

# FINAL DELIVERABLE PROJECT

## DOCUMENTATION

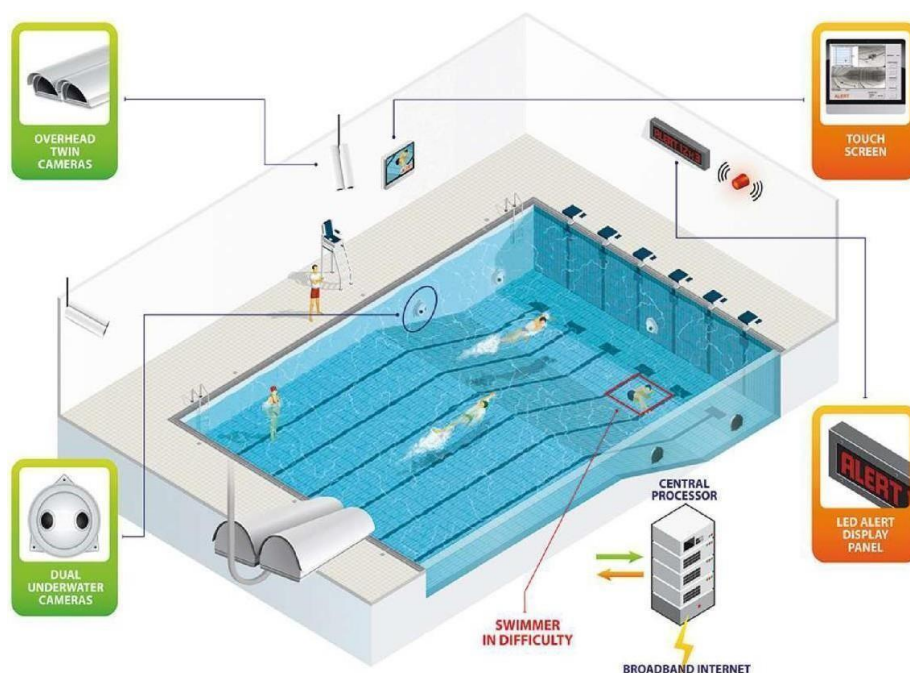
Date	19 November 2022
Team ID	PNT2022TMID06940
Project Name	VirtualEye-Lifeguard for Swimming Pools to Detect the Active Drowning

### 1.INTRODUCTION

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.

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.

#### 1.1 Project Overview



## 2. LITERATURE SURVEY

Of the differing definitions of DDS, most outline three defining elements:

- 1) surveillance,
- 2) detection of a pool user in difficulty, and
- 3) raising an alarm

The presented algorithm for this system is tested on several video sequences recorded in swimming pools in real conditions and the results are of high accuracy with a high capability of tracking individuals in real time. According to the evaluation results, the number of false alarms generated by the system

And mesh lifting system to prevent drowning incidents. The system will contain a mesh which will help the drowning person to lift up in the water, this mesh movement will be controlled by the stepper motors which are connected to the Arduino Nano board, and there will be buzzer that will alert the people near swimming pool and. The drowning person is detected machine learning, this system is used to monitor the swimming pool, track swimmers in that, if any person is in drowning condition raspberry pi will detect it and it will send command to Arduino nano board to lift the mesh up

Swimming pool surveillance systems plays an essential role in safeguarding the premises. In this project differential pressure approach is used for detection of drowning incidents in swimming pools at the earliest possible stage. The children's life is saved during drowning incidents in the swimming pool by lifting the acrylic plate.

### 2.1 Existing Problem

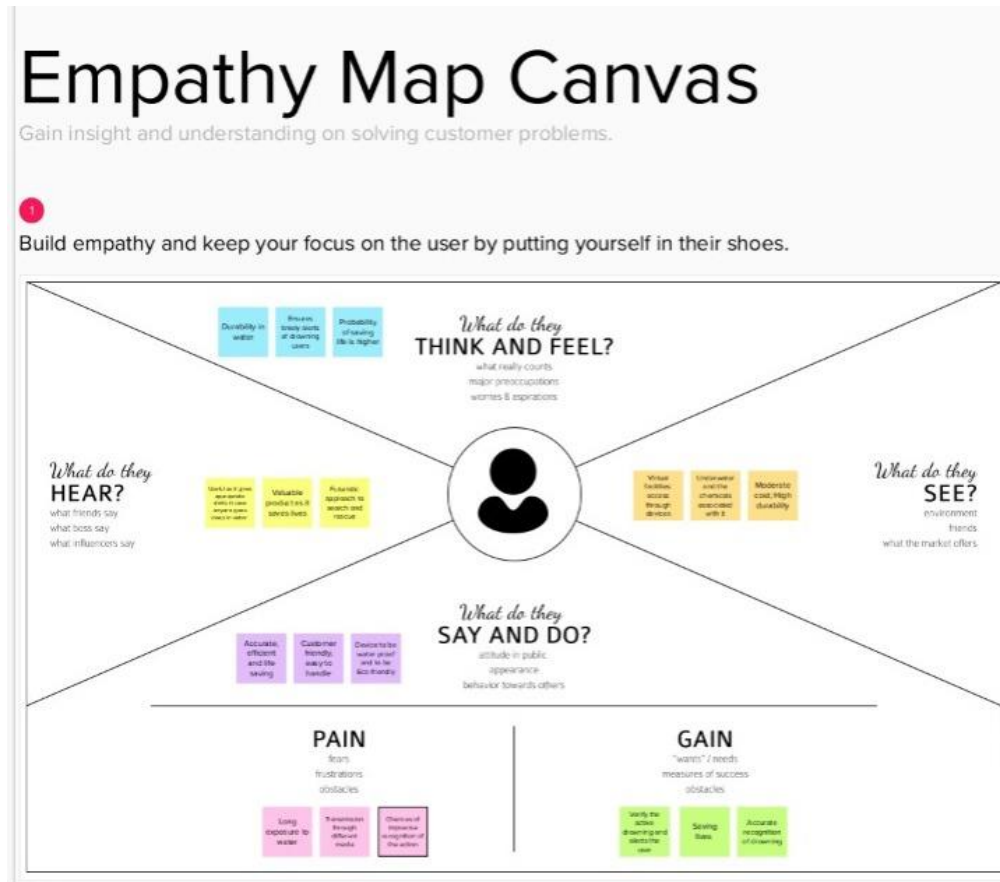
Whilst literature on DDS mostly agrees on areas such as the risks and issues associated with DDS performance, there are other areas where sources offer differing points of view, for example, DDS and their co- existence with lifeguards. There is debate around whether DDS can be helpful or harmful towards lifeguarding practices and how DDS may change the landscape of traditional lifeguarding, as well as some disagreement on whether they serve as justification for reducing lifeguard numbers. The term 'blended lifeguarding' or 'modern lifeguarding' has been newly coined to describe the concept of traditional lifeguarding practices being blended with technology for drowning detection (Swimming Pool Scene, 2017).

## 2.2 References

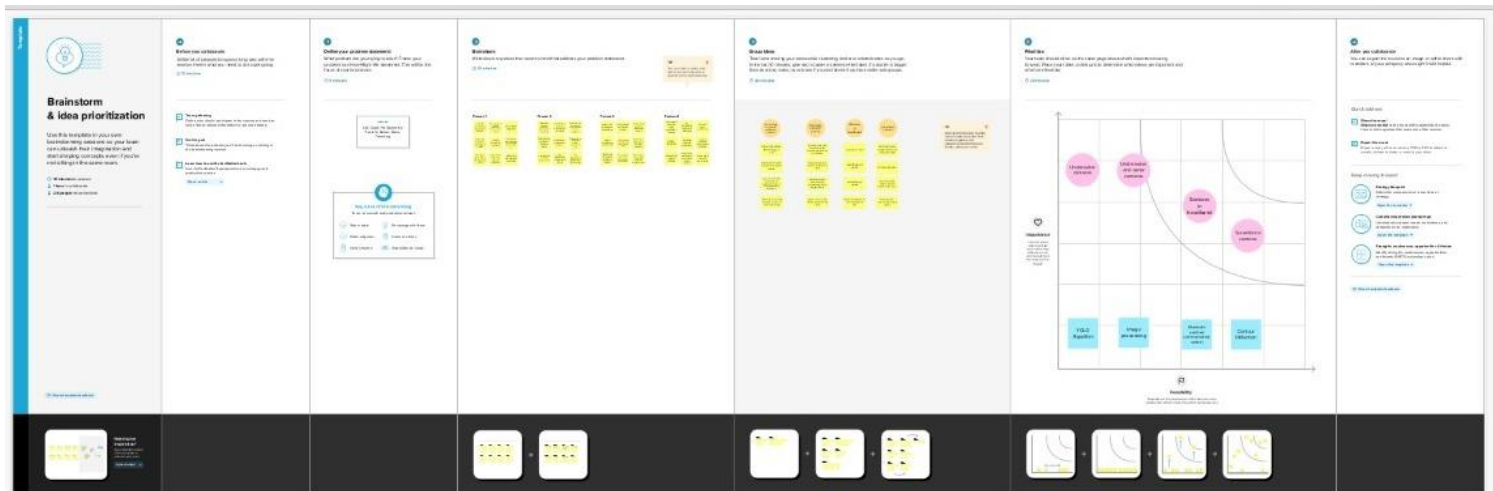
- [1] AngelEye. (2019). AngelEye – Distributors. Retrieved from: <https://www.angeleye.it/news.php?id=28&newscat=10>
- [2] British Standards Institution 1. (2018). BS EN 15288-2, Swimming pools for public use. Safety requirements for operation. Retrieved from: <https://shop.bsigroup.com/ProductDetail/?pid=000000000030360257>
- [3] Drowning Prevention. (2017). The Need. Retrieved from: <https://www.drowningprevention.com.au/>
- [4] German Institute for Standardization. (2019). German national guideline DGfDB R 94.15 “Test methods for camera-based drowning detection systems under operational conditions” (German Association for Public Swimming Pools).
- [5] Health and Safety Executive. (2018). HSG179, Health and safety in swimming pools (Fourth edition).
- [6] ISO (2017) ISO\_20380, First edition, Public swimming pools — Computer vision systems for the detection of drowning accidents in swimming pools — Safety requirements and test methods.

# 3.IDEATION & PROPOSED SOLUTION

## 1.1 EMPATHY MAP CANVAS



## 1.2 IDEATION & BRAINSTORMING



## 1.3 PROPOSED SOLUTION

### Proposed Solution Template:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To detect the active drowning in swimming pools.
2.	Idea / Solution description	Detecting the drowning in the swimming pool and the immediate alert using Convolutional Neural Network technology.
3.	Novelty / Uniqueness	Using image processing and YOLO algorithm for the precise and immediate alerting system.
4.	Social Impact / Customer Satisfaction	To save the lives of regular swimmers. Helps save the lives of children below 6 yrs of age who are prone to drowning.
5.	Business Model (Revenue Model)	This advanced technology can also provide real-time location and image of the danger Which makes rescuing operations easier
6.	Scalability of the Solution	This device can be used in theme parks, rivers, and lakes and specifically swimming pools at home and major efficiency of the device is to send alert signals for lifeguard.

# 1.4 PROBLEM SOLUTION FIT

Project Title: VirtualEye - Life Guard for Swimming Pools to Detect Active Drowning

Project Design Phase-I - Solution Fit Template

Team ID: PNT2022TMID06940

Define CS, fit into CC	<b>1. CUSTOMER SEGMENT(S)</b> <b>CS</b> Children of young age 3 - 12	<b>6. CUSTOMER CONSTRAINTS</b> <b>CC</b> What constraints prevent your customers from taking action or limit their choices of solutions? Lack of technology, budget issues, available devices	<b>5. AVAILABLE SOLUTIONS</b> <b>AS</b> 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?  Swim ring and pool float are alternatives to keep customer floating	Explore AS, differentiate
	<b>2. JOBS-TO-BE-DONE / PROBLEMS</b> <b>J&amp;P</b> Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one, explore different sides. Active drowning can happen anytime, not all customer would be attentive at the time of drowning. Lack of awareness about such technology, need to learn about the know hows of the technology	<b>9. PROBLEM ROOT CAUSE</b> <b>RC</b> What is the real reason that this problem exists? What is the back story behind the need to do this job? Customers have to do it because it is difficult to detect active drowning. Immediate detection will help greatly	<b>7. BEHAVIOUR</b> <b>BE</b> What does your customer do to address the problem and get the job done? Find the proper sensors and algorithms needed, get the surveillance cameras installed, and get it checked, and regular maintenance of the device	
Identify strong TR & EM	<b>3. TRIGGERS</b> <b>TR</b> What triggers customers to act? Reading more about active drowning and realizing same could happen to them, probably had a drowning experience	<b>10. YOUR SOLUTION</b> <b>SL</b> If you are working on an existing business, write down your current solution first, fit in the canvas, and check how much it fits reality. If you are working on a new business proposition then keep it blank until you fit in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behavior.  Surveillance camera with image processing using Convolutional Neural networks and YOLO algorithm	<b>8. CHANNELS of BEHAVIOUR</b> <b>CH</b> <b>8.1 ONLINE</b> What kind of actions do customer take online? Extract online channels from #7 Virtually, customers can monitor the activities in the swimming pool and can take appropriate measures.  <b>8.2 OFFLINE</b> What kind of actions do customer take offline? Extract offline channels from #7 and use them for customer development Being present offline, customers can readily jump in and save them and help them recover from the trauma.	Identify strong TR & EM

## 2. REQUIREMENT ANALYSIS

### 2.1 FUNCTIONAL REQUIREMENT

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Installation	Install the camera inside the underwater, connect necessary app in the phone or other device
FR-2	Detection	Near swimming pool area use detection room for monitor or use IBM cloud for storage purpose of the details
FR-3	Audio	Give the alert signal for the people enter into the underwater and leaving into underwater
FR-4	Support	Extra support from the lifeguard if any person pulse rate will decrease inside the water
FR-5	Prior alert	Extreme level problem should be occurs give the alert signal for the entire pool

### 2.2 NON-FUNCTIONAL REQUIREMENT

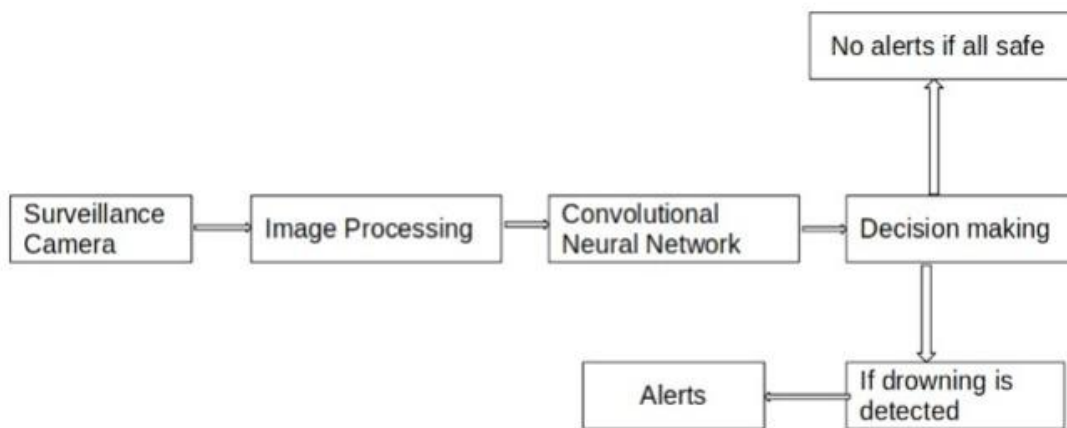
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	A Lifeguard should be present in all the time near pool
NFR-2	Security	Alert message or signal should be give by the lifeguard of swimmer
NFR-3	Reliability	Triggers if any immediate needs of the swimmer inside the pool
NFR-4	Performance	If any unwanted position changes and the pulse rate will decrease this will detect it.
NFR-5	Availability	Equipment and other requirement should be checked by the lifeguards
NFR-6	Scalability	Virtual eye lifeguard detects potential drownings and it should be notifies you.

### 3. PROJECT DESIGN

#### 3.1 DATAFLOW 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 stored

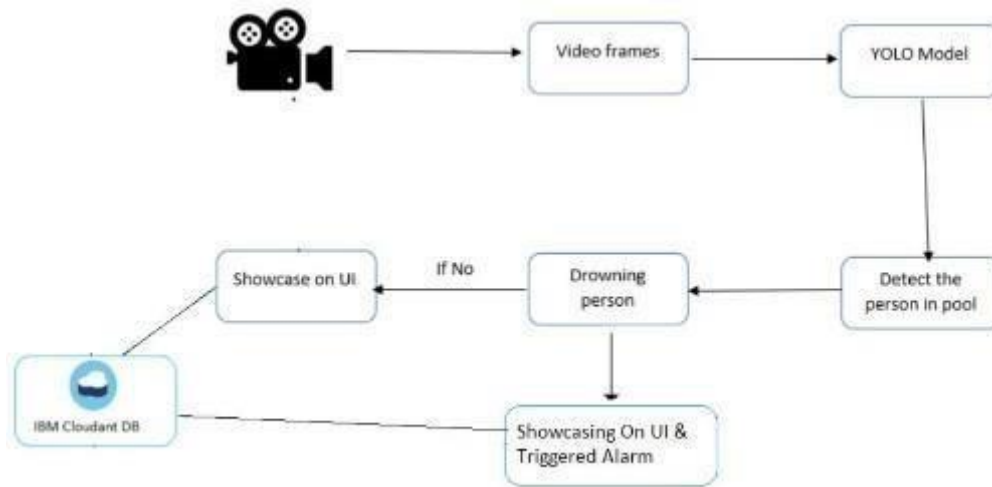


#### 3.2 SOLUTION & TECHNICAL ARCHITECTURE

##### Solution Architecture:

- ❖ To find underwater movement while person is drowning they have any Problem or anything else we will find the solution using the Artificial Intelligence (AI) detection technology.
- ❖ Usually, such systems can be developed by installing more than 16 cameras underwater and ceiling and analyzing the video feeds to detect any anomalies. 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.





### 3.3 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 (Pool owner)	Installation	USN-1	Install the camera inside the underwater, connect necessary app in the phone or other device	I can cameras to the IBM cloudDB	High	Sprint-1
Customer (Lifeguard)	Secure the people	USN-2	As a user, I can secure the drowning persons from the pool	I can save the drowning person	High	Sprint-1
Customer (swimmers)	safety	USN-3	As a user, I can swim inside the underwater without fear of the Drowning	I can swim safely	medium	Sprint-2
Customer care (Executive)	Contact	USN-4	As a user, I Can resolve if any problem occurs with any device technically	I can contact the customer care executive to resolve any issues	Medium	Sprint-3
Administrator	Dashboard	USN-5	Management of the drowning detection system and database management	I can access the system's logs and any other data instantly	High	Sprint-4

## 6.1 SPRINT PLANNING & ESTIMATION

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

Sprint-1	Login	VLGFSP - 6	As a user, I can log into the application by entering email & password	1	High	Sneha K, Suvetha S, Shakthipriya M, Sanfiya Banu M
Sprint-2	Dataset Collect	VLGFSP - 11	Collect number of datasets and get accuracy	2	Medium	Sneha K, Suvetha S, Shakthipriya M, Sanfiya Banu M
Sprint-2	Pre-processing	VLGFSP - 12	The dataset is extracted	2	High	Sneha K, Suvetha S, Shakthipriya M, Sanfiya Banu M
Sprint-2	Train the model	VLGFSP - 13	Train the model.	4	High	Sneha K, Suvetha S, Shakthipriya M, Sanfiya Banu M
Sprint-2	Test the model	VLGFSP - 14	Test the model	6	High	Sneha K, Suvetha S, Shakthipriya M, Sanfiya Banu M
Sprint-3	Detection	VLGFSP - 15	Load the trained model.	3	High	Sneha K, Suvetha S, Shakthipriya M, Sanfiya Banu M
Sprint-3	Detection	VLGFSP - 16	Identify the person by collecting real-time data through a webcam.	5	Medium	Sneha K, Suvetha S, Shakthipriya M, Sanfiya Banu M
Sprint-3	Detection	VLGFSP - 16	classify it by using a trained model to predict the output	8	High	Sneha K, Suvetha S, Shakthipriya M, Sanfiya Banu M
Sprint-4	Detection	VLGFSP - 17	If person is drowning, the system will ring an alarm to give signal	7	High	Sneha K, Suvetha S, Shakthipriya M, Sanfiya Banu M
Sprint-4	Detection	VLGFSP - 18	As a User, I can detect the drowning person.	3	Medium	Sneha K, Suvetha S, Shakthipriya M, Sanfiya Banu M
Sprint-4	Logout	VLGFSP - 19	As a User, I can logout the application.	2	Low	Sneha K, Suvetha S, Shakthipriya M, Sanfiya Banu M

## 7. CODING & SOLUTION

### 7.1 FEATURE 1

Importtime

```
import cv2
import numpy as np
from cloudant.client import Cloudant
from flask import Flask, request, render_template, redirect, url_for
from playsound import playsound
import cvlib as cv
from cvlib.object_detection import draw_bbox
# Loading the model
# Authenticate using an IAM API key
client = Cloudant.iam('5a1ffd26-d995-410e-af77-546fb6498fd8-
bluemix','5rUgrIfVeYtIyTqJ0hAvTOvIYvJDNIklr-sDUHQRClnN', connect=True)
# Create a database using an initialized client
my_database = client.create_database('my_database')
app=Flask(__name__)
#default home page or route
@app.route('/')
def index():
    return render_template('index.html')
@app.route('/index.html')
def home():
    return render_template("index.html")
#registration page
@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], # Setting _id is optional
        '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)
    #response = requests.get(url)
    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")
#login page
@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('drowning.mp4')
    if not webcam.isOpened():
        print("Could not open webcam")

```

```

exit()

t0 = time.time() #gives time in seconds after 1970
#variable dcount stands for how many seconds the person has been standing
still for
centre0 = np.zeros(2)
isDrowning = False

#this loop happens approximately every 1 second, so if a person doesn't
move,
#or moves very little for 10seconds, we can say they are drowning
#loop through frames
while webcam.isOpened():
# read frame from webcam
status, frame = webcam.read()
if not status:
print("Could not read frame")
exit()
# apply object detection
bbox, label, conf = cv.detect_common_objects(frame)
#simplifying for only 1 person
#s = (len(bbox), 2)
if(len(bbox)>0):
bbox0 = bbox[0]
#centre = np.zeros(s)
centre = [0,0]
#for i in range(0, len(bbox)):
#centre[i]
=[(bbox[i][0]+bbox[i][2])/2,(bbox[i][1]+bbox[i][3])/2 ]
centre = [(bbox0[0]+bbox0[2])/2,(bbox0[1]+bbox0[3])/2 ]
#make vertical and horizontal movement variables
hmov = abs(centre[0]-centre0[0])
vmov = abs(centre[1]-centre0[1])
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
#print('bounding box: ', bbox, 'label: ' label , 'confidence: '
conf[0], 'centre: ', centre)
#print(bbox,label ,conf, centre)
print('bbox: ', bbox, 'centre:', centre, 'centre0:', centre0)
print('Is he drowning: ', isDrowning)
centre0 = centre

```

```

# draw bounding box over detected objects
out = draw_bbox(frame, bbox, label, conf,isDrowning)
#print('Seconds since last epoch: ', time.time()-t0)
# display output
cv2.imshow("Real-time object detection", out)
if(isDrowning == True):
    playsound('alarm.mp3')
    webcam.release()
    cv2.destroyAllWindows()
    return render_template('prediction.html',prediction="Emergency
!!! The Person is drowining")
#return render_template('base.html')
# press "Q" to stop
if cv2.waitKey(1) & 0xFF == ord('q'):
    break
# release resources
    webcam.release()
    cv2.destroyAllWindows()
    #return render_template('prediction.html',)
""" Running our application """
if __name__ == "__main__":
    app.run(debug=True)

```

## 7.2 FEATURE 2

```

# -*-
coding:

```

utf-8 -

\*\_

```
"""Untitled0.ipynb
Automatically generated by Colaboratory.
Original file is located at
https://colab.research.google.com/drive/1vfvfVXSHLLkGIWyc9tUcW2oNno9AFwKZ
"""

import cv2
import torch
from tqdm.auto import tqdm

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model = (
    torch.hub.load("ultralytics/yolov5", "yolov5s", pretrained=True).eval().to(device)
)

model.conf = 0.35

def detect(source_path, num_track_seconds=5):

    cap = cv2.VideoCapture(source_path)
    FPS = cap.get(cv2.CAP_PROP_FPS)
    total_frames = cap.get(cv2.CAP_PROP_FRAME_COUNT)
    print("FPS: ", FPS)
    print("Total Frames: ", total_frames)

    # imageWidth = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
    # imageHeight = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))

    # save_filename = source_path.split(".")[0] + "_result.mp4"
    # writer = cv2.VideoWriter(
    #     save_filename,
    #     cv2.VideoWriter_fourcc("m", "p", "4", "v"),
    #     FPS,
    #     (imageWidth, imageHeight),
    # )

    prev_center = None
    not_moving_frame_count = 0
    is_drowning = False
    for frame_num in tqdm(range(int(total_frames))):
        success, frame = cap.read()
        if success:
            with torch.inference_mode():
                results = model(frame)

                xyxys = results.xyxy[0].cpu().numpy()
                for xyxy in xyxys:
```

```

center = ((xyxy[0] + xyxy[2]) // 2, (xyxy[1] + xyxy[3]) // 2)
# check if the detected object is a person
if xyxy[-1] == 0 and prev_center is not None:
    # check for no movement
    if (
        abs(prev_center[0] - center[0]) < 20
        and abs(prev_center[1] - center[1]) < 20
    ):
        not_moving_frame_count += 1

prev_center = center

bbox, conf, class_id = xyxy[:4].astype(int), xyxy[4] * 100, xyxy[5]

if not_moving_frame_count >= (num_track_seconds * FPS):
    color = (0, 0, 255)
    frame = cv2.putText(
        frame,
        "Drowning: Yes",
        (80, 50),
        cv2.FONT_HERSHEY_DUPLEX,
        1,
        color,
        2,
        cv2.LINE_AA,
    )
    is_drowning = True

else:
    color = (0, 255, 0)
    frame = cv2.putText(
        frame,
        "Drowning: No",
        (80, 50),
        cv2.FONT_HERSHEY_DUPLEX,
        1,
        color,
        2,
        cv2.LINE_AA,
    )
out_frame = cv2.rectangle(frame, bbox[:2], bbox[2:], color, 2)
out_frame = cv2.putText(
    out_frame,
    f"conf: {conf:.2f}",
    bbox[:2],
    cv2.FONT_HERSHEY_DUPLEX,
    0.6,
    color,
    2,
    cv2.LINE_AA,

```



```

    )
    center_pt = list(map(int, center))
    out_frame = cv2.circle(out_frame, center_pt, 3, color, -1)

    ret, buffer = cv2.imencode(".jpg", out_frame)
    out_frame = buffer.tobytes()

    yield (
        b"--frame\r\n"
        b"Content-Type: image/jpeg\r\n\r\n" + out_frame + b"\r\n"
    )

#         # writer.write(frame)
#         cv2.imshow("Real-time object detection", out_frame)
#         if is_drowning == True:
#             cap.release()
#             cv2.destroyAllWindows()

#         # press "Q" to stop
#         if cv2.waitKey(1) & 0xFF == ord("q"):
#             break

# # release resources
# cap.release()
# cv2.destroyAllWindows()

if __name__ == "__main__":
    detect("swim.mp4")
    detect("standby.mp4")

```

## CHAPTER-8

### 8. TESTING

#### 8.1 TEST CASES

Test case ID	Feature Type		Test Scenario	Steps TO Execute	Test	Expected Result	Actual Result
LoginPage_TC_001	Functional	Home Page	Verify user is able to see the Login/Signup popup when user clicked on My account button	1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Signup popup displayed or not	Login.html	Login/Signup popup should display	Working as
LoginPage_TC_002		Home Page	Verify the UI elements in Login/Signup popup	1.Enter URL and click go 2.Click on My Account dropdown 3.Verify login/Signup popup with below UI elements: a.email text box b.password text box c. Login button d.New customer? Create account link e. Last password? Recovery password link	Login.html	Application should show below elements: a.email text box b.password text box c.Login button with orange colour d. New customer? Create account link e.Last password? Recovery password link	Working as expected
LoginPage_TC_003	Functional	Home page	Verify user is able to log into application with Valid credentials	1.Enter URL and click go 2.Click on My Account dropdown 3.Enter Valid username/email in Email text 4.Enter valid password in password text box 5. Click On in button	Username:lax@gmail password: lax26	User should navigate to prediction homepage	working as
LoginPage_TC_004	Functional	Login page	Verify user is able to log into application with Invalid credentials	1. Enter URL and click go 2.Click on My Account dropdown button 3.Enter Invalid username/email in Email text box 4.Enter valid password in password text box 5.Click on in button	Username:lax password:lax26	Application should show 'incorrect email or password' validation message.	working as
LoginPage_TC_004	Functional	Login page	Verify user is able to log into application with Invalid credentials	1-Enter URL and click go 2.Click On My Account dropdown 3.Enter Valid username/email in Email text box 4.Enter Invalid password in password text box 5.Click on in button	username:lax26@mail password:lax26	Application should show 'incorrect email or password' validation message.	working as
LoginPage_TC_005	Functional	Login page	Verify user is able to into application with Invalid credentials	1.Enter URL and click go 2.Click on My Account dropdown 3.Enter Invalid username/email in Email text box 4. Enter Invalid password in password text box 5. Click on in button	username:lax26@mail password:1803	Application should show 'incorrect email or password' validation message.	working as
Predictionpage_TC_006	Functional	Prediction Page	Page should display whether the person is drowning or not	1. Camera should take pictures of people swimming in pools. 2. It should predict the probability of drowning 3. 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

## CHAPTER-9

### 9.RESULT

#### 9.1 PERFORMANCE METRICS

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
```

```
  <title>High Quality Facial Recognition</title>
```

```
  <link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css" rel="stylesheet">
```

```
  <script src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js">
```

```
</script>
```

```
  <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js">
```

```
</script>
```

```
  <script src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js">
```

```
</script>
```

```

<link href="{{ url_for('static', filename='css/main.css') }}" rel="stylesheet">
    <style>
        .bg-dark {
            background-color: #42678c!important;
        }
        #result {
            color: #0a1c4ed1;
        }
    </style>
</head>

```

```

<body style="background-color:black;">
<header id="head" class="header">
    <section id="navbar">
        <h1 class="nav-heading"><i>Virtual Eye</i></h1>
        <div class="nav--items">
            <ul>
                <li><a href="{{ url_for('index') }}">Home</a></li>
                <li><a
href="{{ url_for('logout') }}">Logout</a></li>
                <!-- <li><a href="#about">About</a></li>
                <li><a href="#services">Services</a></li> -->

            </ul>
        </div>
    </section>
    </header>
    <div class="container">
        <div id="content" style="margin-top:2em">
            <div class="container">
                <div class="row">
                    <div class="col-sm-6 bd" >

```

`<h2><em style="color:white;">High Quality Facial  
Recognition</em></h2>`

`<br>`

`<p><h5><i style="color:white;">Emotion Detection Through  
Facial Feature Recognition</i></h5></p>`

``

`</div>`

`<div class="col-sm-6">`

`<div>`

`<h4 style="color:white;">Upload  
Image Here</h4>`

`<form action = "http://localhost:5000/" id="upload-file"  
method="post" enctype="multipart/form-data">`

`<label for="imageUpload" class="upload-  
label">`

`Choose Image`

`</label>`

`<input type="file" name="image"  
id="imageUpload" accept=".png, .jpg, .jpeg, .pdf">  
</form>`

`<div class="image-section" style="display:none;">`

`<div class="img-preview">`

`<div id="imagePreview">`

`</div>`

`</div>`

`<div>`

`<button type="button" class="btn btn-info btn-lg "  
id="btn-predict">Analyse</button>`

</div>

</div>

<div class="loader" style="display:none;"></div>

<h3>

<span id="result"> </span>

</h3>

</div>

</div>

</div>

</div>

</div>

</div>

</body>

<footer>

<script src="{ { url\_for('static', filename='js/main.js') }}"  
type="text/javascript"></script>

</footer>

</html>

## Index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<!--Bootstrap -->

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl" crossorigin="anonymous"></script>

<script src="https://kit.fontawesome.com/8b9cdc2059.js" crossorigin="anonymous"></script>

<link href="https://fonts.googleapis.com/css2?family=Akronim&family=Roboto&display=swap" rel="stylesheet">

<link rel="stylesheet" href="../static/style.css">

<!-- <script defer src="../static/js/main.js"></script> -->

<title>Virtual Eye</title>

```

</head>
<body>
    <header id="head" class="header">
    <section id="navbar">
        <h1 class="nav-heading"></i>Virtual Eye</h1>
        <div class="nav--items">
            <ul>
                <li><a
href="{{ url_for('index') }}">Home</a></li>
                <li><a
href="{{ url_for('login') }}">Login</a></li>
                <li><a
href="{{ url_for('register') }}">Register</a></li>
                <li><a href="{{ url_for('login') }}">Demo</a></li>
            </ul>
        </div>
    </section>
    <section id="slider">
    <div id="carouselExampleIndicators" class="carousel" data-ride="carousel">
        <ol class="carousel-indicators">
            <li data-target="#carouselExampleIndicators" data-slide- to="0"
class="active "></li>
            <li data-target="#carouselExampleIndicators" data-slide-to="1"></li>
            <li data-target="#carouselExampleIndicators" data-slide-to="2"></li>
        </ol>
        <div class="carousel-inner">

            <div class="carousel-item active">
            

```



```
        </div>
        <div class="carousel-item">
          
        </div>
        <div class="carousel-item">
          
        </div>
      </div>
      <a class="carousel-control-prev" href="#carouselExampleIndicators"
role="button" data-slide="prev">
        <span class="carousel-control-prev-icon" aria-
hidden="true"></span>
        <span class="sr-only">Previous</span>
      </a>
      <a class="carousel-control-next" href="#carouselExampleIndicators"
role="button" data-slide="next">
        <span class="carousel-control-next-icon" aria-
hidden="true"></span>
        <span class="sr-only">Next</span>
      </a>
    </div>
```

```
  </section>
</header>
<section id="about">
  <div class="top">
    <h3 class="title text-muted">
      ABOUT PROJECT
    </h3>
```

<div class="line"></div>

</div>

<div class="body">

<div class="left">

<h2>Problem:</h2>

<p>

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 the hotels, weekend tourist spots and barely people have in their house backyard. Beginners, especially often feel it difficult to breathe under water and causes breathing trouble which in turn cause 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.

</p>

</div>

<div class="left">

<h2>Solution:</h2>

<p>

To overcome the conflict, a meticulous system is to be implemented along the swimming pools to save the human life. By studying body movement patterns and connecting cameras to an artificial intelligence (AI) system 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 analysing 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 thanan alert will be generated to attract lifeguards attention.

</p>

</div>

</div>

<div class="bottom">

<p ><b>

Note : The system is not designed to replace a lifeguard or other human monitor, but to act as an additional tool. It helpsthe lifeguard to detect the underwater situation where they canâ€™t easily observe.

</b></p>

</div>

</section>

<section id="footer">

<p>Copyright Â© 2022. All Rights Reserved</p>

<div class="social">

<a href="#" target="\_blank"><i class="fab fa-2x fa-twitter-square"></i></a>

<a href="#" target="\_blank">

<i class="fab fa-2x fa-linkedin"></i></a>

<a href="#">

<i class="#"></i>

</a>

</div>

</section>

</body>

</html>

## Logout.html

```
<!DOCTYPE html>
<html >

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Virtual Eye</title>
    <link                href='https://fonts.googleapis.com/css?family=Pacifico'
rel='stylesheet' type='text/css'>
    <link                href='https://fonts.googleapis.com/css?family=Arimo'
rel='stylesheet' type='text/css'>
    <link                href='https://fonts.googleapis.com/css?family=Hind:300'    rel='stylesheet'
type='text/css'>
    <link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300'
rel='stylesheet' type='text/css'>

    <link                href='https://fonts.googleapis.com/css?family=Merriweather'
rel='stylesheet'>
    <link                href='https://fonts.googleapis.com/css?family=Josefin      Sans'
rel='stylesheet'>
    <link                href='https://fonts.googleapis.com/css?family=Montserrat'
rel='stylesheet'>

<style>
.header {
                top:0; margin:0px;
```

```

        left: 0px;
        right: 0px;
        position: fixed;
        background-color: #28272c;
        color: white;
        box-shadow: 0px 8px 4px grey;
        overflow: hidden;
        padding-left: 20px;
        font-family: 'Josefin Sans';
        font-size: 2vw;
        width: 100%;
        height: 8%;
        text-align: center;
    }
    .topnav {
        overflow: hidden;
        background-color: #333;
    }

    .topnav-right a {
        float: left; color:
        #f2f2f2;
        text-align: center;
        padding: 14px 16px; text-
        decoration: none; font-
        size: 18px;
    }

    .topnav-right a:hover {
        background-color: #ddd;
        color: black;
    }

```

```

    .topnav-right a.active {
        background-color: #565961;
        color: white;
    }

    .topnav-right {
        float: right;
        padding-right: 100px;
    }

    .login{
        margin-top: -70px;
    }
    body {

        background-color: #ffffff;
        background-repeat: no-repeat;
        background-size: cover;
        background-position: 0px 0px;
    }
    .main{
        margin-top: 100px;
        text-align: center;
    }
    form { margin-left: 400px; margin-right: 400px; }

    input[type=text], input[type=email], input[type=number], input[type=password] {
        width: 100%;
        padding: 12px 20px;
        display: inline-block;
        margin-bottom: 18px;
        border: 1px solid #ccc;
    }

```

```
        box-sizing: border-box;
    }

    button {
        background-color: #28272c;
        color: white;
        padding: 14px 20px;
        margin-bottom: 8px;
        border: none; cursor:
        pointer; width: 20%;
    }

    button:hover {
        opacity: 0.8;
    }

    .cancelbtn {
        width: auto;
        padding: 10px 18px;
        background-color: #f44336;
    }

    .imgcontainer { text-
        align: center;
        margin: 24px 0 12px 0;
    }

    img.avatar {
        width: 30%;
        border-radius: 50%;
    }
```

```

        .container      {
            padding: 16px;
        }

        span.psw      {
            float: right;
            padding-top: 16px;
        }

/* Change styles for span and cancel button on extra small screens
*/
        @media screen and (max-width: 300px) {
            span.psw {
                display: block;
                float: none;
            }
            .cancelbtn {
                width: 100%;
            }
        }

</style>
</head>

<body style="font-family:Montserrat;">

<div class="header">
    <div style="width:50%;float:left;font-size:2vw;text-align:left;color:white; padding-top:1%">Virtual eye</div>
    <div class="topnav-right" style="padding-top:0.5%;">

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

```



```

        <a href="{{ url_for('login')}}">Login</a>
        <a href="{{ url_for('register')}}">Register</a>
    </div>
</div>
<div class="main">
<h1>Successfully Logged Out!</h1>
<h3 style="color:#4CAF50">Login for more information</h3>

        <a href="{{ url_for('login')}}"><button
type="submit">Login</button></a>
</form>
</div>

</body>
</html>

```

## Prediction.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!--Bootstrap -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/boo
tstrap.min.css" integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGg
FAW/dAiS6JXm" crossorigin="anonymous">
    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
integrity="sha384-

```

```
KJ3o2DKtIkVYIK3UENzmM7KCKRr/rE9/Qpg6aAZGJwFDMVNA/GpG
FF93hXpG5KkN" crossorigin="anonymous"></script>
  <script      src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/
popper.min.js"      integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPsk
vXusvfa0b4Q" crossorigin="anonymous"></script>
  <script      src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootst
rap.min.js"      integrity="sha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5
+76PVCmYl" crossorigin="anonymous"></script>
```

```
  <script      src="https://kit.fontawesome.com/8b9cdc2059.js"
crossorigin="anonymous"></script>
  <link      href="https://fonts.googleapis.com/css2?family=Akronim&family=
Roboto&display=swap" rel="stylesheet">
  <link rel="stylesheet" href="../static/style.css">
```

```
  <script defer src="../static/js/JScript.js"></script>
  <title>Prediction</title>
</head>
<body>
  <header id="head" class="header">
    <section id="navbar">
      <h1 class="nav-heading"><i>Virtual Eye</i></h1>
      <div class="nav--items">
        <ul>
          <li><a href="{{ url_for('index') }}">Home</a></li>
```

```

        <li><a
href="{{ url_for('logout')}}">Logout</a></li>
        <!-- <li><a href="#about">About</a></li>
        <li><a href="#services">Services</a></li> -->

</ul>
</div>
</section>
</header>
<!-- dataset/Training/metal/metal326.jpg -->
</br>
<section id="prediction">
<h2 class="title text-muted">Virtual Eye- Life Guard forSwimming Pools to
Detect Active Drowning</h1>
<div class="line" style="width: 900px;"></div>
</section>
</br>
<section id="about">

```

```

<div class="body">
<div class="left">
    <p>

```

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 the hotels, weekend tourist spots and barely people have in their house backyard. Beginners, especially often feel it difficult to breathe under water and causes breathing trouble which in turn cause 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.

```

        </p>
</div>
<div class="left">

    <div class="prediction-input">
    
        <br>
        <form    id="form"    action="/result"    method="post"
enctype="multipart/form-data">

            <input type="submit" class="submitbtn" value="ClickMe! For a
Demo">

                </form>
            </div>
            <h5 style="text-color:Red">
                <b style="text-color:Red">{{ prediction }}<b>
            </h5>
        </div>
    </div>
</section>

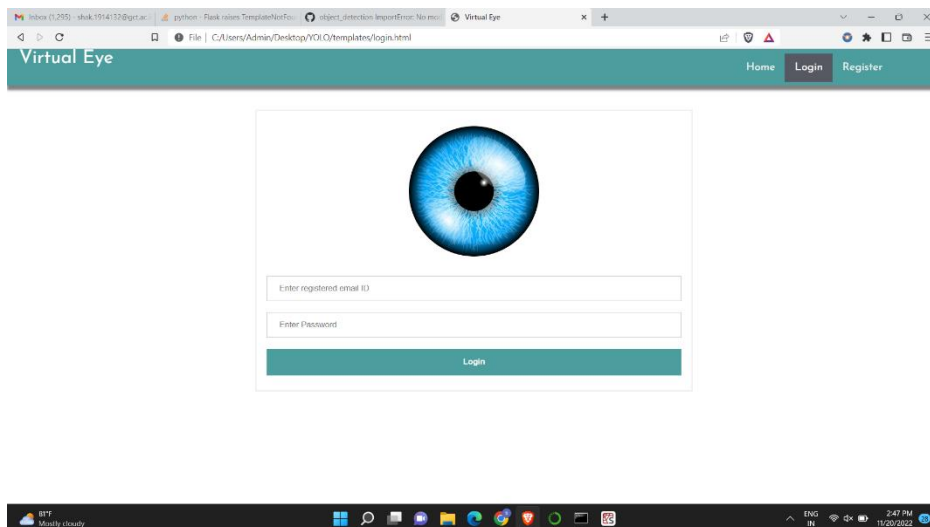
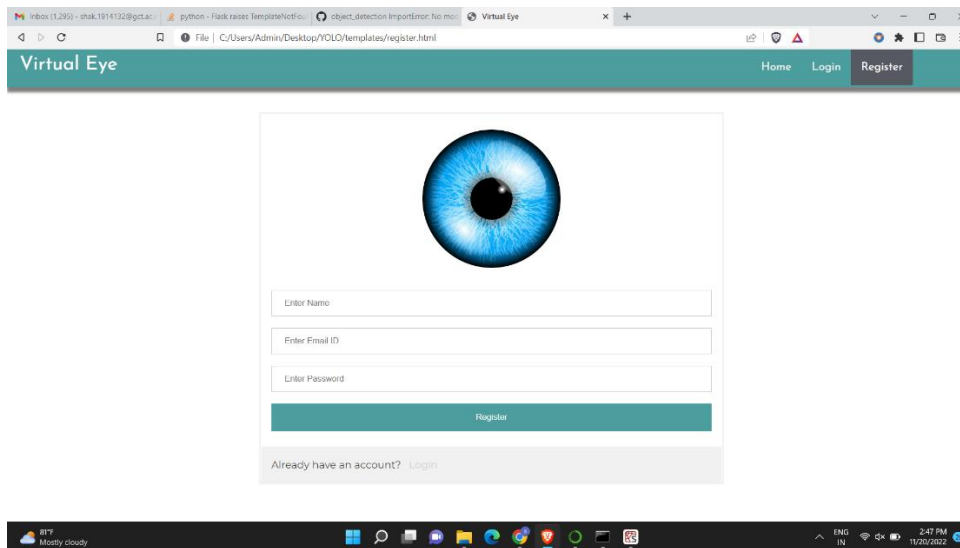
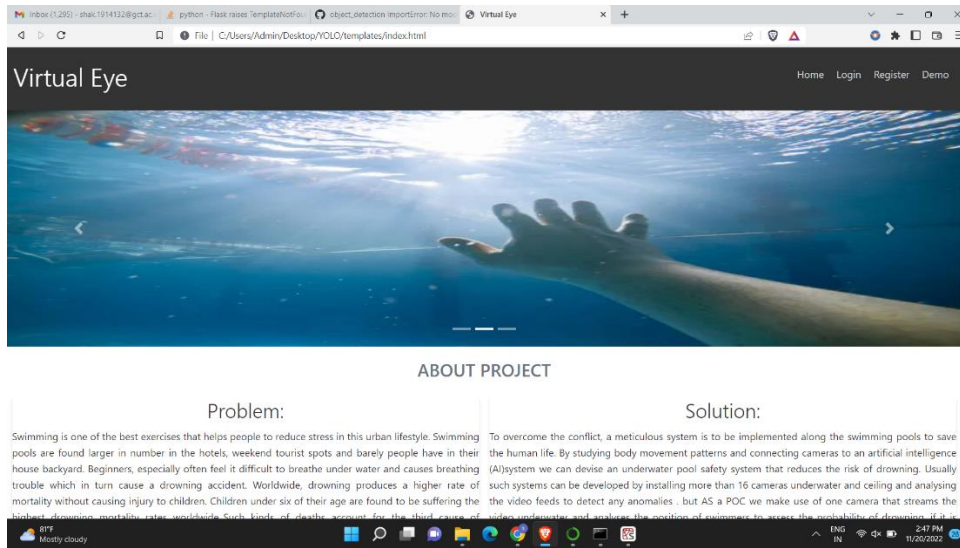
    <br><br>

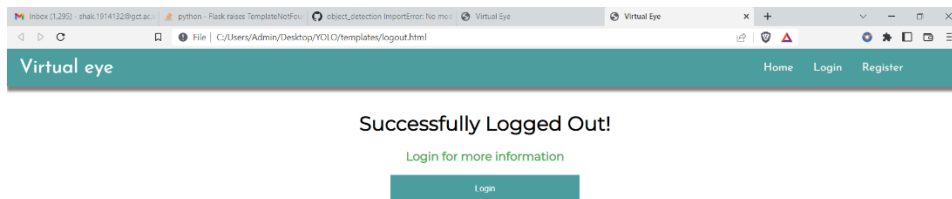
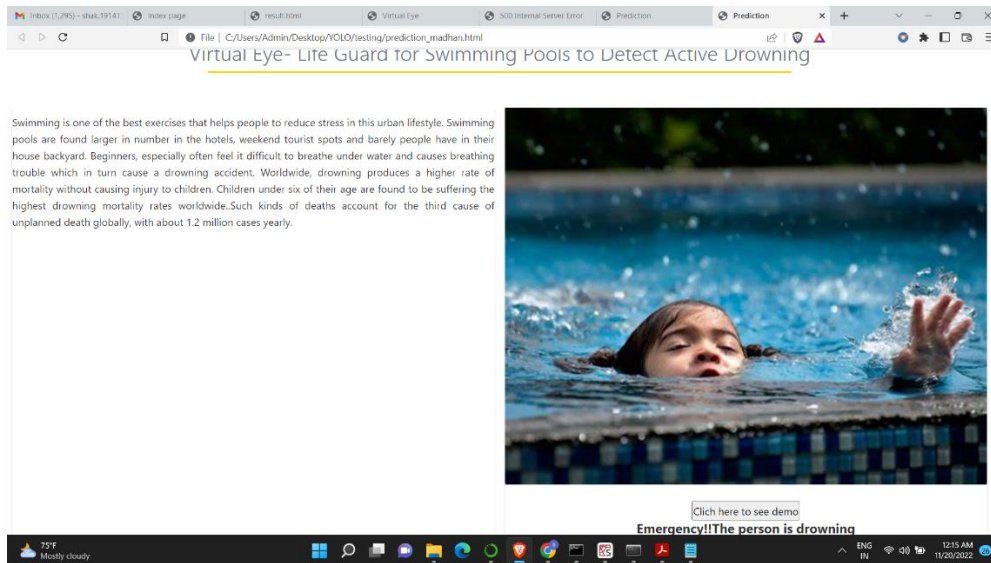
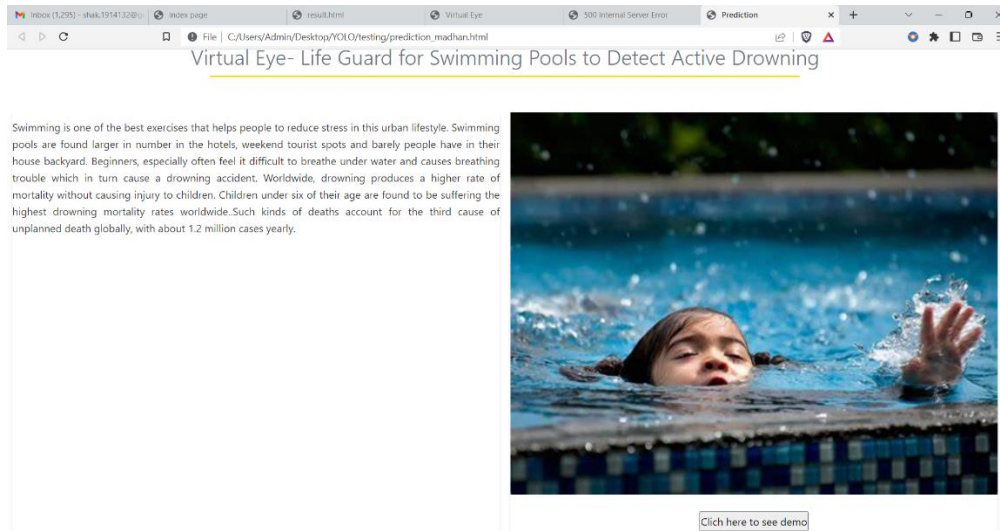
    <section id="footer">
        <p>Copyright Â© 2021. All Rights Reserved</p>

    </section>
</body>

</html>

```





## **10. ADVANTAGES & DISADVANTAGES**

### **\*ADVANTAGES:**

- (i) user feel comfortable and more secure
- (ii) Children, adult, pet animal , old age people are used
- (iii) spending more time for family, freedom for safety guards near the Swimming pool
- (iv) Swimmers, resort are gain in the financial
- (v) drowning should be monitored

### **\*DISADVANTAGE:**

- (i) For uneducated people will suffer from this technology
- (ii) Electricity will be required
- (iii) Software and hardware requirement will need

## **CHAPTER-11**

### **11. CONCLUSION**

This section will draw from three core documents: ISO\_20380, HSG179, and the recently published German guideline, DGfdB R 94.15. A summary of each is given, outlining the key messages they disseminate and what this means for those involved with DDS.

ISO\_20380 This document focuses on the requirements for the installation, operation, maintenance and performance of DDS, the testing methods, and the information required from the supplier in the operating manual. These international standards do not apply to systems used in domestic pools or pools smaller than 150m<sup>2</sup>.

Prior to the installation of any DDS, 'a technical study shall be carried out by the supplier in consultation with or based on information provided by the swimming pool's owner/operator'. This is to establish the quantity and positioning of the equipment making up the system such as cameras, central processing unit, alarm tools, and other related equipment. The technical study must also provide a technical drawing of the pool basin, showing areas of 'coverage' and 'non-coverage', as well as the minimum lighting levels required above and below the water surface for the DDS to operate within performance requirements. To carry out the study, a list of factors to consider are given, outlining the variables that make each pool unique such as the architecture, and alarm reception coverage area of mobile devices to be used with the system. With this information all in one document, the technical study can be used to help optimise performance of the system, and forms part of the contract between the supplier and the pool operator. The next area of the standard is the performance requirements. This outlines the requirements needed to pass the regular maintenance testing and performance requirements for normal operation. This section covers the alarm set off time for operational performance, which is to be 15 seconds or less and displayed on the system interface. It also states that the alarm set off time must be built-in and shall not be changeable by staff. The section also discusses the areas covered by the DDS and highlights that each trained staff member must be aware of these areas. Another coverage-related requirement is that the DDS must be able to temporarily create areas where detection is disabled, to manage specific activities such as rescue drills.



## **FUTURE SCOPE:**

This lifeguard system consists of three main components, i.e., the drowning detection, the rescuing drone, and the hazardous activity detection. All three components combined will create a system capable of detecting drowning victims, dispatching an inflatable tube using a drone (as depicted in Fig.9) and detecting hazardous activities—eventually becoming an entity that could assist a lifeguard. The system is accessible to its primary user, presumably a pool owner or a lifeguard, in the form of an interface with a sound alarm and an android mobile service that holds the capabilities of receiving Firebase notifications. 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 is becoming a state-of-the-art.

This limitation could be omitted with the use of multiple cameras that could be placed over the premises in several ground coordinates, increasing the accuracy of the computer vision algorithms. Moreover, due to the inability to fly a drone in extreme weather conditions such as rain, strong winds or lightning, the system is limited to be used under few specifications. As swimming in extreme weather conditions is not preferred either, the system could be further improved to emit a warning signal if a person was to swim in any of the above weather conditions, bypassing the need to fly the drone. Additionally, all the processing is done on the client side of the applications on the Jetson Nano board, preventing any security and privacy issues that might arise due to the sensitive information inputted through the cameras. For future developments convenience wise, the system could benefit by having an additional set of cameras to identify and verify a drowning or a hazardous activity on the premises. Accessibility could also be improved by extending the Android service to be an application both in Android and iOS platforms that could hold the details of each premise individually, making a centralized system that watches over the decentralized pool premises. Both drown and hazardous activity detection could be improved by gathering a night time dataset that increases the accuracy of the data in low light.