**SONA COLLEGE OF TECHNOLOGY**

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

**IBM-Project-24477-1659943372**

**MEMBERS:**

**1) Sai Priyan J K**

**2) Pavinesh R A**

**3) Raja Vignesh Phermal R**

**4) Shai Dharshan**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Title** | **Page No** |
| **1** | **Introduction** | **3** |
| **2** | **Literature survey** | **4** |
| **3** | **Theoretical Analysis** | **6** |
| **4** | **Experimental Investigations** | **9** |
| **5** | **Result** | **13** |
| **6** | **Advantages & Disadvantages** | **15** |
| **7** | **Applications** | **16** |
| **8** | **Conclusion** | **16** |
| **9** | **Future Scope** | **16** |
| **10** | **Source code** | **17** |

**1. 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.

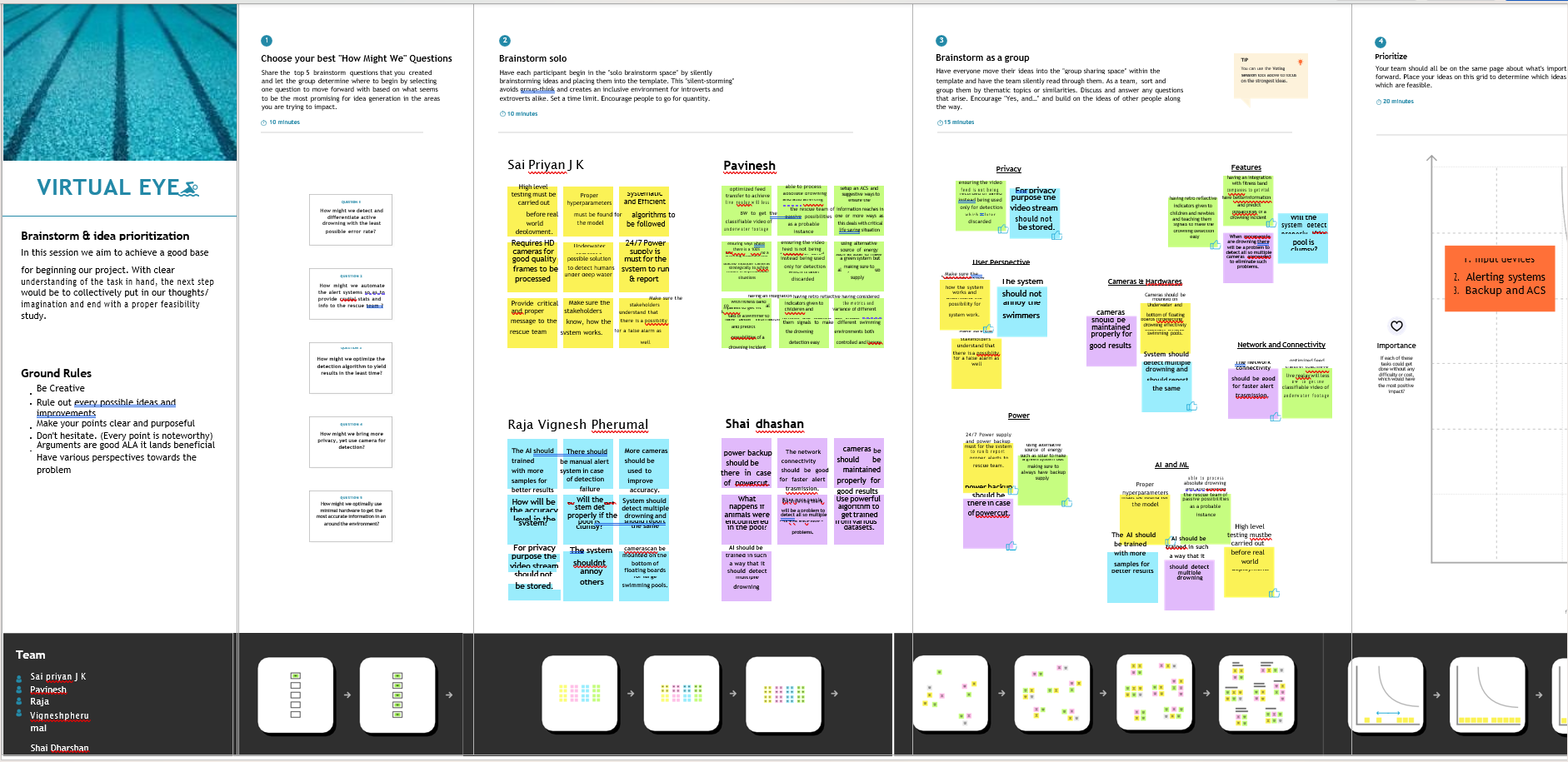
2. Applying the CNN algorithm to the dataset.Beginners, especially, often feel it difficult to breathe underwater which causes breathing trouble which in turn causes a drowning accident.

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

**3) IDEATION & PROPOSED SOLUTION**

* 1. **) Empathy M ap Canvas** Diagram

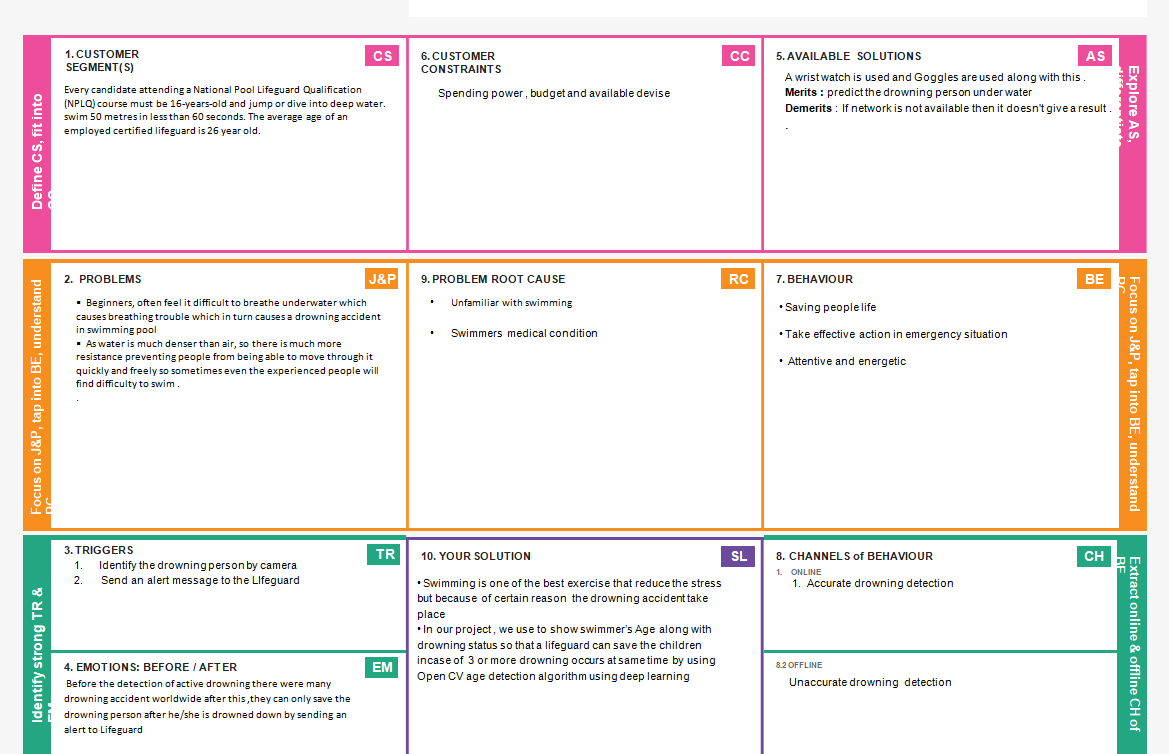
     Description automatically generated
  2. **Ideation & Brainstorming**

****

* 1. **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.

* 1. **Problem Solution fit**



**4.REQUIREMENT ANALYSIS**

* 1. **Functional requirement**

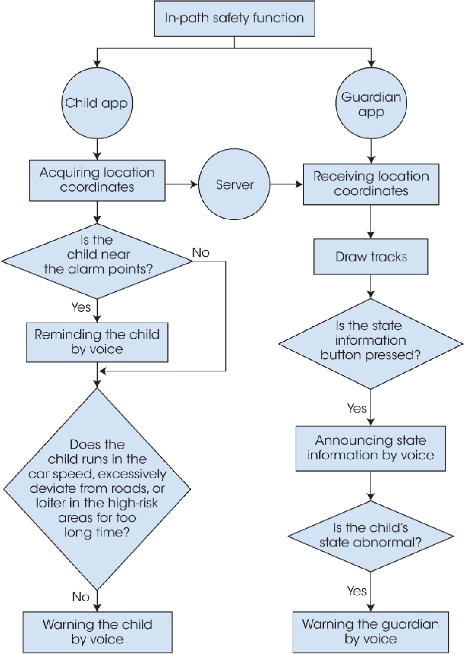
|  |  |  |
| --- | --- | --- |
| **FR No.** | **Functional Requirement (Epic)** | **Sub Requirement (Story / Sub-Task)** |
| FR-1 | Installation | Needed to be fixed under the water in the swimming pool |
| FR-2 | Detection | Either horrified or in unconscious |
| FR-3 | Audio | Ask for help or stay 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**

* 1. **Data Flow Diagrams**



* 1. **Solution & Technical Architecture**

**Camera**

**Video**

**frame**

**YOLO**

**model**

**Detect**

**the person in pool**

**YES**

**Showcasing**

**on UI & Triggered Alarm**

**NO**

**Showcase**

**on UI**

**IBM**

**Cloudant DB**

* 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 entering my email, password, and confirming my password. | 2 | High | Sai Priyan J K Pavinesh 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 K Pavinesh 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 K Pavinesh R A Raja Vignesh Pherumal R Shai Dharshan  A M |
| Sprint-2 | Cloudant DB | USN-1 | Create DB | 2 | High | Sai Priyan J K Pavinesh 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("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')

* 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) & *0X*FF == 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')

    else:

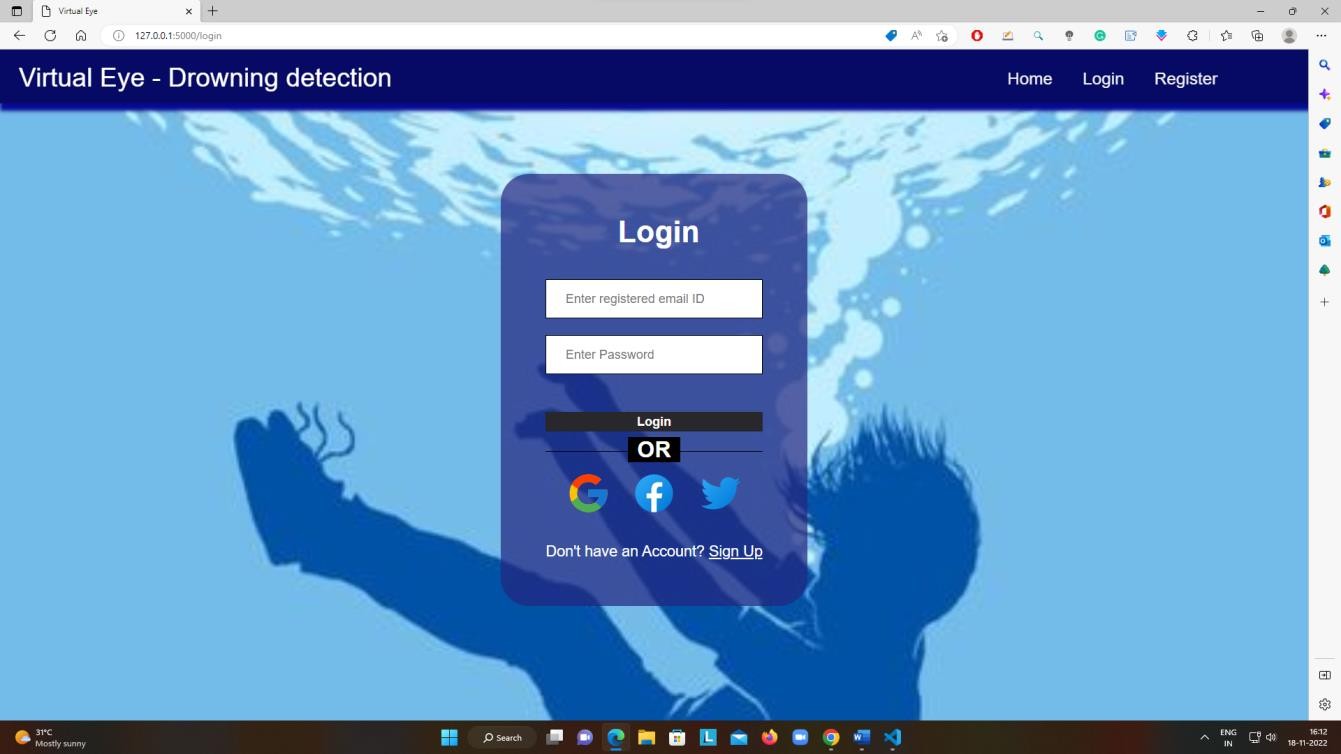
        return render\_template('register.html',*message*="You are alredy a member, please login using your details")

    return "nothing"

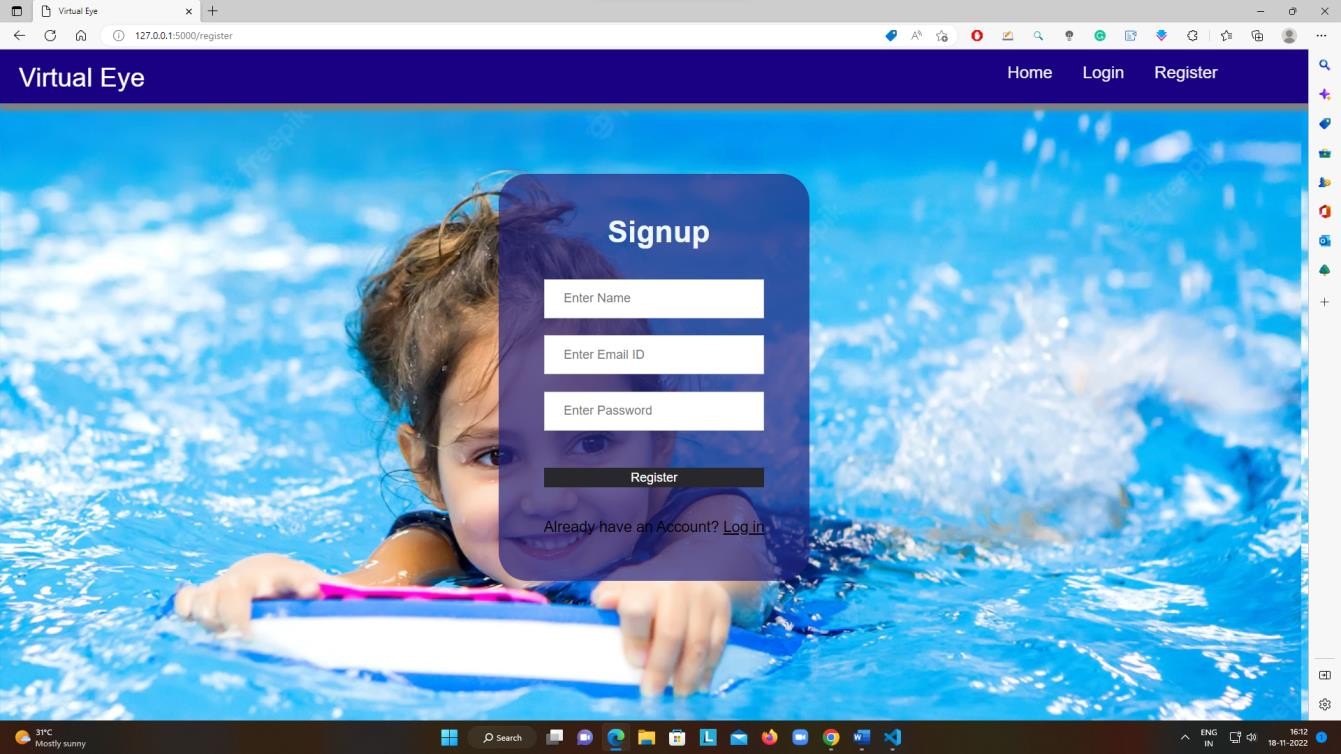
**8.TESTING**

**8.1 Testcases**

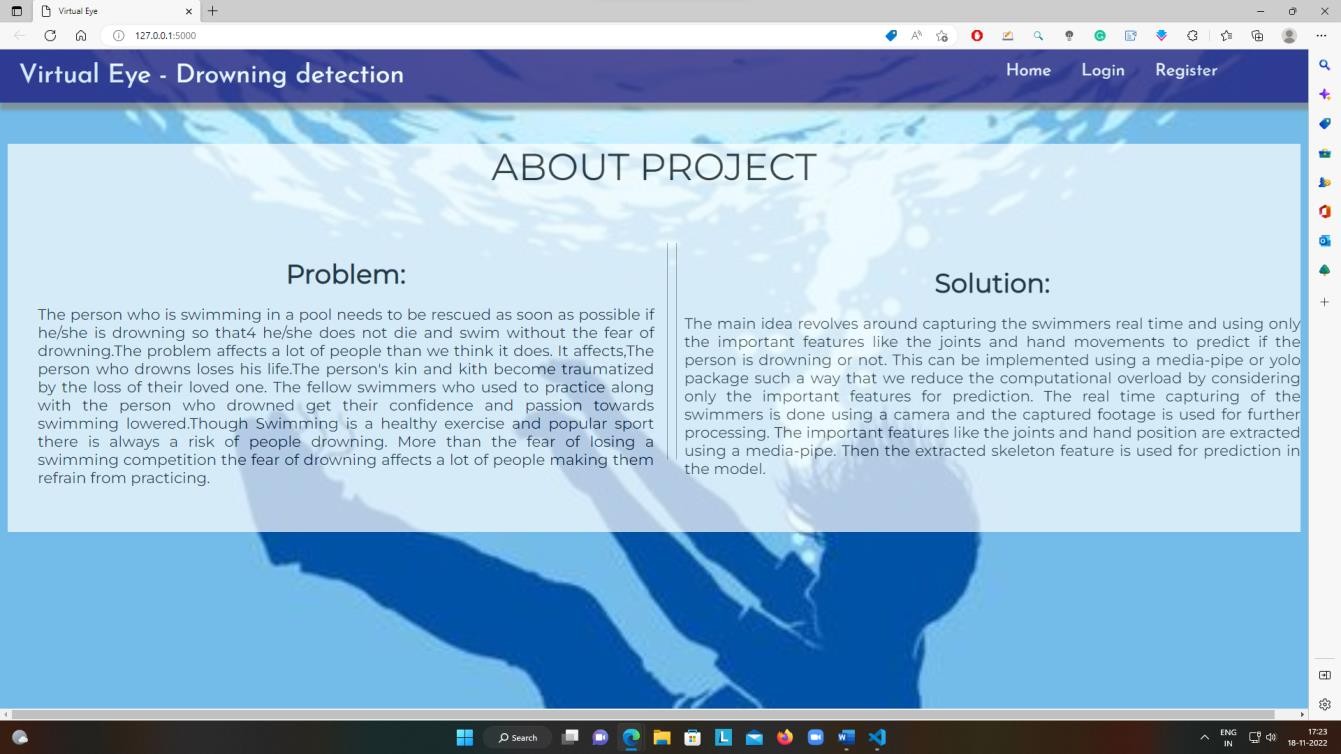
# Login page:



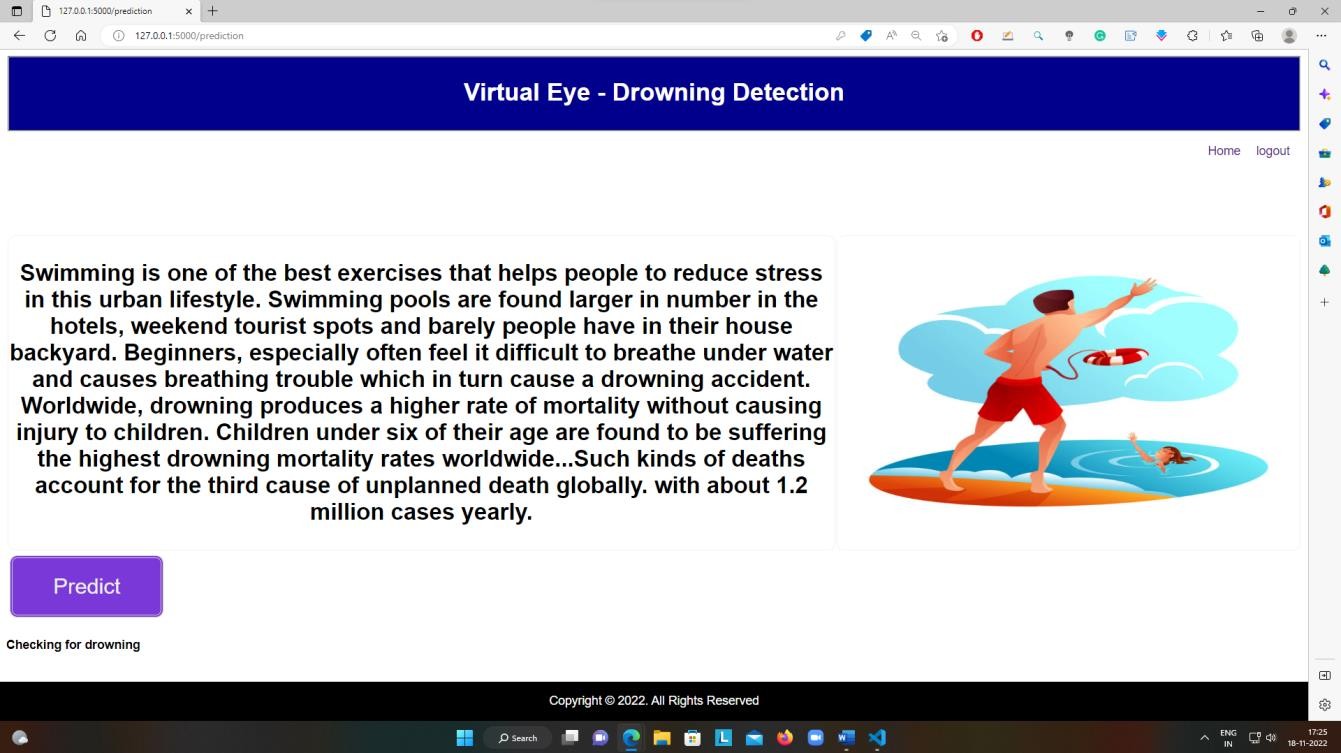
**Register page:**



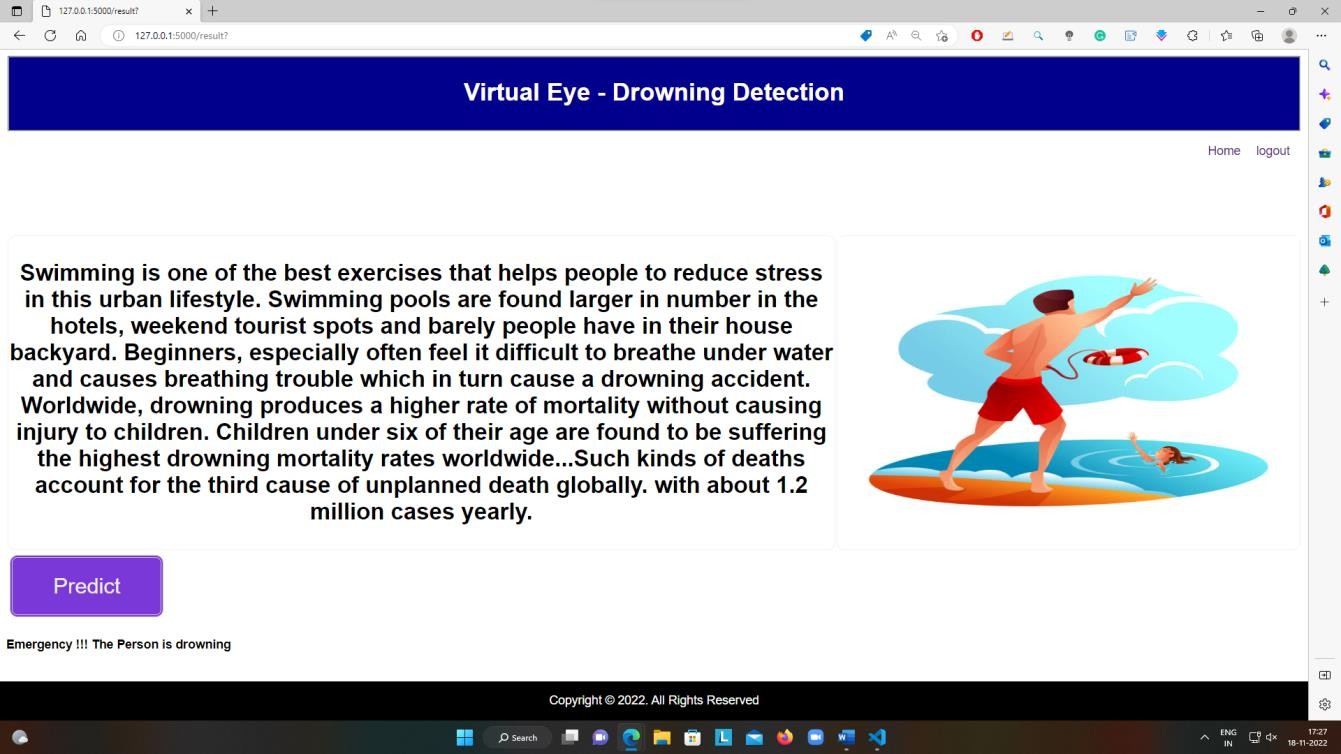
# Home page:



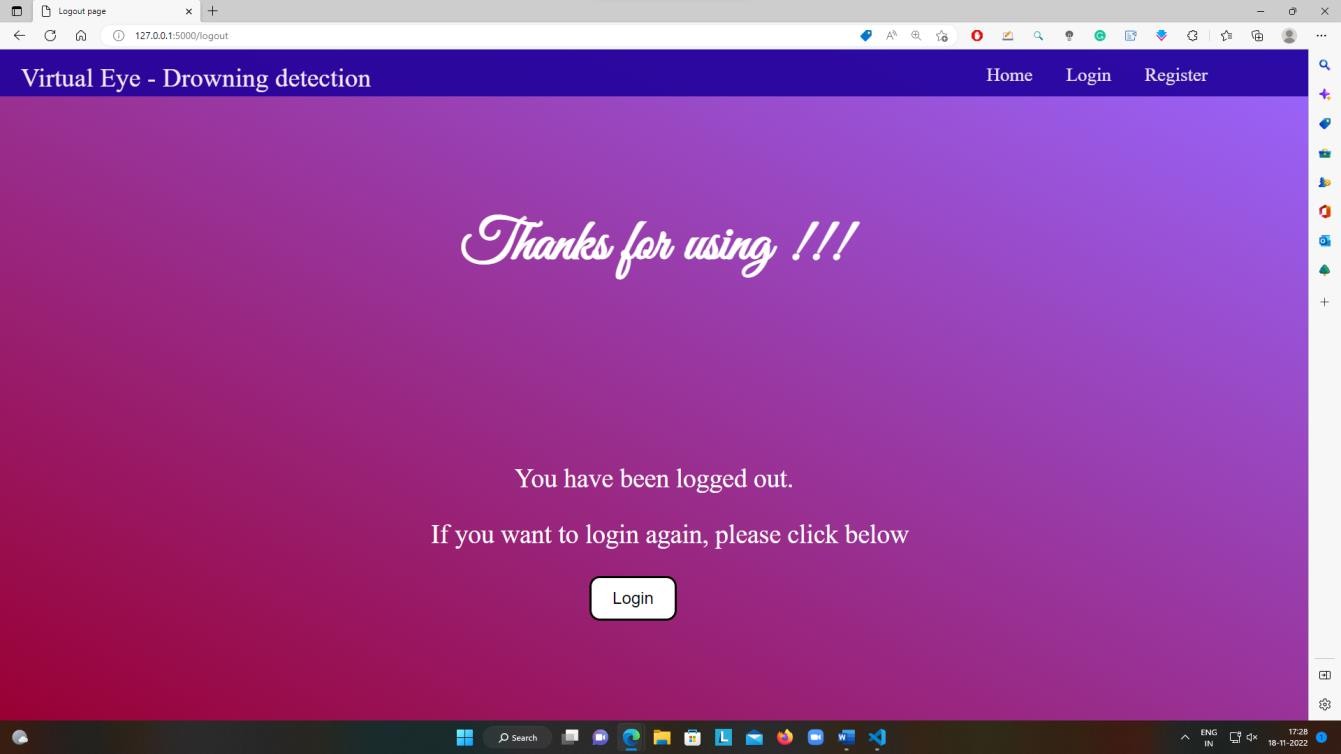
Prediction page (Before prediction):



Prediction page (After Prediction):



Logout page:



* 1. **User Acceptance Testing**

# Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were 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 Report Output | 1 | 0 | 0 | 1 |

# 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 o.f test cases that have passed, failed, and untested

**9.RESULTS**

**9.1 Perfomance Metrics**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | 10-Nov-22 PNT2022TMID18501 |  | | |
| Virtual eye-Lifeguard for swimming 4 marks |
| Test case ID | Feature Type | Home Page | Test Scenario  Verify user is able to see the Login/Signup popup when  user clicked on My account button | Steps TO Execute  I.Enter URL and click go  2.Click on My Account dropdown button 3.Verify login/Singup popup displayed or not | Test  Login.html | Expected Result Login/Signup popup should display | Actual Result  Working as expected |
| LoginPage\_TC\_002 |  | Home Page | Verify the UI elements in Login/Signup popup | I.Enter tJRL and dick go   1. Click on My Account dropdown 2. Verify login/Singup popup with below UI elements:   a.email text box b. Password text box   1. L%in button 2. New customer? Create account link 3. Last password? Recovery password link | Login.html | Application should show below elements:  a.email text box b.password text box   1. Login button with orange colour 2. New custotner? Create account link 3. Last password? Recovery password link | Working as expected |
|  | Functional | Home page | Verify user is able to log into application with Valid credentials | I. Enter URL and dick go   1. Click on My Account dropdown 2. 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 expected |
|  | Functional | Login page | Verify user is able to log into application with Invalidcredentials | 1, Enter URL and click go  2.Click on My Account button 3.Enter Invalid username/email in Email text box   1. Enter valid password in password text box 2. Click on •n button | Username:lax password:lax26 | Application should show 'Incorrect email or password ' validation message. | Working as expected |
| LoginPage\_TC\_004 | Functional | Login page | Verify user is able to log into application with Invalidcredentials | I-Enter URL and click go   1. Click On My Account dropdown 2. 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 expected |
|  | Functional | Login page | Verify user is able to into application with Invalid credentials | I. Enter URL and click go   1. Click on My Account dropdown 2. Enter Invalid username/email in Email text box 3. Enter Invalid password in password text box 4. Click on I in button | username:lax26@mail password:1803 | Application should show 'Incorrect email or password ' validation message. | Working as expected |
| Predictionpage\_TC\_  00 6 |  | 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 expected |

**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)

**GITHUB LINK:** **https://github.com/IBM-EPBL/IBM-Project-24477-1659943372**

**PROJECT DEMO LINK: https://drive.google.com/file/d/15b-gueHif6PhNeu9fcLc\_TydfKIhK2Uz/view?usp=sharing**