SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD SAFETY

Technology - Internet Of Things

Domain -Safety

HX8001-PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY AND ENTREPRENEURSHIP

TEAM ID : PNT2022TMID46185

Selvakumar P (815119106038) Team Lead
Hariharasudhan M (815119106014)
Ramkumar M (815119106032)
Deepak M (815119106009)
Vinith S (815119106047)

1. **INTRODUCTION**

   1.1 Project Overview

1.INTRODUCTION

1.1 Project Overview

To complete this project, you must have knowledge of the following:

You need to have basic knowledge of the following cloud services:

- IBM Watson IoT Platform
- Node-RED Service
- Cloudant DB

1.2 Purpose

By the end of this project you will:

- Gain knowledge of Watson IoT Platform.
- Connecting IoT devices to the Watson IoT platform and exchanging the data and to display values.
- Gain knowledge of OpenWeatherMap API Service
- Creating a Web Application through which the user interacts with the device.

Project Flow:

- Receiving road sign values to the IBM IoT platform from Node-RED Web UI
- Weather conditions can be viewed in the Web Application

To accomplish this, we have to complete all the activities and tasks listed below:

- Create and configure IBM Cloud Services
    - Create IBM Watson IoT Platform
    - Create a device & configure the IBM IoT Platform
    - Create Node-RED service
    - Create a database in Cloudant DB to store location data
- Develop a web Application using Node-RED Service.
    - Develop the web application using Node-RED
- Develop a python script to publish the location details to the IBM IoT platform

2.LITERATURE SURVEY

2.1 Existing problem

- ➤ The plan was costly and challenging
- ➤ It depends 24x7 connectivity power supply
- ➤ Scalability while Block chain not Indestructible

- Security issues in terms of public date
- May load a way terminals social discrimination

2.2 References

[1] European Commission, - “Advanced Driver Assistance Systems (ADAS)”, European Road Safety Observatory,https://ec.europa.eu/transport/road_safety/sites/roadsafety/files/ersosynthesis2016-adas15_en.pdf, 2016

[2]. ‘The UK Highway Code – Traffic Signs, Department for Transport’, https://www.gov.uk/guidance/the-highwaycode/traffic-signs, 2017.

[3] Ziebinski A, Cupek R, et. Al.: Review of Advanced Driver Assistance System (ADAS), AIP Conference Proceedings, Vol 1906, No. 1, 2017.

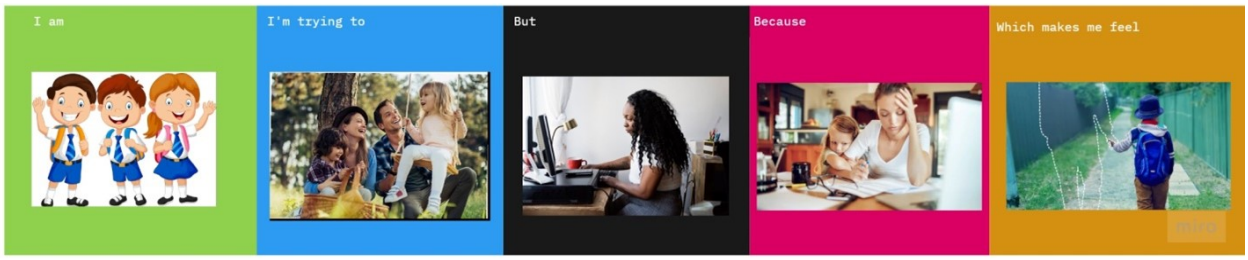
2.3 Problem Statement Definition

- Around the world road traffic injuries cause 1.35 million death and up to 50 million injuries each year. These injuries can lead to life long disability including brain and spinal cord injuries.

PROBLEM STATEMENT I

PROBLEM STATEMENT II

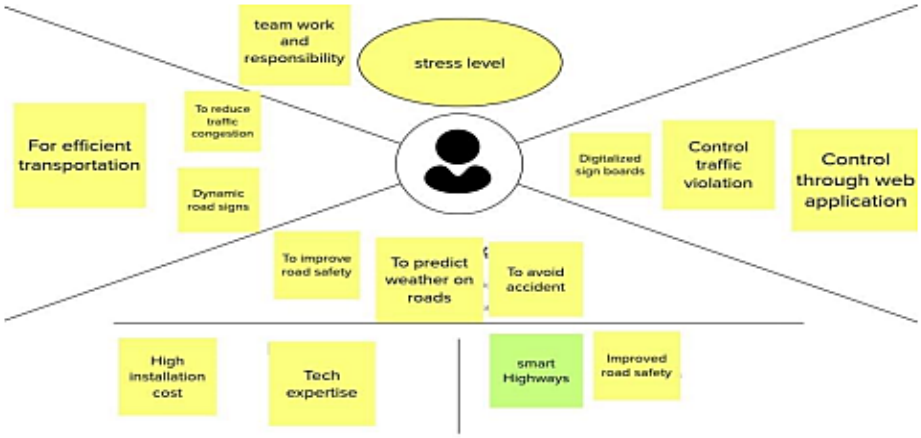
PROBLEM STATEMENT III



3. IDEATION & PROPOSED SOLUTION

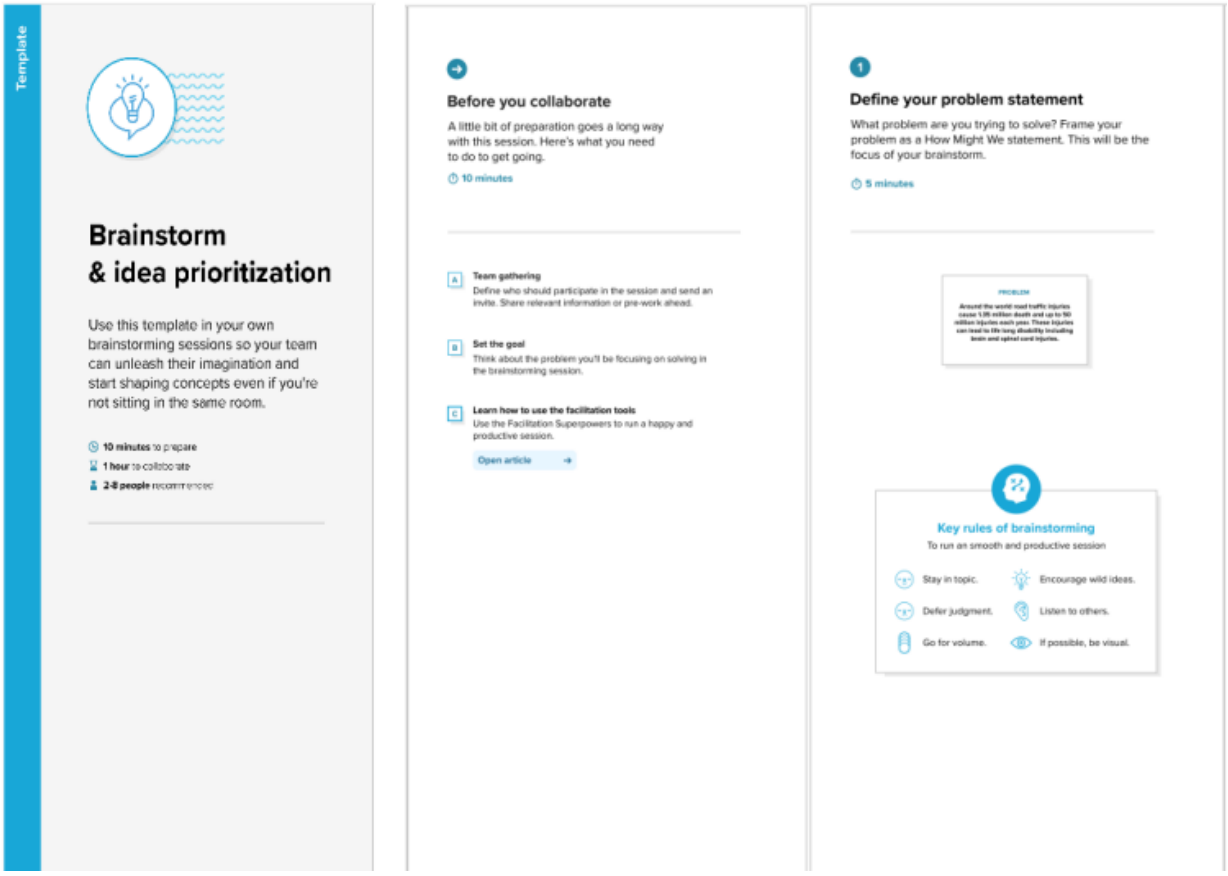
3.1Empathy Map Canvas

- Gain knowledge of Watson IOT Platform.
- Connecting IOT device to the Watson IOT Platform and exchanging the data and dispaly



3.2Ideation & Brainstorming

Step 1: Team Gathering, collaboration and select the problem statement



Template

Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

Open article

1 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

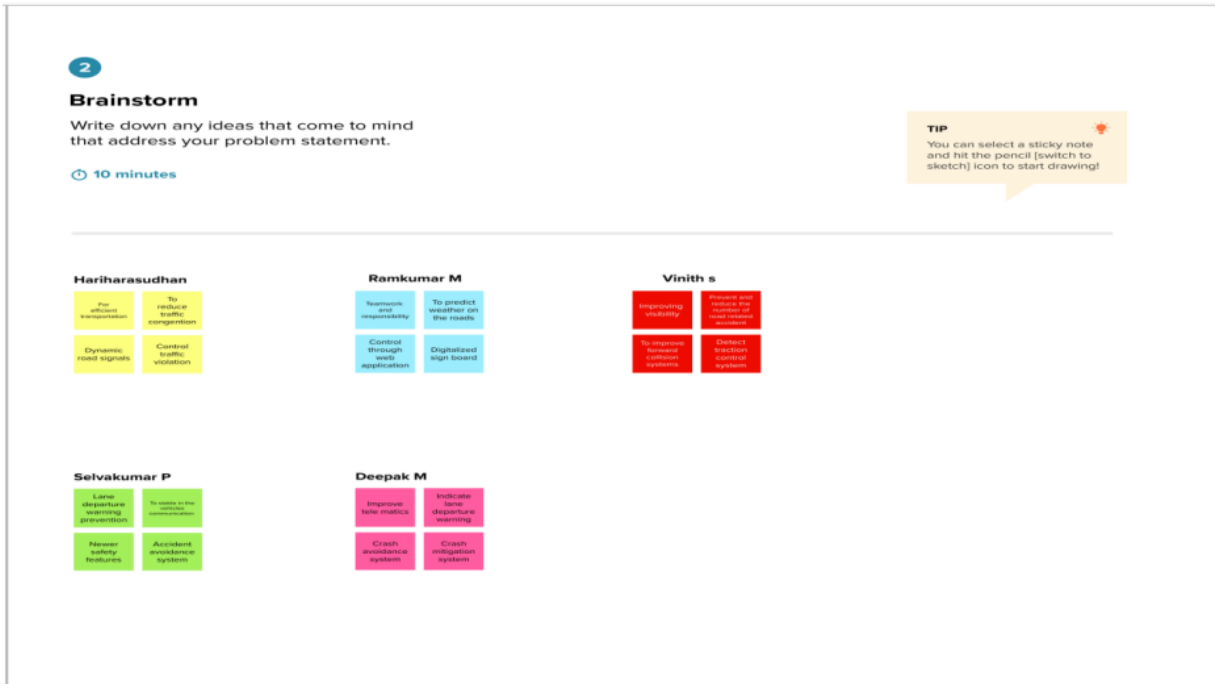
PROBLEM

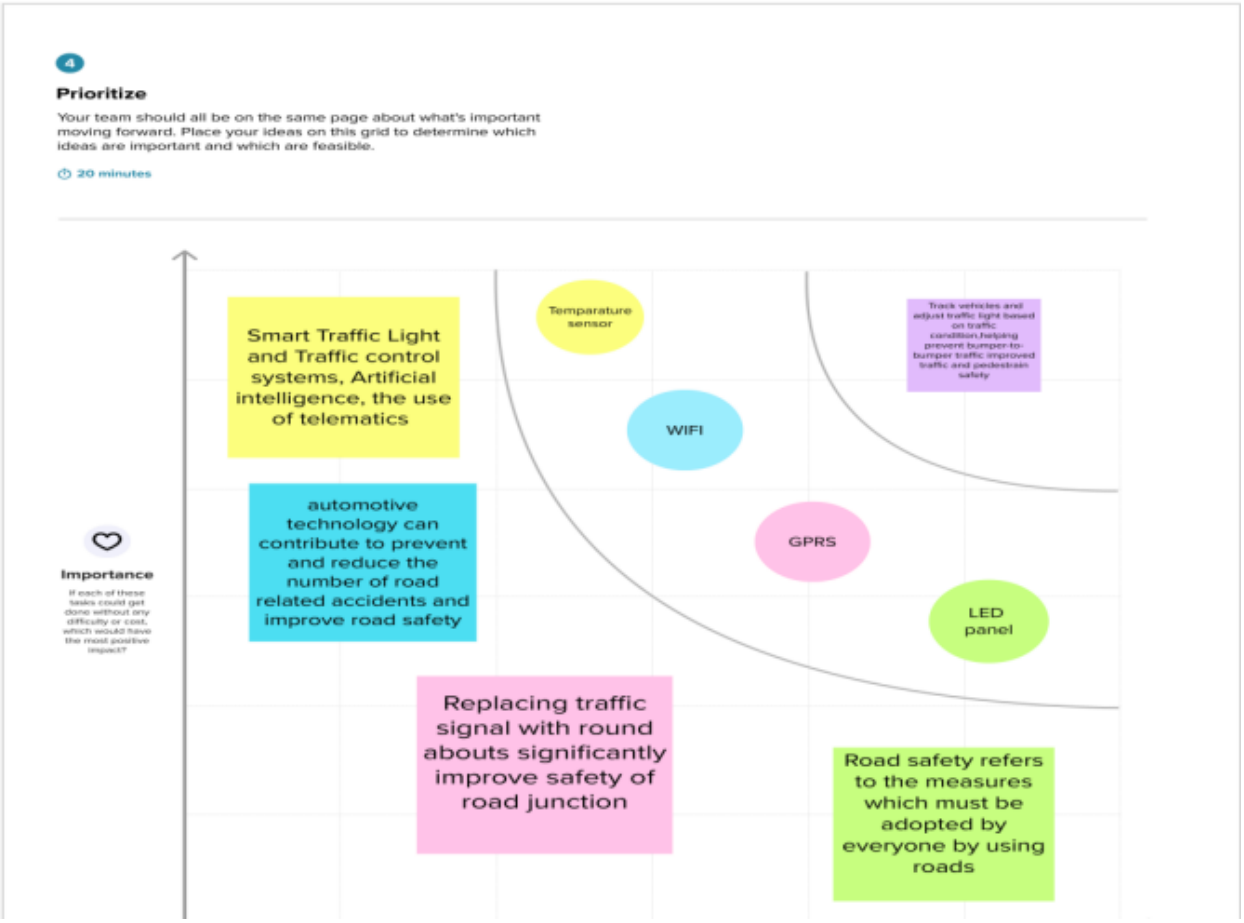
Around the world road traffic injuries cause 1.35 million death and up to 50 million injuries each year. These injuries can lead to life-long disability (including brain and spinal cord injuries.

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.

Step-2: Brainstorm, ideas listening and grouping





3.3Proposed Solution

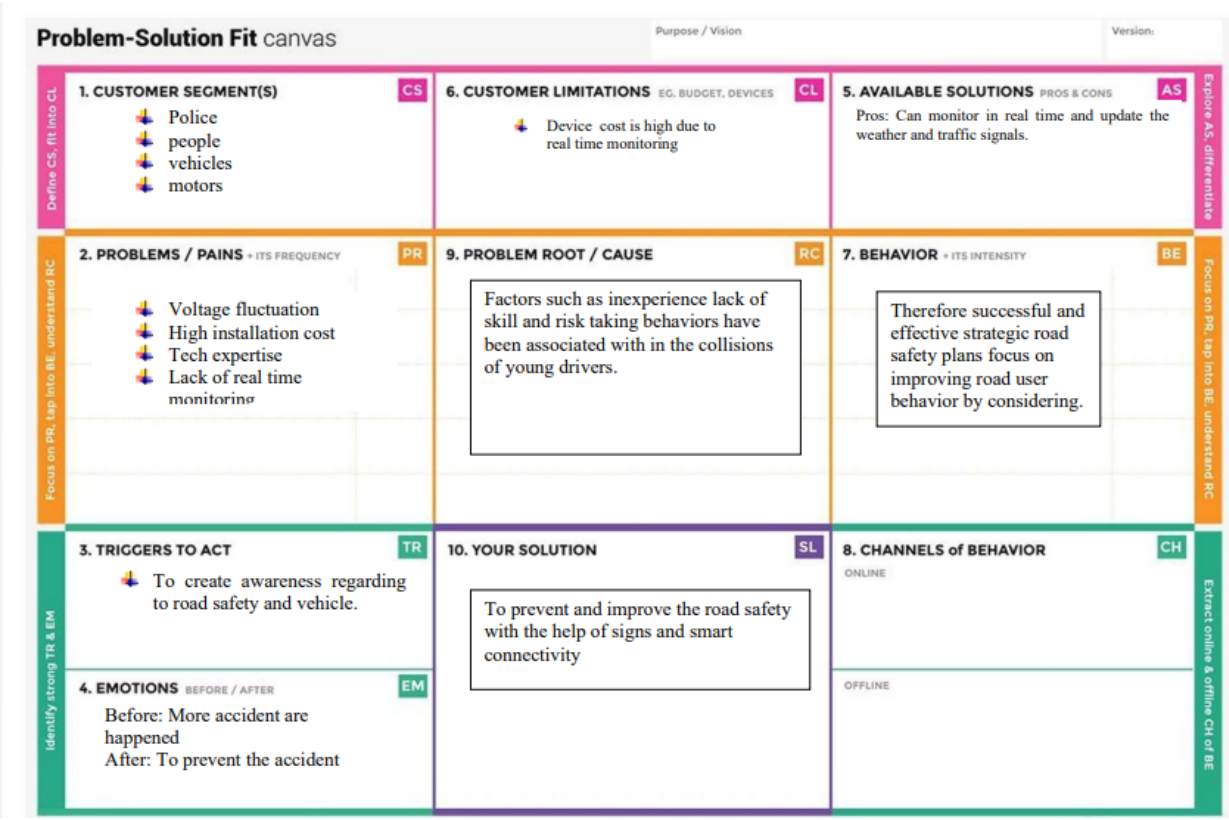
In this activity you are expected to prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Project - Signs with Smart Connectivity for Better Road Safety is used to educate the drivers digitally using IOT who do not have knowledge about traffic signs and weather indication for the drivers and passengers convenience.
2.	Idea / Solution description	Replacing the man made painted signs into digital as well as their name which is more visible compared to current signs and also indicating weather in the same sign boards for driver where weather is not predictable
3.	Novelty / Uniqueness	Weather indication on sign boards is unique which will help mostly the two wheelers from unfortunate heavy rains and winds. Digital traffic signs also educates the drivers to follow traffic rules easily.
4.	Social Impact / Customer Satisfaction	It makes the people to know about traffic signs if they don't know ,it shows signs digitally to avoid the accidents and weather indication based on IOT to avoid accidents and it helps mostly for two wheeler passenger.
5.	Business Model (Revenue Model)	This project can make revenue by selling many equipments to the government sector and also private sectors(educational &medical institutions).Maintain services are also taken by the company

6.	Scalability of the Solution	It makes the daily life of drivers and passengers better. The product can be scalable by adding new features to the product makes more revenue

3.4Problem Solution fit

In this activity you are expected to prepare problem - solution fit document and submit for review.



4. REQUIREMENT ANALYSIS

4.1Functional requirement

Following are the functional requirements of the proposedsolution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story/ Sub-Task)
FR-1	Travelers Registration	Registration in the platform needs for communicatingwith customer throughtheir mobile

FR-2	Transport Agency Registration	Register for getting approval to implement the smartsignboards for betterroad safety
FR-3	Weather Monitoring	Open weather API implemented to monitor weatherreports and updatein database
FR-4	Sensor implementation	Monitoring traffic density and road condition, pedestrian monitoring and controls traffic signals.
FR-5	Database Management	Updating information in the database to intimate theusers aboutthe abnormal situations
FR-6	Information Sharing	Once the situation detected the userget information via the digital display who travels alongthe road also itwillupdate in theplatform, so othersplan accordingly

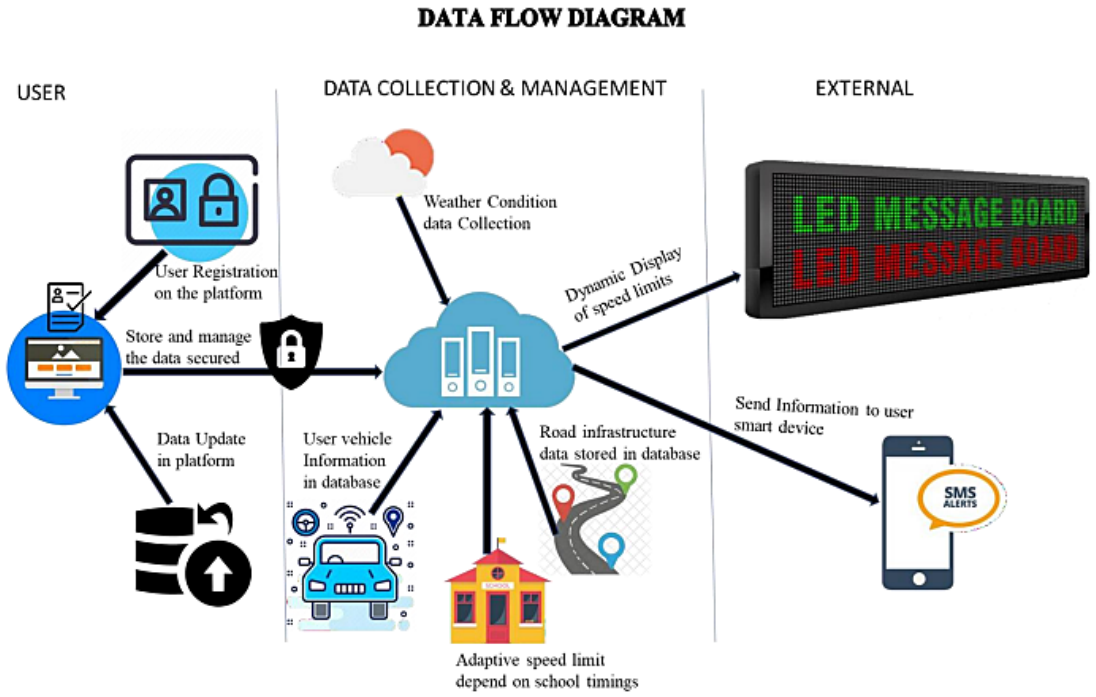
4.2Non-Functional requirements

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Easy to follow instructions displays on the board.Understanding the signsshould be clear.
NFR-2	**Security**	Provide better security, any other thirdparty can't able to display information in the board, Users data are kept confidential.
NFR-3	**Reliability**	It can able to withstand in any weather condition and thehardware parts require periodic monitoring to avoid any damage. It is dynamic in natureandreduce traffic congestion.
NFR-4	**Performance**	The smart displayimproves the safety and it makes user tense freeand keep them in a comfort zone.Also quality of service is improved.
NFR-5	**Availability**	The solution is available 24X7 and also withstandany climate changes.
NFR-6	**Scalability**	It can be implemented efficiently in anywhere and data execution will be faster.Provides better safety

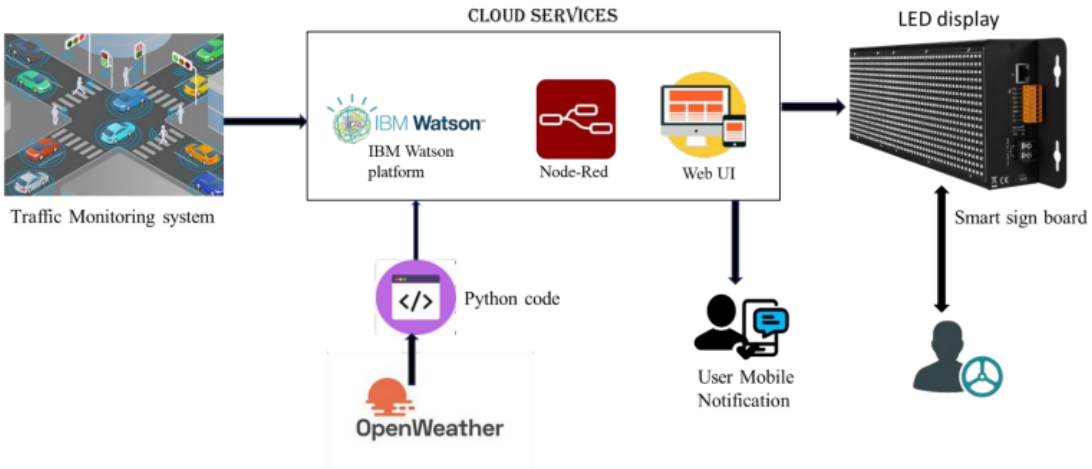
5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2Solution & Technical Architecture

Technical Architecture:



6. PROJECT PLANNING & SCHEDULING

In this milestone you are expected to prepare milestones & tasks, sprint schedules

6.1 Sprint Planning & Estimation

Use the below templateto create productbacklog and sprintschedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	DEEPAK
Sprint-1		USN-2	As a user, I will receive confirmation email onceI have registered for the application	1	High	VINITH

Sprint-1		USN-3	As a user, I can register for the application through Facebook	2	Low	RAMKUMAR
Sprint-1		USN-4	As a user, I can register for the application through Gmail	2	Medium	SELVA KUMAR
Sprint-1	Login	USN-5	As a user, I can log into the application byentering email & password	1	High	HARI HARA SUDHAN
Sprint-1	Dashboard	USN-6	As a user, I can log into the application by entering email & password and access all the resources and services available	2	High	VINITH

Sprint	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Story Points**	**Priority**	**Team Members**
Sprint-2	Login	USN-1	As a weather data controller, I log into my profileand start monitoring the weather updates	3	High	RAMKUMAR
Sprint-2	Dashboard	USN-2	I receiveall the information about weather fromweb from weatherAPI. Whenever thereis change in weather, corresponding updates aremade on sign boards.	2	Medium	DEEPAK
Sprint-3	Login	USN-1	As a imagecontroller, I keepnote of all the images received from various areas	3	High	HARIHARA SUDHAN

			and detecttraffic in thatparticular area.			
Sprint-3	Dashboard	USN-2	With the traffic, updatesI change the status of sign board as “take diversion”.	2	Medium	VINITH
Sprint-4	Login	USN-1	As a zonal officer, I ensure that boards near school display “slowdown” and nearhospitals display “no horn”.	3	High	SELVA KUMAR
Sprint-4	Login	USN-1	As an administrator, I ensure that all departments work co-ordinated and ensure the accuracy and efficiency.	2	Medium	RAMKUMAR

Project Tracker, Velocity& Burndown Chart:

Sprint	**Total Story Points**	**Duration**	**Sprint Start Date**	**Sprint End Date(Planned)**	**Story Points Completed (as on Planned End Date)**	**Sprint Release Date (Actual)**
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022		
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022		
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022		

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

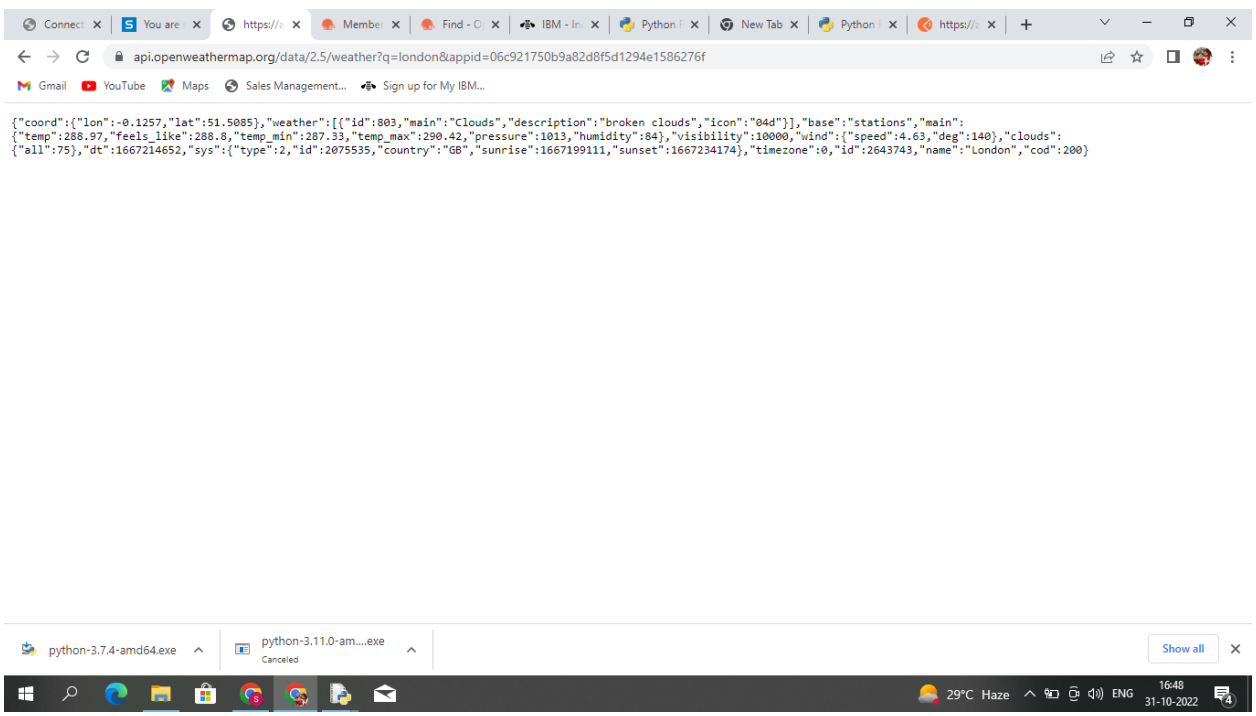
Velocity:

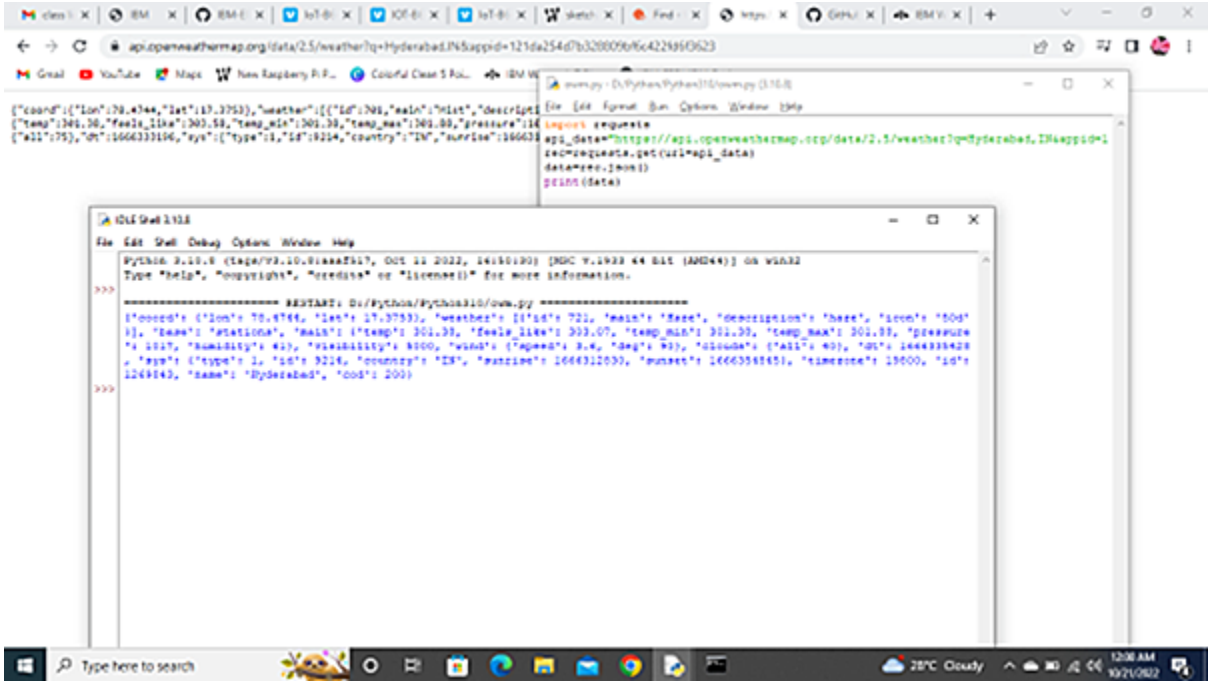
Imagine we havea10-day sprint duration, and the velocityof the team is 20 (points per sprint). Let'scalculate the team's average velocity(AV) per iterationunit (story points per day)

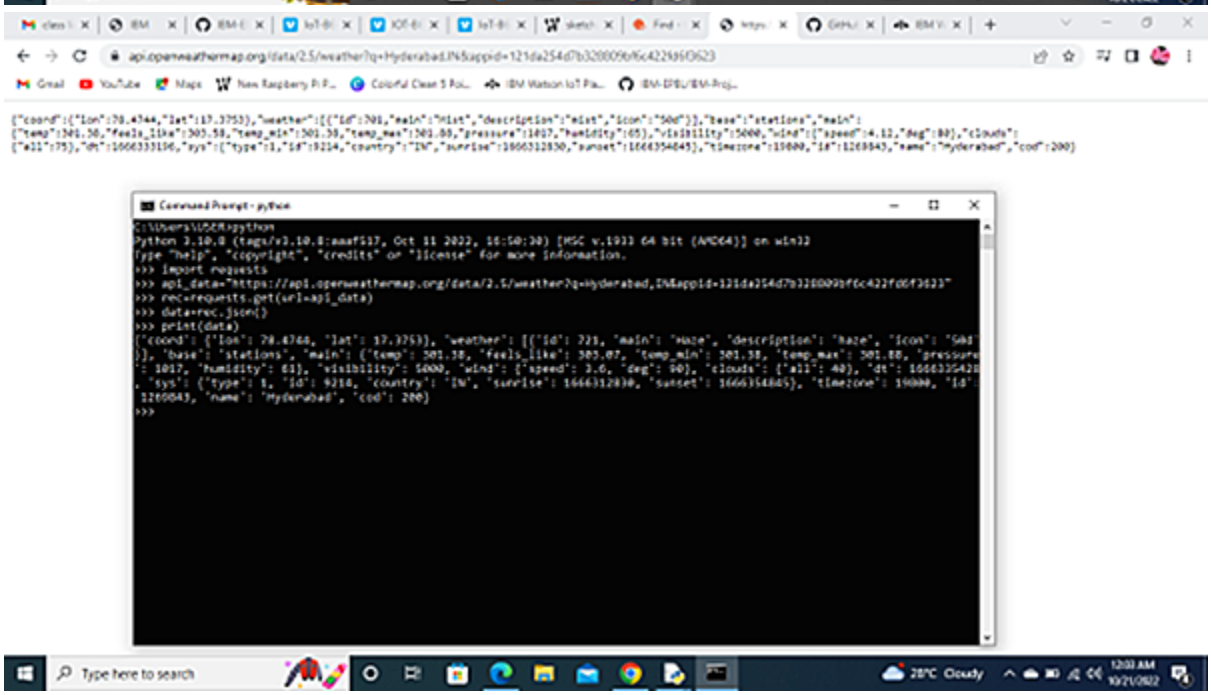
6.2 Sprint Delivery Schedule

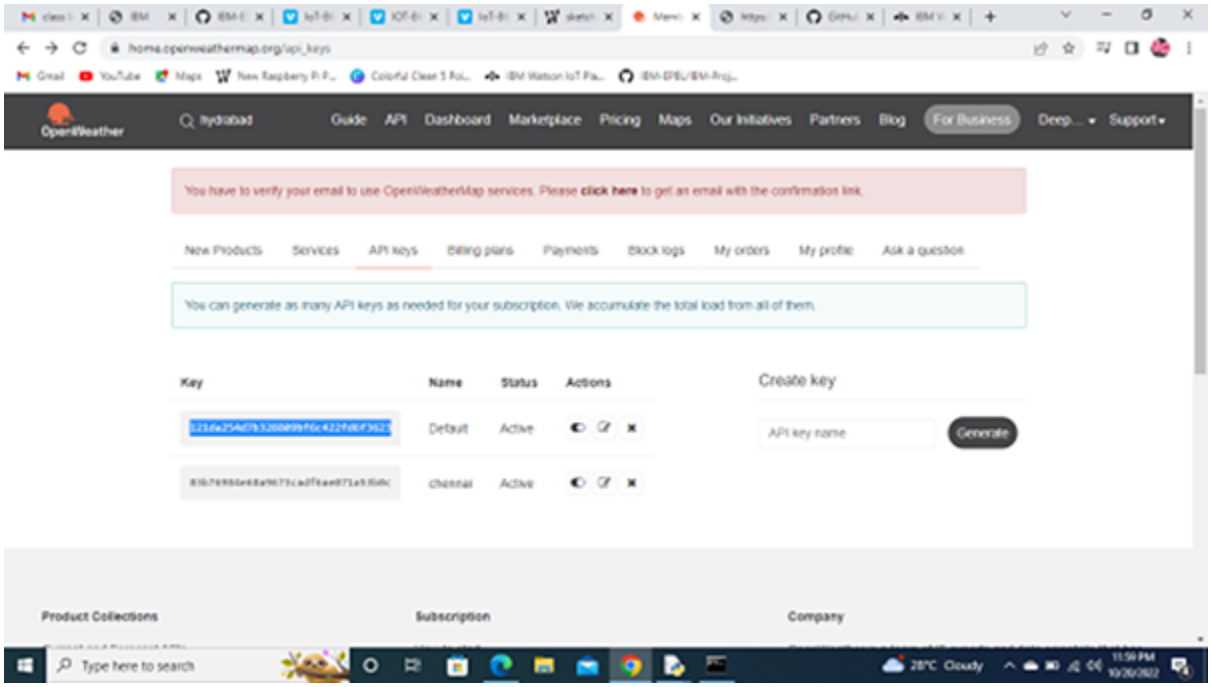
sprint.1

USN-1 Install Watson IoT Python SDK TO Connect to UBM Watson Platform using python:

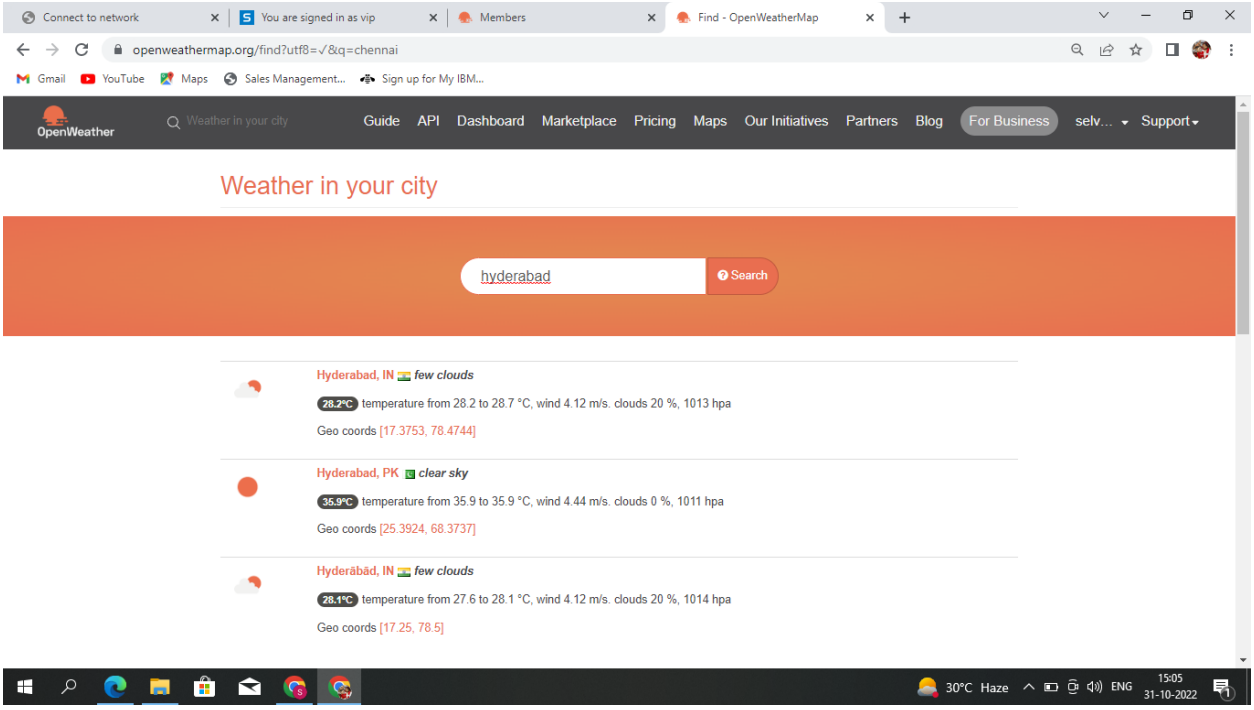



import requests
print(data)
Python 3.10.8 (tags/v3.10.8:aaaf517, Oct 11 2022, 16:50:30) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
Type here to search


Command Prompt - python
Python 3.10.8 (tags/v3.10.8:aaaf517, Oct 11 2022, 16:50:30) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import requests
>>> rec=requests.get(url=api_data)
>>> data=rec.json()
>>> print(data)
Type here to search

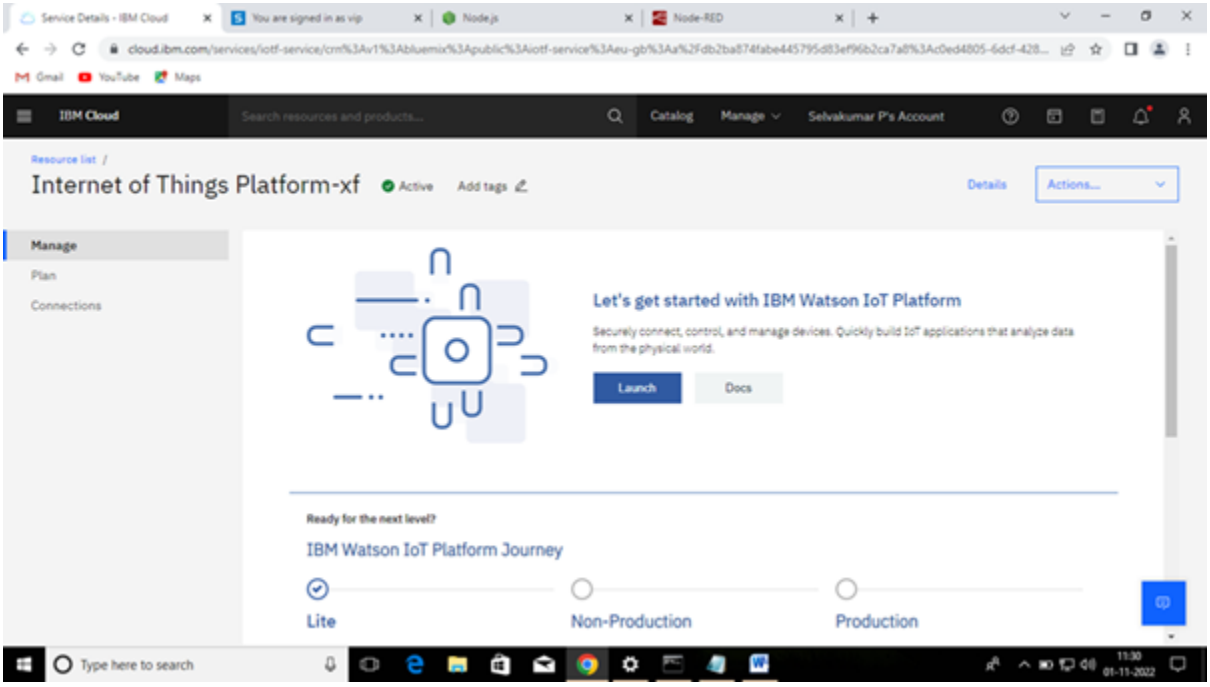


USN-2 Create An Account In Open weather Map Website:

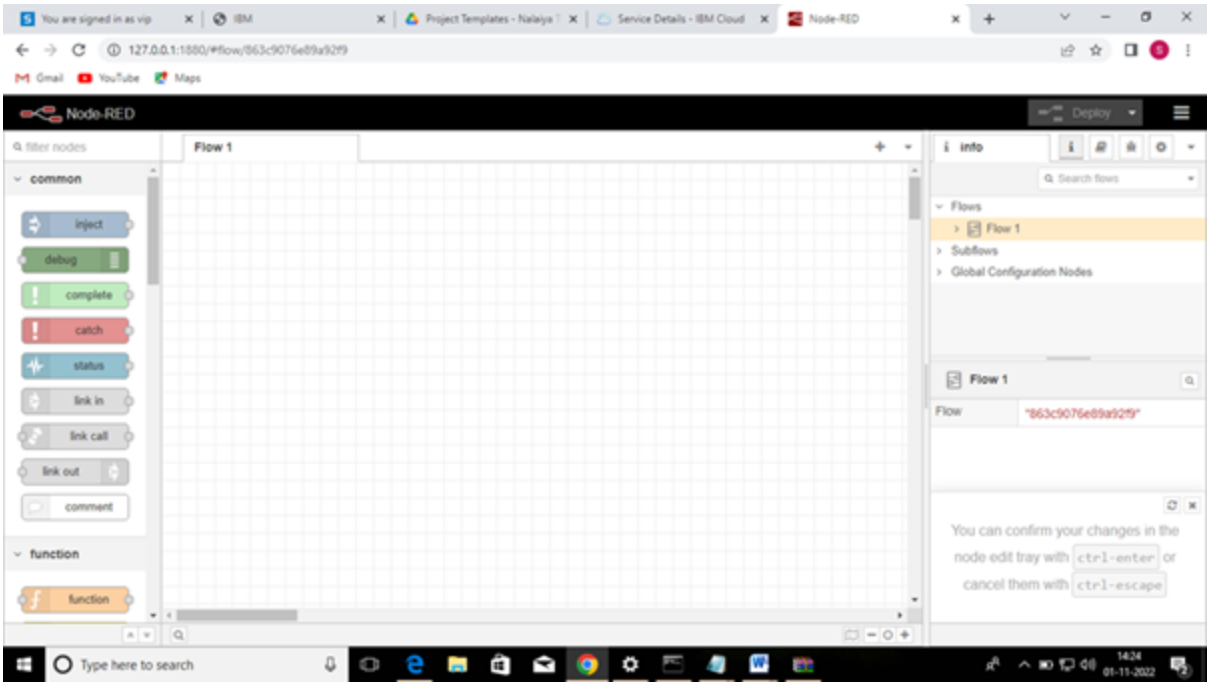


USN-3 IBM Cloud Services:

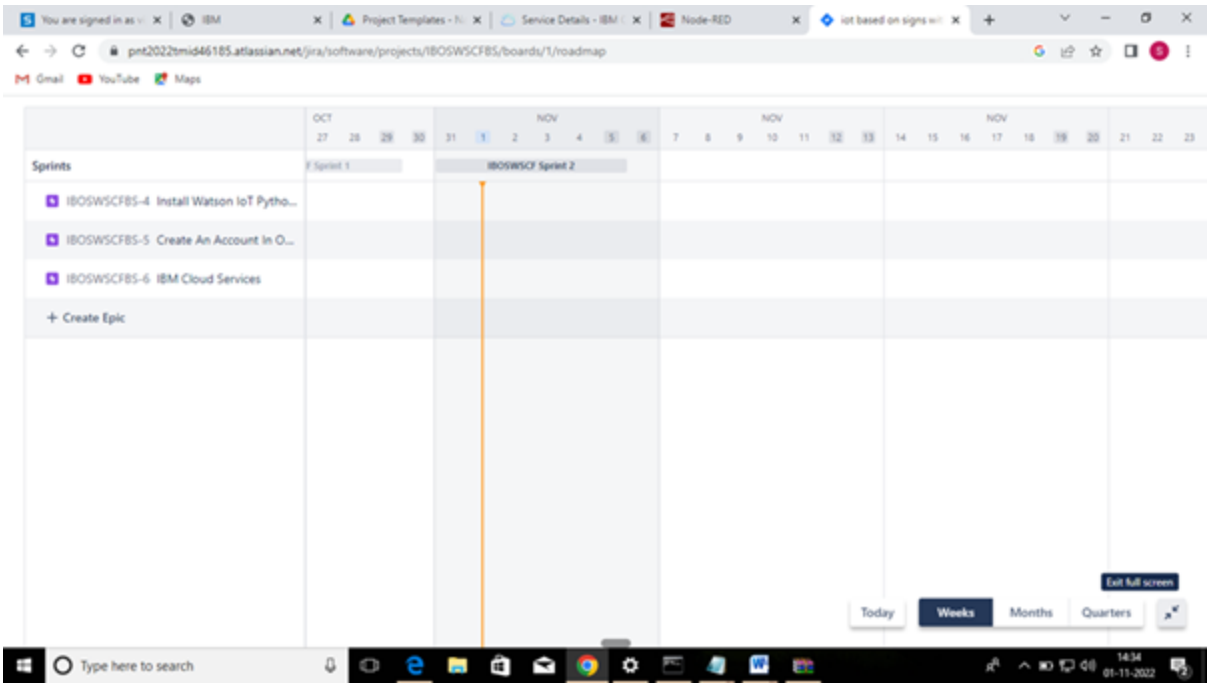
IBM Watson IoT platform



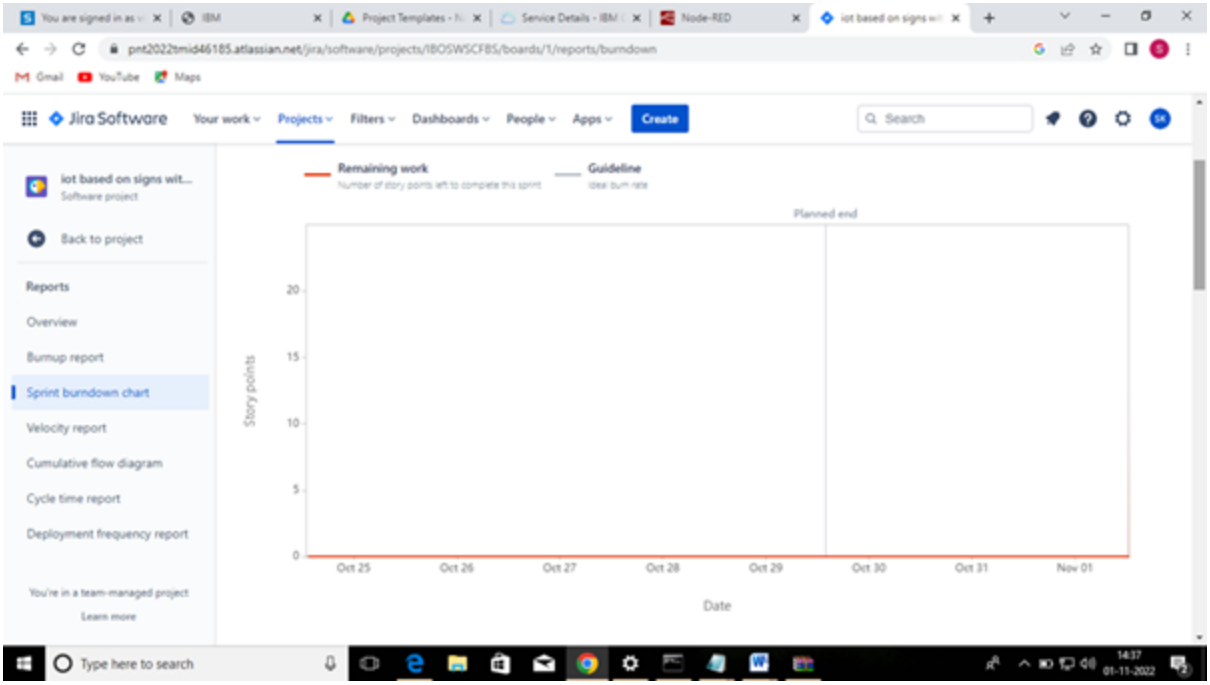
Node-Red Service



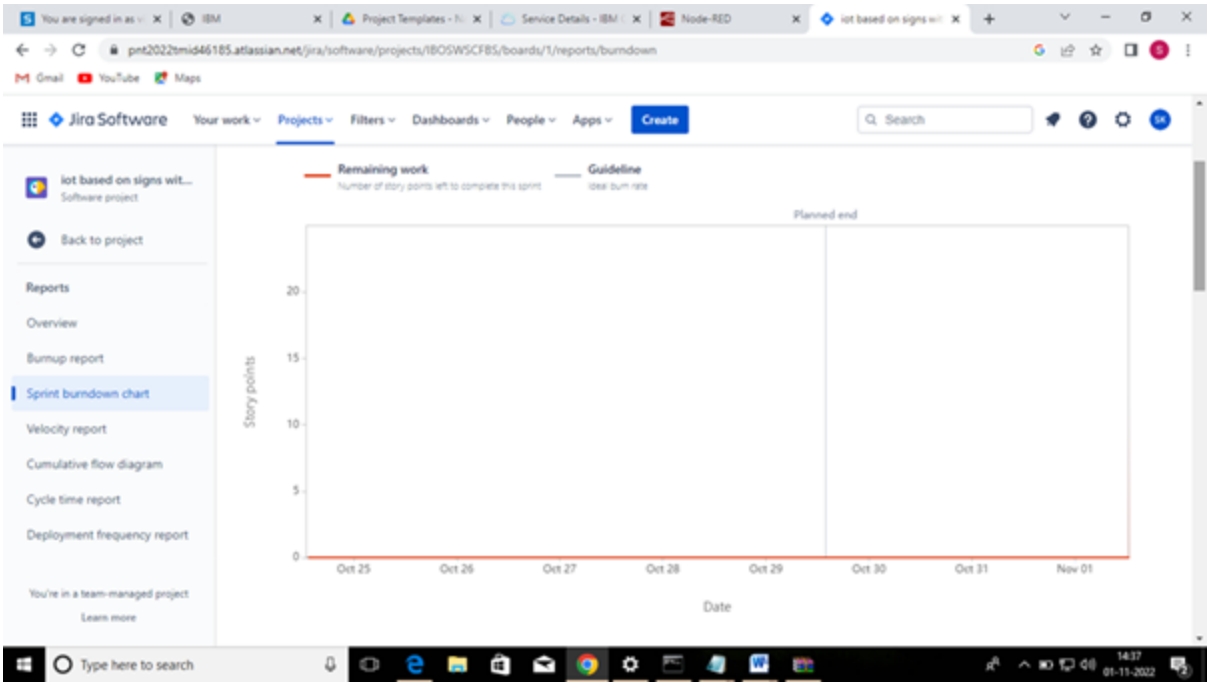
ROADMAP:



BURNDOWN CHART:

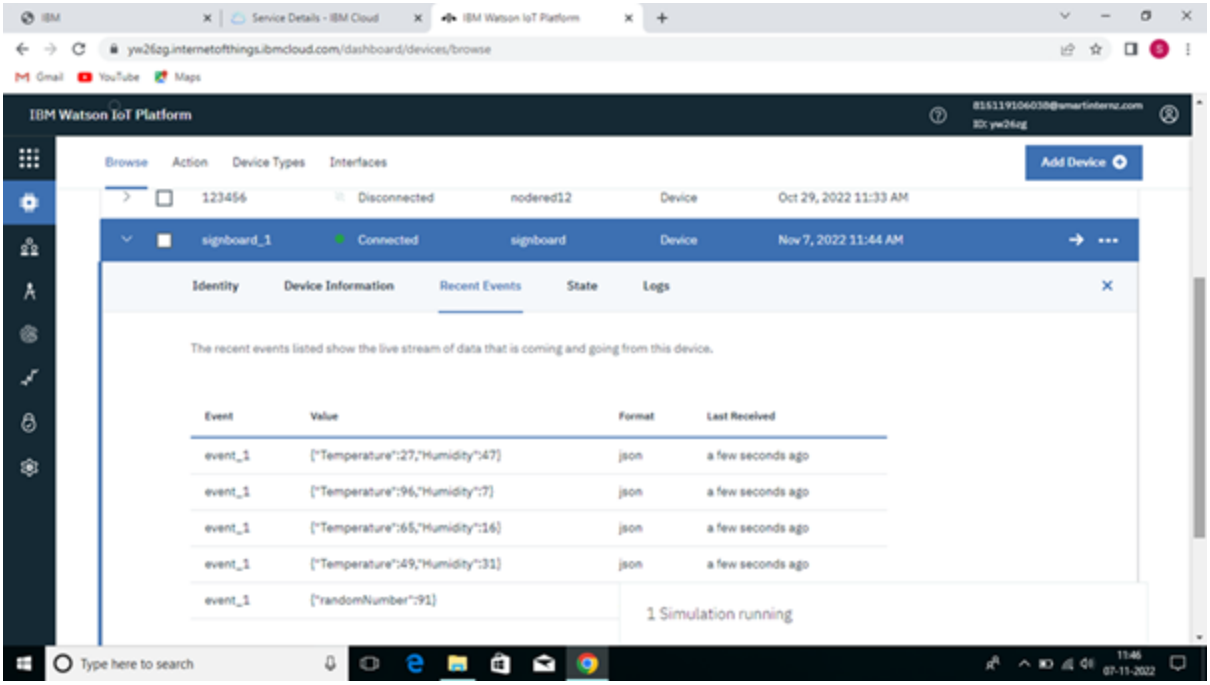


VELOCITY REPORT:

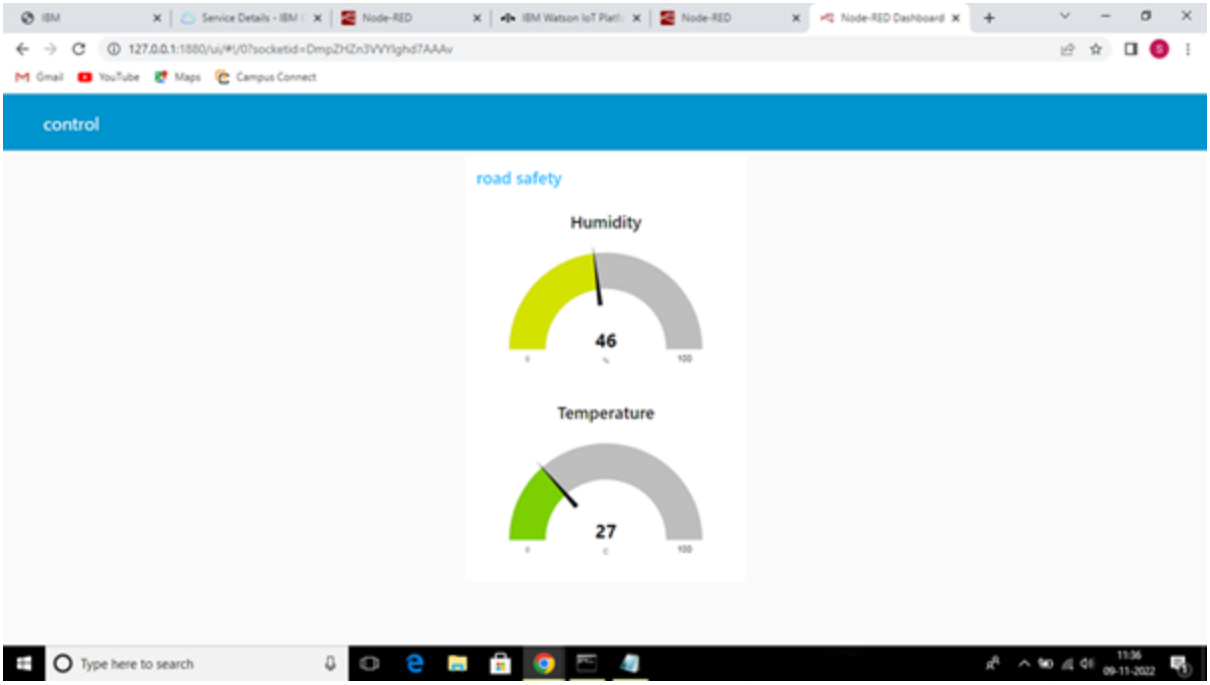


sprint 2

USN-4 –Create and configure IBM Cloud service



USN-5- Create Node –RED Service

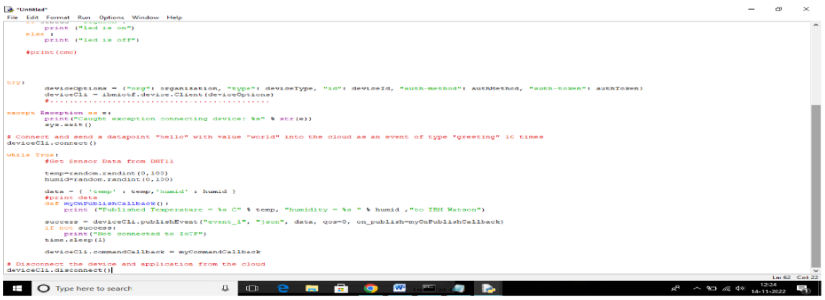


sprint 3

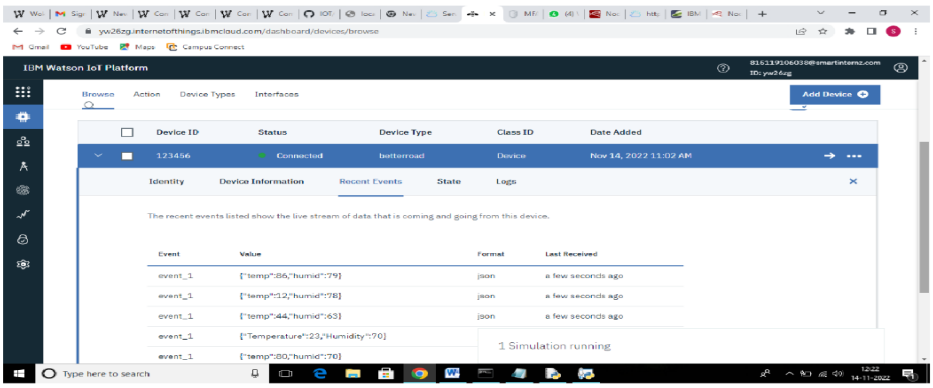
PYTHON CODE

```
data = { 'temp' : temp,'humid' : humid }

#print data

def myOnPublishCallback():

    print ("Published Temperature = %s C" % temp, "humidity = %s " % humid ,"to IBM Watson")


success = deviceCli.publishEvent("event_1", "json", data, qos=0,
on_publish=myOnPublishCallback)

if not success:

    print("Not connected to IoTF")

time.sleep(1)


deviceCli.commandCallback = myCommandCallback


# Disconnect the device and application from the cloud

deviceCli.disconnect()
```

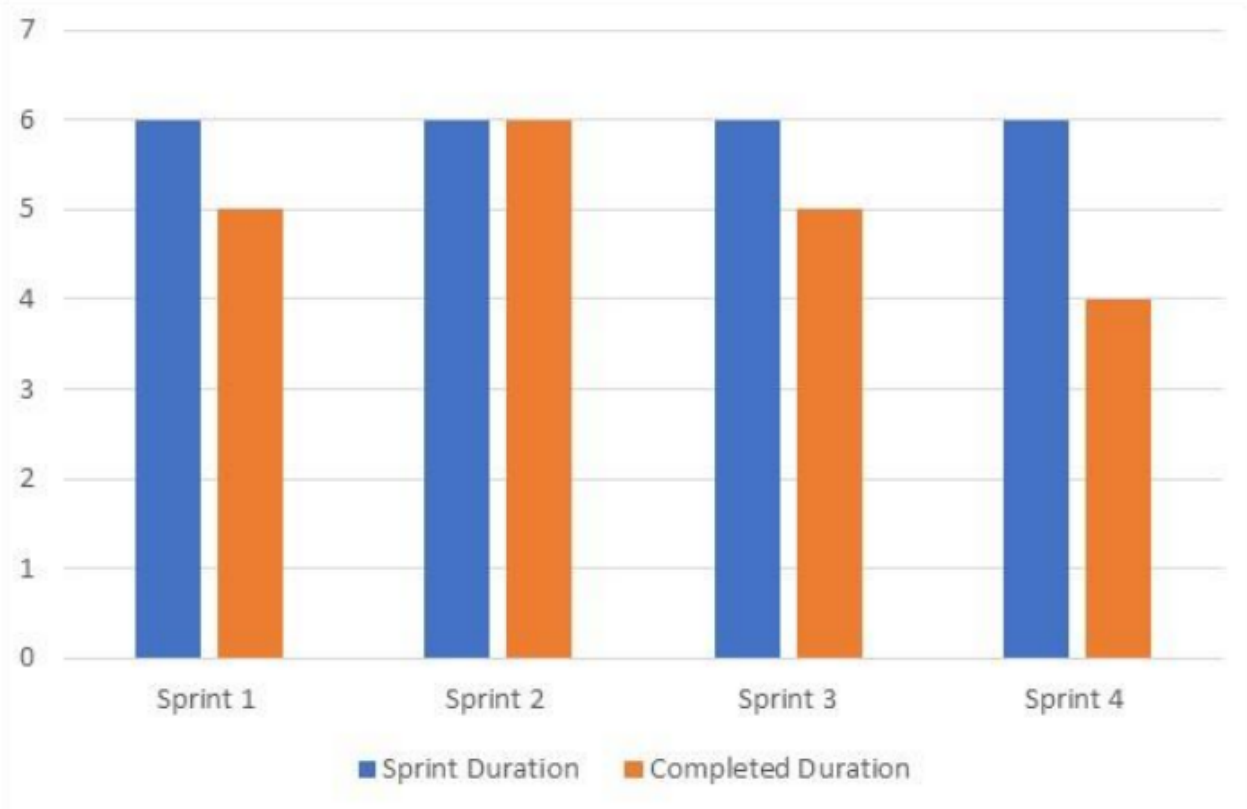
output



OUTPUT:



6.3Reports from JIRA



7 CODING & SOLUTIONING (Explain the featuresadded in the project along with code)

Coding is basically the computer language used to develop apps, websites, and software. Without it, we'd have none of the most popular technology we've come to rely on such as Facebook, our smartphones, the browser we choose to view our favorite blogs, or even the blogs themselves an action or process of solving a problem.

7.1 Feature

PYTHON:

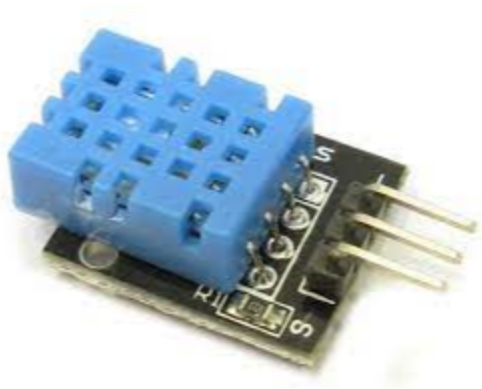
Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including

structured, object-oriented and functional programming

7.2Feature 2

Temparature sensor

The temperature sensor in Arduino **converts the surrounding temperature to voltage**. It further converts the voltage to Celcius, Celcius to Fahrenheit, and prints the Fahrenheit temperature on the LCD screen. We will use a temperature sensor (TMP 36) of low voltage.



NODE- RED

7.3 Database Schema (if Applicable)

8. TESTING

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random



#Provide your IBM Watson Device Credentials
organization = "rv07c6"
deviceType = "riverwaterquality-22_23"
deviceId = "123456"
authMethod = "token"
authToken = "wQ_)43L5c0@ku8)sgd"

# Initialize GPIO



def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="lighton":
        print ("led is on")
    else :
        print ("led is off")
    #print(cmd)



try:
        deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method":
authMethod, "auth-token": authToken}
        deviceCli = ibmiotf.device.Client(deviceOptions)
        #..............................................

except Exception as e:
        print("Caught exception connecting device: %s" % str(e))
        sys.exit()
```

```
# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting" 10 times
deviceCli.connect()

while True:
        #Get Sensor Data from DHT11
        temp=random.randint(0,100)
        ph=random.randint(0,14)
        turb=random.randint(0,100)
        data = { 'temp' : temp, 'ph': ph,'turb' :turb }
        #print data
        def myOnPublishCallback():
            print ("Published Temperature = %s C" % temp, "ph = %s %%" % ph,"turbidity = %s NTU " % turb ,"to IBM Watson")

        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
        if not success:
            print("Not connected to IoTF")
        time.sleep(1)
        deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

8.2 User Acceptance Testing

				Date	10-Nov-22					
				Team ID	PNT2022TMID46185					
				Project Name	Project – signs with smart connectivity for better road safety					
				Maximum Marks	4 marks					
Test case ID	**Feature Type**	**Component**	**Test Scenario**	**Pre-Requisite**	**Steps To Execute**	**Test Data**	**Expected Result**	**Actual Result**	**Status**	**Co**
LoginPage_TC_O O1	Functional	Home Page	Verify user is able to see the Login/Signup popup when user clicked on My account button	IBM Cloud services	1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Singup popup displayed or not	www.cloud.ibm.com	Login/Signup popup should display	Working as expected	Pass	

LoginPage_TC_O O2	UI	Home Page	Verify the UI elements in Login/ Signup popup	IBM Cloud services	1 .Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Singup popup with below UI elements: a.email text box b.password text box c.Login button d.New customer? Create account link e.Last password? Recovery password link	www.cloud.ibm.com	Application should show below UI elements: a.email text box b.password text box c.Login button with orange colour d.New customer? Create account link e.Last password? Recovery password link	Working as expected	Fail	Steps
LoginPage_TC_O O3	Functional	Home page	Verify user is able to log into application with Valid credentials	IBM Cloud services	1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button	Username:815119106038 @smartinternz.com password: Ibmproject	User should navigate to user account homepage	Working as expected	Pass	
LoginPage_TC_O O4	Functional	Login page	Verify user is able to log into application with InValid credentials	IBM Cloud services	1 .Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter InValid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button	Username:815119106038 @smartinternz.com password: Ibmproject	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass	
LoginPage_TC_O O4	Functional	Login page	Verify user is able to log into application with InValid credentials	IBM Cloud services	1 .Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter Invalid password in password text box 5.Click on login button	Username:815119106038 @smartinternz.com password: Ibmproject	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass	
LoginPage_TC_O O5	Functional	Login page	Verify user is able to log into application with InValid credentials	IBM Cloud services	1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter InValid username/email in Email text box 4.Enter Invalid password in password text box 5.Click on login button	Username:815119106038 @smartinternz.com password: Ibmproject	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass	
Designing the circuit _TC__01	Functional	Backend	Creating the design flow and making the proper connection to get the output	Tinkercad	1 .Creating an account in tinkercad. 2.Making the circuit connections . . 3.Editing the program as per the circuit . 4. simulating the project.	LED ON and OFF with Parameter values	The led must be able to operate with the program. The parameters must be obtained.	Not working as expected	Fail	Conn

				Date	10-Nov-22					
				Team ID	PNT2022TMID46185					
				Project Name	Project -signs with smart connectivity for better road safety					
				Maximum Marks	4 marks					
Test case ID	**Feature Type**	**Component**	**Test Scenario**	**Pre-Requisite**	**Steps To Execute**	**Test Data**	**Expected Result**	**Actual Result**	**Status**	**Comm**

Designing the circuit_TC_02	Functional	Backend	Creating the design flow and making the proper connection to get the output	Node-RED	1 .Downloading all the dashboard nodes required. 2.Picking and pasting the dashboard nodes 3.Connecting the nodes 4.Deploying the design flow	Temperature=" " humitity=" "	The Node Red must be able to get the real time values of temperature,pH and turbidity.	Working as expected	Pass	
Designing the circuit_TC _03	Functional	Backend	Creating the design flow and making the proper connection to get the output	Node-RED	1.Downloading all the dashboard nodes required. 2.Picking and pasting the dashboard nodes 3.Connecting the nodes 4.Deploying the design flow	Temperature=" " humitity=" "	The Node Red must be able to get the real time values of temperature,pH and turbidity.	Working as expected	Pass	
Create a program suitable for the circuit and also compile and execute the programs_TC_01	Functional	Backend	Developing the python script to get the parameter values	Python 3.7	1 .Installing python version 3.7.0 2.Developing the python code 3.Resolving the errors 4.Executing the program 5.Obtaining the output	Temperature=" " humitity=" "	The program must be executed without any error and the values must be obtained.	Working as expected	Pass	
Create a program suitable for the circuit and also compile and execute the programs._Tc_02	Functional	Backend	Developing the python script to get the parameter values	Python 3.7	1.Installing python version 3.7.0 2.Developing the python code 3.Resolving the errors 4.Executing the program 5.Obtaining the output	Temperature=" " humitity=" "	The program must be executed without any error and the values must be obtained.	Working as expected	Pass	
Create a program suitable for the circuit and also compile and execute the programs_TC_03	Functional	Backend	Developing the python script to get the parameter values	Python 3.7	1.Installing python version 3.7.0 2.Developing the python code 3.Resolving the errors 4.Executing the program 5.Obtaining the output	Temperature=" " humitity=" "	The program must be executed without any error and the values must be obtained.	Working as expected	Pass	
Create a program suitable for the circuit and also compile and execute the programs_TC_04	Functional	Backend	Developing the python script to get the parameter values	Python 3.7	1 .Installing python version 3.7.0 2.Developing the python code 3.Resolving the errors 4.Executing the program 5.Obtaining the output	Temperature=" " humitity=" "	The program must be executed without any error and the values must be obtained.	Working as expected	Pass	
connect the output values to the cloud services by using NODE RED._TC_01	Functional	Backend	Connecting the python code with the node red by providing the watson credentials	IBM IOT Watson platform and Node-RED	1.Provide the watson credentials in the python script 2.Verify the values are displayed in node red 3.Values must be obtained in watson,Node-red and python	Temperature=" " humitity=" "	The Temperature,pH and Turbidity values must be obtained.	Not working as expected	Fail	Not autho
connect the output values to the cloud services by using NODE RED_TC_02	Functional	Backend	Connecting the python code with the node red by providing the watson credentials	IBM IOT Watson platform and Node-RED	1 .Provide the watson credentials in the python script 2.Verify the values are displayed in node red 3.Values must be obtained in watson,Node-red and python	Temperature=" " humitity=" " "	The Temperature,pH and Turbidity values must be obtained.	Working as expected	Pass	

				Date	10-Nov-22					
				Team ID	PNT2022TMID46185					
				Project Name	Project – signs with smart connectivity for better road saety					
				Maximum Marks	4 marks					
Test case ID	**Feature Type**	**Component**	**Test Scenario**	**Pre-Requisite**	**Steps To Execute**	**Test Data**	**Expected Result**	**Actual Result**	**Status**	**Comm**
connect the output values to the cloud services by using NODE RED_TC_03	Functional	Backend	Connecting the python code with the node red by providing the watson credentials	IBM IOT Watson platform and Node-RED	1 .Provide the watson credentials in the python script 2.Verify the values are displayed in node red 3.Values must be obtained in watson,Node-red and python	Temperature=" " humitity=" "	The Temperature,pH and Turbidity values must be obtained.	Working as expected	Pass	

Make the data's store in IBM cloudant database_TC_01	Functional	Storage	Creating the cloudant DB in IBM cloud services to store the parameter values.	IBM Cloudant DB	1.Create the cloudant dB in IBM cloud services 2.Connect the Cloudant node to the design flow 3.Open cloudant and check whether the values are stored.		The parameters values must be stored in the cloudant DB.	Not working as expected	Fail	Unable t
Make the data's store in IBM cloudant database_TC_02	Functional	Storage	Creating the cloudant DB in IBM cloud services to store the parameter values.	IBM Cloudant DB	1 .Create the cloudant dB in IBM cloud services 2.Connect the Cloudant node to the design flow 3.Open cloudant and check whether the values are stored		The parameters values must be stored in the cloudant DB.	Working as expected	Pass	
Make the data's store in IBM cloudant database_TC_03	Functional	Storage	Creating the cloudant DB in IBM cloud services to store the parameter values.	IBM Cloudant DB	1 .Create the cloudant dB in IBM cloud services 2.Connect the Cloudant node to the design flow 3.Open cloudant and check whether the values are stored	Temperature=" " humitity=" "	The parameters values must be stored in the cloudant DB.	Working as expected	Pass	
Make the data's store in IBM cloudant database_TC_04	Functional	Storage	Creating the cloudant DB in IBM cloud services to store the parameter values.	IBM Cloudant DB	1.Create the cloudant dB in IBM cloud services 2.Connect the Cloudant node to the design flow 3.Open cloudant and check whether the values are stored	Temperature=" " humitity=" "	The parameters values must be stored in the cloudant DB.	Working as expected	Pass	
Make the data's store in IBM cloudant database_TC_05	Functional	Storage	Creating the cloudant DB in IBM cloud services to store the parameter values.	IBM Cloudant DB	1.Create the cloudant dB in IBM cloud services 2.Connect the Cloudant node to the design flow 3.Open cloudant and check whether the values are stored	Temperature=" " humitity=" "	The parameters values must be stored in the cloudant DB.	Working as expected	Pass	
Connects the cloud data with the authorities communication device. _TC_01	Functional	User Interface	Making the parameter values visible in the mobile through MIT app inventor.	MIT app inventor	1.Install MIT Ai2 companion app in mobile phone. 2. Scan QR code with mobile device. 3.Check whether the values can be obtained in the mobile.	Temperature=" " humitity=" "	The parameter values must be visible in the mobile application.	Not working as expected	Fail	Error
Connects the cloud data with the authorities communication device._TC_02	Functional	User Interface	Making the parameter values visible in the mobile through MIT app inventor.	MIT app inventor	1.Install MIT Ai2 companion app in mobile phone. 2. Scan QR code with mobile device. 3.Check whether the values can be obtained in the mobile.	Temperature=" " humitity=" "	The parameter values must be visible in the mobile application.	Working as expected	Pass	

Date	10-Nov-22	
Team ID	PNT2022TMID46185	
Project Name	Project – SIGNS with smart connectivity for better road safety	
Maximum Marks	4 marks	

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets
Connects the cloud data with the authorities communication device._TC_03	Functional	User Interface	Making the parameter values visible in the mobile through MIT app inventor.	MIT app inventor	1.Install MIT Ai2 companion app in mobile phone. 2. Scan QR code with mobile device. 3.Check whether the values can be obtained in the mobile.	Temperature=" " Turbidity=" " ph=" "	The alert messages must be sent to the authorities with the exact values.	Working as expected	Pass	
Alerts has to be sent to the authorities _TC_01	UI	Display	Making the alert messages reach the authorities with the parameter values.	Messaging Tool	1 .Sign in with messaging platforms like Fast SMS. 2.Connect the values and provide the thereashold values. 3.Provide contact numbers or mail id . 4. Check for the alert messages	Alert!!! The water is not fit to use	The alert messages must be sent to the authorities with the exact values.	Not working as expected	Fail	Error

Alerts has to be sent to the authorities _TC_02	UI	Display	Making the alert messages reach the authorities with the parameter values.	Messaging Tool	1.Install MIT Ai2 companion app in mobile phone. 2. Scan QR code with mobile device. 3.Check whether the values can be obtained in the mobile.	Alert!!! The water is not fit to use	The alert messages must be sent to the authorities with the exact values.	Not working as expected	Pass	

9 RESULTS

Performance Metrics

			NFT - Risk Assessment						
S.No	Project Name	Scope/feature	Functional Changes	Hardware Changes	Software Changes	Impact of Downtime	Load/Volume Changes	Risk Score	Justification

1	signs with smart connectivity for better road safety	New	Low	Low	Low	Downtime does not affect the performance much.The errors can be resolved within a short duration of time.	>5 to 10%	ORANGE	As the sensors senses the parameters continuosly ,there will not be any delay.As the sensors are well protected ,there is a low probability of physical damage.
			NFT - Detailed Test Plan						
			S.No	Project Overview	NFT Test approach	ssumptions/Dependencies/Risk	Approvals/SignOff		
			1	smart connectivity for better road safety	LOAD TEST ENDURANCE TEST	The project is capable of dealing with large amount of data (i.e) load. Congestion can be controlled and the system can operate efficiently.	Approved		
			End Of Test Report						
S.No	Project Overview	NFT Test approach	NFR - Met	Test Outcome	GO/NO-GO decision	Recommendations	Identified Defects (Detected/Closed/Open)	Approvals/SignOff	
1	smart connectivity for better road safety	LOAD TEST ENDURANCE TEST	YES	The parameter values of Temperature, hummitity can be obtained continously and the alert messages whenever the smartboard to connect the all aurdino board	GO	The rechargable sensors can be used during the manufacturing of this system.	Closed	Approved	

10 ADVANTAGES & DISADVANTAGES

ADVANTAGES

Multimodal sensors and edge computing help speed up the flow of traffic with real-time processing, reducing congestion and emissions. Smart road technology can assist in optimizing traffic flow and managing road conditions, creating a more sustainable environment within cities

DISADVANTAGES

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, and construction has associated environmental costs.

CONCLUSION

In present Systems the road signs and the speed limits are Static. But the road signs can be changed in some cases. We can consider some cases when there are some road diversions due to heavy

traffic or due to accidents then we can change the road signs accordingly if they are digitalized.This project proposes a system which has digital sign boards on which the signs can be changed dynamically.If there is rainfall then the roads will be slippery and the speed limit would be decreased.There is a web app through which you can enter the data of the road diversions, accident prone areas and the information sign boards can be entered through web app. This data is retrieved and displayed on the sign boards accordingly.

12 FUTURE SCOPE

1. solar powered roadways

Photovoltaic cells are embedded within hexagonal panels made of tempered glass, which are used to pave roads. These panels contain LEDs, microprocessors, snow-melting heating devices and inductive charging capability for electric vehicles when driving. Glass is renewable and can be engineered to be stronger than steel, and to allow cars to stop safely even when traveling at high speeds. While this idea has gained widespread support, scalability is a challenge as it remains expensive.

2. Smart Roads

Specially engineered roadways fitted with smart features, including sensors that monitor and report changing road conditions, and WiFi transmitters that provide broadband services to vehicles, homes and businesses. The smart road can also charge electric cars as they drive.

3. Glow in the dark roads

Glowing markers painted onto existing roadway surfaces use a photo-luminescent powder that absorbs and stores daylight. The 500m long strips glow for 8 hours after dark. This technology is still in the testing phase, and the glow is not yet consistent, but it could be more cost-effective than traditional road lighting technologies.

4. Interactive lights

Road lights activated by motion sensors to illuminate a particular section of the road as cars approach. The lights dim once the car passes. Suited for roads with less traffic, interactive lights provide night visibility as needed and reduce energy wastage when there are no cars. One design, developed in Holland, uses the wind generated by passing vehicles to power lights.

5. Electric priority lane for charging electric vehicles

Embedded cables generate magnetic fields that charge electric vehicles while driving. A receiver coil in the vehicle picks up electromagnetic oscillations from a transmitter coil embedded in the road and converts them to AC, which can then power the car. Inductive charging technology

already exists for static cars, but future wireless technology could charge batteries while in motion, providing distance range solutions for electric vehicles which travel longer journeys.

6. Weather detection

Networks of AI-integrated sensors detect weather conditions that impact road safety. Road Weather Information Systems (RWIS) in use today are limited because they only collect data from a small set of weather stations. A larger future network could use automated weather stations to collect atmospheric and weather data and instantly upload it to the cloud. Dynamic temperature-sensitive paint could be used to highlight invisible roadway conditions like black ice.

7. Traffic detection

Data that helps travelers plan their routes. Sensors lining highways monitor traffic flow and weight load, warn drivers of traffic jams, and automatically alert the authorities about accidents. Fiber-optic cables embedded in the road detect wear and tear, and communication between vehicles and roads can improve traffic management. For example, rapid flow technologies use artificial intelligence (AI) to manage traffic lights, which respond to each other and to cars. Traditional systems were pre-programmed to optimize flow around peak journey times, new technologies are able to process and optimize flows in real times

13 APPENDIX

Source Code

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random


#Provide your IBM Watson Device Credentials
organization = "rv07c6"
deviceType = "riverwaterquality-22_23"
deviceId = "123456"
authMethod = "token"
authToken = "wQ_)43L5c0@ku8)sgd"

# Initialize GPIO
```

```
def myCommandCallback(cmd):
   print("Command received: %s" % cmd.data['command'])
   status=cmd.data['command']
   if status=="lighton":
      print ("led is on")
   else :
      print ("led is off")
   #print(cmd)



try:
         deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method":
authMethod, "auth-token": authToken}
         deviceCli = ibmiotf.device.Client(deviceOptions)
         #..............................................

except Exception as e:
         print("Caught exception connecting device: %s" % str(e))
         sys.exit()

# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type
"greeting" 10 times
deviceCli.connect()

while True:
      #Get Sensor Data from DHT11
      temp=random.randint(0,100)
      ph=random.randint(0,14)
      turb=random.randint(0,100)
      data = { 'temp' : temp, 'ph': ph,'turb' :turb }
      #print data
      def myOnPublishCallback():
         print ("Published Temperature = %s C" % temp, "ph = %s %%" % ph,"turbidity = %s NTU " %
turb ,"to IBM Watson")

      success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)
      if not success:
```

```
        print("Not connected to IoTF")
    time.sleep(1)
    deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

GitHub &Project Demo Link

https://github.com/IBM-EPBLhttps://github.com/IBM-EPBL/IBM-Project-31047-1660195318

YOU TUBE LINK: