This project aims to predict used car prices in UAE based on specifications and condition

In [13]:

```
# Step 1: Load the Data
import pandas as pd

# Load the dataset
df = pd.read_csv('UAE_Used_cars.csv', encoding='UTF-8-SIG')

# Display the first few rows of the dataframe
print(df.head(), df.describe())
```

```
   Car Brand Car Model  Production Year     Mileage  Price  \
0     Nissan    Altima             2005  445,740 km  3,500
1     Toyota     Camry             1999  200,000 km  5,500
2       Ford     Focus             2006  366,135 km  5,500
3     Toyota      Echo             2005  200,000 km  6,000
4  Chevrolet     Epica             2009  250,000 km   6000

                                  Description          Specs  \
0                                       Dubai      GCC Specs
1              Perfect Condition Toyota Camry      GCC Specs
2                                  FORD FOCUS      GCC Specs
3  GCC - TOYOTA ECHO 2005 - Manual, Urgent Sale      GCC Specs
4                             Chevrolet Epica  American Specs

        Timestamp   Location
0  04-03-24 14:49      Dubai
1  04-03-24 14:49      Dubai
2  04-03-24 14:49      Dubai
3  04-03-24 14:49      Dubai
4     45354.94097  Abu Dhabi          Production Year
count       8006.000000
mean        2017.939046
std            5.227208
min         1929.000000
25%         2015.000000
50%         2019.000000
75%         2022.000000
max         2024.000000
```

In [18]:

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load the data
df = pd.read_csv('UAE_Used_cars.csv', encoding='UTF-8-SIG')

# Convert Price to numeric, removing commas and converting to float
df['Price'] = df['Price'].str.replace(',', '').astype(float)

# Sort by Price in descending order
df_sorted = df.sort_values('Price', ascending=False)

# Display the top 20 cars
print(df_sorted[['Car Brand', 'Car Model', 'Production Year', 'Price', 'Mileage']].head(
```

```
     Car Brand        Car Model  Production Year      Price     Mileage
8005      Ford               GT             2022  4750000.0       30 km
8004      Ford               GT             2022  3900000.0       30 km
8003      Ford               GT             2021  2880000.0       75 km
8002      Ford               GT             2020  2649000.0    4,000 km
8001      Ford          Mustang             1967  1600000.0    1,749 km
8000    Nissan             GT-R             1999   949999.0  121,454 km
7999    Nissan             GT-R             1999   949999.0  137,488 km
7998      Ford          Mustang             1968   799999.0   43,200 km
7997    Nissan           Patrol             2013   750000.0  230,000 km
7996    Nissan             GT-R             2022   599000.0    7,300 km
7995      Ford  F-Series Pickup             2023   575000.0       35 km
7993    Toyota          Alphard             2024   550000.0        0 km
7994    Toyota          Alphard             2024   550000.0        0 km
7992    Toyota          Alphard             2024   540000.0        0 km
7991      Ford  F-Series Pickup             2023   539000.0    1,600 km
7990    Toyota          Alphard             2024   530000.0        0 km
7989    Toyota          Alphard             2024   520000.0        0 km
7988    Toyota          Alphard             2024   515000.0    2,200 km
7987      Ford     Shelby Cobra             2015   499000.0    5,000 km
7986      Ford     Shelby Cobra             1965   499000.0   23,433 km
```

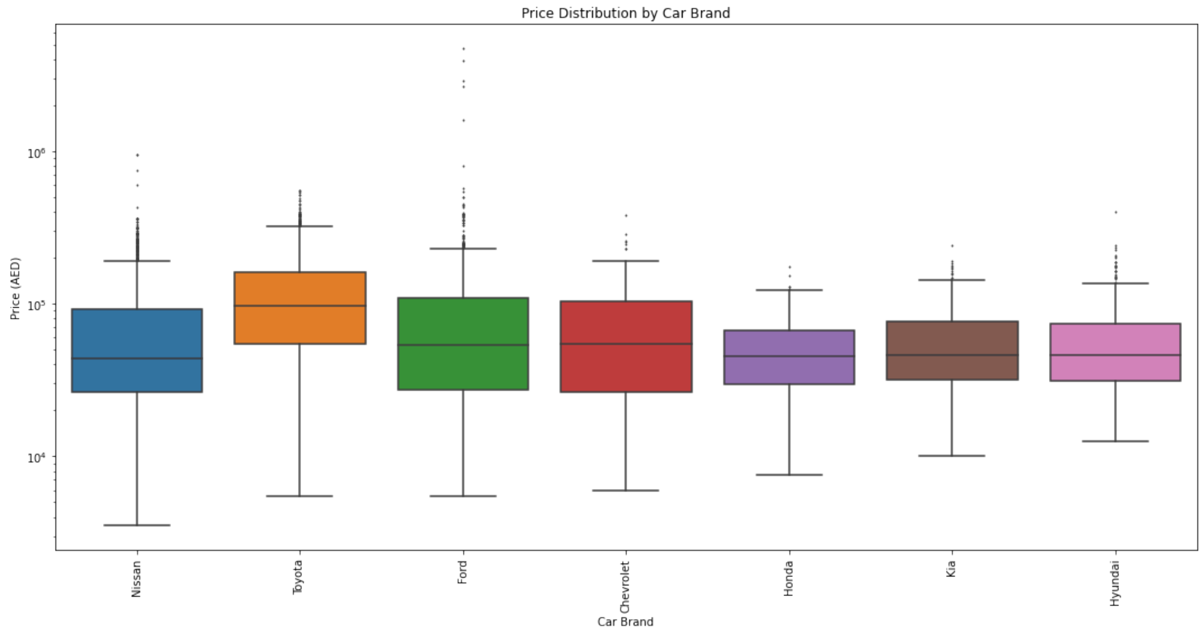
In [25]:

```python
# Calculate and display some statistics
print("\
Price Statistics:")
print(df['Price'].describe())

print("\
Top 5 Most Expensive Car Brands (Average Price):")
print(df.groupby('Car Brand')['Price'].mean().sort_values(ascending=False).head())

print("\
Number of Cars by Brand:")
print(df['Car Brand'].value_counts().head(10))
```

```
Price Statistics:
count    8.006000e+03
mean     8.796613e+04
std      1.131822e+05
min      3.500000e+03
25%      3.242500e+04
50%      5.900000e+04
75%      1.120000e+05
max      4.750000e+06
Name: Price, dtype: float64
Top 5 Most Expensive Car Brands (Average Price):
Car Brand
Toyota       119731.127676
Ford          98059.913651
Chevrolet     79095.475610
Nissan        76283.303059
Kia           57978.809826
Name: Price, dtype: float64
Number of Cars by Brand:
Toyota       2522
Nissan       2125
Ford         1216
Hyundai       850
Kia           631
Honda         580
Chevrolet      82
Name: Car Brand, dtype: int64
```

In [27]:

```
# Create a box plot of prices by car brand
plt.figure(figsize=(15, 8))
sns.boxplot(x='Car Brand', y='Price', data=df, fliersize=1)
plt.title('Price Distribution by Car Brand')
plt.xticks(rotation=90)
plt.ylabel('Price (AED)')
plt.yscale('log')  # Using log scale for better visualization
plt.tight_layout()
plt.show()
```

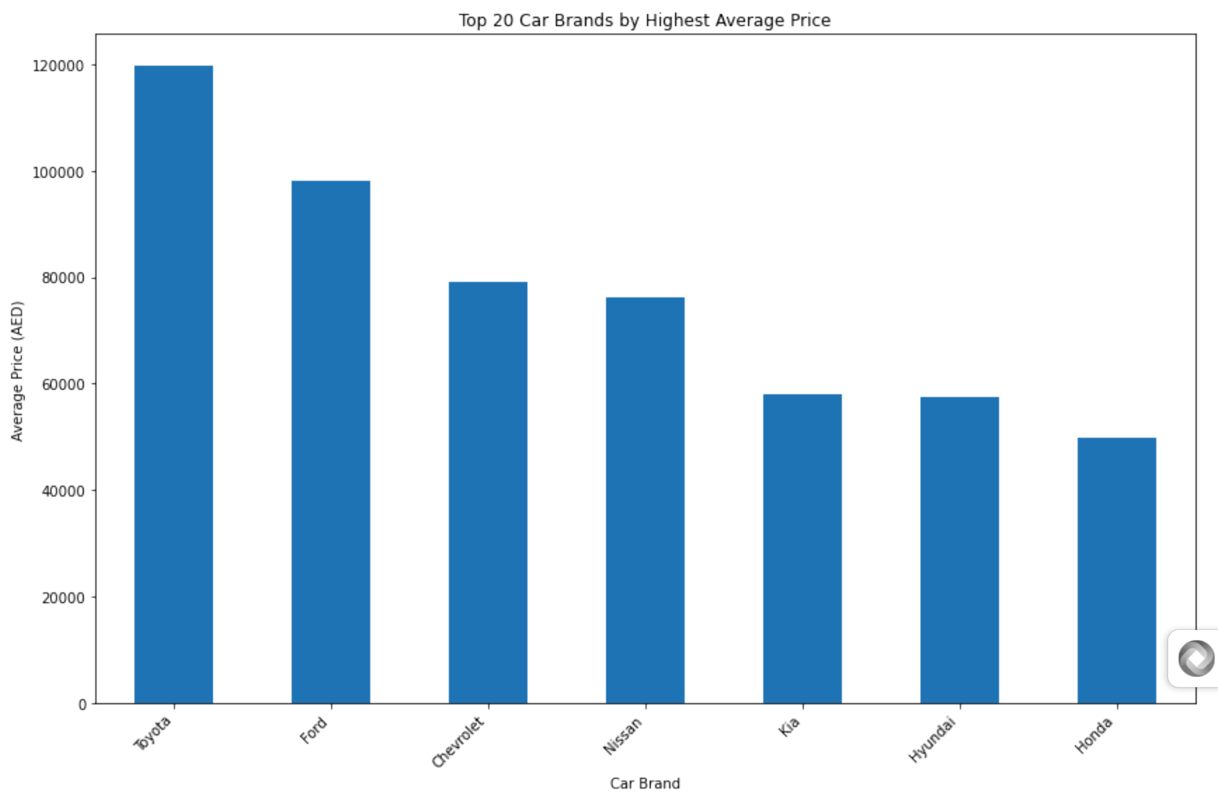


The Ford GT models dominate the top spots with prices ranging from 2,649,000 to 4,750,000 AED. The average price of cars in the dataset is approximately 87,966 AED, with a standard deviation of 113,182 AED. Toyota, Ford, and Chevrolet are among the top brands with the highest average prices. Toyota also has the highest number of cars listed.

In [28]:

```python
# Calculate average price for each brand
brand_avg_price = df.groupby('Car Brand')['Price'].mean().sort_values(ascending=False)

# Get top 20 brands
top_20_brands = brand_avg_price.head(20)

print("Top 20 Car Brands by Highest Average Price:")
print(top_20_brands)

# Create a bar plot
plt.figure(figsize=(12, 8))
top_20_brands.plot(kind='bar')
plt.title('Top 20 Car Brands by Highest Average Price')
plt.xlabel('Car Brand')
plt.ylabel('Average Price (AED)')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```

```
Top 20 Car Brands by Highest Average Price:
Car Brand
Toyota       119731.127676
Ford          98059.913651
Chevrolet     79095.475610
Nissan        76283.303059
Kia           57978.809826
Hyundai       57566.777647
Honda         49913.512069
Name: Price, dtype: float64
```



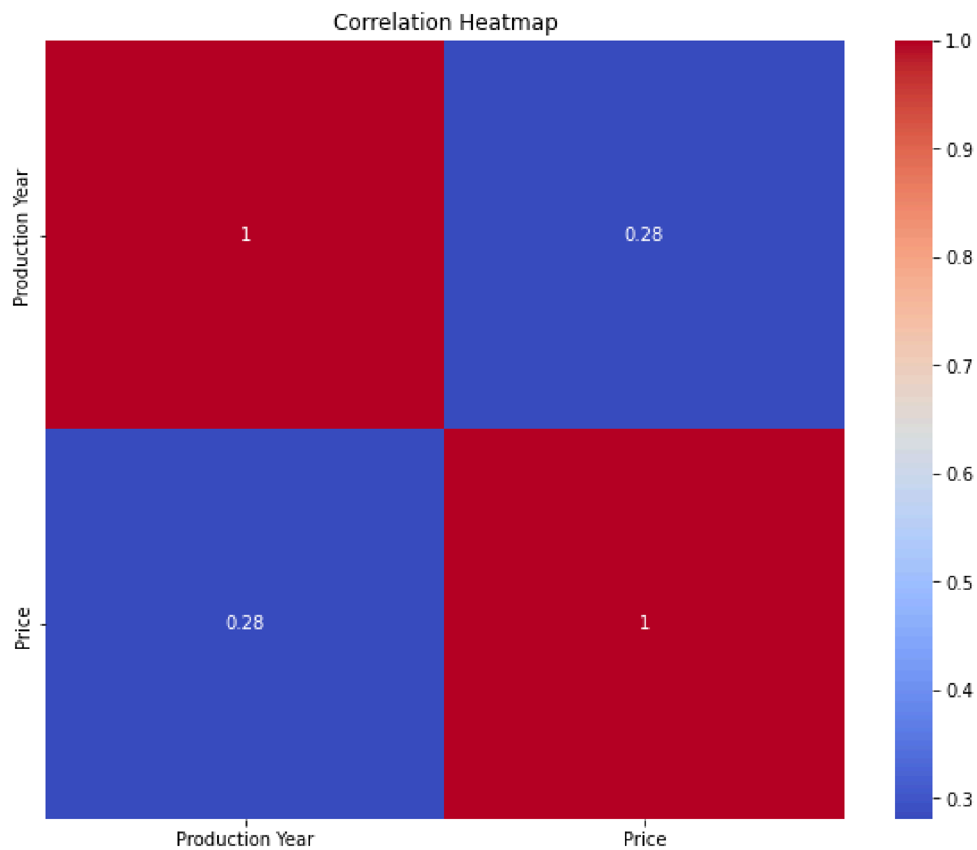
In [2]:

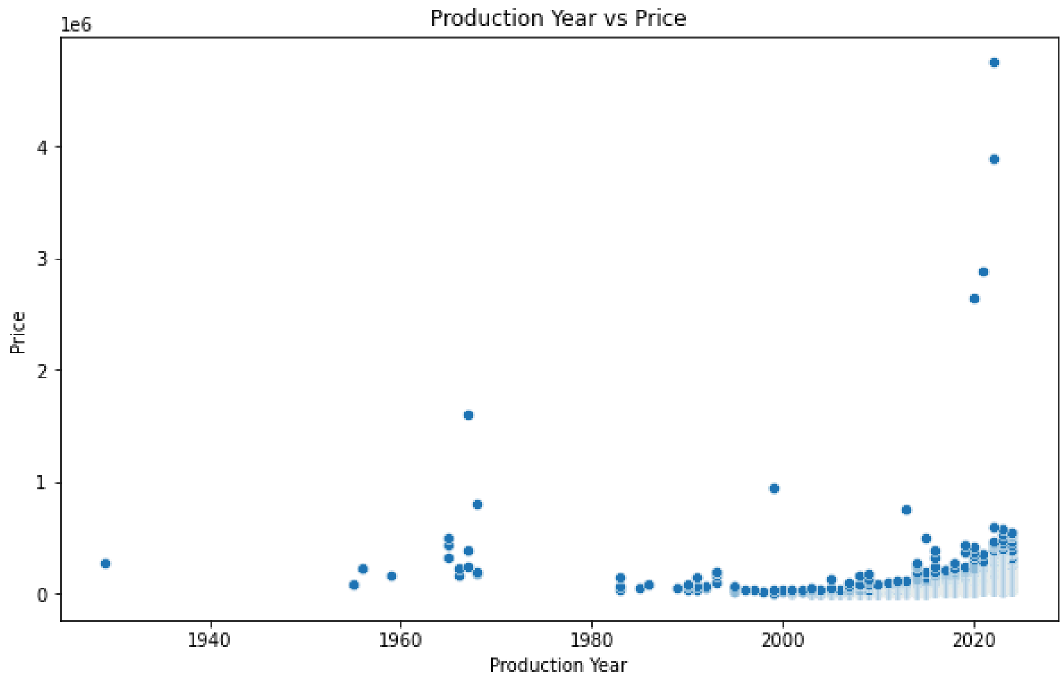
```python
# Step 2: Data Cleaning
# Convert Mileage and Price to numeric values
df['Mileage'] = df['Mileage'].str.replace(' km', '').str.replace(',', '').astype(float)
df['Price'] = df['Price'].str.replace(',', '').astype(float)

# Handle missing values (if any)
df = df.dropna()

# Encode categorical variables
df['Brand_Encoded'] = df['Car Brand'].astype('category').cat.codes
df['Model_Encoded'] = df['Car Model'].astype('category').cat.codes
df['Specs_Encoded'] = df['Specs'].astype('category').cat.codes

# Display the cleaned dataframe
print(df.head())
```

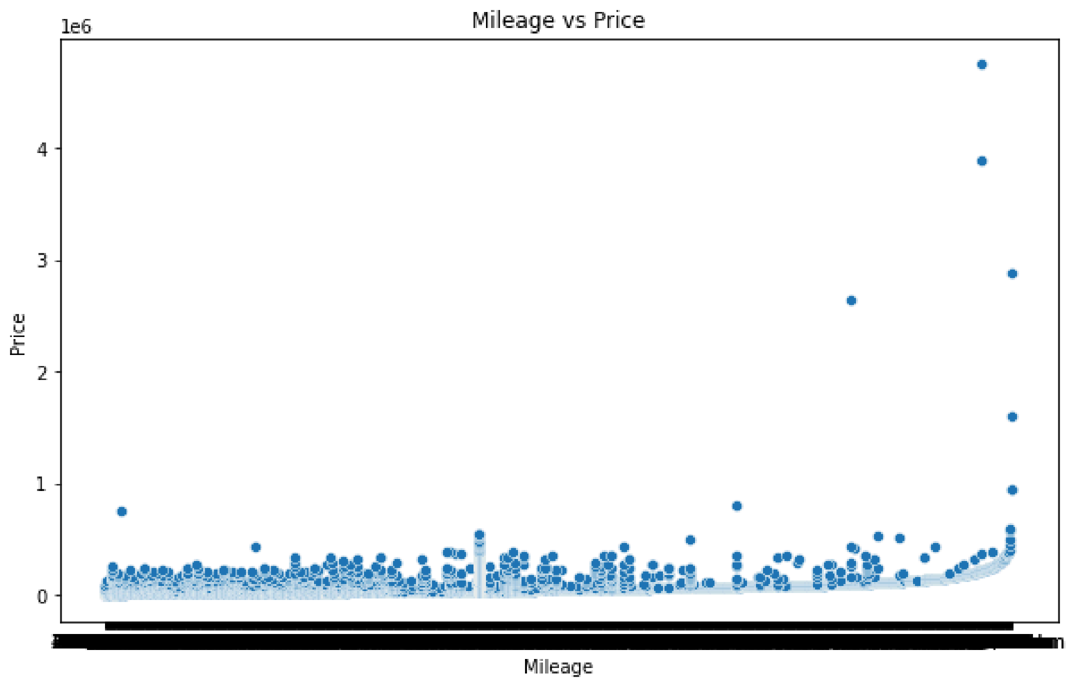
```
   Car Brand Car Model  Production Year   Mileage   Price  \
0     Nissan    Altima             2005  445740.0  3500.0
1     Toyota     Camry             1999  200000.0  5500.0
2       Ford     Focus             2006  366135.0  5500.0
3     Toyota      Echo             2005  200000.0  6000.0
4  Chevrolet     Epica             2009  250000.0  6000.0

                                  Description          Specs  \
0                                       Dubai      GCC Specs
1                Perfect Condition Toyota Camry      GCC Specs
2                                  FORD FOCUS      GCC Specs
3  GCC - TOYOTA ECHO 2005 - Manual, Urgent Sale      GCC Specs
4                              Chevrolet Epica  American Specs

        Timestamp   Location  Brand_Encoded  Model_Encoded  Specs_Encoded
0  04-03-24 14:49      Dubai              5              8              4
1  04-03-24 14:49      Dubai              6             24              4
2  04-03-24 14:49      Dubai              1             60              4
3  04-03-24 14:49      Dubai              6             44              4
4     45354.94097  Abu Dhabi              0             48              0
```

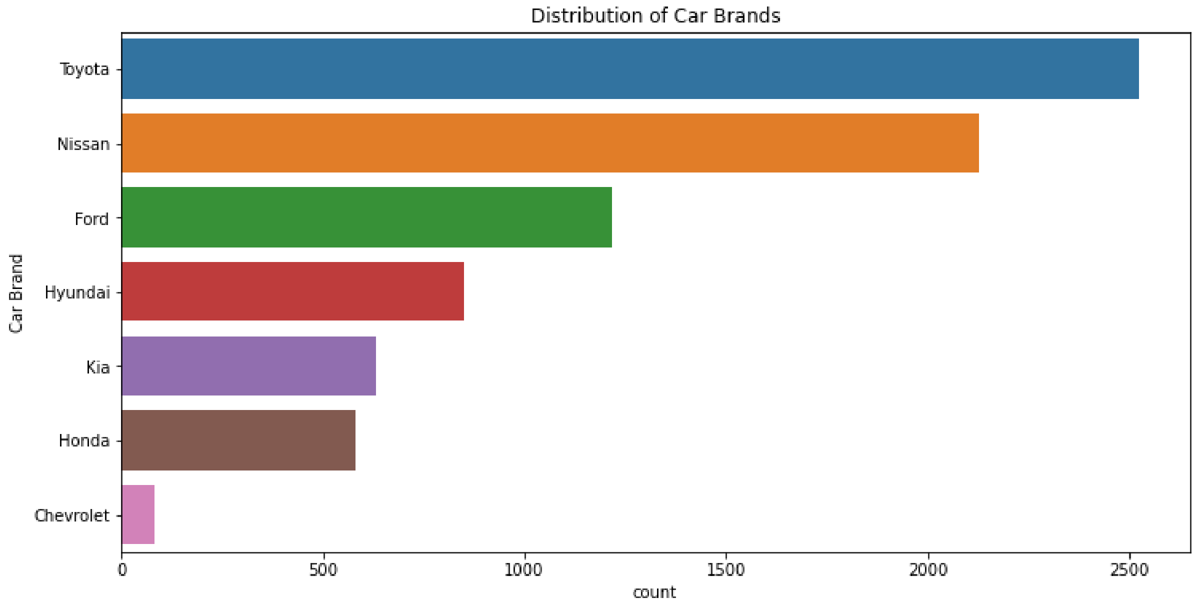
In [16]:

```python
# Correlation heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()

# Scatter plot: Production Year vs Price
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Production Year', y='Price', data=df)
plt.title('Production Year vs Price')
plt.show()

# Scatter plot: Mileage vs Price
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Mileage', y='Price', data=df)
plt.title('Mileage vs Price')
plt.show()

# Distribution of car brands
plt.figure(figsize=(12, 6))
sns.countplot(y='Car Brand', data=df, order=df['Car Brand'].value_counts().index)
plt.title('Distribution of Car Brands')
plt.show()
```

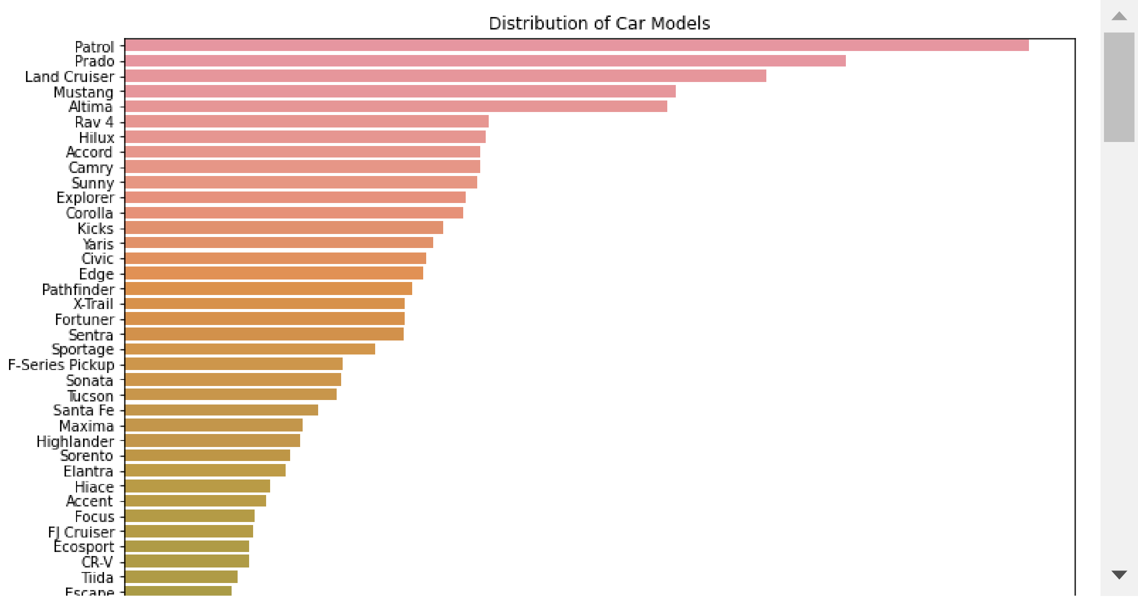



Production Year vs Price
1e6
4
3
2
1
0
Price
1940
1960
1980
2000
2020
Production Year


Mileage vs Price
1e6
4
3
2
1
0
Price
Mileage



In [6]:
```
# Distribution of car models
plt.figure(figsize=(12, 36))
sns.countplot(y='Car Model', data=df, order=df['Car Model'].value_counts().index)
plt.title('Distribution of Car Models')
plt.show()
```



In [7]:
```
# Step 4: Feature Engineering
# Select features and target variable
features = ['Production Year', 'Mileage', 'Brand_Encoded', 'Model_Encoded', 'Specs_Encod
target = 'Price'

X = df[features]
y = df[target]

# Split the data into training and testing sets
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 [10]:

```python
# Step 5: Modeling
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score

# Initialize and train the model
model = RandomForestRegressor(random_state=42)
model.fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)

# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print('Mean Squared Error:', mse)
print('R-squared Score:', r2)

# Feature importance
feature_importance = pd.DataFrame({'feature': features, 'importance': model.feature_impo
print('Feature Importance:')
print(feature_importance.sort_values(by='importance', ascending=False))
```

```
Mean Squared Error: 2467218384.11639
R-squared Score: 0.632873800374377
Feature Importance:
           feature  importance
3    Model_Encoded    0.428705
1          Mileage    0.296152
0  Production Year    0.166136
2    Brand_Encoded    0.064026
4    Specs_Encoded    0.044981
```

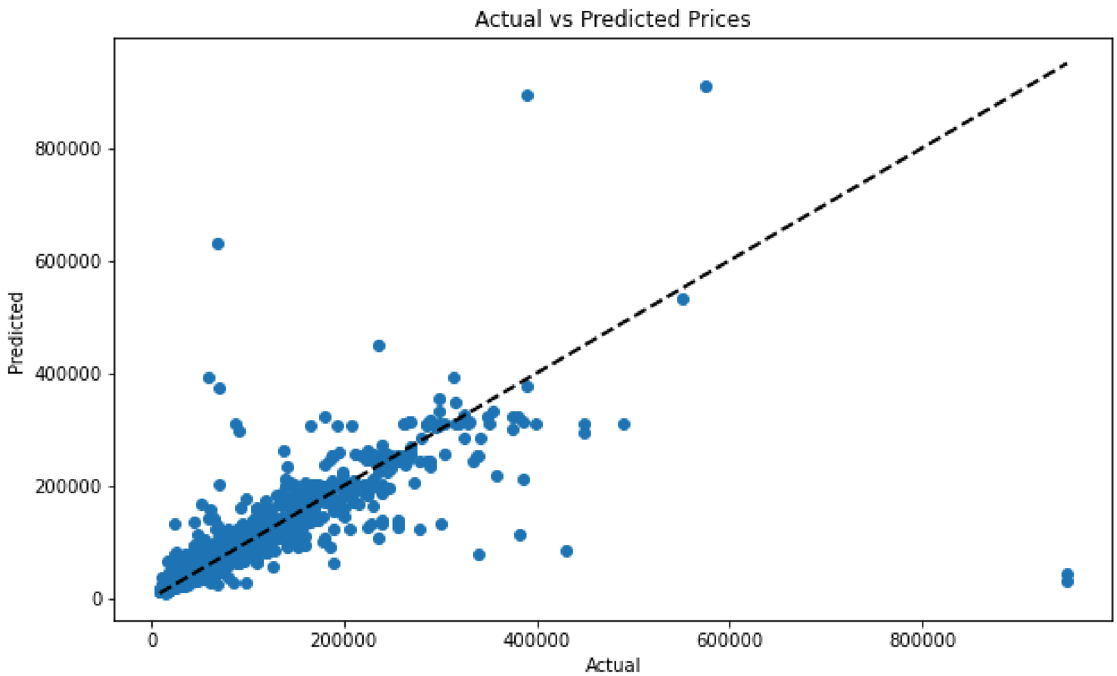
Mean Squared Error: 2467218384.11639

R-squared Score: 0.632873800374377

The R-squared score of about 0.63 indicates that our model explains approximately 63% of the variance in car prices. This suggests that while our model has some predictive power, there are likely other factors influencing car prices that aren't captured in our dataset.

In [9]:

```
# Step 6: Visualizations
# Actual vs Predicted Prices
plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_pred)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'k--', lw=2)
plt.xlabel('Actual')
plt.ylabel('Predicted')
plt.title('Actual vs Predicted Prices')
plt.show()
```



This scatter plot compares the actual prices of cars in our test set with the prices predicted by our model. The diagonal line would represent perfect predictions. The spread of points around this line indicates the model's accuracy.

Based on this analysis, we can conclude that Mileage and Production Year are the most influential factors in determining a used car's price in this UAE dataset. The car's brand also plays a role, with luxury brands generally commanding higher prices. However, there are likely other factors not captured in this dataset that also influence car prices, as our model only explains about 63% of the price variance.

In [ ]: