📚 Bookstore Sales Insights: Advanced SQL Analysis & Data Exploration

📌 Problem Statement

- In today's competitive retail environment, understanding sales performance, customer behavior, inventory efficiency, and hidden data patterns is crucial for business growth.
- Our fictional Bookstore has been experiencing inconsistent revenue, unexplained stockouts, and uncertain customer retention trends.
- Despite collecting amounts of data across orders, customers, books, and inventory, decision-makers struggle to turn this data into actionable insights.

Project Objective

- This project aims to analyze the bookstore's data through SQL and Python to extract more 30 high-impact business insights that support data-driven decision-making. The analysis is divided into four key areas:

- Sales Performance – Understand revenue trends, top-selling products, and growth patterns.

- Customer Behavior – Segment customers by spending and loyalty to improve retention.

- Inventory Management – Optimize stock by identifying overstocked, understocked, and dead items.

- Deep Data Insights – Detect outliers, trends, and anomalies using advanced EDA techniques.

🛠️ Tools Used

- MySQL – to write and run SQL queries for data exploration
- Seaborn – for optional visual representation of trends
- Matplotlib – for professional plotting (if needed)
- stattools
- Pandas
- Numpy

📂 Project Structure

```
bookstore-sql-insights/
│
├── README.md                    → Project documentation and overview
├── SQL_Queries/                 → SQL scripts for all analyses
│   ├── sales_revenue_analysis.sql
│   ├── customer_segment_analysis.sql
│   ├── genre_trend_analysis.sql
│   └── borrowing_analysis.sql
├── Data/                        → Contains raw CSV files
│   ├── books.csv                → 500 rows × 7 columns
│   ├── orders.csv               → 500 rows × 6 columns
│   └── customers.csv            → 500 rows × 6 columns
├── Reports/
└── Visuals/
```

Loading Data File and Building Connection

In [312…
```
!pip install mariadb
```

```
Requirement already satisfied: mariadb in c:\users\aarav computer\anaconda3\lib\site-packages (1.1.13)
Requirement already satisfied: packaging in c:\users\aarav computer\anaconda3\lib\site-packages (from mariadb) (24.2)
```

In [3]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import mysql.connector
import warnings
warnings.filterwarnings('ignore')
```

In [4]:
```
conn = mysql.connector.connect(
    host="localhost",
    user="root",
    password="shahista123@dataanalyst456",
    database="books",
```

```
    charset='utf8mb4',
    collation='utf8mb4_general_ci'
)
```

In [5]:
```
cursor=conn.cursor()
```

Exploaratory Data Analysis(EDA)

In [6]:
```
Books="select * from books"
Books_Data=pd.read_sql(Books,conn)
```

In [6]:
```
Books_Data.head(5)
```

Out[6]:

	Book_ID	Title	Author	Genre	Published_Year	Price	Stock	Books_Segment
0	1	Configurable modular throughput	Joseph Crane	Biography	1949	21.34	100	None
1	2	Persevering reciprocal knowledge user	Mario Moore	Fantasy	1971	35.80	19	None
2	3	Streamlined coherent initiative	Derrick Howard	Non-Fiction	1913	15.75	27	None
3	4	Customizable 24hour product	Christopher Andrews	Fiction	2020	43.52	8	None
4	5	Adaptive 5thgeneration encoding	Juan Miller	Fantasy	1956	10.95	16	None

In [7]:
```
Books_Data.tail(5)
```

Out[7]:

	Book_ID	Title	Author	Genre	Published_Year	Price	Stock	Books_Segment
495	496	Decentralized radical forecast	James Adams	Science Fiction	1966	43.75	96	None
496	497	Function-based local installation	Craig Thompson	Science Fiction	1919	24.10	33	None
497	498	Secured 24/7 neural-net	Heather Marks	Non-Fiction	1975	10.88	22	None
498	499	Compatible transitional budgetary management	Isaac Nelson	Biography	1905	6.94	64	None
499	500	Vision-oriented zero tolerance initiative	David Hatfield	Science Fiction	1921	15.94	64	None

In [8]:
```
Books_Data.isnull().sum()
```

Out[8]:
```
Book_ID            0
Title              0
Author             0
Genre              0
Published_Year     0
Price              0
Stock              0
Books_Segment    500
dtype: int64
```

In [9]:
```
Books_Data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 8 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Book_ID         500 non-null    int64  
 1   Title           500 non-null    object 
 2   Author          500 non-null    object 
 3   Genre           500 non-null    object 
 4   Published_Year  500 non-null    int64  
 5   Price           500 non-null    float64
 6   Stock           500 non-null    int64  
 7   Books_Segment   0 non-null      object 
dtypes: float64(1), int64(3), object(4)
memory usage: 31.4+ KB
```

In [10]:
```
Books_Data.describe()
```

Out[10]:

	Book_ID	Published_Year	Price	Stock
count	500.000000	500.000000	500.000000	500.000000
mean	250.500000	1959.918000	27.367440	50.112000
std	144.481833	36.271511	13.243433	29.920192
min	1.000000	1900.000000	5.070000	0.000000
25%	125.750000	1927.000000	15.390000	25.000000
50%	250.500000	1961.000000	27.300000	49.000000
75%	375.250000	1990.000000	38.510000	77.000000
max	500.000000	2023.000000	49.980000	100.000000

- So as here in the Book_Id column it is easy seen max Books are 500 hundred so total 500 books are there .
- The Maximum Published year (2023) is the latest year and the Minimum (1900) is our earliest year,that means Books data ranges between 1900 to 2023.
- The minimum Price of the books is (5) and maximum Price is (49.980000) that is near to 50. it shows there is a variability in the books prices and some books are high priced and some books are low priced.
- To get to know which segment books have performed well in this years will be the great insight for the business.
- On the other side there is also variability in the stocks where range is starts from 25 to till 100 ,the question arises do business have that much amount of demand of books as the way it has been stocked.

Code 1:Total Unique Books Available

In [11]:
```
TotalUniqueBooks="""
select count(Distinct Book_ID)as Book_Count
from Books """
pd.read_sql(TotalUniqueBooks,conn)
```

Out[11]:

	Book_Count
0	500

Code 2: List of Distinct Book Genres

In [12]:
```
DistinctGenres= "Select Distinct(Genre) from Books"
pd.read_sql(DistinctGenres,conn)
```

Out[12]:

	Genre
0	Biography
1	Fantasy
2	Non-Fiction
3	Fiction
4	Romance
5	Science Fiction
6	Mystery

Code 3 : Count Of Distinct Authors

In [13]:
```
DistinctAuthors="Select Count(Distinct Author)from Books"
pd.read_sql(DistinctAuthors,conn)
```

Out[13]:

	Count(Distinct Author)
0	493

Code 4: Average Quantity Ordered

In [291…
```
AverageQuantityOrdered="""
select avg(Quantity)as Avg_Quantiy_Ordered
from Orders"""
pd.read_sql(AverageQuantityOrdered,conn)
```

Out[291...

	Avg_Quantiy_Ordered
0	5.394

Code 5: Distinct Order Years

In [292...

```
Order_Years="""
select Distinct
(extract(year from Order_Date))as years from Orders
Order by years"""
pd.read_sql(Order_Years,conn)
```

Out[292...

	years
0	2022
1	2023
2	2024

Code 6: Total Sales Generated Per Year

In [295...

```
SalesPerYear="""
select year(Order_Date)as Year,sum(Total_Amount)as Total_Sales
from Orders
group by year(Order_Date)
"""
pd.read_sql(SalesPerYear,conn)
```

Out[295...

	Year	Total_Sales
0	2023	36339.97
1	2024	36775.33
2	2022	2513.36

Code 7: Total Quantity Ordered

In [296...

```
Total_Quantity_Ordered="""
select sum(Quantity)as Total_Quantity_ordered
from Orders"""
pd.read_sql(Total_Quantity_Ordered,conn)
```

Out[296...

	Total_Quantity_ordered
0	2697.0

Code 8: Unique Cities Count

In [44]:

```
City_Count="""select count(Distinct City)as Cities
from Customers"""
pd.read_sql(City_Count,conn)
```

Out[44]:

	Cities
0	489

Code 9: Average Book Price Calculation

In [298...

```
AvgBookPrice="""select avg(price)as Avg_BookPrice
from Books"""
pd.read_sql(AvgBookPrice,conn)
```

Out[298...

	Avg_BookPrice
0	27.36744

Code 10: Total Revenue Generated

In [297...

```
Total_Revenue="""select sum(Total_Amount)as Total_Revenue_Generated
from Orders"""
pd.read_sql(Total_Revenue,conn)
```

Out[297…

	Total_Revenue_Generated
0	75628.66

Code 11: One Time Customers

In [157…

```
One_Time_Customers = """
SELECT
COUNT(Distinct Customer_ID) AS One_Time_Customers
FROM Customers
WHERE Customer_ID IN (
SELECT Customer_ID
FROM Orders
GROUP BY Customer_ID
HAVING COUNT(Distinct Order_ID) = 1
)
"""
pd.read_sql(One_Time_Customers,conn)
```

Out[157…

	One_Time_Customers
0	168

Code 12: Count of Repeated Customers

In [158…

```
Repeated_Customers = """
SELECT
    COUNT(Distinct Customer_ID) AS Repeated_Customers
FROM Customers
WHERE Customer_ID IN (
    SELECT Customer_ID
    FROM Orders
    GROUP BY Customer_ID
    HAVING COUNT(Distinct Order_ID) > 1
)
"""
pd.read_sql(Repeated_Customers,conn)
```

Out[158…

	Repeated_Customers
0	139

- In Total from 307 Customers the repeadted Customers are 139 and one time customers are 168.
- that means business in alll the aspects failed to build stong engagement with buyers.

Code 13: Most Recent Customer Details

In [159…

```
Recent_Customer="""
select a.Customer_ID ,
a.Order_Date,
b.Name,
b.Country
,b.City
from Orders as a
join Customers as b
on a.Customer_ID=b.Customer_ID
order by a.Order_Date Desc
limit 5"""
pd.read_sql(Recent_Customer, conn)
```

Out[159…

	Customer_ID	Order_Date	Name	Country	City
0	498	2024-12-07	Brianna Fischer	Gibraltar	Angelatown
1	292	2024-12-06	Lindsey Roberts	Armenia	Lambertfort
2	356	2024-12-05	Mary Winters	Bahamas	Evanshaven
3	310	2024-12-03	Jennifer Lopez	Libyan Arab Jamahiriya	North Emily
4	24	2024-12-02	Christina Mitchell	Trinidad and Tobago	Bridgestown

Code 14: Distinct Country Count

In [161…

```
Total_Countries="Select count(Distinct Country)from Customers"""
pd.read_sql(Total_Countries,conn)
```

Out[161…

	count(Distinct Country)
0	215

Code 15: Churned/Inactive Customers in the Latest Year 2024

In [163…

```
Churned_Customers="""With Churned as (select Customer_ID as Customer,
Max(Order_Date)as Last_Order
from Orders
Group by Customer_ID
Having year(Last_Order)<2024)
 select count(Customer)as churned_Customers
 from Churned"""
pd.read_sql(Churned_Customers, conn)
```

Out[163…

	churned_Customers
0	121

Code 16: Total Orders Placed per Year

In [173…

```
YearlyOrders=""" SELECT count(Distinct Order_ID)as Orders
                 from orders
                 group by year(Order_Date)"""
pd.read_sql(YearlyOrders,conn)
```

Out[173…

	Orders
0	16
1	256
2	228

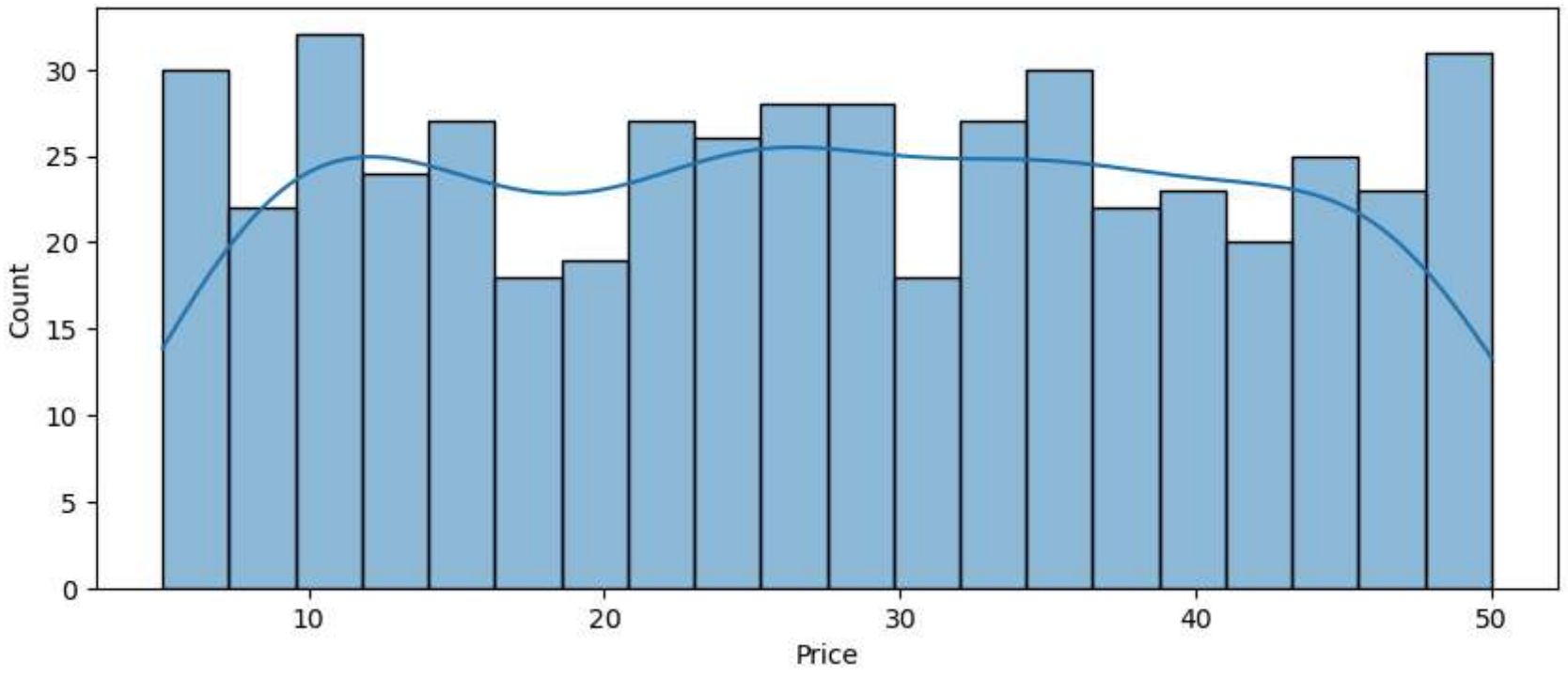
Visualisations & Interpretation of Output/Results

- Visuals such as bar charts, line graphs, and pie charts were used to present trends clearly.
- Sales, customer segments, and genre-wise performance were visualized for better insights.
- Charts helped identify seasonal trends, top-performing books, and customer behavior.
- Understock and overstock patterns were highlighted using conditional formatting and counts.
- Year-wise comparison showed clear growth in low spender segments and one-time buyers.
- Interpretations drawn from visuals guided meaningful business recommendations and actions.

Distribution of Price

In [8]:

```
Price="select Price from Books"
df=pd.read_sql(Price,conn)

plt.figure(figsize=(10,4))
sns.histplot(data=df,x='Price',bins=20,kde=True)
plt.savefig('Price Distibution')
```



- There is the variability in the price range some books are low priced and some are high priced.
- which price segment generates more revenue or having more orders is need to see in detail.
- which will gonna help in understanding customer preferences as well as overall peformance insights

Segmenting Books on the basis of Books Price into 2 Categories High Priced and Low priced.

In [57]:
```
# Adding Column
BooksCategoy= """ Alter table Books
                  Add column Books_Category varchar(20)"""
cursor.execute(BooksCategoy)
conn.commit()
```

In [6]:
```
# Updating Column
BooksSegmentUpdate= """ UPDATE Books
                  SET  Books_Category =  case
                  when price >27 then 'High Priced'
                  else 'Low Priced'
                  END """
cursor.execute(BooksSegmentUpdate)
conn.commit()
```

- So as the Average Book Price is (27.36744) based on this Price books will categories in two segments High priced and Low Priced Books.
- Where if the Books price would be less than average will consider as Low Priced and if > More than average then will be treated as High Priced.

In [94]:
```
# Checkng Query
select="Select * from Books"
pd.read_sql(select,conn)
```

Out[94]:

	Book_ID	Title	Author	Genre	Published_Year	Price	Stock	Books_Category	Stock_Catrgory	Stock_Category	Stock
0	1	Configurable modular throughput	Joseph Crane	Biography	1949	21.34	100	Low Priced	None	None	
1	2	Persevering reciprocal knowledge user	Mario Moore	Fantasy	1971	35.80	19	High Priced	None	None	Optin
2	3	Streamlined coherent initiative	Derrick Howard	Non-Fiction	1913	15.75	27	Low Priced	None	None	Optin
3	4	Customizable 24hour product	Christopher Andrews	Fiction	2020	43.52	8	High Priced	None	None	U
4	5	Adaptive 5thgeneration encoding	Juan Miller	Fantasy	1956	10.95	16	Low Priced	None	None	Optin
...	...	...	...	...	...	...	...	...	...	...	
495	496	Decentralized radical forecast	James Adams	Science Fiction	1966	43.75	96	High Priced	None	None	
496	497	Function-based local installation	Craig Thompson	Science Fiction	1919	24.10	33	Low Priced	None	None	Optin
497	498	Secured 24/7 neural-net	Heather Marks	Non-Fiction	1975	10.88	22	Low Priced	None	None	Optin
498	499	Compatible transitional budgetary management	Isaac Nelson	Biography	1905	6.94	64	Low Priced	None	None	
499	500	Vision-oriented zero tolerance initiative	David Hatfield	Science Fiction	1921	15.94	64	Low Priced	None	None	

500 rows × 11 columns

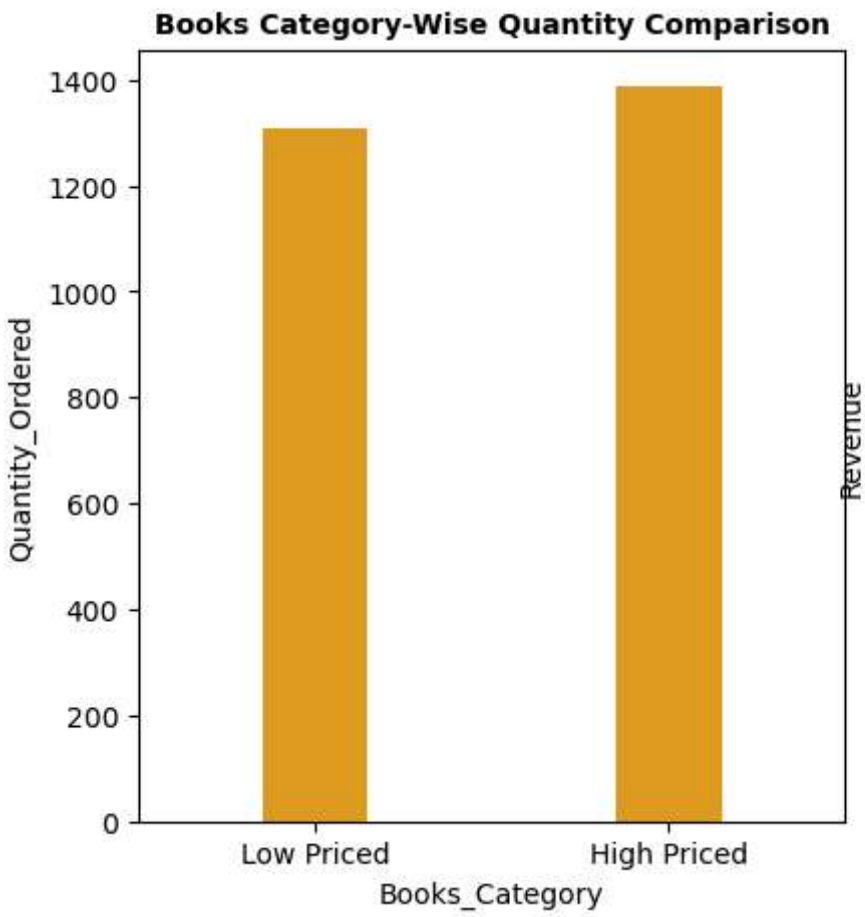
1.Overall Quantity Ordered from Each Category? VS

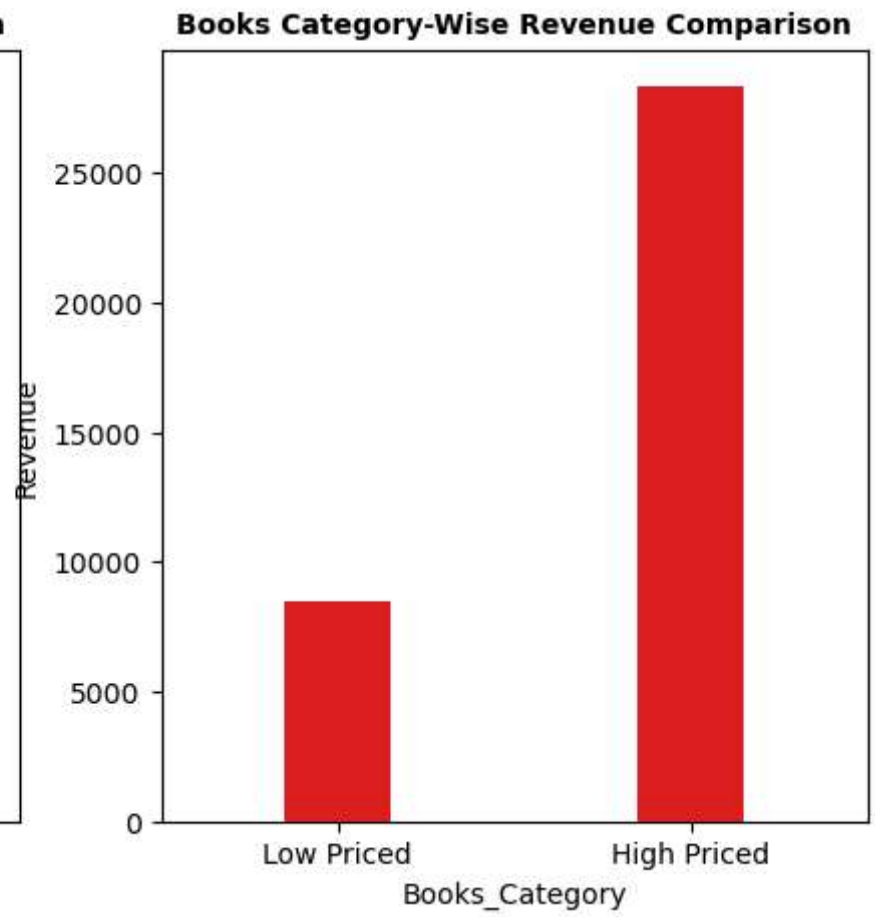
Overall Revenue Generated from Each Category?

In [298…

```
# Executing Query
BooksQuantity =""" SELECT a.Books_Category,sum(b.Quantity ) as Quantity_Ordered
                       FROM Books as a
                       join Orders as b
                       on a.Book_ID= b.Book_ID
                       group by a.Books_Category"""

Data1=pd.read_sql(BooksQuantity ,conn)

BooksRevenue=""" SELECT a.Books_Category,sum(b.Total_Amount) as Revenue
                       FROM Books as a
                       join Orders as b
                       on a.Book_ID= b.Book_ID
                       where year(b.Order_Date)= 2024
                       group by a.Books_Category
                       """
Data2=pd.read_sql(BooksRevenue,conn)

# Printing Query Result
print(Data1)
print(Data2)
plt.figure(figsize=(10,5))
plt.subplot(1,2,1)
sns.barplot(data=Data1,x= 'Books_Category',y='Quantity_Ordered',width=0.3,color='Orange')
plt.title("Books Category-Wise Quantity Comparison",fontsize=10,fontweight='bold')

plt.subplot(1,2,2)
sns.barplot(data=Data2,y='Revenue',x= 'Books_Category',width=0.3,color='Red')
plt.title("Books Category-Wise Revenue Comparison",fontsize=10,fontweight='bold')

plt.savefig('Quantity and Revenue Comparison')
```

```
  Books_Category  Quantity_Ordered
0     Low Priced            1309.0
1    High Priced            1388.0
  Books_Category    Revenue
0     Low Priced   8433.65
1    High Priced  28341.68
```



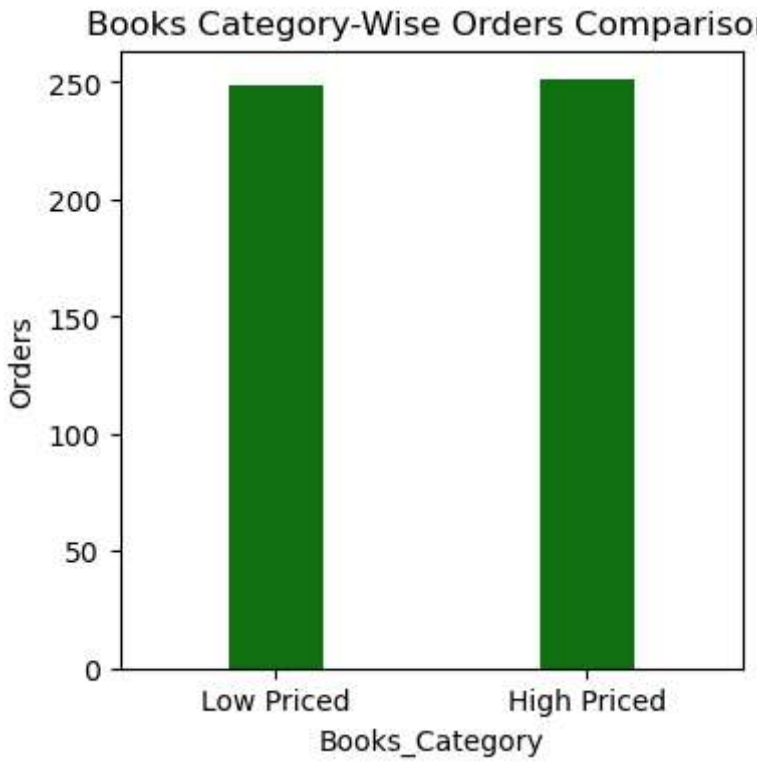


2. Which Books_Category is Customer Preference ? Which Books_Category has Placed More Orders?

In [296…

```
BooksOrders=""" SELECT a.Books_Category,Count(b.Order_ID)as Orders
                        FROM Books as a
                        join Orders as b
                        on a.Book_ID= b.Book_ID
                        group by a.Books_Category"""
Data=pd.read_sql(BooksOrders,conn)
print(Data)

plt.figure(figsize=(4,4))
sns.barplot(data=Data,y='Orders',x= 'Books_Category',width=0.3,color='Green')
plt.title("Books Category-Wise Orders Comparison")
plt.savefig('Orders Comparison')
```

```
  Books_Category  Orders
0     Low Priced     249
1    High Priced     251
```

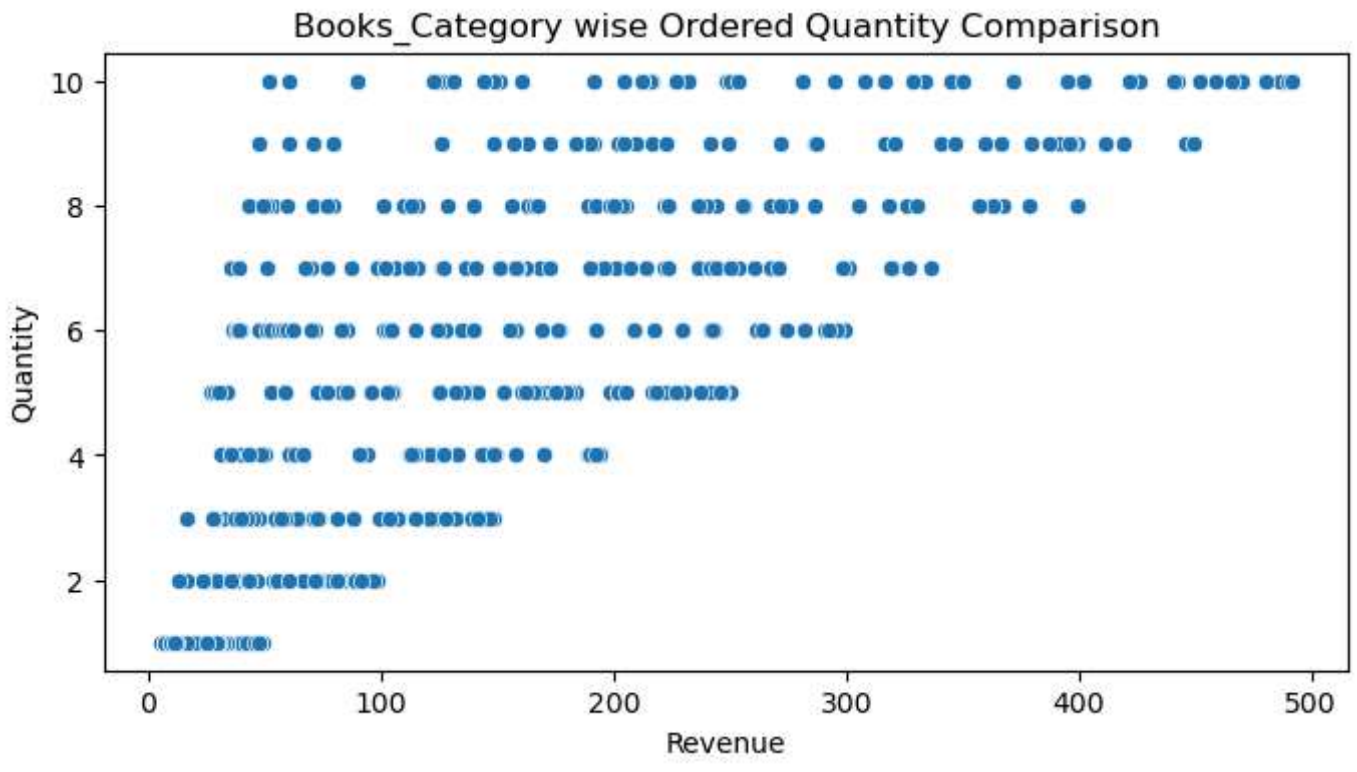


Overall Quantity Ordered from Low Priced Books is (1309) Where as From High Priced Books it is (1388) ,that Higher than Low Priced Books.

- That Means in the aspects of Revenue and Quantity both High Priced Books perfoming better than the Low Priced Books.
- That means We Easily can say that Premium Customers are their who spend good Amount on the Premium Books.

3. Is there a correlation between Quantity and Revenue ? By the Increase in Quantity Can we Expect to generate more Revenue?

In [300…
```
RevenueandOrders="SELECT Quantity ,Total_Amount as Revenue from Orders"
Data=pd.read_sql(RevenueandOrders,conn)
df=pd.DataFrame(Data)
plt.figure(figsize=(8,4))
sns.scatterplot(data=df,x='Revenue',y= 'Quantity')
plt.title("Books_Category wise Ordered Quantity Comparison")
plt.savefig('Quantity and Revenue Corelation')
```



In [226…
```
Quantity_RevenueCorr=df[['Quantity','Revenue']].corr()
Quantity_RevenueCorr
```

Out[226…

	Quantity	Revenue
Quantity	1.000000	0.726438
Revenue	0.726438	1.000000

In [227…
```
print('Correlation coefficient\n------------')
print('Quantity_RevenueCorr:',round(Quantity_RevenueCorr.values[0,1],2))
```

```
Correlation coefficient
------------
Quantity_RevenueCorr: 0.73
```

- So Correlation Between Quantity and Revenue is(0.73) is a positive correlation is their.
- (0.73) is Not really a Very strong Correlation but Good Correlation is there.
- So,yes by the increase in Sales More Quantity Business can Expect Generate More Revenue.

4.What Are the Top 3 Genres in the sense of Orders?

In [88]:
```
Top3Genre_OrdersWise=""" SELECT a.Genre,Count(b.Order_ID)as Orders
                         from Books as a
                         join Orders as b
                         on a.Book_ID=b.Book_ID
                         group by a.Genre
                         Order by Orders desc
                         limit 3"""
pd.read_sql(Top3Genre_OrdersWise,conn)
```

Out[88]:

	Genre	Orders
0	Science Fiction	84
1	Mystery	83
2	Fantasy	81

5.What Are the Top 3 Genres in the sense of Revenue?

In [156…
```
Top3Genre_SalesWise=""" SELECT a.Genre,sum(b.Total_Amount)as Revenue
                        from Books as a
                        join Orders as b
                        on a.Book_ID=b.Book_ID
```

```
                        group by a.Genre
                        Order by Revenue desc
                        limit 3"""
pd.read_sql(Top3Genre_SalesWise,conn)
```

Out[156…

	Genre	Revenue
0	Romance	13086.98
1	Mystery	12788.45
2	Science Fiction	11770.51

- Top 3 Performing Genres are **Science Fiction,Mystery and Fancy** in the case of Orders with Placed Orders **(81,83,84)**.
- on the Other side, **Mystery and Science fictions** also Comes under Top 3 Revenue generating Genres,
- where **Mystery** genre is on the 2 nd position in both orders and Sales Case.
- But **Science fiction** has lost its Position and From 1 st position directly came to 3 rd Position in the Case of Generating Revenue.
- Even if the **Romance** Category is not in the list of Top 3 Ordered Genres ,then also **Romance** is the Top 1 Revenue Generating Genre.
- Romance Genre is Helping Business to Generate More income ,so we Sholud go with Romance Genre.
- and Science Fiction is showing Popularity among Customers we should Provide Some Extra Benefits or Offers On this Genre to Builld Loyal Customers For the Business.
- Total 6 Genre Books are there But,only this 4 Genres perfoming Good.
- other 2 Genres Need More Marketing to shift interest of customers to those genres also.

6.Worst Performing Genre Placing Orders? /Worst Performing Genre Earning Revenue?

In [96]:
```
worstEarningGenre="""SELECT a.Genre,sum(b.Total_Amount)as Revenue
                        from Books as a
                        join Orders as b
                        on a.Book_ID=b.Book_ID
                        group by a.Genre
                        Order by Revenue asc
                        limit 1"""
pd.read_sql(worstEarningGenre,conn)
```

Out[96]:

	Genre	Revenue
0	Fiction	7271.22

In [97]:
```
LessOrderedGenre=""" SELECT a.Genre,Count(b.Order_ID)as Orders
                        from Books as a
                        join Orders as b
                        on a.Book_ID=b.Book_ID
                        group by a.Genre
                        Order by Orders asc
                        limit 1"""
pd.read_sql(LessOrderedGenre,conn)
```

Out[97]:

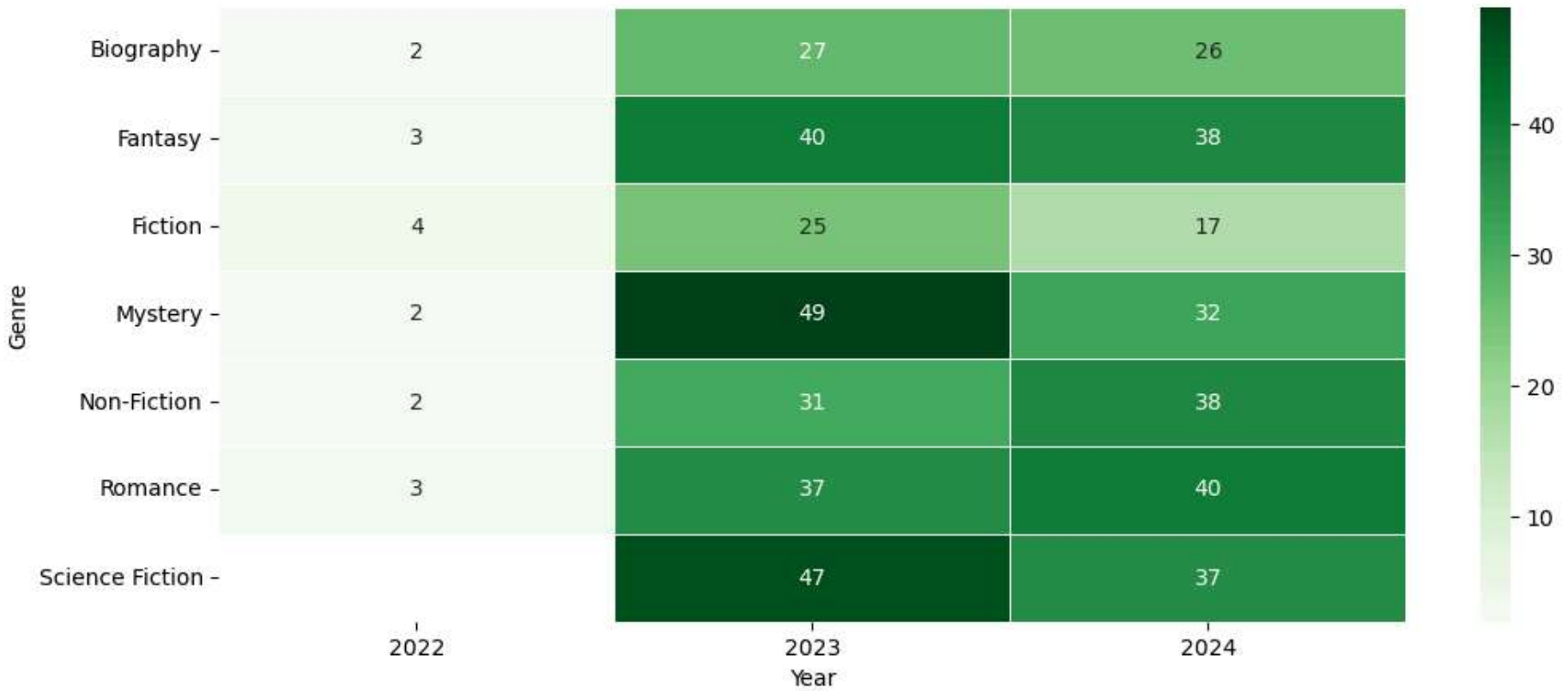
	Genre	Orders
0	Fiction	46

- **Fiction** Genre is the lowest performing Genre ,in the case of both Generating Orders and Generating Revenue.

7.Was there a huge Differenciation in the performance of Good VS Bad performing Genres?

In [293…
```
LessOrderedGenre=""" SELECT a.Genre,Count(b.Order_ID)as Orders,Year(b.Order_Date)as Year
                        from Books as a
                        join Orders as b
                        on a.Book_ID=b.Book_ID
                        group by a.Genre,Year(b.Order_Date)
                        Order by Genre,Year
                        """

GenreComparison=pd.read_sql(LessOrderedGenre,conn)
```

In [314…
```
pivot= GenreComparison.pivot(index='Genre',columns='Year',values='Orders')
plt.figure(figsize=(12,5))

sns.heatmap(pivot, annot=True, cmap='Greens', fmt=",.0f", linewidths=0.5, linecolor='white')

plt.savefig('Genre Orders Comparison')
```

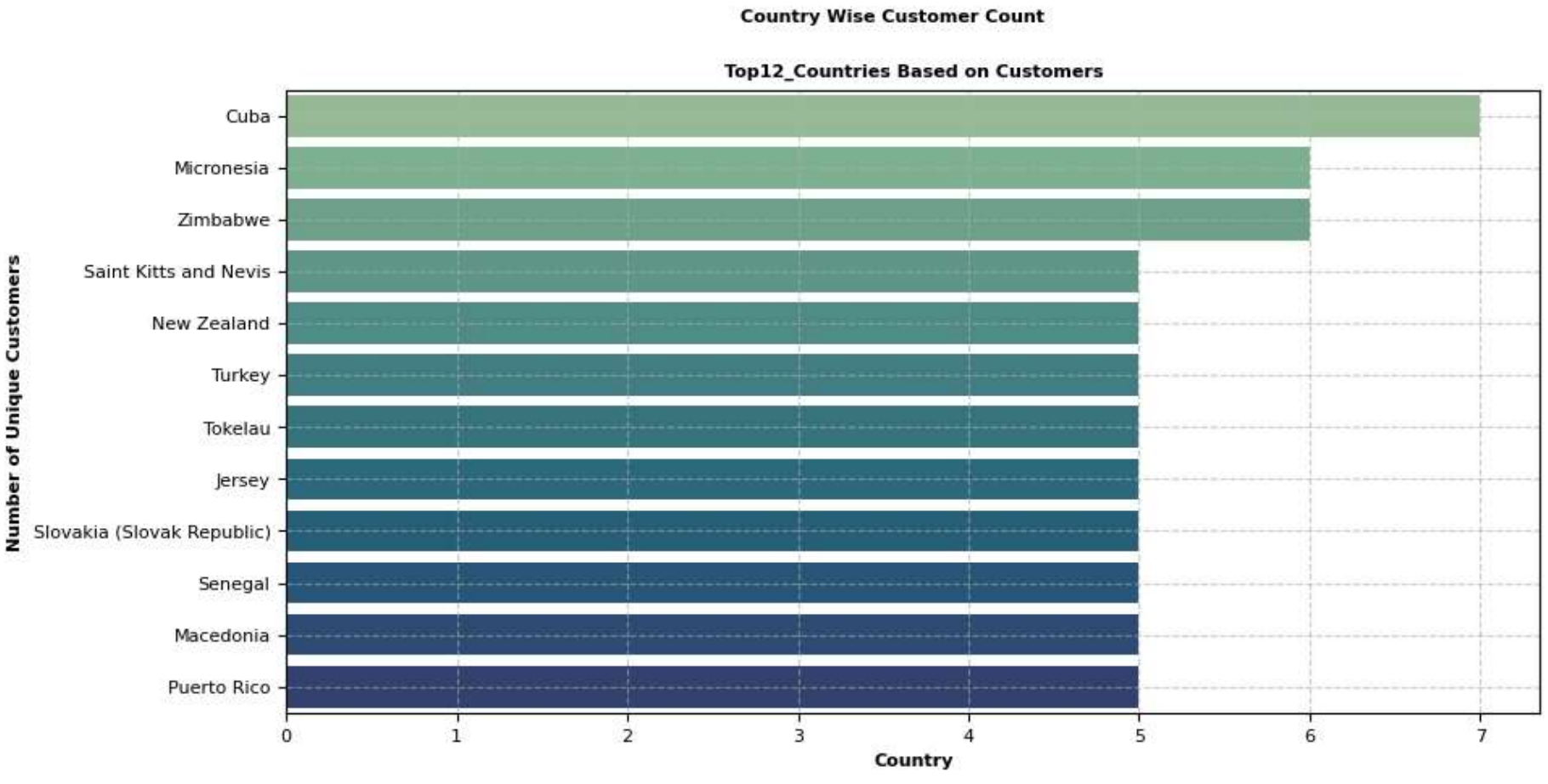


- Here From the Top Orders Placing data we have selected Genres and then Compared with the **Fiction** Genre to understand Why There is a That much difference in the Count of Orders between this **Top Orders Placing Genres VS Fiction Genre**
- In the Initial Year of the Business **Fiction** was on on top 1 but after Years it Failed to compete with other genres and match thier level,as it can be because shift of Customer Preference.
- Where as Biography genre is on second last position.
- There is huge diffeence between this low Performing genres vs Better performing genres.
- where **Maximum Orders Count** of **Biography and Fiction** is **(25 to 27)** only.
- On ther Other Hand, Other Genres **maximum count** ranges Between **(38 to 49)** which a huge big number as compared those 2 Low Performing genres.

8.What are the Top 12 Countries with Customer Count?

In [307…

```
Top12_Customer_Countries="""
select count(Distinct Customer_ID)as
Customer_Count,Country
from Customers
Group by Country
order by Customer_Count desc
limit 12"""
CustomerCount_CountryTop12=pd.read_sql(Top12_Customer_Countries,conn)

plt.figure(figsize=(10,5))


sns.barplot(data=CustomerCount_CountryTop10,
            x='Customer_Count',
            y='Country',
            palette='crest')

plt.title('Top12_Countries Based on Customers', fontsize=8, fontweight='bold')
plt.ylabel('Number of Unique Customers', fontsize=8, fontweight='bold')
plt.xlabel('Country', fontsize=8,fontweight='bold')
plt.xticks(fontsize=8)
plt.yticks(fontsize=8)
plt.grid(True, linestyle='--', alpha=0.6)
plt.suptitle('Country Wise Customer Count',fontsize=8, fontweight='heavy')
plt.tight_layout(rect=[0,0,0,1])
plt.savefig('Country')
plt.show()
```



9.How Many Countries are their where customer count is very low?How the Overall Customer Engament in the Business?

Countries with Customer Count 1

In [40]:
```
Country_Count="""Select Count(Country) as Countries_Count
from Customers
where Country in (Select Country
                  from Customers
                  group by Country
                  having Count(Distinct Customer_ID)=1)
                """
pd.read_sql(Country_Count,conn)
```

Out[40]:

	Countries_Count
0	65

Distinct Country Count

In [45]:
```
Total_Countries="Select count(Distinct Country)from Customers"""
pd.read_sql(Total_Countries,conn)
```

Out[45]:

	count(Distinct Country)
0	215

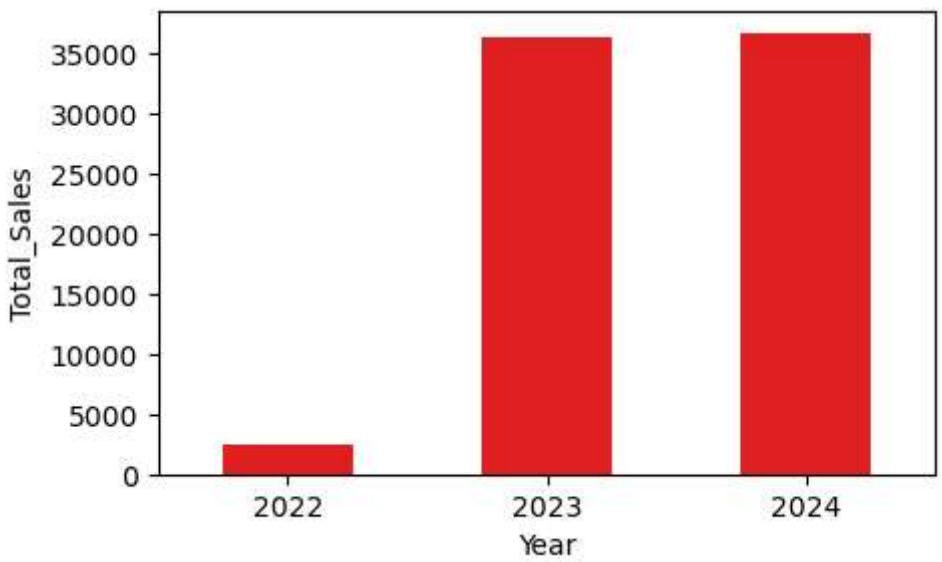
- The business is now after 3 years also in the intial stage only.
- the business has spread eveywhere but not that much awareness is there ,we are placing and getting Customers from the **215** Countries total.
- But from the **215** countries **(65)** Countires are their where the **customer_Count is only 1**
- Needs a Stong Marketing of the Business and Awareness about the products among the Buyers.
- We can provide Discounts on the Particular popular genres tend to Stable Customer Loyalty towards the business.
- Top 3 countries with the highest customer count are **Cuba (7),Zimbabve, and Micronesia**, each receiving between **6 to 7** customers.
- This shows a low customer base in many countries, suggesting limited awareness or outreach.
- Based on this insight, targeted marketing campaigns, promotions, or country-specific strategies are essential to increase global engagement.

10. Is there a positive growth in the Revenue Earning?

In [10]:
```
SalesPerYear="""
select year(Order_Date)as Year,sum(Total_Amount)as Total_Sales
from Orders
group by year(Order_Date)
order by year(Order_Date)
"""
Data=pd.read_sql(SalesPerYear,conn)
```

```
print(Data)

plt.figure(figsize=(5,3))
sns.barplot(data=Data,x='Year',y='Total_Sales',width=0.5,color='Red')

plt.savefig('Yearly Revenue')
plt.show()
```

```
   Year  Total_Sales
0  2022      2513.36
1  2023     36339.97
2  2024     36775.33
```

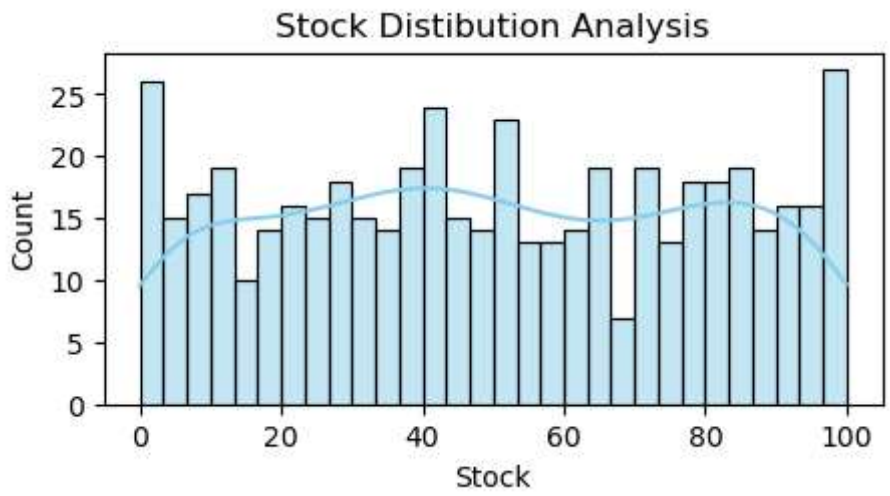


- There is an overall positive growth trend in sales from 2022 to 2024.
- The period from 2022 to 2023 witnessed an explosive rise in sales.
- The growth between 2023 and 2024, while still positive, indicates a stabilization in the growth rate.
- This suggests that while the business scaled up quickly, it may now be entering a mature phase, where growth is steadiy increasing.
- Sales increased from **₹2,513 in 2022 to ₹36,775** in 2024.
- That's nearly 13 times growth in just two years.
- Growth from 2023 to 2024 is around 1.2%, showing steady progress.

11.Is there a Stock of Books available in the optimized quantity?

Stock Distribution

In [17]:
```
Stock="select Stock from Books"
df=pd.read_sql(Stock,conn)

plt.figure(figsize=(5,5))
plt.subplot(2,1,1)
sns.histplot(data=df,x='Stock',bins=30,kde=True,color='skyblue')
plt.title('Stock Distibution Analysis')
plt.savefig('Stock Distribution')
```



- It is Clearly understandable from the visual that there is not an outliers but data is not equallly distibuted as well .
- Some Books possibly overstocked Some in optimum Limit and Some are Understock.
- To find out,do such Overstock books really having that much demand and do this overstock products are contibuting to the revenue or not at such expected level.

Segmentation of Stock into 2 Categories based on Stock Availability

In [90]:
```
StockClassification="""Alter Table Books
                    Add Column Stock_Segment varchar(20)"""

cursor.execute(StockClassification)
conn.commit()
```

In [168…
```
UpdateStock_Category="""update Books
                       Set Stock_Segment =
                       Case
                       when Stock <30 then  'Understock'
                       when Stock >80 then  'Overstock'
                       Else 'Optimum Stock'
                       End"""
cursor.execute(UpdateStock_Category)
conn.commit()
```

In [169…
```
Values="""select a.Stock_Segment,count(b.Order_ID) as Orders
         from Books as a
         join Orders as b
         on a.Book_ID =b.Book_ID
         group by a.Stock_Segment"""

pd.read_sql(Values,conn)
```

Out[169…

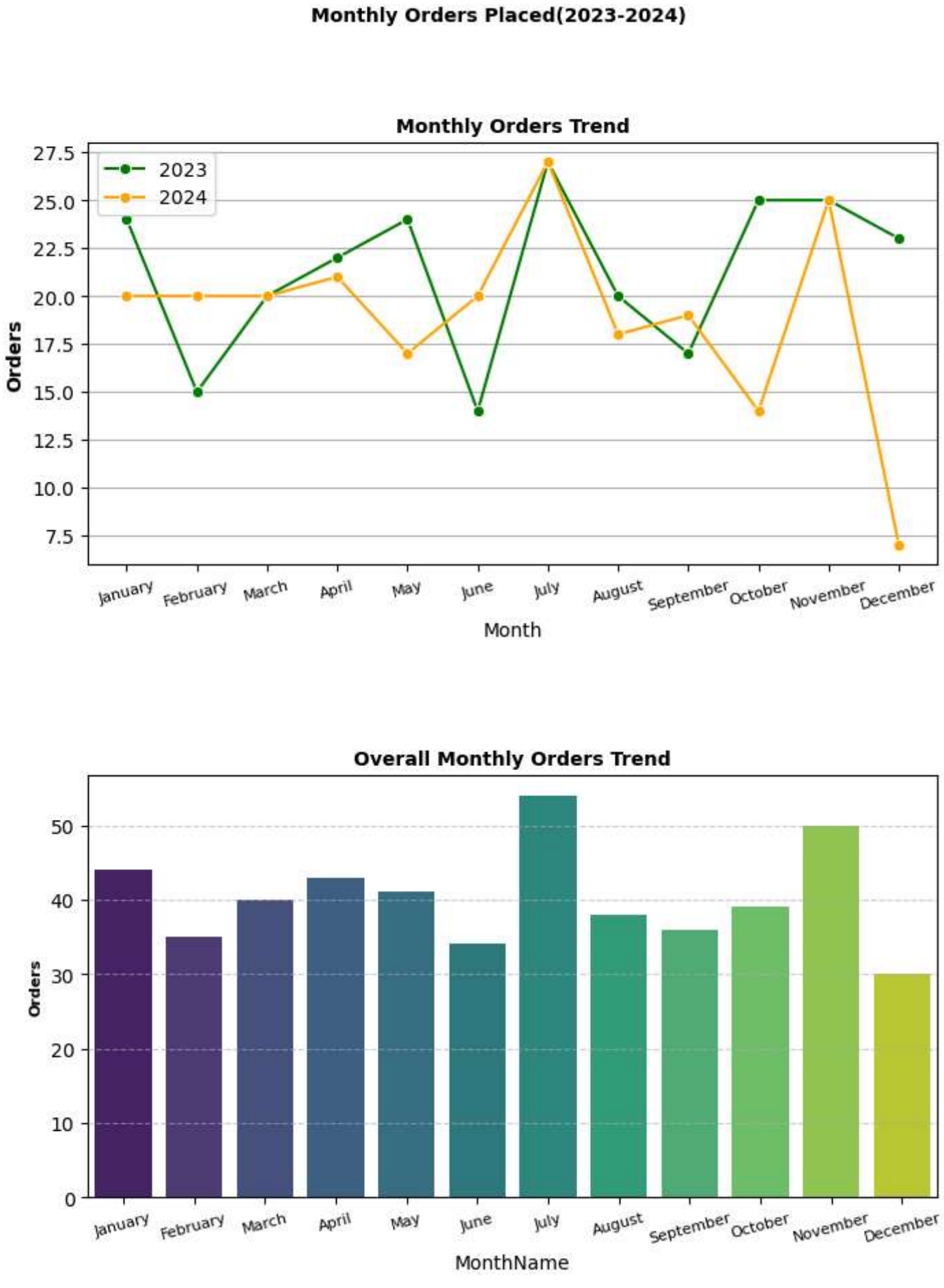
	Stock_Segment	Orders
0	Overstock	101
1	Understock	159
2	Optimum Stock	240

- Optimum Segment Books stocked in the ideal quantity range and received the most customer interest.
- **Understock** Books received **159** orders had stong demand-
- 1.suggests lost sales potential due to insufficient stock.
- 2.These Books needs stock level adjustment.
- **Ovestock** books had only **101** orders ,the lowest among all,indicates **low demand relative to stock levels**.
- The business can definately control stock budgets by adjusting stock levels for this category books or can provide discounts to sale this books more.

12. Key Insights: Monthly Orders Comparison (2023 vs 2024)

In [18]:
```
 # Fetching 2023 Orders insights
Borrowing2023= """
SELECT
    MONTHNAME(Order_Date) AS Month,
    COUNT(DISTINCT Order_ID) AS Borrowing_Events_2023
FROM Orders
WHERE EXTRACT(YEAR FROM Order_Date) = 2023
GROUP BY MONTHNAME(Order_Date), EXTRACT(MONTH FROM Order_Date)
ORDER BY EXTRACT(MONTH FROM Order_Date)
"""
Monthly_Analysis2023=pd.read_sql(Borrowing2023 , conn)
# Fetching 2024 Orders insights
Borrowingand2024 = """
SELECT
    MONTHNAME(Order_Date) AS Month,
    COUNT(DISTINCT Order_ID) AS Borrowing_Events_2024
FROM Orders
WHERE EXTRACT(YEAR FROM Order_Date) = 2024
GROUP BY MONTHNAME(Order_Date), EXTRACT(MONTH FROM Order_Date)
ORDER BY EXTRACT(MONTH FROM Order_Date)
"""
Monthly_Analysis2024= pd.read_sql(Borrowingand2024 , conn)

# Merging Both Months Orders insights
Monthly_Trend_Comparison = pd.merge(
   Monthly_Analysis2023 ,
    Monthly_Analysis2024,
    on='Month',
    how='inner'
).fillna(0)

# fetching Overall Monthly Orders Insights

OrdersPerMonth = """
SELECT
    MONTH(Order_Date) AS Month,
    MONTHNAME(Order_Date) AS MonthName,
     COUNT(DISTINCT Order_ID) as Orders
FROM Orders
```

```
where Year(Order_Date) in (2023,2024)
GROUP BY MONTH(Order_Date), MONTHNAME(Order_Date)
ORDER BY Month
"""
MonthlyOrders= pd.read_sql(OrdersPerMonth, conn)

# plotting Results
plt.figure(figsize=(8,10))
plt.subplot(2,1,1)
sns.lineplot(data=Monthly_Trend_Comparison, x='Month', y='Borrowing_Events_2023', label='2023', marker='o', color='green')
sns.lineplot(data=Monthly_Trend_Comparison, x='Month', y='Borrowing_Events_2024', label='2024', marker='o', color='orange')
plt.title('Monthly Orders Trend', fontweight='heavy', fontsize=10)
plt.ylabel('Orders',fontweight='heavy')
plt.xticks(rotation=15,fontsize=8)
plt.legend()
plt.grid(axis='y')


plt.subplot(2,1,2)
sns.barplot(data=MonthlyOrders,x='MonthName', y='Orders', palette='viridis')

plt.title('Overall Monthly Orders Trend', fontsize=10, fontweight='bold')
plt.ylabel('Orders', fontsize=8, fontweight='bold')
plt.grid(axis='y', linestyle='--', alpha=0.6)
plt.xticks(rotation=15,fontsize=8)
plt.suptitle('Monthly Orders Placed(2023-2024)',fontweight='heavy', fontsize=10)
plt.savefig('Montly Trend.png')
plt.tight_layout(rect=[0,0,0,1])
plt.subplots_adjust(hspace=0.5,wspace=0.5)
plt.show()
```



- **July** month is the peak month.Both in 2023 and 2024 july had the highest orders.
- November is on the Second highest Position.
- In this Both months of July and November fo both years orders are same showing consitent trends in those paticular months.
- 2024 underperformed in several key months (notably May, October, December).
- Focus for improvement in 2024: investigate and address reasons for underperformance in May, October, and December.
- Plan promotions or outreach campaigns targeting low-performing months to balance the trend.
- so in short **November,July** are peak Months.
- **May,October** and Specially **December** Needs attention.
- There is a significant growth in the month of **June and and February.**
- **2023 had stronger and more consistent performance compared to 2024**

13. Who are the Top 10 customers based on orders placed?

In [316…

```
Top10Customers_OrdersBased="""with Top10 as(
                         Select a.Customer_ID,
                        count(Distinct b.Order_ID)as Orders_placed
                        from Customers as a
                        join Orders as b
                        on a.Customer_ID =b.Customer_ID
                        group by a.Customer_ID
                        order by Orders_placed desc
                        limit 10)
                        select a.Name as Customer, b.Orders_placed,a.Phone,a.Email,
                        a.Country,a.City
                        from Customers as a
                        join Top10 as b
                        on a.Customer_ID = b.Customer_ID
                        order by Orders_placed desc"""
pd.read_sql(Top10Customers_OrdersBased,conn)
```

Out[316…

	Customer	Orders_placed	Phone	Email	Country	City
0	Carrie Perez	6	1234568254	chelsea23@gillespie-walker.com	Hungary	Kennethland
1	Anthony Young	5	1234568364	rogersbill@gmail.com	Cook Islands	East Chelsea
2	Amy Hunt	4	1234567997	emilybecker@perkins.com	Aruba	Ericborough
3	Jonathon Strickland	4	1234568064	ryan10@yahoo.com	Dominica	Bakerton
4	Emily Vargas	4	1234568215	lklein@gmail.com	Tonga	Aguilarside
5	Julie Smith	4	1234568295	knightmonica@krueger-hamilton.biz	Vanuatu	Freemanland
6	Ashley Perez	4	1234568315	williamslindsey@yahoo.com	United States Minor Outlying Islands	Elizabethshire
7	Cynthia Cooper	4	1234568327	russellpriscilla@gmail.com	Philippines	Wrightfurt
8	Kim Turner	4	1234568347	jennifer45@weiss-perry.com	Cambodia	South Rachelview
9	Andrew Figueroa	4	1234568375	john28@gmail.com	Macedonia	New Veronicaside

In [189…

```
Top10Customers=pd.read_sql(Top10Customers_OrdersBased,conn)
Top10CustomersDetails=pd.DataFrame(Top10Customers)
```

- Here with the help of Orders count performed query to get the Top customers.
- So that afterwards it willl help the business to make more engagement with them to gain long time loyal customers.
- We can provide Discount to them on there most favouite genre books and also can introduce them new books.

14. Who are the Top 10 customers based on Revenue Generated?

In [204…

```
Top10Customers_RevenueBased="""with Top10 as(
                         Select a.Customer_ID,
                        sum(b.Total_Amount)Revenue_Generated,
                        from Customers as a
                        join Orders as b
                        on a.Customer_ID =b.Customer_ID
                        group by a.Customer_ID
                        order by Revenue_Generated desc
                        limit 10)
                        select a.Name as Customer, b.Revenue_Generated,a.Phone,a.Email,
                        a.Country,a.City
                        from Customers as a
                        join Top10 as b
                        on a.Customer_ID = b.Customer_ID
```

```
                        order by Revenue_Generated desc"""
pd.read_sql(Top10Customers_RevenueBased,conn)
```

Out[204...

	Customer	Revenue_Generated	Phone	Email	Country	City
0	Kim Turner	1398.90	1234568347	jennifer45@weiss-perry.com	Cambodia	South Rachelview
1	Jonathon Strickland	1080.95	1234568064	ryan10@yahoo.com	Dominica	Bakerton
2	Carrie Perez	1052.27	1234568254	chelsea23@gillespie-walker.com	Hungary	Kennethland
3	Julie Smith	991.00	1234568295	knightmonica@krueger-hamilton.biz	Vanuatu	Freemanland
4	Pamela Gordon	986.30	1234568276	mandy28@thomas-white.com	Yemen	East Richardburgh
5	Ashley Perez	942.62	1234568315	williamslindsey@yahoo.com	United States Minor Outlying Islands	Elizabethshire
6	Anthony Young	929.19	1234568364	rogersbill@gmail.com	Cook Islands	East Chelsea
7	Robert Clark	746.65	1234568053	sheilalester@gmail.com	Macao	Lake Charleshaven
8	Justin Spencer	719.93	1234568057	michaelsnyder@gmail.com	South Africa	Christopherchester
9	Alexander Scott	682.15	1234568104	amypierce@hotmail.com	El Salvador	Matthewfurt

15. What are the Top 10 Books Revenue Generated

In [276...

```
Top10BooksRevenueBased="""with Top as(
                         Select a.Book_ID,
                        sum(b.Total_Amount)Revenue_Generated
                        from Books as a
                        join Orders as b
                        on a.Book_ID=b.Book_ID
                        group by a.Book_ID
                        order by Revenue_Generated desc
                        limit 10)
                        select a.Title,a.Author,a.Genre,a.Price,
                        a.Published_Year,b.Revenue_Generated
                        from Books as a
                        join Top as b
                        on a.Book_ID= b.Book_ID
                        order by Revenue_Generated desc"""
pd.read_sql(Top10BooksRevenueBased,conn)
```

Out[276...

	Title	Author	Genre	Price	Published_Year	Revenue_Generated
0	Integrated secondary access	Sheena Harris	Non-Fiction	48.03	1984	1104.69
1	Multi-tiered responsive parallelism	Amanda Wilson	Fiction	48.96	1940	1077.12
2	Switchable modular moratorium	Tonya Saunders	Romance	49.88	2010	1047.48
3	Cross-platform next generation website	Anna Roberts	Romance	45.38	1929	952.98
4	Grass-roots systematic moderator	Joyce Patton	Fantasy	45.91	1919	872.29
5	Innovative empowering concept	Brad Vasquez	Science Fiction	45.20	1964	813.60
6	Assimilated composite archive	Mark Gibson	Fiction	46.66	1957	793.22
7	Robust tangible hardware	Paul Miles	Non-Fiction	40.22	1999	764.18
8	Robust attitude-oriented attitude	Zachary Hayes	Biography	49.50	1955	742.50
9	Stand-alone content-based hub	Lisa Ellis	Fantasy	49.90	1957	698.60

16. What are the Low Perfoming Books ?

In [274...

```
BottomBooksRevenueBased="""with Bottom as(
                         Select a.Book_ID,
                        sum(b.Total_Amount)Revenue_Generated,
                        sum(b.Quantity) as Quantity_Sold,
                        sum(a.Stock) as Stock
                        from Books as a
                        join Orders as b
                        on a.Book_ID=b.Book_ID
                        group by a.Book_ID
                        order by Revenue_Generated
                       )
                        select a.Title,a.Author,a.Genre,a.Price,a.Stock_Segment,a.Stock,
                        a.Published_Year,b.Revenue_Generated,b.Quantity_Sold
                        from Books as a
                        join Bottom as b
                        on a.Book_ID= b.Book_ID
```

```
                      where a.Stock>60 and b.Quantity_Sold<3
                      order by Revenue_Generated,Stock
                      """
pd.read_sql(BottomBooksRevenueBased,conn)
```

Out[274...

	Title	Author	Genre	Price	Stock_Segment	Stock	Published_Year	Revenue_Generated	Quantity_Sold
0	Realigned context-sensitive pricing structure	Jason Rodriguez	Fiction	6.64	Overstock	90	2004	6.64	1.0
1	Seamless analyzing encoding	Kevin Garcia	Mystery	10.17	Optimum Stock	61	1949	10.17	1.0
2	Reduced discrete leverage	Zachary Buchanan	Mystery	20.96	Overstock	90	1911	20.96	1.0
3	Optional stable matrix	Michael Wells	Non-Fiction	11.28	Optimum Stock	62	1947	22.56	2.0
4	Multi-channeled 5thgeneration Internet solution	Jennifer Powell	Biography	24.70	Overstock	94	1963	24.70	1.0
5	Adaptive didactic interface	Natalie Gonzalez	Fiction	25.97	Overstock	94	1923	25.97	1.0
6	Digitized executive flexibility	Lisa Lopez	Non-Fiction	38.01	Overstock	84	1960	38.01	1.0
7	Stand-alone zero administration emulation	Michelle Lyons	Romance	47.76	Overstock	100	1986	47.76	1.0
8	Team-oriented dedicated attitude	Jeffrey Richardson	Biography	25.83	Overstock	91	2006	51.66	2.0
9	Integrated exuding application	Elizabeth Morrison	Romance	27.82	Overstock	81	1923	55.64	2.0
10	Expanded analyzing portal	Lisa Coffey	Fiction	37.51	Optimum Stock	79	1941	75.02	2.0
11	Synergistic user-facing frame	David Olson	Non-Fiction	43.91	Overstock	97	1977	87.82	2.0
12	Stand-alone logistical installation	Elizabeth Williams	Biography	44.60	Overstock	97	1913	89.20	2.0
13	Innovative didactic capacity	Matthew Vazquez	Science Fiction	48.21	Optimum Stock	66	2009	96.42	2.0

- This Bottom Books are the lowest perfoming books are there , Revenue range between **6.64 to 96.42 only**
- Sum of Stock shows the Total Stock of 3 years that this books have always stocked in higher quantity fom the begining even though there is no sale.
- if look at the Quantity sold is only **1 to 2.**
- Where as on the other hand if we see the Top3 Revenue Generated Books Income is between **1047.12 to 1104.69**.
- Showing the difference on largest scale Between this numbers and also falls under Optimum Stock and OverStock Category so we can simply remove them as thie is not at all demand fo this products in the market to reduce Cost
- that will result in stock Optimization and can invest in stocking those books that have demand among buyers.

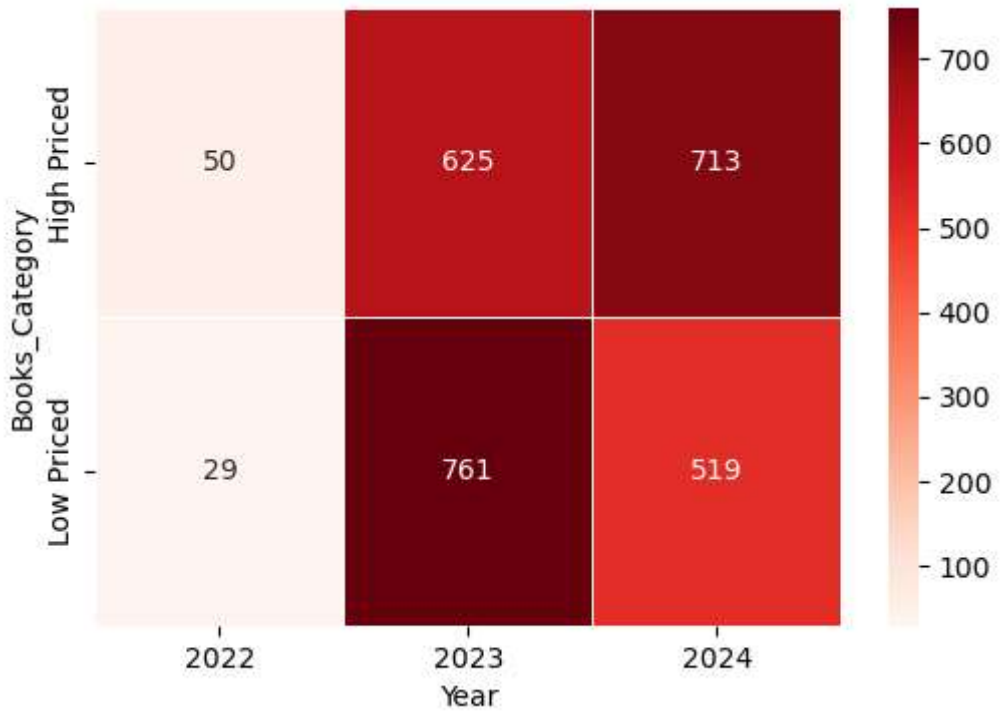
17.How Was the Quantity Trend Over the Years Based on Books Categoy?

In [20]:
```
BooksCategoryQuantity =""" SELECT a.Books_Category,sum(b.Quantity ) as Quantity_Ordered,year(b.Order_Date) as Year
                        FROM Books as a
                        join Orders as b
                        on a.Book_ID= b.Book_ID
                        group by year(b.Order_Date), a.Books_Category"""
Books_Quantity_Data=pd.read_sql(BooksCategoryQuantity ,conn)
print(Books_Quantity_Data)
```

```
  Books_Category  Quantity_Ordered  Year
0     Low Priced             519.0  2024
1     Low Priced              29.0  2022
2    High Priced              50.0  2022
3    High Priced             625.0  2023
4     Low Priced             761.0  2023
5    High Priced             713.0  2024
```

In [21]:
```
pivot=Books_Quantity_Data.pivot(index='Books_Category',columns='Year',values='Quantity_Ordered')
plt.figure(figsize=(6,4))
sns.heatmap(pivot, annot=True, cmap='Reds', fmt=",.0f", linewidths=0.5, linecolor='white')
```

Out[21]: <Axes: xlabel='Year', ylabel='Books_Category'>



- Both in the year (2022,2024) High Priced Books Have Sold More.
- only in the Year 2023 Low Piced Category Placed sold More Quantity.

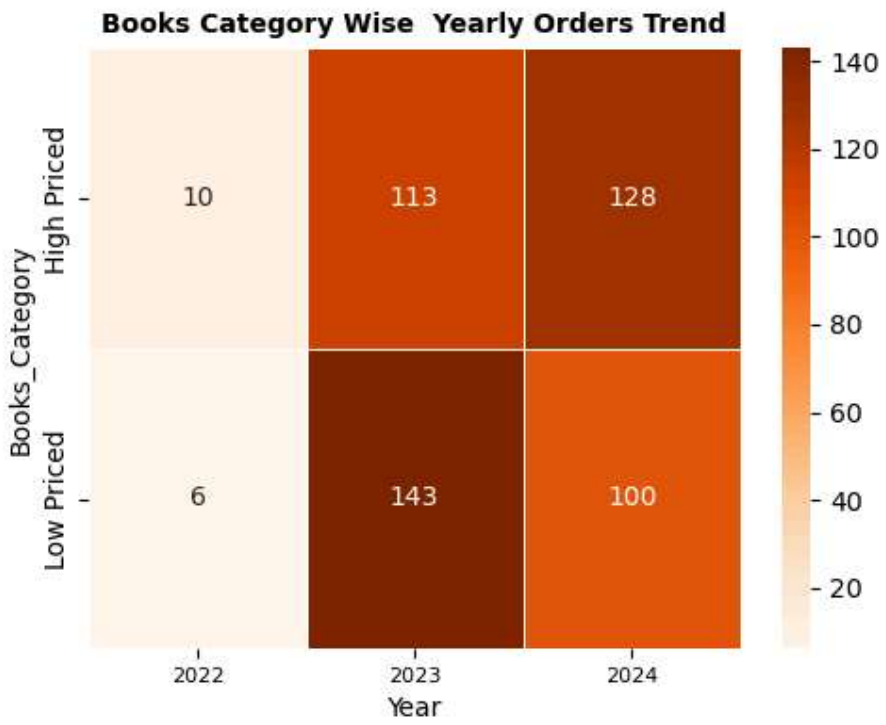
18. How was the Orders and Revenue trend Yearly of Books Category?

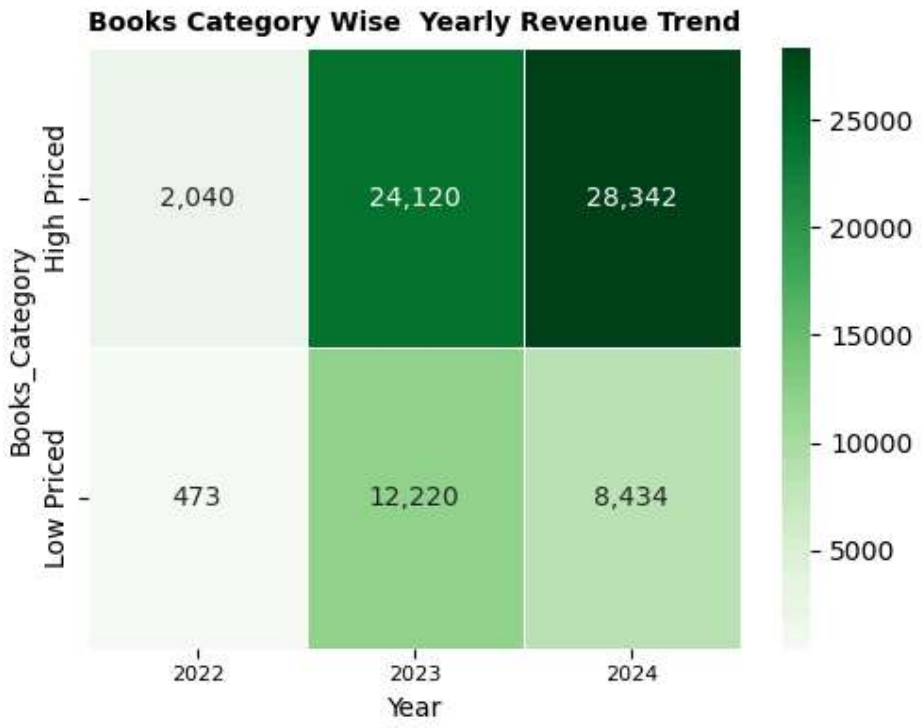
In [23]:
```
BooksCategoryOrders=""" SELECT a.Books_Category,Count(b.Order_ID)as Orders,Year(b.Order_Date) as Year
                        FROM Books as a
                        join Orders as b
                        on a.Book_ID= b.Book_ID
                        group by a.Books_Category,Year(b.Order_Date)"""
BooksOrders_Data=pd.read_sql(BooksCategoryOrders,conn)

BooksCategoyRevenue=""" SELECT a.Books_Category,sum(b.Total_Amount) as Revenue,Year(b.Order_Date) as Year
                        FROM Books as a
                        join Orders as b
                        on a.Book_ID= b.Book_ID
                        group by a.Books_Category,Year(b.Order_Date)"""
BooksRevenue_Data=pd.read_sql(BooksCategoyRevenue,conn)
print('Books_OrdersTrend\n------------')
print(BooksOrders_Data)
print('Books_RevenueTrend\n------------')
print(BooksRevenue_Data)
```

```
Books_OrdersTrend
------------
  Books_Category  Orders  Year
0     Low Priced     100  2024
1     Low Priced       6  2022
2    High Priced      10  2022
3    High Priced     113  2023
4     Low Priced     143  2023
5    High Priced     128  2024
Books_RevenueTrend
------------
  Books_Category   Revenue  Year
0     Low Priced   8433.65  2024
1     Low Priced    473.34  2022
2    High Priced   2040.02  2022
3    High Priced  24120.20  2023
4     Low Priced  12219.77  2023
5    High Priced  28341.68  2024
```

In [24]:
```
pivot=BooksOrders_Data.pivot(index='Books_Category',columns='Year',values='Orders')
plt.figure(figsize=(10,8))

plt.subplot(2,2,1)
sns.heatmap(pivot, annot=True, cmap='Oranges', fmt=",.0f", linewidths=0.5, linecolor='white')
plt.xticks(fontsize=8)
plt.title("Books Category Wise  Yearly Orders Trend",fontsize=10,fontweight='bold')

plt.subplot(2,2,2)
pivot=Books_Quantity_Data.pivot(index='Books_Category',columns='Year',values='Quantity_Ordered')
sns.heatmap(pivot, annot=True, cmap='Reds', fmt=",.0f", linewidths=0.5, linecolor='white')
plt.title("Books Category Wise  Yearly Quantity Ordered Trend",fontsize=10,fontweight='bold')
plt.xticks(fontsize=8)

plt.subplot(2,2,3)
pivot=BooksRevenue_Data.pivot(index='Books_Category',columns='Year',values='Revenue')
sns.heatmap(pivot, annot=True, cmap='Greens', fmt=",.0f", linewidths=0.5, linecolor='white')
plt.title("Books Category Wise  Yearly Revenue Trend",fontsize=10,fontweight='bold')
plt.xticks(fontsize=8)
plt.subplots_adjust(hspace=0.5,wspace=0.5)
```

```
plt.tight_layout()
plt.savefig('Yearly OverallComparison')
plt.show()
```





- **By Comparing 3 visuals here we are tying to figure out that do increase in Orders Increases Quantity and is there a correlation between Orders and Revenue.**
- In High priced Books Case from **2022 to 2024** positive trend is there in both case of Orders and the Quantity where Orders from (10) to increased to the **(128)** in this three years the Quantity from **50** to reached to **713.**
- But in the Case of Low Priced Books Orders increased from **6 to 143** and Quantity supplied from **(29) to (761)** from the year **(2022-2023).**
- But in the the 2024 trend is declined directly from **143 to 100** ,that also impacted the Quantity sold and only **(519)** Quantity got sold in **2024**.
- And Lastly it also visible in the Revenue map where as the Quantity Ordered increases Revenue Also increased and In **2024** in the case of Low Priced Books as the Orders and Quantity Decreased Reveneue also Decreased.
- So in short we can clearly say that Quantity,Orders and Revenue Has a strong Relationship which is Clearly visible in this 3 years Comparison.
- Even though there is a difference in the values but then also eqaully both categoies have performed well and helped business to grow overall.

📍 Market & Customer Insights:

- We have identified 12 countries with more consumers — focus marketing and supply in these regions to create potential loyal customers.
- Explored data of diverse customers — analyze the genres they prefer and recommend similar books to increase engagement.
- We have a list of top revenue-generating and Orders placed customers — provide exclusive offers or personalized messages to maintain their interest.

📌 Recommendations:

- Reduce purchase of overstocked books with low demand Instead invest in High Demand Books.
- July and Novembe are the peak month promote new books in this month as customers tend to spend in this months more.

- Restock Understock books that have high demand.
- Monitor past order quantity to plan stock effectively.
- In the Year 2023 the Performance was better than 2024 latest year.
- There is Strong Correlation between oders,Quantity and Revenue.
- So to generate More income increasing orders is key solution.
- Set alerts for books with critically low stock.
- Use discounts/offers to clear excess inventory.
- Stock Science Fiction ,Mystey and Romance Genre books this are popular genres among the customers.
- Revisit pricing of books with low sales.We can minimize the prices of the low demand books to increase orders.
- Avoid overstocking slow-moving titles.
- Instead of focusing on connecting to world wide try establishing busines in those countries where customer count is high.
- Engage with High Revenue Generated and High Orders Placed Customers to build customer loyalting.
- Connect with Churned Customers through Emails,and Messages provide them best offers and Discounts.
- Manage stock based on customer demand patterns.
- Promote top-selling books more aggressively.

📚 Conclusion:

- Monitored popular genres based on order history to ensure you're offering what readers love.
- This project analyzed 500+ rows of data from Books, Orders, and Customers tables.
- We discovered a rising customer base year by year, with 2023 having the highest engagement.
- Science Fiction genres emerged as the most borrowed, with changing trends over the years.
- Revenue insights showed which books and customers contributed most to sales.
- Monthly trends helped understand demand cycles and seasonality in orders.
- Stock analysis classified books into understocked, overstocked, and optimum levels.
- Repeated and One Time readers were identified, aiding in customer retention strategies.
- Time-based comparisons gave visibility into 2023 vs. 2024 borrowing patterns.
- Overall, the project delivered actionable insights for inventory planning and customer targeting.