In [70]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [37]:
```
df = pd.read_excel(r'UberDataSet.xlsx')
```

In [38]:
```
df.head(2)
```

Out[38]:

	Request id	Pickup point	Status	Trip Status	timestamp
0	867	Airport	Trip Completed	Trip Completed	17:57:00
1	1807	City	Trip Completed	Trip Completed	09:17:00

In [39]:
```
df.isnull().sum() # no null values in the dataset
```

Out[39]:
```
Request id      0
Pickup point    0
Status          0
Trip Status     0
timestamp       0
dtype: int64
```

In [40]:
```
df.dtypes
```

Out[40]:
```
Request id       int64
Pickup point    object
Status          object
Trip Status     object
timestamp       object
dtype: object
```

In [41]:
```
df['timestamp'] = pd.to_datetime(df['timestamp'], format='%H:%M:%S')
```

In [42]:
```
df['timestamp']
```

Out[42]:
```
0      1900-01-01 17:57:00
1      1900-01-01 09:17:00
2      1900-01-01 21:08:00
3      1900-01-01 08:33:16
4      1900-01-01 21:57:28
               ...
6739   1900-01-01 23:49:03
6740   1900-01-01 23:50:05
6741   1900-01-01 23:52:06
6742   1900-01-01 23:54:39
6743   1900-01-01 23:55:03
Name: timestamp, Length: 6744, dtype: datetime64[ns]
```

In [43]:
```
df.dtypes
```

Out[43]:
```
Request id               int64
Pickup point            object
Status                  object
Trip Status             object
```

```
timestamp        datetime64[ns]
dtype: object
```

In [46]:

```
def categorize_period(timestamp):
    hour = timestamp.hour
    if 6 <= hour < 12:
        return 'morning'
    elif 12 <= hour < 18:
        return 'afternoon'
    elif 18 <= hour < 24:
        return 'evening'
    else:
        return 'night'
```

In [47]:

```
df['period'] = df['timestamp'].apply(categorize_period)
```

In [48]:

```
df['period']
```

Out[48]:

```
0        afternoon
1          morning
2          evening
3          morning
4          evening
             ...
6739       evening
6740       evening
6741       evening
6742       evening
6743       evening
Name: period, Length: 6744, dtype: object
```

In [49]:

```
df
```

Out[49]:

	Request id	Pickup point	Status	Trip Status	timestamp	period
0	867	Airport	Trip Completed	Trip Completed	1900-01-01 17:57:00	afternoon
1	1807	City	Trip Completed	Trip Completed	1900-01-01 09:17:00	morning
2	2532	Airport	Trip Completed	Trip Completed	1900-01-01 21:08:00	evening
3	3112	City	Trip Completed	Trip Completed	1900-01-01 08:33:16	morning
4	3879	Airport	Trip Completed	Trip Completed	1900-01-01 21:57:28	evening
...	...	...	...	...	...	...
6739	6745	City	No Cars Available	Trip Not Completed	1900-01-01 23:49:03	evening
6740	6752	Airport	No Cars Available	Trip Not Completed	1900-01-01 23:50:05	evening
6741	6751	City	No Cars Available	Trip Not Completed	1900-01-01 23:52:06	evening
6742	6754	City	No Cars Available	Trip Not Completed	1900-01-01 23:54:39	evening
6743	6753	Airport	No Cars Available	Trip Not Completed	1900-01-01 23:55:03	evening

6744 rows × 6 columns

In [50]:

```
df['timestamp'] = df['timestamp'].dt.round('H')
```

In [52]:
```
df['timestamp'] = df['timestamp'].dt.time
```

In [57]:
```
df
```

Out[57]:

	Request id	Pickup point	Status	Trip Status	timestamp	period
0	867	Airport	Trip Completed	Trip Completed	18:00:00	afternoon
1	1807	City	Trip Completed	Trip Completed	09:00:00	morning
2	2532	Airport	Trip Completed	Trip Completed	21:00:00	evening
3	3112	City	Trip Completed	Trip Completed	09:00:00	morning
4	3879	Airport	Trip Completed	Trip Completed	22:00:00	evening
...	...	...	...	...	...	...
6739	6745	City	No Cars Available	Trip Not Completed	00:00:00	evening
6740	6752	Airport	No Cars Available	Trip Not Completed	00:00:00	evening
6741	6751	City	No Cars Available	Trip Not Completed	00:00:00	evening
6742	6754	City	No Cars Available	Trip Not Completed	00:00:00	evening
6743	6753	Airport	No Cars Available	Trip Not Completed	00:00:00	evening

6744 rows × 6 columns

EDA

In [58]:
```
df.groupby('Status')['Request id'].count()
```

Out[58]:
```
Status
Cancelled            1264
No Cars Available    2650
Trip Completed       2830
Name: Request id, dtype: int64
```

In [62]:
```
Count_by_Trip_Status = df.groupby('Status')['Request id'].count().reset_index()
```

In [68]:
```
Count_by_Trip_Status = Count_by_Trip_Status.rename({'Request id':'Status_count'},axis=1)
```

In [69]:
```
Count_by_Trip_Status
```

Out[69]:

	Status	Status_count
0	Cancelled	1264
1	No Cars Available	2650
2	Trip Completed	2830

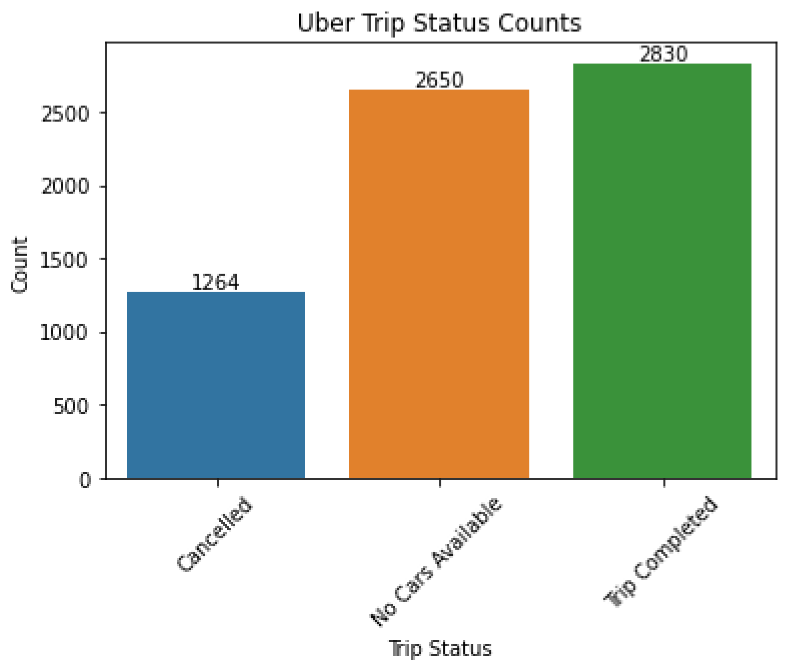
In [81]:
```
sns.barplot(x='Status', y='Status_count', data=Count_by_Trip_Status)

for index, value in enumerate(Count_by_Trip_Status['Status_count']):
    plt.text(index, value, str(value), ha='center', va='bottom')
```

```
# Set the labels and title
plt.xlabel('Trip Status')
plt.ylabel('Count')
plt.title('Uber Trip Status Counts')

# Rotate x-axis labels for better readability if needed
plt.xticks(rotation=45)

# Show the plot
plt.show()
```



In [82]:

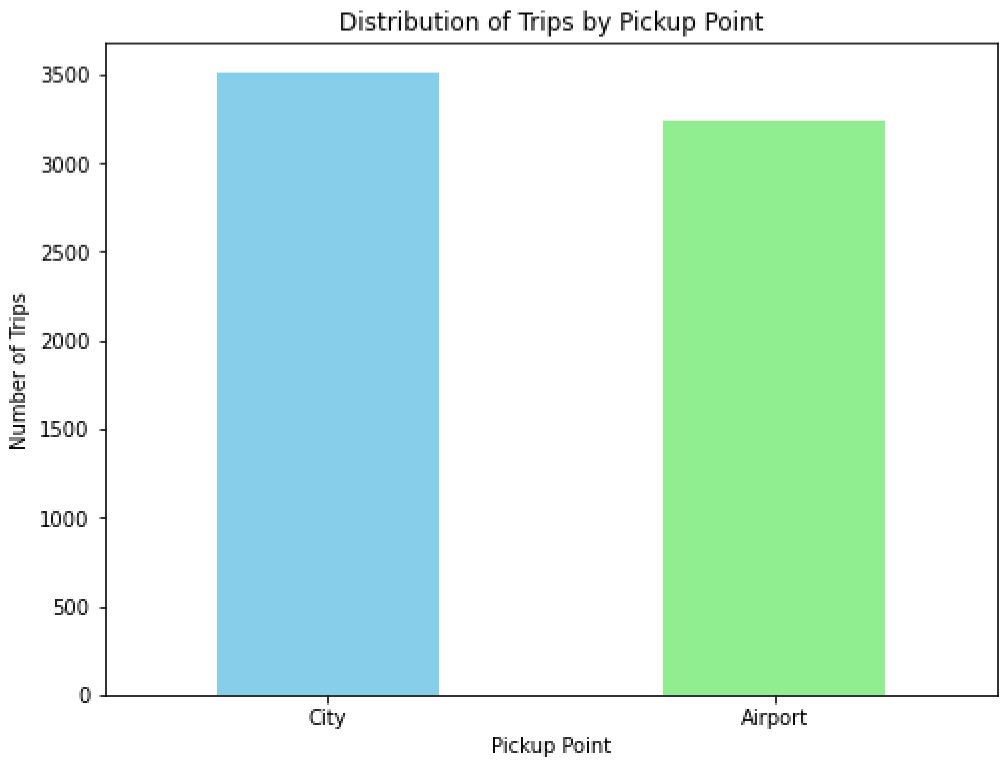
```
df.head()
```

Out[82]:

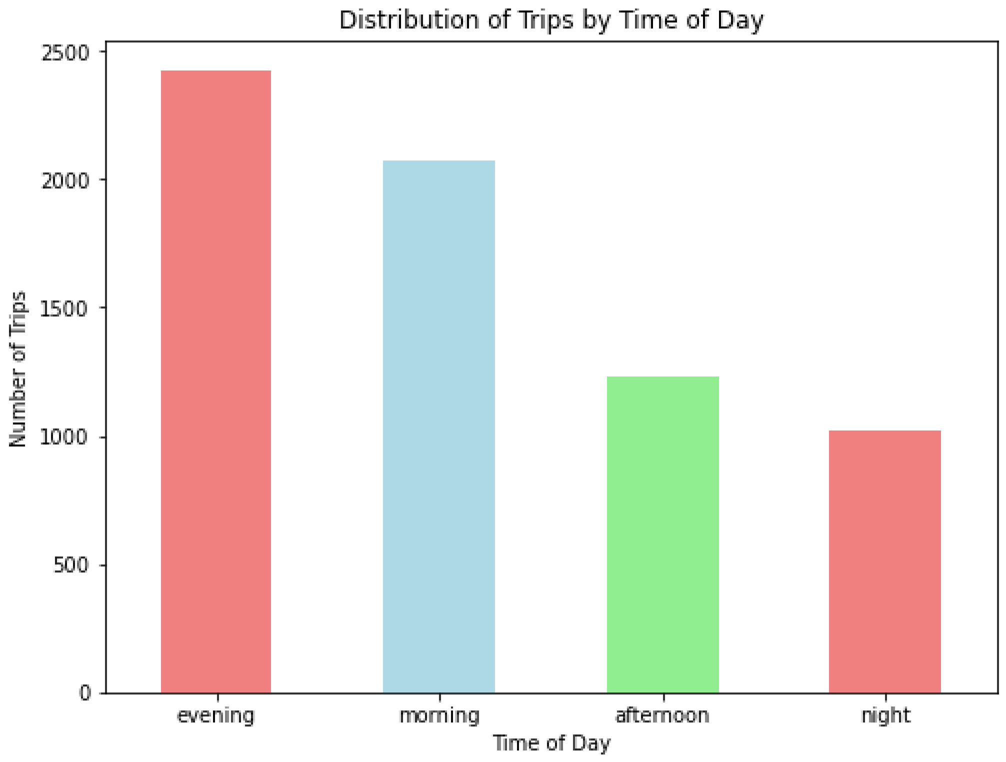
	Request id	Pickup point	Status	Trip Status	timestamp	period
0	867	Airport	Trip Completed	Trip Completed	18:00:00	afternoon
1	1807	City	Trip Completed	Trip Completed	09:00:00	morning
2	2532	Airport	Trip Completed	Trip Completed	21:00:00	evening
3	3112	City	Trip Completed	Trip Completed	09:00:00	morning
4	3879	Airport	Trip Completed	Trip Completed	22:00:00	evening

In [83]:

```
# Distribution of Trips by Pickup Point
pickup_point_counts = df['Pickup point'].value_counts()
plt.figure(figsize=(8, 6))
pickup_point_counts.plot(kind='bar', color=['skyblue', 'lightgreen'])
plt.title('Distribution of Trips by Pickup Point')
plt.xlabel('Pickup Point')
plt.ylabel('Number of Trips')
plt.xticks(rotation=0)
plt.show()
```



In [84]:
```
# Distribution of Trips by Time of Day
time_of_day_counts = df['period'].value_counts()
plt.figure(figsize=(8, 6))
time_of_day_counts.plot(kind='bar', color=['lightcoral', 'lightblue', 'lightgreen'])
plt.title('Distribution of Trips by Time of Day')
plt.xlabel('Time of Day')
plt.ylabel('Number of Trips')
plt.xticks(rotation=0)
plt.show()
```



In [86]:
```
df.head()
```

Out[86]:

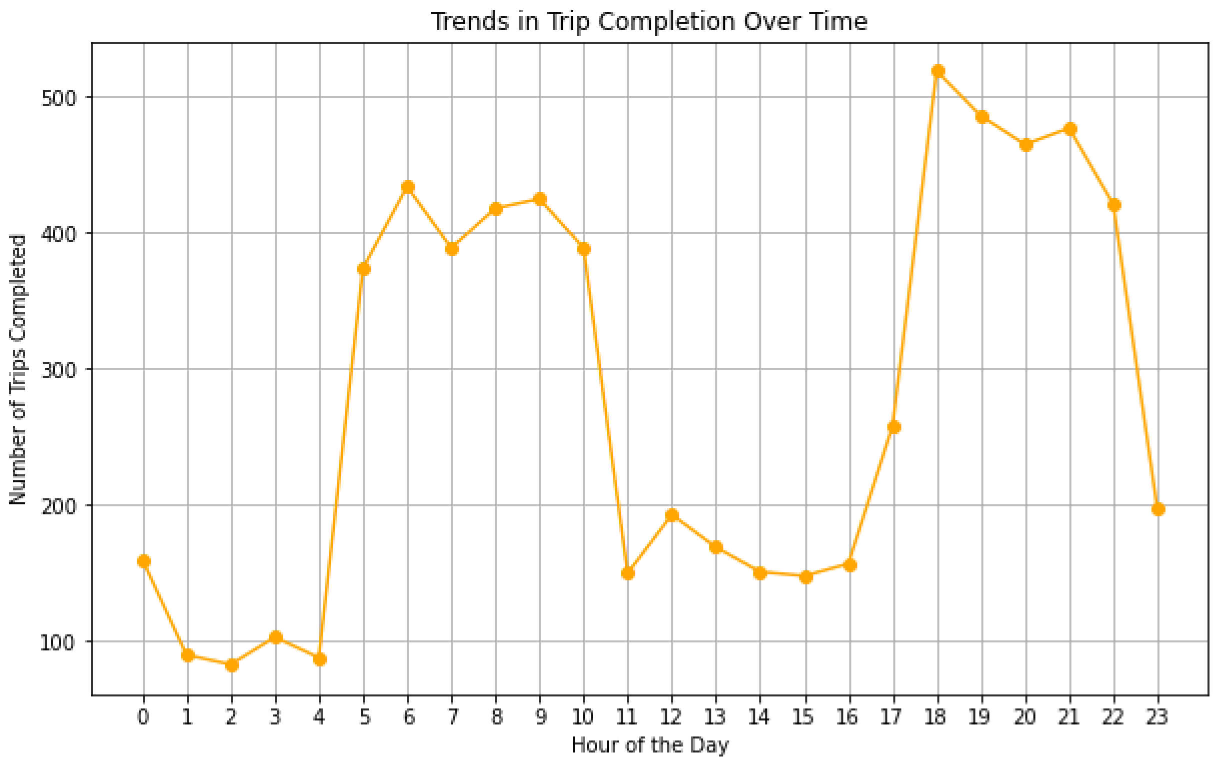
	Request id	Pickup point	Status	Trip Status	timestamp	period
0	867	Airport	Trip Completed	Trip Completed	18:00:00	afternoon
1	1807	City	Trip Completed	Trip Completed	09:00:00	morning
2	2532	Airport	Trip Completed	Trip Completed	21:00:00	evening
3	3112	City	Trip Completed	Trip Completed	09:00:00	morning
4	3879	Airport	Trip Completed	Trip Completed	22:00:00	evening

In [95]:
```
df.dtypes
```

Out[95]:
```
Request id        int64
Pickup point     object
Status           object
Trip Status      object
timestamp        object
period           object
dtype: object
```

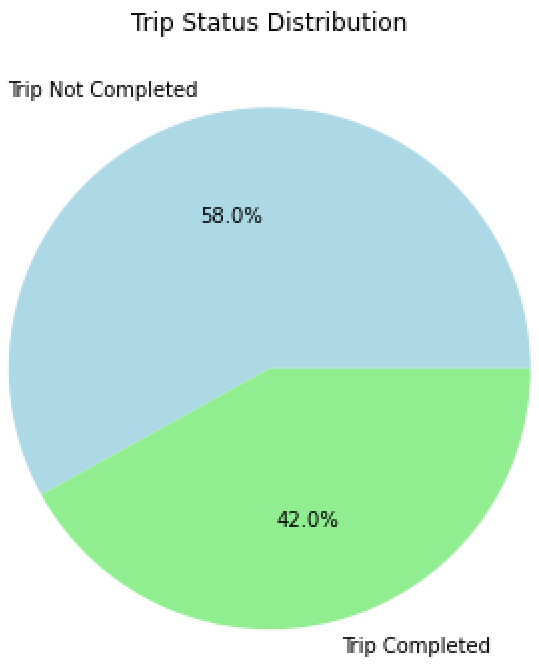
In [96]:
```
df['timestamp'] = pd.to_datetime(df['timestamp'], format='%H:%M:%S')
```

In [98]:
```
df['hour'] = df['timestamp'].dt.hour
trip_completion_trend = df.groupby('hour').size()
plt.figure(figsize=(10, 6))
trip_completion_trend.plot(kind='line', marker='o', color='orange')
plt.title('Trends in Trip Completion Over Time')
plt.xlabel('Hour of the Day')
plt.ylabel('Number of Trips Completed')
plt.xticks(range(0, 24))
plt.grid(True)
plt.show()
```



In [99]:
```
# Trip Status Distribution
trip_status_counts = df['Trip Status'].value_counts()
```

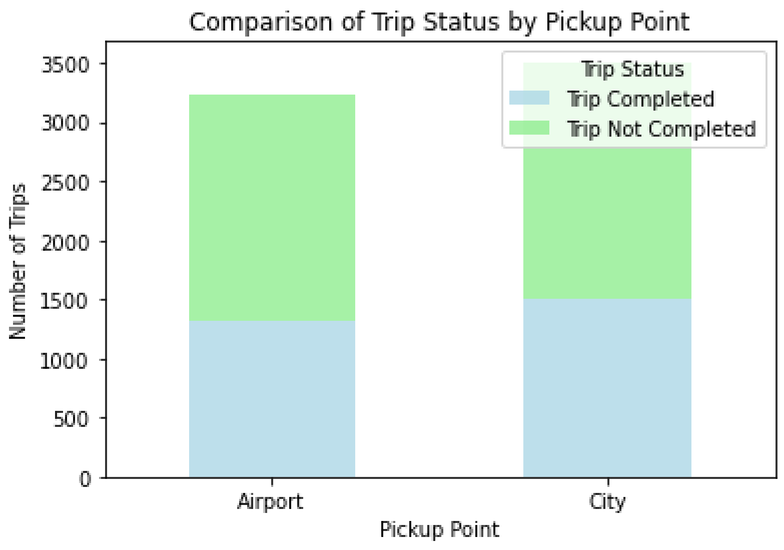
```
plt.figure(figsize=(8, 6))
trip_status_counts.plot(kind='pie', autopct='%1.1f%%', colors=['lightblue', 'lightgreen'])
plt.title('Trip Status Distribution')
plt.ylabel('')
plt.show()
```



In [100…

```
# Comparison of Trip Status by Pickup Point
trip_status_by_pickup = df.groupby(['Pickup point', 'Trip Status']).size().unstack()
plt.figure(figsize=(10, 6))
trip_status_by_pickup.plot(kind='bar', stacked=True, color=['lightblue', 'lightgreen', 'li
plt.title('Comparison of Trip Status by Pickup Point')
plt.xlabel('Pickup Point')
plt.ylabel('Number of Trips')
plt.xticks(rotation=0)
plt.legend(title='Trip Status')
plt.show()
```

```
<Figure size 720x432 with 0 Axes>
```



Supply demand Gap

In [105…

```
pivot_table = df.pivot_table(index=['Pickup point', 'period'], columns='Status', values='P

pivot_table.fillna(0, inplace=True)

print(pivot_table)
```

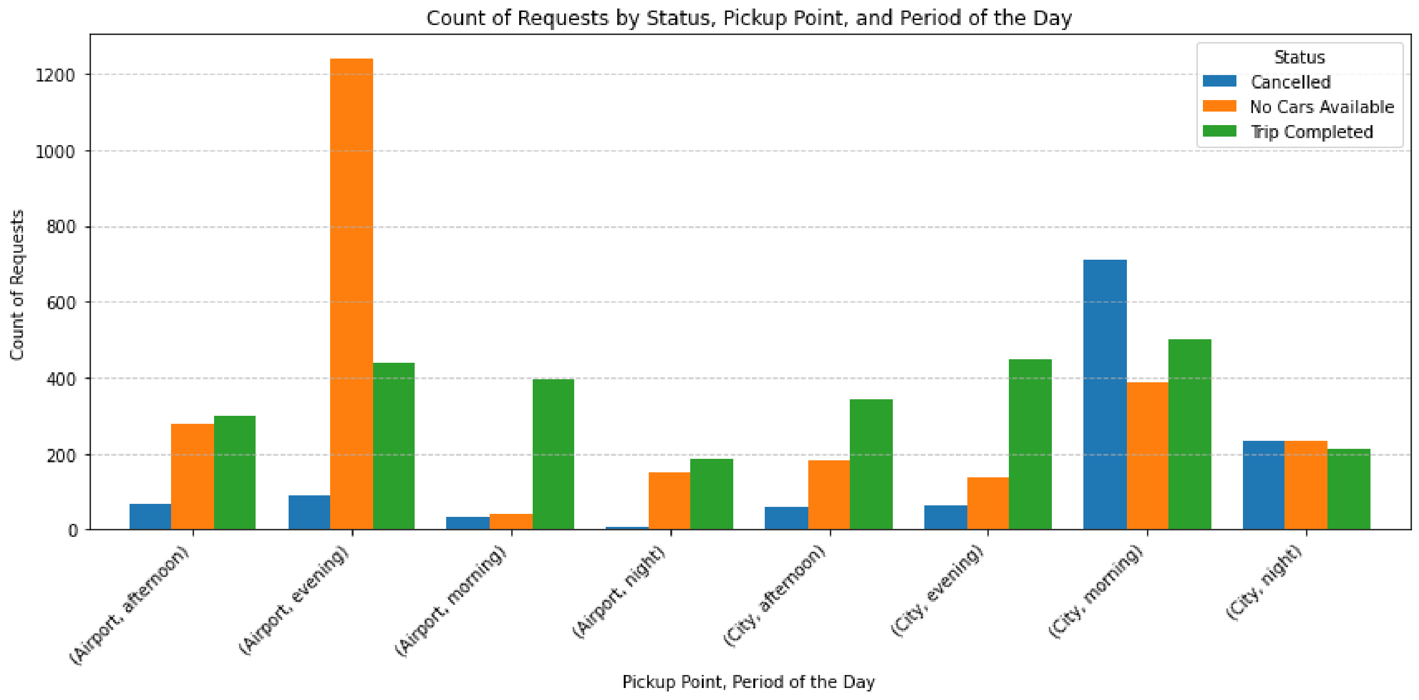
```
Status                    Cancelled  No Cars Available  Trip Completed
Pickup point period
Airport      afternoon           69                279             299
             evening             90               1242             441
             morning             33                 41             398
             night                6                151             188
City         afternoon           57                181             343
             evening             63                137             449
             morning            711                387             501
             night              235                232             211
```

In [106…

```
# Plotting the pivot table on a bar graph
pivot_table.plot(kind='bar', figsize=(12, 6), width=0.8)

# Setting plot attributes
plt.title('Count of Requests by Status, Pickup Point, and Period of the Day')
plt.xlabel('Pickup Point, Period of the Day')
plt.ylabel('Count of Requests')
plt.xticks(rotation=45, ha='right')
plt.legend(title='Status')
plt.grid(axis='y', linestyle='--', alpha=0.7)

# Showing the plot
plt.tight_layout()
plt.show()
```



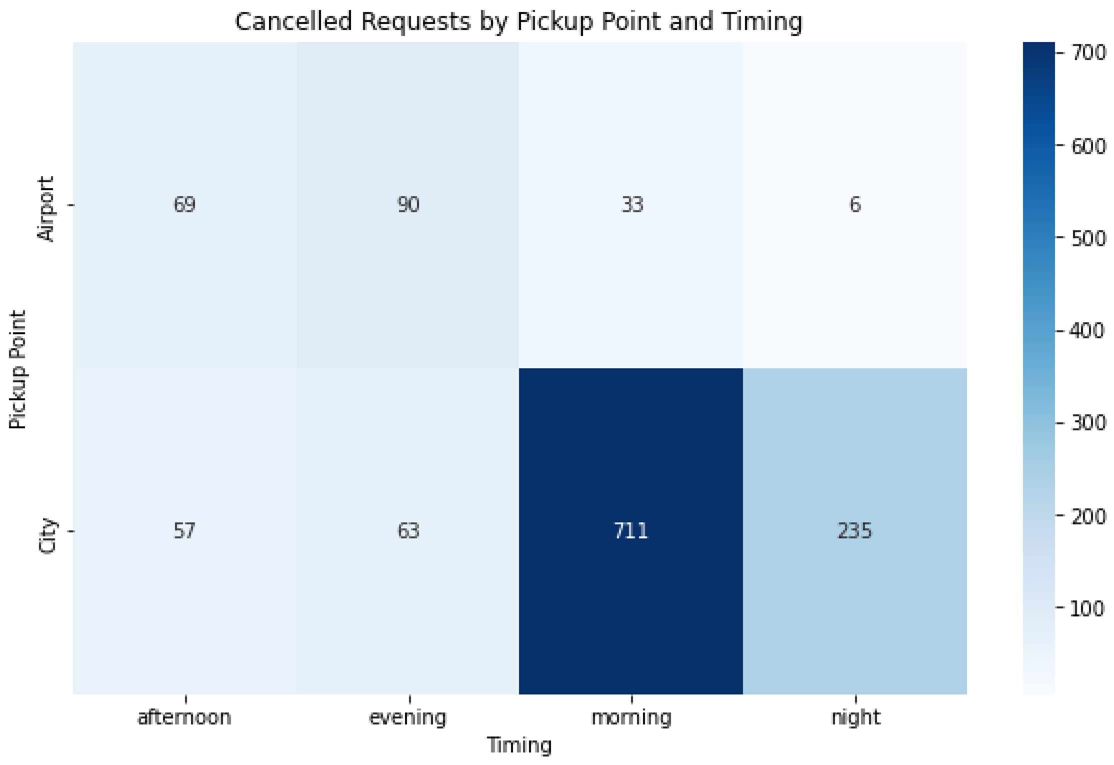
In [108…

```
# Filter data for cancelled requests
cancelled_requests = df[df['Status'] == 'Cancelled']

# Group by pickup point and timing, and count cancelled requests
cancelled_requests_count = cancelled_requests.groupby(['Pickup point', 'period']).size().u

# Visualize the relationship using a heatmap
plt.figure(figsize=(10, 6))
plt.title('Cancelled Requests by Pickup Point and Timing')
```

```
sns.heatmap(cancelled_requests_count, annot=True, cmap='Blues', fmt='g')
plt.xlabel('Timing')
plt.ylabel('Pickup Point')
plt.show()
```



In this heatmap, each cell represents the count of cancelled requests for a specific combination of pickup point and timing. Higher values indicate more cancelled requests. By examining this visualization, you can observe more cancellation requests are from city to airport in the monrning and night times. , indicating a potential relationship between pickup points, timings, and cancelled requests.

```
In [ ]:
```