In [1]:
```python
import pandas as pd

# Load songs and ratings CSV files into Pandas DataFrames
songs_df = pd.read_csv('songs.csv')
ratings_df = pd.read_csv('ratings.csv')
```

In [7]:
```python
# Rename the column 'song id' in songs_df to 'song_id'
songs_df.rename(columns={'song id': 'song_id'}, inplace=True)

# Merge datasets based on 'song_id'
merged_df = pd.merge(ratings_df, songs_df, on='song_id')
```

```
Columns in ratings_df: Index(['user_id', 'song_id', 'rating', 'unix_timestamp'], dtype='object')
Columns in songs_df: Index(['song id', 'song title', 'release date', 'video release date',
       'IMDb URL', 'unknown', 'Pop', 'Rock', 'HipHop', 'Rap', 'Electronic',
       'Country', 'Dance', 'Jazz', 'Blues', 'Reggae', 'Classical', 'R&B',
       'Funk', 'Metal', 'Indie', 'Soul', 'WorldMusic', 'Western'],
      dtype='object')
```

In [12]:
```python
# Calculate average ratings for each genre
genre_ratings = merged_df.groupby(genre_columns)['rating'].mean().reset_index()

# Find the index of the row with the highest average rating
most_popular_genre_index = genre_ratings['rating'].idxmax()

# Get the name of the most popular genre based on the highest average rating
most_popular_genre_name = genre_ratings.iloc[most_popular_genre_index][genre_columns].idxmax()

print("The most popular genre of music is:", most_popular_genre_name)
```

```
The most popular genre of music is: HipHop
```

In [14]:
```python
import seaborn as sns
import matplotlib.pyplot as plt

# Calculate correlation matrix
correlation_matrix = merged_df[genre_columns + ['rating']].corr()

# Plot heatmap
plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Heatmap between Genres and Ratings')
plt.show()
```

Correlation Heatmap between Genres and Ratings
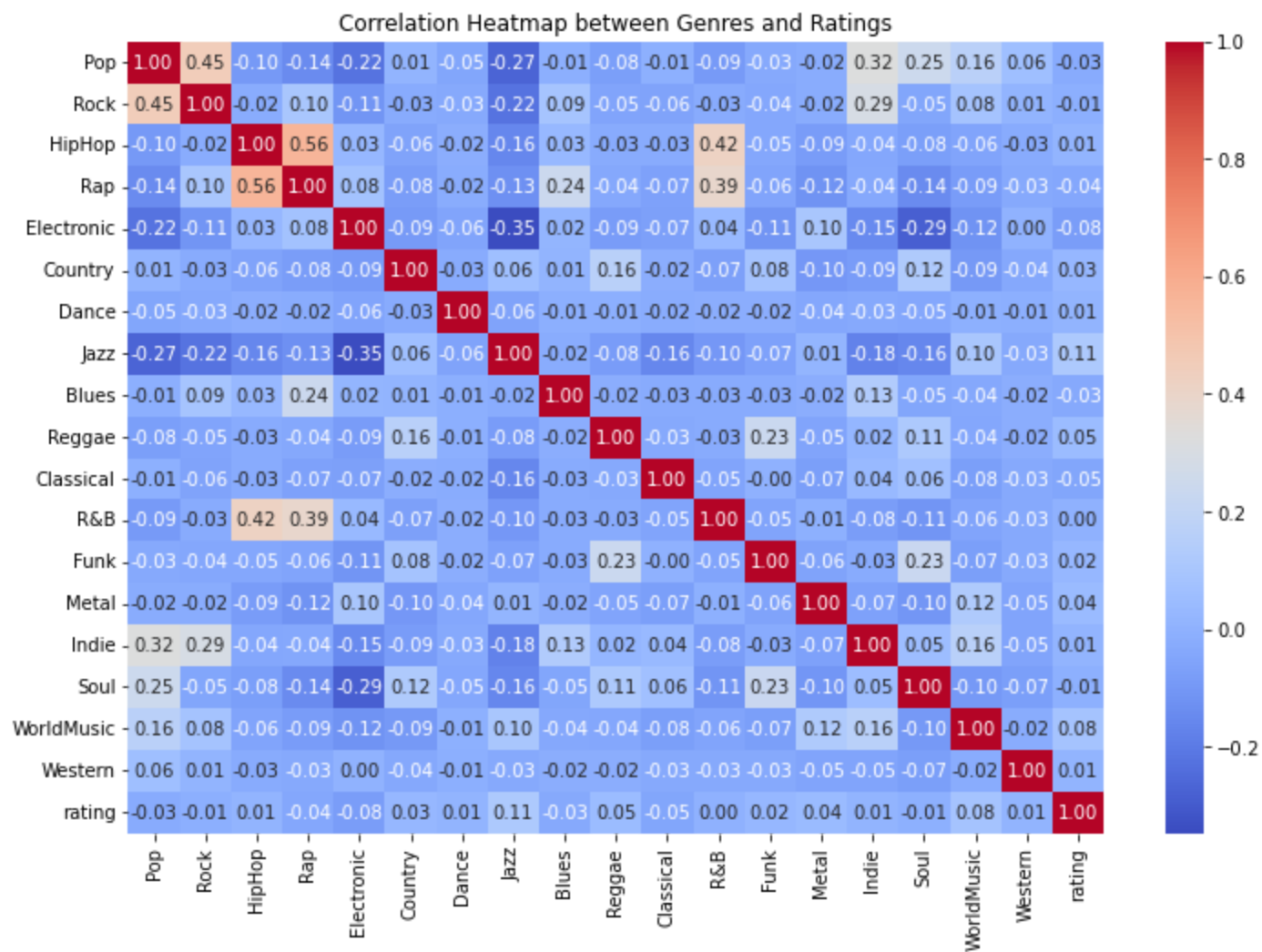
```
In [15]:  # Calculate correlation coefficients between genres and ratings
          genre_correlation = merged_df[genre_columns + ['rating']].corr()['rating']

          # Remove the self-correlation of 'rating' with itself
          genre_correlation = genre_correlation.drop('rating')

          # Plotting the correlation coefficients
          plt.figure(figsize=(10, 6))
          genre_correlation.plot(kind='bar', color='skyblue')
          plt.title('Correlation between Genres and Ratings')
          plt.xlabel('Genre')
```

```python
plt.ylabel('Correlation Coefficient')
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```
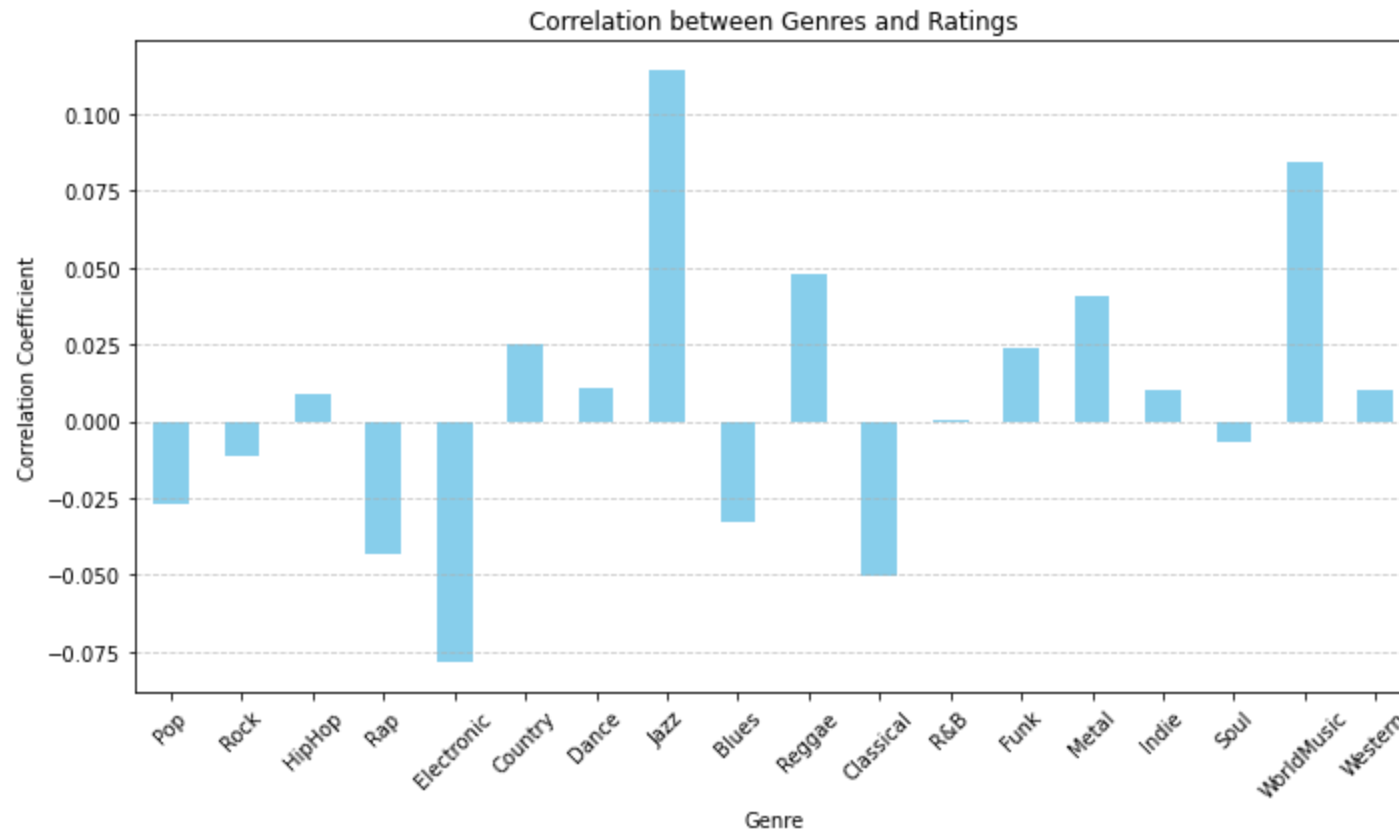

In [17]:
```python
# Display the column names in the DataFrame
print(merged_df.columns)
```

```
Index(['user_id', 'song_id', 'rating', 'unix_timestamp', 'song title',
       'release date', 'video release date', 'IMDb URL', 'unknown', 'Pop',
       'Rock', 'HipHop', 'Rap', 'Electronic', 'Country', 'Dance', 'Jazz',
       'Blues', 'Reggae', 'Classical', 'R&B', 'Funk', 'Metal', 'Indie', 'Soul',
       'WorldMusic', 'Western'],
      dtype='object')
```
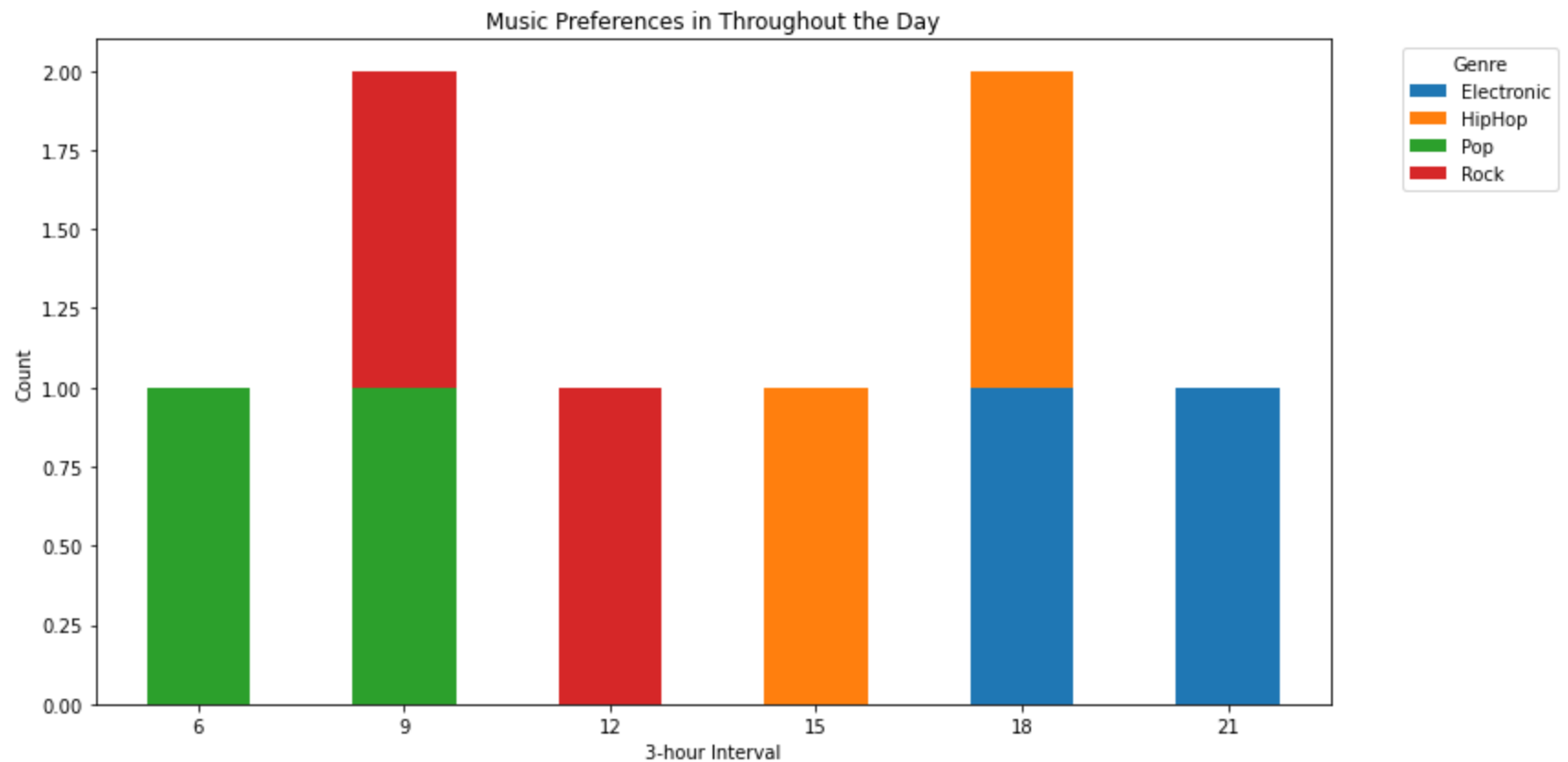
In [24]:
```python
import pandas as pd
import matplotlib.pyplot as plt
```

```python
# Sample data preparation (replace this with your dataset)
data = {
    'timestamp': ['2023-01-01 08:30:00', '2023-01-01 12:15:00', '2023-01-01 18:45:00', '2023-01-01 22:00:00',
                  '2023-01-02 09:30:00', '2023-01-02 11:45:00', '2023-01-02 17:15:00', '2023-01-02 20:30:00']
    'genre': ['Pop', 'Rock', 'HipHop', 'Electronic', 'Rock', 'Pop', 'HipHop', 'Electronic']
}
df = pd.DataFrame(data)
df['timestamp'] = pd.to_datetime(df['timestamp'])

# Extracting hour of the day
df['hour'] = df['timestamp'].dt.hour

# Grouping by 3-hour intervals and counting preferences
df['hour_range'] = (df['hour'] // 3) * 3
hourly_preference = df.groupby(['hour_range'])['genre'].value_counts().unstack().fillna(0)

# Plotting preferences across different 3-hour intervals
hourly_preference.plot(kind='bar', stacked=True, figsize=(12, 6))
plt.title('Music Preferences in Throughout the Day')
plt.xlabel('3-hour Interval')
plt.ylabel('Count')
plt.xticks(rotation=0)
plt.legend(title='Genre', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()
```

Music Preferences in Throughout the Day


```
In [26]:  import pandas as pd
          import matplotlib.pyplot as plt

          # Sample data preparation (replace this with your dataset)
          data = {
              'timestamp': ['2023-01-01 08:30:00', '2023-02-01 12:15:00', '2023-03-01 18:45:00', '2023-04-01 22:00:00',
                            '2023-05-02 09:30:00', '2023-06-02 11:45:00', '2023-07-02 17:15:00', '2023-08-02 20:30:00']
              'genre': ['Pop', 'Rock', 'HipHop', 'Electronic', 'Rock', 'Pop', 'HipHop', 'Electronic']
          }
          df = pd.DataFrame(data)
          df['timestamp'] = pd.to_datetime(df['timestamp'])

          # Extracting month
          df['month'] = df['timestamp'].dt.month

          # Grouping by month and counting preferences
          monthly_preference = df.groupby(['month'])['genre'].value_counts().unstack().fillna(0)
```

```python
# Creating subplots for each month
fig, axs = plt.subplots(2, 4, figsize=(18, 8), sharex=True, sharey=True)
months = df['month'].unique()

for i, month in enumerate(months, 1):
    ax = plt.subplot(2, 4, i)
    monthly_preference.loc[month].plot(kind='bar', stacked=True, ax=ax)
    ax.set_title(f'Month {month}')
    ax.set_xlabel('Genre')
    ax.set_ylabel('Count')
    ax.legend(title='Genre')

plt.tight_layout()
plt.show()
```