Python code to display Audio waveform

Instructions :-

- Open colab.research.google.com
- Create a new notebook and write the code below and save the notebook
- Place the test.wav file in the drive folder where python code is saved

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.io.wavfile import read , write
from IPython.display import Audio
from numpy.fft import fft, ifft
import os
from google.colab import files
from google.colab import drive
drive.mount('GD')
os.chdir("GD/My Drive/Colab Notebooks")

#matpoltlib inline

Fs, data = read('test.wav')
data=data[:,0]
print(" Sampling frequency is ", Fs)

Audio(data,rate=Fs)
plt.figure()
plt.plot(data)
plt.xlabel('Sample Index')
plt.ylabel('Amplitude')
plt.title('MCA3 Audio Practical')
plt.show()

write('output.wav',Fs,data)
```

Python code step by step:

Importing Libraries:

1. import numpy as np
2. import matplotlib.pyplot as plt
3. from scipy.io.wavfile import read, write
4. from IPython.display import Audio
5. from numpy.fft import fft, ifft

6. import os
7. from google.colab import files
8. from google.colab import drive
    - numpy is imported as np for numerical operations.
    - matplotlib.pyplot is imported as plt for plotting graphs.
    - read and write functions from scipy.io.wavfile are used to read and write WAV files.
    - Audio from IPython.display is used to play audio directly in Jupyter notebooks.
    - fft and ifft from numpy.fft are used for Fast Fourier Transform and its inverse.
    - os is used for interacting with the operating system.
    - files and drive from google.colab are used for file operations and mounting Google Drive in Colab.

Mounting Google Drive:

9. drive.mount('GD')
10. os.chdir("GD/My Drive/Colab Notebooks")
    - drive.mount('GD') mounts Google Drive to the Colab environment.
    - os.chdir("GD/My Drive/Colab Notebooks") changes the current working directory to a specific folder in Google Drive.

Reading the Audio File:

11. Fs, data = read('test.wav')
12. data = data[:, 0]
13. print("Sampling frequency is ", Fs)
    - Fs, data = read('test.wav') reads the WAV file test.wav. Fs is the sampling frequency, and data is the audio data.
    - data = data[:, 0] selects the first channel if the audio is stereo.
    - print("Sampling frequency is ", Fs) prints the sampling frequency.

Playing the Audio:

14. Audio(data, rate=Fs)
    - Audio(data, rate=Fs) plays the audio data at the specified sampling rate.

Plotting the Audio Data:

15. plt.figure()

16. plt.plot(data)
17. plt.xlabel('Sample Index')
18. plt.ylabel('Amplitude')
19. plt.title('MCA3 Audio Practical')
20. plt.show()
    - plt.figure() creates a new figure.
    - plt.plot(data) plots the audio data.
    - plt.xlabel('Sample Index') labels the x-axis.
    - plt.ylabel('Amplitude') labels the y-axis.
    - plt.title('MCA3 Audio Practical') sets the title of the plot.
    - plt.show() displays the plot.

Writing the Audio File:

21. write('output.wav', Fs, data)
    - write('output.wav', Fs, data) writes the audio data to a new WAV file output.wav with the same sampling frequency.

```
Sampling frequency is  44100
```