Faculty attendance system using Geofence, Face-recognition and admin portal

Anandteertha Rao [a], Atharva Vaidya [b], Yagnesh Narayanan [c]

[a, b, c] Undergraduate student of University of Mumbai, South Indian Education Society Graduate School of Technology, Nerul in Computer Engineering.

Abstract: Attendance is the most common element throughout our life, starting from schooling days, right to the corporate world. Even though throughout the past few decades a lot of modifications and improvements have been made in the attendance monitoring system, there has always been a few setbacks. Our solution, provides an efficient and a very way to provide attendance in any organization, without using the biometric which tends to be a huge investment and during the pandemic situation, such as the one we are facing now, we can never be too careful. We use face recognition, geofencing and an admin portal to provide three most effective and easy ways for any user to mark their attendance

Keywords: Face Recognition, Geofencing, Authentication, Database, Microservices.

1. Introduction

During the pandemic we faced a lot of difficulties, one of them was "contactless services". As though this problem was dealt with, in an extremely careful way, we are still lagging behind in some aspects, such as faculty attendance or attendance in general corporate environment, where people still need to report. Many companies or organizations who previously used biometric devices for their attendance system now had to stop and come up with alternative ways for attendance. People started going back to the traditional ways, using a register and a pen, and marking their attendance as they walk in the organization. This took up a lot of their valuable time, standing in queue.

1.1. Proposed System

There are three main features using which any faculty or user can mark their attendance. First one is using a facial recognition model, the Ip-camera will be placed in the organization and at most five people can stand and their attendance will be marked provided they have made an account in the portal and uploaded three to four images with their faces in it. The second one is using a geofence which will allow the user to simply click one button and if they are within the organization premises their attendance will marked. Finally, the last one which is an admin portal, if a user due to some reason is not able to mark their attendance using geofence, due to browser issues or face recognition, can send a request to the admin to mark their attendance.

1.2. Objectives

The main objective of this project is to ensure attendance was marked for the user who is present. Encourage more organizations to use non biometric approach for attendance. Provide analysis of each and every user/ faculty to the admin, and include many more features such as entry time and exit time, leave applications, etc.

2. Literature Review

In the recent years, there has been a tremendous growth in Artificial Intelligence (AI) and Image Processing, this growth has led to many outstanding implementations of face detection and recognition algorithms using hog (histogram of oriented gradients) or CNN (convolutional neural networks). Many implementations of the same algorithms are also available as open-source. Over the past few years attendance system using biometrics [1] such as finger print or retinal scan, but this setup required a lot of investment. Using such a system where we have to do a physical contact with the hardware is dangerous considering the pandemic situation such as covid19.

Implementation of smart attendance system using RFID, GPS, etc., has been done in the past years but each application was built for student attendance primarily. The attendance system using GPS for students could not ensure if the student who are physically present inside the college premises are actually attending the lecture or not. The issue with RFID is that the teacher has to keep some amount of time of their lecture for marking the attendance of each student. This is similar to the traditional method of using a register and calling out each student's name. Although a blend of GPS and RFID would give us better results in case of faculty attendance, we would still have a queue in the organization for marking attendance.

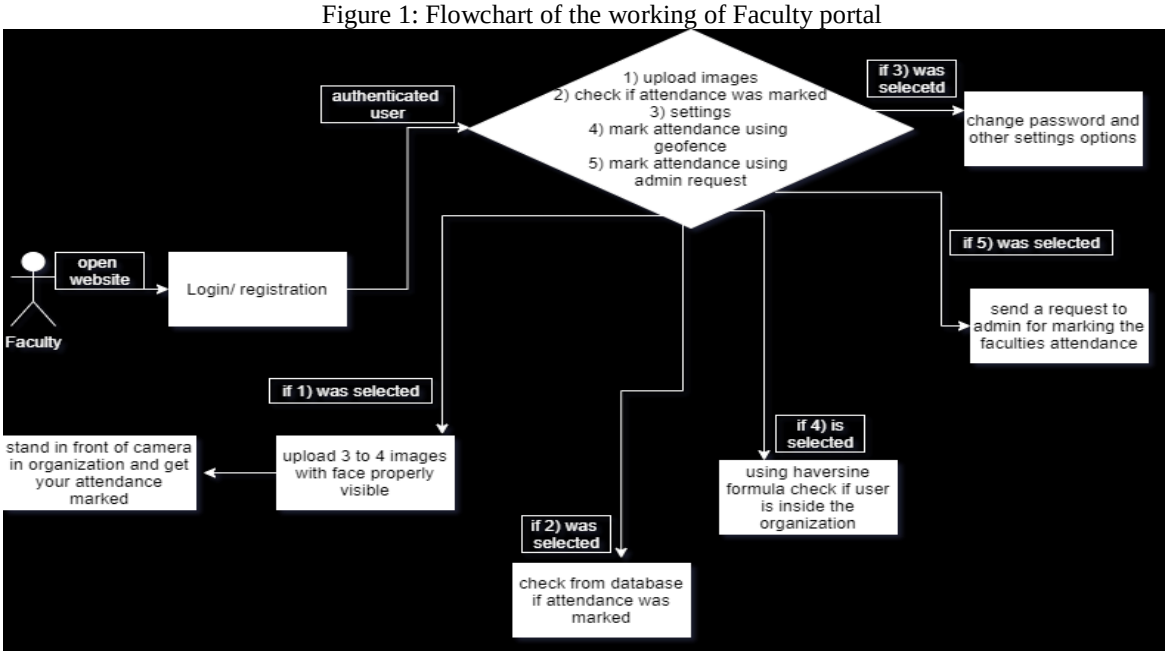
After studying these research papers and looking at the market, we can say that a perfect mix of GPS or geofencing, face recognition and human intervention in marking attendance would result in a nice solution to our problem.

3.System Working

Our system comprises of two main components, one is the faculty portal and the second is the admin portal.

Faculty Portal:

Figure 1: Flowchart of the working of Faculty portal



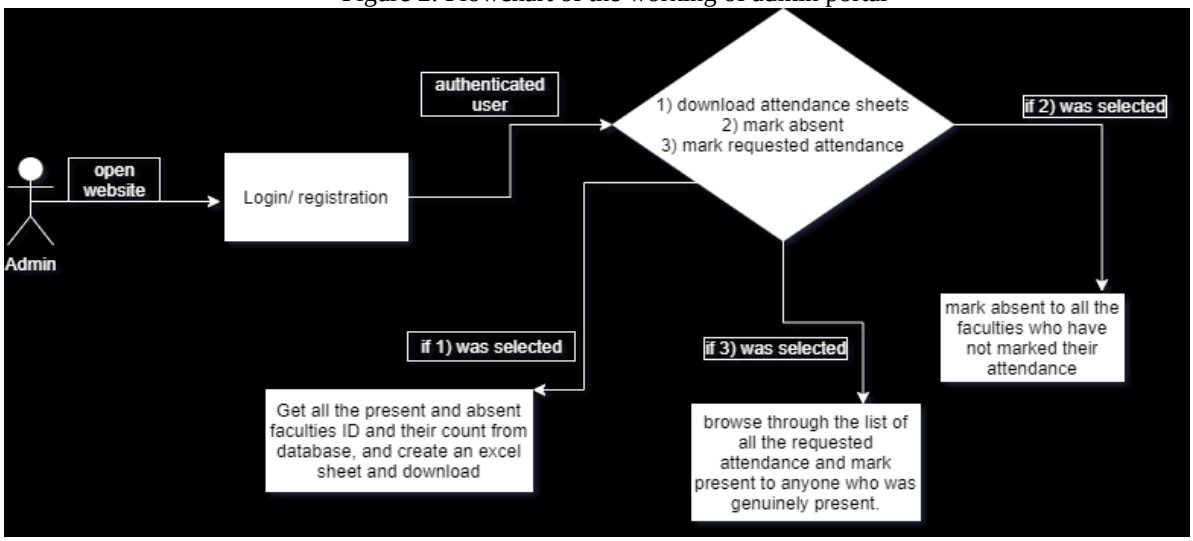
The faculty needs to register in our portal and he/she will get an OTP (one time password) and a link which will be used to verify the email address of the faculty. Once it is done, and the faculty is logged in, they can upload images for facial recognition, check if their attendance was marked for the day, check their total attendance count, change their password from settings, etc.

Suppose, there is a queue for face recognition attendance, they can simply click on one button and their attendance will be marked, provided they are within the college premises, this is the geofencing feature. We are using the haversine formula for calculating distance between any two points on a sphere. Since we can assume that earth's shape is near to sphere.

There may be times when due to slow network connectivity or browser issues, the current latitude and longitude of the user is not correct, at that time they can request the admin to mark their attendance, with just one click. The admin will then see to it that they are indeed present and will mark them present.

Admin portal:

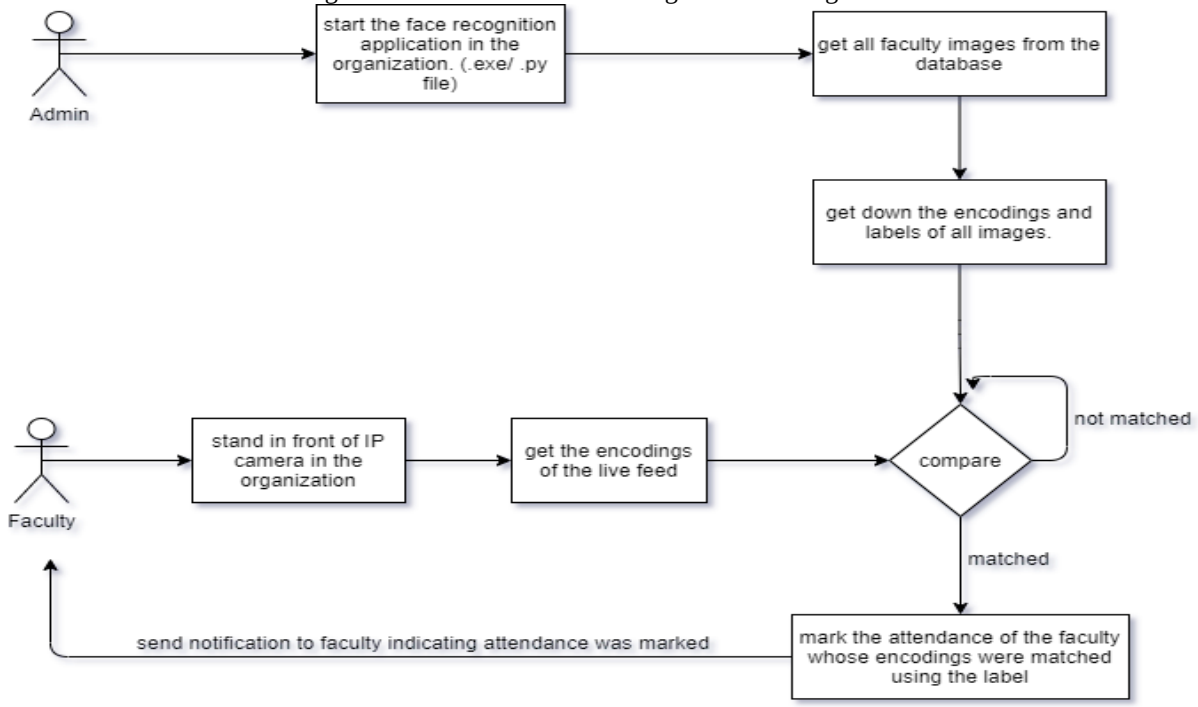
Figure 2: Flowchart of the working of admin portal



The admin needs to login, and he/she can easily download attendance sheet. They can check if the person was present a particular day or not using the data. They can most importantly mark the attendance for faculties by one single click. Before they do that, they to confirm manually if the faculty is indeed present. Here this manual check can be eliminated by using an alternative such as a QR code or some other form of digital print [to be implemented]. At the end of the day, admin can mark absent to anyone who did not mark their attendance.

4. Face Recognition

Figure 3: Flowchart of the working of Face Recognition



For the implementation of face recognition, we have used the Face-recognition [2] package of python. As soon as the admin starts the python script, which will be converted to .exe file in the next version [to be implemented], the images start to flow in from the main server to the local machine. Here a one-step login portal will ensure that the admin is the one who is starting the face recognition model.

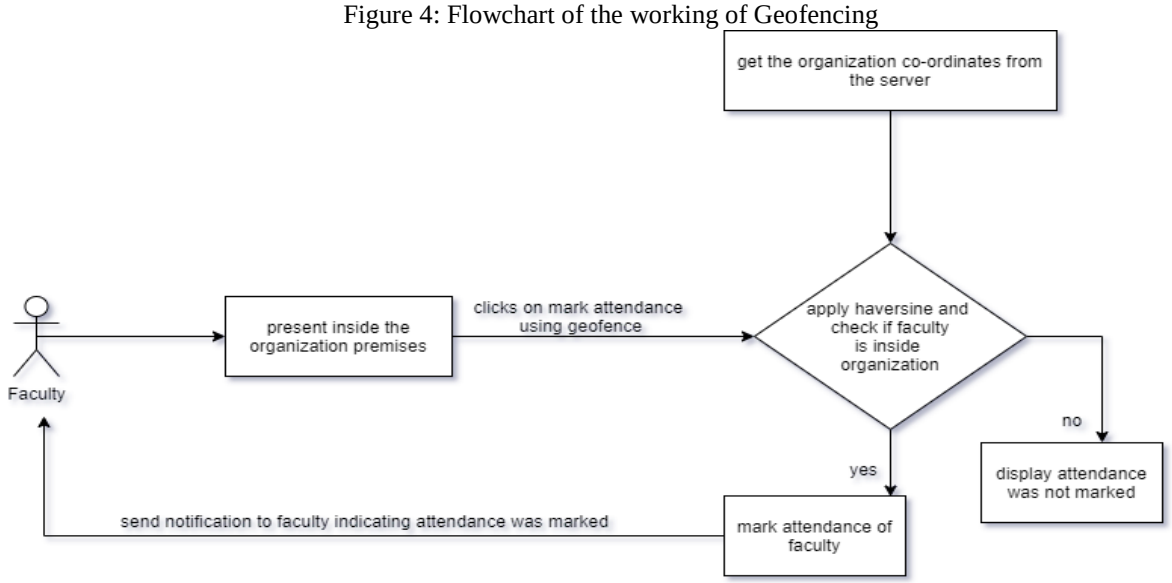
If the images have a face, they will be converted into encodings and it will be stored locally along with the names as the faculty ID. Once this task is done, the IP camera is started which will be present in the same network as the local machine in which the admin is working. The feed will start to come in, and each frame will be analyzed.

The frames that were analyzed will generate their own encodings. These encodings will be used to compare with the locally stored encodings and if we get a match, then that name (faculty ID) will be displayed in the

console to the admin, parallelly a post request will be sent to the main server from the admin machine with the details of the faculty whose encodings were matched.
Once the request reaches the server the attendance will be marked and updated in the database, and a notification will be sent to that faculty in real time in the portal along with necessary emails.

5. Geofencing

Figure 4: Flowchart of the working of Geofencing



Once the faculty, clicks on the mark attendance button using geofence, we get the live co-ordinates of the faculty. We sent these live co-ordinates to the main server.

5.1. Haversine Formula

Figure 5: Haversine Formula [3]

$$\operatorname{hav}(\theta) = \operatorname{hav}(\varphi_2 - \varphi_1) + \cos(\varphi_1)\cos(\varphi_2)\operatorname{hav}(\lambda_2 - \lambda_1)$$

where

- φ_1, φ_2 are the latitude of point 1 and latitude of point 2 (in radians),
- λ_1, λ_2 are the longitude of point 1 and longitude of point 2 (in radians).

We apply the haversine formula to get the distance between two points on a sphere.

Figure 6: Haversine Formula to get distance d [3]

$$d = 2r\arcsin\left(\sqrt{\operatorname{hav}(\varphi_2 - \varphi_1) + \cos(\varphi_1)\cos(\varphi_2)\operatorname{hav}(\lambda_2 - \lambda_1)}\right)$$
$$= 2r\arcsin\left(\sqrt{\sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos(\varphi_1)\cos(\varphi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right)$$

Once, we get the distance in kilometers, we need to compare it with the geofence radius. If our distance d, is greater than the geofence radius then we say that the faculty is not present inside the college premises, else we say he is present inside the college premises.

6. Results

6.1 Geofence

Figure 7: Actual distance and observed distance between thane-Mumbai using Haversine formula

```
D:\projects_internships\faculty-attendence>python haversine.py
enter the latitude of center point:
19.2183                                  //the center lat and lng is of THANE, Maharastra.
enter the longitude of center point:
72.9781
lat of center:  19.2183  lng of center:  72.9781

enter the latitude of your location:     //the location lat and lng is of Mumbai, Maharastra
19.0760
enter the longitude of your location:
72.8777                                                    //the actual distance between
lat of your location:  19.076  lng of your location:  72.8777   thane and mumbai is ~22KM.
the distance between the two points is:  19.015636151592986  Kilometers
```

Figure 8: Actual distance and observed distance between two very close places

```
D:\projects_internships\faculty-attendence>python haversine.py
enter the latitude of center point:
19.1492                                  center is Airoli -> garam masala
enter the longitude of center point:
72.9903                                  location is also in Airoli -> DAV Airoli.
lat of center:  19.1492  lng of center:  72.9903

enter the latitude of your location:     //Distance between the two is around 400 to
19.1477                                  500 meters (0.4 - 0.5 KM)
enter the longitude of your location:
72.9874
lat of your location:  19.1477  lng of your location:  72.9874
the distance between the two points is:  0.3472972994386459  Kilometers
```

Figure 9: Actual distance and observed distance between two sectors within a town

```
D:\projects_internships\faculty-attendence>python haversine.py
enter the latitude of center point:
19.1492                                  center is airoli sector 10
enter the longitude of center point:
72.9903                                  location is airoli sector 8
lat of center:  19.1492  lng of center:  72.9903

enter the latitude of your location:     the actual distance between two place is
19.1483                                  around 700 meters (0.7 KM)
enter the longitude of your location:
72.9948
lat of your location:  19.1483  lng of your location:  72.9948
the distance between the two points is:  0.4831689487889718  Kilometers
```

The error that we got from the above results are as follows: -

1. |19.0157 - 22| = 2.9843 as absolute error from Figure 7.
2. |0.3473 - 0.4| = 0.0527 as absolute error from Figure 8.
3. |0.4832 - 0.7| = 0.2168 as absolute error from Figure 9.

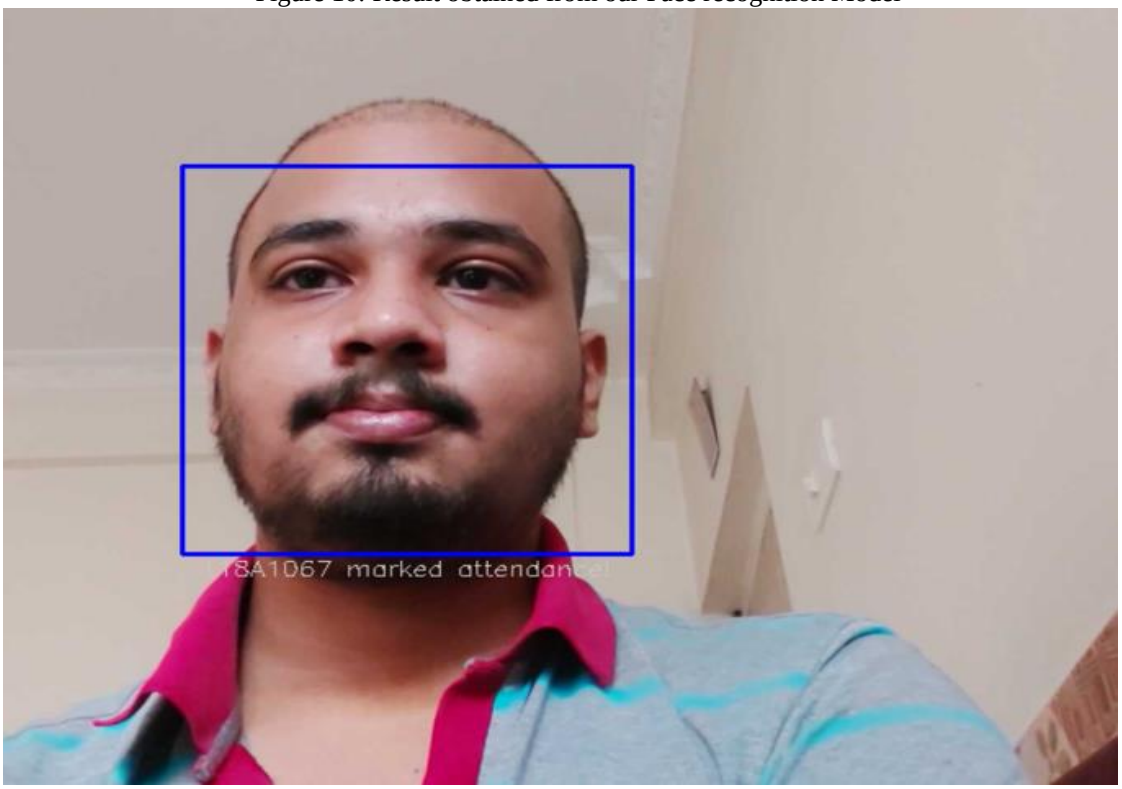
The mean absolute error = 1.0846

The root mean squared error = 2.99

We can see that as the distance between the two places increases the error value increases, this is due to the facts that the haversine formula is used for spheres and earth is not an exact sphere. The distance values are taken from google which gives us the distance value from road. We can safely assume that if a person is within a kilometer or within a few kilometers range the formula will give us correct results.

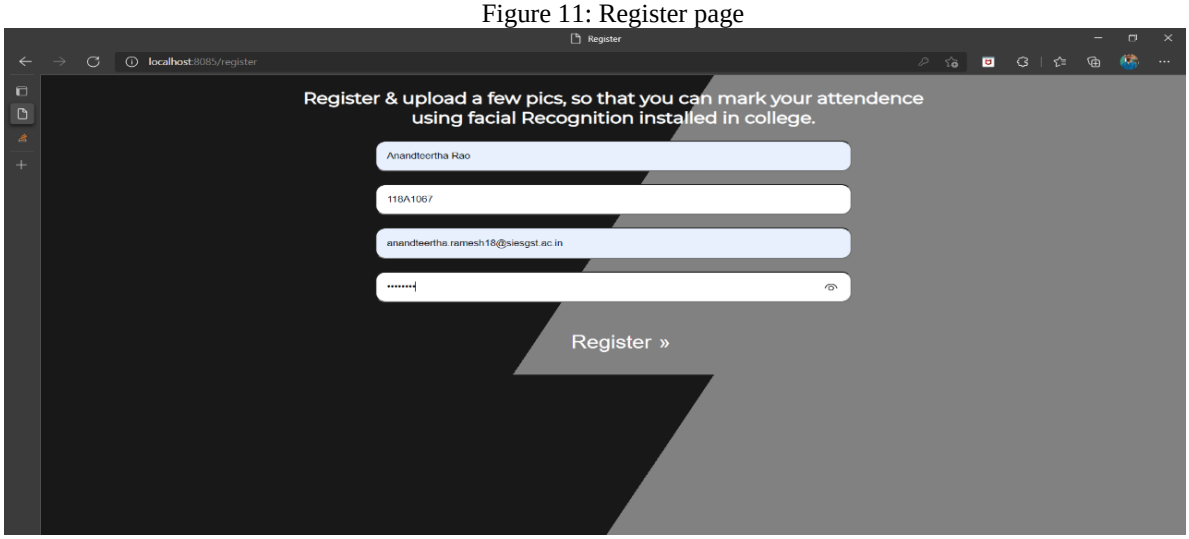
6.2. Face Recognition

Figure 10: Result obtained from our Face recognition Model



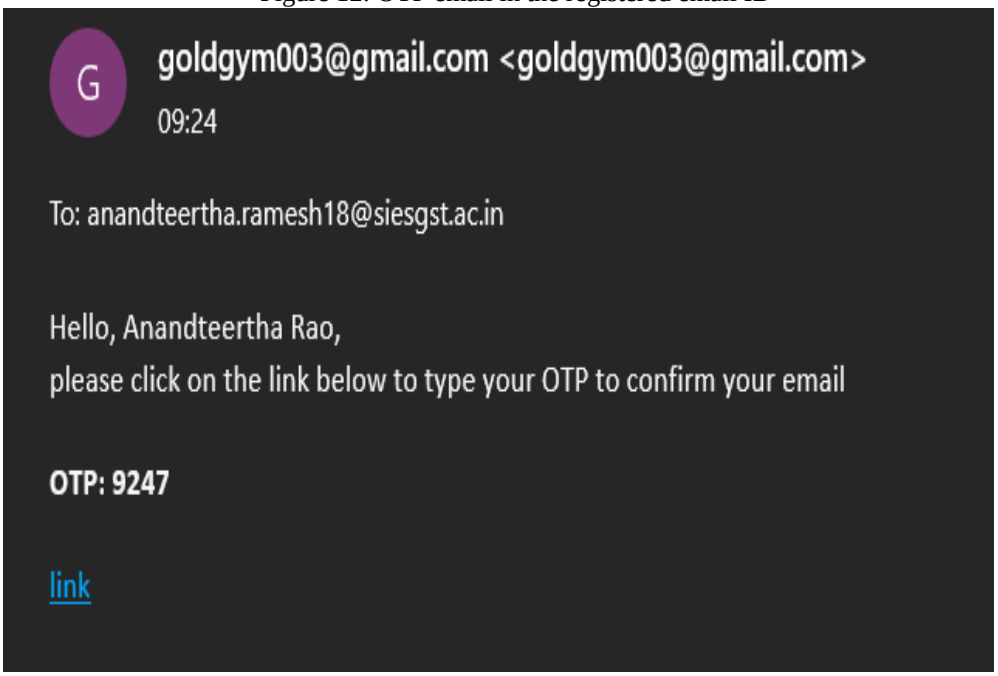
7. Screenshots of working application

Figure 11: Register page



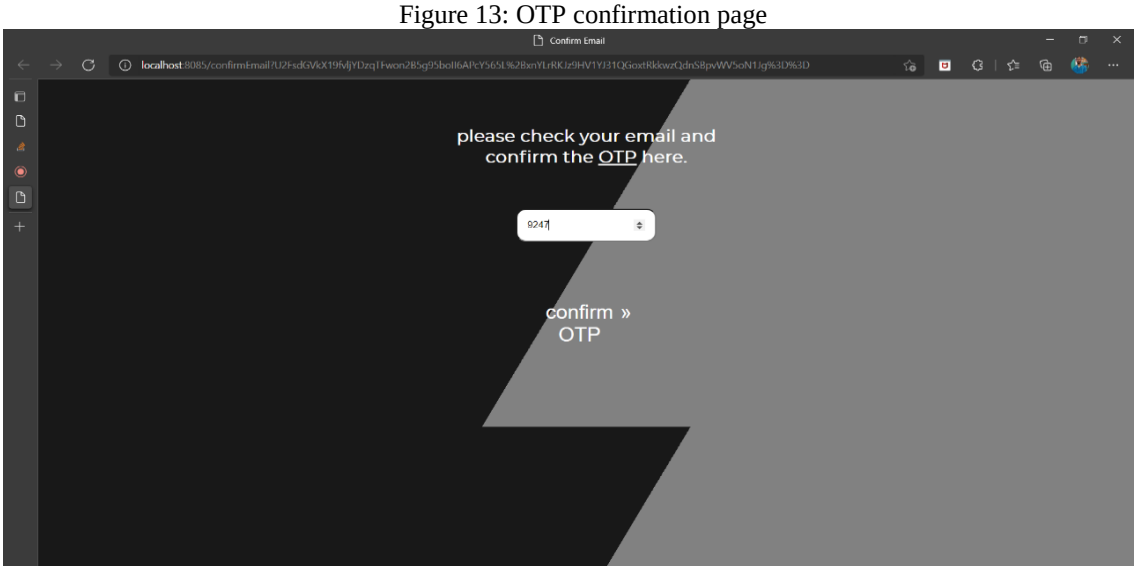
Faculty has to first register into our portal.

Figure 12: OTP email in the registered email ID



Faculty will get an OTP and a link in their registered email address in which they have to click and write their OTP to confirm their email address.

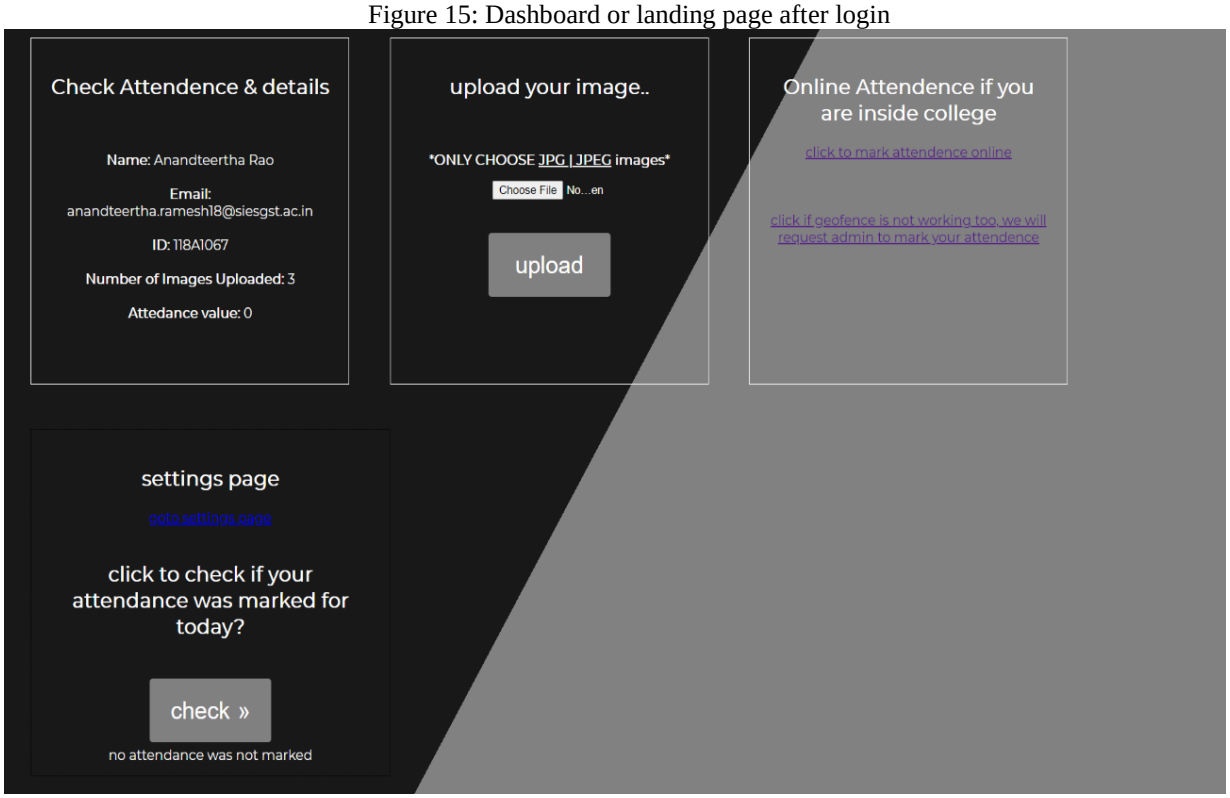
Figure 13: OTP confirmation page



Once their email address is successfully registered, they login

Figure 14: Login page

Figure 15: Dashboard or landing page after login



The dashboard has numerous functionalities, and the user can browse them easily. Using the dashboard, the user can also check their current attendance easily. They can upload their images for face recognition, update their password or any other details. Finally, they can even request the admin to mark their attendance.

Additionally, they can click on mark attendance using geofence if they are inside the college premises.

Figure 16: Attendance *not* marked as user is not within the college

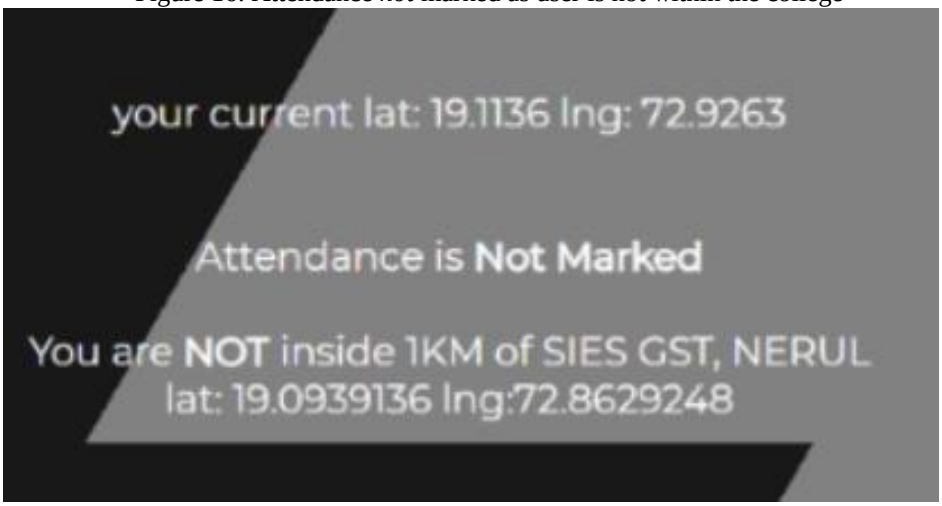
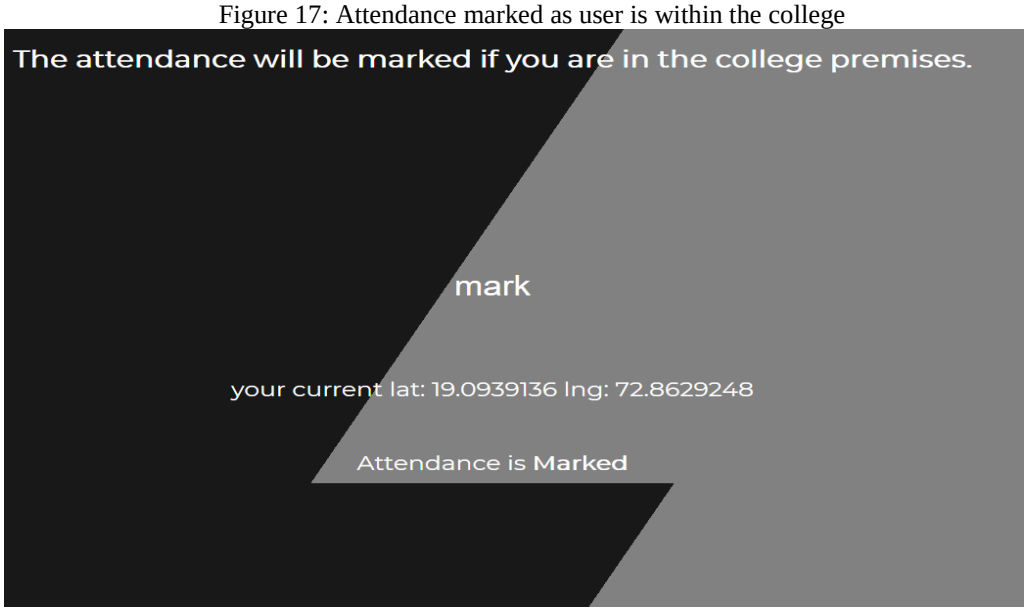


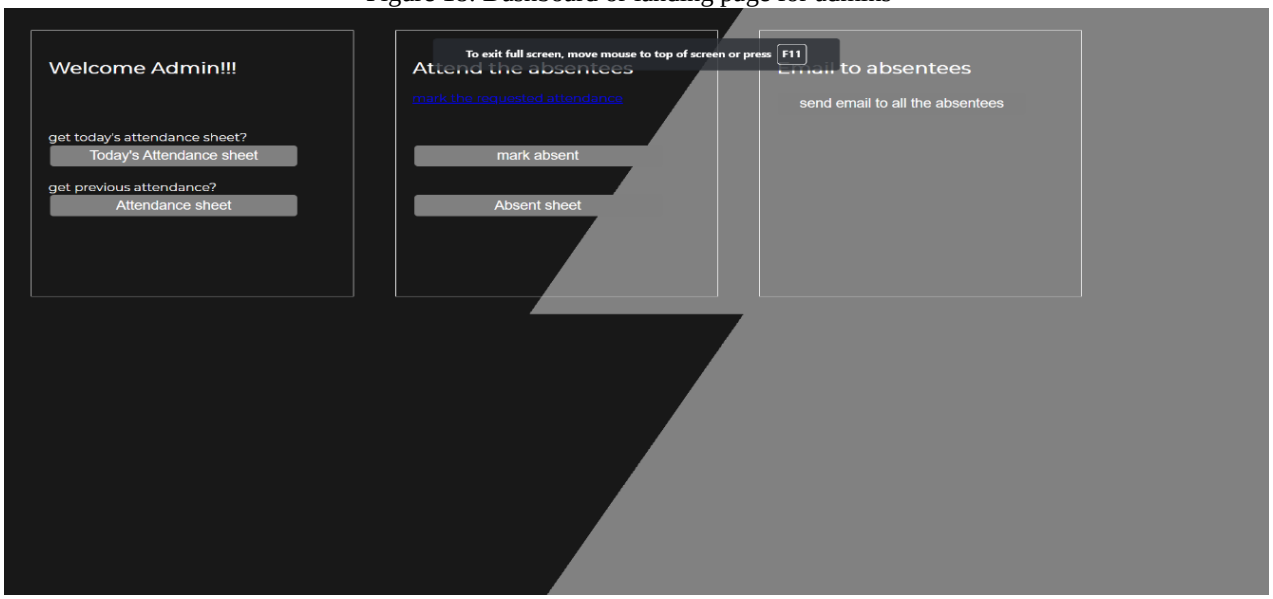
Figure 17: Attendance marked as user is within the college



The user has to stand in front of the camera which will be present in the college and his attendance will marked if he has submitted or uploaded his facial images in the portal.

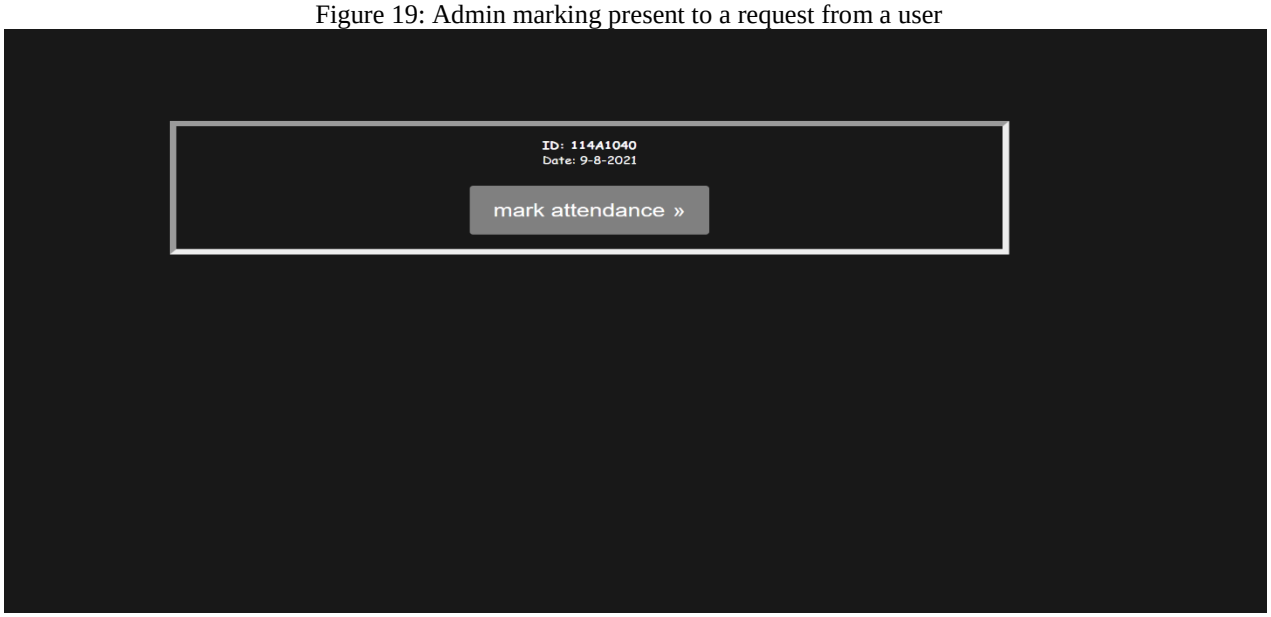
The admin can login to their portal too, using the predefined credentials. Once the login is done, they will be redirected to the landing page for admins.

Figure 18: Dashboard or landing page for admins



The admin portal allows the admin to mark the attendance of those users who have requested him to mark their attendance. It is the job of the admin to know if that user was indeed present or not.

Figure 19: Admin marking present to a request from a user



8. Additional Features

- Once attendance is marked it won't be remarked.
- After the E.O.D the admin marks absent to anyone who had not marked their attendance for the day by a simple click.
- Once the user is marked absent, there won't be any changes done in their attendance for that day.
- As we have allowed for three different ways to mark the attendance, even if there is some hardware issue of camera, the whole system won't go down. There are other alternatives given which makes this system extremely fault tolerant.
- Absentees will be notified using email that they have been marked absent.
- A notification in the portal will be displayed to users as soon as they are marked present.

9. Conclusion

The faculty attendance system was implemented successfully and each module produces the desired results. Our application has a single server which was implemented using node JS and took extensive measures to ensure error free execution of the application. The traffic of our application may be fluctuating, for that reason and for ease of modifications for the next versions, microservices architecture [4] would prove to be more efficient. In our

case, the two microservices would be used for face recognition model and general server which would take care of all database queries and page requests. Many problems such as, faculty trying to mark their attendance once the admin has marked absent to users who failed to mark their attendance for the day, faculty unknowingly trying to mark their attendance multiple times during the same day, get down all the images from the database to the admin local machine, etc., were solved with an efficient solution.

Attendance using geofence gave the desired results. It gave us positive result when we were within the geofence radius. It also displayed not marked attendance when were not within the geofence radius. We can hence conclude that our system is working competently and can be implemented in any organization with slight or no modifications.

References (APA)

Pavithra P, Poornima Ragavi G, Pradep B, & Vinodh Kumar G, UG Scholars, Department of ECE, Kathir College of Engineering, Tamil Nādu, India Lecture access and faculty attendance system.

Adam Geitgey <ageitgey@gmail.com> face-recognition package - https://face-recognition.readthedocs.io/en/latest/face_recognition.html

Geofence haversine - https://en.wikipedia.org/wiki/Haversine_formula.

Microservices - https://www.bmc.com/blogs/microservices-architecture/