WALMART-CUSTOMER-PURCHARSES-EDA-PROJECT-VIVEK-CHAUHAN

WALMART-CUSTOMER-PURCHARSES Analysis

The analysis is divided into four main parts:

1. Data understanding

2. Data cleaning (cleaning missing values, removing redundant columns etc.)

3. Data Analysis

4. Recommendations

In [1]:
```
# Upload the necessary libraries for analysis

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

In [2]:
```
# load the datasets

data = pd.read_csv("Walmart_customer_purchases.csv")
data.head()
```

Out[2]:

	Customer_ID	Age	Gender	City	Category	Product_Name	Purchase_Date	Purchase_Amount	Payment_Method	Discount_Ap
0	84607c1f-910c-44d5-b89f-e1ee06dd34c0	49	Female	New Cynthia	Electronics	Smartphone	2024-08-30	253.26	Cash on Delivery	
1	f2a81712-a73e-4424-8b39-4c615a0bd4ea	36	Other	Cruzport	Clothing	T-Shirt	2024-12-21	73.19	Debit Card	
2	da9be287-8b0e-4688-bccd-1a2cdd7567c6	52	Male	Jeffreytown	Beauty	Perfume	2024-12-26	125.62	Credit Card	
3	50ec6932-3ac7-492f-9e55-4b148212f302	47	Female	Jenniferburgh	Electronics	Smartwatch	2024-11-04	450.32	Credit Card	
4	8fdc3098-fc75-4b0f-983c-d8d8168c6362	43	Other	Kingshire	Electronics	Smartphone	2024-10-07	369.28	Credit Card	

Data-Understanding

In [3]:
```
# print how many columns and rows in dataset

data.shape
```

Out[3]:
```
(50000, 12)
```

In [4]:
```
# print the information about the datasets
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50000 entries, 0 to 49999
Data columns (total 12 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   Customer_ID       50000 non-null  object 
 1   Age               50000 non-null  int64  
 2   Gender            50000 non-null  object 
 3   City              50000 non-null  object 
 4   Category          50000 non-null  object 
 5   Product_Name      50000 non-null  object 
 6   Purchase_Date     50000 non-null  object 
 7   Purchase_Amount   50000 non-null  float64
 8   Payment_Method    50000 non-null  object 
 9   Discount_Applied  50000 non-null  object 
 10  Rating            50000 non-null  int64  
 11  Repeat_Customer   50000 non-null  object 
dtypes: float64(1), int64(2), object(9)
memory usage: 4.6+ MB
```

In [5]:
```
# print the statistics about the dataset

data.describe()
```

Out[5]:

	Age	Purchase_Amount	Rating
count	50000.000000	50000.000000	50000.000000
mean	38.945220	255.532230	2.998680
std	12.398137	141.574416	1.417956
min	18.000000	10.010000	1.000000
25%	28.000000	133.050000	2.000000
50%	39.000000	255.045000	3.000000
75%	50.000000	378.912500	4.000000
max	60.000000	499.990000	5.000000

In [6]:
```
# print all the columns names which is present in the dataset

data.columns
```

Out[6]:
```
Index(['Customer_ID', 'Age', 'Gender', 'City', 'Category', 'Product_Name',
       'Purchase_Date', 'Purchase_Amount', 'Payment_Method',
       'Discount_Applied', 'Rating', 'Repeat_Customer'],
      dtype='object')
```

In [7]:
```
# check the duplicate entry/custormer-id for the customer

data["Customer_ID"].duplicated().sum()
```

Out[7]:
```
0
```

In [8]:
```
# copy the purchase amount column in anther column.

data['Purchase_Amount_Real'] = data["Purchase_Amount"].copy()
data['Purchase_Amount_Real']
```

Out[8]:
```
0        253.26
1         73.19
2        125.62
3        450.32
4        369.28
          ...
49995    391.48
49996    272.09
49997    280.81
49998    140.74
49999     18.90
Name: Purchase_Amount_Real, Length: 50000, dtype: float64
```

Handle Missing Values If Any

In [9]:
```
# check null values in the dataset

data.isnull().sum()
```

Out[9]:
```
Customer_ID             0
Age                     0
Gender                  0
City                    0
Category                0
Product_Name            0
Purchase_Date           0
Purchase_Amount         0
Payment_Method          0
Discount_Applied        0
Rating                  0
Repeat_Customer         0
Purchase_Amount_Real    0
dtype: int64
```

Data-Analysis

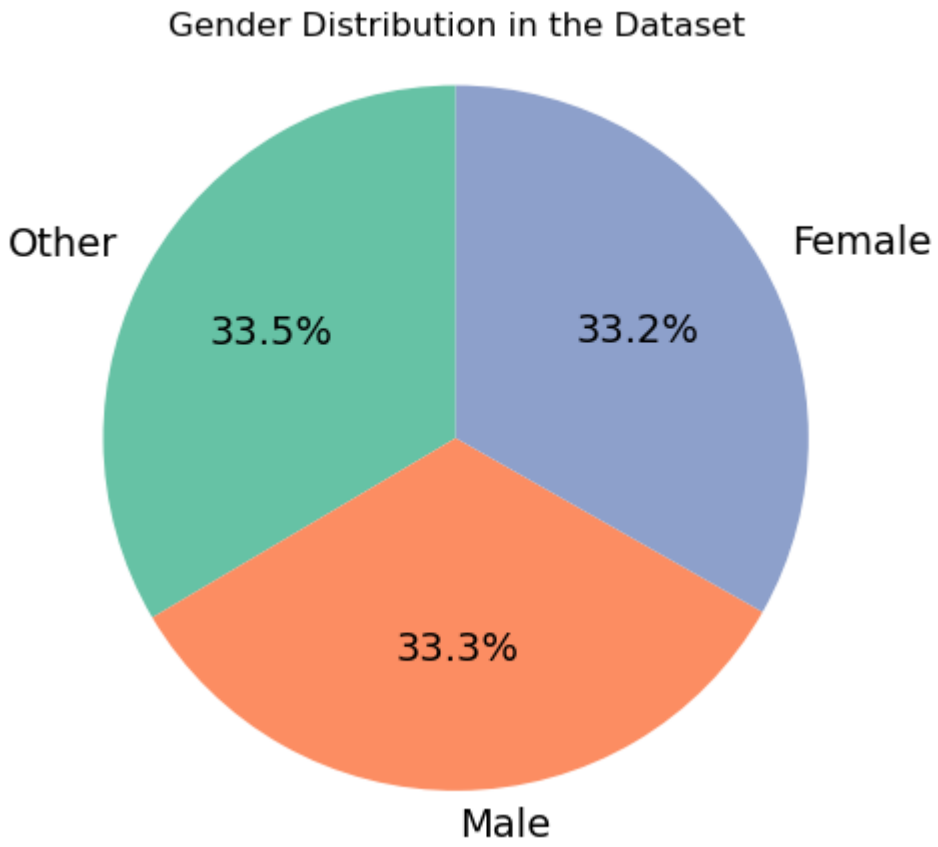
Uni-Variate-Analysis

In [10]:
```
# count the Gender columns

data.Gender.value_counts()
```

Out[10]:
```
Gender
Other     16751
Male      16644
Female    16605
Name: count, dtype: int64
```

The high count in the "Other" gender category may be due to missing or improperly recorded data, and should not be interpreted as a definitive customer trend.

In [11]:
```
gender_counts = data['Gender'].value_counts()

plt.figure(figsize=(5,5))
plt.pie(gender_counts, labels=gender_counts.index, autopct='%1.1f%%', startangle=90, colors=plt.cm.Set2.colors,textprops={'col
plt.title("Gender Distribution in the Dataset")
plt.axis('equal')  # Ensure the pie chart is circular
plt.show()
```



The "Other" gender category shows a higher number of customers in the Walmart purchases dataset; however, this may be due to missing or incorrectly recorded data and should be interpreted with caution.

In [12]:
```
sorted_city = data.City.value_counts().sort_values(ascending = False).head(10)
sorted_city
```

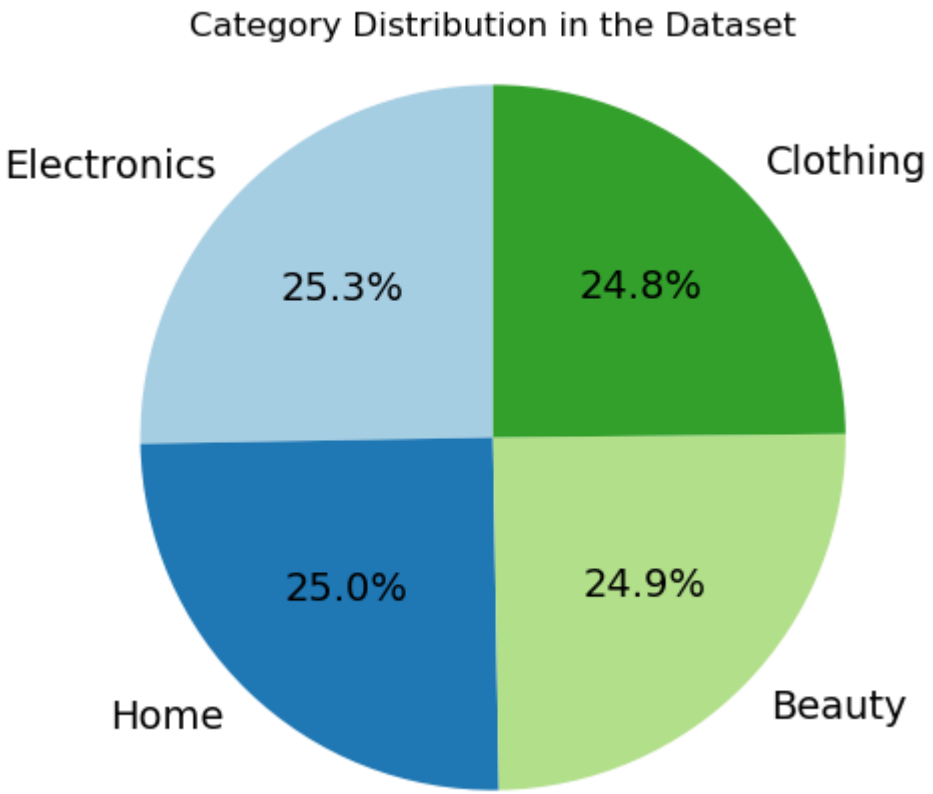
Out[12]:
```
City
New Michael          44
North Michael        42
Lake Michael         41
East Michael         36
South Michael        34
Port Michael         34
West Michael         33
West Jennifer        32
East Christopher     32
East Robert          31
Name: count, dtype: int64
```

"New Michael" appears as the most frequently occurring city, indicating the highest number of customers purchasing from Walmart in that location.

In [13]:
```
sorted_cat = data.Category.value_counts().sort_values(ascending = False)
sorted_cat
```

Out[13]:
```
Category
Electronics    12642
Home           12492
Beauty         12447
Clothing       12419
Name: count, dtype: int64
```

In [14]:
```
category_counts = data['Category'].value_counts()

plt.figure(figsize=(5,5))
plt.pie(category_counts, labels=category_counts.index, autopct='%1.1f%%', startangle=90, colors=plt.cm.Paired.colors,textprops
plt.title("Category Distribution in the Dataset")
plt.axis('equal')  # Ensure the pie chart is circular
plt.show()
```



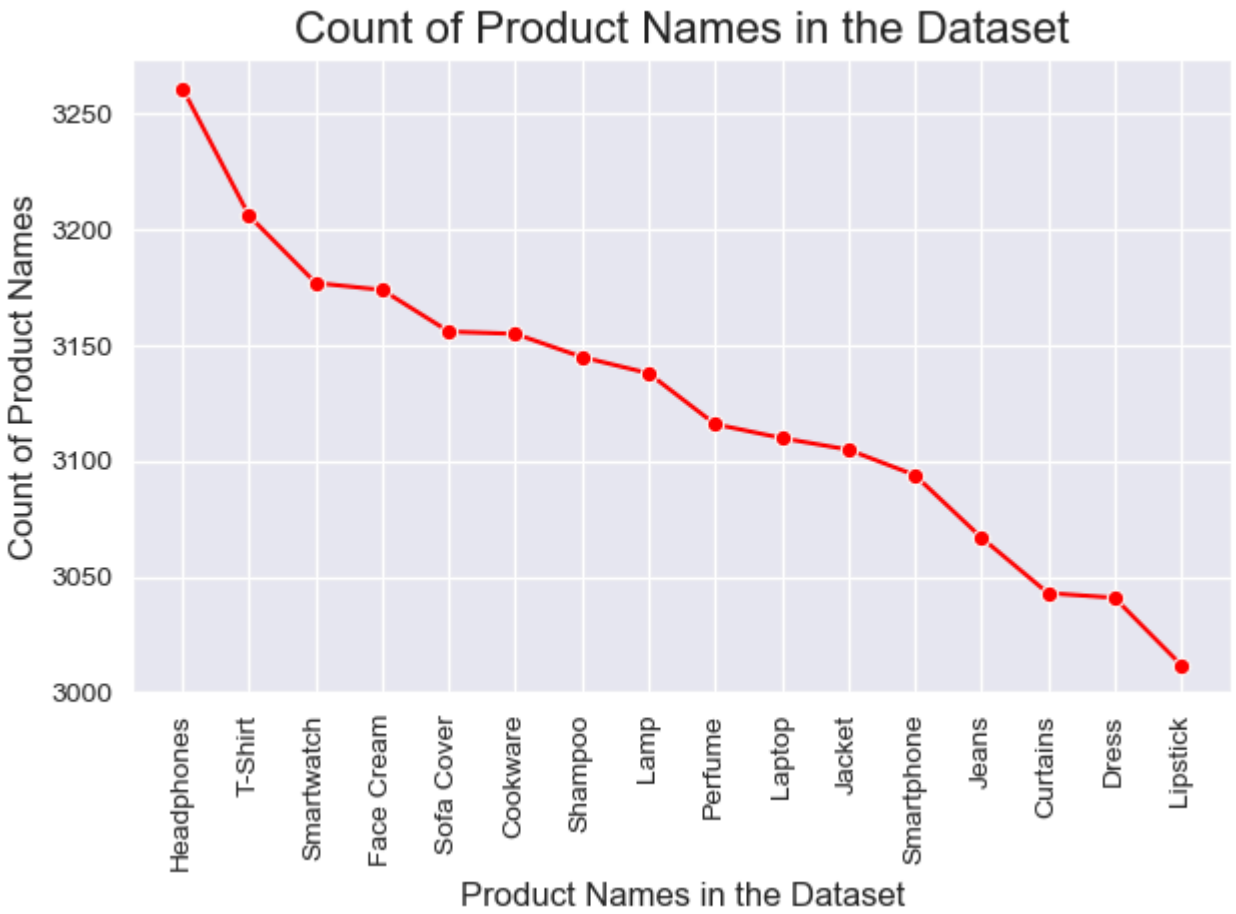
The above chart shows that the Electronics category has the highest number of purchases among Walmart customers.

In [15]:
```
data.Product_Name.value_counts()
```

Out[15]:
```
Product_Name
Headphones    3261
T-Shirt       3206
Smartwatch    3177
Face Cream    3174
Sofa Cover    3156
Cookware      3155
Shampoo       3145
Lamp          3138
Perfume       3116
Laptop        3110
Jacket        3105
Smartphone    3094
Jeans         3067
Curtains      3043
Dress         3041
Lipstick      3012
Name: count, dtype: int64
```

In [68]:
```
# Count the occurrences of each Product_Name
product_counts = data['Product_Name'].value_counts().reset_index()

# Rename the columns for better clarity
product_counts.columns = ['Product_Name', 'Count']

# Set the style for the plot
sns.set_style("darkgrid")

# Create a lineplot with Product_Name on the x-axis and count on the y-axis
sns.lineplot(x='Product_Name', y='Count', data=product_counts, marker='o', color='red')

# Adding title and labels
plt.title("Count of Product Names in the Dataset", fontsize=16)
plt.xlabel("Product Names in the Dataset", fontsize=12)
plt.ylabel("Count of Product Names", fontsize=12)

# Rotating x-axis labels for better readability
plt.xticks(rotation=90)

# Show the plot
```

```
plt.tight_layout()
plt.show()
```



While all products show high sales at Walmart, the Headphones product category is particularly in high demand.

In [17]:
```
# convert the date is object type to date type

data['Purchase_Date'] = pd.to_datetime(data['Purchase_Date'])
```

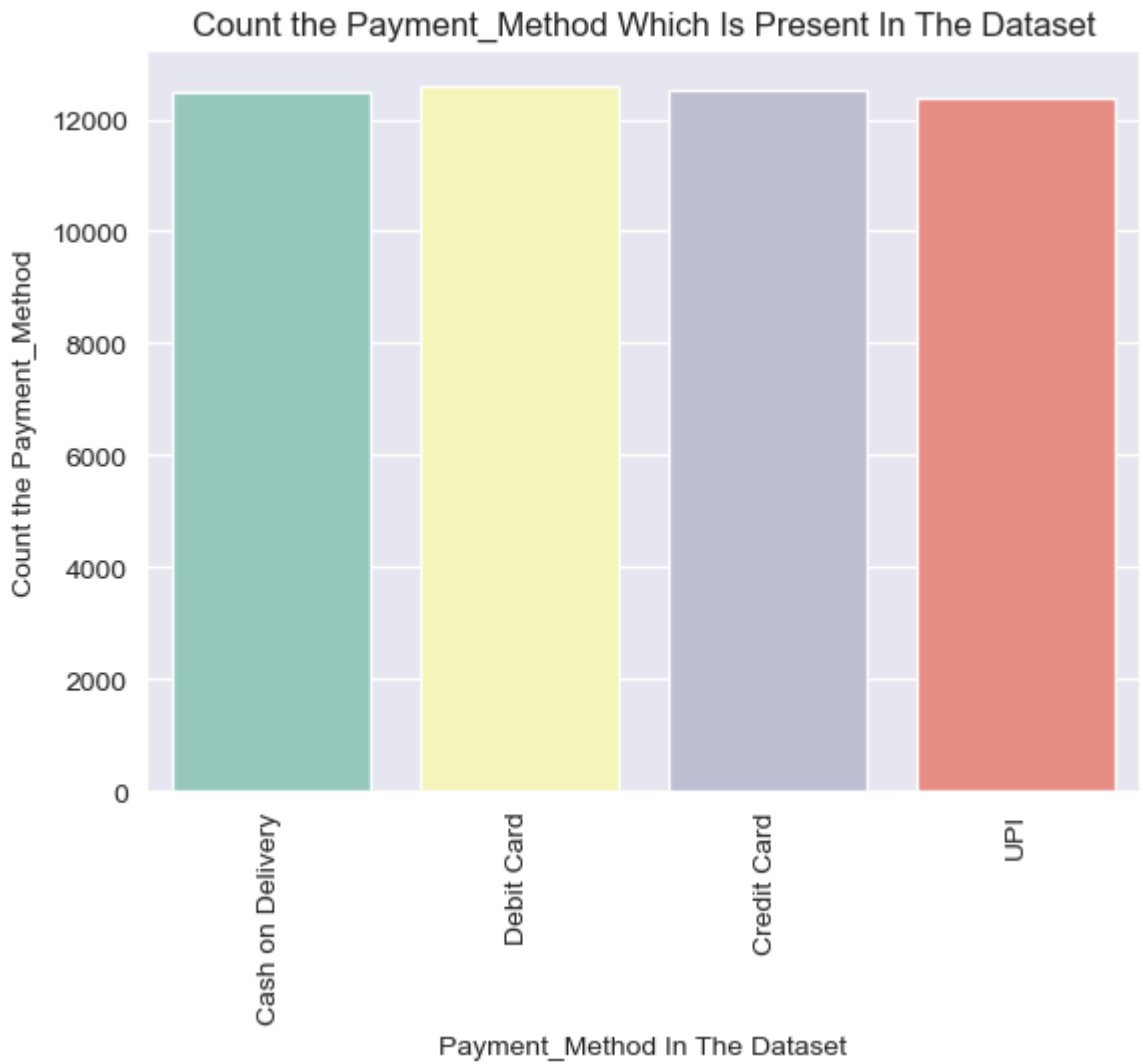
In [18]:
```
data.dtypes
```

Out[18]:
```
Customer_ID                 object
Age                          int64
Gender                      object
City                        object
Category                    object
Product_Name                object
Purchase_Date       datetime64[ns]
Purchase_Amount            float64
Payment_Method              object
Discount_Applied            object
Rating                       int64
Repeat_Customer             object
Purchase_Amount_Real       float64
dtype: object
```

In [19]:
```
# let's count the purchase methods

data.Payment_Method.value_counts()
```

Out[19]:
```
Payment_Method
Debit Card          12589
Credit Card         12528
Cash on Delivery    12496
UPI                 12387
Name: count, dtype: int64
```

In [20]:
```
sns.set_style("darkgrid")
sns.countplot(x = "Payment_Method",data=data,palette = "Set3")
plt.title("Count the Payment_Method Which Is Present In The Dataset")
plt.xlabel("Payment_Method In The Dataset")
plt.ylabel("Count the Payment_Method")
plt.xticks(rotation = "vertical")
plt.show()
```

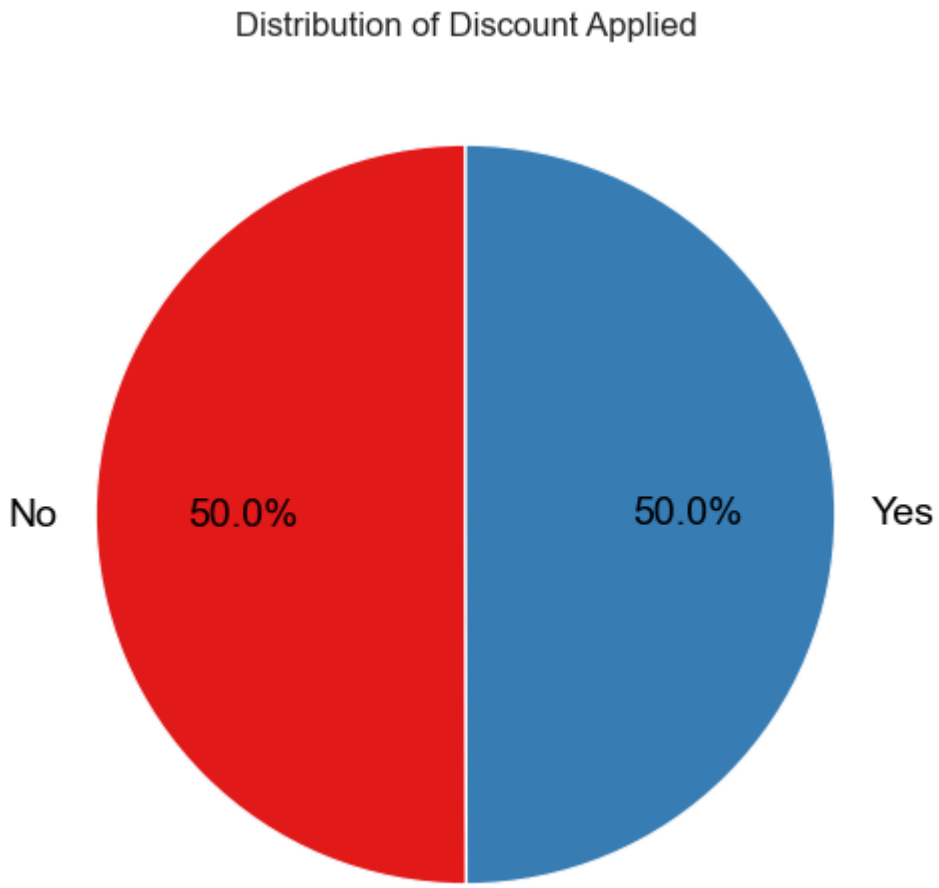


The majority of customers prefer to make payments using Debit Cards.

In [21]:
```
data.Discount_Applied.value_counts()
```

Out[21]:
```
Discount_Applied
No     25018
Yes    24982
Name: count, dtype: int64
```

In [71]:
```
# Assuming 'Discount_Applied' is the column you want to plot a pie chart for
discount_counts = data['Discount_Applied'].value_counts()

# Plotting the pie chart
plt.figure(figsize=(6, 6))
plt.pie(discount_counts, labels=discount_counts.index, autopct='%1.1f%%', startangle=90, colors=sns.color_palette("Set1", len(
plt.title("Distribution of Discount Applied")
plt.show()
```

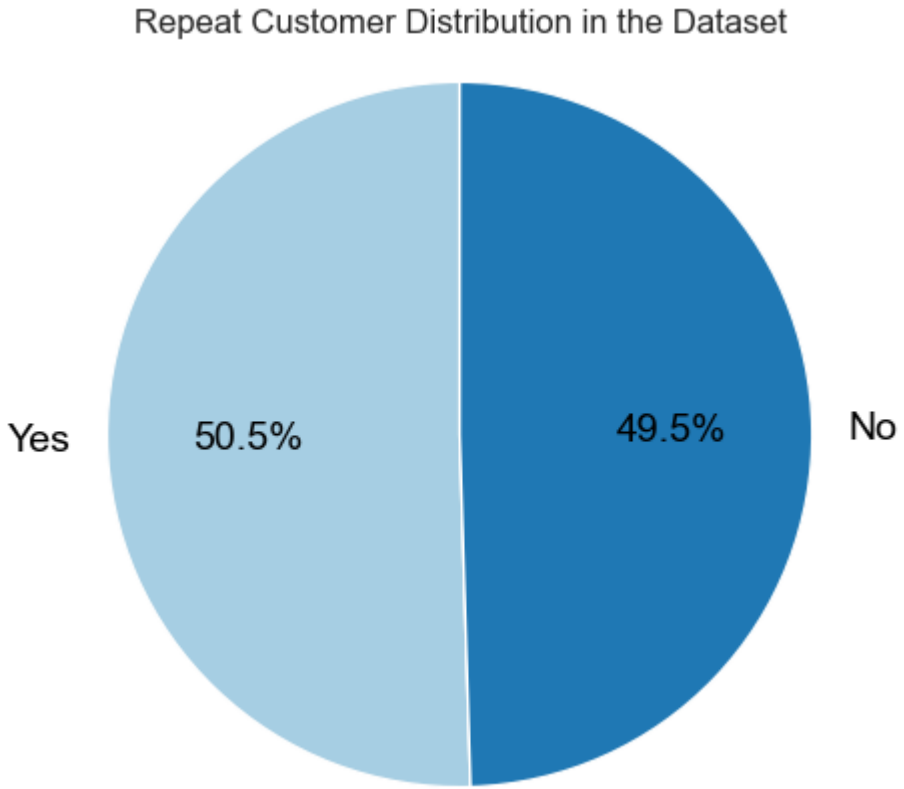


There is no significant difference in the discount applied, but the majority of Walmart products have no discounts applied.

In [23]:
```
data.Repeat_Customer.value_counts()
```

Out[23]:
```
Repeat_Customer
Yes     25244
No      24756
Name: count, dtype: int64
```

In [24]:
```
repeat_customer_counts = data['Repeat_Customer'].value_counts()

plt.figure(figsize=(5,5))
plt.pie(repeat_customer_counts, labels=repeat_customer_counts.index, autopct='%1.1f%%', startangle=90, colors=plt.cm.Paired.co
plt.title("Repeat Customer Distribution in the Dataset")
plt.axis('equal')  # Ensure the pie chart is circular
plt.show()
```



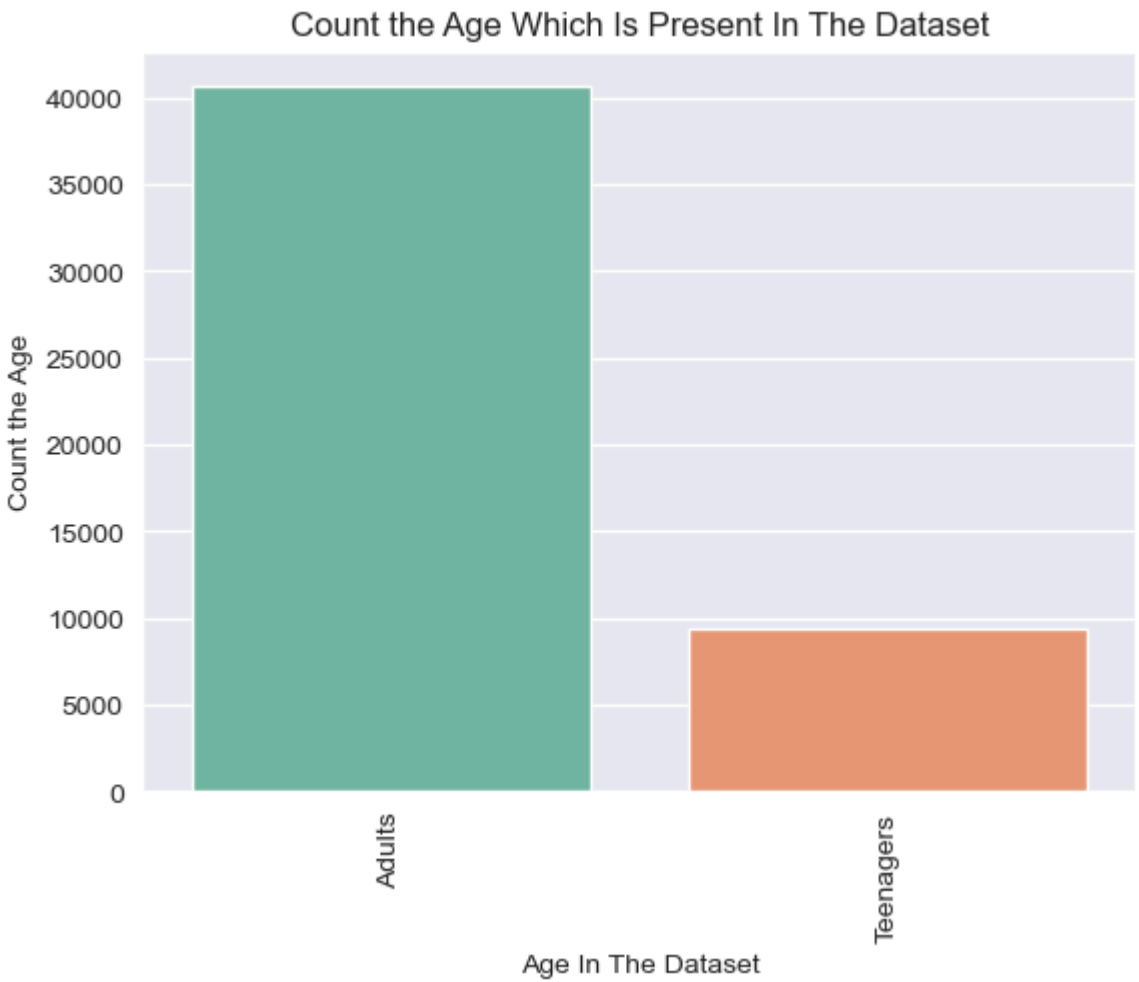
A majority of customers at Walmart are repeat buyers.

In [25]:
```
# let's categorise the age columns so we get better understand walmart purchasers.

def age_cat(x):
    if(x<=25):
        return "Teenagers"
    elif(x>=25 & x<=60):
        return "Adults"
    else:
        return "Senior_Citizens"


data["Age_cat"] = data["Age"].apply(age_cat)
```

In [26]:
```
data["Age_cat"].value_counts()
```

Out[26]:
```
Age_cat
Adults       40617
Teenagers     9383
Name: count, dtype: int64
```

In [27]:
```
sns.set_style("darkgrid")
sns.countplot(x = "Age_cat",data=data,palette = "Set2")
plt.title("Count the Age Which Is Present In The Dataset")
plt.xlabel("Age In The Dataset")
plt.ylabel("Count the Age")
plt.xticks(rotation = "vertical")
plt.show()
```

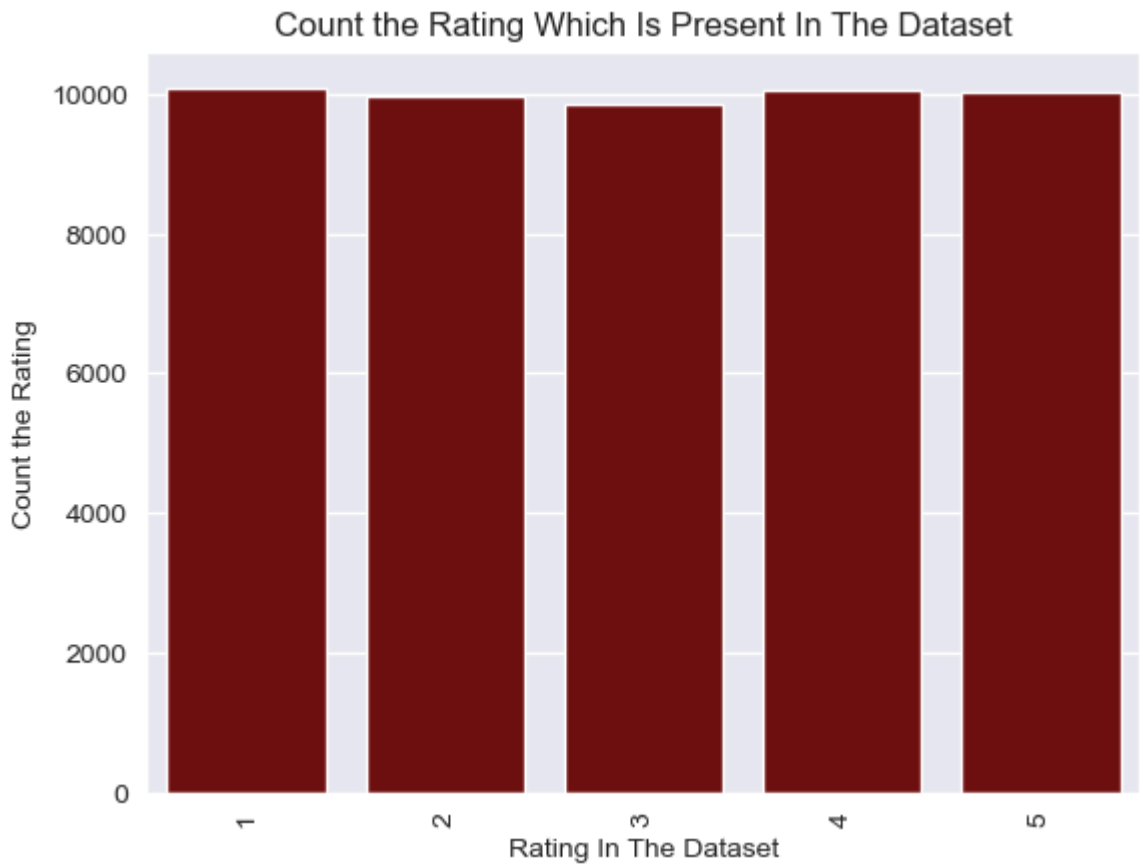


The majority of customers at Walmart are adults.

In [28]:
```
data.Rating.value_counts().sort_values(ascending = False)
```

Out[28]:
```
Rating
1    10099
4    10045
5    10029
2     9971
3     9856
Name: count, dtype: int64
```

In [29]:
```
sns.set_style("darkgrid")
sns.countplot(x = "Rating",data=data,color = "maroon")
plt.title("Count the Rating Which Is Present In The Dataset")
plt.xlabel("Rating In The Dataset")
plt.ylabel("Count the Rating")
plt.xticks(rotation = "vertical")
plt.show()
```



Walmart customers mostly rate products/services with 1, 4, or 5 stars.

In [30]:
```
data.Purchase_Amount.value_counts()
```

Out[30]:
```
Purchase_Amount
420.37    7
58.13     7
240.00    6
253.31    6
221.95    6
         ..
455.12    1
340.73    1
448.61    1
305.00    1
391.48    1
Name: count, Length: 31378, dtype: int64
```

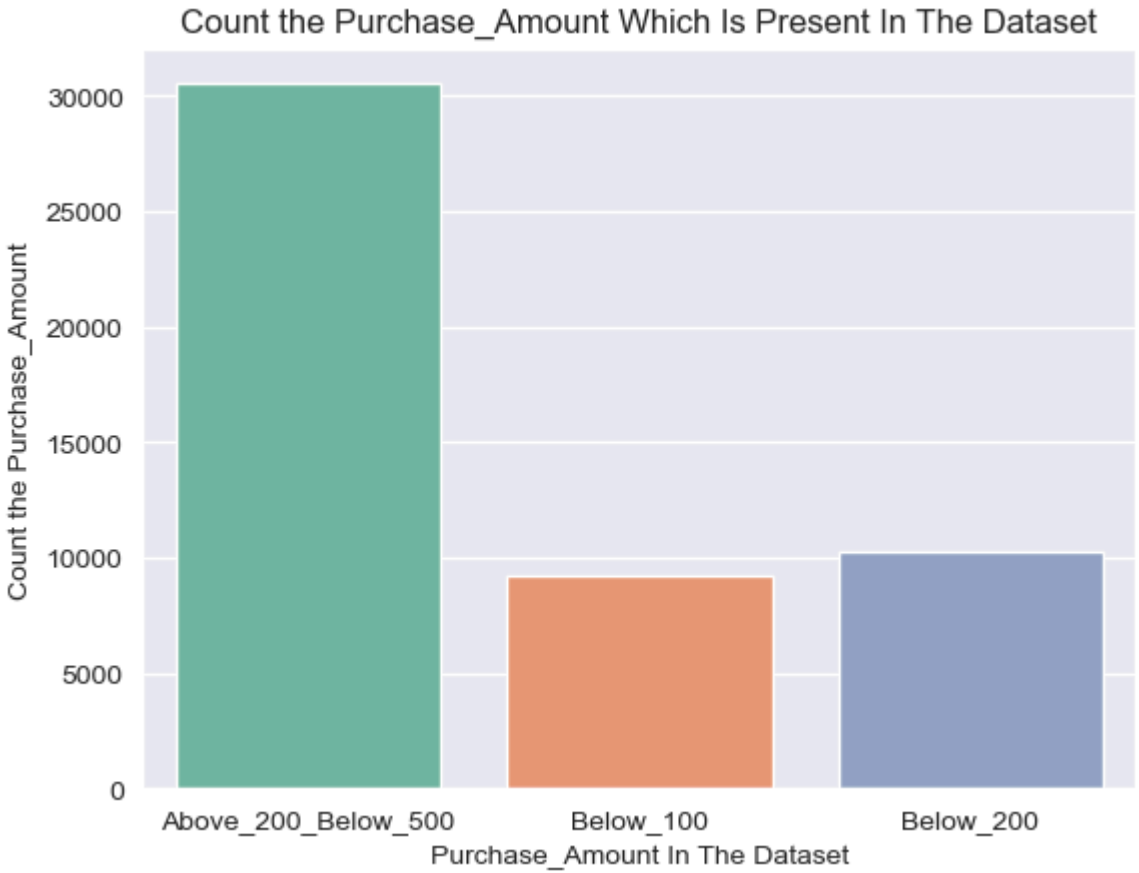
In [31]:
```
# convert the float amount into the integer amount

data['Purchase_Amount']= data.Purchase_Amount.astype(int)
data['Purchase_Amount']
```

Out[31]:
```
0        253
1         73
2        125
3        450
4        369
        ...
49995    391
49996    272
49997    280
49998    140
49999     18
Name: Purchase_Amount, Length: 50000, dtype: int32
```

In [32]:
```
def purchase_cat(x):
    if(x<=100):
        return "Below_100"
    elif(x>100 and x<=200):
        return "Below_200"
    elif(x>200 and x<=500):
        return "Above_200_Below_500"
    else:
```

```
        return "Above_500"

data["Purchase_Amount"] = data['Purchase_Amount'].apply(purchase_cat)
```

In [33]:
```
sns.set_style("darkgrid")
sns.countplot(x = "Purchase_Amount",data=data,palette = "Set2")
plt.title("Count the Purchase_Amount Which Is Present In The Dataset")
plt.xlabel("Purchase_Amount In The Dataset")
plt.ylabel("Count the Purchase_Amount")
plt.show()
```



The majority of customers at Walmart make purchases below the ₹500 amount and Above ₹200 amount.

In [34]:
```
# Create separate columns for Year, Month, and Day

data['year'] = data['Purchase_Date'].dt.year
data['month'] = data['Purchase_Date'].dt.month
data['day'] = data['Purchase_Date'].dt.day
```

In [35]:
```
# year wise values counts

data.year.value_counts()
```

Out[35]:
```
year
2024    44533
2025     5467
Name: count, dtype: int64
```

In [36]:
```
# month wise values counts

data.month.value_counts()
```

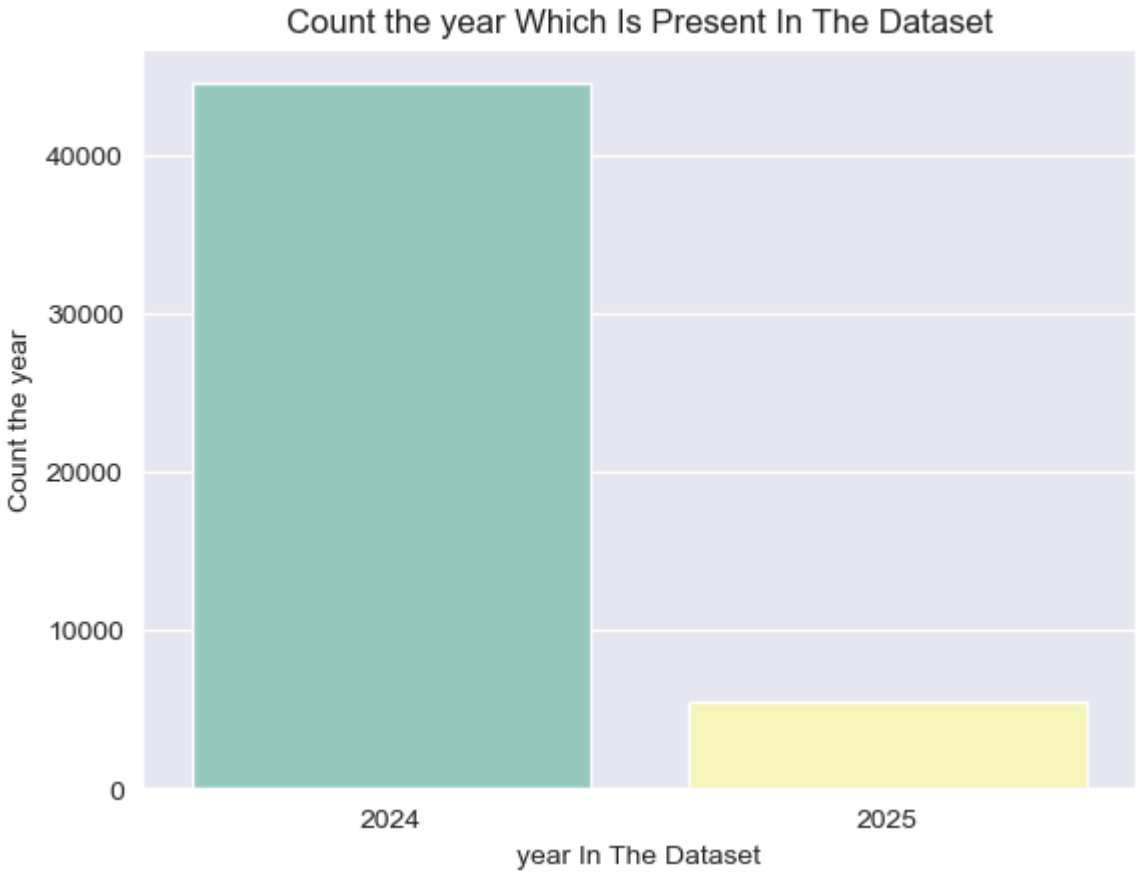
Out[36]:
```
month
3     4301
10    4257
12    4244
1     4230
5     4227
8     4217
7     4197
11    4148
9     4112
4     4079
6     4058
2     3930
Name: count, dtype: int64
```

In [37]:
```
# day wise values counts

data.day.value_counts()
```

Out[37]:
```
day
11    1713
18    1692
8     1689
9     1684
25    1680
14    1680
4     1675
17    1670
27    1668
26    1661
19    1644
3     1640
2     1639
12    1639
29    1635
13    1632
22    1627
20    1625
16    1619
15    1615
1     1614
28    1614
7     1614
24    1600
6     1598
21    1597
10    1596
5     1575
23    1571
30    1554
31     940
Name: count, dtype: int64
```

In [38]:
```
sns.set_style("darkgrid")
sns.countplot(x = "year",data=data,palette = "Set3")
```

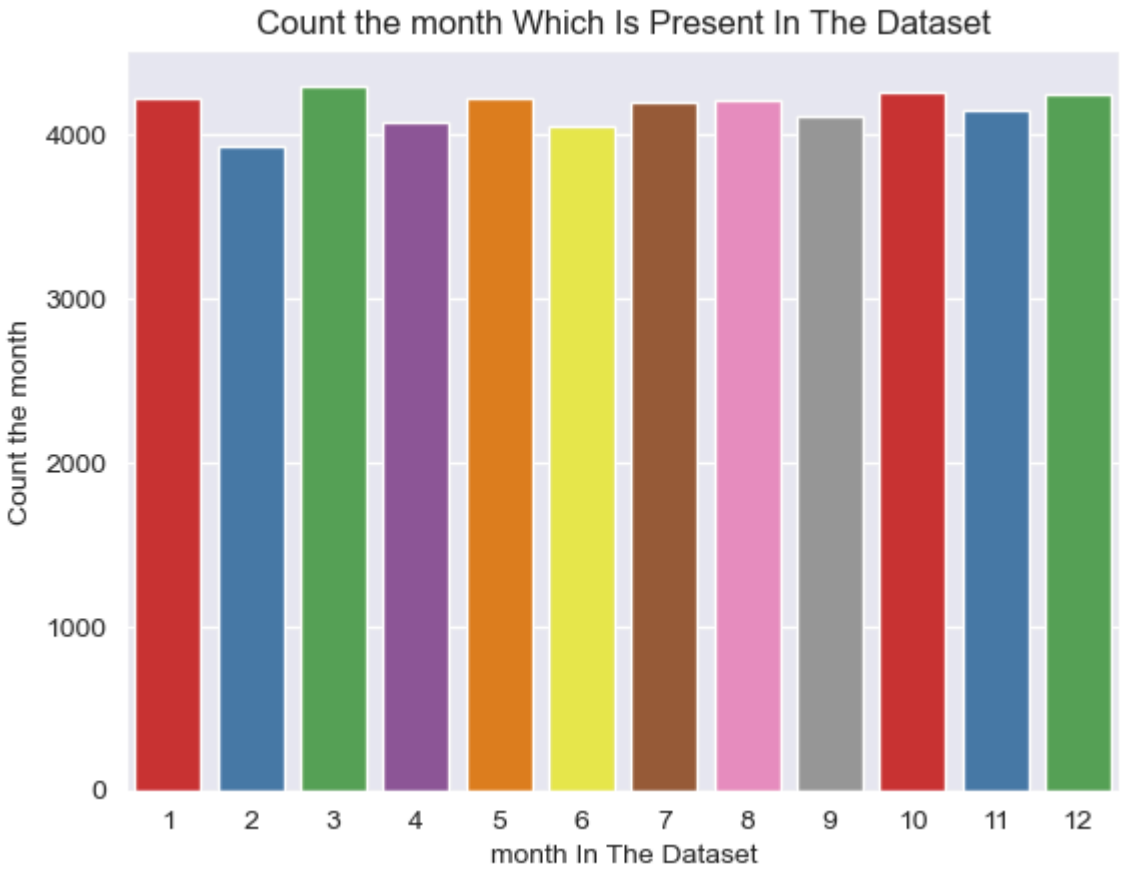
```
plt.title("Count the year Which Is Present In The Dataset")
plt.xlabel("year In The Dataset")
plt.ylabel("Count the year")
plt.show()
```



The highest purchases at Walmart were made in 2024, and the lower purchases in 2025 may be due to the ongoing year.

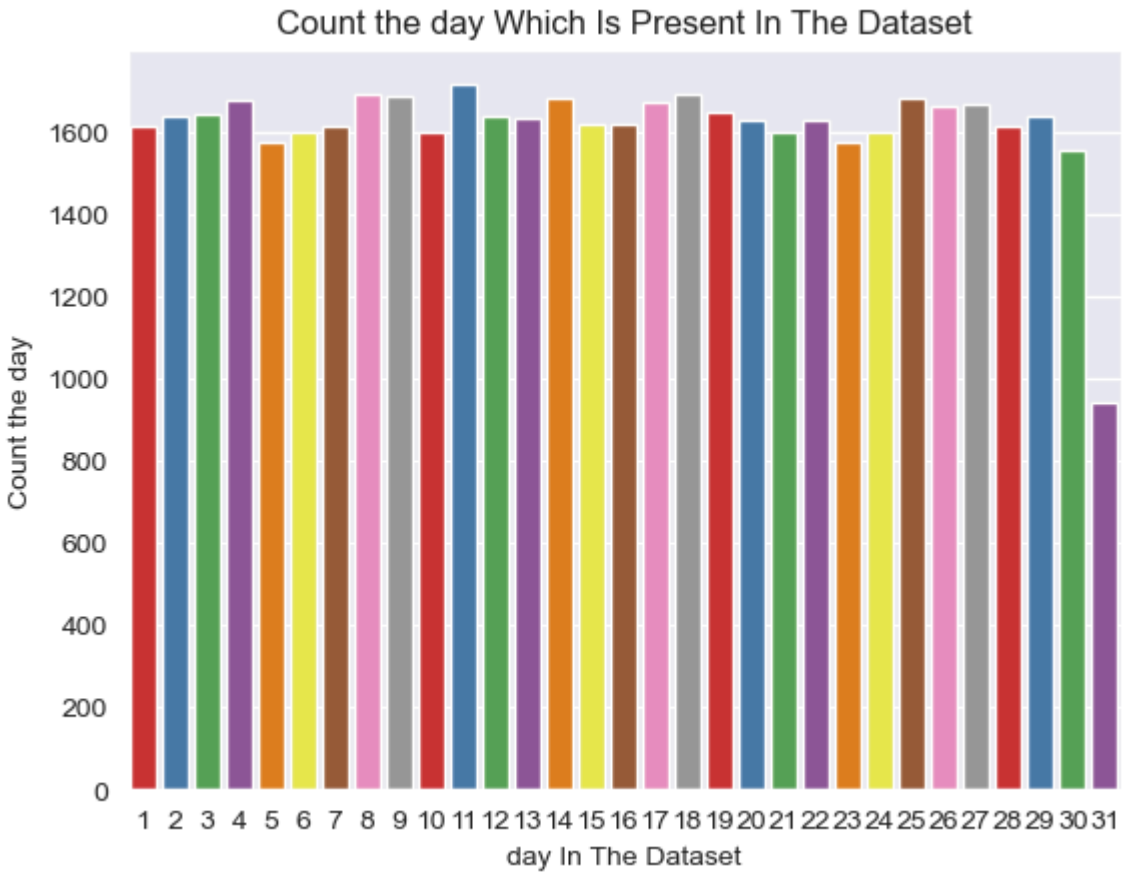
In [39]:
```
sns.set_style("darkgrid")
sns.countplot(x = "month",data=data,palette = "Set1")
plt.title("Count the month Which Is Present In The Dataset")
```

```
plt.xlabel("month In The Dataset")
plt.ylabel("Count the month")
plt.show()
```



Purchases at Walmart are relatively low in the month of February.

In [40]:
```
sns.set_style("darkgrid")
sns.countplot(x = "day",data=data,palette = "Set1")
plt.title("Count the day Which Is Present In The Dataset")
plt.xlabel("day In The Dataset")
plt.ylabel("Count the day")
plt.show()
```



Purchases are low on the 5th, 23rd, and 30th, but interestingly, the 31st shows the lowest purchases among all dates, while purchases on the remaining dates are higher.

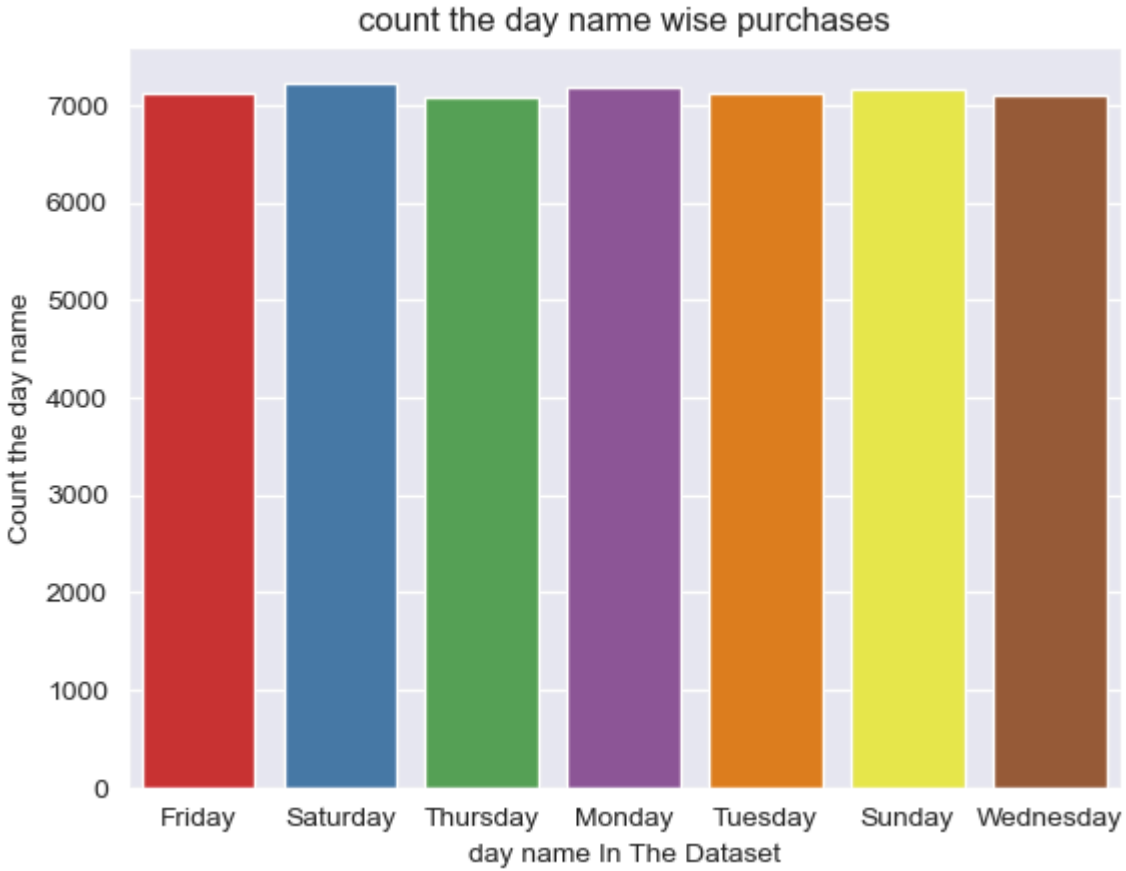
In [41]:
```
# Create new column for day name

data['Day_Name'] = data['Purchase_Date'].dt.day_name()
data['Day_Name']
```

Out[41]:
```
0           Friday
1         Saturday
2         Thursday
3           Monday
4           Monday
           ...
49995       Monday
49996      Tuesday
49997       Sunday
49998     Saturday
49999    Wednesday
Name: Day_Name, Length: 50000, dtype: object
```

In [42]:
```
# count the day

data.Day_Name.value_counts()
```

Out[42]:
```
Day_Name
Saturday     7226
Monday       7192
Sunday       7166
Friday       7120
Tuesday      7117
Wednesday    7100
Thursday     7079
Name: count, dtype: int64
```

In [43]:
```
sns.set_style("darkgrid")
sns.countplot(x = "Day_Name",data=data,palette = "Set1")
plt.title("count the day name wise purchases")
plt.xlabel("day name In The Dataset")
plt.ylabel("Count the day name")
plt.show()
```

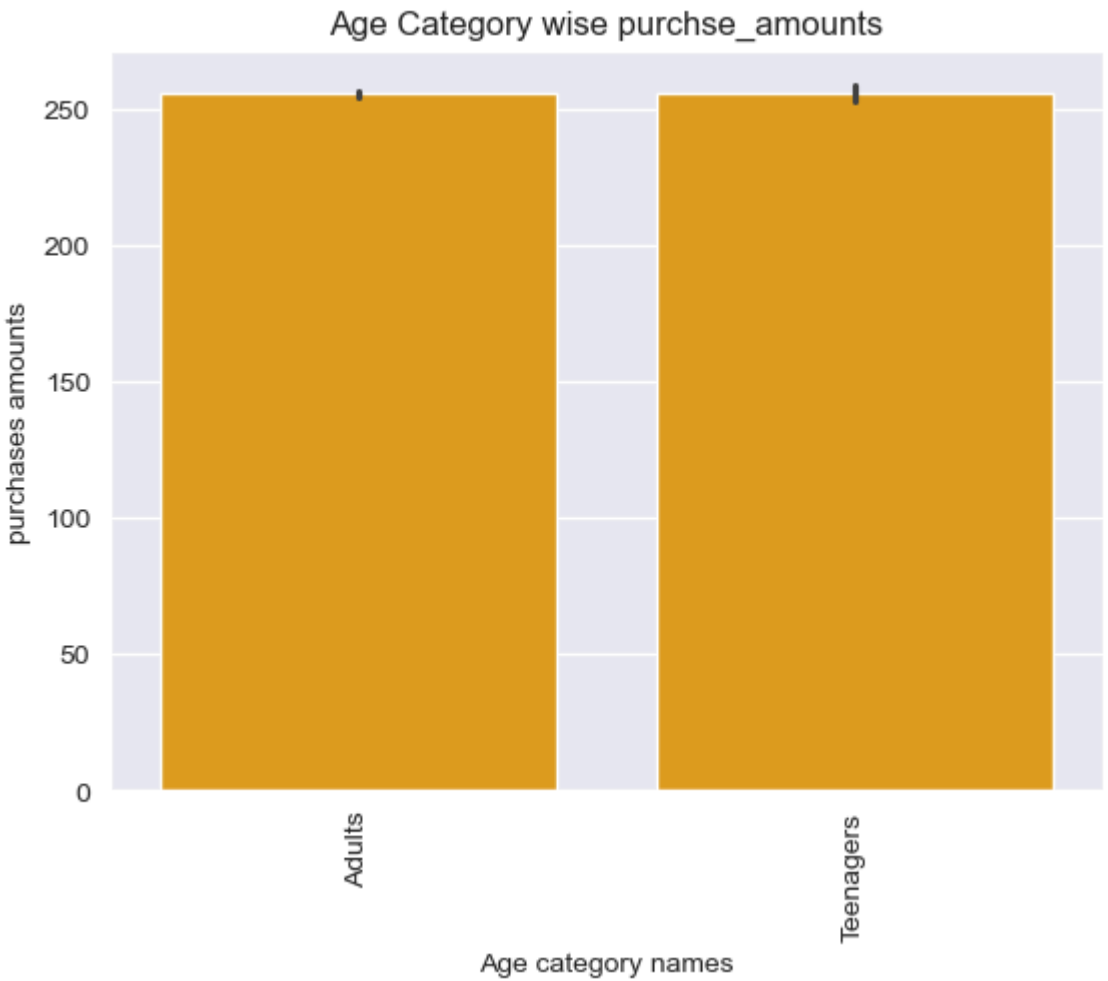


Saturday sees the highest number of customer purchases, while Thursday has the lowest purchases at Walmart.

Bi-Variate-Analysis

In [44]:
```
sns.barplot(x = "Age_cat",y = 'Purchase_Amount_Real',data=data,color = "orange")
plt.title("Age Category wise purchse_amounts")
plt.xlabel("Age category names")
plt.ylabel("purchases amounts")
```

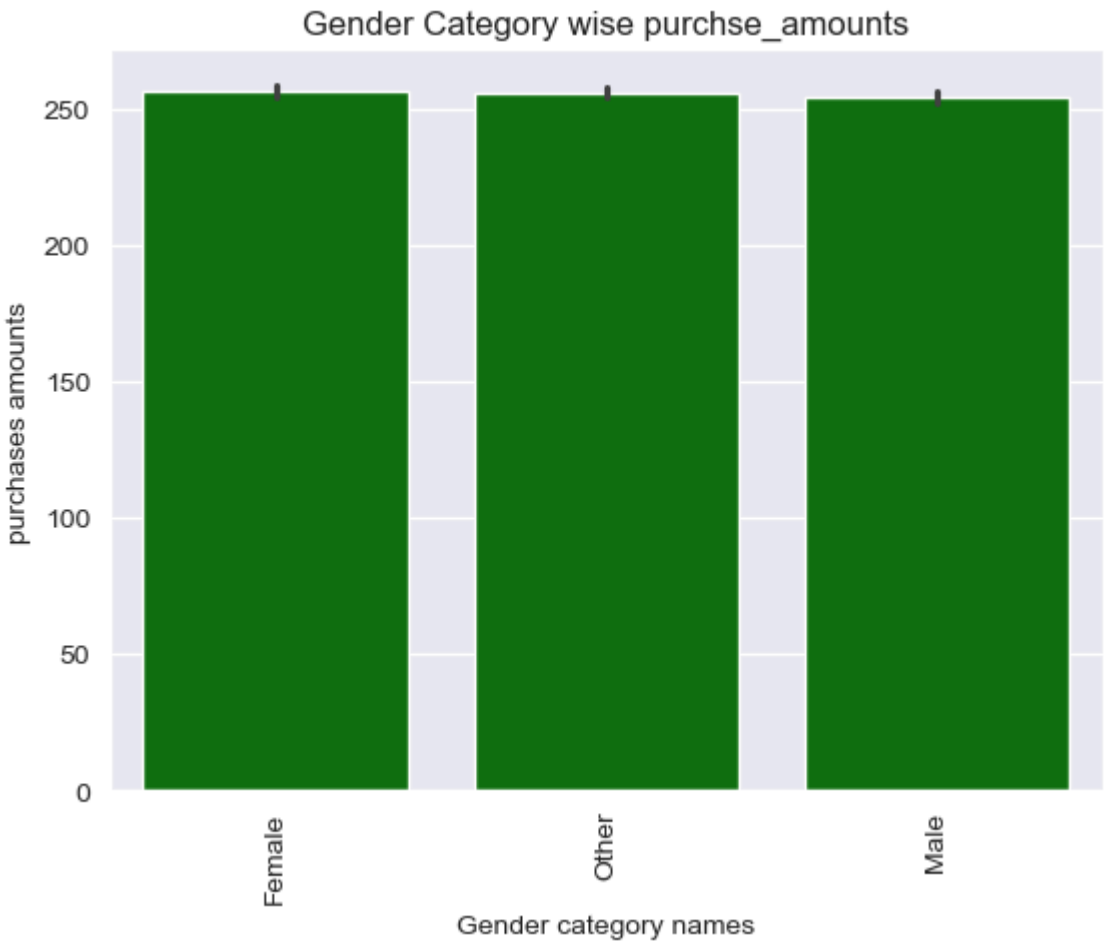
```
plt.xticks(rotation = "vertical")
plt.show()
```



Most adults and teenagers purchase products below the ₹500 amount.

In [45]:
```
sns.barplot(x = "Gender",y = "Purchase_Amount_Real",data=data,color = "green")
plt.title("Gender Category wise purchse_amounts")
plt.xlabel("Gender category names")
plt.ylabel("purchases amounts")
```

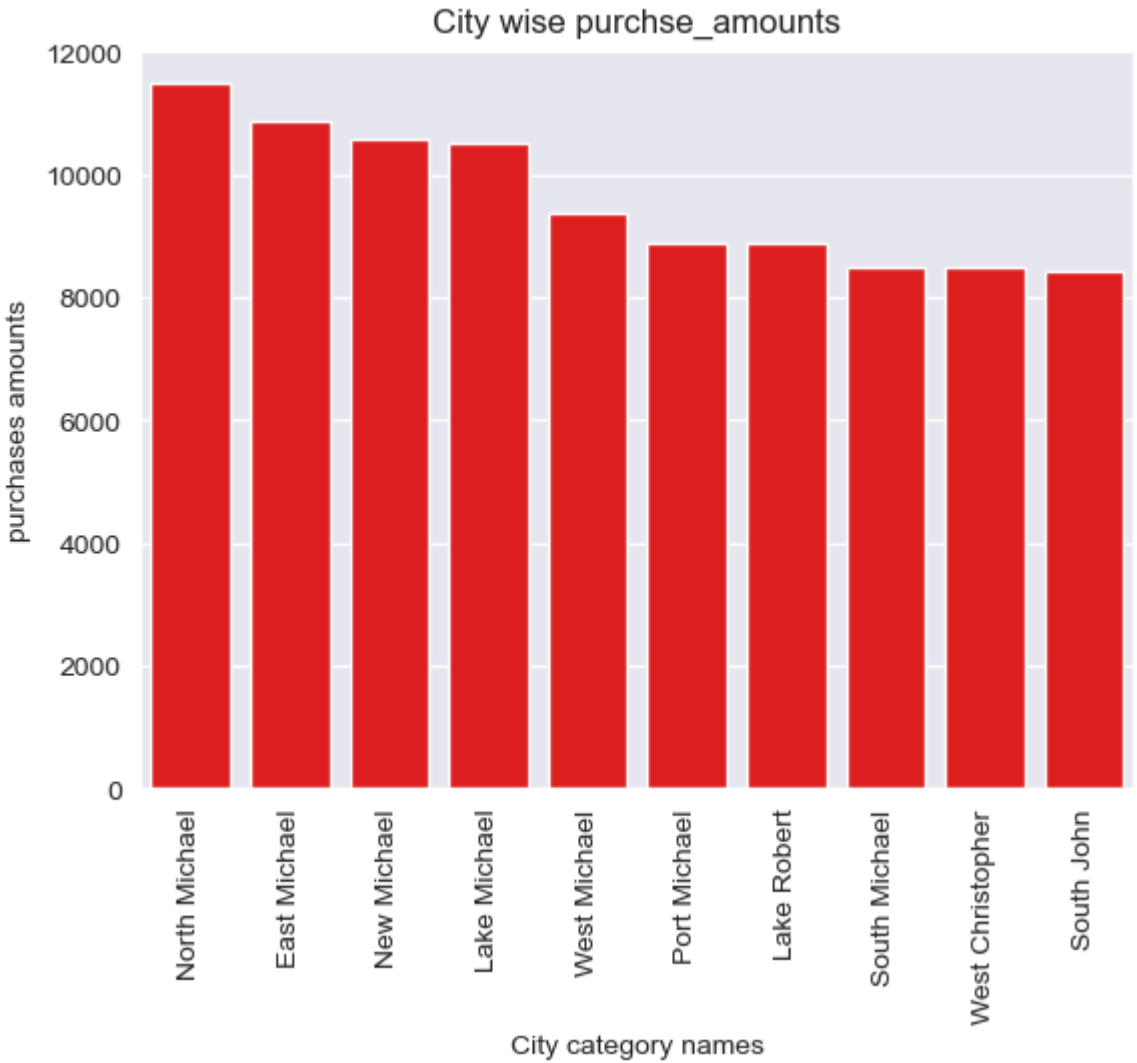
```
plt.xticks(rotation = "vertical")
plt.show()
```



Customers across almost all gender categories make purchases below the ₹500 price range at Walmart.

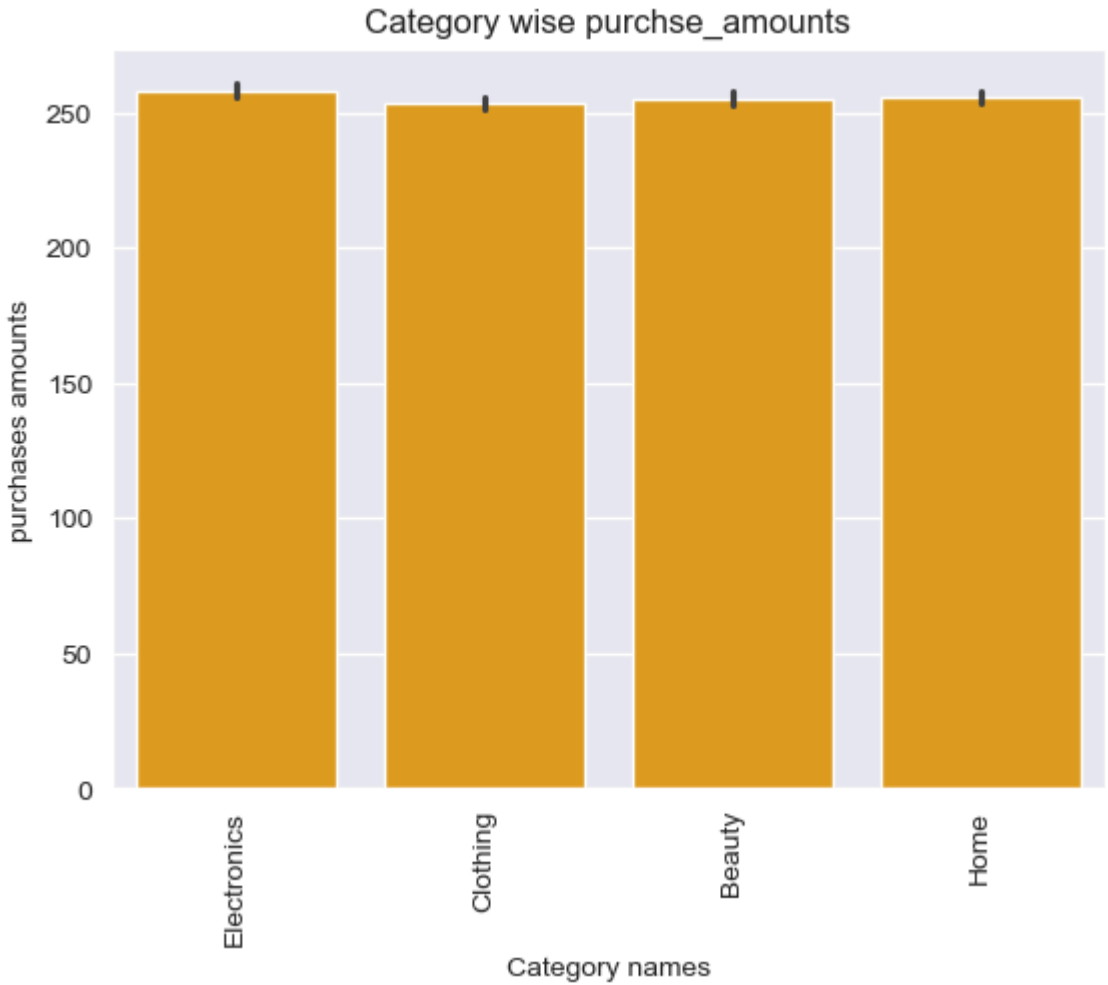
In [46]:
```
top_cities = data.groupby('City')['Purchase_Amount_Real'].sum().nlargest(10).reset_index()

sns.barplot(x = "City",y = 'Purchase_Amount_Real',data=top_cities,color = "red")
```

```
plt.title("City wise purchse_amounts")
plt.xlabel("City category names")
plt.ylabel("purchases amounts")
plt.xticks(rotation = "vertical")
plt.show()
```



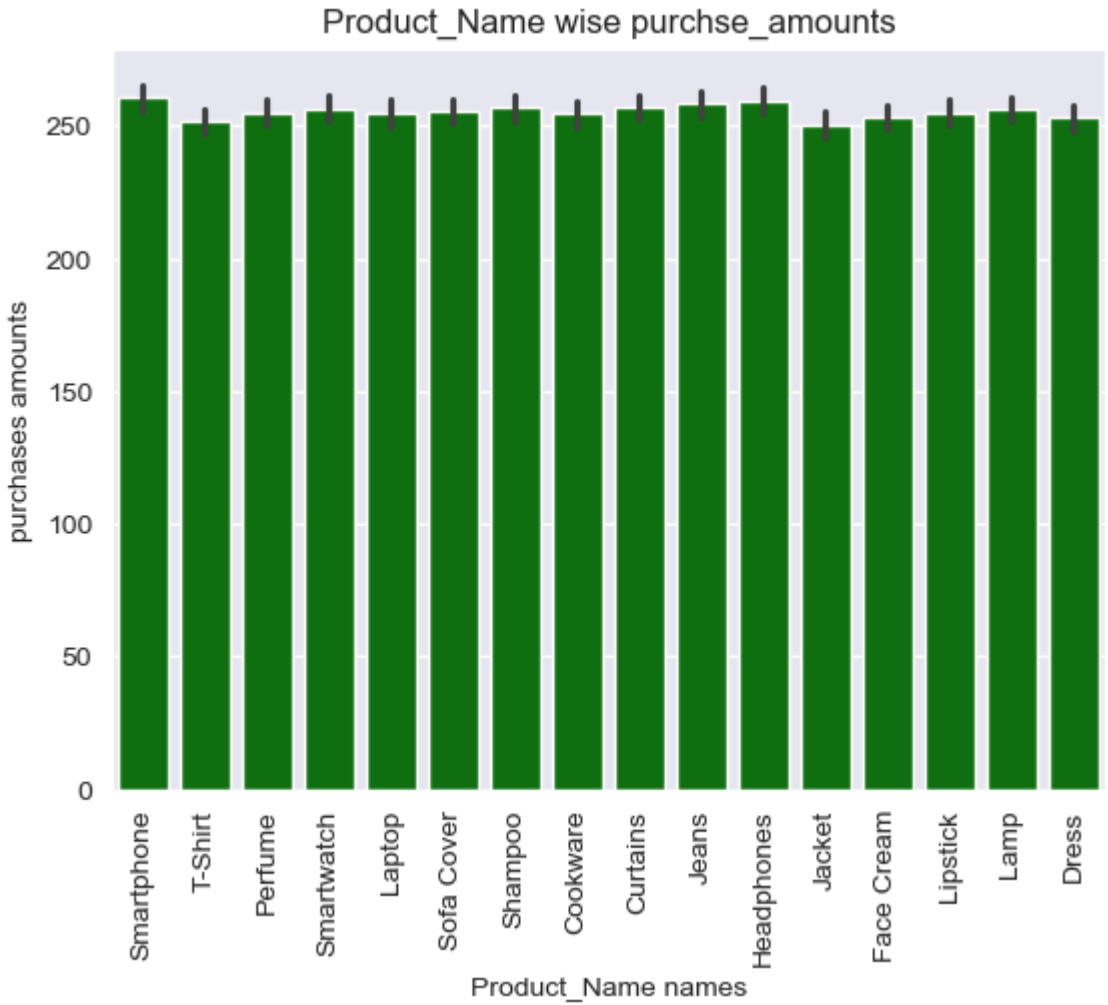
The highest number of customers at Walmart are from North Michael City.

In [47]:
```
sns.barplot(x = "Category",y = 'Purchase_Amount_Real',data=data,color = "orange")
plt.title("Category wise purchse_amounts")
plt.xlabel("Category names")
plt.ylabel("purchases amounts")
plt.xticks(rotation = "vertical")
plt.show()
```



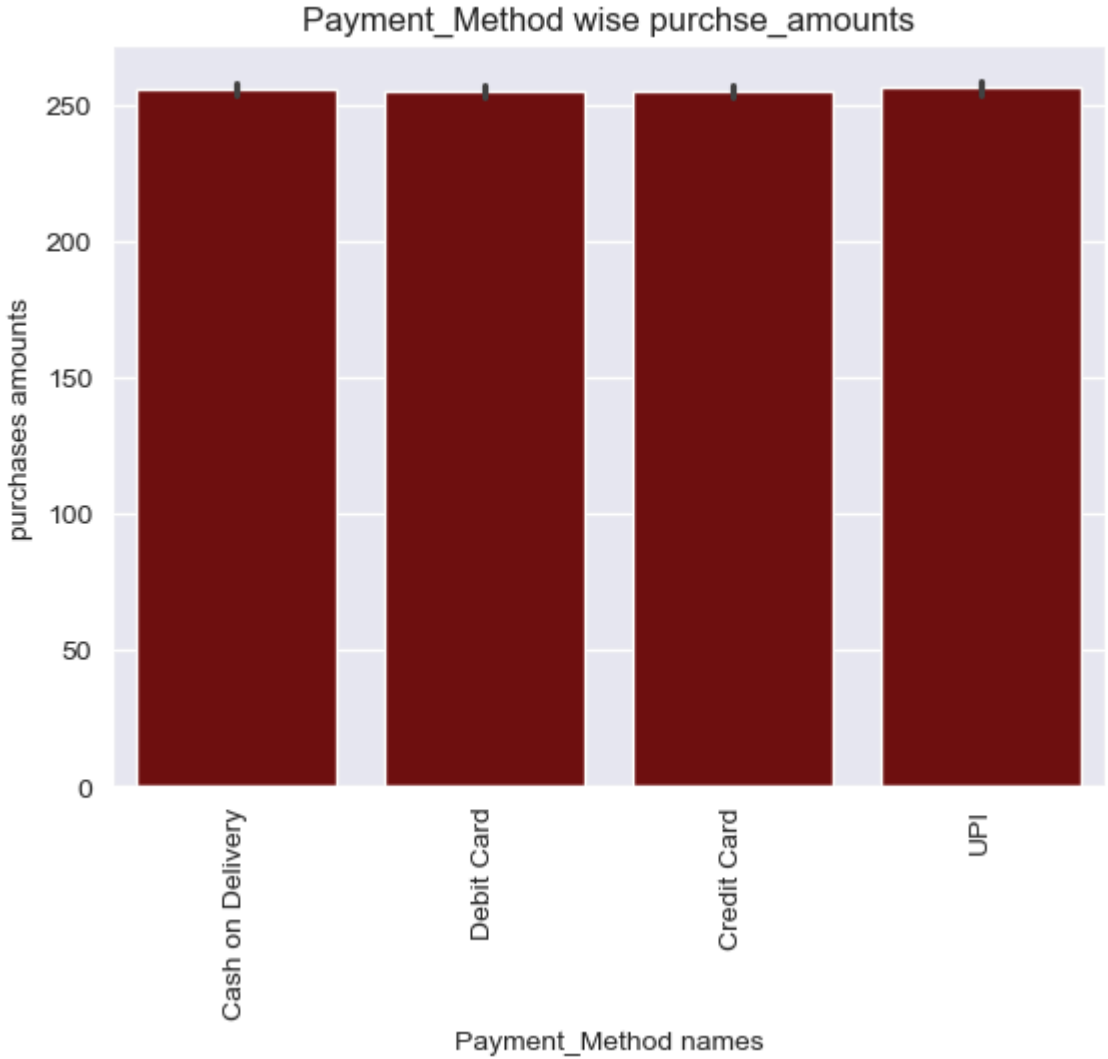
The Electronics category is the most popular among Walmart customers.

In [48]:
```
sns.barplot(x = "Product_Name",y = 'Purchase_Amount_Real',data=data,color = "green")
plt.title("Product_Name wise purchse_amounts")
plt.xlabel("Product_Name names")
plt.ylabel("purchases amounts")
plt.xticks(rotation = "vertical")
plt.show()
```



Smartphones are the highest-selling products at Walmart.

In [49]:
```
sns.barplot(x = "Payment_Method",y = 'Purchase_Amount_Real',data=data,color = "maroon")
plt.title("Payment_Method wise purchse_amounts")
plt.xlabel("Payment_Method names")
plt.ylabel("purchases amounts")
plt.xticks(rotation = "vertical")
plt.show()
```



In [50]:
```
data.Payment_Method.value_counts()
```

Out[50]:
```
Payment_Method
Debit Card          12589
Credit Card         12528
Cash on Delivery    12496
UPI                 12387
Name: count, dtype: int64
```

Customers make purchases using Debit Cards as their preferred payment method at Walmart.

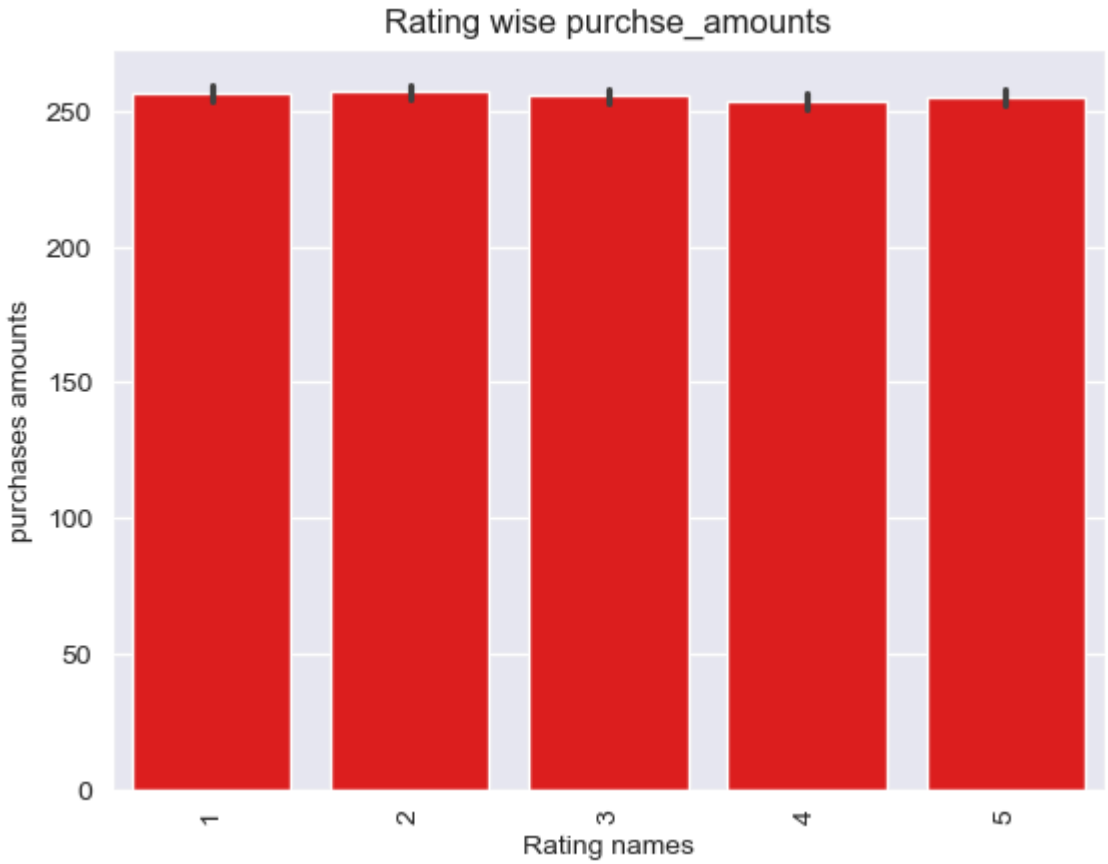
In [51]:
```
sns.barplot(x = "Discount_Applied",y = 'Purchase_Amount_Real',data=data,color = "yellow")
plt.title("Discount_Applied wise purchse_amounts")
plt.xlabel("Discount_Applied names")
plt.ylabel("purchases amounts")
plt.xticks(rotation = "vertical")
plt.show()
```

In [52]:
```
data.Discount_Applied.value_counts()
```

Out[52]:
```
Discount_Applied
No     25018
Yes    24982
Name: count, dtype: int64
```

Interestingly, regardless of whether discounts are applied, the majority of customers still make purchases at Walmart even when no discounts are offered.

In [53]:
```
sns.barplot(x = "Rating",y = 'Purchase_Amount_Real',data=data,color = "red")
plt.title("Rating wise purchse_amounts")
plt.xlabel("Rating names")
plt.ylabel("purchases amounts")
plt.xticks(rotation = "vertical")
plt.show()
```

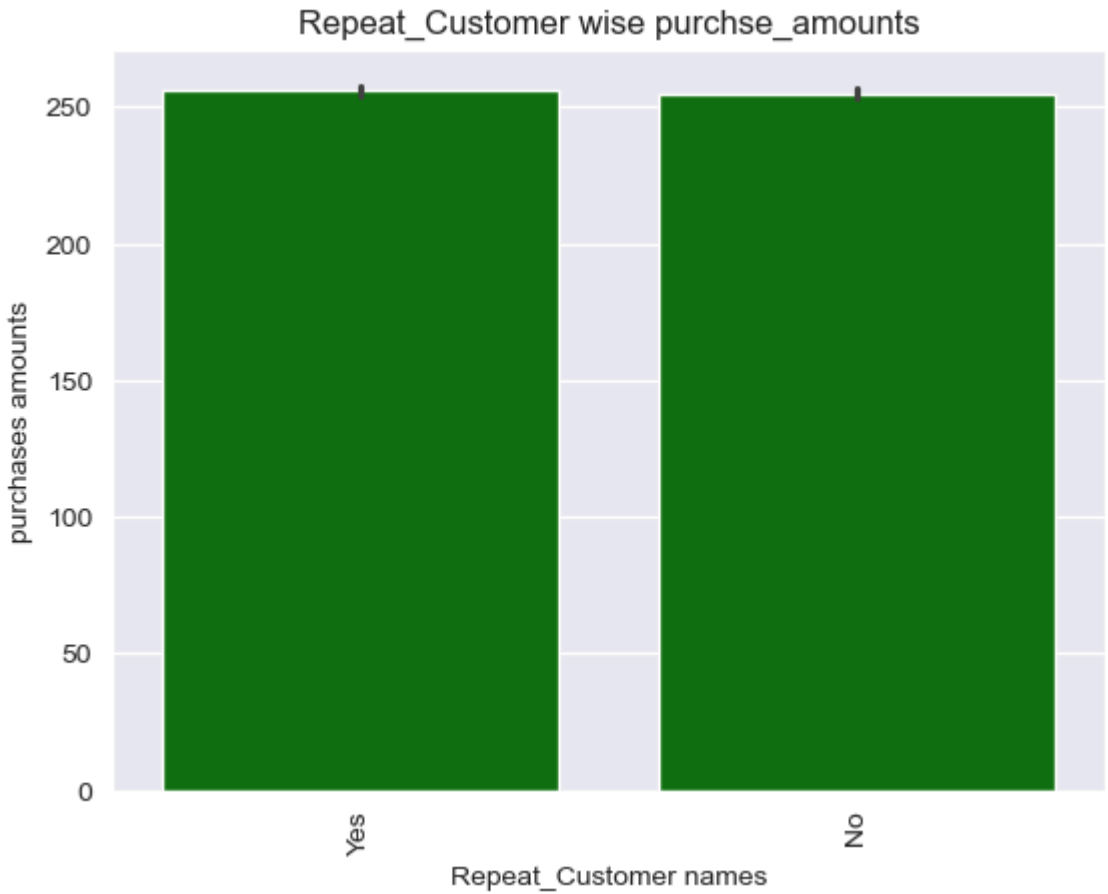


In [54]:
```
data.Rating.value_counts()
```

Out[54]:
```
Rating
1    10099
4    10045
5    10029
2     9971
3     9856
Name: count, dtype: int64
```

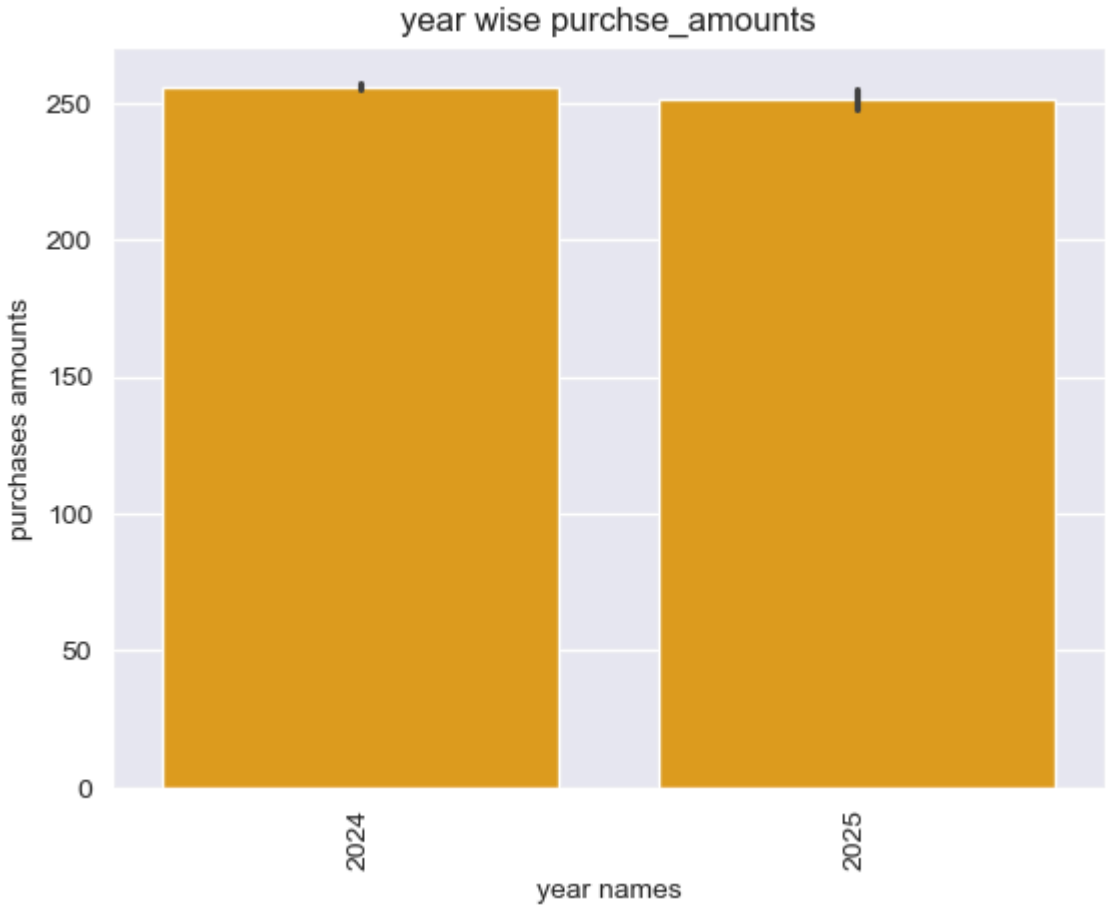
The majority of customers give 1-star ratings on the Walmart platform.

In [55]:
```
sns.barplot(x = "Repeat_Customer",y = 'Purchase_Amount_Real',data=data,color = "green")
plt.title("Repeat_Customer wise purchse_amounts")
plt.xlabel("Repeat_Customer names")
plt.ylabel("purchases amounts")
plt.xticks(rotation = "vertical")
plt.show()
```



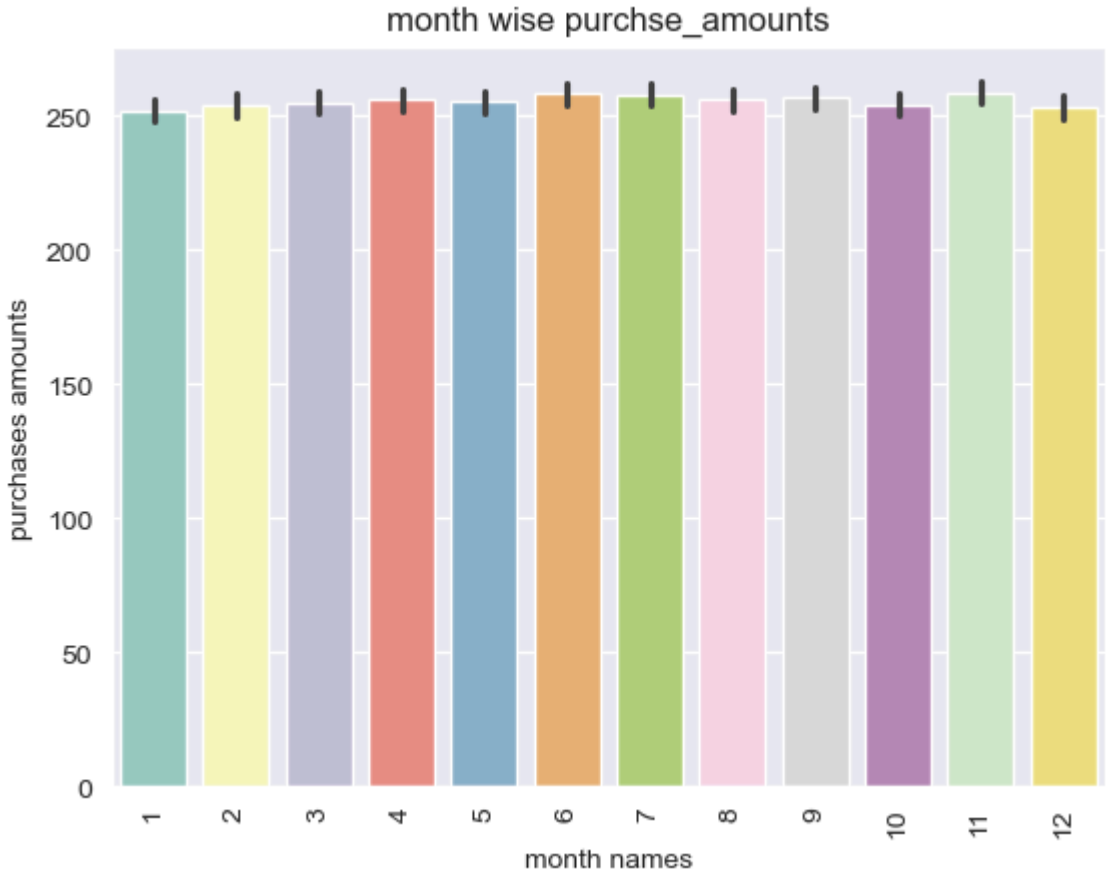
The majority of customers at Walmart are repeat buyers.

In [56]:
```
sns.barplot(x = "year",y = 'Purchase_Amount_Real',data=data,color = "orange")
plt.title("year wise purchse_amounts")
plt.xlabel("year names")
plt.ylabel("purchases amounts")
plt.xticks(rotation = "vertical")
plt.show()
```



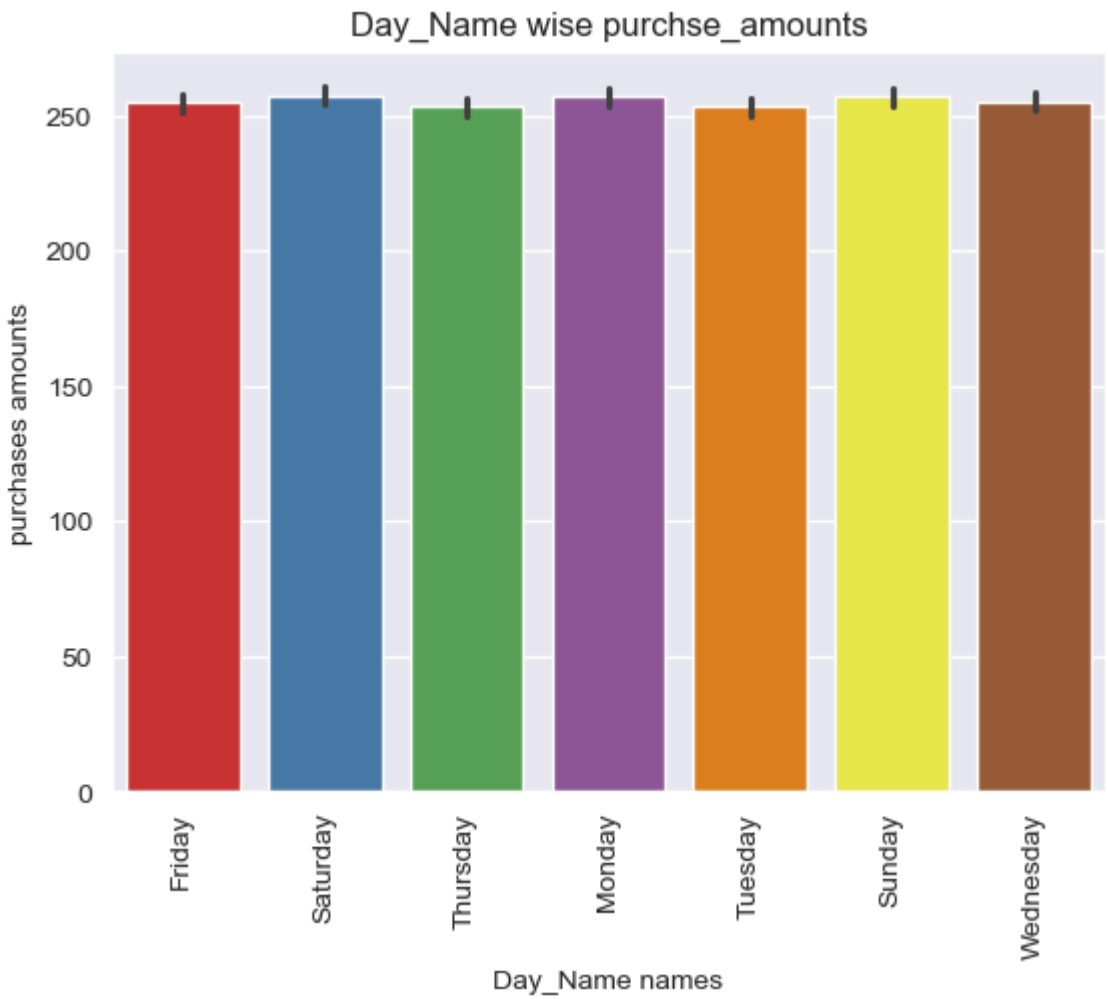
The highest purchases occurred in 2024. Since 2025 is ongoing, we can't predict its final figures yet, but 2025 is expected to surpass previous years in sales.

In [57]:
```
sns.barplot(x = "month",y = 'Purchase_Amount_Real',data=data,palette = "Set3")
plt.title("month wise purchse_amounts")
plt.xlabel("month names")
plt.ylabel("purchases amounts")
plt.xticks(rotation = "vertical")
plt.show()
```



November records the highest sales at Walmart.

In [58]:
```
sns.barplot(x = "Day_Name",y = 'Purchase_Amount_Real',data=data,palette = "Set1")
plt.title("Day_Name wise purchse_amounts")
plt.xlabel("Day_Name names")
plt.ylabel("purchases amounts")
plt.xticks(rotation = "vertical")
plt.show()
```



In [59]:
```
data.Day_Name.value_counts()
```

Out[59]:
```
Day_Name
Saturday     7226
Monday       7192
Sunday       7166
Friday       7120
Tuesday      7117
Wednesday    7100
Thursday     7079
Name: count, dtype: int64
```

Customers make the most purchases on Saturdays at Walmart.

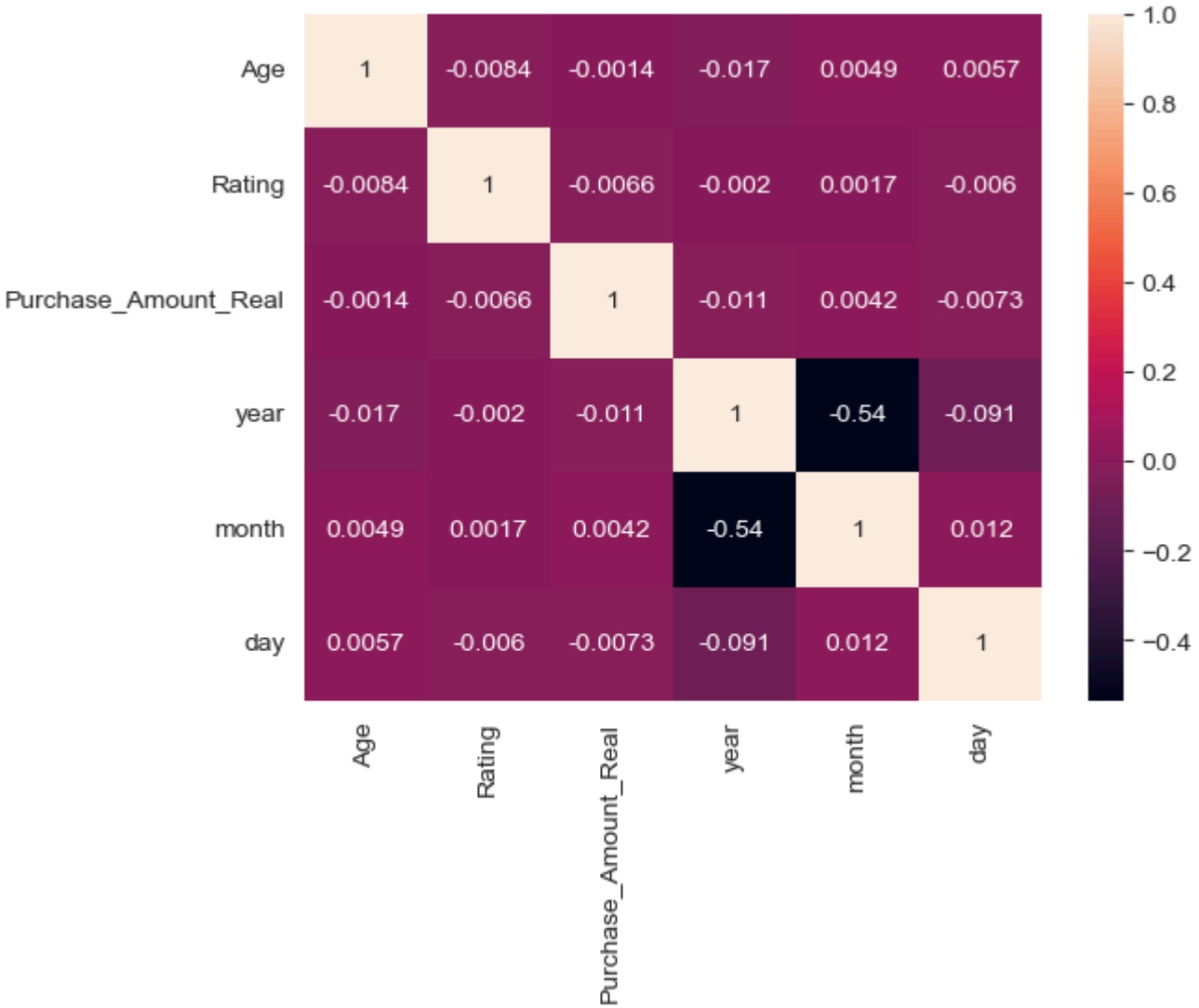
Multi-Variate-Analysis

In [60]:
```
walmart_corr = data.corr(numeric_only = True)
walmart_corr
```

Out[60]:

	Age	Rating	Purchase_Amount_Real	year	month	day
Age	1.000000	-0.008402	-0.001436	-0.017098	0.004871	0.005655
Rating	-0.008402	1.000000	-0.006648	-0.001979	0.001651	-0.005980
Purchase_Amount_Real	-0.001436	-0.006648	1.000000	-0.010784	0.004163	-0.007255
year	-0.017098	-0.001979	-0.010784	1.000000	-0.536865	-0.090545
month	0.004871	0.001651	0.004163	-0.536865	1.000000	0.012054
day	0.005655	-0.005980	-0.007255	-0.090545	0.012054	1.000000

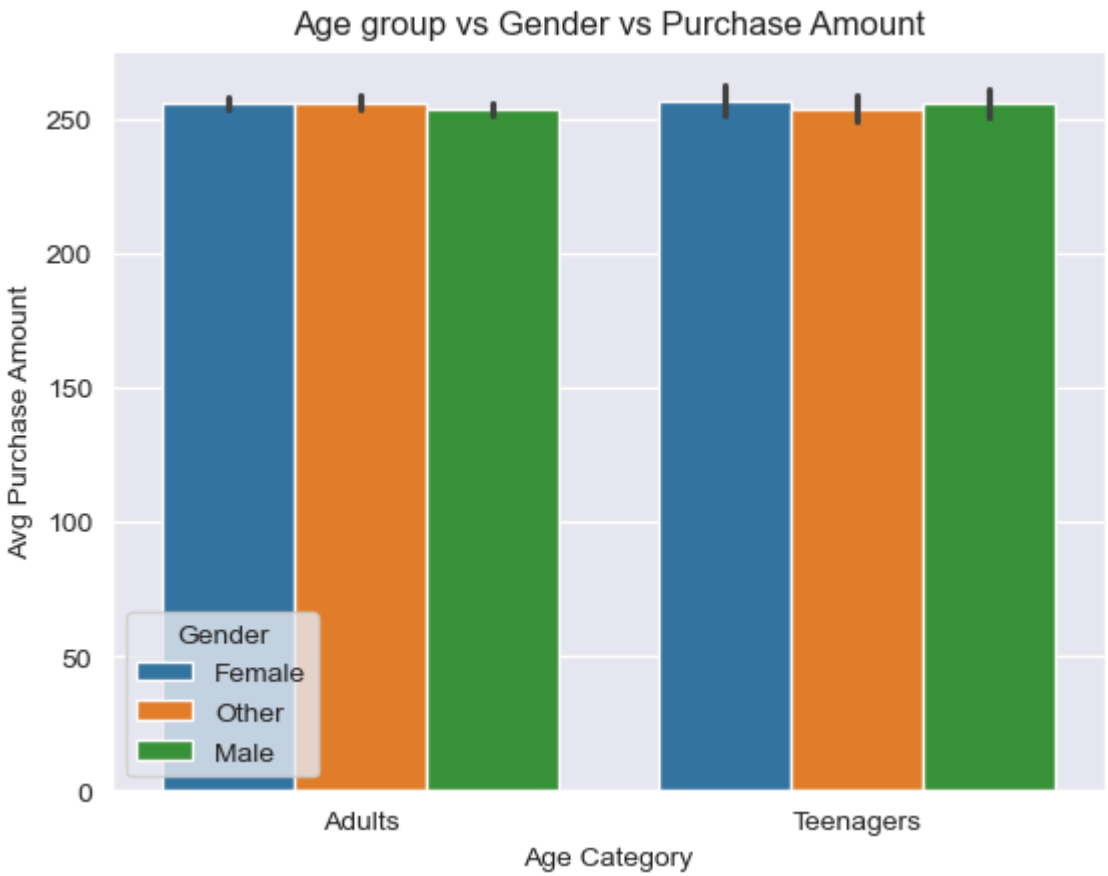
In [61]:
```
sns.heatmap(walmart_corr,annot = True)
plt.show()
```



There is no strong relationship between the variables.

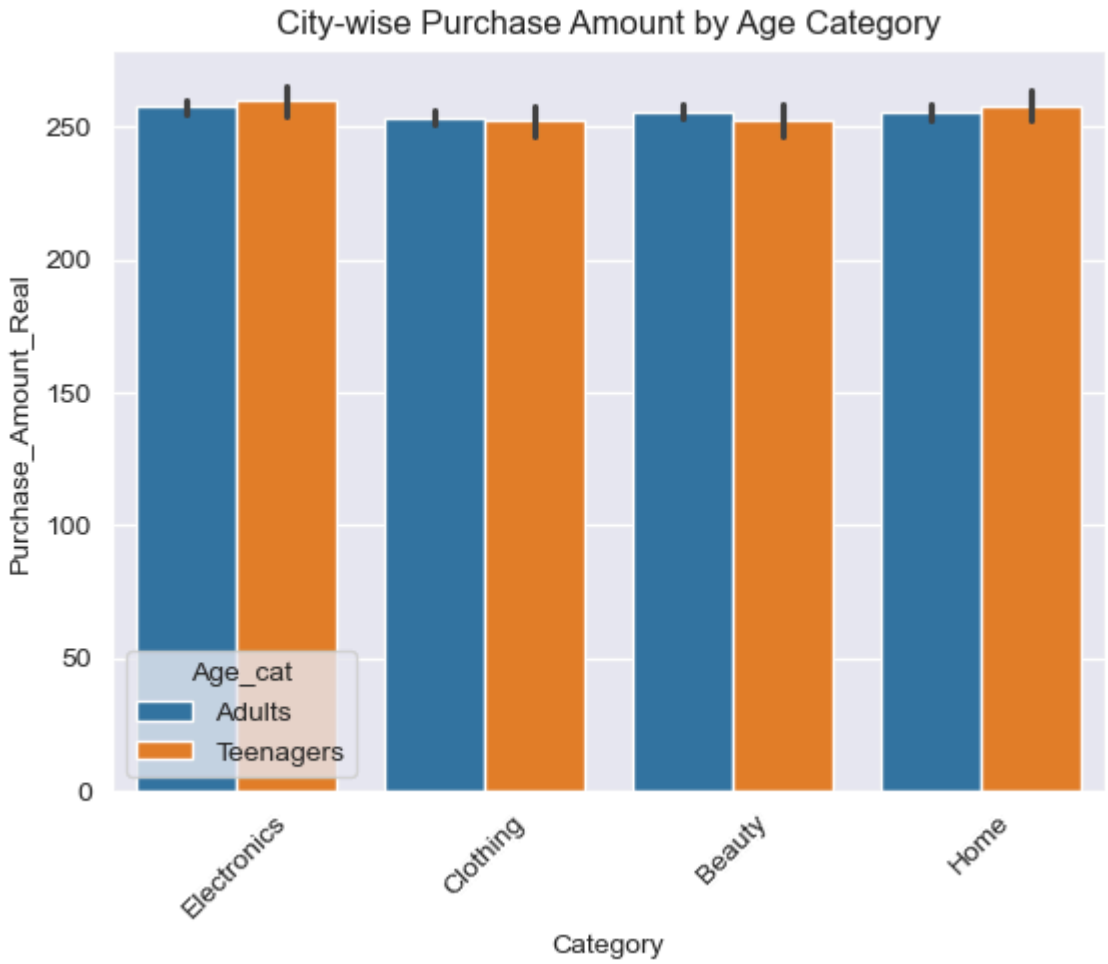
In [62]:
```
sns.barplot(x='Age_cat', y='Purchase_Amount_Real', hue='Gender', data=data)
plt.title("Age group vs Gender vs Purchase Amount")
plt.ylabel("Avg Purchase Amount")
```

```
plt.xlabel("Age Category")
plt.show()
```



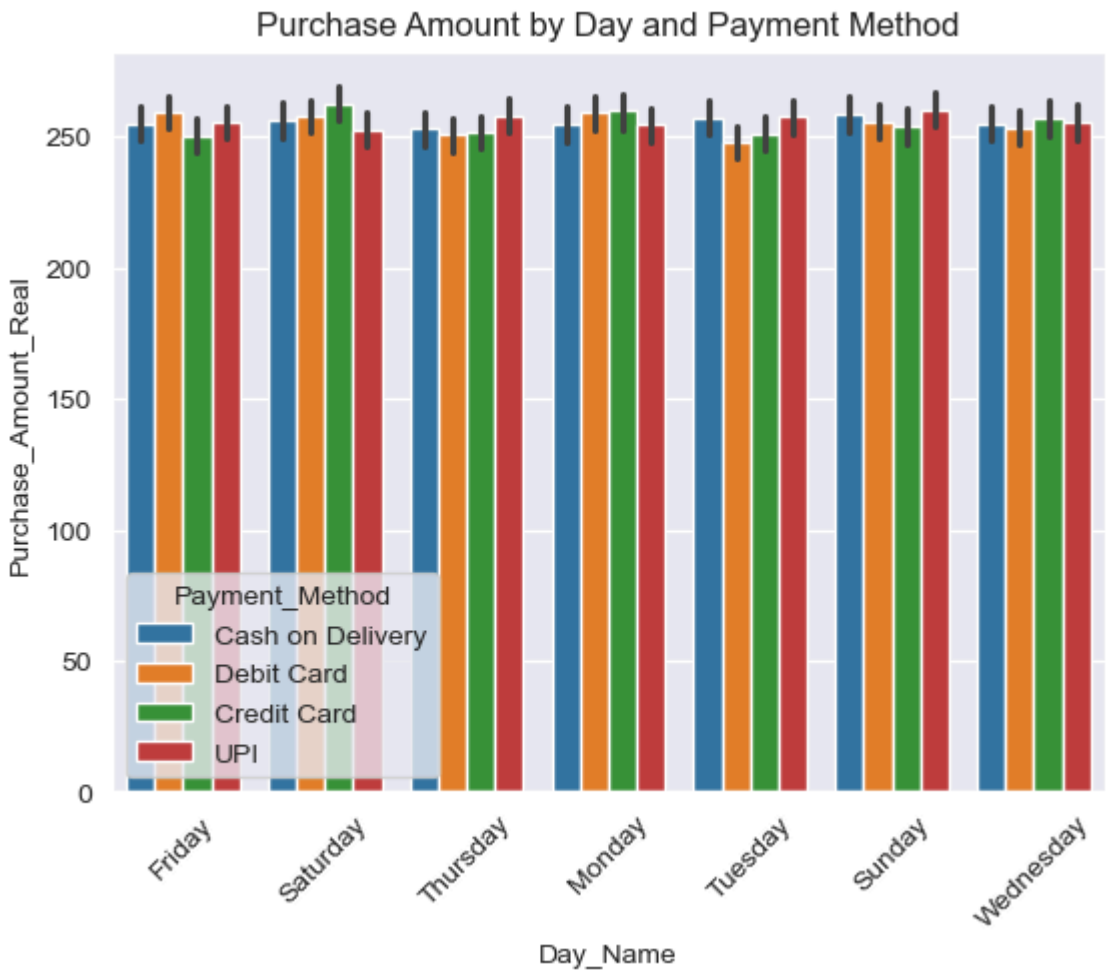
The "Other" gender category has a high ratio among adults, and they make more purchases from Walmart.

In [63]:
```
sns.barplot(x='Category', y='Purchase_Amount_Real', hue='Age_cat', data=data )
plt.title("City-wise Purchase Amount by Age Category")
plt.xticks(rotation=45)
plt.show()
```



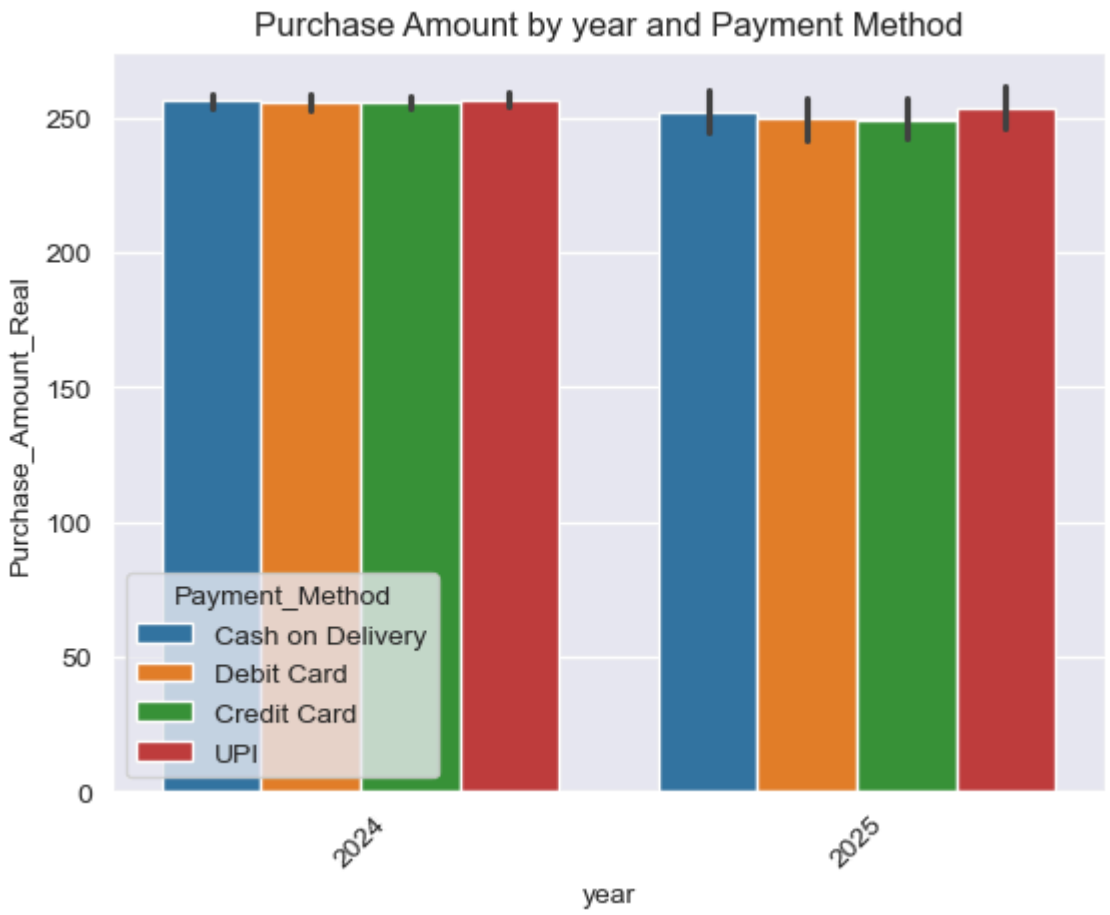
Teenagers make more purchases in the Electronics category at Walmart.

In [64]:
```
sns.barplot(x='Day_Name', y='Purchase_Amount_Real', hue='Payment_Method', data=data)
plt.title("Purchase Amount by Day and Payment Method")
plt.xticks(rotation=45)
plt.show()
```



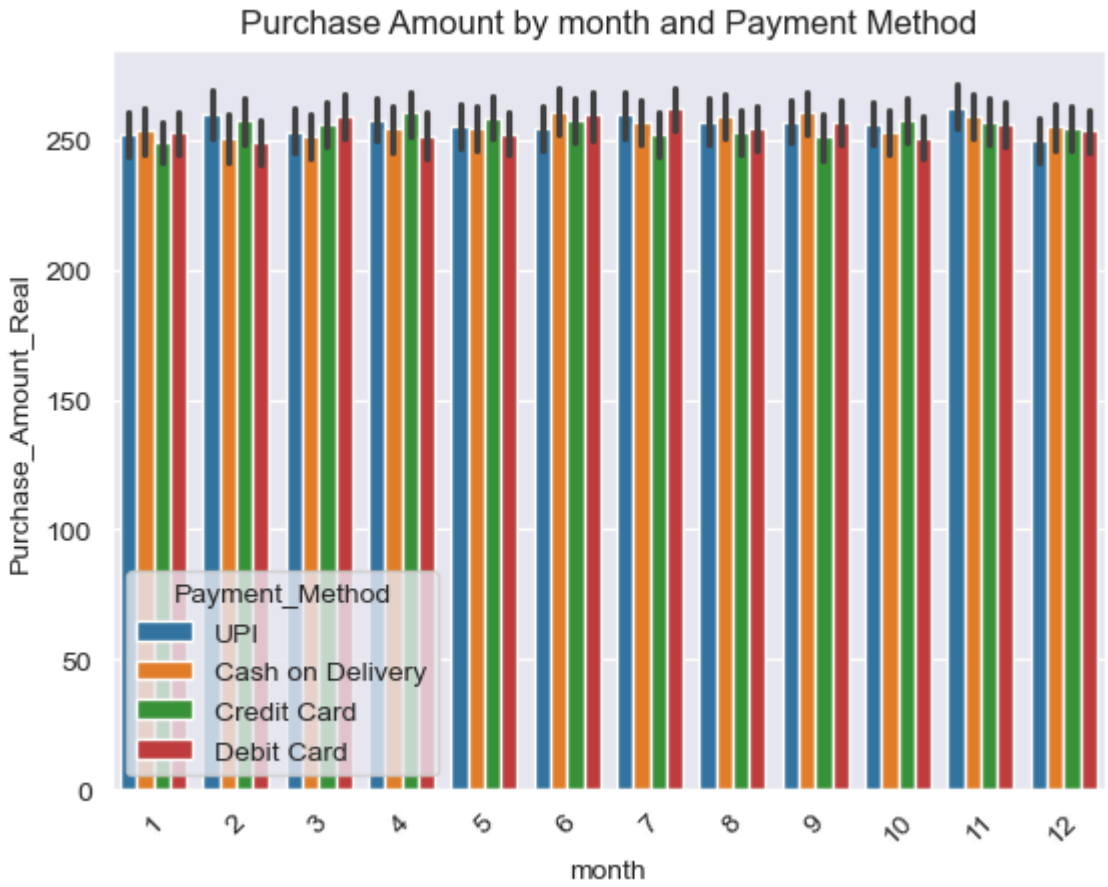
Customers make higher purchases via Credit Card on Saturdays at Walmart.

In [65]:
```
sns.barplot(x='year', y='Purchase_Amount_Real', hue='Payment_Method', data=data)
plt.title("Purchase Amount by year and Payment Method")
plt.xticks(rotation=45)
plt.show()
```



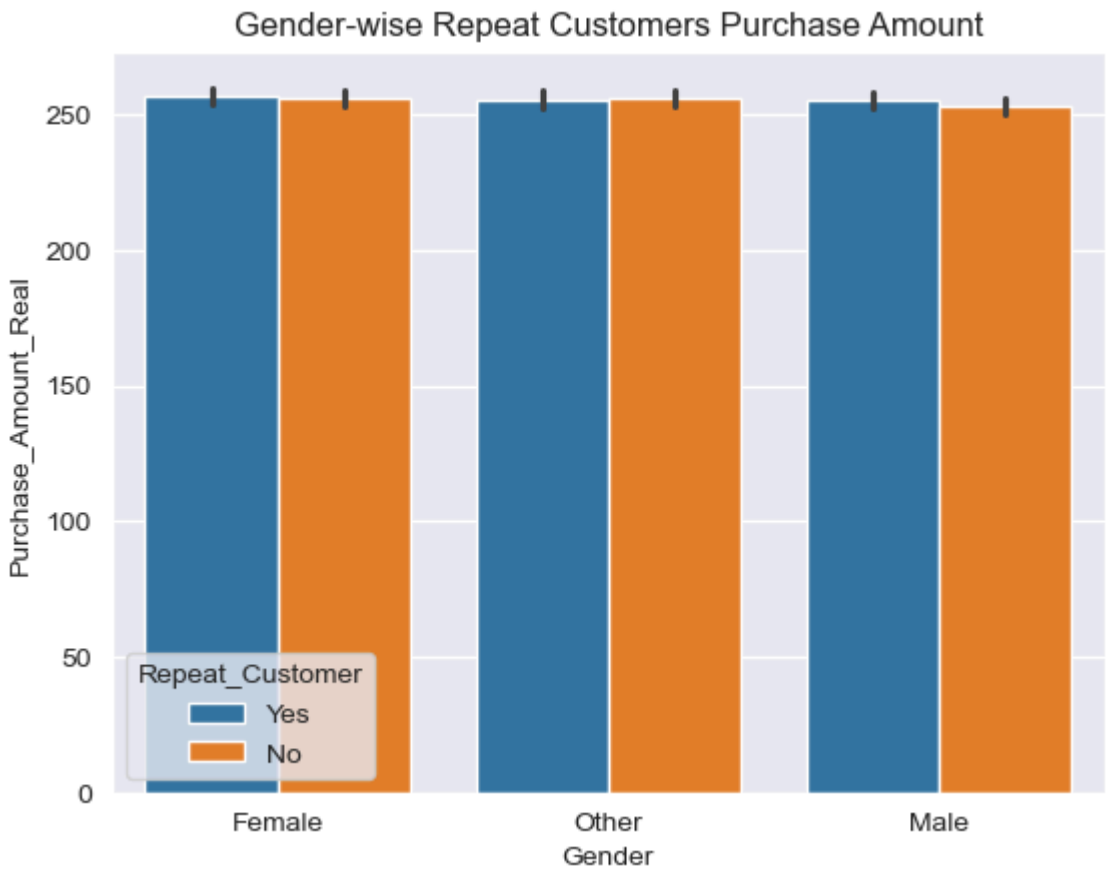
In 2025, the ratio of customers using the UPI payment method is very high.

In [66]:
```
sns.barplot(x='month', y='Purchase_Amount_Real', hue='Payment_Method', data=data)
plt.title("Purchase Amount by month and Payment Method")
plt.xticks(rotation=45)
plt.show()
```



Interestingly, in the month of November, customers preferred to pay for their purchases using UPI.

In [67]:
```
sns.barplot(x='Gender', y='Purchase_Amount_Real', hue='Repeat_Customer', data=data)
plt.title("Gender-wise Repeat Customers Purchase Amount")
plt.show()
```



Once again, we observe that repeat customers are higher in number and are more likely to make purchases at Walmart.

✅ Walmart EDA Project – Key Recommendations:-

1.Target Adults & "Other" Gender – Highest purchase group, run personalized offers.

2.Boost Electronics & Smartphone Sales – Expand variety, offer combos (e.g., phone + headphone).

3.Focus on Teenagers for Electronics – Use social media + student discounts.

4.Promote Debit Cards & UPI – Offer cashback; UPI highest in 2025 & November.

5.Saturday = Sales Day – Plan major campaigns and offers on Saturdays.

6.Support Repeat Customers – Launch a loyalty rewards program.

7.Improve Ratings – Investigate 1-star feedback, upgrade service to turn 4★ into 5★.

8.Increase Purchase Amount – Push combo deals to cross ₹500 mark.

9.Fix Low Sales in Feb & 31st – Run end-of-month or Valentine's offers.