Signs with smart connectivity for better road safety

IBM- DOCUMENTATION

UNDER THE GUIDANCE OF

Industry mentor name: Santoshi

Faculty mentor name: V.SUDHA

TEAM ID: PNT2022TMID38970

SUBMITTED BY:

NAMES and reg.no:

J. Sowndharya - 421319106035

S. Akshaya - 421319106003

K. Ramani - 421319106027

A. Umamaheswari - 421319106039

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
KRISHNASAMY COLLEGE OF ENGINEERING AND TECHNOLOGY 2019-2023
ANNA UNIVERSITY :: 2019-2023

INTRODUCTION

Road traffic accident is a major problem worldwide resulting in significant morbidity and mortality. Advanced driver assistance systems are one of the salient features of intelligent systems in transportation. They improve vehicle safety by providing real-time traffic information to the driver. Road signs play an important role in road safety.

Smart roadway signage is not simply an objective for the future. Two UK companies have collaborated to produce these signs for use on England's roads. The signs are technologically advanced, with graphics and text that drivers can see clearly. The messages are easy to comprehend quickly, keeping drivers informed of route conditions as they change. In addition to enhancing the roadway experience for users, this new signage costs less to maintain than traditional indicators. The new signs require fewer materials and less cabling, resulting in less time, upkeep, and expense.

Increasing volumes of traffic are using municipal road infrastructure, with severe consequences for traffic efficiency and the safety of road users. Vulnerable roads users (VRUs), such as pedestrians or cyclists, are involved in 46 % of lethal accidents. Exchanging information between road users increases their perception and is thus a critical building block to improve this situation. We have presented a system, to alert the driver about the speed limits in specific areas and reduce the speed of the vehicles in sensitive public zones without any interference of the drivers where controls are taken automatically by the use of a wireless local area networks.

PURPOSE

- To improve vehicle safety by providing real-time traffic information to the driver. Road signs play an important role in road safety.
- To be effective, road signs must be visible at a distance that enables drivers to take the necessary actions.
- To avoid the accidents, due to the increasing vehicle speed.
- To provides a voice alert to a needed action that enhances driver's attention.

Literature Survey

Existing Problem

- Increased traffic can increase carbon emissions and other pollution.
- Land use for roads can damage built and natural environment, impose mortality on wildlife if habitats are severed.
- Construction has associated environmental costs.

REFERENCE

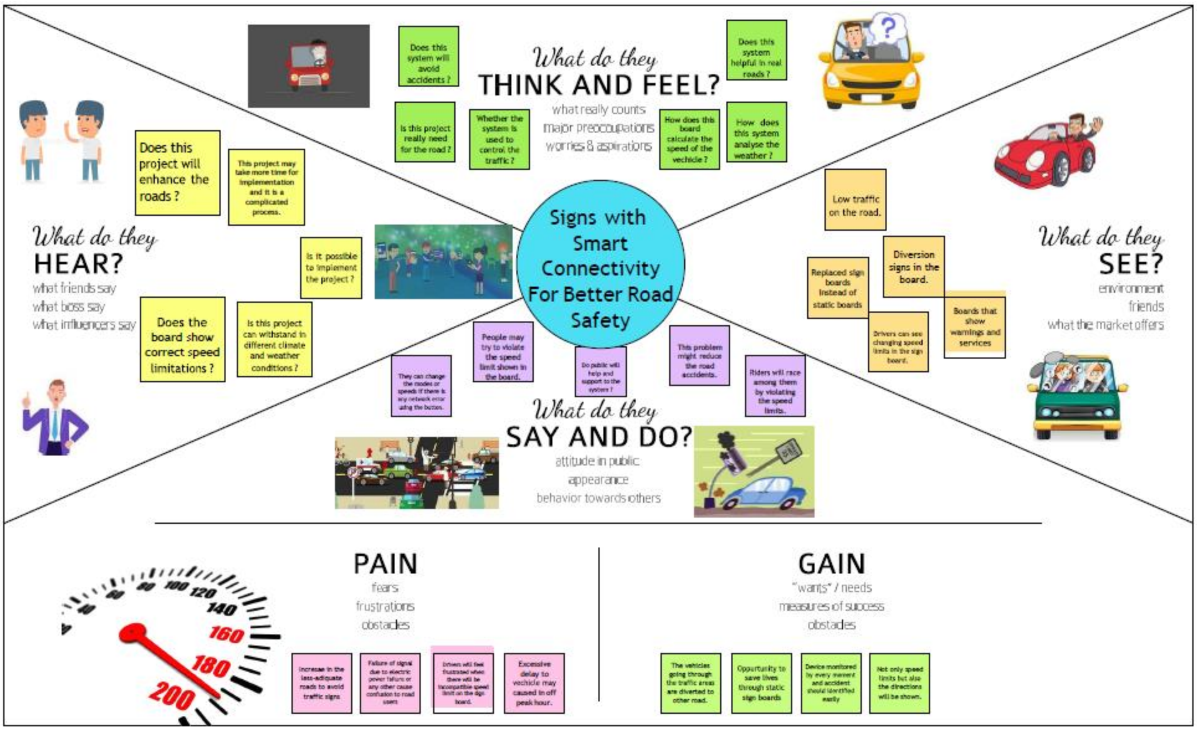
- Samir A. Elagheer Mohamed, Khaleq A. Alshafan and May 2021 – Intelligent traffic management system based on the IOT
- Andrea pompigna, raffaele Mauro and april 2021 – Smart roads: a state of the art of highways innovations in the smart age
- Ranjitha m, spoorthi, jwantia, soumya k and 2021 – IOT based smart traffic management system
- Ashish sharma, yogesh awasthi, sunil kumar and 2020 – The role of block chain, AI and IOT for smart road traffic management system
- Sabeen javid, ali sufrain, saema pervaiz, mehaktanveer and 2018 – Smart traffic management system using IOT

Problem Statement Definition:

- A driver who wants to drive safely on road but there are many obstacles because of heavy traffic, weather condition, etc..,
- A driver who wants to avoid the heavy traffic roads but they are unpredictable because they change from time to time.
- A passenger who wants to travel safely but there are many road accidents because of some drivers who drive very fast and carelessly.
- A driver who wants to reach the destination but unable to choose the route and turn in wrong direction because there are no navigation instructions.

Ideation and Proposed Solution:

Empathy Map:



Signs with Smart Connectivity For Better Road Safety

What do they THINK AND FEEL?

what really counts
major preoccupations
worries & aspirations

Does this system will avoid accidents ?

Is this project really need for the road ?

Whether the system is used to control the traffic ?

How does this board calculate the speed of the vechicle ?

Does this system helpful in real roads ?

How does this system analyse the weather ?

What do they HEAR?

what friends say
what boss say
what influencers say

Does this project will enhance the roads ?

This project may take more time for implementation and it is a complicated process.

Is it possible to implement the project ?

Does the board show correct speed limitations ?

Is this project can withstand in different climate and weather conditions ?

What do they SEE?

environment
friends
what the market offers

Low traffic on the road.

Diversion signs in the board.

Replaced sign boards instead of static boards

Drivers can see changing speed limits in the sign board.

Boards that show warnings and services

What do they SAY AND DO?

attitude in public
appearance
behavior towards others

They can change the modes or speeds if there is any network error using the button.

People may try to violate the speed limit shown in the board.

Do public will help and support to the system ?

This problem might reduce the road accidents.

Riders will race among them by violating the speed limits.

PAIN

fears
frustrations
obstacles

Increase in the less-adiquate roads to avoid traffic signs

Failure of signal due to electric power failure or any other cause confusion to road users

Drivers will feel frustrated when there will be incompatible speed limit on the sign board.

Excessive delay to vechicle may caused in off peak hour.

GAIN

"wants" / needs
measures of success
obstacles

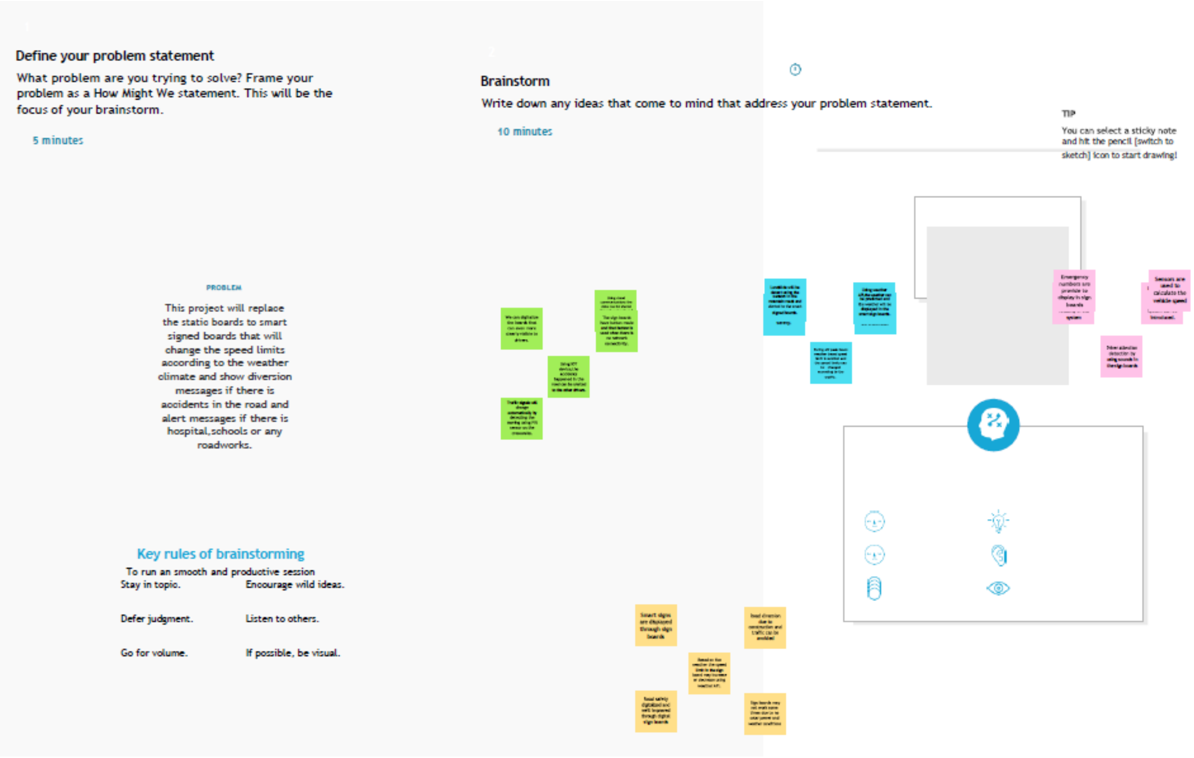
The vehicles going through the traffic areas are diverted to other road.

Oppurtunity to save lives through static sign boards

Device monitored by every moment and accident should identified early

Not only speed limits but also the directions will be shown.

Brainstorming:



Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM

This project will replace the static boards to smart signed boards that will change the speed limits according to the weather climate and show diversion messages if there is accidents in the road and alert messages if there is hospital,schools or any roadworks.

Key rules of brainstorming

To run an smooth and productive session

Stay in topic. Encourage wild ideas.

Defer judgment. Listen to others.

Go for volume. If possible, be visual.

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

We can digitalize the boards that can even more clearly visible to drivers.

Using cloud communications the data can be shared [illegible]

The sign boards have button made and that button is used when there is no network connectivity.

Using IOT device,the accidents happened in the road can be alerted to the other drivers.

Traffic signals will change automatically by detecting the [illegible] using PIR sensor on the crosswalk.

Landslide will be detect using the sensors in the mountain roads and alerted to the smart signal boards.

[illegible] safety.

Using weather API,the weather can be predicted and the weather will be displayed in the smart sign boards.

During off peak hours weather based speed limit is enabled and the speed limits can be changed according to the users.

Emergency numbers are provide to display in sign boards

[illegible] system

Sensors are used to calculate the vehicle speed

[illegible] introduced.

Driver attention detection by using sounds in the sign boards

Smart signs are displayed through sign boards

Road diversion due to construction and traffic can be avoided

Based on the weather the speed limit in the sign board may increase or decrease using weather API.

Road safety digitalized and well improved through digital sign boards

Sign boards may not work some times due to no solar power and weather conditions

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

High brightness signs are used to improve road safety.

To provide the better view for the drivers.

We can digitalize the boards that can even more clear to drivers.

Regulatory mode of the sign boards improve the reliability of the sensor

TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

Smart signs are displayed through sign boards

Using AI the traffic signals would be controlled for the people who using the [illegible].

The sign boards have button made and that button is used when there is no network connectivity.

Traffic Control

Traffic density by using the GPS of mobile phone

During off peak hours weather based speed limit is updated and the speed limits can be changed according to the traffic.

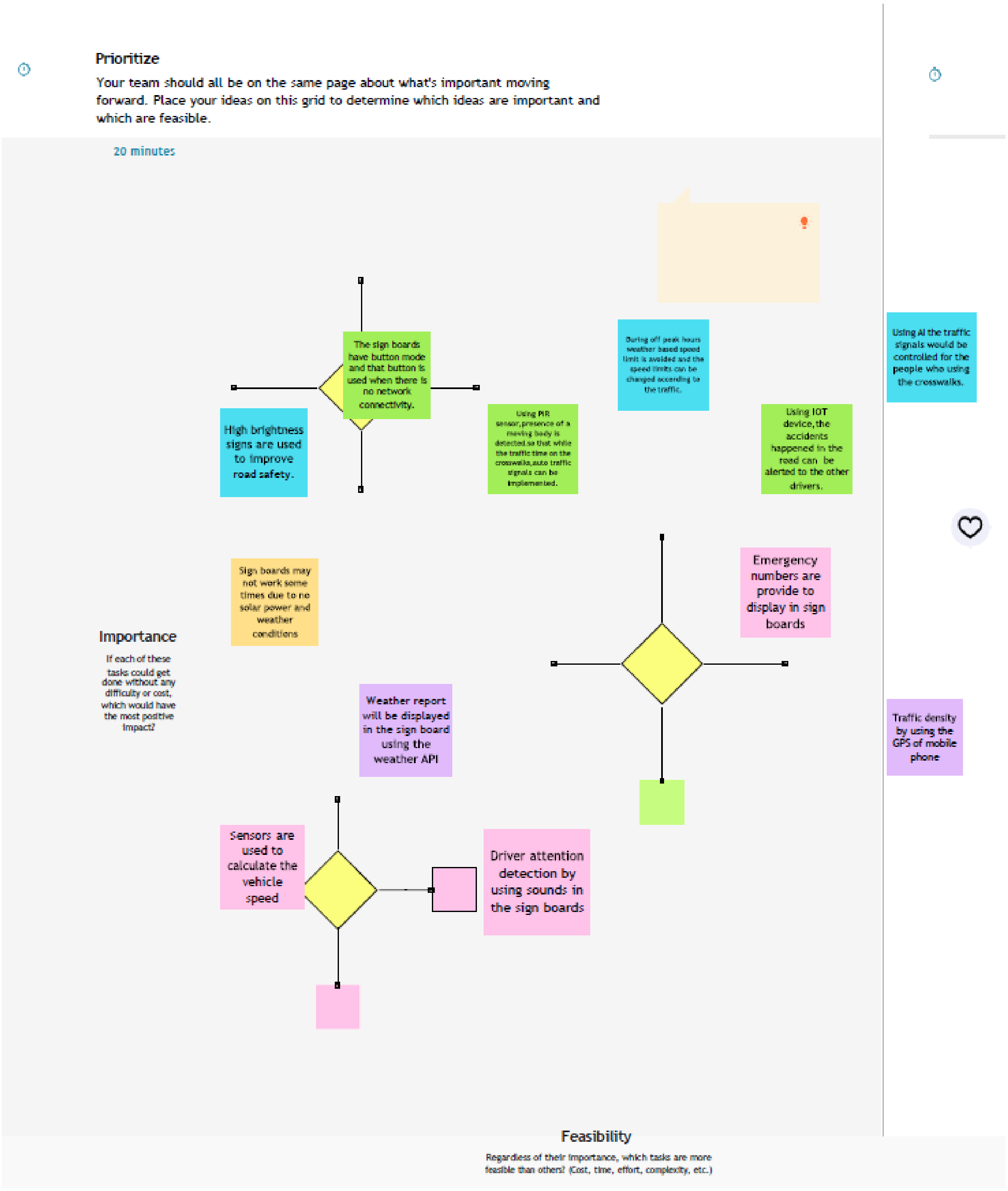
[illegible]

During off peak hours weather has speed limit is understood the speed limits can be changed according to the traffic.

Weather based Controlling.

Weather report will be displayed in the sign board using the weather API

Based on the weather the speed limit in the sign board may increase or decrease using weather API.



Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

The sign boards have button mode and that button is used when there is no network connectivity.

During off peak hours weather based speed limit is avoided and the speed limits can be changed according to the traffic.

Using AI the traffic signals would be controlled for the people who using the crosswalks.

High brightness signs are used to improve road safety.

Using PIR sensor,presence of a moving body is detected,so that while the traffic time on the crosswalks,auto traffic signals can be implemented.

Using IOT device,the accidents happened in the road can be alerted to the other drivers.

Emergency numbers are provide to display in sign boards

Sign boards may not work some times due to no solar power and weather conditions

Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

Weather report will be displayed in the sign board using the weather API

Traffic density by using the GPS of mobile phone

Sensors are used to calculate the vehicle speed

Driver attention detection by using sounds in the sign boards

Feasibility

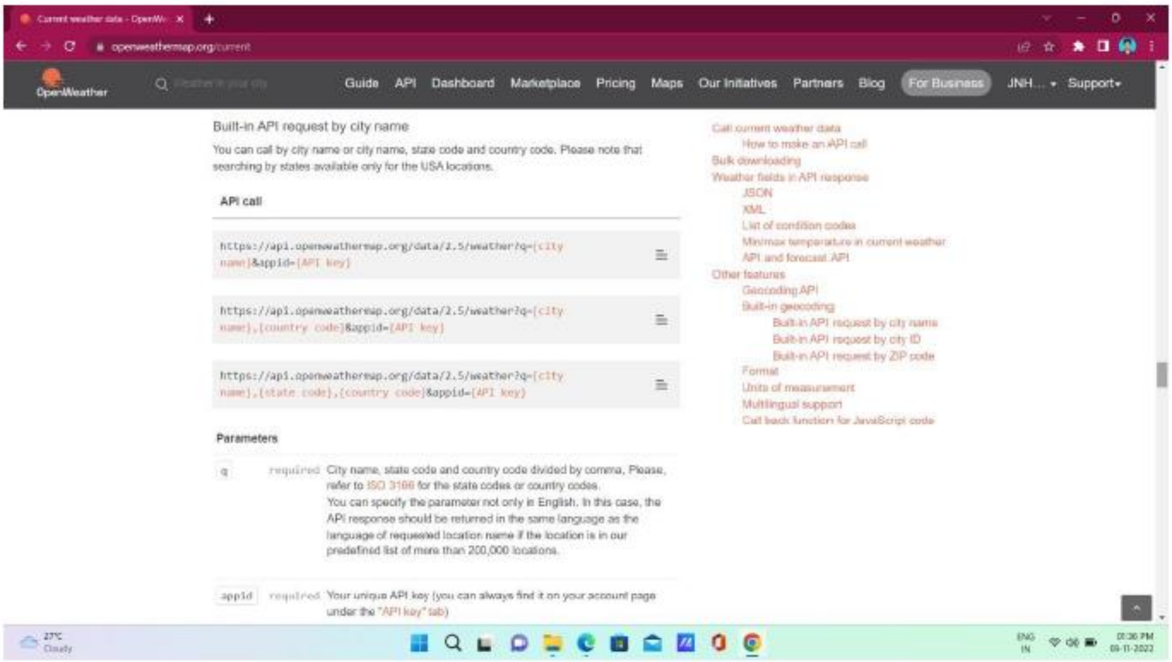
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

Problem Solution Fit:

- CUSTOMER SEGMENTS – It satisfies all customers who drives any kind of vehicle.
- JOBS-TO-BE-DONE/PROBLEMS – By getting information regarding the change in weather conditions, customer can able to find out the path from the web app.
- TRIGGERS – Creating awareness among the public to use the web app.
- EMOTIONS-BEFORE/AFTER – BEFORE: Feeling unsafe , Takes more time to reach; AFTER: Safe to travel , Reaching on- time, Accident prevention.
- AVAILABLE SOLUTION – Static signboards are now available which simply provide constant information about the speed limits, warnings inorder to have a safe travel.
- CUSTOMER CONSTRAINTS – Network connectivity problems.
- BEHAVIOUR – Choosing a right choice of IOT boards.
- CHANNELS OF BEHAVIOUR – Surfing the network regarding issues arised.

- PROBLEM ROOT CAUSE – Road transport drivers find it hard to get adapted to the changes prevailing in the weather, traffic, other fatal situations and thereby applying suitable driving measures like increasing/ decreasing speeds and taking diversion measures accordingly becomes a challenging task.
- YOUR SOLUTION – Rather than using static signboards, smart sign boards can be used which gives the timely updated information regarding roads, roads diversion measures and change in weather is being displayed through the web application developed.

Weather Map:



Built-in API request by city name

You can call by city name or city name, state code and country code. Please note that searching by states available only for the USA locations.

API call

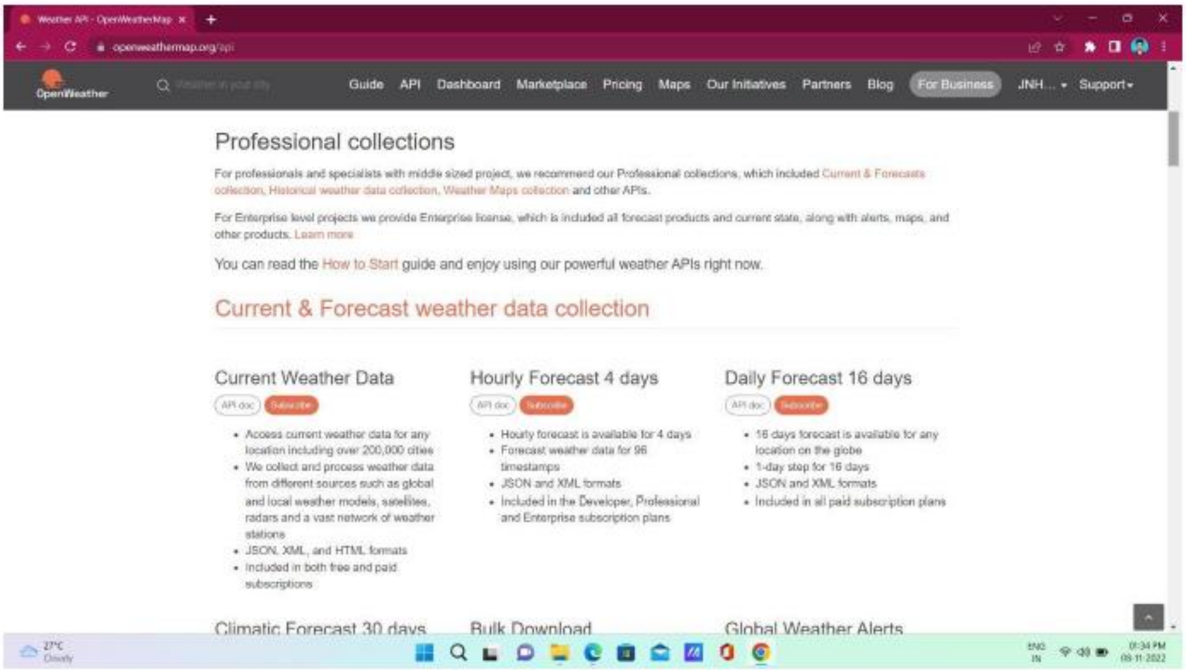
```
https://api.openweathermap.org/data/2.5/weather?q={city name}&appid={API key}
```

```
https://api.openweathermap.org/data/2.5/weather?q={city name},{country code}&appid={API key}
```

```
https://api.openweathermap.org/data/2.5/weather?q={city name},{state code},{country code}&appid={API key}
```

Parameters

q	required	City name, state code and country code divided by comma, Please, refer to ISO 3166 for the state codes or country codes. You can specify the parameter not only in English. In this case, the API response should be returned in the same language as the language of requested location name if the location is in our predefined list of more than 200,000 locations.
appid	required	Your unique API key (you can always find it on your account page under the "API key" tab)



Professional collections

For professionals and specialists with middle sized project, we recommend our Professional collections, which included Current & Forecasts collection, Historical weather data collection, Weather Maps collection and other APIs.

For Enterprise level projects we provide Enterprise license, which is included all forecast products and current state, along with alerts, maps, and other products. Learn more

You can read the How to Start guide and enjoy using our powerful weather APIs right now.

Current & Forecast weather data collection

Current Weather Data

- Access current weather data for any location including over 200,000 cities
- We collect and process weather data from different sources such as global and local weather models, satellites, radars and a vast network of weather stations
- JSON, XML, and HTML formats
- Included in both free and paid subscriptions

Hourly Forecast 4 days

- Hourly forecast is available for 4 days
- Forecast weather data for 96 timestamps
- JSON and XML formats
- Included in the Developer, Professional and Enterprise subscription plans

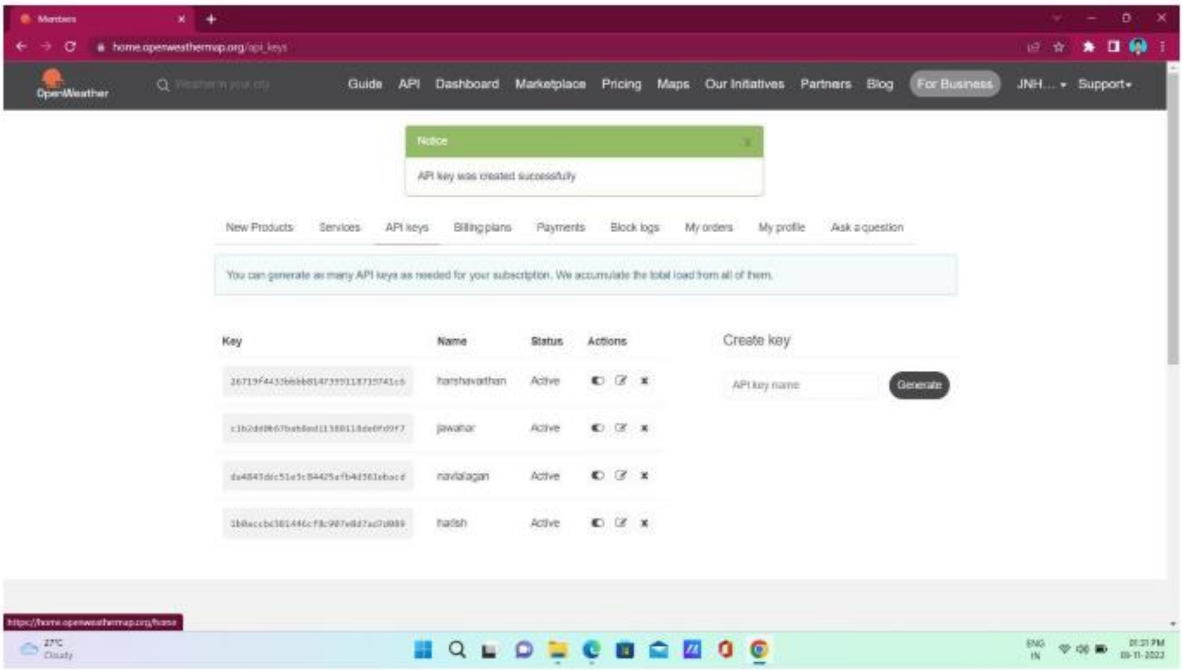
Daily Forecast 16 days

- 16 days forecast is available for any location on the globe
- 1-day step for 16 days
- JSON and XML formats
- Included in all paid subscription plans

Climatic Forecast 30 days

Bulk Download

Global Weather Alerts



Notice

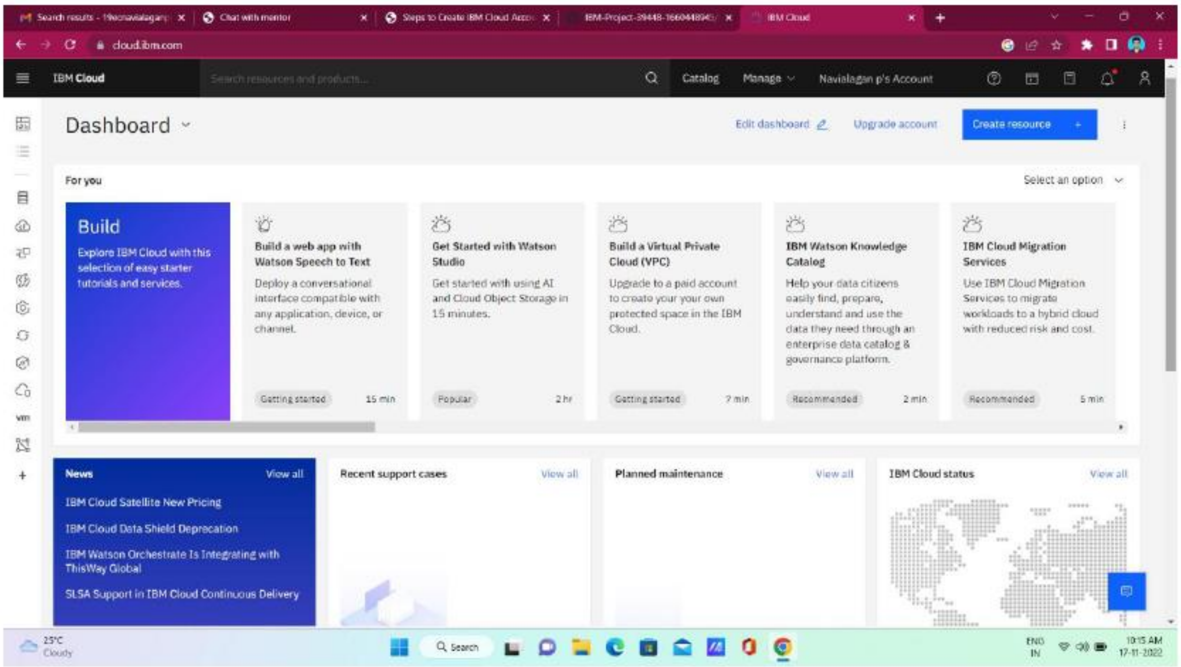
API key was created successfully

You can generate as many API keys as needed for your subscription. We accumulate the total load from all of them.

Key	Name	Status	Actions
26719f4433b6bb8147395118715741c6	harshavardhan	Active	
c1b2dd0b67bab8ad11380118de0fd9f7	jawahar	Active	
da4843dcc51e5c84425efb4d361ebacd	navialagan	Active	
1b0accbd381446cf8c907e8d7ad7d088	harish	Active	

Create key

API key name



For you

Build

Explore IBM Cloud with this selection of easy starter tutorials and services.

Build a web app with Watson Speech to Text

Deploy a conversational interface compatible with any application, device, or channel.

Getting started 15 min

Get Started with Watson Studio

Get started with using AI and Cloud Object Storage in 15 minutes.

Popular 2 hr

Build a Virtual Private Cloud (VPC)

Upgrade to a paid account to create your your own protected space in the IBM Cloud.

Getting started 7 min

IBM Watson Knowledge Catalog

Help your data citizens easily find, prepare, understand and use the data they need through an enterprise data catalog & governance platform.

Recommended 2 min

IBM Cloud Migration Services

Use IBM Cloud Migration Services to migrate workloads to a hybrid cloud with reduced risk and cost.

Recommended 5 min

News View all

IBM Cloud Satellite New Pricing

IBM Cloud Data Shield Deprecation

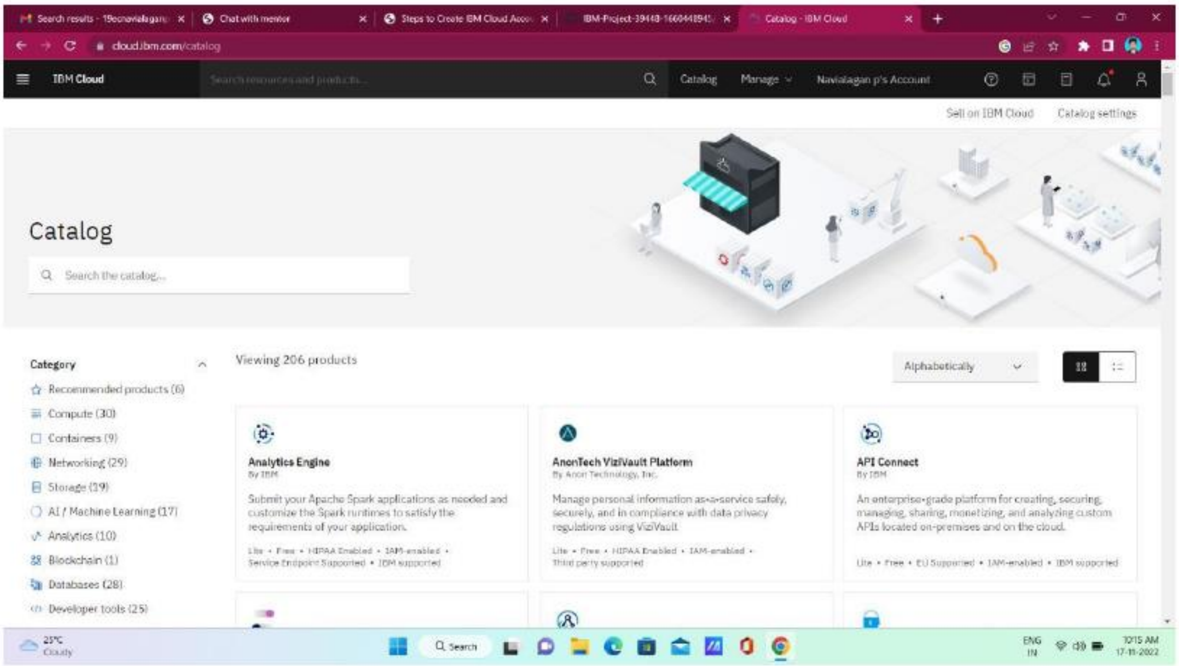
IBM Watson Orchestrate Is Integrating with ThisWay Global

SLSA Support in IBM Cloud Continuous Delivery

Recent support cases View all

Planned maintenance View all

IBM Cloud status View all



Catalog

Search the catalog...

Category

- Recommended products (6)
- Compute (30)
- Containers (9)
- Networking (29)
- Storage (19)
- AI / Machine Learning (17)
- Analytics (10)
- Blockchain (1)
- Databases (28)
- Developer tools (25)

Viewing 206 products

Analytics Engine
By IBM
Submit your Apache Spark applications as needed and customize the Spark runtimes to satisfy the requirements of your application.
Lite • Free • HIPAA Enabled • IAM-enabled • Service Endpoint Supported • IBM supported

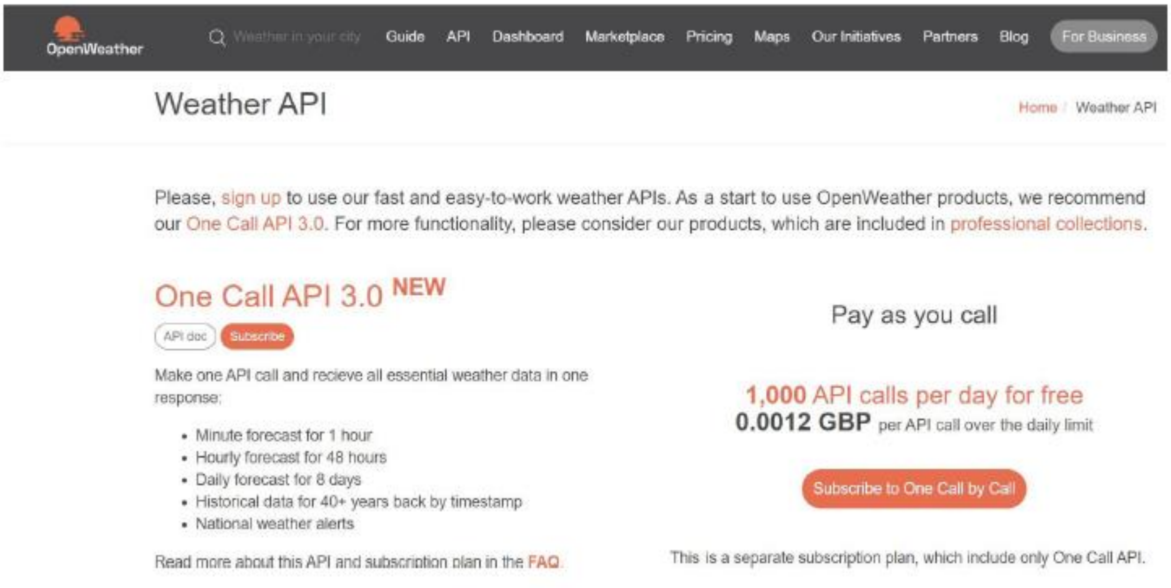
AnonTech ViziVault Platform
By Anon Technology, Inc.
Manage personal information as-a-service safely, securely, and in compliance with data privacy regulations using ViziVault.
Lite • Free • HIPAA Enabled • IAM-enabled • Third party supported

API Connect
By IBM
An enterprise-grade platform for creating, securing, managing, sharing, monetizing, and analyzing custom APIs located on-premises and on the cloud.
Lite • Free • EU Supported • IAM-enabled • IBM supported

Develop a Python Script:

Create a code snippet using python to

- **Extract weather data from OpenWeatherMap using APIs**
- **Send the extracted data to the cloud**
- **Receive data from the cloud and view it in the python compiler**



Weather API

Please, sign up to use our fast and easy-to-work weather APIs. As a start to use OpenWeather products, we recommend our One Call API 3.0. For more functionality, please consider our products, which are included in professional collections.

One Call API 3.0 NEW

Make one API call and recieve all essential weather data in one response:

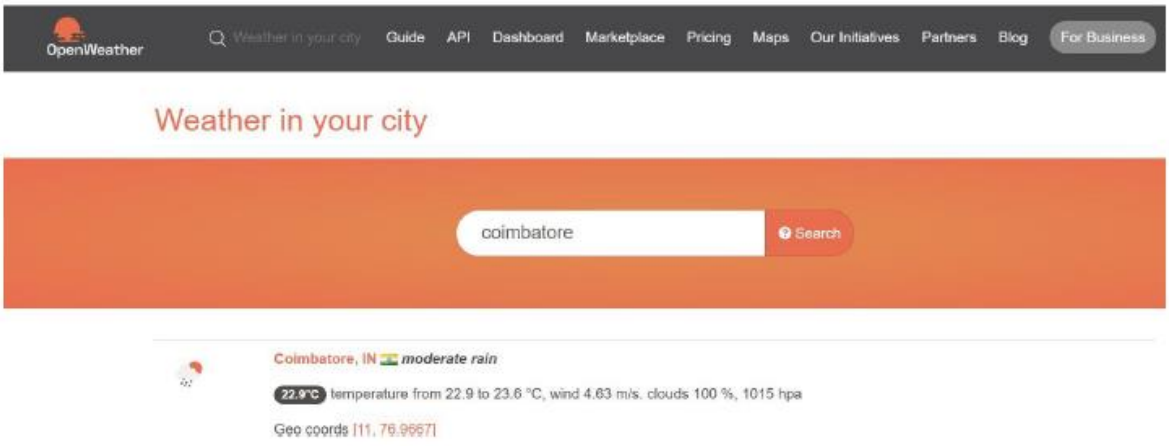
- Minute forecast for 1 hour
- Hourly forecast for 48 hours
- Daily forecast for 8 days
- Historical data for 40+ years back by timestamp
- National weather alerts

Read more about this API and subscription plan in the FAQ.

Pay as you call

1,000 API calls per day for free
0.0012 GBP per API call over the daily limit

This is a separate subscription plan, which include only One Call API.



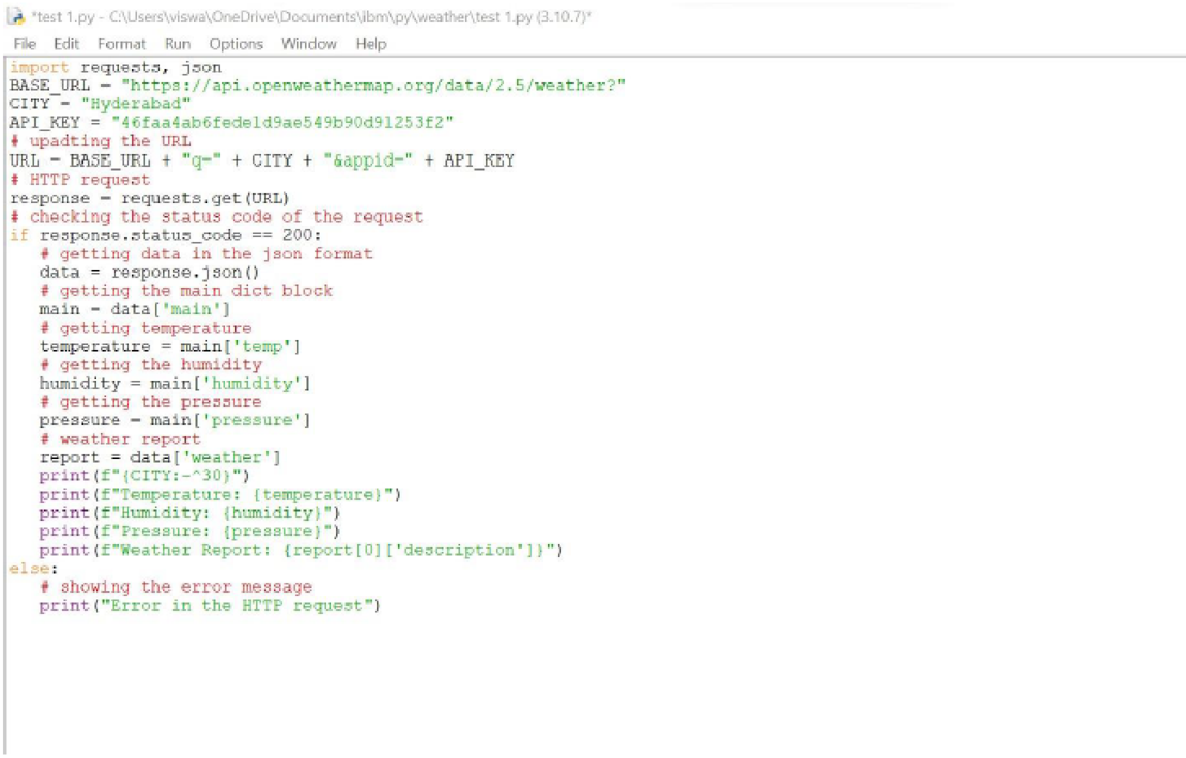
Weather in your city

coimbatore

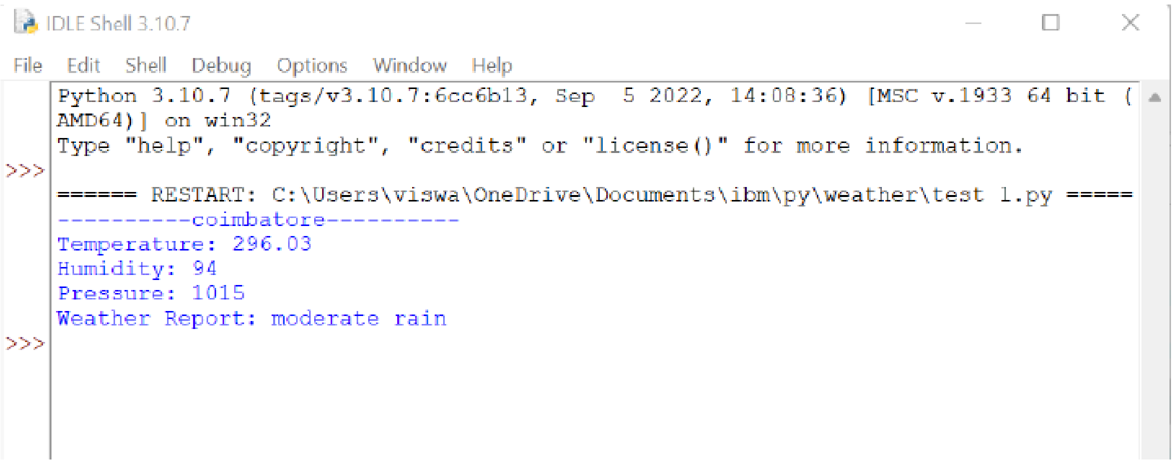
Coimbatore, IN *moderate rain*

22.9°C temperature from 22.9 to 23.6 °C, wind 4.63 m/s. clouds 100 %, 1015 hpa

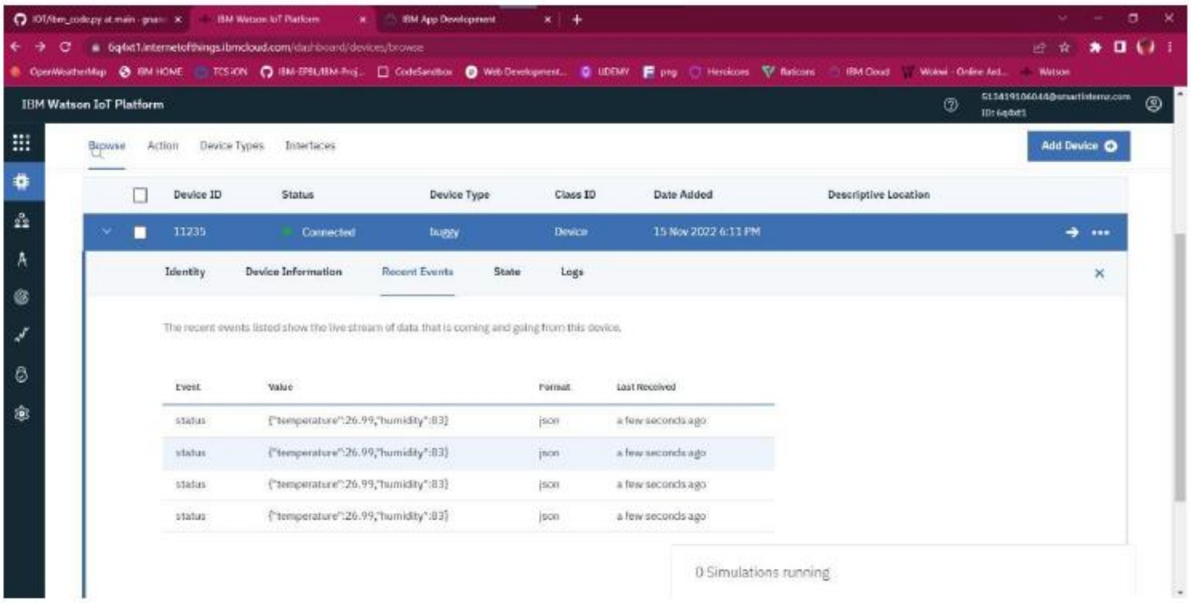
Geo coords [11, 76.9667]


test 1.py - C:\Users\viswa\OneDrive\Documents\ibm\py\weather\test 1.py (3.10.7)
File Edit Format Run Options Window Help

```
import requests, json
BASE_URL = "https://api.openweathermap.org/data/2.5/weather?"
CITY = "Hyderabad"
API_KEY = "46faa4ab6fede1d9ae549b90d91253f2"
# upadting the URL
URL = BASE_URL + "q=" + CITY + "&appid=" + API_KEY
# HTTP request
response = requests.get(URL)
# checking the status code of the request
if response.status_code == 200:
   # getting data in the json format
   data = response.json()
   # getting the main dict block
   main = data['main']
   # getting temperature
   temperature = main['temp']
   # getting the humidity
   humidity = main['humidity']
   # getting the pressure
   pressure = main['pressure']
   # weather report
   report = data['weather']
   print(f"{CITY:-^30}")
   print(f"Temperature: {temperature}")
   print(f"Humidity: {humidity}")
   print(f"Pressure: {pressure}")
   print(f"Weather Report: {report[0]['description']}")
else:
   # showing the error message
   print("Error in the HTTP request")
```



```
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep  5 2022, 14:08:36) [MSC v.1933 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
====== RESTART: C:\Users\viswa\OneDrive\Documents\ibm\py\weather\test 1.py =====
----------coimbatore----------
Temperature: 296.03
Humidity: 94
Pressure: 1015
Weather Report: moderate rain
>>>
```



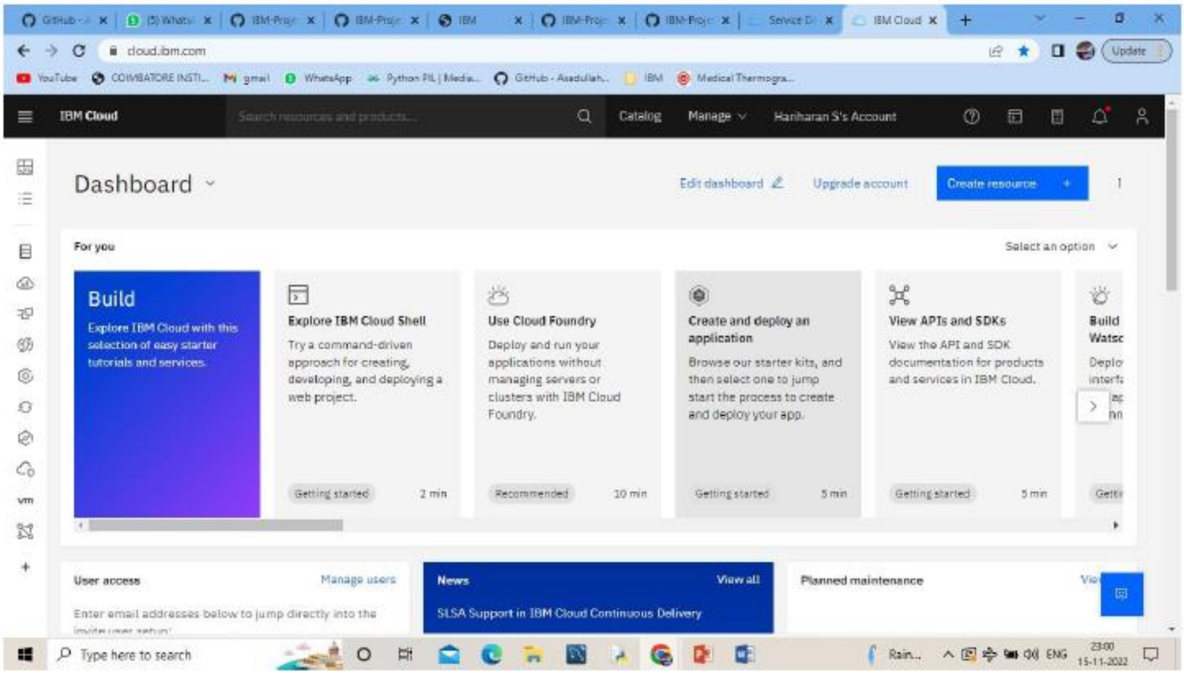
	Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
□	11235	Connected	buggy	Device	15 Nov 2022 6:11 PM	

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
status	{"temperature":26.99,"humidity":83}	json	a few seconds ago
status	{"temperature":26.99,"humidity":83}	json	a few seconds ago
status	{"temperature":26.99,"humidity":83}	json	a few seconds ago
status	{"temperature":26.99,"humidity":83}	json	a few seconds ago

0 Simulations running

Watson IoT platform:



Dashboard

Edit dashboard

Upgrade account

Create resource +

For you

Select an option

Build

Explore IBM Cloud with this selection of easy starter tutorials and services.

Explore IBM Cloud Shell

Try a command-driven approach for creating, developing, and deploying a web project.

Getting started 2 min

Use Cloud Foundry

Deploy and run your applications without managing servers or clusters with IBM Cloud Foundry.

Recommended 10 min

Create and deploy an application

Browse our starter kits, and then select one to jump start the process to create and deploy your app.

Getting started 5 min

View APIs and SDKs

View the API and SDK documentation for products and services in IBM Cloud.

Getting started 5 min

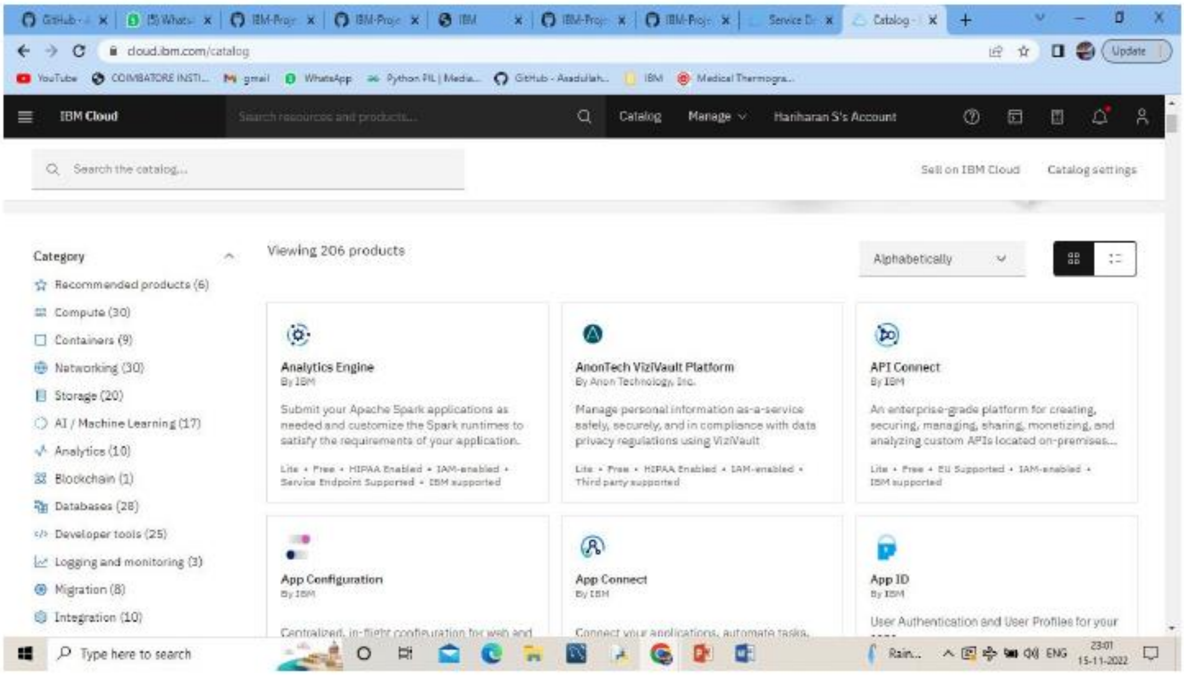
User access Manage users

Enter email addresses below to jump directly into the invite user setup'

News View all

SLSA Support in IBM Cloud Continuous Delivery

Planned maintenance



Search the catalog...

Sell on IBM Cloud

Catalog settings

Category

- Recommended products (6)
- Compute (30)
- Containers (9)
- Networking (30)
- Storage (20)
- AI / Machine Learning (17)
- Analytics (10)
- Blockchain (1)
- Databases (28)
- Developer tools (25)
- Logging and monitoring (3)
- Migration (8)
- Integration (10)

Viewing 206 products

Alphabetically

Analytics Engine
By IBM

Submit your Apache Spark applications as needed and customize the Spark runtimes to satisfy the requirements of your application.

Lite • Free • HIPAA Enabled • IAM-enabled • Service Endpoint Supported • IBM supported

AnonTech ViziVault Platform
By Anon Technology, Inc.

Manage personal information as-a-service safely, securely, and in compliance with data privacy regulations using ViziVault

Lite • Free • HIPAA Enabled • IAM-enabled • Third party supported

API Connect
By IBM

An enterprise-grade platform for creating, securing, managing, sharing, monetizing, and analyzing custom APIs located on-premises...

Lite • Free • EU Supported • IAM-enabled • IBM supported

App Configuration
By IBM

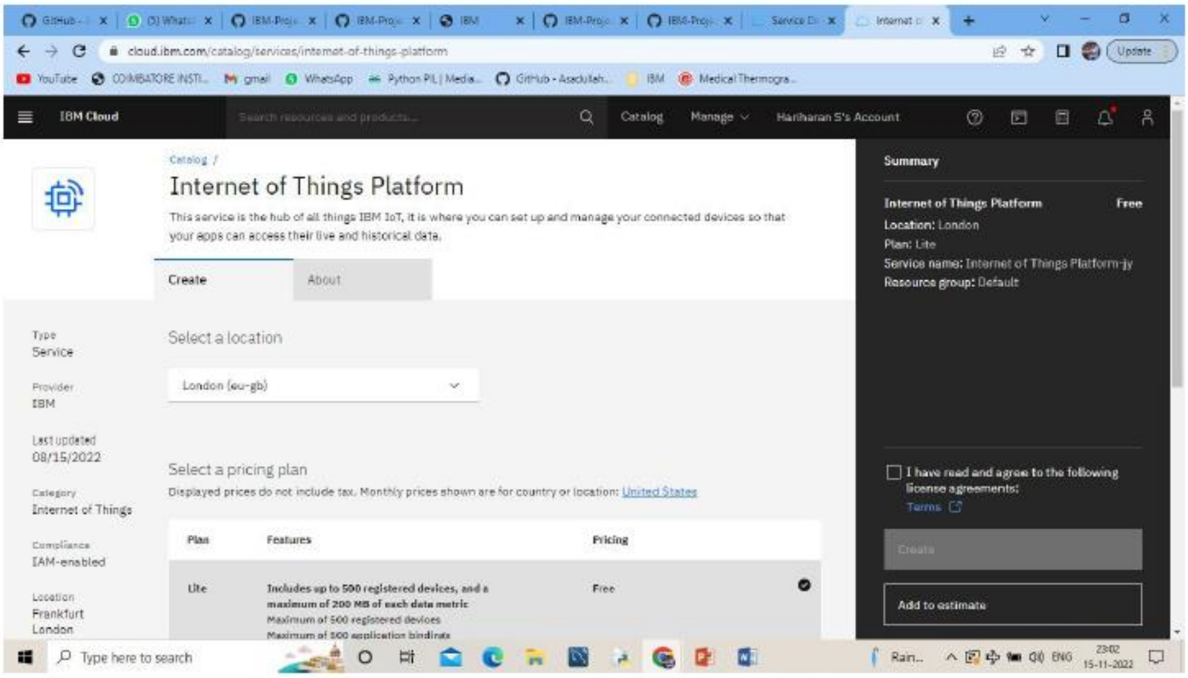
Centralized, in-flight configuration for web and

App Connect
By IBM

Connect your applications, automate tasks,

App ID
By IBM

User Authentication and User Profiles for your



cloud.ibm.com/catalog/services/internet-of-things-platform
IBM Cloud
Search resources and products...
Catalog
Manage
Hariharan S's Account
Type here to search
23:02
15-11-2022

Catalog /

Internet of Things Platform

This service is the hub of all things IBM IoT, it is where you can set up and manage your connected devices so that your apps can access their live and historical data.

Create | About

Type
Service

Provider
IBM

Last updated
08/15/2022

Category
Internet of Things

Compliance
IAM-enabled

Location
Frankfurt
London

Select a location

London (eu-gb)

Select a pricing plan

Displayed prices do not include tax. Monthly prices shown are for country or location: United States

Plan	Features	Pricing
Lite	**Includes up to 500 registered devices, and a maximum of 200 MB of each data metric** Maximum of 500 registered devices Maximum of 500 application bindings	Free

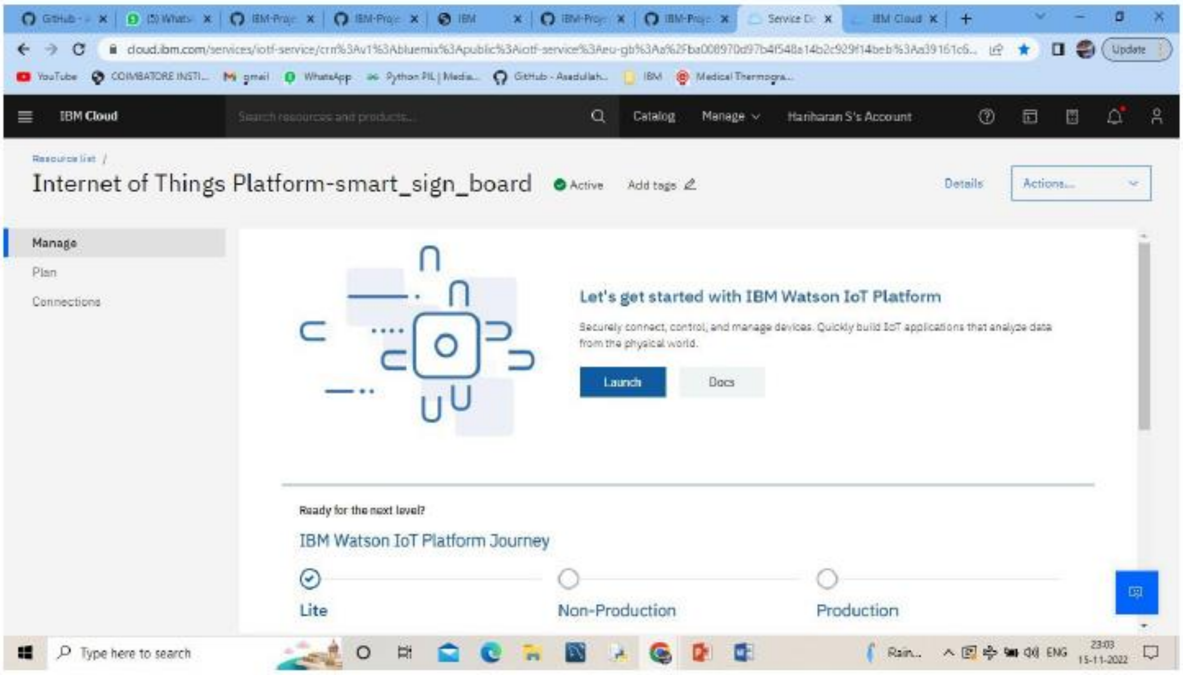
Summary

Internet of Things Platform **Free**
Location: London
Plan: Lite
Service name: Internet of Things Platform-jy
Resource group: Default

□ I have read and agree to the following license agreements:
Terms

Create

Add to estimate



cloud.ibm.com/services/iotf-service/crn%3Av1%3Abluemix%3Apublic%3Aiotf-service%3Aeu-gb%3Aa%2Fba008970d97b4f548a14b2c929f14beb%3Aa39161c6...
IBM Cloud
Search resources and products...
Catalog
Manage
Hariharan S's Account
Type here to search
23:03
15-11-2022

Resource list /

Internet of Things Platform-smart_sign_board

● Active Add tags Details Actions...

Manage
Plan
Connections

Let's get started with IBM Watson IoT Platform

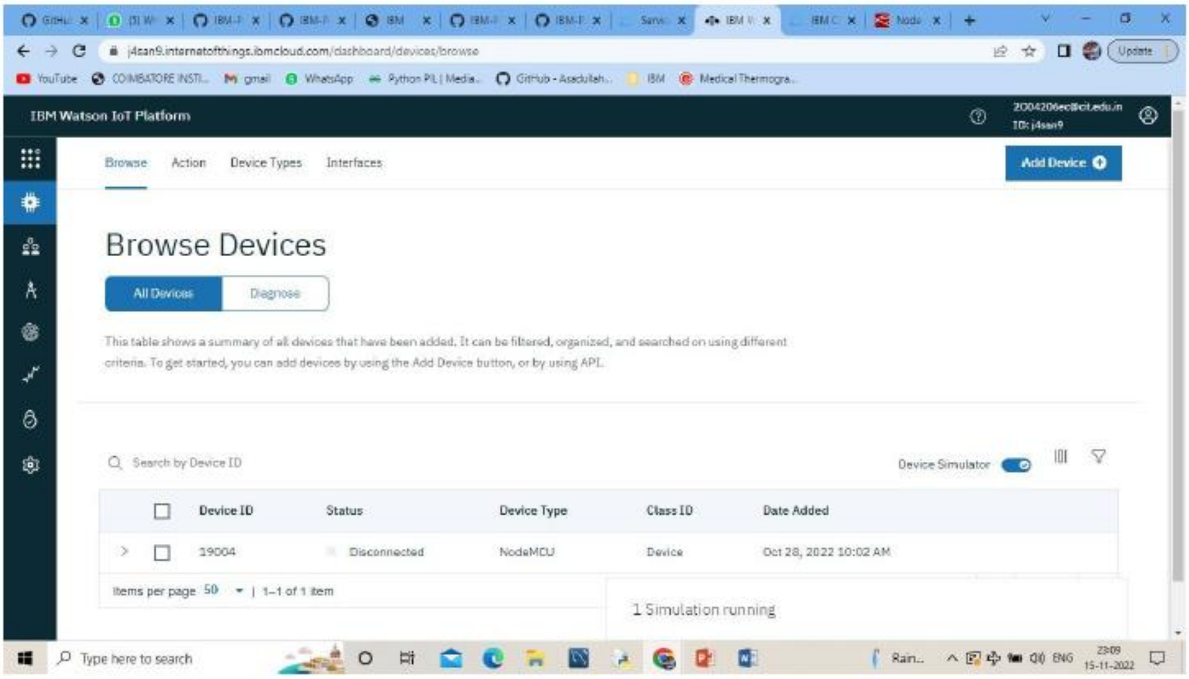
Securely connect, control, and manage devices. Quickly build IoT applications that analyze data from the physical world.

Launch Docs

Ready for the next level?

IBM Watson IoT Platform Journey

✓ Lite ○ Non-Production ○ Production



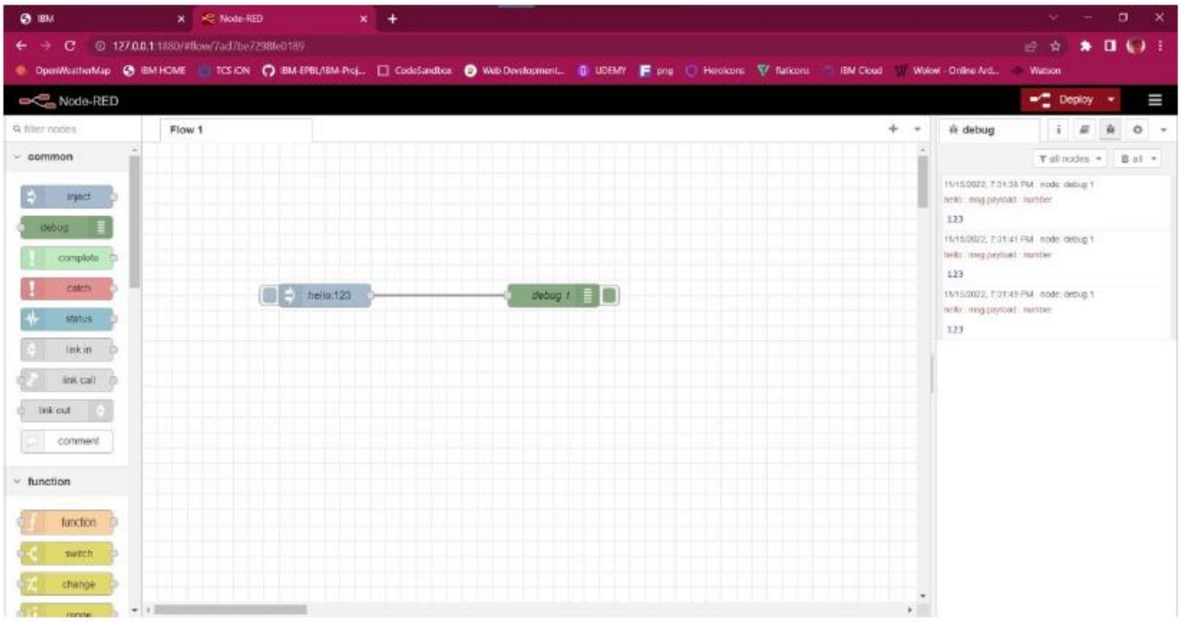
Browse Devices

This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

Device ID	Status	Device Type	Class ID	Date Added
19004	Disconnected	NodeMCU	Device	Oct 28, 2022 10:02 AM

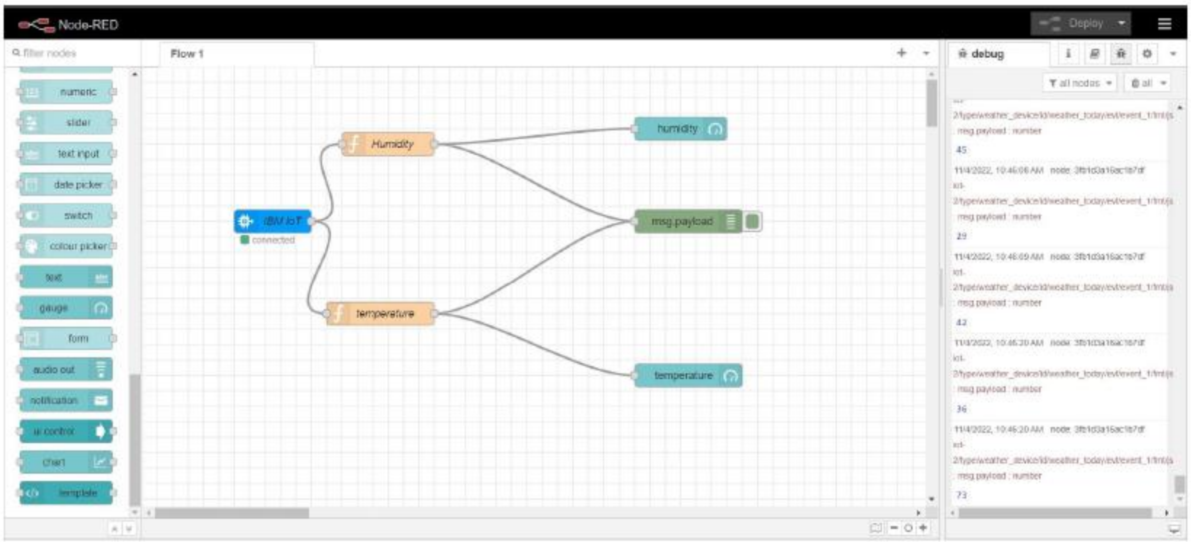
Items per page 50 | 1–1 of 1 item

1 Simulation running

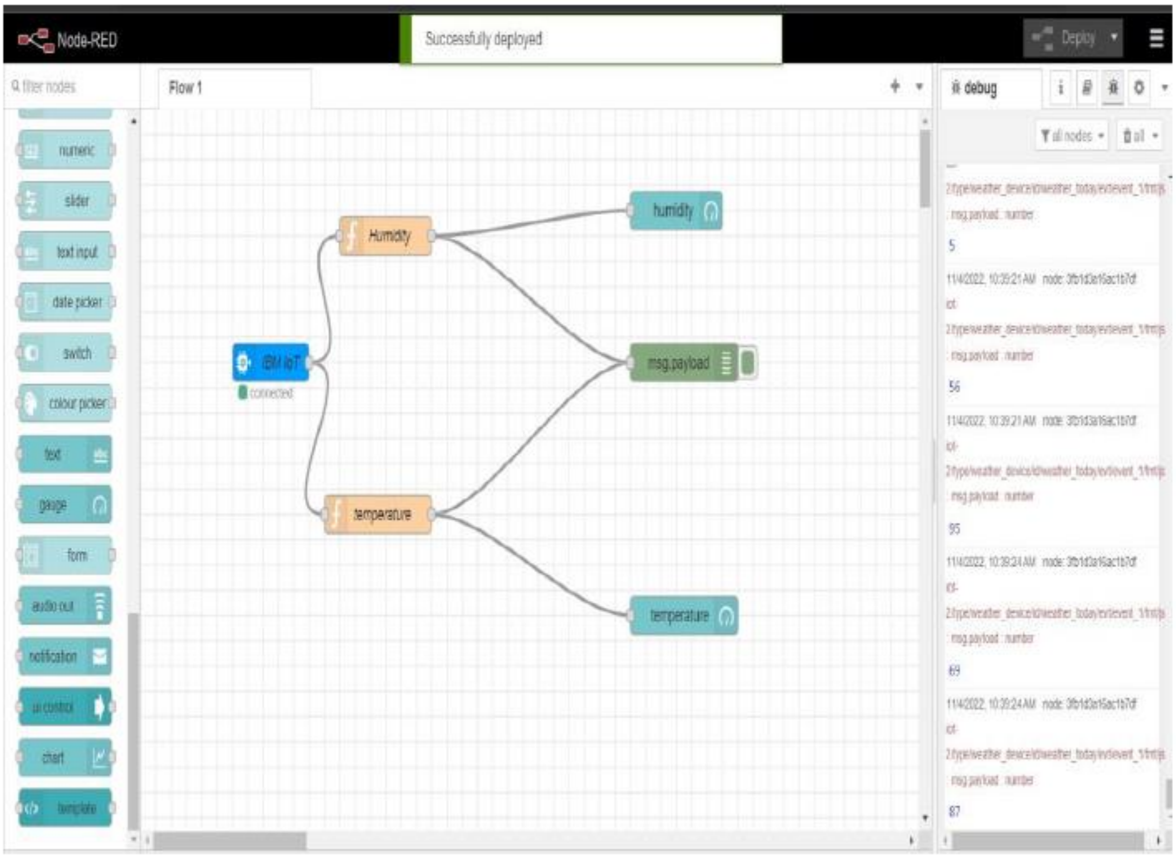


```
11/15/2022, 7:31:38 PM  node: debug 1
hello : msg.payload : number
123
11/15/2022, 7:31:41 PM  node: debug 1
hello : msg.payload : number
123
11/15/2022, 7:31:49 PM  node: debug 1
hello : msg.payload : number
123
```

The Web Application Using Node-RED



```
2/type/weather_device/id/weather_today/evt/event_1/fmt/js
: msg.payload : number
45
11/4/2022, 10:46:06 AM   node: 3fb1d3a16ac1b7df
iot-
2/type/weather_device/id/weather_today/evt/event_1/fmt/js
: msg.payload : number
29
11/4/2022, 10:46:09 AM   node: 3fb1d3a16ac1b7df
iot-
2/type/weather_device/id/weather_today/evt/event_1/fmt/js
: msg.payload : number
42
11/4/2022, 10:46:20 AM   node: 3fb1d3a16ac1b7df
iot-
2/type/weather_device/id/weather_today/evt/event_1/fmt/js
: msg.payload : number
36
11/4/2022, 10:46:20 AM   node: 3fb1d3a16ac1b7df
iot-
2/type/weather_device/id/weather_today/evt/event_1/fmt/js
: msg.payload : number
73
```



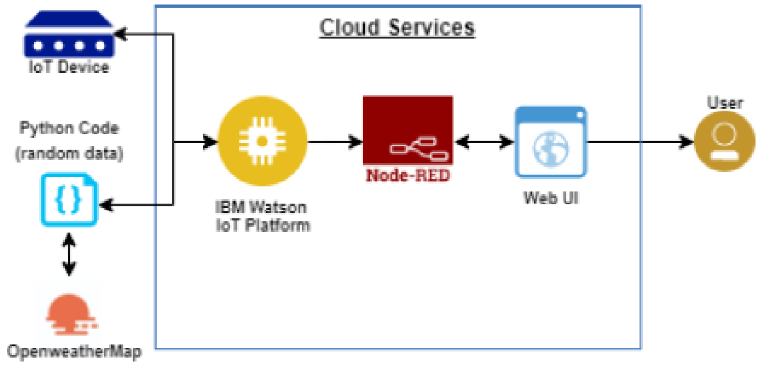
```
2/type/weather_device/id/weather_today/evt/event_1/fmt/js
: msg.payload : number
5
11/4/2022, 10:39:21 AM   node: 3fb1d3a16ac1b7df
iot-
2/type/weather_device/id/weather_today/evt/event_1/fmt/js
: msg.payload : number
56
11/4/2022, 10:39:21 AM   node: 3fb1d3a16ac1b7df
iot-
2/type/weather_device/id/weather_today/evt/event_1/fmt/js
: msg.payload : number
95
11/4/2022, 10:39:24 AM   node: 3fb1d3a16ac1b7df
iot-
2/type/weather_device/id/weather_today/evt/event_1/fmt/js
: msg.payload : number
69
11/4/2022, 10:39:24 AM   node: 3fb1d3a16ac1b7df
iot-
2/type/weather_device/id/weather_today/evt/event_1/fmt/js
: msg.payload : number
87
```

Project Design Phase-I

Proposed Solution

S.No.	Parameter	Description
1.	**Problem Statement (Problem to be solved)**	Road transport drivers find it hard to get adapted to the changes prevailing in the weather, traffic, other fatal situations and thereby applying suitable driving measures like increasing/ decreasing speeds and taking diversion measures accordingly becomes a challenging task. Therefore, smart signboards that provide timely updation are employed.
2.	**Idea / Solution description**	• Static signboards are replaced with smart connected sign boards that get the speed limitations from a web app using weather API and update automatically. • Based on the weather changes the speed may increase or decrease. • Based on the traffic and fatal situations the diversion signs are displayed. • Guide (Schools), Warning and Service (Hospitals, Restaurant) signs are also displayed accordingly.

		• Different modes of operations can be selected with the help of buttons.
3.	**Novelty / Uniqueness**	**Preceding system's objectives:** • A system that alerts drivers about road signs has been developed and tested using a smart mobile phone. • A game theoretical adversarial intervention detection mechanism for reliable smart road signs has been proposed **Proposed system's objectives:** • Receiving road sign values to the IBM IoT platform from Node-RED Web UI • Weather conditions can be viewed in the Web Application. The user will be able to monitor the prevailing changes through the app and act accordingly
4.	**Social Impact / Customer Satisfaction**	**Social Impact:** • Static signboards simply provide constant information about the speed limits, warnings in order to have a safe travel. But there might be some unexpected changes in the weather or traffic situations due to which some other path must be taken. • Such updated information regarding the roads are not provided in case of static signboards, whereas in case of smart sign boards, timely updated information regarding roads and road diversion measures are being displayed through the web app developed. This in turn, proves to be an effective tool for a safe travel **Customer Satisfaction:** The main objectives of the proposed solution are: • **Road safety and accident mitigation:** Ensures safety of the user by making them take suitable driving measures, thereby preventing the occurrence of accidents • **Time saving:** Since regularly updated information regarding roads is provided, the user can take the suitable paths and reach the destination in time

5.	Business Model (Revenue Model)	 IoT Device Cloud Services Python Code (random data) IBM Watson IoT Platform Node-RED Web UI User OpenweatherMap **Description of technical architecture:** • Changing weather conditions are received through the web app using weather API and are displayed on the smart signboards • Warnings regarding lowering speeds at the location of hospitals and schools are displayed on the smart sign board • Based on traffic and fatal situations, diversion signs are also displayed
6.	Scalability of the Solution	• In addition to displaying suitable diversion and weather change signs, these boards can be made to display some other extra features like finding the shortest path in reaching a destination and displaying it, thereby providing much more help to the user • These effective smart signboards could be implemented throughout the entire nation along the most important roadways, thereby users will find it much more useful and feel safe when travelling through those roads and saving their time also.

Solution Architecture:

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- **Find the best tech solution to solve existing business problems.**
- **Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.**

- **Define features, development phases, and solution requirements.**
- **Provide specifications according to which the solution is defined, managed, and delivered.**

Example - Solution Architecture Diagram:

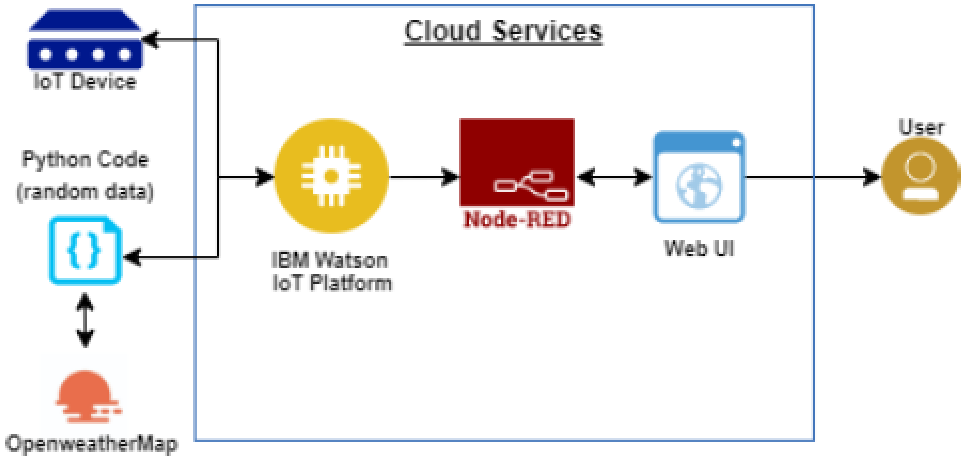
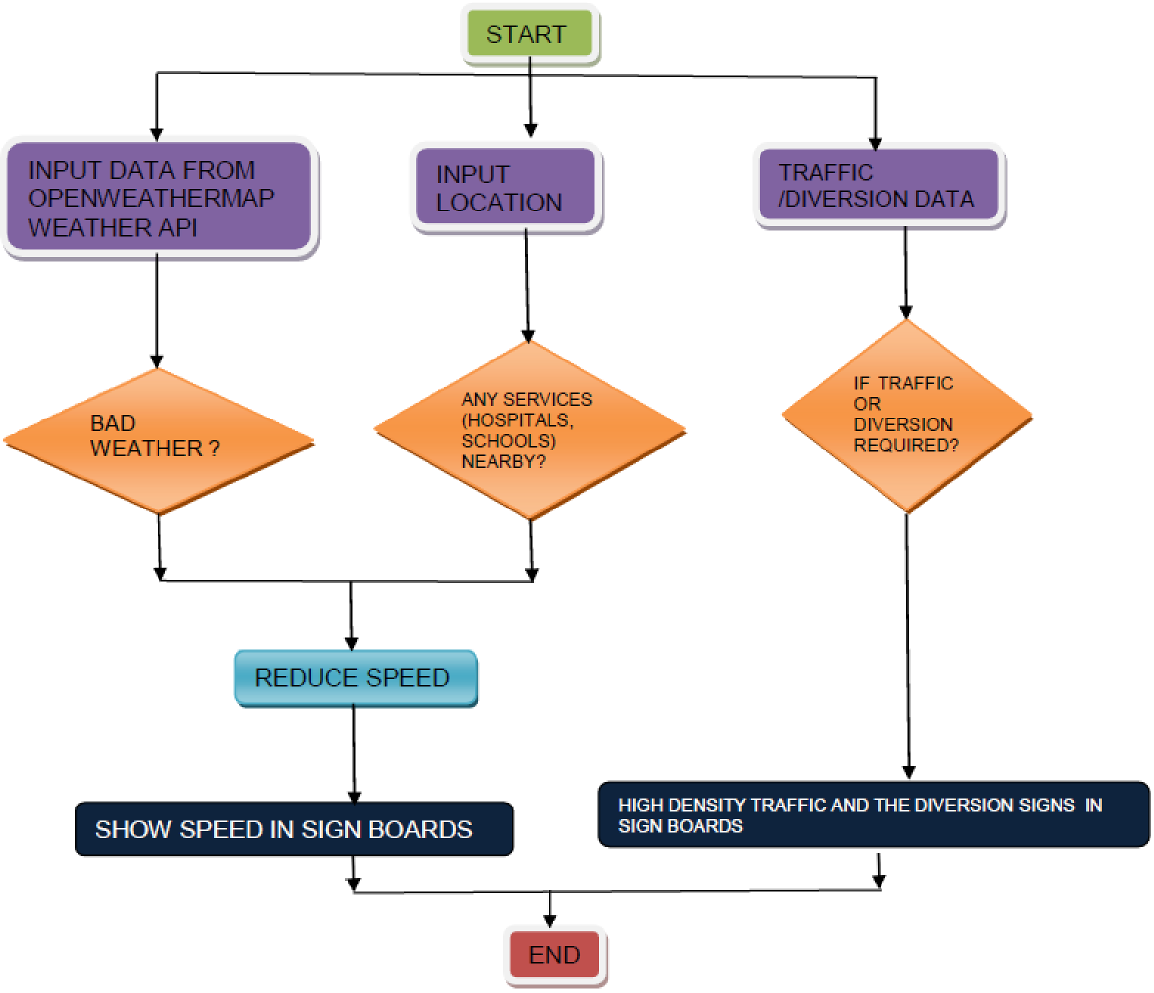


Figure 1: Architecture and data flow of the smart connectivity for better road safety

CUSTOMER JOURNEY MAP

SCENARIO	ENTICE	ENTER	ENGAGE	EXIT	EXTEND
STEPS	Taking action and concerning for road safety.	Selecting an efficient product in order to improve the road safety.	Other products are available including static boards.	Smart boards are more efficient than static boards.	Following product satisfaction.
INTERACTIONS	Buyers are delighted.	The government need not worry about traffic safety after the installation.	The numerous varieties of products are accessible and it entertains the user.	The government will not be concerned about the safety after receiving this.	The government acquires the product after determining its worth.
GOALS & MOTIVATIONS	Help me understand what this new technology is all about.	Help me make sure people don’t forget about this technology so that they don’t get disappointed.	Help me make the most of my travel within a short pace.	Help me spread the world about this technology and provide feedbacks.	Help me see what I’ve gained from old technology to new technology.
POSITIVE MOMENTS	The functioning of this technology provided by the technical team is exciting to see.	Excited about the new technology.	If any technical error happens, it can be rectified in advance.	People will love the usage of smart sign boards, we have a 90% satisfaction rate.	The customers feel delighted to find it helpful for their safe journey.
NEGATIVE MOMENTS	The government express a bit of fear while introducing a new technology.	Trepidation about the product purchase.	Sometimes people may take long time to reach their destination because of internet issues.	People describe leaving a review as an arduous process.	-
AREAS OF OPPURTUNITY	The customer is benefited from increased road safety.	The customer is aware of the product’s manufacturing process.	Other products will be made known to the customer.	The buyer learns which product is the best.	The travel will be safe for the customer.

Data Flow Diagram:


START
INPUT DATA FROM OPENWEATHERMAP WEATHER API
INPUT LOCATION
TRAFFIC /DIVERSION DATA
BAD WEATHER ?
ANY SERVICES (HOSPITALS, SCHOOLS) NEARBY?
IF TRAFFIC OR DIVERSION REQUIRED?
REDUCE SPEED
SHOW SPEED IN SIGN BOARDS
HIGH DENSITY TRAFFIC AND THE DIVERSION SIGNS IN SIGN BOARDS
END

Project Design Phase-II

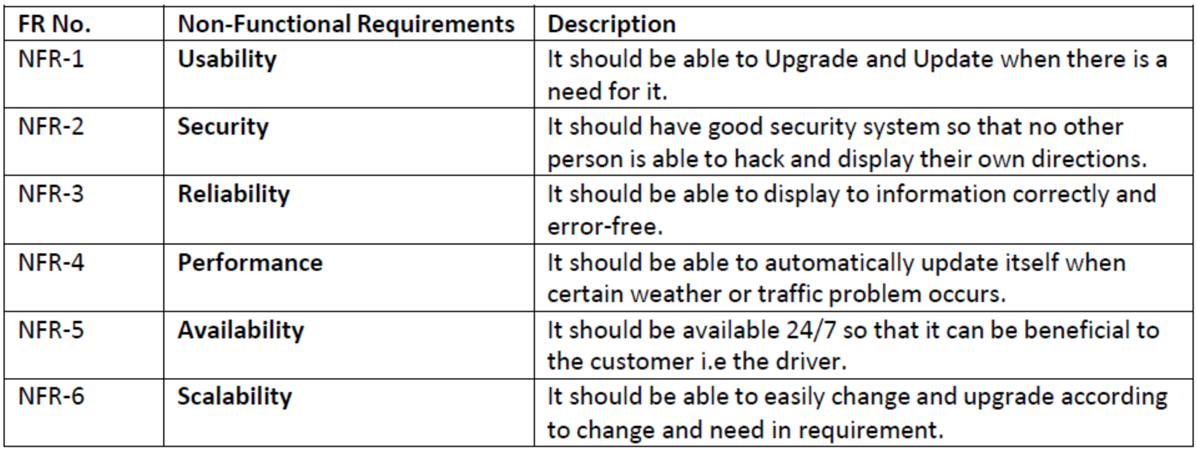
Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement(Epic)	Sub Requirement(Story/Sub-Task)
FR-1	User Visibility	Sign Boards should be made with LED's which are bright colored and are capable of attracting the drivers attention but it should also not be too distracting or blinding cause it may lead to accidents.
FR-2	User Understanding	For better understanding of the driver, the signs should be big, clear and legible and it can also include illustrations which will make it easily understandable to the driver.
FR-3	User Convenience	The display should be big enough that it should even be visible from far distance clearly.

Non-Functional Requirements:

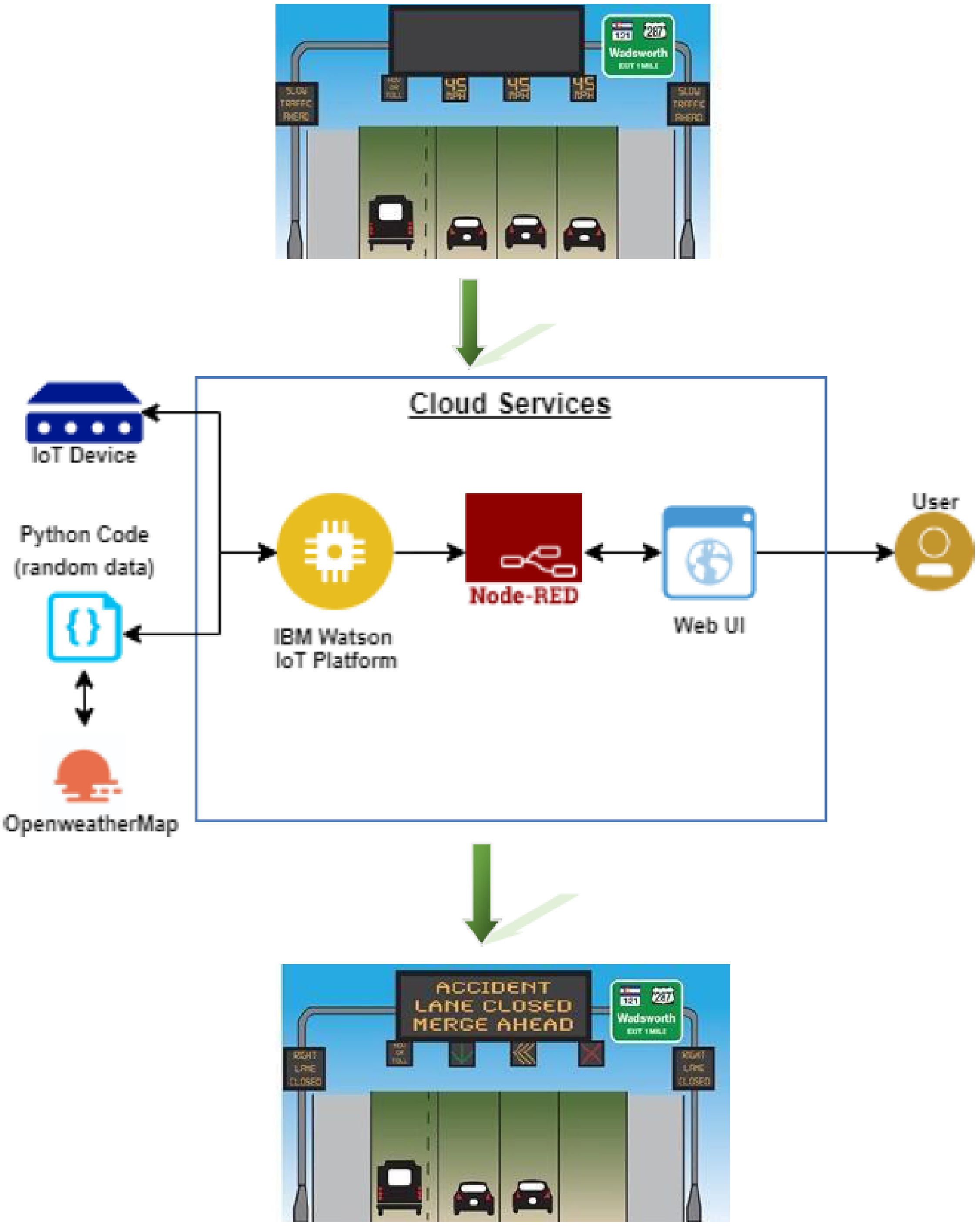
Following are the Non-Functional Requirements of the proposed solution



FR No.	Non-Functional Requirements	Description
NFR-1	**Usability**	It should be able to Upgrade and Update when there is a need for it.
NFR-2	**Security**	It should have good security system so that no other person is able to hack and display their own directions.
NFR-3	**Reliability**	It should be able to display to information correctly and error-free.
NFR-4	**Performance**	It should be able to automatically update itself when certain weather or traffic problem occurs.
NFR-5	**Availability**	It should be available 24/7 so that it can be beneficial to the customer i.e the driver.
NFR-6	**Scalability**	It should be able to easily change and upgrade according to change and need in requirement.

Technical Architecture:

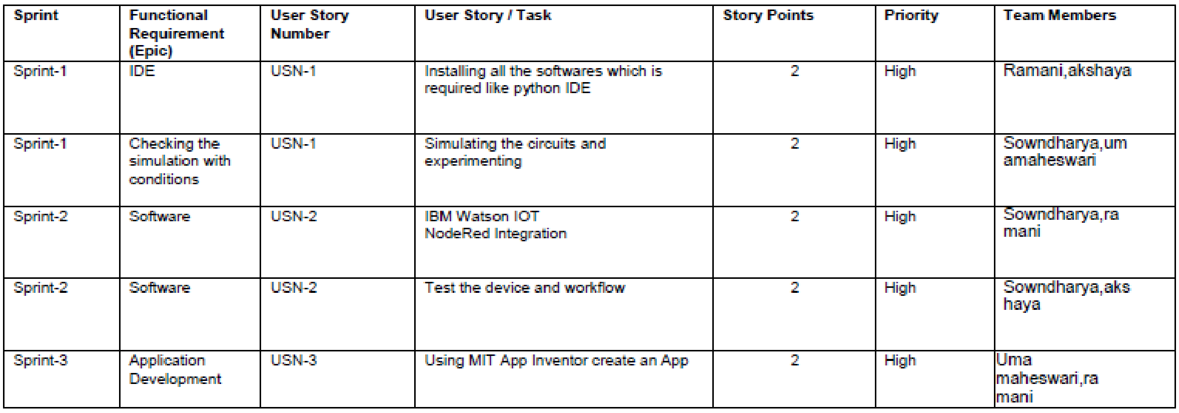
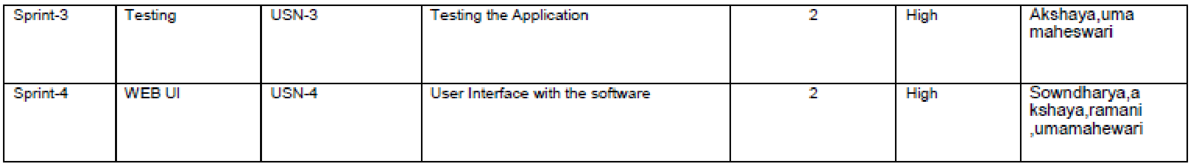
The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2



Project Planning Phase

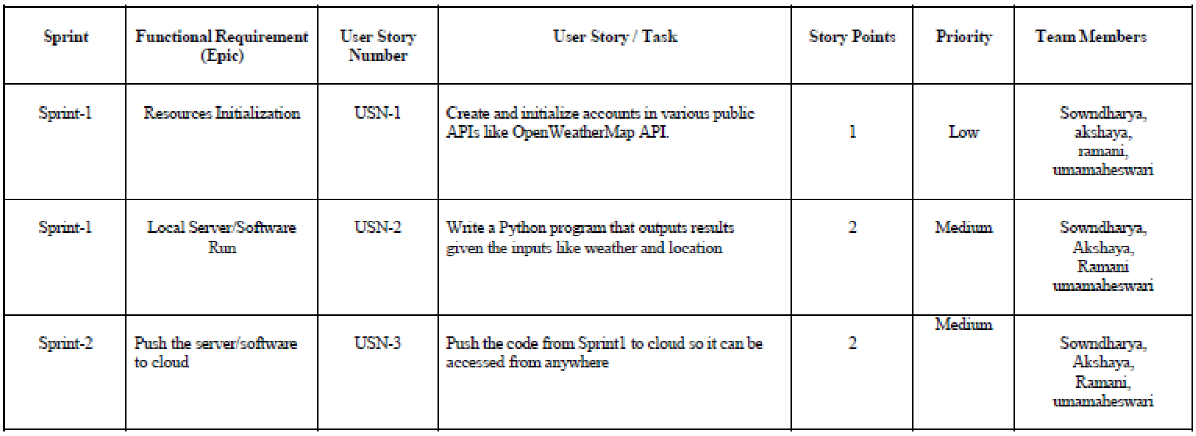
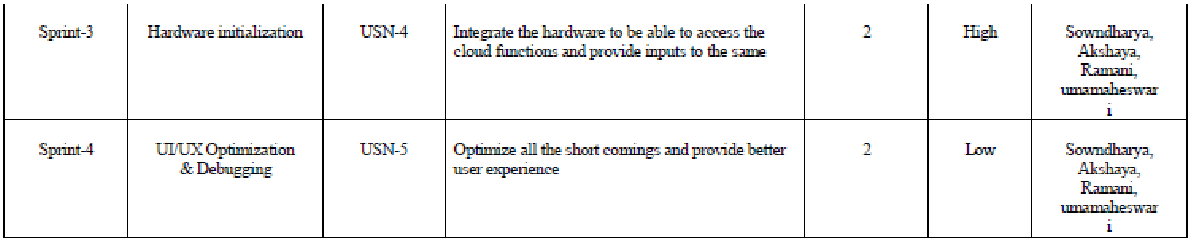
Project Planning Template (Milestone and Activity List)

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	IDE	USN-1	Installing all the softwares which is required like python IDE	2	High	Ramani,akshaya
Sprint-1	Checking the simulation with conditions	USN-1	Simulating the circuits and experimenting	2	High	Sowndharya,umamaheswari
Sprint-2	Software	USN-2	IBM Watson IOT NodeRed Integration	2	High	Sowndharya,ramani
Sprint-2	Software	USN-2	Test the device and workflow	2	High	Sowndharya,akshaya
Sprint-3	Application Development	USN-3	Using MIT App Inventor create an App	2	High	Uma maheswari,ramani
Sprint-3	Testing	USN-3	Testing the Application	2	High	Akshaya,uma maheswari
Sprint-4	WEB UI	USN-4	User Interface with the software	2	High	Sowndharya,akshaya,ramani ,umamahewari

Product Backlog, Sprint Schedule, and Estimation

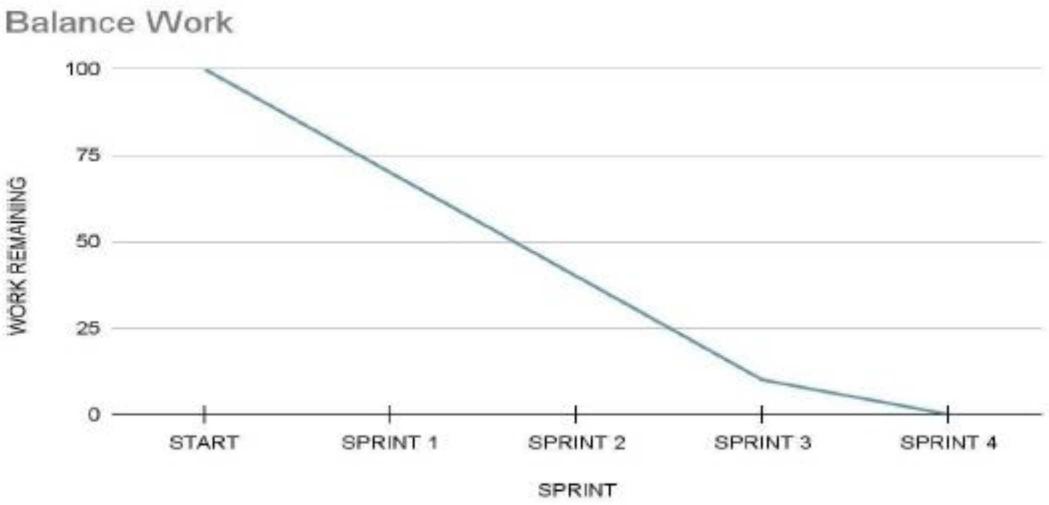
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Resources Initialization	USN-1	Create and initialize accounts in various public APIs like OpenWeatherMap API.	1	Low	Sowndharya, akshaya, ramani, umamaheswari
Sprint-1	Local Server/Software Run	USN-2	Write a Python program that outputs results given the inputs like weather and location	2	Medium	Sowndharya, Akshaya, Ramani umamaheswari
Sprint-2	Push the server/software to cloud	USN-3	Push the code from Sprint1 to cloud so it can be accessed from anywhere	2	Medium	Sowndharya, Akshaya, Ramani, umamaheswari
Sprint-3	Hardware initialization	USN-4	Integrate the hardware to be able to access the cloud functions and provide inputs to the same	2	High	Sowndharya, Akshaya, Ramani, umamaheswari
Sprint-4	UI/UX Optimization & Debugging	USN-5	Optimize all the short comings and provide better user experience	2	Low	Sowndharya, Akshaya, Ramani, umamaheswari

Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

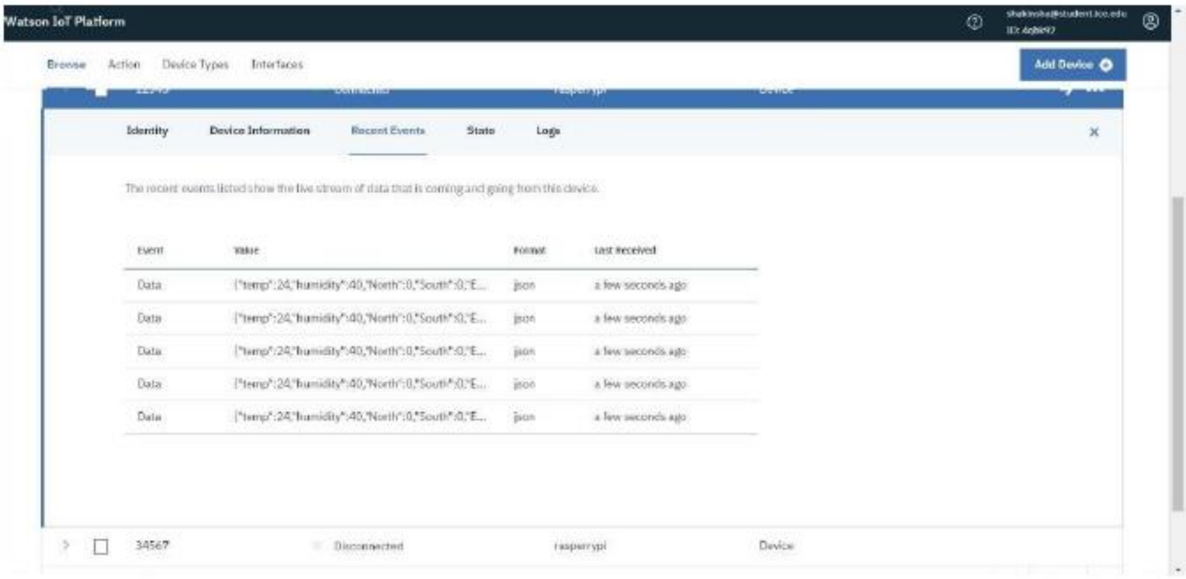
Burndown Chart:



Project Development Phase

Sprint 1

IoT Device – IoT Platform

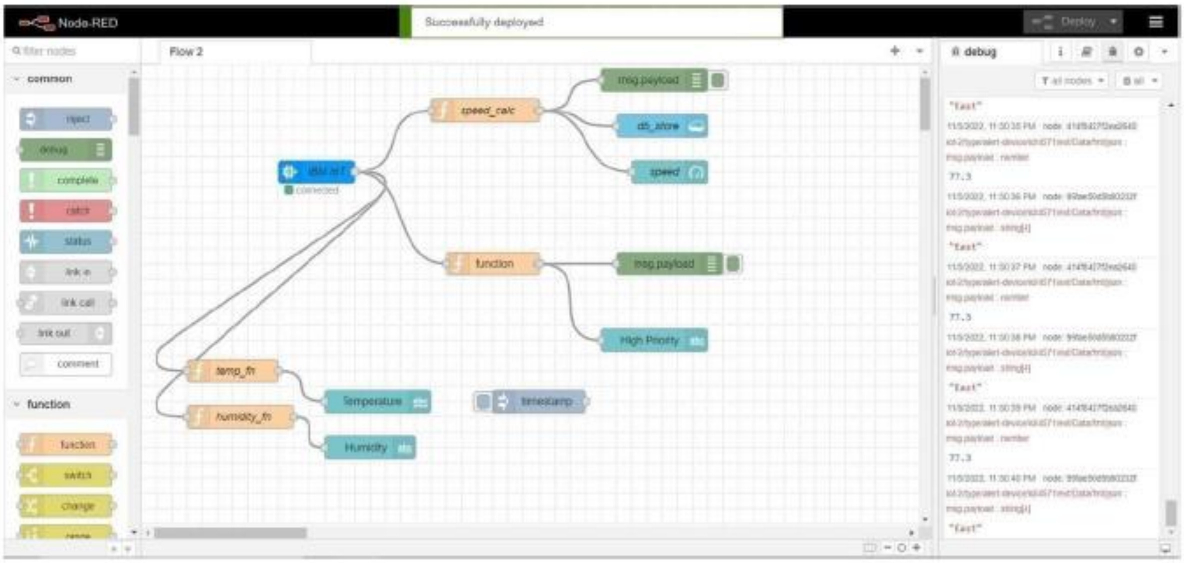


The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
Data	{"temp":24,"humidity":40,"North":0,"South":0,"E...	json	a few seconds ago
Data	{"temp":24,"humidity":40,"North":0,"South":0,"E...	json	a few seconds ago
Data	{"temp":24,"humidity":40,"North":0,"South":0,"E...	json	a few seconds ago
Data	{"temp":24,"humidity":40,"North":0,"South":0,"E...	json	a few seconds ago
Data	{"temp":24,"humidity":40,"North":0,"South":0,"E...	json	a few seconds ago

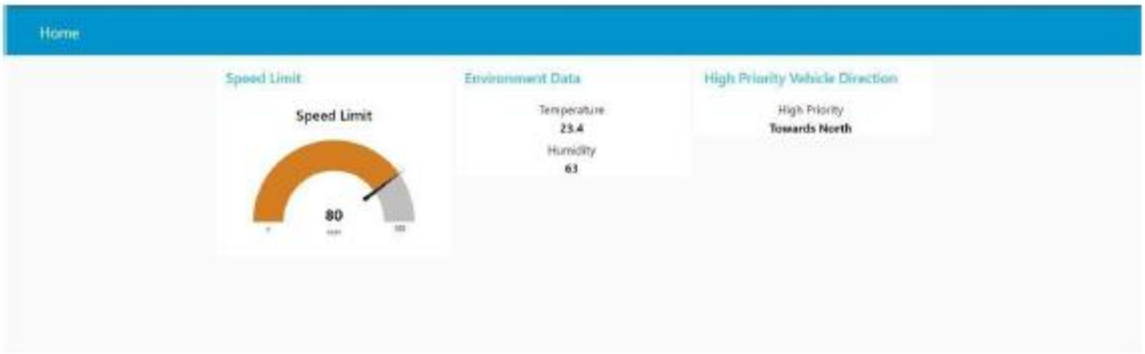
□ 34567 Disconnected raspberrypi Device

Node Red



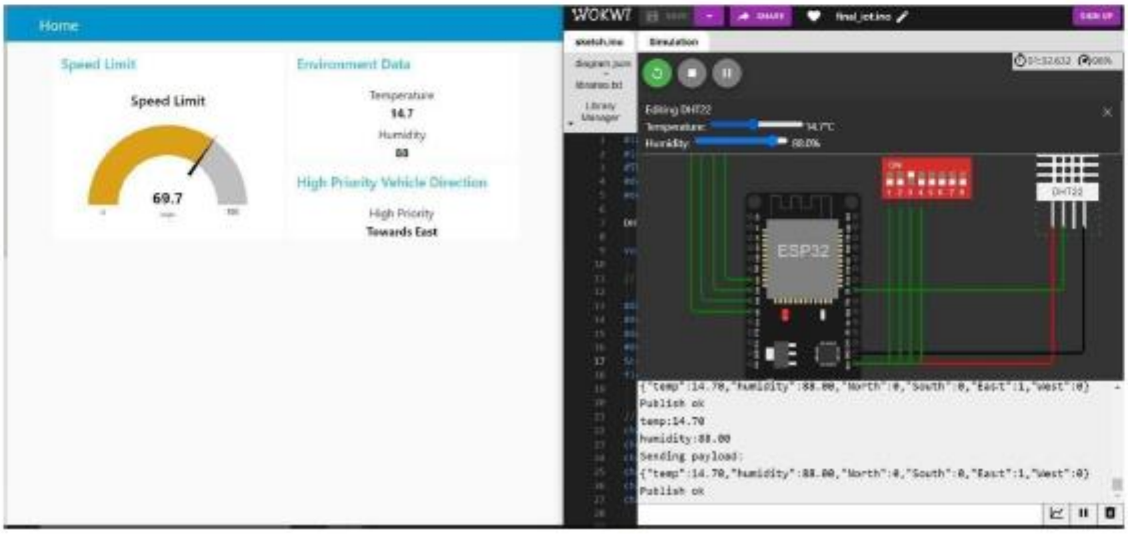
```
"East"
11/5/2022, 11:50:35 PM  node: [illegible]
iot-2/type/alert-device/id/[illegible]/evt/Data/fmt/json :
msg.payload : number
77.3
11/5/2022, 11:50:36 PM  node: [illegible]
iot-2/type/alert-device/id/[illegible]/evt/Data/fmt/json :
msg.payload : string[4]
"East"
11/5/2022, 11:50:37 PM  node: [illegible]
iot-2/type/alert-device/id/[illegible]/evt/Data/fmt/json :
msg.payload : number
77.3
11/5/2022, 11:50:38 PM  node: [illegible]
iot-2/type/alert-device/id/[illegible]/evt/Data/fmt/json :
msg.payload : string[4]
"East"
11/5/2022, 11:50:39 PM  node: [illegible]
iot-2/type/alert-device/id/[illegible]/evt/Data/fmt/json :
msg.payload : number
77.3
11/5/2022, 11:50:40 PM  node: [illegible]
iot-2/type/alert-device/id/[illegible]/evt/Data/fmt/json :
msg.payload : string[4]
"East"
```

Node Red Web UI



Temperature
23.4
Humidity
63

High Priority
Towards North

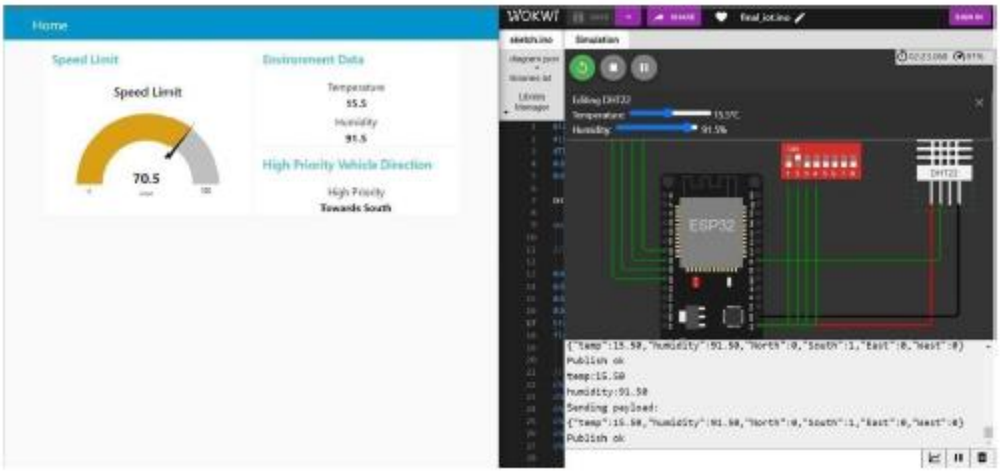


Temperature
14.7
Humidity
88

High Priority
Towards East

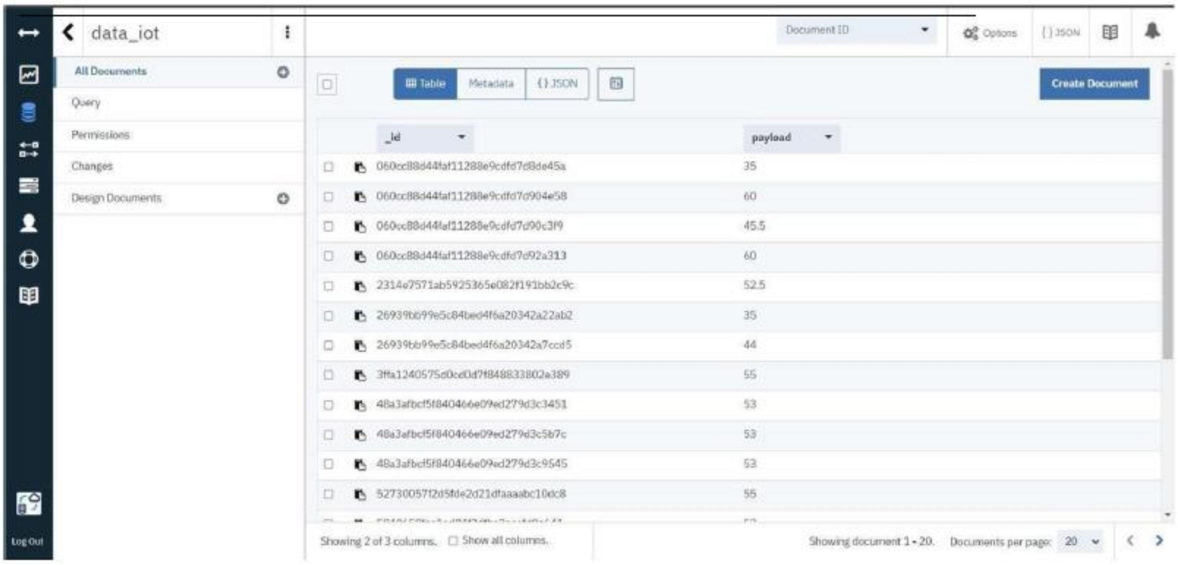
Editing DHT22
Temperature: 14.7°C
Humidity: 88.0%

```
{"temp":14.70,"humidity":88.00,"North":0,"South":0,"East":1,"West":0}
Publish ok
temp:14.70
humidity:88.00
Sending payload:
{"temp":14.70,"humidity":88.00,"North":0,"South":0,"East":1,"West":0}
Publish ok
```



```
{"temp":15.50,"humidity":91.50,"North":0,"South":1,"East":0,"West":0}
Publish ok
temp:15.50
humidity:91.50
Sending payload:
{"temp":15.50,"humidity":91.50,"North":0,"South":1,"East":0,"West":0}
Publish ok
```

Cloudant Database



_id	payload
060cc88d44faf11288e9cdfd7d8de45a	35
060cc88d44faf11288e9cdfd7d904e58	60
060cc88d44faf11288e9cdfd7d90c3f9	45.5
060cc88d44faf11288e9cdfd7d92a313	60
2314e7571ab5925365e082f191bb2c9c	52.5
26939bb99e5c84bed4f6a20342a22ab2	35
26939bb99e5c84bed4f6a20342a7ccd5	44
3ffa1240575d0cd0d7f848833802e389	55
48a3afbcf5f840466e09ed279d3c3451	53
48a3afbcf5f840466e09ed279d3c5b7c	53
48a3afbcf5f840466e09ed279d3c9545	53
5273005712d5fde2d21dfaaabc10dc8	55

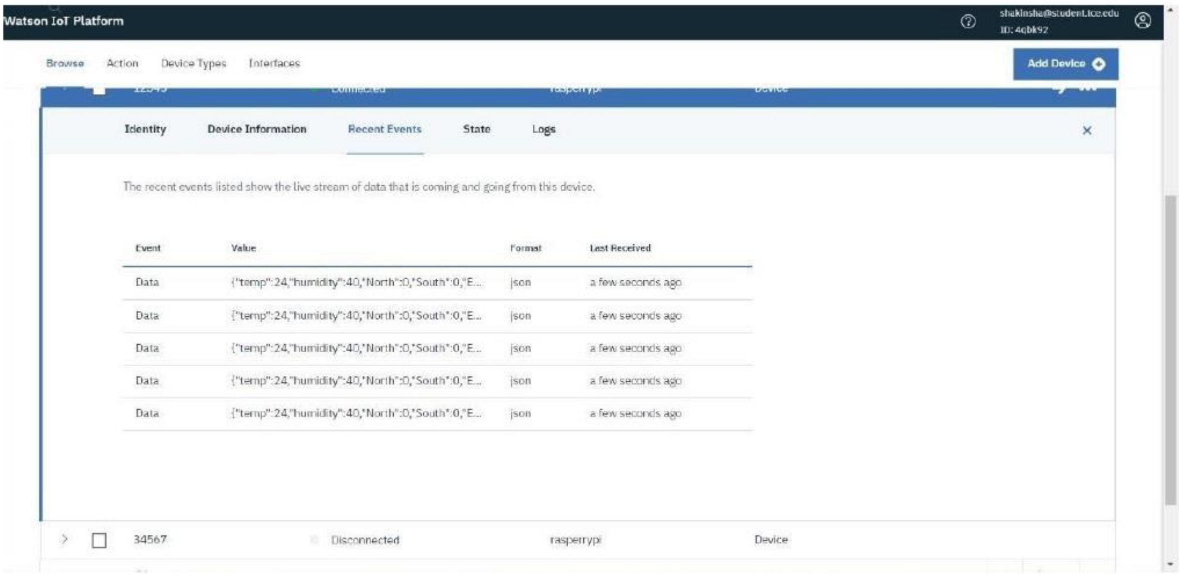
Showing 2 of 3 columns. □ Show all columns.

Showing document 1 - 20. Documents per page: 20

Project Development Phase

Sprint II

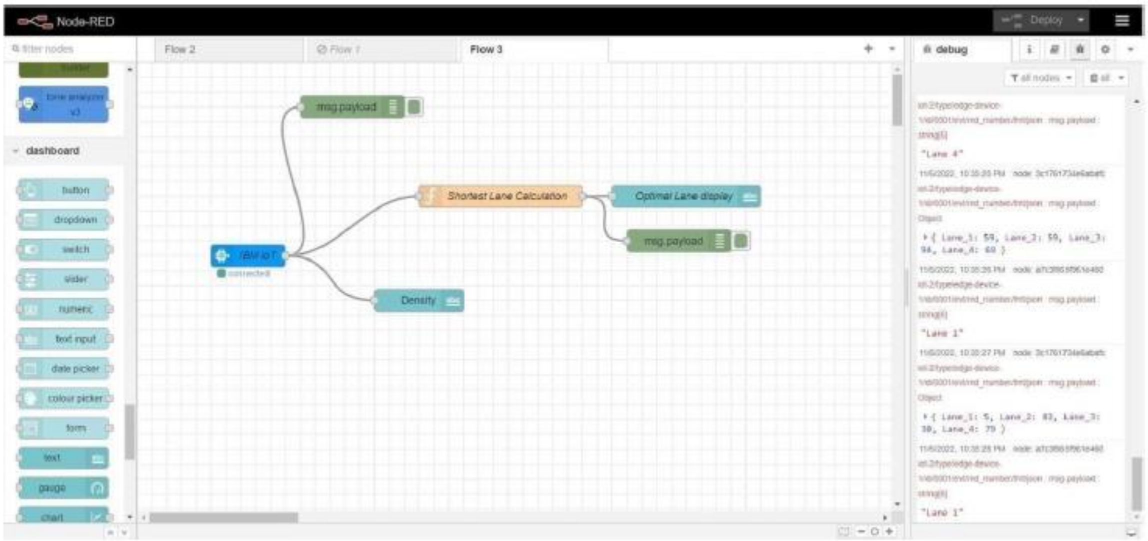
IoT Device – IoT Platform



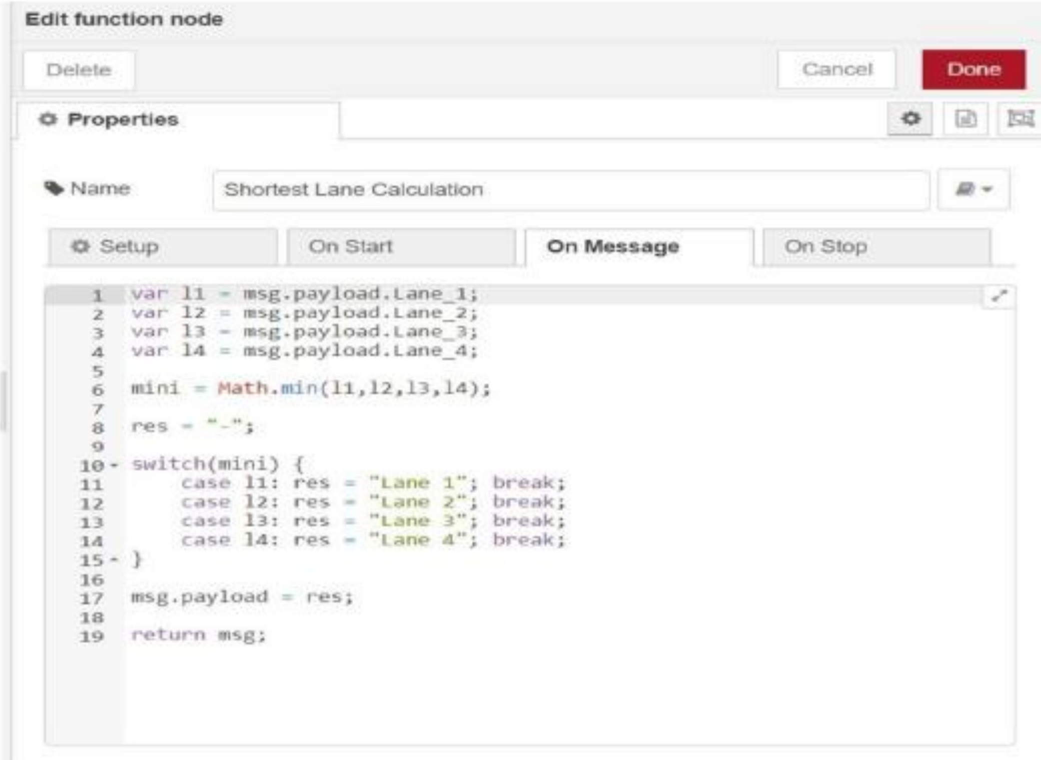
The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
Data	{"temp":24,"humidity":40,"North":0,"South":0,"E...	json	a few seconds ago
Data	{"temp":24,"humidity":40,"North":0,"South":0,"E...	json	a few seconds ago
Data	{"temp":24,"humidity":40,"North":0,"South":0,"E...	json	a few seconds ago
Data	{"temp":24,"humidity":40,"North":0,"South":0,"E...	json	a few seconds ago
Data	{"temp":24,"humidity":40,"North":0,"South":0,"E...	json	a few seconds ago

Node Red



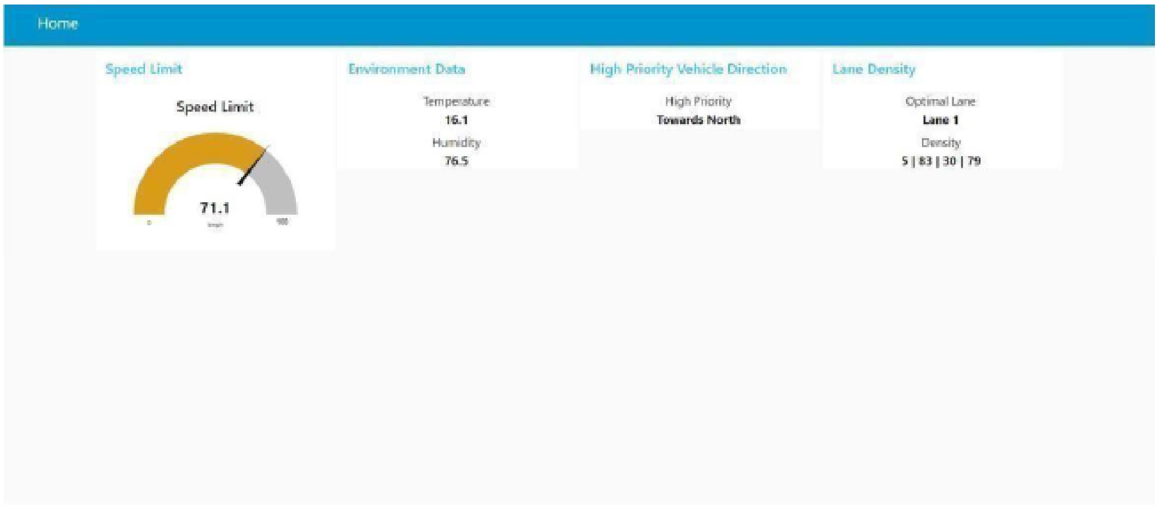
```
iot-2/type/edge-device-1/id/0001/evt/rnd_number/fmt/json : msg.payload : string[6]
"Lane 4"
11/6/2022, 10:35:25 PM  node: 3c176173a4e6abafc
iot-2/type/edge-device-1/id/0001/evt/rnd_number/fmt/json : msg.payload : Object
▸{ Lane_1: 59, Lane_2: 59, Lane_3: 94, Lane_4: 69 }
11/6/2022, 10:35:26 PM  node: a7c3f6659561e480
iot-2/type/edge-device-1/id/0001/evt/rnd_number/fmt/json : msg.payload : string[6]
"Lane 1"
11/6/2022, 10:35:27 PM  node: 3c176173a4e6abafc
iot-2/type/edge-device-1/id/0001/evt/rnd_number/fmt/json : msg.payload : Object
▸{ Lane_1: 5, Lane_2: 83, Lane_3: 30, Lane_4: 79 }
11/6/2022, 10:35:28 PM  node: a7c3f6659561e480
iot-2/type/edge-device-1/id/0001/evt/rnd_number/fmt/json : msg.payload : string[6]
"Lane 1"
```



Name: Shortest Lane Calculation

```
var l1 = msg.payload.Lane_1;
var l2 = msg.payload.Lane_2;
var l3 = msg.payload.Lane_3;
var l4 = msg.payload.Lane_4;

mini = Math.min(l1,l2,l3,l4);

res = "-";

switch(mini) {
     case l1: res = "Lane 1"; break;
     case l2: res = "Lane 2"; break;
     case l3: res = "Lane 3"; break;
     case l4: res = "Lane 4"; break;
}

msg.payload = res;

return msg;
```

Node Red Web UI



Temperature
16.1
Humidity
76.5

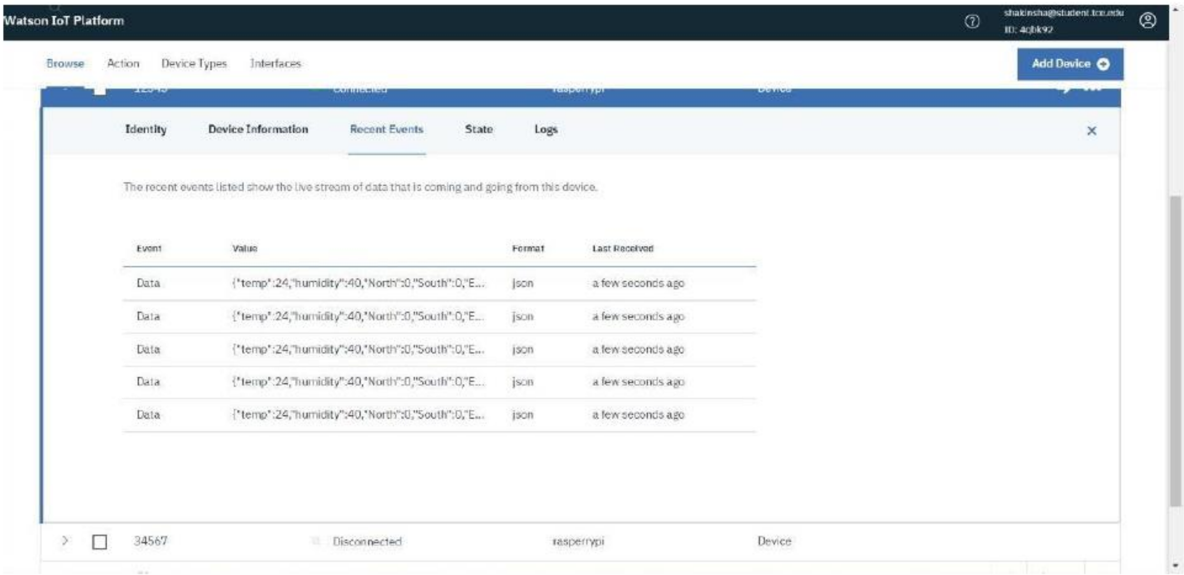
High Priority
Towards North

Optimal Lane
Lane 1
Density
5 | 83 | 30 | 79

Project Development Phase

Sprint III

IoT Device – IoT Platform

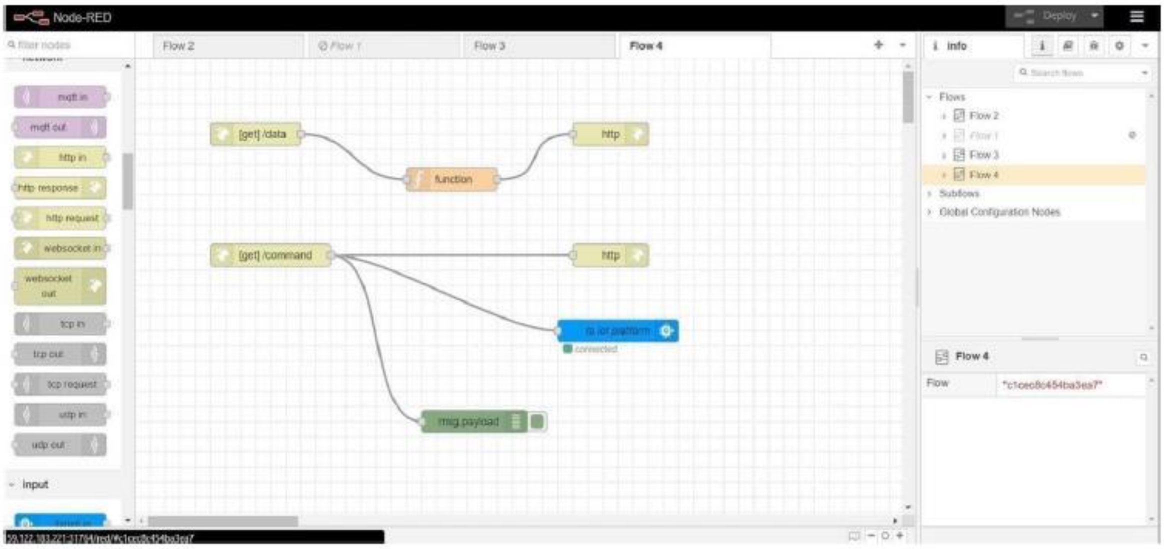


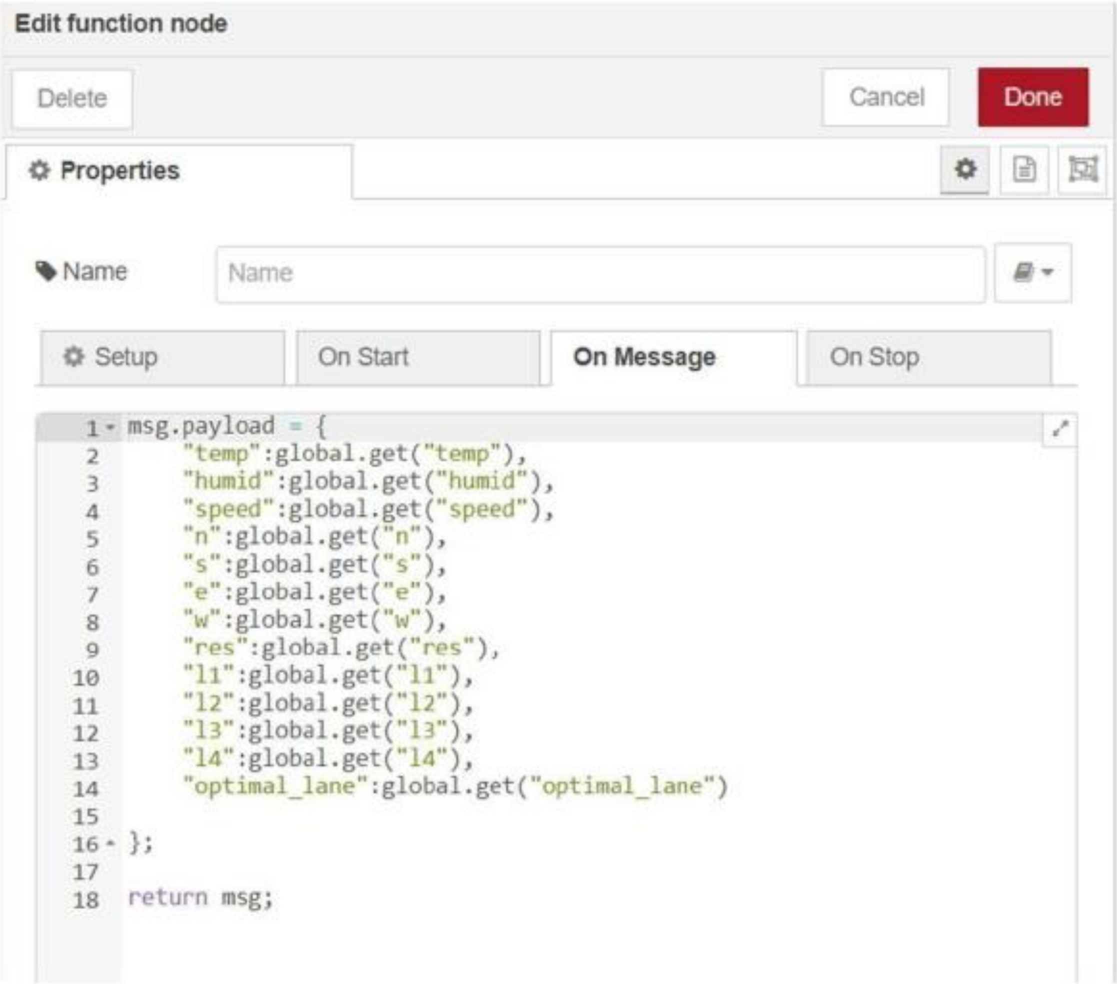
The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
Data	{"temp":24,"humidity":40,"North":0,"South":0,"E...	json	a few seconds ago
Data	{"temp":24,"humidity":40,"North":0,"South":0,"E...	json	a few seconds ago
Data	{"temp":24,"humidity":40,"North":0,"South":0,"E...	json	a few seconds ago
Data	{"temp":24,"humidity":40,"North":0,"South":0,"E...	json	a few seconds ago
Data	{"temp":24,"humidity":40,"North":0,"South":0,"E...	json	a few seconds ago

□	34567	Disconnected	rasperrypi	Device

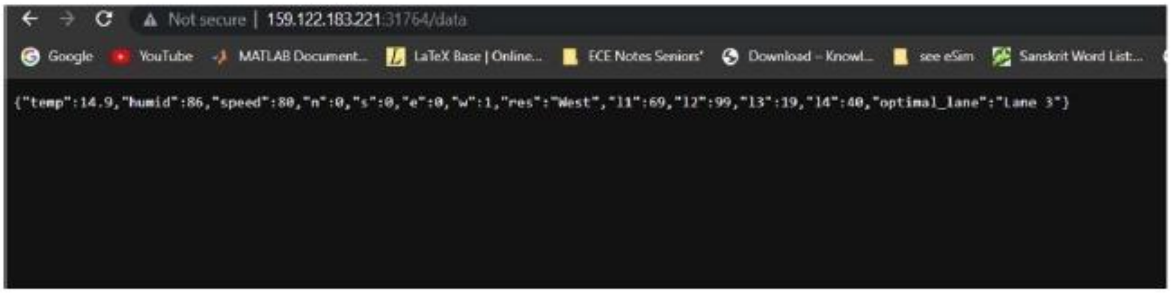
Node Red – Connect with MIT AppInventor





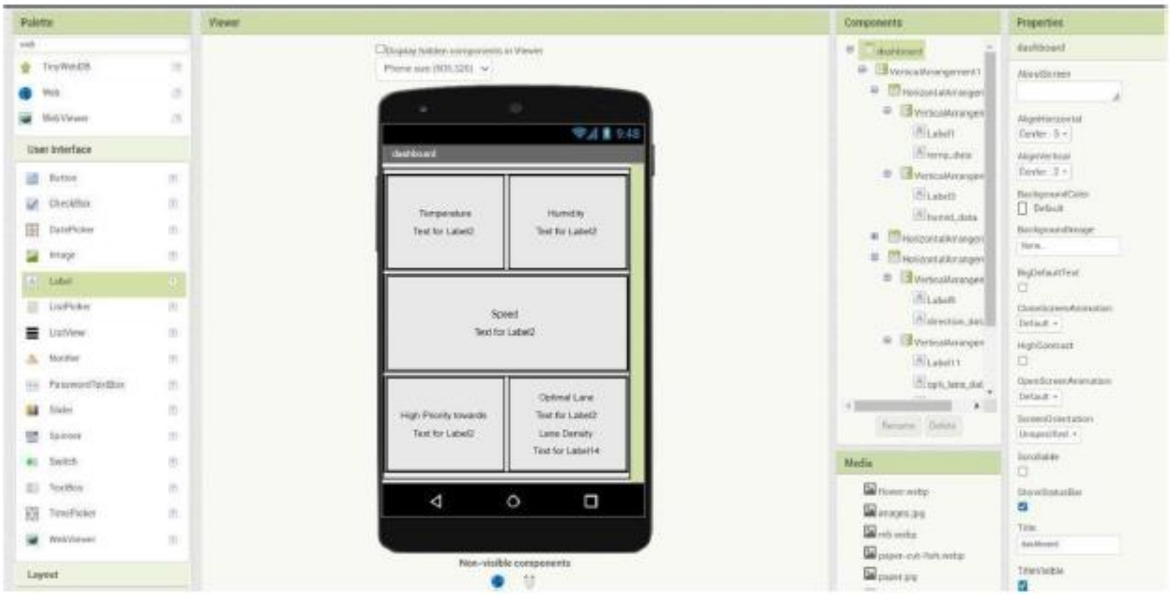
```
msg.payload = {
    "temp":global.get("temp"),
    "humid":global.get("humid"),
    "speed":global.get("speed"),
    "n":global.get("n"),
    "s":global.get("s"),
    "e":global.get("e"),
    "w":global.get("w"),
    "res":global.get("res"),
    "l1":global.get("l1"),
    "l2":global.get("l2"),
    "l3":global.get("l3"),
    "l4":global.get("l4"),
    "optimal_lane":global.get("optimal_lane")

};

return msg;
```

Output from Node red:

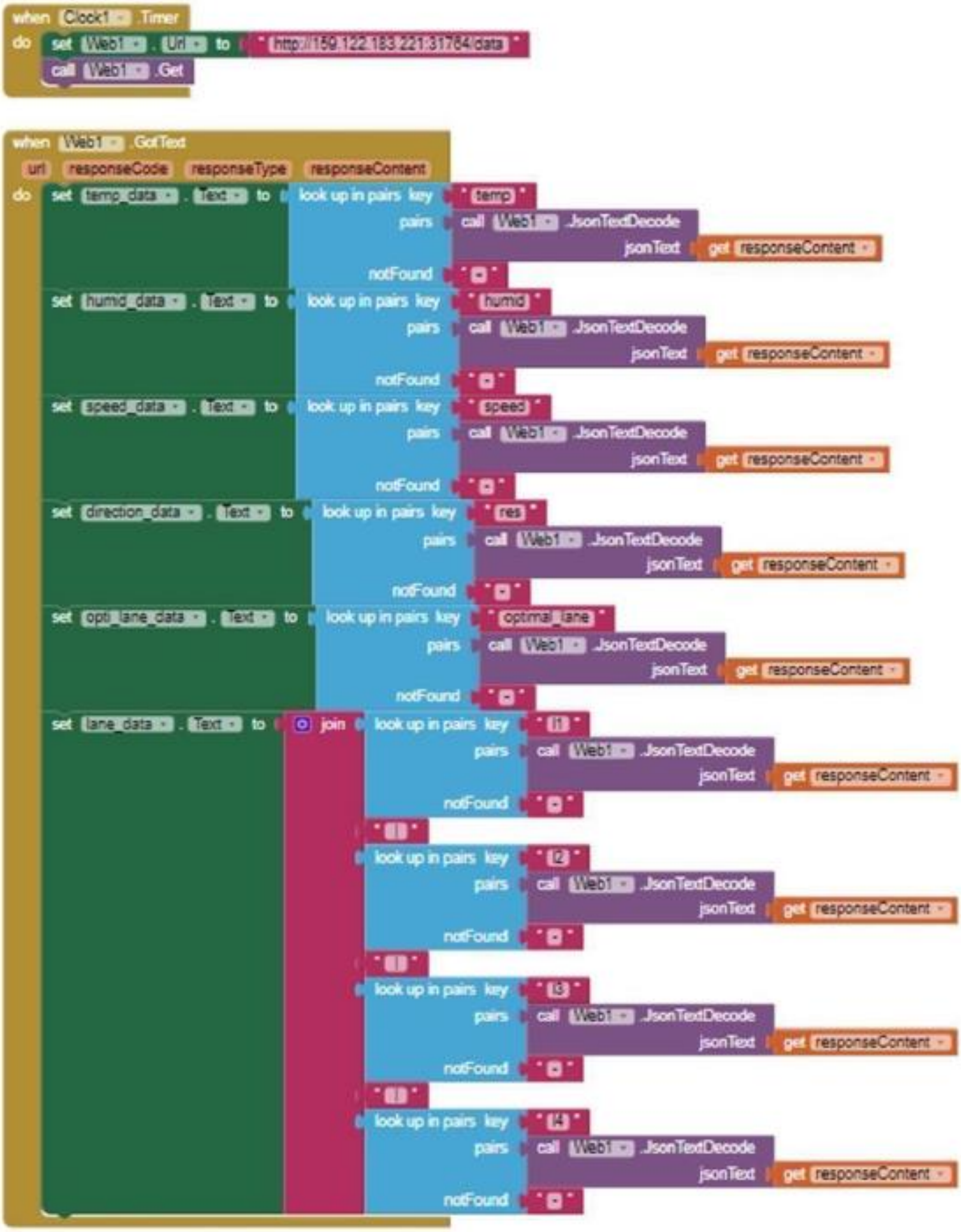


```
{"temp":14.9,"humid":86,"speed":80,"n":0,"s":0,"e":0,"w":1,"res":"West","l1":69,"l2":99,"l3":19,"l4":40,"optimal_lane":"Lane 3"}
```

MIT App Inventor UI design:



MIT App Inventor Backend design:



```
when Clock1 .Timer
do  set Web1 . Url to " http://159.122.183.221:31764/data "
    call Web1 .Get

when Web1 .GotText
  url  responseCode  responseType  responseContent
do  set temp_data . Text to  look up in pairs  key " temp "
                                               pairs  call Web1 .JsonTextDecode
                                                                    jsonText  get responseContent
                                            notFound " - "
    set humid_data . Text to  look up in pairs  key " humid "
                                               pairs  call Web1 .JsonTextDecode
                                                                    jsonText  get responseContent
                                            notFound " - "
    set speed_data . Text to  look up in pairs  key " speed "
                                               pairs  call Web1 .JsonTextDecode
                                                                    jsonText  get responseContent
                                            notFound " - "
    set direction_data . Text to  look up in pairs  key " res "
                                                   pairs  call Web1 .JsonTextDecode
                                                                        jsonText  get responseContent
                                                notFound " - "
    set opti_lane_data . Text to  look up in pairs  key " optimal_lane "
                                                   pairs  call Web1 .JsonTextDecode
                                                                        jsonText  get responseContent
                                                notFound " - "
    set lane_data . Text to  join  look up in pairs  key " l1 "
                                                    pairs  call Web1 .JsonTextDecode
                                                                         jsonText  get responseContent
                                                 notFound " - "
                                   " | "
                                   look up in pairs  key " l2 "
                                                    pairs  call Web1 .JsonTextDecode
                                                                         jsonText  get responseContent
                                                 notFound " - "
                                   " | "
                                   look up in pairs  key " l3 "
                                                    pairs  call Web1 .JsonTextDecode
                                                                         jsonText  get responseContent
                                                 notFound " - "
                                   " | "
                                   look up in pairs  key " l4 "
                                                    pairs  call Web1 .JsonTextDecode
                                                                         jsonText  get responseContent
                                                 notFound " - "
```

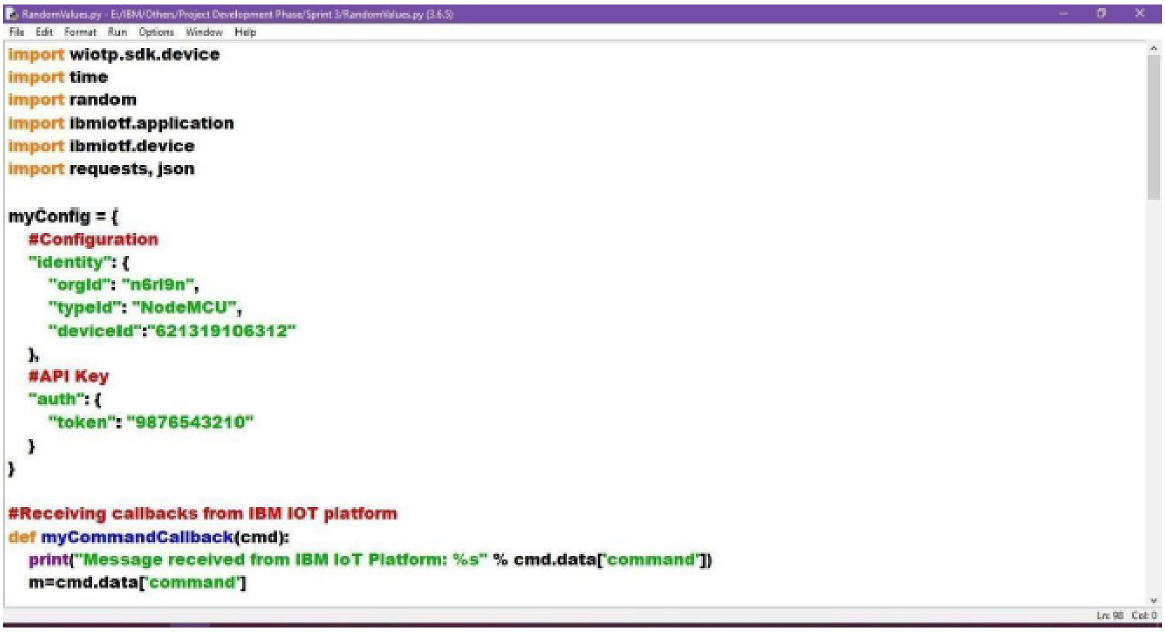
Project Development Phase Sprint IV

Code for print the random temperature, Road signs, Speed limit, Message : (RandomValues.py)

```
import wiotp.sdk.device import time
import random
import ibmiotf.application import ibmiotf.device import requests, json
myConfig = {
#Configuration "identity": {
"orgId": "n6rl9n",
"typeId": "NodeMCU", "deviceId":"621319106312"
},
#API Key
"auth": {
```

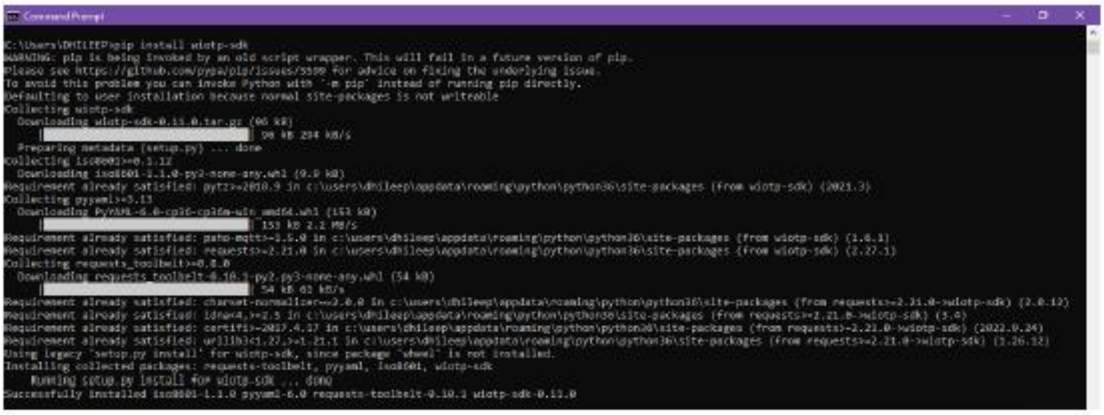
```
"token": "9876543210"
}
}
#Receiving callbacks from IBM IOT platform def myCommandCallback(cmd):
print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
m=cmd.data['command']
client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

#OpenWeatherMap Credentials
BASE_URL = "https://api.openweathermap.org/data/2.5/weather?" CITY = "Salem, IN"
URL = BASE_URL + "q=" + CITY + "&units=metric"+"&appid=" +
"f58e4720c739a54c439aba9b05176839"
while True:
response = requests.get(URL) if response.status_code == 200: data = response.json()
main = data['main'] temperature = main['temp'] humidity = main['humidity'] pressure
= main['pressure'] report = data['visibility']
#messge part msg=random.randint(0,5) if msg==1:
message="GO SLOW, SCHOOL ZONE AHEAD"
elif msg==2:
message="NEED HELP, POLICE STATION AHEAD"
elif msg==3:
message="EMERGENCY, HOSPITAL NEARBY"
elif msg==4:
message="DINE IN, RESTAURENT AVAILABLE"
elif msg==5:
message="PETROL BUNK NEARBY"
else:
message=""
#Speed Limit part speed=random.randint(0,150) if speed>=100:
speedMsg=" Limit Exceeded" elif speed>=60 and speed<100:
speedMsg="Moderate" else:
speedMsg="Slow"
#Diversion part sign=random.randint(0,5) if sign==1:
signMsg="Right Diversion" elif sign==2:
signMsg="Speed Breaker" elif sign==3:
signMsg="Left Diversion" elif sign==4:
signmsg="U Turn" else:
signMsg=""
#Visibility
if temperature < 24:
visibility="Fog Ahead, Drive Slow" elif temperature < 20:
visibility="Bad Weather" else:
visibility="Clear Weather"
else:
print("Error in the HTTP request")

myData={'Temperature':temperature, 'Message':message, 'Sign':signMsg,
'Speed':speedMsg, 'Visibility':visibility}
client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None) #PUBLISHING TO IOT WATSON
print("Published data Successfully: ", myData) print("
") client.commandCallback = myCommandCallback time.sleep(5)
client.disconnect()
```

Python Simulation :

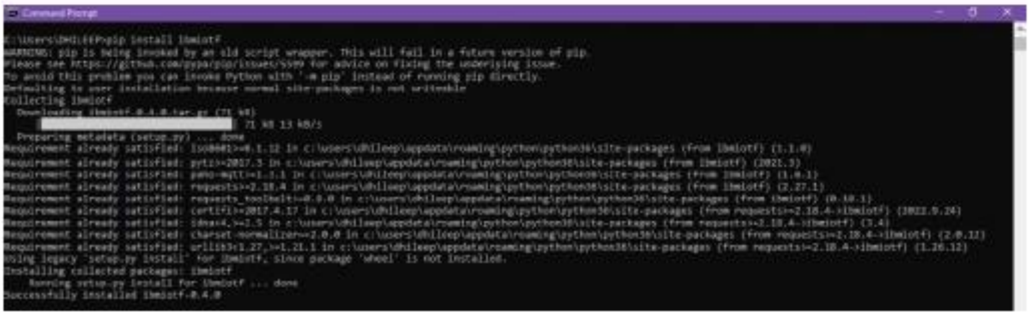

RandomValues.py - E:/IBM/Others/Project Development Phase/Sprint 3/RandomValues.py (3.6.5)
File Edit Format Run Options Window Help
Ln: 98 Col: 0

```python
import wiotp.sdk.device
import time
import random
import ibmiotf.application
import ibmiotf.device
import requests, json

myConfig = {
   #Configuration
   "identity": {
      "orgId": "n6rl9n",
      "typeId": "NodeMCU",
      "deviceId":"621319106312"
   },
   #API Key
   "auth": {
      "token": "9876543210"
   }
}

#Receiving callbacks from IBM IOT platform
def myCommandCallback(cmd):
   print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
   m=cmd.data['command']
```

Import wiotp-sdk & ibmiotf :

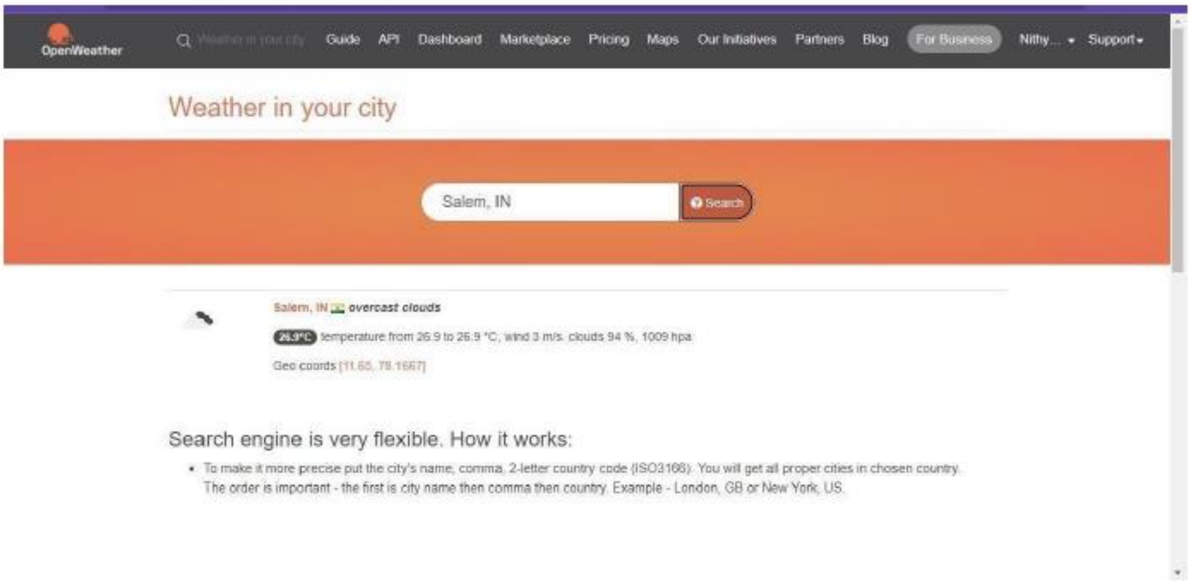


```
C:\Users\DHILEEP>pip install wiotp-sdk
WARNING: pip is being invoked by an old script wrapper. This will fail in a future version of pip.
Please see https://github.com/pypa/pip/issues/5599 for advice on fixing the underlying issue.
To avoid this problem you can invoke Python with '-m pip' instead of running pip directly.
Defaulting to user installation because normal site-packages is not writeable
Collecting wiotp-sdk
  Downloading wiotp-sdk-0.11.0.tar.gz (96 kB)
     |████████████████████████████████| 98 kB 294 kB/s
  Preparing metadata (setup.py) ... done
Collecting iso8601>=0.1.12
  Downloading iso8601-1.1.0-py2-none-any.whl (9.9 kB)
Requirement already satisfied: pytz>=2018.9 in c:\users\dhileep\appdata\roaming\python\python36\site-packages (from wiotp-sdk) (2021.3)
Collecting pyyaml>=5.13
  Downloading PyYAML-6.0-cp36-cp36m-win_amd64.whl (153 kB)
     |████████████████████████████████| 153 kB 2.2 MB/s
Requirement already satisfied: paho-mqtt>=1.5.0 in c:\users\dhileep\appdata\roaming\python\python36\site-packages (from wiotp-sdk) (1.6.1)
Requirement already satisfied: requests>=2.21.0 in c:\users\dhileep\appdata\roaming\python\python36\site-packages (from wiotp-sdk) (2.27.1)
Collecting requests_toolbelt>=0.8.0
  Downloading requests_toolbelt-0.10.1-py2.py3-none-any.whl (54 kB)
     |████████████████████████████████| 54 kB 61 kB/s
Requirement already satisfied: charset-normalizer~=2.0.0 in c:\users\dhileep\appdata\roaming\python\python36\site-packages (from requests>=2.21.0->wiotp-sdk) (2.0.12)
Requirement already satisfied: idna<4,>=2.5 in c:\users\dhileep\appdata\roaming\python\python36\site-packages (from requests>=2.21.0->wiotp-sdk) (3.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\dhileep\appdata\roaming\python\python36\site-packages (from requests>=2.21.0->wiotp-sdk) (2022.9.24)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\dhileep\appdata\roaming\python\python36\site-packages (from requests>=2.21.0->wiotp-sdk) (1.26.12)
Using legacy 'setup.py install' for wiotp-sdk, since package 'wheel' is not installed.
Installing collected packages: requests-toolbelt, pyyaml, iso8601, wiotp-sdk
    Running setup.py install for wiotp-sdk ... done
Successfully installed iso8601-1.1.0 pyyaml-6.0 requests-toolbelt-0.10.1 wiotp-sdk-0.11.0
```



```
C:\Users\DHILEEP>pip install ibmiotf
WARNING: pip is being invoked by an old script wrapper. This will fail in a future version of pip.
Please see https://github.com/pypa/pip/issues/5599 for advice on fixing the underlying issue.
To avoid this problem you can invoke Python with '-m pip' instead of running pip directly.
Defaulting to user installation because normal site-packages is not writeable
Collecting ibmiotf
  Downloading ibmiotf-0.4.0.tar.gz (71 kB)
     |████████████████████████████████| 71 kB 13 kB/s
  Preparing metadata (setup.py) ... done
Requirement already satisfied: iso8601>=0.1.12 in c:\users\dhileep\appdata\roaming\python\python36\site-packages (from ibmiotf) (1.1.0)
Requirement already satisfied: pytz>=2017.3 in c:\users\dhileep\appdata\roaming\python\python36\site-packages (from ibmiotf) (2021.3)
Requirement already satisfied: paho-mqtt>=1.3.1 in c:\users\dhileep\appdata\roaming\python\python36\site-packages (from ibmiotf) (1.6.1)
Requirement already satisfied: requests>=2.18.4 in c:\users\dhileep\appdata\roaming\python\python36\site-packages (from ibmiotf) (2.27.1)
Requirement already satisfied: requests_toolbelt>=0.8.0 in c:\users\dhileep\appdata\roaming\python\python36\site-packages (from ibmiotf) (0.10.1)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\dhileep\appdata\roaming\python\python36\site-packages (from requests>=2.18.4->ibmiotf) (2022.9.24)
Requirement already satisfied: idna<4,>=2.5 in c:\users\dhileep\appdata\roaming\python\python36\site-packages (from requests>=2.18.4->ibmiotf) (3.4)
Requirement already satisfied: charset-normalizer~=2.0.0 in c:\users\dhileep\appdata\roaming\python\python36\site-packages (from requests>=2.18.4->ibmiotf) (2.0.12)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\dhileep\appdata\roaming\python\python36\site-packages (from requests>=2.18.4->ibmiotf) (1.26.12)
Using legacy 'setup.py install' for ibmiotf, since package 'wheel' is not installed.
Installing collected packages: ibmiotf
    Running setup.py install for ibmiotf ... done
Successfully installed ibmiotf-0.4.0
```

OpenWeatherMap - (Ex., Salem, IN) :



Weather in your city

Salem, IN

Salem, IN overcast clouds

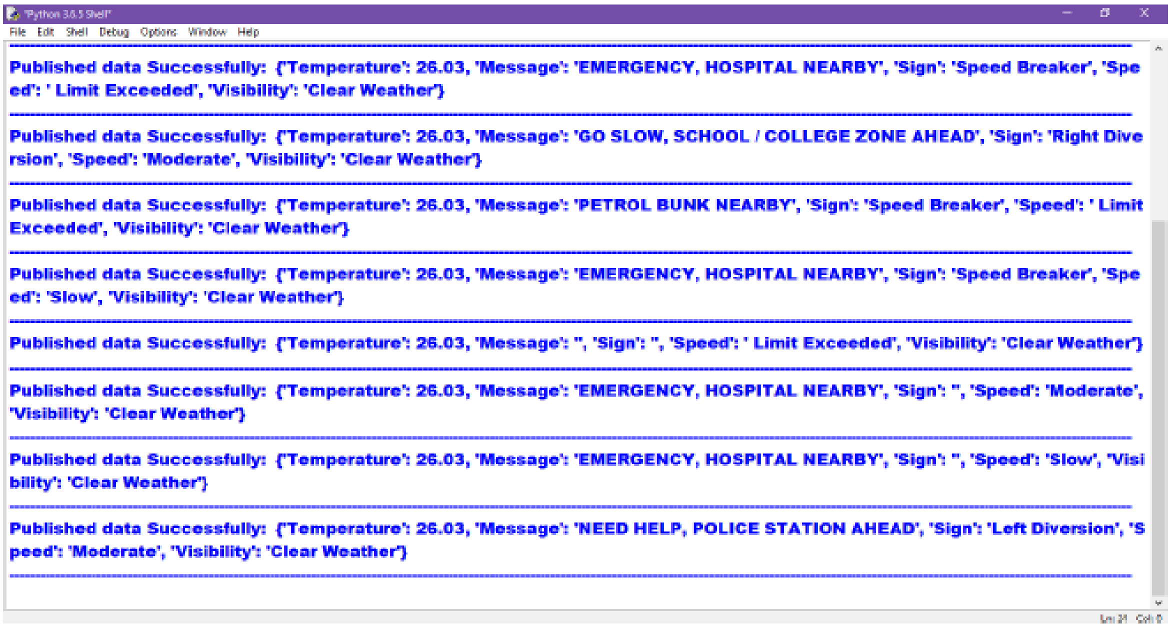
26.9°C temperature from 26.9 to 26.9 °C, wind 3 m/s. clouds 94 %, 1009 hpa

Geo coords [11.65, 78.1667]

Search engine is very flexible. How it works:

- To make it more precise put the city's name, comma, 2-letter country code (ISO3166). You will get all proper cities in chosen country. The order is important - the first is city name then comma then country. Example - London, GB or New York, US.

Python IDLE Output :



```
-------------------------------------------------------------------------------------------------------
Published data Successfully:  {'Temperature': 26.03, 'Message': 'EMERGENCY, HOSPITAL NEARBY', 'Sign': 'Speed Breaker', 'Spe
ed': ' Limit Exceeded', 'Visibility': 'Clear Weather'}
-------------------------------------------------------------------------------------------------------
Published data Successfully:  {'Temperature': 26.03, 'Message': 'GO SLOW, SCHOOL / COLLEGE ZONE AHEAD', 'Sign': 'Right Dive
rsion', 'Speed': 'Moderate', 'Visibility': 'Clear Weather'}
-------------------------------------------------------------------------------------------------------
Published data Successfully:  {'Temperature': 26.03, 'Message': 'PETROL BUNK NEARBY', 'Sign': 'Speed Breaker', 'Speed': ' Limit
Exceeded', 'Visibility': 'Clear Weather'}
-------------------------------------------------------------------------------------------------------
Published data Successfully:  {'Temperature': 26.03, 'Message': 'EMERGENCY, HOSPITAL NEARBY', 'Sign': 'Speed Breaker', 'Spe
ed': 'Slow', 'Visibility': 'Clear Weather'}
-------------------------------------------------------------------------------------------------------
Published data Successfully:  {'Temperature': 26.03, 'Message': '', 'Sign': '', 'Speed': ' Limit Exceeded', 'Visibility': 'Clear Weather'}
-------------------------------------------------------------------------------------------------------
Published data Successfully:  {'Temperature': 26.03, 'Message': 'EMERGENCY, HOSPITAL NEARBY', 'Sign': '', 'Speed': 'Moderate',
'Visibility': 'Clear Weather'}
-------------------------------------------------------------------------------------------------------
Published data Successfully:  {'Temperature': 26.03, 'Message': 'EMERGENCY, HOSPITAL NEARBY', 'Sign': '', 'Speed': 'Slow', 'Visi
bility': 'Clear Weather'}
-------------------------------------------------------------------------------------------------------
Published data Successfully:  {'Temperature': 26.03, 'Message': 'NEED HELP, POLICE STATION AHEAD', 'Sign': 'Left Diversion', 'S
peed': 'Moderate', 'Visibility': 'Clear Weather'}
-------------------------------------------------------------------------------------------------------
```

FINAL CODE

PROGRAM CODE :

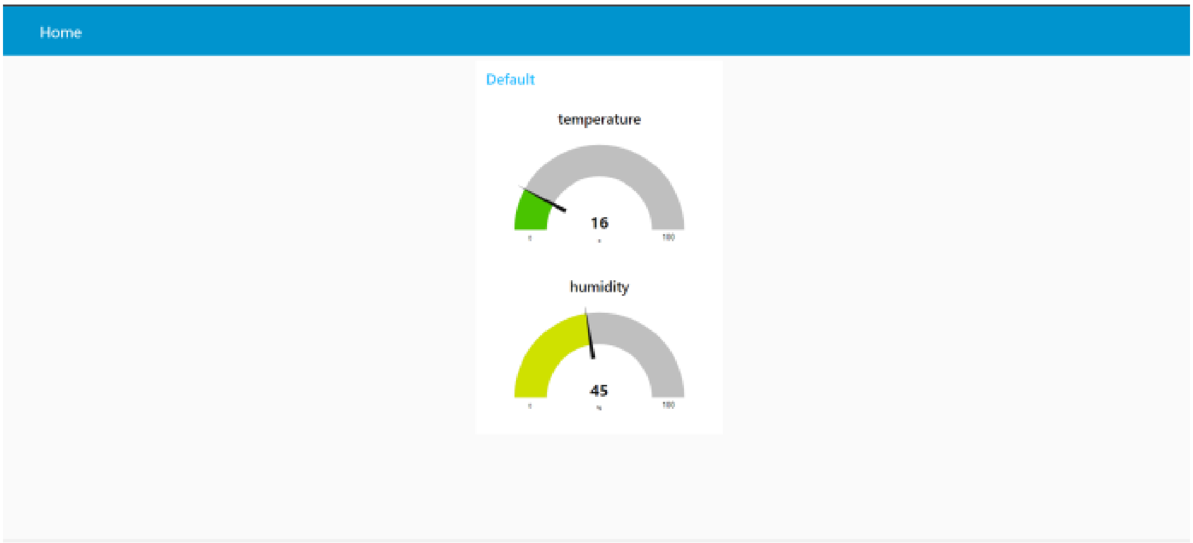
main.py

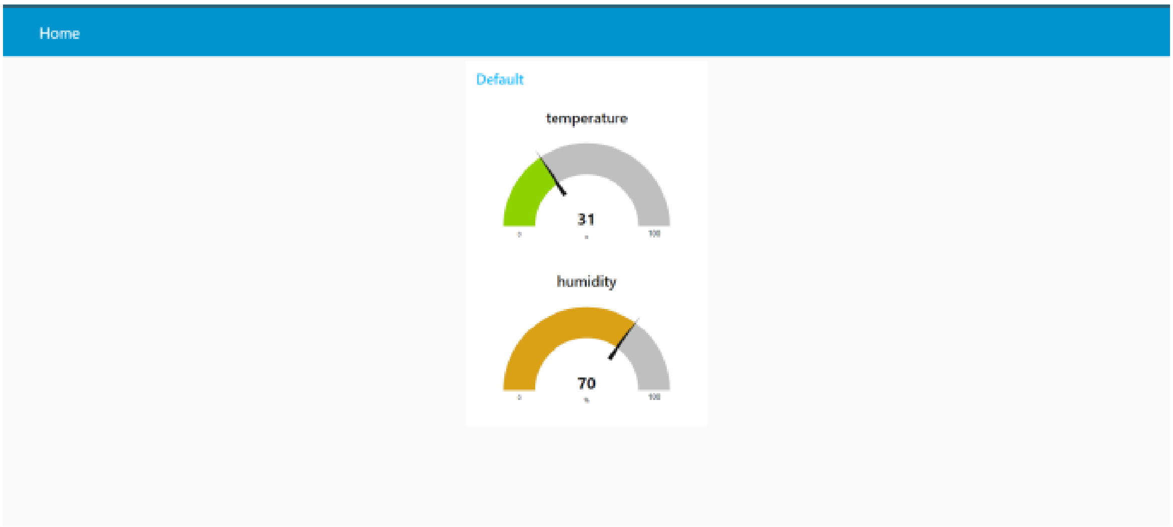
```
#IBM Watson IOT Platform

#pip install wiotp-sdk

import wiotp.sdk.device

import time
import random

myConfig = {

    "identity":  {

        "orgId": "2r52ij",

        "typeId": "Roadsafety",

        "deviceId":"1234"

    },

    "auth": {

        "token": "12345678"

    }

}


def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" %
cmd.data['command'])

    m=cmd.data['command']


client = wiotp.sdk.device.DeviceClient(config=myConfig,
logHandlers=None)

client.connect()


while True:

    temp=random.randint(-20,125)

    hum=random.randint(0,100)

    myData={'temperature':temp, 'humidity':hum}

    client.publishEvent(eventId="status", msgFormat="json",
data=myData, qos=0, onPublish=None)

    print("Published data Successfully: %s", myData)

    client.commandCallback = myCommandCallback

    time.sleep(2)

client.disconnect()
```

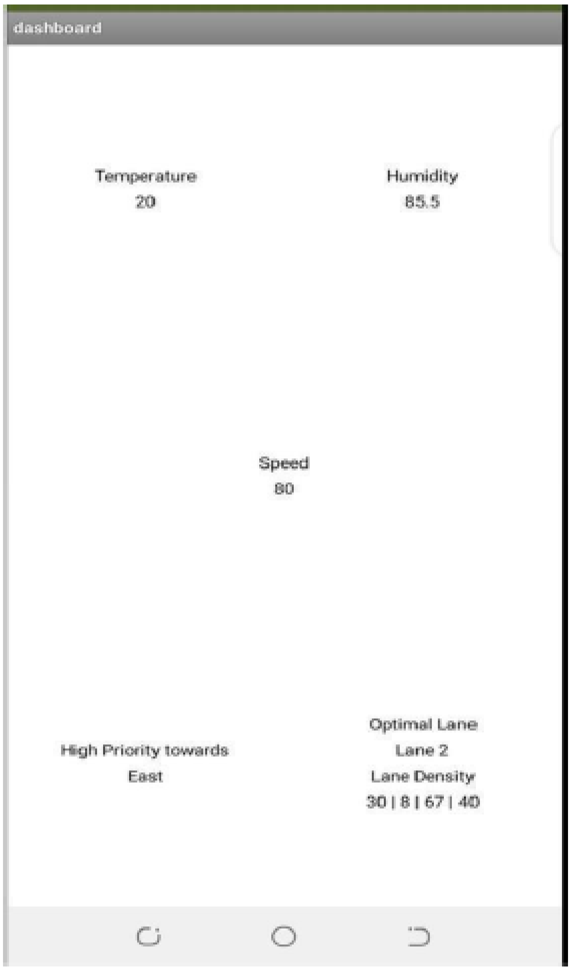
Weather.py

```
import requests as reqs


def get(myLocation,APIKEY):
    apiURL =

f"https://api.openweathermap.org/data/2.5/weather?q={myLocation

}&appid={APIKEY}"

    responseJSON = (reqs.get(apiURL)).json()
    returnObject = {
        "temperature" : responseJSON['main']['temp'] - 273.15,
        "weather" : [responseJSON['weather'][_]['main'].lower() for _
in range(len(responseJSON['weather']))],
        "visibility" : responseJSON['visibility']/100, # visibility in
percentage where 10km is 100% and 0km is 0%
    }
    if("rain" in responseJSON):
        returnObject["rain"] = [responseJSON["rain"][key] for key in
responseJSON["rain"]]
    return(returnObject)
    }
```

OUTPUT:


Home
Default
temperature
16
0
100
humidity
45
0
%
100


Home
Default
temperature
31
0
100
humidity
70
0
%
100



dashboard

Temperature
20

Humidity
85.5

Speed
80

High Priority towards
East

Optimal Lane
Lane 2
Lane Density
30 | 8 | 67 | 40