USE CASE STUDY REPORT

Group 3

Aarthi Thinakaran and Venkata Krishnan

Executive Summary:

The total value of the US car and automobile manufacturing market is $100.9 billion in 2022, with a production of 9.2 million vehicles in 2021.The United States motor vehicle and parts dealers is a trillion dollar industry with a total revenue of $1.53 trillion as of 2021. With such a revenue, the auto industry accounts for 3% of America's GDP.

The idea behind this study is to understand the needs of automobile spare products in the market, identify the gaps in sales and the money spent by customers through unauthorized sources which we call "middlemen" for their automobile spare products. Hence, to overcome this, a solution was provided by building a Dealer Management System database for Automobile Products Booking System, which cuts the cost of money spent on middlemen, where all automobile spare products are brought under one roof. Through this method, consumers get to buy genuine products directly from the manufacturer, which are value for money. Through our database, consumers can buy any spare products across all manufacturers. A margin of 35% imposed by the middlemen on the consumers can be neglected and genuine products can be induced in the market. The market size measured by revenue of the Auto Parts Manufacturing industry is $62.2bn in 2022, which is expected to expand immensely in the upcoming years.

The database was designed in a way that it records information from both the customer and the automobile dealers of multiple car brands. For each order made by the customer, the order details are stored in the Customer Bookings table from which details can be retrieved if the customer wants information. Each one of the products ordered is connected to the product specifications in the Spare Products table and are sent to the Automobile Dealers.

The conceptual data model for the database were designed through EER and UML diagrams. These diagrams are then mapped to logical data model which is a relational data model with all the necessary relations, primary keys and foreign keys. The relational model is then implemented through MySQL. Two of the relations were implemented on MongoDB to check the level of viability of this database in NoSQL platform. The MySQL database is then integrated to Python through PyCharm IDE. This provided the ability to perform some analytics on the data in the database which has been presented through several plots in this project. The MySQL queries and analysis aids to know the trends of sales and this system's impact on the customers from different perspectives such as on the basis of car brands, product type based and also help visualize the seasonality trends if any.

Introduction:

The Automobile industry has elevated its benchmark so high with the introduction of electric cars and its market is booming exponentially every day. Vehicles, especially cars have become one of the necessities of life, and although most people feel positive about buying a vehicle and believe it to be a very valuable investment, for a part of the population the maintenance seems to outweigh the cost for the profit of making this investment.

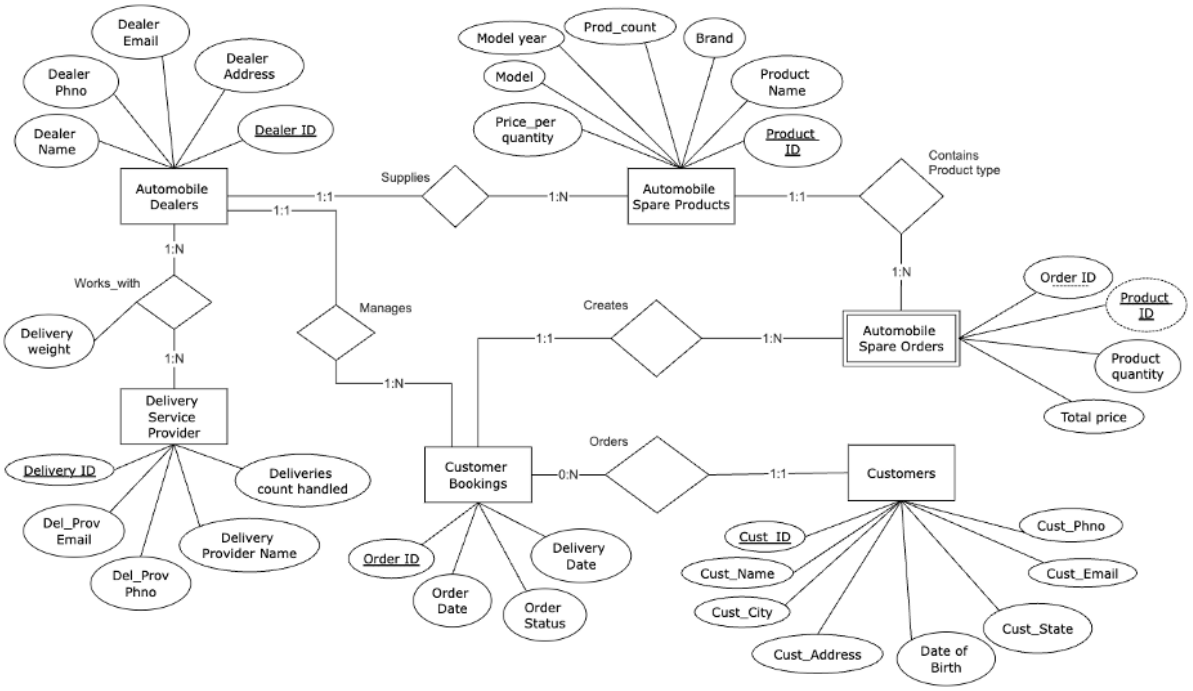
Wear and Tear is a very common practice in the case of Automobile Parts. However, replacements of such parts is nothing unusual, the money spent on this accounts annually for a huge price. Even though they may not seem as a recurring cost or high end pricing, U.S. Auto Parts revenue is $443.9M annually. In a billion dollar industry, genuine parts net profit margin as of September 30, 2022 is 5.55%. Thus 5.55% cannot be avoided and has to come from the pockets of customers imposed by the dealers.

One of the common difficulties every automobile consumer faces is wear and tear of their vehicle's spare parts. The problem of car maintenance costs being expensive has been exacerbated and it brings a need to analyze this issue and find a solution. The issue doesn't stop there. There has been a flow of non-genuine products in the market which places a huge distrust for the consumers on the manufacturers for poor quality spare parts. In most cases, the maintenance costs get expensive because people get their replacement parts through sources that are not authorized by the automobile brand. At times, unethical practices of high prices are imposed where the profit margin of the middlemen is further increased. Geographical disadvantages are still an issue. Consumers from certain remote areas still face the availability issues due to shortage of automobile spares. The market for certain automobile spare products die along with the vehicle's existence in the market.

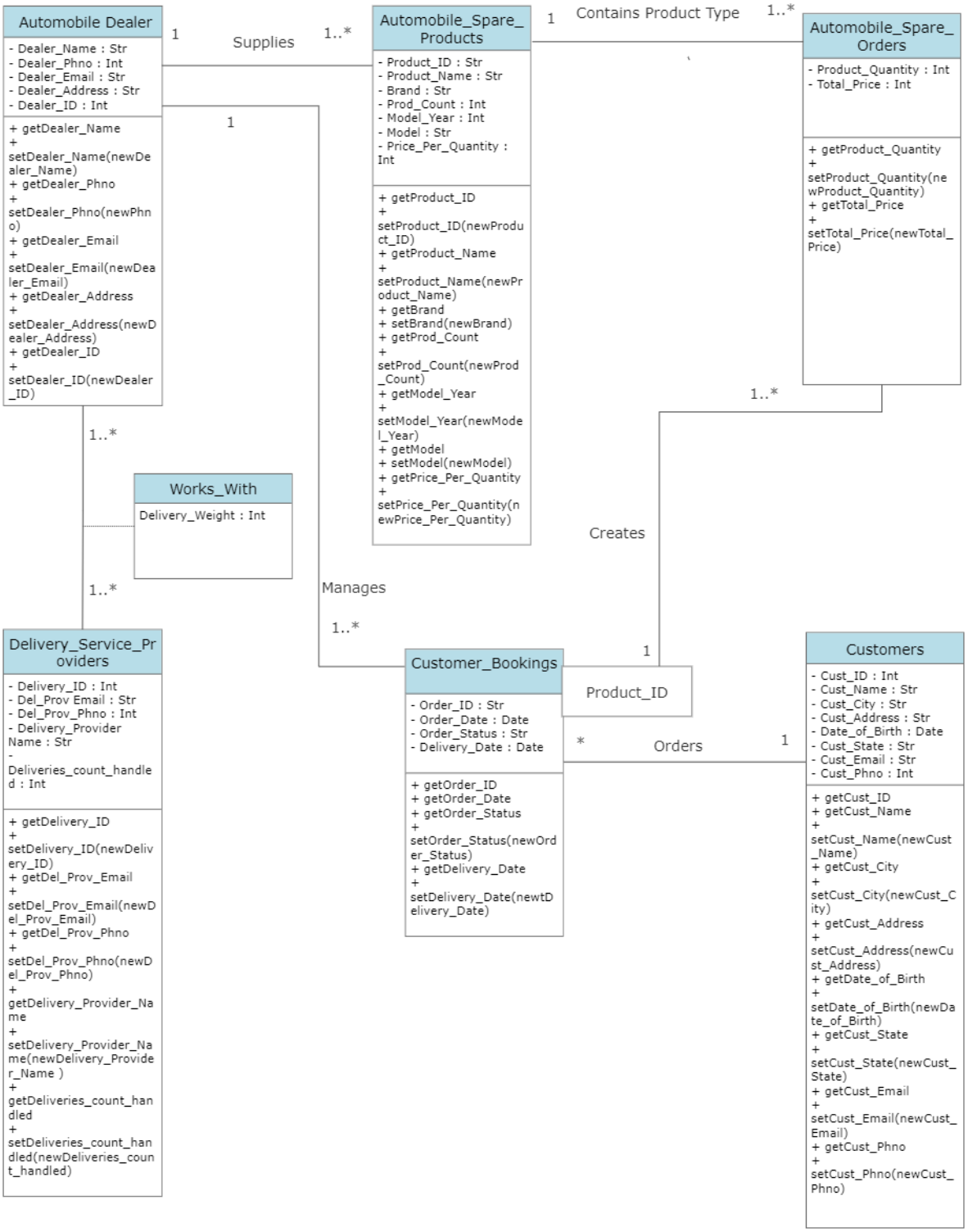
These factors all lead to customer dissatisfaction, which may in turn lead to decrease in sales of specific automobile manufacturers. Hence to overcome all these issues, a relational database model is built, so that there is a direct relationship maintained between the manufacturers and consumers with respect to the automobile spare parts. This relational database accounts for the storage of the automobile manufacturer details along their products. Through this, customers can access genuine spare parts with reasonable pricing directly from the manufacturer. The added benefit of the Automobile Products Booking System is that all automobile dealer's spare parts are available under one roof, under one database. Hence, customers are given the flexibility and advantage of choosing any automobile spare product from the authorized dealer of any car brand in a single integrated place.

CONCEPTUAL DATA MODELING:

1. EER DIAGRAM :



2. UML DIAGRAM:



LOGICAL DATA MODELLING:

Notation:

Solid underline - Primary key
Italics - Foreign key

- Customer(CustomerID, Customer_Name, Customer_Address, Customer_City, State, Ph_No, Date_of_birth, Cust_email)

CustomerID - Primary key of Customer relation

- Automobile_Dealer(DealerID, DealerName, DealerAddress, Contact_number, Dealer_Email)

DealerID - Primary key of Automobile_Dealer relation

- Delivery_Service_Provider(Delivery_ID,Delivery_Provider_Name,Delivery_Provider_Phno, Delivery_Provider_Email, Deliveries_count_handled)

DeliveryID - Primary key of Delivery_Service_Provider relation

- Works_with (*DealerID ,DeliveryID ,* Delivery_weight)

DealerID - Foreign key referring to primary key DealerID of Automobile_Dealer relation; NULL NOT ALLOWED
DeliveryID - Foreign key referring to primary key DeliveryID of Delivery_Service_Provider relation ; NULL NOT ALLOWED

- Customer_Bookings (OrderID, Order_Date, Delivery_Date, Order_Status, *CustomerID*, *DealerID)*

OrderID - Primary key of Customer_Bookings relation
CustomerID - Foreign key referring to primary key CustomerID of Customer relation ; NULL NOT ALLOWED
DealerID - Foreign key referring to primary key DealerID of Automobile_Dealer relation ; NULL NOT ALLOWED

- Spare_Products (ProductID, Product_name, Car_Brand, Car_Model, Car_Model_year, *DealerID,* Product_count, Price_per_quantity)

ProductID - Primary key of Spare_Products relation
DealerID - Foreign key referring to primary key DealerID of Automobile_Dealer relation ; NULL NOT ALLOWED

- Spare_Orders (*OrderID*, *ProductID*, Prod_quantity, Total_price)

OrderID - Foreign key referring to primary key OrderID of relation Customer_Bookings ; NULL NOT ALLOWED
ProductID - Foreign key referring to primary key ProductID of relation Spare_Products ; NULL NOT ALLOWED

IMPLEMENTATION OF RELATIONAL MODEL via MySQL:

MySQL Implementation:

The automobile backup products database was created in MySQL and the below queries were executed.

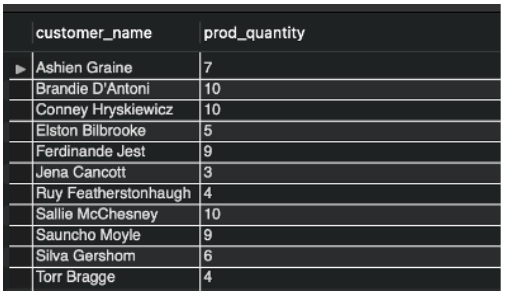
Query 1: Find Car Brands and Car Models with automobile spare products with unit price greater than 100.

```
SELECT car_brand,car_model
FROM spare_products
WHERE price_per_quantity>100
ORDER BY car_brand;
```

car_brand	car_model
Buick	Maybach GLS-SUV
Jaguar	XF
Lamborghini	Diablo
Mercedes-Benz	S-Class
Mercedes-Benz	C-Class
Mercedes-Benz	G-Class
Mercedes-Benz	GLA SUV
Mercedes-Benz	CL-Class
Mercedes-Benz	EQS-450
Mercedes-Benz	A-Class

Query 2: Find Customer names and quantity of automobile part ordered by people who ordered 'Clutch'.

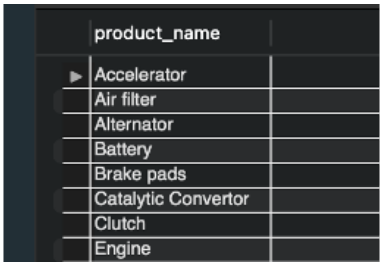
```
SELECT c.customer_name,o.prod_quantity
FROM customers c,customer_bookings cb,
spare_orders o,spare_products p
WHERE c.customer_id = cb.customer_id
AND cb.order_id = o.order_id
AND o.product_id = p.product_id
AND p.product_name = 'Clutch'
ORDER BY c.customer_name ASC;
```



customer_name	prod_quantity
Ashien Graine	7
Brandie D'Antoni	10
Conney Hryskiewicz	10
Elston Bilbrooke	5
Ferdinande Jest	9
Jena Cancott	3
Ruy Featherstonhaugh	4
Sallie McChesney	10
Sauncho Moyle	9
Silva Gershom	6
Torr Bragge	4

Query 3: Find Automobile parts that have ordered count of more than 5.

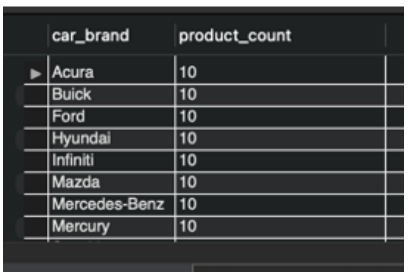
```
SELECT DISTINCT product_name
FROM spare_products
WHERE product_id IN (SELECT product_id
                     FROM spare_orders
                     WHERE prod_quantity>5)
ORDER BY product_name ASC;
```



product_name
Accelerator
Air filter
Alternator
Battery
Brake pads
Catalytic Convertor
Clutch
Engine

Query 4: List the top 3 DISTINCT Car Brand of the Automobile product part which have been ordered the highest. (Using Correlated query)

```
SELECT DISTINCT p1.car_brand,p1.product_count
FROM spare_products p1
WHERE 3 >(SELECT COUNT(DISTINCT product_count)
          FROM spare_products p2
          WHERE p2.product_count > p1.product_count)
```



car_brand	product_count
Acura	10
Buick	10
Ford	10
Hyundai	10
Infiniti	10
Mazda	10
Mercedes-Benz	10
Mercury	10

ORDER BY p1.product_count DESC;

Query 5: Find the product number, automobile product name, automobile product brand and number of days of the orders which took more than 30 days to deliver it.

```
SELECT p.product_id, p.product_name,
p.car_brand,
DATEDIFF(delivery_date,order_date) AS
days_delivered_in
FROM customer_bookings c, spare_orders o,
spare_products p
WHERE c.order_id = o.order_id
AND o.product_id = p.product_id
AND DATEDIFF(c.delivery_date,c.order_date) > 30;
```

product_id	product_name	car_brand	days_delivered...
S2	Accelerator	Toyota	44
S3	Clutch	Ford	46
S20	Brake pads	Mazda	35
S36	Engine	Volvo	38
S50	Alternator	Toyota	40
S51	Battery	GMC	40
S52	Shock absorber	Chevrolet	40
S82	Wiper blades	Chevrolet	41

Query 6: Find the Delivery Service Provider who has made the highest deliveries along the weight of delivery product they handle.

```
SELECT d.delivery_id, d.delivery_provider_name,
w.delivery_weight
FROM delivery_service_providers d,works_with w
WHERE deliveries_count_handled >= ALL
(SELECT deliveries_count_handled
FROM delivery_service_providers )
AND d.delivery_id = w.delivery_id;
```

delivery_id	delivery_provider_na...	delivery_weight
1038	Regrant	192

Query 7: Find the customers whose names starts with ‘A’ and have outstanding orders.

```
SELECT DISTINCT C.customer_id,C.customer_name
FROM customers AS C
WHERE C.customer_name LIKE 'A%'
AND EXISTS (SELECT *
                FROM customer_bookings as CB
               WHERE C.customer_id = CB.customer_id);
```

customer_id	customer_name
2	Anica Fogt
4	Amandi Eaklee
30	Ailbert Wanka
31	Ardella Denington
32	Ashien Graine
49	Austina Sorel
61	Adelice Gawne
68	Adan Kensy

Query 8: Find customers who have born after 1990 or who stay in ‘San Francisco':

```
SELECT customer_name
FROM customers
WHERE YEAR(date_of_birth) > 1990
UNION
SELECT customer_name
FROM customers
WHERE customer_city LIKE 'San Francisco';
```

customer_name
Currey Nottle
Sallie McChesney
Valene Adrianello
Evan Vasyuchov
Milo Beckitt
Ashien Graine
Alice Kemball
Edwin Menego

IMPLEMENTATION OF RELATIONAL MODEL via NoSQL:

NoSQL Implementation:

Two tables Automobile Spare Products and Customer Bookings have been created in MongoDB Compass. The following queries were executed:

Query 1: Find Car Brands and Car Models of the automobile spare products unit price greater than 100.

```
db.Automobile_Spare_Products.find(
        {price_per_quantity : {$gt : 100}},
        {car_brand : 1, car_model :1, _id : 0}
)
```

Automobile_Spare_Products

	car_brand String	car_model String
1	"Mercedes-Benz"	"CL-Class"
2	"Mercedes-Benz"	"S-Class"
3	"Rolls-Royce"	"Phantom"
4	"Mercedes-Benz"	"C-Class"
5	"Jaguar"	"XF"
6	"Mercedes-Benz"	"G-Class"
7	"Mercedes-Benz"	"GLA SUV"
8	"Lamborghini"	"Diablo"
9	"Mercedes-Benz"	"EQS-450"
10	"Mercedes-Benz"	"A-Class"
11	"Buick"	"Maybach GLS-SUV"

Query 2: Find the automobile spare products sales of each Car Brands.

```
db. Automobile_Spare_Products.aggregate(
[{
        $group: {
        _id: '$car_brand',
        Product_count_ordered: {
        $sum: '$product_count'
        }
        }
}, {
        $sort: {
        Product_count_ordered: 1
        }
}]
)
```

```
_id: "Studebaker"
Product_count_ordered: 3

_id: "Rolls-Royce"
Product_count_ordered: 3

_id: "Maybach"
Product_count_ordered: 3

_id: "Cadillac"
Product_count_ordered: 4

_id: "Lamborghini"
Product_count_ordered: 4

_id: "Plymouth"
Product_count_ordered: 5

_id: "Nissan"
Product_count_ordered: 7

_id: "Lotus"
Product_count_ordered: 7
```

Query 3: Find the monthly sales of the automobile spare products orders

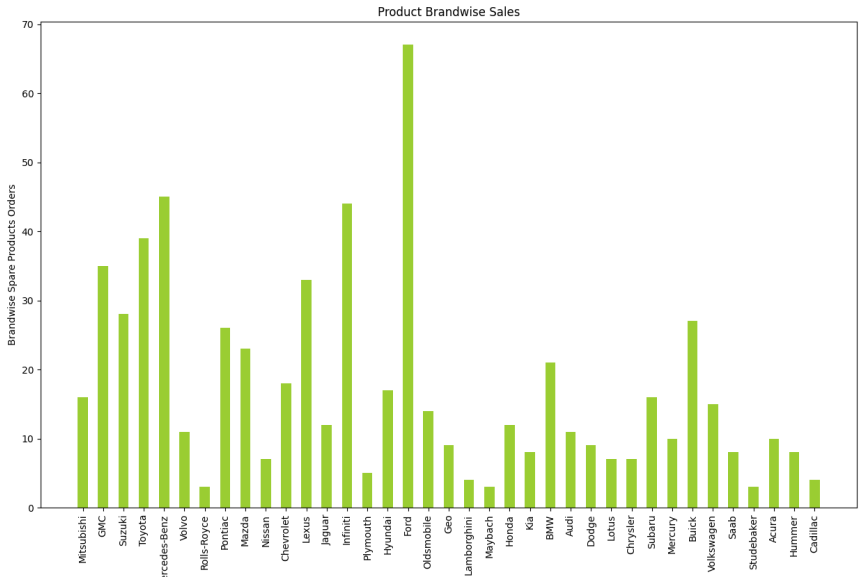
```
db.Customer_Bookings.aggregate(
[{
        $group: {
        _id: {
        $month: '$order_date'
        },
        Monthly_product_orders: {
        $count: {}}
        }
}])
```

```
_id: 8
Monthly_product_orders: 25

_id: 10
Monthly_product_orders: 33

_id: 9
Monthly_product_orders: 32
```

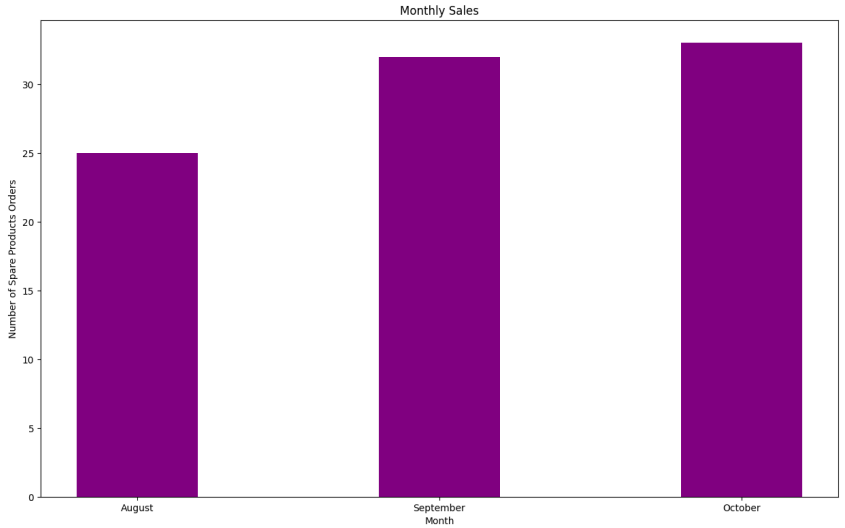
Database Access via Python:

The database data we analyzed through MySQL queries are visualized using Python integration to MySQL database.The connection of Python to MySQL Workbench is done using PyCharm with mysql.connector . Then the cursor created is used to extract the data and it is executed using cursor.execute along fetchall() command to obtain the desired subset of data. The following results are then plotted as graphs using the matplotlib library from Python.

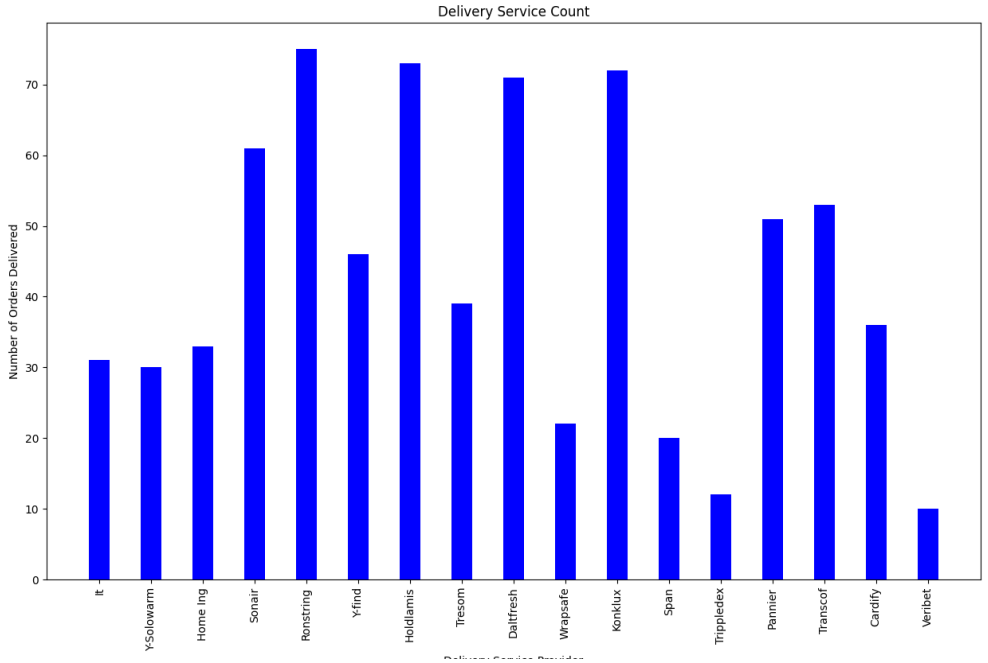
Plot 1: Number of Products Sales in each Automobile Car Brand



Plot 2: Product monthly sales



Plot 3: Number of deliveries handled by each delivery service provider



CONCLUSION AND RECOMMENDATION:

The real world implementation of our Dealer Management System in the Automotive Industry would be a huge benefit for both parties, such as consumers and manufacturers. Accessibility and reliability was a huge barrier for consumer-manufacturer relationships. Through our database, Automobile Spare Products can be genuine, and priced right. Costs can be cut down to the actual manufacturing price. Interesting insights for the aforementioned have been visualized using Python. The relationships between sales, orders and products have been denoted.

Though our database has multiple benefits, it comes with its own disadvantages. The Delivery Service Provider can only deliver ordered products from a single Manufacturer/Dealer at a moment. Likewise, consumers can place an order from only one manufacturer at a time.

Further developments can be made to our database to make it even more user-friendly. Returns and Exchanges play a major role in e-commerce. Same scenario applies with our Dealer Management System as well. The transaction history details can be stored in our database for future payment convenience and suggestions for our consumers.