Project Development Phase
Sprint - 3
Application Building

DATE	9 NOV 2022
TEAM ID	PNT2022TMID54389
PROJECT NAME	Virtual Eye - LifeGuard For Swimming Pools To Detect
MAXIMUM MARKS	8 MARKS

Building Html Pages

Base.html:

```
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>High Quality Facial Recognition</title>
    <link
href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css"
rel="stylesheet">
    <script
src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></scri
pt>
    <script
src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
    <script
src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></scr
ipt>
    <link href="style.css" rel="stylesheet">
    <style>
    .bg-dark {
        background-color: #42678c!important;
    }
    #result {
```

```
        color: #0a1c4ed1;
    }
    </style>
</head>

<body style="background-color:black";>
        <header id="head" class="header">
        <section  id="navbar">
                <h1 class="nav-heading"></i>Virtual Eye</h1>
            <div class="nav--items">
                <ul>
                    <li><a  href="index.html">Home</a></li>
                <li><a href="logout.html">Logout</a></li>
               <!-- <li><a href="#about">About</a></li>
                <li><a href="#services">Services</a></li> -->

                </ul>
            </div>
        </section>
    </header>
    <div class="container">
        <div id="content" style="margin-top:2em">
        <div class="container">
          <div class="row">
            <div class="col-sm-6 bd" >
              <h2><em style="color:white;">High Quality Facial
Recognition</em></h2>
              <br>
              <p><h5><i style="color:white;">Emotion Detection Through
Facial Feature Recognition</i></h5></p>
                <img
src="https://130e178e8f8ba617604b-8aedd782b7d22cfe0d1146da69a52436.ssl.
cf1.rackcdn.com/facial-recognition-use-triggers-gdpr-fine-showcase_imag
e-10-a-12991.jpg" style="height:240px"class="img-rounded"
alt="Gesture">
            </div>
            <div class="col-sm-6">
                <div>
                    <h4 style="color:white;">Upload Image Here</h4>
            <form action = "http://localhost:5000/" id="upload-file"
method="post" enctype="multipart/form-data">
                <label for="imageUpload" class="upload-label">
                    Choose Image
```

```
                </label>
                <input type="file" name="image" id="imageUpload"
accept=".png, .jpg, .jpeg,.pdf">
            </form>

            <div class="image-section" style="display:none;">
                <div class="img-preview">
                    <div id="imagePreview">
                    </div>
                </div>
                <div>
                    <button type="button" class="btn btn-info btn-lg "
id="btn-predict">Analyse</button>
                </div>
            </div>

            <div class="loader" style="display:none;"></div>

            <h3>
                <span id="result"> </span>
            </h3>


        </div>
            </div>

          </div>
        </div>
        </div>
    </div>
</body>

<footer>
    <script src="main.js" type="text/javascript"></script>
</footer>

</html>
```

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+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgF
AW/dAiS6JXm" crossorigin="anonymous">
    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGF
F93hXpG5KkN" crossorigin="anonymous"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper
.min.js"
integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPsk
vXusvfa0b4Q" crossorigin="anonymous"></script>
    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.j
s"
integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUa
r5+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&di
splay=swap" rel="stylesheet">
    <link rel="stylesheet" href="style.css">
    <!-- <script defer src="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="index.html">Home</a></li>
                <li><a href="login.html">Login</a></li>
```

```
                <li><a href="register.html">Register</a></li>
                <li><a href="prediction.html">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="index.html">Home</a>
    <a class="active" href="login.html">Login</a>
    <a  href="register.html">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>

    </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+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgF
AW/dAiS6JXm" crossorigin="anonymous">
    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGF
F93hXpG5KkN" crossorigin="anonymous"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper
.min.js"
integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPsk
vXusvfa0b4Q" crossorigin="anonymous"></script>
    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.j
s"
integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUa
r5+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&di
splay=swap" rel="stylesheet">
    <link rel="stylesheet" href="style.css">


    <script defer src="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="index.html">Home</a></li>
                <li><a href="logout.html">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>
```

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="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;}
```

```css
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="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">
      <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>

    </div>
    <div class="container" style="background-color:#f1f1f1">
    <div class="psw">Already have an account?   <a
href="login.html">Login</a></div >
  </div>
  </form>

</div>


</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="index.html">Home</a>
    <a href="login.html">Login</a>
    <a href="register.html">Register</a>
  </div>
</div>
<div class="main">
<h1>Successfully Logged Out!</h1>
<h3 style="color:#4CAF50">Login for more information<h3>

  <a href="login.html"><button type="submit">Login</button></a>
</form>
</div>

</body>
</html>
```

Python Code:

init

```
from object_detection import detect_common_objects
```

utils

```
import requests
import progressbar as pb
```

```
import os


def download_file(url, file_name, dest_dir):
    if not os.path.exists(dest_dir):
        os.makedirs(dest_dir)

    full_path_to_file = dest_dir + os.path.sep + file_name

    if os.path.exists(dest_dir + os.path.sep + file_name):
        return full_path_to_file

    print("Downloading " + file_name + " from " + url)

    try:
        r = requests.get(url, allow_redirects=True, stream=True)
    except:
        print("Could not establish connection. Download failed")
        return None

    file_size = int(r.headers['Content-Length'])
    chunk_size = 1024
    num_bars = round(file_size / chunk_size)

    bar = pb.ProgressBar(maxval=num_bars).start()

    if r.status_code != requests.codes.ok:
        print("Error occurred while downloading file")
        return None

    count = 0

    with open(full_path_to_file, 'wb') as file:
        for chunk in r.iter_content(chunk_size=chunk_size):
            file.write(chunk)
            bar.update(count)
            count += 1

    return full_path_to_file
```

object_detection

```
import cv2
```

```
import os
import numpy as np
from utils import download_file

initialize = True
net = None
dest_dir = os.path.expanduser(
    '~') + os.path.sep + '.cvlib' + os.path.sep + 'object_detection' +
os.path.sep + 'yolo' + os.path.sep + 'yolov3'
classes = None
# colors are BGR instead of RGB in python
COLORS = [0, 0, 255], [255, 0, 0]


def populate_class_labels():
    # we are using a pre existent classifier which is more reliable and
more efficient than one
    # we could make using only a laptop
    # The classifier should be downloaded automatically when you run
this script
    class_file_name = 'yolov3_classes.txt'
    class_file_abs_path = dest_dir + os.path.sep + class_file_name

    if not os.path.exists(class_file_abs_path):
        download_file(url=url, file_name=class_file_name,
dest_dir=dest_dir)
    f = open(class_file_abs_path, 'r')
    classes = [line.strip() for line in f.readlines()]

    return classes


def get_output_layers(net):
    # the number of output layers in a neural network is the number of
possible
    # things the network can detect, such as a person, a dog, a tie, a
phone...
    layer_names = net.getLayerNames()

    output_layers = [layer_names[i[0] - 1] for i in
net.getUnconnectedOutLayers()]

    return output_layers
```

```
def draw_bbox(img, bbox, labels, confidence, Drowning,
write_conf=False):
    global COLORS
    global classes

    if classes is None:
        classes = populate_class_labels()

    for i, label in enumerate(labels):

        # if the person is drowning, the box will be drawn red instead
of blue
        if label == 'person' and Drowning:
            color = COLORS[0]
            label = 'DROWNING'
        else:
            color = COLORS[1]

        if write_conf:
            label += ' ' + str(format(confidence[i] * 100, '.2f')) +
'%'

        # you only need to points (the opposite corners) to draw a
rectangle. These points
        # are stored in the variable bbox
        cv2.rectangle(img, (bbox[i][0], bbox[i][1]), (bbox[i][2],
bbox[i][3]), color, 2)

        cv2.putText(img, label, (bbox[i][0], bbox[i][1] - 10),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 2)

    return img


def detect_common_objects(image, confidence=0.5, nms_thresh=0.3):
    Height, Width = image.shape[:2]
    scale = 0.00392

    global classes
    global dest_dir
```

```
    # all the weights and the neural network algorithm are already
preconfigured
    # as we are using YOLO

    # this part of the script just downloads the YOLO files
    config_file_name = 'yolov3.cfg'
    config_file_abs_path = dest_dir + os.path.sep + config_file_name

    weights_file_name = 'yolov3.weights'
    weights_file_abs_path = dest_dir + os.path.sep + weights_file_name

    url =
'https://raw.githubusercontent.com/Reema1234ag/Drowning-Risk-Analysis/m
aster/yolov3.cfg'

    if not os.path.exists(config_file_abs_path):
        download_file(url=url, file_name=config_file_name,
dest_dir=dest_dir)

    url = 'https://pjreddie.com/media/files/yolov3.weights'

    if not os.path.exists(weights_file_abs_path):
        download_file(url=url, file_name=weights_file_name,
dest_dir=dest_dir)

    global initialize
    global net

    if initialize:
        classes = populate_class_labels()
        net = cv2.dnn.readNet(weights_file_abs_path,
config_file_abs_path)
        initialize = False

    blob = cv2.dnn.blobFromImage(image, scale, (416, 416), (0, 0, 0),
True, crop=False)

    net.setInput(blob)

    outs = net.forward(get_output_layers(net))

    class_ids = []
    confidences = []
```

```
    boxes = []

    for out in outs:
        for detection in out:
            scores = detection[5:]
            class_id = np.argmax(scores)
            max_conf = scores[class_id]
            if max_conf > confidence:
                center_x = int(detection[0] * Width)
                center_y = int(detection[1] * Height)
                w = int(detection[2] * Width)
                h = int(detection[3] * Height)
                x = center_x - w / 2
                y = center_y - h / 2
                class_ids.append(class_id)
                confidences.append(float(max_conf))
                boxes.append([x, y, w, h])

    indices = cv2.dnn.NMSBoxes(boxes, confidences, confidence,
nms_thresh)

    bbox = []
    label = []
    conf = []

    for i in indices:
        i = i[0]
        box = boxes[i]
        x = box[0]
        y = box[1]
        w = box[2]
        h = box[3]
        bbox.append([round(x), round(y), round(x + w), round(y + h)])
        label.append(str(classes[class_ids[i]]))
        conf.append(confidences[i])

    return bbox, label, conf
```

App.py

```
!pip install matplotlib-venn
!apt-get -qq install -y libfluidsynth1
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
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('35997c82-051e-4f8c-9eb5-27ba173d6dcb-bluemix','Ud0wunTPOI
_8h5ZtEqi1IXk1gIKeYLmpUsCn0EeO8T4z', 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 res():
    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=True)
```