```
function spectogram_plotter_generated_signals_vol2(generated_signal,window_length,window_shift,window_type,sampling_frequency,tone_duration,tone_pause)

    sampling_frequency = sampling_frequency/2;
    % Transpose is taken if the signal is a column vector , instead of a
    % row vector

    if(size(generated_signal,1) > 5)
        generated_signal = generated_signal';
    end

    win = 0;    % initial blank window

    %% Controlling if we satisfy the length constraints:

    if( length(generated_signal) < window_length )

    disp(' Error: The signal is too short for this window length. Submit a new signal or choose a new window length.');

    end

    if( window_shift > window_length )
    disp(' Error: Enter a valid window length that is longer than the window shift.');
    end




    %% Assigning the window type:
    if( strcmp(window_type,'rectwin')) %#ok<*STCMP>
        win=ones(1,window_length);
    elseif(strcmp(window_type,'hamming'))
        win=hamming(window_length)';
    elseif(strcmp(window_type,'tukeywin'))
        win=tukeywin(window_length)';
    else
        win=ones(1,window_length);
    end

    %************************************************

    begin_point = 1;    % Initial point for thewindow
    vertical_line_no=1; % Counter for the vertical lines

    %% Calculating the STFT
    while( begin_point + window_length -1 <= length(generated_signal) )
        STFT(vertical_line_no,:) = abs(fft(generated_signal( begin_point : begin_point+window_length-1 ).*win,window_length*100));
        begin_point = begin_point + window_shift;
        vertical_line_no = vertical_line_no+1 ;
    end
```

```
    signal_duration = length(generated_signal) / (2*sampling_frequency);
    time_values = 0 : size(STFT,1)-1; %Time values to be plotted
    t_values = linspace(0,signal_duration,length(time_values));
    time_values = t_values;

    frequency_values = linspace(0,sampling_frequency,window_length*50); %The↙
default fs = 4000
%   frequency_values = (4000/sampling_frequency)*frequency_values

    STFT_in_DB=(20*log10(STFT))'; % Coverting the magnitudes to decibels for better↙
analysis

    if (sampling_frequency > 1999)
        y_limit = 2000;
    else
        y_limit = sampling_frequency;
    end

    %% Plotting the STFT
    figure
    surf(time_values,frequency_values,STFT_in_DB(1:window_length*50,:));
    shading("interp");
    view(2); %The view should be from above since our plot is 2D in view
    xlabel('Time','Fontsize',12);
    ylabel('Frequency(Hz)','Fontsize',12);
    title('Spectrogram','Fontsize',14);
    colorbar;
    ylim([0 2000]);
    xlim([0 signal_duration]);
%   yticks(linspace(0, sampling_frequency, 11));

end
```