

# **FINAL REPORT**

## **REAL TIME RIVER WATER QUALITY MONITORING AND CONTROL SYSTEM**

### **TEAM DETAILS**

**TEAM ID: PNT2022TMID38637**

**TEAM LEADER: INDIRA PRIYADHARSHINI.V-  
(420419106014)**

#### **TEAM MEMBERS:**

- 1. VAISHNAVI.G-(420419106028)**
- 2. KALPANA.S-(420419106016)**
- 3. ASWINI.E-(420419106003)**
- 4. DHARMADEVI.P-(420419106301)**

## **CONTENTS**

### **1. INTRODUCTION**

- 1.1 Project Overview
- 1.2 Purpose

### **2. LITERATURE SURVEY**

- 2.1 Existing problem
- 2.2 References
- 2.3 Problem Statement Definition

### **3. IDEATION & PROPOSED SOLUTION**

- 3.1 Empathy Map Canvas
- 3.2 Proposed Solution
- 3.3 Problem Solution fit

### **4. REQUIREMENT ANALYSIS**

- 4.1 Functional requirement
- 4.2 Non-Functional requirements

### **5. PROJECT DESIGN**

- 5.1 Data Flow Diagrams
- 5.2 Solution & Technical Architecture
- 5.3 User Stories

### **6. PROJECT PLANNING & SCHEDULING**

- 6.1 Sprint Planning & Estimation
- 6.2 Sprint Delivery Schedule

### **7. PRE-REQUISITES:**

- 7.1 Python
- 7.2 IBM Watson
- 7.3 Node-red
- 7.4 MIT App Inventer
- 7.5 Fast2sms

### **8. CODING & SOLUTIONING (Explain the features added in the project along with code)**

- 8.1 Python code
- 8.2 Output screen

### **9. TESTING**

9.1 Test Cases

9.2 User Acceptance Testing

## **10.RESULTS**

10.1 Performance Metrics

## **11.ADVANTAGES & DISADVANTAGES**

## **12.CONCLUSION**

## **13.FUTURE SCOPE**

## **14.APPENDIX**

Source Code

GitHub & Project Demo Link

## **1. INTRODUCTION:**

### **1.1PROJECT OVERVIEW:**

The project will monitor the quality of river water. Since Hardware is not available random values are generated for sensor data from python code.

The nodes are connected using node-red platform and the output is viewed using an android application.

### **1.2 PURPOSE:**

The purpose of this project is to monitor the quality of river water in run time and alert the user or the respective authority if the quality of water is degraded. The respective user must be notified about the water quality by SMS.

## **2. LITERATURE SURVEY:**

### **2.1 EXISTING PROBLEM:**

The quality of river water is getting degraded day by day due to many pollutants that is let into the river. So, the water becomes polluted and unable to use for domestic purposes. The respective government authority

must be notified about the water quality frequently so that they will take respective action to maintain the water quality.

## **2.2 REFERENCES:**

- Digital Object Identifier 10.1109/ACCESS.2016.2592958 Design of Smart Sensors for Real-Time Water Quality Monitoring NIEL ANDRE CLOETE, REZA MALEKIAN, (Member, IEEE), AND LAKSHMI NAIR, (Member, IEEE) Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria 0002, South Africa Corresponding author: R. Malekian (reza.malekian@ieee.org). Received June 3, 2016, accepted July 16, 2016, date of publication July 19, 2016, date of current version August 26, 2016.
- International Journal for Research in Applied Science & Engineering Technology (IJRASET) ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887 Volume 6 Issue IV, April 2018- Available at [www.ijraset.com](http://www.ijraset.com) ©IJRASET (UGC 74 Approved Journal): All Rights are Reserved IoT based Standard Water Measuring System using GSM S. Suganya<sup>1</sup>, K. Deepa<sup>2</sup>, A.Mahalakshmi<sup>3</sup>, Dr. P. Gomathi<sup>4</sup>, V. Praveen
- Water Quality Monitoring System Based on IOT Vaishnavi V. Daigavane and Dr. M.A Gaikwad Department Electronics & Telecommunication Engineering, Mtech(VLSI), Bapurao Deshmukh College of Engineering, Sevagram, wardha\_442102(M.S.), India
- International Journal of Advanced Research in Computer and Communication Engineering Vol. 9, Issue 1, January 2020 Copyright to IJARCCCE DOI 10.17148/IJARCCCE.2020.9104 24 Water Quality Monitoring Adithya Jayaraj<sup>1</sup>, Esly V Kuriyan<sup>2</sup>, Jossila N J<sup>3</sup>, Reshma Viswanath C<sup>4</sup>, Muneebah Mohyiddeen<sup>5</sup>

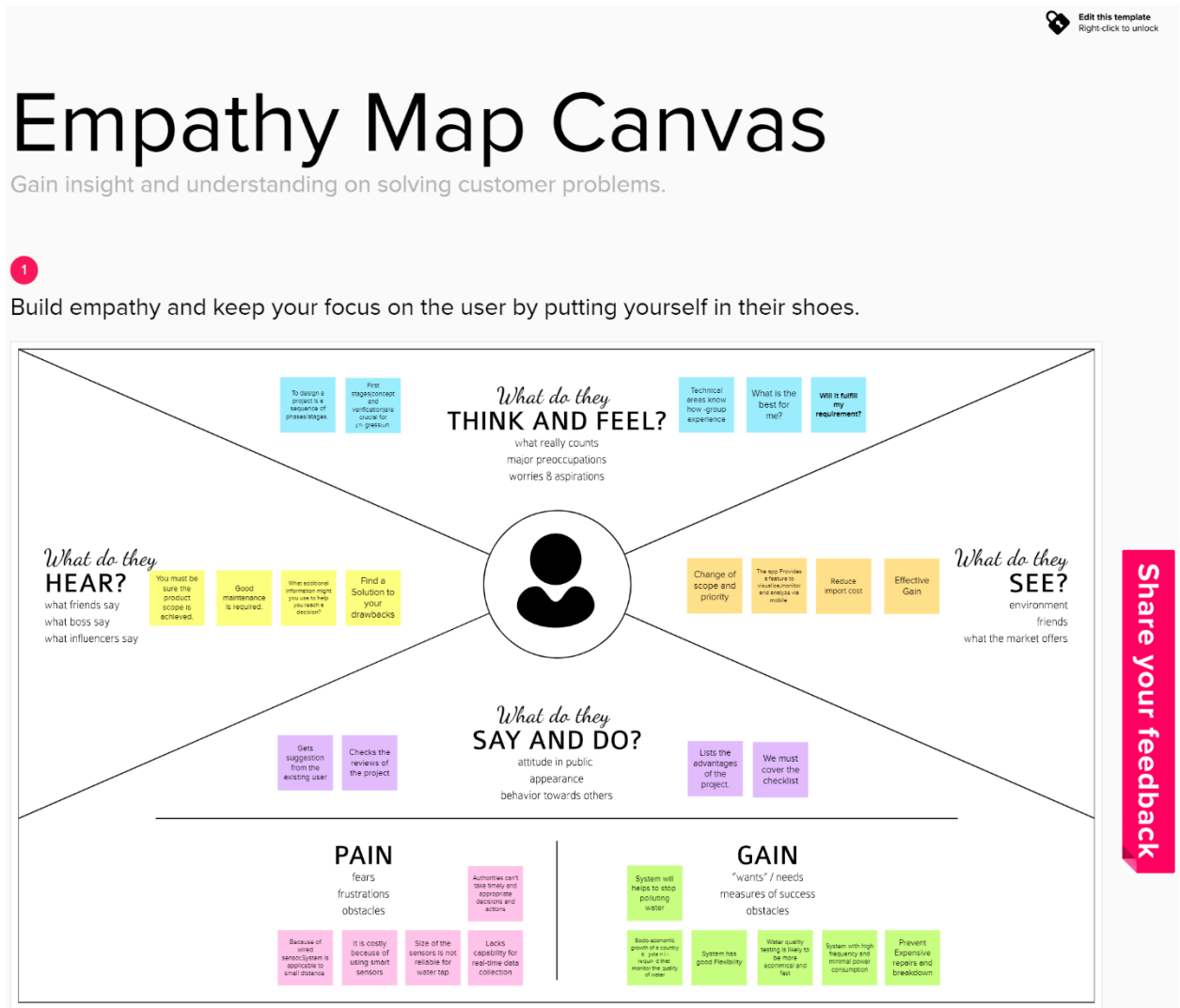
## **2.3 PROBLEM STATEMENT DEFINITION:**

- The traditional water quality monitoring system has certain drawbacks
- It relies on collecting of water samples, testing and analyses in laboratories
- It results in more cost, more man power and time consuming.

- It lacks capability for real-time data collection, hence authorities can't take timely and appropriate decisions and actions.

### 3. IDEATION AND PROPOSED SOLUTION:

#### 3.1 EMPATHY MAP CANVAS:



### 3.2 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	<ul style="list-style-type: none"><li>Water gets polluted due to industrial waste, medical waste, fertilizers or farm wastes. So water cannot be used for domestic purposes.</li></ul>
2.	Idea / Solution description	<ul style="list-style-type: none"><li>To measure various chemical and physical properties of water like pH, temperature and particle density of water using sensors.</li><li>Dispose the industrial waste, toxic chemicals and medical waste properly.</li></ul>
3.	Novelty / Uniqueness	<ul style="list-style-type: none"><li>If the acquired value is above the threshold value automated warning SMS will be sent to the agent.</li><li>The uniqueness of our proposed paper is to obtain the water monitoring system with high frequency, high mobility and low powered</li></ul>
4.	Social Impact / Customer Satisfaction	<ul style="list-style-type: none"><li>Living organism such as aquatics, human beings, animals suffer due to polluted water so this can be avoided.</li><li>Water borne diseases such as Typhoid, Cholera, Polio etc.. can be prevented and avoided.</li></ul>
5.	Business Model (Revenue Model)	<ul style="list-style-type: none"><li>Water quality monitoring system by aeron systems for industrial water treatment plant, river bodies, aqua forming ,digital loggers.</li></ul>
6.	Scalability of the Solution	<ul style="list-style-type: none"><li>Water quality data with a high spatial and temporal resolution for thousands of lakes at a time.</li><li>It supports the evaluation of environmental problems and potential health risks through the analysis of changes in water quality and the detection of harmful algal blooms.</li></ul>

### 3.3 PROBLEM SOLUTION FIT

Define C.S. fit into CC	<b>1.CUSTOMER SEGMENT(S)</b>  Government authorities, Farmers and Drinking Water supplier. People living in rural areas near to the river, who uses river water.	<b>6.CUSTOMER CONSTRAINTS</b>  Water quality monitoring system is used for identifying the water pollution on specific areas. People may find it hard to recover if any fault occurs, this system prevent people from water pollution.	<b>5.AVAILABLE SOLUTIONS</b>  Individual notification to each people could be sent, it is not possible. This system will still notify the corporation and they can further notify the <u>people to</u> aware.	Explore & S. differentiate
Focus on J&P. tap into BE, understand RC	<b>2.JOBS-TO-BE-DONE / PROBLEMS</b>  <ul style="list-style-type: none"> <li>• Check the water quality.</li> <li>• Check the level of chlorine in water.</li> <li>• Check temperature of water.</li> <li>• Check the pH level of water.</li> <li>• Find if the water is suitable for drinking, agriculture and aquaculture.</li> </ul>	<b>9.PROBLEM ROOT CAUSE</b>  Root Cause Analysis supported by input from the problems-sufferers, instruction manual studies, comparing design and actual operating data, gathering know how from relevant literature, tech journals articles and advertisements especially on new products.	<b>7.BEHAVIOUR</b>  Customers could use the user guide provided to them to overcome the problem or else they can report and contact the corporation.	Focus
Ex 3. TRIGGERS	<b>3. TRIGGERS</b> River water quality analysis work by checking the river water quality for providing clean drinking water for the people, farming, promoting aquaculture and other industries. It is a best replacement for checking water quality in laboratories. The best quality is that it is user friendly.	<b>10. YOUR SOLUTION</b>  <ul style="list-style-type: none"> <li>• Implement IOT based river water quality monitoring system to get instant results. Suggestions can be made to solve if any problems <u>arises</u>.</li> </ul>	<b>8.CHANNELS OF BEHAVIOURS</b>  Online portal for making recommendations for problems based on pH parameters using Machine Learning.	Extract online & offline



## 4. REQUIREMENT ANALYSIS:

### 4.1 FUNCTIONAL REQUIREMENT:

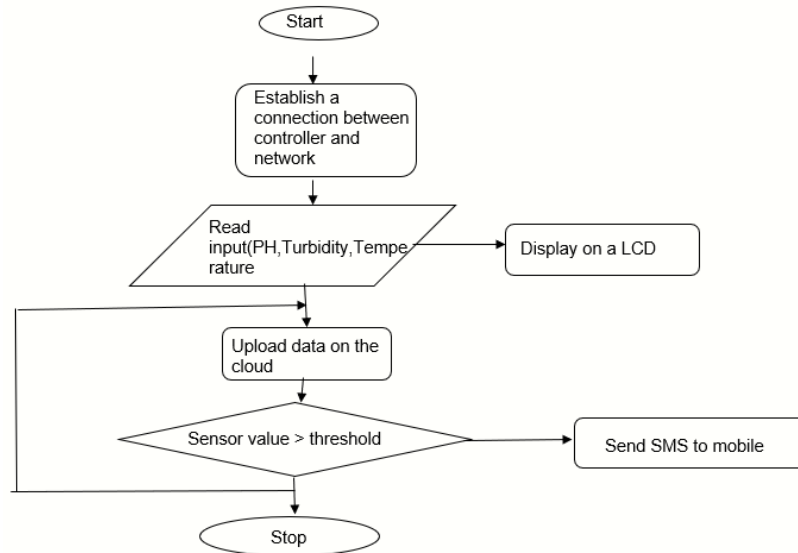
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	River water sensing	The sensing parameters are PH, Turbidity, temperature.
FR-2	Data collection	The accurate value of pH, Temperature, Turbidity are Collected from various samples.
FR-3	Monitor	The collected data can be monitored by using quality monitoring system by using Arduino and python code.
FR-4	Control	The system controls the utilization of degraded water.
FR-5	Data storage	The data can be stored by using cloud service such as (IBM Watson IoT, IBM Cloud.)
FR-6	Intimation to Authority	The stored data can be sent to the Corporation using Short Message Service or email services.

### 4.2 NON-FUNCTIONAL REQUIREMENT:

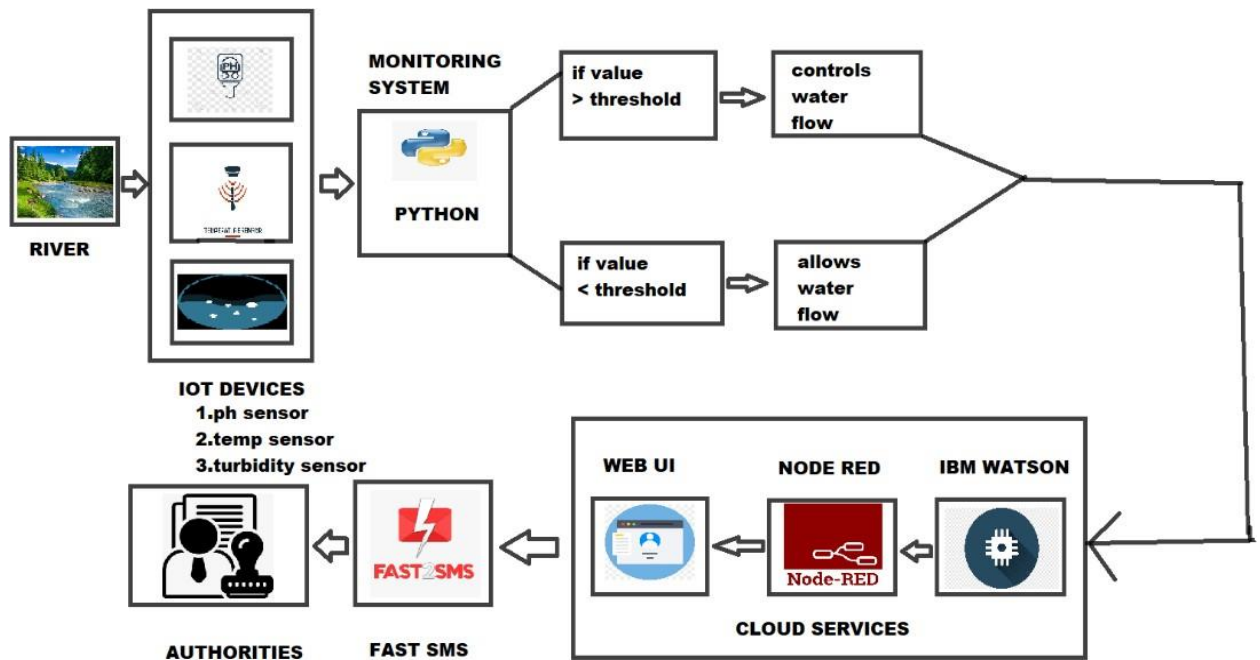
FR No.	Non-Functional Requirement	Description
NFR-1	<b>Usability</b>	To measure water parameters such as pH, dissolved oxygen, turbidity, conductivity, etc. using available sensors at a remote place
NFR-2	<b>Security</b>	This data can be accessed by the authorized users by logging into their accounts using a User ID and password to view data. The data is collected, processed, analyzed, and transmitted and displayed all in real time
NFR-3	<b>Reliability</b>	Wireless sensor networks with low power consumption, low-cost and high detection accuracy in pH, conductivity, turbidity level, etc.
NFR-4	<b>Performance</b>	Creating a system that uses wireless sensor networks to continuously monitor river water quality at remote locations with low power consumption, low cost and highly efficient for monitoring
NFR-5	<b>Availability</b>	The system is available for 24/7 for the regular monitoring of quality water.
NFR-6	<b>Scalability</b>	This project is scalable for a small area.

## 5. PROJECT DESIGN:

### 5.1 DATA FLOW DIAGRAM:



### 5.2 SOLUTION AND TECHNICAL ARCHITECTURE:



### 5.3 USER STORIES:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Circuit designer	Designing the circuit	USN-1	As a user, I can design the circuit by using open source software.	I can get the exact design for my project.	High	Sprint-1
		USN-2	As a user, I can design the circuit by using free web app like Tinkercad.	I can make several attempts to get the right design.	High	Sprint-1
Programmer	Create a program suitable for the circuit	USN-3	As a user, I can create programs in the user friendly language.	I can create a simple program for the circuit	High	Sprint-1
		USN-4	As a user, I can compile and execute the programs.	I can get the program with accurate outputs.	High	Sprint-1
Engineer	Connects the output to the cloud	USN-5	As a user, I can connect the output values to the cloud services by using NODE RED.	I can make the data to receive in cloud.	High	Sprint-1
	Store the output values	USN-6	As a user, I can make the data store in IBM cloudant database.	I can retrieve the data anywhere, anytime.	Medium	Sprint-2
	Connects the cloud data with the authorities communication device.	USN-7	As a user, I can produce connection to the authorities mobile phones so that they can receive the alerts.	I can make the authorities informed about the water's quality.	Medium	Sprint-2
	Alerts has to be sent to the authorities	USN-8	As a user, I can make use of platforms such as Fast SMS to send the timely updates to the authorities.	I can make the authorities to get accurate values and alerts	High	Sprint-1
Authorities	Checks the water quality alerts	USN-9	As a user, I check the quality values of the water that is sent to me .	I can make sure that the people in my zone gets quality water.	High	Sprint-1
Maintenance	Monitoring	USN-10	The entire IOT system is get maintained	It increase the efficiency and functionality of the system.	Low	Sprint-3
Administrator	Manage	USN-11	All the datas are get recorded	The stored data used for future references.	Medium	Sprint-1

## 6. PROJECT PLANNING AND SCHEDULING:

### 6.1 SPRINT PLANNING 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	INDIRA PRIYADHARSHINI.V, VAISHNAVI.G, KALPANA.S
	Registration via <del>facebook</del>	USN-3	As a user, I can register for the application through Facebook	2	Low	
	Registration via Mail ID	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 application by entering email & password	1	High	
	IBM Cloud service access		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 the sensed data on cloud	2	High	DHARMADEVLP, ASWINIE
	Create a node red service	USN-7	To create a node red service to integrate the IBM Watson along with the Web UI	2	Medium	INDIRA PRIYADHARSHINI.V, VAISHNAVI.G
	Create a Web UI	USN-8	To create a Web UI, to access the data from the cloud and display all parameters.	2	Medium	INDIRA PRIYADHARSHINI.V, VAISHNAVI.G
	To develop a Python code	USN-9	Create a python code to sense the physical quantity and store data.	2	Medium	VAISHNAVI.G, KALPANA.S
	Publish Data to cloud.	USN-10	Publish Data that is sensed by the microcontroller to the Cloud	3	High	INDIRA PRIYADHARSHINI.V
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	INDIRA PRIYADHARSHINI.V, VAISHNAVI.G, KALPANA.S,
	Testing	USN-12	Testing of project and final deliverables	3	Medium	ASWINIE, DHARMADEVLP

## 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	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

## 7. PRE-REQUISITES:

### 7.1 PYTHON:

The screenshot shows the Python.org website's download page. At the top, there's a table with two rows of security updates for Python 3.8 and 3.7. Below this, a section titled 'Looking for a specific release?' provides a list of Python releases by version number. The table includes columns for 'Release version', 'Release date', and 'Click for more' (with links to 'Download' and 'Release Notes'). The releases listed are Python 3.6.8, 3.7.1, 3.6.7, 3.5.6, 3.4.9, 3.7.0, 3.6.6, and 2.7.15. At the bottom, there's a 'Sponsors' section mentioning Bloomberg and Meta.

Release version	Release date	Click for more
Python 3.6.8	Dec. 24, 2018	<a href="#">Download</a> <a href="#">Release Notes</a>
Python 3.7.1	Oct. 20, 2018	<a href="#">Download</a> <a href="#">Release Notes</a>
Python 3.6.7	Oct. 20, 2018	<a href="#">Download</a> <a href="#">Release Notes</a>
Python 3.5.6	Aug. 2, 2018	<a href="#">Download</a> <a href="#">Release Notes</a>
Python 3.4.9	Aug. 2, 2018	<a href="#">Download</a> <a href="#">Release Notes</a>
Python 3.7.0	June 27, 2018	<a href="#">Download</a> <a href="#">Release Notes</a>
Python 3.6.6	June 27, 2018	<a href="#">Download</a> <a href="#">Release Notes</a>
Python 2.7.15	May 1, 2018	<a href="#">Download</a> <a href="#">Release Notes</a>

[View older releases](#)

**Sponsors**  
Visionary sponsors help to host Python downloads.

**Bloomberg** **Meta**

```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> print("hello world")
hello world
>>>
```

## 7.2 IBM WATSON:

IBM Watson IoT Platform

420419104028@smartinternz.com  
ID: 002615

Browse

Action

Device Types

Interfaces

Add Device

Browse Devices

All Devices

Diagnose

This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

Search by Device ID

Device Simulator

	Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
>	1607	Connected	gv	Device	Oct 23, 2022 11:33 AM	

Items per page 50

1-1 of 1 item

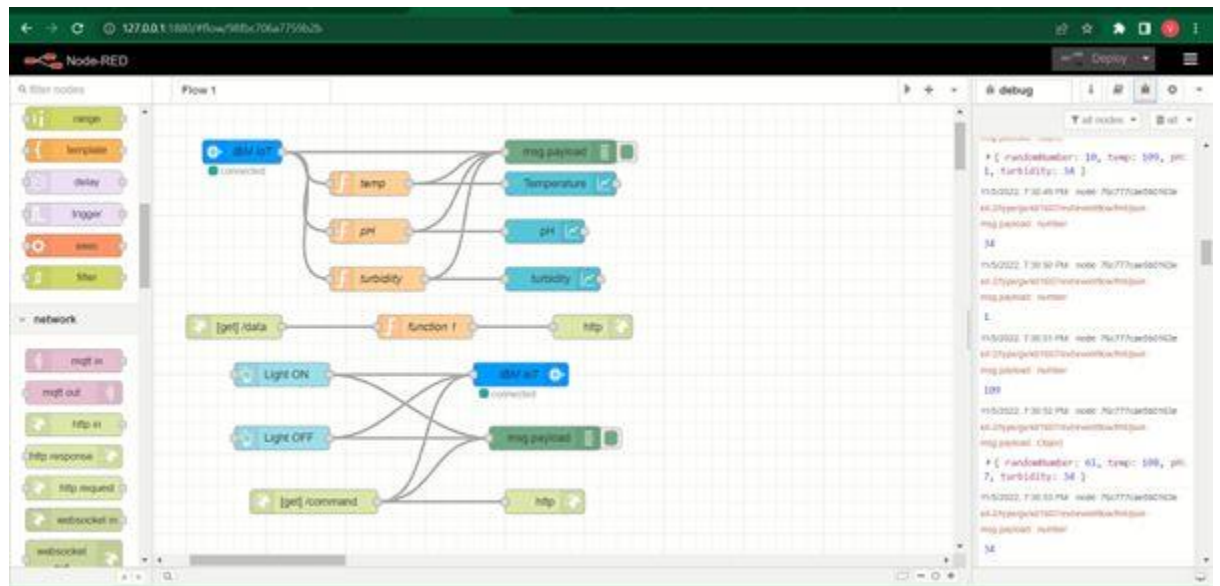
1 of 1 page

<

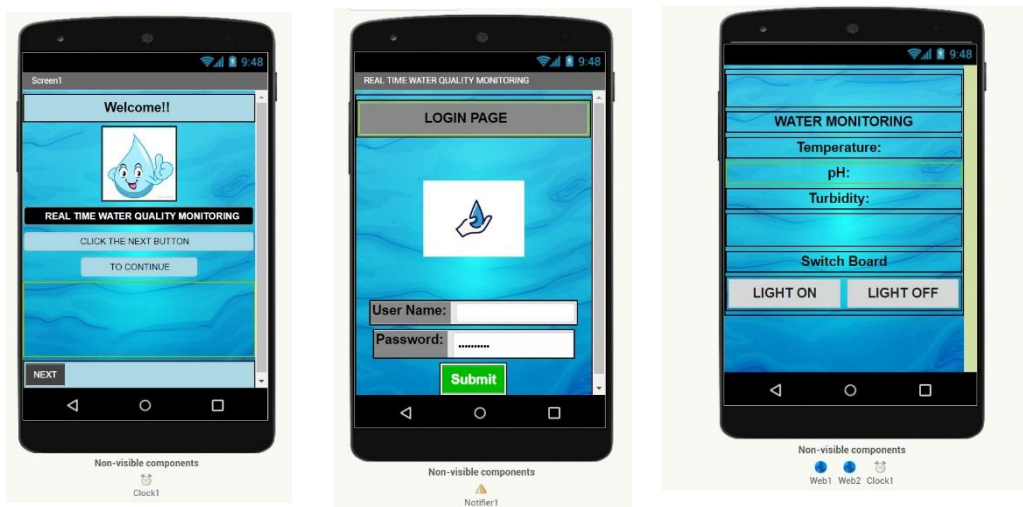
1

>

### 7.3 NODE-RED:



### 7.4 MIT APP INVENTER:



## 7.5 FAST2SMS

The screenshot displays the FAST2SMS developer API dashboard. The top navigation bar includes the FAST2SMS logo, a balance of ₹55.00, and an 'ADD CREDIT' button. The main menu on the left lists various services: Bulk SMS, DLT SMS, Quick SMS, Address Book, Delivery Reports, Transactions, Dev API (selected), Settings, and Help. The 'Dev API' section is active, showing tabs for 'Dev API', 'API Key', and 'Security'. The 'Dev API' tab is selected, displaying a form for configuring a GET request. The form includes fields for Method (GET), Route (Quick SMS), Message (Alert! Alert!! River water quality is getting degraded), and Language. A 'READ API DOCS' button is also present. The right sidebar shows the 'GET https://www.fast2sms.com/dev/bulkV2' endpoint and the resulting query parameters and overall URL.

**FAST2SMS** ₹55.00 [ADD CREDIT](#)

How Developer API Works Account Info 11:52:45 AM Vaishnavi...

**Dev API** API Key Security

For OTP Based SMS use 'OTP SMS API' [READ API DOCS](#)

Method: GET

Route: Quick SMS

Message (NOTE: Per SMS cost ₹ 3.50): Alert! Alert!! River water quality is getting degraded

Language:

**GET https://www.fast2sms.com/dev/bulkV2**

Query Parameter :

authorization = 0Kq7hD9ZIGREyMrPQpJOISNLvzoibldw3UH5FVfgx2W6As1unjfsndCiGF29VyySkWPuxO4MqlprbUD3

route = q

message = Alert! Alert!! River water quality is getting degraded

language = "english"

numbers =

flash = "0"

**Overall URL =** https://www.fast2sms.com/dev/bulkV2?authorization=0Kq7hD9ZIGREyMrPQpJOISNLvzoibldw3UH5FVfgx2W6As1unjfsndCiGF29VyySkWPuxO4MqlprbUD3&route=q&message=Alert!%20Alert!!%20River%20water%20quality%20is%20getting%20degraded&language=english&flash=0&numbers=



## **8. CODING AND SOLUTION:**

### **8.1 PYTHON CODE & FEATURES:**

- In this python code we have included our IBM Watson IoT platform device details so that the python code will connect to the device. For that we have imported ibmiotf module.
- Then we have given the quality parameters temperature, pH and turbidity. Since hardware isn't available, we have generated random values for the sensor data. For that we have imported random module.
- Then to send alert notification to the respective authority by sms, we are importing the twilio module. Twilio account has been created to receive the account\_sid, auth\_token and twilio number to send sms to the target number.
- A threshold value is set so that if the quality parameter values go beyond the threshold value, an alert notification is sent to the user. A function named SMS is defined to perform these function.
- The time is set to 10 seconds so that the random values will be generated every 10 seconds for that we are importing the time module.

## **PROGRAM:**

```
import time

import sys

import ibmiotf.application

import ibmiotf.device

import random

import os

from twilio.rest import Client

account_sid = 'AC85f1dc96918c0626ee5f43f0f850a6e5'

auth_token = '1326cea3c0ddb356975873b4625b5e9c'

twilio_number='+19852988565'

target_number='+919790728634'

#provide Your IBM Watson

Device Credentials

organization = "uwobmo"

deviceType = "Indira"

deviceID = "26"

authMethod = "token"

authToken = "12345678"

#Initialize GPIO
```

```

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

status=cmd.data['comman
d']

if status=="LIGHT ON":
    print ("led is on")
elif status == "LIGHT
OFF":
    print ("led is off")
else:
    print ("please send
proper command")
try:
    deviceOptions =
{'org':organization,'type':de

```

```
viceType,'id':deviceID,'auth
```

```
-method':authMethod,
```

```
'auth-token': authToken}
```

```
deviceCli =
```

```
ibmiotf.device.Client(devic
```

```
eOptions)
```

```
#.....
```

```
.....
```

```
except Exception as e:
```

```
    print("caught
```

```
exception connecting
```

```
device:%s" % str(e))
```

```
    sys.exit()
```

```
# connect and send a
```

```
datapoint "hello"with value
```

```
"world" info the cloud as
```

```
an event of
```

```
type"greetings"10 times
```

```
deviceCli.connect()
```

```
while True:
```

```
    #Get sensor Data
```

```
    from DHT11
```

```
    Temperature=random.rand
```

```
    int(90,110)
```

```
    pH=random.randint(0,14)
```

```
    Turbidity=random.randint(
```

```
    0,100)
```

```
    data = { 'Temperature'
```

```
            : Temperature, 'pH': pH,
```

```
            'Turbidity':Turbidity }
```

```
    def SMS():
```

```
        message = Client.messages\
```

```
            .create(
```

```

        body="Alert! Alert!! River water quality is getting
degraded",

        from_=twilio_number,

        to = target_number)

    print(message.body)

    if(temp>80 or pH<6 or turbidity>70):

        SMS()

#print data

def

myOnPublishCallback():

    print ("published

Temperature = %s C" %

Temperature, "pH = is %s

%%" % pH, "Turbidity= is

%s %%" % Turbidity,"to

IBM Watson")

    success =

    deviceCli.publishEvent("IO

TSensor",

    "json",data,qos=0,on_publi

```

```
sh=myOnPublishCallback)
```

```
if not success:
```

```
    print("Not
```

```
connected to IOTF")
```

```
    time.sleep(10)
```

```
deviceCli.commandCallba
```

```
ck = myCommandCallback
```

```
# Disconnect the device
```

```
and application from the
```

```
cloud
```

```
deviceCli.disconnect()
```

## 8.2 OUTPUT SCREEN:

```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\V.IndrabPriyadarshni\AppData\Local\Programs\Python\Python37\ikmiotpublish.py
2022-11-13 19:54:26,446 ikmiotf.device.Client INFO Connected successfully: d:\uwbmm:indira:26
published Temperature = 109 C pH = is 7 % Turbidity= is 76 % to IBM Watson
published Temperature = 92 C pH = is 4 % Turbidity= is 20 % to IBM Watson
published Temperature = 90 C pH = is 11 % Turbidity= is 76 % to IBM Watson
published Temperature = 106 C pH = is 13 % Turbidity= is 94 % to IBM Watson
published Temperature = 101 C pH = is 2 % Turbidity= is 52 % to IBM Watson
command received: LIGHT ON
led is on
published Temperature = 91 C pH = is 5 % Turbidity= is 4 % to IBM Watson
command received: LIGHT OFF
led is off
published Temperature = 100 C pH = is 4 % Turbidity= is 11 % to IBM Watson
command received: LIGHT ON
led is on
command received: LIGHT OFF
led is off
published Temperature = 96 C pH = is 6 % Turbidity= is 70 % to IBM Watson
|
```

Ln 21 Col 0

26°C Cloudy Search 19:55 13-11-2022



## 9. TESTING:

### 9.2TEST CASES:

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments
LoginPage_TC_OO2	UI	Home Page	Verify the UI elements in Login/Signup popup	Android application	1.Enter the app and click NEXT 2.Verify login popup with below UI elements: a. Username text box b.password text box c. Submit button	<a href="http://ai2.apinventor.mit.edu/#5118929299963904">http://ai2.apinventor.mit.edu/#5118929299963904</a>	Application should show below UI elements: a.Username text box b.password text box c. Submit button with green colour	Working as expected	Pass	No issues faced
LoginPage_TC_OO3	Functional	Home page	Verify user is able to log into application with Valid credentials	Login credentials	1.Enter the app and click NEXT 2.Enter Valid username in Username text box 3.Enter valid password in password text box 4.Click on submit button	Username: Indira password: gviptd	User should navigate to user account homepage	Working as expected	Pass	No issues faced
LoginPage_TC_OO4	Functional	Login page	Verify user is able to log into application with Invalid credentials	Login credentials	1.Enter the app and click NEXT 2.Enter Valid username in Username text box 3.Enter valid password in password text box 4.Click on submit button	Username: kalpana password: kps	Application should show 'Incorrect email or password' validation message.	Working as expected	pass	No issues faced
LoginPage_TC_OO4	Functional	Login page	Verify user is able to log into application with Invalid credentials	Login credentials	1.Enter the app and click NEXT 2.Enter Valid username in Username text box 3.Enter valid password in password text box 4.Click on submit button	Username: Test1 password: abcd	Application should show 'Incorrect email or password' validation message.	Working as expected	pass	No issues faced
MonitoringPage_TC_OO5	Functional	Monitoring page	Verify the simulated sensor data is displayed on the Android application	Random values	1.Enter the app and click NEXT 2.Enter Valid username in Username text box 3.Enter valid password in password text box 4.Click on submit button	Temperature: 108, pH: 13, Turbidity: 13	Application should display the random values that has been generated from the python code	Working as expected	pass	No issues faced

### 9.2 USER ACCEPTANCE TESTING:

#### 1. Defect Analysis

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	7	2	3	2	14
Duplicate	1	0	0	2	3
External	2	4	0	5	11
Fixed	10	2	3	7	22
Not Reproduced	0	0	1	1	2
Skipped	0	0	0	1	1
Won't Fix	0	2	1	3	6
Totals	20	10	8	21	59

## 2. 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	6	0	0	6
Client Application	25	0	0	25
Security	3	0	0	3
Exception Reporting	8	0	0	8
Final Report Output	5	0	0	5
Version Control	2	0	0	2

## 10.RESULTS:

### 10.1 PERFORMANCE METRICS:

NFT Test approach	
Load Test	
Scenario Name	Load Test - River water quality monitor
Scenario Type	Load Test - Duration 15 minutes
Scenario Objectives	To Stimulate Python Code( Quality parameters ) and to monitor the performance of River water quality monitor
Steps	Real-Time River Water Quality Monitoring and Control System
Entry Criteria	Test data is set-up. All the Components( software) is set-up. It is completed successfully.
Exit Criteria	Response time meets the actual Result. Test completion report is agreed upon by mentors



## **12.CONCLUSION:**

Therefore the android application has been designed and the Water quality parameters has been displayed in the monitor screen and when the water quality seems to be degraded the respective authority is being notified about it by sms.

## **13.FUTURE SCOPE:**

- In future we can rebuild the project to test for any chemical contaminants and provide solutions to overcome the chemical pollutants in the water.
- Environmental water quality monitoring aims to provide the data required for safeguarding the environment against adverse biological effects from multiple chemical contamination arising from anthropogenic diffuse emissions and point sources.
- we can integrate the experience of the international EU-funded project SOLUTIONS to shift the focus of water monitoring from a few legacy chemicals to complex chemical mixtures, and to identify relevant drivers of toxic effects.

- Monitoring serves a range of purposes, from control of chemical and ecological status compliance to safeguarding specific water uses, such as drinking water abstraction.
- Various water sampling techniques, chemical target, suspect and non-target analyses as well as an array of in vitro, in vivo and in situ bioanalytical methods were advanced to improve monitoring of water contamination.
- Major improvements for broader applicability include tailored sampling techniques, screening and identification techniques for a broader and more diverse set of chemicals, higher detection sensitivity, standardized protocols for chemical, toxicological, and ecological assessments combined with systematic evidence evaluation techniques.
- (1) identify relevant contaminants,  
(2) assess the impact of contamination in aquatic ecosystems, or  
(3) quantify cause–effect relationships between contaminants and adverse effects.

## **14.APPENDIX:**

### **SOURCE CODE:**

```
import time

import sys

import ibmiotf.application

import ibmiotf.device

import random

import os

from twilio.rest import Client

account_sid = 'AC85f1dc96918c0626ee5f43f0f850a6e5'

auth_token = '1326cea3c0ddb356975873b4625b5e9c'

twilio_number='+19852988565'

target_number='+919790728634'

#provide Your IBM Watson

Device Credentials

organization = "uwobmo"

deviceType = "Indira"

deviceID = "26"

authMethod = "token"

authToken = "12345678"
```

```

#Initialize GPIO

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

status=cmd.data['comman
d']

if status=="LIGHT ON":
    print ("led is on")
elif status == "LIGHT
OFF":
    print ("led is off")
else:
    print ("please send
proper command")
try:
    deviceOptions =

```

```
{'org':organization,'type':deviceType,'id':deviceID,'auth-method':authMethod,'auth-token': authToken}

deviceCli =
ibmiotf.device.Client(deviceOptions)
```

```
#.....
```

```
.....
```

```
except Exception as e:
```

```
    print("caught
```

```
exception connecting
```

```
device:%s" % str(e))
```

```
    sys.exit()
```

```
# connect and send a
```

```
datapoint "hello"with value
```

```
"world" info the cloud as
```

```
an event of
```



```
type "greetings" 10 times
```

```
deviceCli.connect()
```

```
while True:
```

```
    #Get sensor Data
```

```
    from DHT11
```

```
    Temperature=random.rand
```

```
    int(90,110)
```

```
    pH=random.randint(0,14)
```

```
    Turbidity=random.randint(
```

```
    0,100)
```

```
    data = { 'Temperature'
```

```
            : Temperature, 'pH': pH,
```

```
            'Turbidity':Turbidity }
```

```
    def SMS():
```

```
        message = Client.messages\
```

```
            .create(
```

```

        body="Alert! Alert!! River water quality is getting
degraded",

        from_=twilio_number,

        to = target_number)

    print(message.body)

    if(temp>80 or pH<6 or turbidity>70):

        SMS()

#print data

def

myOnPublishCallback():

    print ("published

Temperature = %s C" %

Temperature, "pH = is %s

%%" % pH, "Turbidity= is

%s %%" % Turbidity,"to

IBM Watson")

    success =

    deviceCli.publishEvent("IO

TSensor",

    "json",data,qos=0,on_publi

```

```
sh=myOnPublishCallback)
```

```
if not success:
```

```
    print("Not
```

```
connected to IOTF")
```

```
    time.sleep(10)
```

```
deviceCli.commandCallba
```

```
ck = myCommandCallback
```

```
# Disconnect the device
```

```
and application from the
```

```
cloud
```

```
deviceCli.disconnect()
```

**GITHUB LINK:** <https://github.com/IBM-EPBL/IBM-Project-14021-1659538996>

**VIDEO DEMONSTRATION LINK:**

**DEMO VIDEO LINK:**

- **Drivelink:** [https://drive.google.com/file/d/16MgFyCdwpLmEq4lU61V-jVuWgkHA6msP/view?usp=share\\_link](https://drive.google.com/file/d/16MgFyCdwpLmEq4lU61V-jVuWgkHA6msP/view?usp=share_link)
- **Youtube link:** <https://youtu.be/bj6H7SAKOFg>

## **LINKEDIN PROFILE OF TEAM MEMBERS**

- Indira Priyadharshini.V :  
<https://www.linkedin.com/in/indirapriyadharshiniv>
- Vaishnavi.G :  
<https://www.linkedin.com/in/vaishnavi-g-567aa023a>
- Kalpana.S:  
<https://www.linkedin.com/in/kalpana-sivakumar-1223aa238>