PROJECT REPORT - TEAM ID: PNT2022TMID15102

REAL-TIME RIVER WATER QUALITY MONITORING AND CONTROL SYSTEM

TEAM MEMBERS:

1. DIVIT C – KCT19BEC150
2. PRASAD N – KCT19BEC140
3. RISHI VARDHAN S – KCT19BEC144
4. NANDHA KUMAR S – KCT19BEC210

1. INTRODUCTION:

1.1Project Overview:

Current water quality monitoring system is a manual system with a monotonous process and is very time-consuming. A sensor-based water quality monitoring system is proposed. The system consists of several sensors which is used to measure physical and chemical parameters of the water. The main components includes 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 Internet of Things (IoT) technology.

1.2Purpose:

Monitoring water quality is clearly important: in our seas, our rivers, on the surface and in our ports , for both companies and the public. It enables us to assess how they are changing, analyze trends and to inform plans and strategies that improve water quality and ensures that water meets its designated use. There are several indicators determining water quality. These include dissolved oxygen, turbidity, bio-indicators, nitrates, pH scale and water temperature. Monitoring water quality helps to identify specific pollutants, a certain chemical, and the source of the pollution. Identifying trends, short and long-term, in water quality. Environmental planning methods: water pollution prevention and management.

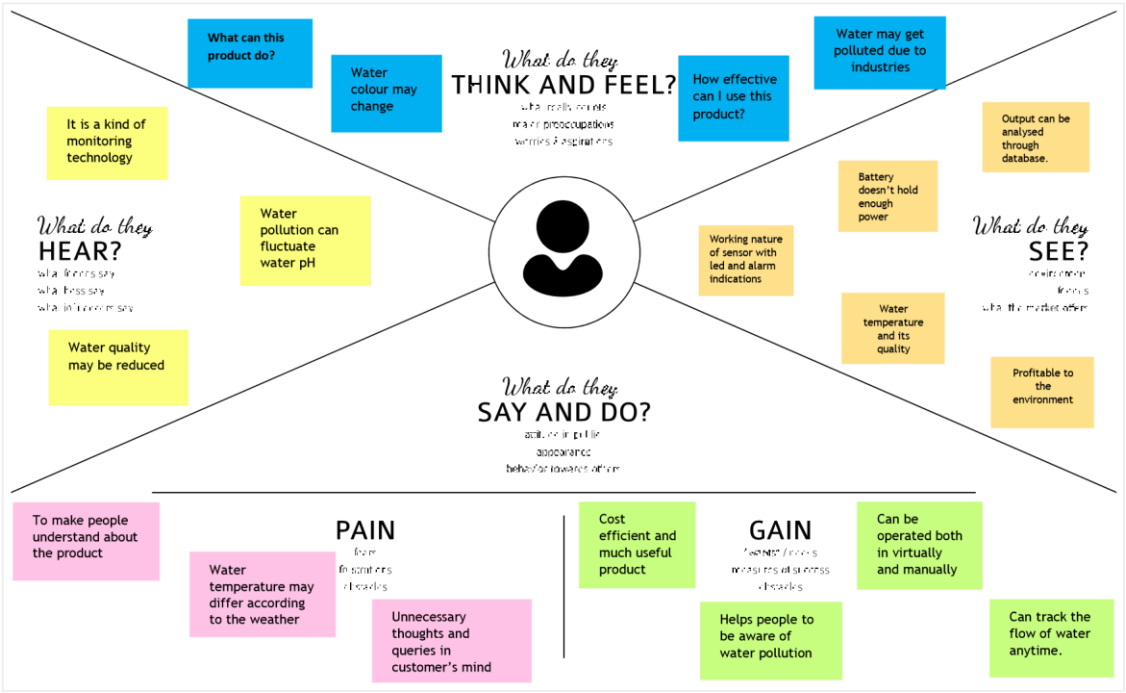
2. LITERATURE SURVEY:

S.NO	LINK	YEAR & JOURNAL	PAPER TITLE	AUTHOR NAME	SOLUTION
1.	Real-time water quality monitoring system using Internet of Things | IEEE Conference Publication | IEEE Xplore	2017 International Conference on Computer, Communications and Electronics (Comptelix)	Real-time water quality monitoring system using Internet of Things	B. Das and P. C. Jain	The water quality measuring system checks the quality of water in real time through various sensors (one for each parameter: pH, conductivity, temperature) to measure the quality of water. A ZigBee module is used in the system to transfer the data collected by the sensors to the microcontroller wirelessly, and a GSM module is used to transfer wirelessly the data further from the microcontroller to the smart phone/PC. The system also has proximity sensors to alert the officials by sending a message to them via the GSM module in case someone tries to pollute the water body.
2.	Elsevier Enhanced Reader	The 16th International Conference on Mobile Systems and Pervasive Computing (MobiSPC) August 19-21, 2019	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 Hossain	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 is to obtain the water monitoring system with high frequency, high mobility, and low powered.

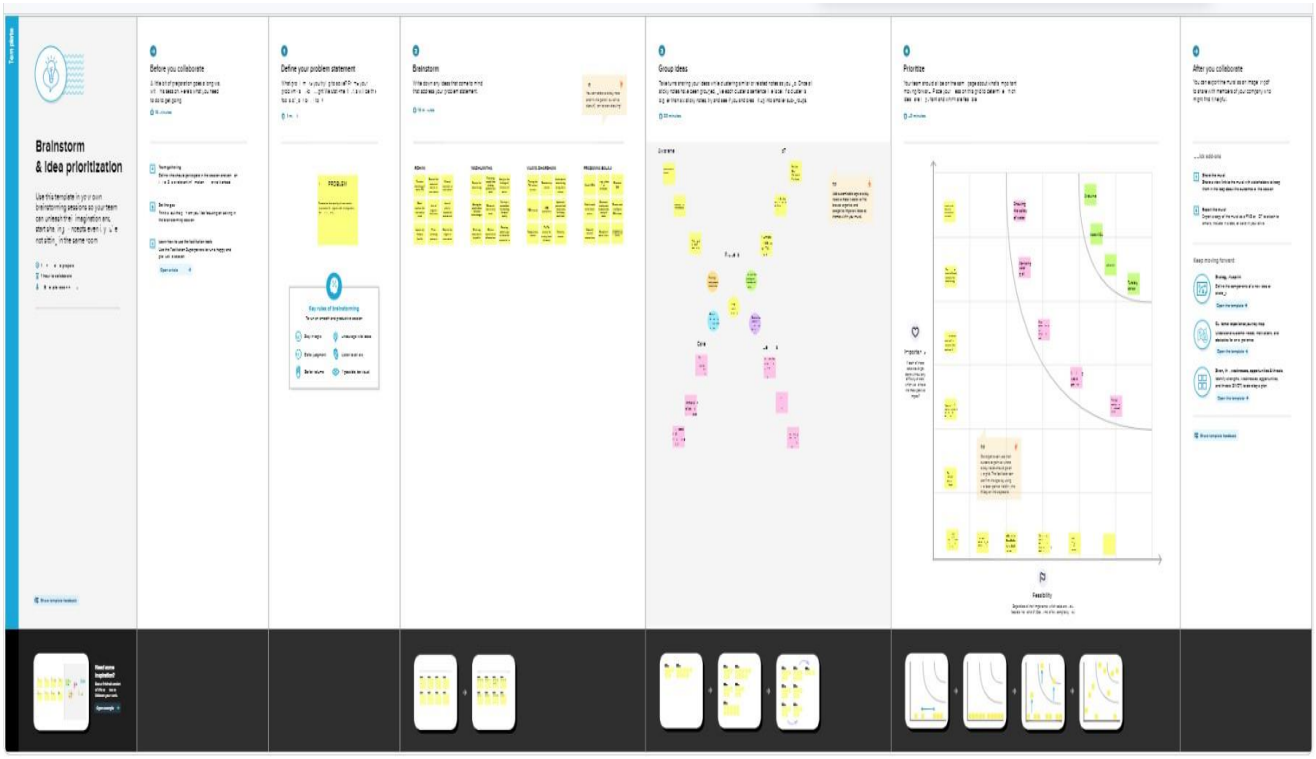
2.3 Customer problem statement:

3. IDEATION AND PROPOSED SOLUTION:

3.1 Empathy Map Canvas:



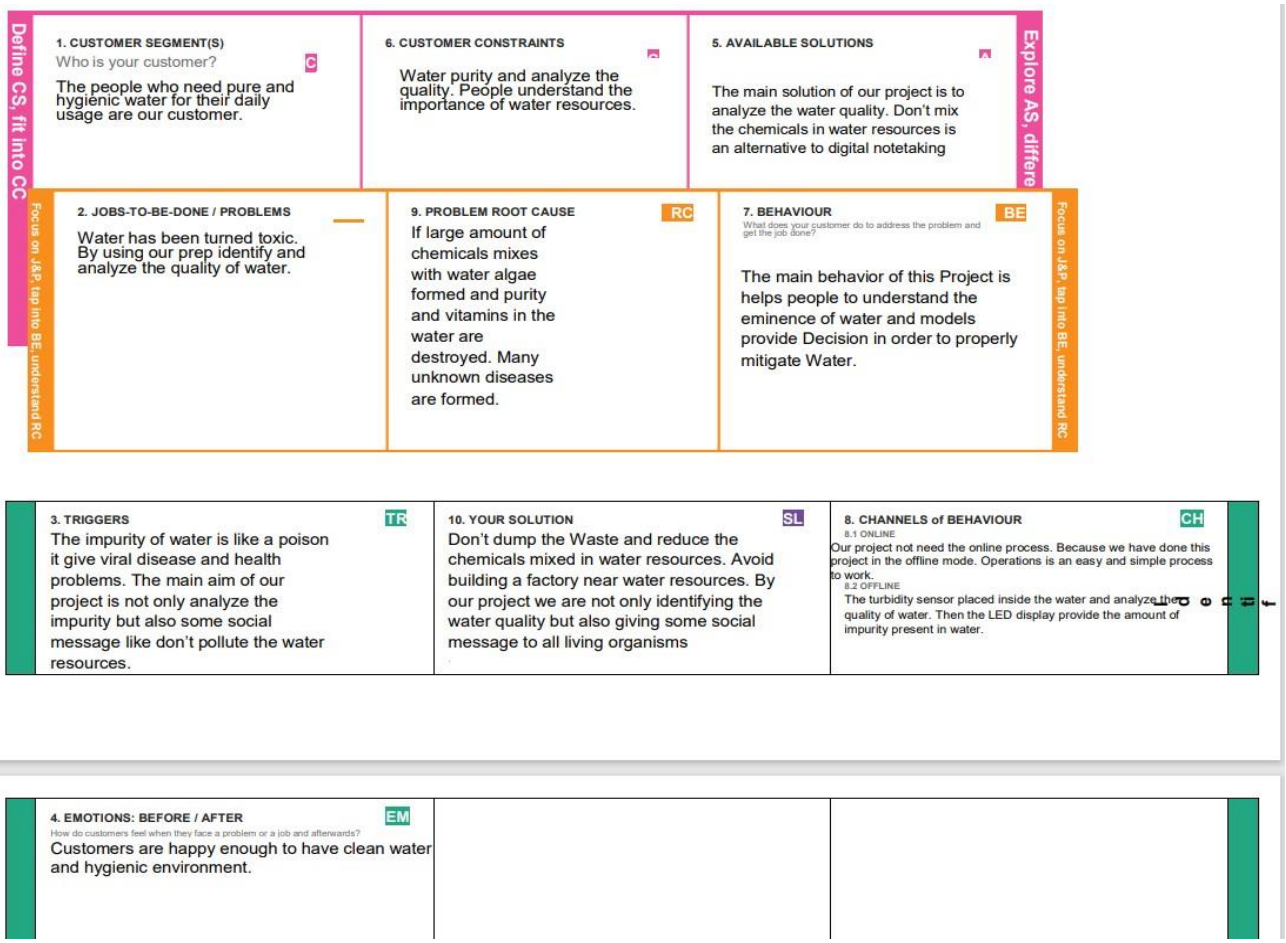
3.2 Ideation and Brainstorming:



3.3 Proposed Solution:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Monitoring the quality of water that used by people for their daily uses and control the pollution of the water.
2.	Idea / Solution description	Monitor the purity percentage that is present in water resources.
3.	Novelty / Uniqueness	Water is an important requirement in this world. So we are analysing using sensors and controlling the pollution by management.
4.	Social Impact / Customer Satisfaction	People are our customer because water is their daily needs. Our project not only satisfy the people but also provide social message to stop polluting water resources and also satisfying by providing necessary actions
5.	Business Model (Revenue Model)	A popular IOT business model is to build a medium cost and usable product/project for people.
6.	Scalability of the Solution	The capability of our project to identify and maintain the data about river sources that polluted by the industries.

3.4 Problem Solution fit:



4. REQUIREMENT ANALYSIS:

4.1 Functional Requirement:

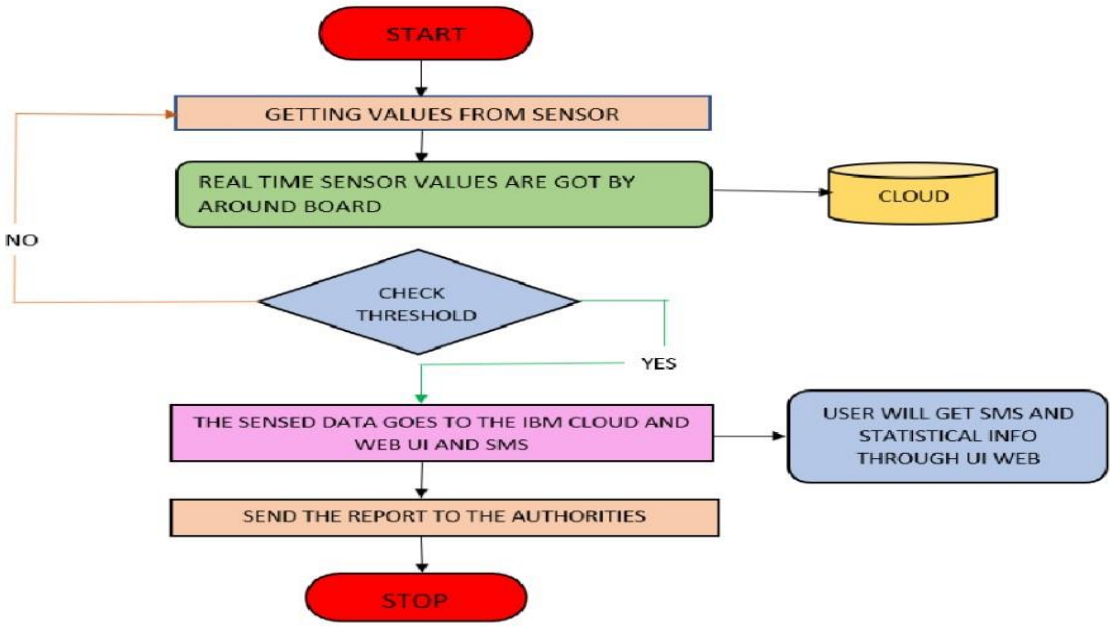
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Users Authorization levels	Complete mapping are given in a hierarchical manner in order to show only the specific Data.
FR-2	Historical Data	The Data are stored in the cloud from the beginning Stage till the Updation .
FR-3	User Authentication	The credentials is accessible only to the authorized users to access the model.
FR-4	Users rules and laws	There is some specific guidelines which has to be followed by the users.

4.2 Non-functional Requirement:

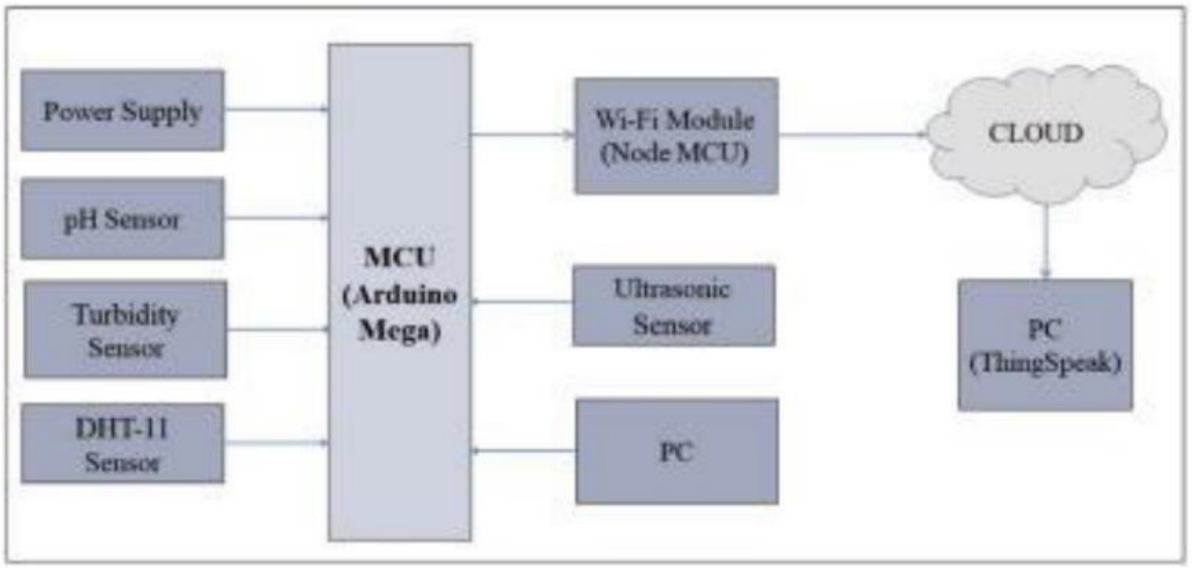
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The Final data should be easily understandable.
NFR-2	**Security**	The model are designed in a secured manner in order to maintain the privacy
NFR-3	**Reliability**	Even if there is a firmware issues (failures) the last updated Data's are stored in a Default manner.
NFR-4	**Performance**	High quality sensors are used to ease the customer's work.
NFR-5	**Availability**	The model are designed in such a way that are available ,usable and can be modified anytime.
NFR-6	**Scalability**	The System are Scaled according to the size of the water body (varies)

5. PROJECT DESIGN:

5.1 Data Flow Diagrams:



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
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering email, password, and confirming my password.	I can access my account/dashboard	High	Sprint-1
		USN-2	As a user, I will receive a confirmation email once I have registered for the application	I can receive e confirmation email & click confirm	High	Sprint 2
		USN-3	As a user, I can register for the application through Google	I can register & access the dashboard with Google	High	Sprint-1
		USN-4	As a user, I can register for the application through Gmail	I 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	I can receive login credentials.	High	Sprint-1
	Interface	USN-6	As a user, the interface should be user-friendly manner	I 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).	I 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)	I 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	I can easily know whether it is salty or not	High	Sprint-1
	Color visibility	USN-10	As a user , I can able predict the water color	I can easily know the condition by color	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 monitor the records properly.	Medium	Sprint-2

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, and password, and confirming my password.	3	High	Prasanna
Sprint-1	Confirmation Email	USN-2	As a user, I will receive a confirmation email once I have registered for the application	4	High	Rohini
Sprint-2	Authentication	USN-3	As a user, I can register for the application through Gmail and mobile app.	4	Medium	Madhumitha, Vijayadharshani
Sprint-2	Login	USN-4	As a user, I can log into the application by entering email & password	3	High	Prasanna, Madhumitha
Sprint-2	IBM Cloud Service Access	USN-5	Get access to IBM cloud services.	4	High	Vijayadharshani, Rohini
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	Madhumitha, Vijayadharshani
Sprint-2	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	Rohini, Prasanna
Sprint-3	Create a Web UI	USN-8	To create a Web UI, to access the data from the cloud And display all parameters.	6	Low	Madhumitha
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-3	To develop a Python code	USN-9	Create a python code to sense the physical quantity and store data	7	Medium	Vijayadharshani, Madhumitha
Sprint-3	Publish Data to cloud.	USN-10	Publish Data that is sensed by the microcontroller to the Cloud	7	High	Rohini
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	6	High	Prasanna, Vijayadharshani, Rohini
Sprint-4	Testing	USN-12	Testing of project and final deliverables	7	High	Prasanna, Vijayadharshani, Madhumitha

6.2 SPRINT DELIVERY SCHEDULE:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planne l)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29)ct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

7.CODING & SOLUTIONING

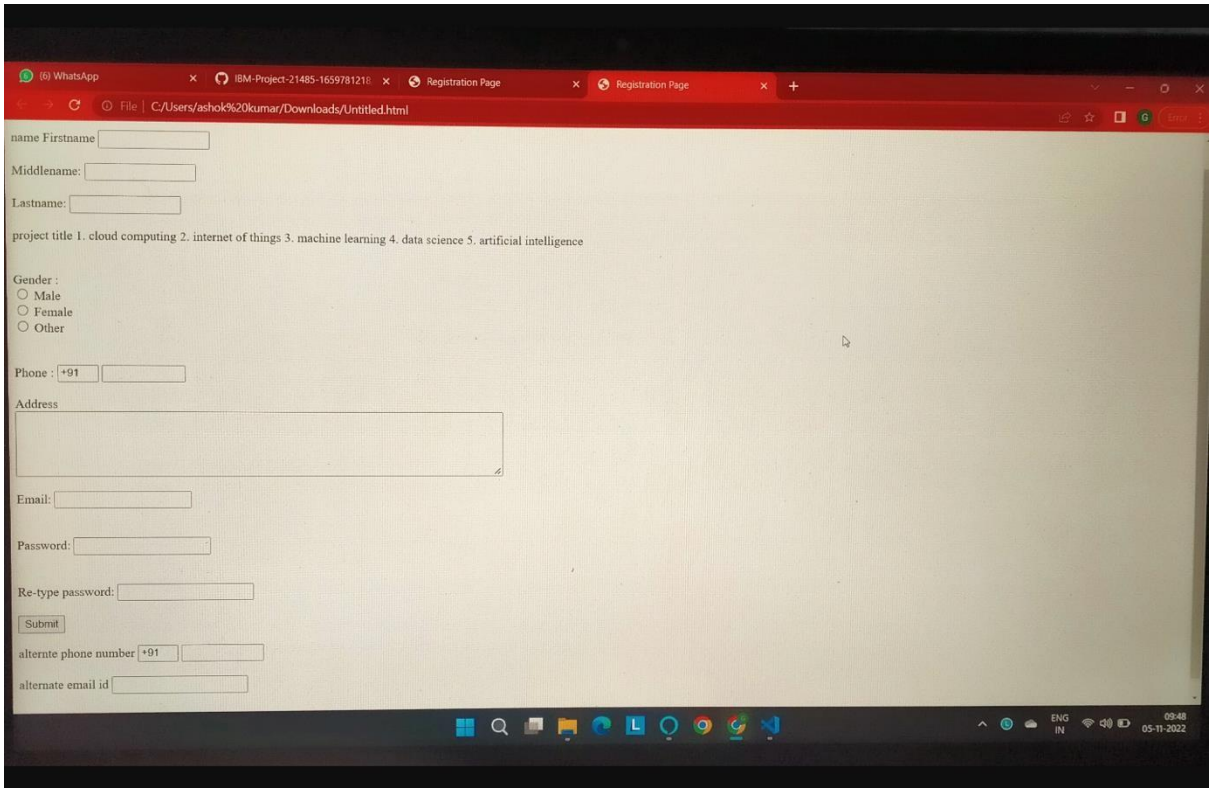
7.1Feature code 1:

This code is written to add the feature of registration page for the user application for getting real time data and notifications

```
<html>
  <head>
    <title>
    Registration Page
    </title>
    </head>
    <body>
    <br>
    <br>
<form>
name
    <label> Firstname </label>
    <input type="text" name="firstname" size="15"/> <br> <br>
    <label> Middlename: </label>
    <input type="text" name="middlename" size="15"/> <br> <br>
```

```
    <label> Lastname: </label>
    <input type="text" name="lastname" size="15"/> <br> <br>
    </select>
project title
    1.<label> cloud computing </label>
    2.<label> internet of things </label>
    3.<label> machine learning </label>
    4.<label> data science </label>
    5.<label> artificial intelligence </label>
     <br>
    <br>
    <br>
<label>
Gender:
    </label><br>
    <input type="radio" name="male"/> Male <br>
    <input type="radio" name="female"/> Female <br>
    <input type="radio" name="other"/> Other
    <br>
    <br>
     <br>
    <label>
Phone:
    </label>
    <input type="text" name="country code"  value="+91" size="2"/>
    <input type="text" name="phone" size="10"/> <br> <br>
    Address
    <br>
    <textarea cols="80" rows="5" value="address">
```

```
    </textarea>
    <br> <br>
Email:
    <input type="email" id="email" name="email"/> <br>
    <br> <br>
Password:
    <input type="Password" id="pass" name="pass"> <br>
    <br> <br>
    Re-type password:
    <input type="Password" id="repass" name="repass"> <br> <br>
    <input type="button" value="Submit"/>
    </form>       </body>
alternte phone number
    <input type="text" name="country code" value="+91" size="2"/>
<input type="text" name="phone" size="10"/> <br> <br>    alternate
email id
    <input type="altrernate email id" name="alternate email"/> <br>
    <br> <br>
    <body>
     <html>
```



7.2Feature code 2:

This code is written to add the feature of log in/sign up page for the user application for getting real time data and notifications

```
<!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 */
```

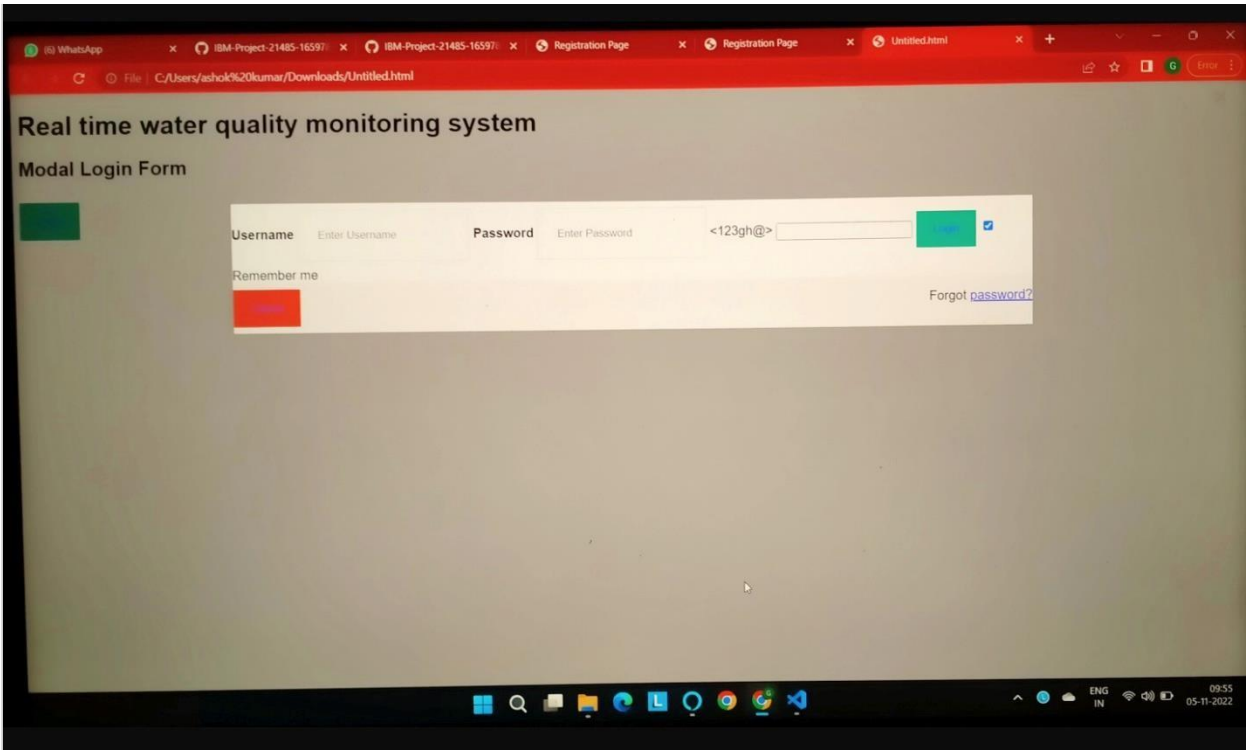
```
.imgcontainer { }   text-
align: right: ;;   margin :
24px 0 12px 0;   position:
relative
}

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>

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

8.1User Acceptance Testing

Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the web page for real time river water quality monitoring and control system project at the time of the release to User Acceptance Testing (UAT).

Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	9	4	2	3	20
Duplicate	2	0	3	0	4
External	3	3	0	1	6
Fixed	10	2	4	18	37

Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	1	3	2	1	8
Totals	24	12	13	24	78

Test Case Analysis

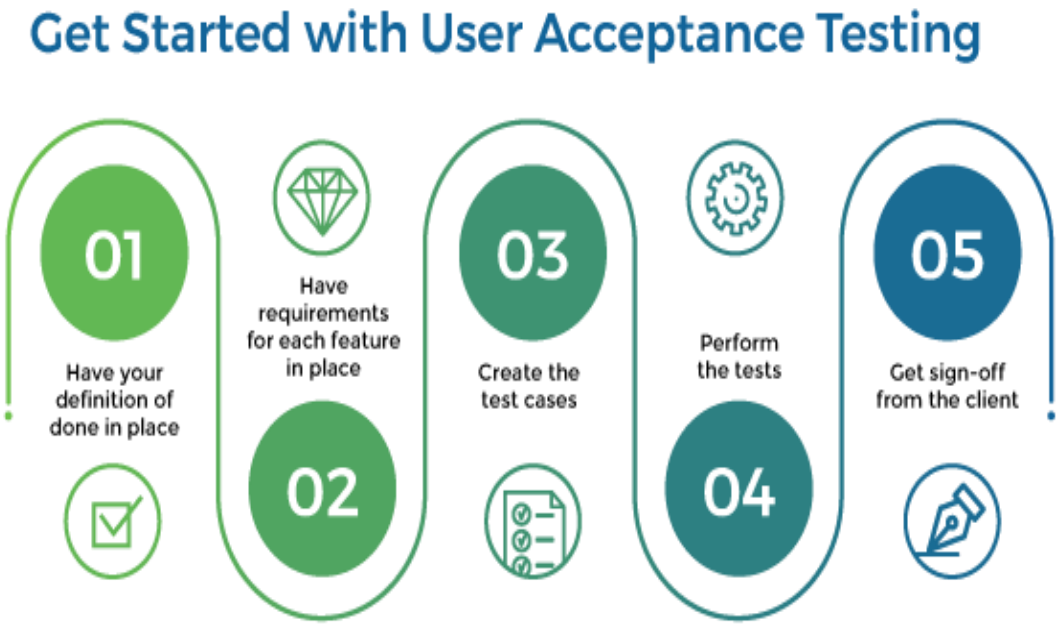
This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Client Application	40	0	2	38
Security	2	0	0	2
Application login	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	3	0	0	3
Version Control	4	0	0	4

8.2Testing tools

User Acceptance Testing

User acceptance testing (UAT) is necessary when implementing changes to an IT landscape due to the ever increasing complexity of software which can cause bugs to slip through even under the most perfect development conditions. Commonly, acceptance testing is performed as the last step before the release of a software, after all other testing phases have exited. As implied by its name, user acceptance testing is typically performed by the end users in a real setting during the unit-, integration- and system testing phases.



Test planning, monitoring and control

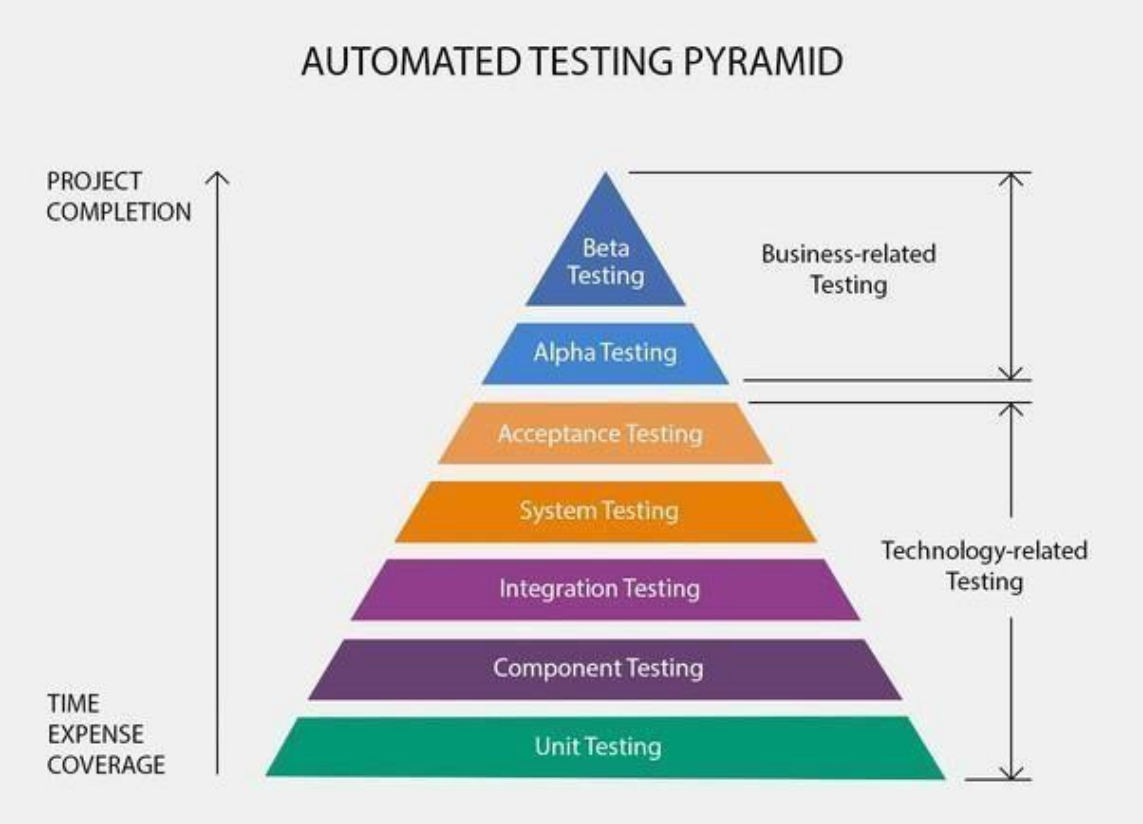
This step entails planning and setting up the base for the execution of the tests. Setting goals and identifying the aim(s) of the test phase based on the requirements supplied**.**

Test cases

In the context of this thesis it is important to note the use of subtle differences in the definition of automated testing. In literature we often find mentions of how automated test cases can require a lot of manual effort coding cases.

Automated Acceptance Testing

Automated acceptance testing (AAT) is the practice of executing business logic test cases automatically and can serve as an addition to user acceptance testing . Some sources also discuss the automation of user acceptance testing, in which case the automated acceptance tests are UAT tests as described above. It mostly revolves around the idea that businesses which procure software define a set of rigorous requirements with clearly defined input and output data. When this is the case, test cases can be automated using a variety of tools making use of UI selectors or visual user interface automation . This means that scripts could be written to automate user acceptance test cases with the intent to reduce the manual effort emerging from the repeated execution of acceptance testing.



9.RESULTS:

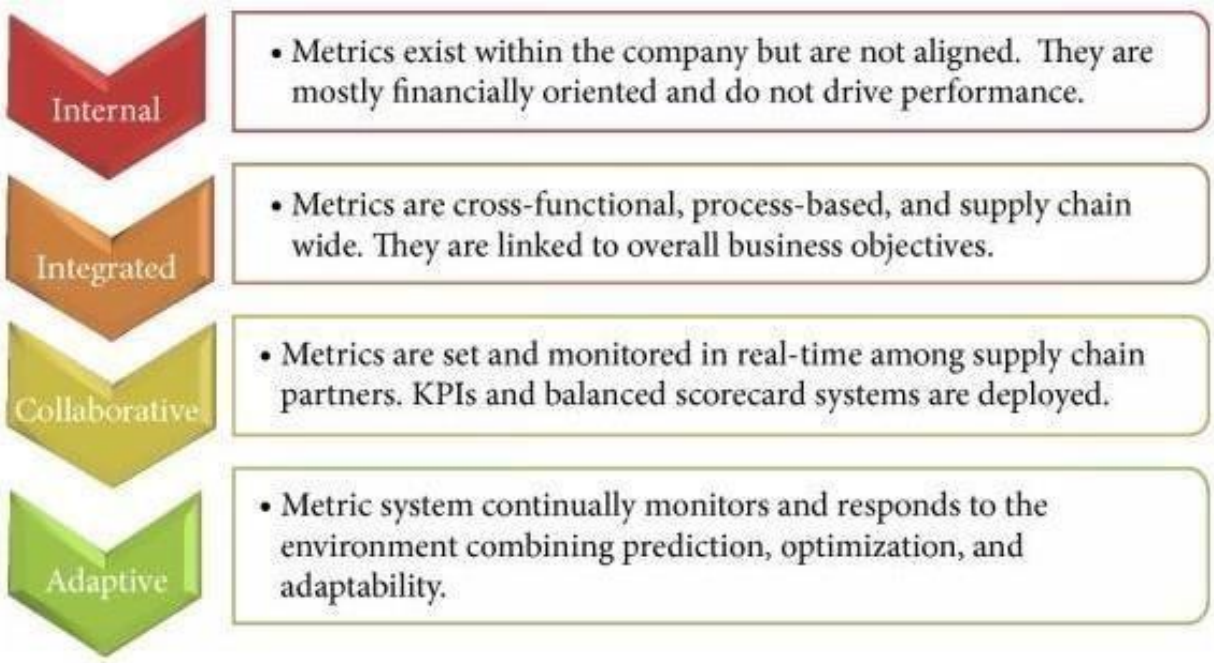
9.1Performance Metrics:

COLLECTION OF PERFORMANCE MEASUREMENTS

Managing application performance requires the continuous collection of data about all relevant parts of the system starting from the end user all the way through the system. This collected data is the basis for getting a holistic end-to-end and up-to-date view of the application state including the end-user experience. In this chapter, we will discuss what data to collect, and from where and how to collect the data in order to achieve this view Most application systems are implemented in a way that, in addition to the application logic executed at the provider's site (referred to as the back-end), parts of the application are executed at client's site. The client site usually constitutes a system tier accessing the back-end

EXTRACTION OF PERFORMANCE-RELEVANT SYSTEM INFORMATION

Summary statistics (e.g., counts, percentile, etc.) over time, execution traces provide a detailed representation of the application-internal control flow that results from individual system requests.



EXTRACTION OF PERFORMANCE-RELEVANT SYSTEM INFORMATION

Collection of performance measurements from the relevant locations of the application system. This chapter focuses on the representation of higher the application system. While time series represent summary statistics (e.g., counts, percentile, etc.) over time, execution traces provide a detailed representation of the application-internal control flow that results from individual system requests.

From this data, architectural information, including logical and physical deployments and interactions (topology), can be extracted. For all cases, we will highlight examples and use cases in the context of APM level performance-relevant information about the system and their end-users that can be extracted from this data and that is used for APM visualization and reasoning, as detailed in the next chapters. Notably, we will focus on three commonly used representations, namely time series, execution traces, and augmented information about the architecture.

When depicting the number of users accessing a system, time series usually show a periodic pattern, e.g., based on the weekdays and the hours of the day. Other interesting patterns are spikes, for instance, indicating peaks in workload or hiccups.

EXECUTION TRACES

A data structure commonly used in APM for this purpose is an execution trace. Informally, an execution trace is a representation of the execution flow of a request through the system–ideally starting from the end user..

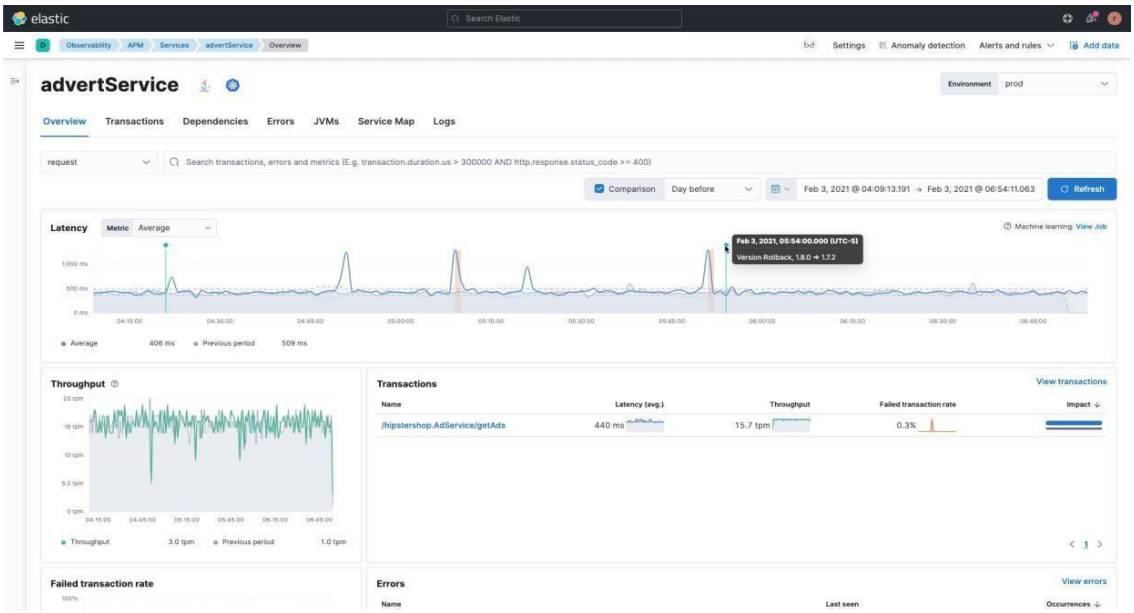
The execution trace starts with an operation called do Filter that is commonly found as an entry point in web-based applications. It can be observed that the execution of the do Filter operation includes a sequence of additional nested operation executions, until the list operation performs a sequence of calls to a database.

In addition to the execution flow, capturing components (e.g., Java classes or microservices) and operations, and locations (e.g., application server, IP address), execution traces usually include further measurements. One type of performance measurement commonly found in execution traces is the response time (or duration) of each operation execution.

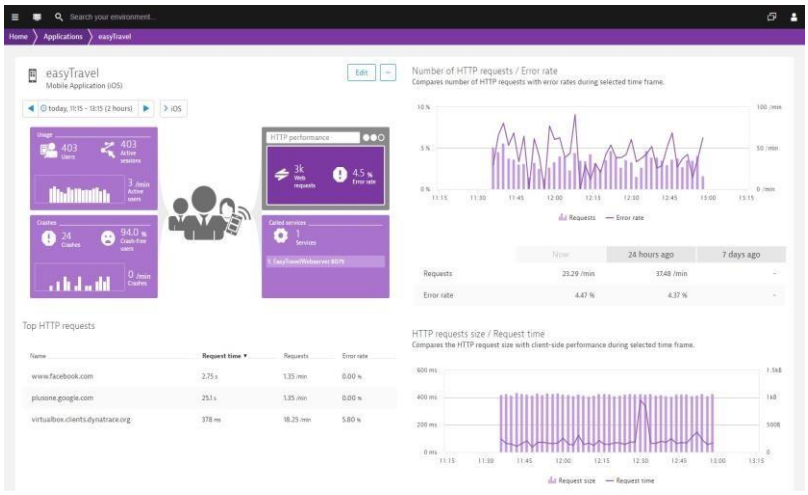
In the example, the response time for each operation execution is included in the second column. Moreover, execution traces may include information such as the parameters of the operation executions.

ARCHITECTURAL INFORMATION

Time series and execution traces allow to analyze the chronological order of performance measurements and of individual requests respectively. This information is commonly used to derive and represent performance-relevant architectural information of a system. The architecture of a system includes structural and dynamic information. Examples for structural information are the existence and deployment of software and hardware components.



The dynamic information includes interactions (e.g., number of calls, average response times) between components and associated information about the runtime behavior, e.g., a health state or time series. In Chapter 4 we include example of performanceaugmented architectural information. This representation is useful to have an overall state of the system and it provides a basis for a detailed manual or automated.



10. ADVANTAGES AND DISADVANTAGES ADVANTAGES:

- The model produced for water quality upkeep is extremely advantageous for defending general wellbeing and furthermore adds to the perfect climate.

- The computerization of this water checking, cleaning and control process eliminates the need of difficult work and consequently sets aside time and cash.

- The computerization of the framework makes the control and checking process more proficient and successful. Continuous checking on cell phone which is conceivable through the connection point of plc with Arduino and Bluetooth module permits remote controlling of the framework.

DISADVANTAGES:

- It is hard to gather the water tests from all the region of the water body. The expense of examination is extremely high.

- The lab testing and examination takes a few time and thus the lab results doesn't reflect constant water quality estimation because of postpone in estimation.

- The cycle is tedious because of slow course of manual information assortment from various areas of the water body.The technique is inclined to human blunders of different structures.

11.CONCLUSION :

Thus our project is used to Monitoring of Turbidity, PH &Temperature of Water makes use of water detection sensor with unique advantage and existing GSM network. The system can monitor water quality automatically, and it is low in cost and does not require people on duty. So the water quality testing is likely to be more economical, convenient and fast. The system has good flexibility. Only by replacing the corresponding sensors and changing the relevant software programs, this system can be used to monitor other water quality parameters. The operation is simple. The system can be expanded to monitor hydrologic, air pollution, industrial and agricultural production and soon. It has widespread application and extension value. By keeping the embedded devices in the environment for monitoring enables self-protection (i.e., smart environment) to the environment. To implement this need to deploy the sensor devices in the environment for collecting the data and analysis. By deploying sensor devices in the environment, we can bring the environment into real life i.e. it can interact with other objects through the network. Then the collected data and analysis results will be available to the end user through the Wi-Fi.

12. FUTURE SCOPE:

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 water. People who are living in rural areas near to the river will be very satisfied with our idea. It will be useful to monitor water pollution in specific area. So this system prevent people from water pollution. It will be used for farming purpose to check quality water, temperature and PH level. Our Impact of this project is also create a social satisfaction for farmers too. The scalabilty of this project gives the addition of more different type of sensors. By interfacing the relay we can control the supply of water. We can also implement as a revenue model. 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.

13. APPENDIX:

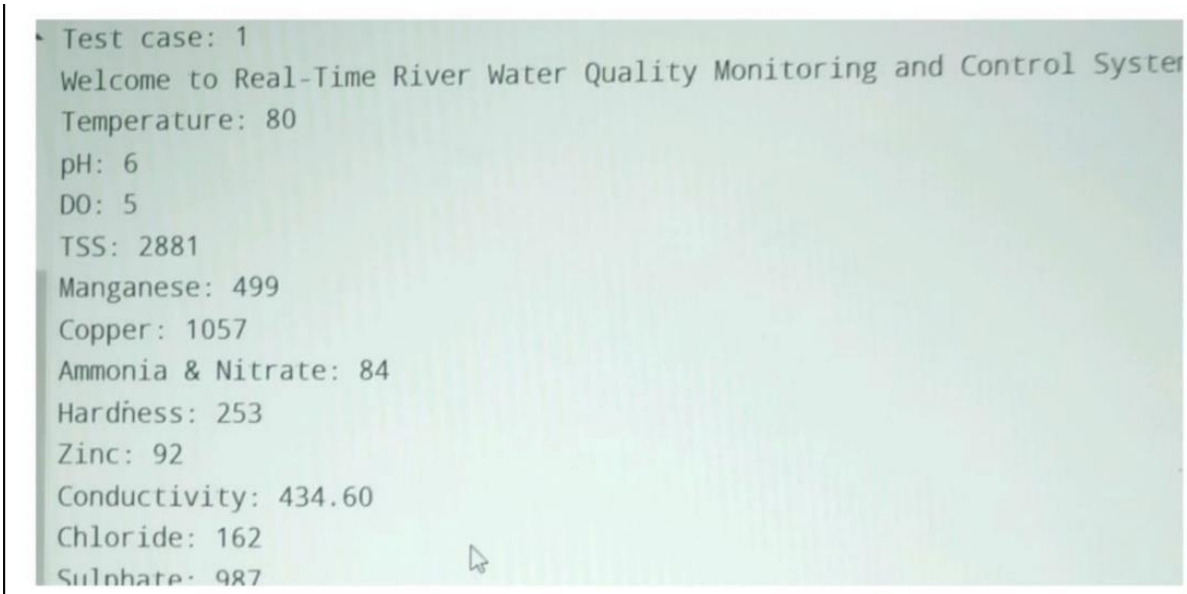
13.1. SOURCE CODE :

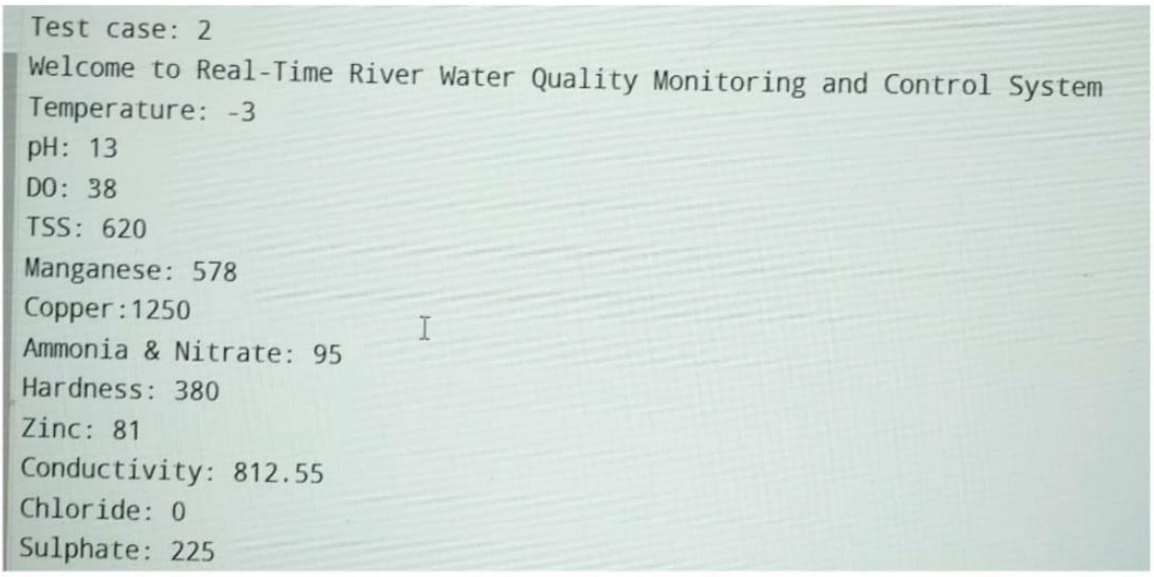
Python code for random value generation:

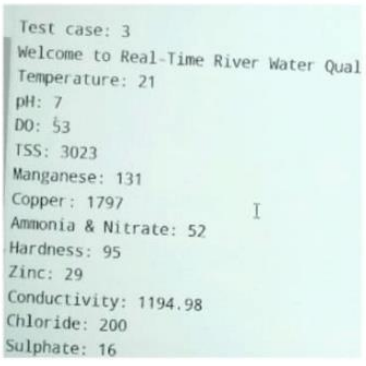
```
#importing Random function to generate the value
import random as rand for i in range(5):
    print("Test case:",i+1)
    print("Welcome to Real-Time River Water Quality Monitoring and Control System")
temperature = int(rand.randint(-40,125))     pH = int(rand.randint(0,14))
    DO = int(rand.randint(0,100))
    TSS = int(rand.randint(0,3700))
    Manganese = int(rand.randint(0,1000))
Copper = int(rand.randint(0,2000))
ammonia_Nitrate = int(rand.randint(0,100))
    Hardness = int(rand.randint(0,1000))
    Zinc =  int(rand.randint(0,100))
    Conductivity =  f"{float(rand.uniform(0.001,2000)):.2f}"
    Chloride = int(rand.randint(0,200))
    Sulphate = int(rand.randint(0,1000))
    #These variables store value of ramdom data to be shared to the cloud
    #printing the values
print(
        "Temperature:", temperature,
```

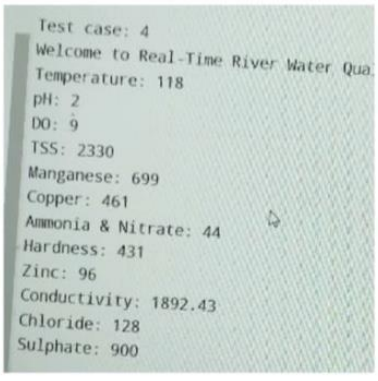
```
    "\npH:", pH,
    "\nDO:", DO,
    "\nTSS:", TSS,
    "\nManganese:", Manganese,
    "\nCopper:", Copper,
    "\nAmmonia & Nitrate:",ammonia_Nitrate,
    "\nHardness:",Hardness,
    "\nZinc:", Zinc,
    "\nConductivity:", Conductivity,
    "\nChloride:", Chloride,
    "\nSulphate:", Sulphate, "\n"
)
```

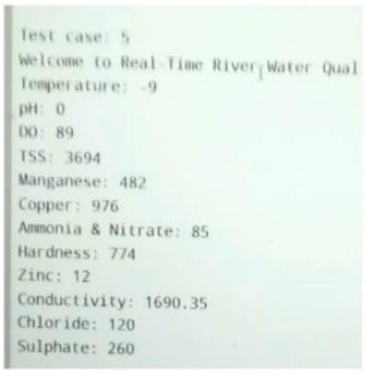
Output:











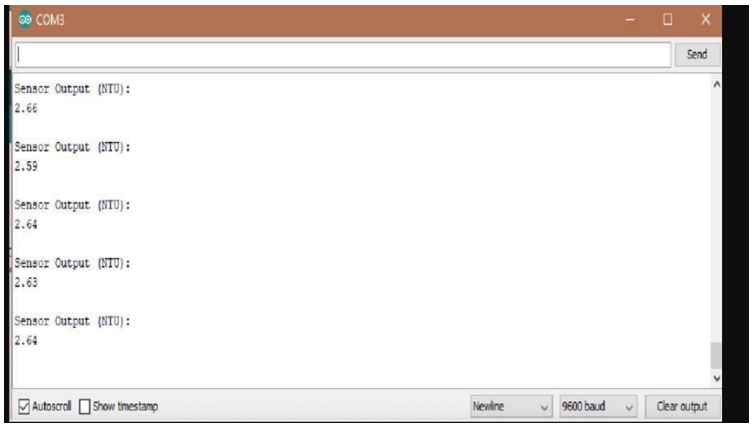
Code for arduino implementation:

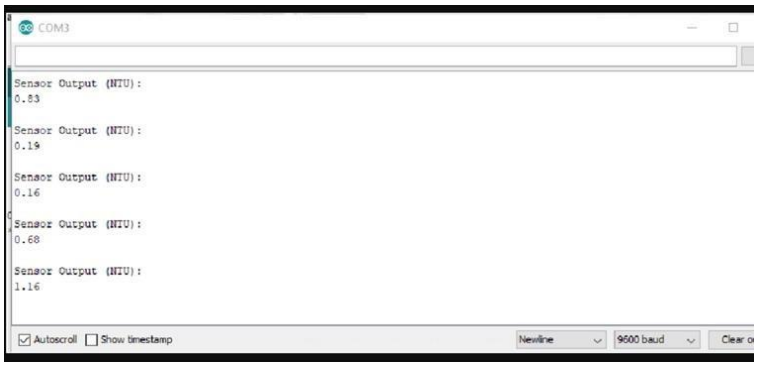
```
#include <OneWire.h>

#include <DallasTemperature.h>
```

```
#define ONE_WIRE_BUS 5
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
float Celcius=0; float Fahrenheit=0;
float voltage=0; const int analogInPin =
A0; int sensorValue = 0; unsigned long
int avgValue; float b; int buf[10],temp;
void setup(void) { Serial.begin(9600);
sensors.begin();  int sensorValue =
analogRead(A1);  voltage =
sensorValue * (5.0 / 1024.0);
} void loop(void) {
sensors.requestTemperatures();
 Celcius=sensors.getTempCByIndex(0);
Fahrenheit=sensors.toFahrenheit(Celcius);  for(int
i=0;i<10;i++)
{
 buf[i]=analogRead(analogInPin);
delay(10); } for(int i=0;i<9;i++)
{  for(int j=i+1;j<10;j++)
 {
if(buf[i]>buf[j])
 {
temp=buf[i];
buf[i]=buf[j];
buf[j]=temp;
 }
  } } for(int i=2;i<8;i++)
avgValue+=buf[i]; float
```

```
pHVol=(float)avgValue*5.0/1024/6; float
phValue = -5.70 * pHVol + 21.34;
Serial.println(phValue);
Serial.print("pH");
 Serial.print(" C ");
 Serial.print(Celcius);
 Serial.print(voltage);
Serial.print("V");  delay(10000);
}
```

Output test cases:





Code for implementation:

```
import serial import time import
csv import numpy as np import
matplotlib.pyplot as plt ser =
serial.Serial('/COM6',9600)
ser_bytes = ser.readline(10) print
(ser_bytes) ser.flushInput()
while True:
 try:
 ser_bytes = ser.readline()
 decoded_bytes = float(ser_bytes[0:len(ser_bytes)-2].decode("utf-8"))
print(decoded_bytes) temp = float(decoded_bytes(1:3)) turb =
float(decoded_bytes(4:6)) pH = float(decoded_bytes(6:8))  with
open("test_data.csv","a") as f:  writer = csv.writer(f,delimiter=",")
writer.writerow([time.time(),decoded_bytes])  except:
print("Keyboard Interrupt")  ser.close()  break()
t = np.arange(0.0, 2.0, 0.01) s = 1 +
np.sin(2*np.pi*t) plt.plot(t, s)
plt.xlabel('time (s)') plt.ylabel('Celsisus
(C)') plt.title('Temperature')
plt.grid(True)
plt.savefig("Temperature.png")
plt.show() Serial.begin(9600);
sensors.begin();  int sensorValue =
analogRead(A1);  voltage =
sensorValue * (5.0 / 1024.0);
}
void loop(void) {
 sensors.requestTemperatures();
```

```
 Celcius=sensors.getTempCByIndex(0);
Fahrenheit=sensors.toFahrenheit(Celcius);  for(int
i=0;i<10;i++)
{
 buf[i]=analogRead(analogInPin);
delay(10); } for(int i=0;i<9;i++)
{  for(int
j=i+1;j<10;j++)
 {
if(buf[i]>buf[j])
 {
temp=buf[i];
buf[i]=buf[j];
buf[j]=temp;
 }
n = 256
X = np.linspace(-np.pi, np.pi, 256, endpoint=True)
C,S = np.cos(X), np.sin(X) plt.plot(X, C)
plt.plot(X,S) plt.show()
print ("Visualization of real time sensor Data.") print("/n")

while True:
try:
 ser_bytes = ser.readline()
 decoded_bytes = float(ser_bytes[0:len(ser_bytes)-2].decode("utf-8"))
print(decoded_bytes) temp = float(decoded_bytes(1:3)) turb =
float(decoded_bytes(4:6)) pH = float(decoded_bytes(6:8))  with
open("test_data.csv","a") as f:  writer = csv.writer(f,delimiter=",")
writer.writerow([time.time(),decoded_bytes])  except:
print("Keyboard Interrupt")  ser.close()  break()
```

t = np.arange(0.0, 2.0, 0.01) s

= 1 + np.sin(2*np.pi*t)

plt.plot(t, s)

GITHUB REPOSITORY:

https://github.com/IBM-EPBL/IBM-Project-8612-1658925919

DEMO LINK:

https://drive.google.com/file/d/1GMME_Dhwi_gD9sl8SsDnn64a5Uc6It15/view?usp=share_link