VIRTUAL EYE - LIFE GUARD FOR SWIMMING POOLS TO DETECT ACTIVE DROWNING

A PROJECT REPORT

Submitted by

PRASANNA M (922119104028)
PRASANTH S (922119104029)
PRAVEEN (922119104030)
TAMILSELVAN (922119104045)

In partial fulfillment for the award of the degree

BACHELOROFENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

SSM INSTITUTE OF ENGINEERING AND TECHNOLOGY,

DINDIGUL 625005

ANNA UNIVERSITY::CHENNAI 600025

TABLE OF CONTENTS

CHAPTER-1

Introduction

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.

1.1 Project Overview

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 the cameras underwater and ceiling and analyzing the video feeds to detect any anomalies.

1.2 Purpose

The Main motive of this project is to protect the swimmer from drowning by analyzing the position of a swimmer using Yolo V3. This works when a person does not move or move slowly for 10sec it means that the person is drowning and 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.

CHAPTER-2

LITERATURESURVEY

2.1 Existing Problem

A real time drowning detection method based on HSV color space analysis is presented which uses prior knowledge of the video sequences to set the best values for the color channels. Our method uses a HSV thresholding mechanism along with Contour detection to detect the region of interest in each frame of video sequences. The presented software can detect drowning person in indoor swimming pools and sends an alarm to the lifeguard rescues if the previously detected person is missing for a specific amount of time.

2.2References

- ✓ 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=000000000030360 254
- ✓ 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/?p id=000000000030360257
- ✓ 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).
- ✓ Health and Safety Executive. (2018). HSG179, Health and safety in swimming pools (Fourth edition)

2.3Problem Statement Definition

Swimming pools are found larger in number in hotels, and weekend tourist spots and barely people

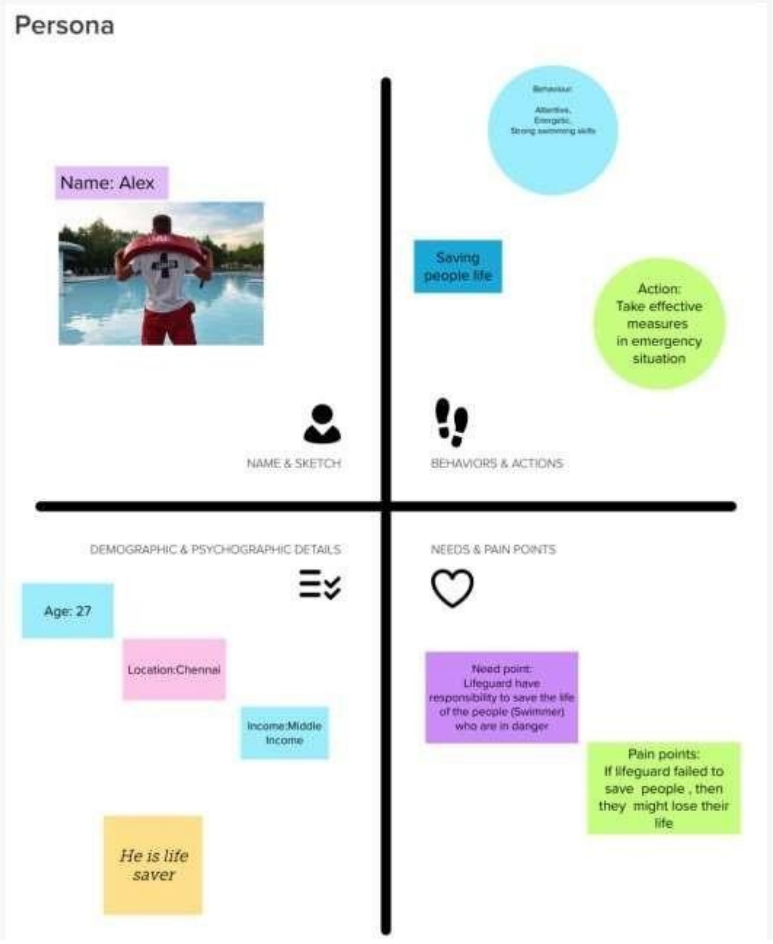
have them in their house backyard. Worldwide, drowning produces a higher rate of mortality without causing injury to children. Children under six of their age are found to be suffering the highest drowning mortality rates world wideso it’s important to save the life of the swimmer from drowning condition .

CHAPTER-3

IDEATION AND PROPOSED SOLUTION

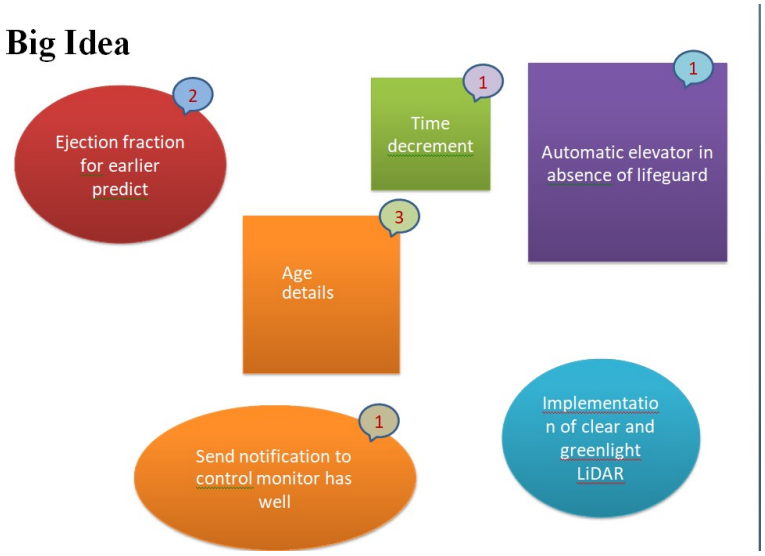
3.1Empathy Map Canvas

An empathy map is a collaborative tool teams can use to gain a deeper insight into their customers. Much like a user persona, an empathy map can represent a group of users, such as a customer segment.



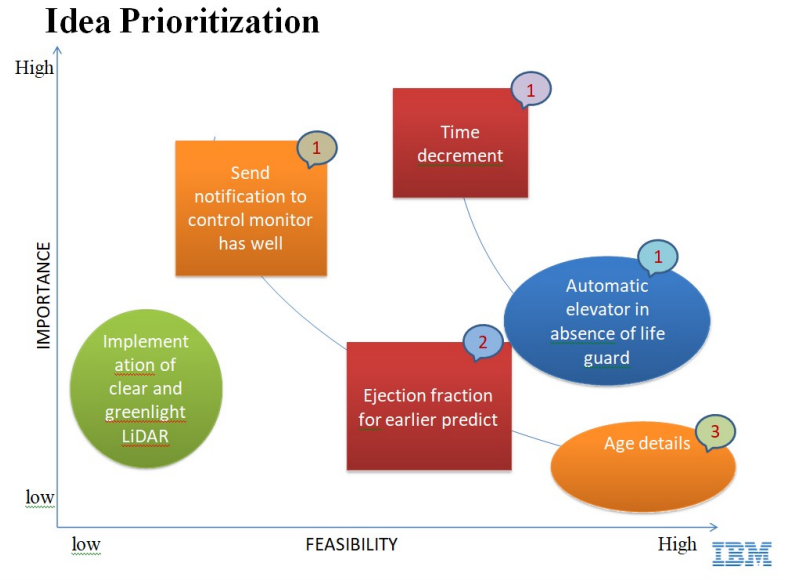
3.2 Big Ideas

It consists of all the ideas of instruments and equipments that we are going to implement in this project.



3.3 Idea Prioritization

It deals with the prioritizing of the big ideas in order of highest to lowest likes.



3.4 Problem Solution Fit

3.5 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	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.
2.	Idea/Solution description	In this project, we use Artificial Intelligence. We install the cameras in underwater to detect the drowning people. Using deep learning, image can be recognized. If the image is detected, it triggers the alarm to alert the Life Guard who rescues the drowning peoples.

3.	Novelty/Uniqueness	The uniqueness of our system software is used to find the swimmer's age along with drowning status so it will be helpful for the swimmer to save the children first incase of 3 or more accident take place at same time. We use YOLO v3 Algorithm to detect the person is drowning or not in accurate time .
4.	Social Impact/Customer Satisfaction	Drowning globally has a higher death rate and is also the third leading cause of unexpected deaths worldwide, especially among child ren under the age of six. To overcome this conflict our drowning detection system will have an Impact on society.
5.	Business Model(Revenue Model)	We can introduce the software-based approach for making a good income. It is extremely useful to lifeguards, swimmers and business operators. The number of features makes it attractive for end users to use our software system.
6.	Scalability of the Solution	Our software system can be used by the company driver whom an ages the pools. We use the IBM cloud server to collect and maintain the data. We will ensure the safety of the swimmers.

CHAPTER-4

REQUIREMENT 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	Deduction	Either horrified or in unconscious
FR-3	Audio	Ask for help or stay quiet if the person is unconscious
FR-4	Support	Take swim tubes or take the help of rescuer
FR-5	PriorAlert	Send alert message to the lifeguard

4.2 Non-Functional Requirements

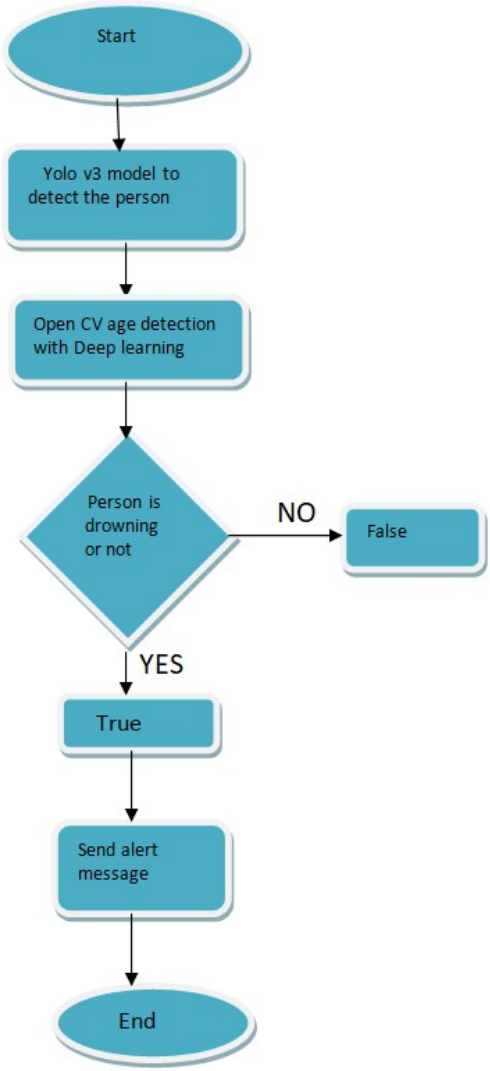
FRNo.	Non-Functional Requirement	Description
NFR-1	**Usability**	To ensure the safety of each and every person present in the pool. A Lifeguard should be present all the time in the pool.

NFR-2	**Security**	Lifeguardsshouldbeawareofthealertmessagetosavethelifeoftheswimmer
NFR-3	**Reliability**	Virtual eye lifeguard triggers an immediate prior alarm if a swimmer is in peril ,helping To avoid panic even in critical situations.
NFR-4	**Performance**	The alarm is triggered when the swimmer is drowning
NFR-5	**Availability**	Equipment and accessories include life saver rings, inflatable vests, a Shepherd'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 safely.
NFR-6	**Scalability**	Virtualeyelifeguarddetectspotentialdrowningandpromptlynotifiesyou.Itfeaturesthelatestartificialintelligence Technology and adapts to the needs of the user.

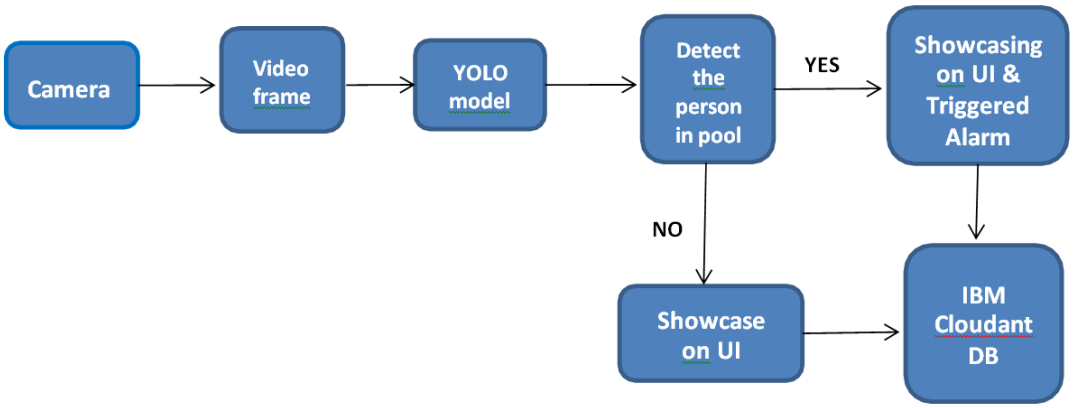
CHAPTER-5

PROJECTDESIGN

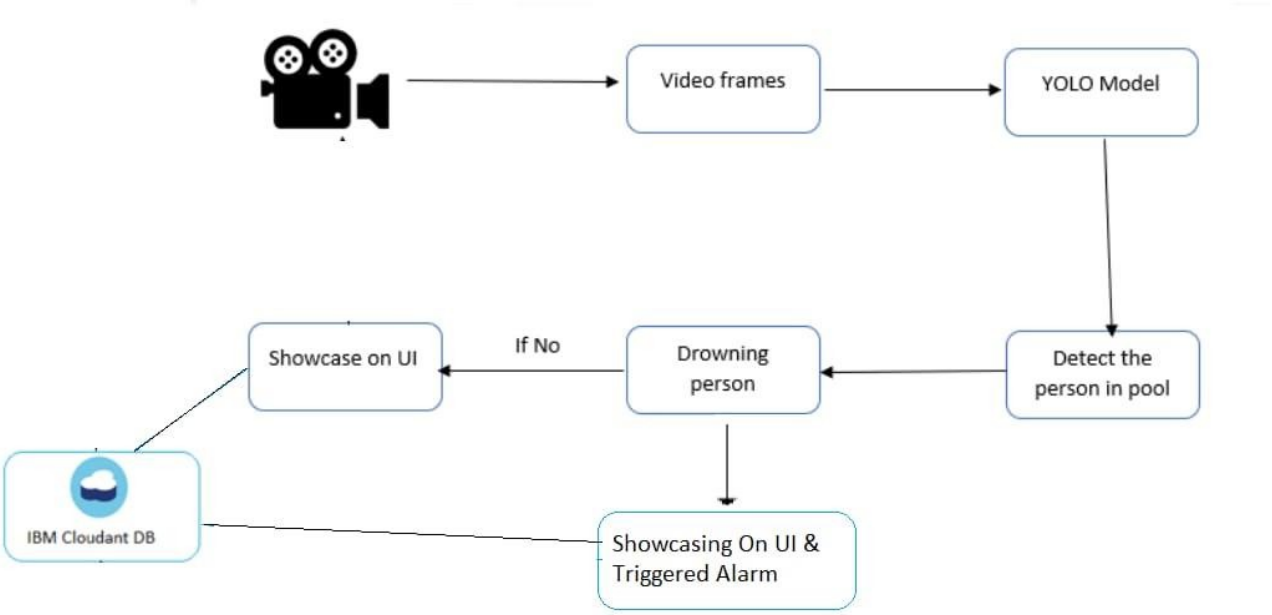
5.1 Data Flow Diagram



5.2 Solution Architecture



Technical Architecture



CHAPTER-6

PROJECT PLANNING PHASE

6.1 Sprint Planning, Schedule &Estimation

Sprint	Functional Requirement(Epic)	User Story Number	User Story/Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a lifeguard, I can register for the application by entering my email, password, and confirming my password.	2	High	PRASANNA M PRAVEEN PRASANTH TAMILSELVAN
Sprint-1	Login	USN-2	As a lifeguard ,I can login to the application by entering email &password	2	High	PRASANNA M PRAVEEN PRASANTH TAMILSELVAN
Sprint-2	Cloudant DB	USN-1	Create DB	2	High	PRASANNA M PRAVEEN PRASANTH TAMILSELVAN
Sprint-3	Coding (Accessing datasets)	USN-1	Coding is a set of instructions used to manipulate information So that a certain nput results in a particular output.	2	High	PRASANNA M PRAVEEN PRASANTH TAMILSELVAN
Sprint-4	Application building	USN-1	As a Lifeguard ,It will show the current Information of the swimming pool	1	Medium	PRASANNA M PRAVEEN

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	20	4Days	25Oct2022	28Oct2022	20	28Oct2022
Sprint-2	20	5Days	28Oct2022	04Nov2022	20	04Nov2022
Sprint-3	20	8Days	02Nov2022	10Nov2022	20	11Nov2022
Sprint-4	20	9Days	10Nov2022	21Nov2022	20	21Nov2022

CHAPTER-7

CODING AND SOLUTION

7.1 Feature 1

- In order to manage a connection from a local system we must first initialize the connection by constructing a Cloudant client. We need to import the cloudant library.
- IBM Cloud Identity & Access Management enables us to securely authenticate users and control access to all cloud resources consistently in the IBM Blue mix Cloud Platform.

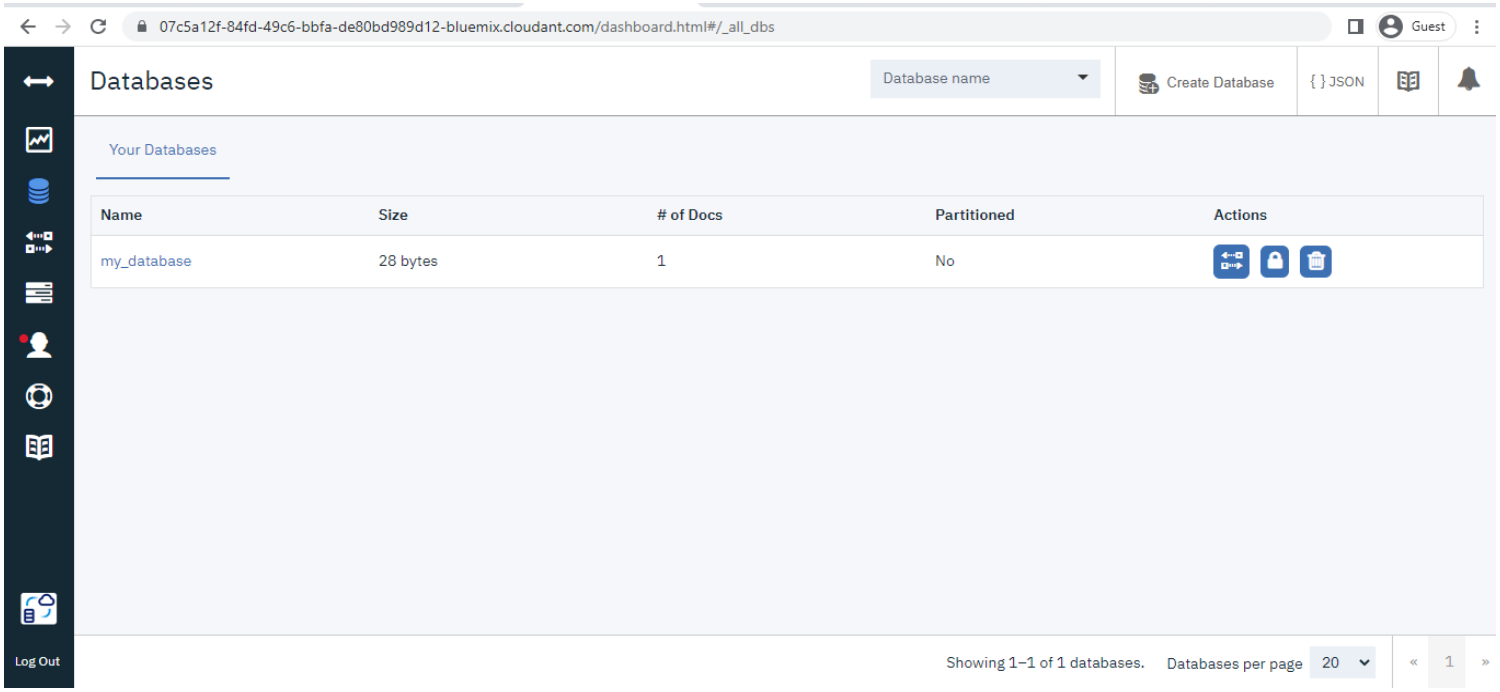
1. Once a connection is established we can create a database, open an existing database.

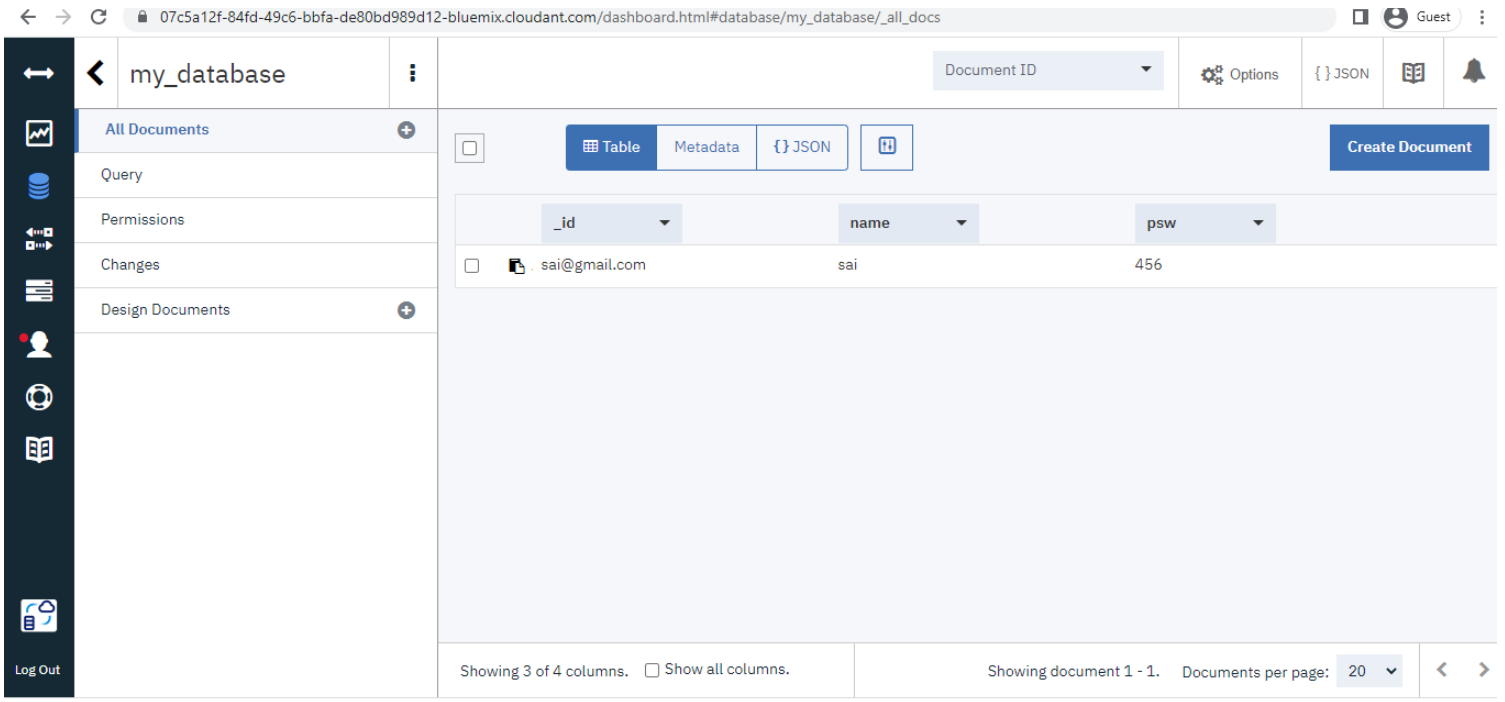
2. Create a database as my_database.

CODE

```
from cloudant.client
import Cloudant
client = Cloudant.iam(
 '07c5a12f-84fd-49c6-bbfa-de80bd989d12-bluemix','Rnz_zCc7hN5Lb5uRHaxn-
WrlN9yqbtz4QKlFVZ4ETZpk',connect=True)
name = 'name'
email = 'a@b.c'
password = '123'
my_database = client.create_database('my_database')
```

Output


07c5a12f-84fd-49c6-bbfa-de80bd989d12-bluemix.cloudant.com/dashboard.html#/_all_dbs
Guest
Databases
Database name
Create Database
{ } JSON
Your Databases
Name
Size
# of Docs
Partitioned
Actions
my_database
28 bytes
1
No
Log Out
Showing 1–1 of 1 databases.
Databases per page
20
1


07c5a12f-84fd-49c6-bbfa-de80bd989d12-bluemix.cloudant.com/dashboard.html#database/my_database/_all_docs
Guest
my_database
Document ID
Options
{ } JSON
All Documents
Query
Permissions
Changes
Design Documents
Table
Metadata
{} JSON
Create Document
_id
name
psw
sai@gmail.com
sai
456
Log Out
Showing 3 of 4 columns.
Show all columns.
Showing document 1 - 1.
Documents per page:
20

7.2 Feature 2

App.py

```
# import necessary packages
import cvlib as cv
from cvlib.object_detection
import draw_bbox
# import necessary packages
from flask import Flask,
        render_template,
        request,redirect,url_for
import requests
import os from sys
 import exit
import cvlib as cv
import cv2
import time
import numpy as np
import math
import argparse
import playsound
from cloudant.client import
        Cloudant
client = Cloudant.iam(
   '07c5a12f-84fd-49c6-bbfa-
        de80bd989d12-
        bluemix','Rnz_zCc7hN5Lb
        5uRHaxn-
        WrlN9yqbtz4QKlFVZ4ET
        Zpk',connect=True)
_id= 'name'
name= 'a@b.c'
psw = '123'
my_database = client.create_database('my_database')

app = Flask(__name__)

@app.route('/')
def index():
   return
        render_template('index.htm
        l')

@app.route('/login')
def login():  # put application's
        code here
   return
```

```
        render_template('login.html
        ')


@app.route('/register')
def register():
    return
        render_template('register.ht
        ml')

@app.route('/home')
def home():
    return render_template('index1.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_res
        ult(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('/step2'))
        else:
            print('Invalid User')

@app.route('/afterreg',
        methods=['POST'])
def afterreg():
    x = [x for x in
        request.form.values()]
    print(x)
    data={
    '_id':x[1],
    'name':x[0],
    'psw':x[2],
    }
    print(data)
```

```
    query = {'_id': {'$eq':
            data['_id']}}

    docs =
            my_database.get_query_res
            ult(query)
    print(docs)

    print(len(docs.all()))
    if(len(docs.all())==0):

            url=my_database.create_do
            cument(data)
        return
            render_template('register.ht
            ml', pred="Registration
            Successful,please login
            using your details")
    else:
        return
            render_template('register.ht
            ml', pred="Registration
            Successful,please login
            using your details")


@app.route('/step2')
def step2():

    print("Begin")

    webcam =
            cv2.VideoCapture("garden.
            mp4")
    padding = 20

    if not webcam.isOpened():
        print("Could not open
            webcam")
        exit()


    t0 = time.time() #gives time in
            seconds after 1970
    #print('t0=',t0)
#variable dcount stands for how
            many seconds the person
```

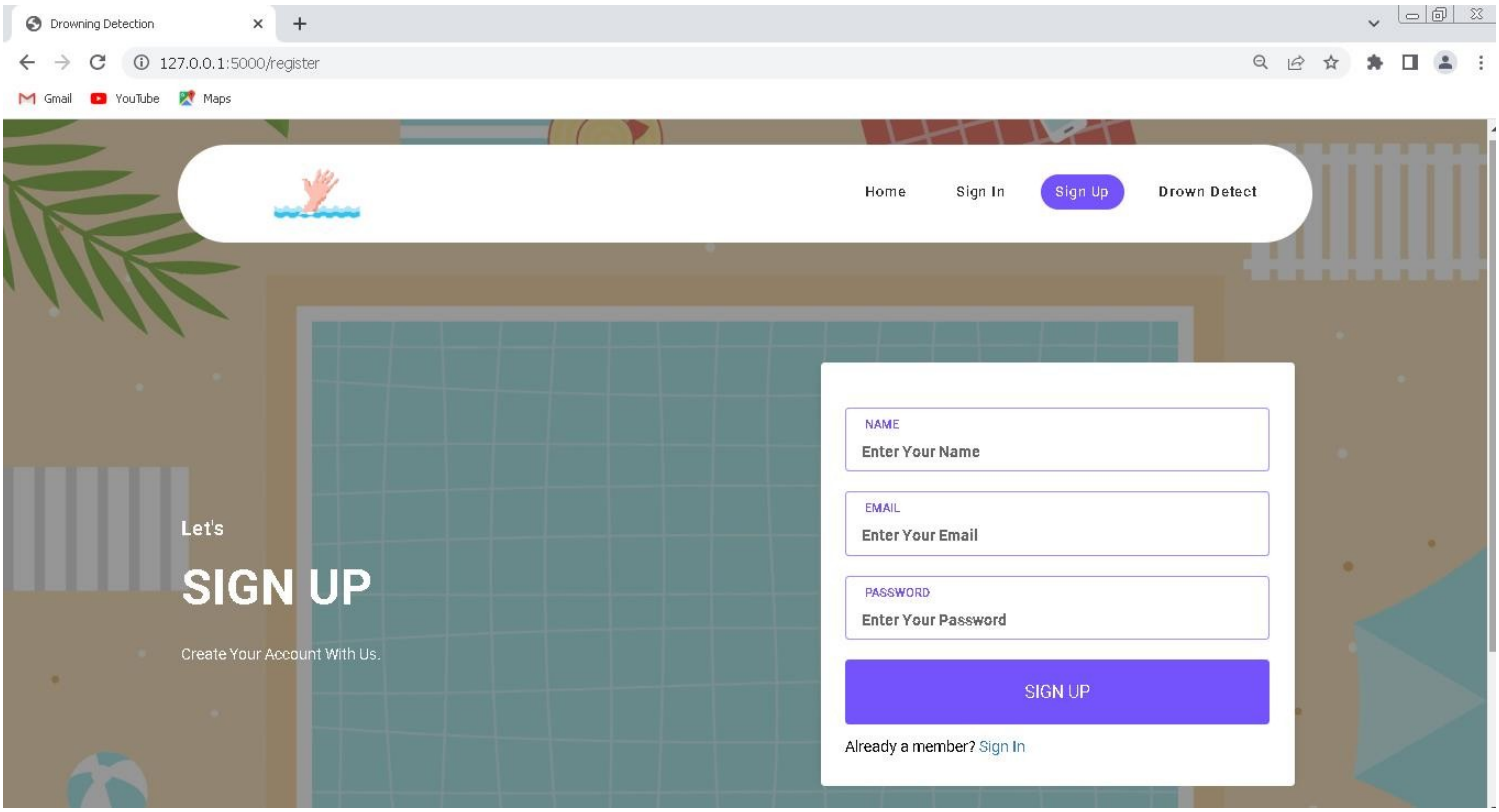
```
        has been standing still for
    centre0 = np.zeros(2)
    isDrowning = False

#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:
            break
    #small_frame =
            cv2.resize(frame,(0,0),fx =
            0.5,fy = 0.5)
    # apply object detection
        bbox, label, conf =
            cv.detect_common_objects(
            frame, confidence=0.25,
            model='yolov3-tiny')

        print(bbox, label, conf)


        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
    #print("hmov=",hmov)
    if(hmov>threshold or
    vmov>threshold):
        print(x-t0, 'sif')
        t0 = time.time()
        isDrowning = False

    else:
        print(x-t0, 'selse')
        if((time.time() - t0) >
    10):
            isDrowning = True

    print('bbox: ', bbox,
    'centre:', centre, 'centre0:',
    centre0)
    print('Is he/she drowning: ',
    isDrowning)
          #print('End of the
    program')


    centre0 = centre
    # draw bounding box over
    detected objects
    # draw bounding box over
    detected objects
out = draw_bbox(frame, bbox,
    label, conf,
    write_conf=True)
    # display output
cv2.imshow("Real-time object
    detection", out)
if(isDrowning == True):
```

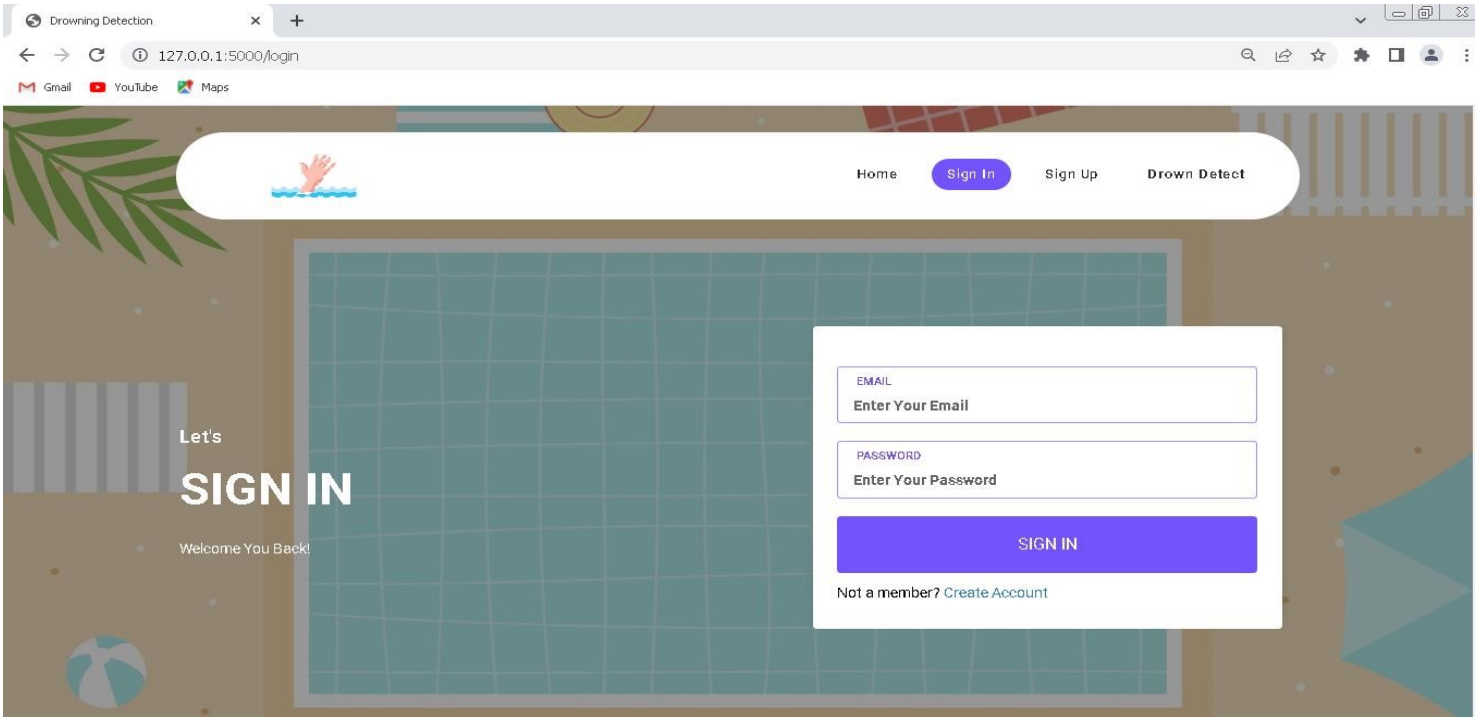
```
            webcam.release()
            cv2.destroyAllWindows()
            return
            render_template('index1.ht
            ml', prediction_text = "1")
    # press "Q" to stop
        if cv2.waitKey(1) & 0xFF ==
            ord('q'):
            break

# release resources
    webcam.release()
    cv2.destroyAllWindows()
if __name__ == '__main__':
     app.run(debug=True)
```

Execution:

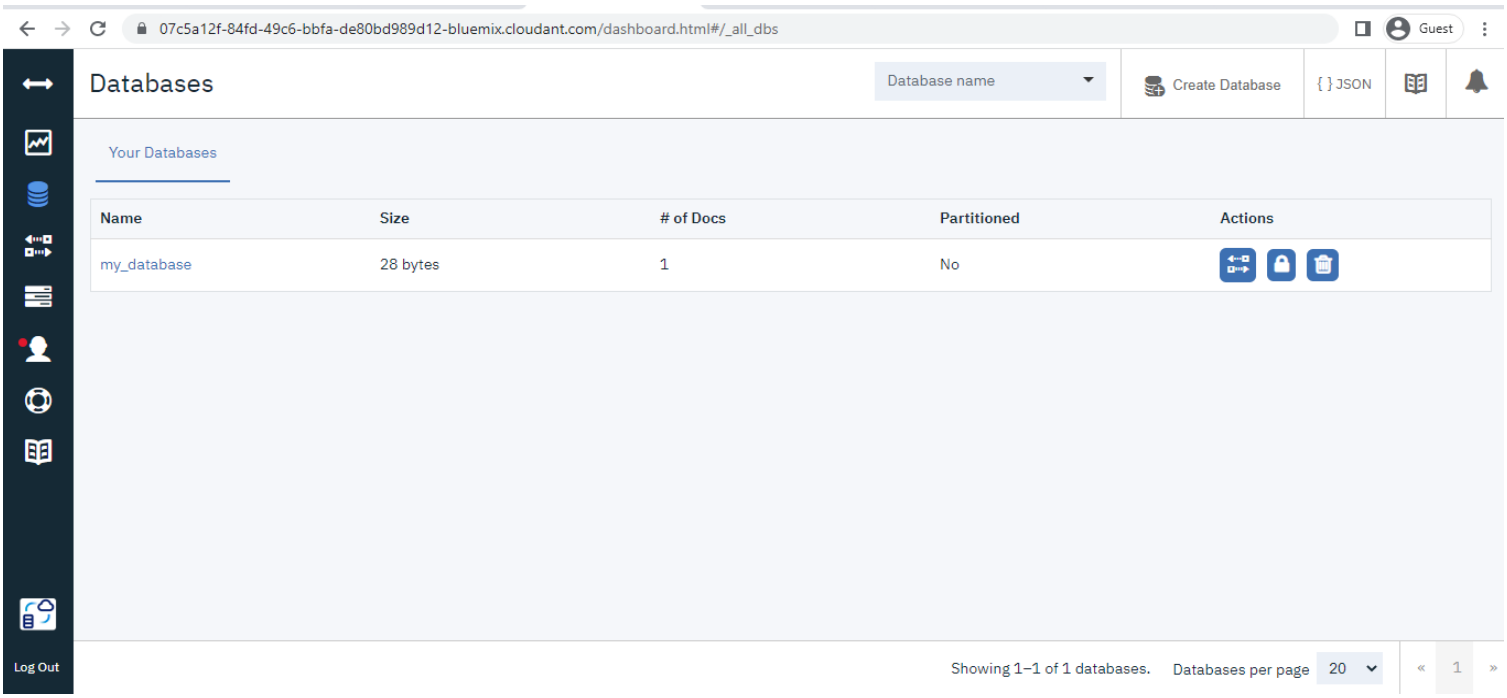
Register page:

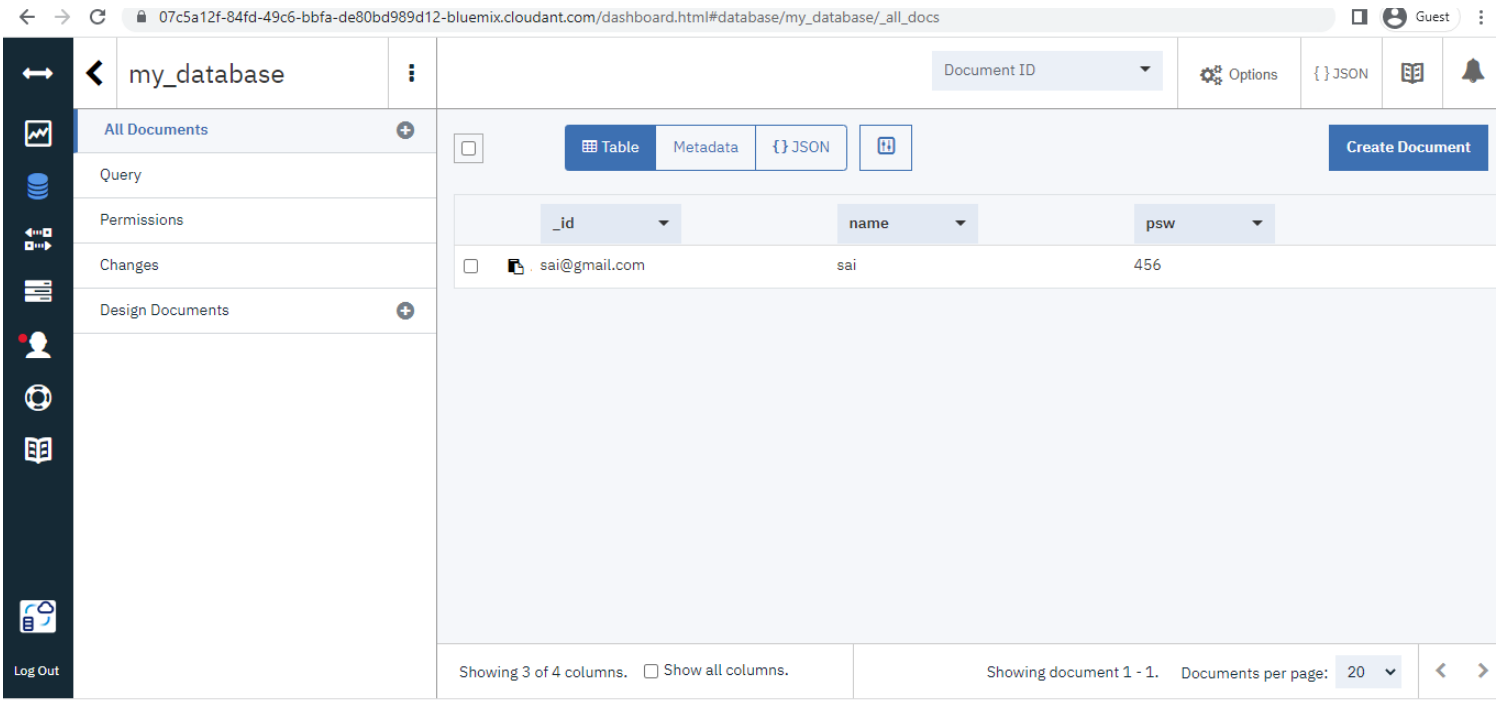


Login Page :

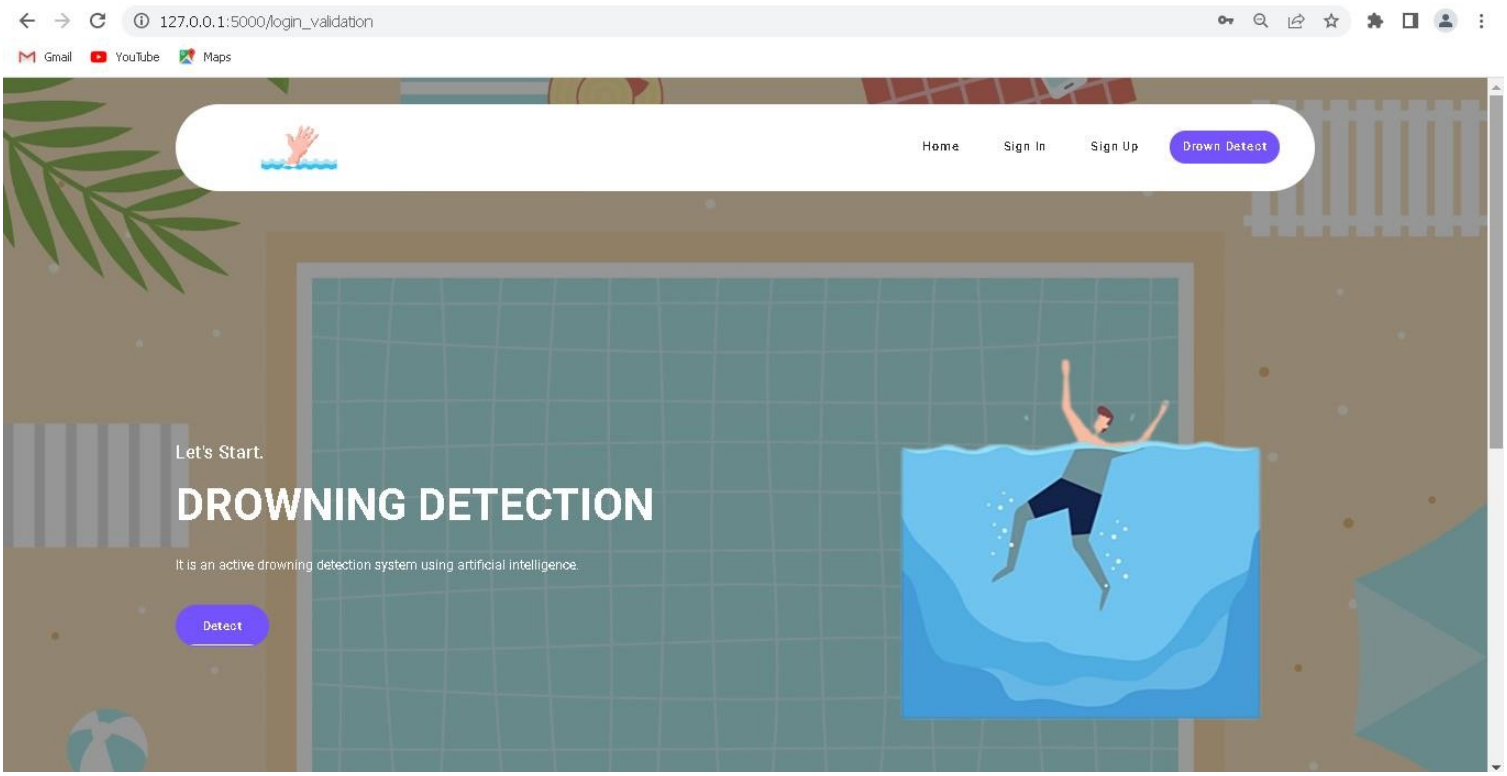


After Register it is stored in Cloud Data Base:

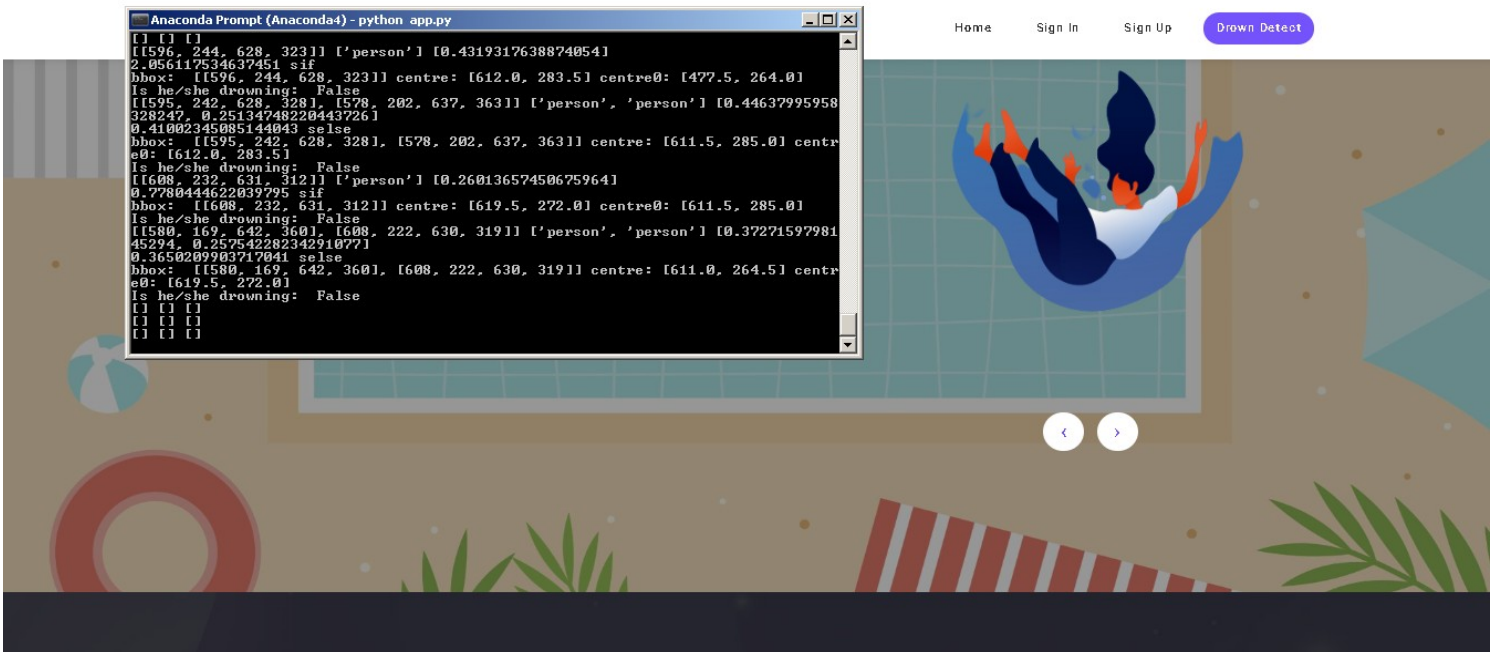



07c5a12f-84fd-49c6-bbfa-de80bd989d12-bluemix.cloudant.com/dashboard.html#database/my_database/_all_docs
Guest
my_database
Document ID
Options
{ } JSON
All Documents
Query
Permissions
Changes
Design Documents
Table
Metadata
{} JSON
Create Document
_id
name
psw
sai@gmail.com
sai
456
Log Out
Showing 3 of 4 columns.
Show all columns.
Showing document 1 - 1.
Documents per page:
20

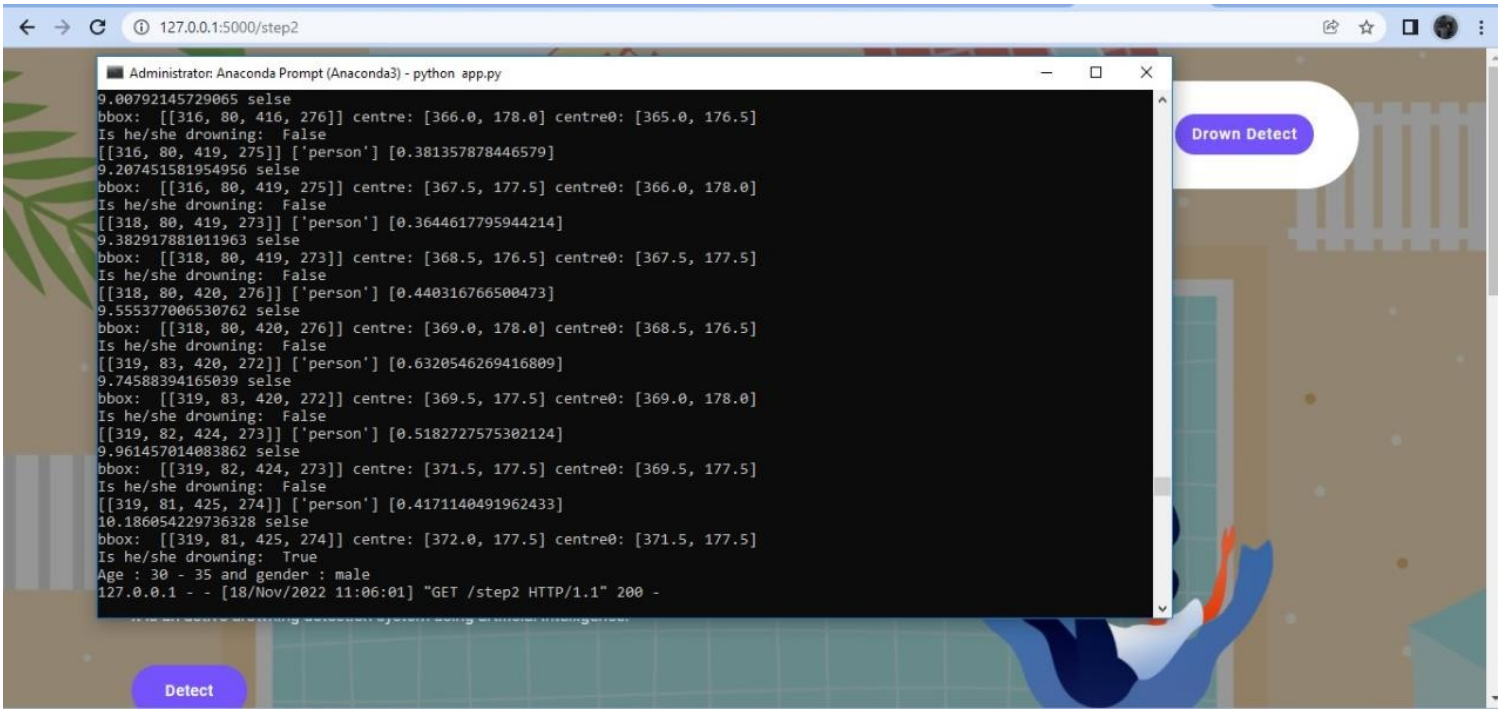
Detection Page:



Before Drowning :



Result :


127.0.0.1:5000/step2
Administrator: Anaconda Prompt (Anaconda3) - python app.py
9.00792145729065 selse
bbox: [[316, 80, 416, 276]] centre: [366.0, 178.0] centre0: [365.0, 176.5]
Is he/she drowning: False
[[316, 80, 419, 275]] ['person'] [0.381357878446579]
9.207451581954956 selse
bbox: [[316, 80, 419, 275]] centre: [367.5, 177.5] centre0: [366.0, 178.0]
Is he/she drowning: False
[[318, 80, 419, 273]] ['person'] [0.3644617795944214]
9.382917881011963 selse
bbox: [[318, 80, 419, 273]] centre: [368.5, 176.5] centre0: [367.5, 177.5]
Is he/she drowning: False
[[318, 80, 420, 276]] ['person'] [0.440316766500473]
9.555377006530762 selse
bbox: [[318, 80, 420, 276]] centre: [369.0, 178.0] centre0: [368.5, 176.5]
Is he/she drowning: False
[[319, 83, 420, 272]] ['person'] [0.6320546269416809]
9.74588394165039 selse
bbox: [[319, 83, 420, 272]] centre: [369.5, 177.5] centre0: [369.0, 178.0]
Is he/she drowning: False
[[319, 82, 424, 273]] ['person'] [0.5182727575302124]
9.961457014083862 selse
bbox: [[319, 82, 424, 273]] centre: [371.5, 177.5] centre0: [369.5, 177.5]
Is he/she drowning: False
[[319, 81, 425, 274]] ['person'] [0.417114049196243 3]
10.186054229736328 selse
bbox: [[319, 81, 425, 274]] centre: [372.0, 177.5] centre0: [371.5, 177.5]
Is he/she drowning: True
Age : 30 - 35 and gender : male
127.0.0.1 - - [18/Nov/2022 11:06:01] "GET /step2 HTTP/1.1" 200 -
Drown Detect
Detect

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	7	3	6	5	21
Duplicate	4	0	3	0	7
External	1	2	0	1	4
Fixed	14	1	3	8	26
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	4	2	0	6
Totals	26	11	18	19	67

Test Case Analysis

Thisreportshowsthenumberoftestcasesthathavepassed,failed,anduntested

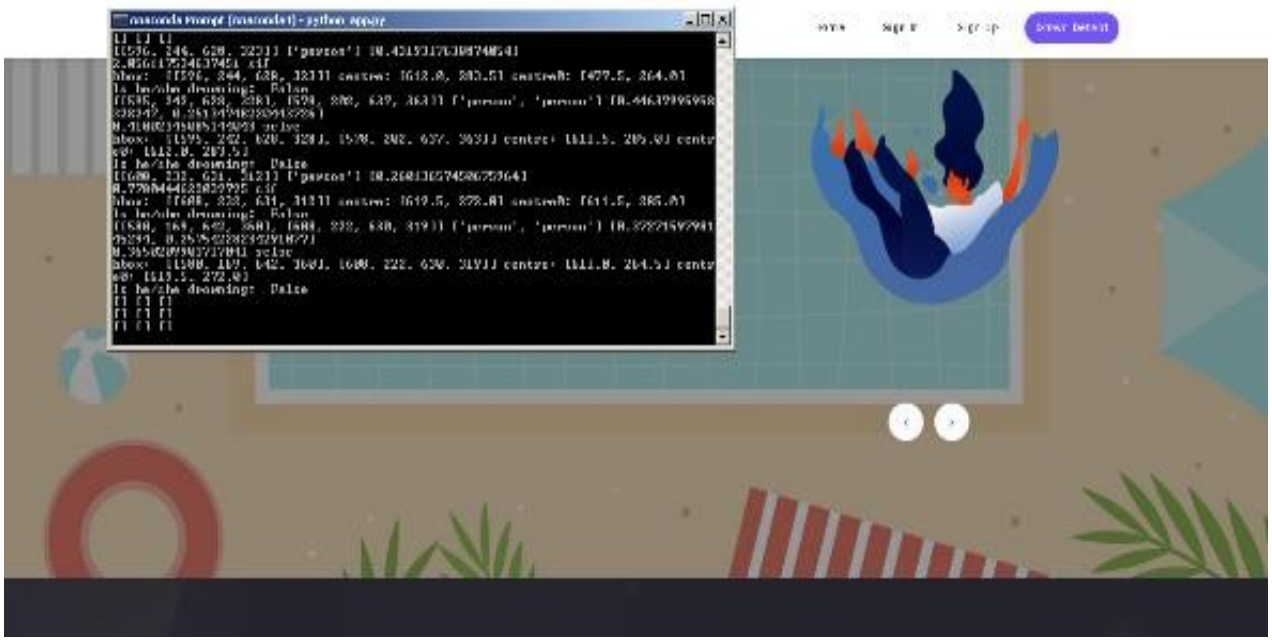
Section	Total Cases	Not Tested	Fail	Pass
Print Engine	5	0	0	5
Client Application	30	0	0	30
Security	2	0	0	2
Outsource Shipping	1	0	0	1
Exception Reporting	7	0	0	7
Final Report Output	9	0	0	9
Version Control	1	0	0	1

CHAPTER-9

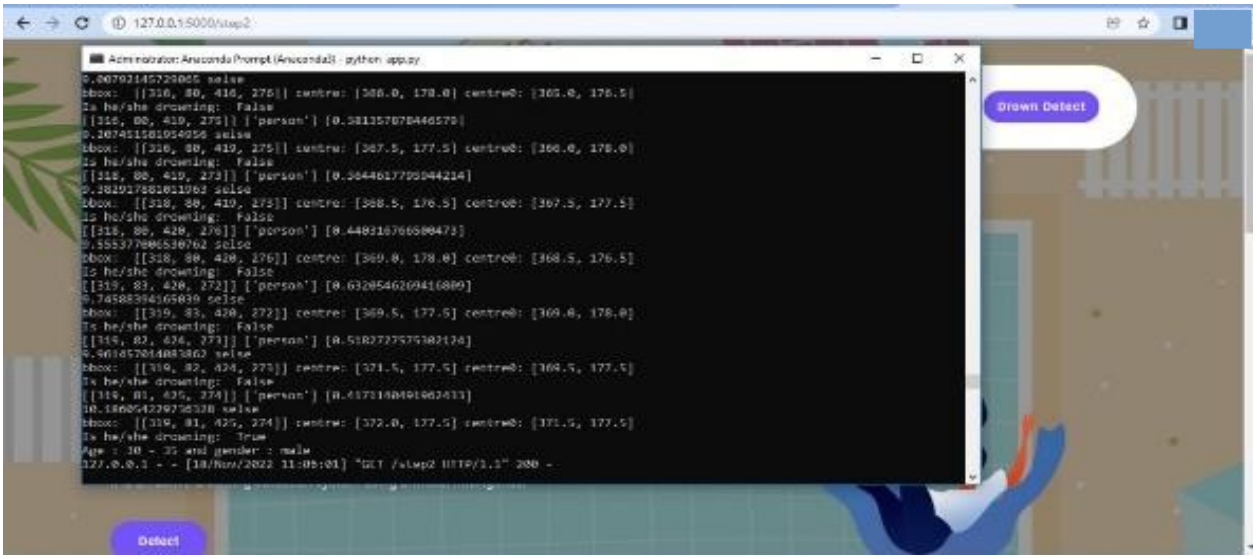
RESULT

9.1PerformanceMetric

Before drowning



After Drowning



CHAPTER-10

ADVANTAGES & DISADVANTAGES

10.1 ADVANTAGE

- User feel comfortable and more secure
- Children, adult, pet animal ,old age people are used
- Spending more time for family, freedom for safety guards near theSwimming pool
- Drowning should be monitored

10.1 DISADVANTAGE:

- For uneducated people will suffer from this technology
- Network connection is always required

CHAPTER-11

CONCLUSION

In this paper, we proposed a method for efficient drowning detection. With the help of the Yolo V3 model we have detected the person and their drowning condition if a person does not move or moved slowly for 10 sec then the alarm is sent to the lifeguard. This system may be further extended for future scope.

CHAPTER-12

FUTURESCOPE

In the current project we have implemented the project that can protect the swimmer from drowning using the Yolo v3 model. So if a person is drowning then a alert message will be sent to the lifeguard and they can save the swimmer. In future we can add or update few more things to this project. We can update the this project by using Pulse Rate Detection . So that the life guard can save the swimmer life before he/she drown .We can use advanced technology to this project so that it will be easier for the lifeguard guard to save the swimmer life at earlier stage .