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

TEAM ID: PNT2022TMID02240

A PROJECT REPORT

Submitted by

ROHIT M	**(2116190701166)**
SAATHVIK KRISHNAN	**(2116190701171)**
SHRIJEETH S	**(2116190701203)**
SIDHARTH RAJ M	**(2116190701205)**

In partial fulfillment for the award of the degree

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE,

CHENNAI-602105

ANNA UNIVERSITY::CHENNAI 600025

TABLE OF CONTENTS

CHAPTER-1

Introduction

In modern metropolitan lifestyle, swimming is one of the finest activities for stress reduction. Swimming pools are more prevalent at hotels and weekend tourist destinations, and hardly anyone has one in their backyard. Beginners, in particular, frequently experience difficulty breathing underwater, which leads to respiratory issues, which ultimately leads to a drowning disaster. Without harming kids, drowning results in a higher death rate globally. The highest worldwide rates of drowning death are observed to occur among children under the age of six. With around 1.2 million incidents each year, these types of fatalities rank third among all unexpected deaths worldwide. To resolve this dispute, a careful system has to be put in place around the swimming pools to save human life.

1.1 Project Overview

We propose an underwater pool safety system that lowers the danger of drowning by looking at body movement patterns and integrating cameras with artificial intelligence (AI) technologies. Typically, such systems may be created by mounting cameras in the water, and then reviewing the video streams for any irregularities.

1.2 Purpose

The primary goal of this research is to prevent drowning by utilizing Yolo V5 object detection algorithm to analyze the swimmer's location. This function detects drowning when a person does not move or moves very slowly for 10 seconds. The system is not intended to take the position of a lifeguard or other human monitor, but rather to serve as an extra tool. "It aids the lifeguard in seeing underwater situations that are difficult for them to see.

CHAPTER-2

LITERATURE SURVEY

2.1 Existing Problem

Existing drowning detection methods include vision-based systems and wearable sensor-based systems. Vision-based technologies are further classified as those that employ underwater cameras and those that employ above-water cameras. Underwater cameras have the disadvantage of missing the first battle above the water. Failure to detect a drowning incident early on may result in a lengthier rescue time, which is a key factor to consider in a time-critical emergency. The major downside of a wearable-based system is the pain of use, which may lead to younger children seeking relief by removing the device, which is an unfounded notion.

2.2 References

- 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
- Health and Safety Executive. (2018). HSG179, Health and safety in swimming pools (Fourth edition)
- AngelEye.(2019).AngelEye–Distributors.Retrieved from: https://www.angeleye.it/news. php?id=28&newscat=10
- 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).

2.3 Problem Statement Definition

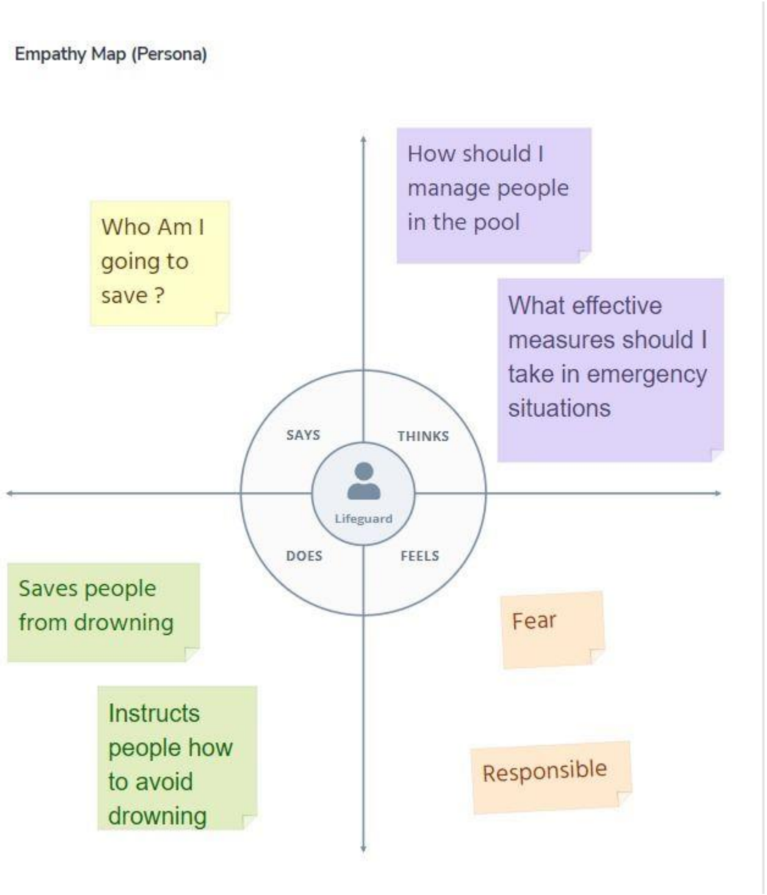
Swimming is one of the best exercises that helps people to reduce stress in this urban lifestyle. Swimming pools are found larger in number in hotels, and weekend tourist spots and barely people have them in their house backyard. Beginners, especially, often feel it difficult to breathe underwater which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

CHAPTER-3

IDEATION AND PROPOSED SOLUTION

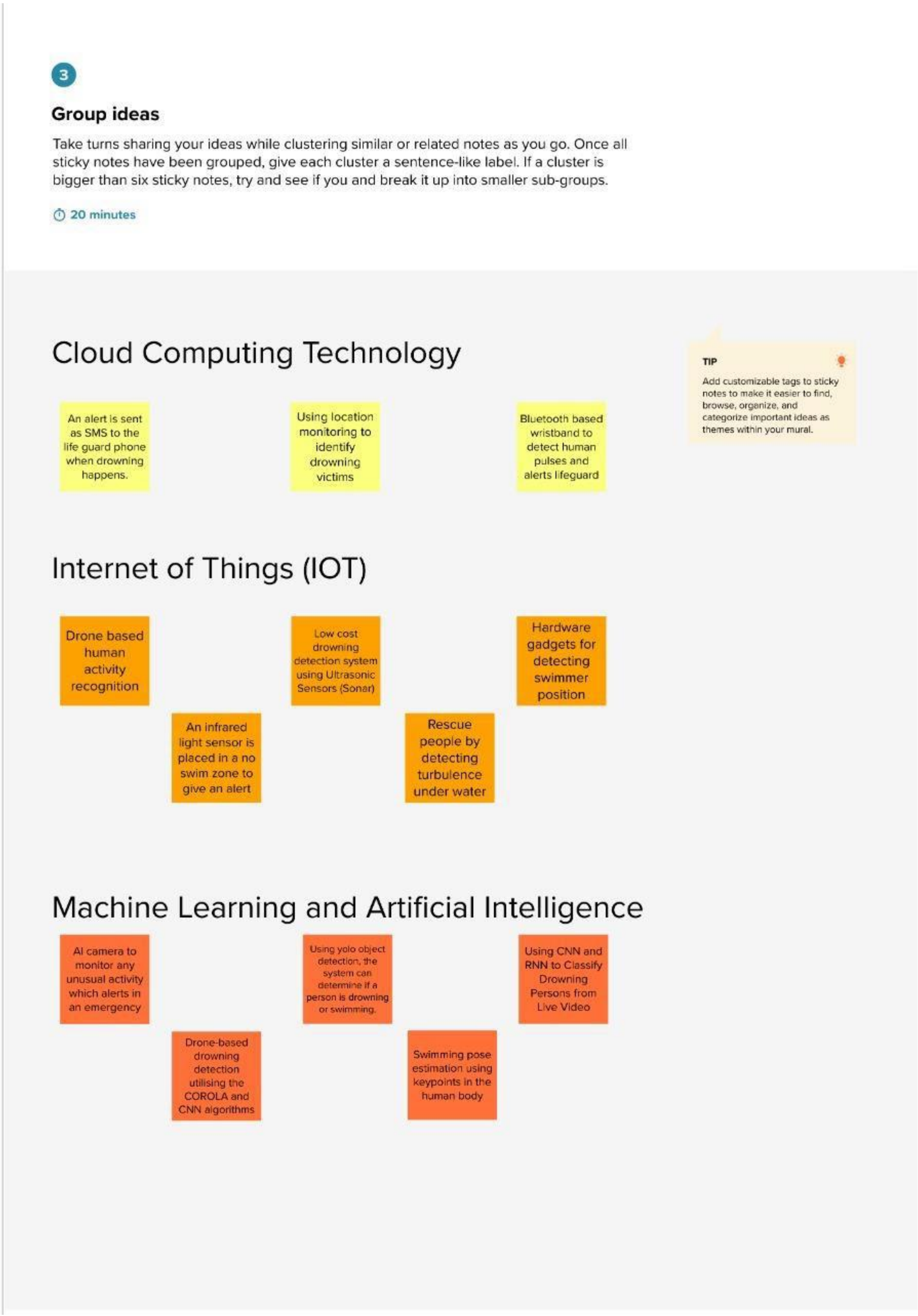
3.1 Empathy Map Canvas

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.


Empathy Map (Persona)
How should I manage people in the pool
Who Am I going to save ?
What effective measures should I take in emergency situations
SAYS
THINKS
Lifeguard
DOES
FEELS
Saves people from drowning
Fear
Instructs people how to avoid drowning
Responsible

3.2 Big Ideas

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



3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, 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

Cloud Computing Technology

- An alert is sent as SMS to the life guard phone when drowning happens.
- Using location monitoring to identify drowning victims
- Bluetooth based wristband to detect human pulses and alerts lifeguard

TIP

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

Internet of Things (IOT)

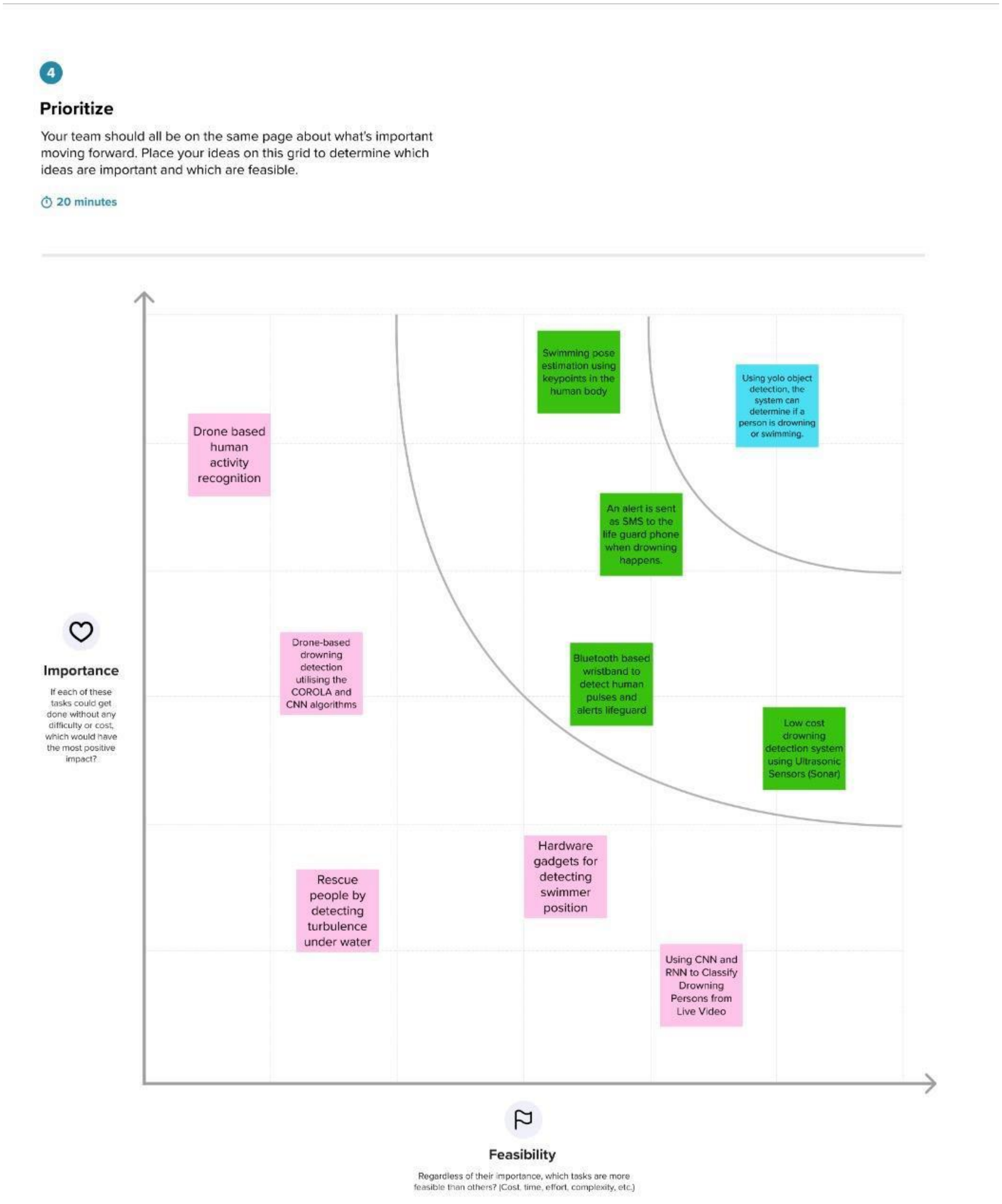
- Drone based human activity recognition
- Low cost drowning detection system using Ultrasonic Sensors (Sonar)
- Hardware gadgets for detecting swimmer position
- An infrared light sensor is placed in a no swim zone to give an alert
- Rescue people by detecting turbulence under water

Machine Learning and Artificial Intelligence

- AI camera to monitor any unusual activity which alerts in an emergency
- Using yolo object detection, the system can determine if a person is drowning or swimming.
- Using CNN and RNN to Classify Drowning Persons from Live Video
- Drone-based drowning detection utilising the COROLA and CNN algorithms
- Swimming pose estimation using keypoints in the human body

3.3 Idea Prioritization

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



4

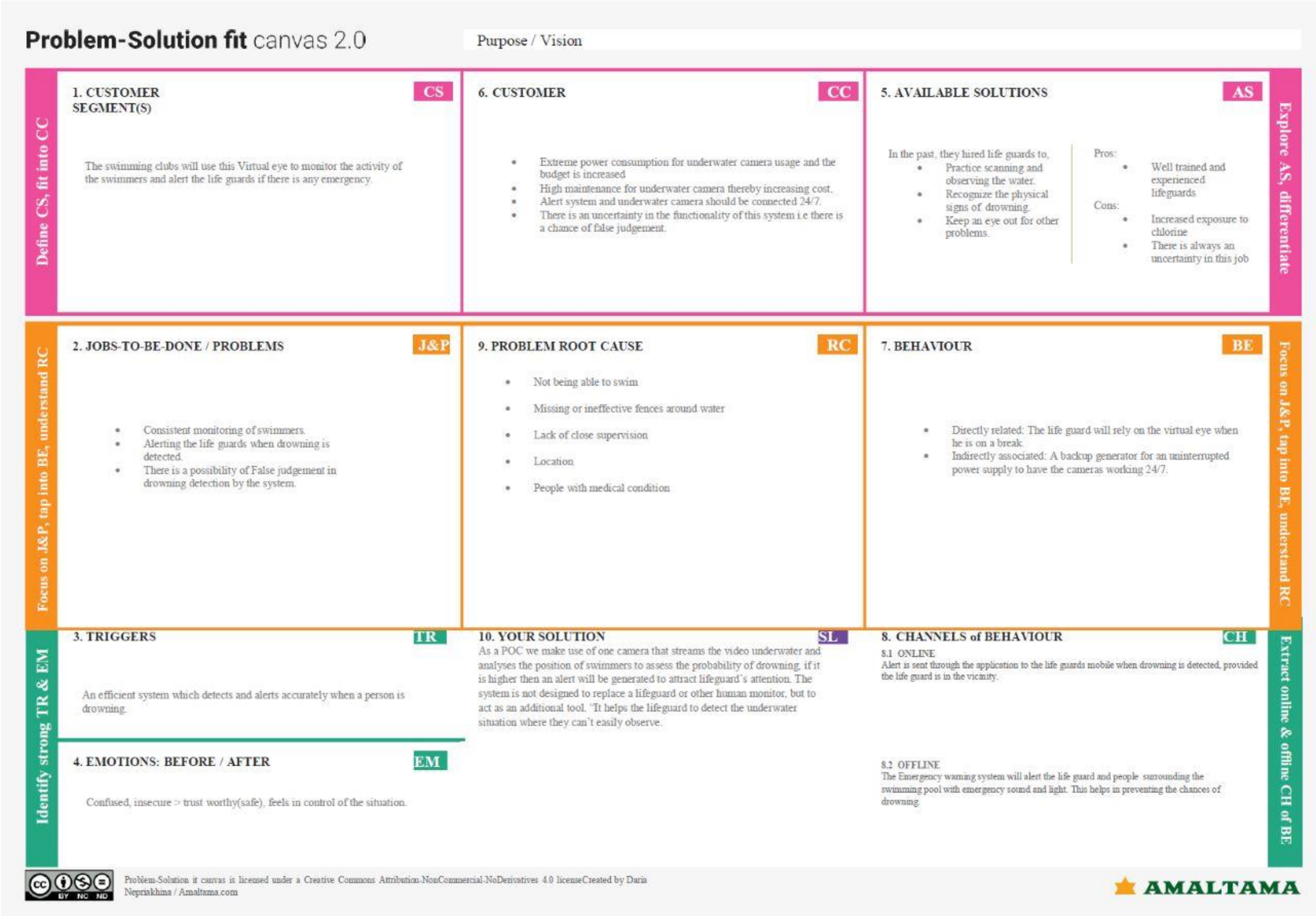
Prioritize

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

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.)
Swimming pose estimation using keypoints in the human body
Using yolo object detection, the system can determine if a person is drowning or swimming.
Drone based human activity recognition
An alert is sent as SMS to the life guard phone when drowning happens.
Drone-based drowning detection utilising the COROLA and CNN algorithms
Bluetooth based wristband to detect human pulses and alerts lifeguard
Low cost drowning detection system using Ultrasonic Sensors (Sonar)
Hardware gadgets for detecting swimmer position
Rescue people by detecting turbulence under water
Using CNN and RNN to Classify Drowning Persons from Live Video

3.4 Problem Solution Fit


Define CS, fit into CC
Explore AS, differentiate
Focus on J&P, tap into BE, understand RC
Focus on J&P, tap into BE, understand RC
Identify strong TR & EM
Extract online & offline CH of BE
CS
CC
AS
J&P
RC
BE
TR
SL
CH
EM
AMALTAMA

Problem-Solution fit canvas 2.0

Purpose / Vision

1. CUSTOMER SEGMENT(S)

The swimming clubs will use this Virtual eye to monitor the activity of the swimmers and alert the life guards if there is any emergency.

6. CUSTOMER

- Extreme power consumption for underwater camera usage and the budget is increased
- High maintenance for underwater camera thereby increasing cost.
- Alert system and underwater camera should be connected 24/7.
- There is an uncertainty in the functionality of this system i.e there is a chance of false judgement.

5. AVAILABLE SOLUTIONS

In the past, they hired life guards to,

- Practice scanning and observing the water.
- Recognize the physical signs of drowning.
- Keep an eye out for other problems.

Pros:

- Well trained and experienced lifeguards

Cons:

- Increased exposure to chlorine
- There is always an uncertainty in this job

2. JOBS-TO-BE-DONE / PROBLEMS

- Consistent monitoring of swimmers.
- Alerting the life guards when drowning is detected.
- There is a possibility of False judgement in drowning detection by the system.

9. PROBLEM ROOT CAUSE

- Not being able to swim
- Missing or ineffective fences around water
- Lack of close supervision
- Location
- People with medical condition

7. BEHAVIOUR

- Directly related: The life guard will rely on the virtual eye when he is on a break.
- Indirectly associated: A backup generator for an uninterrupted power supply to have the cameras working 24/7.

3. TRIGGERS

An efficient system which detects and alerts accurately when a person is drowning.

4. EMOTIONS: BEFORE / AFTER

Confused, insecure > trust worthy(safe), feels in control of the situation.

10. YOUR SOLUTION

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 lifeguard's attention. 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.

8. CHANNELS of BEHAVIOUR

8.1 ONLINE
Alert is sent through the application to the life guards mobile when drowning is detected, provided the life guard is in the vicinity.

8.2 OFFLINE
The Emergency warning system will alert the life guard and people surrounding the swimming pool with emergency sound and light. This helps in preventing the chances of drowning.

3.5 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	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.
2.	Idea/Solution description	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 analyzing 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. 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.

3.	Novelty/Uniqueness	Instead of using 16 cameras we are using only a single camera to analyze the position of swimmers and to assess the probability of drowning.
4.	Social Impact/Customer Satisfaction	Saving people at the right time of drowning and active watching of the swimming pool for any such incidents. Due to increasing deaths in swimming people because of drowning our project will be very much useful in saving lives of people in a very short time..
5.	Business Model(Revenue Model)	Since there is a system to ensure the safety of swimmers, It will attract more people to learn swimming and boost the business.
6.	Scalability of the Solution	Our software system can be used by thecompany driver whom an ages the pools. We use the IBM cloud server to collect andmaintain 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	User Registration	Registration through Form Registration through Gmail.
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Alarm system	Monitor and detect the drowning person Alert the lifeguard by trigger the alarm
FR-4	Output	Visual representation Image detection Report generation

4.2 Non-Functional Requirements

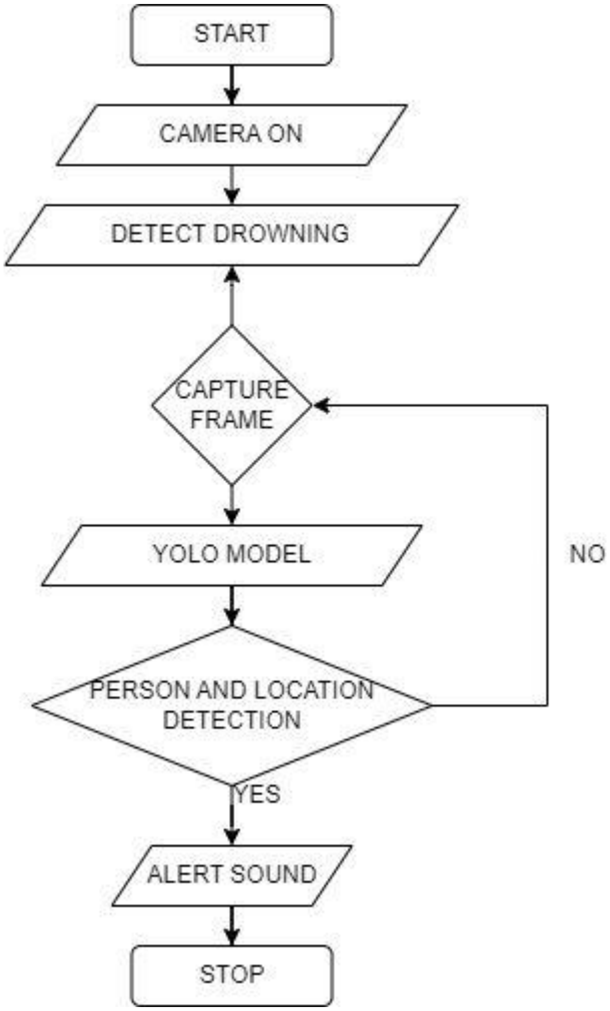
FRNo.	Non-Functional Requirement	Description
NFR-1	**Usability**	Eco – Friendly.

NFR-2	**Security**	Observing each and every body movement of the swimmers.
NFR-3	**Reliability**	Suitable for all the swimming pools.
NFR-4	**Performance**	Life guard can visually access the developing situation within seconds of the event first occurring and initiate the rescue procedure when necessary
NFR-5	**Availability**	24/7 monitoring cameras
NFR-6	**Scalability**	Its comfortable for all swimmers. The lifespan is high. Work more efficiently.

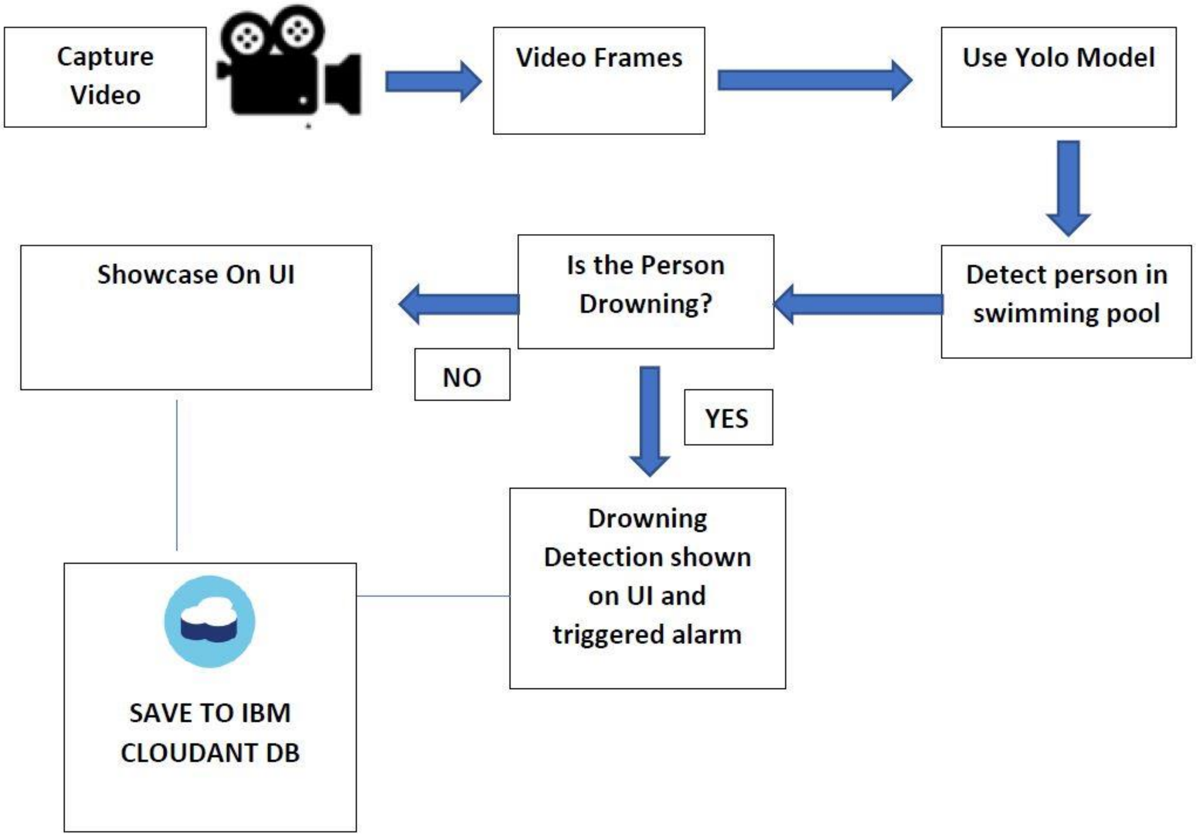
CHAPTER-5

PROJECTDESIGN

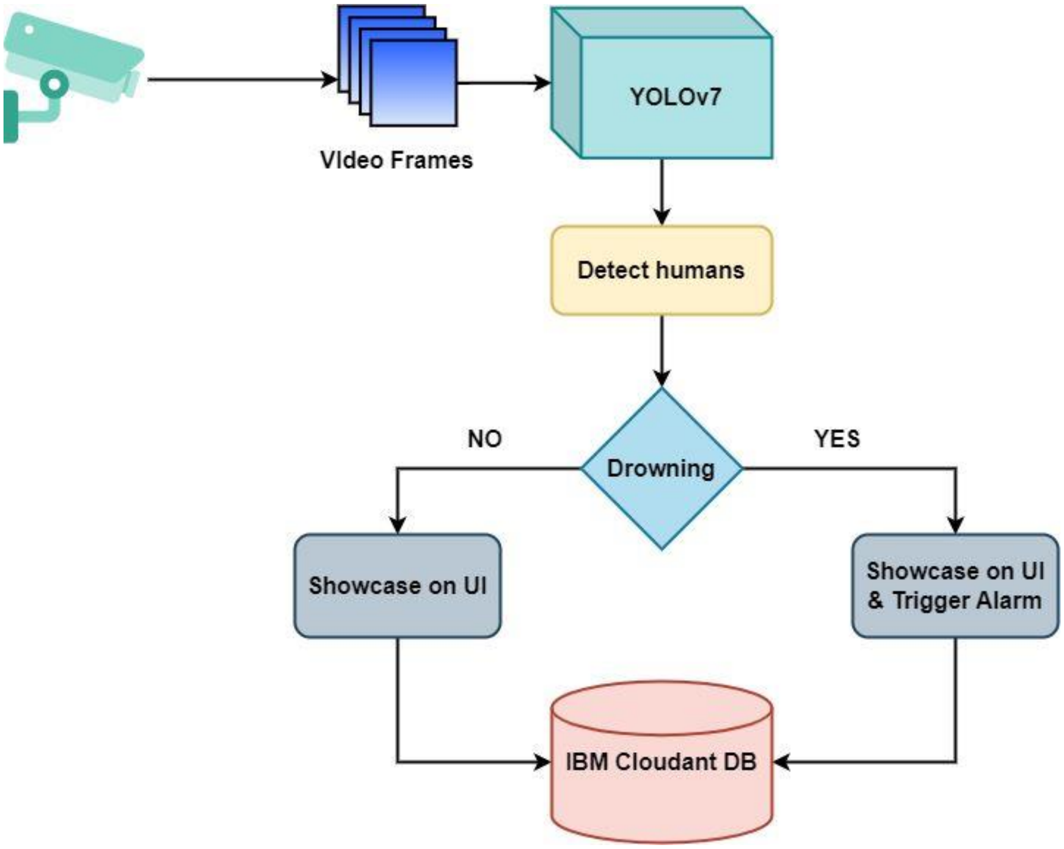
5.1 Data Flow Diagram



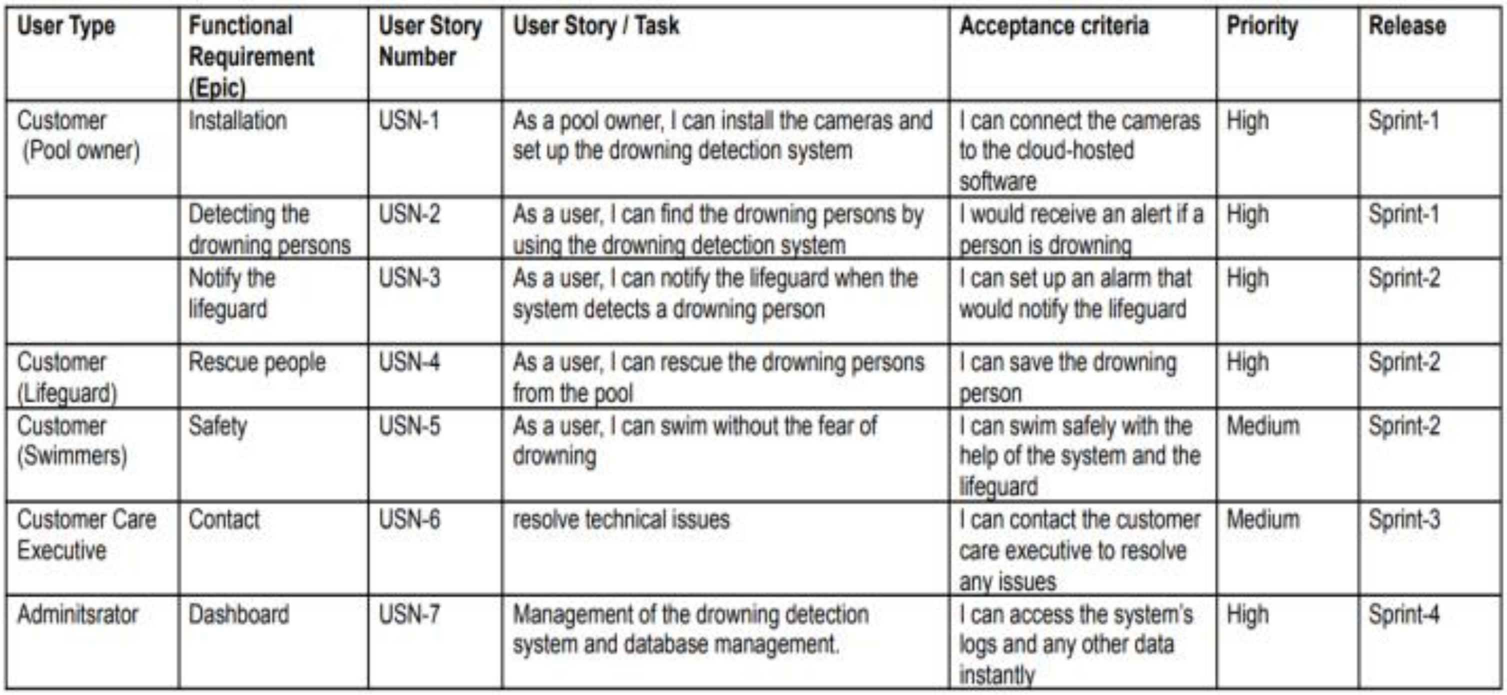
5.2 Solution Architecture



Technical Architecture



5.3 User stories

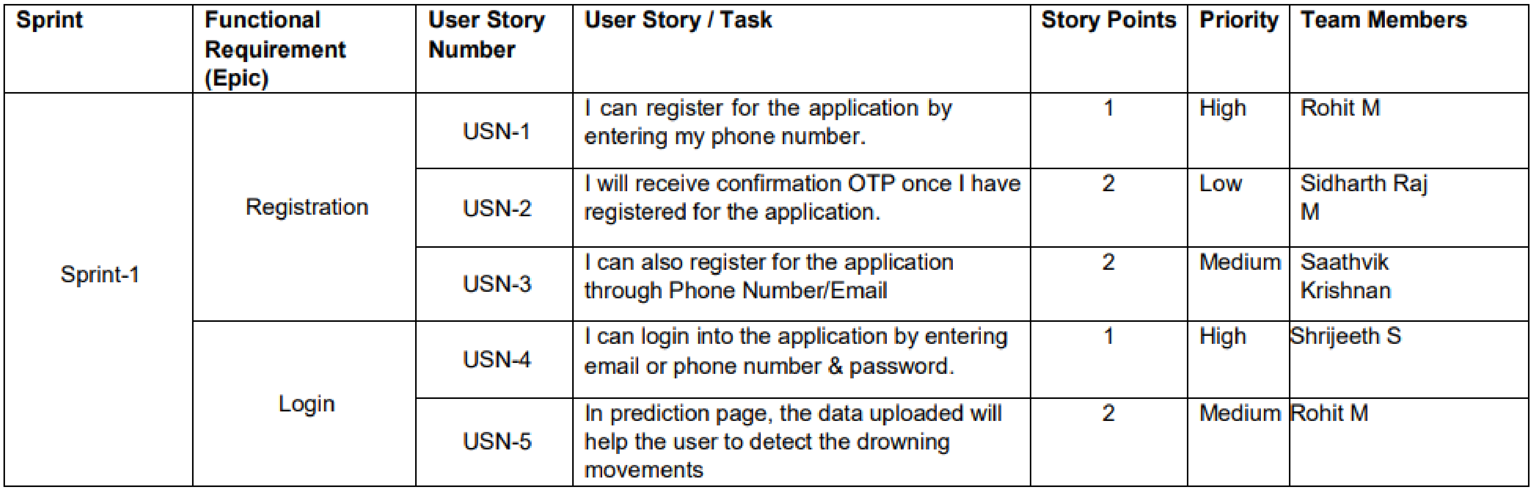


User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Pool owner)	Installation	USN-1	As a pool owner, I can install the cameras and set up the drowning detection system	I can connect the cameras to the cloud-hosted software	High	Sprint-1
	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
Adminitsrator	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

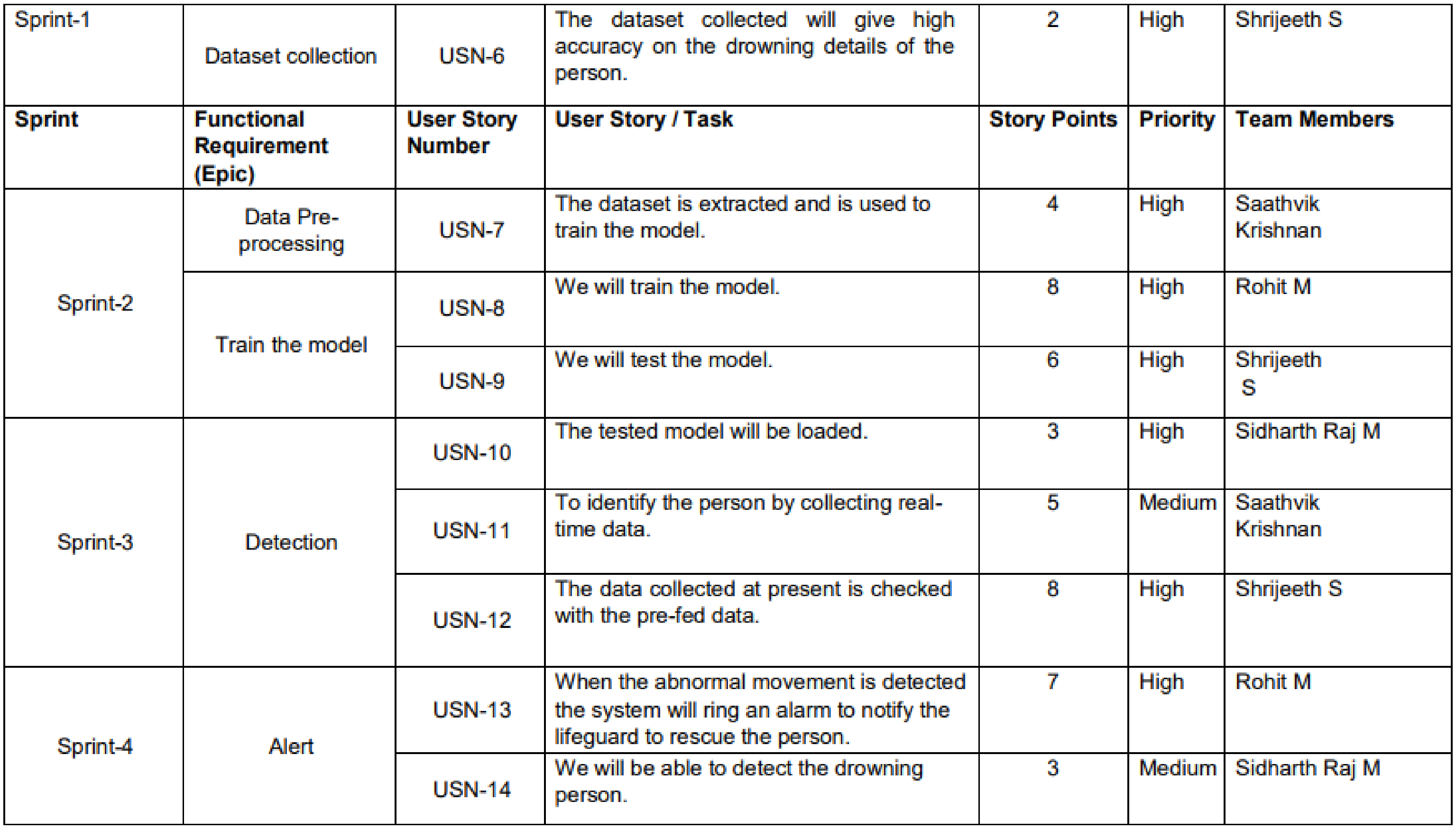
CHAPTER-6

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	I can register for the application by entering my phone number.	1	High	Rohit M
		USN-2	I will receive confirmation OTP once I have registered for the application.	2	Low	Sidharth Raj M
		USN-3	I can also register for the application through Phone Number/Email	2	Medium	Saathvik Krishnan
	Login	USN-4	I can login into the application by entering email or phone number & password.	1	High	Shrijeeth S
		USN-5	In prediction page, the data uploaded will help the user to detect the drowning movements	2	Medium	Rohit M



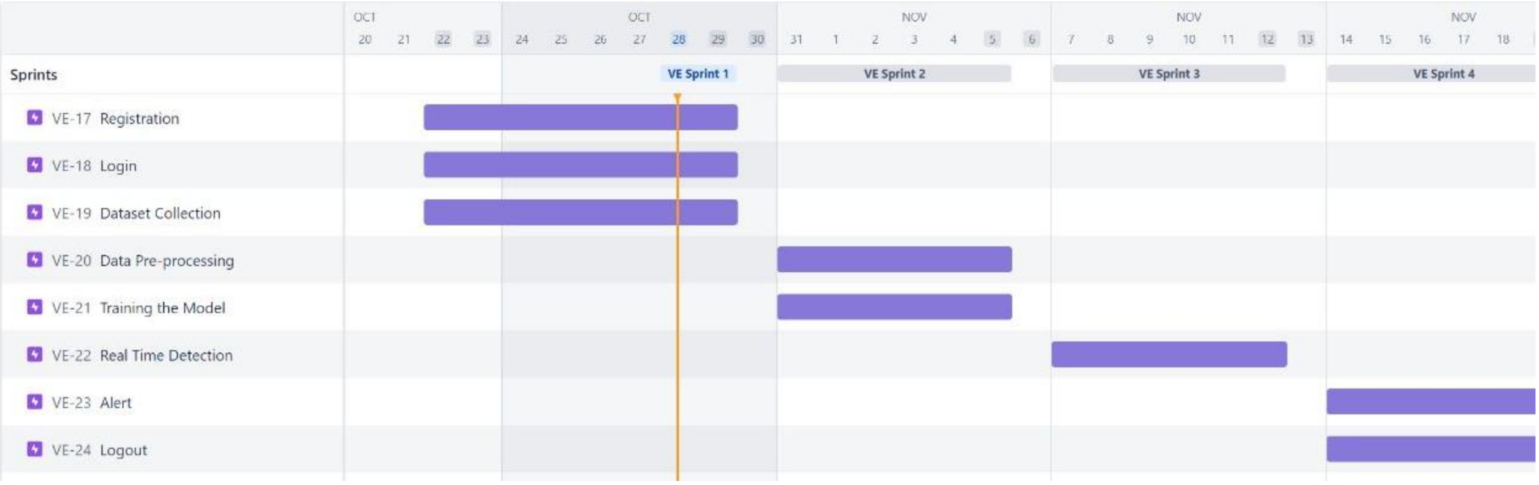
Sprint-1	Dataset collection	USN-6	The dataset collected will give high accuracy on the drowning details of the person.	2	High	Shrijeeth S
Sprint	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Story Points**	**Priority**	**Team Members**
Sprint-2	Data Pre-processing	USN-7	The dataset is extracted and is used to train the model.	4	High	Saathvik Krishnan
	Train the model	USN-8	We will train the model.	8	High	Rohit M
		USN-9	We will test the model.	6	High	Shrijeeth S
Sprint-3	Detection	USN-10	The tested model will be loaded.	3	High	Sidharth Raj M
		USN-11	To identify the person by collecting real-time data.	5	Medium	Saathvik Krishnan
		USN-12	The data collected at present is checked with the pre-fed data.	8	High	Shrijeeth S
Sprint-4	Alert	USN-13	When the abnormal movement is detected the system will ring an alarm to notify the lifeguard to rescue the person.	7	High	Rohit M
		USN-14	We will be able to detect the drowning person.	3	Medium	Sidharth Raj M

Sprint-4	Logout	USN-15	User can logout of the application.	2	Low	Sidharth Raj M

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	10	6Days	24Oct2022	29Oct2022	10	29Oct2022
Sprint-2	18	6Days	31Oct2022	05Nov2022	18	05Nov2022
Sprint-3	16	6Days	07Nov2022	12Nov2022	16	12Nov2022
Sprint-4	12	6Days	14Nov2022	19Nov2022	12	19Nov2022

6.3 Reports from JIRA



CHAPTER-7

CODING AND SOLUTION

7.1 Feature 1

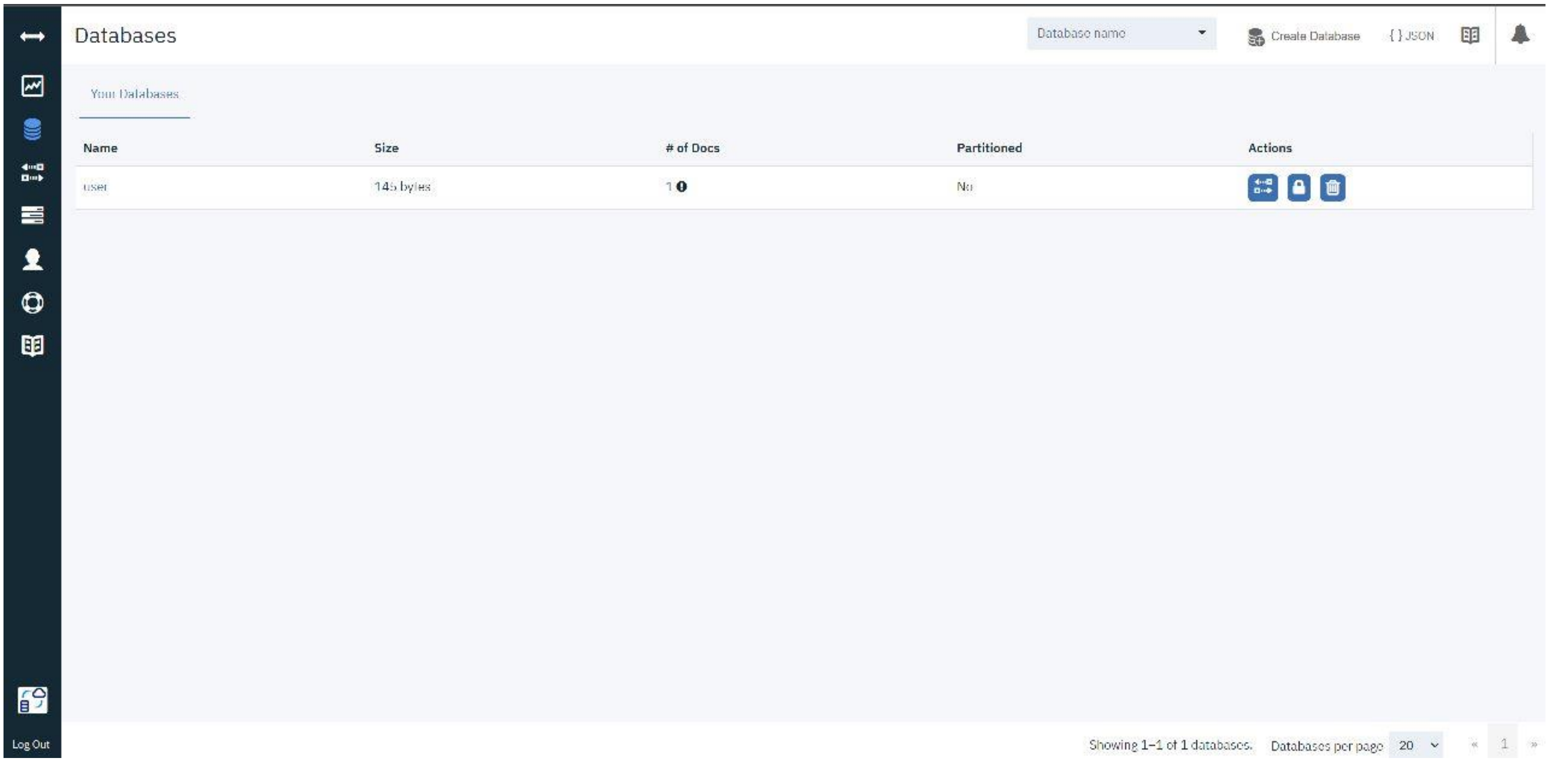
- In order to manage a connection from a local system we must first initialize the connection by constructing a Cloudant client. We need to import the cloudant library.
- IBM Cloud Identity & Access Management enables us to securely authenticate users and control access to all cloud resources consistently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

1. Once a connection is established we can create a database, open an existing database.
2. Create a database as my_database.

CODE

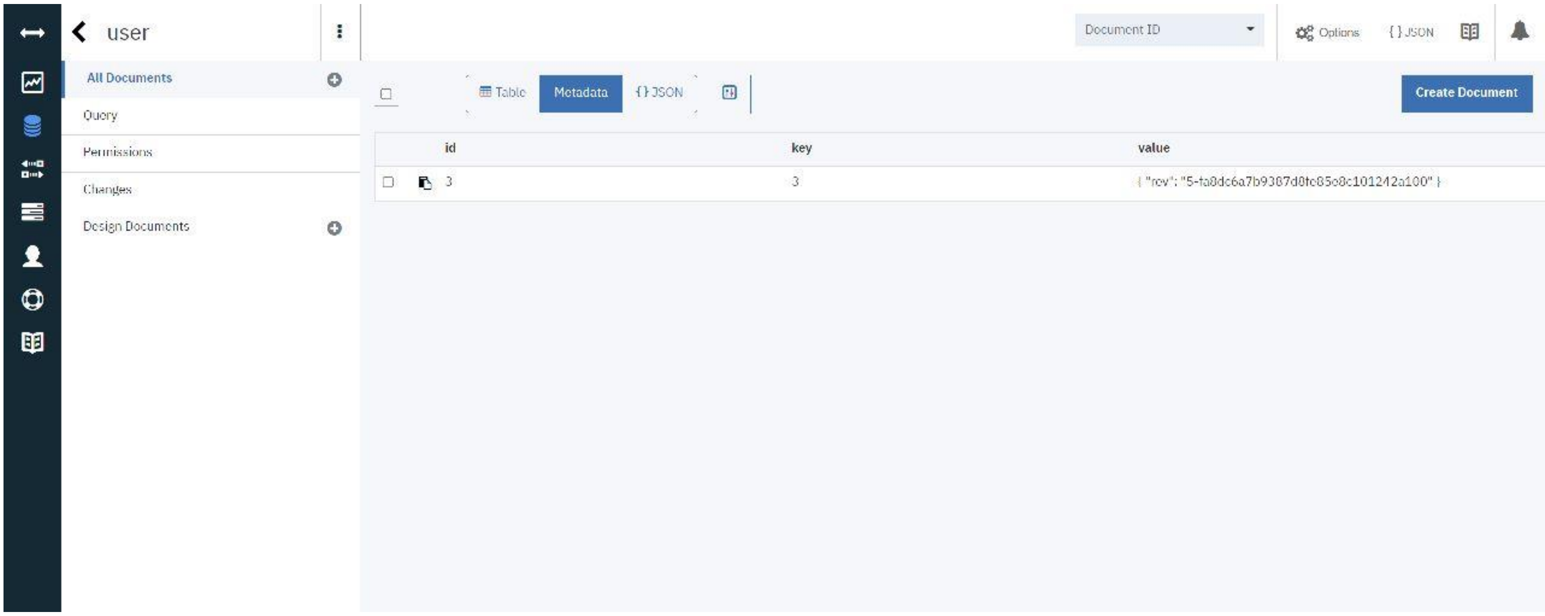
```
from cloudant.client
import Cloudant
client = Cloudant.iam(
 '08c5a12f-25fd-49c6-tbfa-de80ad989d12-cloudant','Rnz_zCc7hN5Lb5uRHaxn-
WrlN9yqbtz4QKlFVZ4ETZpk',connect=True)
name = 'name'
email = 'a@b.c'
password = '123'
my_database = client.create_database('my_database')
```

Output



Name	Size	# of Docs	Partitioned	Actions
user	145 bytes	1	No	

Showing 1–1 of 1 databases. Databases per page 20



	id	key	value
□	3	3	{ "rev": "5-fa8dc6a7b9387d0fc85e8c101242a100" }

7.2 Feature 2

App.py

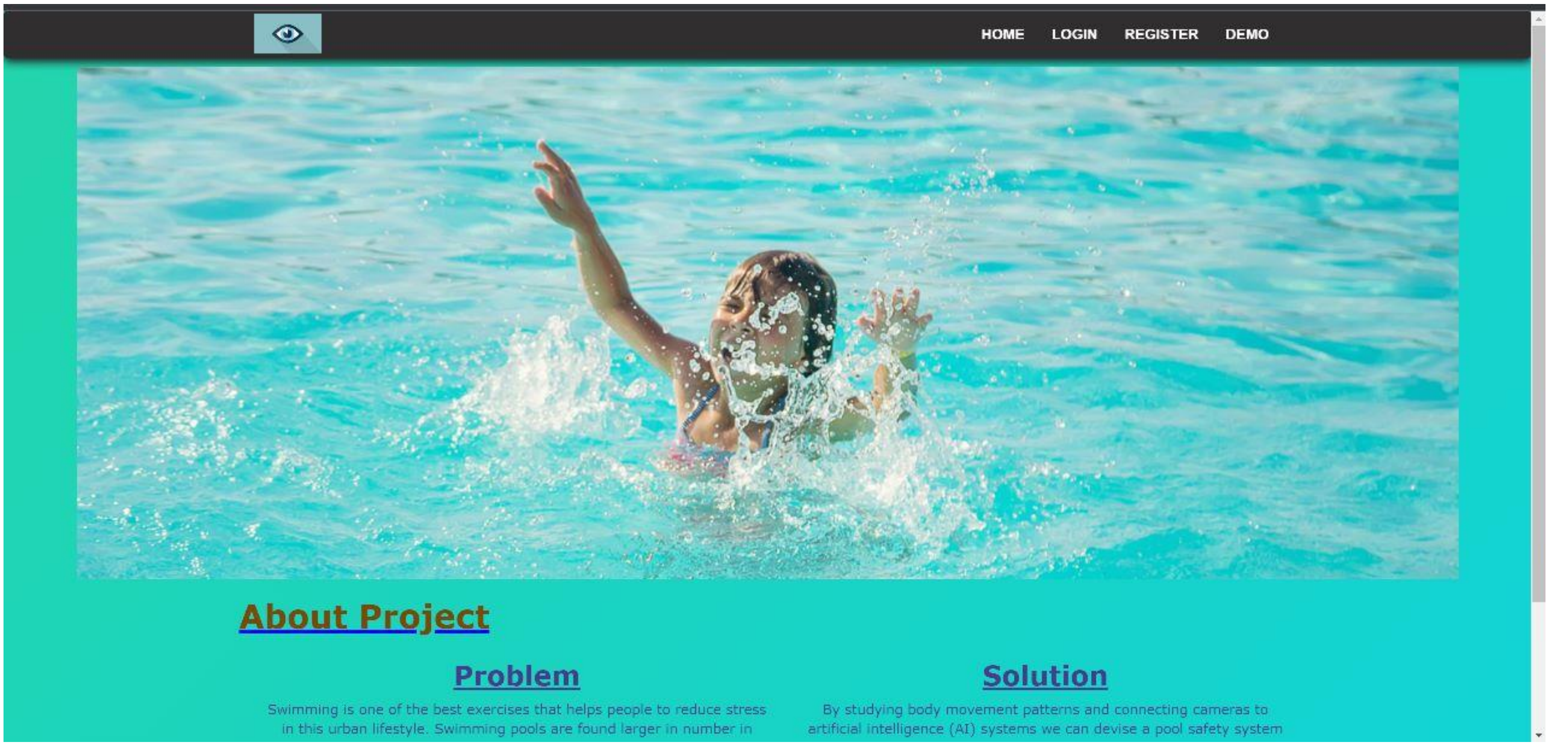
```
import datetime
from flask import Flask, render_template, request, redirect, session, url_for, Response
from flask_caching import Cache
from ibmcloudant.cloudant_v1 import CloudantV1, Document
import hashlib
import os
from dotenv import load_dotenv
from sendgrid import SendGridAPIClient
from sendgrid.helpers.mail import Mail, To, Email
import string
import random
import torch
import cv2
import time
from playsound import playsound


load_dotenv("./.env")
app = Flask(_name_)
app.config["SECRET_KEY"] = "r3qwrqweqq2r324ewf"
app.config["CACHE_TYPE"] = "SimpleCache"
cache = Cache(app)


service = CloudantV1.new_instance()
user_id = int(service.get_database_information(db=os.getenv("USER_DB")).get_result()['doc_count']) + 1
model = torch.hub.load("ultralytics/yolov5", "yolov5m")


def user_exists(email_id):
    query = {"email": email_id}
    result = service.post_find(os.getenv("USER_DB"), selector=query).get_result()['docs']
    return len(result) == 1, result


def hash_text(text, start_salt="123", end_salt="789"):
    original_text = start_salt + text + end_salt
    return hashlib.sha256(original_text.encode()).hexdigest()


def hash_password(email, password):
    return hash_text(
        password,
        hash_text(email, os.getenv("VIRTUAL_EYE_START_SALT"),
os.getenv("VIRTUAL_EYE_END_SALT")),
        hash_text(email, os.getenv("VIRTUAL_EYE_START_SALT"),
os.getenv("VIRTUAL_EYE_END_SALT"))
    )


def send_registration_mail(email, username):
    from_email = Email(email=os.getenv("SENDGRID_FROM_MAIL"))
    to_emails = [To(email=email, dynamic_template_data={"first_name": username})]
    message = Mail(from_email=from_email, to_emails=to_emails)
    message.template_id = os.getenv("SENDGRID_REGISTER_TEMPLATE_ID")
    try:
        sendgrid_client = SendGridAPIClient(os.getenv("SENDGRID_APIKEY"))
        response = sendgrid_client.send(message)
        return response.status_code == 202
    except Exception as e:
        print(e)
        return False


def send_forgot_password_mail(email, pass_code):
    from_email = Email(email=os.getenv("SENDGRID_FROM_MAIL"))
    to_emails = [To(email=email, dynamic_template_data={"password_code": pass_code})]
    message = Mail(from_email=from_email, to_emails=to_emails)
    message.template_id = os.getenv("SENDGRID_FORGOT_PASSWORD_TEMPLATE_ID")
    try:
        sendgrid_client = SendGridAPIClient(os.getenv("SENDGRID_APIKEY"))
        response = sendgrid_client.send(message)
        return response.status_code == 202
    except Exception as e:
        print(e)
        return False


def generate_passcode():
    code = ''.join(random.choices(string.ascii_uppercase + string.ascii_lowercase + string.digits, k=6))
    return str(code)


def detect_person(image):
    detection_results = model(image)
    persons = []
    for detections in detection_results.xyxy[0]:
        if detections[-1] == 0:
            persons.append(detections[:-1])
    return persons


def is_above_threshold(person_bbox, center0, threshold=10):
    center = [(person_bbox[0] + person_bbox[2]) / 2, (person_bbox[1] + person_bbox[3]) / 2]
    hmov = abs(center[0] - center0[0])
    vmov = abs(center[1] - center0[1])
    if (hmov > threshold) or (vmov > threshold):
        return True, center
    return False, center


def gen_frames(src, dest):
    webcam = cv2.VideoCapture(src)
    fourcc = cv2.VideoWriter_fourcc(*'mp4v')
    out = cv2.VideoWriter(dest, fourcc, 20.0, (640, 640))
    t0 = dict()
    isDrowning = dict()
    center0 = dict()
    center = dict()
    start = time.time()
    playFrame = 0
    limit = 0
    while webcam.isOpened():
        limit = time.time() - start
        status, frame = webcam.read()
        if not status:
            break
        if frame is None:
            continue
        frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        frame = cv2.resize(frame, (640, 640))
        persons = detect_person(frame)
        if len(persons) == 0:
            limit = 0
        for ind, person in enumerate(persons):
            person = list(map(int, person.cpu().numpy().round().tolist()))
            t0[ind] = t0.get(ind, time.time())
            isDrowning[ind] = isDrowning.get(ind, False)
            center0[ind] = center0.get(ind, [0, 0])
            bbox = person.copy()
            x = time.time()
            aboveThresh, center = is_above_threshold(bbox, center0[ind], threshold=30)
            if aboveThresh:
                t0[ind] = time.time()
                isDrowning[ind] = False
            else:
                if time.time() - t0[ind] > 20:
                    isDrowning[ind] = True
            center0[ind] = center
            start_point = (person[0], person[1])
            end_point = (person[2], person[3])
            if isDrowning[ind]:
                color = (255, 0, 0)
            else:
                color = (0, 0, 255)
            thickness = 2
            frame = cv2.rectangle(frame, start_point, end_point, color, thickness)
            frame = cv2.cvtColor(frame, cv2.COLOR_RGB2BGR)
        for person_id, drown_status in isDrowning.items():
            if drown_status:
                try:
                    if playFrame % 100 == 0:
                        print(f"Drowning Detected on {datetime.datetime.now()}")
                        playsound(os.path.dirname(_file_) + "\\static\\sounds\\alarm.mp3.wav")
                        playFrame = 0
                except Exception as e:
                    continue
                playFrame += 1
        out.write(frame)
        ret, buffer = cv2.imencode('.jpg', frame)
        buffer = buffer.tobytes()
        yield (b'--frame\r\n'
               b'Content-Type: image/jpeg\r\n\r\n' + buffer + b'\r\n')
    webcam.release()
    out.release()


@app.route("/")
def index():
    return render_template("index.html")


@app.route("/login", methods=["GET", "POST"])
def login():
    if request.method == "POST":
        email = request.form.get("email")
        password = hash_password(email, request.form.get("password"))
        exist, result = user_exists(email)
        if exist:
            if result[0]['password'] == password:
                session['username'] = result[0]['username']
                return redirect(url_for("prediction", username=session['username']))
            return render_template("login.html", alert_message="Wrong Password, Please try again")
        return render_template("login.html", alert_message="Invalid User")
    return render_template("login.html")


@app.route("/register", methods=["GET", "POST"])
def register():
    if request.method == "POST":
        global user_id
        username = request.form.get("username")
```

```
            email = request.form.get("email")
            password = hash_password(email, request.form.get("password"))
            confirm_password = hash_password(email, request.form.get("confirm_password"))
            if password == confirm_password:
                user_data = Document(username=username, email=email, password=password)
                exist, _ = user_exists(email)
                if exist:
                    return render_template("login.html", alert_message="User already exists, Please Login")
                response = service.put_document(db=os.getenv("USER_DB"), document=user_data,
doc_id=str(user_id))
                if response:
                    user_id += 1
                    result = send_registration_mail(email, username)
                    if not result:
                        result = send_registration_mail(email, username)
                    return render_template("login.html", success_message="Registration Success")
                return render_template("register.html", alert_message="Registration Failure, Please try again")
            return render_template("register.html", alert_message="Passwords does not match")
        return render_template("register.html")


    @app.route("/forgot_password", methods=["GET", "POST"])
    def forgot_password():
        if request.method == "POST":
            email = request.form.get("email")
            pass_code = request.form.get("password_code")
            new_password = request.form.get("new_password")
            confirm_password = request.form.get("confirm_password")
            if new_password:
                new_password = hash_password(email, new_password)
            if confirm_password:
                confirm_password = hash_password(email, confirm_password)
            if not pass_code:
                exist, _ = user_exists(email)
                if exist:
                    original_code = generate_passcode()
                    result = send_forgot_password_mail(email, original_code)
                    if not result:
                        result = send_forgot_password_mail(email, original_code)
                    cache.set(email, original_code)
                    return render_template("forgot_password.html", success_message="Verification Code sent to your
email", email=email)
                return render_template("forgot_password.html", alert_message="Invalid User")
            original_code = cache.get(email)
            cache.delete(email)
            if original_code != pass_code:
                return render_template("forgot_password.html", alert_message="Invalid Verification Code")
            if new_password == confirm_password:
                _, user = user_exists(email)
```

```
                    user_data = Document(id=user[0]["_id"], rev=user[0]["_rev"], username=user[0]["username"],
email=user[0]["email"], password=new_password)
                    response = service.post_document(db=os.getenv("USER_DB"), document=user_data)
                    if response:
                        return render_template("login.html", success_message="Password Changed Successfully")
                    return render_template("forgot_password.html", alert_message="Password Change Failed, Please try
again")
                return render_template("forgot_password.html", alert_message="Passwords does not match")
            return render_template("forgot_password.html")


    @app.route("/prediction")
    def prediction():
        if session.get('username'):
            return render_template("prediction.html", username=session['username'], video_path="main_video_feed")
        return redirect("/login")


    @app.route("/demo_1")
    def demo_1():
        if session.get('username'):
            return render_template("prediction.html", username=session['username'], video_path="sample_1")
        return redirect("/login")


    @app.route("/demo_2")
    def demo_2():
        if session.get('username'):
            return render_template("prediction.html", username=session['username'], video_path="sample_2")
        return redirect("/login")


    @app.route("/demo_3")
    def demo_3():
        if session.get('username'):
            return render_template("prediction.html", username=session['username'], video_path="sample_3")
        return redirect("/login")


    @app.route("/demo_4")
    def demo_4():
        if session.get('username'):
            return render_template("prediction.html", username=session['username'], video_path="sample_4")
        return redirect("/login")


    @app.route("/main_video_feed")
    def main_video_feed():
        return Response(gen_frames(0, 'output.mp4'), mimetype='multipart/x-mixed-replace; boundary=frame')
    @app.route("/sample_1")
    def sample_1():
        return Response(gen_frames("sample_drowning-1.mp4", 'sample_drowning-1-output.mp4'),
mimetype='multipart/x-mixed-replace; boundary=frame')


    @app.route("/sample_2")
    def sample_2():
        return Response(gen_frames("sample_drowning-2.mp4", 'sample_drowning-2-output.mp4'),
mimetype='multipart/x-mixed-replace; boundary=frame')


    @app.route("/sample_3")
    def sample_3():
        return Response(gen_frames("sample_swimming-1.mp4", 'sample_swimming-1-output.mp4'),
mimetype='multipart/x-mixed-replace; boundary=frame')


    @app.route("/sample_4")
    def sample_4():
        return Response(gen_frames("sample_swimming-2.mp4", 'sample_swimming-2-output.mp4'),
mimetype='multipart/x-mixed-replace; boundary=frame')


    @app.route("/demo")
    def demo():
        return render_template("demo.html")


    @app.route("/logout")
    def logout():
        session.pop('username')
        return render_template("logout.html")


    if _name_ == '_main_':
        app.run(host='0.0.0.0')
```

Execution:

Home page:



About Project

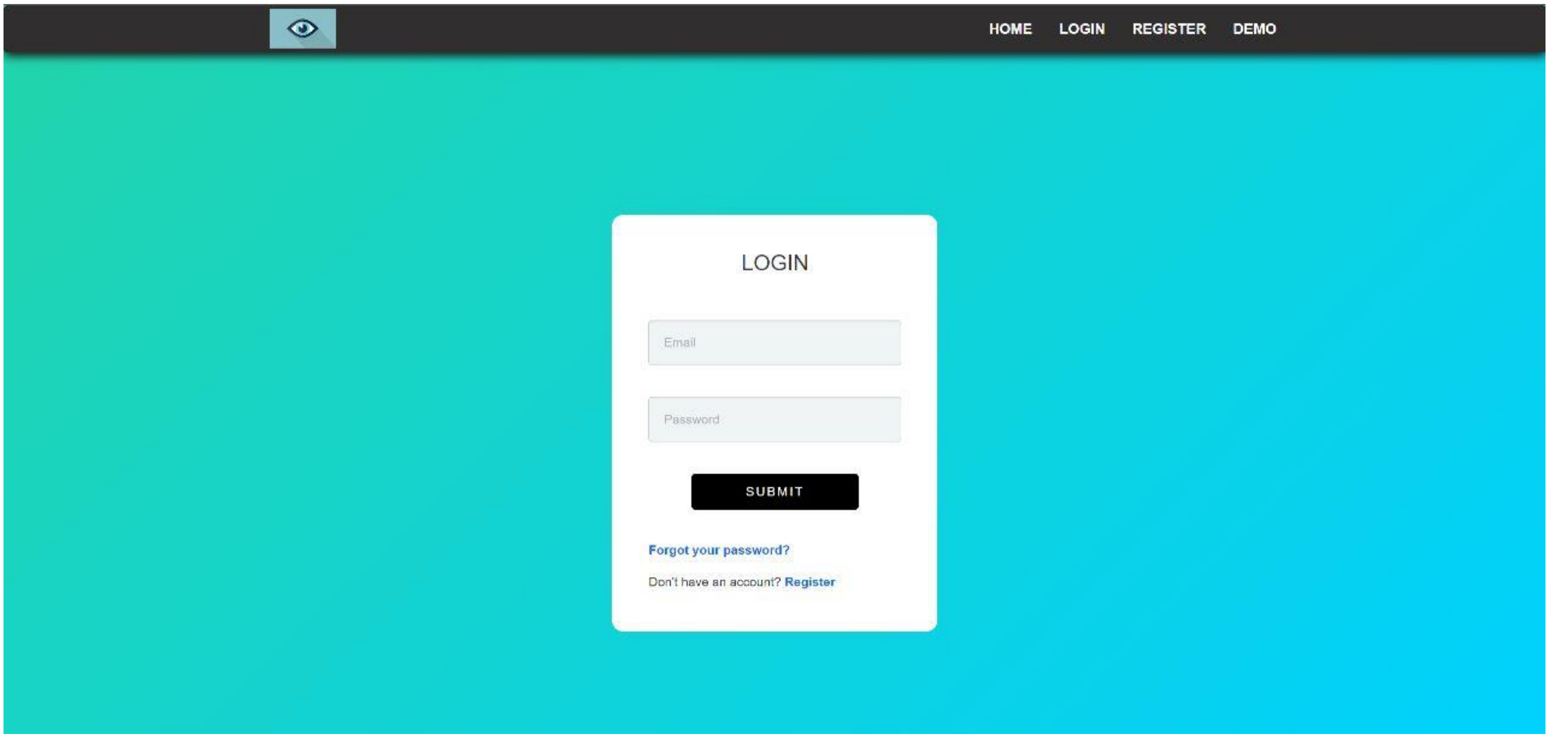
Problem

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

Solution

By studying body movement patterns and connecting cameras to artificial intelligence (AI) systems we can devise a pool safety system

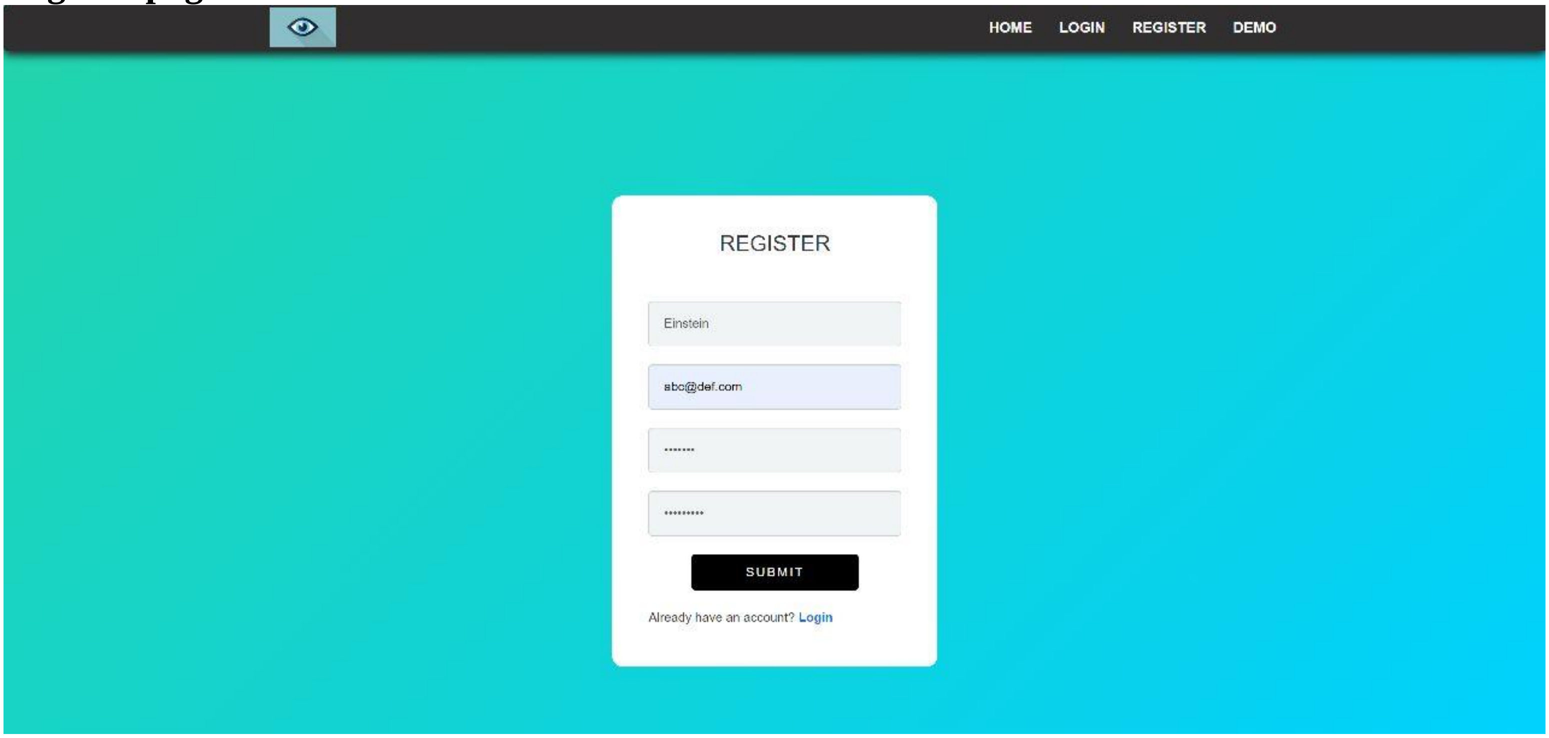
Login Page :



Forgot your password?

Don't have an account? Register

Register page:



Einstein

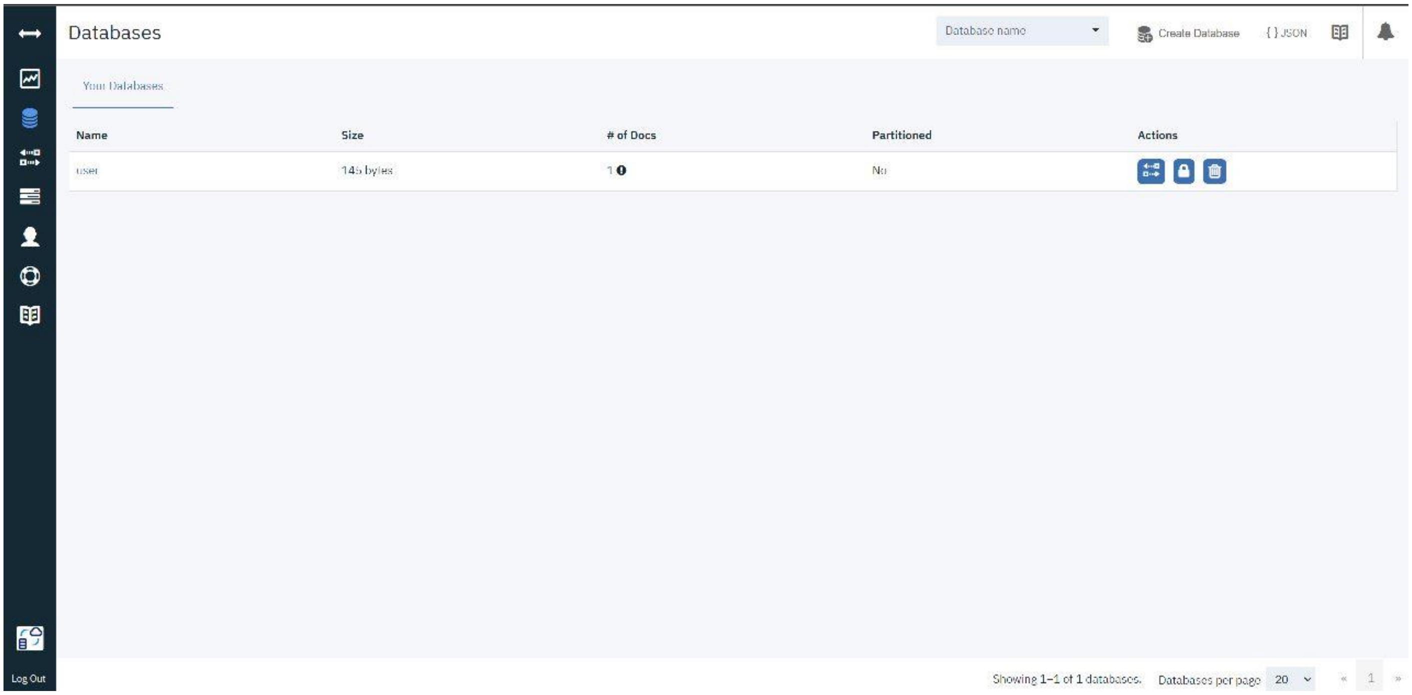
abc@def.com

•••••••

•••••••••

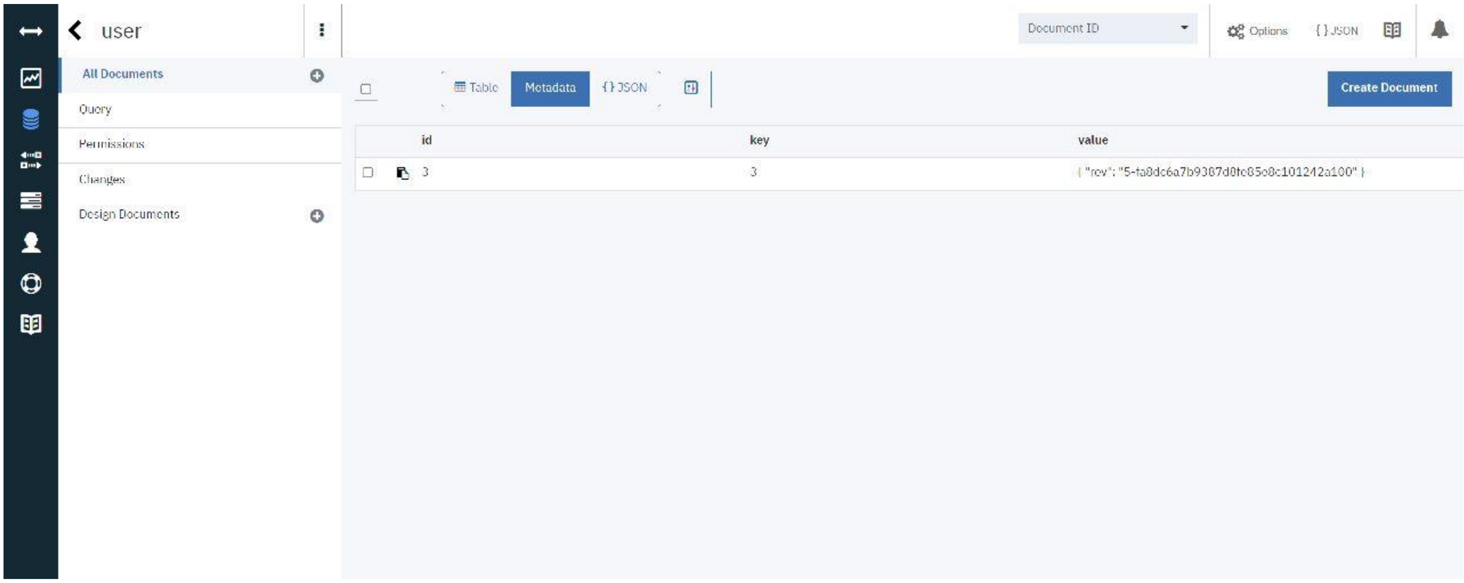
Already have an account? Login

After Register it is stored in Cloud Data Base:



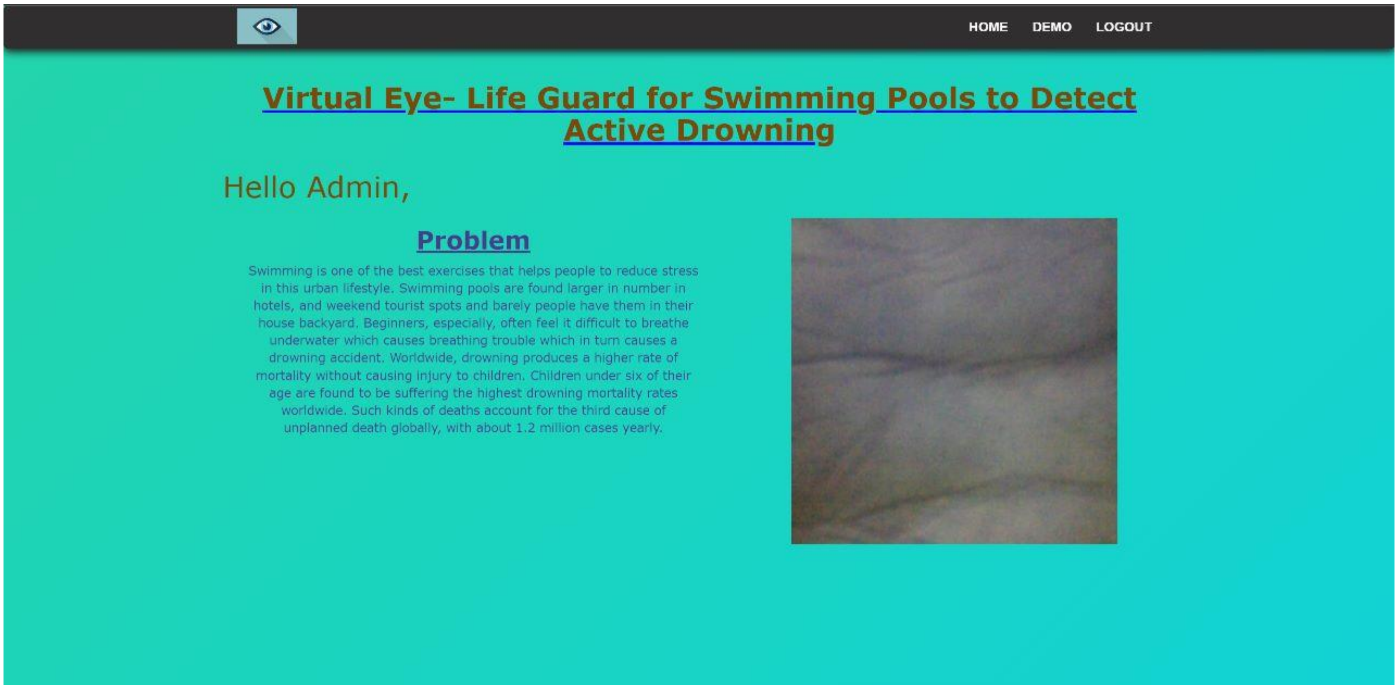
Name	Size	# of Docs	Partitioned	Actions
user	145 bytes	1	No	

Showing 1–1 of 1 databases. Databases per page 20



	id	key	value
□	3	3	{ "rev": "5-fa8dc6a7b9387d0fc85e8c101242a100" }

Detection Page:



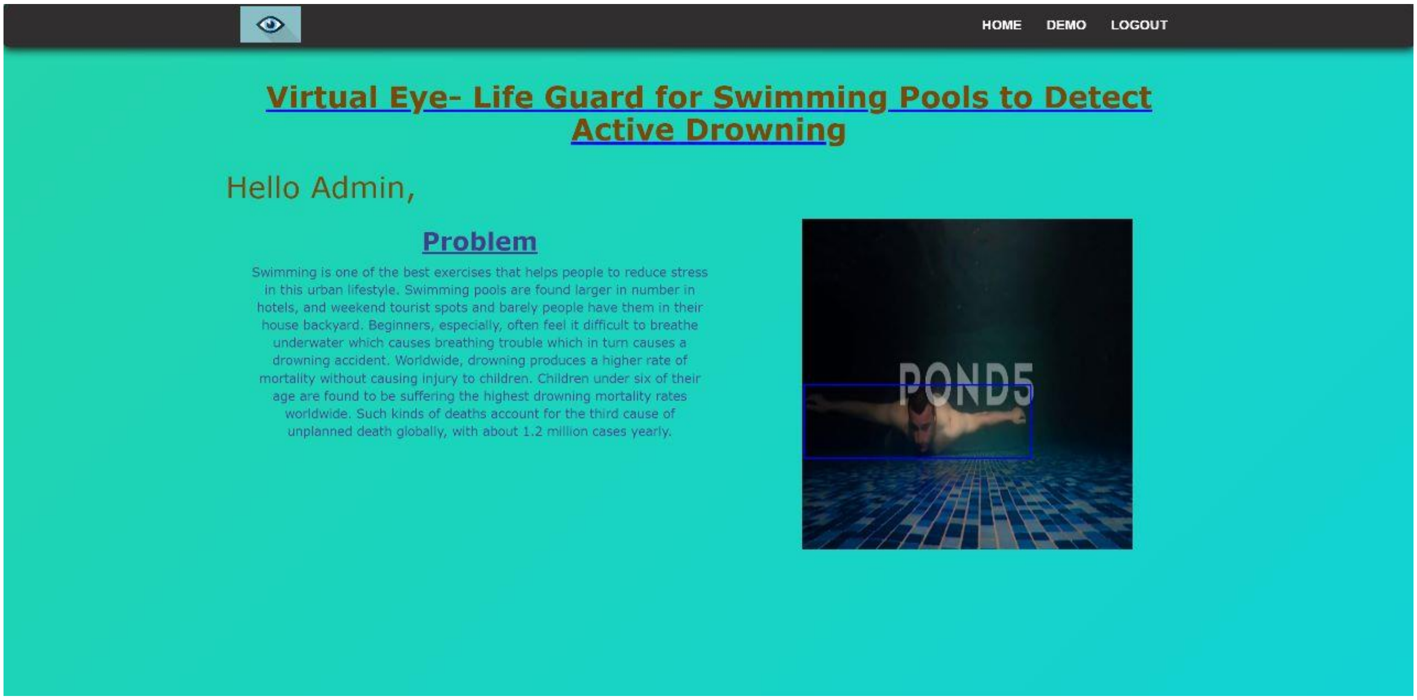
Virtual Eye- Life Guard for Swimming Pools to Detect Active Drowning

Hello Admin,

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 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.

Before Drowning :



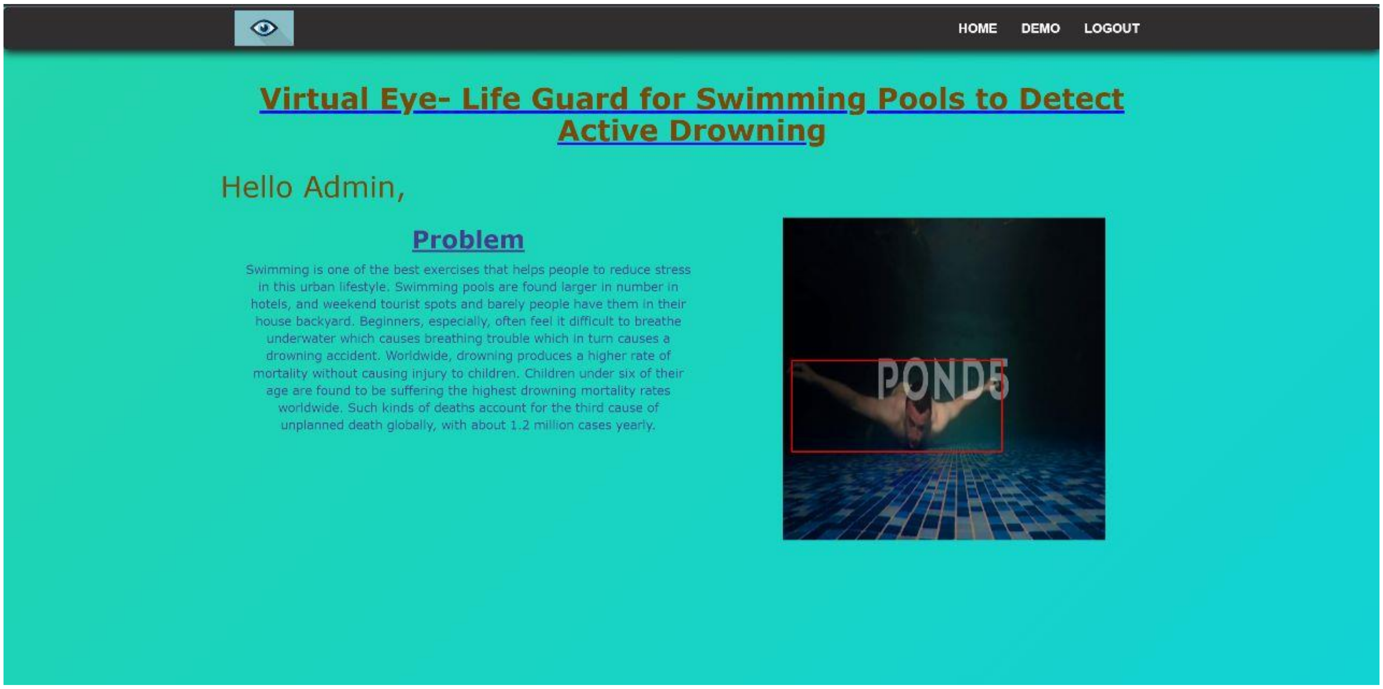
Virtual Eye- Life Guard for Swimming Pools to Detect Active Drowning

Hello Admin,

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 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.

After Drowning:



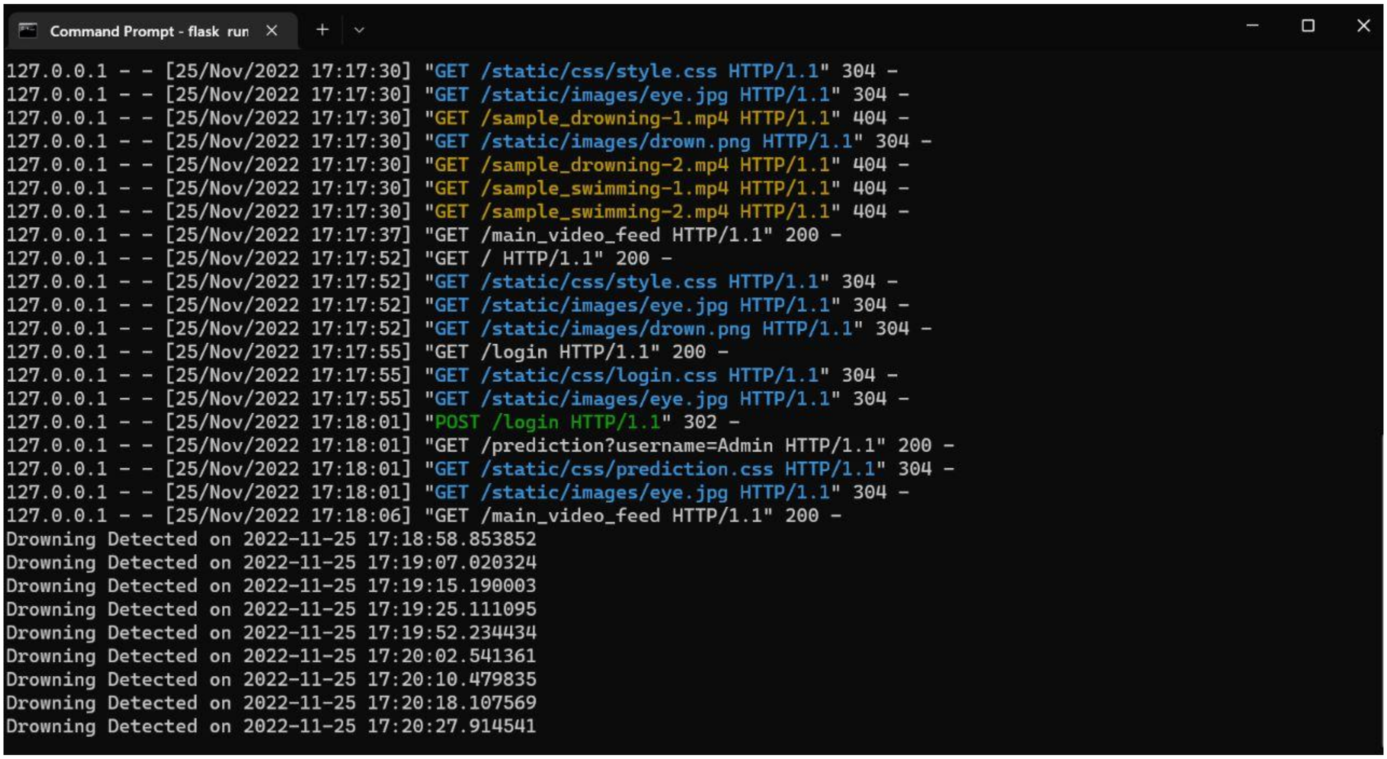
Virtual Eye- Life Guard for Swimming Pools to Detect Active Drowning

Hello Admin,

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 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.

Result :

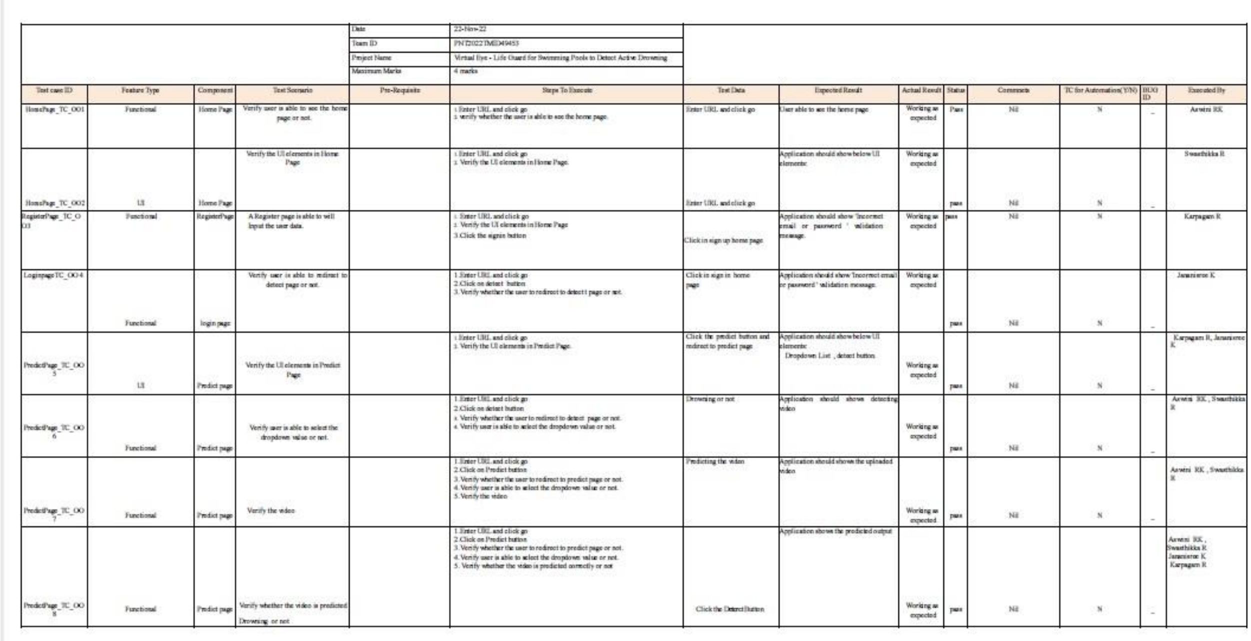


```
127.0.0.1 - - [25/Nov/2022 17:17:30] "GET /static/css/style.css HTTP/1.1" 304 -
127.0.0.1 - - [25/Nov/2022 17:17:30] "GET /static/images/eye.jpg HTTP/1.1" 304 -
127.0.0.1 - - [25/Nov/2022 17:17:30] "GET /sample_drowning-1.mp4 HTTP/1.1" 404 -
127.0.0.1 - - [25/Nov/2022 17:17:30] "GET /static/images/drown.png HTTP/1.1" 304 -
127.0.0.1 - - [25/Nov/2022 17:17:30] "GET /sample_drowning-2.mp4 HTTP/1.1" 404 -
127.0.0.1 - - [25/Nov/2022 17:17:30] "GET /sample_swimming-1.mp4 HTTP/1.1" 404 -
127.0.0.1 - - [25/Nov/2022 17:17:30] "GET /sample_swimming-2.mp4 HTTP/1.1" 404 -
127.0.0.1 - - [25/Nov/2022 17:17:37] "GET /main_video_feed HTTP/1.1" 200 -
127.0.0.1 - - [25/Nov/2022 17:17:52] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [25/Nov/2022 17:17:52] "GET /static/css/style.css HTTP/1.1" 304 -
127.0.0.1 - - [25/Nov/2022 17:17:52] "GET /static/images/eye.jpg HTTP/1.1" 304 -
127.0.0.1 - - [25/Nov/2022 17:17:52] "GET /static/images/drown.png HTTP/1.1" 304 -
127.0.0.1 - - [25/Nov/2022 17:17:55] "GET /login HTTP/1.1" 200 -
127.0.0.1 - - [25/Nov/2022 17:17:55] "GET /static/css/login.css HTTP/1.1" 304 -
127.0.0.1 - - [25/Nov/2022 17:17:55] "GET /static/images/eye.jpg HTTP/1.1" 304 -
127.0.0.1 - - [25/Nov/2022 17:18:01] "POST /login HTTP/1.1" 302 -
127.0.0.1 - - [25/Nov/2022 17:18:01] "GET /prediction?username=Admin HTTP/1.1" 200 -
127.0.0.1 - - [25/Nov/2022 17:18:01] "GET /static/css/prediction.css HTTP/1.1" 304 -
127.0.0.1 - - [25/Nov/2022 17:18:01] "GET /static/images/eye.jpg HTTP/1.1" 304 -
127.0.0.1 - - [25/Nov/2022 17:18:06] "GET /main_video_feed HTTP/1.1" 200 -
Drowning Detected on 2022-11-25 17:18:58.853852
Drowning Detected on 2022-11-25 17:19:07.020324
Drowning Detected on 2022-11-25 17:19:15.190003
Drowning Detected on 2022-11-25 17:19:25.111095
Drowning Detected on 2022-11-25 17:19:52.234434
Drowning Detected on 2022-11-25 17:20:02.541361
Drowning Detected on 2022-11-25 17:20:10.479835
Drowning Detected on 2022-11-25 17:20:18.107569
Drowning Detected on 2022-11-25 17:20:27.914541
```

CHAPTER-8

TESTING

8.1 Test cases



Date	22-Nov-22
Team ID	PNT2022TMID49453
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-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	BUG ID	Executed By
HomePage_TC_OO1	Functional	Home Page	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	Nil	N	_	Aswini RK
HomePage_TC_OO2	UI	Home Page	Verify the UI elements in Home Page		1.Enter URL and click go 2. Verify the UI elements in Home Page.	Enter URL and click go	Application should show below UI elements:	Working as expected	pass	Nil	N	_	Swasthikka R
RegisterPage_TC_OO3	Functional	RegisterPage	A Register page is able to will Input the user data.		1. Enter URL and click go 2. Verify the UI elements in Home Page 3.Click the signin button	Click in sign up home page	Application should show 'Incorrect email or password ' validation message.	Working as expected	pass	Nil	N		Karpagam R
LoginpageTC_OO4	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 to redirect to detect page or not.	Click in sign in home page	Application should show 'Incorrect email or password' validation message.	Working as expected	pass	Nil	N	_	Janansree K
PredictPage_TC_OO5	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	Nil	N	_	Karpagam R, Janansree K
PredictPage_TC_OO6	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 to redirect to detect page or not. 4. Verify user is able to select the dropdown value or not.	Drowning or not	Application should shows detecting video	Working as expected	pass	Nil	N	_	Aswini RK , Swasthikka R
PredictPage_TC_OO7	Functional	Predict page	Verify the video		1.Enter URL and click go 2.Click on Predict button 3. Verify whether the user to redirect to predict page or not. 4. Verify user is able to select the dropdown value or not. 5. Verify the video	Predicting the video	Application should shows the uploaded video	Working as expected	pass	Nil	N	_	Aswini RK , Swasthikka R
PredictPage_TC_OO8	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 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	Nil	N	_	Aswini RK , Swasthikka R Janansree K Karpagam R

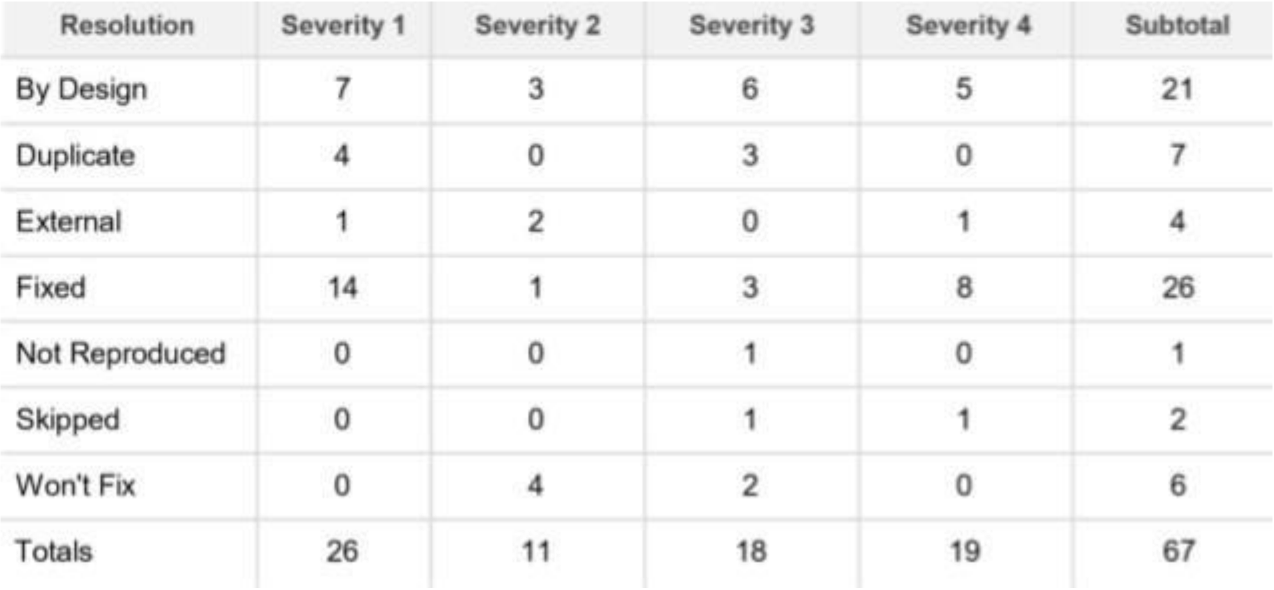
8.2 User Acceptance Testing

- **Purpose of Document**

This report's objective is to succinctly outline the test coverage and outstanding problems for 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**

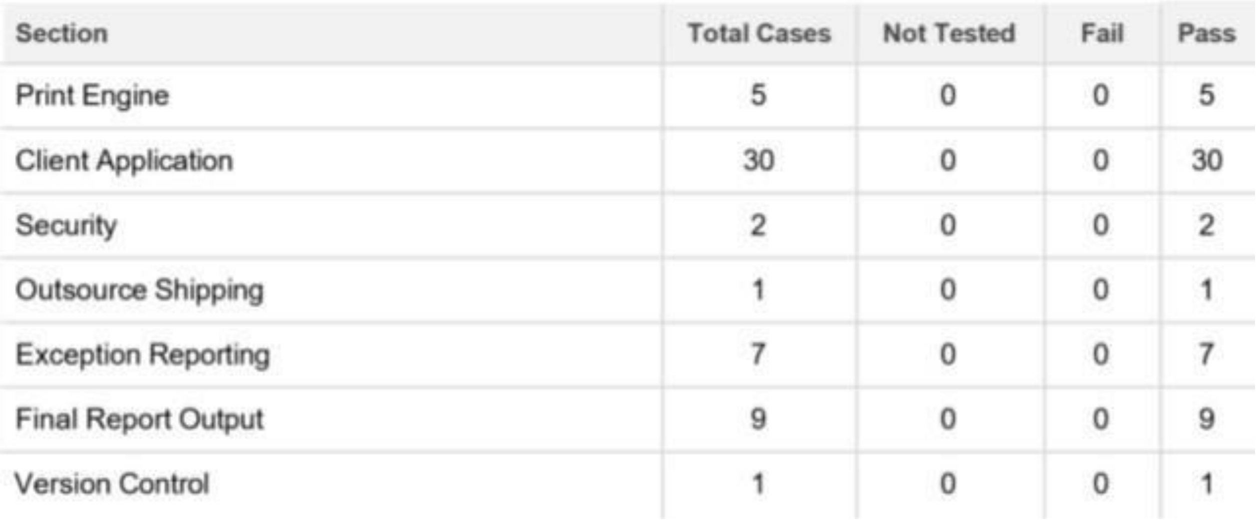
This reports how is the number of resolved or closed bugs at each severity level, and how they were resolved.



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**

Thisreportshowsthenumberoftestcasesthathavepassed,failed,anduntested



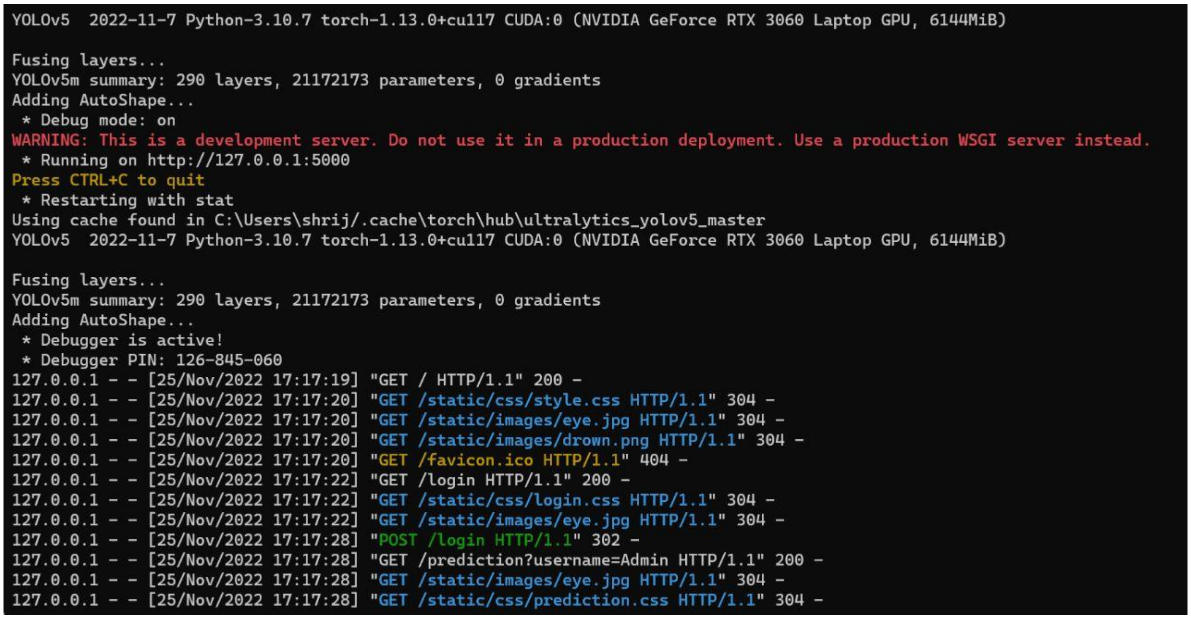
Section	Total Cases	Not Tested	Fail	Pass
Print Engine	5	0	0	5
Client Application	30	0	0	30
Security	2	0	0	2
Outsource Shipping	1	0	0	1
Exception Reporting	7	0	0	7
Final Report Output	9	0	0	9
Version Control	1	0	0	1

CHAPTER-9

RESULT

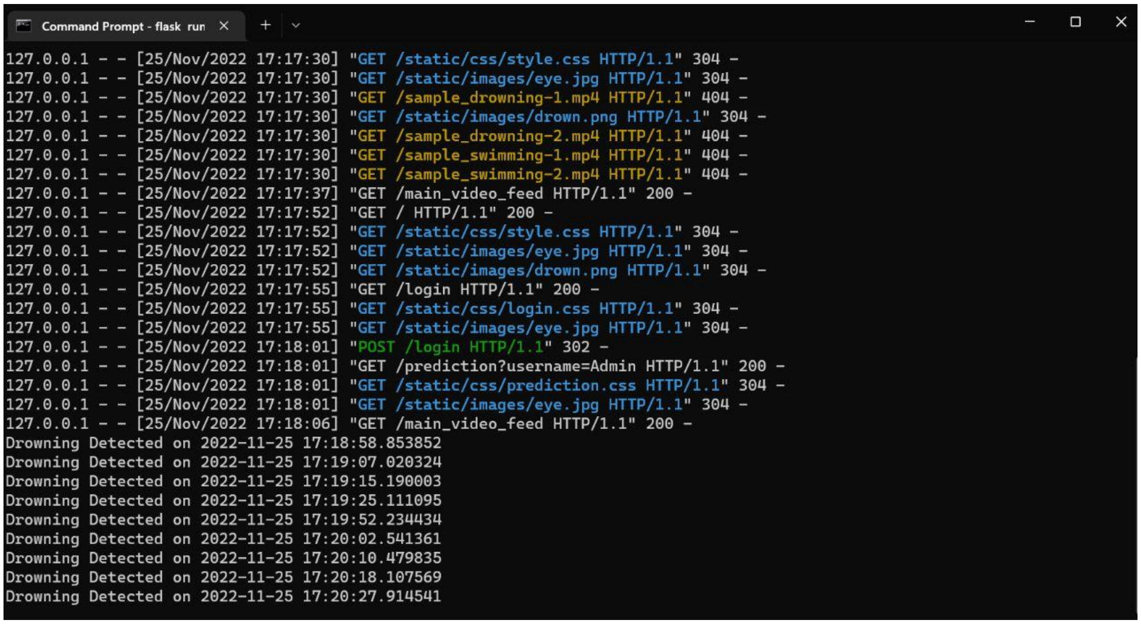
9.1PerformanceMetric

Before drowning:



```
YOLOv5  2022-11-7 Python-3.10.7 torch-1.13.0+cu117 CUDA:0 (NVIDIA GeForce RTX 3060 Laptop GPU, 6144MiB)

Fusing layers...
YOLOv5m summary: 290 layers, 21172173 parameters, 0 gradients
Adding AutoShape...
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
Using cache found in C:\Users\shrij/.cache\torch\hub\ultralytics_yolov5_master
YOLOv5  2022-11-7 Python-3.10.7 torch-1.13.0+cu117 CUDA:0 (NVIDIA GeForce RTX 3060 Laptop GPU, 6144MiB)

Fusing layers...
YOLOv5m summary: 290 layers, 21172173 parameters, 0 gradients
Adding AutoShape...
 * Debugger is active!
 * Debugger PIN: 126-845-060
127.0.0.1 - - [25/Nov/2022 17:17:19] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [25/Nov/2022 17:17:20] "GET /static/css/style.css HTTP/1.1" 304 -
127.0.0.1 - - [25/Nov/2022 17:17:20] "GET /static/images/eye.jpg HTTP/1.1" 304 -
127.0.0.1 - - [25/Nov/2022 17:17:20] "GET /static/images/drown.png HTTP/1.1" 304 -
127.0.0.1 - - [25/Nov/2022 17:17:20] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [25/Nov/2022 17:17:22] "GET /login HTTP/1.1" 200 -
127.0.0.1 - - [25/Nov/2022 17:17:22] "GET /static/css/login.css HTTP/1.1" 304 -
127.0.0.1 - - [25/Nov/2022 17:17:22] "GET /static/images/eye.jpg HTTP/1.1" 304 -
127.0.0.1 - - [25/Nov/2022 17:17:28] "POST /login HTTP/1.1" 302 -
127.0.0.1 - - [25/Nov/2022 17:17:28] "GET /prediction?username=Admin HTTP/1.1" 200 -
127.0.0.1 - - [25/Nov/2022 17:17:28] "GET /static/images/eye.jpg HTTP/1.1" 304 -
127.0.0.1 - - [25/Nov/2022 17:17:28] "GET /static/css/prediction.css HTTP/1.1" 304 -
```

After Drowning:



```
127.0.0.1 - - [25/Nov/2022 17:17:30] "GET /static/css/style.css HTTP/1.1" 304 -
127.0.0.1 - - [25/Nov/2022 17:17:30] "GET /static/images/eye.jpg HTTP/1.1" 304 -
127.0.0.1 - - [25/Nov/2022 17:17:30] "GET /sample_drowning-1.mp4 HTTP/1.1" 404 -
127.0.0.1 - - [25/Nov/2022 17:17:30] "GET /static/images/drown.png HTTP/1.1" 304 -
127.0.0.1 - - [25/Nov/2022 17:17:30] "GET /sample_drowning-2.mp4 HTTP/1.1" 404 -
127.0.0.1 - - [25/Nov/2022 17:17:30] "GET /sample_swimming-1.mp4 HTTP/1.1" 404 -
127.0.0.1 - - [25/Nov/2022 17:17:30] "GET /sample_swimming-2.mp4 HTTP/1.1" 404 -
127.0.0.1 - - [25/Nov/2022 17:17:37] "GET /main_video_feed HTTP/1.1" 200 -
127.0.0.1 - - [25/Nov/2022 17:17:52] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [25/Nov/2022 17:17:52] "GET /static/css/style.css HTTP/1.1" 304 -
127.0.0.1 - - [25/Nov/2022 17:17:52] "GET /static/images/eye.jpg HTTP/1.1" 304 -
127.0.0.1 - - [25/Nov/2022 17:17:52] "GET /static/images/drown.png HTTP/1.1" 304 -
127.0.0.1 - - [25/Nov/2022 17:17:55] "GET /login HTTP/1.1" 200 -
127.0.0.1 - - [25/Nov/2022 17:17:55] "GET /static/css/login.css HTTP/1.1" 304 -
127.0.0.1 - - [25/Nov/2022 17:17:55] "GET /static/images/eye.jpg HTTP/1.1" 304 -
127.0.0.1 - - [25/Nov/2022 17:18:01] "POST /login HTTP/1.1" 302 -
127.0.0.1 - - [25/Nov/2022 17:18:01] "GET /prediction?username=Admin HTTP/1.1" 200 -
127.0.0.1 - - [25/Nov/2022 17:18:01] "GET /static/css/prediction.css HTTP/1.1" 304 -
127.0.0.1 - - [25/Nov/2022 17:18:01] "GET /static/images/eye.jpg HTTP/1.1" 304 -
127.0.0.1 - - [25/Nov/2022 17:18:06] "GET /main_video_feed HTTP/1.1" 200 -
Drowning Detected on 2022-11-25 17:18:58.853852
Drowning Detected on 2022-11-25 17:19:07.020324
Drowning Detected on 2022-11-25 17:19:15.190003
Drowning Detected on 2022-11-25 17:19:25.111095
Drowning Detected on 2022-11-25 17:19:52.234434
Drowning Detected on 2022-11-25 17:20:02.541361
Drowning Detected on 2022-11-25 17:20:10.479835
Drowning Detected on 2022-11-25 17:20:18.107569
Drowning Detected on 2022-11-25 17:20:27.914541
```

CHAPTER-10

ADVANTAGES & DISADVANTAGES

10.1 ADVANTAGE

- The user feels more safe and at ease
- Children, adults, pets, and senior citizens are all involved.
- More family time and flexibility for the security personnel stationed around the swimming pool
- Drowning should be closely watched

10.1 DISADVANTAGE:

- Because of this technology, the uneducated will suffer.
- There must always be a network connection.

CHAPTER-11

CONCLUSION

In this paper, we suggested a technique for effective drowning detection. With the use of the Yolo V5 model, we have been able to identify individuals and their state of drowning. If an individual remains still for 10 seconds or moves slowly, an alarm is transmitted to the lifeguard. For potential future use, this system may be substantially expanded.

CHAPTER-12

FUTURESCOPE

The Yolo v5 model has been used to implement the project in the present project that can prevent the swimmer from drowning. So, if someone is drowning, the lifeguard will receive a warning and can save the swimmer. Future updates and additions to this project are possible. Pulse rate detection can be used to update this project. to give the lifeguard a chance to save the swimmer from drowning. We can implement cutting-edge technologies in this project to make it simpler for the lifeguard to save the swimmer's life early.

CHAPTER-13

APPENDIX

Github link: https://github.com/IBM-EPBL/IBM-Project-13952-1659536607

Demo Link: https://drive.google.com/file/d/19mG2z-GZ19y7N5Ql-E4p86M1qMjrMwoL/view