Aum Amriteshwaryai Namah

Real Time Attendance Monitoring System

## Abstract

This project is based on real time attendance using image processing technique. The objective of our project is to provide real time attendance of students in the class to the faculties. For this we make use of Image Processing using python as a platform.

The camera will capture image of its nearby surrounding and detect the facial part of that image and send this image information to the processing systems. Now as per the algorithm which we are using , will compare the input image with the given data base(set of student images in class ) and search for a matching image. If there is a positive match, then the students attendance is stored to a database.

## Objectives

1. Capture video input from the camera
2. Detect faces in the frame and process the image
3. Compare the processed image with student database
4. Store best fit faces in a new CSV file

## Assumptions and Limitations

1. There is enough lighting to recognize faces.
2. Not more than one face in the frame.
3. The device from which we are taking the input must be set appropriately

System Architecture

1. Block diagram consisting of various modules of the project and its description

Diagram

Description automatically generated

Diagram

Description automatically generated

1. Elaborate the Image processing modules - The algorithm description in one para and its purpose in the project
2. face\_recognition: Its mainly used for finding faces and manipulating them.

Functions used:

* load\_image\_file: Used to load an image into a numpy array.
* face\_encodings: On passing an image, it returns the 128-dimension face encoding for each face in the image.

Application: For encoding all the loaded images.

* face\_locations: Returns an array of bounding boxes of human faces in an image.

Application: for finding a face in the input video source. If found , focus on these locations

* compare\_faces: It compares a list of face encodings against a candidate encoding to see if they match. This returns a list Boolean values.

Application: for checking if the face found in input video matches with any of the faces of the students.

* face\_distance: Given a list of face encodings, compare them to a known face encoding and get a Euclidean distance for each comparison face. The distance tells you how similar the faces are.

It returns a numpy array with the distances.

Application: Using the Boolean values returned by compare\_faces , we calculate the distance for more accuracy. This would work efficiently on large databases.

1. cv2 : It is a library of Python bindings designed to solve computer vision problems.

Functions used:

* VideoCapture: Used for video input from a webcam
* resize: Used for scaling the image.

Application: for reducing the size of input , so that it takes less time to process the image.

1. Numpy: It’s a library mainly used for arrays and has many numerical methods to be used on.

Functions used:

* argmin: Returns indices of the min element of the array in a particular axis

Application: To get the best fitted image. We used this to check with compare\_faces list. If there exist a face, the condition is set to true and stores the name.

Contribution of each member of the team

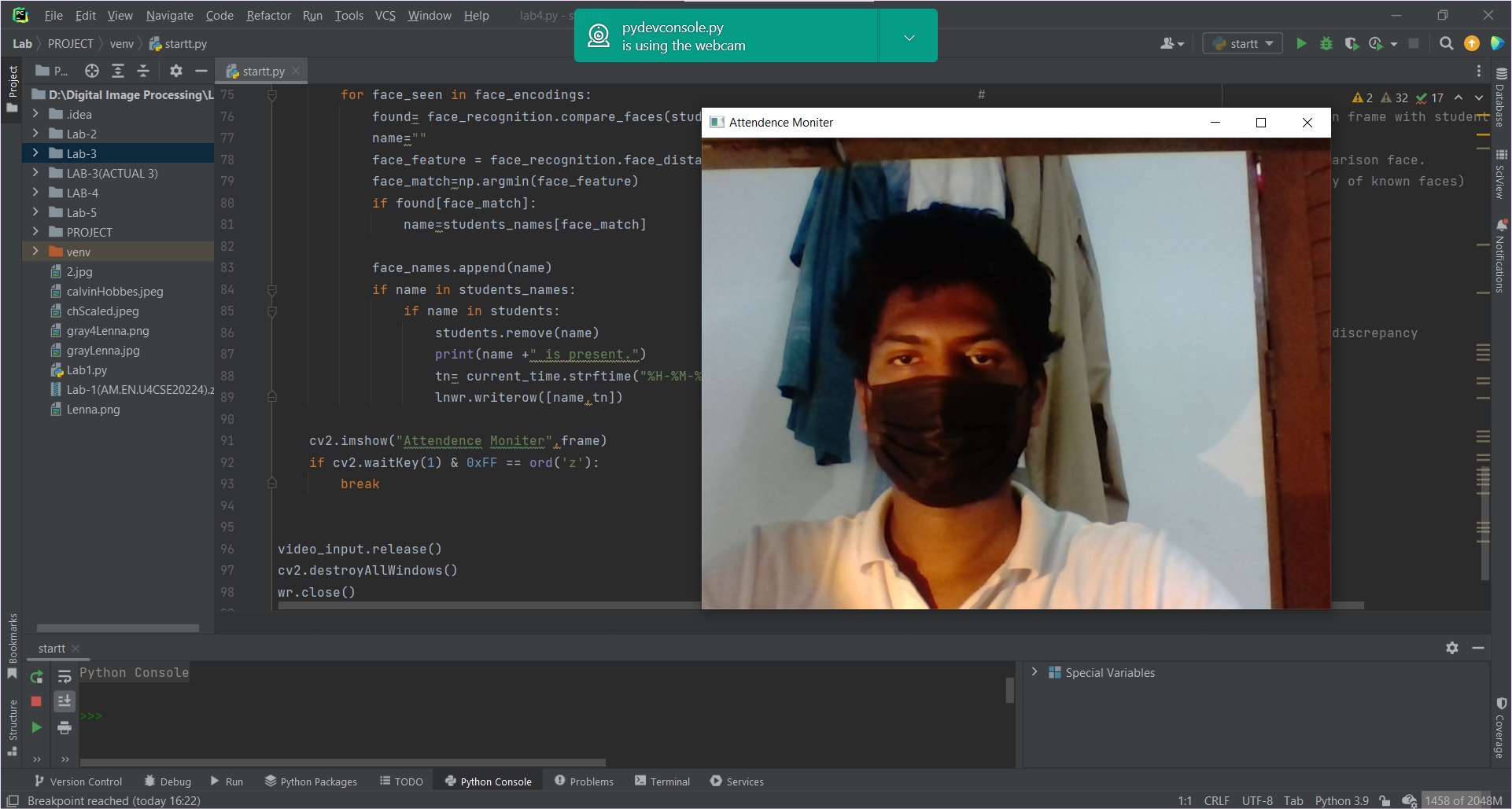
3-4 lines description from each member on their technical contribution - should be obviously related to image processing modules of your solution

Input to the system

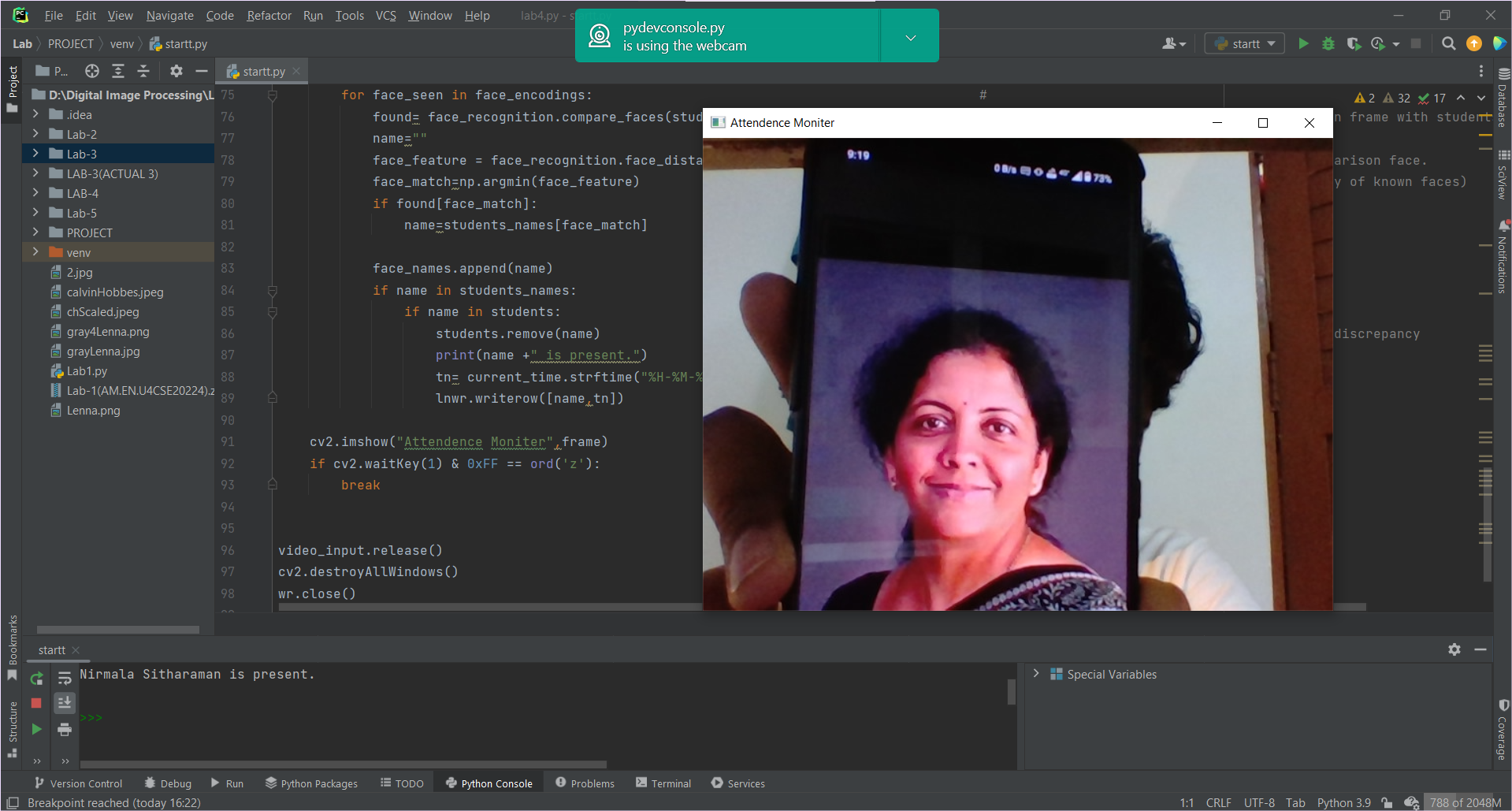
Live video from a webcam.

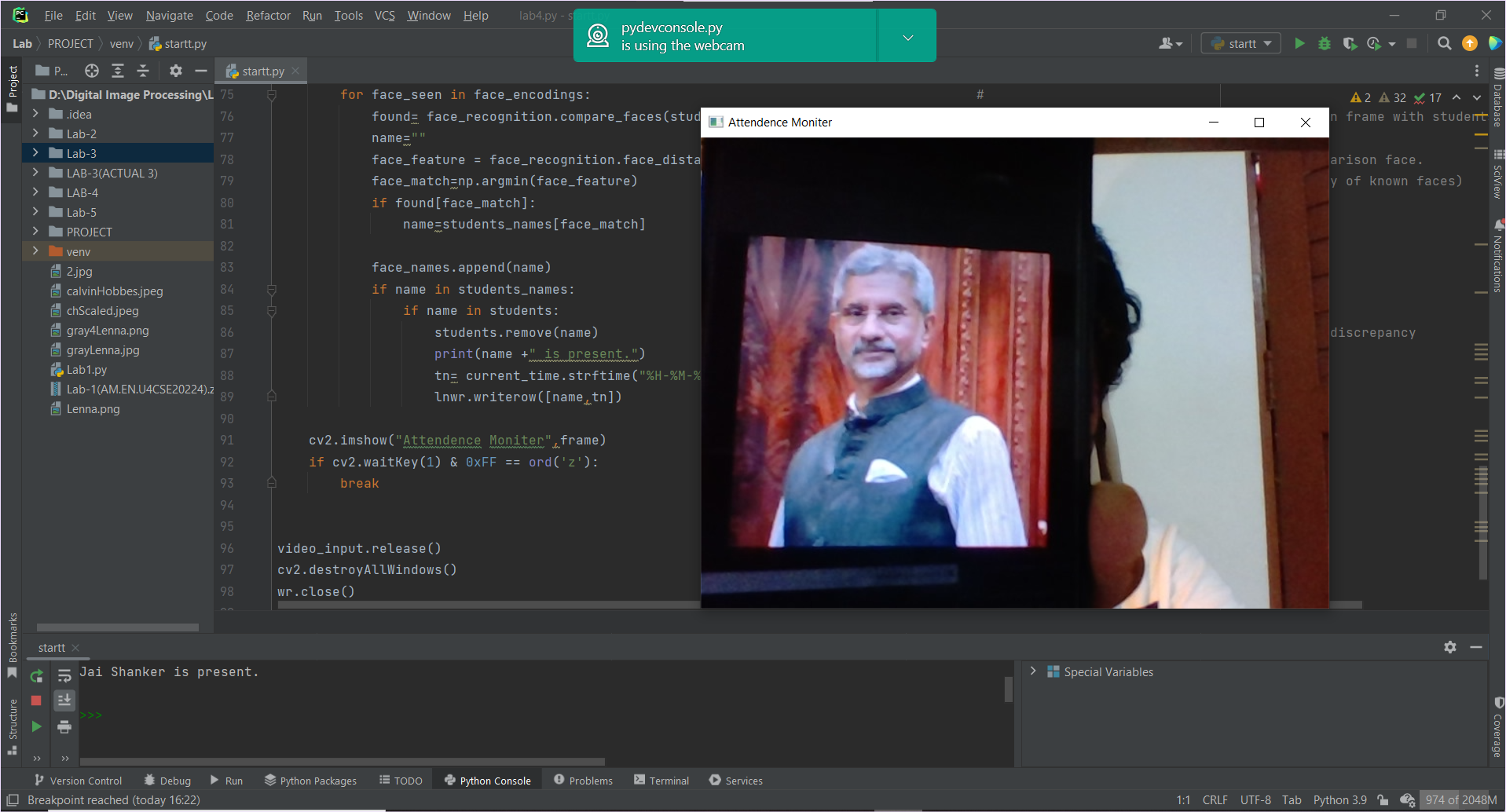
Output

On clicking run option.



Comparing all the students face(***examples***):





Graphical user interface, application

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

Terminal Output:

A screenshot of a computer

Description automatically generated

On clicking ‘z’, Student attendance details are written to a csv file and saved.

Graphical user interface, text, application, email

Description automatically generated

-------------------------------------------------------------------------------------------