Real-Time River Water Quality Monitoring and Control System

TEAM ID: PNT2022TMID11075

TEAM LEADER :

SANTHOSH KUMAR S

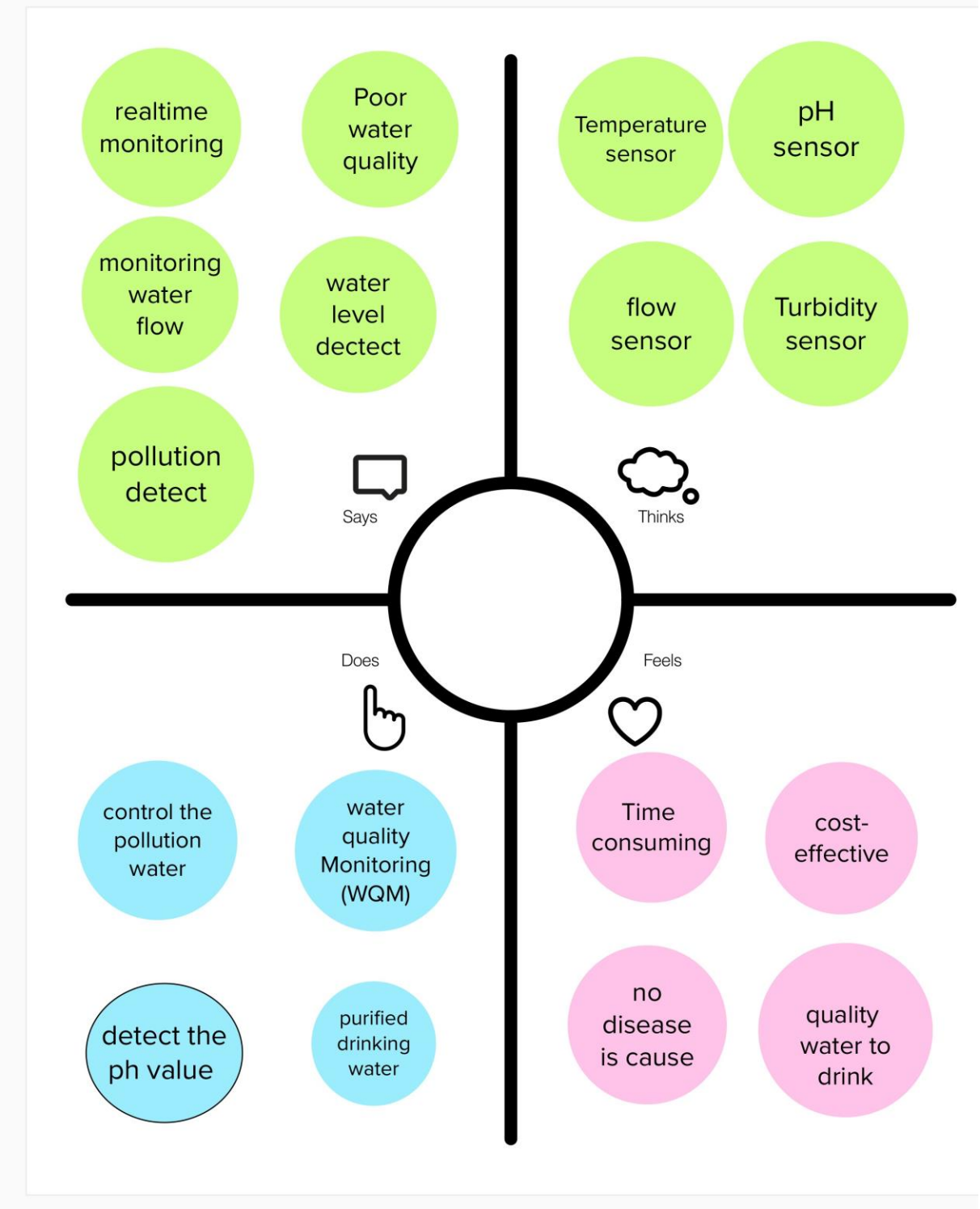
TEAM MEMBERS :

Santhosh S

Sudharsan S

Stanly mathew raj A

EMPATHY MAP


realtime monitoring
Poor water quality
Temperature sensor
pH sensor
monitoring water flow
water level dectect
flow sensor
Turbidity sensor
pollution detect
Says
Thinks
Does
Feels
control the pollution water
water quality Monitoring (WQM)
Time consuming
cost-effective
detect the ph value
purified drinking water
no disease is cause
quality water to drink

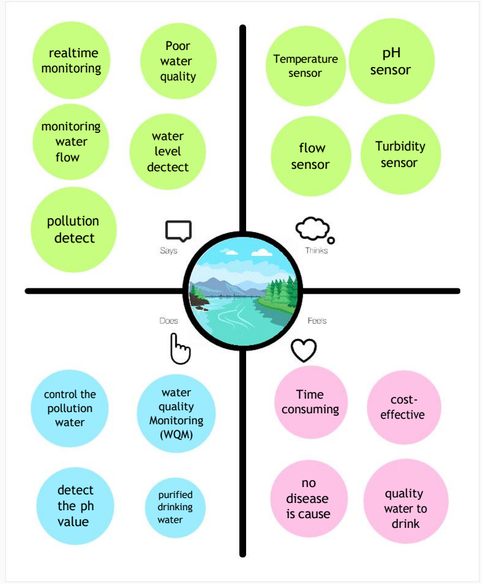
Edit this template
Right-click to unlock

Empathy Map

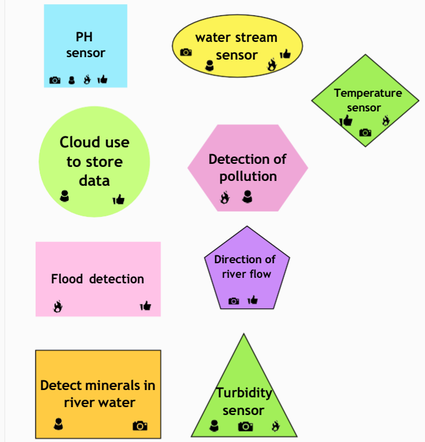
Dive into the mind of the user for focused product development

Build empathy and keep your focus on the user by putting yourself in their shoes.

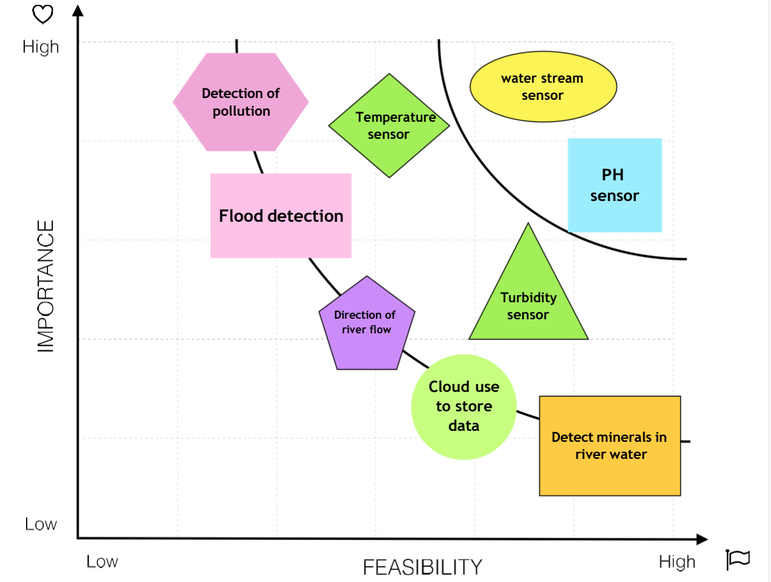
SNO	NAME	POSITION	COLLEGE
1	SANTHOSH KUMAR S	TEAM LEADER	K.RAMAKRISHNAN COLLEGE OF ENGINEERING
2	SUDHARSAN S	TEAM MEMBER 1	K.RAMAKRISHNAN COLLEGE OF ENGINEERING
3	SANTHOSH S	TEAM MEMBER 2	K.RAMAKRISHNAN COLLEGE OF ENGINEERING
4	STANLY MATHEW RAJ A	TEAM MEMBER 3	K.RAMAKRISHNAN COLLEGE OF ENGINEERING



BIG IDEAS



Idea Prioritization



NEED STATEMENT

The system	finds a way for	**Supervising and Monitoring**	The real time river water so that	**Quality & flow can be maintained**

The system finds a way for supervising and monitoring the real time river water so that quality & flow can be maintained

Real-Time River Water Quality Monitoring and Control System

TEAM MEMBERS :-

Santhosh Kumar S

Santhosh S

Sudharsan S

Stanly mathew raj A

LITERATURE SURVEY :-

Pranita Mahajan Published on "International Conference on IoT based Control Networks and Intelligent Systems (ICICNIS 2020)".The quality of potable water is crucial for socioeconomic factors. To guarantee the purity of drinking water, many researchers created numerous techniques. Manual sample collection and laboratory manual analysis are how the conventional system operates.creating lag time and human mistake. The analysis is not performed at the user location, thus existing systems may slow operations while reducing errors.Users require a system that will dynamically monitor and guarantee water quality. The suggested system has a number of sensors to Based on pH, temperature, conductivity, turbidity, ORP, nitrate, and free residual chlorine, evaluate and confirm the water's quality.Sensors gather data, which is then sent for additional processing. General users can use the system's LEDs to Determine the water's purity right away.

Mrs. R.J. Sapkal Published on "INTERNATIONAL RESEARCH JOURNAL OF ENGINEERING AND TECHNOLOGY".This essay explains The key drivers behind the requirement for effective and efficient water level monitoring and regulation of water quality in flat systems are to maintain the sustainability and health of the human resource base and to minimise the consumption of water for domestic use.

The water system has a significant negative impact on the natural environment as a result of climate change and fluctuation. Only in water laboratories are incredible methods for sample collection, testing, and analysis used.However, gathering, analysing, and quickly disseminating information to the appropriate people so they may make informed decisions when they are needed is not always simple. This research presents a water sensor system prototype for societal water level and quality monitoring.

Shudong Wang Published on College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou,Gansu, 730050, China Real-time monitoring of water quality is essential since human activity and production have contributed to varied degrees of water contamination since the turn of the twenty-first century. The approach for monitoring three water quality parameters—water temperature, PH level, and turbidity—that is suggested in this research is based on STM32. The STM32 series single-chip microcomputer's extended circuit, which includes PH sensor control circuit module, temperature sensor circuit module, wireless network communication circuit module, turbidity sensor circuit module, etc., serves as the foundation for the water quality monitoring technique. Then, using the C programming language, the PC programme created using the virtual instrument design language platform and the MCU data acquisition programme were created.

A.N.Prasad Published on School of Engineering and Physics,University of the South Pacific,Laucala, Fiji Islands Modern research uses remote sensing (RS) and internet of things (IoT) technologies to monitor, gather, and analyse data from far-off sites. The quality of water that is available to people has significantly declined as a result of the enormous rise in global industrial output, rural-urban migration, overuse of land and marine resources, and other factors. The widespread use of fertilisers in agriculture as well as other chemicals in industries like mining and construction has significantly lowered the quality of water worldwide. Water is a need for human survival, hence measures must be taken to

rigorously monitor its quality before it is made available for consumption in a community.

N. Thirupathi Rao Published on International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8 Issue-5 March, 2019.The manual laboratory testing of samples for drinking water quality characteristics like turbidity, pH, conductivity, and temperature, among others, might take time. In order to address this, an effort has been made to design a smart and affordable IoT system in the current paper.Temperature, turbidity, pH, and conductivity are the variables taken into account when evaluating the water quality. The aforementioned metrics are measured using sensors submerged in water samples.The Raspberry Pi Unit received the detected data from the sensors. The parameters of the sensed data were compared to the baseline values included in the Raspberry Pi unit. The IOT can access the data on the Raspberry Pi device (cloud).

Phil Jordan Published on School of Geography and Environmental Sciences, Ulster University, Coleraine, United Kingdom, 2 Agri-Environment Branch, Agri-Food and Biosciences Institute, Belfast, United KingdomAfter a series of study visits, this paper examines improved methods for monitoring river water quality in north-western Europe (11 sites in 7 countries). Options were developed and assessed for their potential to meet specific water quality monitoring objectives with an emphasis on bringing about behavioural change based on the data gained. Numerous parameters and nutrients were sampled in sub-hourly intervals in autonomous, high-specification, bank-side or mobile laboratories as part of the monitoring programmes, which ranged from increased grab sampling and laboratory analysis to these programmes. Out of all the cases analysed, only one programme was able to quickly pinpoint the influences that had led to stakeholders' changing behaviour. This was mostly due to the other initiatives' emphasis on top-down policy reform or surveillance rather than their targeted approach to behaviour change.

Vaishnavi V Published on Department Electronics & Telecommunication Engineering, Mtech(VLSI), Bapurao Deshmukh College of Engineering,Sevagram, wardha_442102(M.S.), India.One of the main concerns for the green globalisation is water contamination. Real-time quality monitoring is required to guarantee the supply of drinking water is secure. In this study, we propose the design and creation of a low cost system for internet of things (IoT) real-time water quality monitoring. The system, which consists of numerous sensors, is used to measure the water's physical and chemical characteristics. It is possible to measure the water's parameters, including temperature, PH, turbidity, and flow sensor. The core controller is capable of processing the measured values from the sensors. A core controller can be created using the Arduino model. Finally, utilising a WI-FI setup, the sensor data may be seen online.

Alexander T. Demetillo Published on School of Engineering, University of San Carlos, Cebu City 6000, Philippines.An affordable, real-time water quality monitoring system that can be used in far-off rivers, lakes, coastal areas, and other water bodies is presented in this study. Off-the-shelf electrochemical sensors, a microcontroller, a wireless communication system, and the bespoke buoy make up the system's basic components. It measures pH, dissolved oxygen, and water temperature during a pre-set time period. To better serve interested end users, the built prototype disseminates the obtained data in graphical and tabular representations via a tailored web-based portal and preregistered mobile phones. The stability of the buoy in challenging environmental circumstances, system energy consumption, data transmission effectiveness, and web-based information display were all rigorously assessed to test the system's efficacy.

Mithila Barabde Published on International Journal of Innovative Research in Computer and Communication Engineering One of the main concerns for the green globalisation is water contamination. Water characteristics including pH,

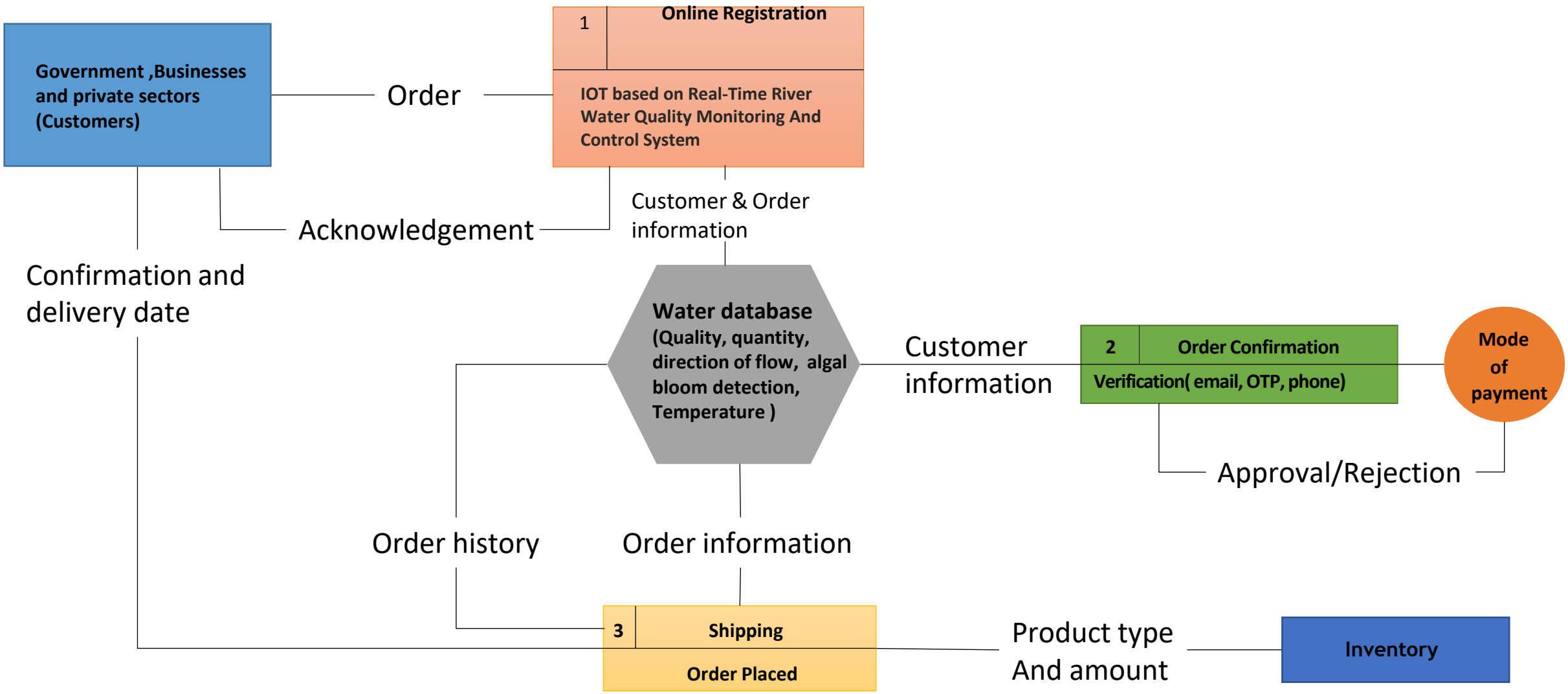
turbidity, conductivity, and other variables must first be estimated in order to prevent pollution because variations in these parameters' values indicate the presence of contaminants. Water parameters are currently determined using chemical tests or laboratory tests, where the testing apparatus is stationary and samples are fed into the apparatus. As a result, the current technique for checking the quality of the water is manual, laborious, and time-consuming. The testing instrument can be submerged in river water to increase frequency, and remote pollution detection is also an option. A sensor-based water quality monitoring system is suggested in this research.

Deepthi N Published on International Journal of Progressive Research in Science and Engineering ,Volume-1, Issue-4, July-2020 .To confirm if the quality of the water is good or not, the quality should be adequately monitored. In this method, we discuss a plan and the creation of a low-cost system for assessing the water quality utilising internet of things technology. The most useful tools we utilised to measure various water properties were the sensors. Here, the primary variables are PH, temperature, and turbidity. We will use the core controller to monitor the values when we have obtained them from the various sensors.Keywords: sensors, core controller, PH, turbidity, temperature, IOT.

CUSTOMER JOURNEY MAP

PHASES	MOTIVATION	INFORMATION GATHERING	ANALZES VARIOUS PRODUCT	CHOOSE THE MOST EFFICIENT PRODUCT	PAYMENT
ACTIONS	Wants to reduce the river contamination	Wants to choose an efficient product to monitor and prevent contamination	Other water quality testing systems	Iot based sensor system (or) more efficient than classic system	After the product satisfication
TOUCHPOINTS	Buy as feel excited	After installation,people no need to worry about water quality	User amuse by various products which are available	After getting this no need to worry about water contamination and water quality	After find the product worthy people will buy it
CUSTOMER FEELING	☺	☺	☺	☺	☺
CUSTOMER THOUGHTS	Customer thinks it will helpful for better status of river water	It will leads for longer time	Customer things alter solution will available	The product choosing will be easy and comfortable for them	They think the product will be user friendly
OPPORTUNITIES	The people get better quality of river water	Customer know about the process of the system	will aware about other products in the market	People will get knowledge about the product and differentiate which is best	People utilize the product

DATA FLOW DIAGRAM:



Functional Requirements:

Following are the functional requirements of the proposed solution.

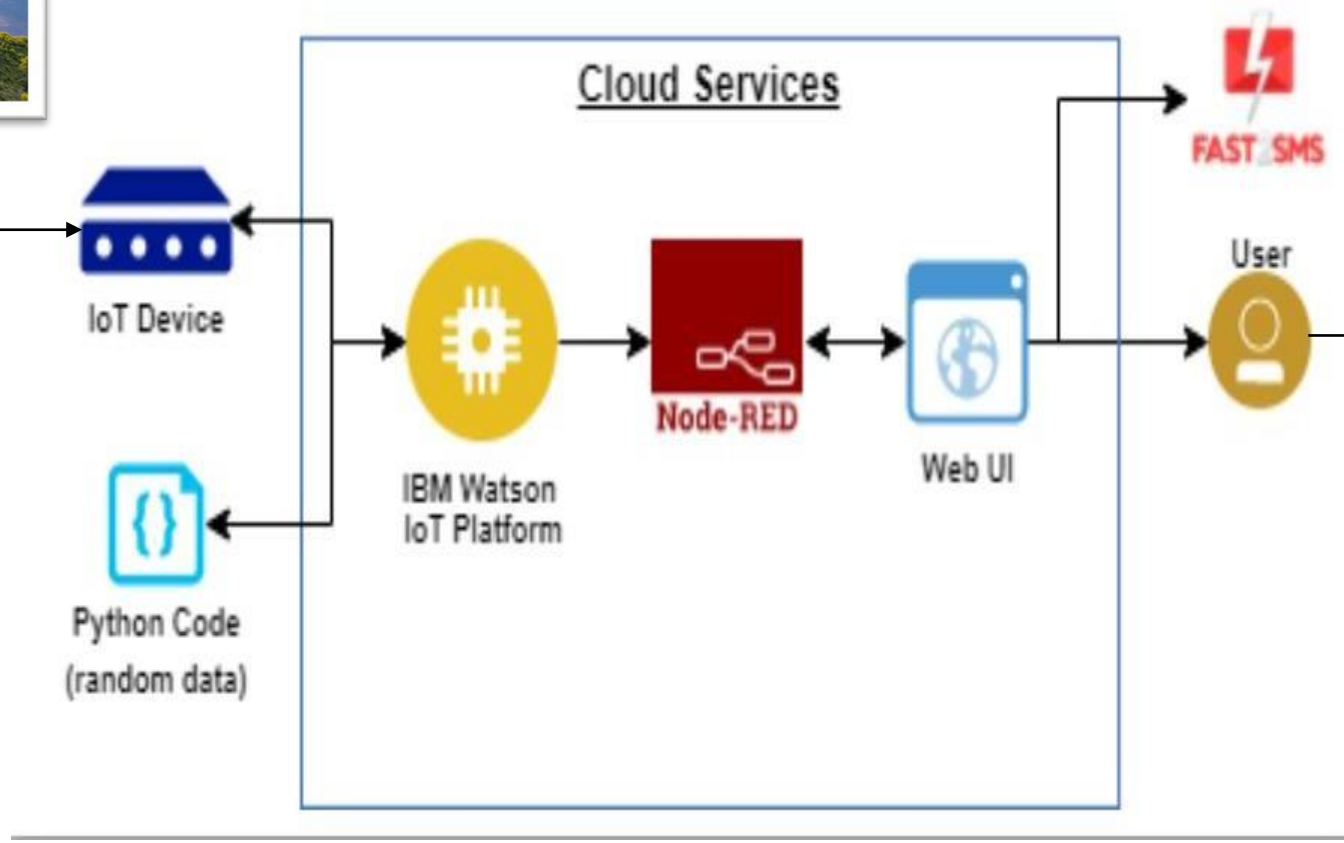
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Requirements	Monitoring river water quality, water flow, humidity, and temperature to control the algal bloom
FR-2	User Registration	Manual Sign-Up using a Website or Gmail
FR-3	User Confirmation	OTP authentication through phone, email, and confirmation
FR-4	Payments options	Bank transfers, credit cards, debit cards, and ATMs with UPI
FR-5	Product Delivery and installation	Take away Free Installation and 1 year Warranty
FR-6	Product Feedback	Through a website, a phone conversation, and Gmail

Non-functional Requirements:

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

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Have self-explanatory products that are easy to use and have clear product instructions. and Have an easy-to understand guidebook. simpler to use
NFR-2	**Security**	Application security requires two-step authorization. The user's needs will determine how passwords and passkeys are assigned. The network must contain cloud data, condensing it to be Avoid real-time avoidance, and keep an eye on the board at all times.
NFR-3	**Reliability**	Hardware needs to be checked and maintained regularly. Periodic software updates are possible. Any system breakdown will result in an immediate alarm.
NFR-4	**Performance**	The Application must give accurate results, have a user-friendly interface, and improve the user experience.
NFR-5	**Availability**	Depending on the requirements of the user, all required functions will be offered. When a user requests a feature or makes a tweak, all features will be made available.
NFR-6	**Scalability**	Regardless of size, the product must fill the entire river's space. The product is based on monitoring water quality, flow, humidity, and temperature, as well as controlling algal blooms.

TECHNOLOGY ARCHITECTURE



Problem-Solution fit canvas 2.0

Purpose / Vision

1. CUSTOMER SEGMENT(S) — CS

- Aqua ponics
- Dam safety organisation (SDSO)
- Fish culture (Pisciculture)
- Wholesaler of mineral water

5. CUSTOMER CONSTRAINTS — CC

- Sensors are used
- Compact in size
- Clouds for storage purpose
- Consumes low power
- Without proper network connecion the device be used
- Adaptable for the users

8. AVAILABLE SOLUTIONS — AS

- The technology develops a means to supervise and track river water in real time so that quality and flow can be maintained to use less electricity and deliver at a lower cost
- The device will be small and simple to operate and cons is Device use without sufficient network connection

2. JOBS-TO-BE-DONE / PROBLEMS — J&P

- To control the flow of water using IOT
- To identify the ph value and mineral content in the water
- To identify the presents of algal bloom in the tank or water bodies
- The quality , quantity and temperature of the water can be maintained

6. PROBLEM ROOT CAUSE — RC

- It involves improper upkeep of the water supply and inappropriate upkeep of the people.
- Lack of system administration and upkeep is the problem.
- It uses a lot of electricity.

9. BEHAVIOUR — BE

- To recognise the tank's algae growth. checks the PH level, mineral content, temperature, water flow direction, and water quantity.
- These are portable and are easily maintainable.
- It uses less data and power.Additionally, it might serve as a reference for the best safety steps to take.

3. TRIGGERS — TR

- They are able to recognise the issue with the water without anyone's assistance.
- It uses little energy and is small in size. Customers will find it easy to use

4. EMOTIONS: BEFORE / AFTER — EM

- Before :Anxity,time consumption and unaware of things
- After:aware of things ,less time consumption and pleasure

7. YOUR SOLUTION — SL

- The system finds a way for supervising and monitoring the real time river water so that quality & flow can be maintained
- To consume less powerconsumption and to provide in cheaper cost
- The device will be in compact size and user friendly to use

10. CHANNELS of BEHAVIOUR — CH

ONLINE

- The cloud storage can be used to regulate water flow.

OFFLINE

- The proposed system includes a number of sensors to test and guarantee the water's quality based on factors including pH, temperature, conductivity, turbidity, and ardunio.

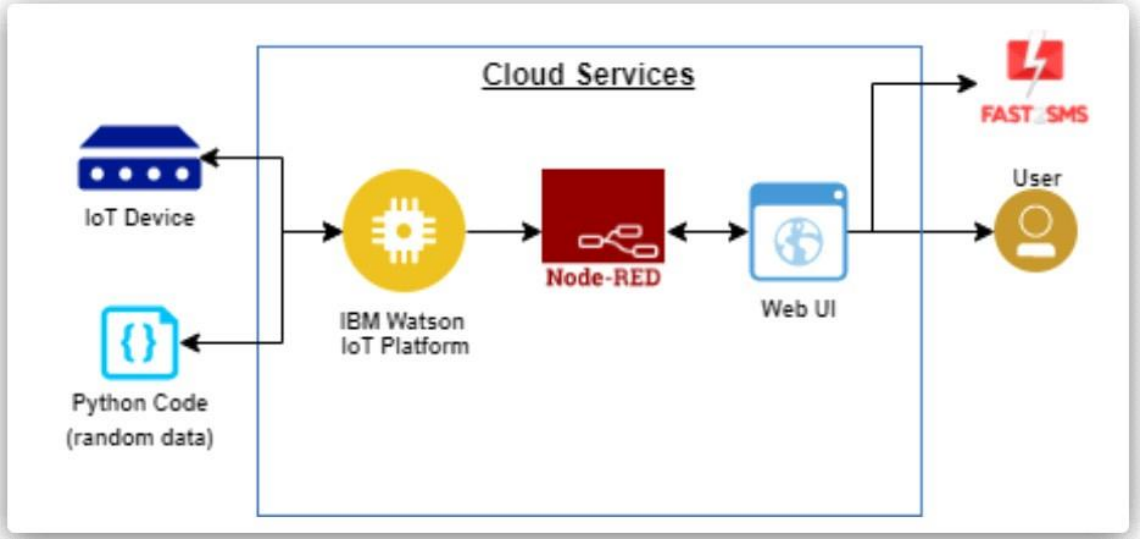
AMALTAMA

Proposed Solution Template:

Project team shall fill the following information in proposed solution template.

S.No.	Parameter	Description
•	Problem Statement (Problem to be solved)	To monitor the water parameters such as turbidity ,pH, dissolved solvents and To Control the Algal bloom
•	Idea / Solution description	Monitoring water parameters with sensors and control measures by ultrasonic frequencies
•	Novelty / Uniqueness	They exclusively use the Raspberry Pi in 2019, however we utilise Arduino, which requires less power and fewer code. Water Parameters Based on Sensors Identification of biological and chemical changes in water by monitoring
•	Social Impact / Customer Satisfaction	It gives the consumer knowledge about the river water quality that could aid them in the water usage should be based on its Rural residents can utilise this product because of its high quality. It uses a very small amount of power.
•	Business Model (Revenue Model)	The monitoring system could be sold in the market for the purpose of testing water quality.
•	Scalability of the Solution	The model could be scaled according to size of the water body about to be tested

Proposed solution



PROJECT PLANNING PHASE

PROJECT MILESTONE

S.NO	ACTIVITY TITLE	ACTIVITY DESCRIPTION	DURATION
1	Understanding the project requirement	Assign the team members and create repository in the Github, Assign the task to each members and teach how to use and open and class the Github and IBM career education	1 WEEK
2	Starting of project	Advice students to attend classes of IBM portal create and develop an rough diagram based on project description and gather of information on IOT and IBM project and team leader assign task to each member of the project	1 WEEK
3	Attend class	Team members and team lead must watch and learn from classes provided by IBM and NALAYATHIRAN and must gain	4 WEEK

		access of MIT license for their project	
4	Budget and scope of project	Budget and analyze the use of IOT in the project and discuss with team for budget prediction to predict the favorability for the customer to buy	1 WEEK

Project Planning Phase

Sprint Delivery Plan

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

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	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Santhosh kumar S
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Stanly Mathew Raj A
Sprint-2		USN-3	As a user, I can register for the application through Facebook	2	Low	Sudharsan S
Sprint-1		USN-4	As a user, I can register for the application through Gmail	2	Medium	Santhosh S
Sprint-1	Login	USN-5	As a user, I can log into the application by Entering email & password	1	High	Stanly Mathew raj A

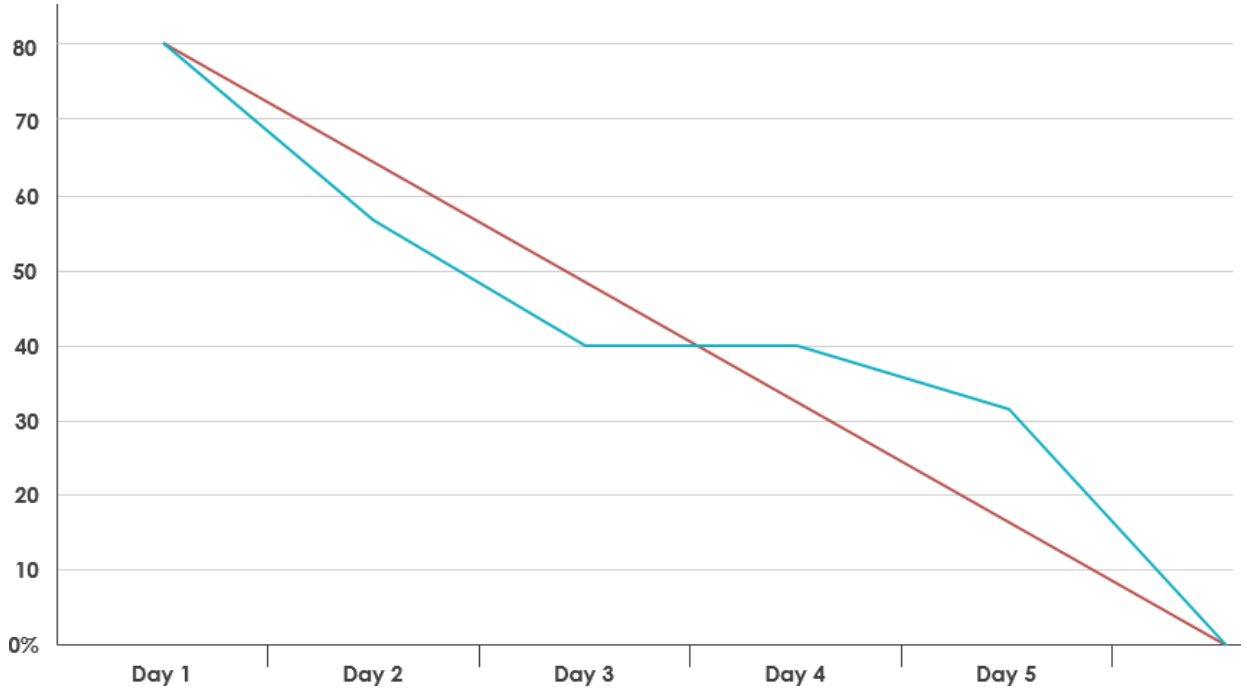
Project Tracker, Velocity & Burndown Chart: (4 Marks)

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	5 Nov 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	30	8 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	49	14 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	50	19 Nov 2022

Velocity:

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

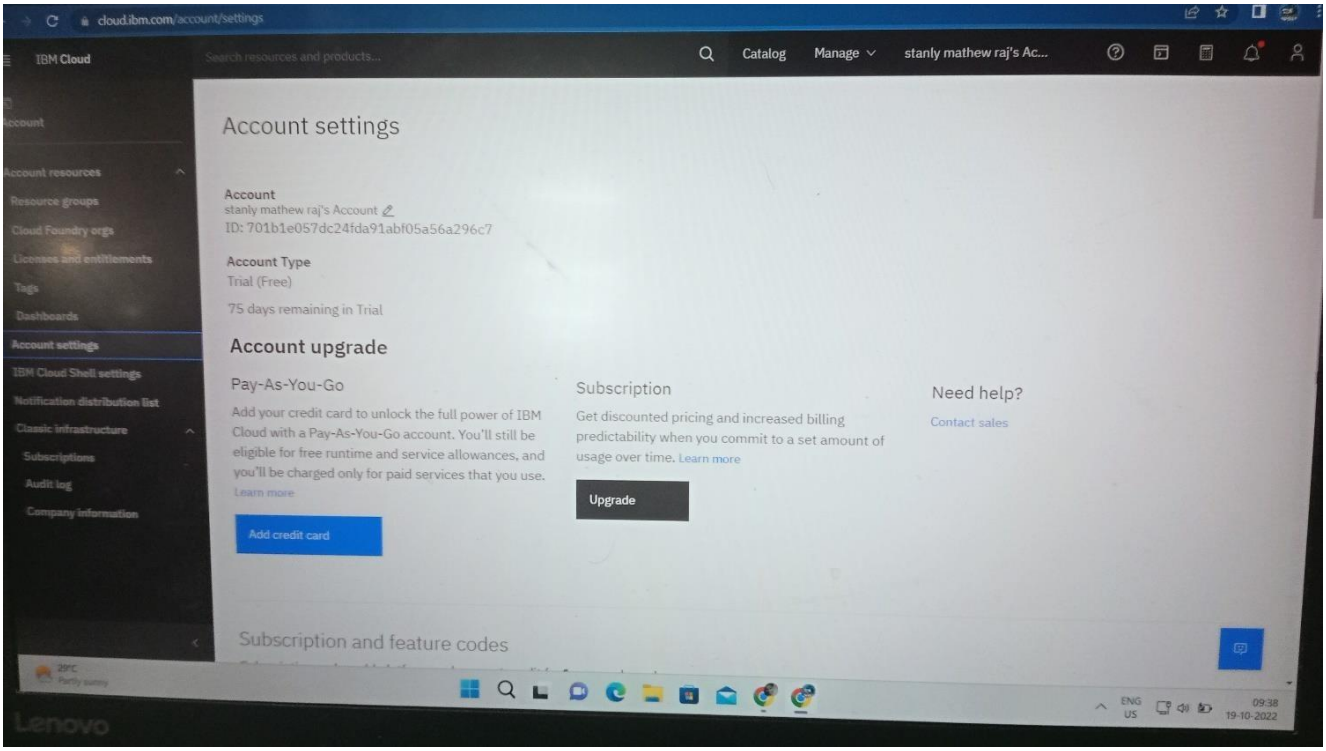
Burndown Chart:



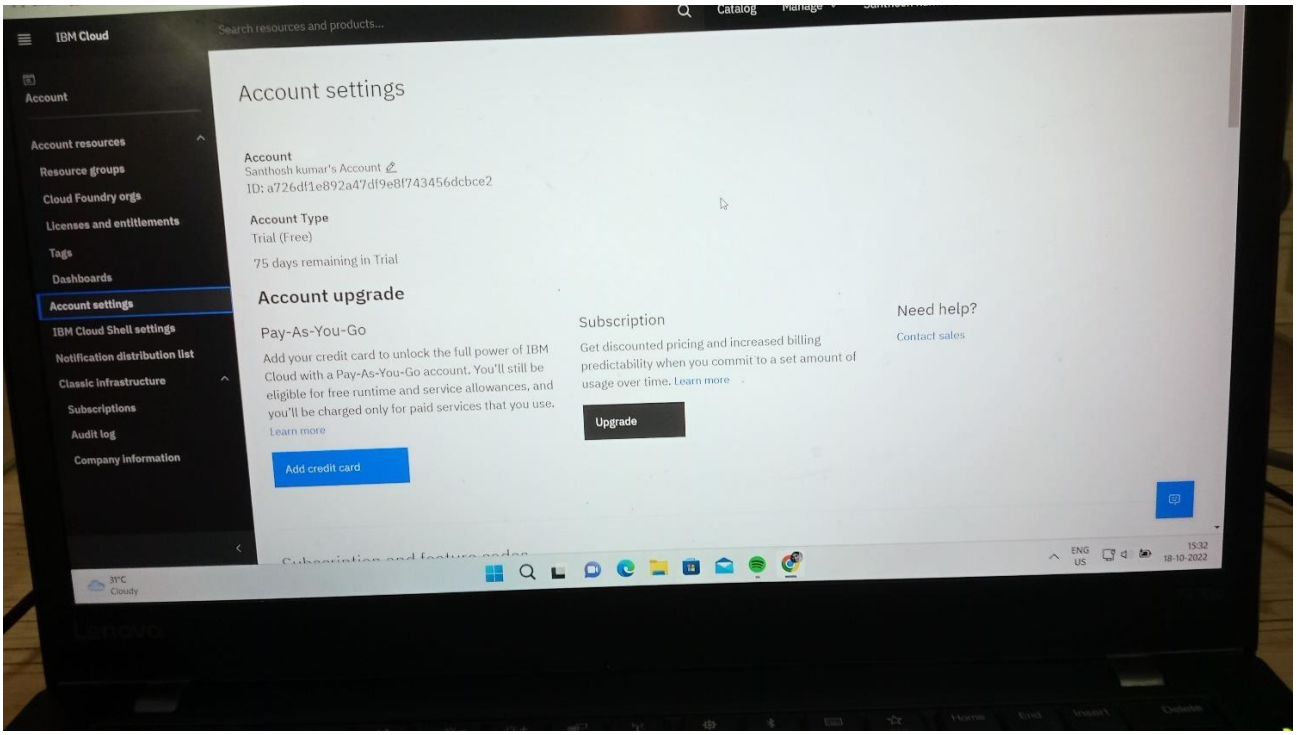
.

IBM CLOUD SERVICE

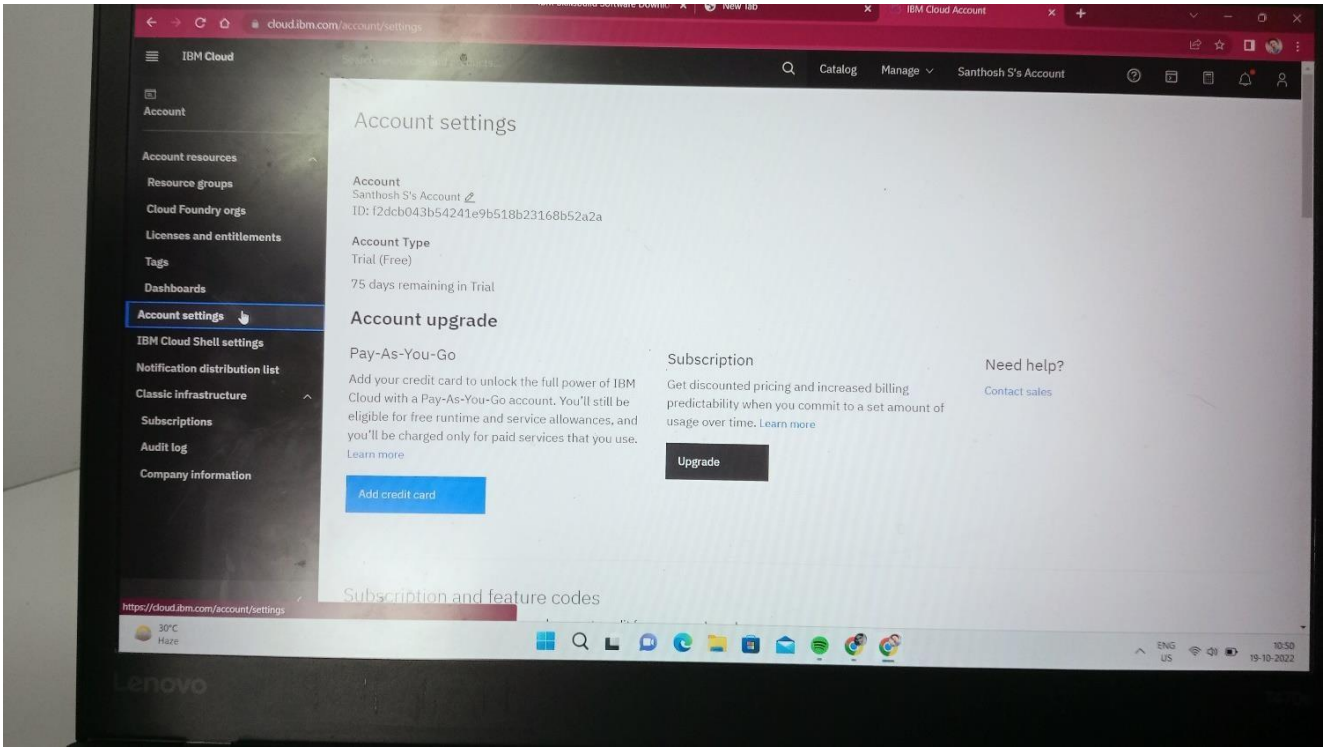
Stanly Mathew raj



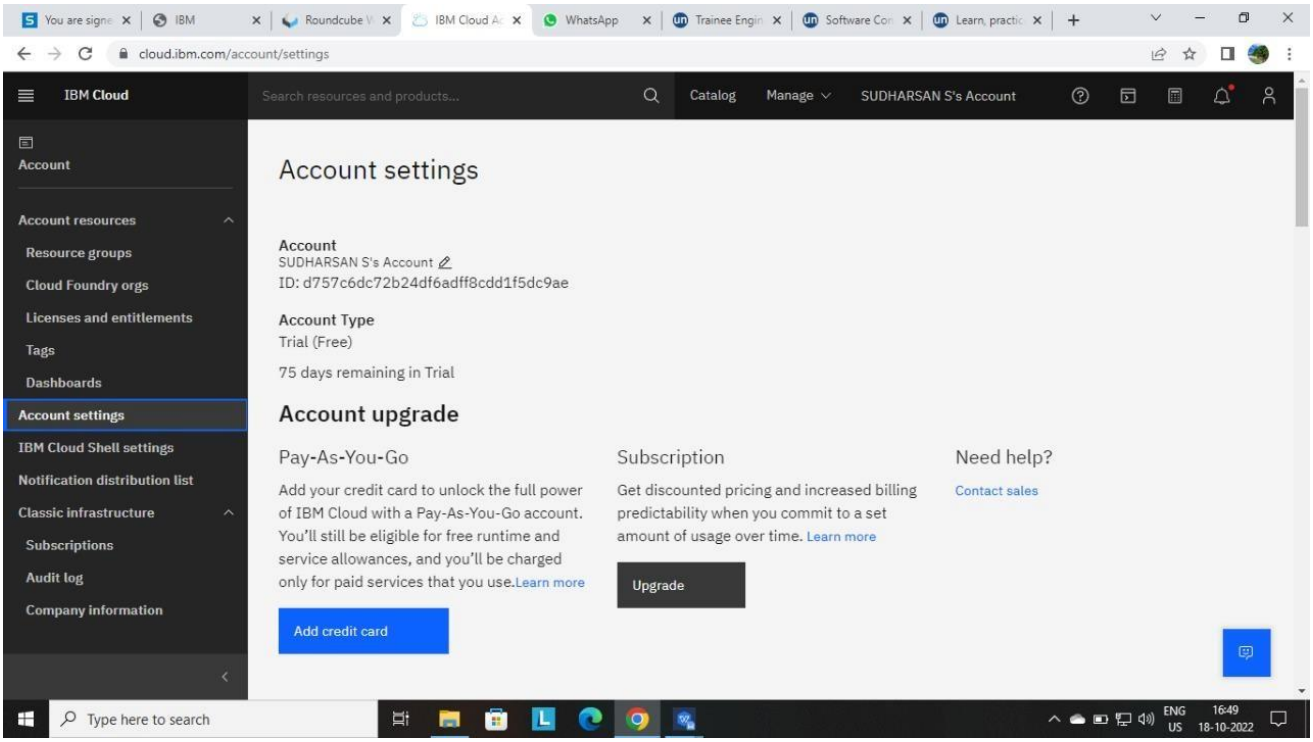
Santhosh kumar



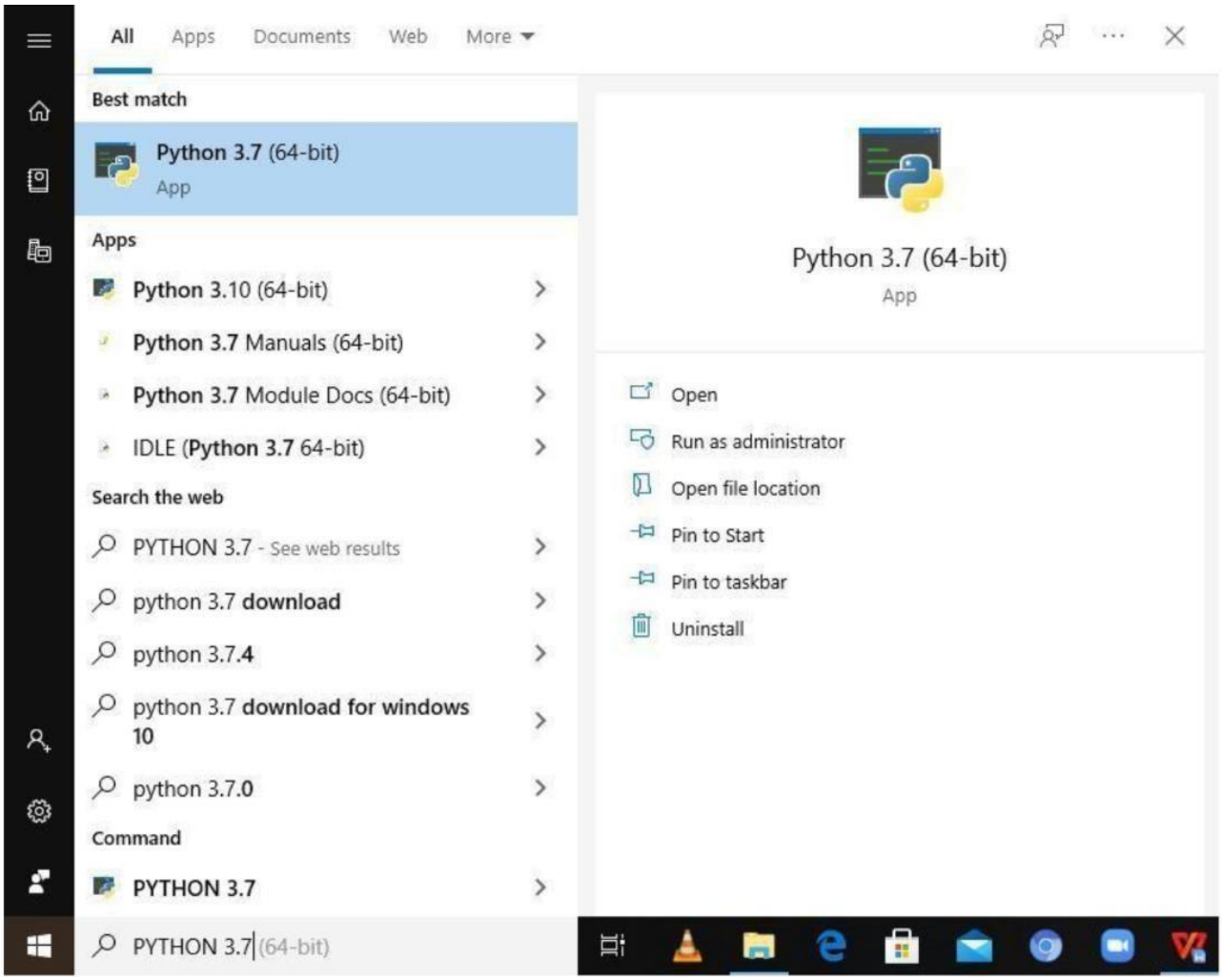
Santhosh s



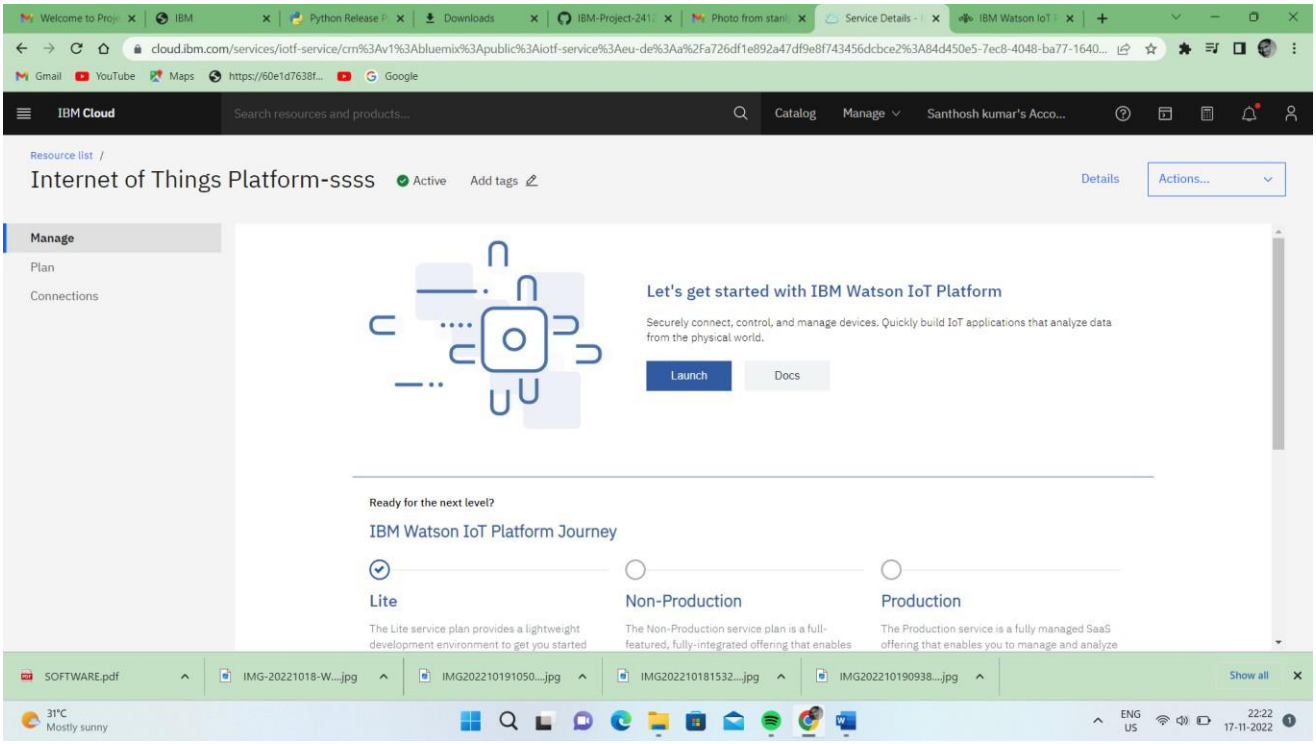
Sudharsan

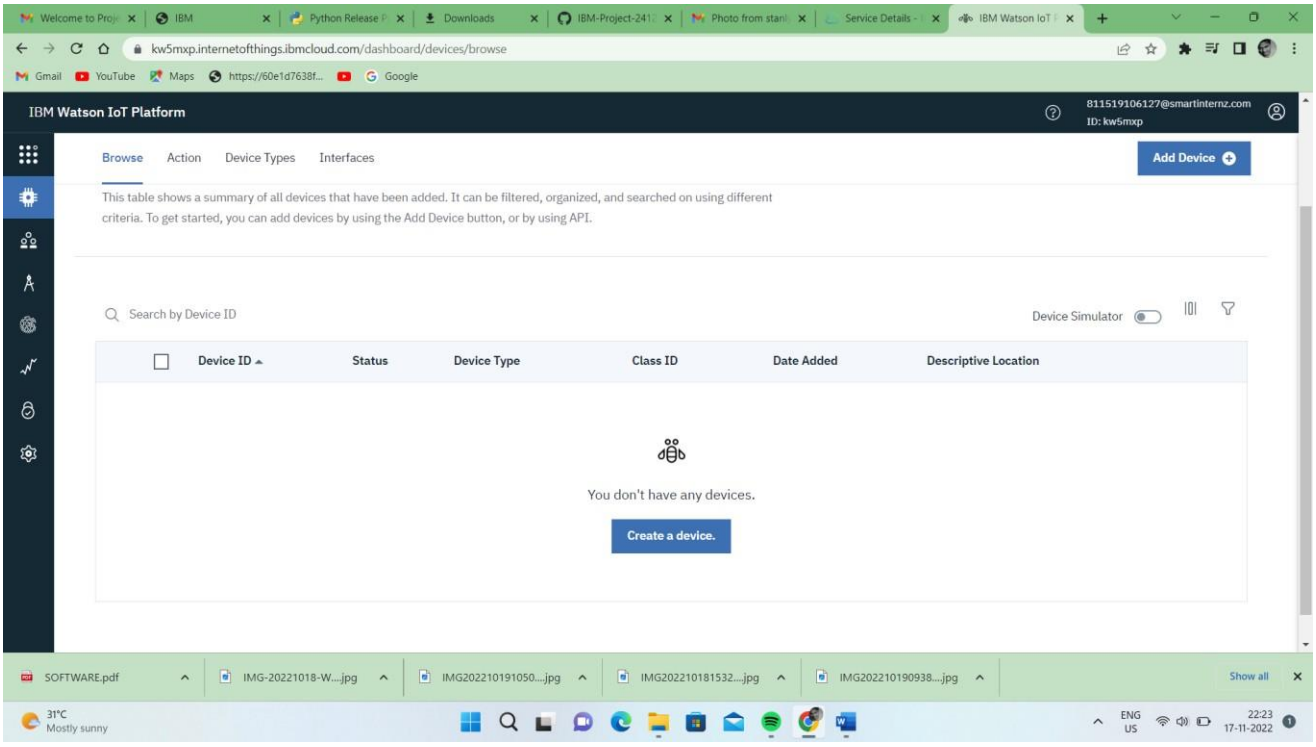


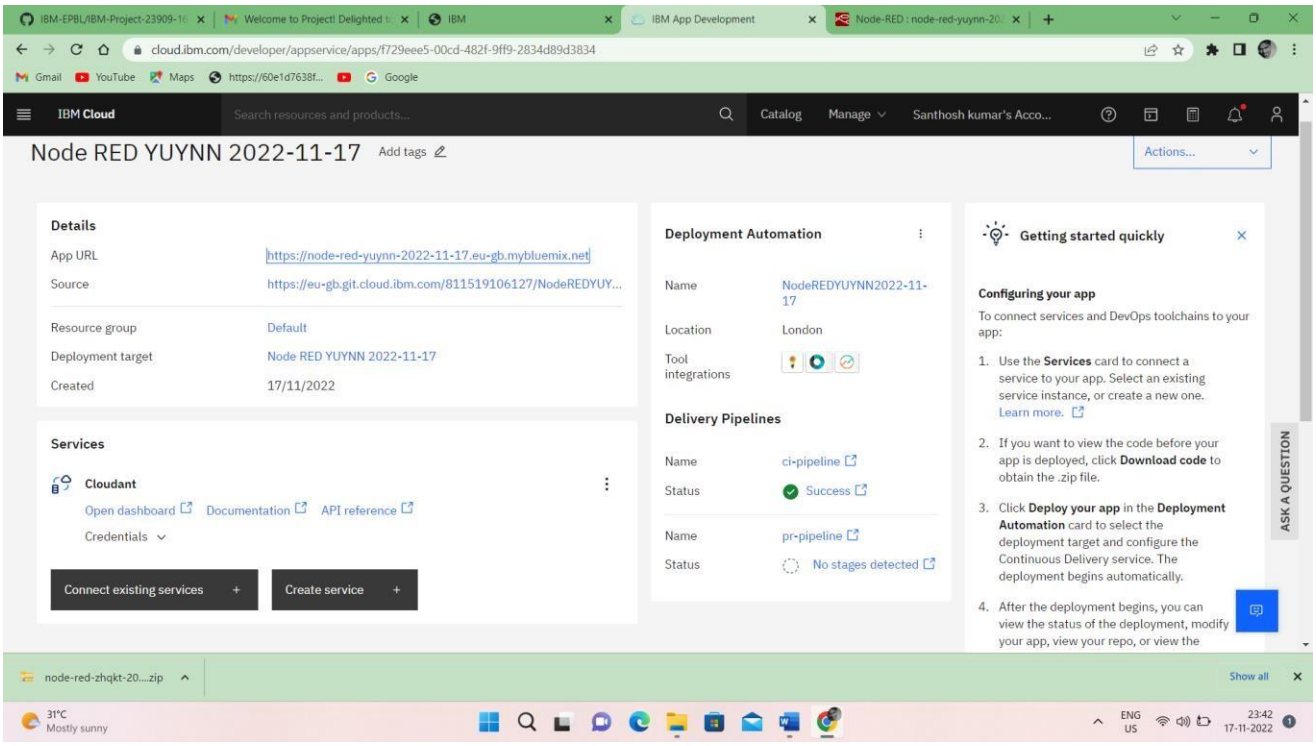
PYTHON SOFTWARE

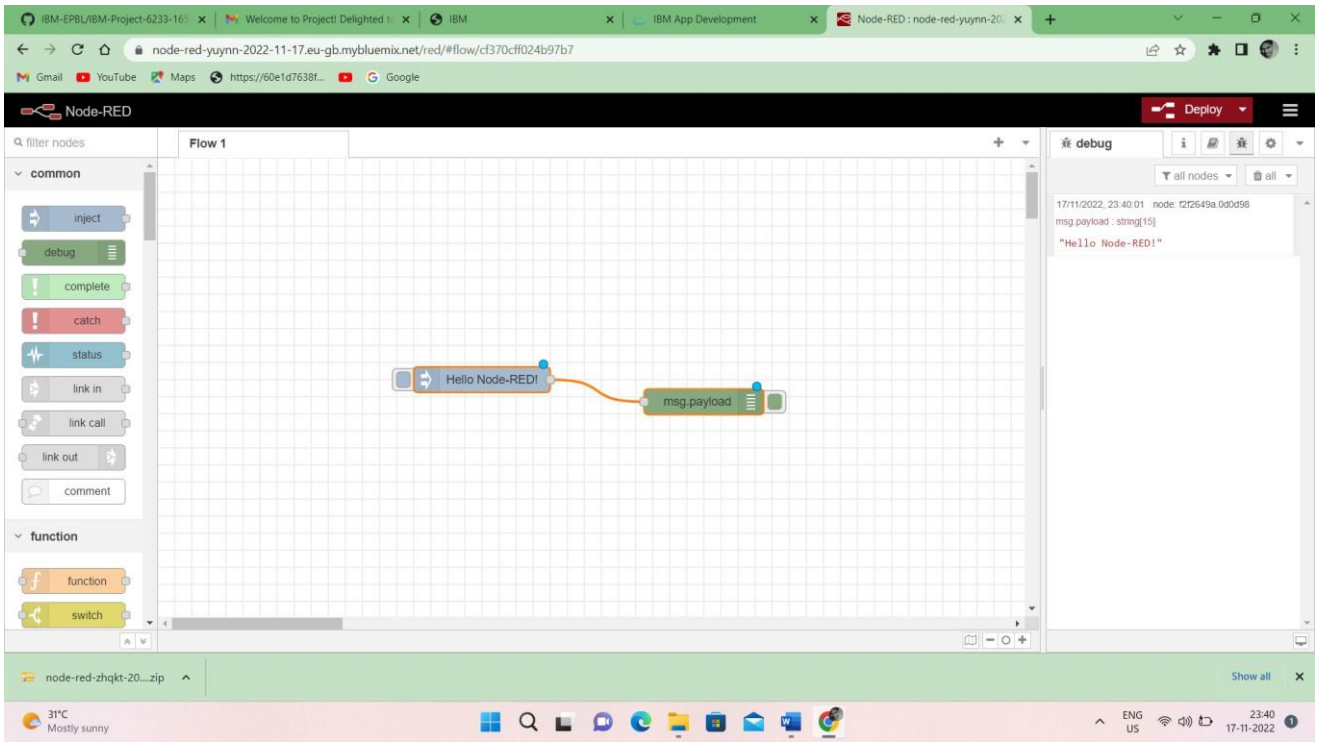


IBM WATSON SERVICE






cloud.ibm.com/developer/appservice/apps/f729eee5-00cd-482f-9ff9-2834d89d3834
IBM Cloud
Search resources and products...
Catalog
Manage
Santhosh kumar's Acco...
Node RED YUYNN 2022-11-17
Add tags
Actions...
Details
App URL
https://node-red-yuynn-2022-11-17.eu-gb.mybluemix.net
Source
https://eu-gb.git.cloud.ibm.com/811519106127/NodeREDYUY...
Resource group
Default
Deployment target
Node RED YUYNN 2022-11-17
Created
17/11/2022
Services
Cloudant
Open dashboard
Documentation
API reference
Credentials
Connect existing services
Create service
Deployment Automation
Name
NodeREDYUYNN2022-11-17
Location
London
Tool integrations
Delivery Pipelines
Name
ci-pipeline
Status
Success
Name
pr-pipeline
Status
No stages detected
Getting started quickly
Configuring your app
To connect services and DevOps toolchains to your app:
1. Use the Services card to connect a service to your app. Select an existing service instance, or create a new one. Learn more.
2. If you want to view the code before your app is deployed, click Download code to obtain the .zip file.
3. Click Deploy your app in the Deployment Automation card to select the deployment target and configure the Continuous Delivery service. The deployment begins automatically.
4. After the deployment begins, you can view the status of the deployment, modify your app, view your repo, or view the
ASK A QUESTION
node-red-zhqkt-20....zip
Show all
31°C
Mostly sunny
ENG
US
23:42
17-11-2022

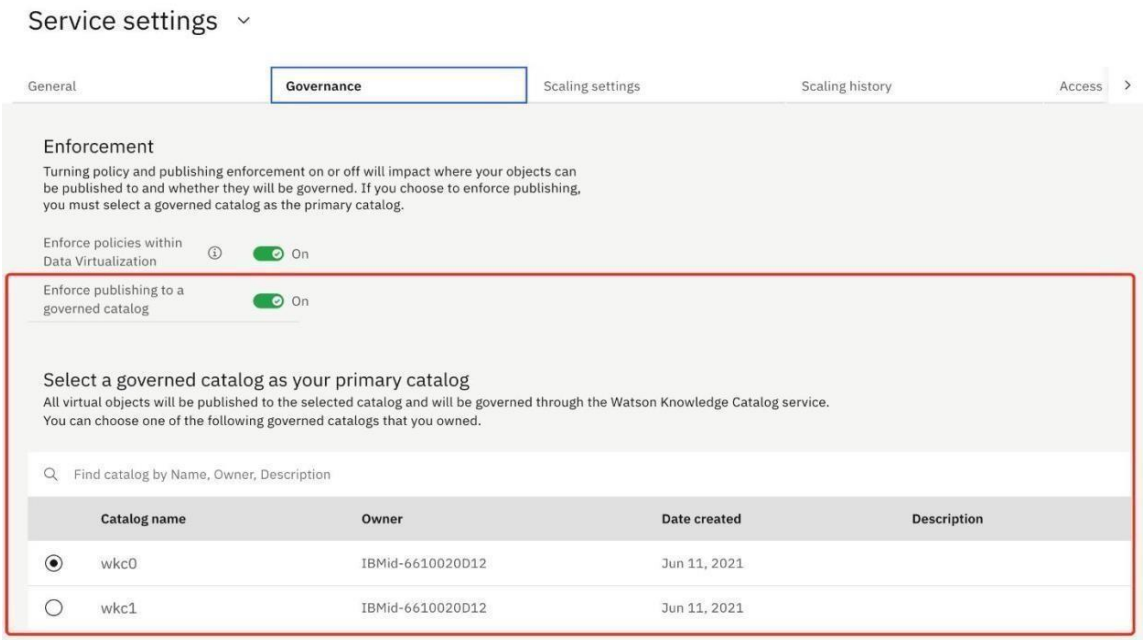

node-red-yuynn-2022-11-17.eu-gb.mybluemix.net/red/#flow/cf370cff024b97b7
Node-RED
Deploy
filter nodes
Flow 1
common
inject
debug
complete
catch
status
link in
link call
link out
comment
function
function
switch
Hello Node-RED!
msg.payload
debug
all nodes
all
17/11/2022, 23:40:01 node: f2f2649a.0d0d98
msg.payload : string[15]
"Hello Node-RED!"
node-red-zhqkt-20....zip
Show all
31°C
Mostly sunny
ENG
US
23:40
17-11-2022

PUBLISH DATA TO THE IBM CLOUD

You can publish your virtual data to catalogs in Watson™ Knowledge Catalog. An administrator can configure Data Virtualization to automatically publish all virtual objects that are created in the user interface to a configured primary catalog.

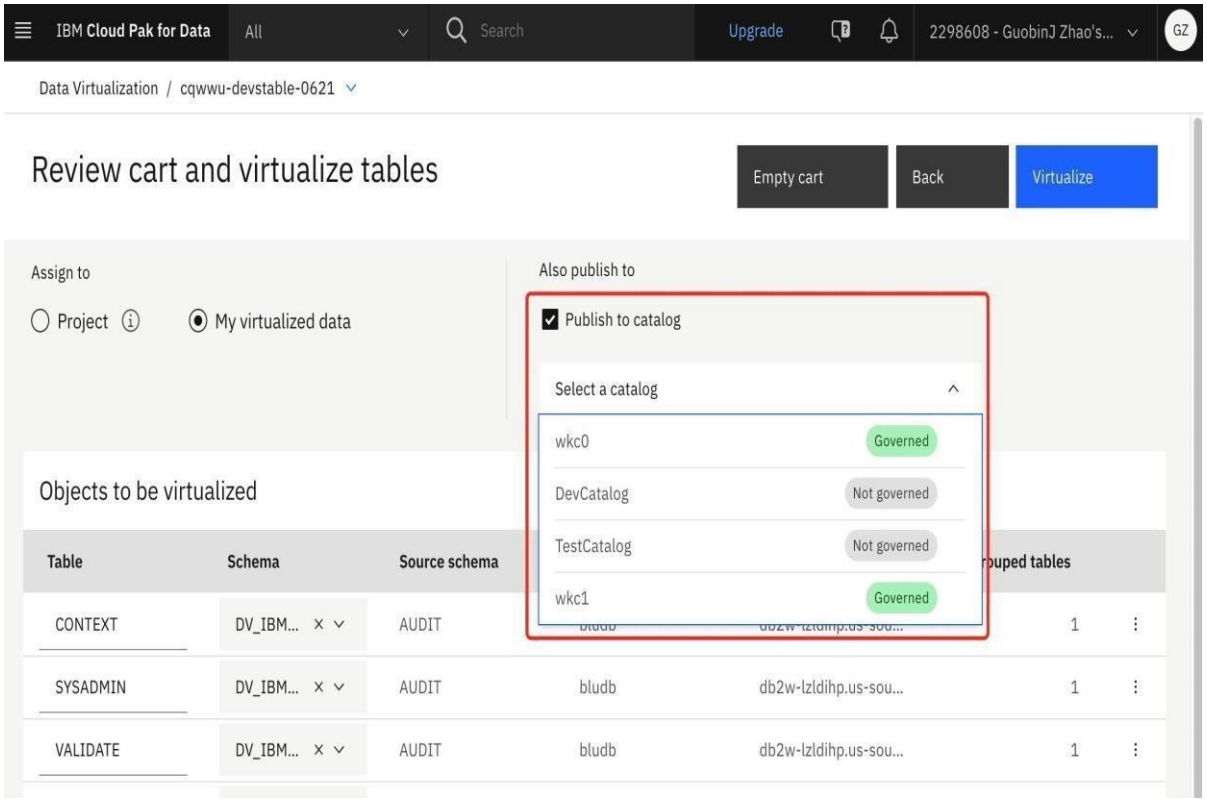
By publishing your virtual data to a catalog, you can:

*Organize, label, classify, and search for the published data assets with global search.
*Govern the virtual data asset and subject it to data protection rules byenablingpolicy enforcementin Data
*Navigate to **Service settings**.
*On the **Governance** tab, enable the **Enforce publishing to a governed catalog** option.
A list of governed catalogs that you have Admin access to is shown. You must select a governed catalog as your primary catalog. When you set a primary catalog, all



virtualized objects are published to this catalog automatically. The **Publish to catalog** option is disabled when you review your cart and virtualize tables.

1. On the service menu, click **Virtualization >Virtualized data**.
2. Select the virtual objects that you want to publish and click **Publish to catalog**.



3. Continue with the virtualization process

Python code

File Edit Format Run Options Window Help

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


# Provide your IBM Watson Device Credentials

organization = "sfptfb"  # repalce it with organization ID
deviceType = "NodeMCU"  # replace it with device type
deviceId = "19141"  # repalce with device id
authMethod = "use-token-auth"
authToken = "1914137383010209"  # repalce with token


def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status == 'lighton':
        print("LIGHT ON")
    elif status == 'lightoff':
        print("LIGHT OFF")
    else:
        print ("please send proper command")


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()

deviceCli.connect()

while True:
    pH = random.randint(0,100)
```

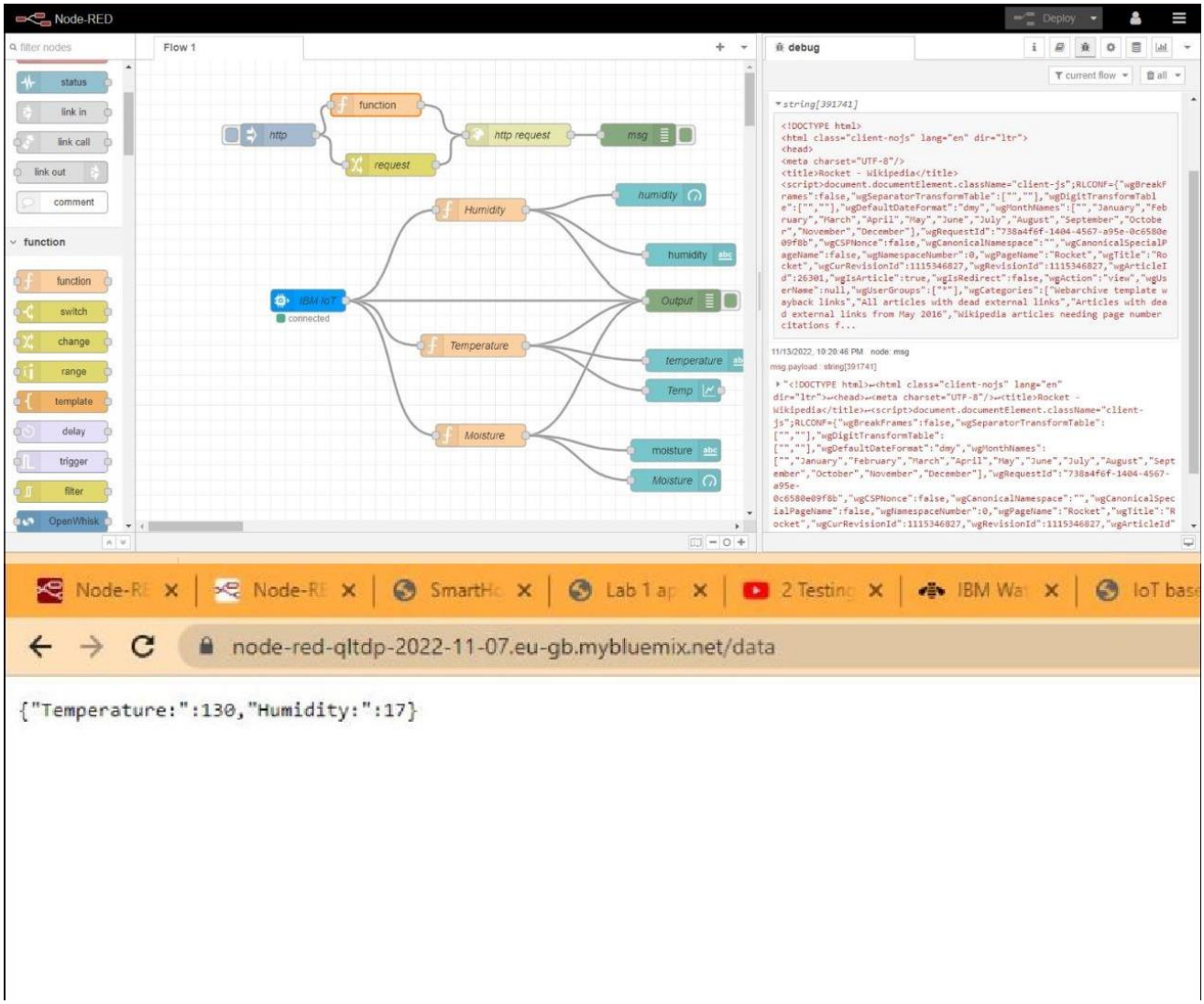
Ln: 1 Col: 0

File Edit Format Run Options Window Help

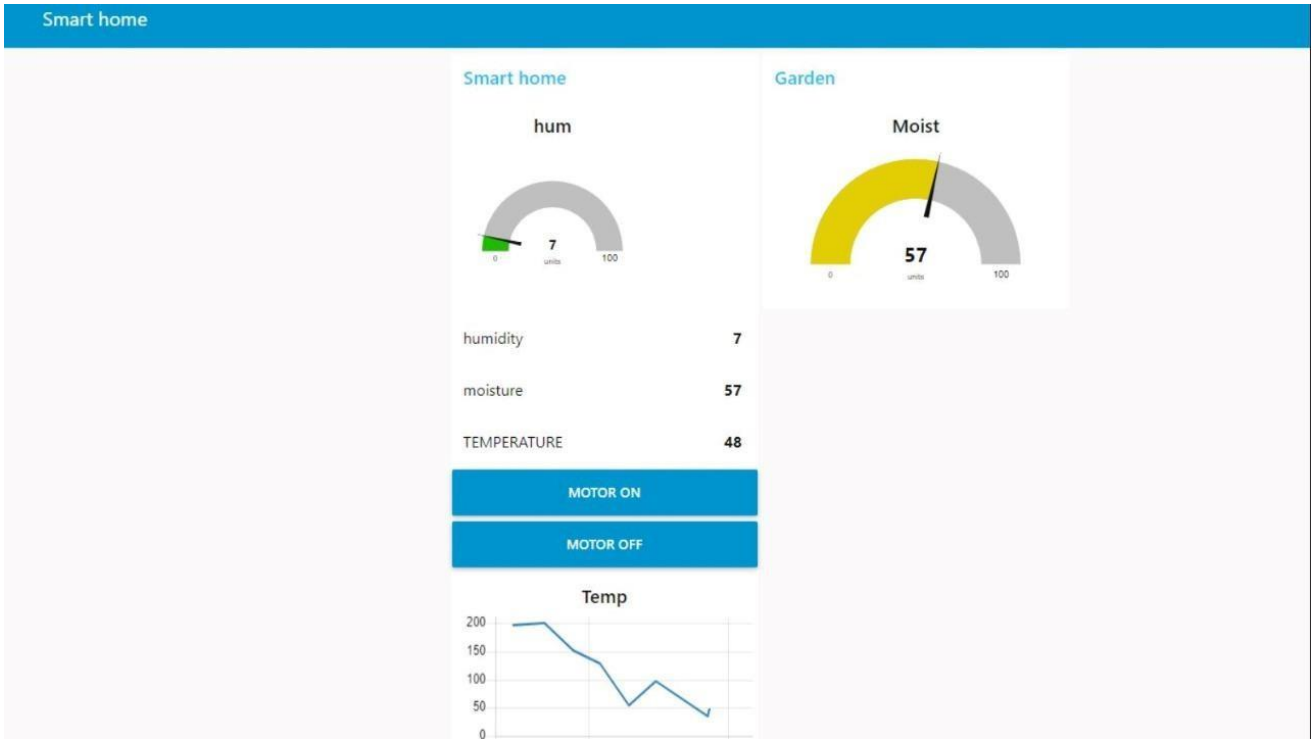
```
        print("LIGHT OFF")
    else:
        print ("please send proper command")


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()

deviceCli.connect()

while True:
    pH = random.randint(0,100)
    conductivity = random.randint(0,100)
    T = random.randint(0,100)
    oxygen = random.randint(0,100)
    turbidity = random.randint(0,100)
    # Send Temperature & Humidity to IBM Watson
    data = {'temperature': T,'ph':pH,'conductivity':conductivity,'oxygen':oxygen,"turbidity":turbidity}


    # print data
    def myOnPublishCallback():
        print("Published data",data, "to IBM Watson")


    success = deviceCli.publishEvent("event", "json", data, 0, myOnPublishCallback)
    if not success:
        print("Not connected to IoTF")
    time.sleep(5)

    deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud
```

Ln: 1 Col: 0

DEVELOP THE WEB APPLICATION




Node-RED
Deploy
filter nodes
Flow 1
dashboard
Layout
Site
Theme
Tabs & Links
Smart Home
Hall AC
Garden
dropdown
switch
slider
numeric
text input
date picker
colour picker
form
text
gauge
chart
audio out
notification
ui control
template
Humidity
Hum
IBM IoT
connected
Output
Temperature
Temp
Moisture
Moisture
Smart Home
Hall AC
Humidity
100
80
60
40
20
0
20:19:49
20:20:19
20:20:49
20:21:20
Temp
100
80
60
40
20
0
20:19:49
20:20:19
20:20:49
20:21:20
Garden
Moist
73
0
units
10

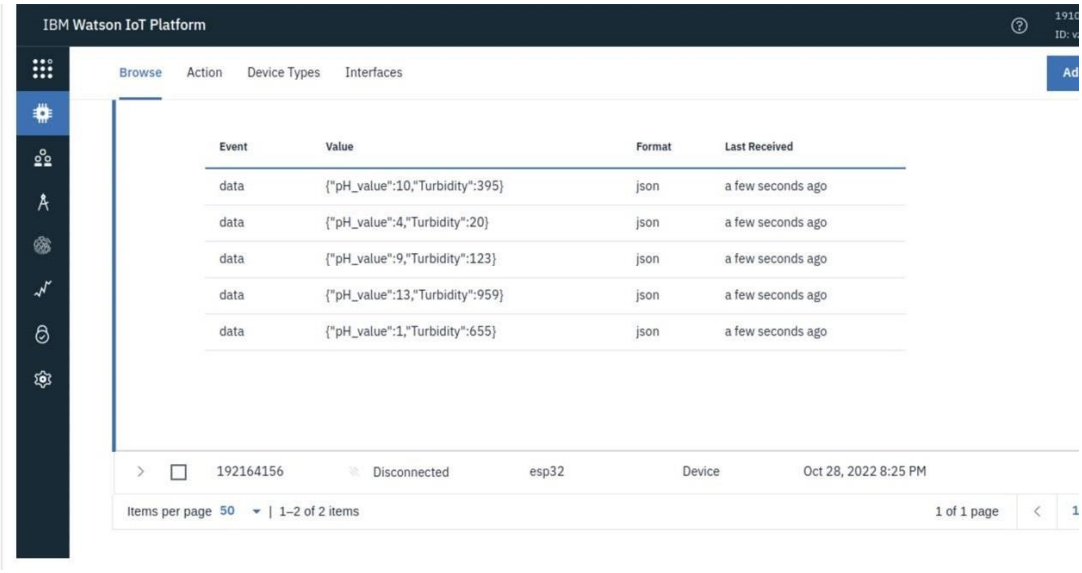

Smart home
Smart home
hum
7
0
100
humidity
7
moisture
57
TEMPERATURE
48
MOTOR ON
MOTOR OFF
Temp
200
150
100
50
0
Garden
Moist
57
0
100

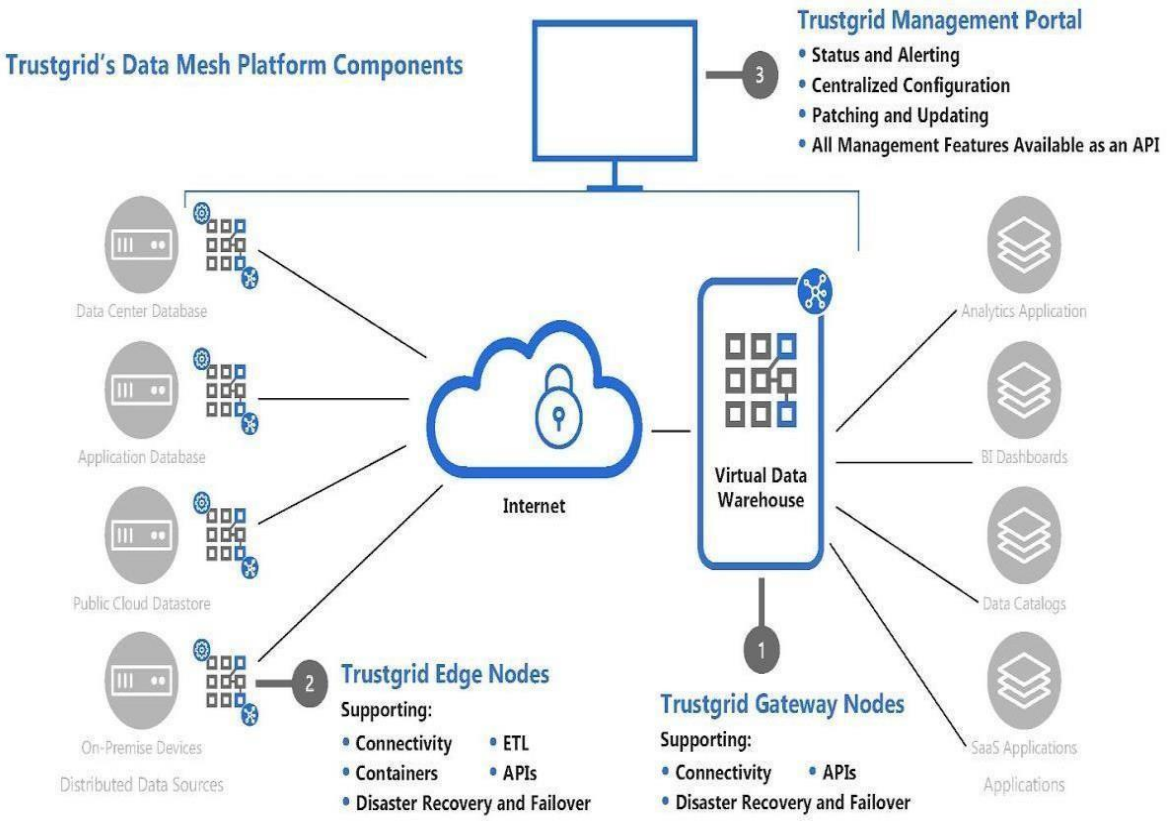
BUILD MOBILE APP

CONFIGURE THE APPLICATION TO RECEIVE THE DATA FROM CLOUD

This is created through the use of gateway nodes to create a **Virtual Data Warehouse.** This Virtual Data Warehouse allows application developers to map access to remote data points.

This software-defined gateway is run adjacent to the application it serves and can be deployed within a cloud environment or in a data center.





This Virtual Data Warehouse allows for the virtual aggregation of data so that an application (or many applications) can easily consume it. Once a data source is added to the Virtual Data Warehouse an application has secure, real-time, persistent access to that data set.

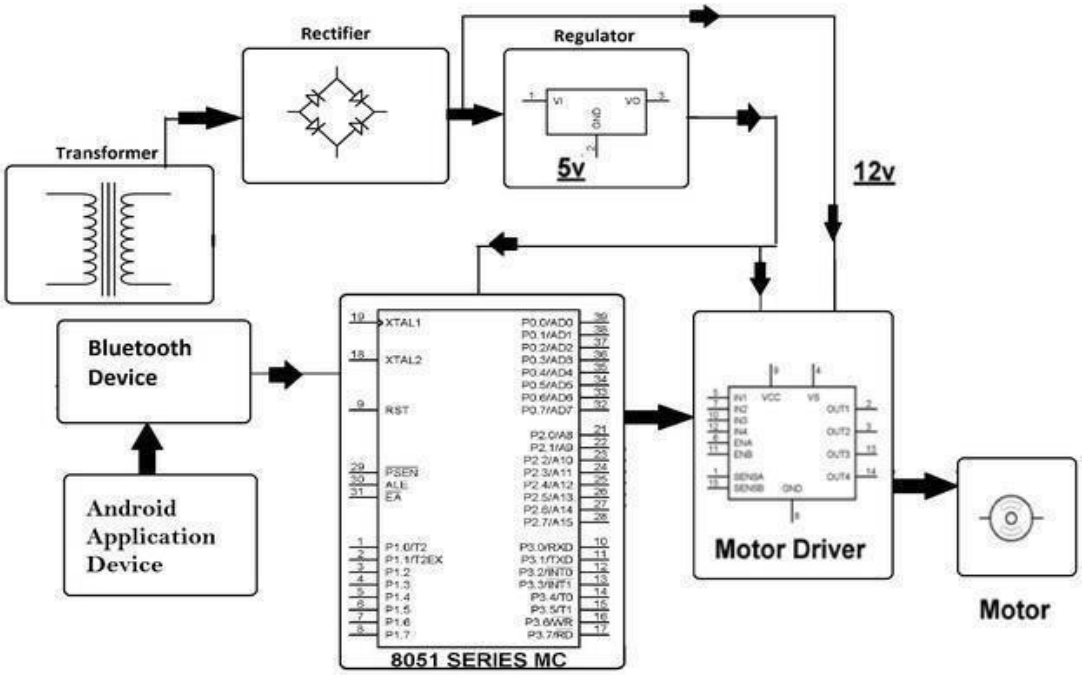
Hardware device – The hardware device is one of the easiest methods of deployment because Trust grid handles all of the software imaging, logistics and deployment support for the end-user. A hardware appliance is ideal for environments with limited onsite support

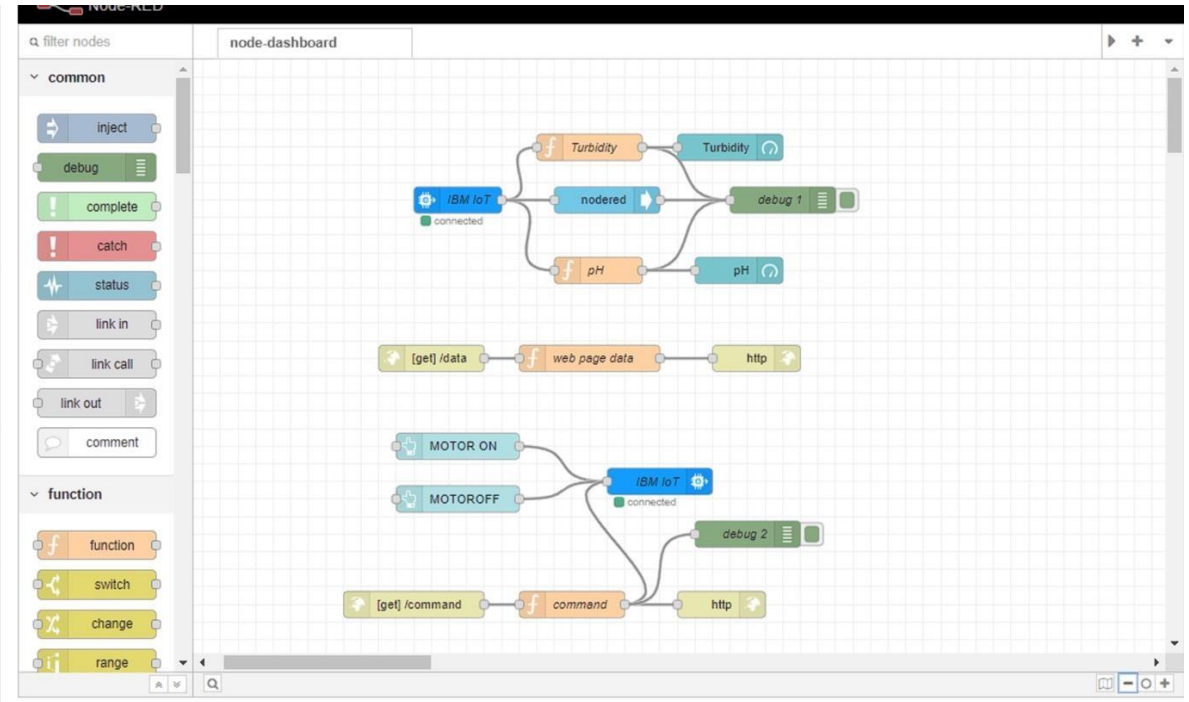
BUILD MOBILE APP

CONFIGURE THE MOBILE APP FOR CONTROLLING MOTOR

USING BUTTONS

This system DC motor Controller by Android is developed to control the speed of the DC motor in both clockwise and anticlockwise direction. For this DC motor is interfaced to the 8051 micro controller. A Bluetooth modem is used to receive direction commands and PWM commands. When an Android device sends commands, it is received by the Bluetooth modem which then sends the commands to the microcontroller. The microcontroller the controls the DC motor through motor driver. The entire system is powered by 12V transformer. LCD display is used to show the status and the speed of the DC motor. The android application is used to control the entire system. The start button is first clicked to start the motor and then the motor can run in both clockwise and anticlockwise direction. Simultaneously the status of the system is displayed on the LCD screen and also the speed of the DC motor is displayed on the screen. Thus the speed of the motor can be increased or decreased in clockwise or anticlockwise direction with the help of this android application.

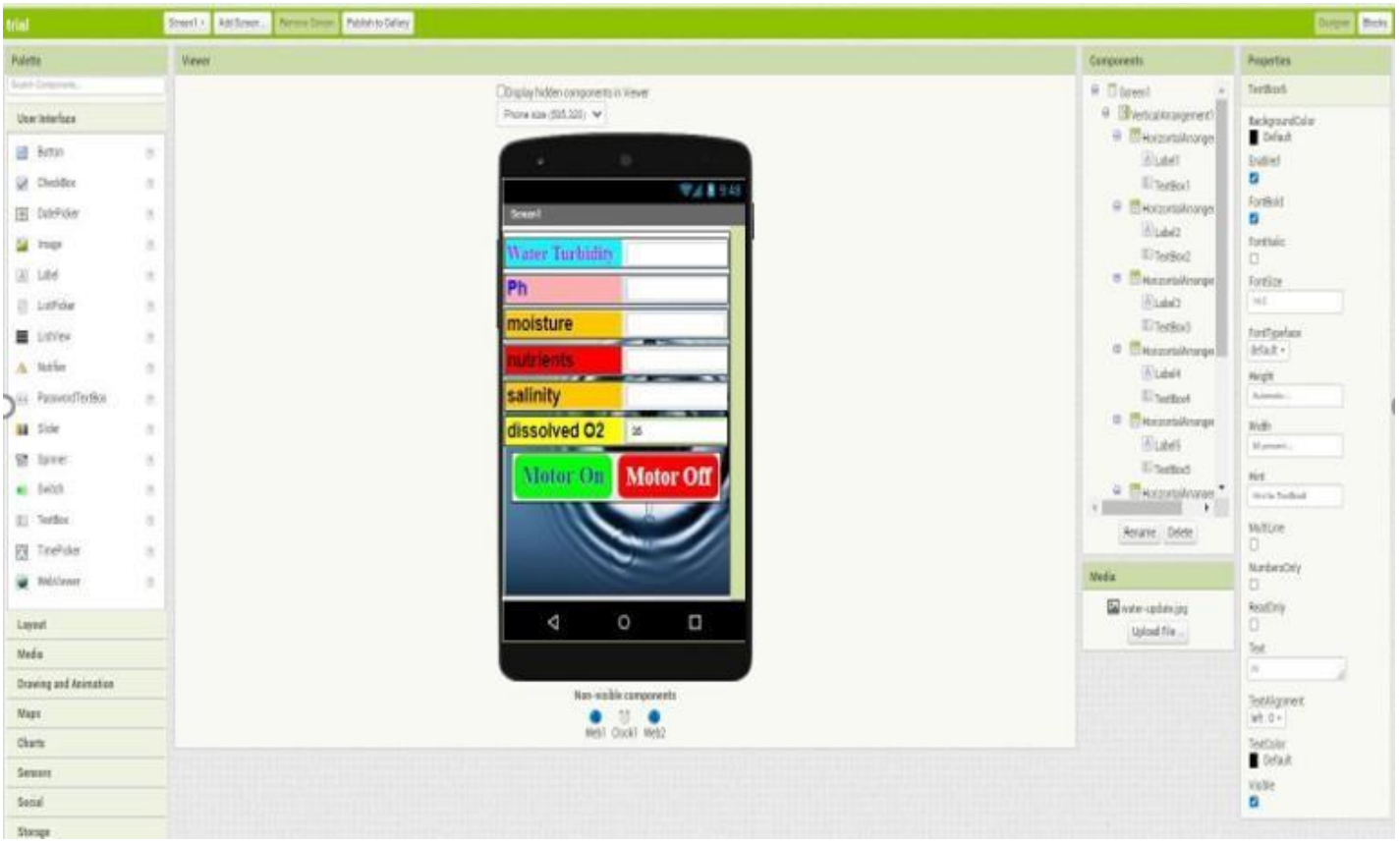



filter nodes
node-dashboard
common
inject
debug
complete
catch
status
link in
link call
link out
comment
function
function
switch
change
range
Turbidity
Turbidity
IBM IoT
connected
nodered
debug 1
pH
pH
[get] /data
web page data
http
MOTOR ON
MOTOROFF
IBM IoT
connected
debug 2
[get] /command
command
http

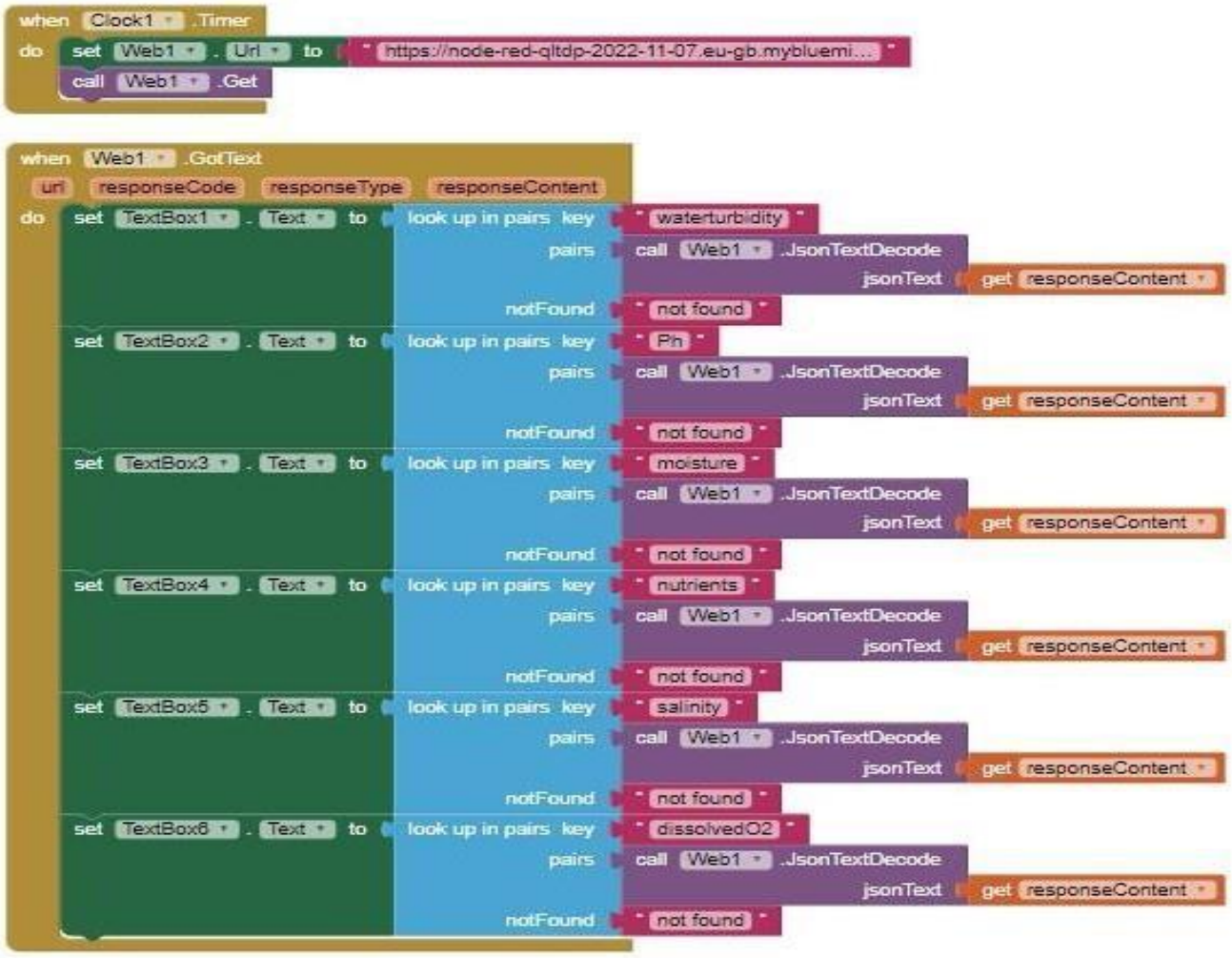
DESIGN AN APP IN MIT APP INVENTOR

PARAMETERS ARE:

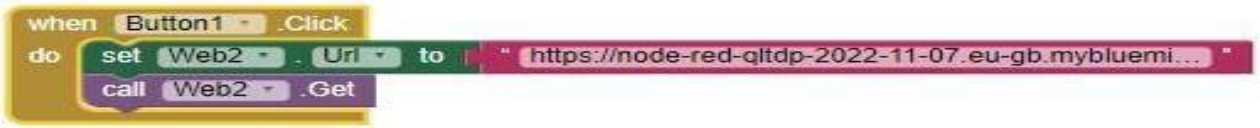
1. **Ph**
2. **Water turbidity**
3. **Moisture**
4. **Salinity**
5. **Dissolved Oxygen**
6. **Nutrients**

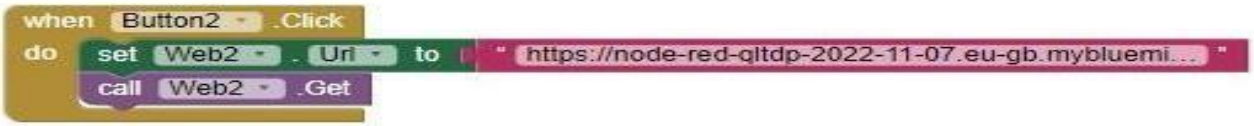


FRAME THE BLOCKS FOR FUNCTIONING



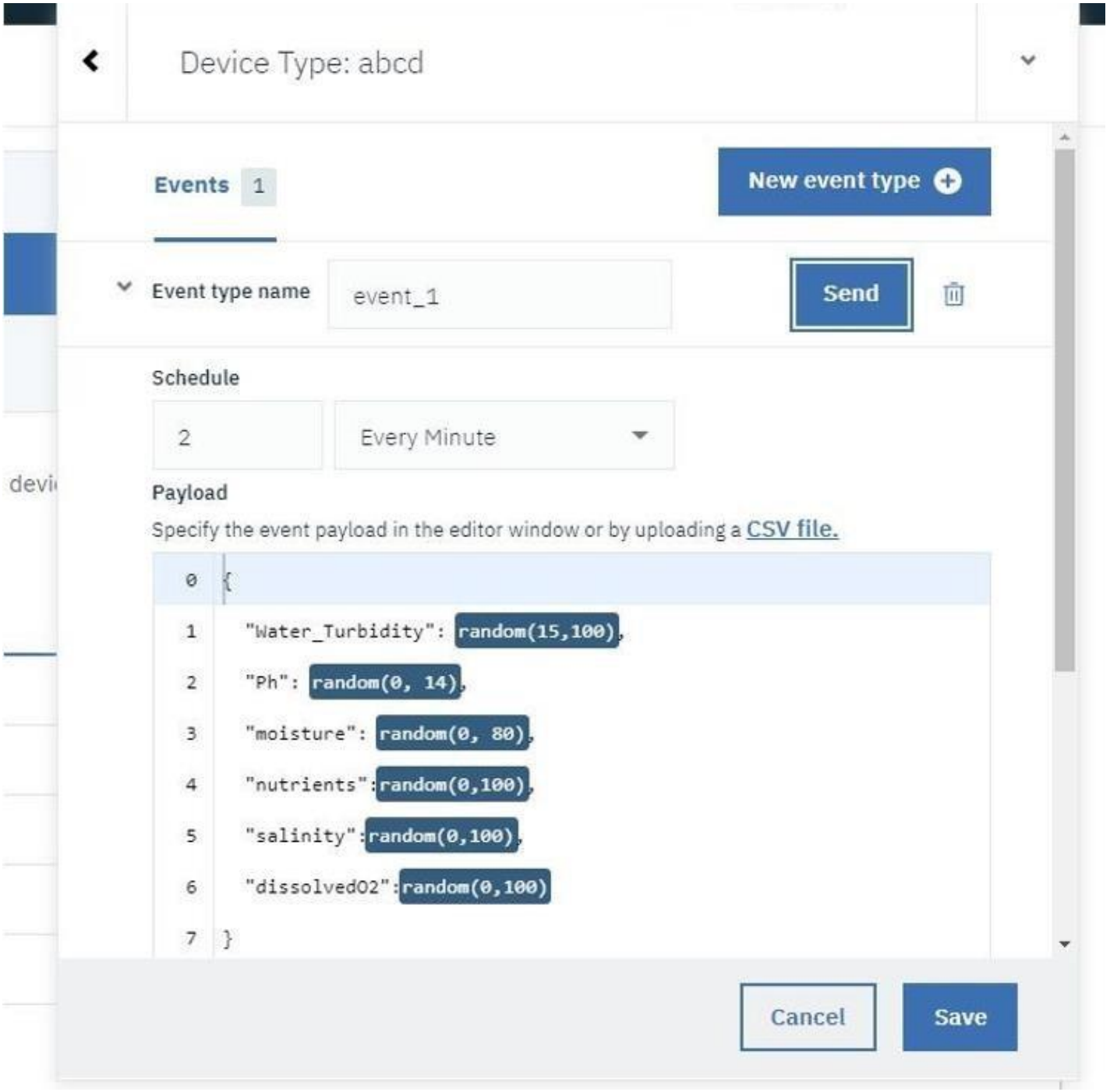
FOR BUTTONS

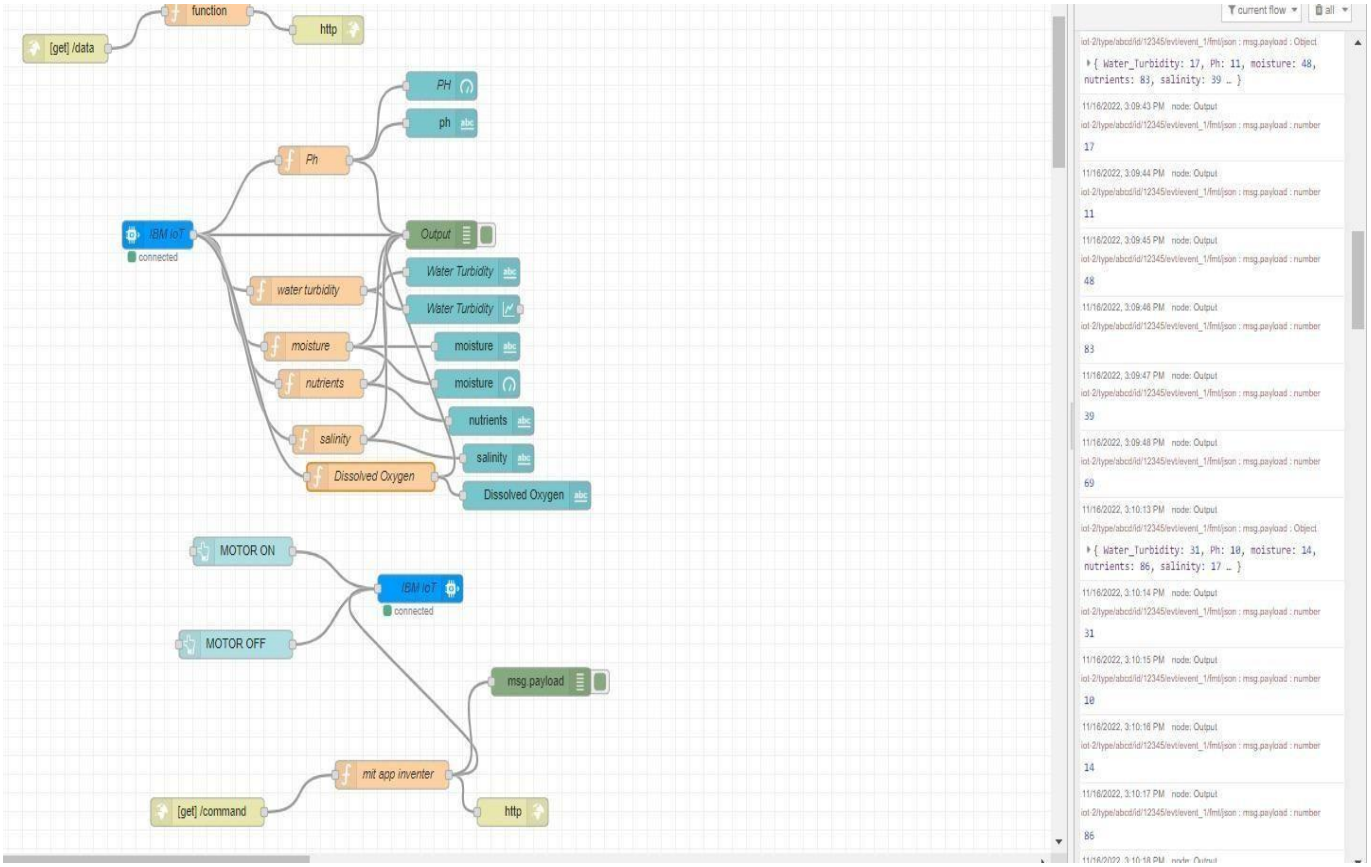




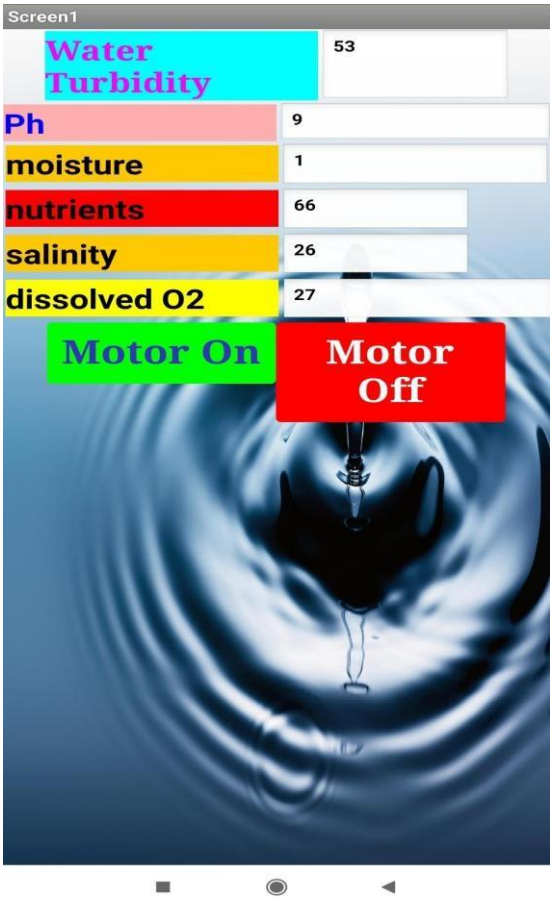
https://node-red-qltdp-2022-11-07.eu-gb.mybluemix.net/data

{"Ph":6,"waterturbidity":78,"moisture":73,"nutrients":10,"salinity":38,"dissolvedO2":50}


Device Type: abcd
Events 1
New event type
Event type name
event_1
Send
Schedule
2
Every Minute
Payload
Specify the event payload in the editor window or by uploading a CSV file.
{
"Water_Turbidity": random(15,100),
"Ph": random(0, 14),
"moisture": random(0, 80),
"nutrients":random(0,100),
"salinity":random(0,100),
"dissolved02":random(0,100)
}
Cancel
Save



APP IN MOBILE



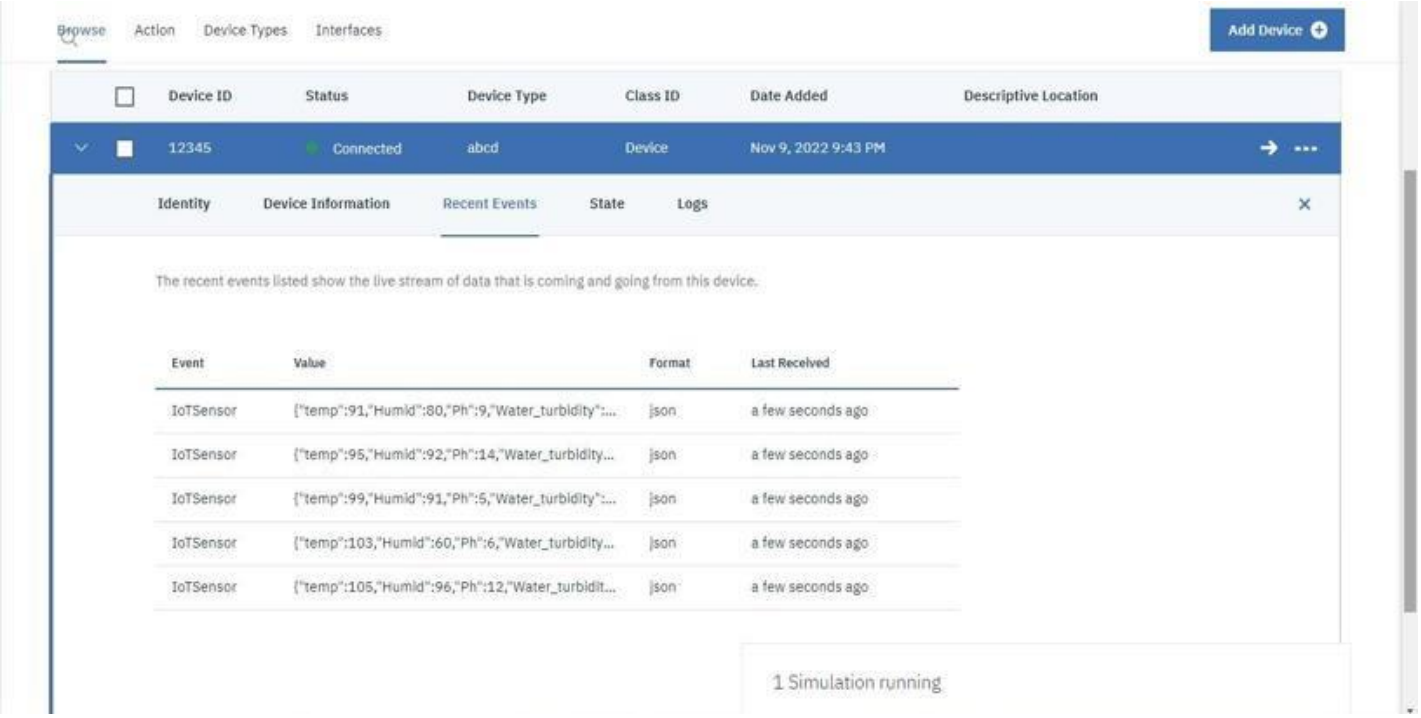
SPRINT 1

PYTHON CODE:

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device Credentials
organization = "84708c"
deviceType = "abcd"
deviceId = "12345"
authMethod = "token"
authToken = "12345678"
def myCommandCallback (cmd):
    print ("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status== "motoron":
        print ("motor is on")
    elif status == "motoroff":
        print ("motor is off")
    else:
        print ("please send proper command")
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 evention connecting device: %s" % str(e))
        sys.exit()
```

```
deviceCli.connect()
while True:
    temp=random.randint (90,110)
    Humid=random.randint (60,100)
    Ph=random.randint (0,14)
    Water_turbidity=random.randint (15,60)
    data = {'temp' : temp, 'Humid': Humid, 'Ph' : Ph, 'Water_turbidity' : Water_turbidity}
    def myonPublishCallback():
        print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid,"Ph = %s" % Ph,"Water Turbidity = %s NTU" % Water_turbidity, "to IBM Watson")
    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish = myonPublishCallback)
    if not success:
        print("Not connected to IOTF")
        time.sleep (10)
        deviceCli.commandCallback = myCommandCallback
deviceCli.disconnect()
```

OUTPUT:



SPRINT 2

AIM:

To create device in the IOT Watson Platform and Configure Node Red Services.

REQUIREMENT:

IBM cloud, IBM IOT WATSON PLATFORM,NODE RED SERVICES.

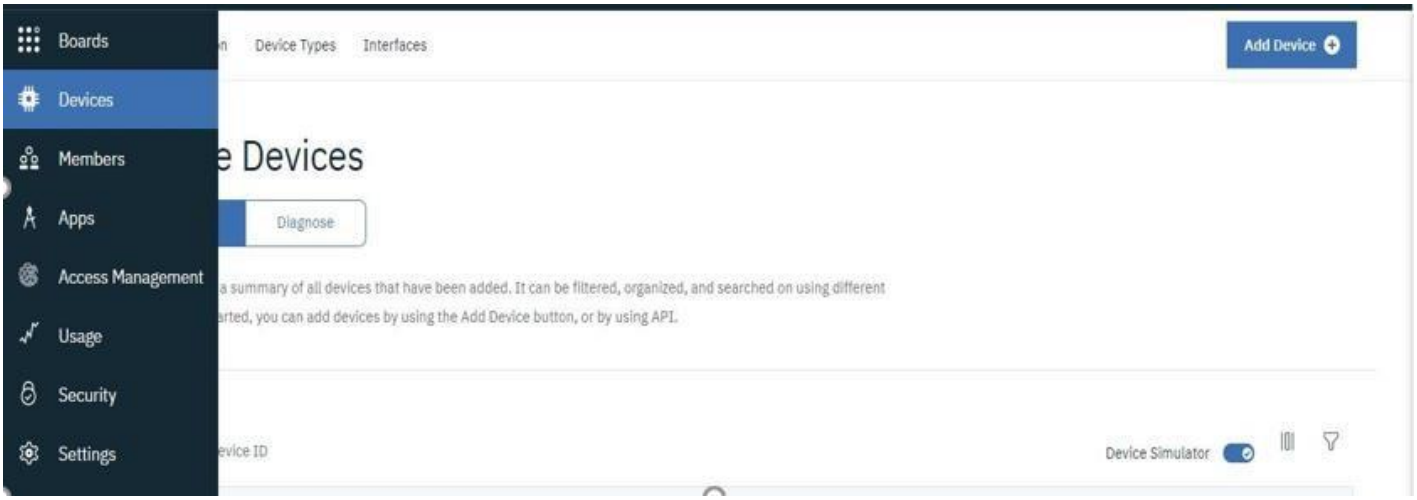
WORKFLOW:

STEP 1:

Log on to IBM cloud and create IBM Watson IOT Platform from IBM cloud Dashboard.

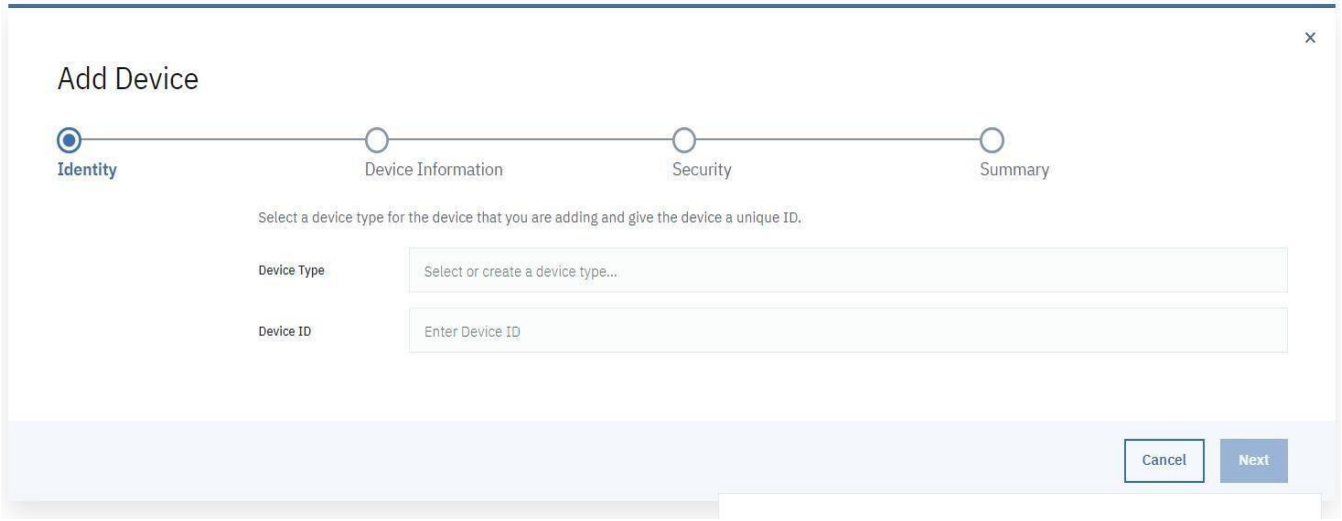
STEP 2:

After Creating IBM Watson IOT Platform,create an Organization (ex.84708c ID: 84708c Bluemix Free)



STEP 3:

Create an device IBM IOT PALTFORM.



TYPE THE REQUIRED FIELDS (TYPE: ESP32 , ID: 1234)
GIVE AUTH-TOKEN.

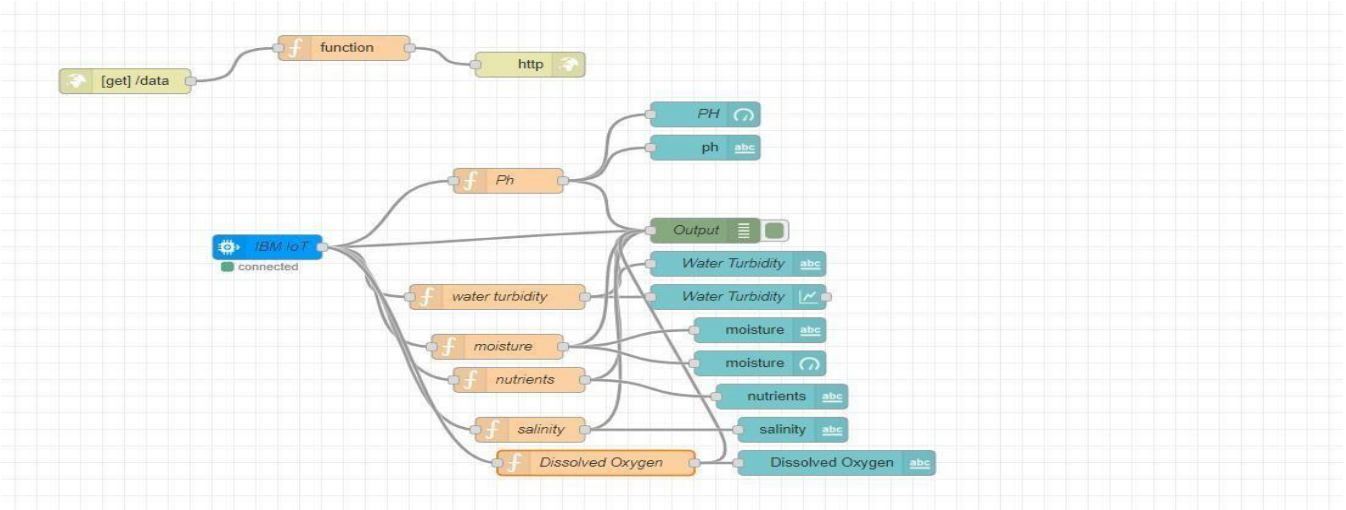
STEP 4:

NODE RED SERVICE

- INSTALL IBM IOT IN MANGE PALETTE.
- INSTALL NODE RED DASHBOARD.

STEP 5:

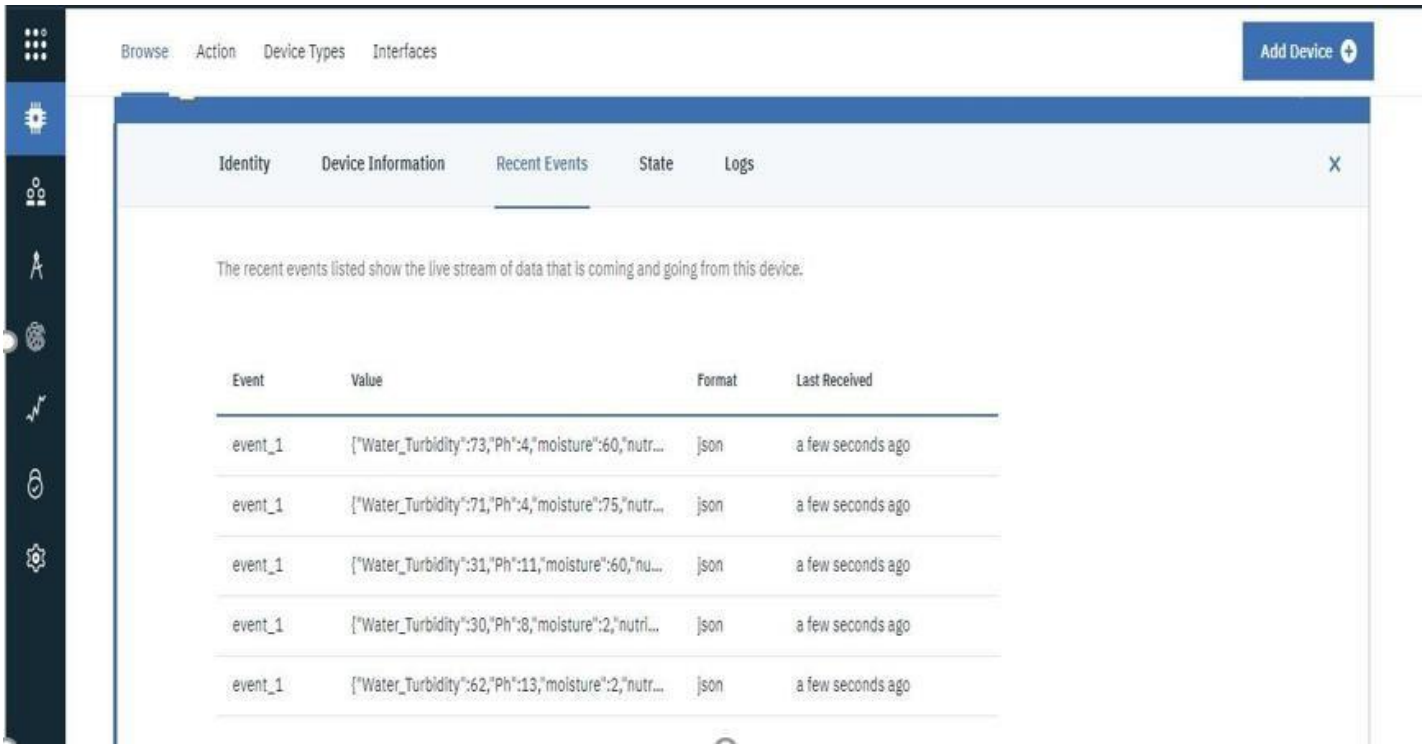
Configuring the corresponding nodes

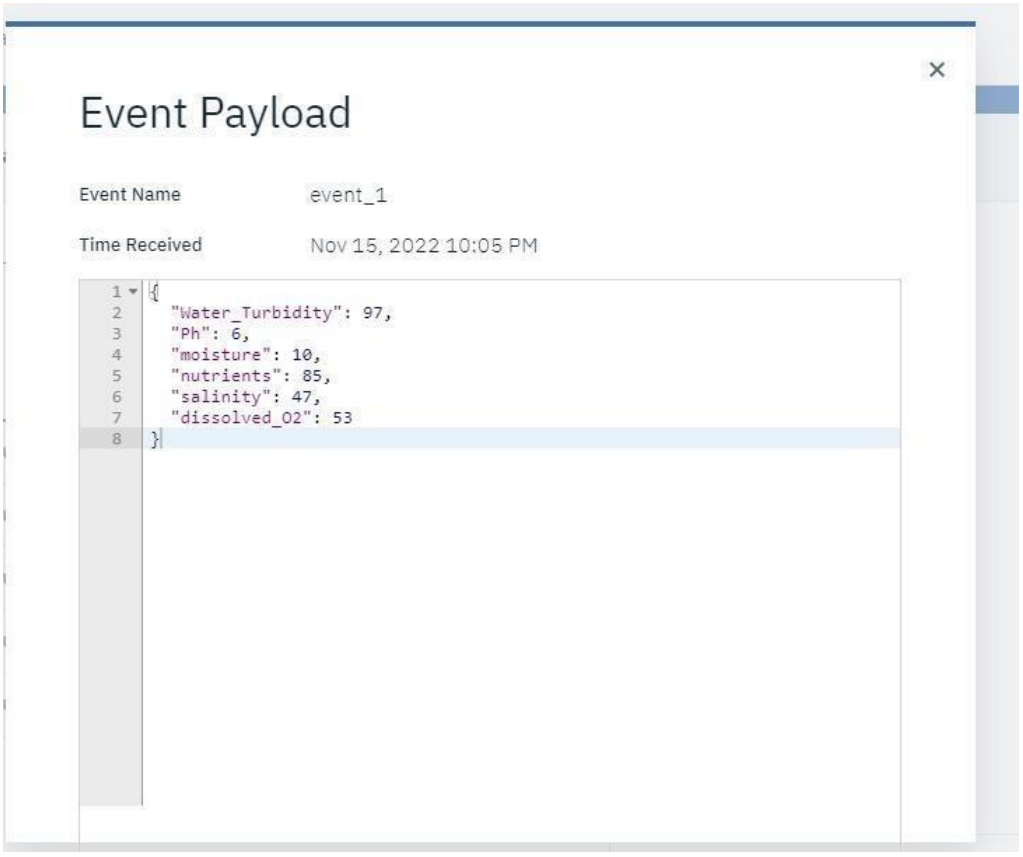


STEP 6:

Deploy the Services and verify the output values.

OUTPUT IN IBM WATSON IOT PLATFORM:



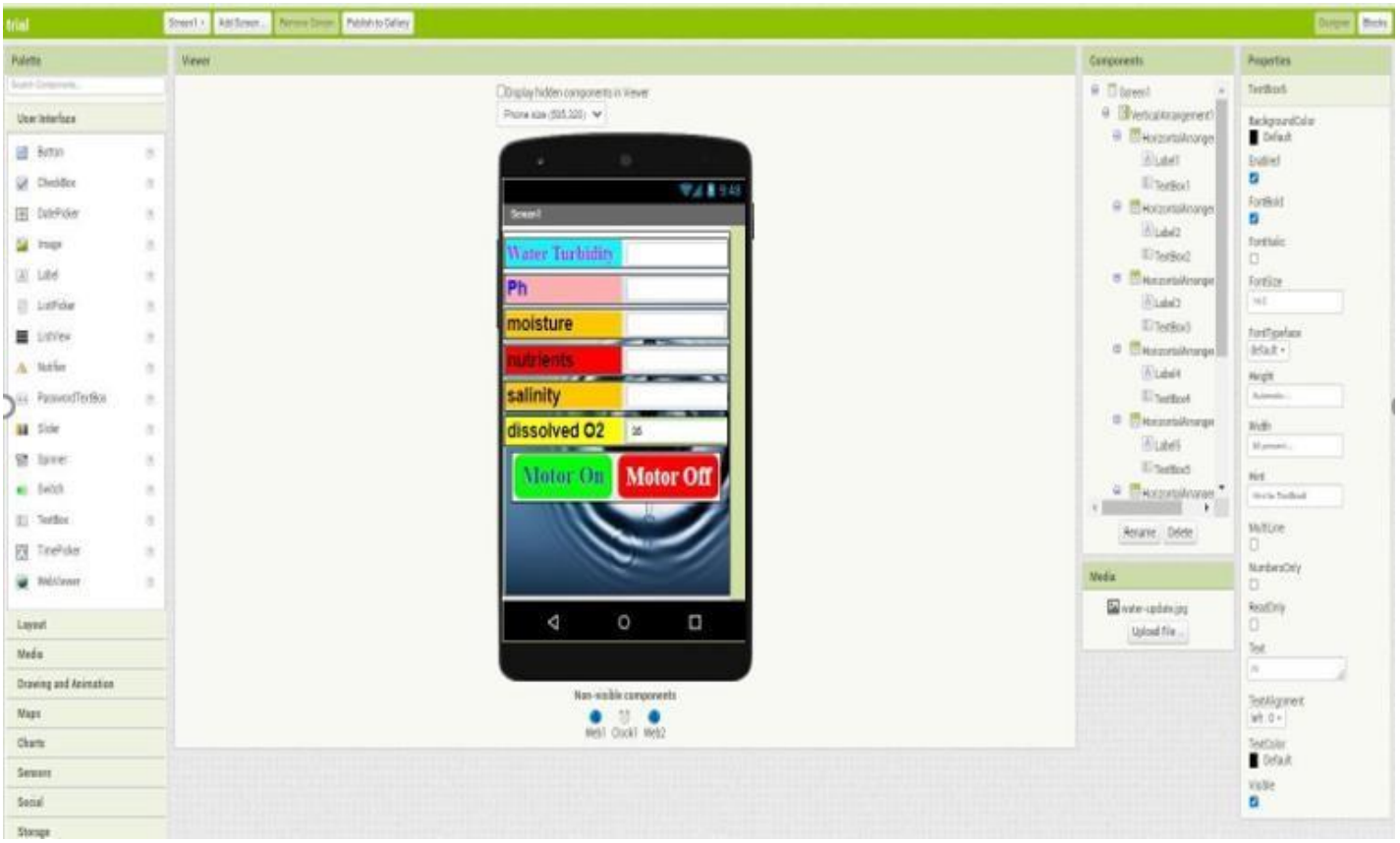


SPRINT 3

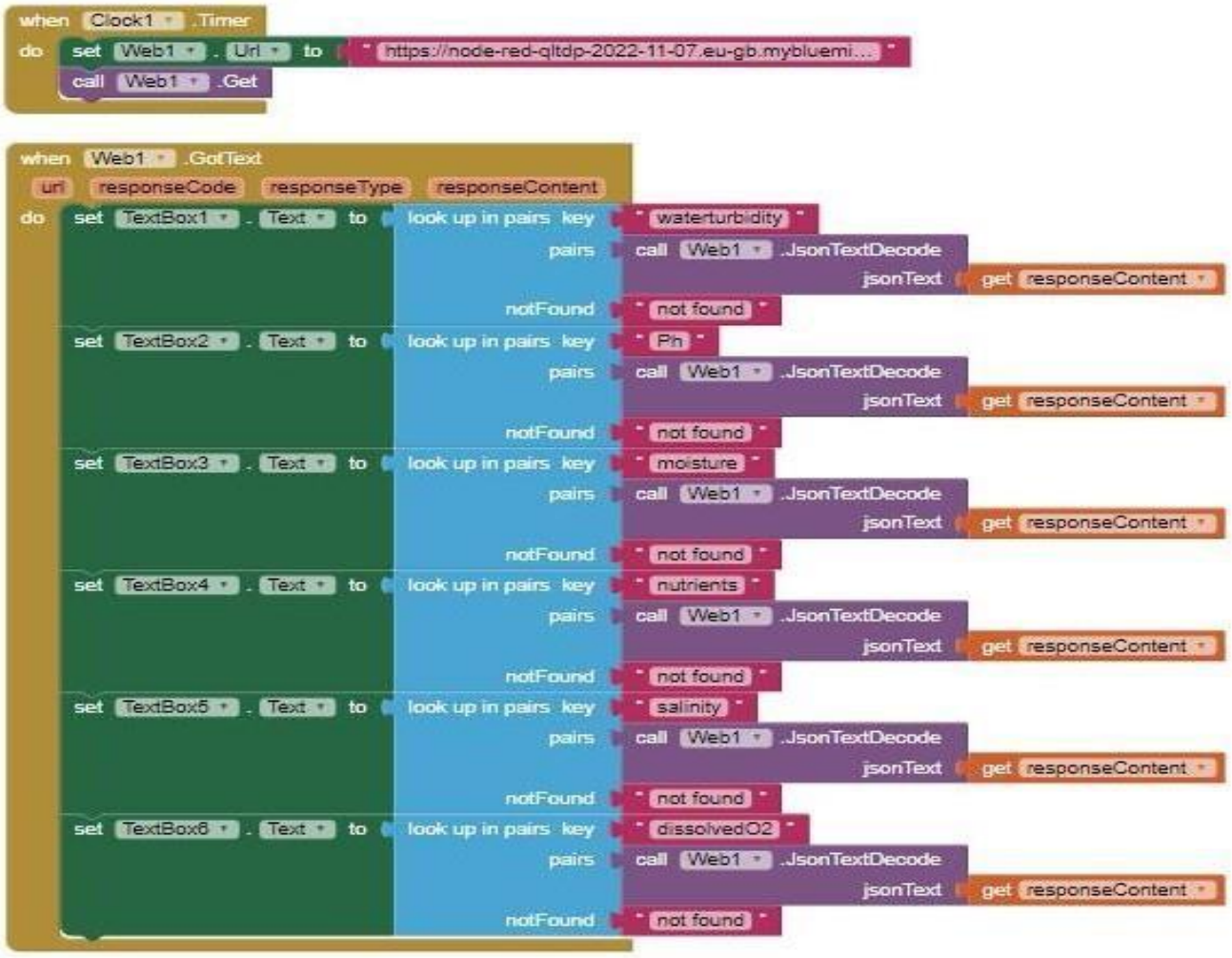
DESIGN AN APP IN MIT APP INVENTOR

PARAMETERS ARE:

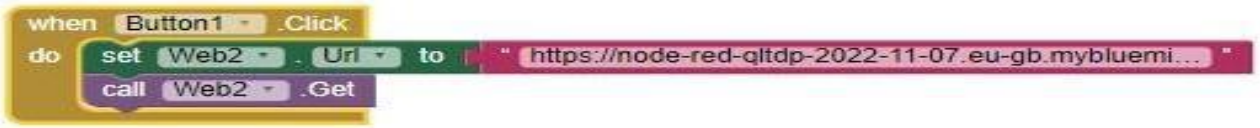
1. **Ph**
2. **Water turbidity**
3. **Moisture**
4. **Salinity**
5. **Dissolved Oxygen**
6. **Nutrients**

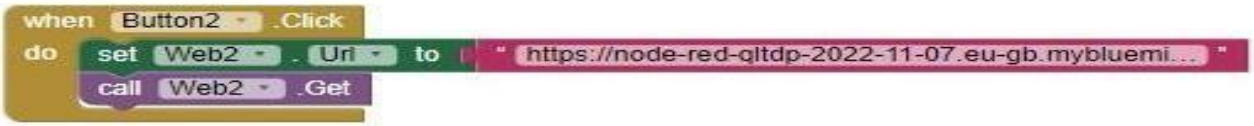


FRAME THE BLOCKS FOR FUNCTIONING



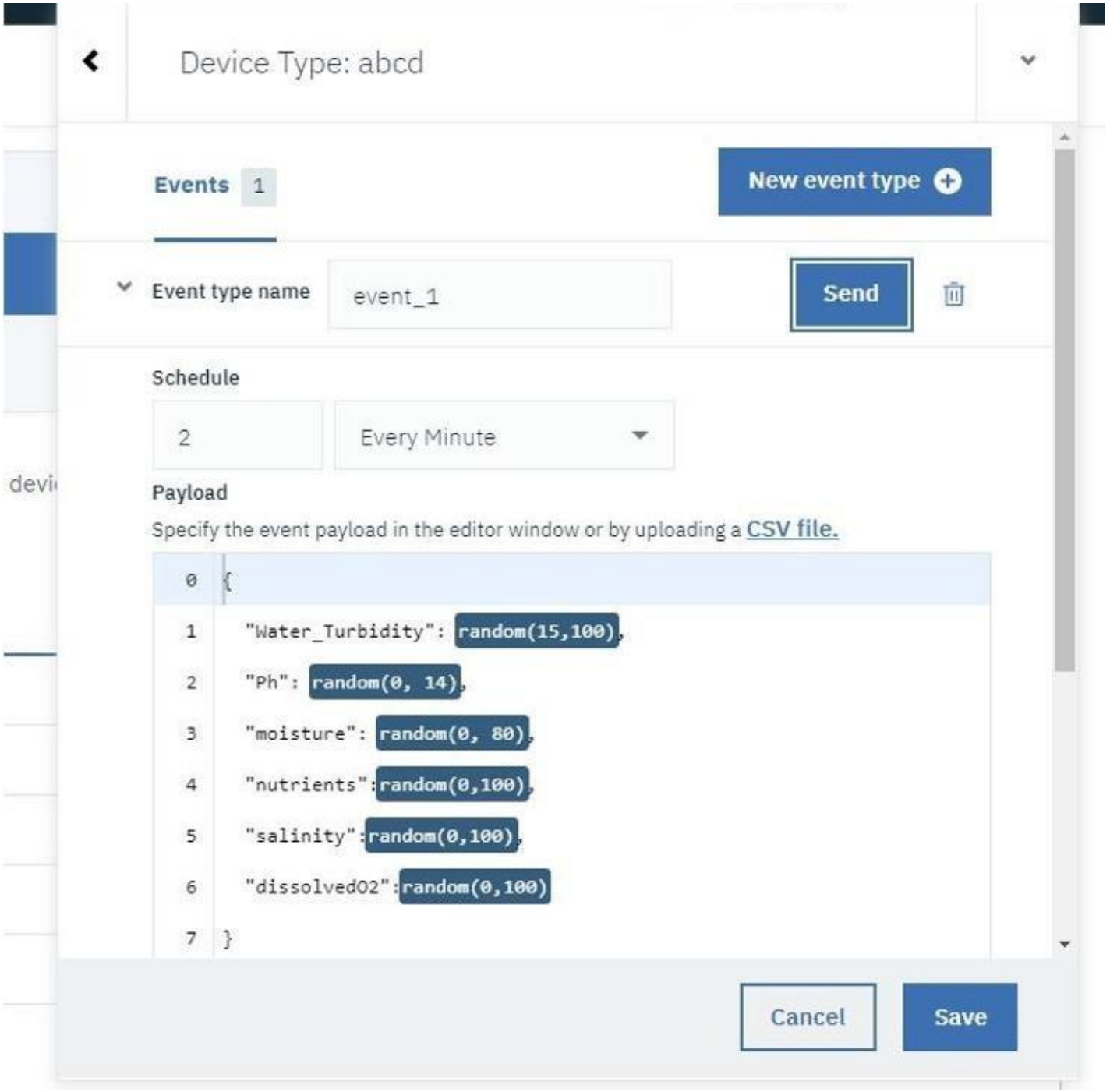
FOR BUTTONS

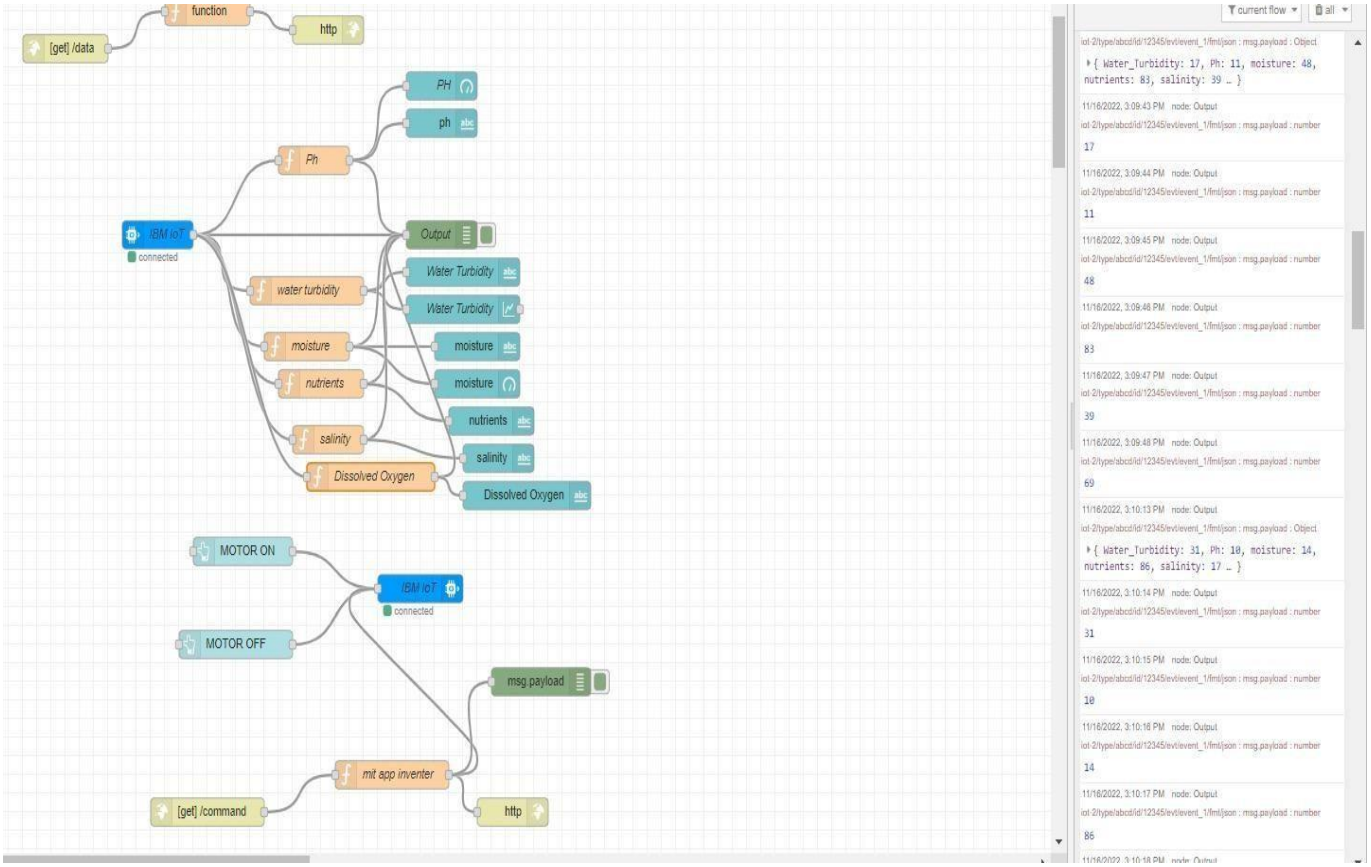




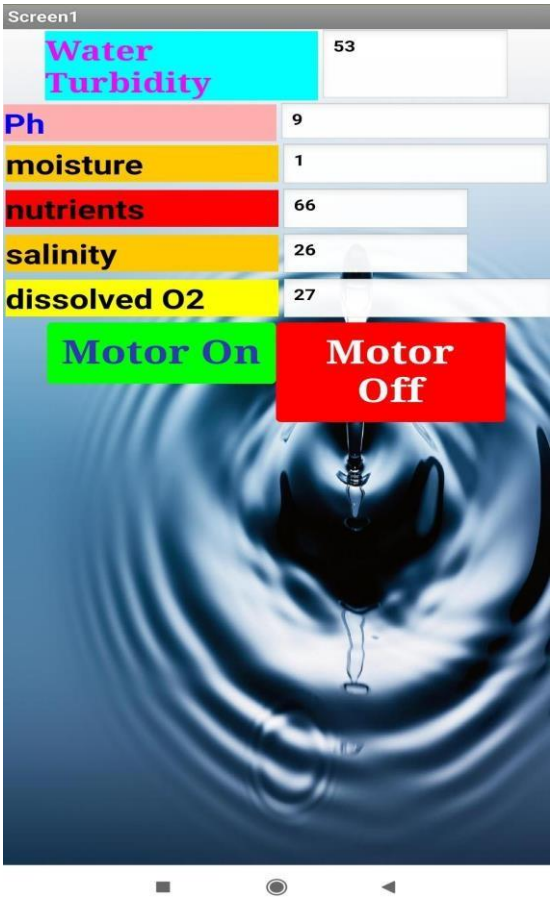
https://node-red-qltdp-2022-11-07.eu-gb.mybluemix.net/data

{"Ph":6,"waterturbidity":78,"moisture":73,"nutrients":10,"salinity":38,"dissolvedO2":50}


Device Type: abcd
Events 1
New event type
Event type name
event_1
Send
Schedule
2
Every Minute
Payload
Specify the event payload in the editor window or by uploading a CSV file.
{
"Water_Turbidity": random(15,100),
"Ph": random(0, 14),
"moisture": random(0, 80),
"nutrients":random(0,100),
"salinity":random(0,100),
"dissolved02":random(0,100)
}
Cancel
Save



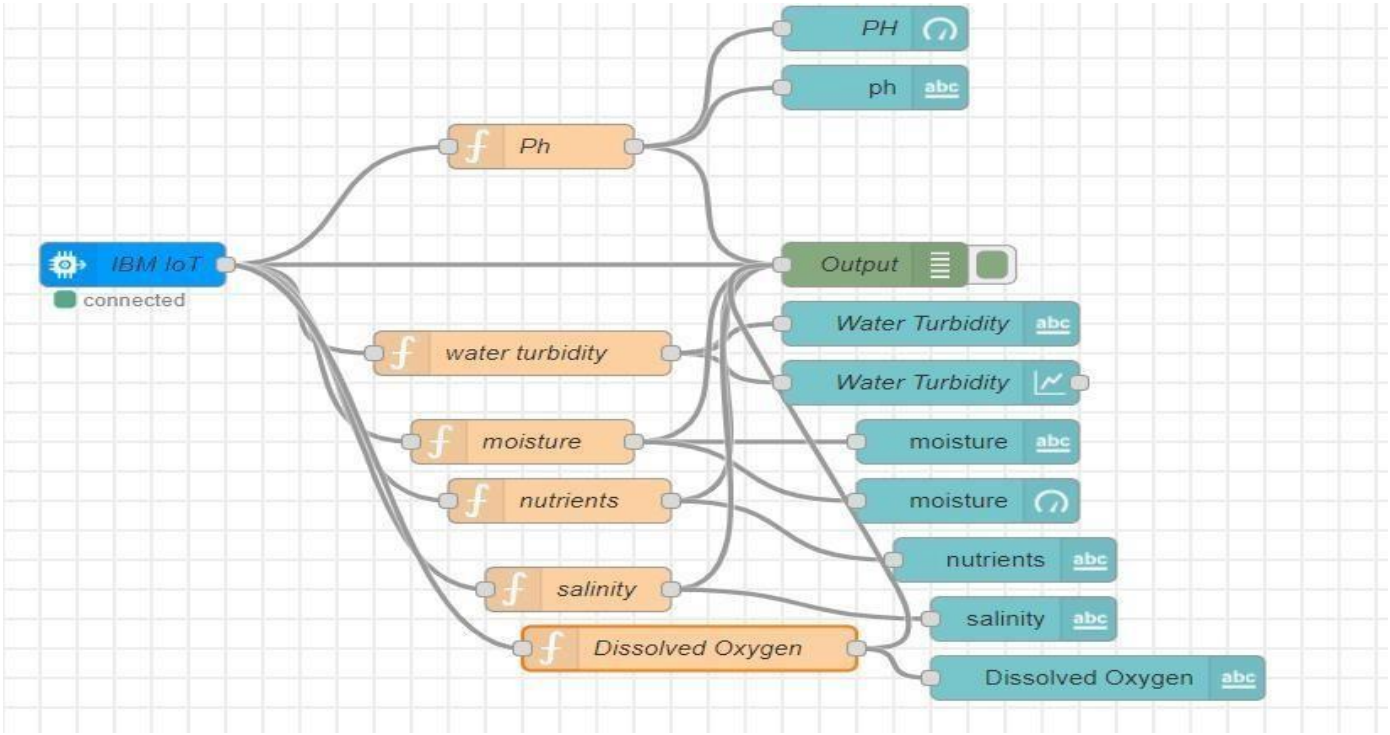
APP IN MOBILE

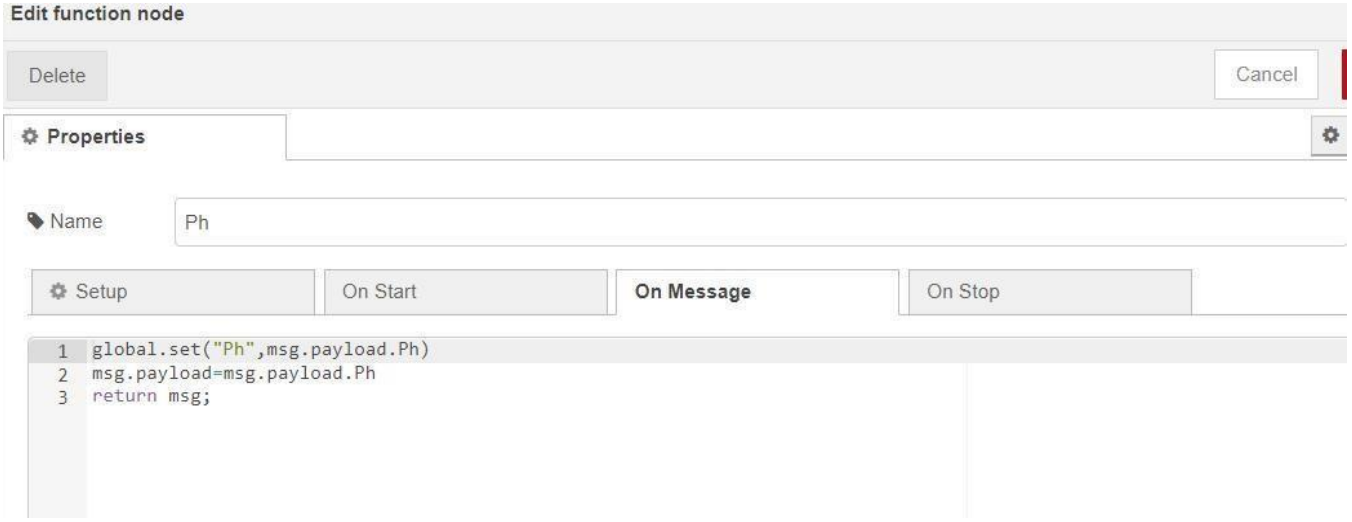


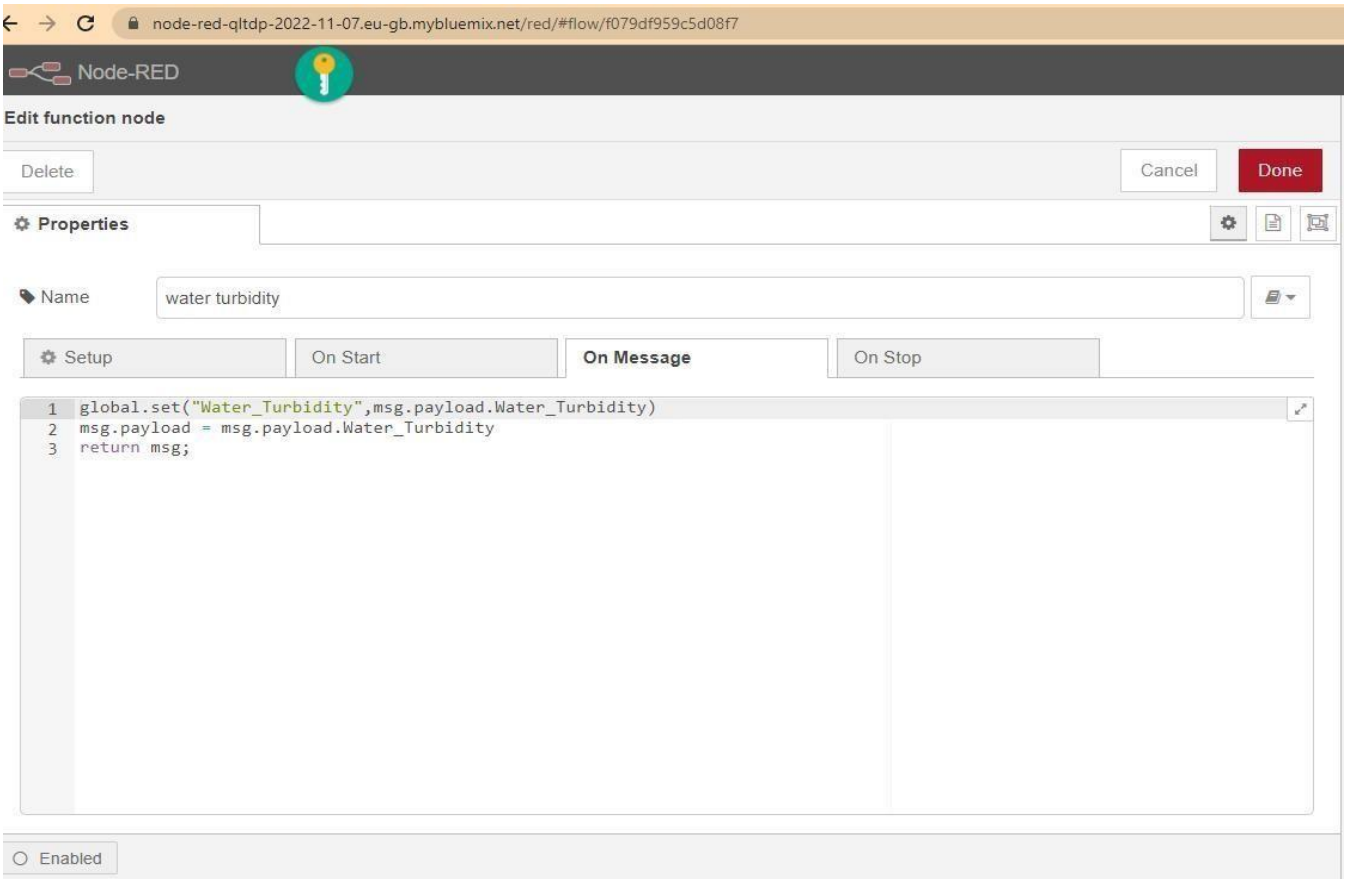
SPRINT 4

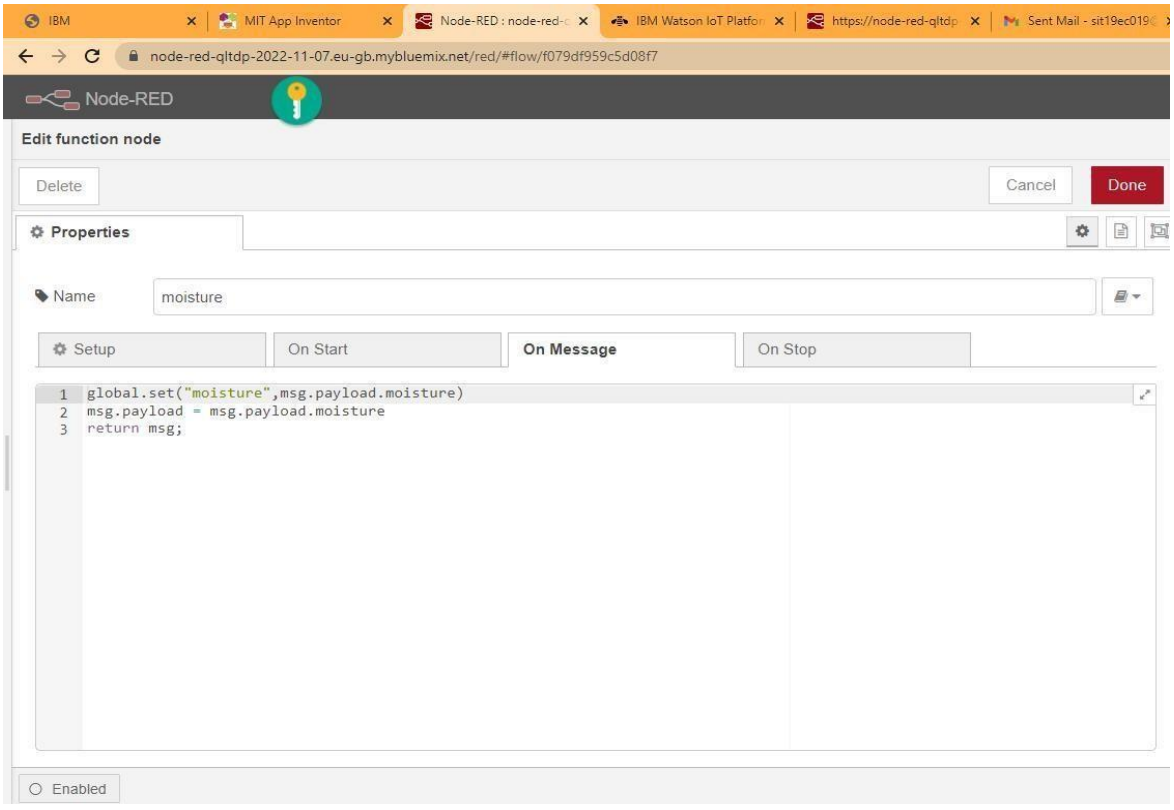
NODE RED UI:

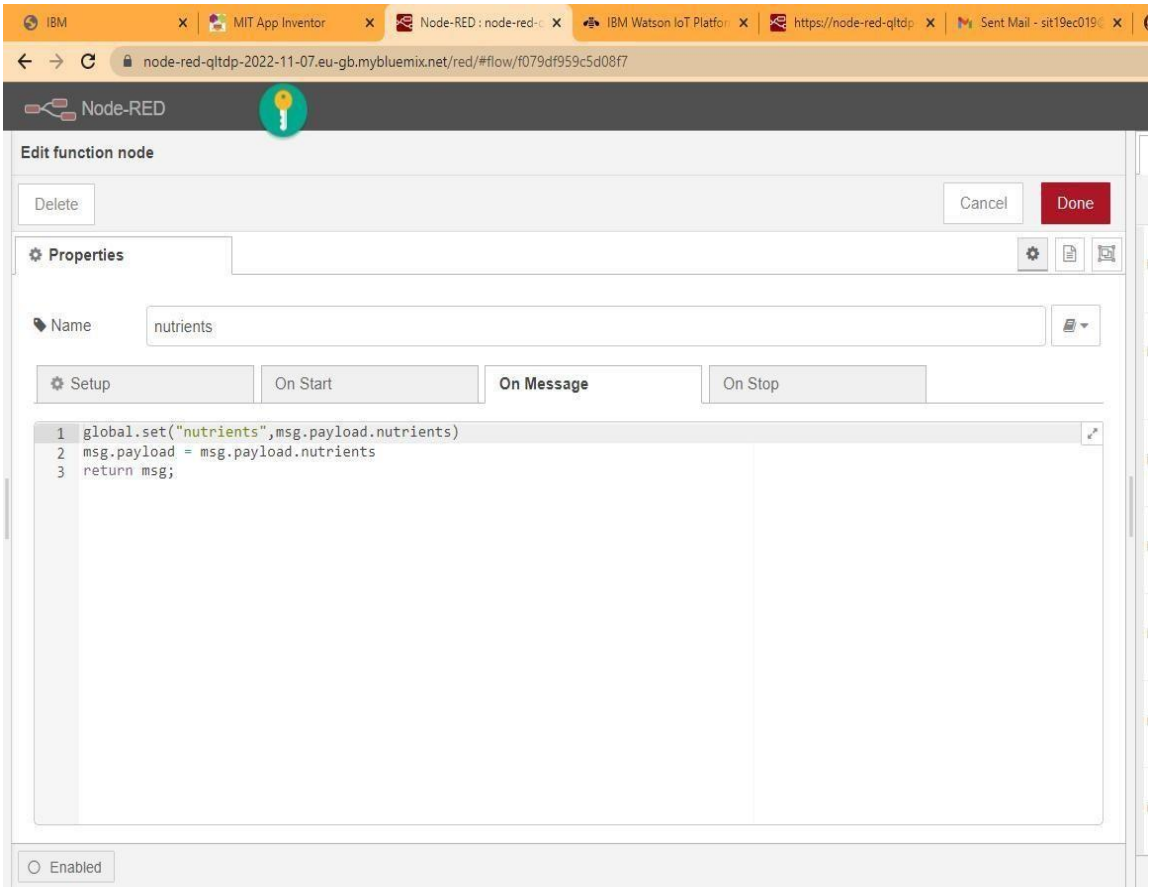
Node-red-dashboard

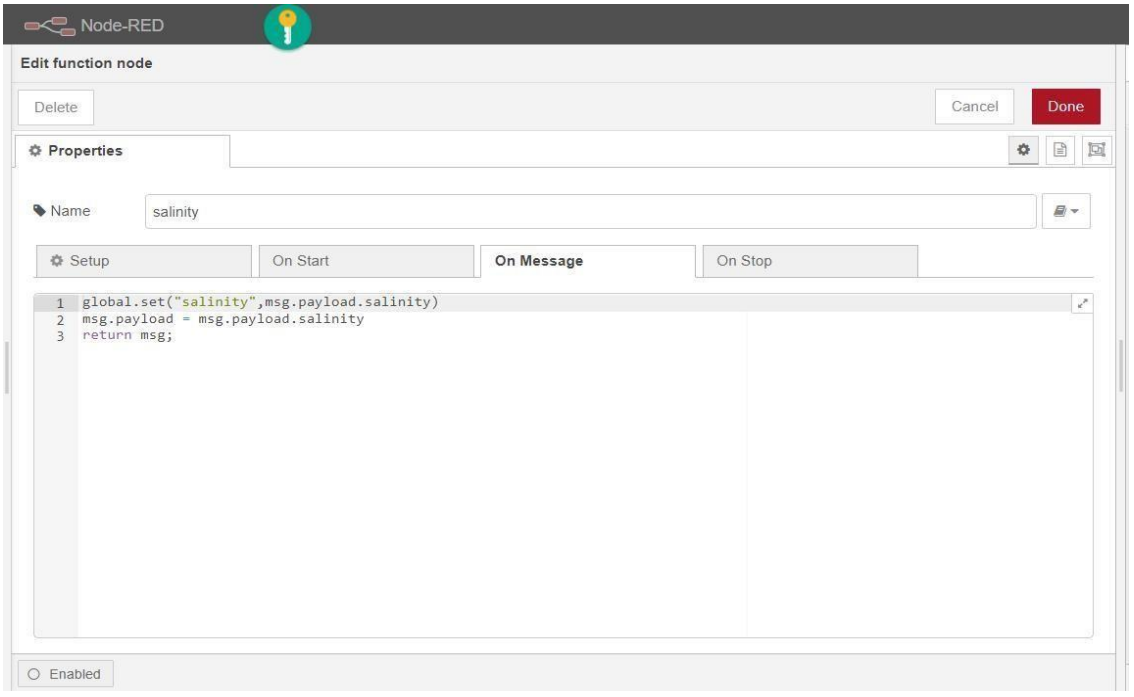




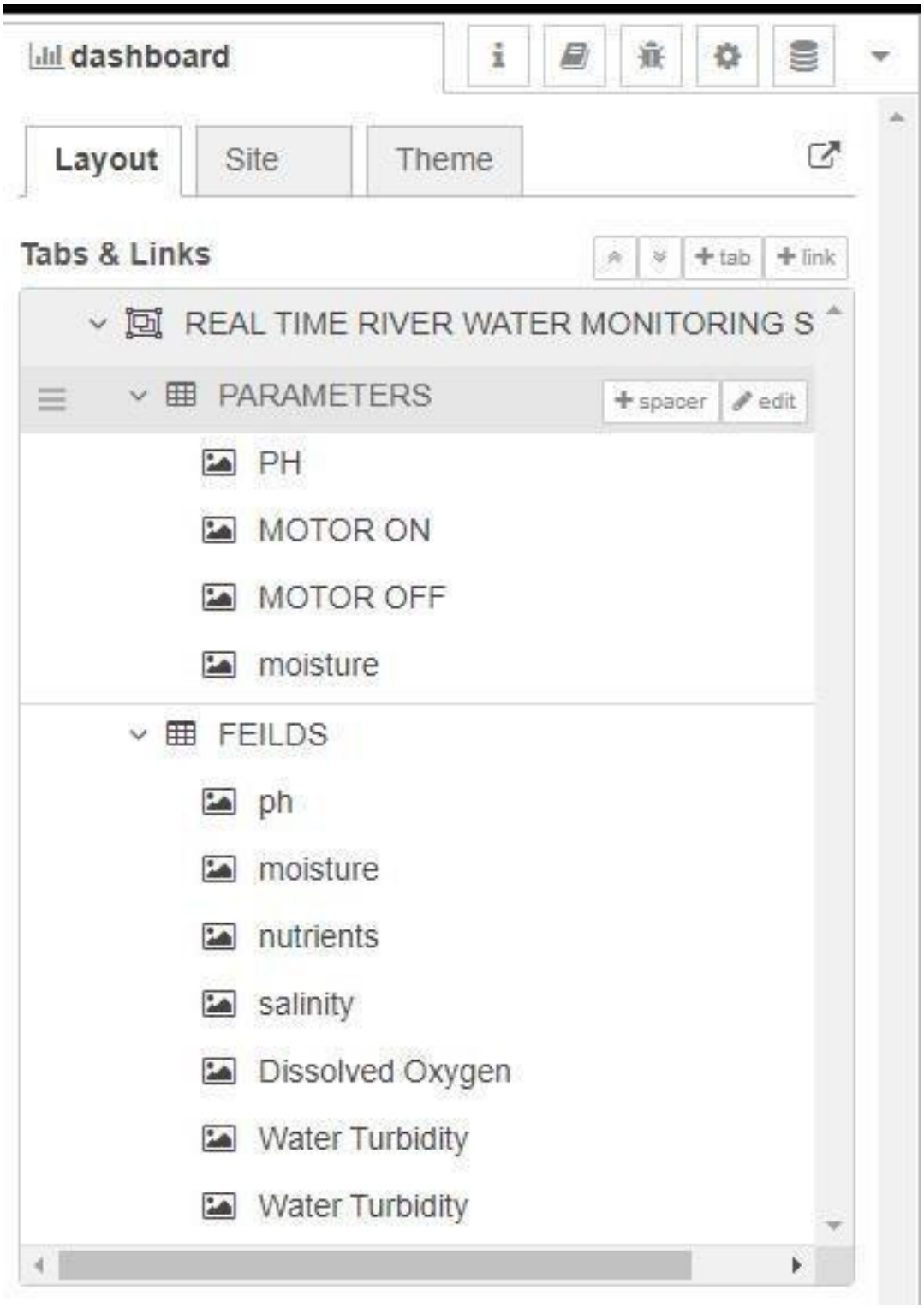

node-red-qltdp-2022-11-07.eu-gb.mybluemix.net/red/#flow/f079df959c5d08f7
Node-RED
Edit function node
Delete
Cancel
Done
Properties
Name
water turbidity
Setup
On Start
On Message
On Stop
1 global.set("Water_Turbidity",msg.payload.Water_Turbidity)
2 msg.payload = msg.payload.Water_Turbidity
3 return msg;
Enabled

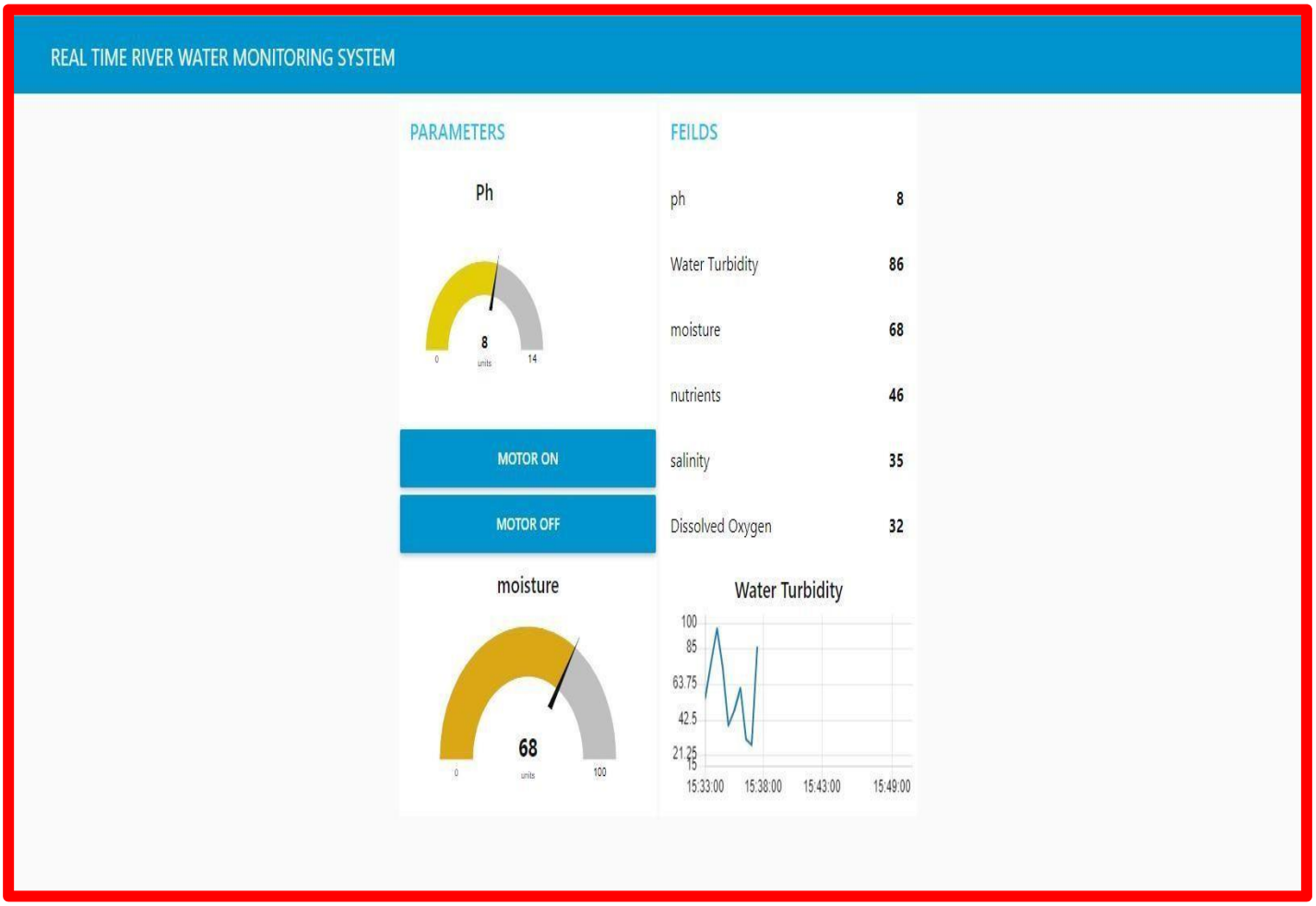

IBM
MIT App Inventor
Node-RED : node-red-
IBM Watson IoT Platfor
https://node-red-qltdp-
Sent Mail - sit19ec019
node-red-qltdp-2022-11-07.eu-gb.mybluemix.net/red/#flow/f079df959c5d08f7
Node-RED
Edit function node
Delete
Cancel
Done
Properties
Name
moisture
Setup
On Start
On Message
On Stop
1 global.set("moisture",msg.payload.moisture)
2 msg.payload = msg.payload.moisture
3 return msg;
Enabled


IBM
MIT App Inventor
Node-RED : node-red-c
IBM Watson IoT Platfor
https://node-red-qltdp
Sent Mail - sit19ec019
node-red-qltdp-2022-11-07.eu-gb.mybluemix.net/red/#flow/f079df959c5d08f7
Node-RED
Edit function node
Delete
Cancel
Done
Properties
Name
nutrients
Setup
On Start
On Message
On Stop
1 global.set("nutrients",msg.payload.nutrients)
2 msg.payload = msg.payload.nutrients
3 return msg;
Enabled


Node-RED
Edit function node
Delete
Cancel
Done
Properties
Name
salinity
Setup
On Start
On Message
On Stop
1 global.set("salinity",msg.payload.salinity)
2 msg.payload = msg.payload.salinity
3 return msg;
Enabled

UI


dashboard
Layout
Site
Theme
Tabs & Links
+ tab
+ link
REAL TIME RIVER WATER MONITORING S
PARAMETERS
+ spacer
edit
PH
MOTOR ON
MOTOR OFF
moisture
FEILDS
ph
moisture
nutrients
salinity
Dissolved Oxygen
Water Turbidity
Water Turbidity



GITHUB link

https://github.com/IBM-EPBL/IBM-Project-6233-1658824960

video link

https://youtu.be/eX8ZXIB6nZ8