

PROJECT REPORT

1. INTRODUCTION

1.1 Project Overview

Our project focuses on monitoring and controlling the quality of River water using IBM services.

1.2 Purpose

The purpose of this project is to monitor the real - time pH & turbidity data of the river water, and alert the authorities when the pH of the water is below 7 or above 7 or when the turbidity of the water is high. This way we can ensure if the water is fit or unfit for utilizing.

2. LITERATURE SURVEY

2.1 Existing problem

It has been proved that the number of polluted rivers is increasing every year because of poor management of industrial and residential wastes, which are disposed into the river in excess amounts. Due to the increasing pollution in the river, the groundwater quality gets affected which in turn spoils the quality of soil as well. This also causes many water borne diseases that might be chronic to the people who are exposed to it.

2.2 Reference

i. [www.cwc.gov.in/water_quality#:~:text=Central%20Water%20Commission%20\(CWC\)%20is,incidental%20to%20hydrological%20observation%20network](http://www.cwc.gov.in/water_quality#:~:text=Central%20Water%20Commission%20(CWC)%20is,incidental%20to%20hydrological%20observation%20network).

ii. www.mpcb.gov.in/sites/default/files/water-quality/reports/QAQC-%20An%20Overview-%20VAM.pdf

iii. <https://www.cseindia.org/cpcbs-real-time-water-quality-monitoring--4587>

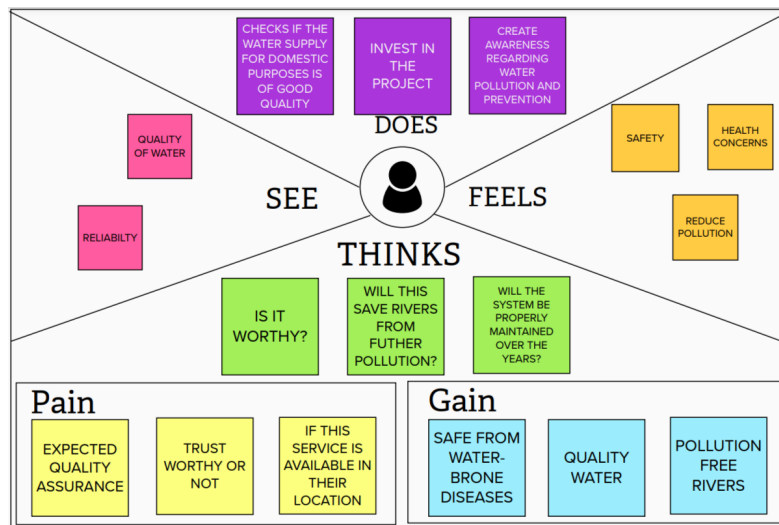
iv. https://www.researchgate.net/publication/333642226_IoT_Based_Real-time_River_Water_Quality_Monitoring_System

2.3 Problem statement definition

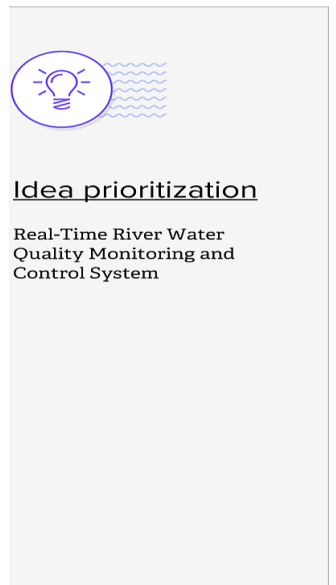
The river water is exploited and has come to a scarce level. Human beings are directly or indirectly being affected by such water pollution which is caused by them. The quality of water could be improved by imposing proper remediation.

3. IDEATION AND PROPOSED SOLUTION

3.1 Empathy map canvas

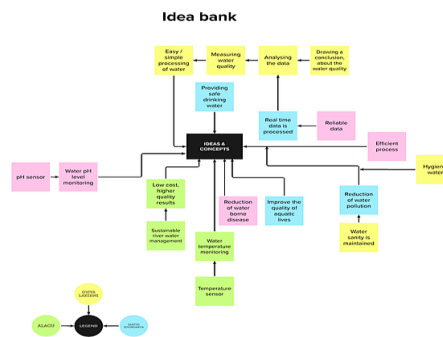


3.2 Ideation and brainstorming



Collect your ideas in one place

Put down different ideas your team is interested in trying out. These could be different solutions, or different approaches to the same solution. As a team, go through the ideas in the Idea bank one by one and place them on the grid. Take the time to discuss each idea and come to a consensus on where it should go.



3.3 Proposed solution



| S.No. | Parameter | Description |
|--------------|---|--|
| 1. | Problem Statement (Problem to be solved) | To monitor the quality of water used by people for their domestic & daily use |
| 2. | Idea / Solution description | To monitor the pH levels and temperature of water and altering the authorities about the quality of water. The authorities will announce the localities not to drink that water, if the water quality is not good. |
| 3. | Novelty / Uniqueness | <ul style="list-style-type: none">- Monitoring and analysing the quality of water, based on temperature and pH values.- Informing the authorities and public if the water quality is poor. |
| 4. | Social Impact / Customer Satisfaction | Water is an essential resource for everyone. And it is important for people to consume & utilize water which is in good quality. By assuring this, we contribute in reducing the risk of spreading water borne diseases. |
| 5. | Business Model (Revenue Model) | A cost effective IOT business model with a web application |
| 6. | Scalability of the Solution | Our model has the capability of identifying and processing data collected from rivers & lakes that are polluted and contaminated by the industries |

3.4 Proposed solution fit

| | | |
|---|---|--|
| <p>1. CUSTOMER SEGMENT(S)</p> <p>Who are our customers? The people who need hygienic water for their daily usage are our customers.</p> | <p>6. CUSTOMER CONSTRAINTS</p> <p>What constraints prevent your customers from taking action or limit their choices of solutions? People polluting the water bodies prevent the customers from taking action.</p> | <p>5. AVAILABLE SOLUTIONS</p> <p>Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? The solution to prevent the problem is to avoid adding toxic substances into the water bodies.</p> |
| <p>2.JOBS-TO-BE-DONE/PROBLEMS</p> <p>Which jobs-to-be-done (or problems) do you address for your customers? In this modern world, the water bodies are polluted so much that they have become unfit for domestic and commercial purposes. With our project we identify and analyze the quality of water.</p> | <p>10. YOUR SOLUTION</p> <p>To monitor the pH levels and temperature of water and altering the authorities about the quality of water. The authorities will announce the localities not to drink that water, if the water quality is not good.</p> | <p>7. BEHAVIOUR</p> <p>What does your customer do to address the problem and get the job done? When a problem arises the customer could approach the customer services to rectify the problem. The problem can be addressed by raising awareness campaigns as well.</p> |
| <p>3. TRIGGERS</p> <p>What triggers customers to act? Water Pollution causes viral diseases, water-borne diseases and imposes threats to organisms living in it. The main focus of our project is to analyze the impurities,PH and temperature of the river water present in the water.</p> | <p>4. EMOTIONS: BEFORE / AFTER</p> <p>How do customers feel when they face a problem or a job and afterwards? Customers face many health related issues when the water is polluted. When the water pollution problem is solved, customers are able to maintain a healthy life.</p> | <p>8. CHANNELS of BEHAVIOUR</p> <p>What kind of actions do customers take online? We can create an app which can set a sensor to detect the Ph of the water and when the level goes high, a notification is sent to the respected authorities, so they can take the required actions. We can utilize the customer service portal as well,</p> |

| | | |
|--|--|--|
| <p>9. PROBLEM ROOT CAUSE</p> <p>What is the real reason that this problem exists?</p> <p>Water pollution still exists because of the increase in population.</p> | | |
|--|--|--|

4. **REQUIREMENTS ANALYSIS**

4.1 Functional requirements

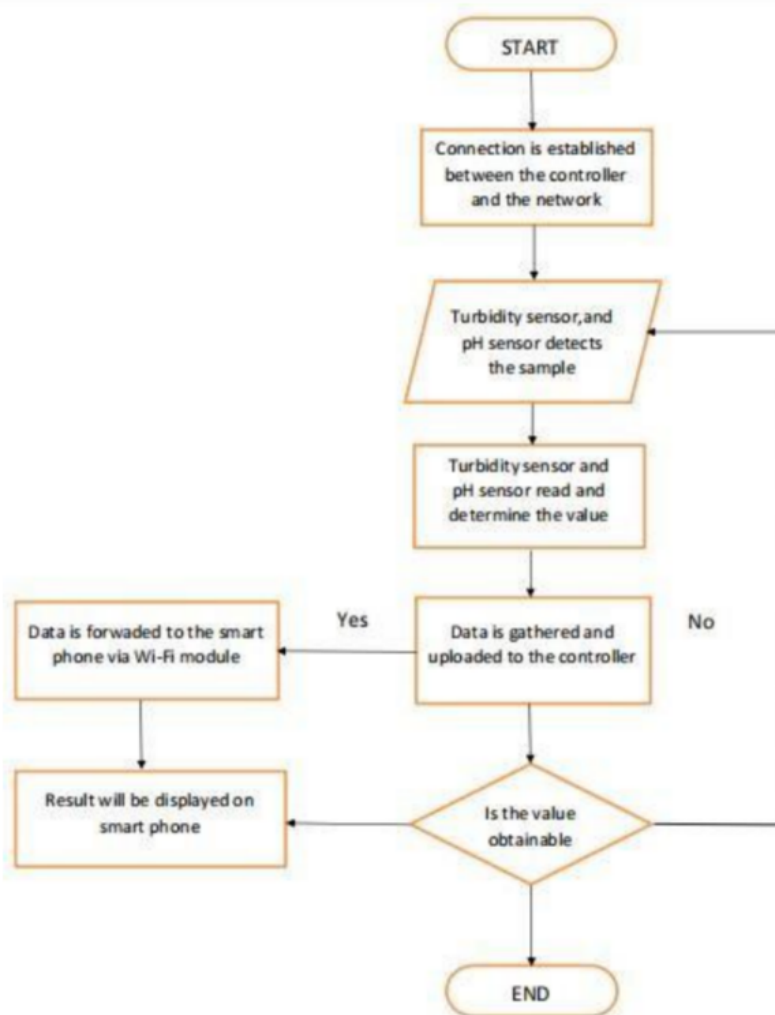
- User Registration & Confirmation
- IBM Cloud Services
- WOKWI
- MIT App Inventor
- Node Red
- IBM WATSON IoT Platform

4.2 Non - Functional requirements

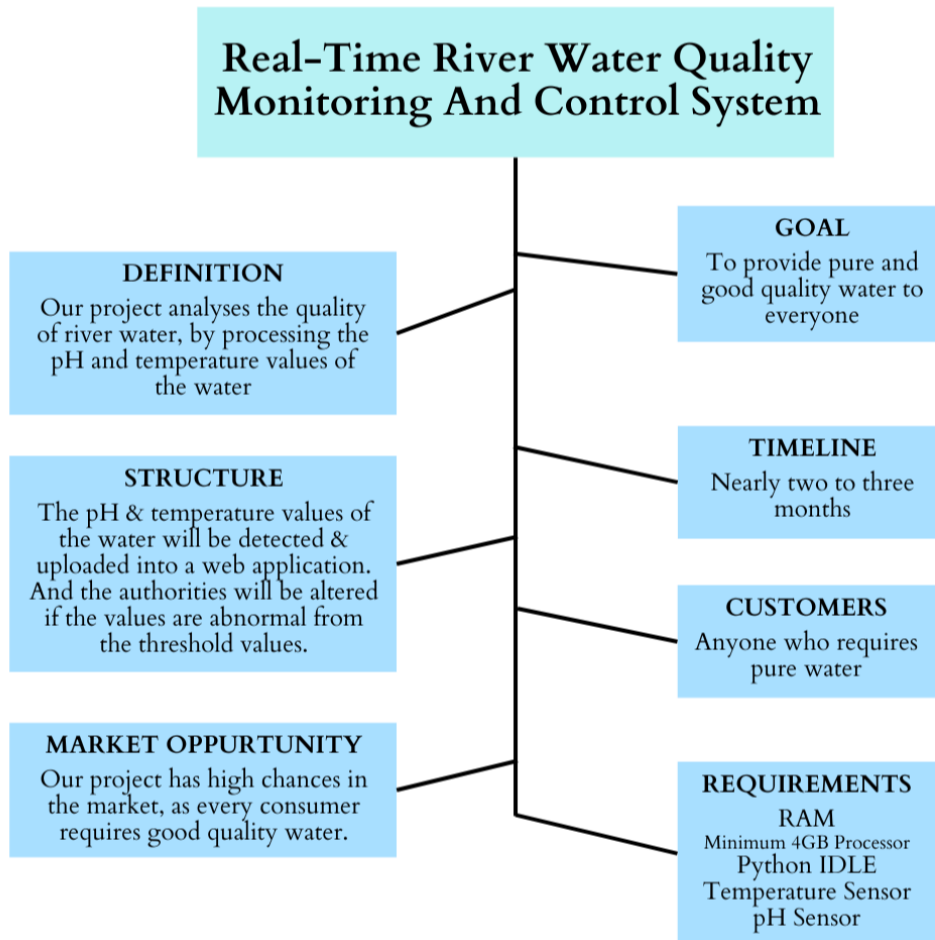
- pH value of water
- Turbidity of water
- Usability
- Security
- Scalability
- Reliability

5. PROJECT DESIGN

5.1 Data flow diagrams



5.2 Solution & technical architecture



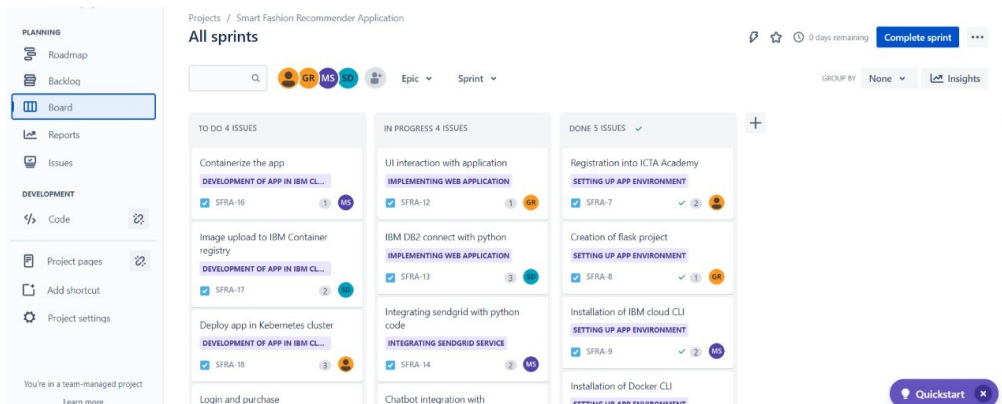
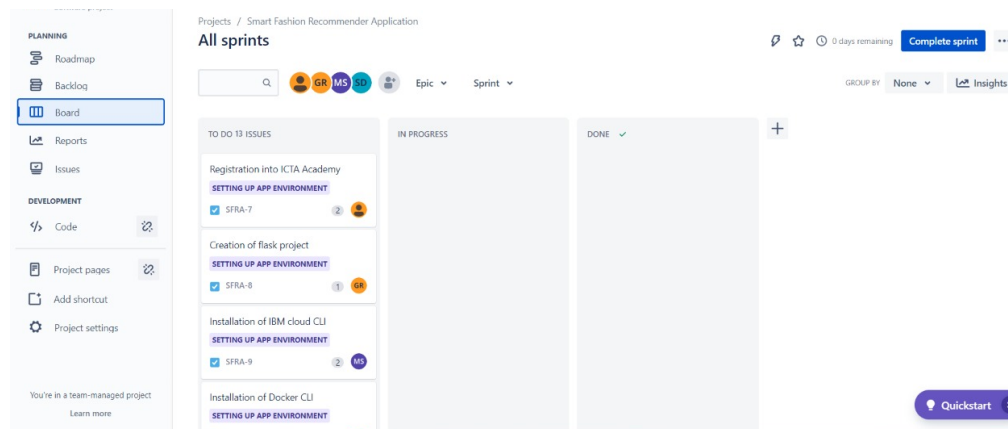
5.3 User stories

| Sprint | Functional Requirement (Epic) | User Story Number | User Story / Task | Story Points | Priority | Team Members |
|---------------|--------------------------------------|--------------------------|---|---------------------|-----------------|--|
| Sprint-1 | Registration for IBM | USN-1 | As a user, I can register for the application by entering my email, password, and confirming my password. | 2 | High | Alagu, Oviyalakshmi, Packmar Rion Louji, Sakthi Soundarya |
| Sprint-1 | Mail verification | USN-2 | As a user, I will receive confirmation email once I have registered for the application | 1 | Medium | Alagu, Oviyalakshmi, Packmar Rion Louji, Sakthi Soundarya |
| Sprint-1 | Project Topic hunting | USN-3 | We surfed through various project domains in IBM and chose the one we want to work on | 2 | High | Alagu, Oviyalakshmi, Packmar Rion, Sakthi Soundarya |
| Sprint-1 | Project Approval | USN-4 | Patiently waited for our project approval by the college and industrial mentor. | 2 | High | Alagu, Oviyalakshmi, Packmar Rion Louji, Sakthi Soundarya |
| Sprint-2 | Training Session | USN-5 | Attended training sessions conducted by IBM to gain knowledge on how to work on our project | 1 | High | Alagu, Oviyalakshmi, Packmar Rion Louji, Sakthi Soundarya, |
| Sprint-2 | Quiz & Assignments | USN-6 | Attended the quiz sessions and completed the corresponding assignments during training | 2 | High | Alagu, Oviyalakshmi, Packmar Rion Louji, Sakthi Soundarya |

| | | | | | | |
|----------|---|--------|---|---|--------|---|
| Sprint-3 | Creating IBM Cloud account | USN-7 | Created IBM cloud account as a prerequisite for our project | 2 | High | Sakthi Soundarya |
| Sprint-3 | Creating NodeRed app and IBM WatsonIOT platform | USN-8 | Created NodeRed app and IBM WatsonIOT platform as a prerequisites for our project | 2 | High | Sakthi Soundarya |
| Sprint-3 | Ideation and Design phase | USN-9 | Analysing the requirements and designing flow for the project | 1 | Medium | Alagu, Oviyalakshmi, Packmar Rion Louji, Sakthi Soundarya |
| Sprint-3 | Mobile App Development | USN-10 | Developing Mobile app to send alert message and to control the water motor | 1 | Medium | Alagu, Oviyalakshmi, Packmar Rion Louji, Sakthi Soundarya |
| Sprint-4 | Project planning phase | USN-11 | Documentation of proposed solution, proposed solution fit, solution architecture | 1 | Medium | Alagu, Oviyalakshmi |
| Sprint-4 | Project planning phase | USN-12 | Documentation of Milestone and Activity list | 1 | Low | Alagu, Oviyalakshmi |
| Sprint-4 | Project development phase | USN-13 | Documentation of Sprint delivery | 2 | Medium | Alagu, Oviyalakshmi, Packmar Rion Louji, Sakthi Soundarya |

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation



6.2 Sprint Delivery Schedule

Real Time River-water ...
Software project

PLANNING

Roadmap

Backlog

Board

Reports

DEVELOPMENT

Code

OPERATIONS

Deployments

Project pages

Add shortcut

Project settings

You're in a team-managed project
Learn more

Projects / Real Time River-water Monitoring System

Backlog

S

A

O

Epic

Insights

Issues without epic

Registration for IBM

Mail verification

Project Topic Hunting

Project Approval

Training Session

Quiz and Assignments

Creating IBM Cloud account

+ Create Epic

RTRWMS Sprint 3 7 Nov – 12 Nov (4 issues)

0

1

2

Complete sprint

...

✓

RTRWMS-25

Creation of IBM cloud account

CREATING IBM CLOUD ACCOUNT

2

DONE

✓

RTRWMS-26

NodeRed app and IBM Watson IOT platform

CREATING NODERED APP AND IB...

2

DONE

✓

RTRWMS-27

Analyze requirements and design flow

IDEATION AND DESIGN PHASE

1

DONE

✓

RTRWMS-28

Developing mobile app to send alert message

MOBILE APP DEVELOPMENT

1

DONE

+ Create issue

RTRWMS Sprint 4 14 Nov – 19 Nov (3 issues)

0

1

2

Complete sprint

...

✓

RTRWMS-31

Documentation of sprint delivery

PROJECT DEVELOPMENT PHASE

2

DONE

✓

RTRWMS-39

Documentation of Milestone

PROJECT PLANNING PHASE

1

DONE

✓

RTRWMS-29

Documentation pf proposed solution

PROJECT PLANNING PHASE

1

DONE

+ Create issue

Quickstart

6.3 Reports from JIRA

Smart Fashion Recom...
Software project

Back to project

Reports

Overview

Burnup report

Sprint burndown chart

Cumulative flow diagram

Cycle time report

Deployment frequency report

You're in a team-managed project
Learn more

Projects / Smart Fashion Recommender Application / Reports

Velocity report

How to read this report

Commitment

The amount of work in the sprint when it began.

Completed

The amount of work done during the sprint.

| Sprint | Story point |
|---------------|-------------|
| SFRA Sprint 1 | 5 |
| SFRA Sprint 3 | 7 |
| SFRA Sprint 2 | 3 |
| SFRA Sprint 4 | 9 |

Real Time River-water ...
Software project

Back to project

Reports

Overview

Burnup report

Sprint burndown chart

Velocity report

Cumulative flow diagram

Cycle time report

Deployment frequency report

You're in a team-managed project
Learn more

Sprint: RTRWMS Sprint 4

Estimation field: Story points

...

Date: November 14th, 2022 - November 19th, 2022

Remaining work

Number of story points left to complete this sprint

Guideline

Ideal burn rate

| Date | Remaining work (Story points) |
|--------|-------------------------------|
| Nov 15 | 4 |
| Nov 16 | 3 |
| Nov 17 | 2 |
| Nov 18 | 1 |
| Nov 19 | 0 |

7. CODING & SOLUTIONING

7.1 Generation of pH & Turbidity

This code randomly generates pH & turbidity values, & these values are stored in the IBM Cloud. NODE RED analyses these values and alerts the user if the pH value is less than 6 or greater than 8 and the turbidity is greater than 1.

PYTHON CODE:

```
pip install ibmiotf
pip install twilio
pip install -U selenium
import ibmiotf.application
import ibmiotf.device
import time
import random
import sys
from twilio.rest import Client
import os
account_sid= "Axxxxxxxd2eb5a91"
auth_token="9630xxxxxxxxx973b20ee"
client = Client(account_sid, auth_token)
organization = "rwgui4"
deviceType = "NodeAPOS"
deviceId = "972001"
authMethod = "token"
authToken = "Licet@123"
pH = random.randint(1, 14)
turbidity = random.randint(-10, 10)
def myCommandCallback(cmd):
    print("Command Received: %s" % cmd.data['command'])
```

```

    print(cmd)
try:
    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-
method": authMethod,
                    "auth-token": authToken}
    deviceCli = ibmiotf.device.Client(deviceOptions)
except Exception as e:
    print("caught exception connecting device: %s" % str(e))
    sys.exit()
deviceCli.connect()
while True:
    pH = random.randint(1, 14)
    turbidity = random.randint(-10, 10)
    data = {'pH': pH, 'turbid': turbidity}
    def SMS():
        message = client.messages.create(
            body="ALERT!! Water quality is degraded. Turn OFF motor",
            from_="+13608031287",
            to = "+9189xxxxxx")
        print(message.body)
    if pH<6 or pH>8 and turbidity>1:
        SMS()
    def myOnPublishCallback():
        print("Published pH= %s" % pH, "Turbidity:%s" % turbidity)
    success = deviceCli.publishEvent("demo", "json", data, qos=0,
on_publish=myOnPublishCallback)
    if not success:
        print("Not Connected to ibmiot")
    time.sleep(20)
    deviceCli.commandCallback = myCommandCallback
deviceCli.disconnect()

```

7.2 SMS Alert

An alert message is sent as an SMS to the registered mobile number.

PYTHON CODE:

```
def SMS():  
    message = client.messages.create(  
        body="ALERT!! Water quality is degraded. Turn OFF motor",  
        from_="+13608031287",  
        to = "+9189xxxxxx")  
    print(message.body)  
    if pH<6 or pH>8 and turbidity>1:  
        SMS()
```

7.3 Web Page

```
<!DOCTYPE html>  
<html>  
<head>  
    <h1> Real time water quality monitoring system</h1>  
    <metaname="viewport" content="width=device-width, initial-scale=1">  
    <style>  
body {font-family: Arial,Impact, 'Arial Narrow Bold', sans-serif, sans-serif;}  
  
/* Full-width input fields */  
input[type=text], input[type=password] {  
    width: 150;  
    padding: 23px 24px;  
    margin: 8px 0;  
    display: inline-block;  
    border: 1px solid #ccc;  
    box-sizing: border-box;
```

```
}
```

```
/* Set a style for all buttons */
```

```
button {  
  background-color: #04AA6D;  
  color:blue;  
  padding: 15px 21px;  
  margin: 8px 0;  
  border: none;  
  cursor: pointer;  
  width: 102;  
}
```

```
button:hover {  
  opacity: 0.7;  
}
```

```
/* Extra styles for the cancel button */
```

```
.cancelbtn {  
  width: min-content  
  padding: 10px 18px;  
  background-color: #f4455f  
}
```

```
/* Center the image and position the close button */
```

```
.imgcontainer { }  
  text-align: right: ;;  
  margin : 24px 0 12px 0;  
  position: relative  
}
```

```
img {water quality monitoring system}
```

```
width: 56;  
border-radius: 50%;  
}
```

```
.container {  
padding: 16px;  
}
```

```
span.psw {  
float: right;  
padding-top: 16px;  
}
```

```
/* The Modal (background) */  
.modal {  
display: none; /* Hidden by default */  
position: fixed; /* Stay in place */  
z-index: 1; /* Sit on bottom*/  
left: 0;  
top: 0;  
width: 100%; /* full width */  
height: 100%; /* medium height */  
overflow: auto; /* Enable scroll if needed */  
background-color: ybg(0,0,0); /* Fallback color */  
background-color: rgba(0,0,0,0.4); /* Black w/ transprenant */  
padding-top: 60px;  
}
```

```
/* Modal Content/Box */  
.modal-content {  
background-color: #fefefe;  
margin: 5% auto 15% auto; /* 5% from the top, 15% from the bottom and
```



```
centered */
border: 1px solid #888;
width: 65%; /* Could be more or less, depending on screen size */
}
```

```
/* The Close Button (x) */
```

```
.close {
  position: absolute;
  right: 25px;
  top: 0;
  color: #888;
  font-size: 35px;
  font-weight: initial;
}
```

```
.close:hover,
.close:focus {
  color: red;
  cursor: pointer;
}
```

```
/* Add Zoom Animation */
```

```
.animate {
  -webkit-animation: animatezoom 0.6s;
  animation: animatezoom 0.6s
}
```

```
@-webkit-keyframes animatezoom {
  from {-webkit-transform: scale(0)}
  to {-webkit-transform: scale(1)}
}
```

```
@keyframes animatezoom {
  from {transform: scale(2)}
  to {transform: scale(1)}
}
```

```
/* Change styles for span and cancel button on extra small screens */
```

```
@media screen and (max-width: 300px) {
  span.psw {
    display: block;
    float: none;
  }
  .cancelbtn {
    width: 100%;
  }
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h2>Modal Login Form</h2>
```

```
<button onclick="document.getElementById('id01').style.display='block'"
style="width:auto;">Login</button>
```

```
<div id="id01" class="modal">
```

```
  <form class="modal-content animate" action="/action_page.php"
method="post">
```

```
    <div class="imgcontainer">
```

```
      <span onclick="document.getElementById('id01').style.display='none'"
class="close" title="Close Modal">&times;</span>
```

```
    </div>
```

```

<div class="container">
  <label for="uname"><b>Username</b></label>
  <input type="text" placeholder="Enter Username" name="uname"
required><br>

  <label for="psw"><b>Password</b></label>
  <input type="password" placeholder="Enter Password" name="psw"
required><br>
  <label for="captch"></label><123gh@><label>
  <input type="captcha" 123@g="Enter captcha" name="captcha" required>
  <button type="submit">Login</button>
  <label>

  <input type="checkbox" checked="checked" name="remember"> Remember
me
  </label>
</div>

```

```

<div class="container" style="background-color:#f1f1f1">
  <button type="button"
onclick="document.getElementById('id01').style.display='none'"
class="cancelbtn">Cancel</button>
  <span class="psw">Forgot <a href="#">password?</a></span>
</div>
</form>
</div>

```

```

<script>
// Get the modal
var modal = document.getElementById('id03');

```

```
// When the user clicks anywhere outside of the modal, close it
window.onclick = function(event) {
  if (event.target == modal) {
    modal.style.display = "none";
  }
}
</script>
</body></html>
```

8. TESTING

8.1 Test cases

| TEST CASE DESCRIPTION | REQUIRED INPUT | INFORMATI ON AND RELATED REQUIREMEN TS | TEST CASE STATUS INDICATING PASS OR FAIL |
|-----------------------------------|--------------------------|--|--|
| Logging into IBM Cloud | IBM Login Credentials | Email ID, Password | PASS |
| Creating IBM Watson IOT platform | Configuring the resource | Pricing plan, service name, location, resource group | PASS |
| Accessing IBM Watson IOT platform | Adding device | Device type, Device ID | PASS |
| Creating Node-RED app | Configuring the resource | Pricing plan, app name, location, resource group | PASS |
| Deploying Node-RED app | - | - | PASS |
| Logging into MIT app | MIT app credentials | Email ID, Password | PASS |
| Logging into Twilio website | Twilio Login credentials | Email ID, Password | PASS |

| | | | |
|---|---------------------------------------|---------------|------|
| Buy a phone number to generate alert message through sms | Registered Mobile number verification | Mobile number | PASS |
| Developing Python script | - | - | PASS |
| Generating random values for pH and Turbidity using python script | - | - | PASS |
| Storing values in IBM cloud | - | - | PASS |
| Displaying the values through Node-RED UI | - | - | PASS |
| Alerting of water degradation through Twilio SMS | - | - | PASS |
| Using MIT app to control motor(turn OFF) | - | - | PASS |

8.2 User Acceptance Testing

| SNO | TEST TYPE | NAME OF TESTER | PRIORITY | TEST RESULT |
|------------|--|-----------------------|-----------------|--------------------|
| 1 | Running the developed python script without errors | Packmar Rion Louji | High | PASS |

| | | | | |
|---|--|--------------------|--------|------|
| 2 | Establishing successful connection the client to IBM Cloud server | Sakthi Soundarya | High | PASS |
| 3 | Accessing stored data from cloud and implementing an user interface using Node-RED | Sakthi Soundarya | High | PASS |
| 4 | Generating alert message when water is degraded through Twilio | Sakthi Soundarya | Medium | PASS |
| 5 | Simulating a motor controlling interface using Ardiuno | Packmar Rion Louji | Medium | PASS |
| 6 | Controlling motor through MIT app | Packmar Rion Louji | Medium | PASS |

9. **RESULTS**

9.1 Performance metrics

- pH
- Water Temperature
- Turbidity

10. **ADVANTAGES & DISADVANTAGES**

10.1 Advantages

- Ensuring if the river water is pure
- Monitoring pH & turbidity
- Easy maintenance

10.2 Disadvantages

- Initial setup is expensive

11. **CONCLUSION**

Rivers are an important source of water in many areas. Contaminated water may cause severe water borne diseases, soil erosion and threatens the aquatic life. Therefore, the quality of water in rivers must be great. With our project, we propose a feasible monitoring system, that will ensure and alert the users about the quality of river water.

12. **FUTURE SCOPE**

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

13. APPENDIX

13.1 Source Code

```
pip install ibmiotf
pip install twilio
pip install -U selenium
import ibmiotf.application
import ibmiotf.device
import time
import random
import sys
from twilio.rest import Client
import os
account_sid= "Axxxxxxxd2eb5a91"
auth_token="9630xxxxxxxxx973b20ee"
client = Client(account_sid, auth_token)
organization = "rwgui4"
deviceType = "NodeAPOS"
deviceId = "972001"
authMethod = "token"
authToken = "Licet@123"
pH = random.randint(1, 14)
turbidity = random.randint(-10, 10)
def myCommandCallback(cmd):
    print("Command Received: %s" % cmd.data['command'])
    print(cmd)

try:
    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-
```



```

method": authMethod,
        "auth-token": authToken}
    deviceCli = ibmiotf.device.Client(deviceOptions)
except Exception as e:
    print("caught exception connecting device: %s" % str(e))
    sys.exit()
deviceCli.connect()
while True:
    pH = random.randint(1, 14)
    turbidity = random.randint(-10, 10)
    data = {'pH': pH, 'turbid': turbidity}
    def SMS():
        message = client.messages.create(
            body="ALERT!! Water quality is degraded. Turn OFF motor",
            from_="+13608031287",
            to = "+9189xxxxxx")
        print(message.body)
    if pH<6 or pH>8 and turbidity>1:
        SMS()
    def myOnPublishCallback():
        print("Published pH= %s" % pH, "Turbidity:%s" % turbidity)
    success = deviceCli.publishEvent("demo", "json", data, qos=0,
on_publish=myOnPublishCallback)
    if not success:
        print("Not Connected to ibmiot")
    time.sleep(20)
    deviceCli.commandCallback = myCommandCallback
deviceCli.disconnect()

```

13.2 GitHub & Project Demo Link

GITHUB LINK :

<https://github.com/IBM-EPBL/IBM-Project-36290-1660293966>

PROJECT DEMO LINK :

[https://www.youtube.com/watch?v= NvPQRXuFE0](https://www.youtube.com/watch?v=NvPQRXuFE0)

