EX.NO:20   **AUTOMATED ATTENDANCE USING AI - FACE RECOGNITION**

DATE :21/09/2021   **MINI PROJECT**

OBJECTIVE:

The main objective is to make an attendance list using AI-Facial Recognition**.**

DESCRIPTION:

Over the past decade face detection and recognition have transcended from esoteric to popular areas of research in computer vision and one of the better and successful applications of image analysis and algorithm based understanding. Because of the intrinsic nature of the problem, computer vision is not only a computer science area of research, but also the object of neuro-scientific and psychological studies also,mainly because of the general opinion that advances in computer image processing and understanding research will provide insights into how our brain work and vice versa.A general statement of the face recognition problem (in computer vision) can be formulated as follows: given still or video images of a scene, identify or verify one or more persons in the scene using a stored database of faces.

SOFTWARE USED:

PYCHARM COMMUNITY EDITION OR VISUAL STUDIO CODE

MODULES USED:

PIP ,OPEN-CV, CMAKE, DLIB ,FACE-RECOGNITION,etc.,

REFERENCE:

Article : https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78

Course Website : https://www.computervision.zone/courses/face-attendance/

CONCEPTS AND LOGIC :

The original image is turned into a HOG representation that captures the major features of the image regardless of image brightnesss.Encode a picture using the HOG algorithm to create a simplified version of the image. Using this simplified image, find the part of the image that most looks like a generic HOG encoding of a face.
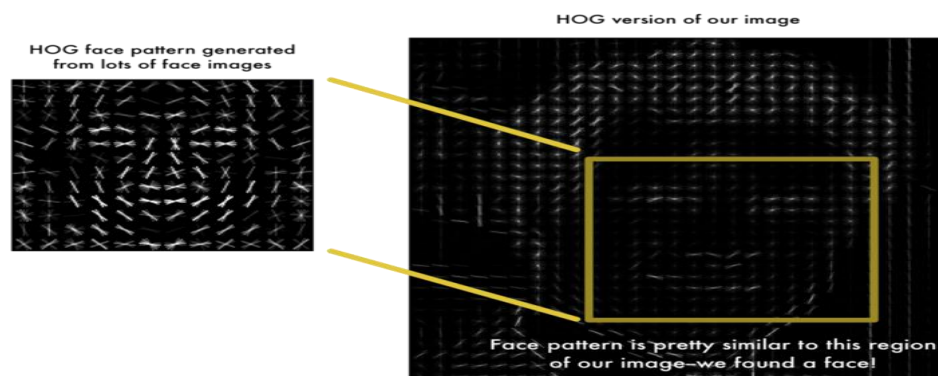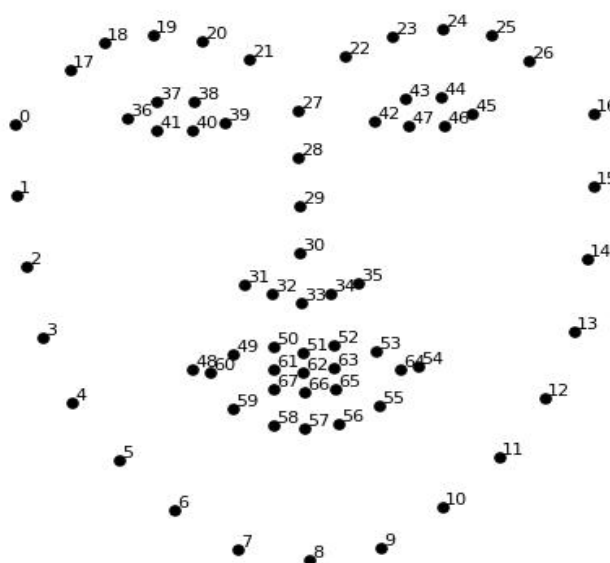

Figure out the pose of the face by finding the main landmarks in the face. Once we find those landmarks, use them to warp the image so that the eyes and mouth are centered.The 68 landmarks we will locate on every face.


Pass the centered face image through a neural network that knows how to measure features of the face. Save those 128 measurements. Looking at all the faces we've measured in the past, see which person has the closest measurements to our face's measurements. That's our match!
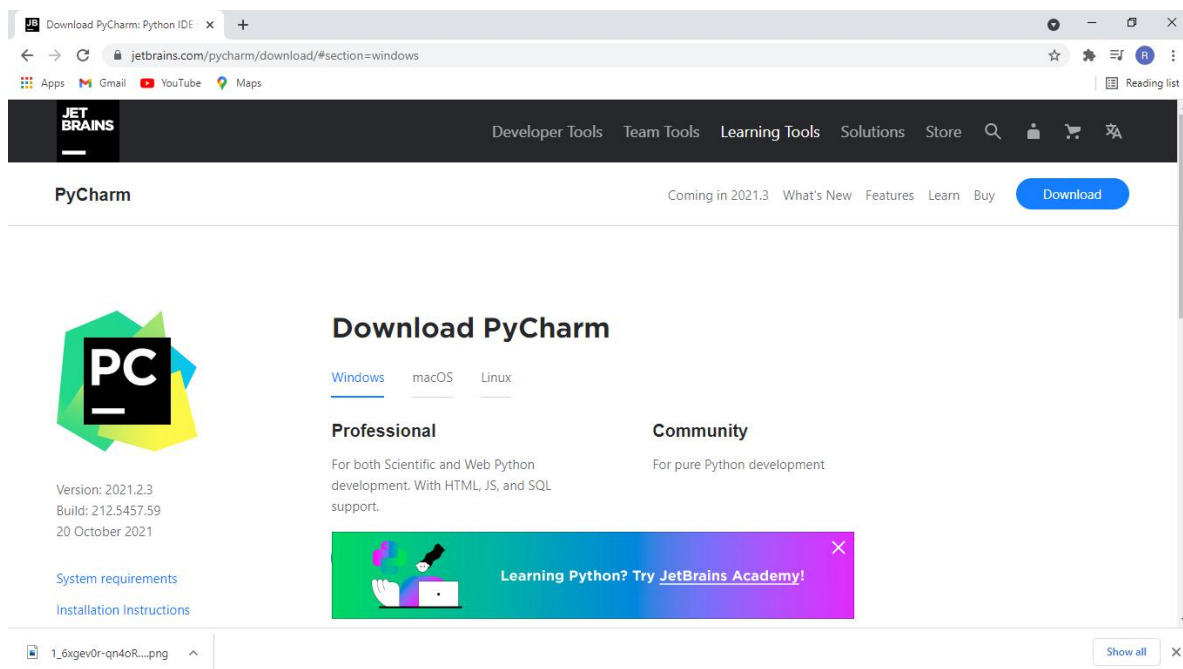
.

SOFTWARES AND MODULES :

Make sure to install Any version of Python 3.6, so that the program doesn't get any error.
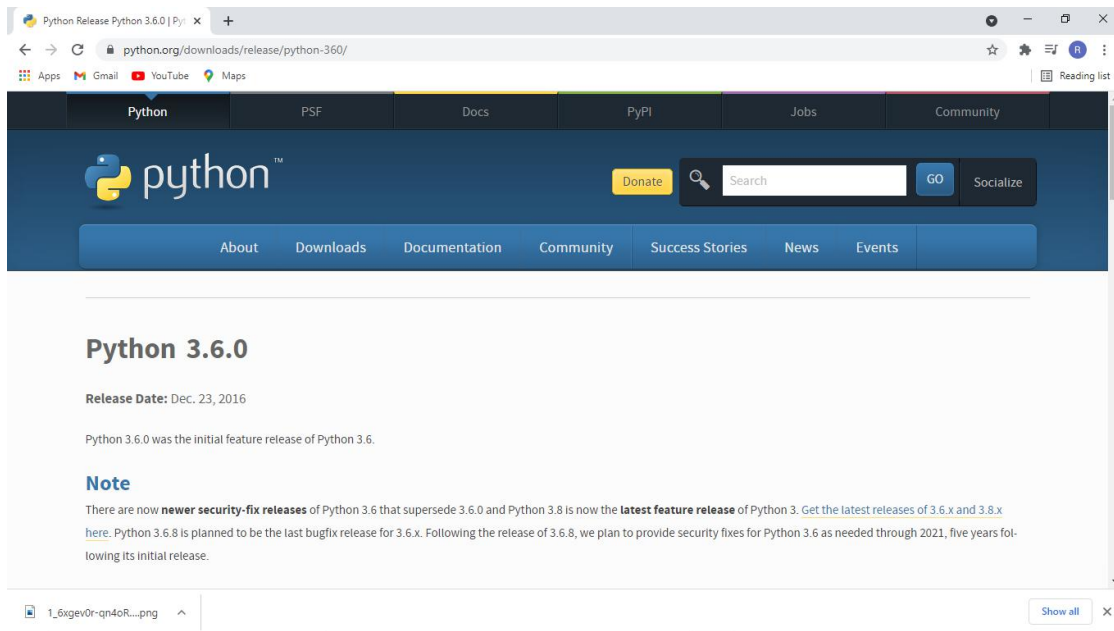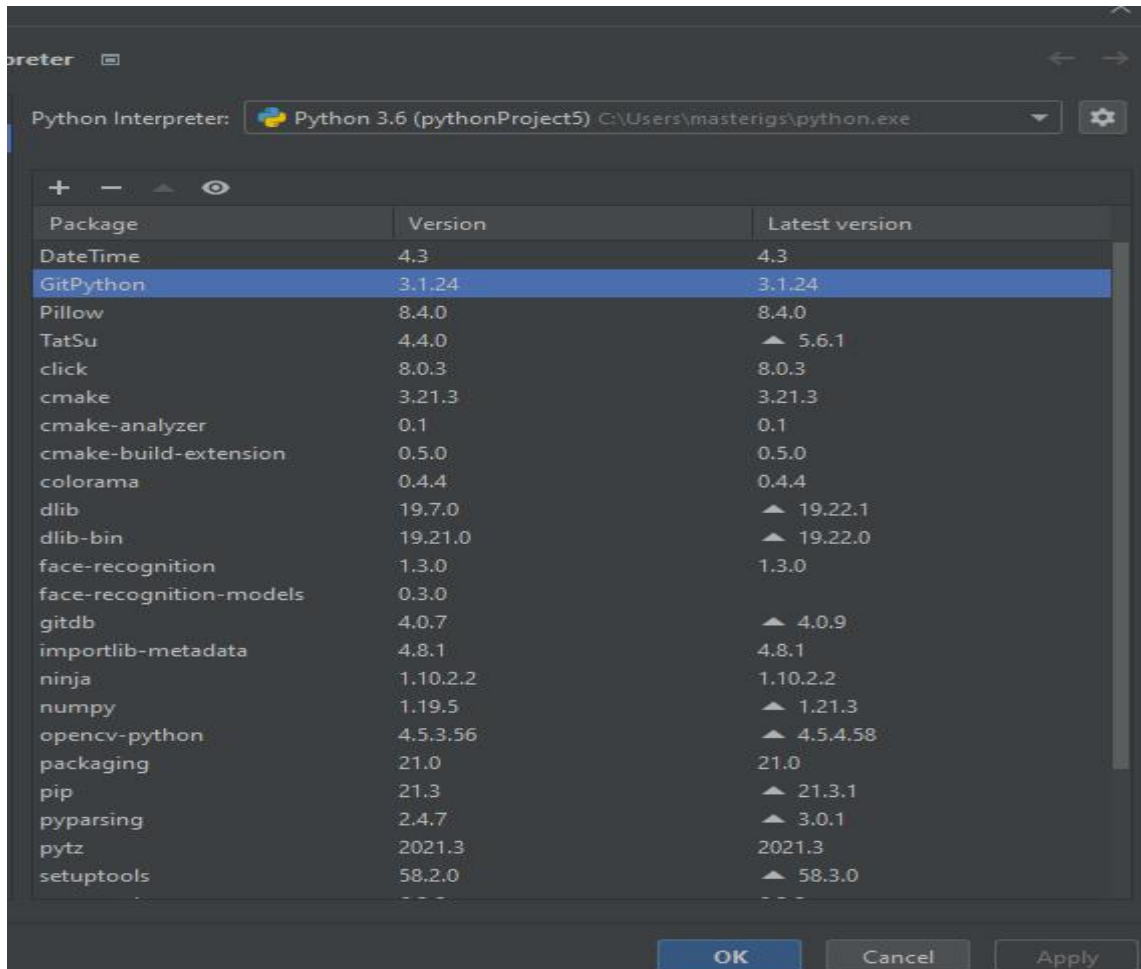

Now, install Pycharm Community Edition,

Now in Pycharm software download the required dependencies,

To install the required dependencies , we can use another method by opening the terminal and selecting command prompt in Pycharm Software,
Give command one by one ,

COMMANDS:

pip install opencv - python
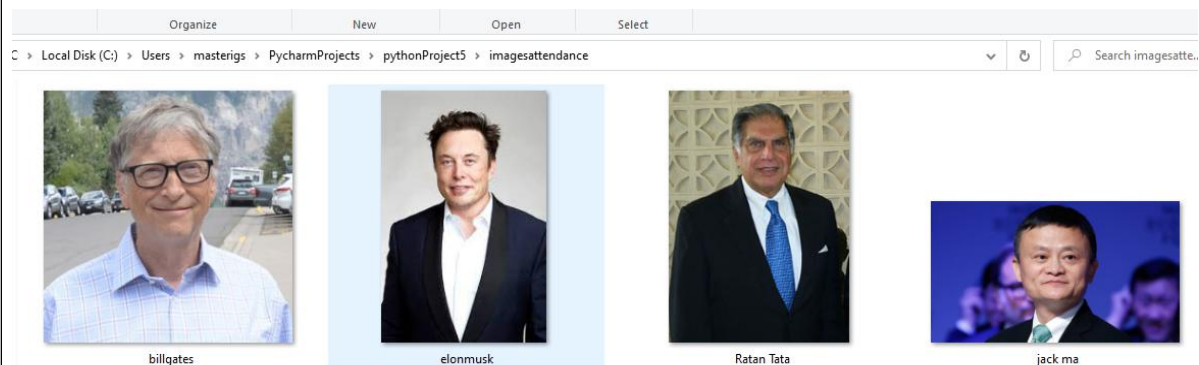
pip install cmake

pip install dlib
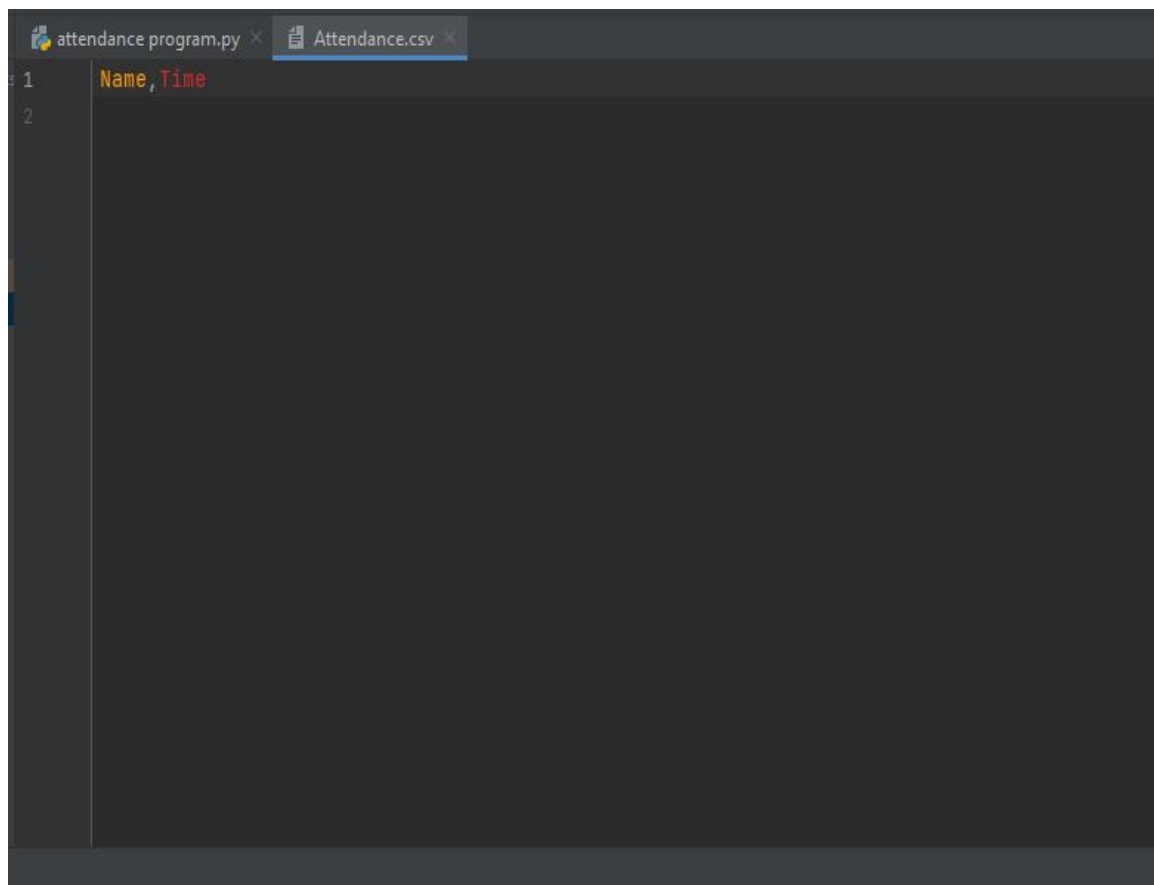
pip install face-recognition

Now , Follow the steps ,

Create an folder and add the images of the student database ,here I have used the following images as example

Also create an .csv File so that the attendance list to be used in the program.

PROGRAM:

```python
#To import the reliant dependant modules
import cv2
import numpy as np
import face_recognition
from datetime import datetime
import os

#To create an folder of the class students images and add its path
path= 'imagesattendance'
images= []
classNames = []
classList = os.listdir(path)
#print(classList)

#Adding the students images into an list
for cls in classList:
    curImg = cv2.imread(f'{path}/{cls}')
    images.append(curImg)
    classNames.append(os.path.splitext(cls)[0])
print(classNames)
```

```python
    #To find the encodings of the images
def findEncodings(images):
    encodeList = []
    for img in images:
        img =cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
        encode = face_recognition.face_encodings(img)[0]
        #print(encode)
        encodeList.append(encode)
    return encodeList

#To create an attendance list(.csv) of students with time
def markAttendance(name):
    with open('Attendance.csv','r+') as f:
        classDataList = f.readlines()
        nameList = []
        for line in classDataList:
            entry = line.split(',')
            nameList.append(entry[0])
        if name not in nameList:
            now = datetime.now()
            dtString = now.strftime('%H:%M:%S')
            f.writelines(f'\n{name},{dtString}')
```

```python
encodeListKnown = findEncodings(images)
print('Encoding completed')
#To capture the image using webcam and to find the encodings of the captured images
cap = cv2.VideoCapture(0)
while True:
    success,img = cap.read()
    imgS = cv2.resize(img,(0,0),None,0.25,0.25)
    imgS = cv2.cvtColor(imgS, cv2.COLOR_BGR2RGB)
    faceCurFrame = face_recognition.face_locations(imgS)
    encodeCurFrame = face_recognition.face_encodings(imgS,faceCurFrame)
    for encodeFace,faceLoc in zip(encodeCurFrame,faceCurFrame):
        matches = face_recognition.compare_faces(encodeListKnown,encodeFace)
        faceDis = face_recognition.face_distance(encodeListKnown,encodeFace)
        #print(faceDis)
        matchIndex = np.argmin(faceDis)
        if matches[matchIndex]:
            name = classNames[matchIndex].upper()
            #print(name)
            y1,x2,y2,x1 = faceLoc
            y1, x2, y2, x1 = y1*4,x2*4,y2*4,x1*4
            cv2.rectangle(img,(x1,y1),(x2,y2),(0,255,0),2)
            cv2.rectangle(img, (x1, y2-35), (x2, y2), (0, 255, 0),cv2.FILLED)
            cv2.putText(img,name,(x1+6,y2-6),cv2.FONT_HERSHEY_COMPLEX,1,(255,255,255),2)
            markAttendance(name)
    cv2.imshow('WEBCAM',img)
    cv2.waitKey(1)
```

EXECUTION AND OUTPUT :

WEBCAM CAPTURES:
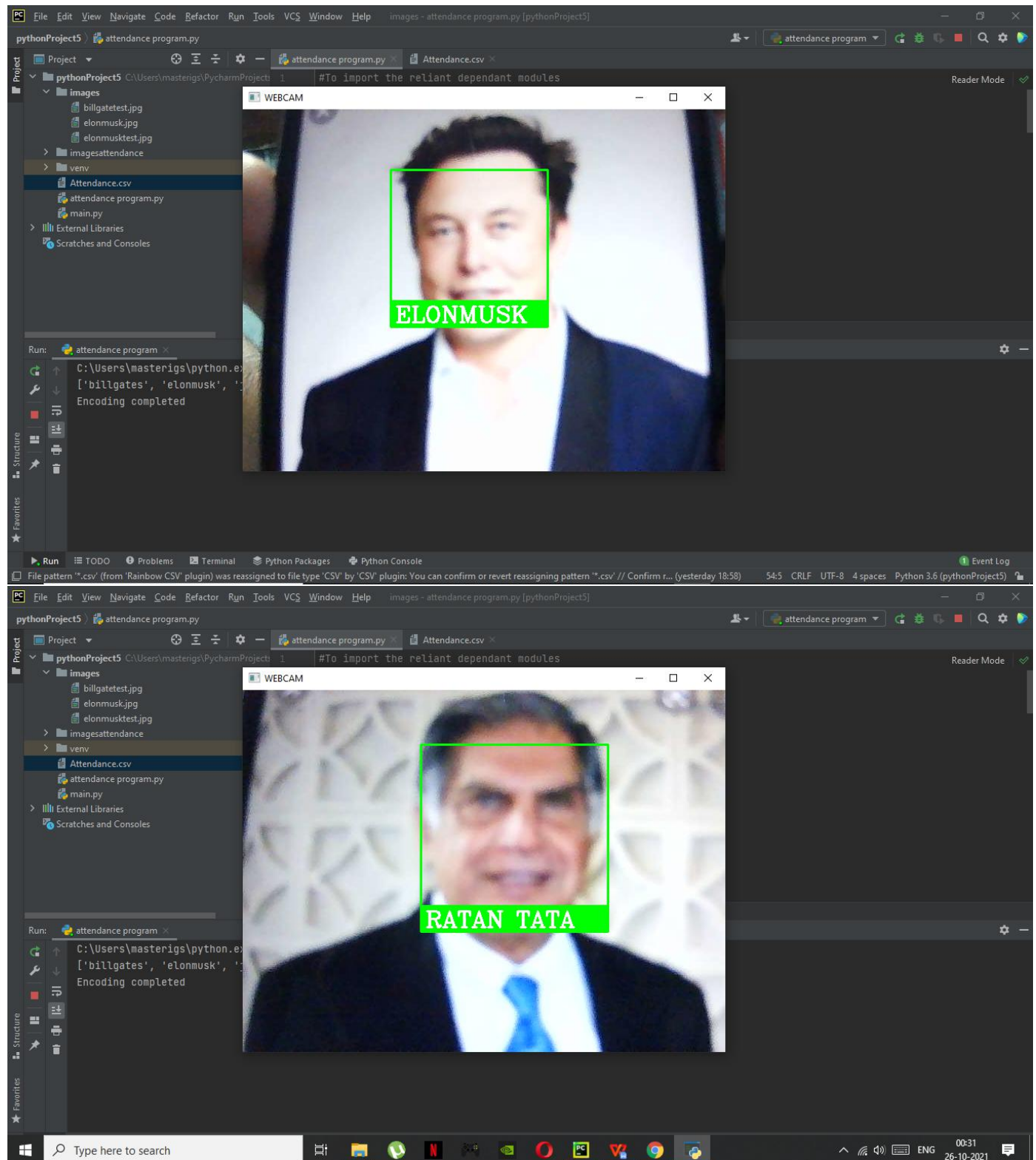
FILLED ATTENDANCE SHEET:

The csv File is filled with name and time ,


```
Name,Time
ELONMUSK,12:28:09
RATAN TATA,12:30:57
```

Now you can open this file in excel for your convenience,

	A	B	C	D	E	F	G
1	Name	Time					
2	ELONMUSK	12:28:09					
3	RATAN TAT	12:30:57					

RESULT:

The program to mark attendence of students using AI - Face recognition is executed Successfully.