VIRTUAL EYE LIFEGUARD FOR SWIMMING POOL TO DETECT ACTIVE DROWNING SYSTEM

TEAM ID: PNT2022TMID5075

1. INTRODUCTION

When we started looking at the statistics about drowning deaths, we noticed that drowning is the second leading cause of death especially for children under 15. Even if they are under the supervision of guardian but a slightest ignorance can result into disaster.1.2 million people around the world die or get injured by drowning every year, i.e more than 2 people per minute. Detection and fast response time are most important factor to save lives from drowning and avoid permanent damage. Thus our project would be very useful in the school swimming pools for the kids who are not so used to of swimming. It would be very useful in saving the lives of kids. It also helps in places where even adult drowning risk is high. Thus our project would be very useful in the school swimming pools for the kids who are not so used to of swimming. It would be very useful in saving the lives of kids. It also helps in places where even adult drowning risk is high.

1.1Project Overview

Using YOLO object detection, this program will detect whether a person is drowning or not. This software can be used with a Raspberry Pi Camera, which can then be placed underwater with an appropriate case.

The algorithm consists of three core steps: pre-processing, objects-detection and drowning detection.

1.2Purpose

The term 'drowning detection system' (DDS) is used to describe various electronic systems that are designed to assist with the surveillance of swimmers within the water of a swimming pool. Detection and fast response time are most important factor to save lives from drowning and avoid permanent damage.

2. LITERATURE SURVEY

A literature review is a comprehensive summary of previous research on a topic. The literature review surveys scholarly articles, books, and other sources relevant to a particular area of research. The review should enumerate, describe, summarize, objectively evaluate and clarify this previous research.

2.1 Existing Problem

A system to prevent accidental drowning during bathing using the processing of information acquired by a Intel RealSense D415 depth sensor. It gives the false positive during drowning. Most of the problem in the detection and monitoring of swimming players are with occlusion, scale changes, changes of appearance. These problem can be overcome using the proposed YOLO algorithm along with CNN model.

LITERATURE SURVEY

Table 1: Literature Survey

Sl. No.	Title	Author & Publications	Year	Description
1.	Development of a Bathing Accident Monitoring System Using a Depth Sensor	Yoshiaki Endo, Chinthaka Premachandra & *IEEE Sensors Letters*.	2017	Intel RealSense D415 depth sensor is used. It measures distance by the active stereo method.

2.2 References

[1] Yoshiaki Endo , Chinthaka Premachandra , “Development of a Bathing Accident Monitoring System Using a Depth Sensor”, *IEEE Sensors Letters* ,Vol. 6, Issue 2,February 2017.

[2] Seiji Matsuguma, Akihiro Kajiwara, “Bathroom accident detection With 79GHz-bandmillimeter wave sensor’’, *Proc. of 2019 IEEE Instrumentation and Measurement Society*, March 2019.

[3] Goncalo Simoes, Carolina Dionisio, Andre Gloria, Pedro Sebastiao, Nuno Souro, "Smart System for Monitoring and Control of Swimming Pools”, *Proc. of IEEE*, April 2019.

[4] Samuel Ndueso John, Ukpabio Imelda Godswill, Omoruyi Osemwegie, Godfrey Onyiagha, Etinosa Noma-Osaghae and Kennedy Okopujie, “DESIGN OF A DROWNING RESCUE ALERT SYSTEM", *International Journal of Mechanical Engineering and Technology (IJMET)*,Vol.10,Issues.01,pp.1987-1995,January2019.

[5] Upulie Handalage, Nisansali Nikapotha, Chanaka Subasinghe, Tereen Prasanga, Thusithanjana Thilakarthna, Dharshana Kasthurirathna, "Computer Vision Enabled Drowning Detection Systems", *Proc. of 3rd International Conference on Advancements in Computing(ICAC)*, December 2021.

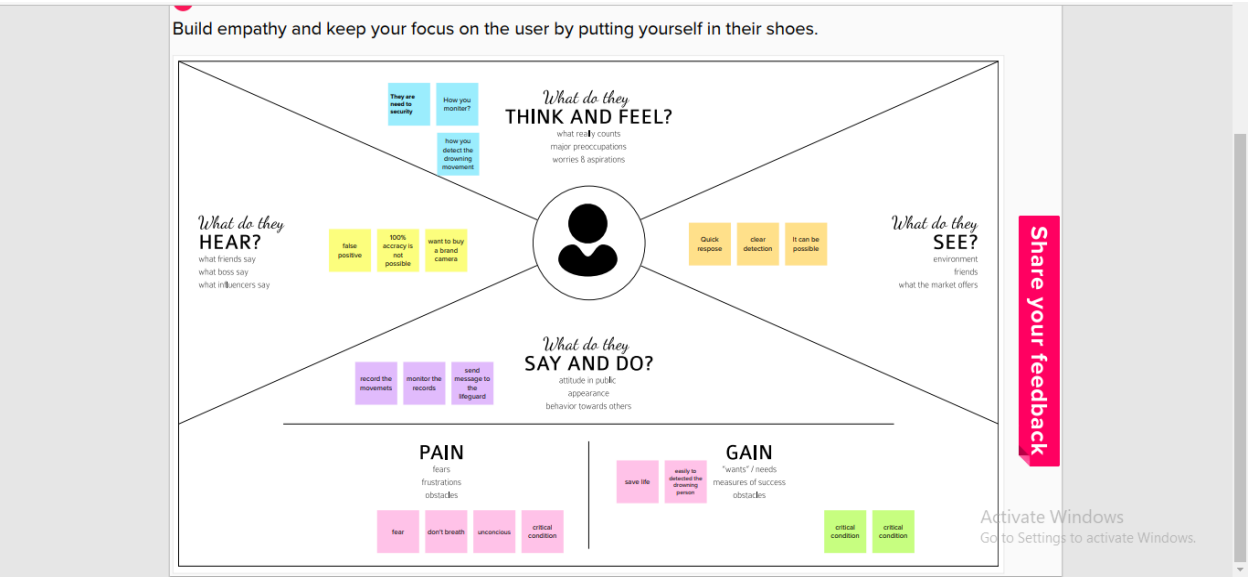
2.3 PROBLEM STATEMENT DEFINITION

1.Virtual eye - life guard for swimming pools to detect active drowning. Swimming is one of the best exercises that helps people to reduce stress in this urban lifestyle. Beginners, especially, often feel it difficult to breathe underwater which causes breathing trouble which in turn causes a drowning accident.

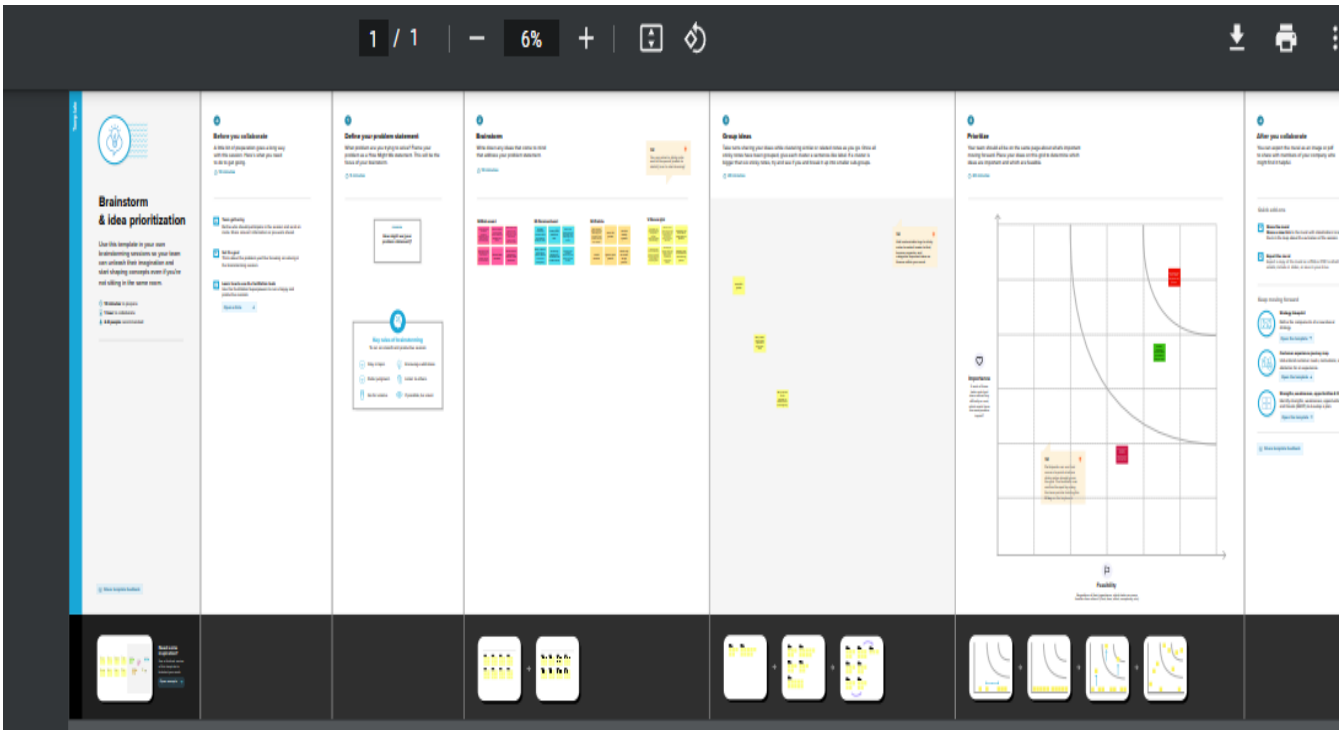
2. The drowning detection system for the prevention of drowning incidents in swimming pools. It works like an “extra lifeguard” under the water of swimming pools This detection system tracks the movements of everything inside the water bodies and will help to guard the lives by finding them easily.

3. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



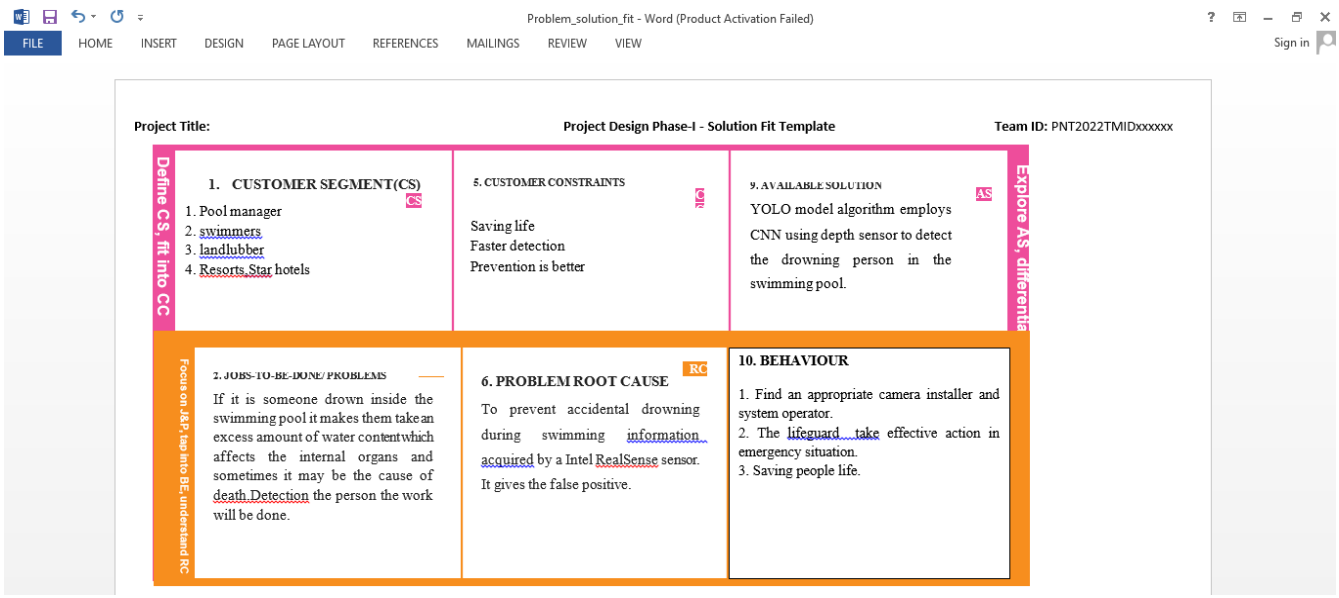
3.2 IDEATION & BRAINSTROMING

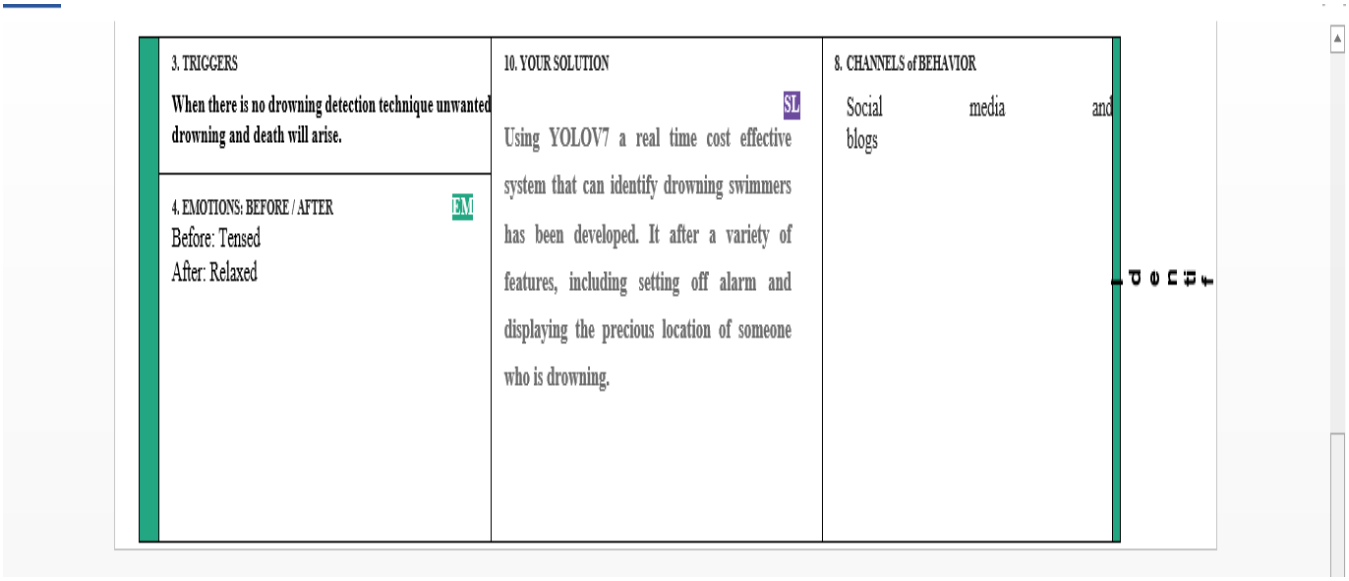


3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
•	Problem Statement (Problem to be solved)	This project describes the drowning detection system for the prevention of drowning incidents in swimming pools. The problem boundary clearly distinguishes between the positive samples which are inside the boundary to those that are less relevant and outside the boundary. It works like an “extra lifeguard” under the water of swimming pools. For instance, If it happens to someone to drown inside the swimming pool, it makes them take an excess amount of water content which affects the internal organs and sometimes it may be the cause of death. This detection system tracks the movements of everything inside the water bodies and will help to guard the lives by finding them easily.
•	Idea / Solution description	In order to detect the Victim we are converting video into image.Yolo Model Algorithm employs Convoloutional Nerural Network to detect the drowning person in the swimming pool.This module used to alert the lifeguard by using the triggered alarm. If any person not drowning in the pool not use the alarm to notify to the safeguard.
•	Novelty / Uniqueness	Availability of better dataset, modern methodologies, and technologies with high computational power accompanied by highquality surveillance cameras, will help to improve the accuracy of drowning detection & even can be used in adverse conditions.
•	Social Impact / Customer Satisfaction	In case of an incident it is possible to extract and store not only the video but also pulse rate of a victim so it will be useful to indentify the reason behind his/her drowness.
•	Business Model (Revenue Model)	After the implementation of all these essentials, this system also can be used on sea beaches for drowning detection.
•	Scalability of the Solution	Using Image processing technique in achieving expected results. Peak Signal to Noise Ratio (PSNR), Mean Square Error(MSE) and Correlation Index(CI) its depends upon achieve result.

3.4 PROBLEM SOLUTION FIT





4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Installation	Needed to be fixed under the water of swimming pool, without creating any disturbance (such as water bodies, people, other objects).
FR-2	Detection	To detect the person in unconscious state.
FR-3	Support	Use a YOLO algorithm and database it required for to detect the person and take the help of lifeguard.
FR-4	Audio	Shout for help or keep calm if the person is unconscious.
FR-5	Alert	Set alarm if any person drowning inside the swimming pool to send the message for lifeguard to save the victim life.

4.2Non-functional Requirements:

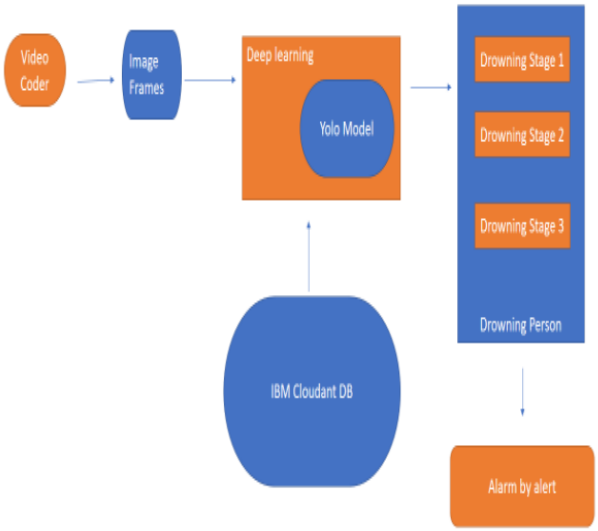
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The lifeguard should be present all the time in the swimming pool.
NFR-2	**Security**	Lifeguard should be awareness of the alert message to save the life of the victim.
NFR-3	**Reliability**	To shout the alarm and send the message helping to avoid panic even in critical situations.
NFR-4	**Performance**	The alarm is triggered when the swimmer is detected as drowning situations.
NFR-5	**Availability**	Equipment and accessories include lifesaver rings, skimmer, telescopic handle, triggered alarm, first aid kit, spine boards, rescue tubes.
NFR-6	**Scalability**	Virtual eye lifeguard detects the potential drownings and promptly notifies you. It features the latest artificial intelligence technology.

5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS

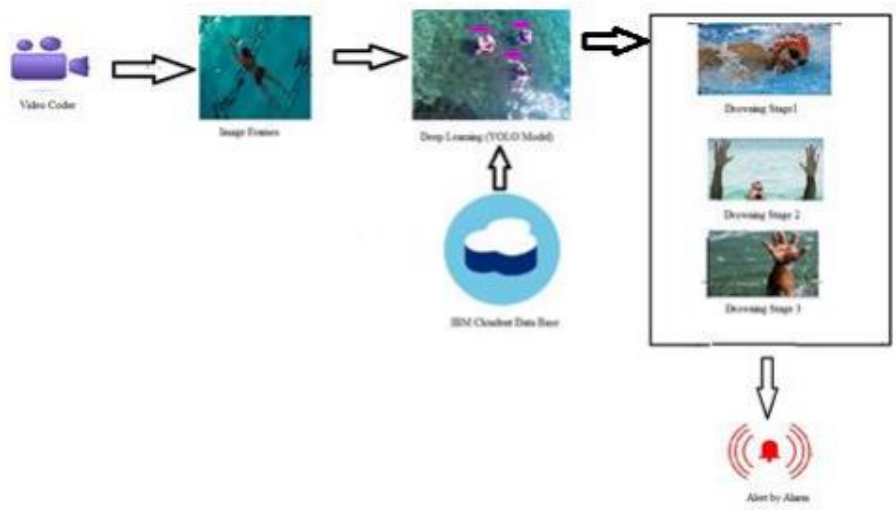
Data Flow Diagrams: A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



5.2 SOLUTION & TECHNICAL ARCHITECTURE

To find underwater movement while person in drowning they have and Problem or anything else we will find the solution using the Artificial Intelligence (AI) detection technology.

Usually, such systems can be developed by installing more than 16cameras underwater and ceiling and analyzing the video feedsto detect any anomalies. AS a POC we make use of one camera that streams thevideo underwater and analyses the positionof swimmers to assess theprobability of drowning, if higher then an alert will be generated to attract lifeguards' attention.



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	Install the camera inside the underwater, connect necessary app in the phone or other device	I can cameras to the IBM cloud DB	High	Sprint-1
Customer (Lifeguard	Secure thepeople	USN-2	As a user, I can secure the drowning personsfrom the pool	I can save the drowning person	High	Sprint-1
Customer (swimmers)	safety	USN-3	As a user, I can swim inside the underwater without fear of the Drowning	I can swim safely	medium	Sprint-2

Customer care (Executive)	Contact	USN-4	As a user, I Can resolve if any problem occurs with any device technically	I can contact the customer care executiveto resolve any issues	Medium	Sprint-3
Administrator	Dashboard	USN-5	Management of the drowning detection systemand database management	I can access the system's logs and any other data instantly	High	Sprint-4

6. PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION

Project Tracker, Velocity & Burndown Chart: (4 Marks)

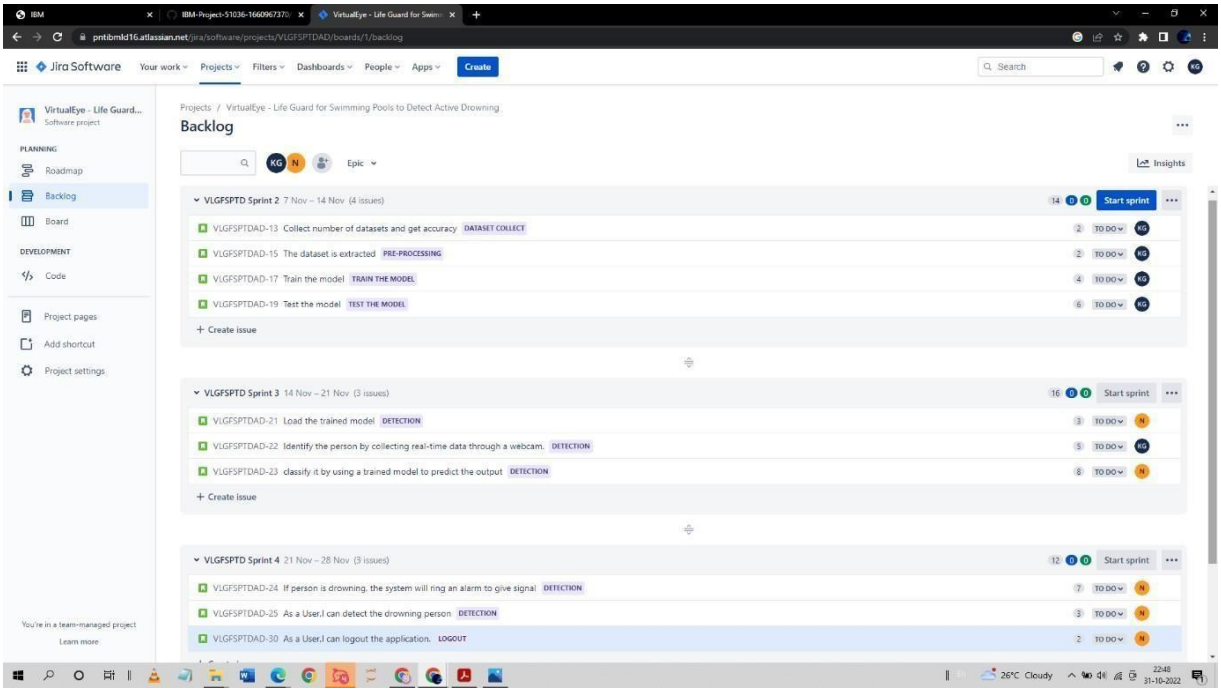
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

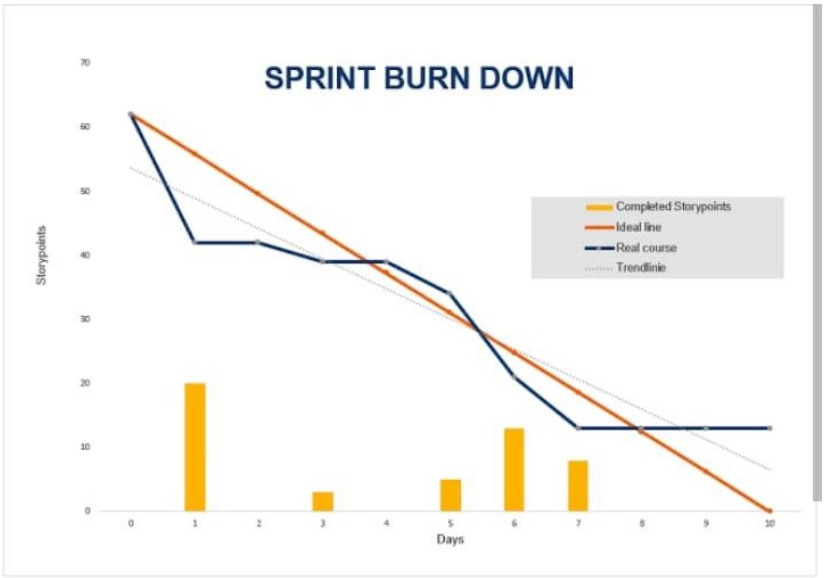
6.2 SPRINT DELIVERY SCHEDULE

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint1	Registration		As a lifeguard , I can register for the application by entering my email, password, and confirming my password.	2	Medium	Maheswari M Parameshwari M Prabha M Sivaranjani V
Sprint1	User Confirmation	USN-2	As a lifeguard, I will receive confirmation email once I have registered for the application	1	Medium	Maheswari M Parameshwari M Prabha M Sivaranjani V
Sprint2	Login		As a lifeguard , I can log into the application by entering email& password	2	High	Maheswari M Parameshwari M Prabha M Sivaranjani V
Sprint3	Cloudant DB		Create DB	2	High	Maheswari M Parameshwari M Prabha M
Sprint4	Application building		As a Lifeguard , It will show the current Information of the swimming pool	1	Medium	Maheswari M Parameshwari M Prabha M Sivaranjani V

6.3 REPORTS FROM JIRA

Backlog (scrum)





7. CODING & SOLUTION

7.1 FEATURE 1

```
[net]
# Testing
# batch=1
# subdivisions=1
# Training
batch=64
subdivisions=16
width=608
height=608
channels=3
momentum=0.9
decay=0.0005
angle=0
saturation = 1.5
exposure = 1.5
hue=.1

learning_rate=0.01
burn_in=1000
max_batches = 500200
policy=steps
steps=400000,450000
scales=.1,.1
```

```
[convolutional]
batch_normalize=1
filters=32
size=3
stride=1
pad=1
activation=leaky

# Downsample

[convolutional]
batch_normalize=1
filters=64
size=3
stride=2
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=32
size=1
stride=1
pad=1
```

activation=leaky

[convolutional]
batch_normalize=1
filters=64
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

Downsample

[convolutional]
batch_normalize=1
filters=128
size=3
stride=2
pad=1
activation=leaky

[convolutional]

batch_normalize=1
filters=64
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=128
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

[convolutional]
batch_normalize=1
filters=64
size=1
stride=1
pad=1

activation=leaky

[convolutional]
batch_normalize=1
filters=128
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

Downsample

[convolutional]
batch_normalize=1
filters=256
size=3
stride=2
pad=1
activation=leaky

[convolutional]

batch_normalize=1
filters=128
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=256
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
pad=1

activation=leaky

[convolutional]
batch_normalize=1
filters=256
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=256

size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=256
size=3
stride=1
pad=1
activation=leaky

```
[shortcut]
from=-3
activation=linear


[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=256
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear
```

```
[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=256
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
```

pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=256
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1

filters=256
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

Downsample

[convolutional]
batch_normalize=1
filters=512
size=3
stride=2
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=256
size=1
stride=1

pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=512
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

[convolutional]
batch_normalize=1
filters=256
size=1
stride=1
pad=1
activation=leaky

[convolutional]

batch_normalize=1
filters=512
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

[convolutional]
batch_normalize=1
filters=256
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=512
size=3
stride=1

pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

[convolutional]
batch_normalize=1
filters=256
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=512
size=3
stride=1
pad=1
activation=leaky

[shortcut]

```
from=-3
activation=linear

[convolutional]
batch_normalize=1
filters=256
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=512
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear


[convolutional]
```

batch_normalize=1
filters=256
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=512
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

[convolutional]
batch_normalize=1
filters=256
size=1
stride=1

pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=512
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

[convolutional]
batch_normalize=1
filters=256
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1

```
filters=512
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

# Downsample

[convolutional]
batch_normalize=1
filters=1024
size=3
stride=2
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=512
size=1
stride=1
```

pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=1024
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

[convolutional]
batch_normalize=1
filters=512
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1

filters=1024
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

[convolutional]
batch_normalize=1
filters=512
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=1024
size=3
stride=1
pad=1
activation=leaky

```
[shortcut]
from=-3
activation=linear

[convolutional]
batch_normalize=1
filters=512
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=1024
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear
```

```
[convolutional]
batch_normalize=1
filters=512
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
size=3
stride=1
pad=1
filters=1024
activation=leaky

[convolutional]
batch_normalize=1
filters=512
size=1
stride=1
pad=1
activation=leaky

[convolutional]
```

batch_normalize=1
size=3
stride=1
pad=1
filters=1024
activation=leaky

[convolutional]
batch_normalize=1
filters=512
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
size=3
stride=1
pad=1
filters=1024
activation=leaky

[convolutional]
size=1

stride=1

pad=1

filters=255

activation=linear

[yolo]

mask = 6,7,8

anchors = 10,13, 16,30, 33,23, 30,61, 62,45, 59,119, 116,90, 156,198, 373,326

classes=80

num=9

jitter=.3

ignore_thresh = .7

truth_thresh = 1

random=1

[route]

layers = -4

[convolutional]

batch_normalize=1

filters=256

size=1

stride=1

pad=1
activation=leaky

[upsample]
stride=2

[route]
layers = -1, 61

[convolutional]
batch_normalize=1
filters=256
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
size=3
stride=1
pad=1
filters=512

activation=leaky

[convolutional]
batch_normalize=1
filters=256
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
size=3
stride=1
pad=1
filters=512
activation=leaky

[convolutional]
batch_normalize=1
filters=256
size=1
stride=1
pad=1
activation=leaky

[convolutional]

batch_normalize=1

size=3

stride=1

pad=1

filters=512

activation=leaky

[convolutional]

size=1

stride=1

pad=1

filters=255

activation=linear

[yolo]

mask = 3,4,5

anchors = 10,13, 16,30, 33,23, 30,61, 62,45, 59,119, 116,90, 156,198, 373,326

classes=80

num=9

jitter=.3

ignore_thresh = .7

truth_thresh = 1

random=1

[route]
layers = -4

[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
pad=1
activation=leaky

[upsample]
stride=2

[route]
layers = -1, 36

[convolutional]
batch_normalize=1

filters=128
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
size=3
stride=1
pad=1
filters=256
activation=leaky

[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
size=3

stride=1
pad=1
filters=256
activation=leaky

[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
size=3
stride=1
pad=1
filters=256
activation=leaky

[convolutional]
size=1
stride=1
pad=1

filters=255

activation=linear

[yolo]

mask = 0,1,2

anchors = 10,13, 16,30, 33,23, 30,61, 62,45, 59,119, 116,90, 156,198, 373,326

classes=80

num=9

jitter=.3

ignore_thresh = .7

truth_thresh = 1

random=1

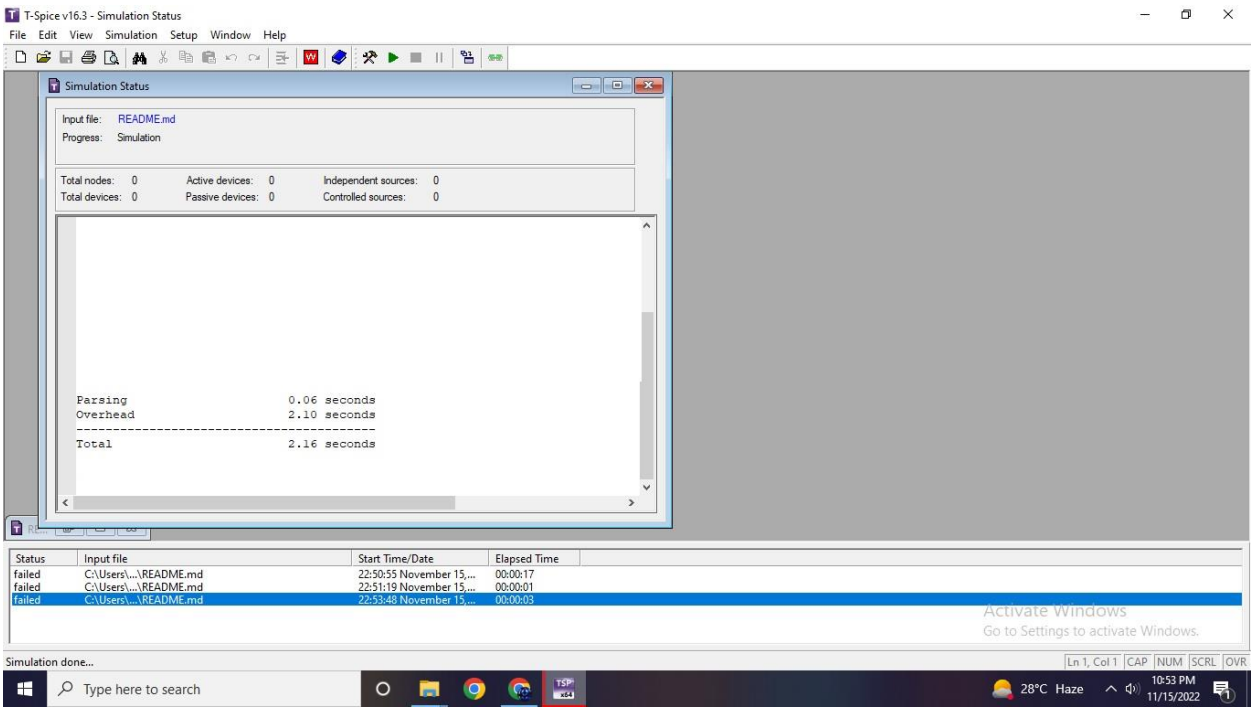
7.2 FEATURE 2

Drown Detect

```
import cvlib as cv
from cvlib.object_detection import draw_bbox
import cv2
import time
import numpy as np
# from picamera Import PiCamera
# camera = PiCamera
# camera.start_preview()
# open webcam
webcam = cv2.VideoCapture(0)
if not webcam.isOpened():
 print("Could not open webcam")
 exit()
t0 = time.time() # gives time in seconds after 1970
# variable dcount stands for how many seconds the person has been standing still for
centre0 = np.zeros(2)
isDrowning = False
# this loop happens approximately every 1 second, so if a person doesn't move,
# or moves very little for 10seconds, we can say they are drowning
# loop through frames
while webcam.isOpened():
 # read frame from webcam
```

```
status, frame = webcam.read()
if not status:
print("Could not read frame")
exit()
# apply object detection
bbox, label, conf = cv.detect_common_objects(frame)
# simplifying for only 1 person
# s = (len(bbox), 2)
if (len(bbox) > 0):
bbox0 = bbox[0]
# centre = np.zeros(s)
centre = [0, 0]
# for i in range(0, len(bbox)):
# centre[i] =[(bbox[i][0]+bbox[i][2])/2,(bbox[i][1]+bbox[i][3])/2 ]
centre = [(bbox0[0] + bbox0[2]) / 2, (bbox0[1] + bbox0[3]) / 2]
# make vertical and horizontal movement variable
hmov = abs(centre[0] - centre0[0])
vmov = abs(centre[1] - centre0[1])
# there is still need to tweek the threshold
# this threshold is for checking how much the centre has moved
x = time.time()
threshold = 10
if (hmov > threshold or vmov > threshold):
print(x - t0, 's')
```

```
 t0 = time.time()
 isDrowning = False
 else:
 print(x - t0, 's')
 if ((time.time() - t0) > 10):
 isDrowning = True
 # print('bounding box: ', bbox, 'label: ' label ,'confidence: ' conf[0], 'centre: ', centre)
 # print(bbox,label ,conf, centre)
 print('bbox: ', bbox, 'centre:', centre, 'centre0:', centre0)
 print('Is he drowning: ', isDrowning)
 centre0 = centre
 # draw bounding box over detected objects
 out = draw_bbox(frame, bbox, label, conf, isDrowning)
 # print('Seconds since last epoch: ', time.time()-t0)
 # display output
 cv2.imshow("Real-time object detection", out)
 # press "Q" to stop
 if cv2.waitKey(1) & 0xFF == ord('q'):
 break
# release resources
webcam.release()
cv2.destroyAllWindows()
```


T-Spice v16.3 - Simulation Status
File Edit View Simulation Setup Window Help
Simulation Status
Input file: README.md
Progress: Simulation
Total nodes: 0
Total devices: 0
Active devices: 0
Passive devices: 0
Independent sources: 0
Controlled sources: 0
Parsing 0.06 seconds
Overhead 2.10 seconds
Total 2.16 seconds
Status
Input file
Start Time/Date
Elapsed Time
failed C:\Users\...\README.md 22:50:55 November 15,... 00:00:17
failed C:\Users\...\README.md 22:51:19 November 15,... 00:00:01
failed C:\Users\...\README.md 22:53:48 November 15,... 00:00:03
Activate Windows
Go to Settings to activate Windows.
Simulation done...
Ln 1, Col 1 CAP NUM SCRL OVR
Type here to search
28°C Haze
10:53 PM
11/15/2022

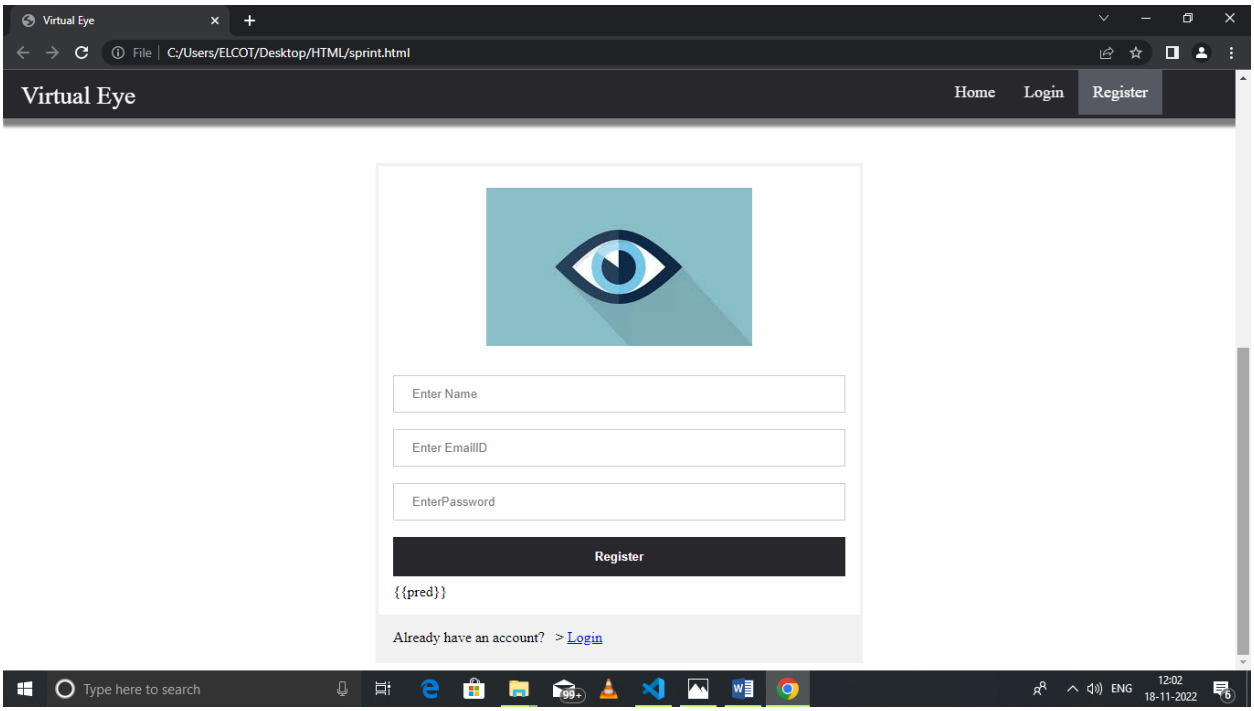
8. TESTING

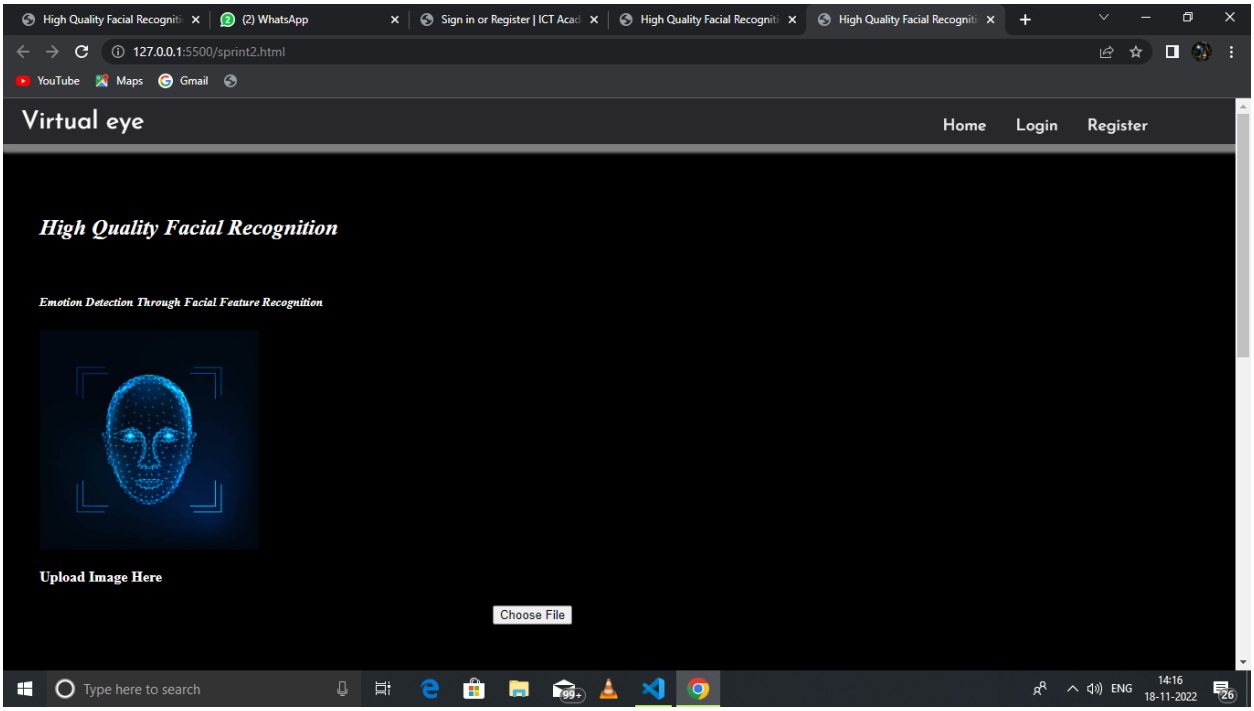
8.1 TEST CAUSES

Test case ID	Feature Type		Test Scenario	Steps TO Execute	Test	Expected Result	Actual Result
LoginPage_TC_OO1	Functional	Home Page	Verify user is able to see the Login/Signup popup when user clicked on My account button	I.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Singup popup displayed or not	Login.html	Login/Signup popup should display	Working as
LoginPage_TC_002		Home Page	Verify the UI elements in Login/Signup popup	I.Enter tJRL and dick go 2.Click on My Account dropdown 3.Verify login/Singup popup with below UI elements: a.email text box b.password text box c. L%in button d.New customer? Create account link e. Last password? Recovery password link	Login.html	Application should show below elements: a.email text box b.password text box c.Login button with orange colour d. New custotner? Create account link e.Last password? Recovery password link	Working as expected
LoginPage_TC_OO3	Functional	Home page	Verify user is able to log into application with Valid credentials	I.Enter URL and dick go 2.Click on My Account dropdown 3.Enter Valid username/email in Email text 4.Enter valid password in password text box 5. Click On in button	Username:lax@gmail password: lax26	User should navigate to prediction homepage	working as
LoginPage_TC_OO4	Functional	Login page	Verify user is able to log into application with Invalid credentials	1, Enter URL and click go 2.Click on My Account dropdownbutton 3.Enter Invalid username/email in Email text box 4.Enter valid password in password text box 5.Click on •n button	Username:lax password:lax26	Application should show 'Incorrect email or password ' validation message.	working as
LoginPage_TC_004	Functional	Login page	Verify user is able to log into application with Invalid credentials	I-Enter URL and click go 2.Click On My Account dropdown 3.Enter Valid username/email in Email text box 4.Enter Invalid password in password text box 5.Click on in button	username:lax26@mail password:lax26	Application should show •Incorrect email or password ' validation message.	working as
LoginPage_TC_OO5	Functional	Login page	Verify user is able to into application with InValid credentials	I.Enter URL and click go 2.Click on My Account dropdown 3.Enter Invalid username/email in Email text box 4. Enter Invalid password in password text box 5. Click on I in button	username:lax26@mail password:1803	Application should show 'Incorrect email or password ' validation message.	working as
Predictionpage_TC_ 00 6	Functional	Prediction Page	Page should display whether the person is drowning or not	1, Camera should take pictures of people swimming in pools 2. It should predict the probability of drowning 3. It should show a bounding box displaying the probability Of drowning	image Of people drowning	generate a alert to lifeguard if people are drowning	Working as

9. RESULTS

9.1 PERFORMANCE MATRICES





10. ADVANTAGES & DISADVANTAGES

*ADVANTAGES

1. It represents an additional level of safety and protection for swimmers.

2. Swimmers, resort are gain in the financial.

3. It ensures effective and reliable drowning detection by limiting the number of alarms generated by disturbance factors. Prevents drowning accidents by improving the rescue time of the lifeguards.

***DISADVANTAGES**

A limitation of this equipment is that if too many swimmers, the occlusion problem arises. The other is that the camera is mounted upon the water, and monitors the Swimmer posture change.

11. CONCLUSION

Detection and fast response time are most important factor to save lives from drowning and avoid permanent damage. Thus our project would be very useful in the school swimming pools for the kids who are not so used to of swimming. It would be very useful in saving the lives of kids. It also helps in places where even adult drowning risk is high. Thus our project would be very useful in the school swimming pools for the kids who are not so used to of swimming. It would be very useful in saving the lives of kids. It also helps in places where even adult drowning risk is high.

12. FUTURE SCOPE

After the implementation of all these essentials, this system also can be used on sea beaches for drowning detection.

13. APPENDIX

IBM-Project-46565-1660750261

https://drive.google.com/drive/folders/1mZi3E2JDHpmpEwhQL