Task 1: Implement a linear regression model to predict the prices of houses based on their square footage and the number of bedrooms and bathrooms.

In [40]:

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

Load the dataset

In [47]:

```
data = pd.read_csv(r'C:\Users\Lenovo\Downloads\train.csv')
```

Select relevant features (Square Footage, Bedrooms, and Bathrooms) and the target (Price)

In [51]:

```
features = ['GrLivArea', 'BedroomAbvGr', 'FullBath']
target = 'SalePrice'
```

Extract features (X) and target (y)

In [54]:

```
X = data[features]
y = data[target]
```

Handle missing values by filling them with the median

In [57]:

```
X = X.fillna(X.median())
```

Split the data into training and testing sets

In [60]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Train the Linear Regression model

In [63]:

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

Out[63]:

```
▾ LinearRegression
LinearRegression()
```

Make predictions

In [66]:

```
y_pred = model.predict(X_test)
```

Evaluate the model

In [69]:

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

Output evaluation metrics

In [72]:

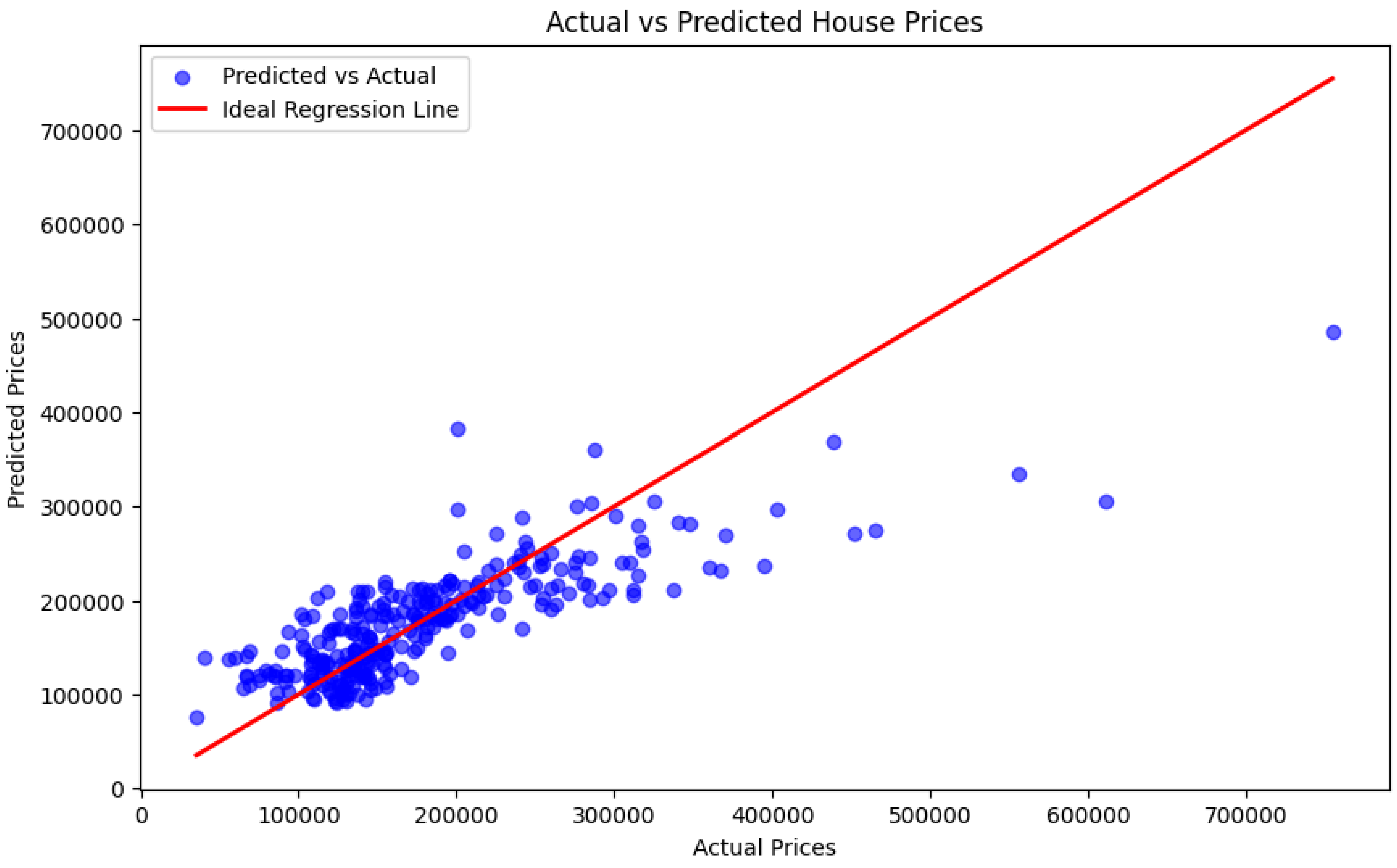
```
print("Model Coefficients:", model.coef_)
print("Model Intercept:", model.intercept_)
print("Mean Squared Error (MSE):", mse)
print("R-squared (R²):", r2)
```

```
Model Coefficients: [   104.02630701 -26655.16535734  30014.32410896]
Model Intercept: 52261.74862694448
Mean Squared Error (MSE): 2806426667.247852
R-squared (R²): 0.6341189942328374
```

Plotting Actual Prices vs Predicted Prices

In [75]:

```
plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_pred, color='blue', alpha=0.6, label='Predicted vs Actual')
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], color='red', linewidth=2, label='Ideal Regression Line')
plt.xlabel("Actual Prices")
plt.ylabel("Predicted Prices")
plt.title("Actual vs Predicted House Prices")
plt.legend()
plt.show()
```


Actual vs Predicted House Prices
Predicted vs Actual
Ideal Regression Line
Predicted Prices
Actual Prices
0
100000
200000
300000
400000
500000
600000
700000