```
                print(body2)
                send_message(body2, info_dict)

    # If significant amount of detections (more than half of detection_thresh) has
occured then we reset the Initial Time.
    elif status2 and sum(de2) > (detection_thresh/2):
        initial_time2 = None

    # Get the current time in the required format
    current_time2 =
datetime.datetime.now().strftime("%A, %I:%M:%S %p %d %B %Y")

    # Display the FPS
    cv2.putText(annotated_image2, 'FPS: {:.2f}'.format(
        fps2), (510, 450), cv2.FONT_HERSHEY_COMPLEX, 0.6, (255, 40, 155),
2)

    # Display Time
    cv2.putText(annotated_image2, current_time2, (310, 20),
                cv2.FONT_HERSHEY_COMPLEX, 0.5, (0, 0, 255), 1)

    # Display the Room Status
    cv2.putText(annotated_image2, 'Room Occupied: {}'.format(str(status2)), (10,
20), cv2.FONT_HERSHEY_SIMPLEX, 0.6,
                (200, 10, 150), 2)

    # Show the patience Value
    if initial_time2 is None:
        text = 'Patience: {}'.format(patience2)
    else:
        text = 'Patience: {:.2f}'.format(
            max(0, patience2 - (time.time() - initial_time2)))

    cv2.putText(annotated_image2, text, (10, 450),
                cv2.FONT_HERSHEY_COMPLEX, 0.6, (255, 40, 155), 2)

    # If status is true save the frame
    # if status:
    #       out.write(annotated_image)

    frame= np.hstack((frame, frame2))
    # Show the Frame
    cv2.imshow('frame', frame)
```