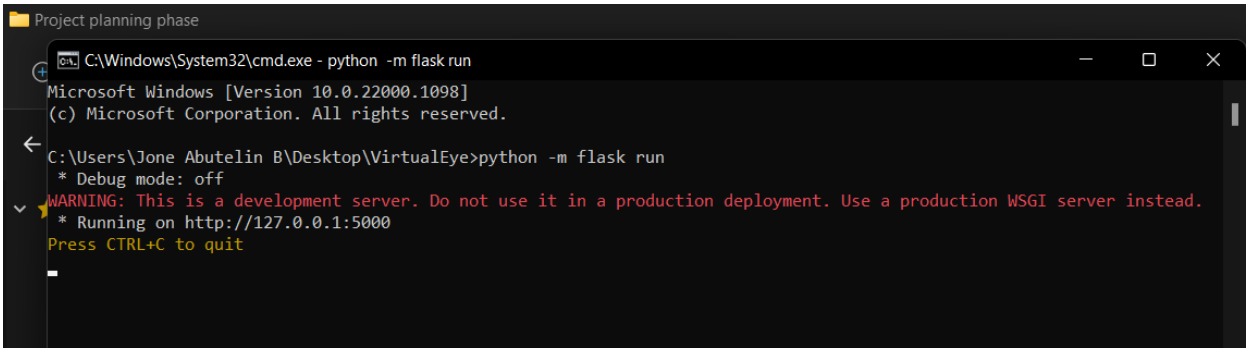
SPRINT-4

Date	17 Nov 2022
Team ID	PNT2022TMID34450
Project Name	Virtual Eye - Life Guard For Swimming Pools To Detect Active Drowning

- Run the application using the below command

  Python -m flask run



- Detection.py

```
import cvlib as cv
from cvlib.object_detection import draw_bbox
import cv2
import time
import numpy as np
from playsound import playsound
#for PiCamera
#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 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
 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)
 if(isDrowning == True):
 playsound('alarm.mp3')
 # press "Q" to stop
 if cv2.waitKey(1) & 0xFF == ord('q'):
 break
# release resources
webcam.release()
cv2.destroyAllWindows(
```

- Logout.html code

```
<!DOCTYPE html>
<html >
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1">
 <title>Virtual Eye</title>
 <link href='https://fonts.googleapis.com/css?family=Pacifico'
rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet'
type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300'
rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300'
rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Merriweather'
rel='stylesheet'>
<link href='https://fonts.googleapis.com/css?family=Josefin Sans'
rel='stylesheet'>
<link href='https://fonts.googleapis.com/css?family=Montserrat'
rel='stylesheet'>
<style>
.header {
 top:0;
 margin:0px;
 left: 0px;
 right: 0px;
 position: fixed;
 background-color: #28272c;
 color: white;
 box-shadow: 0px 8px 4px grey;
 overflow: hidden;
 padding-left:20px;
 font-family: 'Josefin Sans';
 font-size: 2vw;
```

```
 width: 100%;
 height:8%;
 text-align: center;
 }
 .topnav {
 overflow: hidden;
 background-color: #333;
}
.topnav-right a {
 float: left;
 color: #f2f2f2;
 text-align: center;
 padding: 14px 16px;
 text-decoration: none;
 font-size: 18px;
}
.topnav-right a:hover {
 background-color: #ddd;
 color: black;
}
.topnav-right a.active {
 background-color: #565961;
 color: white;
}
.topnav-right {
 float: right;
 padding-right:100px;
}

.login{
margin-top:-70px;
}
body {
 background-color:#ffffff;
 background-repeat: no-repeat;
 background-size:cover;
 background-position: 0px 0px;
 }
.main{
 margin-top:100px;
 text-align:center;
}
form { margin-left:400px;margin-right:400px;}
input[type=text], input[type=email],input[type=number],input[type=password] {
 width: 100%;
 padding: 12px 20px;
 display: inline-block;
 margin-bottom:18px;
```

```
 border: 1px solid #ccc;
 box-sizing: border-box;
}
button {
 background-color: #28272c;
 color: white;
 padding: 14px 20px;
 margin-bottom:8px;
 border: none;
 cursor: pointer;
 width: 20%;
}
button:hover {
 opacity: 0.8;
}
.cancelbtn {
 width: auto;
 padding: 10px 18px;
 background-color: #f44336;
}
.imgcontainer {
 text-align: center;
 margin: 24px 0 12px 0;
}
img.avatar {
 width: 30%;
 border-radius: 50%;
}
.container {
 padding: 16px;
}
span.psw {
 float: right;
 padding-top: 16px;

}
/* Change styles for span and cancel button on extra small screens */
@media screen and (max-width: 300px) {
 span.psw {
 display: block;
 float: none;
 }
 .cancelbtn {
 width: 100%;
 }
}

</style>
```

```
</head>
<body style="font-family:Montserrat;">
<div class="header">
<div style="width:50%;float:left;font-size:2vw;text-align:left;color:white;
padding-top:1%">Virtual eye</div>
 <div class="topnav-right" style="padding-top:0.5%;">

 <a href="{{ url_for('home')}}">Home</a>
 <a href="{{ url_for('login')}}">Login</a>
 <a href="{{ url_for('register')}}">Register</a>
 </div>
</div>
<div class="main">
<h1>Successfully Logged Out!</h1>
<h3 style="color:#4CAF50">Login for more information<h3>
 <a href="{{ url_for('login') }}"><button type="submit">Login</button></a>
</form>
</div>
</body>
</html>
```