Data Exploration

In [59]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import datetime as datetime
```

In [61]:
```
#Load data
sales_transactions = pd.read_csv('sales_transactions.csv')
menu_data = pd.read_csv('menu_data.csv')
inventory_ingredients = pd.read_csv('inventory_ingredients.csv')
customer_reviews = pd.read_csv('customer_reviews.csv')
```

In [63]:
```
#Displaying first few table rows
print("Sales Transactions:")
print(sales_transactions.head(), "\n")
```

```
Sales Transactions:
   Transaction_ID        Date      Time  Item_ID   Item_Name  Quantity  Price  \
0               1  2023-01-01  18:28:53        1  Margherita         1     32
1               2  2023-01-01  17:34:20        9    Tiramisu         1     18
2               3  2023-01-01  19:48:57        8   Coca-Cola         2     20
3               4  2023-01-01  16:43:30        2     Diavola         3    105
4               5  2023-01-01  17:23:35        7    Focaccia         1     15

   Payment_Method
0  Mobile Payment
1     Credit Card
2            Cash
3  Mobile Payment
4     Credit Card
```

In [98]:
```
sales_transactions.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45585 entries, 0 to 45584
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Transaction_ID        45585 non-null  int64
 1   Date                  45585 non-null  datetime64[ns]
 2   Time                  45585 non-null  object
 3   Item_ID               45585 non-null  int64
 4   Item_Name             45585 non-null  object
 5   Quantity              45585 non-null  int64
 6   Price                 45585 non-null  int64
 7   Payment_Method        45585 non-null  object
 8   Transaction_DateTime  45585 non-null  datetime64[ns]
 9   Hour                  45585 non-null  int32
 10  Quarter               45585 non-null  period[Q-DEC]
 11  Revenue               45585 non-null  int64
dtypes: datetime64[ns](2), int32(1), int64(5), object(3), period[Q-DEC](1)
memory usage: 4.0+ MB
```

In [116…
```
sales_transactions.describe()
```

Out[116…

	Transaction_ID	Date	Item_ID	Quantity	Price	Tra
count	45585.000000	45585	45585.000000	45585.000000	45585.000000	
mean	22793.000000	2023-06-29 14:55:05.034550784	5.996337	2.001316	50.516968	
min	1.000000	2023-01-01 00:00:00	1.000000	1.000000	5.000000	
25%	11397.000000	2023-03-29 00:00:00	3.000000	1.000000	24.000000	
50%	22793.000000	2023-07-02 00:00:00	6.000000	2.000000	37.000000	
75%	34189.000000	2023-09-28 00:00:00	9.000000	3.000000	74.000000	
max	45585.000000	2023-12-31 00:00:00	11.000000	3.000000	117.000000	
std	13159.400347	NaN	3.156099	0.818865	33.957259	

In [65]:
```
print("Menu Data:")
print(menu_data.head(), "\n")
```

```
Menu Data:
   Item_ID    Item_Name Category  Price  Available
0        1   Margherita    Pizza     32       True
1        2      Diavola    Pizza     35       True
2        3       Vegana    Pizza     36       True
3        4  Amatriciana    Pasta     37       True
4        5    Carbonara    Pasta     38       True
```

In [118…
```
menu_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11 entries, 0 to 10
Data columns (total 5 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   Item_ID    11 non-null     int64 
 1   Item_Name  11 non-null     object
 2   Category   11 non-null     object
 3   Price      11 non-null     int64 
 4   Available  11 non-null     bool  
dtypes: bool(1), int64(2), object(2)
memory usage: 495.0+ bytes
```

In [120…
```
menu_data.describe()
```

Out[120…

	Item_ID	Price
count	11.000000	11.000000
mean	6.000000	25.181818
std	3.316625	13.121114
min	1.000000	5.000000
25%	3.500000	13.500000
50%	6.000000	32.000000
75%	8.500000	36.500000
max	11.000000	39.000000

In [67]:
```
print("Inventory Ingredients:")
print(inventory_ingredients.head(), "\n")
```

```
Inventory Ingredients:
   Item_ID    Ingredient  Quantity_Per_Item  Stock_Level
0        1  Tomato Sauce                100         2000
1        1        Cheese                150         1500
2        1         Basil                 10          500
3        2  Tomato Sauce                100         2000
4        2        Cheese                150         1600
```

In [122…
```
inventory_ingredients.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27 entries, 0 to 26
Data columns (total 4 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   Item_ID            27 non-null     int64 
 1   Ingredient         27 non-null     object
 2   Quantity_Per_Item  27 non-null     int64 
 3   Stock_Level        27 non-null     int64 
dtypes: int64(3), object(1)
memory usage: 996.0+ bytes
```

In [126…
```
inventory_ingredients.describe()
```

Out[126...

	Item_ID	Quantity_Per_Item	Stock_Level
count	27.000000	27.000000	27.000000
mean	4.962963	71.814815	909.259259
std	2.848501	47.345731	549.481755
min	1.000000	1.000000	200.000000
25%	3.000000	30.000000	500.000000
50%	5.000000	80.000000	800.000000
75%	7.000000	100.000000	1350.000000
max	11.000000	150.000000	2000.000000

In [69]:
```
print("Customer Reviews:")
print(customer_reviews.head(), "\n")
```

```
Customer Reviews:
                      author_name  rating  \
0  Pedro Henrique Leitao de Souza       5
1     Marcella da Costa Rodrigues       5
2                   Daniel Kustra       5
3            Klaudia Nadziejewiec       5
4                 Monika Bielecka       5

                                                text                 time
0           Perfect Pizza! The true Italian style! 👌  2022-05-11 11:42:46
1                     The pizzas are incredible! ❤️  2023-09-10 18:46:32
2                                           Cudoo !  2019-11-17 14:00:35
3  A very atmospheric place with delicious pizza....  2024-08-05 17:25:49
4  What a shot at Italian flavors this Sunday! Ni...  2024-07-14 18:53:13
```

Analyze basic statistics and identify data trends

In [72]:
```
# Find top selling items
top_selling_items = sales_transactions.groupby('Item_ID')['Quantity'].sum().rese
top_selling_items = top_selling_items.sort_values(by='Quantity', ascending=False
print("Top Selling Menu Items:")
print(top_selling_items)
```

```
Top Selling Menu Items:
    Item_ID  Quantity
5         6      8550
2         3      8419
8         9      8397
4         5      8328
7         8      8268
0         1      8267
3         4      8249
9        10      8245
10       11      8239
6         7      8164
1         2      8104
```

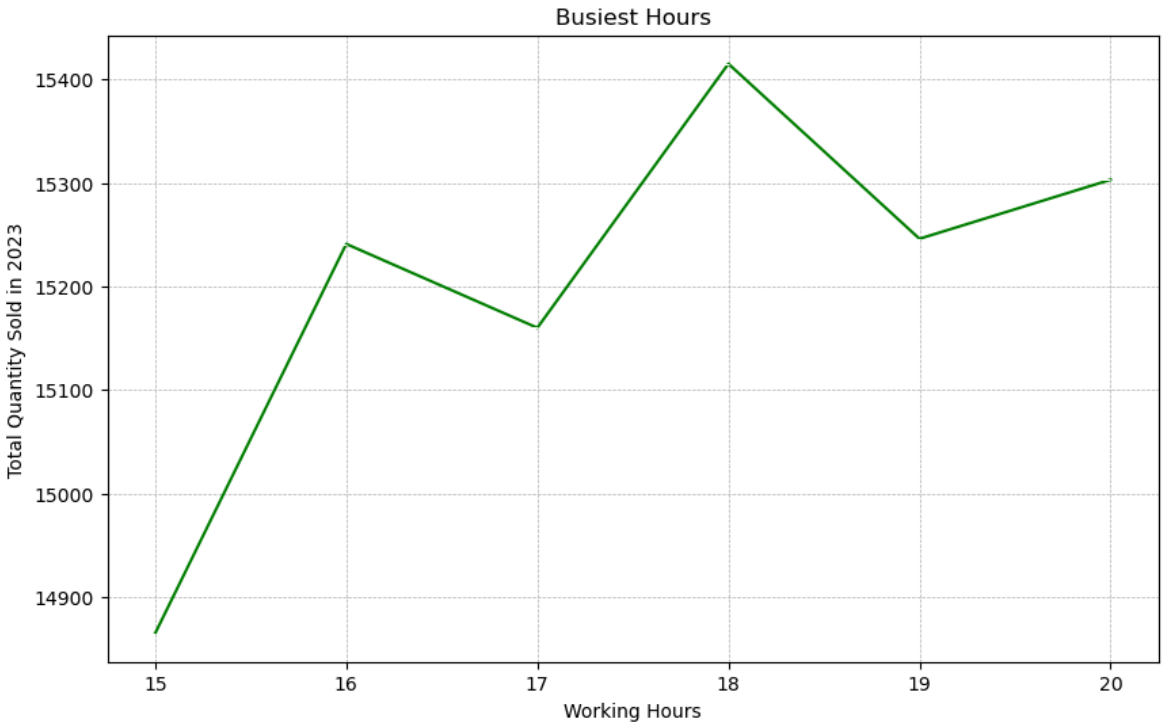
In [74]:
```
# Merge data to get item names
top_selling_items = top_selling_items.merge(menu_data[['Item_ID', 'Item_Name']],
print("Top Selling Menu Items with Names:")
print(top_selling_items)
```

```
Top Selling Menu Items with Names:
    Item_ID  Quantity    Item_Name
0         6      8550    Bolognese
1         3      8419       Vegana
2         9      8397     Tiramisu
3         5      8328    Carbonara
4         8      8268    Coca-Cola
5         1      8267   Margherita
6         4      8249  Amatriciana
7        10      8245     Espresso
8        11      8239        Water
9         7      8164     Focaccia
10        2      8104      Diavola
```

In [112…
```
# Find the busiest hours
busiest_hours = sales_transactions.groupby('Hour')['Quantity'].sum().reset_index
busiest_hours = busiest_hours.sort_values(by='Quantity', ascending=False)
print("Busiest Hours:")
print(busiest_hours)
```

```
Busiest Hours:
   Hour  Quantity
3    18     15415
5    20     15303
4    19     15246
1    16     15241
2    17     15160
0    15     14865
```

In [143…
```
#Create line plot to show busiest hours
plt.figure(figsize=(10, 6))
sns.lineplot(data=busiest_hours, x='Hour', y='Quantity', marker=False, color='g'
plt.title('Busiest Hours')
plt.xlabel('Working Hours')
plt.ylabel('Total Quantity Sold in 2023')
plt.grid(visible=True, which='both', linestyle='--', linewidth=0.5)
plt.xticks(range(15, 21))
plt.savefig('busiest_hours.png')
plt.show()
```



In [128…
```
# Calculate the total number of transactions for each payment method
payment_method_counts = sales_transactions['Payment_Method'].value_counts(normal
payment_method_counts.columns = ['Payment_Method', 'Percentage']

# Convert to percentage format
payment_method_counts['Percentage'] *= 100

# Display the percentage breakdown
print(payment_method_counts)
```

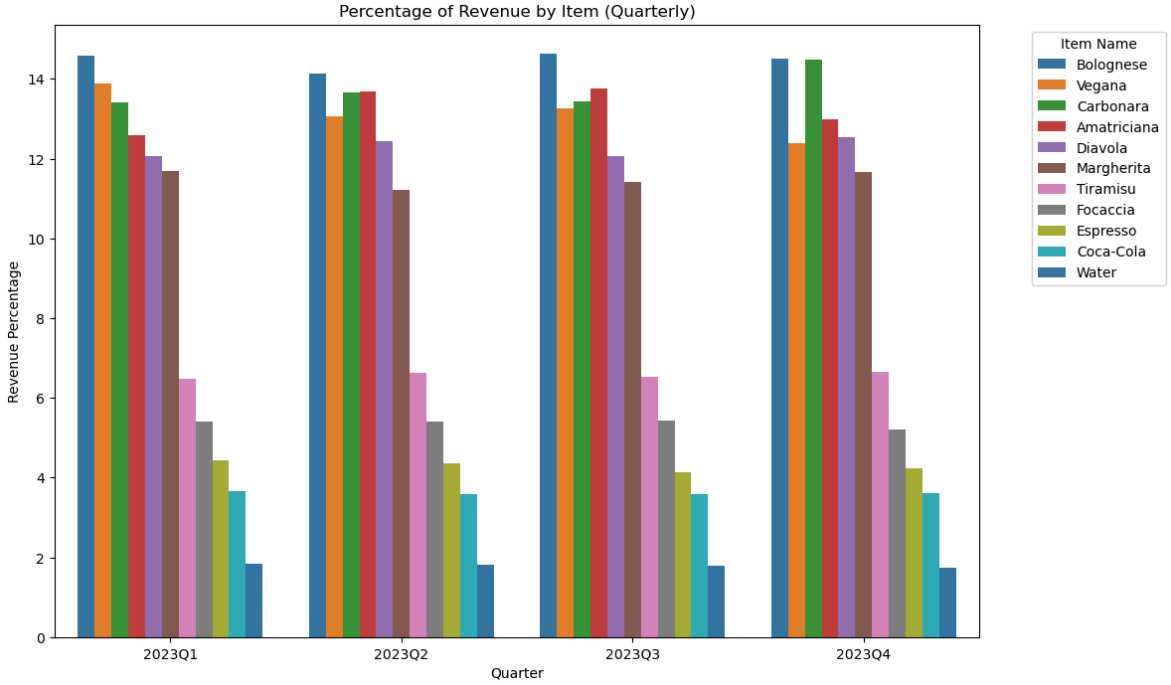
```
  Payment_Method  Percentage
0  Mobile Payment   33.581222
1     Credit Card   33.302621
2            Cash   33.116157
```

In [100…
```
#Calculate percentage revenue by quarter for each item

# Add a Quarter column
sales_transactions['Quarter'] = sales_transactions['Date'].dt.to_period('Q')

# Calculate revenue for each transaction
sales_transactions['Revenue'] = sales_transactions['Quantity'] * sales_transacti

# Group by Quarter and Item_Name, and sum revenue
quarterly_revenue = sales_transactions.groupby(['Quarter', 'Item_Name'])['Revenu

# Calculate total revenue per quarter
total_revenue_per_quarter = quarterly_revenue.groupby('Quarter')['Revenue'].tran

# Add a percentage column
quarterly_revenue['Percentage'] = (quarterly_revenue['Revenue'] / total_revenue_

# Sort values
quarterly_revenue = quarterly_revenue.sort_values(by=['Quarter', 'Percentage'],

print(quarterly_revenue)
```

	Quarter	Item_Name	Revenue	Percentage
1	2023Q1	Bolognese	201591	14.586523
9	2023Q1	Vegana	191988	13.891679
2	2023Q1	Carbonara	185250	13.404137
0	2023Q1	Amatriciana	173826	12.577531
4	2023Q1	Diavola	166635	12.057211
7	2023Q1	Margherita	161632	11.695209
8	2023Q1	Tiramisu	89388	6.467849
6	2023Q1	Focaccia	74640	5.400728
5	2023Q1	Espresso	61236	4.430854
3	2023Q1	Coca-Cola	50580	3.659818
10	2023Q1	Water	25270	1.828462
12	2023Q2	Bolognese	182091	14.132420
11	2023Q2	Amatriciana	176453	13.694844
13	2023Q2	Carbonara	176054	13.663877
20	2023Q2	Vegana	168408	13.070457
15	2023Q2	Diavola	160195	12.433031
18	2023Q2	Margherita	144352	11.203426
19	2023Q2	Tiramisu	85356	6.624637
17	2023Q2	Focaccia	69630	5.404113
16	2023Q2	Espresso	56124	4.355888
14	2023Q2	Coca-Cola	46310	3.594205
21	2023Q2	Water	23490	1.823102
23	2023Q3	Bolognese	203658	14.629742
22	2023Q3	Amatriciana	191549	13.759893
24	2023Q3	Carbonara	186998	13.432973
31	2023Q3	Vegana	184608	13.261288
26	2023Q3	Diavola	167790	12.053169
29	2023Q3	Margherita	158752	11.403926
30	2023Q3	Tiramisu	90774	6.520737
28	2023Q3	Focaccia	75465	5.421017
27	2023Q3	Espresso	57528	4.132515
25	2023Q3	Coca-Cola	49900	3.584559
32	2023Q3	Water	25060	1.800181
34	2023Q4	Bolognese	190320	14.509161
35	2023Q4	Carbonara	189886	14.476075
33	2023Q4	Amatriciana	170459	12.995045
37	2023Q4	Diavola	164360	12.530084
42	2023Q4	Vegana	162504	12.388591
40	2023Q4	Margherita	152896	11.656119
41	2023Q4	Tiramisu	87300	6.655369
39	2023Q4	Focaccia	68205	5.199650
38	2023Q4	Espresso	55668	4.243884
36	2023Q4	Coca-Cola	47310	3.606707
43	2023Q4	Water	22815	1.739315

In [145…

```
#Visualize percentage of revenue by item, quarterly
plt.figure(figsize=(12, 8))
sns.barplot(data=quarterly_revenue, x='Quarter', y='Percentage', hue='Item_Name'
plt.title('Percentage of Revenue by Item (Quarterly)')
plt.xlabel('Quarter')
plt.ylabel('Revenue Percentage')
plt.legend(title='Item Name', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.savefig('quarterly_revenue.png')
plt.show()
```


Percentage of Revenue by Item (Quarterly)
Revenue Percentage
Quarter
2023Q1
2023Q2
2023Q3
2023Q4
0
2
4
6
8
10
12
14
Item Name
Bolognese
Vegana
Carbonara
Amatriciana
Diavola
Margherita
Tiramisu
Focaccia
Espresso
Coca-Cola
Water

In [ ]: