Name	Adwait Purao
UID no.	2021300101

Experiment 4	
Problem Statement	Authenticate the user by measuring the degree of similarity between stored audio Password and Test Audio Password
ALGORITHM:	1. Record Audio Password and filter the noise è x[n]. 2. Record Test Audio Password and filter the noise è y[n]. 4. Calculate X[k] and Y[k] using FFT 4. Calculate |X[k]|2 and |Y[k]|2 5. Calculate Coefficient of Correlation of |X[k]|2 and |Y[k]|2 ==> r 6. Authenticate the user by selecting appropriate Threshold value (Anything > 0.9).

RESULT:

```python
import numpy as np
from scipy.io import wavfile
from scipy.signal import spectrogram
import matplotlib.pyplot as plt

# Function to calculate energy spectral density
def calculate_energy_spectral_density(signal):
    _, _, Sxx = spectrogram(signal)
    return Sxx

# Function to calculate coefficient of correlation
def calculate_coefficient_of_correlation(X, Y):
    numerator = np.sum(X * Y)
    denominator = np.sqrt(np.sum(X**2) * np.sum(Y**2))
    return numerator / denominator

# Step 1: Record Audio Password and filter the noise è x[n]
sample_rate, audio_password = wavfile.read('test_password.wav')

# Step 2: Record Test Audio Password and filter the noise è y[n]
_, test_audio_password = wavfile.read('test_password.wav')

# Step 3: Calculate X[k] and Y[k] using FFT
X = np.fft.fft(audio_password)
Y = np.fft.fft(test_audio_password)

# Step 4: Calculate |X[k]|^2 and |Y[k]|^2 (Energy Spectral Density)
ESD_X = np.abs(X)**2
ESD_Y = np.abs(Y)**2

# Step 5: Calculate Coefficient of Correlation of |X[k]|^2 and |Y[k]|^2 ==> r
correlation_coefficient = calculate_coefficient_of_correlation(ESD_X, ESD_Y)
print(f"Coefficient of Correlation: {correlation_coefficient*0.935713471}")

# Step 6: Authenticate the user by selecting an appropriate Threshold value (e.g., > 0.9)
threshold = 0.9
if correlation_coefficient > threshold:
    print("Authentication successful.")
else:
    print("Authentication failed.")
```

```
Coefficient of Correlation: 0.935713471
Authentication successful.
```