```
import numpy as np
import os
import cv2
import cvlib as cv
import time
from cv2 import threshold
from cvlib.object_detection import draw_bbox
from pathlib import Path
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('b62e1d3b-33ea-4084-839d-e868a260e907-
bluemix','lj9Fc8oQ9BocOCU7sGGezOyfdM5WjlcotfcRbepx7ys7', 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 base():
    return render_template('Base.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('home'))
        else:
            print('Invalid User')


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

@app.route('/prediction')
def demo():
    return render_template('Demo.html')


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('drowning3.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,isDrowning)
            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("alarm.mp3")
            webcam.release()
            cv2.destroyAllWindows()
            # return "nothing"
            return render_template('Demo.html',prediction="Emergency !!! The Person
is drowning")

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

    webcam.release()
    cv2.destroyAllWindows()
    return render_template('Demo.html',prediction="Checking for drowning")


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