Prediction House price by Linear Regression:

In [61]:
```
# Import libraries:
import numpy as np
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, mean_squared_error, r2_score
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
```

In [62]:
```
# Dataset:
data = pd.read_csv('C:/Users/Lenovo/Documents/House.Price.csv')
data.head()
```

Out[62]:

	House Price	Living Area
0	218000	1580
1	158900	1100
2	229000	2560
3	94000	1064
4	233000	2080

In [63]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 2 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   House Price  1000 non-null   int64
 1   Living Area  1000 non-null   int64
dtypes: int64(2)
memory usage: 15.8 KB
```

In [64]:
```
data.shape
```

Out[64]:
```
(1000, 2)
```

In [65]:
```
data.describe()
```

Out[65]:

	House Price	Living Area
count	1000.000000	1000.00000
mean	220287.902000	1857.98000
std	94463.059716	648.29803
min	52500.000000	540.00000
25%	149975.000000	1385.00000
50%	199950.000000	1759.50000
75%	273300.000000	2286.75000
max	590000.000000	4859.00000

In [66]:
```
data.isnull().sum()
```

Out[66]:
```
House Price    0
Living Area    0
dtype: int64
```

In [73]:
```
# Train_Test_Split:
X=np.array(data['Living Area']).reshape(-1,1)
y=np.array(data['House Price']).reshape(-1,1)
```

In [74]:
```
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.2, ran
```

In [75]:
```
#preprocessing:
scaler = StandardScaler()
```

In [76]:
```
X_train_scaled = scaler.fit_transform(X_train)
```

In [77]:
```
X_test_scaled = scaler.transform(X_test)
```

In [78]:
```
# Trainning the model:
model = LinearRegression()
model.fit(X_train_scaled, y_train)
```

Out[78]:
```
LinearRegression()
```

In [86]:
```
y_pred = model.predict(X_test_scaled)
y_pred = np.round(y_pred,decimals=2)
y_pred
```

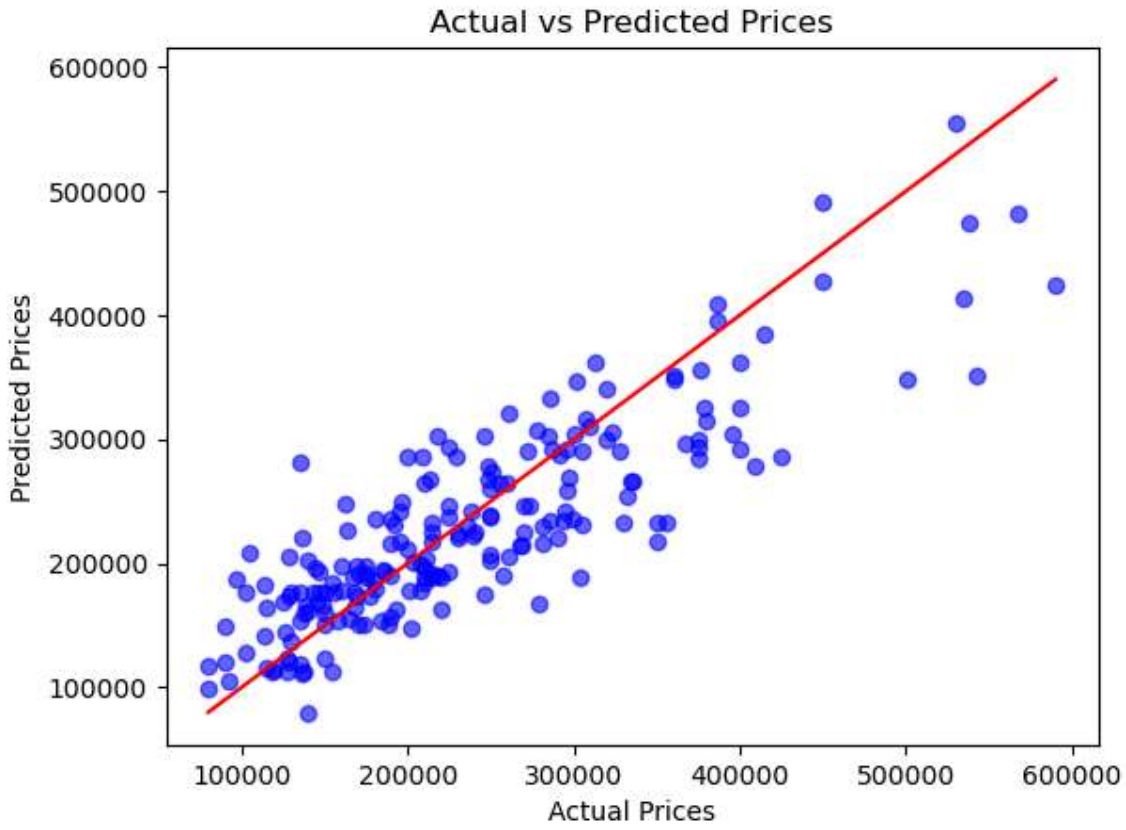
Out[86]:
```
array([[150430.46],
       [268429.97],
       [264956.1 ],
       [176428.45],
       [217666.64],
       [291850.57],
       [188194.78],
       [362000.33],
       [193349.56],
       [220356.09],
       [266188.76],
       [214753.08],
       [204219.41],
       [169256.59],
       [182703.83],
       [307202.84],
       [290057.61],
       [154240.51],
       [247026.45],
```

In [81]:
```
# Evaluate the model:
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)

print(f"Root Mean Squared Error (RMSE): {rmse:.2f}")
```

```
Root Mean Squared Error (RMSE): 53823.97
```

In [82]:
```
# Visualization:
plt.scatter(y_test, y_pred, color='blue', alpha=0.6)
plt.xlabel("Actual Prices")
plt.ylabel("Predicted Prices")
plt.title("Actual vs Predicted Prices")
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], color=
plt.show()
```



In [ ]: