🏆 Sports Data Visualization

Done by – Divyansh Dwivedi

📝 Introduction

This project explores **FIFA 22 player data**, focusing exclusively on players from the **32 national teams that participated in the 2022 FIFA World Cup**. Through strategic filtering, analysis, and visualization, the aim is to uncover insights about:

- Player skill distribution
- Best players from each country
- Team-wise average overall ratings
- Optimal team formations based on player strengths

The visualizations created offer an interactive way to understand and compare teams and players, aiding in strategic decision-making, scouting analysis, and predictive analytics in sports.

📦 Step 1: Importing Libraries and Loading Data

We begin by importing the necessary libraries (`pandas` , `matplotlib.pyplot` , and `seaborn`) and loading the FIFA 22 dataset.

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
sns.set_style('darkgrid')
```

In [ ]:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
sns.set_style('darkgrid')
```

✂️ Step 2: Selecting and Cleaning the Data

We select only relevant columns (e.g., name, age, nationality, rating, positions, club, etc.), simplify player position data to retain only the primary position, and remove rows with missing values.

In [33]:

```
# nation_position, club_position, player_positions
df = pd.read_csv('players_22.csv', low_memory=False)
```

```python
# selecting column
df = df[['short_name', 'age', 'nationality_name', 'overall', 'potential',
         'club_name', 'value_eur', 'wage_eur', 'player_positions']]

# selecting only one position
df['player_positions'] = df['player_positions'].str.split(',', expand=True)[0]

# dropping nan
df.dropna(inplace=True)
```

🚫 Step 3: Removing Injured or Excluded Players

Certain prominent players missed the World Cup due to injury or other reasons. We drop them from our analysis to maintain squad accuracy.

In [5]:
```python
players_missing_worldcup = ['K. Benzema', 'S. Mané', 'S. Agüero', 'Sergio Ramos', '
                            'M. Reus', 'Diogo Jota', 'A. Harit', 'N. Kanté', 'G. Lo

# dropping injured players
drop_index = df[df['short_name'].isin(players_missing_worldcup)].index
df.drop(drop_index, axis=0, inplace=True)
```

🌍 Step 4: Filtering World Cup Teams

We filter the dataset to include only the 32 qualified teams in the 2022 FIFA World Cup.

In [7]:
```python
teams_worldcup = [
    'Qatar', 'Brazil', 'Belgium', 'France', 'Argentina', 'England', 'Spain', 'Portu
    'Mexico', 'Netherlands', 'Denmark', 'Germany', 'Uruguay', 'Switzerland', 'Unite
    'Senegal', 'Iran', 'Japan', 'Morocco', 'Serbia', 'Poland', 'South Korea', 'Tuni
    'Cameroon', 'Canada', 'Ecuador', 'Saudi Arabia', 'Ghana', 'Wales', 'Costa Rica'
]

# filtering only national teams in the world cup
df = df[df['nationality_name'].isin(teams_worldcup)]
```

In [8]:
```python
df
```

Out[8]:

	short_name	age	nationality_name	overall	potential	club_name	value_eur	w
0	L. Messi	34	Argentina	93	93	Paris Saint-Germain	78000000.0	3
1	R. Lewandowski	32	Poland	92	92	FC Bayern München	119500000.0	2
2	Cristiano Ronaldo	36	Portugal	91	91	Manchester United	45000000.0	2
3	Neymar Jr	29	Brazil	91	91	Paris Saint-Germain	129000000.0	2
4	K. De Bruyne	30	Belgium	91	91	Manchester City	125500000.0	3
...	...	...	...	...	...	...	...	
19183	F. Emmings	17	United States	48	73	Minnesota United FC	130000.0	
19197	J. Neal	17	United States	48	69	LA Galaxy	140000.0	
19216	H. Wiles-Richards	19	England	48	65	Bristol City	110000.0	
19217	J. Affonso	23	Uruguay	48	55	Cerro Largo Fútbol Club	90000.0	
19230	N. Saliba	17	Canada	47	69	Club de Foot Montréal	150000.0	

12235 rows × 9 columns

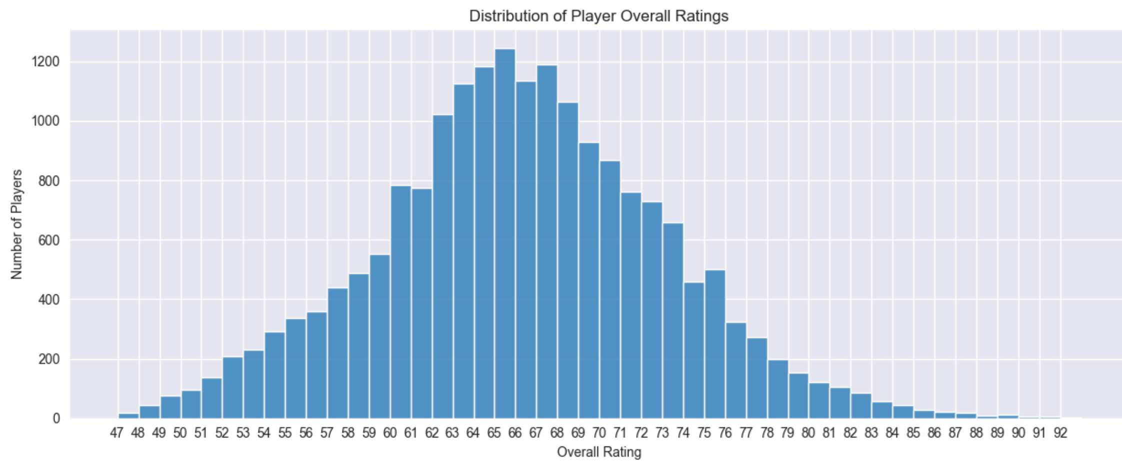
📊 Step 5: Sorting the Best Players

We sort players based on their `overall`, `potential`, and `value_eur` to prepare for deeper visual insights and to identify top-tier players.

In [21]:
```python
# best players
# Ensure it's a fresh copy if it came from slicing
df = df.copy()

# Then sort without inplace
df = df.sort_values(by=['overall', 'potential', 'value_eur'], ascending=False)
```

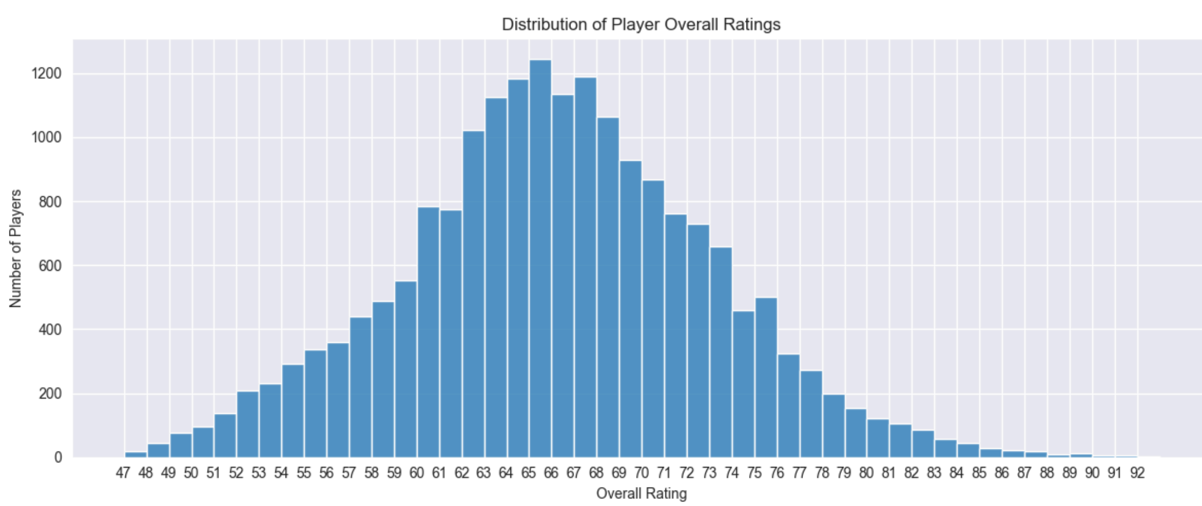
📈 Step 6: Distribution of Player Overall Ratings

We plot a histogram to visualize how player `overall ratings` are distributed. This helps identify the overall quality density among players from all teams.



In [30]:

```
import numpy as np
fig, ax = plt.subplots(figsize=(12, 5), tight_layout=True)

sns.histplot(df, x='overall', binwidth=1)
bins = np.arange(df['overall'].min(), df['overall'].max(), 1)
plt.xticks(bins)
plt.title("Distribution of Player Overall Ratings")
plt.xlabel("Overall Rating")
plt.ylabel("Number of Players")

plt.savefig("plot_1_player_overall_distribution.png")
plt.show()
```



DREAM TEAM

In [11]:

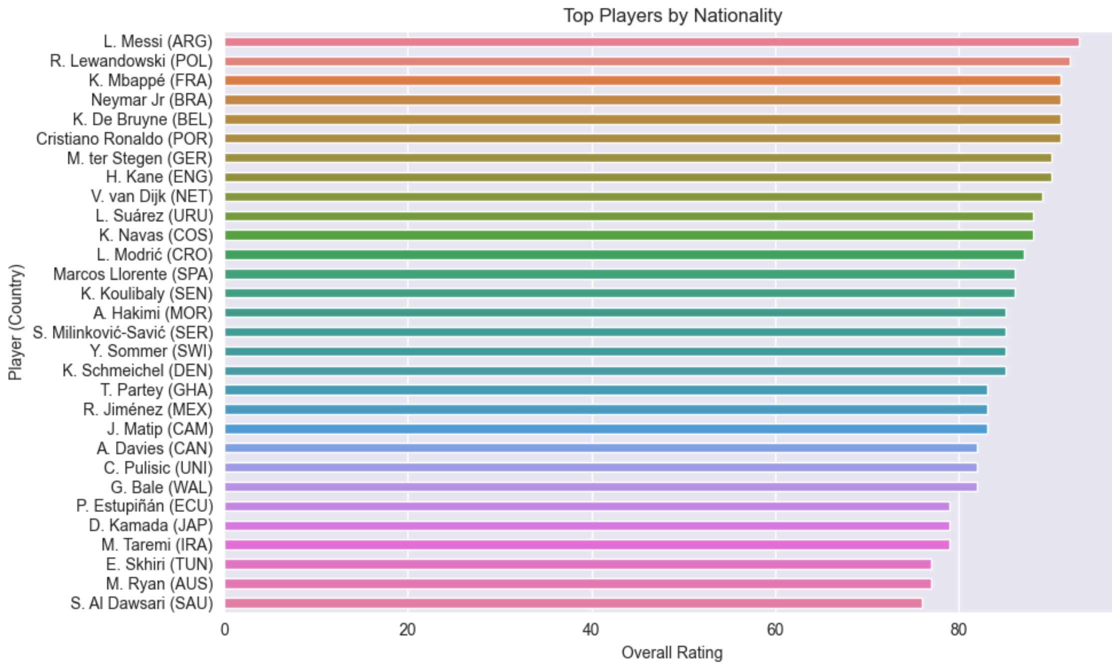
```
df.drop_duplicates('player_positions')
# viz -> https://trinket.io/python/0813ea96f6
```

Out[11]:

	short_name	age	nationality_name	overall	potential	club_name	value_eur	wag
0	L. Messi	34	Argentina	93	93	Paris Saint-Germain	78000000.0	320
1	R. Lewandowski	32	Poland	92	92	FC Bayern München	119500000.0	270
3	Neymar Jr	29	Brazil	91	91	Paris Saint-Germain	129000000.0	270
4	K. De Bruyne	30	Belgium	91	91	Manchester City	125500000.0	350
8	M. ter Stegen	29	Germany	90	92	FC Barcelona	99000000.0	250
19	J. Kimmich	26	Germany	89	90	FC Bayern München	108000000.0	160
15	V. van Dijk	29	Netherlands	89	89	Liverpool	86000000.0	230
28	Bruno Fernandes	26	Portugal	88	89	Manchester United	107500000.0	250
44	T. Alexander-Arnold	22	England	87	92	Liverpool	114000000.0	150
45	J. Sancho	21	England	87	91	Manchester United	116500000.0	150
41	P. Dybala	27	Argentina	87	88	Juventus	93000000.0	160
64	K. Coman	25	France	86	87	FC Bayern München	81000000.0	120
50	Jordi Alba	32	Spain	86	86	FC Barcelona	47000000.0	200
180	Angeliño	24	Spain	83	86	RB Leipzig	46000000.0	77
379	R. James	21	England	81	86	Chelsea	37000000.0	76

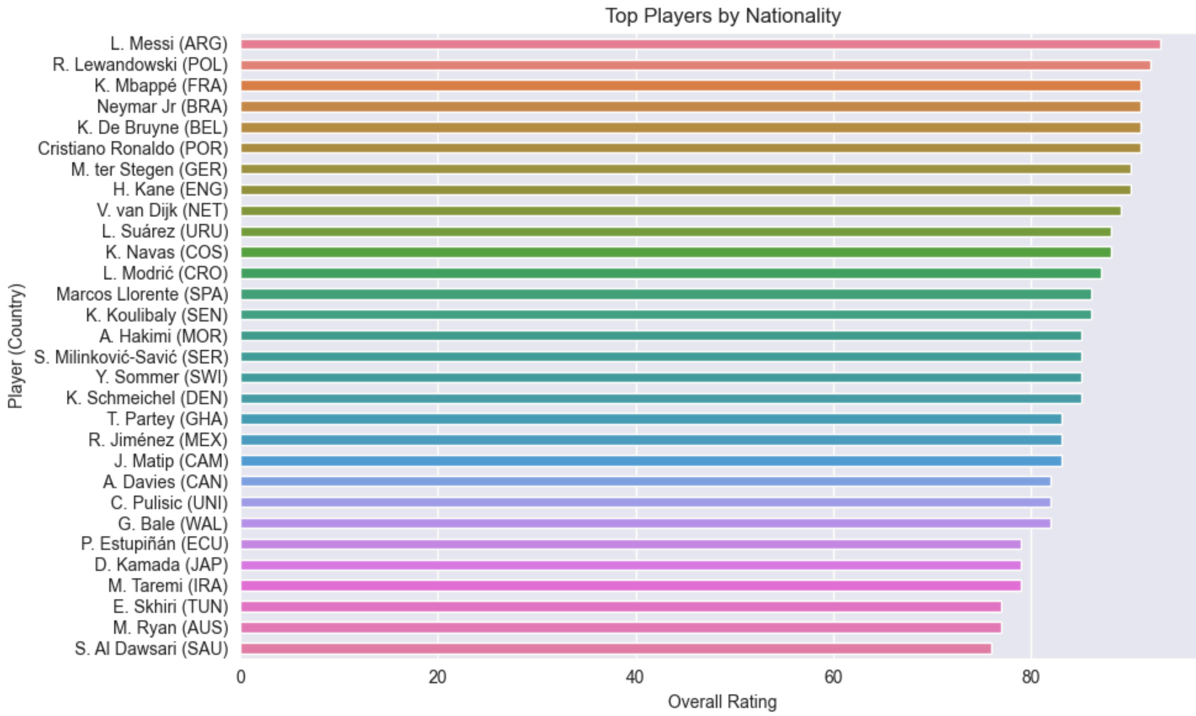
UN Step 7: Best Player from Each Country

We extract the top player from each nation based on `overall rating` and visualize them with a colorful bar chart.



In [22]:

```
df_best_players = df.copy()
df_best_players = df_best_players.drop_duplicates('nationality_name').reset_index(d

country_short = df_best_players['nationality_name'].str.extract(r'(^\w{3})', expand
df_best_players['name_nationality'] = df_best_players['short_name'] + ' (' + countr

fig, ax = plt.subplots(figsize=(10, 6), tight_layout=True)
sns.barplot(
    data=df_best_players,
    x='overall',
    y='name_nationality',
    hue='name_nationality',
    palette=sns.color_palette('husl', n_colors=len(df_best_players)),
    width=0.5,
    legend=False
)
plt.title("Top Players by Nationality")
plt.xlabel("Overall Rating")
plt.ylabel("Player (Country)")
plt.show()
```



🧠 Step 8: Best Squad per Country (by Position)

We define a function `best_squad()` that selects the top two players for each position within a country. This helps model realistic team selection patterns and ensures all key roles are considered.

In [13]:
```
def best_squad(nationality):
    df_best_squad = df.copy()
    df_best_squad = df_best_squad.groupby(['nationality_name', 'player_positions'])
    df_best_squad = df_best_squad[df_best_squad['nationality_name']==nationality].s
    return df_best_squad
```

In [14]:
```
best_squad('Brazil')
```

Out[14]:

	short_name	age	nationality_name	overall	potential	club_name	value_eur	wag
191	Gabriel Jesus	24	Brazil	83	87	Manchester City	52500000.0	150
268	Richarlison	24	Brazil	82	87	Everton	46500000.0	100
5069	Paolinho Leima	21	Brazil	70	70	Clube Atlético Mineiro	1700000.0	12
8031	Jadenilson Baia	33	Brazil	67	67	Sport Club Corinthians Paulista	525000.0	9
662	Antony	21	Brazil	79	88	Ajax	39500000.0	17
656	Rodrygo	20	Brazil	79	88	Real Madrid CF	38500000.0	115
271	Raphinha	24	Brazil	82	87	Leeds United	46000000.0	89
318	Lucas Moura	28	Brazil	81	81	Tottenham Hotspur	26000000.0	105
311	Danilo	29	Brazil	81	81	Juventus	22500000.0	83
484	Maikel Catarino	25	Brazil	80	80	Sport Club Corinthians Paulista	21000000.0	33
367	Adryan Zonta	29	Brazil	81	81	RB Bragantino	22500000.0	24
7248	Vitinho	21	Brazil	68	77	KSV Cercle Brugge	2600000.0	4
3	Neymar Jr	29	Brazil	91	91	Paris Saint-Germain	129000000.0	270
499	Vinícius Jr.	20	Brazil	80	90	Real Madrid CF	46500000.0	120
465	Everton	25	Brazil	80	83	SL Benfica	28000000.0	17
727	Felipe Anderson	28	Brazil	78	78	Lazio	14000000.0	58
153	Alex Sandro	30	Brazil	83	83	Juventus	31500000.0	95
245	Alex Telles	28	Brazil	82	82	Manchester United	27500000.0	130
18	Ederson	27	Brazil	89	91	Manchester City	94000000.0	200
20	Alisson	28	Brazil	89	90	Liverpool	82000000.0	190
190	Arthur	24	Brazil	83	85	Juventus	47000000.0	90

	short_name	age	nationality_name	overall	potential	club_name	value_eur	wag
149	Paulinho	32	Brazil	83	83	Al Ahli	28500000.0	6
85	Roberto Firmino	29	Brazil	85	85	Liverpool	54000000.0	18
246	Anderson Talisca	27	Brazil	82	83	Al Nassr	35500000.0	6
14	Casemiro	29	Brazil	89	89	Real Madrid CF	88000000.0	31
61	Fabinho	27	Brazil	86	88	Liverpool	73500000.0	16
39	Marquinhos	27	Brazil	87	90	Paris Saint-Germain	90500000.0	13
71	Thiago Silva	36	Brazil	85	85	Chelsea	9500000.0	10
189	Ronaldo Cabrais	29	Brazil	83	83	Grêmio	35500000.0	4
210	Oscar	29	Brazil	82	82	Shanghai Port FC	30000000.0	3

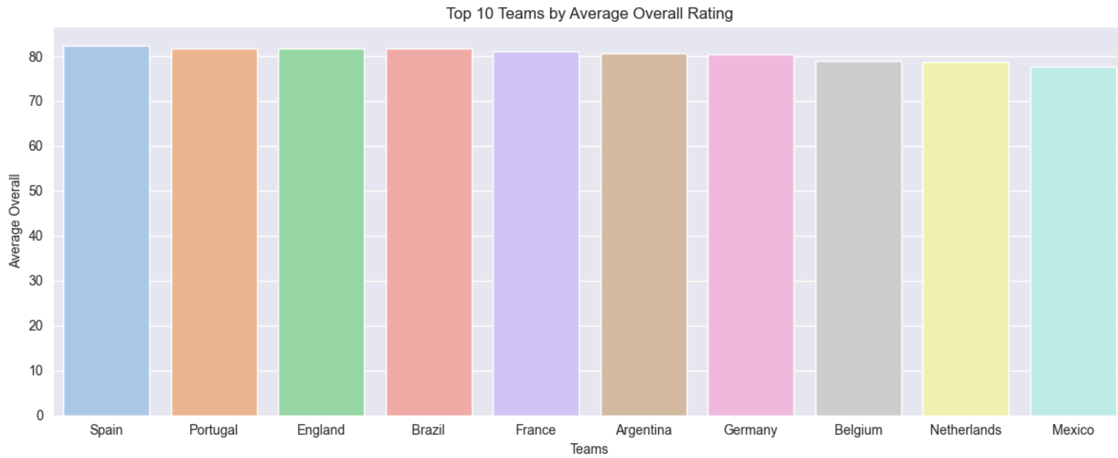
In [15]:
```
average_overall = [best_squad(team)['overall'].mean() for team in teams_worldcup]

df_average_overall = pd.DataFrame({'Teams': teams_worldcup, 'AVG_Overall': average_
df_average_overall = df_average_overall.dropna()
df_average_overall = df_average_overall.sort_values('AVG_Overall', ascending=False)
df_average_overall
```

Out[15]:

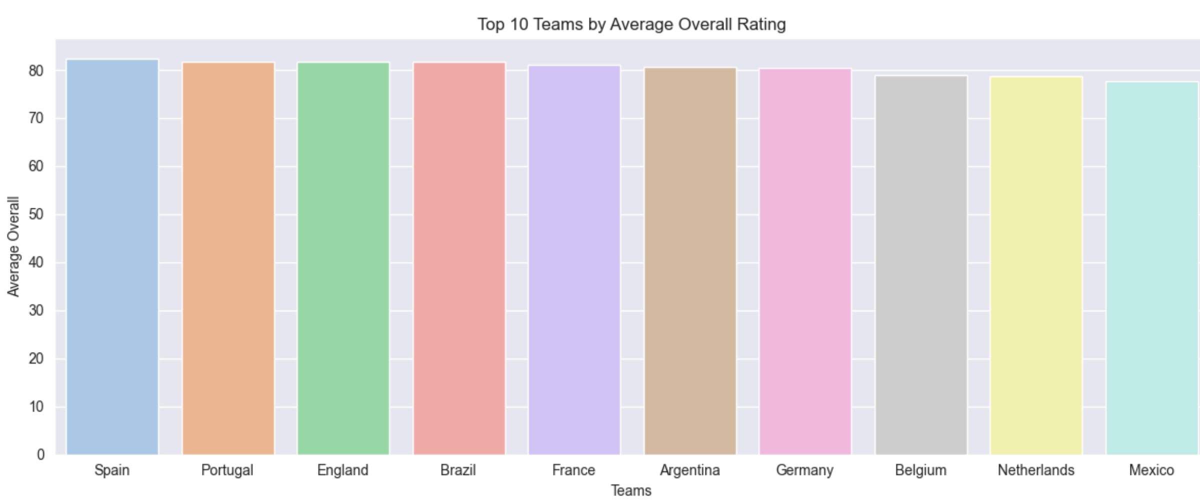
	Teams	AVG_Overall
6	Spain	82.400000
7	Portugal	81.733333
5	England	81.700000
1	Brazil	81.666667
3	France	81.000000
4	Argentina	80.566667
11	Germany	80.433333
2	Belgium	79.034483
9	Netherlands	78.758621
8	Mexico	77.727273
15	Croatia	76.760000
12	Uruguay	76.692308
20	Serbia	76.260870
19	Morocco	75.920000
10	Denmark	75.133333
16	Senegal	74.727273
13	Switzerland	74.535714
18	Japan	73.592593
14	United States	73.259259
21	Poland	73.111111
28	Ghana	72.777778
24	Cameroon	72.578947
26	Ecuador	71.076923
29	Wales	70.821429
30	Costa Rica	70.466667
31	Australia	70.214286
17	Iran	69.705882
25	Canada	68.840000
23	Tunisia	68.578947
27	Saudi Arabia	68.375000

📊 Step 9: Average Team Rating Comparison

We calculate and visualize the **average overall ratings** of each national team's best players, sorted to highlight the strongest squads.



In [32]:
```
fig, ax = plt.subplots(figsize=(12, 5), tight_layout=True)

sns.barplot(
    data=df_average_overall[:10],
    x='Teams',
    y='AVG_Overall',
    hue='Teams',
    palette=sns.color_palette('pastel'),
    legend=False
)

plt.title("Top 10 Teams by Average Overall Rating")
plt.xlabel("Teams")
plt.ylabel("Average Overall")

plt.savefig("plot_3_top10_teams_avg_rating.png")
plt.show()
```



⚽ Step 10: Finding the Best Formation for Each Team

Using a dictionary of standard football formations (`4-3-3` , `4-4-2` , `4-2-3-1`), we define a function `best_lineup()` that evaluates which formation results in the highest average overall rating for a given country.

In [17]:
```
def best_lineup(nationality, lineup):
    lineup_count = [lineup.count(i) for i in lineup]

    df_lineup = pd.DataFrame({'position': lineup, 'count': lineup_count})
    positions_non_repeated = df_lineup[df_lineup['count'] <= 1]['position'].values
    positions_repeated = df_lineup[df_lineup['count'] > 1]['position'].values

    df_squad = best_squad(nationality)

    df_lineup = pd.concat([
        df_squad[df_squad['player_positions'].isin(positions_non_repeated)].drop_du
        df_squad[df_squad['player_positions'].isin(positions_repeated)]]
    )
    return df_lineup[['short_name', 'overall', 'club_name', 'player_positions']]
```

In [18]:
```
dict_formation = {
    '4-3-3': ['GK', 'RB', 'CB', 'CB', 'LB', 'CDM', 'CM', 'CAM', 'RW', 'ST', 'LW'],
    '4-4-2': ['GK', 'RB', 'CB', 'CB', 'LB', 'RM', 'CM', 'CM', 'LM', 'ST', 'ST'],
    '4-2-3-1': ['GK', 'RB', 'CB', 'CB', 'LB', 'CDM', 'CDM', 'CAM', 'CAM', 'CAM', 'S
}
```

In [19]:
```
for index, row in df_average_overall[:9].iterrows():
    max_average = None
    for key, values in dict_formation.items():
        average = best_lineup(row['Teams'], values)['overall'].mean()
        if max_average is None or average>max_average:
            max_average = average
            formation = key
    print(row['Teams'], formation, max_average)
```

```
Spain 4-2-3-1 85.1
Portugal 4-2-3-1 84.9
England 4-4-2 84.45454545454545
Brazil 4-3-3 84.81818181818181
France 4-2-3-1 83.9
Argentina 4-3-3 83.54545454545455
Germany 4-2-3-1 84.1
Belgium 4-3-3 82.54545454545455
Netherlands 4-4-2 82.54545454545455
```

In [20]:
```
best_lineup('Brazil', dict_formation['4-3-3'])
```

Out[20]:

	short_name	overall	club_name	player_positions
191	Gabriel Jesus	83	Manchester City	ST
662	Antony	79	Ajax	RW
311	Danilo	81	Juventus	RB
3	Neymar Jr	91	Paris Saint-Germain	LW
153	Alex Sandro	83	Juventus	LB
18	Ederson	89	Manchester City	GK
190	Arthur	83	Juventus	CM
14	Casemiro	89	Real Madrid CF	CDM
189	Ronaldo Cabrais	83	Grêmio	CAM
39	Marquinhos	87	Paris Saint-Germain	CB
71	Thiago Silva	85	Chelsea	CB

In [28]:
```
best_lineup('Spain', dict_formation['4-2-3-1'])
```

Out[28]:

	short_name	overall	club_name	player_positions
59	Gerard Moreno	86	Villarreal CF	ST
87	Carvajal	85	Real Madrid CF	RB
50	Jordi Alba	86	FC Barcelona	LB
106	De Gea	84	Manchester United	GK
67	Rodri	86	Manchester City	CDM
52	Sergio Busquets	86	FC Barcelona	CDM
22	Sergio Ramos	88	Paris Saint-Germain	CB
63	A. Laporte	86	Manchester City	CB
72	David Silva	85	Real Sociedad	CAM
108	Luis Alberto	84	Lazio	CAM

In [29]:
```
best_lineup('Argentina', dict_formation['4-3-3'])
```

Out[29]:

	short_name	overall	club_name	player_positions
30	S. Agüero	87	FC Barcelona	ST
0	L. Messi	93	Paris Saint-Germain	RW
818	G. Montiel	78	Sevilla FC	RB
171	L. Ocampos	83	Sevilla FC	LW
134	M. Acuña	84	Sevilla FC	LB
113	E. Martínez	84	Aston Villa	GK
247	R. De Paul	82	Atlético de Madrid	CM
206	É. Banega	82	Al Shabab	CDM
69	A. Gómez	85	Sevilla FC	CAM
269	C. Romero	82	Tottenham Hotspur	CB
302	N. Otamendi	81	SL Benfica	CB

✅ Conclusion

This sports data visualization project presents a structured approach to analyzing FIFA 22 player data with an emphasis on:

- **Player quality distribution**
- **Top-performing individuals per country**
- **Team-wise comparative strength**
- **Optimal formations for maximizing performance**

Through this data-driven framework, we not only uncover valuable insights but also pave the way for deeper explorations like predictive modeling, player scouting automation, or game strategy optimization. The analysis is modular and can be extended further with advanced analytics such as clustering, regression, or simulation-based modeling.

> **Author**: Divyansh Dwivedi
> **Tools Used**: Python, Pandas, Seaborn, Matplotlib
> **Dataset**: FIFA 22 Player Stats

In [ ]:

```

```