**D-mart Store Sales Dashboard**

**Overview**

This Power BI dashboard provides an interactive and comprehensive view of Smart Store sales performance, focusing on key metrics like total sales, total quantity sold, total profit, and total unique orders. The dashboard is designed to help businesses monitor their sales trends, identify top-performing products and regions, and gain insights into profit distribution over time.

**Features**

- \*\*KPIs\*\*: The dashboard displays essential KPIs, including:

- \*\*Total Sales\*\*: Sum of all sales transactions

- \*\*Total Quantity Sold\*\*: Total number of items sold

- \*\*Total Profit\*\*: Profit generated from all sales

- \*\*Total Unique Orders\*\*: The count of distinct orders placed

- \*\*Treemap by Year\*\*: A visual breakdown of sales over different years, highlighting the proportion of yearly sales in a hierarchical view.

- \*\*Monthly Sales (Column Chart)\*\*: This chart provides a month-by-month comparison of total sales, helping users track seasonal trends and monthly performance.

- \*\*Profit by Time (Line Chart)\*\*: A line chart visualizing profit by month, year, and quarter, allowing users to analyze profit growth over time.

- \*\*Top Sales by Weekday (Column Chart)\*\*: This chart shows the sales distribution across weekdays, helping to identify the best-performing days for sales.

- \*\*Top and Bottom 5 Performing Cities\*\*: A comparison chart that highlights the top 5 cities with the highest sales and the bottom 5 cities with the lowest sales, offering insights into geographical performance.

- \*\*Sales by Product Category\*\*: A detailed breakdown of sales by different product categories, providing a clear view of which product lines are driving revenue.

- \*\*Map View\*\*: A geographic map that gives an overview of total sales by location, allowing users to see sales distribution across different regions at a glance.

**Data Model**

The data model behind this dashboard consists of multiple fact and dimension tables, optimized for performance in Power BI. The relationships between these tables ensure efficient querying and smooth interactivity, especially when slicing and dicing data based on categories like time, geography, and product details.

**Key DAX Measures**

- `Total Sales = SUM(Sales[SalesAmount])`

- `Total Quantity = SUM(Sales[Quantity])`

- `Total Profit = SUM(Sales[Profit])`

- `Total Unique Orders = DISTINCTCOUNT(Sales[OrderID])`

These DAX measures power the KPIs and ensure accurate real-time calculations across the dashboard.

**Technology Stack**

- \*\*Power BI\*\*: Used for data modeling, visualization, and interactivity.

- \*\*DAX\*\*: For creating calculated fields and KPIs.

- \*\*Power Query\*\*: For data cleaning and transformation.

- \*\*SQL Database\*\*: Data source for sales transactions and product data.

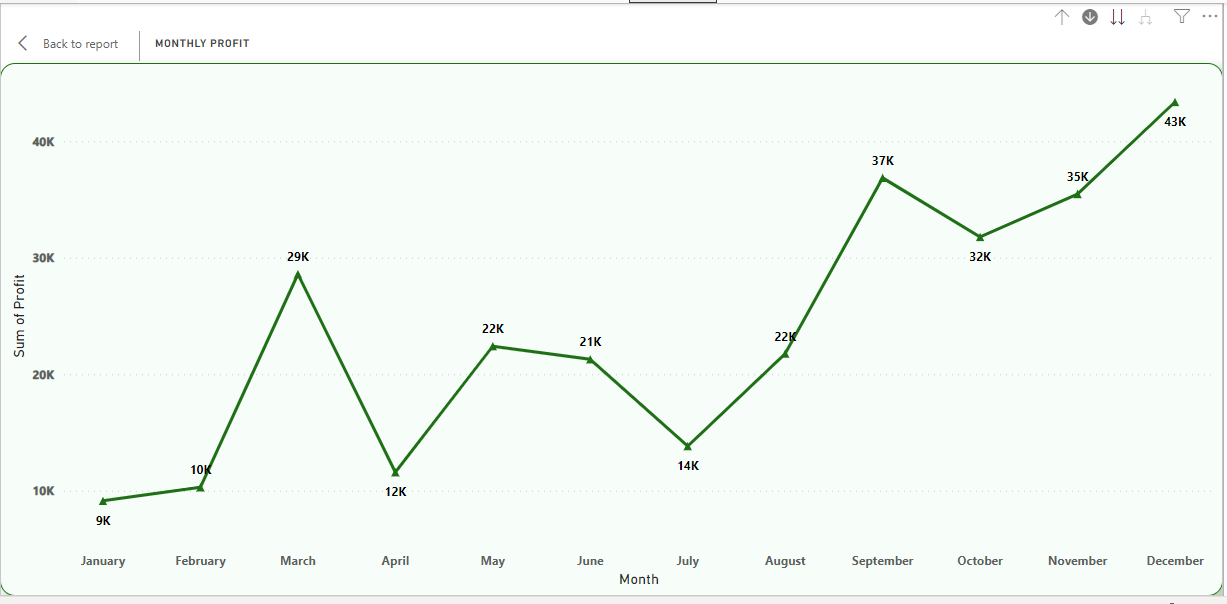
**Screenshots**

Overall View

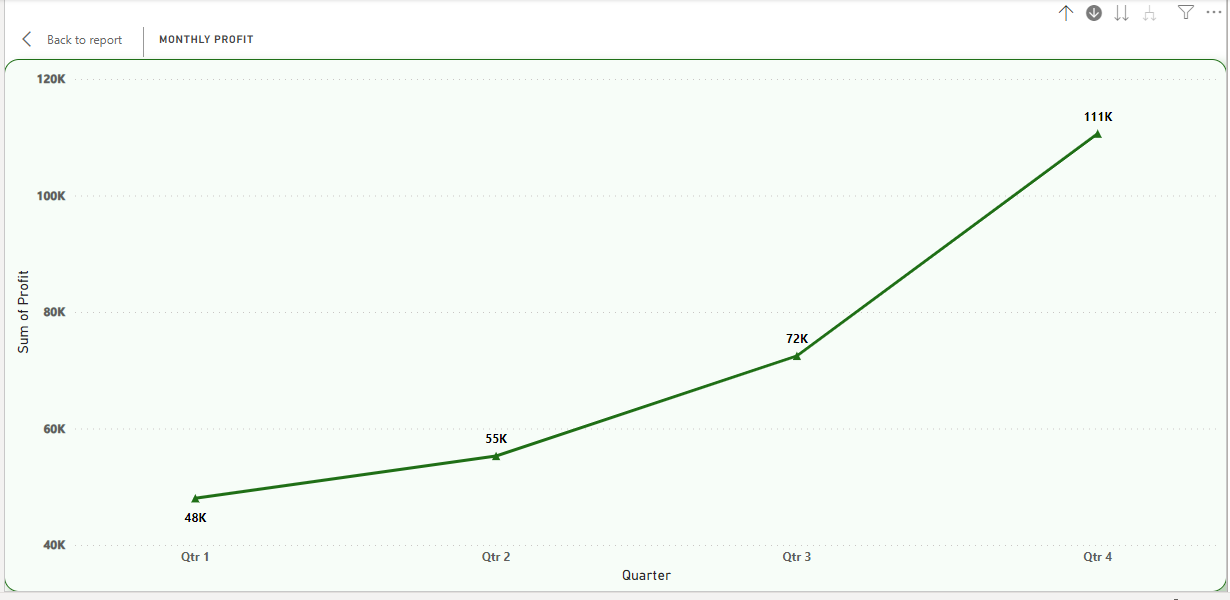


**Profit Summary**

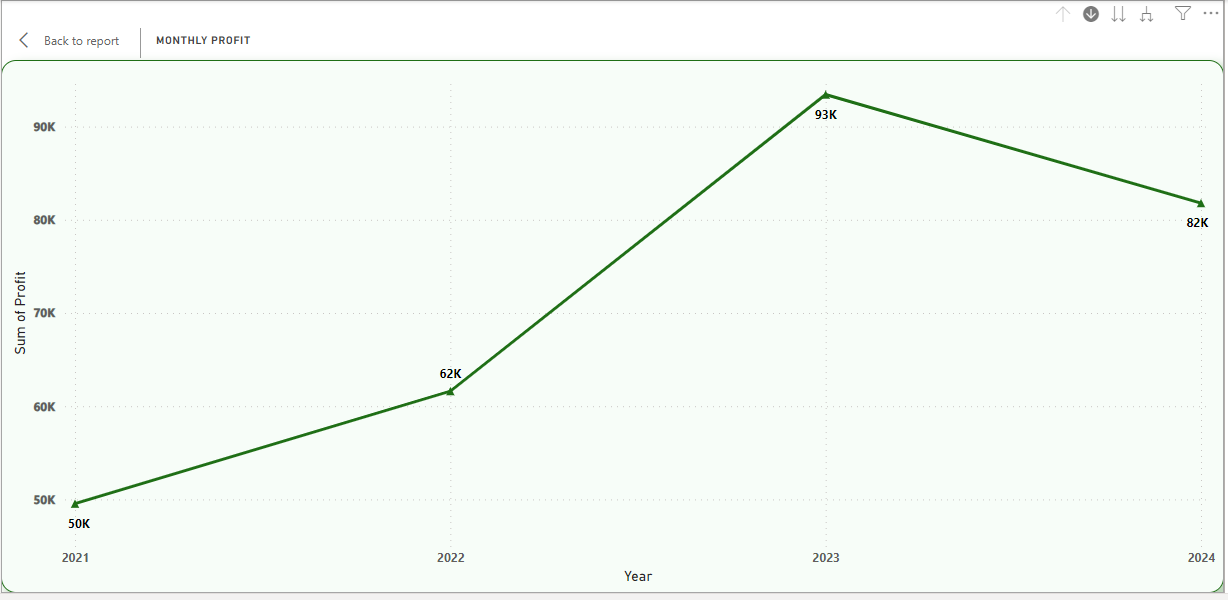
Monthly Profit View



Quarter Profit view

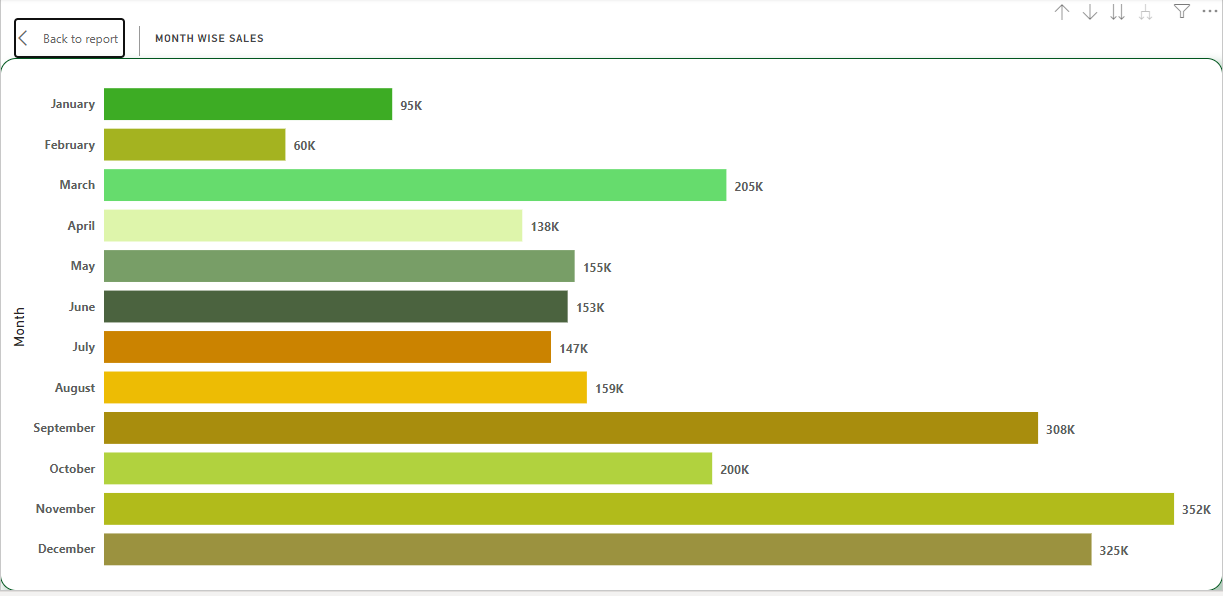


Yearly profit view

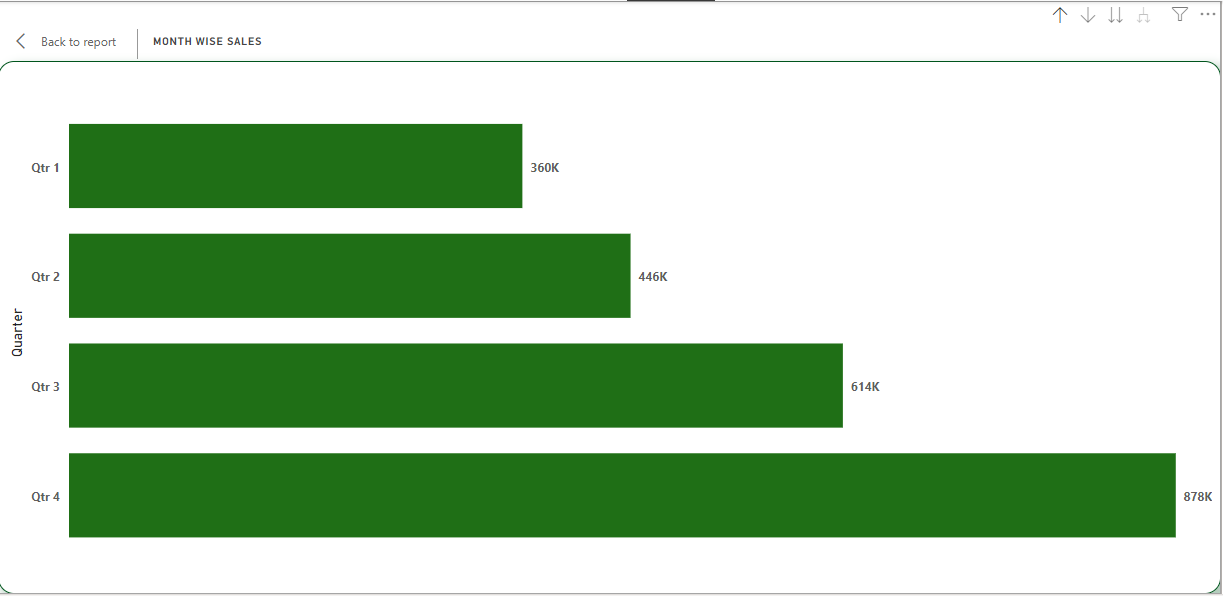


**Sales/ Revenue Summary**

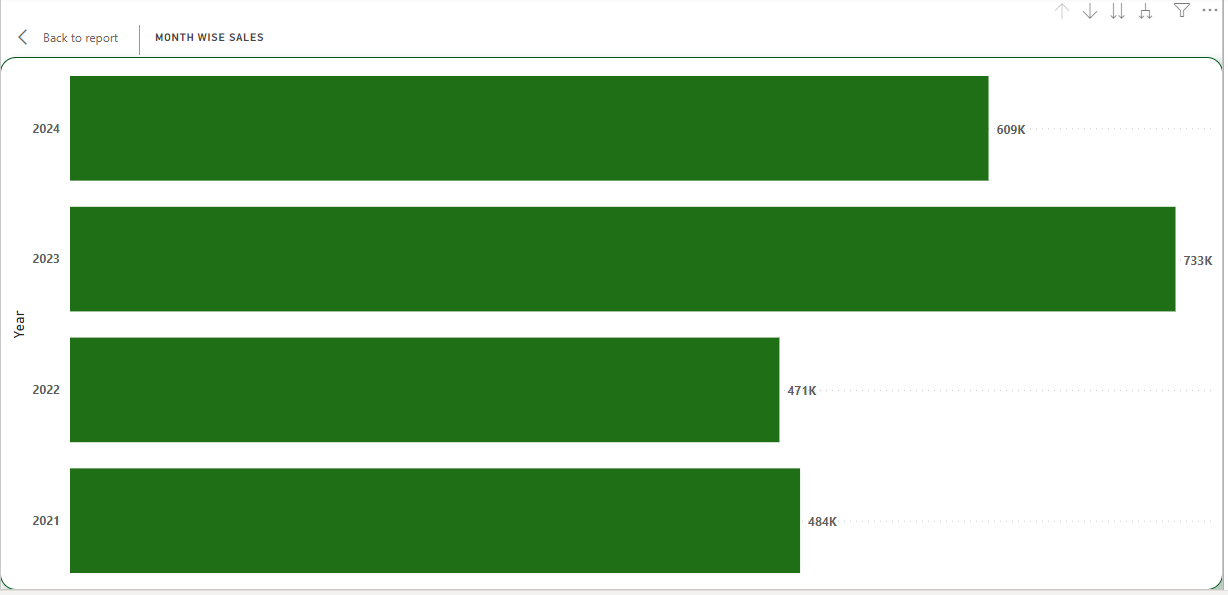
Monthly Sales



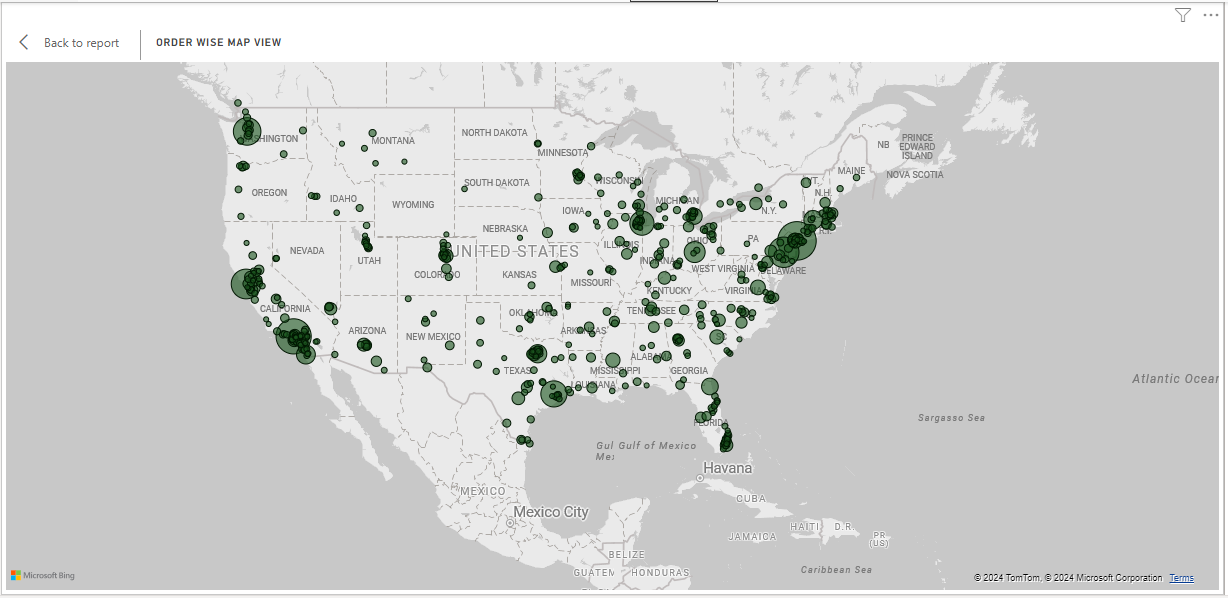
Quarter Sales



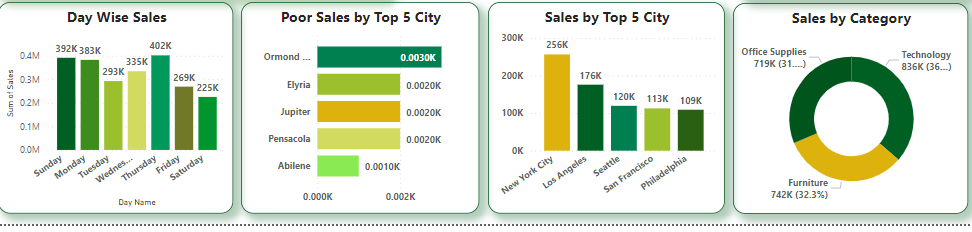
Yearly Sales



Location Wise Sales Map view



Week Day Wise Top Performing Days/ Poor & top performing sales cities, Sales by category.



**Challenges & Learnings**

While building this dashboard, I learned to balance visual complexity with performance optimization, especially when handling large data volumes. Additionally, I explored new features of Power BI, like map visuals, to represent geographic data effectively.

**Future Enhancements**

- Add more granular filtering options for product categories, cities, and customer segments.

- Integrate a forecast model to predict future sales and profit based on historical trends.

- Implement additional drill-down options to analyze individual store performance.

**Setup Instructions**

1. Clone the repository: `git clone <repository-url>`

2. Download and install \*\*Power BI Desktop\*\*

3. Open the Power BI file (.pbix) from this repository.

4. Ensure your SQL Server connection is active to refresh the data.

**Thank You!**