PROJECT BASED EXPERIENTIAL LEARNING PROGRAM (NALAIYA THIRAN)

Real-Time River Water Quality Monitoring and Control System

A PROJECT REPORT

Submitted by

Harshavarthini R (111719106052)

Preetha S (111719106302)

Ravilla Nisha (111719106125)

Sakthipriya B (111719106134)

Sangavi N (111719106135)

TEAM ID : PNT2022TMID16162

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

R.M.K.ENGINEERING COLLEGE

(An Autonomous Institution)

R.S.M. Nagar, Kavaraipettai-601 206

NOVEMBER 2022

1. INTRODUCTION

1.1 Project Overview

The need for effective and efficient monitoring, evaluation and control of water quality in residential area has become more demanding in this era of urbanization, pollution and population growth. Ensuring safe water supply of drinking water is big challenge for modern civilization. Traditional methods that rely on collecting water samples, testing and analyses in water laboratories are not only costly but also lack capability for real-time data capture, analyses and fast dissemination of information to relevant stakeholders for making timely and informed decisions. In this paper, a real time water quality monitoring system prototype developed for water quality monitoring in Residential home is presented. The development was preceded by evaluation of prevailing environment including availability of cellular network coverage at the site of operation. It detects water temperature, dissolved oxygen, pH, and electrical conductivity in real-time and disseminates the information in graphical and tabular formats to relevant stakeholders through a web-based portal and mobile phone platforms. The experimental results show that the system has great prospect and can be used to operate in real world environment for optimum control and protection of water resources by providing key actors with relevant and timely information to facilitate quick action taking.

1.2 Purpose

Monitoring data is used to determine whether or not pollution regulations are being complied with. From oil spills and radiation leaks to floods and mass erosion, water quality monitoring data is a must when developing emergency strategies. To protect, restore, and enhance environmental quality towards good public health, environmental integrity, and economic viability. Monitoring is essential to ensure that the intended project objective can be achieved within the given time frame following the activities as planned to be carried out by project personnel. Quality control consists of evaluating 2 main processes during data collection. One is instrumentation control, which monitors the instrument used to collect data. The other is population quality control, which monitors the sample to make sure it is representative of the target population.

2. **LITERATURE SURVEY**

AUTHOR	PARAMETERS	MODULES	POWER SUPPLY	APPLICATION SCENARIOS
Shoa et al.(2012)	Temperature, pH	LPC2138 ARM-7 core, CC2530 ZigBee module	Not Specified	Sewage treatment plants, Range is limited to 1.5km
Khetre and Hate(2013)	Temperature, Turbidity, Water level, Salinity	ARM-7 MCU, ZigBee module (30m range)	Not Specified	Surface water sources such rivers, lakes and dams
Vijayakumar and Ramya(2015)	Temperature, Turbidity, pH, Conductivity, Dissolved oxygen	Raspberry Pi model B+, USR WIFI 232 module	Not Specified	Water sources
Chung and Yoo(2015)	Temperature, Dissolved oxygen, pH, Conductivity, Turbidity, Depth of water	ATmega 128, IEEE 802.15.4(100M RANGE)	12V battery + solar panel	Streams, Rivers and Coastal areas
He and Zhang(2013)	Temperature, PH, Turbidity	CC2430 MCU, ZigBee	Batteries	Industrial plants and Aquaculture

2.1Existing problem

Rivers and streams drain water that falls in upland areas. Moving water dilutes and decomposes pollutants more rapidly than standing water, but many rivers and streams are significantly polluted all around the world. Water pollution has very negative effects on public health. A lot of diseases result from drinking or being in contact with contaminated water, such as diarrhea, cholera, typhoid, dysentery or skin infections. In zones where there is no available drinking water, the main risk is dehydration obviously.

References

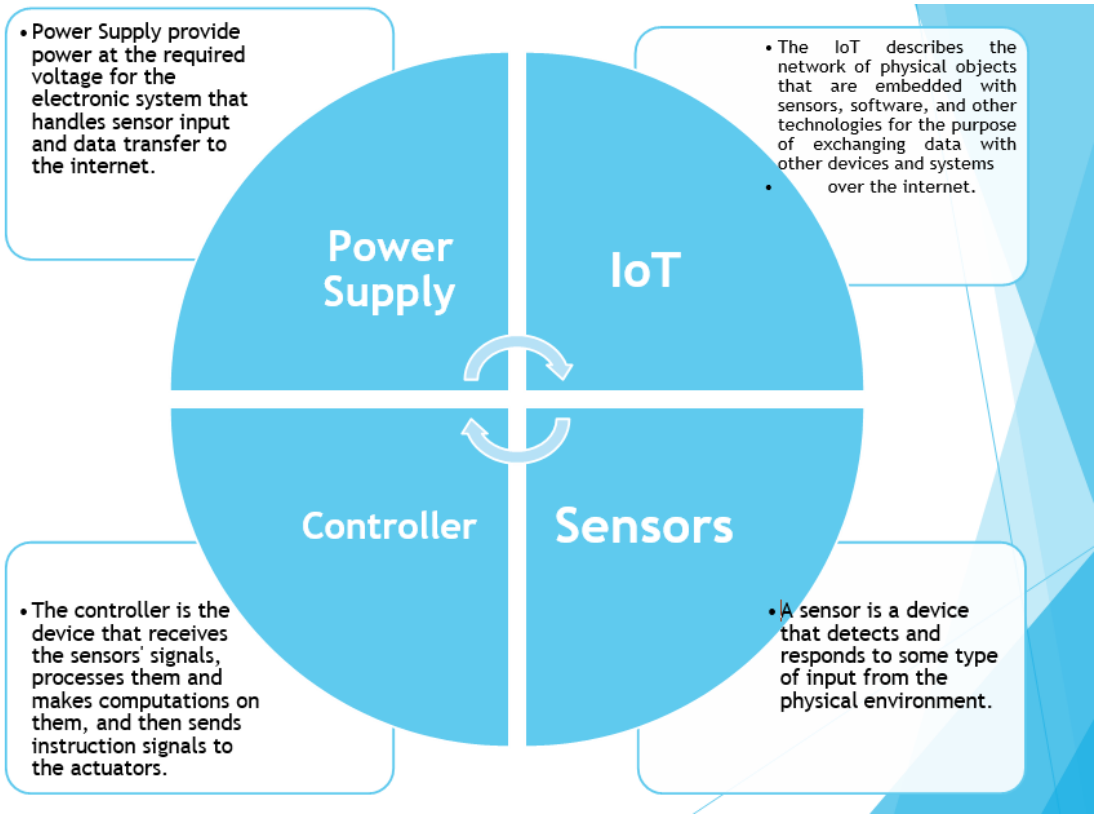
1. Sridharan, S. (2014) Water Quality Monitoring System Using Wireless Sensor Network. International Journal of Electronic Communications Engineering Advanced Research,3,399-402
2. Losilla, F., Garcia-Sanchez, A.-J., Garcia-Sanchez, F., Garcia-Haro, J. and Haas, Z.J. (2011) A Comprehensive Approach to WSN-Based ITS Application. Sensors, 10, 10220-10265. http://dx.doi.org/10.3390/s111110220
3. Mo Deqing, Zhao Ying, Chen Shangsong, "Automatic Measurement and Reporting System of Water Quality Based on GSM," 2012 International Conference on Intelligent System Design and Engineering Application.
4. Nikhil Kedia, Water Quality Monitoring for Rural Areas- A Sensor Cloud Based Economical Project, in 1st International Conference on Next Generation Computing Technologies (NGCT2015) Dehradun, India, 4-5 September 2015. 978- 1-4673-6809-4/15/$31.00 ©2015 IEEE

Problem Statement Definition

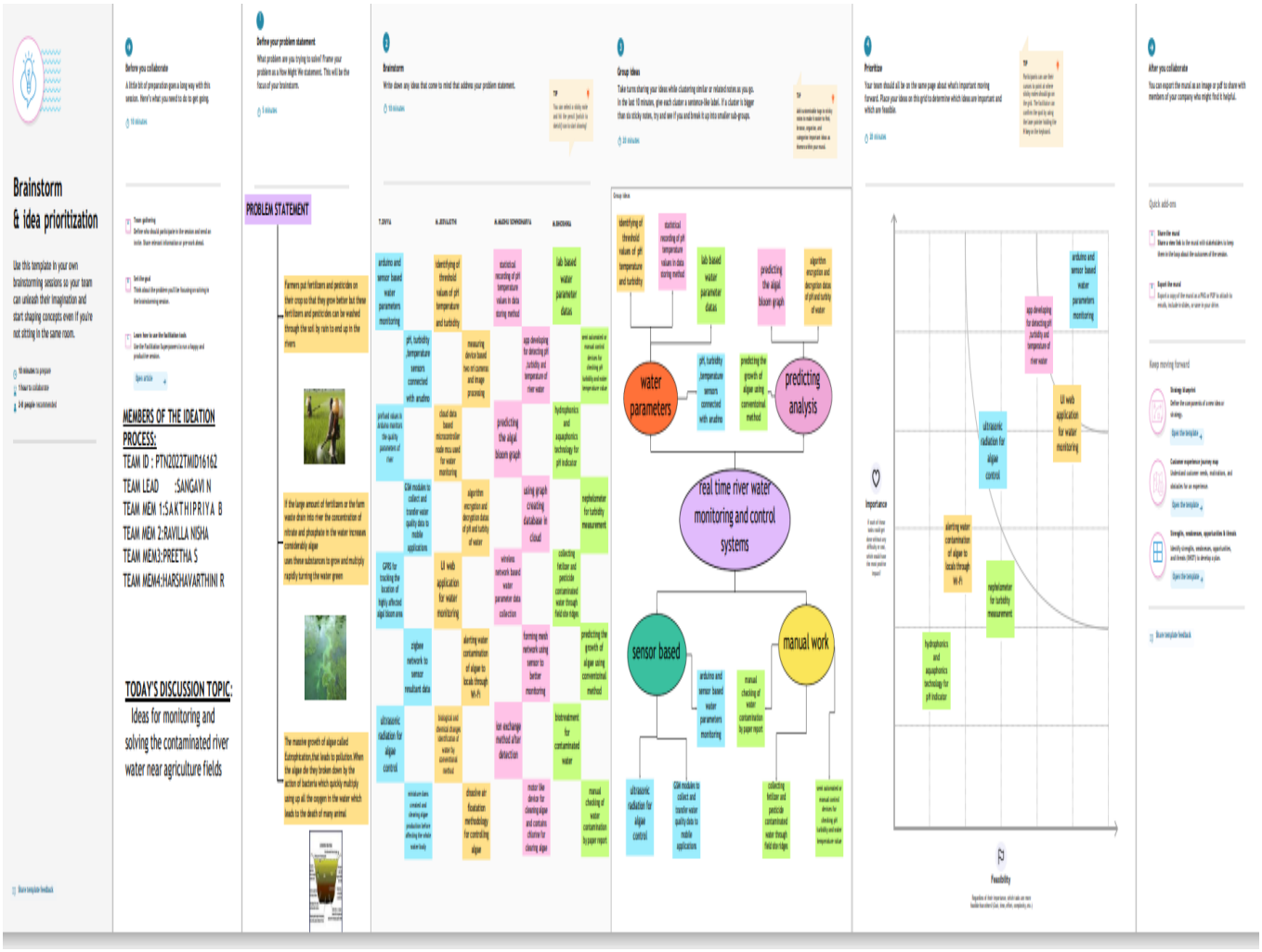
Due to the fast growing urbanization supply of safe drinking water is a challenge for the every city authority. Water can be polluted any time. So the water we reserved in the water tank at our roof top or basement in our society or apartment may not be safe. Still in India most of the people use simple water purifier that is not enough to get surety of pure water. Sometimes the water has dangerous particles or chemical mixed and general purpose water purifier cannot purify that. And it's impossible to check the quality of water manually in every time. So an automatic real-time monitoring system is required to monitor the health of the water reserved in our water tank of the society or apartment. So it can warn us automatically if there is any problem with the reserved water. And we can check the quality of the water anytime and from anywhere. By keeping this mind we designed this system especially for residential areas.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming



3.3 Proposed Solution

The main aim is to develop a system for continuous monitoring of river waterquality at remote places using wireless sensor networks with low power consumption, low-cost and high detection accuracy. pH, conductivity, turbiditylevel, etc. are the limits that are analysed to improve the water quality.

Following are the aims of idea implementation

(a) To measure water parameters such as pH, dissolved oxygen, turbidity, conductivity, etc. using available sensors at a remote place.

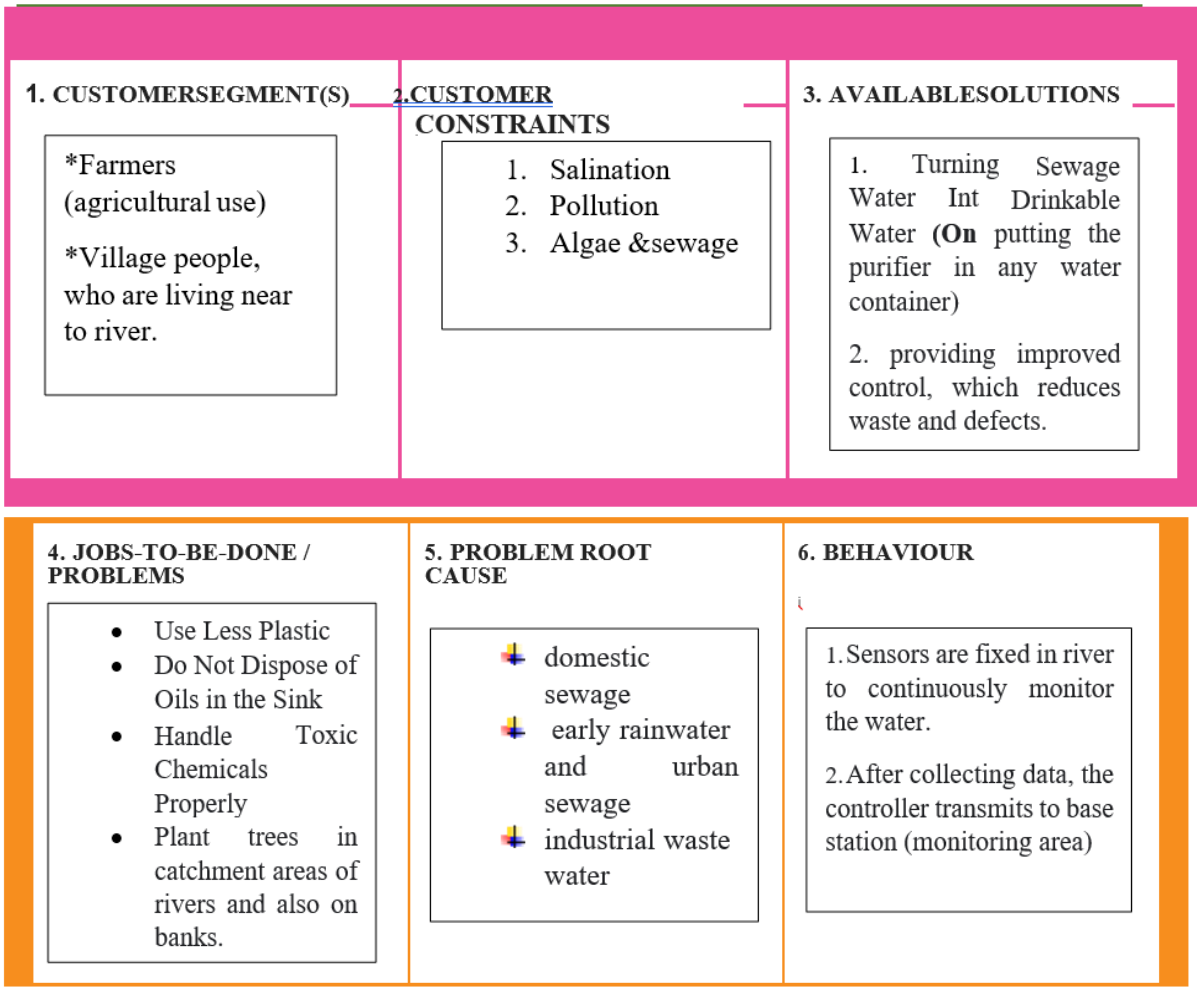
(b) To assemble data from various sensor nodes and send it to the base station by the wireless channel.

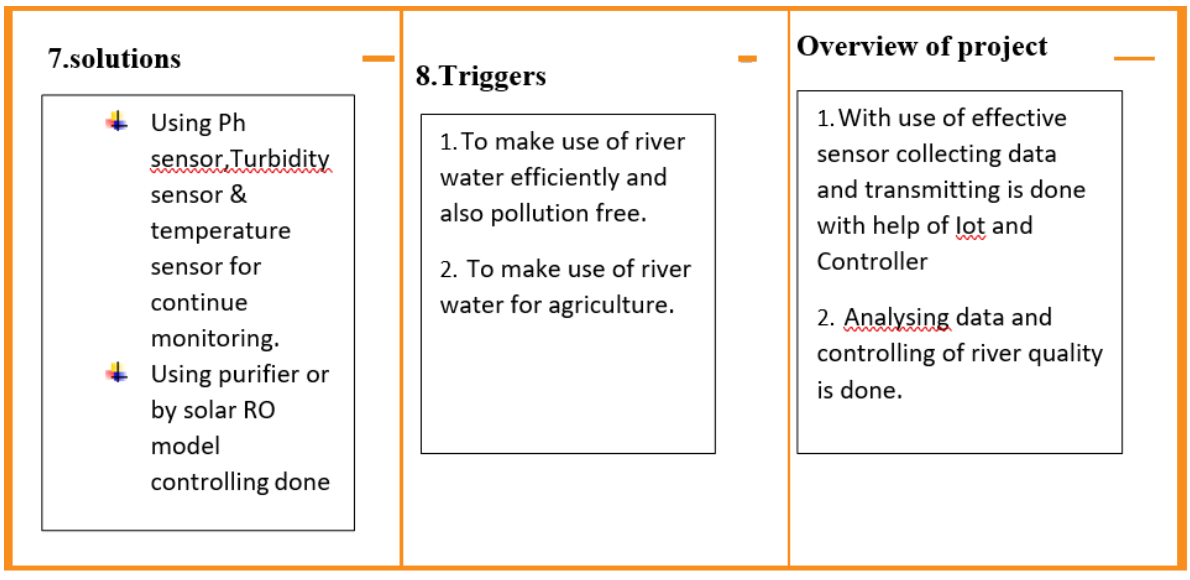
(c) To simulate and evaluate quality parameters for quality control.

(d) To intimate to an authorized person routinely when water quality detected does not match the pre-set standards, so that, necessary actions can be taken.

In the proposed architecture, each water reservoir will be attached with a sensor node equipped with a set of sensor probes capable of measuringthe parameters like pH, turbidity etc. According to the specifications of the sensor probes and the processor board of the sensor the signal conditioning circuit will be designed to generate the sensor output to the processor board through Analog to Digital Converter. The processor board processes the data according to the quality specifications and transmits to the central server through the transceiver. The measured data in each of the reservoir shall be sent to the central server through the respective transceivers either directly orindirectly through other sensor or repeater nodes.

3.4 Problem Solution fit





4. REQUIREMENT ANALYSIS

4.1 Functional requirement

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Arduino(control system)	Sensors are interfaced to Arduino and it collects measurements data periodically from sensors.
FR-2	WSN Sensor	Multiple sensor nodes installed for the detection of pH, temperature, dust particles, turbidity.
FR-3	Software Design Requirements	WSN requires IoT platform which requires Neural Network Model to classify water quality as Good Or Bad. IoT integrated big data analytics to store data in cloud and analyze it constantly.
FR-4	LCD/PC/Mobile display	Displays the resulting sensed pH, temperature, turbidity. If ,acquired value > Threshold value, then comment=BAD. If, acquired value < Threshold value, then comment=GOOD.

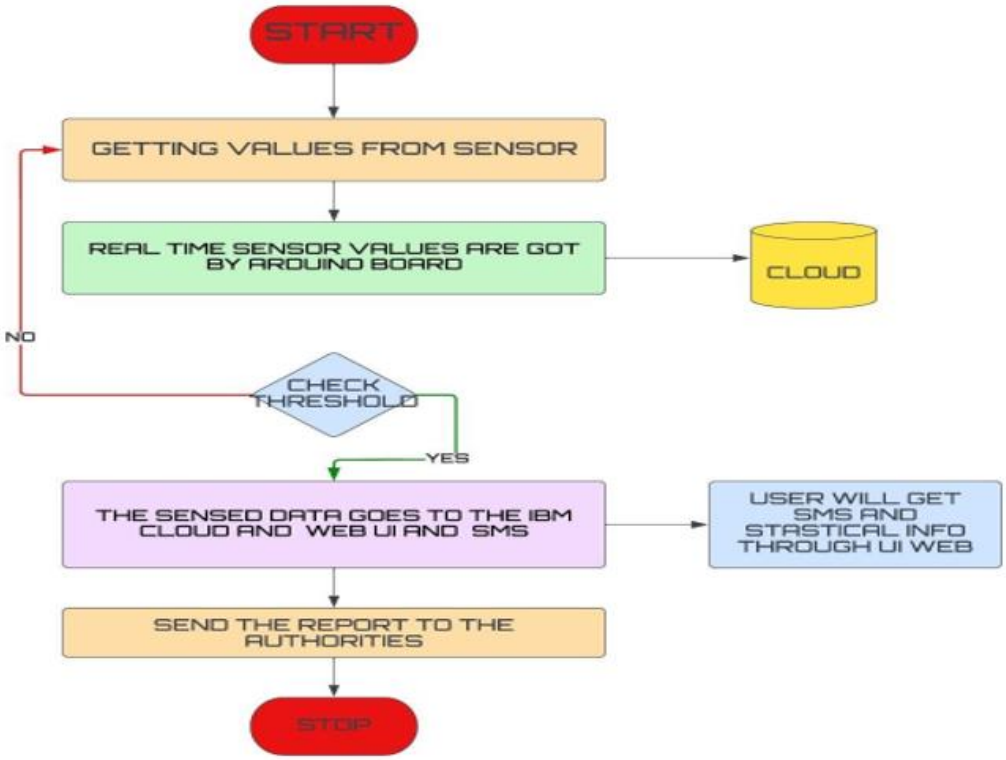
Non-functional Requirements:

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

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	It is important to monitor water quality to ensure that, it is safe for humans to drink it as well as for wild life and marine life and to understand environmental impacts and to not harm sea life.
NFR-2	**Security**	The IoT networks are incredibly safe and communication speed is also high. The technology comfortably resolves all the issues.
NFR-3	**Reliability**	The water quality and monitoring system is reliable and it's output can be assured. Since standardized hardware components and software designs are used.
NFR-4	**Performance**	Real-time quality of water is executed and alertring the authorities if water quality is not good.
NFR-5	**Availability**	The monitoring system is made available for use at any time with accuracy.
NFR-6	**Scalability**	The system with high frequency, high mobility and low powered and cost-effective.

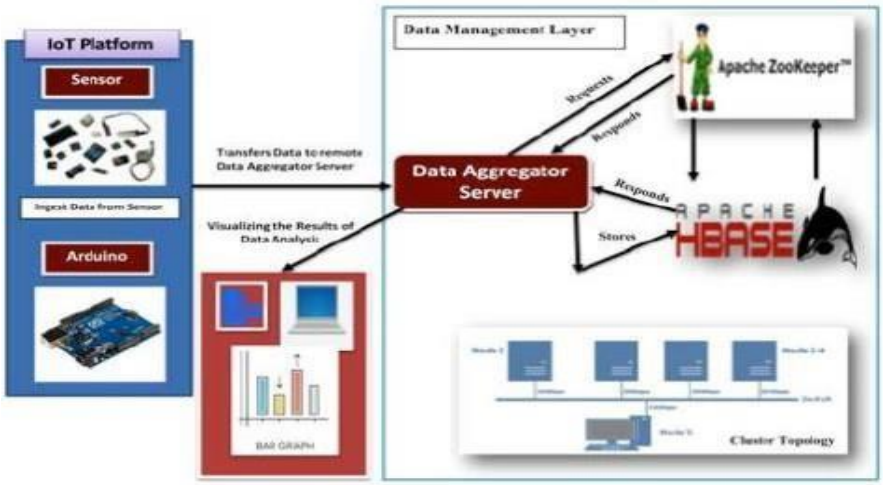
5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution & Technical 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:



- **Arduino MegaBoard.**

Arduino is an open-source electronics prototyping platform based on flexible, easy-to- use hardware and software. Its intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments. Or more simply, you load on some code and it can read sensors, perform actions based on inputs from buttons, control motors, and accept shields to further expand its capabilities. Really, you can do almost anything.

All Arduino boards have one thing in common: they are programmed through the Arduino IDE. This is the software that allows you to write and upload code. Beyond that, there can be a lot of differences. The number of inputs and outputs (how many sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, and form factor are just a few of the variables. Some boards are designed to be embedded and have no programming interface (hardware) which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V.

Check the chart on the next page to find the right Arduino for your project.

Fig 2 :Ardino mega board

- **liquid-crystal display (LCD):**

It is a flat-panel display or other electronically modulated optical device that uses the light-modulating properties of liquid crystals. Liquid crystals do not emit light directly, instead using a backlight or reflector to produce images in color or monochrome.[1] LCDs are available to display arbitrary images (as in a general-purpose computer display) or fixed images with low information content, which can be displayed or hidden, such as preset words, digits, and seven- segment displays, as in a digital clock.

- **The Temperature and PH sensor.**

To measure the temperature of a water tank and log it via the Arduino. The idea is to maintain the temperature of the water at 25-30C at all times. I've noticed that most applications have used a sensor such as DS18S20 or TMP35/TMP36/TMP37. but since my application requires to measure the temperature in water, I think a more suitable sensor should have a waterproof probe (or external probe).

The usual way is to contain the water inside a tank / container which can transmit heat – usually metal. To the outside of this is then bonded the temperature sensor – be that a simple bi- metallic strip thermostat, or a more complex temperature sensing transducer. Of course, this requires a metal tank, and that will radiate heat, which will be wasteful. Ideally you would want some form of waterproof probe. You haven't mentioned the amount of water you're dealing with – how big is the tank? How deep especially.There are thermocouples available in a rigid probe form – quite how waterproof these are I'm not sure, but these are never very long, so you won't be able to get it more than 6 inches or so into the water before you risk complete submersion.

- ## The Turbidity Sensor

Turbidity is an indicator often used to find the amount of suspended sediment in water. By cumbersome mechanical

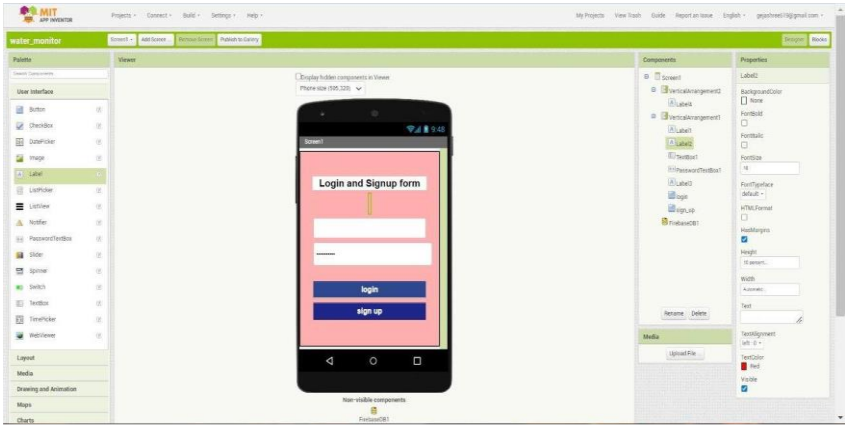
sampling, it is possible to measure the concentration of suspended solids (in mg/l) in water, but turbidity is increasingly used instead, as it is easy to use and cheaper too. It is an ecologically important parameter as the various effects of suspended solids in aquatic ecosystems are due to their light scattering properties rather than their absolute mass.

5.3 User Stories

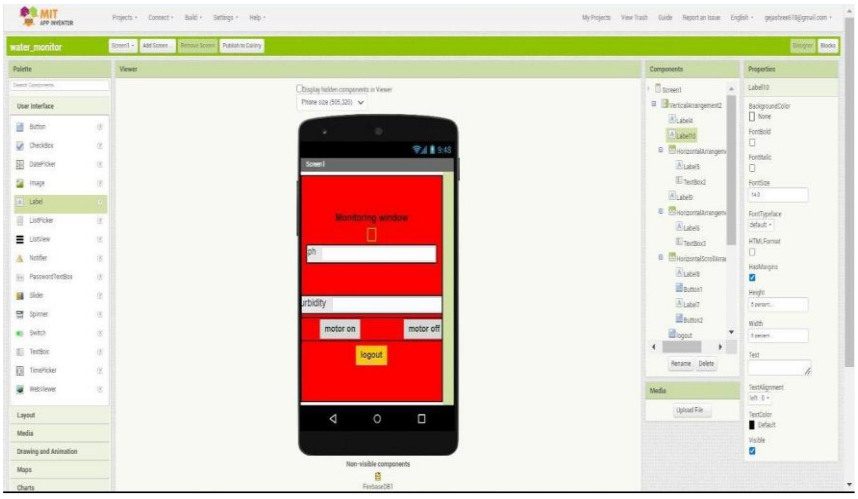
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering email, password, and confirming my password.	I can access my account/dashboard	High	Sprint-1
		USN-2	As a user, I will receive a confirmation email once I have registered for the application	I can receive e confirmation email & click confirm	High	Sprint-2
		USN-3	As a user, I can register for the application through Google	I can register & access the dashboard with Google	High	Sprint-1
		USN-4	As a user, I can register for the application through Gmail	I can register through the mail.	Medium	Sprint-2
	Login	USN-5	As a user, I can log into the application by entering email, password & captcha	I can receive login credentials.	High	Sprint-1
	Interface	USN-6	As a user, the interface should be user-friendly manner	I can able to access easily.	Medium	Sprint-1
Customer (Web user)	dashboard	WUSN-1	As a web user, I can access the specific info (ph value, temp, humidity, quality).	I can able to know the quality of the water.	High	Sprint-1
Customer Care Executive (input)	View manner	CCE-1	As a customer care, I can view data in visual representation manner(graph)	I can easily understand by visuals.	High	Sprint-1
	Taste	CCE-2	As a customer care, I can able to view the quality(salty) of the water	I can easily know whether it is salty or not	High	Sprint-1
	Color visibility	CCE-3	As a customer care, I can able predict the water color	I can easily know the condition by color	High	Sprint-1
Administrator	Risk tolerant	ADMIN-1	An administrator who Is handling the system should update and take care of the application.	Admin should monitor the records properly.	High	Sprint-2

6. PROJECT PLANNING & SCHEDULING

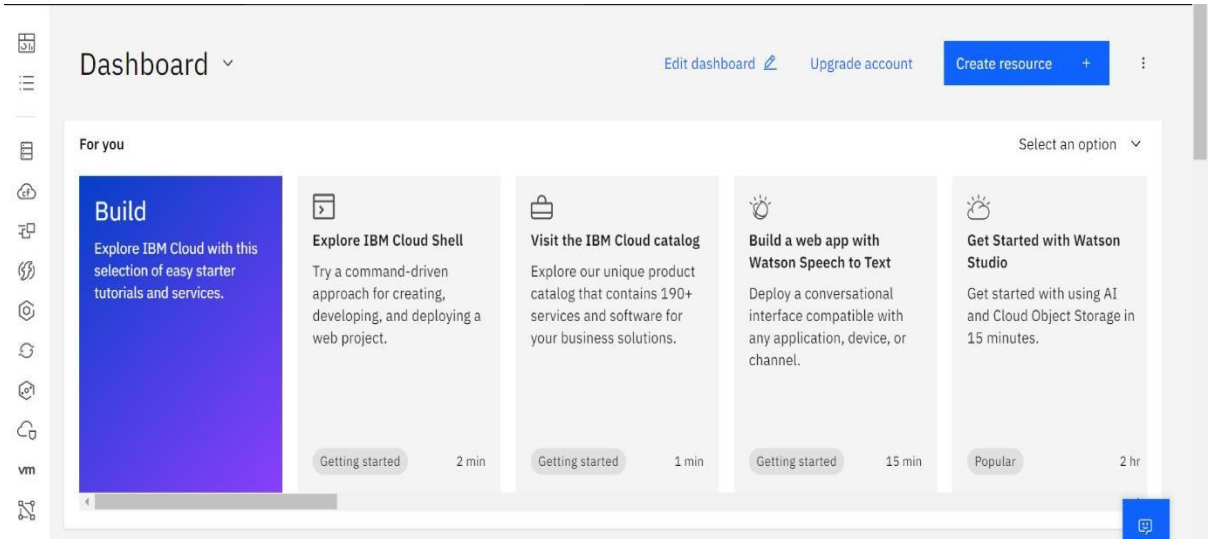
SPRINT 1 : LOGIN PAGE:



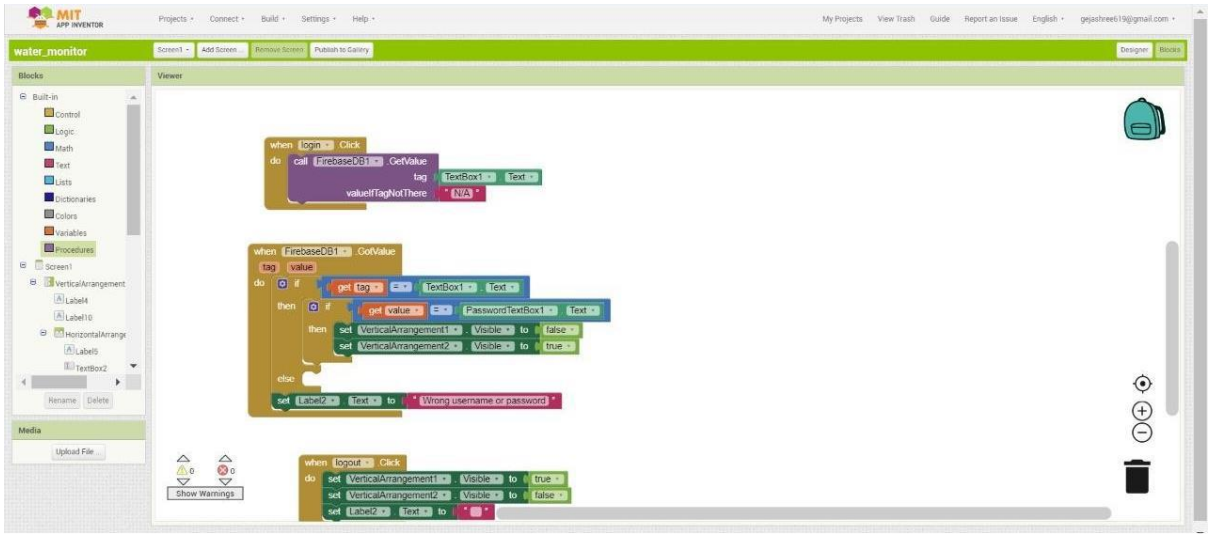
MONITORING WINDOW:



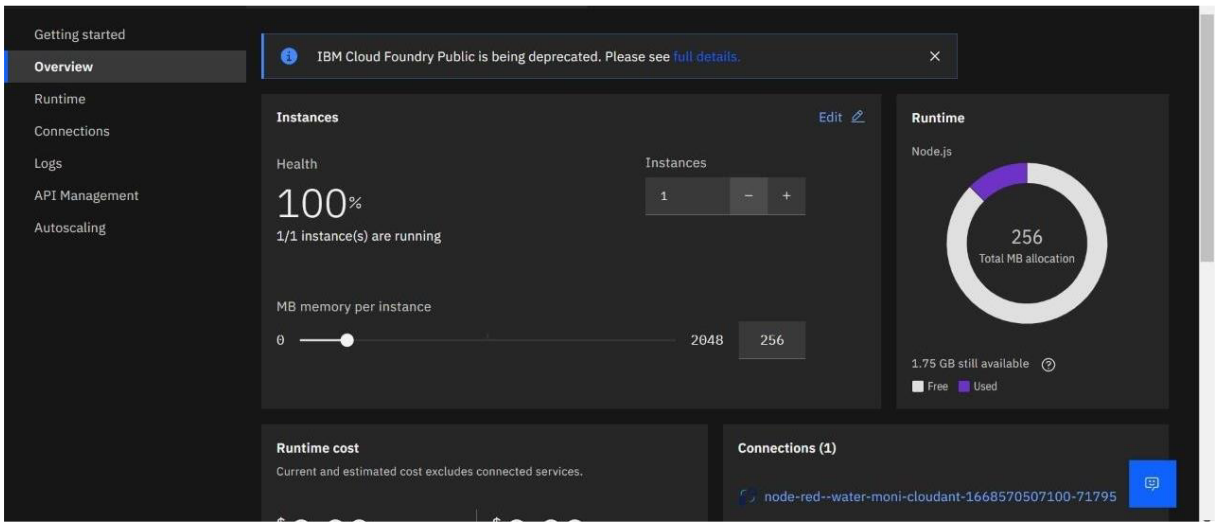
IBM CLOUD ACCOUNT- DASHBOARD:



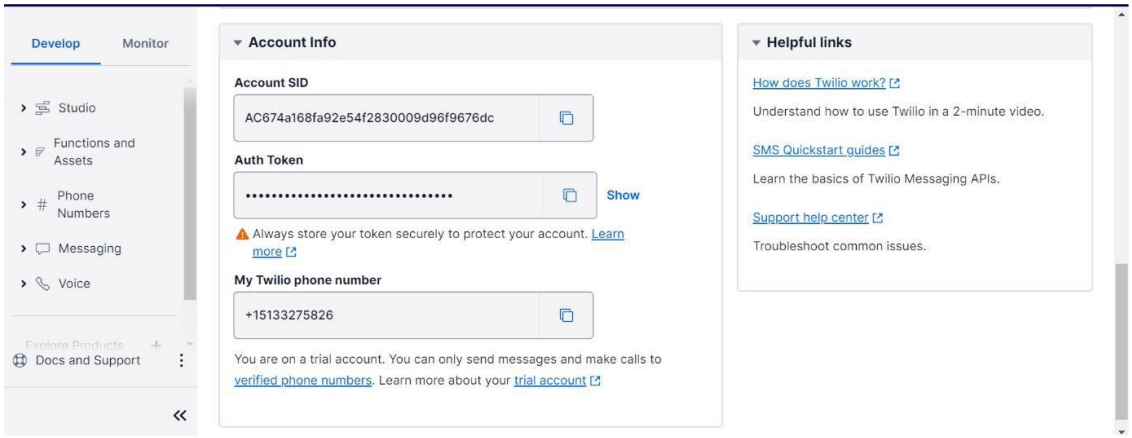
SECURE LOGIN: : Real Time Database Connection



NODE RED FLOW



Twilio account creation:



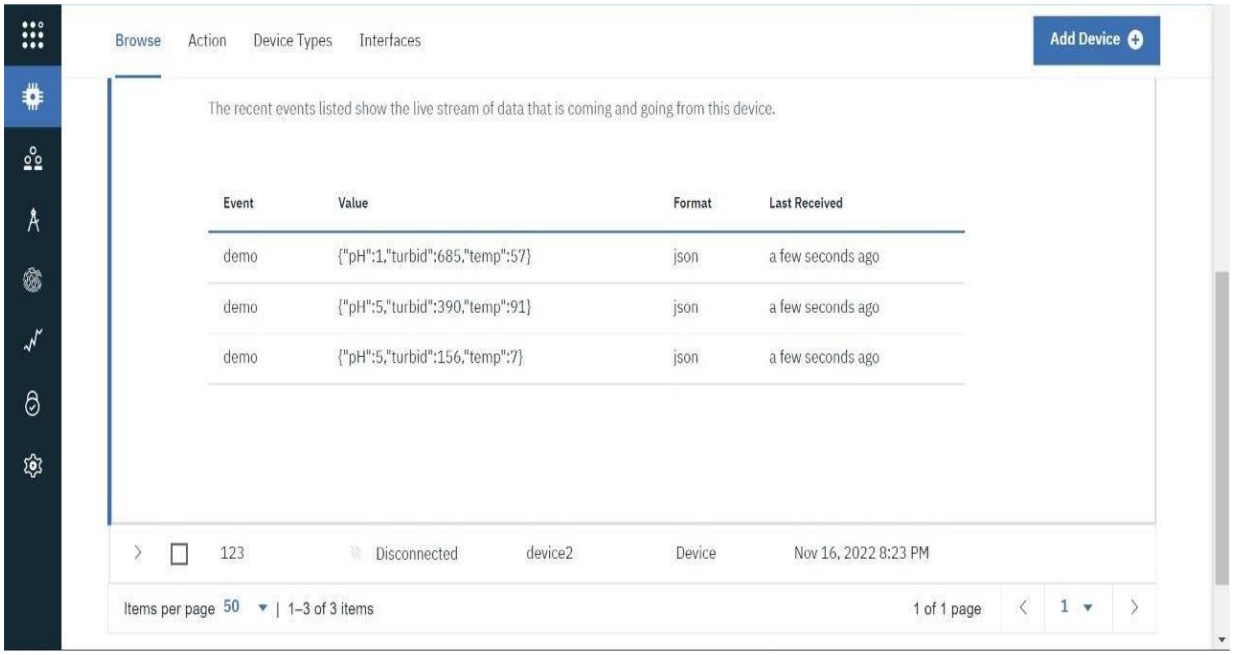
SPRINT 2

PYTHON CODE:

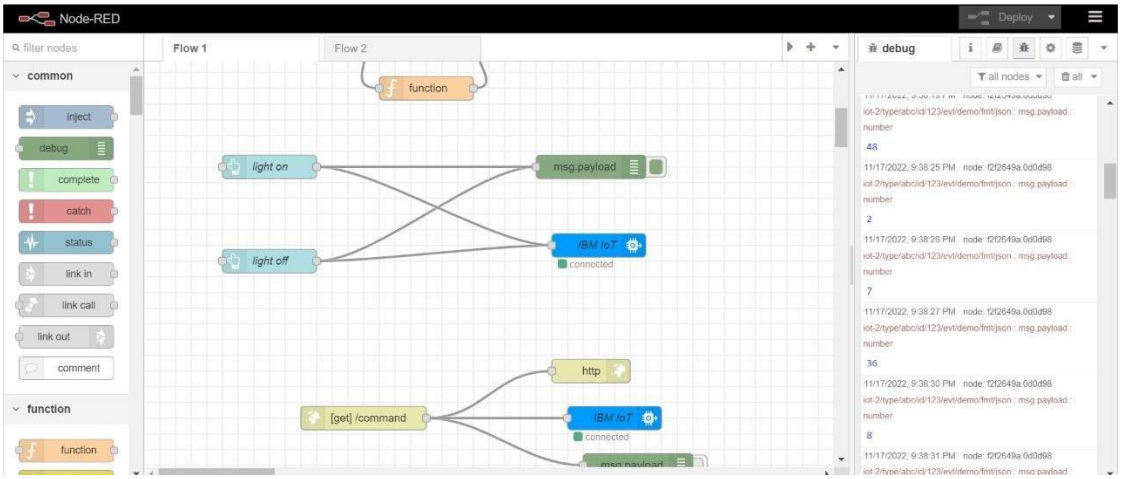
```
import ibmiotf.device
import ibmiotf.application
import time import
random import sys
from twilio.rest import Client
import keys
Client = Client(keys.account_sid, keys.auth_token)
organization = "15rapi"
deviceType = "abc" deviceId = "123" authMethod = "token"
authToken = "12345678"
pH = random.randint(1, 14)
turbidity = random.randint(1, 1000)
temperature = random.randint(0, 100)
def myCommandCallback(cmd):
print("Command Received: %s" % cmd.data['command'])
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()
deviceCli.connect()
while True:
pH = random.randint(1, 14)
turbidity = random.randint(1, 1000)
temperature = random.randint(0, 100)
data = {'pH': pH, 'turbid': turbidity, 'temp': temperature}
def SMS(): message = Client.messages.create(
```

```
body="ALERT!! THE WATER QUALITY IS DEGRADED",
from_=keys.twilio_number, to = keys.target_number) print(message.body)
if temperature>70 or pH<6 or turbidity>500:
SMS()
def myOnPublishCallback(): print("Published pH= %s" % pH, "Turbidity:%s"
% turbidity, "Temperature:%s" % temperature)
success = deviceCli.publishEvent("demo", "json", data, qos=0,
on_publish=myOnPublishCallback)
if not success: print("Not Connected to ibmiot") time.sleep(5)
deviceCli.commandCallback = myCommandCallback
deviceCli.disconnect()


#Twilio Account Credentials account_sid
='AC674a168fa92e54f2830009d96f9676dc'

auth_token ='a0127bca9a184493c92a4f6e5db2c91b'

twilio_number ='+15133275826' target_number ='+919361564622'
```

Data publish in IBM Watson Cloud

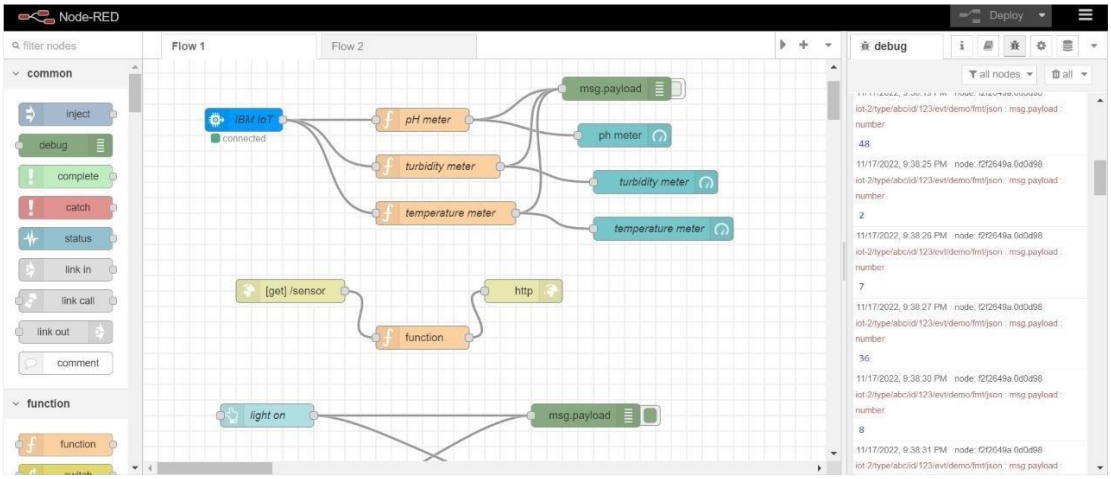


SPRINT 3 NODE RED ARCHITECTURE

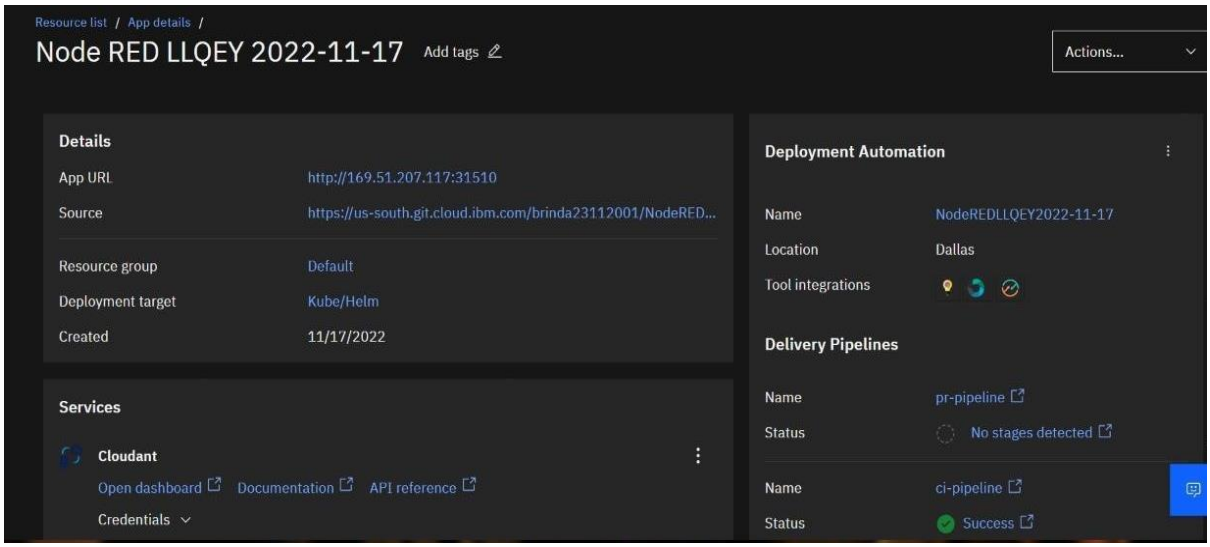


*

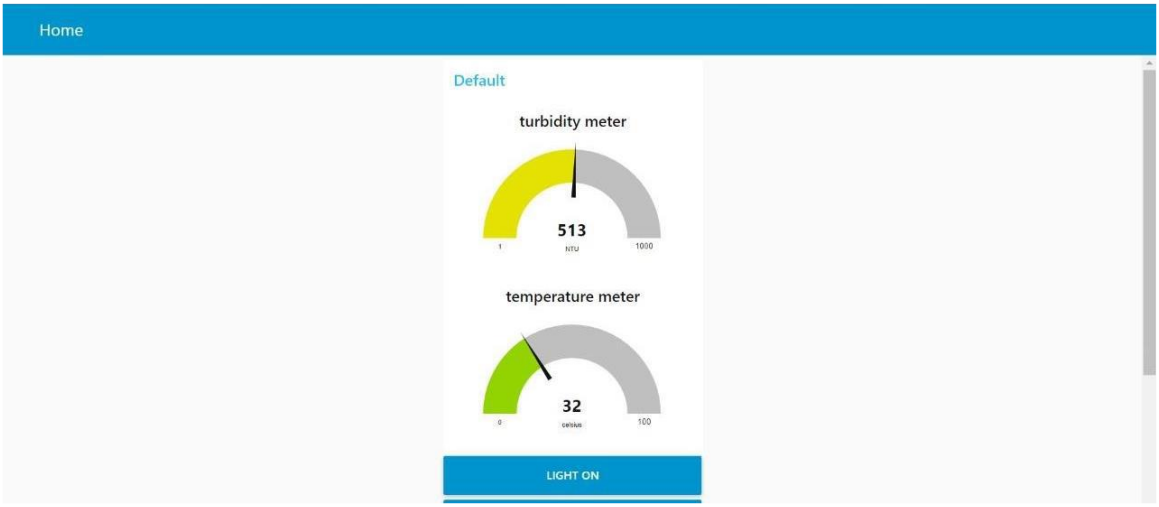
*

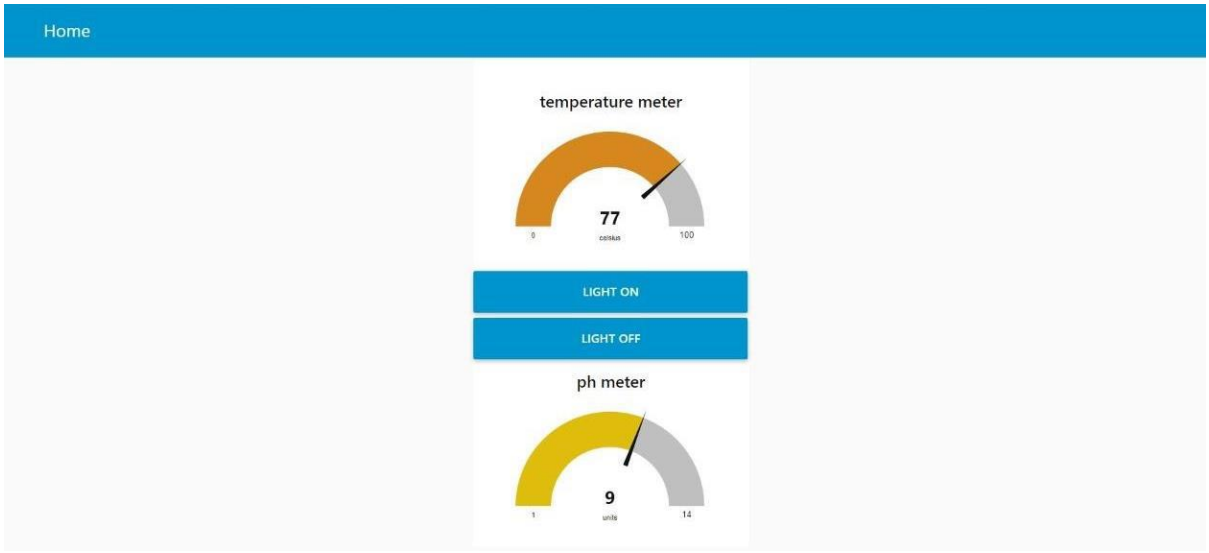


GENERATE A LINK TO INTERFACE THE NODE RED SERVICE WITH THE WEB UI/MOBILE APP



WEB UI FOR USER TO INTERACT WITH THE



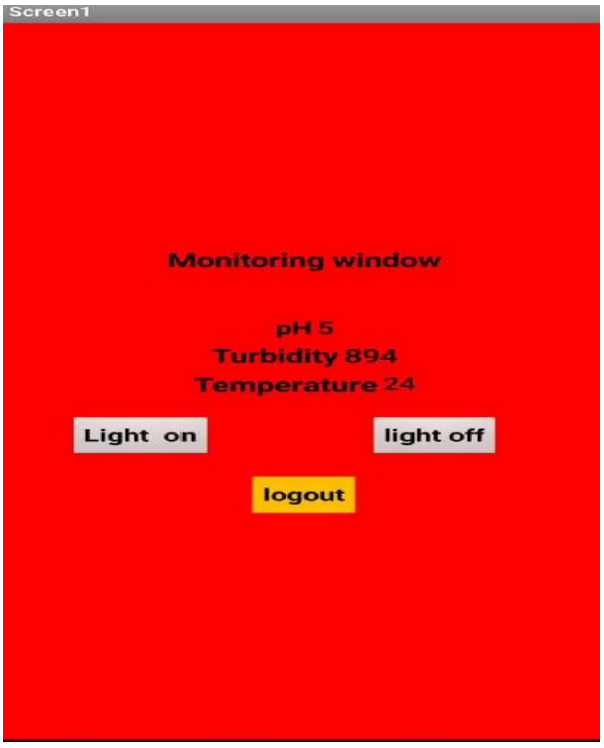


SOFTWARE:

SPRINT 4

Alert message sent to user:

RECEIVE DATA FROM CLOUD AND DISPLAY IN MOBILE APP:



MIT APP COMPONENTS BLOCK : LOGIC SECTION:

6.1 Sprint Planning & Estimation

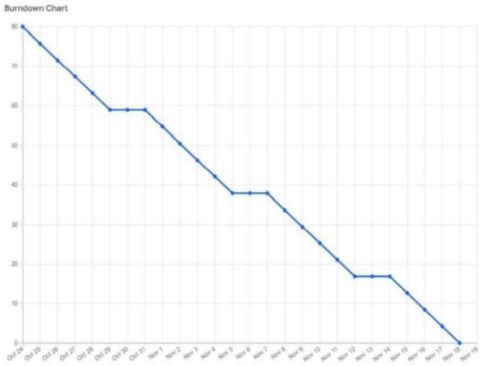
TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Gather/collect the relevant information on project use case, refer the existing solutions, technical papers, research publications etc.	03 October 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements	3 October 2022
Ideation	List the by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	13 October 2022
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	13 October 2022
Problem Solution Fit	Prepare problem - solution fit document.	13 October 2022
Solution Architecture	Prepare solution architecture document.	13 October 2022
Customer Journey Map	Prepare the customer journey maps to understand the user interactions & experiences with the application (entry to exit).	19 October 2022
Functional Requirement	Prepare the functional requirement document.	19 October 2022
Data Flow Diagrams	Draw the data flow diagrams and submit for review.	19 October 2022
Technology Architecture	Prepare the technology architecture diagram.	19 October 2022
Prepare Milestone & Activity List	Prepare the milestones & activity list of the project.	15 November 2022
Project Development - Delivery of Sprint-1, 2, 3 & 4	Develop & submit the developed code by testing it.	IN PROGRESS..

-

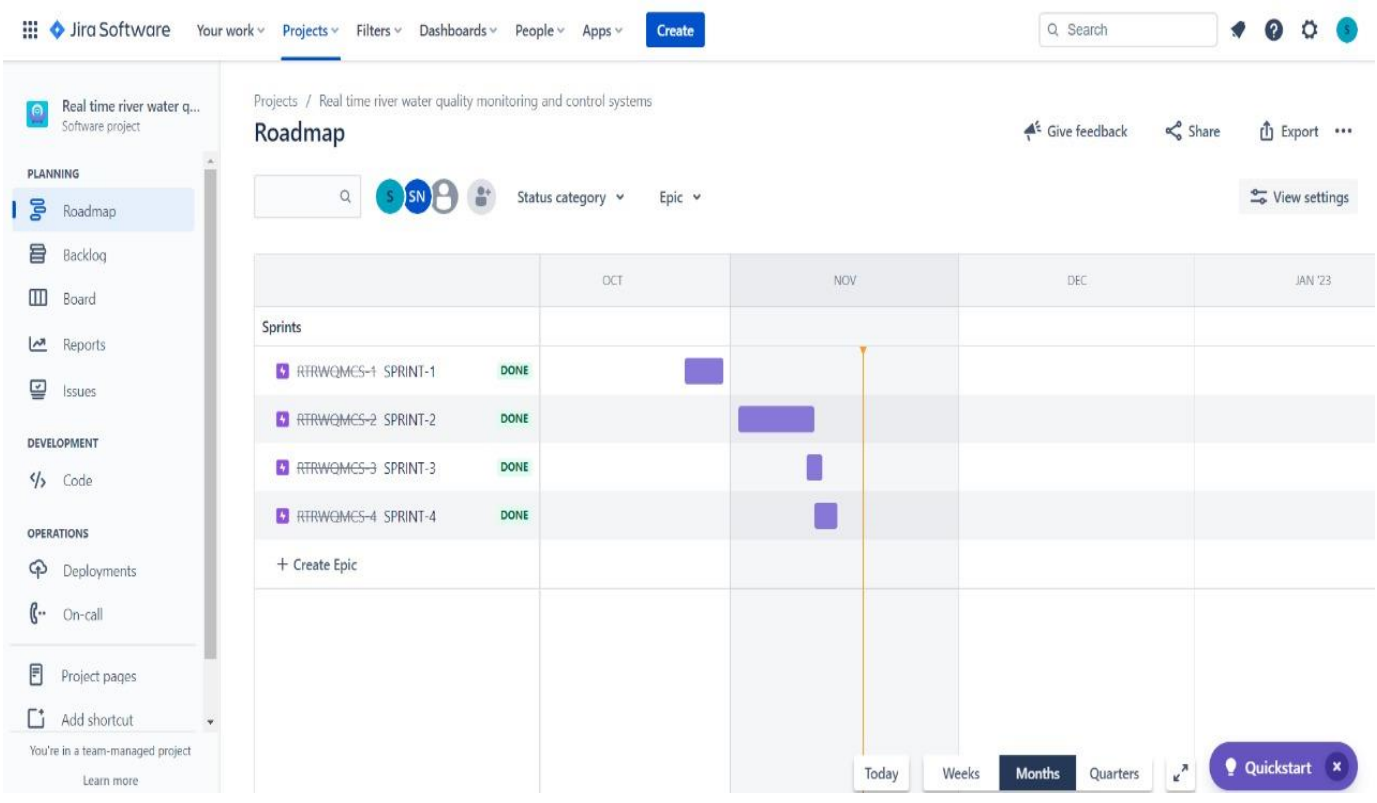
6.2 Sprint Delivery Schedule & Estimation

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	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

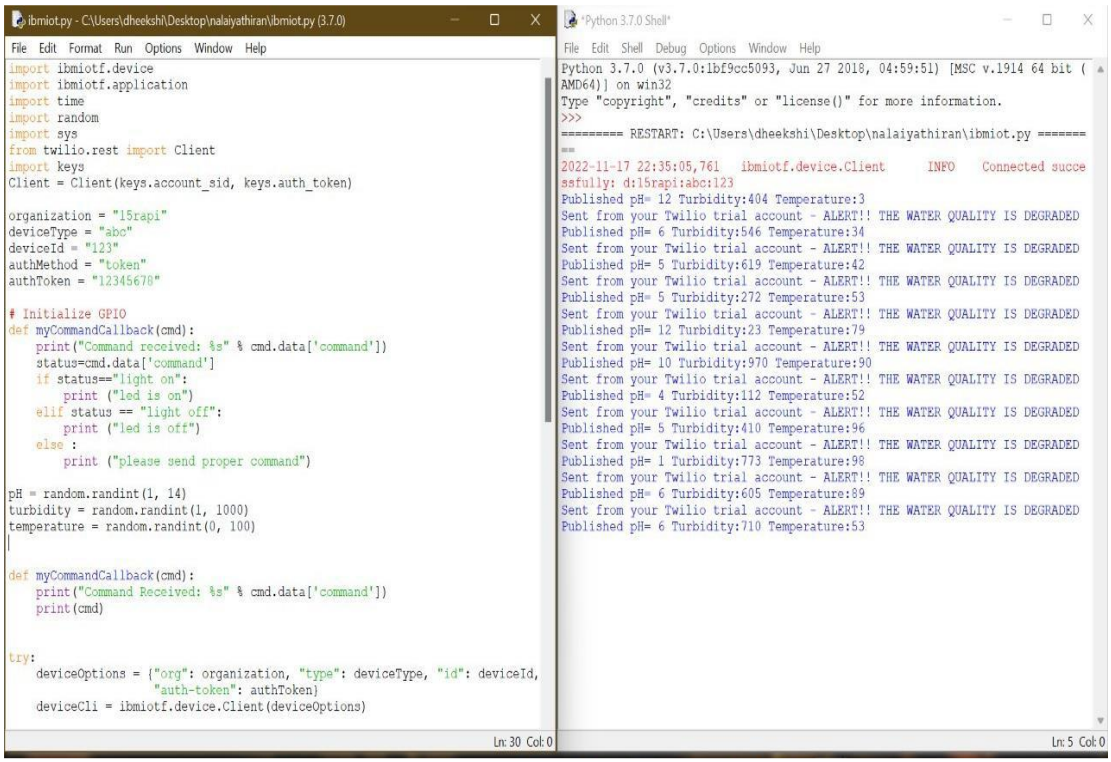
Burndown Chart:



6.3 Reports from JIRA



7. CODING & SOLUTION



ibmiot.py - C:\Users\dheekshi\Desktop\nalaiyathiran\ibmiot.py (3.7.0)

File Edit Format Run Options Window Help

```
import ibmiotf.device
import ibmiotf.application
import time
import random
import sys
from twilio.rest import Client
import keys
Client = Client(keys.account_sid, keys.auth_token)

organization = "15rapi"
deviceType = "abc"
deviceId = "123"
authMethod = "token"
authToken = "12345678"

# Initialize GPIO
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="light on":
        print ("led is on")
    elif status == "light off":
        print ("led is off")
    else :
        print ("please send proper command")

pH = random.randint(1, 14)
turbidity = random.randint(1, 1000)
temperature = random.randint(0, 100)


def myCommandCallback(cmd):
    print("Command Received: %s" % cmd.data['command'])
    print(cmd)


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

Ln: 30 Col: 0

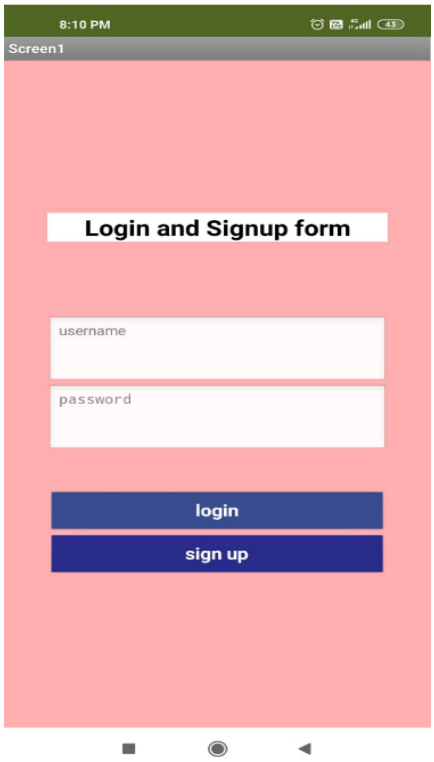
Python 3.7.0 Shell

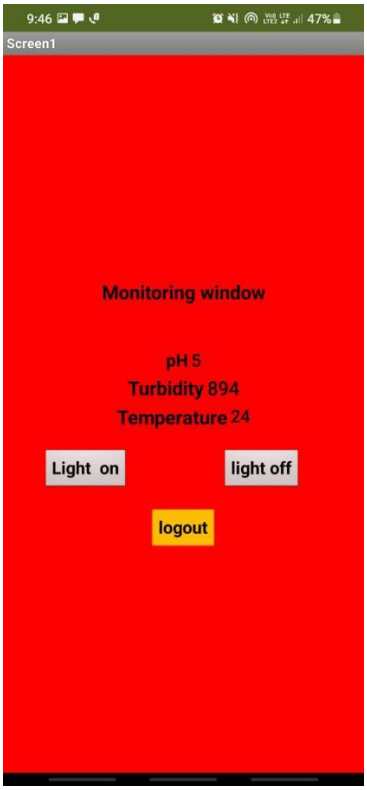
File Edit Shell Debug Options Window Help

```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (
AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
========= RESTART: C:\Users\dheekshi\Desktop\nalaiyathiran\ibmiot.py =======
==
2022-11-17 22:35:05,761   ibmiotf.device.Client      INFO    Connected succe
ssfully: d:15rapi:abc:123
Published pH= 12 Turbidity:404 Temperature:3
Sent from your Twilio trial account - ALERT!! THE WATER QUALITY IS DEGRADED
Published pH= 6 Turbidity:546 Temperature:34
Sent from your Twilio trial account - ALERT!! THE WATER QUALITY IS DEGRADED
Published pH= 5 Turbidity:619 Temperature:42
Sent from your Twilio trial account - ALERT!! THE WATER QUALITY IS DEGRADED
Published pH= 5 Turbidity:272 Temperature:53
Sent from your Twilio trial account - ALERT!! THE WATER QUALITY IS DEGRADED
Published pH= 12 Turbidity:23 Temperature:79
Sent from your Twilio trial account - ALERT!! THE WATER QUALITY IS DEGRADED
Published pH= 10 Turbidity:970 Temperature:90
Sent from your Twilio trial account - ALERT!! THE WATER QUALITY IS DEGRADED
Published pH= 4 Turbidity:112 Temperature:52
Sent from your Twilio trial account - ALERT!! THE WATER QUALITY IS DEGRADED
Published pH= 5 Turbidity:410 Temperature:96
Sent from your Twilio trial account - ALERT!! THE WATER QUALITY IS DEGRADED
Published pH= 1 Turbidity:773 Temperature:98
Sent from your Twilio trial account - ALERT!! THE WATER QUALITY IS DEGRADED
Published pH= 6 Turbidity:605 Temperature:89
Sent from your Twilio trial account - ALERT!! THE WATER QUALITY IS DEGRADED
Published pH= 6 Turbidity:710 Temperature:53
```

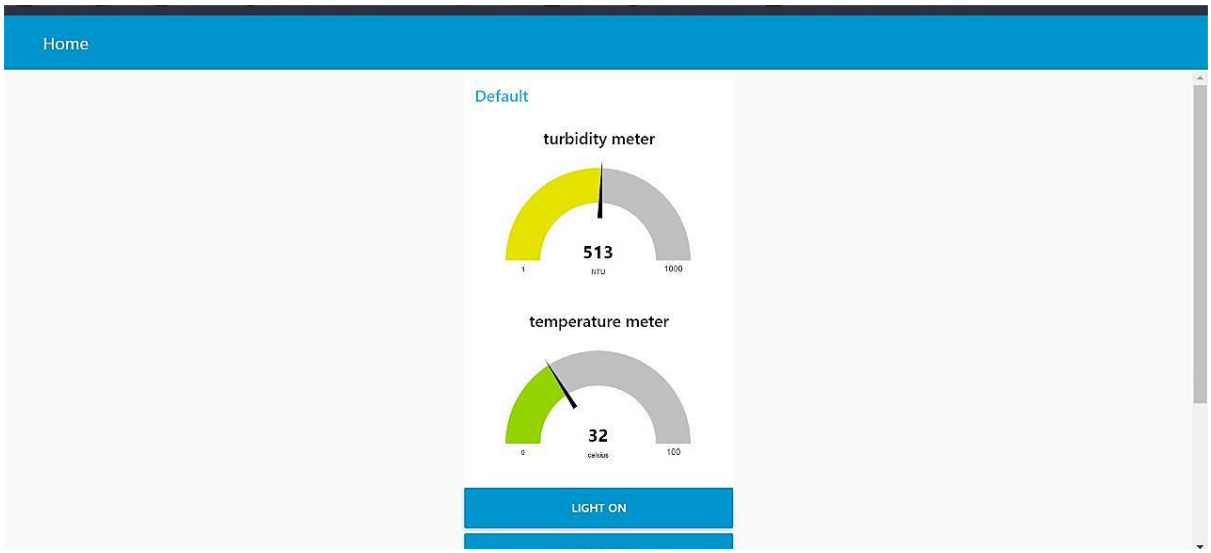
Ln: 5 Col: 0

LOGIN &MONITORING:

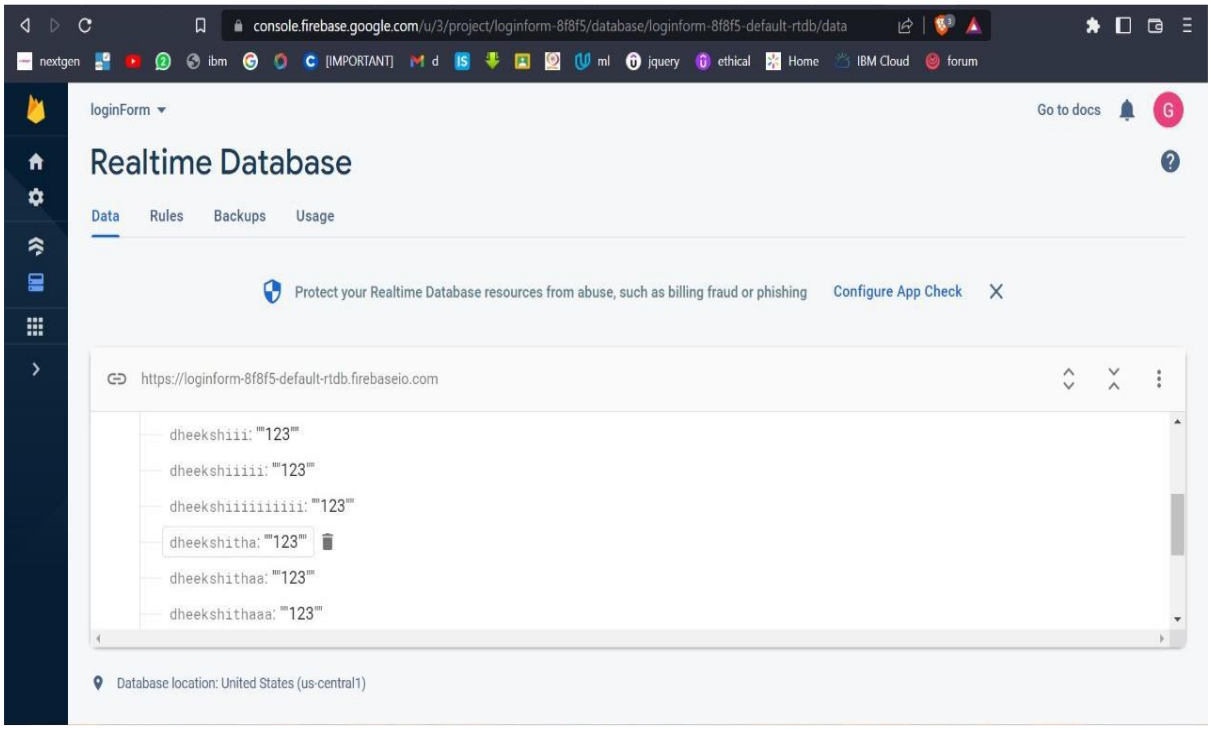




WEB UI



Database Schema (FireBase)



8. TESTING & RESULTS

- ### 8.1. Test Cases

Test case ID	Test case	Input	Output	Status
1	When the values of the parameters	Temperature=3,pH=3,Turbidity=404	ALERT!!THE WATER QUALITY IS DEGRADED	Pass
2	When the values of the parameters are within the range	Temperature=45,pH=7,Turbidity=350	(no alert message is sent)	Pass

8.2 User Acceptance Testing

Defect Analysis **Resolution**	**Severity 1**	**Severity 2**	**Severity 3**	**Severity 4**	**Subtotal**
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77

9.Test Case Analysis

Test Case Analysis **Section**	**Total Cases**	**Not Tested**	**Fail**	**Pass**
Client Application	51	0	0	51
Security	2	0	0	2
Exception Reporting	9	0	0	9
Final ReportOutput	4	0	0	4
Version Control	2	0	0	2

9. ADVANTAGE

Benefits or Advantages of IoT based Water Quality Monitoring System

Following are the benefits or advantages of IoT based Water Quality Monitoring System are as follows.

➤The boat is mobile in nature and hence large number of samples are easily collected from different locations in less time.

➤It is very easy to maintain the IoT based water quality monitoring system as all the electronic boards are available in the boat itself.

➤The system is very cheap as the hardware and software does not cost much.

➤Machine learning techniques have made it very easy to plot the data collected in various formats for proper analysis.

➤Cloud storage platforms such as adafruit, azure helps in storing the sensor data immediately and wirelessly to the robust servers.

- Based on a study of existing water quality monitoring system and scenario of water we can say that proposed system is more suitable to monitor water quality parameters in real time.

- The River Water Management and Alert System built
- on this architecture enable access, control and management of river water pollution.
- The water quality parameters such as pH, Temperature, Turbidity can be monitored

DISADVANTAGES

- • There are no specific management plans or sanctions on water extractions in many areas, such as pumping groundwater or rivers. These have caused less water to be

soluble and even led to the mining of that resource in some respects. This hampers the water levels and increases the risk of contaminated water.

- • This paper focuses only on the pH, turbidigity, temperature of river water here the other parameters such as conductivity is not considered.

- • The disadvantage of this system is, water is not monitoring seamlessly, and it always needs a human intervention.

CONCLUSION

An IoT system was developed to monitor river water in real time. The IoT system was used to collect the data from identified stations for different water quality parameters such as pH, turbidity, temperature and conductivity to generate a data set that was used to monitor the quality of water.
The collected data set can also be used in future to make the system intelligent by applying machine learning techniques.

Real-time monitoring of water quality by using IoT integrated will immensely help people to become conscious against using contaminated water as well as to stop polluting the water. The research is conducted focusing on monitoring river water quality in real-time. Therefore, IoT integrated big data analytics is appeared to be a better solution as reliability, scalability, speed, and persistence can be provided

11..FUTURE SCOPE

In this proposed system we only focus on measuring the quality of river water parameters. This project can be extended into an efficient water management system of a local area. Moreover, other parameters which wasn't the scope of this project such as total dissolved solid, chemical oxygen demand and dissolved oxygen can also be quantified.

12. APPENDIX

Source Code

```
import ibmiotf.device
import ibmiotf.application
import time
import random
import sys
from twilio.rest import Client
import keys

Client = Client(keys.account_sid, keys.auth_token)
organization = "15rapi"
deviceType = "abc"
deviceId = "123"
authMethod = "token"
authToken = "12345678"

pH = random.randint(1, 14)
turbidity = random.randint(1, 1000)
temperature = random.randint(0, 100)


def myCommandCallback(cmd):
print("Command Received: %s" % cmd.data['command'])
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()
deviceCli.connect()
while True:


pH = random.randint(1, 14)
turbidity = random.randint(1, 1000)
temperature = random.randint(0, 100)

data = {'pH': pH, 'turbid': turbidity, 'temp': temperature}
def SMS():

message = Client.messages.create(
body="ALERT!! THE WATER QUALITY IS
DEGRADED",
from_=keys.twilio_number,
to = keys.target_number)

print(message.body)
if temperature>70 or pH<6 or turbidity>500:
SMS()
def myOnPublishCallback():
print("Published pH= %s" % pH, "Turbidity:%s" %

turbidity, "Temperature:%s" % temperature)
success = deviceCli.publishEvent("demo", "json", data,
qos=0, on_publish=myOnPublishCallback)
if not success:
```

```
print("Not Connected to ibmiot")
time.sleep(5)
deviceCli.commandCallback = myCommandCallback
        deviceCli.disconnect()
```

GitHub Link:

https://github.com/IBM-EPBL/IBM-Project-30604-1660150336/tree/main/SANGAVI

Project Demo Link:

https://drive.google.com/file/d/1q334656k_WaOe8jgSFfbalS_d3cdF_jX/view?usp=sharing