PROVIDENCE
COLLEGE OF ENGINEERING
&
SCHOOL OF BUSINESS

ATTENDANCE MANAGEMENT SYSTEM USING FACE RECOGNITION

PROJECT REPORT

Submitted by

LIYA ELZA THOMAS PRC19CS034

PREYAS K A PRC19CS036

ROSHNI SUSAN RAJU PRC19CS040

SHIBIN S P PRC19CS043

JULY 2022

PROVIDENCE COLLEGE OF ENGINEERING

CSE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Attendance Management System Using Face Recognition

MINI PROJECT REPORT

Submitted by

LIYA ELZA THOMAS	**PRC19CS034**
PREYAS K A	**PRC19CS036**
ROSHNI SUSAN RAJU	**PRC19CS040**
SHIBIN S P	**PRC19CS043**

To

APJ Abdul Kalam Technological University

in partial fulfillment of the requirements for the award of B. Tech Degree in Computer Science and Engineering

Department of Computer Science and Engineering

Providence College of Engineering, Chengannur July 2022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PROVIDENCE COLLEGE OF ENGINEERING, CHENGANNUR

CERTIFICATE

Certified that this report entitled '*Attendance Management System Using Face Recognition*' is the report of Mini Project completed by the following students during **2021-2022** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering.

LIYA ELZA THOMAS	**PRC19CS034**
PREYAS K A	**PRC19CS036**
ROSHNI SUSAN RAJU	**PRC19CS040**
SHIBIN S P	**PRC19CS043**

Mr. Girish K K (Project Supervisor)
Assistant Professor, Department of Computer Science & Engineering (CSE)

Ms. S Prasanthi (Mini Project Assistant Coordinator)
Assistant Professor, Department of Computer Science & Engineering (CSE)

Ms. Jeni Moni
Mini Project Coordinator-CSE
Dept. of Computer Science & Engineering
Providence College of Engineering

Dr. Bibin Vincent
Head of the Department
Dept.of Computer Science & Engineering
Providence College of Engineering

PROVIDENCE
COLLEGE OF ENGINEERING

DECLARATION

We, hereby declare that, this project report '*Attendance Management System Using Face Recognition*' entitled is the bonafide work of ours carried out under the supervision of Mr. Girish K K, Assistant Professor, Department of Computer Science and Engineering. We declare that, to the best of our knowledge, the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion to any other candidate. The content of this report is not being presented by any other student to this or any other University for the award of a degree.

Sl. No	Name of Student	Roll Number	Signature
1	LIYA ELZA THOMAS	PRC19CS034	
2	PREYAS K A	PRC19CS036	
3	ROSHNI SUSAN RAJU	PRC19CS040	
4	SHIBIN S P	PRC19CS043	

Mr. Girish K K (Project Supervisor)

Dr. Bibin Vincent

Head of the Department - Computer Science and Engineering

Providence College of Engineering

Date: 25/07/2022

ACKNOWLEDGEMENTS

We take this opportunity to express our deep sense of gratitude and sincere thanks to all who helped us to complete the mini project successfully.

We are deeply indebted to our Project Supervisor **Mr. Girish K K**, Assistant Professor for his excellent guidance, positive criticism, and valuable comments. We record our appreciation and sincere thanks to our Departmental Mini Project Coordinators **Ms. Jeni Moni**, Assistant Professor and **Ms. S Prasanthi**, Assistant Professor for their overall coordination and timely guidelines.

We are also greatly thankful to our Head of Department **Dr. Bibin Vincent**, Associate Professor for his continuous support.

Finally, we thank our parents, family members and friends who directly and indirectly contributed to the successful completion of our mini project.

Liya Elza Thomas
Preyas K A
Roshini Susan Raju
Shibin S P

Date: 25/07/2022

ABSTRACT

PRO MARK, a facial recognition-based attendance management system, is used to automatically take attendance. The system accepts images as input, and it generates a .csv file with details about the students' attendance. The method of taking attendance is capturing an image from the class and separates the faces and non faces from image and finds match with the detected face using the OpenCV library from the database which was created with the details of students. The database used is MongoDB and the algorithm used is face recognition. Face detection is carried out and compared based on the data's accuracy utilising the face_recognition module. The two basic approaches used by the system are face detection and face recognition. This system can be implemented using a Web application. The 'PRO MARK' (web application) is developed using the Flask micro web framework.

Keywords: Face detection, Face recognition, MongoDB, Flask, OpenCV

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Expansion
ANN	Artificial Neural Network
APK	Android Package
BoF	Bag of Features
BSA	Backtracking Search Algorithm
CCD	Charged Couple Device
CNNF	Convolution Neural Network Features
COCOMO	Constructive Cost Model
CSS	Cascading Style Sheets
GB	Giga Bytes
GBSA	Genetic Backtracking Search Algorithm
GLCM	Gray Level Co-occurrence Matrix
GPU	Graphics Processing Unit
HD	High Definition
HSV	Hue Saturation Value
HTML	Hyper Text Markup Language
IDE	Integrated Development Environment
IPS	In-Plane Switching
KLOC	Kilo Lines of Code
KNN	K-Nearest Neighbors
LCD	Liquid Crystal Display
LOC	Lines of Code
ML	Machine Learning
OS	Operating System
PSO	Particle Swarm Optimization
RAM	Random Access Memory
RBF	Radial Basis Function
REQ	Requirement
RGB	Red Green Blue
SVM	Support Vector Machine
UML	Unified Modeling Language
VEGA	Vector Evaluation Genetic Algorithm

CHAPTER 1

INTRODUCTION

1.1 Background

Traditionally, student attendances are taken by manually calling out the name of each student which is very time consuming. This method of taking attendance is difficult, and there is a chance of proxy attendance. In order to avoid this, a face recognition-based attendance marking system is introduced.

Facial recognition is a technique used to identify or confirm a person's identity using their face. Facial recognition technology can identify people in real time as well as in still photos and movies. Face recognition is a subtype of biometric security. Additional biometric software kinds include voice, fingerprint, and ocular retina or iris identification. The majority of the technology's applications still fall under security and law enforcement, despite increased interest in employing it in other fields. Despite the fact that facial recognition systems might vary, they generally work as follows

- Facial recognition
- Analyzing the face
- Converting the images to data
- Finds a matching face from database

The purpose of the project is to construct a facial recognition-based attendance monitoring system. Here, a person's face will be taken into account while recording attendance. We proposed a face recognition system in which each student in the class would be photographed and their information would be kept on a server. The teacher will take some few pictures of the classroom, and the system will identify the faces and verify if each student is present or not.

1.2 Existing System

The existing system for marking attendance is the conventional way of calling out each student individually. It is a very time consuming and tedious process as there are chances for making human errors. Even students can call a proxy while taking the attendance. Students can easily edit the attendance sheet and mark attendance.

- Traditional methods are less accurate and slower.
- Chances for fake attendance marking.
- Real time monitoring is not available
- Difficult to manage the attendance records in manual registers.

1.3 Problem Statement

There is a need for a system that can act as an aid to mark class room attendances in a quick, cost effective, portable, and accurate manner.

1.4 Objectives

The following are some of the goals of our proposed system:

- To ease the process of classroom attendance marking
- To facilitate enough time for the course coordinators to cover the course
- within the stipulated time
- To make attendance marking fault free and error free

1.5 Scope

Despite the fact that this technique is used all over the world, the scope of this study is restricted to classes at the Providence College of Engineering in Chengannur, Kerala, with a maximum of 60 students. The professors in the department of computer science and engineering at Providence College of Engineering in Chengannur, Kerala, will be the system's users.

CHAPTER 2

LITERATURE REVIEW

We have reviewed a wide range of previously published studies on facial recognition-based attendance control. The 8 papers that are the most important are summarized here.

Raghuwanshi et al. [1] proposed the system to take attendance automatically using a face detection method. An excel sheet listing the attendance of the pupils present in class serves as the input for the system that detects faces, and it serves as the output. The paper talks about taking attendance from the video taken from the class and separates the faces from the video and finds a match with the detected face from the database which created with the details of students. Using the Principle Component Analysis (PCA) and Linear Discriminant (LDA) techniques, face detection is carried out and compared based on the data's accuracy. We can get the face detection by using either video recording or by capturing the images with the camera which is placed inside the class covering the whole area of the class. The system involves mainly two methods which use face detection and face recognition. Good face detection is required for better face recognition. In this study, the system made use of many approaches, including database creation, video recording, face identification, face recognition, and attendance tracking on excel sheets. The training set's database houses the photographs of the enrolled students. About 400 images of 40 individuals taken as the dataset. Taking video by using the good quality camera which is efficient for detecting the face and it stored on the connected devices. After taking the video and reading the frames it sending for the face recognition by using the computer vision. The "CascadeObjectDetector" command is a built-in MATLAB function for Viola-Jones method detection. After face recognition is finished, the attendance register-taking procedure is used. If the detected faces have been recognized, then the attendance marked on the excel sheet. We can easily export or import data to either of them using ‘xlswrite’ and 'xlsread' commands.

Trinos et al. [2] proposed an attendance monitoring system using smart face recognition which can be applied in real time classes. To check the attendance in real time a software is developed by using image processing technology applied in facial

recognition. It is very hard to mark attendance manually also to keep the records in order to avoid the pain stake of faculties this software was introduced. Attendance was marking without any student interaction; this system identifies the student inside the classroom and records their presence. This application helps the client to scan the face of students according to the information and training given. The recognized image was converted to machine language and the converted image was compared to the data that was stored in local databases. Finally monitor the student's attendance by generate an output of student records required.

Rani et al. [3] proposed a student attendance management system using face recognition which is automated. For the beginning and ending of a class attendance is an important part of evaluation the student's presence. Recognition of a student's face for taking attendance using face recognition technology based on high-definition monitor video and other information technology. This system helps to reduce the human errors which occurs during the time of marking of attendance manually and keeping the records and make it more efficient. By using high quality webcam, the system can capture the face of the student with high quality image even if there is poor light in the classroom.

Smitha et al. [4] proposed an attendance management system using face recognition. This paper tells how the face recognition-based attendance management system helps many institutes as the traditional method of attendance marking is very time consuming. Verification and face recognition are the two basic kinds of face recognition systems. The database generation, face detection, face recognition, and attendance marking are the system's four steps. The dataset will contain the essential student data as well as images of the students. The Haar-Cascade classifier and Local Binary Histogram algorithm are used to recognize faces. Faces will be identified from classroom live streaming video during each lesson, and the identified faces will be compared to the photos contained in the dataset. The attendance will be recorded for the specific student if it matches. A list of absentees will be forwarded to the appropriate faculty member managing that class at the end of the class.

Bharadwaj et al. [5] proposed attendance management using facial recognition. Calling names of each student in each hour and keeping a record of the attendance is quite time-consuming, with a potential for proxy attendance. The main objective of the following system is to accurately identify students' faces by use of Principle Component Analysis

(PCA). The suggested system includes capabilities for face detection, face feature extraction, face feature detection, and face feature analysis. The face recognition algorithm and the Local Binary Pattern Histogram (LBP) method, a fundamental technique used to detect faces from the front side, are the algorithms that are employed. This indicates that the system enters pupils' names, unique IDs, and faces into the database. The system will then assign each recognized face in the database a status of attendance.

Sai et al. [6] proposed the system to taking attendance using the real time face detection method for the large number of students in the classrooms In this paper the HaarCascade classifier used to determine the characteristics of the face as well as the Local binary pattern histogram for face recognition which implemented in python and OpenCV library, user interface is used with tkinter GUI interface. The approach recommended in this project is to track attendance using face recognition. Faces are sensed in picture format by the camera and video feeds. The database will link the recognised faces, and an Excel spreadsheet will be used to track attendance. Machine learning may be trained to recognized things in an image format utilizing face identification using the Haar-cascade Classifier technique. It is necessary to condition a Haar cascade with both positive and negative pictures. The objective is to determine which set of features best describes a face. The item is there in the positive image, while the face is absent from the negative. With the use of student facial photos, train the algorithm. Additionally, each image needs its own individual identifying number to be recognized..

Kainz et al. [7] proposed a system to recognize faces through an attendance monitoring system in the video. Face recognition in the live video stream is done using the HOG method and the Viola-Jones algorithm. The convolutional neural network (CNN) is used to identify the student. The attendance list of current pupils may be noted after face identification is complete. The technology then gives the administrator a visual confirmation for identified faces.

Dev and Patnaik. [8] proposed a student attendance system using face recognition. The most effective image processing application that plays a significant part in technology is face recognition. The system's expansion aims to digitally replace the outdated method of collecting attendance by calling names and keeping handwritten records. This tactic saves time since manual recording of attendance data is simple to manipulate. There is no installation required, and the system is economical.

CHAPTER 3

SYSTEM ANALYSIS

The study of various functional requirements, non-functional requires, and design constraints of the system all are parts of the system analysis. This chapter provides a detailed system analysis of attendance management systems using face recognition.

3.1 Expected System Requirements

The proposed attendance management system using face recognition is expecting the following requirements.

REQ 1: Develop a user-friendly Web application that can capture attendance of a 60 members classroom.

Description: The faculty members should have a user-friendly application inside their hand-held mobile device which can capture the physical presence of all the students registered in a batch.

Input: Image captured via camera of the mobile device.

Output: List of students present and the corresponding time stamps in a .csv file.

REQ 2: The Web application should have provisions for administrator profiles.

Description: Administrator should have provision for creating separate profiles.

Input: The photos of students that are captured during class time by the faculty.

Output: The data is accepted and verified.

REQ 3: Marking attendance using the verified outputs

Description: The Web application will mark the attendance by analysing the input with the given data that is stored in a database.

Input: Student images which are verified by the system

Output: Attendance marked.

REQ 4: Display the attendance using excel sheet

Description: Display the attendance using the excel sheet based on the recognition and verification of the face of the students.

Input: Attendance marked details

Output: Excel sheet which gives the attendance of the class.

3.2 Feasibility Analysis

The feasibility analysis for the mentioned requirements is for the evaluation of project success and discuss the possibility for building such a system . The technical, operational and economic feasibility analysis are discussed below.

3.2.1 Technical feasibility

The technical feasibility analysis is used to analyze whether the system can be built within the existing technology. For the attendance management system using face recognition is easy to use a Web application in any devices. The mobile camera is used to take photos of students which are used as the main input of the system.

3.2.2 Operational feasibility

The proposed system's ability to address the user's issue is evaluated using the operational feasibility study. The web application was created using Flask and is compatible with all computers and smartphones.

3.2.3 Economic feasibility

The economic feasibility analysis is used to analyze whether the system can be built within the given budget. The system is more economical as only a smart phone or a computer is required for the attendance management purpose. So, this project will be economically feasible if we develop it on a large scale.

3.3 Software Requirements

- Visual Studio Code
- Python IDE
- MongoDB
- Flask
- OpenCV
- HTML
- CSS

3.4 Cost Estimation

3.4.1 Software Cost Estimation

For the cost estimation, the COCOMO Model (Constructive Cost Estimation Model) is used.

Table 3.1 COCOMO model coefficients

Software Project	a	b	c	d
Organic	2.4	1.05	2.5	0.38
Semi-detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

Software Project Category: Semi Detached

Estimated Lines of Code(LOC): 1500 LOC = 1.5 KLOC

Effort applied = $a*(KLOC)^b$ [Person-Months] = $3(1.5)^{1.12}$ = 4.72 PM

Development Time = $c*(Effort)^d$ = $2.5(4.72)^{0.35}$ = 4.30 months/131 days

3.5 Project Schedule Using Gantt Chart

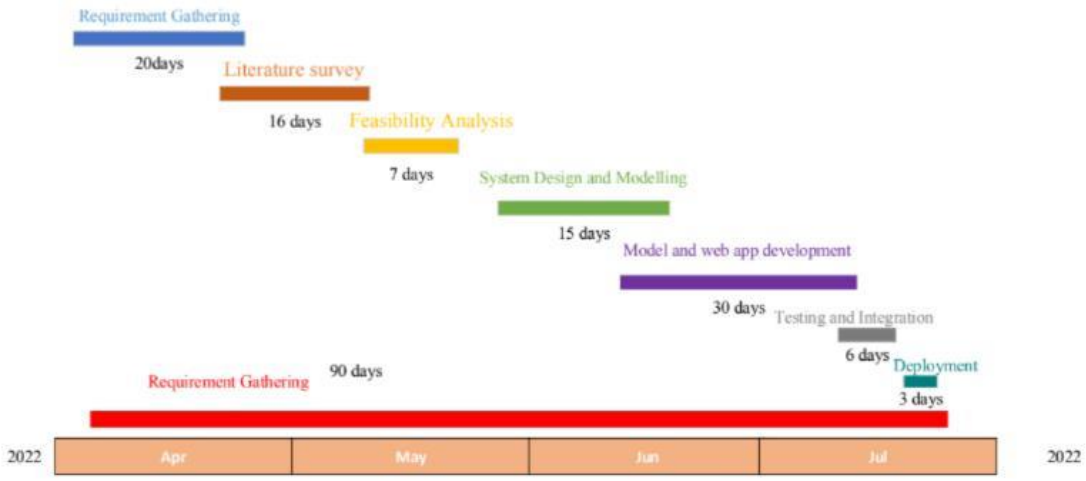

Figure 3.2 Gantt Chart

The schedule of the project is displayed on the Gantt chart. The project's start and end dates are displayed in a Gantt chart. The project phases will be finished in accordance with the timetable shown in the aforementioned Gantt chart, Figure 3.2. The project began on 29/04/2022, and it ends on 25/07/2022.

CHAPTER 4

METHODOLOGY

4.1 Proposed System

The described system comprises of a Web application which marks the attendance of the students by capturing the image of the class. The system's camera is utilized to identify the students' faces in the image that was captured. Each student's attendance will be noted in the.csv file following the face recognition.4.1.1 Web application

When first the app launches, the student is presented with a display asking the user id and password to login to capture the image of the class to monitor the attendance. If the user has not yet logged in ,then there is a sign-up button. After selecting the sign-up button name, email id, user id and password should be entered and thus creating a profile. The next screen will be presented with a button to capture the image. Then the faces of the students will be detected and attendance will be marked.

4.1.2 Database

MongoDB is the system's database of choice. The system/cloud-installed database is accessed via the pymongo module. When the application first launches, the database is configured. It keeps track of the user's username, password, and subject. The password is hashed and saved using the SHA-256 algorithm.

4.1.3 Face Recognition

This is accomplished using the Face recognition package for Python. The photos from the sample directory are initially gathered by the system. The photos that are acquired are encoded and kept within a list. Following a check and match with the list of encoded pictures, the collected photos are then identified.

4.1.4 Image Dataset

Dataset is a vital component and it is critical to choose the correct input data for our purposes. In this the dataset will contain the pictures of each student taken from all sides.

4.2 Advantages of Proposed System

The following are the benefits of the suggested system:

- More efficient.
- Works in real-time.
- No proxy attendance.
- More user-friendly.
- It can be used on any device globally.

CHAPTER 5

SYSTEM DESIGN

The recommended system model is represented using a variety of system modelling techniques, such as a flowchart diagram, use case diagram, an activity diagram, and a sequence diagram.

5.1 Flowchart

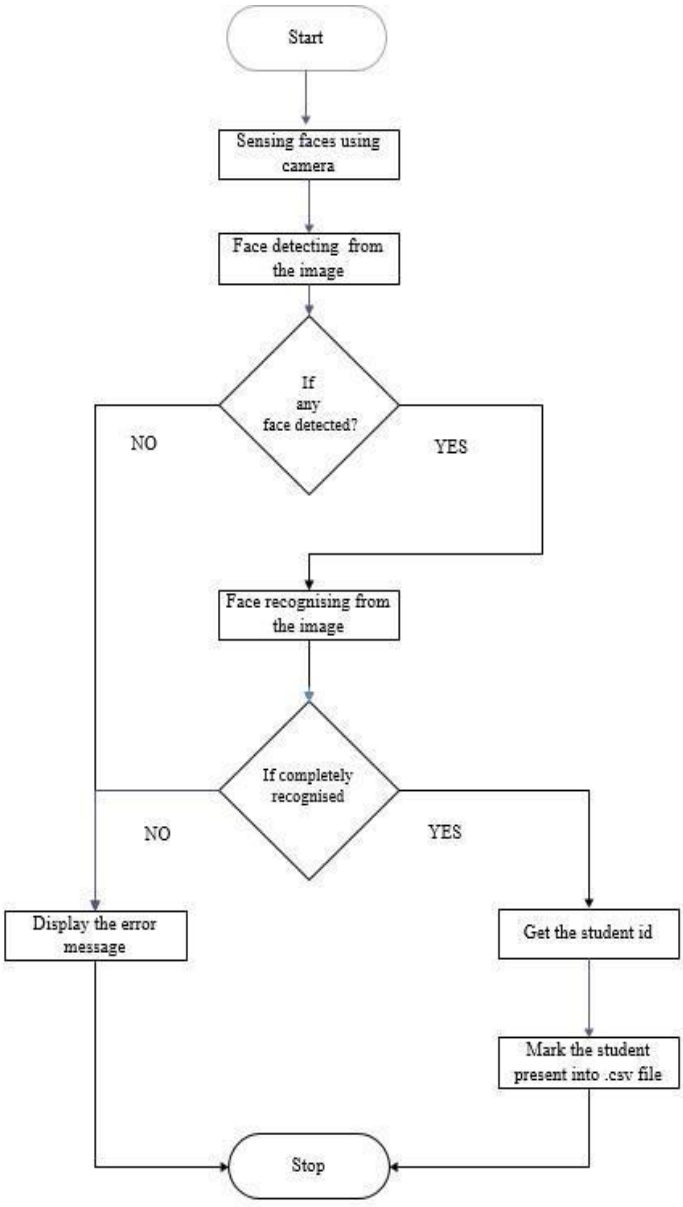

Figure 5.1 Flowchart

Figure 5.1 illustrates the flow chart of an attendance management system.

Description:

Figure 5.1 illustrates the flow chart of an attendance management system using face recognition. Faces will be recognised from the image once it has been captured. If and only if the face was recognised in the taken image, the image will be compared to the built-in database. The built database will be used to identify the identified face. Get the student ID and record the student's attendance into a.csv file if the faces can be fully recognised. If the faces are not totally and correctly recognised, an error notice will be displayed.

5.2 Use Case Diagram

A use case diagram is a diagram in the Unified Modelling Language (UML) that depicts the link between different use cases and actors. There are two actors in the system the faculty and admin.

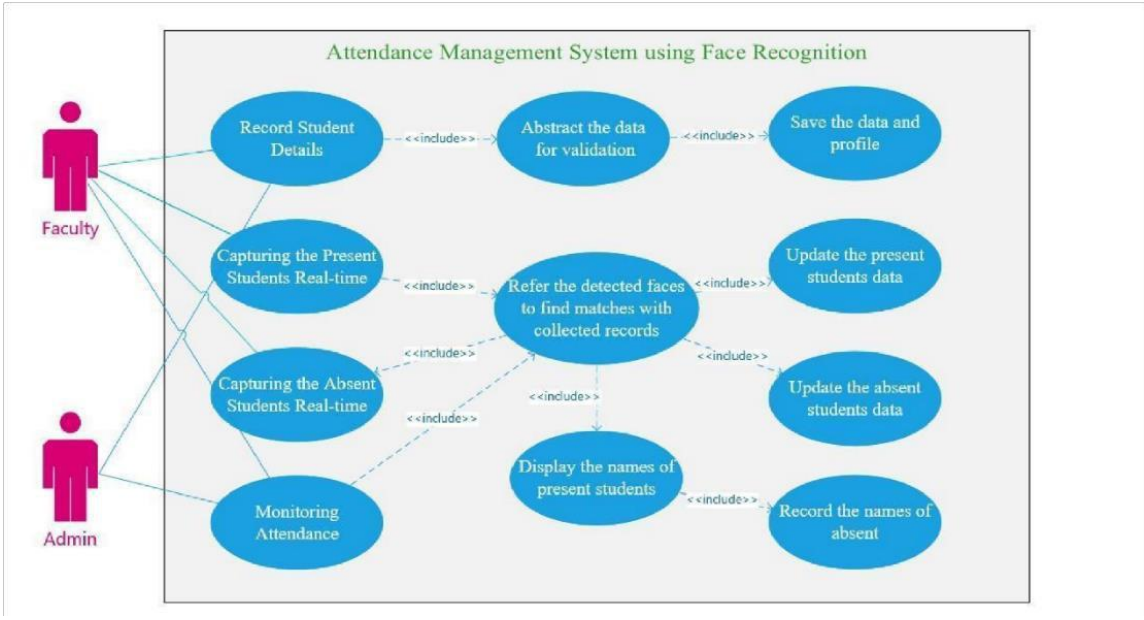

Figure 5.2 Use Case Diagram

Description:

Figure 5.2 the Use Case Diagram shows the application of the system in attendance management. The end user of this system will be the faculty(user). The user can use the web application to capture the attendance and to get data into the excel as .csv as output.

5.3 Activity Diagram

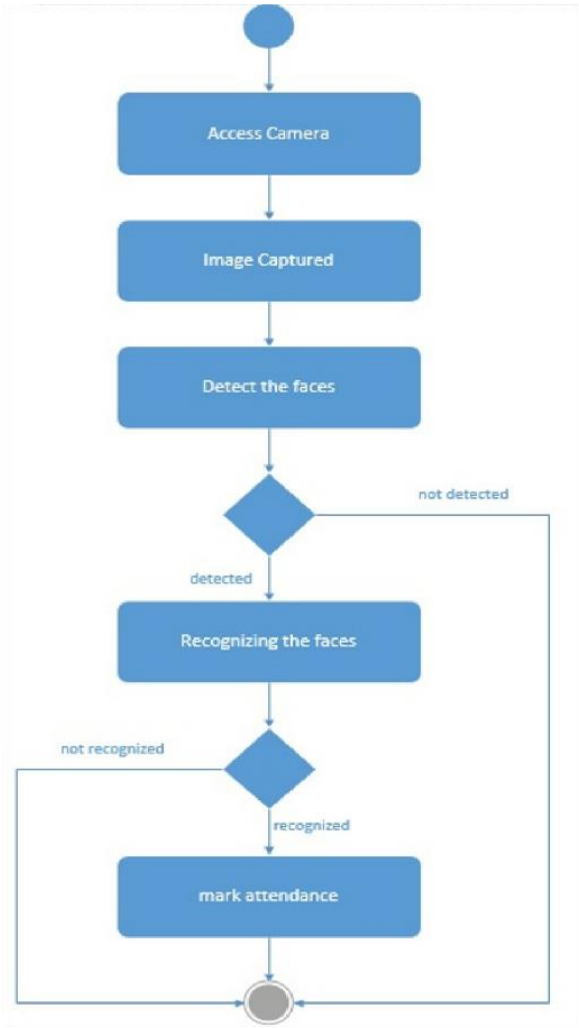

Figure 5.3 Activity Diagram

Description:

The Web application's execution order is depicted in Figure 5.3. The device camera is initially assessed when the app is launched. Then the required image is captured. If the application is unable to detect the faces, then the execution ends with an error message shown. Otherwise, the application will recognize the faces of each student and the attendance will be marked for them. If the application is unable to recognize the detected faces, then the execution ends with an error message shown.

5.4 Sequence Diagram

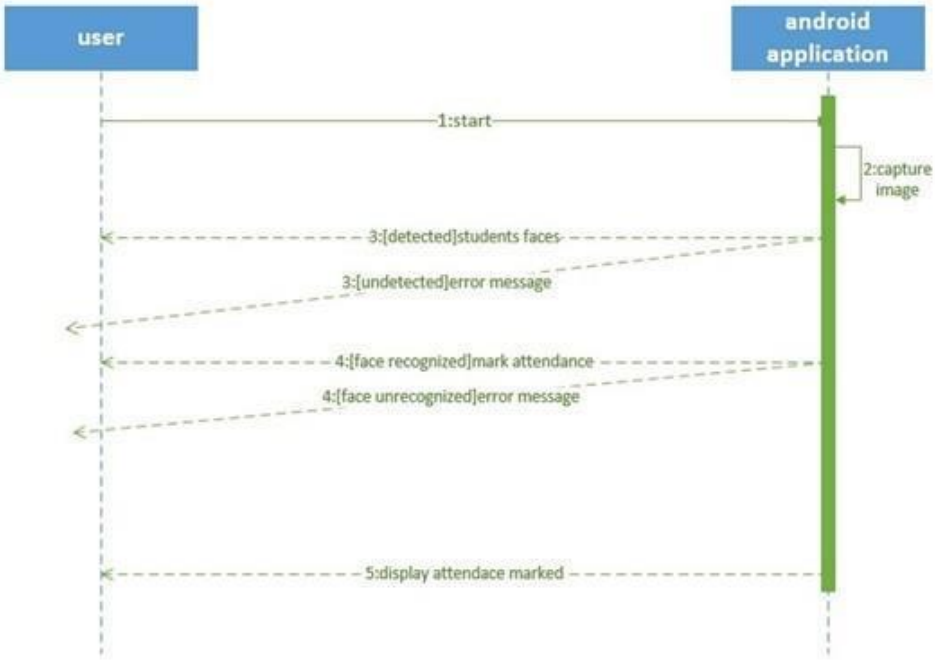

Figure 5.4 Sequence Diagram

Description:

The interaction between the user and the application is depicted in Figure 5.4 by a sequence diagram. The user first opens the application to capture the image of the class. The application will respond by recording the attendance in the .csv file.

5.5 UI Design

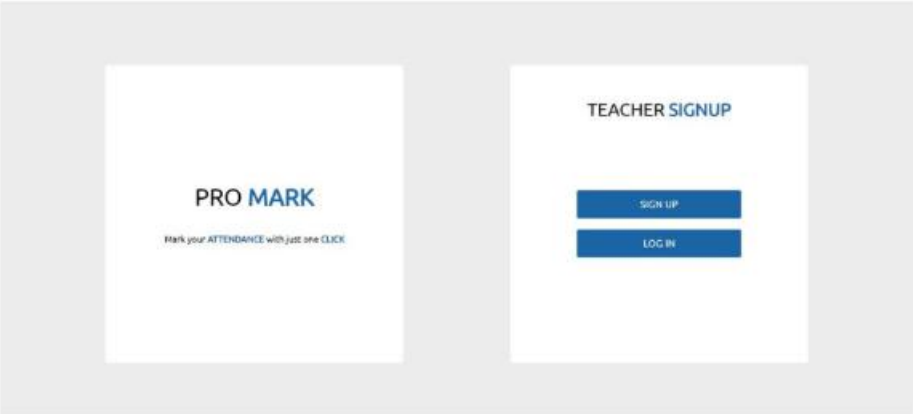

Figure 5.5 Home page

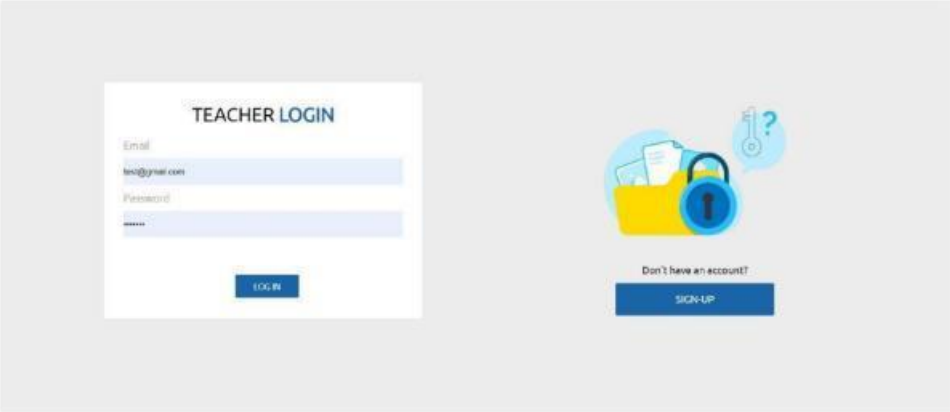

Figure 5.6 Login page

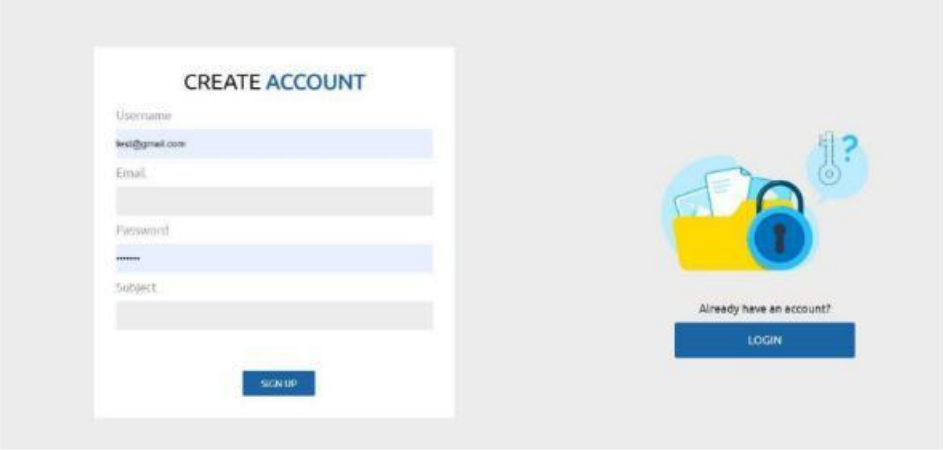

Figure 5.7 Sign up page

Description:

In Figure 5.5, Figure 5.6 & Figure 5.7 shows the UI design of the web application, the starting page, sign up page and log in page.

CHAPTER 6

SYSTEM IMPLEMENTATION

6.1 Pre-processing Images

The technology takes pictures of each person's face. The photographs are saved in a folder after being converted to jpeg files. A name and ID specific to that individual will be kept with the stored photographs.

6.2 OpenCV

OpenCV is a name for a free and open-source software library for computer vision and machine learning. To hasten the inclusion of artificial intelligence into goods, OpenCV was used to develop a common infrastructure for computer vision applications. Since OpenCV is a BSD-licensed software, businesses can easily use and modify the code. The collection contains more than 2500 optimized algorithms, including several both established and cutting-edge computer vision and machine learning methods. These algorithms can be applied to a variety of tasks, including finding related images in a database of images, removing red eyes from flash-taken photos, tracking eye movements, identifying objects, classifying human actions in videos, tracking camera movements, tracking moving objects, extracting 3D models of objects, creating 3-point clouds from stereo cameras, stitching together images to create high-resolution scenes, and creating markers to overlay images. It offers C++, Python, Java, and MATLAB interfaces and supports Windows, Linux, Android, and Mac OS.

6.3 Flask

Flask is a small, lightweight Python web framework that makes it easier to create online applications by providing useful tools and features. It gives developers flexibility and is a more approachable framework for new developers because you can easily design a web application using just one Python file. Additionally, Flask may be expanded and doesn't require a specific directory structure or a lot of boilerplate code to use it. Flask requires Python 2.6 or higher to be installed and is based on the WSGI (Web Server Gateway Interface) toolkit. Jinja2 is also a template engine used by Flask. Start by importing the flask package into any Python IDE.

6.4 Face Recognition

The easiest face recognition library in the world enables you to recognize and manipulate faces from Python or the command line. built utilizing dlib's most cutting-edge facial recognition technology. The model's accuracy on the benchmark for Labeled Faces in the Wild is 99.38 percent. In addition, an easy command line application for face recognition is included, enabling you to run face recognition on a directory of images right from the command line. With this package, real-time facial recognition using other Python packages is possible.

6.5 MongoDB

MongoDB is a free and open-source NoSQL database. Since it is a non-relational database, it can handle structured, semi-structured, and unstructured data. It uses a document-oriented, non-relational data model with an unstructured query language. MongoDB's great versatility allows you to integrate and store a wide variety of data types. It also maintains and handles a greater volume of data than traditional relational databases. BSON, a more adaptable binary variant of JSON that is utilized by MongoDB, is the name of the document storage format used by this database (JavaScript Object Notation). For our project, we will utilize a real-time MongoDB database since it is adaptable, immediately updates as the content is modified, and automatically updates across the application. Additionally, it is simple to use, and using the MongoDB Compass app, users can quickly find data in the MongoDB database without using commands.

CHAPTER 7

TESTING

The testing is a process of verifying and validating whether the actual results of the software is matching to the expected results and making sure that the software is free of bugs and errors. This will make the users confidence gained by the quality of the product. Verification is a set of activities that verify the correctness of the implemented software.

7.1 Web App Testing

The Web app is created using Flask. Web App can operate on both smartphones and computers. It is also compatible with practically all browsers, thus the Web app will work on all devices and computer

7.2 Test Result

Illustrates the sample result of the attendance marking

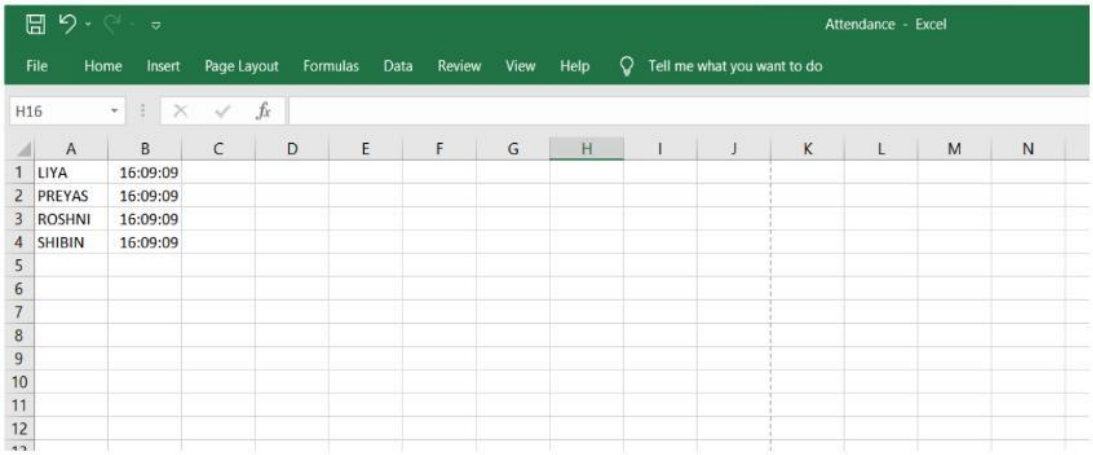

Figure 7.1 Sample Result

7.3 Test Report

Table 7.1 Test Report

Test case ID	Test Case	Test Step	Test Data	Expected Result	Result Obtained	Status (Pass/Fail)
T1	User enters the wrong user id or password and gets alerted.	User enters the incorrect user id or password	<User Id and password>	Alert showing incorrect user id or password	Alert showing incorrect user id or password	Pass
T2	User enters the correct user id and password	User enters the correct user id or password	<user id and password>	Redirect the user to his/her dashbord	User login successfully	Pass
T3	New user creates a new account using create new account tab	New user creates a new account by entering new user id and password using create new account tab	<new user id and password>	Give message as user as successfully created the account	Successfully created the account	Pass
T4	User captures the images and system recognizes the people in image and mark the attendance	User capture the image	<image of the students>	People present in the image was identified and attendance was marked	People present in the image was identified and attendance was marked	Pass
T5	Attendance displaying in a csv file	User capture the image	<input image>	Displays the attendance of students in a csv file	Displays the attendance of students in a csv file	Pass
T6	User Download the marked attendance of students	User click the Download attendance tab	<input csv file>	User can easily download the file	Download the file	Pass

CHAPTER 8

RESULTS

We have implemented a web application for attendance management where the user marks the attendance using face recognition. They can also download the marked attendance as a .csv file.

At first, the user can see the homepage of the web app.

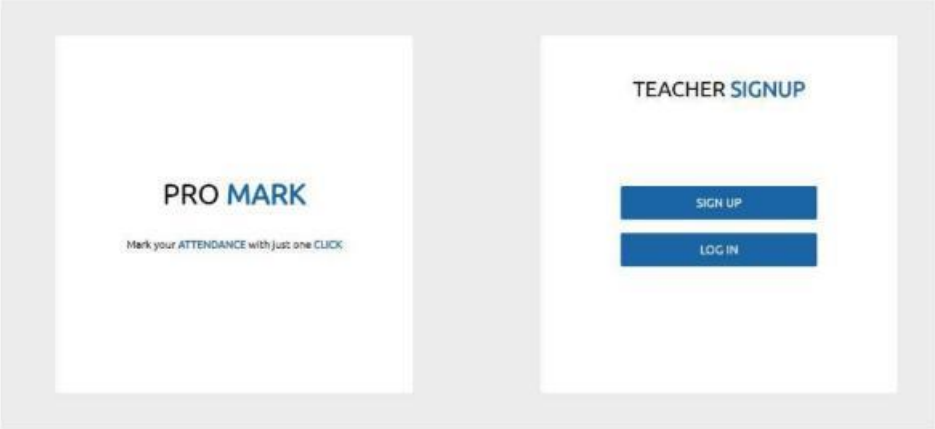

Figure 8.1 Home page

Then the user can move into the signup or login page where the user has the option to login and register by filling the credentials needed.

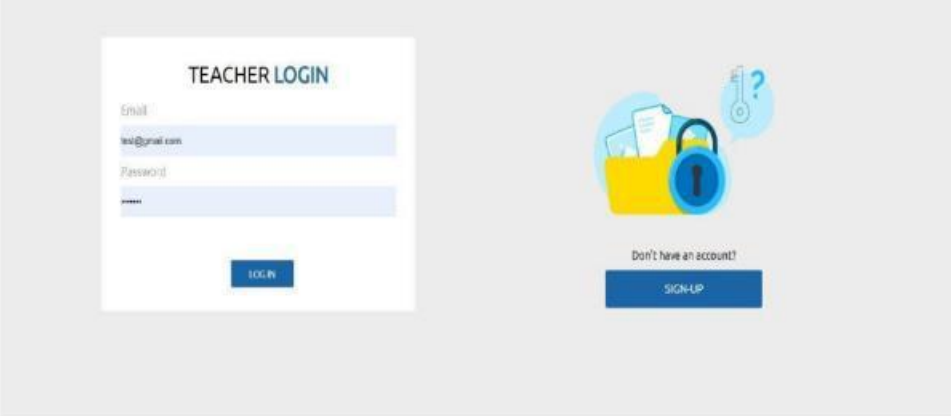

Figure 8.2 Login page

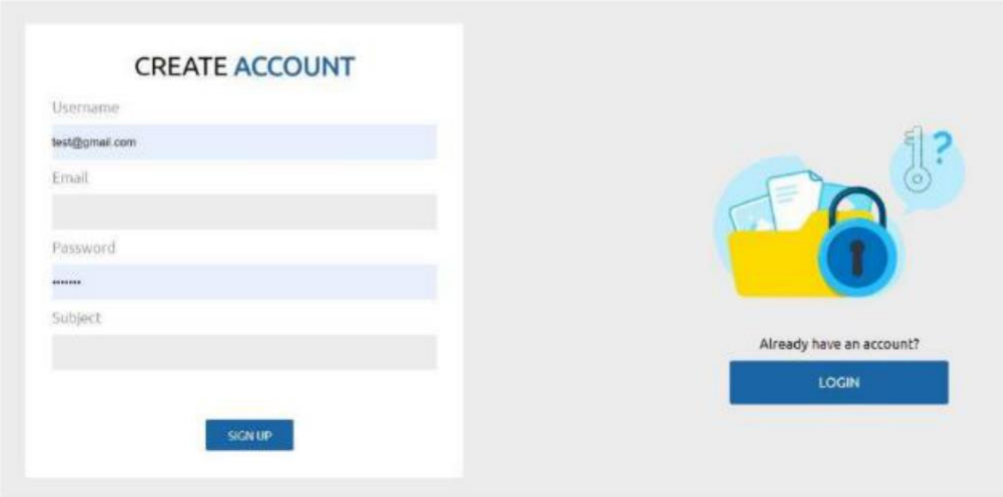

Figure 8.3 Sign up page

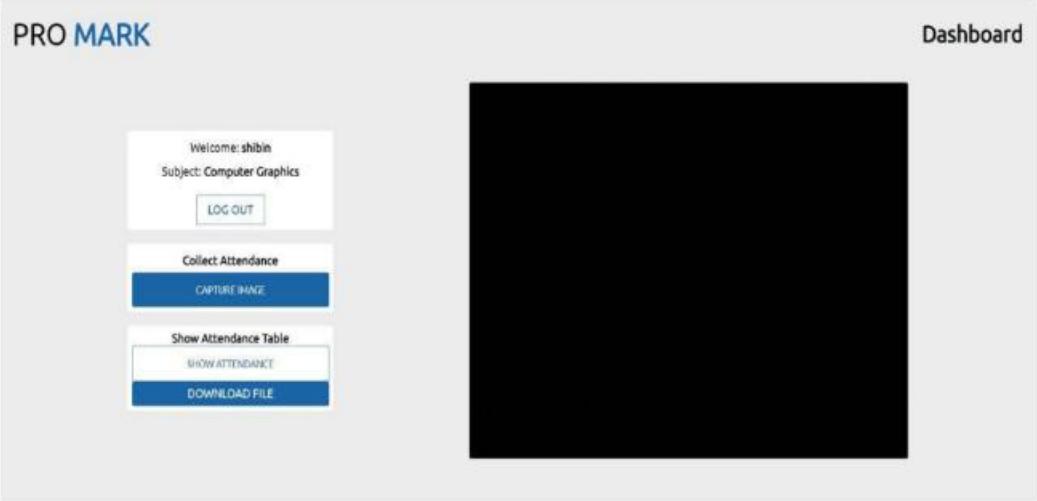

Figure 8.4 Dashboard

Here the user can take the attendance by clicking “Capture Image” and the camera is opened in a window to capture the image of the student.

By clicking on the “Show attendance” button the user can see the marked attendance and clicking the “Download file” will download the marked attendance as .csv format into the device.

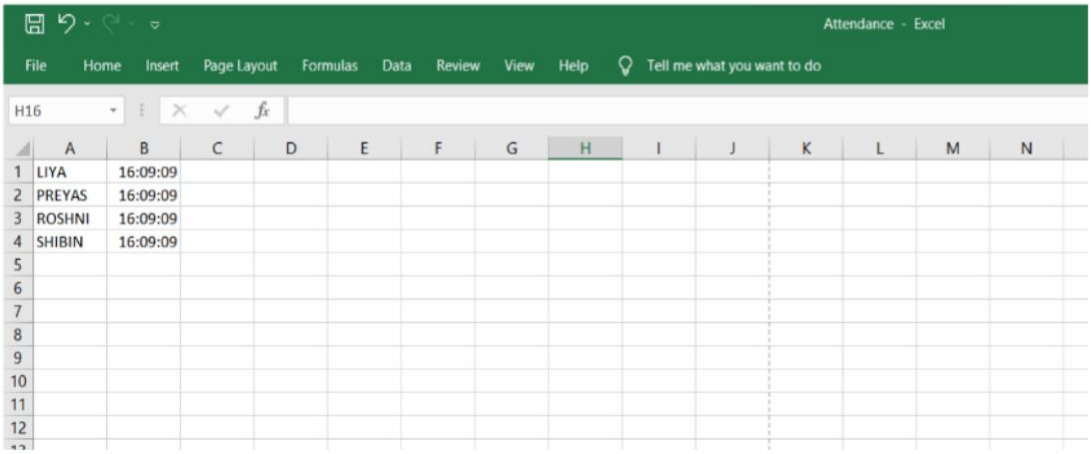

Figure 8.5: Downloaded .csv file

CHAPTER 9

CONCLUSIONS

We have created a web application to manage attendance. It provides the option to take a photo of each student in a class to record their attendance. The collected faces will be recognized by the machine vision training module, which will be running in the background. Additionally, the Web application will record their attendance and show it to the user in a.csv file. As a result, the users of our suggested model will be able to note the students' attendance in class and show that information together with the current time. The system is accurate up to 94% of the time, according to the results of our experiments. As a result, our Web application helps the institutions that manually record attendance.

REFERENCES

[1] A. Raghuwanshi and D. P. D. Swami, "An Automated Classroom Attendance System Using Video Based Face Recognition," in *IEEE International Conference On Recent Trends in Electronics Information & Communication Technology (RTEICT)*, India, 2017.

[2] M. I. P. D. Trinos, J. H. Rios, K. G. O. Portades, P. R. O. Portades, R. M. P. Langreo and M. B. Abisado, "Real-time Class Attendance Monitoring using Smart Face Recognition," IEEE Xplore, 2020.

[3] M. K. G. Rani, P. S and S. S, "AN AUTOMATED STUDENT ATTENDANCE," *Wutan Huatan Jisuan Jishu*, vol. XVI, no. VI, 2020.

[4] S. P. S. Hegde and P. , "Face Recognition based Attendance Management," *International Journal of Engineering Research & Technology (IJERT)*, vol. 9, no. 05, 2020.

[5] R. S. Bharadwaj, T. S. Rao and V. T. R, "Attendance Management Using Facial Recognition," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 8, no. 6, 2019.

[6] E. C. SAI, S. KHAJA, S. A. HUSSAIN and SHYAMAMARA, "STUDENT ATTENDANCE MONITORING SYSTEM USING FACE RECOGNITION," SSRN, 2021.

[7] O. Kainz, M. Nguyen, F. Jakab, R. Petija, M. Michalko and G. Alexandrova, "Students' Attendance Monitoring through the Face Recognition," in *International Conference on Emerging eLearning Technologies and Applications (ICETA)*, Slovakia, 2019.

[8] S. Dev and T. Patnaik, "Student Attendance System using Face Recognition," in *International Conference on Smart Electronics and Communication (ICOSEC)*, 2020.

APPENDIX

Sample code for attendance marking

```
import cv2 as cam
import numpy as np
import face_recognition as fr
import os
from datetime import datetime
import time

class DetectingAttendance:
    def DetectingAttendance(self):
        path = r'C:\Users\Mini_Project_G9\Samples'
        images = []
        className = []

        List = os.listdir(path)
        print(List)

        for cl in List:
            curImge = cam.imread(f'{path}/{cl}')
            images.append(curImge)
            className.append(os.path.splitext(cl)[0])
        print(className)



        def findEncodings(images):
            eList = []
            for imge in images:
                imge = cam.cvtColor(imge, cam.COLOR_BGR2RGB)
                try:
                    encode = fr.face_encodings(imge)[0]
                except IndexError as e:
                    print(e)
```

```
            exit(1)
        print(len(encode))
        eList.append(encode)
    return eList


def AttendanceMark(person):
    with open('attendance.csv', 'r+') as f:
        DataList = f.readlines()
        nameList = []
        for line in DataList:
            entry = line.split(',')
            nameList.append(entry[0])
        if person not in nameList:
            current = datetime.current()
            dString = current.strftime('%H:%M:%S:')
            f.writelines(f'\n{person}, {dString}')


eListKnown = findEncodings(images)
print(' The Encoding is Completed')

 capture = cam.VideoCapture(0)

while True:
    success, imge = capture.read()
     imgeD = cam.resize(imge, (0, 0), None, 0.25, 0.25)
     imgeD = cam.cvtColor(imgeD, cam.COLOR_BGR2RGB)
     faceCurrentFrame = fr.face_locations(imgeD)
    encodeCurrentFrame=fr.face_encodings(imgeD,   faceCurrentFrame)

    for eFace, faceLoc in zip(encodeCurrentFrame, faceCurrentFrame):

 matches = fr.compare_faces(eListKnown, eFace)
```

```
            faceDist = fr.face_distance(eListKnown, eFace)
            print(faceDist)
            matchIndex = np.argmin(faceDist)

           if matches[matchIndex]:
               person = className[matchIndex].upper()
               print(person)
               y1, x2, y2, x1 = faceLoc
               y1, x2, y2, x1 = y1 * 4, x2 * 4, y2 * 4, x1 * 4 cam.rectangle(imge,
               (x1, y1), (x2, y2), (0, 255, 0), 2) cam.rectangle(imge, (x1, y2 - 35),
               (x2, y2), (0, 255, 0), cam.FILLED)
               cam.putText(imge, person, (x1 + 6, y2 - 6),
cam.FONT_HERSHEY_COMPLEX, 1, (255, 255, 255), 2)
    imge_path = r'C:\Users\Mini_Project_G9\shots'
    dir_path = r'C:\Users \Mini_Project_G9\shots'
    imageList = os.listdir(imge_path)

    for images in imageList:
        count = 0
        new_path = imge_path + '\\' + imageList[count]

        count += 1

        print(new_path)
        imge = cam.imread(new_path)

        imgeD = cam.resize(imge, (0, 0), None, 0.25, 0.25)
        imgeD = cam.cvtColor(imgeD, cam.COLOR_BGR2RGB)
        faceCurrentFrame = fr.face_locations(imgeD)
        encodeCurrentFrame=fr.face_encodings(imgeD, faceCurrentFrame)

        for eFace, faceLoc in zip(encodeCurrentFrame, faceCurrentFrame):
            matches = fr.compare_faces(eListKnown, eFace)
            faceDist = fr.face_distance(eListKnown, eFace)
```

```
            print(faceDist)
            matchIndex = np.argmin(faceDist)

            if matches[matchIndex]:
                person = className[matchIndex].upper()
                print(person)
                print("Face Detection completed")
                y1, x2, y2, x1 = faceLoc
                y1, x2, y2, x1 = y1 * 4, x2 * 4, y2 * 4, x1 * 4 cam.rectangle(imge,
                (x1, y1), (x2, y2), (0, 255, 0), 2) cam.rectangle(imge, (x1, y2 - 35),
                (x2, y2), (0, 255, 0), cam.FILLED)
                cam.putText(imge, person, (x1 + 6, y2 - 6),
cam.FONT_HERSHEY_COMPLEX, 1, (255, 255, 255), 2)
                AttendanceMark(person)
        print('Program Execution Completed')
        time.sleep(2)
```