**[Supply Chain](https://datatoinfolabs.com/supply-chain-management-for-car-in-power-bi-dashboard-project/)** [**Management for Car in**](https://datatoinfolabs.com/supply-chain-management-for-car-in-power-bi-dashboard-project/)[**Power BI: Dashboard**](https://datatoinfolabs.com/supply-chain-management-for-car-in-power-bi-dashboard-project/)[**Project**](https://datatoinfolabs.com/supply-chain-management-for-car-in-power-bi-dashboard-project/)

Supply Chain Management for Car in Power BI project includes working of Power BI, SQL, and Python. This project is for all beginners who wants to learn and understand about Power BI end-to-end dashboard project.



# Overview

Supply Chain Management for Cars in Power BI provides a comprehensive overview of the entire supply chain process for automotive companies. This includes tracking the flow of materials, components, and nnished products from suppliers to manufacturers, distributors, and ultimately to customers.

Using Power BI, companies can visualize key supply chain metrics such as inventory levels, production schedules, delivery performance, and supplier quality. They can also analyze historical data to identify trends, optimize operations, and make informed decisions to improve efficiency and reduce costs throughout the supply chain. Using this dashboard project we will try to touch all fundamental topics of data analysis using **Power BI**, **SQL**,and **Python**.

**Flow of the Project**

1. Project Architecture
2. Project BRD or FRD Documents
3. Data Gathering
4. Data Cleaning / Data Transformation
5. Data Modeling
6. Mockup Preparation
7. DAX Functions (DAX Calculations)
8. Create Visuals (For Dashboard)
9. Add Navigation

Project Architecture



**Data Gathering**

According to the BRD document, we have to connect the MySQL database with Kaggle APIs and bring data automatically into database by writing some python code. After then from database to the Power BI Desktop using ‘Get Data’ option. First we will gather and manipulate data using Pandas library in python.

Firstly go to the Kaggle website dataset name, ‘**Supply chain management for Car**‘ For python code we are going to use Jupyter notebook.

*#!pip install pandas #!pip install kagglehub #!pip install numpy #!pip install pymysql*

Below is the python code showcasing how to download dataset from Kaggle to your local and then by using pandas library read the dataset. We also have to put that data into MySQL database using python and then from database to Power BI Desktop using ‘Get Data’ option.

*#pip install pandas*

*#pip install numpy*

*#pip install zipfile*

*#pip install kaggle*

*#pip install pymysql*

*#pip install kagglehub*

In [1]:

**import** pandas **as** pd

**import** numpy **as** np

In [2]:

**import** zipfile

Creating a Api key in kaggle and saving it in the c useer so that when the api is being called it give the response to the system and hence download the dataset with no problem

**Api called the dataset through kagglehub**

In [6]:

**import** kagglehub

*# Download latest version*

path **=** kagglehub**.**dataset\_download("prashantk93/supply-chain-management-for-car")

print("Path to dataset files:", path)

Path to dataset files: C:\Users\RajabhishekAditya\.cache\kagglehub\datasets\prashantk93\supply-chain-management-for-car\versions\1

**Reading the Dataset called by api**

In [8]:

cars **=** pd**.**read\_csv("car\_supply\_chain.csv")

In [11]:

cars

**Getting the information of the dataset present in it with summary like index ,coloumn details , data type, memory useage and non-null count .**

In [14]:

cars**.**info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 1000 entries, 0 to 999

Data columns (total 33 columns):

cars**.**shape *# shapes help in finding how many rows and column are there in the dataset .*

Out[16]:

(1000, 33)

In [51]:

cars**.**columns

Out[51]:

Index(['SupplierID', 'SupplierAddress', 'SupplierName',

'SupplierContactDetails', 'ProductID', 'CarMaker', 'CarModel',

'CarColor', 'CarModelYear', 'CarPrice', 'CustomerID', 'CustomerName',

'Gender', 'JobTitle', 'PhoneNumber', 'EmailAddress', 'City', 'Country',

'CountryCode', 'State', 'CustomerAddress', 'OrderDate', 'OrderID',

'ShipDate', 'ShipMode', 'Shipping', 'PostalCode', 'Sales', 'Quantity',

'Discount', 'CreditCardType', 'CreditCard', 'CustomerFeedback'],

dtype='object')

In [90]:

cars**.**nunique()

Out[90]:

SupplierID 1000

SupplierAddress 1000

SupplierName 362

SupplierContactDetails 1000

ProductID 883

CarMaker 54

CarModel 504

CarColor 19

CarModelYear 47

CarPrice 1000

CustomerID 994

CustomerName 1000

Gender 2

JobTitle 182

PhoneNumber 1000

EmailAddress 1000

City 300

Country 1

CountryCode 1

State 49

CustomerAddress 1000

OrderDate 337

OrderID 993

ShipDate 168

ShipMode 4

Shipping 2

PostalCode 777

Sales 1000

Quantity 2

Discount 66

CreditCardType 16

CreditCard 1000

CustomerFeedback 5

dtype: int64

**EDA**

In [68]:

**import** matplotlib.pyplot **as** plt

**import** seaborn **as** sns

In [70]:

sns**.**set(style**=**"whitegrid")

In [78]:

*# Histogram*

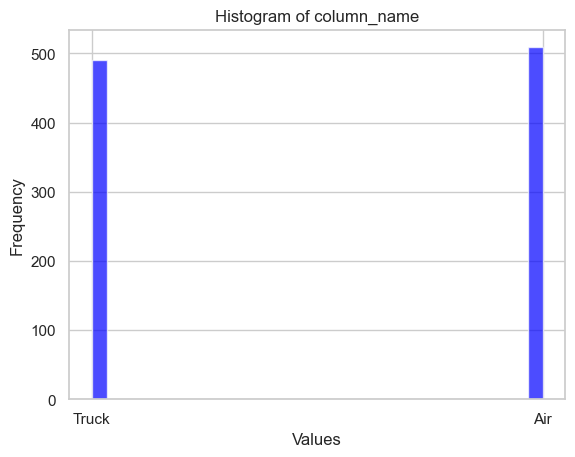
cars['Shipping']**.**hist(bins**=**30, color**=**'blue', alpha**=**0.7)

plt**.**title('Histogram of column\_name')

plt**.**xlabel('Values')

plt**.**ylabel('Frequency')

plt**.**show()

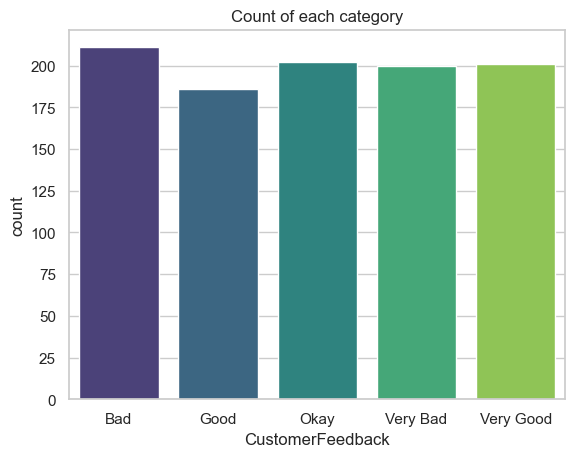


*# Count Plot*

sns**.**countplot(data**=**cars, x**=**'CustomerFeedback', palette**=**'viridis')

plt**.**title('Count of each category')

plt**.**show()



**Now connect to the MySQL database and put all the data into database**

In [33]:

**import** pymysql

**import** numpy **as** np

create a database name project and then use the database in my sql workbench

In [35]:

myconnection **=** pymysql**.**connect(host**=**'localhost', user**=**'root', passwd**=**'Aditya@raj12', database**=**'project') *# 4 patametrs are needed to be provided so that we get connected to the sql worlbench database name as cars*

In [37]:

cur **=** myconnection**.**cursor() *#this is used to establised the connection between the workbench and dataset*

In [39]:

num\_cols **=** len(cars**.**columns) *# counts how many columns (like headers in a table) are in the cars dataset.*

placeholders **=** ','**.**join(['%s'] **\*** num\_cols) *#Joinning the coloumns from csv file to sql table and This is often used in SQL queries to match the number of columns dynamically.*

In [41]:

cur**.**execute("create table cars (SupplierID text, SupplierAddress text, SupplierName text, SupplierContactDetails text, ProductID text, CarMaker text, CarModel text, CarColor text, CarModelYear text, CarPrice text, CustomerID text, CustomerName text, Gender text, JobTitle text, PhoneNumber text, EmailAddress text, City text, Country text, CountryCode text, State text, CustomerAddress text, OrderDate text, OrderID text, ShipDate text, ShipMode text, Shipping text, PostalCode text, Sales text, Quantity text, Discount text, CreditCardType text, CreditCard text, CustomerFeedback text)") *# The cur.execute command is used in Python to execute a SQL query through a cursor object when working with a database.*

Out[41]:

0

In [43]:

sql **=** "INSERT INTO cars VALUES ({}) " **.**format(placeholders) *# inserting all the values from csv file to sql table where the table name is cars*

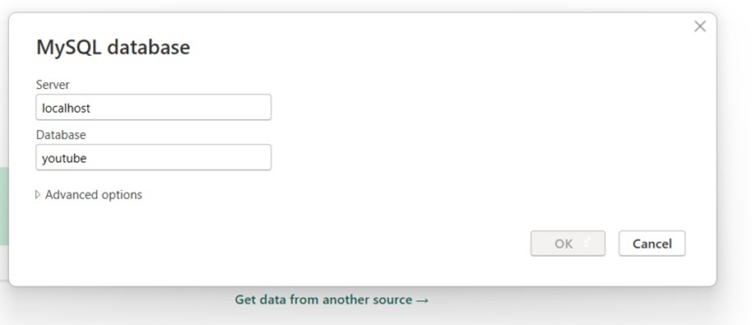
In [45]:

**for** i **in** range(len(cars)):

cur**.**execute(sql, tuple(cars**.**iloc[i]))

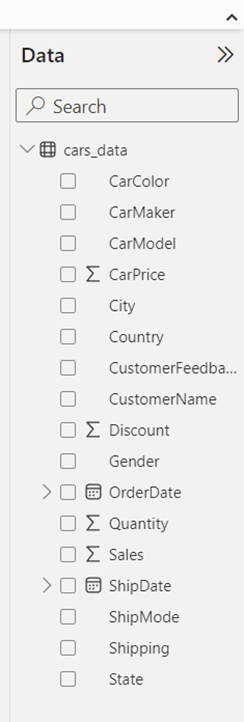
myconnection**.**commit() *#The code is inserting rows of data from a DataFrame (cars) into a database. Each row in the DataFrame is processed one at a time, and the data is added to the database.*

Use get data option to bring data into Power BI.



Data Cleaning :

Let’s transform data on Power Query Editor, we need to remove some unnecessary attributes from the table and just keep the important columns which we can use for visualization. Below is the list of columns we are gonna work on, and rest of it has removed.



Apart from ‘cars\_data’ table, we need to create a separate Calendar table (DateMaster). It is always a good practice to have a separate date table for time intelligence functions.

DateMaster = CALENDAR(MIN(cars\_data[OrderDate]), MAX(cars\_data[OrderDate]))

OrderMonth Num = MONTH(DateMaster[OrderDate])

OrderMonth Name = FORMAT(DateMaster[OrderDate], "MMM")

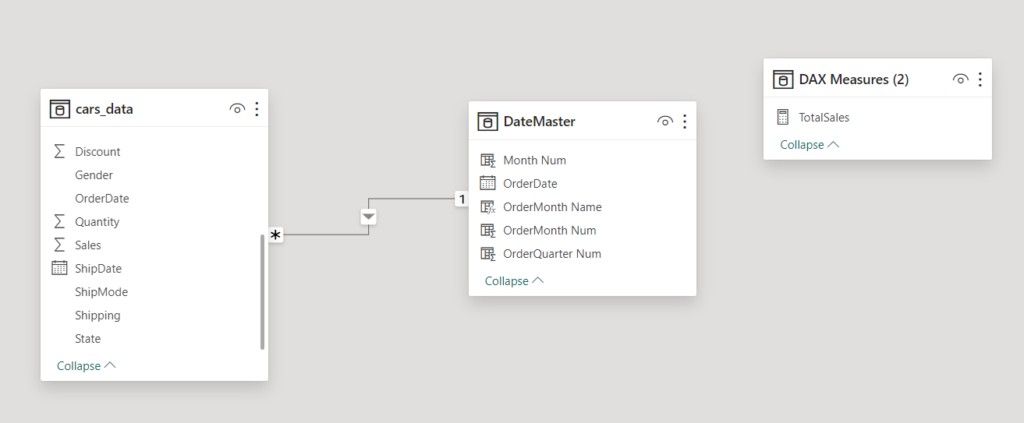
OrderQuarter Num = QUARTER(DateMaster[OrderDate])

Month Num = DateMaster[OrderDate].[MonthNo]

Modeling In Power BI

After creating the date master table, establish the relationship between these two tables using modelling. Go to the model view tab and create one-to-many relationship between date columns in both the tables. Also create a new measure for ‘Total Sales’ and keep all the new measures in separate table called ‘DAX Measures’.

TotalSales = SUM(cars\_data[Sales])

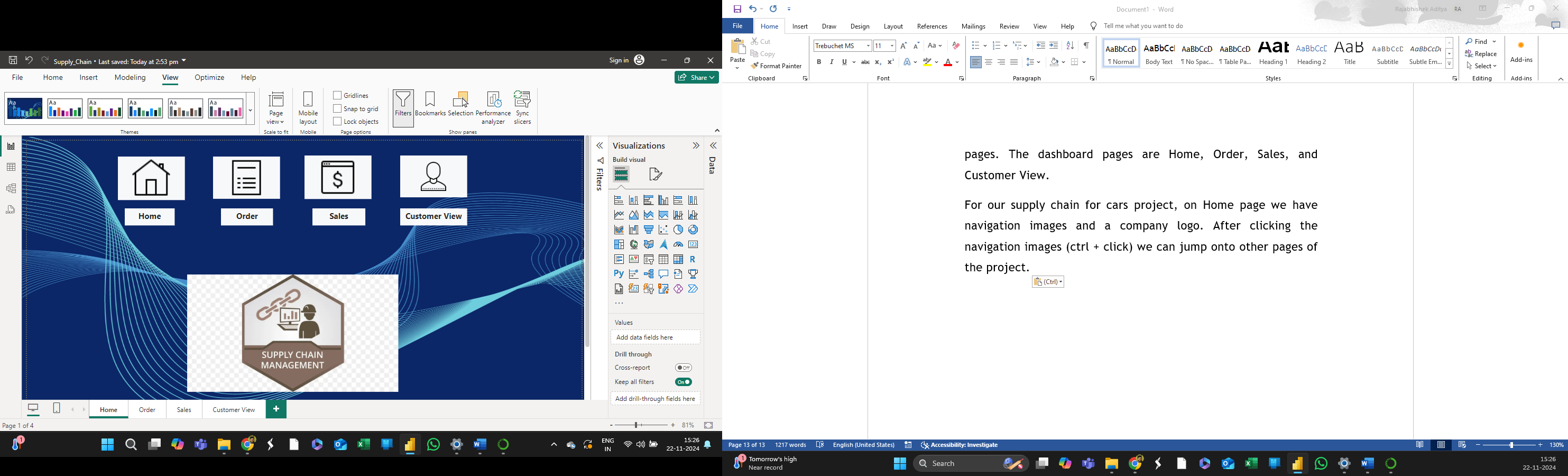


Before start creating the dashboard, we must spend some time on creating a prototype or mockup preparation. It is always a good practice to draw and imagine what to create on pen and paper nrst.

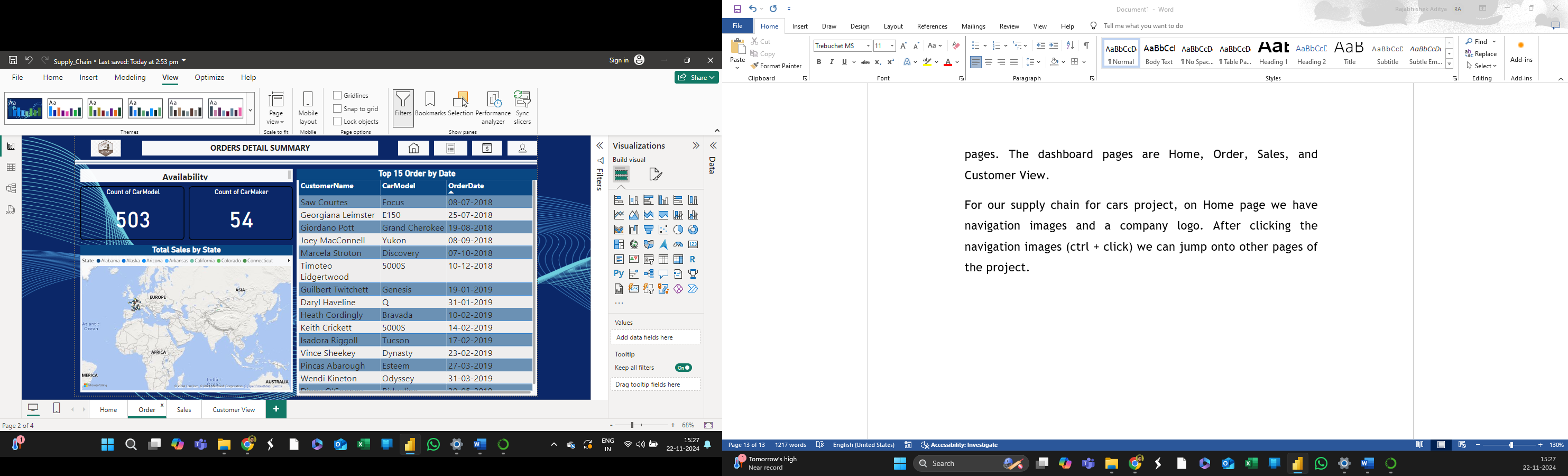
# Dashboard Creation In Power BI

In our dashboard project, we are gonna create four different pages and implement navigation between all these four dashboard pages. The dashboard pages are Home, Order, Sales, and Customer View.

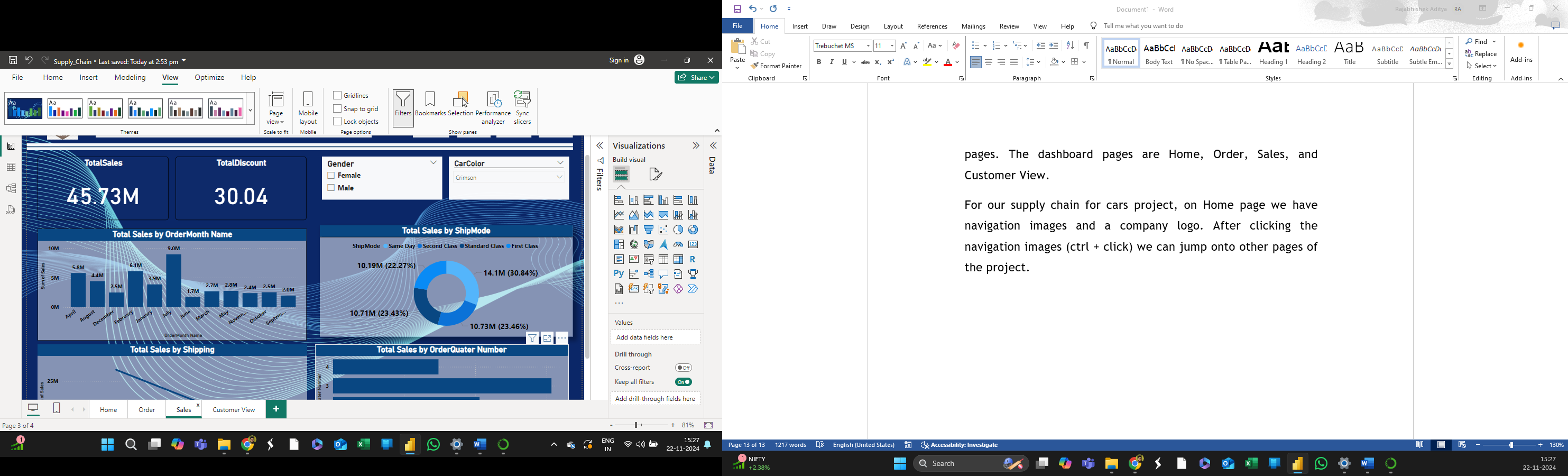
For our supply chain for cars project, on Home page we have navigation images and a company logo. After clicking the navigation images (ctrl + click) we can jump onto other pages of the project.



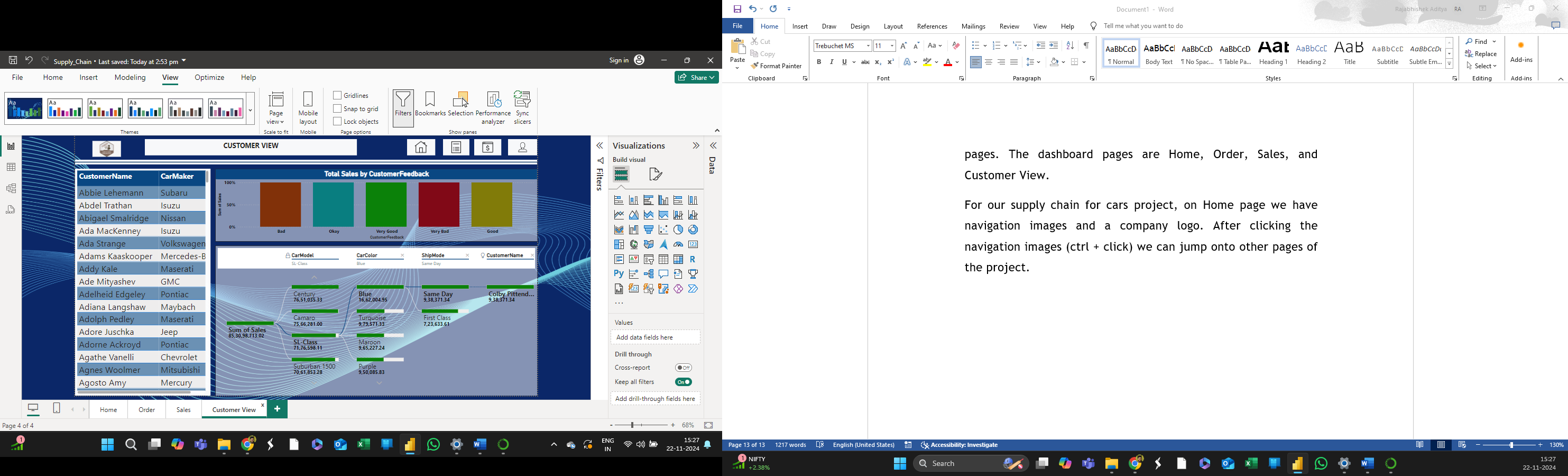
According to the BRD document, on **Order** page we have to show details related to the cars order. Show the availability of cars model and car makers. Also show the top 10 to 15 orders by sales, and nnally create a map chart visual for sales or by state neld. We have also added the new visual called scroller on top to show some car price by car model.



On **Sales** page according to requirement, show the total sales and discount values on card visuals. Also sales by month, sales by shipping, sales by ship mode, and sales by quarter using visuals column chart, line chart, donut chart, bar chart respectively. Here we added two slicers as well using **Gender** and **CarColor** columns which will act as a nlter.



On **Customer view** page we have to show data related to the customers like show some sentiment analysis values using column chart visual. Customer names on a table visual. Also we have to create a drill down view on our supply chain model using decomposition tree visual.



Don’t forget to include the navigation images on every dashboard page. Now you have complete supply chain dashboard project with you. Please refer to the provided video for more detailed understanding. I hope you guys must have liked the dashboard project of supply chain management for cars in Power BI.

Overall, Supply Chain Management for Cars in Power BI empowers automotive companies to gain insights into their supply chain operations, improve collaboration with suppliers and partners, and ultimately deliver better products and services to customers.