REAL-TIME RIVER WATER QUALITY MONITORING AND CONTROL SYSTEM

PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY AND ENTREPRENEURSHIP

PROJECT REPORT
TEAM ID: PNT2022TMID13315

Team Leader : Muthukumar P -951919CS058

Team member : Ariram R - 951919CS012

Team member : Ganesan P - 951919CS023

Team member : Kannan R -951919CS046

BACHELOR OF ENGINEERING IN
COMPUTER SCIENCE AND ENGINEERING
P.S.R. ENGINEERING COLLEGE -SIVAKASI
November 2022

INTRODUCTION

Project Overview:

Current water quality monitoring system is a manual system with a monotonous process and is very time-consuming. This paper proposes a sensor-based water quality monitoring system. The system consists of several sensors which is used to measure physical and chemical parameters of the water. The main components of Wireless Sensor Network (WSN) include a microcontroller for processing the system, communication system for inter and intra node communication and several sensors. Real-time data access can be done by using remote monitoring and Internet of Things (IoT) technology. Data collected at the apart site can be displayed in a visual format on a server PC with the help of Spark streaming analysis through Spark MLlib, Deep learning neural network models, Belief Rule Based (BRB) system and is also compared with standard values. If the acquired value is above the threshold value automated warning SMS alert will be sent to the agent. The uniqueness of our proposed paper is to obtain the water monitoring system with high frequency, high mobility, and low powered. Therefore, our proposed system will immensely help Bangladeshi populations to become conscious against contaminated water as well as to stop polluting the water.

Purpose :

Water pollution is a foremost global problem which needs ongoing evaluation and adaptation of water resource directorial principle at the levels of ot nwod lanoitanretni neeb sah tI .sllew laudividnistudied that water pollution is the leading cause of mortalities and diseases worldwide. The records show that more than 14,000 people die daily worldwide due to water pollution. In many developing countries, dirty or contaminated water is being used for drinking without any proper prior treatment. One of the reasons for this happening is the ignorance of public and administration and the lack of water quality monitoring system which makes serious hea htlissues

T fo ytilauq eht rotinom owater with the support of information sensed by the sensors dipped in water. Using different sensors, this system can collec sretemarap suoirav t .no os dna ,erutarepmet ,ytivitcudnoc ,ytidibrut ,negyxo devlossid ,Hp sa hcus ,retaw morf

- To make use of IBM Cloud.
- To utilize IBM Watson platform for user interface
- To create Web page using python and others for user friendly application
- To provide user with Real time information and creating customerized application according to need
- Making people get access to quality water with pocket friendly application for all kinds of u

LITERATURE SURVEY

REFERENCES

With literature survey we have done the pre study required for knowing about the project and applications through already published research papers.

1) loT-based System for Real-time Water Pollution Monitoring of River

Mohammad Ariful Islam Khan; Mohammad Akidul Hoque; Sabbir Ahmed

IEEE September 2021

The research proposes a system to remotely monitor the water quality of a river so that the authorities can gather better insights about the condition of that particular river and predict the critical future phenomena. Consequently, they will be able to take auspicious steps in order to protect the rivers and save the environment. The proposed framework can observe the real-time value of pH, conductivity, turbidity, temperature and flow of the water by utilizing various sensors. Furthermore, through our device, effective predictions about imminent floods can be made. Thus, authorities can commence early warning for floods and ensure prompt evacuation. Thus, our technique can significantly minimize the casualties caused by this disaster. In this context, real-time feeds are obtained through Internet of Things (IOT). For wireless data transmission Message Queuing Telemetry Transport (MQTT) is used.

2) A Development and Implementation of Water Quality Assessment Monitoring (WQAM) System using the Internet of Things (IOT) in Water Environment

Muhammad Farhan Johan, S. Abdullah, A. Zanal Saurabh S. Soman, Hamidreza Zareipour, 0m Malik

JEVA - 23 November 2021

This paper presents the development and implementation of Water Quality Assessment and Monitoring (WQAM) system. The system development used Wi-Fi enabled microcontroller to connect with the IOT environment and store the data in the IOT cloud server. The microcontroller used is Arduino UNO that interacts with three types of sensor probes which are pH, turbidity and temperature probe. All the data measurements is transferred using a Wi-Fi module which is ESP8266. The IOT cloud used to utilize the data frame is Thing Speak. This system was implemented on Bandar Pereda Lake and Deraa River in Pulao Pinang with two systems implemented at each location. The sensors were placed on the water surface for more accurate measurements. This system continuously measures the readings of pH, turbidity dan temperature on the lake/river for every 1 hour. Twenty readings were taken for every 1 hour within the first 20 minutes with 1 minute interval and the readings were stored in the IOT cloud server.

3)Review of Water Quality Monitoring using Internet of Things (IOT) Mr. A. P. Roger Rozario, R. Surya

IEEE, 2019

The quality of the water must be monitored in real-time to ensure its safety and supply. Monitoring water in traditional ways takes longer, which can take up to from 24 to 96 hours to identify contaminants in water supplies, which are more time taking. This project aims at developing a water quality monitoring system using sensors and IOT (Internet of Things). The water quality parameters like temperature, pH, and turbidity are measures using sensors and the water quality index is determined. The measured values from the sensors will be processed using a microcontroller, and alert message will be sent to the user via an android application developed using MIT app inventor in case of any abnormalities.4) An IOT Based Smart Water Quality Monitoring System using Cloud

Ajith Jerom B.; R. Manimegalai; R. Manimegalai

IEEE - April 2020

Other sources of pollution include agricultural runoff and unregulated small scale industry that results in polluting, most of the rivers, lakes and surface water in India. In this paper, An IOT Based Smart Water Quality Monitoring System using Cloud and Deep Learning is proposed to monitor the quality of the water in water-bodies. In conventional systems, the monitoring process involves the manual collection of sample water from various regions, followed by laboratory testing and analysis. This process is ineffective, as this process is arduous and time-consuming and it does not provide real-time results. The quality of water should be monitored continuously, to ensure the safe supply of water from any water bodies and water resources. Hence, the design and development of a low-cost system for real-time monitoring of water quality using the Internet of Things (IOT) is essential. Monitoring water quality in water bodies using Internet of Things (IOT) helps in combating environmental issues and improving the health and living standards of all living things.

5) Design and Implementation of Real Time Approach for The Monitoring of Water Quality Parameters

Siti Aishah Binti Makhtar; Norhafizah Binti Burham; Anees Bt Abdul Aziz

IEEE - June 2022

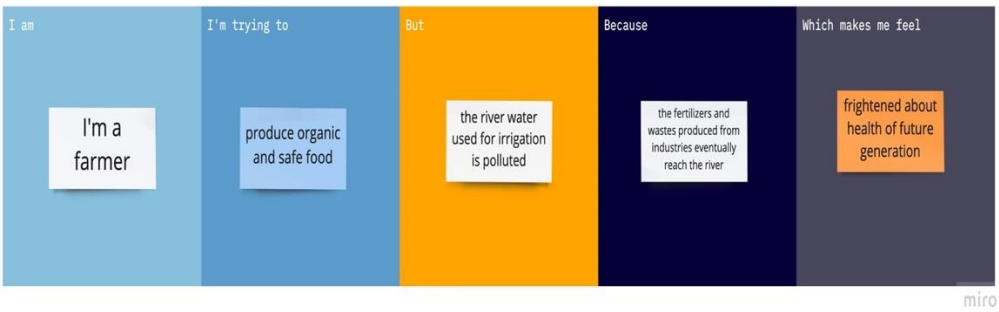
Access to safe drinking water is essential to nurturing human life on earth. Polluted air and unsanitary water can cause health problems. Unhygienic water can cause stomach and health-related problems. A specific range of water quality parameters, mainly temperature, pH, total dissolved solids (TDS) and turbidity, can degrade the growth of this bacteria. This presented paperwork is to develop a smart water quality monitoring system using four sensors and an IOT platform to help determine water quality. It is to analyse the parameters of water samples such as tap water, co way water, river water, pond water, and lake water whether these water samples are in the threshold range for drinking or not. The device is initially used to measure pH, turbidity, total dissolved solids (TDS) and temperature, and then sent the information to the microcontroller Arduino Uno.

Problem Statement Definition :

Still in India most of the people use simple water purifier that is not enough to get surety of pure water. Sometimes the water has dangerous particles or chemical mixed and general purpose water purifier cannot purify that. And its impossible to check the quality of water manually in every time. This may cause many problems to aquatic life and many humans whose life is related to river Water system.there a huge need for a such system which can give real time quality monitoring.

So an automatic real-time monitoring system is required to monitor the health of the water reserved in our rivers. So it can warn us automatically if there is any problem with the reserved water. And we can check the quality of the water anytime and from anywhere. By keeping this mind we designed this system needed for real time applications

Problem Statement (PS)	**I am (Customer)**	**I'm trying to**	**But**	**Because**	**Which makes me feel**
PS-1	I'm a villager	Have safe drinking water for my family	Water in our neighbourhood causing many skin allergies and health issues	We only access to simple water purifiers for water purification	Anxious about our future
PS-2	I'm a village councillor	Provide people with safe drinking water who are suffering from health issues caused by intake of impure river water	I can't achieve it due to purification system which we are using now isn't enough for complete purification	Increasing urbanisation and population is always requires more technology and awareness in people	Curious to learn about new technologies and applications where I can use it

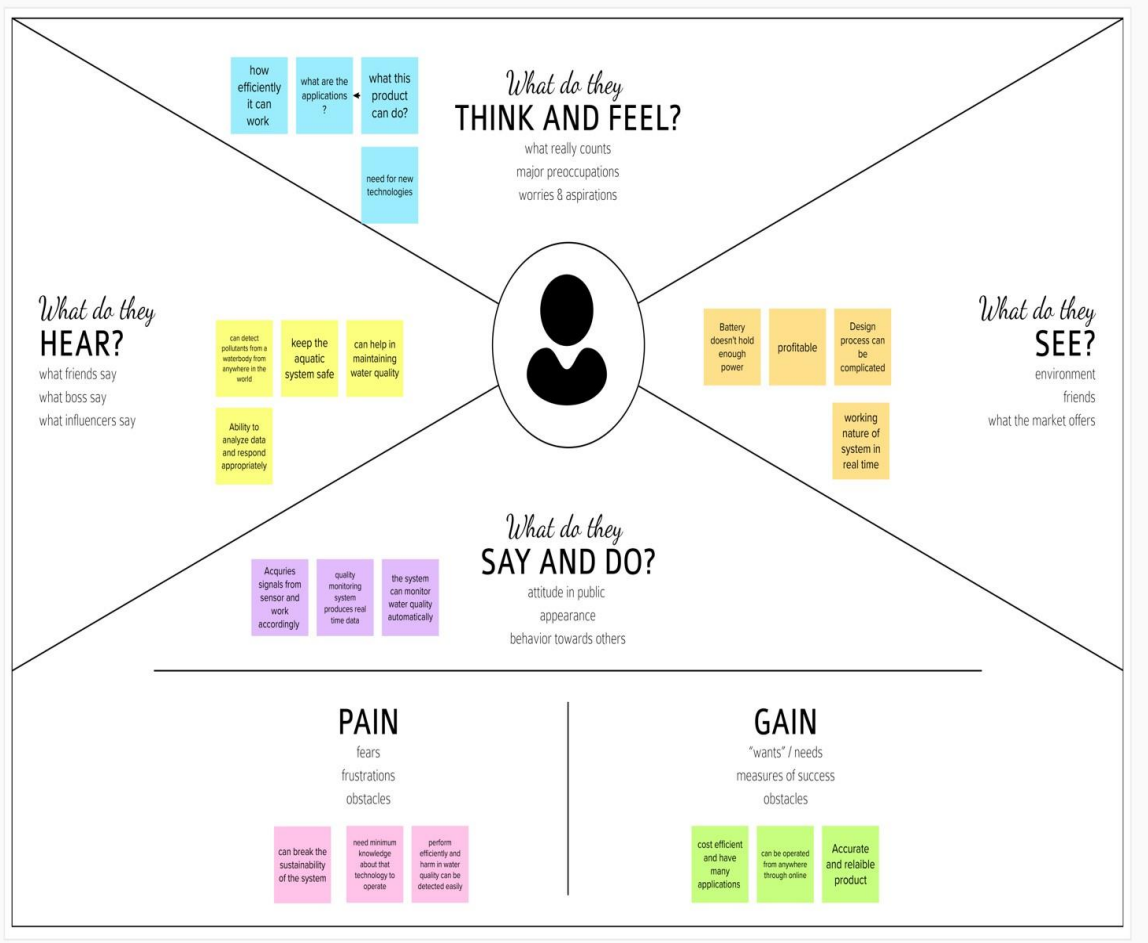


IDEATION & PROPOSED SOLUTION

Empathy Map Canvas:

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes.

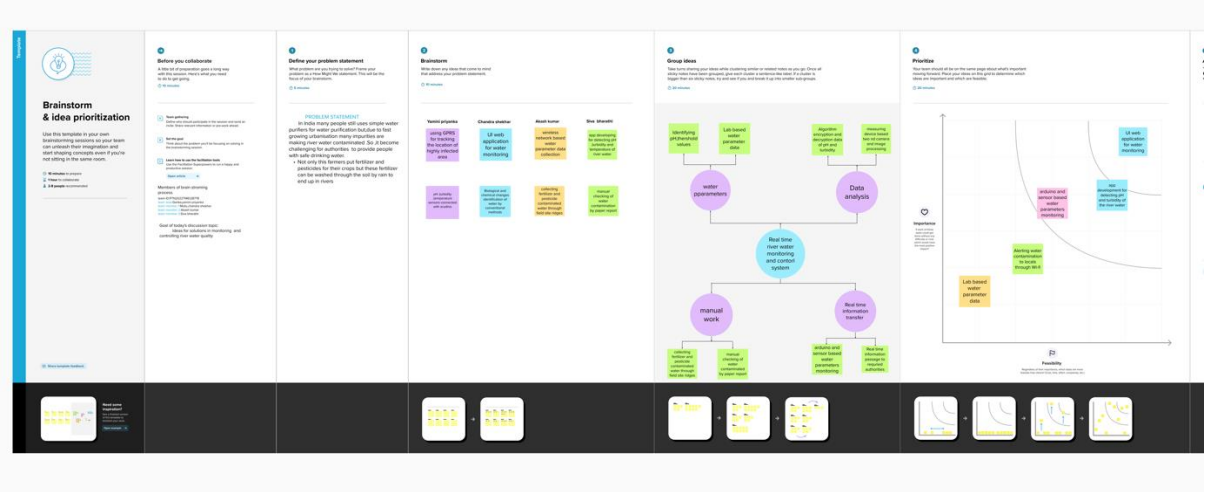
It is a useful tool to helps teams better understand their users.Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.Here project objectives and solutions are discussed and noted in empathy map canvas



Ideation & Brainstorming :

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

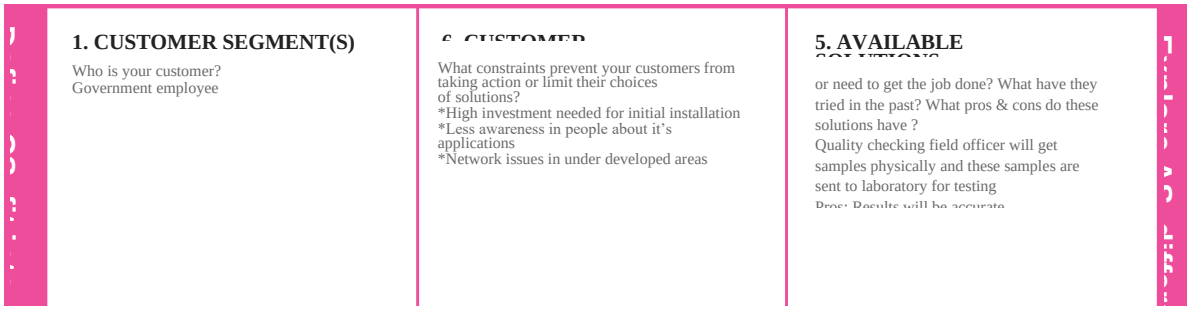


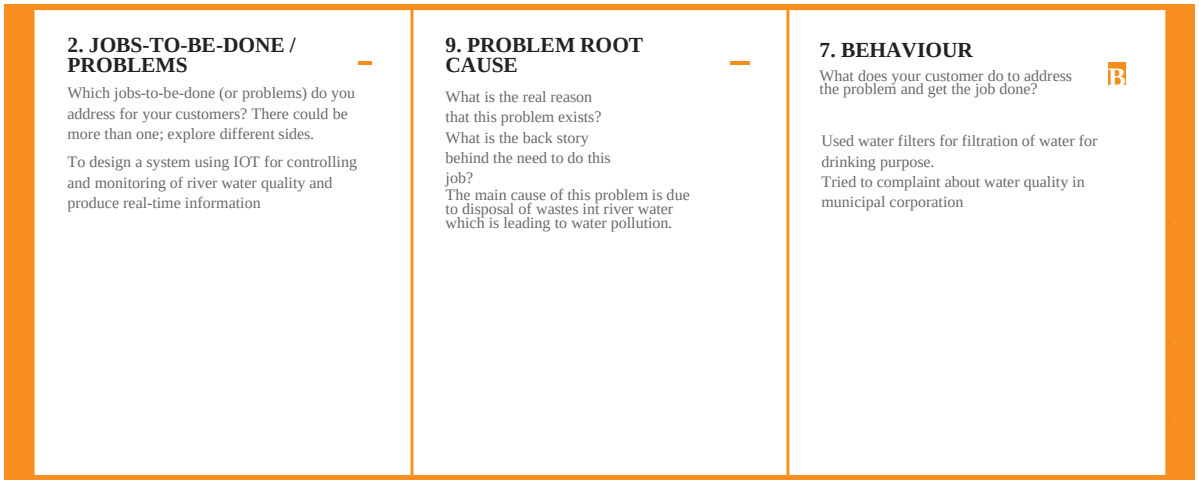
Proposed Solution:

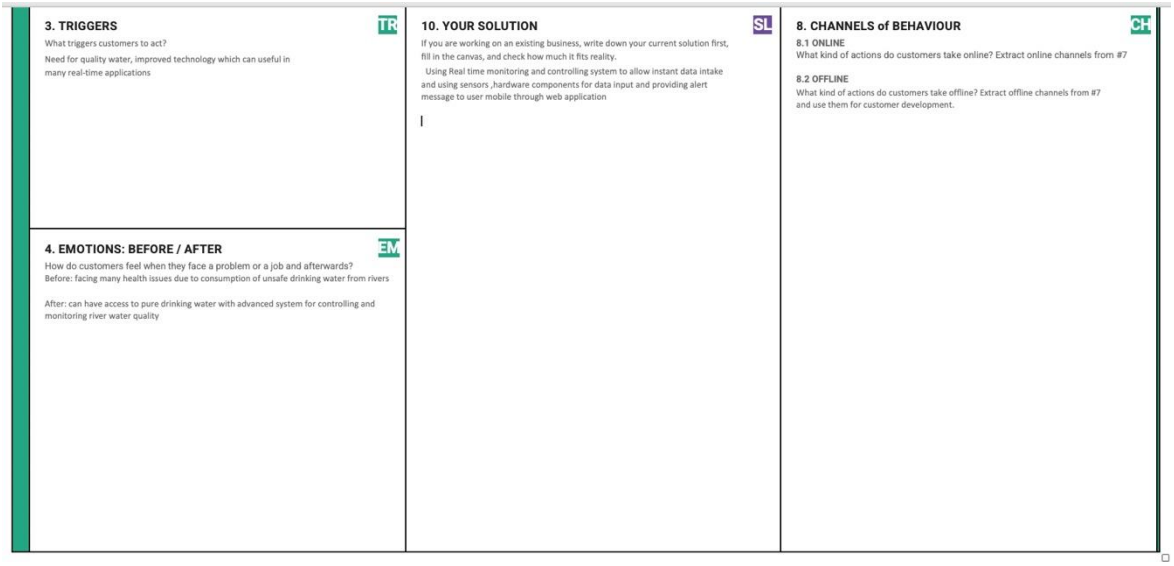
A proposed solution document has been created including all the possible problems which have the scope to arise and providing clear description all those parameters

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To get information about real time water quality by using parameters such as pH, turbidity and temperature
2.	Idea / Solution description	Monitoring water parameters by using Arduino and sensors and giving real time information through mobile application
3.	Novelty / Uniqueness	Low investment and maintainace cost , IOT have many applications
4.	Social Impact / Customer Satisfaction	People will have awareness about new technologies and can have access to quality water
5.	Business Model (Revenue Model)	IOT based Water monitoring and control model
6.	Scalability of the Solution	The process of operating is easy and it can designed according to customer needs.

Problem Solution fit:







REQUIREMENT ANALYSIS

Functional requirement:

Functional requirements which are needed in this project are listed down and the description for all the needed software is given .

Following are the functional requirements of the proposed solution

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through mobile number
FR-2	User Confirmation	Confirmation via OTP Conformation via text message
FR-3	Temperature increase detection	Increase in temperature of river water can be detected by sensors and can send an alert message to user mobile
FR-4	Sample data	Real-time data sample information will be collected through software
FR-5	Web application	A mobile application will be developed and water quality details, analysis report can be viewed by user through smart phone

Non-Functional requirements

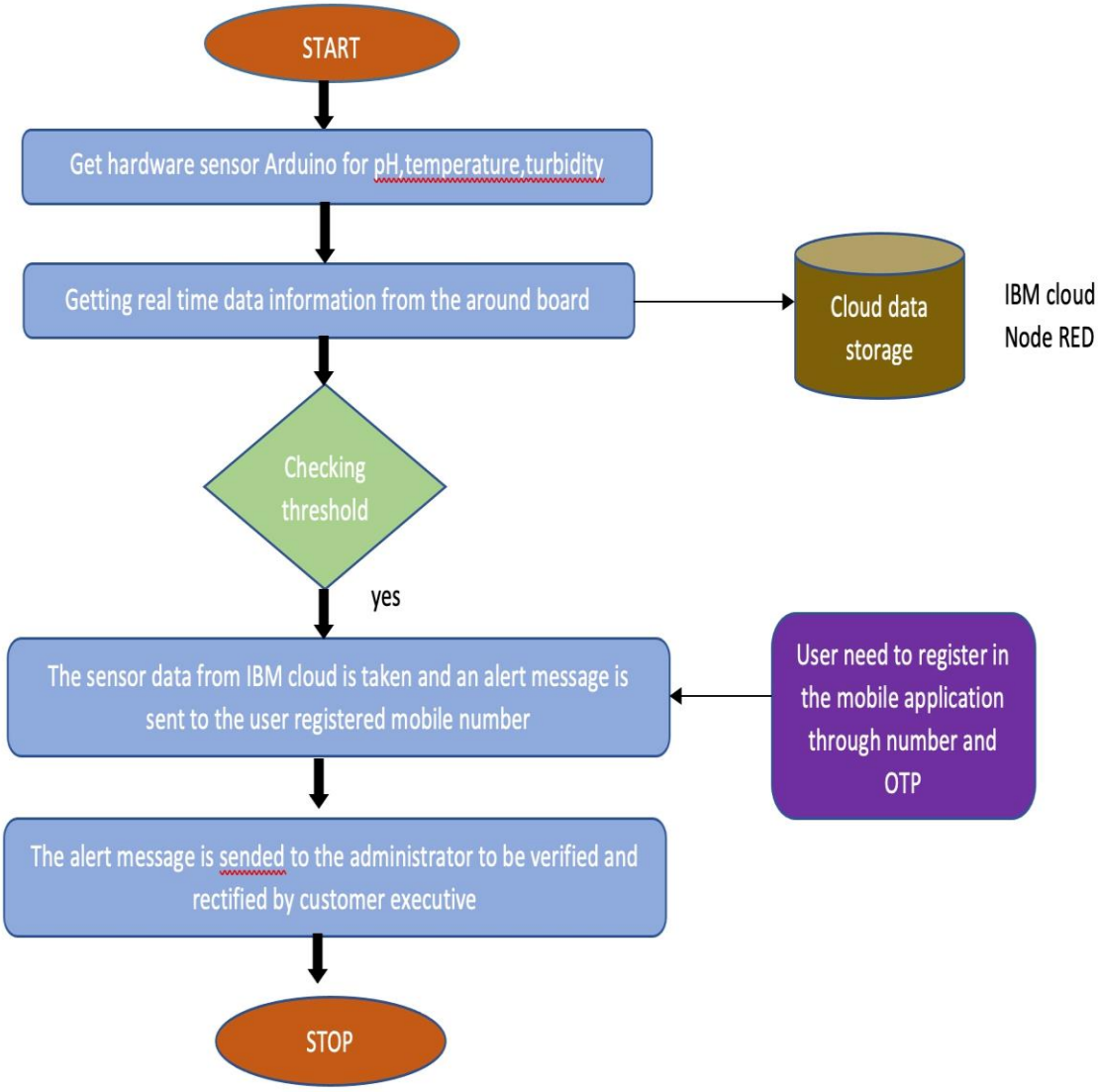
Non-functional requirements deals with all non-functional and hardware components which are involved in this project and their description is listed
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Can be viewed through user interface screens which will be not more than 5 seconds
NFR-2	**Security**	User account can be password protected with two step authentication which can be verified through OTP
NFR-3	**Reliability**	the system can produce real time analysed data through mobile application
NFR-4	**Performance**	Log in information can be verified within 10 seconds and user can login ,can get accurate data almost all the times
NFR-5	**Availability**	Maximum down time will be about 4 hours
NFR-6	**Scalability**	System can handle user traffic without crashing upto 1000 users

PROJECT DESIGN

Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

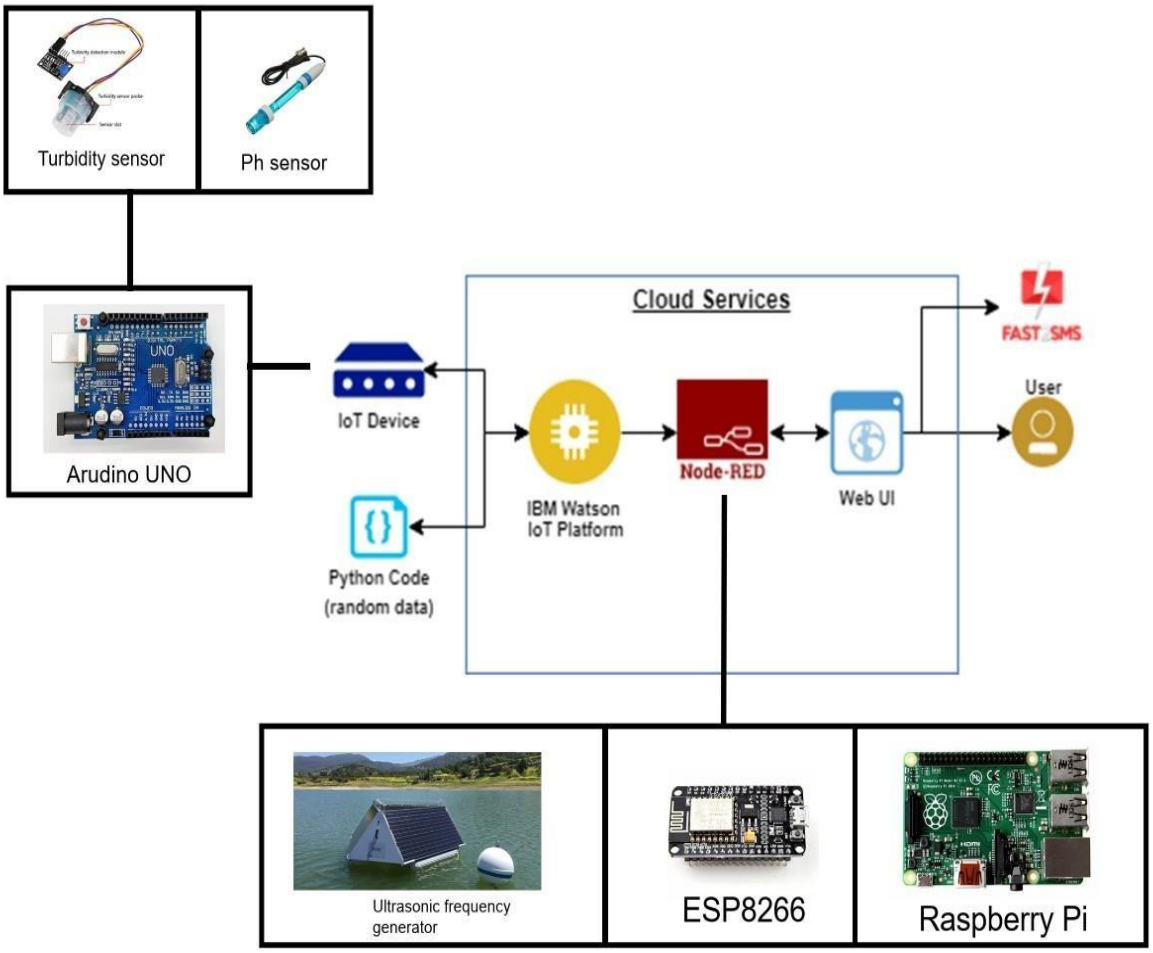


Solution & Technical Architecture:

Solution Architecture:

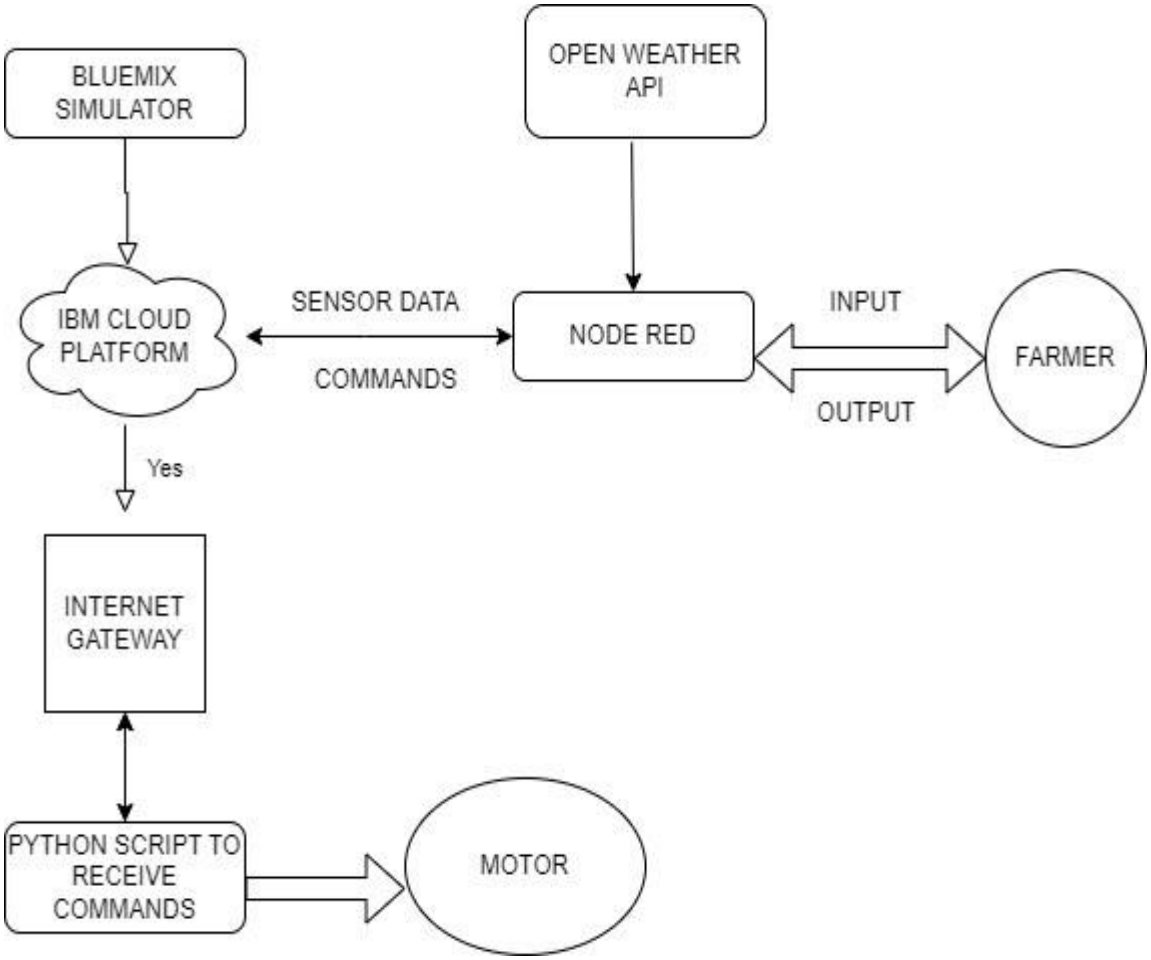
Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

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



TECHNICAL ARCHITECTURE:

Technical Architecture include all the Components & Technologies required and Application Characteristics



Application Characteristics

S.No	Characteristics	Description	Technology
1.	PH level monitoring	The PH level of river water can be monitored via placing sensors in rivers	PH-sensor
2.	Temperature monitoring	The temperature of river water can be monitored	Temperature sensor
3.	Pollution monitoring	The clarity and purity of river water can be monitored	Conductive sensor
4.	Soil level monitoring	The amount of soil mixed in river water can be measured	Turbidity sensor

Components & Technologies

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application	HTML, CSS, Bootstrapping
2.	Application Logic-1	Logic for a process in the application	JAVA/PYTHON
3.	Application Logic-2	Logic for a process in the application	IBM WATSON STT services
4.	Application Logic-3	Logic for a process in the application	BM WATSON Assistant
5.	Database	Data Type, Configurations etc	MySQL,
6.	Cloud Database	Database Service on Cloud	IBM cloud
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	Purpose of External API used in the application	IBM Weather API, etc
9.	External API-2	Purpose of External API used in the application	Aadhar API, etc
10.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration	Local, Cloud Foundry, Kubernetes, etc.

User stories:

List of all the user stories for the product are given below

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my registered mobile number	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive OTP once I have registered for the application	I can receive confirmation OTP & click confirm link	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering mobile number and OTP received	I can access easily	High	Sprint-1
		USN-6	As a user my mobile application should work without errors about 90% of time	I can get reliable information	Medium	Sprint -1
Customer (Web user)	Dashboard	USN-7	As a user I need information simplified and easily understandable	I can understand info better	High	Sprint -1
		USN-8	As a user I like to edibility quality of water	I can use know about water quality	Medium	Sprint -1
Customer Care Executive		USN-9	As a user I prefer to have executives who can help out in problem shooting	I can solutions easily	High	Sprint -2
Administrator	Risk tollerance	USN-5	As an administrator who can handle system maintainance	Should monitor the records properly	High	Sprint -1

PROJECT PLANNING & SCHEDULING

Sprint Planning & Estimation:

Sprint	Functional Requirement(Epic)	User Story Number	User Story/Task	Story Points	Priority	Team Members
Sprint 1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Yamini priyanka
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Siva bharathi
Sprint-2		USN-3	As a user, I can register for the application through Facebook	2	Low	Akash kumar
Sprint-1		USN-4	As a user, I can register for the application through Gmail	2	Medium	Chandhra shekhar

Sprint-1	Login	USN-5	As a user, I can log into the application by Entering email & password	1	High	Yamini priyanka

Project Tracker, Velocity & Burndown Charts (4 Marks):

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	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	30	30 Oct 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	49	06 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	50	07 Nov 2022

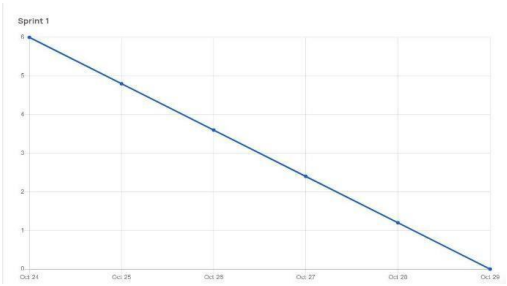
Estimated Effort:

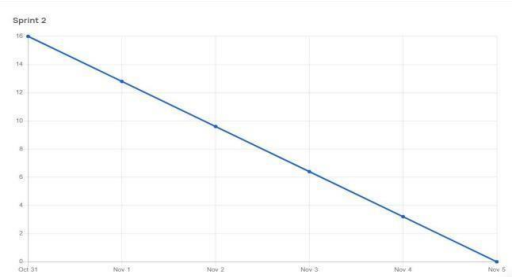
Velocity:

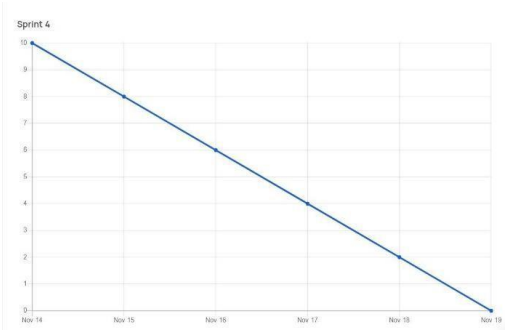
$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

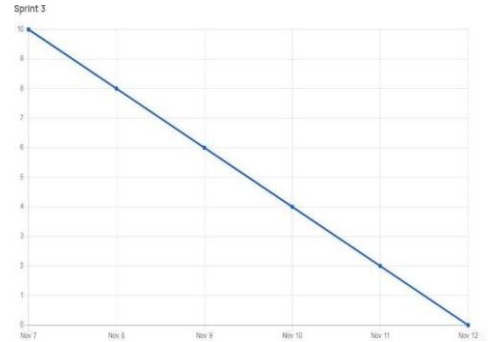
Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies suchas Scrum. However, burn down charts canbe applied to any project containing measurable progress over time.

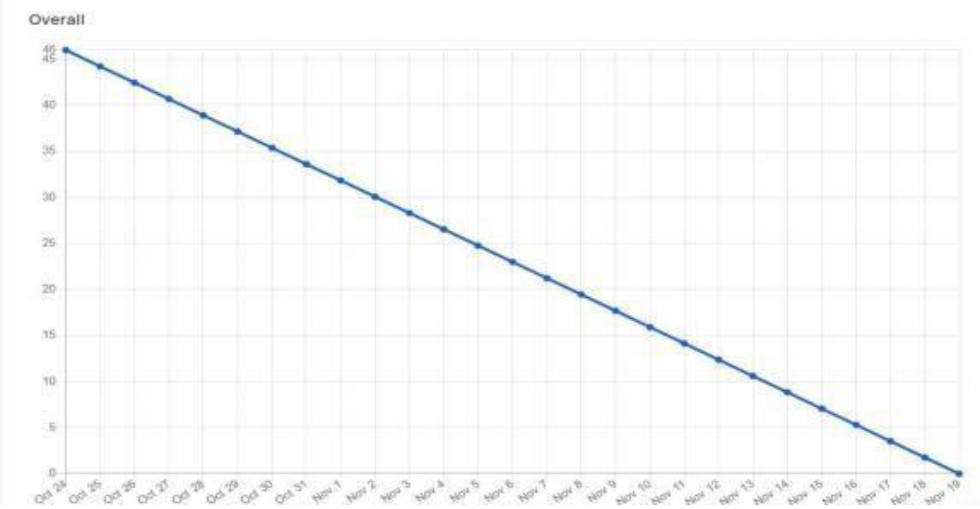








Overall burndown chart:

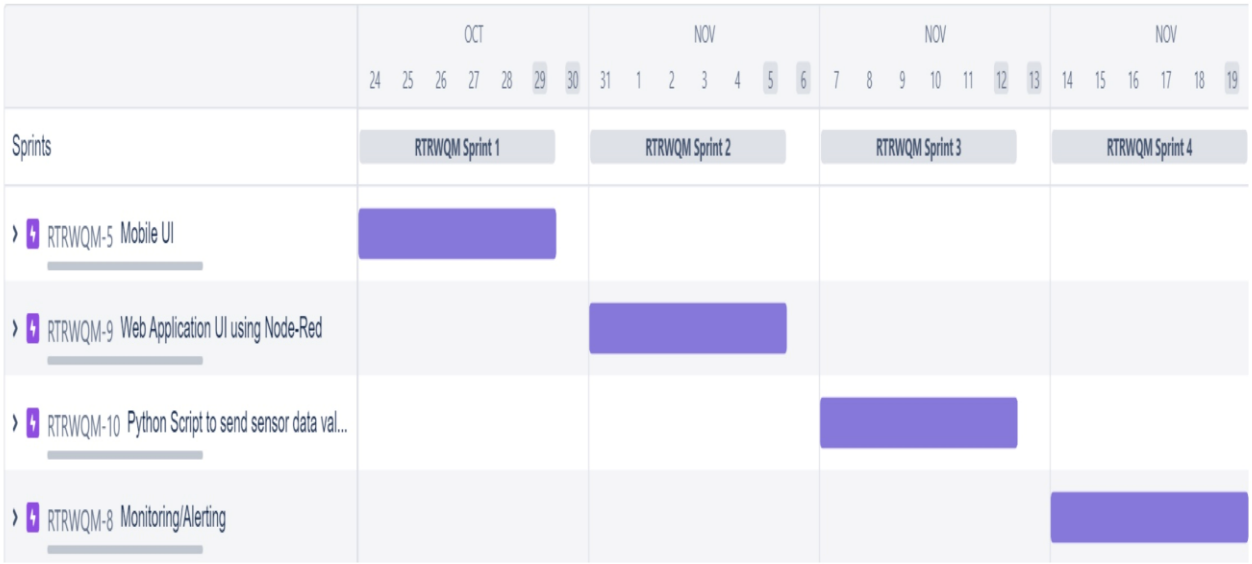


Sprint Delivery Schedule:

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature survey on the selected project is done by gathering information about related details on technical papers and web browsing	28 SEPTEMBER 2022
Empathy Map	Prepare Empathy Map Canvas to combine thoughts and pains, gains of the project with all team members	24 SEPTEMBER 2022
Ideation	Brainstorming session is conducted with all team members to list out all the ideas and prioritise the top 3 ideas	25 SEPTEMBER 2022
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	23 SEPTEMBER 2022
Problem Solution Fit	Prepare problem - solution fit document.	30 SEPTEMBER 2022
Solution Architecture	Prepare solution architecture document with basic design ideas	28 SEPTEMBER 2022
Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application	20 OCTOBER 2022

Functional Requirement	functional requirement is prepared with listing both functional and non -functional needs	8 OCTOBER 2022
Data Flow Diagrams	Draw the data flow diagrams and submit user stories	9 OCTOBER 2022
Technology Architecture	Prepare architecture with components & technologies, application characteristics	10 OCTOBER 2022
Prepare Milestone & Activity List	Prepare the milestones & activity list of the project.	22 OCTOBER 2022
Project Development - Delivery of Sprint-1, 2, 3 & 4	Develop & submit the developed code by testing it.	IN PROGRESS..

Reports from JIRA



CODING & SOLUTIONING

Feature 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"/>

Address

<textarea cols="80" rows="5" value="address">
</textarea>

Email:
<input type="email" id="email" name="email"/>

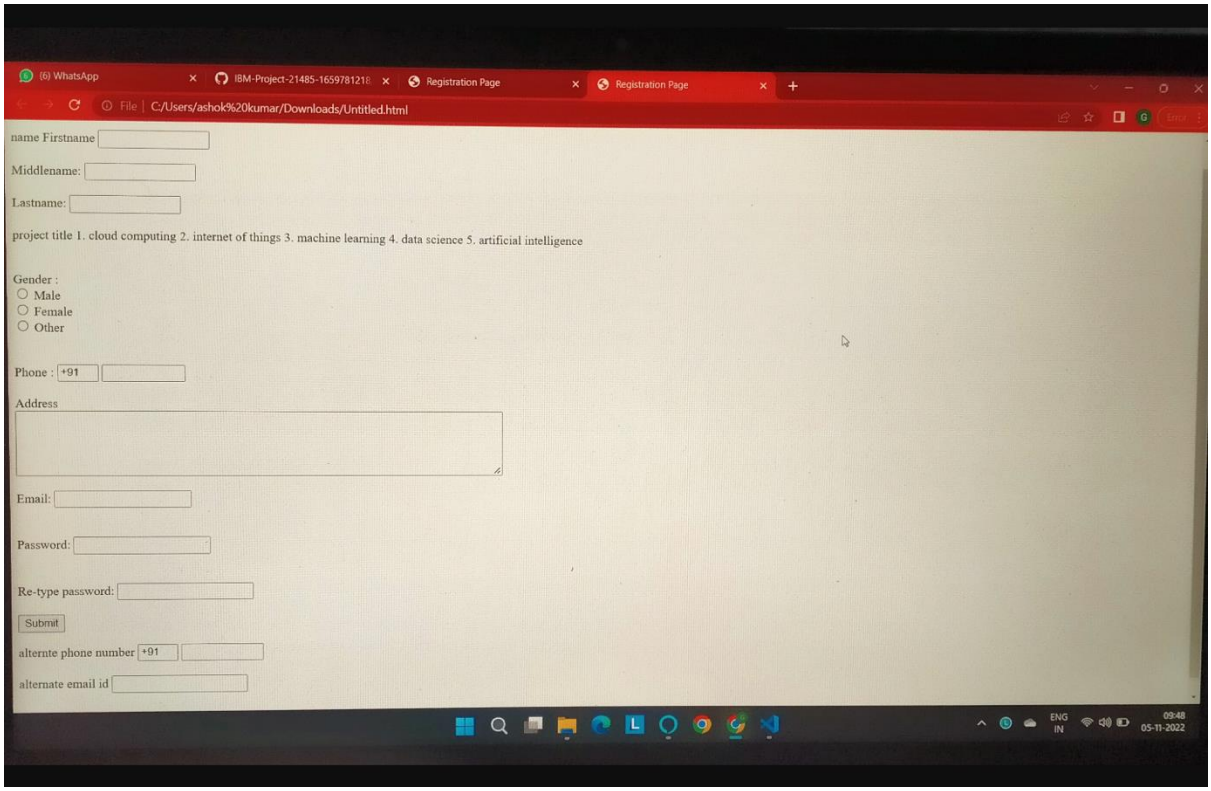
Password:
<input type="Password" id="pass" name="pass">

Re-type password:
<input type="Password" id="repass" name="repass">

<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"/>

alternate email id
<input type="altrernate email id" name="alternate email"/>

<body>
<html>



Feature code 2:

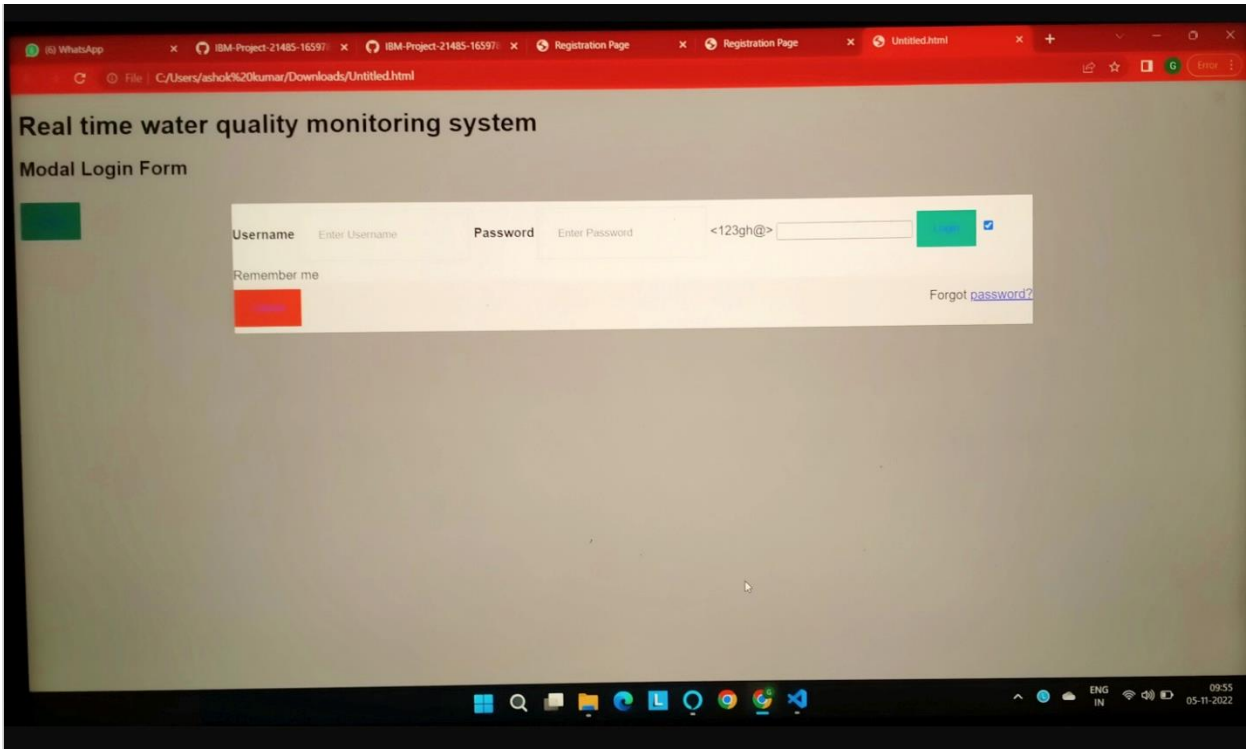
This code is written to add the feature of log in/sign up page for the user application for getting real time data and notifications

```
<!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>
```


Registration Page
Registration Page
Untitled.html
C:/Users/ashok%20kumar/Downloads/Untitled.html
Real time water quality monitoring system
Modal Login Form
Username
Enter Username
Password
Enter Password
<123gh@>
Remember me
Forgot password?

TESTING

User 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

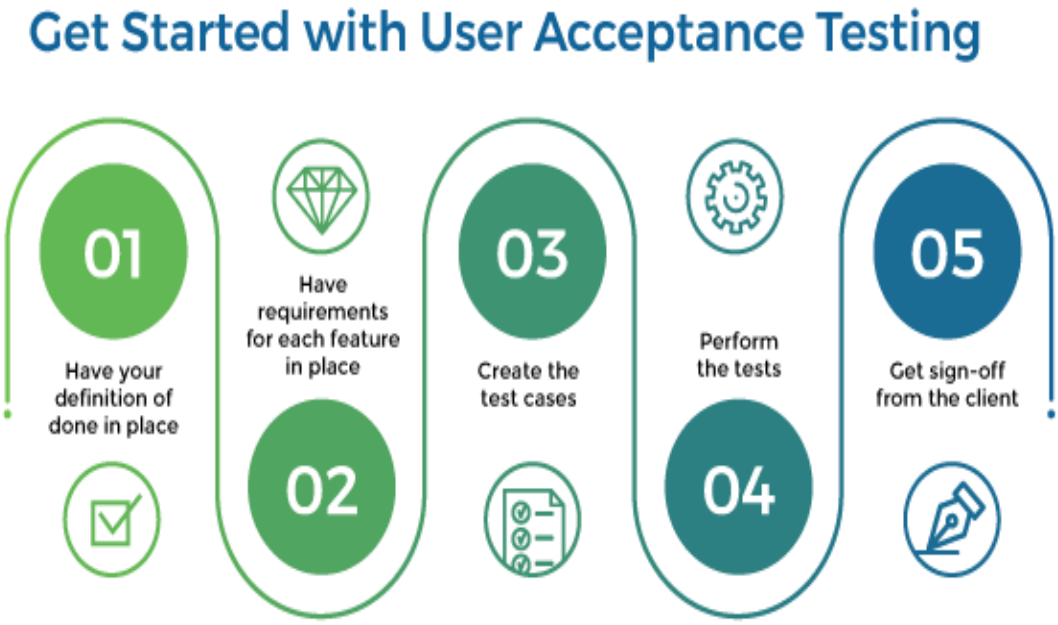
This report shows the number of test cases that have passed, failed, and untested

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

Testing 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

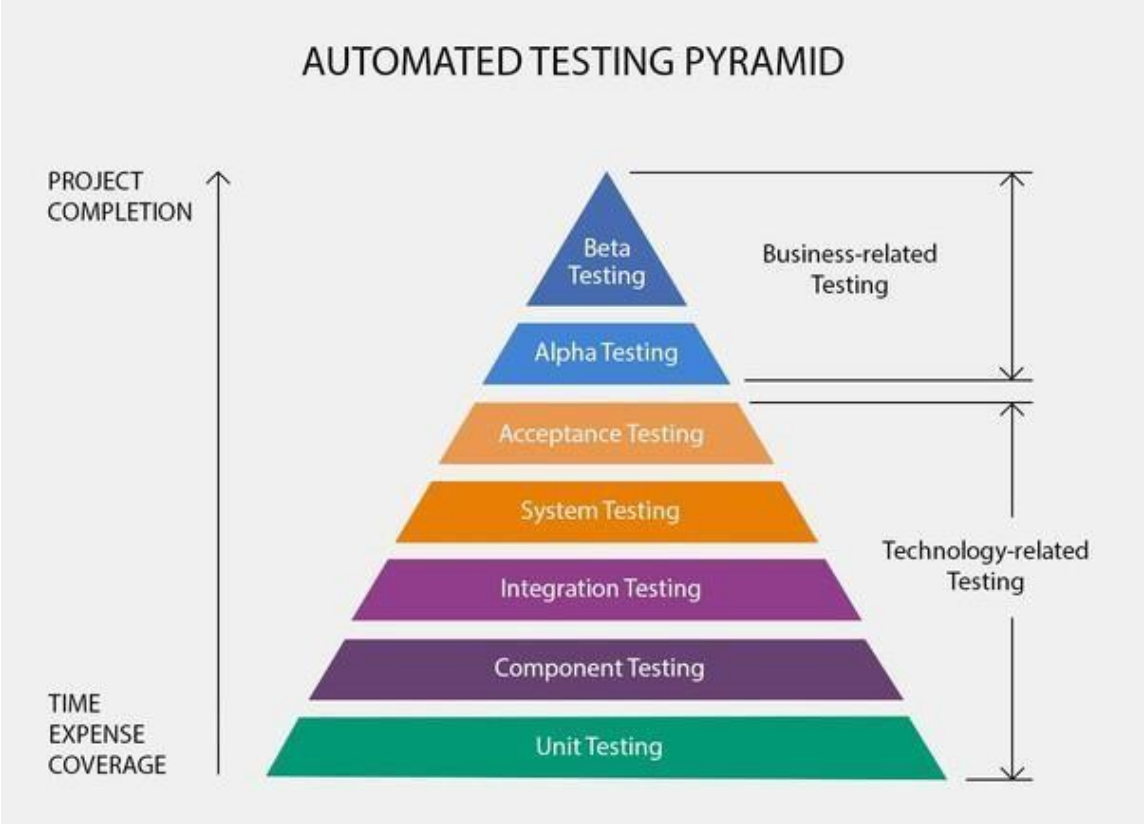
This step entails planning and setting up the base for the execution of the tests. Setting goals and identifying the aim(s) of the test phase based on the requirements supplied**.**

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.


AUTOMATED TESTING PYRAMID
PROJECT COMPLETION
TIME EXPENSE COVERAGE
Beta Testing
Alpha Testing
Acceptance Testing
System Testing
Integration Testing
Component Testing
Unit Testing
Business-related Testing
Technology-related Testing

RESULTS

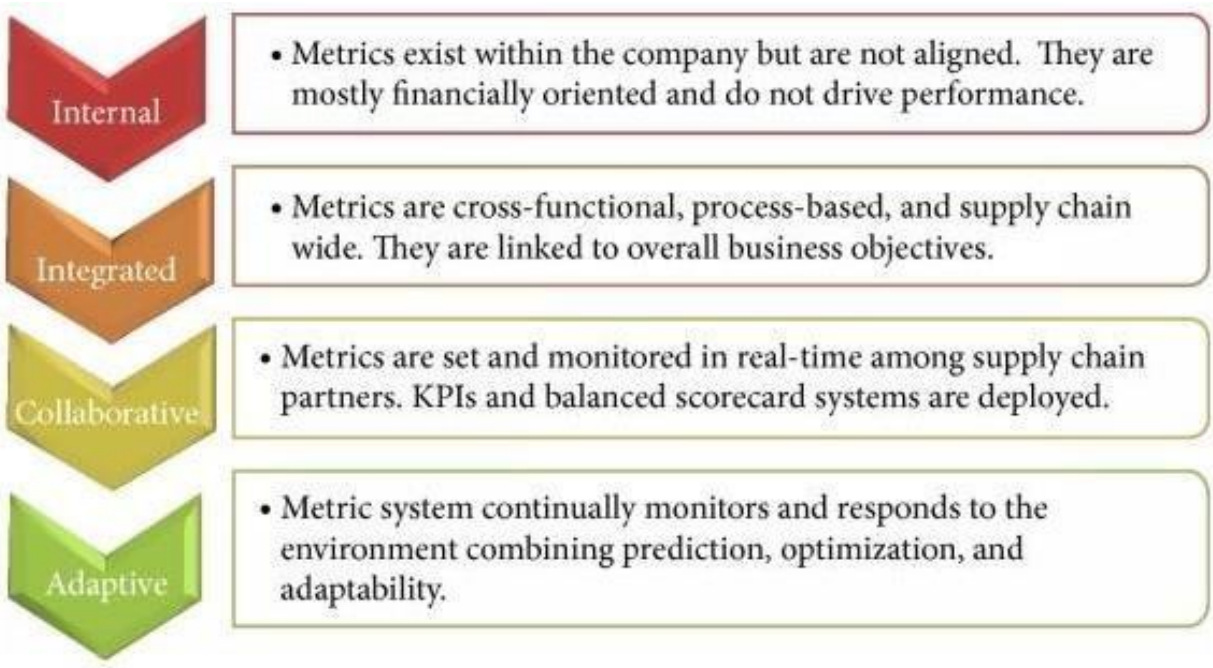
Performance Metrics:

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..

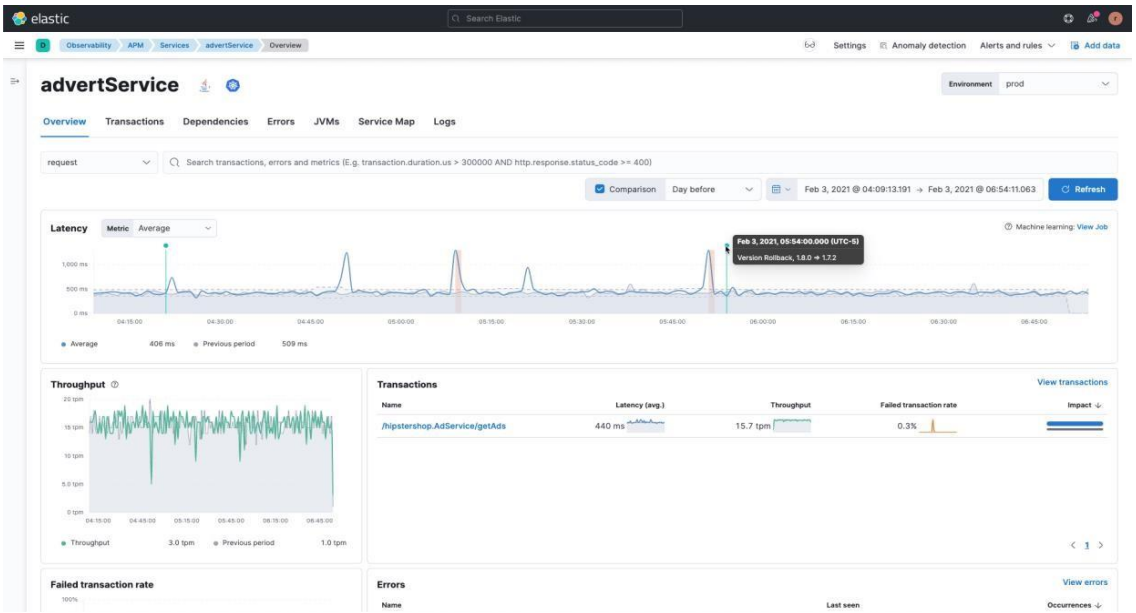
The execution trace starts with an operation called do Filter
that is commonly found as an entry point in web-based applications. It can be observed that the execution of the do Filter operation includes a sequence of additional nested operation executions, until the list operation performs a sequence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

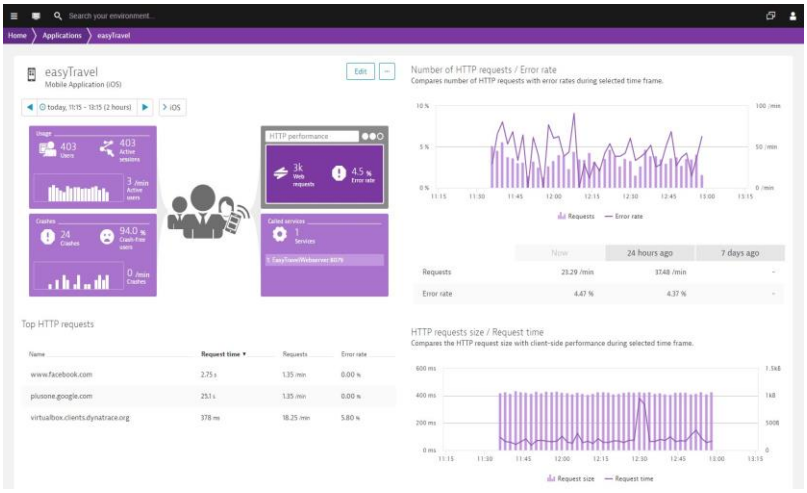
In the example, the response time for each operation execution is
included in the second column. Moreover, execution traces may include information such as the parameters of the operation executions.

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 performance-augmented 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.



ADVANTAGES & DISADVANTAGE

Benefits or Advantages of IoT based Water Quality Monitoring System

Following are the benefits or advantages of IoT based Water Quality Monitoring System are as follows.

➤The boat is mobile in nature and hence large number of samples are easily collected from different locations in less time.

➤It is very easy to maintain the IoT based water quality monitoring system as all the electronic boards are available in the boat itself.

➤The system is very cheap as the hardware and software does not cost much.

➤Machine learning techniques have made it very easy to plot the data collected in various formats for proper analysis.

➤Cloud storage platforms such as IBM CLOUD,azure helps in storing the sensor data immediately and wirelessly to the robust servers.

Disadvantages of IoT based Water Quality Monitoring System

Following are the disadvantages of IoT based Water Quality Monitoring System are as follows.

➤the system cannot provide real time monitoring of water parameters

➤For trouble shooting the system technicians is required and this process might take some time

CONCLUSION

Water pollution is one of the biggest threats to all living beings. Polluted water causes various diseases in humans, plants, animals, which, in turn, negatively impact the life cycle of the ecosystem. If the contamination is detected early on, suitable measures can be taken to preserve water quality or even upgrade it.

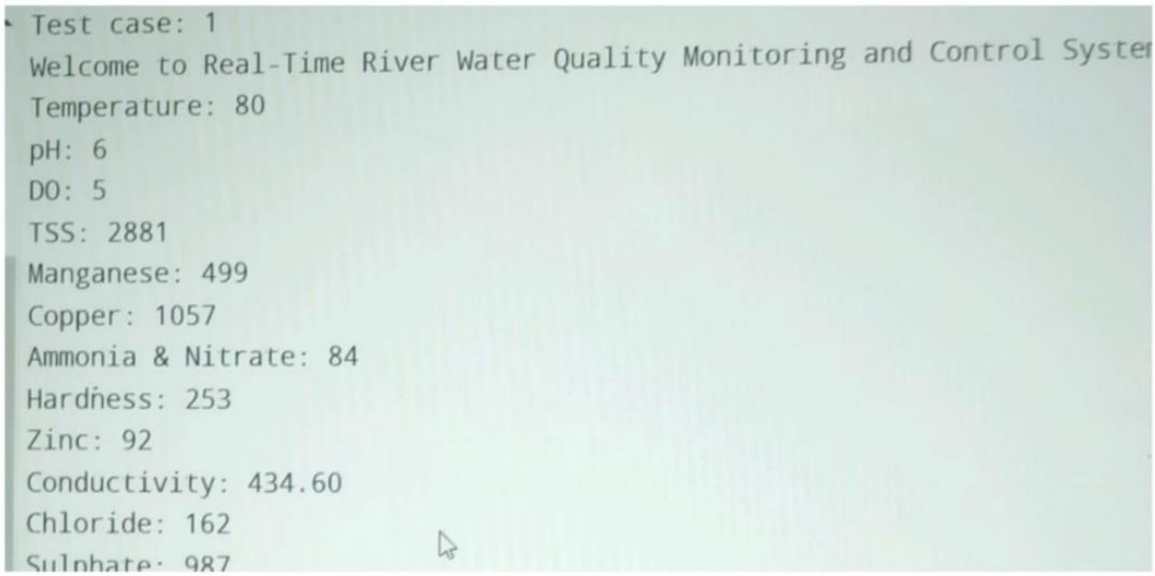
Therefore, Smart Water Quality Monitoring using IoT is paramount to supply pure water in real-time. Thanks to innovation in sensors, wireless modules, and communication devices, the activity is easy.

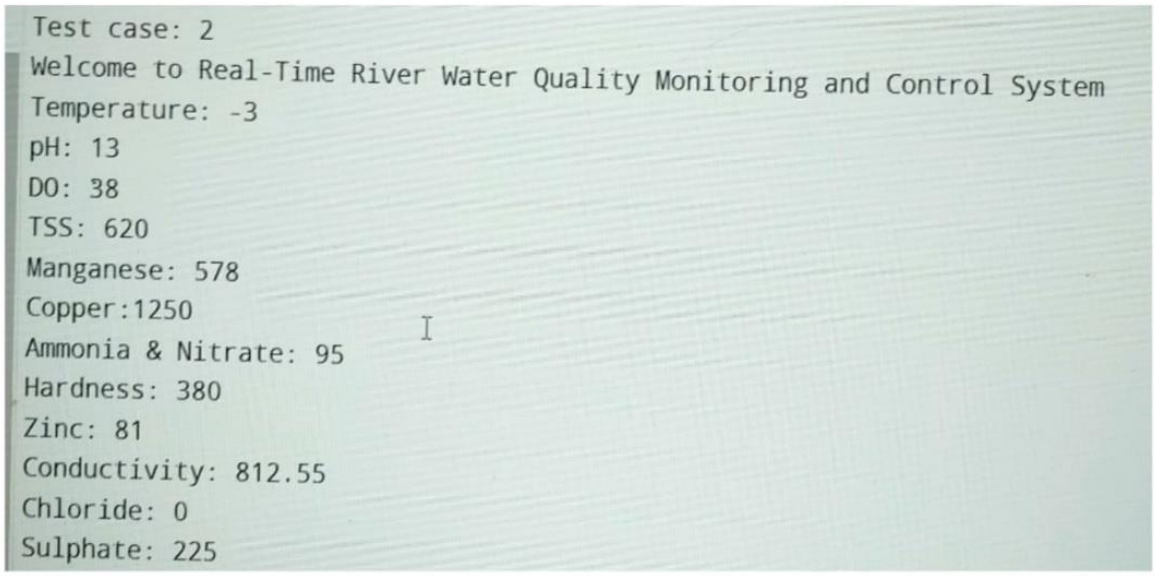
Source Code

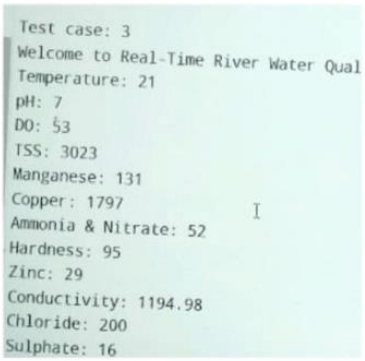
Python code for random value generation:

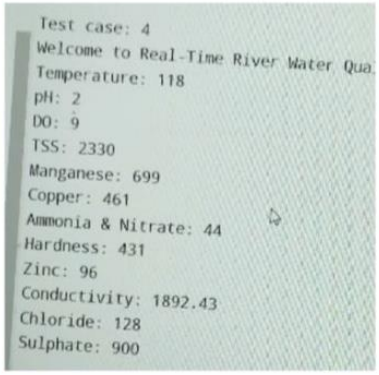
```
#importing Random function to generate the value
import random as rand


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"
   )
```

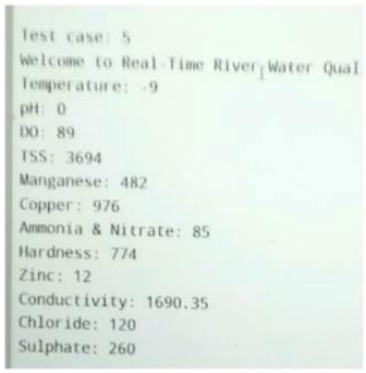
Output:


Test case: 1
Welcome to Real-Time River Water Quality Monitoring and Control Syste
Temperature: 80
pH: 6
DO: 5
TSS: 2881
Manganese: 499
Copper: 1057
Ammonia & Nitrate: 84
Hardness: 253
Zinc: 92
Conductivity: 434.60
Chloride: 162
Sulphate: 987


Test case: 2
Welcome to Real-Time River Water Quality Monitoring and Control System
Temperature: -3
pH: 13
DO: 38
TSS: 620
Manganese: 578
Copper:1250
Ammonia & Nitrate: 95
Hardness: 380
Zinc: 81
Conductivity: 812.55
Chloride: 0
Sulphate: 225


Test case: 3
Welcome to Real-Time River Water Qual
Temperature: 21
pH: 7
DO: 53
TSS: 3023
Manganese: 131
Copper: 1797
Ammonia & Nitrate: 52
Hardness: 95
Zinc: 29
Conductivity: 1194.98
Chloride: 200
Sulphate: 16


Test case: 4
Welcome to Real-Time River Water Qua
Temperature: 118
pH: 2
DO: 9
TSS: 2330
Manganese: 699
Copper: 461
Ammonia & Nitrate: 44
Hardness: 431
Zinc: 96
Conductivity: 1892.43
Chloride: 128
Sulphate: 900

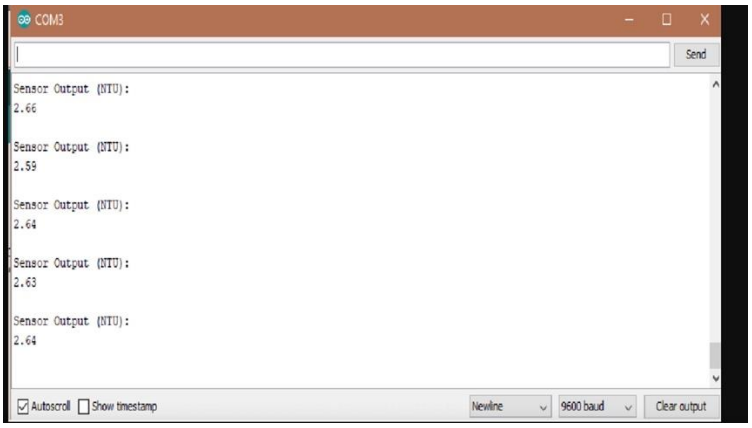

Test case: 5
Welcome to Real-Time River Water Qual
Temperature: -9
pH: 0
DO: 89
TSS: 3694
Manganese: 482
Copper: 976
Ammonia & Nitrate: 85
Hardness: 774
Zinc: 12
Conductivity: 1690.35
Chloride: 120
Sulphate: 260

Code for aurdino 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)
```

Git hub repository:

https://github.com/IBM-EPBL/IBM-Project-50295-1660902219.git

Demo video Link:

https://drive.google.com/file/d/1JNtO7mG3I9FBI_BUUDOtGYnT2_4QRbiS/view?usp=sharing

References:

[1]. Pandian D R, Dr. Mala K, “Smart Device to monitor water quality to avoid pollution in IoT environment” - International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE) ISSN: 0976-1353 Volume 12 Issue 2 –JANUARY 2015.

[2]. Shruti Sridharan, "Water Quality Monitoring System Using Wireless Sensor Network" - International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE) Volume 3, Issue 4, April 2014

[3]. R.Karthik Kumar, M.Chandra Mohan, S.Vengateshapandiyan, M.Mathan Kumar, R.Eswaran, " Solar based advanced water quality monitoring system using wireless sensor network " - International Journal of Science, Engineering and Technology Research (IJSETR), Volume 3, Issue 3, March 2014 ISSN: 2278 – 7798.

[4]. Jinmei Lu, Fuqing Yuan, Fengying Zhang, “Use of SVM Technique for Comparison of Water Quality: A Case Study from China” –International Journal of Performability Engineering, Vol. 11, No. 5, September 2015, pp.473-480.

[5]. Yue Liao, Jianyu Xu, Wenjing Wang, “A Method of Water Quality Assessment Based on Biomonitoring and Multiclass Support Vector Machine” – 2011 3rd International Conference on Environmental Science and Information Application Technology (ESIAT 2011)

[6]. Kashif Iqbal, Michael Odetayo, Anne James, “Enhancing The Low Quality Image Using Unsupervised Colour Correction Method”

[7]. Chong- Yi Li, Run-Min Cong, Yan-Wei Pang, Bo Wang, “Underwater Image Enhancement by Dehazing With Minimum Information Loss and HistogramDistribution Prior” – IEEE Transaction on Image Processing, Vol. 25, No.12, December 2016.

[8]. Zulhani Rasin and Mohd Rizal Abdullah, "Water Quality Monitoring System Using Zigbee Based Wireless Sensor Network "- International Journal of Engineering & Technology IJET - IJENS Vol:09 No:10.

[9]. Kirankumar G.Sutar , Prof.Ramesh T.Patil ," Wireless Sensor Network System to Monitor The Fish Farm" - Int. Journal of Engineering Research and Applications Vol. 3, Issue 5, Sep-Oct 2013, pp.194-197.

[10]. O’Flynn,B,Martinez-Catala, " A wireless sensor Network for water quality moniotoring"; 32 nd IEEE Conference on Local compute

[11]. Robby T. Tan, “Visibility in Bad Weather from a Single Image” – Imperial College London

[12]. An Ning, An Yu(2015),Monitoring System For Water Quality.

[13]. Dong He,Li-Xin Zhang(2017),The Water Quality Monitoring System Based On WSN

[14]. Flona Regan,Audrey Mc Carthy(2010),Smart Water Quality Monitoring System.

[15]. Izzatdin Abdul Aziz,Mazlina(2011),Water Quality Monitoring Systrm Using Wired Sensors.

[16]. Jayti Bhatia,Jignesh(2008),Real Time Water Quality Monitoring System.

[17]. Taufik Ibnu Salim,Hilman S.Alam and Rian P.Pratama(2018),Portable And Online Water Quality Monitoring System Using Wirless Sensor Network,435-440.

[18]. Vaishnavi V. Daigavane and Dr. M.A Gaikwad(2017), Water Quality Monitoring System Based On IOT ,10.

[19]. S. Geetha and S. Gouthami (2017), Internet Of Things Enable Realtime Water Quality Monitoring System

[20]. A.N.Prasad, K. A. Mamun, F. R. Islam, H. Haqva,(2015), Smart Water Quality Monitoring System.

[21]. Cho Zin Myint, Lenin Gopal, and Yan Lin Aung(2017), Reconfigurable Smart Water Quality Monitoring System in IoT Environment.