REAL TIME RIVER WATER QUALITY MONITORING AND CONTROL SYSTEM

IBM-NALAYATHIRAN 2022

PNT2022TMID28239

Submitted by

S.NO	NAME	REGISTER NUMBER
1	DELIN REXY	312419106021
2	I.N.GIFTY	312419106039
3	R.G.HARSHINE	312419106046
4	J.HEPHZIBA MERLIN	312419106049

Students Of

ST.JOSEPH'S INSTITUTE OF TECHNOLOGY,CHENNAI

PROJECT GUIDES

Industry Mentor Name: Mr.Bharadwaj

Faculty Mentor Name :Mrs.P. PRINCY MAGDALINE

INDEX

1. INTRODUCTION

1.1 Project Overview:

River Water quality monitoring System

Wireless communication developments are creating new sensor capabilities. The current developments in the field of sensor networks are critical for environmental applications. Internet of Things (IoT) allows connections among various devices with the ability to exchange and gather data. IoT also extends its capability to environmental issues in addition to automation industry by using industry 4.0. As water is one of the basic needs of human survival, it is required to incorporate some mechanism to monitor water quality time to time. Around 40% of deaths are caused due to contaminated water in the world. Hence, there is a necessity to ensure supply of purified drinking water for the people both in cities and villages. Water Quality Monitoring (WQM) is a cost-effective and efficient system designed to monitor drinking water quality which makes use of Internet of Things (IoT) technology. In this paper, the proposed system consists of several sensors to measure various parameters such as pH value, the turbidity in the water, level of water in the tank, temperature and humidity of the surrounding atmosphere. And also, the Microcontroller Unit (MCU) interfaced with these sensors and further processing is performed at Personal Computer (PC). The obtained data is sent to the cloud by using IoT Watson Cloud to monitor the quality of the water.

1.2 Purpose:

Current water quality monitoring system is a manual system with a monotonous process and is very time-consuming. This paper proposes a sensor-based water quality monitoring system. The main components of Wireless Sensor Network (WSN) include a microcontroller for processing the system, communication system for inter and intra node communication and several sensors. Real-time data access can be done by using remote monitoring and Internet of Things (IoT) technology. 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. Therefore, our proposed system will immensely help Bangladeshi populations to become conscious against contaminated water as well as to stop polluting the water.

2. LITERATURE SURVEY

2.1 Existing Problem:

Due to population growth, urbanization ,and climatic change ,competition for water resources is expected to increase, with a particular impact on agriculture, river water. Water will be suitableness to potable water monitoring compound spillage identification done rivers, remote estimation for swimming pools. It holds self-sufficient hubs that unite with the cloud to ongoing water control .The River water needed to be treated before it is used in agriculture feilds,hence the parameters affecting the quality of river-water need to be analysed and to be used for water treatement purpose.

2.2 References:

1. **IoT Based Real-time River Water Quality Monitoring System**

Mohammad Salah Uddin Chowdurya, Talha Bin Emranb , Subhasish Ghosha , Abhijit Pathaka , Mohd. Manjur Alama , Nurul Absara , Karl Anderssonc , Mohammad Shahadat Hossaind

2. **Smart water quality monitoring system with cost-effective using IoT**

Department of Electronics and Communication Engineering Chaitanya Bharathi Institute of Technology, Hyderabad (TS), India

2.3 Problem Statement:

1. Real water is highly polluted in present situation due to suspended materials

2. Farmers put fertilizers and pesticides on their crop so that they grow better but these fertilizers and pesticides can be washed through the soil by rain to end up in the rivers which causes concentration of nitrate and phosphate in the water increases considerably.

3. The massive growth of algae called Eutrophication, which leads to pollution. When the algae die they broken down by the action of bacteria which quickly multiply using up all the oxygen in the water which leads to the death of many animal

4. Some of the time the water has perilous particles or compound blended and broadly useful water purifier can't refine that. Furthermore, checking the nature of water physically in each time is unthinkable.

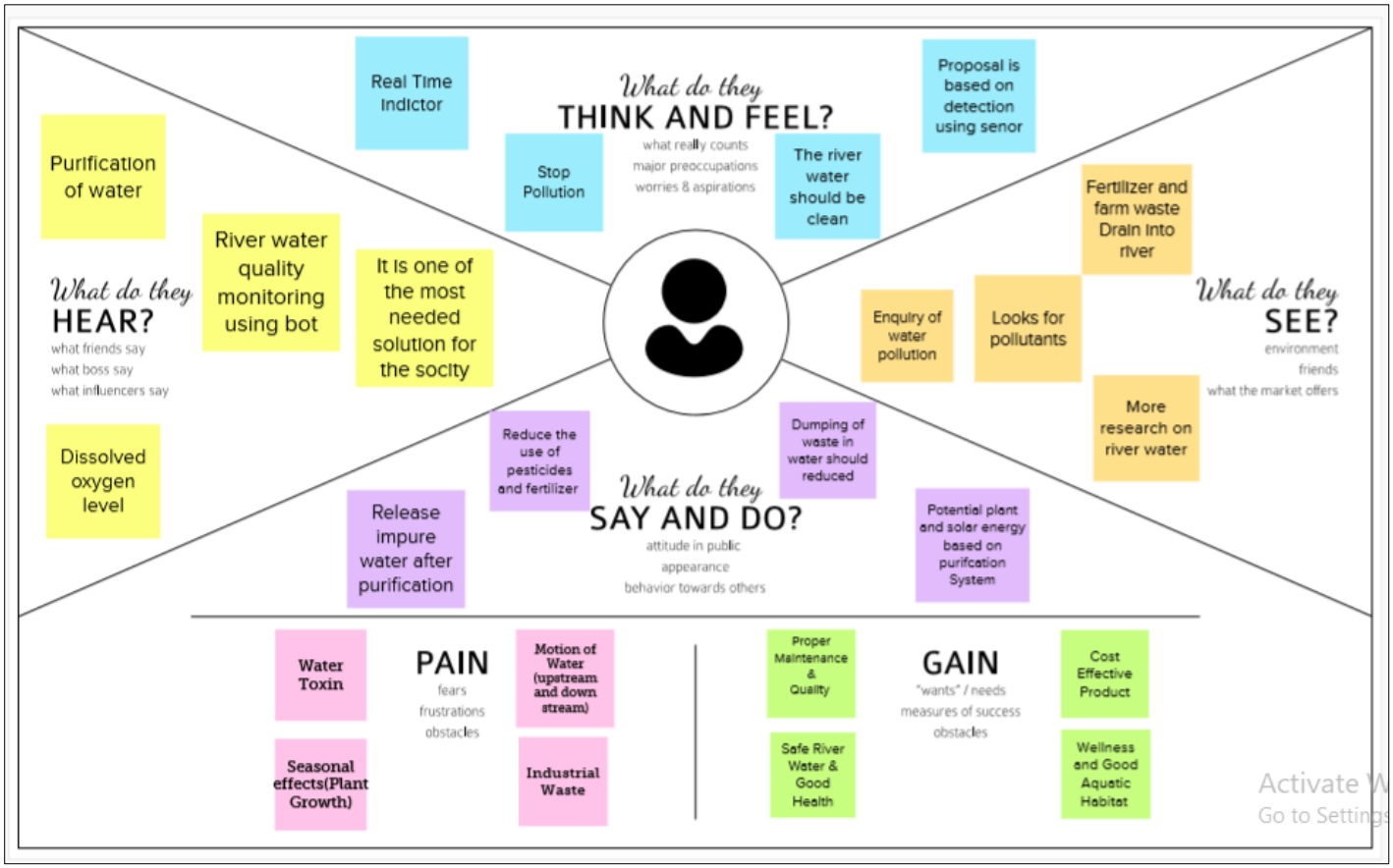
5. The ability to make real-time decisions during critical moments can be vital in preventing expensive repairs and breakdown. Water quality analysis is to measure the required parameters of water, following standard methods, to check whether they are in accordance with the standard. If the water quality is poor and it is below the permissible level it makes decision by comparing it with the previous reading date.

6. By timing update we can save to many lives and even save the livestock like cows etc. by preventing them from drinking river water. So a programmed ongoing checking framework is expected to screen the strength of the water held in our water tank of the general public or loft. So it can caution us consequently assuming that there is any issue with the saved water. What's more, we can check the nature of the water whenever and from anyplace.

IDEATION & PROPOSED SOLUTION

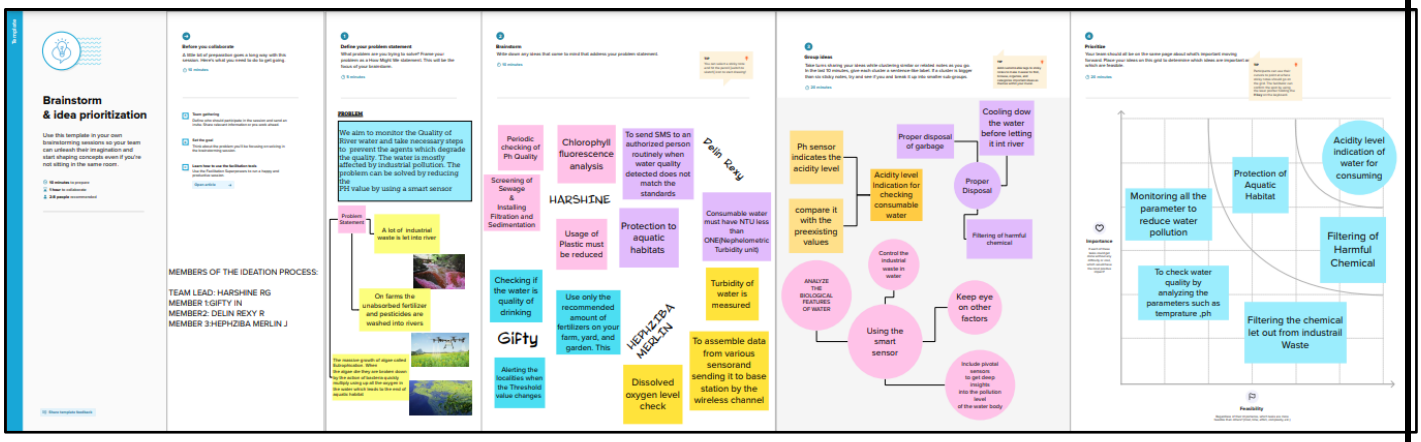
3.1 Empathy Map Canvas:

Various types of parameters are measured with sensors by placing them into different solutions of water. Data generated is compared with standard values in cloud and if exceeds then message sent from cloud to the users mobile. The given paper presents a detailed information of recent works carried out in smart water quality monitoring. Also, a power efficient, simpler solution for in pipe water quality monitoring based on Internet of Things technology is presented. The system developed in this paper is generally used for testing water samples and the data uploaded over the Internet are analyzed. The paper presents a detailed survey on the different techniques implemented in existing smart water quality monitoring systems. Also, a low cost, less complex water quality monitoring system is proposed



3.2 Ideation & Brainstorming:

1.To avoid Water pollution which is one of the biggest and serious threats to society. Water has a significant impact on human health.
2.Safe water is becoming a scarce resource, due to the combined effects of increased population, pollution, and climate changes. Water quality monitoring is thus paramount, especially for domestic water.
3.In-stream water-quality measurements made available on the web in real-time.



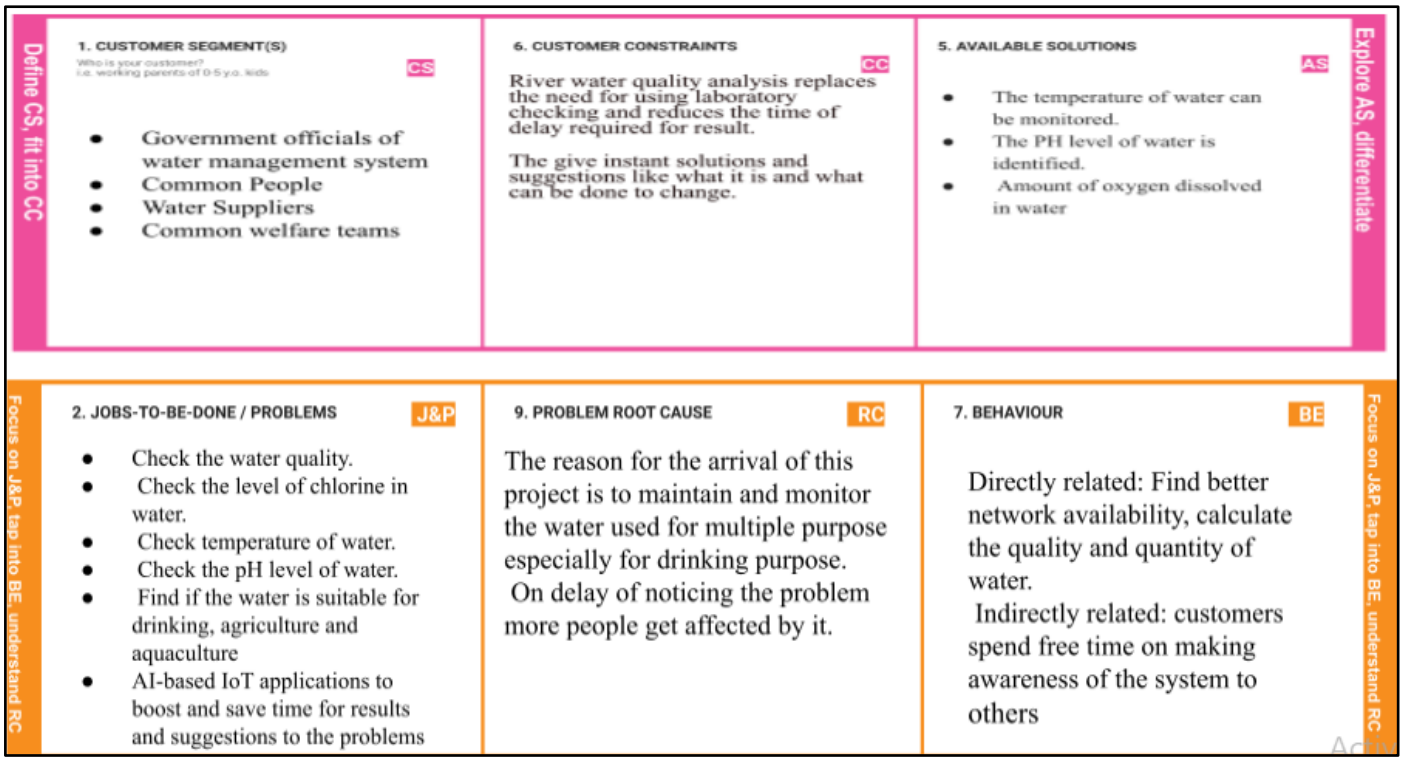
3.3 Proposed Solution:

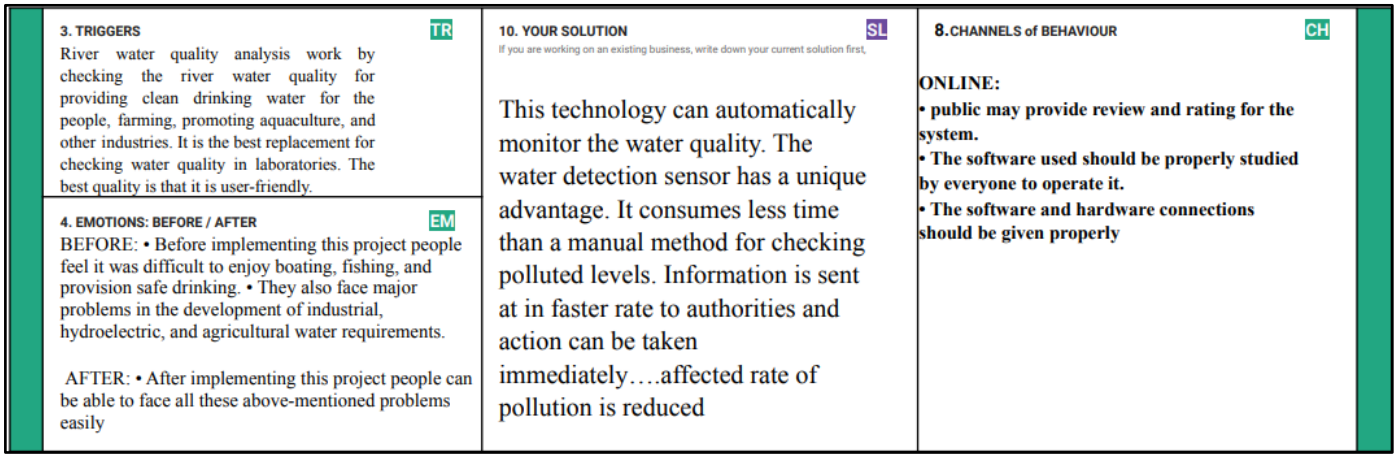
S.No.	Parameter	Description
1.	Problem Statement	River water is polluted by many sources like excess fertilizer from farms, algal blooms, and many industrial wastes are let into the river. So river water becomes polluted. People consume it causes disease and children are suffering from many diseases at young age. So river water is needed continuous quality checking and controlling system.The old manual method is time-consuming and the accuracy rate is less
2.	Idea / Solution description	The proposed method is placing a sensor for ph, conductivity, dissolved oxygen, and turbidity level it is measured at a regular interval of time. If the level from the sensor is above the threshold value should be sent out to water treatment plants. Checking where the level changes high whether it is from factories or farms it should be taken into action for improper letting of water without filtration.Algal blooms formed due to excess fertilizer from farms can be controlled using ultrasonic frequencies.Cloud

		storage is available for storing the collected data from various places on rivers. Using the GSM module message can be sent to authorities of river water management on level changes in the quality of river water .Real-time data access can be done using remote monitoring and IoT technology .A web application is created that is connected with cloud storage . Users can access a web application to check the water status of an area…it can be displayed in a visual format with help of ML lib and Deep learning Neural network models.
3.	Novelty / Uniqueness	We use water detection sensor has unique advantage.It consumes less time to monitor than a manual method for checking polluted levels and notifies immediately to reduce affected rate of pollution in wat er.
4.	Social Impact /Customer Satisfaction	This technology can automatically monitor the water quality. The water detection sensor has a unique advantage. It consumes less time than a manual method for checking polluted level. Information is sent at in faster rate to authorities and action can be taken immediately. Affected rate of pollution is reduced
5.	Business Model (Revenue Model)	This method is very adaptable. As a result, water quality testing will be more cost-effective, convenient, and quick. It can make use of water for drinking and other purposes.

6.	Scalability of the Solution	Can be used to monitor the water quality of rivers and lakes. Whatever may be the water quality either good or bad the device does not fail to give the result correctly

3.4 PROBLEM SOLUTION FIT:





4 REQUIREMENT ANALYSIS

4.1 Functional Requirements:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through product mobile UI
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Ph level detection	Ph sensor is used to monitor the water quality and the signals are send to Arduino.
FR-4	Turbidity Detection	Turbidity sensor TS-300B measures the turbidity (counter of suspended matter) in the wash water and the signals are send to Arduino..
FR-5	Ultrasonic generator	It provides remote monitoring of water flow channels used for irrigation systems. This solution uses the sensor connected to a microcontroller-based hardware that sends real-time information about the level of water flowing in a channel.

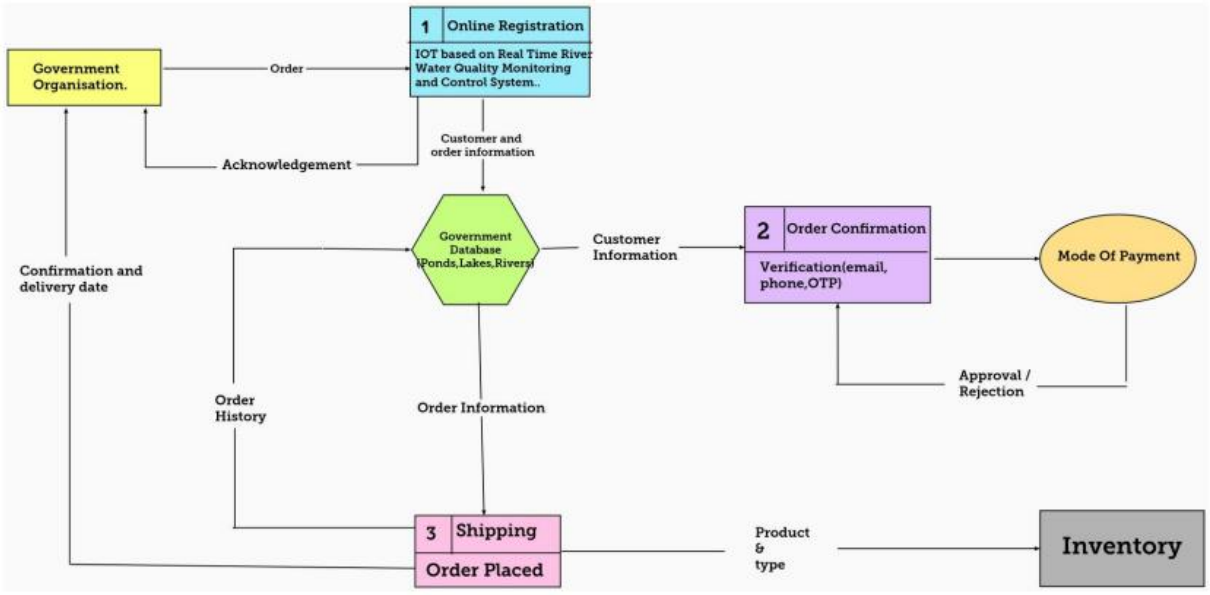
4.2Non-functional Requirements:

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

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Efficient to use and has simple monitoring system.
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 . Assurance for aquaculture safety
NFR-4	**Performance**	Greater performance and environmental safe mode
NFR-5	**Availability**	In 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**	It is highly stable .
NFR-8	**Efficiency**	It is highly efficient and it has simple monitoring system .

5 PROJECT DESIGN

5.1 Data Flow Diagrams:



5.2 SOLUTION AND TECHNICAL ARCHITECTURE

INTRODUCTION:

To assists and monitor the quality of water with the support of information sensed by the sensors dipped in 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 it is used in different fields for collecting, monitoring and analysis of data from remote locations.

SOLUTION REQUIREMENTS:

1. To avoid Water pollution which is one of the biggest and serious threats to society. Water has a significant impact on human health.
2. Safe water is becoming a scarce resource, due to the combined effects of increased population, pollution, and climate changes. Water quality monitoring is thus paramount, especially for domestic water.
3. In-stream water-quality measurements made available on the web in real-time
4. To assemble data from various sensor nodes and send it to the base station by the wireless channel.
5. To send SMS to an authorized person routinely when water quality detected does not match the preset standards, so that, necessary actions can be taken.
6. River water quality can be monitored by the web application.
7. Can be able to know if there are any dust particles present in the water.
8. The PH level of the water can be monitored.
9. Water temperature can be monitored.
10. Alerting the authorities if the water quality is not good so that they can go and announce the localities not to drink that water.

TECHNICAL SOLUTION:

- CDOM/FDOM levels can be measured using electrical optical sensors that use fluorometers and sapphire lens.
- Chlorophyll fluorescence, measured using algae toximeters, indicates the percentage of wetchemical chlorophyll and active chlorophyll in the water sample under illumination.
- By using the technology of Internet of Things, the method of measuring water quality can be done automatically from far off locations via advanced telematics.
- The measured values from the sensors will be processed using a microcontroller, and alert message will be sent to the user via an android application developed using MIT app inventor in case of any abnormalities.

For communication between the measuring and notification nodes, ZigBee receiver and transmitter modules are used in this design. When the water quality parameters reach unsafe levels, the notification node displays the sensor readings and generates an audio alert. The sensors are shown to work within the accuracy ranges that they were designed for. The measurement node can send data to the notification node via ZigBee for audio and visual display. The sensor node is a piece of modular hardware that includes control, communication, and sensor module for measuring pH, conductivity, and temperature. All data is managed by a software platform built with opensource tools, which allows for the control of water pollution and its impact on pisciculture. presents a reconfigurable smart sensor interface device for a water quality monitoring system in an IoT environment. The smart WQM system is comprised of an FPGA design board, sensors, a Zigbee-based wireless communication module, and a personal computer (PC).

SOFTWARE REQUIREMENTS :

Current water quality monitoring system is a manual system with a monotonous process and is very time-consuming.

Due to the fast growing urbanization supply of safe drinking water is a challenge.So these are the following technology is used:

- IOT BASED COMMUNICATION: IoT network technologies to be aware of toward the bottom of the protocol stack include cellular, wifi, and Ethernet, as well as more specialized solutions such as LPWAN, Bluetooth Low Energy (BLE), ZigBee, NFC, and RFID
- IBM WATSON IOT PLATFORM : IBM Watson IoT platform acts as the mediator to connect the web application to IoT device.In order to connect the IoT device to the IBM cloud, we create a device in the IBM Watson IoT platform and get the device credentials
- IBM CLOUD: Python code is used to send random sensor data to the cloud and also to receive commands from the cloud.

REQUIREMENT OF THE PROJECT WORK

The traditional method for monitoring of the water quality is such that the water sample is taken and sent to the laboratory to be tested manually by analytical methods. Although by this method the chemical, physical, and biological agents of the water can be analyzed, it has several drawbacks. Firstly, it is time consuming and labor intensive. Secondly, the cost for this controlled, displayed, and transferred. Compared to the conventional water quality testing techniques, sensor based water quality testing has many advantages such as accurate, high sensitivity, good selectivity, speed, fast response, low cost etc.

METHODOLOGY The system functions automatically and independently according to the code uploaded In this system, three sensors are used to measure the essential water parameters. The four essential water parameters which are temperature, pH level and turbidity can be measured by this proposed system. Sensors circuits are connected to the microcontroller and the probes of the turbidity, pH, and temperature sensors placed inside the water.

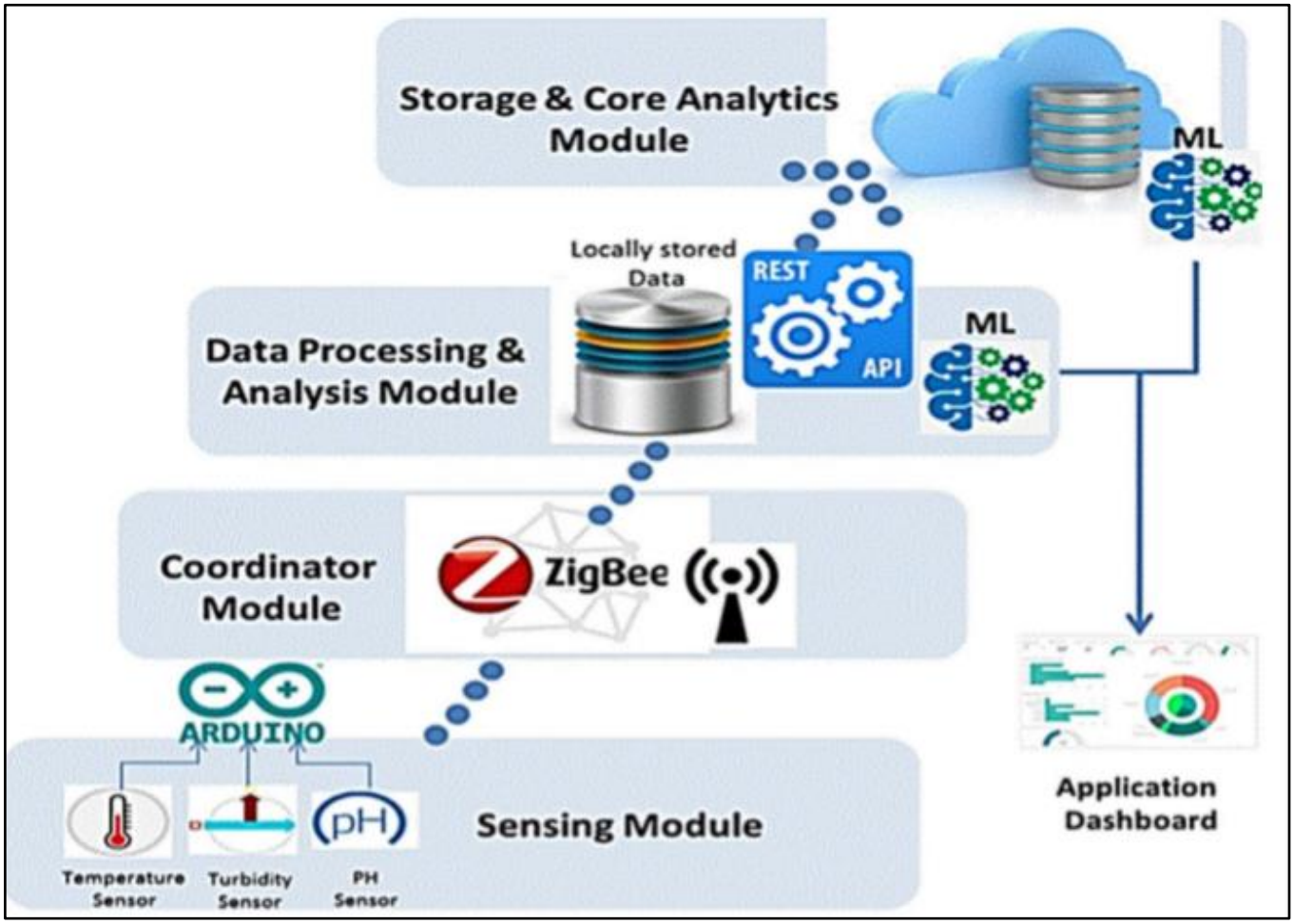
- A water proof temperature sensor is used to avoid any damage or electrical shock to the system and the user.
- An ultrasonic sensor is used to measure the level of the water in thecontainer.

The ultrasonic sensor is connected in the system such that it will be placed on the top of the surface water.The ultrasonic sensor sends electromagnetic waves to the water surface and receives the wave back after touched the water surface. All sensors read the water quality parameters and send the data to the device in the form of electrical signals. In case of any abnormality in a water parameter detected by the device, the buzzer will buzz to indicate that the water is not proper for use. To show the sensor readings (The water parameters) on the device itself, an LCD (Liquid Crystal Display) screen is used. The LCD screen is connected to the project, and through the wired connection, it receives the sensor readings and displays them accordingly.

CONSTRAINTS OF EXISTING SYSTEMS

1. The cost of analysis is very high.
2. The lab testing and analysis takes some time and hence the lab results does not reflect real time water quality measurement due to delay in measurement.
3. The method is prone to human errors of various forms.
4. It is difficult to collect the water samples from all the area of the water body.
5. The process is time consuming due to slow process of manual data collection from different locations of the water body.

SOLUTION ARCHITECTURE:



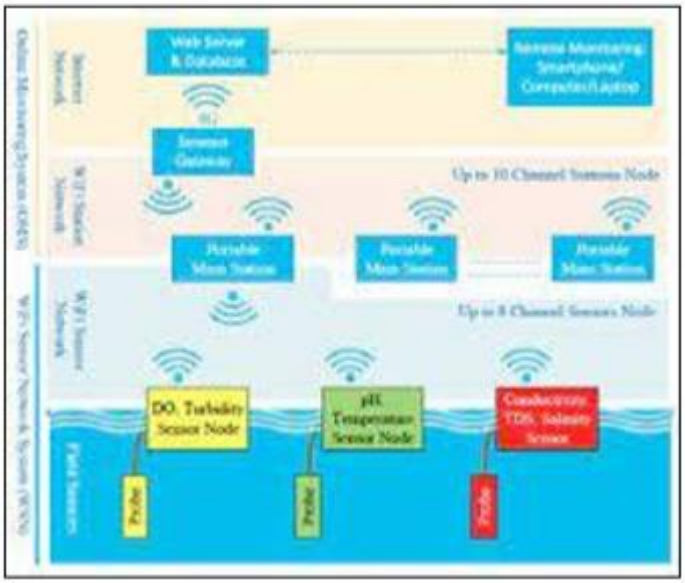
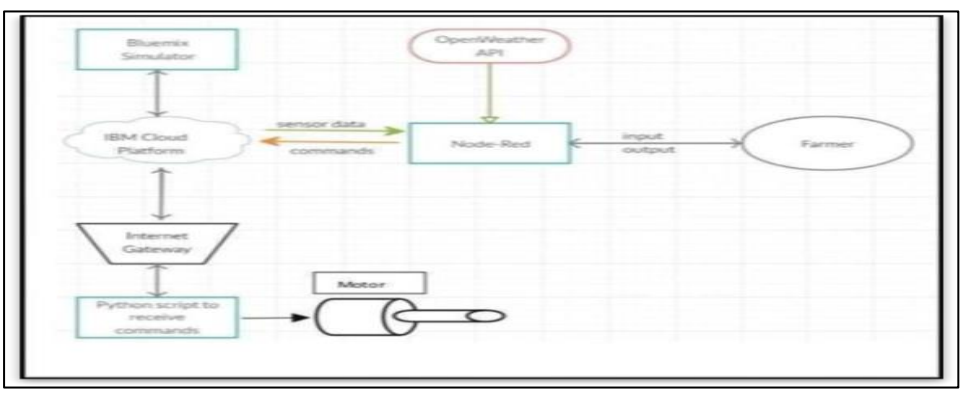


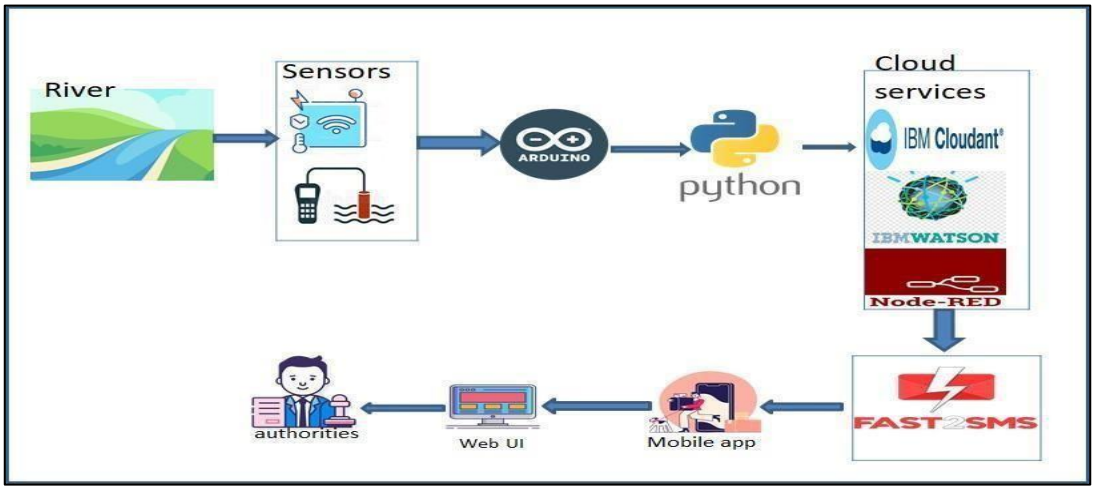
Table-1 : Components & Technologies:

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





User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering email, password, and confirming my password.	They can access my account/dash board	HIGH	SPRINT -1
		USN-2	As a user, I will receive a confirmation email once I have registered for the application	They can receive e confirmation email &click confirm	HIGH	SPRINT -2
		USN-3	As a user, I can register for the application through Google	They can register & access the dashboard with Google	HIGH	SPRINT -1
		USN-4	As a user, I can register for the application through Gmail	They can register through the mail.	MEDIUM	SPRINT -2
	Login	USN-5	As a user, I can log into the application by entering email, password & captcha	They can receive login credentials.	HIGH	SPRINT -1
	Interface	USN-6	As a user, the interface should be user-friendly manner	They can able to access easily.	MEDIUM	SPRINT -1
Customer (Web user)	Dashboard	USN-7	As a user, I can access the specific info(ph value, temp, humidity, quality).	They can able to know the quality of the water	HIGH	SPRINT -1

Customer (input)	View manner	USN-8	As a user, I can view data in visual representation manner(graph)	They can easily understand by visuals.	HIGH	SPRINT -1
	Taste	USN-9	As a user , I can able to view the quality(salty) of the water	They can easily know whether it is salty or not	HIGH	SPRINT -1
	Colour visibility	USN-10	As a user , I can able predict the water colour	Thry can easily know the condition by colour	HIGH	SPRINT -1
Administrator	Risk tolerant	USN-11	An administrator who Is handling the system should update and take care of the application	Admin should monitorthe records properly.	MEDIUM	SPRINT -3

6.PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING & SCHEDULING:

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

6.2 SPRINT DELIVERY SCHEDULE

Product Backlog, Sprint Schedule, and 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	GIFTY IN HEPHZIBA MERLINJ HARSHINE RG DELIN REXY R
	Registration	USN-3	As a user, I can register for the application through Facebook	2	Low	
	Registration	USN-4	As a user, I can register for the application through Gmail	2	Medium	
Sprint -2	Confirmation	USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	
	Login	USN-5	As a user, I can log into the applicationby entering email & password	1	High	
	IBM Cloud serviceaccess		Get access to IBM cloud services.	2	High	
Sprint -3	Create the IBM Watson IoT and device Settings	USN-6	To create the IBM Watson IoT Platform and integrate the microcontroller with it, to send thesensed data on cloud	2	High	HARSHINE RG DELIN REXY R
	Create a node redservice	USN-7	To create a node red service to integrate the IBM Watson along with the Web UI	2	Medium	HARSHINE RG GIFTY IN HEPHZIBA MERLINJ
	Create a Web UI	USN-8	To create a Web UI, to access the datafrom the cloud and display all parameters.	2	Medium	DELIN REXY R GIFTY IN
	To develop a Python code	USN-9	Create a python code to sense thephysical quantity and store data.	2	Medium	HARSHINE RG DELIN REXY R

	Publish Data to cloud.	USN-10	Publish Data that is sensed by the microcontroller to the Cloud	3	High	GIFTY IN HEPHZIBA MERLIN J
Sprint-4	Fast-SMS Service	USN-11	Use Fast SMS to send alert messages once the parameters like pH, Turbidity and temperature goes beyond the threshold	3	High	HARSHINE RG DELIN REXY R GIFTY IN HEPHZIBA MERLIN
	Testing	USN-12	Testing of project and final deliverables	3	Medium	

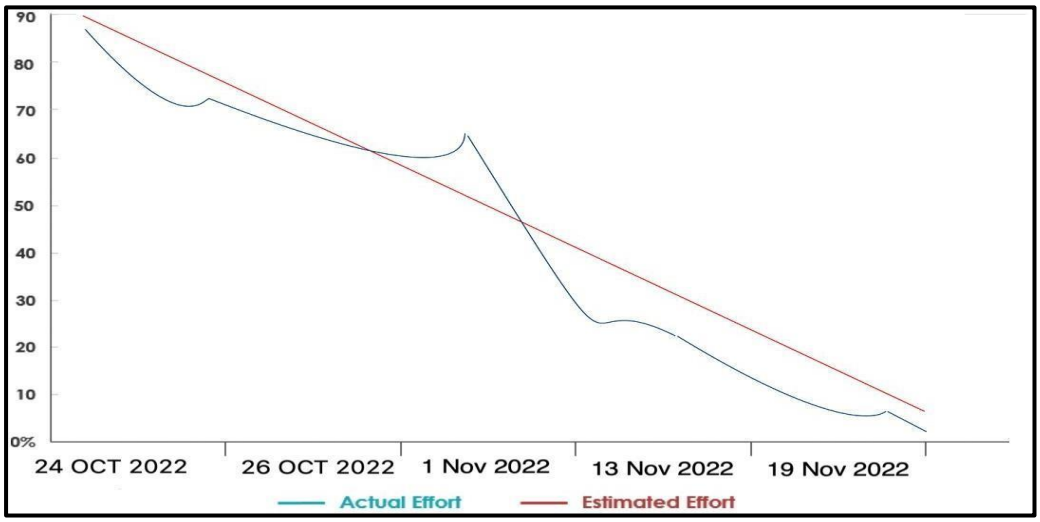
Project Tracker, Velocity & Burndown Charts

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	2 Days	24 Oct 2022	26 Oct 2022	20	29 Oct 2022
Sprint-2	20	4 Days	26 Oct 2022	30 Oct 2022	40	
Sprint-3	20	12 Days	1 Nov 2022	12 Nov 2022	60	
Sprint-4	20	6 Days	13 Nov 2022	19 Nov 2022	80	19 Nov 2022

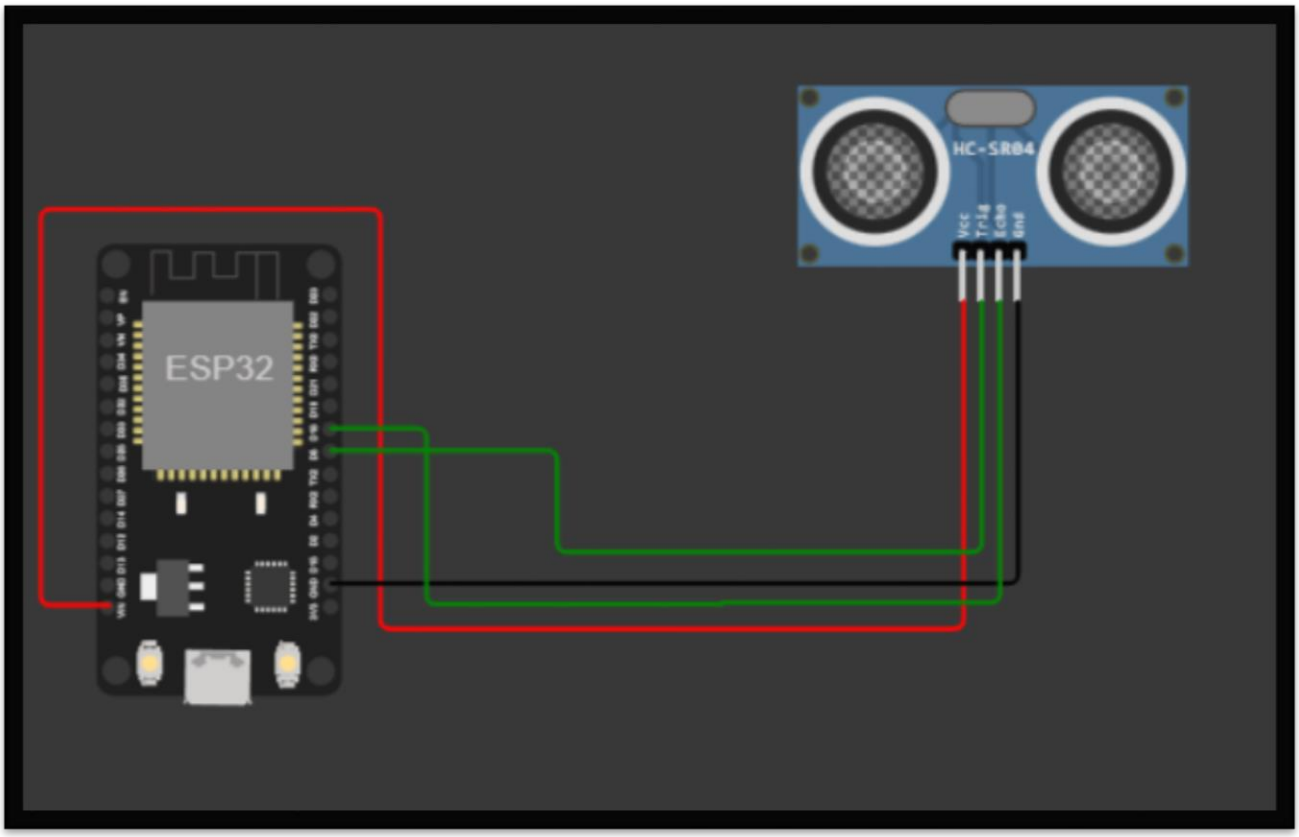
Velocity:

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

Burndown Chart:



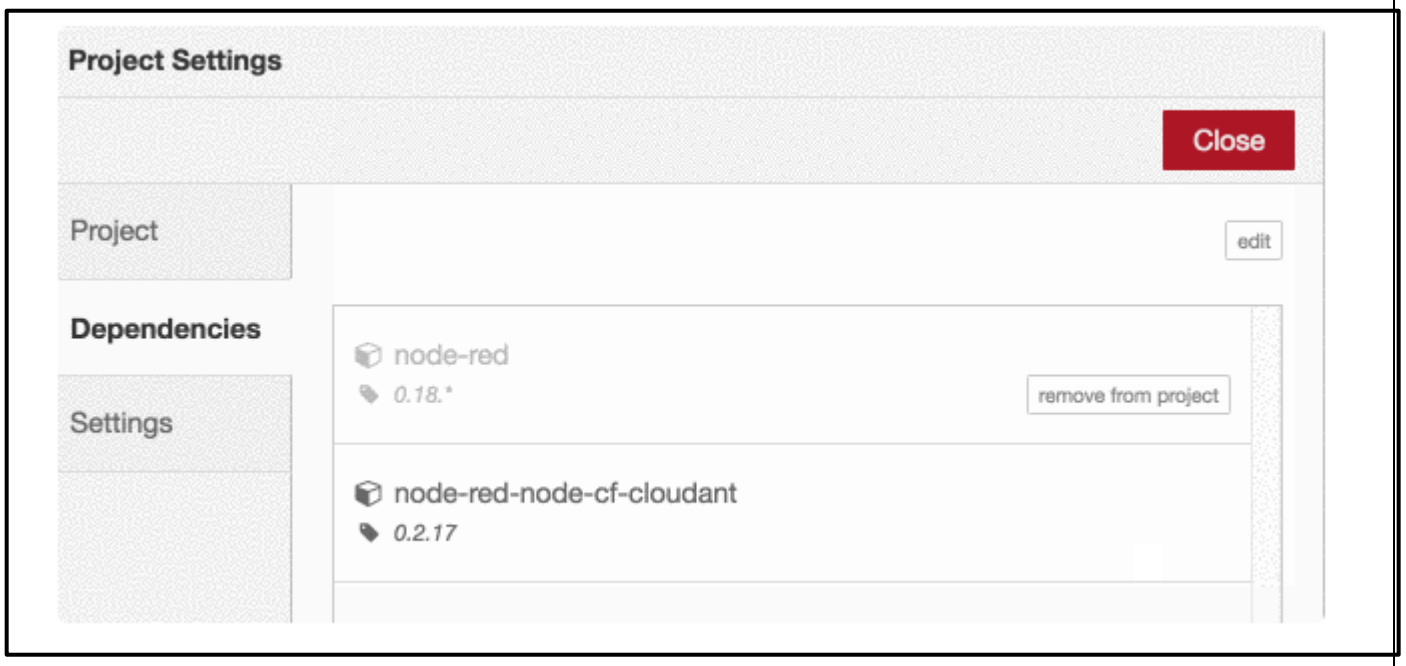
6.3 Reports:




```
ALERT, THE DISTANCE IS LESS THAN 100CM
ALERT, THE DISTANCE IS LESS THAN 100CM
ALERT, THE DISTANCE IS LESS THAN 100CM
```

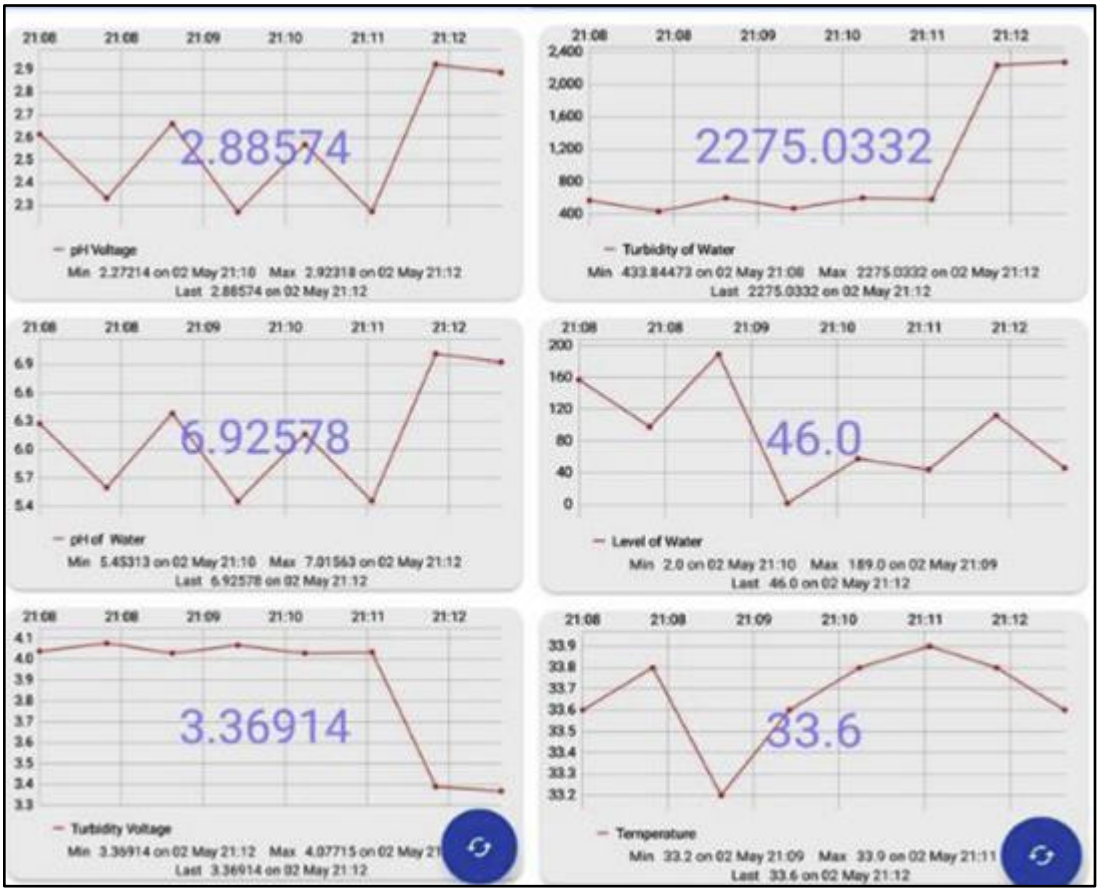
6. CODING AND SOLUTIONING

7.1 NODE RED SERVICE ASSOCIATED WITH IBM CLOUD:




```
#include <ESP8266HTTPClient.h>
#include <FirebaseArduino.h> #include
<DNSServer.h>
#include <ESP8266WiFi.h> #include
<ESP8266WebServer.h>#include
<OneWire.h>
#include <TimeLib.h> //library to get time and date#include
<WiFiUdp.h>
#include <OneWire.h>

#define StartConvert 0
#define ReadTemperature 1
#define ecSwitch D6 #define
tempSwitch D7 #define
turbiditySwitch D8

/* EC and Temp */
const byte numReadings = 20;              //the number of sample times byte
ECsensorPin = A0;  //EC Meter analog output,pin on analog 1byte
DS18B20_Pin = D2; //DS18B20 signal, pin on digital 2 unsigned int
AnalogSampleInterval=25,printInterval=700,tempSampleInterval=850 unsigned int
readings[numReadings];
byte indx = 0;
unsigned long AnalogValueTotal = 0;
unsigned int AnalogAverage = 0,averageVoltage=0;
```

```
unsigned long AnalogSampleTime,printTime,tempSampleTime;float
temperature,ECcurrent;


void setup()
{
   Serial.begin(115200);
```



```
// connect to wifi using WifiManager library.
WiFiManager wifiManager;
   //wifiManager.autoConnect("AutoConnectAP");
   wifiManager.autoConnect("PureraWater");

   Serial.println(); Serial.print("connected:
   "); Serial.println(WiFi.localIP());



   pinMode(MUX_A, OUTPUT); pinMode(MUX_B,
   OUTPUT); pinMode(phSwitch,OUTPUT);
   pinMode(ecSwitch,OUTPUT);
   pinMode(turbiditySwitch,OUTPUT);

   myservo.attach(servoPin);
   Udp.begin(localPort);
   //Serial.print("Local port: ");
   //Serial.println(Udp.localPort());
   //Serial.println("waiting for sync");
   setSyncProvider(getNtpTime);
   setSyncInterval(300);

   Serial.begin(115200);
}

 #include <ESP8266HTTPClient.h>

 #include <FirebaseArduino.h> #include

 <DNSServer.h>

 #include <ESP8266WiFi.h> #include

 <ESP8266WebServer.h>#include

 <WiFiManager.h> #include

 <OneWire.h>

 #include <TimeLib.h> //library to get time and date#include

 <WiFiUdp.h>

 #include <OneWire.h>

 #include <Servo.h>


 // Set these to run example.

 #define FIREBASE_HOST "iot839-a034d.firebaseio.com"#define

 FIREBASE_AUTH
```

```
#define WIFI_SSID "xxx"
#define WIFI_PASSWORD "yyy"

#define StartConvert 0
#define ReadTemperature 1
#define ecSwitch D6 #define
tempSwitch D7 #define
turbiditySwitch D8

/* EC and Temp */
const byte numReadings = 20;             //the number of sample times byte
ECsensorPin = A0;  //EC Meter analog output,pin on analog 1byte
DS18B20_Pin = D2; //DS18B20 signal, pin on digital 2 unsigned int
AnalogSampleInterval=25,printInterval=700,tempSampleInterval=850;  //analogsample
interval;serial print interval;temperature sample interval
unsigned int readings[numReadings];              // the readings from the analog inputbyte
indx = 0;                          // the indx of the current reading
unsigned long AnalogValueTotal = 0;                        // the running total unsigned
int AnalogAverage = 0,averageVoltage=0;                    // the averageunsigned long
AnalogSampleTime,printTime,tempSampleTime;
float temperature,ECcurrent;

//Temperature chip i/o
OneWire ds(DS18B20_Pin);  // on digital pin 2

#define pHsensorPin A0
#define phSwitch D5
unsigned long int avgValueForPH;  //Store the average value of the ph sensorfeedback
int pHbuffer[10],tempValueForPH;float
phValue;
```

/*CLOUD FUNCTIONS*

```
"description": "Cloud Functions for Firebase","scripts": {
    "lint": "eslint .",
    "serve": "firebase serve --only functions", "shell": "firebase
    experimental:functions:shell","start": "npm run shell",
    "deploy": "firebase deploy --only functions","logs": "firebase
    functions:log"
  },
  "dependencies": {
    "firebase-admin": "~5.10.0",
    "firebase-functions": "^0.9.0"
  },
  "devDependencies": {"eslint":
    "^4.12.0",
    "eslint-plugin-promise": "^3.6.0"
  },
  "private": true
```

7. TESTING

7.1 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	10	0	0	10
Client Application	20	0	0	20
Security	5	0	0	5
Outsource Shipping	1	0	0	1
Exception Reporting	15	0	0	15
Final Report Output	10	0	0	10
Version Control	2	0	0	2

7.2 USER ACCEPTANCE TESTING:

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of by the end user or the client to verify/accept the software of Real-Time River Water Quality Monitoring and Control System project at the time of the release to Customers

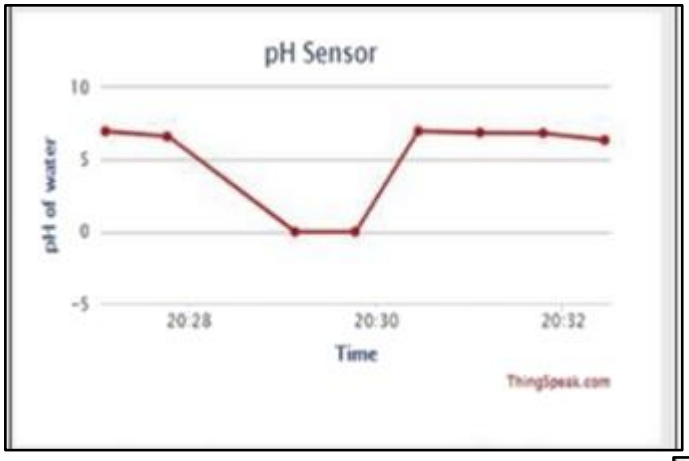
1. Defect Analysis

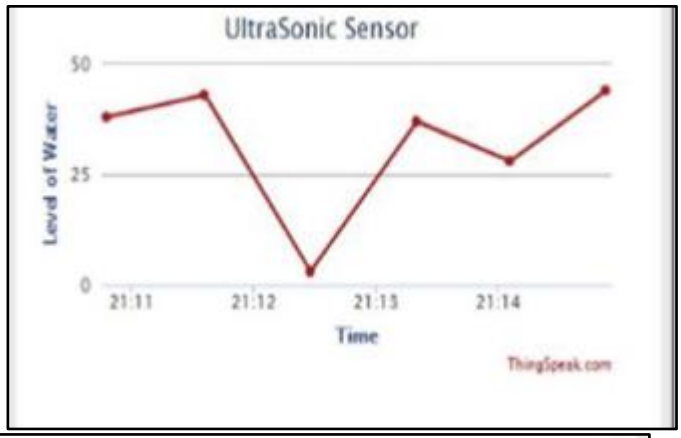
Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	6	6	5	5	22
Duplicate	1	0	1	0	2
External	7	2	1	2	12
Fixed	10	5	5	0	20
Not Reproduced	0	0	0	0	0
Skipped	0	1	1	1	3
Won't Fix	1	0		1	5
Totals	25	13	13	9	64

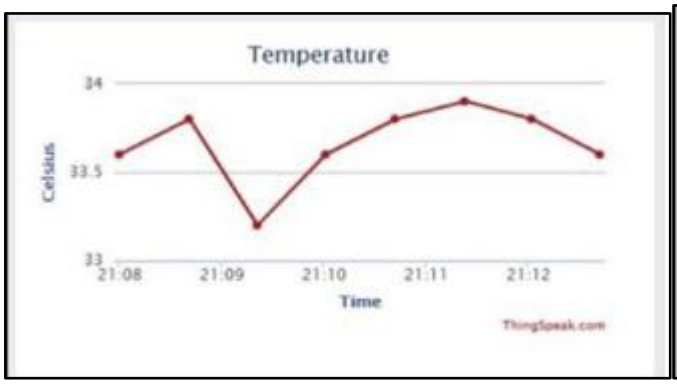
9. RESULT

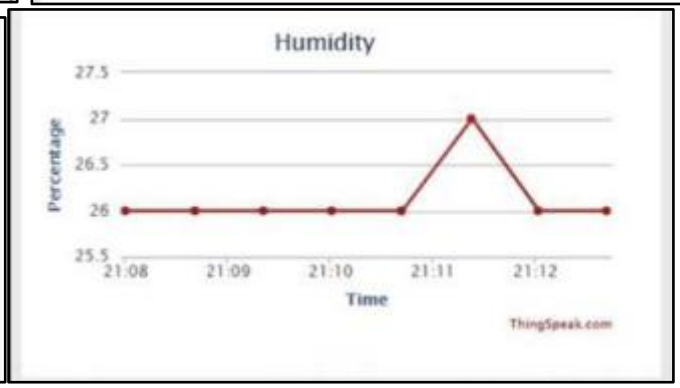
9.1 PERFORMANCE METRICS:

PARAMETER	PERFORMANCE	DESCRIPTION
Admin testing	85%-100%	The testing done before it is Deployed as an app
Customer satisfaction	65-85%	The customer need to be satisfied withthe mobile Application
User interface	55-75%	The app can used by anyone
Server response	20-35%	URL - response
Data validation With no. Of testcase	50-100% 20 Testcase	Valid data from theapp
ERROR	1-2%	Real-time delay May occur









10. ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

- To avoid Water pollution which is one of the biggest and serious threats to society. Water has a significant impact on human health.
- This method is very adaptable. As a result, water quality testing will be more cost-effective, convenient, and quick.
- It can make use of water for drinking and other purposes

DISADVANTAGES:

- The cost of analysis is very high..
- The process is time consuming due to slow process of manual data collection from different locations of the water body.
- The method is prone to human errors of various forms.

11. CONCLUSION

In cities, the population is huge and so are the needs. In today's manual system, we have a tank fitted in very house for water storage which is connected somewhere to the main reservoir or tank. Thus, everyone receives water on a daily basis irrespective of whether it will be used or not. Also, we often tend to forget to close the valve when the tank gets filled. This leads to wastage. What if the tank stops taking water once it is filled? What if a user can decide how much water is to be filled? And that too by the click of a button. This will help the user to have a better control over his/her budget. For eg. if there is no one at home, we can always send a prompt to make the valve remain closed. And too, from anywhere.

A web application is created that is connected with cloud storage . Users can access a web application to check the water status of an area…it can be displayed in a visual format with help of ML lib and Deep learning Neural networksmodels.

This technology can automatically monitor the water quality. The water detection sensor has a unique advantage.

It consumes less time than a manual method for checking polluted level.

Information is sent at in faster rate to authorities and action can be taken immediately.

Affected rate of pollution is reduced

FUTURE SCOPE

1. By leveraging on low-cost computing, the cloud, big data, analytics and mobile technologies, physical devices can be connected to the internet to share and obtain data in real time with minimal human intervention.
2. The water quality measuring system that we have implemented checks the quality of water in real time through various sensors (one for each parameter: pH, conductivity, temperature) to measure the quality of water.
3. We will deploy a wireless sensor network to measure both water level and quality. The observations will then be assimilated into the proposed modeling and prediction framework.
4. An IoT-based water quality monitoring application is beneficial in treating wastewater before it is transferred to freshwater bodies.

13.APPENDIX

13.1 SOURCE CODE:

```
#include <ESP8266HTTPClient.h>
#include <FirebaseArduino.h>
 #include <DNSServer.h>
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
#include <WiFiManager.h>
#include <OneWire.h>
#include <TimeLib.h> //library to get time and date
 #include <WiFiUdp.h>
#include <OneWire.h>
#include <Servo.h>
// Set these to run example.
#define FIREBASE_HOST "iot839-a034d.firebaseio.com" #define
FIREBASE_AUTH
#define WIFI_SSID "xxx"
#define WIFI_PASSWORD "yyy"
#define StartConvert 0
#define ReadTemperature 1
 #define ecSwitch D6 #define tempSwitch D7 #define turbiditySwitch D8
/* EC and Temp */
const byte numReadings = 20;      //the number of sample times byte ECsensorPin
```

```
= A0; //EC Meter analog output,pin on analog 1 byte DS18B20_Pin = D2;
//DS18B20 signal, pin on digital 2 unsigned int
AnalogSampleInterval=25,printInterval=700,tempSampleInterval=850; //analog
sample interval;serial print interval;temperature sample interval
unsigned int readings[numReadings];   // the readings from the analog input byte
indx = 0;    // the indx of the current reading
unsigned long AnalogValueTotal = 0;  // the running total unsigned int
AnalogAverage = 0,averageVoltage=0;         // the average unsigned long
AnalogSampleTime,printTime,tempSampleTime;
float temperature,ECcurrent;
//Temperature chip i/o
OneWire ds(DS18B20_Pin); // on digital pin 2
#define pHsensorPin A0 #define phSwitch D5
unsigned long int avgValueForPH; //Store the average value of the ph sensor
feedback
int pHbuffer[10],tempValueForPH; float phValue;
 #define turbiditysensorPin A0     //turbidity meter Analog output to Arduino
Analog Input 0
unsigned long int avgValueForTurbidity; //Store the average value of the turbidity
sensor feedback
int turbiditybuffer[10],tempValueForTurbidity; float turbidityValue;
#define MUX_A D3 #define MUX_B D4
Servo myservo; #define servoPin D1
/ NTP Servers:
static const char ntpServerName[] = "asia.pool.ntp.org";
const int timeZone = +6; // Convert to Bangladesh Standard Time (BST)
```

```
WiFiUDP Udp;
unsigned int localPort = 8888; // local port to listen for UDP packets
time_t getNtpTime();
void sendNTPpacket(IPAddress &address);
void setup()
{Serial.begin(115200);
// connect to wifi using WifiManager library. WiFiManager wifiManager;
//wifiManager.autoConnect("AutoConnectAP");
wifiManager.autoConnect("PureraWater");
Serial.println(); Serial.print("connected: "); Serial.println(WiFi.localIP());
for (byte thisReading = 0; thisReading < numReadings; thisReading++)
readings[thisReading] = 0;
TempProcess(StartConvert);        //let the DS18B20 start the convert
AnalogSampleTime=millis();
printTime=millis(); tempSampleTime=millis();
pinMode(MUX_A, OUTPUT); pinMode(MUX_B, OUTPUT);
pinMode(phSwitch,OUTPUT); pinMode(ecSwitch,OUTPUT);
pinMode(turbiditySwitch,OUTPUT);
myservo.attach(servoPin); Udp.begin(localPort);
//Serial.print("Local port: ");
//Serial.println(Udp.localPort());
//Serial.println("waiting for sync"); setSyncProvider(getNtpTime);
setSyncInterval(300);
Serial.begin(115200);
}
```

```
/* READING PH */ motorOn(); delay(3000);
digitalWrite(phSwitch, HIGH); //power up ph sensor changeMux(LOW, LOW); //
selector S1=0, S0 = 0; ph setup in Y0 delay(3000);
phRead(); //taking reading
digitalWrite(phSwitch, LOW); //power down ph sensor after reading delay(3000);
//reading complete, now preaparing to take next reading
motorOFF(); delay(3000);
for (int i=0;i<3;i++){ digitalWrite(ecSwitch,HIGH);
 changeMux(LOW, HIGH); // selector S1=0, S0 = 0; ph setup in Y1 delay(3000);
EcAndTempReading(); digitalWrite(ecSwitch,LOW); delay(3000);
}
while(ECcurrent<0){ digitalWrite(ecSwitch,HIGH);
changeMux(LOW, HIGH); // selector S1=0, S0 = 0; ph setup in Y1 delay(3000);
EcAndTempReading(); digitalWrite(ecSwitch,LOW); delay(3000);
}
motorOn(); delay(3000);
/* READING turbidity */
digitalWrite(turbiditySwitch, HIGH); //power up turbidity sensor
changeMux(HIGH, LOW); // selector S1=1, S0 = 0; turbidity setup in Y2
delay(3000);
turbidityRead();//taking reading
digitalWrite(turbiditySwitch, LOW);//power down turbidity sensor after reading
delay(3000);//reading complete, now preaparing to take next reading

StaticJsonBuffer<200> jsonBuffer; JsonObject& root =
jsonBuffer.createObject();
```

```
String date = (String) day()+'/'+month()+'/'+year(); String timee = (String)
hour()+':'+minute()+':'+second(); root["date"] = date;
root["time"] = timee; root["turbidity"] = turbidityValue; root["ph"] = phValue;
root["temp"] = temperature; root["ec"]=ECcurrent; Serial.println("Firebase data:
"); Serial.print("Date: "); Serial.println(date); Serial.print("Time: ");
Serial.println(timee); Serial.print("Turbidity: "); Serial.println(turbidityValue);
Serial.print("PH: "); Serial.println(phValue); Serial.print("Temperature: ");
Serial.println(temperature); Serial.print("EC: "); Serial.println(ECcurrent);
void changeMux(int b, int a) { digitalWrite(MUX_A, a);
 digitalWrite(MUX_B, b);
}
void motorOn(){ myservo.write(180);
}
void motorOFF(){ myservo.write(0);
}
void phRead(){
for(int i=0;i<10;i++)   //Get 10 sample value from the sensor for smooth the value
{
pHbuffer[i]=analogRead(pHsensorPin);
// Serial.println(pHbuffer[i]); delay(10);
}
avgValueForPH=0;
for(int i=2;i<8;i++)     //take the average value of 6 center sample
avgValueForPH+=pHbuffer[i];
phValue=(float)avgValueForPH*3.33/1024/6; //convert the analog into millivolt
Serial.print(phValue,2); Serial.println(" ");
```

```
/* READING PH */motorOn(); delay(3000);
digitalWrite(phSwitch, HIGH); //power up ph sensor changeMux(LOW, LOW); //
selector S1=0, S0 = 0; ph setup in Y0delay(3000);
phRead(); //taking reading
digitalWrite(phSwitch, LOW); //power down ph sensor after readingdelay(3000);
//reading complete, now preaparing to take next reading

motorOFF();delay(3000);
for (int i=0;i<3;i++){ digitalWrite(ecSwitch,HIGH);
changeMux(LOW, HIGH); // selector S1=0, S0 = 0; ph setup in Y1delay(3000);
EcAndTempReading(); digitalWrite(ecSwitch,LOW);delay(3000);
}

while(ECcurrent<0){ digitalWrite(ecSwitch,HIGH);
changeMux(LOW, HIGH); // selector S1=0, S0 = 0; ph setup in Y1delay(3000);
EcAndTempReading(); digitalWrite(ecSwitch,LOW);delay(3000);
}
motorOn(); delay(3000);
/* READING turbidity */
digitalWrite(turbiditySwitch, HIGH); //power up turbidity sensor
changeMux(HIGH, LOW); // selector S1=1, S0 = 0; turbidity setup in
Y2delay(3000); turbidityRead();//taking reading
digitalWrite(turbiditySwitch, LOW);//power down turbidity sensor after reading
delay(3000);//reading complete, now preaparing to take next reading
```

```
StaticJsonBuffer<200> jsonBuffer; JsonObject& root =
jsonBuffer.createObject(); String date = (String) day()+'/'+month()+'/'+year();
String timee = (String) hour()+':'+minute()+':'+second();root["date"] = date;
root["time"] = timee; root["turbidity"] = turbidityValue;root["ph"] = phValue;
root["temp"]
= temperature; root["ec"]=ECcurrent; Serial.println("Firebase data: ");
Serial.print("Date: "); Serial.println(date); Serial.print("Time: ");
Serial.println(timee); Serial.print("Turbidity: "); Serial.println(turbidityValue);
Serial.print("PH: "); Serial.println(phValue); Serial.print("Temperature: ");
Serial.println(temperature); Serial.print("EC: "); Serial.println(ECcurrent);

// append a new value to /logDHT
String name = Firebase.push("/sensor_data", root);delay(500000);//END
TAKING ALL READING
//delay(3000);//END TAKING ALL READING



}

void changeMux(int b, int a) {digitalWrite(MUX_A, a); digitalWrite(MUX_B,
b);
}

void motorOn(){ myservo.write(180);

}
```

```
void motorOFF(){ myservo.write(0);
}

void phRead(){
for(int i=0;i<10;i++)   //Get 10 sample value from the sensor for smooth thevalue
{
pHbuffer[i]=analogRead(pHsensorPin);
// Serial.println(pHbuffer[i]);delay(10);
}
avgValueForPH=0;
for(int i=2;i<8;i++)     //take the average value of 6 center sample
avgValueForPH+=pHbuffer[i];
phValue=(float)avgValueForPH*3.33/1024/6; //convert the analog into millivolt
phValue=4.7*phValue;       //convert the millivolt into pH value Serial.print("
pH:");
Serial.print(phValue,2);Serial.println(" ");
}


void EcAndTempReading(){
/*
Every once in a while,sample the analog value and calculate the average.
*/
if(millis()-AnalogSampleTime>=AnalogSampleInterval)
{
```

```
AnalogSampleTime=millis();
// subtract the last reading:
AnalogValueTotal = AnalogValueTotal - readings[indx];
// read from the sensor:
readings[indx] = analogRead(ECsensorPin);
// add the reading to the total:
AnalogValueTotal = AnalogValueTotal + readings[indx];
// advance to the next position in the array:indx = indx + 1;

// if we're at the end of the array...if (indx >= numReadings)
// ...wrap around to the beginning:
indx = 0;
// calculate the average:
AnalogAverage = AnalogValueTotal / numReadings;
}
/*
Every once in a while,MCU read the temperature from the DS18B20 and then
letthe DS18B20 start the convert.
Attention:The interval between start the convert and read the temperature
shouldbe greater than 750 millisecond,or the temperature is not accurate!
*/
if(millis()-tempSampleTime>=tempSampleInterval)
{
tempSampleTime=millis();
temperature = TempProcess(ReadTemperature); // read the current
temperaturefrom the DS18B20
```

```
TempProcess(StartConvert);        //after the reading,start the convert fornext
reading
}
/*
Every once in a while,print the information on the serial monitor.
*/
if(millis()-printTime>=printInterval)
{
printTime=millis();  averageVoltage=AnalogAverage*(float)5000/1024;
//averageVoltage = averageVoltage *12; //to adjustSerial.print("Analog value:");
Serial.print(AnalogAverage);       //analog average,from 0 to 1023Serial.print("
      Voltage:");
Serial.print(averageVoltage); //millivolt average,from 0mv to
4995mVSerial.print("mV ");
Serial.print("temp:");
Serial.print(temperature);    //current temperatureSerial.print("^C    EC:");

float TempCoefficient=1.0+0.0185*(temperature-25.0);
      //temperaturecompensation formula: fFinalResult(25^C) =
fFinalResult(current)/(1.0+0.0185*(fTP-25.0));
float  CoefficientVolatge=(float)averageVoltage/TempCoefficient;
if(CoefficientVolatge<1)
Serial.println("No solution!");      //25^C 1413us/cm<-->about 216mv if the
```

```
voltage(compensate)<150,that is <1ms/cm,out of the range
else if(CoefficientVolatge>3300)
Serial.println("Out of the range!"); //>20ms/cm,out of the rangeelse
{
if(CoefficientVolatge<=448)
ECcurrent=(6.84*CoefficientVolatge)-62.32;       //1ms/cm<EC<=3ms/cm else

if(CoefficientVolatge<=1457) ECcurrent=(6.98*CoefficientVolatge)-125;
//3ms/cm<EC<=10ms/cm else ECcurrent=(5.3*CoefficientVolatge)+2280;
//10ms/cm<EC<20ms/cm
ECcurrent/=1000;       //convert us/cm to ms/cmSerial.print(ECcurrent,2); //two
decimal Serial.println("ms/cm");
}
}
}
/*
ch=0,let the DS18B20 start the convert;ch=1,MCU read the current
temperaturefrom the DS18B20.
*/
float TempProcess(bool ch)
{
//returns the temperature from one DS18B20 in DEG Celsiusstatic byte data[12];
static byte addr[8];
static float TemperatureSum;if(!ch){ if ( !ds.search(addr)) {
Serial.println("no more sensors on chain, reset search!");ds.reset_search(); return
0;
}
```

```
if ( OneWire::crc8( addr, 7) != addr[7]) {Serial.println("CRC is not valid!");
return 0;
}
if ( addr[0]  != 0x10 && addr[0]  != 0x28) { Serial.print("Device is not
recognized!"); return 0;
}

ds.reset(); ds.select(addr);
ds.write(0x44,1); // start conversion, with parasite power on at the end
}
else{
byte present = ds.reset();ds.select(addr); ds.write(0xBE); // Read Scratchpad
for (int i = 0; i < 9; i++) { // we need 9 bytesdata[i] = ds.read();
}
ds.reset_search(); byte MSB = data[1];byte LSB = data[0];
float tempRead = ((MSB << 8) | LSB); //using two's complimentTemperatureSum
= tempRead / 16;
}
return TemperatureSum;
}

void turbidityRead(){
Serial.println("Taking Readings from turbidity Sensor");turbidityValue = 0; for
(int i=0; i<10; i++){
turbidityValue += analogRead(turbiditysensorPin),delay(10),
```

```
}


turbidityValue /= 10;
float turbidityV = turbidityValue/100; turbidityV = round_to_dp(turbidityV, 1);
turbidityValue = turbidityV;
//Serial.print("Turbidity level: ");

//Serial.println(turbidityV);if( turbidityV > 9){ Serial.print("Turbidity Level:
");Serial.println(turbidityV);


// Serial.println("NTU"); Serial.println("Very Clean");
// delay(3000);
}

if( turbidityV >= 8.5 && turbidityV <= 9 ){Serial.print("Turbidity Level: ");
Serial.println(turbidityV);
// Serial.println("NTU");Serial.println("Clean");
}

if(turbidityV >= 6 && turbidityV < 8.5){Serial.print("Turbidity Level: ");
Serial.println(turbidityV);
// Serial.println("NTU");Serial.println("Dirty");
// delay(3000);
}
if( turbidityV < 6){ Serial.print("Turbidity Level: "),Serial.println(turbidityV),
```

```
// Serial.println("NTU"); Serial.println("Very Dirty");
// delay(3000);
}
}

float round_to_dp( float in_value, int decimal_place )
{
float multiplier = powf( 10.0f, decimal_place ); in_value = roundf( in_value *
multiplier ) / multiplier;return in_value;
}

/*-------- NTP code Don't change Anything   */


const int NTP_PACKET_SIZE = 48; // NTP time is in the first 48 bytes
ofmessage byte packetBuffer[NTP_PACKET_SIZE]; //buffer to hold incoming &
outgoingpackets

time_t getNtpTime()
{
IPAddress ntpServerIP; // NTP server's ip address

while (Udp.parsePacket() > 0) ; // discard any previously received packets
Serial.println("Transmit NTP Request");
// get a random server from the pool WiFi.hostByName(ntpServerName,
ntpServerIP); Serial.print(ntpServerName);
Serial.print(": "); Serial.println(ntpServerIP);
```

```
sendNTPpacket(ntpServerIP);uint32_t beginWait = millis();
while (millis() - beginWait < 1500) {int size = Udp.parsePacket();
if (size >= NTP_PACKET_SIZE) { Serial.println("Receive NTP Response");


Udp.read(packetBuffer, NTP_PACKET_SIZE); // read packet into the
bufferunsigned long secsSince1900;
// convert four bytes starting at location 40 to a long integersecsSince1900 =
(unsigned long)packetBuffer[40] << 24; secsSince1900 |= (unsigned
long)packetBuffer[41] << 16; secsSince1900 |= (unsigned long)packetBuffer[42]
<< 8; secsSince1900 |= (unsigned long)packetBuffer[43];
return secsSince1900 - 2208988800UL + timeZone * SECS_PER_HOUR;
}
}


Serial.println("No NTP Response :-("); return 0; // return 0 if unable to get the
time
}

// send an NTP request to the time server at the given addressvoid
sendNTPpacket(IPAddress &address)
{

memset(packetBuffer, 0, NTP_PACKET_SIZE);

// Initialize values needed to form NTP request
```

/*WEBSITE PAGE CREATION*/

<!DOCTYPE HTML>
<html>
<head>
<title>Easy Admin Panel an Admin Panel Category Flat Bootstrap Responsive Website Template | Blank Page :: Real</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<meta name="keywords" content="Easy Admin Panel Responsive web template, Bootstrap Web Templates, Flat Web Templates, Android Compatible web template,
Smartphone Compatible web template, free webdesigns for Nokia, Samsung, LG, SonyEricsson, Motorola web design" />
<script type="application/x-javascript"> addEventListener("load", function() { setTimeout(hideURLbar, 0); }, false); function hideURLbar(){
window.scrollTo(0,1); } </script>
<!-- Bootstrap Core CSS -->
<link href="css/bootstrap.min.css" rel='stylesheet' type='text/css' />
<!-- Custom CSS -->
<link href="css/style.css" rel='stylesheet' type='text/css' />
<!-- Graph CSS -->
<link href="css/font-awesome.css" rel="stylesheet">
<!-- jQuery -->
<!-- lined-icons -->
<link rel="stylesheet" href="css/icon-font.min.css" type='text/css' />

```
<!-- //lined-icons -->
<!-- chart -->
<script src="js/Chart.js"></script>
<!-- //chart -->
<!--animate-->
<link href="css/animate.css" rel="stylesheet" type="text/css" media="all">
<script src="js/wow.min.js"></script>
<script>
new WOW().init();
</script>
<!--//end-animate-->
<!----webfonts--->
<link
href='//fonts.googleapis.com/css?family=Cabin:400,400italic,500,500italic,600,60
0italic,700,700italic' rel='stylesheet' type='text/css'>

<!---//webfonts--->
<!-- Meters graphs -->
<script src="js/jquery-1.10.2.min.js"></script>
<!-- Placed js at the end of the document so the pages load faster -->

</head>

<body class="sticky-header left-side-collapsed" onload="initMap()">
<section>
<!-- left side start-->
<div class="left-side sticky-left-side">
```

```
<!--logo and iconic logo start-->
<div class="logo">
<h1><a href="index.html">Easy
<span>Admin</span></a></h1>
</div>
<div class="logo-icon text-center">
<a href="index.html"><i class="lnr lnr-home"></i> </a>
</div>
<!--logo and iconic logo end-->
<div class="left-side-inner">
<!--sidebar nav start-->
<ul class="nav nav-pills nav-stacked custom-nav">
<li><a href="index.html"><i class="lnr lnr-
powerswitch"></i><span>Dashboard</span></a></li>
<li class="menu-list">
<a href="#"><i class="lnr lnr-cog"></i>
<span>Components</span></a>
<ul class="sub-menu-list">
<li><a href="grids.html">Grids</a> </li>
<li><a href="widgets.html">Widgets</a></li></ul>
</li>
<li><a href="forms.html"><i class="lnr lnr-spell-check"></i>
<span>Forms</span></a></li>
<li><a href="tables.html"><i class="lnr lnr-menu"></i>
<span>Tables</span></a></li>
<li class="menu-list"><a href="#"><i class="lnr lnr- envelope"></i>
<span>MailBox</span></a>
```

```
<ul class="sub-menu-list">
<li><a href="inbox.html">Inbox</a> </li>

<li><a href="compose-mail.html">Compose Mail</a></li>
</ul></li>
<li class="menu-list"><a href="#"><i class="lnr lnr-indent-increase"></i>
<span>Menu Levels</span></a>
<ul class="sub-menu-list">
<li><a href="charts.html">Basic Charts</a> </li>
</ul>
</li><li><a href="codes.html"><i class="lnr lnr-pencil"></i>
<span>Typography</span></a></li>
<li><a href="media.html"><i class="lnr lnr-select"></i> <span>Media
Css</span></a></li>
<li class="menu-list act"><a href="#"><i class="lnr lnr-book"></i>
<span>Pages</span></a>
<ul class="sub-menu-list">
<li><a href="sign-in.html">Sign In</a> </li>
<li><a href="sign-up.html">Sign Up</a></li>
<li><a href="blank_page.html">Blank Page</a></li></ul></li></ul>
<!--sidebar nav end--></div></div>
<!-- left side end--><!-- main content start-->
<div class="main-content main-content2 main-content2copy">
<!-- header-starts -->
<div class="header-section">
<!--toggle button start-->
<a class="toggle-btn menu-collapsed"><i class="fa fa-bars"></i></a>
```

```
<!--toggle button end-->
<!--notification menu start -->
<div class="menu-right">
<div class="user-panel-top">
<div class="profile_details_left">
<ul class="nofitications-dropdown">
<li class="dropdown">
<a href="#" class="dropdown-toggle" data-toggle="dropdown" aria-
expanded="false"><i class="fa fa-envelope"></i><span
class="badge">3</span></a>
<ul class="dropdown-menu"><li>
<div class="notification_header">
<h3>You have 3 new messages</h3>
</div>
</li>
<li><a href="#">

<div class="user_img"><img src="images/1.png" alt=""></div>
<div class="notification_desc">
<p>Lorem ipsum dolor sit amet</p>
<p><span>1 hour ago</span></p>
</div>
<div class="clearfix"></div>
</a></li>
<li class="odd"><a href="#">
<div class="user_img"><img src="images/1.png" alt=""></div>
<div class="notification_desc">
```

```
<p>Lorem ipsum dolor sit amet </p>
<p><span>1 hour ago</span></p>
</div>
<div class="clearfix"></div> </a></li>
<li><a href="#">


<div class="user_img"><img src="images/1.png" alt=""></div>
<div class="notification_desc">
<p>Lorem ipsum dolor sit amet </p>
<p><span>1 hour ago</span></p></div>
<div class="clearfix"></div>
</a></li><li><div class="notification_bottom">
<a href="#">See all messages</a>
</div> </li></ul></li>
<li class="login_box" id="loginContainer">
<div class="search-box">
<div id="sb-search" class="sb-search"><form>
<input class="sb-search-input" placeholder="Enter your search term..."
type="search" id="search">
<input class="sb-search-submit" type="submit" value="">
<span class="sb-icon-search"> </span></form></div>
</div><!-- search-scripts -->
<script src="js/classie.js"></script>
<script src="js/uisearch.js"></script>
<script>new UISearch( document.getElementById( 'sb-search' ) );
</script>
```

```
<!-- //search-scripts -->
</li><li class="dropdown">
<a href="#" class="dropdown-toggle" data-toggle="dropdown" aria-
expanded="false"><i class="fa fa-bell"></i><span class="badge
blue">3</span></a><ul class="dropdown-menu"><li>


<div class="notification_header">
<h3>You have 3 new notification</h3></div></li>
<li><a href="#">
<div class="user_img"><img src="images/1.png" alt=""></div> <div
class="notification_desc">
<p>Lorem ipsum dolor sit amet</p>
<p><span>1 hour ago</span></div>
<div class="clearfix"></div>       </a></li>
<li class="odd"><a href="#">
<div class="user_img"><img src="images/1.png" alt=""></div>
<div class="notification_desc">
<p>Lorem ipsum dolor sit amet </p>
<p><span>1 hour ago</span></p></div>
<div class="clearfix"></div>       </a></li>
<li><a href="#">
<div class="user_img"><img src="images/1.png" alt=""></div>
<div class="notification_desc">
<p>Lorem ipsum dolor sit amet </p>
<p><span>1 hour ago</span></p></div>
<div class="clearfix"></div>       </a></li>
```

```
<li><div class="notification_bottom">
<a href="#">See all notification</a></div>
</li></ul></li>
<li class="dropdown">
<a href="#" class="dropdown-toggle" data-toggle="dropdown" aria-


expanded="false"><i class="fa fa-tasks"></i><span class="badge
blue1">22</span></a>
<ul class="dropdown-menu"><li>
<div class="notification_header"></div>
</li><li><a href="#">
<div class="task-info">
<span class="task-desc">Database update</span><span
class="percentage">40%</span>
<div class="clearfix"></div>
</div>
<div class="progress progress-striped active">
<div class="bar yellow" style="width:40%;"></div></div></a></li>
<li><a href="#"><div class="task-info">
<span class="task-desc">Dashboard done</span>
<span class="percentage">90%</span><div class="clearfix"></div> </div>

<div class="progress progress-striped active">
<div class="bar green" style="width:90%;"></div></div></a></li>
<li><a href="#">
<div class="task-info">
```

```
<span class="task-desc">Mobile App</span>
<span class="percentage">33%</span>
<div class="clearfix"></div>       </div>
<div class="progress progress-striped active">
<div class="bar red" style="width: 33%;"></div></div>



</a></li><li><a href="#">
<div class="task-info">
<span class="task-desc">Issues fixed</span><span
class="percentage">80%</span>
<div class="clearfix"></div>       </div>
<div class="progress progress-striped active"><div class="bar blue"
style="width: 80%;"></div></div></a></li><li>
<a href="#" class="dropdown-toggle" data-toggle="dropdown" aria-
expanded="false">
<div class="profile_img"><span style="background:url(images/1.jpg) no-repeat
center"> </span> <div class="user-name"></div>
<i class="lnr lnr-chevron-down"></i>
<i class="lnr lnr-chevron-up"></i>
<div class="clearfix"></div>       </div>       </a>
<ul class="dropdown-menu drp-mnu"><li>
</div>
<div class="social_icons">
<div class="col-md-4 social_icons-left"><a href="#" class="yui"><i class="fa fa-
facebook i1"></i><span>300<sup>+</sup> Likes</span></a></div>
<div class="col-md-4 social_icons-left pinterest">
```

```
</div>
<div class="col-md-4 social_icons-left twi">
<a href="#"><i class="fa fa-twitter i1"></i><span>500<sup>+</sup>
Tweets</span></a></div>
<div class="clearfix"> </div></div>



<div class="clearfix"></div></div>
</div><!--notification menu end --></div>
<!-- //header-ends -->
<div id="page-wrapper">
<div class="graphs"><div class="error-main">

<h3><i class="fa fa-exclamation-triangle"></i> <span>404</span></h3>
<div class="col-xs-7 error-main-left">
<span>Oops!</span>
<p>The page you're looking for could not be found.</p>
<div class="error-btn">
<a href="index.html">Go back?</a></div></div>
<div class="col-xs-5 error-main-right">
<img src="images/7.png" alt=" " class="img-responsive" /></div>
<div class="clearfix"> </div></div></div></div>
<script src="js/jquery.nicescroll.js"></script>
<script src="js/scripts.js"></script>
<!-- Bootstrap Core JavaScript -->
<script src="js/bootstrap.min.js"></script>
</body>
```

</html>

/*Motor Controller*/

import receiveData

import sendData

import Ultrasonic1

import Valve_WaterFlow from firebase import firebase import time

import getArduinodata count= len(result)

sendData.initializeConnection() for i in range(count):

user_id = result[i]

Turbidity = getArduinodata.get_control_Arduino('1')

index = sendData.sendData(fb,user_id,"Turbidity",Turbidity,-1)

Updated_data1={'qualityTurbidity':1} Updated_data2={'qualityTurbidity':0}

if Turbidity > 70: sendData.updateData(user_id,Updated_data1)

else:

sendData.updateData(user_id,Updated_data2) print Turbidity

water_level = Ultrasonic1.ultrasonic1(user_id,result)

sendData.sendData(fb,user_id,"Waterlevel",water_level,index)

#Updated_data = {'water_level':water_level,'Turbidity':Turbidity}

#sendData.sendData(user_id,Updated_data)

#max_cap = result['User'+str(i+1)]['maxCap'] MaxCap =

fb.get(user_id+'/','MaxCap')

```
if water_level < MaxCap:
fetch = receiveData.fetchData(user_id)
reply = fb.get(user_id+'/Notify/','usercontrolwater') #max_cap = fetch['maxCap']
if reply == 0: print 'Ready' time.sleep(2) water_meter =



Valve_WaterFlow.Valve_WaterFlow(user_id,water_level,MaxCap,result)
sendData.sendData(fb,user_id,'waterFlow',water_meter,index) print water_meter
Updated_data = {'usercontrolwater':1}
sendData.updateData(user_id,Updated_data)
elif reply==1:
print "Water Not Required" if i==1:
i=0 continue
/*Ultrasonic Sensor */
import RPi.GPIO as GPIO import time TRIG1,ECHO1 = 7,12
TRIG2,ECHO2 = 16,15 #GPIO.setmode(GPIO.BOARD)
def ultrasonic1(user_id,result):
GPIO.setmode(GPIO.BOARD)
if user_id == result[0] :
TRIG,ECHO = TRIG1,ECHO1



elif user_id == result[1] : TRIG,ECHO = TRIG2,ECHO2
GPIO.setup(TRIG,GPIO.OUT)
```

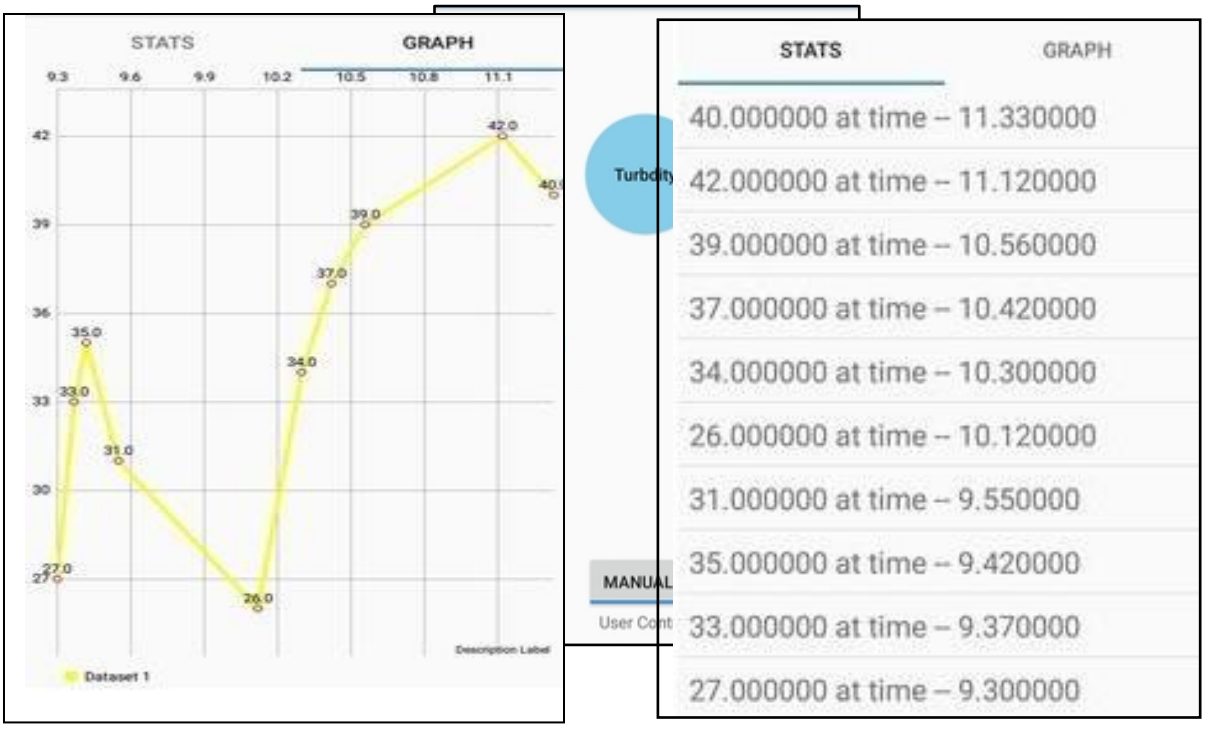
```
GPIO.output(TRIG,0)
GPIO.setup(ECHO, GPIO.IN)
time.sleep(0.1) print "Start"



GPIO.output(TRIG,1) time.sleep(0.00001) GPIO.output(TRIG,0)
while GPIO.input(ECHO) == 0: pass start = time.time()
while GPIO.input(ECHO) == 1: pass
stop = time.time()
#print (stop - start) * 17000 GPIO.cleanup()
depth = (stop - start) * 17000 #print depth
length = 13
water_level = (length - depth)/length * 100 if depth > 13:
return 0
print water_level return water_level #print water_level
#ultrasonic1(2)
/*Valve Controller*/
import RPi.GPIO as GPIO import time,sys
import Ultrasonic1 import getArduinodata

#GPIO.setmode(GPIO.BOARD)
inpt1,inpt2=13,18

def Valve_WaterFlow(user_id,water_level,maxCap,result):
GPIO.setmode(GPIO.BOARD)
```

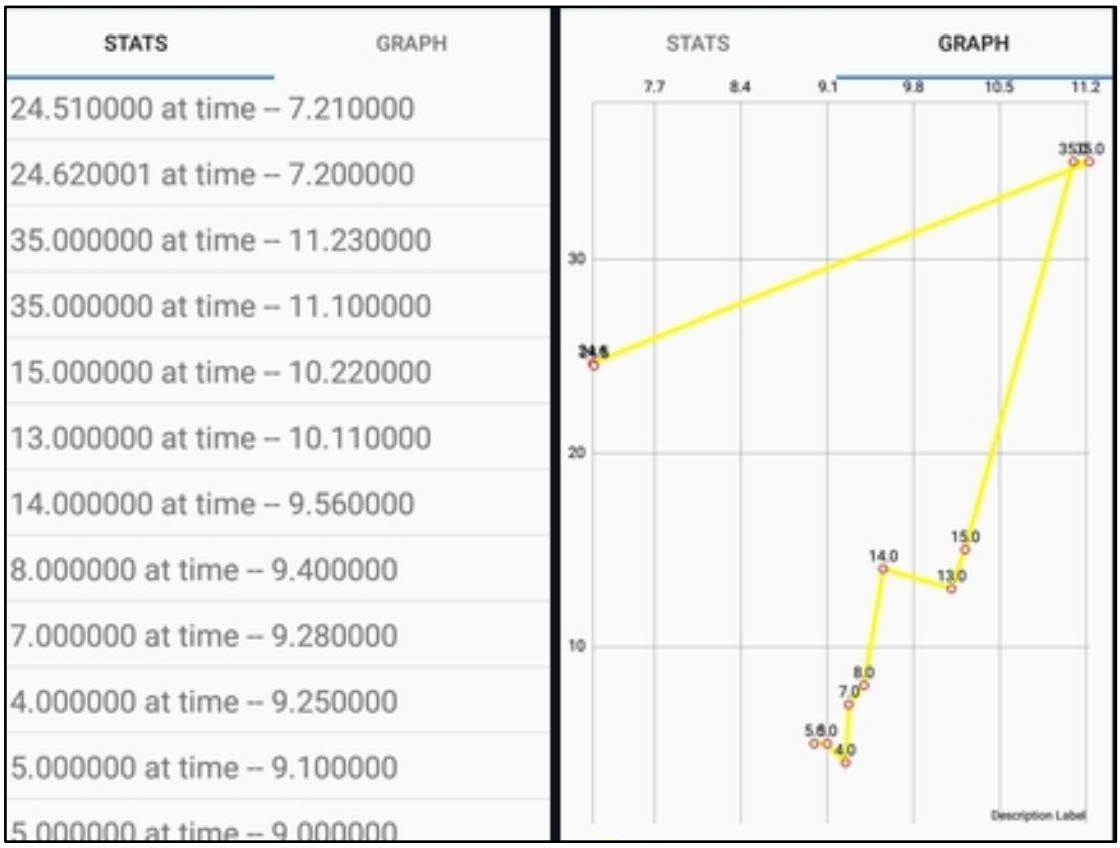
```
if user_id == result[0]: inpt = inpt1

code = '2'
elif user_id == result[1]: inpt = inpt2
code = '4'



GPIO.setup(inpt,GPIO.IN) rate_cnt=0
tot_cnt=0 minutes=0 count=0 constant=0.05 time_new=0.0 #openValve()
getArduinodata.get_control_Arduino(code) water_meter = 0
#print('Water Flow-Approx') while True:
time_new=time.time() + 5 rate_cnt=0
while time.time()<=time_new: if GPIO.input(inpt)!=0:
rate_cnt+=1 tot_cnt+=1
count+=1 try:
print(GPIO.input(inpt)) #None
except KeyboardInterrupt: print('\CTRL C - Exiting nicely') GPIO.cleanup()
sys.exit() #print(count) #minutes+=1
water_meter+=round(rate_cnt*constant,4)

if Ultrasonic1.ultrasonic1 >= maxCap: #closeValve
getArduinodata.get_control_Arduino(str(int(code)+1)) break

#print('\nLitres/min',round(rate_cnt*constant,4)) #print('Total
Liters',round(tot_cnt*constant,4))

#print('Done') #break
```

GPIO.cleanup() print water_meter return water_meter

#Valve_WaterFlow(2,50,100)

APPLICATION OUTPUT





13.2. GitHub & Project Demo Link

GITHUB :

https://github.com/IBM-EPBL/IBM-Project-23677-1659892340

PROJECT DEMO LINK:

https://drive.google.com/drive/folders/1faJGL2m8Zi6R-bYfe8a98KBQXVwdvMN9?usp=sharing