In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:
```
import warnings
warnings.filterwarnings('ignore')
```

In [3]:
```
data = pd.read_csv("players_fifa23.csv")
data.head()
```

Out[3]:

	ID	Name	FullName	Age	Height	Weight	
0	158023	L. Messi	Lionel Messi	35	169	67	https://cdn.sofifa.net/players/158/023/
1	165153	K. Benzema	Karim Benzema	34	185	81	https://cdn.sofifa.net/players/165/153/
2	188545	R. Lewandowski	Robert Lewandowski	33	185	81	https://cdn.sofifa.net/players/188/545/
3	192985	K. De Bruyne	Kevin De Bruyne	31	181	70	https://cdn.sofifa.net/players/192/985/
4	231747	K. Mbappé	Kylian Mbappé	23	182	73	https://cdn.sofifa.net/players/231/747/

5 rows × 90 columns

In [4]:
```
data.shape
```

Out[4]:
```
(18539, 90)
```

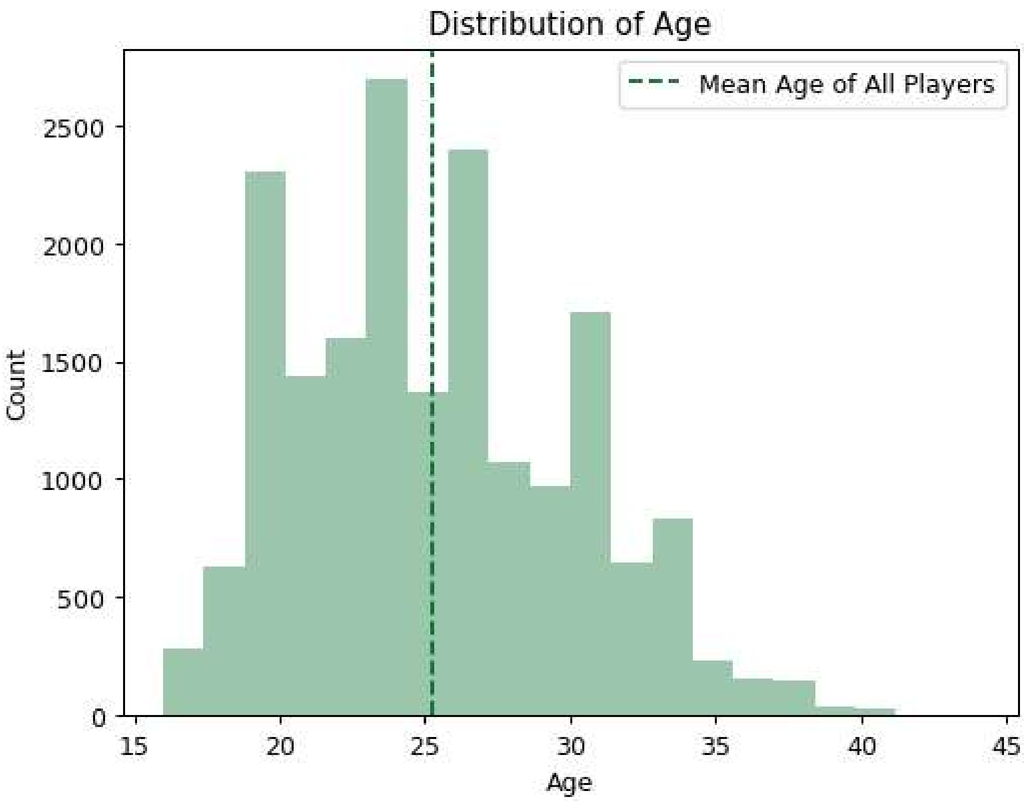
In [5]:
```
needed_columns = ['ID', 'Name', 'Age', 'Height', 'Weight',
        'Overall', 'Potential', 'Growth', 'TotalStats',
        'BaseStats', 'BestPosition', 'Club', 'ValueEUR', 'WageEUR',
        'ReleaseClause', 'ContractUntil', 'ClubJoined', 'OnLoad',
        'PreferredFoot', 'IntReputation', 'WeakFoot','Nationality',
        'SkillMoves', 'AttackingWorkRate', 'DefensiveWorkRate', 'PaceTotal',
        'ShootingTotal', 'PassingTotal', 'DribblingTotal', 'DefendingTotal',
        'PhysicalityTotal', 'Crossing', 'Finishing', 'HeadingAccuracy',
        'ShortPassing', 'Volleys', 'Dribbling', 'Curve', 'FKAccuracy',
        'LongPassing', 'BallControl', 'Acceleration', 'SprintSpeed', 'Agilit
        'Reactions', 'Balance', 'ShotPower', 'Jumping', 'Stamina', 'Strength
        'LongShots', 'Aggression', 'Interceptions', 'Positioning', 'Vision',
        'Penalties', 'Composure', 'Marking', 'StandingTackle', 'SlidingTackl
        'GKDiving', 'GKHandling', 'GKKicking', 'GKPositioning', 'GKReflexes'


data = data[needed_columns]
```

EDA

In [6]:

```
sns.set_palette("Greens_r")
plt.figure(dpi=90)
sns.distplot(x=data['Age'], kde=False, bins=20)
plt.axvline(x=np.mean(data['Age']), ls='--', label='Mean Age of All Players
plt.legend()
plt.xlabel('Age')
plt.ylabel('Count')
plt.title('Distribution of Age')
plt.show()
```

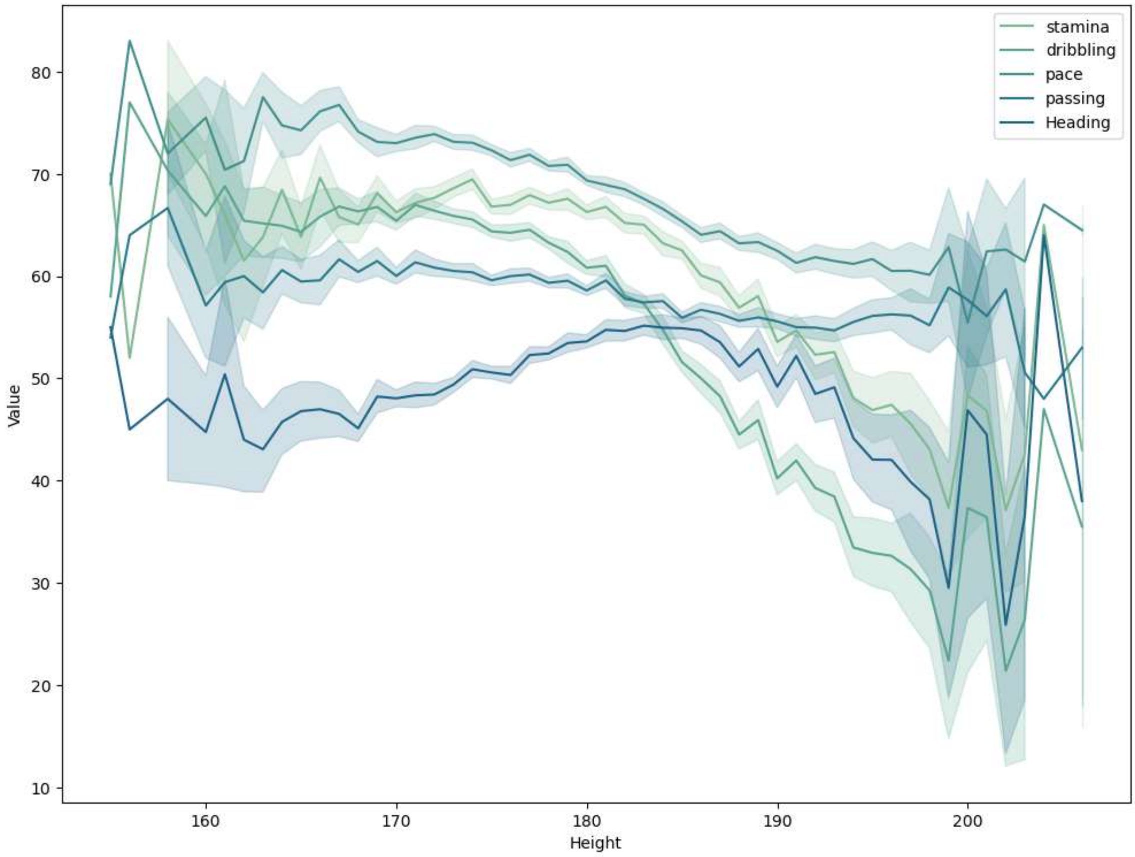


Height affects different factors like stamina, dribbling, pace, passing and HeadingAccuracy:

In [7]:
```
sns.set_palette("crest")
fig = plt.gcf()
fig.set_size_inches(12, 9)
plt.ylabel("Value")

sns.lineplot(x='Height', y='Stamina', data=data, legend='brief', label='sta
sns.lineplot(x='Height', y='Dribbling', data=data, legend='brief', label='d
sns.lineplot(x='Height', y='PaceTotal', data=data, legend='brief', label='p
sns.lineplot(x='Height', y='PassingTotal', data=data, legend='brief', label
sns.lineplot(x='Height', y='HeadingAccuracy', data=data, legend='brief', la
```

Out[7]:
```
<Axes: xlabel='Height', ylabel='Value'>
```



Top Quikest Players:

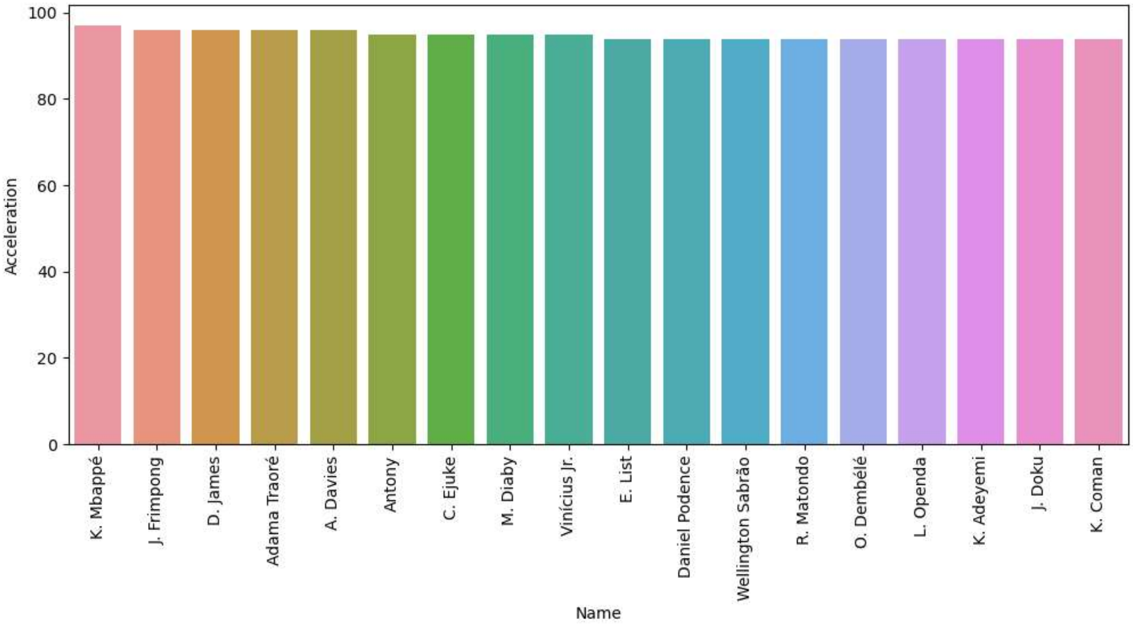
In [8]:
```
top_acc = data.sort_values(by=["Acceleration"], ascending=False)
```

In [17]:

```
plt.figure(figsize=(12, 5))
plt.xticks(rotation=90)
sns.barplot(x="Name", y = "Acceleration", data=top_acc.head(20))
```

Out[17]:

```
<Axes: xlabel='Name', ylabel='Acceleration'>
```



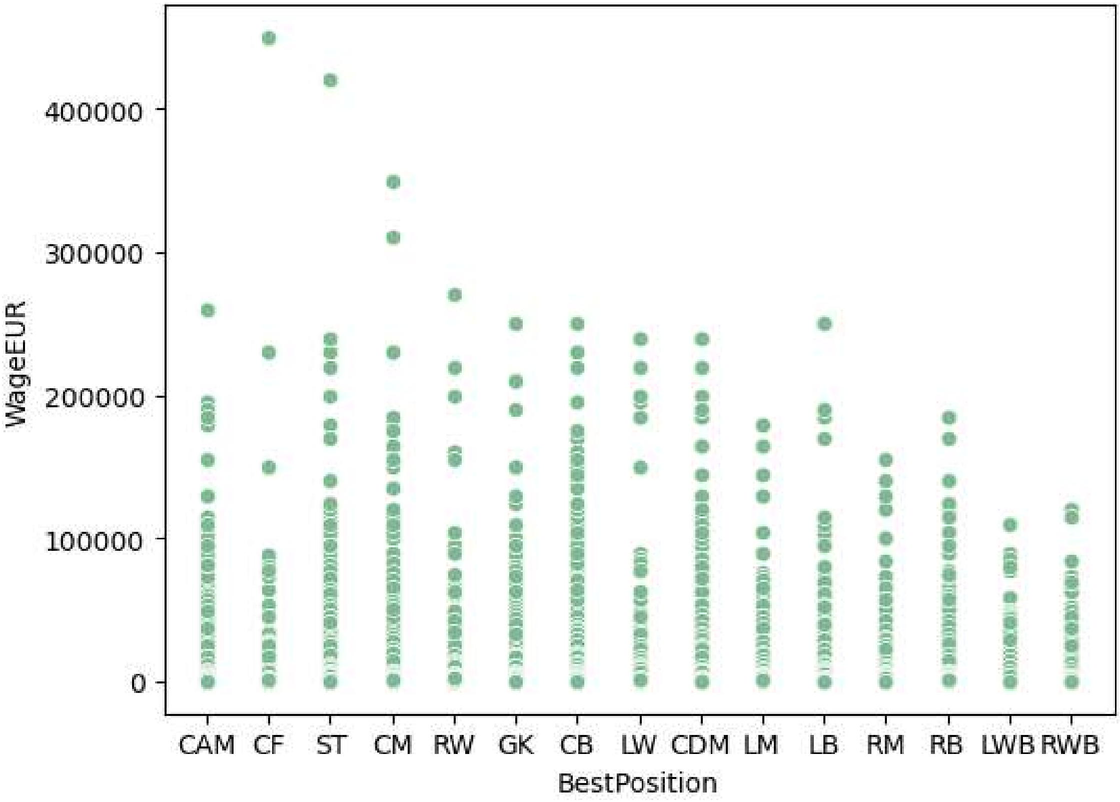
Relation between the Position of the Player and his Wage and Value

In [10]:

```
sns.scatterplot(x="BestPosition", y="WageEUR", data=data)
```

Out[10]:

```
<Axes: xlabel='BestPosition', ylabel='WageEUR'>
```

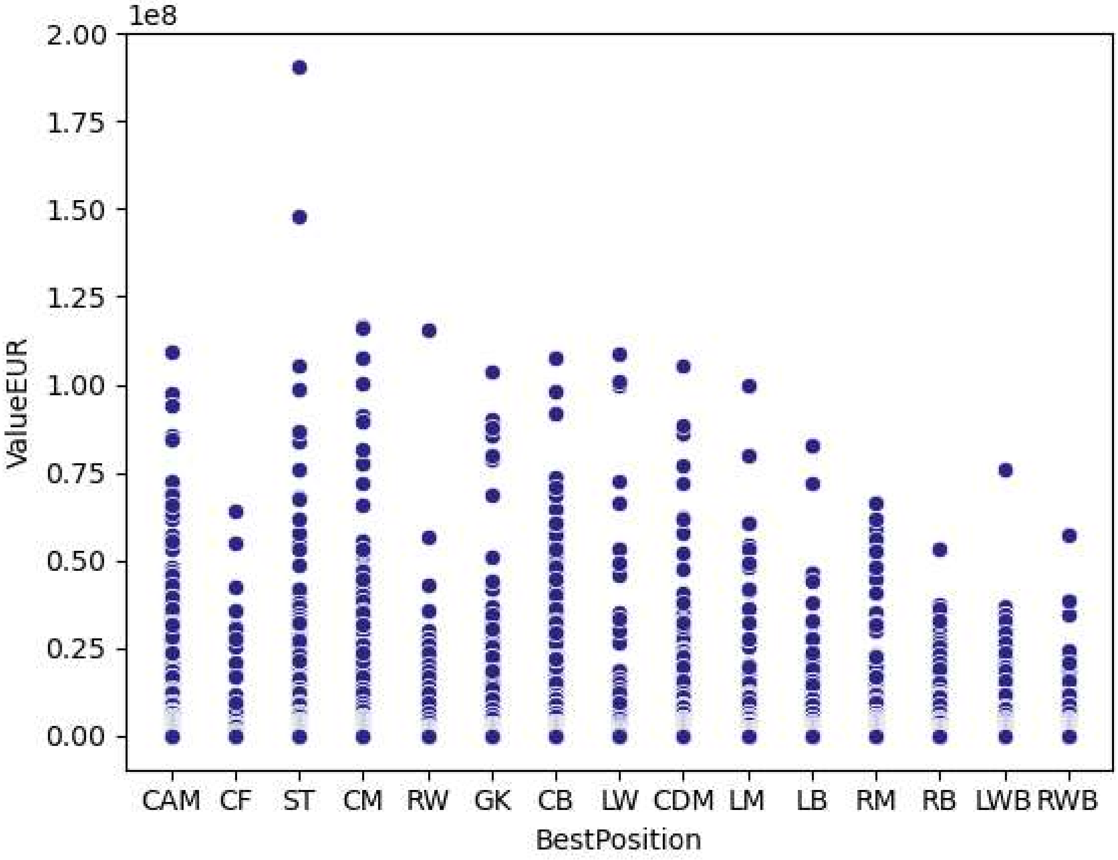


In [11]:

```
sns.set_palette("CMRmap")
sns.scatterplot(x="BestPosition", y="ValueEUR", data=data)
```

Out[11]:

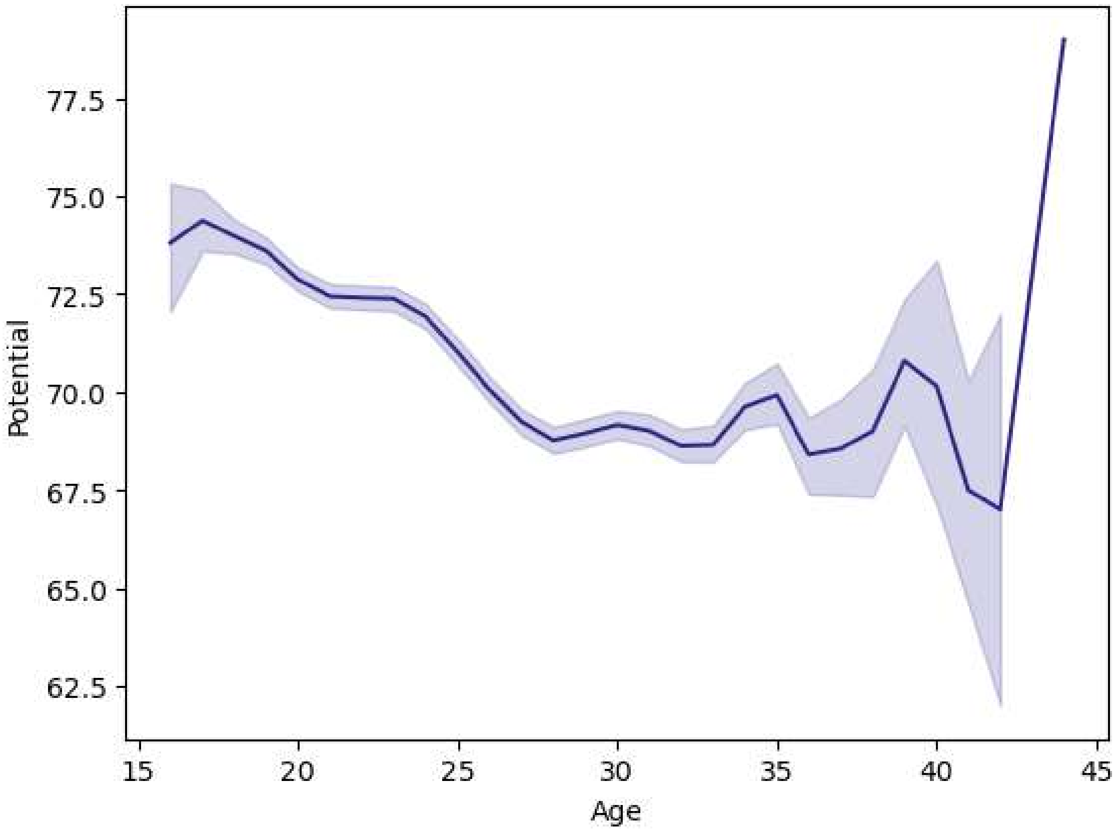
```
<Axes: xlabel='BestPosition', ylabel='ValueEUR'>
```



Effect of the Age on the Potential of the Players:

In [12]:
```
sns.lineplot(x="Age", y="Potential", data=data)
```

Out[12]:
```
<Axes: xlabel='Age', ylabel='Potential'>
```



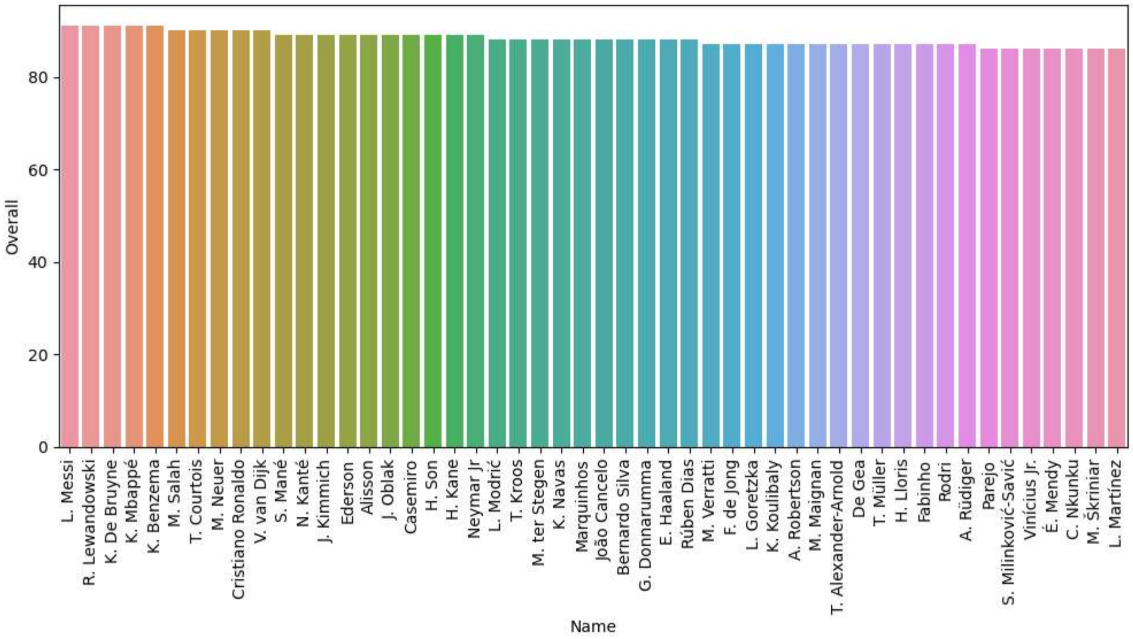
View the Top 50 Players and their Clubs:

In [13]:
```
top_rated = data.sort_values(by=["Overall"], ascending=False)
top50 = top_rated.head(50)
```

Top 50 Players

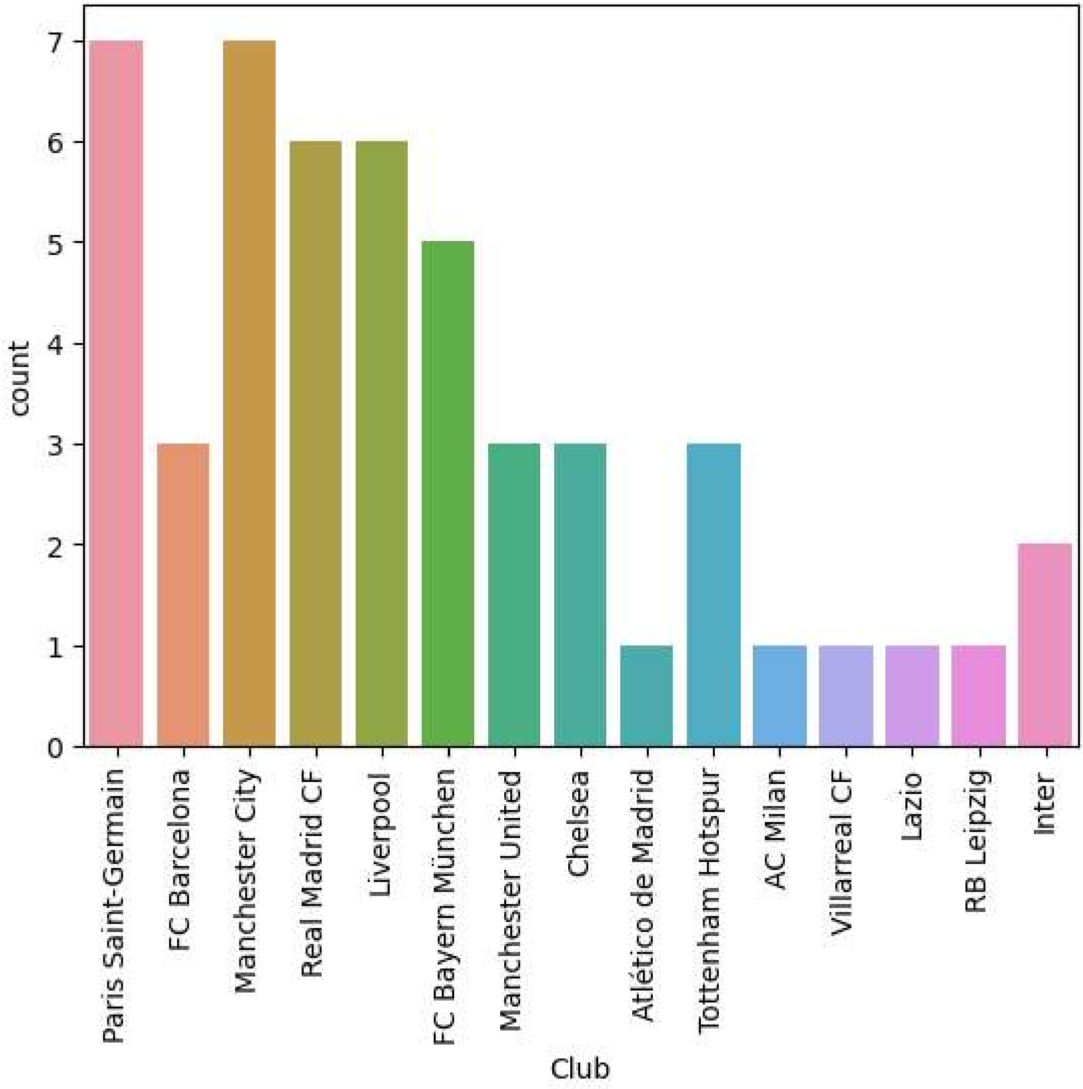
In [14]:
```
plt.figure(figsize=(12, 5))
plt.xticks(rotation=90)
sns.barplot(x="Name", y = "Overall", data=top50)
```

Out[14]:
```
<Axes: xlabel='Name', ylabel='Overall'>
```



```
In [15]: plt.figure(dpi=100)
         plt.xticks(rotation=90)
         sns.countplot(x="Club", data=top50)
```

```
Out[15]: <Axes: xlabel='Club', ylabel='count'>
```


count
Paris Saint-Germain
FC Barcelona
Manchester City
Real Madrid CF
Liverpool
FC Bayern München
Manchester United
Chelsea
Atlético de Madrid
Tottenham Hotspur
AC Milan
Villarreal CF
Lazio
RB Leipzig
Inter
Club