

AUTOMATED ATTENDANCE MANGEMENT SYSTEM

A MINI PROJECT REPORT

Submitted By

D.SPARGEN PAUL RAJ

191201045

S. SRIVATSAN

191201047

S. SUBIN

191201048

In partial fulfillment for the award of the

degree of

BACHELOR OF ENGINEERING

IN

MECHATRONICS ENGINEERING



RAJALAKSHMI ENGINEERING COLLEGE

ANNA UNIVERSITY: CHENNAI 600 025

JUNE 2022

RAJALAKSHMI ENGINEERING COLLEGE
(AN AUTONOMOUS INSTITUTION AFFILIATED TO ANNA
UNIVERSITY)

BONAFIDE CERTIFICATE

Certified that this project report “**AUTOMATED ATTENDANCE MANAGEMENT SYSTEM**” is the bonafide work of **SPARGEN PAUL RAJ [Reg No. 191201045]**, **SRIVATSAN S [Reg No.191201047]**, **SUBIN S [Reg No. 191201048]** who carried out the project under my supervision.

SIGNATURE

Dr. V SANTHANAM
Head of the Department

Professor
Department of Mechatronics
Engineering
Rajalakshmi Engineering College
Chennai – 602 105

SIGNATURE

Ms. NARESH BABU
Supervisor

Assistant Professor
Department of Mechatronics
Engineering
Rajalakshmi Engineering College
Chennai – 602 105

This project is submitted for VIVA-VOCE examination held on -05-2022.....
at Rajalakshmi Engineering College, Autonomous. Chennai.

INTERNAL EXAMINER

EXTERNAL EXAMINER

ABSTRACT

Currently manual student attendance marking technique is often facing a lot issues and a very slow process. Teachers or faculty calling names of student from their data sheet and student responding to them. But this existing process becomes very complex in large classes that consists so many students. Many times, students also mark proxies by responding to fake name. This makes disturbance in class and distracts the students during the exam times. Apart from calling names attendance sheet is passed around classroom during lectures especially the classes consisting large number of students might find it hard to have attendance sheet being passed around the class.

To overcome this, we developed an **AUTOMATED ATTENDANCE MANAGEMENT SYSTEM WITH FACE RECOGNITION**. The main objective of this project is to make face recognition based automated attendance system. This system is very useful in daily life applications especially in security and surveillance systems. Overall, we have created a program that take the image from the database and make all the necessary conversions for recognition and then verifies the image in the videos or in the real time by accessing the camera through user friendly interface.

ACKNOWLEDGEMENT

Initially we thank the almighty for being with us through every walk of life it is our privilege to express our sincerest thanks to our respected Founder and Chairman **Mr.S.Meganathan, B.E., F.I.E.**, our Chairperson **Dr. (Mrs)Thangam Meganathan ,M.A., M.Phil., Ph.D** and **Mr.Abhay Shankar Meganathan, B.E., M.S.**, Vice Chairman of Rajalakshmi Engineering College for providing us with the requisite infrastructure and extending support in all endeavors.

Our heartfelt thanks to **Dr. S. N. Murugesan**, our Principal for his kind support and resources provided to complete our work in time. We also thank **Dr. S. P Srinivasan M.E., Ph.D**, Director of Mechanical Sciences for his suggestion and guidance for completion of our project.

We deeply express our sincere thanks to **Dr. Santhanam V, M.E., Ph.D.** Head of our Department, for his encouragement and continues support to complete the project in time.

We are glad to express our sincere indebtedness to our project coordinator **Mr. Eswaran A, M.E.** Department of Mechatronics engineering for his constructive criticism throughout the duration of our project.

We express our sincere thanks to **Dr. Santhanam V**, Head of the Department, Mechatronics Engineering for his guidance and encouragement throughout the project work. We convey our sincere and deepest gratitude to our internal guide, **Mr. Naresh Babu G**, Assistant Professor, Department of Mechatronics Engineering, for his valuable guidance throughout the course of the project.

Finally, we express our gratitude to our parents and classmates for their moral support and valuable suggestions during the course of the project.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	ii
	LIST OF TABLES	vi
	LIST OF FIGURES	vi
1.	INTRODUCTION	01
	1.1 DESCRIPTION	01
	1.2 PROJECT OBJECTIVES	04
	1.3 PROJECT SPECIFICATION	04
	1.4SCOPE OF THE PROJECT	05
2.	LITERATURE SURVEY	06
3.	METHODOLOGY	09
	3.1 DATA CREATION	09
	3.2 FACE DETECTION	10
	3.3 FACE RECOGNITION	10
	3.4 ATTENDANCE UPDATION	11
	3.5 PSEUDOCODE	11

4.	COMPONENTS DESCRIPTION	12
	4.1 HARDWARE	12
	4.1.1 ESP32 CAM MODULE	12
	4.1.2 POWER BANK	13
	4.1.3 FDTI MODULE	15
	4.1.4 OVA2043 CAM MODULE	16
	4.2 SOFTWARE	17
	4.2.1 PYCHARM	17
	4.2.2 OPENCV	18
	4.2.3 PYTHON GUI TKINTER	18
5.	PROJECT DESIGN	19
6.	RESULT AND DISCUSSION	22
7.	CONCLUSION	23
	REFERENCE	24
	APPENDIX	25

LIST OF TABLES

TABLE NO.	TITLE	PAGE NO.
7.1	COST ESTIMATION	46

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO.
1.1	BLOCK DIAGRAM OF GENERAL FRAMEWORK	04
3.1	FLOW CHART	09
4.1	ESP32 MODULE	12
4.2	POWER BANK	14
4.3	FDTI MODULE	15
4.4	OV2640-Cam module	17
5.1	GUI-INTERFACE	19
5.2	ASSEMBLED VIEW OF PROJECT	20
5.3	FACE RECOGNITION	20
5.4	EXCEL SHEET	21
5.5	INTERNAL CIRCUIT	21

CHAPTER 1

1. INTRODUCTION

1.1 DESCRIPTION:

Attendance management system is developed in order to monitor the attendance of students primarily in schools, colleges and universities. It facilitates to store the information taken from the students and store it. This will help in sorting the attendance of the student and effectively monitor them. An accurate result of the student's attendance will be generated. This will be useful in monitoring the student's eligibility criteria through their attendance. The existing system is a model that is visible throughout institutions and universities. It is a manual entry attendance system where the professor would call out the names and roll numbers of the students and thereby, manually mark them in an attendance register. This would then be used to notify the students of their attendance percentage and thereby, in a way monitor them. There are a lot of difficulties faced in this system. There is the effort of calling out the students roll nos. and individually mark the attendance which would lead to ineffective consumption of time. There is also the question of accurately marking the attendance despite the disturbances caused in an average classroom. The right inputs need to be entered into the respective fields and supposing there is a slight misinput, it will ultimately lead to miscalculation of the attendance. So, considering the above factors, it can be

inferred that the user finds it difficult to manage the attendance, at least to a certain degree if not completely. The purpose of developing the automated system is to computerize the traditional way of managing attendance. Another important aspect of this project would be to generate the report automatically and thus manage it effectively. The system which is developed can be used remotely via a laptop/desktop through wireless networks. The main advantage of this is that it can be run remotely and the attendance will be registered automatically. However, this would require a wireless connection or a local host in order to perform its task. Development of this application is highly economically feasible. The only thing to be done is making an environment with effective supervision. It is cost effective in the sense that has eliminated the paper work completely. The system is also time effective because the calculations are automated which are made at the end of the month or as per the user requirement` Advantages of Proposed System:

- It is trouble-free to use.
- It is a relatively fast approach to enter attendance
- Is highly reliable, approximate result from user
- Best user Interface.
- Efficient reports

The system working is quite easy to use and learn due to its simple but attractive interface. User requires no special training for operating the system. Technical performance includes issues such as determining whether the system can provide

the right information for the Department personnel student details, and whether the system can be organized so that it always delivers this information at the right place and on time using internet services. Acceptance revolves around the current system and its personnel. Report can be taken by daily, weekly and can consolidate:

- weekly report gets all hour details of attendance starting date to ending date and display the status
- Consolidate report get all student attendance details starting date ending date status help for the eligibility criteria of the student to attend the examination.

The output is designed in such a way that it is attractive, convenient and informative. Forms are designed with various features, which make the console output more pleasing to the user. As the outputs are the most important sources of information to the users, better design should improve the system's relationships with us and also will help in decision making. Form design elaborates the way output is presented and the layout available for capturing information. The output is designed in such a way that it is attractive, convenient and informative. Forms are designed with various features, which make the console output more pleasing.

1.2 Project Objectives:

1. Reducing time wastage during conventional class attendance.
2. Utilizing latest trends in machine vision to implement a feasible solution for class attendance system.
3. Automating the whole process so that we have digital environment.
4. Preventing fake roll calls as one to one attendance marking is possible only.
5. Encouraging the use of technology in daily lives.

1.3 Project Specifications:

- a. Uses Pattern Matching algorithm for face detection.
- b. Score of minimum 600 required to perfectly match a face.
- c. Metric: Camera Resolution.
- d. For prototype fixed to 10 users only but scalable design.
- e. Requires good lighting condition for better camera capture capability.
- f. Attendance sheet is .xlsx format and can be digitally distributed and maintained.

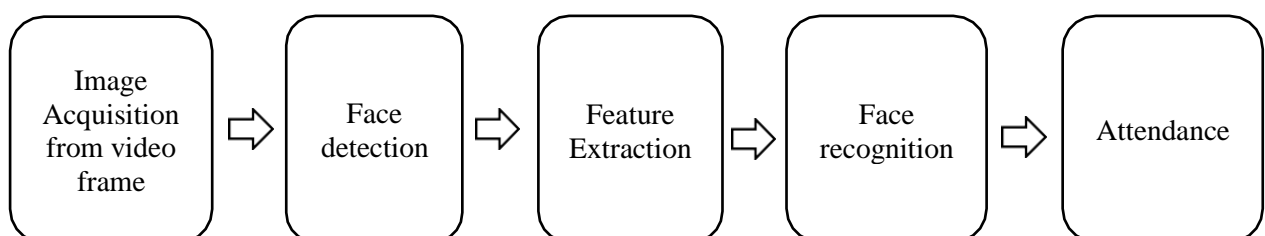


FIG 1.1: BLOCK DIAGRAM OF GENERAL FRAMEWORK

2.1 SCOPE OF THIS PROJECT:

- Provides a valuable attendance service for both teachers and students
- Reduce manual process errors by provide automated and a reliable attendance system uses face recognition technology.
- Produce monthly reports for lecturers.
- Flexibility, Lectures capability of editing attendance records.
- Calculate absenteeism percentage and send reminder messages to students. And having proof to check when any obligation is raised against institution.

CHAPTER 2

2. LITERATURE SURVEY

INTRODUCTION

Automatic face recognition (AFR) technologies have made many improvements in the changing world. Smart Attendance using Real-Time Face Recognition is a real-world solution which comes with day-to-day activities of handling student attendance system. Face recognition-based attendance system is a process of recognizing the students face for taking attendance by using face biometrics based on high - definition monitor video and other information technology. We proposed about the research, observation and finding that have been made regarding this project field. The biggest advantage of human beings are we are the best creatures in adapting towards the situation. All the related research papers and journals that provide thought and concept concerning this project ground also is explained into a simple means.

1)Bernie Di Dario, Michael Dobson, Douglas Halers, Feb.16, 2006. “Attendance Tracking System”, United States Patent Application Publication, Pub. No.: US 2006/0035205 A1.

This framework consists of identity tags, with wireless communication capabilities for each attendee and the scanners for detecting the attendee's tags as they enter in that allocated room.

2) O.A. Idowu and O. Showed, May 2012 (Spring), “Attendance

Management System using Biometrics”. The Pacific Journal of Science And Technology, Vol.13, Number1.

Attendance is taken with the help of a finger print device and the records of attendance is stored in the database. Attendance is marked after successful identification.

3) Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on (Vol. 1, pp. I-511)

It describes a machine learning approach for visual object detection which is capable of processing images extremely rapidly and achieving high detection rates.

4) S. Kadri and K. Ismaili - "A design and implementation of a wireless iris recognition attendance management system", Information Technology and control, vol. 36, no.3, pp. 323329, 2007.

the authors have proposed Doughmans algorithm-based Iris recognition system. This system uses iris recognition management system that does capturing the image of iris recognition, extraction, storing and matching.

5) T. Lim, S. Sim, and M. Mansoor - "Rd based attendance system", In Industrial Electronics Applications, 2009. ISIEA 2009. IEEE Symposium on, vol. 2. IEEE, 2009, pp. 778782.

the authors have proposed RFID based system in which students carry a

RFID tag type ID card and they need to place that on the card reader to record their attendance.

6) International Journal of Engineering Applied Sciences and Technology, 2021 Vol. 5, Issue 12, ISSN No. 2455-2143, Pages 233-241 Published Online April 2021 in IJEAST.

The author has proposed the system that works automatically once the registration of individual student is created by the administration. There is a need to utilize few algorithms that can perceive the appearances in order to improve the system performance and recognition accuracy.

7) Real-Time Group Face-Detection for an Intelligent Class-Attendance System I.J. Information Technology and Computer Science, 2015, 06, 66-73 Published Online May 2015.

In this paper, the author has proposed a group face detection algorithm to detect student faces during any class lecture. This algorithm works in real time and depends on catching the video from a PC digital camera. This algorithm was implemented using MATLAB 2012.

CHAPTER 3

3. METHODOLOGY

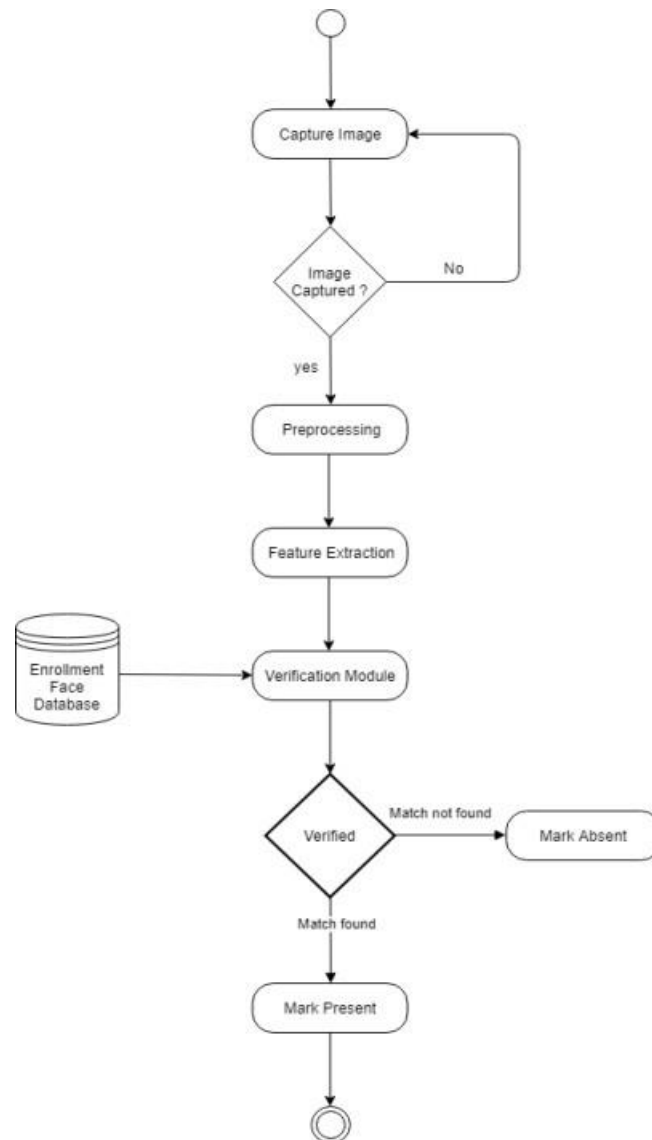


Fig.3.1: Flow Chart

The process can be divided into four stages,

3.1 : Dataset Creation:

Images of students are captured using a cam module of esp32. Multiple images of single student will be acquired with varied gestures and angles. These images

undergo pre-processing. Next step is to resize the cropped images to particular pixel position. Then these images will be converted from RGB to Gray scale images. And then these images will be saved as the names of respective student Id in a folder.

3.2: Face Detection:

Face detection here is performed using Har-Cascade Classifier with OpenCV. Har Cascade algorithm needs to be trained to detect human faces before it can be used for face detection. This is called feature extraction. The hair cascade training data used is an xml file- haarcascade_frontalface_default. Here we are using detect Multiscale module from OpenCV. This is required to create a rectangle around the faces in an image. It has got three parameters to consider- scale Factor, neighbours, min-size. scale Factor is used to indicate how much an image must be reduced in each image scale. neighbours specify how many neighbours each candidate rectangle must have. Higher values usually detect less faces but detects high quality in image. min-size specifies the minimum object size. By default, it is (30,30) [8]. The parameters used in this system is scale Factor and neighbours with the values 1.3 and 5 respectively.

3.3: Face Recognition:

Face recognition process can be divided into three steps- prepare training data, train face recognizer, prediction. Here training data will be the images present in the dataset. They will be assigned with an integer label of the student it belongs to. These images are then used for face recognition. Face recognizer used in this

system is Local Binary Pattern Histogram. Initially, the list of local binary patterns (LBP) of entire face is obtained. These LBPs are converted into decimal number and then histograms of all those decimal values are made. At the end, one histogram will be formed for each image in the training data. Later, during recognition process histogram of the face to be recognized is calculated and then compared with the already computed histograms and returns the best matched label associated with the student it belongs to.

3.4: Attendance Updating:

After face recognition process, the recognized faces will be marked as present in the excel sheet with time and date.

3.5 Pseudocode:

Input: live video with student face visible

Output: attendance excel sheet

1. Transform each frame from RGB to grayscale
2. Apply the Har Cascade classifier for face detection and get the Region of Interest (ROI).
3. Now apply the LBPH algorithm on the ROI to get the features.
4. if for enrolment then features are stored in the database
else if for verification then do post processing.

CHAPTER 4

4. COMPONENTS DESCRIPTION

4.1 HARDWARE

4.1.1 ESP32 WIFI CAM MODULE

The ESP32-CAM is a small size, low-power consumption camera module based on ESP32. It comes with an OV2640 camera and provides an on-board TF card slot. The ESP32-CAM can be widely used in intelligent IoT applications such as wireless video monitoring, WIFI image upload, QR identification, and Face recognition etc.



Fig. 4.1: Esp32module

Specifications:

- WIFI module: ESP-32S
- Processor: ESP32-D0WD
- Built-in Flash: 32Mbit
- RAM: Internal 512KB + External 4M PSRAM
- Antenna: Onboard PCB antenna
- WIFI protocol: IEEE 802.11 b/g/n/e/I
- Bluetooth: Bluetooth 4.2 BR/EDR and BLE
- WIFI mode: Station / SoftAP / Contestation
- Security: WPA/WPA2/WPA2-Enterprise/WPS
- Output image format: JPEG (OV2640 support only), BMP, GRAYSCALE
- Supported TF card: up to 4G
- Peripheral interface: UART/SPI/I2C/PWM
- IO port: 9
- UART baud rate rate: default 115200bps
- Power supply: 5V

4.1.2 POWER BANK

A Power Bank is a portable charger designed to recharge your electronic devices when you're on the move. Ranging in size from slim, pocket-sized devices up to larger, higher-capacity Power Banks – they can be used to

charge smartphones, tablets.



Fig. 4.2: Power bank

Specifications:

- Discharge current up to 10A
- Full Charge in 40 to 90 minutes depending upon special charger
- Lead acid battery RB1213
- Voltage: 12V
- Capacity: 1.2Ah
- Size: 98mm x 43mm x 52 mm
- Weight: 0.450kg

4.1.3 FDTI MODULE

The FTDI USB to TTL serial converter module is a UART (universal asynchronous receiver-transmitter) board used for TTL serial communication. It is a breakout board for the FTDI FT232R chip with a USB interface, can use 3.3 or 5 V DC and has Tx/Rx and other breakout points. Includes over-current protection, using a 500ma self-restoring fuse. This board includes a DTR pin needed to auto-reset when downloading to your device.

GND—>GND
VCC—>VCC
TX—>RX
RX—>TX
DTR—>GRN

The internal clock (6MHz, 12MHz, 24MHz and 48MHz) can be brought out of the device and used to drive a microcontroller or external logic.

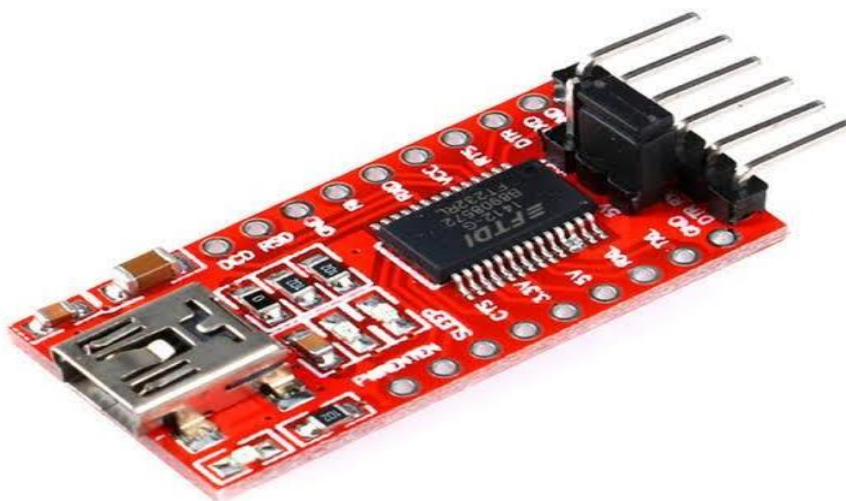


Fig. 4.3: FDTI-module

Specifications:

- Chip: FT232R (FTDI)
- Pins: 6 pin headers and 18 through-hole connectors. DTR, RX, TX, VCC, CTS, GND and PWR, TEN, SLEEP, CTS, 3.3V, 5V, RXL, TXL, GND x 2, TXD, DTR, RTS, VCC, RXD, R1, RSD, DCD (all marked)
- Working voltage: 5V DC
- Level output: 3.3 – 5 V DC (jumper)
- Interfaces: Serial/UART
- On-board LEDs: power on, Rx, Tx transmission, working status
- Operating current: 500mA
- Module size: 36 x 18 mm
- Pin size: male, 5 x 2.54 mm
- Ports: Micro-B USB
- Breadboard friendly: yes

4.1.4 OV2640-CAM MODULE

The OV2640 is the world's first 1/4-inch 2-megapixel sensor. It's a fully integrated Camera Chip™ built noneviction's latest 2.2 micron

OmniPixel2™ architecture and can be assembled in the very popular 8 x 8 mm sockets with a building height as low as 5 mm, essentially thinner than today's 1.3megapixel camera modules.

Specifications:

- power supply core: 1.3VDC
- power consumption active (full resolution) 125 MW (for 15 fps, YUV mode) 140 MW (for 15 fps, compressed mode).

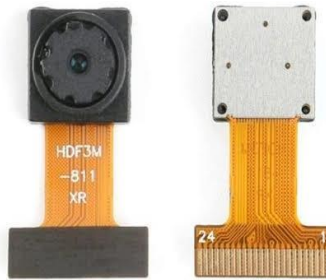


Fig. 4.4: OV2640-Cam module.

4.2 SOFTWARE:**4.2.1 PYCHARM:**

PyCharm is a hybrid platform developed by JetBrains as an IDE for Python. It is commonly used for Python application development. We can run PyCharm on Windows, Linux, or Mac OS. Additionally, it contains modules and packages that help programmers develop software using Python in less time and with minimal effort. Further, it can also be customized according to the requirements of developers.

Features of PyCharm are as follows

1. It acts as an intelligent code editor.

2. It helps developers in editing and enhancing the code with less effort and time.
3. It helps in making efficient and quick changes.
4. It supports popular web technologies and also used to create web application.
5. It supports Python's scientific libraries such as Matplotlib, NumPy and Anaconda.

4.2.2 OpenCV:

OpenCV is the huge open-source library for the computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today's systems. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human. When it integrated with various libraries, such as NumPy, python is capable of processing the OpenCV array structure for analysis. To Identify image pattern and its various features we use vector space and perform mathematical operations on these features.

4.2.3 Python-GUI tinder:

Tintern is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

CHAPTER 5

5. PROJECT DESIGN

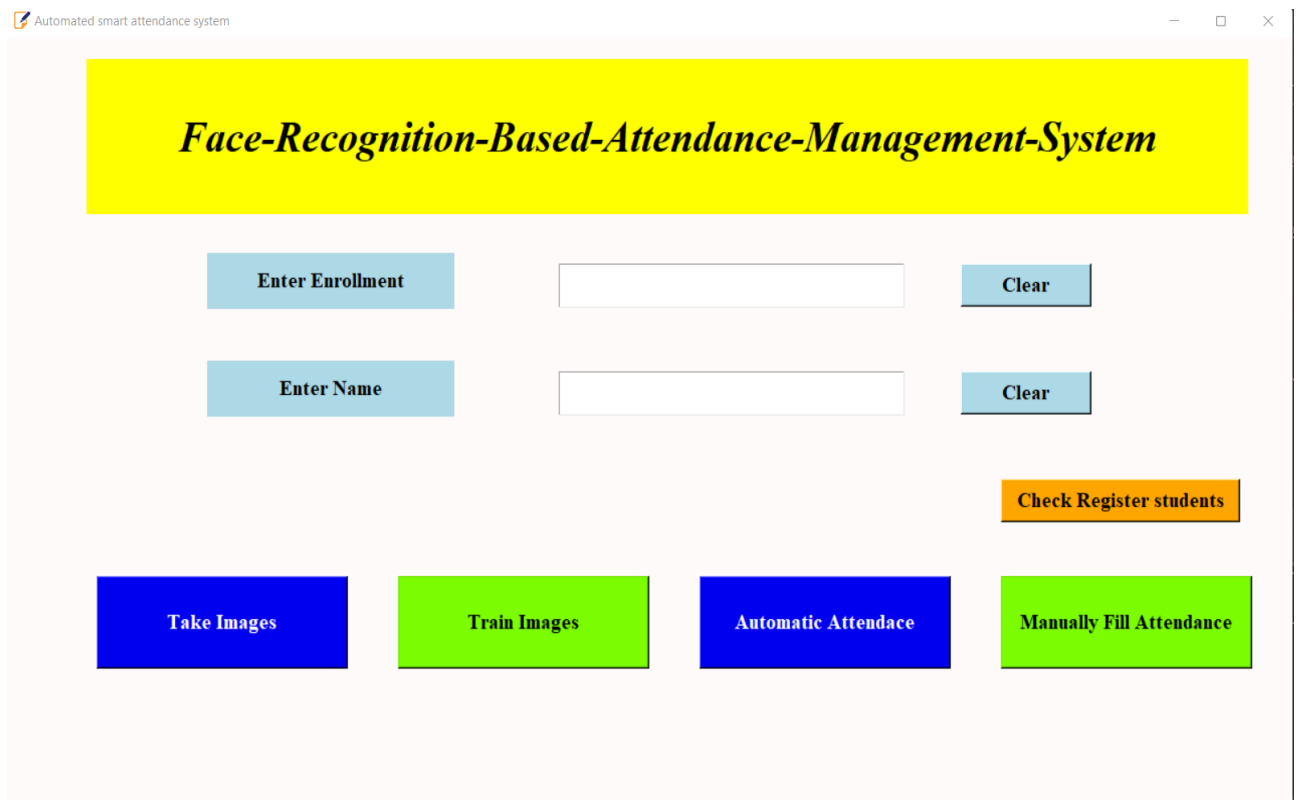


Fig.5.1 GUI-INTERFACE



Fig.5.2 ASSEMBLED VIEW OF PROJECT

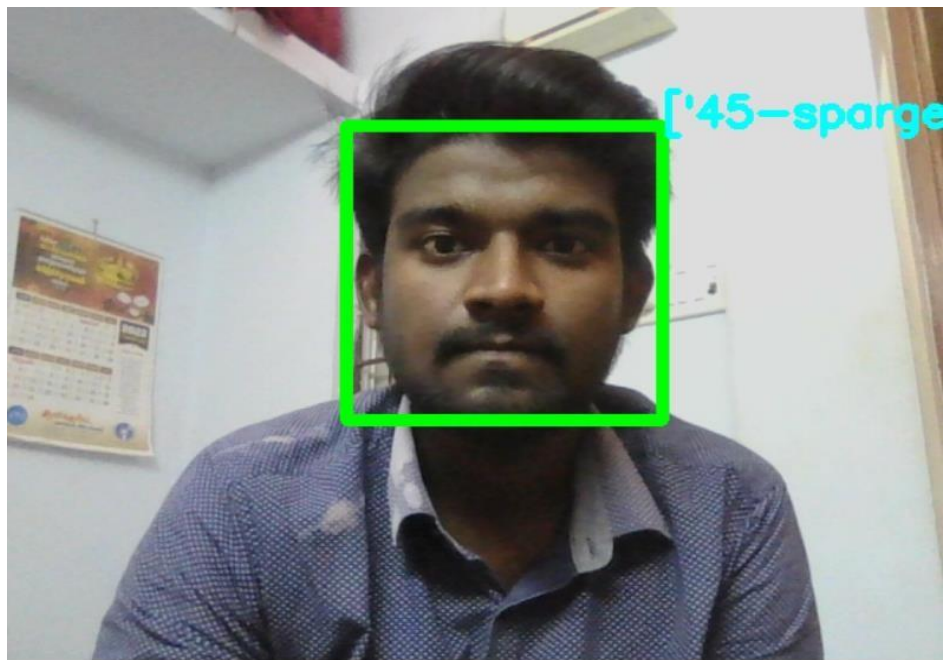


Fig.5.3 FACE RECOGNITION

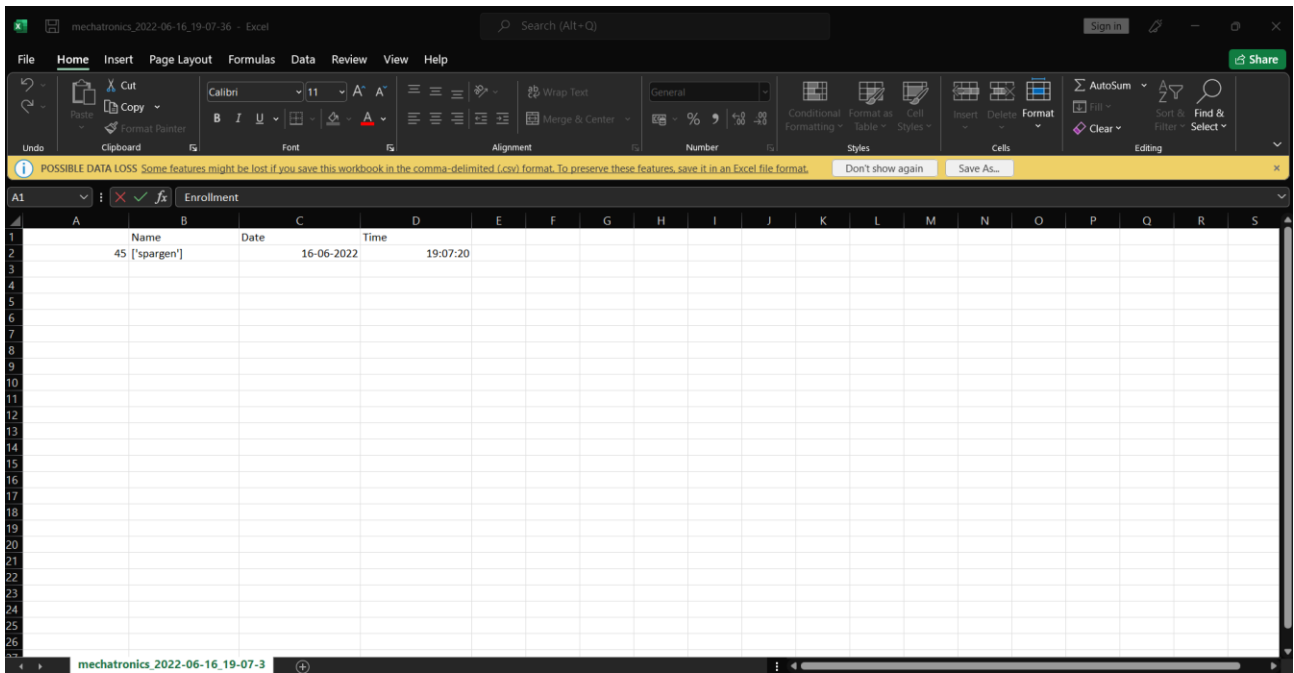


Fig.5.4 EXCEL SHEET

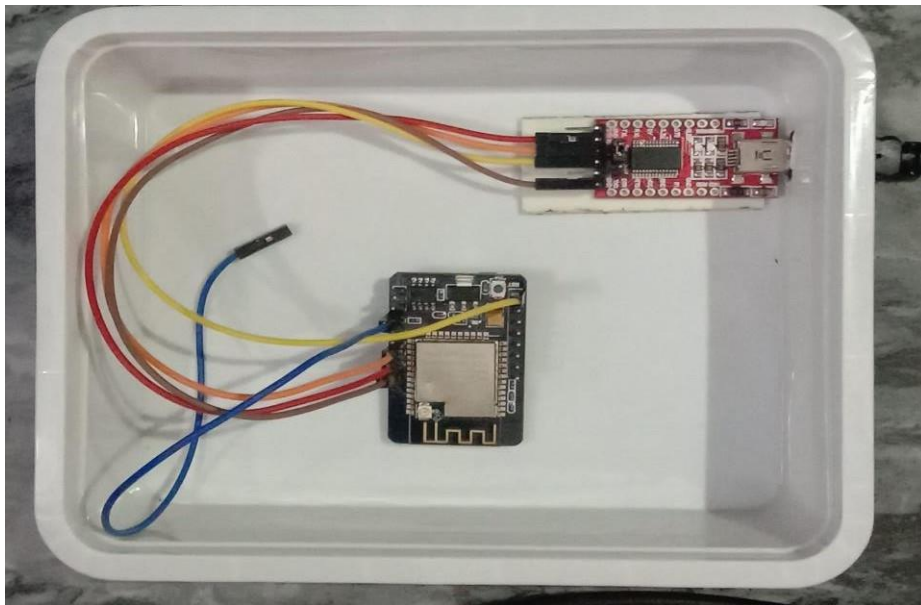


Fig.5.5 INTERNAL CIRCUIT

CHAPTER 6

6. RESULT AND DISCUSSION

In this paper, our goal was to create an automated smart attendance system that can automatically mark the attendance of people and store them in an excel sheet thereby reducing the hectic task of marking attendance manually.

We are looking forward to continue this project and improvise it by adding voice message to notify user after marking attendance increasing accuracy by using AI and ML technologies for fast results in low lights. In this digitalizing world, this technique will contribute greatly to making classrooms digital and help to eliminate the traditional attendance systems problems. this technique allows the record keeper to see attendance automatically without any extra cost and energy whereas the proposed system needs very elementary things such as; a camera, laptop or personal computer and native network. This method is secure, reliable and easy to use.

CHAPTER 7

7. CONCLUSION

The automated smart attendance system is very helpful in making contact less attendance using facial Recognition as it is fast reliable and accurate in marking attendance it is the modern way of taking attendance and is very hard to cheat and can provide data in an excel sheet with proof so no obligation can occur and the false claim cannot be raised. It is one of the major applications compared to other different identification ways like the fingerprint, iris scanner and RFID. The clear image and correct pose may increase the face recognition accuracy. Our system successfully recognizes a student with unintentional changes like wearing glasses or growing beard. It helps to reduce time and effort, especially in the case of large number of students and lectures to be marked attendance. The whole system is implemented in Python programming language. Facial recognition techniques used in the system for the purpose of the student attendance in schools, colleges, university etc....

REFERENCE

- [1] Jomon Joseph and K. P. Zacharia, "Automatic Attendance Management System Using Face Recognition", *International Journal of Science and Research (IJSR)*, 2013.
- [2] Ononiwu G. Chiagozie and Okorafor G. Nwaji, "Radio-frequency identification (RFID)based Attendance System with Automatic Door Unit", *Academic Research International*, 2012.
- [3] O. Shoewu, O.A. Idowu and B. Sc, "Development of Attendance Management System using Biometrics", *The Pacific Journal of Science and Technology*, 2012.
- [4] T. Ahonen, A. Hadid and M. Pietikäinen, "Face Recognition with Local Binary Patterns", *Proceedings of the Advances in Visual Computing*, vol. 3021, pp. 469-481, 2004
- [5] A. Ozdil and M. M. Ozbilen, "A survey on comparison of face recognition algorithms", *2014 IEEE 8th International Conference on Application of Information and Communication Technologies (AICT)*, pp. 1-3, 2014
- [6] R. Samet and M. Tanriverdi, "Face Recognition-Based Mobile Automatic Classroom Attendance Management System", *2017 International Conference on Cyberworlds (CW)*, pp. 253-256, 2017
- [7] A. Ahmed, J. Guo, F. Ali, F. Deebea and A. Ahmed, "LBPH based improved face recognition at low resolution", *2018 International Conference on Artificial Intelligence and Big Data (ICAIBD)*, pp. 144-147, 2018.

APPENDIX

A- SOURCE CODE

Esp32 WIFI cam module (URL Transmission and reception)

```
#include <WebServer.h>
#include <WiFi.h>
#include "esp32cam.h"

const char* WIFI_SSID = "realme 3 Pro";
const char* WIFI_PASS = "12345678";

WebServer server(80);

static auto loRes = esp32cam::Resolution::find(320, 240);
static auto midRes = esp32cam::Resolution::find(350, 530);
static auto hiRes = esp32cam::Resolution::find(800, 600);
void serveJpg()
{
    auto frame = esp32cam::capture();
    if (frame == nullptr) {
        Serial.println("CAPTURE FAIL");
        server.send(503, "", "");
        return;
    }
    Serial.printf("CAPTURE OK %dx%d %db\n", frame->getWidth(), frame->getHeight(),
        static_cast<int>(frame->size()));

    server.setContentLength(frame->size());
    server.send(200, "image/jpeg");
    WiFiClient client = server.client();
    frame->writeTo(client);
}

void handleJpgLo()
{
    if (!esp32cam::Camera.changeResolution(loRes)) {
        Serial.println("SET-LO-RES FAIL");
    }
    serveJpg();
}

void handleJpgHi()
{
    if (!esp32cam::Camera.changeResolution(hiRes)) {
        Serial.println("SET-HI-RES FAIL");
    }
}
```



```

    }
    serveJpg();
}

void handleJpgMid()
{
    if (!esp32cam::Camera.changeResolution(midRes)) {
        Serial.println("SET-MID-RES FAIL");
    }
    serveJpg();
}

void setup(){
    Serial.begin(115200);
    Serial.println();
    {
        using namespace esp32cam;
        Config cfg;
        cfg.setPins(pins::AiThinker);
        cfg.setResolution(hiRes);
        cfg.setBufferCount(2);
        cfg.setJpeg(80);

        bool ok = Camera.begin(cfg);
        Serial.println(ok ? "CAMERA OK" : "CAMERA FAIL");
    }
    WiFi.persistent(false);
    WiFi.mode(WIFI_STA);
    WiFi.begin(WIFI_SSID, WIFI_PASS);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
    }
    Serial.print("http://");
    Serial.println(WiFi.localIP());
    Serial.println(" /cam-lo.jpg");
    Serial.println(" /cam-hi.jpg");
    Serial.println(" /cam-mid.jpg");

    server.on("/cam-lo.jpg", handleJpgLo);
    server.on("/cam-hi.jpg", handleJpgHi);
    server.on("/cam-mid.jpg", handleJpgMid);

    server.begin();
}

void loop()
{
    server.handleClient();
}

```

OPENCV FACE RECOGNITION CODE

AMS_RUN.PY CODE

```
import tkinter as tk
from tkinter import *
import cv2
import csv
import os
import numpy as np
from urllib.request import urlopen
from PIL import Image, ImageTk
import pandas as pd
import datetime
import time
url = r'http://192.168.43.53/cam-hi.jpg'
#####Window is our Main frame of system
window = tk.Tk()
window.title("Automated smart attendance system")

window.geometry('1280x720')
window.configure(background='snow')

#####GUI for manually fill attendance

def manually_fill():
    global sb
    sb = tk.Tk()
    sb.iconbitmap('AMS.ico')
    sb.title("Enter subject name...")
    sb.geometry('580x320')
    sb.configure(background='snow')

def err_screen_for_subject():

    def ec_delete():
        ec.destroy()
    global ec
    ec = tk.Tk()
    ec.geometry('580x320')
    ec.iconbitmap('AMS.ico')
    ec.title('Warning!!')
    ec.configure(background='snow')
    Label(ec, text='Please enter your subject name!!!', fg='red', bg='white', font=('times',
```

```

16, ' bold ')).pack()
    Button(ec, text='OK', command=ec_delete, fg="black", bg="lawn green", width=9,
height=1, activebackground="Red",
        font=('times', 15, ' bold ')).place(x=90, y=50)

def fill_attendance():
    ts = time.time()
    Date = datetime.datetime.fromtimestamp(ts).strftime('%Y_%m_%d')
    timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
    Time = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
    Hour, Minute, Second = timeStamp.split(":")
    #####Creatting csv of attendance

    ##Create table for Attendance
    date_for_DB = datetime.datetime.fromtimestamp(ts).strftime('%Y_%m_%d')
    global subb
    subb=SUB_ENTRY.get()
    DB_table_name = str(subb + "_" + Date + "_Time_" + Hour + "_" + Minute + "_" +
Second)

    import pymysql.connections

    ###Connect to the database
    try:
        global cursor
        connection = pymysql.connect(host='localhost', user='root', password="",
db='manually_fill_attendance')
        cursor = connection.cursor()
    except Exception as e:
        print(e)

    sql = "CREATE TABLE " + DB_table_name + "
        (ID INT NOT NULL AUTO_INCREMENT,
        ENROLLMENT varchar(100) NOT NULL,
        NAME VARCHAR(50) NOT NULL,
        DATE VARCHAR(20) NOT NULL,
        TIME VARCHAR(20) NOT NULL,
        PRIMARY KEY (ID)
        );
        "

    try:
        cursor.execute(sql) ##for create a table
    except Exception as ex:

```

```

print(ex) #

if subb=="":
    err_screen_for_subject()
else:
    sb.destroy()
    MFW = tk.Tk()
    MFW.iconbitmap('AMS.ico')
    MFW.title("Manually attendance of "+ str(subb))
    MFW.geometry('880x470')
    MFW.configure(background='snow')

def del_errsc2():
    errsc2.destroy()

def err_screen1():
    global errsc2
    errsc2 = tk.Tk()
    errsc2.geometry('330x100')
    errsc2.iconbitmap('AMS.ico')
    errsc2.title('Warning!!')
    errsc2.configure(background='snow')
    Label(errsc2, text='Please enter Student & Enrollment!!!', fg='red', bg='white',
          font=('times', 16, ' bold ')).pack()
    Button(errsc2, text='OK', command=del_errsc2, fg="black", bg="lawn green",
width=9, height=1,
          activebackground="Red", font=('times', 15, ' bold ')).place(x=90, y=50)

def testVal(inStr, acttyp):
    if acttyp == '1': # insert
        if not inStr.isdigit():
            return False
        return True

    ENR = tk.Label(MFW, text="Enter Enrollment", width=15, height=2, fg="white",
bg="blue2",
                  font=('times', 15, ' bold '))
    ENR.place(x=30, y=100)

    STU_NAME = tk.Label(MFW, text="Enter Student name", width=15, height=2,
fg="white", bg="blue2",
                      font=('times', 15, ' bold '))
    STU_NAME.place(x=30, y=200)

global ENR_ENTRY

```

```

ENR_ENTRY = tk.Entry(MFW, width=20, validate='key', bg="yellow", fg="red",
font=('times', 23, ' bold '))
ENR_ENTRY['validatecommand'] = (ENR_ENTRY.register(testVal), '%P', '%d')
ENR_ENTRY.place(x=290, y=105)

def remove_enr():
    ENR_ENTRY.delete(first=0, last=22)

STUDENT_ENTRY = tk.Entry(MFW, width=20, bg="yellow", fg="red",
font=('times', 23, ' bold '))
STUDENT_ENTRY.place(x=290, y=205)

def remove_student():
    STUDENT_ENTRY.delete(first=0, last=22)

####get important variable
def enter_data_DB():
    ENROLLMENT = ENR_ENTRY.get()
    STUDENT = STUDENT_ENTRY.get()
    if ENROLLMENT=="":
        err_screen1()
    elif STUDENT=="":
        err_screen1()
    else:
        time = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
        Hour, Minute, Second = time.split(":")
        Insert_data = "INSERT INTO " + DB_table_name + "
(ID,ENROLLMENT,NAME,DATE,TIME) VALUES (0, %s, %s, %s,%s)"
        VALUES = (str(ENROLLMENT), str(STUDENT), str(Date), str(time))
        try:
            cursor.execute(Insert_data, VALUES)
        except Exception as e:
            print(e)
        ENR_ENTRY.delete(first=0, last=22)
        STUDENT_ENTRY.delete(first=0, last=22)

def create_csv():
    import csv
    cursor.execute("select * from " + DB_table_name + ";")
    csv_name='C:/Users/kusha/PycharmProjects/Attendace          managemnt
system/Attendance/Manually Attendance/'+DB_table_name+'.csv'
    with open(csv_name, "w") as csv_file:
        csv_writer = csv.writer(csv_file)
        csv_writer.writerow([i[0] for i in cursor.description]) # write headers
        csv_writer.writerows(cursor)

```

```

O="CSV created Successfully"
Notifi.configure(text=O, bg="Green", fg="white", width=33, font=('times', 19,
'bold'))
    Notifi.place(x=180, y=380)
import csv
import tkinter
root = tkinter.Tk()
root.title("Attendance of " + subb)
root.configure(background='snow')
with open(csv_name, newline="") as file:
    reader = csv.reader(file)
    r = 0

    for col in reader:
        c = 0
        for row in col:
            # i've added some styling
            label = tkinter.Label(root, width=13, height=1, fg="black", font=('times',
13, ' bold '),
                                bg="lawn green", text=row, relief=tkinter.RIDGE)
            label.grid(row=r, column=c)
            c += 1
            r += 1
        root.mainloop()

Notifi = tk.Label(MFW, text="CSV created Successfully", bg="Green", fg="white",
width=33,
                    height=2, font=('times', 19, 'bold'))

clear_enroll = tk.Button(MFW, text="Clear", command=remove_enr, fg="black",
bg="deep pink", width=10,
                        height=1,
                        activebackground="Red", font=('times', 15, ' bold '))
clear_enroll.place(x=690, y=100)

clear_student = tk.Button(MFW, text="Clear", command=remove_student,
fg="black", bg="deep pink", width=10,
                        height=1,
                        activebackground="Red", font=('times', 15, ' bold '))
clear_student.place(x=690, y=200)

DATA_SUB = tk.Button(MFW, text="Enter Data",command=enter_data_DB,
fg="black", bg="lime green", width=20,
                    height=2,

```

```

        activebackground="Red", font=('times', 15, ' bold '))
DATA_SUB.place(x=170, y=300)

MAKE_CSV = tk.Button(MFW, text="Convert to CSV",command=create_csv,
fg="black", bg="red", width=20,
        height=2,
        activebackground="Red", font=('times', 15, ' bold '))
MAKE_CSV.place(x=570, y=300)

def attf():
    import subprocess
    subprocess.Popen(r'explorer /select,"C:\Users\kusha\PycharmProjects\Attendace
managemnt system\Attendance\Manually Attendance\-----Check attendance-----"')

    attf = tk.Button(MFW, text="Check Sheets",command=attf,fg="black" ,bg="lawn
green" ,width=12 ,height=1 ,activebackground = "Red" ,font=('times', 14, ' bold '))
    attf.place(x=730, y=410)

MFW.mainloop()

SUB = tk.Label(sb, text="Enter Subject", width=15, height=2, fg="white", bg="blue2",
font=('times', 15, ' bold '))
SUB.place(x=30, y=100)

global SUB_ENTRY

SUB_ENTRY = tk.Entry(sb, width=20, bg="yellow", fg="red", font=('times', 23, ' bold
'))
SUB_ENTRY.place(x=250, y=105)

fill_manual_attendance = tk.Button(sb, text="Fill
Attendance",command=fill_attendance, fg="white", bg="deep pink", width=20, height=2,
        activebackground="Red", font=('times', 15, ' bold '))
fill_manual_attendance.place(x=250, y=160)
sb.mainloop()

##For clear textbox
def clear():
    txt.delete(first=0, last=22)

def clear1():
    txt2.delete(first=0, last=22)
def del_sc1():
    sc1.destroy()

```

```

def err_screen():
    global sc1
    sc1 = tk.Tk()
    sc1.geometry('300x100')
    sc1.iconbitmap('AMS.ico')
    sc1.title('Warning!!')
    sc1.configure(background='snow')
    Label(sc1,text='Enrollment & Name required!!!',fg='red',bg='white',font=('times', 16, '
bold')).pack()
    Button(sc1,text='OK',command=del_sc1,fg="black",bg="lawn green",width=9
,height=1,activebackground="Red",font=('times', 15, ' bold')).place(x=90,y= 50)

##Error screen2
def del_sc2():
    sc2.destroy()
def err_screen1():
    global sc2
    sc2 = tk.Tk()
    sc2.geometry('300x100')
    sc2.iconbitmap('AMS.ico')
    sc2.title('Warning!!')
    sc2.configure(background='snow')
    Label(sc2,text='Please enter your subject name!!!',fg='red',bg='white',font=('times', 16, '
bold')).pack()
    Button(sc2,text='OK',command=del_sc2,fg="black",bg="lawn green",width=9
,height=1,activebackground="Red",font=('times', 15, ' bold')).place(x=90,y= 50)

####For take images for datasets
def take_img():
    l1 = txt.get()
    l2 = txt2.get()
    if l1 == "":
        err_screen()
    elif l2 == "":
        err_screen()
    else:
        try:
            detector = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
            Enrollment = txt.get()
            Name = txt2.get()
            sampleNum = 0
            while (True):
                img_resp = urlopen(url)
                imgnp = np.asarray(bytearray(img_resp.read()), dtype="uint8")
                img = cv2.imdecode(imgnp, -1)

```



```

gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
faces = detector.detectMultiScale(gray, 1.3, 5)
for (x, y, w, h) in faces:
    cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)
    # incrementing sample number
    sampleNum = sampleNum + 1
    # saving the captured face in the dataset folder
    cv2.imwrite("TrainingImage/ " + Name + "." + Enrollment + '.' +
str(sampleNum) + ".jpg",
                gray[y:y + h, x:x + w])
    cv2.imshow('Frame', img)
    # wait for 100 milliseconds
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
    # break if the sample number is morethan 100
    elif sampleNum > 70:
        break
## cam.release()
cv2.destroyAllWindows()
ts = time.time()
Date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')
Time = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
row = [Enrollment, Name, Date, Time]
with open('StudentDetails\\StudentDetails.csv', 'a+') as csvFile:
    writer = csv.writer(csvFile, delimiter=',')
    writer.writerow(row)
    csvFile.close()
res = "Images Saved for Enrollment : " + Enrollment + " Name : " + Name
Notification.configure(text=res, bg="SpringGreen3", width=50, font=('times', 18,
'bold'))
Notification.place(x=250, y=400)
except FileNotFoundError as F:
    f = 'Student Data already exists'
    Notification.configure(text=f, bg="Red", width=21)
    Notification.place(x=450, y=400)

```

####for choose subject and fill attendance

```

def subjectchoose():
    def Fillattendances():
        sub=tx.get()
        now = time.time() ###For calculate seconds of video
        future = now + 20
        if time.time() < future:
            if sub == "":

```

```

err_screen1()
else:
    recognizer = cv2.face.LBPHFaceRecognizer_create() #
cv2.createLBPHFaceRecognizer()
    try:
        recognizer.read("TrainingImageLabel\Trainer.yml")
    except:
        e = 'Model not found,Please train model'
        Notifica.configure(text=e, bg="red", fg="black", width=33, font=('times', 15,
'bold'))
        Notifica.place(x=20, y=250)

harcascadePath = "haarcascade_frontalface_default.xml"
faceCascade = cv2.CascadeClassifier(harcascadePath)
df = pd.read_csv("StudentDetails\StudentDetails.csv")
cam = cv2.VideoCapture(0)
font = cv2.FONT_HERSHEY_SIMPLEX
col_names = ['Enrollment', 'Name', 'Date', 'Time']
attendance = pd.DataFrame(columns=col_names)
while True:
    img_resp = urlopen(url)
    imgnp = np.asarray(bytearray(img_resp.read()), dtype="uint8")
    im = cv2.imdecode(imgnp, -1)
    gray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
    faces = faceCascade.detectMultiScale(gray, 1.2, 5)
    for (x, y, w, h) in faces:
        global Id

        Id, conf = recognizer.predict(gray[y:y + h, x:x + w])
        if (conf < 70):
            print(conf)
            global Subject
            global aa
            global date
            global timeStamp
            Subject = tx.get()
            ts = time.time()
            date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')
            timeStamp =
datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
            aa = df.loc[df['Enrollment'] == Id]['Name'].values
            global tt
            tt = str(Id) + "-" + aa
            En = '15624031' + str(Id)
            attendance.loc[len(attendance)] = [Id, aa, date, timeStamp]

```

```

cv2.rectangle(im, (x, y), (x + w, y + h), (0, 260, 0), 7)
cv2.putText(im, str(tt), (x + h, y), font, 1, (255, 255, 0), 4)

else:
    Id = 'Unknown'
    tt = str(Id)
    cv2.rectangle(im, (x, y), (x + w, y + h), (0, 25, 255), 7)
    cv2.putText(im, str(tt), (x + h, y), font, 1, (0, 25, 255), 4)
if time.time() > future:
    break

attendance = attendance.drop_duplicates(['Enrollment'], keep='first')
cv2.imshow('Filling attendance..', im)
key = cv2.waitKey(30) & 0xff
if key == 27:
    break

ts = time.time()
date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')
timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
Hour, Minute, Second = timeStamp.split(":")
fileName = "Attendance/" + Subject + "_" + date + "_" + Hour + "-" + Minute +
"-" + Second + ".csv"
attendance = attendance.drop_duplicates(['Enrollment'], keep='first')
print(attendance)
attendance.to_csv(fileName, index=False)

##Create table for Attendance
date_for_DB = datetime.datetime.fromtimestamp(ts).strftime('%Y_%m_%d')
DB_Table_name = str( Subject + "_" + date_for_DB + "_Time_" + Hour + "_" +
Minute + "_" + Second)
import pymysql.connections

###Connect to the database
try:
    global cursor
    connection = pymysql.connect(host='localhost', user='root', password="",
db='Face_reco_fill')
    cursor = connection.cursor()
except Exception as e:
    print(e)

sql = "CREATE TABLE " + DB_Table_name + ""
(ID INT NOT NULL AUTO_INCREMENT,
ENROLLMENT varchar(100) NOT NULL,

```

```

NAME VARCHAR(50) NOT NULL,
DATE VARCHAR(20) NOT NULL,
TIME VARCHAR(20) NOT NULL,
PRIMARY KEY (ID)
);
"""

####Now enter attendance in Database
insert_data = "INSERT INTO " + DB_Table_name + "
(ID,ENROLLMENT,NAME,DATE,TIME) VALUES (0, %s, %s, %s,%s)"
VALUES = (str(Id), str(aa), str(date), str(timeStamp))
try:
    cursor.execute(sql) ##for create a table
    cursor.execute(insert_data, VALUES)##For insert data into table
except Exception as ex:
    print(ex) #

M = 'Attendance filled Successfully'
Notifica.configure(text=M, bg="Green", fg="white", width=33, font=('times', 15,
'bold'))
Notifica.place(x=20, y=250)

cam.release()
cv2.destroyAllWindows()

import csv
import tkinter
root = tkinter.Tk()
root.title("Attendance of " + Subject)
root.configure(background='snow')
cs = 'C:/Users/kusha/PycharmProjects/Attendace managemnt system/' + fileName
with open(cs, newline="") as file:
    reader = csv.reader(file)
    r = 0

    for col in reader:
        c = 0
        for row in col:
            # i've added some styling
            label = tkinter.Label(root, width=8, height=1, fg="black", font=('times',
15, 'bold '),
                                bg="lawn green", text=row, relief=tkinter.RIDGE)
            label.grid(row=r, column=c)
            c += 1
        r += 1
    root.mainloop()

```

```

print(attendance)

###windo is frame for subject chooser
windo = tk.Tk()
windo.iconbitmap('AMS.ico')
windo.title("Enter subject name...")
windo.geometry('580x320')
windo.configure(background='snow')
Notifica = tk.Label(windo, text="Attendance filled Successfully", bg="Green",
fg="white", width=33,
                    height=2, font=('times', 15, 'bold'))

def Attf():
    import subprocess
    subprocess.Popen(r'explorer /select,"C:\Users\kusha\PycharmProjects\Attendace
managemnt system\Attendance\-----Check attendance-----"')

    attf = tk.Button(windo, text="Check Sheets",command=Attf,fg="black" ,bg="lawn
green" ,width=12 ,height=1 ,activebackground = "Red" ,font=('times', 14, ' bold '))
    attf.place(x=430, y=255)

    sub = tk.Label(windo, text="Enter Subject", width=15, height=2, fg="white",
bg="blue2", font=('times', 15, ' bold '))
    sub.place(x=30, y=100)

    tx = tk.Entry(windo, width=20, bg="yellow", fg="red", font=('times', 23, ' bold '))
    tx.place(x=250, y=105)

    fill_a = tk.Button(windo, text="Fill Attendance", fg="white",command=Fillattendances,
bg="deep pink", width=20, height=2,
                    activebackground="Red", font=('times', 15, ' bold '))
    fill_a.place(x=250, y=160)
    windo.mainloop()

def admin_panel():
    win = tk.Tk()
    win.iconbitmap('AMS.ico')
    win.title("LogIn")
    win.geometry('880x420')
    win.configure(background='snow')

    def log_in():
        username = un_entr.get()
        password = pw_entr.get()

```

```

if username == 'kushal' :
    if password == 'kushal14320':
        win.destroy()
        import csv
        import tkinter
        root = tkinter.Tk()
        root.title("Student Details")
        root.configure(background='snow')

        cs = 'C:/Users/kusha/PycharmProjects/Attendace managemnt
system/StudentDetails/StudentDetails.csv'
        with open(cs, newline='') as file:
            reader = csv.reader(file)
            r = 0

            for col in reader:
                c = 0
                for row in col:
                    # i've added some styling
                    label = tkinter.Label(root, width=8, height=1, fg="black", font=('times',
15, 'bold '),
                                bg="lawn green", text=row, relief=tkinter.RIDGE)
                    label.grid(row=r, column=c)
                    c += 1
                r += 1
            root.mainloop()
        else:
            valid = 'Incorrect ID or Password'
            Nt.configure(text=valid, bg="red", fg="black", width=38, font=('times', 19,
'bold'))
            Nt.place(x=120, y=350)

        else:
            valid = 'Incorrect ID or Password'
            Nt.configure(text=valid, bg="red", fg="black", width=38, font=('times', 19, 'bold'))
            Nt.place(x=120, y=350)

Nt = tk.Label(win, text="Attendance filled Successfully", bg="Green", fg="white",
width=40,
            height=2, font=('times', 19, 'bold'))
# Nt.place(x=120, y=350)

un = tk.Label(win, text="Enter username", width=15, height=2, fg="white", bg="blue2",
            font=('times', 15, 'bold '))

```

```

un.place(x=30, y=50)

pw = tk.Label(win, text="Enter password", width=15, height=2, fg="white", bg="blue2",
               font=('times', 15, ' bold '))
pw.place(x=30, y=150)

def c00():
    un_entr.delete(first=0, last=22)

un_entr = tk.Entry(win, width=20, bg="yellow", fg="red", font=('times', 23, ' bold '))
un_entr.place(x=290, y=55)

def c11():
    pw_entr.delete(first=0, last=22)

pw_entr = tk.Entry(win, width=20, show="*", bg="yellow", fg="red", font=('times', 23, '
bold '))
pw_entr.place(x=290, y=155)

c0 = tk.Button(win, text="Clear", command=c00, fg="black", bg="deep pink",
               width=10, height=1,
               activebackground="Red", font=('times', 15, ' bold '))
c0.place(x=690, y=55)

c1 = tk.Button(win, text="Clear", command=c11, fg="black", bg="deep pink",
               width=10, height=1,
               activebackground="Red", font=('times', 15, ' bold '))
c1.place(x=690, y=155)

Login = tk.Button(win, text="LogIn", fg="black", bg="lime green", width=20,
                  height=2,
                  activebackground="Red", command=log_in, font=('times', 15, ' bold '))
Login.place(x=290, y=250)
win.mainloop()

####For train the model
def training():
    recognizer = cv2.face.LBPHFaceRecognizer_create()
    global detector
    detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
    try:
        global faces, Id
        faces, Id = getImagesAndLabels("TrainingImage")
    except Exception as e:

```

```

l='please make "TrainingImage" folder & put Images'
Notification.configure(text=l, bg="SpringGreen3", width=50, font=('times', 18, 'bold'))
Notification.place(x=350, y=400)

recognizer.train(faces, np.array(Id))
try:
    recognizer.save("TrainingImageLabel\Trainer.yml")
except Exception as e:
    q='Please make "TrainingImageLabel" folder'
    Notification.configure(text=q, bg="SpringGreen3", width=50, font=('times', 18,
'bold'))
    Notification.place(x=350, y=400)

res = "Model Trained" # +",".join(str(f) for f in Id)
Notification.configure(text=res, bg="SpringGreen3", width=50, font=('times', 18, 'bold'))
Notification.place(x=250, y=400)

def getImagesAndLabels(path):
    imagePath = [os.path.join(path, f) for f in os.listdir(path)]
    # create empty face list
    faceSamples = []
    # create empty ID list
    Ids = []
    # now looping through all the image paths and loading the Ids and the images
    for imagePath in imagePath:
        # loading the image and converting it to gray scale
        pilImage = Image.open(imagePath).convert('L')
        # Now we are converting the PIL image into numpy array
        imageNp = np.array(pilImage, 'uint8')
        # getting the Id from the image

        Id = int(os.path.split(imagePath)[-1].split(".")[1])
        # extract the face from the training image sample
        faces = detector.detectMultiScale(imageNp)
        # If a face is there then append that in the list as well as Id of it
        for (x, y, w, h) in faces:
            faceSamples.append(imageNp[y:y + h, x:x + w])
            Ids.append(Id)
    return faceSamples, Ids

window.grid_rowconfigure(0, weight=1)
window.grid_columnconfigure(0, weight=1)
window.iconbitmap('AMS.ico')

def on_closing():

```



```

from tkinter import messagebox
if messagebox.askokcancel("Quit", "Do you want to quit?"):
    window.destroy()
window.protocol("WM_DELETE_WINDOW", on_closing)

message = tk.Label(window, text="Face-Recognition-Based-Attendance-Management-System", bg="cyan", fg="black", width=50,
                    height=3, font=('times', 30, 'italic bold '))

message.place(x=80, y=20)

Notification = tk.Label(window, text="All things good", bg="Green", fg="white",
                        width=15,
                        height=3, font=('times', 17, 'bold'))

lbl = tk.Label(window, text="Enter Enrollment", width=20, height=2, fg="black",
               bg="deep pink", font=('times', 15, ' bold '))
lbl.place(x=200, y=200)

def testVal(inStr,acttyp):
    if acttyp == '1': #insert
        if not inStr.isdigit():
            return False
        return True

txt = tk.Entry(window, validate="key", width=20, bg="yellow", fg="red", font=('times', 25,
' bold '))
txt['validatecommand'] = (txt.register(testVal), '%P', '%d')
txt.place(x=550, y=210)

lbl2 = tk.Label(window, text="Enter Name", width=20, fg="black", bg="deep pink",
                height=2, font=('times', 15, ' bold '))
lbl2.place(x=200, y=300)

txt2 = tk.Entry(window, width=20, bg="yellow", fg="red", font=('times', 25, ' bold '))
txt2.place(x=550, y=310)

clearButton = tk.Button(window, text="Clear",command=clear,fg="black" ,bg="deep pink" ,width=10 ,height=1 ,activebackground = "Red" ,font=('times', 15, ' bold '))
clearButton.place(x=950, y=210)

clearButton1 = tk.Button(window, text="Clear",command=clear1,fg="black" ,bg="deep pink" ,width=10 ,height=1 ,activebackground = "Red" ,font=('times', 15, ' bold '))
clearButton1.place(x=950, y=310)

```

```
AP = tk.Button(window, text="Check Register students",command=admin_panel,fg="black",bg="cyan",width=19,height=1,activebackground = "Red",font=('times', 15, ' bold '))
AP.place(x=990, y=410)
```

```
takeImg = tk.Button(window, text="Take Images",command=take_img,fg="white",bg="blue2",width=20,height=3,activebackground = "Red",font=('times', 15, ' bold '))
takeImg.place(x=90, y=500)
```

```
trainImg = tk.Button(window, text="Train Images",fg="black",command=training,bg="lawn green",width=20,height=3,activebackground = "Red",font=('times', 15, ' bold '))
trainImg.place(x=390, y=500)
```

```
FA = tk.Button(window, text="Automatic Attendace",fg="white",command=subjectchoose,bg="blue2",width=20,height=3,activebackground = "Red",font=('times', 15, ' bold '))
FA.place(x=690, y=500)
```

```
quitWindow = tk.Button(window, text="Manually Fill Attendance",command=manually_fill,fg="black",bg="lawn green",width=20,height=3,activebackground = "Red",font=('times', 15, ' bold '))
quitWindow.place(x=990, y=500)
```

```
window.mainloop()
```

TESTING.PY FILE

```
import cv2
from urllib.request import urlopen
import numpy as np

recognizer = cv2.face.LBPHFaceRecognizer_create()
recognizer.read("TrainingImageLabel/trainer.yml")
cascadePath = "haarcascade_frontalface_default.xml"
faceCascade = cv2.CascadeClassifier(cascadePath)
font = cv2.FONT_HERSHEY_SIMPLEX
url = r'http://192.168.43.53/cam-hi.jpg'
##cam = cv2.VideoCapture(0)

while True:
    img_resp = urlopen(url)
    imgnp = np.asarray(bytearray(img_resp.read()), dtype="uint8")
    im = cv2.imdecode(imgnp, -1)
    ##ret, im =cam.read()
    gray=cv2.cvtColor(im,cv2.COLOR_BGR2GRAY)
```

```

faces=faceCascade.detectMultiScale(gray, 1.2,5)
for(x,y,w,h) in faces:
    Id, conf = recognizer.predict(gray[y:y+h,x:x+w])

    ## else:
    ##     Id="Unknown"
    # cv2.rectangle(im, (x-22,y-90), (x+w+22, y-22), (0,255,0), -1)
    cv2.rectangle(im, (x, y), (x + w, y + h), (0, 260, 0), 7)
    cv2.putText(im, str(Id), (x,y-40),font, 2, (255,255,255), 3)

    # cv2.putText(im, str(Id), (x + h, y), font, 1, (0, 260, 0), 2)
cv2.imshow('im',im)
if cv2.waitKey(10) & 0xFF==ord('q'):
    break
cam.release()
cv2.destroyAllWindows()

```

TRAINING.PY FILE

```

import cv2,os
import numpy as np
from PIL import Image
#
recognizer = cv2.face.LBPHFaceRecognizer_create()
#recognizer=cv2.face.createFisherFaceRecognizer_create()
detector= cv2.CascadeClassifier("haarcascade_frontalface_default.xml")

def getImagesAndLabels(path):
    #get the path of all the files in the folder
    imagePath=[os.path.join(path,f) for f in os.listdir(path)]
    #create empty face list
    faceSamples=[]
    #create empty ID list
    Ids=[]
    #now looping through all the image paths and loading the Ids and the images
    for imagePath in imagePath:
        #loading the image and converting it to gray scale
        pilImage=Image.open(imagePath).convert('L')
        #Now we are converting the PIL image into numpy array
        imageNp=np.array(pilImage,'uint8')
        #getting the Id from the image

        Id = int(os.path.split(imagePath)[-1].split(".")[1])
        # extract the face from the training image sample

```

```
faces=detector.detectMultiScale(imageNp)
#If a face is there then append that in the list as well as Id of it
for (x,y,w,h) in faces:
    faceSamples.append(imageNp[y:y+h,x:x+w])
    Ids.append(Id)
return faceSamples,Ids
```

```
faces,Ids = getImagesAndLabels('TrainingImage')
recognizer.train(faces, np.array(Ids))
recognizer.save('TrainingImageLabel/trainer.yml')
```

	BILL OF MATERIALS	
Component	Quantity	Price
ESP32 CAM MODULE	1	700
FTDI MODULE	1	300
BREAD BOARD	1	60
JUMPER CABLE	10	50
MICRO USB CABLE	1	70
CAMERA CONATINER	1	20
POWER BANK	1	1000
TOTAL		₹ 2200