```
clc;
close all;

Fs = 8e3;
channel = 1;
bits = 16;
ID = -1;

% Audio Recording Section
recaudio = audiorecorder(Fs,bits,channel); % data of audio processing
(object file)
disp('Recording Started');
```

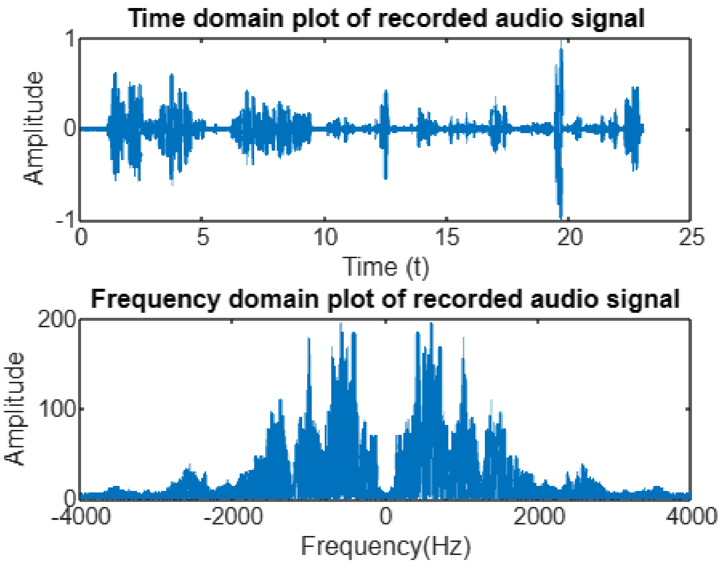
```
Recording Started
```

```
duration = 23; % duration of audio
recordblocking(recaudio,duration); % it will record audio
disp('Recording Stopped');
```

```
Recording Stopped
```

```
p = play(recaudio);


mySpeech = getaudiodata(recaudio); % audio convert to data(numeric array)
sound(mySpeech,Fs,bits); % it will convert array to sound(electrical
signals)

% audio file save and write operation
filename = 'myvoice.wav'; % file banayi
audiowrite(filename,mySpeech,Fs); % from now recorded audio will store in
this file


% Time Domain Analysis
t = 0:1/Fs:(length(mySpeech)-1)/Fs; % duration of signal
subplot(2,1,1);
plot(t,mySpeech,'LineWidth',1.5);
xlabel('Time (t)');
ylabel('Amplitude');
title('Time domain plot of recorded audio signal');

% Frequency Domain Analysis

n = length(mySpeech);
f = 0:(n-1)*Fs/n; % k/n = f/fs;
y = fft(mySpeech,n);
f_0 = (-n/2:n/2-1).*(Fs/n); % frequency not centered at 0 so it will
centered it to 0
y_0 = fftshift(y); % shifting of fft
ay_0=abs(y_0);
```

```
subplot(2,1,2);
plot(f_0,ay_0,'LineWidth',1.5);
xlabel('Frequency(Hz)');
ylabel('Amplitude');
title('Frequency domain plot of recorded audio signal');
```


Time domain plot of recorded audio signal
Amplitude
Time (t)
Frequency domain plot of recorded audio signal
Amplitude
Frequency(Hz)