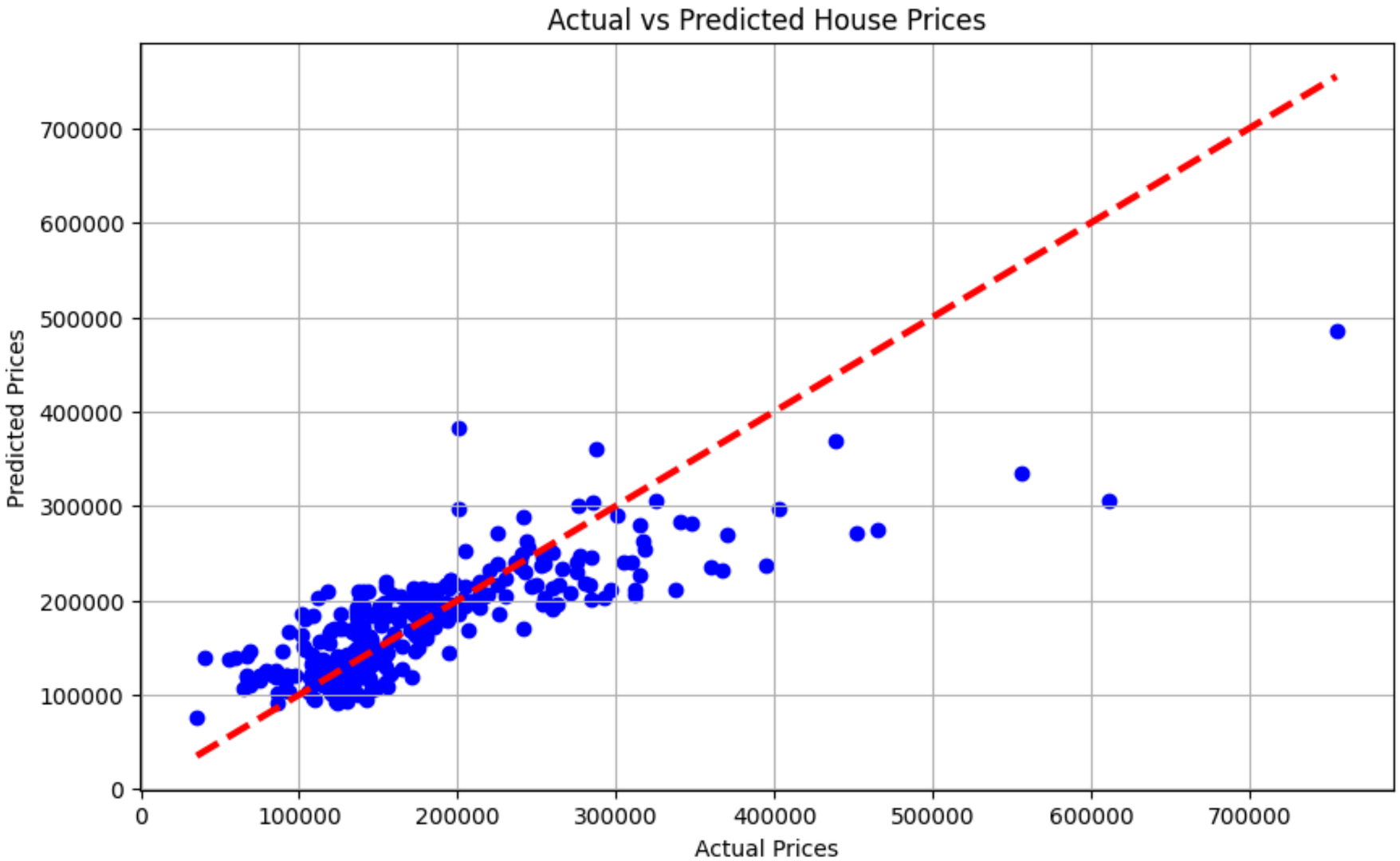
Linear Regression Model

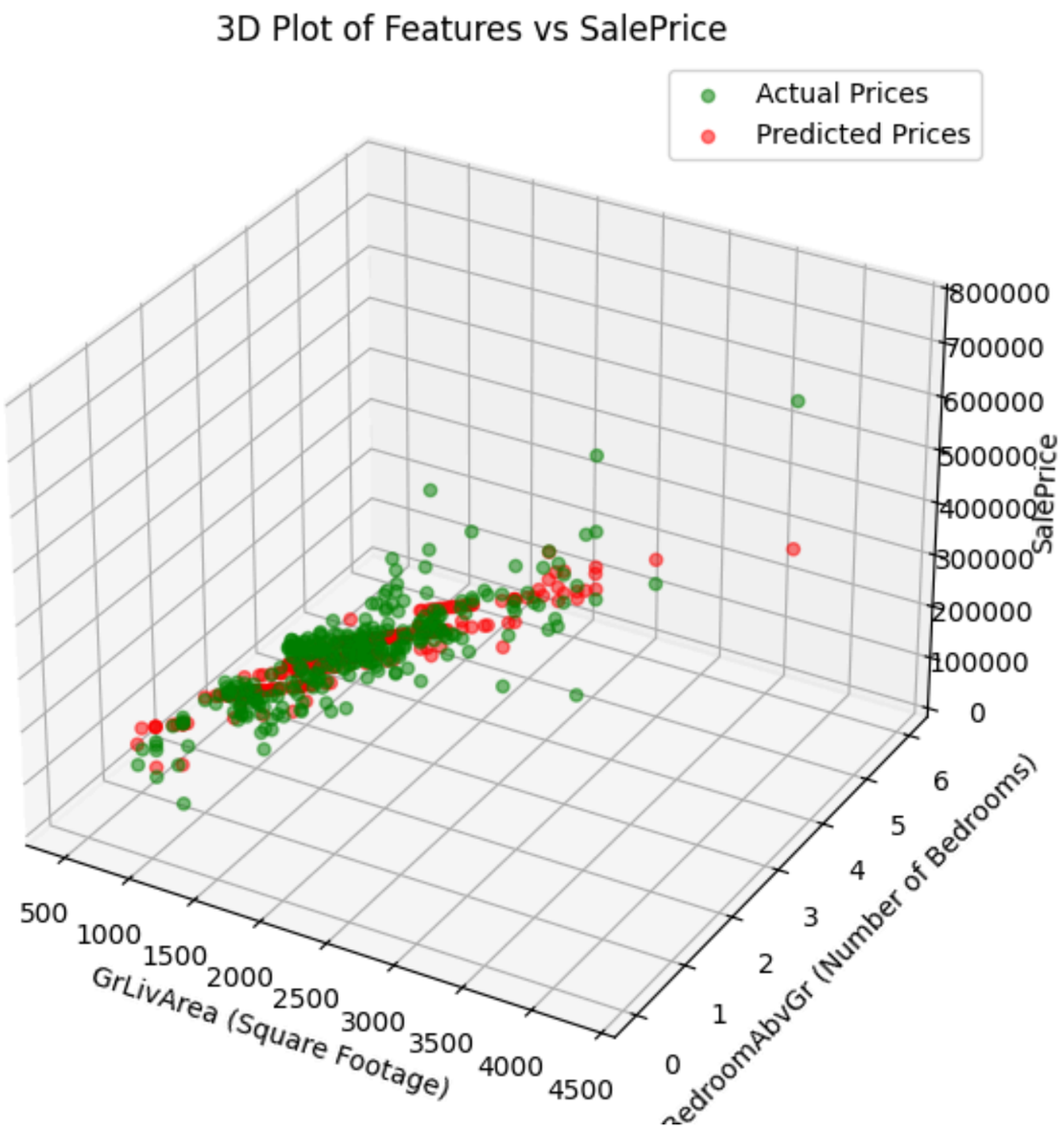
In [2]:
```
# Importing required libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import numpy as np

# Step 1: Load the dataset
train_data_path = 'C:/Users/Suryanshi/Desktop/MY DA PROJECTS/Machine Learning Internship/Task_1/train.csv'  # Path to the train.csv file
train_data = pd.read_csv(train_data_path)

# Step 2: Data Cleaning
# Selecting relevant columns
relevant_columns = ['GrLivArea', 'BedroomAbvGr', 'FullBath', 'SalePrice']
df = train_data[relevant_columns].dropna()  # Remove rows with missing values

# Defining feature variables (X) and target variable (y)
X = df[['GrLivArea', 'BedroomAbvGr', 'FullBath']]
y = df['SalePrice']

# Step 3: Train-Test Split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Step 4: Model Implementation - Linear Regression
model = LinearRegression()
model.fit(X_train, y_train)

# Step 5: Model Evaluation
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)

# Output the results
print(f'Root Mean Squared Error (RMSE): {rmse}')
print(f'Coefficients: {model.coef_}')
print(f'Intercept: {model.intercept_}')
```

```
Root Mean Squared Error (RMSE): 52975.71771338122
Coefficients: [   104.02630701 -26655.16535734  30014.32410896]
Intercept: 52261.748626944594
```

Visualization of the Linear Regression

In [3]:
```
# Import necessary libraries for visualization
import matplotlib.pyplot as plt
import seaborn as sns
from mpl_toolkits.mplot3d import Axes3D

# 1. Scatter plot: Actual vs Predicted Prices
plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_pred, color='blue')
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--', lw=3)
plt.xlabel('Actual Prices')
plt.ylabel('Predicted Prices')
plt.title('Actual vs Predicted House Prices')
plt.grid(True)
plt.show()

# 2. 3D Plot: GrLivArea, BedroomAbvGr, FullBath vs SalePrice
fig = plt.figure(figsize=(10, 7))
ax = fig.add_subplot(111, projection='3d')

# Scatter points
ax.scatter(X_test['GrLivArea'], X_test['BedroomAbvGr'], y_test, color='green', label='Actual Prices', alpha=0.5)
ax.scatter(X_test['GrLivArea'], X_test['BedroomAbvGr'], y_pred, color='red', label='Predicted Prices', alpha=0.5)

ax.set_xlabel('GrLivArea (Square Footage)')
ax.set_ylabel('BedroomAbvGr (Number of Bedrooms)')
ax.set_zlabel('SalePrice')
ax.set_title('3D Plot of Features vs SalePrice')

plt.legend()
plt.show()
```


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


3D Plot of Features vs SalePrice
Actual Prices
Predicted Prices
SalePrice
800000
700000
600000
500000
400000
300000
200000
100000
0
GrLivArea (Square Footage)
500
1000
1500
2000
2500
3000
3500
4000
4500
BedroomAbvGr (Number of Bedrooms)
0
1
2
3
4
5
6