PROJECT REPORT ON

Automated and No-Touch Attendance System

DECLARATION

We hereby declare that the project entitled "Automated and No-Touch Attendance System" is an authentic record of our own work carried out at Punjab Engineering College (Deemed to be University), as a requirement of Minor Project for the award of degree of Bachelor of Technology (Computer Science of Engineering), under the guidance of Prof. Poonam Saini and Prof. Amandeep Kaur (Department of Computer Science and Engineering) during the period August 2020 to December 2020.

Bhanvi Dewan	Abhishek Chauhan	Harmanpreet Kaur	Arzoo Goyal
(18103014)	(18103029)	(18103037)	(18103087)
_____	_____	_____	_____

CERTIFICATE

This is to certify that the project entitled "Automated and No-Touch Attendance System" submitted by Bhanvi Dewan, Abhishek Chauhan, Harmanpreet Kaur and Arzoo Goyal in partial fulfillment for the requirements for the award of the degree of Bachelor of Technology (Computer Science of Engineering) from the institute Punjab Engineering College (Deemed to be University). This project is a Bonafide work carried out by them under my guidance and supervision and the report has been approved as it satisfies the academic requirement in respect of Minor work. The results of the work have not been submitted to any other institute or university.

Prof. Poonam Saini
PEC, Chandigarh
__ December 2020

Prof. Amandeep Kaur
PEC, Chandigarh
__ December 2020

ACKNOWLEDGEMENT

We have taken a lot of deliberations in this venture. But this would not have been possible without the helping and backing of many people. We want to extend our heartfelt appreciation and thank them. We would like to take this opportunity to express our profound gratitude and deep regards to our mentor Prof. Poonam Saini and Prof. Amandeep Kaur for their exemplary guidance, monitoring and constant encouragement throughout the course of this project.

TABLE OF CONTENTS

INTRODUCTION

Our project aims to develop a system wherein a face recognition module will identify a person and then mark his/her attendance in the respective organization without using traditional methods like register entry or fingerprint-based biometric attendance, which are tedious and time-consuming. This will reduce the amount of paper resources needed in the management of attendance data while gauging and eliminating proxies. This could be even more beneficial in the recent pandemic period (Covid-19). A major benefit of this will be the eradication of the need to compile attendance manually, thus reducing the amount of administrative work.

The human face is a unique representation of individual identity. Face recognition is defined as a biometric method in which identification of an individual is performed by comparing real-time captured images with stored images of that person in the database.

MOTIVATION

Our project aims to develop a system wherein a face recognition module will identify a person and then help to mark his/her attendance in the respective organization without using traditional methods like register entry or thumb or fingerprint-based biometric attendance. This could be even more beneficial in the recent pandemic period (Covid-19).

Hence, we have proposed an **automated attendance system** based on face recognition with the following benefits:

- Eliminate the time and effort wasted in taking attendance manually in organizations.
- Reducing the amount of paper resources needed in attendance data management.
- Gauging and eliminating proxies.
- Not having to compile attendance manually, instead store it in a database that can be accessed by authorized personnel only.
- Reduce the amount of administrative work.

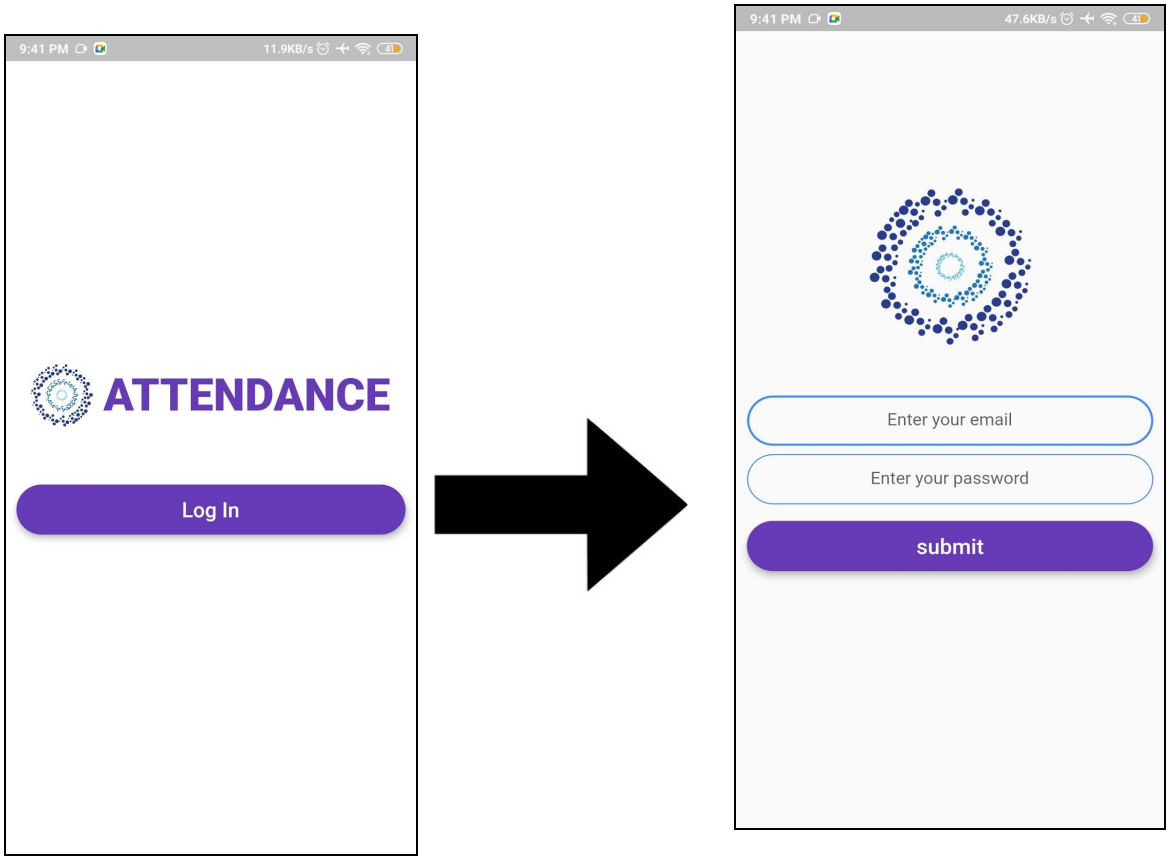
AIMS AND OBJECTIVES

- The objective of our project is to develop face recognition based automated attendance system with following tasks:
- To detect the face segment from the image.
- To extract the useful features from the face detected.
- To classify the features in order to recognize the face detected.
- To record the attendance of the User.
- To notify each user that his/her attendance has been marked.
- To present a complete record of the user's attendance, including leaves, percentage, etc.

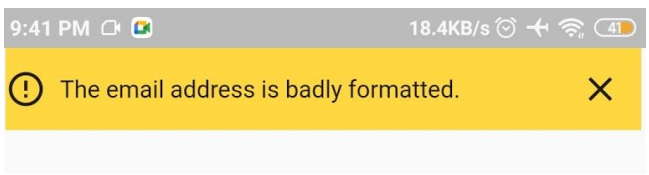
APP EXECUTION

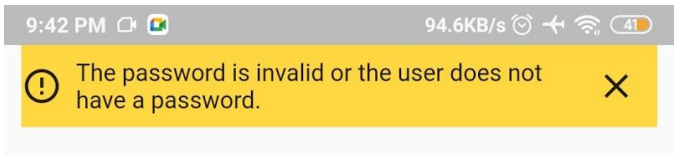
1. LOGIN

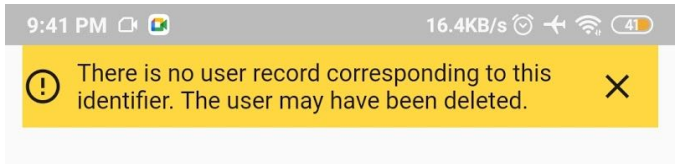
- **When opening the Application for the first time, the user is shown the first screen for the login process.**



- **If the entered email or password is wrong, then the errors are shown as below:**







2. HOME TAB

- **The home tab includes attendance percentage, number of days present and absent (unpaid leaves) in a particular month. Also, the paid leaves which are recorded annually.**

Fig.1

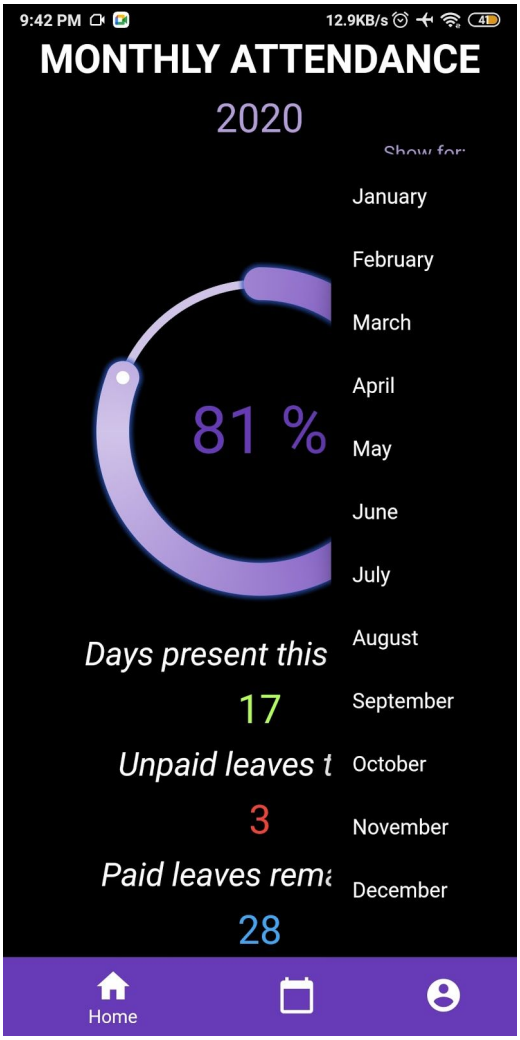


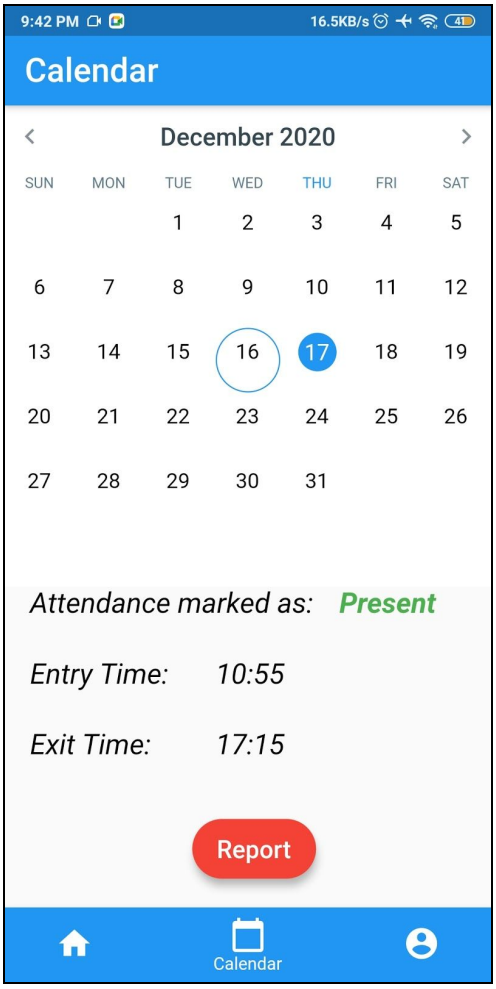
Fig.2

- **On changing the month from the dropdown button, the details will be changed of that particular month, except the annual paid leaves. (Fig.2)**

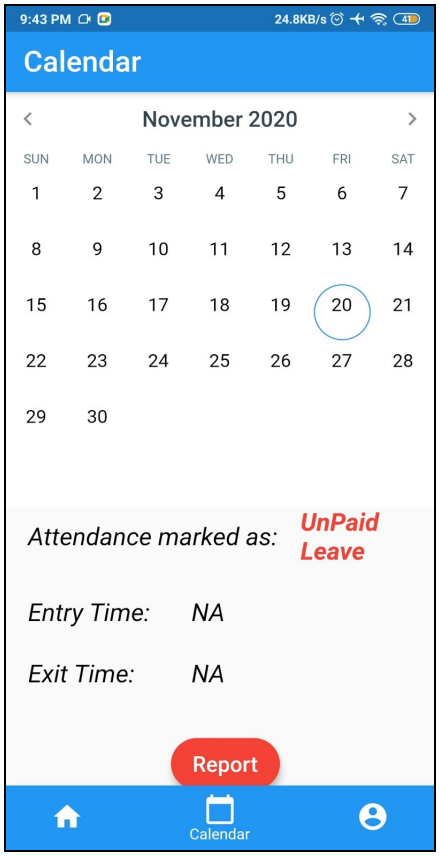
3. CALENDAR TAB

- **The calendar tab contains detailed information for each day of the year, displaying the attendance status for the day, and if the person were present, the entry and exit time.**

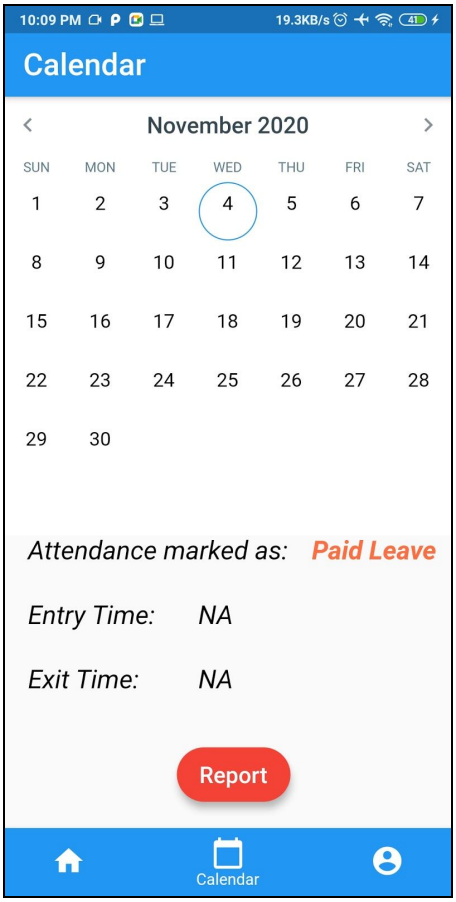
  Present shown along with entry time and exit time:



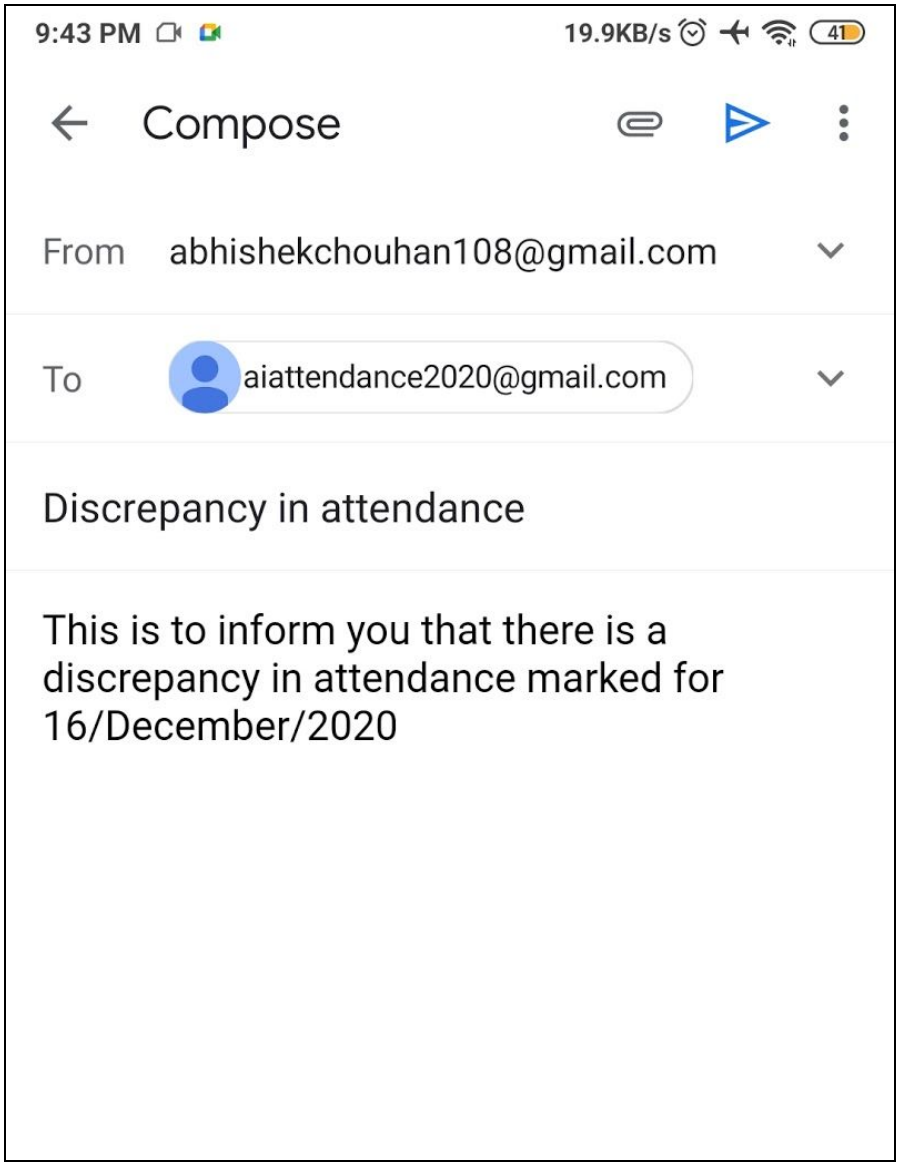
Absent (unpaid leave):



Absent (paid leave):

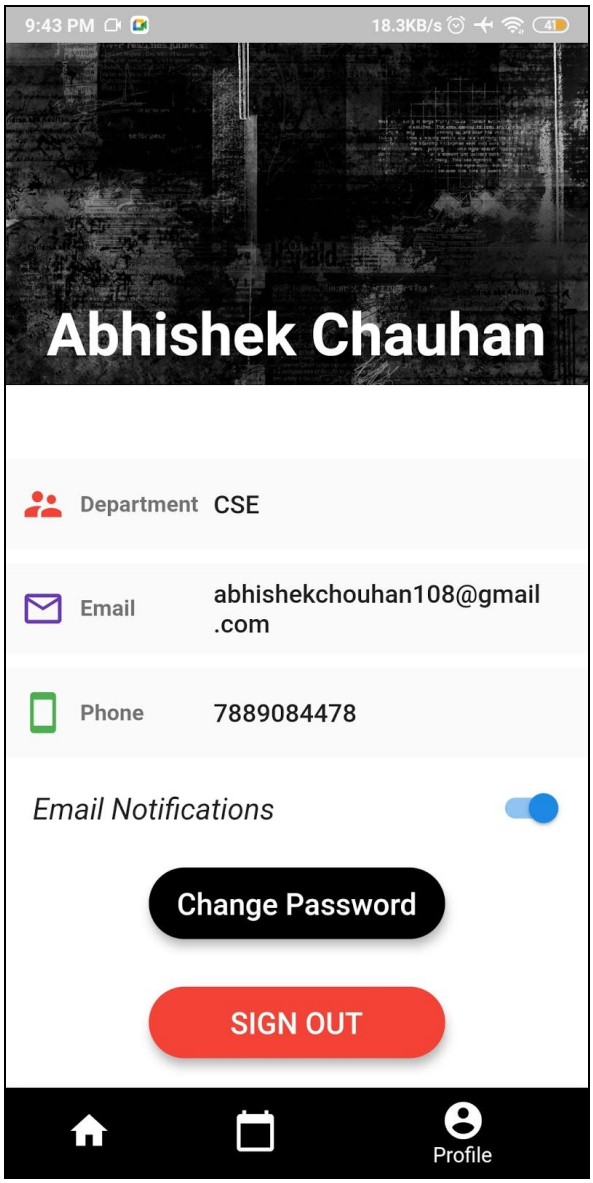


- **If the user finds any discrepancy in the record, he can report this error through an email which will be sent to admin.**

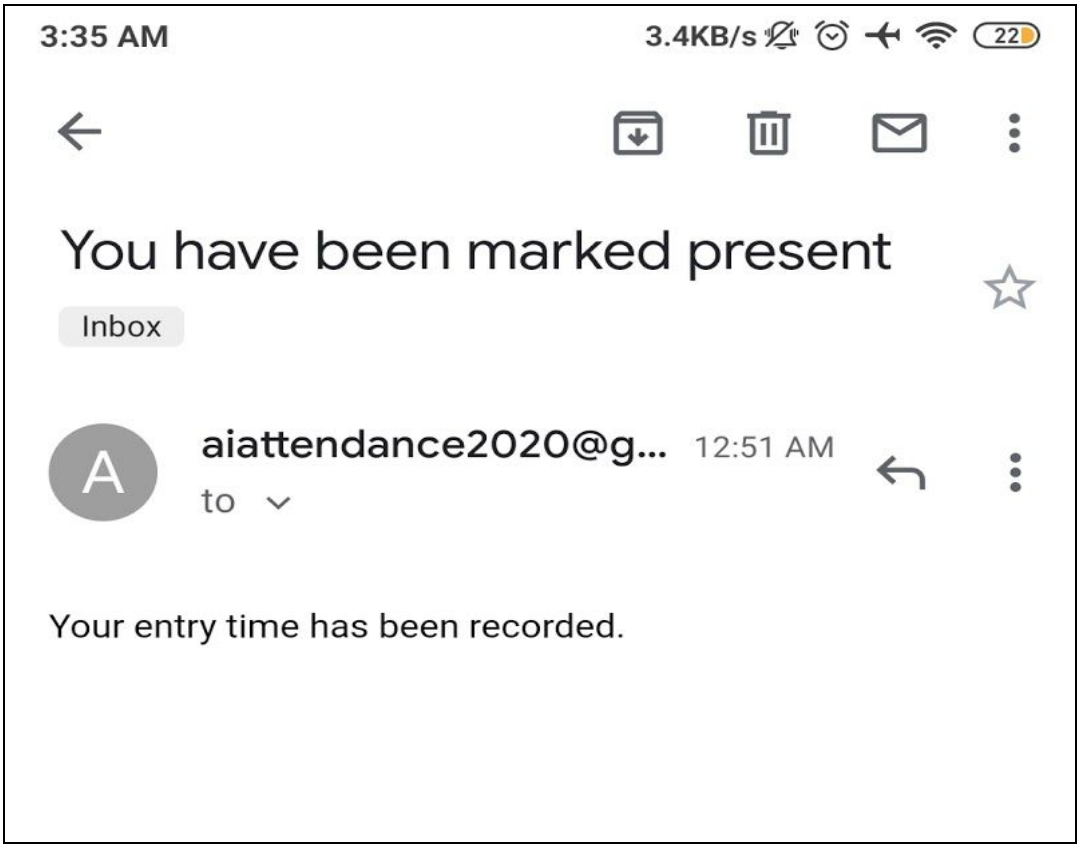


4. PROFILE TAB

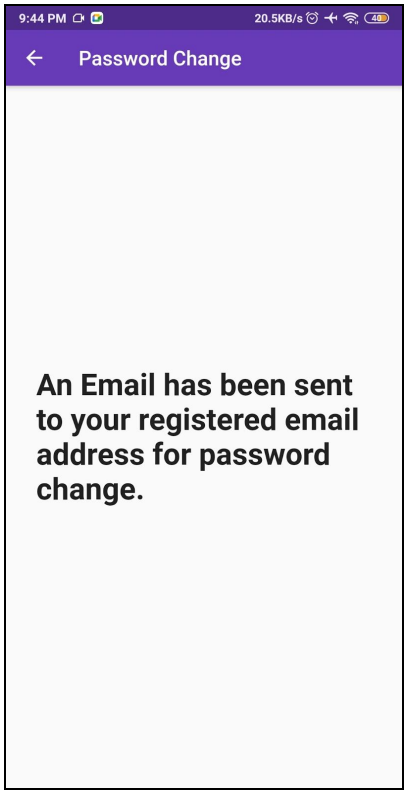
The profile tab contains the user's name, email id, department, and phone number. A switch button to enable email notifications is also included, so is a "Change Password" button. A sign-out button signs out the user. Unless the person signs out he/she will be kept logged in.

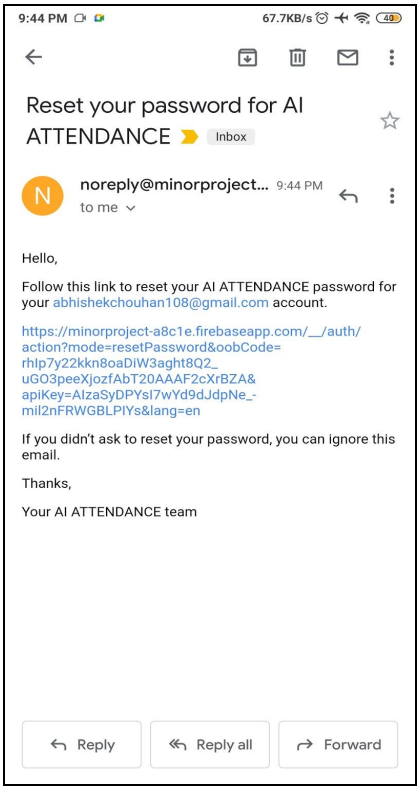


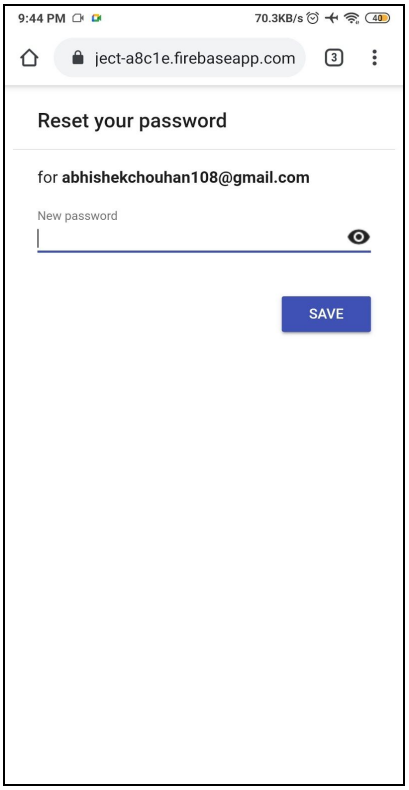
- **On enabling email notification, an email notification is sent to the person confirming that his/her attendance has been marked and the changes updated in the database.**



- **If the user wants to change password, an email to change password is sent to him.**







- **If the user doesn't sign out, he/she will automatically be signed in the next time he/she opens the app. After the user signs out, he is redirected to the login page.**

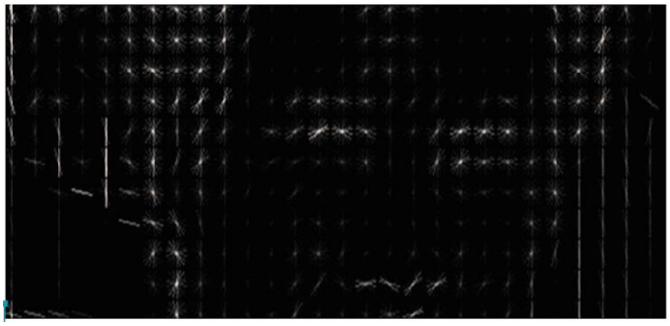
INCORPORATION OF ARTIFICIAL INTELLIGENCE:

Artificial intelligence has been used for face recognition. We're using a library (named face-recognition) that uses a method called Histogram of Oriented Gradients — or just *HOG* for short.

It starts by making the image black and white because we don't need color data to find faces. The system figures out how dark the current pixel is compared to the pixels directly surrounding it. Then it draws an arrow showing in which direction the image is getting darker. Repeating that process for every single pixel in the image, we end up with every pixel being replaced by an arrow. These arrows are called *gradients* and they show the flow from light to dark across the entire image.

But we do not need the data for every single pixel, so we'll break up the image into small squares of 16x16 pixels each. In each square, it'll count up how many gradients point in each major direction (how many point up, point up-right, point right, etc.).

Then it'll replace that square in the image with the arrow directions that were the strongest. The end result is that it turns the original image into a very simple representation that captures the basic structure of a face regardless of image brightness.



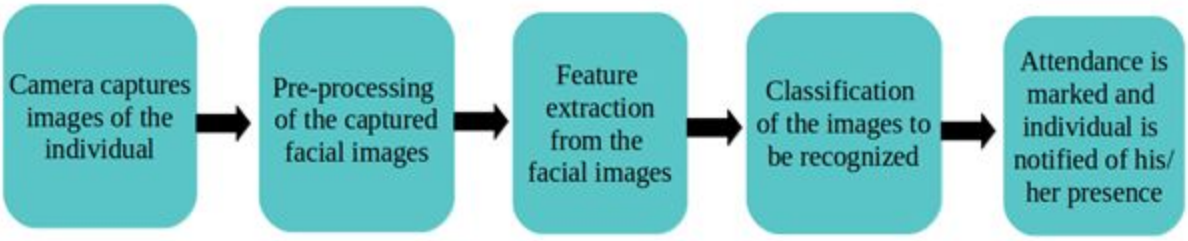
Next, it will try to warp each picture so that the eyes and lips are always in the sample place in the image. To do this, it uses an algorithm called face landmark estimation. It comes up with 68 specific points (called *landmarks*) that exist on every face — the top of the chin, the outside edge of each eye, the inner edge of each eyebrow, etc. Then it will train a machine learning algorithm to be able to find these 68 specific points on any face.

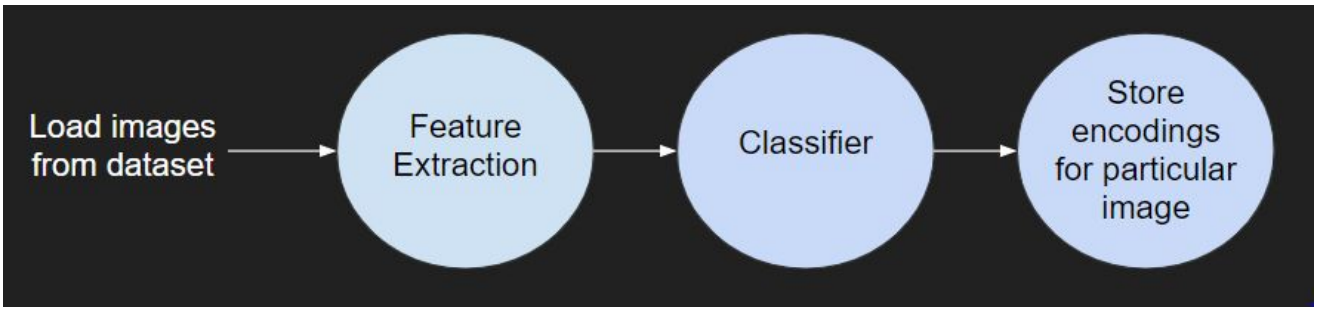
It'll simply rotate, scale and shear the image so that the eyes and mouth are centred as best as possible. It is only going to use basic image transformations like rotation and scale that preserve parallel lines (called affine transformations).

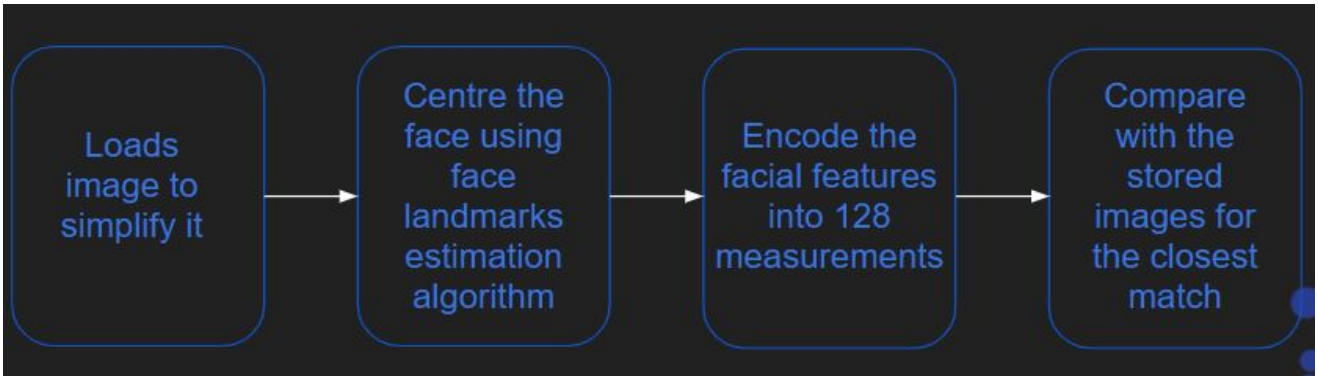
But we also need to be able to recognize faces in milliseconds, not hours. What we need is a way to extract a few basic measurements from each face. Then we could measure our unknown face the same way and find the known face with the closest measurements. The solution is to train the model to generate 128 measurements for each face.

The model learns to reliably generate 128 measurements for each person. The 128 measurements of each face are called an embedding. Now it finds the person in our database of known people who have the closest measurements to our test image. It uses a simple linear SVM classifier.

Work-Flow


Camera captures images of the individual
Pre-processing of the captured facial images
Feature extraction from the facial images
Classification of the images to be recognized
Attendance is marked and individual is notified of his/ her presence


Load images from dataset
Feature Extraction
Classifier
Store encodings for particular image

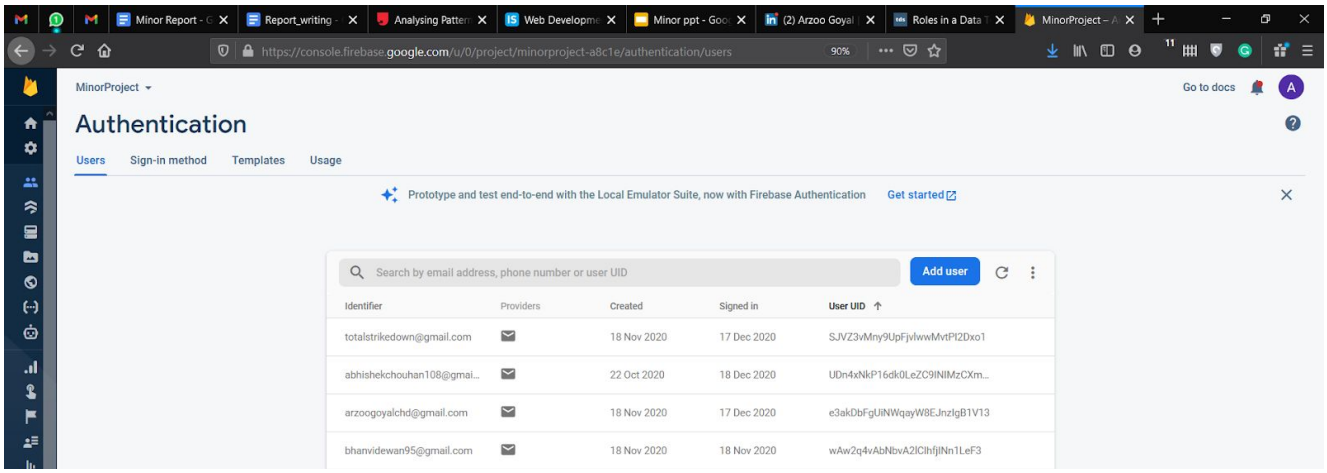

Loads image to simplify it
Centre the face using face landmarks estimation algorithm
Encode the facial features into 128 measurements
Compare with the stored images for the closest match

INCORPORATION OF WEB TECHNOLOGIES:

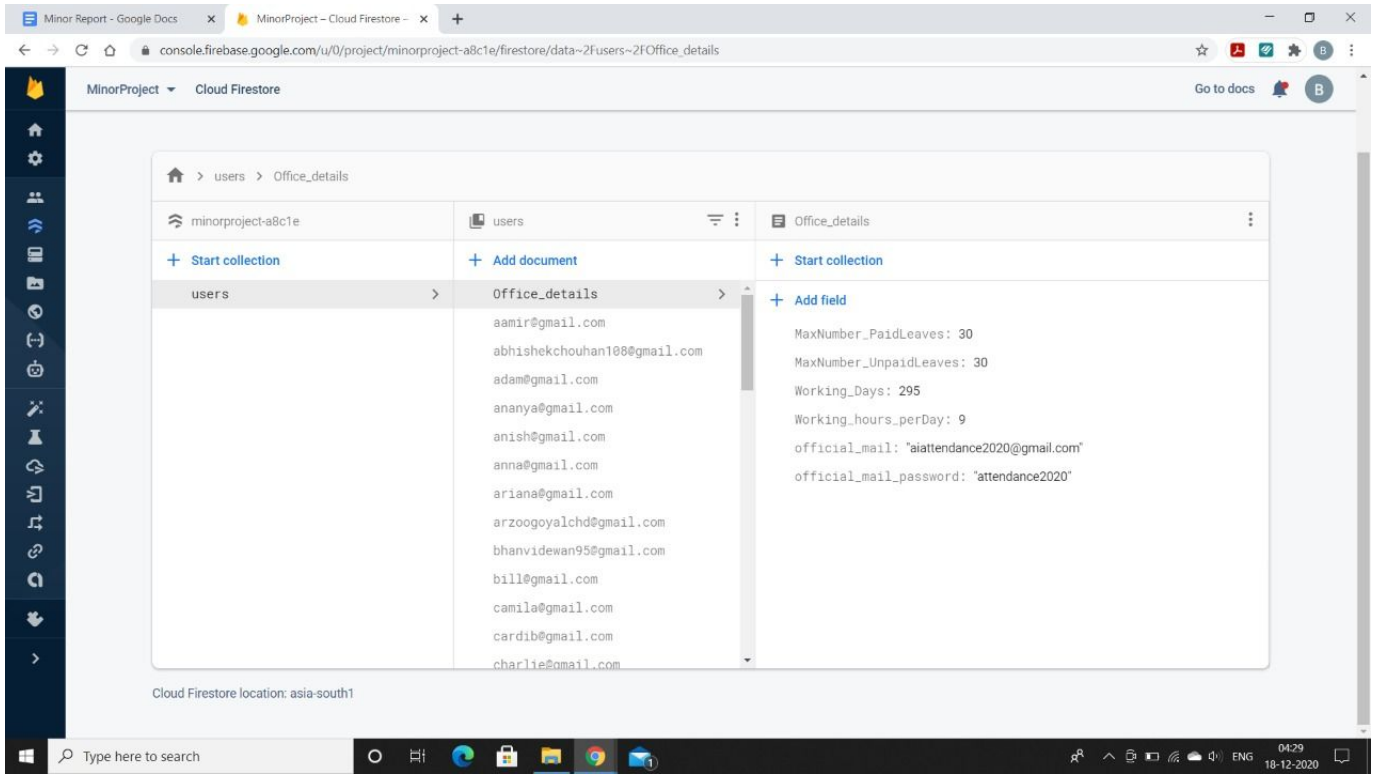
The technologies which we have utilized are Google Firebase and Flutter. **Flutter** in Android Studio makes up our front end. The app has been written entirely in Dart (a language in the Flutter package).

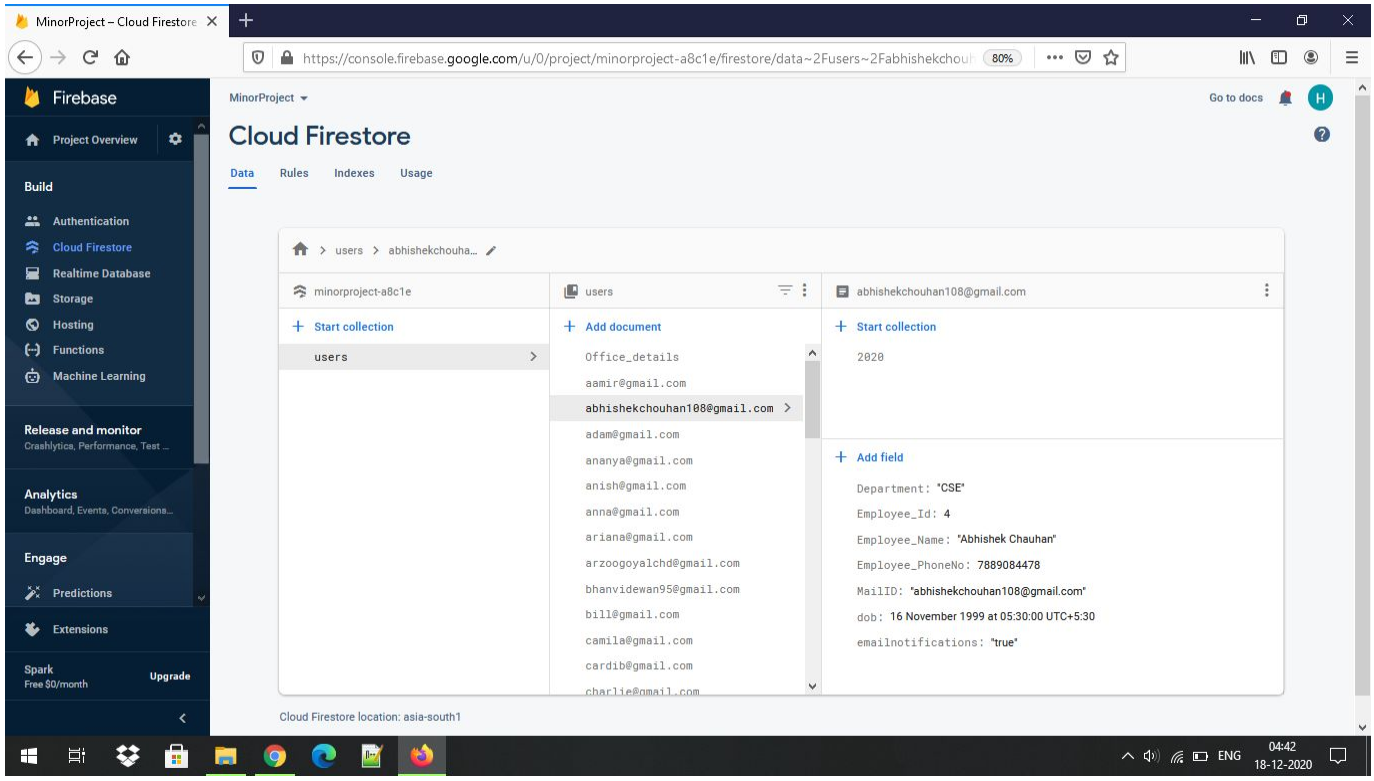
Google Firebase is the server for the project, of which we have utilized the following services:

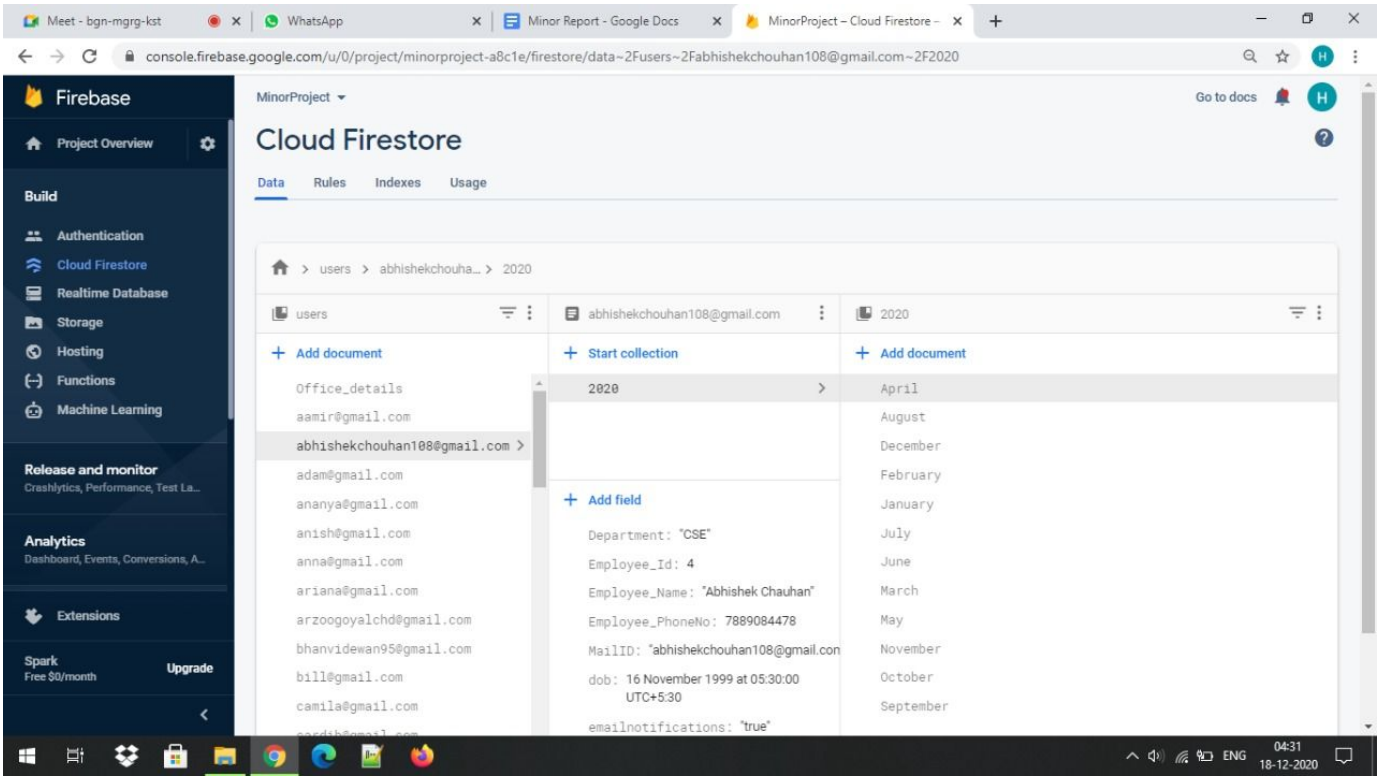
- **Firebase Authentication** is employed at the login-in stage to authenticate the user.



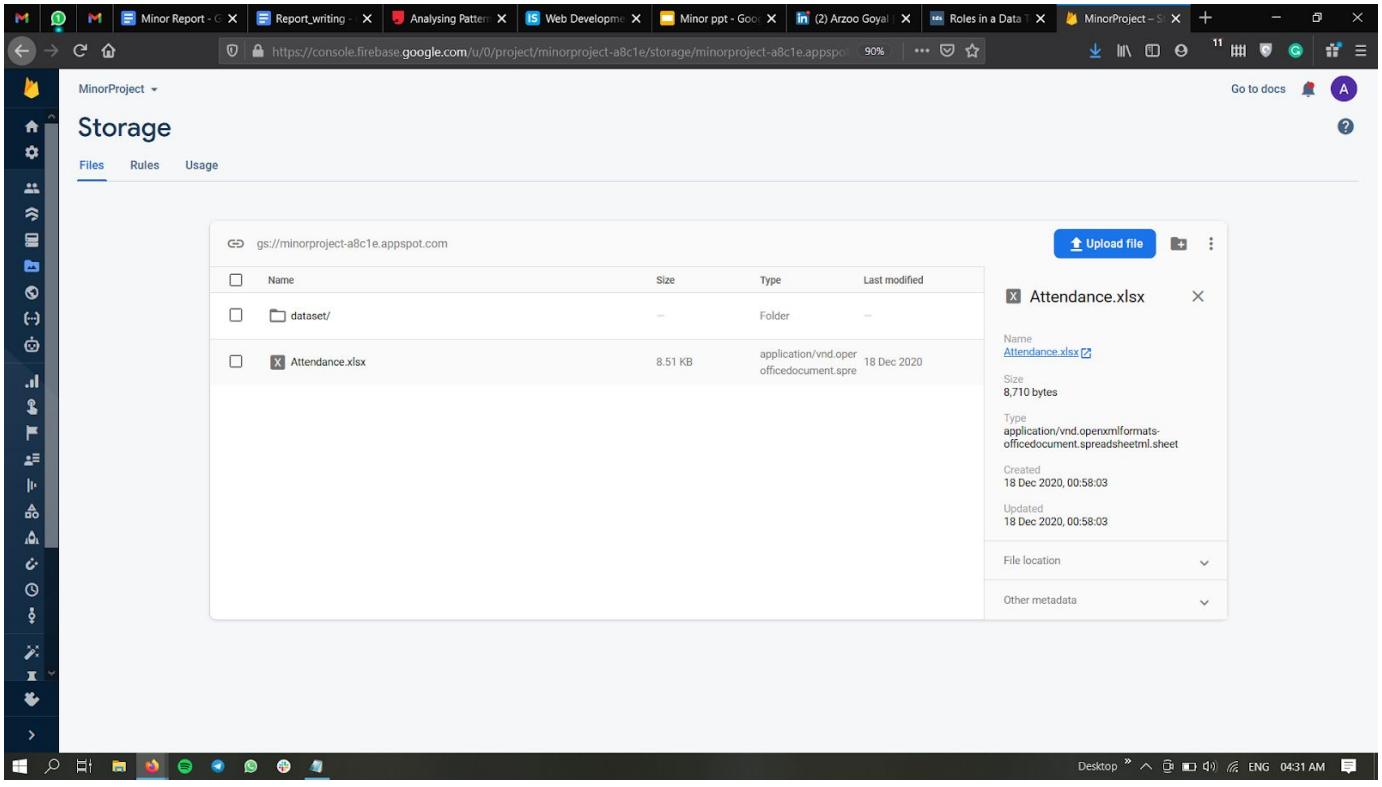
- We have used **Cloud Firestore**, a NoSQL document-oriented database in which the structures alternate between documents and collections of documents. This contains the entire record of the attendance. The entire database is created and the data updated using python. Cloud Firestore is accessed by this code that marks attendance using face recognition, and also by the app, that reads and displays the data to the user.

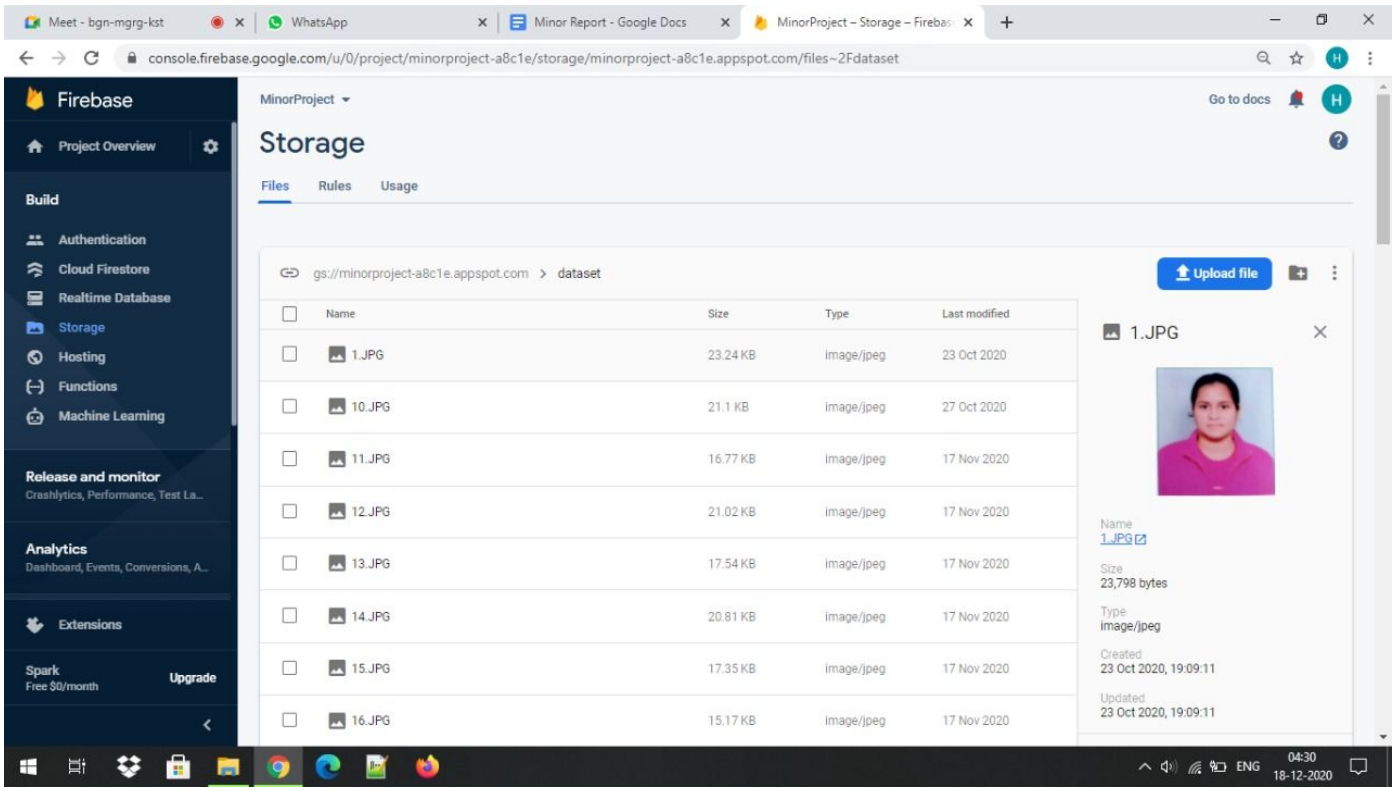

Minor Report - Google Docs
MinorProject – Cloud Firestore -
console.firebase.google.com/u/0/project/minorproject-a8c1e/firestore/data~2Fusers~2FOffice_details
MinorProject
Cloud Firestore
Go to docs
users
Office_details
minorproject-a8c1e
Start collection
users
Add document
Office_details
aamir@gmail.com
abhishekchouhan108@gmail.com
adam@gmail.com
ananya@gmail.com
anish@gmail.com
anna@gmail.com
ariana@gmail.com
arzoogoyalchd@gmail.com
bhanvidewan95@gmail.com
bill@gmail.com
camila@gmail.com
cardib@gmail.com
Add field
MaxNumber_PaidLeaves: 30
MaxNumber_UnpaidLeaves: 30
Working_Days: 295
Working_hours_perDay: 9
official_mail: "aiattendance2020@gmail.com"
official_mail_password: "attendance2020"
Cloud Firestore location: asia-south1
Type here to search
ENG
04:29
18-12-2020


MinorProject – Cloud Firestore
https://console.firebase.google.com/u/0/project/minorproject-a8c1e/firestore/data~2Fusers~2Fabhishekchou
80%
Firebase
MinorProject
Go to docs
Project Overview
Build
Authentication
Cloud Firestore
Realtime Database
Storage
Hosting
Functions
Machine Learning
Release and monitor
Crashlytics, Performance, Test ...
Analytics
Dashboard, Events, Conversions...
Engage
Predictions
Extensions
Spark
Free $0/month
Upgrade
Cloud Firestore
Data
Rules
Indexes
Usage
users
abhishekchouha...
minorproject-a8c1e
Start collection
users
Add document
Office_details
aamir@gmail.com
abhishekchouhan108@gmail.com
adam@gmail.com
ananya@gmail.com
anish@gmail.com
anna@gmail.com
ariana@gmail.com
arzoogoyalchd@gmail.com
bhanvidewan95@gmail.com
bill@gmail.com
camila@gmail.com
cardib@gmail.com
abhishekchouhan108@gmail.com
Start collection
2020
Add field
Department: "CSE"
Employee_Id: 4
Employee_Name: "Abhishek Chauhan"
Employee_PhoneNo: 7889084478
MailID: "abhishekchouhan108@gmail.com"
dob: 16 November 1999 at 05:30:00 UTC+5:30
emailnotifications: "true"
Cloud Firestore location: asia-south1
ENG
04:42
18-12-2020

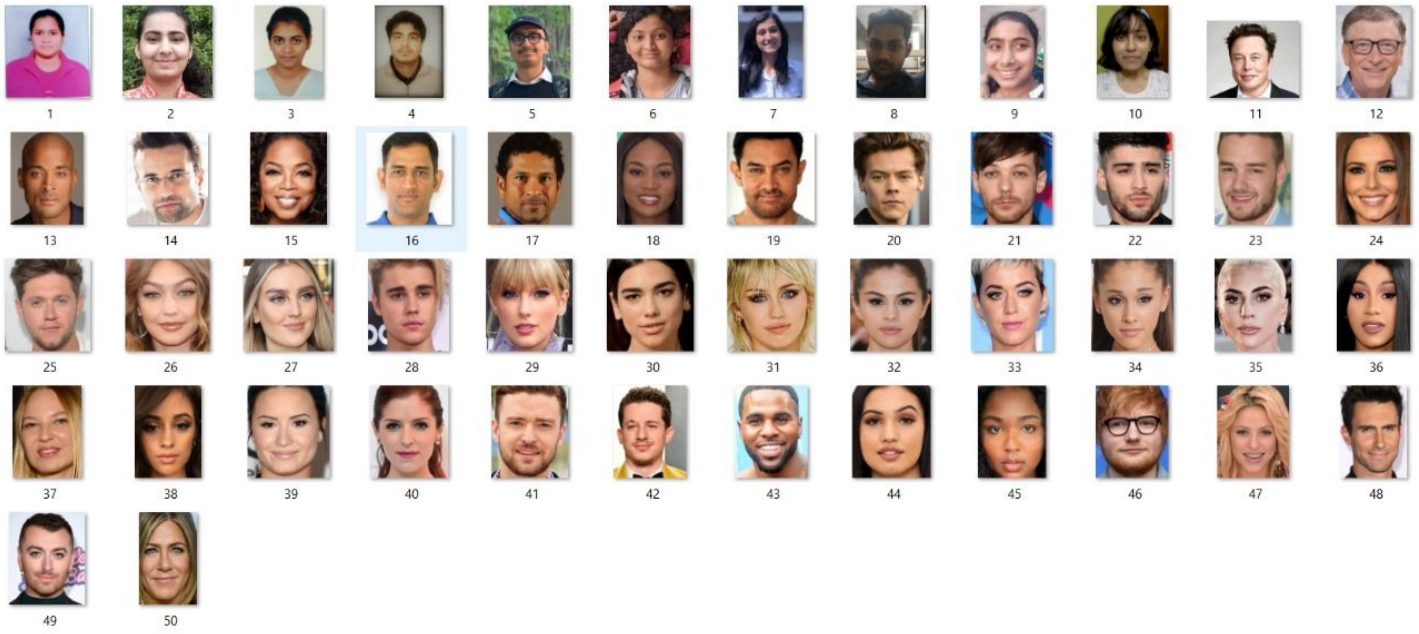


- **Firebase Storage contains the folder for the images of the employees which are matched against the pictures taken every day by the office cameras.**

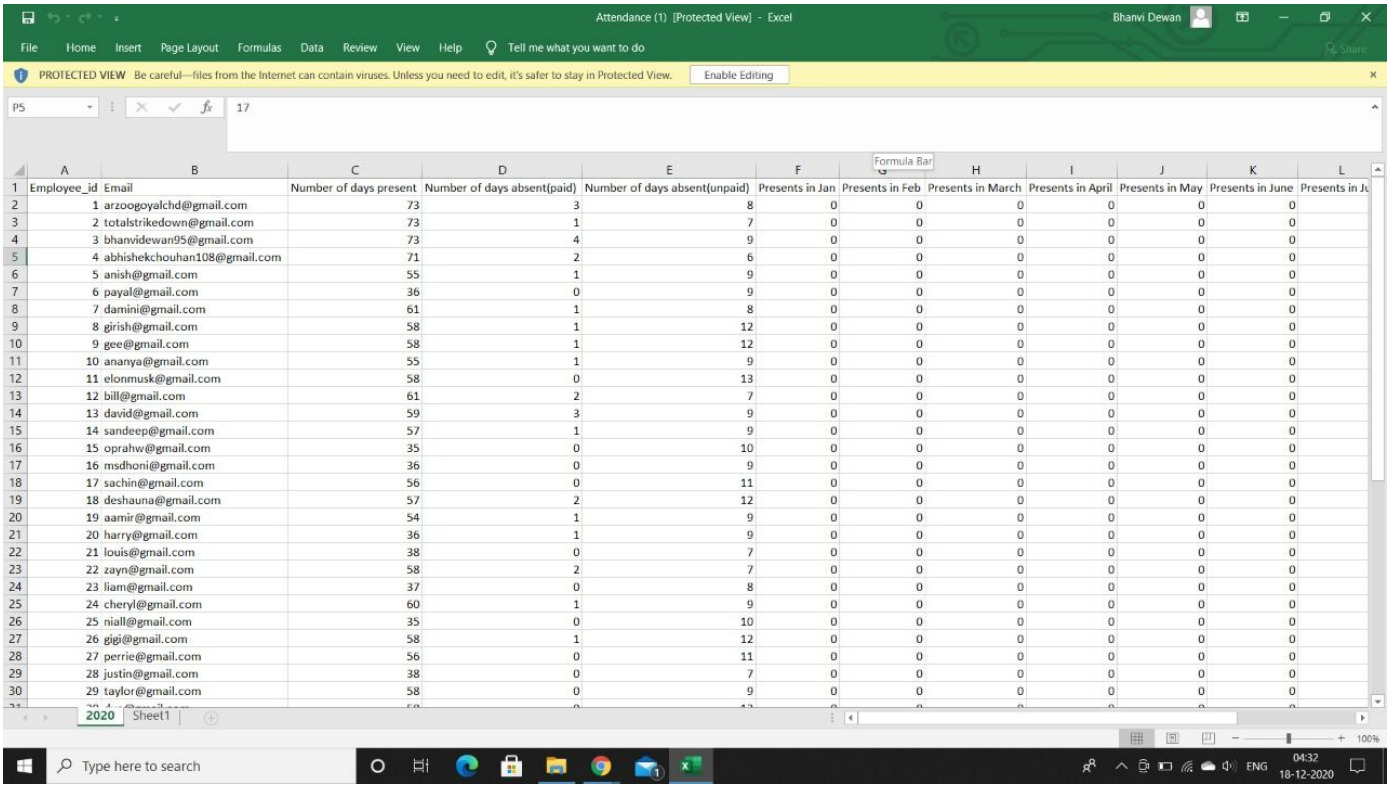



Meet - bgn-mgrg-kst
WhatsApp
Minor Report - Google Docs
MinorProject – Storage – Firebase
console.firebase.google.com/u/0/project/minorproject-a8c1e/storage/minorproject-a8c1e.appspot.com/files~2Fdataset
Firebase
Project Overview
Build
Authentication
Cloud Firestore
Realtime Database
Storage
Hosting
Functions
Machine Learning
Release and monitor
Crashlytics, Performance, Test La...
Analytics
Dashboard, Events, Conversions, A...
Extensions
Spark
Free $0/month
Upgrade
MinorProject
Go to docs
Storage
Files
Rules
Usage
gs://minorproject-a8c1e.appspot.com > dataset
Upload file
Name
Size
Type
Last modified
1.JPG 23.24 KB image/jpeg 23 Oct 2020
10.JPG 21.1 KB image/jpeg 27 Oct 2020
11.JPG 16.77 KB image/jpeg 17 Nov 2020
12.JPG 21.02 KB image/jpeg 17 Nov 2020
13.JPG 17.54 KB image/jpeg 17 Nov 2020
14.JPG 20.81 KB image/jpeg 17 Nov 2020
15.JPG 17.35 KB image/jpeg 17 Nov 2020
16.JPG 15.17 KB image/jpeg 17 Nov 2020
1.JPG
Name
1.JPG
Size
23,798 bytes
Type
image/jpeg
Created
23 Oct 2020, 19:09:11
Updated
23 Oct 2020, 19:09:11
ENG
04:30
18-12-2020

Dataset:


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50

This is the downloadable excel sheet containing the attendance data of all the employees.



Attendance (1) [Protected View] - Excel

Bhanvi Dewan

File Home Insert Page Layout Formulas Data Review View Help Tell me what you want to do

PROTECTED VIEW Be careful—files from the Internet can contain viruses. Unless you need to edit, it's safer to stay in Protected View. Enable Editing

P5 17

Formula Bar

	A	B	C	D	E	F	G	H	I	J	K	L
1	Employee_id	Email	Number of days present	Number of days absent(paid)	Number of days absent(unpaid)	Presents in Jan	Presents in Feb	Presents in March	Presents in April	Presents in May	Presents in June	Presents in Ju
2	1	arzoogoyalchd@gmail.com	73	3	8	0	0	0	0	0	0	
3	2	totalstrikedown@gmail.com	73	1	7	0	0	0	0	0	0	
4	3	bhanvidewan95@gmail.com	73	4	9	0	0	0	0	0	0	
5	4	abhishekchouhan108@gmail.com	71	2	6	0	0	0	0	0	0	
6	5	anish@gmail.com	55	1	9	0	0	0	0	0	0	
7	6	payal@gmail.com	36	0	9	0	0	0	0	0	0	
8	7	damini@gmail.com	61	1	8	0	0	0	0	0	0	
9	8	girish@gmail.com	58	1	12	0	0	0	0	0	0	
10	9	gee@gmail.com	58	1	12	0	0	0	0	0	0	
11	10	ananya@gmail.com	55	1	9	0	0	0	0	0	0	
12	11	elonmusk@gmail.com	58	0	13	0	0	0	0	0	0	
13	12	bill@gmail.com	61	2	7	0	0	0	0	0	0	
14	13	david@gmail.com	59	3	9	0	0	0	0	0	0	
15	14	sandeep@gmail.com	57	1	9	0	0	0	0	0	0	
16	15	oprahw@gmail.com	35	0	10	0	0	0	0	0	0	
17	16	msdhoni@gmail.com	36	0	9	0	0	0	0	0	0	
18	17	sachin@gmail.com	56	0	11	0	0	0	0	0	0	
19	18	deshauna@gmail.com	57	2	12	0	0	0	0	0	0	
20	19	aamir@gmail.com	54	1	9	0	0	0	0	0	0	
21	20	harry@gmail.com	36	1	9	0	0	0	0	0	0	
22	21	louis@gmail.com	38	0	7	0	0	0	0	0	0	
23	22	zayn@gmail.com	58	2	7	0	0	0	0	0	0	
24	23	liam@gmail.com	37	0	8	0	0	0	0	0	0	
25	24	cheryl@gmail.com	60	1	9	0	0	0	0	0	0	
26	25	niall@gmail.com	35	0	10	0	0	0	0	0	0	
27	26	gigi@gmail.com	58	1	12	0	0	0	0	0	0	
28	27	perrie@gmail.com	56	0	11	0	0	0	0	0	0	
29	28	justin@gmail.com	38	0	7	0	0	0	0	0	0	
30	29	taylor@gmail.com	58	0	9	0	0	0	0	0	0	

2020 Sheet1

100%

Type here to search

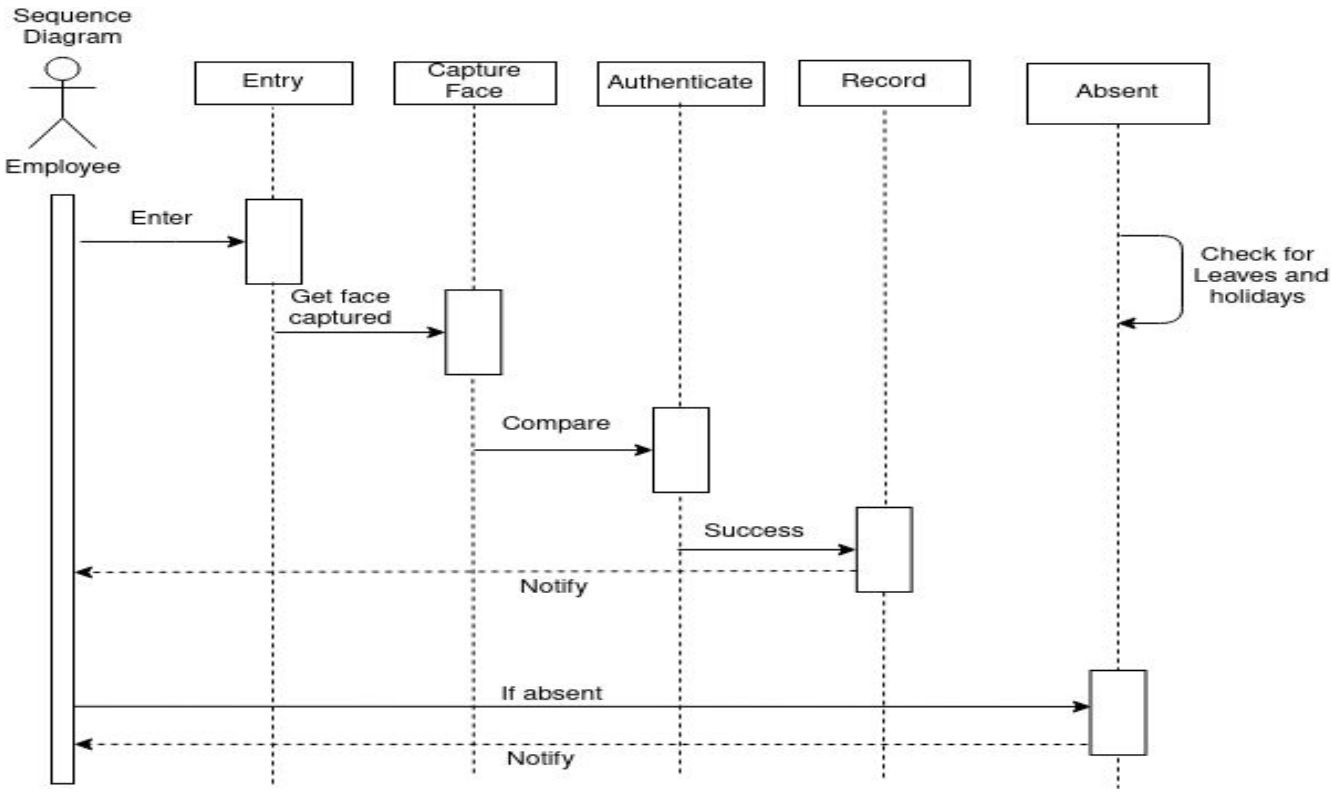
ENG 04:32 18-12-2020

INCORPORATION OF SOFTWARE ENGINEERING:

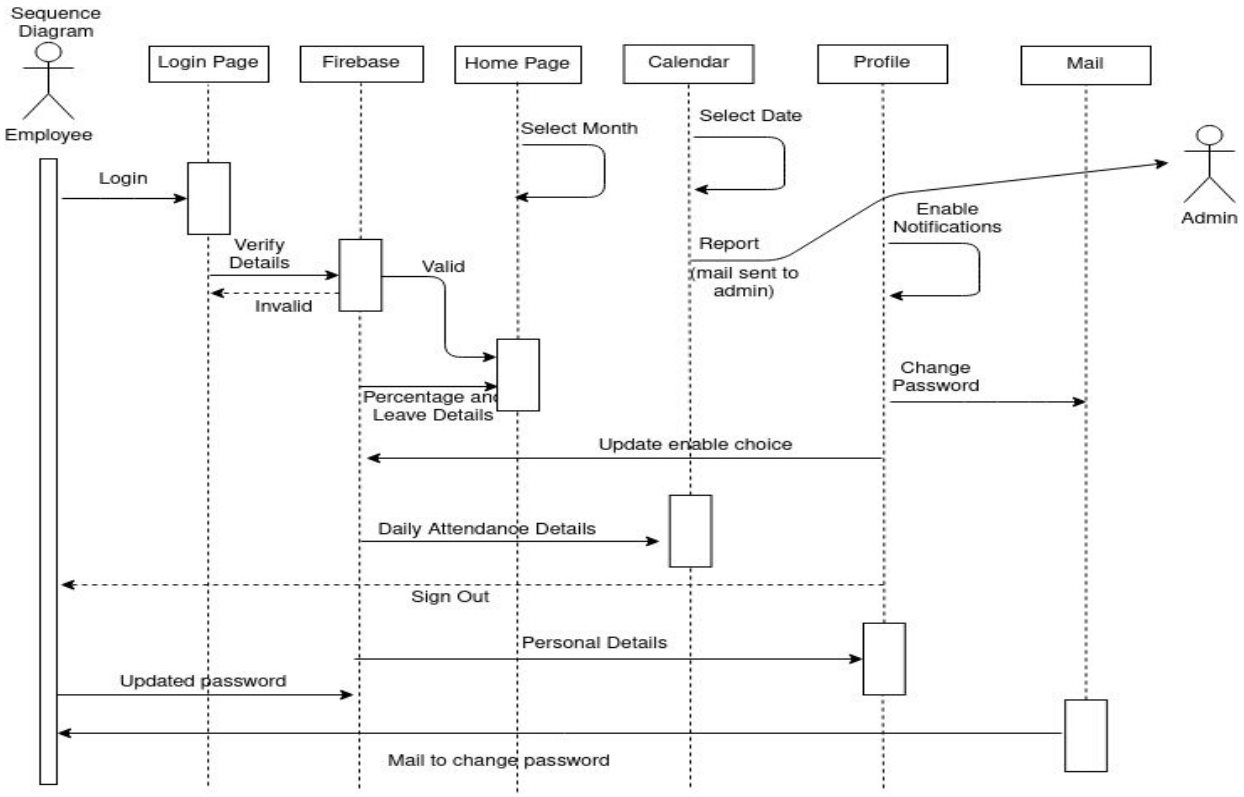
Software Engineering forms the basis of all projects as it entails how to go about developing it. The process model which we have used is the Iterated model or Incremental delivery model. We added requirements as we were developing the project and in each iteration, the design, coding and testing have been done.

1. Sequence diagrams

a. Marking attendance

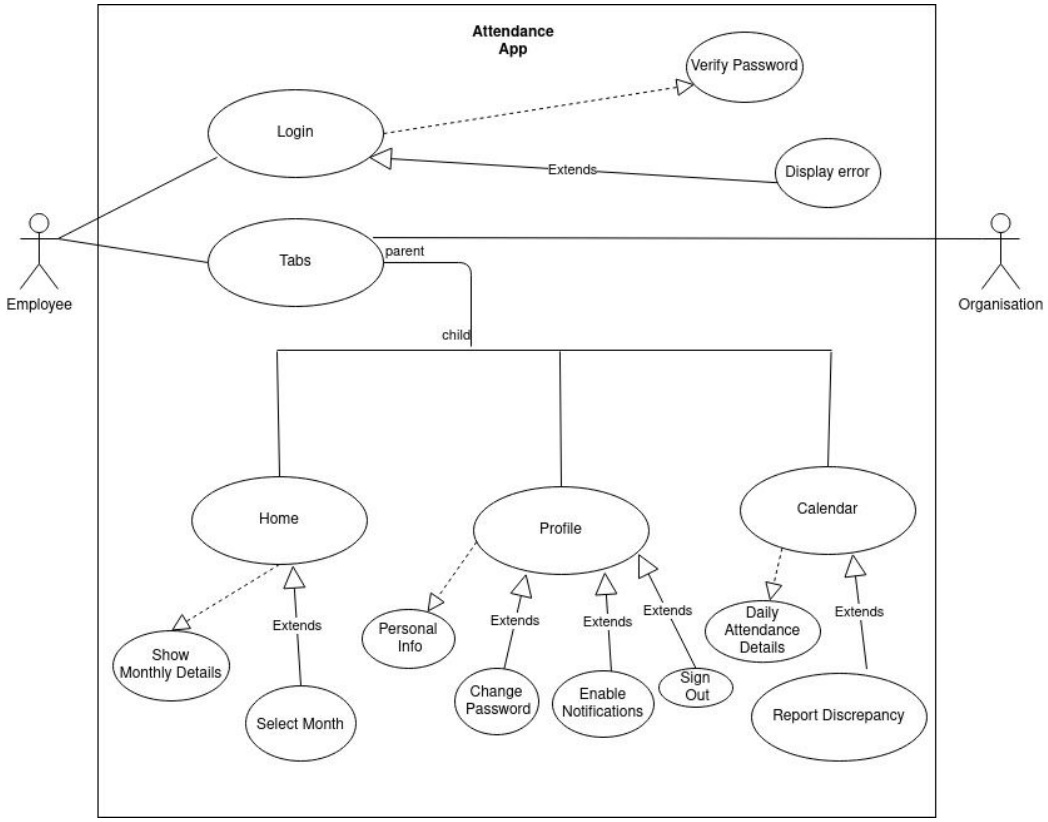


b. App

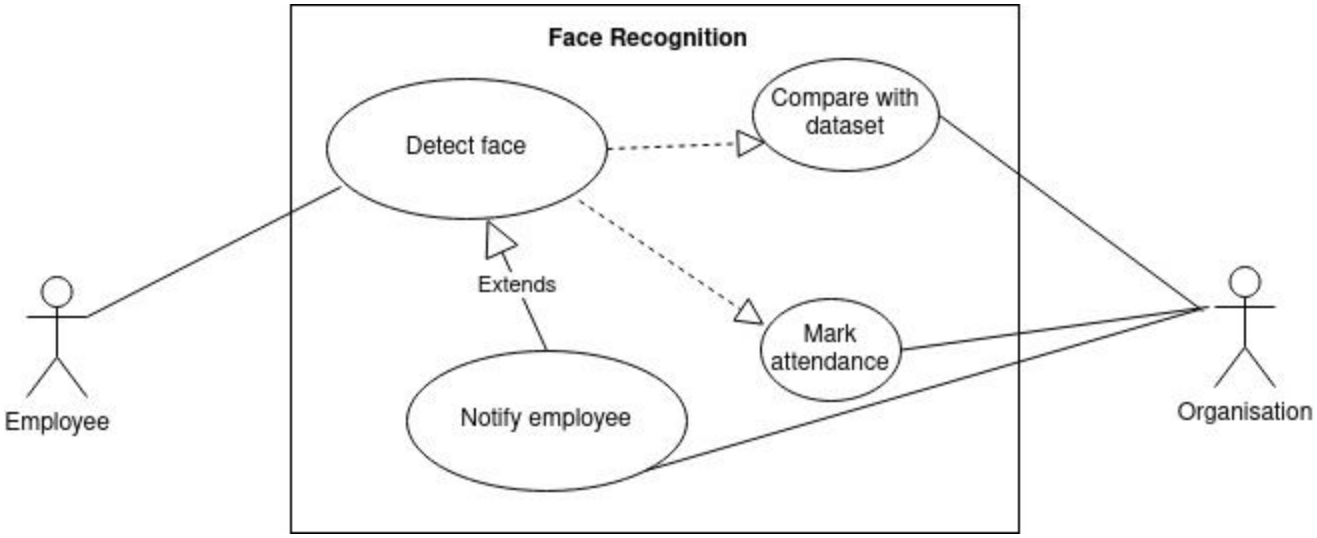

Sequence Diagram
Employee
Login Page
Firebase
Home Page
Calendar
Profile
Mail
Admin
Login
Verify Details
Invalid
Valid
Select Month
Select Date
Report
(mail sent to admin)
Enable Notifications
Change Password
Percentage and Leave Details
Update enable choice
Daily Attendance Details
Sign Out
Personal Details
Updated password
Mail to change password

2. Use Case Diagrams

a. App



b. Attendance



SOFTWARE TOOLS USED

- The whole project is built using **Android Studio**.
- **Android Studio** is a build on JetBrains' IntelliJ IDEA software and official IDE for google android operating system. Android Studio is a designed based operating system, especially for android development. It is accessible for download on Windows, macOS and Linux based operating systems.

  The following features are provided in the current stable version:

  - Gradle-based form support
  - Android-particular refactoring and handy solutions
  - Build up devices to get execution, convenience, adaptation similarity and different issues
  - ProGuard integration and application marking capacities
  - Layout based wizards to make regular Android designs and components
  - A rich format editorial manager that enables users to drag-and-drop UI components, the alternative to see designs on various screen setups
  - Support for building Android Wear applications
  - Worked to help with Google Cloud Platform, empowering mix with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine
  - Android Virtual Device (Emulator) to run and troubleshoot applications in the Android studio.
  - The language used in this application is dart.

DART

Dart is a client-optimized programming language for apps on multiple platforms. It is developed by Google and is used to build mobile, desktop, server, and web applications.

Dart is an object-oriented, class-based, garbage-collected language with C-style syntax. Dart can compile to either native code or JavaScript. It supports interfaces, mixins, abstract classes, reified generics, and type inference.

FLUTTER

Flutter is an open-source UI software development kit created by Google. It is used to develop applications for Android, iOS, Linux, Mac, Windows, Google Fuchsia,and the web from a single codebase.

The first version of Flutter was known as codename "Sky" and ran on the Android operating system. It was unveiled at the 2015 Dart developer summit, with the stated intent of being able to render consistently at 120 frames per second. During the keynote of Google Developer Days in Shanghai, Google announced Flutter Release Preview 2, which is the last big release before Flutter 1.0. On December 4, 2018, Flutter 1.0 was released at the Flutter Live event, denoting the first "stable" version of the Framework. On December 11, 2019, Flutter 1.12 was released at the Flutter Interactive event.

On May 6, 2020, the Dart SDK in version 2.8 and the Flutter in version 1.17.0 were released, where support was added to the Metal API, improving performance on iOS devices (approximately 50%), new Material widgets, and new network tracking.

Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together.

Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

The biggest strength of Python is huge collection of standard library which can be used for the following:

- **Machine Learning**
- **GUI Applications (like Kivy, Tkinter, PyQt etc.)**
- **Web frameworks like Django (used by YouTube, Instagram, Dropbox)**
- **Image processing (like OpenCV, Pillow)**
- **Web scraping (like Scrapy, BeautifulSoup, Selenium)**
- **Test frameworks**
- **Multimedia**
- **Scientific computing**
- **Text processing and many more..**

DELIVERABLES

Our efficient and no-touch automated attendance marking system based upon facial recognition provides:

1. **automated compilation of recorded attendance**
2. **elimination of proxy**
3. **record of leaves taken and remaining**
4. **entire record of the attendance data in app**
5. **downloadable excel sheet that contains the data for all the employees**

CONCLUSION

During the duration of the project, we accomplished our goal of developing a system for marking attendance and previewing the data on an android application. This attendance system will allow organisations to do away with traditional attendance systems. In order to achieve the best results we have also implemented the auto border detection for the document so that the application can automatically detect the borders and edges of the scanned image.

REFERENCES

- https://pypi.org/project/face-recognition/
- https://firebase.google.com/docs
- https://pub.dev/packages/syncfusion_flutter_calendar
- https://flutter.dev/docs