In [1]:
```
import pandas as pd
```

In [2]:
```
df = pd.read_csv('USA_Housing.csv')
print(df)
```

```
      Avg. Area Income  Avg. Area House Age  Avg. Area Number of Rooms  \
0         79545.458574             5.682861                   7.009188
1         79248.642455             6.002900                   6.730821
2         61287.067179             5.865890                   8.512727
3         63345.240046             7.188236                   5.586729
4         59982.197226             5.040555                   7.839388
...                ...                  ...                        ...
4995      60567.944140             7.830362                   6.137356
4996      78491.275435             6.999135                   6.576763
4997      63390.686886             7.250591                   4.805081
4998      68001.331235             5.534388                   7.130144
4999      65510.581804             5.992305                   6.792336

      Avg. Area Number of Bedrooms  Area Population         Price  \
0                             4.09     23086.800503  1.059034e+06
1                             3.09     40173.072174  1.505891e+06
2                             5.13     36882.159400  1.058988e+06
3                             3.26     34310.242831  1.260617e+06
4                             4.23     26354.109472  6.309435e+05
...                            ...              ...           ...
4995                          3.46     22837.361035  1.060194e+06
4996                          4.02     25616.115489  1.482618e+06
4997                          2.13     33266.145490  1.030730e+06
4998                          5.44     42625.620156  1.198657e+06
4999                          4.07     46501.283803  1.298950e+06

                                          Address
0     208 Michael Ferry Apt. 674\nLaurabury, NE 3701...
1     188 Johnson Views Suite 079\nLake Kathleen, CA...
2     9127 Elizabeth Stravenue\nDanieltown, WI 06482...
3                           USS Barnett\nFPO AP 44820
4                          USNS Raymond\nFPO AE 09386
...                                               ...
4995                   USNS Williams\nFPO AP 30153-7653
4996              PSC 9258, Box 8489\nAPO AA 42991-3352
4997  4215 Tracy Garden Suite 076\nJoshualand, VA 01...
4998                           USS Wallace\nFPO AE 73316
4999  37778 George Ridges Apt. 509\nEast Holly, NV 2...

[5000 rows x 7 columns]
```

In [3]:
```
print(df.info())
print(df.describe())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 7 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   Avg. Area Income              5000 non-null   float64
 1   Avg. Area House Age           5000 non-null   float64
 2   Avg. Area Number of Rooms     5000 non-null   float64
 3   Avg. Area Number of Bedrooms  5000 non-null   float64
 4   Area Population               5000 non-null   float64
 5   Price                         5000 non-null   float64
 6   Address                       5000 non-null   object 
dtypes: float64(6), object(1)
memory usage: 273.6+ KB
None
       Avg. Area Income  Avg. Area House Age  Avg. Area Number of Rooms  \
count       5000.000000          5000.000000                5000.000000   
mean       68583.108984             5.977222                   6.987792   
std        10657.991214             0.991456                   1.005833   
min        17796.631190             2.644304                   3.236194   
25%        61480.562388             5.322283                   6.299250   
50%        68804.286404             5.970429                   7.002902   
75%        75783.338666             6.650808                   7.665871   
max       107701.748378             9.519088                  10.759588   

       Avg. Area Number of Bedrooms  Area Population         Price  
count                   5000.000000      5000.000000  5.000000e+03  
mean                       3.981330     36163.516039  1.232073e+06  
std                        1.234137      9925.650114  3.531176e+05  
min                        2.000000       172.610686  1.593866e+04  
25%                        3.140000     29403.928702  9.975771e+05  
50%                        4.050000     36199.406689  1.232669e+06  
75%                        4.490000     42861.290769  1.471210e+06  
max                        6.500000     69621.713378  2.469066e+06  
```

In [4]:
```
X = df.drop(['Price', 'Address'], axis=1)
y = df['Price']
```

In [5]:
```
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=42)
```

In [6]:
```
from sklearn.linear_model import LinearRegression
```

```
model = LinearRegression()
model.fit(X_train, y_train)
```

Out[6]:

LinearRegression

LinearRegression()

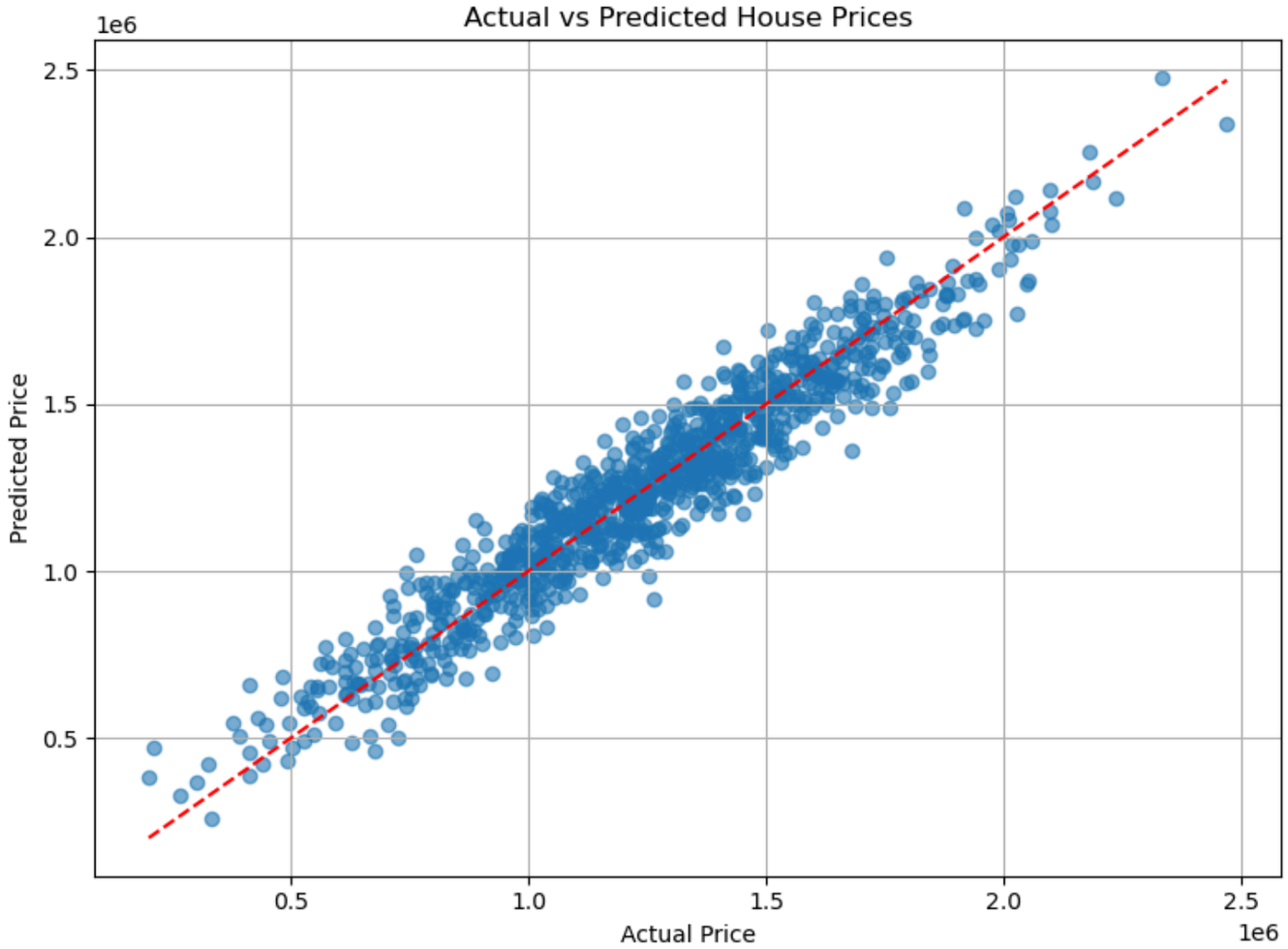
In [7]:

```
from sklearn.metrics import mean_squared_error, r2_score
import numpy as np

y_pred = model.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)

absolute_errors = abs(y_pred - y_test)
percentage_errors = (absolute_errors / y_test) * 100

mean_absolute_error = np.mean(absolute_errors)
mean_percentage_error = np.mean(percentage_errors)

print(f"Mean Squared Error        : {mse:.4f}")
print(f"Root Mean Squared Error   : {rmse:.4f}")
print(f"Mean Absolute Error       : {mean_absolute_error:.2f}")
print(f"Mean Percentage Error     : {mean_percentage_error:.2f}%")
print(f"R2 Score (Overall Fit)    : {r2_score(y_test, y_pred):.4f}")
```

```
Mean Squared Error        : 10089009300.8929
Root Mean Squared Error   : 100444.0606
Mean Absolute Error       : 80879.10
Mean Percentage Error     : 7.39%
R2 Score (Overall Fit)    : 0.9180
```

In [11]:

```
import matplotlib.pyplot as plt

plt.figure(figsize=(8, 6))
plt.scatter(y_test, y_pred, alpha=0.6)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--')
plt.xlabel("Actual Price")
plt.ylabel("Predicted Price")
plt.title("Actual vs Predicted House Prices")
plt.grid(True)
plt.tight_layout()
plt.show()
```



In [12]:
```
index = 0

sample_input = X_test.iloc[[index]]
actual_price = y_test.iloc[index]
predicted_price = model.predict(sample_input)[0]

print(f"Actual Price        : ${actual_price:,.2f}")
print(f"Predicted Price     : ${predicted_price:,.2f}")
print(f"Absolute Error      : ${abs(predicted_price - actual_price):,.2f}")
print(f"Percentage Error    : {abs(predicted_price - actual_price) / actual_price * 100:.2f}%")
```

```
Actual Price        : $1,339,096.08
Predicted Price     : $1,308,587.93
Absolute Error      : $30,508.15
Percentage Error    : 2.28%
```

In [ ]: