SONA COLLEGE OF TECHNOLOGY

PROJECT TITLE: Virtual Eye - LifeGuard for swimming pool to detect active drowning

IBM-Project-

MEMBERS:

1) **Sai Priyan J K**
2) **Pavinesh R A**
3) **Raja Vignesh Pherumal R**
4) **Shai Dharshan**

INTRODUCTION

1.1) 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. Beginners, especially, often feel it difficult to breathe underwater which causes breathing trouble which in turn causes a drowning accident. To overcome this conflict, a meticulous system is to be implemented along the swimming pools to save human life.

1.2) Purpose

The purpose of the proposed system is to detect the movement of the persons in the swimming pool and detect whether the person in the pool is swimming without any problem or the person is drowning using the machine learning model which uses yolo and opencv to capture frames from the camera and process it to identify drowning and alert the lifeguards to save the person from drowning.

2. LITERATURE SURVEY

2.1) Existing Problem

The person who is swimming in a pool needs to be rescued as soon as possible if he/she is drowning so that he/she does not die and swim without the fear of drowning. The problem affects a lot of people than we think it does. It affects ,The person who drowns loses his life.The person's kin and kith become traumatized by the loss of their loved one. The fellow swimmers who used to practice along with the person who drowned get their confidence and passion towards swimming lowered.Though Swimming is a healthy exercise and popular sport there is always a risk of people drowning. More than the fear of losing a swimming competition the fear of drowning affects a lot of people making them refrain from practicing**.**

2.2) Proposed Solution

The main idea revolves around capturing the swimmers real time and using only the important features like the joints and hand movements to predict if the person is drowning or not. This can be implemented using a media-pipe or yolo package such a way that we reduce the computational overload by considering only the important features for prediction.The real time capturing of the swimmers is done using a camera and the captured footage is used for further processing. The important features like the joints and hand position are extracted using a media-pipe. Then the extracted skeleton feature is used for prediction in the model.

2.3) Problem Statement Definition

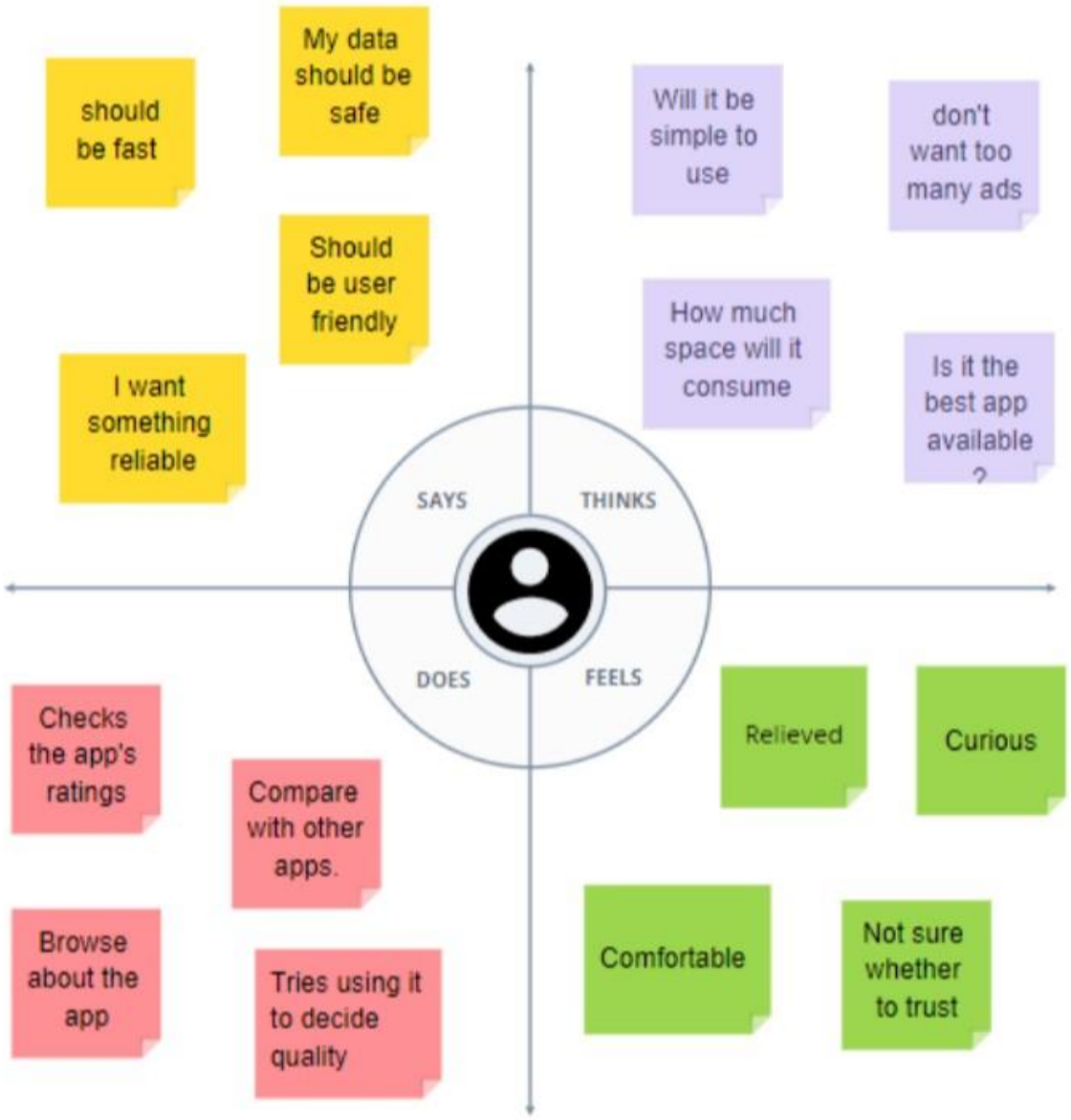
1. Swimming is one of the best exercises that helps people to reduce stress in this urban lifestyle. Swimming pools are found larger in number in hotels.

2. Applying the CNN algorithm to the dataset.Beginners, especially, often feel it difficult to breathe underwater which causes breathing trouble which in turn causes a drowning accident.

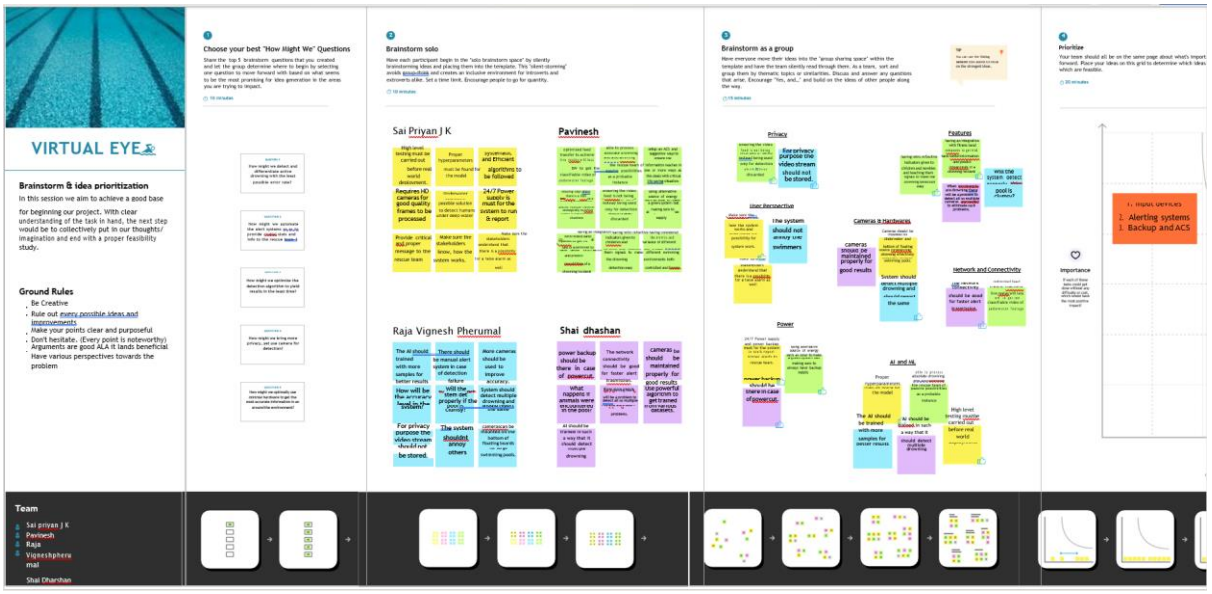
3.To overcome this conflict, a meticulous system is to be implemented along the swimming pools to save human life.

3) IDEATION & PROPOSED SOLUTION

3.1) Empathy Map Canvas



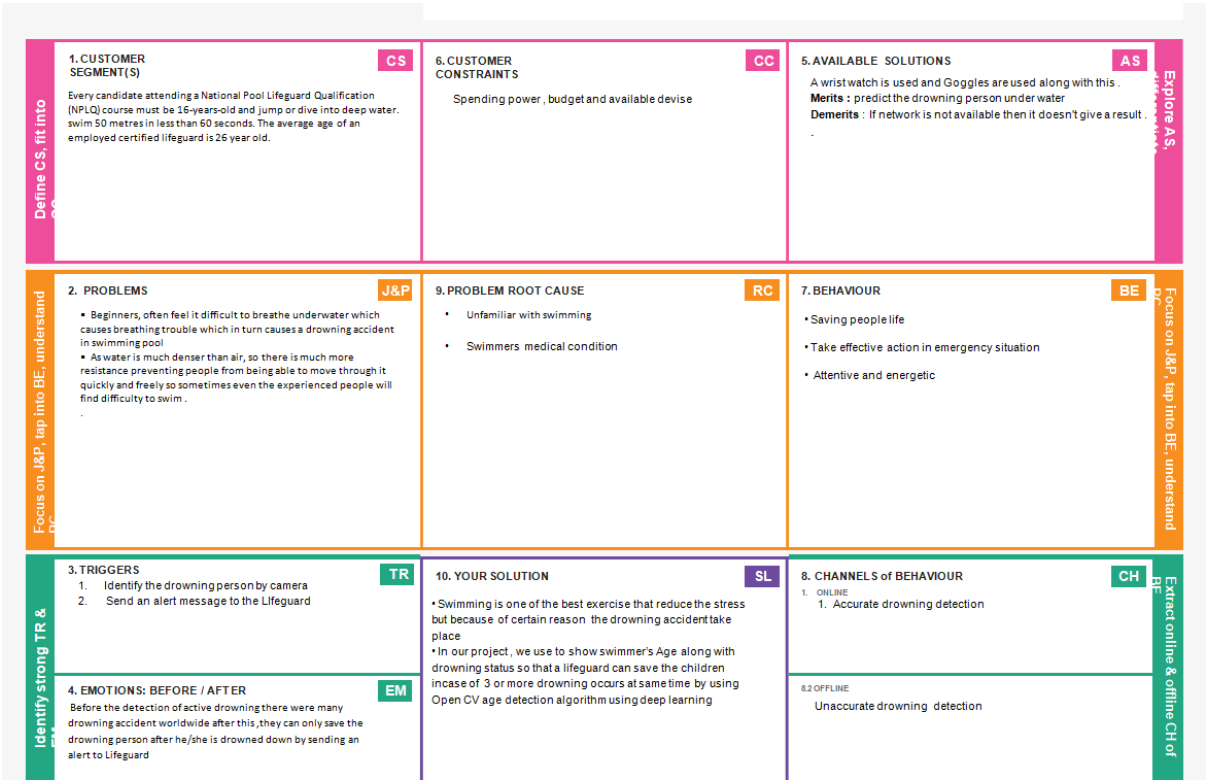
3.2Ideation & Brainstorming



3.3Proposed Solution

The main idea revolves around capturing the swimmers real time and using only the important features like the joints and hand movements to predict if the person is drowning or not. This can be implemented using a media-pipe or yolo package such a way that we reduce the computational overload by considering only the important features for prediction.The real time capturing of the swimmers is done using a camera and the captured footage is used for further processing. The important features like the joints and hand position are extracted using a media-pipe. Then the extracted skeleton feature is used for prediction in the model.

3.4Problem Solution fit



4.REQUIREMENT ANALYSIS

4.1 Functional requirement

5

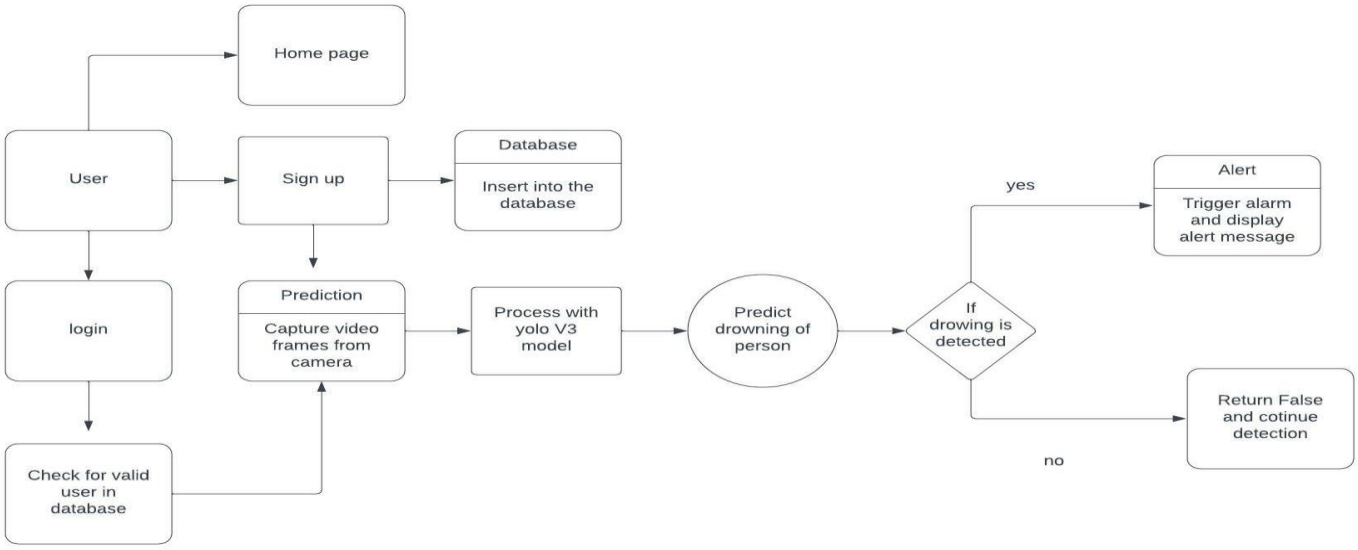
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Installation	Needed to be fixed under the water in the swimming pool
FR-2	Detection	Either horrified or in unconscious
FR-3	Audio	Ask for help or stay quiet if the person is unconscious
FR-4	Support	Take swim tubes or take the help of rescuer
FR-5	Prior Alert	Send alert message to the lifeguard

4.2 Non-functional Requirements:

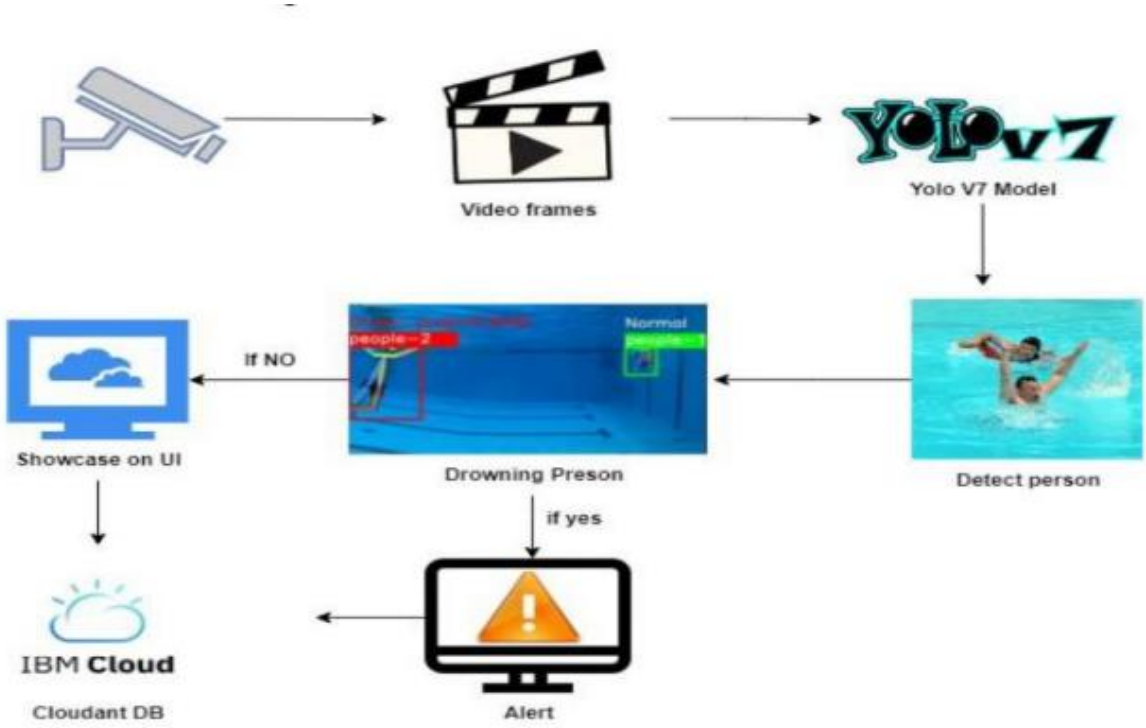
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	To ensure the safety of each and every person present in the pool. A Lifeguard should be present all the time in the pool.
NFR-2	**Security**	Lifeguards should be aware of the alert message to save the life of the swimmer
NFR-3	**Reliability**	Virtual eye lifeguard triggers an immediate prior alarm if a swimmer is in peril, helping to avoid panic even in critical situations.
NFR-4	**Performance**	The performance of the tool works better than available tools
NFR-5	**Availability**	Equipment and accessories include lifesaver rings, inflatable vests, life hooks, spine boards, rescue tubes, and a first aid kit.
NFR-6	**Scalability**	Virtual eye lifeguard detects potential drowning and promptly notifies you. It features the latest artificial intelligence technology and adapts to the needs of the user.

5.PROJECT DESIGN

5.1Data Flow Diagrams



Solution & Technical Architecture



5.1 User Stories

The feedback from a lot of the users have been collected to make the system more efficient and precise. It makes the system to make accurate results in drowning detection and help to save the lives.

6.PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

S.NO	MILESTONE	DESCRIPTION	DURATION
1	Prerequisites	Prerequisites are all the needs at the requirement level needed for the execution of the different phases of a project.	1 WEEK
2	Create & Configure IBM cloud services	IBM Cloud provides solutions that enable higher levels of compliance, security, and management, with proven architecture patterns and methods for rapid delivery for running mission-critical workloads.	1 WEEK
3	Develop the python script	A Python script is a set of commands included in a file that is intended to be run similarly to a program. The concept is that the file will be run or performed from the command line or from within a Python interactive shell to perform a particular activity. Of course, the file includes methods and imports different modules.	3 WEEKS

4	Develop web application	A web application (or web app) is application software that runs in a web browser, unlike software programs that run locally and natively on the operating system (OS) of the device.	1 WEEK

5	Ideation phase	Ideation is the process where you generate ideas and solutions through sessions such as Sketching, Prototyping, Brainstorming, Brain writing, Worst Possible Idea, and a wealth of other ideation techniques.	1 WEEK
6	Project design phases	Project design is an early phase of a project where the project's key features, structure, criteria for success, and major deliverables are planned out. The aim is to develop one or more designs that can be used to achieve the desired project goals.	1 WEEK
7	Project planning phase	In the Planning Phase, the Project Manager works with the project team to create the technical design, task list, resource plan, communications plan, budget, and initial schedule for the project, and establishes the roles and responsibilities of the project team and its stakeholders.	1 WEEK
8	Project development phase	Project development is the process of planning and allocating resources to fully develop a project or product from concept to go-live.	4 WEEKS

6.2 Sprint Delivery Schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a lifeguard , I can register for the application by enteringmy email, password, and confirming my password.	2	High	Sai Priyan J KPavinesh R A Raja Vignesh Pherumal R Shai Dharshan A M
Sprint 1	User conformation	USN-2	As a lifeguard, I will receive the conformation mail once I have registered for the application	2	Medium	Sai Priyan J KPavinesh R A Raja Vignesh Pherumal R Shai Dharshan A M
Sprint-1	Login	USN-3	As a lifeguard , I can log into the application by entering email& password	2	High	Sai Priyan J KPavinesh R A Raja Vignesh Pherumal R Shai Dharshan A M
Sprint-2	Cloudant DB	USN-1	Create DB	2	High	Sai Priyan J KPavinesh R A Raja Vignesh Pherumal R Shai Dharshan A M
Sprint-3	Coding (Accessing datasets)	USN-1	Coding is a set of instructions used to manipulate information so that a certain input results in a particular output.	2	High	Sai Priyan J K Pavinesh R A Raja Vignesh Pherumal R Shai Dharshan A M
Sprint-4	Application building	USN-1	As a Lifeguard , It will show the current Information of the swimming pool	1	Medium	Sai Priyan J K Pavinesh R A Raja Vignesh

						Pherumal R Shai Dharshan A M

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

Sprint	**Total Story Points**	**Duration**	**Sprint Start Date**	**Sprint End Date (Planned)**	**Story Points Completed (as on Planned End Date)**	**Sprint Release Date(Actual)**
Sprint-1	20	4 Days	24 Oct 2022	27 Oct 2022	20	29 Oct 2022
Sprint-2	20	5 Days	28 Oct 2022	01 Nov 2022	20	04 Nov 2022
Sprint-3	20	8 Days	02 Nov 2022	09 Nov 2022	20	11 Nov 2022
Sprint-4	20	9 Days	10 Nov 2022	18 Nov 2022	20	19 Nov 2022

7.CODING & SOLUTIONING

7.1 Feature 1

```
@app.route('/login')
def login():
    return render_template('login.html',message="")

@app.route('/afterlogin',methods=['POST'])
def afterlogin():
    x=[x for x in request.form.values()]
    user =x[0]
    passw=x[1]
    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):
        print("login")
        return render_template('login.html',message="The user is not found")
    else:
        print("holaaaaaaaaa")
        if((user==docs[0][0]['_id'] and passw==docs[0][0]['psw'])):
            return redirect(url_for('prediction'))
        else:
            print('Invalid User')
            # flash("invalid")
            return render_template('login.html',message="invalid credentials")
    return "nothing"

@app.route('/logout')
```

```
def logout():
    return render_template('logout.html')
```

- Creating and maintaing the sessions for every user and storing the credentials in the database.
- Securing the data by providing a login for users.
- Alerting the user for wrong credentials and emails.
- Providing the sessions for every instance.

7.2 Feature 2

```
    @app.route('/prediction')
def prediction():
    return render_template('prediction.html',prediction="Checking for drowning")

def draww(frame,bbox,conf):
    for i in range(len(bbox)):
        print(conf)
        start_point = (bbox[i][0], bbox[i][1])
        end_point = (bbox[i][2], bbox[i][3])
        color = (255, 0, 0)
        thickness = 2
        frame = cv2.rectangle(frame, start_point, end_point, color, thickness)
    return frame

@app.route('/result',methods=['GET',"POST"])
def res():
    webcam =cv2.VideoCapture('drowninga.mp4')

    if not webcam.isOpened():
        print("Could Not Open Webcam")
        exit()
```

```
t0=time.time()
center0=np.zeros(2)
isDrowning=False

while webcam.isOpened():
    status,frame=webcam.read()
    bbox,label,conf=cv.detect_common_objects(frame)
    print("seeeeeeee")
    print("-----------------------------------------")
    print(bbox)
    print("-----------------------------------------")
    if(len(bbox)>0):

        bbox0=bbox[0]

        center =[0,0]

        center=[(bbox0[0]+bbox0[2])/2,(bbox0[1]+bbox0[3])/2]

        hmov=abs(center[0]-center0[0])
        vmov= abs(center[1]-center0[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('bbox: ',bbox,'center:',center, 'center0:',center0 )
```

```
            print('Is he drowning: ',isDrowning)

            center0 =center

            # out=draw_bbox(frame,bbox,label,conf,isDrowning)

            # print(bbbox.x0)
            # out=draw_bbox(frame,bbbox,label,conf)
            # out=draw_bbox(bbox,frame)

            # frame=draww(frame,bbox,conf)
            # out=frame
            out= draw_bbox(frame, bbox, label, conf)
            cv2.imshow("Real-Time objects detection",out)
        else:
            out=frame
            cv2.imshow("Real-Time objects detection",out)
        # cv2.imshow("Real-Time objects detection",frame)
        if(isDrowning==True):
            #audio =os.path.dirname(__file__)+"/s.wav"
            #playsound(audio)
            playsound("C:\\Users\\SAI\\Downloads\\IBM-Project-2094-1658428458-main\\IBM-Project-2094-1658428458-main\\Project development phase\\sprint 2/a.mp3")
            webcam.release()
            cv2.destroyAllWindows()
            # return "nothing"
            return render_template('prediction.html',prediction="Emergency !!! The Person is drowning")

        if cv2.waitKey(1) & 0XFF == ord('q'):
            break

    webcam.release()
    cv2.destroyAllWindows()
```

return render_template('prediction.html',prediction="Checking for drowning")

- Implemented the computer vison to capture the frames and analyze the input.
- Machine learning model is used to predict the drowning of the person

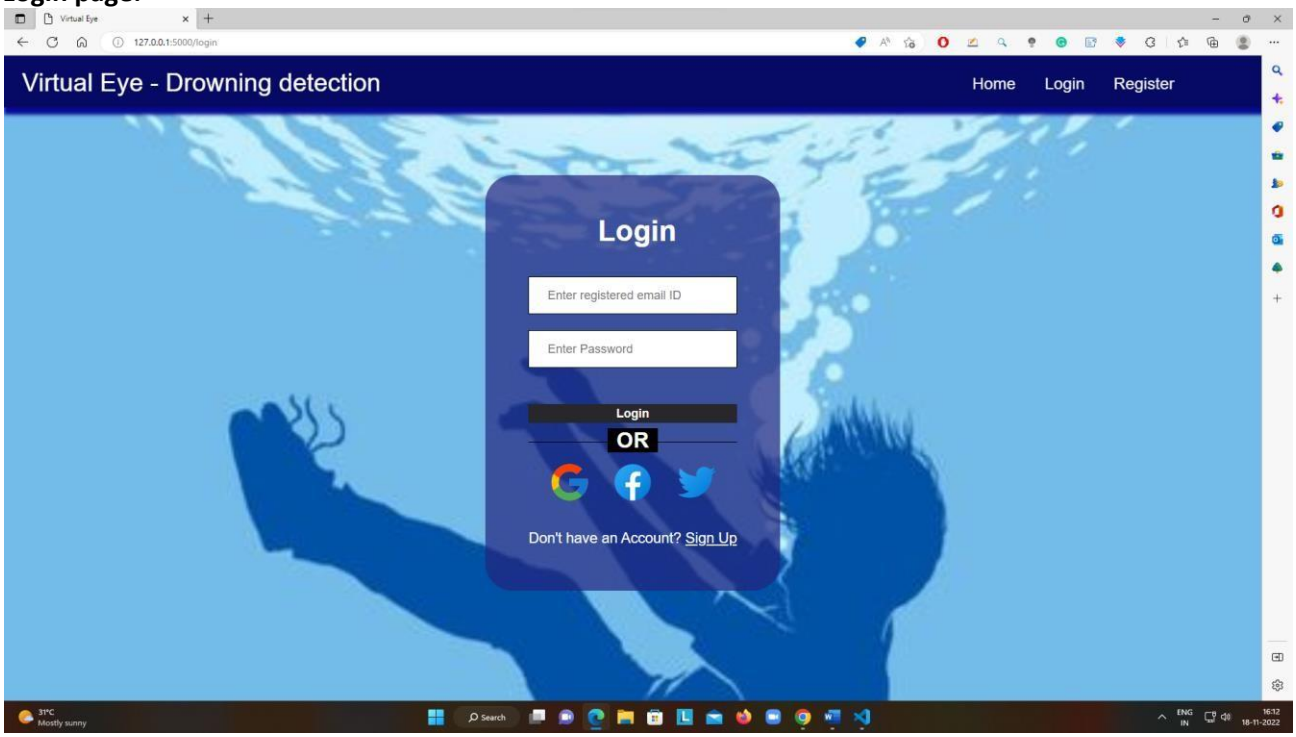
7.3 Database Schema

```
@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',message='Registration Successful, Please login using your details')
```

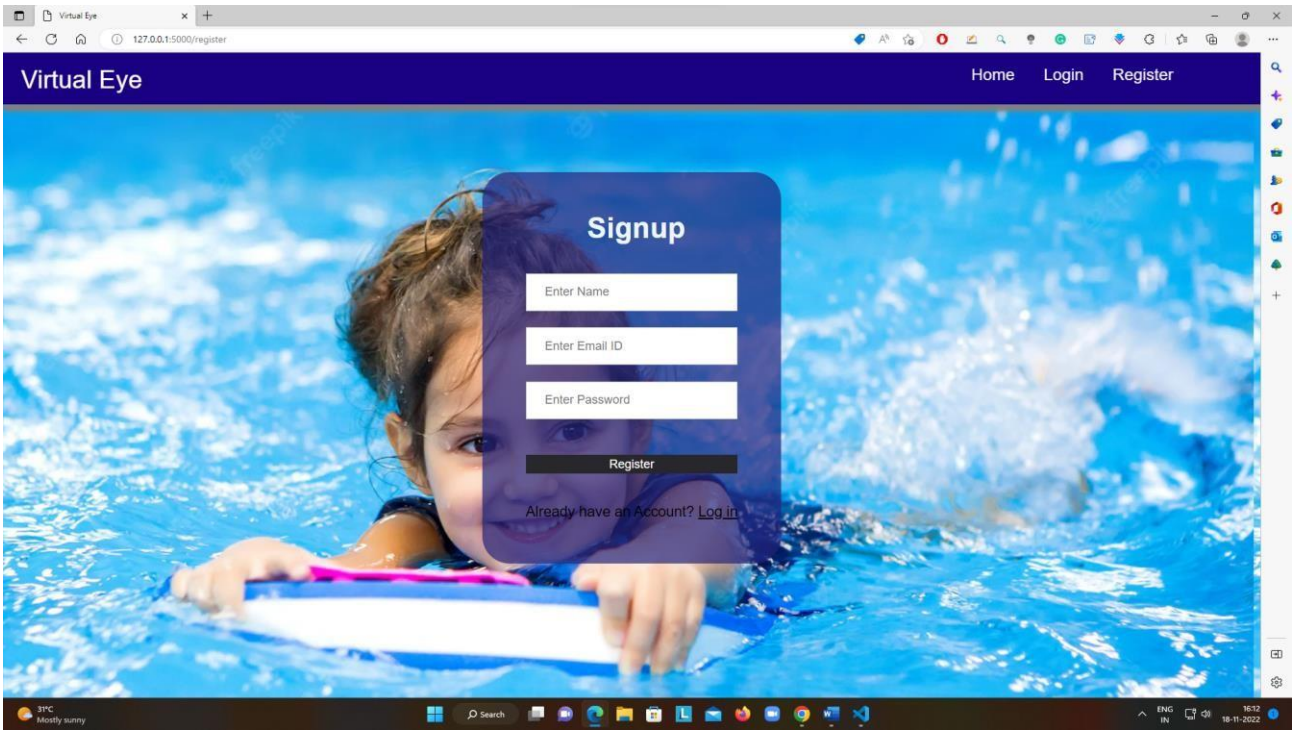
8.TESTING

8.1 Testcases

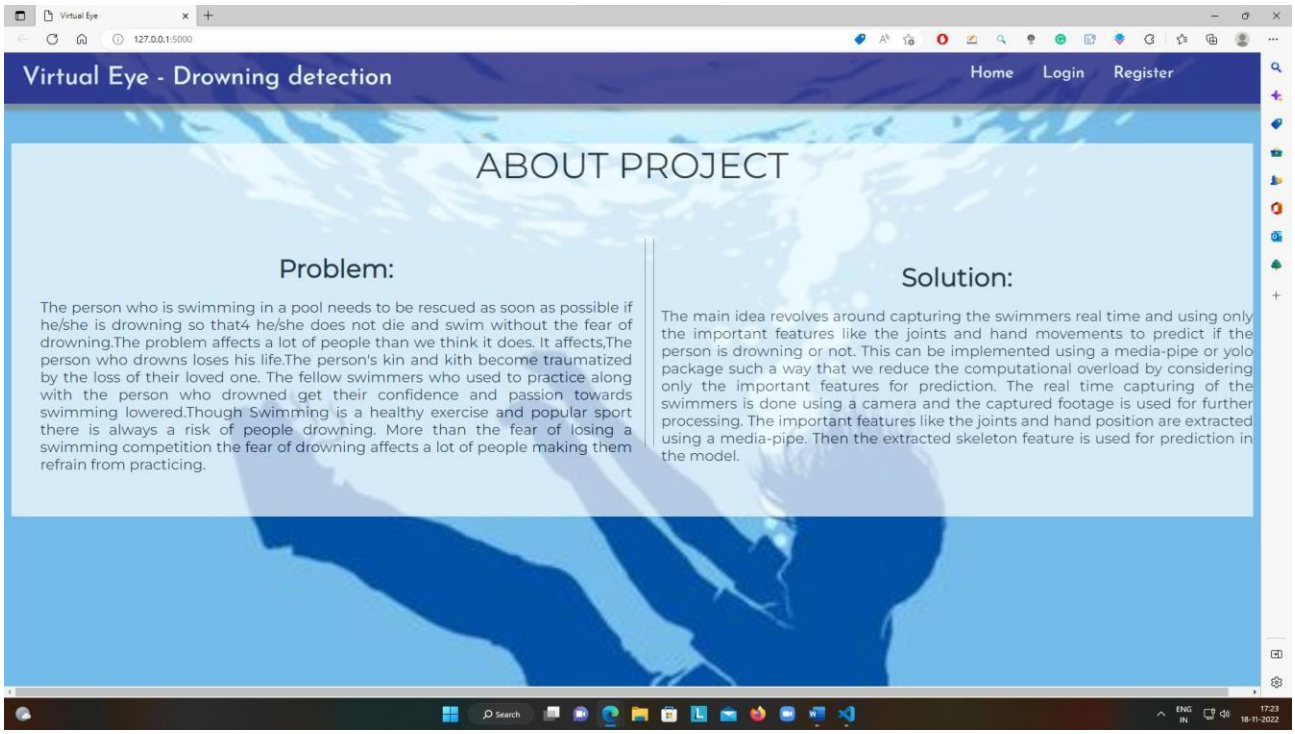
Login page:



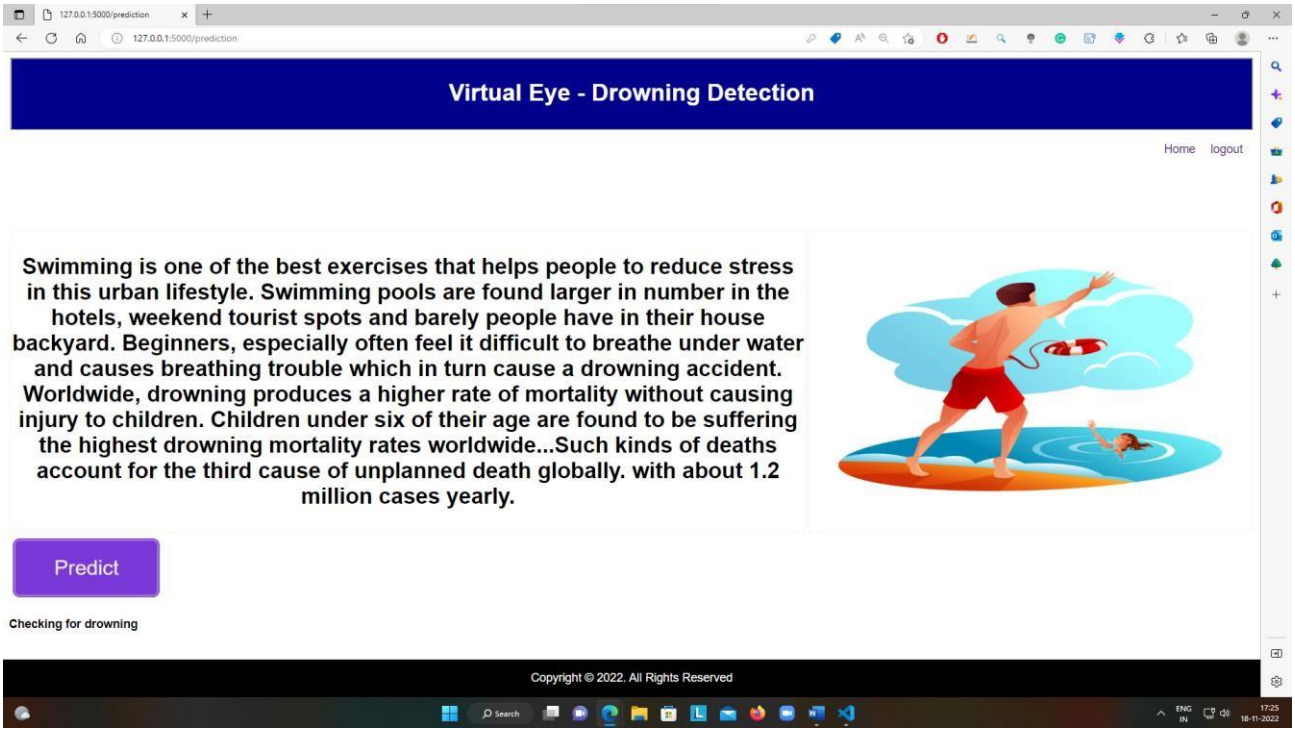
Register page:



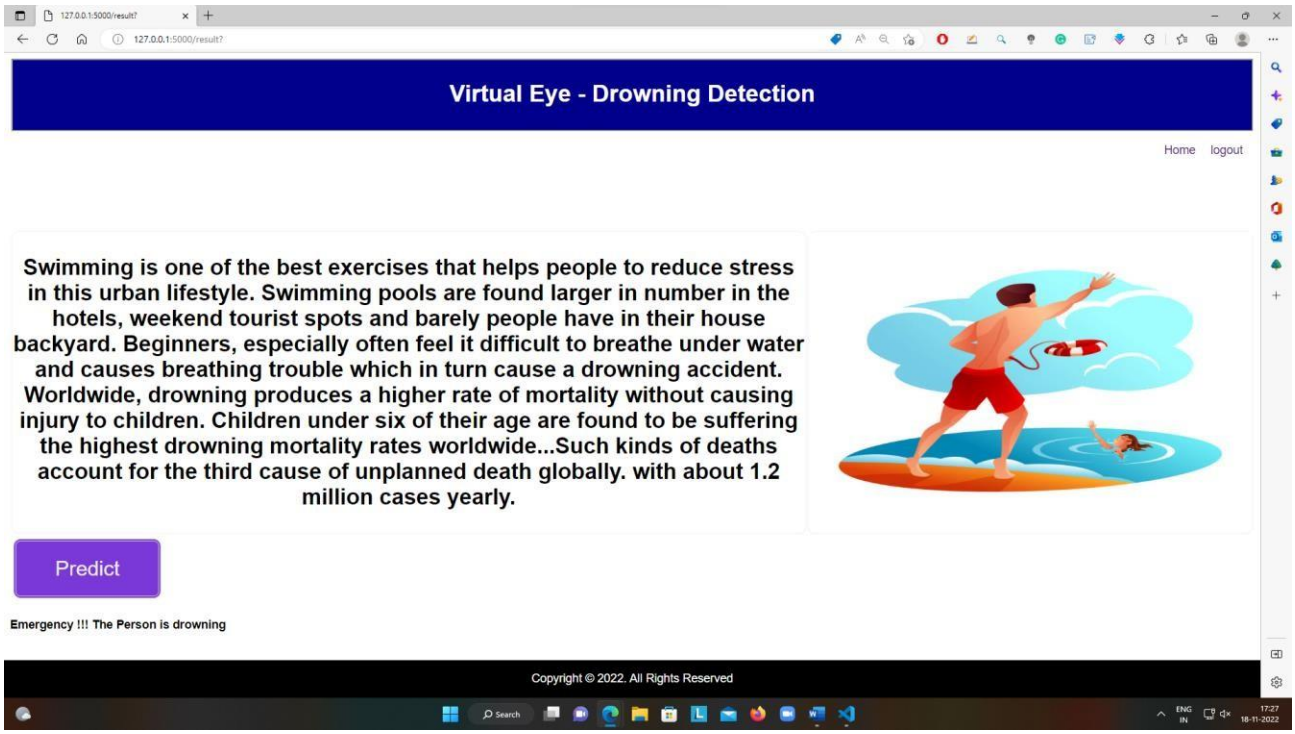
Home page:



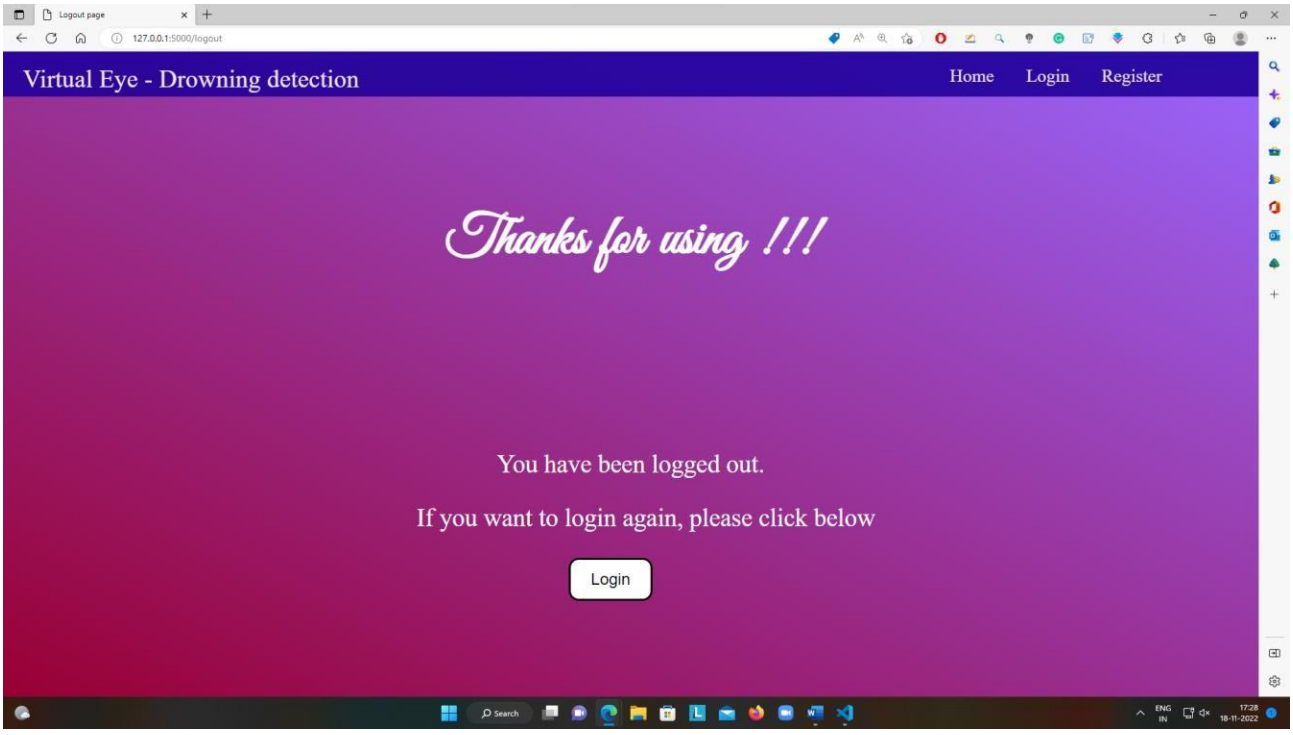
Prediction page (Before prediction):



Prediction page (After Prediction):



Logout page



8.2 User Acceptance Testing

1. Test Case Analysis

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	5	4	2	0	11
Duplicate	0	0	0	0	0
External	2	3	0	2	4
Fixed	1	2	2	0	5
Not Reproduced	0	0	0	0	0
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	6	9	4	2	21

This report shows the number o.f test cases that have passed, failed, and untested

9.RESULTS

9.1 Performance Metrics

S.No	Parameter	Values
1	Drowning detection accuracy	91.002654%
2	Yolo	93.4%
3	Drowning alert	97.9%
4	Application accuracy	93.5532465%

10. ADVANTAGES & DISADVANTAGES

10.1 Advantages :

- Simple and computationally fast
- Easy for the user to detect drowning
- Detects so quick and accurate

10.2 Disadvantages :

- It should be installed with water resistant cameras
- It acts as a add on devices.
- It should be connected to gpu.

11. CONCLUSION :

- 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. Beginners, especially, often feel it difficult to breathe underwater which causes breathing trouble which in turn causes a drowning accident. To overcome this conflict, a meticulous system is to be implemented along the swimming pools to save human life. The main idea revolves around capturing the swimmers real time and using only the important features like the joints and hand movements to predict if the person is drowning or not. This can be implemented using a media-pipe or yolo package such a way that we reduce the computational overload by considering only the important features for prediction.

- The real time capturing of the swimmers is done using a camera and the captured footage is used for further processing. The important features like the joints and hand position are extracted using a media-pipe. Then the extracted skeleton feature is used for prediction in the model.

12. FUTURE SCOPE :

we propose integrating the findings of this research in a large-scale machine learning based online monitoring system using only the cameras of the required parameters. The tested algorithms would predict the drowing person immediately based on the real-time data fed from the camera.
The proposed system would employ the parameter frames and parameter readings and communicate those readings using an maching learning model and
cloud. It would identify drowning of the person before he drowns and
alert concerned authorities. It will hopefully result in curtailment of people
and consequently de-escalate harrowing diseases. In this regard,
the application of a prescriptive analysis from the expected values would lead to future facilities to support decision and policy makers

13. Appendix

SOURCE CODE

```
import cv2

import os

import numpy as np

from pathlib import Path

import cvlib as cv

import time

from cv2 import threshold

from cvlib.object_detection import draw_bbox

# from matplotlib.patches import draw_bbox


from flask import Flask , request, render_template , redirect , url_for


from playsound import playsound

# from utils import download_file
```

```
from cloudant.client import Cloudant

ACCOUNT_NAME, API_KEY="bd84549c-d8e0-47c4-9fac-c68107bcf136-bluemix","M2omO01qPVjfoQ0tmEoHfmWIJiYVYIu2JpT9w0puZ1h0"

client=Cloudant.iam(ACCOUNT_NAME, API_KEY, connect=True)


my_database=client.create_database('my_database')

app=Flask(__name__)


@app.route('/')

def index():

        return render_template('index.html')


@app.route('/index')

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',message='Registration Successful, Please login using your details')
        else:
        return render_template('register.html',message="You are alredy a member, please login using your details")
        return "nothing"
```

```
@app.route('/login')
def login():
        return render_template('login.html',message="")

@app.route('/afterlogin',methods=['POST'])
def afterlogin():
        x=[x for x in request.form.values()]
        user =x[0]
        passw=x[1]
        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):
        print("login")
        return render_template('login.html',message="The user is not found")
        else:
       print("holaaaaaaaaaa")
```

```
        if((user==docs[0][0]['_id'] and passw==docs[0][0]['psw'])):
          return redirect(url_for('prediction'))
          else:
            print('Invalid User')
          # flash("invalid")
          return render_template('login.html',message="invalid credentials")
          return "nothing"


@app.route('/logout')
def logout():
          return render_template('logout.html')


# class dotdict(dict):
#     """dot.notation access to dictionary attributes"""
#         __getattr__ = dict.get
#         __setattr__ = dict.__setitem__
#         __delattr__ = dict.__delitem__


@app.route('/prediction')
def prediction():
          return render_template('prediction.html',prediction="Checking for drowning")


def draww(frame,bbox,conf):
```

```
        for i in range(len(bbox)):
        print(conf)
        start_point = (bbox[i][0], bbox[i][1])
        end_point = (bbox[i][2], bbox[i][3])
        color = (255, 0, 0)
        thickness = 2
        frame = cv2.rectangle(frame, start_point, end_point, color, thickness)
        return frame

@app.route('/result',methods=['GET',"POST"])
def res():
        webcam =cv2.VideoCapture('drowninga.mp4')

        if not webcam.isOpened():
       print("Could Not Open Webcam")
        exit()
        t0=time.time()
    center0=np.zeros(2)
        isDrowning=False

        while webcam.isOpened():
       status,frame=webcam.read()
       bbox,label,conf=cv.detect_common_objects(frame)
       print("seeeeeeee")
```

```
print("-----------------------------------------")
 print(bbox)
print("-----------------------------------------")
if(len(bbox)>0):

   bbox0=bbox[0]

 center =[0,0]

   center=[(bbox0[0]+bbox0[2])/2,(bbox0[1]+bbox0[3])/2]

   hmov=abs(center[0]-center0[0])
 vmov= abs(center[1]-center0[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('bbox: ',bbox,'center:',center, 'center0:',center0 )
     print('Is he drowning: ',isDrowning)

  center0 =center

  # out=draw_bbox(frame,bbox,label,conf,isDrowning)

  # print(bbbox.x0)
  # out=draw_bbox(frame,bbbox,label,conf)
  # out=draw_bbox(bbox,frame)

  # frame=draww(frame,bbox,conf)
  # out=frame
  out= draw_bbox(frame, bbox, label, conf)
    cv2.imshow("Real-Time objects detection",out)
  else:
  out=frame
    cv2.imshow("Real-Time objects detection",out)
  # cv2.imshow("Real-Time objects detection",frame)
 if(isDrowning==True):
  #audio =os.path.dirname(__file__)+"/s.wav"
    #playsound(audio)
```

```
            playsound("C:\\Users\\SAI\\Downloads\\IBM-Project-2094-1658428458-main\\IBM-Project-2094-1658428458-main\\Project development phase\\sprint 2/a.mp3")

            webcam.release()

            cv2.destroyAllWindows()

        # return "nothing"

        return  render_template('prediction.html',prediction="Emergency  !!! The Person is drowning")



        if cv2.waitKey(1) & 0XFF == ord('q'):

        break



        webcam.release()

    cv2.destroyAllWindows()

        return  render_template('prediction.html',prediction="Checking  for drowning")



if __name__ =='main':

        app.run(debug=True)
```

GITHUBLINK:
https://github.com/IBM-EPBL/IBM-Project-24477-1659943372

PROJECTDEMOLINK:

https://drive.google.com/file/d/15b-gueHif6PhNeu9fcLc_TydfKIhK2Uz/view?usp=sharing