Date	**18 November 2022**
Team ID	**PNT2022TMID51401**
Project Name	**VirtualEye-life guard for swimming pools to detect active drowning**

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

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. INTRODUCTION

1.1 PROJECT OVERVIEW

Swimming is one of the best exercises that helps people to reduce stress in this urban lifestyle. Swimming pools are found larger in number in hotels, and weekend tourist spots and barely people have them in their house backyard. Beginners, especially, often feel it difficult to breathe underwater which causes breathing trouble which in turn causes a drowning accident. Worldwide, drowning produces a higher rate of mortality without causing injury to children. Children under six of their age are found to be suffering the highest drowning mortality rates worldwide. Such kinds of deaths account for the third cause of unplanned death globally, with about 1.2 million cases yearly. To overcome this conflict, a meticulous system is to be implemented along the swimming pools to save human life.

1.2 PURPOSE

Internet is basically system of interconnected computers through network. But now its use is changing with changing world and it is not just confined to emails or web browsing. Today's internet also deals with embedded sensors and has led to development of smart homes, smart rural area, e-health care's etc. and this introduced the concept of IoT . Internet of Things refers to interconnection or communication between two or more devices without humanto-human and human-to-computer interaction. Connected devices are equipped with sensors or actuators perceive their surroundings. IOT has four major components which include sensing the device, accessing the device, processing the information of the device, and provides application and services. In addition to this it also provides security and privacy of data . Automation has affected every aspect of our daily lives. More improvements are being introduced in almost all fields to reduce human effort and save time. Thinking of the same is trying to introduce automation in the field of testing. vistualEye is an integral part of any company's asset base, since it provides them with the necessary business functionality. Problems that occur due to problems in water need to be overcome. The latest method used by the Real time river water is which requires a lot of manpower and is time-consuming

LITERATURE SURVEY

2. LITERATURE SURVEY

2.1 EXISTING SYSTEM

One important environment that the need for monitoring systems is crucially sensed is the swimming pool. Each year many people including children are drowned or very close to drowning in the deeps of the swimming pools, and the life guards are not trained well enough to handle these problems . This raises the need for having a system that will automatically detect the drowning person and alarm the lifeguards of such danger. Real-time detection of a drowning person in swimming pools is a challenging task that requires an accurate system. The challenge is due to the presence of water.

2.2 REFERENCES

[1] Foresti, Gian Luca, Petri Mähönen, and Carlo S. Regazzoni, eds. Multimedia video-based surveillance systems: Requirements, Issues and Solutions. Vol. 573. Springer Science & Business Media, 2012.

[2] Jones, Graeme A., Nikos Paragios, and Carlo S. Regazzoni, eds. Video-based surveillance systems: computer vision and distributed processing. Springer Science & Business Media, 2012.

[3] Conde, Cristina, et al. "HoGG: Gabor and HoG-based human detection for surveillance in non-controlled environments." Neurocomputing 100 (2013): 19-30.

[4] Wang, Xiaogang. "Intelligent multi-camera video surveillance: A review." Pattern recognition letters 34.1 (2013): 3-19.

[5] Gudyś, Adam, et al. "Tracking people in video sequences by clustering feature motion paths." Computer Vision and Graphics. Springer International Publishing, 2014. 236-245.

[6] Vezzani, Roberto, Davide Baltieri, and Rita Cucchiara. "People reidentification in surveillance and forensics: A survey." ACM Computing Surveys (CSUR) 46.2

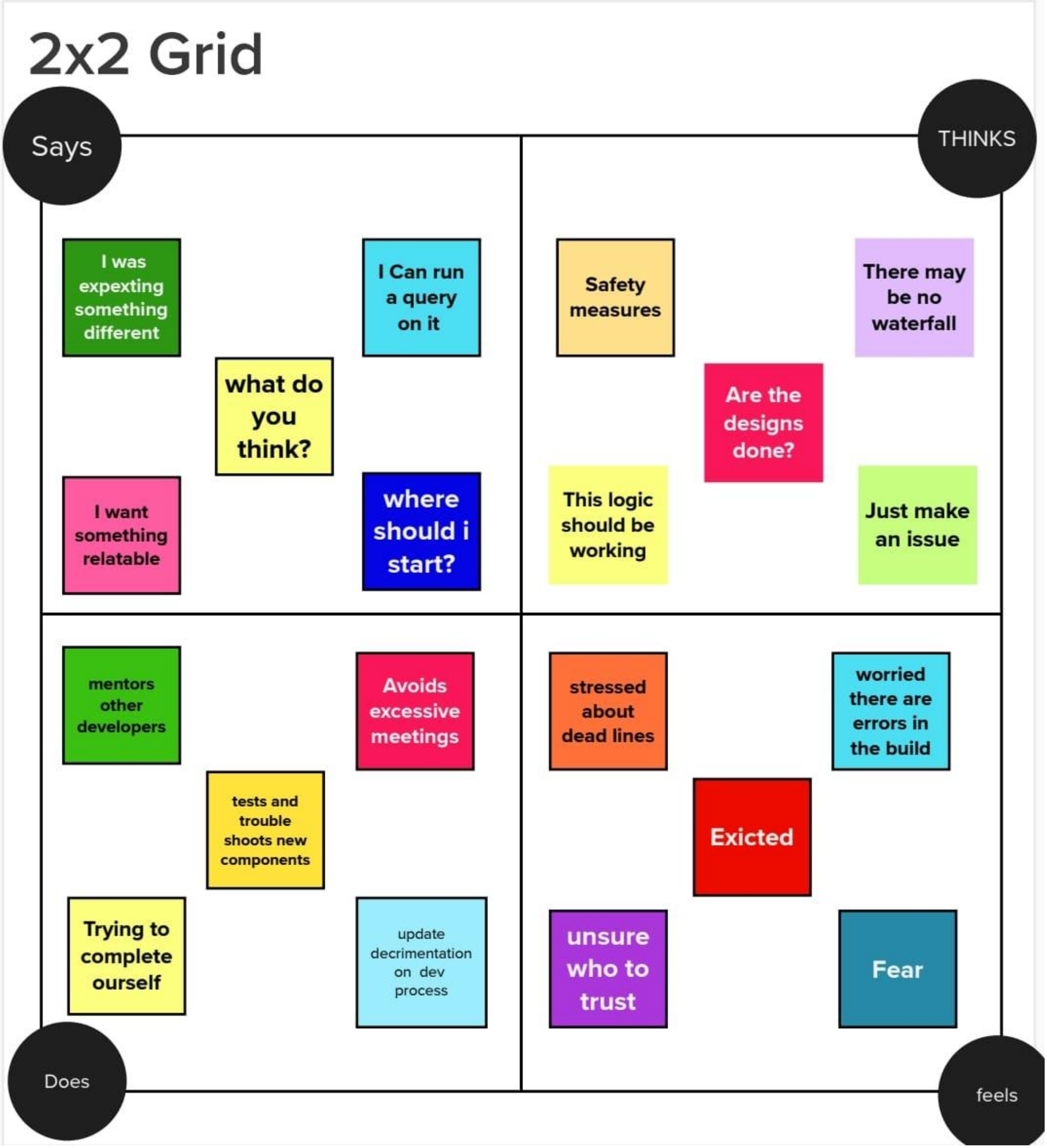
2.3 PROBLEM STATEMENT DEFINITION

Our system is based on real time video analysis of the cameras installed around the swimming pool in a way which the entire swimming pool can be covered. Each camera is mounted on pool walls oriented downwards with a sharp angle, so that it can minimize the effect of lightening system which causes occlusions and foreshadowing. In this work, a ODROID-XU as a distributed system is installed in the swimming pool to collect all the video signals collected from cameras and process them using computer vision methods. The used hardware including the distributing system known as ODROID-XU, and our Logitech HD.

IDEATION AND PROPOSED SOLUTION

3. IDEATION AND PROPOSED SOLUTON

3.1 EMPATHY MAP CANVAS



2x2 Grid

Says

I was expexting something different

I Can run a query on it

what do you think?

I want something relatable

where should i start?

THINKS

Safety measures

There may be no waterfall

Are the designs done?

This logic should be working

Just make an issue

Does

mentors other developers

Avoids excessive meetings

tests and trouble shoots new components

Trying to complete ourself

update decrimentation on dev process

feels

stressed about dead lines

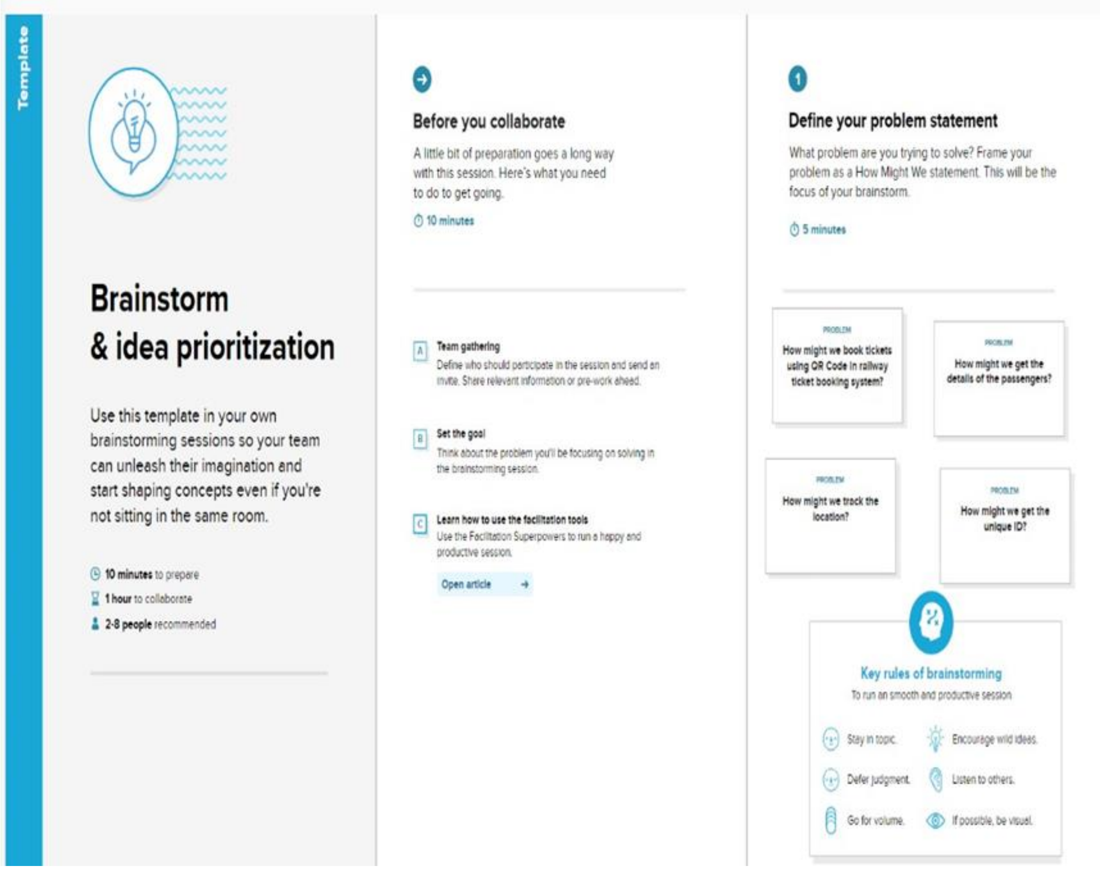
worried there are errors in the build

Exicted

unsure who to trust

Fear

3.2 IDEATION & BRAINSTORMING



Template

Brainstorm & idea prioritization

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.

- **10 minutes** to prepare
- **1 hour** to collaborate
- **2-8 people** recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

1 Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM
How might we book tickets using QR Code in railway ticket booking system?

PROBLEM
How might we get the details of the passengers?

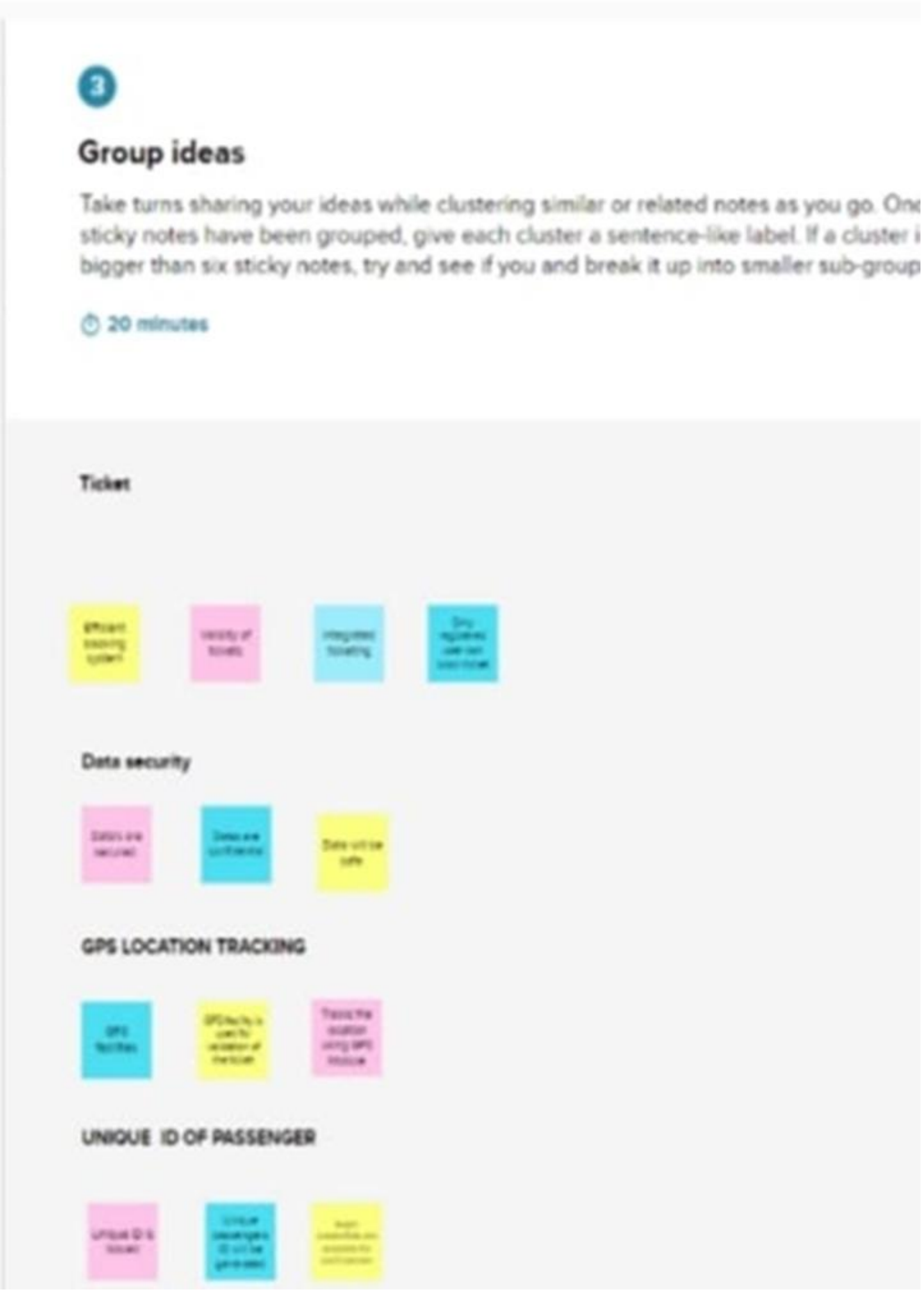
PROBLEM
How might we track the location?

PROBLEM
How might we get the unique ID?

Key rules of brainstorming

To run an smooth and productive session

- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, be visual.



3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Onc
sticky notes have been grouped, give each cluster a sentence-like label. If a cluster i
bigger than six sticky notes, try and see if you and break it up into smaller sub-group

20 minutes

Ticket

Efficient booking system

Validity of tickets

Integrated ticketing

Only registered user can book ticket

Data security

Data's are secured

Datas are confidential

Data will be safe

GPS LOCATION TRACKING

GPS facilities

GPS facility is used for validation of the ticket

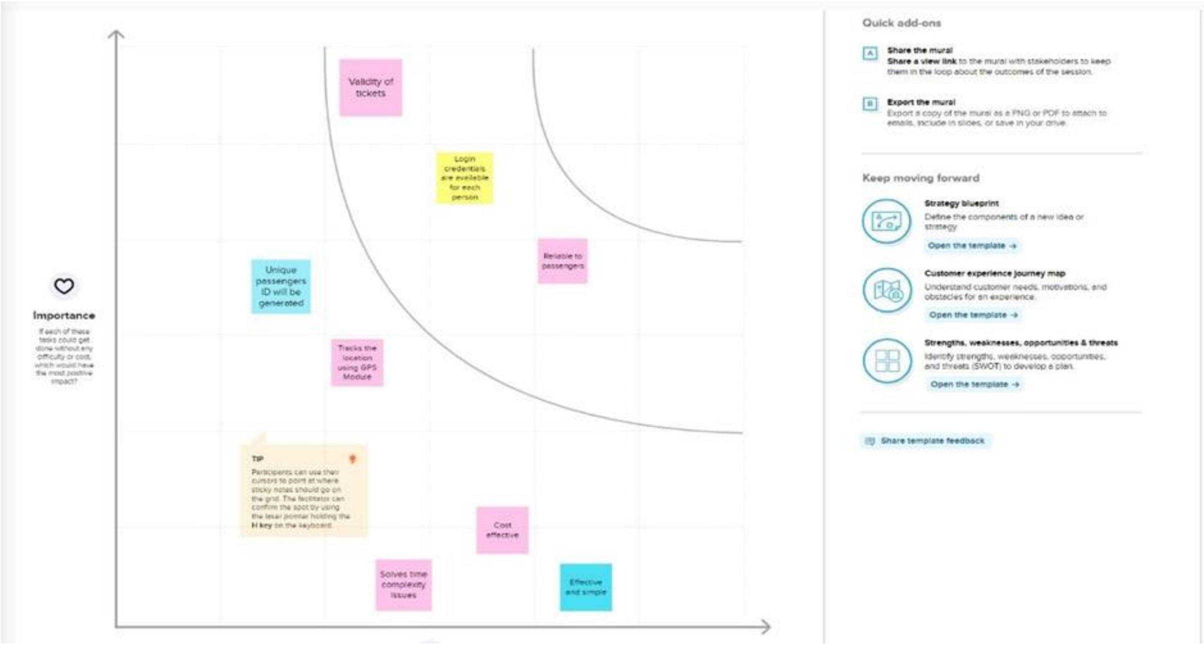
Tracks the location using GPS module

UNIQUE ID OF PASSENGER

Unique ID is issued

Unique passengers ID will be generated

[illegible]


Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?
Validity of tickets
Login credentials are available for each person
Reliable to passengers
Unique passengers ID will be generated
Tracks the location using GPS Module
Cost effective
Solves time complexity issues
Effective and simple

TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.

Quick add-ons

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

B **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

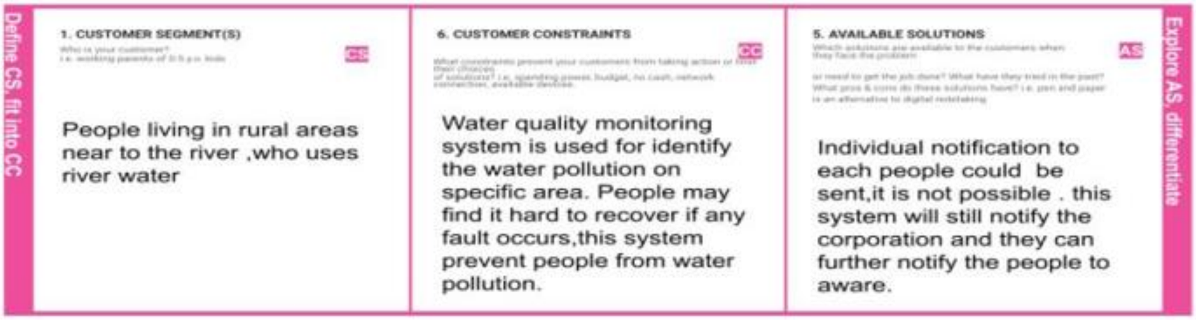
3.3 PROPOSED SOLUTION

S.NO	PARAMETERS	DESCRIPTIONS
1	Problem Statement (Problem to be solved)	Such kind of deaths account for the thied cause of unplanned death globally,withabout 1.2 million case yearly.To overcome this conflict,a meticulous system is to be implemented along the swimming pools to save human life

2	Idea / Solution description	Pulse oximeters make it possible to detect a drop in oxygen saturation,which may indicate the loss of consciousness under the water and the initial stage of drowning.the addition of a small optical sensor to commonly used pool wristbands is a convenient solution both for swimming pool visitors and staff.however,no such devices available on the market can be treated as a “gold standard”.
3	Novelty / Uniqueness	It helps the life guard to detect thr underwater situvatiob where they cant easily observe.
4.	Social Impact / Customer Satisfaction	Expected to perform rescues to prevent drownings and to provide immediate first aid and CPR.
5	Business Model (Revenue Model)	Almost every one earns 2,360EUR or less in spain.

6	Scalability of the Solution	1.local health aware problem. 2.local health aware suspicion. 3.buddy system.

3.4 PROBLEM SOLUTION FIT



Define CS, fit into CC

1. CUSTOMER SEGMENT(S)

CS

Who is your customer?
i.e. working parents of 0-5 y.o. kids

People living in rural areas near to the river ,who uses river water

6. CUSTOMER CONSTRAINTS

CC

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

Water quality monitoring system is used for identify the water pollution on specific area. People may find it hard to recover if any fault occurs,this system prevent people from water pollution.

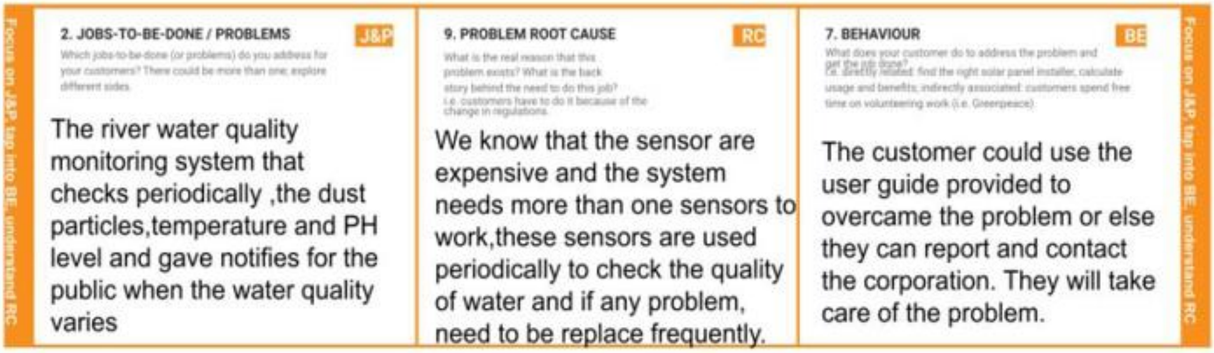
5. AVAILABLE SOLUTIONS

AS

Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking

Individual notification to each people could be sent,it is not possible . this system will still notify the corporation and they can further notify the people to aware.

Explore AS, differentiate



Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS

J&P

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

The river water quality monitoring system that checks periodically ,the dust particles,temperature and PH level and gave notifies for the public when the water quality varies

9. PROBLEM ROOT CAUSE

RC

What is the real reason that this problem exists? What is the back story behind the need to do this job?
i.e. customers have to do it because of the change in regulations.

We know that the sensor are expensive and the system needs more than one sensors to work,these sensors are used periodically to check the quality of water and if any problem, need to be replace frequently.

7. BEHAVIOUR

BE

What does your customer do to address the problem and get the job done?
i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

The customer could use the user guide provided to overcame the problem or else they can report and contact the corporation. They will take care of the problem.

Focus on J&P, tap into BE, understand RC

REQUIREMENT ANALYSIS

4. REQUIREMENT ANALYSIS

4.1. FUNCTIONAL REQUIREMENTS

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Unique	• VitualEye by using IoT intergrated big data Analytics.
FR-2	options	Only a few water quality parametes.

FR-3	Mandatory fields	□ System should only allow users to move to payment only when mandatory fields such as date, time, location has been mentioned
FR-4	Synchronization	□ System should consider timezone synchronisation when accepting from different timezones
FR-5	Authentication	□ confirmation should be sent to user to the specified contact details

4.2. NON-FUNCTIONAL REQUIREMENTS

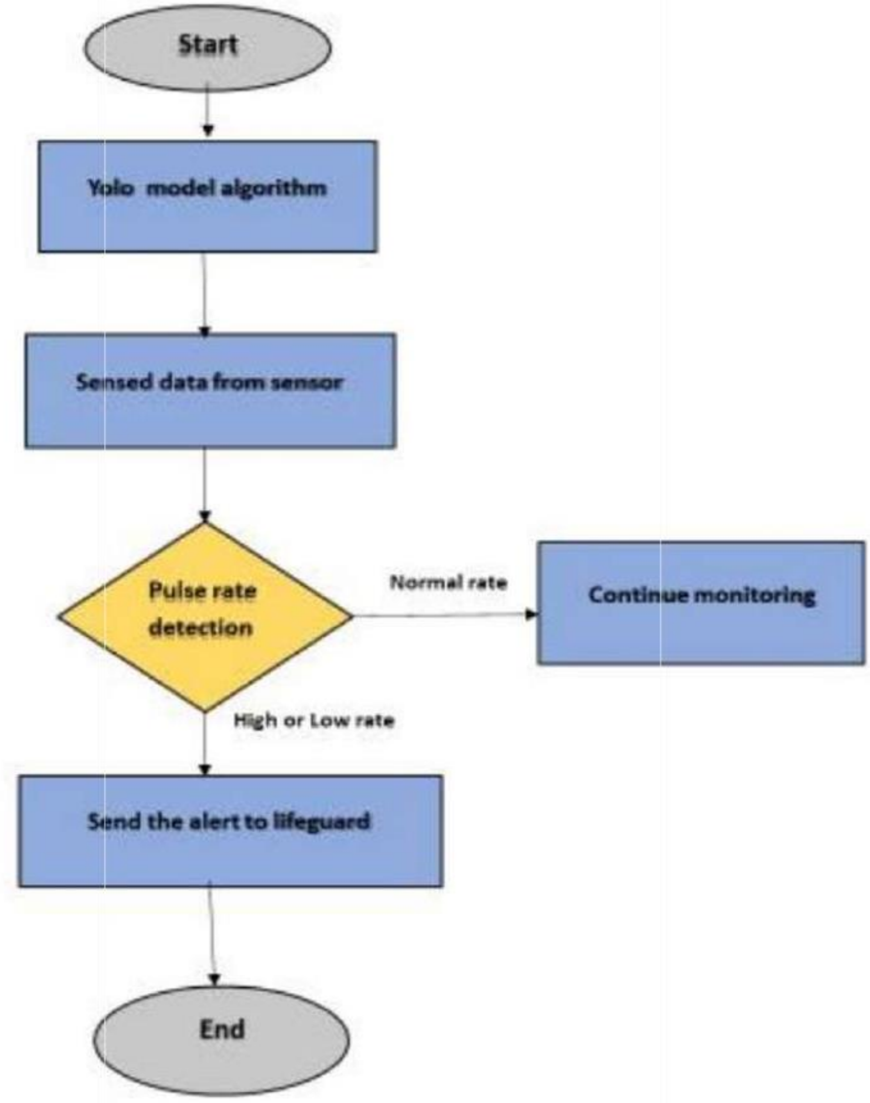
FR No.	**Non-Functional Requirement**	**Description**
NFR-1	**Usability**	□ Search results should populate within acceptable time limits
NFR-2	**Security**	□ System should visually confirm as well as send confirmation to the user's contact

NFR-3	**Reliability**	□ System should accept payments via different payment methods, like PayPal, wallets, cards, vouchers, etc
NFR-4	**Performance**	□ Search results should populate within acceptable time limits
NFR-5	**Availability**	□ User should be helped appropriately to fill in the mandatory fields, incase of invalid input
NFR-6	**Scalability**	□ Use of captcha and encryption to avoid bots

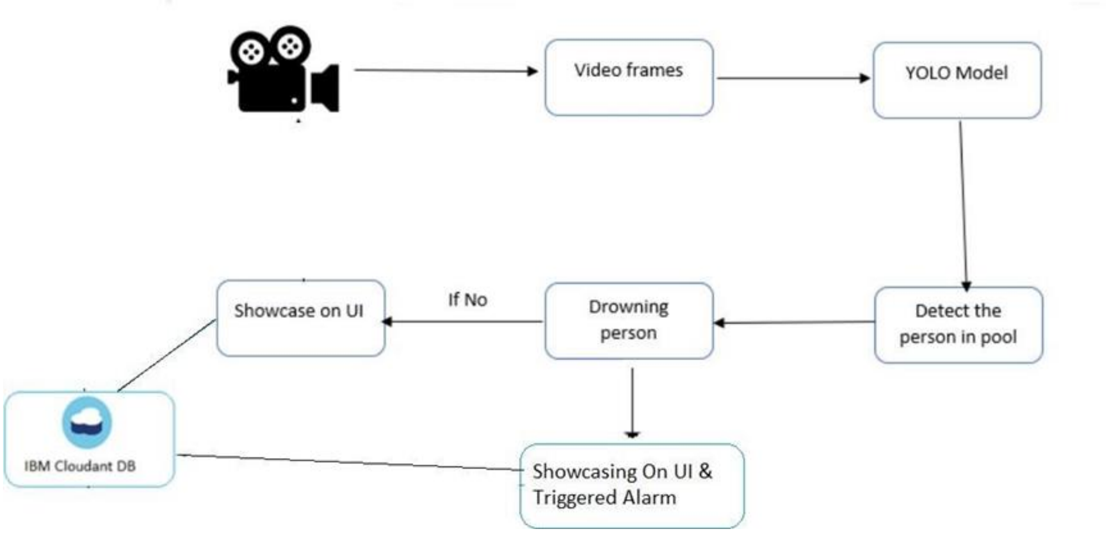
PROJECT DESIGN

5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS



5.2 SOLUTION & TECHNICAL ARCHITECTURE



5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user, Web user)	Registration	USN-1	As a user, I can register through the form by Filling in my details	I can register and create my account / dashboard	High	Sprint-1
		USN-2	As a user, I can register through phone numbers, Gmail, Facebook or other social sites	I can register & create my dashboard with Facebook login or other social sites	High	Sprint-2
	Conformation	USN-3	As a user, I will receive confirmation through email or OTP once registration is successful	I can receive confirmation email & click confirm.	High	Sprint-1

	Authentication/Login	USN-4	As a user, I can login via login id and password or through OTP received on register phone number	I can login and access my account/dashboard	High	Sprint-1
	Display Train details	USN-5	As a user, I can enter the start and destination to get the list of trains available connecting the above	I can view the train details (name & number), corresponding routes it passes through based on the start and destination entered.	High	Sprint-1
	Booking	USN-6	As a use, I can provide the basic details such as a name, age, gender etc…	I will view, modify or confirm the details enter.	High	Sprint-1
		USN-7	As a user, I can choose the class, seat/berth. If a preferred seat/berth isn't available I can be allocated based on the availability.	I will view, modify or confirm the seat/class berth selected	High	Sprint-1
	Payment	USN-8	As a user, I can choose to pay through credit Card/debit card/UPI.	I can view the payment Options available and select my desirable choice To proceed with the payment	High	Sprint-1
		USN-9	As a user, I will be redirected to the selected Payment gateway and upon successful	I can pay through the payment portal and confirm the booking if any changes need to	High	Sprint-1
User Type	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Acceptance criteria**	**Priority**	**Release**
			completion of payment I'll be redirected to the booking website.	be done I can move back to the initial payment page		
	Ticket generation	USN-10	As a user, I can download the generated e-ticket for my journey along with the QR code which is used for authentication during my journey.	I can show the generated QR code so that authentication can be done quickly.	High	Sprint-1

	Ticket status	USN-11	As a user, I can see the status of my ticket Whether it’s confirmed/waiting/RAC.	I can confidentially get the Information and arrange alternate transport if the ticket isn’t Confirmed	High	Sprint-1
	Remainders notification	USN-12	As a user, I get remainders about my journey A day before my actual journey.	I can make sure that I don’t miss the journey because of the constant notifications.	Medium	Sprint-2
		USN-13	As a user, I can track the train using GPS and can get information such as ETA, Current stop and delay.	I can track the train and get to know about the delays pian accordingly	Medium	Sprint-2
	Ticket cancellation	USN-14	As a user, I can cancel my tickets if there’s any Change of plan	I can cancel the ticket and get a refund based on how close the date is to the journey.	High	Sprint-1
	Raise queries	USN-15	As a user, I can raise queries through the query box or via mail.	I can view my pervious queries.	Low	Sprint-2
Customer care Executive	Answer the queries	USN-16	As a user, I will answer the questions/doubts Raised by the customers.	I can view the queries and make it once resolved	Medium	Sprint-2
Administrator	Feed details	USN-17	As a user, I will feed information about the trains delays and add extra seats if a new compartment is added.	I can view and ensure the corrections of the information fed.	High	Sprint-1

PROJECT PLANNING AND SCHEDULING

6. PROJECT PLANNING AND SCHEDULING

6.1. 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 through the form by Filling in my details	2	High	Keerthika
Sprint-1		USN-2	As a user, I can register through phone numbers, Gmail, Facebook or other social sites	1	High	Pandiselvi
Sprint-1	Conformation	USN-3	As a user, I will receive confirmation through email or OTP once registration is successful	2	Low	Buvaneshwari
Sprint-1	login	USN-4	As a user, I can login via login id and password or through OTP received on register phone number	2	Medium	Viji
Sprint-1	Display Train details	USN-5	As a user, I can enter the start and destination to get the list of trains available connecting the above	1	High	Priya
Sprint-2	data	USN-6	As a use, I can provide the basic details such as a name, age, gender etc…	2	High	Keerthika

Sprint-2	Water quality	USN-7	As a user, I can choose the water good or bad	1	Low	Pandiselvi
Sprint-2	Payment	USN-8	As a user, I can choose to pay through credit Card/debit card/UPI.	1	High	Viji
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2		USN-9	As a user, I will be redirected to the selected	2	High	
Sprint-3	generation	USN-10	As a user, I can download the generated e- ticket for my journey along with the QR code which is used for authentication during my journey.	1	High	Pandiselvi
Sprint-3	status	USN-11	As a user, I can see the status of temparature	2	High	Viji
			Whether it's confirmed/waiting/RAC.			
Sprint-3	Remainders notification	USN-12	As a user, I get remainders about my journey A day before my actual journey.	1	High	Buvaneshwari
Sprint-3	cancellation	USN-13	As a user, I can track the temperature using GPS and can get information such as ETA, Current stop and delay	2	High	Keerthika
Sprint-4		USN-14	As a user, water real time quality monitoring	1	High	Priya
Sprint-4	Raise queries	USN-15	As a user, I can raise queries through the query box or via mail.	2	Medium	Pandiselvi
Sprint-4	Answer the queries	USN-16	As a user, I will answer the questions/doubts Raised by the customers.	2	High	Buvaneshwari
Sprint-4	Feed details	USN-17	As a user, I will feed information about the trains delays and add extra seats if a new compartment is added.	1	High	Keerthika

6.2. SPRINT DELIVERY SCHEDULE

Sprint	**Total Story Points**	**Duration**	**Sprint Start Date**	**Sprint End Date (Planned)**	**Story Points Completed (as on Planned End Date)**	**Sprint Release Date (Actual)**
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	5 Nov 2022
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-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 Nov2022

CODING AND SOLUTIONING

7. CODING AND SOLUTIONING

7.1. FEATURE 1 ○

- IOT device
- IBM Watson platform
- Node red
- Cloudant DB
- Web UI
- Geofence □ MIT App
- Python code

7.2. FEATURE 2

- Registration
- Login
- Verification
- SMS
- Adding Queries

```
labl_0 = Label(base, text="Registration form",width=20,font=("bold",
20))
labl_0.place(x=90,y=53)
```

```
lb1= Label(base, text="Enter Name", width=10, font=("arial",12))
lb1.place(x=20, y=120)   en1= Entry(base)
en1.place(x=200, y=120)

lb3= Label(base, text="Enter Email", width=10, font=("arial",12))
lb3.place(x=19, y=160)   en3= Entry(base)
en3.place(x=200, y=160)

lb4= Label(base, text="Contact Number", width=13,font=("arial",12))
lb4.place(x=19, y=200)   en4= Entry(base)
en4.place(x=200, y=200)

lb5= Label(base, text="Select Gender", width=15, font=("arial",12))
lb5.place(x=5, y=240)
var = IntVar()

Radiobutton(base, text="Male", padx=5,variable=var,
value=1).place(x=180, y=240)

Radiobutton(base, text="Female", padx =10,variable=var,
value=2).place(x=240,y=240)

Radiobutton(base, text="others", padx=15, variable=var,
value=3).place(x=310,y=240)

list_of_cntry = ("United States", "India", "Nepal", "Germany")   cv
= StringVar()   drplist= OptionMenu(base, cv, *list_of_cntry)
drplist.config(width=15)   cv.set("United States")   lb2= Label(base,
text="Select Country", width=13,font=("arial",12))
lb2.place(x=14,y=280)
```

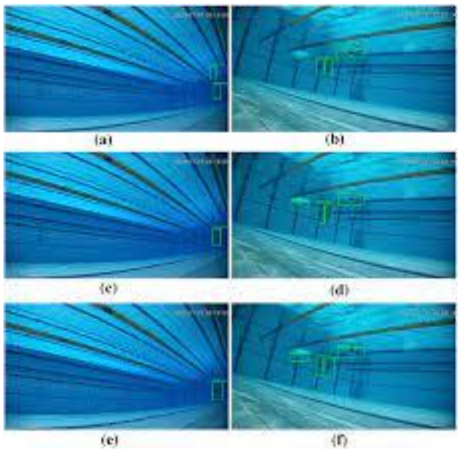
```
drplist.place(x=200, y=275)

lb6= Label(base, text="Enter Password", width=13,font=("arial",12))
lb6.place(x=19, y=320)   en6= Entry(base, show='*')
en6.place(x=200, y=320)

lb7= Label(base, text="Re-Enter Password",
width=15,font=("arial",12))
lb7.place(x=21, y=360)   en7
=Entry(base, show='*')
en7.place(x=200, y=360)

Button(base, text="Register", width=10).place(x=200,y=400)
base.mainloop()




def generateOTP() :

   # Declare a digits variable
# which stores all digits
digits = "0123456789"
   OTP = ""

  # length of password can be changed
# by changing value in range     for i
in range(4) :
      OTP += digits[math.floor(random.random() * 10)]
```

```
    return OTP

# Driver code if __name__ ==
"__main__" :

    print("OTP of 4 digits:", generateOTP())

digits="0123456789" OTP=""
for i in range(6):
    OTP+=digits[math.floor(random.random()*10)]
otp = OTP + " is your OTP" msg= otp s =
smtplib.SMTP('smtp.gmail.com', 587)
s.starttls()
s.login("Your Gmail Account", "You app password") emailid
= input("Enter your email: ")
s.sendmail('&&&&&&&&&&&',emailid,msg)
a = input("Enter Your OTP >>: ") if a ==
OTP:
    print("Verified") else:
    print("Please Check your OTP again") roo
```

TESTING

8. TESTING

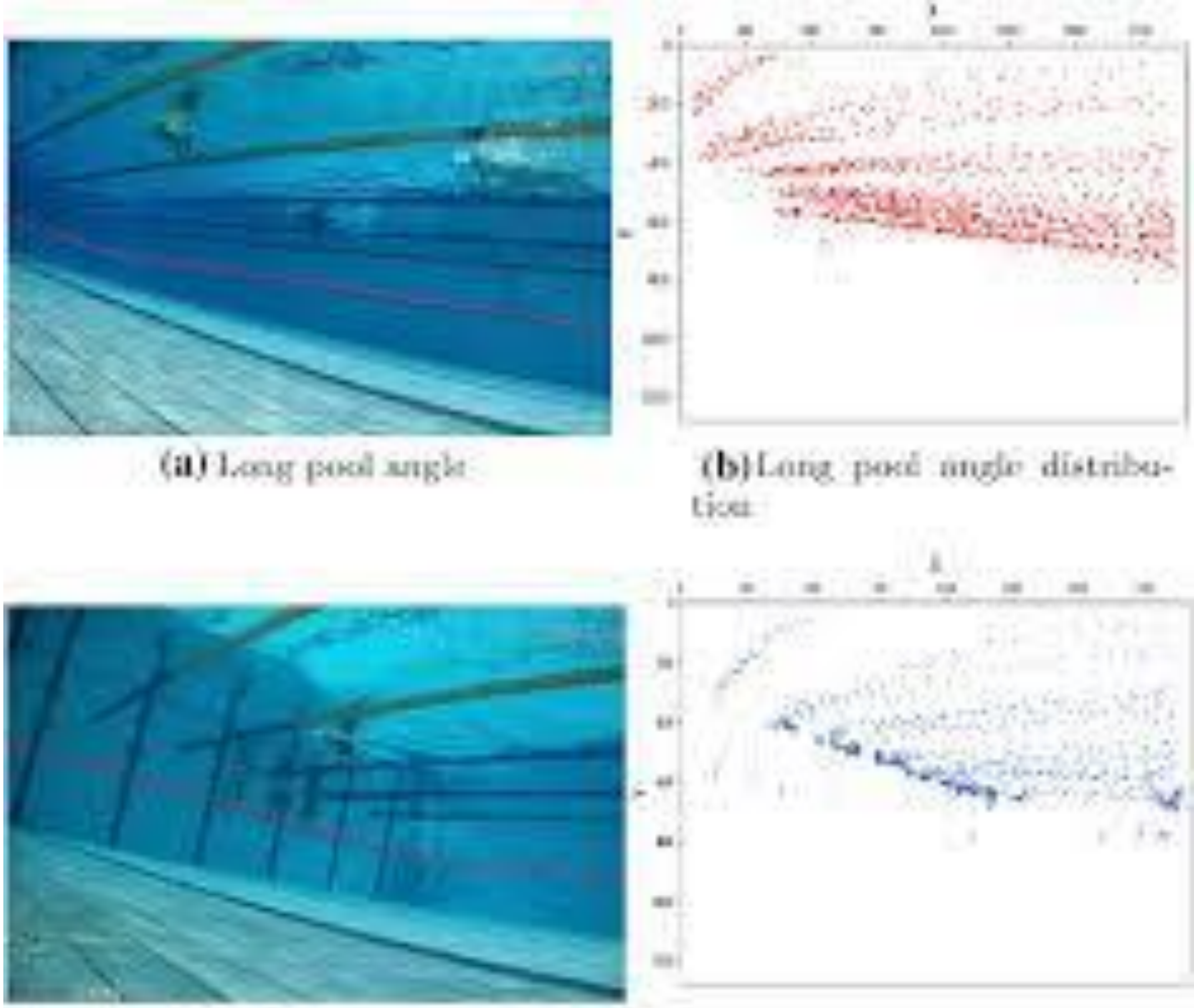
8.1.TEST CASES



RESULTS

9. RESULTS

9.1.PERFORMANCE METRICS



(a) Long pool angle

(b) Long pool angle distribution

(c) Short pool angle

(d) Short pool angle distribution

ADVANTAGES &DISADVANTAGES

10.ADVANTAGES &DISADVANTAGES

10.1.ADVANTAGES

It will reduce the time to measure the parameters .

This is economically affordable for common people.

10.2.DISADVANTAGES

- If any fault occurs in hardware devices then we are not able to monitor the quality of water.
- When quantity of water increase then each and every time we have change the system range.

CONCLUSION

11.CONCLUSION

we provided a method to robust human tracking and semantic event detection within the context of video surveillance system capable of automatically detecting drowning incidents in a swimming pool. In the current work, an effective background detection that incorporates prior knowledge using HSV color space and contour detection enables swimmers to be reliably detected and tracked despite the significant presence of water ripples. The system has been tested on several instances of simulated water conditions such as water reflection, lightening condition and false alarms. Our algorithm was able to detect all the drowning conditions along with the exact position of the drowning person in the swimming pool and had an average detection delay of 1.53 seconds, which is relatively low compared to the needed rescue time for a lifeguard operation. Our results show that the proposed method can be used as a reliable multimedia video-based surveillance system.

FUTURE SCOPE

12.FUTURE SCOPE

In future we use IoT Concept in this project.

Detecting the more parameters for most secure purpose

Increase the parameters by addition of multiple sensor.

By interfacing relay we controls the supply of water.

APPENDIX

13.APPENDIX

13.1.SOURCE PROGRAM

```
               #include <OneWire.h>
#include <DallasTemperature.h>
#include <SoftwareSerial.h>
#include <NewPing.h>
#define SensorPin A2            //pH meter Analog output to Arduino Analog Input 0
#define Offset 0.00           //deviation compensate
unsigned long int avgValue;     //Store the average value of the sensor feedback

#define TRIGGER_PIN  23  // Arduino pin tied to trigger pin on ping sensor.
#define ECHO_PIN     22  // Arduino pin tied to echo pin on ping sensor.
#define MAX_DISTANCE 200 // Maximum distance we want to ping for (in
centimeters). Maximum sensor distance is rated at 400-500cm.

NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE); // NewPing
setup of pins and maximum distance.
```

```
unsigned int pingSpeed = 50; // How frequently are we going to send out a ping (in
milliseconds). 50ms would be 20 times a second.
unsigned long pingTimer;     // Holds the next ping time.

// Data wire is plugged into pin 2 on the Arduino
#define ONE_WIRE_BUS 6

SoftwareSerial mySerial(7, 8);

// Setup a oneWire instance to communicate with any OneWire devices (not just
Maxim/Dallas temperature ICs)
OneWire oneWire(ONE_WIRE_BUS);

// Pass our oneWire reference to Dallas Temperature.
DallasTemperature sensors(&oneWire);

#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

//const int pingPin =22;
int sensorPin = A0;
int blueled = 13;
int redled = 24;
int greenled = 25;
int tempblueled = 32;
int tempredled = 33;
int tempgreenled = 34;
int levblueled = 35;
int levredled = 36;
int levgreenled = 37;
int turbblueled = 38;
int turbredled = 39;
int turbgreenled = 40;
int buzzer = 31;
float phValue;
float temperatureC;
long duration, cm;

void setup(void)
{
  // start serial port
  Serial.begin(9600);
  pingTimer = millis(); // Start now.
  // Start up the library
  sensors.begin(); // IC Default 9 bit. If you have troubles consider upping it 12. Ups the delay giving the IC more time to process the temperature measurement
  lcd.begin(16, 2); //initilise lcd with num of coloums 16 ,by row 2.
  lcd.clear(); //clears lcd just incase there is anytin been displayed
  pinMode(blueled, OUTPUT);
  pinMode(redled, OUTPUT);
  pinMode(greenled, OUTPUT);
  pinMode(tempblueled, OUTPUT);
  pinMode(tempredled, OUTPUT);
  pinMode(tempgreenled, OUTPUT);
  pinMode(levblueled, OUTPUT);
  pinMode(levredled, OUTPUT);
  pinMode(levgreenled, OUTPUT);
  pinMode(turbblueled, OUTPUT);
  pinMode(turbredled, OUTPUT);
  pinMode(turbgreenled, OUTPUT);
  pinMode(buzzer, OUTPUT);
    digitalWrite(buzzer, LOW);
  //initialization();
}

void loop() {
```

```
sensors.requestTemperatures(); // Send the command to get temperature
Serial.println(sensors.getTempCByIndex(0));
  int reading = analogRead(sensorPin);

// converting that reading to voltage,
float voltage = reading * 5.0;
voltage /= 1024.0;

// now print out the temperature
float temperatureC = (voltage - 0.5) * 100 ;
PH();
  digitalWrite(blueled, LOW);
digitalWrite(redled, LOW);
digitalWrite(greenled, LOW);
temperature ();
digitalWrite(tempblueled, LOW);
digitalWrite(tempredled, LOW);
digitalWrite(tempgreenled, LOW);
Water_level();
    digitalWrite(levblueled, LOW);
digitalWrite(levredled, LOW);
digitalWrite(levgreenled, LOW);
turbidity();
 digitalWrite(turbblueled, LOW);
digitalWrite(turbredled, LOW);
digitalWrite(turbgreenled, LOW);
send_sms();
delay(4000);
send_sms1();
  digitalWrite(greenled, LOW);
digitalWrite(tempgreenled, LOW);
digitalWrite(levgreenled, LOW);
digitalWrite(turbgreenled, LOW);
```

```
  delay(8000);
}



///////////////////////////////////////////////////////////// FUNCTIONS
/////////////////////////////////////////////////////////////////////

void PH(){
  Serial.println(" ");
  lcd.clear( );
  digitalWrite(blueled, LOW);
  digitalWrite(greenled, LOW);
  digitalWrite(redled, LOW);
  digitalWrite(buzzer, LOW);
  lcd.setCursor(0,0);//set cursor (colum by row) indexing from 0
  lcd.print("TAKING READINGS");
  lcd.setCursor(1,1);
  lcd.print("FROM PH SENSOR");
  Serial.println("Taking Readings from PH Sensor");
  PHblink();
  int buf[10];                //buffer for read analog
  for(int i=0;i<10;i++)       //Get 10 sample value from the sensor for smooth the value
  {
    buf[i]=analogRead(SensorPin);
    delay(10);
  }
  for(int i=0;i<9;i++)        //sort the analog from small to large
  {
    for(int j=i+1;j<10;j++)
    {
      if(buf[i]>buf[j])
      {
```

```
      int temp=buf[i];
      buf[i]=buf[j];
      buf[j]=temp;
    }
  }
}
avgValue=0;
for(int i=2;i<8;i++)                    //take the average value of 6 center sample
  avgValue+=buf[i];
float phValue=(float)avgValue*3.8/1030/6; //convert the analog into millivolt
phValue=3.3*phValue+Offset;                    //convert the millivolt into pH value
Serial.print("pH:");
Serial.print(phValue,2);
Serial.println(" ");

if(phValue >= 7.30){
  lcd.clear();
  digitalWrite(blueled, LOW);
  digitalWrite(greenled, LOW);
  digitalWrite(redled, HIGH);
  digitalWrite(buzzer, HIGH);
  lcd.setCursor(1,0);//set cursor (colum by row) indexing from 0
  lcd.print("PH VALUE:");
  lcd.setCursor(10,0);
  lcd.print(phValue);
  lcd.setCursor(0,1);
  Serial.print("PH VALUE: ");
  Serial.println(phValue);
  lcd.setCursor(0,1);//set cursor (colum by row) indexing from 0
  lcd.print("ALKALINITY HIGH");
  Serial.println("Water Alkalinity high");
  delay(3000);
 }
```

```
if(phValue >= 6.90 && phValue <= 7.19){
  digitalWrite(blueled, HIGH);
  digitalWrite(greenled,LOW);
  digitalWrite(redled, LOW);
  digitalWrite(buzzer, LOW);
  lcd.clear();
  lcd.setCursor(1,0);//set cursor (colum by row) indexing from 0
  lcd.print("PH VALUE:");
  lcd.setCursor(10,0);
  lcd.print(phValue);
  lcd.setCursor(0,1);
  Serial.print("PH VALUE: ");
  Serial.println(phValue);
  lcd.setCursor(1,1);//set cursor (colum by row) indexing from 0
  lcd.print("WATER IS SAFE");
  Serial.println("Water Is  neutral (safe)");
}

if(phValue < 6.89){
  lcd.clear();
  digitalWrite(blueled, LOW);
  digitalWrite(greenled, LOW);
  digitalWrite(redled, HIGH);
  digitalWrite(buzzer, HIGH);
  lcd.setCursor(1,0);//set cursor (colum by row) indexing from 0
  lcd.print("PH VALUE:");
  lcd.setCursor(10,0);
  lcd.print(phValue);
  lcd.setCursor(0,1);
  Serial.print("PH VALUE: ");
  Serial.println(phValue);
  lcd.setCursor(2,1);//set cursor (colum by row) indexing from 0
```

```
    lcd.print("ACIDITY HIGH");
    Serial.println("Water Acidity High");
    delay(3000);
   }

  delay(8000);
}


void temperature (){
  Serial.println(" ");
  lcd.clear( );
  lcd.setCursor(0,0);//set cursor (colum by row) indexing from 0
  lcd.print("TAKING READINGS");
  lcd.setCursor(0,1);
  lcd.print("FROM TEMP SENSOR");
  Serial.println("Taking Readings from Temperature Sensor");
  TEMPblink();
  temp_check_surr();
  delay(4000);
  temp_check_water();
}


void temp_check_surr(){
  digitalWrite(tempblueled, LOW);
  digitalWrite(tempgreenled, LOW);
  digitalWrite(tempredled, LOW);
  digitalWrite(buzzer, LOW);
  int reading = analogRead(sensorPin);

  // converting that reading to voltage,
  float voltage = reading * 5.0;
  voltage /= 1024.0;

  // now print out the temperature
  float temperatureC = (voltage - 0.5) * 100 ;  //converting from 10 mv per degree wit
500 mV offset
  //to degrees ((voltage - 500mV) times 100)
  lcd.clear();
  Serial.print("Surrounding Temperature: ");
  Serial.println(temperatureC);

  if(temperatureC > 50){
    digitalWrite(tempblueled, LOW);
    digitalWrite(tempgreenled, LOW);
    digitalWrite(tempredled, HIGH);
    digitalWrite(buzzer, HIGH);
    lcd.setCursor(0,0);//set cursor (colum by row) indexing from 0
    lcd.print("SUR TEMP:");
    lcd.setCursor(9,0);
    lcd.print(temperatureC);
    lcd.setCursor(14,0);
    lcd.print("*C");
    lcd.setCursor(0,1);
    Serial.print("Surrounding Temperature: ");
    Serial.print(temperatureC);
    Serial.println(" degree C");
    lcd.setCursor(0,1);//set cursor (colum by row) indexing from 0
    lcd.print("SURR TEMP HIGH");
    Serial.println("Surrounding Temperature high");
    delay(3000);
   }

  if(temperatureC >= 10 && temperatureC <= 50){
    digitalWrite(tempblueled, HIGH);
    digitalWrite(tempgreenled,LOW);
    digitalWrite(tempredled, LOW);
    digitalWrite(buzzer, LOW);
    lcd.setCursor(0,0);//set cursor (colum by row) indexing from 0
    lcd.print("SUR TEMP:");
    lcd.setCursor(9,0);
    lcd.print(temperatureC);
    lcd.setCursor(14,0);
    lcd.print("*C");
    lcd.setCursor(0,1);
    Serial.print("Surrounding Temperature: ");
    Serial.print(temperatureC);
    Serial.println(" degree C");
    lcd.setCursor(0,1);//set cursor (colum by row) indexing from 0
    lcd.print("SURR TEMP NORMAL");
    Serial.println("Surrounding Temperature normal");
  }

  if(temperatureC < 10){
    digitalWrite(tempblueled, LOW);
    digitalWrite(tempgreenled, LOW);
    digitalWrite(tempredled, HIGH);
    digitalWrite(buzzer, HIGH);
    lcd.setCursor(0,0);//set cursor (colum by row) indexing from 0
    lcd.print("SUR TEMP:");
    lcd.setCursor(9,0);
    lcd.print(temperatureC);
    lcd.setCursor(14,0);
    lcd.print("*C");
    lcd.setCursor(0,1);
    Serial.print("Surrounding Temperature: ");
    Serial.print(temperatureC);
    Serial.println(" degree C");
```

```
    lcd.setCursor(0,1);//set cursor (colum by row) indexing from 0
    lcd.print("SURR TEMP LOW");
    Serial.println("Surrounding Temperature low");
    delay(3000);
   }
  delay(8000);
}


void temp_check_water(){
  lcd.clear();
  digitalWrite(tempblueled, LOW);
  digitalWrite(tempgreenled, LOW);
  digitalWrite(tempredled, LOW);
  digitalWrite(buzzer, LOW);
  sensors.requestTemperatures(); // Send the command to get temperature
  Serial.print("Water Temperature: ");
  Serial.println(sensors.getTempCByIndex(0));

  if(sensors.getTempCByIndex(0) > 40){
    digitalWrite(tempblueled, LOW);
    digitalWrite(tempgreenled, LOW);
    digitalWrite(tempredled, HIGH);
    digitalWrite(buzzer, HIGH);
    lcd.setCursor(0,0);//set cursor (colum by row) indexing from 0
    lcd.print("WAT TEMP:");
    lcd.setCursor(9,0);
    lcd.print(sensors.getTempCByIndex(0));
    lcd.setCursor(14,0);
    lcd.print("*C");
    lcd.setCursor(0,1);
    Serial.print("Water Temperature: ");
    Serial.print(sensors.getTempCByIndex(0));
  Serial.println(" degree C");
  lcd.setCursor(0,1);//set cursor (colum by row) indexing from 0
  lcd.print("WATER TEMP HIGH");
  Serial.println("Water Temperature high");
  delay(3000);
}

if(sensors.getTempCByIndex(0) >= 15 && sensors.getTempCByIndex(0) <= 40){
  digitalWrite(tempblueled, HIGH);
  digitalWrite(tempgreenled,LOW);
  digitalWrite(tempredled, LOW);
  digitalWrite(buzzer, LOW);
  lcd.setCursor(0,0);//set cursor (colum by row) indexing from 0
  lcd.print("WAT TEMP:");
  lcd.setCursor(9,0);
  lcd.print(sensors.getTempCByIndex(0));
  lcd.setCursor(14,0);
  lcd.print("*C");
  lcd.setCursor(0,1);
  Serial.print("Water Temperature: ");
  Serial.print(sensors.getTempCByIndex(0));
  Serial.println(" degree C");
  lcd.setCursor(0,1);//set cursor (colum by row) indexing from 0
  lcd.print("WATER TEMP NORMAL");
  Serial.println("Water Temperature normal");
}

if(sensors.getTempCByIndex(0) < 15){
  digitalWrite(tempblueled, LOW);
  digitalWrite(tempgreenled, LOW);
  digitalWrite(tempredled, HIGH);
  digitalWrite(buzzer, HIGH);
  lcd.setCursor(0,0);//set cursor (colum by row) indexing from 0
```

```
    lcd.print("WAT TEMP:");
    lcd.setCursor(9,0);
    lcd.print(sensors.getTempCByIndex(0));
    lcd.setCursor(14,0);
    lcd.print("*C");
    lcd.setCursor(0,1);
    Serial.print("Water Temperature: ");
    Serial.print(sensors.getTempCByIndex(0));
    Serial.println(" degree C");
    lcd.setCursor(0,1);//set cursor (colum by row) indexing from 0
    lcd.print("WATER TEMP LOW");
    Serial.println("Water Temperature low");
    delay(3000);
  }
  delay(8000);
}


void Water_level() {
    Serial.println(" ");
  digitalWrite(levblueled, LOW);
  digitalWrite(levgreenled, LOW);
  digitalWrite(levredled, LOW);
  digitalWrite(buzzer, LOW);
  lcd.clear( );
  lcd.setCursor(2,0);//set cursor (colum by row) indexing from 0
  lcd.print("READINGS FROM");
  lcd.setCursor(0,1);
  lcd.print("WATER LEVEL SENS");
  Serial.println("Taking Readings from Water Level Sensor");
  LEVblink();
  level_check();
  delay(8000);
```

```
}



void level_check(){
  digitalWrite(levblueled, LOW);
  digitalWrite(levgreenled, LOW);
  digitalWrite(levredled, LOW);
  digitalWrite(buzzer, LOW);

    // Notice how there's no delays in this sketch to allow you to do other processing in-line while doing distance pings.
  if (millis() >= pingTimer) {   // pingSpeed milliseconds since last ping, do another ping.
    pingTimer += pingSpeed;      // Set the next ping time.
    sonar.ping_timer(echoCheck); // Send out the ping, calls "echoCheck" function every 24uS where you can check the ping status.
//delay(1000);
}
}

void echoCheck() { // Timer2 interrupt calls this function every 24uS where you can check the ping status.
  // Don't do anything here!
  if (sonar.check_timer()) { // This is how you check to see if the ping was received.
   if(sonar.ping_result / US_ROUNDTRIP_CM > 7){
    digitalWrite(levblueled, LOW);
    digitalWrite(levgreenled, LOW);
    digitalWrite(levredled, HIGH);
    digitalWrite(buzzer, HIGH);
    lcd.clear( );
    lcd.setCursor(0,0);//set cursor (colum by row) indexing from 0
    lcd.print("WATER LEVEL:");
    lcd.setCursor(12,0);
    lcd.print(sonar.ping_result / US_ROUNDTRIP_CM);
    lcd.setCursor(14,0);
    lcd.print("cm");
    lcd.setCursor(0,1);
    Serial.print("Water Level: ");
    Serial.print(sonar.ping_result / US_ROUNDTRIP_CM);
    Serial.println("cm");
    lcd.setCursor(0,1);//set cursor (colum by row) indexing from 0
    lcd.print("WATER LEVEL LOW");
    Serial.println("Water Level low");
    delay(3000);

  }

  if(sonar.ping_result / US_ROUNDTRIP_CM >= 5 && sonar.ping_result /
US_ROUNDTRIP_CM <= 7){
     digitalWrite(levblueled, HIGH);
  digitalWrite(levgreenled, LOW);
  digitalWrite(levredled, LOW);
  digitalWrite(buzzer, LOW);
    lcd.clear( );
    lcd.setCursor(0,0);//set cursor (colum by row) indexing from 0
    lcd.print("WATER LEVEL:");
    lcd.setCursor(12,0);
    lcd.print(sonar.ping_result / US_ROUNDTRIP_CM);
    lcd.setCursor(14,0);
    lcd.print("cm");
    lcd.setCursor(0,1);
    Serial.print("Water Level: ");
    Serial.print(sonar.ping_result / US_ROUNDTRIP_CM);
    Serial.println("cm");
    lcd.setCursor(0,1);//set cursor (colum by row) indexing from 0
    lcd.print("WATER LEVEL NORMAL");
```

```
    Serial.println("Water Level normal");
  }

  if(sonar.ping_result / US_ROUNDTRIP_CM < 5){
    digitalWrite(levblueled, LOW);
    digitalWrite(levgreenled, LOW);
    digitalWrite(levredled, HIGH);
    digitalWrite(buzzer, HIGH);
    lcd.clear( );
    lcd.setCursor(0,0);//set cursor (colum by row) indexing from 0
    lcd.print("WATER LEVEL:");
    lcd.setCursor(12,0);
    lcd.print(sonar.ping_result / US_ROUNDTRIP_CM);
    lcd.setCursor(14,0);
    lcd.print("cm");
    lcd.setCursor(0,1);
    Serial.print("Water Level: ");
    Serial.print(sonar.ping_result / US_ROUNDTRIP_CM);
    Serial.println("cm");
    lcd.setCursor(0,1);//set cursor (colum by row) indexing from 0
    lcd.print("WATER LEVEL HIGH");
    Serial.println("Water Level high");
    delay(2000);
      }
  }
 }

/*long microsecondsToCentimeters(long microseconds)
{
  return microseconds / 29 / 2;
}
*/
```

```
void turbidity() {
  Serial.println(" ");
  digitalWrite(turbblueled, LOW);
  digitalWrite(turbgreenled, LOW);
  digitalWrite(turbredled, LOW);
  digitalWrite(buzzer, LOW);
  lcd.clear( );
  lcd.setCursor(1,0);//set cursor (colum by row) indexing from 0
  lcd.print("READINGS FROM");
  lcd.setCursor(0,1);
  lcd.print("TURBIDITY SENSOR");
  Serial.println("Taking Readings from turbidity Sensor");
  TURBblink();
  int turbidityValue = analogRead(A1);
  float turbidityV = turbidityValue/100;
  Serial.print("Turbidity level: ");
  Serial.println(turbidityV);

  if( turbidityV > 9){
    digitalWrite(turbblueled, LOW);
    digitalWrite(turbgreenled, LOW);
    digitalWrite(turbredled, HIGH);
    digitalWrite(buzzer, HIGH);
    lcd.clear( );
    lcd.setCursor(0,0);//set cursor (colum by row) indexing from 0
    lcd.print("TURBI LEV:");
    lcd.setCursor(11,0);
    lcd.print(turbidityV);
    lcd.setCursor(14,0);
    lcd.print("NTU");
    lcd.setCursor(0,1);
    Serial.print("Turbidity Level: ");
    Serial.print(turbidityV);
    Serial.println("NTU");
    lcd.setCursor(0,1);//set cursor (colum by row) indexing from 0
    lcd.print("WATER VERY CLEAN");
    Serial.println("Water Very Clean ");
    delay(3000);
  }

  if( turbidityV >= 6 && turbidityValue/100 <= 9 ){
    digitalWrite(turbblueled, HIGH);
    digitalWrite(turbgreenled, LOW);
    digitalWrite(turbredled, LOW);
    digitalWrite(buzzer, LOW);
    lcd.clear( );
    lcd.setCursor(0,0);//set cursor (colum by row) indexing from 0
    lcd.print("TURBI LEV:");
    lcd.setCursor(10,0);
    lcd.print(turbidityV);
    lcd.setCursor(13,0);
    lcd.print("NTU");
    lcd.setCursor(0,1);
    Serial.print("Turbidity Level: ");
    Serial.print(turbidityV);
    Serial.println("NTU");
    lcd.setCursor(0,1);//set cursor (colum by row) indexing from 0
    lcd.print("WATER NORM CLEAN");
    Serial.println("Water Clean ");
  }

  if( turbidityV < 6){
    digitalWrite(turbblueled, LOW);
    digitalWrite(turbgreenled, LOW);
    digitalWrite(turbredled, HIGH);
    digitalWrite(buzzer, HIGH);
```

```
    lcd.clear( );
    lcd.setCursor(0,0);//set cursor (colum by row) indexing from 0
    lcd.print("TURBI LEV:");
    lcd.setCursor(11,0);
    lcd.print(turbidityV);
    lcd.setCursor(14,0);
    lcd.print("NTU");
    lcd.setCursor(0,1);
    Serial.print("Turbidity Level: ");
    Serial.print(turbidityV);
    Serial.println("NTU");
    lcd.setCursor(0,1);//set cursor (colum by row) indexing from 0
    lcd.print("WATER VERY DIRTY");
    Serial.println("Water Very Dirty ");
    delay(3000);
   }
  delay (8000);
}


void initialization(){
  lcd.setCursor(0,0);//set cursor (colum by row) indexing from 0
  lcd.print("INITIALIZING ALL");
  lcd.setCursor(0,1);
  lcd.print("PARAMETERS");
  delay(3000);
  lcd.setCursor(0,1);
  lcd.print("PARAMETERS.");
  delay(3000);
  lcd.setCursor(0,1);
  lcd.print("PARAMETERS..");
  delay(3000);
  lcd.setCursor(0,1);
  lcd.print("PARAMETERS...");
  delay(3000);
  lcd.setCursor(0,1);
  lcd.print("PARAMETERS....");
  delay(3000);
  lcd.setCursor(0,1);
  lcd.print("PARAMETERS.....");
  delay(3000);
  lcd.setCursor(0,1);
  lcd.print("PARAMETERS......");
  Serial.println("Initializing All Parameters........");
  delay(8000);
  Serial.println("Initializing Done.");
  lcd.clear();
  lcd.setCursor(1,0);
  lcd.print("INITIALIZATION");
  lcd.setCursor(6,1);
  lcd.print("DONE  ");
  delay(2000);
}

//////////////////////////////////////////////////////// SMS FUNCTION
////////////////////////////////////////////////////////////////////



void send_sms(){
  lcd.clear();
  lcd.setCursor(2,0);//set cursor (colum by row) indexing from 0
  lcd.print("ATTENTION!!! ");
  lcd.setCursor(2,1);
  lcd.print("SENDING SMS ");
  String temp;
  String lev;
  String phm;
  String turb;
  int turbidityValue = analogRead(A1);
  float turbidityV = turbidityValue/100;

  int buf[10];                //buffer for read analog
  for(int i=0;i<10;i++)       //Get 10 sample value from the sensor for smooth the value
  {
    buf[i]=analogRead(SensorPin);
    delay(10);
  }
  for(int i=0;i<9;i++)        //sort the analog from small to large
  {
    for(int j=i+1;j<10;j++)
    {
      if(buf[i]>buf[j])
      {
        int temp=buf[i];
        buf[i]=buf[j];
        buf[j]=temp;
      }
    }
  }
  avgValue=0;
  for(int i=2;i<8;i++)                      //take the average value of 6 center sample
    avgValue+=buf[i];
  float phValue=(float)avgValue*3.8/1030/6; //convert the analog into millivolt
  phValue=3.3*phValue+Offset;

    if(sensors.getTempCByIndex(0) > 40){
    temp = String("HIGH");
    }
    if(sensors.getTempCByIndex(0) >= 10 && sensors.getTempCByIndex(0) <= 40){
    temp = String("NORMAL");
    }
    if(sensors.getTempCByIndex(0) < 10){
     temp = String("LOW");
    }

    if(sonar.ping_result / US_ROUNDTRIP_CM > 8){
    lev = String("LOW");
    }
    if(sonar.ping_result / US_ROUNDTRIP_CM >= 5 && sonar.ping_result /
 US_ROUNDTRIP_CM <= 8){
    lev = String("NORMAL");
    }
    if(sonar.ping_result / US_ROUNDTRIP_CM < 5){
    lev = String("HIGH");
    }

    if(phValue >= 7.30){
    phm = String("ALKALINE");
    }
    if(phValue >= 6.90 && phValue <= 7.19){
    phm = String("NORMAL");
    }
    if(phValue < 6.89){
     phm = String("ACIDIC");
    }

    if(turbidityV >= 6 && turbidityValue/100 <= 9){
    turb = String("CLEAN");
    }
    if(turbidityV < 6){
    turb = String("DIRTY");
    }
```

```
  mySerial.begin(19200);  //Default serial port setting for the GPRS modem is 19200bps 8-N-1
  mySerial.print("\r");
  digitalWrite(buzzer, LOW);
  digitalWrite(blueled, LOW);
  digitalWrite(greenled, LOW);
  digitalWrite(redled, LOW);
  delay(1000);                    //wait for a second while the modem sends an "OK"
  mySerial.print("AT+CMGF=1\r");    //Because we want to send the SMS in text mode
  delay(1000);
  mySerial.print("AT+CMGS=\"+233540518223\"\r");    //Start accepting the text for the message
  delay(1000);
  mySerial.print(temp);
  mySerial.print(" \r");
  mySerial.print("WATER TEMPERATURE= \r");   //The text for the message
  mySerial.print(sensors.getTempCByIndex(0));
  mySerial.print("*C\r");
  mySerial.println("\r");
  mySerial.print(lev);
  mySerial.print(" \r");
  mySerial.print("WATER LEVEL= \r");   //The text for the message
  mySerial.print(sonar.ping_result / US_ROUNDTRIP_CM);
  mySerial.print("cm\r");
  mySerial.println("\r");
  mySerial.print(phm);
  mySerial.print(" \r");
  mySerial.print("WATER PH VALUE= \r");   //The text for the message
  mySerial.print(phValue);
  mySerial.println("\r");
  mySerial.print(turb);
  mySerial.print(" \r");
  mySerial.print("WATER TURBIDITY= \r");   //The text for the message
  mySerial.print(turbidityV);
  mySerial.print("NBT\r");

  digitalWrite(greenled, HIGH);
  digitalWrite(tempgreenled, HIGH);
  digitalWrite(levgreenled, HIGH);
  digitalWrite(turbgreenled, HIGH);
  delay(3000);
  /*lcd.clear();
  lcd.setCursor(5,0);//set cursor (colum by row) indexing from 0
  lcd.print("SMS SENT ");
  lcd.setCursor(2,1);
  lcd.print("SUCCESSFULLY ");*/
  mySerial.write(0x1A);  //Equivalent to sending Ctrl+Z
}

void send_sms1(){
  lcd.clear();
  lcd.setCursor(2,0);//set cursor (colum by row) indexing from 0
  lcd.print("ATTENTION!!! ");
  lcd.setCursor(2,1);
  lcd.print("SENDING SMS ");
  String temp;
  String lev;
  String phm;
  String turb;
  int turbidityValue = analogRead(A1);
  float turbidityV = turbidityValue/100;

  int buf[10];                //buffer for read analog
  for(int i=0;i<10;i++)       //Get 10 sample value from the sensor for smooth the value
```

```
{
  buf[i]=analogRead(SensorPin);
  delay(10);
}
for(int i=0;i<9;i++)        //sort the analog from small to large
{
  for(int j=i+1;j<10;j++)
  {
    if(buf[i]>buf[j])
    {
      int temp=buf[i];
      buf[i]=buf[j];
      buf[j]=temp;
    }
  }
}
avgValue=0;
for(int i=2;i<8;i++)                      //take the average value of 6 center sample
  avgValue+=buf[i];
float phValue=(float)avgValue*3.8/1030/6; //convert the analog into millivolt
phValue=3.3*phValue+Offset;

  if(sensors.getTempCByIndex(0) > 40){
  temp = String("HIGH");
  }
  if(sensors.getTempCByIndex(0) >= 10 && sensors.getTempCByIndex(0) <= 40){
  temp = String("NORMAL");
  }
  if(sensors.getTempCByIndex(0) < 10){
   temp = String("LOW");
  }

  if(sonar.ping_result / US_ROUNDTRIP_CM > 8){
   lev = String("LOW");
   }
   if(sonar.ping_result / US_ROUNDTRIP_CM >= 5 && sonar.ping_result /
US_ROUNDTRIP_CM <= 8){
   lev = String("NORMAL");
   }
   if(sonar.ping_result / US_ROUNDTRIP_CM < 5){
   lev = String("HIGH");
   }

   if(phValue >= 7.30){
   phm = String("ALKALINE");
   }
   if(phValue >= 6.90 && phValue <= 7.19){
   phm = String("NORMAL");
   }
   if(phValue < 6.89){
    phm = String("ACIDIC");
   }

   if(turbidityV >= 6 && turbidityValue/100 <= 9){
   turb = String("CLEAN");
   }
   if(turbidityV < 6){
   turb = String("DIRTY");
   }

  mySerial.begin(19200);  //Default serial port setting for the GPRS modem is
19200bps 8-N-1
  mySerial.print("\r");
  digitalWrite(buzzer, LOW);
  digitalWrite(blueled, LOW);
  digitalWrite(greenled, LOW);
  digitalWrite(redled, LOW);
  delay(1000);                    //wait for a second while the modem sends an "OK"
  mySerial.print("AT+CMGF=1\r");    //Because we want to send the SMS in text
mode
  delay(1000);
  mySerial.print("AT+CMGS=\"+233265188849\"\r");    //Start accepting the text
for the message
  delay(1000);
  mySerial.print(temp);
  mySerial.print(" \r");
  mySerial.print("WATER TEMPERATURE= \r");   //The text for the message
  mySerial.print(sensors.getTempCByIndex(0));
  mySerial.print("*C\r");
  mySerial.println("\r");
  mySerial.print(lev);
  mySerial.print(" \r");
  mySerial.print("WATER LEVEL= \r");   //The text for the message
  mySerial.print(sonar.ping_result / US_ROUNDTRIP_CM);
  mySerial.print("cm\r");
  mySerial.println("\r");
  mySerial.print(phm);
  mySerial.print(" \r");
  mySerial.print("WATER PH VALUE= \r");   //The text for the message
  mySerial.print(phValue);
  mySerial.println("\r");
  mySerial.print(turb);
  mySerial.print(" \r");
  mySerial.print("WATER TURBIDITY= \r");   //The text for the message
  mySerial.print(turbidityV);
  mySerial.print("NBT\r");

  digitalWrite(greenled, HIGH);
  digitalWrite(tempgreenled, HIGH);
  digitalWrite(levgreenled, HIGH);
  digitalWrite(turbgreenled, HIGH);
  delay(3000);
  lcd.clear();
  lcd.setCursor(5,0);//set cursor (colum by row) indexing from 0
  lcd.print("SMS SENT ");
  lcd.setCursor(2,1);
  lcd.print("SUCCESSFULLY ");
  mySerial.write(0x1A);  //Equivalent to sending Ctrl+Z
}

void PHblink() {
  digitalWrite(blueled, HIGH);
  delay(1000);
  digitalWrite(blueled, LOW);
  delay(1000);
  digitalWrite(blueled, HIGH);
  delay(1000);
  digitalWrite(blueled, LOW);
  delay(1000);
  digitalWrite(blueled, HIGH);
  delay(1000);
  digitalWrite(blueled, LOW);
  delay(1000);
   digitalWrite(blueled, HIGH);
  delay(1000);
  digitalWrite(blueled, LOW);
  delay(1000);
}

void TEMPblink() {
  digitalWrite(tempblueled, HIGH);
  delay(1000);
  digitalWrite(tempblueled, LOW);
  delay(1000);
  digitalWrite(tempblueled, HIGH);
  delay(1000);
  digitalWrite(tempblueled, LOW);
  delay(1000);
  digitalWrite(tempblueled, HIGH);
  delay(1000);
  digitalWrite(tempblueled, LOW);
  delay(1000);
   digitalWrite(tempblueled, HIGH);
  delay(1000);
  digitalWrite(tempblueled, LOW);
  delay(1000);
}

void LEVblink() {
  digitalWrite(levblueled, HIGH);
  delay(1000);
  digitalWrite(levblueled, LOW);
  delay(1000);
  digitalWrite(levblueled, HIGH);
  delay(1000);
  digitalWrite(levblueled, LOW);
  delay(1000);
  digitalWrite(levblueled, HIGH);
  delay(1000);
  digitalWrite(levblueled, LOW);
  delay(1000);
   digitalWrite(levblueled, HIGH);
  delay(1000);
  digitalWrite(levblueled, LOW);
  delay(1000);
```

```
}

void TURBblink() {
  digitalWrite(turbblueled, HIGH);
  delay(1000);
  digitalWrite(turbblueled, LOW);
  delay(1000);
  digitalWrite(turbblueled, HIGH);
  delay(1000);
  digitalWrite(turbblueled, LOW);
  delay(1000);
  digitalWrite(turbblueled, HIGH);
  delay(1000);
  digitalWrite(turbblueled, LOW);
  delay(1000);
   digitalWrite(turbblueled, HIGH);
  delay(1000);
  digitalWrite(turbblueled, LOW);
  delay(1000);
}
```

13.2.GIT HUB LINK

https://github.com/IBM-Project-47610-1660800442