CSE 3013: Artificial Intelligence

Slot:- A2

Submitted to Prof. Rajeshkannan R

TEAM 9

20BCE2870 – Gopesh Kumar Pathak

20BCE2119 – Anshuman Gupta

20BCE0478 - Prince Choudhary

20BCE2136 - G Aditya Kumar

Multimodal security authentication using real-time face recognition and speech-to-text verification

1. Abstract :

Passwords are not a foolproof method of keeping unwanted people out of your system. Biometric authentication systems have proven to be extremely useful in such situations. In our proposed model, we'll employ a hybrid model that identifies and grants access to people based on both their voice and their face data. Our approach will employ artificial intelligence to recognize users even if their faces or voices alter slightly. The project's goal is to make biometric authentication available to web-based applications providing a four-factor security authentication rather than the existing two-layer model commonly used. This can be accomplished with the help of various commonly used computer vision techniques and speech recognition APIs. Some text-based attacks, such as password brute force and SQL injection, will not be possible on the system. As a result, using Multimodal authentication is always a good idea for a smooth authentication process.

2. Introduction :

Data security has become a matter of great concern over the years, especially in today's time with all the data leaks such as the 2014 Facebook-Cambridge Leak, the 2018 Twitter Leak, the 2021 Twitch Leak, etc. releasing very sensitive data to the public, data that was personal to the users specifically, causing a large outcry as cybercrimes are far from over and only increase in intensity. Therefore, databases need to be protected securely.

One of the most critical tasks that a database administrator has to carry out is keeping track of the system's configuration. Many users have been observed to reuse their passwords, so a 'strong' password is not a good enough security measure against unauthorized access. It can't be relied upon to guard the framework from gate-crashers and in such cases, implementing a biometric authentication system is extremely useful.

Running a pre-trained basic AI program has become a straightforward and trivial task as computer processing power has increased. Running an AI program that can operate as an authentication program is a highly viable option now that the computing capacity is accessible.

By merging our AI applications with the existing security measures prevalent in today's time, we aim to present a four-factor security authentication measure.

3. Literature Review :

Paper Title Journal Details	Method/ Algorithm	Challenges	Observations
Thiang and Dhanny Wijaya, Limited Speech Recognition for Controlling Movement of Mobile Robot Implemented on ATmega162 Microcontroller. [15]	Input signals were sampled directly from the microphone and then the extraction was done by Linear Predictive Coding (LPC) and Artificial Neural Network (ANN)	The accuracy output is at 84 percent only. Very inaccurate in establishing a speech if external factors are involved such as if a person is chewing.	A pioneer in speech recognition AI software, the project while laying the framework needs significant work to provide a good model.
V. Blanz and T. Vetter, "Face recognition based on fitting a 3D morphable model," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1063-1074, Sept. 2003. **[13]**	The face recognition system presented in this paper combines deformable 3D models with a computer graphics simulation of projection and illumination. This makes intrinsic shape and texture fully independent of extrinsic parameters. Given a single image of a person, the algorithm automatically estimates 3D shape, texture, and all relevant 3D scene parameters.	Ignores glasses, beards, facial hair, and any kind of strands of hair on the model, thereby not being entirely accurate	The 3D morphable model is indeed a powerful and versatile representation for human faces and is straightforward to extend the morphable model to different ages, ethnic groups, and facial expressions by including face vectors from more 3D scans.

Shaik Subhani. Pattern Recognition of Speech Signals Using Curvelet Transform and Artificial Intelligence, IJCRT | Volume 6, Issue 2 April 2018. **[14]**	Authority speech recognition system based on curvelet transform and artificial neural network techniques to enhance the recognition rate.	Can get too complex in its proposed theory and structure, is difficult to implement, and is still not wholly accurate.	In terms of efficiency and a fast performance rate, this model proved to be quite successful, although its complexity hampers its chances to be used for our model
Salama AbdELminaam D, Almansori AM, Taha M, Badr E (2020) A deep facial recognition system using computational intelligent algorithms. PLoS ONE 15(12): e0242269. **[7]**	In the beginning, the face detector is utilized on videos or images to detect faces. The prominent feature detector aligns each face to be normalized and recognized with the best match. Finally, the face images are fed into the FR module with the aligned results.	One of the main challenges in FR applications is representing variation; in this paper, summarized are the face-processing deep methods for poses. Similar techniques can solve other changes. The face-processing techniques are categorized as "one-to-many augmentation" and "many-to-one normalization"	The experimental outcomes of the developed FR system and its comparison with various other techniques are presented and expanded upon. It has been noted that the outcomes of the proposed algorithm outperformed most of its peers in this department, especially in terms of precision.
L. Li, X. Mu, S. Li and H. Peng, "A Review of Face Recognition Technology," in IEEE Access, vol. 8, pp. 139110-139120, 2020.	Principal Component Analysis. Linear Discriminant Analysis.	PIE problem Non-negative Matrix Factorization	Due to the automated features of face recognition technology, similar related information may be processed or decided

[2]			through automation, lacking transparency and not easy to supervise even in the event of errors or discrimination.
W. -W. Koc, Y. -T. Chang, J. -Y. Yu and T. -U. İk, "Text-to-Speech with Model Compression on Edge Devices," 2021 22nd Asia-Pacific Network Operations and Management Symposium (APNOMS), 2021, pp. 114-119. **[1]**	Parameter Quantization Model Pruning Low-rank matrix approximation	The challenge being posed is for the speech synthesis recognition to be ported to edge devices with the advent of the fifth-generation mobile communication making it difficult.	In terms of model optimization, the use of parameter quantization and structured pruning for model compression has successfully reduced the model file size by 86%, increased the inference speed by 1.91 times and reduced the memory usage by nearly a half.
K. . -F. Lee, H. . -W. Hon and R. Reddy, "An overview of the SPHINX speech recognition system," in IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 38, no. 1, pp. 35-45, Jan. 1990. **[4]**	HMMs are the predominant approach to speech recognition. The success of HMM's is largely due to the forward-backward reestimation algorithms, which is a special case of the EM algorithm. Every iteration of the algorithm modifies the parameters to increase the probability of the training data until a	Speaker independence has been viewed as the most difficult constraint to overcome. This is because most parametric representations of speech are highly speaker dependent, and a set of reference patterns suitable for one speaker may perform poorly for another speaker. Researchers have found that errors increased by 300-500 percent when a	Large-vocabulary speaker-independent continuous speech recognition is feasible. We believe that with a powerful learning paradigm, the performance of a system can always be improved with more training data, subject to our ability to make the models more sophisticated. The sophisticated modeling techniques introduced in this paper reduced the error rate of the baseline system by as much as 85 percent,

	local maximum has been reached.	speaker-dependent system is trained and tested in speaker-independent mode. This training phase typically requires several hundred sentences. While speaker-trained systems are useful for some applications, they are inconvenient, less robust, more wasteful, and simply unusable for some applications.Speaker-independent systems must train on less appropriate training data. However, many more data can be acquired, which may compensate for being speaker-dependent. In other words, they require a speaker to “train” the system before reasonable performance can occur	resulting in inaccuracies of 71, 94, and 96 percent for a 997-word vocabulary with grammars of perplexity 997, 60, and 20.
Kortli, Y.; Jridi, M.; Al Falou, A.; Atri, M. Face Recognition Systems: A Survey. *Sensors* 2020, *20*, 342.	The basic purpose of Holistic methods is to represent the face image with a pixel matrix, which is	Pose variations, various lighting conditions, facial expressions, and low resolution.	Easy to implement, allowing an analysis of images in a difficult environment in real-time. Invariant to

[6]	frequently converted to feature vectors. The basic goal of these approaches is to find distinguishing characteristics. To extract local features, local appearance-based techniques are applied. In order to extract the features centered on these places, key-points-based algorithms.	Different illumination conditions, scaling, facial expressions.	size, orientation, and lighting When frontal views of faces are used, these techniques provide good performance. Recognition is effective and simple. Dimensionality reduction represents global information.
D. Mu, T. Zhu, G. Xu, H. Li, D. Guo, and Y. Liu, "Attention Based Speech Model for Japanese Recognition," 2019 IEEE International Conference on Smart Internet of Things (SmartIoT), 2019, pp. 402-406. **[3]**	Attention Mechanism: It is the matching degree between encoding input and decoding output. LSTM is the big breakthrough in the RNN model.	The challenge is not just to improve the accuracy of speech recognition but also the parallel efficiency. However, due to the whole encoding and decoding process, based on recurrent neural networks, there are still limitations of RNN in Parallel Computing.	Attention's model was used in the decoding process and can get different attention scores according to time series, to save all the attention history. It can also be applied in Japanese phoneme and text alignment tasks.

Shwetank Arya, Neeraj Pratap, Karamjit Bhatia,Future of Face Recognition: A Review,Procedia Computer Science,Volume 58,2015,Pages 578-585,ISSN 1877-0509. **[5]**	The FR methods in IR have similarities with VS but are less complex in comparison to it. Mostly the appearance based methods fully support the complicated statistical techniques rather than the inclusion of data specific knowledge.	IR faces several challenges of FR methods such as opaqueness of eyeglasses and the dependence of the acquired data on the emotional and physical condition of the subject. Another problem in FR is time-lapse.	IR imaging attracts the researchers to pay attention in multi-dimensional imaging systems to get more accurate results in the unfavorable conditions like object illumination, expression changes, facial disguises and dark environments.

4. PROBLEM STATEMENT :

As of January 2021, there are 4.66 billion users connected online to the internet, making use of data storage, and accessing crucial information. This number is just talking about the individuals in terms of the population, not in terms of work profiles or normal user profiles which must be significantly higher in comparison considering the amount of work that is to be done using laptops and the like these days. With so many billions of profiles, and the work that must be stored for corporations, industries, agencies etc., **data security is of the utmost importance,** and herein lies the **problem.**

Even in an individualistic demographic data privacy is valued by every user, therefore having secure access to their data isn't just a **matter of concern** for the companies but also for the regular casual user who might want to access the internet for entertainment purposes.

The concept of data privacy and security isn't a shocking revelation as it is a fairly obvious concern of the masses and is something that has been implemented since the onset of production of computers, however how secure the security measures are have been a concern for decades.

With the technological boom of the 21st century, the casual individual possesses **knowledge enough to hack databases and acquire personal information. Data breaches** and **invasion of**

privacy has become a worrisome concern as sensitive information being leaked has become commonplace.

IT Governance discovered 1,243 security incidents in 2021, which accounted for 5,126,930,507 breached records. That represents an 11% increase in security incidents compared to 2020 (1,120). By contrast, there was a significant decrease in the number of breached records over the same period (20.1 billion).

Clearly this shows that the current security measures in place are not effective at all and need proper protection.

Currently, in a majority of sites and applications one layer authentication is used as a precautionary security measure to make sure the user is secure in their data and can access it. The one-layer authentication system works by asking for a username/email address along with a designated/user chosen password to enter into the system and access the data present.

This can be described as **the barest minimum of security offered**, while in the early 2000s seemed impassable, in today's time it's hardly a difficult procedure to overcome by people well versed in the field, let alone top hackers and cyber criminals.

The only major solution offered in today's time is a second layer of authentication in the form of OTPs (One Time Passwords) making use of the smartphone's abundance of usage by making the user enter a second code to access their data. However, such measures are just not enough to secure the data in the wake of the aforementioned technological boom as **SQL injections and more advanced cyber warfare rage on the internet.**

To combat the chaos caused by the technological boom, it is only fair to counter it with advancements in the technology itself

Therefore, we aim to combat the situation by making use of artificial intelligence to provide even better layers of authentication to protect the user’s data and keep it secure.

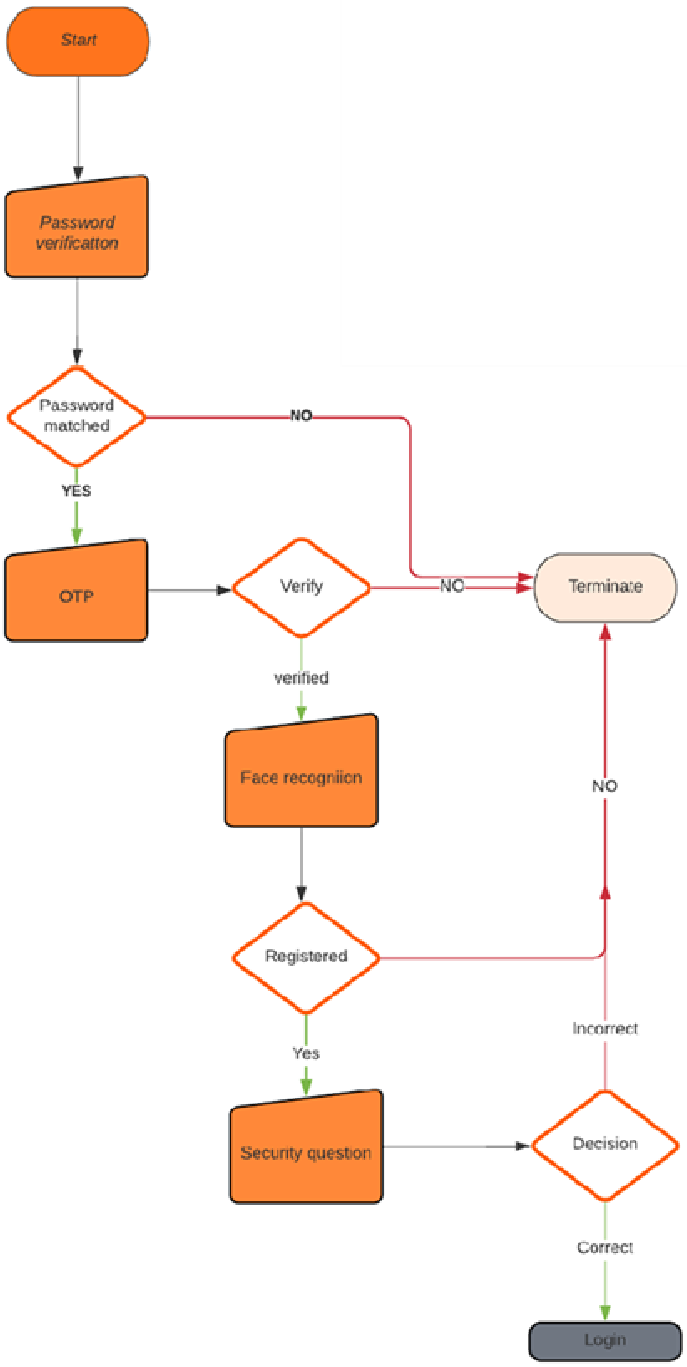
Therefore, a **Four-Layer authentication** security measure by making the use of Artificial Intelligence will be much more efficient, secure, and usable than the commonplace **Two Layer Authentication** existing in the applications as of now.
By making the use of **Artificial Intelligence** we **upgrade** the existing Two-layer Authentication security measure to be a Four-Layer Authentication security measure, the usage of AI helps us add more levels as would be described in the methodology with the specifics to combat this devious problem.

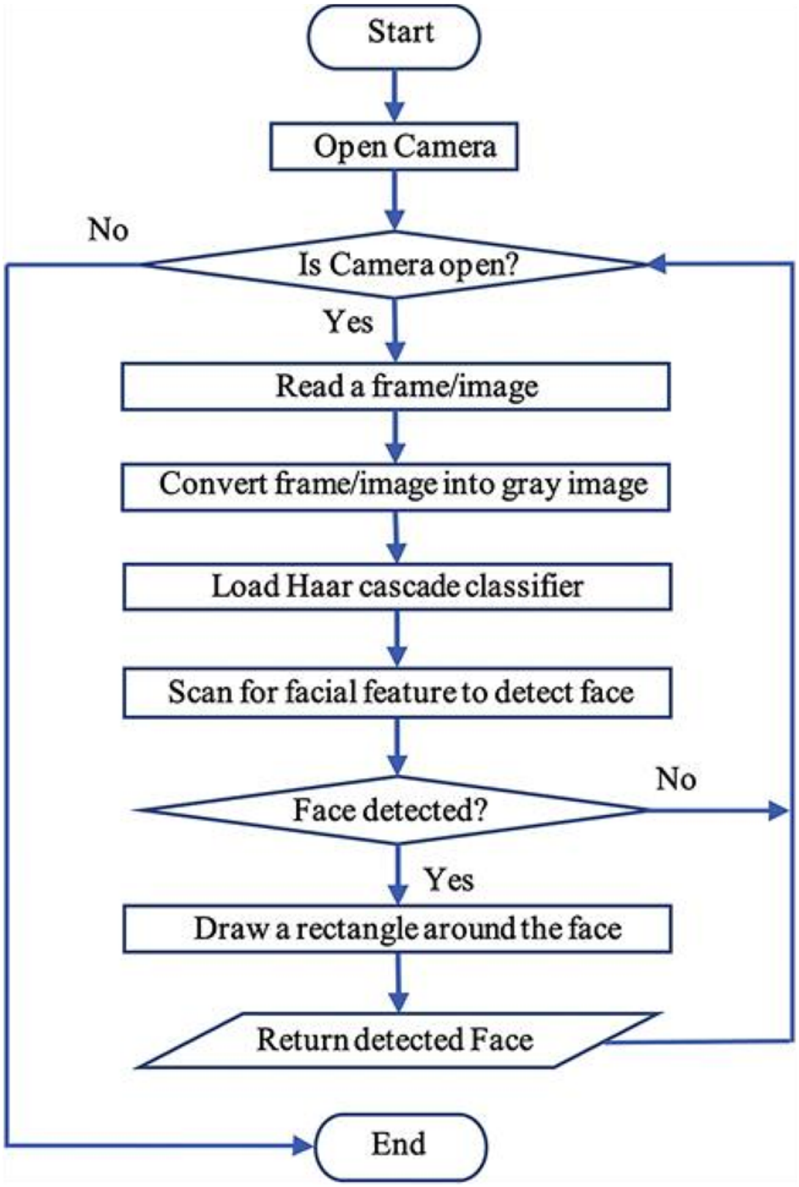
Therefore, in summary, the **problem** that is **defined** is the following:

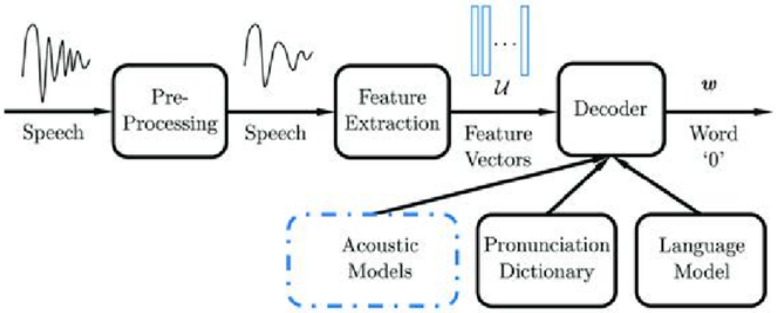
SQL Injections and data breaches require protection that is unable to be provided by existing security layers of authentication, that is the problem which we **counter** by using AI to provide four-layer multimodal security authentication.

By focusing our efforts on this field, we hope to gain more insight into the applications of security on the databases and how security can be continuously improved. By researching various papers dealing with topics related to what we wish to use to enhance our security goal, we hope to become more knowledgeable as we develop a program that doesn’t use common algorithms to display a security system of our own that can rightfully be a four-factor authentication covered security system, providing more accuracy and safety to strike a major blow on cybercrime warfare.

Diagrams :


Start
Password verificatton
Password matched
NO
YES
OTP
Verify
NO
Terminate
verified
Face recogniion
NO
Registered
Yes
Incorrect
Security question
Decision
Correct
Login


Start
Open Camera
No
Is Camera open?
Yes
Read a frame/image
Convert frame/image into gray image
Load Haar cascade classifier
Scan for facial feature to detect face
No
Face detected?
Yes
Draw a rectangle around the face
Return detected Face
End


Speech
Pre-Processing
Speech
Feature Extraction
u
Feature Vectors
Decoder
w
Word '0'
Acoustic Models
Pronunciation Dictionary
Language Model

Pseudocode :

```
begin

    def addDetails to add the details to the database

    create connection(using sqlite3 module) to myData.db file

    create table for the database if it not exists with the following attributes

        fname TEXT

        lname TEXT

        email TEXT

        question TEXT

        answer TEXT

        username TEXT

        password TEXT

    store the details received as input parameter to the function

    close the connection once the database file is updated


    def getDetails to get the details of username

    connect to myData.db(using sqlite3 module) file

    result = SELECT question, answer, email, password FROM Details WHERE
username = username(given as input parameter)

    for ques, ans, email, password in result:

   display(ques, ans, email, password)


    def gen

    while True:

   frame, names = camera.get_frame()

     set global variable got_names

   set temporary variable flag to 0
```

```
        assign present_list to got_names

        for name in names:

            if "Unknown" in name:

                continue

            for i in present_list:

                if name in i:

                    assign flag to 1

                    break

            if flag = 0:

        #          got_names.append({"name": name, "status": "Present"})

                got_names.append(name)

        yield (b'--frame\r\n'

               b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n\r\n')
```

FOR VIDEO FEED

```
            def video_feed

            return Response(gen(VideoCamera()),

                   mimetype='multipart/x-mixed-replace; boundary=frame')
```

FOR EMAIL AUTHENTICATION

```
            def email_authentication():

    name = request.args.get("userName")

    print(name)

    ques, ans, email, password = getDetails(name)

    if email!="":

        return render_template('email_auth.html', name=name, email=email,
password=password)
```

```
    else:

        return render_template('fail.html')
```

FOR FACE AUTHENTICATOIN

```
            def face_authentication():

    name = request.args.get("userName")

    print(name)

    ques, ans, email, password = getDetails(name)

    if email!="":

        return render_template('face_auth.html', name=name, email=email, password=password)

    else:

        return render_template('fail.html')
```

FOR PASSWORD AUTHENTICATION

```
            def pass_authentication():

    name = request.args.get("userName")

    print(name)

    ques, ans, email, password = getDetails(name)

    if email!="":

        return render_template('password_auth.html', name=name, email=email, password=password)

    else:

        return render_template('fail.html')
```

FOR NAMES

```
            def send_names():

            global got_names
```

```
      data = json.dumps(got_names)
            return data

      TO ADD NEW IMG
      def add_new():
  name = request.args.get('name')
      return Response(newEntry(VideoCameraSave(), name),
  mimetype='multipart/x-mixed-replace; boundary=frame')

      FOR LOGIN PAGE
      def login():
      return render_template('login.html')

  @app.route('/welcome', methods=["GET"])
  def welcome():
    name = request.args.get("userName")
    print(name)
    ques, ans, email, password = getDetails(name)
    if ques!="":
      return render_template('security_ques.html', name=name, question=ques,
answer=ans, email=email)
    else:
      return render_template('fail.html')


  TO RESET LOGIN
  def resetLogin():
    global got_names
    got_names = []
    return render_template('login.html')
```

TO LOGOUT

```
def reset_state():
    global got_names
    got_names = []
    return render_template('index.html')
```

TO REGISTER

```
def register():
    return render_template('register.html')


@app.route('/storeData', methods=["GET"])
def store():
    fname = str(request.args.get("fname"))
    lname = str(request.args.get("lname"))
    email = str(request.args.get("email"))
    username = str(request.args.get("username"))
    password = str(request.args.get("password"))
    question = str(request.args.get("question"))
    answer = str(request.args.get("answer"))
    addDetails(fname, lname, email, username, password, question, answer)
    return render_template("snap.html", name=username)
```

FOR REGISTRATION

```
def reg_success():

    return render_template('reg_success.html')
```

LOGGED IN

```
def final_success():

    return render_template('login_success.html')
```

Explanation :

We have used different modules like haar cascade, siamese neural networks, multispectral imaging for face detection, and many other modules like a pillow, NumPy, sqlite3, flask, etc. to integrate HTML, CSS and query language all into python and achieve multiple-step verification. The user is prompted for a password as soon as he/she tries to log in. If the password is correct then the user is prompted for OTP. If the OTP is verified then he/she is prompted for Face recognition based on the data associated to the user if the face matches then he/she is prompted for the last verification step i.e. security question. If all of them are verified then the user is allowed to login else they are returned to the home page.

Layer 1: Password Verification

When a user first signs up for our website, they're asked to choose a username and password to identify themselves. Providing a password adds a simple layer of security over the other authentication methods.

In an ideal world, the user would always pick a strong and unique password so that it's harder for an attacker to guess.

Layer 2: Email-OTP Verification

The Email OTP method enables you to authenticate using the one-time password (OTP) that is sent to the registered email address. When you try to authenticate on any device, the server sends an email to the registered email address with the OTP. We can use this OTP for single authentication within a short time frame.

Layer 3: Face recognition

Face recognition is a technology capable of identifying or verifying a subject through an image, video, or any audiovisual element of his face. Generally, this identification is used to access an application, system, or service.

It is a method of biometric identification that uses that body measures, in this case, face and head, to verify the identity of a person through its facial biometric pattern and data. The technology collects a set of unique biometric data of each person associated with their face and facial expression to identify, verify and/or authenticate a person.

Layer 4: Security Question (with speech-to-text answering feature)

Security questions are a common method of identity authentication. When creating an account or signing up for a service online, users will confidentially share the answers to secret questions with a provider. Typically, these security questions and answers are used for self-service password recovery, inputting the correct answer verifies the user and allows them to reset their password—though we can also implement security questions as an additional authentication factor for logins.

We also added a special feature of speech-to-text to this module which allows the users to answer their security questions using their microphones. In this, we have used the web-speech API to achieve this feature.

5. Experiment and Results:

Complexity Analysis

There are three ways you could measure accuracy in a face recognition task. The one that was most appropriate would depend to an extent on what the end goal was.

i. How accurate is the algorithm at detecting one person from a data set containing many images of one person and many images of different people

ii. How accurate is the algorithm at learning a set of faces from training images and then correctly identifying the same people from a test set of different images, where both image sets contain the same people

iii. How accurate is the algorithm at detecting multiple people from a dataset containing images of these people and other people

For the speech recognition task, the most common measure is the so-called word error rate (WER). Such a performance is computed by comparing a reference transcription with the transcription output by the speech recognizer. From this comparison, it is possible to compute the number of errors, which typically belong to 3 categories:

i. Insertions I (when in the output of the ASR it is present a word not present in the reference)

ii. Deletions D(a word is missed in the ASR output)

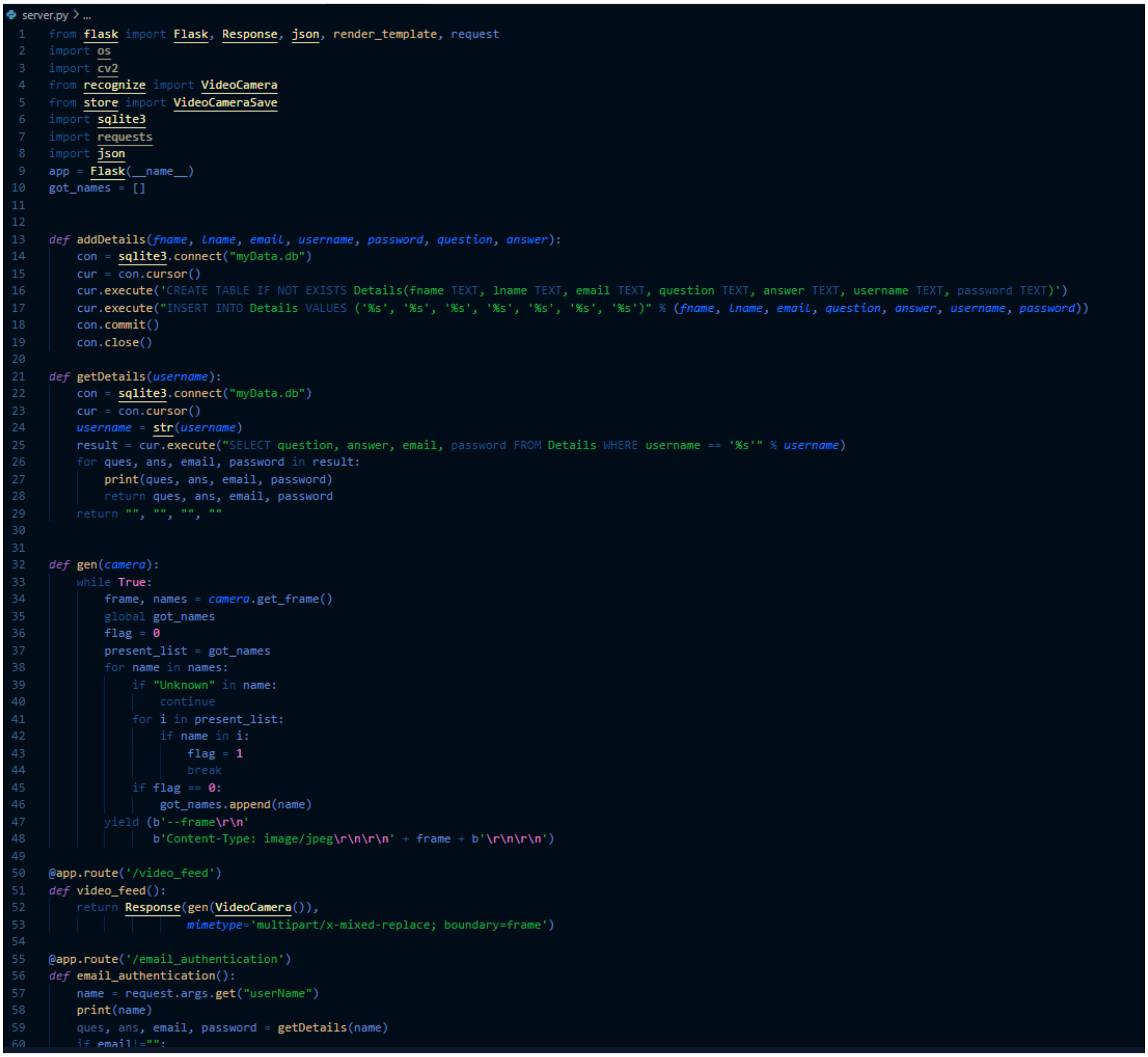
iii. Substitutions S (a word is confused with another one)

WER= (S+D+I)/N - where N is the number of words

Methodology of getting the Output

1. Initially we launch the server application that is running on a flask application. The server application connects all of the files together such as the html, css, python, flask, database, and the internal system.
2. After initializing the server.py file the index page is launched, and the user is prompted to register themselves to log in to the system.

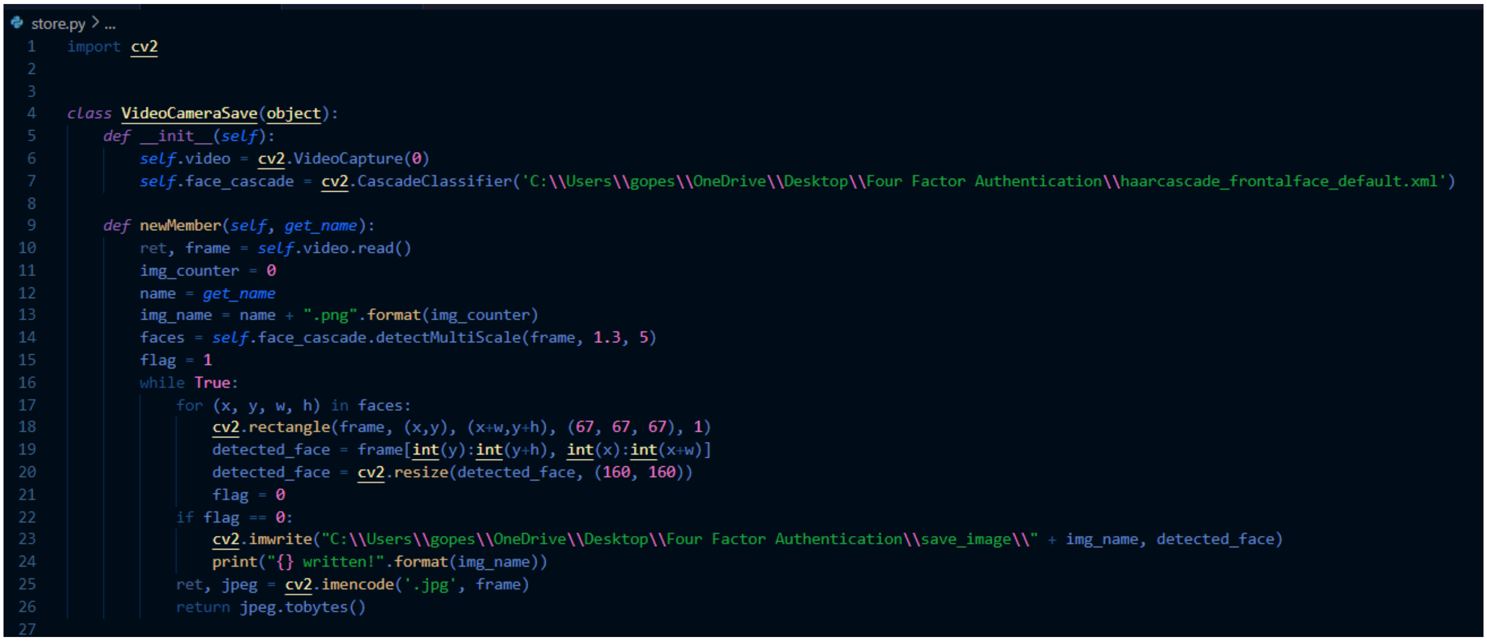
server.py



```python
from flask import Flask, Response, json, render_template, request
import os
import cv2
from recognize import VideoCamera
from store import VideoCameraSave
import sqlite3
import requests
import json
app = Flask(__name__)
got_names = []


def addDetails(fname, lname, email, username, password, question, answer):
    con = sqlite3.connect("myData.db")
    cur = con.cursor()
    cur.execute('CREATE TABLE IF NOT EXISTS Details(fname TEXT, lname TEXT, email TEXT, question TEXT, answer TEXT, username TEXT, password TEXT)')
    cur.execute("INSERT INTO Details VALUES ('%s', '%s', '%s', '%s', '%s', '%s', '%s')" % (fname, lname, email, question, answer, username, password))
    con.commit()
    con.close()

def getDetails(username):
    con = sqlite3.connect("myData.db")
    cur = con.cursor()
    username = str(username)
    result = cur.execute("SELECT question, answer, email, password FROM Details WHERE username == '%s'" % username)
    for ques, ans, email, password in result:
        print(ques, ans, email, password)
        return ques, ans, email, password
    return "", "", "", ""


def gen(camera):
    while True:
        frame, names = camera.get_frame()
        global got_names
        flag = 0
        present_list = got_names
        for name in names:
            if "Unknown" in name:
                continue
            for i in present_list:
                if name in i:
                    flag = 1
                    break
            if flag == 0:
                got_names.append(name)
        yield (b'--frame\r\n'
               b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n\r\n')

@app.route('/video_feed')
def video_feed():
    return Response(gen(VideoCamera()),
                    mimetype='multipart/x-mixed-replace; boundary=frame')

@app.route('/email_authentication')
def email_authentication():
    name = request.args.get("userName")
    print(name)
    ques, ans, email, password = getDetails(name)
    if email!="":
```

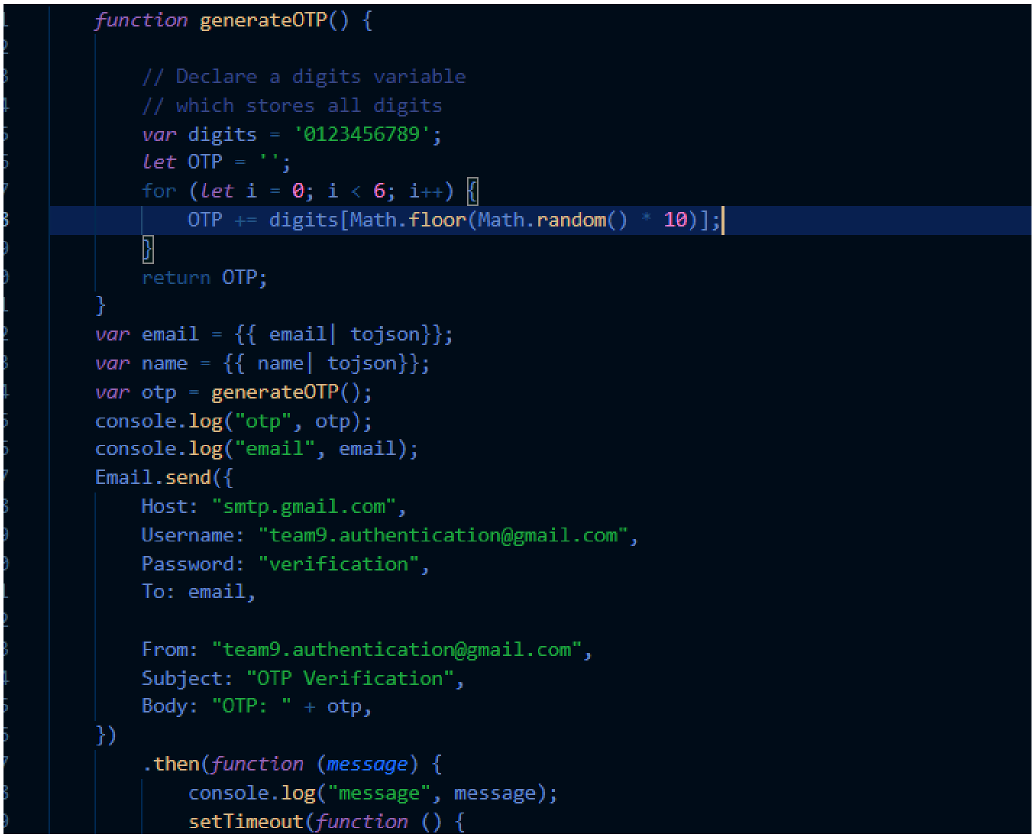
3. The code stores all the details input by the user into the myData.db database file. And this data will be shown and used later while authenticating the user.
4. Once, the user inputs the details and is prompted to the next step - face authentication. Here we have written the code (store.py) which stores creates a grid around the user face and stores it in the folder called save_image. The code analyses the image and runs it through the haarcascade xml fil which is the classifier and it identifies the image and will use it for further authentication along the way.

store.py



```python
import cv2


class VideoCameraSave(object):
    def __init__(self):
        self.video = cv2.VideoCapture(0)
        self.face_cascade = cv2.CascadeClassifier('C:\\Users\\gopes\\OneDrive\\Desktop\\Four Factor Authentication\\haarcascade_frontalface_default.xml')

    def newMember(self, get_name):
        ret, frame = self.video.read()
        img_counter = 0
        name = get_name
        img_name = name + ".png".format(img_counter)
        faces = self.face_cascade.detectMultiScale(frame, 1.3, 5)
        flag = 1
        while True:
            for (x, y, w, h) in faces:
                cv2.rectangle(frame, (x,y), (x+w,y+h), (67, 67, 67), 1)
                detected_face = frame[int(y):int(y+h), int(x):int(x+w)]
                detected_face = cv2.resize(detected_face, (160, 160))
                flag = 0
            if flag == 0:
                cv2.imwrite("C:\\Users\\gopes\\OneDrive\\Desktop\\Four Factor Authentication\\save_image\\" + img_name, detected_face)
                print("{} written!".format(img_name))
            ret, jpeg = cv2.imencode('.jpg', frame)
            return jpeg.tobytes()

```

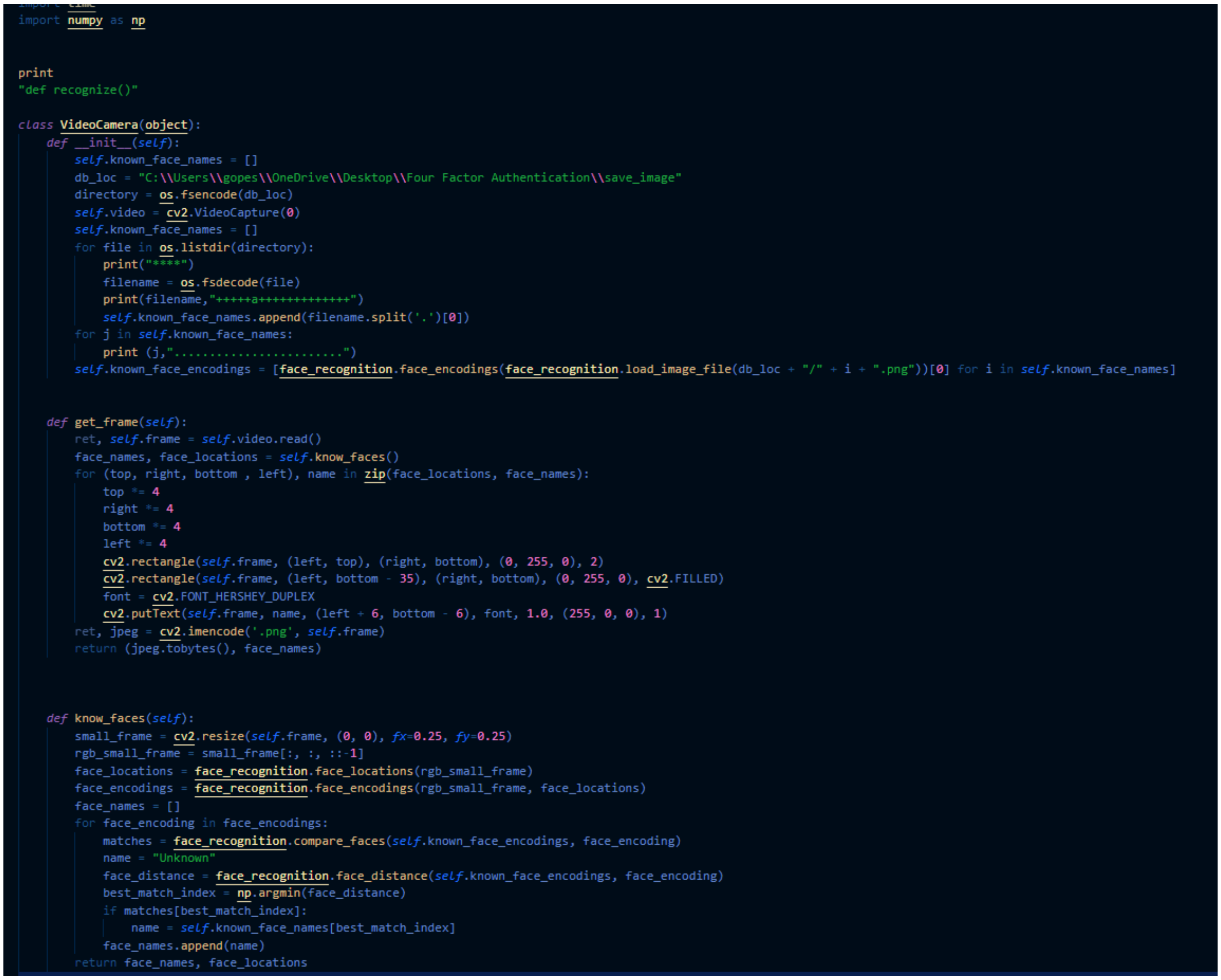
5. Once the image is snapped and stored then the user goes to log in via their credentials. Now the user will be intimidated about an OTP sent to their respective email that they had added earlier into the details section. This OTP has to be written in the space given.



```
function generateOTP() {

    // Declare a digits variable
    // which stores all digits
    var digits = '0123456789';
    let OTP = '';
    for (let i = 0; i < 6; i++) {
        OTP += digits[Math.floor(Math.random() * 10)];
    }
    return OTP;
}
var email = {{ email| tojson}};
var name = {{ name| tojson}};
var otp = generateOTP();
console.log("otp", otp);
console.log("email", email);
Email.send({
    Host: "smtp.gmail.com",
    Username: "team9.authentication@gmail.com",
    Password: "verification",
    To: email,

    From: "team9.authentication@gmail.com",
    Subject: "OTP Verification",
    Body: "OTP: " + otp,
})
    .then(function (message) {
        console.log("message", message);
        setTimeout(function () {
```

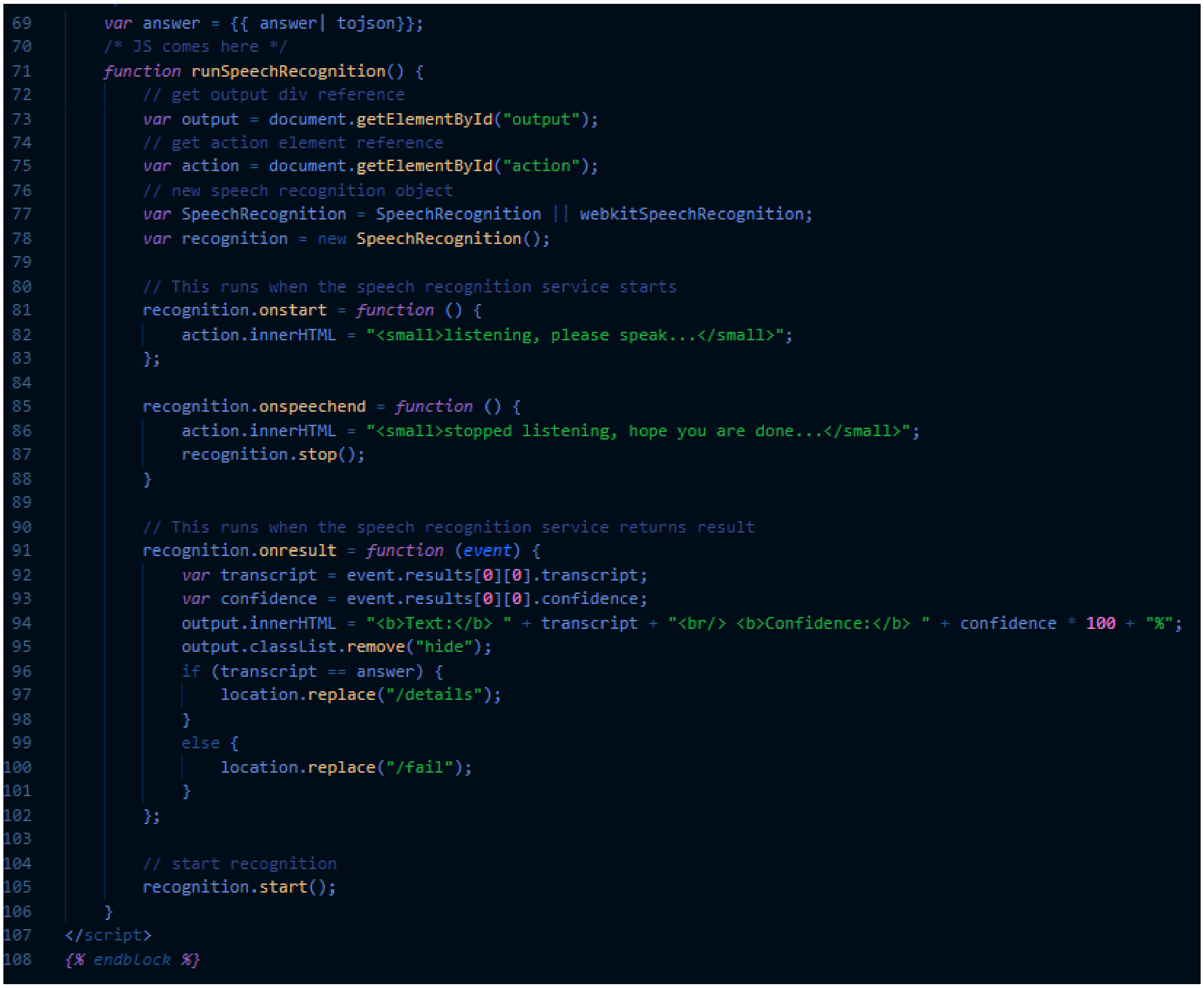
6. After that, the user is prompted to look straight into the camera so that the code to recognize the image that the user will be showcasing matches that of the image that is stored within the database.
7. It matches the face whether or not the characteristics within the grid match those of the one’s taken during the registration. If they do then the image would show a green box indicating the user’s name below else it would display “unknown”.

recognize.py



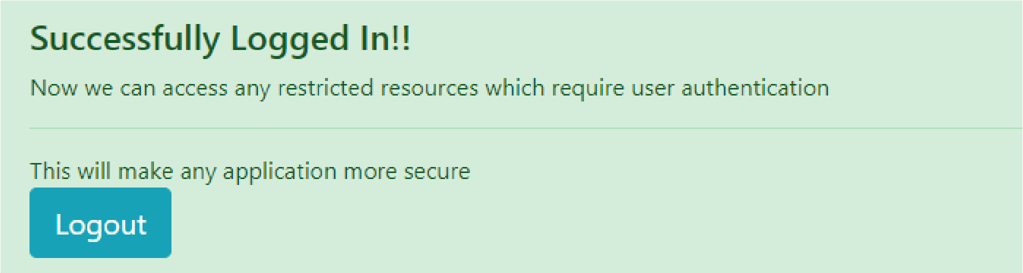
```
import time
import numpy as np


print
"def recognize()"

class VideoCamera(object):
    def __init__(self):
        self.known_face_names = []
        db_loc = "C:\\Users\\gopes\\OneDrive\\Desktop\\Four Factor Authentication\\save_image"
        directory = os.fsencode(db_loc)
        self.video = cv2.VideoCapture(0)
        self.known_face_names = []
        for file in os.listdir(directory):
            print("****")
            filename = os.fsdecode(file)
            print(filename,"+++++a+++++++++++++")
            self.known_face_names.append(filename.split('.')[0])
        for j in self.known_face_names:
            print (j,"........................")
        self.known_face_encodings = [face_recognition.face_encodings(face_recognition.load_image_file(db_loc + "/" + i + ".png"))[0] for i in self.known_face_names]


    def get_frame(self):
        ret, self.frame = self.video.read()
        face_names, face_locations = self.know_faces()
        for (top, right, bottom , left), name in zip(face_locations, face_names):
            top *= 4
            right *= 4
            bottom *= 4
            left *= 4
            cv2.rectangle(self.frame, (left, top), (right, bottom), (0, 255, 0), 2)
            cv2.rectangle(self.frame, (left, bottom - 35), (right, bottom), (0, 255, 0), cv2.FILLED)
            font = cv2.FONT_HERSHEY_DUPLEX
            cv2.putText(self.frame, name, (left + 6, bottom - 6), font, 1.0, (255, 0, 0), 1)
        ret, jpeg = cv2.imencode('.png', self.frame)
        return (jpeg.tobytes(), face_names)



    def know_faces(self):
        small_frame = cv2.resize(self.frame, (0, 0), fx=0.25, fy=0.25)
        rgb_small_frame = small_frame[:, :, ::-1]
        face_locations = face_recognition.face_locations(rgb_small_frame)
        face_encodings = face_recognition.face_encodings(rgb_small_frame, face_locations)
        face_names = []
        for face_encoding in face_encodings:
            matches = face_recognition.compare_faces(self.known_face_encodings, face_encoding)
            name = "Unknown"
            face_distance = face_recognition.face_distance(self.known_face_encodings, face_encoding)
            best_match_index = np.argmin(face_distance)
            if matches[best_match_index]:
                name = self.known_face_names[best_match_index]
            face_names.append(name)
        return face_names, face_locations
```

8. Now, that the user has input all the details and his face matches the given one, they will now be able to access the voice recognition feature. Here once the words spoken match the input words only then will the system log them in.



```
    var answer = {{ answer| tojson}};
    /* JS comes here */
    function runSpeechRecognition() {
        // get output div reference
        var output = document.getElementById("output");
        // get action element reference
        var action = document.getElementById("action");
        // new speech recognition object
        var SpeechRecognition = SpeechRecognition || webkitSpeechRecognition;
        var recognition = new SpeechRecognition();

        // This runs when the speech recognition service starts
        recognition.onstart = function () {
            action.innerHTML = "<small>listening, please speak...</small>";
        };

        recognition.onspeechend = function () {
            action.innerHTML = "<small>stopped listening, hope you are done...</small>";
            recognition.stop();
        }

        // This runs when the speech recognition service returns result
        recognition.onresult = function (event) {
            var transcript = event.results[0][0].transcript;
            var confidence = event.results[0][0].confidence;
            output.innerHTML = "<b>Text:</b> " + transcript + "<br/> <b>Confidence:</b> " + confidence * 100 + "%";
            output.classList.remove("hide");
            if (transcript == answer) {
                location.replace("/details");
            }
            else {
                location.replace("/fail");
            }
        };

        // start recognition
        recognition.start();
    }
</script>
{% endblock %}
```

9. Finally, once the user has completed all the steps successfully the system then lets the user go ahead into the system and the verification completes. The server.py then offers either to log out or continue with the system.



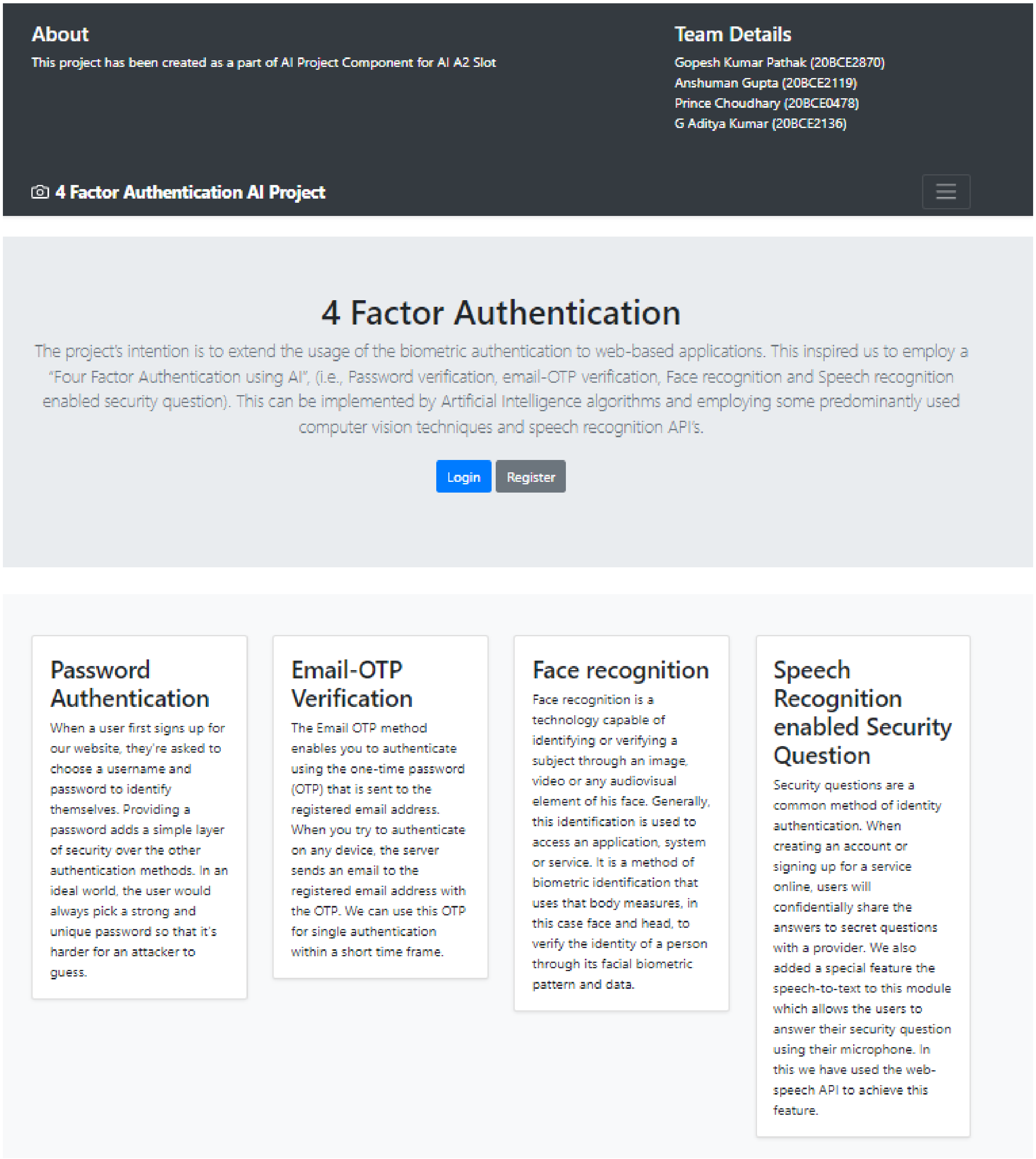
Successfully Logged In!!

Now we can access any restricted resources which require user authentication

This will make any application more secure

OUTPUT SCREENS

Index page



About

This project has been created as a part of AI Project Component for AI A2 Slot

Team Details

Gopesh Kumar Pathak (20BCE2870)
Anshuman Gupta (20BCE2119)
Prince Choudhary (20BCE0478)
G Aditya Kumar (20BCE2136)

4 Factor Authentication AI Project

4 Factor Authentication

The project's intention is to extend the usage of the biometric authentication to web-based applications. This inspired us to employ a "Four Factor Authentication using AI", (i.e., Password verification, email-OTP verification, Face recognition and Speech recognition enabled security question). This can be implemented by Artificial Intelligence algorithms and employing some predominantly used computer vision techniques and speech recognition API's.

Password Authentication

When a user first signs up for our website, they're asked to choose a username and password to identify themselves. Providing a password adds a simple layer of security over the other authentication methods. In an ideal world, the user would always pick a strong and unique password so that it's harder for an attacker to guess.

Email-OTP Verification

The Email OTP method enables you to authenticate using the one-time password (OTP) that is sent to the registered email address. When you try to authenticate on any device, the server sends an email to the registered email address with the OTP. We can use this OTP for single authentication within a short time frame.

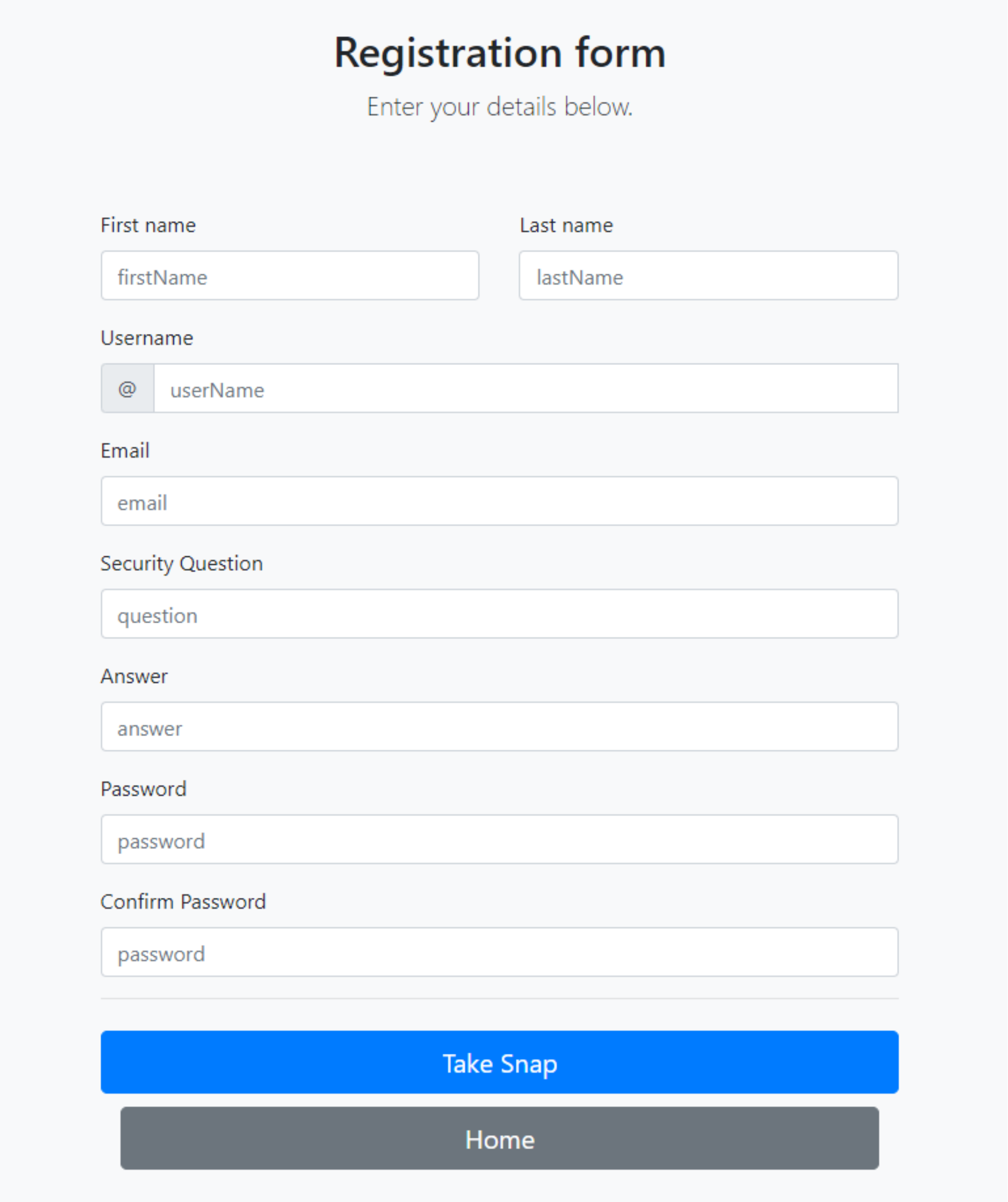
Face recognition

Face recognition is a technology capable of identifying or verifying a subject through an image, video or any audiovisual element of his face. Generally, this identification is used to access an application, system or service. It is a method of biometric identification that uses that body measures, in this case face and head, to verify the identity of a person through its facial biometric pattern and data.

Speech Recognition enabled Security Question

Security questions are a common method of identity authentication. When creating an account or signing up for a service online, users will confidentially share the answers to secret questions with a provider. We also added a special feature the speech-to-text to this module which allows the users to answer their security question using their microphone. In this we have used the web-speech API to achieve this feature.

Registration



Registration form

Enter your details below.

First name: firstName

Last name: lastName

Username: @ userName

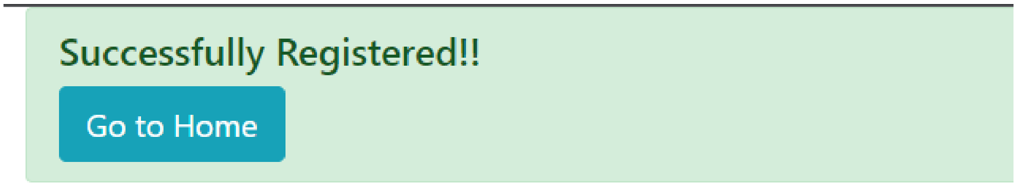
Email: email

Security Question: question

Answer: answer

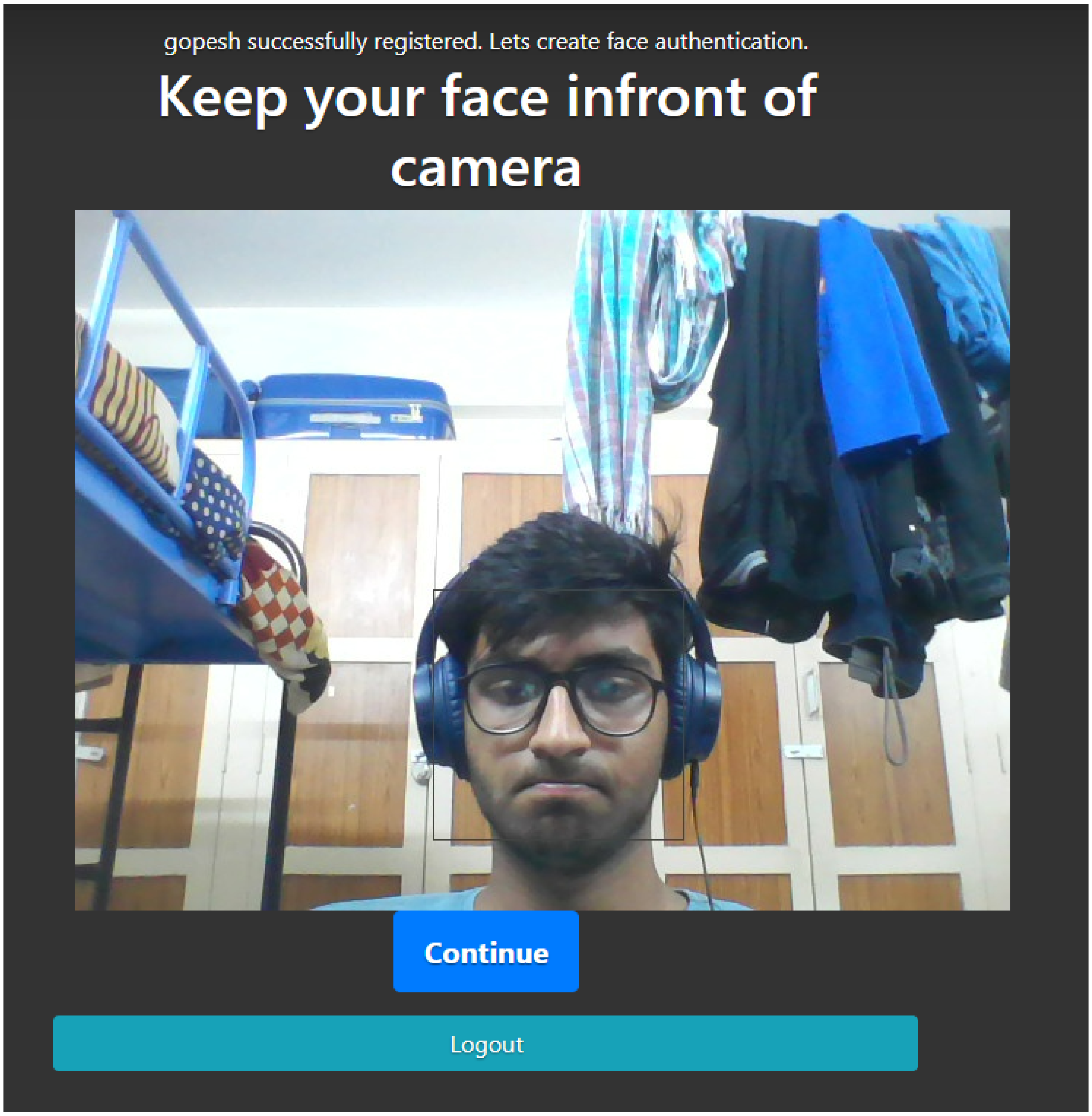
Password: password

Confirm Password: password



Successfully Registered!!

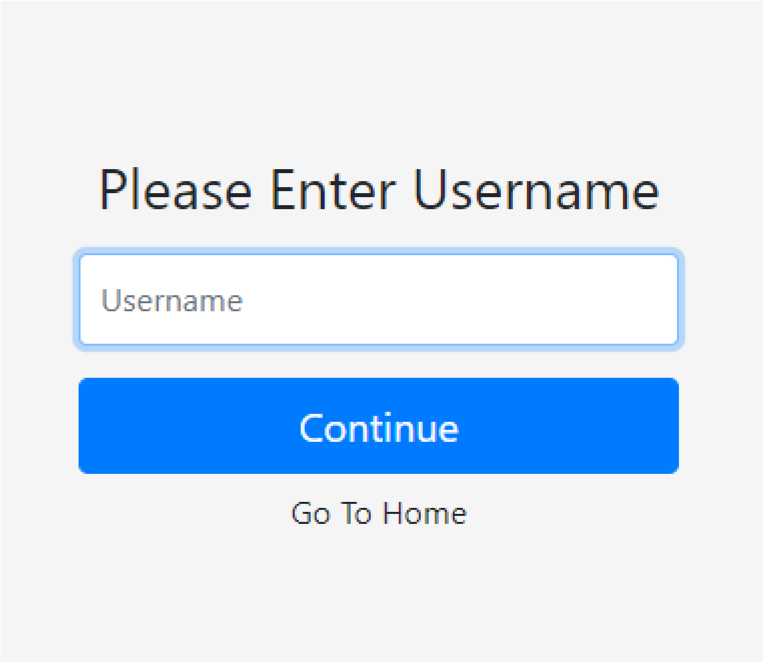
Face Snap



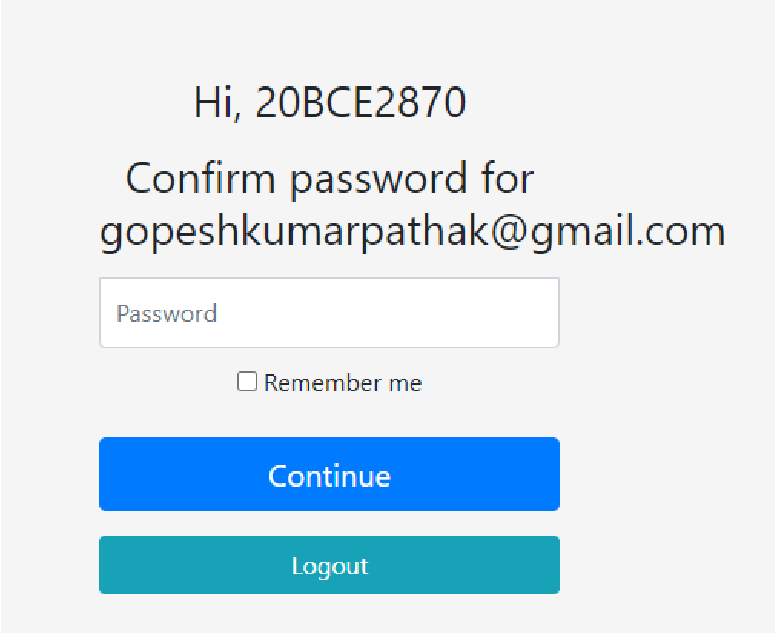
gopesh successfully registered. Lets create face authentication.

Keep your face infront of camera

Login



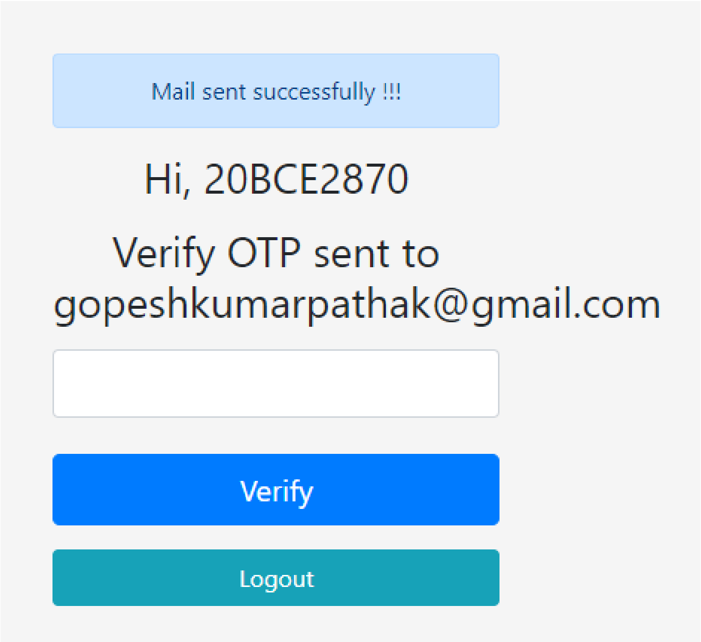
Please Enter Username



Hi, 20BCE2870

Confirm password for gopeshkumarpathak@gmail.com

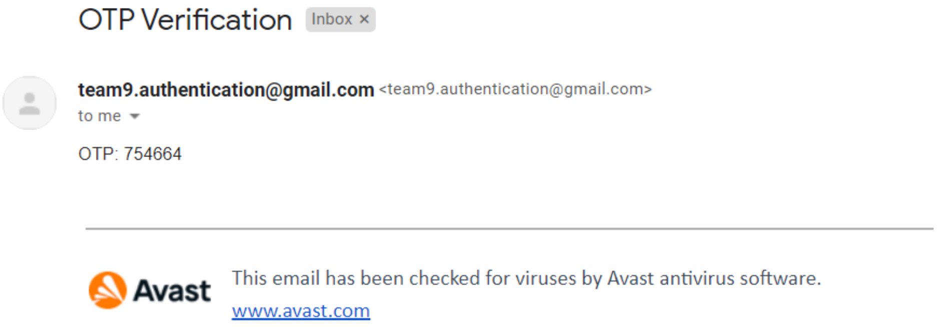
OTP Confirmation



Mail sent successfully !!!

Hi, 20BCE2870

Verify OTP sent to gopeshkumarpathak@gmail.com



OTP Verification

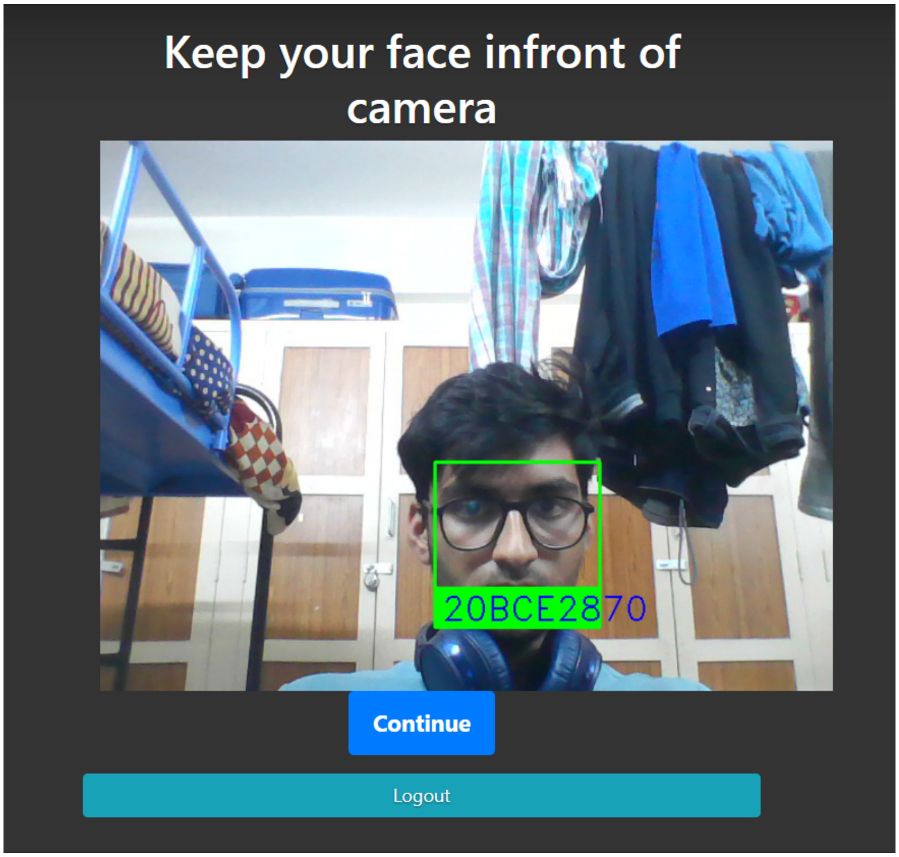
team9.authentication@gmail.com <team9.authentication@gmail.com>

to me

OTP: 754664

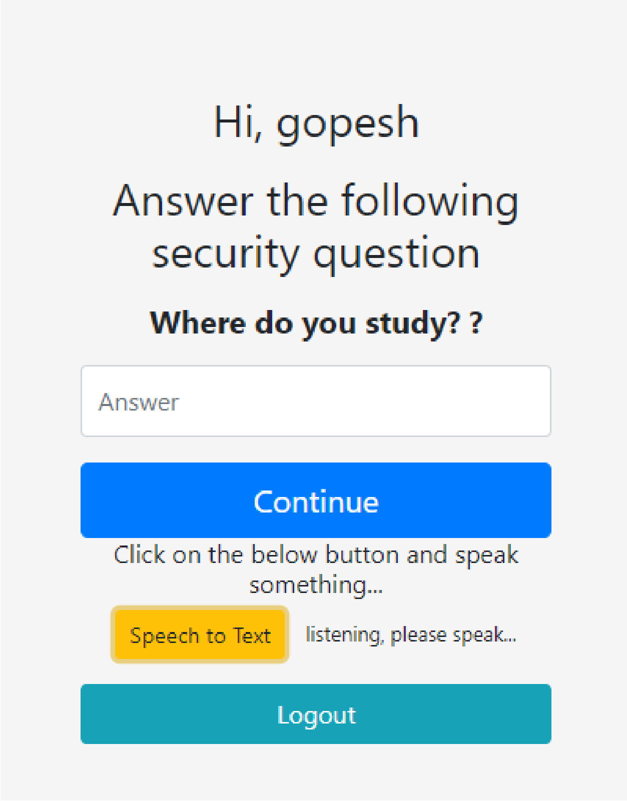
This email has been checked for viruses by Avast antivirus software.
www.avast.com

Recognize Face



Keep your face infront of camera

Speech Recognition



Hi, gopesh

Answer the following security question

Where do you study? ?

Answer

Click on the below button and speak something...

listening, please speak...

6. Conclusion:

Hence, the proposed four-factor authentication model provides much better security than the commonplace two-factor authentication model. The model so far easily takes into account the face and speech inputs it needs to authenticate and allow the users in, working in conjunction with the first two layers of authentication as well.

Further accuracy can be improved, especially with the usage of more 3D techniques to prepare a better model of the face provided so as to combat the instances where it may be obscured by lighting, glasses or hair. Similarly, when it comes to the speech recognizer with the major ailments being the sound of muffled voice while eating, needing more clarity than a normal user cares to give while accessing the device

Further layers of authentication can also be implemented with the concept of the fingerprint scanner, something more readily available in all smartphones and laptops today but not as accessible as webcams and microphones, at least at the present time. Working in conjunction with the above, a fingerprint scanner may provide the **fifth layer** if needed but till the usage of fingerprint scanners in devices becomes more common in the next few years, isn't urgent.

7. References

1. https://ieeexplore.ieee.org/document/9562651
2. https://www.semanticscholar.org/paper/A-Review-of-Face-Recognition-Technology-Li-Mu/2216885b057dace68ce7307d77bfe4b5ceefa9ad
3. https://journalofcloudcomputing.springeropen.com/track/pdf/10.1186/s13677-020-00186-7.pdf
4. https://ieeexplore.ieee.org/document/45616
5. https://cyberleninka.org/article/n/308728.pdf
6. https://www.mdpi.com/1424-8220/20/2/342
7. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0242269
8. https://link.springer.com/article/10.1007/s11042-020-10139-6
9. https://www.sciencedirect.com/science/article/abs/pii/S0925231220316945
10. http://alweb.ehu.es/ccwintco/uploads/d/d2/PFC-IonMarqu%C3%A9s.pdf
11. https://sci-hub.hkvisa.net/10.1109/icsai.2012.6223418
12. https://arxiv.org/abs/1804.06655
13. https://gravis.dmi.unibas.ch/publications/pami03.pdf
14. https://ijcrt.org/download.php?file=IJCRT1892450.pdf
15. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1077.8640&rep=rep1&type=pdf