Importing Necessary Libraries

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [5]:
```
df=pd.read_csv('C:/Users/Ajit Tiwari/Desktop/Jar Assignment/Walmart Sales.csv')
```

Cleaning the Data

In [6]:
```
df.head()
```

Out[6]:

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Date	Time	Payment	Rating	Revenue
0	750-67-8428	A	Yangon	Member	Female	Health and beauty	74.69	7	01/05/2019	13:08	Ewallet	9.1	522.83
1	226-31-3081	A	Naypyitaw	Normal	Female	Electronic accessories	15.28	5	03/08/2019	10:29	Cash	9.6	76.40
2	631-41-3108	A	Yangon	Normal	Male	Home and lifestyle	46.33	7	03/03/2019	13:23	Credit card	7.4	324.31
3	123-19-1176	B	Yangon	Member	Male	Health and beauty	58.22	8	1/27/2019	20:33	Ewallet	8.4	465.76
4	373-73-7910	C	Yangon	Normal	Male	Sports and travel	86.31	7	02/08/2019	10:37	Ewallet	5.3	604.17

In [7]:
```
df.tail()
```

Out[7]:

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Date	Time	Payment	Rating	Revenue
995	233-67-5758	A	Naypyitaw	Normal	Male	Health and beauty	40.35	1	1/29/2019	13:46	Ewallet	6.2	40.35
996	303-96-2227	A	Mandalay	Normal	Female	Home and lifestyle	97.38	10	03/02/2019	17:16	Ewallet	4.4	973.80
997	727-02-1313	A	Yangon	Member	Male	Food and beverages	31.84	1	02/09/2019	13:22	Cash	7.7	31.84
998	347-56-2442	B	Yangon	Normal	Male	Home and lifestyle	65.82	1	2/22/2019	15:33	Cash	4.1	65.82
999	849-09-3807	C	Yangon	Member	Female	Fashion accessories	88.34	7	2/18/2019	13:28	Cash	6.6	618.38

In [8]:
```
df.shape
```

Out[8]:
```
(1000, 13)
```

In [9]:
```
df.describe()
```

Out[9]:

	Unit price	Quantity	Rating	Revenue
count	1000.000000	1000.000000	1000.00000	1000.00000
mean	55.672130	5.510000	6.97270	307.58738
std	26.494628	2.923431	1.71858	234.17651
min	10.080000	1.000000	4.00000	10.17000
25%	32.875000	3.000000	5.50000	118.49750
50%	55.230000	5.000000	7.00000	241.76000
75%	77.935000	8.000000	8.50000	448.90500
max	99.960000	10.000000	10.00000	993.00000

In [10]:
```
df.isnull().sum()
```

Out[10]:

```
Invoice ID       0
Branch           0
City             0
Customer type    0
Gender           0
Product line     0
Unit price       0
Quantity         0
Date             0
Time             0
Payment          0
Rating           0
Revenue          0
dtype: int64
```

No Null Values

In [12]:

```
df.duplicated().sum()
```

Out[12]:

```
0
```

No Duplicate Values

In [14]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 13 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Invoice ID     1000 non-null   object 
 1   Branch         1000 non-null   object 
 2   City           1000 non-null   object 
 3   Customer type  1000 non-null   object 
 4   Gender         1000 non-null   object 
 5   Product line   1000 non-null   object 
 6   Unit price     1000 non-null   float64
 7   Quantity       1000 non-null   int64  
 8   Date           1000 non-null   object 
 9   Time           1000 non-null   object 
 10  Payment        1000 non-null   object 
 11  Rating         1000 non-null   float64
 12  Revenue        1000 non-null   float64
dtypes: float64(3), int64(1), object(9)
memory usage: 101.7+ KB
```

Exploratory Data Analysis

Q.1 Analyze the performance of sales and revenue at the city and branch level

In [15]:

```
sales=df.groupby(['City','Branch'])['Quantity'].sum().reset_index()
sales
```

Out[15]:

	City	Branch	Quantity
0	Mandalay	A	637
1	Mandalay	B	664
2	Mandalay	C	519
3	Naypyitaw	A	648
4	Naypyitaw	B	604
5	Naypyitaw	C	579
6	Yangon	A	598
7	Yangon	B	631
8	Yangon	C	630

In [16]:

```
revenue=df.groupby(['City','Branch'])['Revenue'].sum().reset_index()
revenue
```

Out[16]:

	City	Branch	Revenue
0	Mandalay	A	34130.09
1	Mandalay	B	37215.93
2	Mandalay	C	29794.62
3	Naypyitaw	A	35985.64
4	Naypyitaw	B	35157.75
5	Naypyitaw	C	34160.14
6	Yangon	A	33647.27
7	Yangon	B	35193.51
8	Yangon	C	32302.43

In [17]:

```
performance=pd.merge(sales, revenue, on=['City','Branch'])
performance
```

Out[17]:

	City	Branch	Quantity	Revenue
0	Mandalay	A	637	34130.09
1	Mandalay	B	664	37215.93
2	Mandalay	C	519	29794.62
3	Naypyitaw	A	648	35985.64
4	Naypyitaw	B	604	35157.75
5	Naypyitaw	C	579	34160.14
6	Yangon	A	598	33647.27
7	Yangon	B	631	35193.51
8	Yangon	C	630	32302.43

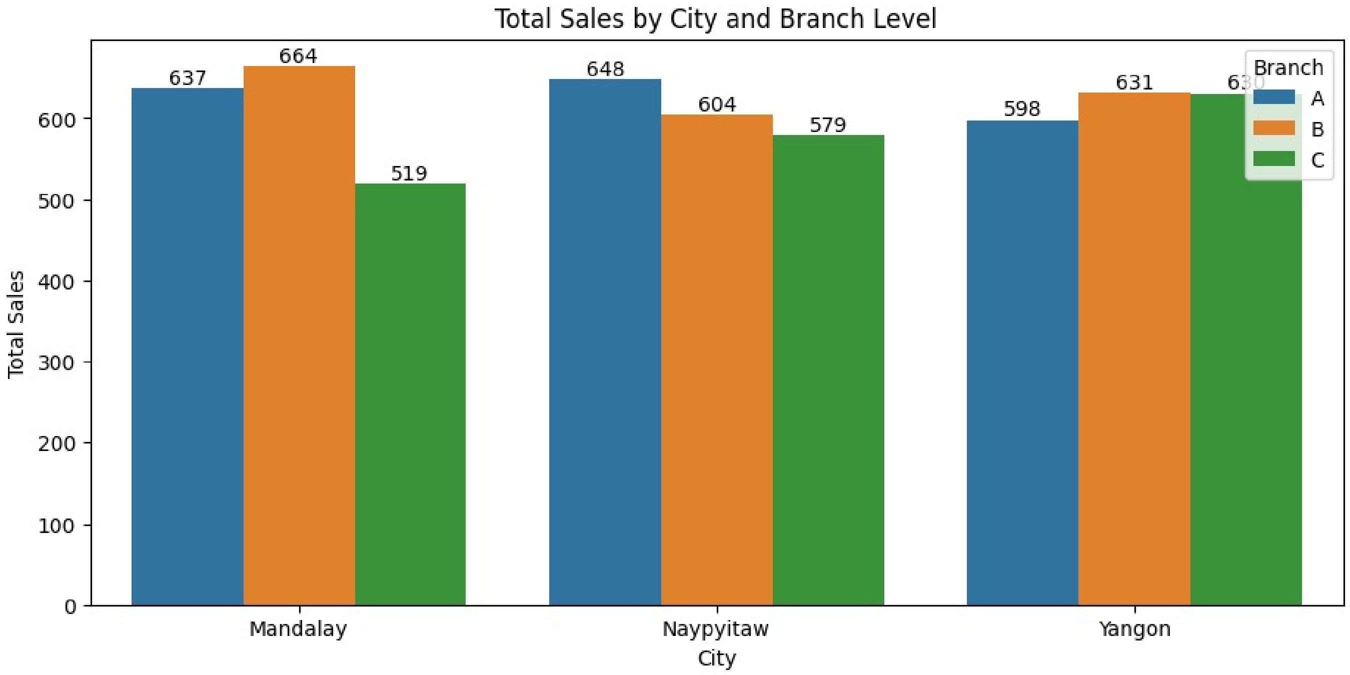
Visualization

In [25]:

```
plt.figure(figsize=(11,5))
ax=sns.barplot(x='City', y='Quantity',hue='Branch', data=performance)
ax.bar_label(ax.containers[0])
ax.bar_label(ax.containers[1])
ax.bar_label(ax.containers[2])
plt.ylabel('Total Sales')
plt.title('Total Sales by City and Branch Level')
```

Out[25]:

```
Text(0.5, 1.0, 'Total Sales by City and Branch Level')
```



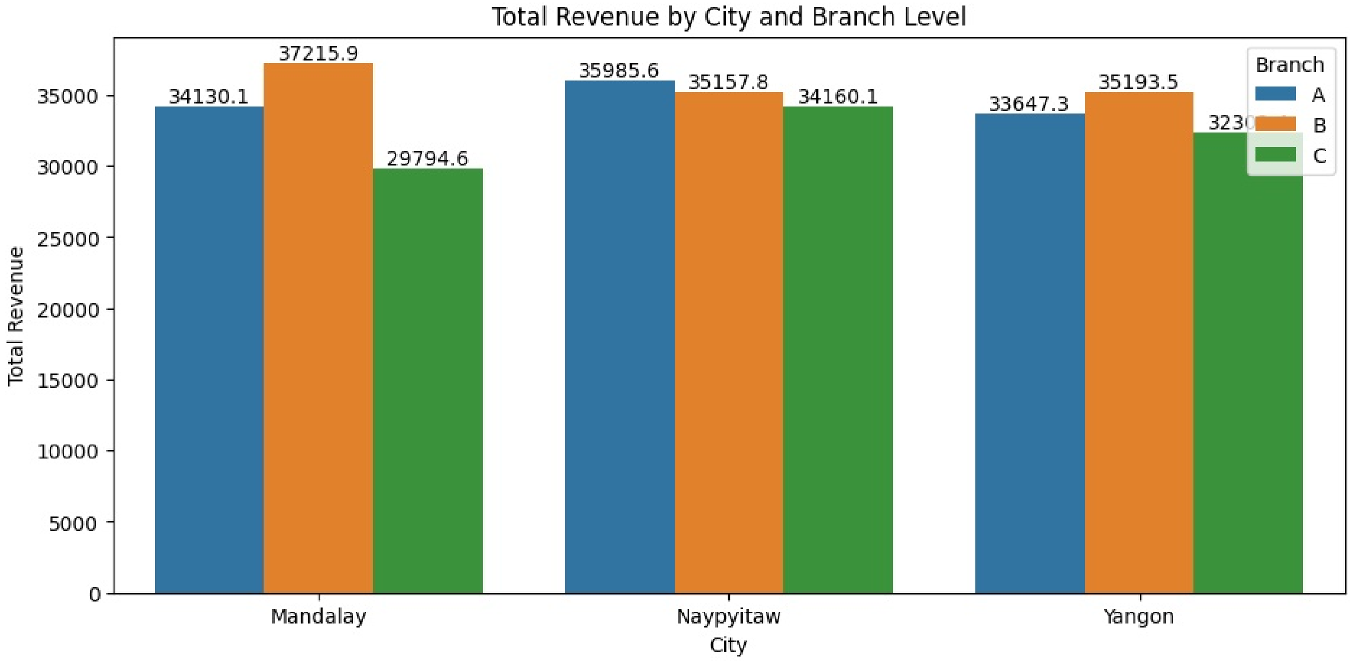
It is clear from the graph that Branch 'B' in the 'Mandalay' City is having maximum sales

In [24]:

```
plt.figure(figsize=(11,5))
ax=sns.barplot(x='City', y='Revenue',hue='Branch', data=performance)
ax.bar_label(ax.containers[0])
ax.bar_label(ax.containers[1])
ax.bar_label(ax.containers[2])
plt.ylabel('Total Revenue')
plt.title('Total Revenue by City and Branch Level')
```

Out[24]:

```
Text(0.5, 1.0, 'Total Revenue by City and Branch Level')
```



It is obvious that Branch 'B' in the 'Mandalay' City is driving maximum Revenue

Q.2 What is the average price of an item sold at each branch of the city

In [21]:

```
avg_price=df.groupby(['Branch','City'])['Unit price'].mean().reset_index()
avg_price.columns=['Branch','City','Avg Price']
avg_price
```

Out[21]:

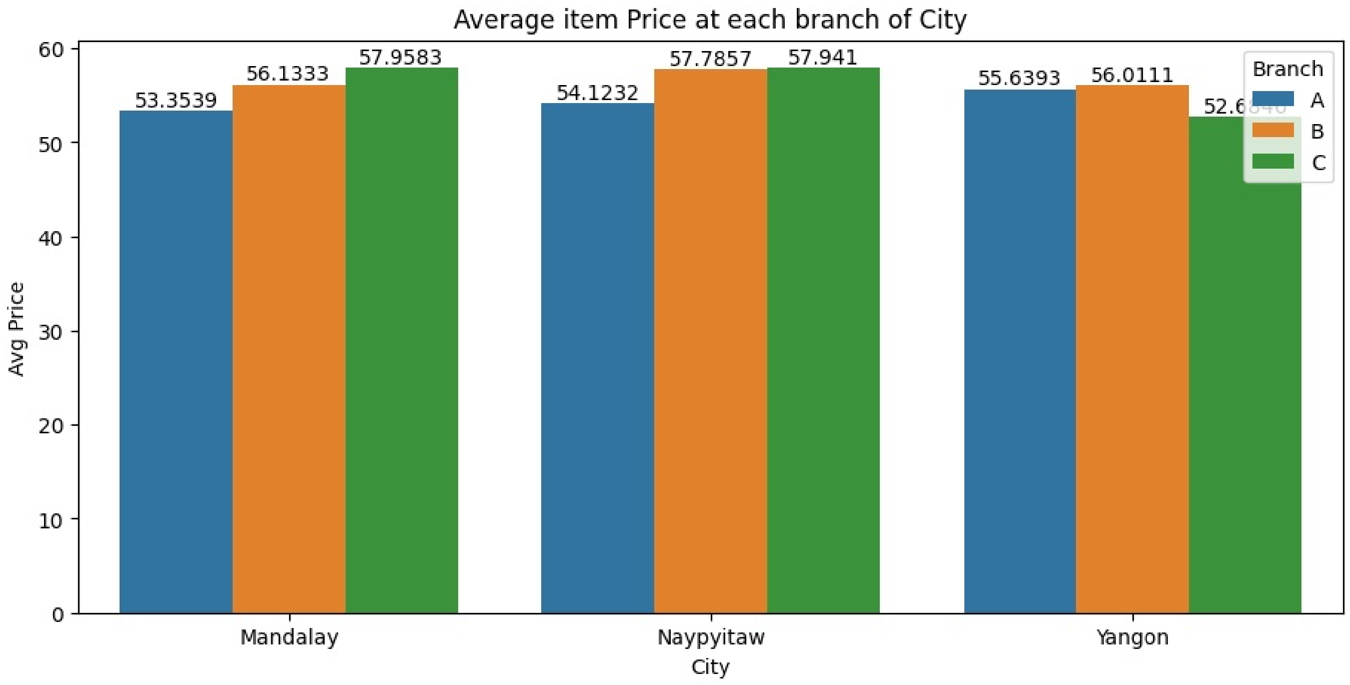
	Branch	City	Avg Price
0	A	Mandalay	53.353866
1	A	Naypyitaw	54.123182
2	A	Yangon	55.639298
3	B	Mandalay	56.133305
4	B	Naypyitaw	57.785688
5	B	Yangon	56.011062
6	C	Mandalay	57.958316
7	C	Naypyitaw	57.941009
8	C	Yangon	52.684602

In [29]:

```
plt.figure(figsize=(11,5))
ax=sns.barplot(x='City', y='Avg Price', hue='Branch', data=avg_price)
ax.bar_label(ax.containers[0])
ax.bar_label(ax.containers[1])
ax.bar_label(ax.containers[2])
plt.title('Average item Price at each branch of City')
```

Out[29]:

```
Text(0.5, 1.0, 'Average item Price at each branch of City')
```



It is clearly seen that average price of an item in Branch 'C' of City 'Mandalay' as well as city 'Naypyitaw' is greater as compare to any other

Q.3 Analyze the performance of sales and revenue, Month over Month across the Product line, Gender, and Payment Method, and identify the focus areas to get better sales for April 2019.

```
In [30]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 13 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Invoice ID     1000 non-null   object 
 1   Branch         1000 non-null   object 
 2   City           1000 non-null   object 
 3   Customer type  1000 non-null   object 
 4   Gender         1000 non-null   object 
 5   Product line   1000 non-null   object 
 6   Unit price     1000 non-null   float64
 7   Quantity       1000 non-null   int64  
 8   Date           1000 non-null   object 
 9   Time           1000 non-null   object 
 10  Payment        1000 non-null   object 
 11  Rating         1000 non-null   float64
 12  Revenue        1000 non-null   float64
dtypes: float64(3), int64(1), object(9)
memory usage: 101.7+ KB
```

Converting Data type of 'Date' column to 'datetime' format from 'Object'

```
In [31]: df['Date']= pd.to_datetime(df['Date'])
```

```
In [32]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 13 columns):
 #   Column         Non-Null Count  Dtype         
---  ------         --------------  -----         
 0   Invoice ID     1000 non-null   object        
 1   Branch         1000 non-null   object        
 2   City           1000 non-null   object        
 3   Customer type  1000 non-null   object        
 4   Gender         1000 non-null   object        
 5   Product line   1000 non-null   object        
 6   Unit price     1000 non-null   float64       
 7   Quantity       1000 non-null   int64         
 8   Date           1000 non-null   datetime64[ns]
 9   Time           1000 non-null   object        
 10  Payment        1000 non-null   object        
 11  Rating         1000 non-null   float64       
 12  Revenue        1000 non-null   float64       
dtypes: datetime64[ns](1), float64(3), int64(1), object(8)
memory usage: 101.7+ KB
```

In [34]:

```
df['Months']= df['Date'].dt.month             #making a month column
df
```

Out[34]:

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Date	Time	Payment	Rating	Revenue	Months
0	750-67-8428	A	Yangon	Member	Female	Health and beauty	74.69	7	2019-01-05	13:08	Ewallet	9.1	522.83	1
1	226-31-3081	A	Naypyitaw	Normal	Female	Electronic accessories	15.28	5	2019-03-08	10:29	Cash	9.6	76.40	3
2	631-41-3108	A	Yangon	Normal	Male	Home and lifestyle	46.33	7	2019-03-03	13:23	Credit card	7.4	324.31	3
3	123-19-1176	B	Yangon	Member	Male	Health and beauty	58.22	8	2019-01-27	20:33	Ewallet	8.4	465.76	1
4	373-73-7910	C	Yangon	Normal	Male	Sports and travel	86.31	7	2019-02-08	10:37	Ewallet	5.3	604.17	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	233-67-5758	A	Naypyitaw	Normal	Male	Health and beauty	40.35	1	2019-01-29	13:46	Ewallet	6.2	40.35	1
996	303-96-2227	A	Mandalay	Normal	Female	Home and lifestyle	97.38	10	2019-03-02	17:16	Ewallet	4.4	973.80	3
997	727-02-1313	A	Yangon	Member	Male	Food and beverages	31.84	1	2019-02-09	13:22	Cash	7.7	31.84	2
998	347-56-2442	B	Yangon	Normal	Male	Home and lifestyle	65.82	1	2019-02-22	15:33	Cash	4.1	65.82	2
999	849-09-3807	C	Yangon	Member	Female	Fashion accessories	88.34	7	2019-02-18	13:28	Cash	6.6	618.38	2

1000 rows × 14 columns

In [35]:

```
monthly_performance= df.groupby(['Months','Product line','Gender','Payment'])[['Quantity','Revenue']].sum().res
monthly_performance
```

Out[35]:

	Months	Product line	Gender	Payment	Quantity	Revenue
0	1	Electronic accessories	Female	Cash	52	2731.86
1	1	Electronic accessories	Female	Credit card	54	3045.42
2	1	Electronic accessories	Female	Ewallet	43	1576.48
3	1	Electronic accessories	Male	Cash	62	3380.29
4	1	Electronic accessories	Male	Credit card	43	2248.65
...	...	...	...	...	...	...
103	3	Sports and travel	Female	Credit card	52	2863.86
104	3	Sports and travel	Female	Ewallet	53	3398.57
105	3	Sports and travel	Male	Cash	36	2084.19
106	3	Sports and travel	Male	Credit card	60	3633.90
107	3	Sports and travel	Male	Ewallet	86	4930.61

108 rows × 6 columns

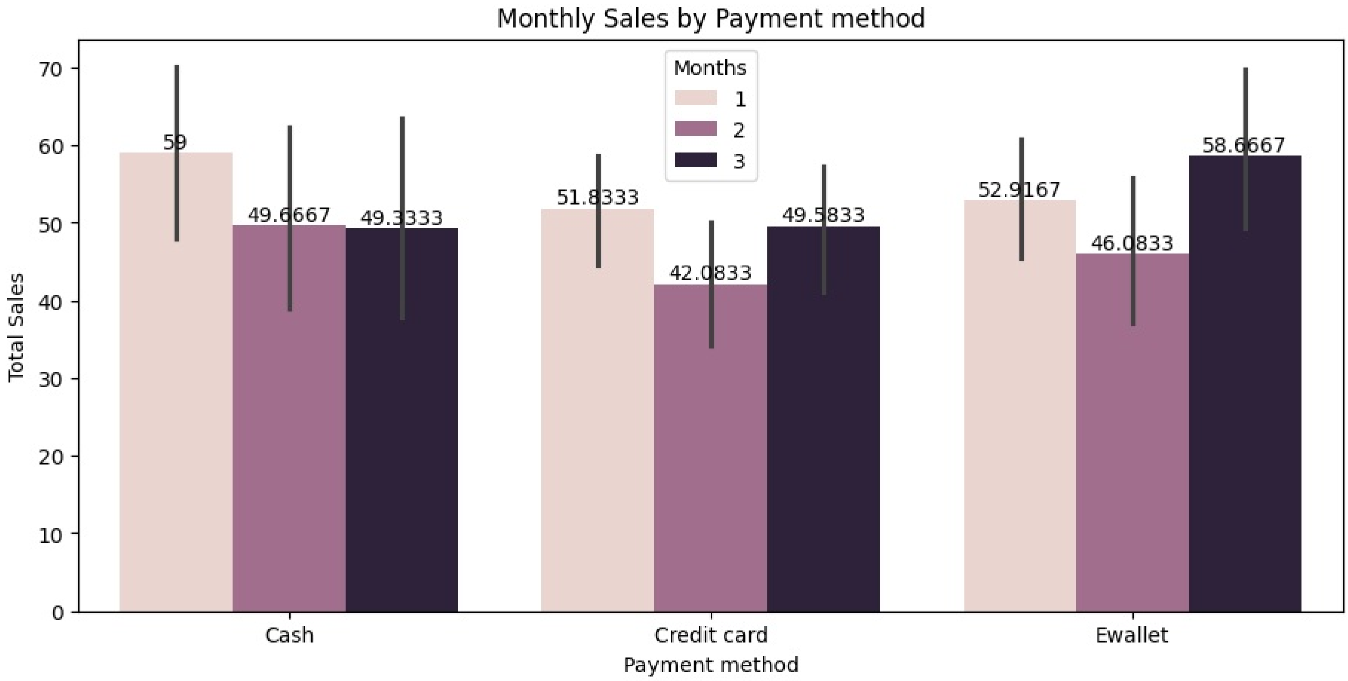
Visualization

In [38]:

```
plt.figure(figsize=(11,5))
ax=sns.barplot(x='Payment',y='Quantity', hue='Months', data=monthly_performance)
ax.bar_label(ax.containers[0])
ax.bar_label(ax.containers[1])
ax.bar_label(ax.containers[2])
plt.ylabel('Total Sales')
plt.xlabel('Payment method')
plt.title('Monthly Sales by Payment method')
```

Out[38]:

```
Text(0.5, 1.0, 'Monthly Sales by Payment method')
```



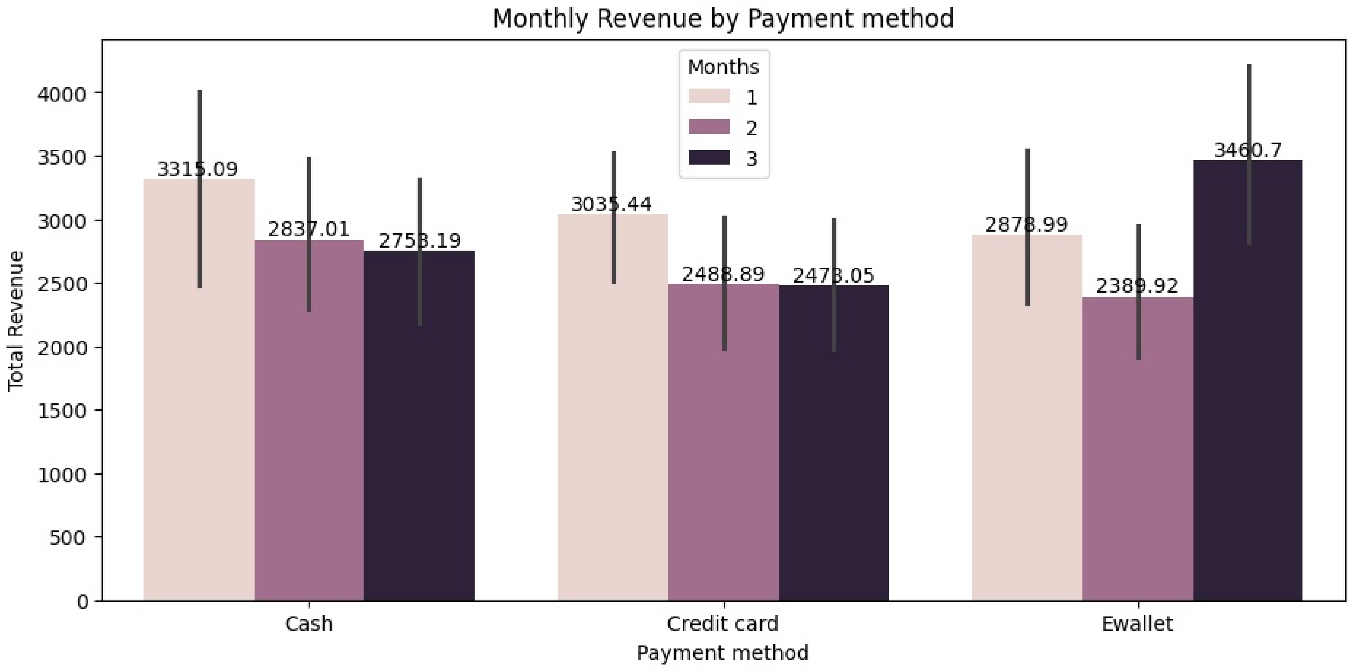
It is clearly seen from the graph that 'Ewallet' payment methods popularity is increasing in the month of March. Also, customer like 'Cash' payment method most

In [40]:

```
plt.figure(figsize=(11,5))
ax=sns.barplot(x='Payment',y='Revenue', hue='Months', data=monthly_performance)
ax.bar_label(ax.containers[0])
ax.bar_label(ax.containers[1])
ax.bar_label(ax.containers[2])
plt.ylabel('Total Revenue')
plt.xlabel('Payment method')
plt.title('Monthly Revenue by Payment method')
```

Out[40]:

```
Text(0.5, 1.0, 'Monthly Revenue by Payment method')
```

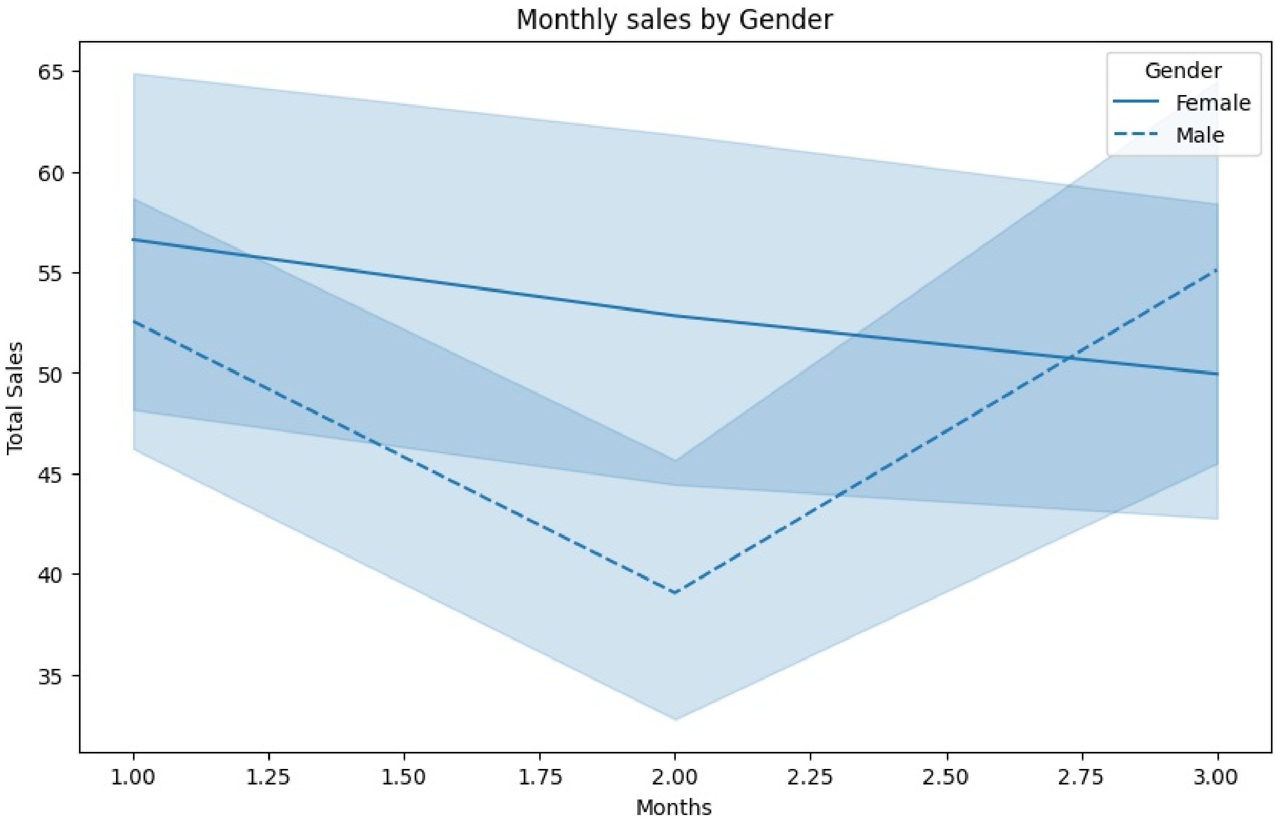


Now in terms of revenue we can see that Ewallet is top payment method to drive maximum revenue in March Month. In rest of

the month 'Cash' payment method is used most

```
In [43]: plt.figure(figsize=(10,6))
         sns.lineplot(x='Months', y='Quantity', style='Gender',  data=monthly_performance)
         plt.title('Monthly sales by Gender')
         plt.ylabel('Total Sales')
```

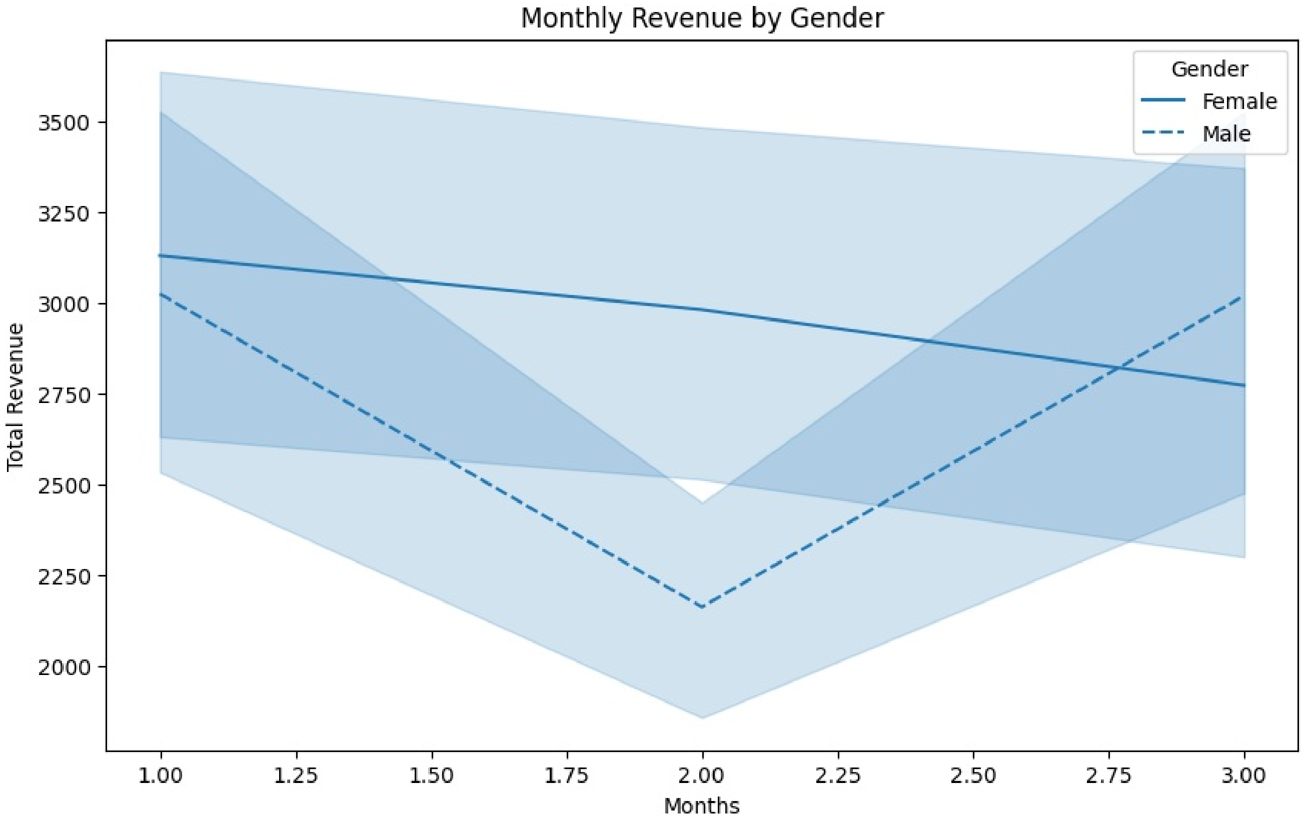
```
Out[43]: Text(0, 0.5, 'Total Sales')
```



It is interesting to note that Female is purchasing more than Males but the trend is decreasing in the March Month

```
In [44]: plt.figure(figsize=(10,6))
         sns.lineplot(x='Months', y='Revenue', style='Gender',  data=monthly_performance)
         plt.title('Monthly Revenue by Gender')
         plt.ylabel('Total Revenue')
```

```
Out[44]: Text(0, 0.5, 'Total Revenue')
```



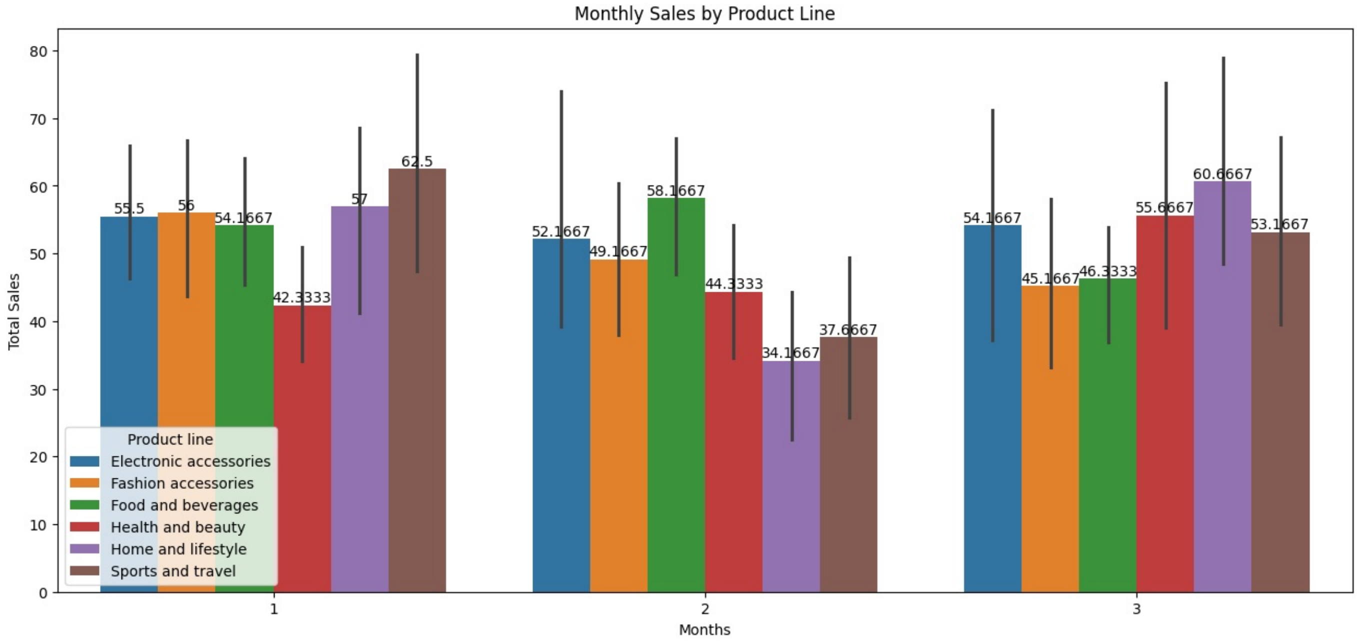
It is obvious that Female is driving more Revenue than Males but the trend is decreasing in the March Month

In [51]:

```
plt.figure(figsize=(16,7))
ax=sns.barplot(x='Months',y='Quantity', hue='Product line', data=monthly_performance)
ax.bar_label(ax.containers[0])
ax.bar_label(ax.containers[1])
ax.bar_label(ax.containers[2])
ax.bar_label(ax.containers[3])
ax.bar_label(ax.containers[4])
ax.bar_label(ax.containers[5])
plt.ylabel('Total Sales')
plt.xlabel('Months')
plt.title('Monthly Sales by Product Line')
```

Out[51]:

```
Text(0.5, 1.0, 'Monthly Sales by Product Line')
```



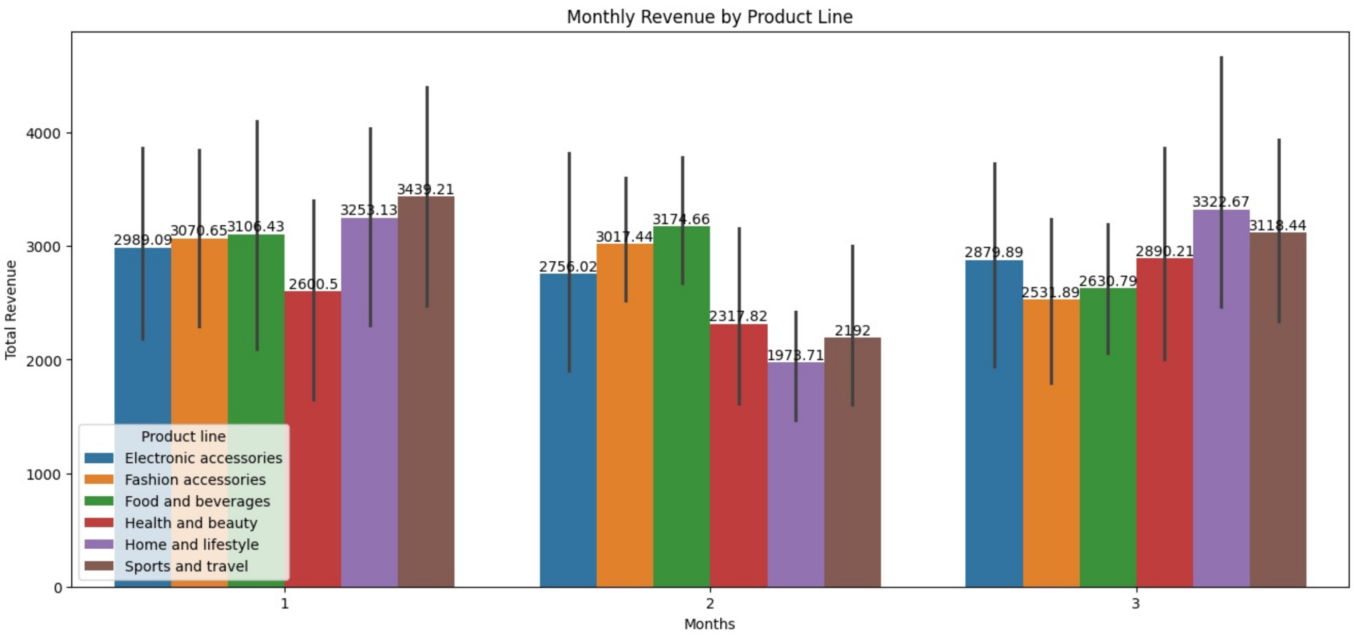
It is clear from the graph that 'Home and lifestyle' and 'Sports and travel' product lines were perfoming good in terms of sales in January month but the trends decreased in February month but then it can be seen that in March month the trend is starting to increase

In [53]:

```
plt.figure(figsize=(16,7))
ax=sns.barplot(x='Months',y='Revenue', hue='Product line', data=monthly_performance)
ax.bar_label(ax.containers[0])
ax.bar_label(ax.containers[1])
ax.bar_label(ax.containers[2])
ax.bar_label(ax.containers[3])
ax.bar_label(ax.containers[4])
ax.bar_label(ax.containers[5])
plt.ylabel('Total Revenue')
plt.xlabel('Months')
plt.title('Monthly Revenue by Product Line')
```

Out[53]:

```
Text(0.5, 1.0, 'Monthly Revenue by Product Line')
```



Same thing in terms of Revenue : It is clear from the graph that 'Home and lifestyle' and 'Sports and travel' product lines were perfoming good in terms of Revenue in January month but the trends decreased in February month but then it can be seen that in March month the trend is starting to increase

CONCLUSION

It is clear from our analysis to increase the sales in 'April 2019'
1. Runnning some offers to attract customers to use 'Ewallet' since the trend of using Ewallet in 'March' month is increasing as seen in the graph.
2. Also, attracting 'Female' customers is very important since majority are Female buyers and it is seen that the trend was decreasing in the month on March.
3.Running some offers on Products like 'Home and lifestyle' and 'Sports and travel' may increase the sales in April 2019.

Project Submitted By
Ajit Tiwari
My Portfolio: https://ajitiwari.github.io/

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js