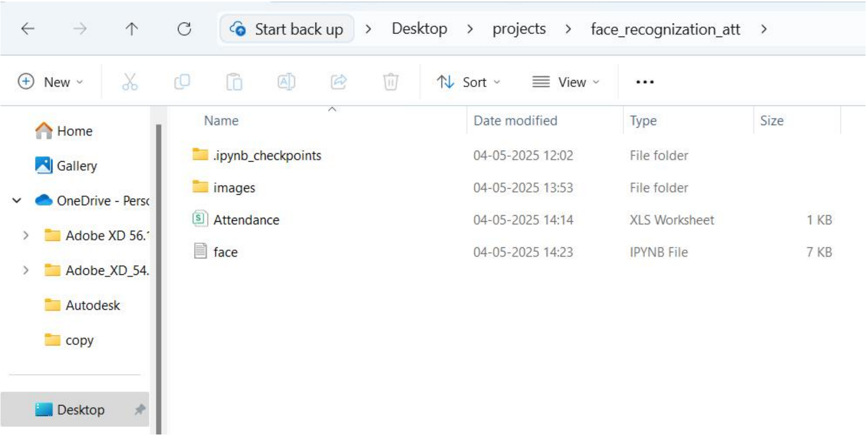
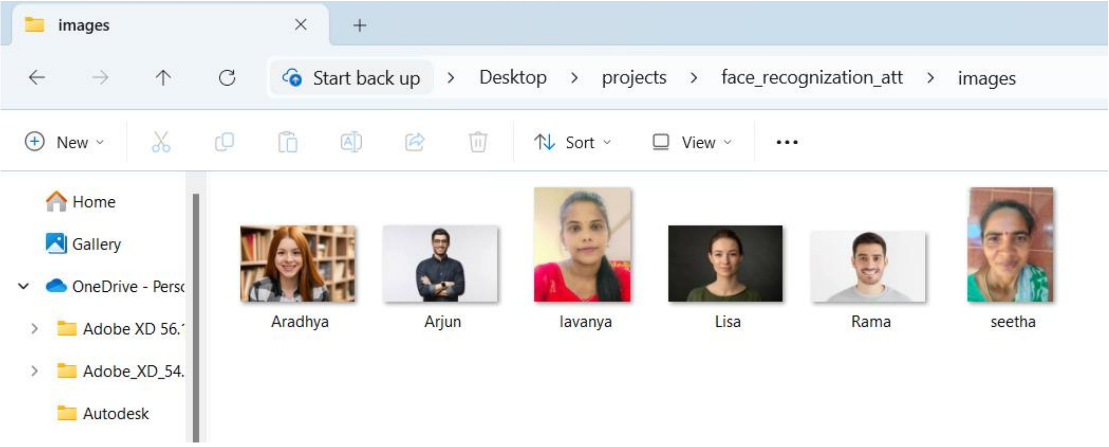
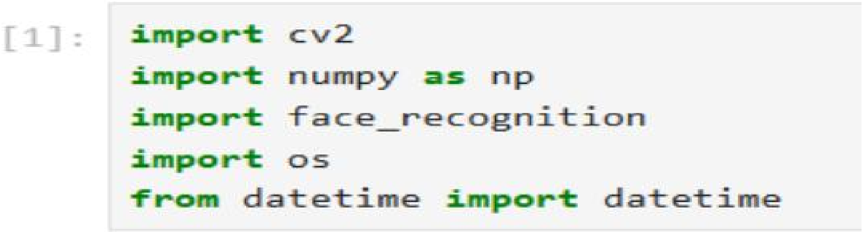
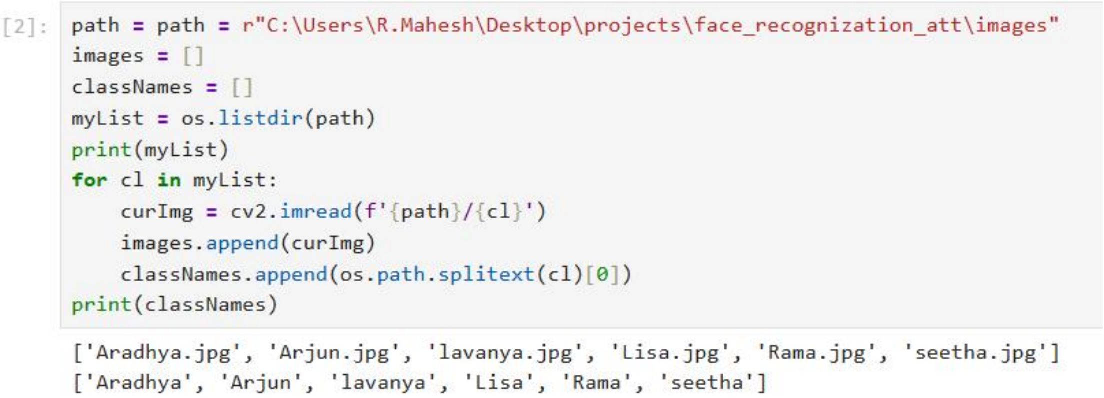


Name	Date modified	Type	Size
.ipynb_checkpoints	04-05-2025 12:02	File folder	
images	04-05-2025 13:53	File folder	
Attendance	04-05-2025 14:14	XLS Worksheet	1 KB
face	04-05-2025 14:23	IPYNB File	7 KB



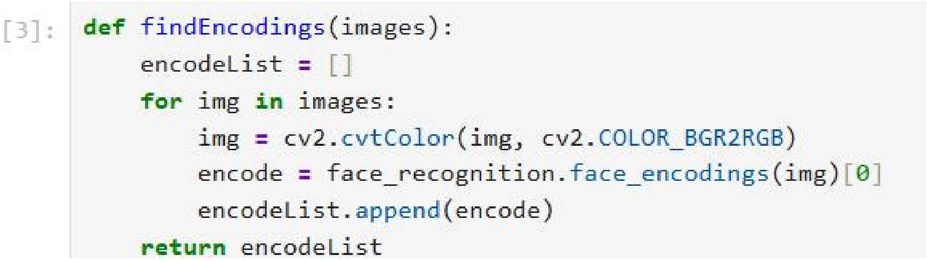

[1]:

```
import cv2
import numpy as np
import face_recognition
import os
from datetime import datetime
```


[2]:

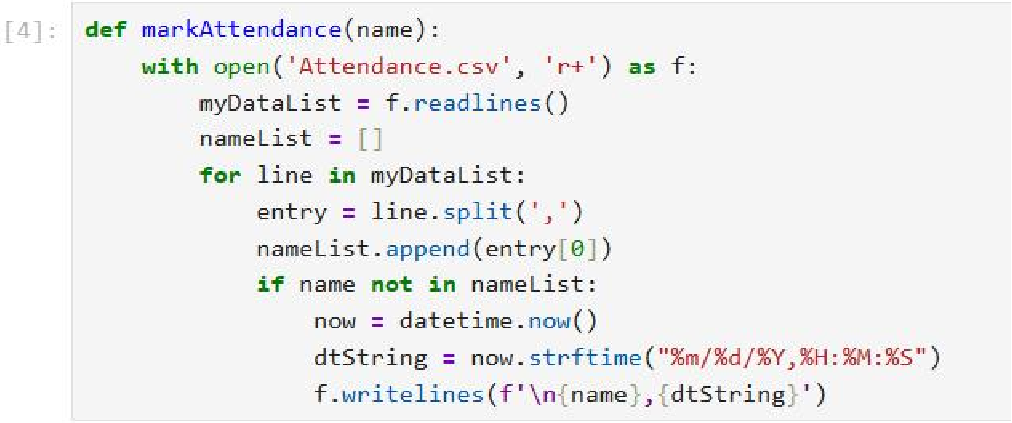
```
path = path = r"C:\Users\R.Mahesh\Desktop\projects\face_recognization_att\images"
images = []
classNames = []
myList = os.listdir(path)
print(myList)
for cl in myList:
    curImg = cv2.imread(f'{path}/{cl}')
    images.append(curImg)
    classNames.append(os.path.splitext(cl)[0])
print(classNames)
```

```
['Aradhya.jpg', 'Arjun.jpg', 'lavanya.jpg', 'Lisa.jpg', 'Rama.jpg', 'seetha.jpg']
['Aradhya', 'Arjun', 'lavanya', 'Lisa', 'Rama', 'seetha']
```



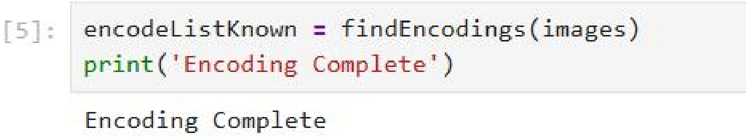
[3]:

```
def findEncodings(images):
    encodeList = []
    for img in images:
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        encode = face_recognition.face_encodings(img)[0]
        encodeList.append(encode)
    return encodeList
```



[4]:

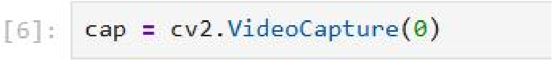
```
def markAttendance(name):
    with open('Attendance.csv', 'r+') as f:
        myDataList = f.readlines()
        nameList = []
        for line in myDataList:
            entry = line.split(',')
            nameList.append(entry[0])
            if name not in nameList:
                now = datetime.now()
                dtString = now.strftime("%m/%d/%Y,%H:%M:%S")
                f.writelines(f'\n{name},{dtString}')
```



[5]:

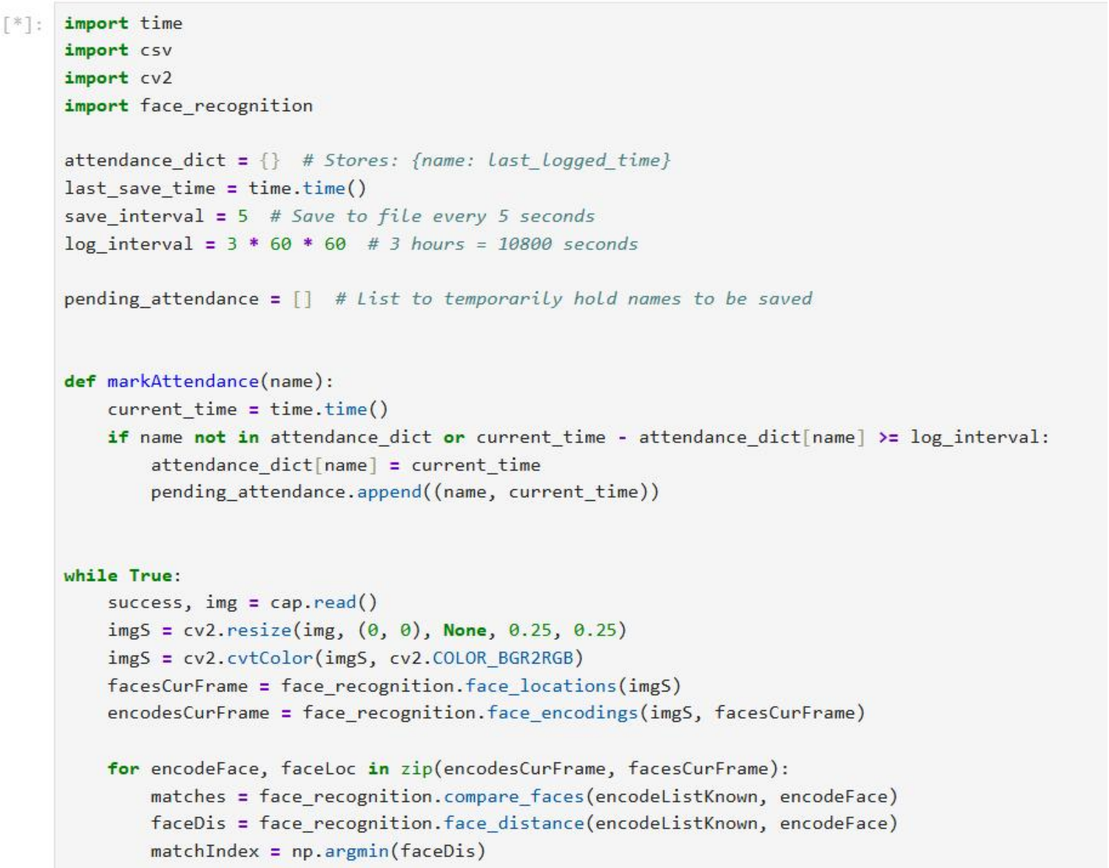
```
encodeListKnown = findEncodings(images)
print('Encoding Complete')
```

```
Encoding Complete
```



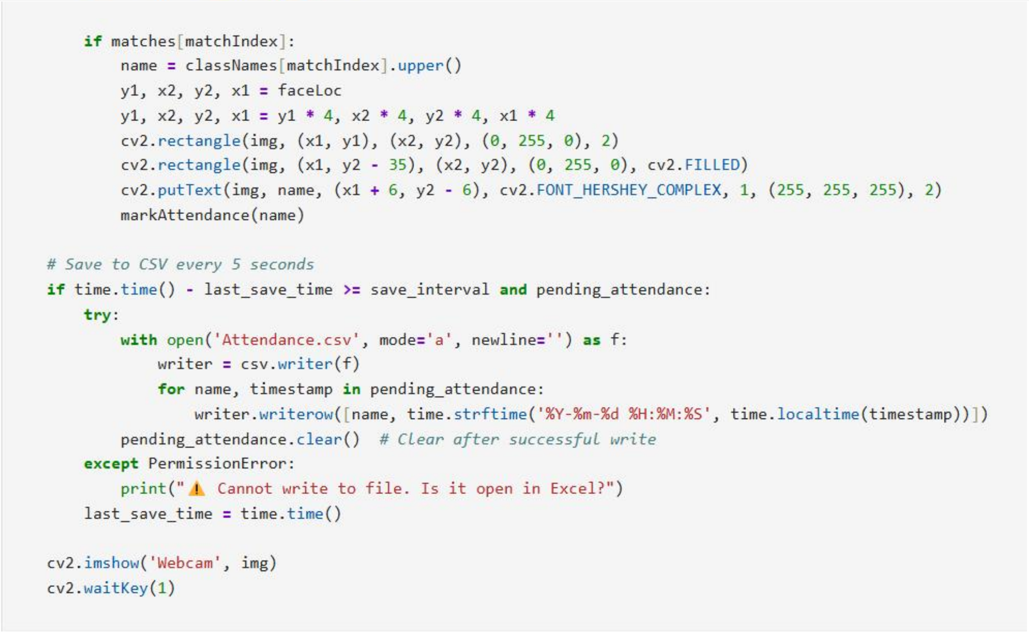
[6]:

```
cap = cv2.VideoCapture(0)
```

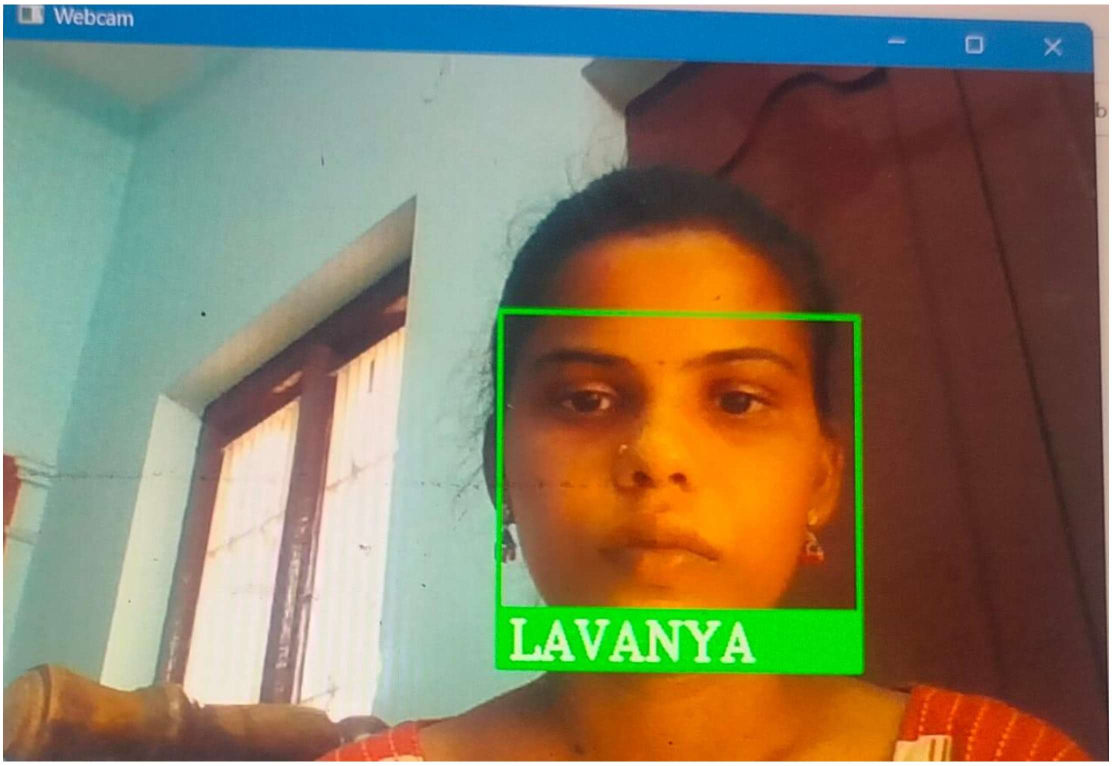


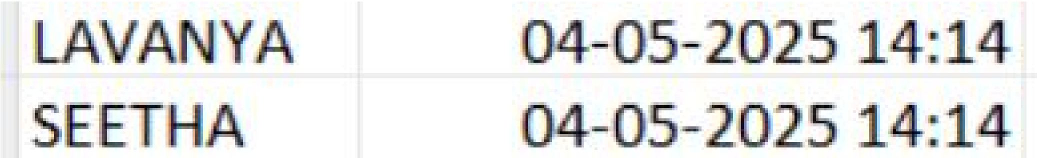
[*]:

```
import time
import csv
import cv2
import face_recognition

attendance_dict = {}  # Stores: {name: last_logged_time}
last_save_time = time.time()
save_interval = 5  # Save to file every 5 seconds
log_interval = 3 * 60 * 60  # 3 hours = 10800 seconds

pending_attendance = []  # List to temporarily hold names to be saved


def markAttendance(name):
    current_time = time.time()
    if name not in attendance_dict or current_time - attendance_dict[name] >= log_interval:
        attendance_dict[name] = current_time
        pending_attendance.append((name, current_time))


while True:
    success, img = cap.read()
    imgS = cv2.resize(img, (0, 0), None, 0.25, 0.25)
    imgS = cv2.cvtColor(imgS, cv2.COLOR_BGR2RGB)
    facesCurFrame = face_recognition.face_locations(imgS)
    encodesCurFrame = face_recognition.face_encodings(imgS, facesCurFrame)

    for encodeFace, faceLoc in zip(encodesCurFrame, facesCurFrame):
        matches = face_recognition.compare_faces(encodeListKnown, encodeFace)
        faceDis = face_recognition.face_distance(encodeListKnown, encodeFace)
        matchIndex = np.argmin(faceDis)
```



```
        if matches[matchIndex]:
            name = classNames[matchIndex].upper()
            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)

    # Save to CSV every 5 seconds
    if time.time() - last_save_time >= save_interval and pending_attendance:
        try:
            with open('Attendance.csv', mode='a', newline='') as f:
                writer = csv.writer(f)
                for name, timestamp in pending_attendance:
                    writer.writerow([name, time.strftime('%Y-%m-%d %H:%M:%S', time.localtime(timestamp))])
            pending_attendance.clear()  # Clear after successful write
        except PermissionError:
            print("⚠️ Cannot write to file. Is it open in Excel?")
        last_save_time = time.time()

    cv2.imshow('Webcam', img)
    cv2.waitKey(1)
```





LAVANYA	04-05-2025 14:14
SEETHA	04-05-2025 14:14