Assignment 2

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

2. Load the Dataset

In [1]:

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns
```

In [5]:

```
Housing = pd.read_csv('House Price India.csv')
```

In [6]:

```
Housing
```

Out[6]:

throoms	living_area	lot_area	number_of_floors	waterfront_present	number_of_views	conditi
2.50	3650	9050	2.0	0	4	
2.50	2920	4000	1.5	0	0	
2.75	2910	9480	1.5	0	0	
2.50	3310	42998	2.0	0	0	
2.00	2710	4500	1.5	0	0	
...	...	...	...	...	...	
1.50	1556	20000	1.0	0	0	
2.00	1680	7000	1.5	0	0	
1.00	1070	6120	1.0	0	0	
1.00	1030	6621	1.0	0	0	
1.00	900	4770	1.0	0	0	

3. Perform the Below Visualizations.

● Univariate Analysis

- **Bi - Variate Analysis**

- **Multivariate Analysis**

In [9]:

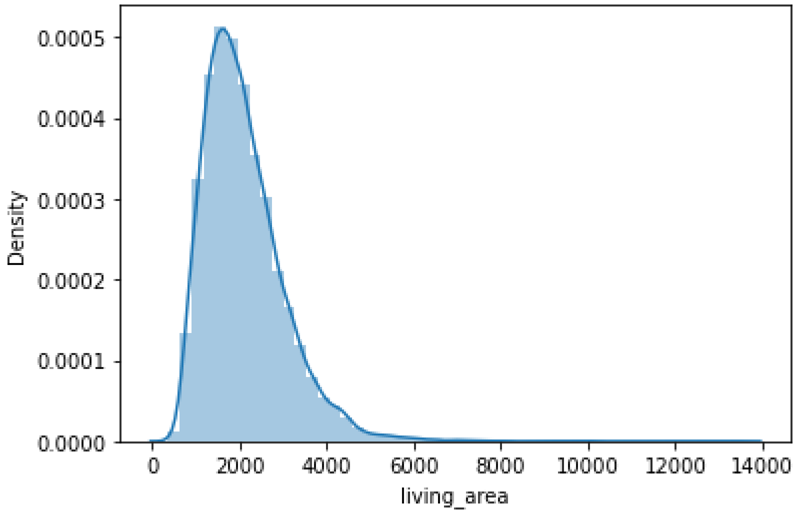
```
# Univariate Analysis (Analysis on single feature 'living area')

sns.distplot(Housing['living_area'])
```

```
C:\Users\Harsh\anaconda3\lib\site-packages\seaborn\distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figur
e-level function with similar flexibility) or `histplot` (an axes-level f
unction for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[9]:

```
<AxesSubplot:xlabel='living_area', ylabel='Density'>
```



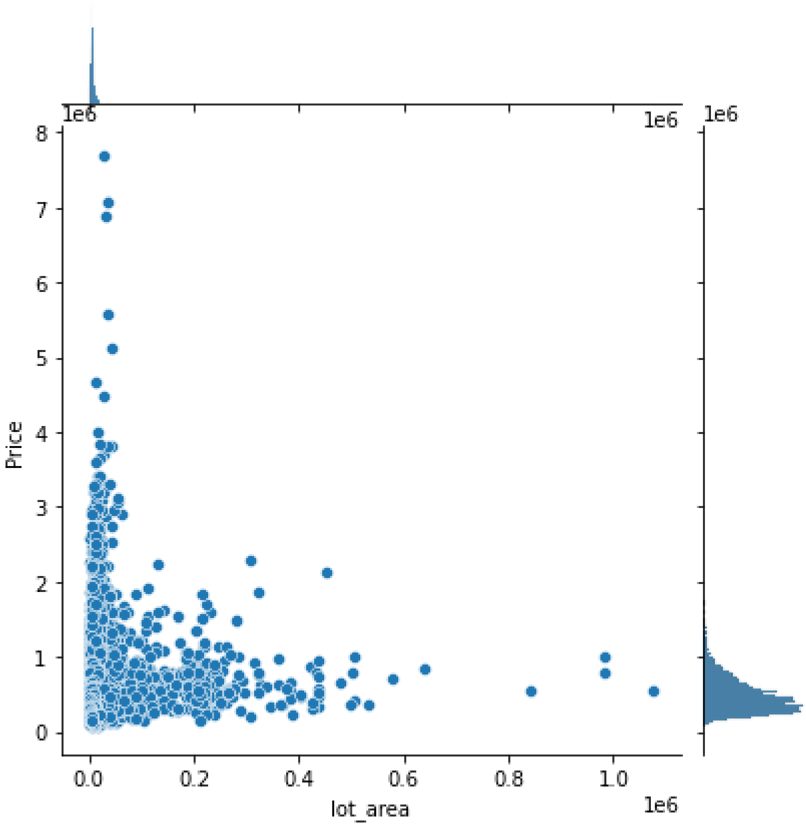
In [10]:

```
# Bivariate Analysis (Comparision between 'lot_area' feature and 'Price')

sns.jointplot(x='lot_area',y='Price',data=Housing)
```

Out[10]:

```
<seaborn.axisgrid.JointGrid at 0x22460fced30>
```



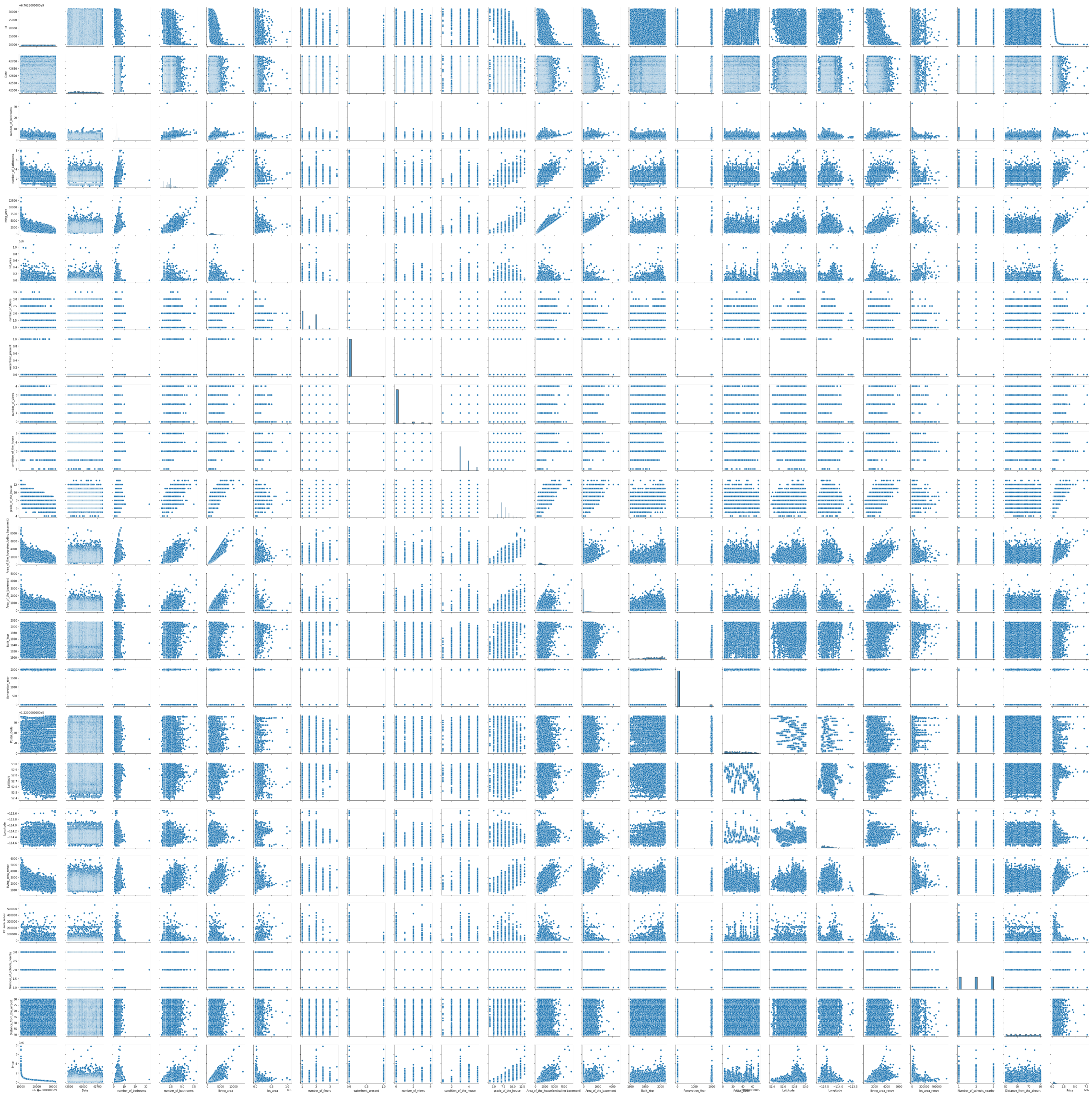
In [11]:

```
# Multivariate analysis

sns.pairplot(Housing)
```

Out[11]:

```
<seaborn.axisgrid.PairGrid at 0x224634d3670>
```



4. Perform descriptive statistics on the dataset.

In [12]:

```
Housing.describe()
```

Out[12]:

	id	Date	number_of_bedrooms	number_of_bathrooms	living_ar
count	1.462000e+04	14620.000000	14620.000000	14620.000000	14620.0000
mean	6.762821e+09	42604.538646	3.379343	2.129583	2098.2629
std	6.237575e+03	67.347991	0.938719	0.769934	928.2757
min	6.762810e+09	42491.000000	1.000000	0.500000	370.0000
25%	6.762815e+09	42546.000000	3.000000	1.750000	1440.0000
50%	6.762821e+09	42600.000000	3.000000	2.250000	1930.0000
75%	6.762826e+09	42662.000000	4.000000	2.500000	2570.0000
max	6.762832e+09	42734.000000	33.000000	8.000000	13540.0000

8 rows × 23 columns

5. Handle the Missing values.

In [14]:

```
#Checking is there any null values in our dataset
Housing.isnull().any()
```

Out[14]:

```
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
```

No null values in our dataset

In [ ]: