Importing Libraries

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use('dark_background')
```

Reading CSV

In [2]:
```
df = pd.read_csv('../input/zomato-bangalore-restaurants/zomato.csv')
df.head()
```

Out[2]:

	url	address	name	online_order	book_table	rate	votes	
0	https://www.zomato.com/bangalore/jalsa-banasha...	942, 21st Main Road, 2nd Stage, Banashankari, ...	Jalsa	Yes	Yes	4.1/5	775	080 422 974
1	https://www.zomato.com/bangalore/spice-elephan...	2nd Floor, 80 Feet Road, Near Big Bazaar, 6th ...	Spice Elephant	Yes	No	4.1/5	787	080
2	https://www.zomato.com/SanchurroBangalore?cont...	1112, Next to KIMS Medical College, 17th Cross...	San Churro Cafe	Yes	No	3.8/5	918	+91
3	https://www.zomato.com/bangalore/addhuri-udupi...	1st Floor, Annakuteera, 3rd Stage, Banashankar...	Addhuri Udupi Bhojana	No	No	3.7/5	88	+91
4	https://www.zomato.com/bangalore/grand-village...	10, 3rd Floor, Lakshmi Associates, Gandhi Baza...	Grand Village	No	No	3.8/5	166	+91 802 990

In [3]:
```
df.shape
```

Out[3]:
```
(51717, 17)
```

In [4]:
```
df.columns
```

Out[4]:
```
Index(['url', 'address', 'name', 'online_order', 'book_table', 'rate', 'votes',
       'phone', 'location', 'rest_type', 'dish_liked', 'cuisines',
       'approx_cost(for two people)', 'reviews_list', 'menu_item',
       'listed_in(type)', 'listed_in(city)'],
      dtype='object')
```

In [5]:

```python
df = df.drop(['url', 'address', 'phone', 'menu_item', 'dish_liked', 'reviews_list'], axis = 1)
df.head()
```

Out[5]:

	name	online_order	book_table	rate	votes	location	rest_type	cuisines	approx_cost(for two people)	listed_in(typ
0	Jalsa	Yes	Yes	4.1/5	775	Banashankari	Casual Dining	North Indian, Mughlai, Chinese	800	Buffet
1	Spice Elephant	Yes	No	4.1/5	787	Banashankari	Casual Dining	Chinese, North Indian, Thai	800	Buffet
2	San Churro Cafe	Yes	No	3.8/5	918	Banashankari	Cafe, Casual Dining	Cafe, Mexican, Italian	800	Buffet
3	Addhuri Udupi Bhojana	No	No	3.7/5	88	Banashankari	Quick Bites	South Indian, North Indian	300	Buffet
4	Grand Village	No	No	3.8/5	166	Basavanagudi	Casual Dining	North Indian, Rajasthani	600	Buffet

In [6]:

```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51717 entries, 0 to 51716
Data columns (total 11 columns):
 #   Column                       Non-Null Count  Dtype 
---  ------                       --------------  ----- 
 0   name                         51717 non-null  object
 1   online_order                 51717 non-null  object
 2   book_table                   51717 non-null  object
 3   rate                         43942 non-null  object
 4   votes                        51717 non-null  int64 
 5   location                     51696 non-null  object
 6   rest_type                    51490 non-null  object
 7   cuisines                     51672 non-null  object
 8   approx_cost(for two people)  51371 non-null  object
 9   listed_in(type)              51717 non-null  object
 10  listed_in(city)              51717 non-null  object
dtypes: int64(1), object(10)
memory usage: 4.3+ MB
```

Dropping Duplicates

In [7]:

```python
df.drop_duplicates(inplace = True)
df.shape
```

Out[7]:

```
(51609, 11)
```

Cleaning Rate Column

In [8]:

```python
df['rate'].unique()
```

Out[8]:

```
array(['4.1/5', '3.8/5', '3.7/5', '3.6/5', '4.6/5', '4.0/5', '4.2/5',
       '3.9/5', '3.1/5', '3.0/5', '3.2/5', '3.3/5', '2.8/5', '4.4/5',
       '4.3/5', 'NEW', '2.9/5', '3.5/5', nan, '2.6/5', '3.8 /5', '3.4/5',
       '4.5/5', '2.5/5', '2.7/5', '4.7/5', '2.4/5', '2.2/5', '2.3/5',
       '3.4 /5', '-', '3.6 /5', '4.8/5', '3.9 /5', '4.2 /5', '4.0 /5',
       '4.1 /5', '3.7 /5', '3.1 /5', '2.9 /5', '3.3 /5', '2.8 /5',
       '3.5 /5', '2.7 /5', '2.5 /5', '3.2 /5', '2.6 /5', '4.5 /5',
       '4.3 /5', '4.4 /5', '4.9/5', '2.1/5', '2.0/5', '1.8/5', '4.6 /5',
       '4.9 /5', '3.0 /5', '4.8 /5', '2.3 /5', '4.7 /5', '2.4 /5',
       '2.1 /5', '2.2 /5', '2.0 /5', '1.8 /5'], dtype=object)
```

Removing "NEW" , "-" and "/5" from Rate Column

In [9]:
```
def handlerate(value):
    if(value=='NEW' or value=='-'):
        return np.nan
    else:
        value = str(value).split('/')
        value = value[0]
        return float(value)

df['rate'] = df['rate'].apply(handlerate)
df['rate'].head()
```

Out[9]:
```
0    4.1
1    4.1
2    3.8
3    3.7
4    3.8
Name: rate, dtype: float64
```

Filling Null Values in Rate Column with Mean

In [10]:
```
df['rate'].fillna(df['rate'].mean(), inplace = True)
df['rate'].isnull().sum()
```

Out[10]:
```
0
```

In [11]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 51609 entries, 0 to 51716
Data columns (total 11 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   name                         51609 non-null  object 
 1   online_order                 51609 non-null  object 
 2   book_table                   51609 non-null  object 
 3   rate                         51609 non-null  float64
 4   votes                        51609 non-null  int64  
 5   location                     51588 non-null  object 
 6   rest_type                    51382 non-null  object 
 7   cuisines                     51564 non-null  object 
 8   approx_cost(for two people)  51265 non-null  object 
 9   listed_in(type)              51609 non-null  object 
 10  listed_in(city)              51609 non-null  object 
dtypes: float64(1), int64(1), object(9)
memory usage: 4.7+ MB
```

Dropping Null Values

In [12]:

```
df.dropna(inplace = True)
df.head()
```

Out[12]:

	name	online_order	book_table	rate	votes	location	rest_type	cuisines	approx_cost(for two people)	listed_in(type
0	Jalsa	Yes	Yes	4.1	775	Banashankari	Casual Dining	North Indian, Mughlai, Chinese	800	Buffet
1	Spice Elephant	Yes	No	4.1	787	Banashankari	Casual Dining	Chinese, North Indian, Thai	800	Buffet
2	San Churro Cafe	Yes	No	3.8	918	Banashankari	Cafe, Casual Dining	Cafe, Mexican, Italian	800	Buffet
3	Addhuri Udupi Bhojana	No	No	3.7	88	Banashankari	Quick Bites	South Indian, North Indian	300	Buffet
4	Grand Village	No	No	3.8	166	Basavanagudi	Casual Dining	North Indian, Rajasthani	600	Buffet

In [13]:

```
df.rename(columns = {'approx_cost(for two people)':'Cost2plates', 'listed_in(type)':'Type'}, i
nplace = True)
df.head()
```

Out[13]:

	name	online_order	book_table	rate	votes	location	rest_type	cuisines	Cost2plates	Type	listed_in
0	Jalsa	Yes	Yes	4.1	775	Banashankari	Casual Dining	North Indian, Mughlai, Chinese	800	Buffet	Banasha
1	Spice Elephant	Yes	No	4.1	787	Banashankari	Casual Dining	Chinese, North Indian, Thai	800	Buffet	Banasha
2	San Churro Cafe	Yes	No	3.8	918	Banashankari	Cafe, Casual Dining	Cafe, Mexican, Italian	800	Buffet	Banasha
3	Addhuri Udupi Bhojana	No	No	3.7	88	Banashankari	Quick Bites	South Indian, North Indian	300	Buffet	Banasha
4	Grand Village	No	No	3.8	166	Basavanagudi	Casual Dining	North Indian, Rajasthani	600	Buffet	Banasha

In [14]:

```
df['location'].unique()
```

Out[14]:

```
array(['Banashankari', 'Basavanagudi', 'Mysore Road', 'Jayanagar',
       'Kumaraswamy Layout', 'Rajarajeshwari Nagar', 'Vijay Nagar',
       'Uttarahalli', 'JP Nagar', 'South Bangalore', 'City Market',
       'Nagarbhavi', 'Bannerghatta Road', 'BTM', 'Kanakapura Road',
       'Bommanahalli', 'CV Raman Nagar', 'Electronic City', 'HSR',
       'Marathahalli', 'Wilson Garden', 'Shanti Nagar',
       'Koramangala 5th Block', 'Koramangala 8th Block', 'Richmond Road',
       'Koramangala 7th Block', 'Jalahalli', 'Koramangala 4th Block',
       'Bellandur', 'Sarjapur Road', 'Whitefield', 'East Bangalore',
       'Old Airport Road', 'Indiranagar', 'Koramangala 1st Block',
       'Frazer Town', 'RT Nagar', 'MG Road', 'Brigade Road',
       'Lavelle Road', 'Church Street', 'Ulsoor', 'Residency Road',
       'Shivajinagar', 'Infantry Road', 'St. Marks Road',
       'Cunningham Road', 'Race Course Road', 'Commercial Street',
       'Vasanth Nagar', 'HBR Layout', 'Domlur', 'Ejipura',
       'Jeevan Bhima Nagar', 'Old Madras Road', 'Malleshwaram',
       'Seshadripuram', 'Kammanahalli', 'Koramangala 6th Block',
       'Majestic', 'Langford Town', 'Central Bangalore', 'Sanjay Nagar',
       'Brookefield', 'ITPL Main Road, Whitefield',
       'Varthur Main Road, Whitefield', 'KR Puram',
       'Koramangala 2nd Block', 'Koramangala 3rd Block', 'Koramangala',
       'Hosur Road', 'Rajajinagar', 'Banaswadi', 'North Bangalore',
       'Nagawara', 'Hennur', 'Kalyan Nagar', 'New BEL Road', 'Jakkur',
       'Rammurthy Nagar', 'Thippasandra', 'Kaggadasapura', 'Hebbal',
       'Kengeri', 'Sankey Road', 'Sadashiv Nagar', 'Basaveshwara Nagar',
       'Yeshwantpur', 'West Bangalore', 'Magadi Road', 'Yelahanka',
       'Sahakara Nagar', 'Peenya'], dtype=object)
```

In [15]:
```
df['listed_in(city)'].unique()
```
Out[15]:
```
array(['Banashankari', 'Bannerghatta Road', 'Basavanagudi', 'Bellandur',
       'Brigade Road', 'Brookefield', 'BTM', 'Church Street',
       'Electronic City', 'Frazer Town', 'HSR', 'Indiranagar',
       'Jayanagar', 'JP Nagar', 'Kalyan Nagar', 'Kammanahalli',
       'Koramangala 4th Block', 'Koramangala 5th Block',
       'Koramangala 6th Block', 'Koramangala 7th Block', 'Lavelle Road',
       'Malleshwaram', 'Marathahalli', 'MG Road', 'New BEL Road',
       'Old Airport Road', 'Rajajinagar', 'Residency Road',
       'Sarjapur Road', 'Whitefield'], dtype=object)
```

Listed in(city) and location, both are there, lets keep only one.

In [16]:
```
df = df.drop(['listed_in(city)'], axis = 1)
```
In [17]:
```
df['Cost2plates'].unique()
```
Out[17]:
```
array(['800', '300', '600', '700', '550', '500', '450', '650', '400',
       '900', '200', '750', '150', '850', '100', '1,200', '350', '250',
       '950', '1,000', '1,500', '1,300', '199', '80', '1,100', '160',
       '1,600', '230', '130', '50', '190', '1,700', '1,400', '180',
       '1,350', '2,200', '2,000', '1,800', '1,900', '330', '2,500',
       '2,100', '3,000', '2,800', '3,400', '40', '1,250', '3,500',
       '4,000', '2,400', '2,600', '120', '1,450', '469', '70', '3,200',
       '60', '560', '240', '360', '6,000', '1,050', '2,300', '4,100',
       '5,000', '3,700', '1,650', '2,700', '4,500', '140'], dtype=object)
```

Removing , from Cost2Plates Column

In [18]:
```
def handlecomma(value):
    value = str(value)
    if ',' in value:
        value = value.replace(',', '')
        return float(value)
    else:
        return float(value)

df['Cost2plates'] = df['Cost2plates'].apply(handlecomma)
df['Cost2plates'].unique()
```
Out[18]:
```
array([ 800.,  300.,  600.,  700.,  550.,  500.,  450.,  650.,  400.,
        900.,  200.,  750.,  150.,  850.,  100., 1200.,  350.,  250.,
        950., 1000., 1500., 1300.,  199.,   80., 1100.,  160., 1600.,
        230.,  130.,   50.,  190., 1700., 1400.,  180., 1350., 2200.,
       2000., 1800., 1900.,  330., 2500., 2100., 3000., 2800., 3400.,
         40., 1250., 3500., 4000., 2400., 2600.,  120., 1450.,  469.,
         70., 3200.,   60.,  560.,  240.,  360., 6000., 1050., 2300.,
       4100., 5000., 3700., 1650., 2700., 4500.,  140.])
```
In [19]:
```
df.head()
```
Out[19]:

	name	online_order	book_table	rate	votes	location	rest_type	cuisines	Cost2plates	Type
0	Jalsa	Yes	Yes	4.1	775	Banashankari	Casual Dining	North Indian, Mughlai, Chinese	800.0	Buffet
1	Spice Elephant	Yes	No	4.1	787	Banashankari	Casual Dining	Chinese, North Indian, Thai	800.0	Buffet
2	San Churro Cafe	Yes	No	3.8	918	Banashankari	Cafe, Casual Dining	Cafe, Mexican, Italian	800.0	Buffet
3	Addhuri Udupi Bhojana	No	No	3.7	88	Banashankari	Quick Bites	South Indian, North Indian	300.0	Buffet
4	Grand Village	No	No	3.8	166	Basavanagudi	Casual Dining	North Indian, Rajasthani	600.0	Buffet

Cleaning Rest Type Column

In [20]:
```
rest_types = df['rest_type'].value_counts(ascending  = False)
rest_types
```

Out[20]:
```
Quick Bites                  19010
Casual Dining                10253
Cafe                          3682
Delivery                      2574
Dessert Parlor                2242
                             ...
Dessert Parlor, Kiosk            2
Pop Up                           2
Bakery, Food Court               2
Sweet Shop, Dessert Parlor       1
Quick Bites, Kiosk               1
Name: rest_type, Length: 93, dtype: int64
```

In [21]:
```
rest_types_lessthan1000 = rest_types[rest_types<1000]
rest_types_lessthan1000
```

Out[21]:
```
Beverage Shop                863
Bar                          686
Food Court                   616
Sweet Shop                   468
Bar, Casual Dining           411
                            ...
Dessert Parlor, Kiosk          2
Pop Up                         2
Bakery, Food Court             2
Sweet Shop, Dessert Parlor     1
Quick Bites, Kiosk             1
Name: rest_type, Length: 85, dtype: int64
```

Making Rest Types less than 1000 in frequency as others

In [22]:
```
def handle_rest_type(value):
    if(value in rest_types_lessthan1000):
        return 'others'
    else:
        return value

df['rest_type'] = df['rest_type'].apply(handle_rest_type)
df['rest_type'].value_counts()
```

Out[22]:
```
Quick Bites           19010
Casual Dining         10253
others                 9003
Cafe                   3682
Delivery               2574
Dessert Parlor         2242
Takeaway, Delivery     2008
Bakery                 1140
Casual Dining, Bar     1130
Name: rest_type, dtype: int64
```

Cleaning Location Column

In [23]:
```
location = df['location'].value_counts(ascending  = False)

location_lessthan300 = location[location<300]



def handle_location(value):
    if(value in location_lessthan300):
        return 'others'
    else:
        return value

df['location'] = df['location'].apply(handle_location)
df['location'].value_counts()
```

Out[23]:
```
BTM                      5056
others                   4954
HSR                      2494
Koramangala 5th Block    2479
JP Nagar                 2218
Whitefield               2105
Indiranagar              2026
Jayanagar                1916
Marathahalli             1805
Bannerghatta Road        1609
Bellandur                1268
Electronic City          1246
Koramangala 1st Block    1236
Brigade Road             1210
Koramangala 7th Block    1174
Koramangala 6th Block    1127
Sarjapur Road            1047
Koramangala 4th Block    1017
Ulsoor                   1011
Banashankari              902
MG Road                   893
Kalyan Nagar              841
Richmond Road             803
Malleshwaram              721
Frazer Town               714
Basavanagudi              684
Residency Road            671
Brookefield               656
New BEL Road              644
Banaswadi                 640
Kammanahalli              639
Rajajinagar               591
Church Street             566
Lavelle Road              518
Shanti Nagar              508
Shivajinagar              498
Cunningham Road           490
Domlur                    482
Old Airport Road          437
Ejipura                   433
Commercial Street         370
St. Marks Road            343
Name: location, dtype: int64
```

Cleaning Cuisines Column

In [24]:

```
cuisines = df['cuisines'].value_counts(ascending  = False)



cuisines_lessthan100 = cuisines[cuisines<100]




def handle_cuisines(value):
    if(value in cuisines_lessthan100):
        return 'others'
    else:
        return value

df['cuisines'] = df['cuisines'].apply(handle_cuisines)
df['cuisines'].value_counts()
```

Out[24]:

```
others                                26159
North Indian                           2852
North Indian, Chinese                  2351
South Indian                           1820
Biryani                                 903
                                      ...
South Indian, Chinese, North Indian     105
South Indian, Fast Food                 104
North Indian, Mughlai, Chinese          104
Italian, Pizza                          102
North Indian, Chinese, Seafood          102
Name: cuisines, Length: 70, dtype: int64
```

In [25]:

```
df.head()
```

Out[25]:

	name	online_order	book_table	rate	votes	location	rest_type	cuisines	Cost2plates	Type
0	Jalsa	Yes	Yes	4.1	775	Banashankari	Casual Dining	North Indian, Mughlai, Chinese	800.0	Buffet
1	Spice Elephant	Yes	No	4.1	787	Banashankari	Casual Dining	others	800.0	Buffet
2	San Churro Cafe	Yes	No	3.8	918	Banashankari	others	others	800.0	Buffet
3	Addhuri Udupi Bhojana	No	No	3.7	88	Banashankari	Quick Bites	South Indian, North Indian	300.0	Buffet
4	Grand Village	No	No	3.8	166	Basavanagudi	Casual Dining	others	600.0	Buffet

Data is Clean, Lets jump to Visualization

Count Plot of Various Locations

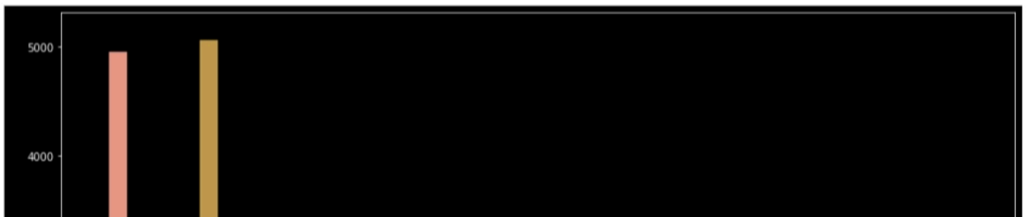
In [26]:

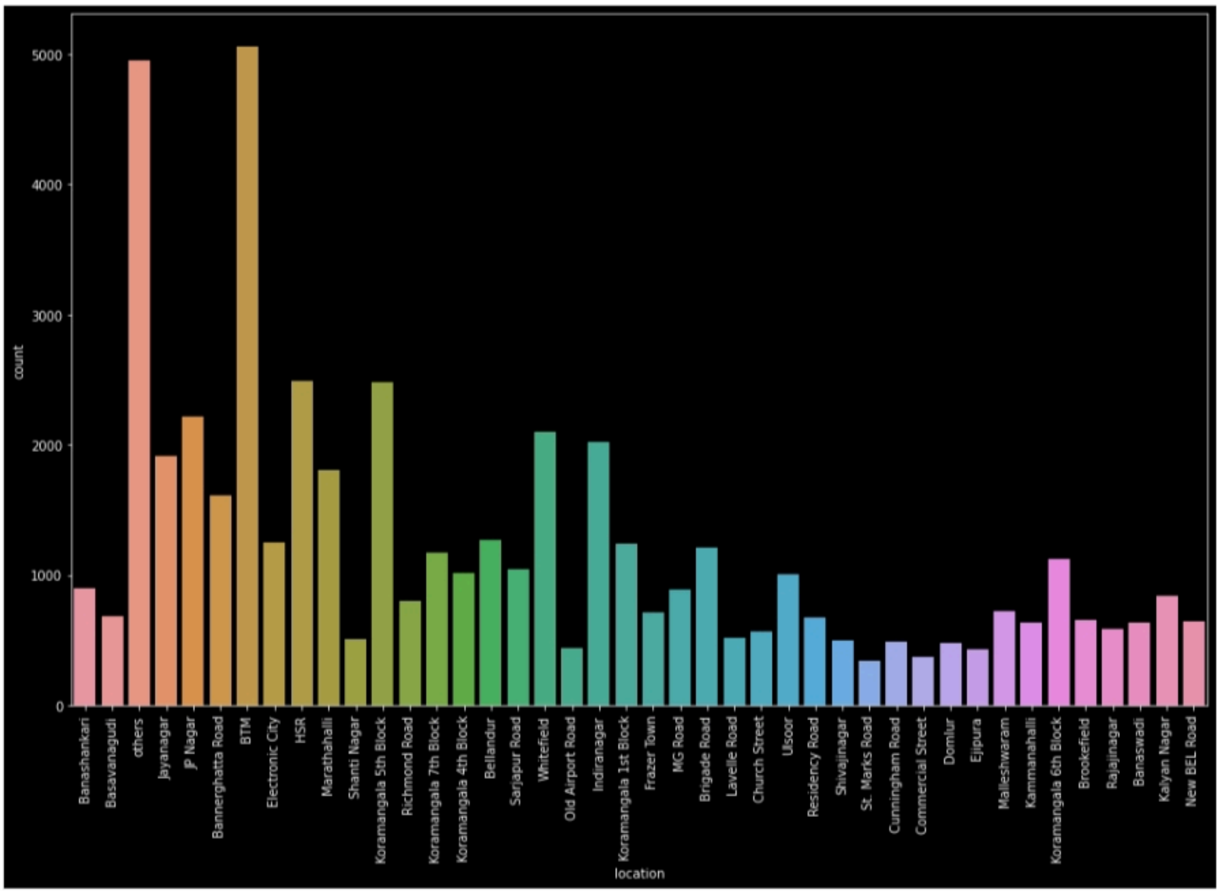
```
plt.figure(figsize = (16,10))
ax = sns.countplot(df['location'])
plt.xticks(rotation=90)
```

```
/opt/conda/lib/python3.7/site-packages/seaborn/_decorators.py:43: FutureWarning: Pass the f
ollowing variable as a keyword arg: x. From version 0.12, the only valid positional argumen
t will be `data`, and passing other arguments without an explicit keyword will result in an
error or misinterpretation.
  FutureWarning
```

Out[26]:

```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
        34, 35, 36, 37, 38, 39, 40, 41]),
 [Text(0, 0, 'Banashankari'),
  Text(1, 0, 'Basavanagudi'),
  Text(2, 0, 'others'),
  Text(3, 0, 'Jayanagar'),
  Text(4, 0, 'JP Nagar'),
  Text(5, 0, 'Bannerghatta Road'),
  Text(6, 0, 'BTM'),
  Text(7, 0, 'Electronic City'),
  Text(8, 0, 'HSR'),
  Text(9, 0, 'Marathahalli'),
  Text(10, 0, 'Shanti Nagar'),
  Text(11, 0, 'Koramangala 5th Block'),
  Text(12, 0, 'Richmond Road'),
  Text(13, 0, 'Koramangala 7th Block'),
  Text(14, 0, 'Koramangala 4th Block'),
  Text(15, 0, 'Bellandur'),
  Text(16, 0, 'Sarjapur Road'),
  Text(17, 0, 'Whitefield'),
  Text(18, 0, 'Old Airport Road'),
  Text(19, 0, 'Indiranagar'),
  Text(20, 0, 'Koramangala 1st Block'),
  Text(21, 0, 'Frazer Town'),
  Text(22, 0, 'MG Road'),
  Text(23, 0, 'Brigade Road'),
  Text(24, 0, 'Lavelle Road'),
  Text(25, 0, 'Church Street'),
  Text(26, 0, 'Ulsoor'),
  Text(27, 0, 'Residency Road'),
  Text(28, 0, 'Shivajinagar'),
  Text(29, 0, 'St. Marks Road'),
  Text(30, 0, 'Cunningham Road'),
  Text(31, 0, 'Commercial Street'),
  Text(32, 0, 'Domlur'),
  Text(33, 0, 'Ejipura'),
  Text(34, 0, 'Malleshwaram'),
  Text(35, 0, 'Kammanahalli'),
  Text(36, 0, 'Koramangala 6th Block'),
  Text(37, 0, 'Brookefield'),
  Text(38, 0, 'Rajajinagar'),
  Text(39, 0, 'Banaswadi'),
  Text(40, 0, 'Kalyan Nagar'),
  Text(41, 0, 'New BEL Road')])
```





Visualizing Online Order

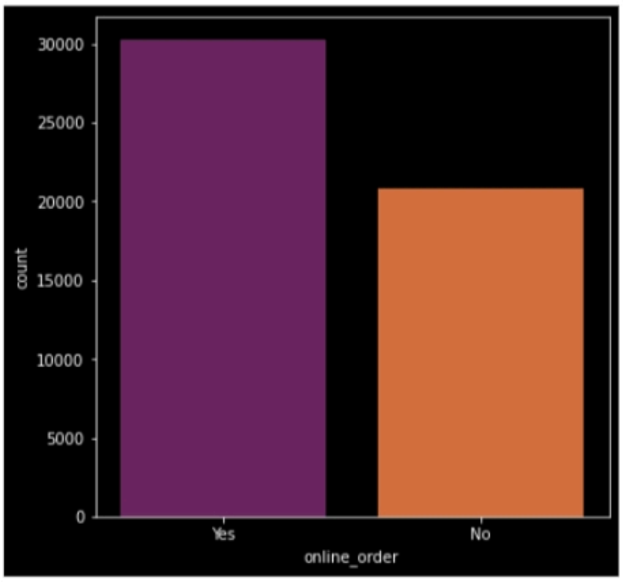
In [27]:

```
plt.figure(figsize = (6,6))
sns.countplot(df['online_order'], palette = 'inferno')
```

```
/opt/conda/lib/python3.7/site-packages/seaborn/_decorators.py:43: FutureWarning: Pass the f
ollowing variable as a keyword arg: x. From version 0.12, the only valid positional argumen
t will be `data`, and passing other arguments without an explicit keyword will result in an
error or misinterpretation.
  FutureWarning
```

Out[27]:

```
<AxesSubplot:xlabel='online_order', ylabel='count'>
```

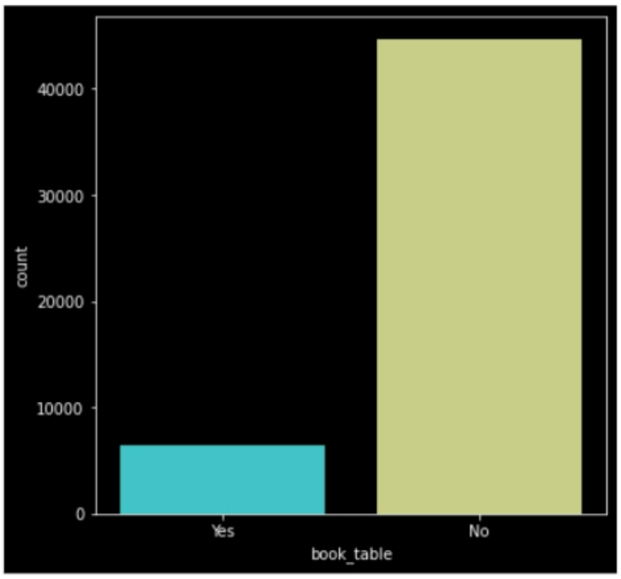


Visualizing Book Table

In [28]:
```python
plt.figure(figsize = (6,6))
sns.countplot(df['book_table'], palette = 'rainbow')
```

```
/opt/conda/lib/python3.7/site-packages/seaborn/_decorators.py:43: FutureWarning: Pass the f
ollowing variable as a keyword arg: x. From version 0.12, the only valid positional argumen
t will be `data`, and passing other arguments without an explicit keyword will result in an
error or misinterpretation.
  FutureWarning
```

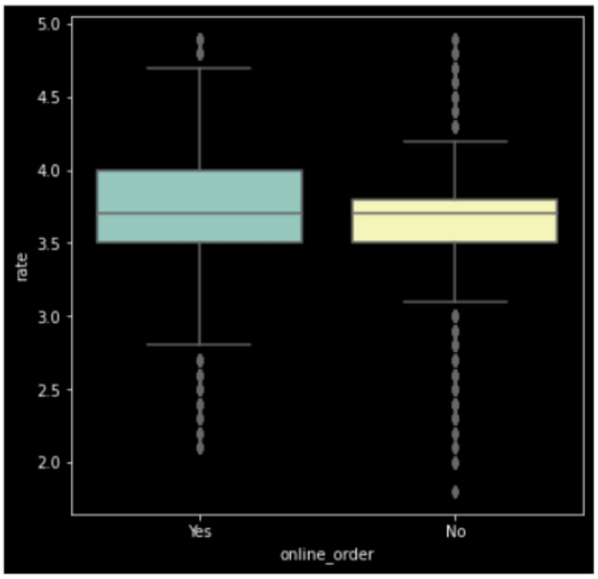
Out[28]:
```
<AxesSubplot:xlabel='book_table', ylabel='count'>
```



Visualizing Online Order vs Rate

In [29]:
```python
plt.figure(figsize = (6,6))
sns.boxplot(x = 'online_order', y = 'rate', data = df)
```

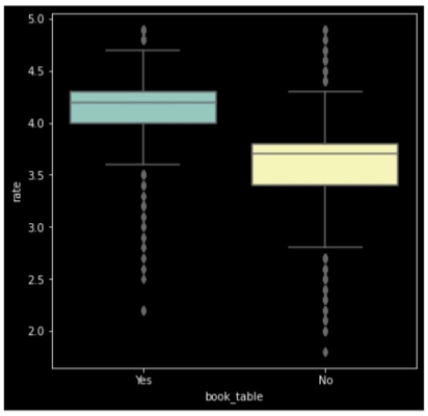
Out[29]:
```
<AxesSubplot:xlabel='online_order', ylabel='rate'>
```



Visualizing Book Table vs Rate

In [30]:
```
plt.figure(figsize = (6,6))
sns.boxplot(x = 'book_table', y = 'rate', data = df)
```

Out[30]:
```
<AxesSubplot:xlabel='book_table', ylabel='rate'>
```



Visualizing Online Order Facility, Location Wise

In [31]:
```
df1 = df.groupby(['location','online_order'])['name'].count()
df1.to_csv('location_online.csv')
df1 = pd.read_csv('location_online.csv')
df1 = pd.pivot_table(df1, values=None, index=['location'], columns=['online_order'], fill_valu
e=0, aggfunc=np.sum)
df1
```

Out[31]:

	name	
online_order	No	Yes
location		
BTM	1763	3293
Banashankari	397	505
Banaswadi	302	338
Bannerghatta Road	685	924
Basavanagudi	243	441
Bellandur	517	751
Brigade Road	552	658
Brookefield	239	417
Church Street	226	340
Commercial Street	228	142
Cunningham Road	168	322
Domlur	247	235
Ejipura	214	219
Electronic City	676	570
Frazer Town	287	427
HSR	584	1910
Indiranagar	697	1329
JP Nagar	911	1307
Jayanagar	552	1364
Kalyan Nagar	350	491
Kammanahalli	264	375
Koramangala 1st Block	384	852
Koramangala 4th Block	459	558
Koramangala 5th Block	866	1613
Koramangala 6th Block	445	682
Koramangala 7th Block	389	785
Lavelle Road	315	203
MG Road	520	373
Malleshwaram	309	412
Marathahalli	701	1104
New BEL Road	255	389
Old Airport Road	221	216
Rajajinagar	286	305
Residency Road	424	247
Richmond Road	557	246
Sarjapur Road	323	724
Shanti Nagar	289	219
Shivajinagar	354	144
St. Marks Road	176	167
Ulsoor	389	622
Whitefield	986	1119
others	2064	2890