In [77]:
```
## imported requried libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use('dark_background')
```

In [78]:
```
# imported data file for further process
df = pd.read_csv("zomato_02.csv")
df.head()
```

Out[78]:

	url	address	name	online_order	book_table	rate	votes	phone	location	rest_type
0	https://www.zomato.com/bangalore/jalsa-banasha...	942, 21st Main Road, 2nd Stage, Banashankari, ...	Jalsa	Yes	Yes	4.1/5	775	080 42297555\r\n+91 9743772233	Banashankari	Casual Dining
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 41714161	Banashankari	Casual Dining
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 9663487993	Banashankari	Cafe, Casual Dining
3	https://www.zomato.com/bangalore/addhuri-udupi...	1st Floor, Annakuteera, 3rd Stage, Banashankar...	Addhuri Udupi Bhojana	No	No	3.7/5	88	+91 9620009302	Banashankari	Quick Bites
4	https://www.zomato.com/bangalore/grand-village...	10, 3rd Floor, Lakshmi Associates, Gandhi Baza...	Grand Village	No	No	3.8/5	166	+91 8026612447\r\n+91 9901210005	Basavanagudi	Casual Dining

In [6]:
```
# checking the shape of the data
df.shape
```

Out[6]:
```
(51717, 17)
```

In [7]:
```
# checking the columns name
df.columns
```

Out[7]:
```
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')
```

dropping the columns are not required for analysis

In [8]:
```
df = df.drop(['url', 'address', 'phone', 'menu_item', 'dish_liked', 'reviews_list'], axis = 1)
df.head()
```

Out[8]:

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

In [9]:
```
## printed the info.

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 the duplicate (columns).

In [10]:
```
df.drop_duplicates(inplace = True)
df.shape
```

Out[10]:
```
(51609, 11)
```

In [11]:
```
##checking the unique values in the data

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

Out[11]:
```
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 [13]:
```
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[13]:
```
0    4.1
1    4.1
2    3.8
3    3.7
4    3.8
Name: rate, dtype: float64
```

In [14]:
```
##Filling Null Values in Rate Column with Mean.

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

Out[14]:
```
0
```

In [15]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 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
```

In [16]:
```
##Dropping Null Values are present in the data

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

Out[16]:

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

In [17]:
```
## renamed the columns names accordingly.

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

Out[17]:

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

In [18]:
```
## printed the unique values are presented in the data different locations

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

Out[18]:
```
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 [19]:
```
##printed the unique values from the cloumn listed in city.

df['listed_in(city)'].unique()
```

Out[19]:
```
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)
```

In [20]:
```
##Listed in(city) and location, both are there, lets keep only one.
```

```
df = df.drop(['listed_in(city)'], axis = 1)
df['Cost2plates'].unique()
```

Out[20]:
```
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)
```

In [21]:
```
##Removing , from Cost2Plates Column


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[21]:
```
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 [22]:
```
df.head()
```

Out[22]:

	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

In [23]:
```
##Cleaning Rest Type Column

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

Out[23]:
```
rest_type
Quick Bites                   19010
Casual Dining                 10253
Cafe                           3682
Delivery                       2574
Dessert Parlor                 2242
                              ...
Dessert Parlor, Kiosk             2
Food Court, Beverage Shop         2
Dessert Parlor, Food Court        2
Quick Bites, Kiosk                1
Sweet Shop, Dessert Parlor        1
Name: count, Length: 93, dtype: int64
```

In [24]:
```
rest_types_lessthan1000 = rest_types[rest_types<1000]
rest_types_lessthan1000
```

Out[24]:
```
rest_type
Beverage Shop                 863
Bar                           686
Food Court                    616
Sweet Shop                    468
Bar, Casual Dining            411
                             ...
Dessert Parlor, Kiosk           2
Food Court, Beverage Shop       2
Dessert Parlor, Food Court      2
Quick Bites, Kiosk              1
Sweet Shop, Dessert Parlor      1
Name: count, Length: 85, dtype: int64
```

In [25]:
```
##Making Rest Types less than 1000 in frequency as others¶


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[25]:
```
rest_type
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: count, dtype: int64
```

In [31]:
```
##Cleaning Location Column.


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[31]:

location	
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: count, dtype: int64

In [79]:
```
##Cleaning Cuisines Column.


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[79]:
```
cuisines
others                                26460
North Indian                           2913
North Indian, Chinese                  2385
South Indian                           1828
Biryani                                 918
                                      ...
South Indian, Chinese, North Indian     105
Italian, Pizza                          105
North Indian, Mughlai, Chinese          104
South Indian, Fast Food                 104
North Indian, Chinese, Seafood          102
Name: count, Length: 70, dtype: int64
```

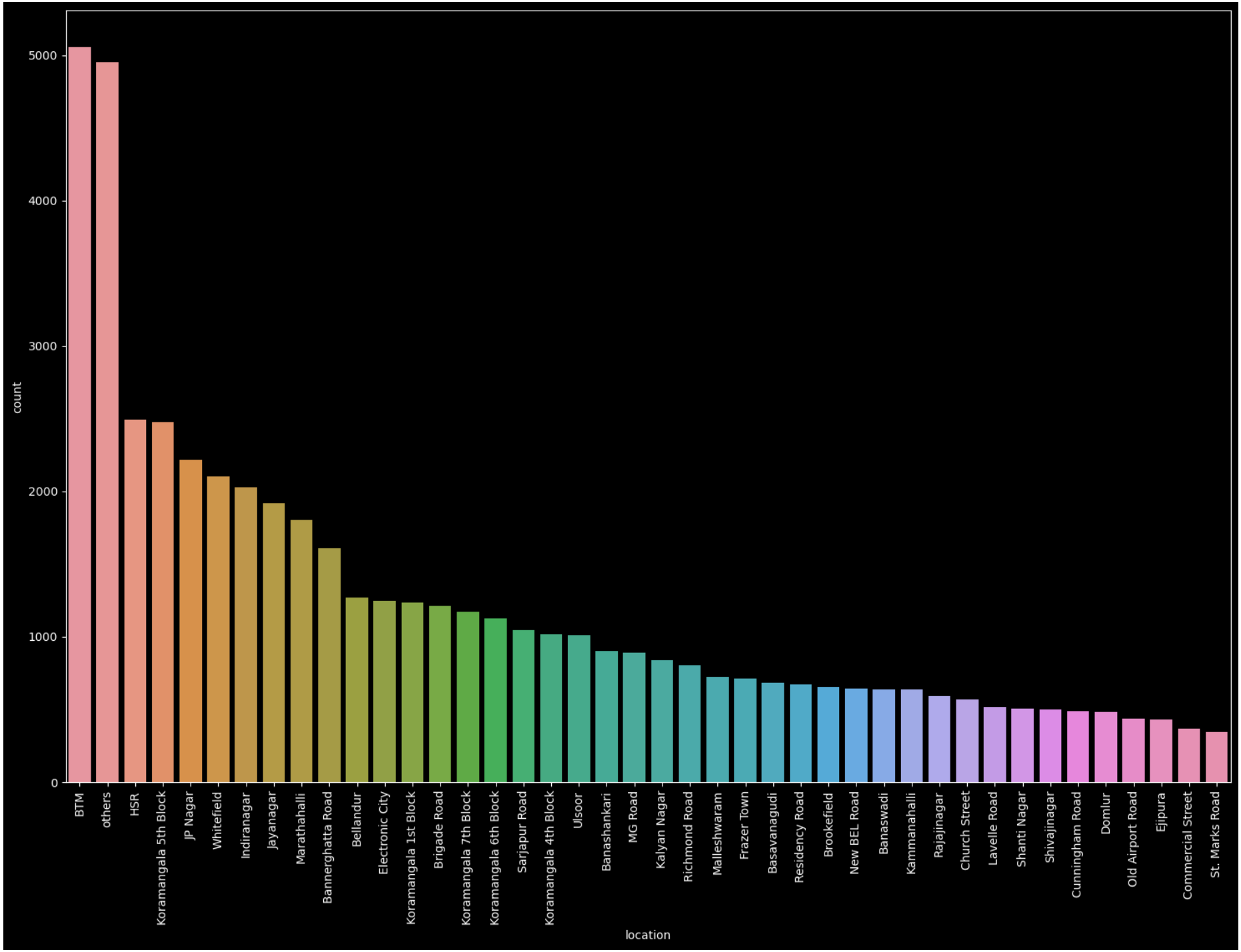
In [28]:
```
df.head()
```

Out[28]:

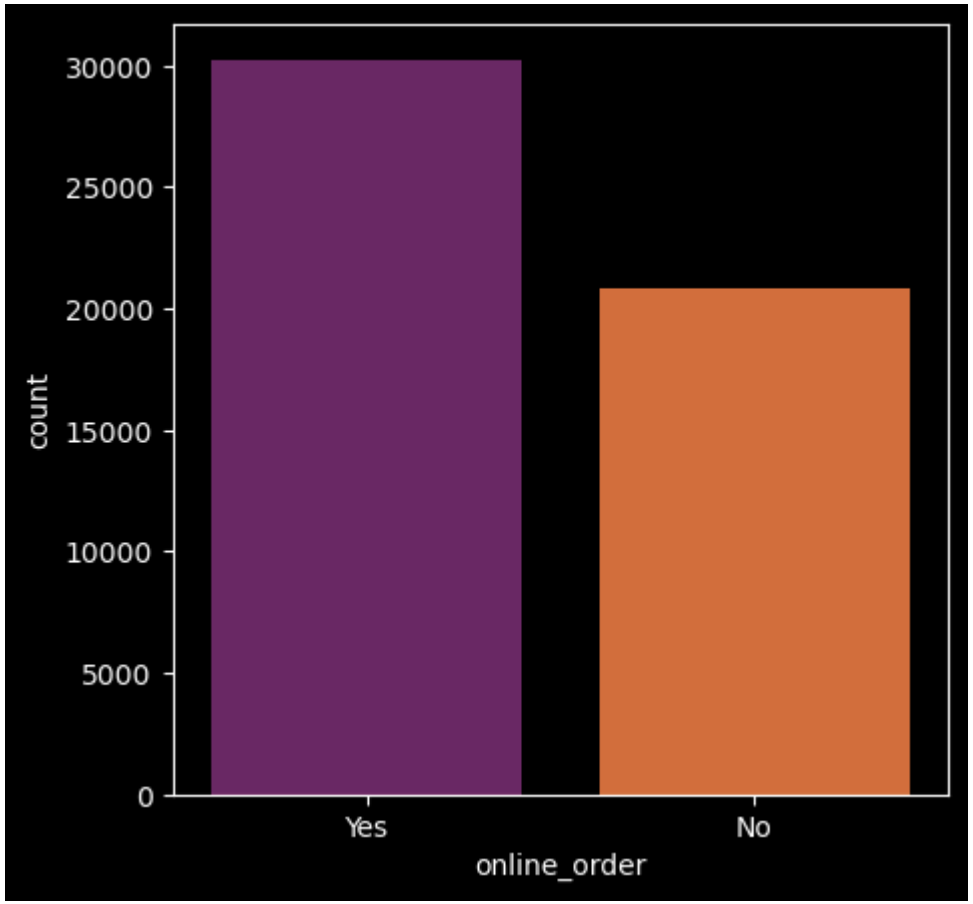
	name	online_order	book_table	rate	votes	location	rest_type	cuisines	Cost2plates	Type
0	Jalsa	Yes	Yes	4.1	775	Banashankari	Casual Dining	North Indian, Mughlai, Chinese	800.0	Buffet
1	Spice Elephant	Yes	No	4.1	787	Banashankari	Casual Dining	others	800.0	Buffet
2	San Churro Cafe	Yes	No	3.8	918	Banashankari	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 as requried for analysis Visualization of Count Plot of Various Locations

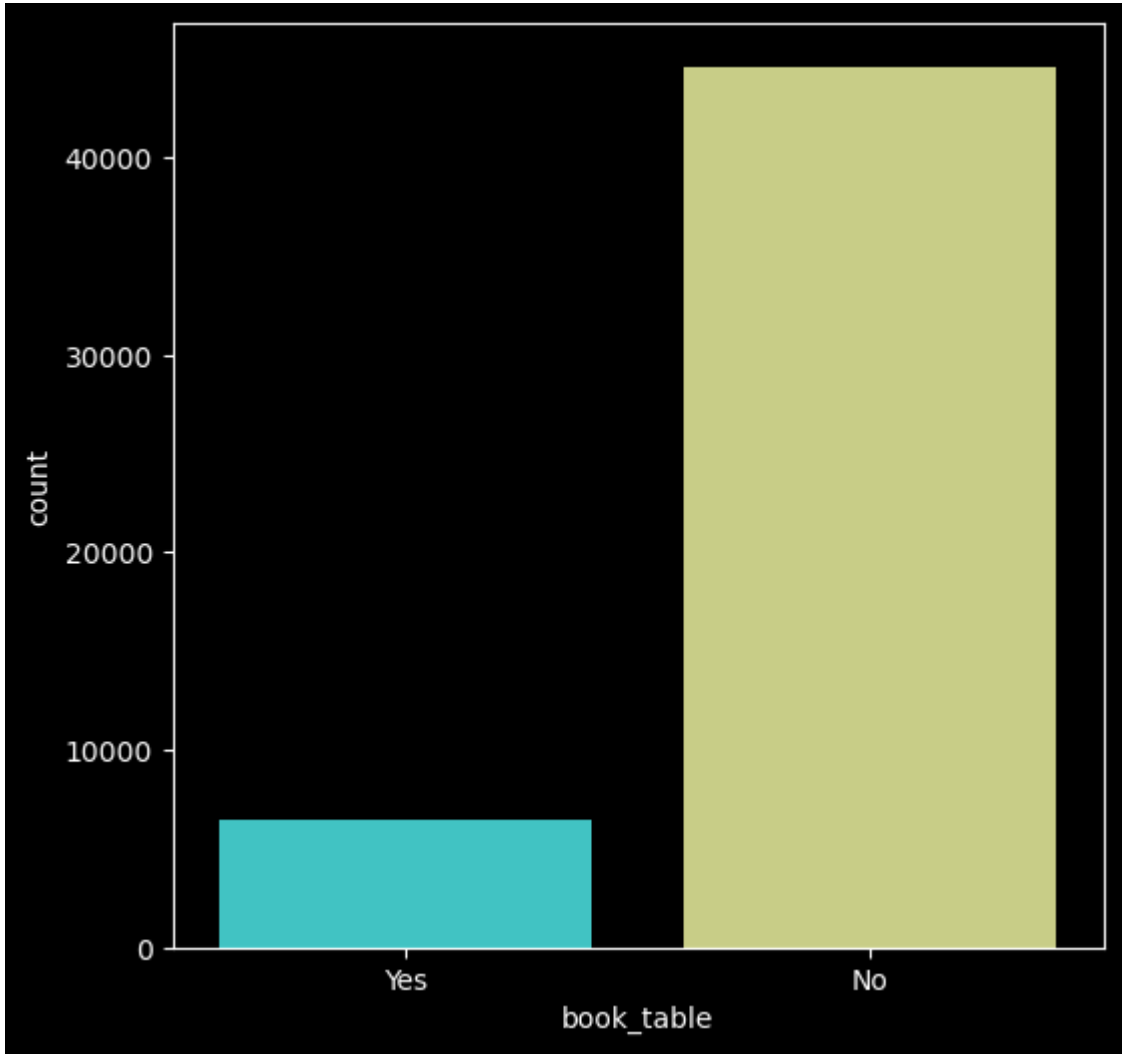
In [41]:
```
## plotted the graph of location column

plt.figure(figsize=(18, 12))
ax = sns.countplot(data=df, x='location', order=df['location'].value_counts().index)
plt.xticks(rotation=90)
plt.show()
```


count
location
0
1000
2000
3000
4000
5000
BTM
others
HSR
Koramangala 5th Block
JP Nagar
Whitefield
Indiranagar
Jayanagar
Marathahalli
Bannerghatta Road
Bellandur
Electronic City
Koramangala 1st Block
Brigade Road
Koramangala 7th Block
Koramangala 6th Block
Sarjapur Road
Koramangala 4th Block
Ulsoor
Banashankari
MG Road
Kalyan Nagar
Richmond Road
Malleshwaram
Frazer Town
Basavanagudi
Residency Road
Brookefield
New BEL Road
Banaswadi
Kammanahalli
Rajajinagar
Church Street
Lavelle Road
Shanti Nagar
Shivajinagar
Cunningham Road
Domlur
Old Airport Road
Ejipura
Commercial Street
St. Marks Road

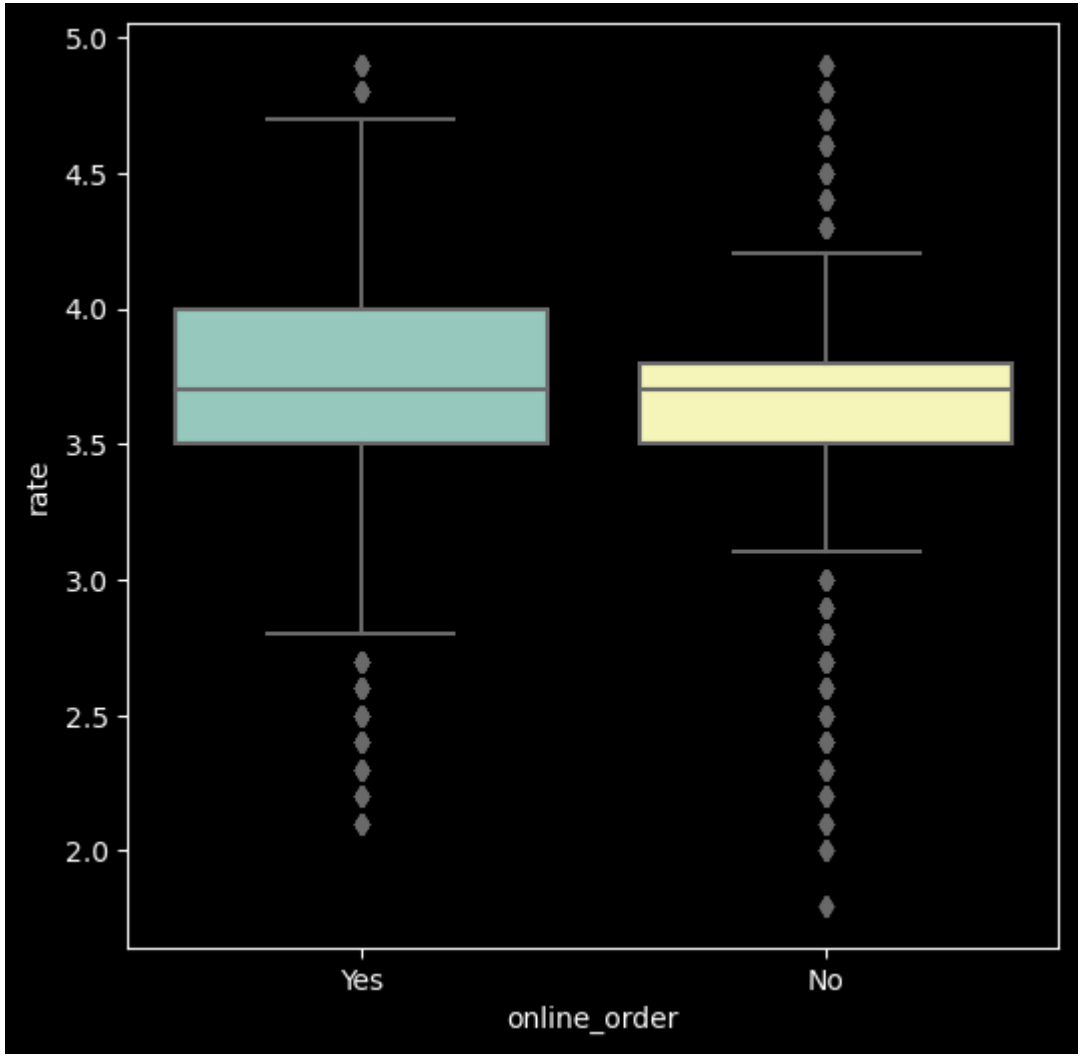
In [44]:
```
##Visualizing Online Order


plt.figure(figsize=(5, 5))
sns.countplot(data=df, x='online_order', palette='inferno', order=['Yes', 'No'])
plt.show()
```



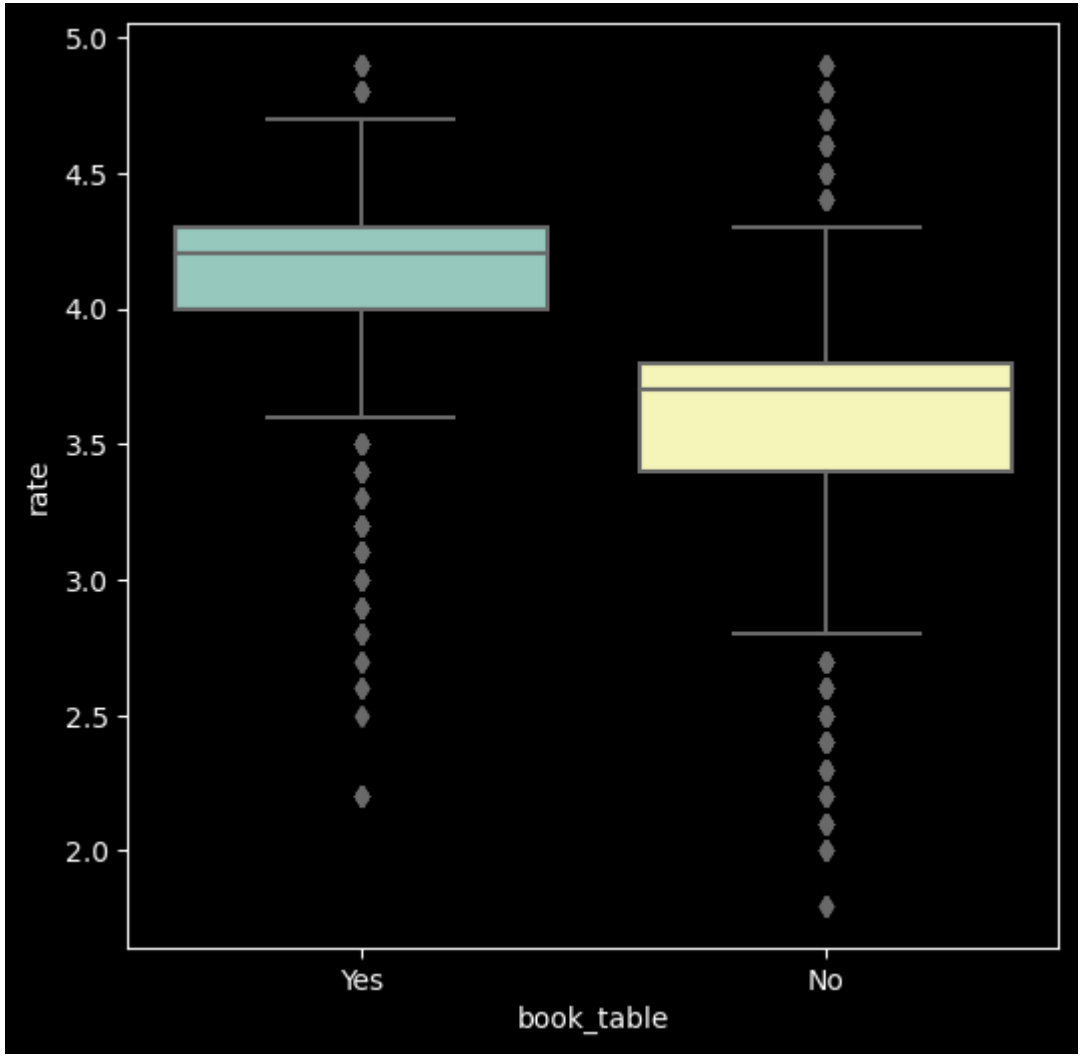
In [46]:
```
##Visualizing Book Table

plt.figure(figsize = (6,6))
sns.countplot(data=df, x='book_table', palette='rainbow', order=['Yes', 'No'])
plt.show()
```



In [48]:
```
#Visualizing Online Order vs Rate

plt.figure(figsize = (6,6))
sns.boxplot(x = 'online_order', y = 'rate', data = df)
plt.show()
```



In [49]:
```
#Visualizing Book Table vs Rate


plt.figure(figsize = (6,6))
sns.boxplot(x = 'book_table', y = 'rate', data = df)
plt.show()
```



In [51]:
```
#Visualizing Online Order Facility, Location Wise

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_value=0, aggfunc=np.sum)
```

In [52]:
```
df1
```

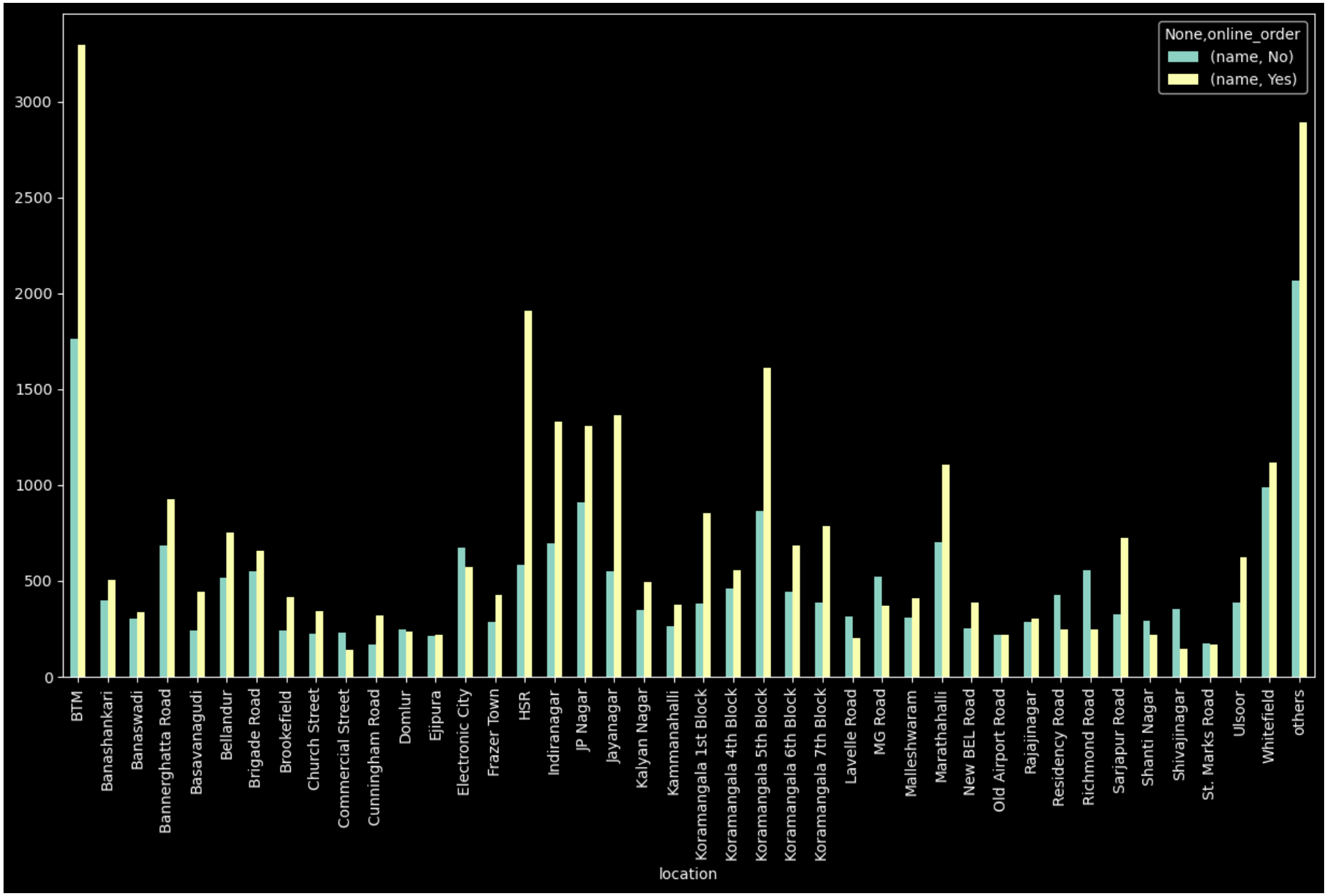
Out[52]:

online_order / location	name: No	name: Yes
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

	name	
online_order	No	Yes
location		
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

In [53]:
```
df1.plot(kind = 'bar', figsize = (15,8))
```

Out[53]: <Axes: xlabel='location'>



In [54]:
```
#Visualizing Book Table Facility, Location Wise
```

```
df2 = df.groupby(['location','book_table'])['name'].count()
df2.to_csv('location_booktable.csv')
df2 = pd.read_csv('location_booktable.csv')
df2 = pd.pivot_table(df2, values=None, index=['location'], columns=['book_table'], fill_value=0, aggfunc=np.sum)
df2
```

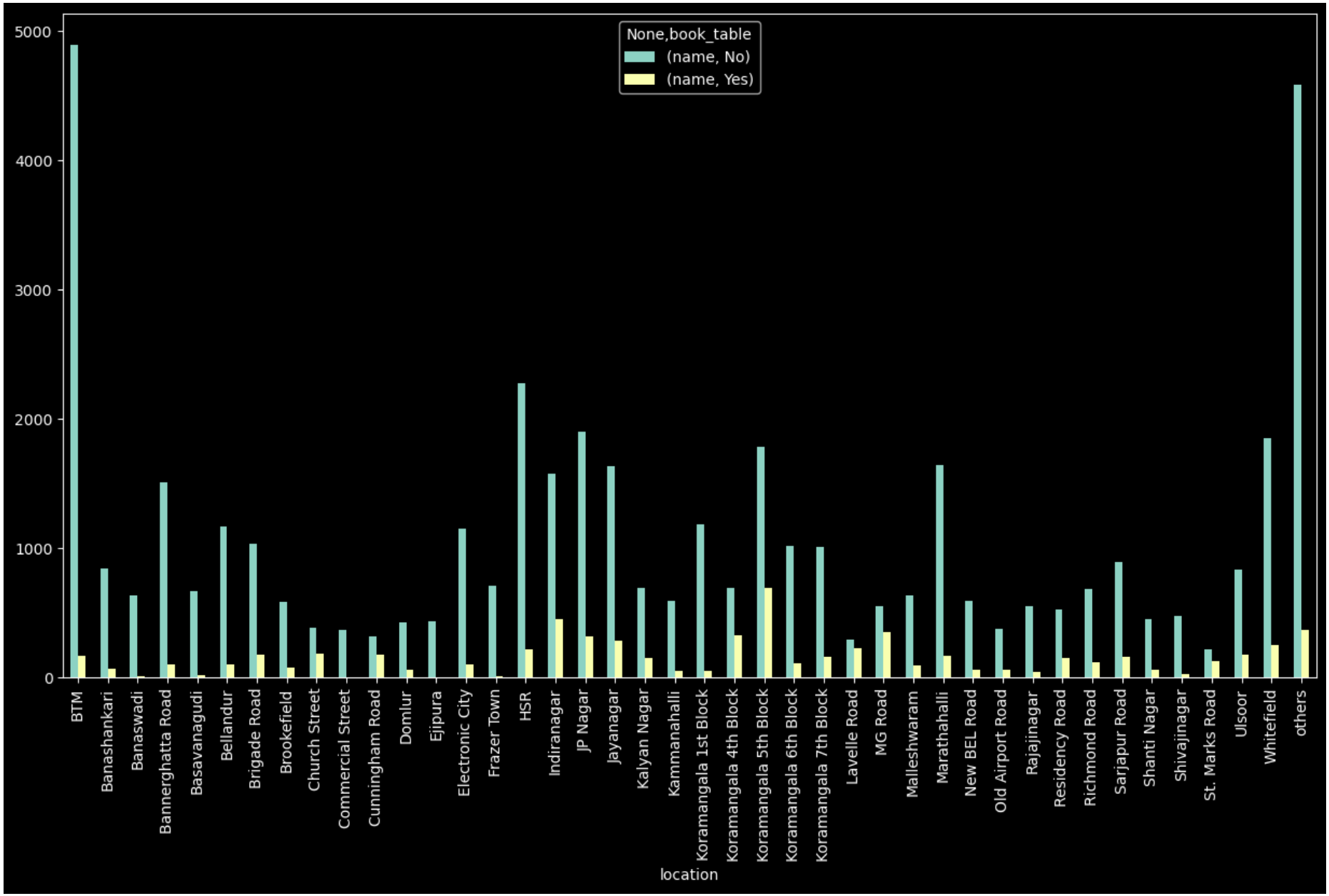
Out[54]:

name		
book_table	No	Yes
location		
BTM	4889	167
Banashankari	839	63
Banaswadi	632	8
Bannerghatta Road	1510	99
Basavanagudi	668	16
Bellandur	1170	98
Brigade Road	1034	176
Brookefield	582	74
Church Street	385	181
Commercial Street	370	0
Cunningham Road	315	175
Domlur	427	55
Ejipura	433	0
Electronic City	1148	98
Frazer Town	706	8
HSR	2277	217
Indiranagar	1578	448
JP Nagar	1903	315
Jayanagar	1637	279
Kalyan Nagar	692	149
Kammanahalli	590	49
Koramangala 1st Block	1186	50

name		
book_table	No	Yes
location		
Koramangala 4th Block	695	322
Koramangala 5th Block	1787	692
Koramangala 6th Block	1015	112
Koramangala 7th Block	1012	162
Lavelle Road	290	228
MG Road	546	347
Malleshwaram	632	89
Marathahalli	1642	163
New BEL Road	588	56
Old Airport Road	378	59
Rajajinagar	550	41
Residency Road	522	149
Richmond Road	687	116
Sarjapur Road	893	154
Shanti Nagar	451	57
Shivajinagar	475	23
St. Marks Road	219	124
Ulsoor	834	177
Whitefield	1852	253
others	4587	367

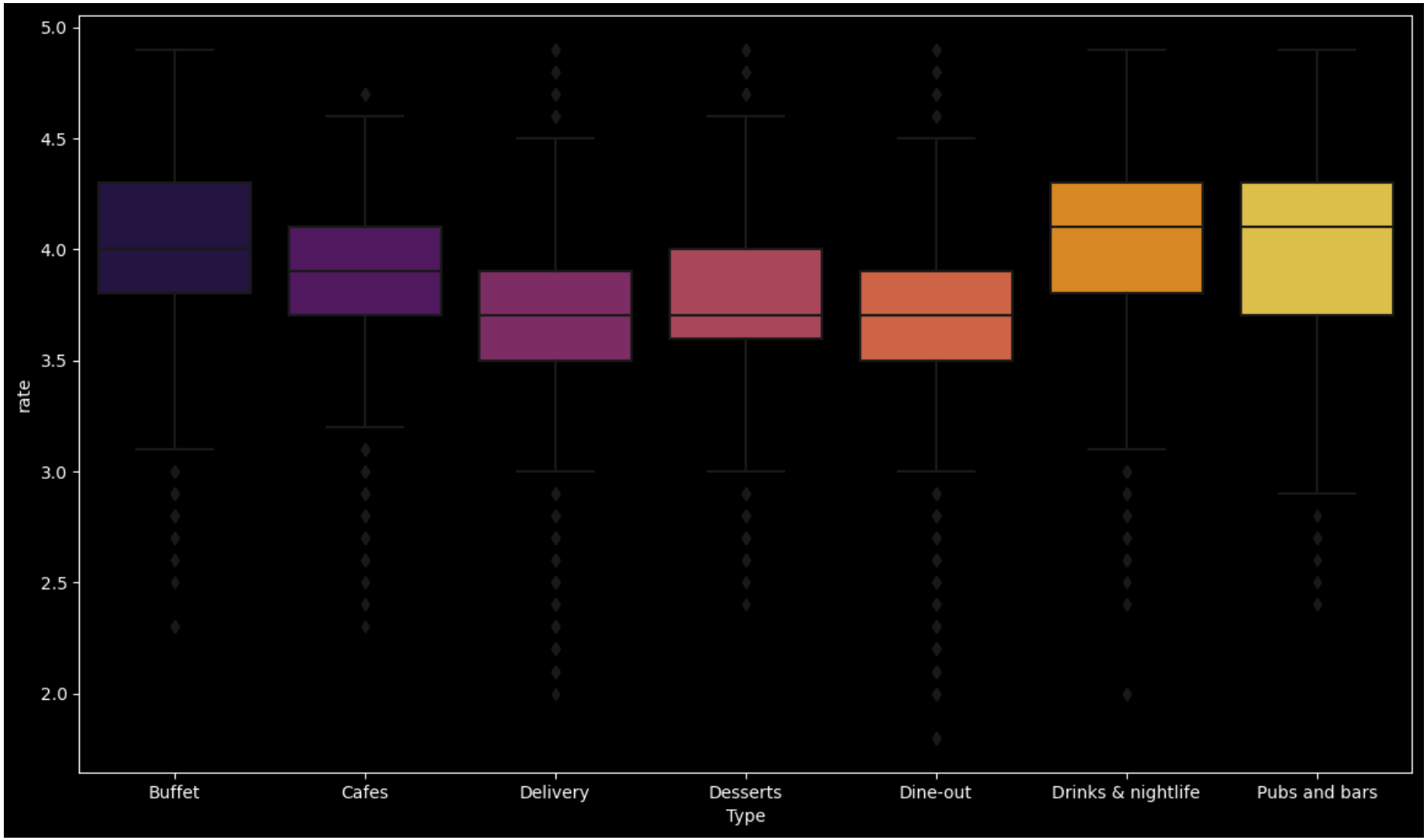
In [55]:
```
## plotted the graph for better understanding

df2.plot(kind = 'bar', figsize = (15,8))
```

Out[55]: <Axes: xlabel='location'>



In [56]:
```
##visualizing Types of Restaurents vs Rate
```

```
plt.figure(figsize = (14, 8))
sns.boxplot(x = 'Type', y = 'rate', data = df, palette = 'inferno')
plt.show()
```



In [57]:
```
#Grouping Types of Restaurents, location wise

df3 = df.groupby(['location','Type'])['name'].count()
df3.to_csv('location_Type.csv')
df3 = pd.read_csv('location_Type.csv')
```

```
df3 = pd.pivot_table(df3, values=None, index=['location'], columns=['Type'], fill_value=0, aggfunc=np.sum)

df3
```

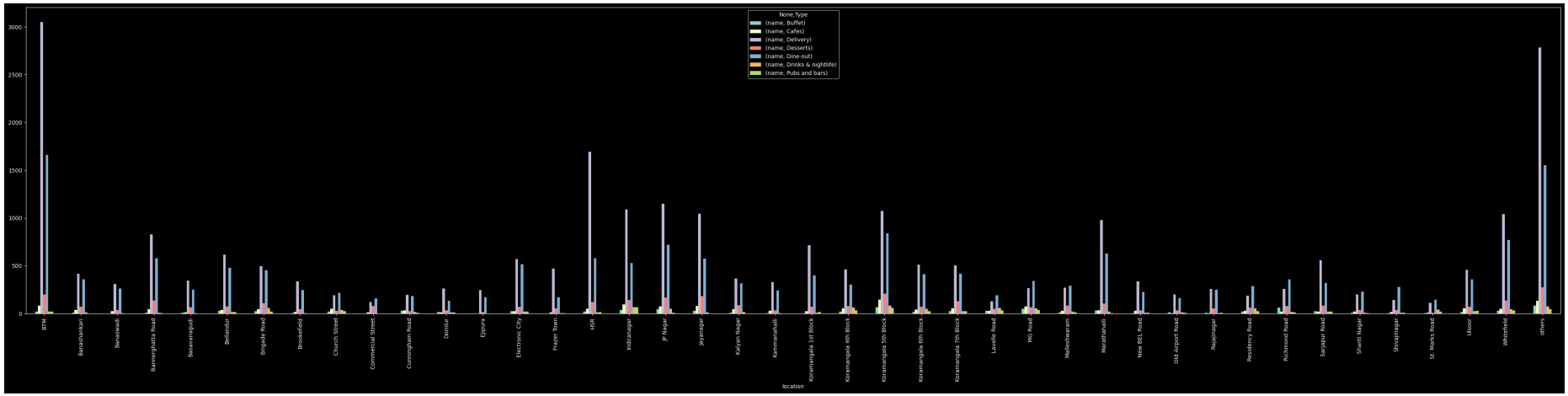
Out[57]:

name							
Type	Buffet	Cafes	Delivery	Desserts	Dine-out	Drinks & nightlife	Pubs and bars
location							
BTM	21	83	3053	198	1660	22	19
Banashankari	7	36	418	71	356	14	0
Banaswadi	0	24	310	37	262	6	1
Bannerghatta Road	9	46	828	137	578	9	2
Basavanagudi	7	11	344	66	251	5	0
Bellandur	28	36	617	75	479	17	16
Brigade Road	25	46	497	108	455	57	22
Brookefield	6	17	339	45	245	4	0
Church Street	19	51	193	29	215	36	23
Commercial Street	0	13	121	77	159	0	0
Cunningham Road	29	34	194	26	184	16	7
Domlur	15	13	261	35	135	12	11
Ejipura	0	0	245	16	172	0	0
Electronic City	23	24	570	71	516	21	21
Frazer Town	1	11	470	56	172	2	2
HSR	19	49	1694	120	580	14	18
Indiranagar	38	97	1091	140	529	65	66
JP Nagar	45	76	1151	166	722	51	7
Jayanagar	27	77	1043	182	575	12	0
Kalyan Nagar	9	45	366	88	315	18	0
Kammanahalli	2	27	329	35	240	6	0
Koramangala 1st Block	3	26	716	70	398	7	16

name Type location	Buffet	Cafes	Delivery	Desserts	Dine-out	Drinks & nightlife	Pubs and bars
Koramangala 4th Block	21	53	464	81	302	62	34
Koramangala 5th Block	65	146	1075	209	842	84	58
Koramangala 6th Block	18	43	511	70	411	51	23
Koramangala 7th Block	25	52	503	127	417	25	25
Lavelle Road	30	27	127	50	191	59	34
MG Road	51	76	266	68	343	53	36
Malleshwaram	11	31	269	85	291	20	14
Marathahalli	34	32	980	105	630	22	2
New BEL Road	4	29	338	33	224	8	8
Old Airport Road	12	5	200	35	164	12	9
Rajajinagar	10	4	258	55	251	3	10
Residency Road	20	31	187	63	289	55	26
Richmond Road	63	21	257	78	356	16	12
Sarjapur Road	25	22	558	82	319	19	22
Shanti Nagar	9	22	198	39	229	9	2
Shivajinagar	6	17	143	37	280	7	8
St. Marks Road	5	10	111	10	145	40	22
Ulsoor	16	56	456	71	359	23	30
Whitefield	28	51	1041	137	768	47	33
others	83	133	2787	276	1553	75	47

In [80]:
```
df3.plot(kind = 'bar', figsize = (50,10))
```

Out[80]: <Axes: xlabel='location'>



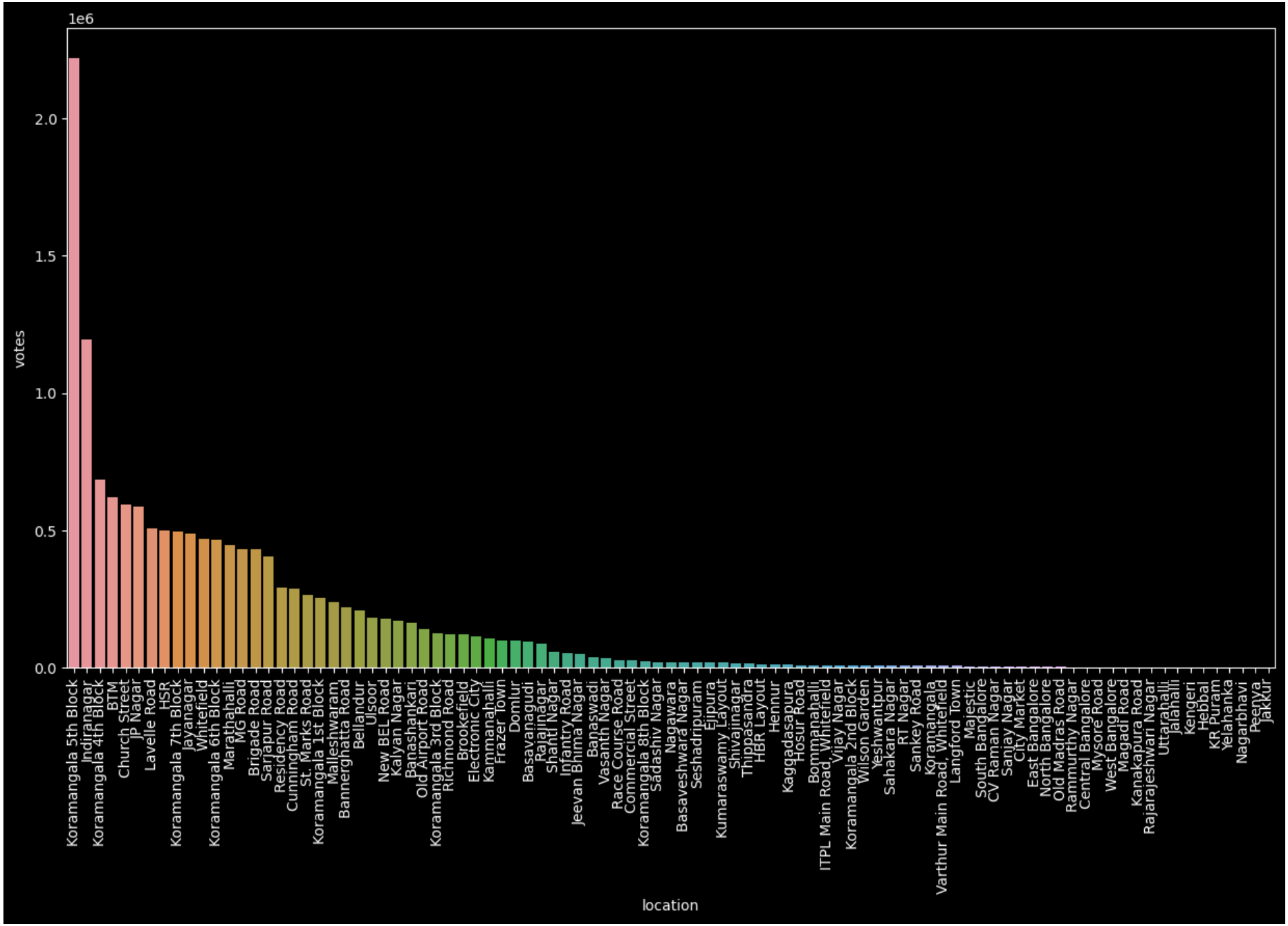
In [82]:
```
#No. of Votes, Location Wise

df4 = df[['location', 'votes']]
df4.drop_duplicates()
df5 = df4.groupby(['location'])['votes'].sum()
df5 = df5.to_frame()
df5 = df5.sort_values('votes', ascending=False)
df5.head()
```

Out[82]:

location	votes
Koramangala 5th Block	2219506
Indiranagar	1196007
Koramangala 4th Block	685156
BTM	619376
Church Street	594979

In [83]:
```
## plotted the graphplot accordlingly.
plt.figure(figsize=(15, 8))
sns.barplot(x=df5.index, y=df5['votes'])
```

```
plt.xticks(rotation=90)
plt.show()
```


1e6
2.0
1.5
1.0
0.5
0.0
votes
Koramangala 5th Block
Indiranagar
Koramangala 4th Block
BTM
Church Street
JP Nagar
Lavelle Road
HSR
Koramangala 7th Block
Jayanagar
Whitefield
Koramangala 6th Block
Marathahalli
MG Road
Brigade Road
Sarjapur Road
Residency Road
Cunningham Road
St. Marks Road
Koramangala 1st Block
Malleshwaram
Bannerghatta Road
Bellandur
Ulsoor
New BEL Road
Kalyan Nagar
Banashankari
Old Airport Road
Koramangala 3rd Block
Richmond Road
Brookefield
Electronic City
Kammanahalli
Frazer Town
Domlur
Basavanagudi
Rajajinagar
Shanti Nagar
Infantry Road
Jeevan Bhima Nagar
Banaswadi
Vasanth Nagar
Race Course Road
Commercial Street
Koramangala 8th Block
Sadashiv Nagar
Nagawara
Basaveshwara Nagar
Seshadripuram
Ejipura
Kumaraswamy Layout
Shivajinagar
Thippasandra
HBR Layout
Hennur
Kaggadasapura
Hosur Road
Bommanahalli
ITPL Main Road, Whitefield
Vijay Nagar
Koramangala 2nd Block
Wilson Garden
Yeshwantpur
Sahakara Nagar
RT Nagar
Sankey Road
Koramangala
Varthur Main Road, Whitefield
Langford Town
Majestic
South Bangalore
CV Raman Nagar
Sanjay Nagar
City Market
East Bangalore
North Bangalore
Old Madras Road
Rammurthy Nagar
Central Bangalore
Mysore Road
West Bangalore
Magadi Road
Kanakapura Road
Rajarajeshwari Nagar
Uttarahalli
Jalahalli
Kengeri
Hebbal
KR Puram
Yelahanka
Nagarbhavi
Peenya
Jakkur
location

In [71]:
```
df.head()
```

Out[71]:

	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

In [72]:
```
##Visualizing Top Cuisines column.

df6 = df[['cuisines', 'votes']]
df6.drop_duplicates()
df7 = df6.groupby(['cuisines'])['votes'].sum()
df7 = df7.to_frame()
df7 = df7.sort_values('votes', ascending=False)
df7.head()
```

Out[72]:

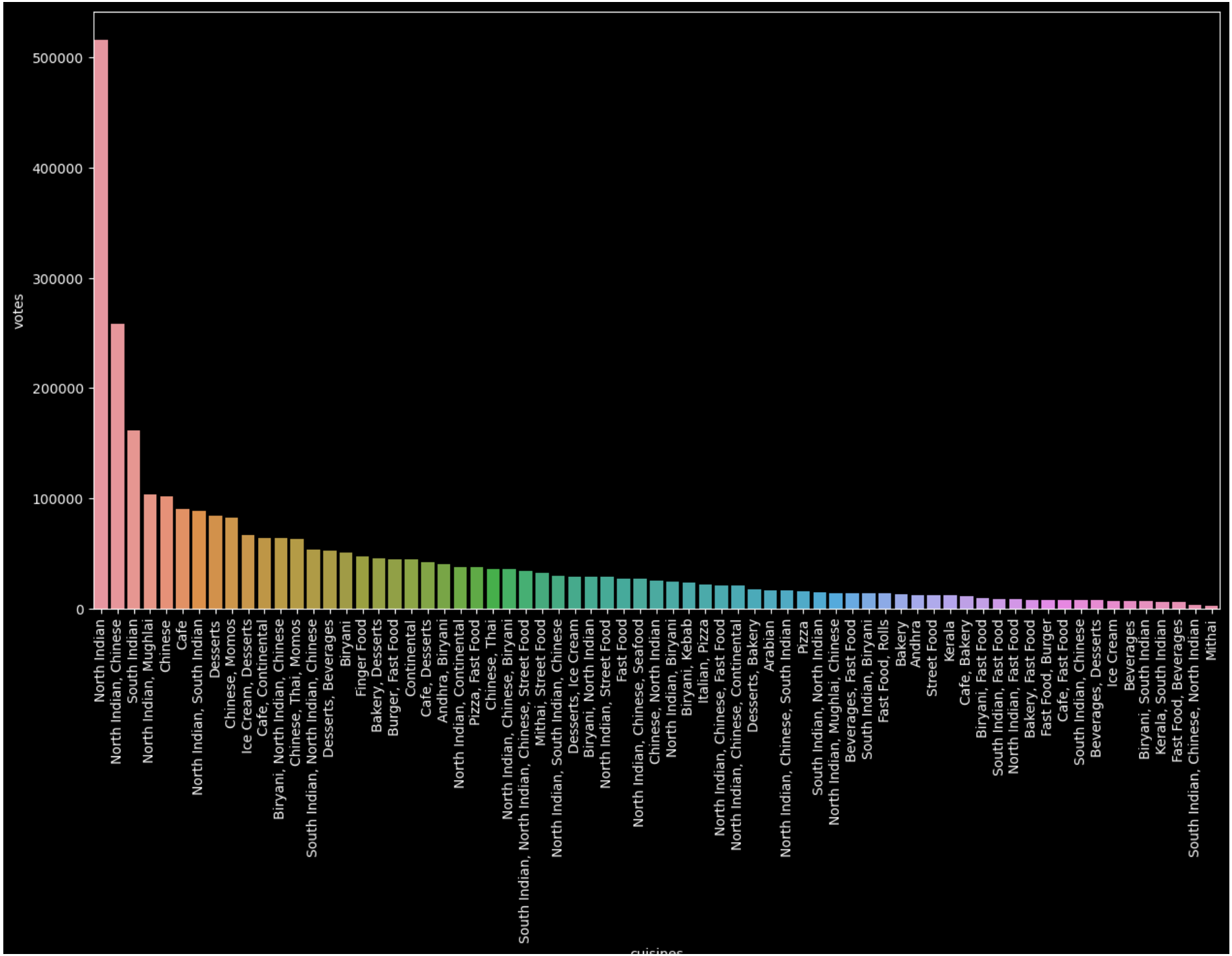
cuisines	votes
others	11542182
North Indian	516310
North Indian, Chinese	258225
South Indian	161975
North Indian, Mughlai	103706

In [73]:
```
# using the silicing for the selecting the column.
df7 = df7.iloc[1:, :]
df7.head()
```

Out[73]:

cuisines	votes
North Indian	516310
North Indian, Chinese	258225
South Indian	161975
North Indian, Mughlai	103706
Chinese	101728

In [76]:
```
## plotted the cuisines graph.

plt.figure(figsize=(15, 8))
sns.barplot(x=df7.index, y=df7['votes'])
plt.xticks(rotation=90)
plt.show()
```


votes
0
100000
200000
300000
400000
500000
North Indian
North Indian, Chinese
South Indian
North Indian, Mughlai
Chinese
Cafe
North Indian, South Indian
Desserts
Chinese, Momos
Ice Cream, Desserts
Cafe, Continental
Biryani, North Indian, Chinese
Chinese, Thai, Momos
South Indian, North Indian, Chinese
Desserts, Beverages
Biryani
Finger Food
Bakery, Desserts
Burger, Fast Food
Continental
Cafe, Desserts
Andhra, Biryani
North Indian, Continental
Pizza, Fast Food
Chinese, Thai
North Indian, Chinese, Biryani
South Indian, North Indian, Chinese, Street Food
Mithai, Street Food
North Indian, South Indian, Chinese
Desserts, Ice Cream
Biryani, North Indian
North Indian, Street Food
Fast Food
North Indian, Chinese, Seafood
Chinese, North Indian
North Indian, Biryani
Biryani, Kebab
Italian, Pizza
North Indian, Chinese, Fast Food
North Indian, Chinese, Continental
Desserts, Bakery
Arabian
North Indian, Chinese, South Indian
Pizza
South Indian, North Indian
North Indian, Mughlai, Chinese
Beverages, Fast Food
South Indian, Biryani
Fast Food, Rolls
Bakery
Andhra
Street Food
Kerala
Cafe, Bakery
Biryani, Fast Food
South Indian, Fast Food
North Indian, Fast Food
Bakery, Fast Food
Fast Food, Burger
Cafe, Fast Food
South Indian, Chinese
Beverages, Desserts
Ice Cream
Beverages
Biryani, South Indian
Kerala, South Indian
Fast Food, Beverages
South Indian, Chinese, North Indian
Mithai
cuisines

cuisines

In [ ]: