In [1]:
```
import numpy as np
import pandas as pd
import matplotlib as plt
import seaborn as sns
```

In [2]:
```
df = pd.read_csv(r"C:\Users\prach\Downloads\ATLIQ.csv")
```

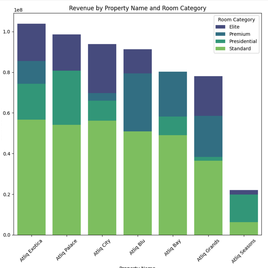
In [3]:
```
df.head(3)
```

Out[3]:

	property_id	booking_date	check_in_date	checkout_date	no_guests	room	booking_platform	ratings_given	booking_status	revenue_generated	revenue_realized	ro
0	16558	27-04-2022	01-05-2022	02-05-2022	3	RT1	direct online	1.0	Checked Out	10010	10010	
1	16558	30-04-2022	01-05-2022	02-05-2022	2	RT1	others	NaN	Cancelled	9100	3640	
2	16558	28-04-2022	01-05-2022	04-05-2022	2	RT1	logtrip	5.0	Checked Out	9100	9100	

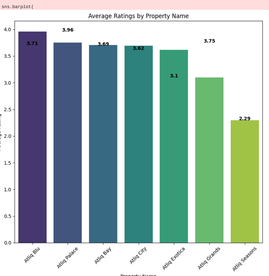
In [4]:
```
import matplotlib.pyplot as plt
```

In [29]:
```
Revenue = df.groupby(['prop name', 'room_category']).agg(sum_revenue=('revenue_realized', 'sum')).reset_index()
Revenue = Revenue.sort_values(by='sum_revenue', ascending=False)
plt.figure(figsize=(8, 6))
p13 = sns.barplot(
    x='prop name', y='sum_revenue', hue='room_category',
    data=Revenue, dodge=False, palette='viridis', order=Revenue['prop name'].unique()
)

# Adding text labels
plt.xticks(rotation=45)
plt.xlabel('Property Name')
plt.ylabel('Sum Revenue')
plt.title('Revenue by Property Name and Room Category')
plt.legend(title='Room Category')
plt.tight_layout()
plt.show()
```

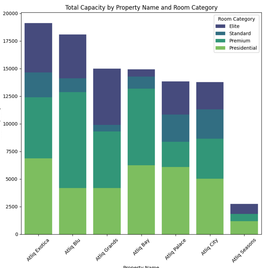


In [28]:
```
avg_ratings = df.groupby('prop name').agg(avg_rating=('ratings_given', 'mean')).reset_index()
avg_ratings = avg_ratings.sort_values(by='avg_rating', ascending=False)
plt.figure(figsize=(8, 6))
sns.barplot(
    x='prop name', y='avg_rating',
    data=avg_ratings, palette='viridis', order=avg_ratings['prop name']
)
for index, row in avg_ratings.iterrows():
    plt.text(index, row['avg_rating'], round(row['avg_rating'], 2), color='black', ha="center", size=10, weight='bold')

plt.xticks(rotation=45)
plt.xlabel('Property Name')
plt.ylabel('Average Rating')
plt.title('Average Ratings by Property Name')
plt.tight_layout()
plt.show()
```

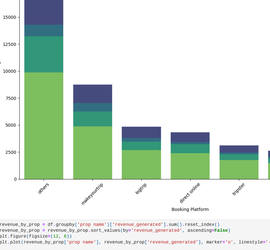
```
C:\Users\prach\AppData\Local\Temp\ipykernel_9624\4791974O6.py:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(
```



In [39]:
```
total_capacity = df.groupby(['prop name', 'room_category']).agg(total_capacity=('no_guests', 'sum')).reset_index()

# Sort the DataFrame by 'total_capacity' in descending order within each property
total_capacity = total_capacity.sort_values(by='total_capacity', ascending=False)

# Plot
plt.figure(figsize=(8, 8))
sns.barplot(
    x='prop name', y='total_capacity', hue='room_category',
    data=total_capacity, dodge=False, palette='viridis'
)

plt.xticks(rotation=45)
plt.xlabel('Property Name')
plt.ylabel('Total Capacity')
plt.title('Total Capacity by Property Name and Room Category')
plt.legend(title='Room Category')
plt.tight_layout()
plt.show()
```



In [36]:
```
booking_counts = df.groupby(['booking_platform', 'city']).size().reset_index(name='count')
booking_counts = booking_counts.sort_values(by='count', ascending=False)
plt.figure(figsize=(12, 8))
sns.barplot(
    x='booking_platform', y='count', hue='city',
    data=booking_counts, dodge=False, palette='viridis'
)
plt.xticks(rotation=45)
plt.xlabel('Booking Platform')
plt.ylabel('Count of Bookings')
plt.title('Count of Bookings by Booking Platform and City')
plt.legend(title='City')
plt.tight_layout()
plt.show()
```



In [37]:
```
revenue_by_prop = df.groupby('prop name')['revenue_generated'].sum().reset_index()
revenue_by_prop = revenue_by_prop.sort_values(by='revenue_generated', ascending=False)
plt.figure(figsize=(12, 6))
plt.plot(revenue_by_prop['prop name'], revenue_by_prop['revenue_generated'], marker='o', linestyle='-')

# Adding labels and title
plt.xlabel('Property Name')
plt.ylabel('Revenue Generated')
plt.title('Revenue Generated by Property Name')
plt.xticks(rotation=45)
plt.grid(True)
plt.tight_layout()
plt.show()
```

In [38]:
```
revenue_by_prop = df.groupby('city')['revenue_generated'].sum().reset_index()
revenue_by_prop = revenue_by_prop.sort_values(by='revenue_generated', ascending=False)
plt.figure(figsize=(12, 6))
plt.plot(revenue_by_prop['city'], revenue_by_prop['revenue_generated'], marker='o', linestyle='-')

# Adding labels and title
plt.xlabel('city')
plt.ylabel('Revenue Generated')
plt.title('Revenue Generated by city')
plt.xticks(rotation=45)
plt.grid(True)
plt.tight_layout()
plt.show()
```

In [44]:
```
filtered_data = df[df['booking_status'].isin(['Checked Out', 'Cancelled'])]
booking_counts = filtered_data['booking_status'].value_counts()
plt.figure(figsize=(8, 6))
booking_counts.plot(kind='bar', color=['blue', 'red'])

# Adding labels and title
plt.xlabel('Booking Status')
plt.ylabel('Count')
plt.title('Count of Bookings by Booking Status')
plt.xticks(rotation=0)
plt.tight_layout()
plt.show()
```