

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

A PROJECT REPORT

Submitted by

**ASWINI RK (923819104007)
JANANISREE K (923819104015)
KARPAGAM R (923819104018)
SWASTHIKKA R (923819104048)**

In partial fulfillment for the award of the degree

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

**MANGAYARKARASI COLLEGE OF ENGINEERING,
MADURAI 625402**



ANNA UNIVERSITY::CHENNAI 600025

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGENO
1	INTRODUCTION 1.1 Project Overview 1.2 Purpose	1
2	LITERATURESURVEY 2.1 Existing System 2.2 References 2.3 Problem Statement Definition	2
3	IDEATION & PROPOSEDSOLUTION 3.1 Empathy Map Canvas 3.2 Ideation & Brainstorming 3.3 Proposed Solution 3.4 Problem Solution Fit	3
4	REQUIREMENTANALYSIS 4.1 Functional Requirement 4.2 Non-Functional Requirement	7
5	PROJECTDESIGN 5.1 Data-Flow Diagrams 5.2 Solution & Technical Architecture 5.3 User Stories	9
6	PROJECT PLANNING&SCHEDULING 6.1 Sprint Planning ,Schedule &Estimation 6.2 Sprint Delivery Schedule 6.3 Reports From JIRA	12
7	CODING&SOLUTIONING 7.1Feature 1 7.2 Feature 2	15
8	TESTING 8.1 Test cases 8.2 User Acceptance Testing	29

9	RESULTS 9.1 Performance Metrics	31
10	ADVANTAGES & DISADVANTAGES 10.1 Advantages 10.2 Disadvantages	32
11	CONCLUSION	33
12	FUTURE SCOPE	34
13	APPENDIX	35

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

LITERATURE SURVEY

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.

2.2 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>
- ✓ 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>
- ✓ 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)

2.3 Problem Statement Definition

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

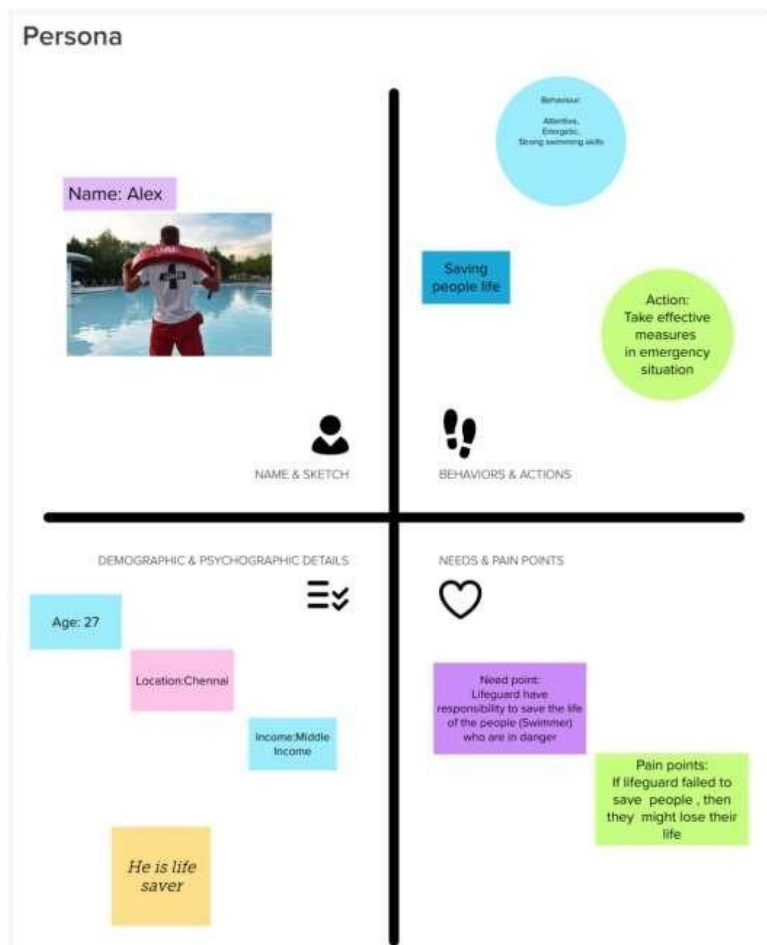
have them in their house backyard. Worldwide, drowning produces a higher rate of mortality without causing injury to children. Children under six of their age are found to be suffering the highest drowning mortality rates world wide so it's important to save the life of the swimmer from drowning condition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

CHAPTER-3

IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map Canvas

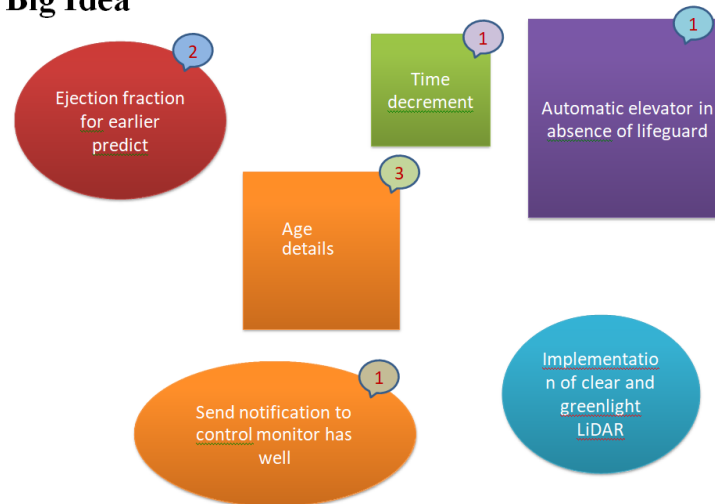
An empathy map is a collaborative tool teams can use to gain a deeper insight into their customers. Much like a user persona, an empathy map can represent a group of users, such as a customer segment.



3.2 Big Ideas

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

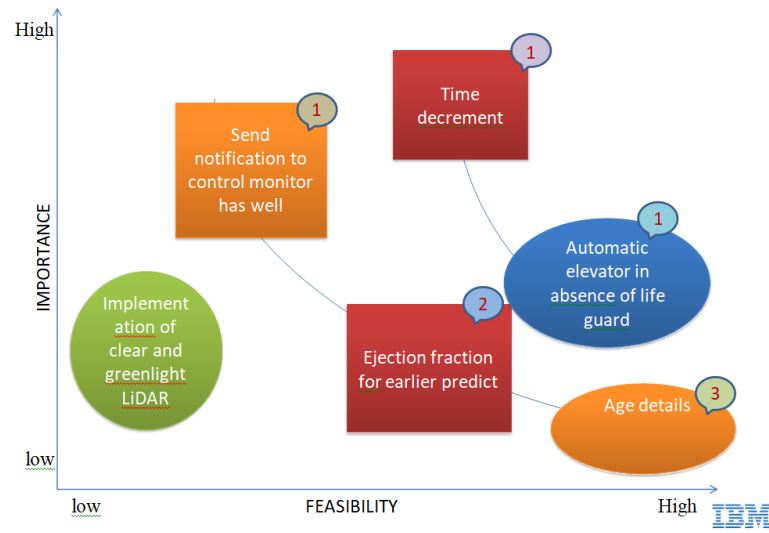
Big Idea



3.3 Idea Prioritization

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

Idea Prioritization



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

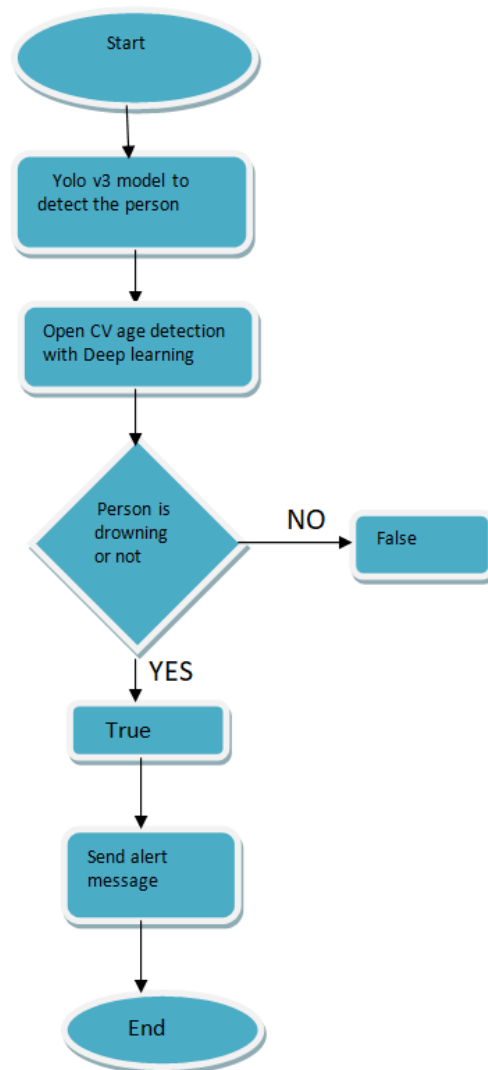
FRNo.	Non-Functional Requirement	Description
NFR-1	Usability	To ensure the safety of each and every person present in the pool. A Lifeguard should be present all the time in the pool.

NFR-2	Security	Lifeguardsshouldbeawareofthealertme ssagetosavethelifeoftheswimmer
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	Virtualeyelifeguarddetectspotentialdrownin gandpromptlynotifiesyou.Itfeaturesthelatestart ificialintelligence Technology and adapts to the needs of the user.

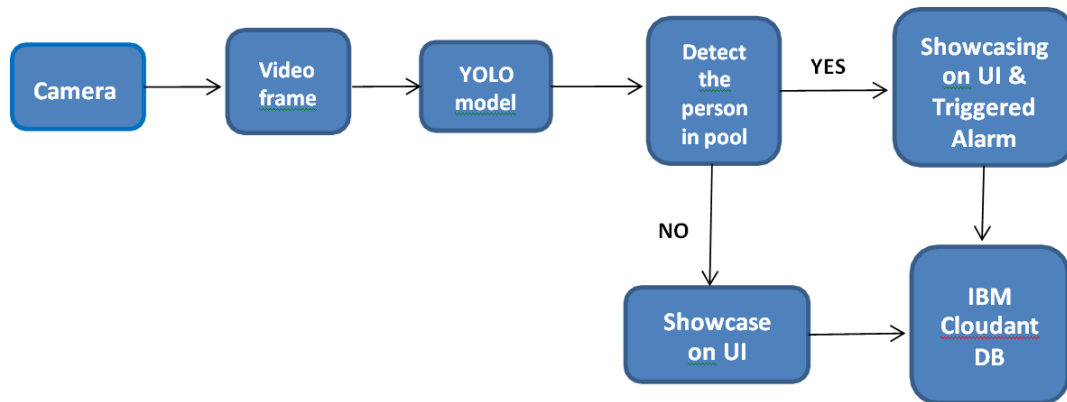
CHAPTER-5

PROJECTDESIGN

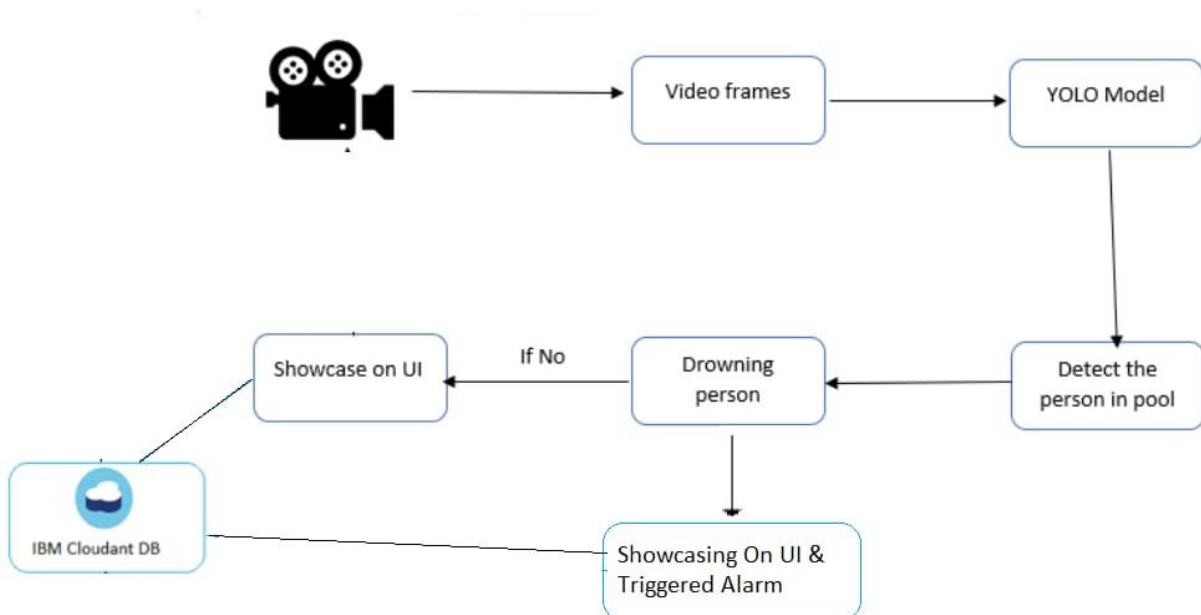
5.1 Data Flow Diagram



5.2 Solution Architecture



Technical Architecture



5.3 User stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Pool owner)	Installation	USN-1	As a pool owner, I can install the cameras and set up the drowning detection system	I can connect the cameras to the cloud-hosted software	High	Sprint-1
	Detecting the drowning persons	USN-2	As a user, I can find the drowning persons by using the drowning detection system	I would receive an alert if a person is drowning	High	Sprint-1
	Notify the lifeguard	USN-3	As a user, I can notify the lifeguard when the system detects a drowning person	I can set up an alarm that would notify the lifeguard	High	Sprint-2
Customer (Lifeguard)	Rescue people	USN-4	As a user, I can rescue the drowning persons from the pool	I can save the drowning person	High	Sprint-2
Customer (Swimmers)	Safety	USN-5	As a user, I can swim without the fear of drowning	I can swim safely with the help of the system and the lifeguard	Medium	Sprint-2
Customer Care Executive	Contact	USN-6	resolve technical issues	I can contact the customer care executive to resolve any issues	Medium	Sprint-3
Administrator	Dashboard	USN-7	Management of the drowning detection system and database management.	I can access the system's logs and any other data instantly	High	Sprint-4

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	Aswini RK Swasthikka R Jananisree K Karpagam R
Sprint-1	Login	USN-2	As a lifeguard ,I can login to the application by entering email &password	2	High	Aswini RK Swasthikka R Jananisree K Karpagam R
Sprint-2	Cloudant DB	USN-1	Create DB	2	High	Aswini RK Swasthikka R Jananisree K Karpagam R
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	Aswini RK SwasthikkaR Jananisree K Karpagam R
Sprint-4	Application building	USN-1	As a Lifeguard ,It will show the current Information of the swimming pool	1	Medium	Aswini RK SwasthikkaR

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	24Oct2022	27Oct2022	20	29Oct2022
Sprint-2	20	5Days	28Oct2022	01Nov2022	20	04Nov2022
Sprint-3	20	8Days	02Nov2022	09Nov2022	20	11Nov2022
Sprint-4	20	9Days	10Nov2022	18Nov2022	20	19Nov2022

6.3 Reports from JIRA

	OCT	NOV	DEC	
Sprints				
> ⚡ VELGFSPTDA-5 Registration				
> ⚡ VELGFSPTDA-6 Create Database				
> ⚡ VELGFSPTDA-7 Coding				
> ⚡ VELGFSPTDA-8 Application building				

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	<div></div> <div></div> <div></div>

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

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-
    WrlN9yqbtz4QKIFVZ4ET
    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.html', 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)

```

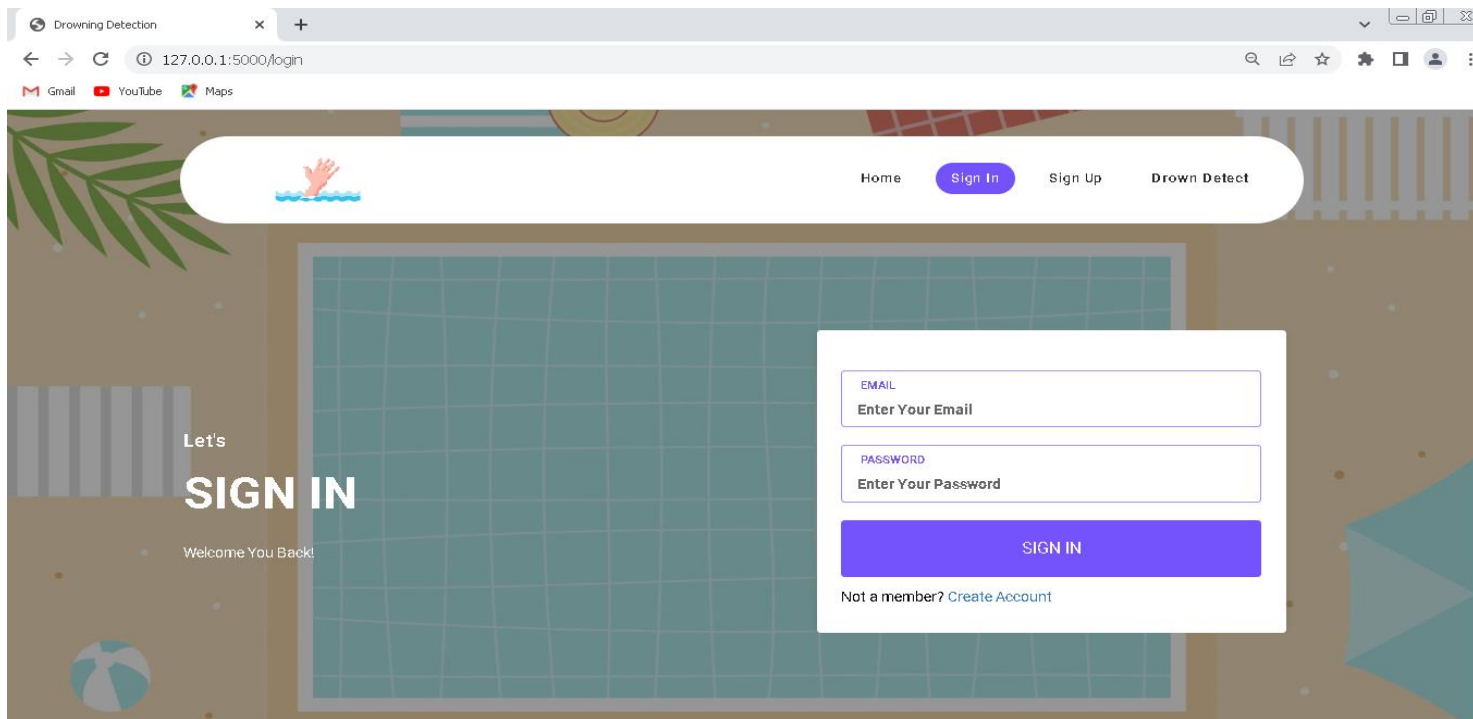
Execution:

Register page:

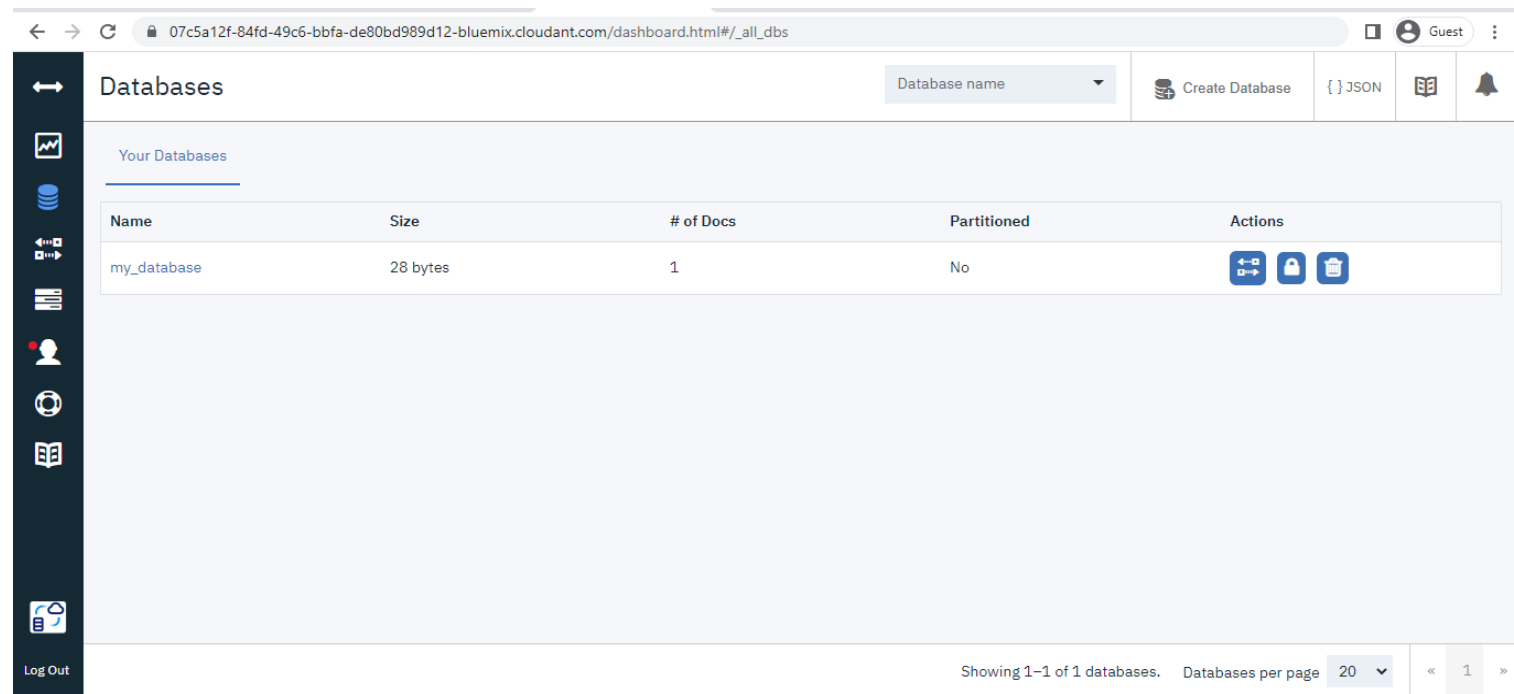
The screenshot shows a web browser window with the title 'Drowning Detection'. The address bar displays '127.0.0.1:5000/register'. The page has a navigation bar with links: Home, Sign In, Sign Up, and Drown Detect. A large 'SIGN UP' button is visible. A registration form is shown on the right side of the page, containing the following fields and buttons:

- NAME**: Enter Your Name
- EMAIL**: Enter Your Email
- PASSWORD**: Enter Your Password
- SIGN UP** button
- Already a member? [Sign In](#)

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

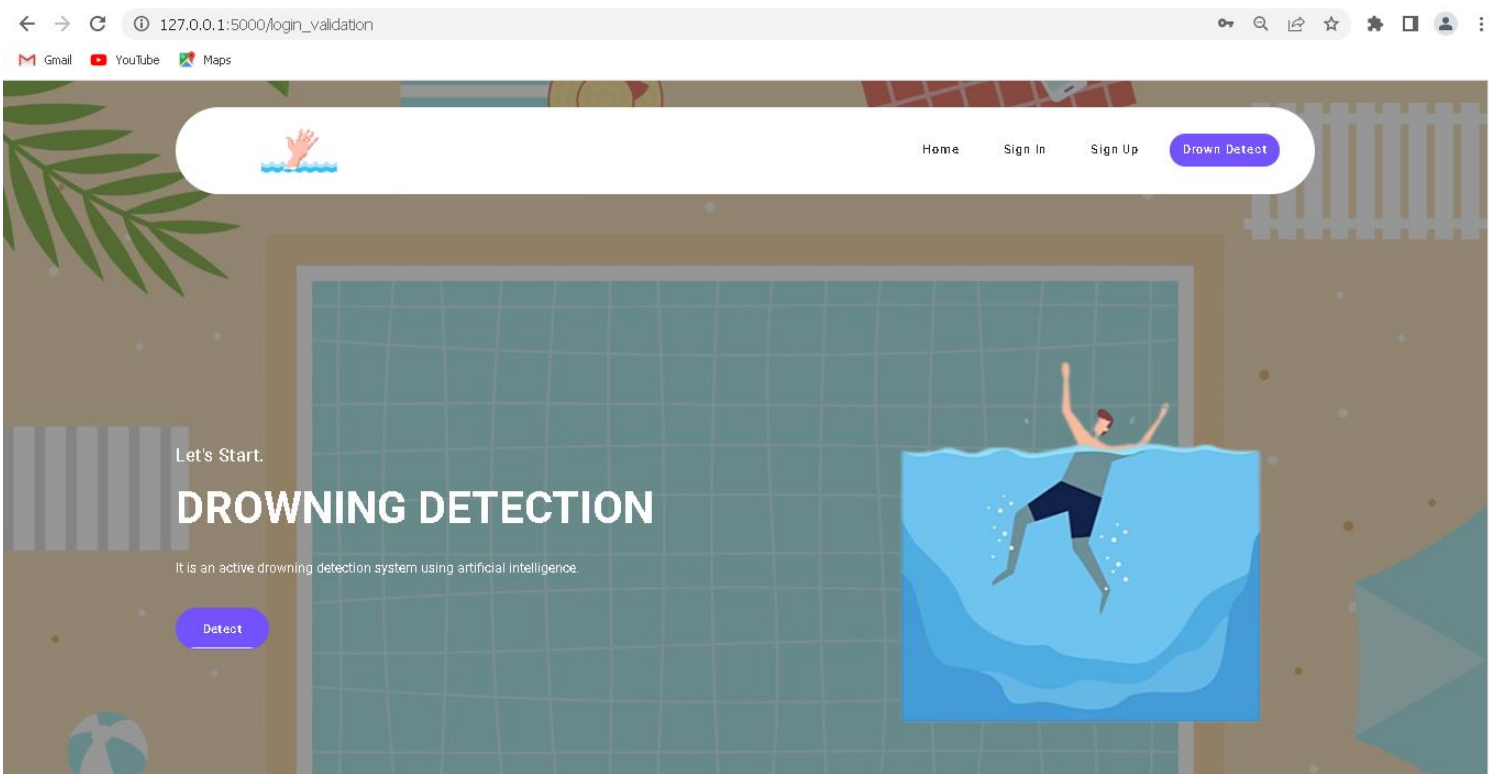
Showing 3 of 4 columns. ☐ Show all columns.

Showing document 1 - 1. Documents per page: 20

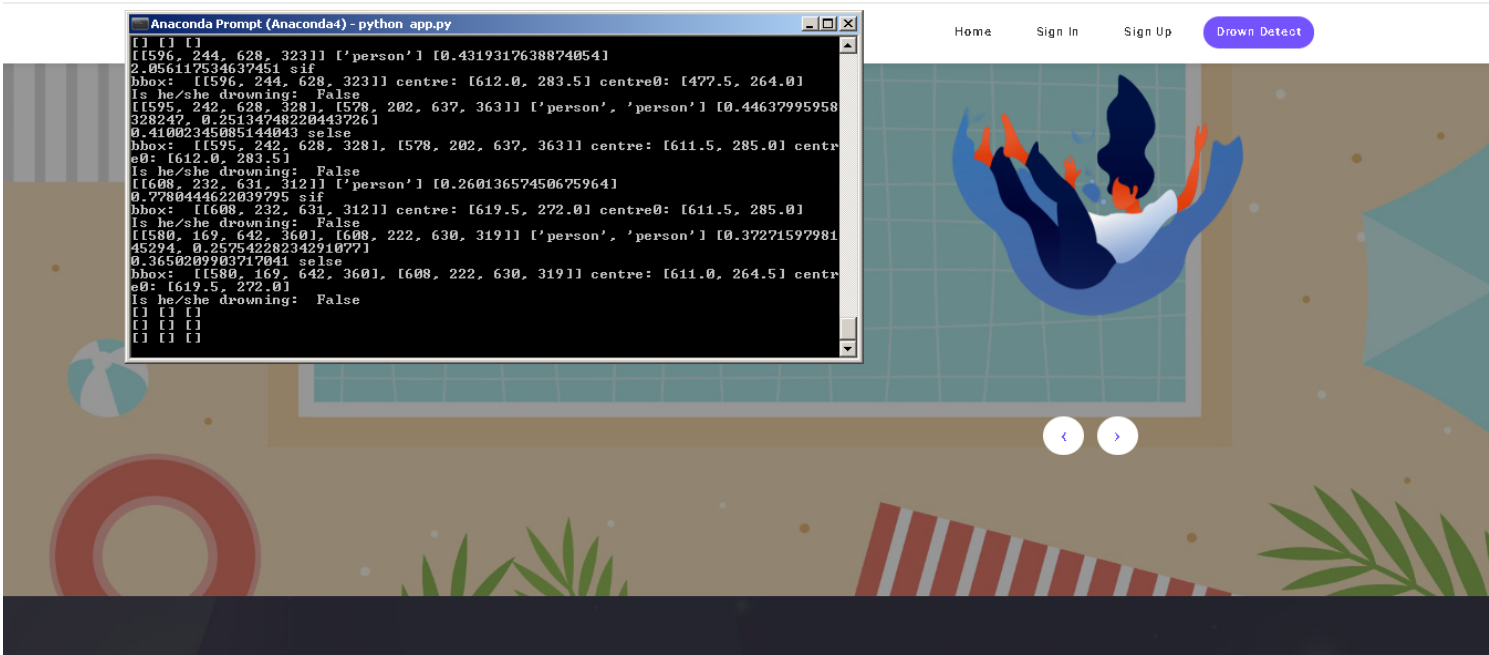
⏪

⏩

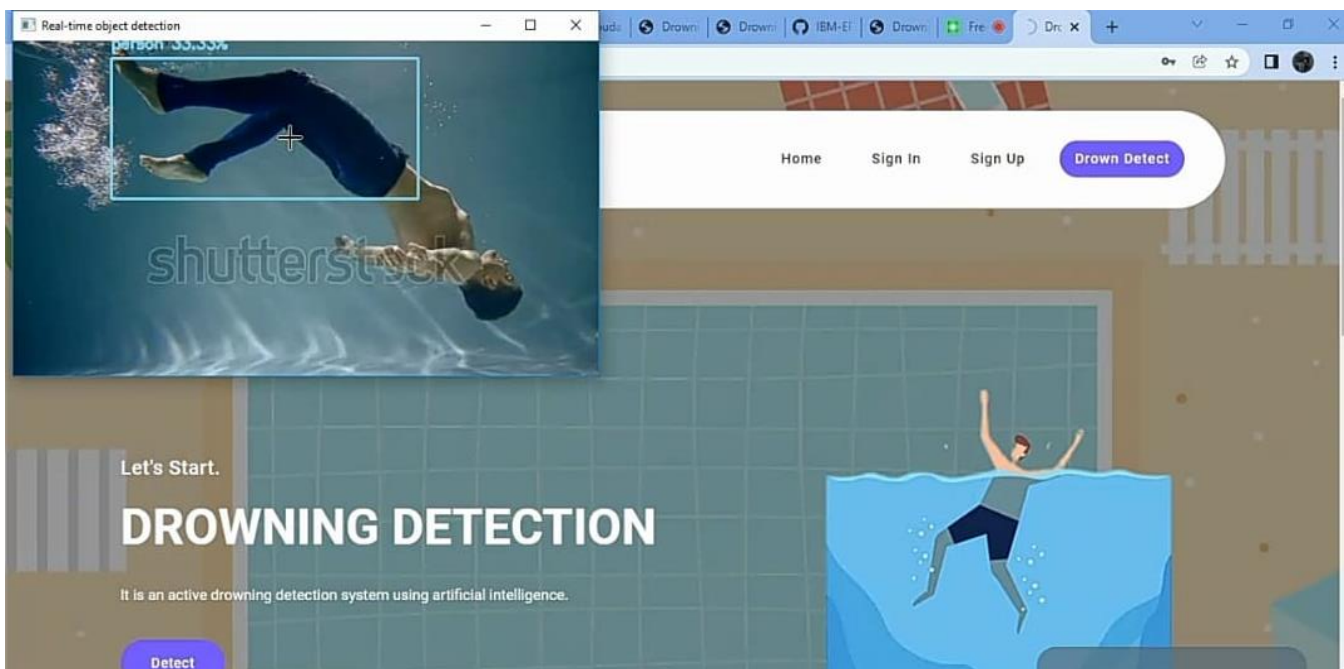
Detection Page:



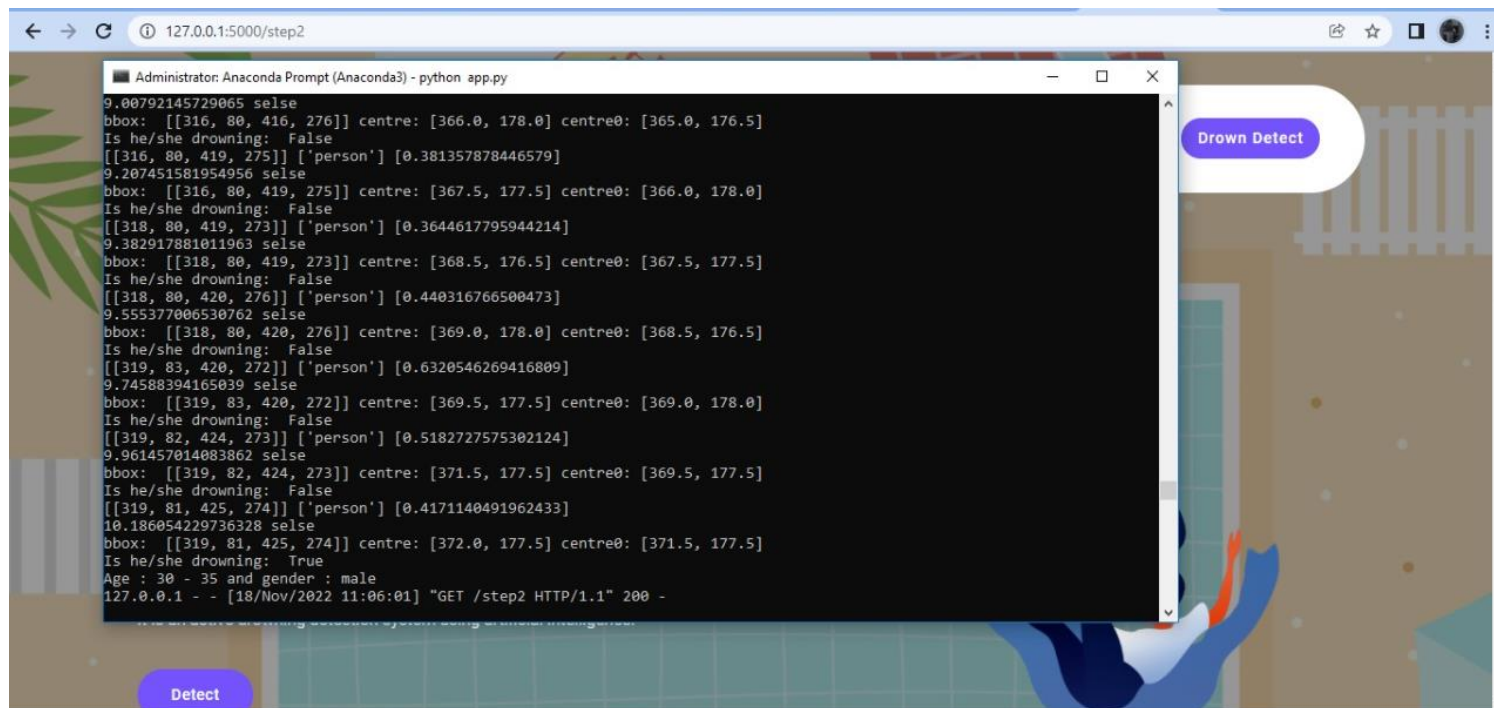
Before Drowning :



After Drowning:



Result :



The screenshot shows a web browser window at the URL `127.0.0.1:5000/step2`. The browser displays a background image of a person in a body of water. Overlaid on the browser is a terminal window titled "Administrator: Anaconda Prompt (Anaconda3) - python app.py". The terminal displays the output of a Python script, which includes bounding box coordinates, center coordinates, and a boolean result for "Is he/she drowning:". The final output line indicates "Age : 30 - 35 and gender : male" and "127.0.0.1 - - [18/Nov/2022 11:06:01] "GET /step2 HTTP/1.1" 200 -".

```
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'] [0.4171140491962433]
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 -
```


CHAPTER-8

TESTING

8.1 Test cases

			Date	22-Nov-22									
			Test ID	PC-0002 (Reliability)									
			Project Name	Virtual Eye - Life Guard for Swimming Pools to Detect Active Drowning									
			Maximum Marks	4 marks									
Test case ID	Feature Type	Component	Test Scenario	Pre-Requests	Steps To Execute	Test Data	Expected Results	Actual Results	Status	Comments	TC for Automation (Y/N)	Exec ID	Executed By
Homepage_TC_001	Functional	Homepage	Verify user is able to see the home page or not.		1. Enter URL, and click go 2. Verify whether the user is able to see the home page.	Enter URL, and click go	User able to see the home page	Working as expected	Pass	NI	N	-	Aarsha RK
Homepage_TC_002	UI	Homepage	Verify the UI elements in Home Page		1. Enter URL, and click go 2. Verify the UI elements in Home Page.		Application should show below UI elements	Working as expected	Pass	NI	N	-	Swarthika R
Registerpage_TC_003	Functional	Register page	A Register page is able to well input the user data.		1. Enter URL, and click go 2. Verify the UI elements in Home Page 3. Click the sign up button	Enter URL, and click go Click on sign up home page	Application should show 'Incorrect email' or 'password' validation message.	Working as expected	Pass	NI	N	-	Karthigan R
Loginpage_TC_004	Functional	Login page	Verify user is able to redirect to detect page or not.		1. Enter URL, and click go 2. Click on detect button 3. Verify whether the user is able to redirect to detect page or not.	Click on sign in home page	Application should show 'Incorrect email' or 'password' validation message.	Working as expected	Pass	NI	N	-	Jayashree K
Predictpage_TC_005	UI	Predict page	Verify the UI elements in Predict Page		1. Enter URL, and click go 2. Verify the UI elements in Predict Page.	Click the predict button and redirect to predict page	Application should show below UI elements: Dropdown List, detect button.	Working as expected	Pass	NI	N	-	Karthigan R, Jayashree K
Predictpage_TC_006	Functional	Predict page	Verify user is able to select the dropdown value or not.		1. Enter URL, and click go 2. Click on detect button 3. Verify whether the user is able to redirect to detect page or not. 4. Verify user is able to select the dropdown value or not.	Dropping or not	Application should show detecting video	Working as expected	Pass	NI	N	-	Aarsha RK, Swarthika R
Predictpage_TC_007	Functional	Predict page	Verify the video		1. Enter URL, and click go 2. Click on Predict button 3. Verify whether the user is able to select the dropdown value or not. 4. Verify user is able to select the dropdown value or not. 5. Verify the video	Predicting the video	Application should show the uploaded video	Working as expected	Pass	NI	N	-	Aarsha RK, Swarthika R
Predictpage_TC_008	Functional	Predict page	Verify whether the video is predicted Drowning, or not		1. Enter URL, and click go 2. Click on Predict button 3. Verify whether the user is able to redirect to predict page or not. 4. Verify user is able to select the dropdown value or not. 5. Verify whether the video is predicted correctly or not	Click the Detect button	Application shows the predicted output	Working as expected	Pass	NI	N	-	Aarsha RK, Swarthika R, Jayashree K, Karthigan R

8.2 User Acceptance Testing

- Purpose of Document**

The purpose of this document is to briefly explain the test coverage and open issues of the [Virtual Eye - Life Guard for Swimming Pools to Detect Active Drowning] project at the time of the release to User Acceptance Testing (UAT).

- Defect Analysis**

This reports how is the number of resolved or closed bugs at each severity level, and how they were resolved.

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**

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

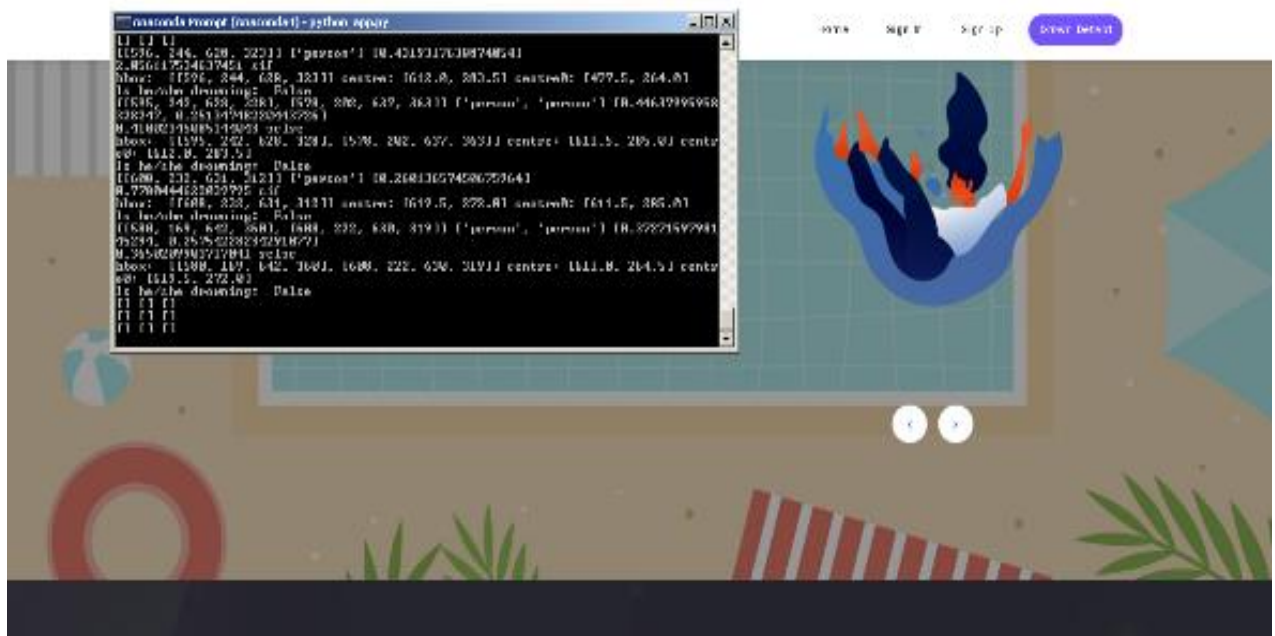
Section	Total Cases	Not Tested	Fail	Pass
Print Engine	5	0	0	5
Client Application	30	0	0	30
Security	2	0	0	2
Outsource Shipping	1	0	0	1
Exception Reporting	7	0	0	7
Final Report Output	9	0	0	9
Version Control	1	0	0	1

CHAPTER-9

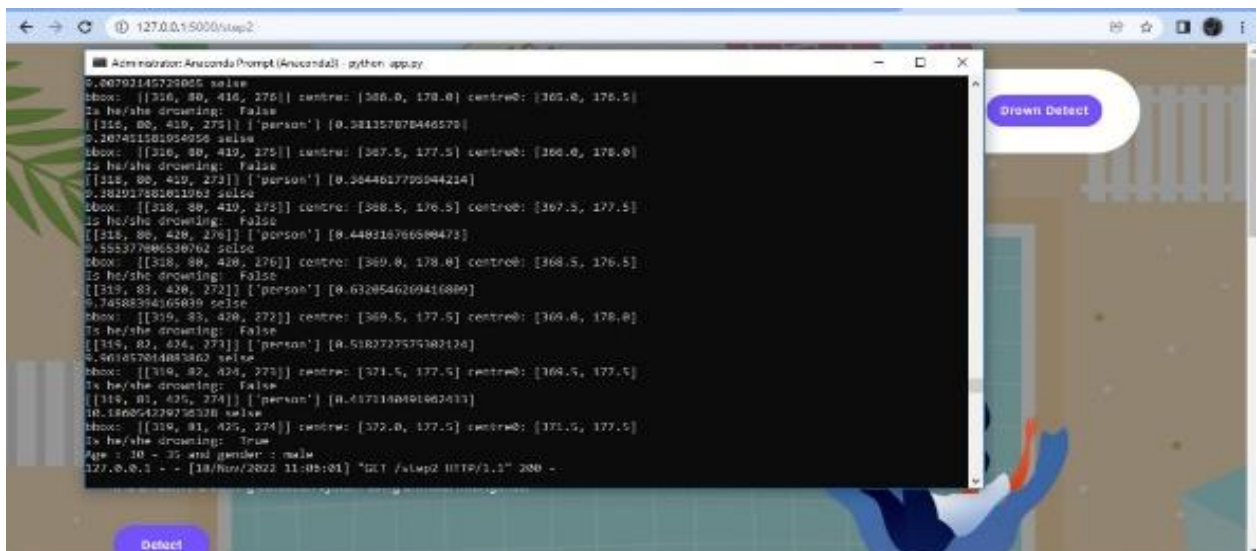
RESULT

9.1 Performance Metric

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 .

CHAPTER-13

APPENDIX

Github link: <https://bit.ly/3EU4T8r>

Demo Link: <https://bit.ly/3XsFsSA>