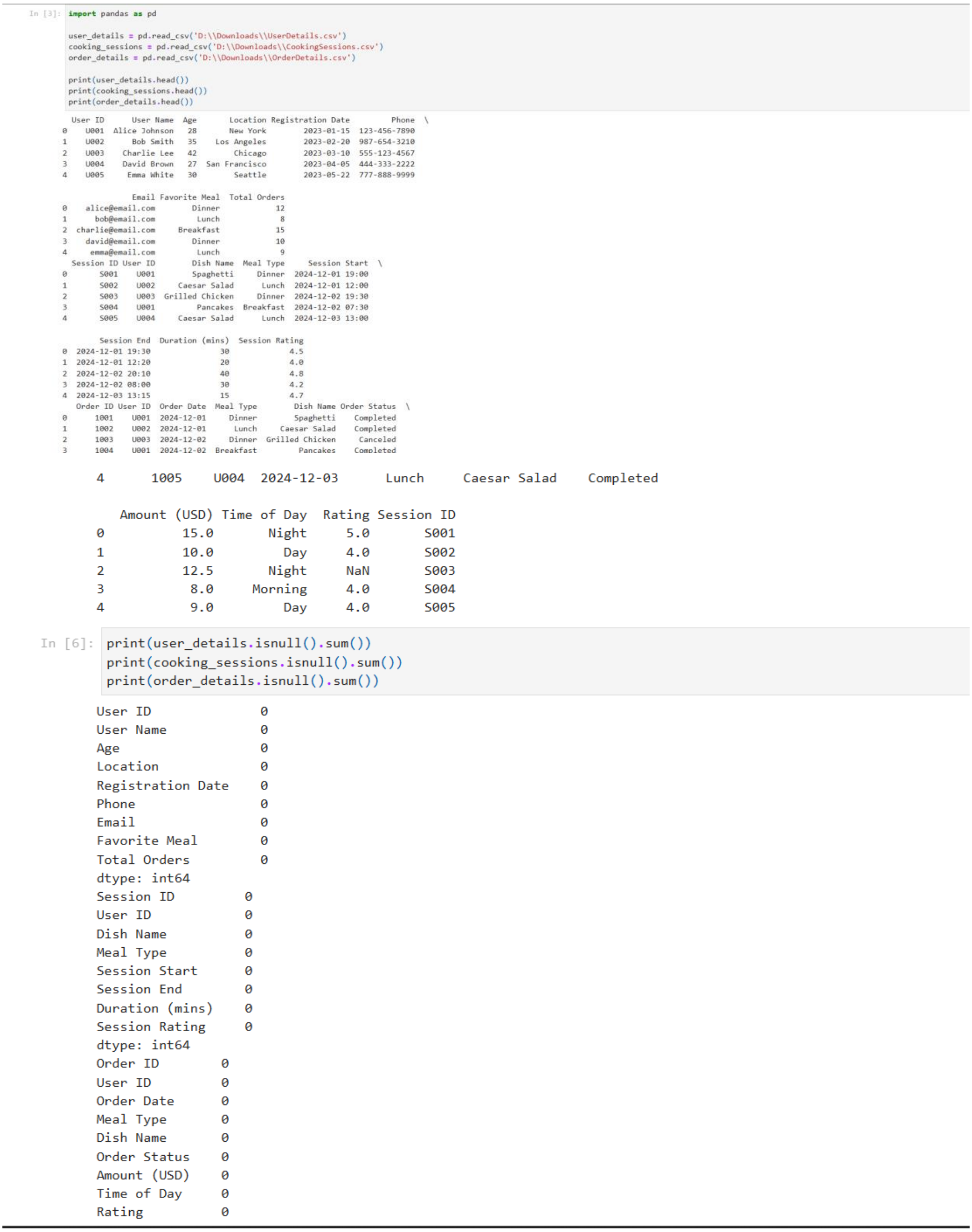


```
In [3]: import pandas as pd

        user_details = pd.read_csv('D:\\Downloads\\UserDetails.csv')
        cooking_sessions = pd.read_csv('D:\\Downloads\\CookingSessions.csv')
        order_details = pd.read_csv('D:\\Downloads\\OrderDetails.csv')

        print(user_details.head())
        print(cooking_sessions.head())
        print(order_details.head())
```

```
  User ID      User Name  Age       Location Registration Date         Phone  \
0    U001  Alice Johnson   28       New York        2023-01-15  123-456-7890
1    U002      Bob Smith   35    Los Angeles        2023-02-20  987-654-3210
2    U003    Charlie Lee   42        Chicago        2023-03-10  555-123-4567
3    U004    David Brown   27  San Francisco        2023-04-05  444-333-2222
4    U005     Emma White   30        Seattle        2023-05-22  777-888-9999

               Email Favorite Meal  Total Orders
0    alice@email.com        Dinner            12
1      bob@email.com         Lunch             8
2  charlie@email.com     Breakfast            15
3    david@email.com        Dinner            10
4     emma@email.com         Lunch             9
  Session ID User ID        Dish Name  Meal Type     Session Start  \
0       S001    U001        Spaghetti     Dinner  2024-12-01 19:00
1       S002    U002     Caesar Salad      Lunch  2024-12-01 12:00
2       S003    U003  Grilled Chicken     Dinner  2024-12-02 19:30
3       S004    U001         Pancakes  Breakfast  2024-12-02 07:30
4       S005    U004     Caesar Salad      Lunch  2024-12-03 13:00

        Session End  Duration (mins)  Session Rating
0  2024-12-01 19:30               30             4.5
1  2024-12-01 12:20               20             4.0
2  2024-12-02 20:10               40             4.8
3  2024-12-02 08:00               30             4.2
4  2024-12-03 13:15               15             4.7
   Order ID User ID  Order Date  Meal Type        Dish Name Order Status  \
0      1001    U001  2024-12-01     Dinner        Spaghetti    Completed
1      1002    U002  2024-12-01      Lunch     Caesar Salad    Completed
2      1003    U003  2024-12-02     Dinner  Grilled Chicken     Canceled
3      1004    U001  2024-12-02  Breakfast         Pancakes    Completed
4      1005    U004  2024-12-03      Lunch     Caesar Salad    Completed

   Amount (USD) Time of Day  Rating Session ID
0          15.0       Night     5.0       S001
1          10.0         Day     4.0       S002
2          12.5       Night     NaN       S003
3           8.0     Morning     4.0       S004
4           9.0         Day     4.0       S005
```

```
In [6]: print(user_details.isnull().sum())
        print(cooking_sessions.isnull().sum())
        print(order_details.isnull().sum())
```

```
User ID              0
User Name            0
Age                  0
Location             0
Registration Date    0
Phone                0
Email                0
Favorite Meal        0
Total Orders         0
dtype: int64
Session ID         0
User ID            0
Dish Name          0
Meal Type          0
Session Start      0
Session End        0
Duration (mins)    0
Session Rating     0
dtype: int64
Order ID        0
User ID         0
Order Date      0
Meal Type       0
Dish Name       0
Order Status    0
Amount (USD)    0
Time of Day     0
Rating          0
Rating          0
Session ID      0
dtype: int64
```

In [7]:
```
user_details.drop_duplicates(inplace=True)
cooking_sessions.drop_duplicates(inplace=True)
order_details.drop_duplicates(inplace=True)
```

In [8]:
```
# Convert date columns to datetime format
user_details['Registration Date'] = pd.to_datetime(user_details['Registration Date'])
cooking_sessions['Session Start'] = pd.to_datetime(cooking_sessions['Session Start'])
cooking_sessions['Session End'] = pd.to_datetime(cooking_sessions['Session End'])
order_details['Order Date'] = pd.to_datetime(order_details['Order Date'])
```

In [9]:
```
# Merge UserDetails with CookingSessions based on User ID
user_sessions = pd.merge(user_details, cooking_sessions, on='User ID', how='left')
print(user_sessions.head())
```

```
  User ID      User Name  Age     Location Registration Date         Phone  \
0    U001  Alice Johnson   28     New York        2023-01-15  123-456-7890
1    U001  Alice Johnson   28     New York        2023-01-15  123-456-7890
2    U001  Alice Johnson   28     New York        2023-01-15  123-456-7890
3    U002      Bob Smith   35  Los Angeles        2023-02-20  987-654-3210
4    U002      Bob Smith   35  Los Angeles        2023-02-20  987-654-3210

             Email Favorite Meal  Total Orders Session ID        Dish Name  \
0  alice@email.com        Dinner            12       S001        Spaghetti
1  alice@email.com        Dinner            12       S004         Pancakes
2  alice@email.com        Dinner            12       S009  Grilled Chicken
3    bob@email.com         Lunch             8       S002     Caesar Salad
4    bob@email.com         Lunch             8       S006        Spaghetti

   Meal Type       Session Start         Session End  Duration (mins)  \
0     Dinner 2024-12-01 19:00:00 2024-12-01 19:30:00             30.0
1  Breakfast 2024-12-02 07:30:00 2024-12-02 08:00:00             30.0
2     Dinner 2024-12-05 19:00:00 2024-12-05 19:40:00             40.0
3      Lunch 2024-12-01 12:00:00 2024-12-01 12:20:00             20.0
4     Dinner 2024-12-03 18:30:00 2024-12-03 19:00:00             30.0

   Session Rating
0             4.5
1             4.2
2             4.9
3             4.0
4             4.3
```

In [10]:
```
# Merge the resulting data with OrderDetails based on User ID
final_data = pd.merge(user_sessions, order_details, on='User ID', how='left')
print(final_data.head())
```

```
  User ID      User Name  Age  Location Registration Date         Phone  \
0    U001  Alice Johnson   28  New York        2023-01-15  123-456-7890
1    U001  Alice Johnson   28  New York        2023-01-15  123-456-7890
2    U001  Alice Johnson   28  New York        2023-01-15  123-456-7890
3    U001  Alice Johnson   28  New York        2023-01-15  123-456-7890
4    U001  Alice Johnson   28  New York        2023-01-15  123-456-7890

             Email Favorite Meal  Total Orders Session ID_x  ...  \
0  alice@email.com        Dinner            12         S001  ...
1  alice@email.com        Dinner            12         S001  ...
2  alice@email.com        Dinner            12         S001  ...
3  alice@email.com        Dinner            12         S004  ...
4  alice@email.com        Dinner            12         S004  ...

  Session Rating Order ID Order Date Meal Type_y      Dish Name_y  \
0            4.5   1001.0 2024-12-01      Dinner        Spaghetti
1            4.5   1004.0 2024-12-02   Breakfast         Pancakes
2            4.5   1009.0 2024-12-05      Dinner  Grilled Chicken
3            4.2   1001.0 2024-12-01      Dinner        Spaghetti
4            4.2   1004.0 2024-12-02   Breakfast         Pancakes

   Order Status  Amount (USD) Time of Day Rating Session ID_y
0     Completed          15.0       Night    5.0         S001
1     Completed           8.0     Morning    4.0         S004
2     Completed          12.0       Night    5.0         S009
3     Completed          15.0       Night    5.0         S001
4     Completed           8.0     Morning    4.0         S004

[5 rows x 25 columns]
```

In [12]:
```
print(final_data.columns)
```

```
Index(['User ID', 'User Name', 'Age', 'Location', 'Registration Date', 'Phone',
       'Email', 'Favorite Meal', 'Total Orders', 'Session ID_x', 'Dish Name_x',
       'Meal Type_x', 'Session Start', 'Session End', 'Duration (mins)',
       'Session Rating', 'Order ID', 'Order Date', 'Meal Type_y',
       'Dish Name_y', 'Order Status', 'Amount (USD)', 'Time of Day', 'Rating',
       'Session ID_y'],
      dtype='object')
```

In [13]:
```
final_data.rename(columns={'Session ID_x': 'Session ID'}, inplace=True)
final_data.drop(columns=['Session ID_y'], inplace=True)
session_orders = final_data.groupby('Session ID')['Order ID'].count().reset_index()
session_orders.rename(columns={'Order ID': 'Number of Orders'}, inplace=True)
```

In [14]:
```
# Check how many orders were made after each cooking session
session_orders = final_data.groupby('Session ID')['Order ID'].count().reset_index()
session_orders.rename(columns={'Order ID': 'Number of Orders'}, inplace=True)

# Display sessions with the most orders
print(session_orders.sort_values(by='Number of Orders', ascending=False))
```

```
   Session ID  Number of Orders
0        S001                 3
1        S002                 3
2        S003                 3
3        S004                 3
5        S006                 3
7        S008                 3
8        S009                 3
9        S010                 3
10       S011                 3
4        S005                 2
6        S007                 2
11       S012                 2
12       S013                 2
13       S014                 1
14       S015                 1
15       S016                 1
```

In [16]:
```
# Grouping by the correct column, assuming 'Dish Name_y' is the correct one for orders
popular_dishes = final_data.groupby('Dish Name_y')['Order ID'].count().reset_index()
popular_dishes.rename(columns={'Order ID': 'Number of Orders'}, inplace=True)
popular_dishes = popular_dishes.sort_values(by='Number of Orders', ascending=False)

# Display top 10 most ordered dishes
print(popular_dishes.head(10))
```

```
       Dish Name_y  Number of Orders
1  Grilled Chicken                 9
4        Spaghetti                 9
0     Caesar Salad                 7
3         Pancakes                 6
5    Veggie Burger                 4
2          Oatmeal                 3
```

In [18]:
```
# Grouping by 'Dish Name_x' if it's from the cooking session
session_dishes = final_data.groupby('Dish Name_x')['Session ID'].count().reset_index()
session_dishes.rename(columns={'Session ID': 'Number of Sessions'}, inplace=True)
session_dishes = session_dishes.sort_values(by='Number of Sessions', ascending=False)

# Display top 10 dishes cooked
print(session_dishes.head(10))
```

```
       Dish Name_x  Number of Sessions
1  Grilled Chicken                   9
4        Spaghetti                   9
0     Caesar Salad                   7
3         Pancakes                   6
5    Veggie Burger                   4
2          Oatmeal                   3
```

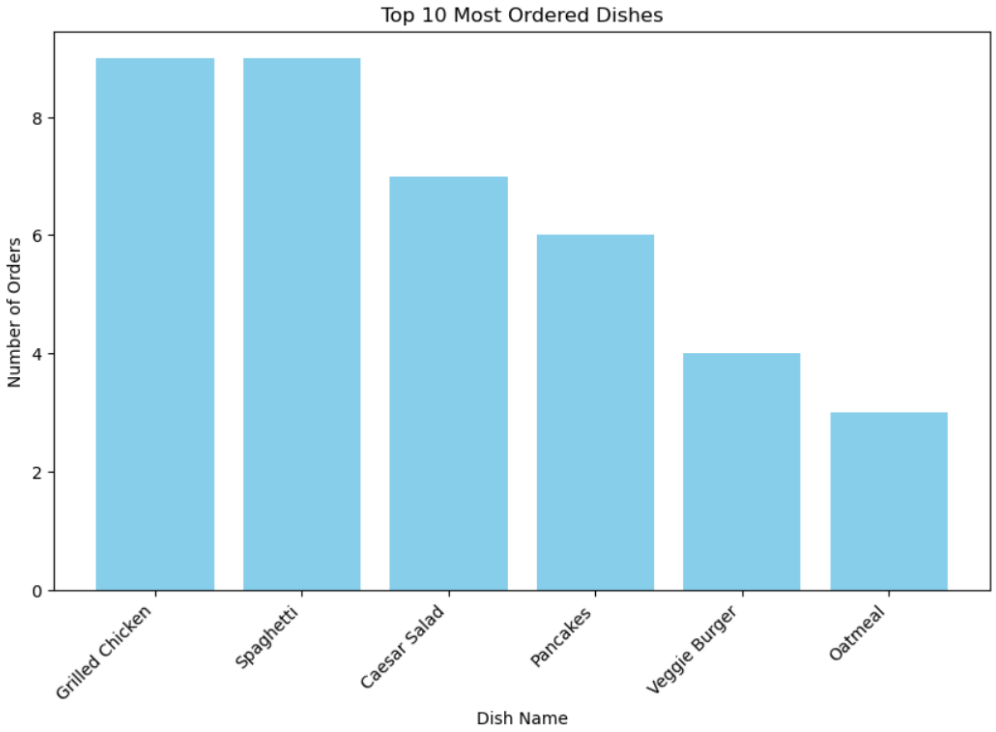
In [20]:

```
# Group by Age and count the number of orders
age_orders = final_data.groupby('Age')['Order ID'].count().reset_index()
age_orders.rename(columns={'Order ID': 'Number of Orders'}, inplace=True)

# Group by Location and count the number of orders
location_orders = final_data.groupby('Location')['Order ID'].count().reset_index()
location_orders.rename(columns={'Order ID': 'Number of Orders'}, inplace=True)

# Display Age and Location-based orders
print(age_orders)
print(location_orders)
```

```
   Age  Number of Orders
0   25                 1
1   27                 4
2   28                 9
3   29                 0
4   30                 4
5   31                 1
6   33                 0
7   35                 9
8   38                 1
9   42                 9
        Location  Number of Orders
0         Austin                 1
1         Boston                 1
2        Chicago                 9
3         Dallas                 0
4    Los Angeles                 9
5          Miami                 1
6       New York                 9
7        Phoenix                 0
8  San Francisco                 4
9        Seattle                 4
```

In [22]:

```
import matplotlib.pyplot as plt

# Bar plot for popular dishes
plt.figure(figsize=(10, 6))
plt.bar(popular_dishes['Dish Name_y'].head(10), popular_dishes['Number of Orders'].head(10), color='skyblue')
plt.xticks(rotation=45, ha='right')
plt.xlabel('Dish Name')
plt.ylabel('Number of Orders')
plt.title('Top 10 Most Ordered Dishes')
plt.show()
```



In [24]:

```
print(final_data.columns)
```

```
Index(['User ID', 'User Name', 'Age', 'Location', 'Registration Date', 'Phone',
       'Email', 'Favorite Meal', 'Total Orders', 'Session ID', 'Dish Name_x',
       'Meal Type_x', 'Session Start', 'Session End', 'Duration (mins)',
       'Session Rating', 'Order ID', 'Order Date', 'Meal Type_y',
       'Dish Name_y', 'Order Status', 'Amount (USD)', 'Time of Day', 'Rating'],
      dtype='object')
```

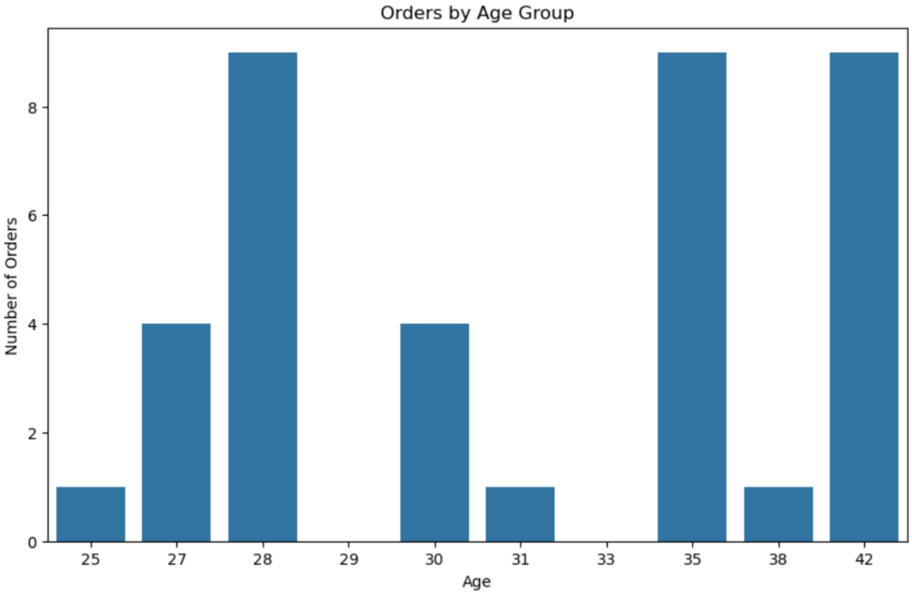
In [29]:

```
print(final_data.columns.duplicated())
```

```
[False False False False False False False False False False False False
 False False False False False False False  True False False False False]
```

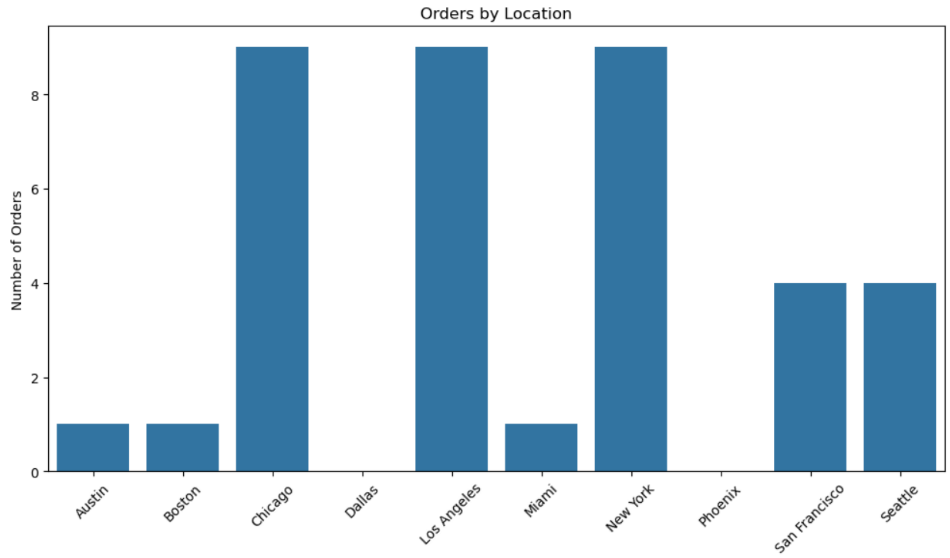
In [34]:

```
# Plot number of orders by Age
plt.figure(figsize=(10, 6))
sns.barplot(data=age_orders, x='Age', y='Number of Orders')
plt.title('Orders by Age Group')
plt.show()
```



In [35]:

```
# Plot number of orders by Location
plt.figure(figsize=(12, 6))
sns.barplot(data=location_orders, x='Location', y='Number of Orders')
plt.title('Orders by Location')
plt.xticks(rotation=45)
plt.show()
```



In [36]:

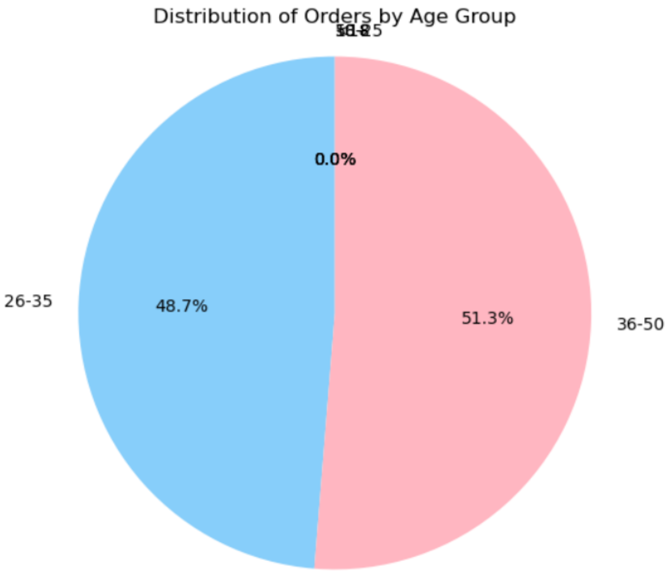
```
# Define age bins and labels
bins = [0, 18, 25, 35, 50, 100]  # Adjust the ranges as needed
labels = ['<18', '18-25', '26-35', '36-50', '50+']

# Create an Age Group column
final_data['Age Group'] = pd.cut(final_data['Age'], bins=bins, labels=labels, right=False)

# Group by Age Group and calculate the total number of orders
orders_by_age_group = final_data.groupby('Age Group').agg({'Total Orders': 'sum'}).reset_index()
orders_by_age_group.rename(columns={'Total Orders': 'Number of Orders'}, inplace=True)

# Display the resulting DataFrame
print(orders_by_age_group)
```

```
  Age Group  Number of Orders
0       <18                 0
1     18-25                 0
2     26-35               210
3     36-50               221
4       50+                 0
```

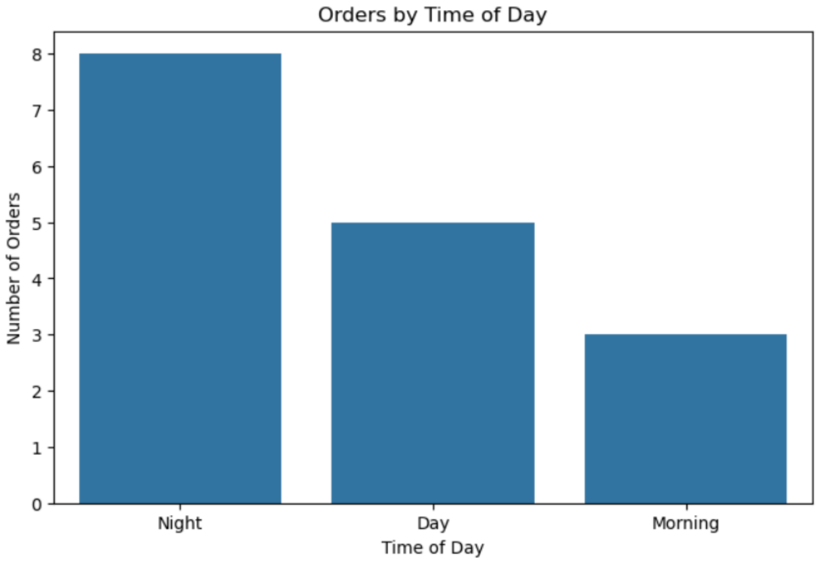
```
C:\Users\pooja\AppData\Local\Temp\ipykernel_18252\3562983514.py:9: FutureWarning: The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observ
ed=False to retain current behavior or observed=True to adopt the future default and silence this warning.
  orders_by_age_group = final_data.groupby('Age Group').agg({'Total Orders': 'sum'}).reset_index()
```

In [38]:

```
# Pie chart for orders by age group
plt.figure(figsize=(6, 6))
plt.pie(
    orders_by_age_group['Number of Orders'],
    labels=orders_by_age_group['Age Group'],
    autopct='%1.1f%%',
    startangle=90,
    colors=['lightgreen', 'lightcoral', 'lightskyblue', 'lightpink', 'lightyellow']
)
plt.title('Distribution of Orders by Age Group')
plt.axis('equal')  # Equal aspect ratio ensures that pie chart is circular
plt.show()
```



In [39]:
```
orders_time_of_day = order_details['Time of Day'].value_counts()
plt.figure(figsize=(8, 5))
sns.barplot(x=orders_time_of_day.index, y=orders_time_of_day.values)
plt.title('Orders by Time of Day')
plt.xlabel('Time of Day')
plt.ylabel('Number of Orders')
plt.show()
```


Orders by Time of Day
Number of Orders
Night
Day
Morning
Time of Day