In [1]:
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm
```

```
Matplotlib is building the font cache; this may take a moment.
```

In [2]:
```
df = pd.read_csv('CarPrice_Assignment.csv')
df.head()
```

Out[2]:

	car_ID	symboling	CarName	fueltype	aspiration	doornumber	carbody	drivewheel	enginel
0	1	3	alfa-romero giulia	gas	std	two	convertible	rwd	
1	2	3	alfa-romero stelvio	gas	std	two	convertible	rwd	
2	3	1	alfa-romero Quadrifoglio	gas	std	two	hatchback	rwd	
3	4	2	audi 100 ls	gas	std	four	sedan	fwd	
4	5	2	audi 100ls	gas	std	four	sedan	4wd	

5 rows × 26 columns

In [3]:
```
X = df[['enginesize', 'horsepower', 'citympg', 'highwaympg']]
y = df['price']
```

In [4]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_s
```

In [5]:
```
scaler= StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

In [6]:
```
model = LinearRegression()
model.fit(X_train_scaled, y_train)
```

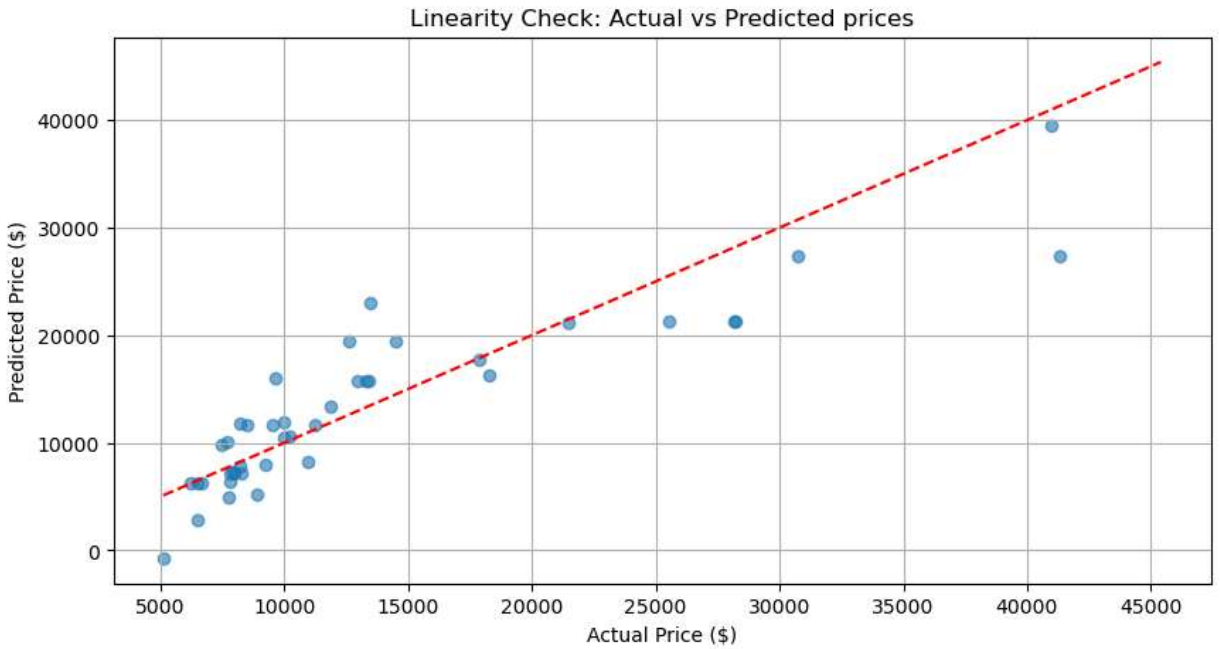
Out[6]:
```
▾ LinearRegression
LinearRegression()
```

In [7]:
```
y_pred = model.predict(X_test_scaled)
```

In [8]:
```
print('Name: Raghul.S')
print('Reg. No: 212225040325')
print("MODEL COEFFICIENTS:")
for feature, coef in zip(X.columns, model.coef_):
    print(f"{feature:>12}: {coef:>10.2f}")
print(f"{'Intercept':>12}: {model.intercept_:>10.2f}")

print("\nMODEL PERFORMANCE:")
print(f"{'MSE':>12}: {mean_squared_error(y_test, y_pred):>10.2f}")
print(f"{'RMSE':>12}: {np.sqrt(mean_squared_error(y_test, y_pred)):>10.2f}")
print(f"{'R-squared':>12}: {r2_score(y_test, y_pred):>10.2f}")
print("="*50)
```

```
Name: Raghul.S
Reg. No: 212225040325
MODEL COEFFICIENTS:
  enginesize:    4523.40
  horsepower:    1694.22
     citympg:    -392.57
  highwaympg:    -816.36
   Intercept:   13223.41

MODEL PERFORMANCE:
         MSE: 16471505.90
        RMSE:    4058.51
   R-squared:       0.79
==================================================
```

In [9]:
```
plt.figure(figsize=(10, 5))
plt.scatter(y_test, y_pred, alpha=0.6)
plt.plot([y.min(), y.max()],[y.min(), y.max()], 'r--')
plt.title("Linearity Check: Actual vs Predicted prices")
plt.xlabel("Actual Price ($)")
plt.ylabel("Predicted Price ($)")
plt.grid(True)
plt.show()
```



In [10]:
```
residuals = y_test - y_pred
dw_test = sm.stats.durbin_watson(residuals)
print(f"\nDurbin-Watson Statistic: {dw_test:.2f}",)
```

```
Durbin-Watson Statistic: 2.28
```

In [11]:
```
residuals = y_test - y_pred
dw_test = sm.stats.durbin_watson(residuals)
print(f"\nDurbin-Watson Statistic: {dw_test:.2f}",
     "/n(Values close to 2 indicate no autocorrelation)")
```

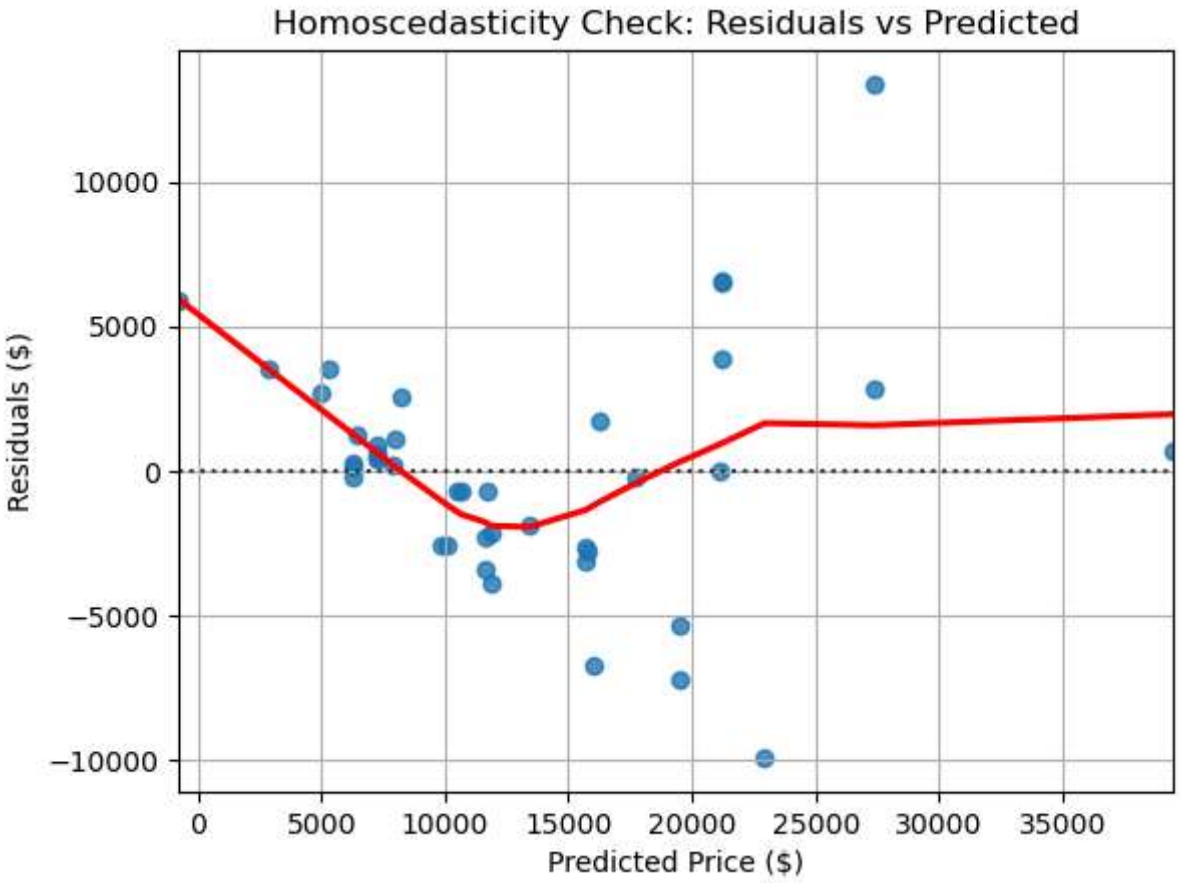
```
Durbin-Watson Statistic: 2.28 /n(Values close to 2 indicate no autocorrelation)
```

In [12]:
```
plt.figure(figsize=(10, 5))
```

Out[12]:
```
<Figure size 1000x500 with 0 Axes>
```

```
<Figure size 1000x500 with 0 Axes>
```

In [13]:
```
sns.residplot(x=y_pred, y=residuals, lowess=True, line_kws={'color': 'red'})
plt.title("Homoscedasticity Check: Residuals vs Predicted")
plt.xlabel("Predicted Price ($)")
plt.ylabel("Residuals ($)")
plt.grid(True)
plt.show()
```

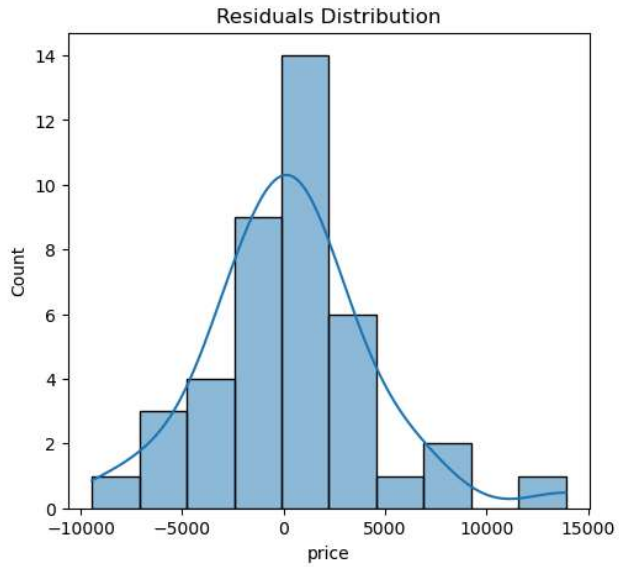


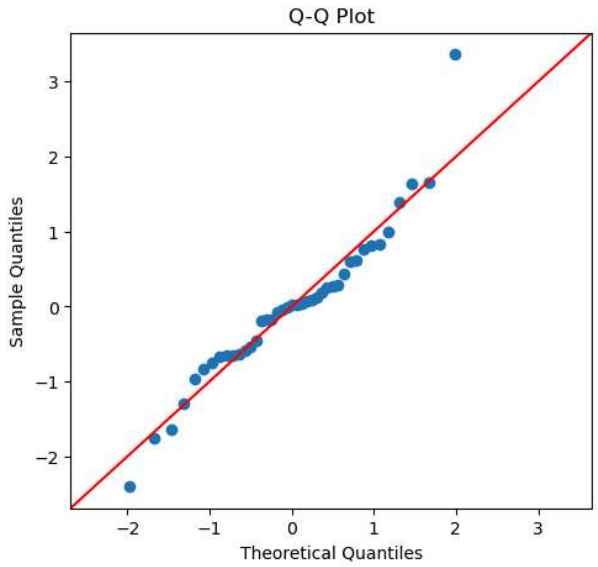
In [15]:
```
fig, (ax1, ax2) = plt.subplot(1, 2, figsize=(12, 5))
sns.histplot(residuals, kde=True, ax=ax1)
ax1.set_title("Residuals Distribution")
sm.qqplot(residuals, line='45',fit=True, ax=ax2)
ax2.set_title("Q-Q Plot")
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[15], line 1
----> 1 fig, (ax1, ax2) = Plt.subplot(1, 2, figsize=(12, 5))
      2 sns.histplot(residuals, kde=True, ax=ax1)
      3 ax1.set_title("Residuals Distribution")

NameError: name 'Plt' is not defined
```

In [16]:
```
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 5))

sns.histplot(residuals, kde=True, ax=ax1)
ax1.set_title("Residuals Distribution")

sm.qqplot(residuals, line='45', fit=True, ax=ax2)
ax2.set_title("Q-Q Plot")
```

Out[16]: Text(0.5, 1.0, 'Q-Q Plot')





In [ ]: