TEAM DETAILS

TEAM ID	**PNT2022TMID22019**
BATCH	**B4-4M6E**
TEAM MEMBERS	**Madhan Raj G** **Abinav Gowtham N** **Annamalai S N** **Jeyanth J**
PROJECT TITLE	**Virtual Eye - Life Guard For Swimming Pools To Detect Active Drowning**

TABLE OF CONTENTS

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 backyards. 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.

By studying body movement patterns and connecting cameras to artificial intelligence (AI) systems we can devise an underwater pool safety system that reduces the risk of drowning. Usually, such systems can be developed by installing more than 16 cameras underwater and ceiling and analyzing the video feeds to detect any anomalies. but AS a POC we make use of one camera that streams the video underwater and analyses the position of swimmers to assess the probability of drowning, if it is higher then an alert will be generated to attract lifeguards' attention.

Fig : 1.1 Swimming and Drowning

Note: The system is not designed to replace a lifeguard or other human monitor, but to act as additional tool. “It helps the lifeguard to detect the underwater situation where they can’t easily observe.

1.2 Purpose

The main purpose of the project is to avoid death in the swimming pool caused by drowning. This project uses AI technology for detecting the movements of humans. While people are swimming in the swimming pool, the camera detects the human and uses AI. The normal movements are detected as the green color and the drowning movements are detected as red color and give the beep alarm. The beep alarm is attract the lifeguard and the lifeguard enters the pool and saves the drowning person.

The specific goals of the drowning project:

1) Implement a surveillance system to capture public swimming pool-related injuries and deaths occurring.

2) AI used in this project.

3) Increase awareness of drowning risk.

2. LITERATURE SURVEY

2.1 Existing Problem

1. Automated Vision-based Surveillance System to Detect Drowning Incidents in Swimming Pools, 2020, Abdel Ilah N. Alshbatat, Shamma Alhameli, Shamsa Almazrouei, Salama Alhameli, Wadhha Almara

The system consists of a Raspberry Pi with the Raspbian operating system,a Pixy camera, an Arduino Nano board, stepper motors, an alarm system, and motor drivers. The proposed system is based on the color-based algorithm to position and rescue swimmers who are drowning. The device then sends an alarm to the lifeguards. This model not only detects drowning but also tracks the swimmers.The system performed well during several experiments carried out in the laboratory. There is no proof that this system will work in any pool.This system is customly built for a particular environment.And the effectiveness of the model is not tested in any new environment

2. Computer Vision Enabled Drowning Detection System,2021, U. Handalage, N. Nikapotha, C. Subasinghe, T. Prasanga, T. Thilakarthna and D. Kasthurirathna

Using convolutional neural network (CNN) models, it can detect a drowning person in three stages(drowning detection, the rescuing drone, and the hazardous activity detection). Whenever such a situation like this is detected, the inflatable tube-mounted selfdriven drone will go on a rescue mission, sounding an alarm to inform the nearby lifeguards. Identifies drowning victims in a minimum amount of time and dispatches an automated drone to save them .Confined with a few of the hardware limitations, such as the use of a single camera and the Jetson Nano at the presence of better-quality hardware, could affect the speed and accuracy of the overall system.

3. Video Based Drowning Detection System, 2021, Pavithra P, Nandini S,Nanthana A,Noor Tabreen Aslam, Praveen Kumar P

The proposed system structure here comprises of a raspberry pi (Single Board Computer) equipped with a USB camera for taking the live feed from the pool area. The system also covers the alerting phenomena using a buzzer so that necessary actions are taken intermittently without any delay. Alerting a drowning state is done without any delay here, GPIO system for alerting and short message service used in cohession with a raspberry pi computer makes this possible .A working implementation of this module is quite extensive to implement, and multiple hardware compenents working to near proximity of water can also lead to some malfunctioning.

4. Deep Learning Used to Recognition Swimmers Drowning, 2021 ,Jia-Xian Jian, Chuin-Mu Wang

Using image processing technology to introduce artificial intelligence motion technology,mounting the camera on the bottom of the swimming pool, and use OpenPose to mark the image joint point features, and input the captured joint point features into the recursive neural network to determine whether the swimmer is drowning. The final training result is about 89.4% accurate, so it can be used to assist on-site lifeguards to detect swimmers who may be drowning.Too much air bubbles generated by the drowning swimmer in the water will also occur. There is a chance that the action cannot be captured by the computer

5. Identification of Drowning Victims in Freshwater Bodies using Drift Prediction and Image Processing based on Deep Learning, 2022, Anjana Unnikrishnan, Roshni A T, Anusha P R, Anju M Vinny, Anuraj CK

Using multiple sensor data in underwater human rescue detection system to spot drifting and drowning person in a natural water eco system. The water flow sensor which is attached to the portable device calculates the drift distance and tracks drowning person. The Approach detected human drifting and drowning up to a range of 5m in water bodies. The final result achieved an average of 82.10% accuracy. The performance of the model depends on the nature of the water body concerned as the drift distance is different for different water ecosystems.

2.2 References

[1] A Survey of Drowning Detection Techniques; Nagato Konishi , Yo Ishigaki ,Seizi linuma ,Tsubasa Nakada ,Taisuke Hoshino, Wataru Nemoto, Kazunori Ohkawara; 2021 International Mobile, Intelligent,and Ubiquitous Computing,09 June.

[2] Automated Vision-based Surveillance System to Detect Drowning Incidents in Swimming Pools; Abdel llah N. Alshbatat, Shamma Alhameli, Shamsa Almazrouei, Salama Alhameli, Wadhha Almarar; 2020 Advances in Science and Engineering Technology International Conferences (ASET), 16 June 2020.

[3] Computer Vision Enabled Drowning Detection System; Upulie Handalage,Nisansali Nikapotha,Chanaka Subasinghe ,Tereen Prasanga ,Thusithanjana Thilakarthn; 2021 3rd International Conference on Advancements in Computing (ICAC), 11 January 20.

[4] A Novel Drowning Detection Method for Safety of Swimmers; : Ajil Roy, K. Srinivasan; 2018 20th National Power Systems Conference (NPSC), 25 July 2019.

[5] . Identification of Drowning Victims in Freshwater Bodies using Drift Prediction and Image Processing based on Deep Learning, 2022, Anjana Unnikrishnan, Roshni A T, Anusha P R, Anju M Vinny, Anuraj CK.

[6] Deep Learning Used to Recognition Swimmers Drowning, 2021 ,Jia-Xian Jian, Chuin-Mu Wang

[7] Video Based Drowning Detection System, 2021, Pavithra P, Nandini S,Nanthana A,Noor Tabreen Aslam, Praveen Kumar P

[8] Automated Vision-based Surveillance System to Detect Drowning Incidents in Swimming Pools, 2020, Abdel Ilah N. Alshbatat, Shamma Alhameli, Shamsa Almazrouei, Salama Alhameli, Wadhha Almara

[9] Computer Vision Enabled Drowning Detection System,2021, U. Handalage, N. Nikapotha, C. Subasinghe, T. Prasanga, T. Thilakarthna and D. Kasthurirathna

[10] Drowning in Swimming Pools: clinical features and safety recommendations based on a study of descriptive records by emergency medical services attending to 995 calls Joanna Shi-En Chan, MBBS, MCEM,[1] Marie Xin Ru Ng, BSc, MPH,[2] and Yih Yng Ng, MBBS, MPH[2] ;Singapore Medical Journal, Jan 2018.

2.3 Problem Statement Definition

Swimming is a great urban stress-reliever. Hotels and tourist spots have more swimming pools than private homes. Beginners have trouble breathing underwater, causing breathing problems and drowning. Drowning increases global mortality without harming children. Under-6-year-olds have the highest drowning mortality rates globally. These deaths are the third cause of unplanned death globally, with 1.2 million cases yearly. To save lives, a meticulous system must be implemented along swimming pools. By studying body movement patterns and connecting cameras to AI systems, we can create a safer underwater pool. Installing 16 underwater and ceiling cameras and analysing video feeds can create such systems. As a POC, we use one camera that streams underwater video and analyses swimmers' positions to assess drowning risk; if it's high, an alert is generated to alert lifeguards.

PROBLEM - 1

I am	I'm trying to	But	Because	Which makes me feel
Lifeguard	Save the people.	I can't save those people without prior intimation.	There is no detection system.	Helpless

PROBLEM - 2

I am	I'm trying to	But	Because	Which makes me feel
Beginner in swimming	Swim on the pool.	It hesitates me a little.	I don't know swimming.	Panic

PROBLEM - 3

I am	I'm trying to	But	Because	Which makes me feel
Parent	Get my kid into swimming.	I can't leave him alone to swim.	Drowning is more possible.	Fear

PROBLEM - 4

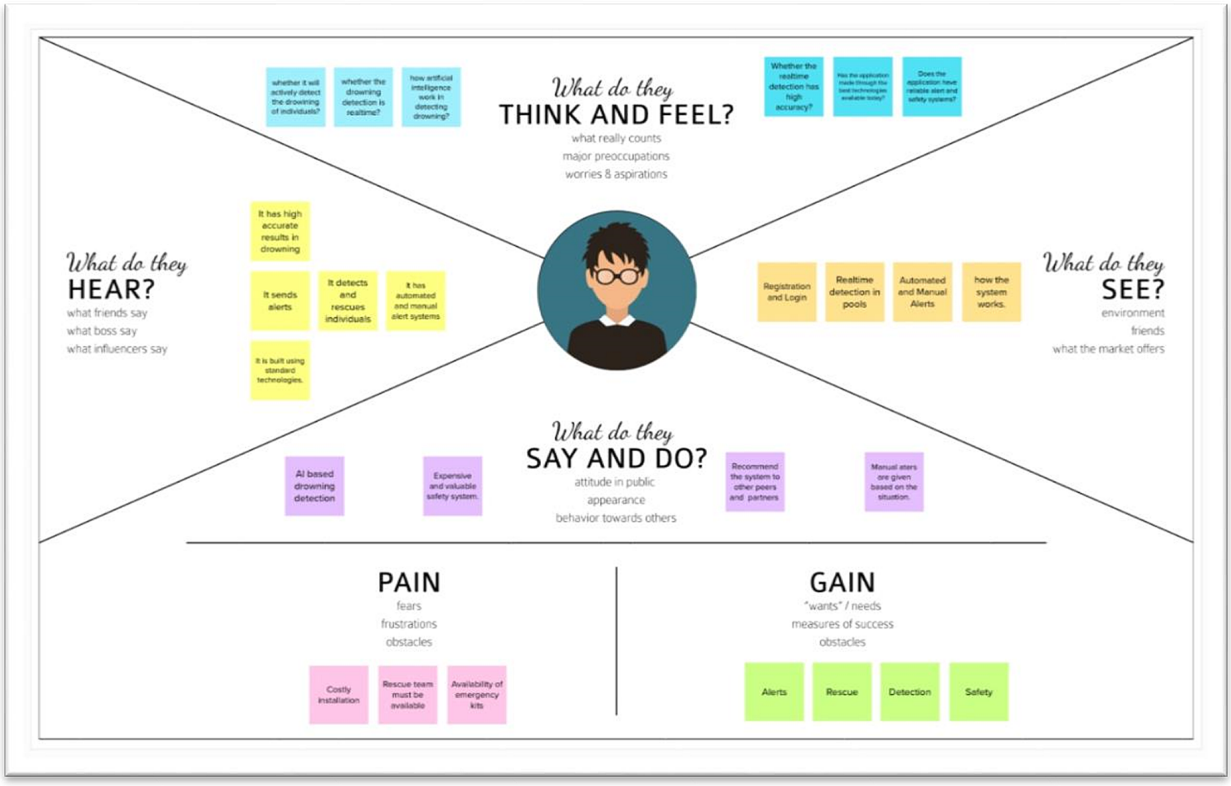
I am	I'm trying to	But	Because	Which makes me feel
Depressed person.	Relax my mind by swimming.	I can't swim on my own	If I accidently drown.	Afraid

PROBLEM - 5

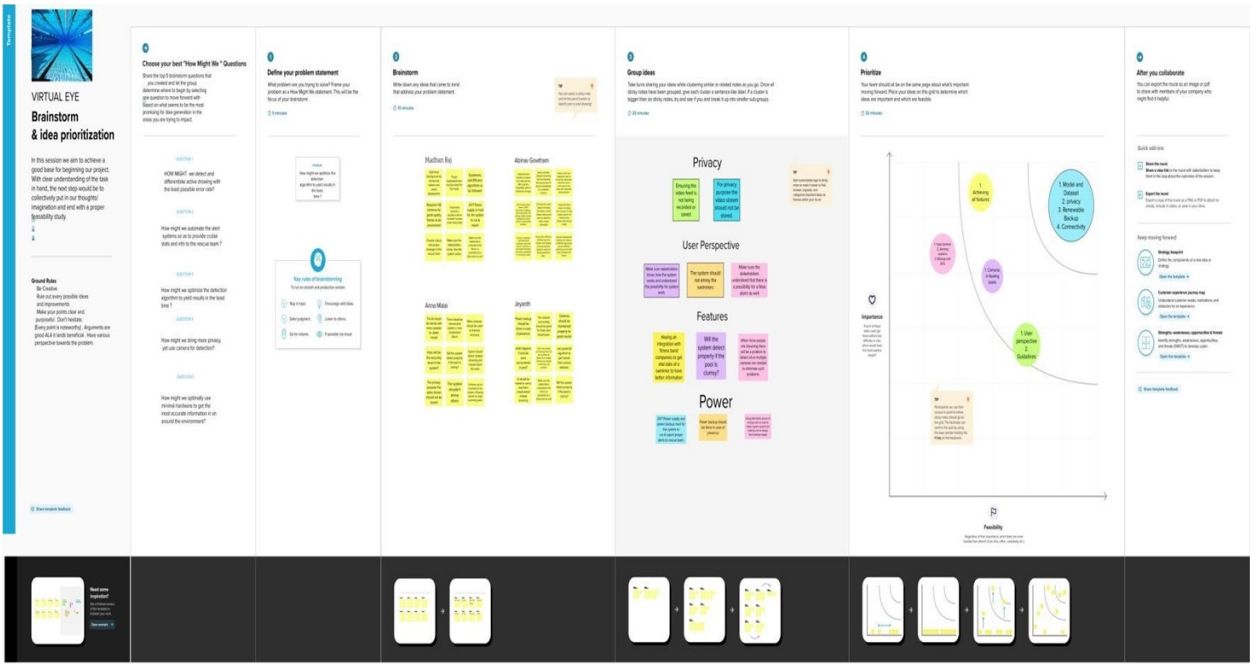
I am	I'm trying to	But	Because	Which makes me feel
Pool owner	Give high security.	I can't ensure the safety.	More likely to drown.	Pressure

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming

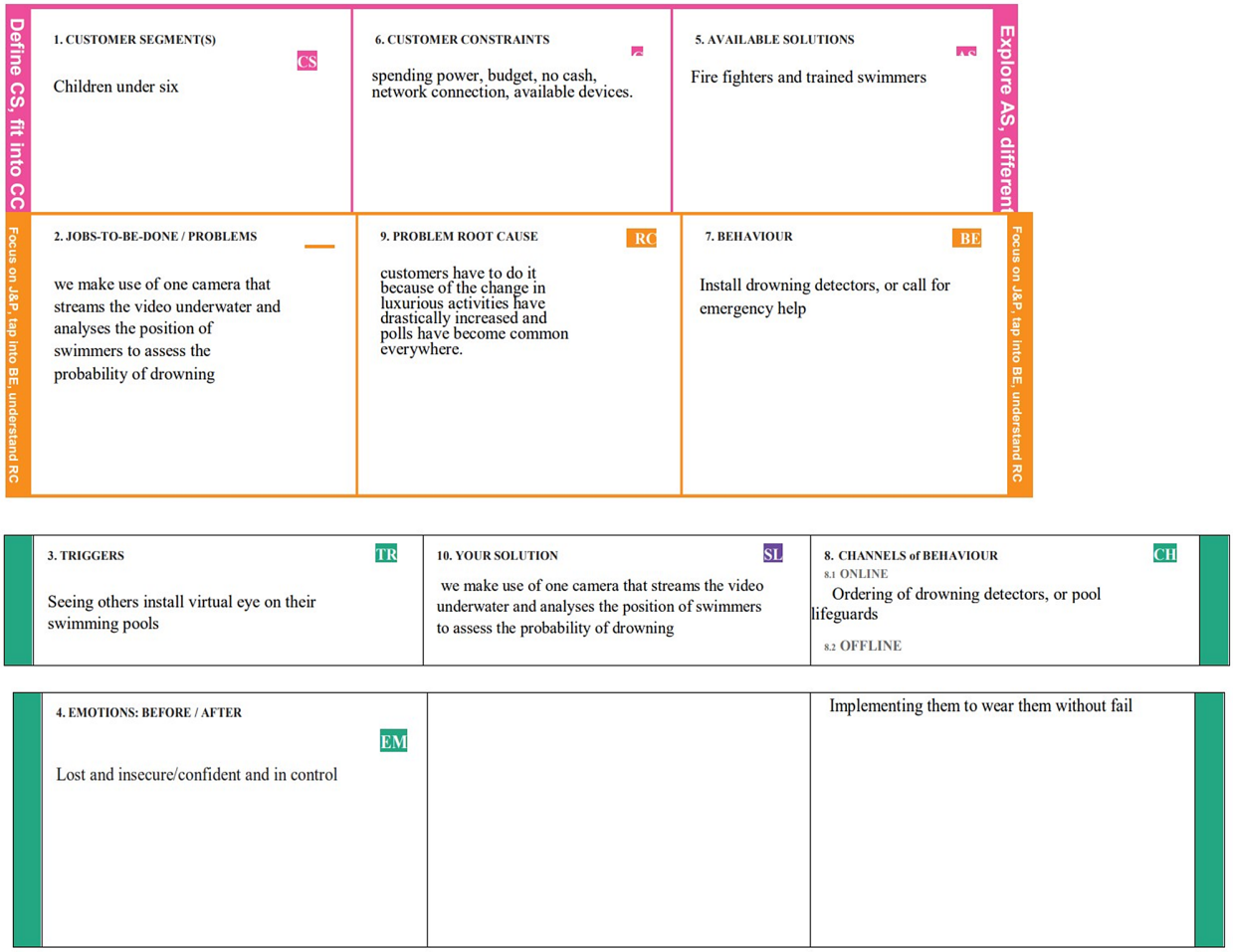


3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Swimming pools are generally places of fun and healthy exercise, but they can be deadly as well. Even with a lifeguard observer on duty, swimmers may still have trouble in underwater or in parts of the pool beyond the lifeguard's field of view.
2.	Idea / Solution description	In this project, we use Artificial Intelligence. We install the cameras in underwater to detect the drowning people. Using deep learning, image can be recognized. If the image is detected, it triggers the alarm to alert the Life Guard who rescue the drowning peoples.
3.	Novelty / Uniqueness	The uniqueness of our system software to track the position and the location of a drowning person. We use YOLO Algorithm. Because of its high accuracy and fast detection speed. So it helps lifeguard to save people within seconds.
4.	Social Impact / Customer Satisfaction	Drowning globally has a higher death rate and is also the third leading cause of unexpected deaths worldwide, especially among children under the age of six. To overcome this conflict our drowning detection system will have an impact on society.
5.	Business Model (Revenue Model)	We can introduce the software-based approach for making a good income. It is extremely useful to lifeguards, swimmers and business operators. The number of features makes it attractive for end users to use our software system.

6.	Scalability of the Solution	Our software system can be used by the company driver who manages the pools. We use the IBM cloud server to collect and maintain the data. We will ensure the safety of the swimmers.

3.4 PROBLEM SOLUTION FIT



4. REQUIREMENT ANALYSIS

4.1 Functional Requirement

Product Requirements :

- Python (libraries, packages, open CV and Flask, etc..,)
- Web Languages
- IBM Cloud

4.2 Non Functional Requirement

Hardware Requirements :

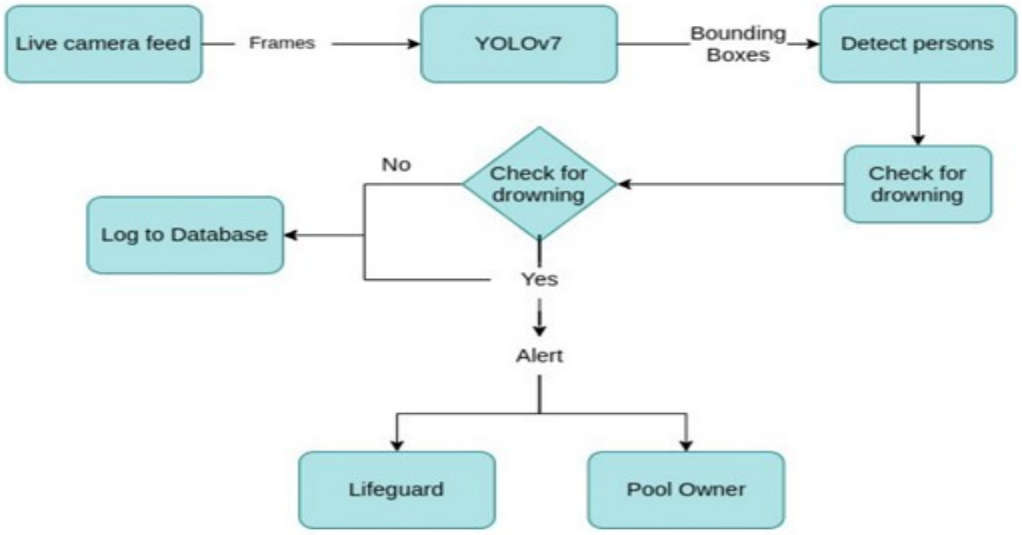
- CPU Type : Intel i3 Core or above
- Clock Speed : 3.0 GHz
- RAM Size : 4GB or above
- Hard Disk Capacity : 1TB
- Camera : 1920 x 1080 px

Software Requirements :

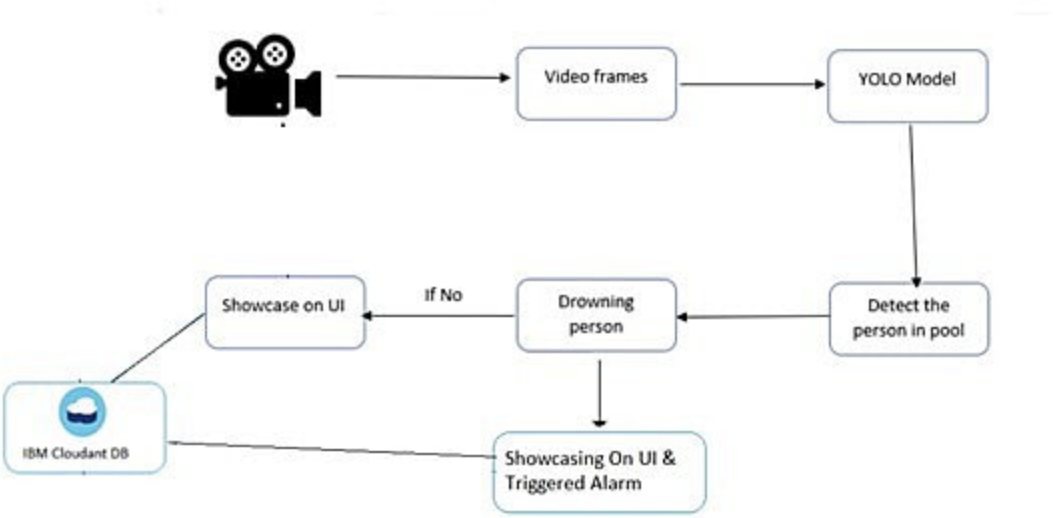
- Operating System : Windows 10/11
- Language : Python
- IDE : Pycharm

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 (Pool owner)	Installation	USN-1	As a pool owner, I can install the cameras and set up the drowning detection system	I can connect the cameras to the cloud-hosted software	High	Sprint-1
	Detecting the drowning persons	USN-2	As a user, I can find the drowning persons by using the drowning detection system	I would receive an alert if a person is drowning	High	Sprint-1
	Notify the lifeguard	USN-3	As a user, I can notify the lifeguard when the system detects a drowning person	I can set up an alarm that would notify the lifeguard	High	Sprint-2
Customer (Lifeguard)	Rescue people	USN-4	As a user, I can rescue the drowning persons from the pool	I can save the drowning person	High	Sprint-2

Customer (Swimmers)	Safety	USN-5	As a user, I can swim without the fear of drowning	I can swim safely with the help of the system and the lifeguard	Medium	Sprint-2
Customer Care Executive	Contact	USN-6	resolve technical issues	I can contact the customer care executive to resolve any issues	Medium	Sprint-3
Administrator	Dashboard	USN-7	Management of the drowning detection system and database management.	I can access the system's logs and any other data instantly	High	Sprint-4

6. PROJECT PLANNING & SCHEDULING

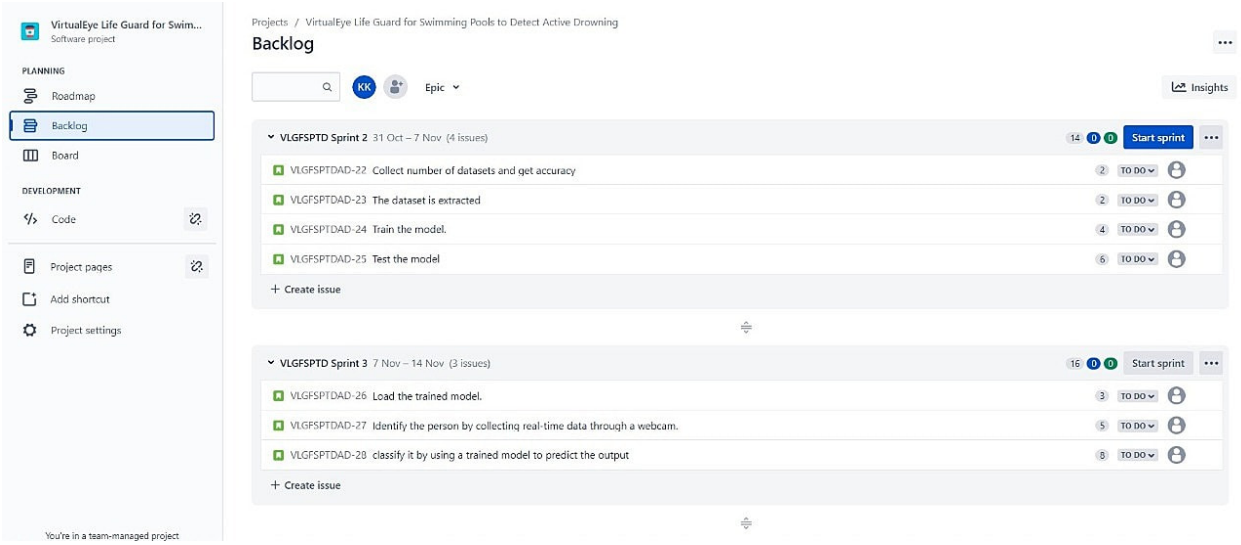
6.1 Sprint Planning & Estimation

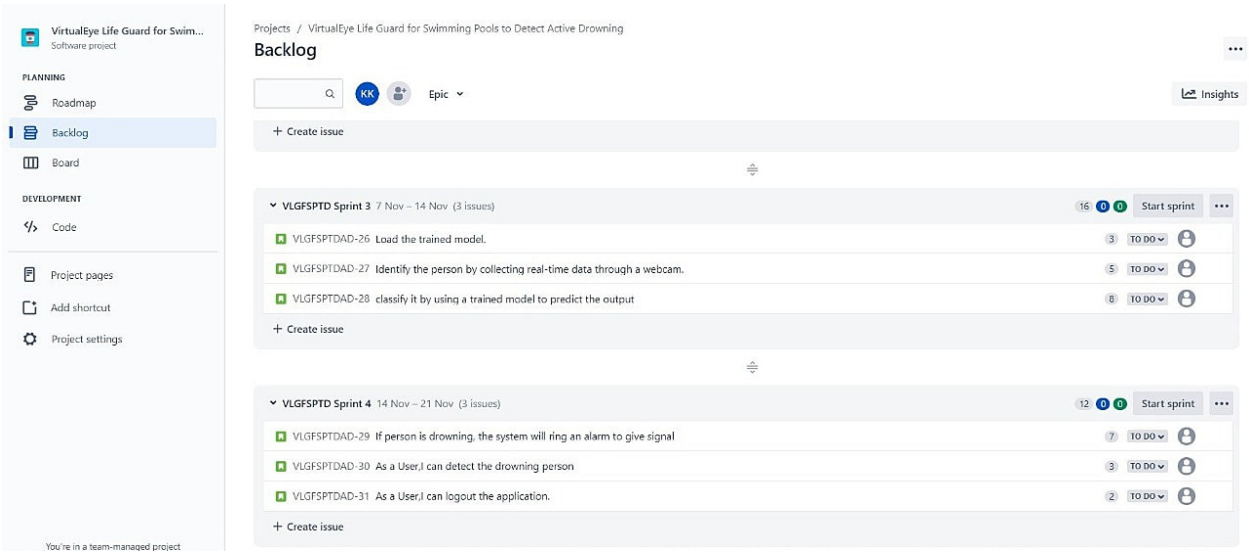
User Type	**Functional Requirement (Epic)**	**Release**
Customer (Pool owner)	Installation	Sprint-1
	Detecting the drowning persons	Sprint-1
	Notify the lifeguard	Sprint-2
Customer (Lifeguard)	Rescue people	Sprint-2
Customer (Swimmers)	Safety	Sprint-2
Customer Care Executive	Contact	Sprint-3
Administrator	Dashboard	Sprint-4

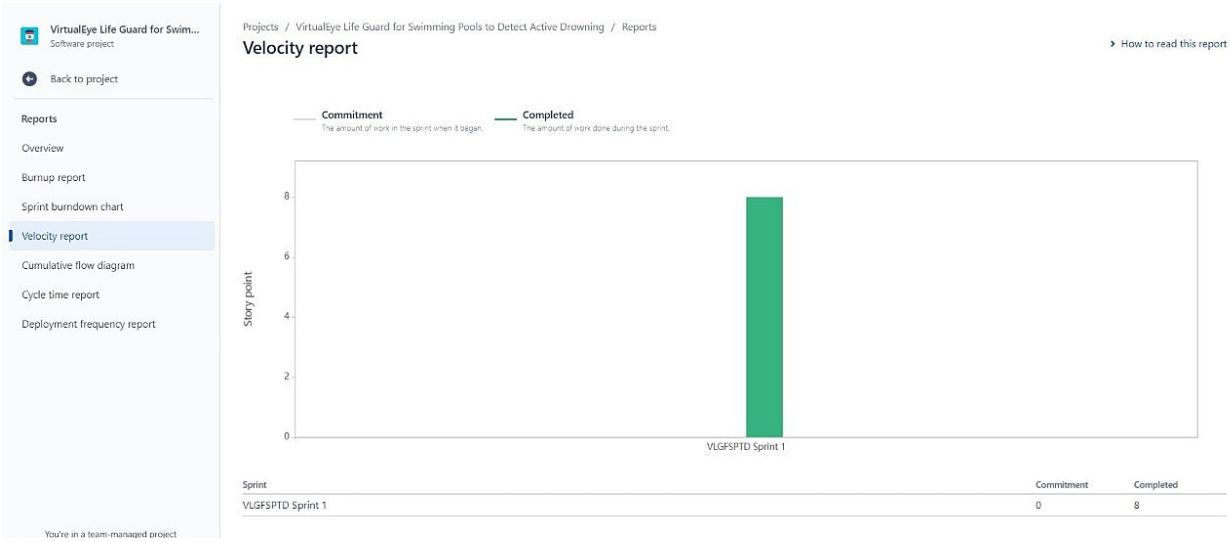
6.2 Sprint Delivery Schedule

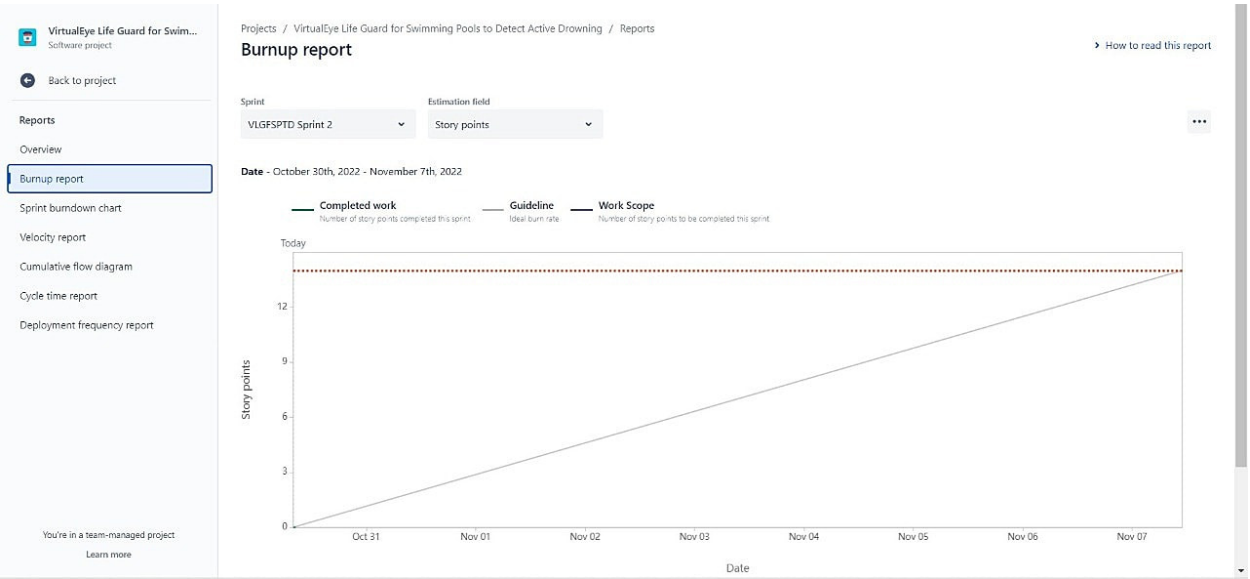
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	8	6 Days	24 Oct 2022	29 Oct 2022	6	29 Oct 2022
Sprint-2	14	6 Days	31 Oct 2022	05 Nov 2022	12	05 Nov 2022
Sprint-3	16	6 Days	07 Nov 2022	12 Nov 2022	11	12 Nov 2022
Sprint-4	12	6 Days	14 Nov 2022	19 Nov 2022	12	19 Nov 2022

6.3 Reports from JIRA




VirtualEye Life Guard for Swim...
Software project
PLANNING
Roadmap
Backlog
Board
DEVELOPMENT
Code
Project pages
Add shortcut
Project settings
You're in a team-managed project
Projects / VirtualEye Life Guard for Swimming Pools to Detect Active Drowning
Backlog
Epic
Insights
Create issue
VLGFSPTD Sprint 3 7 Nov – 14 Nov (3 issues)
Start sprint
VLGFSPTDAD-26 Load the trained model.
VLGFSPTDAD-27 Identify the person by collecting real-time data through a webcam.
VLGFSPTDAD-28 classify it by using a trained model to predict the output
TO DO
Create issue
VLGFSPTD Sprint 4 14 Nov – 21 Nov (3 issues)
Start sprint
VLGFSPTDAD-29 If person is drowning, the system will ring an alarm to give signal
VLGFSPTDAD-30 As a User,I can detect the drowning person
VLGFSPTDAD-31 As a User,I can logout the application.
TO DO
Create issue


VirtualEye Life Guard for Swim...
Software project
Back to project
Reports
Overview
Burnup report
Sprint burndown chart
Velocity report
Cumulative flow diagram
Cycle time report
Deployment frequency report
You're in a team-managed project
Projects / VirtualEye Life Guard for Swimming Pools to Detect Active Drowning / Reports
Velocity report
How to read this report
Commitment
The amount of work in the sprint when it began.
Completed
The amount of work done during the sprint.
Story point
0
2
4
6
8
VLGFSPTD Sprint 1
Sprint
Commitment
Completed
VLGFSPTD Sprint 1
0
8



7. CODING & SOLUTIONING

7.1 Feature 1 – WEB UI

The below code is used to go into web user interface so the user uses the application in very friendly.

```
app=Flask(__name__)
@app.route('/')
def index():
    return render_template('index.html')
@app.route('/index.html')
def home():
    return render_template("index.html")
@app.route('/register')
def register():
    return render_template('register.html')
```

```
@app.route('/afterreg', methods = ['POST'])
def afterreg():
    x = [x for x in request.form.values()]
    print(x)
    data = {
    '_id': x[1],
    'name': x[0],
    'psw': x[2]
    }
    print(data)
    query = {'_id': {'$eq': data['_id']}}
    docs = my_database.get_query_result(query)
    print(docs)
    print(len(docs.all()))
    if(len(docs.all())==0):
        url = my_database.create_document(data)
          return render_template('register.html', pred = "Registration Successful, please login using your details")
    else:
        return render_template('register.html', pred = "You are already a member, please login using your details")
@app.route('/login')
def login():
    return render_template('login.html')
@app.route('/afterlogin',methods = ['POST'])
def afterlogin():
    user = request.form['_id']
```

```
    passw = request.form['psw']
    print(user,passw)
 query = {'_id':{'$eq': user}}
docs = my_database.get_query_result(query)
    print(docs)
print(len(docs.all()))
if(len(docs.all())==0):
        return render_template('login.html', pred = "The username is not found.")
    else:
        if((user==docs[0][0]['_id'] and passw==docs[0][0]['psw'])):
            return redirect(url_for('prediction'))
        else:
            print('Invalid User')
@app.route('/logout')
def logout():
    return render_template('logout.html')
@app.route('/prediction')
def prediction():
    return render_template('prediction.html')
@app.route('/result', methods=["GET","POST"])
def res():
    webcam = cv2.VideoCapture('swimming.mp4')
    if not webcam.isOpened():
        print("Could not open webcam")
        exit()
    t0 = time.time()
    centre0 = np.zeros(2)
```

7.2 Feature 2 – Alarm Beep

The below code segment has the feature to give the beep sound.

```
if(isDrowning == True):
        playsound('alarm.mp3')
        webcam.release()
        cv2.destroyAllWindows()
            return render_template('prediction.html',prediction="Emergency !!! The Person is drowning")
```

7.3 Feature 3 – Cloudant DB

The below cloud segment is used for connecting to the cloud to store the data.

```
from cloudant.client import Cloudant
client = Cloudant.iam('a2aae5ca-a0d0-40c9-aa41-b2fe01527d5f-bluemix','FF_IpVZywe3MAZlo4X5QDT97kQU0vgS-52jqLtFrjh9b', connect = True)
my_database = client.create_database('my_database')
query = {'_id': {'$eq': data['_id']}}
 docs = my_database.get_query_result(query)
```

8.TESTING

8.1 Test Cases

Testing is one of the most crucial stages in the software development process. The main goal of the testing phase in the software development life cycle (SDLC) is to ensure that the developed software meets the necessary functionality and performance. There are several test cases related to the project.

Libraries and packages	Console working
Code Segments Execution	Detection
Running the application	Drowning Detection
Navigating the web pages	Alarm Beep

Test Case ID	Feature Type		Test Scenerio	Step to Execute	Test	Expected result	Actual Result
	Feature Type	Home Page	Test Scenario Verify user is able to see the Login/Signup popupwhen user clickedon My account button	Steps TO Execute I.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Sign up popup displayed or not	Test Login.html	Expected Result Login/Signup popup should display	ActualResult Working as
LoginPage_TC_002		Home Page	Verify the UI elements in Login/Signup popup	I.Enter URL and dickgo Click on My Account dropdown Verify login/Sign up popup with below UIelements: a.email text box b.password text box Link button New customer? Create account link Last password? Recovery password link	Login.html	Application should show belowelements: a.email text box b.password text box Login buttonwith orange colour New customer? Createaccount link Last password? Recovery passwordlink	Working as expected
	Functional	Home page	Verify user is able to log into application withValid credentials	I.Enter URL and dickgo Click on My Account dropdown Enter Valid username/email in Email text 4.Enter validpassword in password text box 5. Click On in button	Username:User1@gmail.com password: user@123	User shouldnavigate to predictionhomepage	working as
	Functional	Login page	Verify user is able to log intoapplication with Invalid credentials	1. EnterURL and clickgo 2.Click on My Account button 3.Enter Invalidusername/email in Emailtext box 4.Enter validpassword in password text box 5.Clickon •n button	Username:user1@gmail.com password:user@123	Application shouldshow 'Incorrect emailor password validation message.	working as

LoginPage_TC_004	Functional	Login page	Verify user is able to log intoapplication with Invalid credentials	I-Enter URLand click go Click on Account dropdown Enter Valid username/email in Email text box4.Enter Invalid password in password text box 5.Click on in button	username:user1@gmail.com password:user@123	Application should show •Incorrectemail or password ' validation message.	working as
	Functional	Login page	Verify user is able to into application with Invalid credentials	I.Enter URL and click go Click on My Account dropdown Enter Invalid username/email inEmail text box Enter Invalid password in password textbox Click on Iin button	username:user1@gmail.com password:user@123	Application should show 'Incorrectemail or password ' validation message.	working as
Predictionpage_TC_006_		Prediction Page	Page should display whether the personis drowning or not	1, Camerashould take pictures of peopleswimming in pools 2. It shouldpredict the probability of drowning It should show a bounding box displaying the probability Of drowning	image Of people drowning	generate a alert to lifeguard if people are drowning	Working as

8.2 User Acceptance Testing

Test Case No	Test Case	Testing Status (Yes / No)	Result (Pass / Fail)
01	Libraries and packages	Yes	Pass
02	Code Segments Execution	Yes	Pass
03	Running the application	Yes	Pass
04	Navigating the web pages	Yes	Pass
05	Console working	Yes	Pass
06	Detection	Yes	Pass
07	Drowning Detection	Yes	Pass
08	Alarm Beep	Yes	Pass
09	Cloudant	Yes	Pass

1.Defect Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	2	0	0	2
Client Application	2	0	0	2
Security	1	0	0	1
Outsource Shipping	1	0	0	1
Exception Reporting	2	0	0	2
Final ReportOutput	1	0	0	1

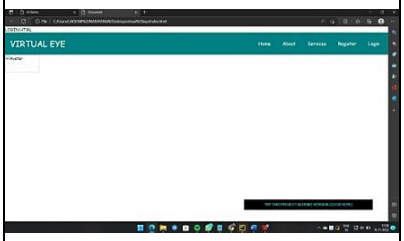
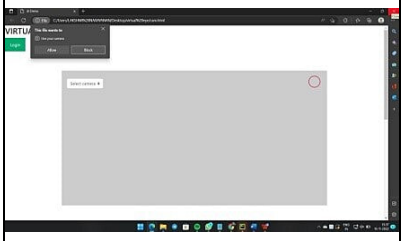
2. Test Case Analysis

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77

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

9. RESULTS

9.1 Performance Metrics

S.No.	Parameter	Values	Screenshot
1.	Model Summary	-	
2.	Accuracy	Training Accuracy - 28 Validation Accuracy - 44	

10. ADVANTAGES & DISADVANTAGES

ADVANTAGES :

a. Avoid injuries and death in the swimming pools.

b. Implementation of AI for detecting.

c. Intimation to life guard.

d. Alarm Beep.

e. Avoid Manual Checking

DISADVANTAGES :

a. Using a camera in the swimming pool.

b. No Privacy

11 . CONCLUSION

This system is mainly used for people to avoid death in swimming pools caused by drowning. This system intimates the lifeguard by using the alarm beep when people are drowning in the swimming pool. The alarm is intimate to the lifeguard. The lifeguard is enter the swimming pool and saves the person from drowning. The alarm is ringing until the lifeguard saves the drowned people.

12. FUTURE SCOPE

In future works maybe we are going to extend some features in this project. We will try to save the drowned people without a manual lifeguard. We will use another algorithm or new algorithm to be used in this project and improve the accuracy, speed, and efficiency of the project. And we will be published the journal of this project in international journals or conferences.

13. APPENDIX

Source Code

Index.html

```
<h1 align="CENTER" style="font-style:italic; color:blue;"></i>Virtual Eye</h1>

<div style ="background-color:rgba(0,0,0,0.5);"nav--items" >

        <h5 style=" margin-left: 20px; "><a a style="color: aqua;"href="{{ url_for('index')}}">Home</a></h5>

    <h5 style="margin-top: -32px; margin-left: 100px;"><a style="color: aqua;"href="{{ url_for('login')}}">Login</a></h5>
```

```
        <h5 style="margin-top: -32px; margin-left: 170px;"><a style="color: aqua;"href="{{ url_for('register')}}">Register</a></h5>

      <h5 style="margin-top: -32px; margin-left: 270px;"><a a style="color: aqua;" href="{{ url_for('login')}}">Demo</a></h5>

</div>


<div style="width:50%;float:left;font-size:2vw;text-align:left;color:white; padding-top:1%">Virtual Eye</div>

<div class="topnav-right" >

<a href="{{ url_for('home')}}">Home</a>

<a href="{{ url_for('login')}}">Login</a>

<a class="active" href="{{ url_for('register')}}">Register</a>

</div>

</div>

<div id="login" class="login">

<form action="{{url_for('afterreg')}}" method="post">

<div class="imgcontainer">

<img src="https://cdn.digitalhealth.net/wp-content/uploads/2017/03/eye_image_generic_555.jpg" alt="Avatar" class="avatar">

</div>

<div class="container">

<input type="text" placeholder="Enter Name" name="name" required><br>

<input type="email" placeholder="Enter Email ID" name="_id" required><br>

<input type="password" placeholder="Enter Password" name="psw" required>

<button type="submit">Register</button><br>

</div>

<div class="container" style="background-color:#f1f1f1">

<div class="psw">Already have an account?&amp;nbsp; &amp;nbsp;<a
```

```
href="{{ url_for('login') }}">Login</a></div >

<div style="width:50%;float:left;font-size:2vw;text-align:left;color:white; padding-top:1%">Virtual Eye</div>

<div class="topnav-right" >

<a href="{{ url_for('home')}}">Home</a>

<a href="{{ url_for('login')}}">Login</a>

<a class="active" href="{{ url_for('register')}}">Register</a>

</div>

</div>

<div id="login" class="login">

<form action="{{url_for('afterreg')}}" method="post">

<div class="imgcontainer">

<imgsrc="https://cdn.digitalhealth.net/wp-content/uploads/2017/03/eye_image_generic_555.jpg" alt="Avatar" class="avatar">

</div><div class="container">

<input type="text" placeholder="Enter Name" name="name" required><br>

<input type="email" placeholder="Enter Email ID" name="_id" required><br>

<input type="password" placeholder="Enter Password" name="psw" required>

<button type="submit">Register</button><br>

</div>

<div class="container" style="background-color:#f1f1f1">

<div class="psw">Already have an account?   <a

href="{{ url_for('login') }}">Login</a></div >
```

Login.html

```
<div id="login" class="login">

<form action="{{url_for('afterlogin')}}" method="post">

<div class="imgcontainer">
```

<img style=""

src="https://cdn.digitalhealth.net/wp-content/uploads/2017/03/eye_image_generic_555.jpg" alt="Avatar" class="avatar">

<input type="email" placeholder="Enter registered email ID" name="_id" required>

<input type="password" placeholder="Enter Password" name="psw" required>

<button type="submit">Login</button>

</form>

Virtual eye

Home

Login

Register

App.py

// Packages and Libraries

import cv2

import os

import numpy as np

from cloudant.client import Cloudant

// Cloud

client = Cloudant.iam('a2aae5ca-a0d0-40c9-aa41-b2fe01527d5f-

```
bluemix','FF_IpVZywe3MAZlo4X5QDT97kQU0vgS-52jqLtFrjh9b', connect = True)
my_database = client.create_database('my_database')
// Flask
app=Flask(__name__)
@app.route('/')
def index():
return render_template('index.html')

@app.route('/index.html')
def home():
return render_template("index.html")
@app.route('/login')
def login():
return render_template('login.html')

@app.route('/afterlogin',methods = ['POST'])
def afterlogin():
user = request.form['_id']
passw = request.form['psw']
print(user,passw)
query = {'_id':{'$eq': user}}



print(len(docs.all()))
if(len(docs.all())==0):
return render_template('login.html', pred = "The username is not found.")
else:
```

```
if((user==docs[0][0]['_id'] and passw==docs[0][0]['psw'])):
return redirect(url_for('prediction'))
else:
print('Invalid User')
@app.route('/logout')
def logout():
return render_template('logout.html')
@app.route('/prediction')
def prediction():
return render_template('prediction.html')
isDrowning = False
while webcam.isOpened():
status, frame = webcam.read()
if not status:
print("Could not read frame")
exit()
bbox, label, conf = cv.detect_common_objects(frame)
if(len(bbox)>0):
bbox0 = bbox[0]
centre = [0,0]
centre = [(bbox0[0]+bbox0[2])/2,(bbox0[1]+bbox0[3])/2]
hmov = abs(centre[0]-centre0[0])
x = time.time()
threshold = 10
if(hmov>threshold or vmov>threshold):
print(x-t0, 's')
t0 = time.time()
```

```
isDrowning = False
else:
print(x-t0, 's')
if((time.time() - t0) > 10):
isDrowning = True
// Beep Sound
if(isDrowning == True):
playsound('alarm.mp3')
webcam.release()
cv2.destroyAllWindows()
return render_template('prediction.html',prediction="Emergency !!! The Person is drowning")
if cv2.waitkey(1) & 0xFF == ord('q'):
break
webcam.release()
cv2.destroyAllWindows()
```

GitHub Link : https://github.com/IBM-EPBL/IBM-Project-17563-1659673413

Video Demo Link : https://youtu.be/Ps0fUag4s00