## 

## **Practical-7**

**Student Name:** Gauri Prabhakar **UID:** 18BCS6201

**Branch:** 18AITAIML-2 **Section/Group:** B

**Semester:** 7 **Date of Performance:** 12th October, 2021

**Subject Name:** Computer Vision Lab **Subject Code:** CSF - 432

# Aim/Overview of the practical:

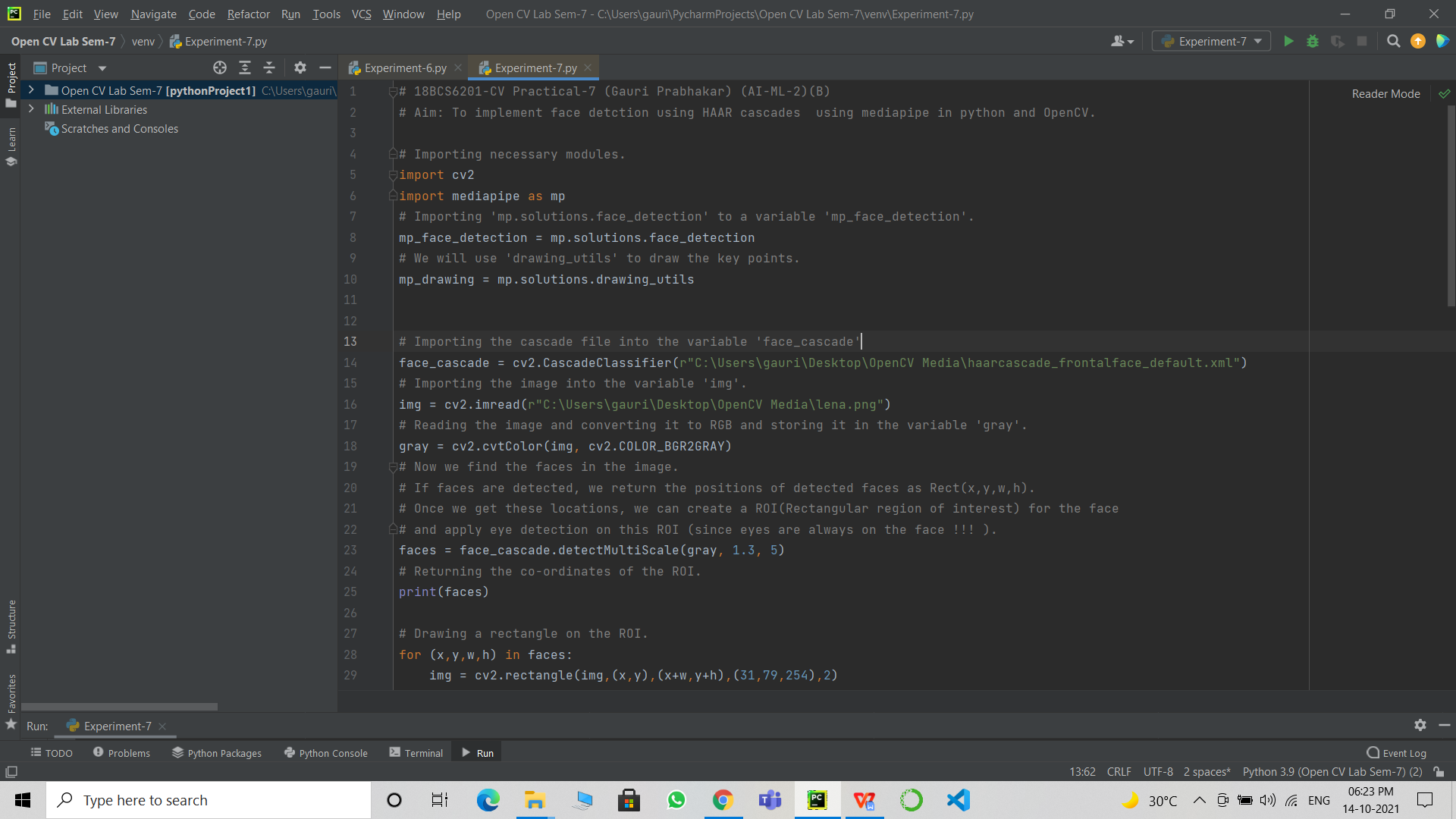
To implement face detection using HAAR cascades using mediapipe in python and OpenCV.

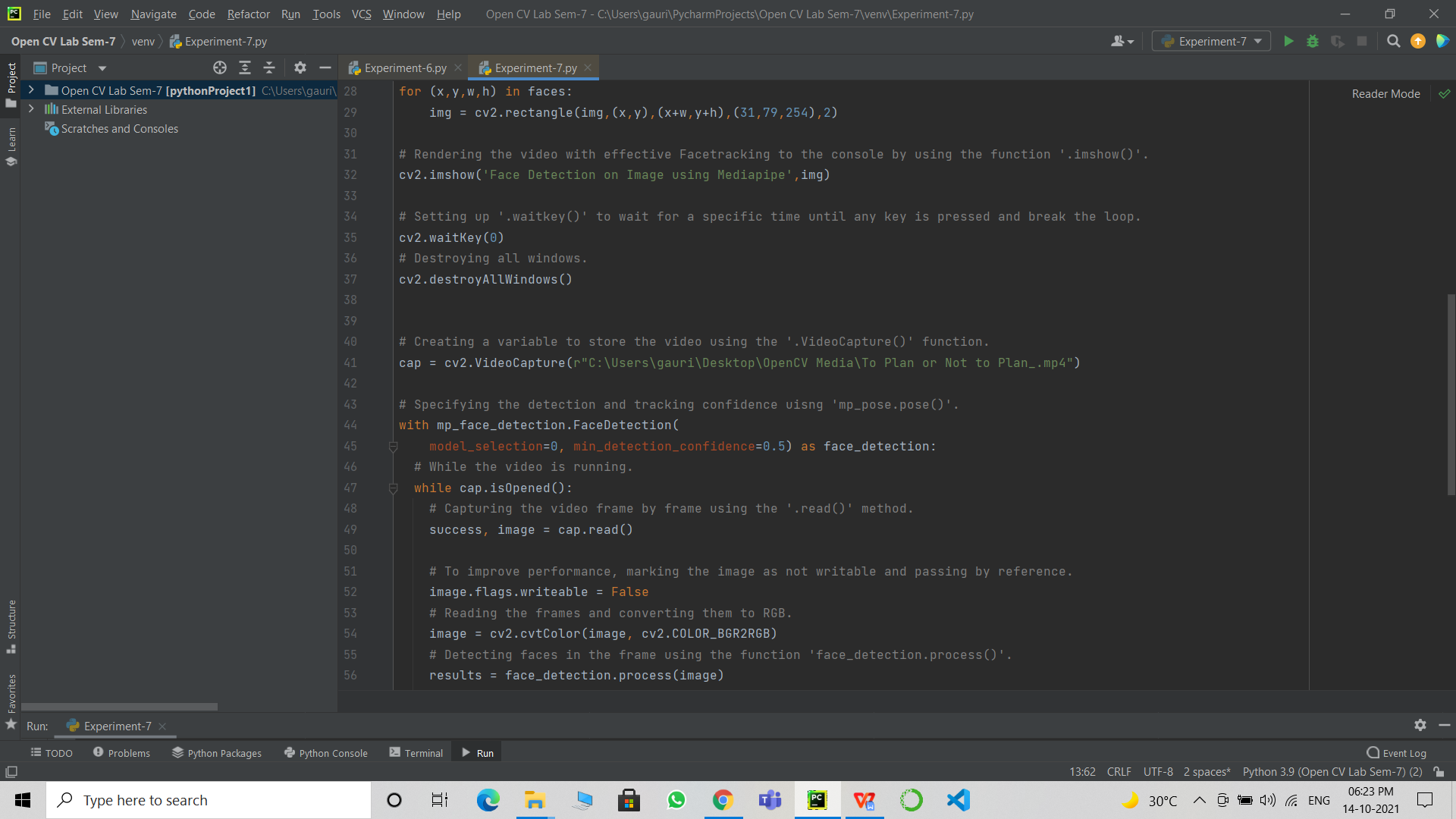
# Task to be done:

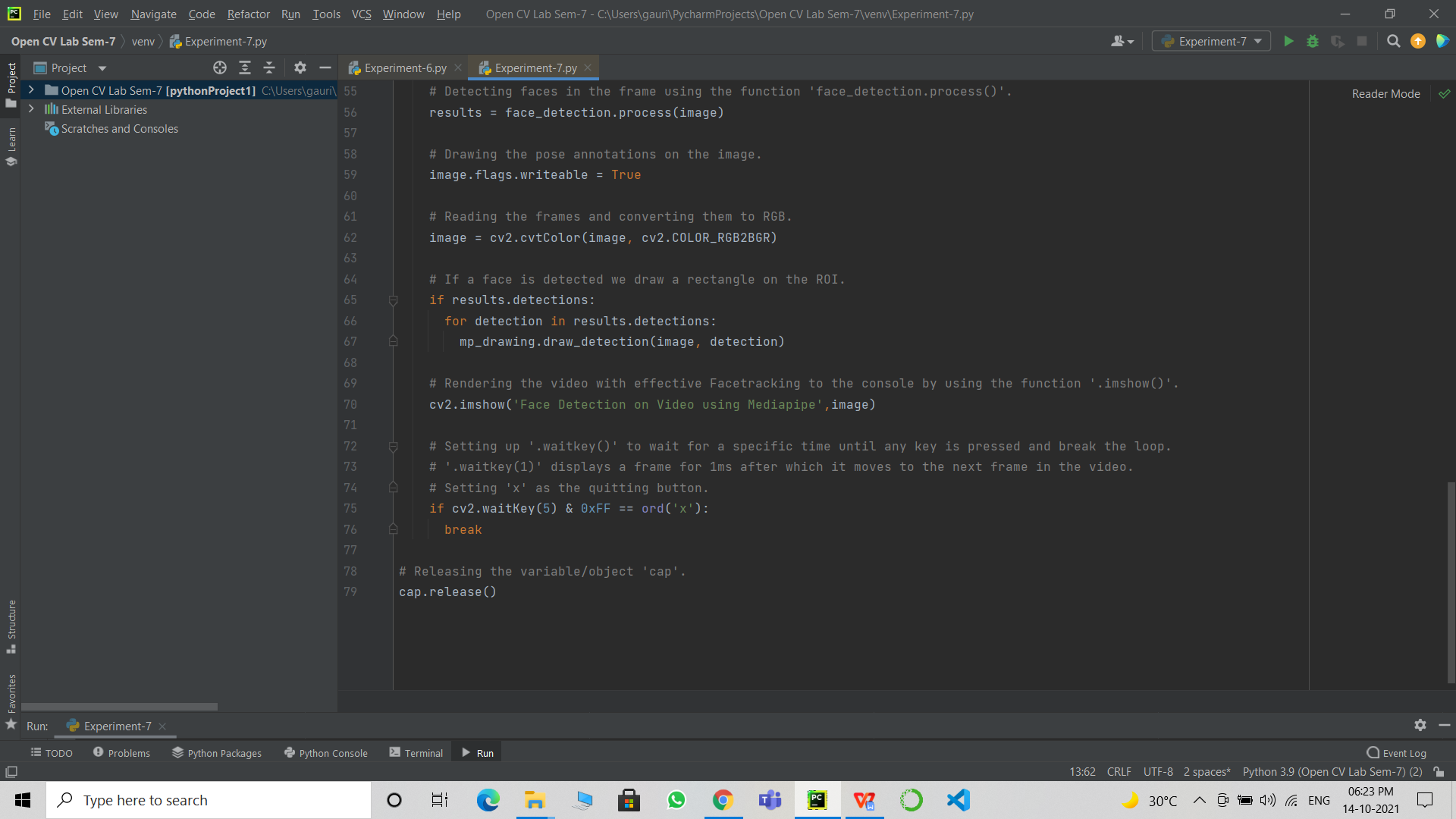
To implement face detection using HAAR cascades using mediapipe in python and OpenCV.

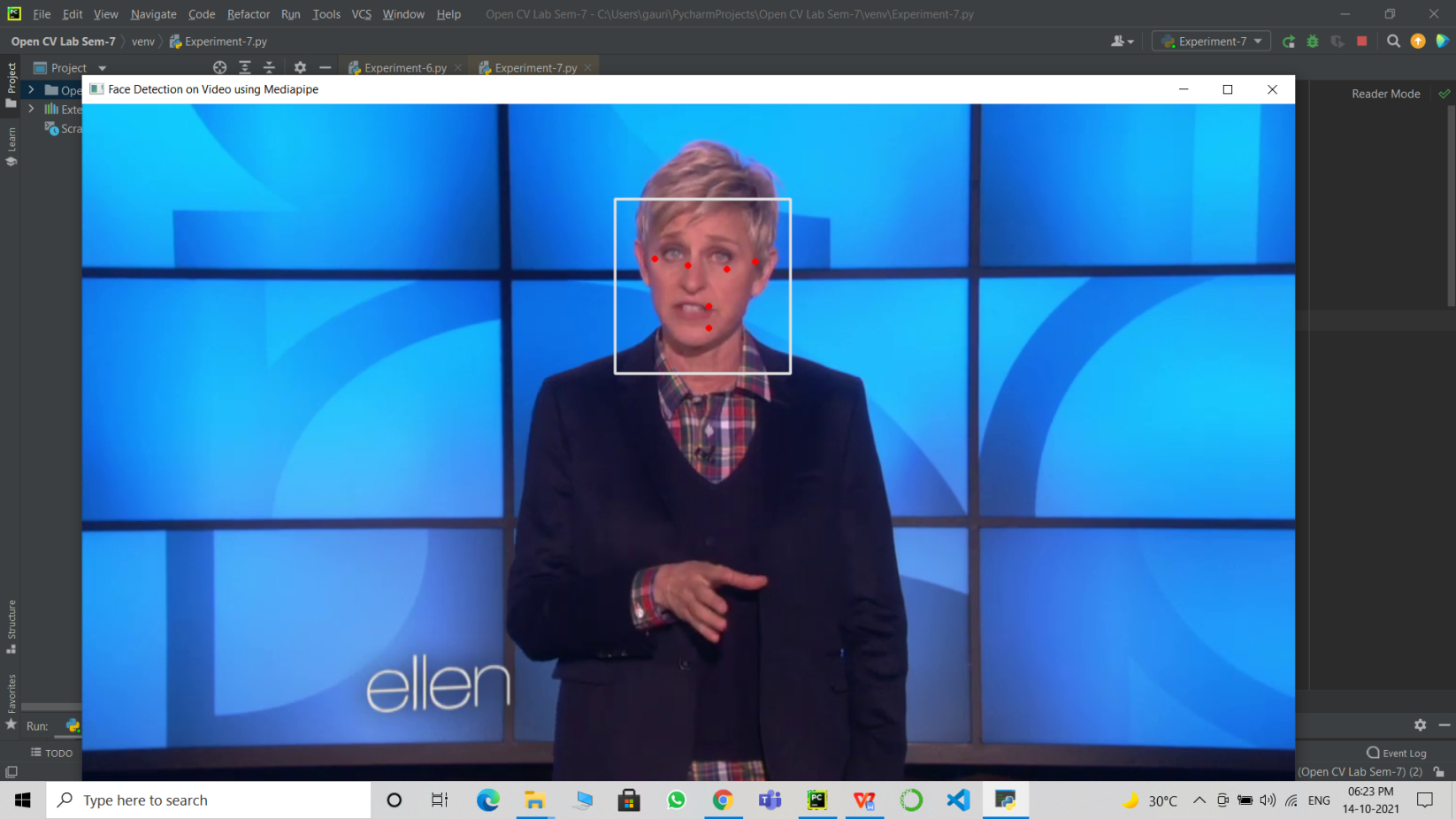
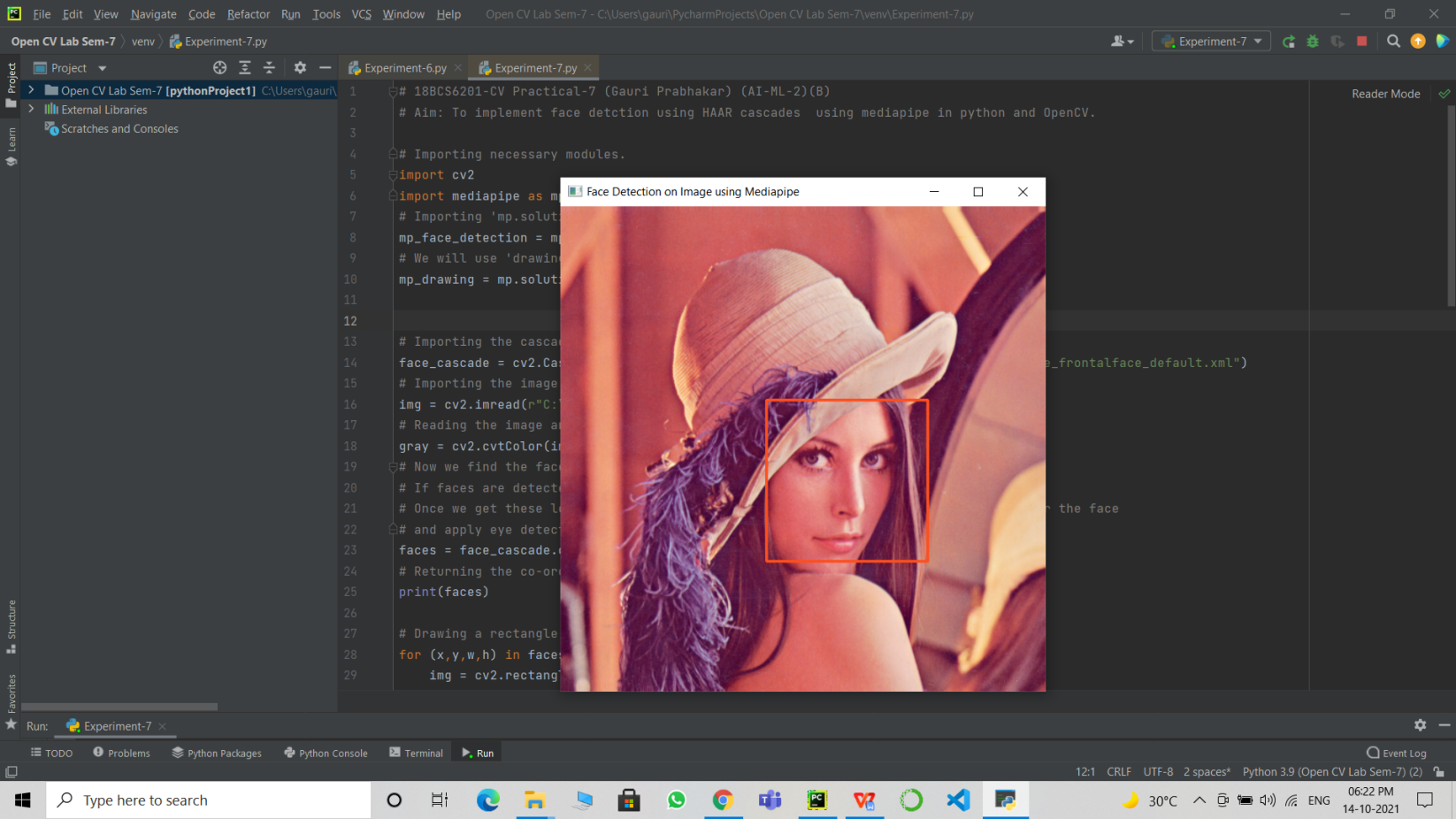
# Steps to be followed:

1. Importing necessary modules.
2. Importing 'mp.solutions.face\_detection' to a variable 'mp\_face\_detection'.
3. We will use 'drawing\_utils' to draw the key points.
4. Importing the cascade file into the variable 'face\_cascade'.
5. Importing the image into the variable 'img'.
6. Reading the image and converting it to RGB and storing it in the variable 'gray'.
7. Now we find the faces in the image.
8. If faces are detected, we return the positions of detected faces as Rect(x,y,w,h).
9. Once we get these locations, we can create a ROI(Rectangular region of interest) for the face
10. and apply eye detection on this ROI (since eyes are always on the face !!! ).
11. Returning the co-ordinates of the ROI.
12. Rendering the video with effective Facetracking to the console by using the function '.imshow()'.
13. Setting up '.waitkey()' to wait for a specific time until any key is pressed and break the loop.
14. Destroying all windows.
15. Creating a variable to store the video using the '.VideoCapture()' function.
16. Specifying the detection and tracking confidence uisng 'mp\_pose.pose()'.
17. While the video is running.
18. Capturing the video frame by frame using the '.read()' method.
19. To improve performance, marking the image as not writable and passing by reference.
20. Reading the frames and converting them to RGB.
21. Detecting faces in the frame using the function 'face\_detection.process()'.
22. Drawing the pose annotations on the image.
23. Reading the frames and converting them to RGB.
24. If a face is detected we draw a rectangle on the ROI.
25. Rendering the video with effective Posetracking to the console by using the function '.imshow()'.
26. Setting up '.waitkey()' to wait for a specific time until any key is pressed and break the loop.
27. '.waitkey(1)' displays a frame for 1ms after which it moves to the next frame in the video.
28. Setting 'x' as the quitting button.
29. Releasing the variable/object 'cap'.
30. **Result/Output/Writing Summary:**









# Learning outcomes (What I have learnt):

* Open CV modules.
* The mediapipe library.
* Detect faces using the mediapipe library and HAAR cascades.
* Face tracking a saved video.
* Face tracking a saved image.
* Highlighting key points.

**Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Parameters | Marks Obtained | Maximum Marks |
| 1. |  |  |  |
| 2. |  |  |  |
| 3. |  |  |  |