VIRTUALEYE-LIFE GUARD FOR SWIMMING POOLS TO DETECT ACTIVE DROWNING

TEAM MEMBER:

D.SHAVANI

V.SHRUTHI

S.SRUTHI

S.SUDHA SANGAVI

TEAM ID:PNT2022TMID28319

9. CONCLUSION

Source Code

GitHub & Project Demo Link

1. Introduction:

1.1 Project overview:-

The death from drowning has caused one third of untimely death in the world. This happens to small children and newbie swimmers to prevent this from happening there are various safety measures taken to prevent drowning in swimming pool. The spatial relationship between the location information of the target and swimming/drowning area of swimming pool is analyzed to further determine the swimmer's drowning or swimming behavior. This paper compares the detection accuracy of different detection algorithms and analyzes the detection effect of different pool angles and different swimmer densities.

1.2 Purpose

Drowning detection in dynamic swimming environments is a challenging problem in computer vision, for which no satisfiable solutions have been found. Currently known methods primarily rely on background subtraction-based techniques; however, random motion caused by water rippling, splashing, and moving reflections frequently result in interference and inaccuracies. It is mainly used to prevent drowning and rescuing the victim in golden hour.

2. Literature Survey

2.1 Existing problem

There are various drowning detection available nowadays, which helps people who are drowning but with these detection mechanism can cause hindrance while swimming. The newbie swimmers, children will find it difficult to swim and may cause drowning while swimming.

Swimming is one of the exercises done by modern people to relieve stress from their daily life. But the unplanned death from drowning is in the third place in the world. There is a need to find a solution to this problem. In Project - VirtualEye - Life Guard for Swimming Pools to Detect Active Drowning we find solution to this problem by detecting active drowning with the help of live feeds to alert the lifeguard. The novice swimmers, children find it hard to breath underwater and are not accustomed to swimming like veteran swimmers this causes a lot of drowning incident. Even if the lifeguard are on their toes it is easy to miss details of drowning. This causes us to lose our loved ones. In this system we detect the objects in the swimming pool with the help of cameras. The swimming pool is recorded with the cameras and the live feed is used to detect drowning and give alert to the lifeguard. This helps the lifeguard to take action as soon as he/she gets the alert. Here we use YOLO algorithm to train our model to identify the active drowning movements. For this purpose we train our model to detect objects and then to identify drowning movements with the help of images and videos which helps to identify drowning movements in real time. This system assures public to have a safe and secure time of swimming and help the lifeguard to save lives without any regrets.

2.2 Reference

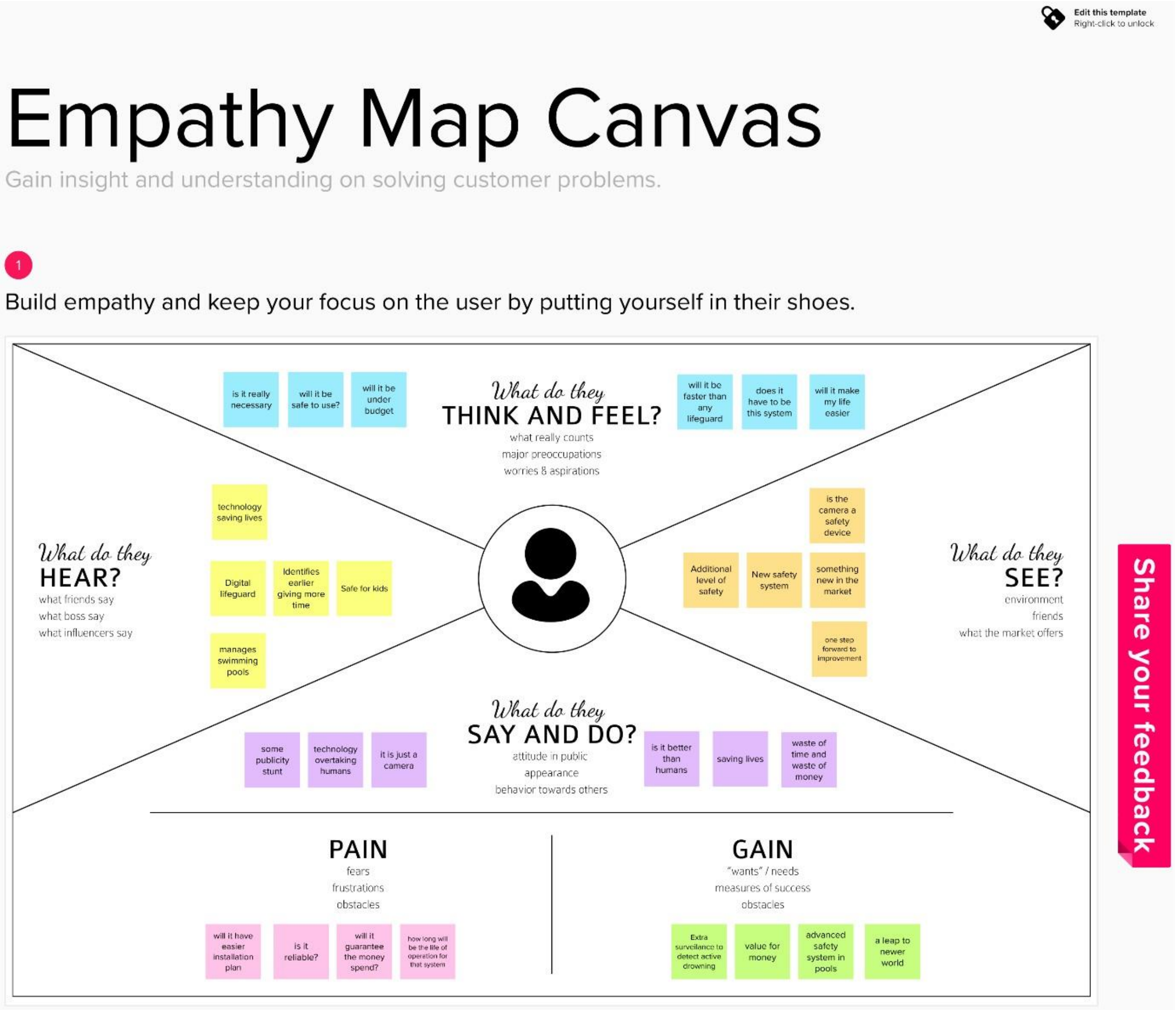
Project description from dashboard, https://www.thewirh.com/blog/dds-how-do-they-work Artificial Intelligence usecases, AngelEye, SwimEye

2.3 Problem Statement Definition

When a newbie swimmer, parent of a child wants a safe, hassle free environment and to stress about drowning and to have a pleasant environment during the swimming space it is necessary to use some preventive measures. This system uses the surveillance camera to detect whether the person is drowning or not.

3. Ideation and Proposed solution

3.1 Empathy Map Canvas



Edit this template
Right-click to unlock

Empathy Map Canvas

Gain insight and understanding on solving customer problems.

1

Build empathy and keep your focus on the user by putting yourself in their shoes.

What do they
THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations

- is it really necessary
- will it be safe to use?
- will it be under budget
- will it be faster than any lifeguard
- does it have to be this system
- will it make my life easier

What do they
HEAR?
what friends say
what boss say
what influencers say

- technology saving lives
- Digital lifeguard
- Identifies earlier giving more time
- Safe for kids
- manages swimming pools

What do they
SEE?
environment
friends
what the market offers

- is the camera a safety device
- Additional level of safety
- New safety system
- something new in the market
- one step forward to improvement

What do they
SAY AND DO?
attitude in public
appearance
behavior towards others

- some publicity stunt
- technology overtaking humans
- it is just a camera
- is it better than humans
- saving lives
- waste of time and waste of money

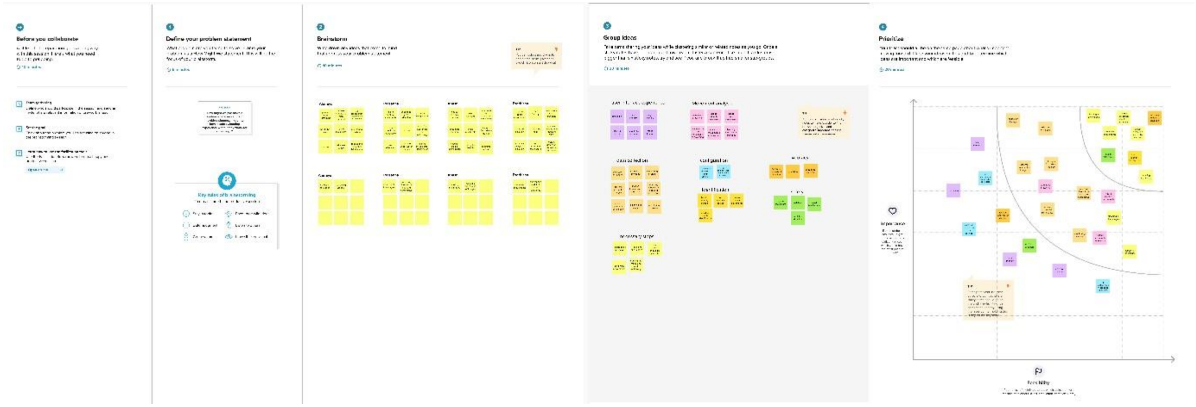
PAIN
fears
frustrations
obstacles

- will it have easier installation plan
- is it reliable?
- will it guarantee the money spend?
- how long will be the life of operation for that system

GAIN
"wants" / needs
measures of success
obstacles

- Extra surveilance to detect active drowning
- value for money
- advanced safety system in pools
- a leap to newer world

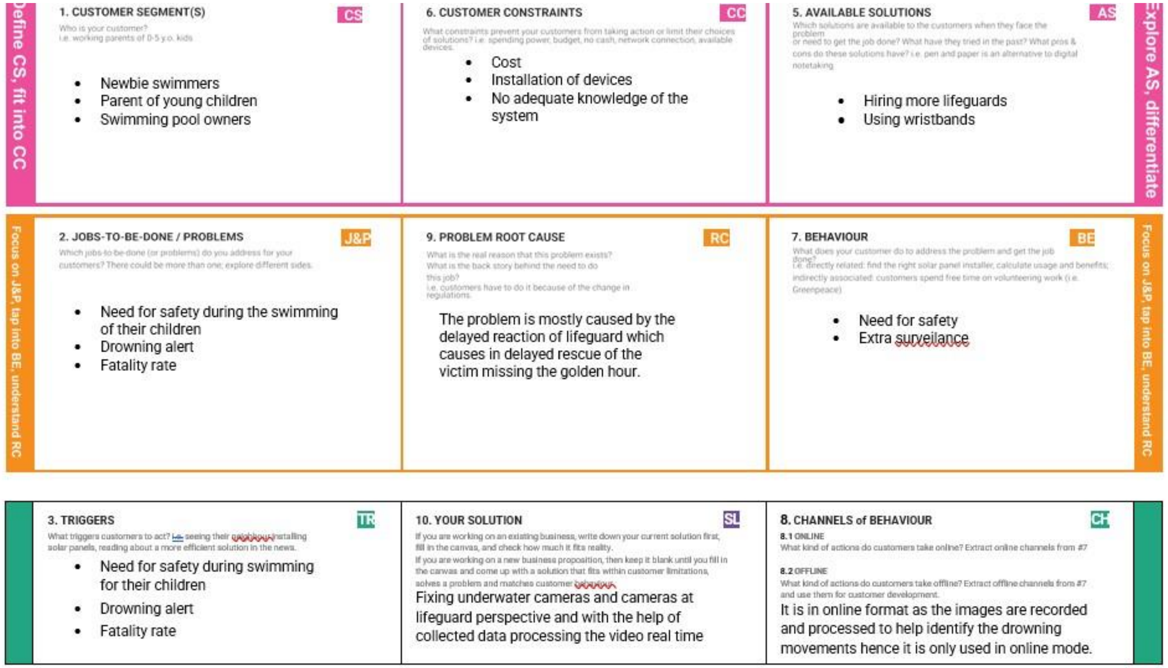
3.2 Ideation and Brainstorming



3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To prevent newbie swimmer or children to avoid drowning in swimming pools and to decrease the fatality rate.
2.	Idea / Solution description	To prevent drowning of swimmers in swimming pools here we will use cameras to record the pool and any movement of drowning will be captured by the cameras which will be processed to know whether the person is actually drowning or not. This will help the life guard to have extra pair of eyes and will help them to swiftly rescue the victim.
3.	Novelty / Uniqueness	This process uses the real time images and identifies drowning movements and alerting the lifeguard to not miss the golden hour of rescue.
4.	Social Impact / Customer Satisfaction	Helps in reducing the fear of drowning and gives assurance of being safe and sound while spending some quality time with their friends and families
5.	Business Model (Revenue Model)	It helps the lifeguard in reducing this dangerous events in happening to a considerable amount. It can also help in collaborating with the maritime sectors and swimming pool authorities.
6.	Scalability of the Solution	As it uses images to identify movements The camera can have blind spots which will affect the performance of the system

3.4 Problem Solution Fit



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?
i.e. working parents of 0-5 y.o. kids

- Newbie swimmers
- Parent of young children
- Swimming pool owners

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.

- Cost
- Installation of devices
- No adequate knowledge of the system

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

- Hiring more lifeguards
- Using wristbands

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.

- Need for safety during the swimming of their children
- Drowning alert
- Fatality rate

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 problem is mostly caused by the delayed reaction of lifeguard which causes in delayed rescue of the victim missing the golden hour.

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)

- Need for safety
- Extra surveilance

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

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.

- Need for safety during swimming for their children
- Drowning alert
- Fatality rate

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.

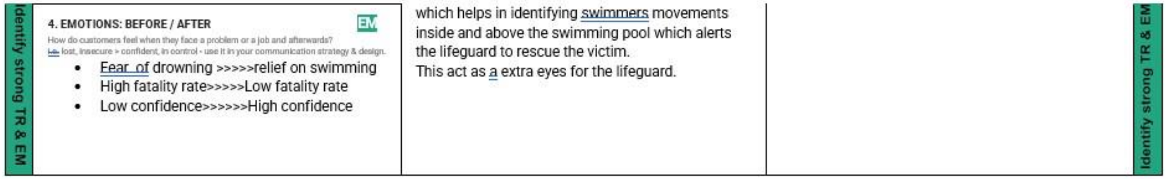
Fixing underwater cameras and cameras at lifeguard perspective and with the help of collected data processing the video real time

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.

It is in online format as the images are recorded and processed to help identify the drowning movements hence it is only used in online mode.



Identify strong TR & EM

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.

- Fear of drowning >>>>>relief on swimming
- High fatality rate>>>>>Low fatality rate
- Low confidence>>>>>>High confidence

which helps in identifying swimmers movements inside and above the swimming pool which alerts the lifeguard to rescue the victim.
This act as a extra eyes for the lifeguard.

Identify strong TR & EM

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 from above	Images of drowning from above the pool. Videos of drowning from above the pool.
FR-2	Under water camera	Images of drowning inside the pool. Videos of drowning inside the pool.
FR-3	Software requirements	Windows 11
FR-4	Machine learning software	Pytorch, Keras, Tensorflow
FR-5	Programming languages	Python, HTML, CSS

4.2 Non Functional requirements

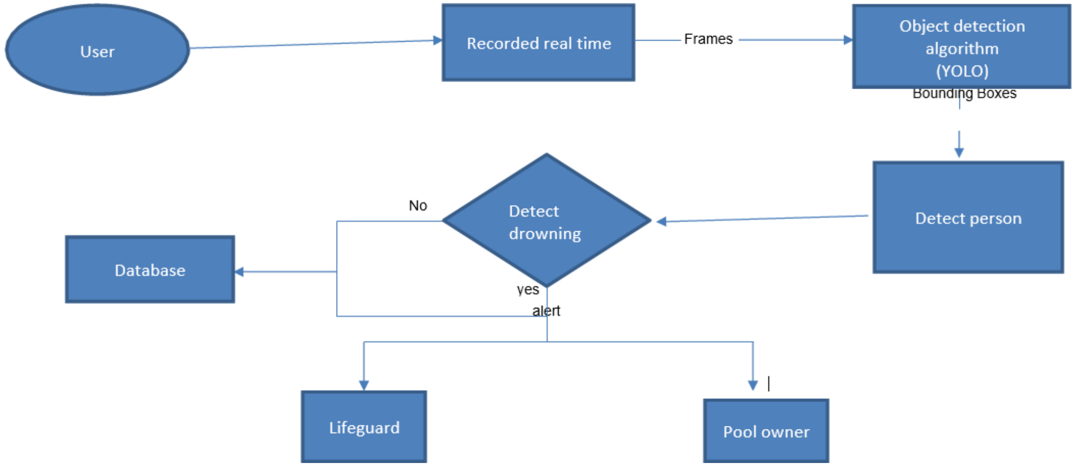
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	It can be used in public pools and swimming to alert the lifeguard indicating someone is drowning.
NFR-2	**Security**	As the rescue is done as soon as the alert is on it can help in saving life.
NFR-3	**Reliability**	It gives an extra pair of an eyes i.e., virtual eye to our lifeguard which helps him/her to detect drowning easily.
NFR-4	**Performance**	It is faster than naked eyes which helps in rescue of the victim without missing the golden hour.
NFR-5	**Availability**	It can be made available to swimming pool owners, and for public pools to avoid drowning.
NFR-6	**Scalability**	As it uses images to identify movements The camera can have blind spots which will affect the performance of the system

5. Project Design

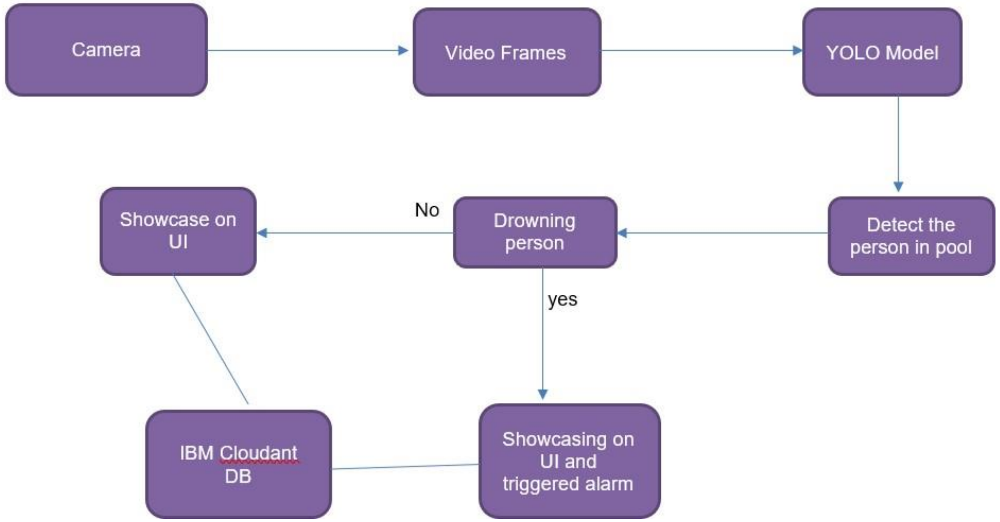
5.1 Data Flow Diagrams

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 rightamount 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 and Technical Architecture

Technical Architecture:



S.No	Component	Description	Technology
1.	User Interface	How user interacts with application	HTML, CSS, JavaScript
2.	Application Logic-1	Frames extraction from the live video	Python
3.	Application Logic-2	Detecting person	Python
4.	Application Logic-3	Drowning detection	Python
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL
6.	Cloud Database	Database Service on Cloud	IBM Cloudant
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	Machine learning Model	Detecting human beings	Object detection model(YOLOv7)

9.	Infrastructure (Server / Cloud)	Application Deployment on Cloud	Cloud Foundry, Docker

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Anaconda Navigator, Pytorch, Flask	Technology of Opensource framework
2.	Security Implementations	Security and access control	IAMControls
3.	Scalable Architecture	Scalable architecture can load without compromising the application integruty	Microservices, Progressive web apps
4.	Availability	Use ofload balancers, distributed servers	IBM Cloud
5.	Performance	Designing the system software that can monitor a wide range of swimming pool at a time without anny delay	IBM instance

5.3 User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (pool owners)	Installation of devices	USN-1	As the owner of the pool I can install cameras on my pool and can set the drowning detection system.	I can connect the camera to the database	High	Sprint-1
Customer (Lifeguard)	Detecting drowning	USN-2	As a user, I can detect if whether someone is drowning or not	I will receive an alert which notifies me.	High	Sprint-1
	Rescue	USN-3	As a user, on receiving the alert I can	I can rescue the drowning person	High	Sprint-1

			rescue the drowning victim			
Customer (Swimmer)	Safety	USN-4	As a user, I can swim without an worry.	I can swim with the assurance of the system and the lifeguard	Medium	Sprint-2
Customer Care Executive	Contact	USN-5	Technical issues are resolved	I can call the customer care executive to resolve the issues	Medium	Sprint-3
Administrator	Dashboard	USN-6	Drowning detection system management and Database management	I can access all the data in the system anytime	High	Sprint-4

6. Project planning and Scheduling

6.1 Sprint Schedule and Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Index	USN-1	As a user, I can view the home page of the document which gives me the information of the application	2	Low	D.Shivani V.Shruthi S.Sruthi S.Sudha Sangavi
Sprint-1	Registration	USN-2	As a user, I can register into the application using my email id and newly created password	2	Low	S.Shruthi
Sprint-1	Login	USN-3	As a user I can login into my existing account by giving my credentials	2	Low	S.Sudha Sangavi
Sprint-3	Detection	USN-4	As a user I can detect if some one is drowning or not	5	High	D.Shivani
Sprint-3	Alarm	USN-5	As a user I can hear the sound of the alarm which indicates someone is drowning	2	Low	S.Sruthi D.Shivani
Sprint-2	Prediction	USN-6	As a user I can save the person from drowning by taking swift action regarding the matter	3	Medium	S.Sudha Sangavi
Sprint-2	Logout	USN-7	As a user I can logout from the application when needed	2	Low	S.Shruthi

Sprint-4	Whole application	USN-8	As a user I can use the application efficiently	2	Low	D.Shivani

6.2 Sprint Delivery Schedule

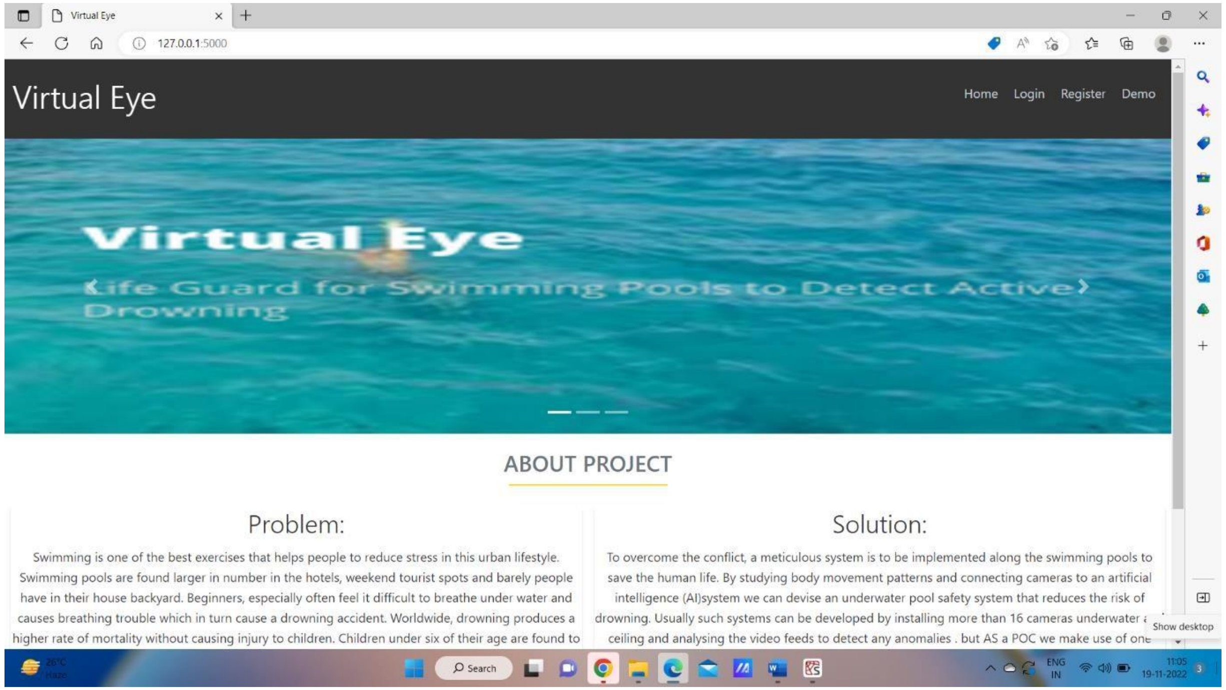
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date(Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date(Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022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

7. Coding and Solution

7.1 Html pages

Case1: Index page

In this page we will see the home page of our website where there will be options to register into the application, login into the application, see the prediction of the application and know about our application.



ABOUT PROJECT

Problem:

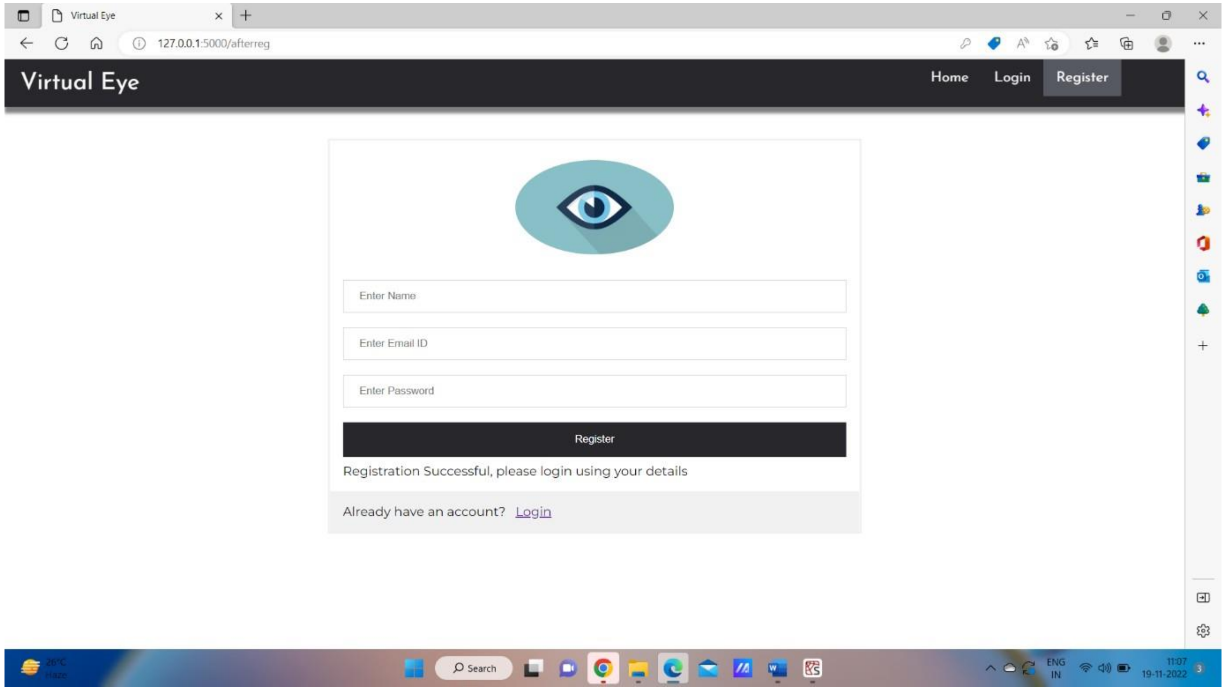
Swimming is one of the best exercises that helps people to reduce stress in this urban lifestyle. Swimming pools are found larger in number in the hotels, weekend tourist spots and barely people have in their house backyard. Beginners, especially often feel it difficult to breathe under water and causes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

Solution:

To overcome the conflict, a meticulous system is to be implemented along the swimming pools to save the human life. By studying body movement patterns and connecting cameras to an artificial intelligence (AI)system 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 a ceiling and analysing the video feeds to detect any anomalies . but AS a POC we make use of one

Case2: Register page

We can register into the account using this page, by using our credentials we can register our account in the application.

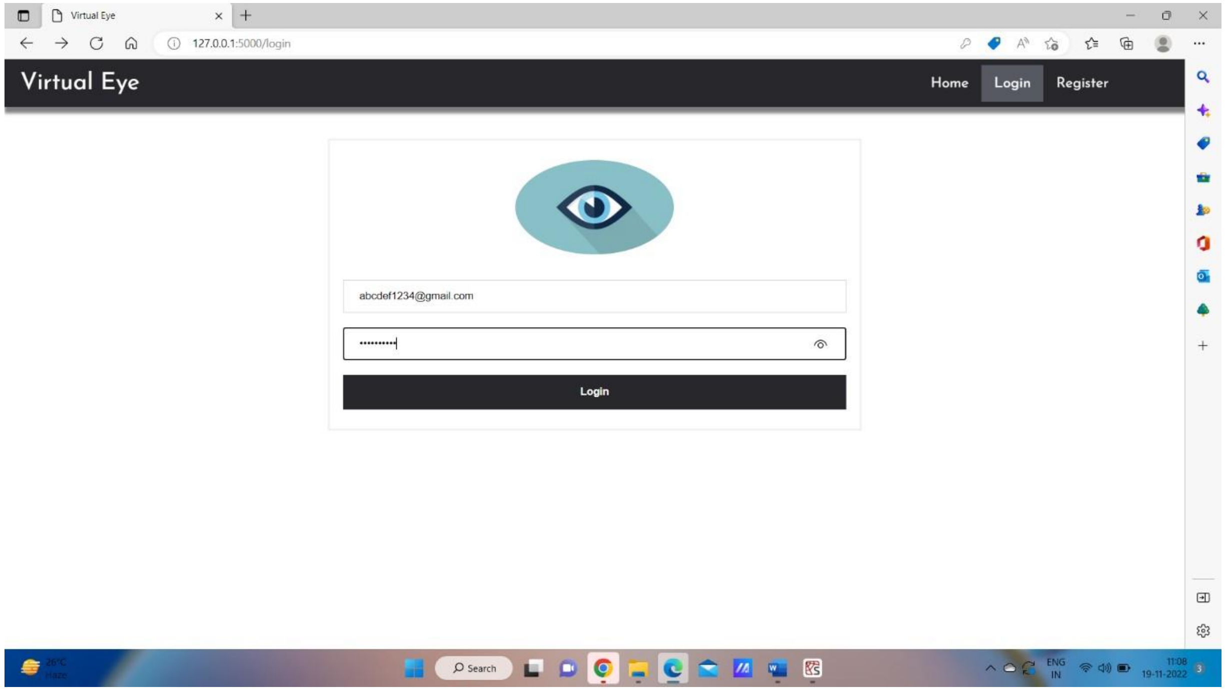


Registration Successful, please login using your details

Already have an account? Login

Case3: Login page

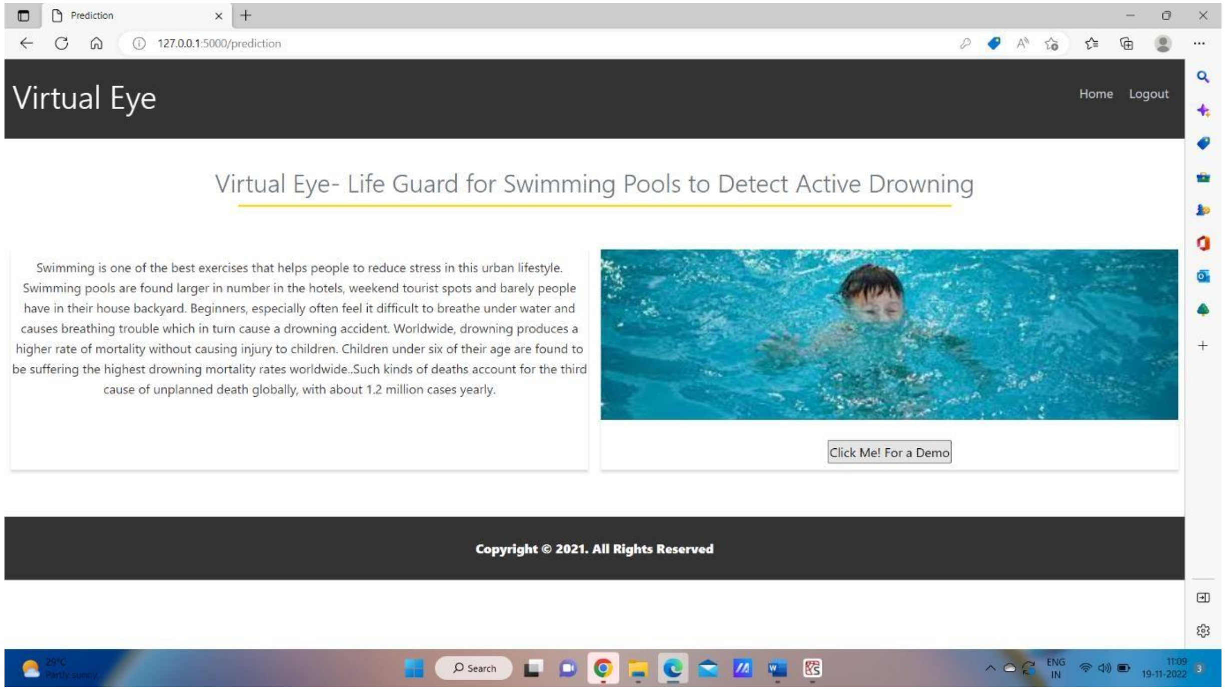
In this page with the credentials we used to register we can login into our account to try the demo of our project



abcdef1234@gmail.com

Case4: Prediction page

This page helps in showcasing our demo of the project. On clicking the button for demo we can get the results in the project.

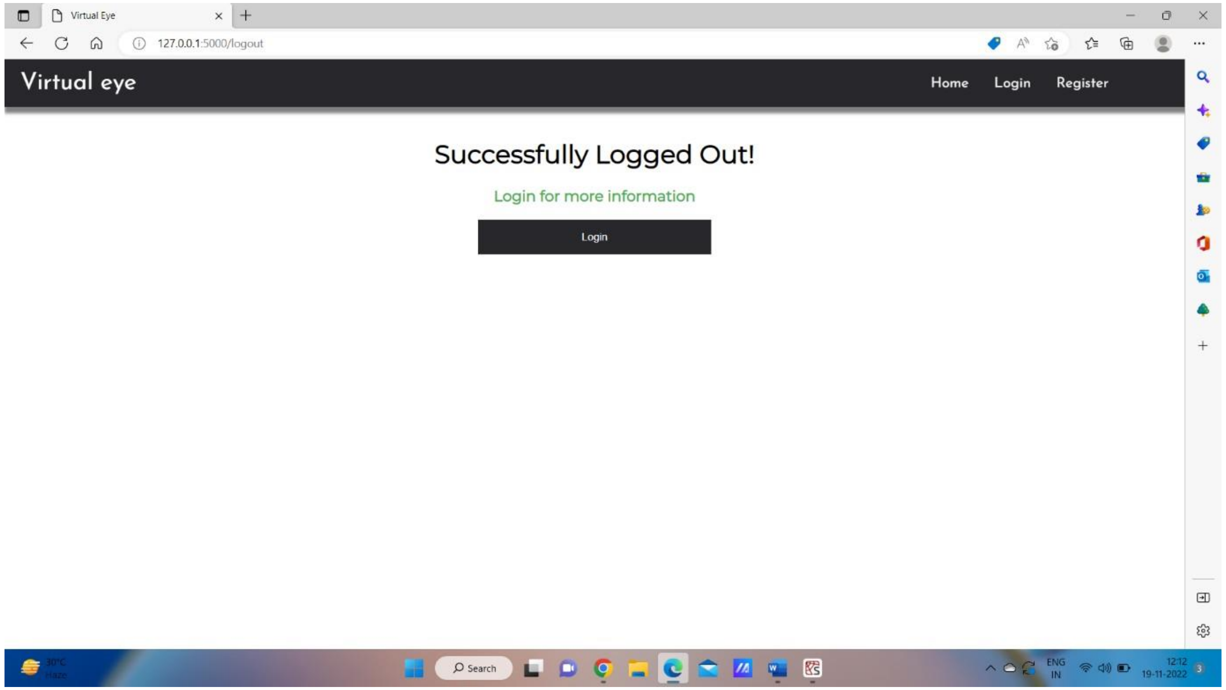


Virtual Eye- Life Guard for Swimming Pools to Detect Active Drowning

Swimming is one of the best exercises that helps people to reduce stress in this urban lifestyle. 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.

Case5: Logout page

In this page we can come out of our application by safely logging off our account.



Successfully Logged Out!

Login for more information

8. Testing

8.1 User Acceptance testing

1. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and howthey were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	4	5	1	2	12
Duplicate	2	0	2	0	3

External	4	3	1	1	9
Fixed	15	4	3	23	45
Not Reproduced	0	0	0	0	0
Skipped	0	0	0	0	0
Won't Fix	0	1	0	0	1
Totals	25	13	7	26	70

2. Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	10	0	0	10
Client Application	15	0	0	15
Security	2	0	0	2
Outsource Shipping	0	0	0	0

9. Conclusion

This system allows us to use the swimming pools without any worry and helps the lifeguard with an extra pair of eyes so that he/she can rescue people in a given time.

Source code:

final.css

```
.img-preview {
    width: 256px;
    height: 256px;
    position: relative;
    border: 5px solid #F8F8F8;
    box-shadow: 0px 2px 4px 0px rgba(0, 0, 0, 0.1);
    margin-top: 1em;
    margin-bottom: 1em;
}

.img-preview>div {
    width: 100%;
    height: 100%;
    background-size: 256px 256px;
    background-repeat: no-repeat;
    background-position: center;
}

input[type="file"] {
    display: none;
}

.upload-label{
    display: inline-block;
    padding: 12px 30px;
    background: #28272c;
    color: #fff;
    font-size: 1em;
    transition: all .4s;
    cursor: pointer;
}

.upload-label:hover{
    background: #C2C5A8;
    color: #39D2B4;
}

.loader {
    border: 8px solid #f3f3f3; /* Light grey */
    border-top: 8px solid #28272c; /* Blue */
    border-radius: 50%;
    width:  50px;
    height: 50px;
    animation: spin 1s linear infinite;
}

@keyframes spin {
    0% { transform: rotate(0deg); }
    100% { transform: rotate(360deg); }
}
```

Jscript.js

```
'use strict'
const demo = document.querySelector('#demo');
const imageUpload = document.getElementById('imageupload');
const dataAttributeEL = document.querySelectorAll(`div[data-type]`);
const displayAll = function () {
    dataAttributeEL.forEach(el => {
        el.classList.remove('hidden')
    })
}



imageUpload.addEventListener('change', (event) => {
    const fileList = event.target.files[0];

    //console.log(URL.createObjectURL(fileList));
    if (fileList) {
        demo.src =URL.createObjectURL(fileList);
    }
    displayAll();

});
```

```
const prediction = document.querySelector('#result')
dataAttributeEL.forEach(el => {
    if (el.dataset.type !== prediction.innerHTML.trim()) {
        el.classList.add('hidden')
    };
})
```

main.js

```
$(document).ready(function () {
    // Init
    $('.image-section').hide();
    $('.loader').hide();
    $('#result').hide();

    // Upload Preview
    function readURL(input) {
        if (input.files && input.files[0]) {
            var reader = new FileReader();
            reader.onload = function (e) {
                $('#imagePreview').css('background-image', 'url(' + e.target.result + ')');
                $('#imagePreview').hide();
                $('#imagePreview').fadeIn(650);
            }
            reader.readAsDataURL(input.files[0]);
        }
    }
    $("#imageUpload").change(function () {
        $('.image-section').show();
        $('#btn-predict').show();
        $('#result').text('');
        $('#result').hide();
        readURL(this);
    });

    // Predict
    $('#btn-predict').click(function () {
        var form_data = new FormData($('#upload-file')[0]);

        // Show loading animation
        $(this).hide();
        $('.loader').show();

        // Make prediction by calling api /predict
        $.ajax({
            type: 'POST',
            url: '/predict',
            data: form_data,
            contentType: false,
            cache: false,
            processData: false,
            async: true,
            success: function (data) {
                // Get and display the result
                $('.loader').hide();
                $('#result').fadeIn(600);
                $('#result').text('Prediction: '+data);
                console.log('Success!');
            },
        });
    });

});
```

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!--Bootstrap -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">
    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl" crossorigin="anonymous"></script>



    <script src="https://kit.fontawesome.com/8b9cdc2059.js" crossorigin="anonymous"></script>
    <link href="https://fonts.googleapis.com/css2?family=Akronim&family=Roboto&display=swap" rel="stylesheet">
    <link rel="stylesheet" href="../static/style.css">
    <!-- <script defer src="../static/js/main.js"></script> -->
    <title>Virtual Eye</title>
</head>
<body>
    <header id="head" class="header">
   <section  id="navbar">
          <h1 class="nav-heading"></i>Virtual Eye</h1>
       <div class="nav--items">
          <ul>
```

```
                                        <li><a  href="{{ url_for('index')}}">Home</a></li>
                                        <li><a href="{{ url_for('login')}}">Login</a></li>
                                        <li><a href="{{ url_for('register')}}">Register</a></li>
                    <li><a href="{{ url_for('login')}}">Demo</a></li>
                  </ul>
              </div>
      </section>
      <section id="slider">
       <div id="carouselExampleIndicators" class="carousel" data-ride="carousel">
           <ol class="carousel-indicators ">
               <li data-target="#carouselExampleIndicators" data-slide-to="0" class="active "></li>
               <li data-target="#carouselExampleIndicators"  data-slide-to="1"></li>
               <li data-target="#carouselExampleIndicators"  data-slide-to="2"></li>
           </ol>
           <div class="carousel-inner">

               <div class="carousel-item active">
                   <img class="d-block w-100" src="../static/img/1.png" alt="First slide">
               </div>
               <div class="carousel-item">
                   <img class="d-block w-100" src="../static/img/second.jpg" alt="Second slide">
               </div>
               <div class="carousel-item">
```

```
                <img class="d-block w-100" src="../static/img/third.jpg" alt="Third slide">
            </div>
        </div>
        <a class="carousel-control-prev" href="#carouselExampleIndicators" role="button" data-slide="prev">
            <span class="carousel-control-prev-icon" aria-hidden="true"></span>
            <span class="sr-only">Previous</span>
        </a>
        <a class="carousel-control-next" href="#carouselExampleIndicators" role="button" data-slide="next">
            <span class="carousel-control-next-icon" aria-hidden="true"></span>
            <span class="sr-only">Next</span>
        </a>
    </div>

   </section>
</header>
<section id="about">
    <div class="top">
        <h3 class="title text-muted">
            ABOUT PROJECT
        </h3>
        <div class="line"></div>
    </div>
<div class="body">
```

```
<div class="left">
    <h2>Problem:</h2>
    <p>
       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.


                        </p>
</div>
<div class="left">
    <h2>Solution:</h2>
    <p>
        To overcome the conflict, a meticulous system is to be implemented along the swimming pools to save the human life. By studying body movement
patterns and connecting cameras to an artificial intelligence (AI)system 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 analysing 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 than an alert will be generated to attract lifeguards attention.


    </p>
</div>
</div>
<div class="bottom">
        <p ><b>
```

Note : The system is not designed to replace a lifeguard or other human monitor, but to act as an additional tool. “It helps the lifeguard to detect the underwater situation where they can’t easily observe.

```
        </b></p>


    </div>
</section>


<section id="footer">
    <p>Copyright © 2021. All Rights Reserved</p>
    <div class="social">
      <a href="#" target="_blank"><i class="fab fa-2x fa-twitter-square"></i></a>
       <a href="#" target="_blank">
        <i class="fab fa-2x  fa-linkedin"></i></a>
        <a href="#">
            <i class="#"></i>
        </a>
    </div>
</section>
</body>
</html>
```

Login.html

```
<!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 rel="stylesheet" href="{{ url_for('static', filename='css/style.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;
  }
.login{
          margin-top:100px;
}
form {border: 3px solid #f1f1f1; 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: 100%;
  font-weight:bold;
}

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('index')}}">Home</a>

    <a class="active" href="{{ url_for('login')}}">Login</a>

    <a  href="{{ url_for('register')}}">Register</a>


  </div>

</div>

<div id="login" class="login">




            <form action="{{url_for('afterlogin')}}" method="post">

                        <div class="imgcontainer">

                                    <img      style=""      src="https://cdn.digitalhealth.net/wp-content/uploads/2017/03/eye_image_generic_555.jpg"      alt="Avatar"
class="avatar">

                        </div>
```

```
                <div class="container">
                        <input type="email" placeholder="Enter registered email ID" name="_id" required><br>


                        <input type="password" placeholder="Enter Password" name="psw" required>


                        <button type="submit">Login</button><br>
    {{pred}}
                </div>

        </form>

</div>



</body>
</html>
```

Register.html

```
<!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 rel="stylesheet" href="{{ url_for('static', filename='css/style.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;
  }
.login{
        margin-top:100px;
}
form {border: 3px solid #f1f1f1; 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: 100%;
}
```

```
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" >

    <a  href="{{ url_for('home')}}">Home</a>
    <a href="{{ url_for('login')}}">Login</a>
    <a class="active"  href="{{ url_for('register')}}">Register</a>

  </div>
</div>
<div id="login" class="login">



            <form action="{{url_for('afterreg')}}" method="post">
                        <div class="imgcontainer">
                                    <img style="" src="https://cdn.digitalhealth.net/wp-content/uploads/2017/03/eye_image_generic_555.jpg" alt="Avatar"
class="avatar">
                        </div>

                        <div class="container">
                                    <input type="text" placeholder="Enter Name" name="name" required><br>
                                    <input type="email" placeholder="Enter Email ID" name="_id" required><br>
                                    <input type="password" placeholder="Enter Password" name="psw" required>
                        <button type="submit">Register</button><br>
   {{pred}}
                </div>

                <div class="container" style="background-color:#f1f1f1">
   <div class="psw">Already have an account?&nbsp; &nbsp;<a href="{{ url_for('login') }}">Login</a></div  >
  </div>
        </form>


</div>




</body>
</html>
```

Prediction.html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <!--Bootstrap -->
```

```
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>

    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>

    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl" crossorigin="anonymous"></script>



    <script src="https://kit.fontawesome.com/8b9cdc2059.js" crossorigin="anonymous"></script>

    <link href="https://fonts.googleapis.com/css2?family=Akronim&family=Roboto&display=swap" rel="stylesheet">

    <link rel="stylesheet" href="../static/style.css">



    <script defer src="../static/js/JScript.js"></script>

    <title>Prediction</title>

</head>

<body>

    <header id="head" class="header">

        <section  id="navbar">

                <h1 class="nav-heading"></i>Virtual Eye</h1>

            <div class="nav--items">

                <ul>
```

```
               <li><a  href="{{ url_for('index')}}">Home</a></li>
                                    <li><a href="{{ url_for('logout')}}">Logout</a></li>
           <!-- <li><a href="#about">About</a></li>
           <li><a href="#services">Services</a></li> -->

            </ul>
         </div>
      </section>
   </header>
   <!-- dataset/Training/metal/metal326.jpg -->
   </br>
   <section id="prediction">
   <h2 class="title text-muted">Virtual Eye- Life Guard for Swimming Pools to Detect Active Drowning</h1>
   <div class="line" style="width: 900px;"></div>
                  </section>
                  </br>
         <section id="about">

<div class="body">
<div class="left">
   <p>
```

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.

```
            </p>
</div>
<div class="left">

  <div class="prediction-input">
    <img class="d-block w-100" src="../static/img/second.jpg" alt="Second slide">
    </br>
        <form id="form" action="/result" method="post"  enctype="multipart/form-data">

          <input type="submit" class="submitbtn" value="Click Me! For a Demo">
         </form>
      </div>
      <h5 style="text-color:Red">
      <b style="text-color:Red">{{prediction}}<b>
     </h5>
</div>
</div>
</section>

   </br></br>
```

```
    <section id="footer">
        <p>Copyright © 2021. All Rights Reserved</p>

    </section>
</body>

</html>
```

Logout.html

```
<!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>
```

App.py

```
import re
import numpy as np
import os
from flask import Flask, app,request,render_template
from tensorflow.keras import models
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
from tensorflow.python.ops.gen_array_ops import concat
from tensorflow.keras.applications.inception_v3 import preprocess_input
import cvlib as cv
from cvlib.object_detection import draw_bbox
import cv2
import time
import numpy as np
from playsound import playsound
import requests
from flask import Flask, request, render_template, redirect, url_for
#Loading the model


from cloudant.client import Cloudant


# Authenticate using an IAM API key
client = Cloudant.iam("76128e4e-285c-4ba2-b72c-0f3167cb8b12-bluemix","KTrf_y6V11tOnTPw8_2fXQmR2sRSae-Um-1NrKZ5xGH9", 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('/')
def index():
    return render_template('index.html')

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


#registration page
@app.route('/register')
def register():
    return render_template('register.html')

@app.route('/afterreg', methods=['POST'])
def afterreg():
    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 Successful, please login using your details")
    else:
        return render_template('register.html', pred="You are already a member, please login using your details")

#login page
@app.route('/login')
def login():
    return render_template('login.html')
```

```
@app.route('/afterlogin',methods=['POST'])
def afterlogin():
    user = request.form['_id']
    passw = request.form['psw']
    print(user,passw)

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

    docs = my_database.get_query_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')
def logout():
    return render_template('logout.html')

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



@app.route('/result',methods=["GET","POST"])
def result():
    webcam = cv2.VideoCapture("drowning.mp4")

    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")
    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=False)
```

Github and Project demo link

Github:- https://github.com/IBM-EPBL/IBM-Project-49078-1660815719
Project Demo link:- https://youtu.be/2Szm3_5cWtY