REAL TIME RIVER WATER QUALITY MONITORING AND CONTROL SYSTEM

Category: INTERNET OF THINGS

A PROJECT REPORT

SUBMITTED BY

TEAM ID: PNT2022TMID29705

TEAM :

- *SANTHOSH KUMAR J [Team Leader]*
- *MOHAN RAJ D*
- *SURENDER D*
- *TEENA KUMARI M*

FROM

THANTHAI PERIYAR GOVERNMENT INSTITUTE OF TECHNOLOGY – VELLORE

In fulfillment of project in IBM-NALAIYATHIRAN 2022

INDEX

INTRODUCTION

INTRODUCTION

"A healthy life does not include polluted water in it"

Water is a key driver of economic and social development while it also has a basic function in maintaining the integrity of the natural environment. However, water is only one of a number of vital natural resources and it is imperative that water issues are not considered in isolation.

There are great differences in water availability from region to region - from the extremes of deserts to tropical forests. In addition, there is variability of supply through time as a result both of seasonal variation and inter-annual variation. All too often the magnitude of variability and the timing and duration of periods of high and low supply are not predictable; this equates to unreliability of the resource which poses great challenges to water managers in particular and to societies as a whole.

There is enough freshwater on the planet for seven billion people but it is distributed unevenly and too much of it is wasted, polluted and un-sustainably managed.

This document involves the use of IoT technology with the app to obtain the data of river water conditions and intimate the authorities if there is an alert. The data obtain in this app helps in knowing about the river state i.e., about its contamination, pH values and so on. By knowing this kind of information, one is able to know the availability of good water that is suitable for drinking.

The app frontend was designed using MIT App Inventor and backend is programmed using python script and Node RED. The user interface is designed in easier access manner.

Project Overview

Treatment plants are designed and operated to treat contaminants known to occur in water, comply with the drinking water standards. Unanticipated changes in quality or the presence of unusual contaminants in source water can adversely impact the ability of a utility to meet these objectives. Our app can improve a utility's ability to detect variations in water quality.

Our project involves the measurement of various water quality parameters in water. Basic water parameters like temperature, pH, its flow rate and ppm that indicates the contamination levels of the water.

IoT technology involves temperature sensor for temperature measurement, pH sensor for pH value, conductivity sensor for ppm measurement and flow sensor measures flow rate.

PURPOSE

Water quality refers to chemical, physical biological and radio logical characteristics of Water. It is a measure of the condition of water relative to the necessities of one or more biotic species and or to any human need or purposes. Water quality monitoring is defined as a Sampling and analysis of the water in lake, stream, ocean and river and conditions of the water Body. Smart water quality monitoring is a process of real-time monitoring and the analysis of Water to identify changes in parameters based on the physical, chemical and biological Characteristics. Monitoring water quality is clearly important: in our seas, our rivers, on the Surface and in our ports, for both companies and the public. It enables us to assess how they Are changing, analyse trends and to inform plans and strategies that improve water quality and Ensures that water meets its designated use. There are several indicators determining water Quality. These include dissolved oxygen, turbidity, bio indicators, nitrates, pH scale and water Temperature. Monitoring water quality helps to identify specific pollutants, a certain chemical, And the source of the pollution. There are many sources of water pollution: wastewater from Sewage seeping into the water supply; agricultural practices (e.g., the use of pesticides and Fertilizer); oil pollution, river and marine dumping, port, shipping and industrial activity. Monitoring water quality and a water quality assessment regularly provides a source of data Identifying trends, short and long-term, in water quality.

● Data collected over a period of time will show trends, for example identifying increasing concentrations of nitrogen pollution in a river or an inland waterway. The total data will then help to identify key water quality parameters.

● Environmental planning methods: water pollution prevention and management.

● Collecting, interpreting and using data is essential for the development of a sound and effective water quality strategy. The absence of real-time data will however hamper the development of strategies and limit the impact on pollution control. Using digital systems and programs for data collection and management is a solution to this challenge.

● Monitoring water quality is a global issue and concern: on land and at sea. Within the European Union, the European Green Deal sets out goals for restoring biological biodiversity and reducing water pollution, as well as publishing various directives to ensure standards of water quality. Individual nation states, for example France, have also clear regulatory frameworks requiring the effective monitoring of water quality. In the United States, the Environmental Protection Agency (EPA) enforces regulations to address water pollution in each state. Across the world, countries increasingly understand the importance of effective water quality monitoring parameters and methods.

LITERATURE SURVEY

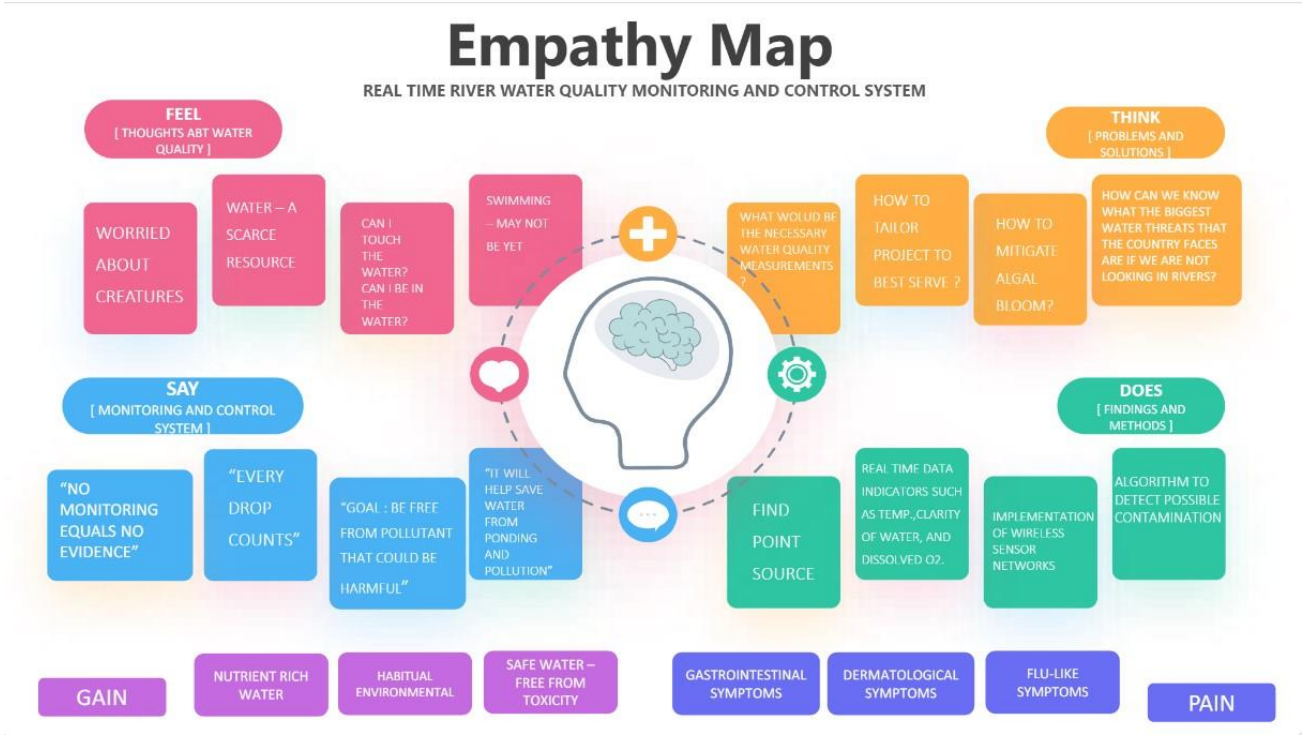
WORKED MODELS	METHODOLOGY	PUBLISHED YEAR	JOURNAL NAME
IOT Based Real-time River Water Quality Monitoring System	Real-time remote monitoring; Sensor technology; Telemetry	FEB 2004	Journal of Experimental Marine Biology and Ecology
Water Quality Monitoring Using Wireless Sensor Networks	Sensor networks; Sensors and actuators	January 2017	ACM Transactions on Sensor Networks
Using Synchronous Fluorescence Technique as a Water Quality Monitoring Tool	Water quality monitoring BOD Source discrimination.; Fluorescence measurement	January 2008	Springer
Real-time monitoring of water quality to identify pollution pathways	Diffuse and point sources, Mobile station, Online measurement Pollution Sensor, Wet chemical analyser	October 2018	Science of the Total Environment
Internet of things enabled real time water quality monitoring system	Water quality, Smart solution, Internet of things, Wi-Fi, Cloud storage	July 2017	Springer Nature
IoT Based Real-time River Water Quality Monitoring System	Sensors**;** Water quality monitoring**;** Internet of things; Big Data Analytics System	August 2019	Elsevier
A remote wireless system for water quality online monitoring in intensive Fish culture	web-server-embedded technology; mobile telecommunication technology; artificial neural network (ANNs)	October 2009	Elsevier
Design of Smart Sensors for Real-Time Water Quality Monitoring	Water quality monitoring, flow sensor, pH sensor, conductivity sensor, temperature sensor, ORP sensor, ZigBee, wireless sensor networks	July 2016	IEEE

Problem statement

Due to the fast-growing urbanization supply of safe drinking water is a challenge for 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.

IDEATION & PROPOSED SOLUTION

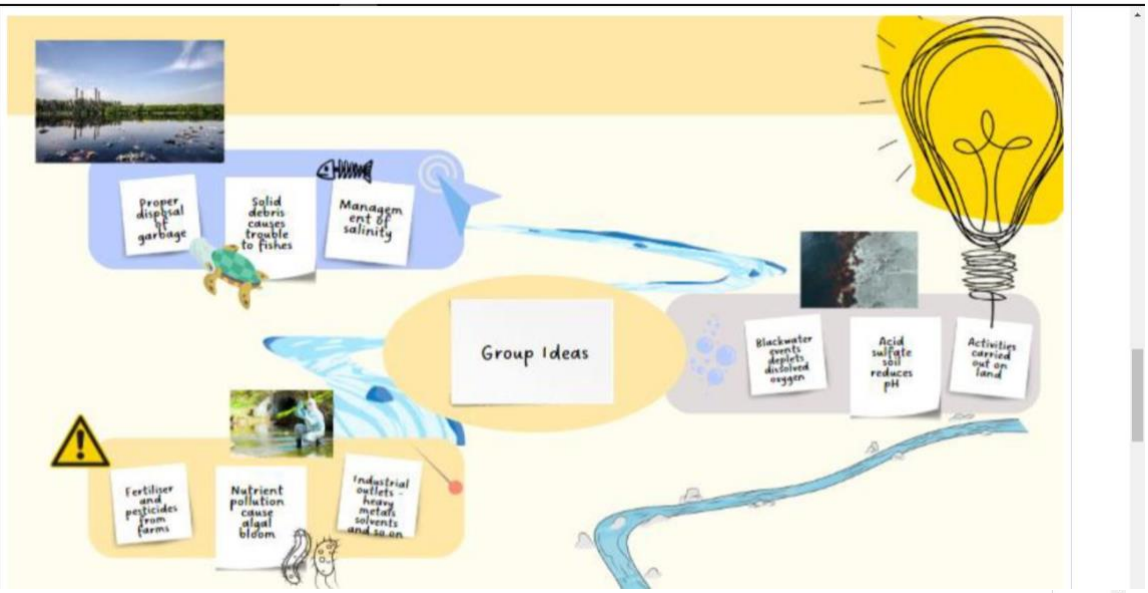
Empathy Map Canvas

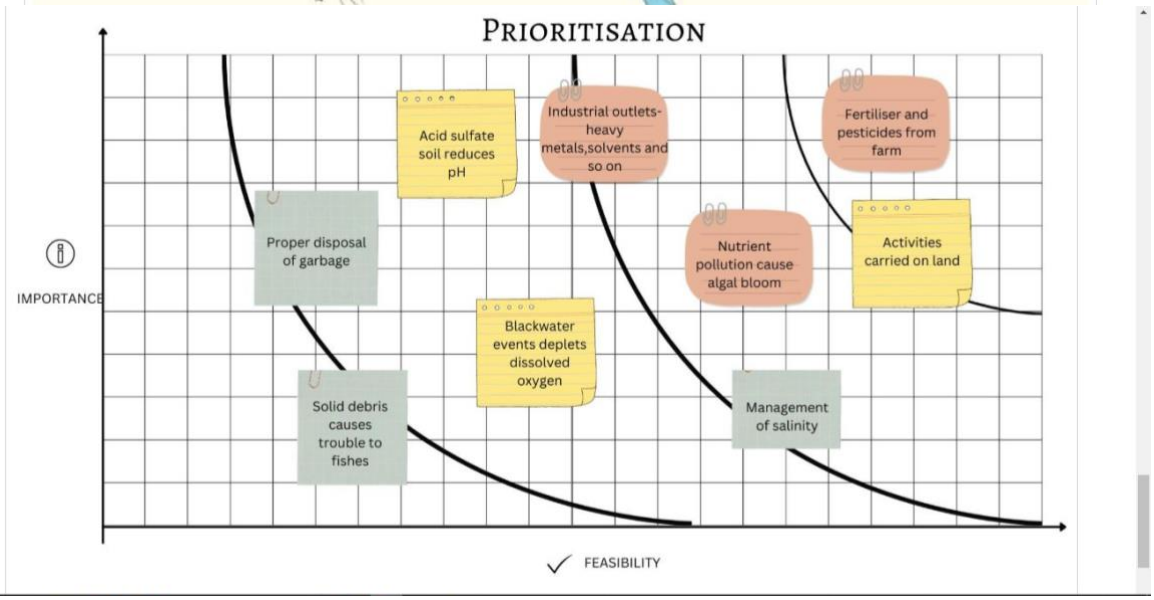


Ideation & Brainstorming

BRAINSTROM


SANTHOSH KUMAR J
Fertilizers and pesticides from farms.
Stormwater runoff may add oil,chemicals from surface
Aquatic life span reduces
Very high or very low pH makes the water toxic for things living in it
Proper disposal of garbage is required
Acid sulfate soil reduces pH
Flooding dumps waste into water
Amplified demand for water
TEENA KUMARI M
livestock operations wash nutrients
Industrial outlets contains thin metals,solvents and so on
Eutrophication creates dead zone
Aquatic life depend on suitable temperature
Pollutants from the drainage basins
Blackwater events deplets dissolved oxygen
Hardiness depends on calcium deposits
Bathe affects alkalinity
SURENDER D
Nutrient pollution cause algal blooms
Municipal wastewater contaminate waterways
Solid debris cause trouble to fishes
Too much ammonia is toxic to aquatic life
Atmospheric inputs from fossil fuel burning and bush fires
Management of salinity
Fertilisers contains nitrogen and phosphorus that cause bloom
Excessive usage of insecticides and pesticides
MOHANRAJ D
Animal waste also adds in water
Used water from households
Activities carried out on the land
E coli bacteria is a harmful microbe present in water
Lack of inadequate treatment of human wastes
Increase in emperature affect aquatic life
Poor sewage systems
Insufficient wastewater management

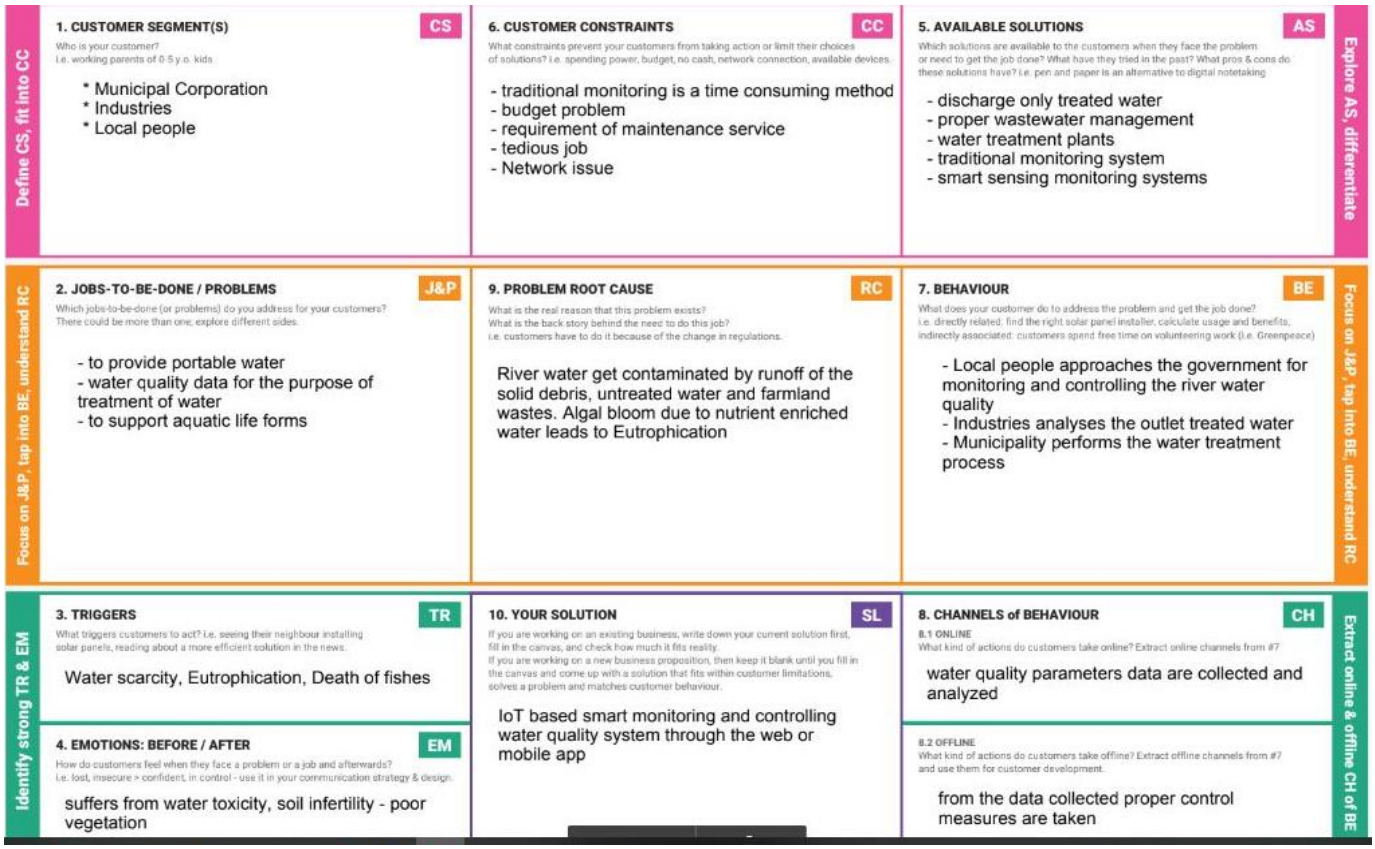

Proper disposal of garbage
Solid debris causes trouble to fishes
Managem ent of salinity
Group Ideas
Blackwater events deplets dissolved oxygen
Acid sulfate soil reduces pH
Activities carried out on land
Fertiliser and pesticides from farms
Nutrient pollution cause algal bloom
Industrial outlets - heavy metals solvents and so on


PRIORITISATION
Acid sulfate soil reduces pH
Industrial outlets- heavy metals,solvents and so on
Fertiliser and pesticides from farm
Proper disposal of garbage
Nutrient pollution cause algal bloom
Activities carried on land
Blackwater events deplets dissolved oxygen
Solid debris causes trouble to fishes
Management of salinity
IMPORTANCE
FEASIBILITY

Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Quality of river water is degraded by runoff from farmlands, livestock ,industries and nutrient rich environment leads algal bloom that results in Eutrophication
2.	Idea / Solution description	Measure the water parameters like Flow, Turbidity, Total dissolved solids(TDS),Temperature, Dissolved oxygen and pH level of water and check whether it is in specified standard levels.
3.	Novelty / Uniqueness	By using IoT Technology, a web based application monitors the quality of river water and alerts through notification if there is any threats.
4.	Social Impact / Customer Satisfaction	By monitoring and controlling the quality of water we can quench the thirsty throats and maintain aquatic culture
5.	Business Model (Revenue Model)	Water quality sensors from NKE Instrumentation, Pipe Scan water quality monitoring system and Underwater Wireless Sensor Network
6.	Scalability of the Solution	Real time continuous monitoring makes a better system for water quality monitoring and the water resources can be made safe by immediate response

Problem Solution Fit



REQUIREMENT ANALYSIS

Functional Requirement

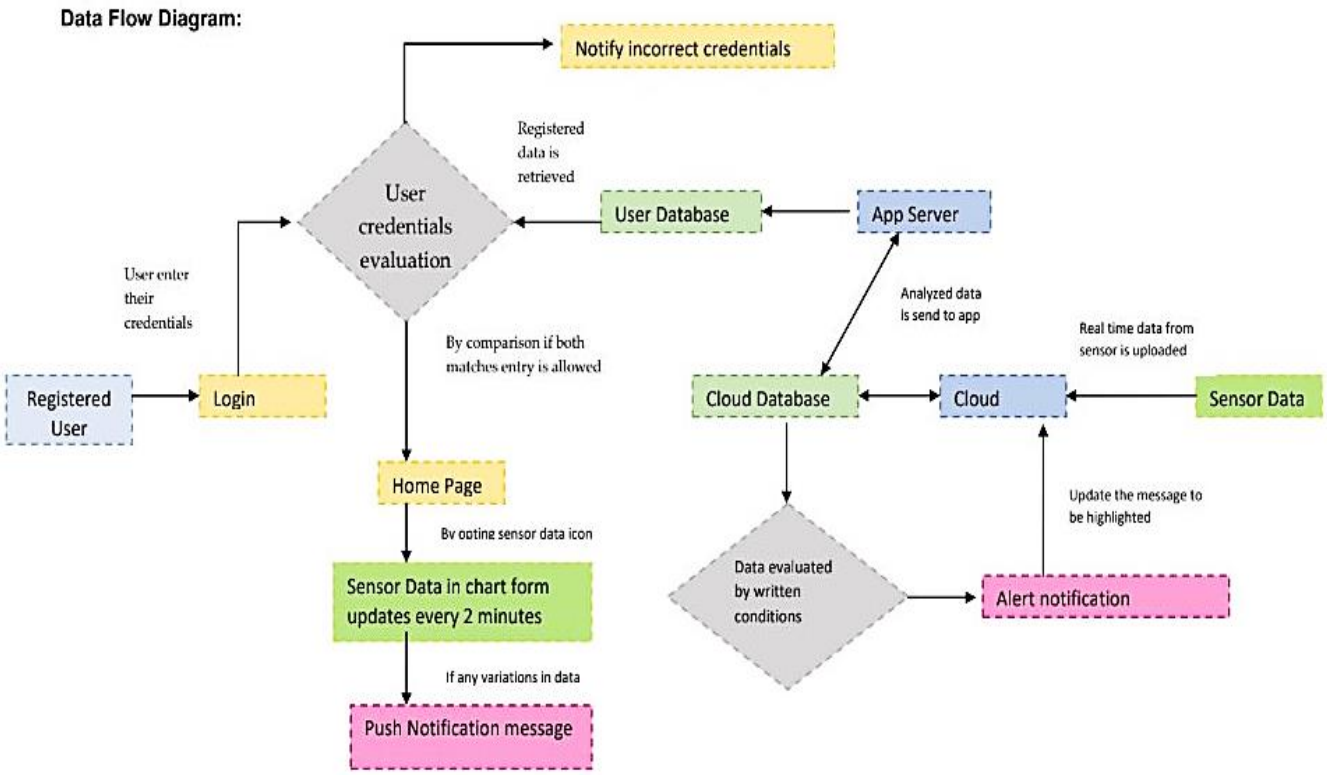
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Navigation	Forward and backward icons to move to wished screen
FR-4	Home Menu	Icon on top left corner with displays sub menus on click
FR-5	Analysis of Data	Display the gathered data in chart , gauge and histogram forms
FR-6	Push Notification	Alert message to be displayed for 1minute

Non-Functional Requirements

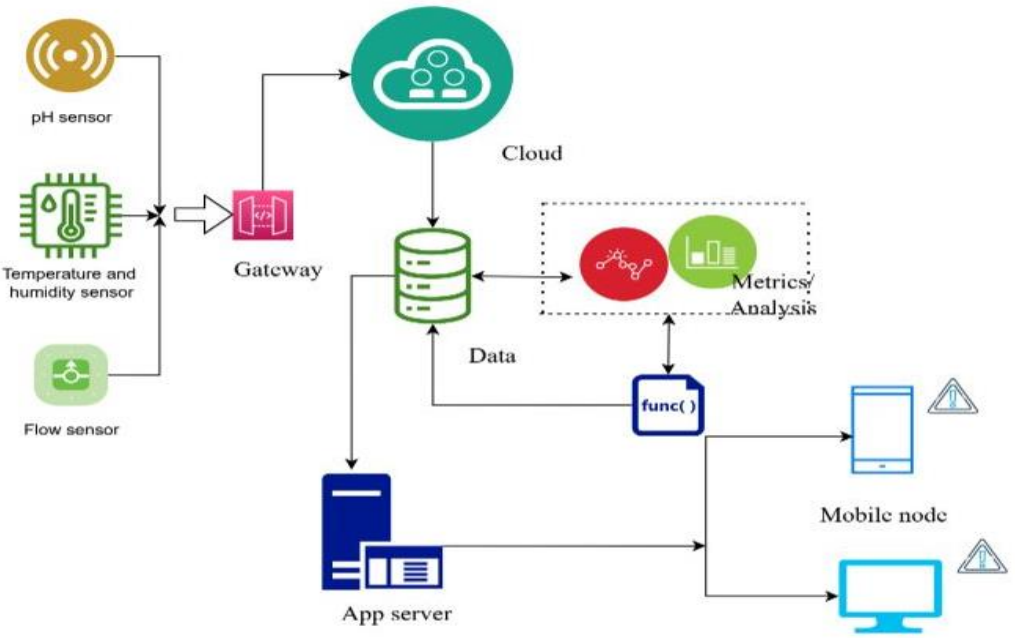
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Interface must be user friendly and ease to access tools
NFR-2	**Security**	Crashing of the app should rectified and notification to be send if non user logins
NFR-3	**Reliability**	When crash or slow network occurs the app should be able to restart and previous screen to be restored
NFR-4	**Performance**	Stable and continuous monitoring has to be done
NFR-5	**Availability**	Updates in the app must be ease and backup history to be made in somewhere
NFR-6	**Scalability**	System must be stable during undesirable conditions

PROJECT DESIGN

Data flow Diagrams



Solution &Technical Architecture



Technical Architecture:

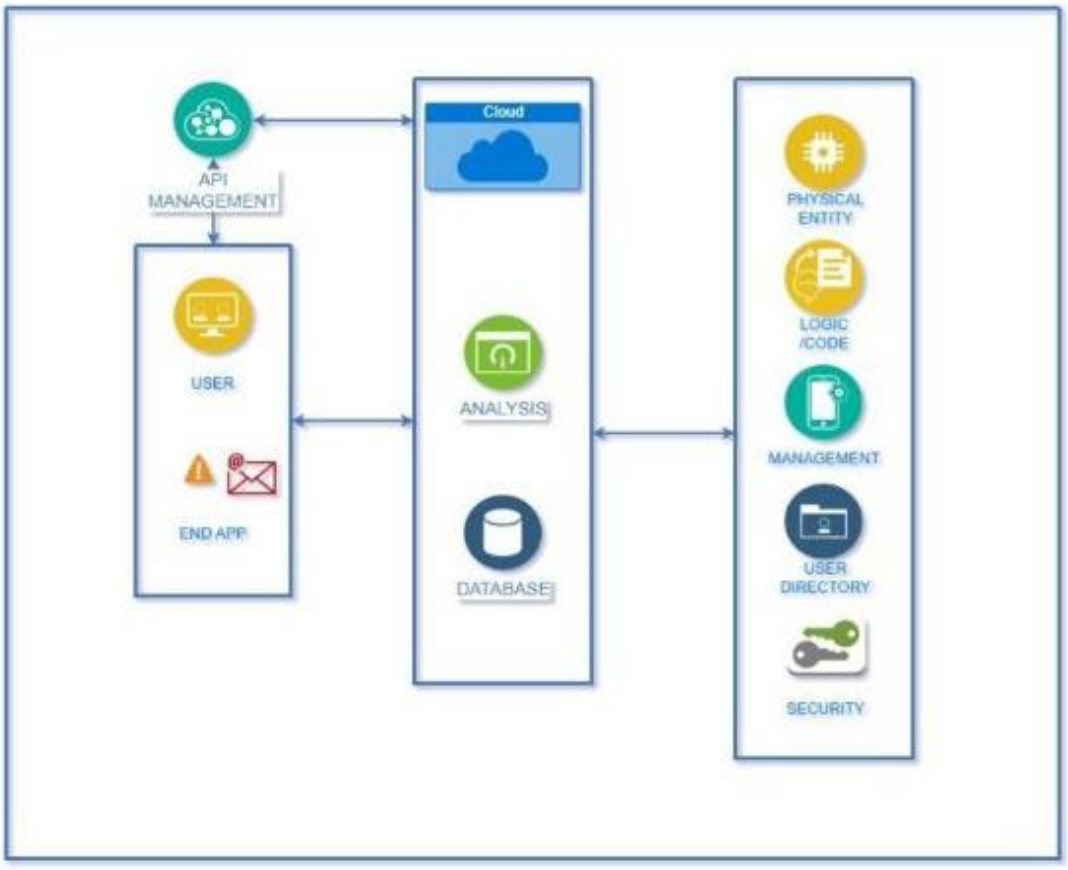


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS,MIT App Invetor
2.	Application Logic-1	Logic for a process in the application	Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	Purpose of External API used in the application	IBM Weather API, etc.
9.	External API-2	Purpose of External API used in the application	Device/Sensor API.
10.	Machine Learning Model	Purpose of Machine Learning Model	Object Detection model, Object Tracking model.
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Django, Pytorch
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	e.g. SHA-256, Encryptions, IAM Controls, OWASP,CAA, 6LoWPAN,Bluetooth
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	MQTT,LwM2M
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	VMware, IBM Cloud Load Balancer , Kubernetes
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	IBM Watson IoT Platform

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 my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can access the app by email account	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can login to the app	High	Sprint-1
		USN-6	As a user, I can reset the password if i have forgotten	I can reset the password by link sent to mail	High	Sprin-1
	Account	USN-4	As a user, I can logout of the app	I can easily logout by logout option	High	Sprint-1
	Dashboard	USN-1	As a user, I prefer an well organised page that is ease to access	I can easily view the contents	High	Sprint-1
Customer (Web user)	Registration	USN-6	As a user, I can register by entering my email, password, and confirming my password.	I can access my account / dashboard	Medium	Sprint-2
Customer Care Executive	Login issue	USN-2	As a user, I want to communicate to customer care service	I can contact through mail	High	Sprint-1
	App freeze	USN-7	As a user, I want to inform about the performance of the app	I can send feedback	Medium	Sprint -2
Administrator	Technical support	USN-8	As a admin, I want to rectify the issues in the app	I need to provide immediate solutions	High	Sprint-1
	Updates	USN-9	As a user, I want new features or upgraded workspace for easier interpretation	Admin need to make upgrades	High	Sprint-1
	Security	USN-10	As a admin, I want to secure the app from virus attack	I want to establish highly secure protocol	High	Sprint-1
	Maintenance	USN-11	As a user, I want backup and see history of data	I can access the past data from cloud storage	High	Sprint-1

PROJECT PLANNING & SCHEDULING

Sprint Planning & Estimation

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	Mohanraj. D
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Mohanraj. D
Sprint-1		USN-3	Login through the website	2	Low	Mohanraj. D
Sprint-1		USN-4	As a user, I can register for the application through Gmail	2	Medium	Mohanraj .D
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	Teena Kumari. M
Sprint-1	Dashboard	USN-6	As a user home icon is available and can access menu	1	High	Santhosh kumar. J
Sprint-1	Option icon	USN-7	Display Data and chart	2	High	Surender. D
Sprint-2	Node red	USN-8	Connection establishment	1	High	Santhosh kumar. J
Sprint-2	Event creation	USN-9	Program each sensor	2	High	Santhosh kumar. J
Sprint-2	Testing	USN-9	Observe the data	2	High	Mohanraj .D

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-3	Circuit wokwi	USN-10	Build circuit in node red	2	High	Mohanraj .D
Sprint-3	Testing	USN-11	Code testing	2	High	Mohanraj .D
Sprint-4	MIT app code	USN-12	Backend coding	2	High	Surender. D
Sprint-4		USN-13	Testing code	2	High	Teena Kumari. M
Sprint-4		USN-14	Testing overall app	2	High	Teena Kumari. M

Velocity:

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

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

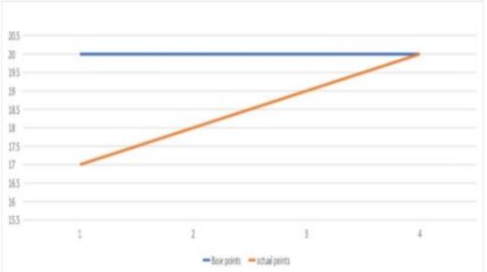
velocity (AV) per iteration unit (story points per day)

AV=Sprint duration/ velocity =20/4=5

Table:

scrumdown chart

	sprint1	sprint 2	sprint3	sprint4
Days	0	4	8	12
Base points	20	20	20	20
actual points	20	17	12	10



Sprint Delivery Schedule

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	4 Days	01 Nov 2022	04 Nov 2022	20	06 Nov 2022
Sprint-2	20	4 Days	04 Nov 2022	08 Nov 2022	20	11 Nov 2022
Sprint-3	20	4 Days	08 Nov 2022	12 Nov 2022	20	14 Nov 2022
Sprint-4	20	4 Days	12 Nov 2022	19 Nov 2022	20	19 Nov 2022

CODING AND SOLUTIONING

Python Code:

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

#Provide your IBM Watson Device Credentials
organization = "khwc4s"
deviceType = "check"
deviceId = "2468"
authMethod = "token"
authToken = "09876543"

# Initialize GPIO
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="lighton":
        print ("led is on")
    elif status == "lightoff":
        print ("led 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 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)
        ppm=random.randint(0,1000)
        flow=random.randint(0,200)

        data = { 'temp' : temp, 'ph': ph,'ppm':ppm,'flow':flow }
        #print data
        def myOnPublishCallback():
            print ("Published Temp = %s C" % temp, "ph = %s " % ph,"ppm = %s ppm" % ppm,"flow =
%s l/m" % flow, "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

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

TESTING:

- Testing is the most important part of the software development process.

Some of the reasons for its importance are as follows:

● Testing helps find and fix the bugs in the software which prevent the program from performing as required or as efficiently as needed.

● Bug fixing in the early development stages helps to save a lot of time and effort.

● Testing is very essential to make sure that the final output product will work well without any errors once deployed.Testing improves the quality of the software.

● Validation is the process of ensuring that the software built is in accordance with the expected business requirements.

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [Real time River water quality monitoring and control system] project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

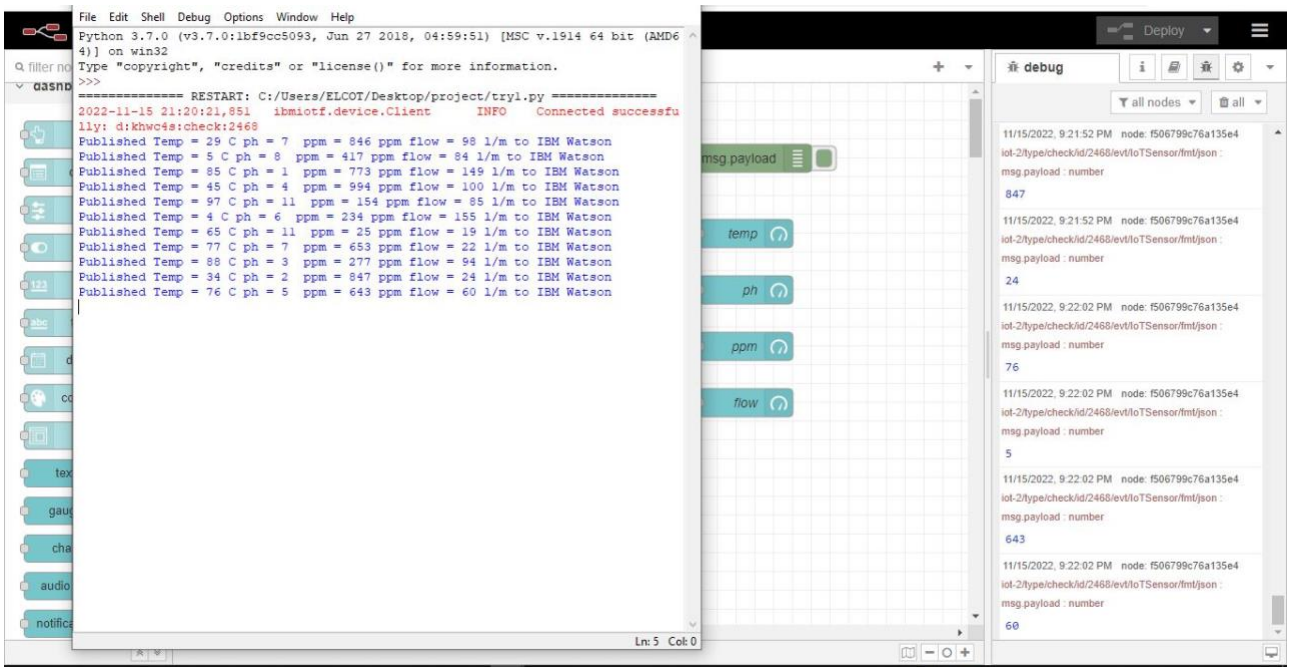
Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	8	4	2	3	17
Duplicate	1	0	4	0	5
External	2	3	0	1	6
Fixed	10	2	4	16	32
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	2	2	1	5
Totals	21	11	14	22	68

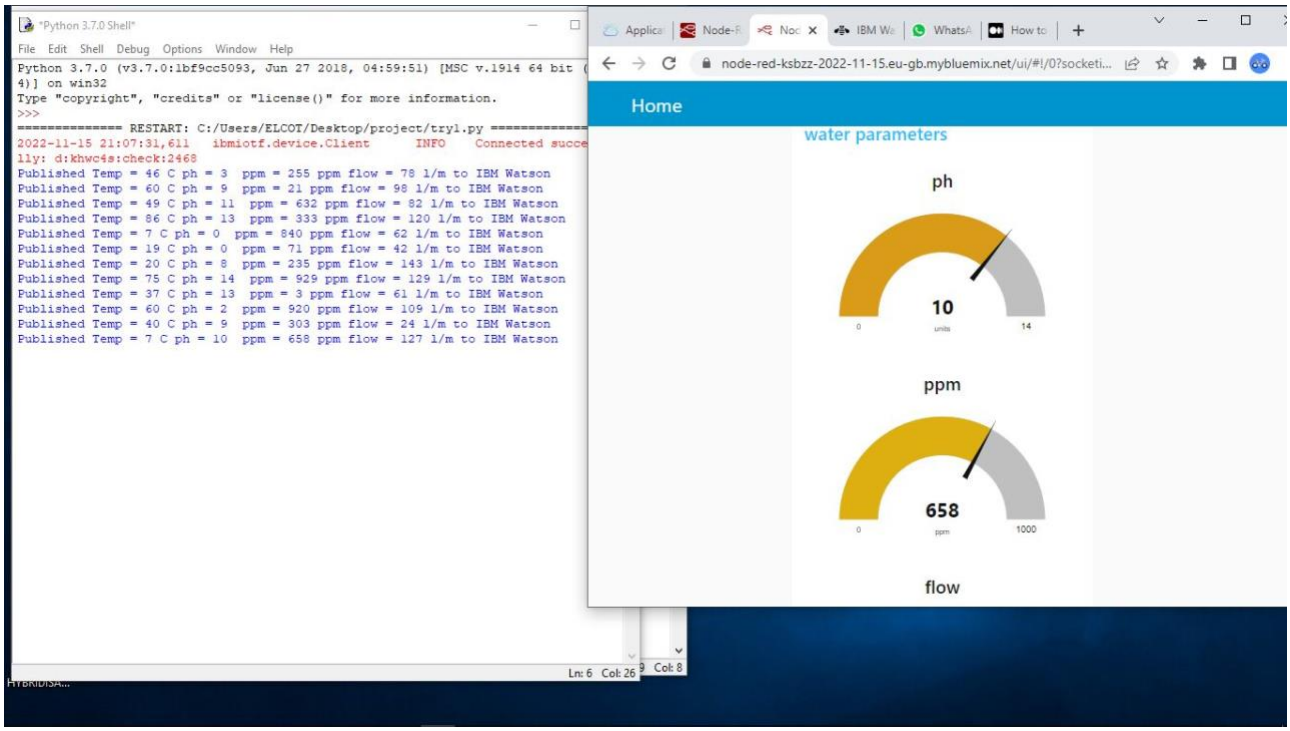
3. Test Case Analysis

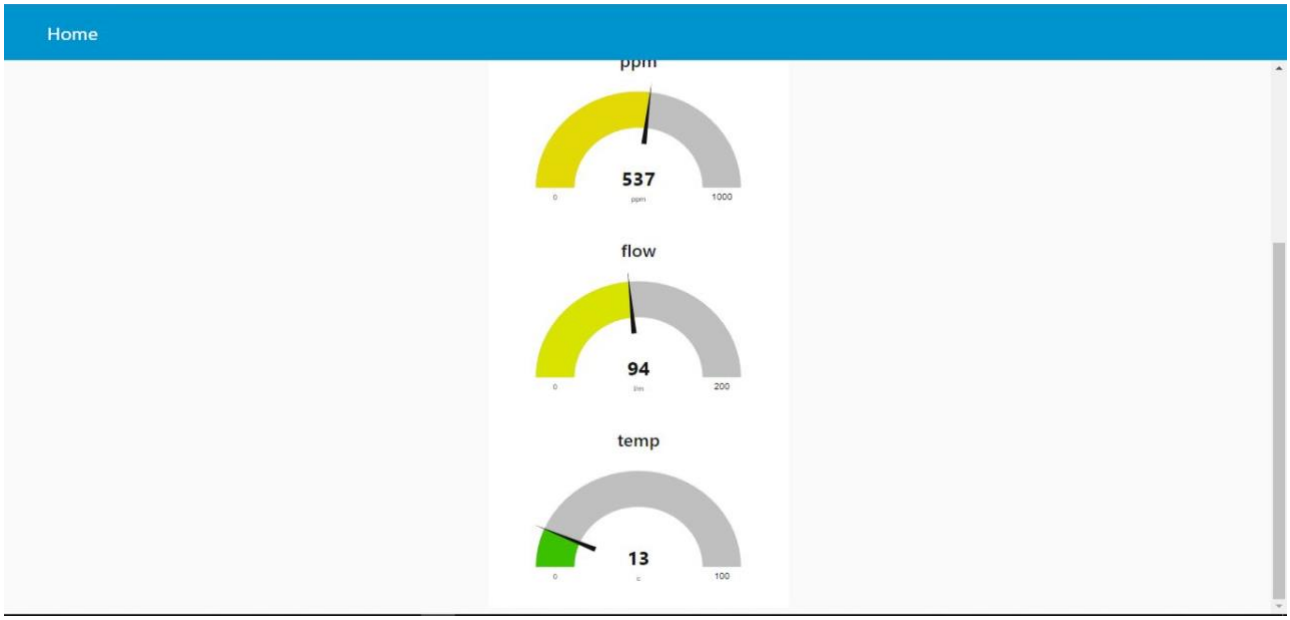
This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	2	0	0	1
Client Application	5	0	0	3
Security	2	0	0	2
Exception Reporting	4	0	0	3
Final Report Output	2	0	0	2
Version Control	2	0	0	2

RESULT


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/ELCOT/Desktop/project/try1.py ==========
2022-11-15 21:20:21,851 ibmiotf.device.Client INFO Connected successfully: d:khwc4s:check:2468
Published Temp = 29 C ph = 7 ppm = 846 ppm flow = 98 l/m to IBM Watson
Published Temp = 5 C ph = 8 ppm = 417 ppm flow = 84 l/m to IBM Watson
Published Temp = 85 C ph = 1 ppm = 773 ppm flow = 149 l/m to IBM Watson
Published Temp = 45 C ph = 4 ppm = 994 ppm flow = 100 l/m to IBM Watson
Published Temp = 97 C ph = 11 ppm = 154 ppm flow = 85 l/m to IBM Watson
Published Temp = 4 C ph = 6 ppm = 234 ppm flow = 155 l/m to IBM Watson
Published Temp = 65 C ph = 11 ppm = 25 ppm flow = 19 l/m to IBM Watson
Published Temp = 77 C ph = 7 ppm = 653 ppm flow = 22 l/m to IBM Watson
Published Temp = 88 C ph = 3 ppm = 277 ppm flow = 94 l/m to IBM Watson
Published Temp = 34 C ph = 2 ppm = 847 ppm flow = 24 l/m to IBM Watson
Published Temp = 76 C ph = 5 ppm = 643 ppm flow = 60 l/m to IBM Watson
Ln: 5 Col: 0
Deploy
debug
all nodes
all
msg payload
temp
ph
ppm
flow
11/15/2022, 9:21:52 PM node: f506799c76a135e4
iot-2/type/check/id/2468/evt/IoTSensor/fmt/json : msg.payload : number
847
11/15/2022, 9:21:52 PM node: f506799c76a135e4
iot-2/type/check/id/2468/evt/IoTSensor/fmt/json : msg.payload : number
24
11/15/2022, 9:22:02 PM node: f506799c76a135e4
iot-2/type/check/id/2468/evt/IoTSensor/fmt/json : msg.payload : number
76
11/15/2022, 9:22:02 PM node: f506799c76a135e4
iot-2/type/check/id/2468/evt/IoTSensor/fmt/json : msg.payload : number
5
11/15/2022, 9:22:02 PM node: f506799c76a135e4
iot-2/type/check/id/2468/evt/IoTSensor/fmt/json : msg.payload : number
643
11/15/2022, 9:22:02 PM node: f506799c76a135e4
iot-2/type/check/id/2468/evt/IoTSensor/fmt/json : msg.payload : number
60


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/ELCOT/Desktop/project/try1.py ==========
2022-11-15 21:07:31,611 ibmiotf.device.Client INFO Connected successfully: d:khwc4s:check:2468
Published Temp = 46 C ph = 3 ppm = 255 ppm flow = 78 l/m to IBM Watson
Published Temp = 60 C ph = 9 ppm = 21 ppm flow = 98 l/m to IBM Watson
Published Temp = 49 C ph = 11 ppm = 632 ppm flow = 82 l/m to IBM Watson
Published Temp = 86 C ph = 13 ppm = 333 ppm flow = 120 l/m to IBM Watson
Published Temp = 7 C ph = 0 ppm = 840 ppm flow = 62 l/m to IBM Watson
Published Temp = 19 C ph = 0 ppm = 71 ppm flow = 42 l/m to IBM Watson
Published Temp = 20 C ph = 8 ppm = 235 ppm flow = 143 l/m to IBM Watson
Published Temp = 75 C ph = 14 ppm = 929 ppm flow = 129 l/m to IBM Watson
Published Temp = 37 C ph = 13 ppm = 3 ppm flow = 61 l/m to IBM Watson
Published Temp = 60 C ph = 2 ppm = 920 ppm flow = 109 l/m to IBM Watson
Published Temp = 40 C ph = 9 ppm = 303 ppm flow = 24 l/m to IBM Watson
Published Temp = 7 C ph = 10 ppm = 658 ppm flow = 127 l/m to IBM Watson
Ln: 6 Col: 26
node-red-ksbzz-2022-11-15.eu-gb.mybluemix.net/ui/#!/0?socketi...
Home
water parameters
ph
10
ppm
658
1000
flow



Performance metrics

Epoch	Accuracy	Loss	Val-loss	Val Accuracy
1	0.8612	0.5750	0.9289	0.2057
2	0.9151	0.2386	0.9151	0.1584
3	0.9461	0.1595	0.9316	0.1794
4	0.9526	0.1530	0.9026	0.2408
5	0.9480	0.1525	0.9237	0.1976
6	0.9342	0.1873	0.9289	0.1569
7	0.9507	0.1657	0.9421	0.1541
8	0.9533	0.1549	0.9316	0.1917
9	0.9513	0.1451	0.9368	0.1848
10	0.9520	0.1589	0.1496	0.9447

ADVANTAGES & DISADVANTAGES

Advantages

Identifying the health of your water will help you to discover where it may need some help. Ultimately, finding a source of pollution, or remaining proactive with your monitoring will enable you to save money in the long term.
The more information that you can obtain will assist you with your decision on what product you may need to improve the condition of your water. Simply guessing and buying products based on a hunch or a general trend is ill-advised, as each body of water has unique properties that can only be discovered through testing.

Measuring the amount of dissolved oxygen in your water is another important advantage of water quality testing, as typically the less oxygen, the higher the water temperature, resulting in a more harmful environment for aquatic life.
These levels do fluctuate slightly across the seasons, but regular monitoring of your water quality will allow you to discover trends over time, and whether there are other factors that may be contributing to the results you discover.

Disadvantages

The system is less effective as sensors are installed very deep inside the water and their positions are fixed.

The sensors are very expensive. Moreover, their maintenance cost is also very high.

The sensors which work on power source may often required to be replaced in case of malfunctioning.
Mounted Sensors may get damage during natural disasters and often by aquatic animals.

CONCLUSION

Real-time monitoring of water quality by using IoT 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 is appeared to be a better solution as reliability, scalability, speed, and persistence can be provided.

FUTURE SCOPE

The system has good flexibility. Only by replacing the corresponding sensors and changing the relevant software programs, this system can be used to monitor other water quality parameters. The operation is simple. The system can be expanded to monitor hydrologic, air pollution, industrial and agricultural production and so on. It has widespread application and extension value.

APPENDIX Source Code:

PROGRAM:

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

#Provide your IBM Watson Device Credentials
organization = "khwc4s"
deviceType = "check"
deviceId = "2468"
authMethod = "token"
authToken = "09876543"

# Initialize GPIO
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="lighton":
        print ("led is on")
    elif status == "lightoff":
        print ("led 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 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)
        ppm=random.randint(0,1000)
        flow=random.randint(0,200)

        data = { 'temp' : temp, 'ph': ph,'ppm':ppm,'flow':flow }
        #print data
        def myOnPublishCallback():
            print ("Published Temp = %s C" % temp, "ph = %s " % ph,"ppm = %s ppm" % ppm,"flow =
%s l/m" % flow, "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

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

GITHUB LINK: https://github.com/IBM-EPBL/IBM-Project-19778-1659706367

https://drive.google.com/file/d/1pmpR30xteVwbb5HcXnvy1vq4Up1BZXZu/view?usp=drivesdk