# REAL TIME RIVER WATER MONITORING AND CONTROL SYSTEM

**A PROJECT REPORT**

***Submitted by***

***Team ID: PNT2022TMID44989***


| | |
|---|---|
| **SURENDRAN. S** | **811219106009** |
| **DINESH. J** | **811219106003** |
| **SRIKANTH. M** | **811219106008** |
| **SURYA. V** | **811219106010** |
| **SHYAM. C. R** | **811219106007** |


***In partial fulfillment for the award of the***

***degree of***

**BACHELOR OF ENGINEERING**

**IN**

**ELECTRONICS AND COMMUNICATION ENGINEERING**


**INDRA GANESAN COLLEGE OF ENGINEERING, TRICHY**

**ANNA UNIVERSITY: CHENNAI 600 025**


**NOVEMBER 2022**

# TABLE OF CONTENT

# CHAPTER 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. Otherwise, it would severely damage the health of the humans and at the same time affect the ecological balance among other species.

Water pollution is a foremost global problem which needs ongoing evaluation and adaptation of water resource directorial principle at the levels of international down to individual wells. It has been studied that water pollution is the leading cause of mortalities and diseases worldwide. The records show that more than 14,000 people die daily worldwide due to water pollution. In many developing countries, dirty or contaminated water is being used for drinking without any proper prior treatment. One of the reasons for this happening is the ignorance of public and administration and the lack of water quality monitoring system which makes serious health issues.

In this project, we depict the design of Wireless Sensor Network (WSN) that assists to monitor the quality of water with the support of information sensed by the sensors dipped in water and controlling the algae present in 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. Now a day's Internet of things (IoT) is an innovative technological phenomenon. It is shaping today's world and is used in different fields for collecting, monitoring and analysis of data from remote locations. IoT integrated network if everywhere starting from smart cities, smart power grids, and smart supply chain to smart wearable. Though IoT is still under applied in the field of environment it has huge potential. It can be applied to detect forest fire and early earthquake, reduce air population, monitor snow level, prevent landslide, and avalanche etc. Moreover, it can be implemented in the field of water quality monitoring and controlling system.

Water quality monitoring has gained more interest among researchers in this twenty-first century. Numerous works are either done or ongoing in this topic focusing on various aspects of it. The key theme of all the projects was to develop an efficient, cost-effective, real-time water quality monitoring and system which will integrate wireless sensor network and internet of things.

## 1.2 Purpose

The major goal is to create a system that uses wireless sensor networks to continuously monitor river water quality at remote locations with low power consumption, low cost and high detection accuracy. pH, conductivity, turbidity level and other parameters are measured in order to enhance water quality. The remote sensing technology is the cornerstone of IoT-based water quality monitoring. This implement the approach by using the pH sensor, turbidity sensor to obtain analog readings for water contaminates. In addition, for the specific application, we can add extra sensor elements.

# CHAPTER 2

# LITERATURE SURVEY

## 2.1 Existing Problem

- If large amounts of fertilizers or farm waste drain into a river the concentration of nitrate and phosphate in the water increases considerably. Algae use these substances to grow and multiply rapidly turning the water green.
- This massive growth of algae leads to pollution. when the algae die they are broken down by the action of the bacteria which quickly multiply, using up all the oxygen in the water which leads to many problems .
- To avoid those problems, control the algae and monitor the water parameters like PH, temperature in the river water.

## 2.2 References

**1. IOT based Think Speak application for monitoring the quality of the water**. **Pasika and Gandla**

The monitoring system which consists of a number of sensors used to measure several quality parameters like turbidity, pH value, water level in the tank, dampness of the adjoining environment and temperature of the water. The sensors are interfaced with the Microcontroller Unit (MCU) and additional processing is executed by the Personal Computer (PC). The acquired data will be directed to the cloud by means of Internet of Things (IoT) based Think Speak application for monitoring the quality of the water under test. As a future directive, work should be extended for analyzing some other parameters such as nitrates, electrical conductivity, dissolved oxygen in the water and free residual chlorine.

**2. Solar powered water quality monitoring system using WSN.**

**M. Kulkarni Amruta and M. Turkane Satish**

**Published on 2013 by IEEE**

The idea of 'Underwater Wireless Sensor Network' (UWSN) is the basic building block of a water quality monitoring using wireless sensor network (WSN) technology powered by solar panel. To monitor water quality over different sites as a real-time application, an excellent system architecture constituted by distributed sensor nodes and a base station is suggested. The nodes and base station are connected using WSN technology like Zigbee. Design and implementation of a prototype model using one node powered by solar cell and WSN technology is the challenging work. Data collected by various sensors at the node side such as pH, turbidity and oxygen level is sent via WSN to the base station. Data collected from the remote site can be displayed in visual format as well as it can be analyzed using different simulation tools at base station. This novel system has advantages such as no carbon emission, low power consumption, more flexible to deploy at remote site and so on.

**3. IOT based Smart Water Quality Monitoring System.**

**Monjra Mukta, Samia Islam and M.S.H. Khan**

**Published on 1 Feb 2019 (4th ICCCS)**

This paper represents an IOT based smart water quality monitoring(SWQM) system aids in continuous measurement of water condition based on four physical parameters i.e., temperature, Ph, electric conductivity and turbidity properties. Four sensors are connected with Arduino-uno in discrete way to detect the water parameters. Extracted data from the sensors are transmitted to a desktop application developed in NET platform and compared with the WHO standard values. Based on

the measured result, the proposed SWQM system can successfully analyze the water parameters using fast forest binary classifier to classify whether the test water sample is drinkable or not.

**4. Design and Implementation of Real Time Approach for the Monitoring of Water Quality Parameters.**

**Siti Aishah Binti Makhtar, Norhafizah Binti Burham, Anees Bt Abdul Aziz**

**Published on June 2022 by IEEE**

This presented paperwork is to develop a smart water quality monitoring system using four sensors and an IoT platform to help determine water quality. It is to analyze the parameters of water samples such as tap water, coway water, river water, pond water, and lake water whether these water samples are in the threshold range for drinking or not. The device is initially used to measure pH, turbidity, total dissolved solids (TDS) and temperature, and then sent the information to the microcontroller Arduino Uno. Users can connect the device to a mobile phone via Bluetooth, and then an android-based mobile application called HC-05 Bluetooth Terminal displays real-time test data. These values of each parameter are also displayed on the I2C LCD screen connected to the microcontroller.

**5. IoT and Cloud based water conservation and monitoring system**

**Avita Katal, Sharad Singhania and Sakshi Jain**

**Published on 26 Aug 2022 (ASIANCON)**

There have been many researches whose major focus has been on water conservation but none of them provides with the plan on how to utilize water in an effective manner and minimize water wastage. The proposed system uses ultrasonic and water-level sensors to detect multiple metrics such as the vessel's water level as well as the individual’s daily water consumption. These sensors are connected to

the Node Microcontroller Unit (NodeMCU), which performs additional computations. The real time data collected is uploaded to the database. A self-designed web application is used to show the water usage, alerts in case of water wastage and the recommendations to users in order to help them planning better water utilization.

## 2.3 Project Statement Definition

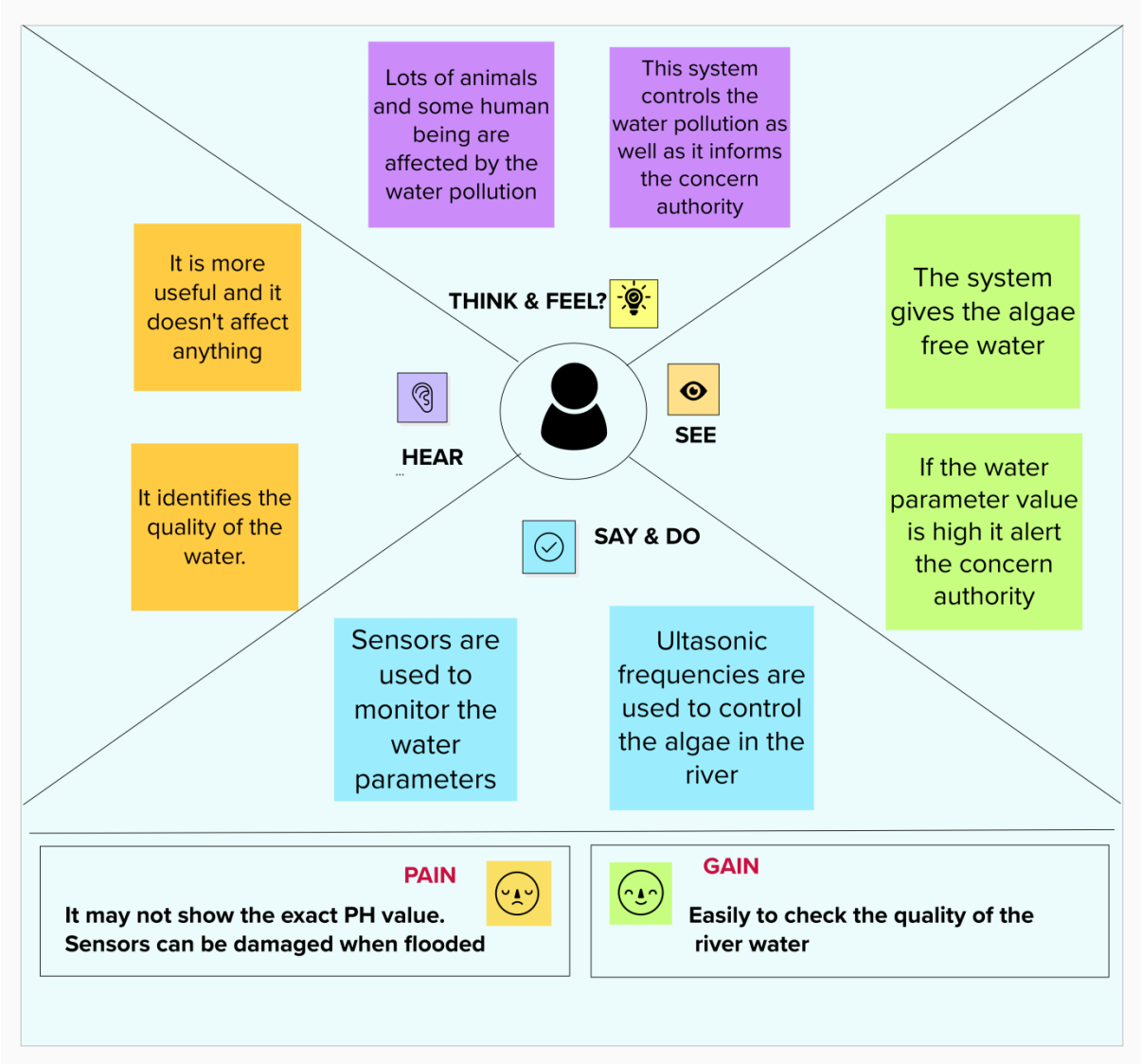
- Farmers put fertilizers and pesticides on their crops so that they grow better. But these fertilizers and pesticides can be washed through the soil by rain, to end up in rivers.
- If large amount of fertilizers or farm waste drain into a river the concentration of nitrate and phosphate in the water increases considerably. Algae use these substances to grow and multiply rapidly turning the water green.
- This massive growth of algae, called eutrophication, leads to pollution. When the algae die they are broken down by the action of the bacteria which quickly multiply, using up all the oxygen in the water which leads to the death of many animals.

| **Problem Statement (PS)** | **I am (Customer)** | **I'm trying to** | **But** | **Because** | **Which makes me feel** |
|---|---|---|---|---|---|
| PS-1 | People | To use the water for cooking. | It was dirty. | Medical wastages and fertilizers are mixed with the river water. | Fear to use the River water. |
| PS-2 | People | I want to drink the river water. | It was not clean. | Algae present in the river. | Hard to drink |

# CHAPTER 3

# IDEATION & PROPOSED SOLUTION

## 3.1 Empathy Map Canvas

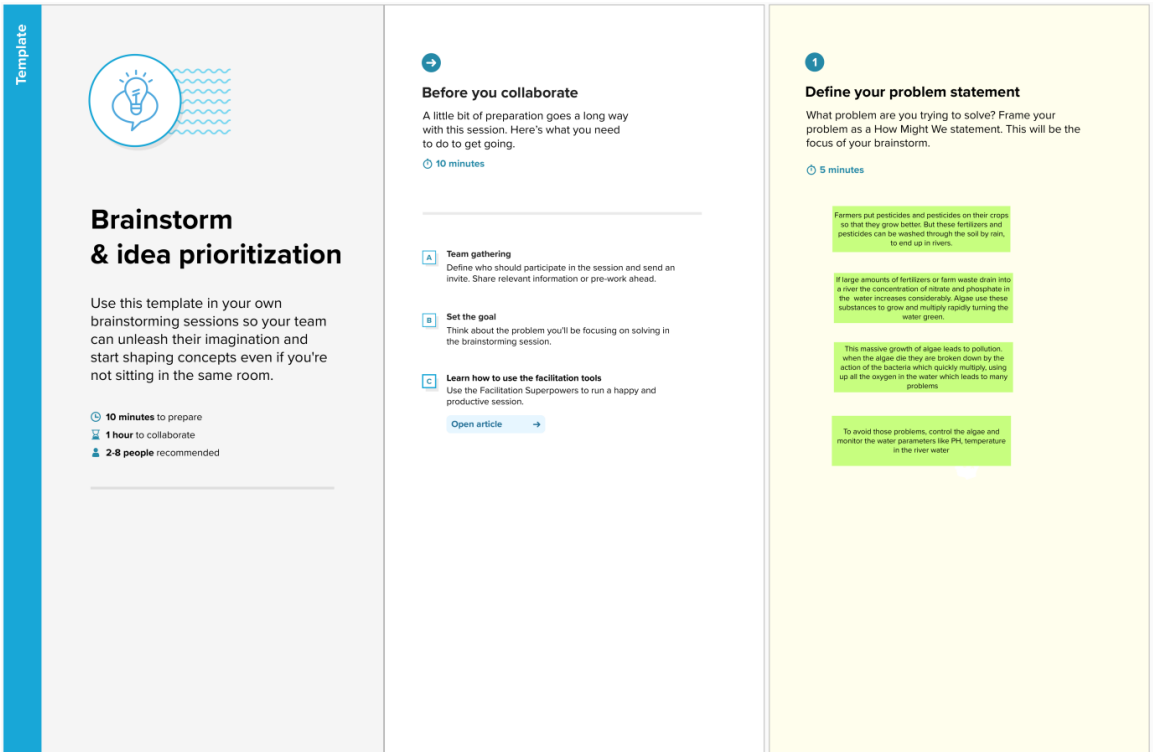



**Reference:**

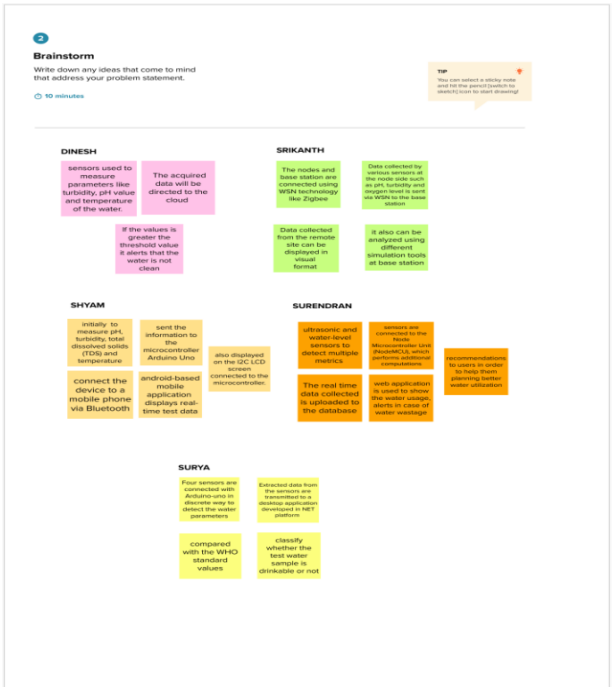
https://app.mural.co/invitation/mural/riverwatermonitoring9467/1667487706579?sender=ue93be257d54d6b3ac1cf1482&key=67c28d3c-a5e2-4aef-8252-95a6cc9a38f3

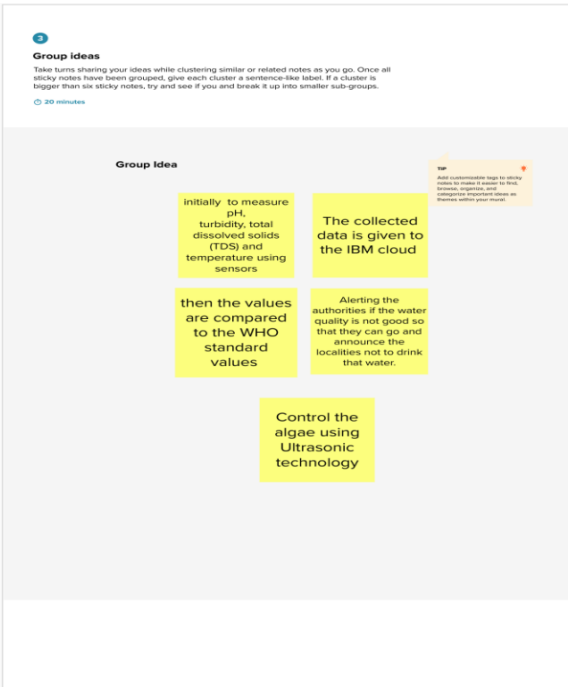
## 3.2 Ideation & Brainstorming

## Step-1: Team Gathering, Collaboration and Select the Problem Statement

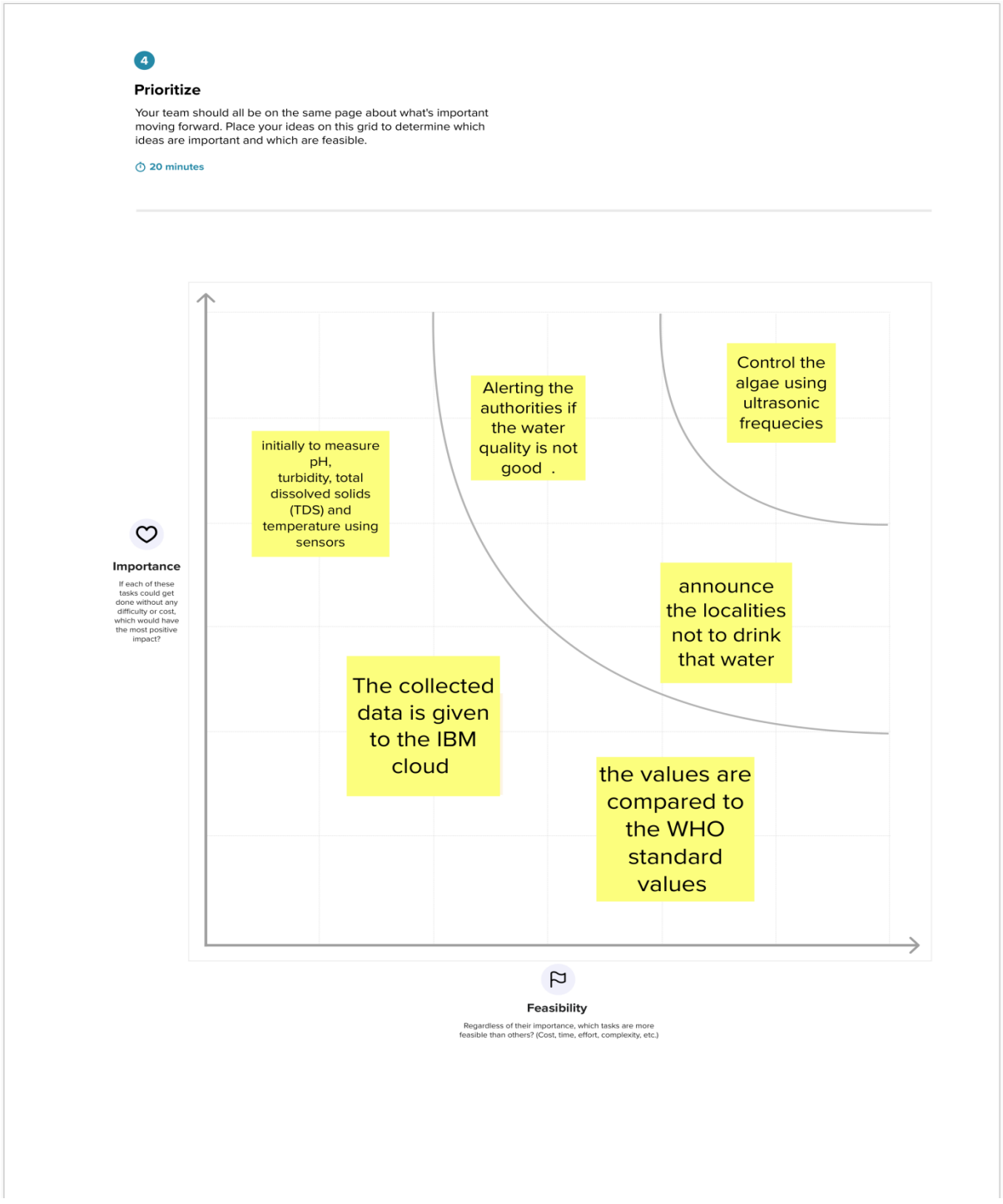



## Step-2: Brainstorm, Idea Listing and Grouping



## Step-3: Idea Prioritization




**Reference:**

https://app.mural.co/invitation/mural/igce0824/1667463328714?sender=ue93be257d54d6b3ac1cf1482&key=8f50e754-280d-49d7-9a9b-6a370a381f79

## 3.3 Proposed Solution

| S. No. | Parameter | Description |
|---|---|---|
| 1. | Problem Statement (Problem to be solved) | To **control the algae** and monitor the value of **PH, Turbidity** present in the river water. |
| 2. | Idea / Solution description | The system consists of more **number of sensors to monitor the PH, Turbidity** and etc., and **control the algae by using ultrasonic frequencies**. |
| 3. | Novelty / Uniqueness | Controlling the algae by using the **ultrasonic technology**. |
| 4. | Social Impact / Customer Satisfaction | River pollution can impact all living things. Better controlling and monitoring can impact clean water and healthy. |
| 5. | Business Model (Revenue Model) | River water controlling and monitoring model. |
| 6. | Scalability of the Solution | It is easy to implement. |

### 3.4 Problem Solution fit

<table>
<tr>
<td>1.Customer Segment(CS)<br><br>Peoples who use the river water</td>
<td>2.Customer Constraints<br><br>Available of devices.</td>
<td>3. Available Solution<br>• IOT based think speak application for monitoring the quality of the water.<br>• IOT based smart water quality monitoring system.<br>• IOT cloud based water conservation and monitoring system.</td>
</tr>
<tr>
<td>4.Jobs-to-be-Done<br>The sensor in the system measures the PH and temperature of the river water. Then the values are compared to the standard values. If it is greater than standard values it alerts the consent authorities.</td>
<td>5. Problem Root Cause<br><br>The problem arises naturally.</td>
<td>6.Behaviour<br><br>After the alert, the people can use water from other till the issue solved by the corporation.</td>
</tr>
<tr>
<td>7.Triggers<br>If the sensors are damaged, the user will not know how to rectify.<br><br>8.Emotions<br>Before, People using the water with a fear and now they are only using the clean water</td>
<td colspan="2">9. Your Solution<br>To control the algae and to monitor the water parameters like PH and temperature in the river water and then alert the consent authorities and the local authorities.<br><br>10.Channels of Behaviour<br>In online service, customer may need to install the mobile application for that.<br><br>In offline service, customer has need to travel and report the issue.</td>
</tr>
</table>

# CHAPTER 4

# REQUIREMENT ANALYSIS

## 4.1 Functional requirement

| FR No. | Functional Requirement (Epic) | Sub Requirement (Story / Sub-Task) |
|---|---|---|
| FR-1 | User Registration | Registration through Form<br>Registration through Gmail |
| FR-2 | User Confirmation | Confirmation via Email<br>Confirmation via OTP |
| FR-3 | PH sensor | PH sensor are used to detect the PH value of the water |
| FR-4 | Turbidity sensor | Turbidity sensors are used to detect the turbidity value of the water. |
| FR-5 | Thermistors or Thermocouples | Thermistors or Thermocouples are used to detect the temperature of the water. |
| FR-6 | Ultrasonic signal generator | Ultrasonic signal generator generates ultrasonic signal to destroy the algae present in the water. |
| FR-7 | Mobile Application | To give the alerts to the corporation and the local authorities. |

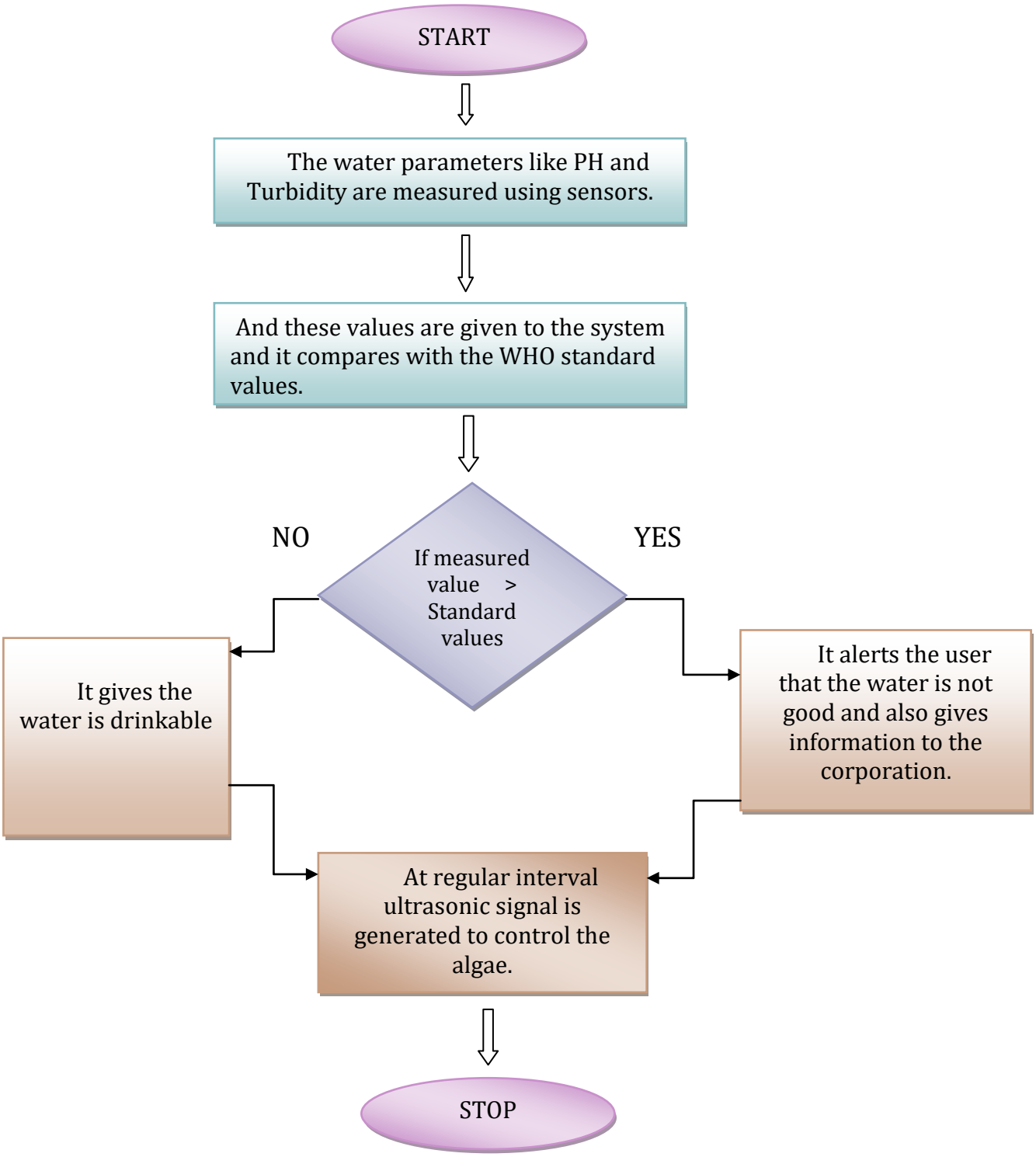
## 4.2 Non-Functional requirements

| NFR No. | Non-Functional Requirement | Description |
|---|---|---|
| NFR-1 | Usability | It is good and efficient to use. |
| NFR-2 | Security | It has high security. |
| NFR-3 | Reliability | Quality assurance, quality control and quality assessment procedures have been implemented. |
| NFR-4 | Performance | The performance of the system good and efficient. |
| NFR-5 | Scalability | The ability of the system is highly scalable. |

# CHAPTER 5

# PROJECT DESIGN

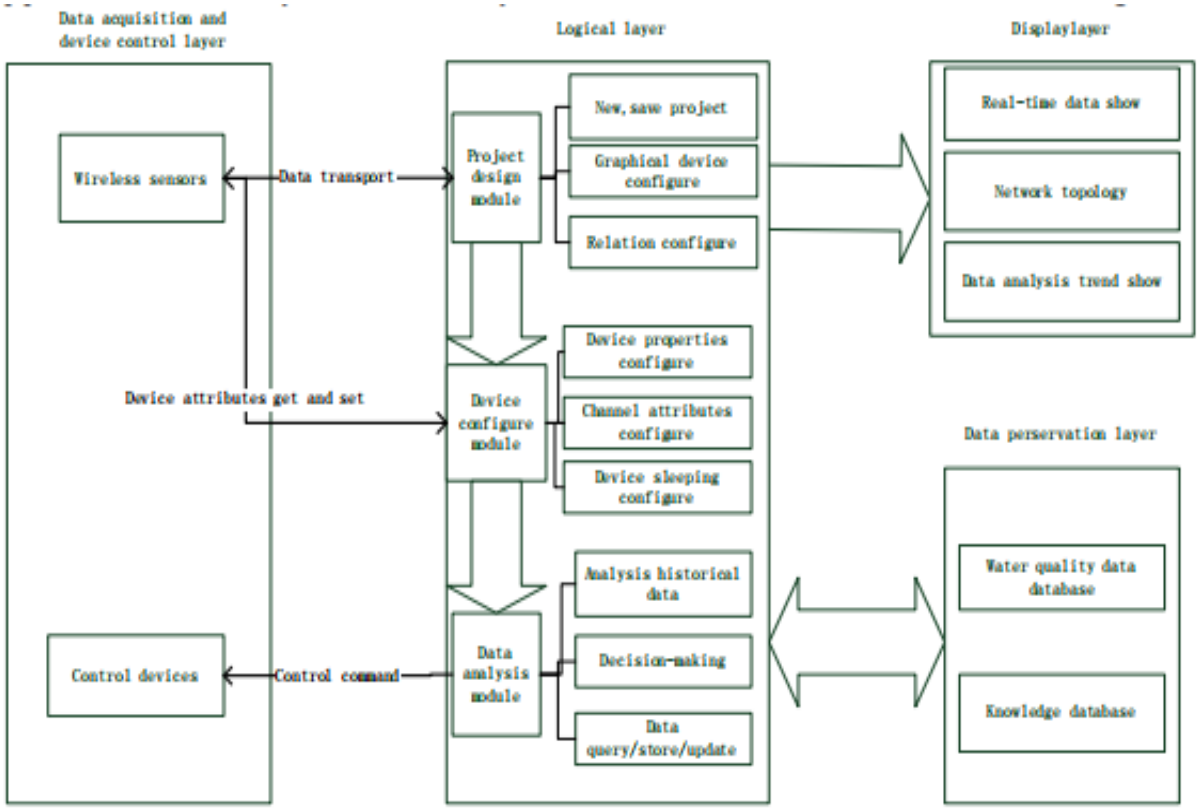
## 5.1 Data Flow Diagrams



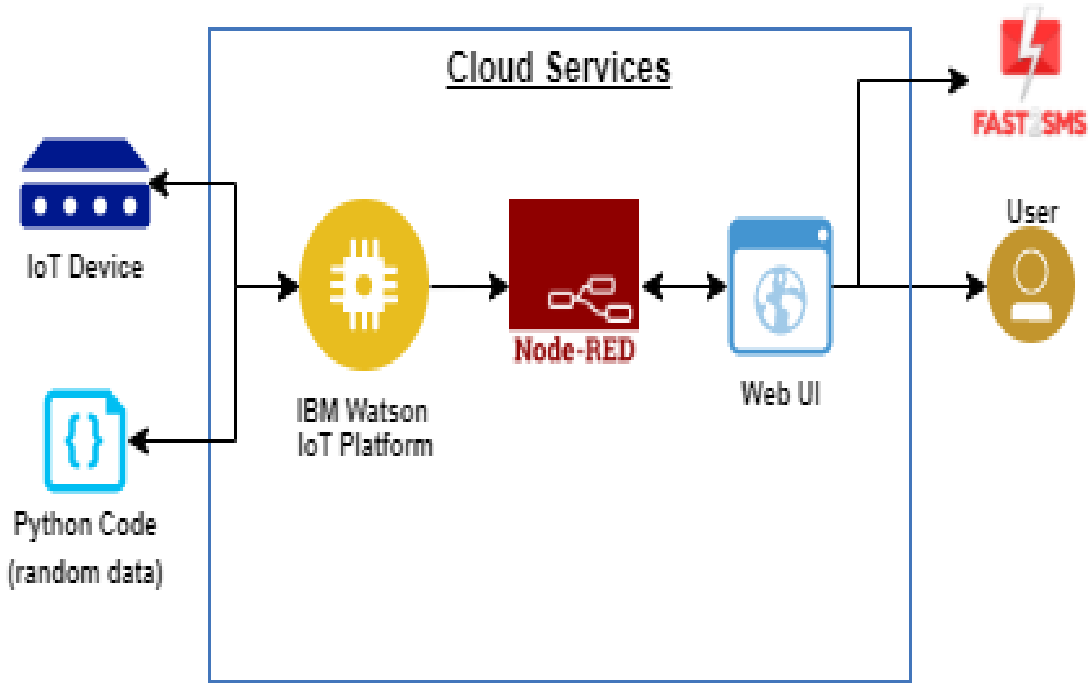
## 5.2 Solution & Technical Architecture

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.




**Solution Architecture Diagram**




**Technical Architecture**

## Components and Technology Table:

| S. No | Component | Description | Technology |
|---|---|---|---|
| 1. | Mobile Application | To give the alerts to the corporation and the local authorities. | SMS service |
| 2. | Web Application | access the data from the cloud | Web UI (using node red service) |
| 3. | PH sensor | detect the PH value of the river water | PH level monitoring |
| 4. | Turbidity sensor | Detect the turbidity level of the water | Turbidity level monitoring |
| 5. | ESP32 | To process the sensed data from the sensors | IBM Watson |

### 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 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 register through the mail | Medium | Sprint-1 |
| | Login | USN-5 | As a user, I can log into the application by entering email & password | I can get login credential | High | Sprint-1 |
| Customer (Web user) | Dashboard | WUSN-1 | As a user, I can login to the web application by using user name and password | I can access my account | High | Sprint-1 |

| User Type | Functional Requirement (Epic) | User Story Number | User Story / Task | Acceptance criteria | Priority | Release |
|---|---|---|---|---|---|---|
| Customer Care Executive | View manner | CCE-1 | As a user, I can see the data in visual view. (graphical representation) | I can easily understand | High | Sprint-1 |
| | Quality | CCE-2 | As a user, I can easily predict the quality of the water | I can easily identify the quality of the water | High | Sprint-1 |
| Administrator | person | Adm-1 | As a admin, I can take all the responsibility about the system | I can monitor the entire system properly | High | Sprint-2 |

# CHAPTER 6

# PROJECT PLANNING & SCHEDULING

## 6.1 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. | 3 | High | Dinesh, Srikanth |
| Sprint-1 | Confirmation | USN-2 | As a user, I will receive confirmation email once I have registered for the application. | 2 | Medium | Dinesh, Srikanth |
| Sprint-1 | Registration using Gmail | USN-4 | As a user, I can register for the application through Gmail. | 2 | Medium | Dinesh, Srikanth |
| Sprint-1 | Login | USN-5 | As a user, I can log into the application by entering email & password. | 3 | High | Dinesh, Srikanth |
| Sprint-2 | IBM cloud | USN-6 | As a user, I can get access to IBM cloud service. | 2 | Medium | Surendran, Shyam |
| Sprint-2 | IBM Watson and device setting | USN-7 | Creating IBM Watson and device setting for integrate the microcontroller to get the sensed data. | 3 | High | Surendran, Shyam |
| Sprint-2 | Node red | USN-8 | To create the Node red service. | 3 | High | Surendran, Shyam |
| Sprint-3 | Create Web UI | USN-9 | To create Web UI to access the data from the cloud. | 3 | High | Srikanth, Surya |
| Sprint-3 | Create web application | USN-10 | To create the web application. | 2 | Medium | Srikanth, Surya |
| Sprint-3 | Source code creation | USN-11 | To create the source code for the project. | 3 | High | Srikanth, Surya |
| Sprint-4 | Publish data | USN-12 | Publish the sensed data to the cloud. | 3 | High | Dinesh, Surya |

| Sprint | Functional Requirement (Epic) | User Story Number | User Story / Task | Story Points | Priority | Team Members |
|---|---|---|---|---|---|---|
| Sprint-4 | SMS | USN-13 | If the sensed values are higher than the standard values it sends the message to the authorities. | 3 | High | Dinesh, Surya |
| Sprint-4 | Testing | USN-14 | Testing the developed project. | 3 | High | Dinesh, Surya |

### 6.2 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 | 10 | 6 Days | 24 Oct 2022 | 29 Oct 2022 | 10 | 29 Oct 2022 |
| Sprint-2 | 10 | 6 Days | 31 Oct 2022 | 05 Nov 2022 | 10 | 05 Nov 2022 |
| Sprint-3 | 10 | 6 Days | 07 Nov 2022 | 12 Nov 2022 | 10 | 12 Nov 2022 |
| Sprint-4 | 10 | 6 Days | 14 Nov 2022 | 19 Nov 2022 | 10 | 19 ov 2022 |

### 6.3 Report from JIRA

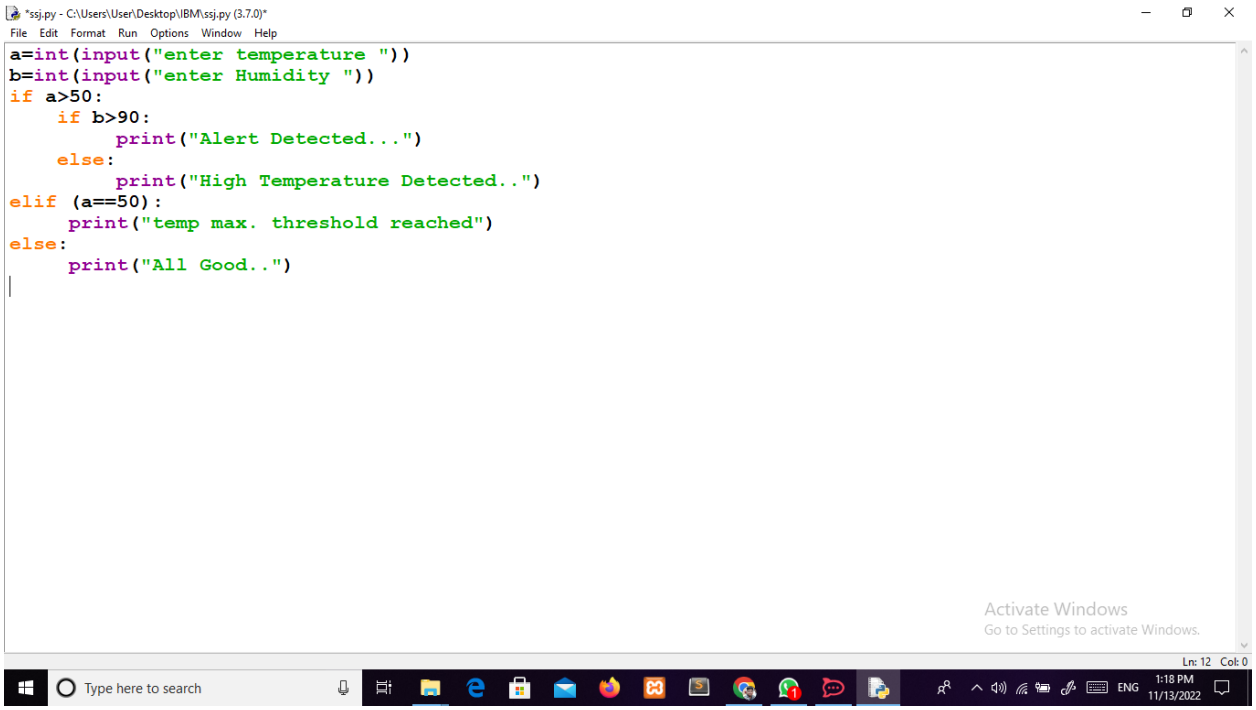
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

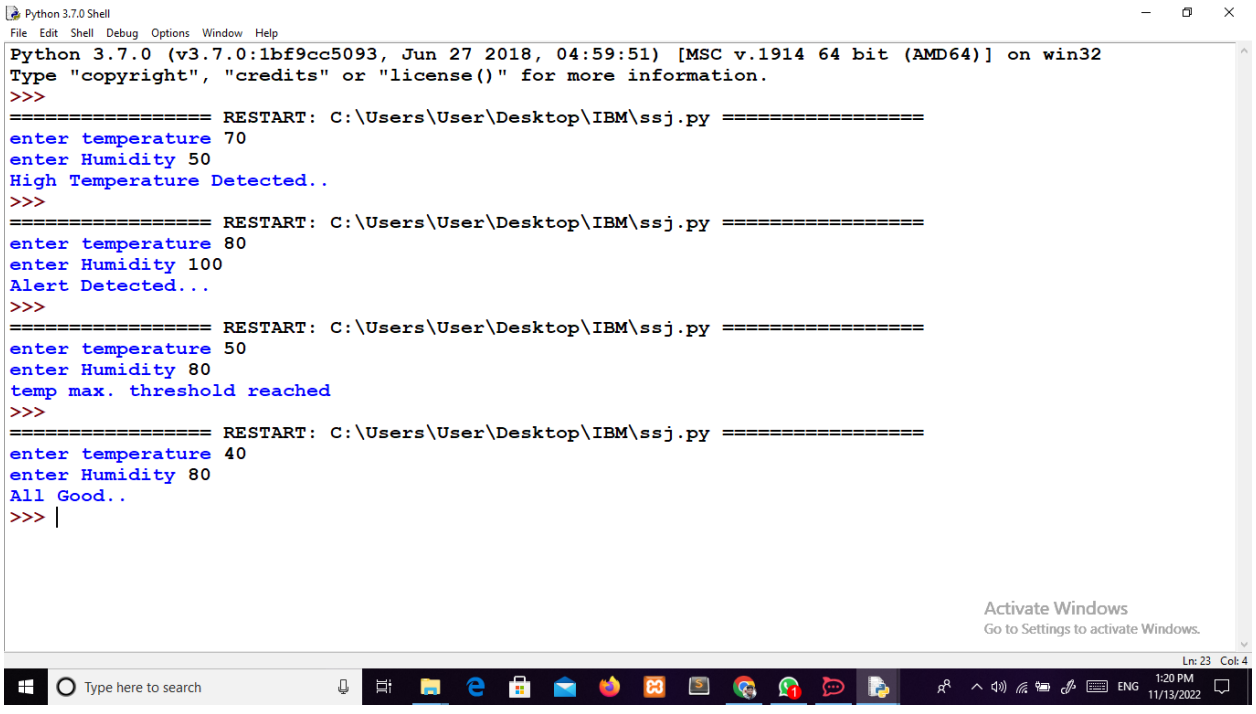
$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

# CHAPTER 7

# CODING AND SOLUTION

## 7.1 Feature 1

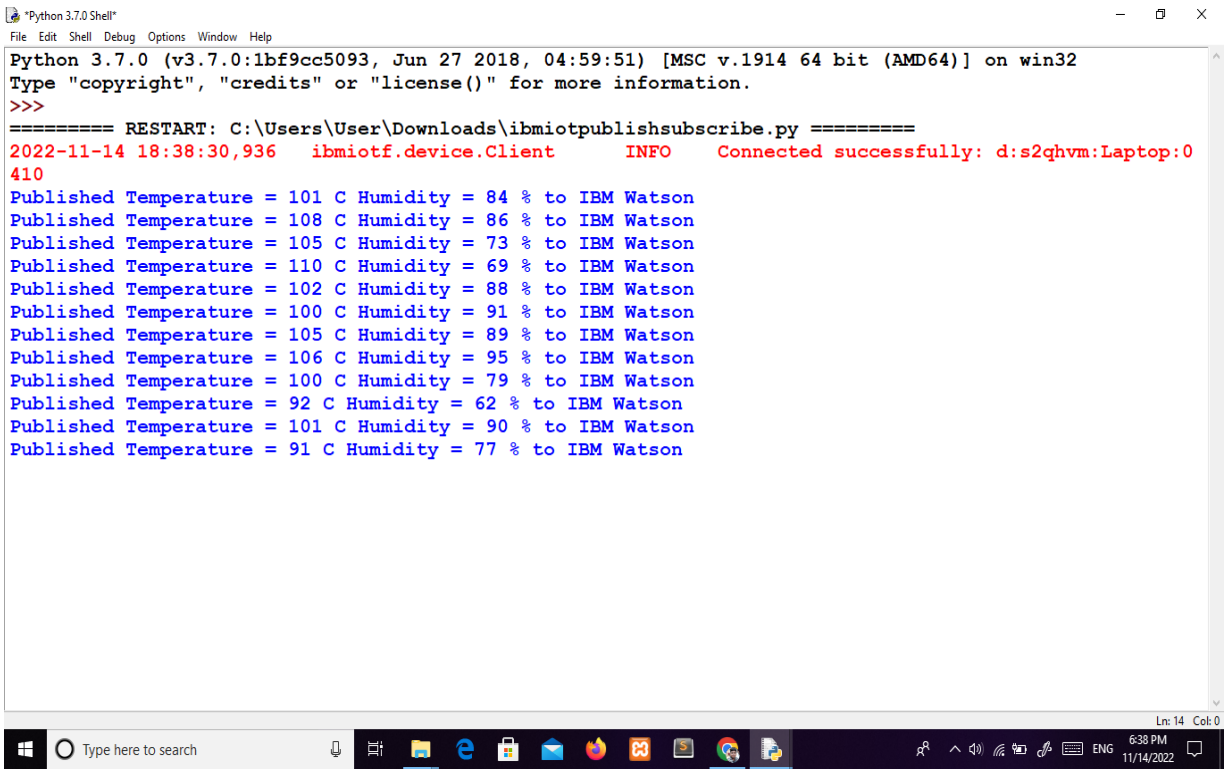


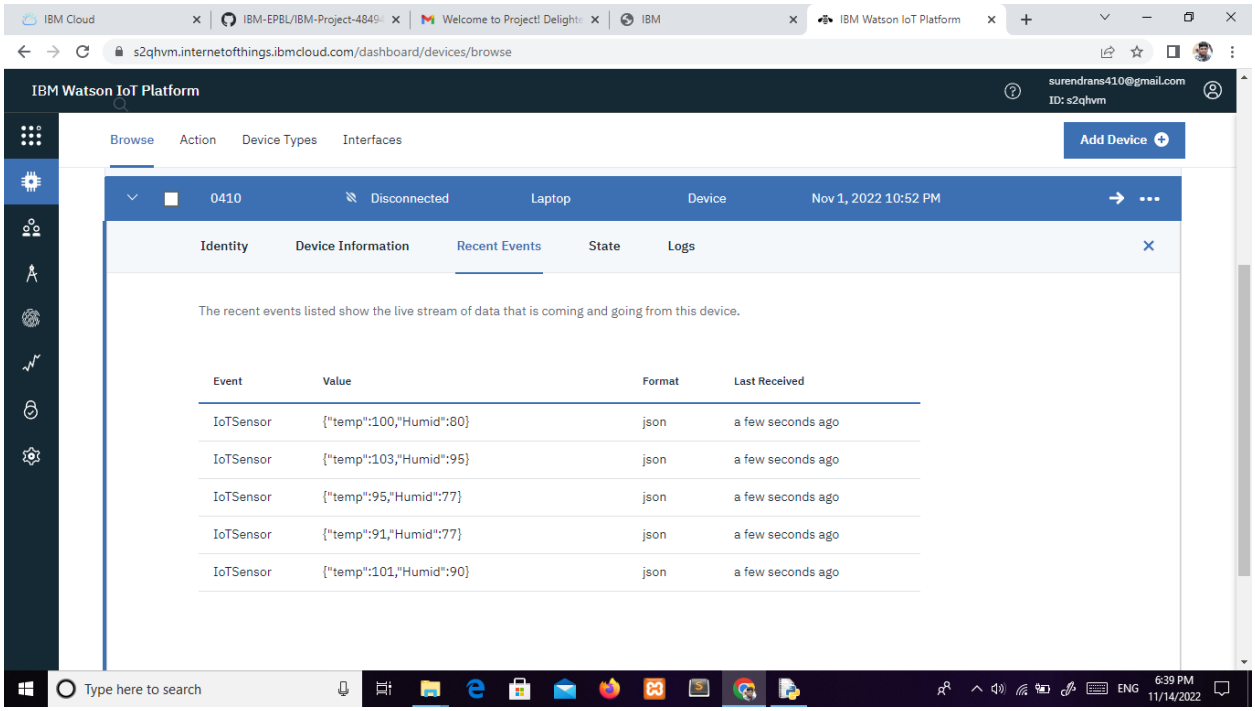



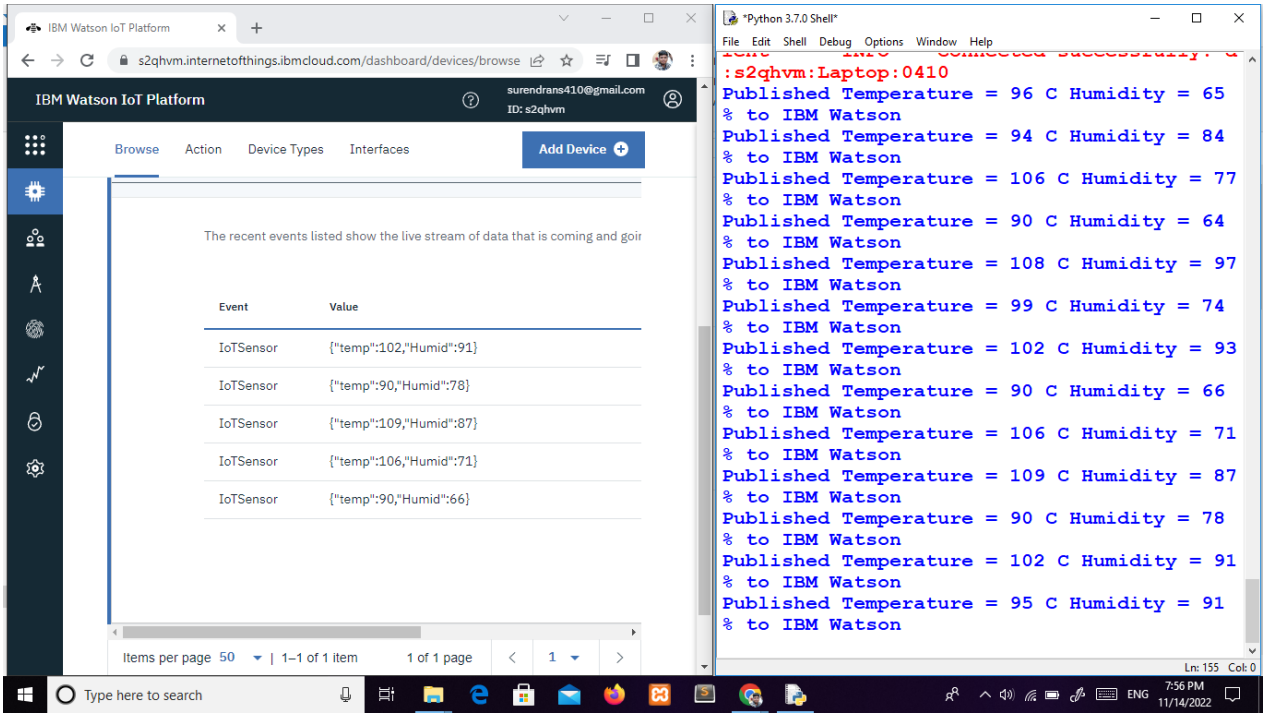
### 7.2 Feature 2

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device Credentials
organization = "s2qhvm"
deviceType = "Laptop"
deviceId = "0410"
authMethod = "token"
authToken = "20011004"

# 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 the proper 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()

# 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(90,110)
        Turbidity=random.randint(60,100)

        data = { 'PH' : PH, 'Turbidity': Turbidity }
        #print data
        def myOnPublishCallback():
           print ("Published PH value = %s C" % 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(1)

        deviceCli.commandCallback = myCommandCallback

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




IBM Watson IoT Platform
s2qhvm.internetofthings.ibmcloud.com/dashboard/devices/browse
surendrans410@gmail.com
ID: s2qhvm
Browse
Action
Device Types
Interfaces
Add Device
0410
Disconnected
Laptop
Device
Nov 1, 2022 10:52 PM
Identity
Device Information
Recent Events
State
Logs
The recent events listed show the live stream of data that is coming and going from this device.
Event
Value
Format
Last Received
IoTSensor {"temp":100,"Humid":80} json a few seconds ago
IoTSensor {"temp":103,"Humid":95} json a few seconds ago
IoTSensor {"temp":95,"Humid":77} json a few seconds ago
IoTSensor {"temp":91,"Humid":77} json a few seconds ago
IoTSensor {"temp":101,"Humid":90} json a few seconds ago
Type here to search
6:39 PM
11/14/2022


IBM Watson IoT Platform
s2qhvm.internetofthings.ibmcloud.com/dashboard/devices/browse
surendrans410@gmail.com
ID: s2qhvm
Browse
Action
Device Types
Interfaces
Add Device
The recent events listed show the live stream of data that is coming and goi
Event
Value
IoTSensor {"temp":102,"Humid":91}
IoTSensor {"temp":90,"Humid":78}
IoTSensor {"temp":109,"Humid":87}
IoTSensor {"temp":106,"Humid":71}
IoTSensor {"temp":90,"Humid":66}
Items per page 50
1–1 of 1 item
1 of 1 page
*Python 3.7.0 Shell*
File Edit Shell Debug Options Window Help
:s2qhvm:Laptop:0410
Published Temperature = 96 C Humidity = 65
% to IBM Watson
Published Temperature = 94 C Humidity = 84
% to IBM Watson
Published Temperature = 106 C Humidity = 77
% to IBM Watson
Published Temperature = 90 C Humidity = 64
% to IBM Watson
Published Temperature = 108 C Humidity = 97
% to IBM Watson
Published Temperature = 99 C Humidity = 74
% to IBM Watson
Published Temperature = 102 C Humidity = 93
% to IBM Watson
Published Temperature = 90 C Humidity = 66
% to IBM Watson
Published Temperature = 106 C Humidity = 71
% to IBM Watson
Published Temperature = 109 C Humidity = 87
% to IBM Watson
Published Temperature = 90 C Humidity = 78
% to IBM Watson
Published Temperature = 102 C Humidity = 91
% to IBM Watson
Published Temperature = 95 C Humidity = 91
% to IBM Watson
Ln: 155 Col: 0
Type here to search
7:56 PM
11/14/2022

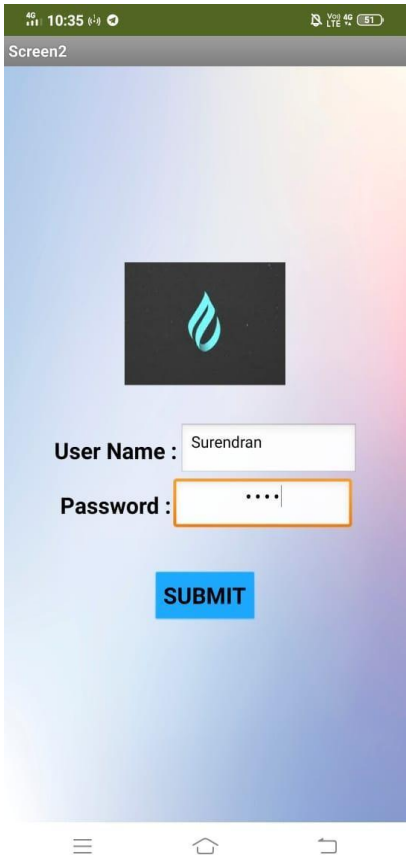
# CHAPTER 8
# TESTING

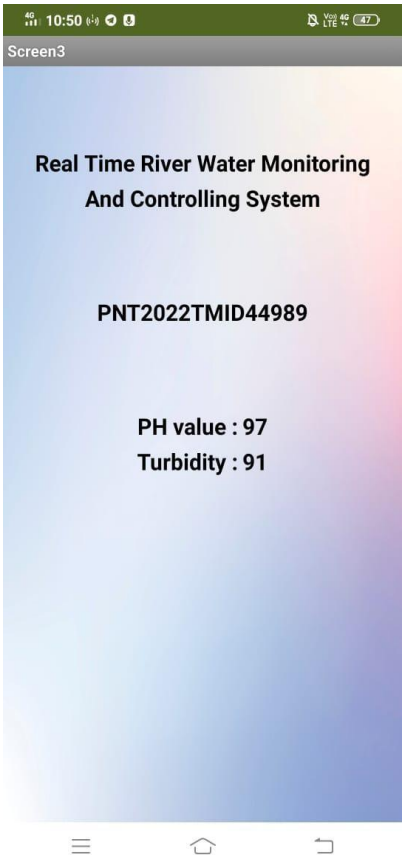
## 8.1 Test Cases

The main benefit of testing is the **identification and subsequent removal of the errors**. However, testing also helps developers and testers to compare actual and expected results in order to improve quality. If the software production happens without testing it, it could be useless or sometimes dangerous for customers.

(i) Performance of the App.
(ii) Analyzing the data from the system.
(iii) Analyze the system performance with the expected performance.

## 8.2 User Acceptance Testing

# CHAPTER 9
# RESULTS

## 9.1 Performance Metrics

- The performance of the system is good and it is easy to find the quality of the water. The performance of the system is achieved by nearly 80% of the expected output of the system.

# CHAPTER 10
# ADVANTAGES AND DISADVANTAGE

**Advantages:**

- Easy detect the quality of the river water.
- Power consumption of the system is low.
- We can give the clean water.
- Easy to access the application.
- People are uses the river water without fear.
- Water pollution can be controlled.

**Disadvantages:**

- Cost of the sensor is high.
- If the problem arises in the system, it cannot be solved by the user.
- Sometimes, sensors give the wrong PH value.
- If the river gets flooded, then the sensors are damaged.
- Need to maintain the system always**.**

# CHAPTER 11
# CONCLUTION

Water turbidity, PH, and temperature are monitored using a water detection sensor that has a unique advantage. The technology can automatically monitor water quality, is low-cost, and does not require personnel to be on duty. As a result, water quality testing will most likely be more cost-effective, convenient, and quick. The method is very adaptable.

This system may be used to monitor different water quality metrics by simply replacing the matching sensors and modifying the required software packages. The procedure is straightforward. The system can be expanded to track hydrologic, air pollution, industrial, and agricultural output, among other things. It is widely used and has a large number of applications. Keeping embedded devices in the environment for monitoring allows the environment to protect itself.

By this Real time river water monitoring and control system, we can easily identity the quality of the river water. Also the people have to use the river water without any Fears.

# CHAPTER 12
# FUTURE SCOPE

The future scope of this project is monitoring environmental conditions, drinking water quality, treatment and disinfection of waste water etc. This system could also be implemented in various industrial processes. The system can be modified according to the needs of the user and can be implemented along with lab view to monitor data on computers.

# CHAPTER 13
# APPENDIX

## 13.1 Source Code

### a. HTML code for registration (UI)

```
 <html>
   <head>
      <h1 style="background_color:darkblue;">IOT</h1>
   </head>
<body>
<p style="background_color:Green;">
<OL>
   <LI>Enter your name</LI>
   <LI>Mail ID</LI>
   <LI>mobile number</LI>
</OL>

<h4><a href="https://project mark.com/">project mark</a></h4>
<form method="post" action="/{{url}}">
   <label for="name">First name:</label><br>
   <input type="text" id="fname" name="fname"><br>
   <label for="mail id">Mail ID:</label><br>
   <input type="text" id="MID" name="MID"><br>
   <label for="mobile number">mobile number:</label><br>
   <input type="text" id="num" name="num"><br>
   <input type="submit" value="submit">
   <input type="reset" value="reset">
</form>
</body>
</html>
```

## b. Arduino code

```
#include <OneWire.h>
#include <DallasTemperature.h>
#define ONE_WIRE_BUS 5
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
float Celcius=0;
float Fahrenheit=0;
float voltage=0;
const int analogInPin = A0;
int sensorValue = 0;
unsigned long int avgValue;
float b;
int buf[10],temp;
void setup(void)
{
        Serial.begin(9600);
         sensors.begin();
         int sensorValue = analogRead(A1);
         voltage = sensorValue * (5.0 / 1024.0);
}
void loop(void)
{
           sensors.requestTemperatures();
          Celcius=sensors.getTempCByIndex(0);
          Fahrenheit=sensors.toFahrenheit(Celcius);
          for(int i=0;i<10;i++)
          {
                buf[i]=analogRead(analogInPin);
                delay(10);
          }
          for(int i=0;i<9;i++)
          {
               for(int j=i+1;j<10;j++)
               {
```

```
                    if(buf[i]>buf[j])
                    {
                      temp=buf[i];
                      buf[i]=buf[j];
                      buf[j]=temp;
                    }
                }
            }
            for(int i=2;i<8;i++)
            avgValue+=buf[i];
            float pHVol=(float)avgValue*5.0/1024/6;
            float phValue = -5.70 * pHVol + 21.34;
            Serial.println(phValue);
            Serial.print("pH");
            Serial.print(" C ");
            Serial.print(Celcius);
            Serial.print(voltage);
            Serial.print("V");
            delay(10000);
}
```

**c. Python code**

```
 import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device Credentials
organization = "s2qhvm"
deviceType = "Laptop"
deviceId = "0410"
authMethod = "token"
authToken = "20011004"

# 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 the proper 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()

# 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(90,110)
        Turbidity=random.randint(60,100)

        data = { 'PH' : PH, 'Turbidity': Turbidity }
        #print data
        def myOnPublishCallback():
            print ("Published PH value = %s C" % 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(1)
```

```
    deviceCli.commandCallback = myCommandCallback

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

### 13.2 GitHup and Project Demo Link

**GitHup Link:**

**https://github.com/IBM-EPBL/IBM-Project-48494-1660808095/**

**Project Demo Link:**

**https://drive.google.com/file/d/108syoMkQY90y-uaZZfPJwiURAAQyNke2/view?usp=drivesdk**