

Project Report

Team ID	PNT2022TMID51022
Project Name	Virtual Eye - Life Guard-for Swimming Pools to Detect Active Drowning

1.Introduction

1.1 Project Overview

Swimming is one of the best exercises that helps people to reduce stress in this urban lifestyle. Swimming pools are found larger in number in hotels, and weekend tourist spots and barely people have them in their house backyard. Beginners, especially, often feel it difficult to breathe underwater which causes breathing trouble which in turn causes a drowning accident. Worldwide, drowning produces a higher rate of mortality without causing injury to children. Children under six of their age are found to be suffering the highest drowning mortality rates worldwide. Such kinds of deaths account for the third cause of unplanned death globally, with about 1.2 million cases yearly. To overcome this conflict, a meticulous system is to be implemented along the swimming pools to save human life.

1.2 Purpose

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.

2.literarure survey

2.1 Existing Problem

The accuracy of some visual searches is crucial (e.g., in radiology and airport security screening), so it's crucial to know what variables most effectively predict performance for theoretical and practical reasons. In order to evaluate group differences in which criteria predict accuracy, the current study gave a visual search task to both professional (Transportation Security Administration Officers) and nonprofessional (members of Duke University) searchers. For nonprofessional searchers (who accounted for 59% of their accuracy-variability) and the least experienced professional searchers (37% of variability), search speed*time taken to terminate search*was the main predictor. For the most skilled professional visual searchers, consistency—or how consistently (in terms of search speed) an individual spent searching from trial to trial—was the main predictor (39% of variability).

2.2 Problem Definition

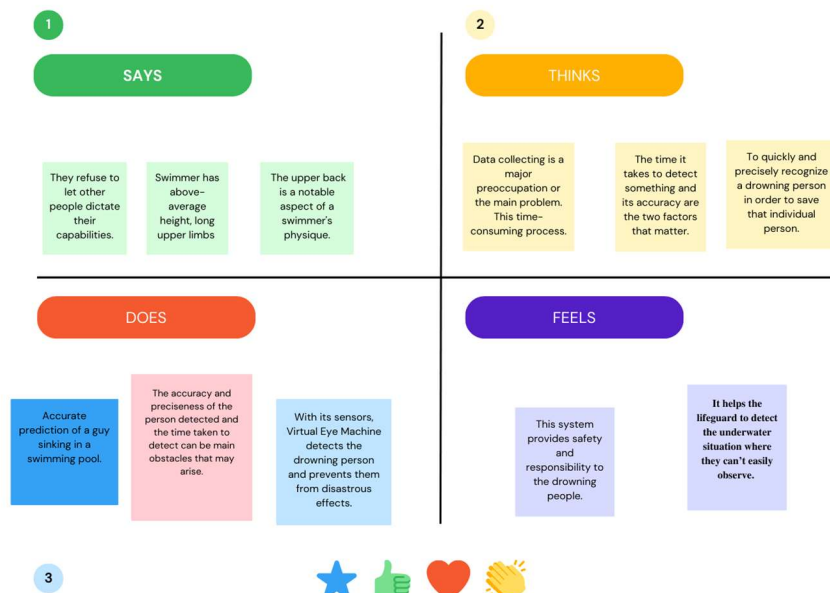
In this research, the problem statement focuses on the point of active drowning happening in swimming pools due to un forcing situations. Beginners, in particular, frequently find it challenging to breathe underwater, which results in respiratory issues and ultimately, a drowning disaster. Thereby, the scope of this research would be coming up with an AI assistant - Virtual Eye. Virtual Eye is an underwater pool safety system that lowers the chance of drowning by analyzing body movement patterns and integrating cameras with Artificial Intelligence in-built systems. Thereby, Virtual Eye helps in minimizing the unpredictable situations due to drowning.

3.Ideation

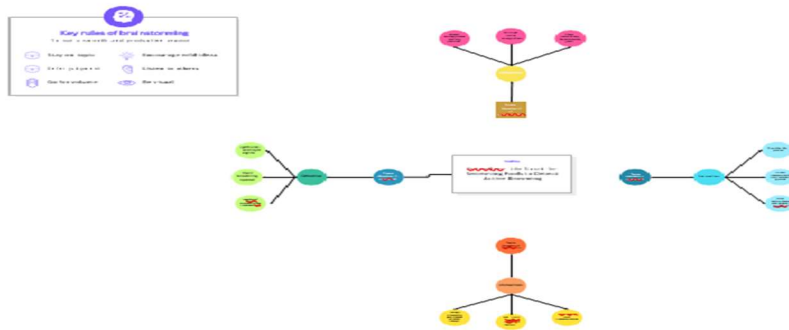
3.1 Empathy Map

EMPATHY MAP

 VIRTUALEYE - LIFE GUARD FOR SWIMMING POOLS TO DETECT ACTIVE DROWNING



3.2 Brain Storming



3.3 Proposed Solution

S.No	Parameter	Description
1.	Problem Statement (Problem to be solved)	<ul style="list-style-type: none"> ● Although swimming pools are often places to have fun and get some good exercise, they may sometimes be fatal. ● Swimming may be difficult underwater or in areas of the pool that are out of the lifeguard's line of vision, even when there is a lifeguard observer on duty.
2.	Idea / Solution description	<ul style="list-style-type: none"> ● We employ artificial intelligence in this project. We set up cameras underwater to spot persons who are drowning. ● Deep learning may be used to recognize images. If the image is found, an alarm is set out to notify the lifeguards who save drowning individuals.
3.	Novelty / Uniqueness	<ul style="list-style-type: none"> ● The ability of our system software to locate and monitor a drowning individual in realtime. ● The YOLO algorithm is used because of its rapid detecting speed and great accuracy. Thus, it aids lifeguards in providing quick rescues.

5.	Business Model (Revenue Model)	<ul style="list-style-type: none"> ● We can describe the software-based strategy for earning a good living. It is quite helpful for company owners, swimmers, and lifeguards. ● The variety of features makes using our software system appealing to end customers..
----	--------------------------------	--

3.4 Solution Fit

1. CUSTOMER SEGMENT(S) Swimmers who are new to swimming, children and trainee.	4. CUSTOMER CONSTRAINTS Expenditure capacity, financial constraints, a lack of accessible funds, and network connectivity.	7. AVAILABLE SOLUTIONS Trained swimmers and firefighters.
2. JOBS-TO-BE-DONE / PROBLEM We employ a single camera that feeds underwater footage and evaluates the position of swimming to evaluate the likelihood of drowning	5. PROBLEM ROOT CAUSE Customers are required to do it as a result of the rapid growth in opulent activities and the widespread use of pools.	8. BEHAVIOUR Online: Ordering pool lifeguards or drowning detectors Offline: putting things into practice to wear them without fail
3. TRIGGERS Observing others set up virtual eyes to keep a watch on their pools	6. EMOTIONS: BEFORE / AFTER Confident and in charge/lost and uneasy	9. YOUR SOLUTION We employ a single camera to feed footage underwater, analyze swimmer positioning, and determine the likelihood of drowning

4. Requirements

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	InstallationSetup/	Install the Drowning Detection SetupBasedAI mechanism connect them with servers with servers
FR-2	Detection/Analyze	Monitor the different states and position of the swimmers and analyzethe data with train data set
FR-3	Support	Give extra care and support by the lifeguards to the swimmers to makethem swim without fear
FR-4	Alert	Make red light alarm or signal to the lifeguards to save the drowning people

4.2Non-functional Requirements:

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

FR No.	NonFunctionoal Requirement	Description
NFR-1	Usability	The lifeguards and drowning detectors checks the drowning person
NFR-2	Security	Gives alert messages or triggers the alarms when know they are in danger
NFR-3	Reliability	Make a quick decision to save the persons

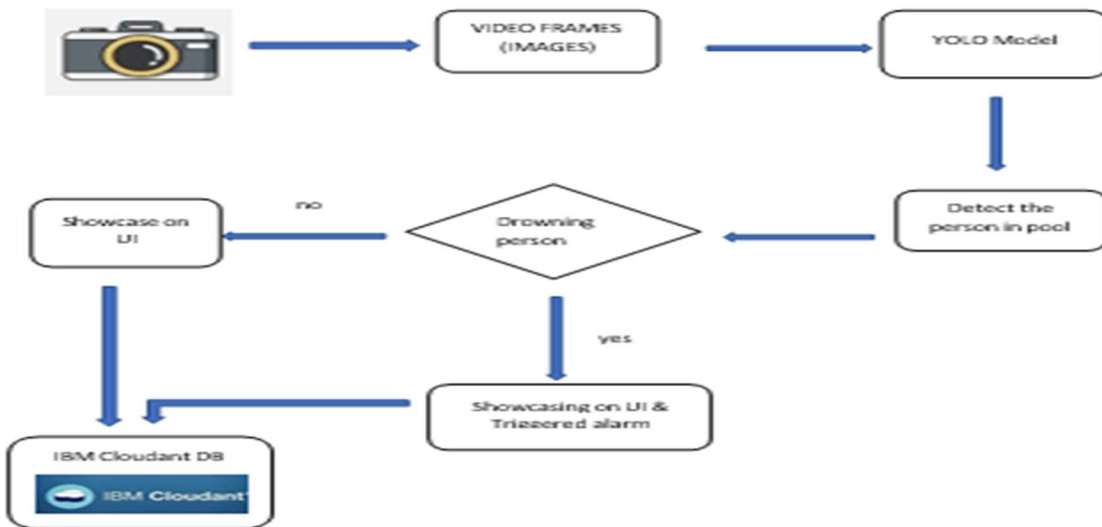
NFR-4	Performance	If any mismatching data is found, action is taken by the lifeguards with theinstruction of administrator.
NFR-5	Availability	Drowning detector which monitors the drowning /Equipment's safety measurement are under control
NFR-6	Scalability	Detects the drowning and immediately notifies the lifeguards

5. Project Design

5.1 Data Flow Diagram



5.2 Technical Architecture



S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, MobileApp, Social media etc.	HTML, CSS,JavaScript /NodeJs etc.
2.	Application Logic-1	Data set design phase	Python
3.	Application Logic-2	Image extraction	Python-YOLO
4.	Application Logic-3	Object Detection	Python-Flask
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant ,Open CV etc.
7.	File Storage	File storage requirements	IBM BlockStorage or OtherStorage Service orLocal File system

8.	Deep Learning Model	Purpose of DeepLearning Model	Object Recognition Model ,CNN , YOLO, etc.
9.	Infrastructure (Server/ Cloud)	Application Deployment on Local System/ Cloud Local Server Configuration: Cloud Server Configuration :	Local, Cloud Foundryetc.



S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Python(Anaconda) open source framework	Python
2.	Security Implementations	Camera Surveillances with security alarms	AI
3.	Scalable Architecture	3-tier Architecture	Python
4.	Availability	Camera available 24/7 when the swimmers are use the poolA	AI
5.	Performance	Detect the drowning person when they are in helpless situation	Python

5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer(Pool owner)	Set up the installation	USN-1	As a Owner install all security measurements	I can connect the drowning detector and	High	Sprint-1

			with alarms ,drowning detector	captured data to the AI based software		
	Detecting the drowning person	USN-2	As a user,I can detect the person who drowning by the system	I get a emergency alert or notify the alarm message	High	Sprint-3
Customer(Lifeguards)	Alert / Notify the Lifeguard or Security Person	USN-3	As a user,I can alert the lifeguards to save the person	Save them and make them to feel swim without fear	Medium	Sprint-2
Customer(Swimmers)	Safety & Protection	USN-4	As a user,I can rescue/ save the life of person from the drowning	Without fear and hopefully enjoy the swimming	High	Sprint-4
Customer Care Executive	Contact/Help	USN-5	Implement or resolve the technical issues	Connect them to solve the issues	Low	Sprint-1
Administrator	Maintain the details	USN-6	Manage and Verify the drowning detection	Access the system and modify the data	High	Sprint4

			System			
--	--	--	--------	--	--	--

6. Project Planning and Scheduling

6.1 Sprint Planning and Estimation

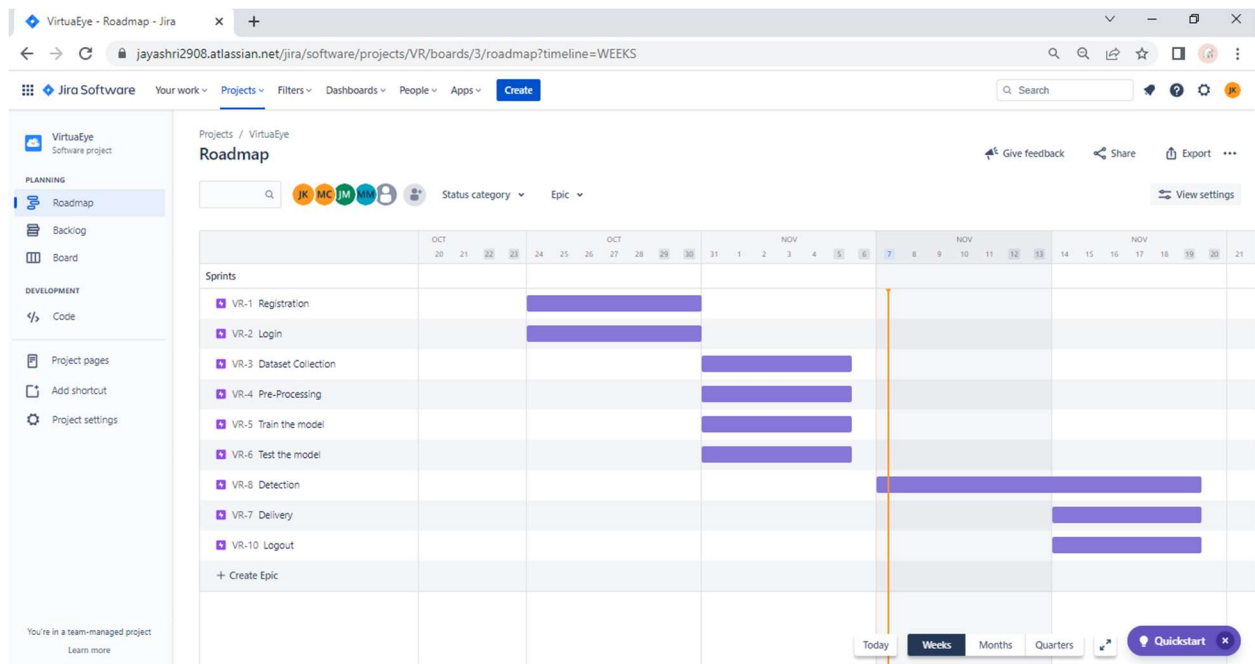
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can sign up for the application by providing my email address, a password, and a password confirmation.	2	High	Malini M
		USN-2	As a user, I can also sign up for the application using Gmail, Phone Number or Facebook	2	Medium	
		USN-3	When I register for the application as a user, I will get a confirmation email	1	Low	
	Login	USN-4	I can access the application as a user by providing my sign-in address and password.	2	High	
	Dataset Collection	USN-5	The uploaded data in the prediction page will assist the user in spotting drowning movements.	2	High	
		USN-6	The dataset gathered will provide highly accurate information on the person's drowning details.	1	Low	
Sprint-2	Data Pre-Processing	USN-7	The extracted dataset is utilized to develop the model.	5	High	Jaya Shri K K
	Train the Model	USN-8	We can train the model then.	8	High	
		USN-9	And we can also test the model.	7	High	

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-3	Detection	USN-10	Load the tested model.	4	High	Javarani M
		USN-11	We can identify the person by collecting real-time data.	6	High	
		USN-12	The real-time data obtained will be checked with the pre-fed data.	8	High	
Sprint-4	Alert	USN-13	If any abnormal movement is detected, the sensor will ring an alarm and notify the lifeguard to save the person.	6	High	Muthuaruna C
		USN-14	The drowning person is detected	4	High	
	Logout	USN-15	Now, the user can logout of the application	2	Medium	

6.2 Project Tracker, Velocity & Burndown Chart:

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

6.3 RoadMap



7.Coding and Solution

7.1 Data Collection

```
from google.colab import drive
drive.mount('/content/drive')
```

Mounted at /content/drive

```
!unzip '/content/drive/MyDrive/Drowning Classification.v1i.folder.zip'
```

```

import numpy as np
import tensorflow as tf
from tensorflow.keras import layers
from tensorflow.keras.models import Sequential
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
import matplotlib.pyplot as plt
batch_size = 32
img_height = 180
img_width = 180
data_dir = "/content/drive/MyDrive/Drowning Classification.v1i.folder.zip"

# Data augmentation on training variable
train_datagen = ImageDataGenerator(rescale=1./255,
zoom_range=0.2,
horizontal_flip=True)

# Data augmentation on testing variable
test_datagen = ImageDataGenerator(rescale=1./255)
xtrain = train_datagen.flow_from_directory('/content/train',
target_size=(64,64),
class_mode='categorical',
batch_size=100)

    Found 678 images belonging to 2 classes.

xtest = test_datagen.flow_from_directory('/content/test',
target_size=(64,64),
class_mode='categorical',
batch_size=100)

    Found 28 images belonging to 2 classes.

model=Sequential()

from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
model=Sequential()

model.add(Convolution2D(32, (3,3), activation = 'relu', input_shape = (64,64,3) ))

```

```
model.add(MaxPooling2D(pool_size = (2,2)))
```

```
model.add(Flatten())
```

```
model.add(Dense(300, activation = "relu"))
```

```
model.add(Dense(150, activation = "relu"))
```

```
model.add(Dense(5, activation = "softmax"))
```

```
model.summary()
```

```
model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])
```

```
print(xtrain.class_indices)
```

```
{'Drowning': 0, 'Not Drowning': 1}
```

```
model.fit(xtrain, epochs = 0, steps_per_epoch = len(xtrain))
```

```
<keras.callbacks.History at 0x7fd28aa70310>
```

```
print(xtest.class_indices)
```

┐

```

{'Drowning': 0, 'Not Drowning': 1}

model.save('Drowning Classification.h5')

from tensorflow.keras.models import load_model
from keras.preprocessing import image
model=load_model("Drowning Classification.h5")

from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image

drown_img = image.load_img('/content/train/Drowning/100_png.rf.11dd1e96267c3d3925cc078cf41
x = image.img_to_array(drown_img)
x = np.expand_dims(x,axis=0)
predicted_class=model.predict(x)

1/1 [=====] - 0s 107ms/step

drown_img
- -


notdrown_img = image.load_img('/content/train/Not Drowning/100_png.rf.7278a89f7b93062d2daa
x = image.img_to_array(drown_img)
x = np.expand_dims(x,axis=0)
predicted_class=model.predict(x)

```

```

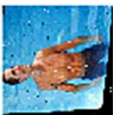
1/1 [=====] - 0s 26ms/step

```

```

notdrown_img

```



7.2 Feature 2

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" />
    <style>
      * {
margin: 0;
padding: 0;
box-sizing: border-box;
}
      body {
        font-family: sans-serif;
      }
      a {
        text-decoration: none;
      }
      li {
        list-style: none;
      }
      .navbar {
        display:
flex;
        align-items: center;
```

```
justify-content: space-between;
padding: 10px;
background-color: blue;
color: white;
}
.nav-links a{color:white
}

/* LOGO */
.logo {
font-size: 32px;
}

/* NAVBARMENU */
.menu {
display:
flex;gap:
1em;
font-size: 18px;
}
.menu li:hover {
background-color:
#4c9e9e;border-radius:
5px; transition: 0.3s ease;
}
.menu li {
padding: 5px 14px;
}
.services{
position:relative;}
```



```
.dropdown {
background-color: rgb(1,139, 139);
padding: 1em 0;
position: absolute;/*WITH RESPECT TO PARENT*/display:
none;
border-radius:
8px;top: 35px;
}

.dropdown li + li
{margin-top:
10px;
}

.dropdown li {
padding:0.5em
1em;width: 8em;
text-align: center;
}

.dropdown li:hover {
background-color:
#4c9e9e;
}

.services:hover .dropdown {
display: block;
}

#example1 {
background: url(swim.jpg);
}

#swim
```

```
{
  height:
200px;width:
50%;
}

</style>

<title>VIRTUAL EYE</title>

</head>

<body>

<nav class="navbar">

  <!-- LOGO -->

  <div class="logo">VIRTUAL EYE</div>

  <!-- NAVIGATION MENU -->

  <ul class="nav-links">

    <!-- USING CHECKBOX HACK -->
    <div class="menu">

      <li><a href="index.html">Home</a></li>

      <li><a href="about.html">About</a></li>

      <li class="services">

        <a href="services.html">Services</a>
      </li>

      <li><a href="register.html"> Register</a></li>

      <li><a href="login.html">Login</a></li>

    </div>

  </ul>

</nav>

<div class="swim">

  
```

```
</div>
</body>
</html>
```

Login.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
  <style>
    * {box-sizing: border-box }
      body
      {
        background-image:url('swim_login.jpg');
        background-size:cover;}

/* Add paddingto containers */
.container {
  padding:
    16px;
}
form
{
  color:blink;
  font-
  size:150%;
```

```
/* Full-width input fields */ input[type=text],
input[type=password] {
    width: 20%;
    padding:
    10px;
    margin: 2px 0 22px
    0;display: inline-
    block; border: none;
    background: #f1f1f1;
}
input[type=text]:focus, input[type=password]:focus {background-color: #ddd;
    outline: none;
}
/* Overwrite default styles of hr */
```

```
/* Set a style for the submit/register button */
```

```
.registerbtn {
    background-color: #04AA6D;
    color: white;
    padding: 16px
    20px;margin: 8px 0;
    border: none;
    cursor: pointer;
    width: 100%;
    opacity: 0.9;
}
.registerbtn:hover {opacity:1;
}
/* Add a blue text color to links */a {
    color: dodgerblue;
```

```
}
/* Set a grey background color and center the text of the "sign in" section */
.signin {
    background-color: #f1f1f1;
    text-align: center;
}

</style>
</head>
<body>
<form><center>
    <div class="container">
        <marquee><h3>VirtualEye - Life Guard for Swimming Pools to Detect Active
Drowning</h3></marquee>
        <h1>LOGIN</h1>
        <p>Login with your credentials.</p>
<label for="email"><b>Email :</b></label>
        <input type="text" placeholder="Enter Email" name="email" id="email"
required><br><br>
<label for="psw"><b>Password :</b></label>
        <input type="password" placeholder="Enter Password" name="psw" id="psw"
required>
<p><a href="/index" class="registerbtn">LOGIN</a></p><br>
        <br>
        <p>Don't have an account?<a href="register.html">Sign Up</a>
    </div></center>
</form>
</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=JosefinSans' rel='stylesheet'>

<link href='https://fonts.googleapis.com/css?family=Montserrat' rel='stylesheet'>

<style>

.header {

top:0;

margin:0p

x;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{  
0%;  
border-radius: 50%;  
}  
.container {  
padding:  
16px;  
}
```



```
span.psw
```

```
{float:
```

```
right;
```

```
padding-top: 16px;margin-top:100px;
```

```
}
```

```
form {border: 3px solid #f1f1f1; margin-left:400px;margin-right:400px;align:center;}
```

```
input[type=text],
```

```
input[type=email],input[type=number],input[type=password] {width:50%;
```

```
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: 50%;
```

```
}
```

```
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: 3}

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

.cancelbtn {
width:
100%;
```

```
}
}
.header {
    margin:
    auto;
}
</style>
</head>
<center>
<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="index.html">Home</a>
<a href="login.html">Login</a>
<a class="active" href="register.html">Register</a></div>
</div>
<div id="login" class="login">
<form action="{{url_for('afterreg')}}" method="post"><div
class="imgcontainer">
<form action="{{url_for('afterreg')}}" method="post"><div
class="imgcontainer">

</div>
<div class="container">
Name:&nbsp;     <input type="text" placeholder="Enter Name" name="name"
required><br>
EmailId:&nbsp;     <input type="email" placeholder="Enter Email
```

```
ID" name="_id" required><br>
Password:&nbsp; &nbsp;<input type="password" placeholder="EnterPassword"
name="psw" required>
<button type="submit">Register</button><br>
</div>
<div class="container" style="background- color:#f1f1f1">
<div class="psw">Already have an account?&nbsp; &nbsp;<a
href="login.html">Login</a></div >
</div>
</form>
</center>
</body>
</html>
```

Prediction.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=JosefinSans' rel='stylesheet'>
```

```
<link href='https://fonts.googleapis.com/css?family=Montserrat' rel='stylesheet'>
```

```
<style>
```

```
.header {
```

```
top:0;
```

```
margin:0p
```

```
x;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;align:center;}  
input[type=text],  
  
input[type=email],input[type=number],input[type=password] {width:  
50%;  
padding: 12px 20px;  
display: inline-block;  
margin-bottom:18px;  
border: 1px solid #ccc;  
box-sizing: border-box;  
}  
.cancelbtn  
{width:  
auto;  
padding: 10px 18px;  
background-color: #f44336;  
}
```

```

.imgcontainer {
text-align:
center;
margin: 24px 0 12px 0;
}
img.aavar {
width: 30%;

16px;
}
span.psw
{float:
right;
padding-top: 16px;
}
/* Changestyles for span and cancel button on extra small /style>
</head>
<center>
<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="index.html">Home</a>
<a href="{{ url_for('login')}}">Login</a>
<a class="active" href="{{ url_for('register')}}">Register</a></div>
</div>
<center>
<h2> VirtualEye - Life Guard for SwimmingPools to DetectActiveDrowning</h2>

```


</center>

<p style="text-align:left;">Swimming is one of the best exercises that helps people to reduce stress in this urban lifestyle.Swimming pools are found larger in number in hotels, and weekend tourist spots and barely people have them in their house backyard.Beginners, especially, often feel it difficult to breathe underwater which causes breathing trouble which in turn causes a drowning accident.Worldwide, drowning-produces a higher rate of mortality without causing injury to children. Children under six of their age are found to be suffering the highest drowning mortality rates worldwide. Such kinds of deaths account for the third cause of unplanned death globally, with about 1.2 million cases yearly.To overcome this conflict, a meticulous system is to be implemented along the swimming pools to save human life. </p>

<video style="width=10;top:150px;left:700px" autoplayloop muted >

<source src="avs.mp4" type="video/mp4">

</video>

<center><button type="button">Click Me!</button>

<p>(prediction)</p></center>

</body>

</html>screens*/@mediascreen and (max-width: 300px) {

span.psw {

display:

block;float:

none;

}

.cancelbtn {

width:

100%;

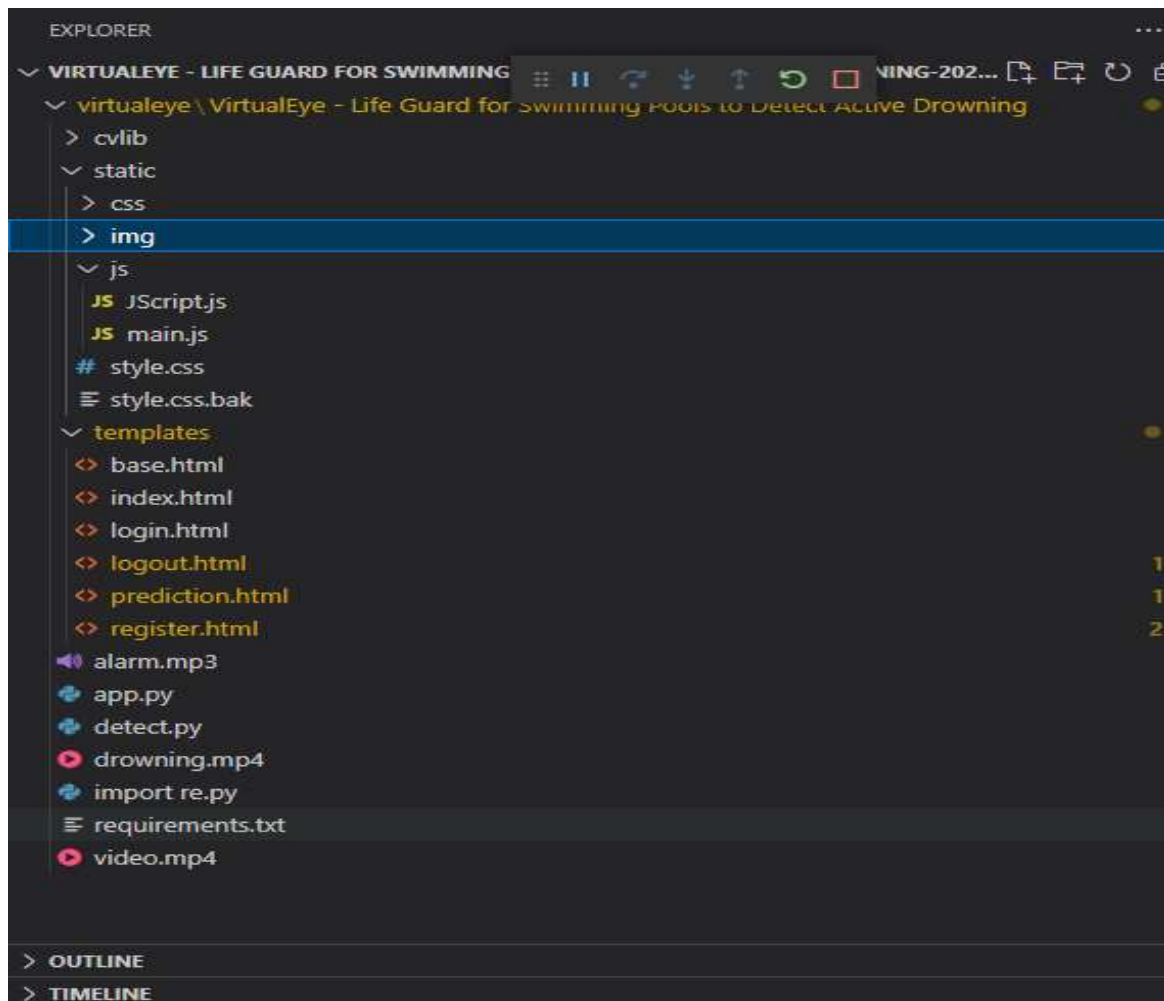
}

}

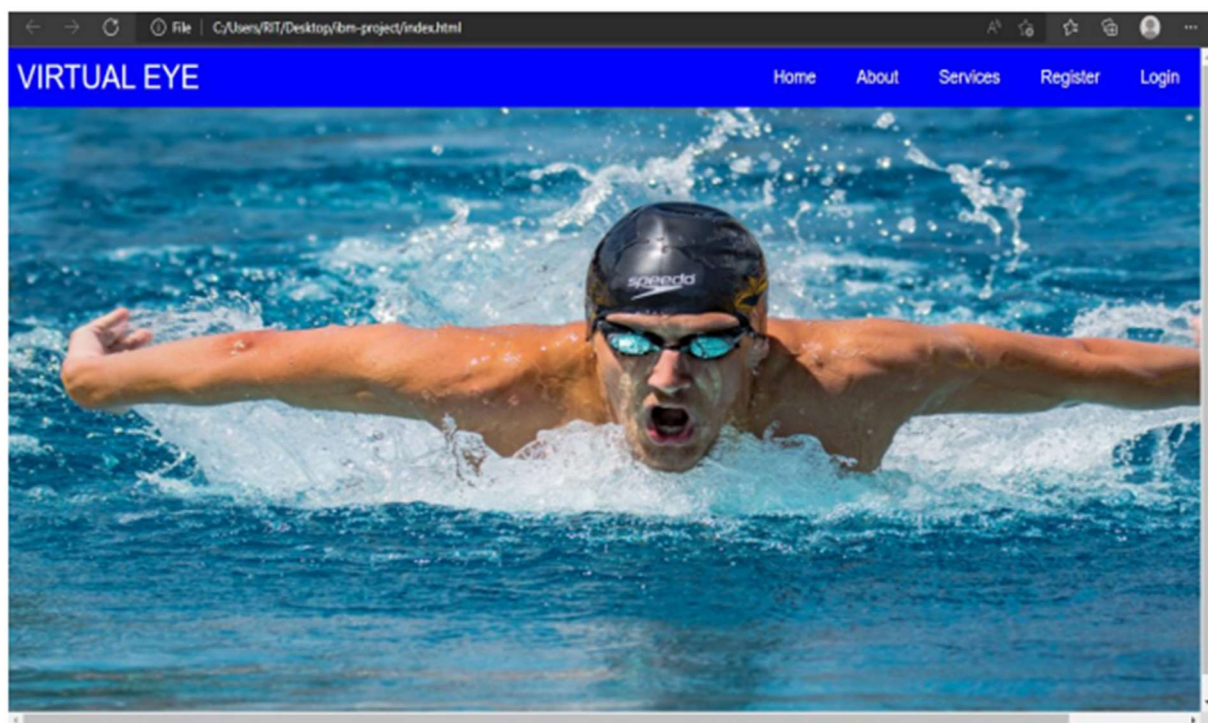
```
.header {  
  margin:  
  auto;  
}
```

8. Testing

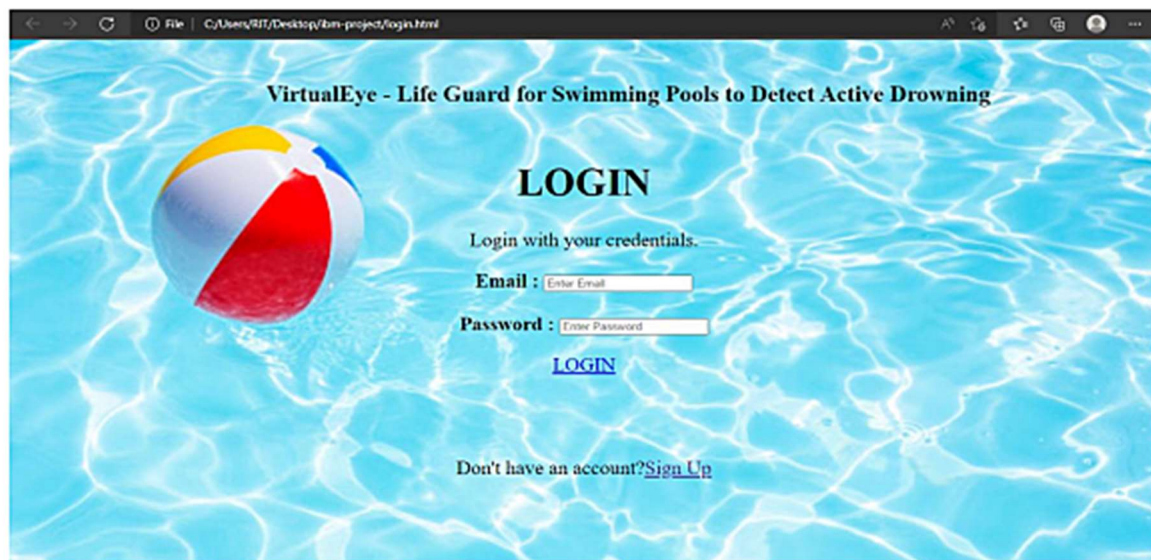
8.1. Test Cases



Homepage:



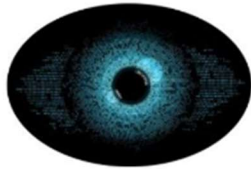
Login page:



Register page:

Virtual Eye

Home Login Register



Name:

Emailid:

Password:


Already have an account? [Login](#)

Prediction page:

Virtual Eye

Home Login Register

Swimming is one of the best exercises that helps people to reduce stress in this urban lifestyle. Swimming pools are found larger in number in hotels, and weekend tourist spots and barely people have them in their house backyard. Beginners, especially, often feel it difficult to breathe underwater which causes breathing trouble which in turn causes a drowning accident. Worldwide, drowning produces a higher rate of mortality without causing injury to children. Children under six of their age are found to be suffering the highest drowning mortality rates worldwide. Such kinds of deaths account for the third cause of unplanned death globally, with about 12 million cases yearly. To overcome this conflict, a meticulous system is to be implemented along the swimming pools to save human life.



(prediction)

Logged out:



Successfully Logged Out!!!!

[login again](#)

10.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

11.Conclusion

Once we have the working drowning detection model we can feed live video footage of the swimming pool to it so that it can keep detecting continuously for any drowning activities. If drowning is detected it will be highlighted on the system screen as well as alarms will be raised to alert security guards so that they can initiate rescue

12.Appendix

Source Code

<https://colab.research.google.com/drive/1kSw48COblIU6C4sfftvKdvQRhhBqvUW3?usp=sharing>

<https://colab.research.google.com/drive/1NIGHcmvI2EXILrh-55nB40wB5AcDYy1?usp=sharing>

Github

<https://github.com/IBM-EPBL/IBM-Project-18424-1659684904>

Videolink

<https://drive.google.com/file/d/1oTQfueW3iTsUAqd1trmB36YHI3kAmbtJ/view?usp=sharing>