CITY ANALYSIS

IMPORTING LIBRARIES

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

DATA COLLECTION

In [2]:

```
df=pd.read_csv('Dataset .csv',encoding='unicode_escape')
```

In [3]:

```
df.head()
```

Out[3]:

	ï»¿Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	Latitude	Cuisines	...	C
0	6317637	Le Petit Souffle	162	Makati City	Third Floor, Century City Mall, Kalayaan Avenu...	Century City Mall, Poblacion, Makati City	Century City Mall, Poblacion, Makati City, Mak...	121.027535	14.565443	French, Japanese, Desserts	...	B
1	6304287	Izakaya Kikufuji	162	Makati City	Little Tokyo, 2277 Chino Roces Avenue, Legaspi...	Little Tokyo, Legaspi Village, Makati City	Little Tokyo, Legaspi Village, Makati City, Ma...	121.014101	14.553708	Japanese	...	B
2	6300002	Heat - Edsa Shangri-La	162	Mandaluyong City	Edsa Shangri-La, 1 Garden Way, Ortigas, Mandal...	Edsa Shangri-La, Ortigas, Mandaluyong City	Edsa Shangri-La, Ortigas, Mandaluyong City, Ma...	121.056831	14.581404	Seafood, Asian, Filipino, Indian	...	B
3	6318506	Ooma	162	Mandaluyong City	Third Floor, Mega Fashion Hall, SM Megamall, O...	SM Megamall, Ortigas, Mandaluyong City	SM Megamall, Ortigas, Mandaluyong City, Mandal...	121.056475	14.585318	Japanese, Sushi	...	B
4	6314302	Sambo Kojin	162	Mandaluyong City	Third Floor, Mega Atrium, SM Megamall, Ortigas...	SM Megamall, Ortigas, Mandaluyong City	SM Megamall, Ortigas, Mandaluyong City, Mandal...	121.057508	14.584450	Japanese, Korean	...	B

5 rows × 21 columns

In [4]:

```
df.shape
```

Out[4]:

```
(9551, 21)
```

In [5]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9551 entries, 0 to 9550
Data columns (total 21 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   ï»¿Restaurant ID      9551 non-null   int64  
 1   Restaurant Name       9551 non-null   object 
 2   Country Code          9551 non-null   int64  
 3   City                  9551 non-null   object 
 4   Address               9551 non-null   object 
 5   Locality              9551 non-null   object 
 6   Locality Verbose      9551 non-null   object 
 7   Longitude             9551 non-null   float64
 8   Latitude              9551 non-null   float64
 9   Cuisines              9542 non-null   object 
 10  Average Cost for two  9551 non-null   int64  
 11  Currency              9551 non-null   object 
 12  Has Table booking     9551 non-null   object 
 13  Has Online delivery   9551 non-null   object 
 14  Is delivering now     9551 non-null   object 
 15  Switch to order menu  9551 non-null   object 
 16  Price range           9551 non-null   int64  
 17  Aggregate rating      9551 non-null   float64
 18  Rating color          9551 non-null   object 
 19  Rating text           9551 non-null   object 
 20  Votes                 9551 non-null   int64  
dtypes: float64(3), int64(5), object(13)
memory usage: 1.5+ MB
```

TO CHECK NULL VALUES

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

Out[6]:
```
ï»¿Restaurant ID        0
Restaurant Name        0
Country Code           0
City                   0
Address                0
Locality               0
Locality Verbose       0
Longitude              0
Latitude               0
Cuisines               9
Average Cost for two   0
Currency               0
Has Table booking      0
Has Online delivery    0
Is delivering now      0
Switch to order menu   0
Price range            0
Aggregate rating       0
Rating color           0
Rating text            0
Votes                  0
dtype: int64
```

TO DELETE NULL VALUES

In [7]:
```
df.dropna(inplace=True)
```

In [9]:
```
df.shape
```

Out[9]:
```
(9542, 21)
```

In [10]:
```
df.describe()
```

Out[10]:

	ï»¿Restaurant ID	Country Code	Longitude	Latitude	Average Cost for two	Price range	Aggregate rating	Votes
count	9.542000e+03	9542.000000	9542.000000	9542.000000	9542.000000	9542.000000	9542.000000	9542.000000
mean	9.043301e+06	18.179208	64.274997	25.848532	1200.326137	1.804968	2.665238	156.772060
std	8.791967e+06	56.451600	41.197602	11.010094	16128.743876	0.905563	1.516588	430.203324
min	5.300000e+01	1.000000	-157.948486	-41.330428	0.000000	1.000000	0.000000	0.000000
25%	3.019312e+05	1.000000	77.081565	28.478658	250.000000	1.000000	2.500000	5.000000
50%	6.002726e+06	1.000000	77.192031	28.570444	400.000000	2.000000	3.200000	31.000000
75%	1.835260e+07	1.000000	77.282043	28.642711	700.000000	2.000000	3.700000	130.000000
max	1.850065e+07	216.000000	174.832089	55.976980	800000.000000	4.000000	4.900000	10934.000000

IDENTIFY THE CITY WITH THE HIGHEST NUMBER OF RESTAURANTS IN THE DATASET.

In [11]:

```
df.columns
```

Out[11]:

```
Index(['ï»¿Restaurant ID', 'Restaurant Name', 'Country Code', 'City',
       'Address', 'Locality', 'Locality Verbose', 'Longitude', 'Latitude',
       'Cuisines', 'Average Cost for two', 'Currency', 'Has Table booking',
       'Has Online delivery', 'Is delivering now', 'Switch to order menu',
       'Price range', 'Aggregate rating', 'Rating color', 'Rating text',
       'Votes'],
      dtype='object')
```

In [15]:

```
 # Group the data by city and count the number of restaurants in each city
city_restaurants_count = df.groupby('City')['Restaurant Name'].count()
```

In [16]:

```
# Find the city with the highest number of restaurants
city_with_most_restaurants = city_restaurants_count.idxmax()
num_restaurants = city_restaurants_count.max()
```

In [17]:

```
print(f"The city with the highest number of restaurants is {city_with_most_restaurants} with {num_restaurants}
```

```
The city with the highest number of restaurants is New Delhi with 5473 restaurants.
```

CALCULATE THE AVERAGE RATING FOR RESTAURANTS IN EACH CITY.

In [18]:

```
df.columns
```

Out[18]:

```
Index(['ï»¿Restaurant ID', 'Restaurant Name', 'Country Code', 'City',
       'Address', 'Locality', 'Locality Verbose', 'Longitude', 'Latitude',
       'Cuisines', 'Average Cost for two', 'Currency', 'Has Table booking',
       'Has Online delivery', 'Is delivering now', 'Switch to order menu',
       'Price range', 'Aggregate rating', 'Rating color', 'Rating text',
       'Votes'],
      dtype='object')
```

In [46]:

```
df.dtypes
```

Out[46]:

```
ï»¿Restaurant ID     int64
Restaurant Name     object
Country Code         int64
dtype: object
```

In [47]:

```
df=pd.read_csv('Dataset .csv',encoding='unicode_escape')
```

In [49]:

```
df.dtypes
```

Out[49]:

```
ï»¿Restaurant ID          int64
Restaurant Name         object
Country Code             int64
City                    object
Address                 object
Locality                object
Locality Verbose        object
Longitude              float64
Latitude               float64
Cuisines                object
Average Cost for two     int64
Currency                object
Has Table booking       object
Has Online delivery     object
Is delivering now       object
Switch to order menu    object
Price range              int64
Aggregate rating       float64
Rating color            object
Rating text             object
Votes                    int64
dtype: object
```

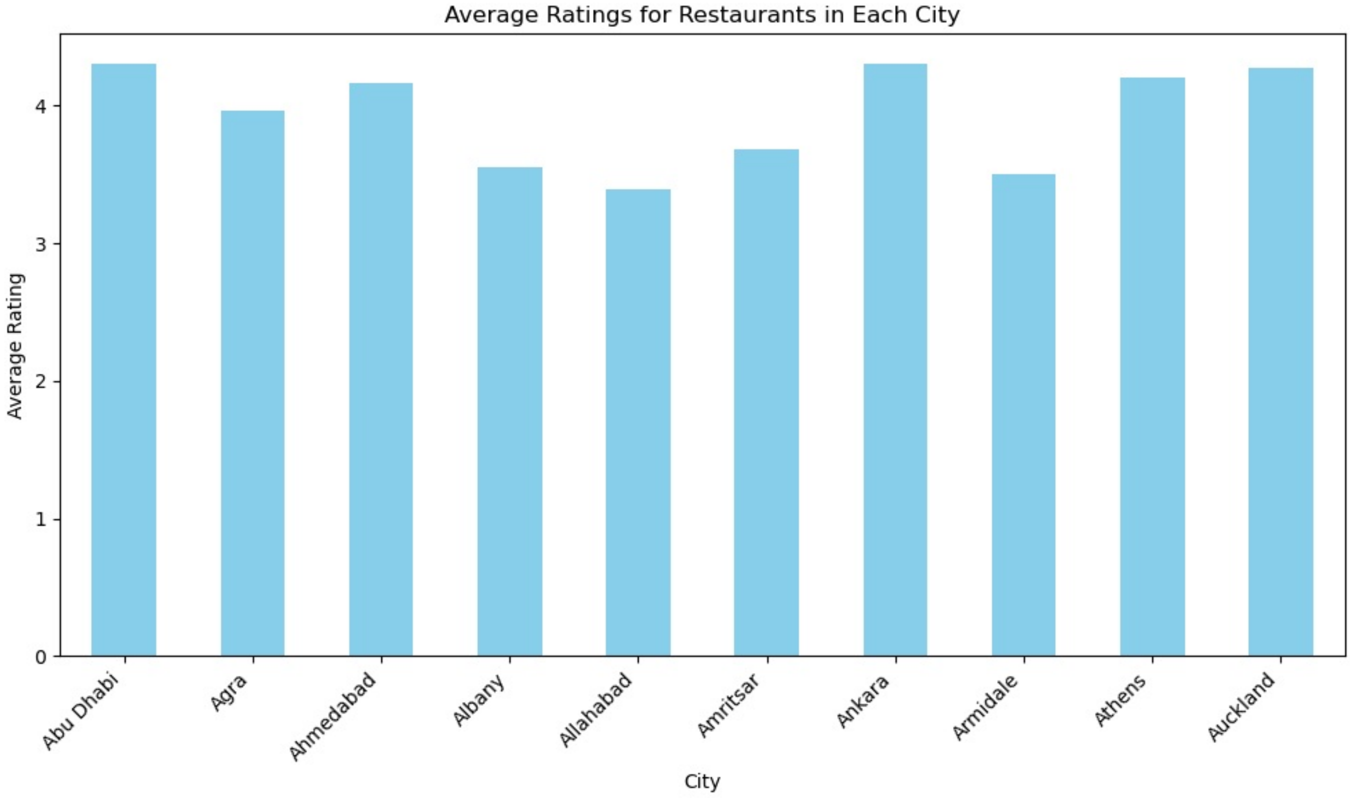
In [50]:

```python
# Group the data by city and calculate the average rating for restaurants in each city
city_avg_rating = df.groupby('City')['Aggregate rating'].mean()

print("Average ratings for restaurants in each city:")
print(city_avg_rating)
```

```
Average ratings for restaurants in each city:
City
Abu Dhabi          4.300000
Agra               3.965000
Ahmedabad          4.161905
Albany             3.555000
Allahabad          3.395000
                     ...
Weirton            3.900000
Wellington City    4.250000
Winchester Bay     3.200000
Yorkton            3.300000
ï¿½ï¿½stanbul      4.292857
Name: Aggregate rating, Length: 141, dtype: float64
```

In [53]:

```python
city_avg_rating = df.groupby('City')['Aggregate rating'].mean().head(10)

# Plot the data
plt.figure(figsize=(10, 6))
city_avg_rating.plot(kind='bar', color='skyblue')
plt.title('Average Ratings for Restaurants in Each City')
plt.xlabel('City')
plt.ylabel('Average Rating')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```



DETERMINE THE CITY WITH THE HIGHEST AVERAGE RATING.

In [54]:

```
df.columns
```

Out[54]:

```
Index(['ï»¿Restaurant ID', 'Restaurant Name', 'Country Code', 'City',
       'Address', 'Locality', 'Locality Verbose', 'Longitude', 'Latitude',
       'Cuisines', 'Average Cost for two', 'Currency', 'Has Table booking',
       'Has Online delivery', 'Is delivering now', 'Switch to order menu',
       'Price range', 'Aggregate rating', 'Rating color', 'Rating text',
       'Votes'],
      dtype='object')
```

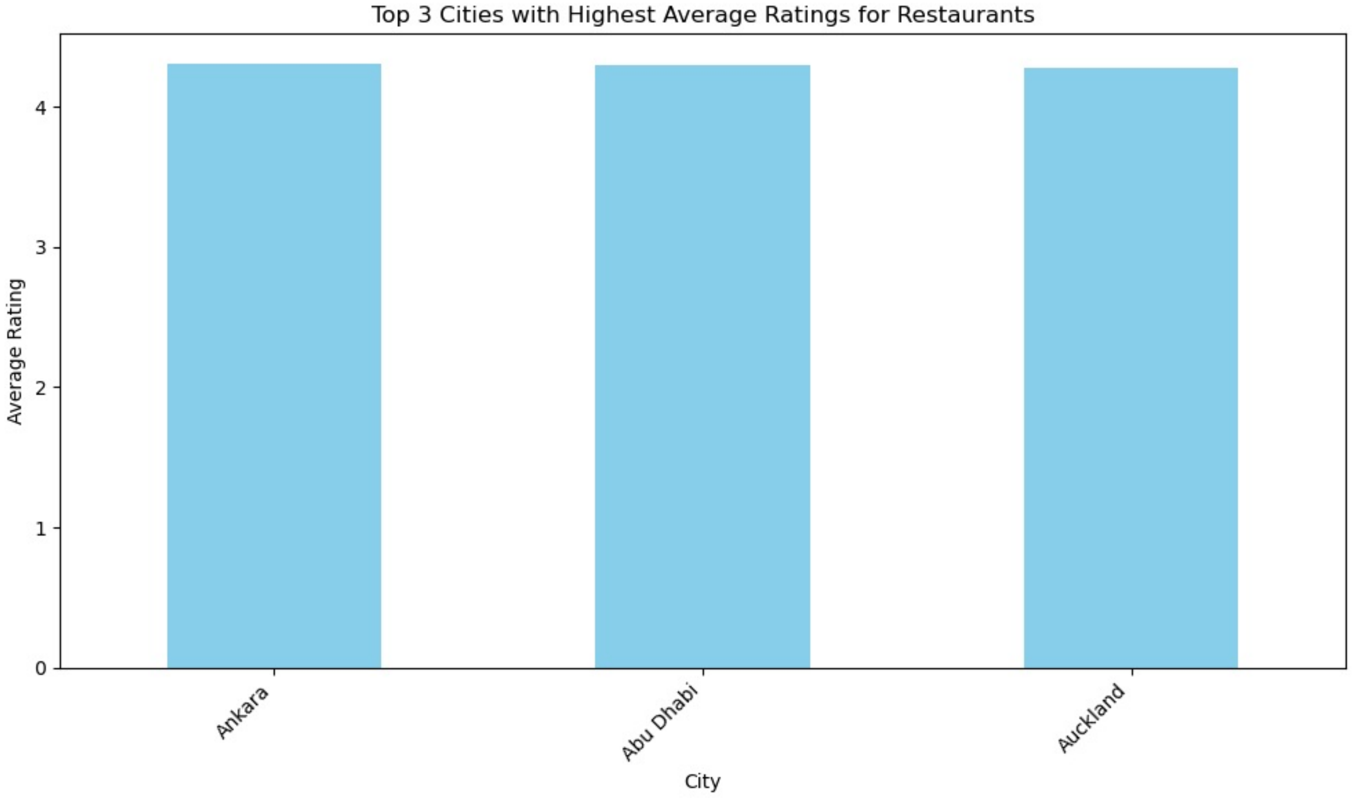
In [55]:

```
# Group the data by city and calculate the average rating for restaurants in each city
city_avg_rating = df.groupby('City')['Aggregate rating'].mean()

# Determine the city with the highest average rating
highest_avg_rating_city = city_avg_rating.idxmax()
highest_avg_rating = city_avg_rating.max()

print(f"The city with the highest average rating is {highest_avg_rating_city} with an average rating of {highes
```

```
The city with the highest average rating is Inner City with an average rating of 4.90.
```

In [58]:

```
# Sort the cities by average rating in descending order and select the top 3
top_3_cities = city_avg_rating.sort_values(ascending=False).head(3)

# Plot the data for top 3 cities
plt.figure(figsize=(10, 6))
top_3_cities.plot(kind='bar', color='skyblue')
plt.title('Top 3 Cities with Highest Average Ratings for Restaurants')
plt.xlabel('City')
plt.ylabel('Average Rating')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```



THANKYOU!

CONNECT WITH ME:

LinkedIn: https://www.linkedin.com/in/harshita-sharma-b68154220/

GitHub: https://github.com/DATAPREDICTS

Instagram: https://www.instagram.com/datapredicts?utm_source=qr&igsh=czVzc2k5c3oxOWQ4

YouTube: https://youtube.com/@Datapredicts?si=eDKAqVciVxg23zab

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js