In [1]:
```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
%matplotlib inline
import matplotlib
matplotlib.rcParams['figure.figsize']=(20,10)
```

In [2]:
```
import warnings
warnings.filterwarnings("ignore")
```

In [3]:
```
df1= pd.read_csv("C:\\Users\\thada\\Downloads\\bangaluru_house_prices.csv")
df1.head()
```

Out[3]:

	area_type	availability	location	size	society	total_sqft	bath	balcony	price
0	Super built-up Area	19-Dec	Electronic City Phase II	2 BHK	Coomee	1056	2.0	1.0	39.07
1	Plot Area	Ready To Move	Chikka Tirupathi	4 Bedroom	Theanmp	2600	5.0	3.0	120.00
2	Built-up Area	Ready To Move	Uttarahalli	3 BHK	NaN	1440	2.0	3.0	62.00
3	Super built-up Area	Ready To Move	Lingadheeranahalli	3 BHK	Soiewre	1521	3.0	1.0	95.00
4	Super built-up Area	Ready To Move	Kothanur	2 BHK	NaN	1200	2.0	1.0	51.00

In [4]:
```
df1.shape
```

Out[4]:
```
(13320, 9)
```

In [5]:
```
df1.groupby('area_type')['area_type'].agg('count')
```

Out[5]:
```
area_type
Built-up  Area          2418
Carpet  Area              87
Plot  Area              2025
Super built-up  Area    8790
Name: area_type, dtype: int64
```

In [6]:
```
df2=df1.drop(['area_type','society','balcony','availability'],axis='columns')
```

In [7]:
```
df2.head()
```

Out[7]:

	location	size	total_sqft	bath	price
0	Electronic City Phase II	2 BHK	1056	2.0	39.07
1	Chikka Tirupathi	4 Bedroom	2600	5.0	120.00
2	Uttarahalli	3 BHK	1440	2.0	62.00
3	Lingadheeranahalli	3 BHK	1521	3.0	95.00
4	Kothanur	2 BHK	1200	2.0	51.00

In [8]:
```
df2.isnull().sum()
```

Out[8]:
```
location       1
size          16
total_sqft     0
bath          73
price          0
dtype: int64
```

In [9]:
```
df3 = df2.dropna()
df3.isnull().sum()
```

Out[9]:
```
location      0
size          0
total_sqft    0
bath          0
price         0
dtype: int64
```

In [10]:
```
df3.shape
```

Out[10]:
```
(13246, 5)
```

In [11]:
```
df3['size'].unique()
```

Out[11]:
```
array(['2 BHK', '4 Bedroom', '3 BHK', '4 BHK', '6 Bedroom', '3 Bedroom',
       '1 BHK', '1 RK', '1 Bedroom', '8 Bedroom', '2 Bedroom',
       '7 Bedroom', '5 BHK', '7 BHK', '6 BHK', '5 Bedroom', '11 BHK',
       '9 BHK', '9 Bedroom', '27 BHK', '10 Bedroom', '11 Bedroom',
       '10 BHK', '19 BHK', '16 BHK', '43 Bedroom', '14 BHK', '8 BHK',
       '12 Bedroom', '13 BHK', '18 Bedroom'], dtype=object)
```

In [12]:
```
df3['bhk']=df3['size'].apply(lambda x: int(x.split(' ')[0]))
df3.bhk.unique()
```

Out[12]:
```
array([ 2,  4,  3,  6,  1,  8,  7,  5, 11,  9, 27, 10, 19, 16, 43, 14, 12,
       13, 18], dtype=int64)
```

In [13]:
```
df3.head()
```

Out[13]:

	location	size	total_sqft	bath	price	bhk
0	Electronic City Phase II	2 BHK	1056	2.0	39.07	2
1	Chikka Tirupathi	4 Bedroom	2600	5.0	120.00	4
2	Uttarahalli	3 BHK	1440	2.0	62.00	3
3	Lingadheeranahalli	3 BHK	1521	3.0	95.00	3
4	Kothanur	2 BHK	1200	2.0	51.00	2

In [14]:
```
df3[df3.bhk>20]
```

Out[14]:

	location	size	total_sqft	bath	price	bhk
1718	2Electronic City Phase II	27 BHK	8000	27.0	230.0	27
4684	Munnekollal	43 Bedroom	2400	40.0	660.0	43

In [15]:
```
df3.total_sqft.unique()
```

Out[15]:
```
array(['1056', '2600', '1440', ..., '1133 - 1384', '774', '4689'],
      dtype=object)
```

In [16]:
```
def is_float(x):
    try:
        float(x)
    except:
        return False
    return True
```

In [17]:
```
df3[~df3['total_sqft'].apply(is_float)]
```

Out[17]:

	location	size	total_sqft	bath	price	bhk
30	Yelahanka	4 BHK	2100 - 2850	4.0	186.000	4
122	Hebbal	4 BHK	3067 - 8156	4.0	477.000	4
137	8th Phase JP Nagar	2 BHK	1042 - 1105	2.0	54.005	2
165	Sarjapur	2 BHK	1145 - 1340	2.0	43.490	2
188	KR Puram	2 BHK	1015 - 1540	2.0	56.800	2
...	...	...	...	...	...	...
12975	Whitefield	2 BHK	850 - 1060	2.0	38.190	2
12990	Talaghattapura	3 BHK	1804 - 2273	3.0	122.000	3
13059	Harlur	2 BHK	1200 - 1470	2.0	72.760	2
13265	Hoodi	2 BHK	1133 - 1384	2.0	59.135	2
13299	Whitefield	4 BHK	2830 - 2882	5.0	154.500	4

190 rows × 6 columns

In [18]:
```
def convert_sqft_to_num(x):
    tokens = x.split('-')
    if len(tokens) == 2:
        return (float(tokens[0])+float(tokens[1]))/2
    try:
        return float(x)
    except:
        return None
```

In [19]:
```
df4 = df3.copy()
```

In [20]:
```
df4['total_sqft']=df4['total_sqft'].apply(convert_sqft_to_num)
```

In [21]:
```
df4.head()
```

Out[21]:

	location	size	total_sqft	bath	price	bhk
0	Electronic City Phase II	2 BHK	1056.0	2.0	39.07	2
1	Chikka Tirupathi	4 Bedroom	2600.0	5.0	120.00	4
2	Uttarahalli	3 BHK	1440.0	2.0	62.00	3
3	Lingadheeranahalli	3 BHK	1521.0	3.0	95.00	3
4	Kothanur	2 BHK	1200.0	2.0	51.00	2

In [22]:
```
df5=df4.copy()
# creating the column with price per sqft in consict price in lakh
df5["price_per_sqft"]=df5['price']*100000/df5['total_sqft']
df5.head()
```

Out[22]:

	location	size	total_sqft	bath	price	bhk	price_per_sqft
0	Electronic City Phase II	2 BHK	1056.0	2.0	39.07	2	3699.810606
1	Chikka Tirupathi	4 Bedroom	2600.0	5.0	120.00	4	4615.384615
2	Uttarahalli	3 BHK	1440.0	2.0	62.00	3	4305.555556
3	Lingadheeranahalli	3 BHK	1521.0	3.0	95.00	3	6245.890861
4	Kothanur	2 BHK	1200.0	2.0	51.00	2	4250.000000

In [23]:
```
df5.location.unique()
```

Out[23]:
```
array(['Electronic City Phase II', 'Chikka Tirupathi', 'Uttarahalli', ...,
       '12th cross srinivas nagar banshankari 3rd stage',
       'Havanur extension', 'Abshot Layout'], dtype=object)
```

In [24]:
```
len(df5.location.unique())
```

Out[24]:
```
1304
```

In [25]:
```
df5.location = df5.location.apply(lambda x: x.strip()) # removing white space

location_stats= df5.groupby('location')['location'].agg('count')
location_stats
```

Out[25]:
```
location
1 Annasandrapalya                                  1
1 Giri Nagar                                       1
1 Immadihalli                                      1
1 Ramamurthy Nagar                                 1
12th cross srinivas nagar banshankari 3rd stage    1
                                                  ..
t.c palya                                          1
tc.palya                                           4
vinayakanagar                                      1
white field,kadugodi                               1
whitefiled                                         1
Name: location, Length: 1293, dtype: int64
```

In [26]:
```
location_stats= df5.groupby('location')['location'].agg('count').sort_values(ascending=False)
location_stats
```

Out[26]:
```
location
Whitefield               535
Sarjapur  Road           392
Electronic City          304
Kanakpura Road           266
Thanisandra              236
                        ...
1 Giri Nagar               1
Kanakapura Road,           1
Kanakapura main  Road      1
Karnataka Shabarimala      1
whitefiled                 1
Name: location, Length: 1293, dtype: int64
```

In [27]:
```
len(location_stats[location_stats<=10])
```

Out[27]:
```
1052
```

In [28]:
```
location_stats_less_then_10 = location_stats[location_stats<=10]
location_stats_less_then_10
```

Out[28]:
```
location
Basapura                 10
1st Block Koramangala    10
Gunjur Palya             10
Kalkere                  10
Sector 1 HSR Layout      10
                         ..
1 Giri Nagar              1
Kanakapura Road,          1
Kanakapura main  Road     1
Karnataka Shabarimala     1
whitefiled                1
Name: location, Length: 1052, dtype: int64
```

In [29]:
```
len(df5.location.unique())
```

Out[29]: 1293

In [30]:
```
df5.location = df5.location.apply(lambda x: 'other' if x in location_stats_less_then_10 else x
```

In [31]:
```
len(df5.location.unique())
```

Out[31]: 242

In [32]:
```
df5.head(10)
```

Out[32]:

	location	size	total_sqft	bath	price	bhk	price_per_sqft
0	Electronic City Phase II	2 BHK	1056.0	2.0	39.07	2	3699.810606
1	Chikka Tirupathi	4 Bedroom	2600.0	5.0	120.00	4	4615.384615
2	Uttarahalli	3 BHK	1440.0	2.0	62.00	3	4305.555556
3	Lingadheeranahalli	3 BHK	1521.0	3.0	95.00	3	6245.890861
4	Kothanur	2 BHK	1200.0	2.0	51.00	2	4250.000000
5	Whitefield	2 BHK	1170.0	2.0	38.00	2	3247.863248
6	Old Airport Road	4 BHK	2732.0	4.0	204.00	4	7467.057101
7	Rajaji Nagar	4 BHK	3300.0	4.0	600.00	4	18181.818182
8	Marathahalli	3 BHK	1310.0	3.0	63.25	3	4828.244275
9	other	6 Bedroom	1020.0	6.0	370.00	6	36274.509804

In [33]:
```
df5[df5.total_sqft/df5.bhk<300].head()
```

Out[33]:

	location	size	total_sqft	bath	price	bhk	price_per_sqft
9	other	6 Bedroom	1020.0	6.0	370.0	6	36274.509804
45	HSR Layout	8 Bedroom	600.0	9.0	200.0	8	33333.333333
58	Murugeshpalya	6 Bedroom	1407.0	4.0	150.0	6	10660.980810
68	Devarachikkanahalli	8 Bedroom	1350.0	7.0	85.0	8	6296.296296
70	other	3 Bedroom	500.0	3.0	100.0	3	20000.000000

In [34]:
```
df5.shape
```

Out[34]: (13246, 7)

In [35]:
```
df6=df5[~(df5.total_sqft/df5.bhk<300)]
df6.shape
```

Out[35]: (12502, 7)

In [36]:
```
df6.price_per_sqft.describe()
```

Out[36]:
```
count     12456.000000
mean       6308.502826
std        4168.127339
min         267.829813
25%        4210.526316
50%        5294.117647
75%        6916.666667
max      176470.588235
Name: price_per_sqft, dtype: float64
```

In [37]:
```
df6
```

Out[37]:

	location	size	total_sqft	bath	price	bhk	price_per_sqft
0	Electronic City Phase II	2 BHK	1056.0	2.0	39.07	2	3699.810606
1	Chikka Tirupathi	4 Bedroom	2600.0	5.0	120.00	4	4615.384615
2	Uttarahalli	3 BHK	1440.0	2.0	62.00	3	4305.555556
3	Lingadheeranahalli	3 BHK	1521.0	3.0	95.00	3	6245.890861
4	Kothanur	2 BHK	1200.0	2.0	51.00	2	4250.000000
...	...	...	...	...	...	...	...
13315	Whitefield	5 Bedroom	3453.0	4.0	231.00	5	6689.834926
13316	other	4 BHK	3600.0	5.0	400.00	4	11111.111111
13317	Raja Rajeshwari Nagar	2 BHK	1141.0	2.0	60.00	2	5258.545136
13318	Padmanabhanagar	4 BHK	4689.0	4.0	488.00	4	10407.336319
13319	Doddathoguru	1 BHK	550.0	1.0	17.00	1	3090.909091

12502 rows × 7 columns

In [38]:
```
def remove_pps_outliers(df):
    df_out = pd.DataFrame()
    for key, subdf in df.groupby('location'):
        m=np.mean(subdf.price_per_sqft)
        st=np.std(subdf.price_per_sqft)
        reduced_df=subdf[(subdf.price_per_sqft>(m-st)) & (subdf.price_per_sqft<(m+st))]
        df_out=pd.concat([df_out,reduced_df],ignore_index=True)
    return df_out
df7 = remove_pps_outliers(df6)
df7.shape
```

Out[38]: (10241, 7)

In [39]:
```
df7
```

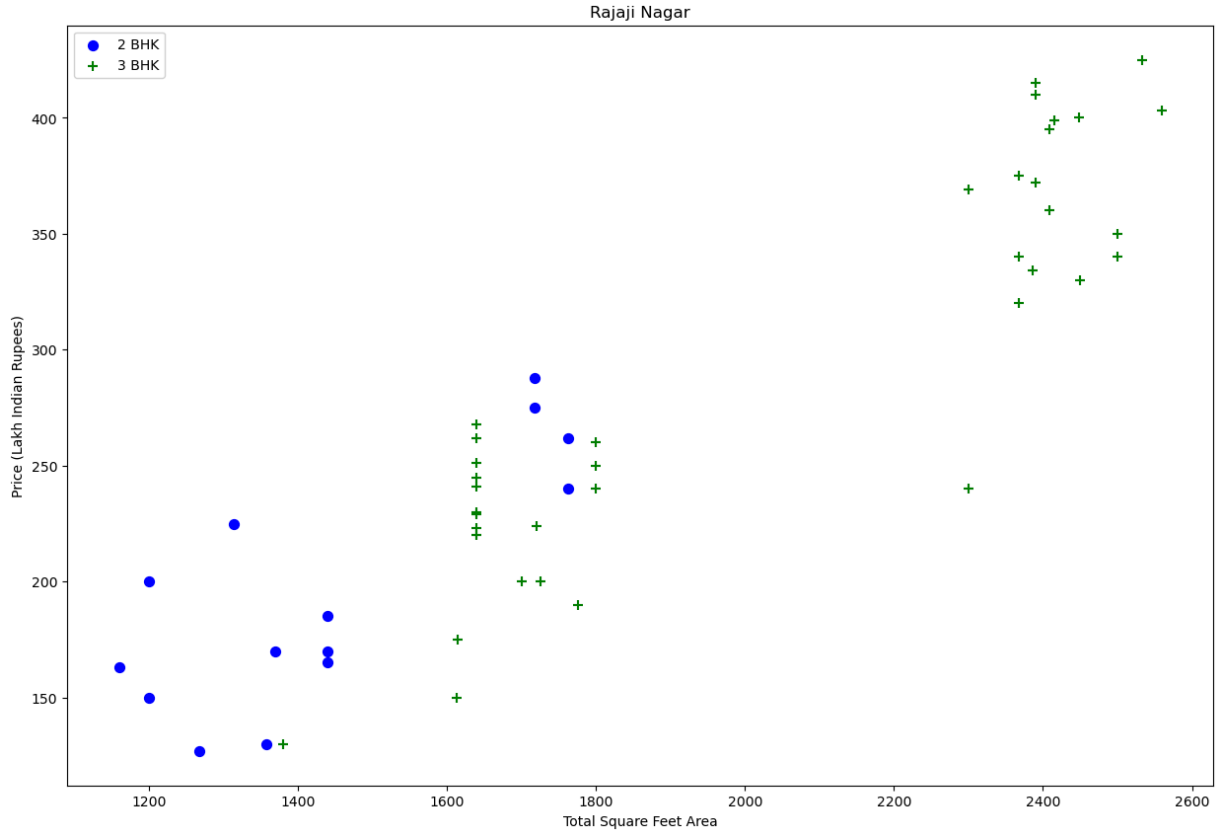
Out[39]:

	location	size	total_sqft	bath	price	bhk	price_per_sqft
0	1st Block Jayanagar	4 BHK	2850.0	4.0	428.00	4	15017.543860
1	1st Block Jayanagar	3 BHK	1630.0	3.0	194.00	3	11901.840491
2	1st Block Jayanagar	3 BHK	1875.0	2.0	235.00	3	12533.333333
3	1st Block Jayanagar	3 BHK	1200.0	2.0	130.00	3	10833.333333
4	1st Block Jayanagar	2 BHK	1235.0	2.0	148.00	2	11983.805668
...	...	...	...	...	...	...	...
10236	other	2 BHK	1353.0	2.0	110.00	2	8130.081301
10237	other	1 Bedroom	812.0	1.0	26.00	1	3201.970443
10238	other	3 BHK	1440.0	2.0	63.93	3	4439.583333
10239	other	2 BHK	1075.0	2.0	48.00	2	4465.116279
10240	other	4 BHK	3600.0	5.0	400.00	4	11111.111111

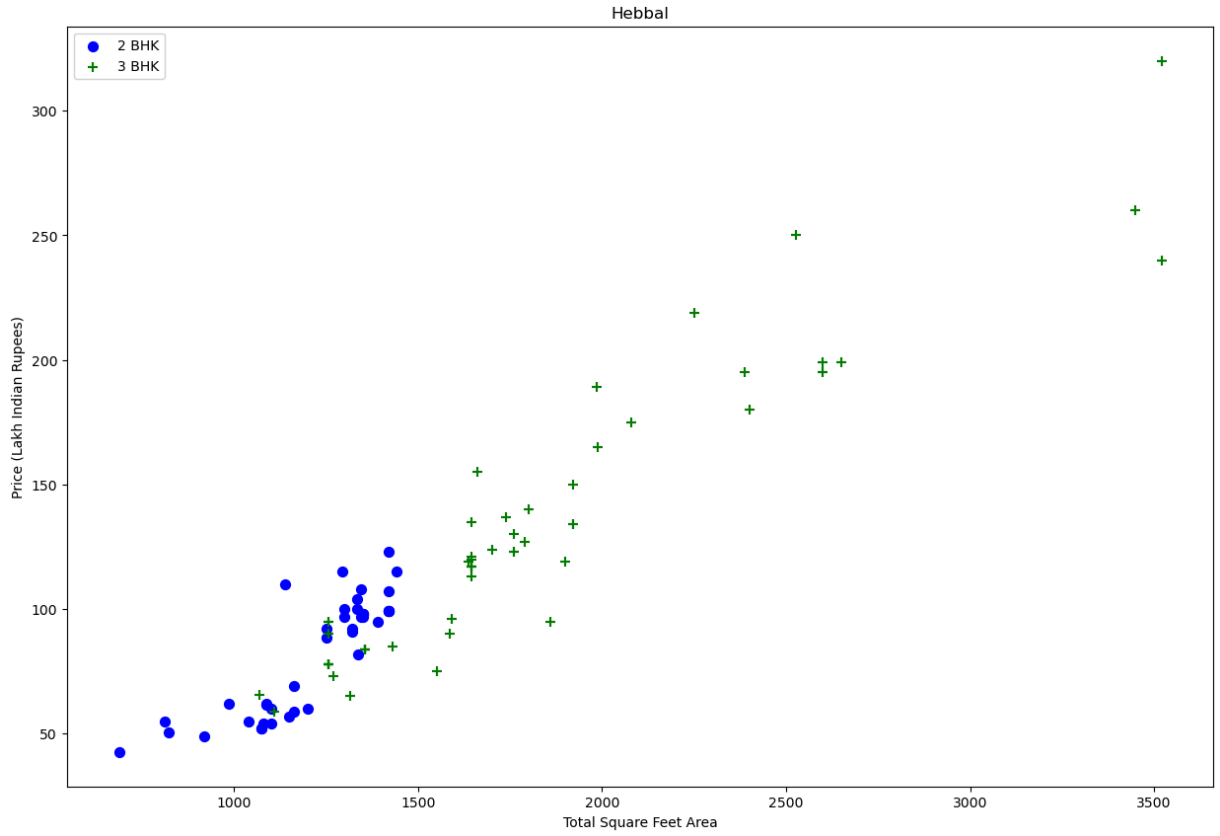
10241 rows × 7 columns

In [40]:
```
def plot_scatter_chart(df,location):
    bhk2 = df[(df.location==location) & (df.bhk==2)]
    bhk3 = df[(df.location==location) & (df.bhk==3)]
    matplotlib.rcParams['figure.figsize'] = (15,10)
    plt.scatter(bhk2.total_sqft,bhk2.price,color='blue',label='2 BHK', s=50)
    plt.scatter(bhk3.total_sqft,bhk3.price,marker='+', color='green',label='3 BHK', s=50)
    plt.xlabel("Total Square Feet Area")
    plt.ylabel("Price (Lakh Indian Rupees)")
    plt.title(location)
    plt.legend()

plot_scatter_chart(df7,"Rajaji Nagar")
```



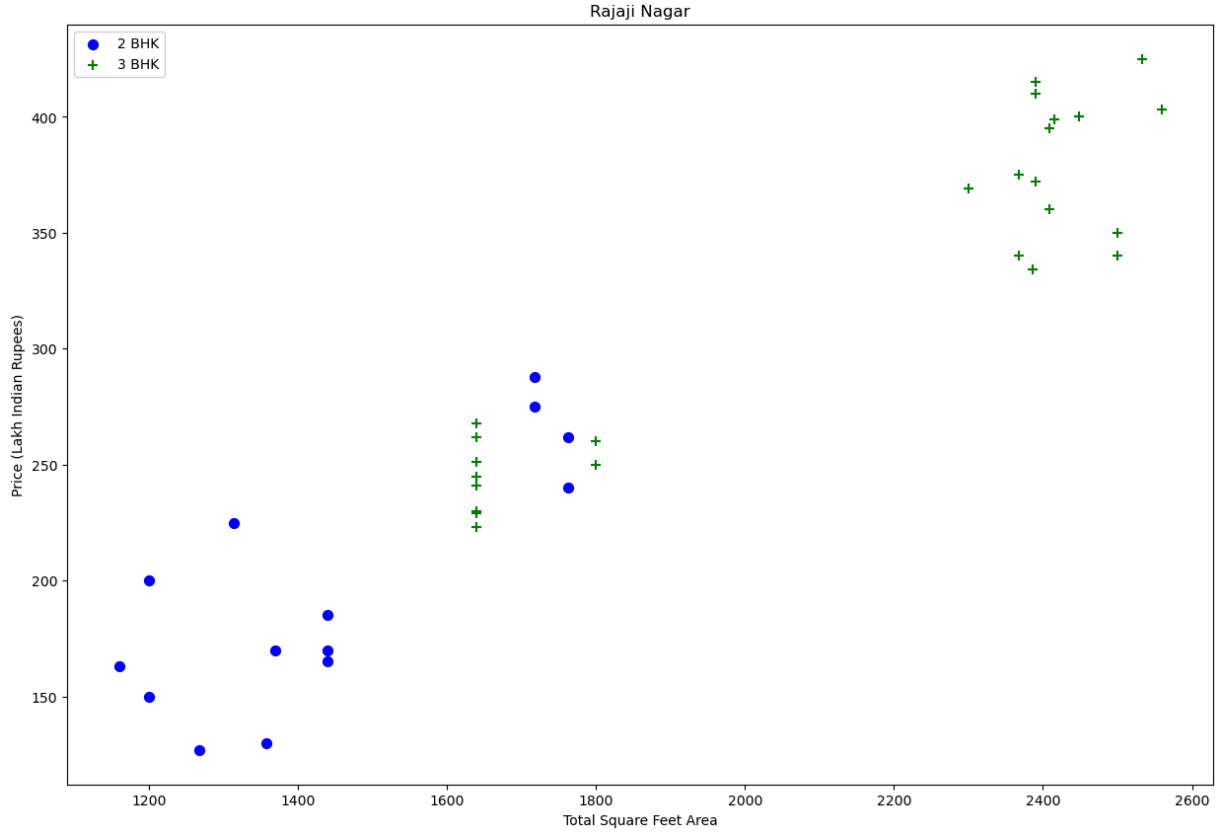
In [41]:
```
plot_scatter_chart(df7,"Hebbal")
```



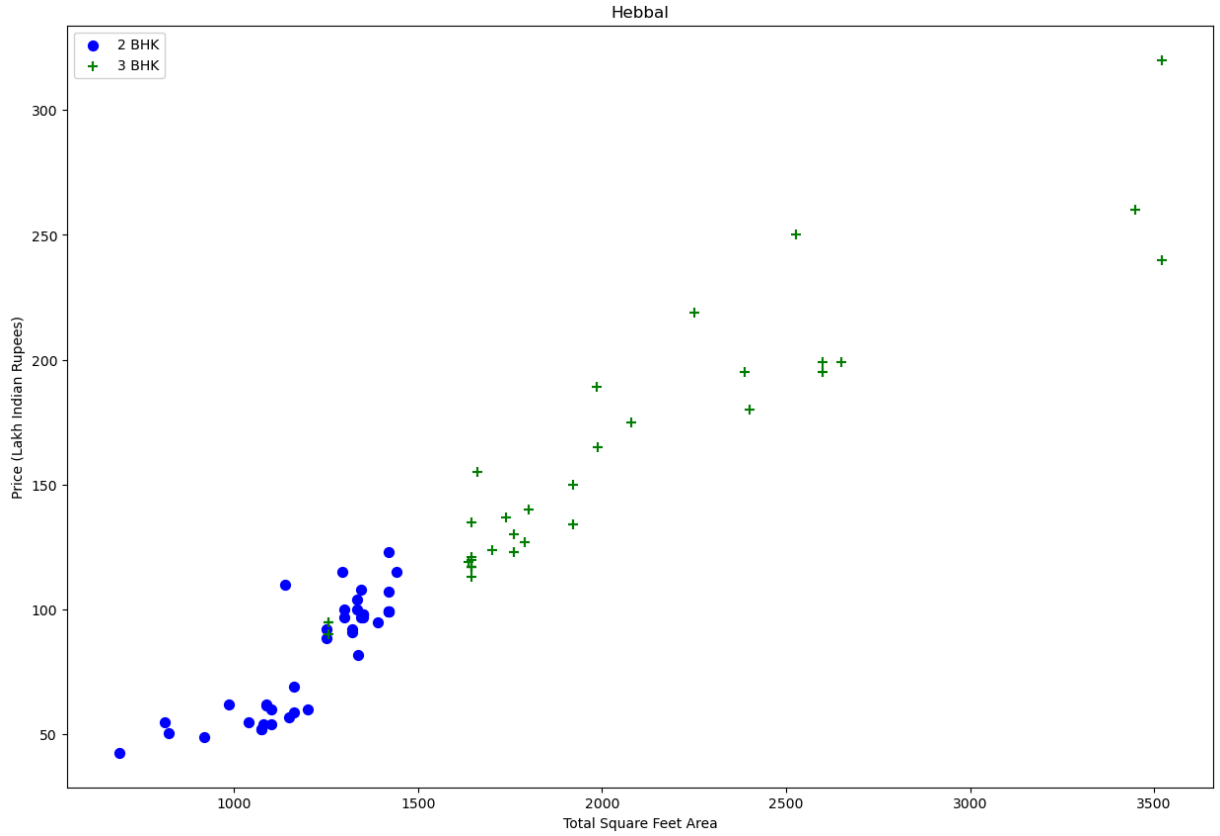
In [42]:
```
def remove_bhk_outliers(df):
    exclude_indices = np.array([])
    for location, location_df in df.groupby('location'):
        bhk_stats = {}
        for bhk, bhk_df in location_df.groupby('bhk'):
            bhk_stats[bhk] = {
                'mean': np.mean(bhk_df.price_per_sqft),
                'std': np.std(bhk_df.price_per_sqft),
                'count': bhk_df.shape[0]
            }
        for bhk, bhk_df in location_df.groupby('bhk'):
            stats = bhk_stats.get(bhk-1)
            if stats and stats['count']>5:
                exclude_indices = np.append(exclude_indices, bhk_df[bhk_df.price_per_sqft<(sta
    return df.drop(exclude_indices,axis='index')
df8 = remove_bhk_outliers(df7)
# df8 = df7.copy()
df8.shape
```

Out[42]: (7329, 7)

In [43]:
```
plot_scatter_chart(df8,"Rajaji Nagar")
```

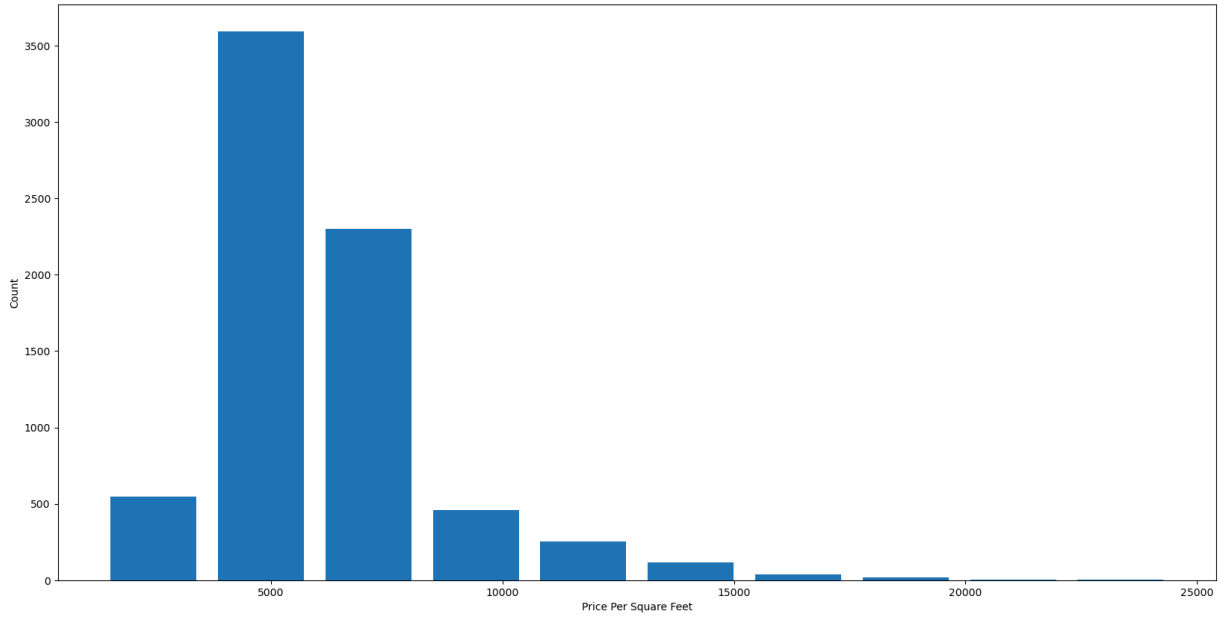


In [44]:
```
plot_scatter_chart(df8,"Hebbal")
```



In [45]:
```
import matplotlib
matplotlib.rcParams["figure.figsize"] = (20,10)
plt.hist(df8.price_per_sqft,rwidth=0.8)
plt.xlabel("Price Per Square Feet")
plt.ylabel("Count")
```

Out[45]:
```
Text(0, 0.5, 'Count')
```



In [46]:
```
df8.bath.unique()
```

Out[46]:
```
array([ 4.,  3.,  2.,  5.,  8.,  1.,  6.,  7.,  9., 12., 16., 13.])
```

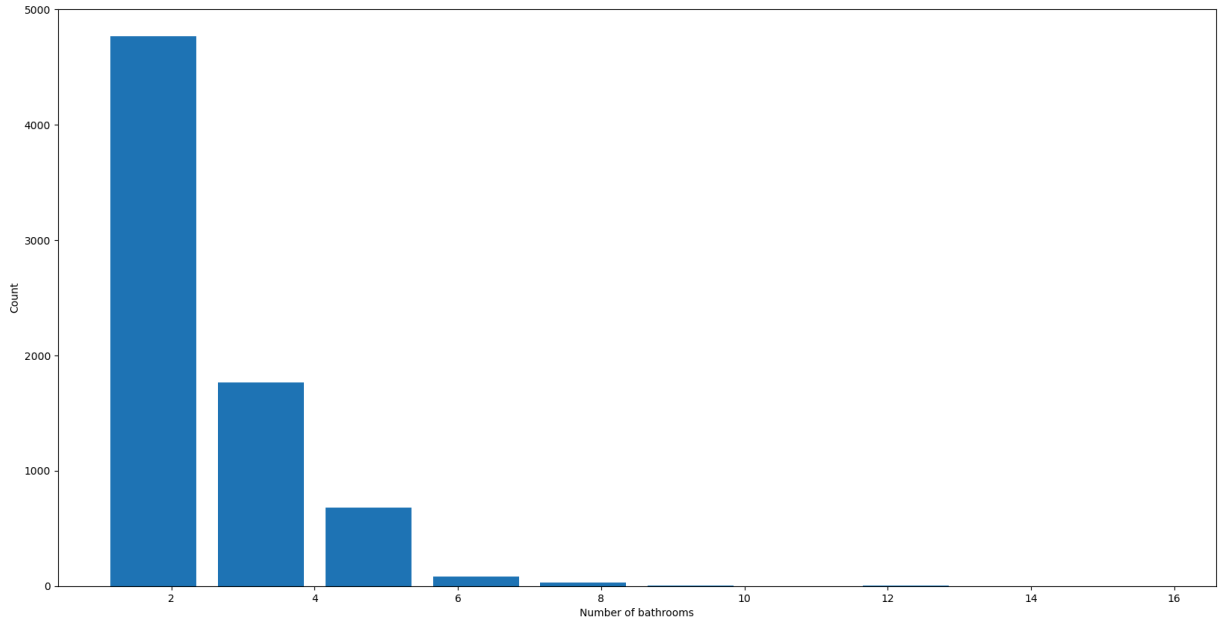
In [47]:
```
df8[df8.bath>10]
```

Out[47]:

	location	size	total_sqft	bath	price	bhk	price_per_sqft
5277	Neeladri Nagar	10 BHK	4000.0	12.0	160.0	10	4000.000000
8486	other	10 BHK	12000.0	12.0	525.0	10	4375.000000
8575	other	16 BHK	10000.0	16.0	550.0	16	5500.000000
9308	other	11 BHK	6000.0	12.0	150.0	11	2500.000000
9639	other	13 BHK	5425.0	13.0	275.0	13	5069.124424

In [48]:
```
plt.hist(df8.bath,rwidth=0.8)
plt.xlabel("Number of bathrooms")
plt.ylabel("Count")
```

Out[48]:
```
Text(0, 0.5, 'Count')
```



In [49]:
```
df8[df8.bath>df8.bhk+2]
```

Out[49]:

	location	size	total_sqft	bath	price	bhk	price_per_sqft
1626	Chikkabanavar	4 Bedroom	2460.0	7.0	80.0	4	3252.032520
5238	Nagasandra	4 Bedroom	7000.0	8.0	450.0	4	6428.571429
6711	Thanisandra	3 BHK	1806.0	6.0	116.0	3	6423.034330
8411	other	6 BHK	11338.0	9.0	1000.0	6	8819.897689

In [50]:
```
df9=df8[df8.bath<df8.bhk+2]
df9.shape
```

Out[50]:
```
(7251, 7)
```

In [51]:
```
df10 = df9.drop(['size','price_per_sqft'],axis='columns')
df10.head(3)
```

Out[51]:

	location	total_sqft	bath	price	bhk
0	1st Block Jayanagar	2850.0	4.0	428.0	4
1	1st Block Jayanagar	1630.0	3.0	194.0	3
2	1st Block Jayanagar	1875.0	2.0	235.0	3

In [52]:
```
dummies = pd.get_dummies(df10.location)
dummies.head()
```

Out[52]:

	1st Block Jayanagar	1st Phase JP Nagar	2nd Phase Judicial Layout	2nd Stage Nagarbhavi	5th Block Hbr Layout	5th Phase JP Nagar	6th Phase JP Nagar	7th Phase JP Nagar	8th Phase JP Nagar	9th Phase JP Nagar	...	Vishveshwarya Layout	Vishwa L
0	True	False	False	False	False	False	False	False	False	False	...	False	
1	True	False	False	False	False	False	False	False	False	False	...	False	
2	True	False	False	False	False	False	False	False	False	False	...	False	
3	True	False	False	False	False	False	False	False	False	False	...	False	
4	True	False	False	False	False	False	False	False	False	False	...	False	

5 rows × 242 columns

In [53]:
```
df11 = pd.concat([df10,dummies.drop('other',axis='columns')],axis='columns')
df11
```

Out[53]:

	location	total_sqft	bath	price	bhk	1st Block Jayanagar	1st Phase JP Nagar	2nd Phase Judicial Layout	2nd Stage Nagarbhavi	5th Block Hbr Layout	...	Vijayanagar	Vish
0	1st Block Jayanagar	2850.0	4.0	428.0	4	True	False	False	False	False	...	False	
1	1st Block Jayanagar	1630.0	3.0	194.0	3	True	False	False	False	False	...	False	
2	1st Block Jayanagar	1875.0	2.0	235.0	3	True	False	False	False	False	...	False	
3	1st Block Jayanagar	1200.0	2.0	130.0	3	True	False	False	False	False	...	False	
4	1st Block Jayanagar	1235.0	2.0	148.0	2	True	False	False	False	False	...	False	
...	...	...	...	...	...	...	...	...	...	...	...	...	
10232	other	1200.0	2.0	70.0	2	False	False	False	False	False	...	False	
10233	other	1800.0	1.0	200.0	1	False	False	False	False	False	...	False	
10236	other	1353.0	2.0	110.0	2	False	False	False	False	False	...	False	
10237	other	812.0	1.0	26.0	1	False	False	False	False	False	...	False	
10240	other	3600.0	5.0	400.0	4	False	False	False	False	False	...	False	

7251 rows × 246 columns

In [54]:
```
df12 = df11.drop('location',axis='columns')
df12.head(2)
```

Out[54]:

	total_sqft	bath	price	bhk	1st Block Jayanagar	1st Phase JP Nagar	2nd Phase Judicial Layout	2nd Stage Nagarbhavi	5th Block Hbr Layout	5th Phase JP Nagar	...	Vijayanagar	Vishveshwa Lay
0	2850.0	4.0	428.0	4	True	False	False	False	False	False	...	False	Fa
1	1630.0	3.0	194.0	3	True	False	False	False	False	False	...	False	Fa

2 rows × 245 columns

In [55]:
```
df12.shape
```

Out[55]:
```
(7251, 245)
```

In [56]:
```
X=df12.drop('price',axis='columns')
X.head()
```

Out[56]:

	total_sqft	bath	bhk	1st Block Jayanagar	1st Phase JP Nagar	2nd Phase Judicial Layout	2nd Stage Nagarbhavi	5th Block Hbr Layout	5th Phase JP Nagar	6th Phase JP Nagar	...	Vijayanagar	Vishveshwa Lay
0	2850.0	4.0	4	True	False	False	False	False	False	False	...	False	F
1	1630.0	3.0	3	True	False	False	False	False	False	False	...	False	F
2	1875.0	2.0	3	True	False	False	False	False	False	False	...	False	F
3	1200.0	2.0	3	True	False	False	False	False	False	False	...	False	F
4	1235.0	2.0	2	True	False	False	False	False	False	False	...	False	F

5 rows × 244 columns

In [57]:
```
y = df12.price
y.head()
```

Out[57]:
```
0    428.0
1    194.0
2    235.0
3    130.0
4    148.0
Name: price, dtype: float64
```

In [58]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2,random_state=10)
```

In [59]:
```
from sklearn.linear_model import LinearRegression
lr_clf = LinearRegression()
lr_clf.fit(X_train,y_train)
lr_clf.score(X_test,y_test)
```

Out[59]:
```
0.845227769787429
```

In [60]:
```
from sklearn.model_selection import ShuffleSplit
from sklearn.model_selection import cross_val_score
# ShuffleSplit will randomize our sample ,so that each of the fold has equal distribution of t
cv = ShuffleSplit(n_splits= 5,test_size=0.2,random_state=0)
cross_val_score(LinearRegression(),X,y,cv=cv)
```

Out[60]:
```
array([0.82430186, 0.77166234, 0.85089567, 0.80837764, 0.83653286])
```

In [61]:
```
from sklearn.model_selection import GridSearchCV
from sklearn.tree import DecisionTreeRegressor
from sklearn.linear_model import Lasso
```

In [63]:
```
from sklearn.model_selection import GridSearchCV, ShuffleSplit
from sklearn.linear_model import LinearRegression, Lasso
from sklearn.tree import DecisionTreeRegressor
import pandas as pd

def find_best_model_using_gridsearchcv(X, y):
    algos = {
        'linear_regression': {
            'model': LinearRegression(),
            'params': {}
        },
        'lasso': {
            'model': Lasso(),
            'params': {
                'alpha': [1, 2],
                'selection': ['random', 'cyclic']
            }
        },
        'decision_tree': {
            'model': DecisionTreeRegressor(),
            'params': {
                'criterion': ['mse', 'friedman_mse'],
                'splitter': ['best', 'random']
            }
        }
    }
    scores = []
    cv = ShuffleSplit(n_splits=5, test_size=0.2, random_state=0)
    for algo_name, config in algos.items():
        gs = GridSearchCV(config['model'], config['params'], cv=cv, return_train_score=False)
        gs.fit(X, y)
        scores.append({
            'model': algo_name,
            'best_score': gs.best_score_,
            'best_params': gs.best_params_
        })

    return pd.DataFrame(scores, columns=['model', 'best_score', 'best_params'])

# Assuming X and y are your features and target variable
find_best_model_using_gridsearchcv(X, y)
```

Out[63]:

	model	best_score	best_params
0	linear_regression	0.818354	{}
1	lasso	0.687429	{'alpha': 1, 'selection': 'cyclic'}
2	decision_tree	0.730355	{'criterion': 'friedman_mse', 'splitter': 'best'}

In [64]:
```
def predict_price(location,sqft,bath,bhk):
    loc_index = np.where(X.columns==location)[0][0]

    x = np.zeros(len(X.columns))
    x[0] = sqft
    x[1] = bath
    x[2] = bhk
    if loc_index >= 0:
        x[loc_index] = 1

    return lr_clf.predict([x])[0]
```

In [65]:
```
predict_price('1st Phase JP Nagar',1000, 2, 2)
```

Out[65]:
```
83.49904677172415
```

In [66]:
```
predict_price('1st Phase JP Nagar',1000, 3, 3)
```

Out[66]:
```
86.80519395199
```

In [67]:
```
predict_price('Indira Nagar',1000, 2, 2)
```

Out[67]:
```
181.27815484006965
```

In [ ]:
```

```