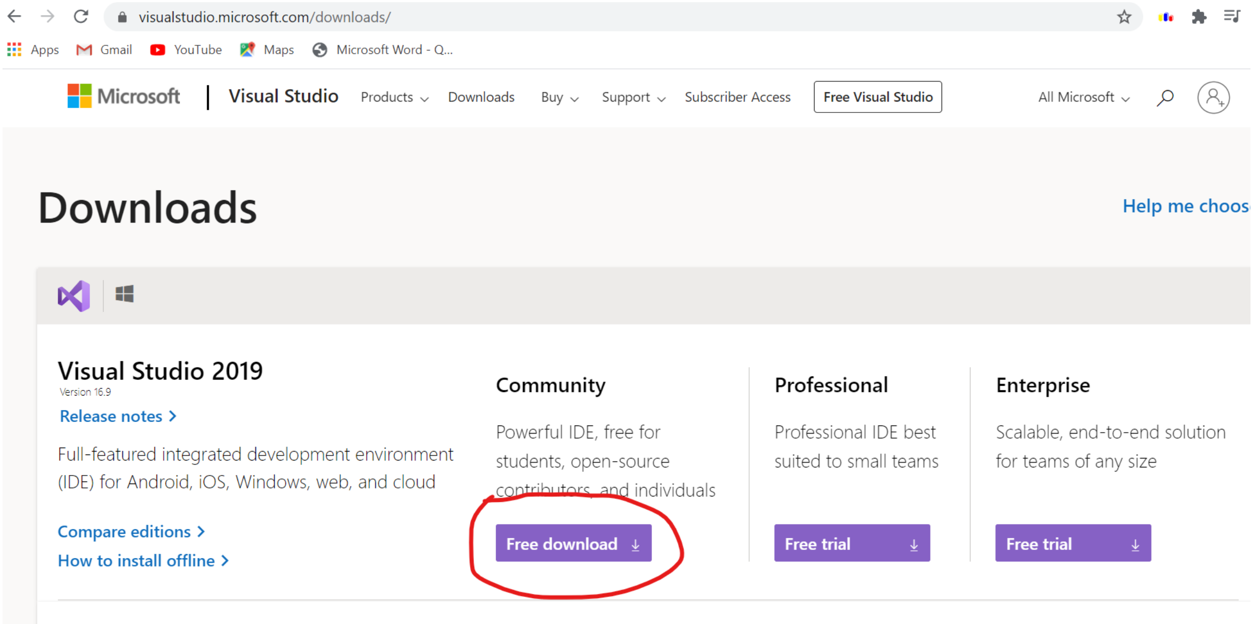
User manual to use face recognition software made by group 1:

This software was designed to work in PyCharm so the first step is to install PyCharm.
Second, we need a c compiler for our software to work, so we need to download the community version of visual studio (free):



Downloads

Visual Studio 2019
Version 16.9

Full-featured integrated development environment (IDE) for Android, iOS, Windows, web, and cloud

Community
Powerful IDE, free for students, open-source contributors, and individuals

Professional
Professional IDE best suited to small teams

Enterprise
Scalable, end-to-end solution for teams of any size

While installing, we will get a lot of options but we only need to check “Desktop Development with C++” box and install.

Create a project in PyCharm

Next we need to install the following in our project in PyCharm for this software to work.

1. cmake
2. dlib 19.18 (mind that I tried with 19.19, 19.20 and 19.21 but none of them worked, only 19.18 did)
3. face-recognition
4. numpy
5. opencv-python

Create a folder named ‘Image Attendance’ (without the quotes) in the same project.
Right click on this folder, open it in explorer and put normal front facing images of every student of the class in this folder.

Create an empty file by the name of ‘Attendance.csv’ (without the quotes) in the same project.
Now double click on it and write “Name,Time” in it.

After having done all the above things, paste the following code and click on run:

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

path = 'ImagesAttendance'
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)

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

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('%H:%M:%S')
            f.writelines(f'\n{name},{dtString}')

encodeListKnown = findEncodings(images)
#print(len(encodeListKnown))
print('Encoding Complete')

cap = cv2.VideoCapture(0)

while True:
    success, img = cap.read()
    img = cv2.flip(img, 1)
    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)
        #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_SIMPLEX, 1, (255, 255, 255), 2)
            markAttendance(name)

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