**SIT PROJECT CODE**

**Speech Recognition in MATLAB   
using FFT**

**Code for voice training**

clear all;

close all;

clc;

%% Creating a recorder object

recorder = audiorecorder(16000, 8, 2); % 16000 Hz, 8 bits, 2 channels

%% Record the user's voice for 5s

disp("Please record your voice:")

drawnow(); % update figures and process callbacks

pause(1)

recordblocking(recorder, 5); % records the audio to recorder object for 5s.

%% Play the user's voice

play(recorder); % plays the audio from the recorder object

data = getaudiodata(recorder); % data is stored in a 80000 x 2 matrix.

plot(data); % plots the data

xlabel("Frequency")

ylabel("Normalized values between -1 to 1")

title("Audio sample")

%% Feature Extraction

f = voiceFeatures(data)

%% Save the user database

user = input("Enter the user number: ");

try

load database

F = [F; f];

C = [C; user];

save database

catch

F = f;

C = user;

save database F C

end

msgbox("Yay! Voice registered!")

**Code for extracting features of recorded audio**

function [x\_Pitch] = voiceFeatures(data)

F = fft(data(:,1)); % converts time domain data to frequency domain of channel 1

plot(real(F)) % plots the real part of the data

m = max(real(F)); % max value of real(F)

x\_Pitch = find(real(F) == m, 1); % finds the first indice of max value of real(F)

end

**Code for testing of an input audio**

clear all;

close all;

clc;

%% Creating a recorder object

recorder = audiorecorder(16000, 8, 2); % 16000 Hz, 8 bits, 2 channels

%% Record the user's voice for 5s

disp("Please record your voice:")

drawnow(); % update figures and process callbacks

pause(1) % pauses for 1s

recordblocking(recorder, 5); % records the audio to recorder object for 5s.

%% Play the user's voice

play(recorder); % plays the audio from the recorder object

data = getaudiodata(recorder); % data is stored in a 8000 x 2 matrix.

plot(data); % plots the data

xlabel("Frequency")

ylabel("Normalized values between -1 to 1")

title("Audio sample")

%% Feature Extraction

f\_test = voiceFeatures(data)

%% Find the closest feature from database.

load database

D = [];

for i = 1:size(F, 1)

d = abs(F(i) - f\_test);

D = [D d];

end

%% Smallest distance

sm = inf;

ind = -1;

for i = 1:length(D)

if D(i) < sm

sm = D(i);

ind = i;

end

end

detected\_class = C(ind);

disp('The detected class is ');

detected\_class