Project Report

1. **INTRODUCTION**
   - 1.1 Project Overview
   - 1.2 Purpose
2. **LITERATURE SURVEY**
   - 2.1 Existing problem
   - 2.2 References
   - 2.3 Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
   - 3.1 Empathy Map Canvas
   - 3.2 Ideation & Brainstorming
   - 3.3 Proposed Solution
   - 3.4 Problem Solution fit
4. **REQUIREMENT ANALYSIS**
   - 4.1 Functional requirement
   - 4.2 Non-Functional requirements
5. **PROJECT DESIGN**
   - 5.1 Data Flow Diagrams
   - 5.2 Solution & Technical Architecture
   - 5.3 User Stories
6. **PROJECT PLANNING & SCHEDULING**
   - 6.1 Sprint Planning & Estimation
   - 6.2 Sprint Delivery Schedule
   - 6.3 Reports from JIRA
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
   - 7.1 Feature 1
   - 7.2 Feature 2
   - 7.3 Database Schema (if Applicable)
8. **TESTING**
   - 8.1 Test Cases
   - 8.2 User Acceptance Testing
9. **RESULTS**
   - 9.1 Performance Metrics
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**

    Source Code

    GitHub & Project Demo Link

INTRODUCTION

1.1 Project Overview

- The conventional method of testing water quality is to gather samples of water manually and send to the lab to test and analyse. This method is time consuming, wastage of man power, and not economical. The water quality measuring system that we have implemented checks the quality of water in real time through various sensors (one for each parameter: pH, conductivity, temperature) to measure the quality of water.

- Sending random pH values and turbidity values will be sent to the IBM IoT platform
- Sensors values can be viewed in the Web Application
- Notifies the admin the random values cross the threshold value

1.2 Purpose

Water is the core resource and a vital for life of all species, as it is a limited resource that needs to be utilized efficiently. Monitoring various aspects of the water quality leads to a clear understanding of the aspects that should be considered for a healthy life and to avoid wastage of water. Using Internet of Things (IoT) should allow for the integration of real time monitoring and controlling of water quality.
The system will lead to real time data acquisition, transmission and processing of water quality data. This will give the ability to automatically react to the changes in the system outputs. Using Internet of Things (IoT) means the system can be accessed from anywhere through Internet, for example through a mobile application remotely.

2. LITERATURE SURVEY

2.1 Existing Problem

In the 21st century, there were lots of inventions, but at the same time were pollutions, global warming and so on are being formed, because of this there is no safe drinking water for the world's pollution. Nowadays, water quality monitoring in real time faces challenges because of global warming limited water resources, growing population, etc. Hence there is need of developing better methodologies to monitor the water quality parameters in real time

2.2 References

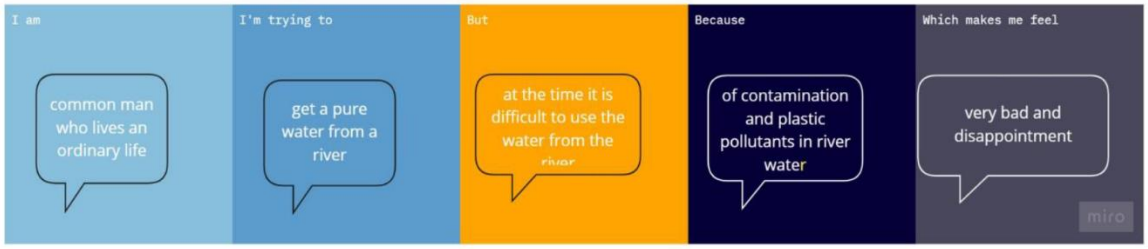
LITERATURE SURVEY:

TITLE	AUTHOR	PUBLICATION	CONTENTS
Water quality monitoring using wireless sensor networks: Current trends and future research directions	K. S. Adu-Manu, C. Tapparello, W. Heinzelman, F. A. Katsriku, and J.-D. Abdulai	ACM Transactions on Sensor Networks (TOSN), vol. 13, p. 4, 2017	Survey of the current state of the art in the design and implementation of WSN-based WQM systems, describing a framework for WSN-based WQM systems and discussing the technologies used at each stage in the monitoring process.
Real-time estimation of population exposure to PM2.5 using mobile- and station-based big data	B. Chen, Y. Song, T. Jiang, Z. Chen, B. Huang, and B. Xu	Int J Environ Res Public Health, vol. 15, Mar 23 2018	The proposed method in this paper can well quantify dynamics of the real-time population distribution and yield the estimation of population exposure to $PM_{2.5}$ concentrations and cumulative inhaled $PM_{2.5}$ masses with a 3-h updating frequency
Sensor based water quality monitoring system	B. Paul	BRAC University, 2018	Causes and effects of water pollution is presented, and comprehensive review of different methods of water quality monitoring and an efficient IoT based method for water quality monitoring has been discussed.
Smart Risk Assessment Systems using Belief-rule-based DSS and WSN Technologies	K. Andersson and M. S. Hossain	International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace and Electronic Systems	Described how a smart risk assessment system using belief-rule-based expert systems and WSN technologies could be built
The use of artificial neural networks for the prediction of water quality parameters	H. R. Maier and G. C. Dandy	Water resources Research, vol. 32, pp. 1013-1022, 1996	Analysis gives that ANN models appear to be a useful tool for forecasting salinity in rivers
The real time monitoring of water quality in IoT environment	N. Vijayakumar and R. Ramya	5 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), 2015, pp. 1-5	The design and development of the real-time monitoring of the water quality parameters in IoT environment is presented using water quality parameter sensors, Raspberry PI B+ core controller and an IoT module (USR WIFI 232)
An Interoperable IP based WSN for Smart Irrigation Systems	M. Z. Abedin, A. S. Chowdhury, M. S. Hossain, K. Andersson, and R. Karim	14th Annual IEEE Consumer Communications & Networking Conference, Las Vegas, 8-11 January 2017, 2017	Functionality of IOT is applied to agriculture like smart irrigation.Analysis of the performance of 6LoWPAN protocol stack

2.3 Problem Statement Definition

Problem Statement:

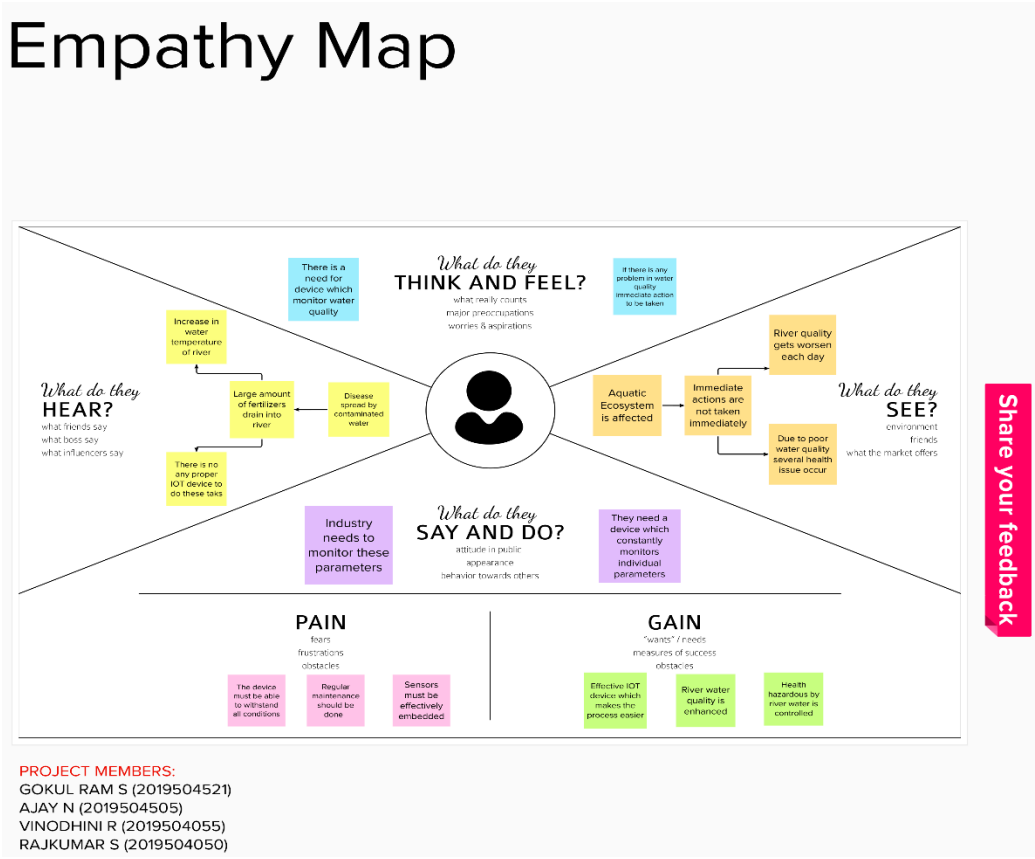
River water is a finite resource that is necessary for agriculture, industry and the survival of all living things on the planet, including humans. Sometimes the dangerous particles or chemicals are mixed in the river water and general purpose water purifier cannot purify that. And it's impossible to check the quality of river water manually in every time. Bathing in contaminated river waters causes skin diseases, allergies, and other such ailments. So an automatic real-time river water quality monitoring and control system is required to monitor the water reserved in our river water. And we can check the qulity of water anytime and from anywhere.



Problem Statement (PS)	Iam (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	common man who lives an ordinary life	get pure water from a river	at the time it is difficult to use the water from the river	of contamination and plastic pollutants in river water	very bad and disappointment
PS-2	A farmer from the agriculture site	get a pollutant free water from the river for agriculture field	at the time it is difficult to use the water from the river	of contamination and plastic pollutants in river water	very bad and disappointment

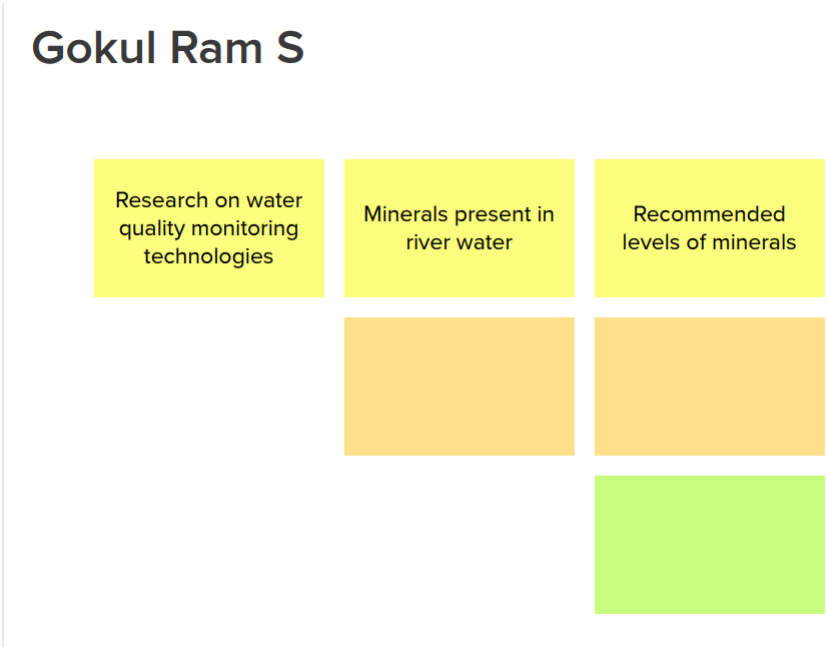
3.IDEATION AND PROPOSED SOLUTION

3.1 Empathy map canvas



PROJECT MEMBERS:
GOKUL RAM S (2019504521)
AJAY N (2019504505)
VINODHINI R (2019504055)
RAJKUMAR S (2019504050)

3.2 Ideation and Brainstorming



Ajay

Data collection of hazardous substance

Root cause of water pollution

Methods to prevent water contamination

Vinodhini R

Research on number of rivers flowing in our country

Comparing pollution levels

Research on real time implementation

Rajkumar S

Collecting Information on projects related to quality checking

Research on latest technologies available which are helpful for the project

Software development

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

Share the mural
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

Export the mural
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

Strategy blueprint
Define the components of a new idea or strategy.
Open the template →

Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.
Open the template →

Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
Open the template →

Share template feedback

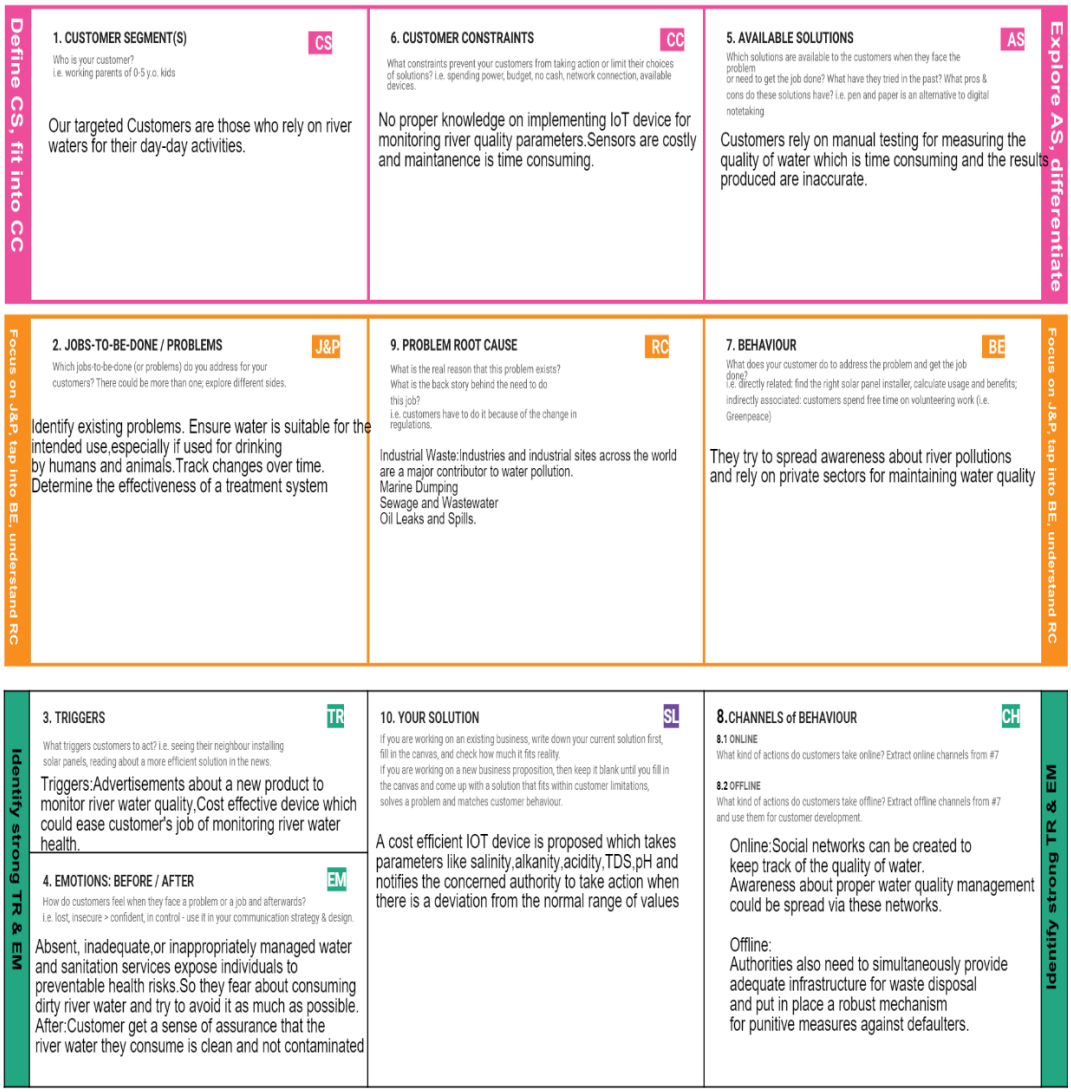
3.3 Proposed Solution

Proposed Solution:

S.No	Parameter	Description
1.	Problem Statement (Problem to be solved)	Due to the fast growing urbanization supply of safe drinking water is a challenge for the every city authority. Water can be polluted any time. So the water we reserved in the water tank at our roof top or basement in our society or apartment may not be safe.
2.	Idea / Solution description	An automatic real-time monitoring system is required to monitor the health of the water reserved in our water tank of the society or apartment. So it can warn us automatically if there is any problem with the reserved water.
3.	Novelty / Uniqueness	We can check the quality of the water anytime and from anywhere.
4.	Social Impact / Customer Satisfaction	The main aim is to develop a system for continuous monitoring of river water quality at remote places using wireless sensor networks with low power consumption, low-cost and high detection accuracy.
5.	Business Model (Revenue Model)	The Proposed IOT can be a good profitable product since the some of the services like Cloud Storage,Web Application involves some subscription fee inorder to access the services.
6.	Scalability of the Solution	The proposed solution involves collection of data from River and the collected data can be monitored from anywhere and at anytime using the help of IOT Technology. The collected data is constantly analyzed and if there

3.4 Proposed Solution Fit

Project Title :Real-Time River Water Quality Monitoring and control system **Project Design Phase-I - Solution Fit Template** **Team ID: PNT2022TMID35857**



4 REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Login	Confirmation through verified password
FR-2	User Authorization levels	Confirmation via Email,Confirmation via OTP
FR-3	Historical Data	The Data are stored in the cloud from the beginning stage till the updation
FR-4	User Authentication	The credentials is accessible only to the authorized users to access the application
FR-5	User Rules and laws	There is some specific guidelines which has to be followed by the users
FR-6	Logout	Logs out the user successfully.

4.2 Non-Functional Requirements

Non-functional Requirements:

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

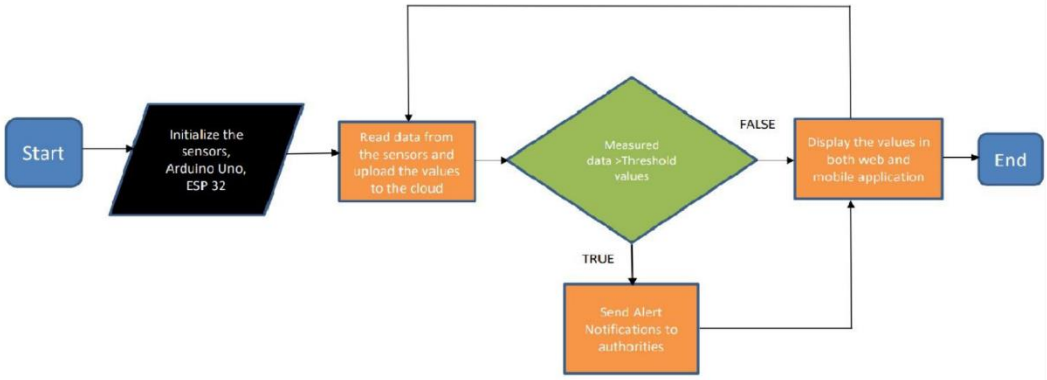
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The final output should be understandable
NFR-2	**Security**	The application is designed in a secure manner to maintain privacy of the user.
NFR-3	**Reliability**	Even if there is any hardware issues the last updated data's are stored in default manner
NFR-4	**Performance**	High hardware components are used which provide meticulous performance
NFR-5	**Availability**	The model is designed in such a way that are available,usable and could be modified anytime
NFR-6	**Scalability**	The system is scaled depending on the size of the water body.

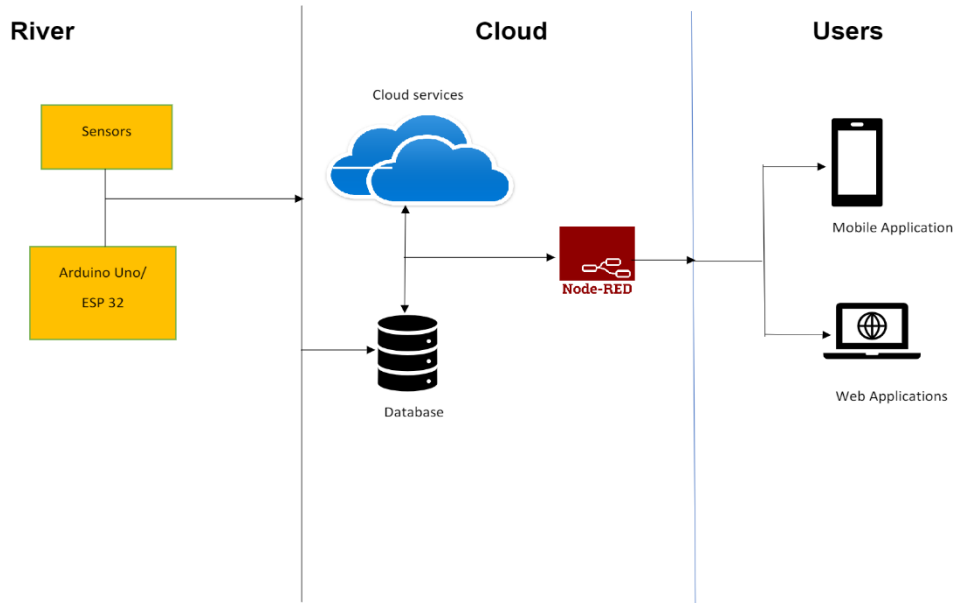
5 PROJECT DESIGN

5.1 Data flow diagrams

Data Flow Diagrams:

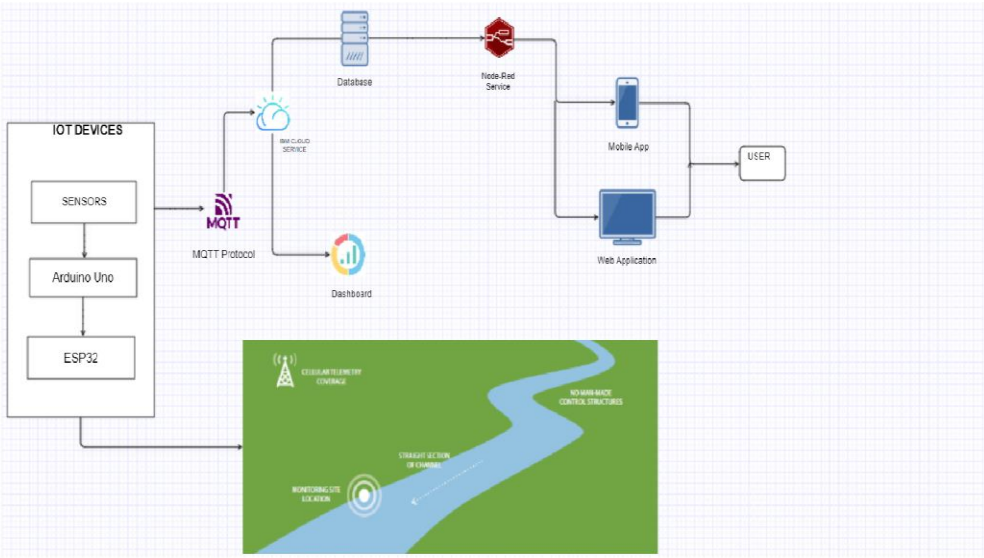
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is



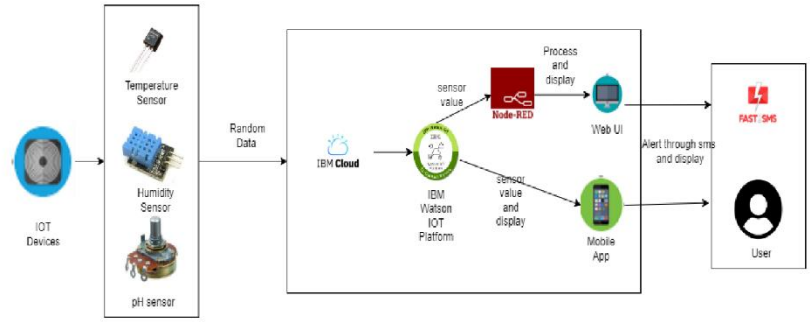


5.2 Solution & Technical Architecture

Solution Architecture:



Technical Architecture:



5.3 User Stories

User Stories

Use the below template to list all the user stories for the product.

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 email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
	Registration	USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-2
	Dashboard	USN-3	As a user, I can monitor the condition of the River water in my mobile application	I can access the dashboard and monitor the quality of river water	High	Sprint-3
Customer (Web user)	Registration	USN-1	As a user, I can login to the web application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
	Registration	USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-2
	Dashboard	USN-3	As a user, I can monitor the condition of the River water in my web application dashboard	I can access the dashboard and monitor the quality of river water	High	Sprint-3
Authorities	Registration	AUT-1	As an authority, I can register for either web/mobile application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
	Registration	AUT-1	As an authority, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-2
	Notification	AUT-1	As an authority I will receive notification when the parameters of water exceed a certain threshold levels	I will receive notifications on certain conditions	High	Sprint-3

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Mobile UI	USN-1	As a user,I can study the river water quality by registering into Mobile app	10	High	Gokul Ram S, Ajay S
Sprint-1	Secure Login	USN-2	As a user, I can login into App securely and my login credentials are securely stored in database	5	High	Gokul Ram S, Ajay S,Vinodhini R
Sprint-1	Alerting Authority	USN-3	As a user, I can alert the authority by sending mail or SMS using Mobile App	5	Low	Ajay S,Rajkumar S
Sprint-2	Node-Red Web UI design	USN-4	As a user, I can see the water parameters in web application dashboard	20	Medium	Gokul Ram S Vinodhini R
Sprint-3	Python code	USN-5	Sending Sensor data values to IBM Watson cloud using python code.	20	High	Vinodhini R Rajkumar S
Sprint-4	Monitoring	USN-6	For Real-Time water quality monitoring, messages are immediately send to concerned authorities when parameters cross threshold.	20	High	Gokul Ram S,Ajay S Vinodhini R,Rajkumar S

6.2 Sprint Delivery Schedule

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

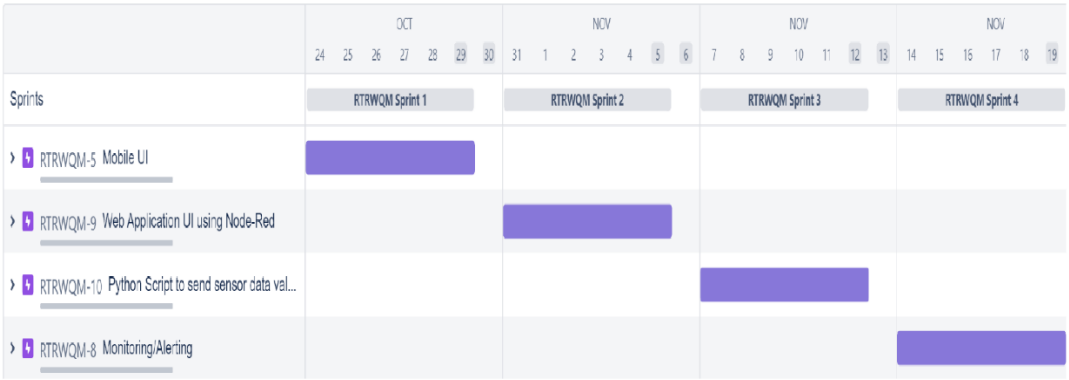
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022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

Velocity:

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

6.3 Reports from JIRA

Project planning using JIRA software

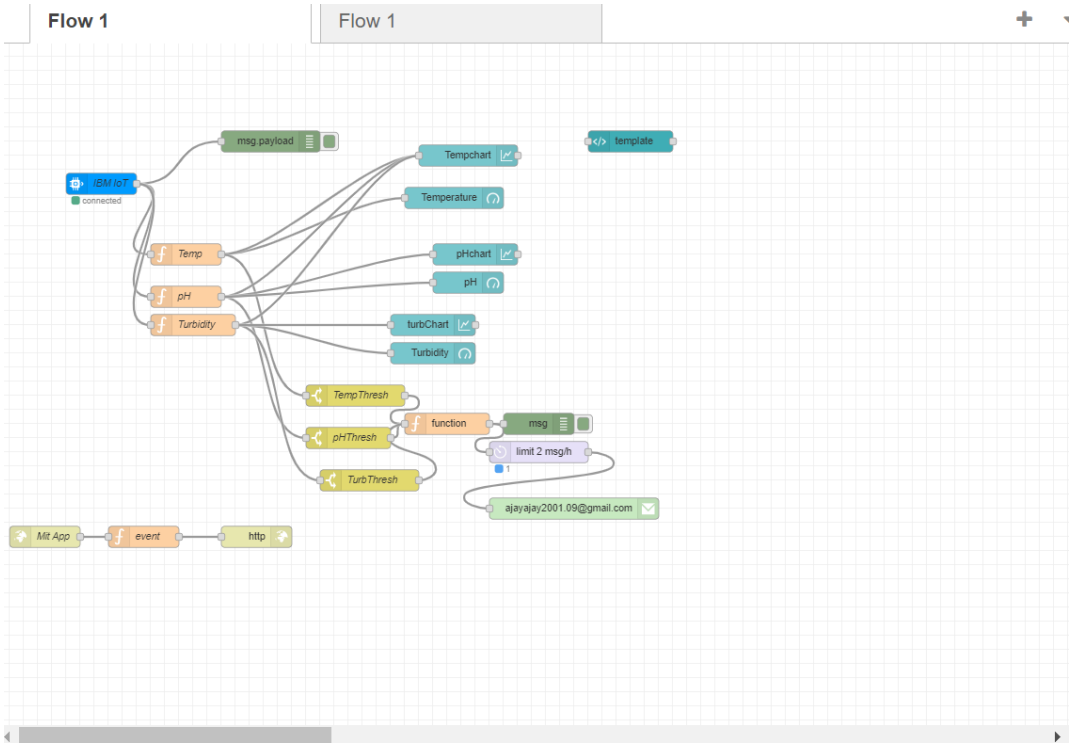


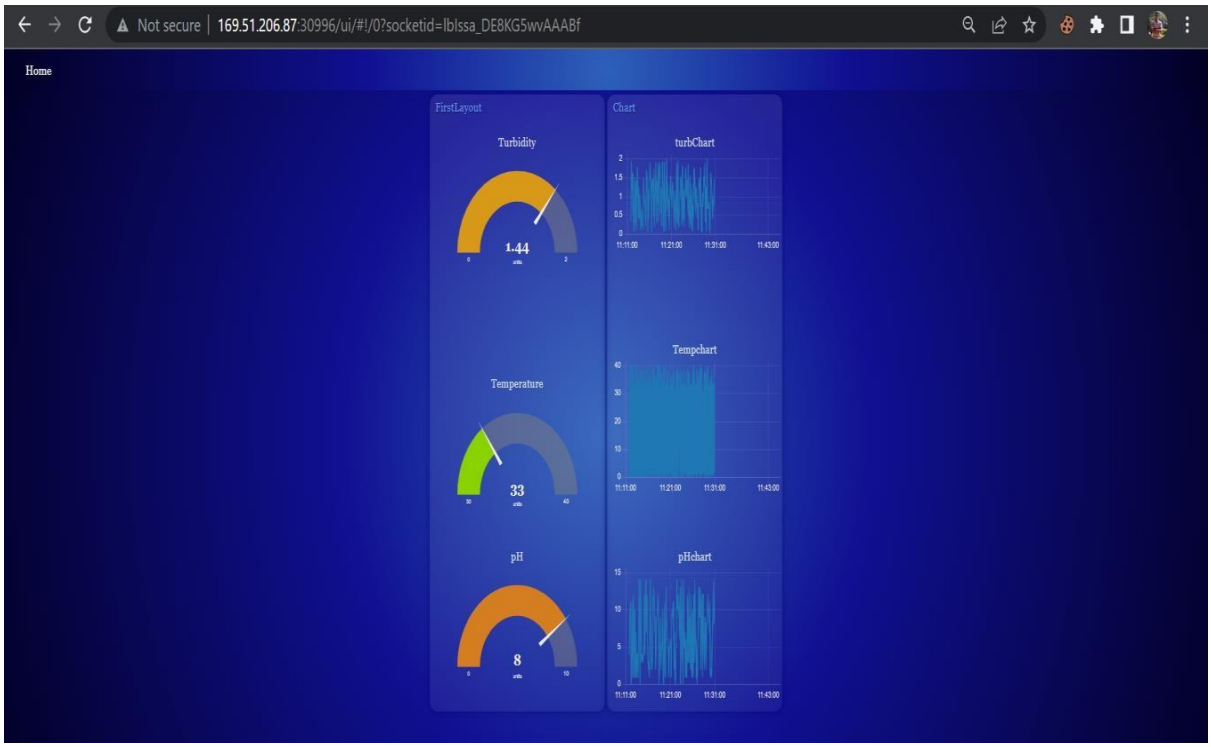
7) CODING AND SOLUTIONING

7.1) Feature 1:

Web UI:

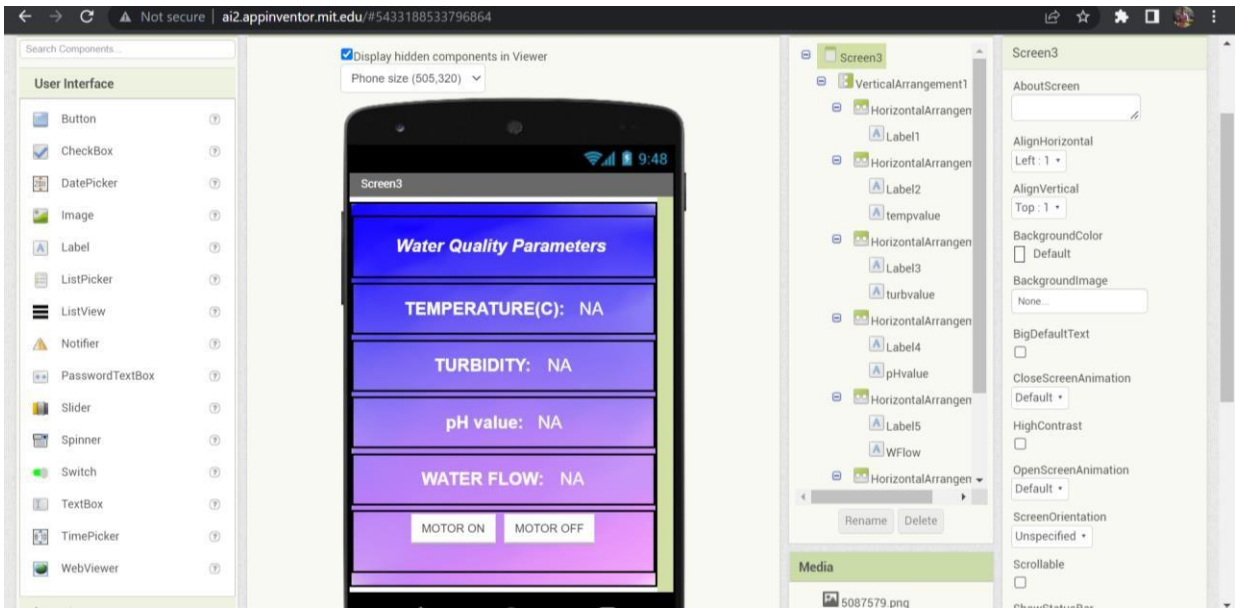
Web UI is created using node-red, which display the current conditions of the river water. The current temperature, turbidity and pH of the water can be visualized in a graphical chart. Real time monitoring of river water is enabled by the Web UI.

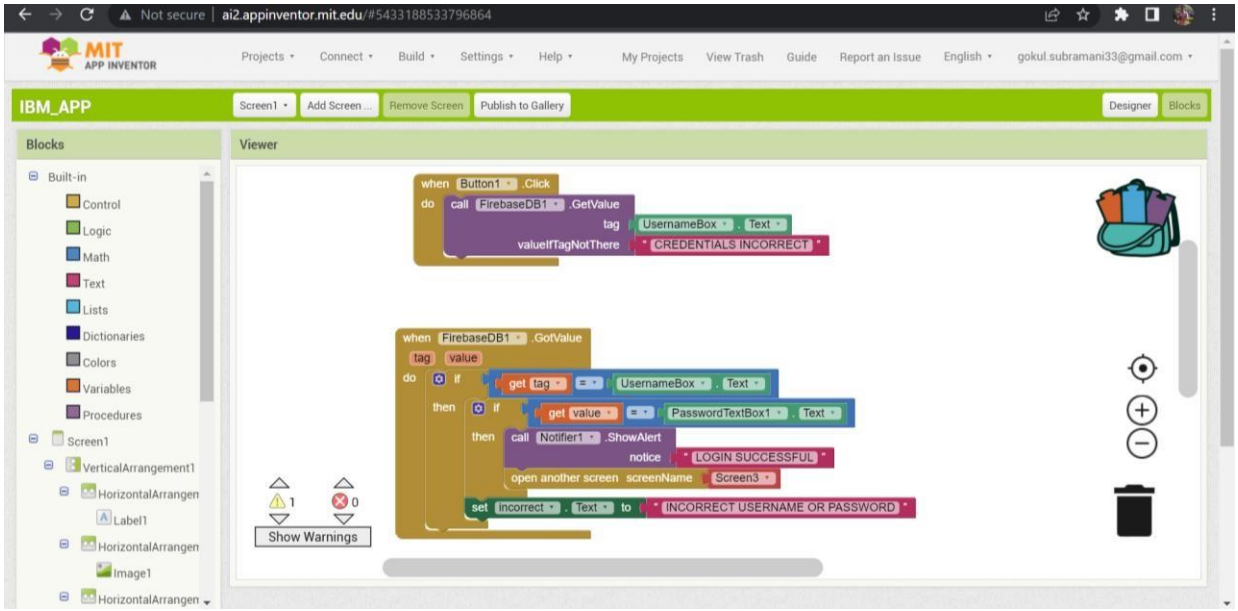


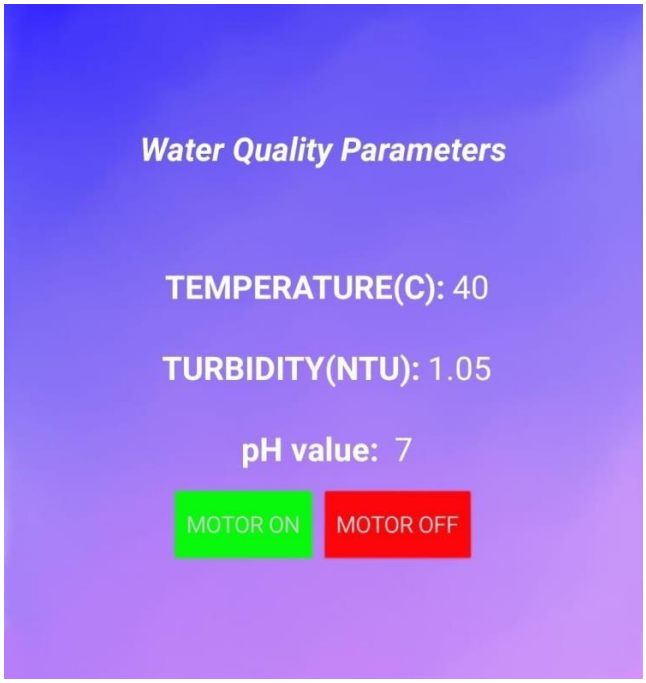


Mobile UI:

Mobile UI is created using MIT App Inventor which can be installed across various mobile devices. This mobile application receives data from IBM Cloud and displays the current condition of the river water. Only registered user can access this mobile application which enables confidentiality.







7.2) Feature 2:

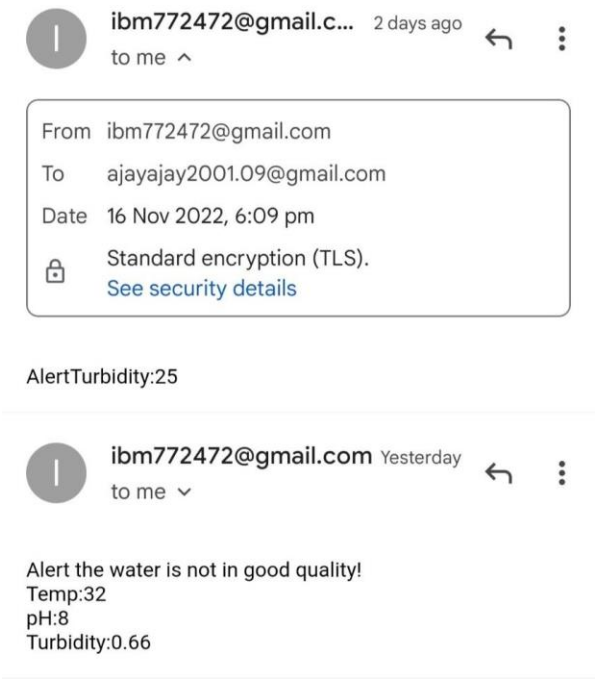
SMS and Mail Alert:

Whenever the quality of water is not optimal, alerts are sent in the form of SMS and mail to the concerned authorities. This enables the authorities to take measures regarding the quality of water. Mail is sent via Node-Red.

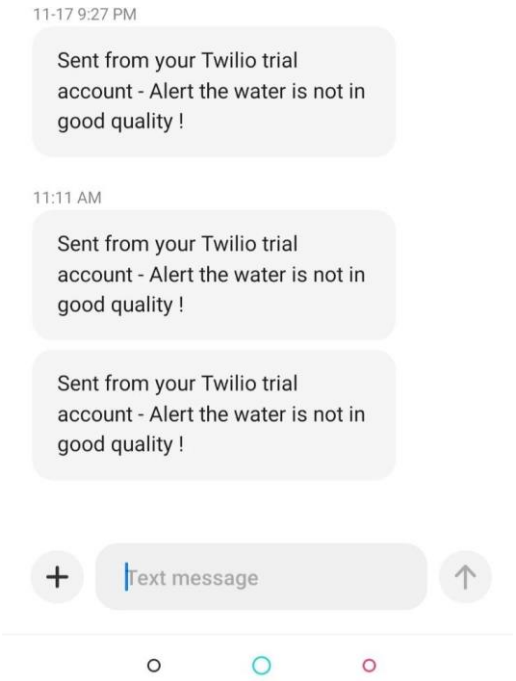
```python
import os
from twilio.rest import Client

account_sid = 'ACb4d033465895822c34e656bf6be69384'
auth_token = '6916b3bf66a451937068378db5a9692a'
client = Client(account_sid, auth_token)

message = client.messages.create(
                              messaging_service_sid='MG3d02a8b50e684c345993182610957703',
                              body='Alert the water is not in good quality !',
                              from_='+16294006922',
                              to='+919442130329'
                          )

print(message.sid)
```

This python script uses Twilio API to send SMS to the concerned authority

Mail sent to the concerned authorities:

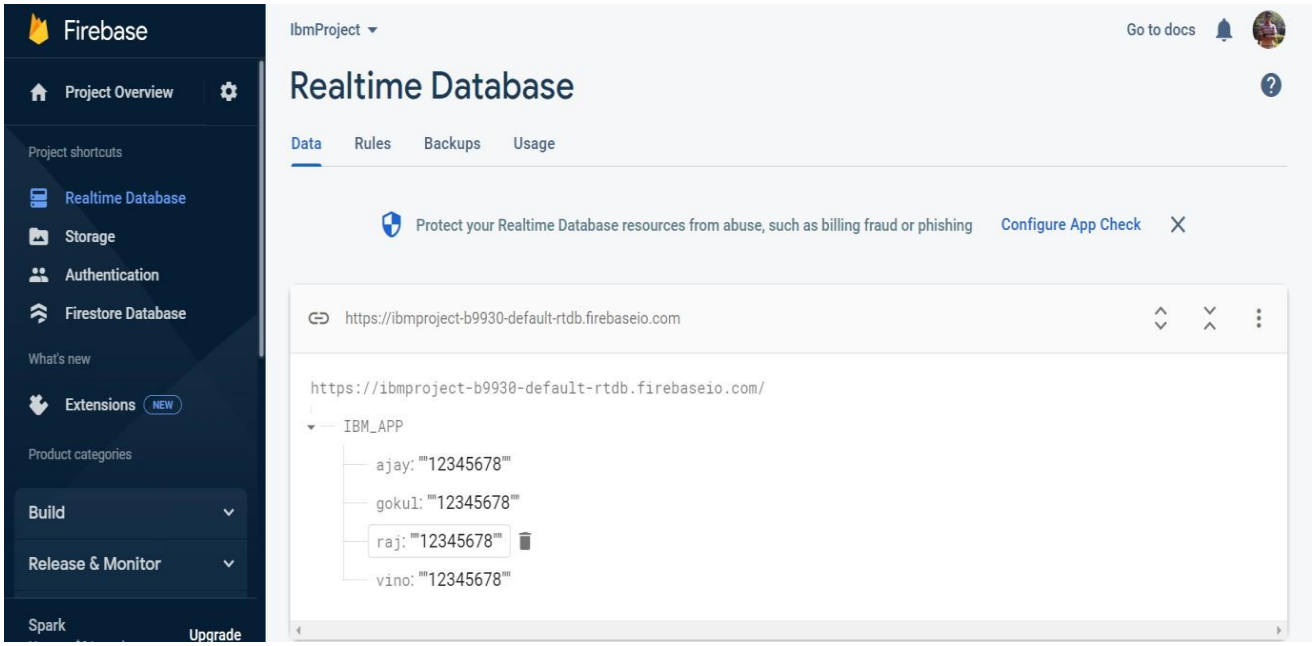


SMS Sent to concerned Authorities:



Database Schema:

The User login credentials of the Mobile UI are stored in Firebase database



8)TESTING:

8.1) Test cases:

Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	5	0	0	5
Client Application	10	0	0	10
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	2	0	0	2
Final Report Output	4	0	0	4
Version Control	3	0	0	3

8.2) User Acceptance testing:

1)Purpose of document:

The purpose of this document is to briefly explain the test coverage and open issues of the Real-Time River Water Quality Monitoring and Control System project at the time of the release to User Acceptance Testing (UAT).

2)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	5	2	1	2	10
Duplicate	1	0	0	1	2
External	2	3	0	1	6
Fixed	8	2	4	14	28
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	0	2	1	3

Totals	16	7	9	20	52

RESULTS:

9.1) Performance Metrics:

NFT- Risk assessment:

S.No	Project Name	Scope/feature	Functional Changes	Hardware Changes	Software Changes	Impact of Downtime	Load/Voluem Changes	Risk Score	Justification
1	Real-Time River Water Quality Monitoring and Control System	New	Low	No Changes	Moderate		>5 to 10%	ORANGE	As we have seen the changes

PERFORMANCE TABLE

PARAMETER	PERFORMANCE	DESCRIPTION
ADMIN TESTING	90%-100%	THE TESTING IS DONE BEFOR THE DEPLOYMENT OF APPLICATION
CUSTOMER SATISFACTION	70-80%	THE SATISFACTION OF THE CUSTOMER WITH RESPECT TO MOBILE AND WEB APPLICATION
USER INTERFACE	80-90%	THE MOBILE APPLICATION CAN BE USED BY REGISTERED USERS
SEVER RESPONSE	50-75%	URL- response

DATA VALIDATION WITH NO. OF TEST CASE	60-80% (20-30 TESTCASE)	VALID DATA FROM THE APP
ERROR	2-3%	REAL-TIME DELAY MAY OCCUR

10.ADVANTAGRES AND DISADVANTAGES:

ADVANTAGES:

- The designed IOT device can be beneficial to many users ranging from farmers to common man. They have the flexibility to monitor the river-water quality parameters anytime and anywhere.
- The automation process involved saves a lot of time,money and also aids to avoid manual labour intervention
- Water quality is monitored in real-time and the water parameters can be constantly seen without having the difficulty to go to the exact River location to monitor.
- The overall condition of Flora and Fauna living in river-water is improved and thereby ensuring a proper ecosystem.

DISADVANTAGES:

- Regular maintenance of IOT device is required to ensure smooth working at all time.

- Water may contain salt which may corrode the IOT device and make it unusable.
- Sometime the water quality may be good in a certain area but bad in other parts of river-water. So wide coverage of monitoring is a problem.
- Sometimes there might be certain latency in receiving the data.

CONCLUSION:

So with the proposed IOT device it is possible to monitor river water quality in real-time. The system is cost-effective,eco-friendly and does not need any people on duty thus ensuring a good efficient solution. The proposed model is also more flexible, if we want to analyse any other parameter of the water quality, the extra sensors can be easily appended with the existing system. With the designed Web-UI and Mobile App it becomes easy for the user to see the water parameters at anytime and anywhere they want. The system can be expanded to monitor hydrologic, air pollution, industrial and agricultural production and so on. It has widespread application and extension value.

FUTURE SCOPE:

- GSM module can be appended to the existing system to know the exact location of the sensors in the river
- A boat like architecture can be designed and controlled by remote anywhere to know the water quality at any location in the river required.
- Several other water parameters can be monitored by making use of appropriate sensors
- Machine learning algorithm can be fed into the system to predict the quality of the water in advance.

13.APPENDIX:

13.1.Source code

Python script to publish data:

```
import paho.mqtt.client as mqtt
import time
import random
import json
import sms
import requests


def run():
    ORG ="q6sux6"
    DEVICE_TYPE ="ESP32"
    DEVICE_ID ="GokulEsp32"
    TOKEN ="gp5PA9ljfw7jf9cV-g"

    server = ORG + ".messaging.internetofthings.ibmcloud.com";
    pubTopic1 = "iot-2/evt/temp/fmt/json"
    pubTopic2 = "iot-2/evt/pH/fmt/json"
    pubTopic3 = "iot-2/evt/turb/fmt/json"
    #pubTopic3 = "iot-2/evt/wf/fmt/json"

    authMethod = "use-token-auth";
    token = TOKEN;
    clientId = "d:" + ORG + ":" + DEVICE_TYPE + ":" + DEVICE_ID;

    mqttc = mqtt.Client(client_id=clientId)
    mqttc.username_pw_set(authMethod, token)
    mqttc.connect(server, 1883, 60)

    while True:
        try:

            # Print the values to the serial p  ort
            temperature_c = random.randint(30,40) * 1.0
            temperature_f = temperature_c * (9 / 5) + 32.0
            pH = random.randint(0,14)* 1.0
            turb=random.uniform(0,2)

            print(
                "Temp: {:.2f} F / {} C    pH: {}  Turbidity:{:.2f}NTU".format(
                    temperature_f, temperature_c, pH,turb
                )
            )
            payload={"temp":temperature_c,"pH":pH,"turb":round(turb,2)}
```

```
            if(temperature_c>35 or not(6.5<pH<8.5) or (turb>1)):
                sms.send_sms()
                print("WATER IN BAD QUALITY,SMS SEND SUCCESSFULLY!")

            mqttc.publish(pubTopic1,json.dumps(payload))

            #mqttc.publish(pubTopic2,pH)
            #mqttc.publish(pubTopic3,round(turb,2))

            print("Published")
            req=requests.get("http://169.51.206.87:30996/motor")
            cmd=req.json()
            if(cmd!={}):
                print("MOTOR IS",cmd['motor'])
            time.sleep(5)

        except RuntimeError as error:
            print(error.args[0])
            time.sleep(2.0)
        except Exception as error:
            print("Error encountered!")
            time.sleep(5.0)
    mqttc.loop_forever()
if __name__=='__main__':
    run()
```

SMS module:

```
import os
from twilio.rest import Client

account_sid = 'ACb4d033465895822c34e656bf6be69384'
auth_token = '[redacted]'
client = Client(account_sid, auth_token)
def send_sms():
    message = client.messages.create(
                              messaging_service_sid='MG3d02a8b50e684c345993182
610957703',
                              body='Alert the water is not in good quality
!',
                              from_='+16294006922',
                              to='+919442130329'
                          )

    print(message.sid)
```

Web UI json object:


```
[{"id":"dce59bb1b197e846","type":"tab","label":"Flow
1","disabled":false,"info":"","env":[]},{"id":"29e840d901135fd2","type":"debug
","z":"dce59bb1b197e846","name":"","active":true,"tosidebar":true,"console":fa
lse,"tostatus":false,"complete":"payload","targetType":"msg","statusVal":"","s
tatusType":"auto","x":390,"y":140,"wires":[]},{"id":"c1bae0273a02f744","type":
"ibmiot
in","z":"dce59bb1b197e846","authentication":"apiKey","apiKey":"65e8b07c8b314e5
4","inputType":"evt","logicalInterface":"","ruleId":"","deviceId":"GokulEsp32"
,"applicationId":"","deviceType":"ESP32","eventType":"+","commandType":"","for
mat":"json","name":"IBM
IoT","service":"registered","allDevices":"","allApplications":"","allDeviceTyp
es":"","allLogicalInterfaces":"","allEvents":true,"allCommands":"","allFormats
":true,"qos":0,"x":150,"y":200,"wires":[["29e840d901135fd2","5b043998cdca8c53"
,"4b56152a0169a71e","0a95aa0c09598668","8f96553a86aabad7"]]},{"id":"5b043998cd
ca8c53","type":"function","z":"dce59bb1b197e846","name":"Temp","func":"msg.pay
load=msg.payload.temp\nglobal.set(\"t\",msg.payload)\nreturn
msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":270,"y
":300,"wires":[["9839e45b5083a4f4","b3cf1e72b912870f","2d10b021817679d4"]]},{"
id":"9839e45b5083a4f4","type":"ui_gauge","z":"dce59bb1b197e846","name":"","gro
up":"bef2c6e7f6eac166","order":4,"width":0,"height":0,"gtype":"gage","title":"
Temperature","label":"units","format":"{{value}}","min":"30","max":"40","color
s":["#00b500","#e6e600","#ca3838"],"seg1":"","seg2":"","className":"","x":650,
"y":220,"wires":[]},{"id":"4b56152a0169a71e","type":"function","z":"dce59bb1b1
97e846","name":"pH","func":"msg.payload=msg.payload.pH\nglobal.set(\"p\",msg.p
ayload)\nreturn
msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":270,"y
":360,"wires":[["b3cf1e72b912870f","ed93af9b46faef11","de2fcf253056e1de","dff3
497162207c84"]]},{"id":"b3cf1e72b912870f","type":"ui_chart","z":"dce59bb1b197e
846","name":"","group":"d633df785d60f78c","order":4,"width":7,"height":5,"labe
l":"Tempchart","chartType":"line","legend":"false","xformat":"HH:mm:ss","inter
polate":"linear","nodata":"","dot":false,"ymin":"","ymax":"","removeOlder":1,"
removeOlderPoints":"","removeOlderUnit":"3600","cutout":0,"useOneColor":false,
"useUTC":false,"colors":["#1f77b4","#aec7e8","#ff7f0e","#2ca02c","#98df8a","#d
62728","#ff9896","#9467bd","#c5b0d5"],"outputs":1,"useDifferentColor":false,"c
lassName":"temp","x":670,"y":160,"wires":[[]]},{"id":"cef2b9df42d00326","type"
:"http in","z":"dce59bb1b197e846","name":"Mit
App","url":"/sensor","method":"get","upload":false,"swaggerDoc":"","x":70,"y":
700,"wires":[["de0c5ccb5a0c042e"]]},{"id":"de0c5ccb5a0c042e","type":"function"
,"z":"dce59bb1b197e846","name":"event","func":"msg.payload={\"temp\":global.ge
t(\"t\"),\"pH\":global.get(\"p\"),\"turb\":global.get(\"tu\")};\nreturn
msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":230,"y
":700,"wires":[["cd4d58b914319ebd"]]},{"id":"cd4d58b914319ebd","type":"http
response","z":"dce59bb1b197e846","name":"","statusCode":"","headers":{},"x":39
0,"y":700,"wires":[]},{"id":"0a95aa0c09598668","type":"function","z":"dce59bb1
b197e846","name":"Turbidity","func":"msg.payload=msg.payload.turb\nglobal.set(
```

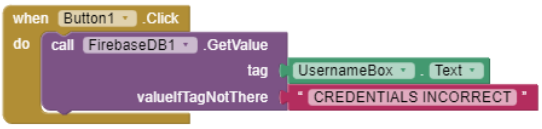
```
\"tu\",msg.payload)\nreturn
msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":280,"y
":400,"wires":[["afbd902ecc12fc70","b3cf1e72b912870f","c1040d0c48eab14f","6b4f
cbd80a06314c"]]},{"id":"afbd902ecc12fc70","type":"ui_gauge","z":"dce59bb1b197e
846","name":"","group":"bef2c6e7f6eac166","order":1,"width":0,"height":0,"gtyp
e":"gage","title":"Turbidity","label":"units","format":"{{value}}","min":"0","
max":"2","colors":["#00b500","#e6e600","#ca3838"],"seg1":"","seg2":"","className
":"","x":620,"y":440,"wires":[]},{"id":"036d350250483fe0","type":"e-
mail","z":"dce59bb1b197e846","server":"smtp.gmail.com","port":"465","secure":t
rue,"tls":true,"name":"ajayajay2001.09@gmail.com","dname":"","x":820,"y":660,"
wires":[]},{"id":"42416ae236f9cf82","type":"function","z":"dce59bb1b197e846","
name":"","func":"msg={\n    payload:\"Alert the water is not in good
quality!\"+\"\\nTemp:\"+global.get(\"t\")+\"\\npH:\"+global.get(\"p\")+\"\\nTu
rbidity:\"+global.get(\"tu\")\n}\nreturn
msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":640,"y
":540,"wires":[["bfa1fa65b3802702","3b7994850ba3ac45"]]},{"id":"bfa1fa65b38027
02","type":"debug","z":"dce59bb1b197e846","name":"","active":true,"tosidebar":
true,"console":false,"tostatus":false,"complete":"true","targetType":"full","s
tatusVal":"","statusType":"auto","x":770,"y":540,"wires":[]},{"id":"2d10b02181
7679d4","type":"switch","z":"dce59bb1b197e846","name":"TempThresh","property":
"payload","propertyType":"msg","rules":[{"t":"gte","v":"35","vt":"num"}],"chec
kall":"true","repair":false,"outputs":1,"x":510,"y":500,"wires":[["42416ae236f
9cf82"]]},{"id":"3b7994850ba3ac45","type":"delay","z":"dce59bb1b197e846","name
":"","pauseType":"rate","timeout":"5","timeoutUnits":"seconds","rate":"2","nbR
ateUnits":"1","rateUnits":"hour","randomFirst":"1","randomLast":"5","randomUni
ts":"seconds","drop":false,"allowrate":false,"outputs":1,"x":770,"y":580,"wire
s":[["036d350250483fe0"]]},{"id":"ed93af9b46faef11","type":"switch","z":"dce59
bb1b197e846","name":"pHThresh","property":"payload","propertyType":"msg","rule
s":[{"t":"btwn","v":"6.5","vt":"num","v2":"8.5","v2t":"num"}],"checkall":"true
","repair":false,"outputs":1,"x":500,"y":560,"wires":[["42416ae236f9cf82"]]},{
"id":"c1040d0c48eab14f","type":"switch","z":"dce59bb1b197e846","name":"TurbThr
esh","property":"payload","propertyType":"msg","rules":[{"t":"gt","v":"1","vt"
:"str"}],"checkall":"true","repair":false,"outputs":1,"x":530,"y":620,"wires":
[["42416ae236f9cf82"]]},{"id":"de2fcf253056e1de","type":"ui_chart","z":"dce59b
b1b197e846","name":"","group":"d633df785d60f78c","order":6,"width":7,"height":
5,"label":"pHchart","chartType":"line","legend":"false","xformat":"HH:mm:ss","
interpolate":"linear","nodata":"","dot":false,"ymin":"","ymax":"","removeOlder
":1,"removeOlderPoints":"","removeOlderUnit":"3600","cutout":0,"useOneColor":f
alse,"useUTC":false,"colors":["#1f77b4","#aec7e8","#ff7f0e","#2ca02c","#98df8a
","#d62728","#ff9896","#9467bd","#c5b0d5"],"outputs":1,"useDifferentColor":fal
se,"className":"","x":680,"y":300,"wires":[[]]},{"id":"6b4fcbd80a06314c","type
":"ui_chart","z":"dce59bb1b197e846","name":"","group":"d633df785d60f78c","orde
r":1,"width":0,"height":0,"label":"turbChart","chartType":"line","legend":"fal
se","xformat":"HH:mm:ss","interpolate":"linear","nodata":"","dot":false,"ymin"
:"","ymax":"","removeOlder":1,"removeOlderPoints":"","removeOlderUnit":"3600",
"cutout":0,"useOneColor":false,"useUTC":false,"colors":["#1f77b4","#aec7e8","#
ff7f0e","#2ca02c","#98df8a","#d62728","#ff9896","#9467bd","#c5b0d5"],"outputs"
:1,"useDifferentColor":false,"className":"","x":620,"y":400,"wires":[[]]},{"id
```

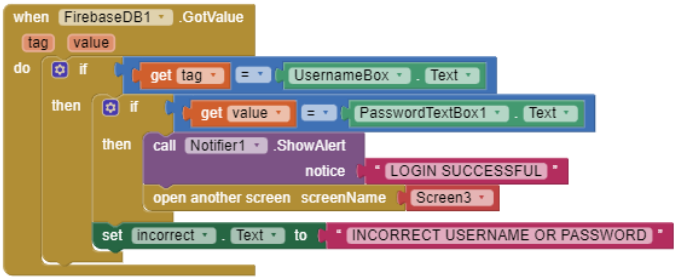
```
":"1a403e1397246142","type":"ui_template","z":"dce59bb1b197e846","group":"bef2
c6e7f6eac166","name":"","order":6,"width":0,"height":0,"format":"<style
id=\"blue-dashboard\">\n\nbody.nr-dashboard-theme {\n    background: radial-
gradient(circle, rgb(45 97 186) 0%, rgb(16 16 146) 40%, rgba(2,0,36,1)
100%);\n}\nbody.nr-dashboard-theme md-toolbar {\n   background: radial-
gradient(circle, rgb(45 97 186) 0%, rgb(16 16 146) 40%, rgba(2,0,36,1)
100%);\n    color: #fff;\n}\nbody.nr-dashboard-theme md-sidenav {\n    color:
rgb(215 223 245);\n    background: radial-gradient(circle, rgb(45 97 186) 0%,
rgb(16 16 146) 40%, rgba(2,0,36,1) 100%);\n}\nbody.nr-dashboard-theme md-
sidenav div.md-list-item-inner {\n    color: rgb(215 223
245);\n    background-color: transparent;\n}\n.nr-dashboard-theme ui-card-
panel {\n    background: linear-gradient(135deg, rgba(255,255,255,0.1) 0%,
rgba(255,255,255,0.05) 100%);\n    border: none;\n    border-radius:
1em;\n    box-shadow: 0 0 1em 0em rgb(0 0 100 / 50%)\n}\nbody.nr-dashboard-
theme md-content md-card {\n    background:transparent;\n    color: rgb(215
223 245);\n}\n.nr-dashboard-theme .nr-dashboard-template
{\n    background:transparent;\n}\n.nr-dashboard-theme ui-card-panel p.nr-
dashboard-cardtitle {\n    color: #6a99f1;\n    text-shadow: 1px 1px 0px rgb(0
0 0 / 50%);\n}\n.md-button{\n    border-radius:0.6em;\n    color:rgb(215 223
245);\n}\n.nr-dashboard-theme .nr-dashboard-button .md-button
{\n    background-color: #053ea8;\n    color:rgb(215 223 245);\n}\n.nr-
dashboard-theme .nr-dashboard-button .md-button:hover {\n    background-color:
#0646bd;\n}\n.nr-dashboard-theme .nr-dashboard-button .md-button:focus
{\n    background-color: #0646bd;\n}\n.nr-dashboard-theme .nr-dashboard-
numeric .value {\n    background:unset;\n    color:rgb(215 223 245);\n}\n.nr-
dashboard-theme .nr-dashboard-form {\n    color:rgb(215 223 245);\n}\n.nr-
dashboard-theme .nr-dashboard-form-button {\n    background-color:
#053ea8;\n    color:rgb(215 223 245);\n}\n\n.nr-dashboard-theme .nr-dashboard-
form-button:hover {\n    background-color: #0646bd !important;\n}\n.nr-
dashboard-theme .nr-dashboard-form-button:focus {\n    background-color:
#0646bd !important;\n}\n.nr-dashboard-theme md-input-container.md-default-
theme .md-input, md-input-container .md-input {\n    color:rgb(215 223
245);\n    border-color: #074fd4 !important;\n}\n.nr-dashboard-theme .nr-
dashboard-switch md-switch .md-thumb {\n    background-color: rgb(25 31
153);\n}\n.nr-dashboard-theme .nr-dashboard-switch md-switch .md-bar
{\n    background-color: rgb(94 94 177);\n    box-shadow: inset 0 0 1px
#0000005e;\n}\n.nr-dashboard-theme .nr-dashboard-switch md-switch.md-checked
.md-bar {\n    background-color: rgb(94 94 177);\n    box-shadow: 0 0 10px
0px #101086;\n}\n.nr-dashboard-theme .nr-dashboard-switch md-switch.md-checked
.md-thumb {\n    background-color: rgb(61 107 232);\n    box-shadow: 0 0 10px
0px #101086;\n}\n.nr-dashboard-theme .nr-dashboard-slider .md-track
{\n    background-color: rgb(125 172 249 / 35%);\n}\nmd-slider .md-thumb
{\n    box-shadow: 0 0 10px 0px #101086;\n}\n\n.nr-dashboard-theme .nr-
dashboard-slider .md-thumb:after {\n    background-color:
#3d6be8;\n    border-color: #3d6be8;\n}\n\n.nr-dashboard-theme md-select-menu,
.nr-dashboard-theme md-select-menu md-option {\n    background-color: rgb(0 0
0 / 12%);\n    color:rgb(215 223 245);\n    border-radius:0.8em;\n}\n\n.nr-
dashboard-theme .nr-dashboard-dropdown md-select .md-select-value, .nr-
```

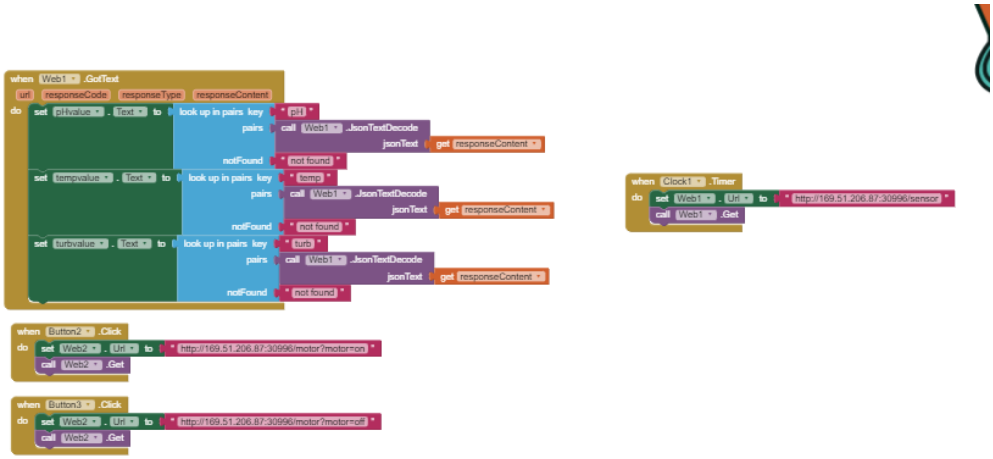
```
dashboard-theme .nr-dashboard-dropdown md-select .md-select-value.md-select-placeholder {\n    margin-top: -3px;\n    border-color: #eeeeee7a;\n    \n}\n.nr-dashboard-theme .md-select-menu-container {\n    max-height: 85%;\n    overflow-y: auto;\n    border: none;\n    border-radius: 0.8em;\n}\n.nr-dashboard-theme .nr-dashboard-dropdown .md-select-icon {\n\n}\n.nr-dashboard-theme md-select-menu md-option {\n    background-color: rgb(40 53 147);\n    \n    height: 29px;\n    border-radius: 0.6em;\n    margin-left: 10px;\n    margin-right: 10px;\n    margin-top: 2px;\n    box-shadow: 0 0 6px 6px #24202133;\n    transition: 0.3s;\n}\n.nr-dashboard-theme md-select-menu md-option[selected] {\n    color: rgb(130 177 255) !important;\n    background-color: rgb(27 90 197) !important;\n}\n.nr-dashboard-theme md-select-menu md-option:nth-child(even) {\n    background-color: rgb(31 31 152);\n}\n.nr-dashboard-theme md-select-menu md-option:last-child {\n    margin-bottom: 8px;\n}\n.nr-dashboard-theme md-select-menu md-option:hover {\n    background-color: rgb(20 91 211) !important;\n    padding-left: 26px;\n}\n.nr-dashboard-theme md-select-menu md-option > .md-ripple-container{\n    border-radius: 14px;\n}\n\n</style>\n    ","storeOutMessages":true,"fwdInMessages":true,"resendOnRefresh":true,"templateScope":"local","className":"","x":900,"y":140,"wires":[[]]},{"id":"dff3497162207c84","type":"ui_gauge","z":"dce59bb1b197e846","name":"","group":"bef2c6e7f6eac166","order":5,"width":0,"height":0,"gtype":"gage","title":"pH","label":"units","format":"{{value}}","min":0,"max":"14","colors":["#00b500","#e6e600","#ca3838"],"seg1":"","seg2":"","className":"","x":670,"y":340,"wires":[]},{"id":"8f96553a86aabad7","type":"function","z":"dce59bb1b197e846","name":"motorcmd","func":"msg.payload=msg.payload.motor\nglobal.set(\"cmd\",msg.payload)\nreturn msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":280,"y":460,"wires":[[]]},{"id":"c1ec7433c4bc61b5","type":"http in","z":"dce59bb1b197e846","name":"Motor","url":"/motor?motor","method":"get","upload":false,"swaggerDoc":"","x":130,"y":780,"wires":[["d5ab7bd9ac57d21e","3a93916515500144"]]},{"id":"d5ab7bd9ac57d21e","type":"ibmiot out","z":"dce59bb1b197e846","authentication":"apiKey","apiKey":"65e8b07c8b314e54","outputType":"evt","deviceId":"GokulEsp32","deviceType":"ESP32","eventCommandType":"motor","format":"json","data":"motorcmd","qos":0,"name":"IBM IoT","service":"registered","x":480,"y":740,"wires":[]},{"id":"3a93916515500144","type":"http response","z":"dce59bb1b197e846","name":"","statusCode":"","headers":{},"x":430,"y":800,"wires":[]},{"id":"7df200000683703e","type":"ui_spacer","z":"dce59bb1b197e846","name":"spacer","group":"bef2c6e7f6eac166","order":2,"width":7,"height":1},{"id":"8ac15f63a1aa14d2","type":"ui_spacer","z":"dce59bb1b197e846","name":"spacer","group":"bef2c6e7f6eac166","order":3,"width":7,"height":1},{"id":"21a9c573c93b4dbe","type":"ui_spacer","z":"dce59bb1b197e846","name":"spacer","group":"d633df785d60f78c","order":2,"width":7,"height":1},{"id":"7094b7e9e0cdda16","type":"ui_spacer","z":"dce59bb1b197e846","name":"spacer","group":"d633df785d60f78c","order":3,"width":7,"height":1},{"id":"2d4dc0206f9d0654","type":"ui_spacer","z":"dce59bb1b197e846","name":"spacer","group":"d633df785d60f78c","order":5,"width":7,"height":1},{"id":"65e8b07c8b314e54","type":"ibmiot","name":"IBMIotAPIKey","keepalive":"60","serverName":"","cleansession":true,"appId":""
```

```
,"shared":false},{"id":"bef2c6e7f6eac166","type":"ui_group","name":"FirstLayout","tab":"338664aeed40a1fc","order":2,"disp":true,"width":"7","collapse":false,"className":""},{"id":"d633df785d60f78c","type":"ui_group","name":"Chart","tab":"338664aeed40a1fc","order":3,"disp":true,"width":7,"collapse":false,"className":""},{"id":"338664aeed40a1fc","type":"ui_tab","name":"Home","icon":"dashboard","disabled":false,"hidden":false}]
```


Mobile App backend:







Github Link:

https://github.com/IBM-EPBL/IBM-Project-850-1658326108

Project demo link:

https://youtu.be/DhfnLT3Jlh0