In [1]:

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```python
car_data = pd.read_csv('new_car.csv')
```

In [3]:

```python
car_data.head()
```

Out[3]:

	Car	Style	Range	Transmission	VehicleType	PriceRange	Rating(out
0	Tata Nexon EV	Compact SUV	312 Km/Full Charge	Automatic	Electric	₹ 13.99 - 17.4 L	
1	Hyundai Grand i10 Nios	Compact Hatchback	20.2 - 26.2 Km/l	AMT,Manual	Petrol,Petrol+CNG,Diesel	₹ 5.3 - 8.51 L	
2	Maruti Suzuki Alto 800	Entry Hatchback	24.7 - 31.39 Km/l	Manual	Petrol,Petrol+CNG	₹ 3.39 - 5.03 L	
3	Tata Punch	Compact SUV	NaN	AMT,Manual	Petrol	₹ 5.83 - 9.49 L	
4	MG Hector	SUV	13.96 - 17.4 Km/l	Automatic,Manual	Petrol,Diesel,Hybrid	₹ 13.5 - 19.36 L	

In [4]:

```python
car_data.tail()
```

Out[4]:

	Car	Style	Range	Transmission	VehicleType	PriceRange	Rating(outof10)
250	Porsche Panamera	Premium Sports Sedan	8 Km/l	Automatic	Petrol	₹ 1.45 - 2.43 Cr	6.6
251	Ferrari 812 Superfast	Premium Sports Coupe	6.7 Km/l	Automatic	Petrol	₹ 5.2 Cr	na
252	Ferrari California	Sports Car	9.5 Km/l	Automatic	Petrol	₹ 3.8 Cr	7.9
253	Ferrari Portofino	Premium Sports Coupe	10 Km/l	Automatic	Petrol	₹ 3.5 Cr	na
254	MINI SE	Luxury Hatchback	270 Km/Full Charge	Automatic	Electric	₹ 47.2 L	na

In [5]:

```python
car_data.shape
```

Out[5]:

```
(255, 7)
```

In [6]:

```python
car_data.columns
```

Out[6]:

```
Index(['Car', 'Style', 'Range', 'Transmission', 'VehicleType', 'PriceRang
e',
       'Rating(outof10)'],
      dtype='object')
```

In [7]:

```python
car_data.nunique()
```

Out[7]:

```
Car                255
Style               31
Range              202
Transmission         5
VehicleType         15
PriceRange         250
Rating(outof10)     31
dtype: int64
```

In [8]:

```python
car_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 255 entries, 0 to 254
Data columns (total 7 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Car              255 non-null    object
 1   Style            255 non-null    object
 2   Range            244 non-null    object
 3   Transmission     255 non-null    object
 4   VehicleType      255 non-null    object
 5   PriceRange       255 non-null    object
 6   Rating(outof10)  255 non-null    object
dtypes: object(7)
memory usage: 14.1+ KB
```

In [9]:

```python
car_data.isnull().sum()
```

Out[9]:

```
Car                0
Style              0
Range             11
Transmission       0
VehicleType        0
PriceRange         0
Rating(outof10)    0
dtype: int64
```

In [10]:

```python
car_data['Style'].unique()
```

Out[10]:

```
array(['Compact SUV', 'Compact Hatchback', 'Entry Hatchback', 'SUV',
       'Subcompact SUV', 'Premium Hatchback', 'Compact MPV',
       'Subcompact Sedan', 'Compact Sedan', 'Premium Fullsize SUV',
       'Luxury SUV', 'Fullsize/Premium MPV', 'Supersports Car',
       'Quadricycle', 'Subcompact MPV', 'Premium Midsize Sedan',
       'Premium Luxury Sedan', 'Premium Compact Sedan',
       'Compact SUV/Crossover', 'Midsize Sedan', 'Premium Coupe',
       'Midsize SUV', 'Performance Sedan', 'Premium Midsize SUV',
       'Premium Fullsize Sedan', 'Luxury Hatchback', 'Sports Car',
       'Premium Sports Sedan', 'Premium Sports Coupe',
       'Premium Compact SUV', 'Fullsize Sedan'], dtype=object)
```

In [11]:

```python
car_data['Style'].value_counts()
```

Out[11]:

```
SUV                      27
Compact SUV              19
Premium Fullsize SUV     19
Premium Sports Coupe     15
Luxury SUV               15
Premium Midsize Sedan    13
Sports Car               12
Subcompact SUV           11
Premium Coupe            11
Premium Luxury Sedan     11
Entry Hatchback          10
Premium Midsize SUV       9
Premium Hatchback         8
Compact Sedan             7
Luxury Hatchback          7
Premium Sports Sedan      7
Premium Fullsize Sedan    7
Compact Hatchback         6
Supersports Car           6
Subcompact Sedan          5
Fullsize/Premium MPV      5
Subcompact MPV            4
Compact SUV/Crossover     4
Premium Compact SUV       3
Compact MPV               3
Performance Sedan         3
Midsize Sedan             2
Premium Compact Sedan     2
Fullsize Sedan            2
Midsize SUV               1
Quadricycle               1
Name: Style, dtype: int64
```
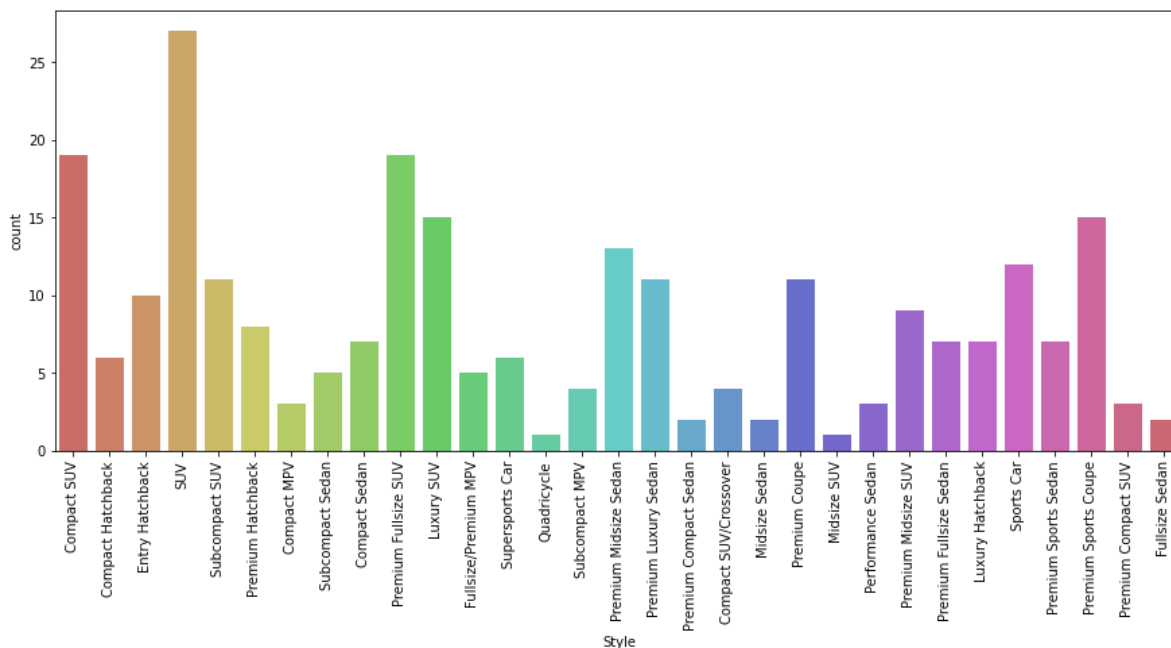
In [12]:

```python
plt.figure(figsize=(15,6))
sns.countplot('Style', data = car_data, palette='hls')
plt.xticks(rotation = 90)
plt.show()
```


In [13]:

```python
car_data['Transmission'].unique()
```

Out[13]:

```
array(['Automatic', 'AMT,Manual', 'Manual', 'Automatic,Manual',
       'AMT,Automatic,Manual'], dtype=object)
```
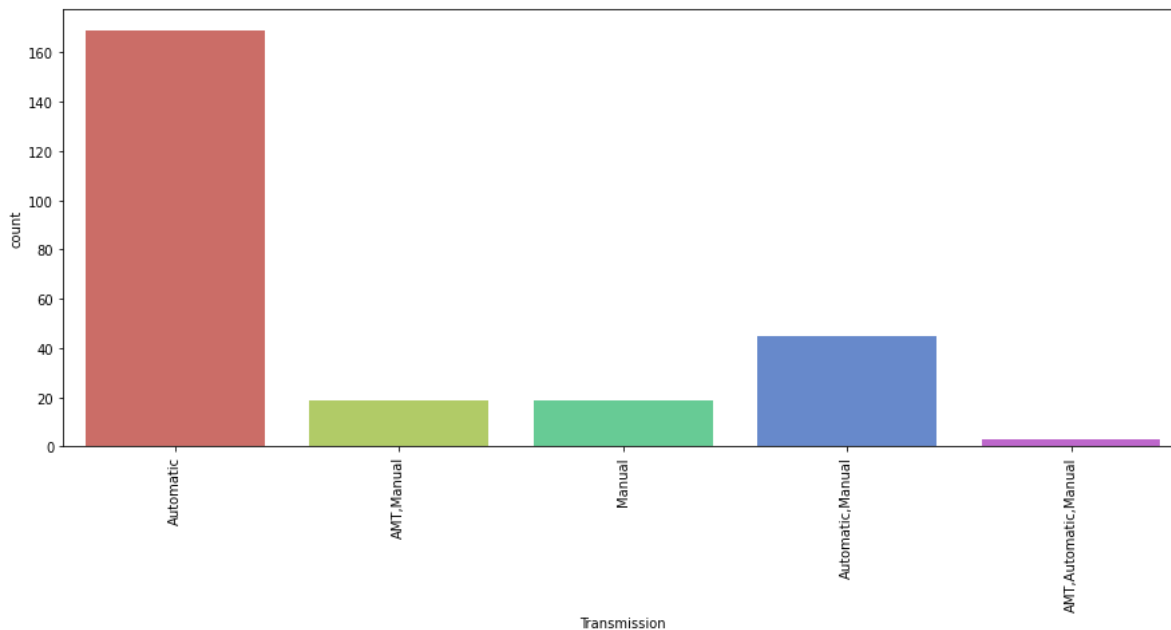
In [14]:

```python
car_data['Transmission'].value_counts()
```

Out[14]:

```
Automatic               169
Automatic,Manual         45
AMT,Manual               19
Manual                   19
AMT,Automatic,Manual      3
Name: Transmission, dtype: int64
```

In [15]:

```python
plt.figure(figsize=(15,6))
sns.countplot('Transmission', data = car_data, palette='hls')
plt.xticks(rotation = 90)
plt.show()
```


In [16]:

```python
car_data['VehicleType'].unique()
```

Out[16]:

```
array(['Electric', 'Petrol,Petrol+CNG,Diesel', 'Petrol,Petrol+CNG',
       'Petrol', 'Petrol,Diesel,Hybrid', 'Petrol,Diesel', 'Diesel',
       'Hybrid', 'Diesel,Petrol+CNG', 'Petrol,Petrol+LPG',
       'Diesel,Petrol', 'Diesel,Petrol,Hybrid', 'Hybrid,Petrol,Diesel',
       'Petrol,Hybrid', 'Diesel,Hybrid'], dtype=object)
```

In [17]:

```python
car_data['VehicleType'].value_counts()
```

Out[17]:

```
Petrol                      127
Petrol,Diesel                46
Diesel                       28
Electric                     15
Petrol,Petrol+CNG            10
Diesel,Petrol                10
Hybrid                        8
Petrol,Petrol+CNG,Diesel      2
Petrol,Diesel,Hybrid          2
Petrol,Hybrid                 2
Diesel,Petrol+CNG             1
Petrol,Petrol+LPG             1
Diesel,Petrol,Hybrid          1
Hybrid,Petrol,Diesel          1
Diesel,Hybrid                 1
Name: VehicleType, dtype: int64
```

In [18]:

```python
plt.figure(figsize=(15,6))
sns.countplot('VehicleType', data = car_data, palette='hls')
plt.xticks(rotation = 90)
plt.show()
```
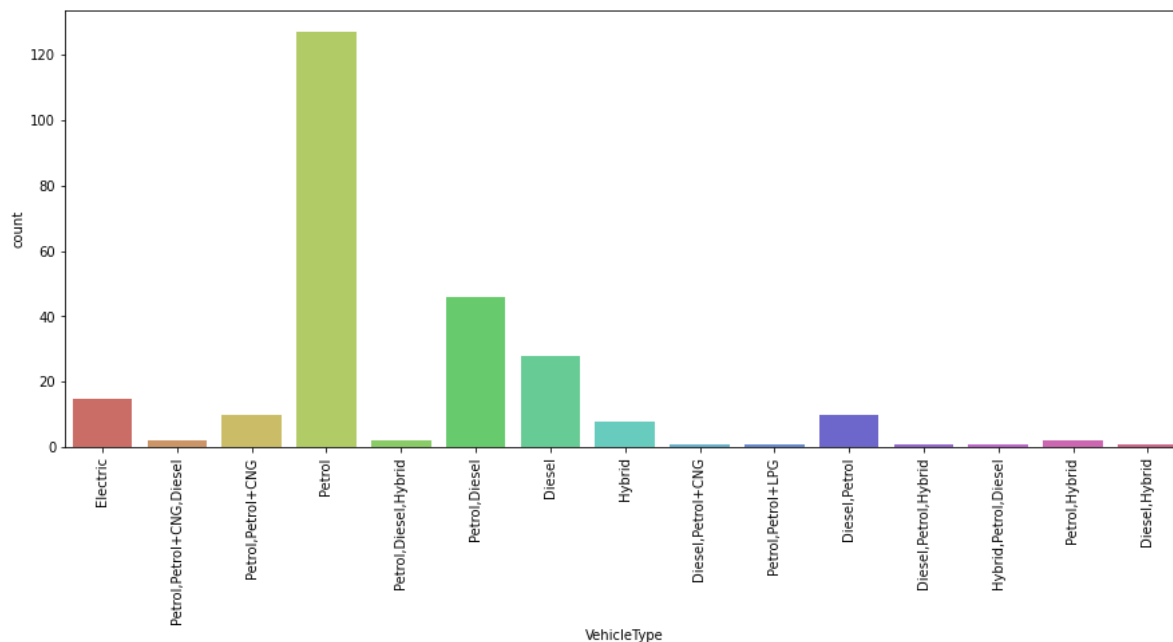
In [19]:

```python
car_data['Rating(outof10)'].unique()
```

Out[19]:

```
array(['8.5', '6.6', 'na', '7.5', '7.8', '8.2', '8.4', '7.3', '8.7',
       '6.1', '7.9', '7.7', '8.3', '7.6', '8.8', '5.5', '5', '8', '6.8',
       '6.5', '5.4', '8.1', '6.9', '7', '7.1', '7.4', '8.6', '9', '7.2',
       '6.4', '6.3'], dtype=object)
```

In [20]:

```python
car_data['Rating(outof10)'].value_counts()
```
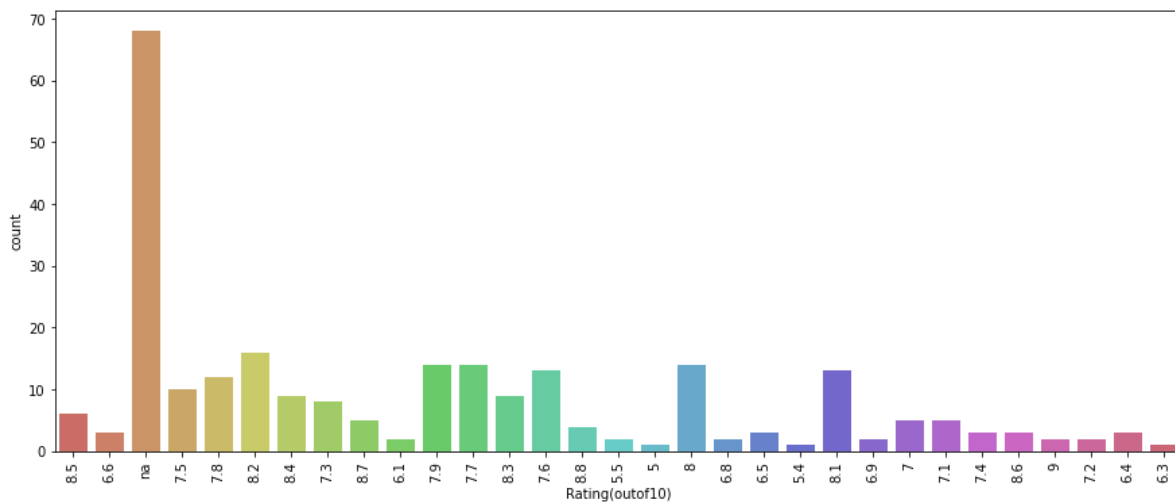
Out[20]:

```
na      68
8.2     16
8       14
7.9     14
7.7     14
7.6     13
8.1     13
7.8     12
7.5     10
8.3      9
8.4      9
7.3      8
8.5      6
7.1      5
8.7      5
7        5
8.8      4
6.6      3
6.4      3
6.5      3
8.6      3
7.4      3
7.2      2
9        2
5.5      2
6.9      2
6.8      2
6.1      2
5.4      1
5        1
6.3      1
Name: Rating(outof10), dtype: int64
```

In [21]:

```python
plt.figure(figsize=(15,6))
sns.countplot('Rating(outof10)', data = car_data, palette='hls')
plt.xticks(rotation = 90)
plt.show()
```
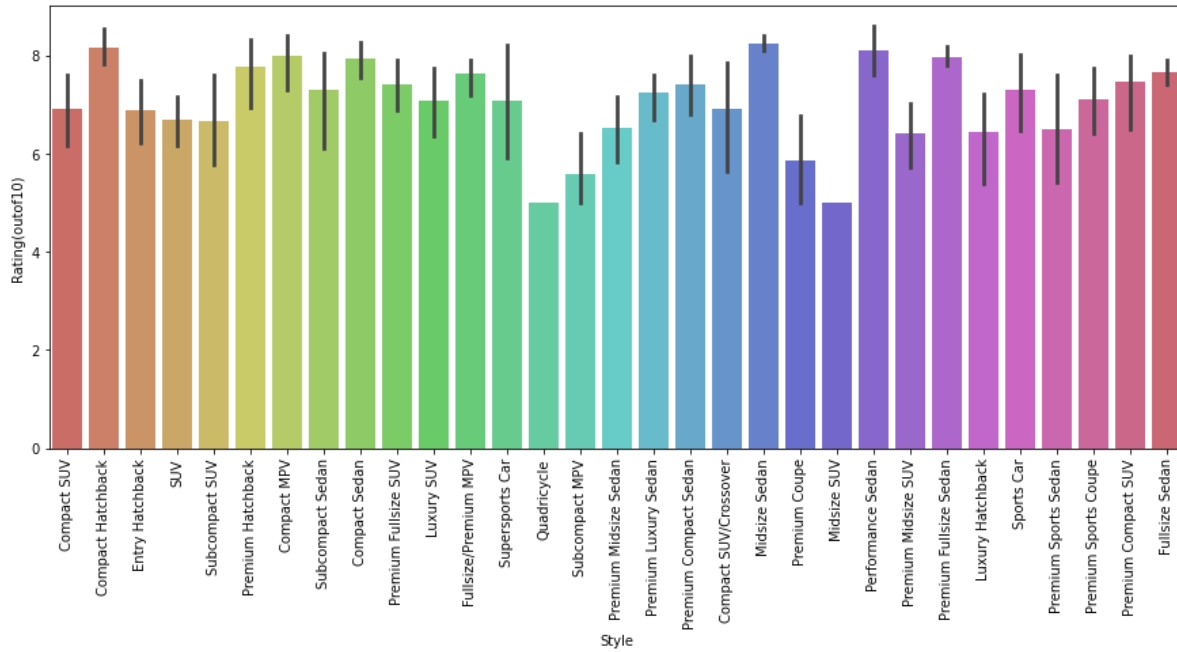

In [22]:

```python
car_data['Rating(outof10)'].replace('na', 5, inplace=True)
```

In [23]:

```python
car_data['Rating(outof10)'] = car_data['Rating(outof10)'].astype(float)
```
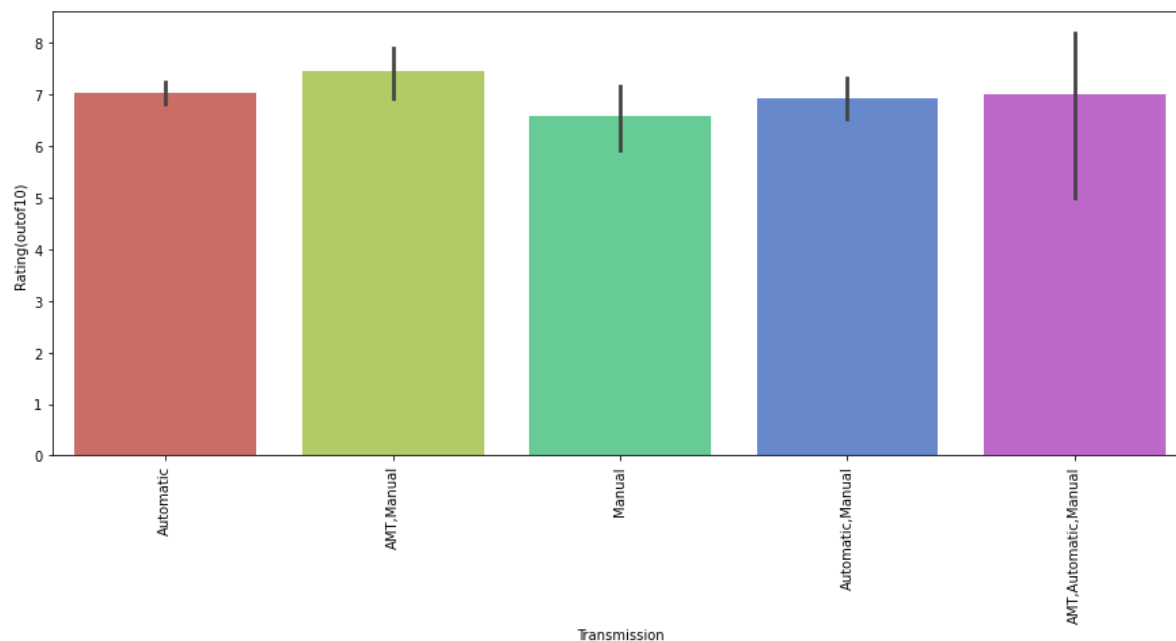
In [25]:

```python
plt.figure(figsize=(15,6))
sns.barplot(x = 'Style', y ='Rating(outof10)', data = car_data,
            palette='hls')
plt.xticks(rotation = 90)
plt.show()
```
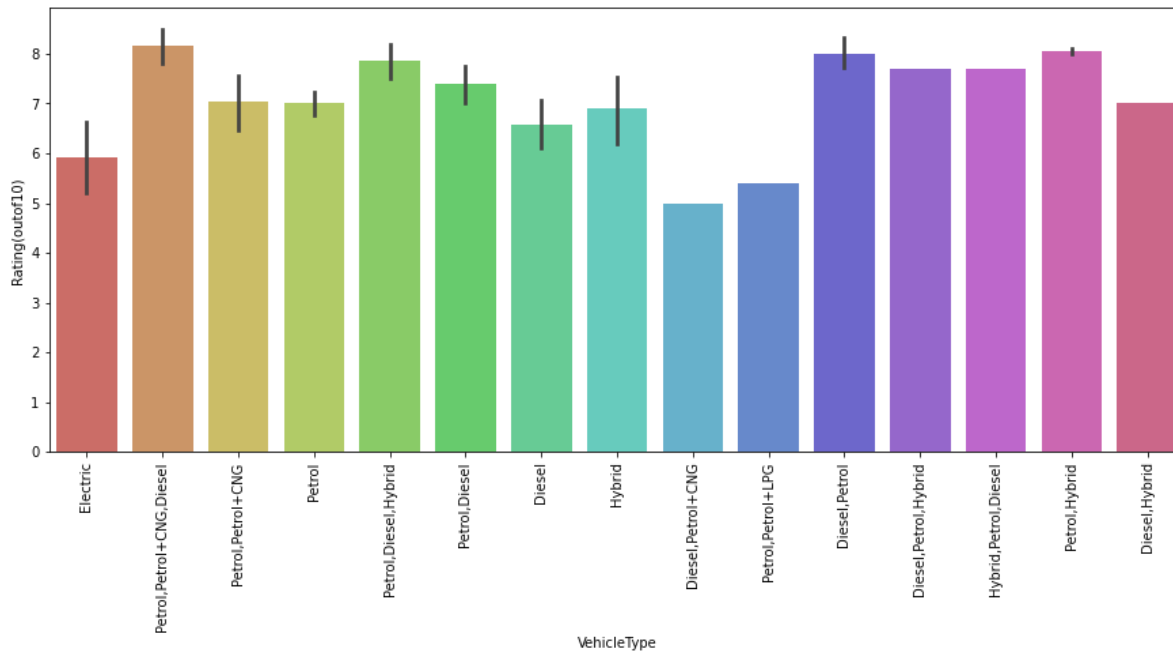
In [26]:

```python
plt.figure(figsize=(15,6))
sns.barplot(x = 'Transmission', y ='Rating(outof10)', data = car_data,
            palette='hls')
plt.xticks(rotation = 90)
plt.show()
```

In [27]:

```python
plt.figure(figsize=(15,6))
sns.barplot(x = 'VehicleType', y ='Rating(outof10)', data = car_data,
            palette='hls')
plt.xticks(rotation = 90)
plt.show()
```


In [28]:

```python
new_car_data = car_data.sort_values(by=['Rating(outof10)'], ascending=False)
```

In [30]:

```python
new_car_data = new_car_data.reset_index()
```

In [32]:

```python
new_car_data = new_car_data.drop(['index'], axis = 1)
```

In [33]:

```python
new_car_data.head()
```

Out[33]:

	Car	Style	Range	Transmission	VehicleType	PriceRange	Rating(outof10)
0	BMW X7	Premium Fullsize SUV	10.54 - 13.38 Km/l	Automatic	Diesel,Petrol	₹ 1.18 - 1.79 Cr	9.0
1	Ford Figo	Compact Hatchback	16 Km/l	Automatic	Petrol	₹ 7.75 - 8.2 L	9.0
2	Jaguar F-Pace	Luxury SUV	18 Km/l	Automatic	Diesel,Petrol	₹ 69.99 L - 1.51 Cr	8.8
3	Tata Altroz	Premium Hatchback	19.05 - 25.11 Km/l	AMT,Manual	Petrol,Diesel	₹ 6.2 - 10.15 L	8.8
4	Ferrari 488 GTB	Sports Car	8.8 Km/l	Automatic	Petrol	₹ 3.88 Cr	8.8

In [34]:

```python
new_car_data.tail()
```

Out[34]:

|	Car	Style	Range	Transmission	VehicleType	PriceRange	Rating(outof10)
250	BMW iX	Premium Fullsize SUV	425 Km/Full Charge	Automatic	Electric	₹ 1.16 Cr	5.0
251	Citroen C5 Aircross	Compact SUV	18.6 Km/l	Automatic	Diesel	₹ 31.3 - 32.8 L	5.0
252	Jaguar I-Pace	Premium Midsize Sedan	470 Km/Full Charge	Automatic	Electric	₹ 1.06 - 1.12 Cr	5.0
253	Land Rover Range Rover Velar	SUV	9.2 - 16.6 Km/l	Automatic	Petrol,Diesel	₹ 79.87 - 80.71 L	5.0
254	MINI SE	Luxury Hatchback	270 Km/Full Charge	Automatic	Electric	₹ 47.2 L	5.0

In [42]:

```python
plt.figure(figsize=(15,6))
sns.boxplot(x = 'Style', y = 'Rating(outof10)', data = new_car_data,
            palette='hls')
plt.xticks(rotation = 90)
plt.show()
```
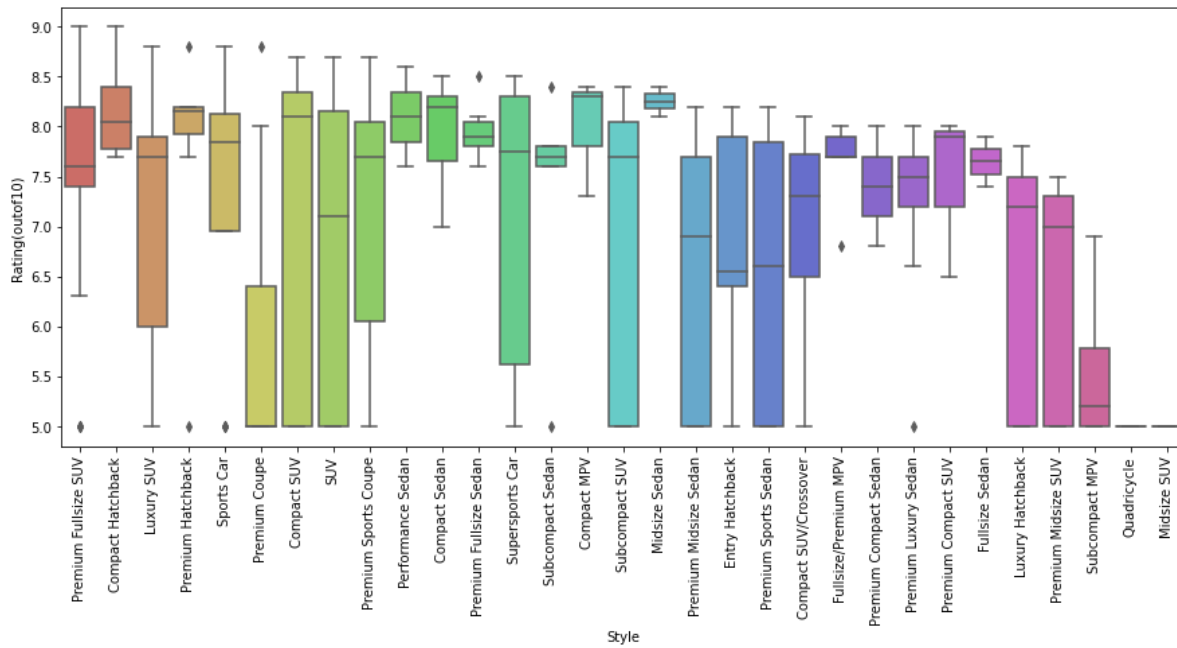
In [43]:

```python
plt.figure(figsize=(15,6))
sns.boxplot(x = 'Transmission', y = 'Rating(outof10)', data = new_car_data,
            palette='hls')
plt.xticks(rotation = 90)
plt.show()
```
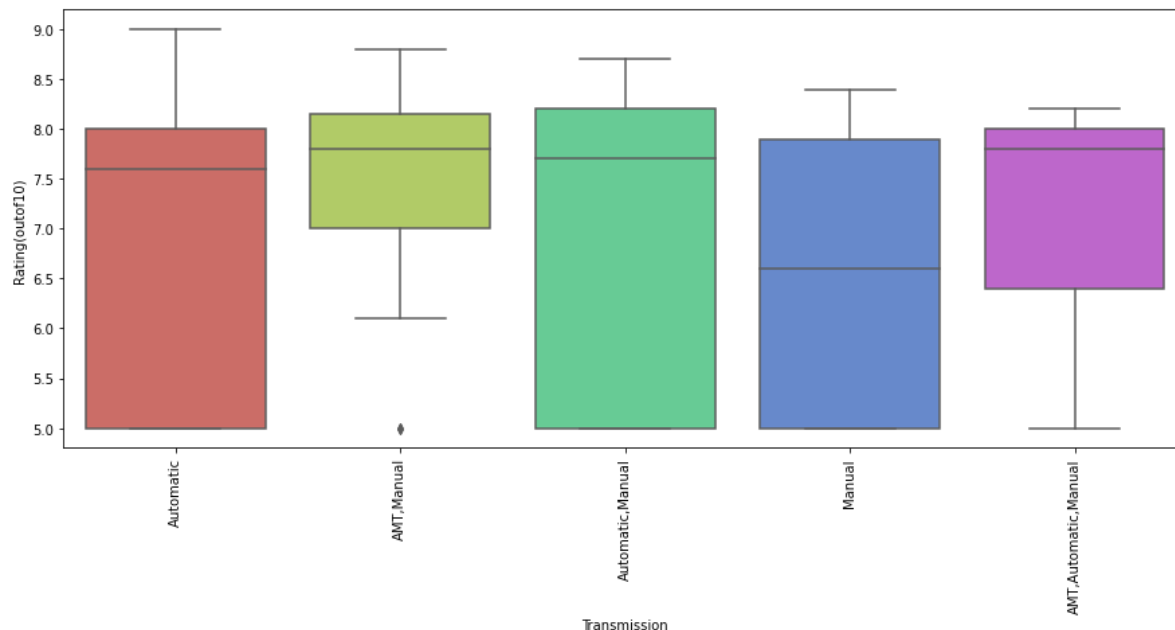

In [44]:

```python
plt.figure(figsize=(15,6))
sns.boxplot(x = 'VehicleType', y = 'Rating(outof10)', data = new_car_data,
            palette='hls')
plt.xticks(rotation = 90)
plt.show()
```