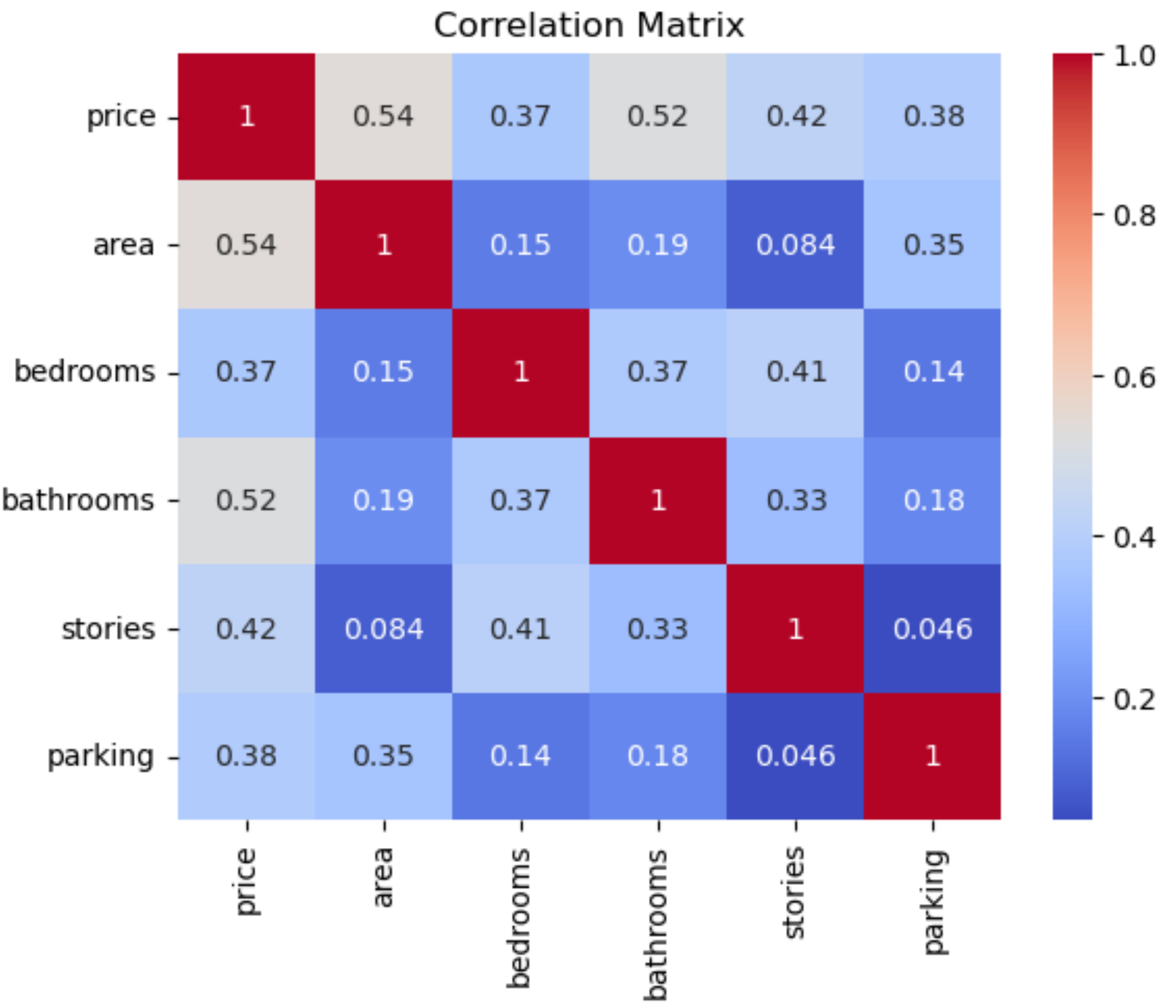
In [4]:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

df = pd.read_csv('Housing.csv')


print(df.head())


print(df.describe())


print(df.isnull().sum())


correlation_matrix = df.corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title("Correlation Matrix")
plt.show()


X = df[['bedrooms', 'bathrooms',]]
y = df['price']


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size
```

```
      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 furnishingstat
us  
0              no             yes        2      yes        furnish
ed  
1              no             yes        3       no        furnish
ed  
2              no              no        2      yes   semi-furnish
ed  
```

```
3              no             yes          3      yes        furnish
ed
4              no             yes          2       no        furnish
ed
              price          area    bedrooms   bathrooms     stor
ies  \
count  5.450000e+02    545.000000  545.000000  545.000000  545.000
000
mean   4.766729e+06   5150.541284    2.965138    1.286239    1.805
505
std    1.870440e+06   2170.141023    0.738064    0.502470    0.867
492
min    1.750000e+06   1650.000000    1.000000    1.000000    1.000
000
25%    3.430000e+06   3600.000000    2.000000    1.000000    1.000
000
50%    4.340000e+06   4600.000000    3.000000    1.000000    2.000
000
75%    5.740000e+06   6360.000000    3.000000    2.000000    2.000
000
max    1.330000e+07  16200.000000    6.000000    4.000000    4.000
000

          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
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

/var/folders/l9/49s71scs52x36hwkp6gmhklr0000gn/T/ipykernel_3135/13
23379855.py:22: FutureWarning: The default value of numeric_only i
n DataFrame.corr is deprecated. In a future version, it will defau
lt to False. Select only valid columns or specify the value of num
eric_only to silence this warning.
  correlation_matrix = df.corr()
```



	price	area	bedrooms	bathrooms	stories	parking
price	1	0.54	0.37	0.52	0.42	0.38
area	0.54	1	0.15	0.19	0.084	0.35
bedrooms	0.37	0.15	1	0.37	0.41	0.14
bathrooms	0.52	0.19	0.37	1	0.33	0.18
stories	0.42	0.084	0.41	0.33	1	0.046
parking	0.38	0.35	0.14	0.18	0.046	1

In [5]:

```

model = LinearRegression()


model.fit(X_train, y_train)
```

Out[5]:

```
▾ LinearRegression
LinearRegression()
```

In [6]:

```
# Model Evaluation
y_pred = model.predict(X_test)


mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print("Mean Squared Error:", mse)
print("R-squared:", r2)
```

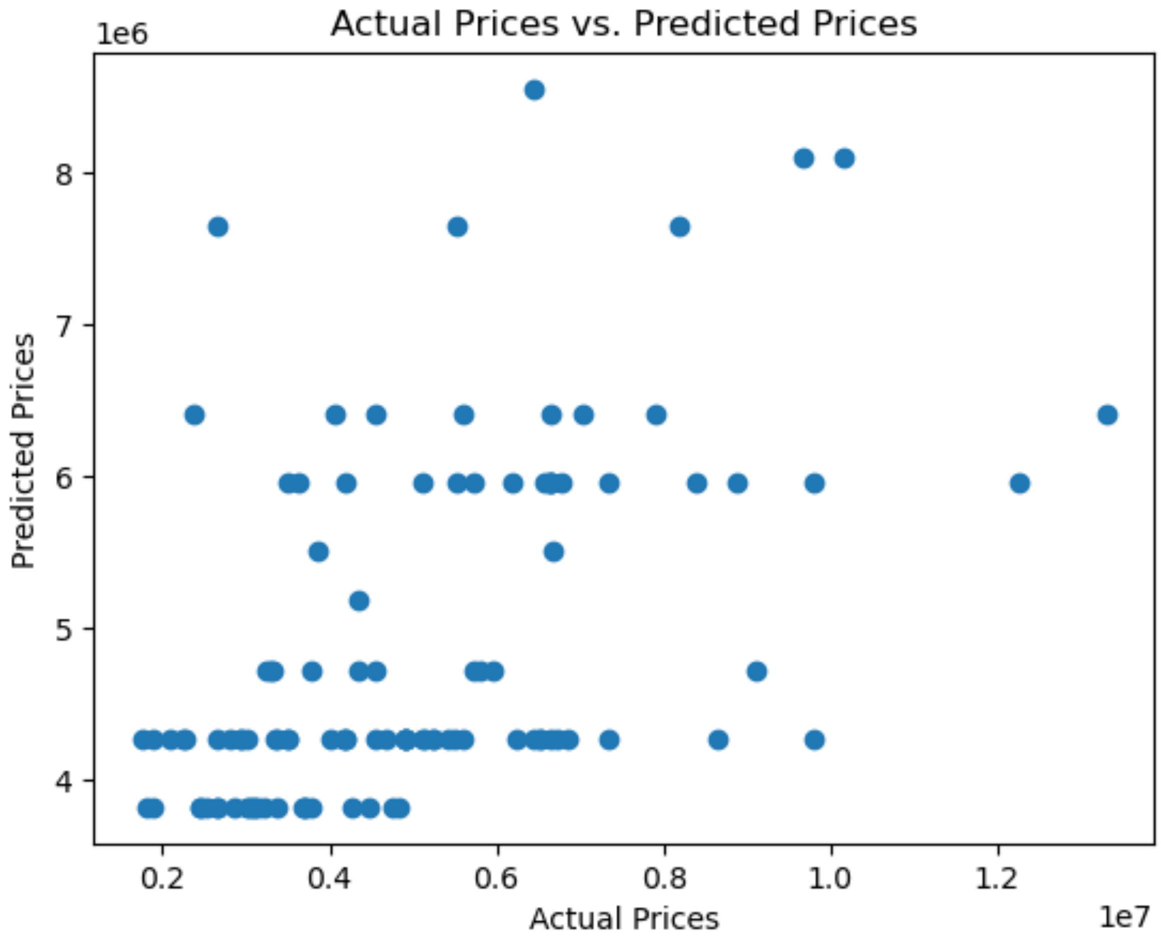
```
Mean Squared Error: 3737500333041.07
R-squared: 0.2605701076698277
```

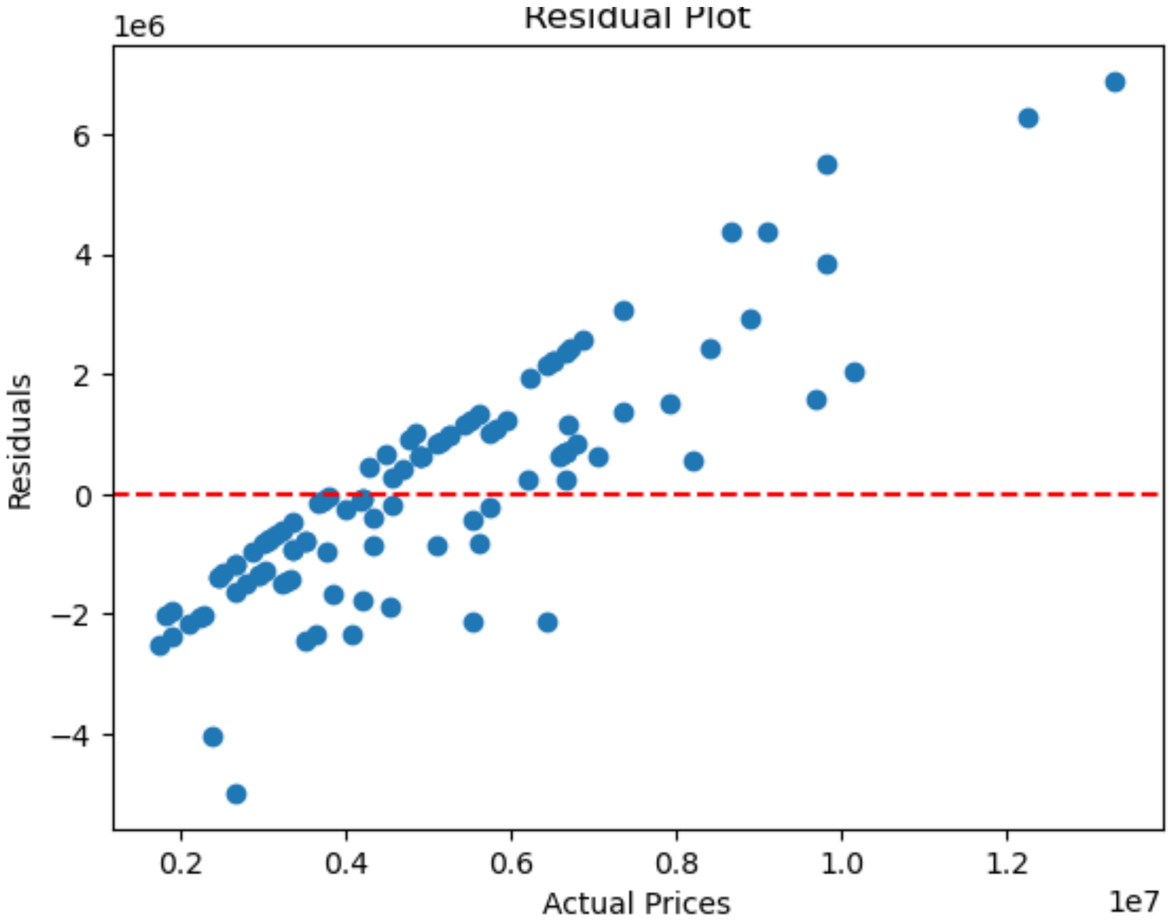
In [ ]:

```

```

In [8]:

```
# Predictions and Visualization

plt.scatter(y_test, y_pred)
plt.xlabel("Actual Prices")
plt.ylabel("Predicted Prices")
plt.title("Actual Prices vs. Predicted Prices")
plt.show()


residuals = y_test - y_pred
plt.scatter(y_test, residuals)
plt.axhline(y=0, color='red', linestyle='--')
plt.xlabel("Actual Prices")
plt.ylabel("Residuals")
plt.title("Residual Plot")
plt.show()


new_data = [[4000,3]]
predicted_price = model.predict(new_data)

print("Predicted Price:", predicted_price[0])
```





```
Predicted Price: 1816016323.403072

/Users/amjaf/anaconda3/lib/python3.11/site-packages/sklearn/base.p
y:439: UserWarning: X does not have valid feature names, but Linea
rRegression was fitted with feature names
  warnings.warn(
```

```
In [ ]:
```