VirtualEye - Life Guard For Swimming Pools To Detect Active Drowning Using IBM Cloud

1. INTRODUCTION

a. Overview

In modern metropolitan lifestyle, swimming is one of the best activities 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 incidents each year, these types of deaths rank third among all unexpected deaths worldwide.

b. Purpose

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

2. LITERATURE SURVEY

a. Existing Problem

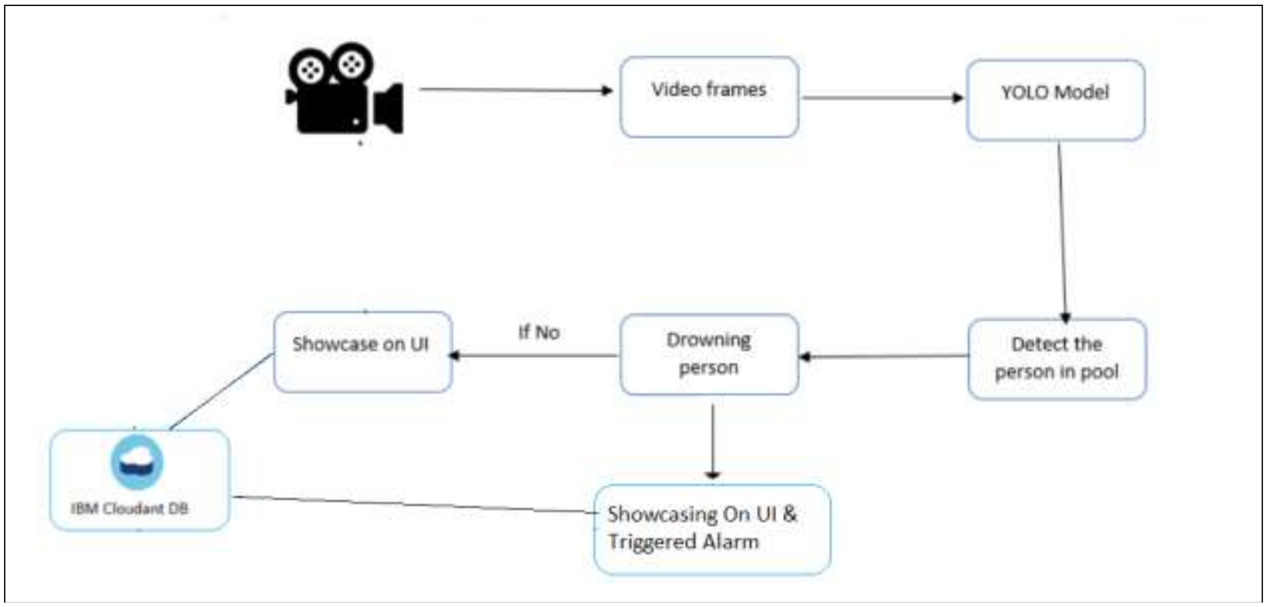
Beginners, in particular, frequently struggle to breathe underwater, resulting in respiratory issues and, eventually, a drowning disaster. Drowning causes a higher mortality rate worldwide while causing no harm to children. Children under the age of six are found to have the highest global drowning fatality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

b. Proposed Solution

This project builds a model which uses YOLO-based Convolutional Neural Network for object detection. The project makes use of the recent YOLOv3 model for implementation. By studying body movement patterns and connecting cameras to artificial intelligence (AI) systems, the project devises an underwater pool safety system that reduces the risk of drowning. For the scope of this project, only one camera is used.

3. Theoretical Analysis

3.1 Block Diagram



3.2 Hardware and Software

Software Requirements

- ∉ IBM Cloud
- ∉ TensorFlow
- ∉ Keras
- ∉ Python Flask

Hardware Requirements

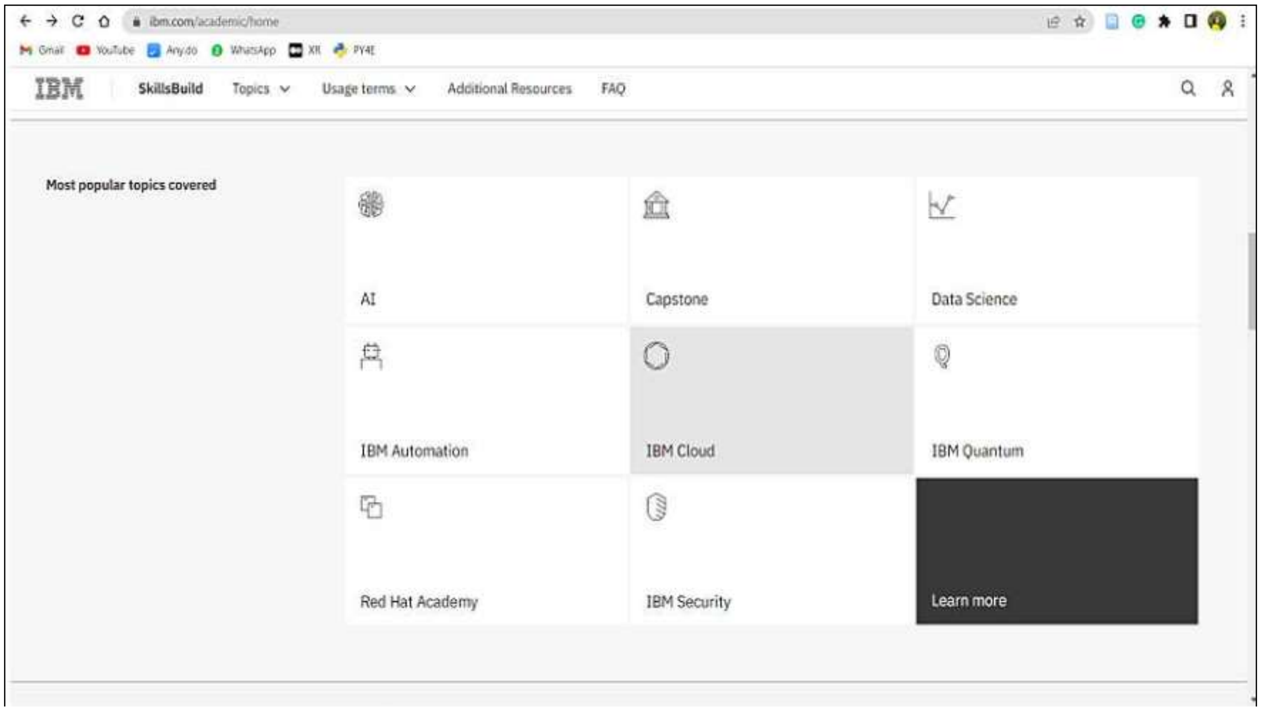
Processor : Intel Core i3

HardDisk Space	: minimum 100 GB
RAM	: 4 Gb
Display	: 14.1” colour monitor (LCD, CRT or LED)
Clock Speed	: 1.67 GHz

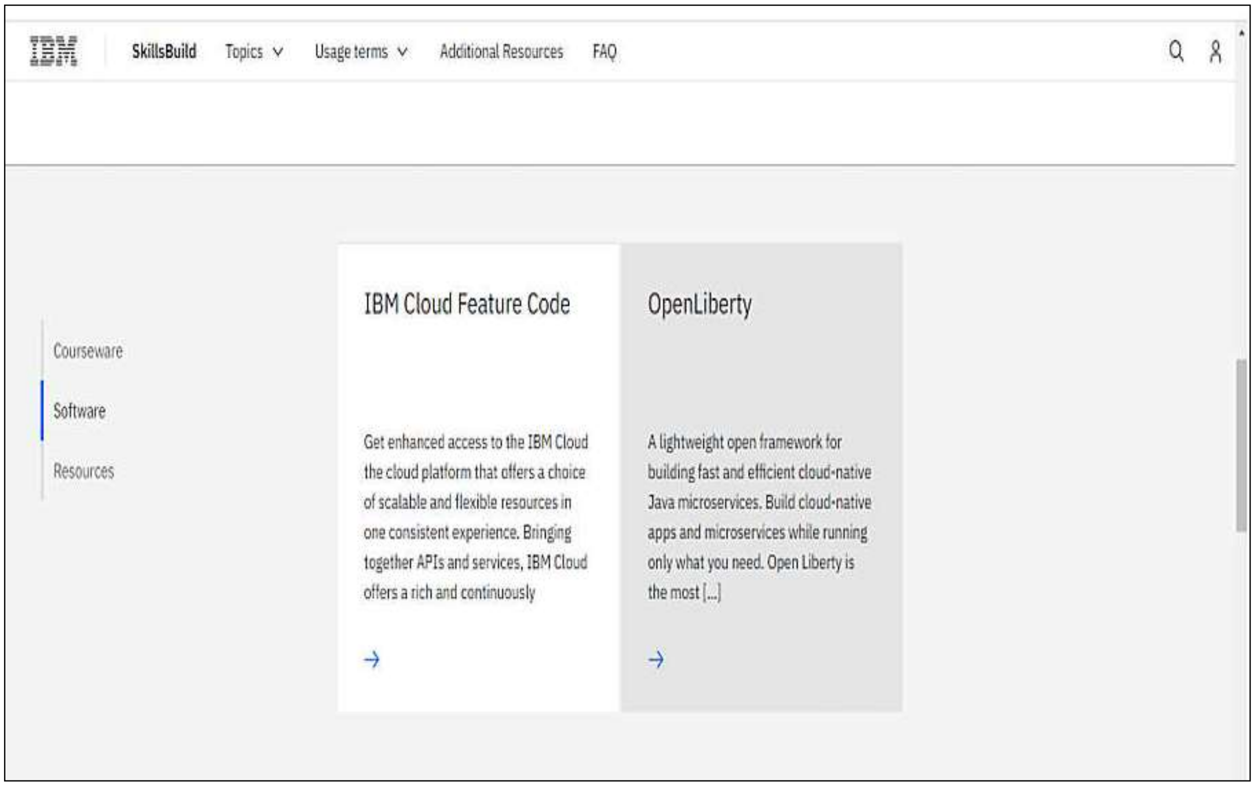
4. IMPLEMENTATION

Creating a database using Cloudant DB by IBM cloud services.

1. Go to IBM Academic Initiative site and sign up using institution email.



2. Generate the IBM cloud feature code for the account.



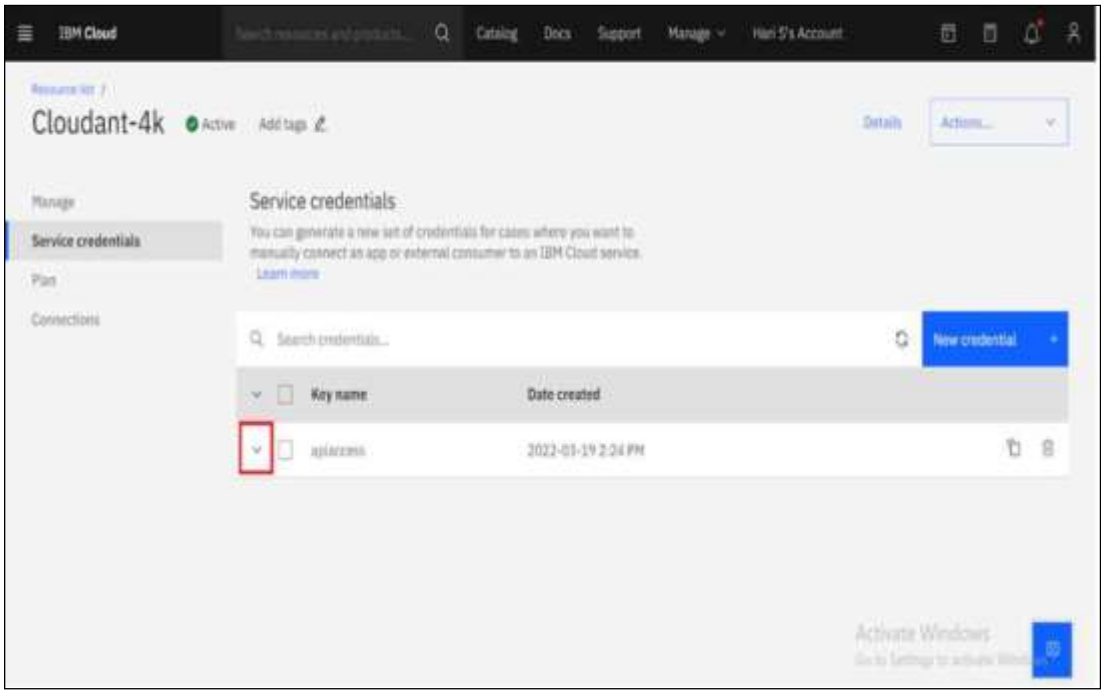
IBM Cloud Feature Code

Get enhanced access to the IBM Cloud the cloud platform that offers a choice of scalable and flexible resources in one consistent experience. Bringing together APIs and services, IBM Cloud offers a rich and continuously

OpenLiberty

A lightweight open framework for building fast and efficient cloud-native Java microservices. Build cloud-native apps and microservices while running only what you need. Open Liberty is the most [...]

3. Search for Cloudant database service and click on create.

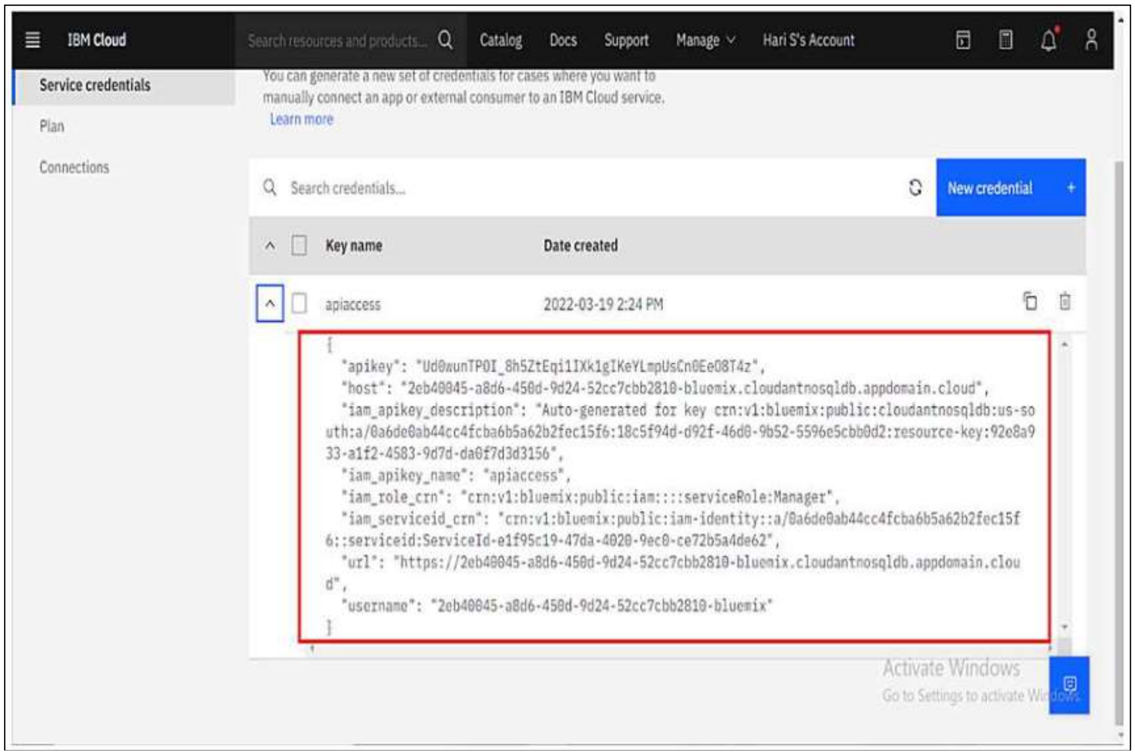


Service credentials

You can generate a new set of credentials for cases where you want to manually connect an app or external consumer to an IBM Cloud service.

Key name	Date created
apiarzms	2022-03-19 2:24 PM

4. Create an API key to access the database from external sources.

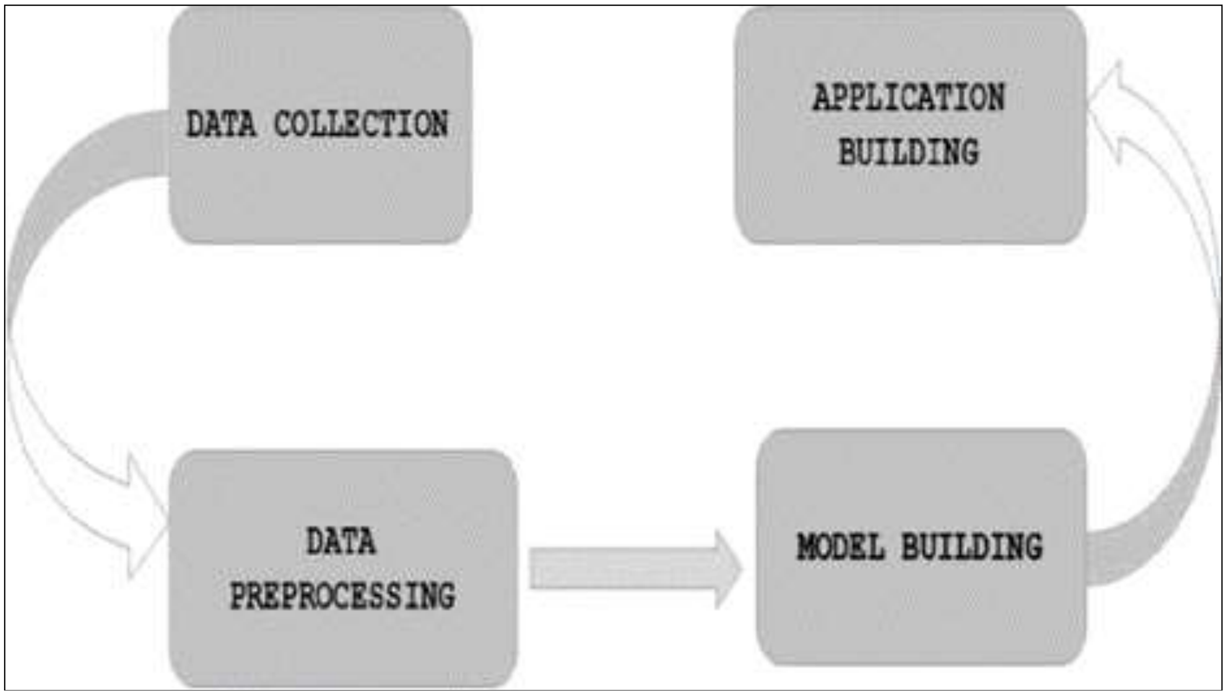


You can generate a new set of credentials for cases where you want to manually connect an app or external consumer to an IBM Cloud service.

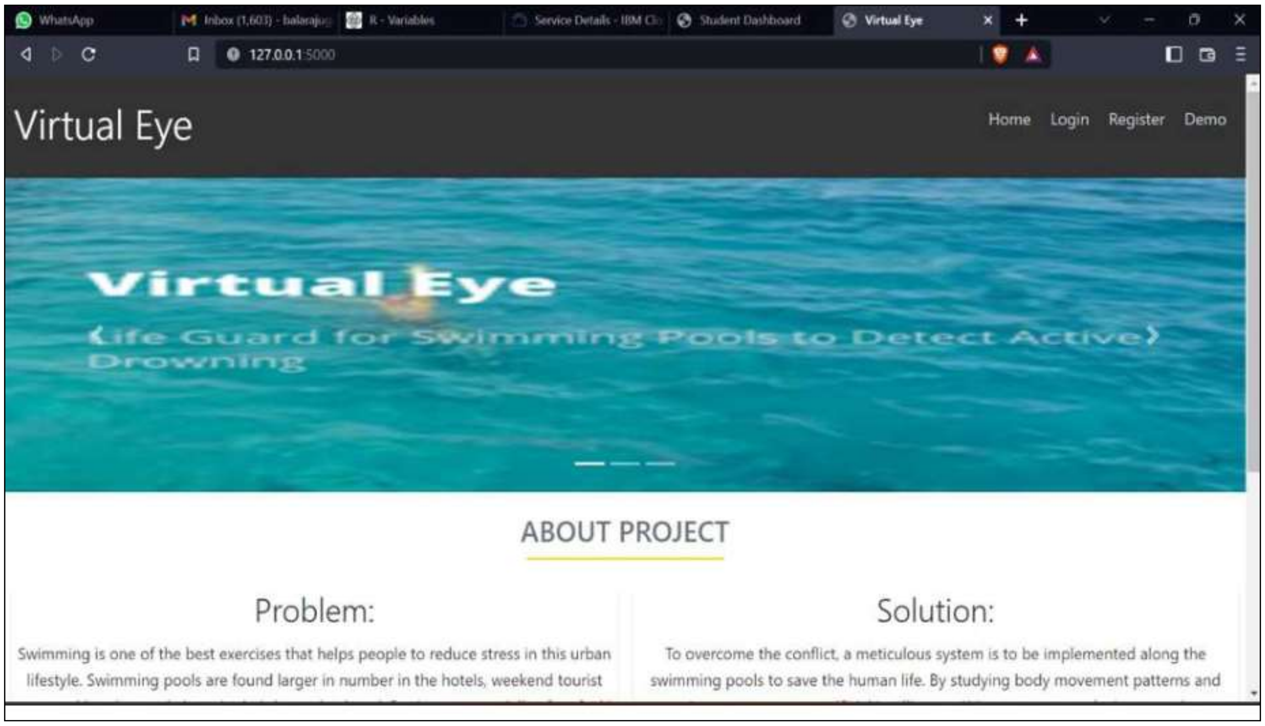
Key name	Date created
apiaccess	2022-03-19 2:24 PM

```
{
  "apikey": "Ud0wunTPOI_8h5ZtEqi1IXk1gIKeYLmpUsCn0Ee08T4z",
  "host": "2eb40045-a8d6-450d-9d24-52cc7cbb2810-bluemix.cloudantnosqldb.appdomain.cloud",
  "iam_apikey_description": "Auto-generated for key crn:v1:bluemix:public:cloudantnosqldb:us-so
uth:a/0a6de0ab44cc4fcba6b5a62b2fec15f6:18c5f94d-d92f-46d0-9b52-5596e5cbb0d2:resource-key:92e8a9
33-a1f2-4583-9d7d-da0f7d3d3156",
  "iam_apikey_name": "apiaccess",
  "iam_role_crn": "crn:v1:bluemix:public:iam::::serviceRole:Manager",
  "iam_serviceid_crn": "crn:v1:bluemix:public:iam-identity::a/0a6de0ab44cc4fcba6b5a62b2fec15f
6::serviceid:ServiceId-e1f95c19-47da-4020-9ec0-ce72b5a4de62",
  "url": "https://2eb40045-a8d6-450d-9d24-52cc7cbb2810-bluemix.cloudantnosqldb.appdomain.clou
d",
  "username": "2eb40045-a8d6-450d-9d24-52cc7cbb2810-bluemix"
}
```

5. FLOWCHART



6. RESULT



Virtual Eye

Home Login Register Demo

Virtual Eye

(Life Guard for Swimming Pools to Detect Active) Drowning

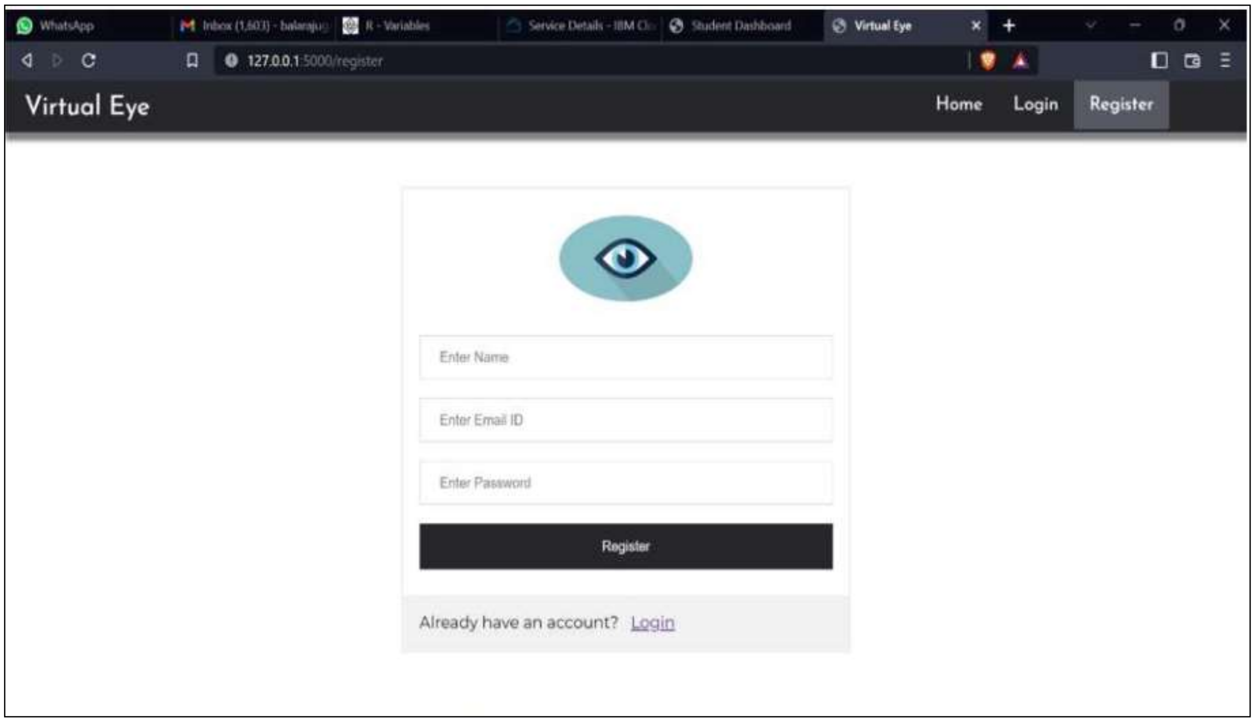
ABOUT PROJECT

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

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



Virtual Eye

Home Login Register

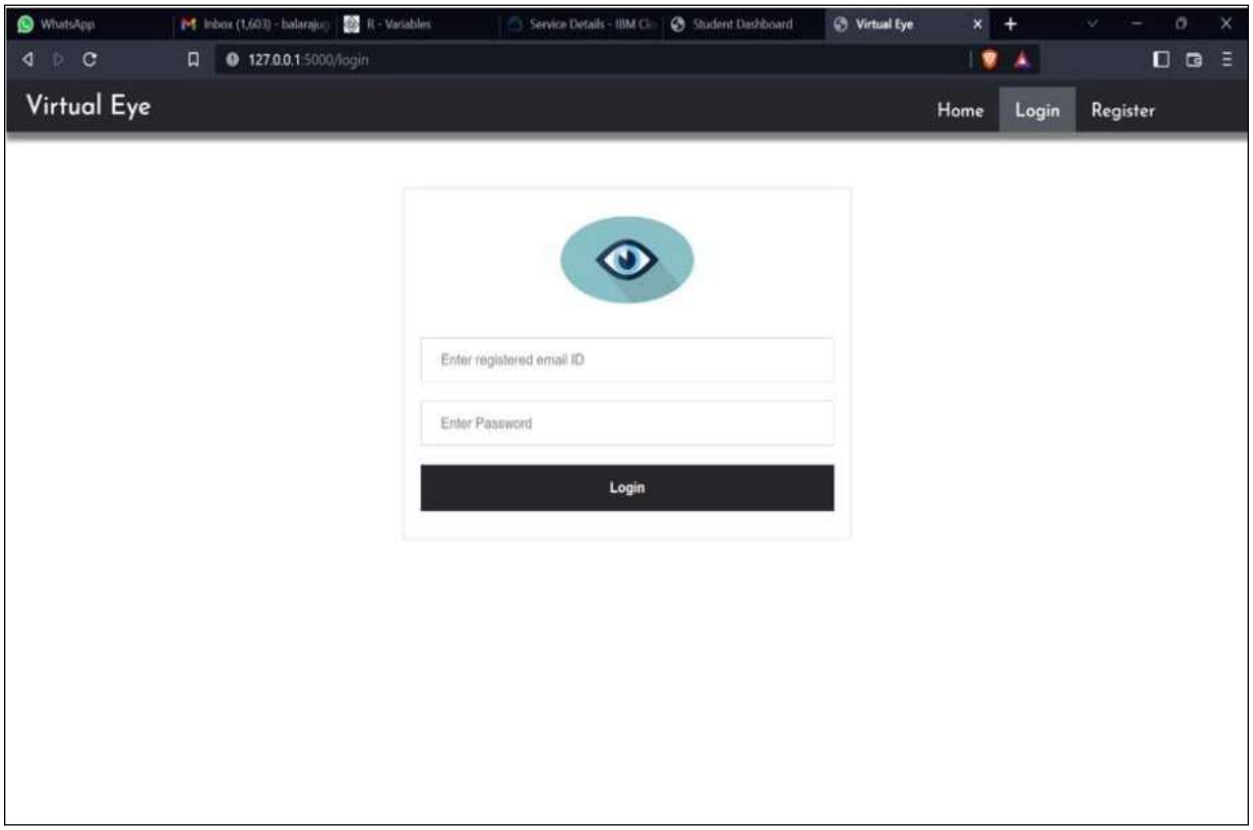
Enter Name

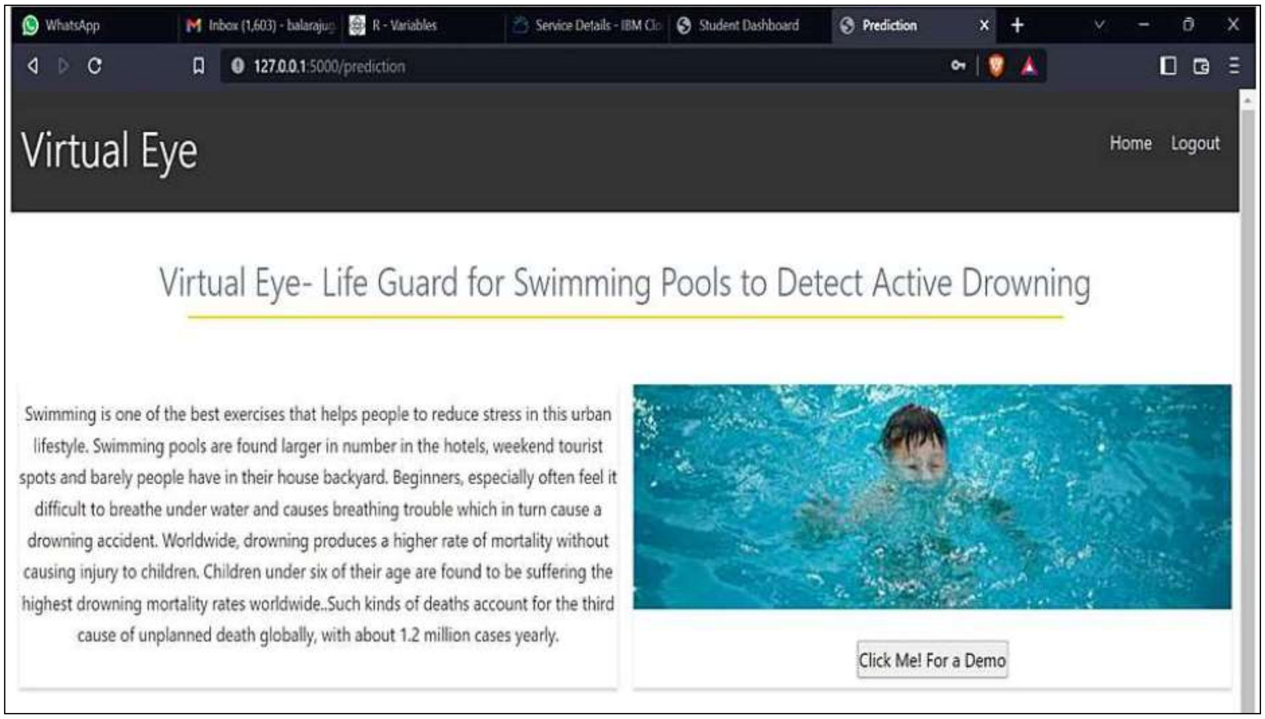
Enter Email ID

Enter Password

Register

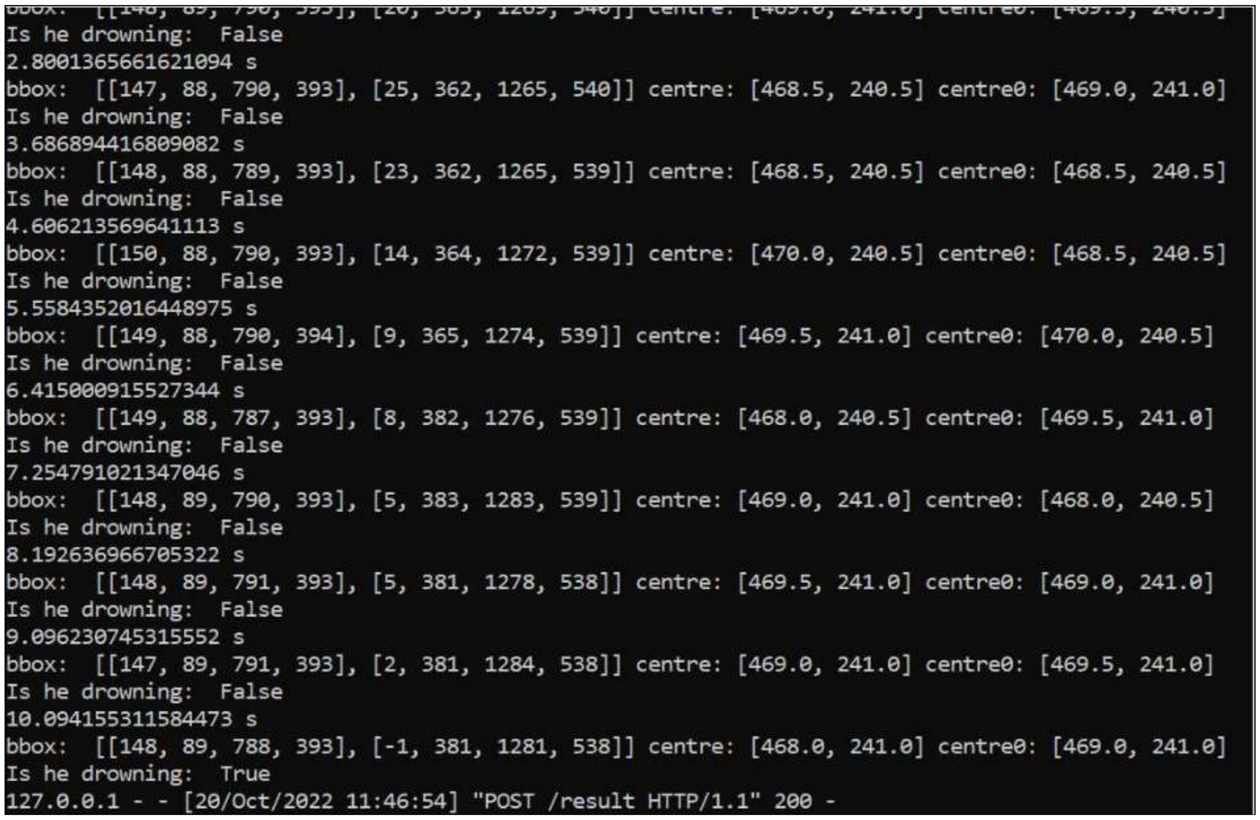
Already have an account? Login





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.



```
bbox:  [[148, 89, 790, 393], [26, 363, 1269, 540]] centre: [469.0, 241.0] centre0: [469.5, 240.5]
Is he drowning:  False
2.8001365661621094 s
bbox:  [[147, 88, 790, 393], [25, 362, 1265, 540]] centre: [468.5, 240.5] centre0: [469.0, 241.0]
Is he drowning:  False
3.686894416809082 s
bbox:  [[148, 88, 789, 393], [23, 362, 1265, 539]] centre: [468.5, 240.5] centre0: [468.5, 240.5]
Is he drowning:  False
4.606213569641113 s
bbox:  [[150, 88, 790, 393], [14, 364, 1272, 539]] centre: [470.0, 240.5] centre0: [468.5, 240.5]
Is he drowning:  False
5.5584352016448975 s
bbox:  [[149, 88, 790, 394], [9, 365, 1274, 539]] centre: [469.5, 241.0] centre0: [470.0, 240.5]
Is he drowning:  False
6.415000915527344 s
bbox:  [[149, 88, 787, 393], [8, 382, 1276, 539]] centre: [468.0, 240.5] centre0: [469.5, 241.0]
Is he drowning:  False
7.254791021347046 s
bbox:  [[148, 89, 790, 393], [5, 383, 1283, 539]] centre: [469.0, 241.0] centre0: [468.0, 240.5]
Is he drowning:  False
8.192636966705322 s
bbox:  [[148, 89, 791, 393], [5, 381, 1278, 538]] centre: [469.5, 241.0] centre0: [469.0, 241.0]
Is he drowning:  False
9.096230745315552 s
bbox:  [[147, 89, 791, 393], [2, 381, 1284, 538]] centre: [469.0, 241.0] centre0: [469.5, 241.0]
Is he drowning:  False
10.094155311584473 s
bbox:  [[148, 89, 788, 393], [-1, 381, 1281, 538]] centre: [468.0, 241.0] centre0: [469.0, 241.0]
Is he drowning:  True
127.0.0.1 - - [20/Oct/2022 11:46:54] "POST /result HTTP/1.1" 200 -
```

7. ADVANTAGES AND DISADVANATGES

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.

8. APPLICATIONS

∉ The system can be deployed in homes, hotels, resorts, and swimming pool centres.

∉ The result is predicted in real-time, thus it can be used in emergency situations.

∉ The use of deep learning ensures that the prediction is as accurate as possible.

9. CONCLUSION

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

10. 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 results faster.

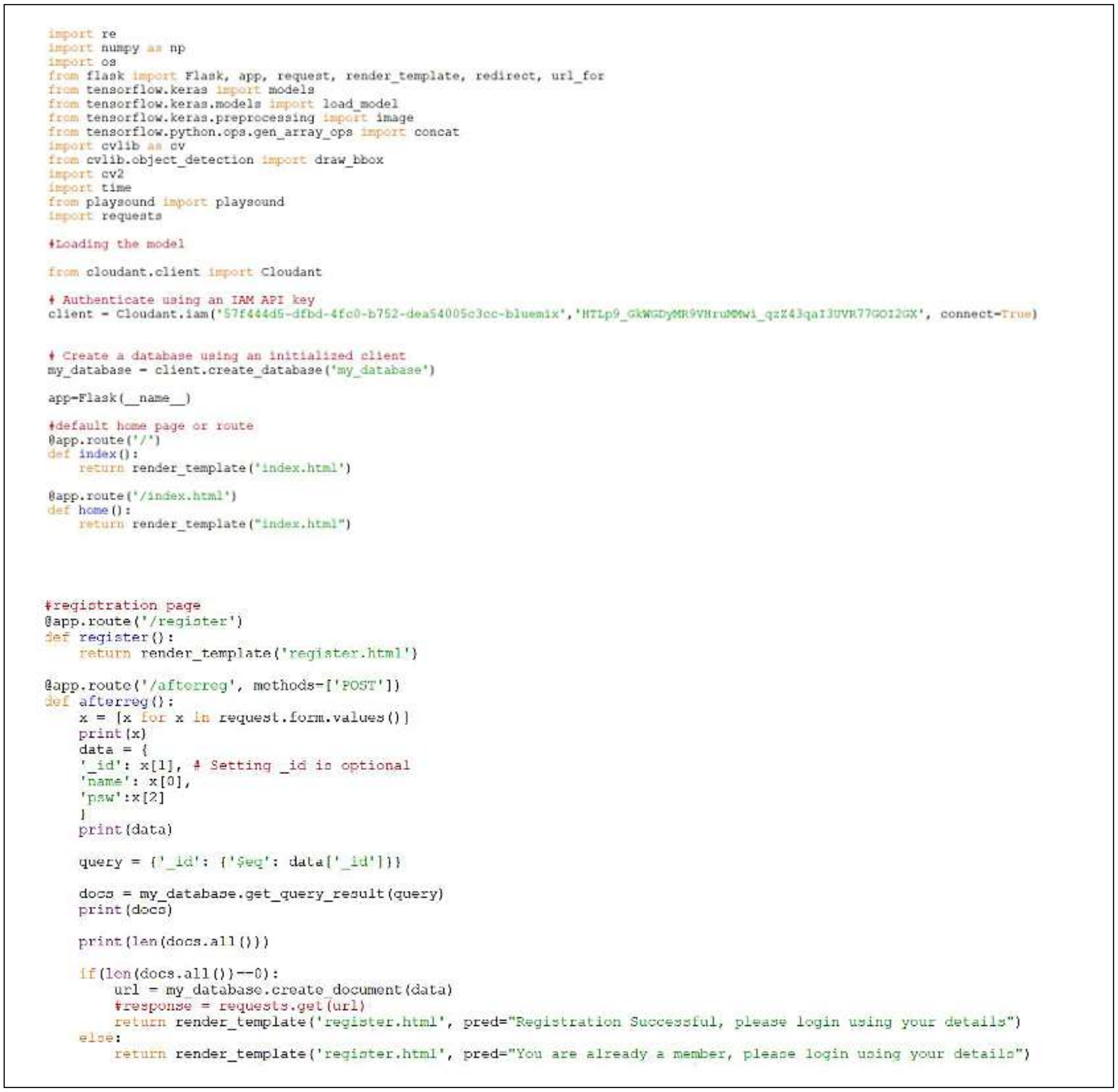
11. BIBLIOGRAPHY

[1] J. Redmond A. Farhadi, ’YOLOv3: An Incremental Improvement’. arXiv, Apr. 08,2018. doi: 10.48550/arXiv.1804.02767.

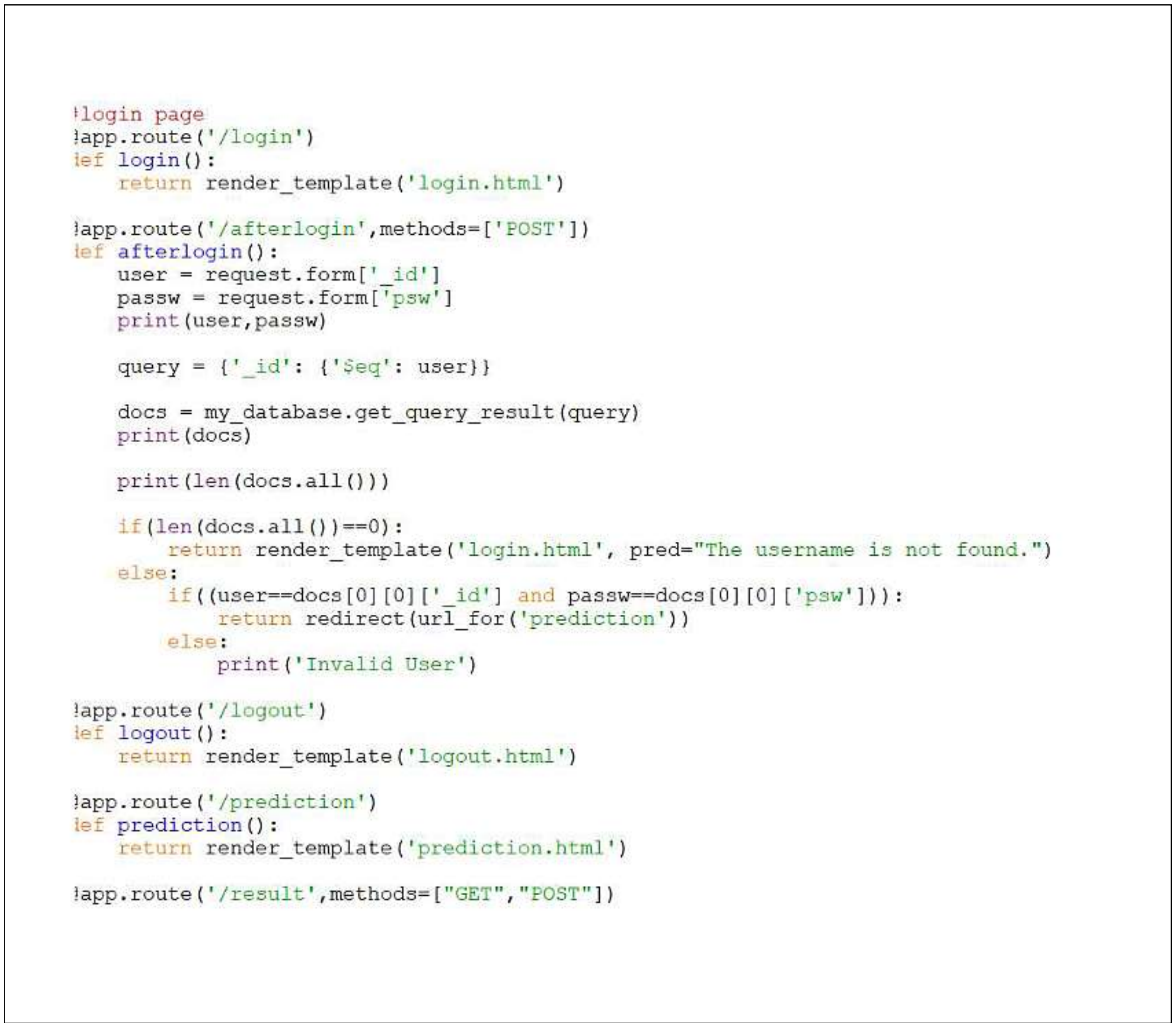
[2] ’YOLO: Real-Time Object Detection’. https://pjreddie.com/darknet/yolo/ (accessed Oct. 15, 2022).

[3] H. Aung,A. V. Bobkov, and N. L. Tun,‘Face Detection in Real Time Live VideoUsing Yolo Algorithm Based on Vgg16 Convolutional Neural Network’, in 2022 International Conference on Industrial Engineering, Applications and Mon«f cturing (ICIEAM), May 2021, pp. 697—702. doi: 10.1109/ICIEAM51226.2021.9446291.

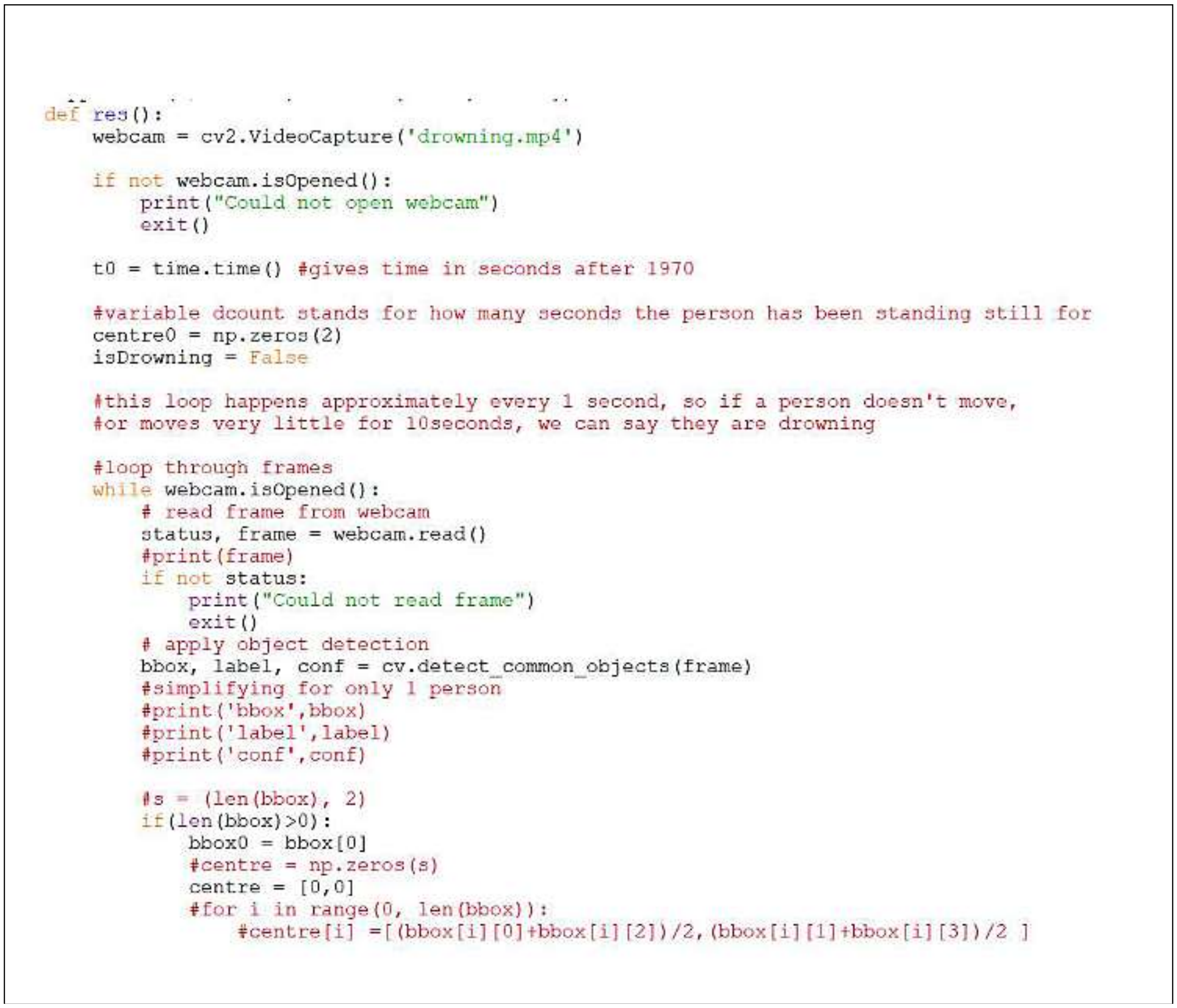
APPENDIX



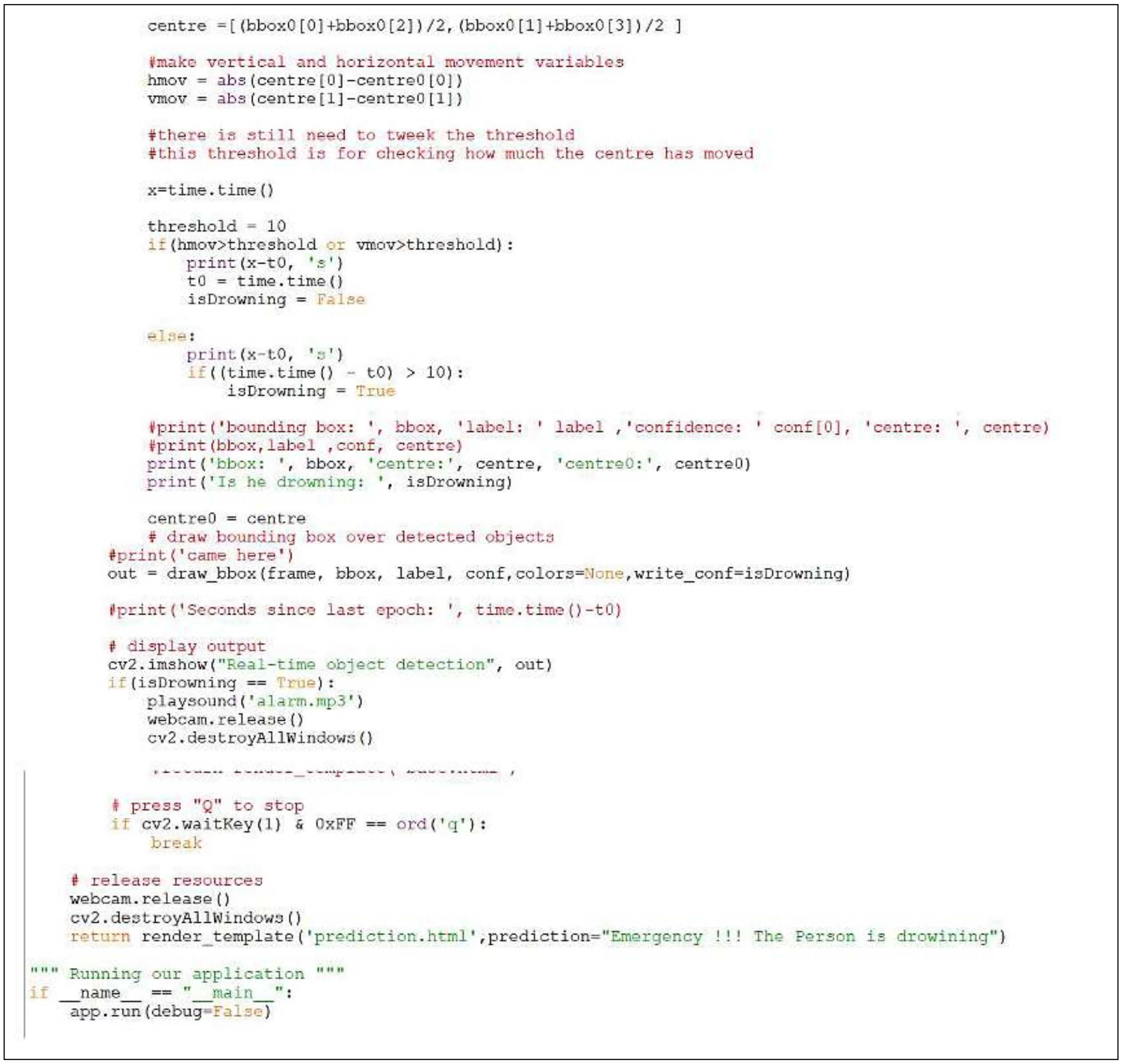
```python
import re
import numpy as np
import os
from flask import Flask, app, request, render_template, redirect, url_for
from tensorflow.keras import models
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
from tensorflow.python.ops.gen_array_ops import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

#Loading the model

from cloudant.client import Cloudant

# Authenticate using an IAM API key
client = Cloudant.iam('57f444d5-dfbd-4fc0-b752-dea54005c3cc-bluemix','HTLp9_GkWGDyMR9VHruMMWi_qzZ43qaI3UVR77GOI2GX', 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')
ef login():
    return render_template('login.html')

app.route('/afterlogin',methods=['POST'])
ef 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')
ef logout():
    return render_template('logout.html')

app.route('/prediction')
ef 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()
        #print(frame)
        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
        #print('bbox',bbox)
        #print('label',label)
        #print('conf',conf)

        #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
        #print('came here')
        out = draw_bbox(frame, bbox, label, conf,colors=None,write_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()

            [illegible]

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

    # release resources
    webcam.release()
    cv2.destroyAllWindows()
    return render_template('prediction.html',prediction="Emergency !!! The Person is drowining")

""" Running our application """
if __name__ == "__main__":
    app.run(debug=False)
```