Project: Milestone 5

connect to db and load data into data frames

Project : MODULE 5

In [ ]:
```
import pandas as pd
import sqlalchemy

# Connect to the SQL database
db_username = 'root'
db_password = ''
db_host = 'localhost'
db_name = 'module5'

# Construct the connection string
connection_str = f'mysql+pymysql://{db_username}:{db_password}@{db_host}/{db_name}'

# Create the database engine
engine = sqlalchemy.create_engine(connection_str)

# Retrieve the list of tables in the database
table_names = engine.table_names()

# Load each table into separate dataframes
dataframes = {}
for table_name in table_names:
    query = f"SELECT * FROM {table_name}"
    dataframe = pd.read_sql_query(query, engine)
    dataframes[table_name] = dataframe

# Print the loaded dataframes
for table_name, dataframe in dataframes.items():
    print(f"Table Name: {table_name}")
    print(dataframe.head())
    print()

# Read the CSV files
products_df = dataframes[0]
exchange_rates_df = dataframes[1]
product_reviews_df = dataframes[2]
```

we can directly load data from files

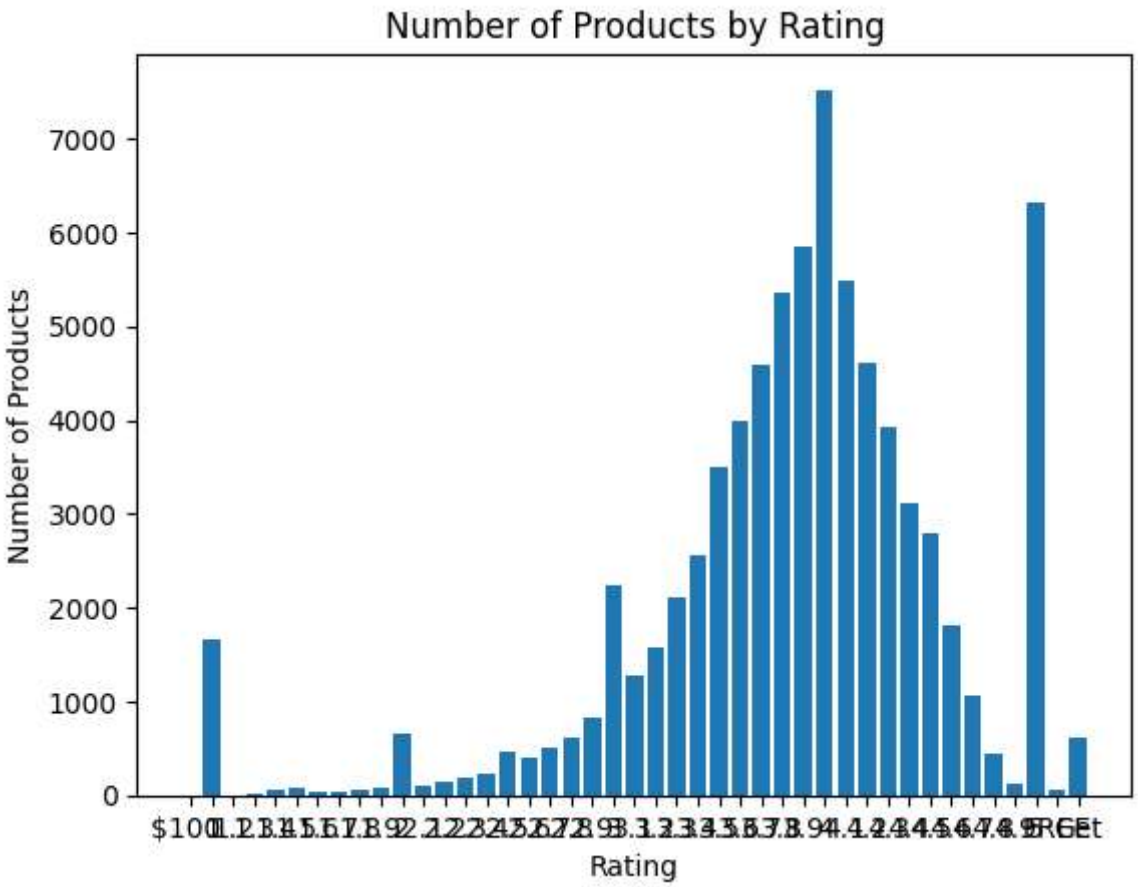
In [4]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Read the CSV files
products_df = pd.read_csv('/content/products.csv')
exchange_rates_df = pd.read_csv('/content/exchangeRate.csv')
product_reviews_df = pd.read_csv('/content/amazon_vfl_reviews.csv')
```

```
# Inspect the data
print(products_df.head())
print(exchange_rates_df.head())
print(product_reviews_df.head())
```

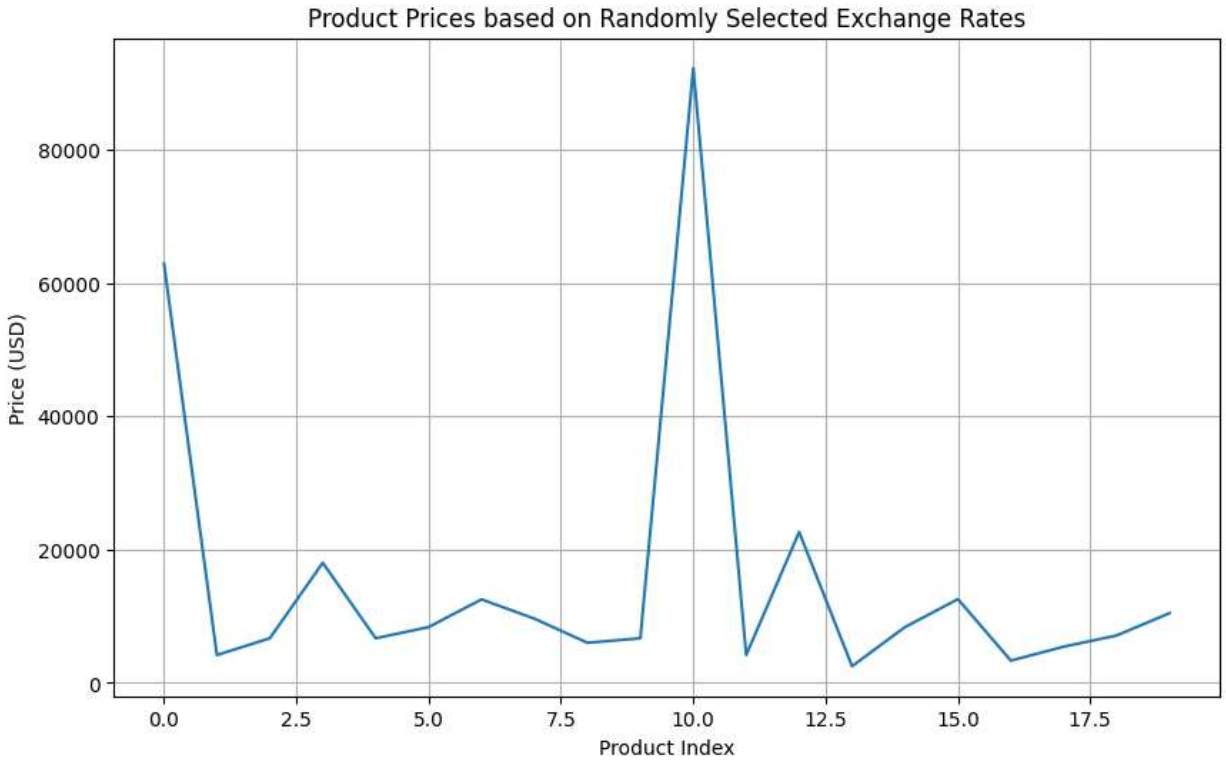
```
                                                name main_category  \
0  Lloyd 1.5 Ton 3 Star Inverter Split Ac (5 In 1...    appliances
1  LG 1.5 Ton 5 Star AI DUAL Inverter Split AC (C...    appliances
2  LG 1 Ton 4 Star Ai Dual Inverter Split Ac (Cop...    appliances
3  LG 1.5 Ton 3 Star AI DUAL Inverter Split AC (C...    appliances
4  Carrier 1.5 Ton 3 Star Inverter Split AC (Copp...    appliances

       sub_category                                              image  \
0  Air Conditioners  https://m.media-amazon.com/images/I/31UISB90sY...
1  Air Conditioners  https://m.media-amazon.com/images/I/51JFb7FctD...
2  Air Conditioners  https://m.media-amazon.com/images/I/51JFb7FctD...
3  Air Conditioners  https://m.media-amazon.com/images/I/51JFb7FctD...
4  Air Conditioners  https://m.media-amazon.com/images/I/41lrtqXPiW...

                                                link ratings no_of_ratings  \
0  https://www.amazon.in/Lloyd-Inverter-Convertib...     4.2         2,255
1  https://www.amazon.in/LG-Convertible-Anti-Viru...     4.2         2,948
2  https://www.amazon.in/LG-Inverter-Convertible-...     4.2         1,206
3  https://www.amazon.in/LG-Convertible-Anti-Viru...       4            69
4  https://www.amazon.in/Carrier-Inverter-Split-C...     4.1           630

  discount_price actual_price
0       $32,999      $58,990
1       $46,490      $75,990
2       $34,490      $61,990
3       $37,990      $68,990
4       $34,490      $67,790
   Unnamed: 0        date      rate
0           0  2021-01-01  4.050499
1           1  2021-01-02  4.050499
2           2  2021-01-03  4.022499
3           3  2021-01-04  4.006499
4           4  2021-01-05  4.017501
         asin                                        name        date  rating  \
0  B07W7CTLD1  Mamaearth-Onion-Growth-Control-Redensyl  2019-09-06       1
1  B07W7CTLD1  Mamaearth-Onion-Growth-Control-Redensyl  2019-08-14       5
2  B07W7CTLD1  Mamaearth-Onion-Growth-Control-Redensyl  2019-10-19       1
3  B07W7CTLD1  Mamaearth-Onion-Growth-Control-Redensyl  2019-09-16       1
4  B07W7CTLD1  Mamaearth-Onion-Growth-Control-Redensyl  2019-08-18       5

                                              review
0  I bought this hair oil after viewing so many g...
1  Used This Mama Earth Newly Launched Onion Oil ...
2  So bad product...My hair falling increase too ...
3  Product just smells similar to navarathna hair...
4  I have been trying different onion oil for my ...
```

In [6]:
```
# Count the number of products in each rating category
rating_counts = products_df['ratings'].value_counts().sort_index()

# Plotting the counts
plt.bar(rating_counts.index, rating_counts.values)
plt.xlabel('Rating')
plt.ylabel('Number of Products')
```

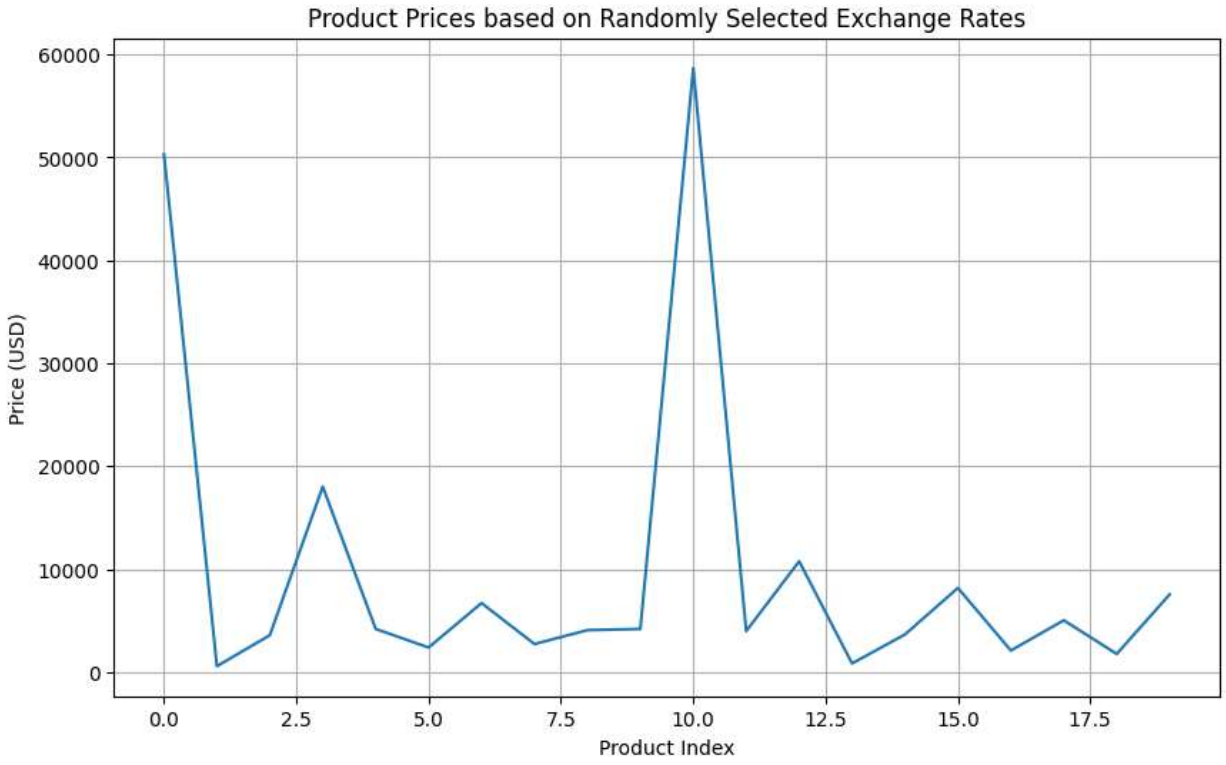
```
plt.title('Number of Products by Rating')
plt.show()
```



In [21]:
```
random_products = products_df.sample(n=20, random_state=42)

# Create an empty list to store the product prices
product_prices = []

# Iterate over the random products and extract their prices based on random exchange r
for _, product in random_products.iterrows():
    exchange_rate = exchange_rates_df.sample(n=1, random_state=42)['rate'].values[0]
    price_integer = int(product['actual_price'].replace('$', '').replace(',', ''))
    product_price = price_integer * float(exchange_rate)
    product_prices.append(product_price)

# Create a line graph to show product prices
plt.figure(figsize=(10, 6))
plt.plot(product_prices)
plt.xlabel('Product Index')
plt.ylabel('Price (USD)')
plt.title('Product Prices based on Randomly Selected Exchange Rates')
plt.grid(True)
plt.show()
```

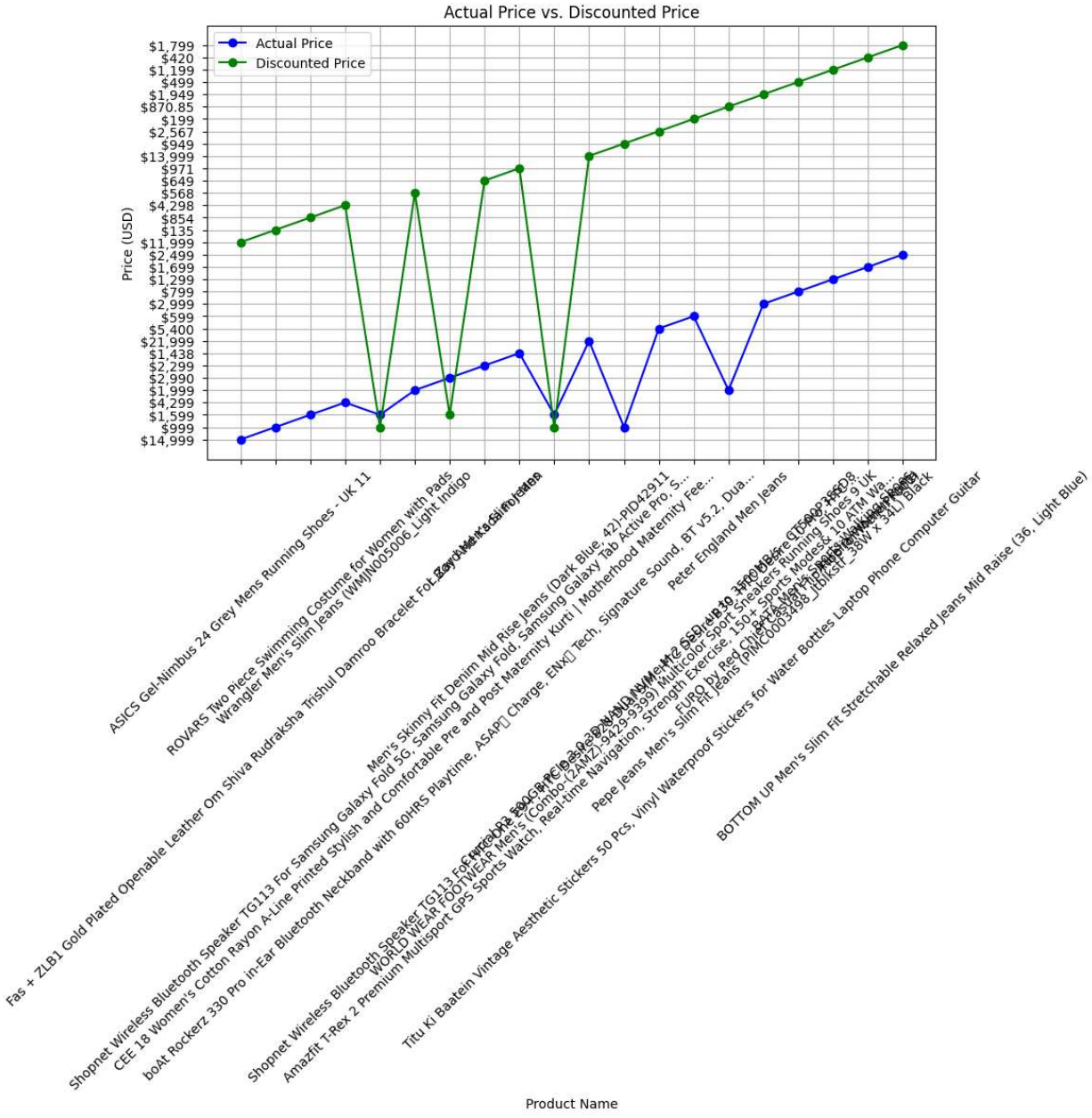


In [24]:
```
# Randomly select 20 products
random_products = products_df.sample(n=20, random_state=42)

# Create an empty list to store the product prices
product_prices = []

# Iterate over the random products and extract their prices based on random exchange r
for _, product in random_products.iterrows():
    exchange_rate = exchange_rates_df.sample(n=1, random_state=42)['rate'].values[0]
    price_integer = float(product['discount_price'].replace('$', '').replace(',', ''))
    product_price = price_integer * float(exchange_rate)
    product_prices.append(product_price)

# Create a line graph to show product prices
plt.figure(figsize=(10, 6))
plt.plot(product_prices)
plt.xlabel('Product Index')
plt.ylabel('Price (USD)')
plt.title('Product Discounted Prices based on Randomly Selected Exchange Rates')
plt.grid(True)
plt.show()
```

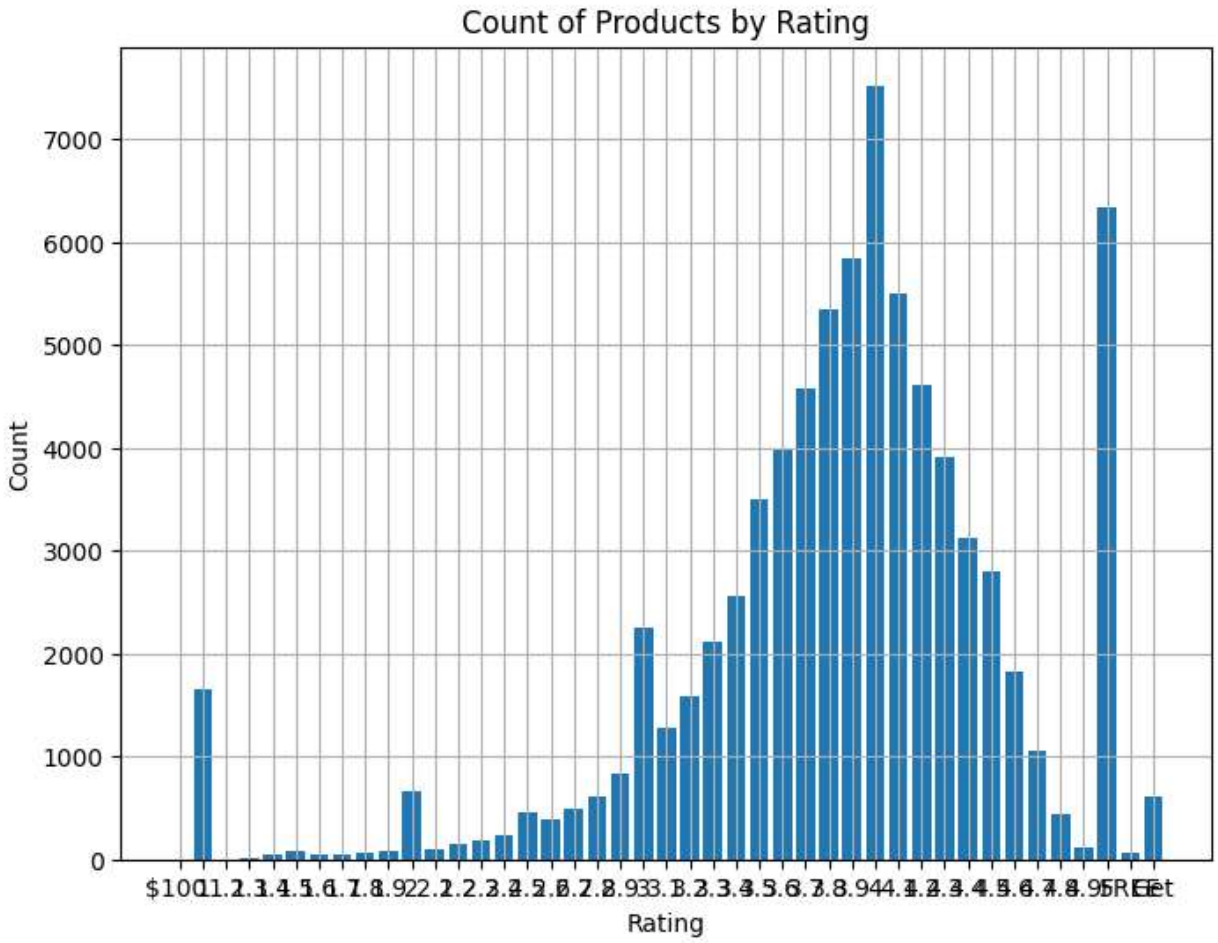


In [29]:
```
# Randomly select 20 products
random_products = products_df.sample(n=20, random_state=42)

# Get the names, actual prices, and discounted prices of the randomly selected product
names = random_products['name']
actual_prices = random_products['actual_price']
discounted_prices = random_products['discount_price']

# Create a line graph to show actual_price and discount_price
plt.figure(figsize=(10, 6))
plt.plot(names, actual_prices, marker='o', color='blue', label='Actual Price')
plt.plot(names, discounted_prices, marker='o', color='green', label='Discounted Price'
plt.xlabel('Product Name')
plt.ylabel('Price (USD)')
plt.title('Actual Price vs. Discounted Price')
plt.xticks(rotation=45)
plt.legend()
plt.grid(True)
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py:151: UserWarning:
Glyph 153 (\x99) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
```

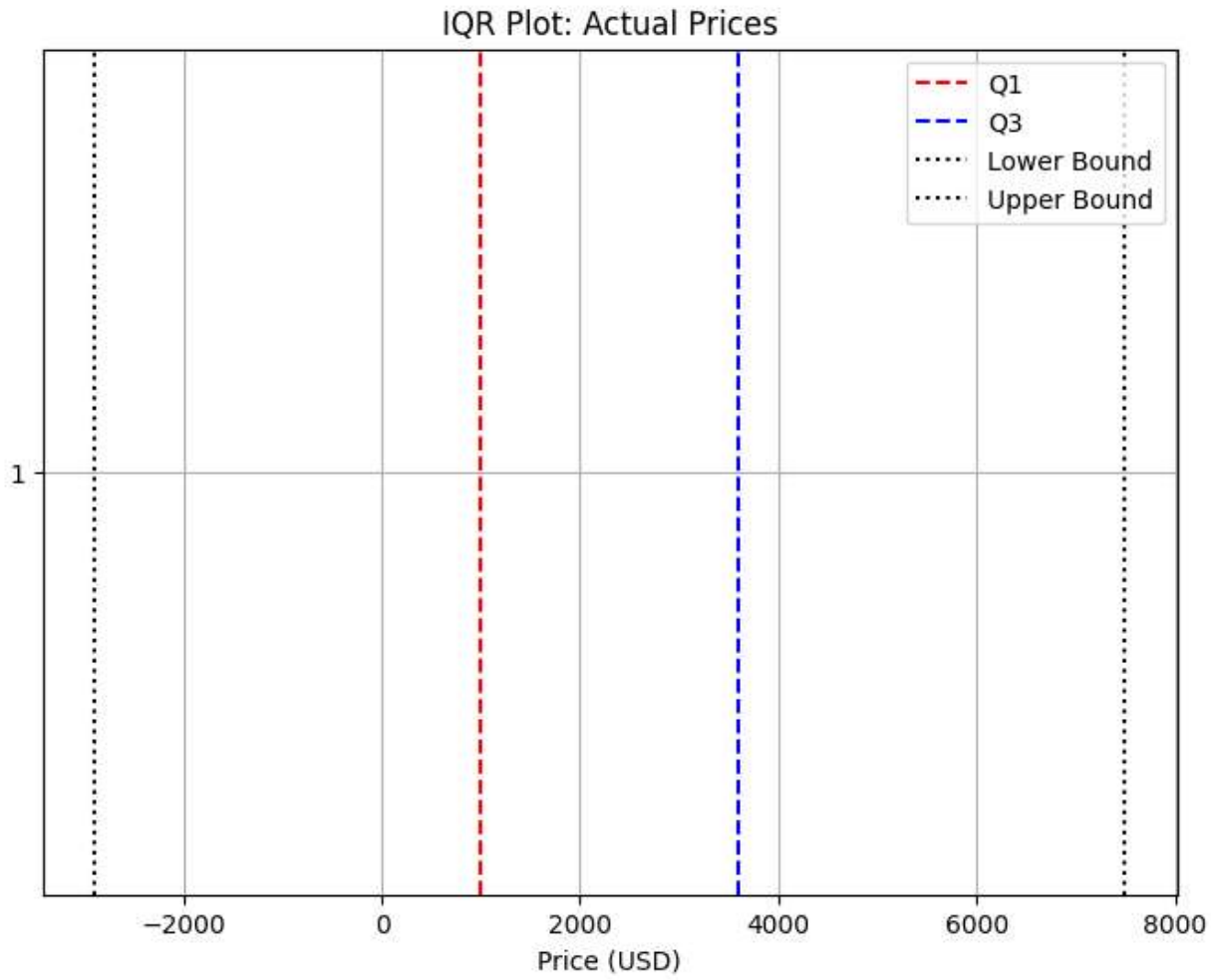


In [31]:
```
# Count the number of products in each rating category
rating_counts = products_df['ratings'].value_counts().sort_index()

# Create a histogram
plt.figure(figsize=(8, 6))
plt.bar(rating_counts.index, rating_counts.values)
plt.xlabel('Rating')
plt.ylabel('Count')
plt.title('Count of Products by Rating')
plt.xticks(rating_counts.index)
plt.grid(True)
plt.show()
```



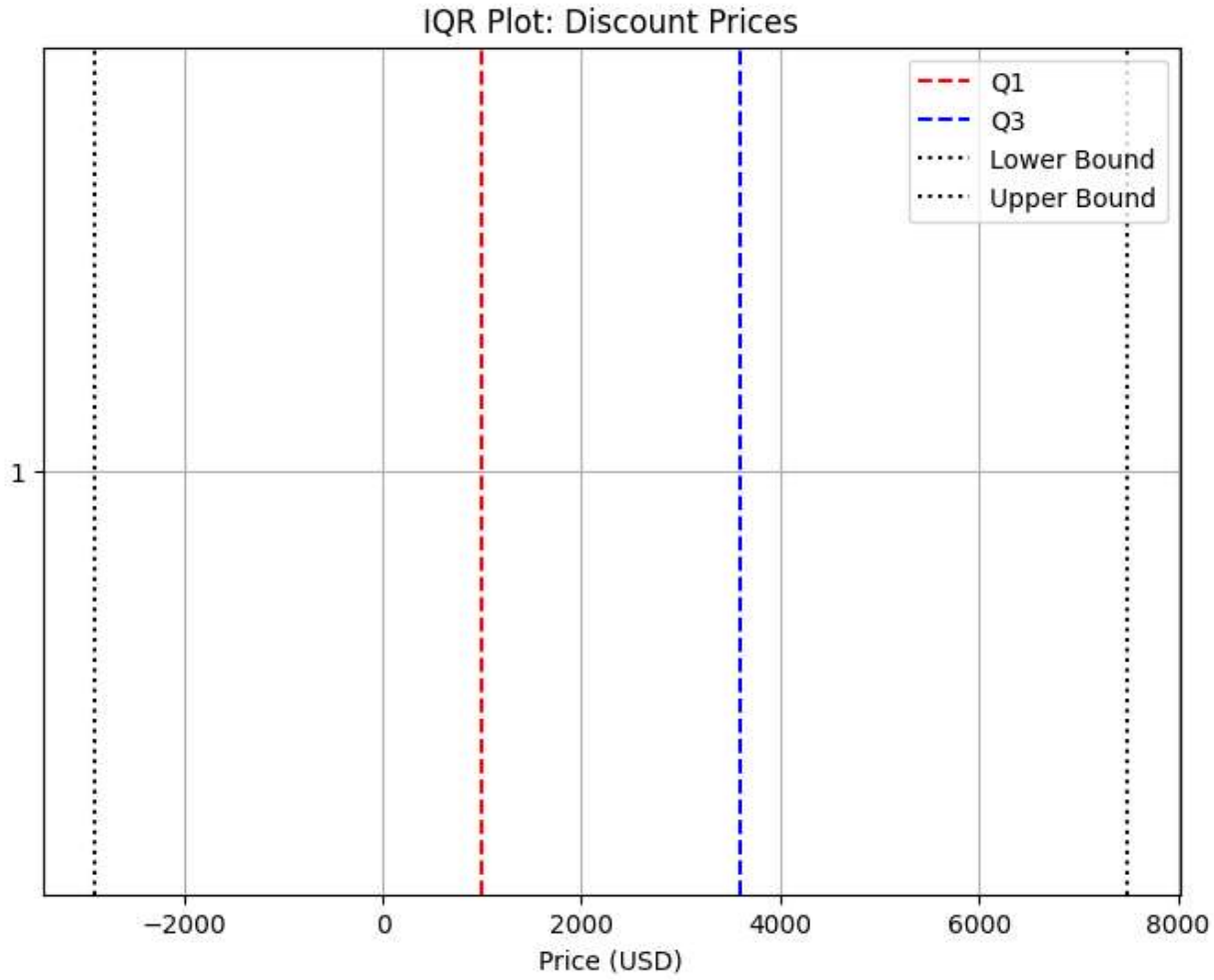
In [42]:
```
# Convert actual_price column to numeric format
products_df['actual_price'] = products_df['actual_price'].str.replace('$', '').str.rep

# Calculate the IQR for actual prices
q1 = products_df['actual_price'].quantile(0.25)
q3 = products_df['actual_price'].quantile(0.75)
iqr = q3 - q1

# Create the IQR plot
plt.figure(figsize=(8, 6))
plt.boxplot(products_df['actual_price'], vert=False, showfliers=False, widths=0.5)
plt.axvline(x=q1, color='red', linestyle='--', label='Q1')
plt.axvline(x=q3, color='blue', linestyle='--', label='Q3')
plt.axvline(x=q1 - 1.5 * iqr, color='black', linestyle=':', label='Lower Bound')
plt.axvline(x=q3 + 1.5 * iqr, color='black', linestyle=':', label='Upper Bound')
plt.xlabel('Price (USD)')
plt.title('IQR Plot: Actual Prices')
plt.legend()
plt.grid(True)
plt.show()
```

```
<ipython-input-42-24c100fa5281>:2: FutureWarning: The default value of regex will cha
nge from True to False in a future version. In addition, single character regular exp
ressions will *not* be treated as literal strings when regex=True.
  products_df['actual_price'] = products_df['actual_price'].str.replace('$', '').str.
replace(',', '').astype(float)
```



In [43]:
```
# Convert actual_price column to numeric format
products_df['discount_price'] = products_df['discount_price'].str.replace('$', '').str

# Calculate the IQR for actual prices
q1 = products_df['actual_price'].quantile(0.25)
q3 = products_df['actual_price'].quantile(0.75)
iqr = q3 - q1

# Create the IQR plot
plt.figure(figsize=(8, 6))
plt.boxplot(products_df['discount_price'], vert=False, showfliers=False, widths=0.5)
plt.axvline(x=q1, color='red', linestyle='--', label='Q1')
plt.axvline(x=q3, color='blue', linestyle='--', label='Q3')
plt.axvline(x=q1 - 1.5 * iqr, color='black', linestyle=':', label='Lower Bound')
plt.axvline(x=q3 + 1.5 * iqr, color='black', linestyle=':', label='Upper Bound')
plt.xlabel('Price (USD)')
plt.title('IQR Plot: Discount Prices')
plt.legend()
plt.grid(True)
plt.show()
```

```
<ipython-input-43-7e98bb1f34ad>:2: FutureWarning: The default value of regex will cha
nge from True to False in a future version. In addition, single character regular exp
ressions will *not* be treated as literal strings when regex=True.
  products_df['discount_price'] = products_df['discount_price'].str.replace('$', '').
str.replace(',', '').astype(float)
```


IQR Plot: Discount Prices
Q1
Q3
Lower Bound
Upper Bound
1
−2000
0
2000
4000
6000
8000
Price (USD)

In [ ]: