VIRTUALEYE - LIFE GUARD FOR SWIMMING POOLS TO DETECT ACTIVE DROWNING

IBM PROJECT REPORT

Submitted By

ABHAYA DEV S K	- 962319104001
GOKUL KRISHNA H S	- 962319104042
BENISHA J	- 962319104031
JOEL PAUL	- 962319104048

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

AMRITA COLLEGE OF ENGINEERING AND TECHNOLOGY

ERACHAKULAM, NAGERCOIL.

ANNA UNIVERSITY::CHENNAI 600 025

JUNE 2022

CERTIFICATE OF EVALUATION

COLLEGE : AMRITA COLLEG OF ENGINEERING AND TECHNOLOGY

NAME

BRANCH : COMPUTER SCIENCE AND ENGINEERING

SEMESTER : VII

TITLE : VIRTUALEYE - LIFE GUARD FOR SWIMMING POOLS TO DETECT ACTIVE DROWNING

TEAM ID : PNT2022TMID51944

STUDENT NAMES	**REGISTRATION NUMBER**	**SUPERVISOR**
ABHAYA DEV S K	-962319104001	
BENISHA J	-962319104031	Mrs. JOTHI LAKSHMI S L
GOKUL KRISHNA H S	-962319104042	
JOEL PAUL	-962319104048	

The report of this project is submitted by the above students in partial fulfilment for the award of Bachelor of Engineering Degree, in Computer Science and Engineering of Anna University are evaluated and confirmed to the reports of the work done by the above students.

MENTOR **EVALUATOR**

ACKNOWLEDGEMENT

First and foremost we would like to express our sincere gratitude to our respected Founder Amma,Mata Amritanandamayi Devi and Chairman Mr. K S Ramasubban IAS(Retd) for their blessings and grace in making our project great success.

We would like to place a record with deep sense of gratitude to our Honorable Principal Dr. T Kannan for having given us the opportunity to pursue B.E., course in this prestigious institution.

We would like to express our sincere thanks to our beloved Head of the Department Dr. P M Siva Raja and Project Coordinator Mrs. S L Jothi Lakshmi, for creating a supportive and a model environment for us to work and build up our innovative skills. We also thank our project Mentor Mr. Anant Raj I V for the kind encouragement and moral support, who has been a constant source of inspiration to us.

We wish to express our sincere sense of gratitude to Dr. P M Siva Raja, Head, Department of Computer Science and Engineering and to our Project Guide who enabled us to complete our project successfully.

Virtual Eye - Life Guard for Swimming Pools to Detect Active Drowning

1. INTRODUCTION

1.1 Project Overview

Objective: -

In Modern metropolitan lifestyle, swimming in one of the best activity for stress reduction .Worldwide, drowning results in a higher mortality rate without harming children .The highest global drowning fatality rates are observed to be among children under the age of six. With around 1.2 million incident each year .These types of deaths rank third among all unexpected deaths worldwide

Abstract: -

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. 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. By building an intelligent Drowning Detection system using an AI device that can accurately and effectively monitor swimmers and detect drowning in pools and automatically alert the lifeguard for his assistance.

1.2 Purpose: -

The aim this project is to design a meticulous system which can be implemented among the swimming pools to save human life. An alarm will be issued to call he attention of the lifeguards .When the video is being streamed underwater and swimmer position is the being examined to determine the like hood of drowning .This will help in the reducing drowning .This will help in reducing drowning rates and create a self environment.

2.LITERATURE SURVEY

2.1 Existing problem

Beginners, in particular, frequently struggle to breathe underwater, resulting in respiratory issues and, eventually, a drowning disaster. Drowning causes a higher mortality rate worldwide while causing no harm to children. Children under the age of six are found to have the highest global drowning fatality rates. These types of deaths rank third among all unexpected deaths worldwide, with approximately 1.2 million incident each year.

2.2 References

AngelEye. (2019). AngelEye – Distributors. Retrieved from: https://www.angeleye.it/news. php?id=28&newscat=10 Aquatics International. (2007).

Traumatic Experiences – Should we make our youngest lifeguards come face to face with death? Retrieved from: https://www.aquaticsintl.com/facilities/traumaticexperiences_o British Standards Institution. (2018).

BS EN 15288-1, Swimming pools for public use. Safety requirements for design. Retrieved from: https://shop.bsigroup.com/ProductDetail/?pid=000000000030360254

DDS Research Project 17 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 Drowning Prevention. (2017).

The Need. Retrieved from: https://www.drowningprevention.com.au/ 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).

Haizhou Li, Haizhou Li, Kar-Ann Toh and Liyuan Li. (2012). Advanced Topics in Biometrics, World Scientific Publishing Co. Pte. Ltd., ISBN-13 978-981-4287-84-5 Health and Safety Executive. (2018).

HSG179, Health and safety in swimming pools (Fourth edition). 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. J. Smith. (2016).

Recognition, vigilance and surveillance techniques. The Science of Beach Lifeguarding, Pages 183 – 192.

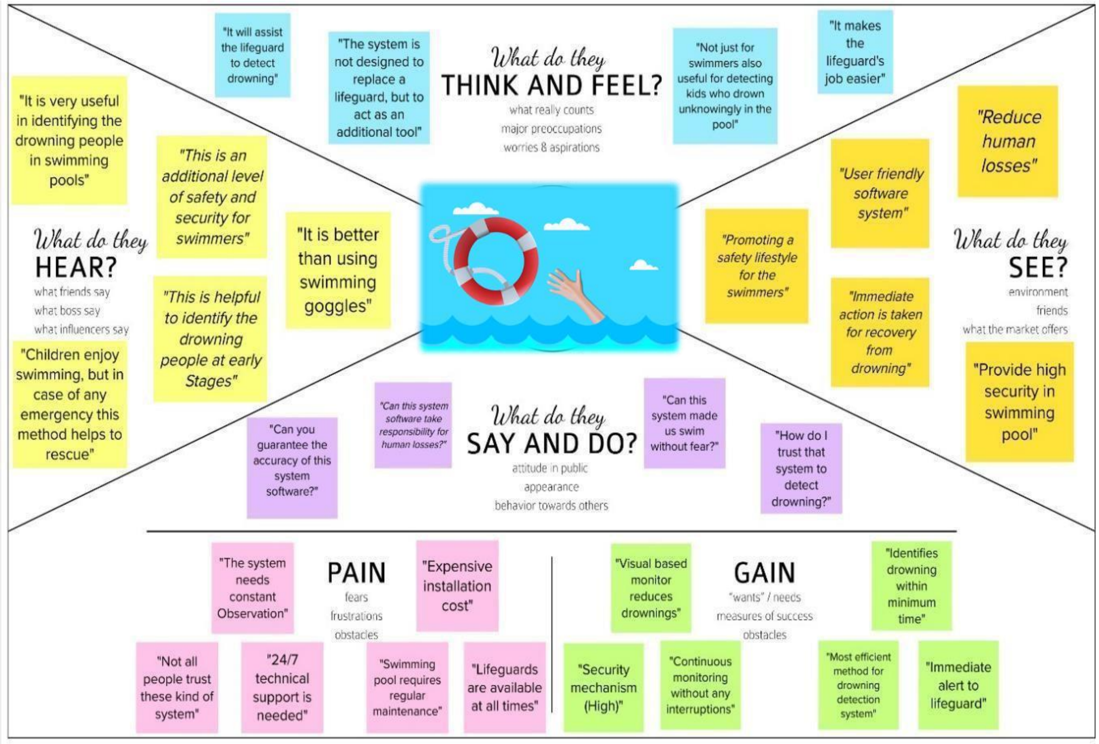
2.3 Problem Statement Definition

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

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user’s behaviour and attitudes. It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user’s perspective along with his or her goals and challenges.



"It will assist the lifeguard to detect drowning"

"The system is not designed to replace a lifeguard, but to act as an additional tool"

"Not just for swimmers also useful for detecting kids who drown unknowingly in the pool"

"It makes the lifeguard's job easier"

"It is very useful in identifying the drowning people in swimming pools"

"This is an additional level of safety and security for swimmers"

"It is better than using swimming goggles"

"This is helpful to identify the drowning people at early Stages"

"Children enjoy swimming, but in case of any emergency this method helps to rescue"

"Reduce human losses"

"User friendly software system"

"Promoting a safety lifestyle for the swimmers"

"Immediate action is taken for recovery from drowning"

"Provide high security in swimming pool"

"Can you guarantee the accuracy of this system software?"

"Can this system software take responsibility for human losses?"

"Can this system made us swim without fear?"

"How do I trust that system to detect drowning?"

"The system needs constant Observation"

"Expensive installation cost"

"Not all people trust these kind of system"

"24/7 technical support is needed"

"Swimming pool requires regular maintenance"

"Lifeguards are available at all times"

"Visual based monitor reduces drownings"

"Identifies drowning within minimum time"

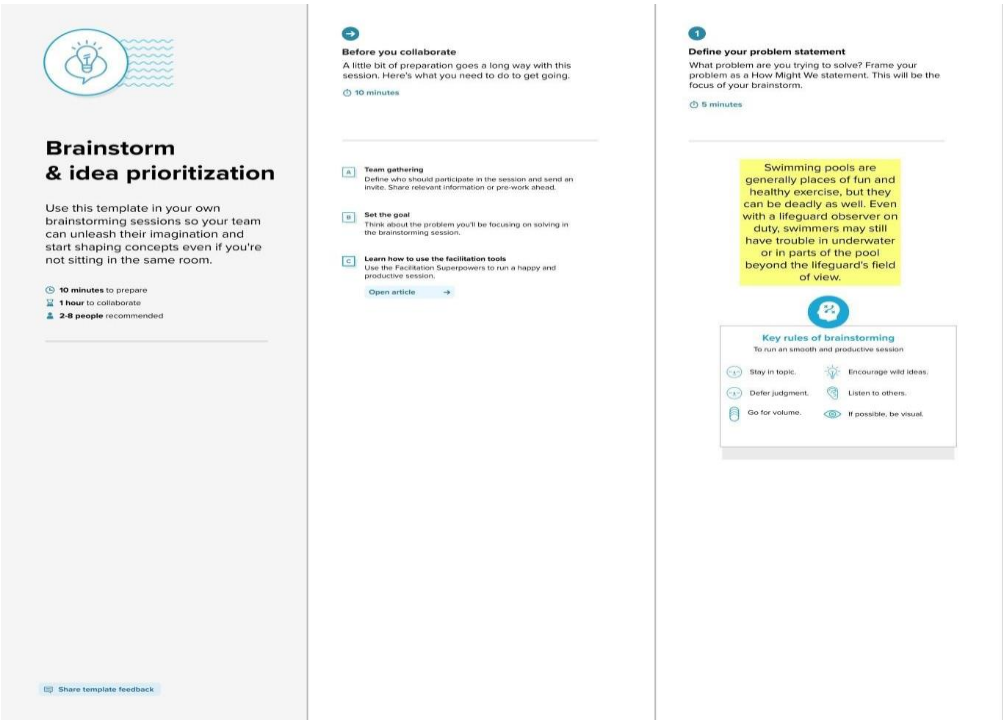
"Security mechanism (High)"

"Continuous monitoring without any interruptions"

"Most efficient method for drowning detection system"

"Immediate alert to lifeguard"

3.2 Ideation & Brainstorming



Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

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

Share template feedback

Before you collaborate

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

10 minutes

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

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

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

Open article →

1 **Define your problem statement**

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

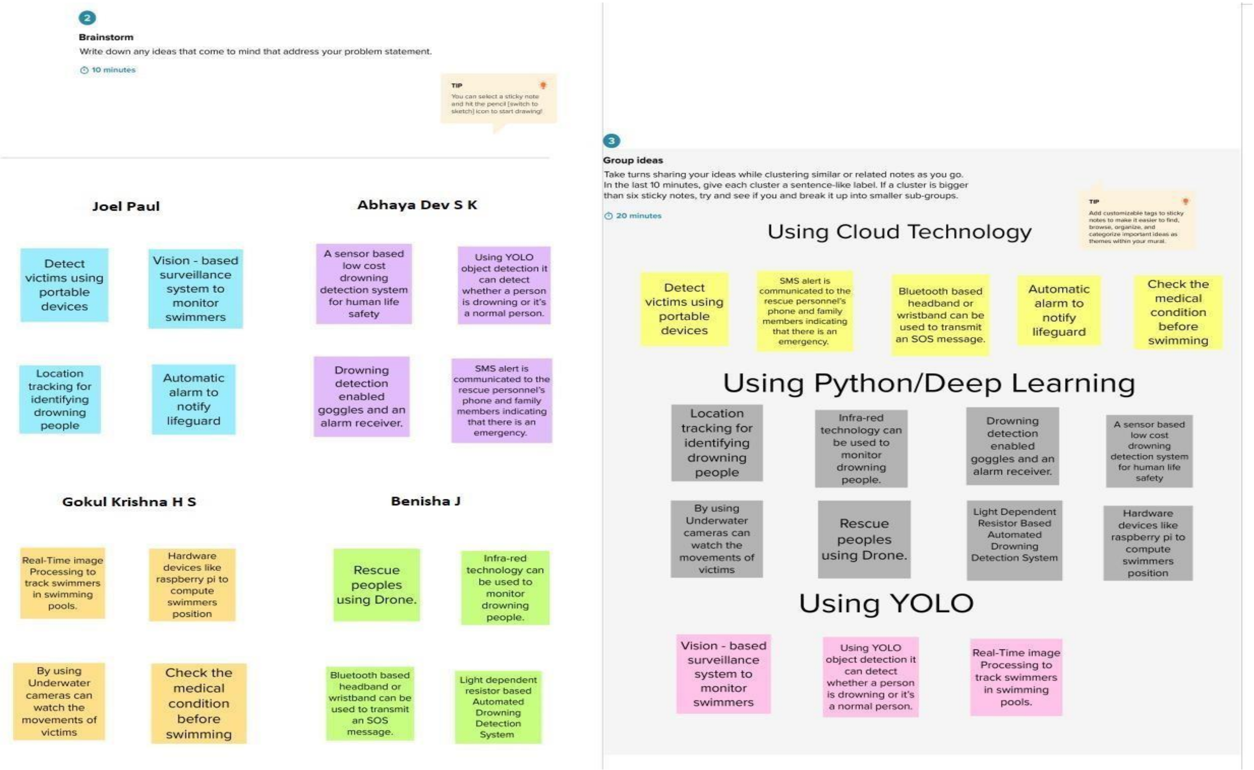
5 minutes

Swimming pools are generally places of fun and healthy exercise, but they can be deadly as well. Even with a lifeguard observer on duty, swimmers may still have trouble in underwater or in parts of the pool beyond the lifeguard's field of view.

Key rules of brainstorming

To run an smooth and productive session

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



2 **Brainstorm**

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP
You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

Joel Paul

Detect victims using portable devices

Vision - based surveillance system to monitor swimmers

Location tracking for identifying drowning people

Automatic alarm to notify lifeguard

Abhaya Dev S K

A sensor based low cost drowning detection system for human life safety

Using YOLO object detection it can detect whether a person is drowning or it's a normal person.

Drowning detection enabled goggles and an alarm receiver.

SMS alert is communicated to the rescue personnel's phone and family members indicating that there is an emergency.

Gokul Krishna H S

Real-Time image Processing to track swimmers in swimming pools.

Hardware devices like raspberry pi to compute swimmers position

By using Underwater cameras can watch the movements of victims

Check the medical condition before swimming

Benisha J

Rescue peoples using Drone.

Infra-red technology can be used to monitor drowning people.

Bluetooth based headband or wristband can be used to transmit an SOS message.

Light dependent resistor based Automated Drowning Detection System

3 **Group ideas**

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

TIP
Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

Using Cloud Technology

Detect victims using portable devices

SMS alert is communicated to the rescue personnel's phone and family members indicating that there is an emergency.

Bluetooth based headband or wristband can be used to transmit an SOS message.

Automatic alarm to notify lifeguard

Check the medical condition before swimming

Using Python/Deep Learning

Location tracking for identifying drowning people

Infra-red technology can be used to monitor drowning people.

Drowning detection enabled goggles and an alarm receiver.

A sensor based low cost drowning detection system for human life safety

By using Underwater cameras can watch the movements of victims

Rescue peoples using Drone.

Light Dependent Resistor Based Automated Drowning Detection System

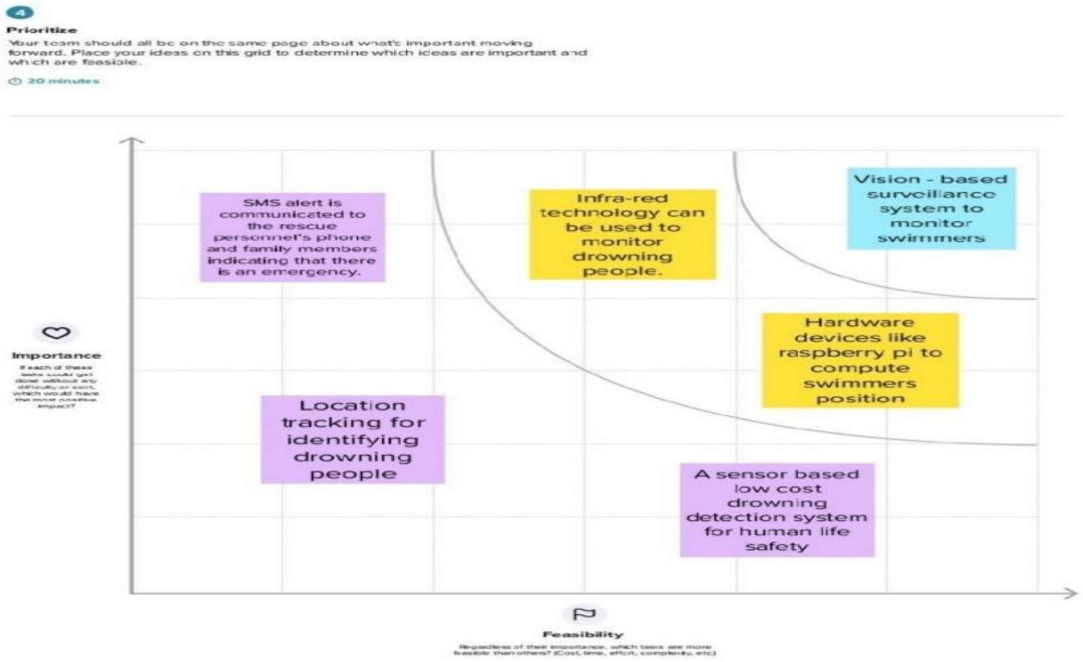
Hardware devices like raspberry pi to compute swimmers position

Using YOLO

Vision - based surveillance system to monitor swimmers

Using YOLO object detection it can detect whether a person is drowning or it's a normal person.

Real-Time image Processing to track swimmers in swimming pools.



4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

SMS alert is communicated to the rescue personnel's phone and family members indicating that there is an emergency.

Infra-red technology can be used to monitor drowning people.

Vision - based surveillance system to monitor swimmers

Hardware devices like raspberry pi to compute swimmers position

Location tracking for identifying drowning people

A sensor based low cost drowning detection system for human life safety

Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

Feasibility

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

3.3 Proposed Solution

Solution description:

Drowning detection system that detects every dangerous situation and accident. This software works in close integration with the cameras installed in the pool to continuously scan the pool. This system can also able to record all the activities in the pools and to classify critical situations from normal ones in order to keep track of what happened. The built-in notification system produces alarms within 10 seconds on onsmartwatches, phones, flashing lights and other configurable devices. Thus 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 a poolsafety system that reduces the risk of drowning.

Novelty:

Virtual eye has developed a novel idea of alerting the ambulance and another life guard if there is any delay in saving the person to death.

Customer satisfaction:

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, 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.

Business model:

There are many products currently available this regard. Our solution, once developed well, has enough possibility to become a good product to save drowning victims.

Scalability of the Solution:

Our proposed solution is very scalable i.e., in future, there are a lot of rooms for evolving our present model by Adding new features to enhance our system in the future.

4. REQUIREMENTS ANALYSIS

4.1 Functional Requirements

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Installation	Needed to be fixed under the water without Creating any disturbance to the people in the swimming pool.
FR-2	User registration	Register via Email/Phone number and get verified for further use
FR-3	Deduction	Either not moving or in unconscious state
FR-4	Support	Take swim tubes or take the help of rescuer
FR-5	Alert	Set alarm and send message through the application to life guard.
FR-6	Output	Vision based monitor Image, position and movement detection Drowning is detected Recue drowning people by Life Guard

4.2 Non-Functional requirements

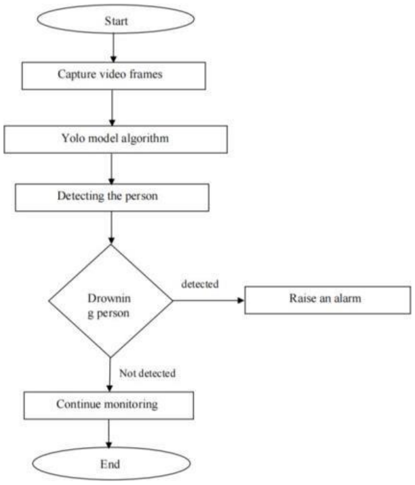
Following are the non-functional requirements of the proposed solution.

S No.	Non-Functional Requirement	Description
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.
2	**Security**	Lifeguards should be aware of the alertmessage to save the life of the swimmer
3	**Reliability**	Virtual eye lifeguard triggers an immediate prior alarm if a swimmer is in peril, helping to avoid panic even in critical situations.
4	**Performance**	The alarm is triggered when the swimmer is detected as drowning
5	**Availability**	Equipment and accessories include lifesaverrings, inflatable vests, aShepherd's Crook, life hooks, spine boards, rescue tubes, and a first aid kit. Remember to keep them accessible to quickly pull someone from the water water safely

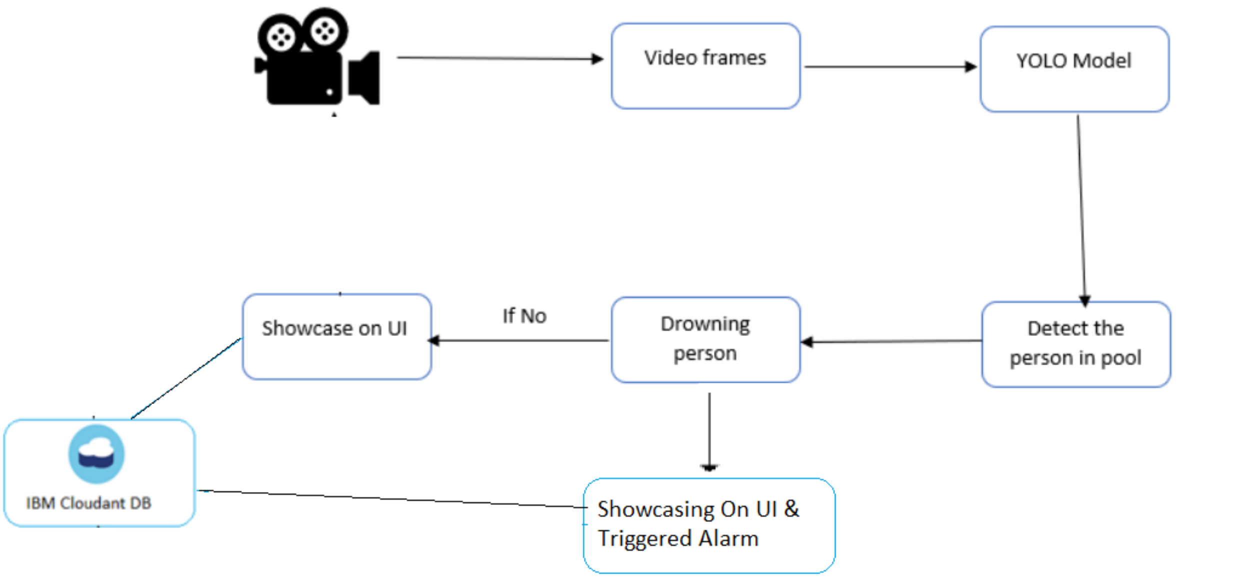
6	**Scalability**	Virtual eye lifeguard detects potentialdrownings and promptly notifies you. It features the latest artificial intelligence technology and adapts to the needs of the user.

5. PROJECT DEESIGN

5.1 Data Flow Diagram



5.2 Solution & Technical Architecture



5.3 User Stories

User Type	Functional Requirem ent	User Story Number	User Story / Task	Acceptance criteria	Priori ty	Release
Customer (Superviso r)	Installati on	USN-1	Theyset camera and install and configure the system in swimming pools	The software is installed and cameras are setup	High	Sprint-1
	Pre-processi ng	USN-2	Train and test the model	Trainthe model byusing datasets	High	Sprint-1

	Detection of drowning	USN-3	The swimmers can monitored by cameras	Camera surveillance	High	Sprint -2
		USN-4	Swimmers can be detected through the actions Detection of drowning	Detection of drowning	High	Sprint-2
	Alarm rings	USN -5	Alarm rings When the system detects drowning person	Alert the lifeguard	High	Sprint-3

User Type	Functional Requirements	User Story Number	User story/Task	Acceptance criteria	Priority	Release
Lifeguard	Saves the person	USN-6	The Lifeguard saves the swimmer who is drowning once the alarm rings	Saves the life of people	High	Sprint-3
Administrator	Register	USN-7	Register into the application	Admin can access the account	Medium	Sprint-2
	Login	USN-8	Login and manage the application	Manage system	Medium	Sprint-2
		USN-9	Stores the database	Storage the database	Medium	Sprint-2

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint planning & Estimation

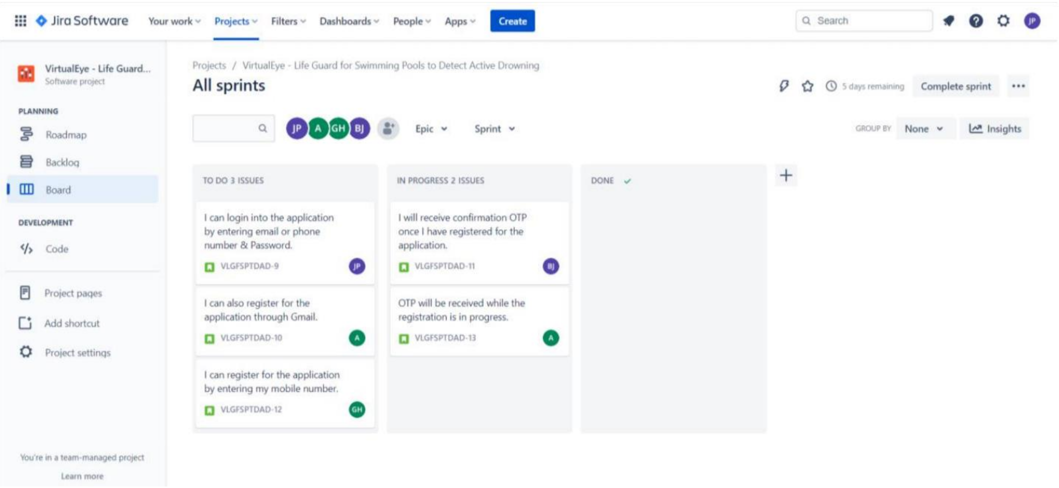
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Gokul Krishna H S
Sprint-1	Registration	USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Benisha J
Sprint-1	Registration	USN-3	As a user, can register for the application through through Facebook	2	Low	Abhaya Dev S K
Sprint-1	Registration	USN-4	As a user, can register for the application through Gmail	2	Medium	Joel Paul
Sprint-1	Login	USN-5	As a user, can log into the application by entering email & password	1	High	Abhaya Dev S K

Sprint-2	Dataset Collect	USN-6	Collect number of datasets and get accuracy	2	Medi um	Joel Paul
Sprint-2	Pre-processing	USN-7	The dataset is extracted	2	High	Benisha J
Sprint-2	Train the model	USN-8	Train the model.	4	High	Gokul Krishna H S
Sprint-2	Test the model	USN-9	Test the model	6	High	Abhaya Dev S K
Sprint-3	Detection	USN-10	Load the trained model.	3	High	Benisha J
Sprint-3	Detection	USN-11	Identify the person by collecting real- time data through a webcam.	5	Medi um	Joel Paul
Sprint-3	Detection	USN-12	classify it by using a trained model to predict the output	8	High	Benisha J
Sprint-4	Detection	USN-13	If person is drown the system will ring alarm to give signal	7	High	Gokul Krishna H S
Sprint-4	Detection	USN-14	As a User, I can detect the drowning person.	3	Medi um	Abhaya Dev S K
Sprint-4	Logout	USN-15	As a User, I can logout application	2	Low	Joel Paul

6.2 Sprint Delivery Schedule

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

6.3 Reports from JIRA Screen Shot 1:



Projects / VirtualEye - Life Guard for Swimming Pools to Detect Active Drowning

All sprints

TO DO 3 ISSUES

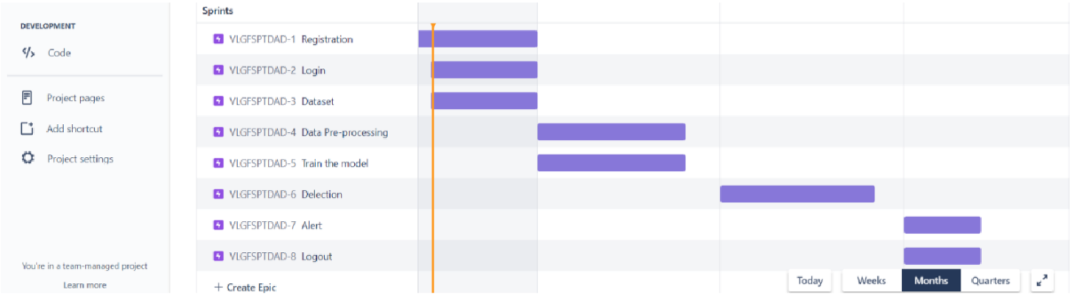
- I can login into the application by entering email or phone number & Password. VLGFSPTDAD-9 JP
- I can also register for the application through Gmail. VLGFSPTDAD-10 A
- I can register for the application by entering my mobile number. VLGFSPTDAD-12 GH

IN PROGRESS 2 ISSUES

- I will receive confirmation OTP once I have registered for the application. VLGFSPTDAD-11 BJ
- OTP will be received while the registration is in progress. VLGFSPTDAD-13 A

DONE ✓

Screen Shot 2:



Sprints

VLGFSPTDAD-1 Registration

VLGFSPTDAD-2 Login

VLGFSPTDAD-3 Dataset

VLGFSPTDAD-4 Data Pre-processing

VLGFSPTDAD-5 Train the model

VLGFSPTDAD-6 Delection

VLGFSPTDAD-7 Alert

VLGFSPTDAD-8 Logout

7. CODING & SOLUTIONING

7.1 Feature 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, 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 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 then an alert will be generated to attract lifeguards' attention.

Software Requirements:

Tensorflow

Keras

IBM Cloudant

Flas

OpenCV Python

Imutils

Progressbar

playsound

Hardware Requirements :

Processor	- Intel core i5
Hard Disk Space	- Minimum 100GB
RAM	- 4GB
Display	-14.1"colour monitor (LCD,CRT or LED)
Clock Speed	- 1.67 GHz

7.2 Feature 2

To design a system in an economically viable and easily accessible way that acts as a virtual eye to detect the drowning person in the swimming pool and alert the lifeguard using alarms to save the drowning person.

The system can be deployed in house, hotels, resorts, and swimming pool centres. The result is predicated in real-time, thus it can be used in emergency situations.

8 . TESTING

8.1 Testcase

Test case	Feature type	Page	Test scenario	Step To Execute	Test	Expected Result	Actual Result
Login page_TC_001	Functional	Home page	Verify user is see the login/sign up popup when user clicked on my Account	1.Enter URL and click go 2.click on my account dropdown 3.verify login/signup popup display or not	Login.html	login/sign popup should display	Result Working
Login page_TC_002	Functional	Home page	Verify the UI element in login/sign up popup	1.Enter URL and click go 2.Click on My account dropdown	Login.html	Application should show below element: a.email text box b.passwo	Working

				3.Verify login/signup below UI a.Mail text box. b.password text box c.L% in button d.New customer?Create an account e.Lost passwor		rd text box c.login button with orange colour d.New Customer? Create account e.Last password ?Recovery password		

	18-Nov-22 PNT2022TMID51944
	Virtual eye-Lifeguard for swimming Pools to detect Active Drowning. 4 marks

Test case ID	Feature Type		Test Scenario	Steps TO Execute	Test	Expected Result	Actual Result
LP_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/Sign up popup displayed or not	Login.html	Login/Signup Popup should display	Working as Expected
LP_TC_002	Functional	Home Page	Verify the UI elements in Login/Signup popup	1.Enter the URL and dick go 2.Click on My Account dropdown 3.Verify login/Signup popup with below UI elements: a email text box password textbox 4.L%in button 5.New customer? Create account link 6.Last password? Recovery password link	Login.html	Application should show below elements: a. email text box b. Password text box c.log in button with orange colour d. new custotner? Create account link e. Last password? Recovery password link	Working as Expected
LP_TC_003	Functional	Home page	Verify user is able to log into application with Valid credentials	1.Enter URL and dick 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: Joelpaul413@gmail password: 12345	User should navigate to prediction homepage	Working as Expected
LP_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 •n button	Username: gokulkrishnagk@gmail password: 12345678	Application should show 'Incorrect email or password ' validation message.	Working as Expected

LP_TC_005	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: abhayadevsk@gmail password: ADev	Application should show Incorrect email or password validation message.	Working as Expected
LP_TC_006	Functional	Login page	Verify user is able to into application with lnvalid 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 I in button	Username: Benishaj@gmail password: 123456	Application should show 'Incorrect email or password ' validation message.	Working as Expected
PP_TC_007	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 Expected

8.2 User Acceptance testing

1. 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

5. Test Case Analysis

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2

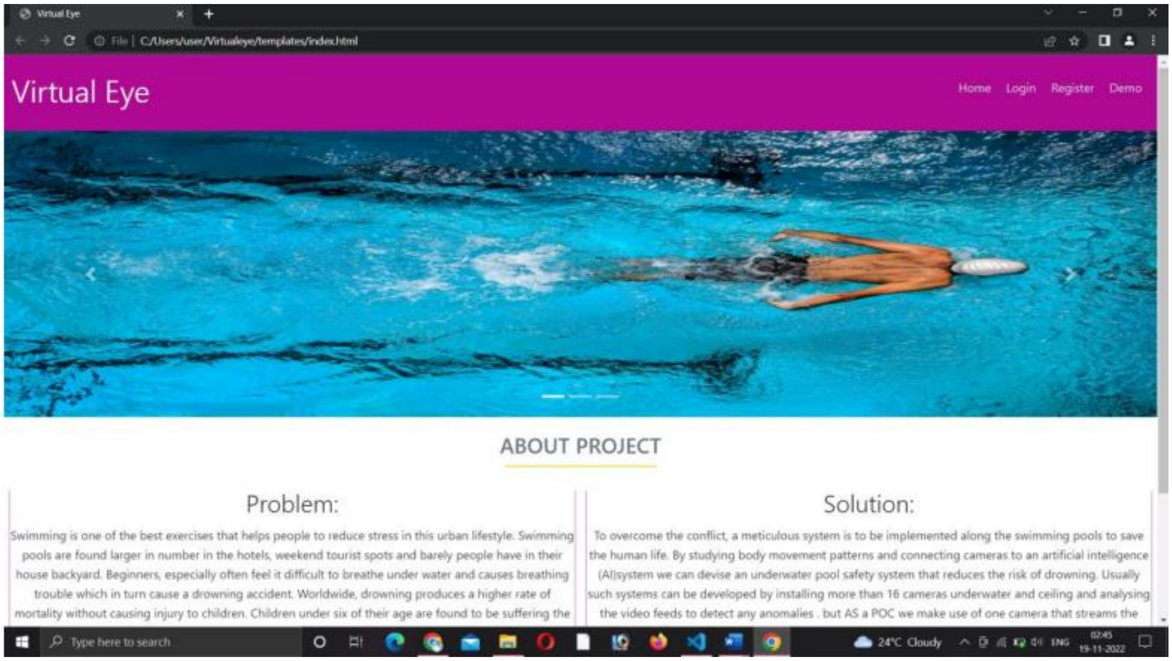
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77

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

9. RESULTS

9.1 Performance Metrics

Home :-



ABOUT PROJECT

Problem:

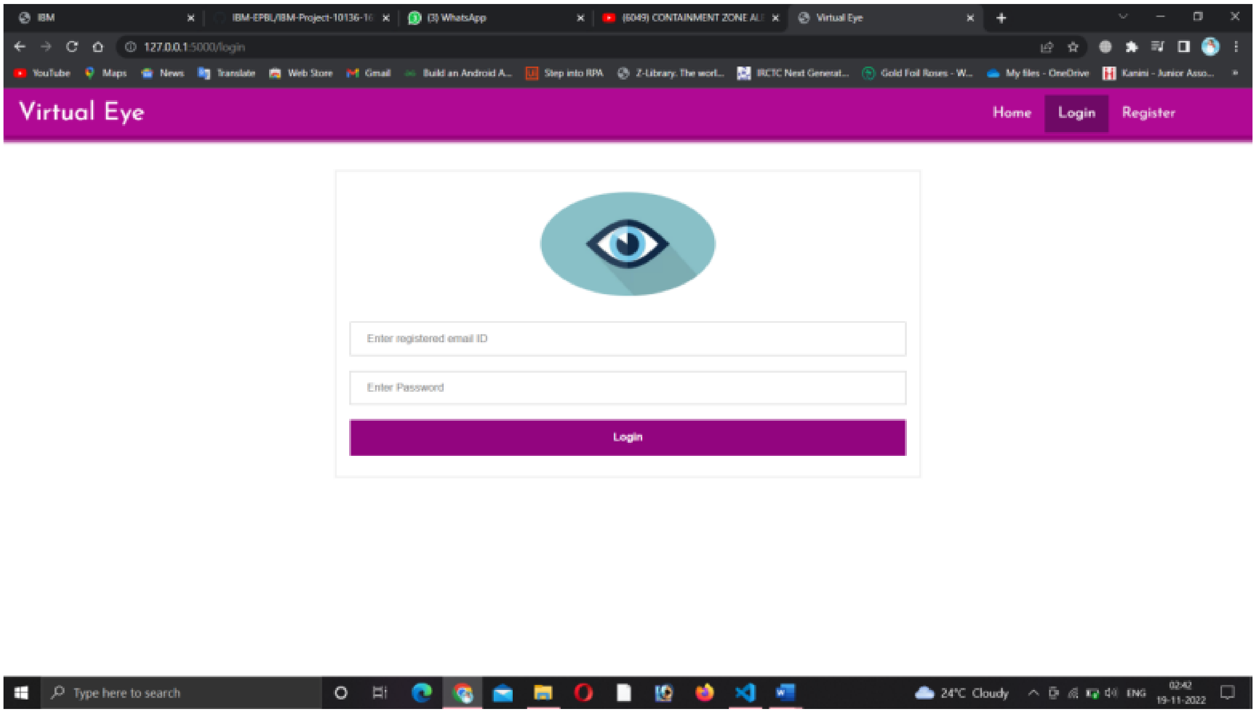
Swimming is one of the best exercises that helps people to reduce stress in this urban lifestyle. Swimming pools are found larger in number in the hotels, weekend tourist spots and barely people have in their house backyard. Beginners, especially often feel it difficult to breathe under water and causes 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

Solution:

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

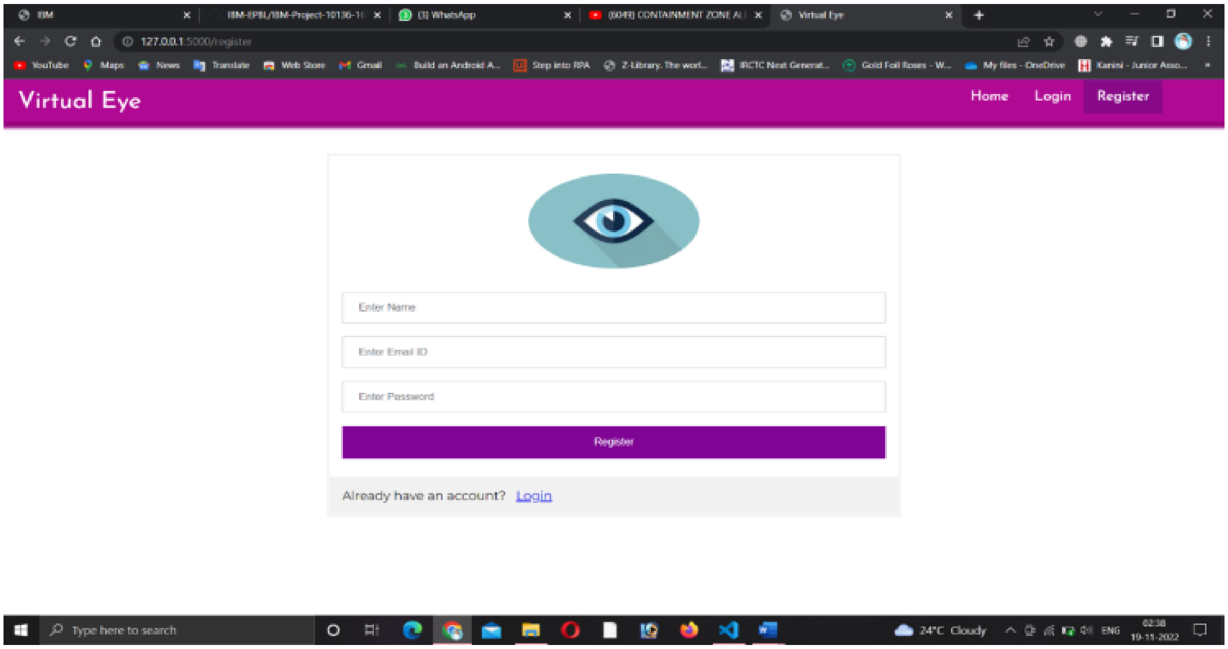
Login :-

- User can login into his account using Username and Password.

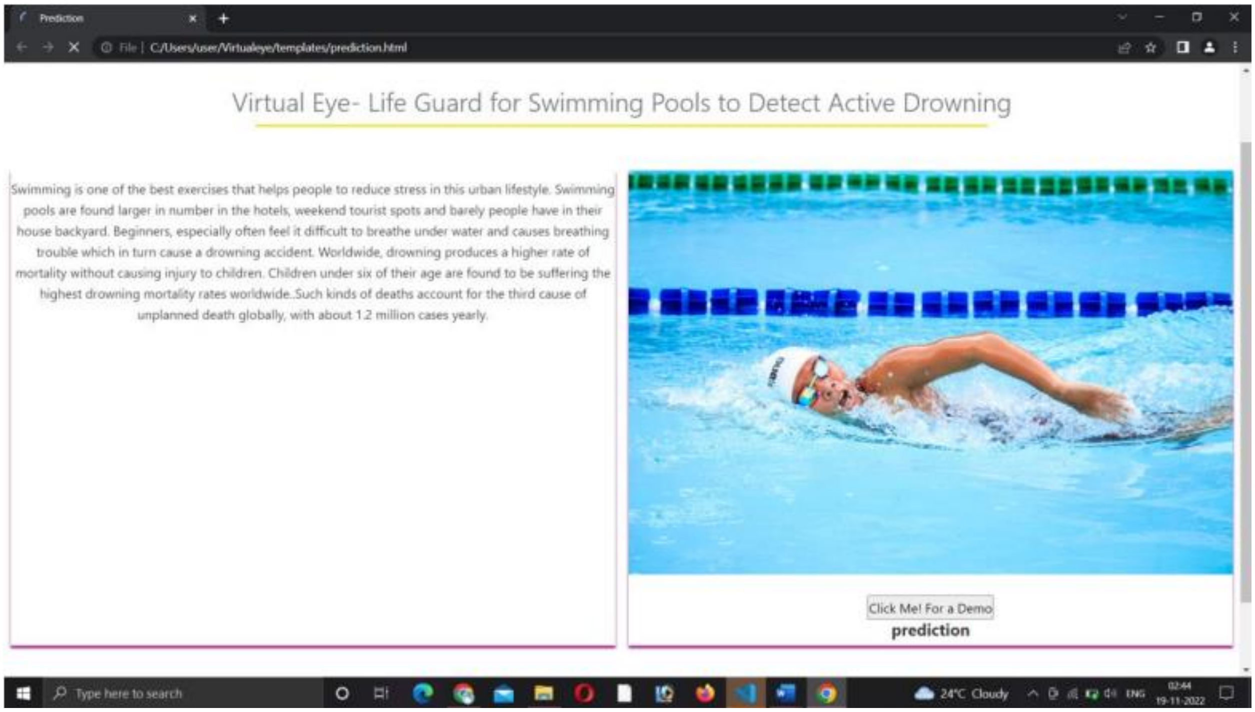


Register :-

- User can register using personal details.



Demo :-



Virtual Eye- Life Guard for Swimming Pools to Detect Active Drowning

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.

10. ADVANTAGES & DISADVANTAGES

Advantages

The use of deep learning gives accurate results after training the model.

YOLOv3 model is fast and can process up to 45 frames per second.

Disadvantages

YOLO has low recall value and struggles to detect very close objects.

11. CONCLUSION

In this project, we have developed a deep learning system using YOLOv3 model to predict if a person is drowning or not. The system is connected to IBM cloud services the user can access through a web application along with the alarm feature system to notify the lifeguard.

12. FUTURE SCOPE

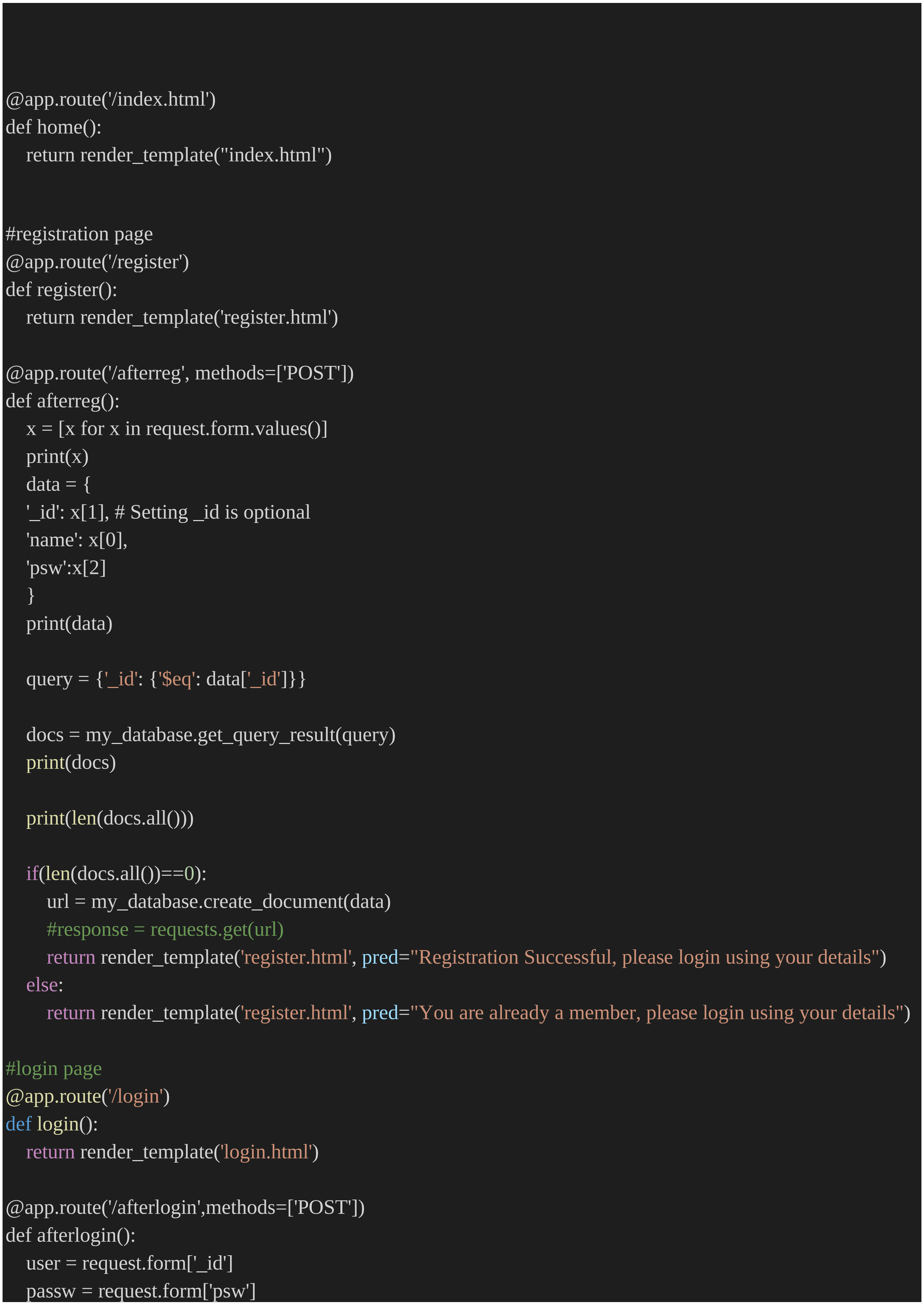
The project can be further extended by deploying multiple cameras underwater to improve accuracy of prediction. The processing speed of the model can be improved to produce the result faster.

13. RESULTS

Source Code

App.py

```
import re
import numpy as np
import os
from flask import Flask, app,request,render_template
import keras
from keras import models
from keras.models import load_model
from keras.preprocessing import image
from tensorflow.python.ops.gen_array_ops import concat
from keras.applications.inception_v3 import preprocess_input
import cvlib as cv
from cvlib.object_detection import draw_bbox
import cv2
import time
import numpy as np
from playsound import playsound
import requests
from flask import Flask, request, render_template, redirect, url_for
#Loading the model

from cloudant.client import Cloudant

# Authenticate using an IAM API key
client = Cloudant.iam('2eb40045-a8d6-450d-9d24-52cc7cbb2810-
bluemix','Ud0wunTPOI_8h5ZtEqi1IXk1gIKeYLmpUsCn0EeO8T4z', 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])

    #there is still need to tweek the threshold
    #this threshold is for checking how much the centre has moved

    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)
```

detect.py

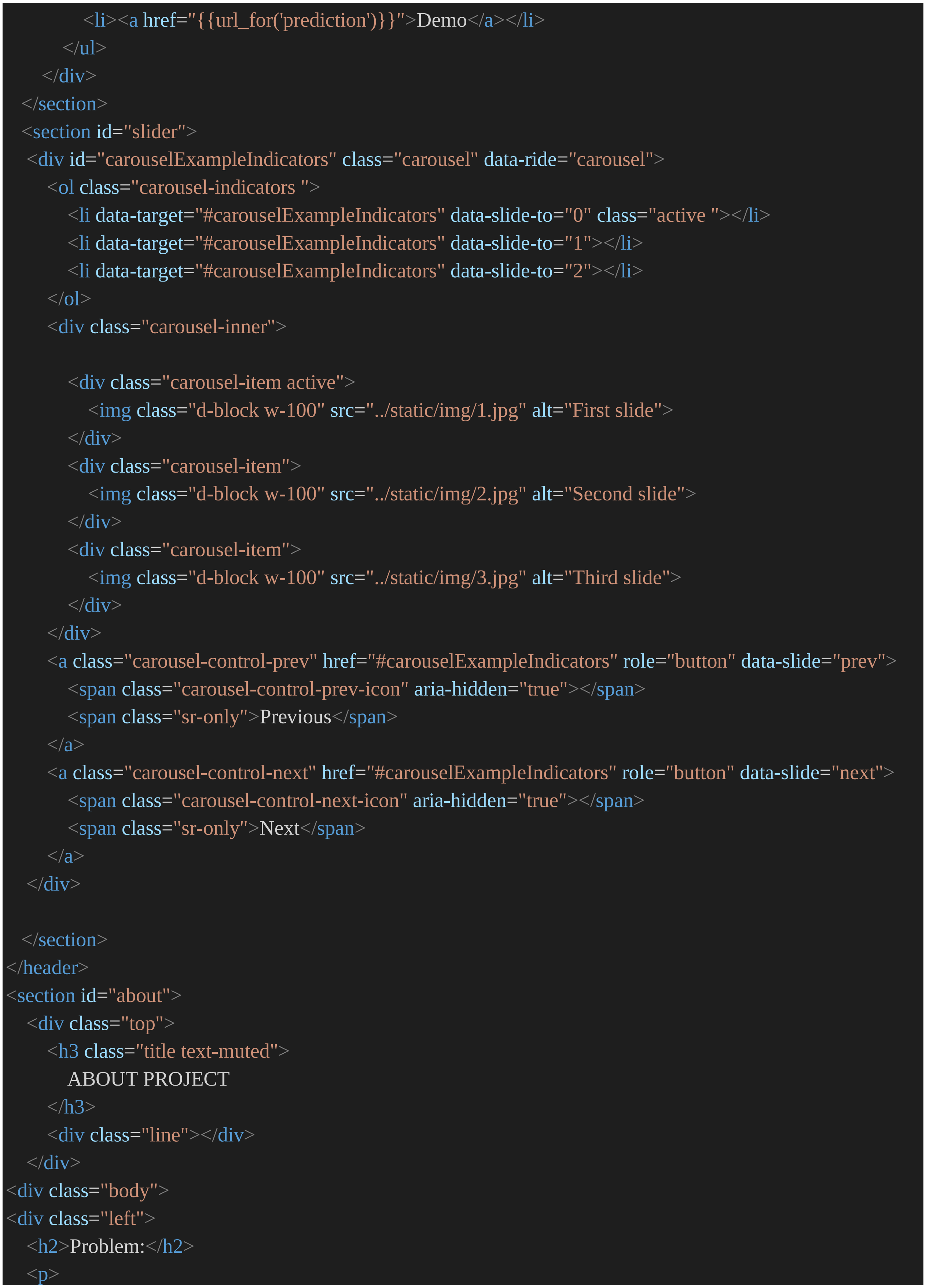
```
import cvlib as cv
from cvlib.object_detection import draw_bbox
import cv2
import time
import numpy as np
from playsound import playsound
#for PiCamera
#from picamera Import PiCamera
```

```
#camera = PiCamera
#camera.start_preview()
# open webcam
webcam = cv2.VideoCapture(0)

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])

            #there is still need to tweek the threshold
            #this threshold is for checking how much the centre has moved

            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(r'C:\Users\user\Virtualeye\alarm.mp3')

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

# release resources
```

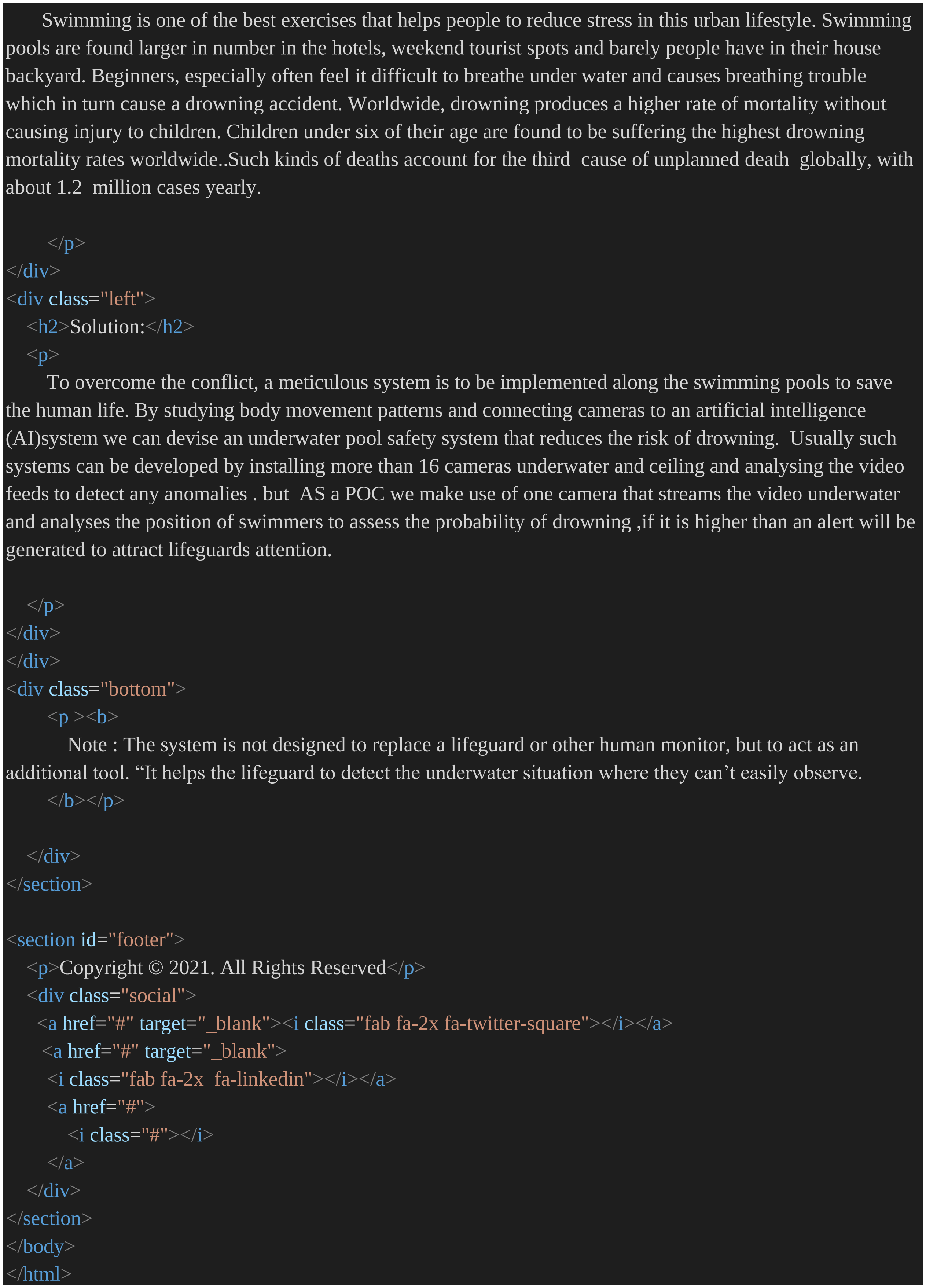
```
webcam.release()
cv2.destroyAllWindows()
```

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="style.css">
    <!-- <script defer src="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('prediction')}}">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">
        <img class="d-block w-100" src="../static/img/1.jpg" alt="First slide">
      </div>
      <div class="carousel-item">
        <img class="d-block w-100" src="../static/img/2.jpg" alt="Second slide">
      </div>
      <div class="carousel-item">
        <img class="d-block w-100" src="../static/img/3.jpg" alt="Third slide">
      </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 than an 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 helps the lifeguard to detect the underwater situation where they can’t easily observe.
    </b></p>

  </div>
</section>

<section id="footer">
  <p>Copyright © 2021. 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>
```

Register.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 rel="stylesheet" href="style.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: rgb(176, 9, 148);
      color: rgb(110, 5, 113);
      box-shadow: 0px 8px 4px rgb(151, 3, 124);
      overflow: hidden;
      padding-left:20px;
      font-family: 'Josefin Sans';
      font-size: 2vw;
      width: 100%;
      height:8%;
      text-align: center;
    }
    .topnav {
  overflow: hidden;
  background-color: rgb(151, 3, 124);
}
```

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

.topnav-right a:hover {
  background-color: rgb(127, 5, 149);
  color: rgb(255, 255, 255);
}

.topnav-right a.active {
  background-color: #880988;
  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;
  }
.login{
  margin-top:100px;
}
form {border: 3px solid #f1f1f1; 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: rgb(127, 5, 149);
  color: white;
  padding: 14px 20px;
  margin-bottom:8px;
  border: none;
  cursor: pointer;
  width: 100%;
}

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" >

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

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


  <form action="{{url_for('afterreg')}}" method="post">
    <div class="imgcontainer">
      <img src="https://cdn.digitalhealth.net/wp-content/uploads/2017/03/eye_image_generic_555.jpg"
alt="Avatar" class="avatar">
    </div>

    <div class="container">
      <input type="text" placeholder="Enter Name" name="name" required><br>
      <input type="email" placeholder="Enter Email ID" name="_id" required><br>
      <input type="password" placeholder="Enter Password" name="psw" required>

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

    </div>
    <div class="container" style="background-color:#f1f1f1">
    <div class="psw">Already have an account?&nbsp; &nbsp;<a href="login.html">Login</a></div >
  </div>
  </form>

</div>



</body>
```

```
</html>
```

Login.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 rel="stylesheet" href="{{ url_for('static', filename='css/style.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: rgb(176, 9, 148);
      color: white;
      box-shadow: 0px 8px 4px rgb(151, 3, 124);
      overflow: hidden;
      padding-left:20px;
      font-family: 'Josefin Sans';
      font-size: 2vw;
      width: 100%;
      height:8%;
      text-align: center;
    }
.topnav {
  overflow: hidden;
  background-color: rgb(146, 5, 127);
}
```

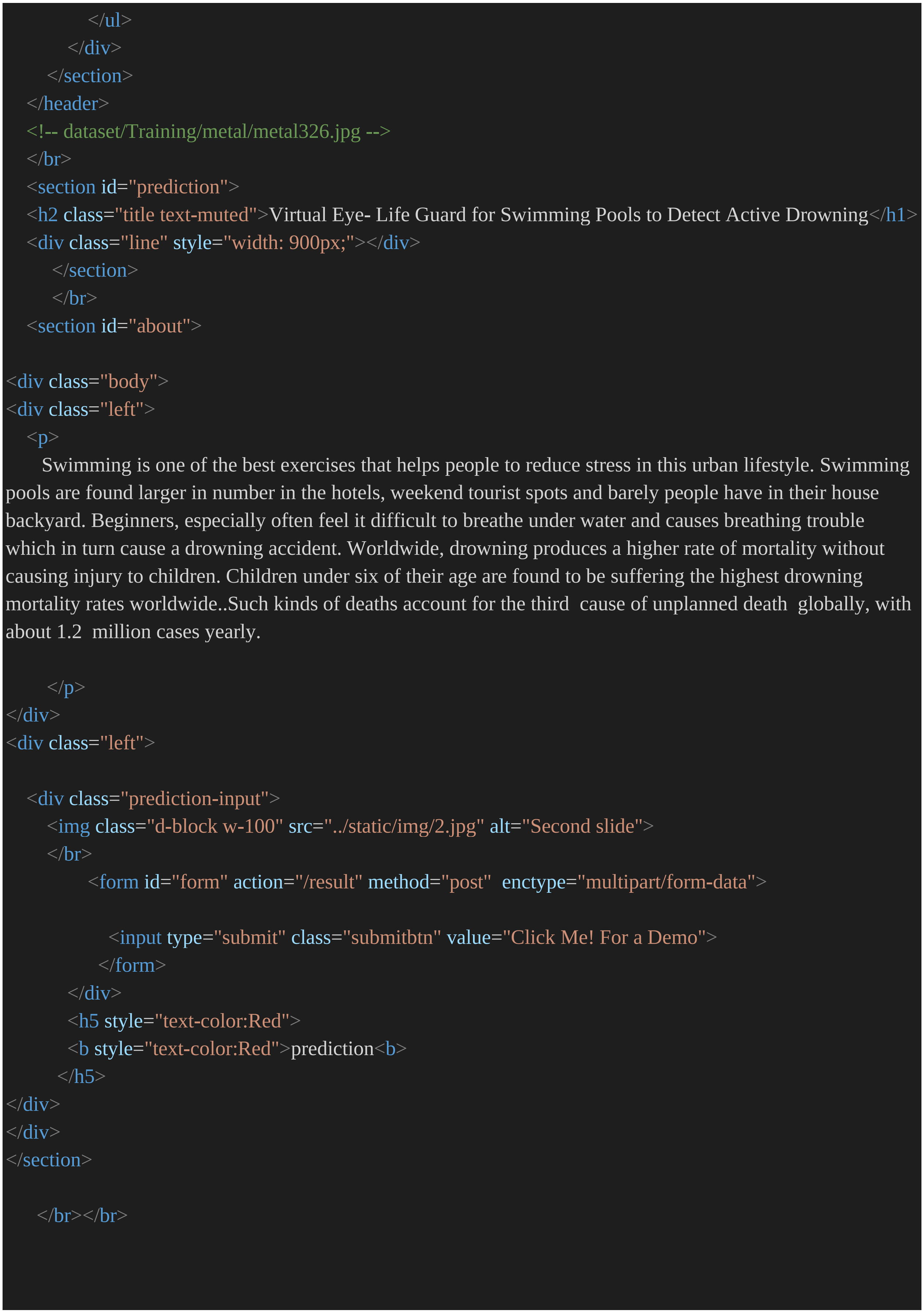
```
.topnav-right a {
  float: left;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  font-size: 18px;
}

.topnav-right a:hover {
  background-color: #ddd;
  color: rgb(113, 8, 115);
}

.topnav-right a.active {
  background-color: #6f0966;
  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;
  }
.login{
  margin-top:100px;
}
form {border: 3px solid #f1f1f1; 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: rgb(146, 5, 127);
  color: white;
  padding: 14px 20px;
  margin-bottom:8px;
  border: none;
  cursor: pointer;
  width: 100%;
  font-weight:bold;
}

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="index.html">Home</a>
    <a class="active" href="login.html">Login</a>
    <a  href="register.html">Register</a>

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


  <form action="{{url_for('afterlogin')}}" method="post">
    <div class="imgcontainer">
      <img src="https://cdn.digitalhealth.net/wp-content/uploads/2017/03/eye_image_generic_555.jpg"
alt="Avatar" class="avatar">
    </div>

    <div class="container">
      <input type="email" placeholder="Enter registered email ID" name="_id" required><br>

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

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

    </div>
  </form>

</div>



</body>
</html>
```

Prediction.html

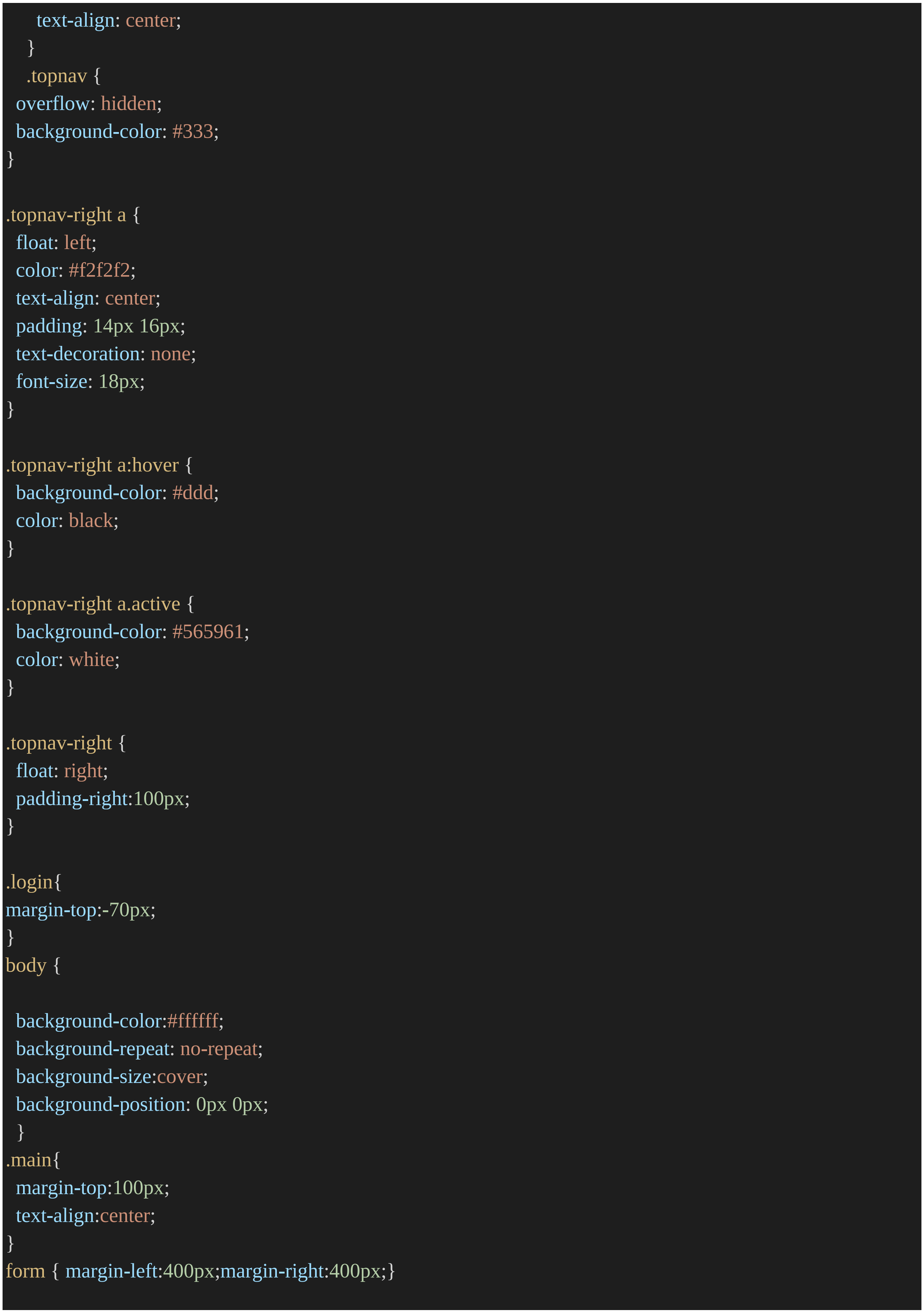
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!--Bootstrap -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
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="style.css">


    <script defer src="JScript.js"></script>
    <title>Prediction</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="index.html">Home</a></li>
                <li><a href="logout.html">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 for Swimming 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">
        <img class="d-block w-100" src="../static/img/2.jpg" alt="Second slide">
        </br>
                <form id="form" action="/result" method="post"  enctype="multipart/form-data">

                    <input type="submit" class="submitbtn" value="Click Me! 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 © 2021. All Rights Reserved</p>

    </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: rgb(176, 9, 148);
      color: white;
      box-shadow: 0px 8px 4px rgb(151, 3, 124)y;
      overflow: hidden;
      padding-left:20px;
      font-family: 'Josefin Sans';
      font-size: 2vw;
      width: 100%;
      height:8%;
```

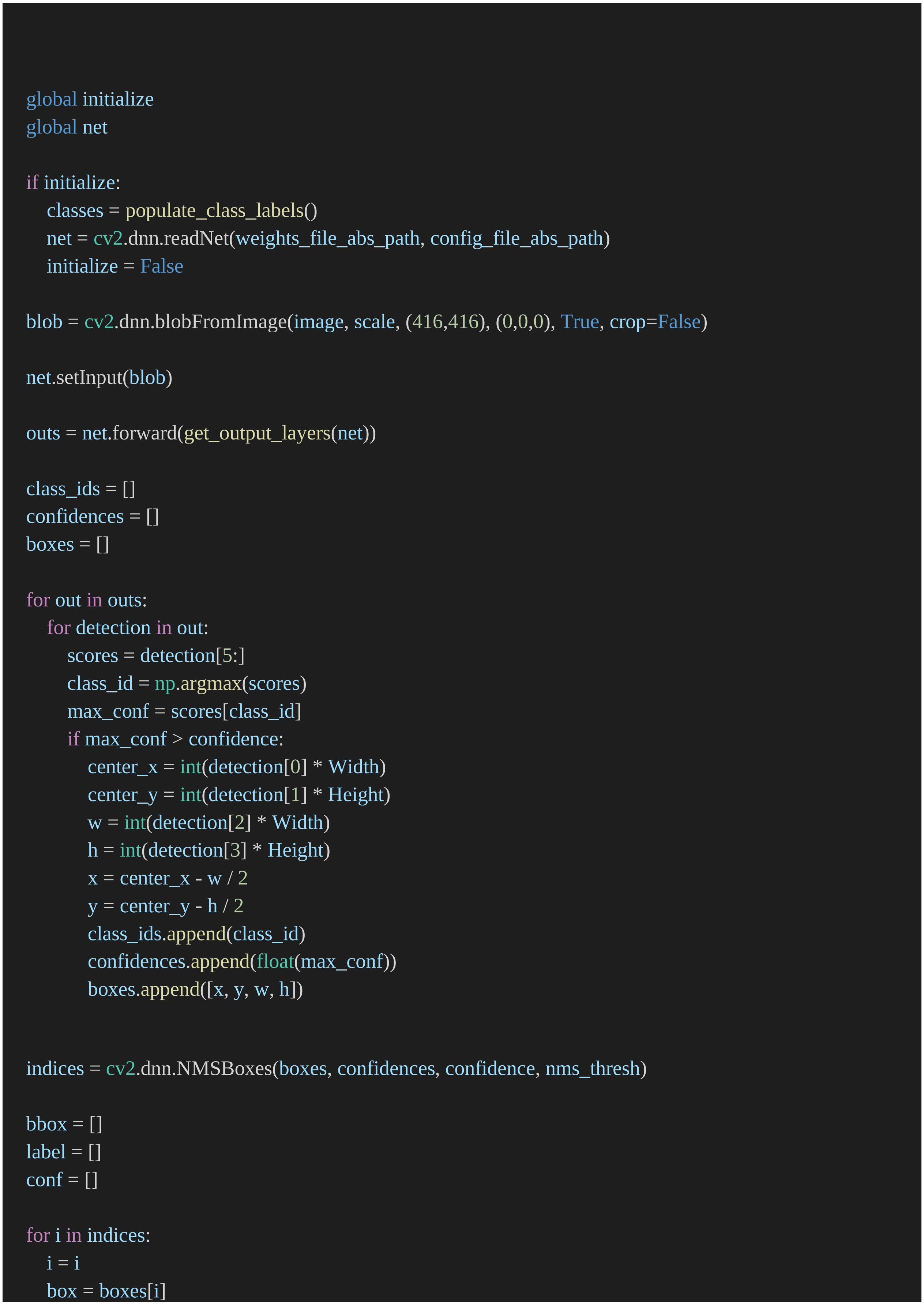


```
     text-align: center;
    }
    .topnav {
  overflow: hidden;
  background-color: #333;
}

.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="index.html">Home</a>
    <a href="login.html">Login</a>
    <a href="register.html">Register</a>
  </div>
</div>
<div class="main">
<h1>Successfully Logged Out!</h1>
<h3 style="color:#4CAF50">Login for more information<h3>

  <a href="login.html"><button type="submit">Login</button></a>
</form>
</div>

</body>
</html>
```

Object_detection.py

```
#import necessary packages
import cv2
import os
import numpy as np
from .utils import download_file

initialize = True
net = None
dest_dir = os.path.expanduser('~') + os.path.sep + '.cvlib' + os.path.sep + 'object_detection' + os.path.sep +
'yolo' + os.path.sep + 'yolov3'
classes = None
#colors are BGR instead of RGB in python
COLORS = [0,0,255], [255,0,0]

def populate_class_labels():

    #we are using a pre existent classifier which is more reliable and more efficient than one
    #we could make using only a laptop
    #The classifier should be downloaded automatically when you run this script
    class_file_name = 'yolov3_classes.txt'
    class_file_abs_path = dest_dir + os.path.sep + class_file_name
    url = 'https://github.com/Nico31415/Drowning-Detector/raw/master/yolov3.txt'
    if not os.path.exists(class_file_abs_path):
        download_file(url=url, file_name=class_file_name, dest_dir=dest_dir)
    f = open(class_file_abs_path, 'r')
    classes = [line.strip() for line in f.readlines()]

    return classes


def get_output_layers(net):

    #the number of output layers in a neural network is the number of possible
    #things the network can detect, such as a person, a dog, a tie, a phone...
    layer_names = net.getLayerNames()

    output_layers = [layer_names[i - 1] for i in net.getUnconnectedOutLayers()]

    return output_layers


def draw_bbox(img, bbox, labels, confidence, Drowning, write_conf=False):

    global COLORS
    global classes

    if classes is None:
        classes = populate_class_labels()

    for i, label in enumerate(labels):
        #if the person is drowning, the box will be drawn red instead of blue
        if label == 'person' and Drowning:
            color = COLORS[0]
            label = 'DROWNING'
        else:
            color = COLORS[1]

        if write_conf:
            label += ' ' + str(format(confidence[i] * 100, '.2f')) + '%'


        #you only need to points (the opposite corners) to draw a rectangle. These points
        #are stored in the variable bbox
        cv2.rectangle(img, (bbox[i][0],bbox[i][1]), (bbox[i][2],bbox[i][3]), color, 2)

        cv2.putText(img, label, (bbox[i][0],bbox[i][1]-10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 2)

    return img

def detect_common_objects(image, confidence=0.5, nms_thresh=0.3):

    Height, Width = image.shape[:2]
    scale = 0.00392

    global classes
    global dest_dir

    #all the weights and the neural network algorithm are already preconfigured
    #as we are using YOLO

    #this part of the script just downloads the YOLO files
    config_file_name = 'yolov3.cfg'
    config_file_abs_path = dest_dir + os.path.sep + config_file_name

    weights_file_name = 'yolov3.weights'
    weights_file_abs_path = dest_dir + os.path.sep + weights_file_name

    url = 'https://github.com/Nico31415/Drowning-Detector/raw/master/yolov3.cfg'

    if not os.path.exists(config_file_abs_path):
        download_file(url=url, file_name=config_file_name, dest_dir=dest_dir)

    url = 'https://pjreddie.com/media/files/yolov3.weights'

    if not os.path.exists(weights_file_abs_path):
        download_file(url=url, file_name=weights_file_name, dest_dir=dest_dir)
```



```
global initialize
global net

if initialize:
    classes = populate_class_labels()
    net = cv2.dnn.readNet(weights_file_abs_path, config_file_abs_path)
    initialize = False

blob = cv2.dnn.blobFromImage(image, scale, (416,416), (0,0,0), True, crop=False)

net.setInput(blob)

outs = net.forward(get_output_layers(net))

class_ids = []
confidences = []
boxes = []

for out in outs:
    for detection in out:
        scores = detection[5:]
        class_id = np.argmax(scores)
        max_conf = scores[class_id]
        if max_conf > confidence:
            center_x = int(detection[0] * Width)
            center_y = int(detection[1] * Height)
            w = int(detection[2] * Width)
            h = int(detection[3] * Height)
            x = center_x - w / 2
            y = center_y - h / 2
            class_ids.append(class_id)
            confidences.append(float(max_conf))
            boxes.append([x, y, w, h])


indices = cv2.dnn.NMSBoxes(boxes, confidences, confidence, nms_thresh)

bbox = []
label = []
conf = []

for i in indices:
    i = i
    box = boxes[i]
```

```
        x = box[0]
        y = box[1]
        w = box[2]
        h = box[3]
        bbox.append([round(x), round(y), round(x+w), round(y+h)])
        label.append(str(classes[class_ids[i]]))
        conf.append(confidences[i])

    return bbox, label, conf
```

utils.py

```
import requests
import progressbar as pb
import os

def download_file(url, file_name, dest_dir):

    if not os.path.exists(dest_dir):
        os.makedirs(dest_dir)

    full_path_to_file = dest_dir + os.path.sep + file_name

    if os.path.exists(dest_dir + os.path.sep + file_name):
        return full_path_to_file

    print("Downloading " + file_name + " from " + url)

    try:
        r = requests.get(url, allow_redirects=True, stream=True)
    except:
        print("Could not establish connection. Download failed")
        return None

    file_size = int(r.headers['Content-Length'])
    chunk_size = 1024
    num_bars = round(file_size / chunk_size)

    bar = pb.ProgressBar(maxval=num_bars).start()

    if r.status_code != requests.codes.ok:
        print("Error occurred while downloading file")
        return None
```

```
count = 0

with open(full_path_to_file, 'wb') as file:
    for chunk in  r.iter_content(chunk_size=chunk_size):
        file.write(chunk)
        bar.update(count)
        count +=1

return full_path_to_file
```

Requirements.txt

```
cloudant==2.15.0
Flask==2.0.3
numpy==1.16.4
opencv_python==4.1.0.25
playsound==1.3.0
progressbar33==2.4
tensorflow==1.15.2
```

GitHub Link :

https://github.com/IBM-EPBL/IBM-Project-19573-1659700838

Demo Link :

https://www.youtube.com/watch?v=ZjxeT6F2-co