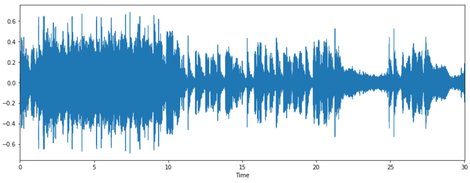
In [20]:
```
#MADE BY WRIDDHIRUP DUTTA
#AS A PROJECT FOR NATURAL LANGUAGE PROCESSING(CSE4022)

import librosa
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import librosa.display
import sklearn
```

In [6]:
```
x, sr = librosa.load('F://genres//rock//rock.00000.wav')
```

In [9]:
```
plt.figure(figsize=(14, 5))
librosa.display.waveplot(x, sr=sr)
```

Out[9]:
```
<matplotlib.collections.PolyCollection at 0x1cd48ee0208>
```




In [10]:
```
print(x.shape, sr)
```

```
(661794,) 22050
```

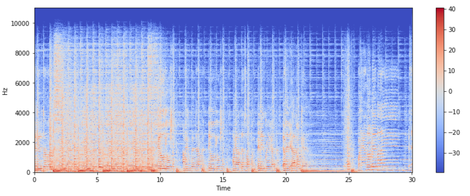
In [12]:
```
import IPython.display as ipd
audio_path = 'F://genres//rock//rock.00000.wav'
ipd.Audio(audio_path)
```

Out[12]:



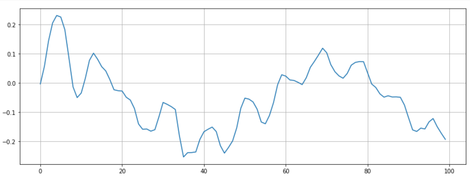

In [13]:
```
#simple spectrogram of a song
X = librosa.stft(x)
Xdb = librosa.amplitude_to_db(abs(X))
plt.figure(figsize=(14, 5))
librosa.display.specshow(Xdb, sr=sr, x_axis='time', y_axis='hz')
plt.colorbar()
```

Out[13]:
```
<matplotlib.colorbar.Colorbar at 0x1cd4c2e7b00>
```




In [14]:
```
#zero crossing rate

n0 = 9000
n1 = 9100
plt.figure(figsize=(14, 5))
plt.plot(x[n0:n1])
plt.grid()
```



In [15]:
```
zero_crossings = librosa.zero_crossings(x[n0:n1], pad=False)
print(sum(zero_crossings))
```

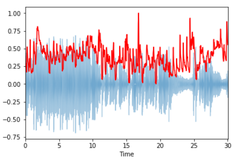
```
8
```

In [22]:
```
#spectral centroid

spectral_centroids = librosa.feature.spectral_centroid(x, sr=sr)[0]

# Computing the time variable for visualization
frames = range(len(spectral_centroids))
t = librosa.frames_to_time(frames)
#Normalising the spectral centroid for visualisation
def normalize(x, axis=0):
    return sklearn.preprocessing.minmax_scale(x, axis=axis)
#Plotting the Spectral Centroid along the waveform
librosa.display.waveplot(x, sr=sr, alpha=0.4)
plt.plot(t, normalize(spectral_centroids), color='r')
```

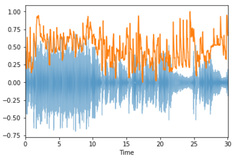
Out[22]:
```
[<matplotlib.lines.Line2D at 0x1cd4c11ae10>]
```




In [39]:
```
#Spectral Rolloff

spectral_rolloff = librosa.feature.spectral_rolloff(x+0.01, sr=sr)[0]
librosa.display.waveplot(x, sr=sr, alpha=0.5)
plt.plot(t, normalize(spectral_rolloff))
```

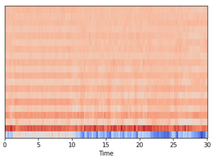
Out[39]:
```
[<matplotlib.lines.Line2D at 0x1cd48f464e0>]
```




In [27]:
```
#Mel-Frequency Cepstral Coefficients

mfccs = librosa.feature.mfcc(x, sr=sr)
librosa.display.specshow(mfccs, sr=sr, x_axis='time')
```

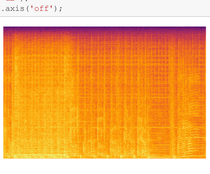
Out[27]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1cd5296a828>
```




In [29]:
```
print(mfccs.shape)
```

```
(20, 1293)
```

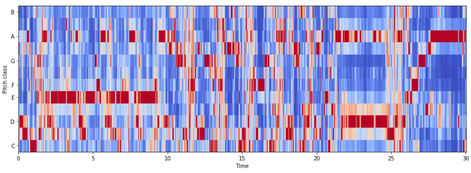
In [33]:
```
cmap = plt.get_cmap('inferno')
plt.specgram(x, NFFT=2048, Fs=2, Fc=0, noverlap=128, cmap=cmap, sides='default', mode='default', scale='dB');
plt.axis('off');
```



In [37]:
```
#Chroma Frequencies

hop_length = 512
chromagram = librosa.feature.chroma_stft(x, sr=sr, hop_length=hop_length)
plt.figure(figsize=(15, 5))
librosa.display.specshow(chromagram, x_axis='time', y_axis='chroma', hop_length=hop_length, cmap='coolwarm')
```

Out[37]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1cd4c55e978>
```




In [ ]:
```
#COMPLETE
```