ASSESMENT 2

21BCE0516

ANUSHKA

anushka.2021a@vitstudent.ac.in (mailto:anushka.2021a@vitstudent.ac.in)

1. Download the dataset: House Price India dataset is downloaded.

In [4]:
```
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import rcParams
import seaborn as sns
```

2. Load The dataset

In [5]:
```
df = pd.read_csv('House Price India.csv')
df.head()
```

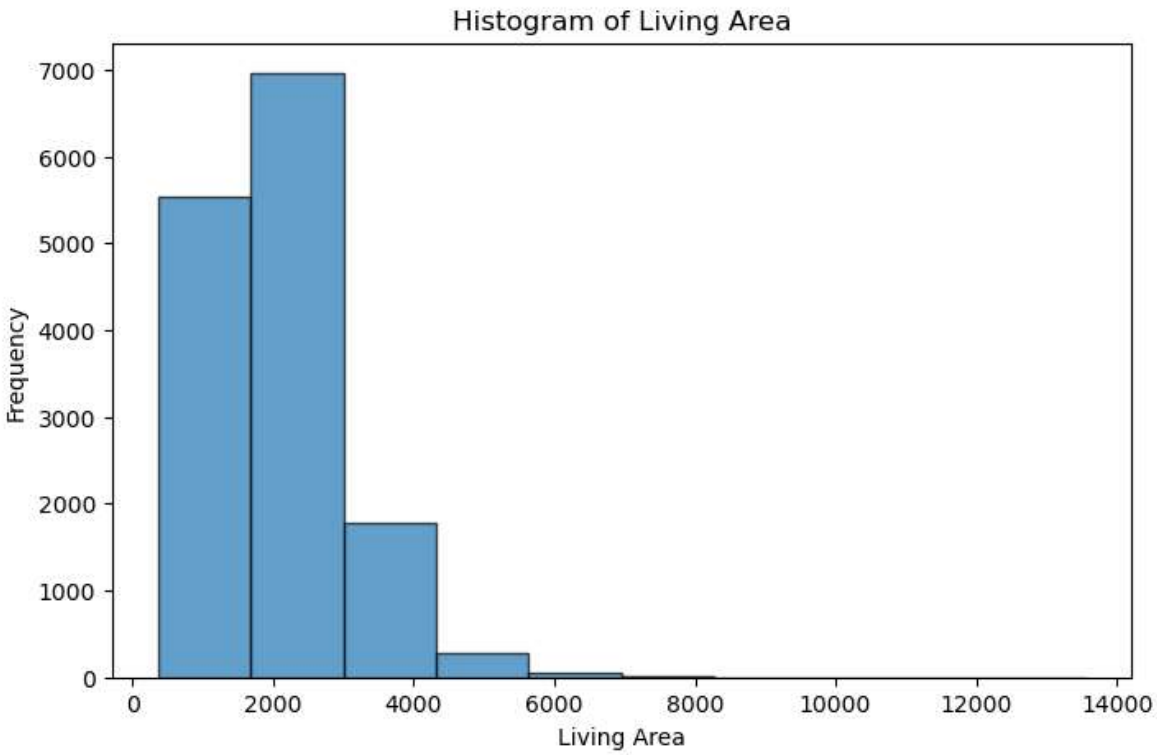
Out[5]:

	id	Date	number of bedrooms	number of bathrooms	living area	lot area	number of floors	waterfront present	number of views	condi of ho
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	

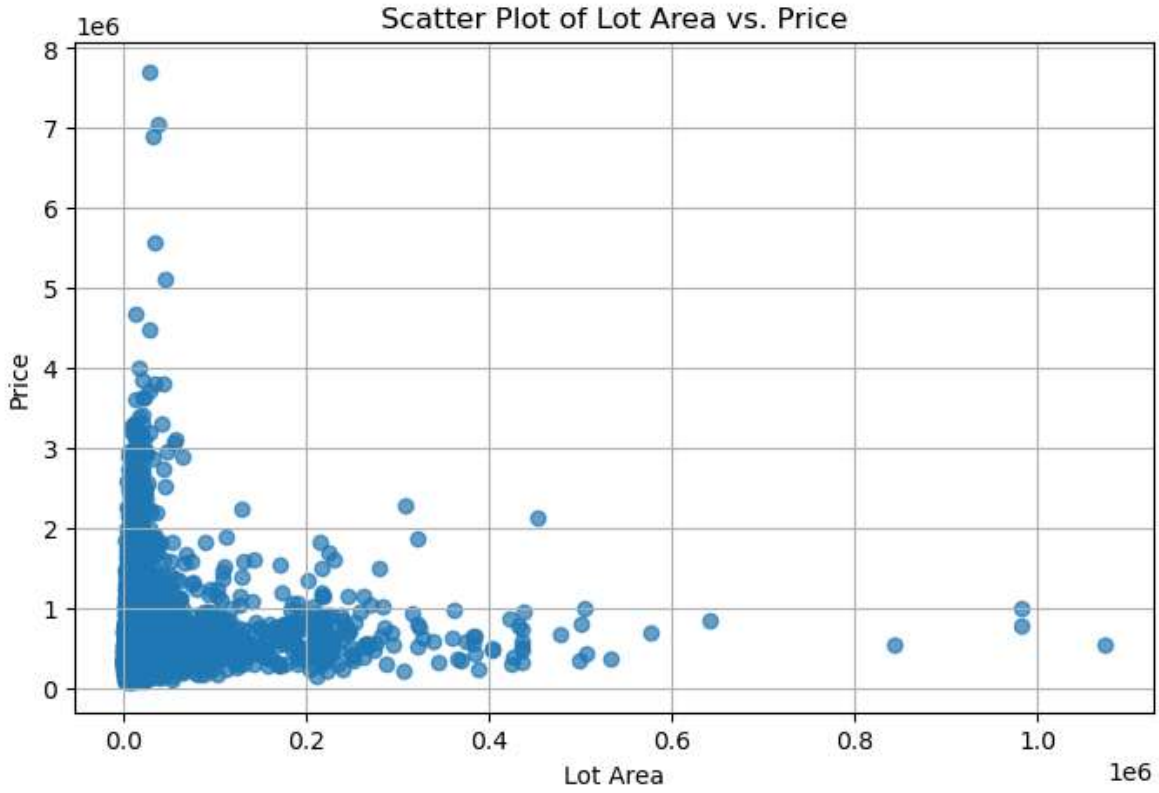
5 rows × 23 columns

3. Perform the Below Visualizations. □ Univariate Analysis □ Bi - Variate Analysis □ Multivariate Analysis

In [6]:
```
# Univariate Analysis (Analysis on single feature 'living area')
plt.figure(figsize=(8, 5))
plt.hist(df['living area'], bins=10, edgecolor='k', alpha=0.7)
plt.title('Histogram of Living Area')
plt.xlabel('Living Area')
plt.ylabel('Frequency')
plt.show()
```

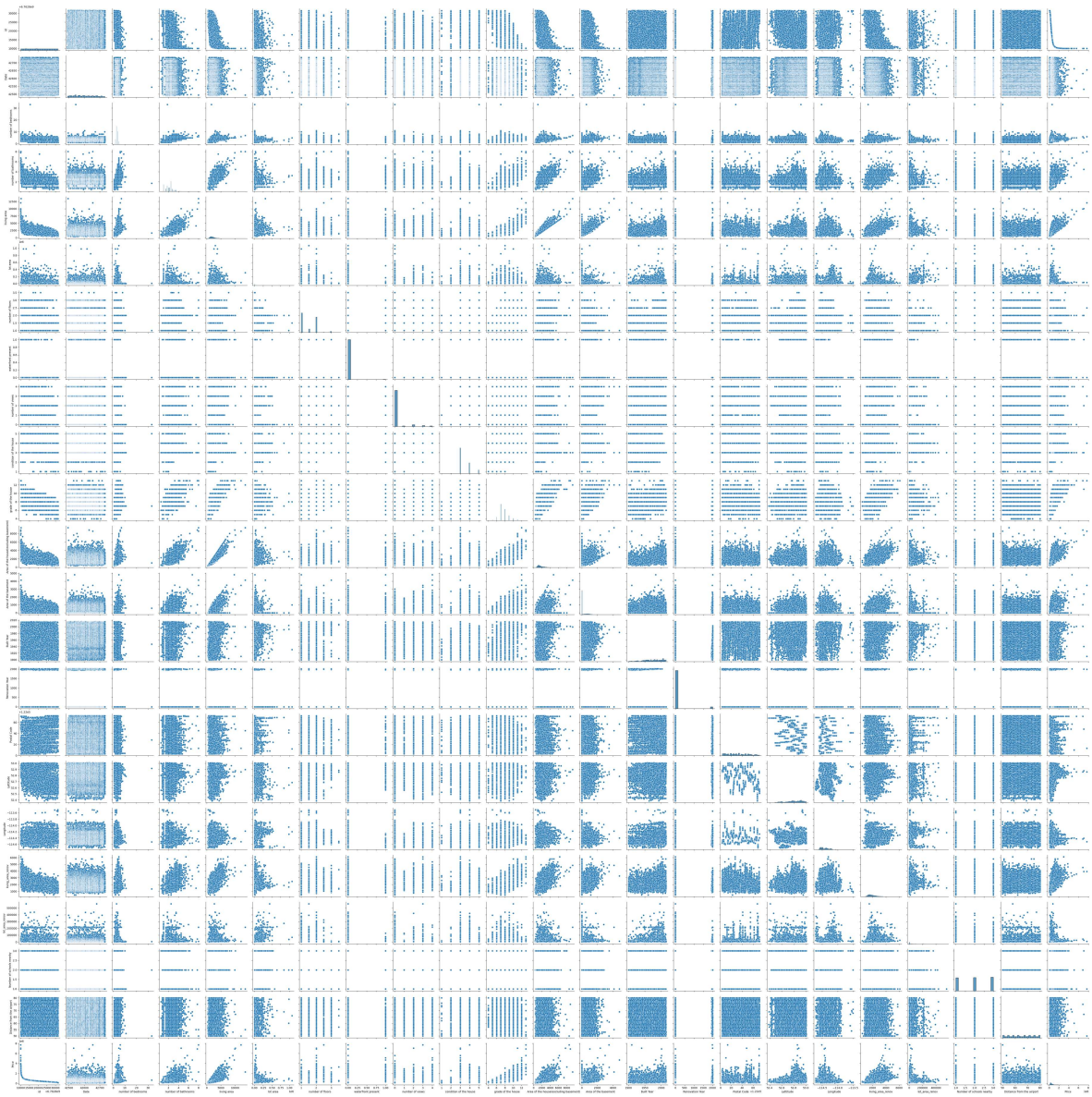


In [7]:
```
#Bi-variate analysis
plt.figure(figsize=(8, 5))
plt.scatter(df['lot area'], df['Price'], alpha=0.7)
plt.title('Scatter Plot of Lot Area vs. Price')
plt.xlabel('Lot Area')
plt.ylabel('Price')
plt.grid(True)  # Add grid lines
plt.show()
```



In [11]:
```
# Multivariate analysis
sns.pairplot(df)
```

Out[11]:
```
<seaborn.axisgrid.PairGrid at 0x1f41fc01f50>
```



4. Perform descriptive statistics on the dataset

In [8]:
```
df.describe()
```

Out[8]:

	id	Date	number of bedrooms	number of bathrooms	living area	lot area	14
count	1.462000e+04	14620.000000	14620.000000	14620.000000	14620.000000	1.462000e+04	
mean	6.762821e+09	42604.538646	3.379343	2.129583	2098.262996	1.509328e+04	
std	6.237575e+03	67.347991	0.938719	0.769934	928.275721	3.791962e+04	
min	6.762810e+09	42491.000000	1.000000	0.500000	370.000000	5.200000e+02	
25%	6.762815e+09	42546.000000	3.000000	1.750000	1440.000000	5.010750e+03	
50%	6.762821e+09	42600.000000	3.000000	2.250000	1930.000000	7.620000e+03	
75%	6.762826e+09	42662.000000	4.000000	2.500000	2570.000000	1.080000e+04	
max	6.762832e+09	42734.000000	33.000000	8.000000	13540.000000	1.074218e+06	

8 rows × 23 columns

5. Handle the Missing values.

In [9]:
```
df.isnull().any() #check n if no null values we dont have to handle it.
```

Out[9]:
```
id                                      False
Date                                    False
number of bedrooms                      False
number of bathrooms                     False
living area                             False
lot area                                False
number of floors                        False
waterfront present                      False
number of views                         False
condition of the house                  False
grade of the house                      False
Area of the house(excluding basement)   False
Area of the basement                    False
Built Year                              False
Renovation Year                         False
Postal Code                             False
Lattitude                               False
Longitude                               False
living_area_renov                       False
lot_area_renov                          False
Number of schools nearby                False
Distance from the airport               False
Price                                   False
dtype: bool
```