Description of columns from dataset documentation

Attributes:

- id :- int64
- Date :- int64
- number of bedrooms :- int64
- number of bathrooms :- float64
- living area :- int64
- lot area :- int64
- number of floors :- float64
- waterfront present :- int64
- number of views :- int64
- condition of the house :- int64
- grade of the house :- int64
- Area of the house(excluding basement) :- int64
- Area of the basement :- int64
- Built Year :- int64
- Renovation Year :- int64
- Postal Code :- int64
- Lattitude :- float64
- Longitude :- float64
- living_area_renov :- int64
- lot_area_renov :- int64
- Number of schools nearby :- int64
- Distance from the airport :- int64
- Price :- int64

There are four float values and 19 int values are present in the dataset.

reference of the dataset:-

https://www.kaggle.com/datasets/mohamedafsal007/house-price-dataset-of-india (https://www.kaggle.com/datasets/mohamedafsal007/house-price-dataset-of-india)

Import Libraries

In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

Reading and Understanding the Dataset.

Import Dataset.

In [2]:

```
data=pd.read_csv('House Price India.csv')
df=pd.DataFrame(data)
```

In [3]:

```
df.head()
```

Out[3]:

	id	Date	number of bedrooms	number of bathrooms	living area	lot area	number of floors	waterfront present	number of views	conditi of t hou
0	6762810145	42491	5	2.50	3650	9050	2.0	0	4	
1	6762810635	42491	4	2.50	2920	4000	1.5	0	0	
2	6762810998	42491	5	2.75	2910	9480	1.5	0	0	
3	6762812605	42491	4	2.50	3310	42998	2.0	0	0	
4	6762812919	42491	3	2.00	2710	4500	1.5	0	0	

5 rows × 23 columns

In [4]:

```
df.shape
```

Out[4]:

```
(14620, 23)
```

From the above we can easily see that :-

- The above dataset have 14620 rows and 18 columns in it.

In [5]:

```
df.tail(3)
```

Out[5]:

	id	Date	number of bedrooms	number of bathrooms	living area	lot area	number of floors	waterfront present	number of views	cor
14617	6762830618	42734	2	1.0	1070	6120	1.0	0	0	
14618	6762830709	42734	4	1.0	1030	6621	1.0	0	0	
14619	6762831463	42734	3	1.0	900	4770	1.0	0	0	

3 rows × 23 columns

Discover Data.

In [6]:

```
rows,col = df.shape
print("Dimensions of Dataset:{}". format (df.shape))
print('Rows:',rows, '\n Columns:',col)
```

```
Dimensions of Dataset:(14620, 23)
Rows: 14620
 Columns: 23
```

```
In [7]: # Statistical details
        df.iloc[:, :-1].describe().T.sort_values(by='std' , ascending = False)\
                                    .style.background_gradient(cmap='PuBu')\
                                    .bar(subset=["max"], color='#F8766D')\
                                    .bar(subset=["mean",], color='#00BFC4')
```

Out[7]:

	count	mean	std	min	
lot area	14620.000000	15093.281122	37919.621304	520.000000	501
lot_area_renov	14620.000000	12753.500068	26058.414467	651.000000	509
id	14620.000000	6762820830.525650	6237.574799	6762810020.000000	676281540
living area	14620.000000	2098.262996	928.275721	370.000000	144
Area of the house(excluding basement)	14620.000000	1801.783926	833.809963	370.000000	120
living_area_renov	14620.000000	1996.702257	691.093366	460.000000	149
Area of the basement	14620.000000	296.479070	448.551409	0.000000	
Renovation Year	14620.000000	90.924008	416.216661	0.000000	
Date	14620.000000	42604.538646	67.347991	42491.000000	4254
Built Year	14620.000000	1970.926402	29.493625	1900.000000	195
Postal Code	14620.000000	122033.062244	19.082418	122003.000000	12201
Distance from the airport	14620.000000	64.950958	8.936008	50.000000	5
grade of the house	14620.000000	7.682421	1.175033	4.000000	
number of bedrooms	14620.000000	3.379343	0.938719	1.000000	
Number of schools nearby	14620.000000	2.012244	0.817284	1.000000	
number of bathrooms	14620.000000	2.129583	0.769934	0.500000	
number of views	14620.000000	0.233105	0.766259	0.000000	
condition of the house	14620.000000	3.430506	0.664151	1.000000	
number of floors	14620.000000	1.502360	0.540239	1.000000	
Longitude	14620.000000	-114.404007	0.141326	-114.709000	-11
Lattitude	14620.000000	52.792848	0.137522	52.385900	5
waterfront present	14620.000000	0.007661	0.087193	0.000000	

The overhead table displays:

- Each feature contains 14620 data recorded.

- There is a negative value in the Longitude feature.

```
In [8]: # Number of uniqe elements in each columns
        unique = df.nunique()
        unique.to_frame().T
```

Out[8]:

	id	Date	number of bedrooms	number of bathrooms	living area	lot area	number of floors	waterfront present	number of views	condition of the house	...
0	14620	241	12	29	865	7451	6	2	5	5	...

1 rows × 23 columns

```
In [9]: # Information about the dataframe.
        df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14620 entries, 0 to 14619
Data columns (total 23 columns):
 #   Column                                 Non-Null Count  Dtype  
---  ------                                 --------------  -----  
 0   id                                     14620 non-null  int64  
 1   Date                                   14620 non-null  int64  
 2   number of bedrooms                     14620 non-null  int64  
 3   number of bathrooms                    14620 non-null  float64
 4   living area                            14620 non-null  int64  
 5   lot area                               14620 non-null  int64  
 6   number of floors                       14620 non-null  float64
 7   waterfront present                     14620 non-null  int64  
 8   number of views                        14620 non-null  int64  
 9   condition of the house                 14620 non-null  int64  
 10  grade of the house                     14620 non-null  int64  
 11  Area of the house(excluding basement)  14620 non-null  int64  
 12  Area of the basement                   14620 non-null  int64  
 13  Built Year                             14620 non-null  int64  
 14  Renovation Year                        14620 non-null  int64  
 15  Postal Code                            14620 non-null  int64  
 16  Lattitude                              14620 non-null  float64
 17  Longitude                              14620 non-null  float64
 18  living_area_renov                      14620 non-null  int64  
 19  lot_area_renov                         14620 non-null  int64  
 20  Number of schools nearby               14620 non-null  int64  
 21  Distance from the airport              14620 non-null  int64  
 22  Price                                  14620 non-null  int64  
dtypes: float64(4), int64(19)
memory usage: 2.6 MB
```

In [10]:
```python
# define the datatypes of the data.
df.dtypes
```

Out[10]:
```
id                                           int64
Date                                         int64
number of bedrooms                           int64
number of bathrooms                        float64
living area                                  int64
lot area                                     int64
number of floors                           float64
waterfront present                           int64
number of views                              int64
condition of the house                       int64
grade of the house                           int64
Area of the house(excluding basement)        int64
Area of the basement                         int64
Built Year                                   int64
Renovation Year                              int64
Postal Code                                  int64
Lattitude                                  float64
Longitude                                  float64
living_area_renov                            int64
lot_area_renov                               int64
Number of schools nearby                     int64
Distance from the airport                    int64
Price                                        int64
dtype: object
```

The information of the dataset shows:

- Lattitude,Longitude ,number of floors and number of bathrooms has an float type and the rest of the features have an int64 type.

Data Cleaning.

In [11]:

```
df.isnull().sum()
```

Out[11]:

```
id                                       0
Date                                     0
number of bedrooms                       0
number of bathrooms                      0
living area                              0
lot area                                 0
number of floors                         0
waterfront present                       0
number of views                          0
condition of the house                   0
grade of the house                       0
Area of the house(excluding basement)    0
Area of the basement                     0
Built Year                               0
Renovation Year                          0
Postal Code                              0
Lattitude                                0
Longitude                                0
living_area_renov                        0
lot_area_renov                           0
Number of schools nearby                 0
Distance from the airport                0
Price                                    0
dtype: int64
```

From above we can easily see that:-

- There is no null value present in dataset.

In [12]:

```
df.duplicated().sum()
```

Out[12]:

```
0
```

From above we can easily see that:-

- There is no duplicate value present in data set.

In [13]:
```
df.head(3)
```

Out[13]:

	id	Date	number of bedrooms	number of bathrooms	living area	lot area	number of floors	waterfront present	number of views	conditio of th hous
0	6762810145	42491	5	2.50	3650	9050	2.0	0	4	
1	6762810635	42491	4	2.50	2920	4000	1.5	0	0	
2	6762810998	42491	5	2.75	2910	9480	1.5	0	0	

3 rows × 23 columns

In [14]:
```
df.drop(['id','Date','Postal Code'],axis=1,inplace=True)
```

In [15]:
```
df.head(3)
```

Out[15]:

	number of bedrooms	number of bathrooms	living area	lot area	number of floors	waterfront present	number of views	condition of the house	grade of the house	Area house(exc bas
0	5	2.50	3650	9050	2.0	0	4	5	10	
1	4	2.50	2920	4000	1.5	0	0	5	8	
2	5	2.75	2910	9480	1.5	0	0	3	8	

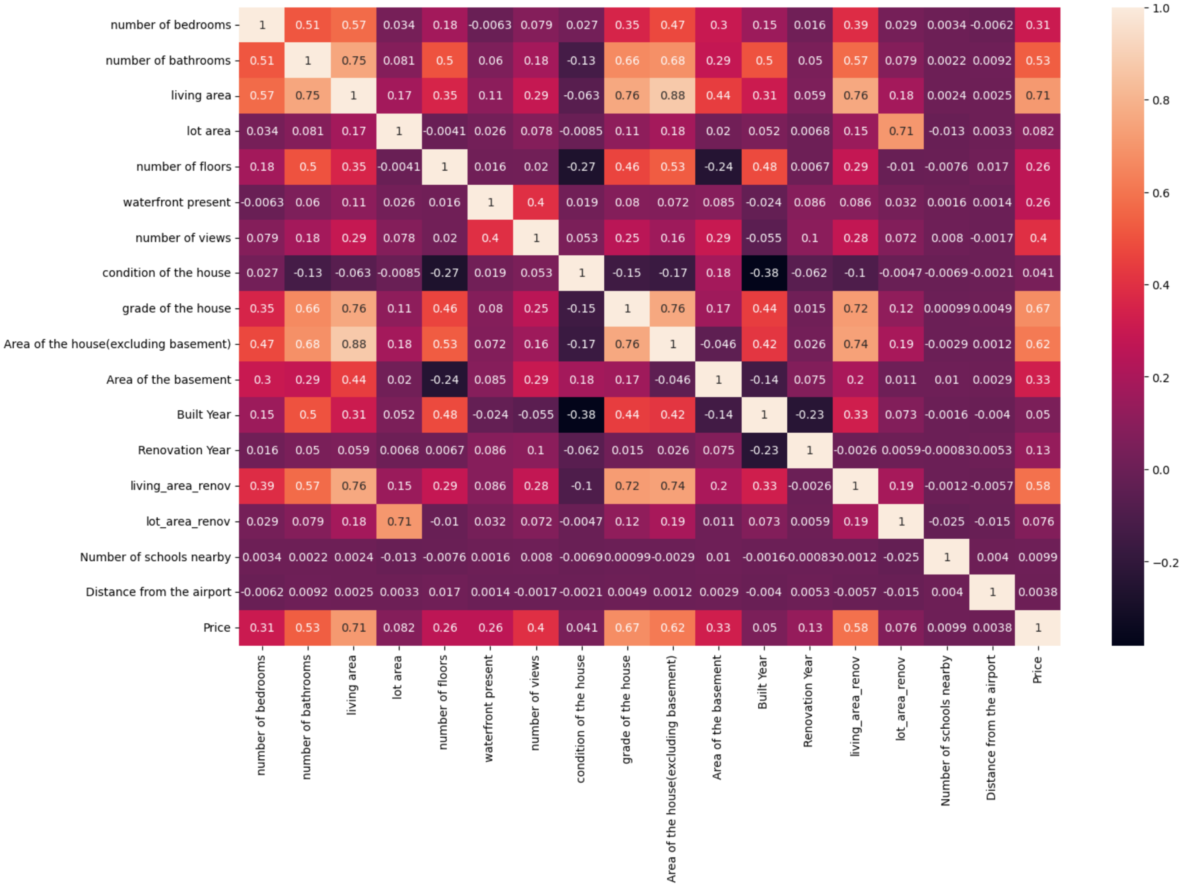
In [16]:
```
df.columns
```

Out[16]:
```
Index(['number of bedrooms', 'number of bathrooms', 'living area', 'lot are
a',
       'number of floors', 'waterfront present', 'number of views',
       'condition of the house', 'grade of the house',
       'Area of the house(excluding basement)', 'Area of the basement',
       'Built Year', 'Renovation Year', 'Lattitude', 'Longitude',
       'living_area_renov', 'lot_area_renov', 'Number of schools nearby',
       'Distance from the airport', 'Price'],
      dtype='object')
```

In [17]:
```
df.drop(['Lattitude','Longitude'],axis=1,inplace=True)
```

EDA Part:-

```
In [18]: plt.figure(figsize=(16,10))
         sns.heatmap(df.corr(),annot=True)
         plt.show()
```

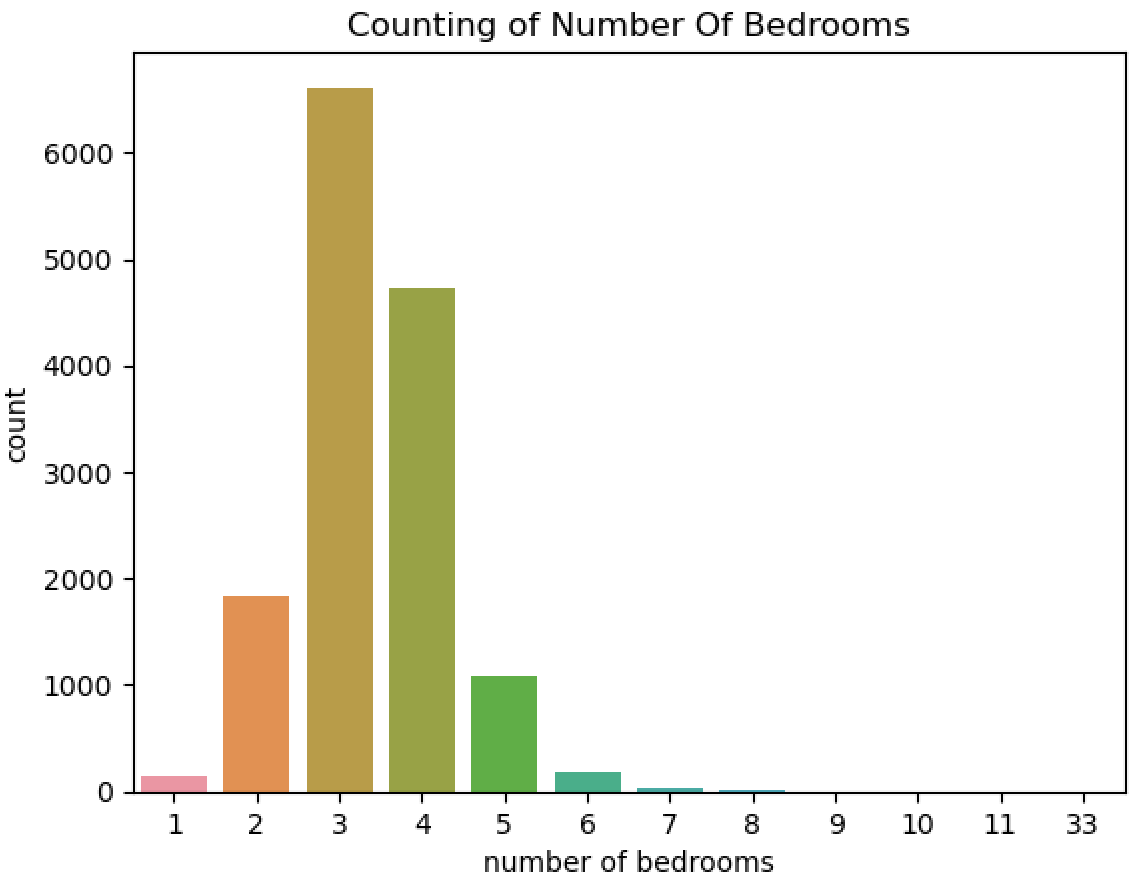


```
In [19]: df.columns

Out[19]: Index(['number of bedrooms', 'number of bathrooms', 'living area', 'lot are
a',
                'number of floors', 'waterfront present', 'number of views',
                'condition of the house', 'grade of the house',
                'Area of the house(excluding basement)', 'Area of the basement',
                'Built Year', 'Renovation Year', 'living_area_renov', 'lot_area_reno
v',
                'Number of schools nearby', 'Distance from the airport', 'Price'],
               dtype='object')
```

In [20]:

```
sns.countplot(data=df,x='number of bedrooms')
plt.title('Counting of Number Of Bedrooms')
plt.show()
```

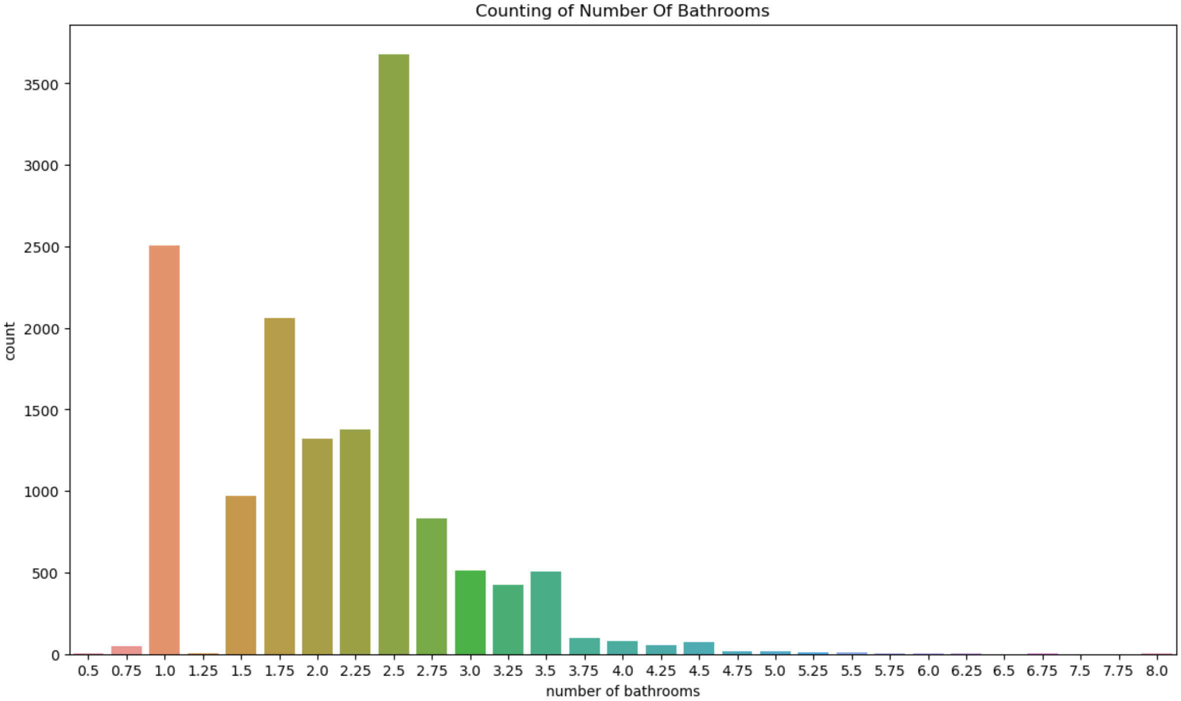


In the above Plot we can easily see that the

- In Maximum No. of houses there are 3 Bedrooms.
- The Houses where More than 7 bedrooms are less in count.

In [21]:

```
plt.figure(figsize=(14,8))
sns.countplot(data=df,x='number of bathrooms')
plt.title('Counting of Number Of Bathrooms')
plt.show()
```

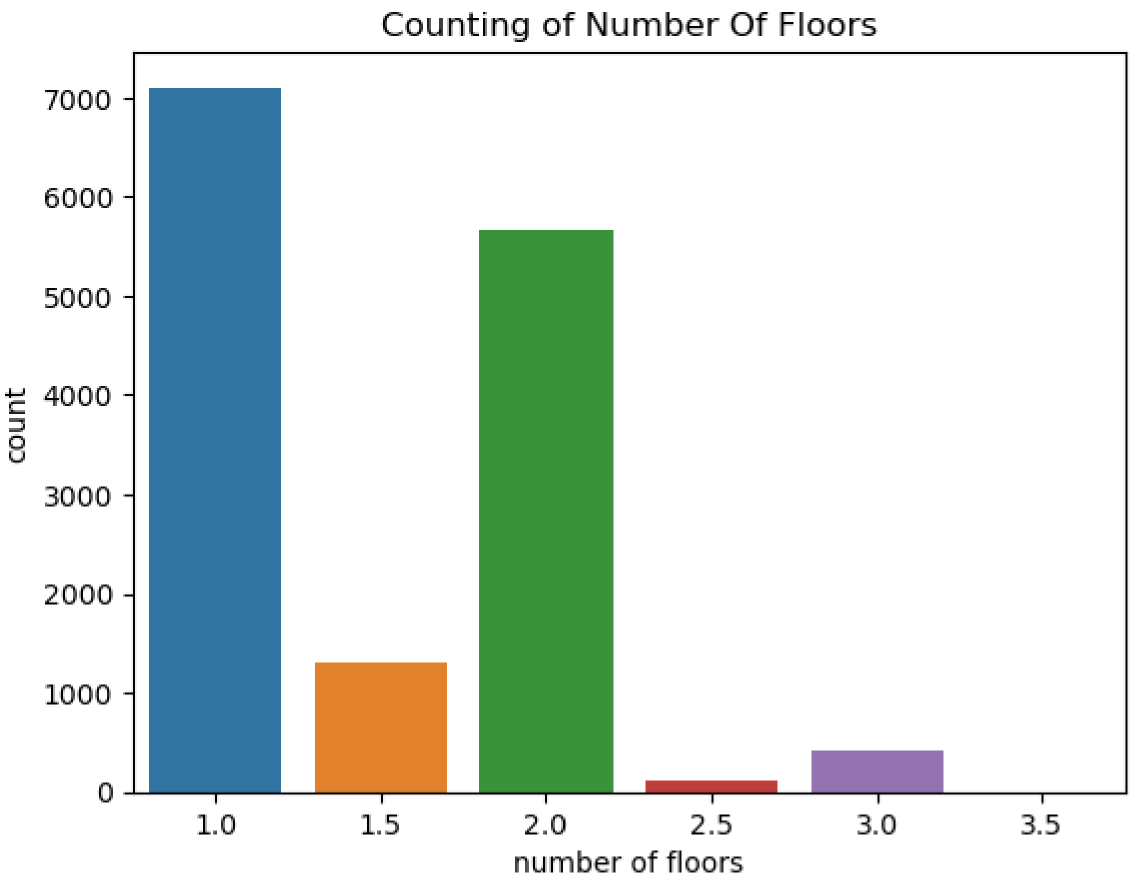


In the above Plot We can easily seen that

- There are many Houses haveing 2.5 bathrooms.
- and Houses haveing 1 bathroom lie on second Number.
- Houses with 0.5,0.75,1.25, and more than 4.75 bathrooms are very less in count.

In [22]:

```
sns.countplot(data=df,x='number of floors')
plt.title('Counting of Number Of Floors')
plt.show()
```

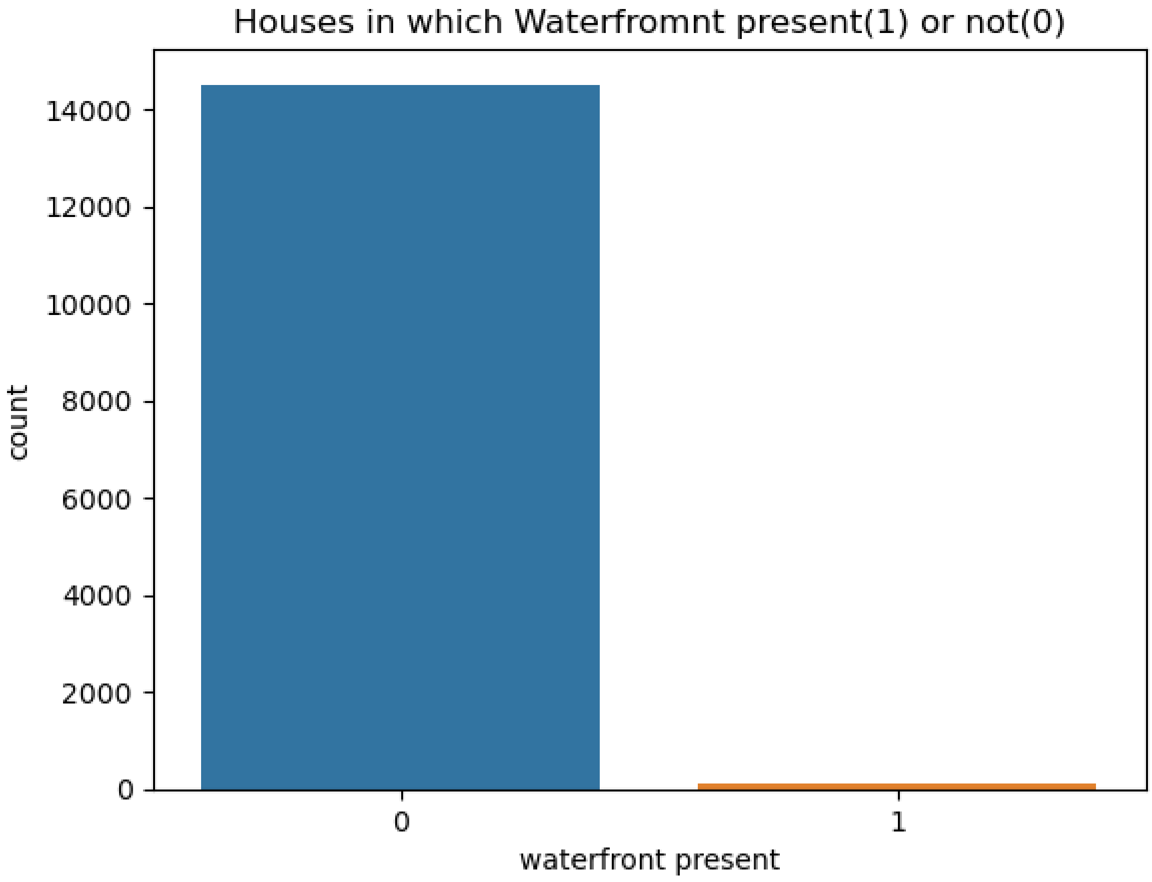


In the above Plot We can easily see that-

- There are Maximum Number Of Houses haveing 1 Floor.

In [23]:

```
sns.countplot(data=df,x='waterfront present')
plt.title('Houses in which Waterfromnt present(1) or not(0)')
plt.show()
```

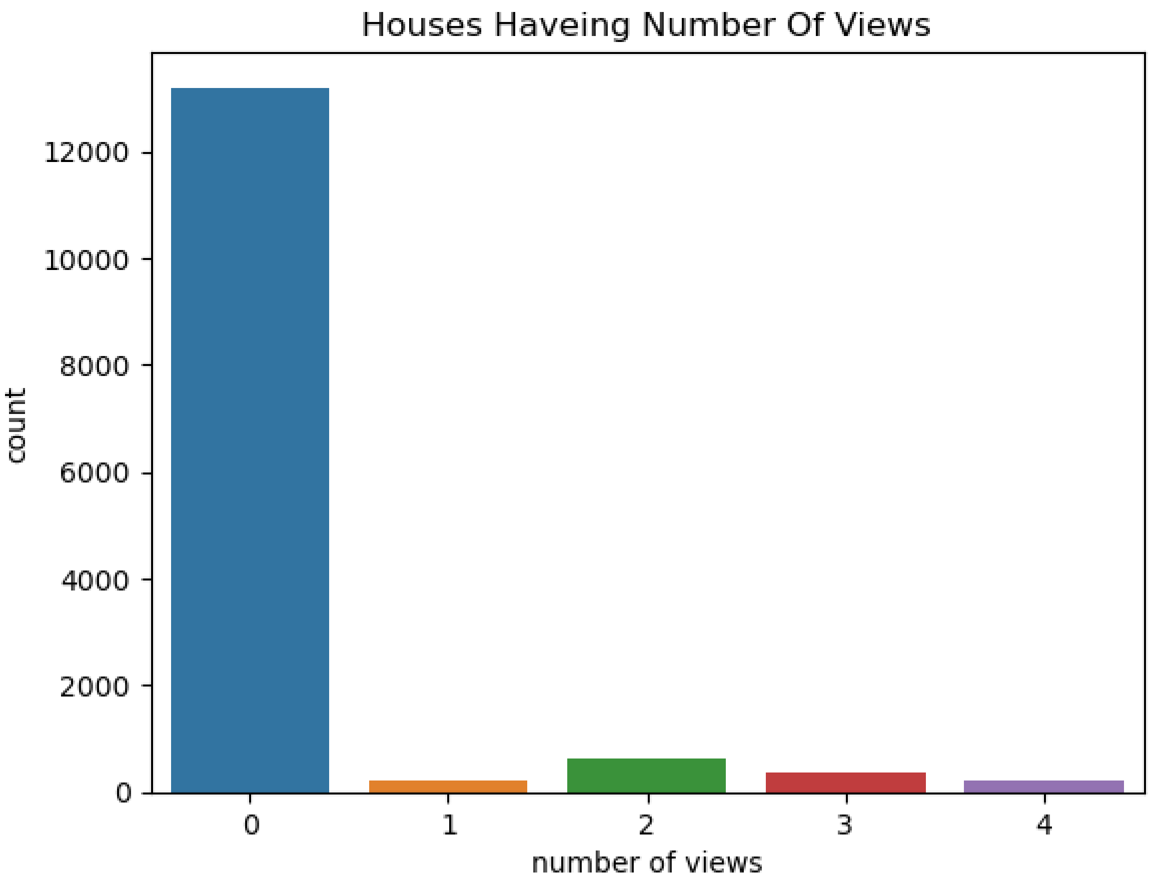


In the above plot we can easily see that

- There are more than 14,000 Houses haveing waterfront not present.
- there are less Houses In which waterfront are present.

In [24]:

```
sns.countplot(data=df,x='number of views')
plt.title('Houses Haveing Number Of Views')
plt.show()
```

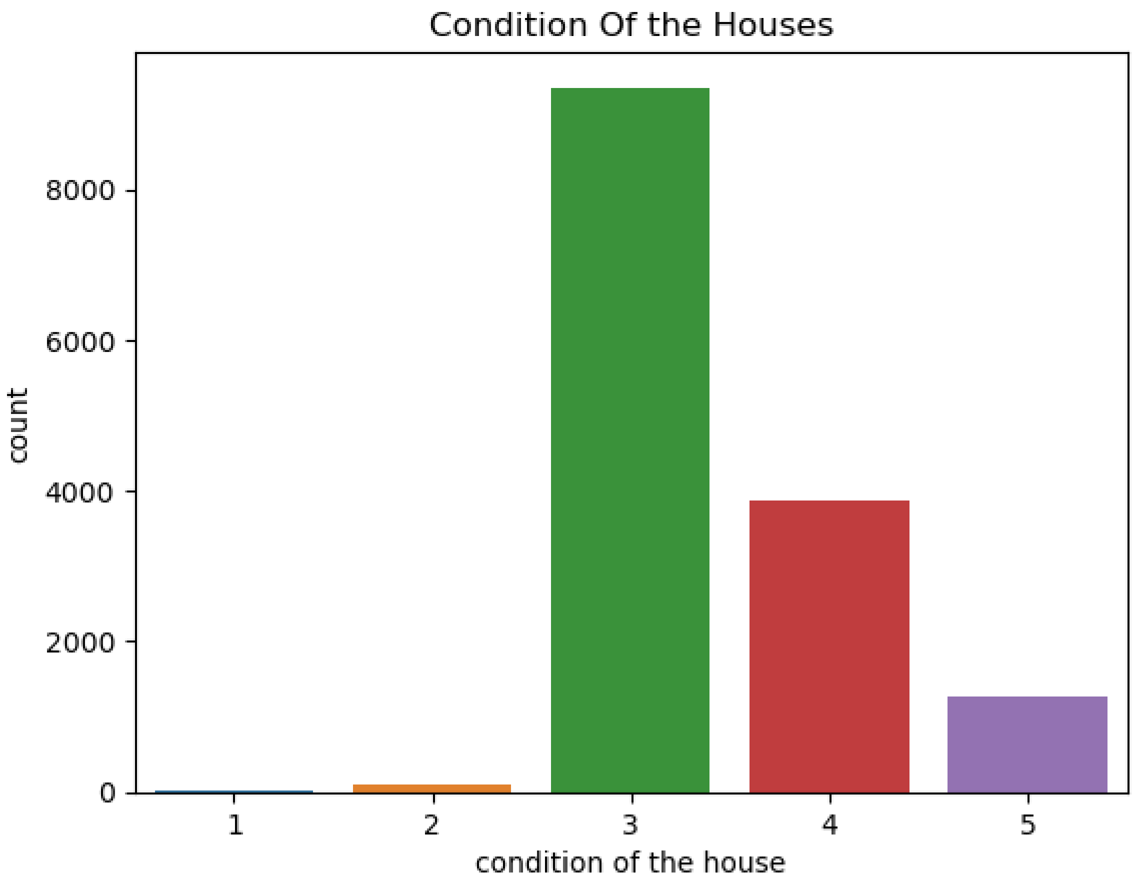


From the above plot we can easily seen that

- There are more than 12,000 houses haveing 0 views.
- Houses haveing 1,4 views are less in Nature.

In [25]:

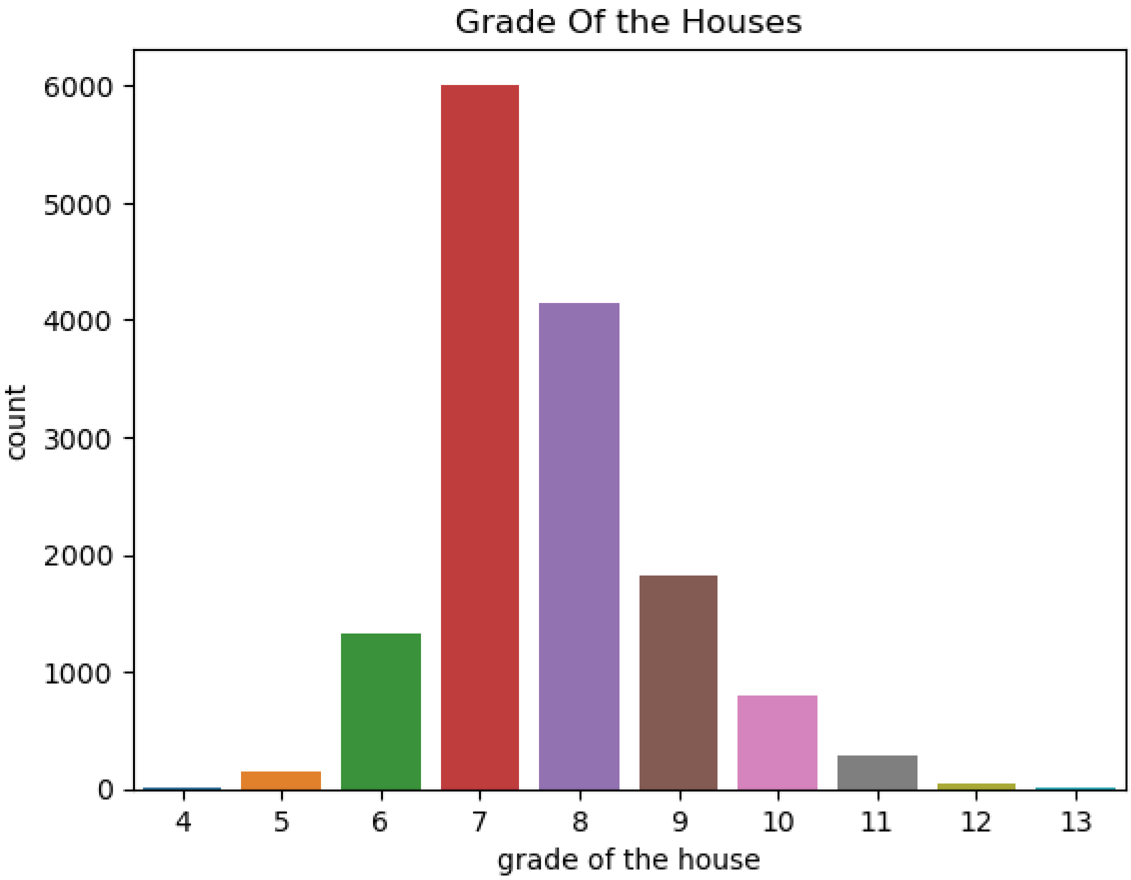
```
sns.countplot(data=df,x='condition of the house')
plt.title('Condition Of the Houses')
plt.show()
```



From the above plot we can easily seen that

- More than 8000 Houses haveing medium condition(3).
- There are less than 2000 Houses haveing excellent condition(5).

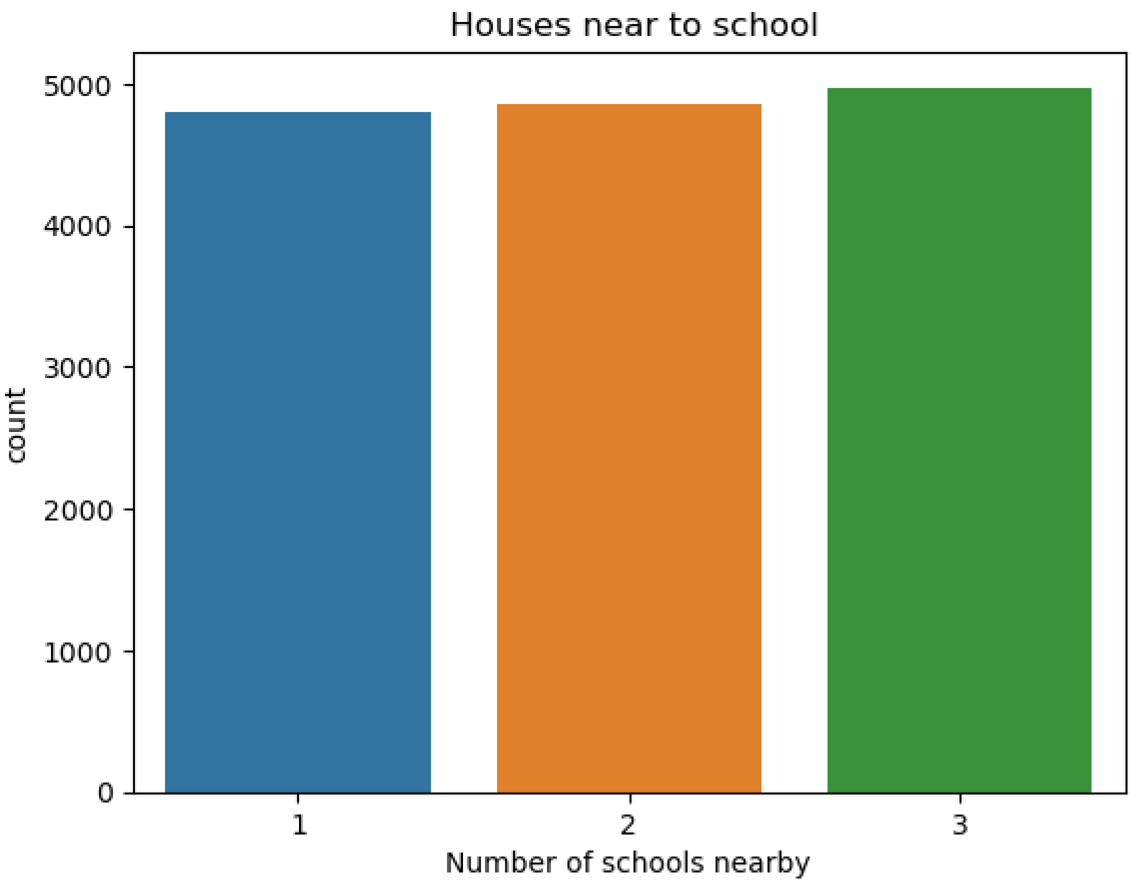
In [26]:
```
sns.countplot(data=df,x='grade of the house')
plt.title('Grade Of the Houses')
plt.show()
```



From the above Plot we can easily see that

- there are approx 6000 Houses haveing grade 7.
- there are about less Houses haveing grade 4,12,13.

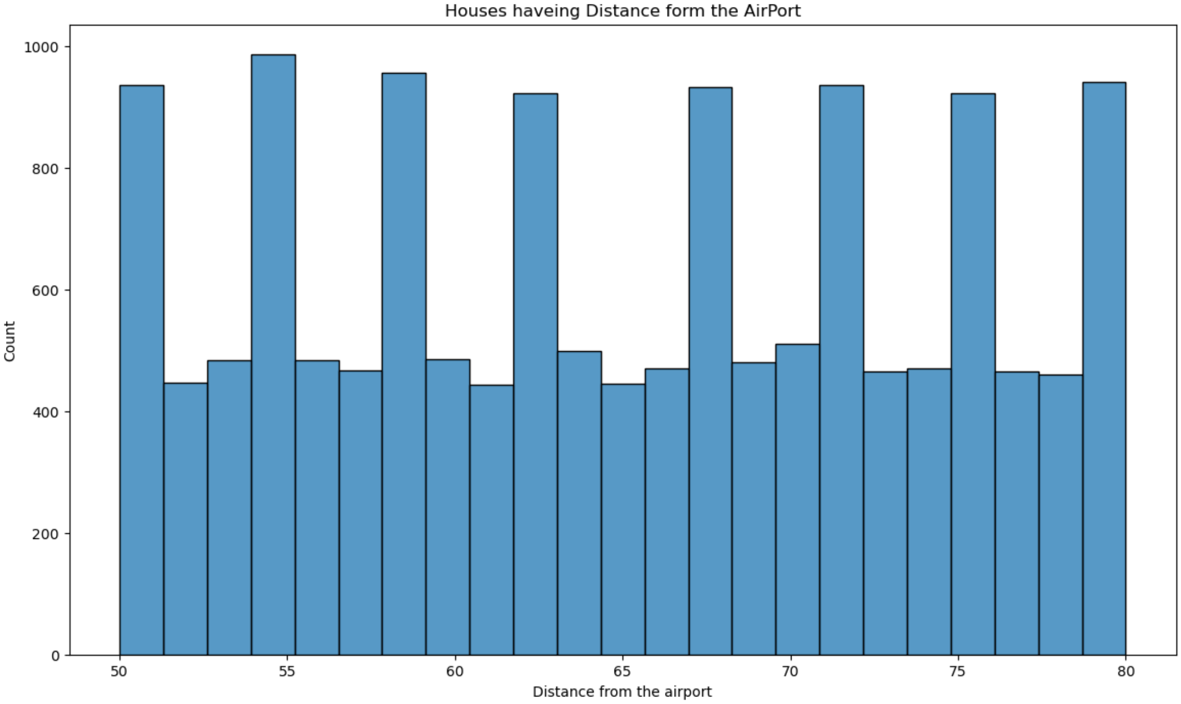
```
In [27]: sns.countplot(data=df,x='Number of schools nearby')
         plt.title(' Houses near to school')
         plt.show()
```



From the above plot we can easily see that

- Maximum Number of 3 schools are present near the House.

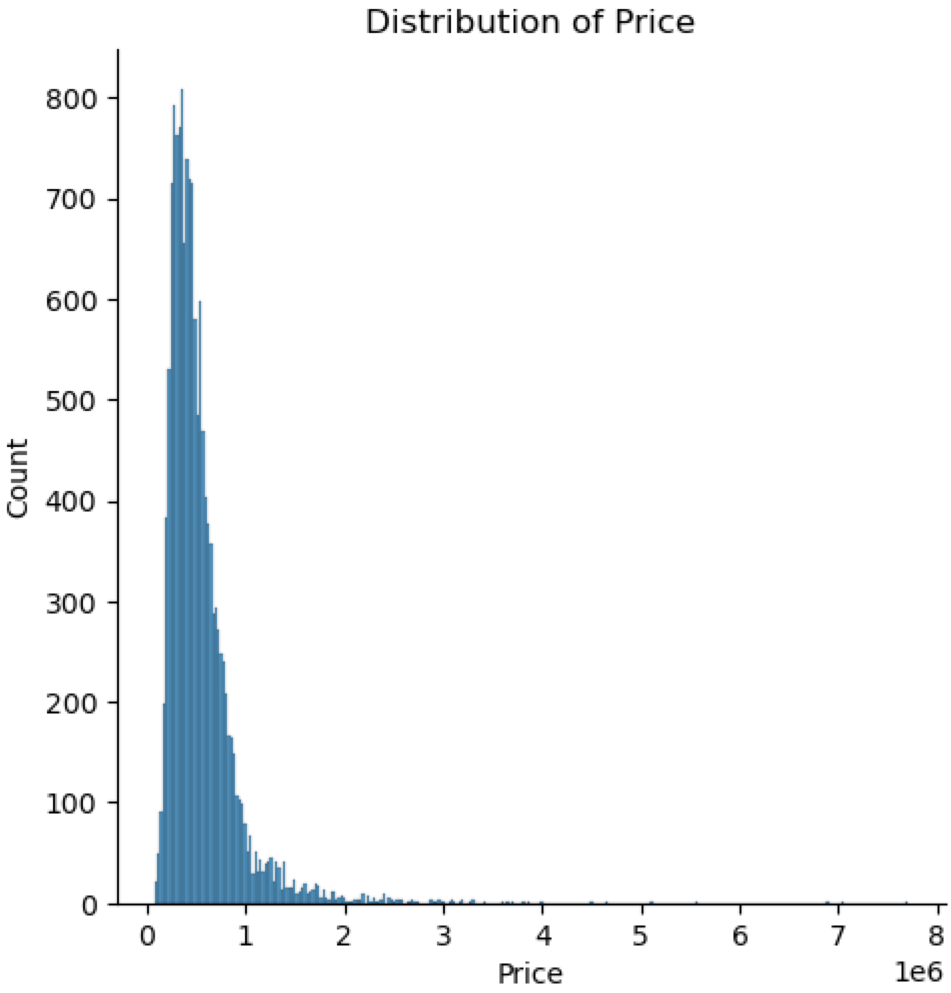
```
In [28]: plt.figure(figsize=(14,8))
         sns.histplot(data=df,x='Distance from the airport')
         plt.title('Houses haveing Distance form the AirPort')
         plt.show()
```



From above plot we can say that-

- There are maximum number houses haveing distance 55km from the airport.

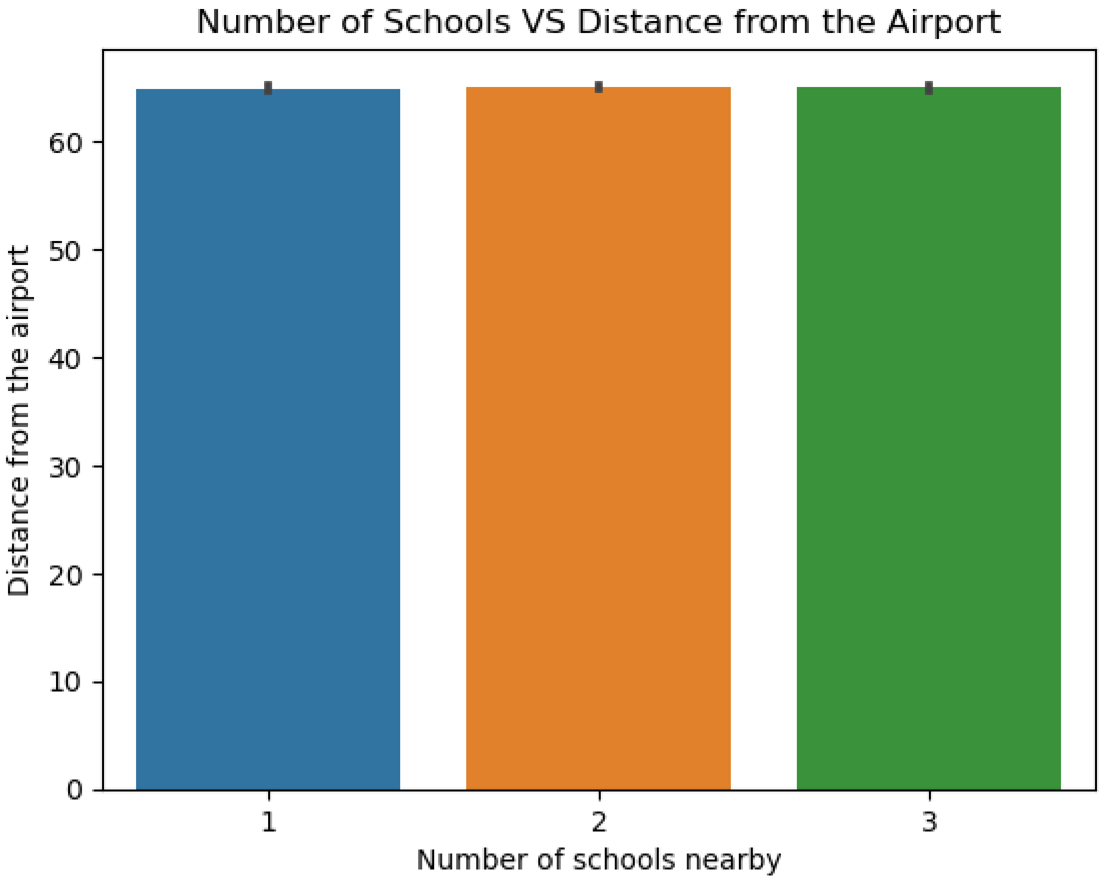
In [29]:
```
sns.displot(df['Price'])
plt.title('Distribution of Price')
plt.show()
```



From the above plot we can say that

- Maximum no. of Houses haveing Price in the range of (0-1).

```
In [30]: sns.barplot(data=df,x='Number of schools nearby',y='Distance from the airport'
         plt.title("Number of Schools VS Distance from the Airport")
         plt.show()
```



From the above Plot We can say that -

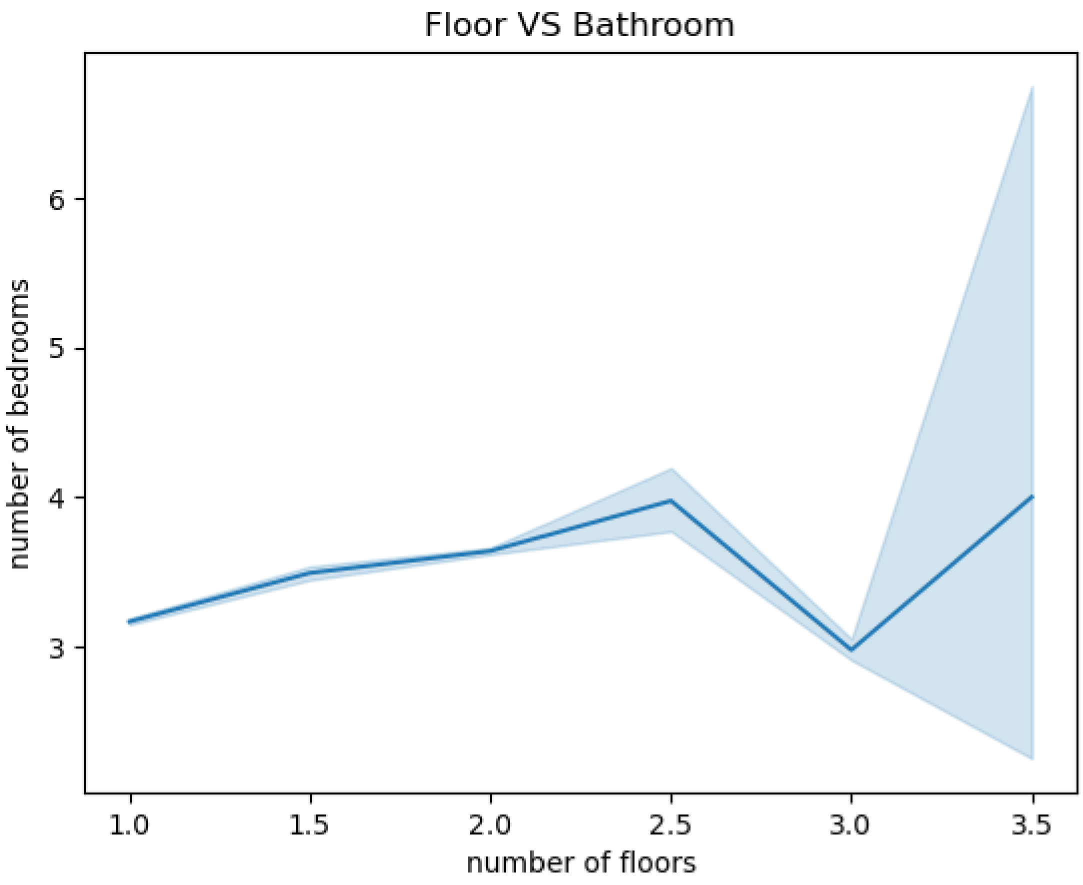
- the distance between the schools and Airports have more than 60km.

```
In [31]: df.columns
```

```
Out[31]: Index(['number of bedrooms', 'number of bathrooms', 'living area', 'lot are
         a',
                'number of floors', 'waterfront present', 'number of views',
                'condition of the house', 'grade of the house',
                'Area of the house(excluding basement)', 'Area of the basement',
                'Built Year', 'Renovation Year', 'living_area_renov', 'lot_area_reno
         v',
                'Number of schools nearby', 'Distance from the airport', 'Price'],
               dtype='object')
```

In [32]:

```
sns.lineplot(data=df,x='number of floors',y='number of bedrooms')
plt.title("Floor VS Bathroom")
plt.show()
```

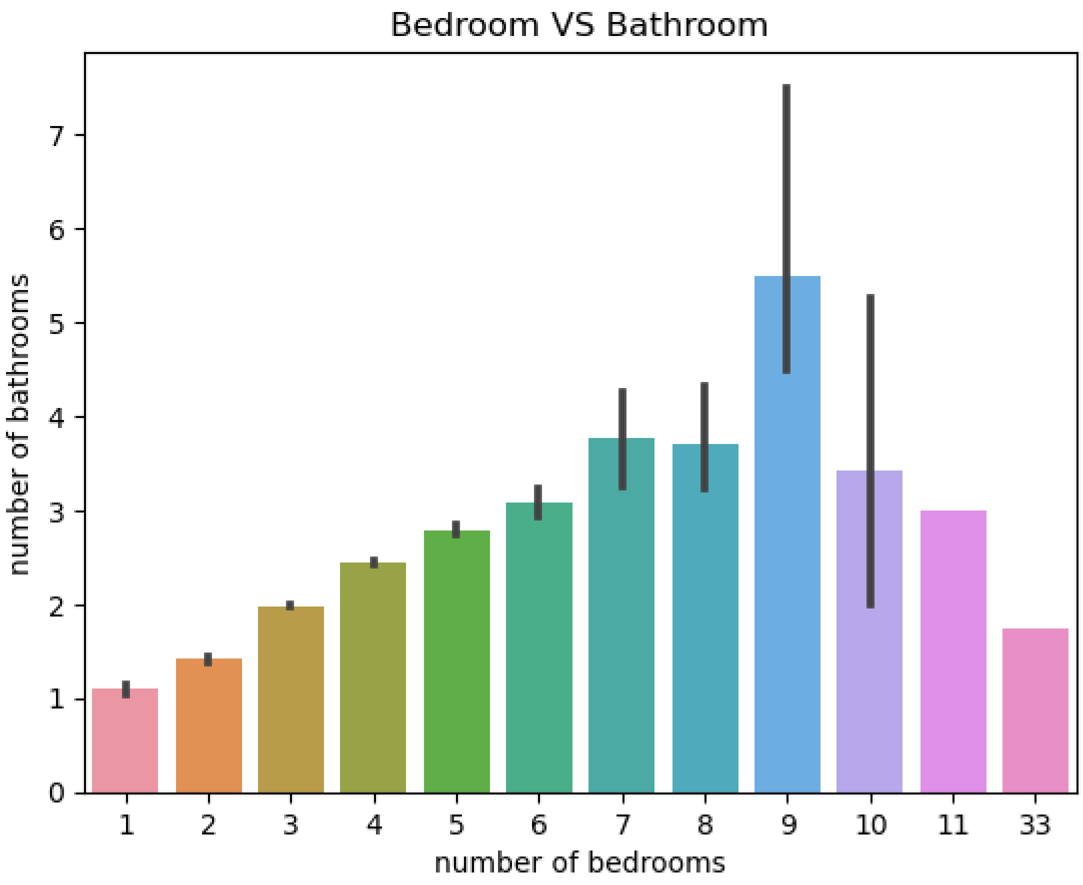


From the above Plot we can easily see that-

- Floor haveing 2.5 and 3.5 haveing more number of bedrooms as comparison to the 3.0.

In [33]:

```
sns.barplot(data=df,x='number of bedrooms',y='number of bathrooms')
plt.title("Bedroom VS Bathroom")
plt.show()
```

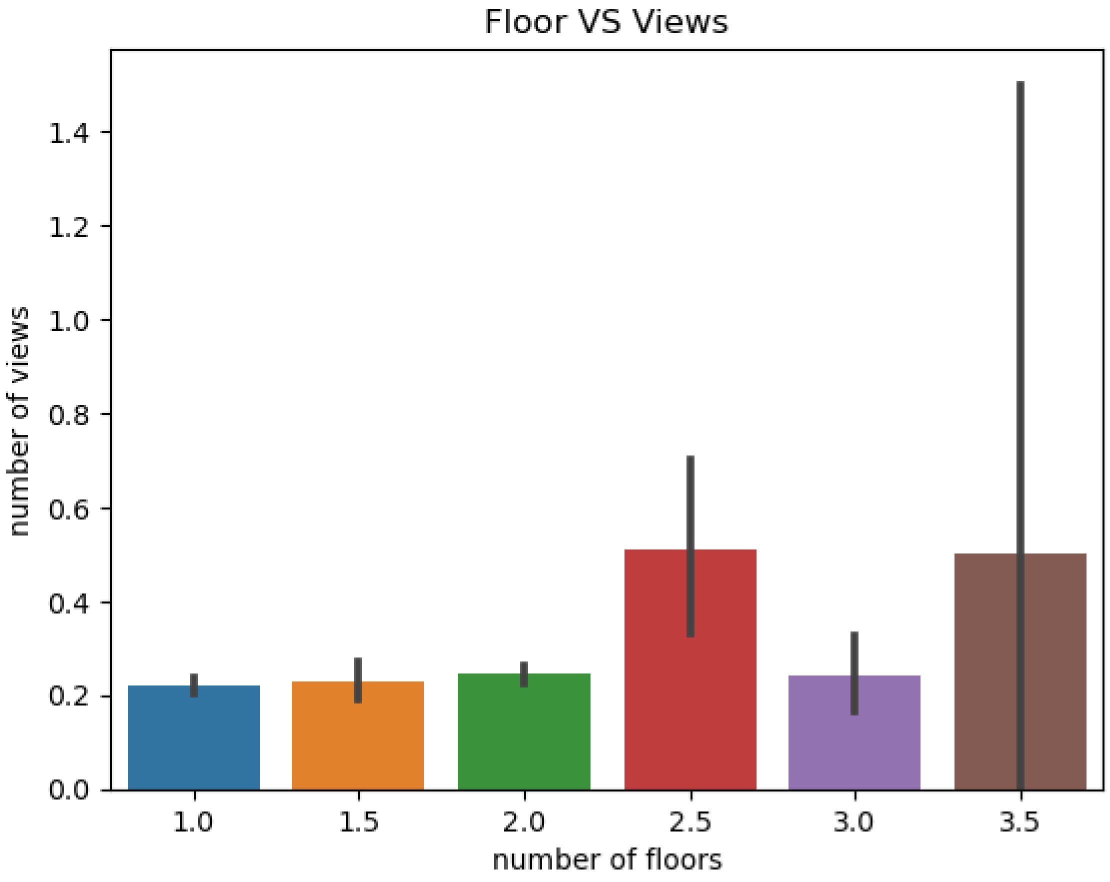


From the above plot we can easily see that-

- 9 bedrooms haveing high number of bathrooms.
- House haveing 1 Bedroom have 1 batroom.

In [34]:

```
sns.barplot(data=df,x='number of floors',y='number of views')
plt.title("Floor VS Views")
plt.show()
```



From the above plot we can conclude that -

- House haveing floor 2.5 and 3.5 haveing approx 0.5 number of views.

Conclusion:-

- In Maximum No. of houses there are 3 Bedrooms. and The Houses where More than 7 bedrooms are less in count.
- There are many Houses haveing 2.5 bathrooms.and Houses haveing 1 bathroom lie on second Number.& Houses with 0.5,0.75,1.25, and more than 4.75 bathrooms are very less in count.
- There are Maximum Number Of Houses haveing 1 Floor.
- 9 bedrooms haveing high number of bathrooms. and House haveing 1 Bedroom have 1 batroom.
- House haveing floor 2.5 and 3.5 haveing approx 0.5 number of views.

------------------------------ Thank you ------------------------------