IOT ENABLED REAL-TIME RIVER WATER QUALITY AND MONITORING SYSTEM

NALAIYATHIRAN PROJECT BASED LEARNING

ON

PROFESSIONAL READINESS FOR INNOVATION EMPLOYABILITY AND ENTREPRENEURSHIP

Project Report Submitted By

820619104002	**Abarna.S**
820619104015	**Harini.K**
820619104040	**Priyadharshini.B**
820619104043	**Priyadharsini.M.R**
820619104046	**Rajashree.S**

Bachelor of Engineering

in

Computer Science and Engineering

Arasu Engineering College : Kumbakonam-612 501

Anna University: Chennai 600 025
NOVEMBER-2022

CONTENTS

1.INTRODUCTION

1.1 PROJECT OVERVIEW

The environment around consists of five key elements e.g., soil, water, climate, natural vegetation, and landforms.Among these water is the utmost crucial element for human life. It is also vital for the persistence of other living habitats.Whether it is used for drinking, domestic use, and food production or recreational purposes, safe and readily available water is the need for public health. So it is highly imperative for us to maintain water quality balance. Data collected at the apart site can be displayed in a visual format on a server PC with the help of Spark streaming analysis through Spark MLlib, Deep learning neural network models, Belief Rule Based (BRB) system and is also compared with standard values. If the acquired value is above the threshold value automated warning SMS alert will be sent to the agent. The uniqueness of our proposed paper is to obtain the water monitoring system with high frequency, high mobility, and low powered.

1.2 PURPOSE

To develop an efficient, cost-effective, real-time water quality monitoring system which will integrate wireless sensor network and internet of things . In this research, we monitor the physical and chemical parameters of water bodies by using an IoT based sensor network.Therefore, our proposed system will immensely helps to become conscious against contaminated water as well as to stop polluting the water.Using different sensors, this system can collect various parameters from water, such as pH, dissolved oxygen, turbidity, conductivity, temperature, and so on. The rapid development of WSN technology provides a novel approach to real-time data acquisition, transmission, and processing. The clients can get ongoing water quality information from far away.

2.LITERATURE SURVEY

2.1.EXSISTING SYSTEM

Central water commission monitors water quality by collecting samples from representative locations within the processing and distribution system. These samples are analyzed at the well-eqquiped laboratories.At these laboratories samples from raw water,filter water and treated water for taken for analysis.The estimation of water parameters like turbidity,PH,dissolved oxygen,etc., is done with the help of meters.

2.2.REFERENCES

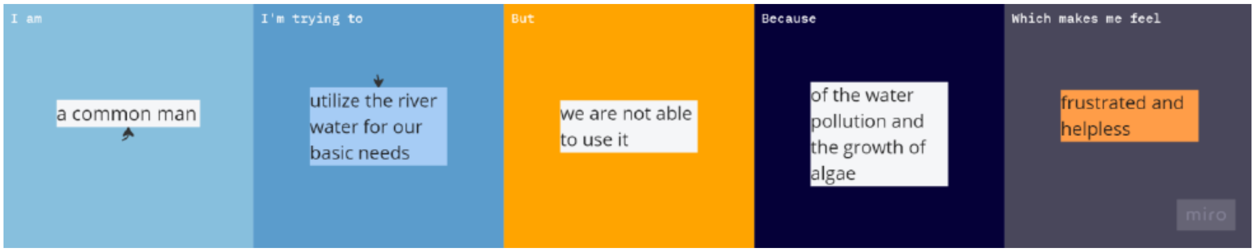
1. Water quality monitoring system based on Internet of Things
Author: Chengcheng Zhang, Jian Wu, Jiancheng Liu
Publication: IEEE 2020 Chengcheng et al presents a solution that integrates the design of STM32 single- chip microcomputer, sensors, WiFi wireless transmission and remote water quality management system. It monitors water quality turbidity, pH value, temperature and uploads the data to the management center through wireless communication.

2. IoT Based Real-time River Water Quality Monitoring System
Author: Mohammad Salah Uddin Chowdurya, Talha Bin Emran b, Subhasish Ghosha , Abhijit Pathak a, Mohd. Manjur Alama, Nurul Absar a, Karl Andersson c, Mohammad Shahadat Hossain d
Publication: Science Direct 2019 Mohammad et al proposed a manual method for sensorbased water quality monitoring system with high frequency, high mobility, and low power. Here the data collected at the site can be displayed in a visual format on a server PC with the help of Spark streaming analysis through Spark MLlib, Deep learning neural network models, Belief Rule Based (BRB) system and is also compared with standard value.

3. Efficient Cloud Based Real Time Water Quality Monitoring System Using IOT
Author: M.Usha Rani, Dr.R.Alageswaran, Sathish Kumar A
Publication: JASC: Journal of Applied Science and Computations(2018) M.Usha Rani et al proposes water sampling system with required sensor. Whenever the water level in the lakes or ponds reaches the lower/upper level it is identified and notification is sent

to the administrator. It can also predict overflow and water scarcity in future from the past results. The parameters like PH, calcium, sulphate and nitrate ions that is present in the water is also identified.

2.3.PROBLEM STATEMENT



I am

a common man

I'm trying to

utilize the river water for our basic needs

But

we are not able to use it

Because

of the water pollution and the growth of algae

Which makes me feel

frustrated and helpless

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	a common man	utilize the river water for our basic needs	we are not able to use it	of the water pollution and growth of algae	frustrated and helpless
PS-2	a farmer	irrigate the crops in my farm	the pH level of the water is imbalanced	of the wastages released from the industries	helpless and unable to irrigate my crops

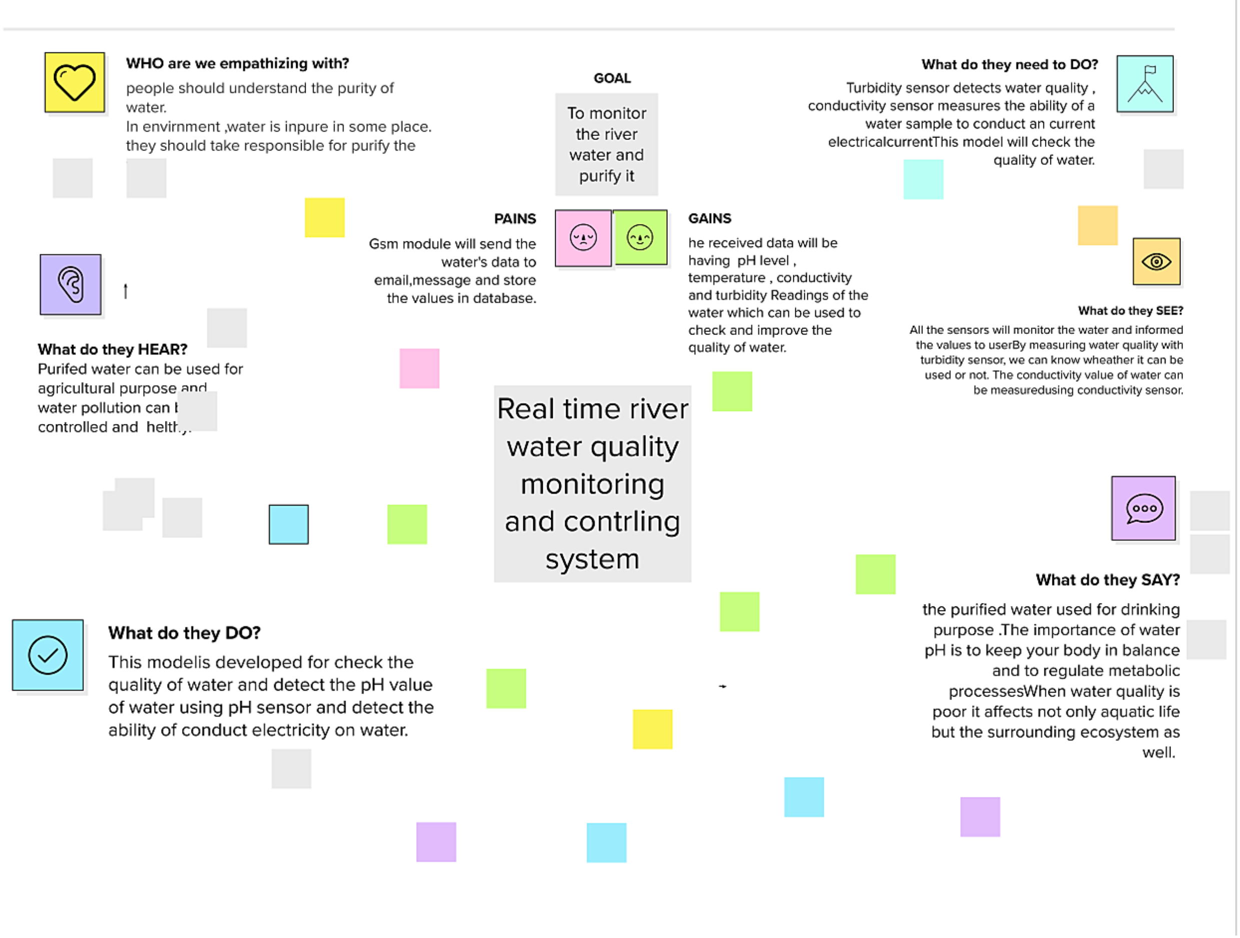
3. IDEATION & PROPOSED SOLUTION

3.1.PREPARE EMPATHY MAP

Empathy Map Canvas:

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user’s behaviours and attitudes.It is a useful tool to helps teams better understand their users.Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user’s perspective along with his or her goals and challenges.

Example



WHO are we empathizing with?

people should understand the purity of water.
In envirnment ,water is inpure in some place.
they should take responsible for purify the

GOAL

To monitor the river water and purify it

What do they need to DO?

Turbidity sensor detects water quality , conductivity sensor measures the ability of a water sample to conduct an current electricalcurrentThis model will check the quality of water.

PAINS

Gsm module will send the water's data to email,message and store the values in database.

GAINS

he received data will be having pH level , temperature , conductivity and turbidity Readings of the water which can be used to check and improve the quality of water.

What do they SEE?

All the sensors will monitor the water and informed the values to userBy measuring water quality with turbidity sensor, we can know wheather it can be used or not. The conductivity value of water can be measuredusing conductivity sensor.

What do they HEAR?

Purifed water can be used for agricultural purpose and water pollution can b controlled and helthy.

Real time river water quality monitoring and contrling system

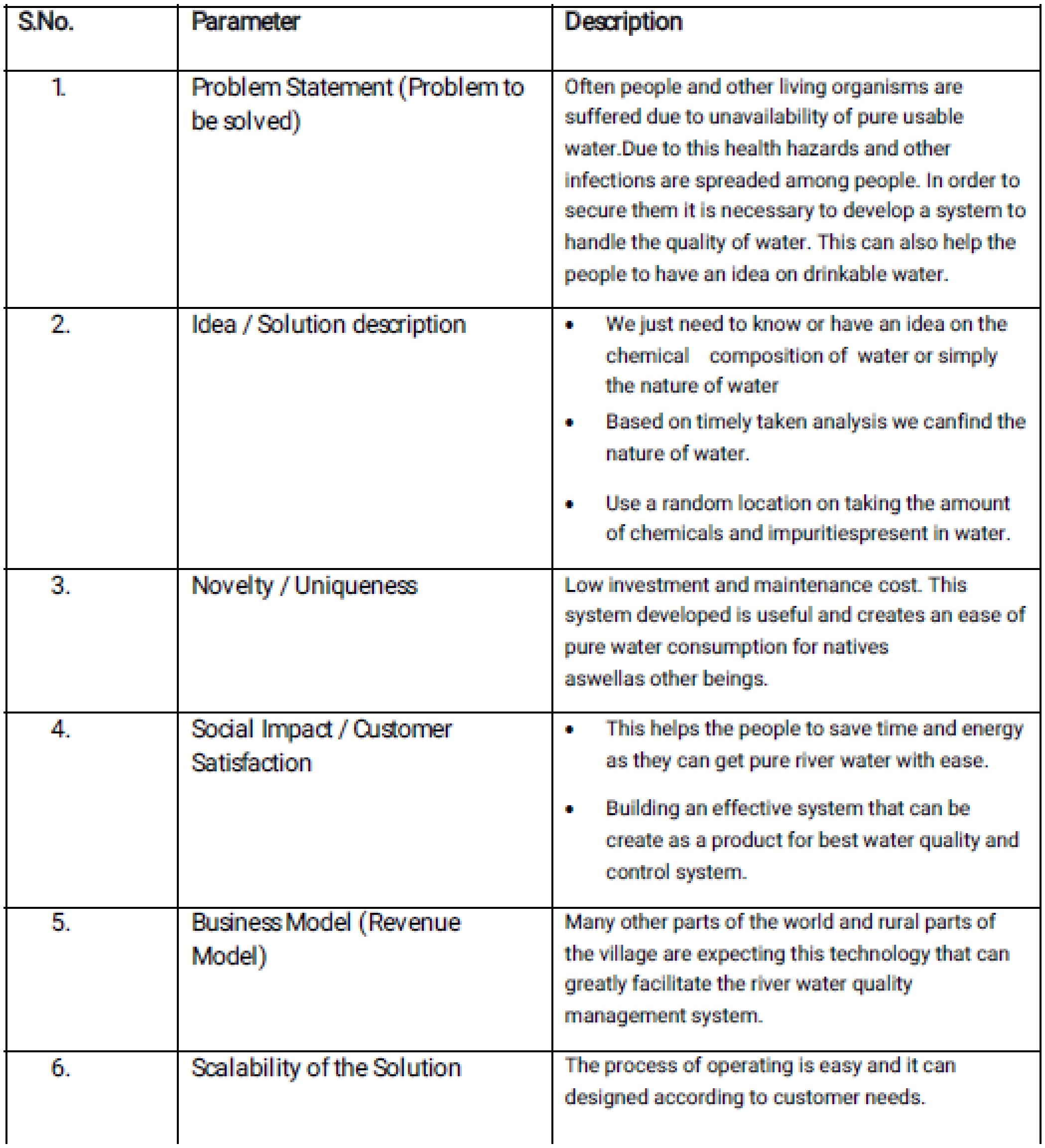
What do they DO?

This modelis developed for check the quality of water and detect the pH value of water using pH sensor and detect the ability of conduct electricity on water.

What do they SAY?

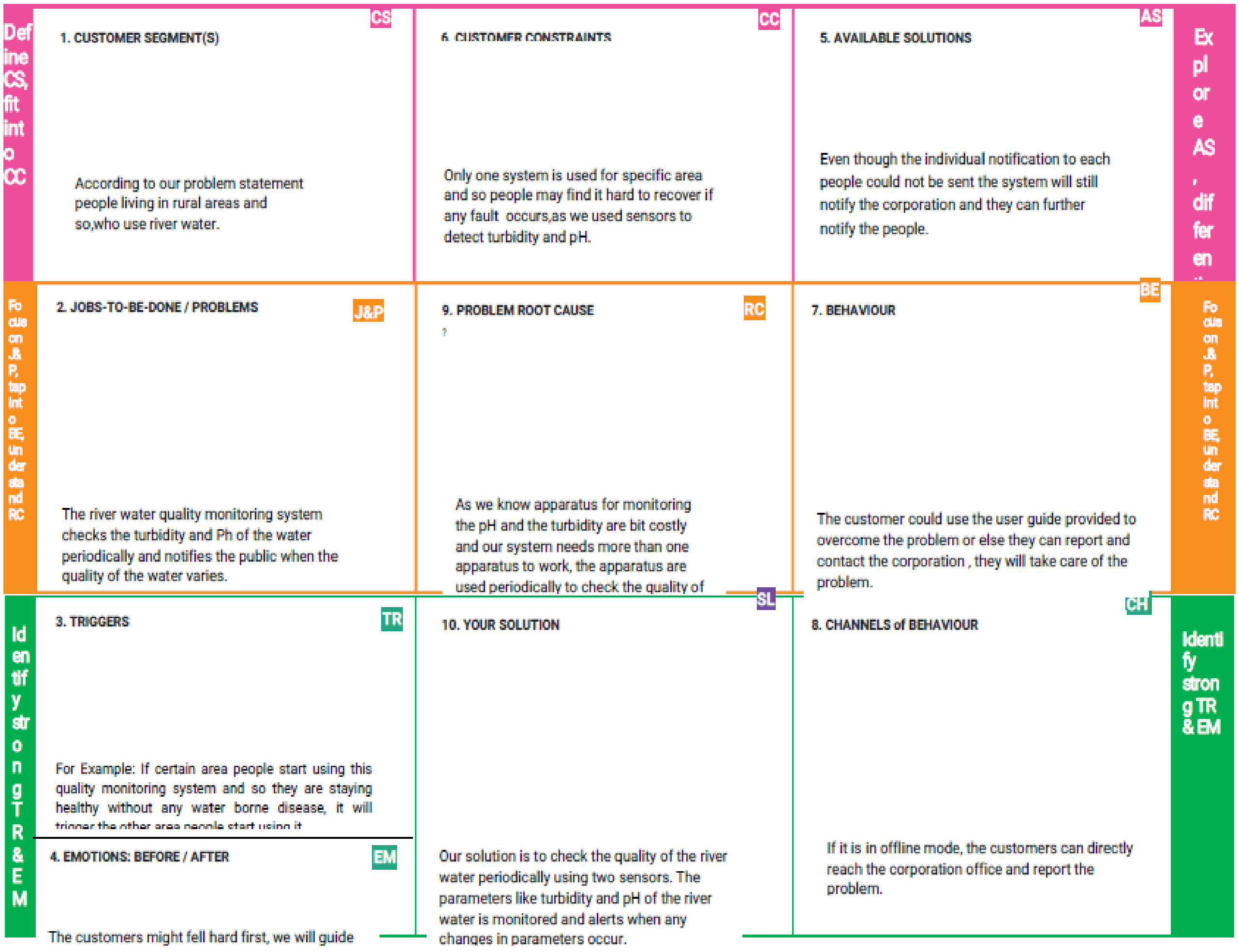
the purified water used for drinking purpose .The importance of water pH is to keep your body in balance and to regulate metabolic processesWhen water quality is poor it affects not only aquatic life but the surrounding ecosystem as well.

3.2.PROPOSED SOLUTION



S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Often people and other living organisms are suffered due to unavailability of pure usable water.Due to this health hazards and other infections are spreaded among people. In order to secure them it is necessary to develop a system to handle the quality of water. This can also help the people to have an idea on drinkable water.
2.	Idea / Solution description	• We just need to know or have an idea on the chemical composition of water or simply the nature of water • Based on timely taken analysis we canfind the nature of water. • Use a random location on taking the amount of chemicals and impuritiespresent in water.
3.	Novelty / Uniqueness	Low investment and maintenance cost. This system developed is useful and creates an ease of pure water consumption for natives aswellas other beings.
4.	Social Impact / Customer Satisfaction	• This helps the people to save time and energy as they can get pure river water with ease. • Building an effective system that can be create as a product for best water quality and control system.
5.	Business Model (Revenue Model)	Many other parts of the world and rural parts of the village are expecting this technology that can greatly facilitate the river water quality management system.
6.	Scalability of the Solution	The process of operating is easy and it can designed according to customer needs.

3.3. PROPOSED SOLUTION FIT



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

According to our problem statement people living in rural areas and so,who use river water.

6. CUSTOMER CONSTRAINTS CC

Only one system is used for specific area and so people may find it hard to recover if any fault occurs,as we used sensors to detect turbidity and pH.

5. AVAILABLE SOLUTIONS AS

Even though the individual notification to each people could not be sent the system will still notify the corporation and they can further notify the people.

Explore AS, differen

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

The river water quality monitoring system checks the turbidity and Ph of the water periodically and notifies the public when the quality of the water varies.

9. PROBLEM ROOT CAUSE RC

?

As we know apparatus for monitoring the pH and the turbidity are bit costly and our system needs more than one apparatus to work, the apparatus are used periodically to check the quality of

7. BEHAVIOUR BE

The customer could use the user guide provided to overcome the problem or else they can report and contact the corporation , they will take care of the problem.

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TR

For Example: If certain area people start using this quality monitoring system and so they are staying healthy without any water borne disease, it will trigger the other area people start using it

4. EMOTIONS: BEFORE / AFTER EM

The customers might fell hard first, we will guide

10. YOUR SOLUTION SL

Our solution is to check the quality of the river water periodically using two sensors. The parameters like turbidity and pH of the river water is monitored and alerts when any changes in parameters occur.

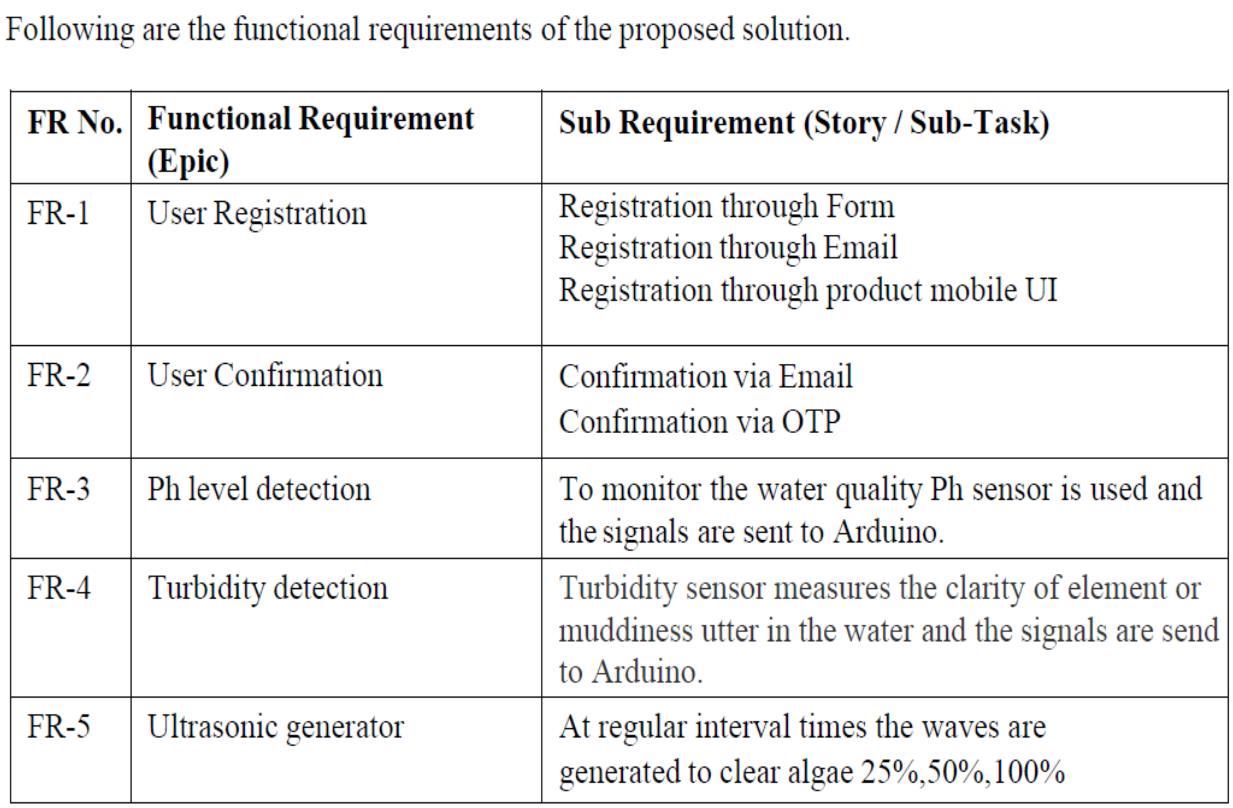
8. CHANNELS of BEHAVIOUR CH

If it is in offline mode, the customers can directly reach the corporation office and report the problem.

Identify strong TR & EM

4. REQUIREMENT ANALYSIS

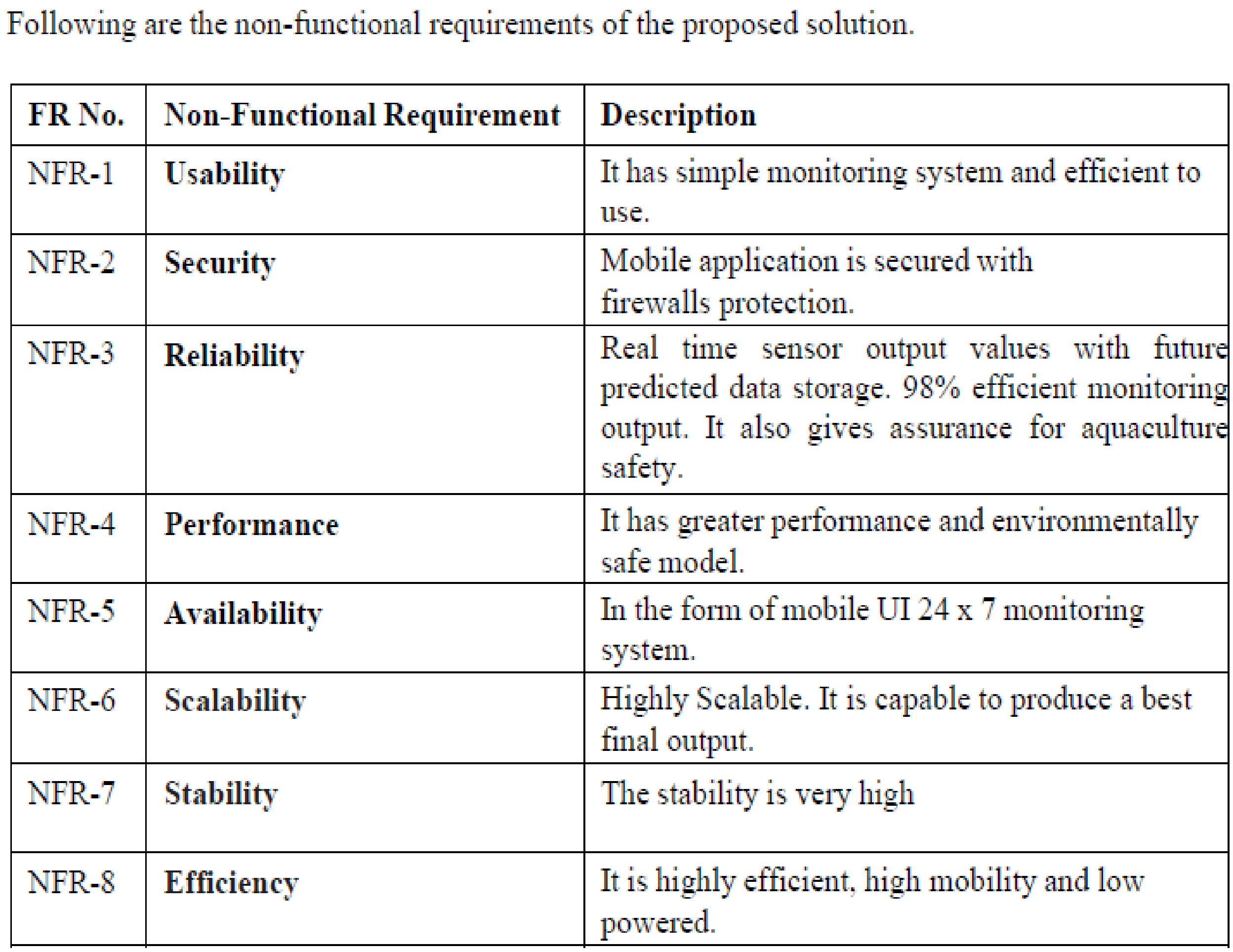
4.1.FUNCTIONAL REQUIREMENT



Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Email Registration through product mobile UI
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Ph level detection	To monitor the water quality Ph sensor is used and the signals are sent to Arduino.
FR-4	Turbidity detection	Turbidity sensor measures the clarity of element or muddiness utter in the water and the signals are send to Arduino.
FR-5	Ultrasonic generator	At regular interval times the waves are generated to clear algae 25%,50%,100%

4.2 .NON- FUNCTIONAL REQUIREMENTS



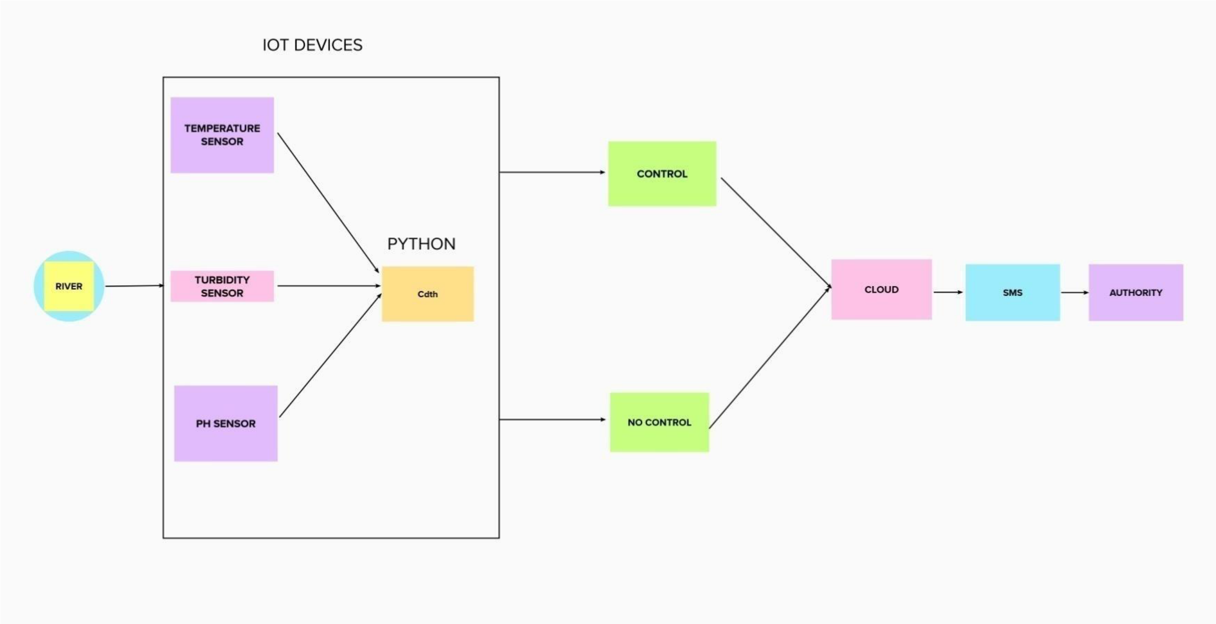
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	It has simple monitoring system and efficient to use.
NFR-2	**Security**	Mobile application is secured with firewalls protection.
NFR-3	**Reliability**	Real time sensor output values with future predicted data storage. 98% efficient monitoring output. It also gives assurance for aquaculture safety.
NFR-4	**Performance**	It has greater performance and environmentally safe model.
NFR-5	**Availability**	In the form of mobile UI 24 x 7 monitoring system.
NFR-6	**Scalability**	Highly Scalable. It is capable to produce a best final output.
NFR-7	**Stability**	The stability is very high
NFR-8	**Efficiency**	It is highly efficient, high mobility and low powered.

5.PROJECT DESIGN

5.1.DATA FLOW DIAGRAM

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



5.2.TECHNOLOGY ARCHITECTURE

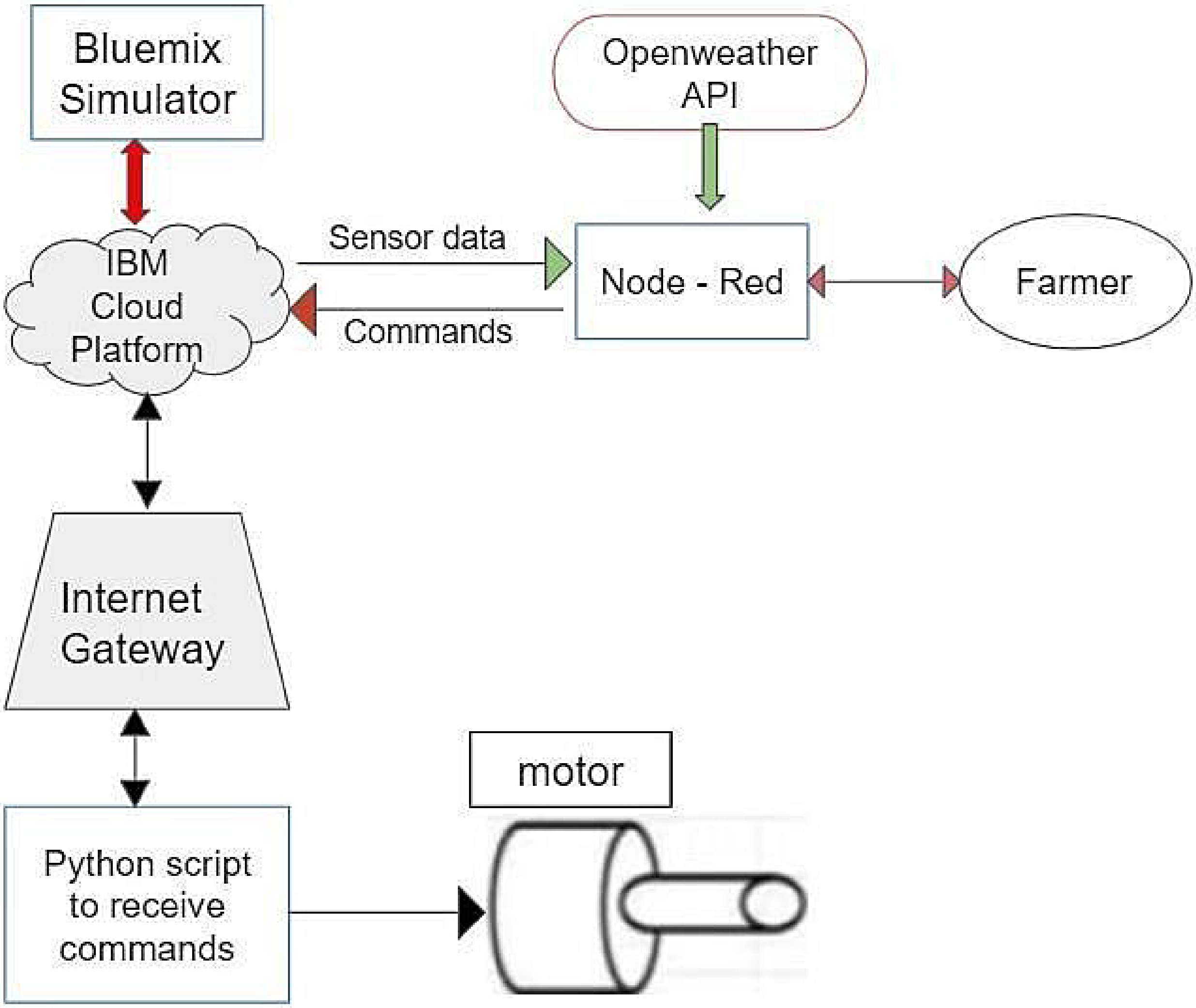
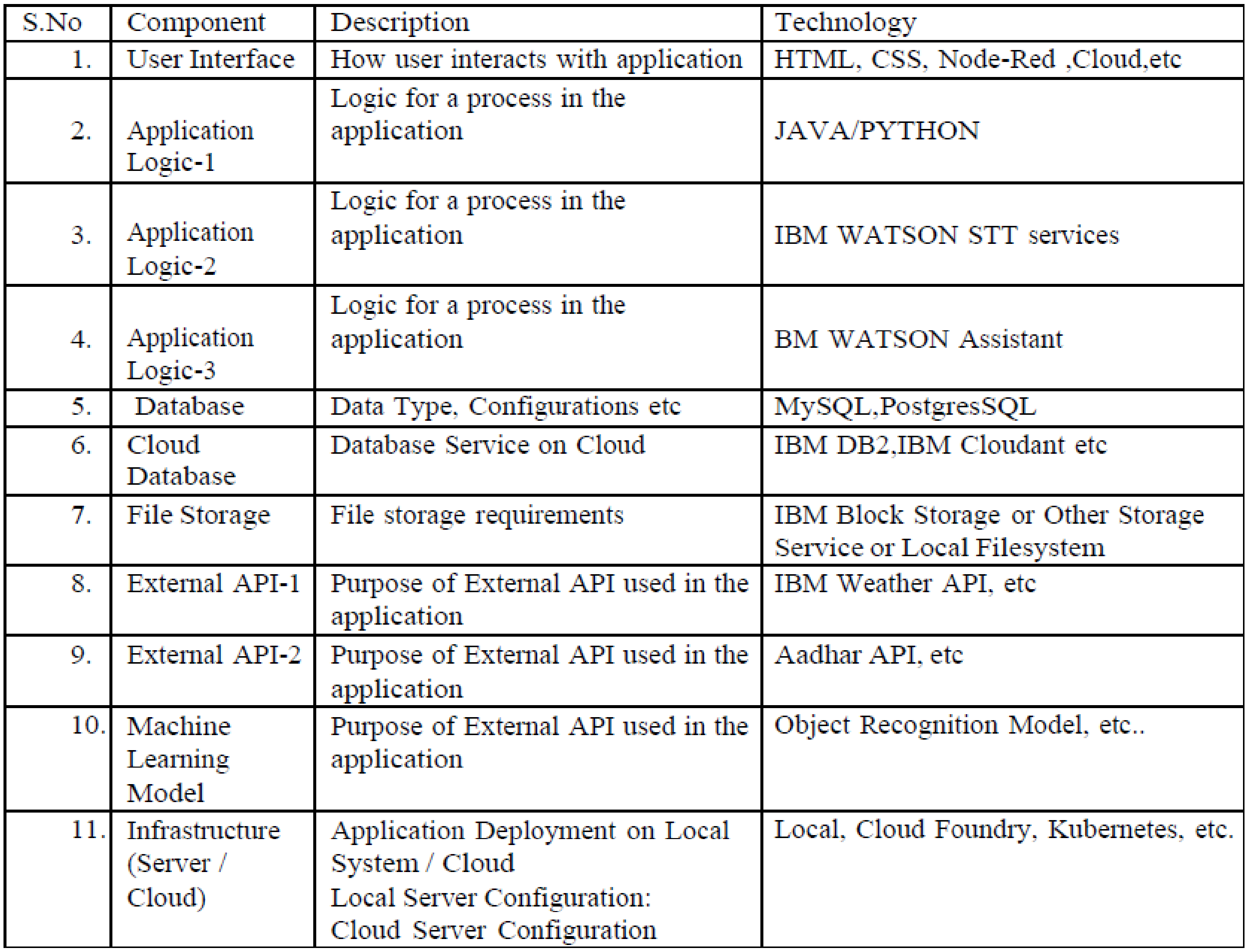
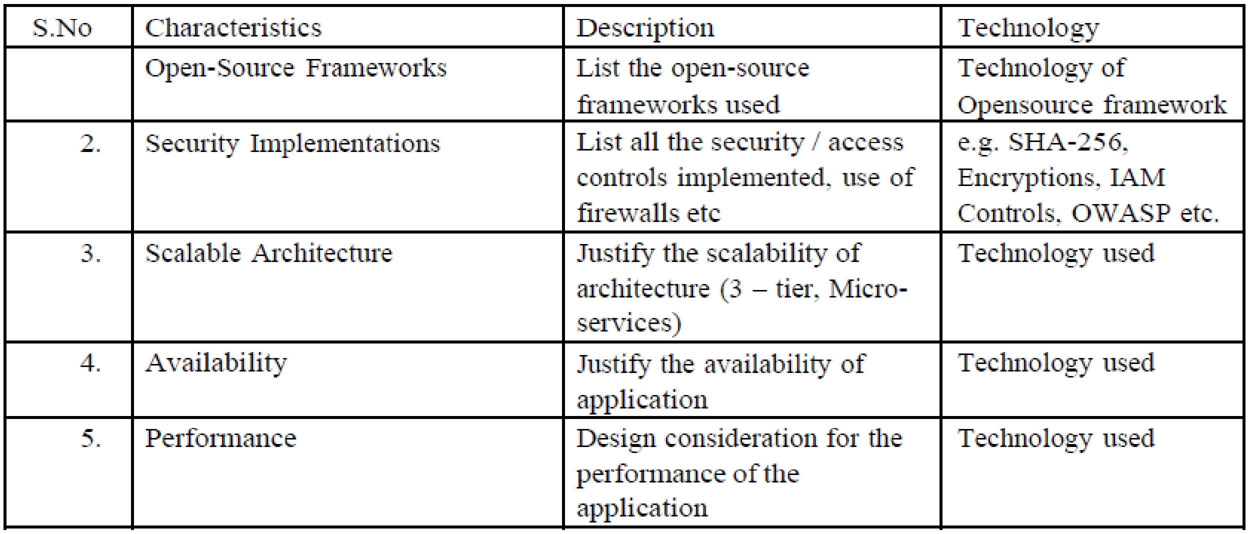


Table-1 : Components & Technologies:



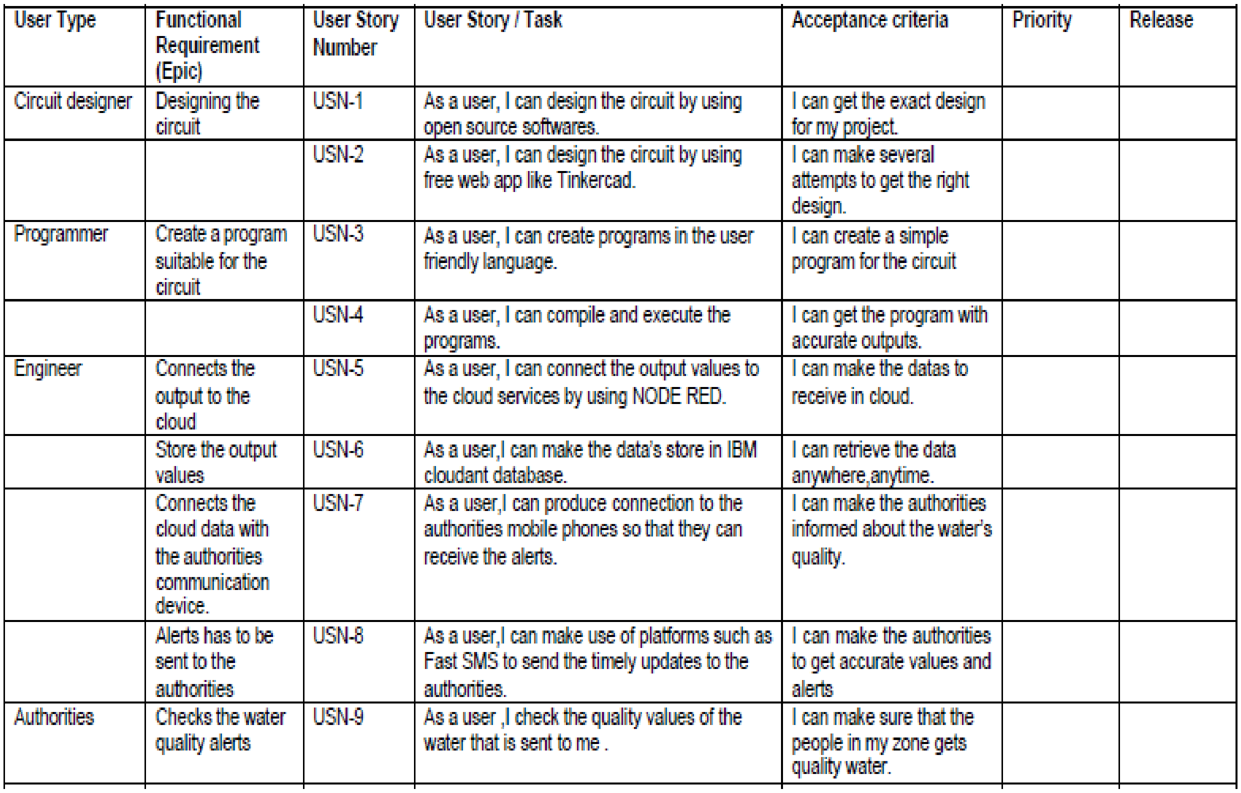
S.No	Component	Description	Technology
1.	User Interface	How user interacts with application	HTML, CSS, Node-Red ,Cloud,etc
2.	Application Logic-1	Logic for a process in the application	JAVA/PYTHON
3.	Application Logic-2	Logic for a process in the application	IBM WATSON STT services
4.	Application Logic-3	Logic for a process in the application	BM WATSON Assistant
5.	Database	Data Type, Configurations etc	MySQL,PostgresSQL
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	Aadhar API, etc
10.	Machine Learning Model	Purpose of External API used in the application	Object Recognition Model, etc..
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
	Open-Source Frameworks	List the open-source frameworks used	Technology of Opensource framework
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc	e.g. SHA-256, Encryptions, IAM Controls, OWASP etc.
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Technology used
4.	Availability	Justify the availability of application	Technology used
5.	Performance	Design consideration for the performance of the application	Technology used

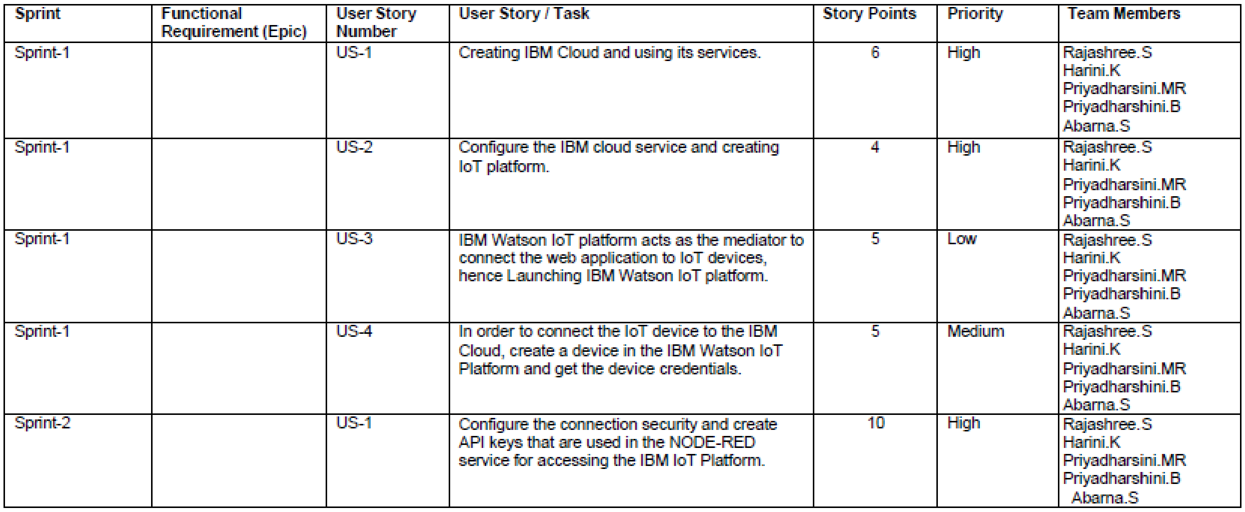
5.3.USER STORIES



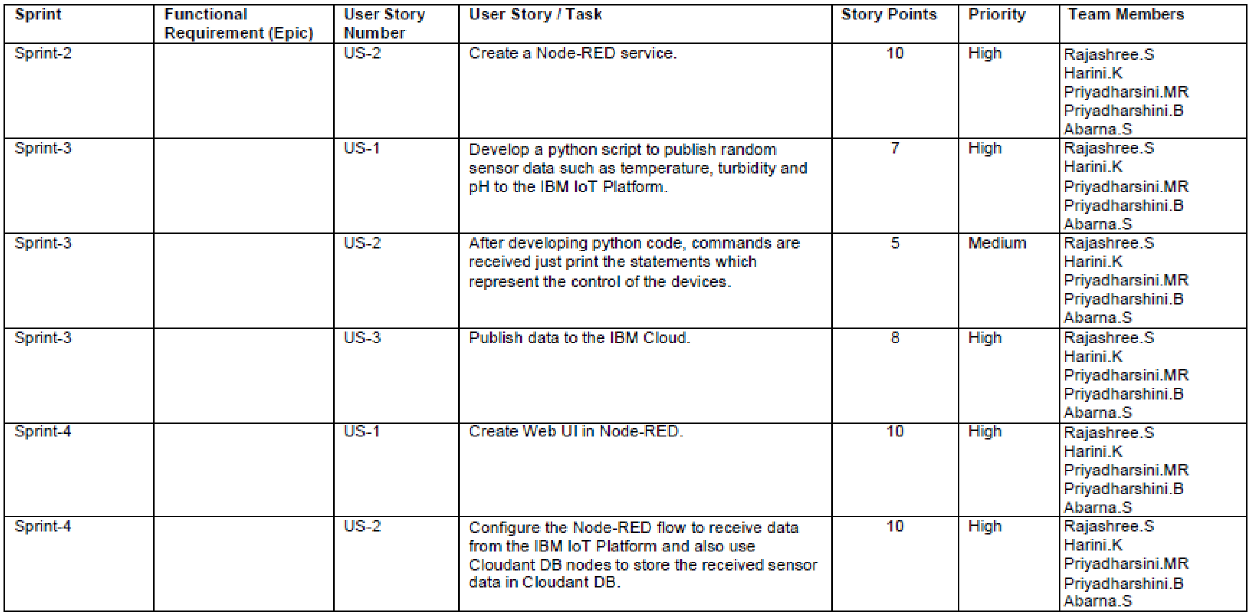
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Circuit designer	Designing the circuit	USN-1	As a user, I can design the circuit by using open source softwares.	I can get the exact design for my project.		
		USN-2	As a user, I can design the circuit by using free web app like Tinkercad.	I can make several attempts to get the right design.		
Programmer	Create a program suitable for the circuit	USN-3	As a user, I can create programs in the user friendly language.	I can create a simple program for the circuit		
		USN-4	As a user, I can compile and execute the programs.	I can get the program with accurate outputs.		
Engineer	Connects the output to the cloud	USN-5	As a user, I can connect the output values to the cloud services by using NODE RED.	I can make the datas to receive in cloud.		
	Store the output values	USN-6	As a user,I can make the data's store in IBM cloudant database.	I can retrieve the data anywhere,anytime.		
	Connects the cloud data with the authorities communication device.	USN-7	As a user,I can produce connection to the authorities mobile phones so that they can receive the alerts.	I can make the authorities informed about the water's quality.		
	Alerts has to be sent to the authorities	USN-8	As a user,I can make use of platforms such as Fast SMS to send the timely updates to the authorities.	I can make the authorities to get accurate values and alerts		
Authorities	Checks the water quality alerts	USN-9	As a user ,I check the quality values of the water that is sent to me .	I can make sure that the people in my zone gets quality water.		

6. PROJECT PLANNING & SCHEDULING

6.1.SPRINT DELIVERY PLANNING SCHEDULE

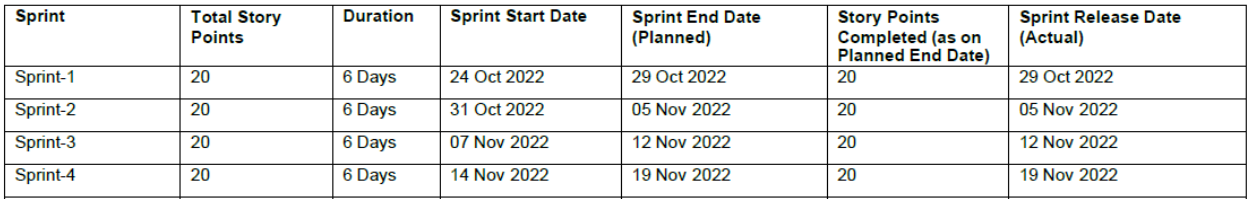


Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1		US-1	Creating IBM Cloud and using its services.	6	High	Rajashree.S Harini.K Priyadharsini.MR Priyadharshini.B Abarna.S
Sprint-1		US-2	Configure the IBM cloud service and creating IoT platform.	4	High	Rajashree.S Harini.K Priyadharsini.MR Priyadharshini.B Abarna.S
Sprint-1		US-3	IBM Watson IoT platform acts as the mediator to connect the web application to IoT devices, hence Launching IBM Watson IoT platform.	5	Low	Rajashree.S Harini.K Priyadharsini.MR Priyadharshini.B Abarna.S
Sprint-1		US-4	In order to connect the IoT device to the IBM Cloud, create a device in the IBM Watson IoT Platform and get the device credentials.	5	Medium	Rajashree.S Harini.K Priyadharsini.MR Priyadharshini.B Abarna.S
Sprint-2		US-1	Configure the connection security and create API keys that are used in the NODE-RED service for accessing the IBM IoT Platform.	10	High	Rajashree.S Harini.K Priyadharsini.MR Priyadharshini.B Abarna.S



Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2		US-2	Create a Node-RED service.	10	High	Rajashree.S Harini.K Priyadharsini.MR Priyadharshini.B Abarna.S
Sprint-3		US-1	Develop a python script to publish random sensor data such as temperature, turbidity and pH to the IBM IoT Platform.	7	High	Rajashree.S Harini.K Priyadharsini.MR Priyadharshini.B Abarna.S
Sprint-3		US-2	After developing python code, commands are received just print the statements which represent the control of the devices.	5	Medium	Rajashree.S Harini.K Priyadharsini.MR Priyadharshini.B Abarna.S
Sprint-3		US-3	Publish data to the IBM Cloud.	8	High	Rajashree.S Harini.K Priyadharsini.MR Priyadharshini.B Abarna.S
Sprint-4		US-1	Create Web UI in Node-RED.	10	High	Rajashree.S Harini.K Priyadharsini.MR Priyadharshini.B Abarna.S
Sprint-4		US-2	Configure the Node-RED flow to receive data from the IBM IoT Platform and also use Cloudant DB nodes to store the received sensor data in Cloudant DB.	10	High	Rajashree.S Harini.K Priyadharsini.MR Priyadharshini.B Abarna.S

Project Tracker, Velocity & Burndown Chart:



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

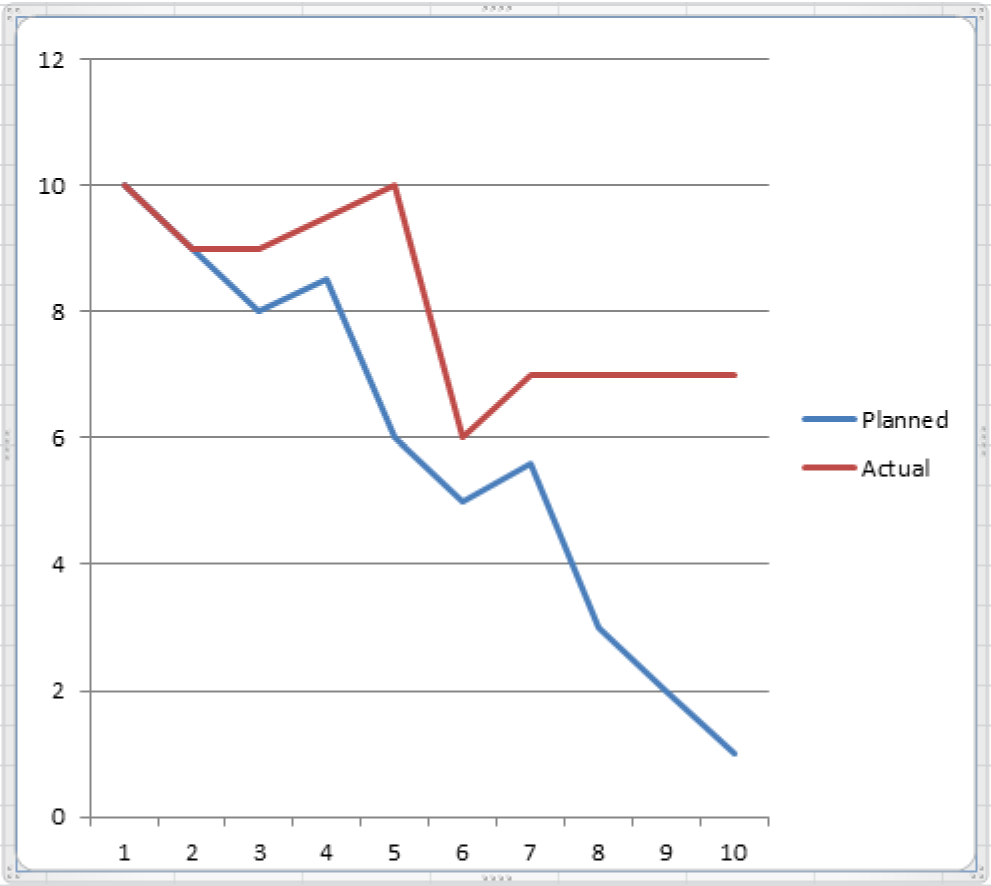
Velocity:

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

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

Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as scrum. However, burn down charts can be applied to any project containing measurable progress over time



7.CODING AND SOLUTION

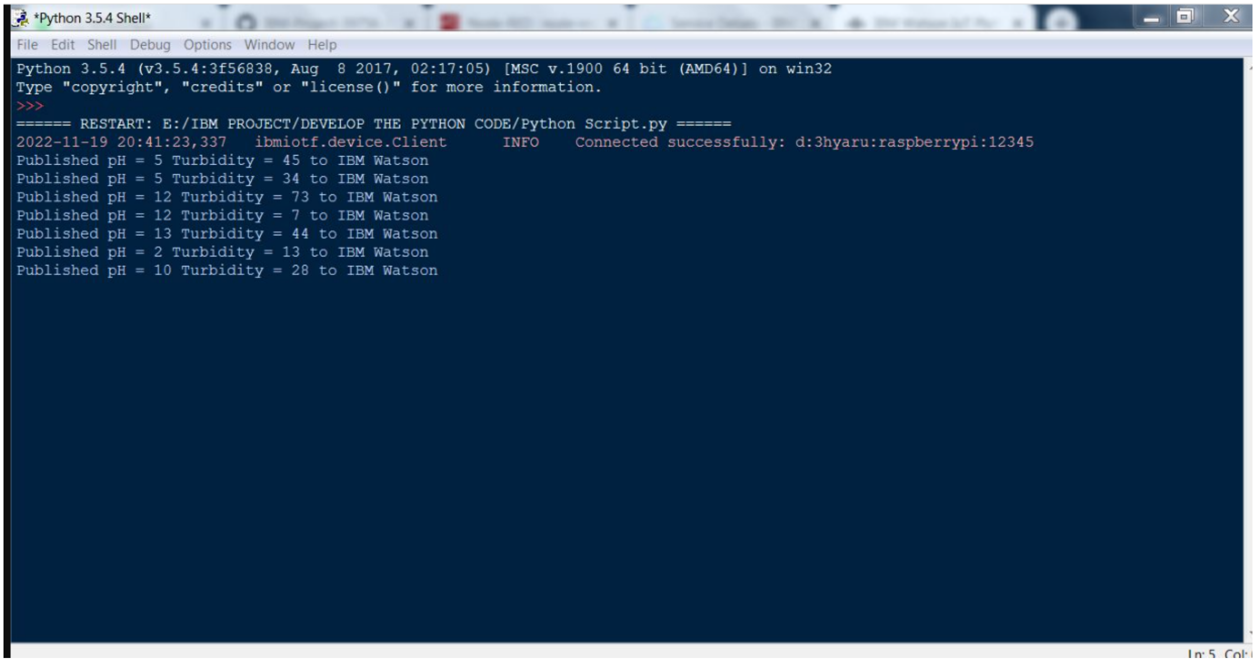
7.1 FEATURE - 1

Receiving commands from IBM cloud using Python program.
This is the Python code to receive commands from cloud to any device like Raspberry Pi in the river.

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <PubSubClient.h>
#include "DHT.h"
const char* ssid = "SMART-G";
const char* password = "10112019";
#define DHTPIN D6
#define G D0
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
#define ID "3hyaru"
#define DEVICE_TYPE "raspberrypi"
#define DEVICE_ID "12345"
#define TOKEN "TEST-12345"
char server[] = ID ".messaging.internetofthings.ibmcloud.com";
char publish_Topic1[] = "iot-2/evt/Data1/fmt/json";
char publish_Topic2[] = "iot-2/evt/Data2/fmt/json";
char publish_Topic3[] = "iot-2/evt/Data2/fmt/json";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ID ":" DEVICE_TYPE ":" DEVICE_ID;
WiFiClient wifiClient;
PubSubClient client(server, 1883, NULL, wifiClient);
void setup()
{
```

```
pinMode(D0,OUTPUT);
digitalWrite(D0,HIGH);
Serial.begin(115200);
dht.begin();
Serial.println();
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.println("");
Serial.println(WiFi.localIP());
if (!client.connected()) {
Serial.print("Reconnecting client to ");
Serial.println(server);
while (!client.connect(clientId, authMethod, token))
{
Serial.print(".");
delay(500);
}
Serial.println("Connected TO IBM IoT cloud!");
}
}
long previous_message = 0;
void loop()
{
client.loop();
long current = millis();
if (current - previous_message > 3000)
{
previous_message = current;
float hum = dht.readHumidity();
float temp = dht.readTemperature();
float MOI = map(analogRead(A0), 0, 1023, 100, 0);
if (isnan(hum) || isnan(temp) )
{
Serial.println(F("Failed to read from DHT sensor!"));
return;
}
Serial.print("Temperature: ");
Serial.print(temp);
Serial.print("°C");
Serial.print(" Humidity: ");
Serial.print(hum);
Serial.print("%");
Serial.print("WATER PH: ");
Serial.print(MOI);
if(MOI<=10)
{
digitalWrite(D0,LOW);
delay(100);
digitalWrite(D0,HIGH);
}
else
{
digitalWrite(D0,HIGH);
}
String payload = "{\"d\":{\"Name\":\"" DEVICE_ID "\"";
payload += ",\"Temperature\":";
payload += temp;
payload += "}}";
Serial.print("Sending payload: ");
Serial.println(payload);
if (client.publish(publish_Topic1, (char*) payload.c_str()))
```

```
{
Serial.println("Published successfully");
} else {
Serial.println("Failed");
}
String payload1 = "{\"d\":{\"Name\":\"" DEVICE_ID "\"";
payload1 += ",\"Humidity\":";
payload1 += hum;
payload1 += "}}";
Serial.print("Sending payload: ");
Serial.println(payload1);
Serial.println('\n');
if (client.publish(publish_Topic2, (char*) payload1.c_str())) {
Serial.println("Published successfully");
} else {
Serial.println("Failed");
}
String payload3 = "{\"d\":{\"Name\":\"" DEVICE_ID "\"";
payload3 += ",\"WATER PH\":";
payload3 += MOI;
payload3 += "}}";
Serial.print("Sending payload: ");
Serial.println(payload3);
if (client.publish(publish_Topic3, (char*) payload3.c_str())) {
Serial.println("Published successfully");
} else {
Serial.println("Failed");
}
}
}
```

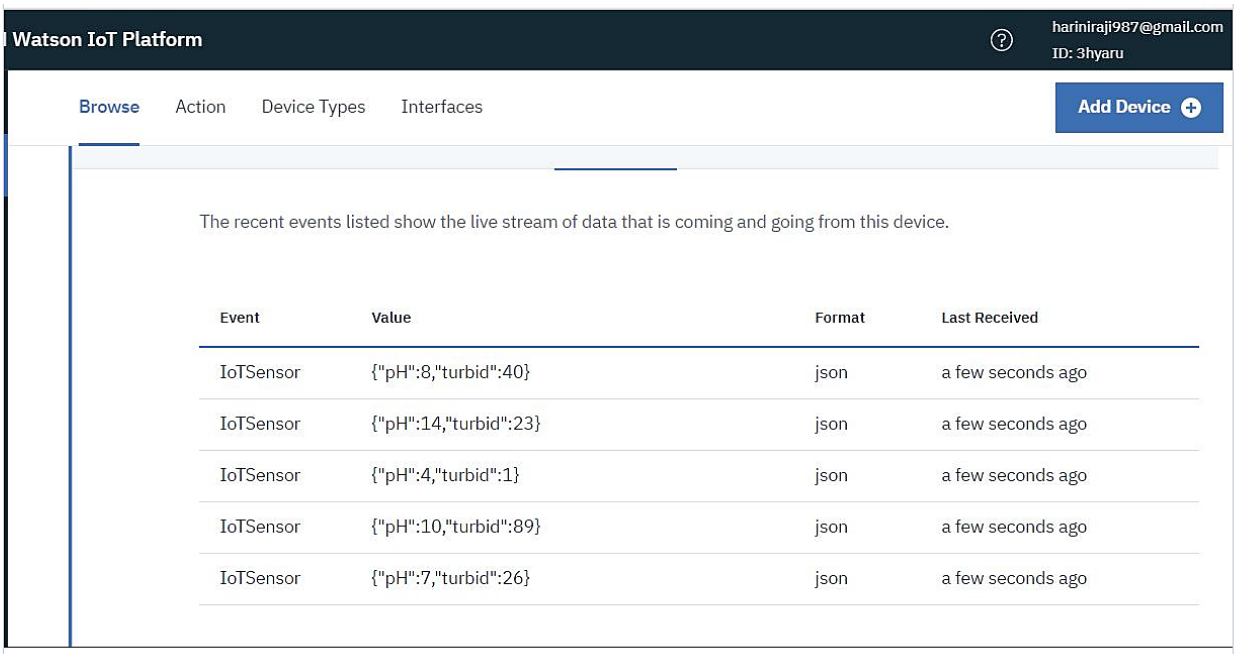


```
Python 3.5.4 (v3.5.4:3f56838, Aug  8 2017, 02:17:05) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
====== RESTART: E:/IBM PROJECT/DEVELOP THE PYTHON CODE/Python Script.py ======
2022-11-19 20:41:23,337   ibmiotf.device.Client      INFO    Connected successfully: d:3hyaru:raspberrypi:12345
Published pH = 5 Turbidity = 45 to IBM Watson
Published pH = 5 Turbidity = 34 to IBM Watson
Published pH = 12 Turbidity = 73 to IBM Watson
Published pH = 12 Turbidity = 7 to IBM Watson
Published pH = 13 Turbidity = 44 to IBM Watson
Published pH = 2 Turbidity = 13 to IBM Watson
Published pH = 10 Turbidity = 28 to IBM Watson
```

Feature 2:

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

#Provide your IBM Watson Device Credentials
organization = "3hyaru"
deviceType = "raspberrypi"
deviceId = "12345"
authMethod = "token"
authToken = "12345678"

# 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

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

        data = { 'pH' : pH, 'turbid': turbidity }
        #print data
        def myOnPublishCallback():
            print ("Published pH = %s" % pH, "Turbidity = %s" %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

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



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

Event	Value	Format	Last Received
IoTSensor	{"pH":8,"turbid":40}	json	a few seconds ago
IoTSensor	{"pH":14,"turbid":23}	json	a few seconds ago
IoTSensor	{"pH":4,"turbid":1}	json	a few seconds ago
IoTSensor	{"pH":10,"turbid":89}	json	a few seconds ago
IoTSensor	{"pH":7,"turbid":26}	json	a few seconds ago

8.ADVANTAGES

1. The prototype developed for water quality management is very beneficial for safeguarding public health.
2. Removes manual labor by automation.
3. Easy to control and alert

DISADVANTAGES

1. Difficult to collect the water samples as the cost of analysis is very high.
2. The method is prone to human errors.

9.CONCLUSION

In this research proposal, a neural network-based solution for automobile detection will be used to address the issues of automotive damage analysis and position and severity prediction. This project does several tasks in one bundle. The method will unquestionably assist the insurance firms in conducting far more thorough and systematic analyses of the vehicle damage. Simply sending the system a photograph of the vehicle, it will evaluate it and determine whether there is damage of any type, where it is located, and how severe it is.

10.FUTURE SCOPE

Due to the limitation of the budget, 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. So the additional budget is required for further improvement of the overall system.

11. APPENDIX

Github Link: https://github.com/IBM-EPBL/IBM-Project-33869-1660228170