Importing the libraries

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

DataFrame Creation

In [2]:
```
df = pd.read_csv("Zomato-data-.csv")
```

Printinng the first 5 rows of the dataframe

In [3]:
```
df.head(5)
```

Out[3]:

| | name | online_order | book_table | rate | votes | approx_cost(for two people) | listed_in(type) |
|---|---|---|---|---|---|---|---|
| **0** | Jalsa | Yes | Yes | 4.1/5 | 775 | 800 | Buffet |
| **1** | Spice Elephant | Yes | No | 4.1/5 | 787 | 800 | Buffet |
| **2** | San Churro Cafe | Yes | No | 3.8/5 | 918 | 800 | Buffet |
| **3** | Addhuri Udupi Bhojana | No | No | 3.7/5 | 88 | 300 | Buffet |
| **4** | Grand Village | No | No | 3.8/5 | 166 | 600 | Buffet |

Fixing the Rate Column i.e. removing the denominator by converting ratings into float

In [4]:
```
def handleRate(value):
    value=str(value).split('/')
    value=value[0]
    return float(value)

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

```
                   name online_order book_table  rate  votes  \
0                 Jalsa          Yes        Yes   4.1    775
1        Spice Elephant          Yes         No   4.1    787
2       San Churro Cafe          Yes         No   3.8    918
3  Addhuri Udupi Bhojana           No         No   3.7     88
4          Grand Village           No         No   3.8    166

   approx_cost(for two people) listed_in(type)
0                          800          Buffet
1                          800          Buffet
2                          800          Buffet
3                          300          Buffet
4                          600          Buffet
```

Summarizing the Data Frame

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 148 entries, 0 to 147
Data columns (total 7 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   name                         148 non-null    object
 1   online_order                 148 non-null    object
 2   book_table                   148 non-null    object
 3   rate                         148 non-null    float64
 4   votes                        148 non-null    int64
 5   approx_cost(for two people)  148 non-null    int64
 6   listed_in(type)              148 non-null    object
dtypes: float64(1), int64(2), object(4)
memory usage: 8.2+ KB
```

Figuring out NULL Values (if any)

In [18]:
```
null_values = df.isnull()
null_counts = null_values.sum()
print("Total number of NULL Values in each column is:{} \n",format(null_counts))
```

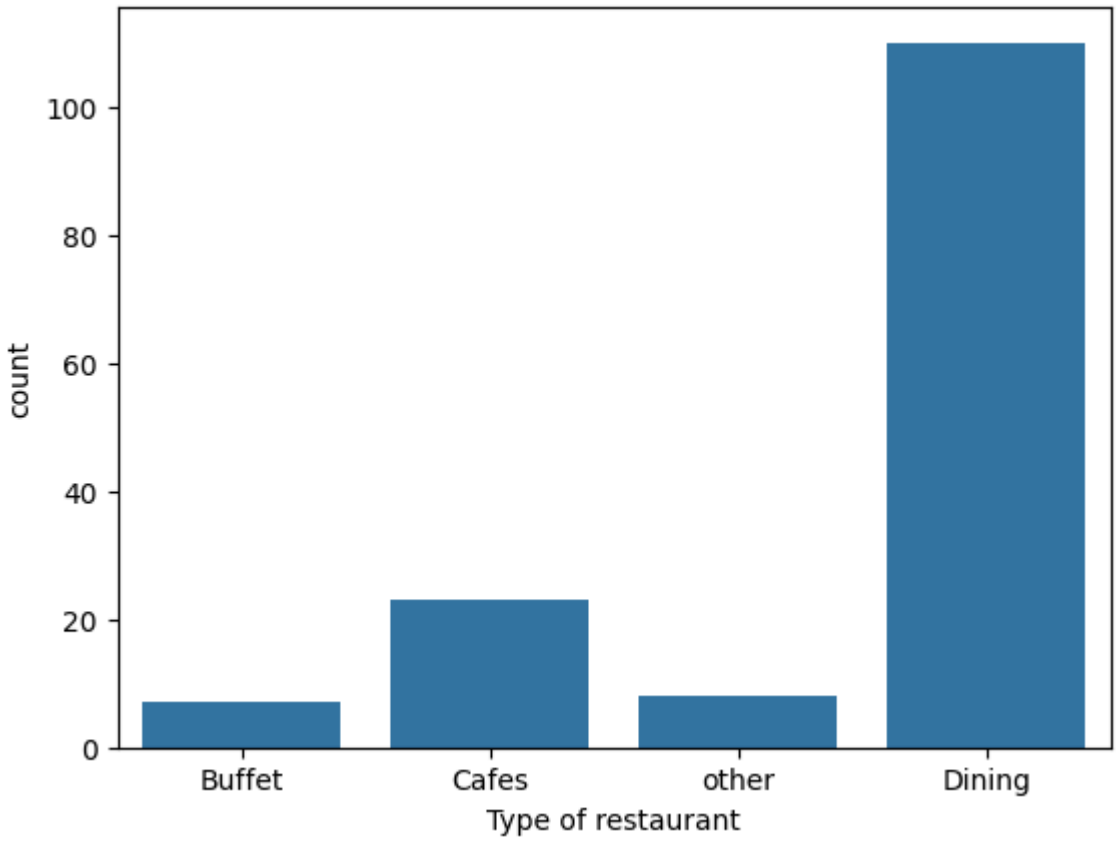
```
Total number of NULL Values in each column is:{}
 name                          0
online_order                   0
book_table                     0
rate                           0
votes                          0
approx_cost(for two people)    0
listed_in(type)                0
dtype: int64
```

Exploring and visualizing the dataset

Plotting the listed_in (type) column of the dataset

In [20]:
```
sns.countplot(x=df['listed_in(type)'])
plt.xlabel("Type of restaurant")
```

Out[20]:
```
Text(0.5, 0, 'Type of restaurant')
```

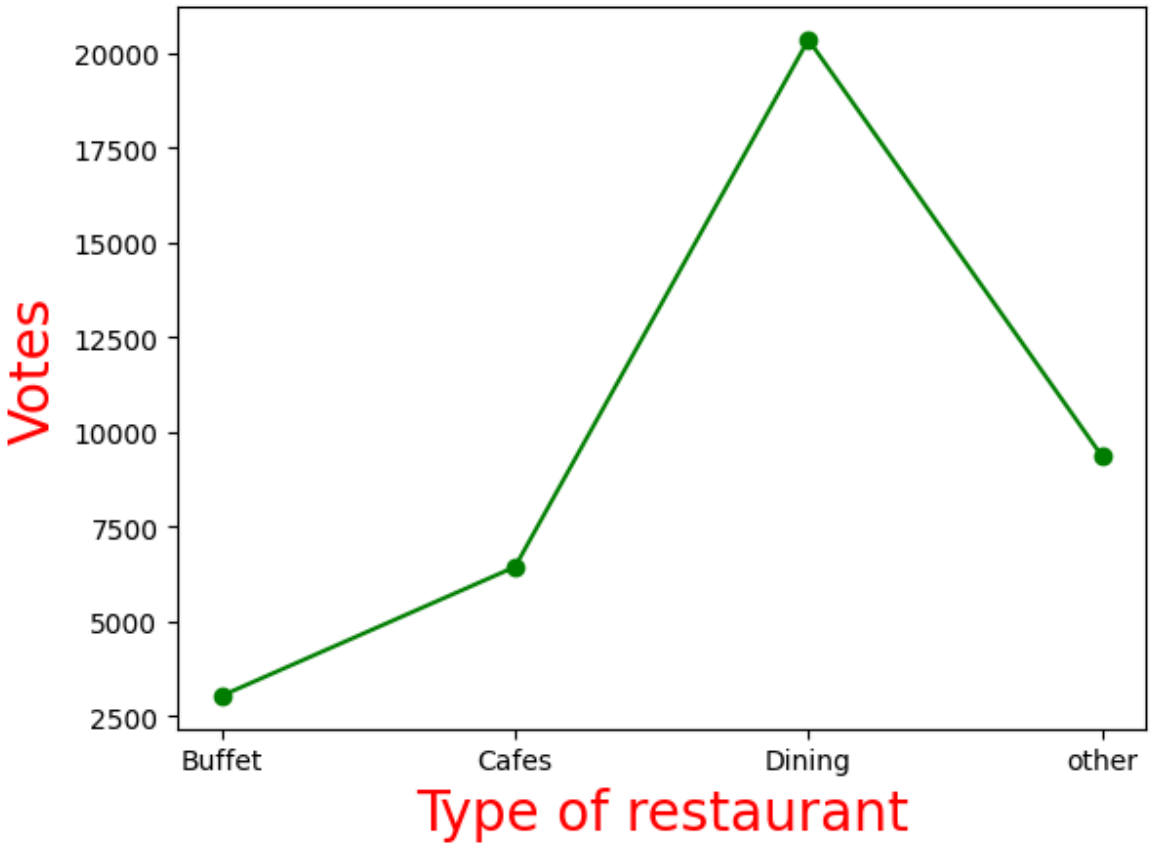



By the generated graph we can conclude the majority of the restaurants fall under Dining Category

Exploring the votes to gather more visuals

In [21]:
```
grouped_data = df.groupby('listed_in(type)')['votes'].sum()
result = pd.DataFrame({'votes': grouped_data})
plt.plot(result, c='green', marker='o')
plt.xlabel('Type of restaurant', c='red', size=20)
plt.ylabel('Votes', c='red', size=20)
```

Out[21]:
```
Text(0, 0.5, 'Votes')
```




Through votes also we conclude Dining restaurants are preferred by a larger number of individuals.

Determing the restaurant with maximum votes

In [22]:
```
max_votes = df['votes'].max()
restaurant_with_max_votes = df.loc[df['votes'] == max_votes, 'name']

print('Restaurant(s) with the maximum votes:')
print(restaurant_with_max_votes)
```

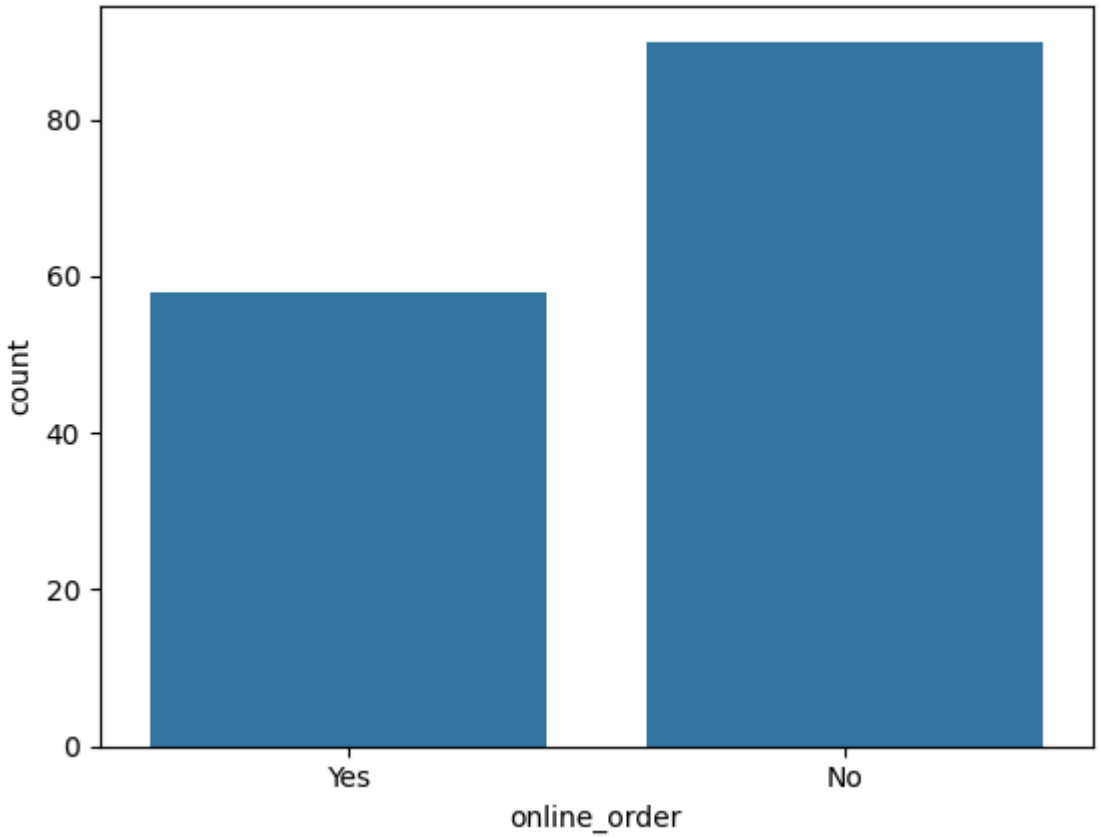
```
Restaurant(s) with the maximum votes:
38    Empire Restaurant
Name: name, dtype: object
```

Thus, the Restaurant with maximum number of votes is "Empire Restaurant"

Now, Exploring the online_order column

In [23]:
```
sns.countplot(x=df['online_order'])
```

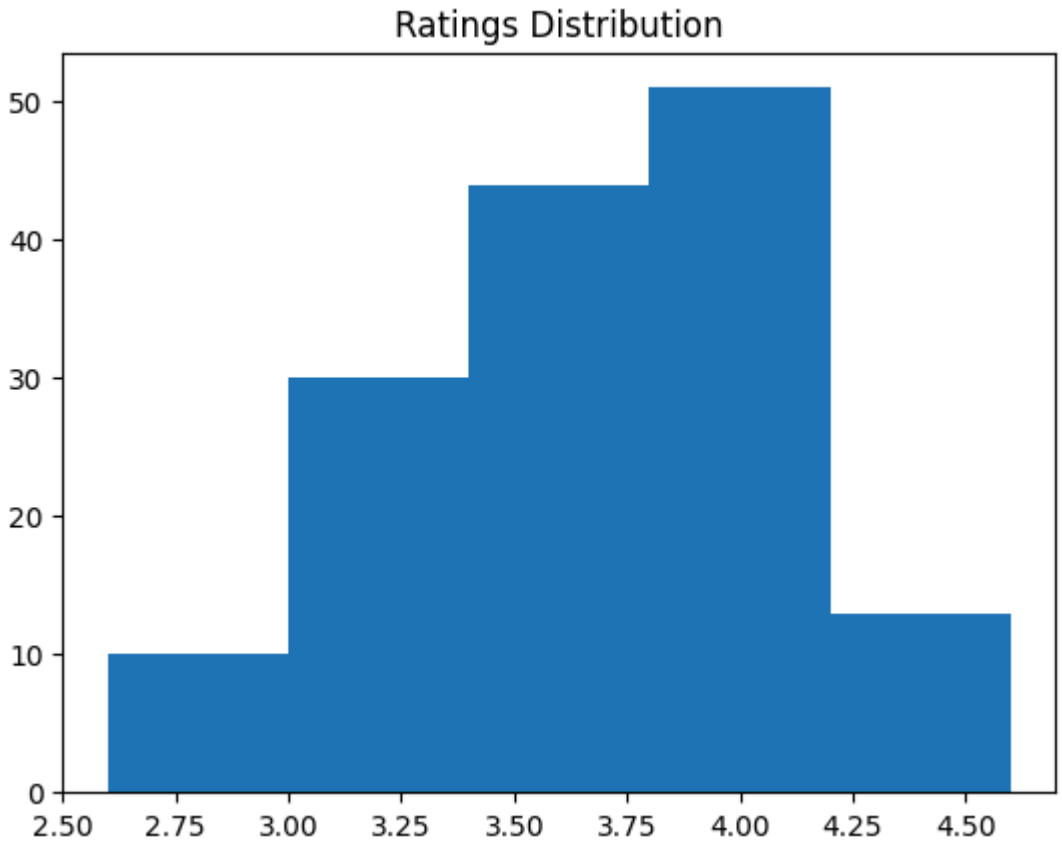
Out[23]:
```
<Axes: xlabel='online_order', ylabel='count'>
```




The graph informs us that a majority of the restaurants do not accept online orders.

Exploring the rate column

In [24]:
```
plt.hist(df['rate'],bins=5)
plt.title('Ratings Distribution')
plt.show()
```

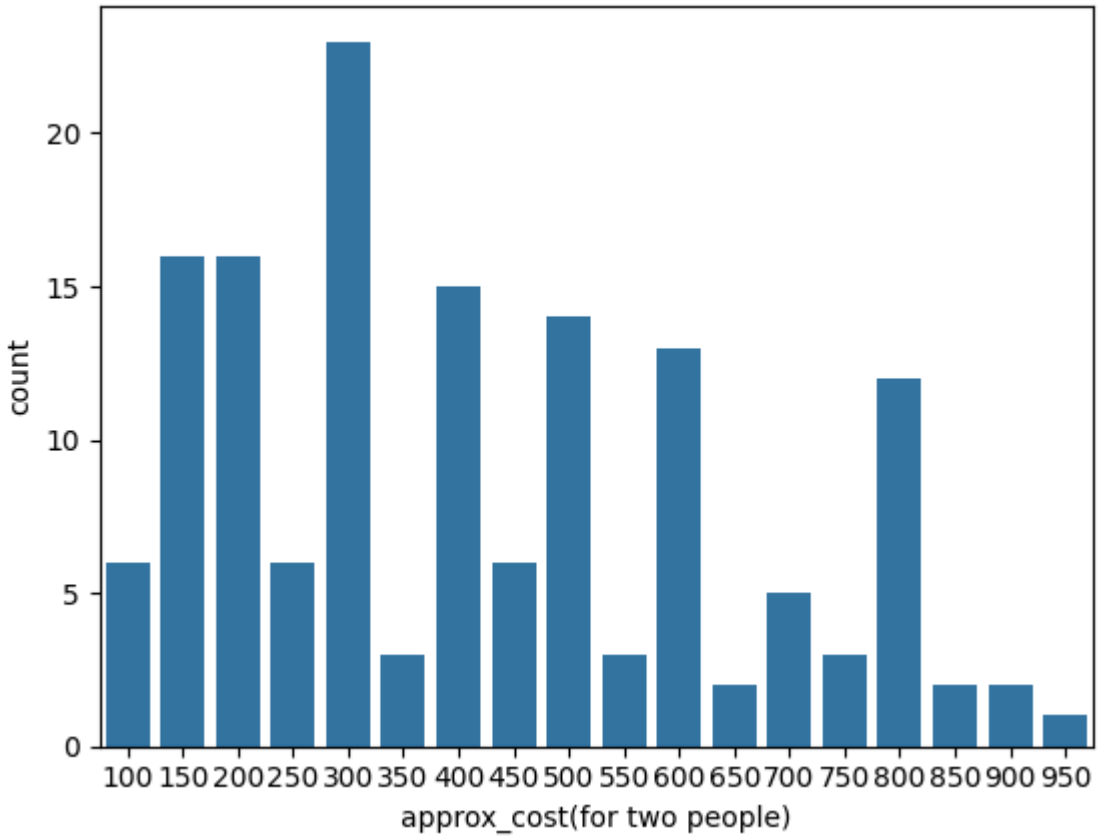



Conclusion : The majority of restaurants received ratings ranging from 3.5 to 4.

Exploring the approximate cost for two people column approx_cost(for two people)

In [25]:
```
couple_data=df['approx_cost(for two people)']
sns.countplot(x=couple_data)
```

Out[25]:
```
<Axes: xlabel='approx_cost(for two people)', ylabel='count'>
```

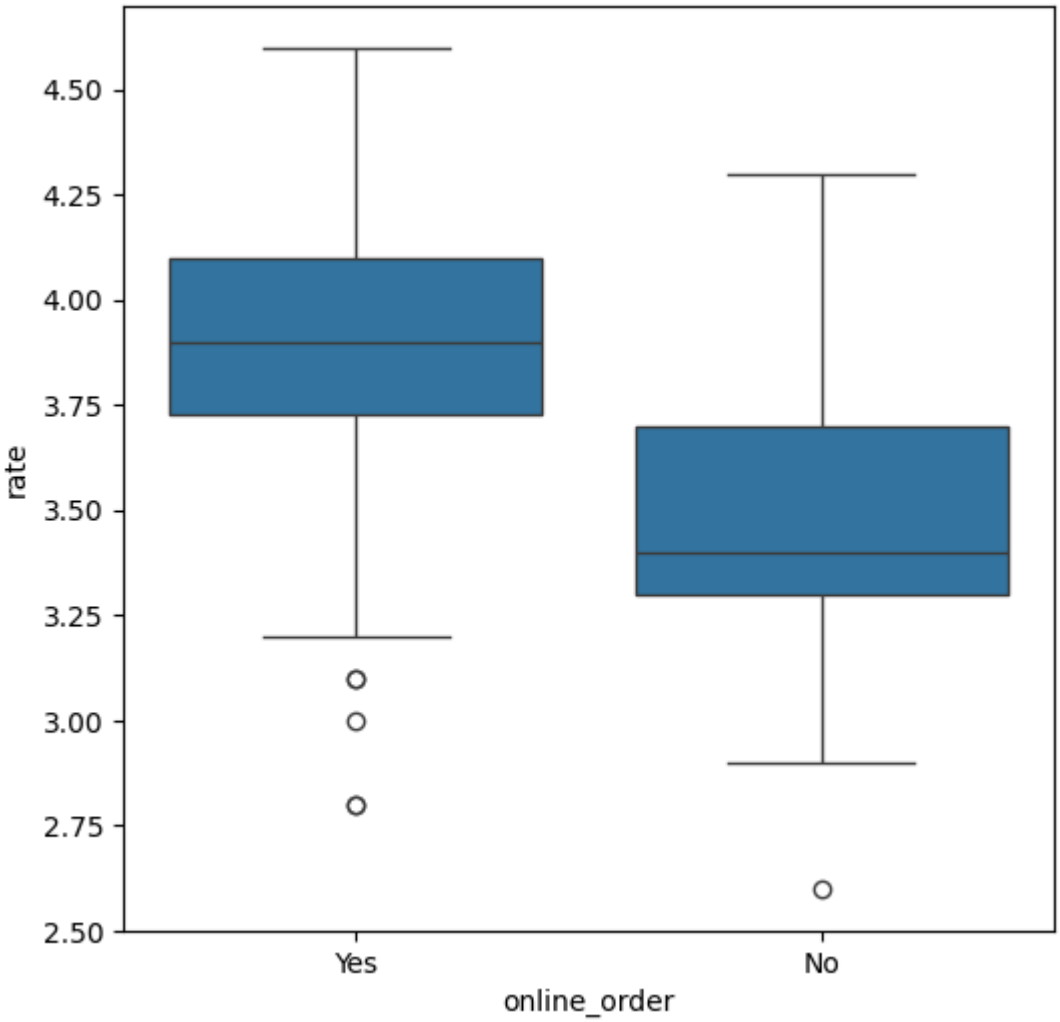



The approximate cost preffered is found out to be 300 rupees

Comparing the ratings between online orders and offline orders

In [26]:
```
plt.figure(figsize = (6,6))
sns.boxplot(x = 'online_order', y = 'rate', data = df)
```

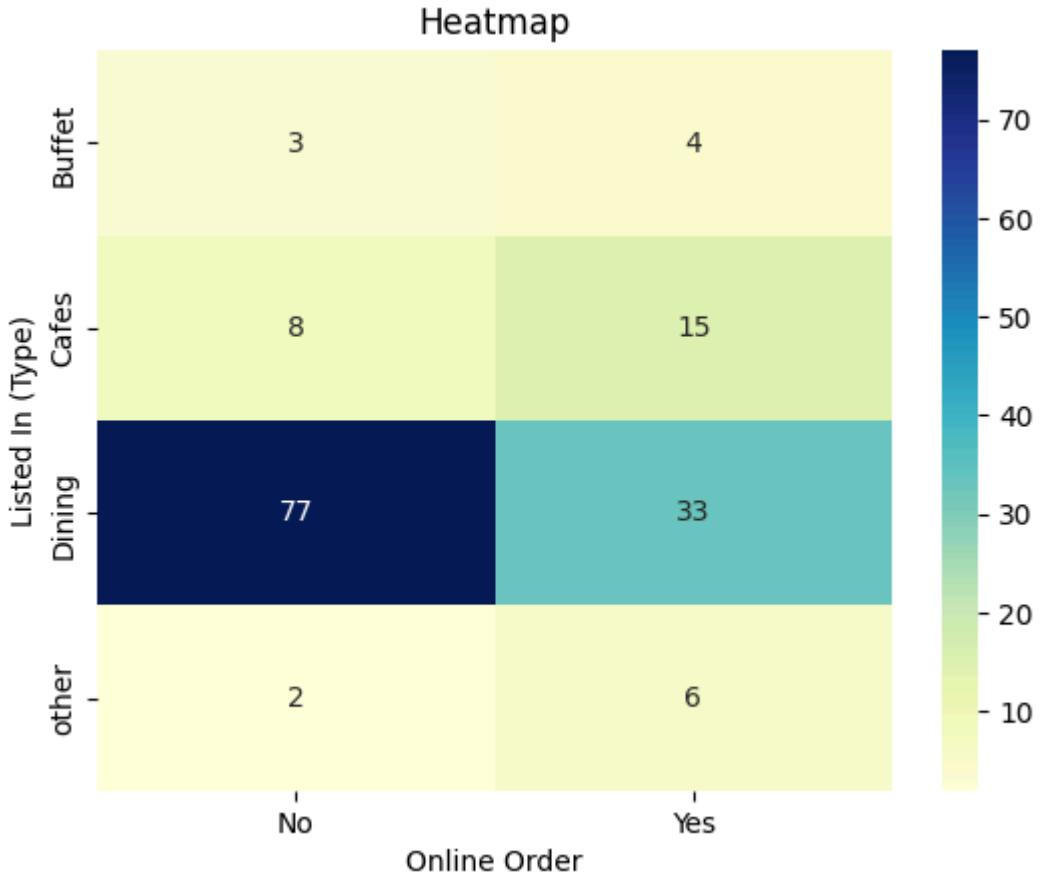
Out[26]:
```
<Axes: xlabel='online_order', ylabel='rate'>
```




CONCLUSION: Offline orders received lower ratings in comparison to online orders, which obtained excellent ratings.

Generating a HeatMap to represent our findings from the dataset

In [27]:
```
pivot_table = df.pivot_table(index='listed_in(type)', columns='online_order', ag
sns.heatmap(pivot_table, annot=True, cmap='YlGnBu', fmt='d')
plt.title('Heatmap')
plt.xlabel('Online Order')
plt.ylabel('Listed In (Type)')
plt.show()
```




CONCLUSION: Dining restaurants primarily accept offline orders, whereas cafes primarily receive online orders. This suggests that clients prefer to place orders in person at restaurants, but prefer online ordering at cafes.