LAB-6

G. ANVITH REDDY

BL.EN.U4AIE21047

In [ ]:

A1. Take a portion of your recorded signal which represents a vowel sound. Perform FFT on the signal snippet and observe the amplitude spectrum. Repeat the same for a few vowel sounds

In [1]:
```
import librosa
import matplotlib.pyplot as plt
```

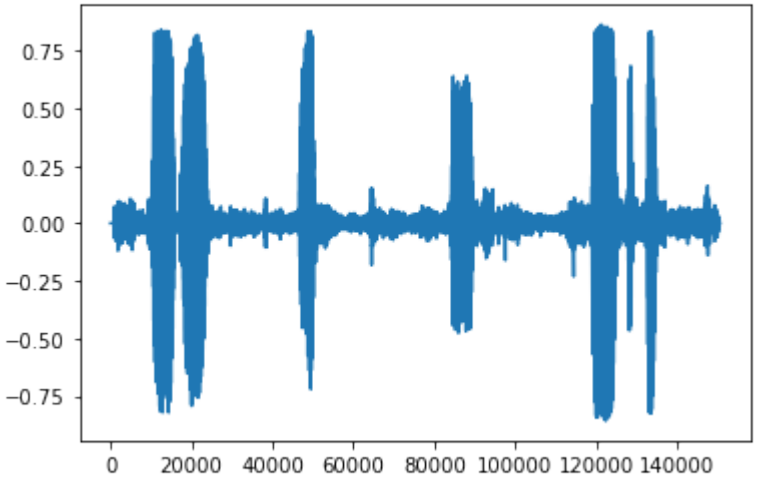
```
C:\Users\anvit\anaconda3\lib\site-packages\numpy\_distributor_init.py:30: UserWarning: loaded more than 1 DLL from .libs:
C:\Users\anvit\anaconda3\lib\site-packages\numpy\.libs\libopenblas.WCDJNK7YVMPZQ2ME2ZZHJJRJ3JIKNDB7.gfortran-win_amd64.dll
C:\Users\anvit\anaconda3\lib\site-packages\numpy\.libs\libopenblas.XWYDX2IKJW2NMTWSFYNGFUWKQU3LYTCZ.gfortran-win_amd64.dll
  warnings.warn("loaded more than 1 DLL from .libs:"
```

In [2]:
```
y, sr = librosa.load('sp2_anvith.wav')
print(y)
print(sr)
```

```
[-4.5693675e-11  4.5294154e-11  1.2557305e-10 ...  2.4299371e-07
 -2.1052608e-07  1.7814057e-07]
22050
```

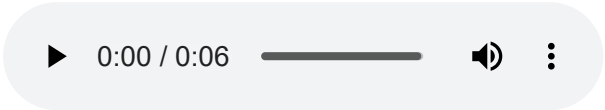
In [3]:
```
plt.plot(y)
```

Out[3]:
```
[<matplotlib.lines.Line2D at 0x1c455aa8eb0>]
```



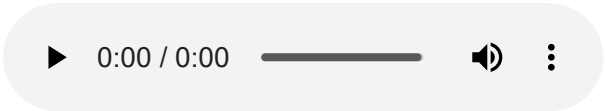
In [4]:
```
from IPython.display import Audio
Audio(data=y, rate=sr)
```

Out[4]:



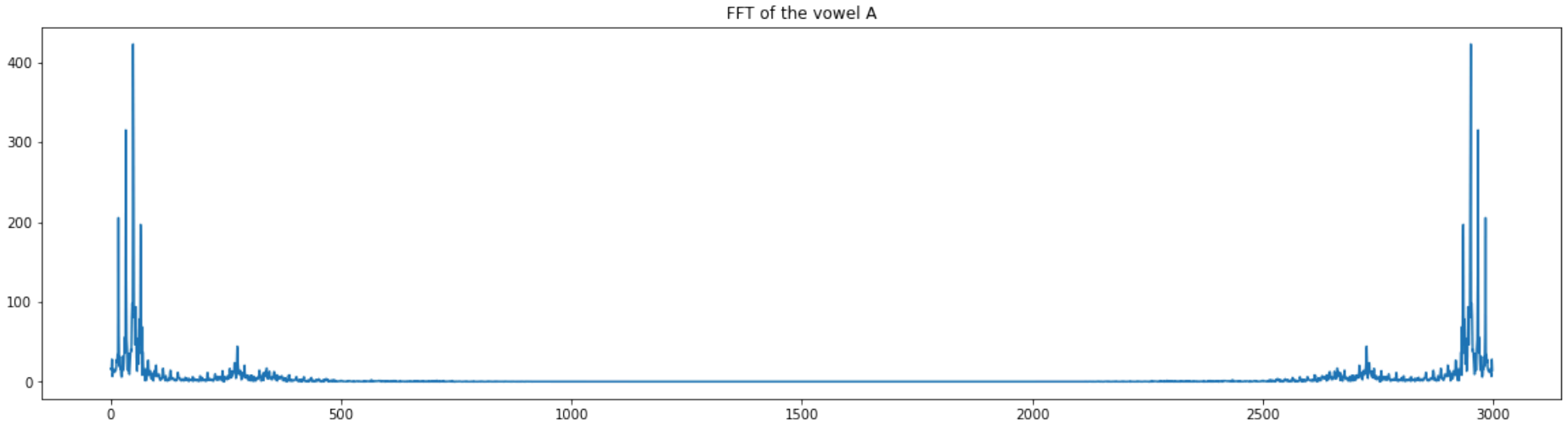
In [5]:
```
Audio(y[10000:13000],rate = sr) # the sound of a
```

Out[5]:



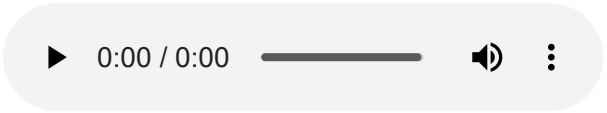
In [6]:
```
import numpy as np

vowel_a = y[10000:13000]
vowel_a_fft = np.fft.fft(vowel_a)  # fft on the sound of vowel a

plt.figure(figsize=(20,5))
plt.plot(np.abs(vowel_a_fft))
plt.title('FFT of the vowel A')
```

Out[6]:
```
Text(0.5, 1.0, 'FFT of the vowel A')
```



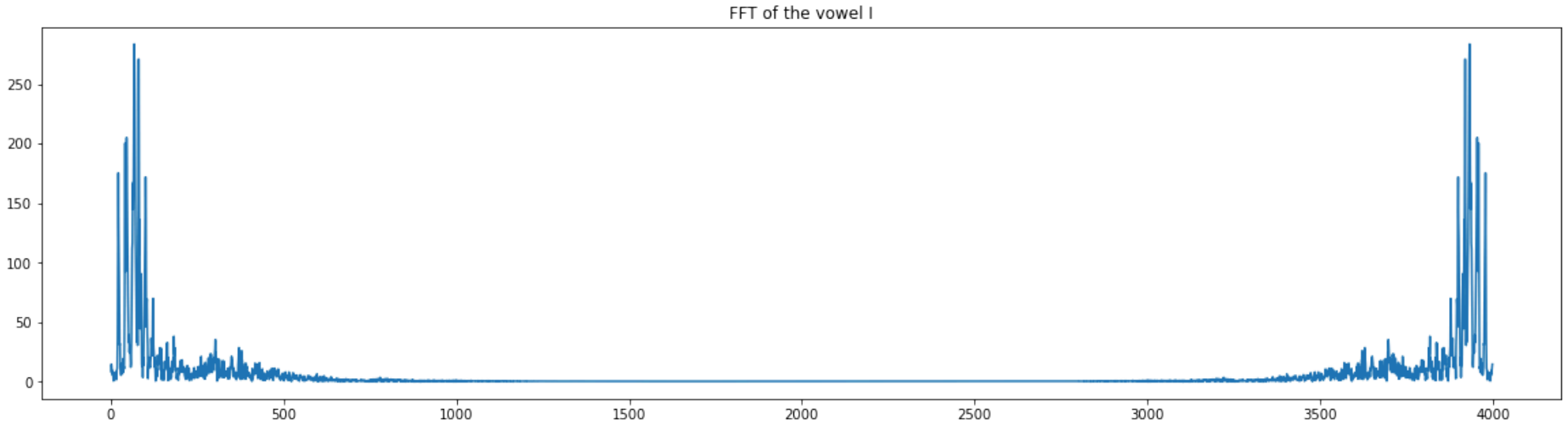
In [7]:
```
Audio(y[17500:21500],rate = sr) # the sound of i
```

Out[7]:



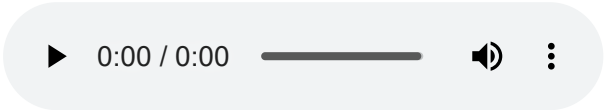
In [8]:
```
vowel_i = y[17500:21500]
vowel_i_fft = np.fft.fft(vowel_i) # fft on vowel i

plt.figure(figsize=(20,5))
plt.plot(np.abs(vowel_i_fft))
plt.title('FFT of the vowel I')
```

Out[8]:
```
Text(0.5, 1.0, 'FFT of the vowel I')
```



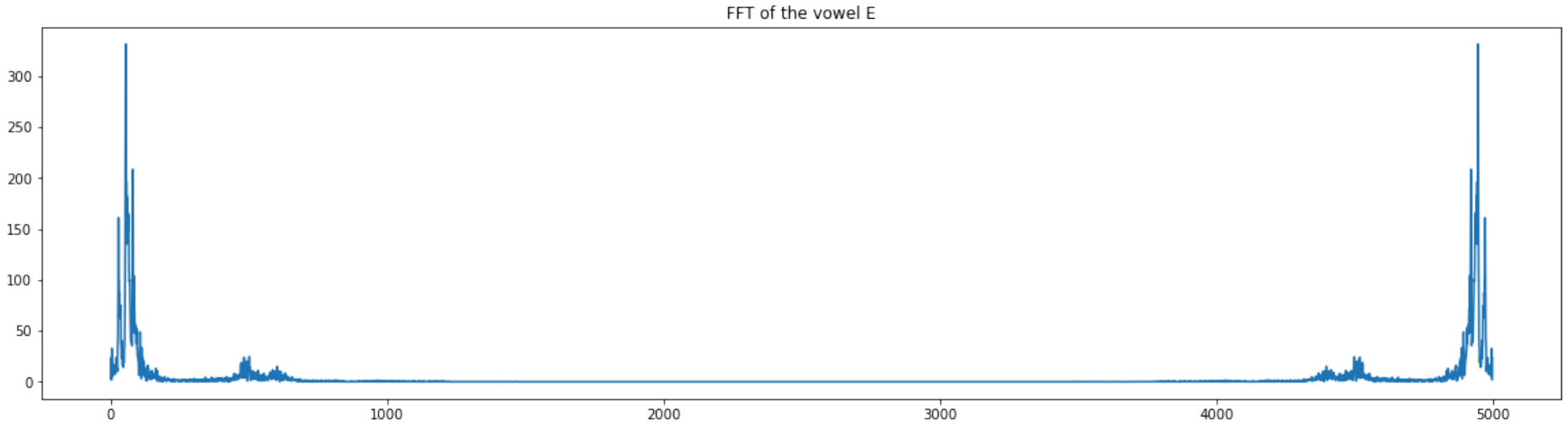
In [9]:
```
Audio(y[85000:90000],rate = sr) # the sound of e
```

Out[9]:



In [10]:
```
vowel_e = y[85000:90000]
vowel_e_fft = np.fft.fft(vowel_e) # fft on e

plt.figure(figsize=(20,5))
plt.plot(np.abs(vowel_e_fft))
plt.title('FFT of the vowel E')
```

Out[10]:
```
Text(0.5, 1.0, 'FFT of the vowel E')
```

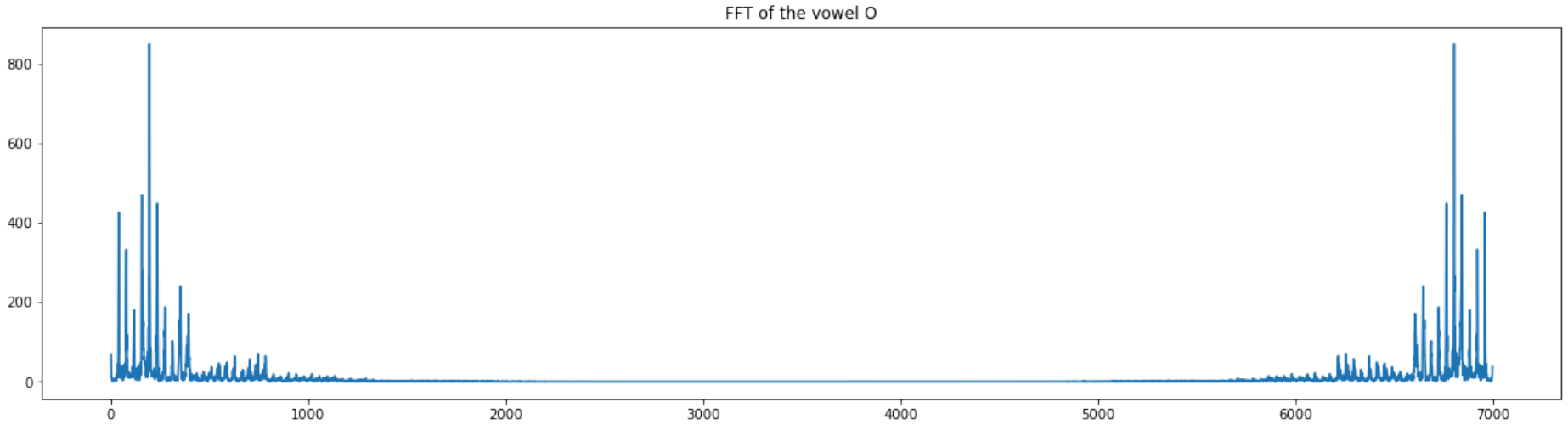


In [11]:
```
Audio(y[120000:127000],rate = sr) # the sound of o
```

Out[11]:

In [12]:
```
vowel_o = y[120000:127000]
vowel_o_fft = np.fft.fft(vowel_o)

plt.figure(figsize=(20,5))
plt.plot(np.abs(vowel_o_fft))  # mod on the fft of vowel o
plt.title('FFT of the vowel O')
```

Out[12]:
```
Text(0.5, 1.0, 'FFT of the vowel O')
```



In [ ]:

A2. Repeat the A1 for a consonant sound. Perform the same for a few consonant sounds.

In [ ]:

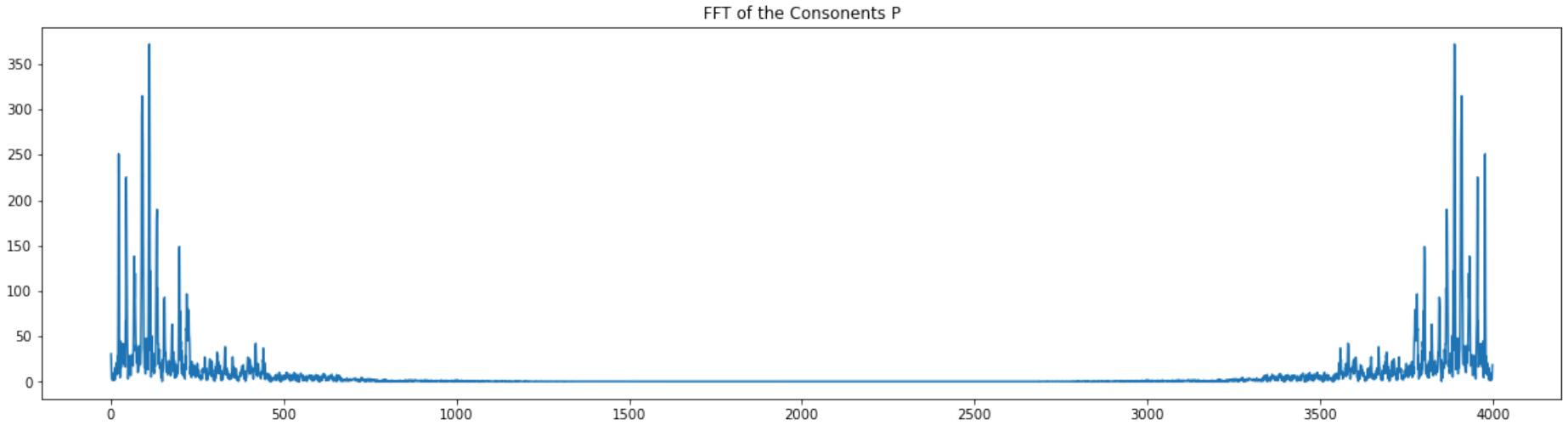
In [13]:
```
Audio(y[118000:122000],rate = sr)
```

Out[13]:

0:00 / 0:00

In [14]:
```
consonant_p = y[118000:122000]
consonant_p_fft = np.fft.fft(consonant_p) # fft on consonant sound

plt.figure(figsize=(20,5))
plt.plot(np.abs(consonant_p_fft))
plt.title('FFT of the Consonents P')
```

Out[14]:
```
Text(0.5, 1.0, 'FFT of the Consonents P')
```



In [ ]:

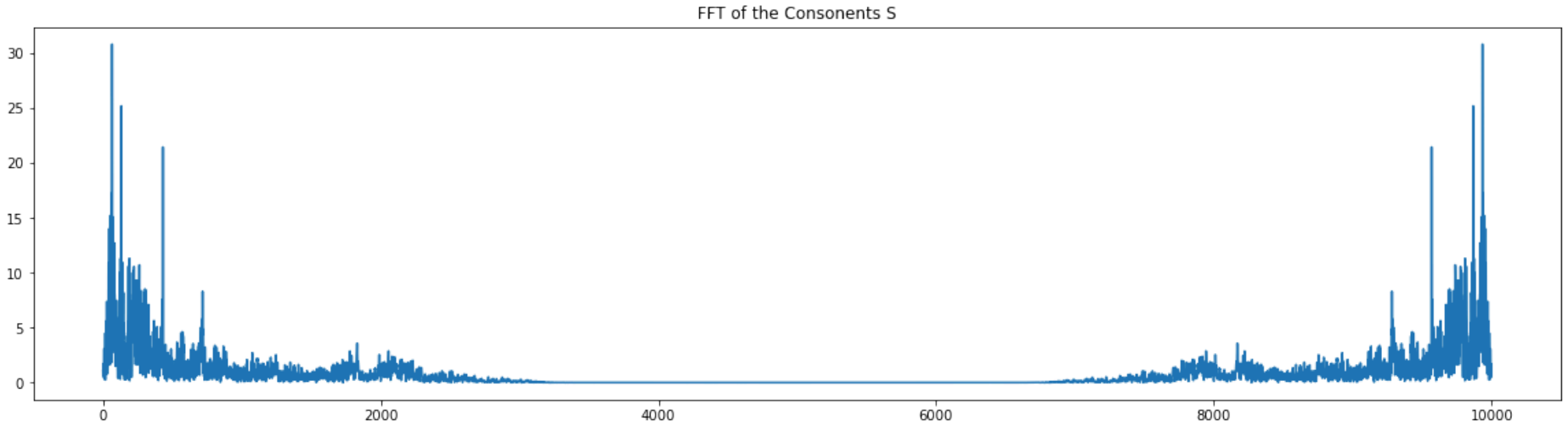
In [33]:
```
Audio(y[70000:80000],rate = sr)
```

Out[33]:

0:00 / 0:00

In [34]:
```
consonant_s = y[70000:80000]
consonant_s_fft = np.fft.fft(consonant_s) # fft on the sound of s

plt.figure(figsize=(20,5))
plt.plot(np.abs(consonant_s_fft))
plt.title('FFT of the Consonents S')
```

Out[34]:
```
Text(0.5, 1.0, 'FFT of the Consonents S')
```



In [ ]:

A3. Repeat A2 for few slices of silence & non-voiced portions of the recorded speech signal.

In [ ]:

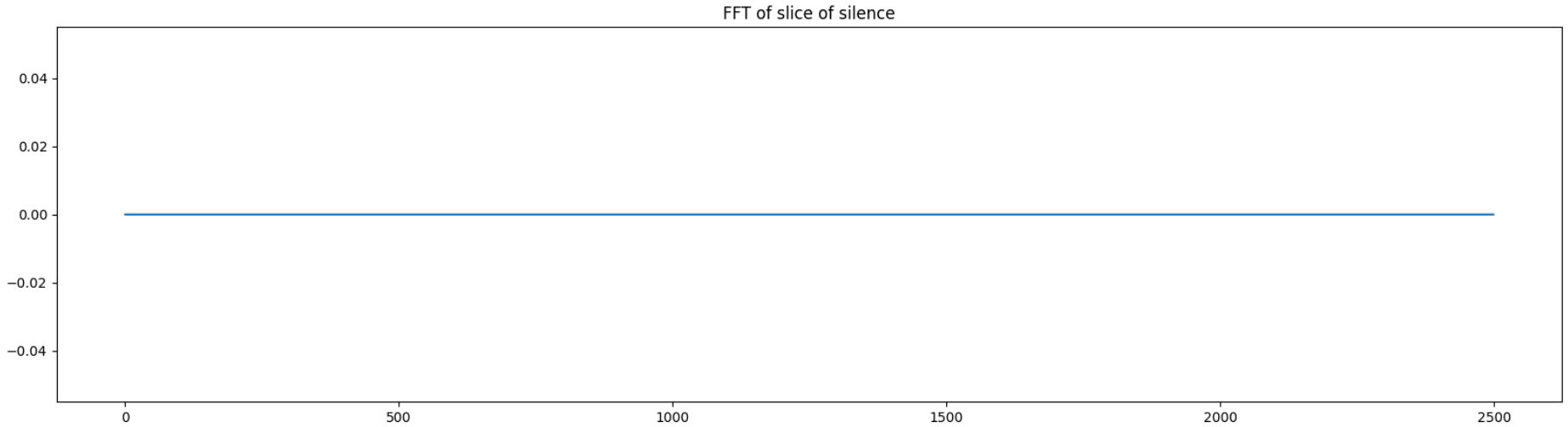
In [35]:
```
Audio(y[0:3000],rate = sr)
```

Out[35]:

0:00 / 0:00

In [64]:
```
silence = y[0:2500]
silence_fft = np.fft.fft(silence) # fft on the sound of silence

plt.figure(figsize=(20,5))
plt.plot(np.abs(silence_fft))
plt.title('FFT of slice of silence')
```

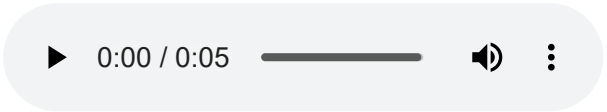
Out[64]:
```
Text(0.5, 1.0, 'FFT of slice of silence')
```



In [ ]:

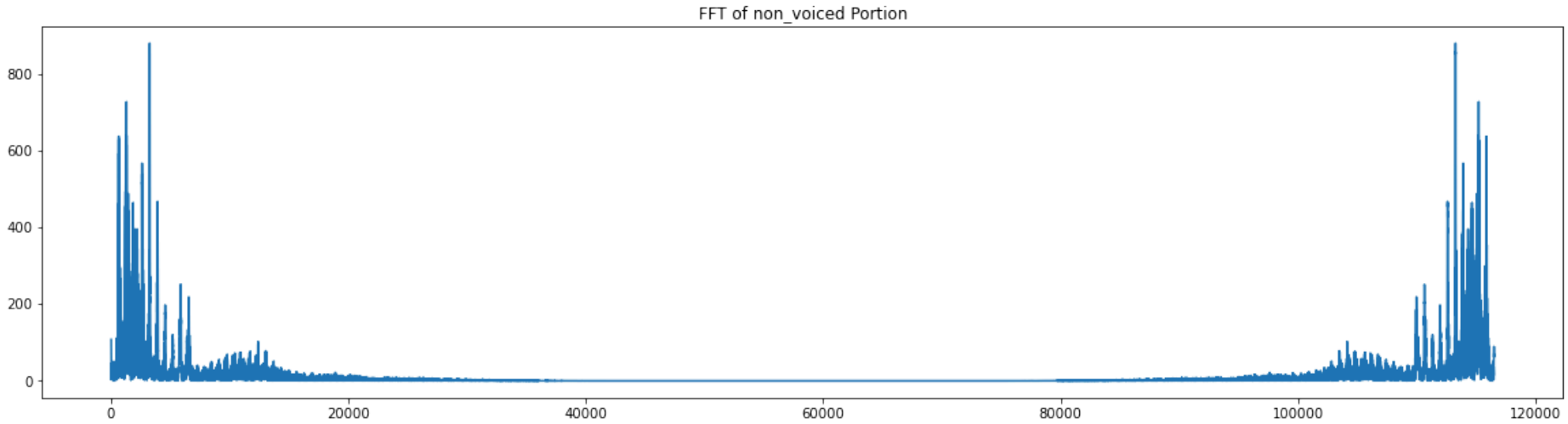
In [36]:
```
Audio(y[34000:],rate = sr)
```

Out[36]:



In [37]:
```
non_voiced = y[34000:]
non_voiced_fft = np.fft.fft(non_voiced) # fft on non voiced

plt.figure(figsize=(20,5))
plt.plot(np.abs(non_voiced_fft))
plt.title('FFT of non_voiced Portion')
```

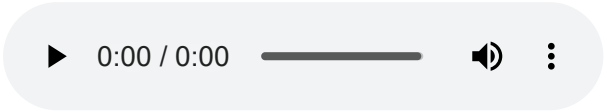
Out[37]:
```
Text(0.5, 1.0, 'FFT of non_voiced Portion')
```



In [ ]:

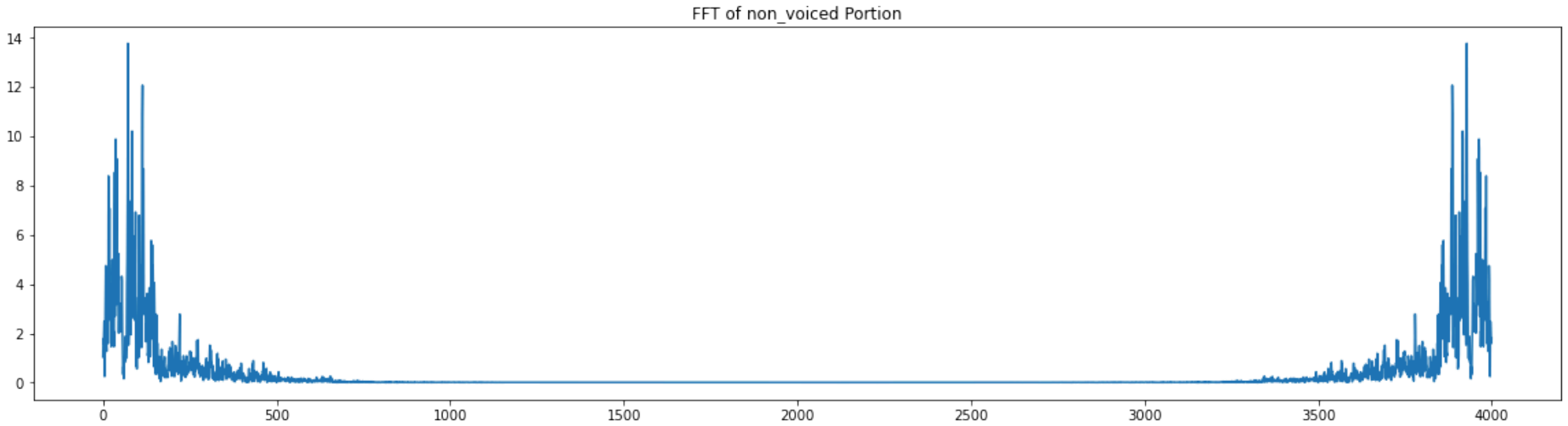
In [38]:
```
Audio(y[30000:34000],rate = sr)
```

Out[38]:



In [39]:
```
non_voiced2 = y[30000:34000]
non_voiced2_fft = np.fft.fft(non_voiced2) # fft on non voiced 2

plt.figure(figsize=(20,5))
plt.plot(np.abs(non_voiced2_fft))
plt.title('FFT of non_voiced Portion')
```

Out[39]:
```
Text(0.5, 1.0, 'FFT of non_voiced Portion')
```

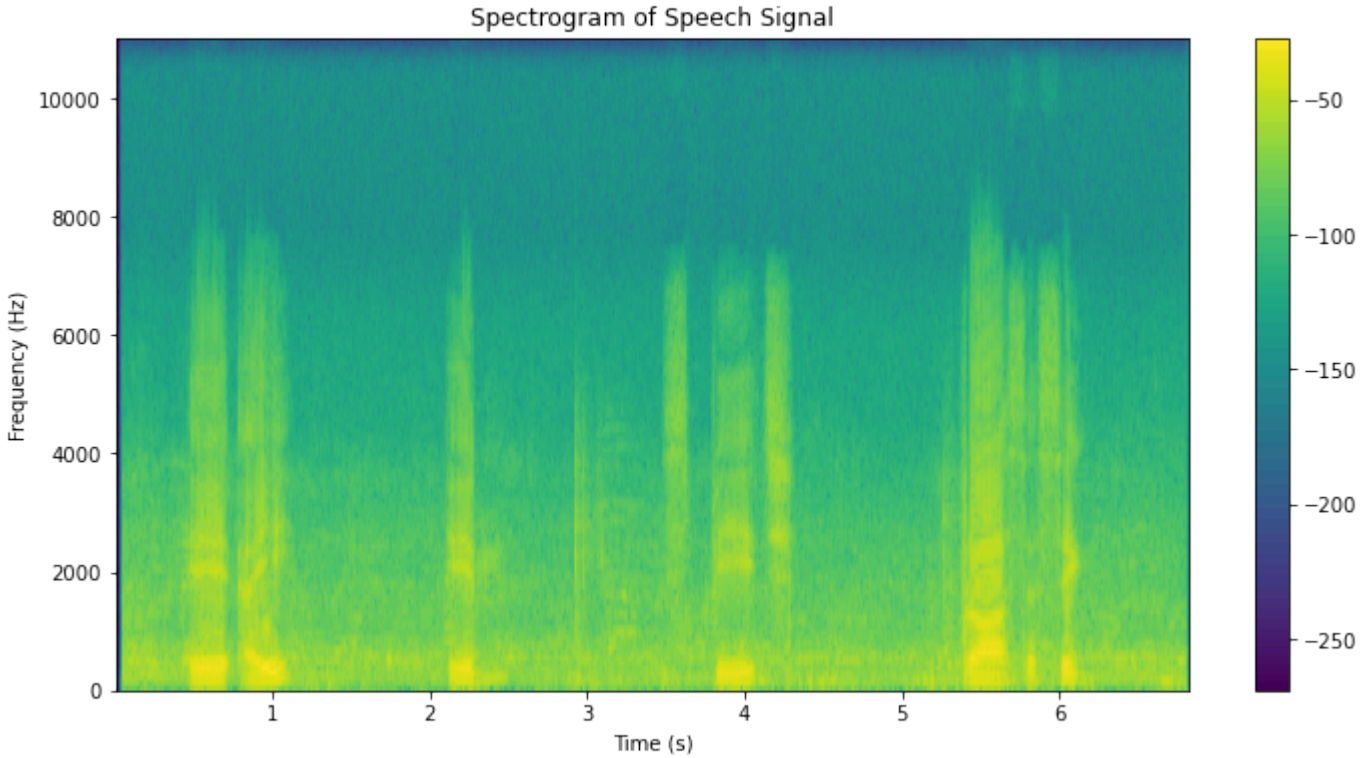


In [ ]:

4. Now you have acquainted yourself with spectral amplitudes of various consonants and vowel□based phonemes. Generate the spectrogram of the signal and observe the change points of the signals with associated speech segments. Observe to identify the consonants and vowels from the spectrogram.

In [ ]:

In [41]:
```
def generate_spectrogram(signal, sample_rate):
    plt.specgram(signal, Fs=sample_rate) # plotting spectogram
    plt.title("Spectrogram of Speech Signal")
    plt.xlabel("Time (s)")
    plt.ylabel("Frequency (Hz)")
    plt.colorbar()

plt.figure(figsize=(12, 6))
generate_spectrogram(y,sr)
plt.show()
```


Spectrogram of Speech Signal
Frequency (Hz)
10000
8000
6000
4000
2000
0
1
2
3
4
5
6
Time (s)
−50
−100
−150
−200
−250

In [ ]: