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

In [3]:
```
#step 2
df = pd.read_csv("Housing.csv")
print("Data set is:")
print(df)
```

```
Data set is:
        price  area  bedrooms  bathrooms  stories mainroad guestroom basement  \
0    13300000  7420         4          2        3      yes        no       no
1    12250000  8960         4          4        4      yes        no       no
2    12250000  9960         3          2        2      yes        no      yes
3    12215000  7500         4          2        2      yes        no      yes
4    11410000  7420         4          1        2      yes       yes      yes
..        ...   ...       ...        ...      ...      ...       ...      ...
540   1820000  3000         2          1        1      yes        no      yes
541   1767150  2400         3          1        1       no        no       no
542   1750000  3620         2          1        1      yes        no       no
543   1750000  2910         3          1        1       no        no       no
544   1750000  3850         3          1        2      yes        no       no

    hotwaterheating airconditioning  parking prefarea furnishingstatus
0                no             yes        2      yes        furnished
1                no             yes        3       no        furnished
2                no              no        2      yes   semi-furnished
3                no             yes        3      yes        furnished
4                no             yes        2       no        furnished
..              ...             ...      ...      ...              ...
540              no              no        2       no      unfurnished
541              no              no        0       no   semi-furnished
542              no              no        0       no      unfurnished
543              no              no        0       no        furnished
544              no              no        0       no      unfurnished

[545 rows x 13 columns]
```

In [4]:
```
# step 3
print(df.head())
```

```
      price  area  bedrooms  bathrooms  stories mainroad guestroom basement  \
0  13300000  7420         4          2        3      yes        no       no
1  12250000  8960         4          4        4      yes        no       no
2  12250000  9960         3          2        2      yes        no      yes
3  12215000  7500         4          2        2      yes        no      yes
4  11410000  7420         4          1        2      yes       yes      yes

  hotwaterheating airconditioning  parking prefarea furnishingstatus
0              no             yes        2      yes        furnished
1              no             yes        3       no        furnished
2              no              no        2      yes   semi-furnished
3              no             yes        3      yes        furnished
4              no             yes        2       no        furnished
```

In [5]:
```
print(df.columns)
```

```
Index(['price', 'area', 'bedrooms', 'bathrooms', 'stories', 'mainroad',
       'guestroom', 'basement', 'hotwaterheating', 'airconditioning',
       'parking', 'prefarea', 'furnishingstatus'],
      dtype='object')
```

In [6]:
```
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 545 entries, 0 to 544
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   price             545 non-null    int64 
 1   area              545 non-null    int64 
 2   bedrooms          545 non-null    int64 
 3   bathrooms         545 non-null    int64 
 4   stories           545 non-null    int64 
 5   mainroad          545 non-null    object
 6   guestroom         545 non-null    object
 7   basement          545 non-null    object
 8   hotwaterheating   545 non-null    object
 9   airconditioning   545 non-null    object
 10  parking           545 non-null    int64 
 11  prefarea          545 non-null    object
 12  furnishingstatus  545 non-null    object
dtypes: int64(6), object(7)
memory usage: 55.5+ KB
None
```

In [7]:
```
#  step 4
# Statistical Summary
print(df.describe())
```

```
              price          area    bedrooms   bathrooms     stories  \
count  5.450000e+02    545.000000  545.000000  545.000000  545.000000
mean   4.766729e+06   5150.541284    2.965138    1.286239    1.805505
std    1.870440e+06   2170.141023    0.738064    0.502470    0.867492
min    1.750000e+06   1650.000000    1.000000    1.000000    1.000000
25%    3.430000e+06   3600.000000    2.000000    1.000000    1.000000
50%    4.340000e+06   4600.000000    3.000000    1.000000    2.000000
75%    5.740000e+06   6360.000000    3.000000    2.000000    2.000000
max    1.330000e+07  16200.000000    6.000000    4.000000    4.000000

          parking
count  545.000000
mean     0.693578
std      0.861586
min      0.000000
25%      0.000000
50%      0.000000
75%      1.000000
max      3.000000
```

In [8]:
```
#step 5
# null values
print(df.isnull().sum())
```

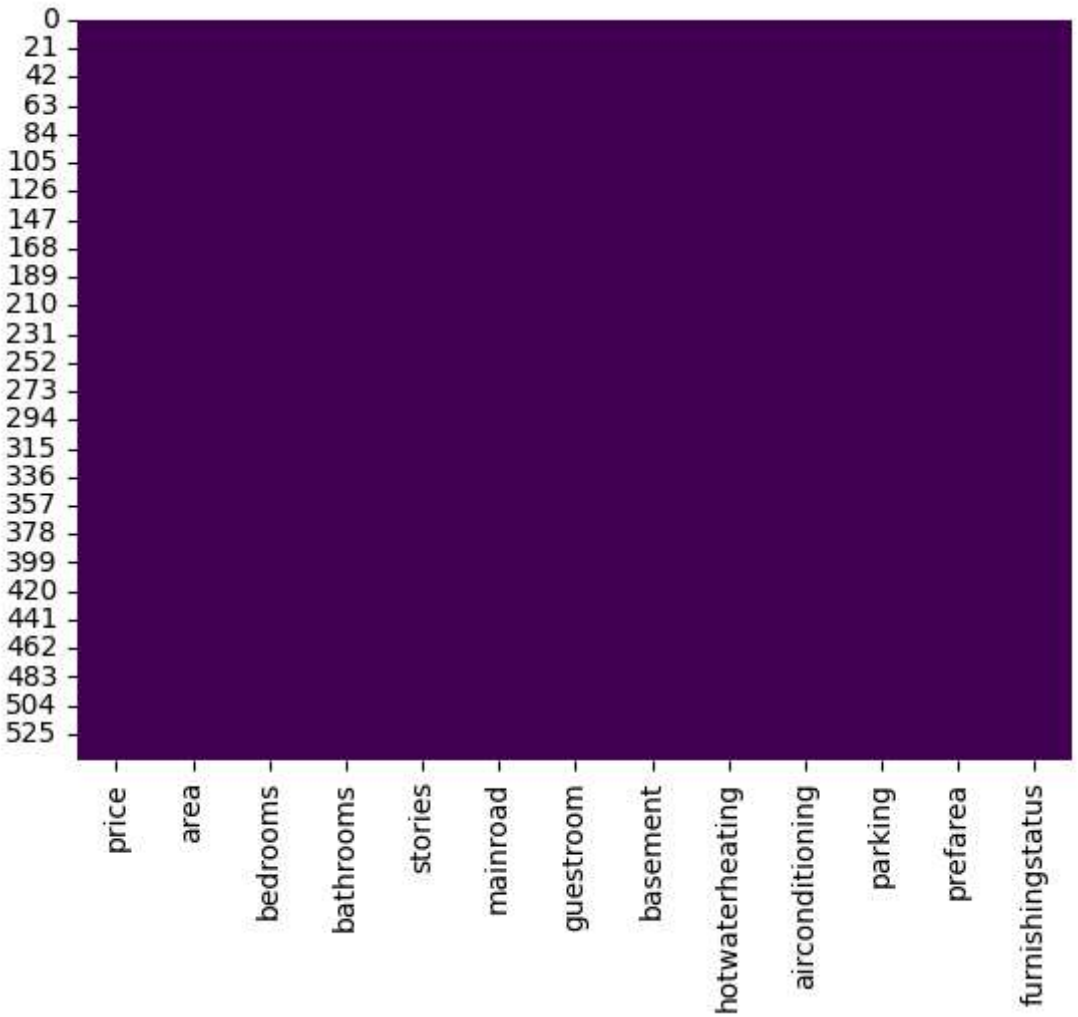
```
price             0
area              0
bedrooms          0
bathrooms         0
stories           0
mainroad          0
guestroom         0
basement          0
hotwaterheating   0
airconditioning   0
parking           0
prefarea          0
furnishingstatus  0
dtype: int64
```

In [9]:
```
# step 6
#  Data Types
print(df.dtypes)
```

```
price              int64
area               int64
bedrooms           int64
bathrooms          int64
stories            int64
mainroad          object
guestroom         object
basement          object
hotwaterheating   object
airconditioning   object
parking            int64
prefarea          object
furnishingstatus  object
dtype: object
```

In [12]:
```
# step 7: Check for Missing Values
print(df.isnull().sum())
sns.heatmap(df.isnull(), cbar=False, cmap="viridis")
plt.show()
```

```
price             0
area              0
bedrooms          0
bathrooms         0
stories           0
mainroad          0
guestroom         0
basement          0
hotwaterheating   0
airconditioning   0
parking           0
prefarea          0
furnishingstatus  0
dtype: int64
```



In [13]:
```
# step 8 : Step: Remove Duplicate Rows from Dataset
# Check for duplicate rows
duplicate_rows = df[df.duplicated()]
print(f"Number of duplicate rows: {duplicate_rows.shape[0]}")
print(duplicate_rows)
```

```
Number of duplicate rows: 0
Empty DataFrame
Columns: [price, area, bedrooms, bathrooms, stories, mainroad, guestroom, basement,
hotwaterheating, airconditioning, parking, prefarea, furnishingstatus]
Index: []
```

In [15]:
```python
# Remove duplicate rows
df_cleaned = df.drop_duplicates()
# Confirm removal
print(f"Shape after removing duplicates: {df_cleaned.shape}")
```

```
Shape after removing duplicates: (545, 13)
```

In [16]:
```python
# step 9:Check and Clean Column Names
print("Original columns:", df.columns.tolist())

# Strip whitespace from column names
df.columns = df.columns.str.strip()

print("Cleaned columns:", df.columns.tolist())
```

```
Original columns: ['price', 'area', 'bedrooms', 'bathrooms', 'stories', 'mainroad',
'guestroom', 'basement', 'hotwaterheating', 'airconditioning', 'parking', 'prefare
a', 'furnishingstatus']
Cleaned columns: ['price', 'area', 'bedrooms', 'bathrooms', 'stories', 'mainroad',
'guestroom', 'basement', 'hotwaterheating', 'airconditioning', 'parking', 'prefare
a', 'furnishingstatus']
```

In [17]:
```python
# Inspect the actual column names
print("Columns in DataFrame:", df.columns.tolist())
```

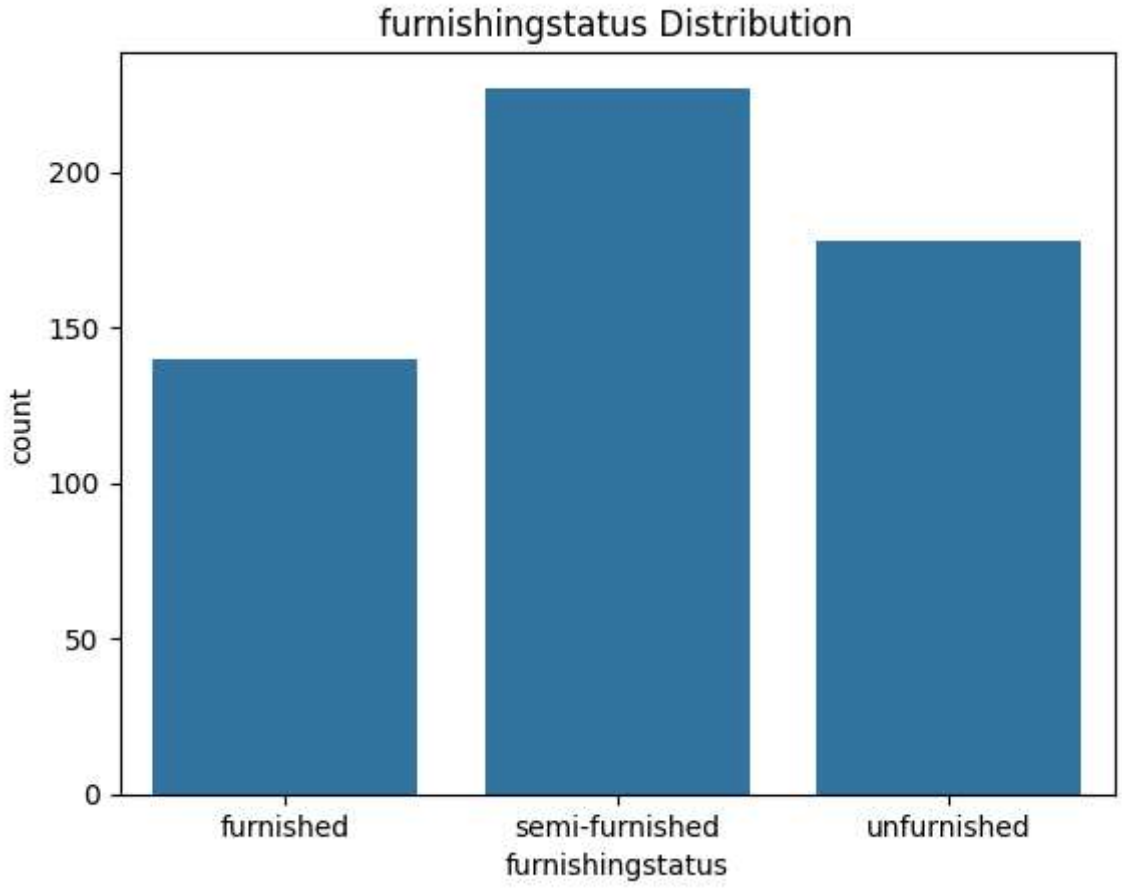
```
Columns in DataFrame: ['price', 'area', 'bedrooms', 'bathrooms', 'stories', 'mainroa
d', 'guestroom', 'basement', 'hotwaterheating', 'airconditioning', 'parking', 'prefa
rea', 'furnishingstatus']
```

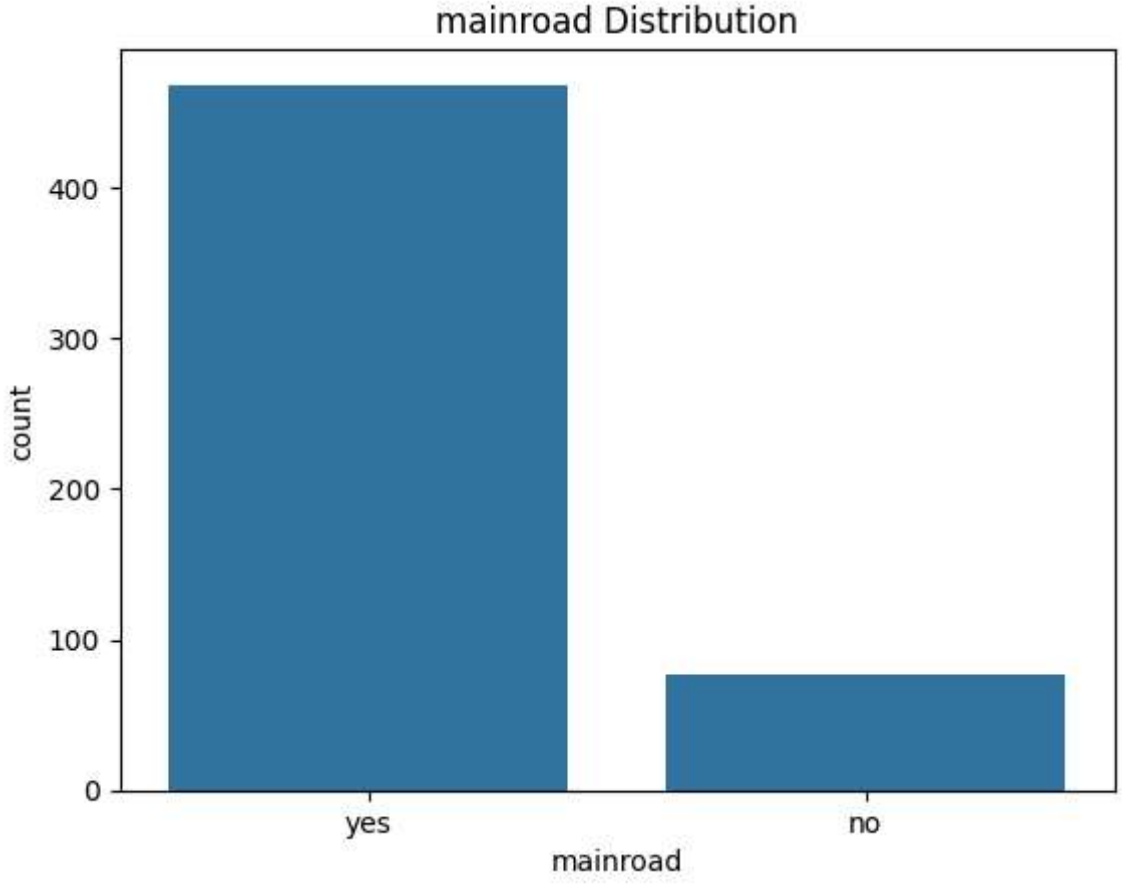
In [18]:
```python
#Clean column names
df.columns = df.columns.str.strip()  # remove leading/trailing spaces
df.columns = df.columns.str.replace('\n', '')  # remove newlines if any
print("Cleaned columns:", df.columns.tolist())
```

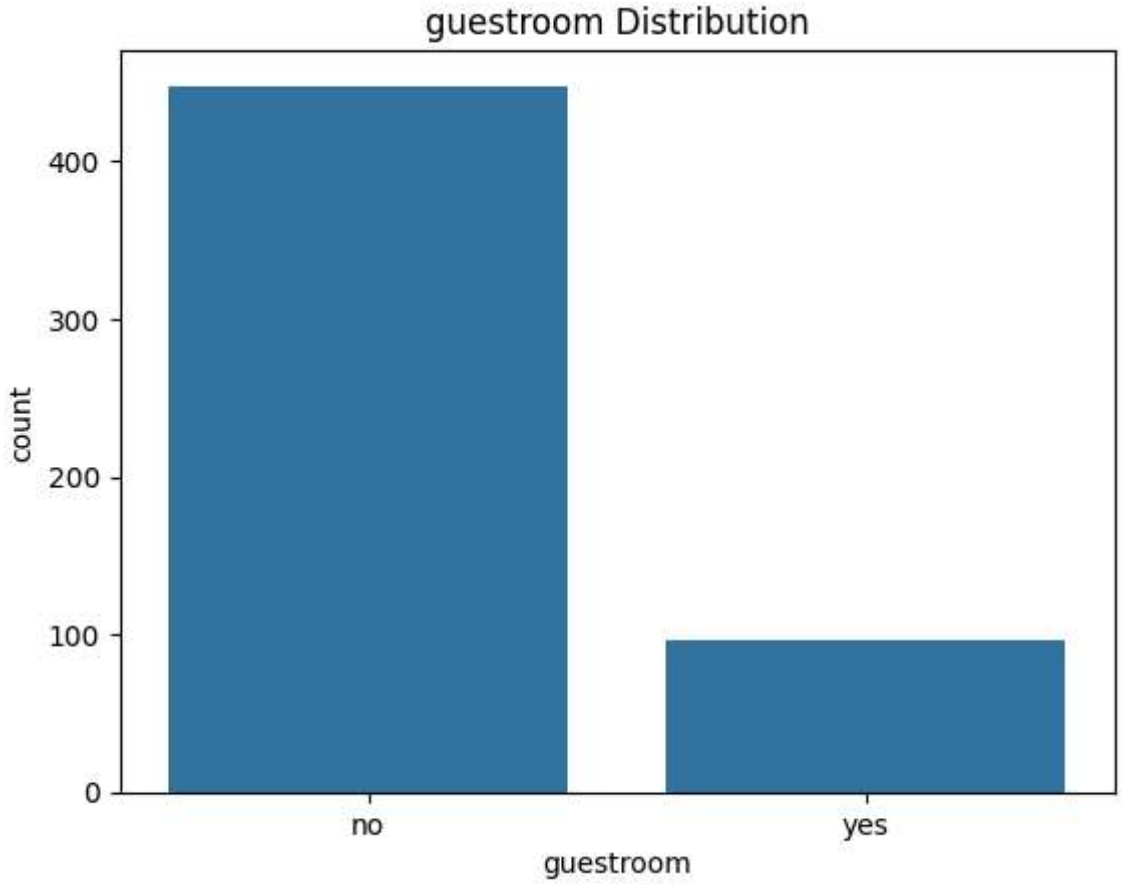
```
Cleaned columns: ['price', 'area', 'bedrooms', 'bathrooms', 'stories', 'mainroad',
'guestroom', 'basement', 'hotwaterheating', 'airconditioning', 'parking', 'prefare
a', 'furnishingstatus']
```

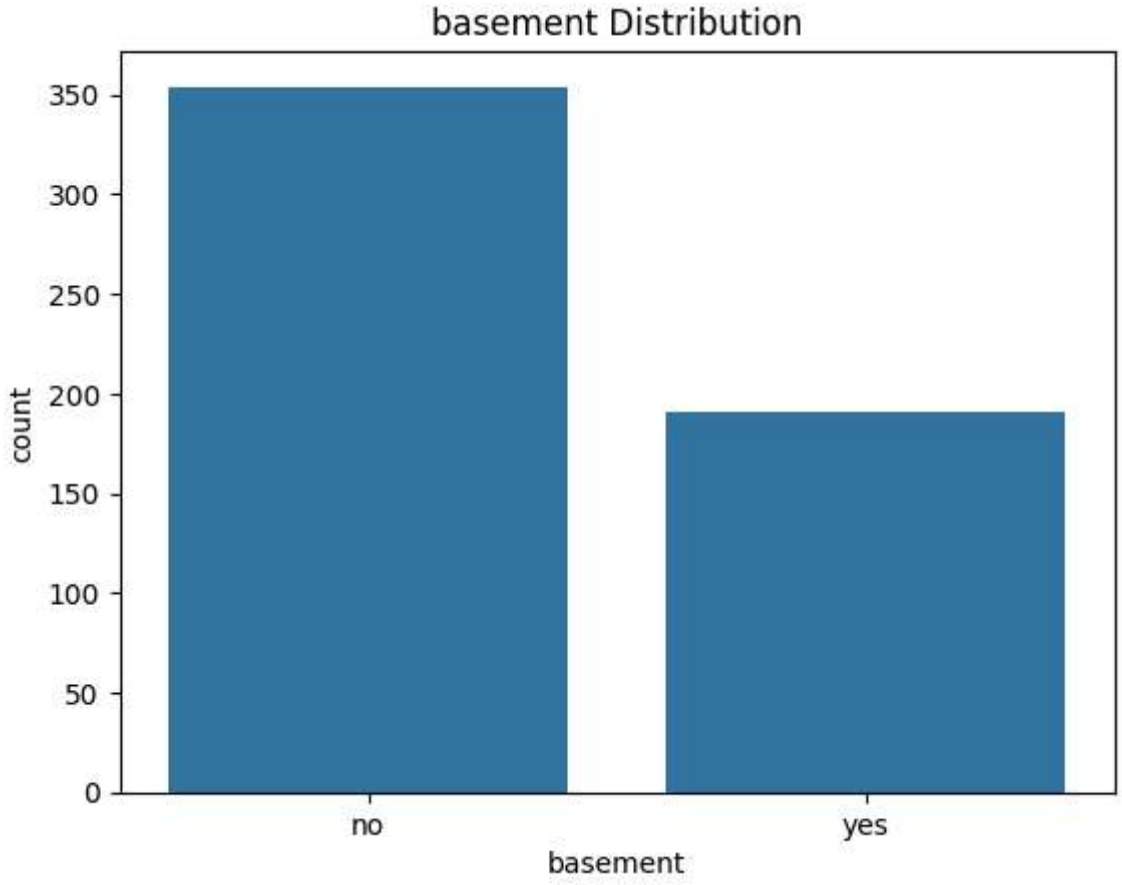
In [19]:
```python
# step 10 (Target Variable)
sns.histplot(df['price'], kde=True)
plt.title('Distribution of House Prices')
plt.show()
```

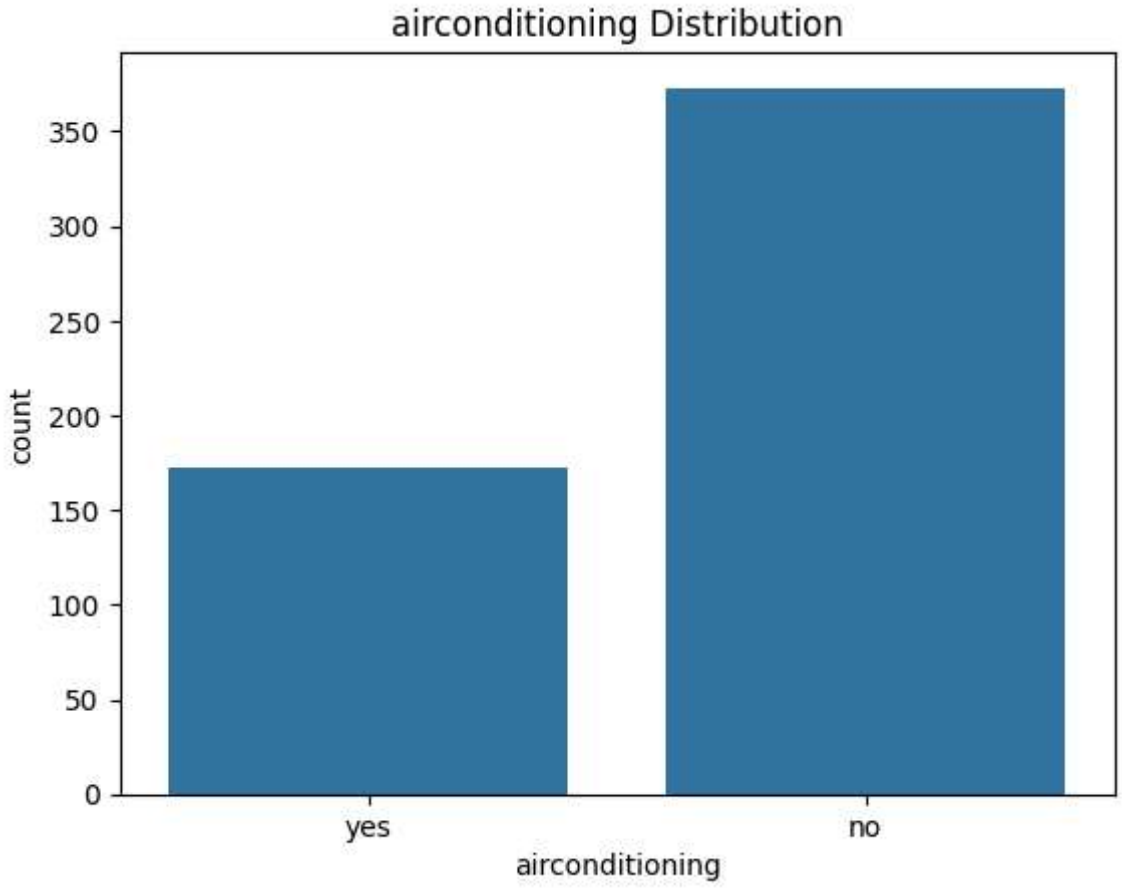
In [20]:
```
# Step 11:  (Categorical Features)
for col in ['furnishingstatus', 'mainroad', 'guestroom', 'basement', 'airconditioni
    sns.countplot(x=col, data=df)
    plt.title(f'{col} Distribution')
    plt.show()
```


furnishingstatus Distribution
count
0
50
100
150
200
furnished
semi-furnished
unfurnished
furnishingstatus

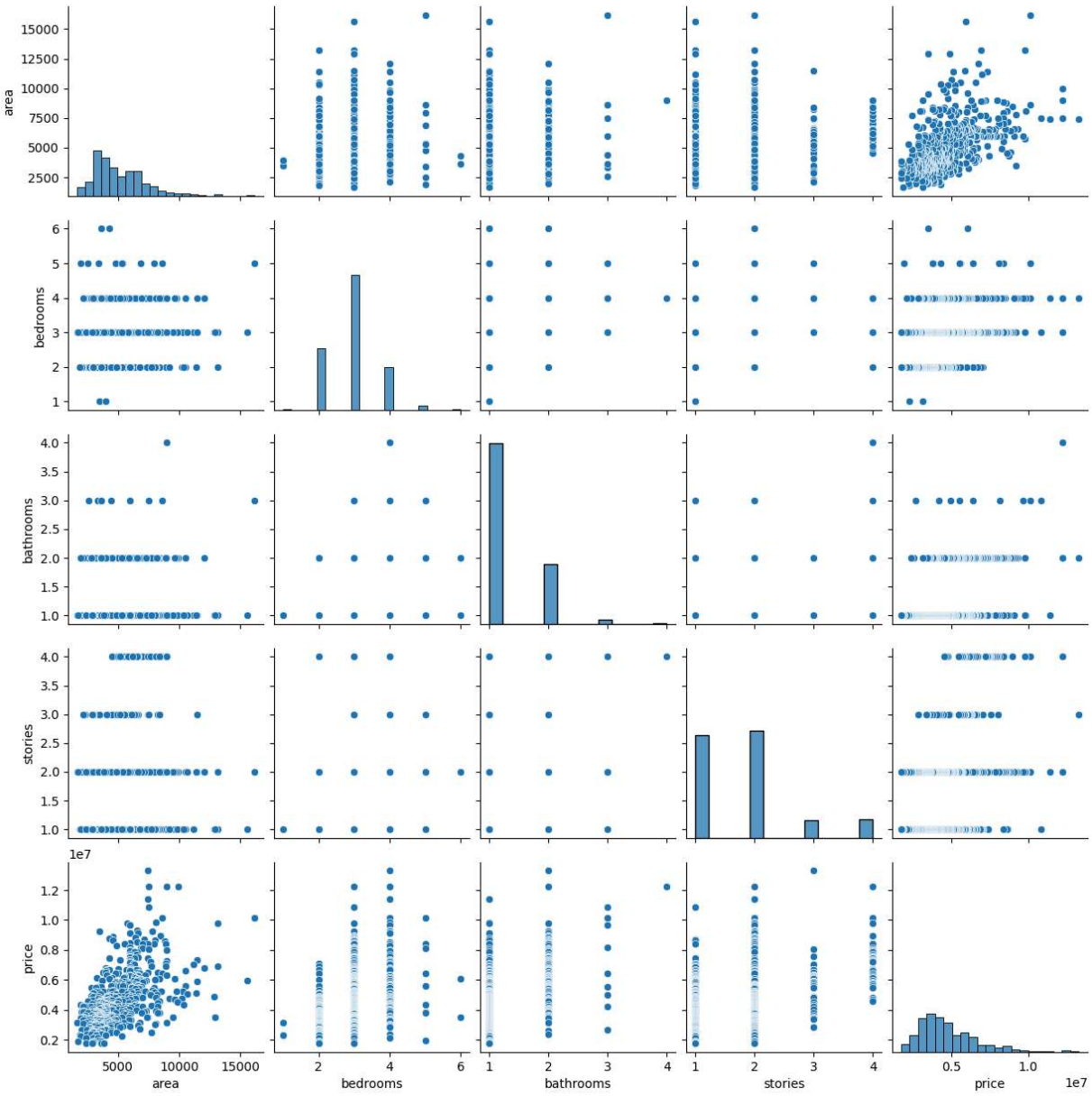

mainroad Distribution
count
0
100
200
300
400
yes
no
mainroad


guestroom Distribution
400
300
200
100
0
count
no
yes
guestroom

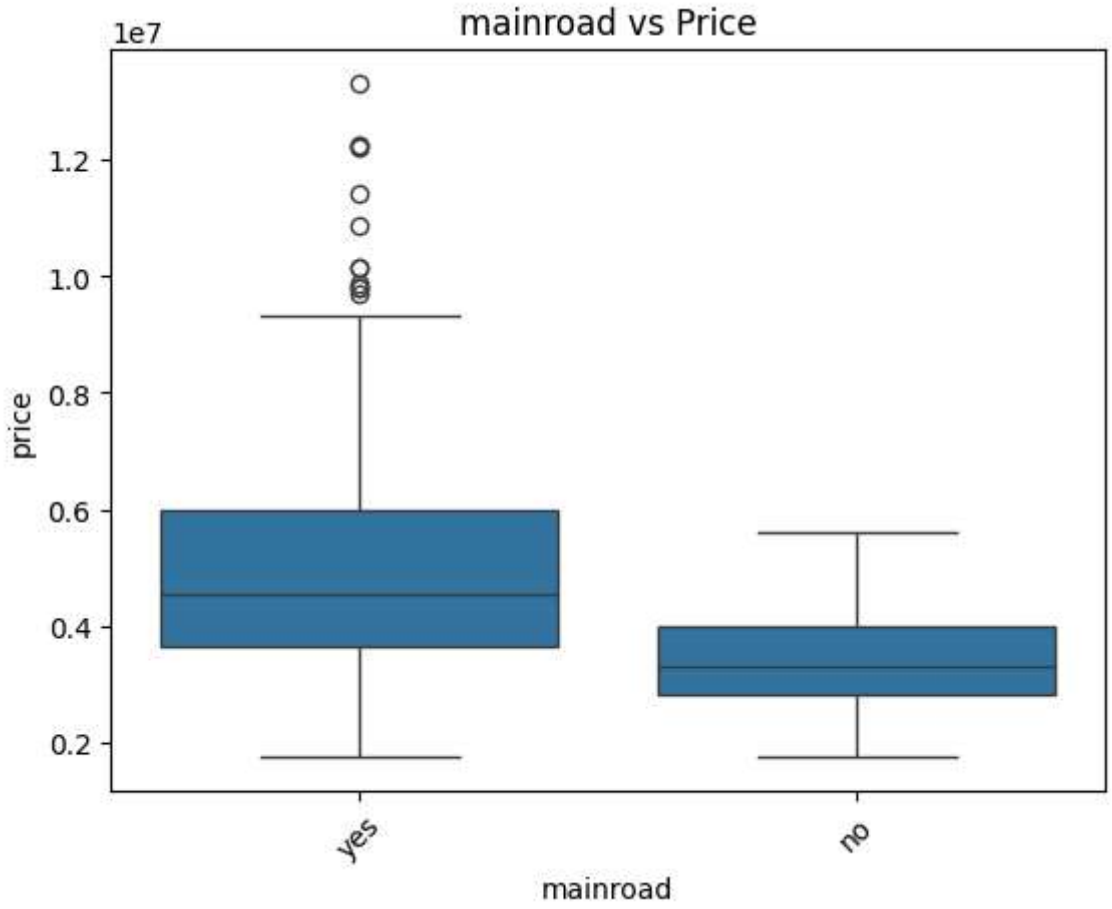

basement Distribution
350
300
250
200
150
100
50
0
count
no
yes
basement

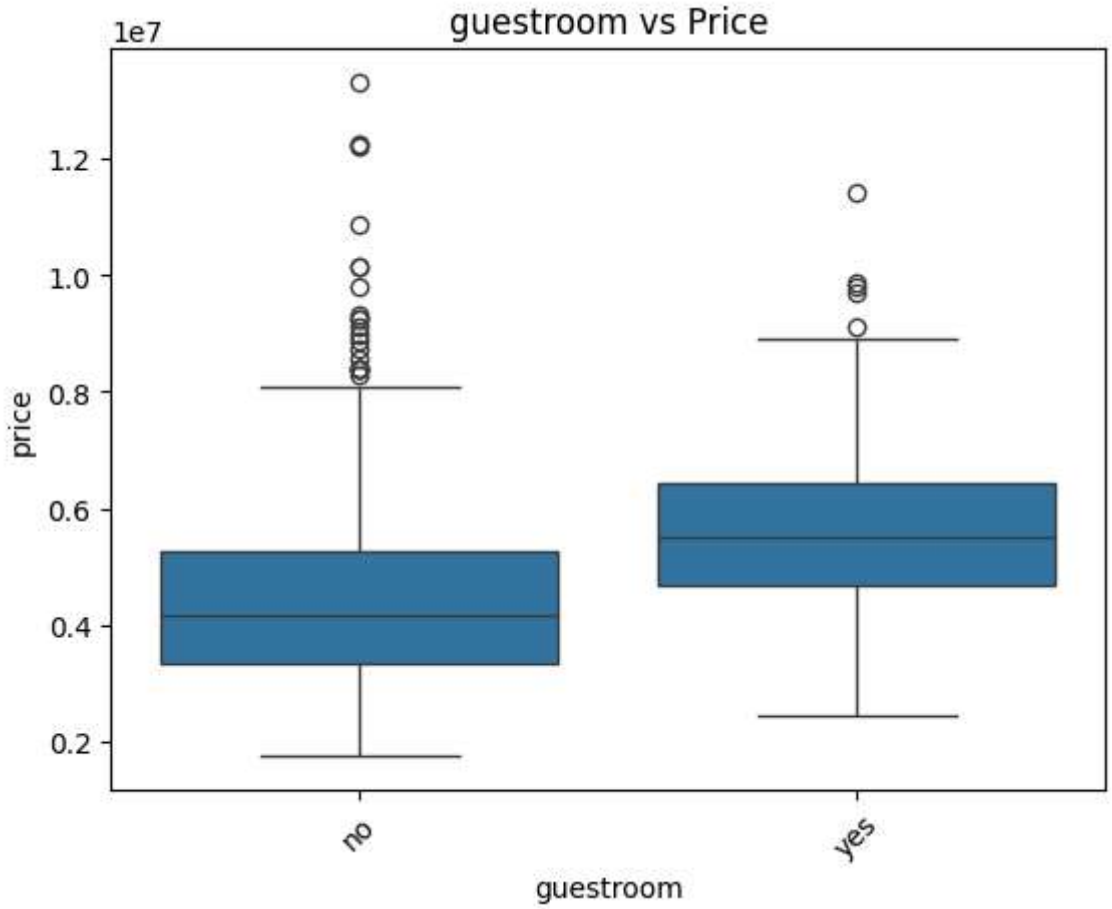


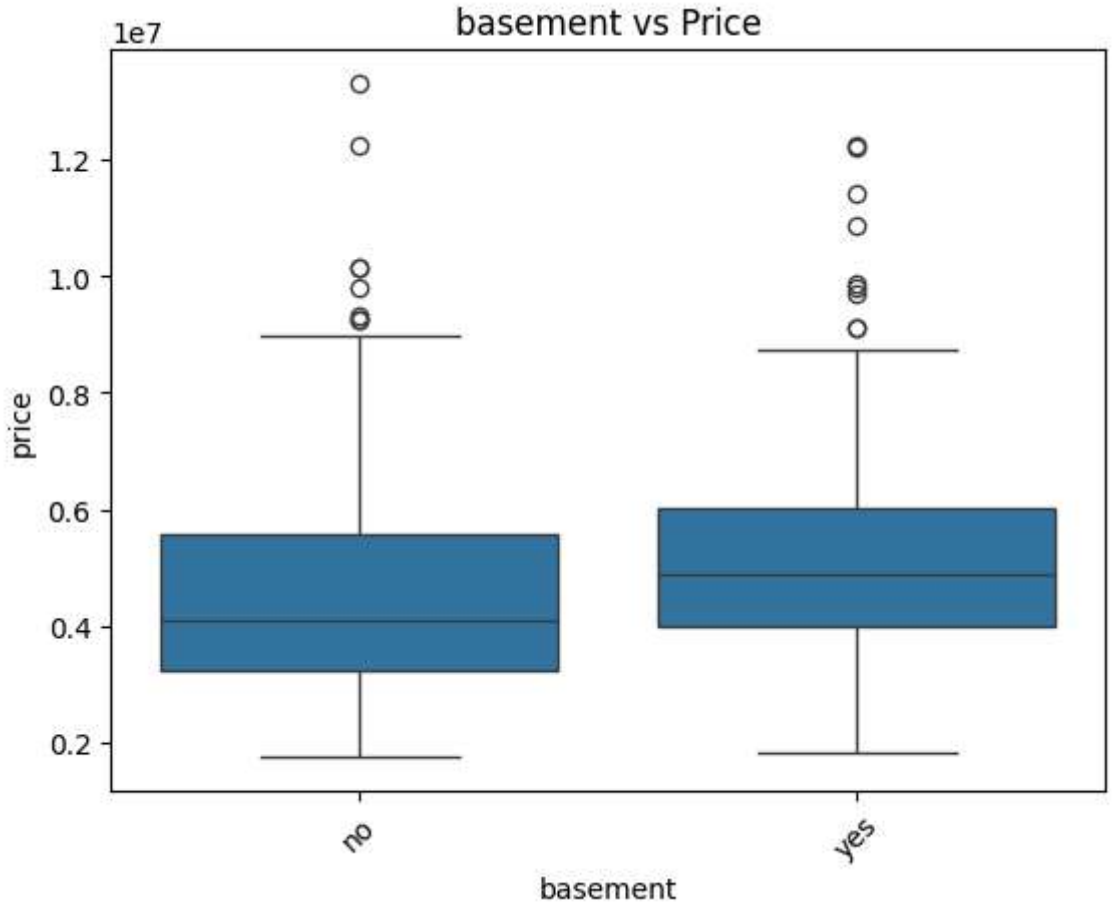
In [22]:
```
# step 12:
sns.pairplot(df[['area', 'bedrooms', 'bathrooms', 'stories', 'price']])
plt.show()
```

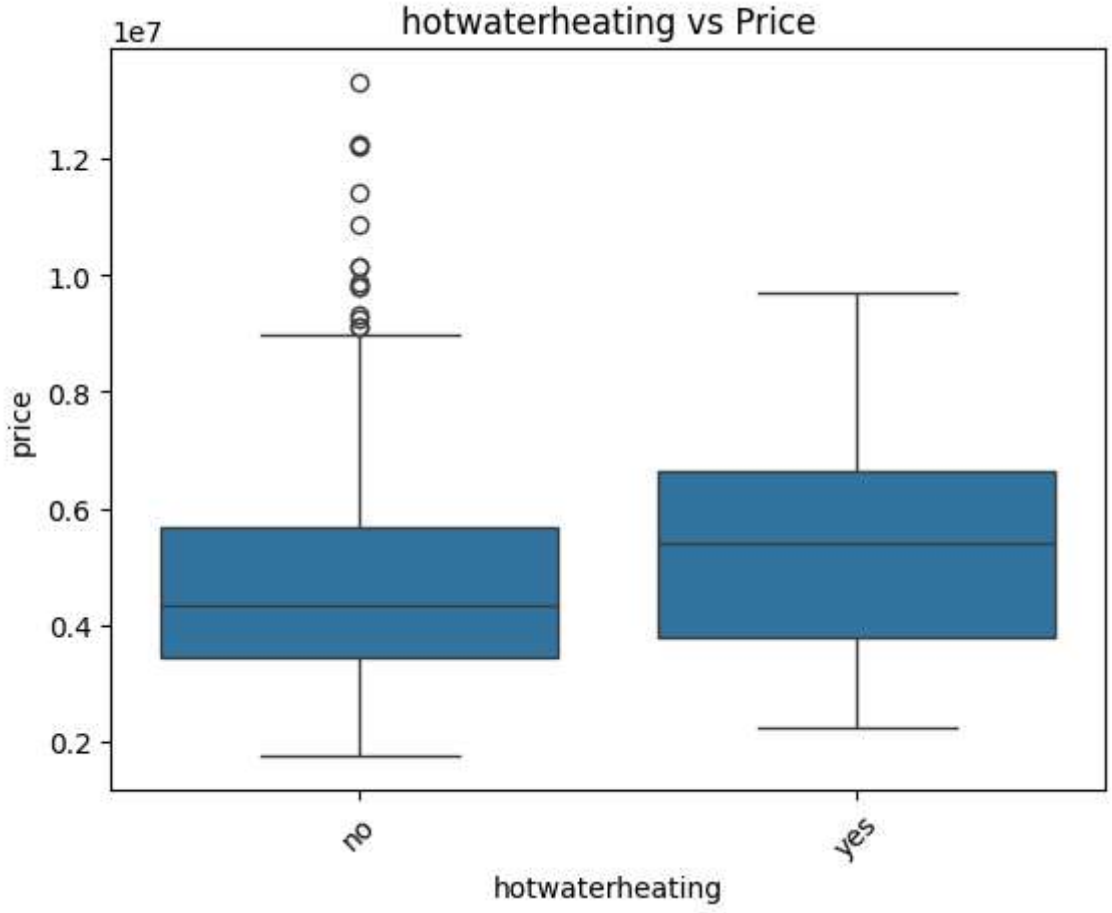


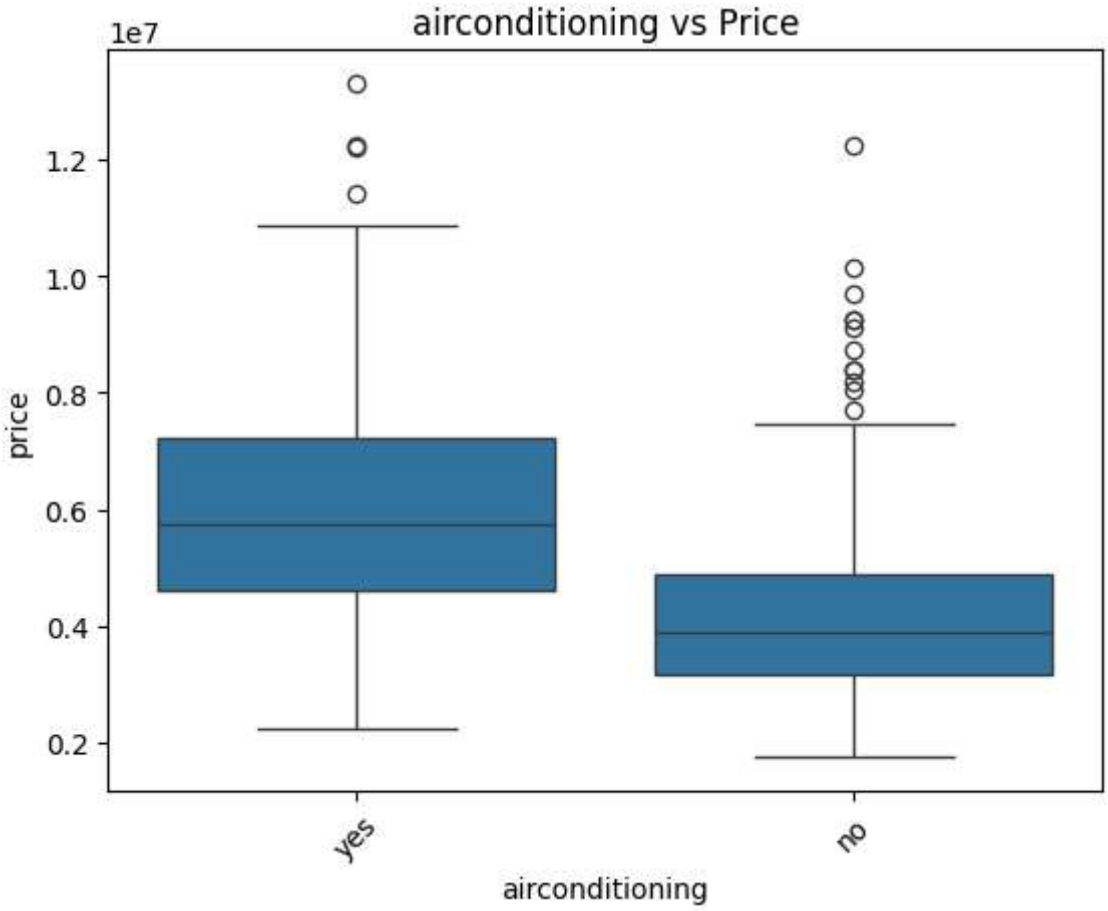
In [23]:
```
# step 13: Boxplot (Categorical vs Price)
for col in ['mainroad', 'guestroom', 'basement', 'hotwaterheating', 'airconditionin
    sns.boxplot(x=col, y='price', data=df)
    plt.title(f'{col} vs Price')
    plt.xticks(rotation=45)
    plt.show()
```

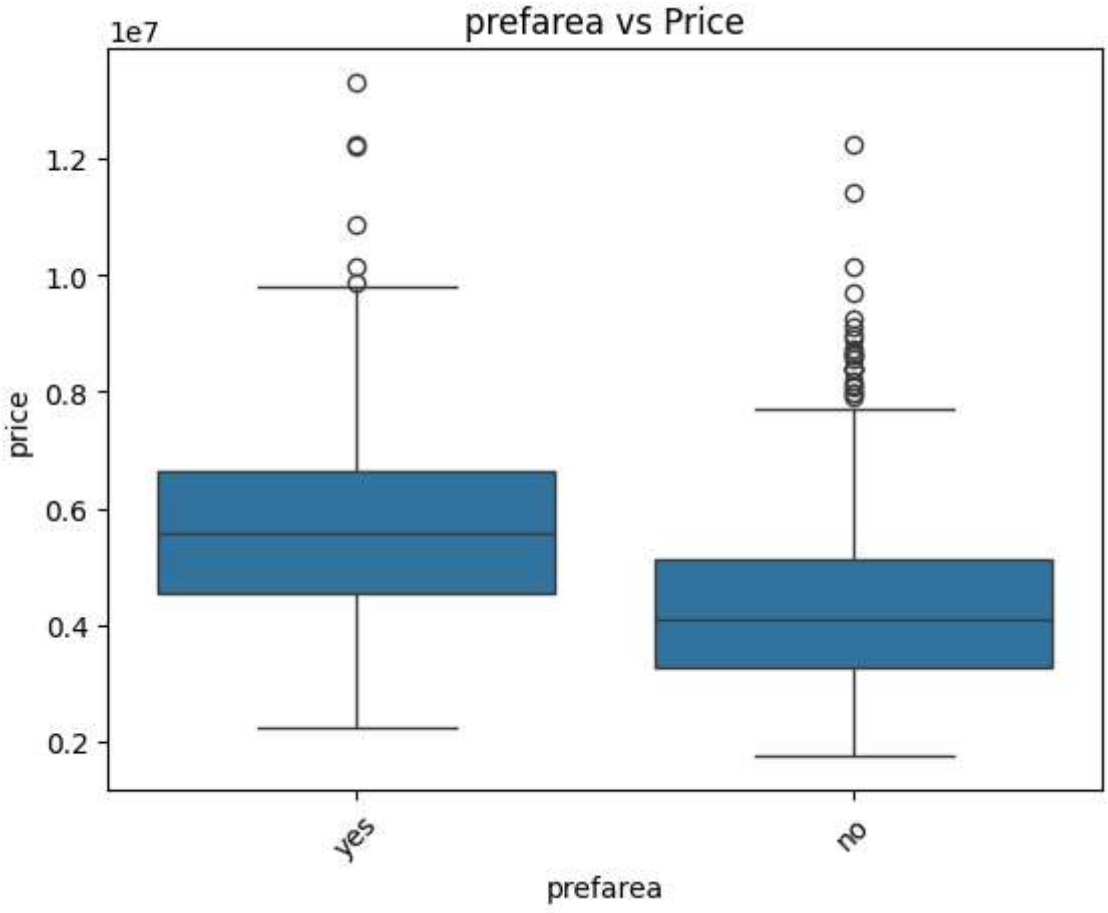

mainroad vs Price
1e7
1.2
1.0
0.8
0.6
0.4
0.2
price
yes
no
mainroad

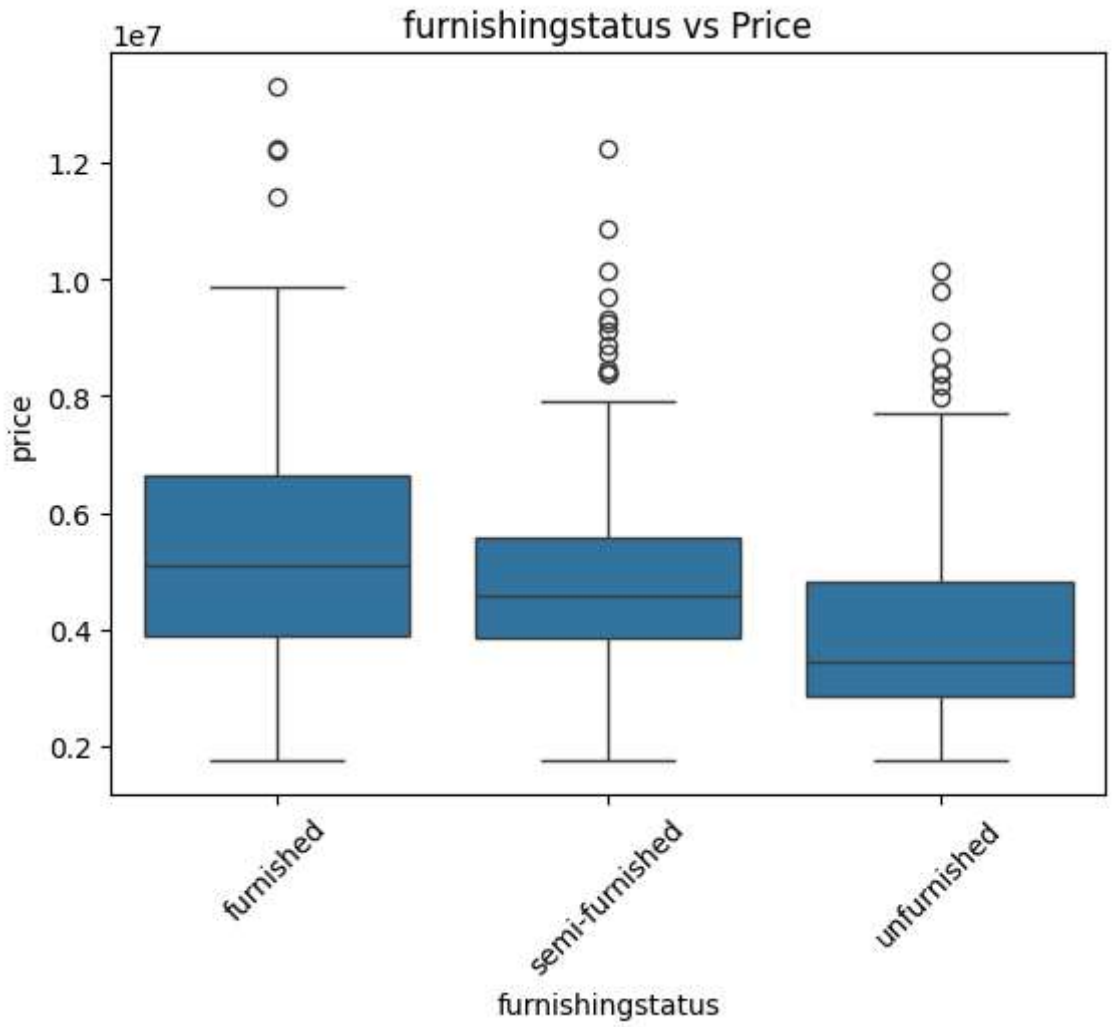

guestroom vs Price
1e7
1.2
1.0
0.8
0.6
0.4
0.2
price
no
yes
guestroom


basement vs Price
1e7
1.2
1.0
0.8
0.6
0.4
0.2
price
no
yes
basement


hotwaterheating vs Price
1e7
1.2
1.0
0.8
0.6
0.4
0.2
price
no
yes
hotwaterheating


airconditioning vs Price
1e7
1.2
1.0
0.8
0.6
0.4
0.2
price
yes
no
airconditioning


prefarea vs Price
1e7
1.2
1.0
0.8
0.6
0.4
0.2
price
yes
no
prefarea

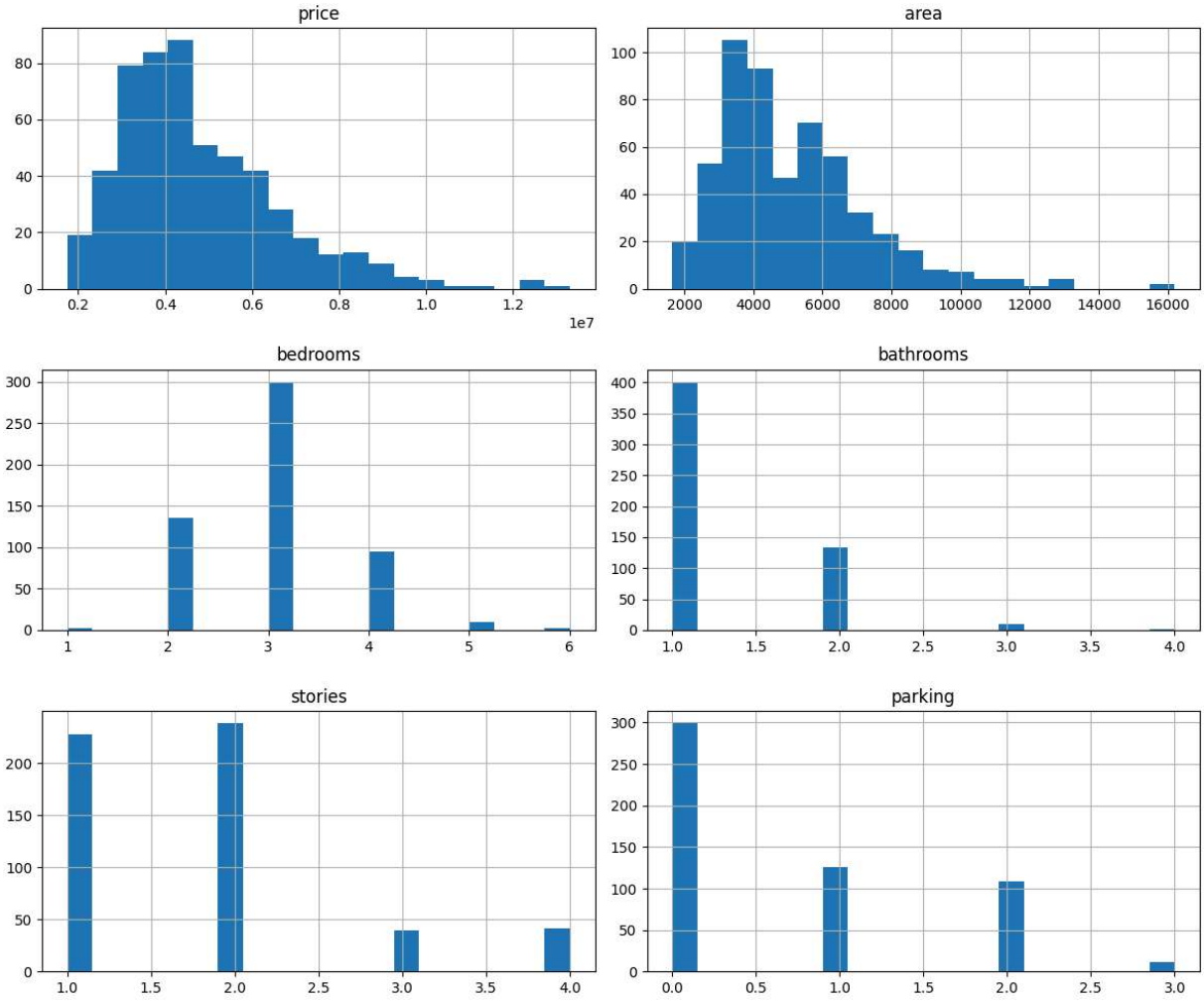


In [25]:
```python
# Step 15: Encode categorical variables
df_encoded = pd.get_dummies(df, drop_first=True)

#  Feature and Target split
X = df_encoded.drop('price', axis=1)
y = df_encoded['price']
```

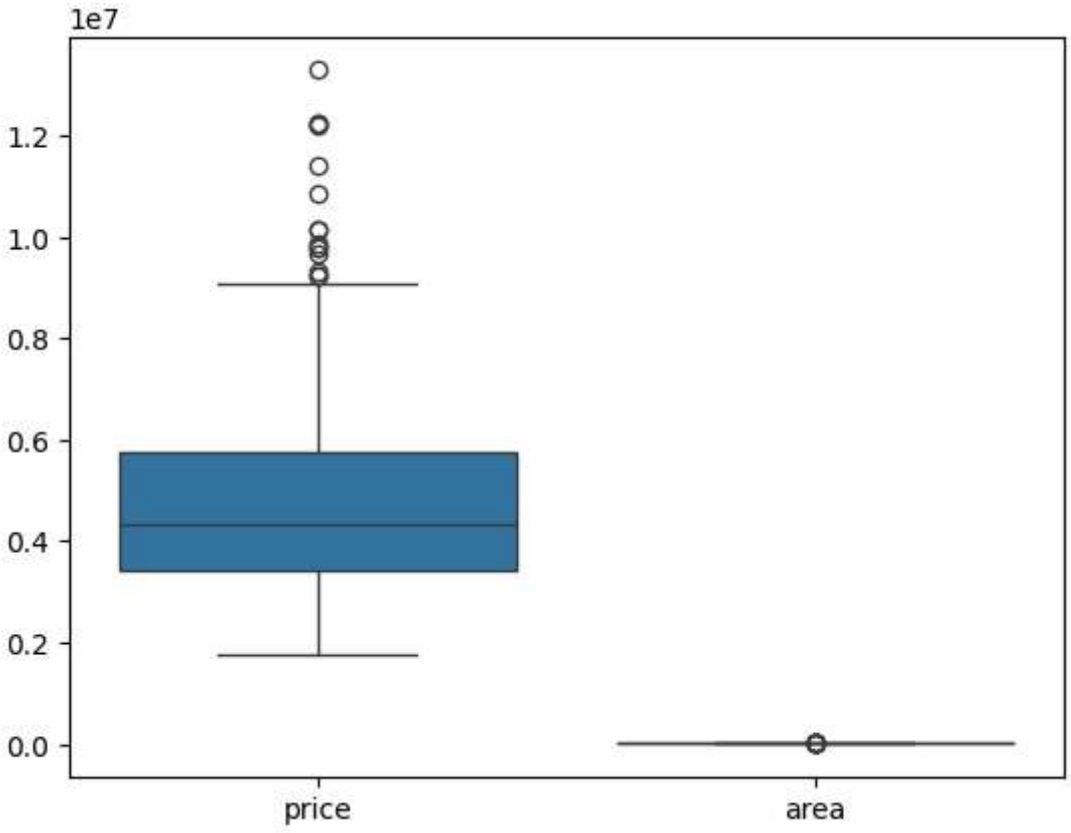
In [26]:
```python
# step 16: Check Skewness
df_encoded.hist(figsize=(12, 10), bins=20)
plt.tight_layout()
plt.show()

print("Skewness:\n", df_encoded.skew(numeric_only=True))
```




```
Skewness:
 price                            1.212239
area                              1.321188
bedrooms                          0.495684
bathrooms                         1.589264
stories                           1.082088
parking                           0.842062
mainroad_yes                     -2.065410
guestroom_yes                     1.688419
basement_yes                      0.628590
hotwaterheating_yes               4.353428
airconditioning_yes               0.795748
prefarea_yes                      1.254361
furnishingstatus_semi-furnished   0.339635
furnishingstatus_unfurnished      0.741509
dtype: float64
```

In [27]:
```
#step 12: Outlier Detection
sns.boxplot(data=df_encoded[['price', 'area']])
plt.show()
```



In [28]:
```
# step 13:
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta

model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

In [29]:
```
# step 14
# Evaluation
print("\nModel Evaluation:")
print("R2 Score:", r2_score(y_test, y_pred))
print("Mean Squared Error:", mean_squared_error(y_test, y_pred))

# Coefficients
coeff_df = pd.DataFrame(model.coef_, X.columns, columns=['Coefficient'])
print("\nFeature Coefficients:\n", coeff_df)
```

```
Model Evaluation:
R2 Score: 0.6529242642153184
Mean Squared Error: 1754318687330.6643

Feature Coefficients:
                                  Coefficient
area                             2.359688e+02
bedrooms                         7.677870e+04
bathrooms                        1.094445e+06
stories                          4.074766e+05
parking                          2.248419e+05
mainroad_yes                     3.679199e+05
guestroom_yes                    2.316100e+05
basement_yes                     3.902512e+05
hotwaterheating_yes              6.846499e+05
airconditioning_yes              7.914267e+05
prefarea_yes                     6.298906e+05
furnishingstatus_semi-furnished -1.268818e+05
furnishingstatus_unfurnished    -4.136451e+05
```

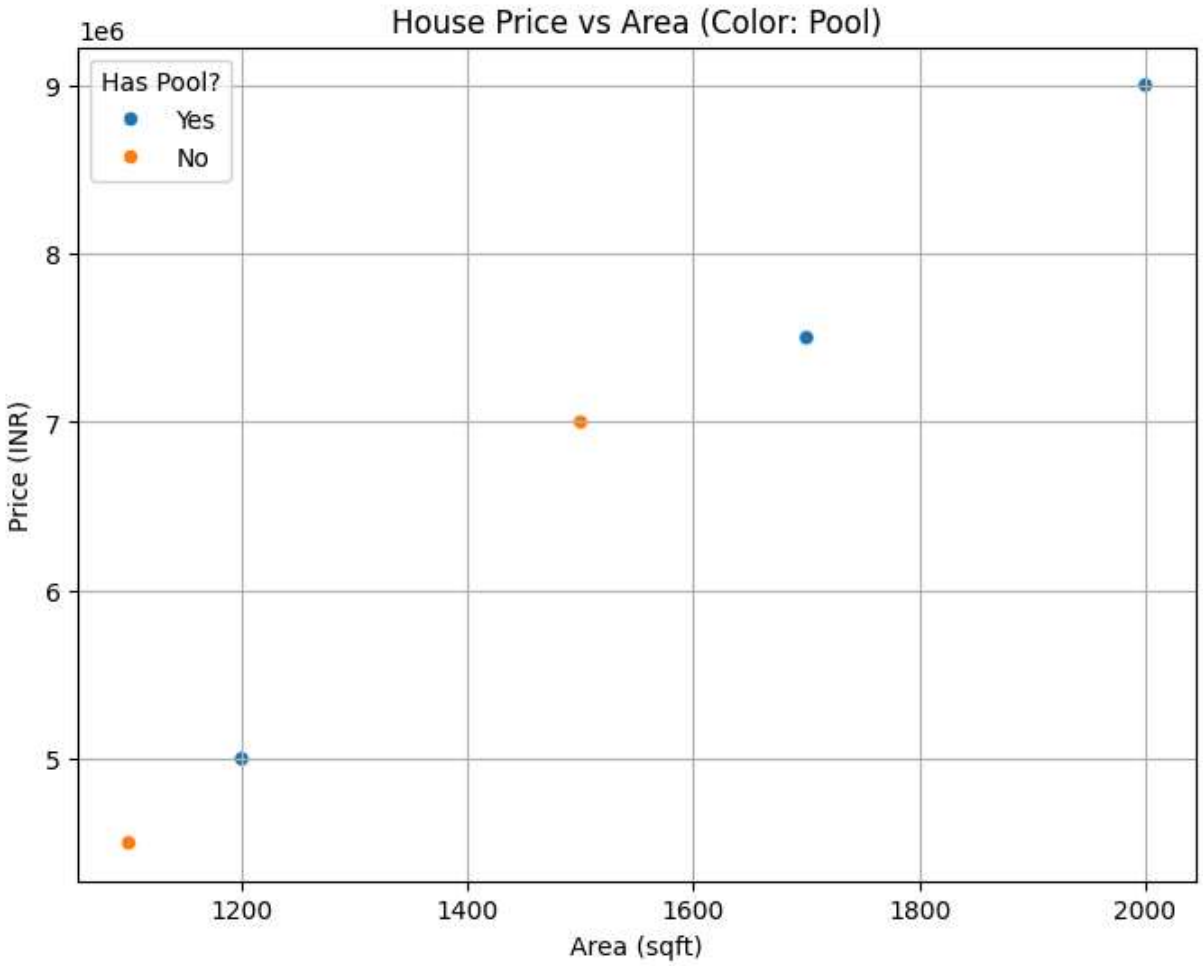
In [30]:

```python
#step 15:
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

# Sample data
data = {
    "area": [1200, 1500, 1700, 1100, 2000],
    "price": [5000000, 7000000, 7500000, 4500000, 9000000],
    "Has Pool?": ["Yes", "No", "Yes", "No", "Yes"]
}
df = pd.DataFrame(data)

# Clean column names
df.columns = df.columns.str.strip()

# Plot
plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x="area", y="price", hue="Has Pool?")
plt.title("House Price vs Area (Color: Pool)")
plt.xlabel("Area (sqft)")
plt.ylabel("Price (INR)")
plt.grid(True)
plt.show()
```



In [31]:
```
#step 16: Filter listings (e.g., houses with 4 bedrooms and air conditioning).
import pandas as pd
from io import StringIO  # This imports StringIO

data = """
price   area    bedrooms        bathrooms       stories mainroad        guestroom
13300000        7420    4       2       3       yes     no      no      no      yes
12250000        8960    4       4       4       yes     no      no      no      yes
12250000        9960    3       2       2       yes     no      yes     no      no
12215000        7500    4       2       2       yes     no      yes     no      yes
11410000        7420    4       1       2       yes     yes     yes     no      yes
10850000        7500    3       3       1       yes     no      yes     no      yes
10150000        8580    4       3       4       yes     no      no      no      yes
10150000        16200   5       3       2       yes     no      no      no      no
9870000 8100    4       1       2       yes     yes     yes     no      yes     2
9800000 5750    3       2       4       yes     yes     no      no      yes     1
"""

df = pd.read_csv(StringIO(data), sep='\t')

filtered_listings = df[(df['bedrooms'] == 4) & (df['airconditioning'] == 'yes')]

print(filtered_listings)
```

```
      price  area  bedrooms  bathrooms  stories mainroad guestroom basement  \
0  13300000  7420         4          2        3      yes        no       no
1  12250000  8960         4          4        4      yes        no       no
3  12215000  7500         4          2        2      yes        no      yes
4  11410000  7420         4          1        2      yes       yes      yes
6  10150000  8580         4          3        4      yes        no       no
8   9870000  8100         4          1        2      yes       yes      yes

  hotwaterheating airconditioning  parking prefarea furnishingstatus
0              no              yes        2      yes        furnished
1              no              yes        3       no        furnished
3              no              yes        3      yes        furnished
4              no              yes        2       no        furnished
6              no              yes        2      yes   semi-furnished
8              no              yes        2      yes        furnished
```

In [32]:

```
# step 17 Find top/bottom listings (e.g., most expensive, cheapest).
import pandas as pd
from io import StringIO

data = """
price   area    bedrooms        bathrooms       stories mainroad        guestroom
13300000        7420    4       2       3       yes     no      no      no      yes
12250000        8960    4       4       4       yes     no      no      no      yes
12250000        9960    3       2       2       yes     no      yes     no      no
12215000        7500    4       2       2       yes     no      yes     no      yes
11410000        7420    4       1       2       yes     yes     yes     no      yes
10850000        7500    3       3       1       yes     no      yes     no      yes
10150000        8580    4       3       4       yes     no      no      no      yes
10150000        16200   5       3       2       yes     no      no      no      no
9870000 8100    4       1       2       yes     yes     yes     no      yes     2
9800000 5750    3       2       4       yes     yes     no      no      yes     1
"""

df = pd.read_csv(StringIO(data), sep='\t')

# Top 3 most expensive listings
top_expensive = df.nlargest(3, 'price')

# Bottom 3 cheapest listings
bottom_cheapest = df.nsmallest(3, 'price')

print("Top 3 most expensive listings:")
print(top_expensive)

print("\nBottom 3 cheapest listings:")
print(bottom_cheapest)
```

```
Top 3 most expensive listings:
      price  area  bedrooms  bathrooms  stories mainroad guestroom basement  \
0  13300000  7420         4          2        3      yes        no       no
1  12250000  8960         4          4        4      yes        no       no
2  12250000  9960         3          2        2      yes        no      yes

  hotwaterheating airconditioning  parking prefarea furnishingstatus
0              no             yes        2      yes        furnished
1              no             yes        3       no        furnished
2              no              no        2      yes   semi-furnished

Bottom 3 cheapest listings:
      price  area  bedrooms  bathrooms  stories mainroad guestroom basement  \
9   9800000  5750         3          2        4      yes       yes       no
8   9870000  8100         4          1        2      yes       yes      yes
6  10150000  8580         4          3        4      yes        no       no

  hotwaterheating airconditioning  parking prefarea furnishingstatus
9              no             yes        1      yes      unfurnished
8              no             yes        2      yes        furnished
6              no             yes        2      yes   semi-furnished
```

In [33]:

```python
#step 18:Visualize trends (e.g., price vs. area scatter plot).
import pandas as pd
import matplotlib.pyplot as plt
from io import StringIO

data = """
price   area    bedrooms        bathrooms       stories mainroad        guestroom
13300000        7420    4       2       3       yes     no      no      no      yes
12250000        8960    4       4       4       yes     no      no      no      yes
12250000        9960    3       2       2       yes     no      yes     no      no
12215000        7500    4       2       2       yes     no      yes     no      yes
11410000        7420    4       1       2       yes     yes     yes     no      yes
10850000        7500    3       3       1       yes     no      yes     no      yes
10150000        8580    4       3       4       yes     no      no      no      yes
10150000        16200   5       3       2       yes     no      no      no      no
9870000 8100    4       1       2       yes     yes     yes     no      yes     2
9800000 5750    3       2       4       yes     yes     no      no      yes     1
"""

# Load data
df = pd.read_csv(StringIO(data), sep='\t')

# Scatter plot
plt.figure(figsize=(10,6))
plt.scatter(df['area'], df['price'], color='blue', alpha=0.6)
plt.title('Price vs Area')
plt.xlabel('Area (sq ft)')
plt.ylabel('Price')
plt.grid(True)
plt.show()
```

In [34]:

```
# step 19: ( predict price based on features).
import pandas as pd
from io import StringIO
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Sample data
data = """
price   area    bedrooms        bathrooms       stories mainroad        guestroom
13300000        7420    4       2       3       yes     no      no      no      yes
12250000        8960    4       4       4       yes     no      no      no      yes
12250000        9960    3       2       2       yes     no      yes     no      no
12215000        7500    4       2       2       yes     no      yes     no      yes
11410000        7420    4       1       2       yes     yes     yes     no      yes
10850000        7500    3       3       1       yes     no      yes     no      yes
10150000        8580    4       3       4       yes     no      no      no      yes
10150000        16200   5       3       2       yes     no      no      no      no
9870000 8100    4       1       2       yes     yes     yes     no      yes     2
9800000 5750    3       2       4       yes     yes     no      no      yes     1
"""

# Load data
df = pd.read_csv(StringIO(data), sep='\t')

# Convert categorical columns (yes/no) to 1/0
for col in ['mainroad', 'guestroom', 'basement', 'hotwaterheating', 'airconditionin
    df[col] = df[col].map({'yes': 1, 'no': 0})

# Convert furnishingstatus to numeric categories
df['furnishingstatus'] = df['furnishingstatus'].map({'unfurnished': 0, 'semi-furnis
```

```
# Features and target
X = df.drop('price', axis=1)
y = df['price']

# Split data into training and testing sets (80/20 split)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta

# Initialize and train the Linear Regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Predict on test data
y_pred = model.predict(X_test)

# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error: {mse:.2f}")
print(f"R-squared: {r2:.2f}")

# Show predictions vs actual
results = pd.DataFrame({'Actual Price': y_test, 'Predicted Price': y_pred})
print(results)
```

```
Mean Squared Error: 13250021815975.09
R-squared: -8.36
   Actual Price  Predicted Price
8       9870000     1.280757e+07
1      12250000     8.022622e+06
```

In [ ]: