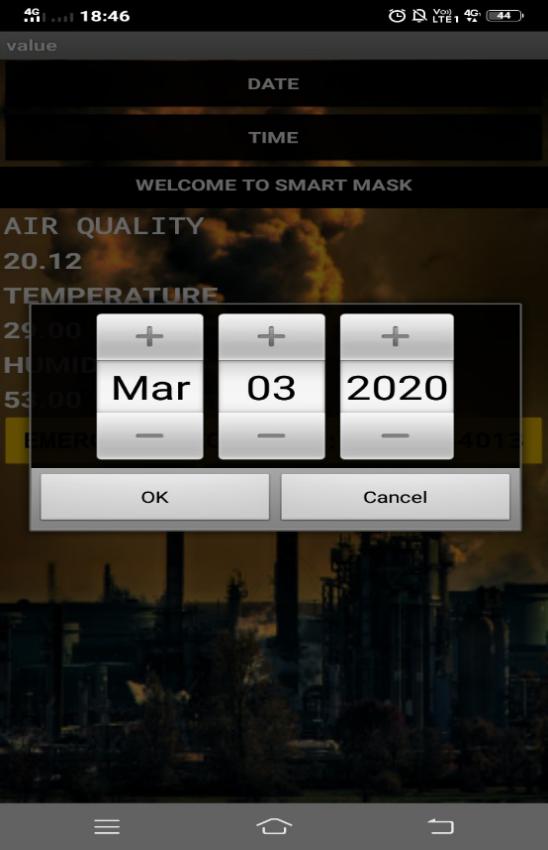
****

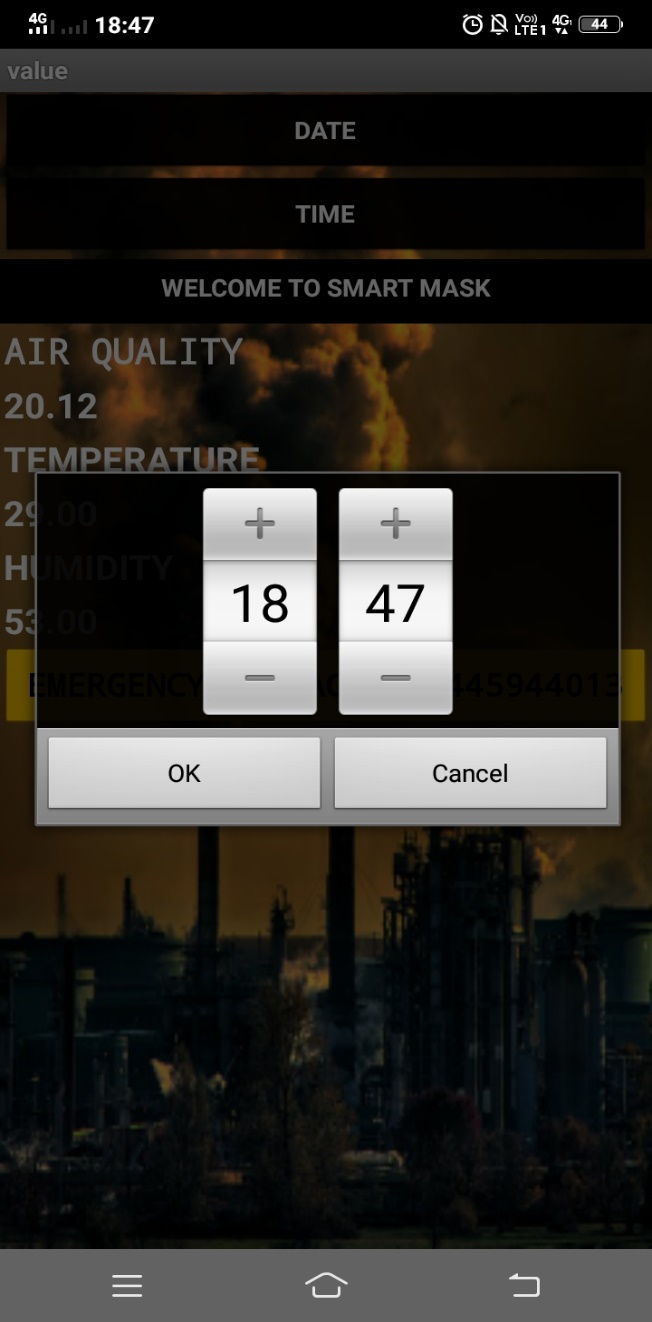
**FIG 4.4.About Us Page**

1. Info page :

Info page of SMART MASK app gives the values of Air quality,Temperature and Humidity that is detected through the SMART MASK

****

****

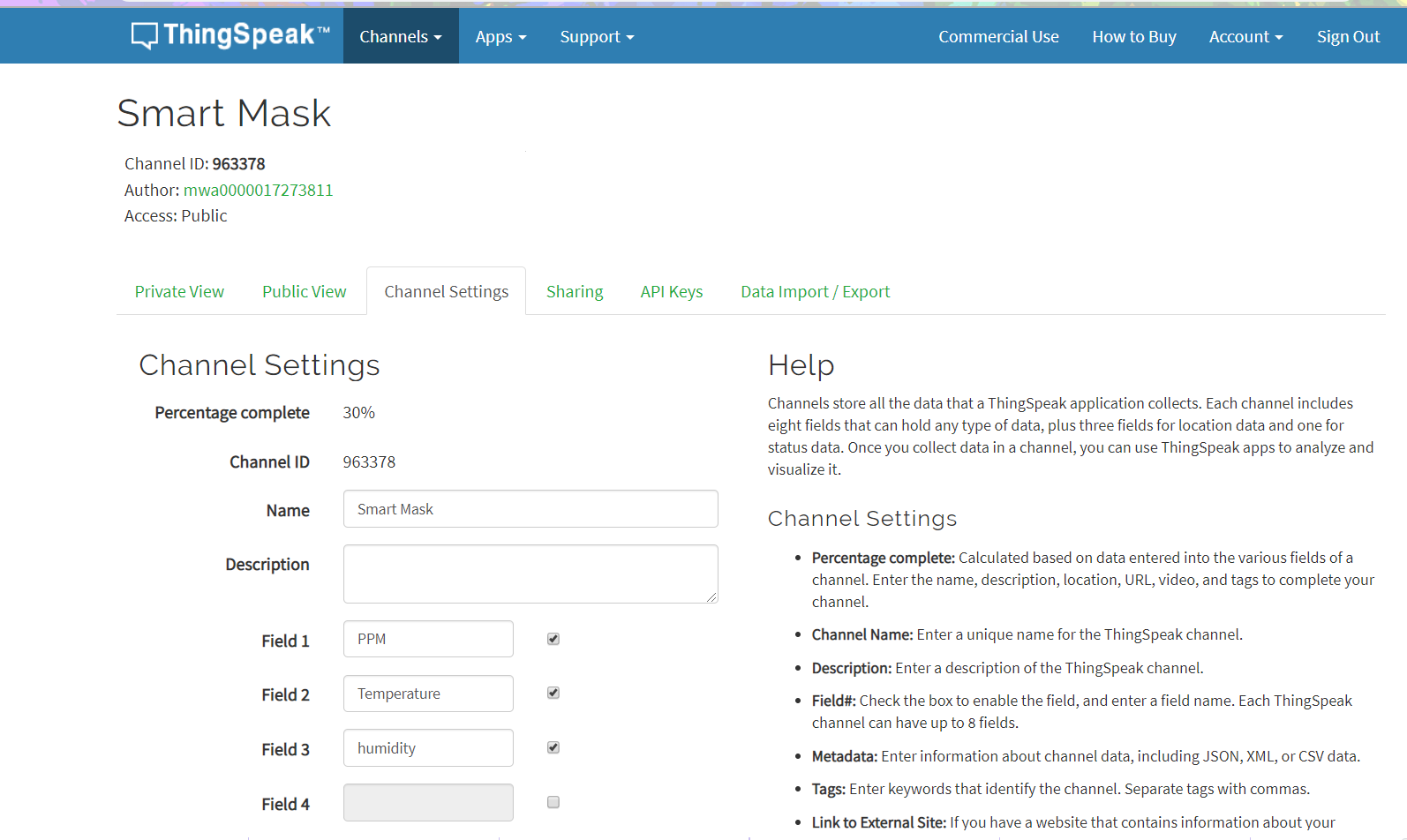
****

**FIG 4.5.Values Page**

**THINGSPEAK**

1. Channel settings :

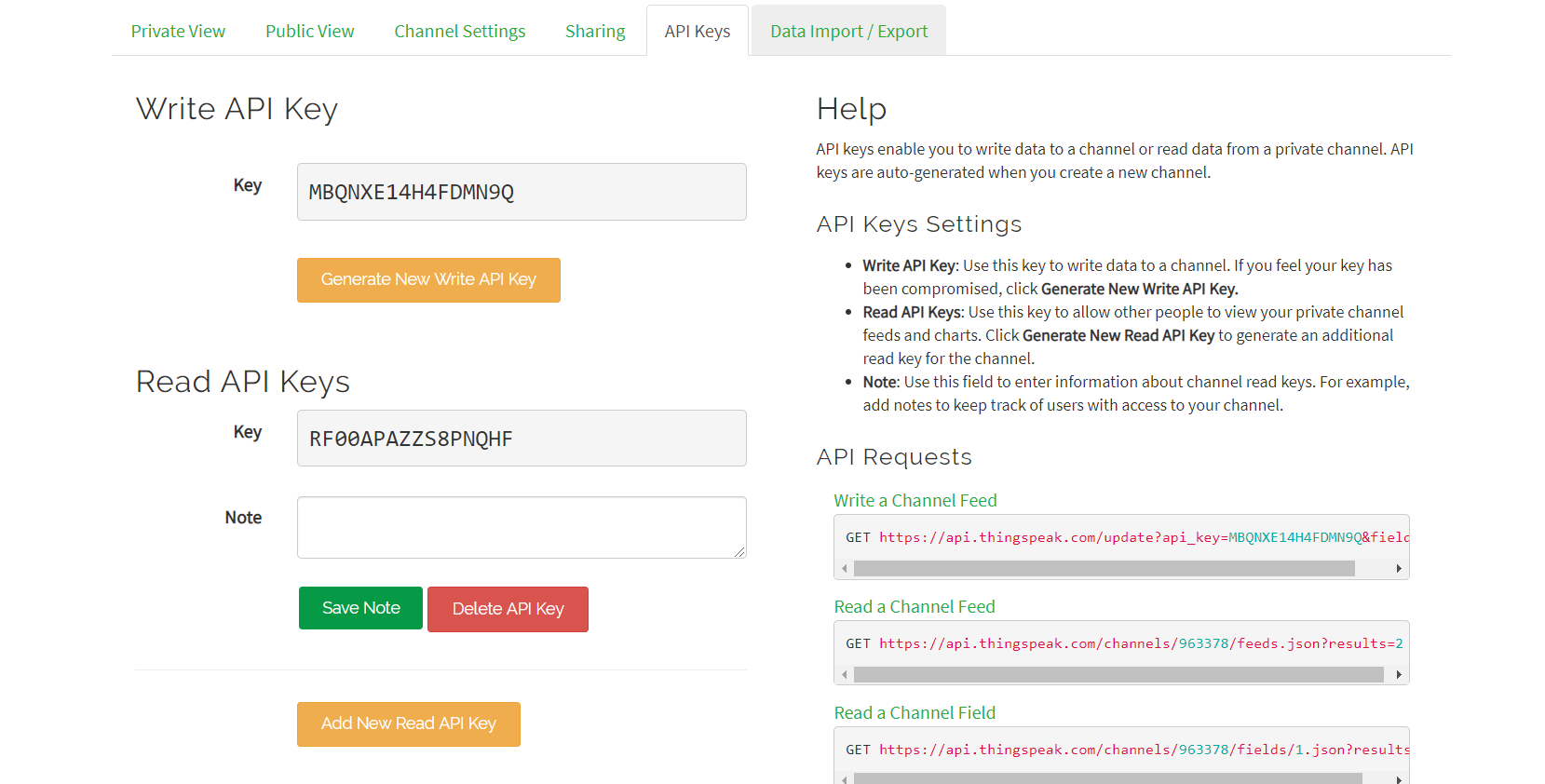
ThingSpeak Channels. Once you have signed up and signed into ThingSpeak, you start by creating a ThingSpeak Channel. A channel is where you send your data to store. Each channel includes 8 fields for any type of data, 3 location fields, and 1 status field.



**FIG 4.6.Channel Settings**

1. API Keys :

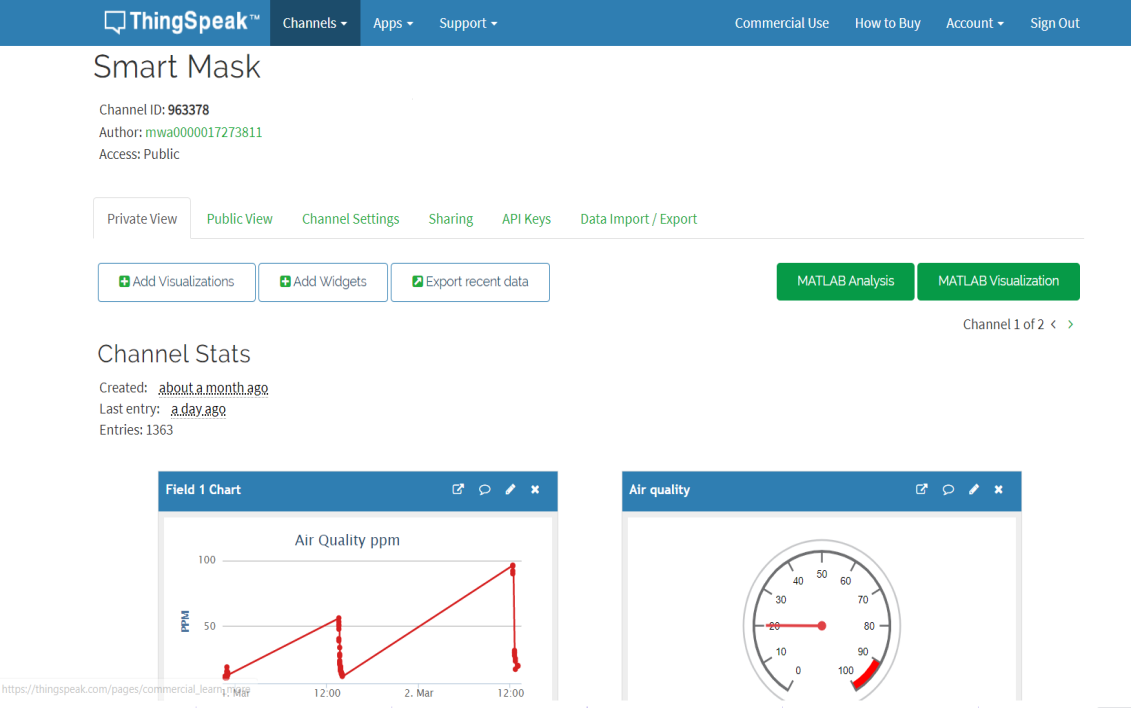
When you read or write data to your channel using the ThingSpeak API or MATLAB code, you need the appropriate read and write permissions.The 16-digit API key allows you to read from a private channel and write to a channel. You do not need an API key to read from a public ThingSpeak channel.



**FIG 4.7.API Keys Description**

1. Data visualization :

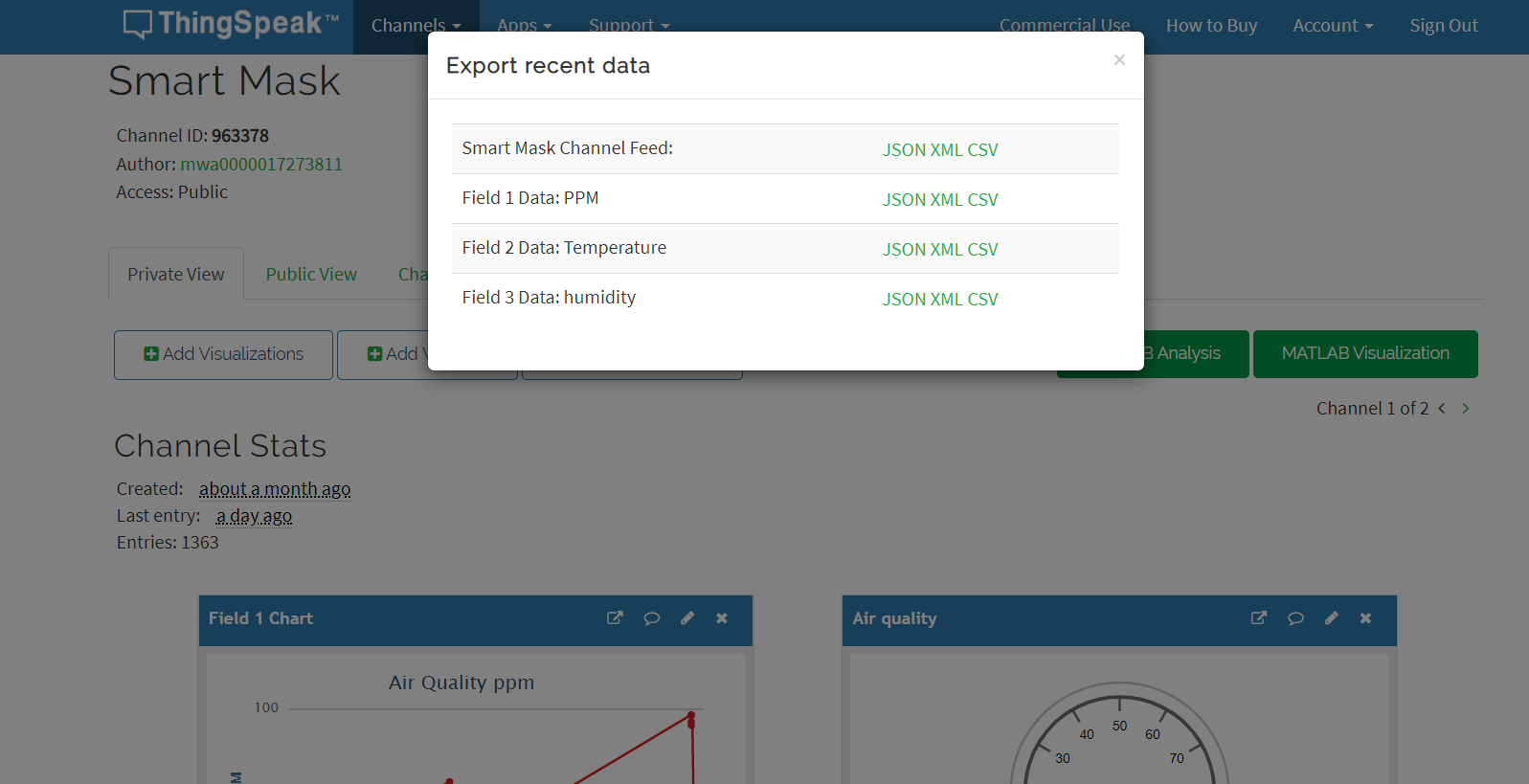
Thingspeak uses MATLAB Visualizations app to visualize data present in the channel. You can view and explore data using interactive visualizations such as an area plot, line plot, or scatter plot in static visualizations using other MATLAB plots.



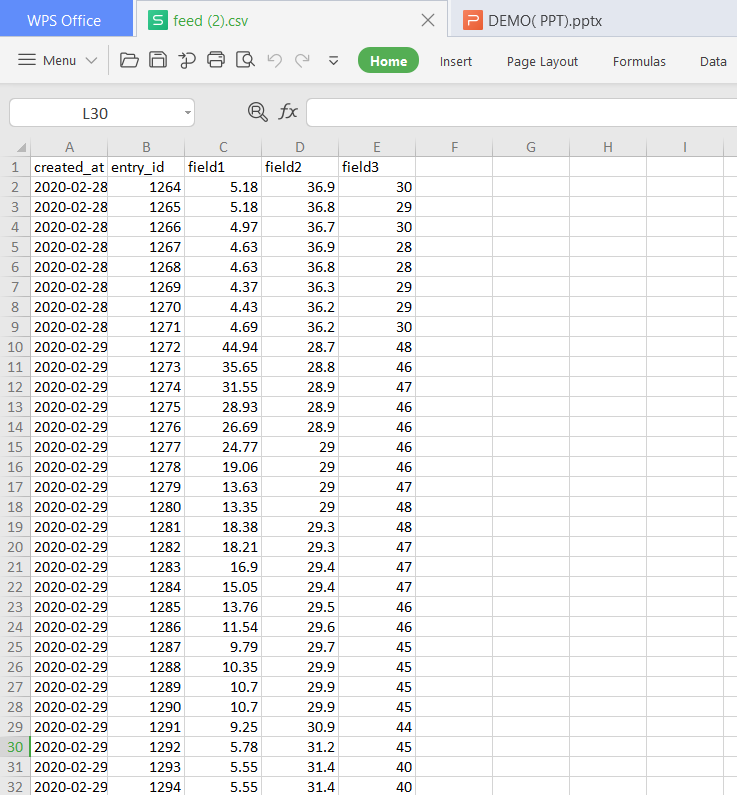
4. Exporting data in different formats (JSON,XML and CSV) :

To export all data from a ThingSpeak channel:

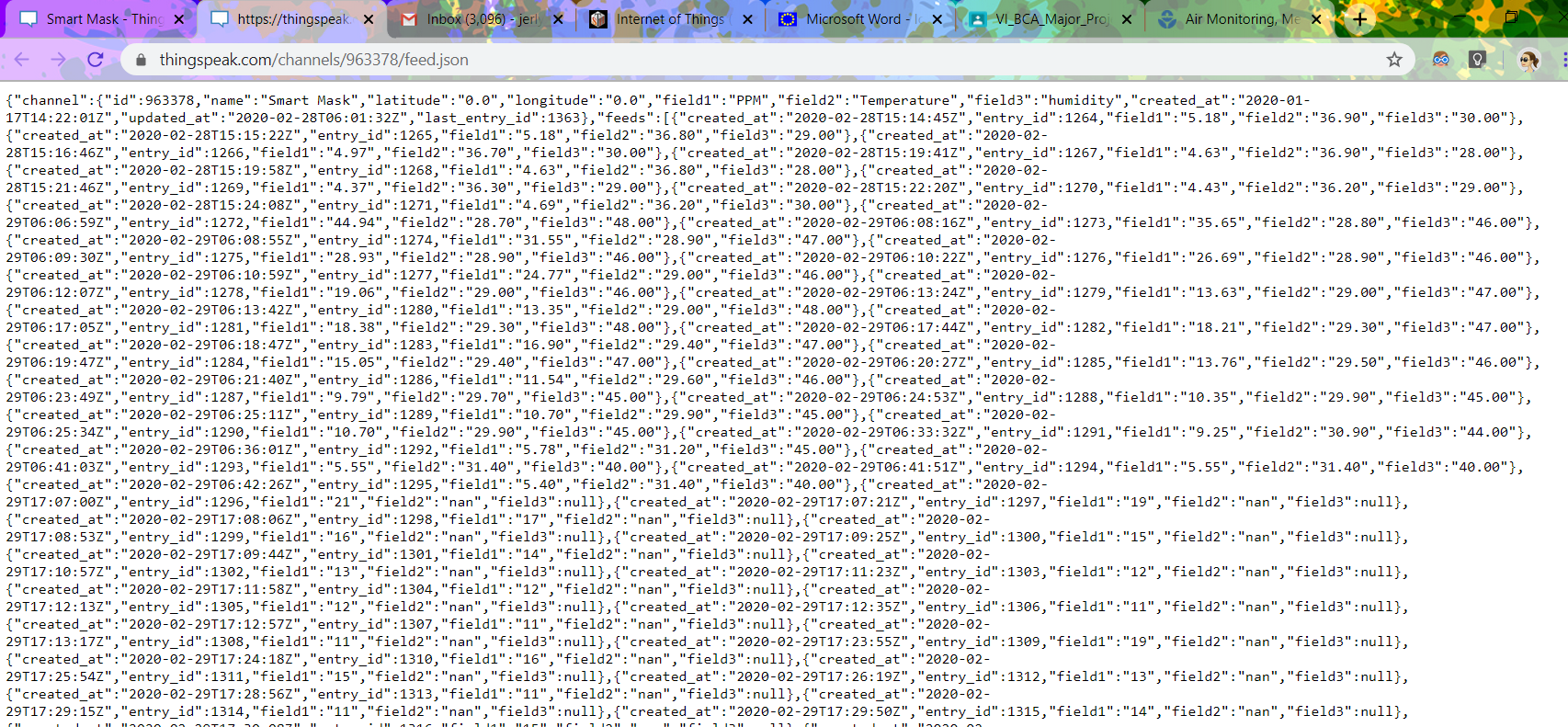
* Select Channels > My Channels.
* Select the channel.
* Select the Data Import / Export tab.
* Use the default or choose a time zone for the export.
* Click Download.



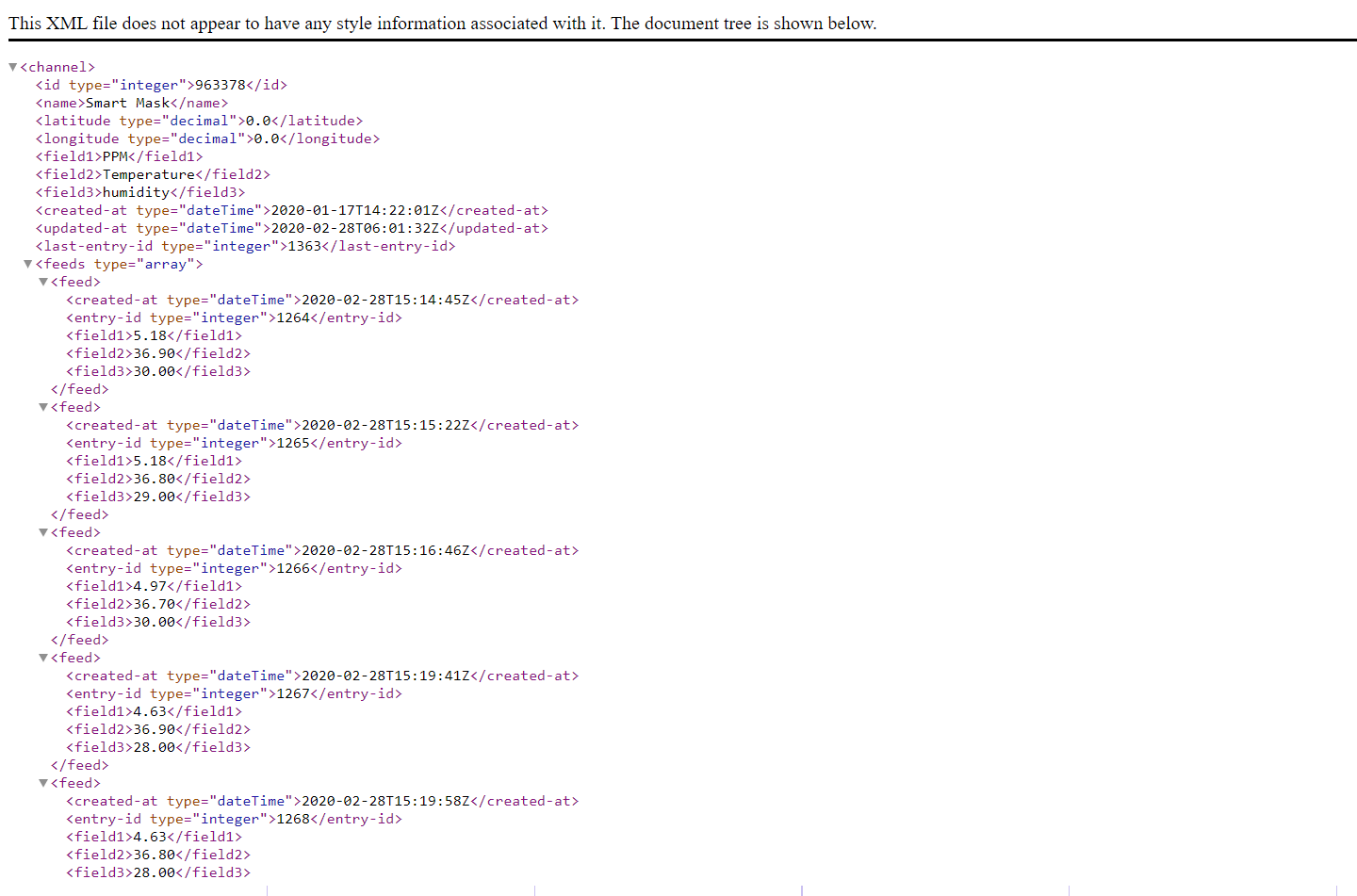
**FIG 4.8.Export Format**



**FIG 4.9.CSV Format**



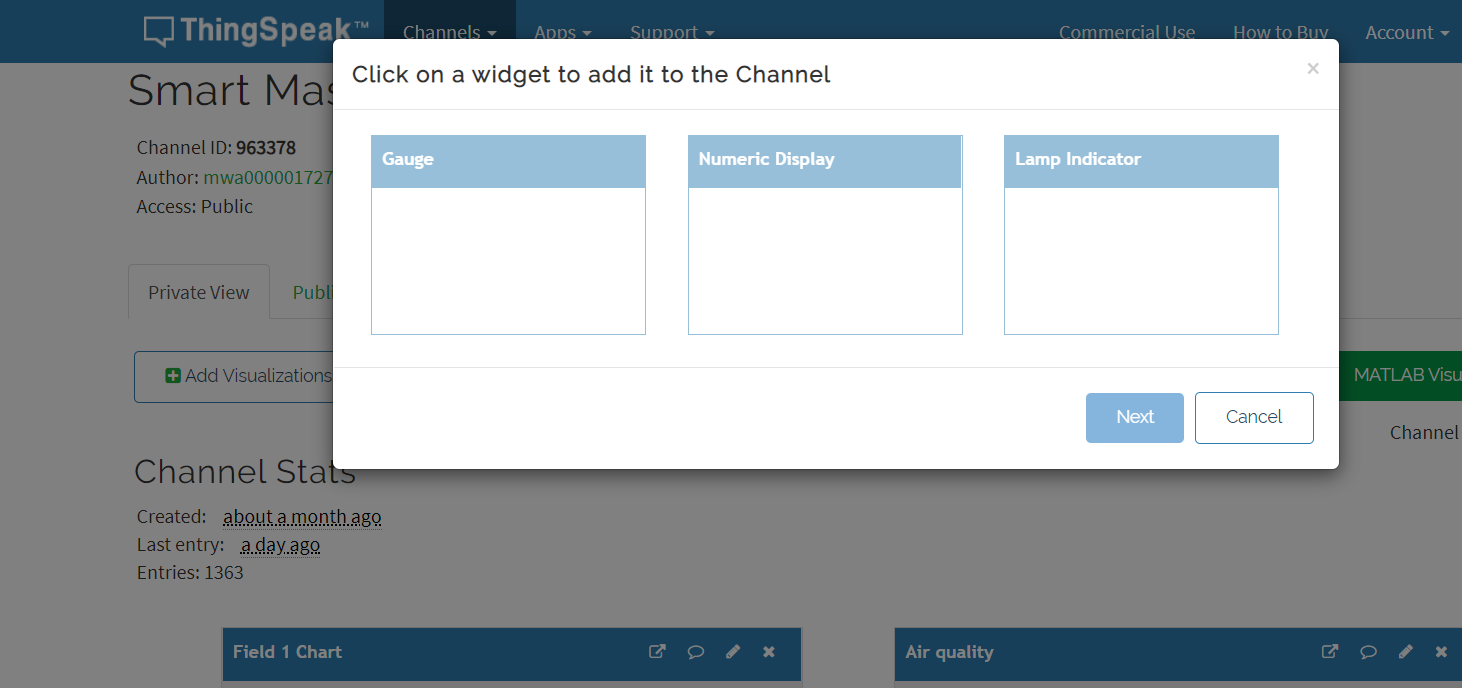
**FIG 4.10.JSON Format**



**FIG 4.11.XML Format**

5.Widgets :

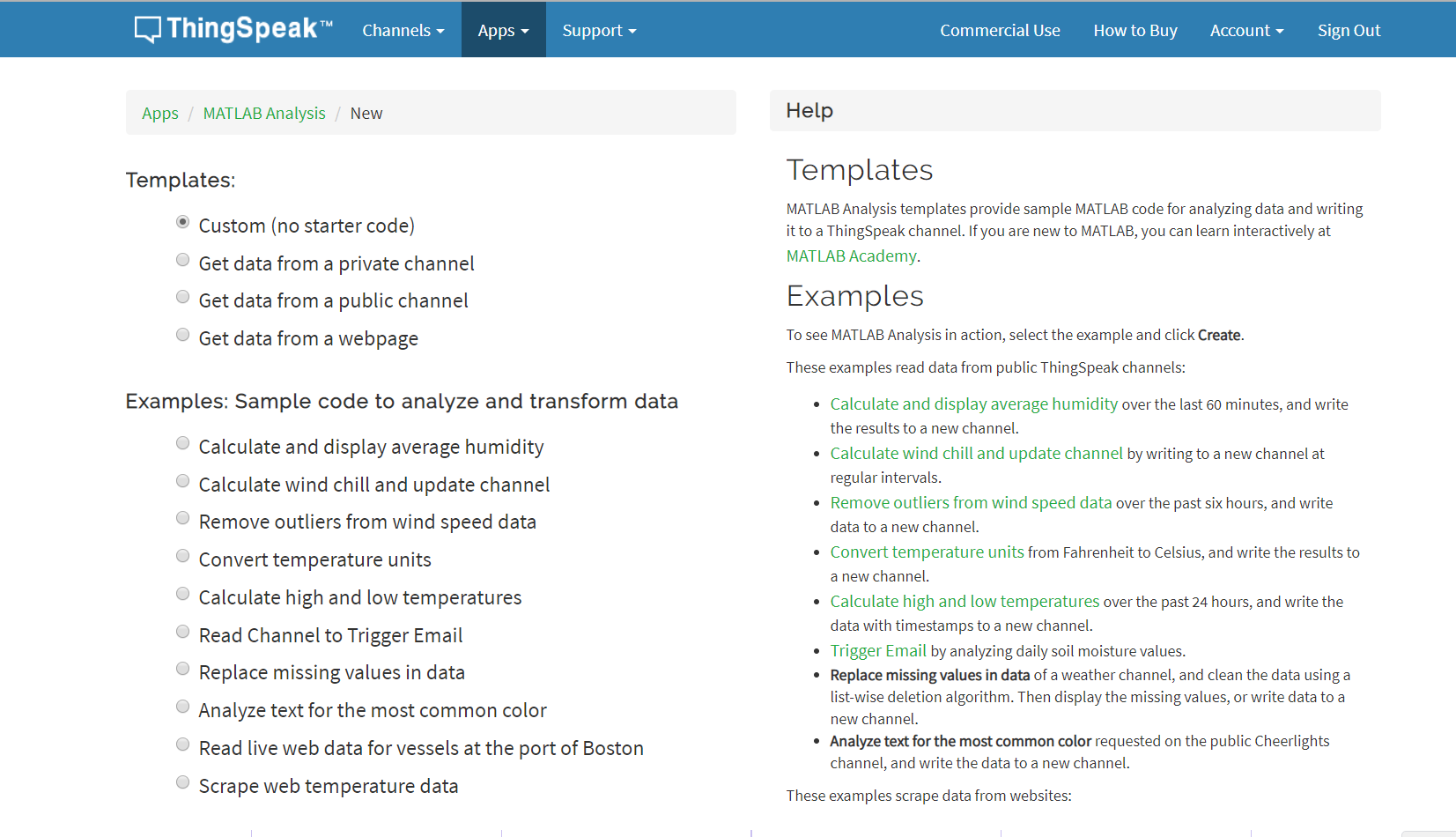
* ThingSpeak Monitor Widget will help the user to be always up to date with current state of your Things connected to ThingSpeak service.
* Every instance of the widget will display last actual Field value of your Channel.
* The values will be refreshed automatically every 30 mins. The user can refresh them immediately tapping any time at a widget center .
* Your Channel and Field name will be got from ThingSpeak. Just set your channel's ID!



**FIG 4.12.WidgetsThingSpeak**

6.MATLAB analysis :

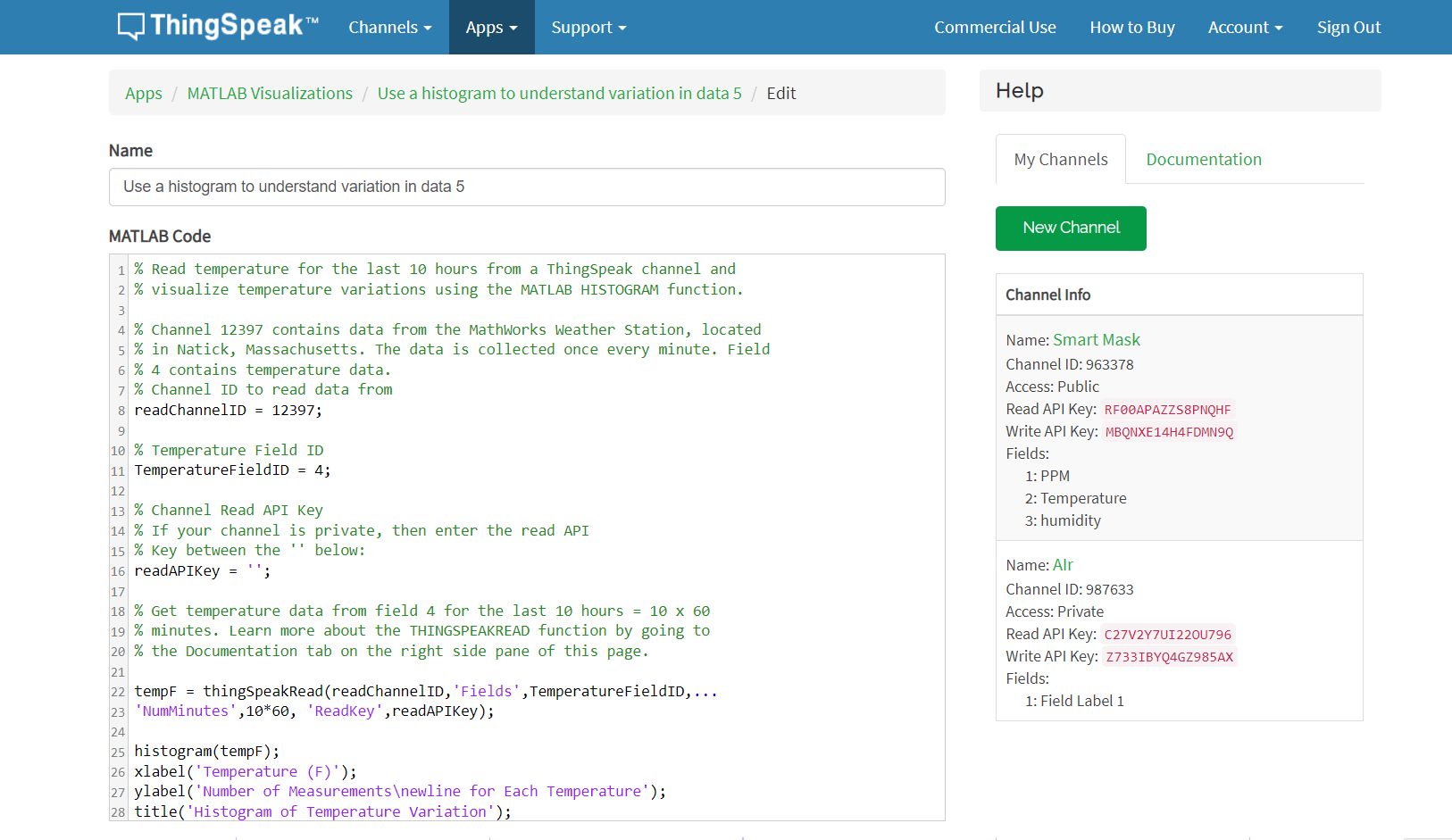
Data collection in the cloud with advanceddata analysis using MATLAB.Thingspeak is an IOT analytics service that allows you to aggregate, visualize, and analyze live data streams in the cloud.With the ability to execute MATLAB code in ThingSpeak, you can perform online analysis and process data as it comes in.



**FIG 4.13.Matlab Analysis ThingSpeak**

7.MATLAB visualization :

ThingSpeak allows users to use MATLAB to analyze and visualize your data. The analysis and visualization apps provide template code to assist with basic operations on historic or live data.

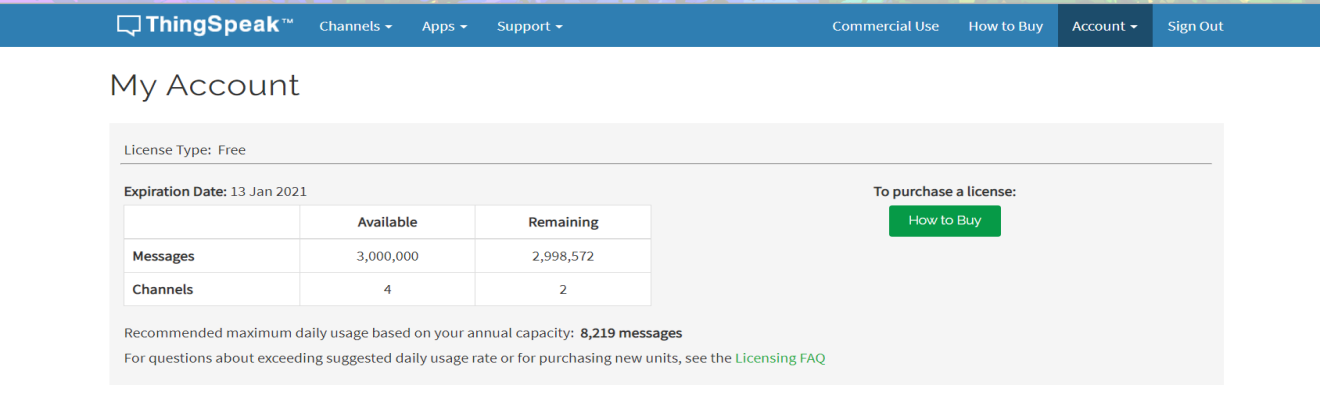


**FIG 4.14.Matlab Visualization ThingSpeak**

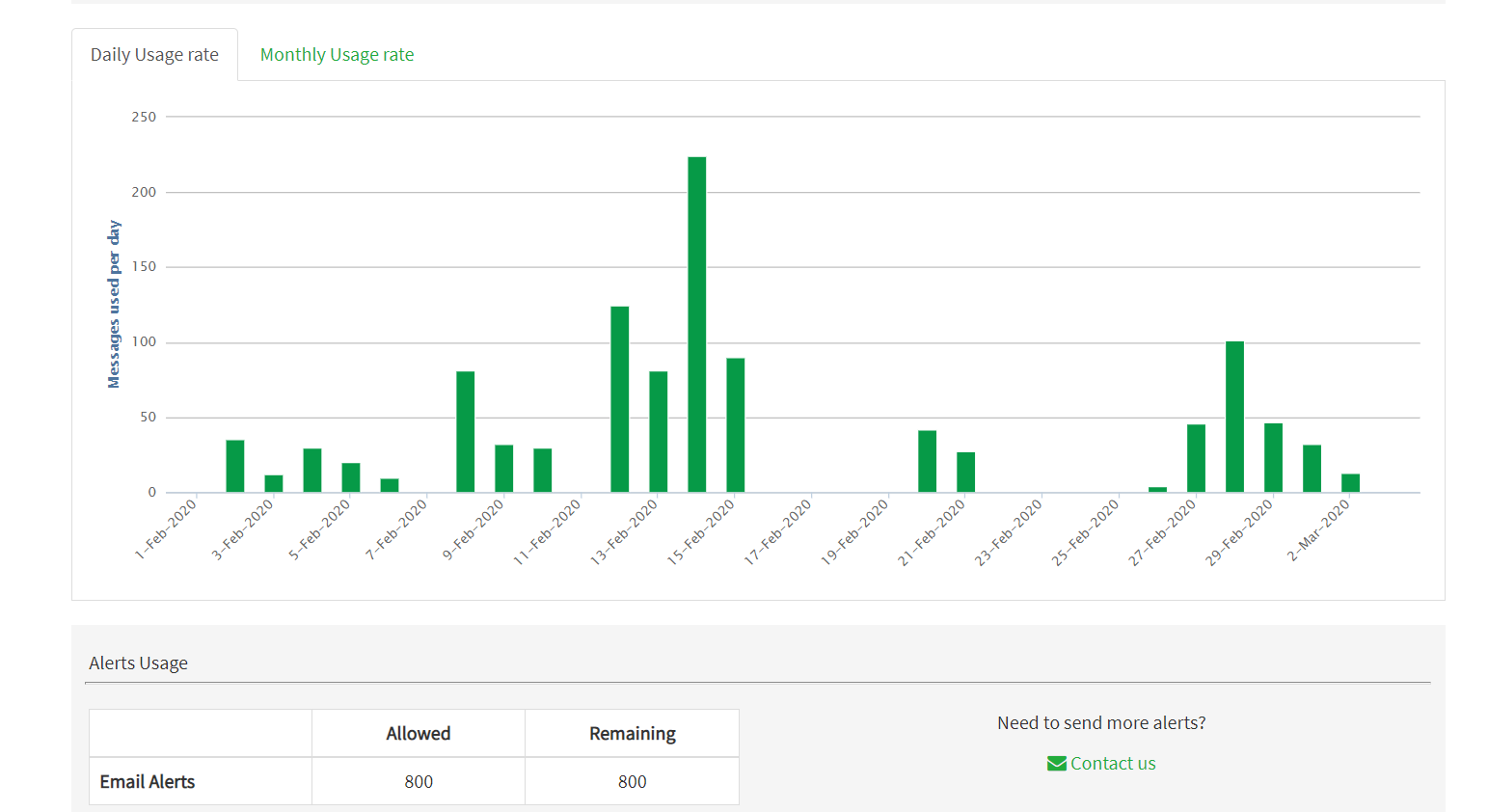
8.My Account :

ThingSpeak enables users to schedule and run MATLAB code using a version of MATLAB that MathWorks hosts in the cloud. Users do not need a MATLAB license to run MATLAB within the ThingSpeak web service.

If user have a MATLAB license, they can also import your data from ThingSpeak into your desktop MATLAB session for offline analysis using the [ThingSpeak Support Toolbox](http://www.mathworks.com/matlabcentral/fileexchange/52244-thingspeak-support-toolbox" \t "https://thingspeak.com/pages/_blank).



**FIG 4.15.MyAccount ThingSpeak**

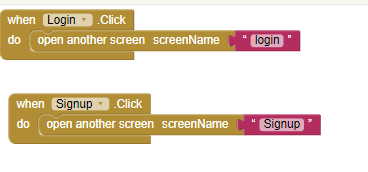


**FIG 4.16.MyAccount ThingSpeak**

**MIT APP INVENTOR (BLOCK DIAGRAMS : CODE PART)**

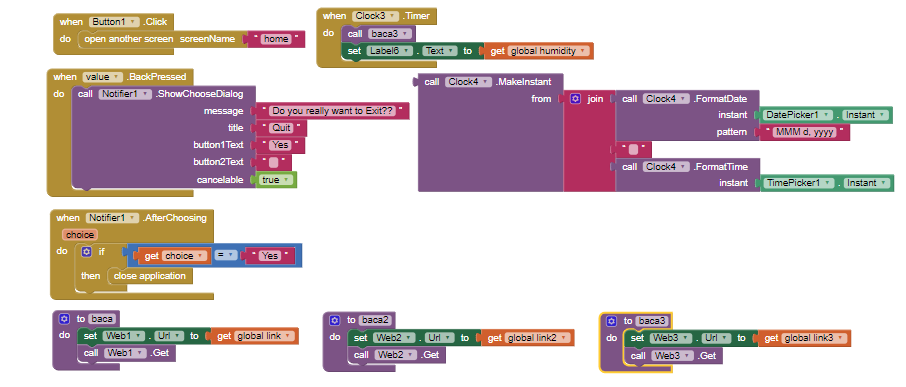
**MIT App Inventor** is an intuitive, visual programming environment that allows everyone – even children – to build fully functional apps for smartphones and tablets. Those new to MIT App Inventor can have a simple first app up and running in less than 30 minutes. And what's more, our blocks-based tool facilitates the creation of complex, high-impact apps in significantly less time than traditional programming environments.

1.Main page

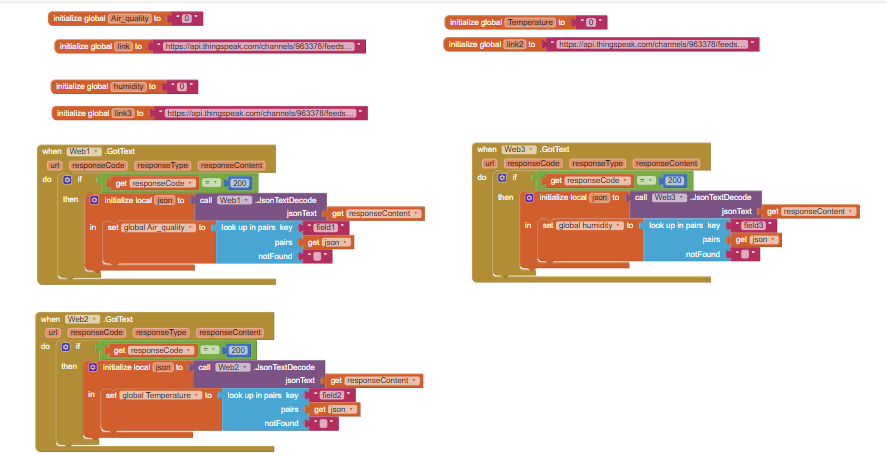


**FIG 4.17.Main Page Code Block**

2.Info page

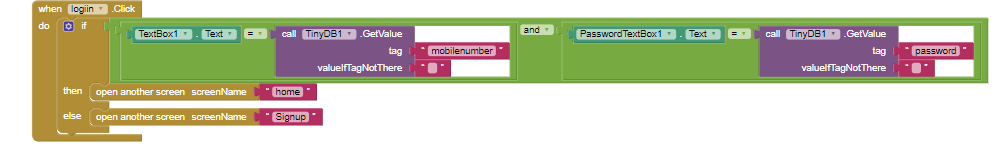


**FIG 4.18.Info page Code Block**



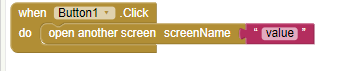
**FIG 4.19.Info page Code Block**

3.Login page



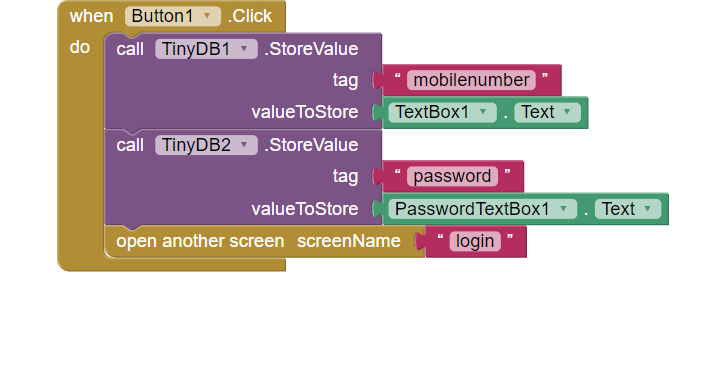
**FIG 4.20.Login page Code Block**

4.Home page



**FIG 4.21.Home page Code Block**

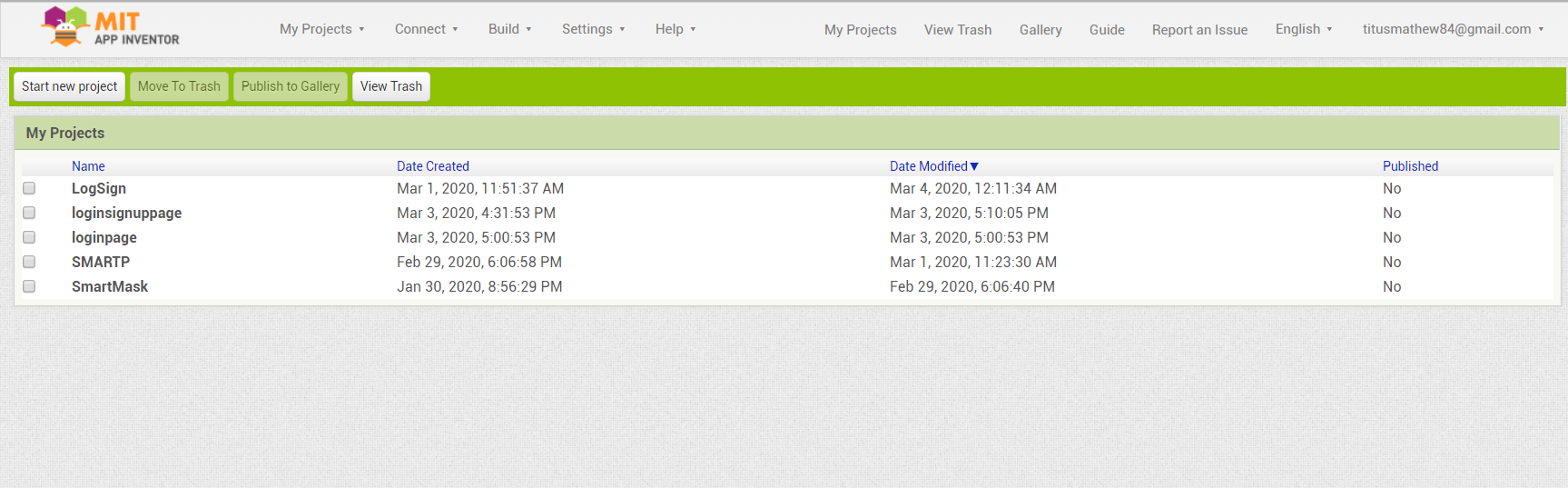
5.SignUp page



**FIG 4.22.SignUp Code Block**

6. MIT App inventor (Home page)

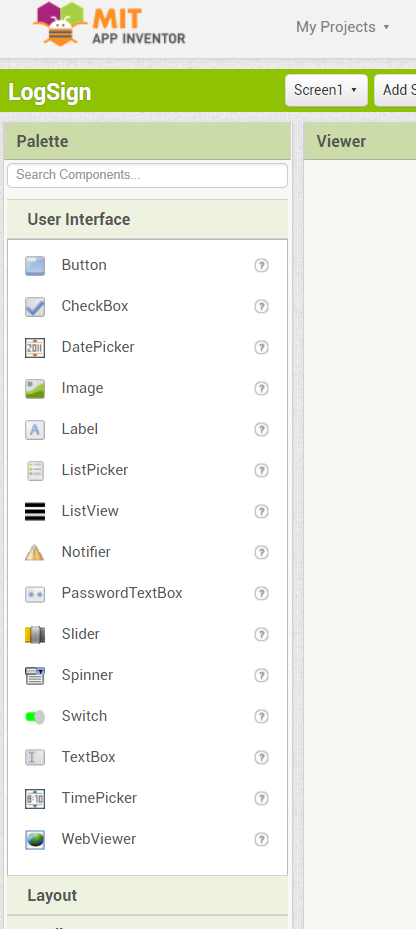
My projects



**FIG 4.23.My Projects**

1. Palette :

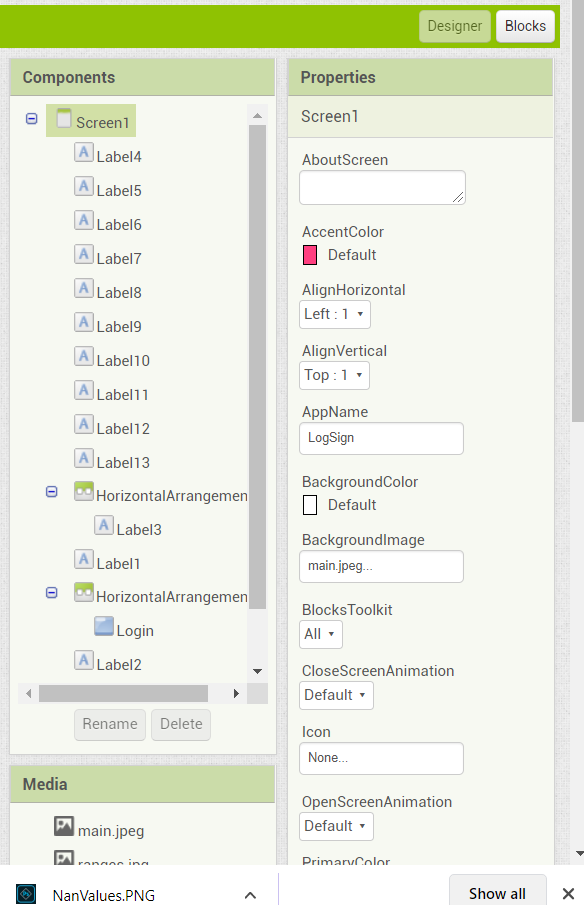
The Palette is where you add all the elements that make the app perform any number of tasks. The components are grouped by functionality, making it easy to find the right component.



**FIG 4.24.User Interface components**

8.Options available with different items on palette:

* This palette contains different drawers such as Basic, Media, Animation, etc. which hold all of the components. These components can be clicked on and dragged onto the screen
* There are two main types of components: visible and non-visible. Visible components such as Button, TextBox, Label, etc. are part of the User Interface. Non-visible components such as Accelerometer, Sound, OrientationSensor are not seen and thus not a part of the UI screen, but they provide access to built-in functions of the Android device.
* When programming in the Blocks Editor, components can be accessed in two ways. Specifically by using the set Label1.Text to or generally by using the set any Label.Text to. If specifically only changing the one label, only that label will get replaced. By using any component block, all components of the type Label will have their Text changed to the user's specification.



**FIG 4.25.Palette Description**

**5.TESTING**

**5.1. TEST CASES**

Cases: Test cases are all about how are we going to test a requirement.Firsly we have to define the various test scenarios present in the project.

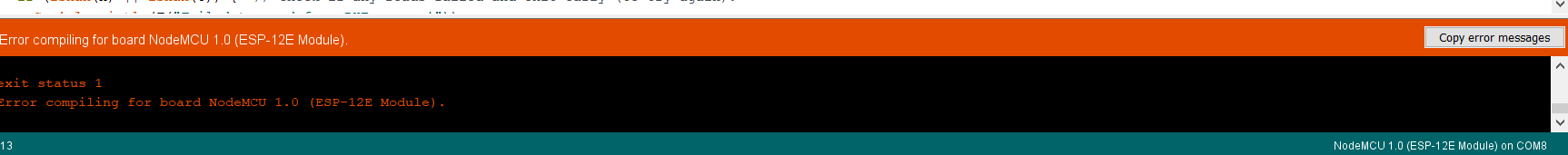
**1.Compilation :**

Validated to check whether the code was burnt on to the board(Node MCU).

Test scenario 1 : Board connection

This scenario requires three test cases (conditions) as follows

* 1.Board connection - Successful
* 2. Board connection - Unsuccessful when the board is not available in the Arduino IDE.
* 3. Board connection - Unsuccessful when board is still busy.



**FIG 4.26.Error Compiling Board**

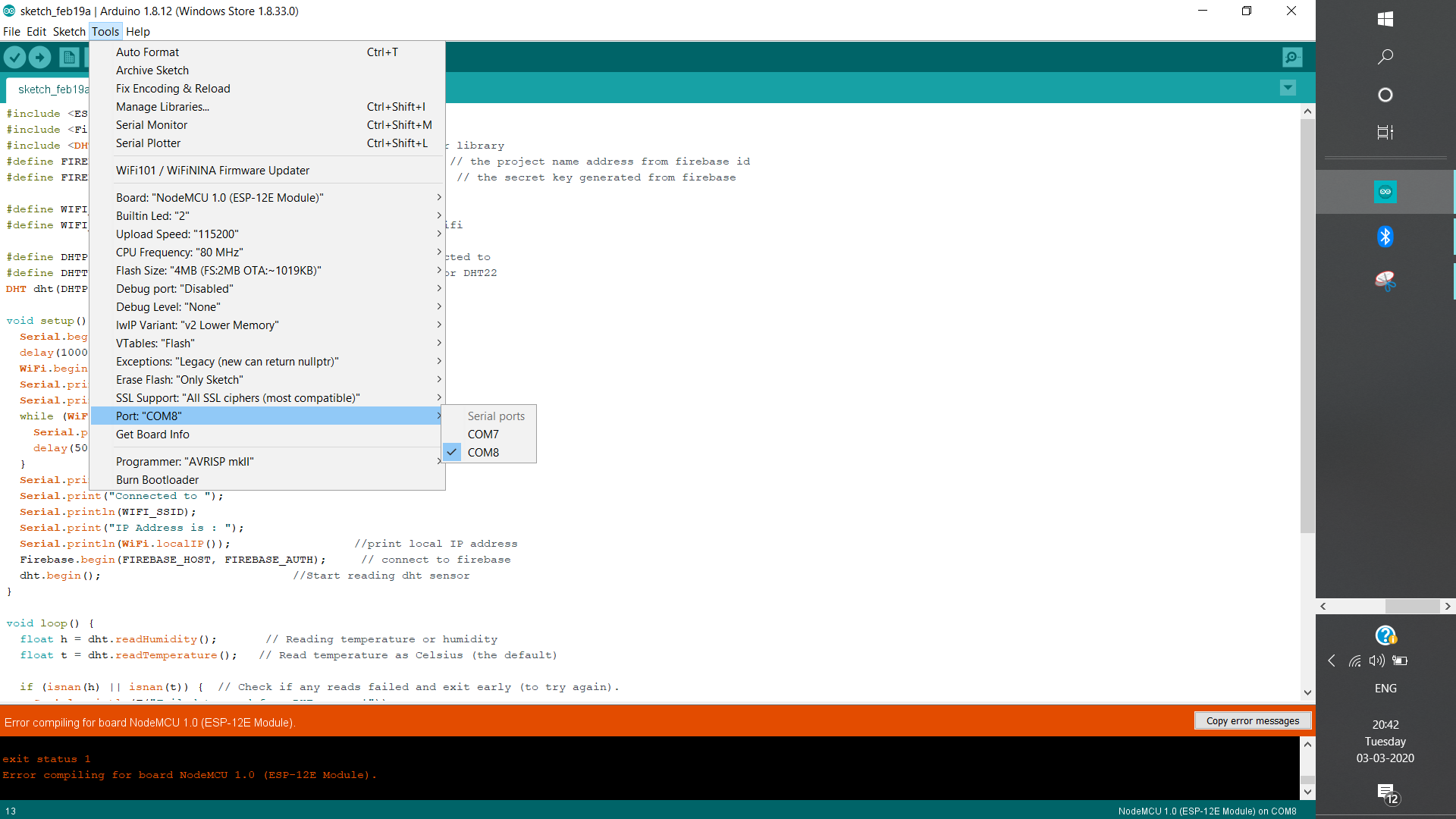
Test scenario 2 : Port connection

This scenario requires two test cases (conditions) as follows

1.Board connection – Successful when port is properly detected

2. Board connection - Unsuccessful when the port is not detected by your device like laptop/mobile.

3. Board connection - Unsuccessful when 2 or more ports are available and don’t know which port should be detected.



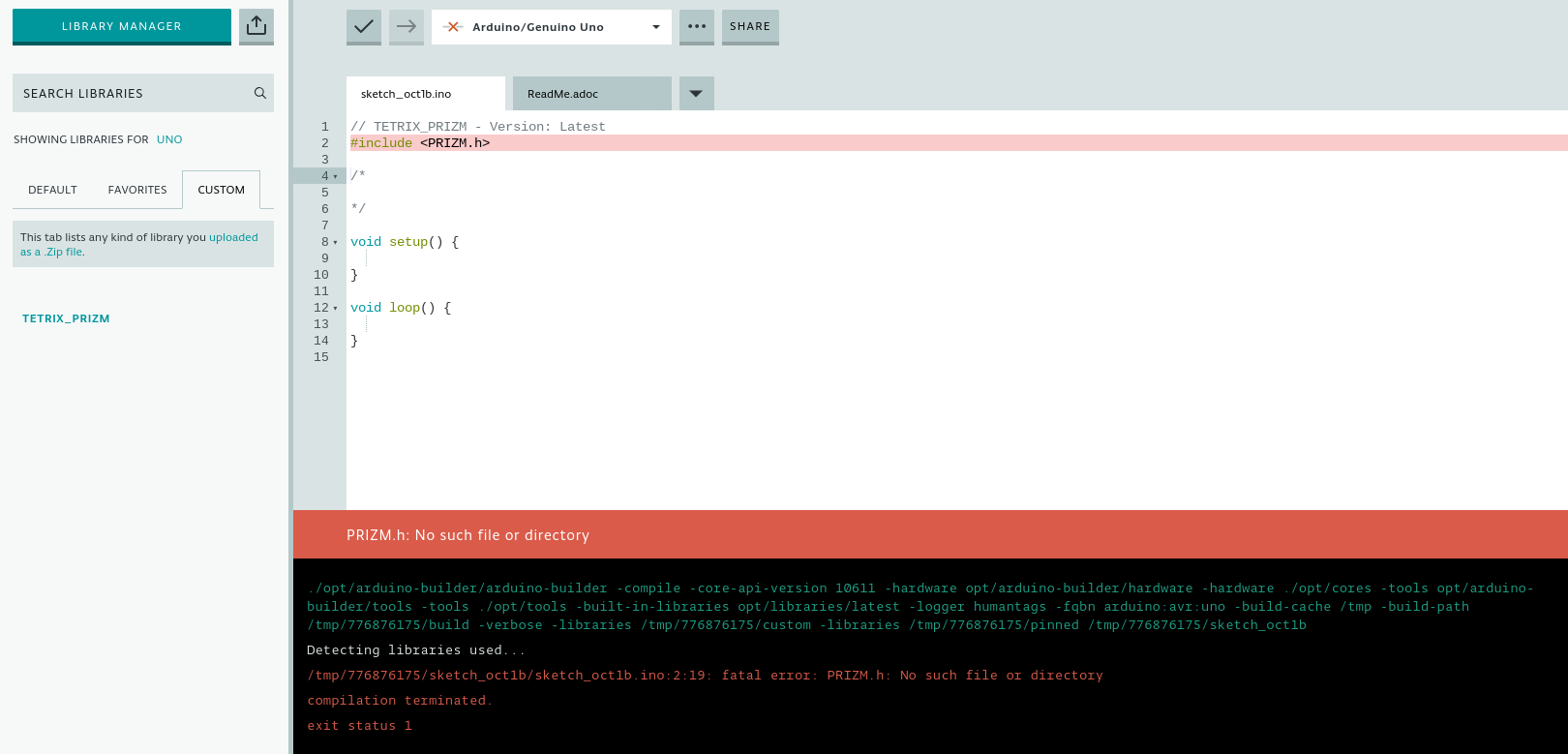
**FIG 4.27.Port not detected**

Test scenario 3 : Library not available

This scenario requires two test cases (conditions) as follows

1.Library found – If all the libraries present in a particular code is downloaded and added to the IDE

2. Library not found error.

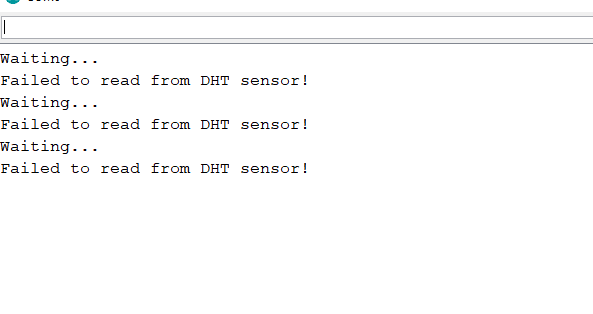


**FIG 4.28.Library not found**

Test scenario 4 : Sensor errors

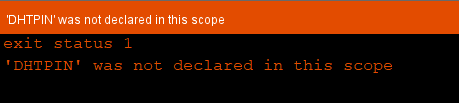
This scenario requires three test cases (conditions) as follows

1.Sensor not working



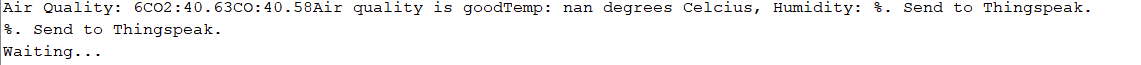
**FIG 4.29.Failed to read DHT sensor**

2.Sensor pin not declared



**FIG 4.30.DHTPIN not declared**

3.NAN (Not A Number) values detected by sensor



**FIG 4.31.Not a Number**

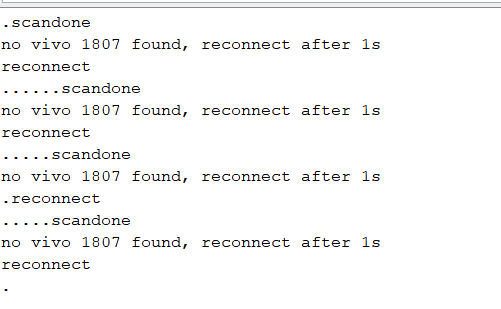
**2.Wifi and Thingspeak connection :**

Test scenario 1: Wifi connection

This scenario requires two test cases (conditions) as follows

1.Wifi found – If there exists a valid Wifi connection with high speed internet connection.

2. Wifi not found error.



**FIG 4.32.Wifi not found error**

**5.2. TEST APPROACHES**

Gray Box testing should be used with IOT testing as it allows to design effective test case. IOT testing ensures that the users get the improved user experience across all the connected IOT devices.

1) Compatibility testing :

* Looking at the architecture of the SMART MASK, compatibility testing is a must.
* Testing items such as, multiple operating system versions, browser types and respective versions, generations of devices, communication modes [**For e.g.** Wifi,Bluetooth] is necessary for IoT compatibility testing.

2) Pilot Testing:

* As far as the IoT is concerned, Pilot testing is a must.
* Only testing in Lab makes sure the product/system works fine. But, this may backfire badly when exposed to real-time conditions/steps/scenarios.

Here the SMART MASK is said to be continuously tested with real time conditions and is said to be compatible.

* During pilot testing, the system was exposed to a limited number of users in the real field.

3) Upgrade testing:

* IoT is a combination of multiple protocols, devices, operating systems, firmware, hardware, networking layers etc.
* Upgrade testing for the SMART MASK is done whenever an upgrade is performed, be it for the system or for any of the involved items. Regression testing should be carried out to overcome upgrade related issues.

**5.3. TEST REPORTS**

|  |  |  |
| --- | --- | --- |
| **PROJECT BASIC INFORMATION** | | |
| **Product Name** | SMART MASK | |
| **Product Description** | Air Quality Detecting mask | |
| **Mission of project** | 1.Detect the quality of air present around in ppm values, and also different gases which acts as the major pollutants.  2.Gives numerical data of air quality along with data visualization of the same in the form of graphs  3.Tells users whether to wear a filtered mask or not at that particular environment. | |
| **Project Type** | IOT (Internet Of Things) project | |
| **Project duration** | Start date:  18th Nov 2019 | End date:  5th Mar 2020 |

## 6.CONCLUSION

**6.1. DESIGN AND IMPLEMENTATION**

1) Usability:

* We need to make sure the usability of each of the device used here such that it should be portable enough to be moved everywhere.
* Usability in terms of displaying data, processing data, pushing job tasks from the devices should be tested thoroughly.
* The equipment should be smart enough to push not only the notifications but also the error messages, warnings etc.

2) IOT Security :

* IoT Security challenges: IoT is data centric where all the devices/system connected operate based on the data that is available.
* When it comes to the data flow between devices, there is always a chance that the data can be accessed or read when getting transferred.
* From a testing standpoint, we need to check if the data is protected/encrypted when getting transferred from one device to the other.

3) Connectivity:

* As it is a SMART MASK, connectivity plays a vital role.
* The system has to be available all the time and should have seamless connectivity with all the devices integrated.
* As per connectivity, two things are very important to test;
  1. Connectivity, transfer of data, receiving job tasks from the devices should be seamless when the connection is UP and running.
  2. The other condition is the connection down scenario. Doesn’t matter how robust is the system and the network, there are chances that the system will go offline. Hence whenever connection is not available there has to be an alert which can prompt the user .On the other hand, there has to be a mechanism in the system which can store all the data in it during the offline period. Once the system comes online, all that data should get propagated. Data loss should not be there in any condition.

4) Performance:

* Since SMART MASK works in a larger domain, we need to make sure the system is scalable enough for wider distance
* When the testing is carried out, the sensor was able to detect real time data continuously.
* When multiple sensors are connected to the system, the data that is propagated is much bigger than the tested data.

**6.2. ADVANTAGES AND LIMITATIONS**

**ADVANTAGES :**

**This project helped us to :**

1) To detect the harmful gases that an individual is inhaling.

2) To predict possible solution to reduce the intake of these gases

3) To provide a proper data analyzation of the gases inhaled by the user in the form of

graphs and numeric data

4) Easy view of the data on their smart devices (Laptops, Mobile phones)

5) To show the harmful gases inhaled on daily basis and to show the increase

or decrease in the intake level

**Contribution of work towards improving the performance of the existing system**

1.The design of mask was implemented because of its wider usage as user-friendly nature.

2. LCD display was integrated to inform users whether the mask was necessary to be worn or not.

3.SMART MASK app gives users more clearer information regarding the different gases present in air, depicts them using graphs and also gives notifications.

**6.3. FUTURE SCOPE OF THE PROJECT**

* Wireless integration of IOT components to obtain a completely wireless mask.
* Size of the mask could be reduced.
* Other than detection of specific gases like CO2,NH3,NO2,alcohol, benzene,smoke,CO2,Ammonia,Sulphide various other pollutants,O2,H2 can also be detected.
* Reducing the expenses of IOT devices by using fewer sensors and components that would have multiple functionalities.
* Implementing various data visualization techniques for depiction of different gases and their comparisions .

**a)AREAS OF INVESTIGATION IDENTIFIED :**

* Next Generation Air Measurement Technologies
* Sensor Evaluations:
* Air Quality Measurement Methods
* Air Toxics Methods
* Stationary Source Emissions Methods and Instruments:
* Air Quality Emissions Inventories Using Measurements and Models (EPA’s SPECIATE Database Support)
* On Road and Off Road Emissions Inventories
* Analytical-Chemical and Optical Methods
* Black Carbon Characterization

**b) Parts of the current work that could not be completed**

**due to time constraints and/or problems encountered.**

* LCD couldn’t be connected to the mask as there was difficulty in powering it up and required few other additional components to be attached to mask to make it work.
* Battery vould not be inbulited in the mask because of its high voltage which would damage the components present and this could be achieved with resistors that would require breadboards which would make the mask more bigger.

**RFERENCES**

1. AIRPARIF:The Automatic Network. 2012,June 2012<http://www.airparif.asso.fr/ >.

[2] A Smart Air Pollution Monitoring System,

<https://www.researchgate.net/publication/328015436 >.

[3] AIRNOW: Local Air Quality Conditions and forecasts. 2012,accessed June 2012,

<http://airnow.gov>.

[4] Electrochemical Sensors-Sensor Technology 2012, accessed June,

<http://www.delphian.com/electrochemical%20sensors.html >.

[5] Air Monitoring, Measuring, and Emissions Research,

<<https://www.epa.gov/air-research/air-monitoring-measuring-and-emissions-research>>

[[6] Cordis,17th rendition,](https://cordis.europa.eu/docs/projects/cnect/5/288385/080/deliverables/001-288385IoTestD17renditionDownload.pdf)

[<cordis.europa.eu/docs/projects/cnect/5/288385/080/deliverables/001-288385.pdf](https://cordis.europa.eu/docs/projects/cnect/5/288385/080/deliverables/001-288385IoTestD17renditionDownload.pdf)>

[7] <https://thingspeak.com/account>

[8]. <https://appinventor.mit.edu/>