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

In [2]:
```
df=pd.read_csv('zomato_dataset.csv')
df.head()
```

Out[2]:

	Restaurant Name	Dining Rating	Delivery Rating	Dining Votes	Delivery Votes	Cuisine	Place Name	City	Item Name	Best
0	Doner King	3.9	4.2	39	0	Fast Food	Malakpet	Hyderabad	Platter Kebab Combo	BESTS
1	Doner King	3.9	4.2	39	0	Fast Food	Malakpet	Hyderabad	Chicken Rumali Shawarma	BESTS
2	Doner King	3.9	4.2	39	0	Fast Food	Malakpet	Hyderabad	Chicken Tandoori Salad	
3	Doner King	3.9	4.2	39	0	Fast Food	Malakpet	Hyderabad	Chicken BBQ Salad	BESTS
4	Doner King	3.9	4.2	39	0	Fast Food	Malakpet	Hyderabad	Special Doner Wrap Combo	MUS

In [3]:
```
df.tail()
```

Out[3]:

	Restaurant Name	Dining Rating	Delivery Rating	Dining Votes	Delivery Votes	Cuisine	Place Name	City	Item Name	Best
123652	Ariena Boutique Hotel	3.9	4.2	13	523	Pizza	Purena	Raipur	Murgh Reshmi Kebab	
123653	Ariena Boutique Hotel	3.9	4.2	13	523	Pizza	Purena	Raipur	Murgh Large Tikka	
123654	Ariena Boutique Hotel	3.9	4.2	13	523	Pizza	Purena	Raipur	Murgh Chukandri Tikka	
123655	Ariena Boutique Hotel	3.9	4.2	13	523	Pizza	Purena	Raipur	Murgh Golden Kebab	
123656	Ariena Boutique Hotel	3.9	4.2	13	523	Pizza	Purena	Raipur	Gosht Gilawat Chop	BESTS

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

Out[4]:
```
(123657, 12)
```

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 123657 entries, 0 to 123656
Data columns (total 12 columns):
 #   Column           Non-Null Count   Dtype  
---  ------           --------------   -----  
 0   Restaurant Name  123657 non-null  object 
 1   Dining Rating    91421 non-null   float64
 2   Delivery Rating  122377 non-null  float64
 3   Dining Votes     123657 non-null  int64  
 4   Delivery Votes   123657 non-null  int64  
 5   Cuisine          123657 non-null  object 
 6   Place Name       123657 non-null  object 
 7   City             123657 non-null  object 
 8   Item Name        123657 non-null  object 
 9   Best Seller      27942 non-null   object 
 10  Votes            123657 non-null  int64  
 11  Prices           123657 non-null  float64
dtypes: float64(3), int64(3), object(6)
memory usage: 11.3+ MB
```

In [6]:
```
df.isnull().sum()
```

Out[6]:
```
Restaurant Name        0
Dining Rating      32236
Delivery Rating     1280
Dining Votes           0
Delivery Votes         0
Cuisine                0
Place Name             0
City                   0
Item Name              0
Best Seller        95715
Votes                  0
Prices                 0
dtype: int64
```

In [7]:
```
df.drop(['Best Seller'],inplace=True,axis=1)
```

In [8]:
```
mean=np.mean(df['Dining Rating'])
mean
```

Out[8]:
```
3.822264031240087
```

In [9]:
```
df['Dining Rating']=df['Dining Rating'].replace(np.nan,3.8)
```

In [10]:
```
df.isnull().sum()
```

Out[10]:
```
Restaurant Name       0
Dining Rating         0
Delivery Rating    1280
Dining Votes          0
Delivery Votes        0
Cuisine               0
Place Name            0
City                  0
Item Name             0
Votes                 0
Prices                0
dtype: int64
```

In [11]:
```
print(np.mean(df['Delivery Rating']))
```

```
3.9631842584799433
```

In [12]:
```
df['Delivery Rating']=df['Delivery Rating'].replace(np.nan,3.9)
```

In [15]:
```
df.isnull().sum()
```

Out[15]:
```
Restaurant Name    0
Dining Rating      0
Delivery Rating    0
Dining Votes       0
Delivery Votes     0
Cuisine            0
Place Name         0
City               0
Item Name          0
Votes              0
Prices             0
dtype: int64
```

In [16]:
```
df.describe()
```

Out[16]:

	Dining Rating	Delivery Rating	Dining Votes	Delivery Votes	Votes	Prices
count	123657.000000	123657.000000	123657.000000	123657.000000	123657.000000	123657.000000
mean	3.816460	3.962530	152.729858	115.763725	24.666772	241.378399
std	0.351543	0.244708	232.214061	243.970828	125.236009	192.830713
min	2.500000	2.500000	0.000000	0.000000	0.000000	0.950000
25%	3.700000	3.800000	0.000000	0.000000	0.000000	130.000000
50%	3.800000	4.000000	30.000000	0.000000	0.000000	208.570000
75%	4.000000	4.100000	217.000000	23.000000	15.000000	299.000000
max	4.800000	4.600000	997.000000	983.000000	9750.000000	12024.000000

In [17]:
```
df.duplicated().sum()
```

Out[17]:
```
26322
```

In [18]:
```
df=df.drop_duplicates()
```

In [19]:
```
df.duplicated().sum()
```

Out[19]:
```
0
```

In [20]:
```
df.describe()
```

Out[20]:

	Dining Rating	Delivery Rating	Dining Votes	Delivery Votes	Votes	Prices
count	97335.000000	97335.000000	97335.000000	97335.000000	97335.000000	97335.000000
mean	3.815949	3.958819	152.631345	115.579771	16.701998	244.016323
std	0.350789	0.244124	231.127900	242.644336	98.271749	198.468133
min	2.500000	2.500000	0.000000	0.000000	0.000000	0.950000
25%	3.700000	3.800000	0.000000	0.000000	0.000000	130.000000
50%	3.800000	4.000000	30.000000	0.000000	0.000000	209.000000
75%	4.000000	4.100000	221.000000	32.000000	9.000000	299.000000
max	4.800000	4.600000	997.000000	983.000000	9750.000000	12024.000000

In [21]:
```
df.corr()
```

```
C:\Users\ANMOL\AppData\Local\Temp\ipykernel_30916\1134722465.py:1: FutureWarning:
The default value of numeric_only in DataFrame.corr is deprecated. In a future ver
sion, it will default to False. Select only valid columns or specify the value of
numeric_only to silence this warning.
  df.corr()
```

Out[21]:

	Dining Rating	Delivery Rating	Dining Votes	Delivery Votes	Votes	Prices
Dining Rating	1.000000	0.262485	0.235939	-0.112168	0.034723	0.058239
Delivery Rating	0.262485	1.000000	0.143883	-0.063411	0.043759	0.053642
Dining Votes	0.235939	0.143883	1.000000	-0.246941	0.004984	0.016352
Delivery Votes	-0.112168	-0.063411	-0.246941	1.000000	-0.054184	0.012276
Votes	0.034723	0.043759	0.004984	-0.054184	1.000000	-0.053287
Prices	0.058239	0.053642	0.016352	0.012276	-0.053287	1.000000

In [23]:
```
df.columns
```

Out[23]:
```
Index(['Restaurant Name', 'Dining Rating', 'Delivery Rating', 'Dining Votes',
       'Delivery Votes', 'Cuisine ', 'Place Name', 'City', 'Item Name',
       'Votes', 'Prices'],
      dtype='object')
```

In [24]:
```
df.nunique()
```

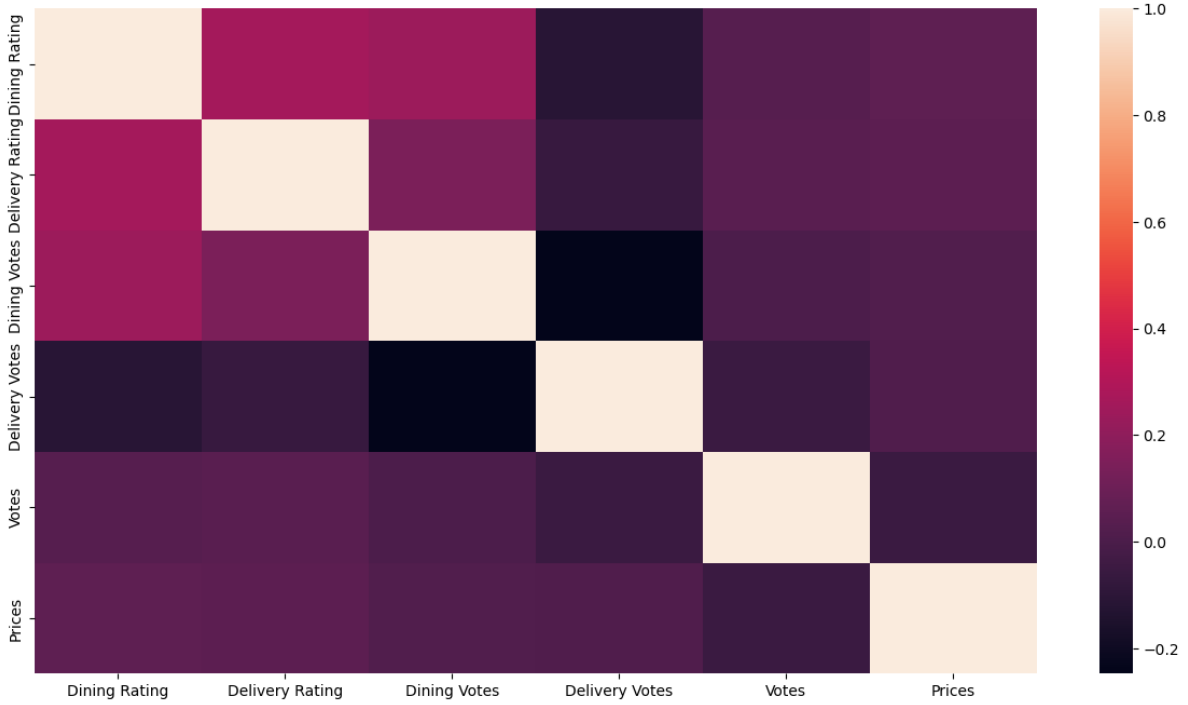
Out[24]:
```
Restaurant Name      826
Dining Rating         24
Delivery Rating       18
Dining Votes         294
Delivery Votes       263
Cuisine               48
Place Name           324
City                  17
Item Name          55693
Votes                760
Prices              2710
dtype: int64
```

In [25]:
```
df['City'].unique()
```

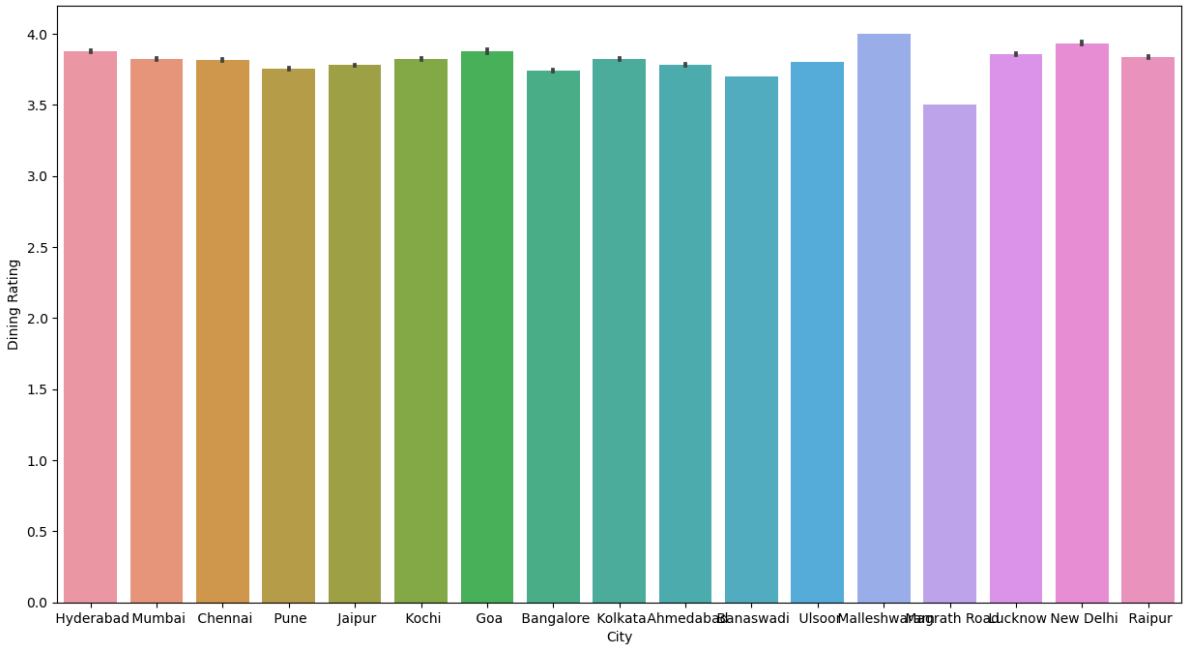
Out[25]:
```
array([' Hyderabad', ' Mumbai', ' Chennai', ' Pune', ' Jaipur', ' Kochi',
       ' Goa', ' Bangalore', ' Kolkata', ' Ahmedabad', ' Banaswadi',
       ' Ulsoor', ' Malleshwaram', ' Magrath Road', ' Lucknow',
       ' New Delhi', ' Raipur'], dtype=object)
```

In [26]:
```
plt.figure(figsize=(15,8))
sns.heatmap(df.corr())
plt.show()
```

```
C:\Users\ANMOL\AppData\Local\Temp\ipykernel_30916\1966855796.py:2: FutureWarning:
The default value of numeric_only in DataFrame.corr is deprecated. In a future ver
sion, it will default to False. Select only valid columns or specify the value of
numeric_only to silence this warning.
  sns.heatmap(df.corr())
```

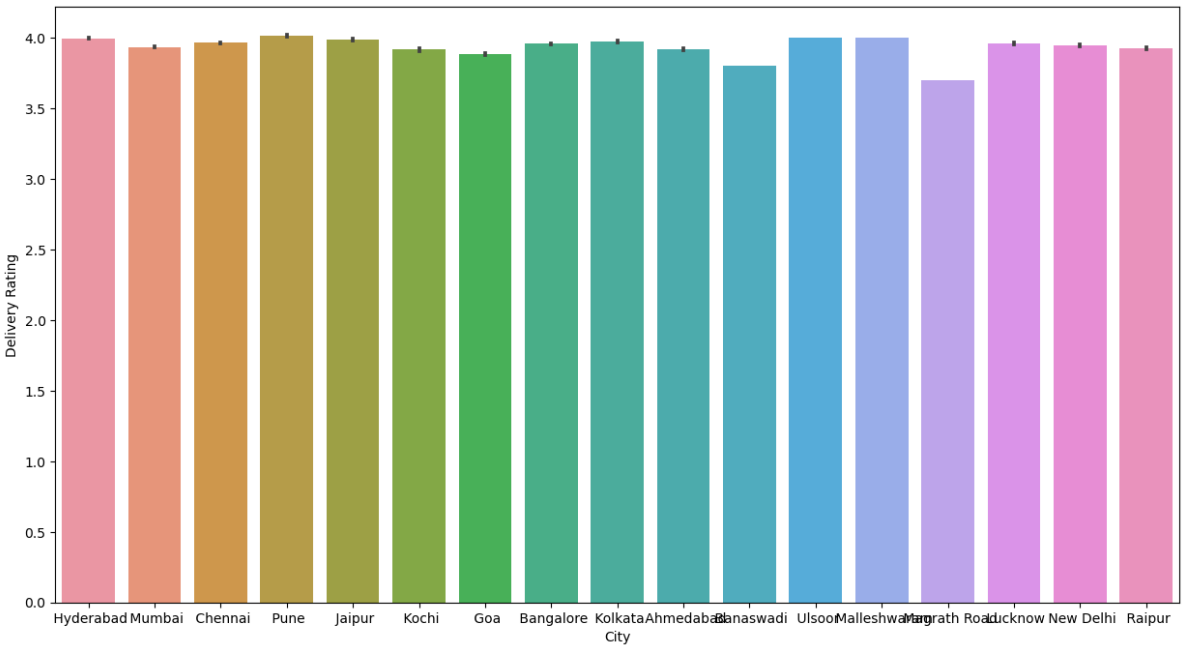


In [28]:
```
plt.figure(figsize=(15,8))
sns.barplot(x='City',y='Dining Rating',data=df)
plt.show()
```



In [29]:
```
plt.figure(figsize=(15,8))
sns.barplot(x='City',y='Delivery Rating',data=df)
```

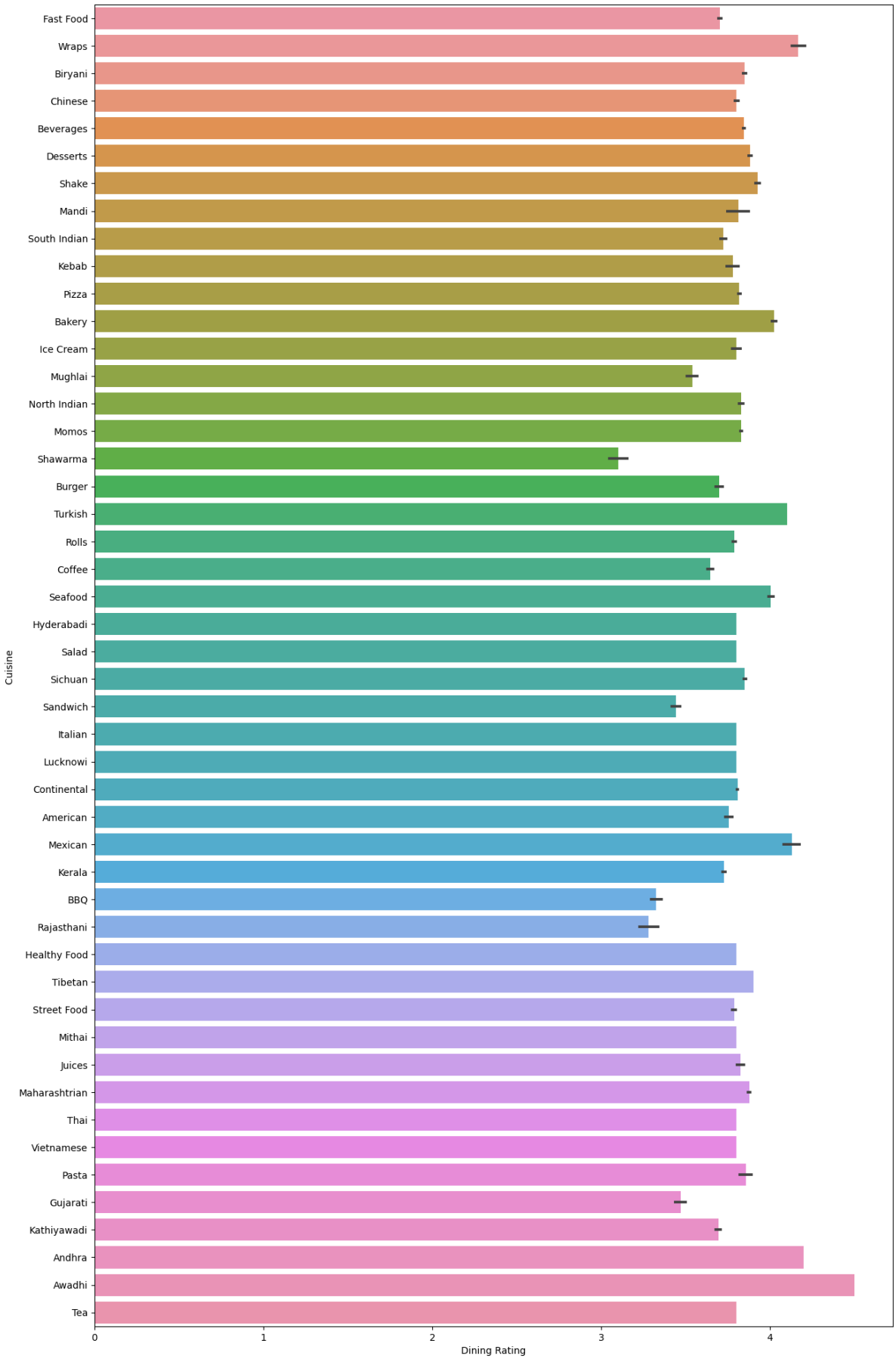
```
plt.show()
```



In [31]:
```
df['Cuisine '].unique()
```

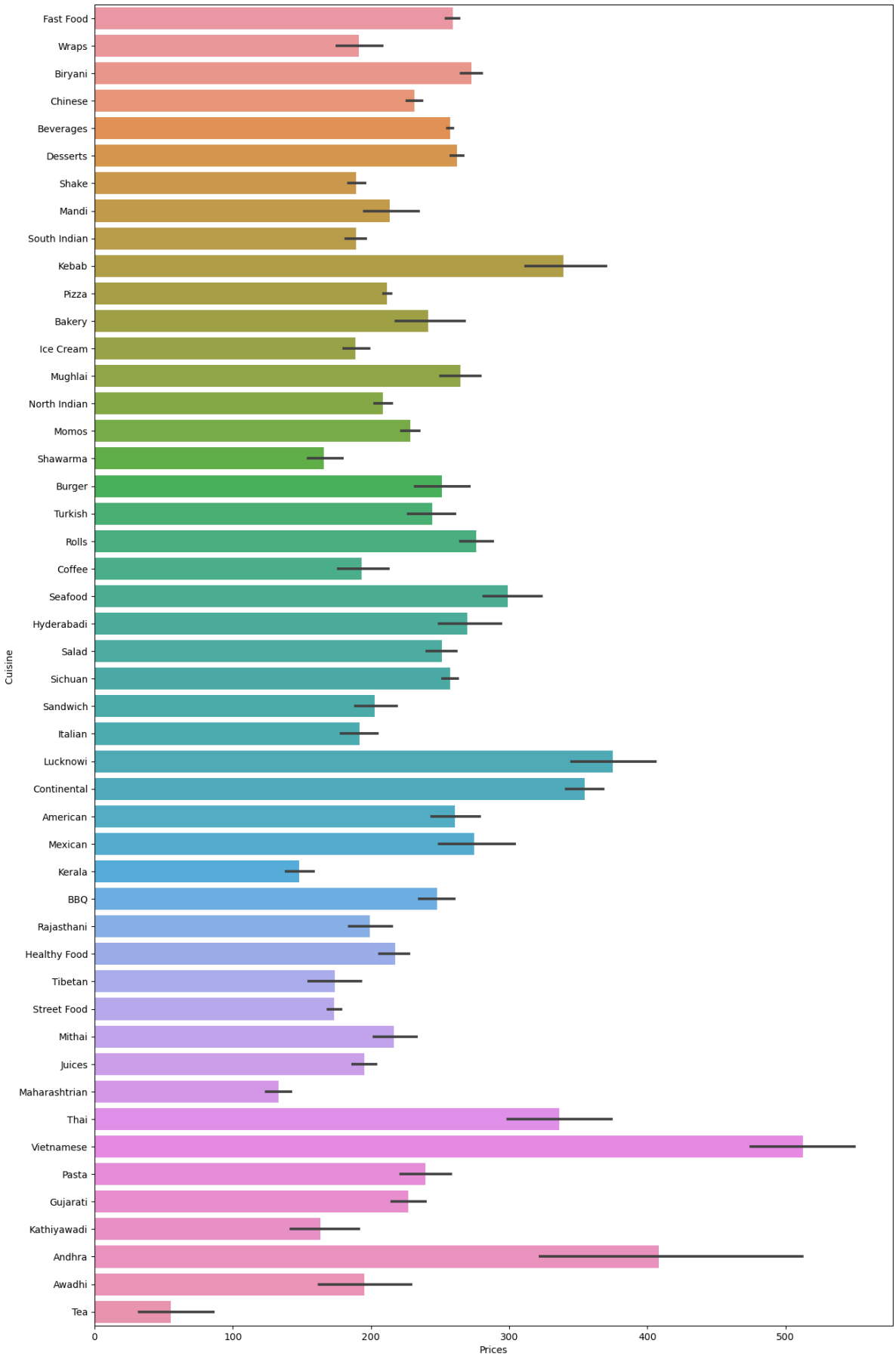
Out[31]:
```
array(['Fast Food', 'Wraps', 'Biryani', 'Chinese', 'Beverages',
       'Desserts', 'Shake', 'Mandi', 'South Indian', 'Kebab', 'Pizza',
       'Bakery', 'Ice Cream', 'Mughlai', 'North Indian', 'Momos',
       'Shawarma', 'Burger', 'Turkish', 'Rolls', 'Coffee', 'Seafood',
       'Hyderabadi', 'Salad', 'Sichuan', 'Sandwich', 'Italian',
       'Lucknowi', 'Continental', 'American', 'Mexican', 'Kerala', 'BBQ',
       'Rajasthani', 'Healthy Food', 'Tibetan', 'Street Food', 'Mithai',
       'Juices', 'Maharashtrian', 'Thai', 'Vietnamese', 'Pasta',
       'Gujarati', 'Kathiyawadi', 'Andhra', 'Awadhi', 'Tea'], dtype=object)
```

In [32]:
```
plt.figure(figsize=(15,25))
sns.barplot(x='Dining Rating',y='Cuisine ',data=df)
plt.show()
```



In [34]:

```
plt.figure(figsize=(15,25))
sns.barplot(x='Prices',y='Cuisine ',data=df)
plt.show()
```



In [35]:
```
#Average Dining Rating
print(np.mean(df['Dining Rating']))
```

```
3.815949041968459
```

In [36]:
```
#Metropolitan having Highest Average Delivery Rating
gb=df.groupby('City')
```

```
gb.mean()
```

```
C:\Users\ANMOL\AppData\Local\Temp\ipykernel_30916\1516328926.py:3: FutureWarning:
The default value of numeric_only in DataFrameGroupBy.mean is deprecated. In a fut
ure version, numeric_only will default to False. Either specify numeric_only or se
lect only columns which should be valid for the function.
  gb.mean()
```

Out[36]:

City	Dining Rating	Delivery Rating	Dining Votes	Delivery Votes	Votes	Prices
Ahmedabad	3.782216	3.917175	142.973428	114.925402	12.531936	225.248306
Banaswadi	3.700000	3.800000	47.000000	139.000000	0.976471	349.466471
Bangalore	3.740401	3.957338	118.559854	94.002549	15.908335	231.867958
Chennai	3.818042	3.964381	206.660974	126.735592	11.900325	259.707505
Goa	3.877447	3.881912	20.161028	215.647722	0.693730	223.341547
Hyderabad	3.880092	3.996788	136.075731	99.663508	34.876486	248.855075
Jaipur	3.779186	3.988339	192.196016	96.865439	20.038057	223.468007
Kochi	3.823380	3.915401	159.499411	117.783538	13.861134	228.144257
Kolkata	3.822085	3.974234	200.036808	52.443657	21.644592	237.867345
Lucknow	3.855960	3.958623	244.714375	108.563969	23.239026	238.958427
Magrath Road	3.500000	3.700000	0.000000	112.000000	1.800000	240.314000
Malleshwaram	4.000000	4.000000	746.000000	0.000000	17.161290	189.354839
Mumbai	3.825207	3.935348	143.072082	150.904364	8.189139	306.032386
New Delhi	3.934645	3.945792	198.145719	52.298361	15.345355	242.091027
Pune	3.752234	4.015382	87.548809	142.068504	10.840262	245.982877
Raipur	3.834241	3.924095	68.001328	166.759880	9.956161	197.693584
Ulsoor	3.800000	4.000000	0.000000	7.000000	0.000000	704.388889

In [37]:
```
df.groupby(['City','Restaurant Name'])['Dining Votes'].sum()
```

Out[37]:
```
City        Restaurant Name
 Ahmedabad  1944 -The HOCCO Kitchen            0
            A-One Bombay Biryani             450
            Al Baik Fast Food               2340
            Alinea Restaurant & Banquet    45738
            Alpine Restaurant & Banquet    56760
                                             ...
 Raipur     Ustaad's Kitchen                5112
            Veggiies                        9548
            Wafflez                         5472
            Xero Degrees                       0
 Ulsoor     Dum Safar Biryani                  0
Name: Dining Votes, Length: 907, dtype: int64
```

In [38]:
```
gb.sum()
```

```
C:\Users\ANMOL\AppData\Local\Temp\ipykernel_30916\624691301.py:1: FutureWarning: T
he default value of numeric_only in DataFrameGroupBy.sum is deprecated. In a futur
e version, numeric_only will default to False. Either specify numeric_only or sele
ct only columns which should be valid for the function.
  gb.sum()
```

Out[38]:

City	Dining Rating	Delivery Rating	Dining Votes	Delivery Votes	Votes	Prices
Ahmedabad	31029.3	32136.5	1172954	942848	102812	1847937.10
Banaswadi	314.5	323.0	3995	11815	83	29704.65
Bangalore	33745.9	35703.1	1069647	848091	143525	2091912.72
Chennai	41139.4	42716.2	2226772	1365576	128226	2798348.37
Goa	8596.3	8606.2	44697	478091	1538	495148.21
Hyderabad	48314.9	49768.0	1694415	1241010	434282	3098743.39
Jaipur	40218.1	42443.9	2045350	1030842	213245	2378146.53
Kochi	22714.7	23261.4	947586	699752	82349	1355405.03
Kolkata	25336.6	26345.2	1326044	347649	143482	1576822.63
Lucknow	18535.6	19029.1	1176342	521867	111710	1148673.16
Magrath Road	122.5	129.5	0	3920	63	8410.99
Malleshwaram	124.0	124.0	23126	0	532	5870.00
Mumbai	43037.4	44276.6	1609704	1697825	92136	3443170.37
New Delhi	10800.6	10831.2	543910	143559	42123	664539.87
Pune	24100.6	25790.8	562326	912506	69627	1579948.02
Raipur	23089.8	23630.9	409504	1004228	59956	1190510.76
Ulsoor	205.2	216.0	0	378	0	38037.00

In [39]:
```
#Restaurant having Highest Average Dining Rating in each City
idx=gb['Dining Rating'].idxmax()
rate=df.loc[idx]
print(rate[['Restaurant Name','City','Dining Rating']])
```

```
                Restaurant Name          City  Dining Rating
89172             Urban Khichdi     Ahmedabad            4.6
94473                    GOPIZZA     Banaswadi            3.7
65625                   Truffles     Bangalore            4.6
32723   AB's - Absolute Barbecues       Chennai            4.7
63090              Ritz Classic           Goa            4.4
15354                    Exotica     Hyderabad            4.6
76069           Thali and More         Jaipur            4.7
56766                  Cafe 17          Kochi            4.6
72232                  Chowman        Kolkata            4.4
100331              Dastarkhwan        Lucknow            4.5
99791       Keventers Ice Cream   Magrath Road            3.5
96031                 Rajdhani   Malleshwaram            4.0
17436                Chaitanya         Mumbai            4.5
106300        Natural Ice Cream      New Delhi            4.8
43248                   Sukanta           Pune            4.2
118057        Creams N Caffeine         Raipur            4.3
94757         Dum Safar Biryani         Ulsoor            3.8
```