# Project Report

## Attendance Capture System Using Face Recognition

Submitted By -

Abhinav Mathur (N029)

Aryan Sinha (N059)

Paramshree Tinjan (N062)

Nimesh Yadav (N063)

(Second Year Group 2)

# Table of Contents

# Background

- **AIM:** The main objective of this project is to offer system that simplify and automate the process of recording and tracking students' attendance through face recognition technology. It is biometric technology to identify or verify a person from a digital image or surveillance video.

- **Technologies:**

  Build With -

  - Python 3.9

  Module Used -

  - OpenCV Contrib 4.0.1
  - Pillow
  - Numpy
  - Pandas
  - CSV
  - PySimpleGUI

  Face Recognition Algorithms -

  - Haar Cascade
  - LBPH (Local Binary Pattern Histogram)

  Software Used -

  - Pycharm
  - VS Code

- **Hardware Architecture:**
  - PC
  - Web Cam

- **Software Architecture:**
  - Windows 7 or higher
  - Pycharm or VS Code

# System

- **Requirement:**

### Functional Requirements:

- Taking and tracking student attendance by facial recognition in specific time.
- Making a excel file marking the attendance of lecture
- Permitting the lecturer to modify student absent or late
- Determining attendance by the duration the student attended the lecture

### User Requirements:

- Device supporting Windows 7 or higher
- High resolution camera and screen

### Environmental Requirements:

- The program should be secure to any unauthorized changes
- Enough storage should be provided for it function properly
- The model should be accurate and identify with great precision

- **Design and Architecture:**

### Software Design:

'Mark Attendance' button:

Teacher/Administrator starts the process by pressing the 'Mark Attendance' button and entering the lecture/meeting duration.

Camera gives image/video as input to the model

Model uses face recognition algorithm to recognise the person.

If person is recognised and fulfils the required parameters then person gets the option to clock in or clock out.

Each person is clocked in or out based on their choice and a record is kept of total time for which they were present.

When Teacher presses the 'Save Attendance' button, attendance report is generated and saved.

'Add Person' button:

Teacher/Administrator starts the process by pressing the ‘Add Person’ button.

Teacher/Administrator enters the ID and Name for the new person.

Camera captures 100 pictures of the person in real time.

‘Train Images’ button:

Teacher/Administrator starts the process by pressing the ‘Train Images’ button.

Training of the model is done with the new database.

‘View Attendance’ button:

Teacher/Administrator starts the process by pressing the ‘View Attendance’ button.

Teacher/Administrator is taken to the directory where all the attendance reports are saved

Teacher/Administrator can view attendance report by clicking ‘Open’ button.

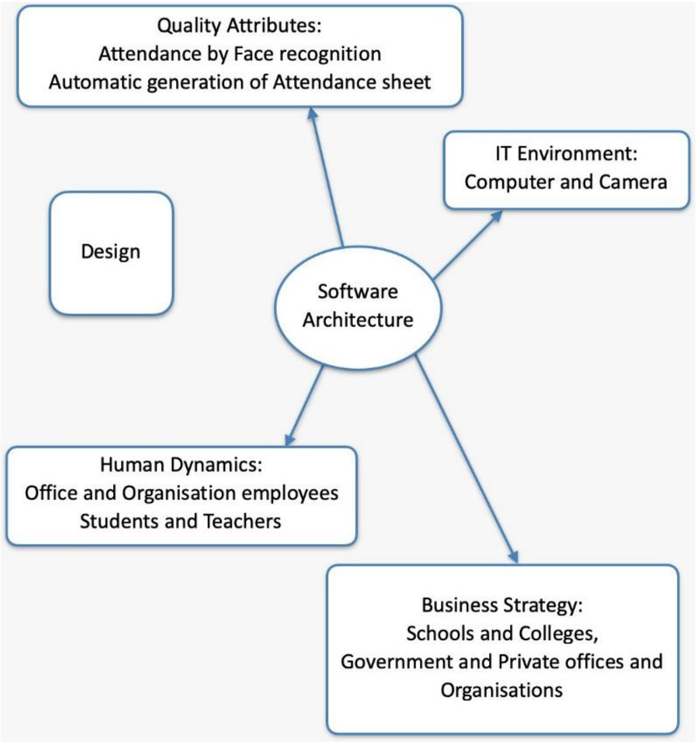
Teacher/Administrator can get back to home by clicking ‘Back’ button.

‘Quit’ button:

Teacher/Administrator starts the process by pressing the ‘Quit’ button.

Window is closed.

## Software Architecture:



- **Implementation:**

Automatically attendance system based on face recognized and also decreases manual work.

Automatic Attendance System Using Face Recognition for lecturers or staffs, implemented the attendance system. In this system, they use the algorithm of face recognition and attendance marking are developed and used.

This system captures two images of a student at two pints in time using a digital camera; one is from the start time of the class and other one is in the end time of the class. Both images will process by this system and will make important role to recognize student using facial recognition. If the student recognized both in the start time and end time classes attendance will be marked for that student.

- **Testing:**

**System testing**: It's used to verify, whether the developed system meets the requirements. Once the entire system has been built then it has to be tested against the Software Requirement Specification and System Specification to check if it delivers the features required. System testing can involve a number of specialist types of test to see if all the functional and non- functional requirements have been met.

**Unit testing:** It's a method of testing that verifies the individual units of source code are working properly. The goal of unit testing is to isolate each part of the program and show that the individual parts are correct.

**Performance Test**

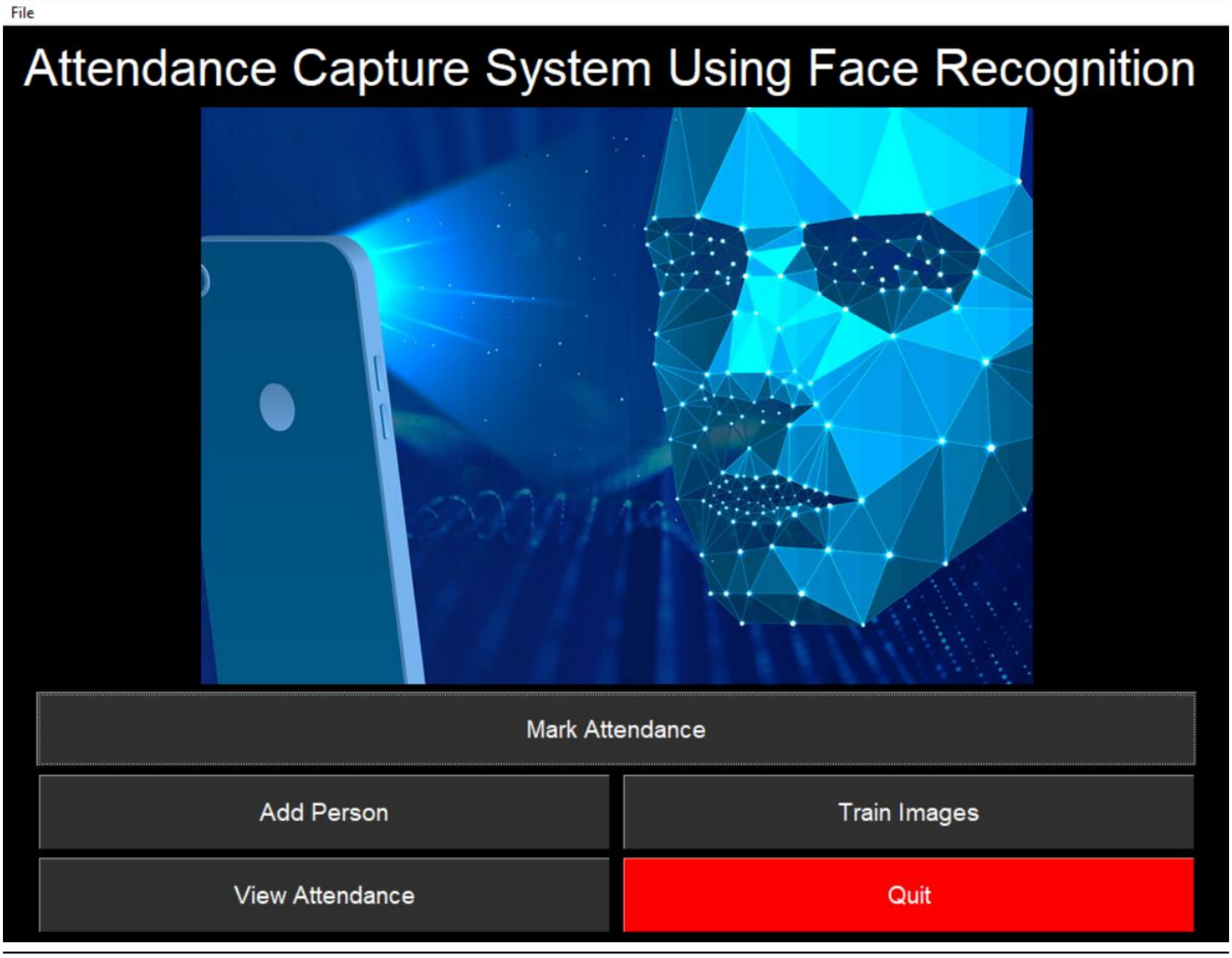
The testing was done to check whether the website is functioning properly or not. When the use sign ups or does the login the details are stored in the database correctly. It shows passwords in the encrypted form when the user enters it. When the user does the login again, he is easily able to access the books.

**User Acceptance Test**

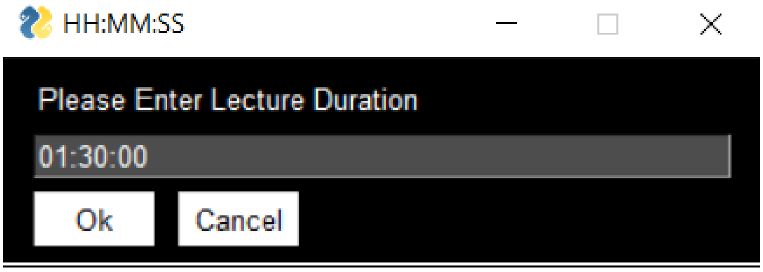
Manual testing is done to test the correctness of all the functionalities by manually entering the data. Test case functionalities for manual testing include:

- Clock IN
- Clock OUT
- View Attendance
- Add Student/Person

- **Graphical User Interface (GUI) Layout:**



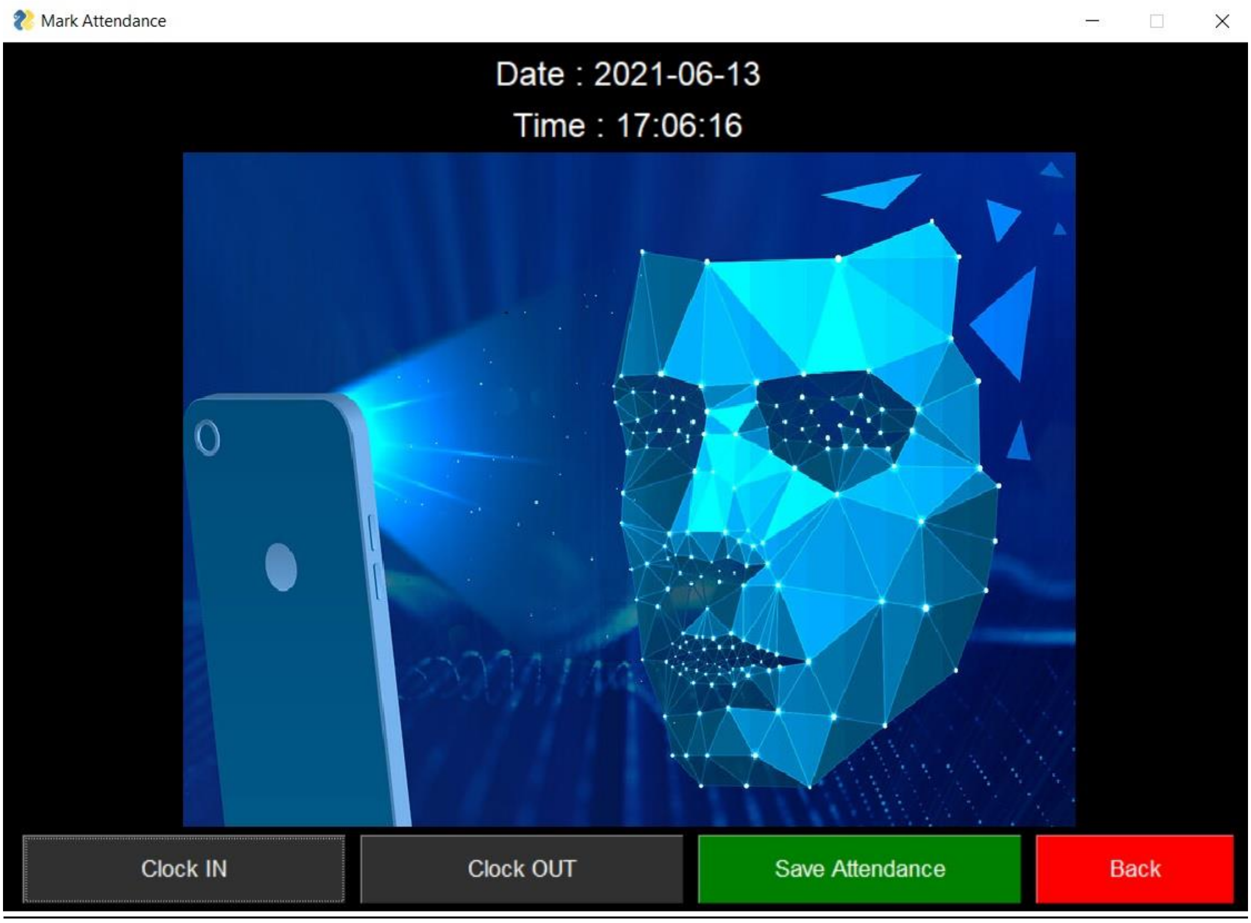
**(Home page)**



Please Enter Lecture Duration

01:30:00

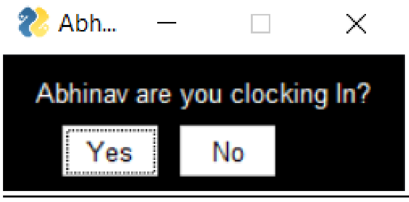
**(Lecture duration popup)**



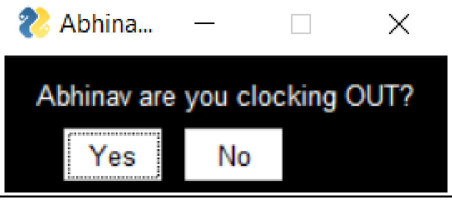
Date : 2021-06-13

Time : 17:06:16

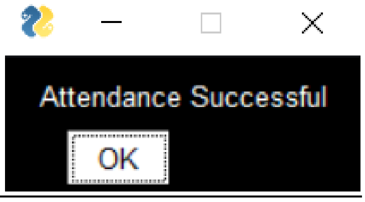
**(Taking Attendance Page)**



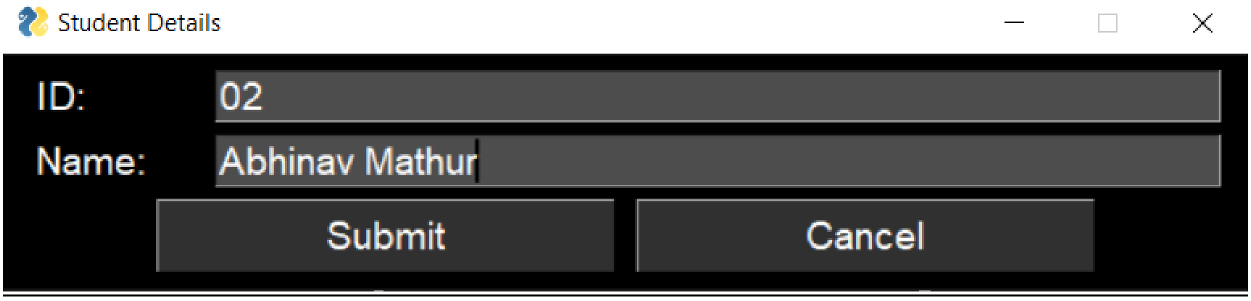
Abhinav are you clocking In?



Abhinav are you clocking OUT?



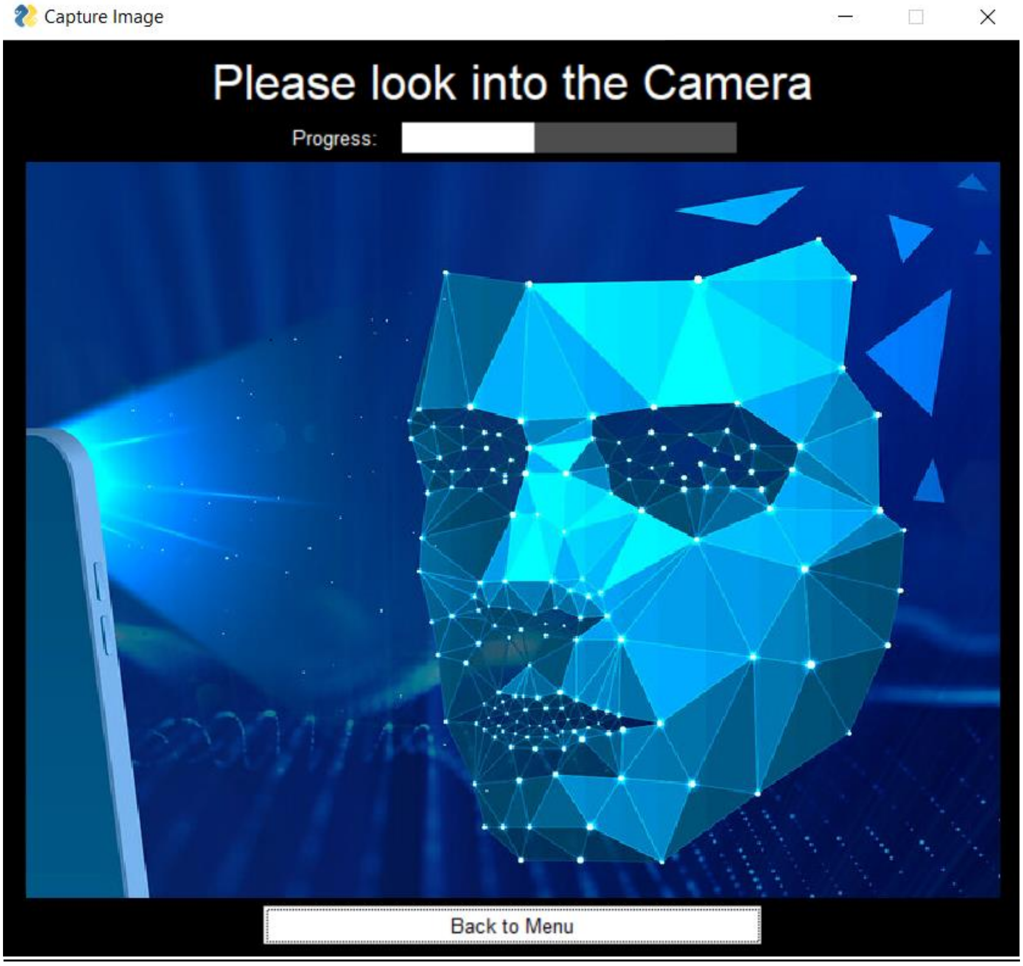
Attendance Successful



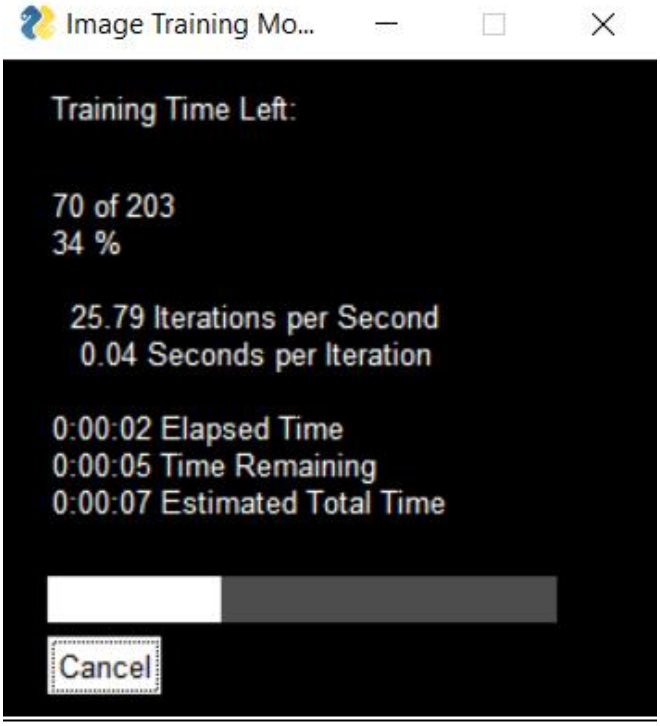
ID: 02

Name: Abhinav Mathur

**(Entering new Student data popup)**



(Taking face images of new student)



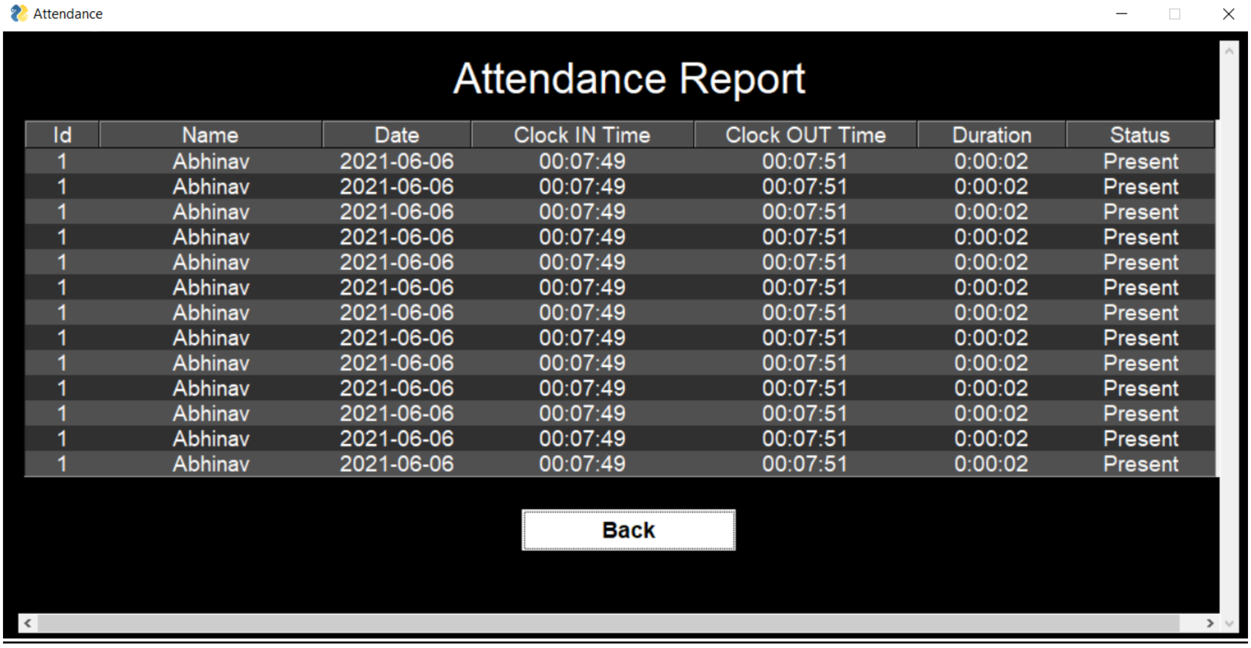
```
Training Time Left:

70 of 203
34 %

  25.79 Iterations per Second
   0.04 Seconds per Iteration

0:00:02 Elapsed Time
0:00:05 Time Remaining
0:00:07 Estimated Total Time
```

(Training the model using all images)



Attendance Report

| Id | Name | Date | Clock IN Time | Clock OUT Time | Duration | Status |
|---|---|---|---|---|---|---|
| 1 | Abhinav | 2021-06-06 | 00:07:49 | 00:07:51 | 0:00:02 | Present |
| 1 | Abhinav | 2021-06-06 | 00:07:49 | 00:07:51 | 0:00:02 | Present |
| 1 | Abhinav | 2021-06-06 | 00:07:49 | 00:07:51 | 0:00:02 | Present |
| 1 | Abhinav | 2021-06-06 | 00:07:49 | 00:07:51 | 0:00:02 | Present |
| 1 | Abhinav | 2021-06-06 | 00:07:49 | 00:07:51 | 0:00:02 | Present |
| 1 | Abhinav | 2021-06-06 | 00:07:49 | 00:07:51 | 0:00:02 | Present |
| 1 | Abhinav | 2021-06-06 | 00:07:49 | 00:07:51 | 0:00:02 | Present |
| 1 | Abhinav | 2021-06-06 | 00:07:49 | 00:07:51 | 0:00:02 | Present |
| 1 | Abhinav | 2021-06-06 | 00:07:49 | 00:07:51 | 0:00:02 | Present |
| 1 | Abhinav | 2021-06-06 | 00:07:49 | 00:07:51 | 0:00:02 | Present |
| 1 | Abhinav | 2021-06-06 | 00:07:49 | 00:07:51 | 0:00:02 | Present |
| 1 | Abhinav | 2021-06-06 | 00:07:49 | 00:07:51 | 0:00:02 | Present |
| 1 | Abhinav | 2021-06-06 | 00:07:49 | 00:07:51 | 0:00:02 | Present |

**(Viewing csv attendance)**

## ➢ <u>Customer Testing:</u>

Manual Testing will be done to ensure the correctness of various parts of the code using test cases generated by the tester. Through the above shown screenshots it can be concluded that the implementation is done correctly

## ➢ <u>Evaluation:</u>

# Conclusion

**Attendance is one of the most important aspect for all the organisations, schools and offices. Our project makes attendance capturing easy, efficient and faster.**

**Our project also makes the attendance system secure since attendance is marked using face recognition algorithm and all the users will be allowed specific ID.**

**The project also allows adding a new user by taking on the spot 100 photographs of the new user, allowing a new user ID with the user's credentials as given by the user, and training of the new model for better prediction.**

**The project automatically keeps track of time and marks attendance with respect it hence it keeps record of the total time for which each employee/student/user was present.**

**This project generates an attendance report and automatically saves it and thus makes the process of taking attendance a lot easier and faster than manually marking attendance.**

# Further Development or Research

**Our Project can be further developed as an app by providing compatibility for various operating systems.**

**Adding feature for storing time tables for each day of the week so that teacher/administrator does not need to enter lecture duration and subject for lecture/details of meeting.**

**Also, the above feature would help making an enhanced attendance system with not only attendance report of each lecture but then could also track attendance of each person/student/employee.**

**This project could also be developed as a more secure attendee system by denying permission to entry to meeting/lecture if person is not recognised and giving an alert to the administrator/teacher asking if the person is authorised or not hence administrator/teacher would also have an option to have the person as a guest/observer in the meeting and hence the attendance report would be generated accordingly.**

**One of the next developments can be to make the program loading time significantly less and making the program to look faster and with smooth animations.**

# References and Appendix

- https://www.ukessays.com/essays/education/recognition-attendance-system-6424.php
- https://pypi.org/project/face-recognition/
- https://pypi.org/project/opencv-python/
- https://www.youtube.com/watch?v=sz25xxF_AVE
- https://analyticsindiamag.com/a-complete-guide-on-building-a-face-attendance-system/
- https://docs.python.org/3/library/datetime.html
- https://github.com/opencv/opencv/tree/master/data/haarcascades
- https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html
- https://www.geeksforgeeks.org/python-haar-cascades-for-object-detection/