In [1]:

```
import face_recognition
import cv2
import numpy as np
import csv
import os
from datetime import datetime
from pymongo import MongoClient
from gridfs import GridFS
from PIL import Image
import pandas as pd
```

In [2]:

```
Connection_string="mongodb://localhost:27017"
client=MongoClient(Connection_string)
client.drop_database('online_attendance')
```

In [3]:

```
dbname=client['online_attendance']
fs = GridFS(dbname,collection='students_pics')
```

In [4]:

```
folder_path = 'C:/Users/hp/Pictures/AI_Project'
class_names=[]
for filename in os.listdir(folder_path):
    if filename.endswith('.jpg') or filename.endswith('.png'):
        with open(os.path.join(folder_path, filename), 'rb') as f:
            data = f.read()
            existing_file = fs.find_one({'filename': filename})
            if existing_file is not None:
                print(f'File {filename} exists in the GridFS collection.')
            else:
                class_names.append(os.path.splitext(filename)[0].upper())
                fs.put(data, filename=filename)
                print(f'{filename} inserted successfully.')

print(class_names)
print("Upload Completed")
```

```
ankita.jpg inserted successfully.
bill1.jpg inserted successfully.
bill2.jpg inserted successfully.
elon1.jpg inserted successfully.
elon2.jpg inserted successfully.
mona_lisa1.jpg inserted successfully.
mona_lisa2.jpg inserted successfully.
tata1.jpg inserted successfully.
tata2.jpg inserted successfully.
['ANKITA', 'BILL1', 'BILL2', 'ELON1', 'ELON2', 'MONA_LISA1', 'MONA_LISA2', 'TATA1',
'TATA2']
Upload Completed
```

In [5]:

```
students = class_names.copy()
images=[]
for file in fs.find():
    file_data = file.read()
    np_data = np.frombuffer(file_data, np.uint8)
    cur_image = cv2.imdecode(np_data, cv2.IMREAD_COLOR)
    images.append(cur_image)
```

In [6]:

```
print(len(images))
```

```
9
```

In [7]:

```
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
```

In [8]:

```
encodeListKnown=findEncodings(images)
print(len(encodeListKnown))
```

```
9
```

In [9]:

```
from datetime import datetime
now = datetime.now()
current_date = now.strftime("%Y-%m-%d")
f = open(current_date+'.csv','w+',newline = '')
header = ['Name_ofStudent','Time_of_entry']
lnwriter = csv.writer(f)
lnwriter.writerow(header)
```

Out[9]:

```
30
```

In [10]:

```
cap=cv2.VideoCapture(0)
while True:
    success,img = cap.read()
    frame = cv2.resize(img,(0,0),None,fx=0.25,fy=0.25)
    frame=cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    face_locations = face_recognition.face_locations(frame)
    face_encodings = face_recognition.face_encodings(frame,face_locations)

    for encodeFace,LocationFace in zip(face_encodings,face_locations):
        matches = face_recognition.compare_faces(encodeListKnown,encodeFace)
        print(matches)
        face_distance = face_recognition.face_distance(encodeListKnown,encodeFace)
        #print(face_distance)
        best_match_index = np.argmin(face_distance)
        if matches[best_match_index]:
            name = class_names[best_match_index].upper()
            print(name)
            y1,x2,y2,x1=LocationFace
            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_SIMPLEX,1,(255,255,255),
            if name in students:
                students.remove(name)
                current_time = now.strftime("%H-%M-%S")
                lnwriter.writerow([name,current_time])


    cv2.imshow("attendence system",img)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
```

```
cap.release()
cv2.destroyAllWindows()
f.close()
```

```
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
```

```
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[False, True, True, False, False, False, False, False, False]
BILL1
[True, False, False, False, False, False, False, False, False]
ANKITA
[False, True, True, False, False, False, False, False, False]
BILL1
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
```

```
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[False, False, False, False, False, False, False, True, True]
TATA1
[True, False, False, False, False, False, False, False, False]
ANKITA
[False, False, False, False, False, False, False, True, True]
TATA1
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
[True, False, False, False, False, False, False, False, False]
ANKITA
```

In [11]:

```python
today = now.strftime("%Y-%m-%d")
collection_name = 'Students-present-on-' + today
collection = dbname[collection_name]
with open(current_date+'.csv', 'r') as csvfile:
    reader = csv.DictReader(csvfile)
    for row in reader:
        collection.insert_one(row)
        print("insert completed")
```

```
insert completed
insert completed
insert completed
```

In [ ]:

```python

```