SPECIMEN

VIRTUALEYE-LIFE GUARD FOR SWIMMING POOLS TO DETECT ACTIVE DROWNING

A PROJECT REPORT

Submitted by

B.Jone Abutelin
K.C.Femi Priya
S.T.Sreema
R.Suthi

in partial fulfilment for the award of the degree
of

BACHELOR OF ENGINEERING
in

COMPUTER SCIENCE AND ENGINEERING

PONJESLY COLLEGE OF ENGINEERING ,NAGERCOIL

ANNA UNIVERSITY::CHENNAI 600 025
NOV 2022

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"VIRTUALEYE-LIFE GUARD FOR SWIMMING POOLS TO DETECT ACTIVE DROWNING"** is the bonafide work of **"B. JONE ABUTELIN(961819104050), K.C. FEMI PRIYA(961819104034), S.T. SREEMA(961819104084), R. SUTHI(961819104086)"** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
MRS.MARIA SHEEBA,M.E.,	**MRS.M.MANCHU,M.TECH.,**
HEAD OF THE DEPARTMENT	**MENTOR**
	Assistant Professor
Department of Computer Science	**Department of Computer Science And**
And Engineering	**Engineering**
Ponjesly College of Engineering,	**Ponjesly College of Engineering,**
Nagercoil-629 003	**Nagercoil-629 003**

TABLE OF CONTENTS

ABSTRACT

Every year, many individuals, including kids under the age of 5 drown in the deep of the swimming pool, and the lifeguards are not well-trained enough to handle these situations. Thus arises the requirement for having a system that will consequently detect the drowning individuals and alarm the lifeguard at such risk. 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. YOLO-based Convolutional Neural Network family of models can be used for object detection

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW:

Safety in water has been a concern for many centuries for the survival of human lives. The latest technology advancements have enabled us to come up with effective drowning detection methods. A recent report from World Health Organization (WHO) gives us some insight into drowning incidents globally. The number of reported drowning deaths globally is 37200. The highest numbers of deaths are in low and middle-income countries. The majority of the drowning deaths are reported from open water bodies like lakes and seas, and not in pools. In the report WHO has recommended various drowning prevention techniques constructing fences across the lakes, to prevent accidental falls to teach school-age children swimming as a part of their curriculum in schools . In active drowning, the victim expresses distress that is noticeable to others. In passive drowning, there is no distress exhibited by the victim.

In order to quickly judge lifesavers whether people are drowning in the swimming pool, one efficient behavior recognition approach by means of video sequences underwater. First, by analyzing the spatial distribution of the swimming pools when swimmers are normally swimming, the data labeling and swimmer detection methods are determined. Second, a behavior recognition framework of swimmers on the basis of the YOLOv3 algorithm is proposed.The spatial

relationship between the location information of the target and the swimming/drowning area of the swimming pool is analyzed to further determine the swimmer’s drowning or swimming behavior. The detection accuracy of different detection algorithms and analyzes the detection effect of different pool angles and different swimmer densities were compared. Test results show that the mean precision rate of drowning is 94.62%, the mean false rate is 1.43%, and the mean missing rate is 3.57%.

1.2 PURPOSE:

When it comes to pool safety, there is no such thing as being too careful. Drowning detection systems have been shown to mitigate the potential for accidental death by drowning in a swimming pool. This is especially true when it comes to children who play unsupervised in and around the pool. Even more nearly drowning victims are left with irreversible injuries, mostly to the brains, due to lack of timely rescue. Drowning is a tragedy that can be easily prevented with proper supervision and the use of pool safety systems. Drowning detection helps the lifeguard to detect the underwater situation where they can't easily observe. As it signals by alarming instantly when it detects any potential difficulty in swimming, it helps the life guard in immediately taking the action and rescuing them. The system also keeps an eye out for potentially dangerous actions that could result in drowning.Therefore, there is a clear need for automated drowning detection systems to provide useful assistance to lifeguards on duty or to enhance the safety of unattended pools.

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING PROBLEM:

The wearable drowning monitor device can detect drowning accident and alarm. The device has seven main modules, including microprocessor, power module, SD memory card module, LED warning module, acceleration sensor module, water pressure sensor module, and keys module. When swimming the human arm must constantly waving in the water, if drowning, arm motion of floating is significantly reduced, and if falling into the water, almost motionless. According to the physiological response of human drowning, it can detect drowning accident by recording arm motion realtime through wearable wrist accelerometer device. This accelerometer is packed with embedded functions with edible user programmable options, configurable to two interrupt pins. The pressure sensor is installed to judge whether the human body is in the water. The red LED is used for drowning warning. One blue LED is used to get the work status of the device which wills flash every few seconds in order to save the precious energy. Because LED lightemitting angle generally relatively small, 5 red LED lights of upward and around direction is installed to make LED alarm signal caller. Two keys are designed for the demo device.

One is the switch for power. The other is a self-help button. If drowning danger occurs, the swimmer can push the button and the blue LED will shine for help, and if a swimmer accidentally hit the button, he can push the button to cancel the alarm. If the swimmer lost consciousness because of drowning, the device detects the drowning accident and will ON LED light to inform the lifeguard. The device is worn on the wrist and move in large amplitude along with the wrist when a human is swimming in the water, and the data acquired from accelerator will dramatically change. If a human is drowning in water, his or her wrist almost motionless, and the data acquired from accelerator will have only small changes due to water movement. The drowning detection method uses threshold. First, data from a water pressure sensor is used to judge whether the human body in the water, if the body in the water, then start drowning judgment process. Then, analog signal obtained from the three axis acceleration sensor is converted to digital signal and three axis acceleration values are gained. Hanning filtering method and the moving average filtering are used to reduce noise error.The problem with this system is that if the swimmer loses his watch then the drowning can't be detected.

2.2 REFERENCES

[1] Muhammad Ramdhan MS1, Muhammad Ali2, et al., "An Early Drowning Detection
System for the Internet of Things (IoT) Applications ", August 2018, TELKOMNIKA, Vol.16, No.4, pp. 1870~1876 ISSN: 1693-6930

[2] L.Fei,W.Xueli,"Drowning Detection Based on Background Subtraction,"
2009 International
Conference on Embedded Software and Systems, Zhejiang, 2009, pp. 341-
343

[3] Ahmad Ilham et al., "Au Float (Autonomous Float) Based-on Artificial Intelligent and
LORA (Long Range) Using Haar Cascade Method for Rescuing of Water Accident Victims," 2019, International Symposium on Electronics and Smart Devices (ISESD),
Badung-Bali, Indonesia, 2019, pp. 1-4.

[4] Wenmiao Lu and Yap-Peng Tan, "Swimmer motion analysis with application to drowning
detection," 2002 IEEE International Symposium on Circuits and Systems. Proceedings (Cat.
No.02CH37353), Phoenix-Scottsdale, AZ, USA, 2002, pp. II-II.

[5] Roy and K. Srinivasan, "A novel drowning detection method for the safety of Swimmers,"
2018 20th National Power Systems Conference (NPSC), Tiruchirappalli,
India, 2018.

[6] S. Sindhuja, "MEMS based-self-regulating airbag drowning aversion system for submerged swimmers," 2015 International Conference on Circuits, Power and

Computing Technologies
[ICCPCT-2015], Nagercoil, 2015, pp. 1-4

[7] CAI Xiaoyang, W. Chen and F. Lei, "Application of Image Restoration Based on Robust
Estimation in Drowning Warning System," Second Workshop on Digital Media and its
Application in Museum & Heritages (DMAMH 2007), Chongqing, 2007, pp.
33-35. 11.

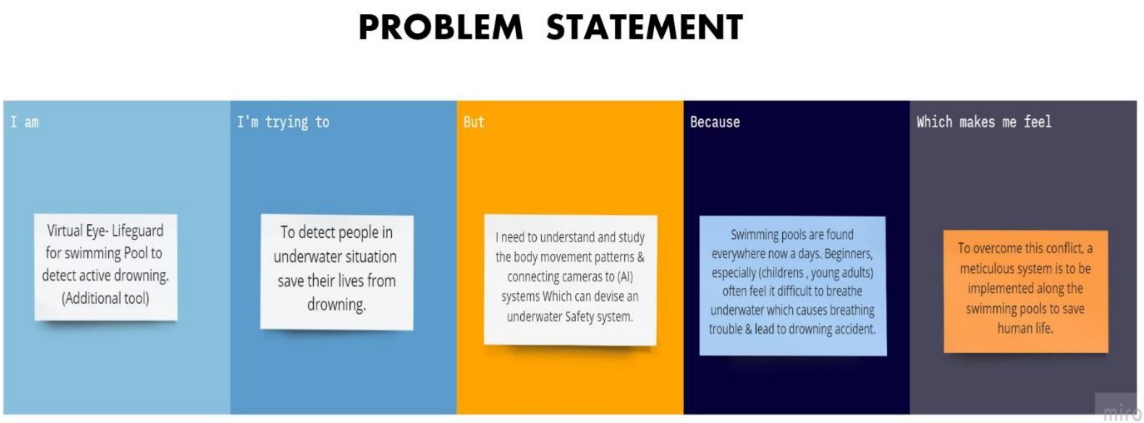
[8] L.Fei, W. Xueli and C. Dongsheng, "Drowning Detection Based on Background Subtraction,"
2009 International Conference on Embedded Software and Systems,
Zhejiang, 2009, pp. 341-
343.

[9] H. Liu, M.B.H. Frej and B. Wen, "A Novel Method for Recognition, Localization, and
Alarming to Prevent Swimmers from Drowning," 2019 IEEE Cloud Summit, Washington,
DC, USA, 2019, pp. 65-71.

2.3 PROBLEM STATEMENT DEFINITION

A problem statement is important to a process improvement project because it helps clearly identify the goals of the project and outline the scope of a project. It also helps guide the activities and decisions of the people who are working on the project. The problem statement can help a business or organization gain support and buy-in for a process improvement project. A good problem statement can be created by identifying and answering several questions related to the problem.

This process involves identifying what the problem is, why it is a problem, when and where the problem was identified, who the problem impacts, how they are impacted by the problem and how much of an impact the problem has. Creating a problem statement to understand customer's point of view. The below shown block diagram is a perfect example for our topic.



PROBLEM STATEMENT

I am	I'm trying to	But	Because	Which makes me feel
Virtual Eye- Lifeguard for swimming Pool to detect active drowning. (Additional tool)	To detect people in underwater situation save their lives from drowning.	I need to understand and study the body movement patterns & connecting cameras to (AI) systems Which can devise an underwater Safety system.	Swimming pools are found everywhere now a days. Beginners, especially (childrens , young adults) often feel it difficult to breathe underwater which causes breathing trouble & lead to drowning accident.	To overcome this conflict, a meticulous system is to be implemented along the swimming pools to save human life.

Fig 1.Problem Statement

CHAPTER 3

IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

Empathy maps are an efficient tool used by designers to not only understand user behaviour, but also visually communicate those findings to colleagues, uniting the team under one shared understanding of the user. Essentially, an empathy map is a square divided into four quadrants with the user or client in the middle. Each of the four quadrants comprises a category that helps us delve into the mind of the user. The four empathy map quadrants look at what the user says, thinks, feels, and does.

With the user at the centre and the categories in each of the four surrounding quadrants, an empathy map arranges all of your research about the user into an easy-toread visual.

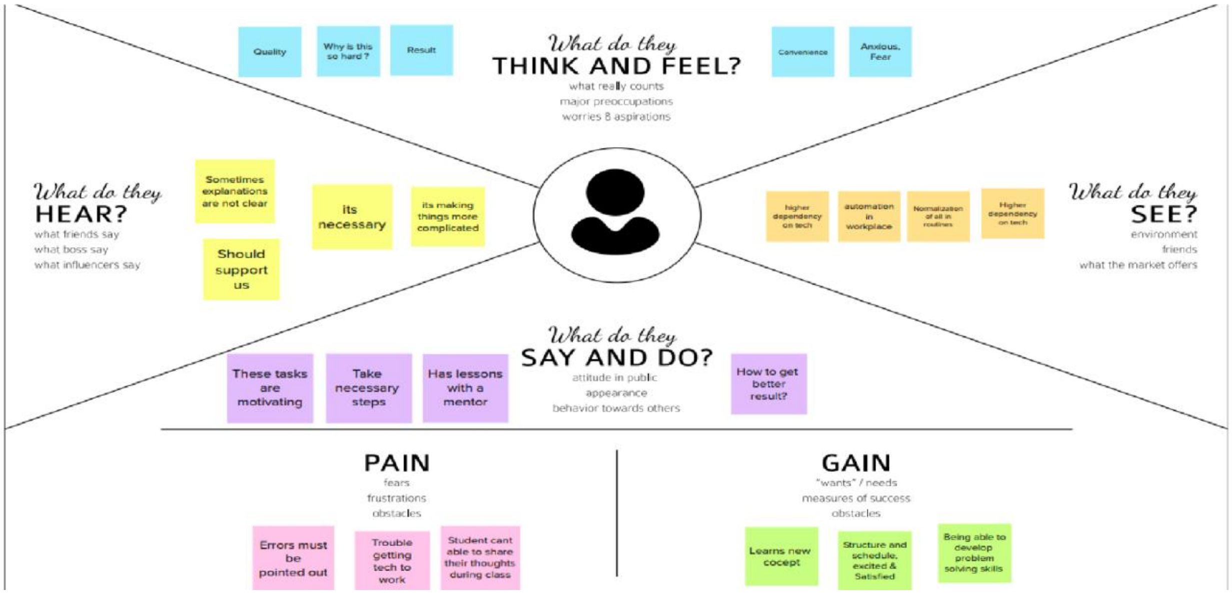


Fig 2.Empathy map

3.2 IDEATION & BRAINSTORMING

Brainstorming is a method design teams use to generate ideas to solve clearly defined design problems. Brainstorming is a method of generating ideas and sharing knowledge to solve a particular commercial or technical problem, in which participants are encouraged to think without interruption. Brainstorming is a group activity where each participant shares their ideas as soon as they come to mind. At the conclusion of the session, ideas are categorised and ranked for follow-on action.

When planning a brainstorming session it is important to define clearly the topic to be addressed. A topic which is too specific can constrict thinking, while an ill-defined topic will not generate enough directly applicable ideas. The composition of the brainstorming group is important too. It should include people linked directly with the subject as well as those who can contribute novel and unexpected ideas. It can comprise staff from inside or outside the organisation.

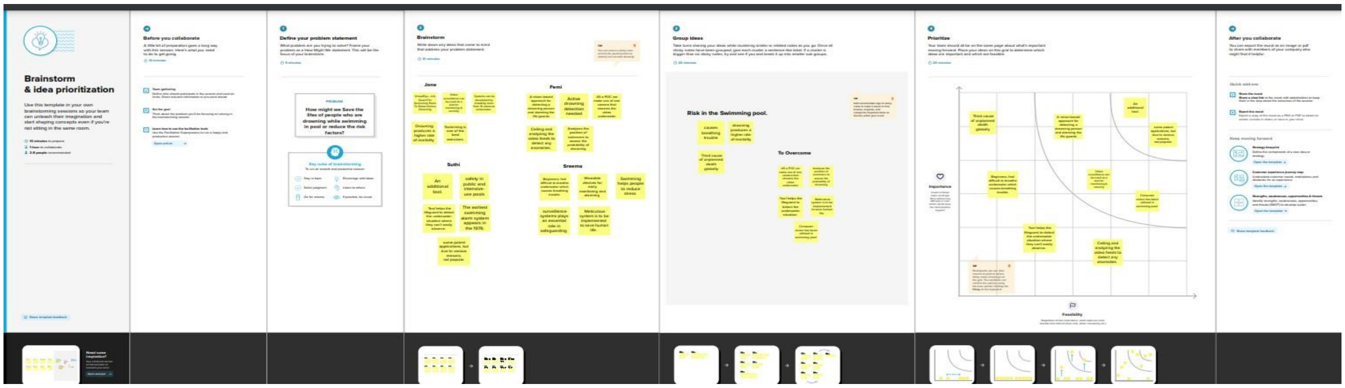

Brainstorm & idea prioritization
Before you collaborate
Define your problem statement
PROBLEM
How might we Save the lifes of people who are drowning while swimming in pool or reduce the risk factors?
Key rules of brainstorming
Brainstorm
Jone
Femi
Suthi
Sreema
Active drowning detection needed
An additional tool
safety in public and intensive-use pools
The earliest swimming alarm system appears in the 1976.
Swimming helps people to reduce stress.
surveillance systems plays an essential role in safeguarding
Group ideas
Risk in the Swimming pool.
causes breathing trouble
drowning produces a higher rate of mortality
Third cause of unplanned death globally
To Overcome
Prioritize
Importance
Feasibility
Ceiling and analyzing the video feeds to detect any anomalies.
After you collaborate

Fig 3:Brainstorming

3.3 PROPOSED SOLUTION:

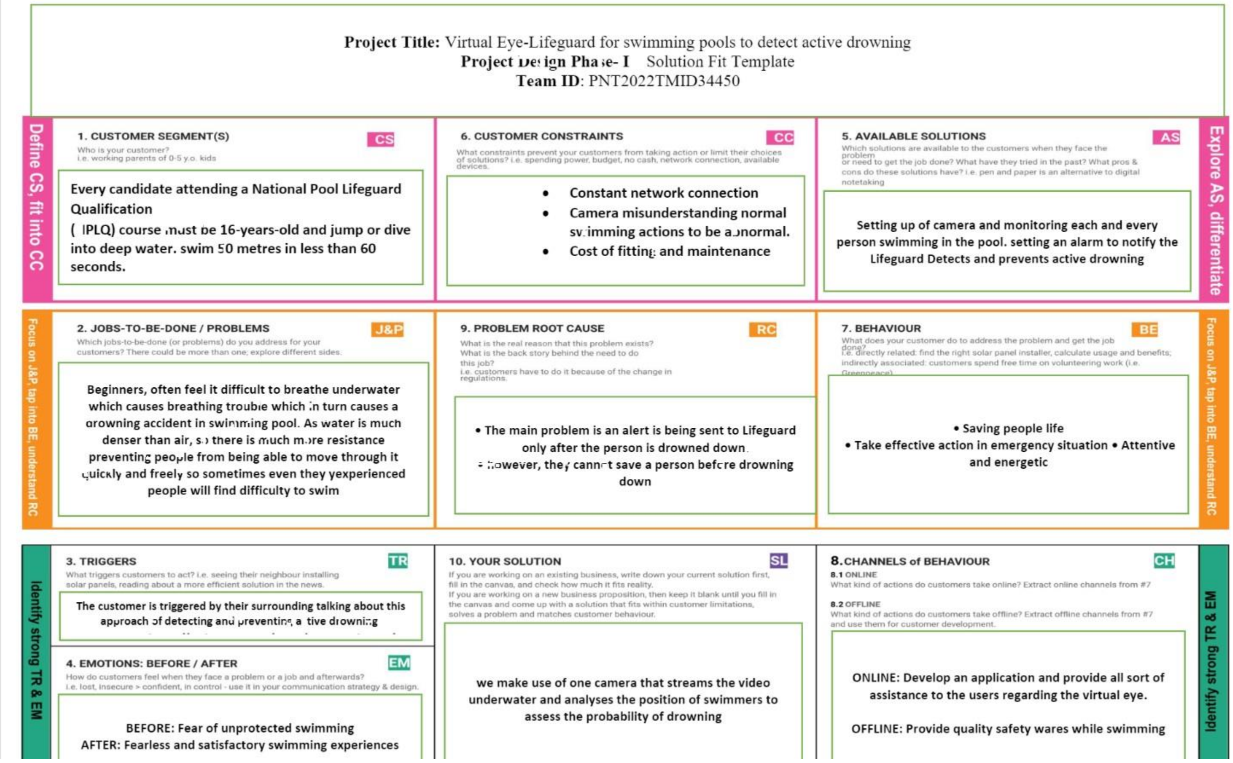
Proposed Solution means the technical solution to be provided by the Implementation agency in response to the requirements and the objectives of the Project.

S.NO	PARAMETER	DESCRIPTION
1	Problem Statement (Problem to be solved)	The aim is to make practical safety alerts to reduce the danger of drowning incidents in swimming pools
2	Idea/Solution description	We came up with a solution that detects drowning people by 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
3	Novelty/ Uniqueness	The proposed system makes 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
4	Social Impact /Customer Satisfaction	This ensures the safety of the people and very positive impact on rates of drowning death and injury.
5	Business Model(Revenue Model)	Subscription model - The subscription business model is a business model in which a customer must pay a recurring price at regular intervals for access to a product or service.

6	Scalability of the Solution	It can be used in all swimming pools because it is budget friendly

3.4 PROBLEM SOLUTION FIT:

Problem-Solution Fit - this occurs when you have evidence that customers care about certain jobs, pains, and gains. At this stage you've proved the existence of a problem and have designed a value proposition that addresses your customers' jobs, pains and gains.



Project Title: Virtual Eye-Lifeguard for swimming pools to detect active drowning
Project Design Phase- I Solution Fit Template
Team ID: PNT2022TMID34450

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS
Who is your customer?
i.e. working parents of 0-5 y.o. kids

Every candidate attending a National Pool Lifeguard Qualification (IPLQ) course must be 16-years-old and jump or dive into deep water. swim 50 metres in less than 60 seconds.

6. CUSTOMER CONSTRAINTS CC
What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

- Constant network connection
- Camera misunderstanding normal swimming actions to be abnormal.
- Cost of fitting and maintenance

5. AVAILABLE SOLUTIONS AS
Which solutions are available to the customers when they face the problem
or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking

Setting up of camera and monitoring each and every person swimming in the pool. setting an alarm to notify the Lifeguard Detects and prevents active drowning

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P
Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

Beginners, often feel it difficult to breathe underwater which causes breathing trouble which in turn causes a drowning accident in swimming pool. As water is much denser than air, so there is much more resistance preventing people from being able to move through it quickly and freely so sometimes even they yexperienced people will find difficulty to swim

9. PROBLEM ROOT CAUSE RC
What is the real reason that this problem exists?
What is the back story behind the need to do this job?
i.e. customers have to do it because of the change in regulations.

- The main problem is an alert is being sent to Lifeguard only after the person is drowned down.
- However, they cannot save a person before drowning down

7. BEHAVIOUR BE
What does your customer do to address the problem and get the job done?
i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

- Saving people life
- Take effective action in emergency situation
- Attentive and energetic

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TR
What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

The customer is triggered by their surrounding talking about this approach of detecting and preventing active drowning

4. EMOTIONS: BEFORE / AFTER EM
How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

BEFORE: Fear of unprotected swimming
AFTER: Fearless and satisfactory swimming experiences

10. YOUR SOLUTION SL
If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

we make use of one camera that streams the video underwater and analyses the position of swimmers to assess the probability of drowning

8. CHANNELS of BEHAVIOUR CH
8.1 ONLINE
What kind of actions do customers take online? Extract online channels from #7
8.2 OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

ONLINE: Develop an application and provide all sort of assistance to the users regarding the virtual eye.

OFFLINE: Provide quality safety wares while swimming

Identify strong TR & EM

Fig 4: Problem Solution Fit

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS:

Following are the functional requirements of the proposed solution

FR-No.	**Functional Requirement (Epic)**	**Sub Requirement (Story / Sub-Task)**
FR-1	Camera Installation	Cameras should be installed inside water and in the walls of the building
FR-2	Sensor Installation	Installed under the water without disturbing the people
FR-3	Deduction	Either not moving or in unconscious state
FR-4	Alert	Sends an alert message to the lifeguard.
FR-5	Support	Lifeguard help or swim tubes
FR-6	Alarm	Rings alarm with drowning detected.

4.2 Non-functional Requirements:

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

NFR-No.	**Non-Functional requirements**	**Requirement Description**
NFR-1	Usability	When someone is drowning, the sensor detects and locate the swimmer who is drowning and alert the people.
NFR-2	Security	Lifeguards will be present in the pool and the cameras are secured by the management and are safe.
NFR-3	Reliability	The process will be a reliable multimedia video based surveillance system.
NFR-4	Performance	The alarm is triggered when the swimmer is detected as drowning.
NFR-5	Availability	Detection equipment includes safety wheel, pool hook, rescue tubes, first aid box etc.
NFR-6	Scalability	Virtual eye lifeguard detects potential drownings and promptly notifies you. It features the latest artificial intelligence technology and adapts to the needs of the user.

CHAPTER 5

PROJECT DESIGN

5.1 DATA FLOW DIAGRAM:

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.

- ➤ The drowning detector captures the video frames using the cameras installed inside the swimming pool.

- ➤ Using the yolo algorithm the person is detected and confirms that is the person is stable or he is drowning.

- ➤ If the person is drowning then it raises an alarm for lifesavers to quicklet taking the action rescuing the drowning person.

- ➤ If it doesn’t detect any drowning it will keep on monitoring for providing security to the swimmers.

The Data Flow Graph of our proposed solution is shown below:

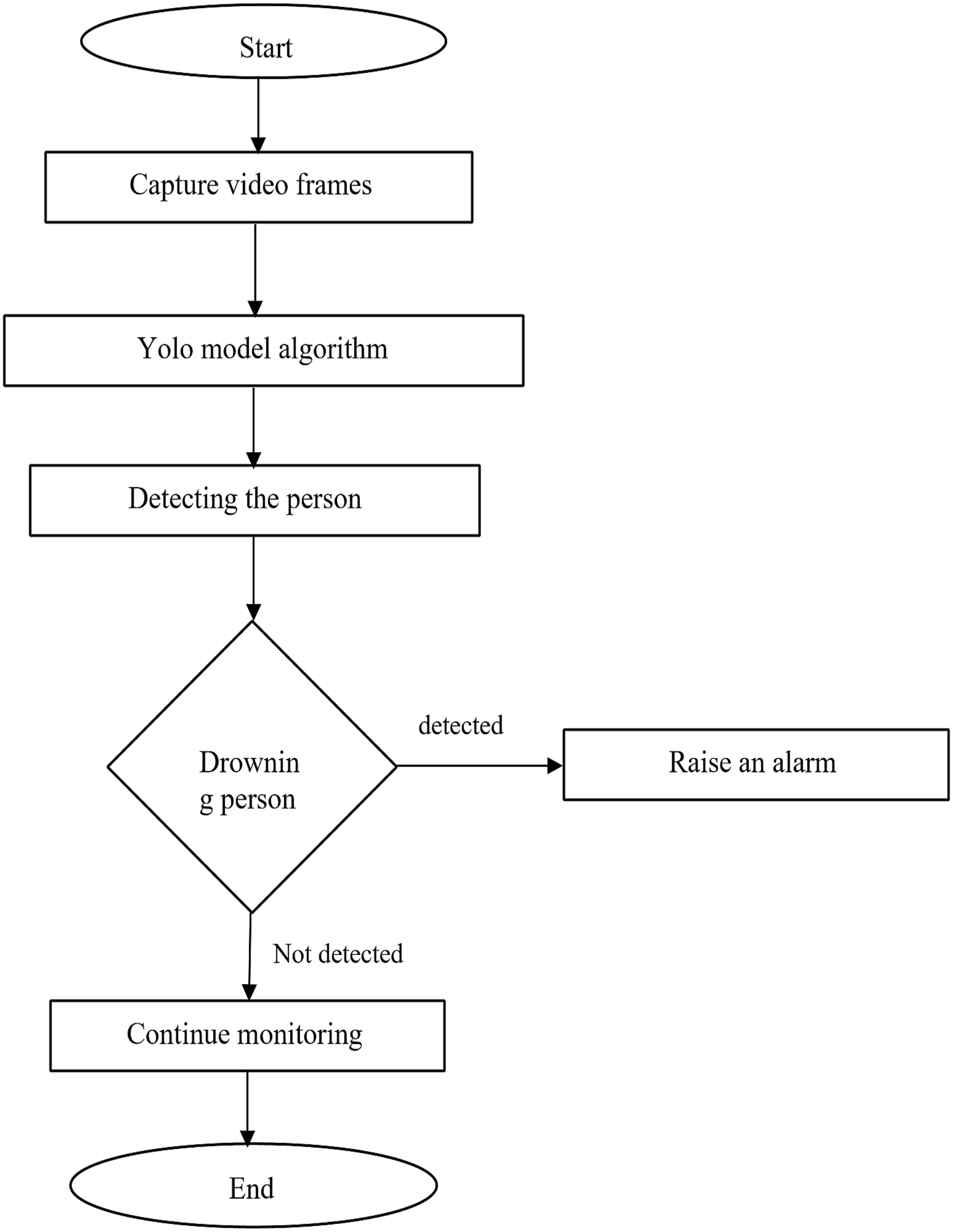


Fig 5. Data Flow Diagram

5.2 SOLUTION & TECHNICAL ARCHITECTURE

Solution Architecture:

Solution architecture is a complex process – with many subprocesses – that bridges the gap between business problems and technology solutions. Its goals are to:

- ➤ Find the best tech solution to solve existing business problems.
- ➤ Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- ➤ Define features, development phases, and solution requirements.
- ➤ Provide specifications according to which the solution is defined, managed, and delivered.

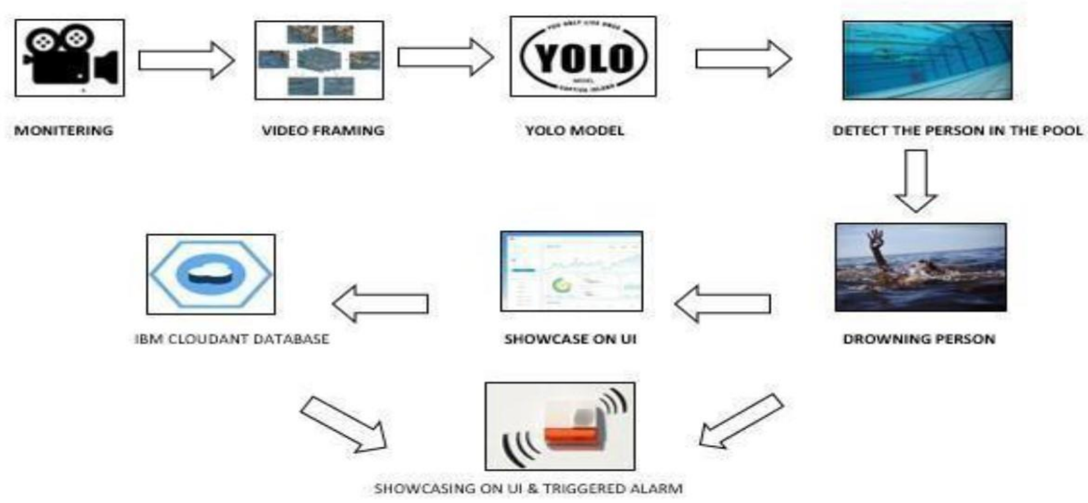


Fig 6. Solution Architecture

Technical Architecture:

Technical architecture—which is also often referred to as application architecture, IT architecture, business architecture, etc.—refers to creating a structured software solution that will meet the business needs and expectations while providing a strong technical plan for the growth of the software application through its lifetime. ITarchitecture is equally important to the business team and the information technology team.

Technical architecture includes the major components of the system, their relationships, and the contracts that define the interactions between the components. The goal of technical architects is to achieve all the business needs with an application that is optimized for both performance and security. IT architects plan for things they know are coming in the future and for things they don’t yet envision or dream. Taking the time to design the architecture at the start will prevent major design changes, code refactoring, and expensive rework later in the project.

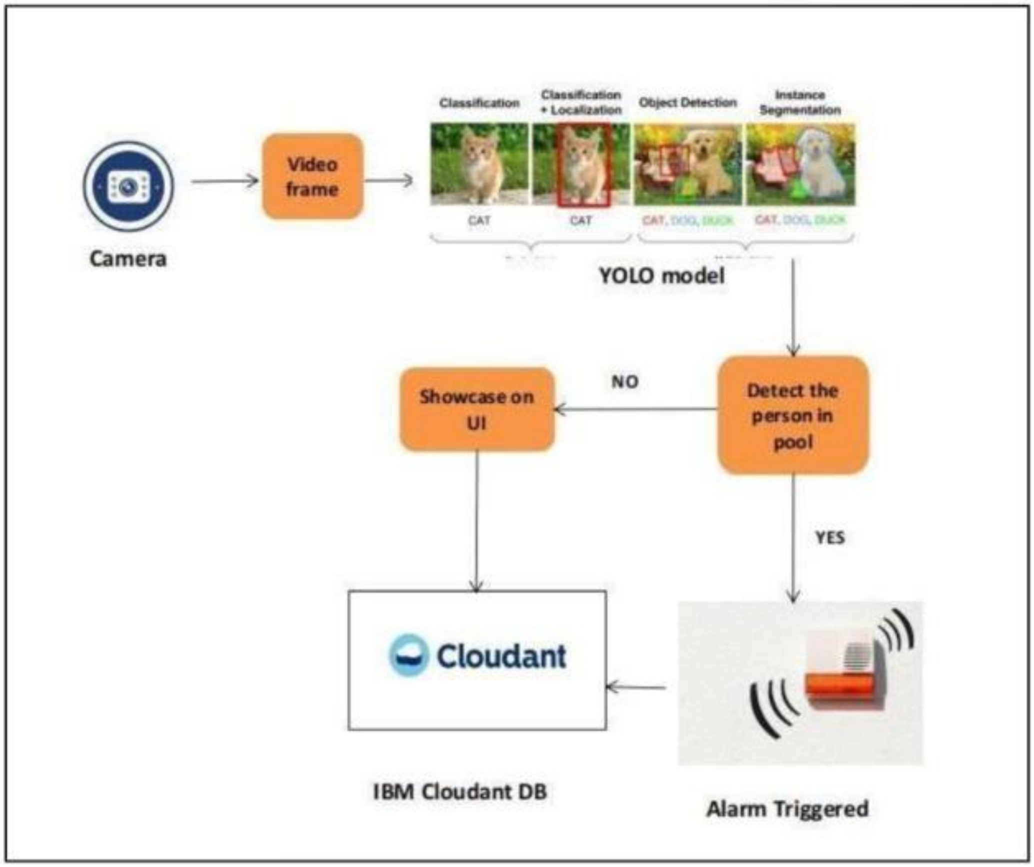
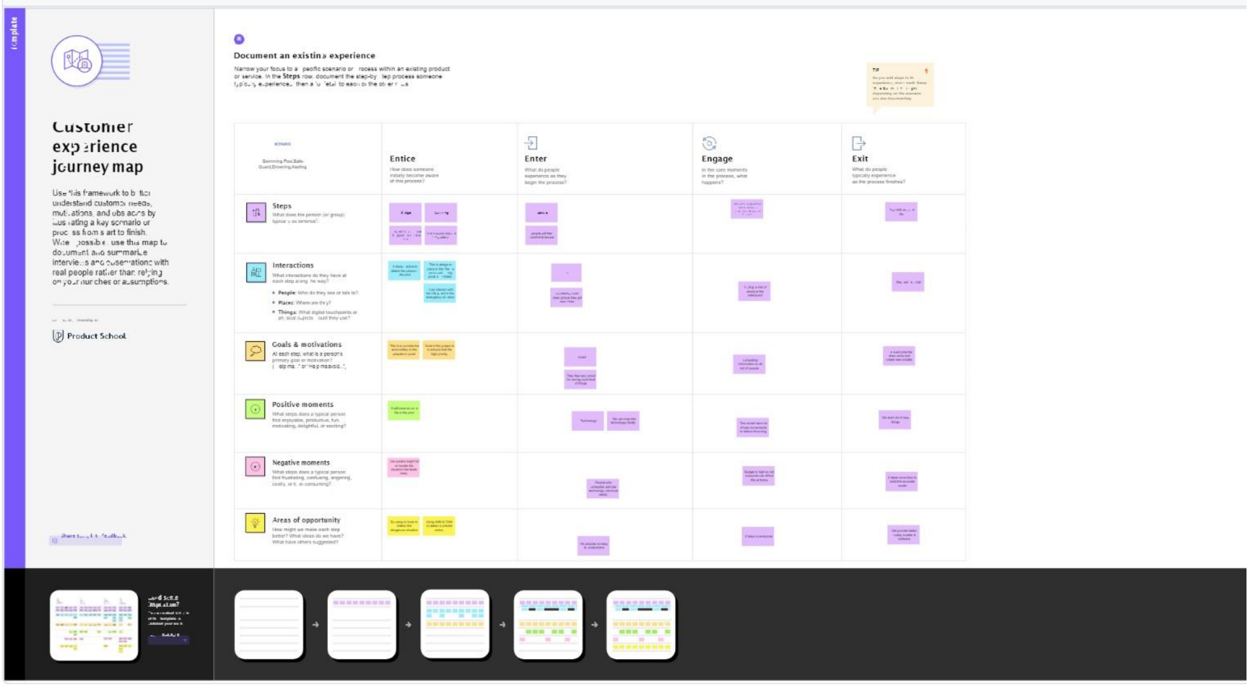


Fig 7. Technical Architecture

5.3 USER STORIES:

A user story is an informal, general explanation of a software feature written from the perspective of the end user. Its purpose is to articulate how a software feature will provide value to the customer. It's tempting to think that user stories are, simply put, software system requirements. But they're not.

A key component of agile software development is putting people first, and a user story puts end users at the center of the conversation. These stories use non technical language to provide context for the development team and their efforts. After reading a user story, the team knows why they are building, what they're building, and what value it creates. User stories are one of the core components of an agile program. They help provide a user-focused framework for daily work — which drives collaboration, creativity, and a better product overall.


template
Customer experience journey map
Use this framework to better understand customers needs, motivations, and obstacles by illustrating a key scenario or process from start to finish. Whenever possible, use this map to document and summarize interviews and observations with real people rather than relying on your hunches or assumptions.
Product School
Document an existing experience
Narrow your focus to a specific scenario or process within an existing product or service. In the Steps row, document the step-by-step process someone typically experiences, then add detail to each of the other rows.
TIP
Swimming Pool Safe-Guard,Drowning,Alerting

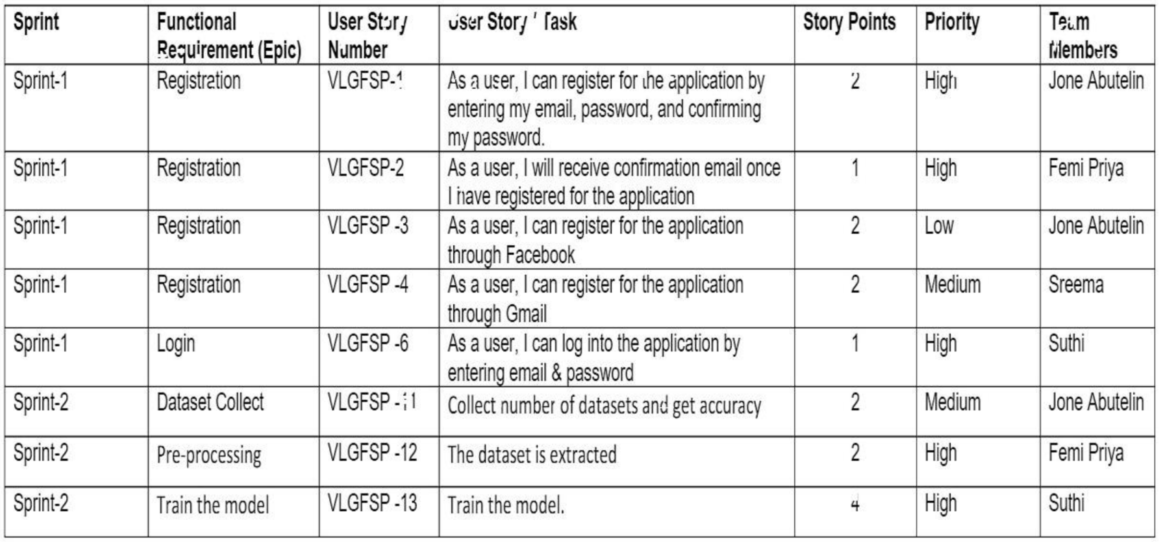
	Entice How does someone initially become aware of this process?	Enter What do people experience as they begin the process?	Engage In the core moments in the process, what happens?	Exit What do people typically experience as the process finishes?
Steps What does the person (or group) typically experience?	[illegible]	[illegible]	[illegible]	[illegible]
Interactions What interactions do they have at each step along the way? People: Who do they see or talk to? Places: Where are they? Things: What digital touchpoints or physical objects would they use?	[illegible]	[illegible]	[illegible]	[illegible]
Goals & motivations At each step, what is a person's primary goal or motivation? ("Help me..." or "Help me avoid..")	[illegible]	[illegible]	[illegible]	[illegible]
Positive moments What steps does a typical person find enjoyable, productive, fun, motivating, delightful, or exciting?	[illegible]	[illegible]	[illegible]	[illegible]
Negative moments What steps does a typical person find frustrating, confusing, angering, costly, or time-consuming?	[illegible]	[illegible]	[illegible]	[illegible]
Areas of opportunity How might we make each step better? What ideas do we have? What have others suggested?	[illegible]	[illegible]	[illegible]	[illegible]

Fig 8. Customer journey

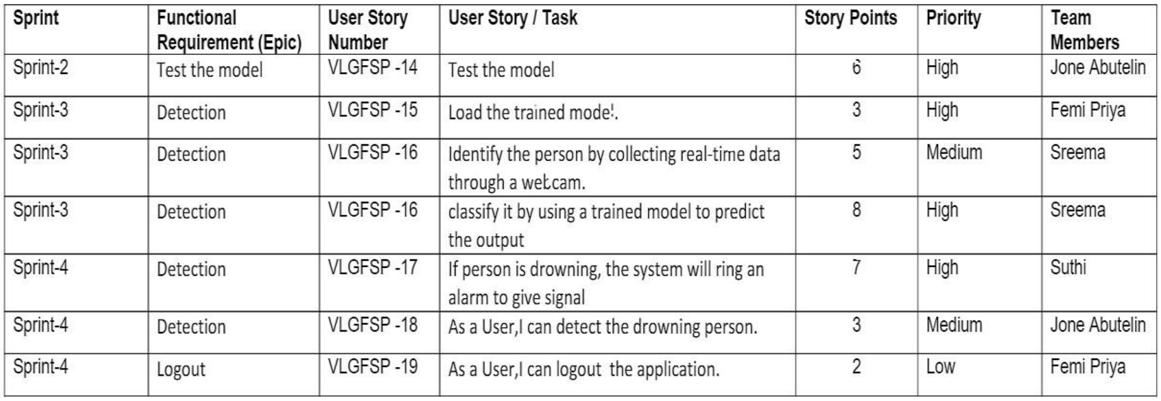
CHAPTER 6

PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION:

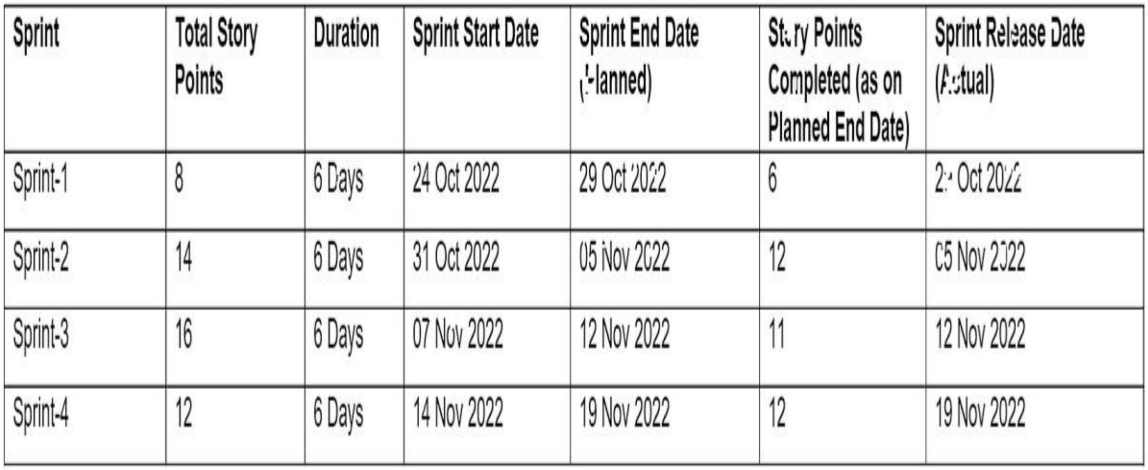


Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	VLGFSP-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Jone Abutelin
Sprint-1	Registration	VLGFSP-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Femi Priya
Sprint-1	Registration	VLGFSP -3	As a user, I can register for the application through Facebook	2	Low	Jone Abutelin
Sprint-1	Registration	VLGFSP -4	As a user, I can register for the application through Gmail	2	Medium	Sreema
Sprint-1	Login	VLGFSP -6	As a user, I can log into the application by entering email & password	1	High	Suthi
Sprint-2	Dataset Collect	VLGFSP -11	Collect number of datasets and get accuracy	2	Medium	Jone Abutelin
Sprint-2	Pre-processing	VLGFSP -12	The dataset is extracted	2	High	Femi Priya
Sprint-2	Train the model	VLGFSP -13	Train the model.	4	High	Suthi



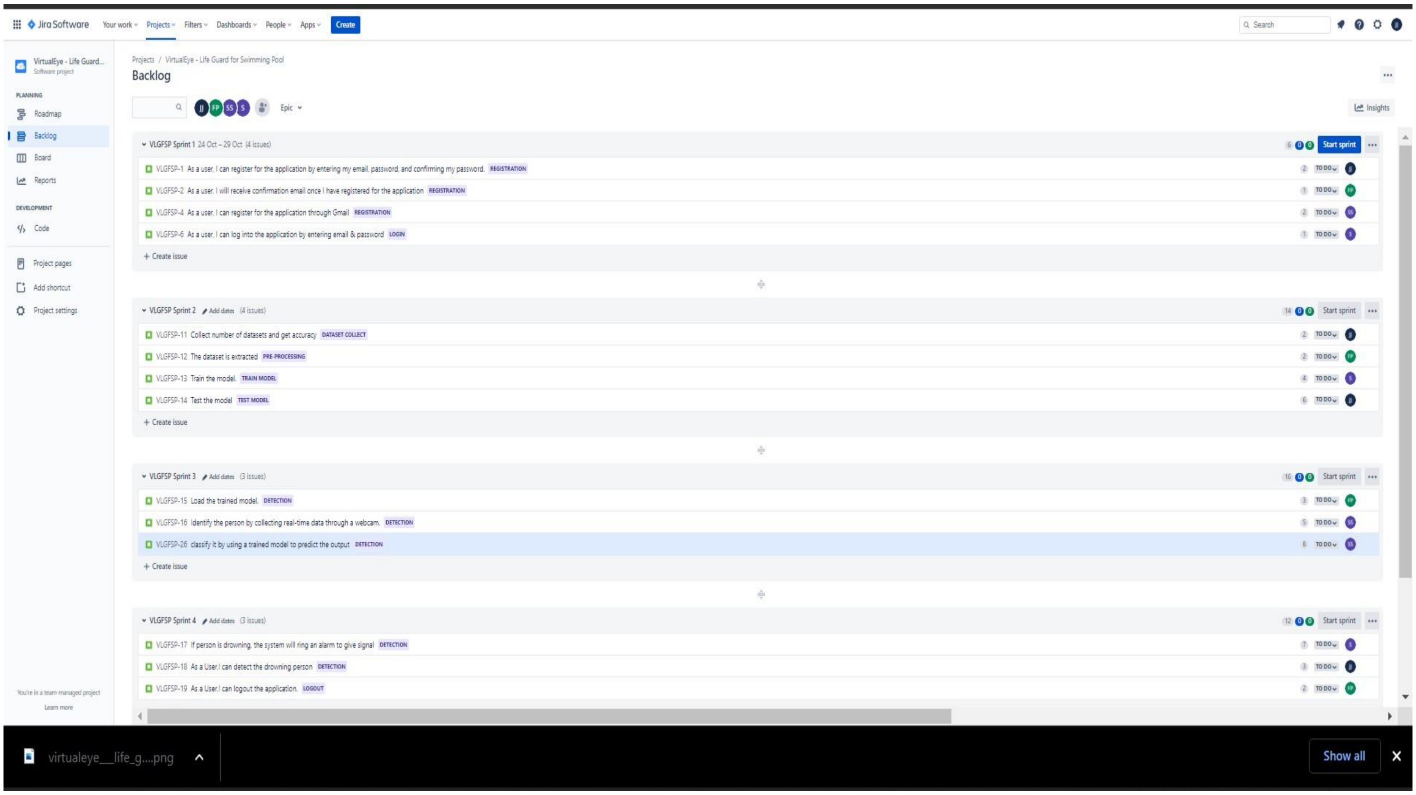
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2	Test the model	VLGFSP -14	Test the model	6	High	Jone Abutelin
Sprint-3	Detection	VLGFSP -15	Load the trained model.	3	High	Femi Priya
Sprint-3	Detection	VLGFSP -16	Identify the person by collecting real-time data through a webcam.	5	Medium	Sreema
Sprint-3	Detection	VLGFSP -16	classify it by using a trained model to predict the output	8	High	Sreema
Sprint-4	Detection	VLGFSP -17	If person is drowning, the system will ring an alarm to give signal	7	High	Suthi
Sprint-4	Detection	VLGFSP -18	As a User,I can detect the drowning person.	3	Medium	Jone Abutelin
Sprint-4	Logout	VLGFSP -19	As a User,I can logout the application.	2	Low	Femi Priya

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	8	6 Days	24 Oct 2022	29 Oct 2022	6	29 Oct 2022
Sprint-2	14	6 Days	31 Oct 2022	05 Nov 2022	12	05 Nov 2022
Sprint-3	16	6 Days	07 Nov 2022	12 Nov 2022	11	12 Nov 2022
Sprint-4	12	6 Days	14 Nov 2022	19 Nov 2022	12	19 Nov 2022

6.3 REPORTS FROM JIRA:



Projects / VirtualEye - Life Guard for Swimming Pool

Backlog

VLGFSP Sprint 1 24 Oct – 29 Oct (4 issues) 6 0 0 Start sprint

- VLGFSP-1 As a user, I can register for the application by entering my email, password, and confirming my password. REGISTRATION 2 TO DO JJ
- VLGFSP-2 As a user, I will receive confirmation email once I have registered for the application REGISTRATION 1 TO DO FP
- VLGFSP-4 As a user, I can register for the application through Gmail REGISTRATION 2 TO DO SS
- VLGFSP-6 As a user, I can log into the application by entering email & password LOGIN 1 TO DO S

VLGFSP Sprint 2 Add dates (4 issues) 14 0 0 Start sprint

- VLGFSP-11 Collect number of datasets and get accuracy DATASET COLLECT 2 TO DO JJ
- VLGFSP-12 The dataset is extracted PRE-PROCESSING 2 TO DO FP
- VLGFSP-13 Train the model. TRAIN MODEL 4 TO DO S
- VLGFSP-14 Test the model TEST MODEL 6 TO DO JJ

VLGFSP Sprint 3 Add dates (3 issues) 16 0 0 Start sprint

- VLGFSP-15 Load the trained model. DETECTION 3 TO DO FP
- VLGFSP-16 Identify the person by collecting real-time data through a webcam. DETECTION 5 TO DO SS
- VLGFSP-26 classify it by using a trained model to predict the output DETECTION 8 TO DO SS

VLGFSP Sprint 4 Add dates (3 issues) 12 0 0 Start sprint

- VLGFSP-17 If person is drowning, the system will ring an alarm to give signal DETECTION 7 TO DO S
- VLGFSP-18 As a User,I can detect the drowning person DETECTION 3 TO DO JJ
- VLGFSP-19 As a User,I can logout the application. LOGOUT 2 TO DO FP

Fig 9. Sprint Assigned Page

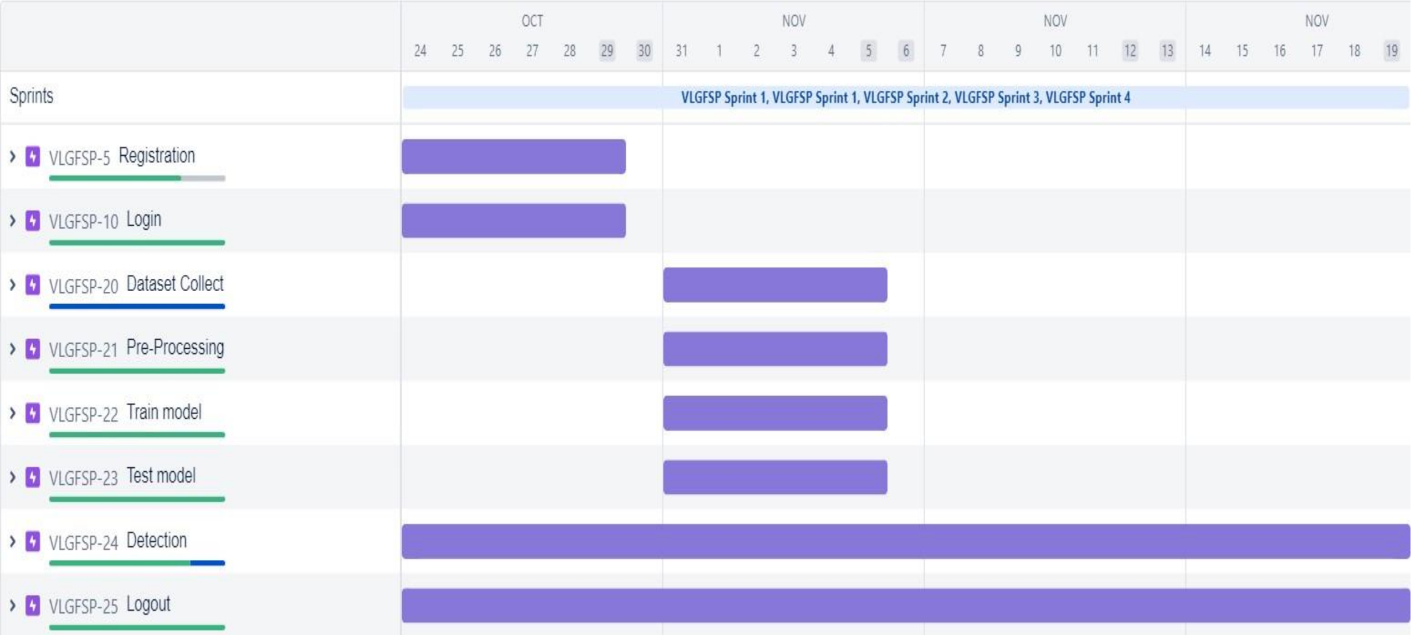


Fig 10. Road Map After Tasks Done

CHAPTER 7

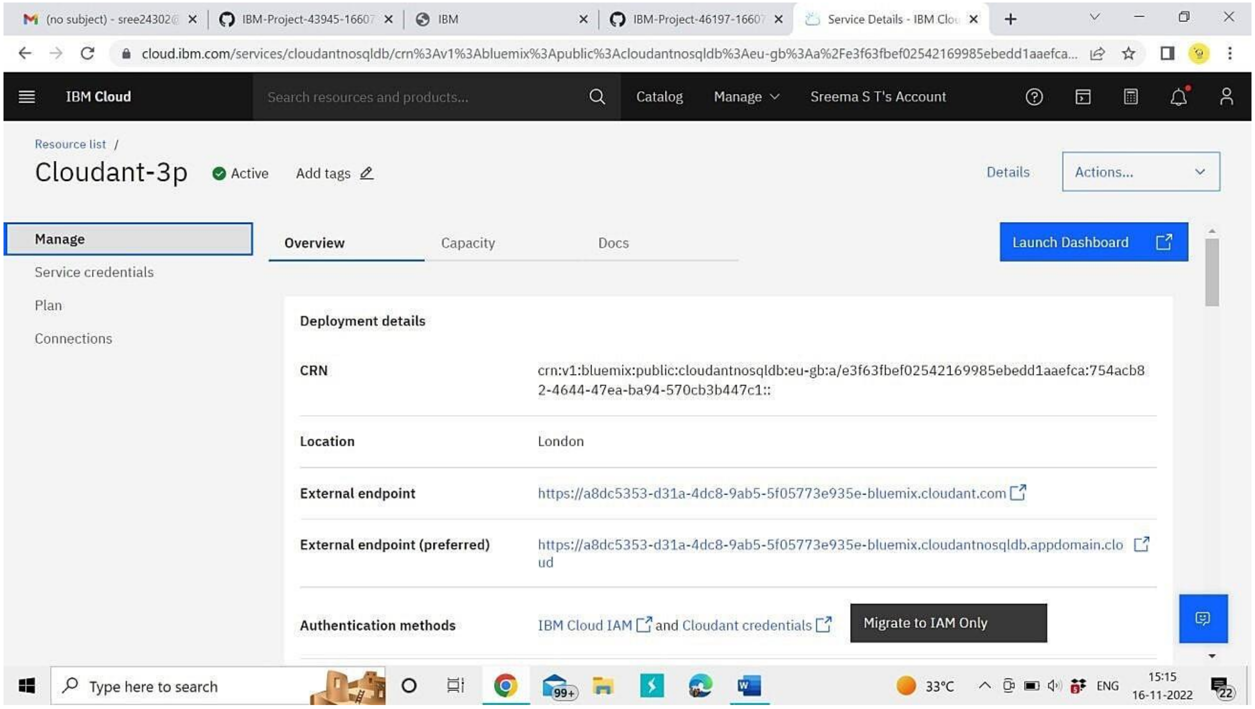
CODING & SOLUTIONING

7.1 FEATURE 1

Launch Cloudant DB

The very first process in this project section is to develop account on IBM cloud

My Cloudant DB launches



Deployment details

CRN crn:v1:bluemix:public:cloudantnosqldb:eu-gb:a/e3f63fbef02542169985ebedd1aaefca:754acb82-4644-47ea-ba94-570cb3b447c1::

Location London

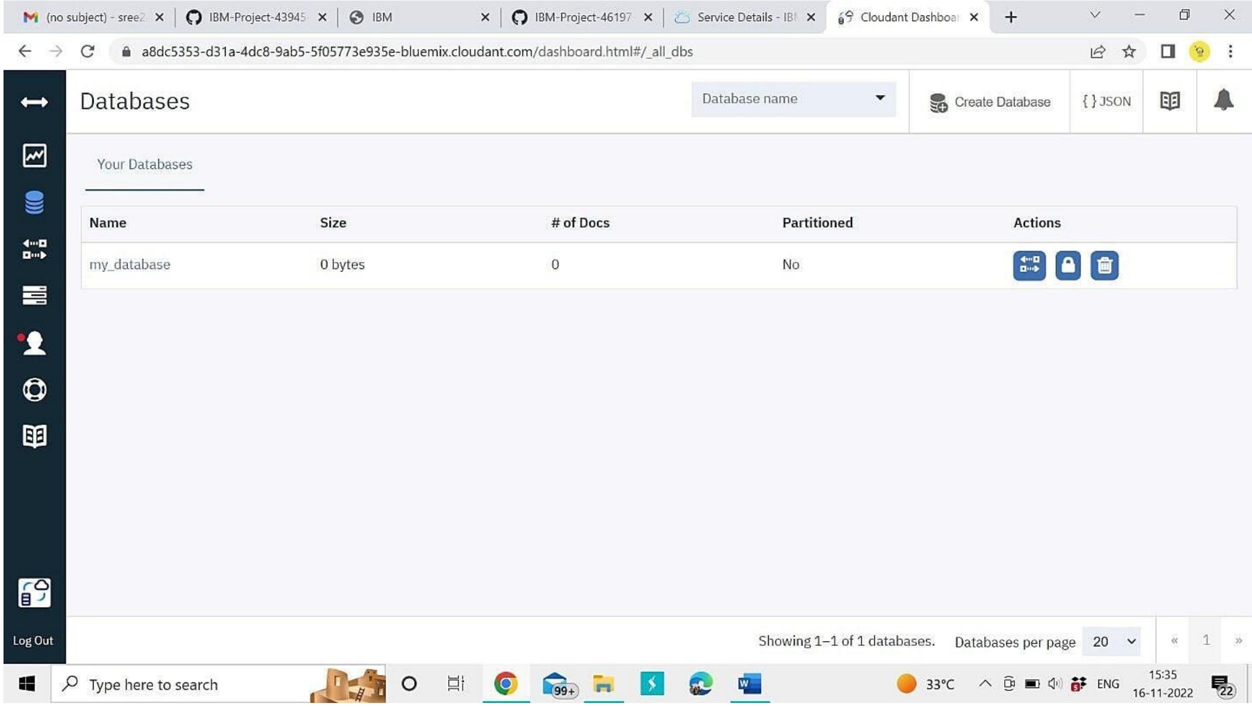
External endpoint https://a8dc5353-d31a-4dc8-9ab5-5f05773e935e-bluemix.cloudant.com

External endpoint (preferred) https://a8dc5353-d31a-4dc8-9ab5-5f05773e935e-bluemix.cloudantnosqldb.appdomain.cloud

Authentication methods IBM Cloud IAM and Cloudant credentials

Fig 11. Launch DataBase

- Click on my_database



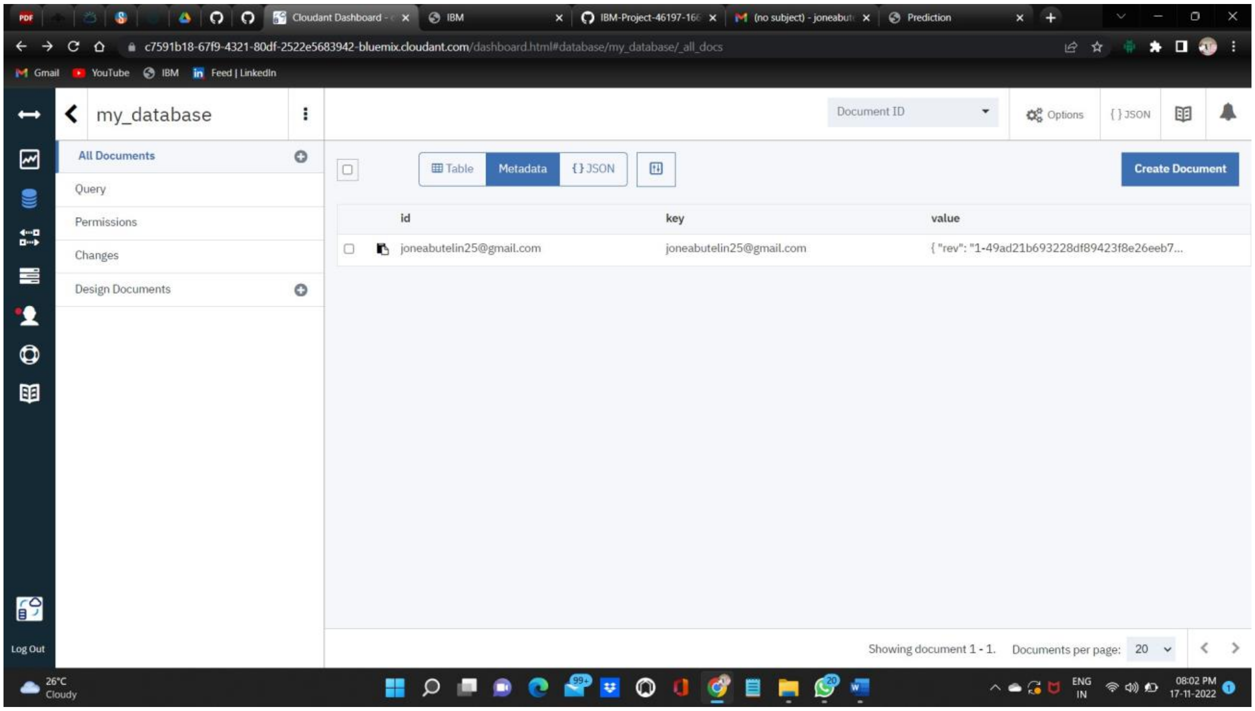
Databases

Your Databases

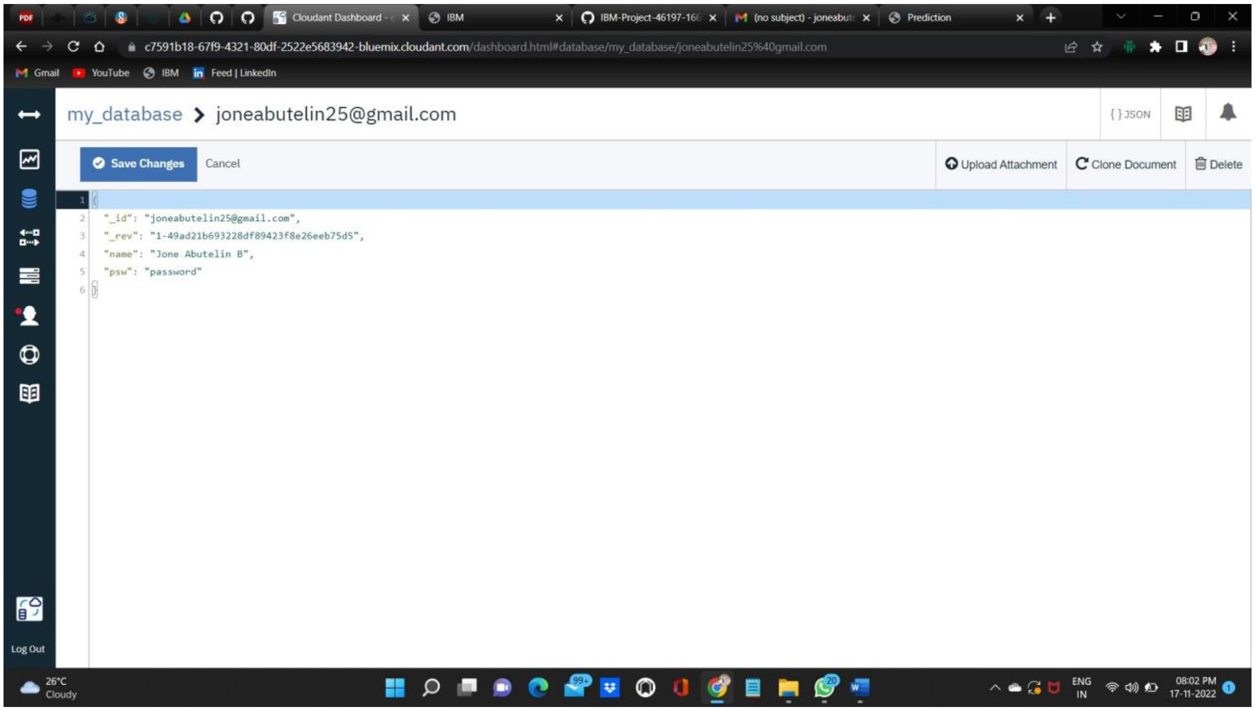
Name	Size	# of Docs	Partitioned	Actions
my_database	0 bytes	0	No	

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

- The username and password given in the register page of application will be stored in the cloud database.



id	key	value
joneabutelin25@gmail.com	joneabutelin25@gmail.com	{ "rev": "1-49ad21b693228df89423f8e26eeb7…



```
{
  "_id": "joneabutelin25@gmail.com",
  "_rev": "1-49ad21b693228df89423f8e26eeb75d5",
  "name": "Jone Abutelin B",
  "psw": "password"
}
```

7.2 FEATURE 2

PYTHON CODE

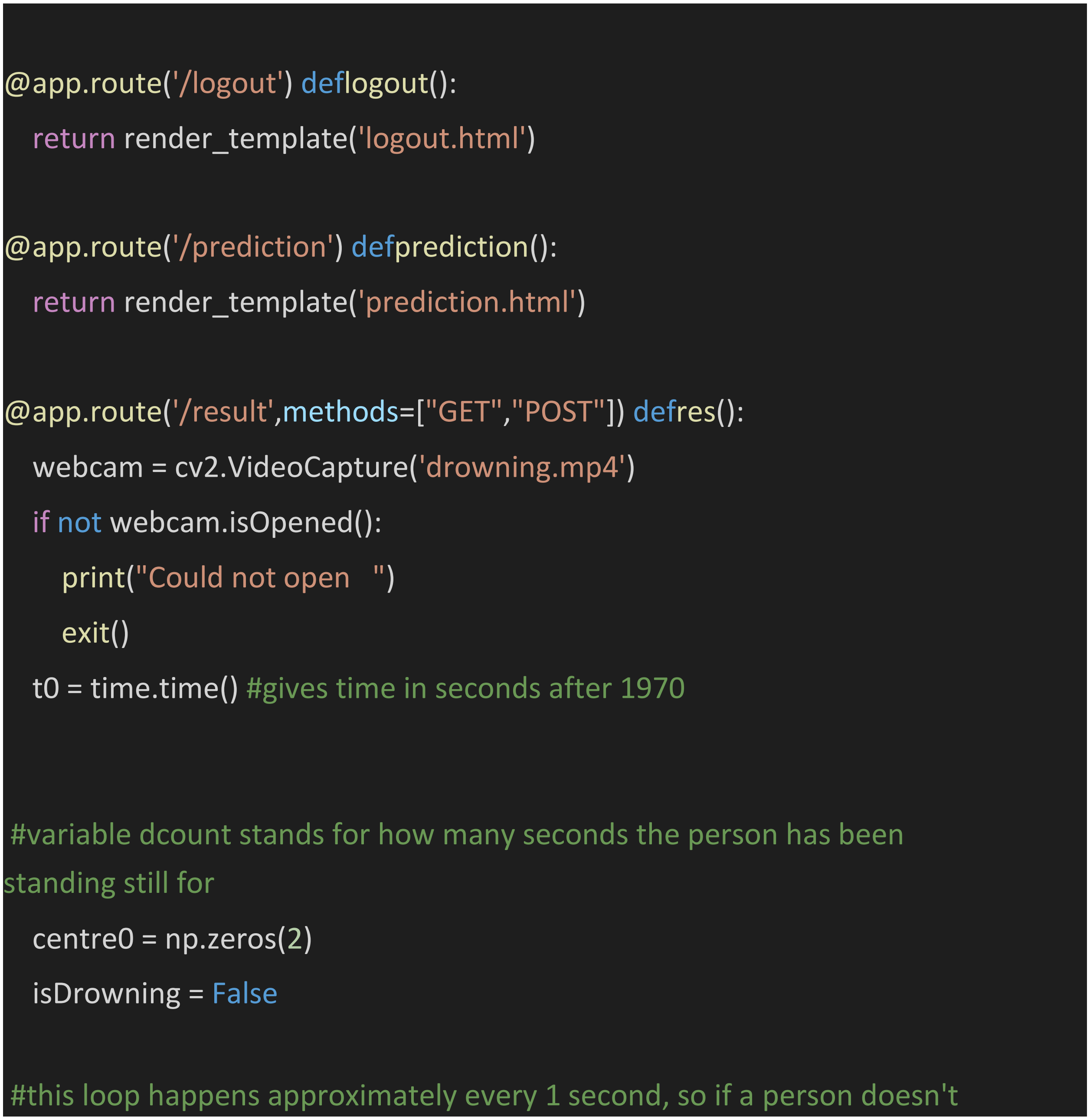
Our Python Code is very Simple and easy to understand. The programs carries our device detalis and the requirements of the project are kept defined. All conditions are made properly and the output is done successfully.

App.py

```
import time import cv2 import numpy as np from cloudant.client
import Cloudant from flask import Flask, request, render_template,
redirect, url_for from playsound import playsound import cvlib as cv
from cvlib.object_detection import draw_bbox
# Loading the model




client          =                Cloudant.iam('c7591b18-67f9-4321-80df-2522e5683942-

bluemix','xPQROZL0yOiM7VyTbPJG4t4DxBu670yEpV7NlTBz25ZN', connect=True)
# Create a database using an initialized client my_database
= client.create_database('my_database')
app=Flask(__name__) #default home page or route
@app.route('/') defindex():
   return render_template('index.html')
@app.route('/index.html')
```

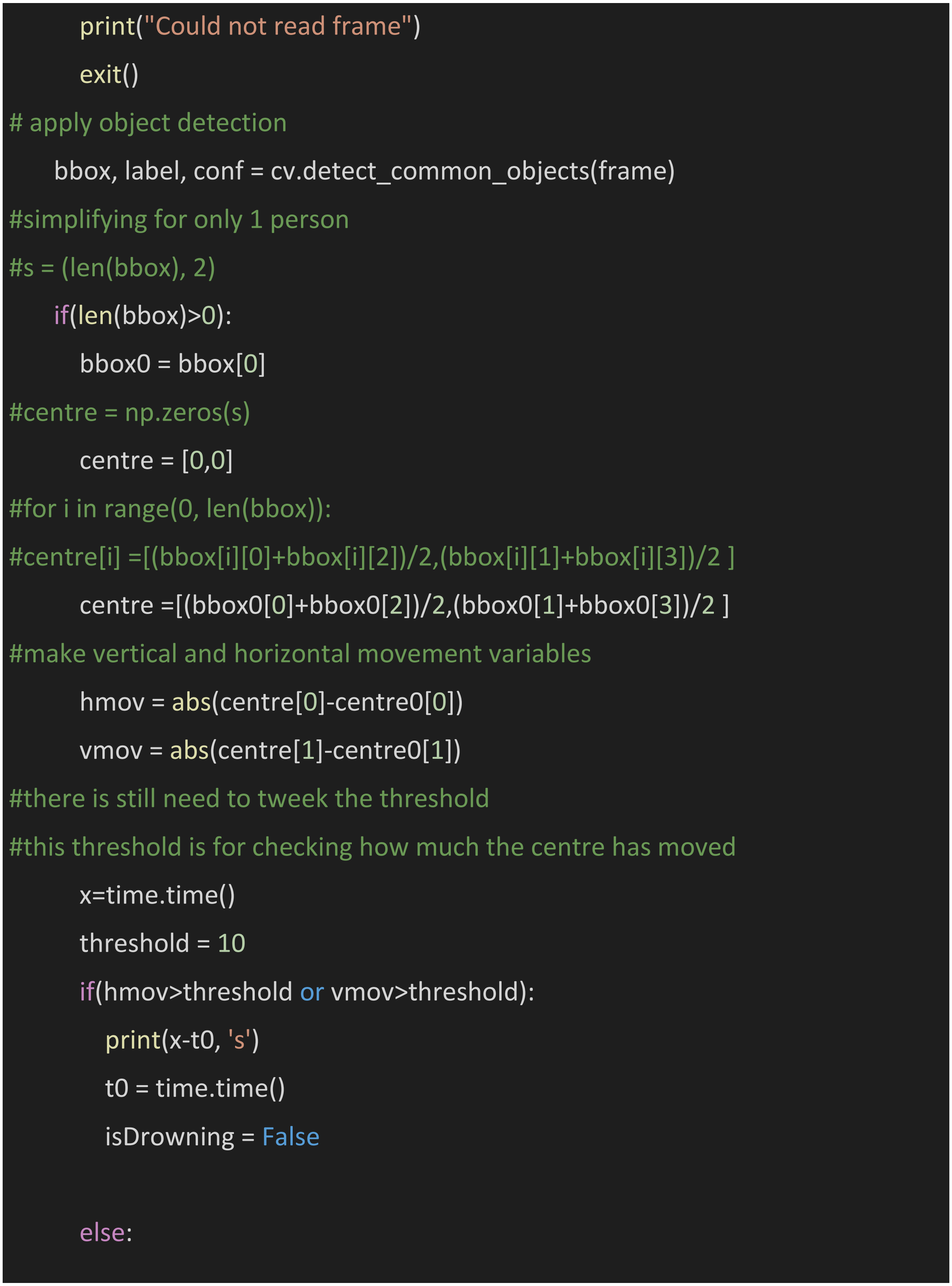
```
defhome():
   return render_template("index.html")
#registration page
@app.route('/register') defregister():
   return render_template('register.html')
@app.route('/afterreg', methods=['POST']) defafterreg():
   x = [x for x in request.form.values()]
   print(x)
   data = {
   '_id': x[1], # Setting _id is optional
   'name': x[0],
   'psw':x[2]
   }
   print(data)

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

   docs = my_database.get_query_result(query)
   print(docs)

   print(len(docs.all()))

   if(len(docs.all())==0):
      url = my_database.create_document(data)
   #response = requests.get(url) return
        render_template('register.html',        pred="Registration
please login using your details")
                                                                   Successful,
```

```
  else:
    return render_template('register.html', pred="You are already a member,
please login using your details")
#login page
@app.route('/login') deflogin():
  return render_template('login.html')
@app.route('/afterlogin',methods=['POST']) defafterlogin():
  user = request.form['_id']
  passw = request.form['psw']
  print(user,passw)

  query = {'_id': {'$eq': user}}

  docs = my_database.get_query_result(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('prediction'))
    else:
      print('Invalid User')
```



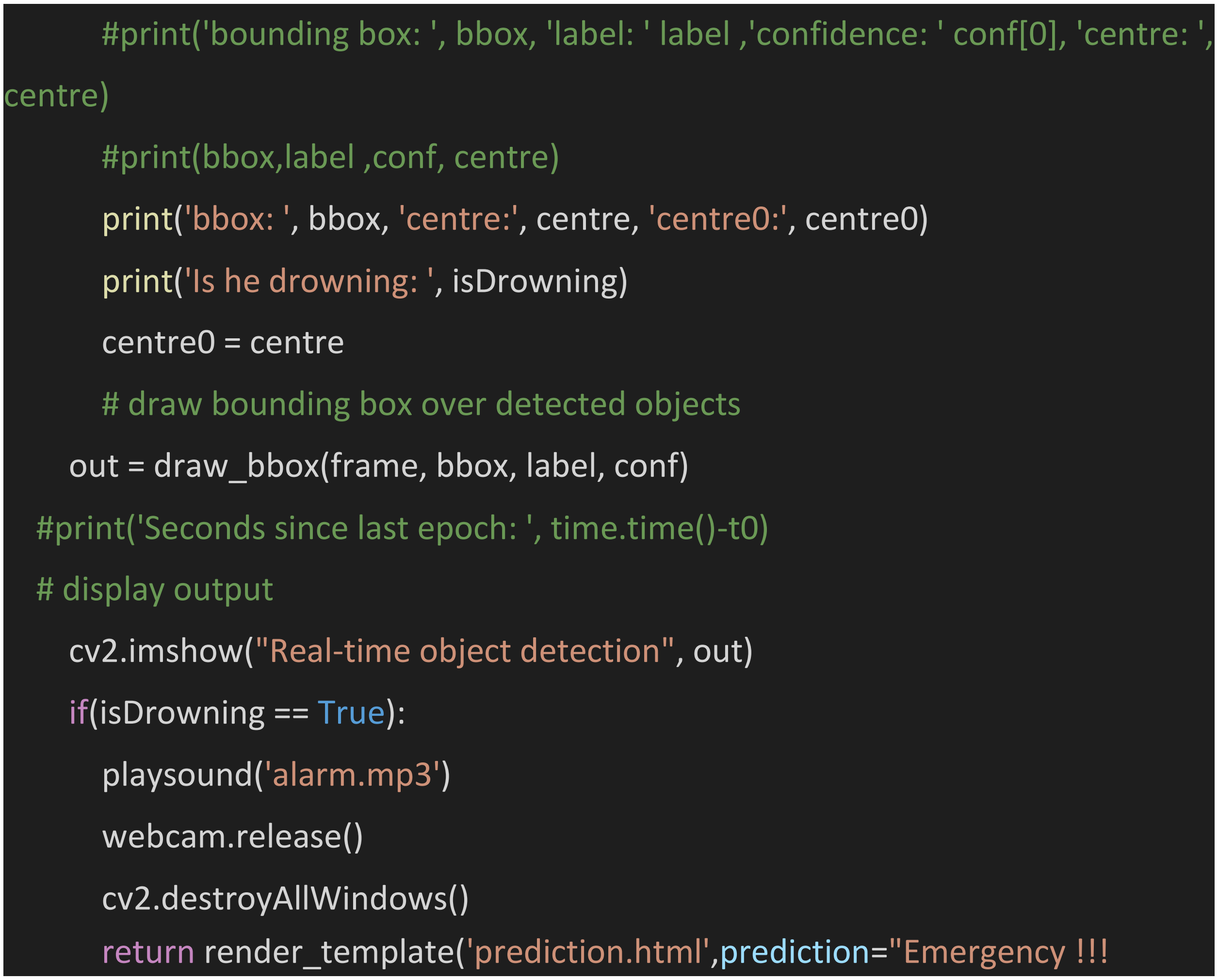
```
@app.route('/logout') deflogout():
   return render_template('logout.html')

@app.route('/prediction') defprediction():
   return render_template('prediction.html')

@app.route('/result',methods=["GET","POST"]) defres():
   webcam = cv2.VideoCapture('drowning.mp4')
   if not webcam.isOpened():
      print("Could not open   ")
      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:
```

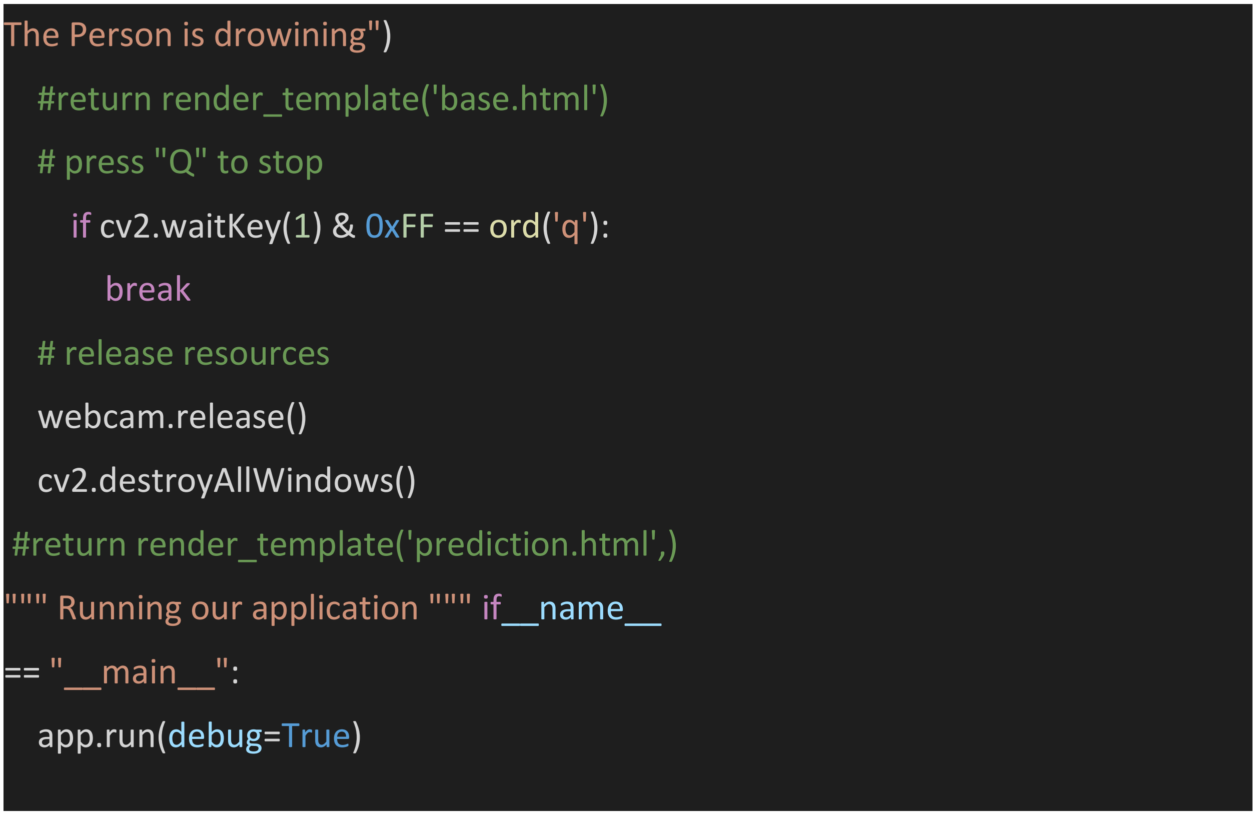


```
        print("Could not read frame")
        exit()
# apply object detection
    bbox, label, conf = cv.detect_common_objects(frame)
#simplifying for only 1 person
#s = (len(bbox), 2)
    if(len(bbox)>0):
        bbox0 = bbox[0]
#centre = np.zeros(s)
        centre = [0,0]
#for i in range(0, len(bbox)):
#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
```



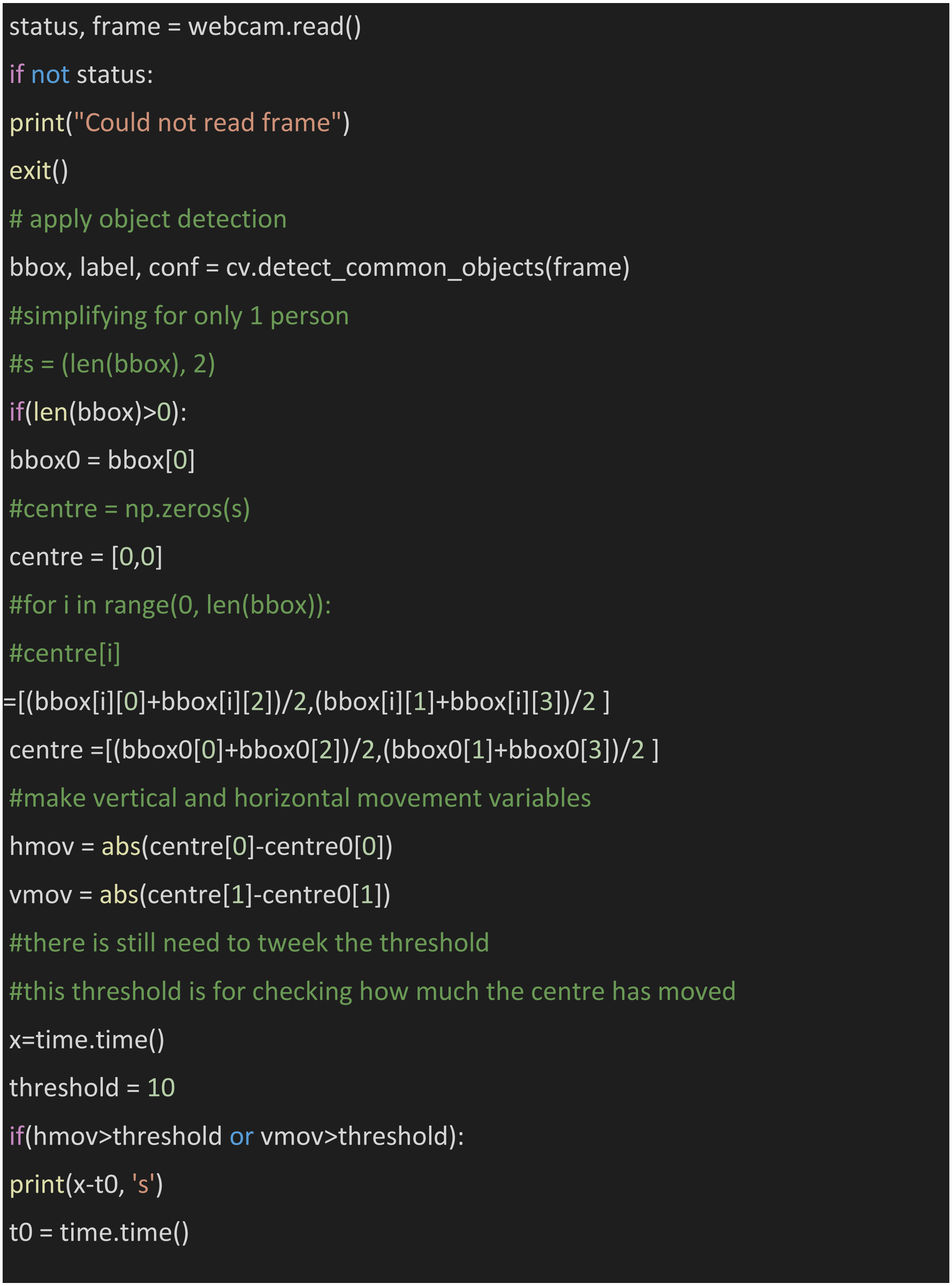
```
            #print('bounding box: ', bbox, 'label: ' label ,'confidence: ' conf[0], 'centre: ', centre)
            #print(bbox,label ,conf, centre)
            print('bbox: ', bbox, 'centre:', centre, 'centre0:', centre0)
            print('Is he drowning: ', isDrowning)
            centre0 = centre
            # draw bounding box over detected objects
        out = draw_bbox(frame, bbox, label, conf)
    #print('Seconds since last epoch: ', time.time()-t0)
    # display output
        cv2.imshow("Real-time object detection", out)
        if(isDrowning == True):
            playsound('alarm.mp3')
            webcam.release()
            cv2.destroyAllWindows()
            return render_template('prediction.html',prediction="Emergency !!!
```



```
The Person is drowining")
   #return render_template('base.html')
   # press "Q" to stop
      if cv2.waitKey(1) & 0xFF == ord('q'):
         break
   # release resources
   webcam.release()
   cv2.destroyAllWindows()
 #return render_template('prediction.html',)
""" Running our application """ if__name__
== "__main__":
   app.run(debug=True)
```

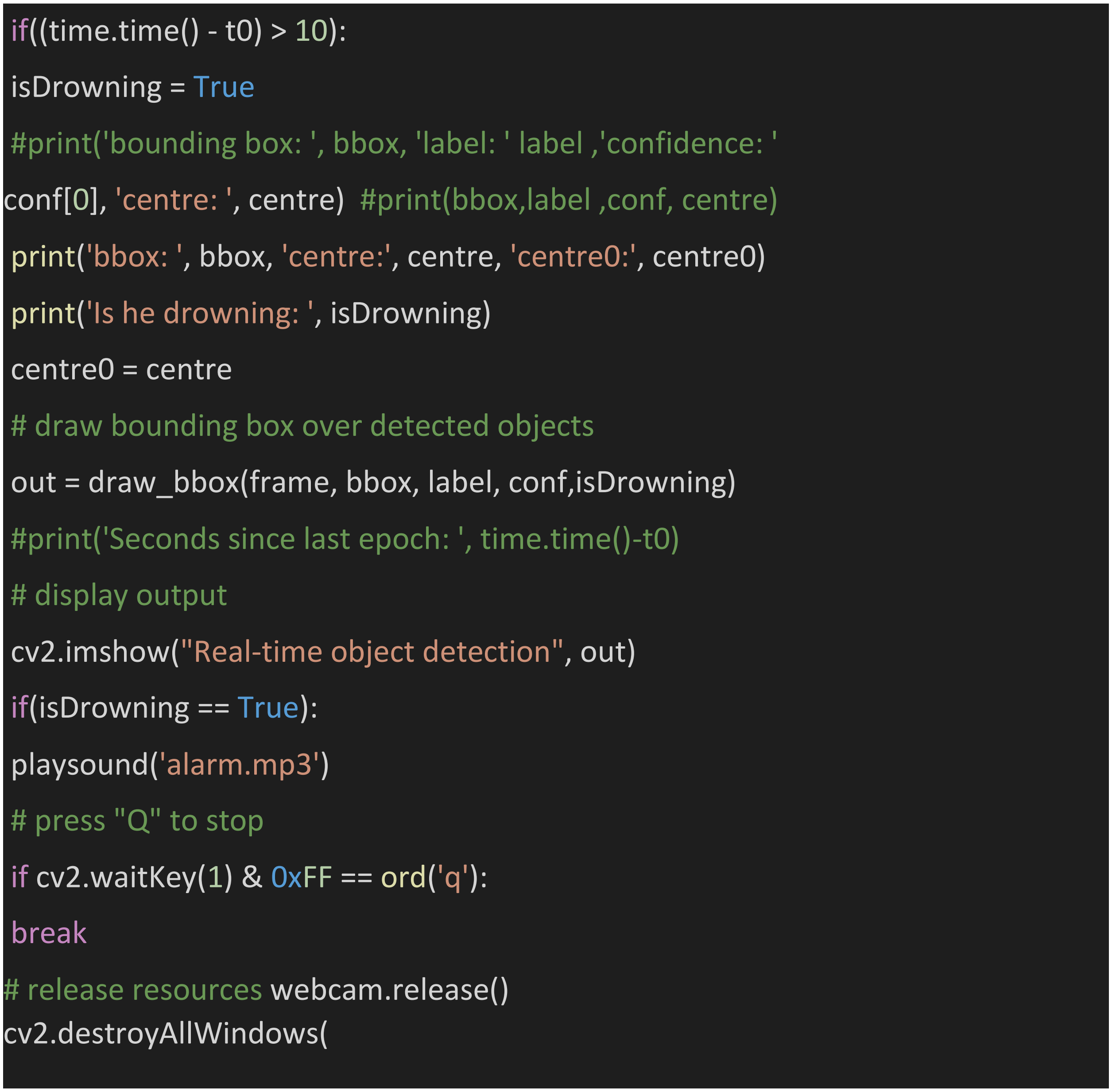
detect.py

```
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) ifnot
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(
```

CHAPTER 8

TESTING

As the code is made to run, the system waits to connect with Database. On account of connection with the IBM Watson Platform, the code displays the output with relevant details. The output is shown in Cloud platform, the links to Node-Red also to the UI section. Finally when the it is operated, the output is also displayed in it

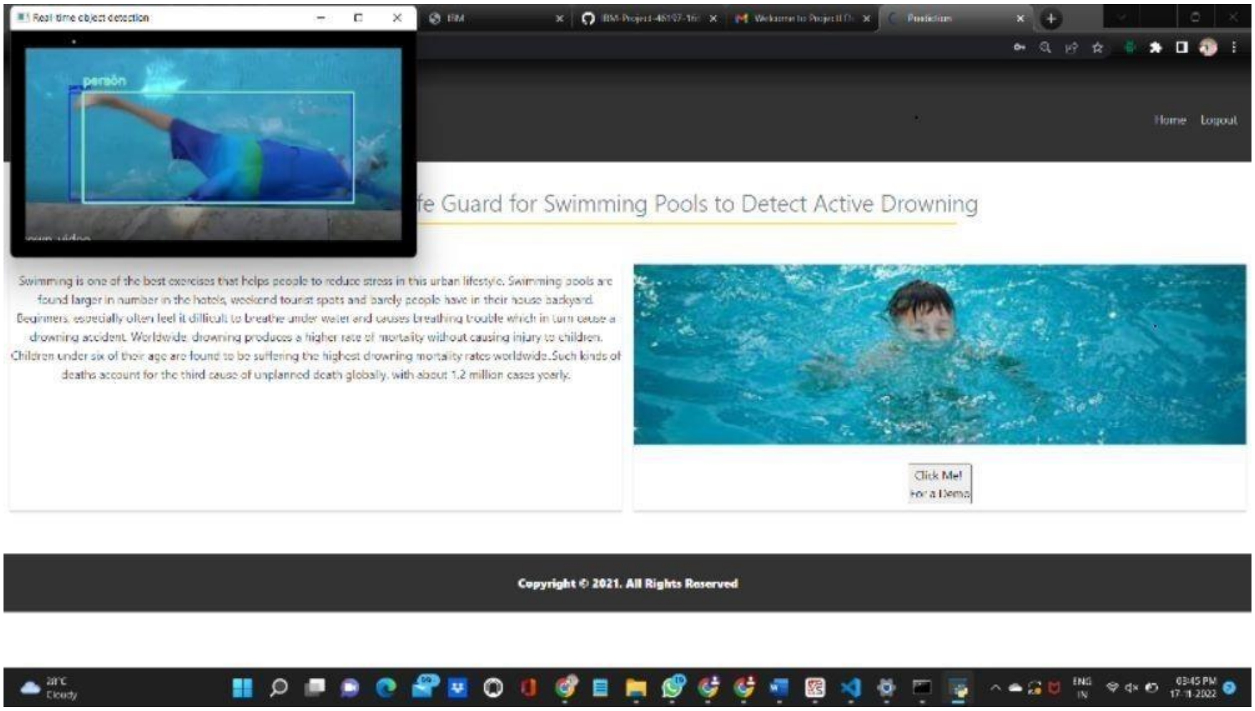
S. No.	Parameter	Values	Screenshot
1.	Model Summary	-	 Virtual Eye Virtual Eye Life Guard for Swimming Pools to Detect Active Drowning
2.	Accuracy	Training Accuracy - 28 Validation Accuracy -44	

CHAPTER 9

RESULTS

9.1 PERFORMANCE METRICS:

The performance and the working of the code is very quick and the results appears in quick succession.



fe 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. Swimming pools are found larger in number in the hotels, weekend tourist spots and barely people have in their house backyard. Beginners, especially often feel it difficult to breathe under water and causes breathing trouble which in turn cause 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.

Copyright © 2021. All Rights Reserved

Fig 12. Detection Output

CHAPTER 10
ADVANTAGES & DISADVANTAGES

ADVANTAGES:

- The use of an automated visual-based monitoring system can help to reduce drownings and assure pool safety effectively.
- It introduces a revolutionary technology that identifies drowning victims in a minimum amount of time and dispatches an automated drone to save them.
- Works like an “extra lifeguard” under the water of your pool.
- In the event of a serious drowning incident, it will provide an alarm to pool lifeguards. This will help lifeguards improve their reaction-time, as they initiate a rescue.
- As we don't need to wear them no risk in losing the protection.

DISADVANTAGES:

The disadvantages of choosing a underground alarm are that they are more permanent, need a strong edge, deck, or railing to attach to, and have a higher price point than other pool alarms.

CHAPTER 11

CONCLUSION

Consistently numerous people, including kids, are suffocated or near suffocating in the deeps of the swimming pools, and the lifeguards are not prepared all around to deal with these issues. It can be installed in

International standardized schools where classes are held for training kids.

The future research plans include improving the underwater communication range by using various other technologies. This will enable to use this system in seas also. The alarm receivers can be easily connected to the buoy With the help of establishing a standard communication protocol, we will be able to communicate more information to the lifeguards, as the name of the victim etc. This will help the lifeguard to search for his previous medical records as does the patient had any heart or lungs diseases etc. This information will provide an additional advantage while doing the rescue operation and while doing first aid. We also have plans to integrate a Global Positioning System (GPS) and pressure sensor. As pressure increases with the depth, the pressure reading will let the lifeguard know the depth at which victim is located. The GPS reading will be saved whenever the signal is available. If the system can tell the last previous GPS reading that was stored, it will enable the lifeguard to know what the approximate location of the victim. This feature will be very timesaving for lifeguards reducing their search time to near locations especially in case of lakes and oceans.

CHAPTER 12

FUTURE SCOPE

Availability of better dataset, modern methodologies, and technologies with high computational power accompanied by high-quality surveillance cameras, will help to improve the accuracy of drowning detection & even can be used in adverse conditions.

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

CHAPTER 13

APPENDIX

SOURCE CODE:

As we successfully developed and programmed our python code, lets this be the final code of execution.

App.py
```
import time import cv2 import numpy as np from
cloudant.client import Cloudant from flask import Flask, request,
render_template, redirect, url_for from playsound import playsound
import cvlib as cv
from cvlib.object_detection import draw_bbox
# Loading the model
# Authenticate using an IAM API key client = Cloudant.iam(&#39;c7591b18-67f9-
4321-80df-2522e5683942-
bluemix&#39;,&#39;xPQROZL0yOiM7VyTbPJG4t4DxBu670yEpV7NlTBz25Z
N&#39;, connect=True)
# Create a database using an initialized client
my_database = client.create_database(&#39;my_database&#39;)
app=Flask(__name__) #default home page or route
@app.route(&#39;/&#39;) def index():
   return render_template(&#39;index.html&#39;)
@app.route(&#39;/index.html&#39;)     def
home():
   return render_template(&quot;index.html&quot;)
#registration page
@app.route(&#39;/register&#39;)     def
register():
   return render_template(&#39;register.html&#39;)
@app.route(&#39;/afterreg&#39;,    methods=[&#39;POST&#39;])    def
afterreg():
   x = [x for x in request.form.values()]
   print(x)
   data = {
   &#39;_id&#39;: x[1], # Setting _id is optional
   &#39;name&#39;: x[0],
   &#39;psw&#39;:x[2]
   }
   print(data)

   query = {&#39;_id&#39;: {&#39;$eq&#39;: data[&#39;_id&#39;]}}

   docs = my_database.get_query_result(query)
   print(docs)

   print(len(docs.all()))

   if(len(docs.all())==0):
       url = my_database.create_document(data)
   #response = requests.get(url)

                                                           return
render_template(&#39;register.html&#39;, pred=&quot;Registration Successful,
please login using your details&quot;)
   else:
       return    render_template(&#39;register.html&#39;,    pred=&quot;You    are
 already a member, please login using your details&quot;)
```

```
#login page
@app.route(&#39;/login&#39;)
def login():
    return render_template(&#39;login.html&#39;)
@app.route(&#39;/afterlogin&#39;,methods=[&#39;POST&#39;])
def afterlogin():
    user = request.form[&#39;_id&#39;]
    passw = request.form[&#39;psw&#39;]
    print(user,passw)

    query = {&#39;_id&#39;: {&#39;$eq&#39;: user}}

    docs = my_database.get_query_result(query)
    print(docs)

    print(len(docs.all()))



    if(len(docs.all())==0):
        return render_template(&#39;login.html&#39;, pred=&quot;The username is
not
found.&quot;)
    else:

if((user==docs[0][0][&#39;_id&#39;]  and passw==docs[0][0][&#39;psw&#39;])):
            return redirect(url_for(&#39;prediction&#39;))
        else:
            print(&#39;Invalid User&#39;)
```

```
@app.route(&#39;/logout&#39;)     def
logout():
   return render_template(&#39;logout.html&#39;)
@app.route(&#39;/prediction&#39;)     def
prediction():
   return render_template(&#39;prediction.html&#39;)

@app.route(&#39;/result&#39;,methods=[&quot;GET&quot;,&quot;POST&qu
ot;]) def res():
   webcam = cv2.VideoCapture(&#39;drowning.mp4&#39;)
   if not webcam.isOpened():
      print(&quot;Could not open   &quot;)
      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&#39;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(&quot;Could not read frame&quot;)
       exit()
 # apply object detection
      bbox, label, conf = cv.detect_common_objects(frame)
 #simplifying for only 1 person
 #s = (len(bbox), 2)
      if(len(bbox)&gt;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&gt;threshold or vmov&gt;threshold):
            print(x-t0, &#39;s&#39;)
            t0 = time.time()
            isDrowning = False

          else:
            print(x-t0, &#39;s&#39;)
            if((time.time() - t0) &gt; 10):
              isDrowning = True
            #print(&#39;bounding box: &#39;, bbox, &#39;label: &#39; label
,&#39;confidence:     &#39;     conf[0],
&#39;centre: &#39;, centre)
           #print(bbox,label ,conf, centre)
             print(&#39;bbox: &#39;, bbox, &#39;centre:&#39;, centre,
&#39;centre0:&#39;, centre0)
            print(&#39;Is he drowning: &#39;, isDrowning)
            centre0 = centre

            # draw bounding box over detected objects
        out = draw_bbox(frame, bbox, label, conf)
    #print(&#39;Seconds since last epoch: &#39;, time.time()-t0)
    # display output
        cv2.imshow(&quot;Real-time object detection&quot;, out)
        if(isDrowning == True):
            playsound(&#39;alarm.mp3&#39;)
            webcam.release()
            cv2.destroyAllWindows()
            return
render_template(&#39;prediction.html&#39;,prediction=&quot;Emergency !!!
The Person is drowining&quot;)
    #return render_template(&#39;base.html&#39;)
    # press &quot;Q&quot; to stop
        if cv2.waitKey(1) &amp; 0xFF == ord(&#39;q&#39;):
            break
    # release resources
    webcam.release()
    cv2.destroyAllWindows()
 #return render_template(&#39;prediction.html&#39;,)
```

```
&quot;&quot;&quot; Running our application &quot;&quot;&quot; if
__name__ == &quot;__main__&quot;:
   app.run(debug=True)
```

Detect.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(&quot;Could not open webcam&quot;)
 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&#39;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(&quot;Could not read frame&quot;)
exit()
# apply object detection
bbox, label, conf = cv.detect_common_objects(frame)
#simplifying for only 1 person
#s = (len(bbox), 2)
if(len(bbox)&gt;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&gt;threshold or vmov&gt;threshold):
print(x-t0, &#39;s&#39;)
t0 = time.time()
isDrowning = False
```

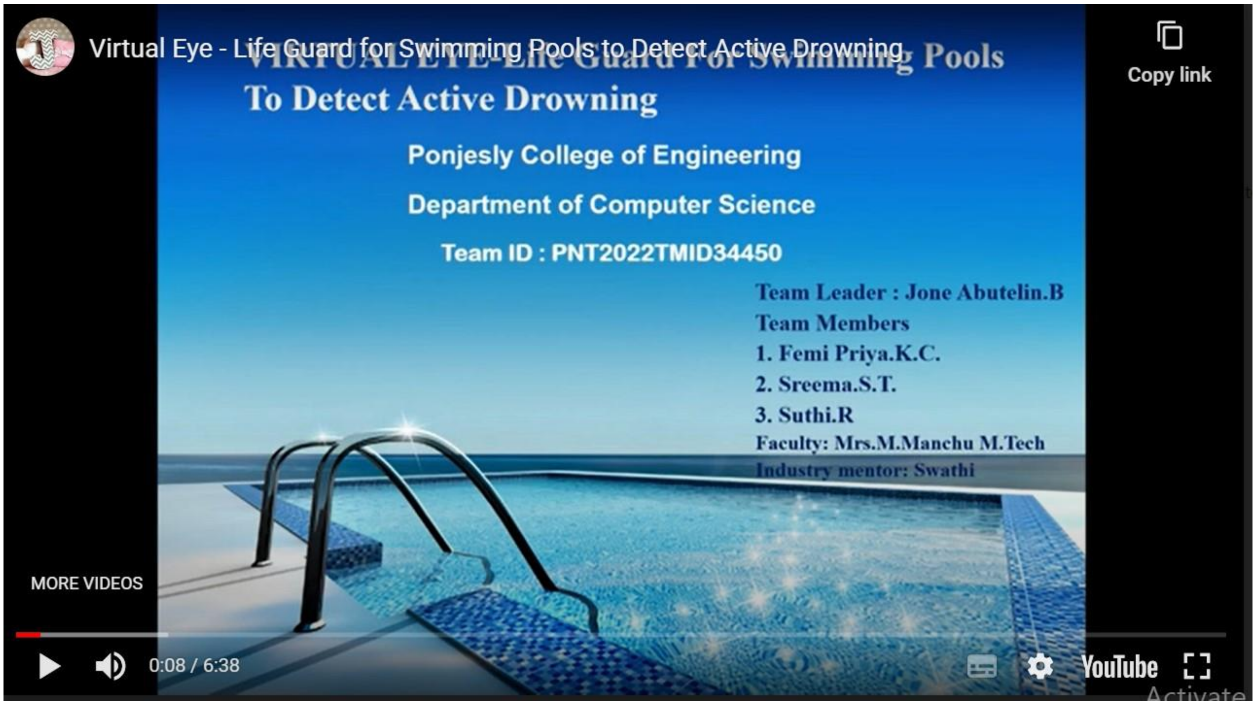
```
 else:
 print(x-t0, &#39;s&#39;)
 if((time.time() - t0) &gt; 10):
 isDrowning = True
    #print(&#39;bounding box: &#39;, bbox, &#39;label: &#39; label
,&#39;confidence: &#39;
conf[0], &#39;centre: &#39;, centre)
 #print(bbox,label ,conf, centre)
      print(&#39;bbox: &#39;, bbox, &#39;centre:&#39;, centre,
&#39;centre0:&#39;, centre0)
 print(&#39;Is he drowning: &#39;, isDrowning)
 centre0 = centre
 # draw bounding box over detected objects
 out = draw_bbox(frame, bbox, label, conf,isDrowning)
 #print(&#39;Seconds since last epoch: &#39;, time.time()-t0)
 # display output
 cv2.imshow(&quot;Real-time object detection&quot;, out)
 if(isDrowning == True):
 playsound(&#39;alarm.mp3&#39;)
 # press &quot;Q&quot; to stop

 if cv2.waitKey(1) &amp; 0xFF == ord(&#39;q&#39;):
 break
# release resources
webcam.release()
cv2.destroyWindows(
```

PROJECT DEMONSTARTION VIDEO UPLOADED HERE

GITHUB LINK: https://github.com/IBM-EPBL/IBM-Project-4619716607417760

PROJECT DEMO LINK:

https://www.youtube.com/embed/euxVed9i1n8



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

VIRTUAL EYE-Life Guard For Swimming Pools To Detect Active Drowning

Ponjesly College of Engineering

Department of Computer Science

Team ID : PNT2022TMID34450

Team Leader : Jone Abutelin.B
Team Members
1. Femi Priya.K.C.
2. Sreema.S.T.
3. Suthi.R
Faculty: Mrs.M.Manchu M.Tech
Industry mentor: Swathi