

Sri Lanka Institute of Information Technology



Information Security Project - IE3092

Intruder Detector for Webinar with Face Detection

Submitted By:

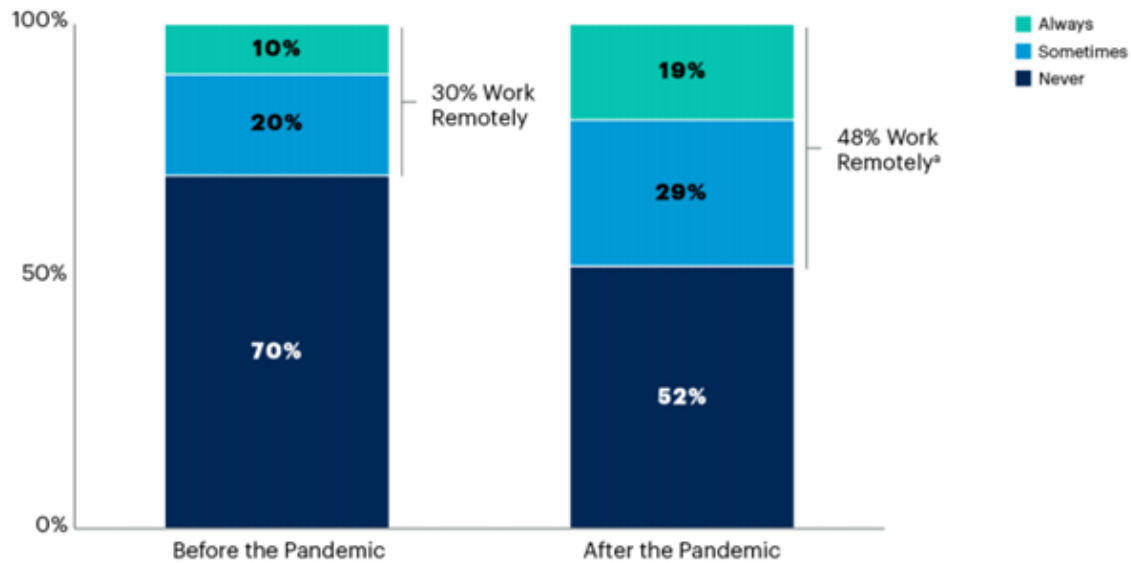
Registration No	Name
IT20228880	Kumara K.K.K.P
IT20223144	Chaminda P.I

Table of Contents

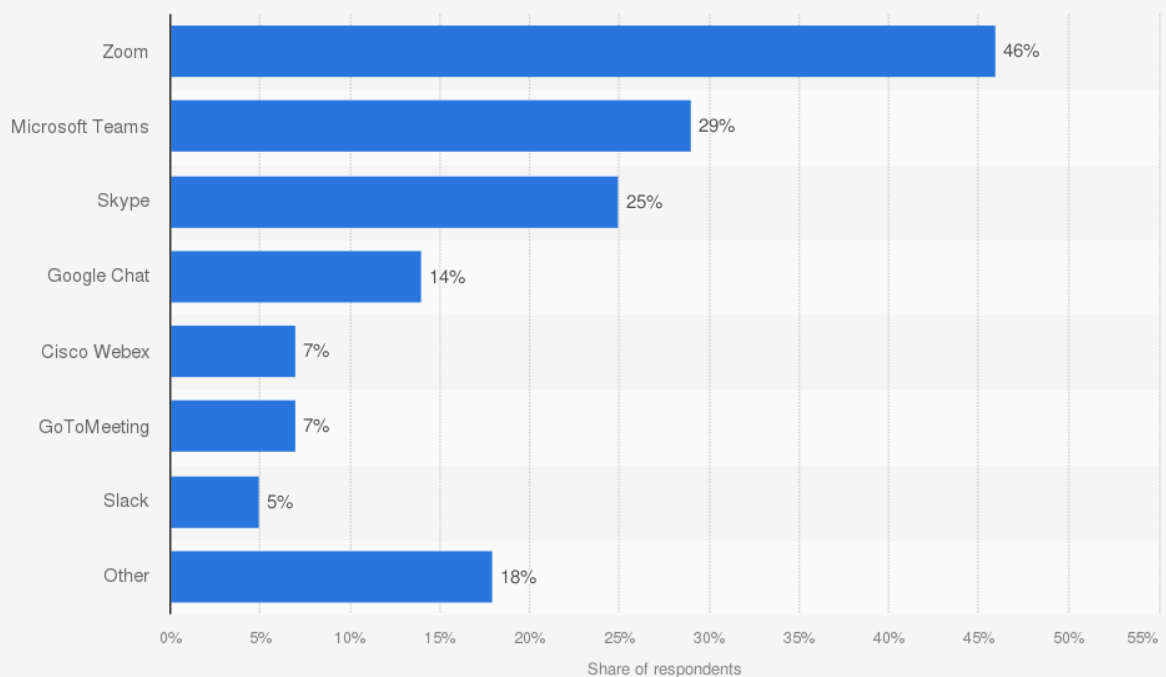
Introduction	3
Why intruder detector?	4
How to use this product?	5
Importance of Intruder Detector	6
What is the face recognition?	7
Main components of Face Recognition system	8
Methodology.....	9
01.Setup dependencies.....	9
02.Data collection for train the model	9
03.Train the model.....	11
04.Face Recognition.....	12
05.Main User interfaces and PHP	16
Full Code of Application	21
Design and Implementation constraints.....	24
Implementation plan	25
How Intruder detector works	26
Train the model.....	27
Secure testing	28
White box testing.....	28
Black box testing	28
Future improvements of application	29
Conclusion.....	29
References	30

Introduction

Projected Percentage of Employees Working Remotely, Before and After the Pandemic



Leading online communication services used on a regular basis worldwide as of March 2021



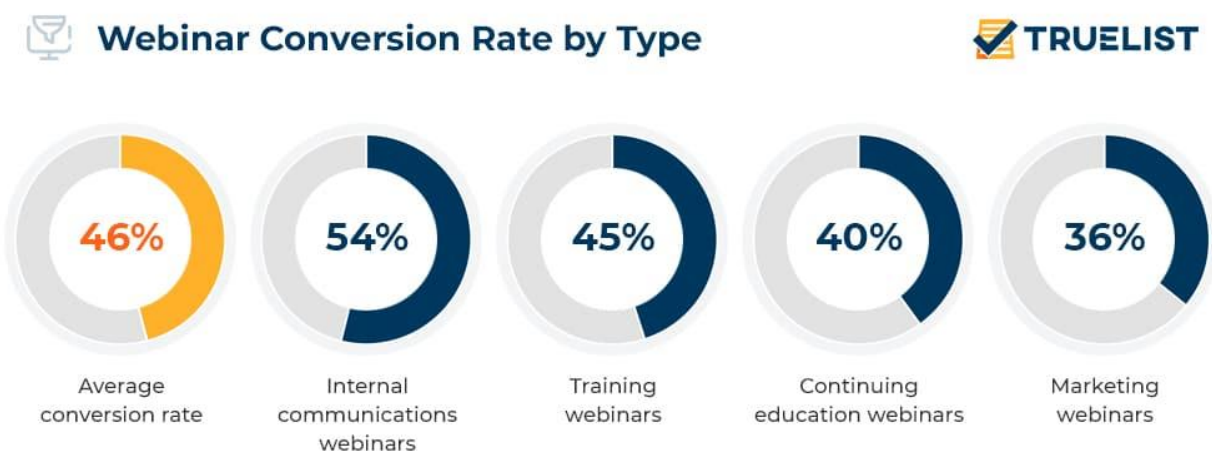
Source
YouGov
© Statista 2022

Additional Information:
Worldwide; YouGov; March 2021; 18 years and older; data concerning working adults in 17 markets

According to the statistics given by above images with the pandemic employees and people and organizations do their day to day operations remotely. As well as according to 2nd image of statistics it is about the using of webinar applications in 2021 with the pandemic situation all meetings of organizations , education sessions , special events are moved to online webinar platforms .

Sensitive data can be leaked which discussed on this webinar meetings such as business goals, trade secrets ,future plans .Anyone can join the session as legitimate user of the organization because there is less authorization methods are implemented in default webinar applications. As an example if there is name is right organization approve the entering to the meeting session. So we need authorization method for identify people who attend to that meetings. For that we introduce to you Intruder Detector V1.0

Why intruder detector?

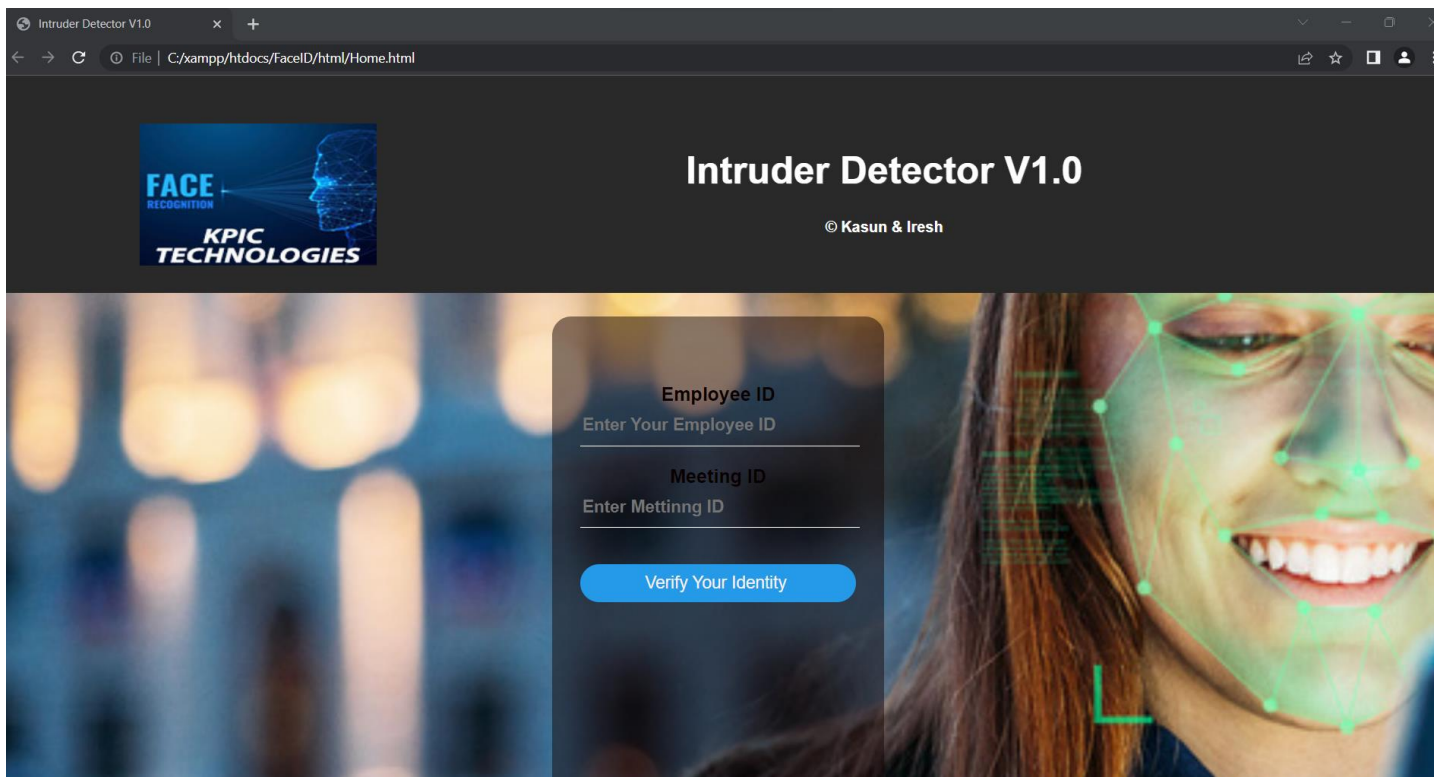


Intruder detector for webinar is a research-based product which using face recognition techniques .

With the provided link any intruder can join the session, but some Online meeting platforms organization can admit only the persons with the names and checking manually in the meeting. It is a time-wasting process. As well as some intruder can identity theft and log to the meetings. So it is a major problem to protect the confidentiality of the company. Because of that problems and as a solution we are introducing our Intruder detector for webinar product.

How to use this product?

Any organization which implemented this product they can identify intruders who coming to the very confidential meetings. Organization provide a invite link before the meeting .Intruders also can get that link .But that link is our product link ,not the real meeting link By this product filtering all people who has the invite link that provided by the organization. After by a face recognition system that implemented in product identify all and only authenticated people are redirecting to the real meeting link



Intruder detector run as a web application by using python and flask environment. As a developing team of intruder detector is trying to provide better interfaces for users of this product.

Importance of Intruder Detector

1. Identify intruders who trying to attend confidential meetings

Attackers who try to compromise a network's security are known as intruders. In this scenario with the face recognition techniques application identifies people who join the sessions so application deny the access request to the meeting to unauthorized people after face recognition process

2. Monitoring people which attending to the meetings and Get an attendance Reports

People should do the security verification with the face recognition after that they can go the meeting after the verification that data goes to the database by that application provides the who were attended to the meeting .This feature can be essential feature for audit team also.

3. Protects the confidentiality

When using the application, keep confidential information and important data safe. When demonstrating the application, protect sensitive data and information by disclosing it for attacker purposes. The confidentiality of information is determined by application data and information that are prioritized based on their sensitivity and whether they are intercepted or observed.

4. Reduce the information leakage

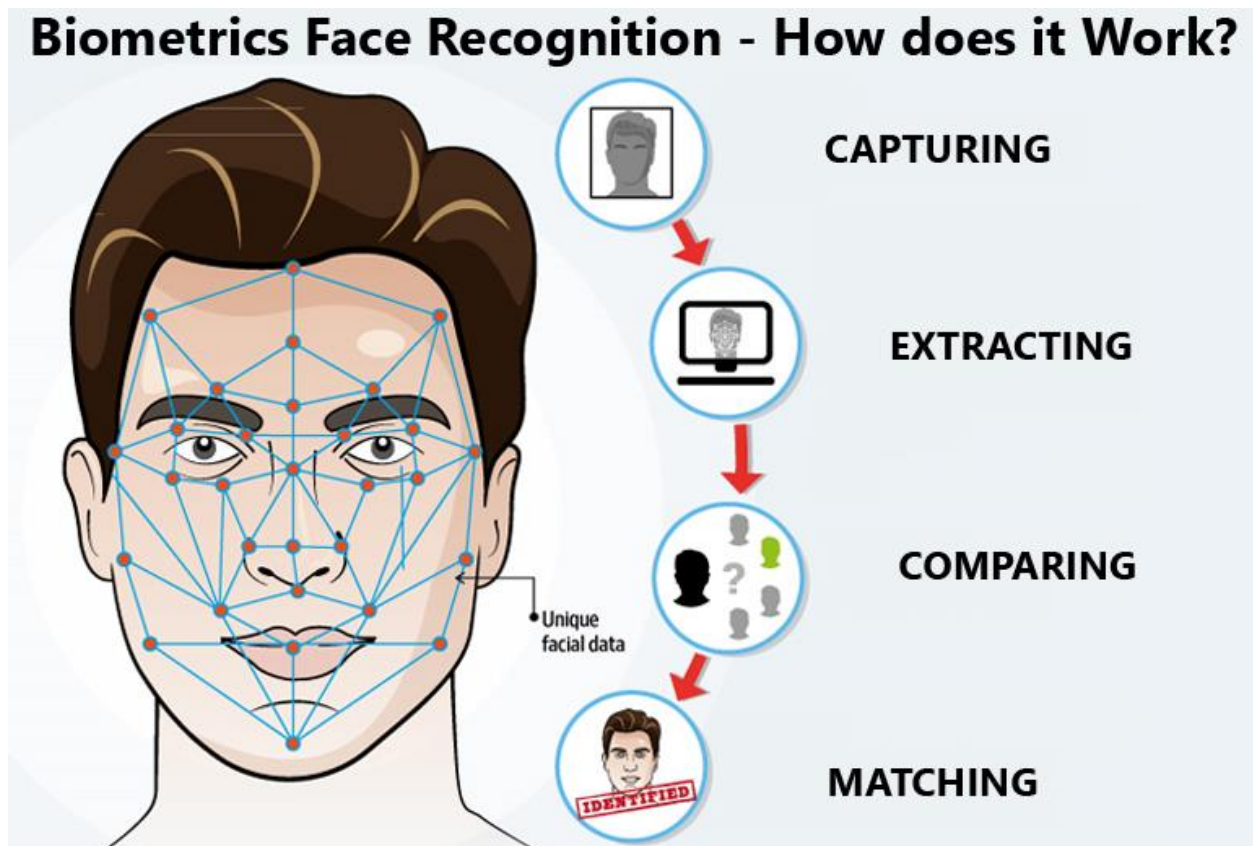
When using applications on the system, information can be more than secure and can't be broken, making it complex for hackers to access the information and data. even if it is preventing unauthorized access by hiding. Given the importance of data loss prevention, this could be a simple system to implement. According to the software implementation, it aids in the prevention of data and information loss.

5. Protects trade secrets

It ensures the legality of protecting the rights of the owner and preventing the disclosure of the application when providing it to the public. when using another's consent for attacking purposes without asking for and granting their access, according to the application company components protected.

What is the face recognition?

Ensuring that the system is based on some activities performed for authentication purposes improves its ability to be secured through the use of some identification methods, such as face recognition. Identifying the authentication face and ensuring that that type of face from people is authenticated to the organization's system security. When using face recognition, you must use system authentication to implement some security controls and functions. This can be useful for those who have been designated as authorized management and employees who require access to information and data. When someone gains unauthorized access to a system, he or she has access to the database, which should be isolated. As a result, he or she can do anything without gaining access to the system. Face identification must involve the user who must access the system as well as the sensitive data.



Main components of Face Recognition system

Detection

Face extraction begins detection. Marking facial features follows. Age and stature don't modify some facial traits. Distance between eyes, eye socket depth, and nose shape. 80 'landmarks' exist. These landmark measurements produce a code. This code is a person's 'faceprint'

Matching

Faceprint is matched with system prints. The image undergoes multiple technological stages to assure accuracy. Most of our datasets are 2D photographs, thus the images must be processed. This entails removing face landmarks to make them 3D. Low-resolution images must be encoded and decoded to obtain high-resolution information. Lighting, face expression, and angles must be considered.

Identification

Identification depends on whether the program is used for monitoring or authentication. This phase should match the subject 1:1. This can be done in several ways: a rapid pass to reduce alternatives, then complicated layers take over. Some companies use skin texture to improve facial recognition accuracy.

Key features of Intruder Detector

- Trained and growing database

Any FRS's accuracy depends on its AI-trained database. The data must increase and be gender and racially diverse. Lighting, angles, and face expressions must vary in training data. A good database includes several image resolutions for the system. The FRS is a fantastic database for machine learning applications to learn from.

- Algorithm accuracy:

When evaluating a FRS, the two most important metrics to consider are the false acceptance rate (FAR) and the false rejection rate (FRR). False-identity matching (FAR) occurs when two images are incorrectly compared and considered to be the same. If you're trying to use it as a security measure, the wrong person might get in. False positive image matching (FRM) is a technique whereby identical pictures are incorrectly categorized as variants. The appropriate individual might be shut out here. The FAR needs to be small, while the FRR needs to be large, for any realistic security scenario.

Methodology

01.Setup dependencies

The main objective of this step is to import dependencies. Mainly numpy, open cv environment, pillow and mysql.connector. Open cv environment is used to get camera environment and pillow supports to the face recognition part as well as we have to install mysql.connector to make connections with the database

User need to install the libraries by using pip install

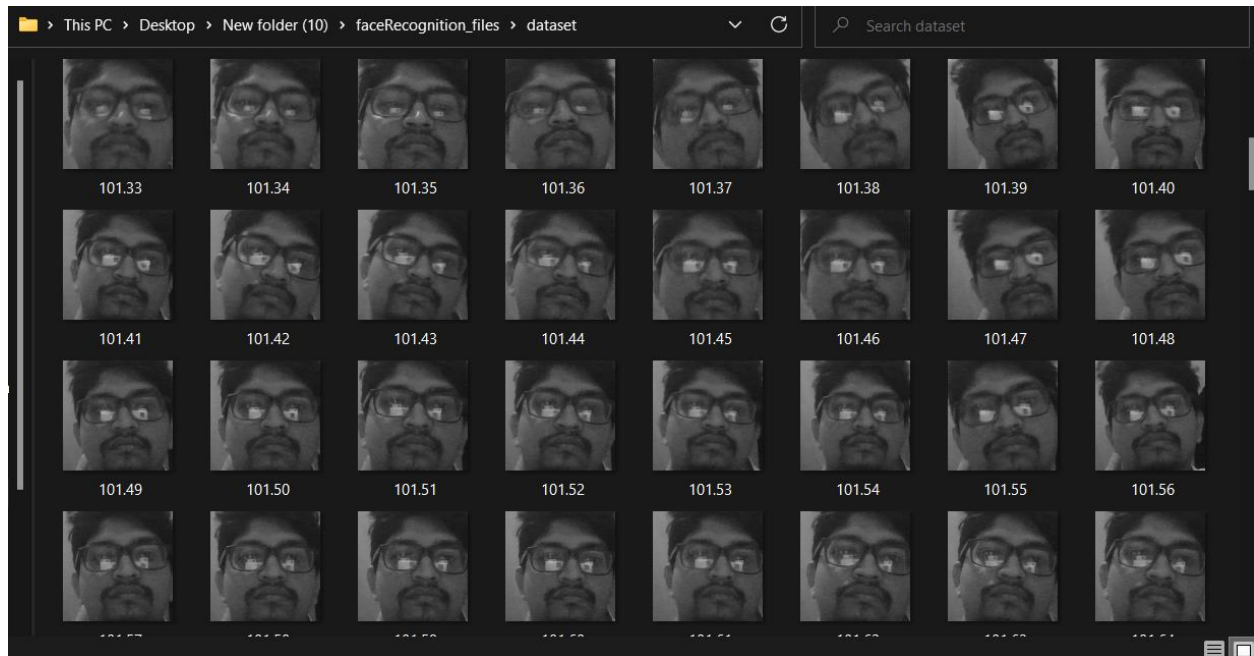
```
faceRecognition_files > app.py > face_recognition > draw_boundary
1 from flask import Flask, render_template, request, session, redirect, url_for, Response, jsonify
2 import mysql.connector
3 import cv2
4 from PIL import Image
5 import numpy as np
6 import os
7 import time
8 from datetime import date
9
```

02.Data collection for train the model

```
41
42 cap = cv2.VideoCapture(0)
43
44 mycursor.execute("select ifnull(max(img_id), 0) from img_dataset")
45 row = mycursor.fetchone()
46 lastid = row[0]
47
48 img_id = lastid
49 max_imgid = img_id + 100
50 count_img = 0
51
52 while True:
53     ret, img = cap.read()
54     if face_cropped(img) is not None:
55         count_img += 1
56         img_id += 1
57         face = cv2.resize(face_cropped(img), (200, 200))
58         face = cv2.cvtColor(face, cv2.COLOR_BGR2GRAY)
59
60         file_name_path = "dataset/"+nbr+"."+str(img_id) + ".jpg"
61         cv2.imwrite(file_name_path, face)
62         cv2.putText(face, str(count_img), (50, 50), cv2.FONT_HERSHEY_COMPLEX, 1, (0, 255, 0), 2)
63
64         mycursor.execute("INSERT INTO `img_dataset` (`img_id`, `img_person`) VALUES ('{}', '{}')".format(img_id, nbr))
65         mydb.commit()
66
67         frame = cv2.imencode('.jpg', face)[1].tobytes()
68         yield (b'--frame\r\n' + b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')
69
70     if cv2.waitKey(1) == 13 or int(img_id) == int(max_imgid):
71         break
72     cap.release()
73     cv2.destroyAllWindows()
74
```

In line 42 system should get the web camera input by using OpenCV dependencies then it goes to a while loop and by these code segment capture frames of the person who Infront the camera .It takes 100 frames to store in application database and they are used to data training session. After taking 100 frames camera should break according to the last line of code segment.

Application takes 100 frames of face and according to the code it assigns a number to each photo with regards of dataset number like following.



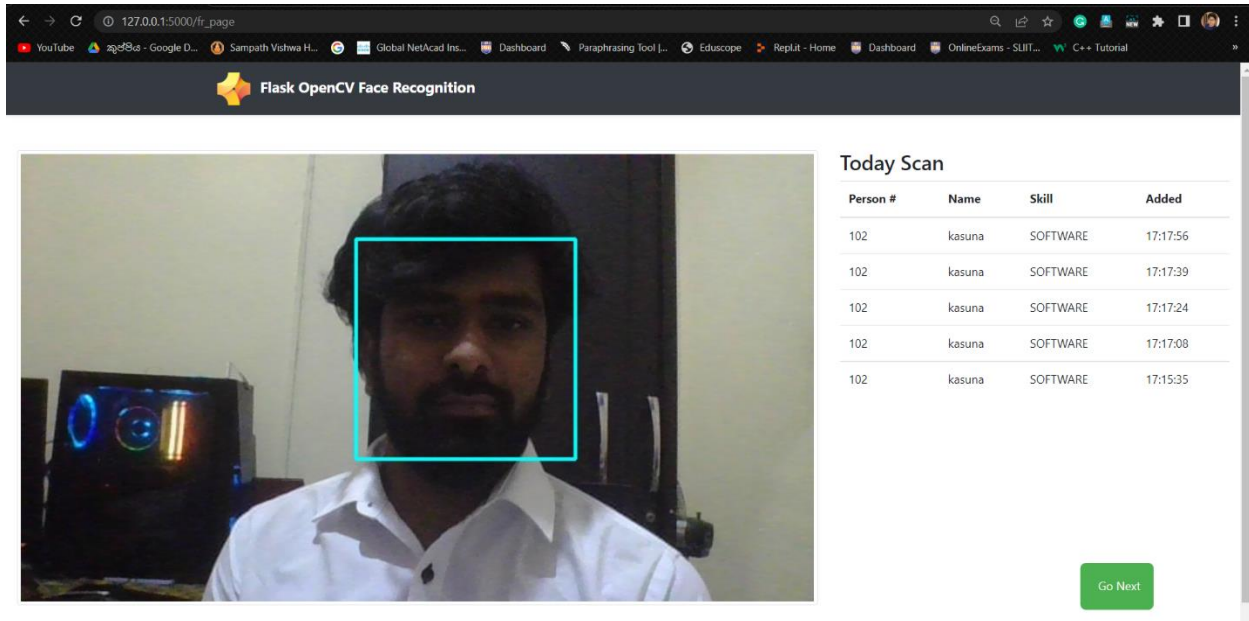
03.Train the model

With the dataset that application have collected .Following coding part do the training the model for face recognition and it writes data on classifier.xml file

[illegible]

04.Face Recognition

By using the dataset application uses that data to verify people who comes to the application .When looking at the code first we had to define sizes of camera input .



The screenshot shows a web browser window with the address bar displaying '127.0.0.1:5000/fr_page'. The browser's tab bar includes several open tabs, including 'YouTube', 'Google D...', 'Sampath Vishwa H...', 'Global NetAcad Ins...', 'Dashboard', 'Paraphrasing Tool J...', 'Eduscope', 'Replit - Home', 'Dashboard', 'OnlineExams - SUIT...', and 'C++ Tutorial'. The web application's header is dark with a logo and the title 'Flask OpenCV Face Recognition'. The main content area is divided into two sections. On the left is a video feed showing a man with a beard and a white shirt, with a red rectangular bounding box around his face. On the right is a section titled 'Today Scan' containing a table with the following data:

Person #	Name	Skill	Added
102	kasuna	SOFTWARE	17:17:56
102	kasuna	SOFTWARE	17:17:39
102	kasuna	SOFTWARE	17:17:24
102	kasuna	SOFTWARE	17:17:08
102	kasuna	SOFTWARE	17:15:35

Below the table is a green button labeled 'Go Next'.

```
wCam, hCam = 400, 400

cap = cv2.VideoCapture(0)
cap.set(3, wCam)
cap.set(4, hCam)

while True:
    ret, img = cap.read()
    img = recognize(img, clf, faceCascade)

    frame = cv2.imencode('.jpg', img)[1].tobytes()
    yield (b'--frame\r\n'
           b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n\r\n')

    key = cv2.waitKey(1)
    if key == 27:
        break
```

[illegible]

```
mycursor.execute("select a.img_person, b.prs_name, b.prs_skill "
                " from img_dataset a "
                " left join prs_mstr b on a.img_person = b.prs_nbr "
                " where img_id = " + str(id))

row = mycursor.fetchone()

pnbr = row[0]
pname = row[1]
pskill = row[2]

if int(cnt) == 30:
    cnt = 0

    mycursor.execute("insert into accs_hist (accs_date, accs_prsn) values('"+str(date.today())+"', '" + pnbr + "')")
    mydb.commit()

    cv2.putText(img, pname + ' | ' + pskill, (x - 10, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (153, 255, 255), 2, cv2.LINE_AA)
    time.sleep(1)

    justscanned = True
    pause_cnt = 0
```

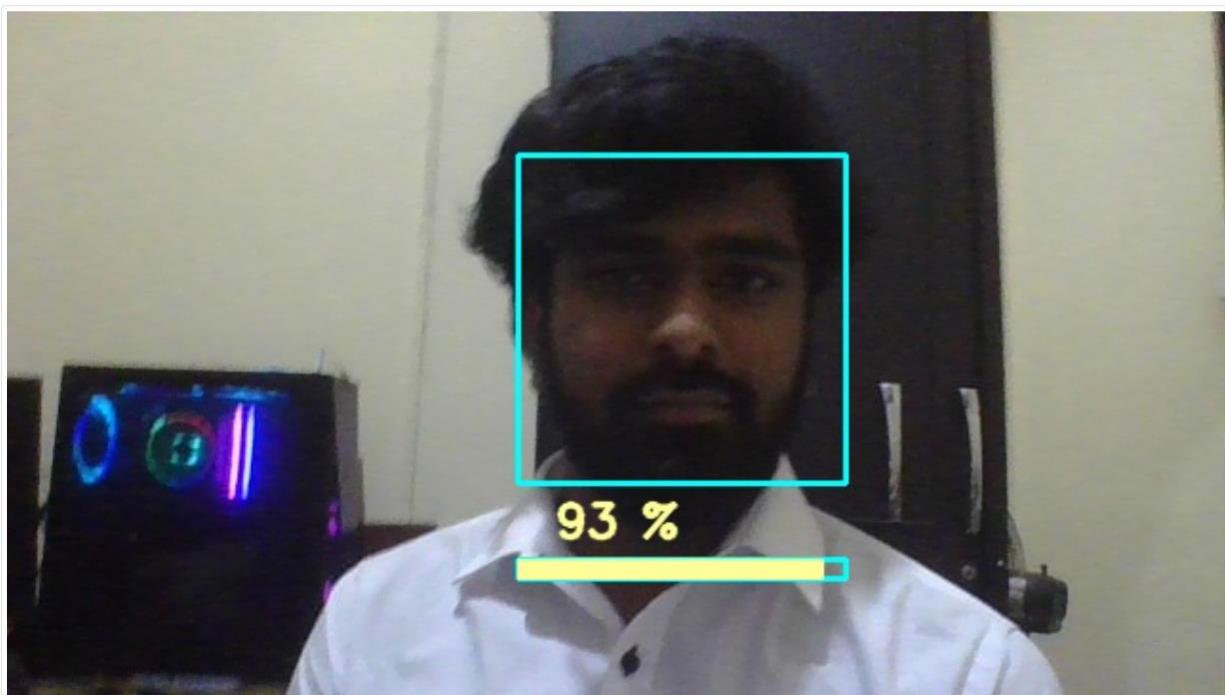
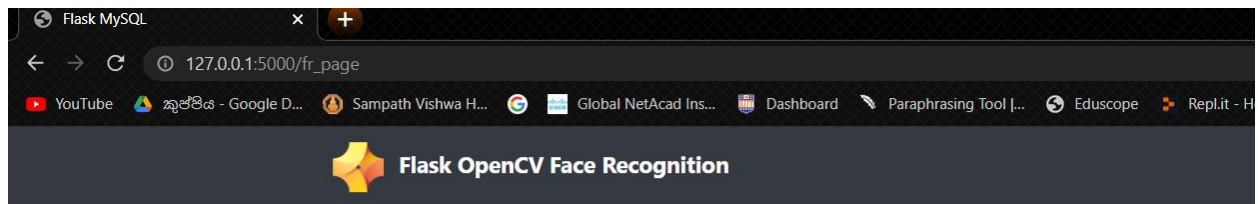
Person #	Name	Skill	Added
102	kasuna	SOFTWARE	17:17:56
102	kasuna	SOFTWARE	17:17:39

When face recognition is started user can see the progress of their scanning with a percentage that part is done by this code segmentation, if there are no data about person who in front of the camera it shows the unknown.

```
else:
    if not justscanned:
        cv2.putText(img, 'UNKNOWN', (x, y - 5), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0, 0, 255), 2, cv2.LINE_AA)
    else:
        cv2.putText(img, ' ', (x, y - 5), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0, 0, 255), 2, cv2.LINE_AA)

    if pause_cnt > 80:
        justscanned = False

    coords = [x, y, w, h]
    return coords
```



After 100% completed scanned data stores in the database and that data can use for create attendance reports also.

This application is using the database for get dataset and some inserting processes should be done by the application . so we used following codes for that .

```

251
252 @app.route('/countTodayScan')
253 def countTodayScan():
254     mydb = mysql.connector.connect(
255         host="localhost",
256         user="root",
257         passwd="",
258         database="flask_db"
259     )
260     mycursor = mydb.cursor()
261
262     mycursor.execute("select count(*) "
263                     " from accs_hist "
264                     " where accs_date = curdate() ")
265     row = mycursor.fetchone()
266     rowcount = row[0]
267
268     return jsonify({'rowcount': rowcount})
269
270
271 @app.route('/loadData', methods = ['GET', 'POST'])
272 def loadData():
273     mydb = mysql.connector.connect(
274         host="localhost",
275         user="root",
276         passwd="",
277         database="flask_db"
278     )
279     mycursor = mydb.cursor()
280

```

phpMyAdmin

Recent Favorites

Server: 127.0.0.1 » Database: flask_db

Structure SQL Search Query Export Import Operations Privileges Routines Events Triggers Tracking Des

Filters

Containing the word:

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> accs_hist		1	InnoDB	latin1_swedish_ci	32.0 KiB	-
<input type="checkbox"/> img_dataset		100	InnoDB	latin1_swedish_ci	16.0 KiB	-
<input type="checkbox"/> prs_mstr		1	InnoDB	latin1_swedish_ci	16.0 KiB	-
3 tables	Sum	102	InnoDB	utf8mb4_general_ci	64.0 KiB	0 B

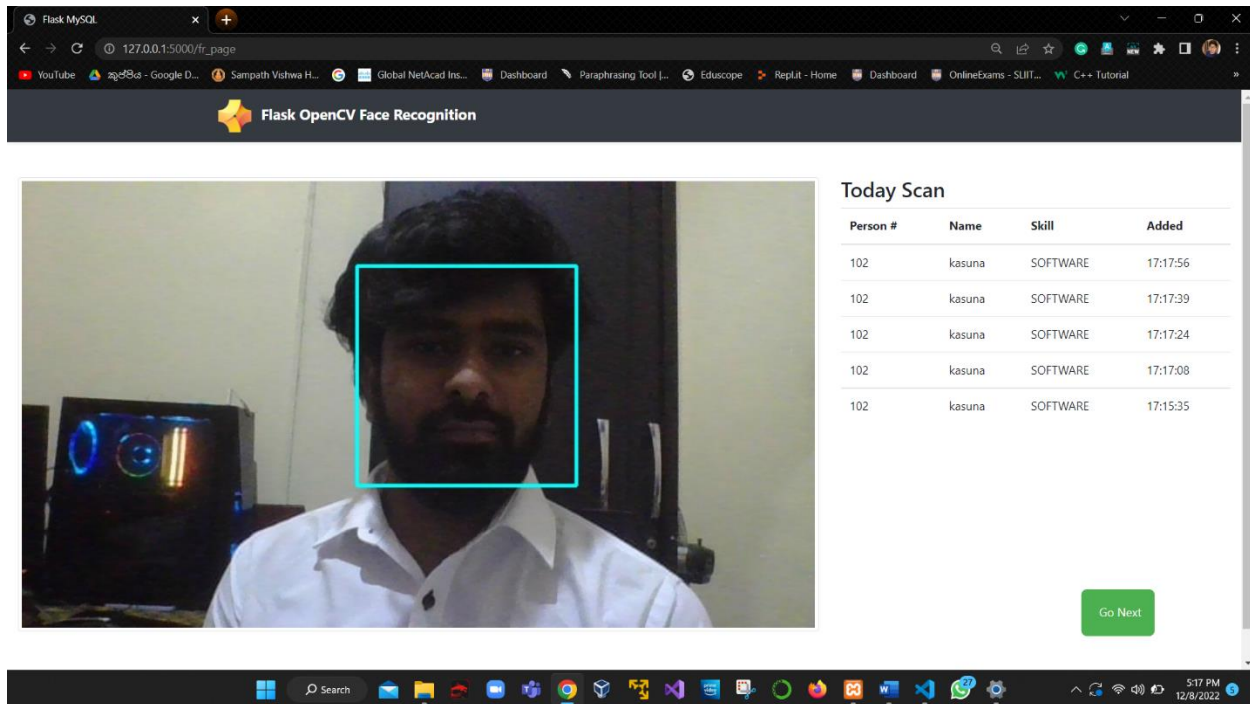
↑ ☐ Check all With selected:

Print Data dictionary

Create new table

Table name Number of columns

05.Main User interfaces and PHP



For face recognition page we need to display video feed in real time so we have used that code segments in python code and html page.

Python code

```
233
234 @app.route('/video_feed')
235 def video_feed():
236     # Video streaming route. Put this in the src attribute of an img tag
237     return Response(face_recognition(), mimetype='multipart/x-mixed-replace; boundary=frame')
238
```

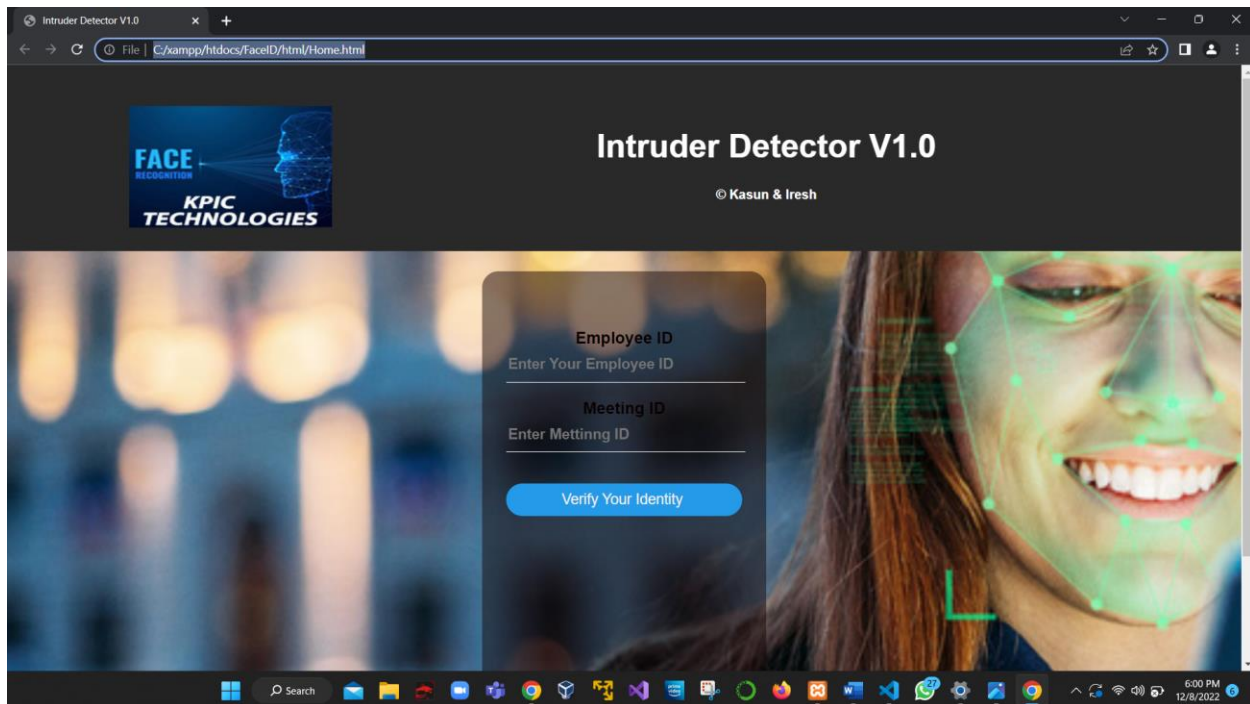
Html page

```
<!DOCTYPE html>
<html lang="en">
<head>
    {% include "head.html" %}
</head>
<body >

    {% include "navbar.html" %}

    <div class="container-fluid">
        <div class="row mt-5 mb-5">
            <div class="col-md-8">
                
            </div>
            <div class="col-md-4" style="height:500px; overflow-y:auto;">
                <h3>Today Scan</h3>
                <table class="table">
                    <thead>
                        <tr>
                            <th>Person #</th>
                            <th>Name</th>
                            <th>Skill</th>
                            <th>Added</th>
                        </tr>
                    </thead>
                    <tbody id="scandata">
```


Home page and how to use PHP to deal with the database



By using the face recognition we need to verify the persons through entering home page by using the employee details. In here we need to get from employee details for logging the system and identified the persons who are access to the meeting link. It can be process the user details to database and identified whether is that person in the data base for joining the meeting.

```
1 <!DOCTYPE html>
2 <html>
3 <title> Intruder Detector V1.0</title>
4 <link rel="stylesheet" href="css/style1.css">
5 <body>
6
7 <div class="header">
8 
9 <center><h1 style="text-shadow:1px 1px 0 #444"> Intruder Detector V1.0 <br></h1></center>
10 <center><h1 style="text-shadow:1px 1px 0 #444;font-size:medium"> © Kasun & Iresh <br></h1></center>
11 </div>
12 <br>
13
14 <center>
15 <div class="login-box">
16
17 <div class="formbox">
18 <div id="error"></div>
19 <form name="signup" id="form" onsubmit="return checkpassword()" action="php/signup.php" method="POST">
20 <p>Employee ID</p>
21 <input type="text" name="username" id="username" placeholder="Enter Your Employee ID" required>
22 <p>Meeting ID</p>
23 <input type="text" name="email" id="email" placeholder="Enter Metting ID" required>
24 <br><br>
25 <input type="submit" name="" value="Verify Your Identity ">
26 </form>
27 </div>
28 </div>
29 </div>
30 </center>
31
32
33
```

Html page for using getting details of the users when their entering to the home html page.

```

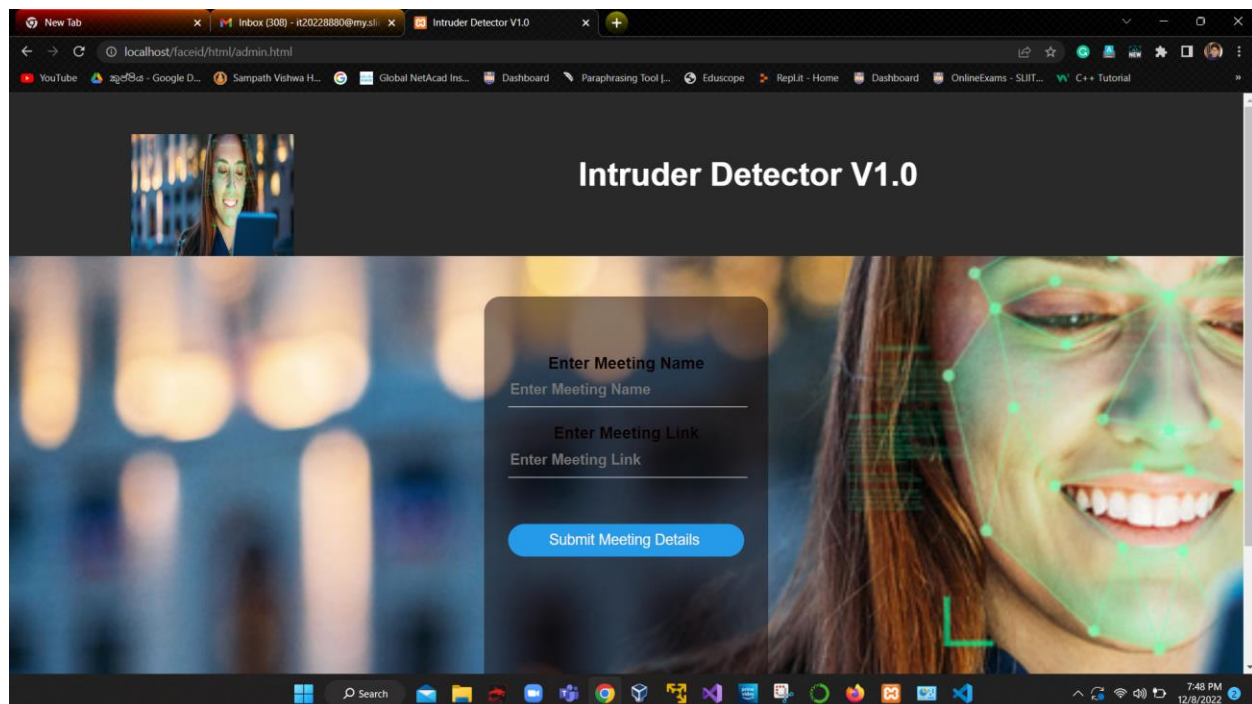
1 <html>
2 <head>
3 <title>PHP Test</title>
4 </head>
5 <body>
6 <?php
7     require 'config.php';
8
9     $name=$_POST["username"];
10    $email=$_POST["email"];
11
12
13
14    $sql="INSERT INTO login(Emp_id,M_id) VALUES ('$name','$email')";
15
16    if (mysqli_query($conn, $sql))
17    {
18        echo "<script type='text/javascript'>
19            alert('Welcome to Intruder Detection V1.0');
20            window.location = '../html/instructions.html';
21        </script>";
22    }
23    else
24    {
25        echo "Error: " . $sql . "<br>" . mysqli_error($conn);
26    }
27
28    mysqli_close($conn);
29
30
31 <?>
32 </body>
33 </html>

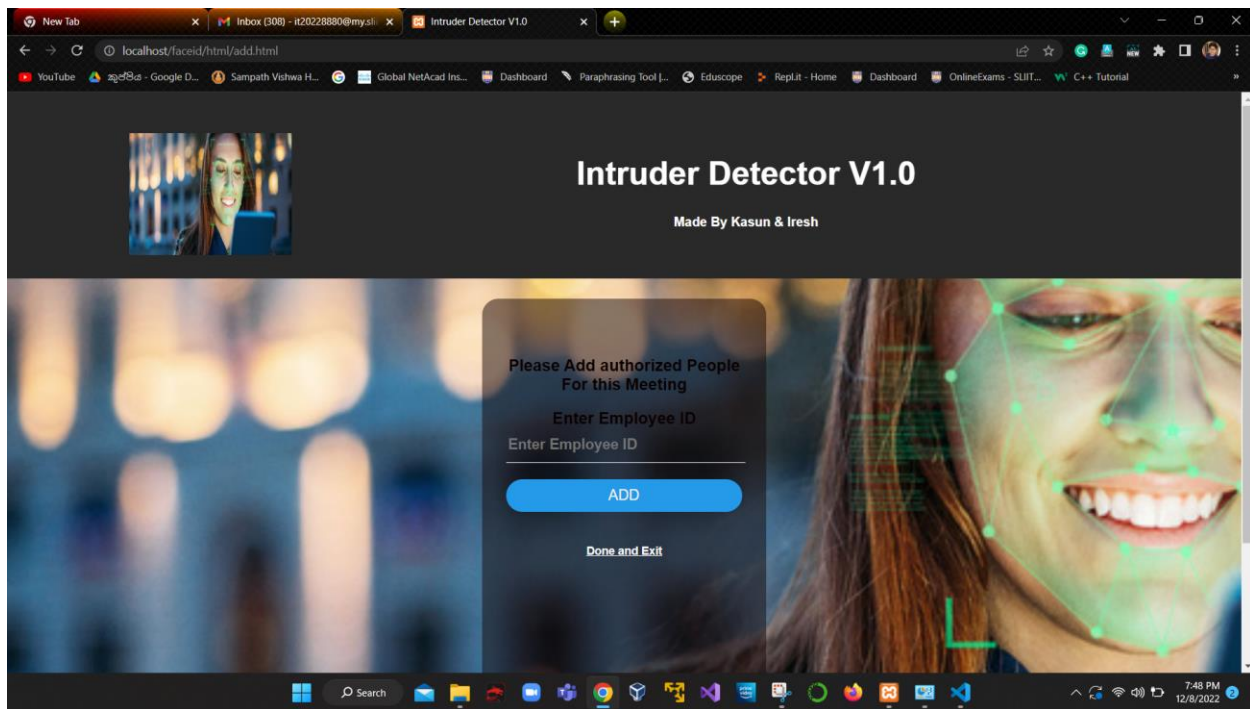
```

According to above php code when user entering the employee id and meeting id ,that data goes to the database with the time also .By this option organization can monitor who use this product and do some accountability based activities such as audits.

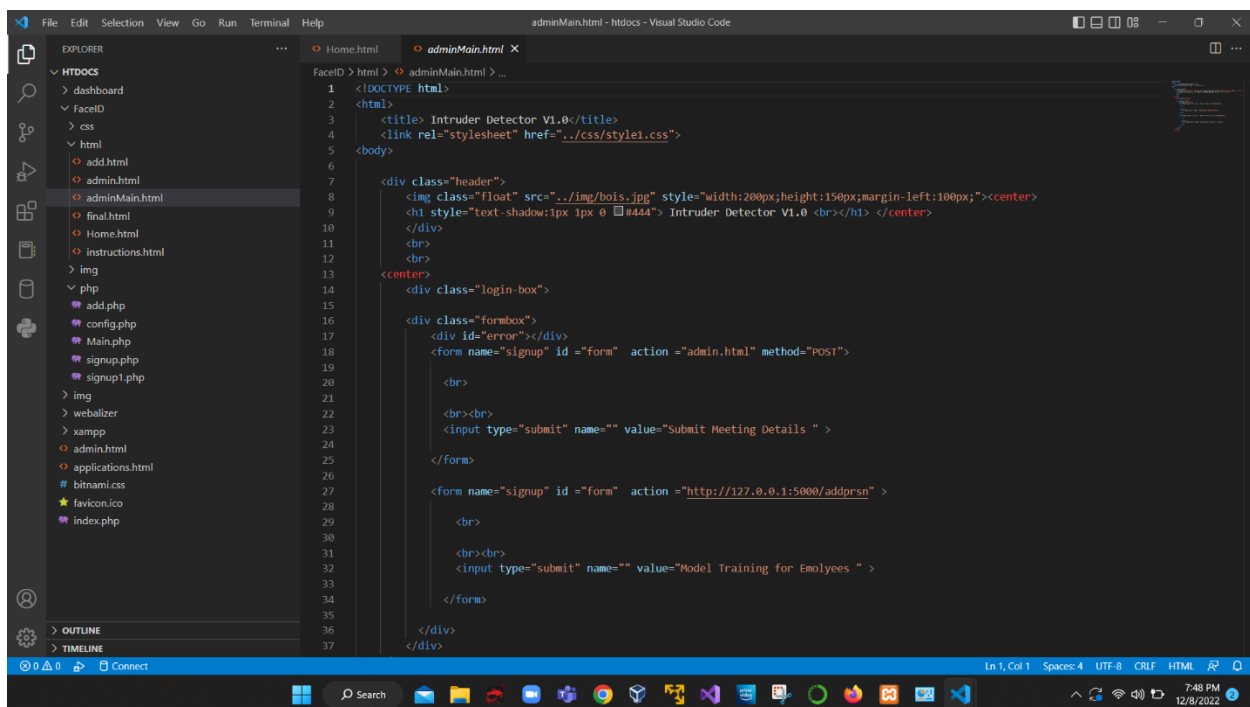
Ensuring the user details we need to create a database for storing the users who are logging for the system in that case we have to create sign up data base for when there user accessing within home page. It redirect for the database and store users data for identified for authenticated users. If its not in the authenticate user but storing details of all entering detail.

Administrator Pages

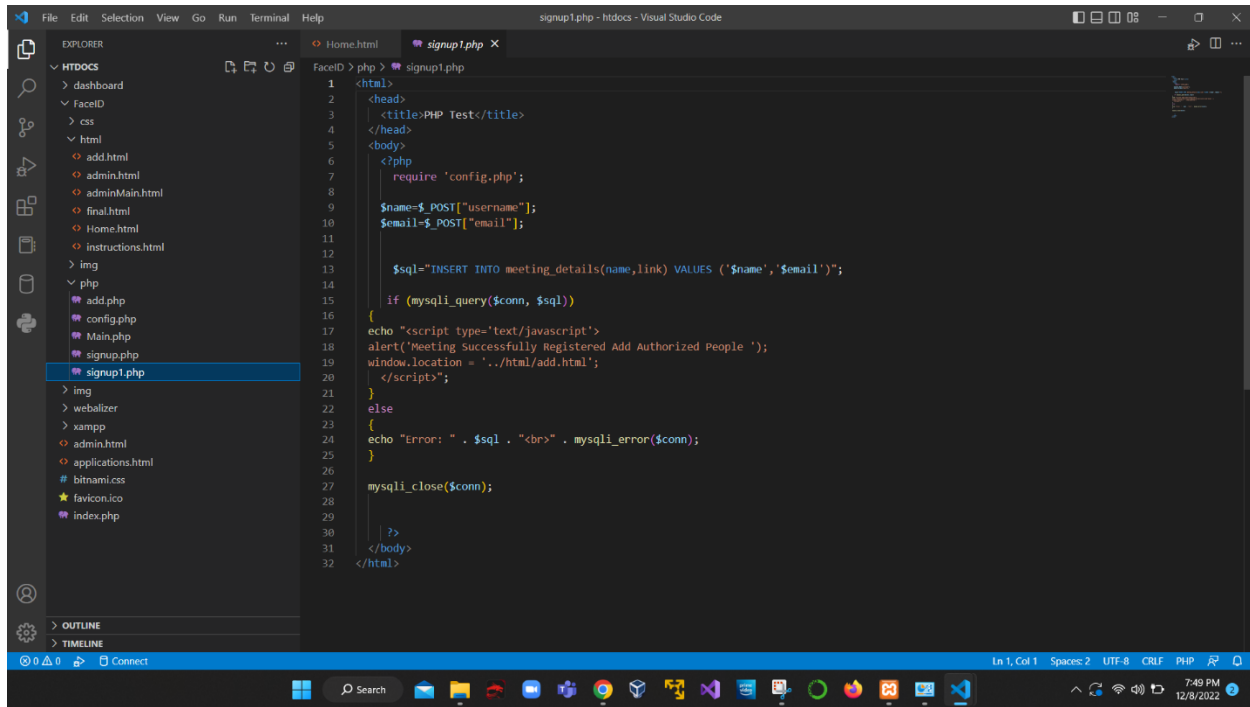




Administrator have to setting up meetings before the meeting happened so he need to enter meeting names and meeting links and application assigns them to a meeting id .As well as admin need to enter the authorized people to meeting that they are going to held.



By a simple forms admin can enter the meeting data and application is using the php for store that data to the database .following coding segments help to do that process.



```
1 <html>
2 <head>
3 | <title>PHP Test</title>
4 </head>
5 <body>
6 <?php
7 |     require 'config.php';
8
9 |     $name=$_POST["username"];
10 |    $email=$_POST["email"];
11
12 |
13 |    $sql="INSERT INTO meeting_details(name,link) VALUES ('$name','$email')";
14
15 |    if (mysqli_query($conn, $sql))
16 |    {
17 |        echo "<script type='text/javascript'>
18 |        alert('Meeting Successfully Registered Add Authorized People ');
19 |        window.location = '../html/add.html';
20 |        </script>";
21 |    }
22 |    else
23 |    {
24 |        echo "Error: " . $sql . "<br>" . mysqli_error($conn);
25 |    }
26
27 |    mysqli_close($conn);
28
29 | }>
30 </body>
31 </html>
```

Full Code of Application

App.py

[illegible]

[illegible]

```

169         return img
170
171     faceCascade = cv2.CascadeClassifier("C:/Users/iresh/OneDrive/Desktop/New folder (10)/faceRecognition_files/resources/haarcascade_frontalface_default.xml")
172     clf = cv2.face.LBPHFaceRecognizer_create()
173     clf.read("classifier.xml")
174
175     wCam, hCam = 400, 400
176
177     cap = cv2.VideoCapture(0)
178     cap.set(3, wCam)
179     cap.set(4, hCam)
180
181     while True:
182         ret, img = cap.read()
183         img = recognize(img, clf, faceCascade)
184
185         frame = cv2.imencode('.jpg', img)[1].tobytes()
186         yield (b'--frame\r\n'
187              | b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n\r\n')
188
189         key = cv2.waitKey(1)
190         if key == 27:
191             break
192
193
194
195 @app.route('/')
196 def home():
197     mycursor.execute("select prs_nbr, prs_name, prs_skill, prs_active, prs_added from prs_mstr")
198     data = mycursor.fetchall()
199
200     return render_template('index.html', data=data)
201
202 @app.route('/addprsn')
203 def addprsn():
204     mycursor.execute("select ifnull(max(prs_nbr) + 1, 101) from prs_mstr")
205     row = mycursor.fetchone()
206     nbr = row[0]
207     # print(int(nbr))
208
209     return render_template('addprsn.html', newnbr=int(nbr))
210
211 @app.route('/addprsn_submit', methods=['POST'])
212 def addprsn_submit():
213     prsnbr = request.form.get('txtnbr')
214     prsname = request.form.get('txtname')
215     prsskill = request.form.get('optskill')
216
217     mycursor.execute("""INSERT INTO `prs_mstr` (`prs_nbr`, `prs_name`, `prs_skill`) VALUES
218                      ('{}', '{}', '{}')""".format(prsnbr, prsname, prsskill))
219     mydb.commit()
220
221     # return redirect(url_for('home'))
222     return redirect(url_for('vfdataset_page', prs=prsnbr))
223
224 @app.route('/vfdataset_page<prs>')
225 def vfdataset_page(prs):
226     return render_template('gendataset.html', prs=prs)
227
228 @app.route('/vidfeed_dataset/<nbr>')
229 def vidfeed_dataset(nbr):
230     #Video streaming route. Put this in the src attribute of an img tag
231     return Response(generate_dataset(nbr), mimetype='multipart/x-mixed-replace; boundary=frame')
232
233
234 @app.route('/video_feed')
235 def video_feed():
236     # Video streaming route. Put this in the src attribute of an img tag
237     return Response(face_recognition(), mimetype='multipart/x-mixed-replace; boundary=frame')
238
239 @app.route('/fr_page')
240 def fr_page():
241     """Video streaming home page."""
242     mycursor.execute("select a.accs_id, a.accs_prsn, b.prs_name, b.prs_skill, a.accs_added "
243                    | " from accs_hist a "
244                    | " left join prs_mstr b on a.accs_prsn = b.prs_nbr "
245                    | " where a.accs_date = curdate() "
246                    | " order by 1 desc")
247     data = mycursor.fetchall()
248
249     return render_template('fr_page.html', data=data)
250
251
252 @app.route('/countTodayScan')
253 def countTodayScan():
254     mydb = mysql.connector.connect(
255         host="localhost",
256         user="root",
257         passwd="",
258         database="flask_db"
259     )
260     mycursor = mydb.cursor()
261
262     mycursor.execute("select count(*) "
263                    | " from accs_hist "
264                    | " where accs_date = curdate() ")
265     row = mycursor.fetchone()
266     rowcount = row[0]
267
268     return jsonify({'rowcount': rowcount})
269
270
271 @app.route('/loadData', methods = ['GET', 'POST'])
272 def loadData():
273     mydb = mysql.connector.connect(
274         host="localhost",
275         user="root",
276         passwd="",
277         database="flask_db"
278     )
279     mycursor = mydb.cursor()
280
281     mycursor.execute("select a.accs_id, a.accs_prsn, b.prs_name, b.prs_skill, date_format(a.accs_added, '%H:%i:%s') "
282                    | " from accs_hist a "
283                    | " left join prs_mstr b on a.accs_prsn = b.prs_nbr "
284                    | " where a.accs_date = curdate() "
285                    | " order by 1 desc")
286     data = mycursor.fetchall()
287
288     return jsonify(response = data)
289
290 if __name__ == "__main__":
291     app.run(host='127.0.0.1', port=5000)

```

Design and Implementation constraints

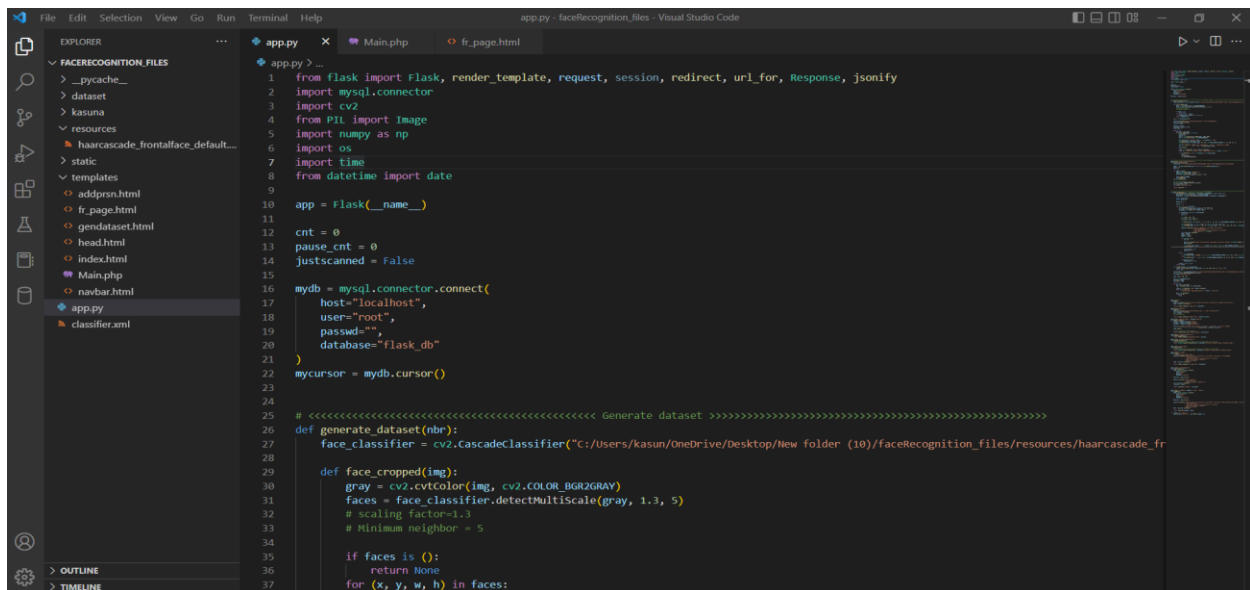
Intruder detector can run on windows operating systems

System (server) requirements : Minimum 8GB RAM (Random Access Memory)

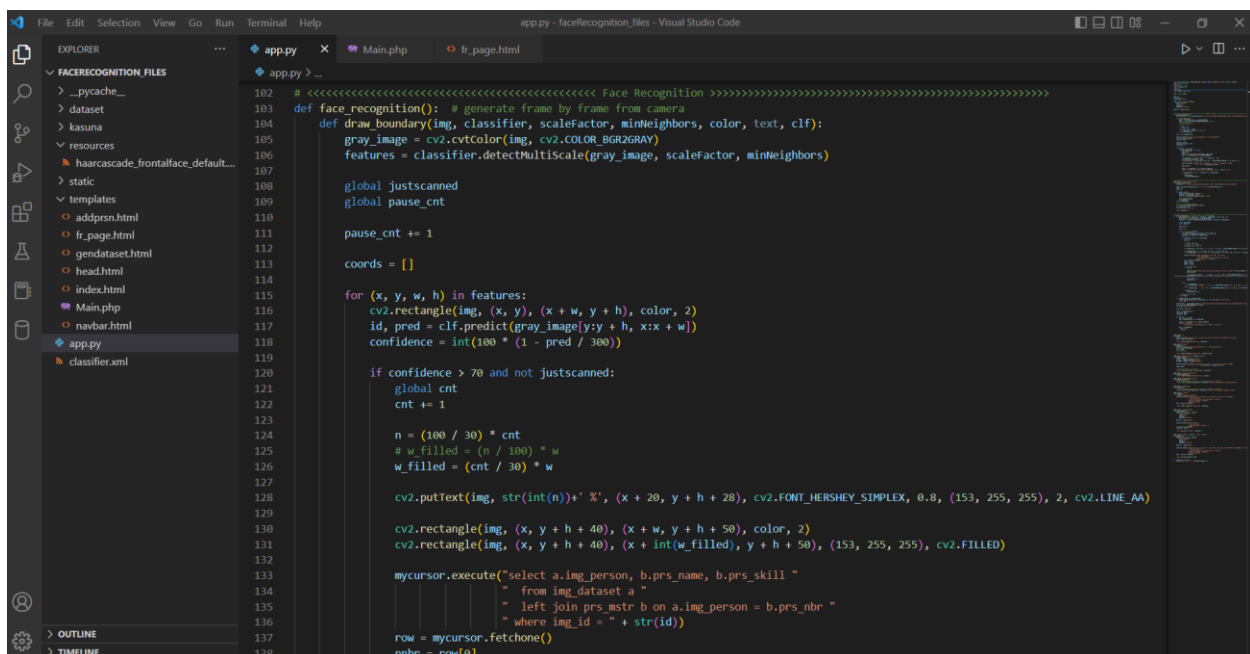
Average CPU usage

Intruder detector v1.0 is powered by python language with flask environment .It need python 3.9 + to run the program. Database should be maintained .

According to the customer requirements can be modified some functions.



```
1 from flask import Flask, render_template, request, session, redirect, url_for, Response, jsonify
2 import mysql.connector
3 import cv2
4 from PIL import Image
5 import numpy as np
6 import os
7 import time
8 from datetime import date
9
10 app = Flask(__name__)
11
12 cnt = 0
13 pause_cnt = 0
14 justscanned = False
15
16 mydb = mysql.connector.connect(
17     host="localhost",
18     user="root",
19     passwd="",
20     database="flask_db"
21 )
22 mycursor = mydb.cursor()
23
24
25 # <----- Generate dataset ----->
26 def generate_dataset(nbr):
27     face_classifier = cv2.CascadeClassifier("C:/Users/kasun/OneDrive/Desktop/New folder (10)/faceRecognition_files/resources/haarcascade_fr
28
29     def face_cropped(img):
30         gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
31         faces = face_classifier.detectMultiScale(gray, 1.3, 5)
32         # scaling factor=1.3
33         # Minimum neighbor = 5
34
35         if faces is ():
36             return None
37         for (x, y, w, h) in faces:
```



```
102 # <----- Face Recognition ----->
103 def face_recognition(): # generate frame by frame from camera
104     def draw_boundary(img, classifier, scaleFactor, minNeighbors, color, text, clf):
105         gray_image = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
106         features = classifier.detectMultiScale(gray_image, scaleFactor, minNeighbors)
107
108         global justscanned
109         global pause_cnt
110
111         pause_cnt += 1
112
113         coords = []
114
115         for (x, y, w, h) in features:
116             cv2.rectangle(img, (x, y), (x + w, y + h), color, 2)
117             id, pred = clf.predict(gray_image[y:y + h, x:x + w])
118             confidence = int(100 * (1 - pred / 300))
119
120             if confidence > 70 and not justscanned:
121                 global cnt
122                 cnt += 1
123
124                 n = (100 / 30) * cnt
125                 # w_filled = (n / 100) * w
126                 w_filled = (cnt / 30) * w
127
128                 cv2.putText(img, str(int(n)) + ' %', (x + 20, y + h + 28), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (153, 255, 255), 2, cv2.LINE_AA)
129
130                 cv2.rectangle(img, (x, y + h + 40), (x + w, y + h + 50), color, 2)
131                 cv2.rectangle(img, (x, y + h + 40), (x + int(w_filled), y + h + 50), (153, 255, 255), cv2.FILLED)
132
133                 mycursor.execute("select a.img_person, b.prs_name, b.prs_skill "
134                                 " from img_dataset a "
135                                 " left join prs_mstr b on a.img_person = b.prs_nbr "
136                                 " where img_id = " + str(id))
137                 row = mycursor.fetchone()
138                 pnbr = row[0]
```


User interfaces are very simple and any user can identify the process of the authorization part .development team is created the user interfaces to understand the every step to verify the people

Implementation plan

Step one : identify the business scope

In the step one team should identify what are the critical information in the organization and what are the content of their meetings ,team should consider

01.content of meetings

02.Are they discuss sensitive information in their online meetings?

03.Who involved into for discuss important facts of organization on meeting?

04.Audit the attendance of the meetings previous

Step 02 : Implementation of the product and enter the necessary data to database

In this step , team implement the intruder detector system and enter the data that identified in step 01 and check whether entered data is correct .

Step 03 : Training models for face recognition

In this phase , application need the 100 photos of each of employees to train the model to identify in face recognition part .So all employees need to train their data and enter the database as registering to the application. If there is no data in database that employee can not attend any meeting in the organization.

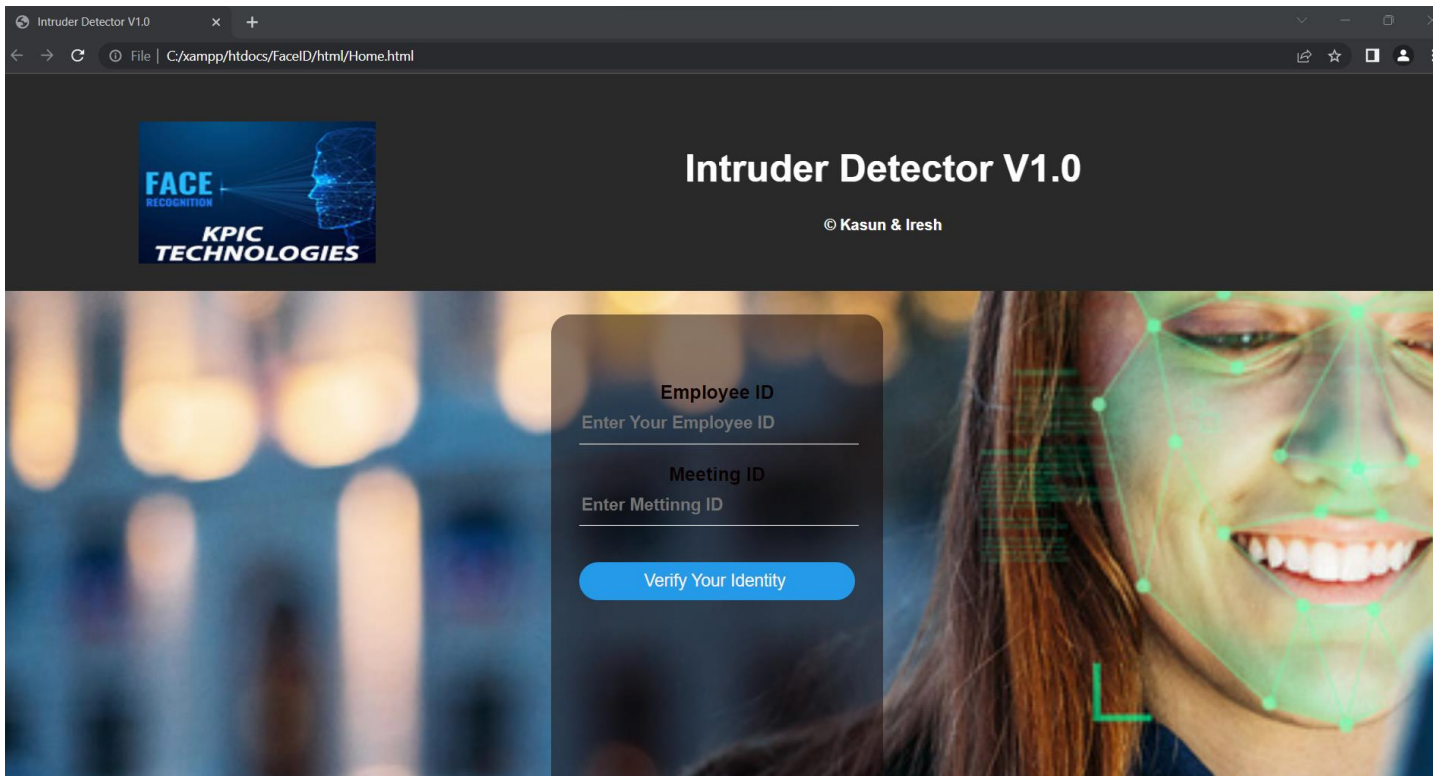
Step 04 : Test the environment

After entering the all necessary data to databases , using some employees as test cases should be tested application properly working or not.

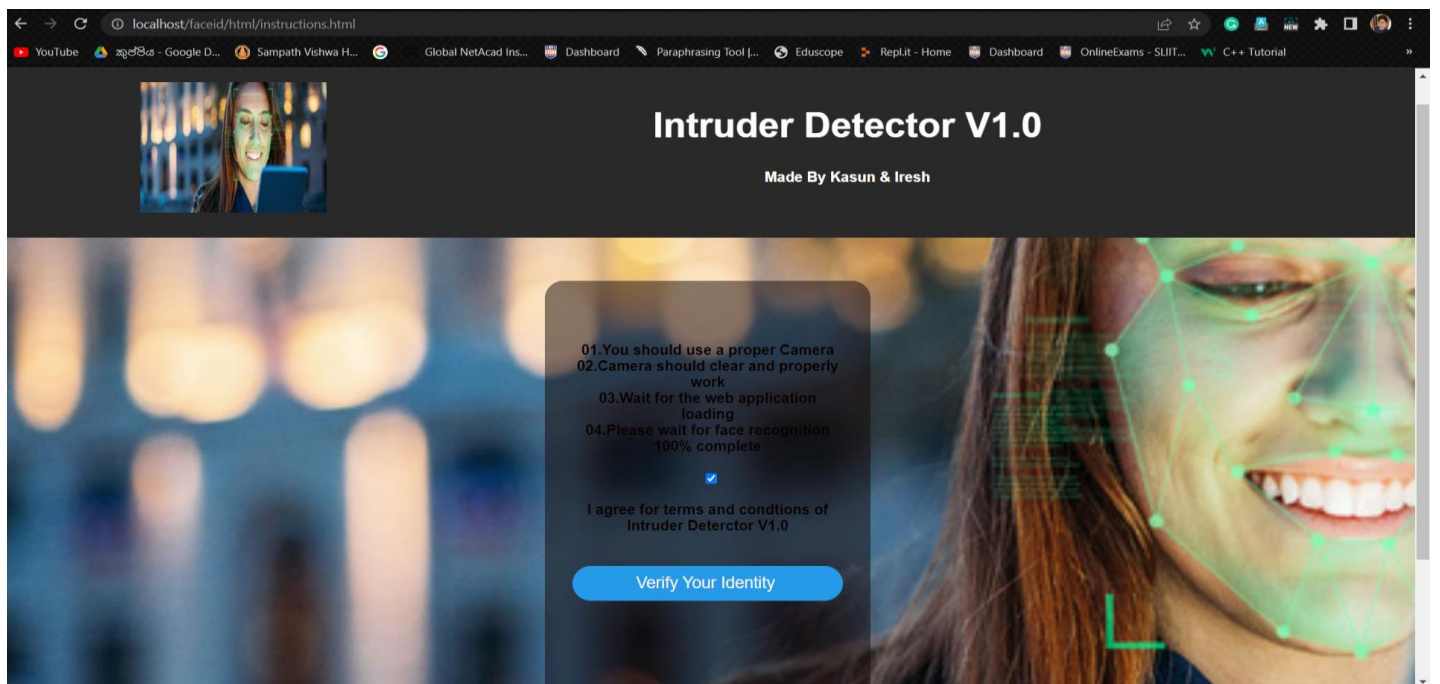
Step 05 : User awareness and training

At the last step , all employees should awareness about the policies .As well as they should training how to work with this application.

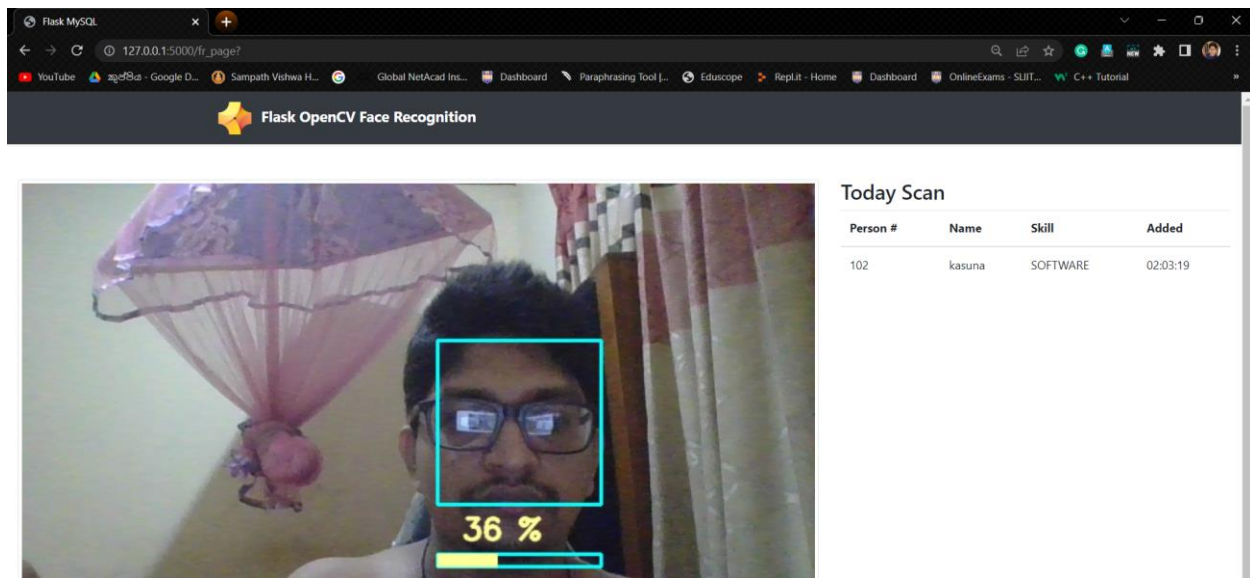
How Intruder detector works



This is the home page of the intruder detector , user have to enter the employee number and given meeting id.



After agree of the terms and conditions user can verify their identity.

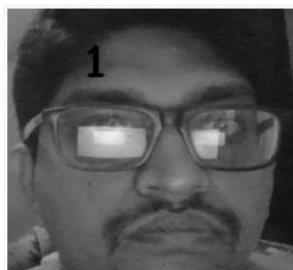


After 100% completion of recognition user can click go next button and application is redirects automatically to the meeting.

If you are not authenticated by the face recognition you can't go the meeting application prompts the error message.

Train the model

Generate Dataset 103



Training

In this stage takes the 100 photos of one user to training model ,that data stored in the data , this data is used in the face recognition authorization process.

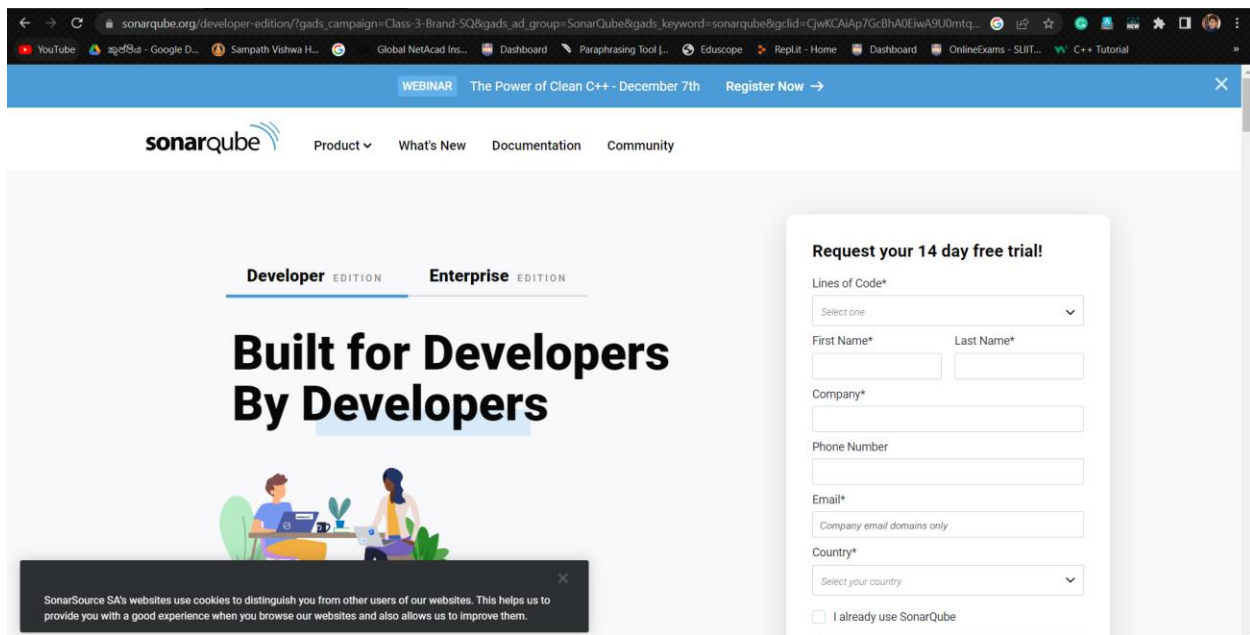
Secure testing

By locating security flaws and vulnerabilities in source code, application security testing (AST) strengthens applications' resistance to security threats.

AST was initially a manual procedure. AST needs to be automated in the modern day due to the increasing modularity of corporate software, the enormous amount of open source components, and the great number of known vulnerabilities and threat vectors. The majority of businesses combine various application security technologies.

White box testing

We can use tool such as SonarQube to identify vulnerabilities of the code reviewing



The screenshot shows the SonarQube website interface. At the top, there's a navigation bar with links for Product, What's New, Documentation, and Community. Below this, the main heading reads "Built for Developers By Developers" with an illustration of two people working on laptops. On the right side, there's a prominent "Request your 14 day free trial!" form. The form includes fields for "Lines of Code*" (a dropdown menu), "First Name*", "Last Name*", "Company*", "Phone Number", "Email*" (with a note "Company email domains only"), and "Country*" (a dropdown menu). At the bottom of the form, there's a checkbox labeled "I already use SonarQube". A cookie consent banner is visible at the bottom left of the page.

Black box testing

run code, inspect it at runtime, and look for problems that might be security flaws. This can involve problems with query strings, requests and answers, script usage, memory leaks, management of cookies and sessions, authentication, running third-party components, data injection, and DOM injection, among other things.

Development team ran the code and it works perfectly according to the purpose so intruder detector could achieve the black box testing easily.

Future improvements of application

Monitor user activities and webinar platform for improvement

According to the implementation of the application, there must be some improvement in the application's effectiveness and efficiency. To do so, we must implement a system for monitoring user logs and user activities. Ensure that when an authenticated user logs in to the system via our application, a record of the user's activities during the specified time period is kept. For the user to use the application, it must have disclosure activations. When a user needs access to a user log or other types of tracking, he should send an email to management. An authenticated user can view the activities performed in the application during their session, as well as to applications running on the system, maintain a webinar template for getting user logs and a database for the identification of user activities.

Conclusion

When it comes to sensitive information, confidentiality refers to the idea and practice of keeping it secret until the owner or data custodian explicitly consents to sharing it with another party. Another definition of confidentiality is the request to uphold the rule and custom.

The primary objective of the application is to preserve the confidentiality. With the face recognition system, only authorized people can access the meeting and they can get the confidential facts of the organization.

References

- [1] N. A. A. R. Z. M. Z. A. Behzad Shoarian Satari, "Face Recognition for Security Efficiency".
- [2] W. L. W. a. S. S. D. M. A. Dabbah, "Secure Authentication for Face Recognition".
- [3] "fitsmallbusiness.com," [Online]. Available: https://fitsmallbusiness.com/remote-work-statistics/?__cf_chl_tk=LG8fgdyi90h0KbM52KkvpnzaPFdzoxCe7S4mqSIV1Ys-1670530413-0-gaNycGzNB30.
- [4] "Webinar Statistics, Trends And Facts 2023," [Online]. Available: <https://abdalslam.com/webinar-statistics>.
- [5] "www.techtarget.com," [Online]. Available: <https://www.techtarget.com/searchenterpriseai/definition/facial-recognition>.
- [6] M. K. D. A. S. a. D. R. P. Nirmalya Kar, "Study of Implementing Automated Attendance System".
- [7] "Top 11 Facial Recognition Software in 2021," [Online].
- [8] C. Mingsung, "Research on the Application of Face Recognition System".
- [9] "www.sonarqube.org," [Online]. Available: https://www.sonarqube.org/developer-edition/?gads_campaign=Class-3-Brand-SQ&gads_ad_group=SonarQube&gads_keyword=sonarqube&gclid=CjwKCAiAs8acBhA1EiwAgRFdwwp1Ao4fQsofPGU_6CeXDPIkcnaCvRw0NNvKGxoj8BJluy5GIUvLDBoCuksQAvD_BwE.
- [10] "www.guru99.com," [Online]. Available: <https://www.guru99.com/black-box-testing.html>.