VEHICLE MANAGEMENT SYSTEM

Harshan Ganugula
ganugula.h@northeastern.edu

Firdose Shaik
shaik.fi@northeastern.edu

Jenvith Manduva
manduva.j@northeastern.edu

Abstract:

The Vehicle Management System will assist showroom dealers in keeping track of their employee's pay as well as the sales of their vehicles, client information, and revenue. Employees have access to update vehicle data and make modifications as needed. In addition, our web application will assist consumers in purchasing vehicles, scheduling appointments, requesting vehicle parts, and providing reviews. Customers can read the reviews and choose a vehicle in the most informed way possible. And to filter the inappropriate words from the sentences we applied Sentiment Analysis using Machine Learning.

1. Introduction

At least 70% of the customers select their automobiles based on customer service convenience which indicates service convenience plays a significant role in the sales of the vehicles. So, we have created a vehicle management system that will automate the existing manual system by computerizing the data with easy access and manipulation.

However, after-service management also plays a vital role to ensure the long-term loyalty of customers. So, this will help the organization in better utilization of resources which will eventually assist the vehicle dealer in reducing the manual work and delivering a high level of service and support to the customers.

This vehicle management system facilitates effective services in processing the customer's vehicle reservation orders and services, it also includes customer requests for vehicle scheduling and servicing, organizing sales data in an economic order, storing, and retrieving reviews given by the customers, and assessing that data to predict the sentiment of the customer on the vehicle he bought which eventually helps in narrowing the flawless in the service. The user-friendly interface provides ease of use, and the integration technology establishes an effortless and hassle-free business process into Vehicle history files, customer details, employee records, sales data produced by each branch, and the service record of a vehicle.

Service monitoring and analysis increase Visibility altogether in fixed Operations and increase the service Capacity competence and decrease operating costs.

2. Objective

The objective of our system is to arrange a platform to enable showroom dealers to maintain the records of the employees, their salaries, and sales of the vehicles, which helps in forming tighter relations with the customers and providing a high level of service to the customers after the sales.

Problems with existing System:

1. Lack of Data security
2. Manual work
3. Lots of time-consuming
4. Cannot access/manipulate the data easily
5. Lack of customer service
6. No proper business decision (Revenue and sales of vehicles).

Our proposed system solves the following:

1. High-level data security with login credentials for both customers and showroom dealers
2. Reduces the manual work (data entry) and computerizes it
3. Less time-consuming with easy access
4. Data can be easily accessed and manipulated by the backend users.
5. Provides high-level customer service with the web application we have created in which customers can order vehicles, and vehicle parts, book appointments, and provide reviews about the product
6. Our proposed system gives a detailed overview of the sales, employee salaries, and revenue generated, which helps in taking accurate business decisions.

Software Requirements:

- Operating System: Windows,
  macOS

- Software: Jupyter Notebook,
  Visual Studio Code,
  MySQL
- Programming language: Python
- Web-Technology: Flask, Selenium
- Back-End: MySQL Server
- Libraries: Textblob, Plotly,
  Profanity

3. Data Collection

Our data is in the form of tables. Each table includes a minimum of 12 rows of data. Vehicles and services data was collected through Web Scraping using selenium from third-party vehicle websites and the user's data is populated using user FakeAPI endpoints, and for the rest of the tables, the data was populated manually.

The data needs to be cleaned because it contains many null or duplicate values before being converted to the format that was required when the tables were created.

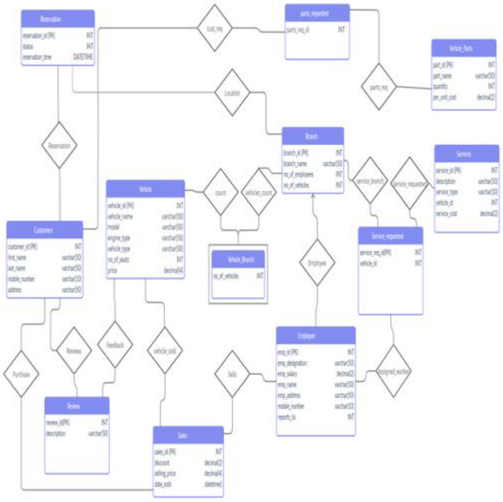
4. Database Design

The system will keep a record of the information regarding the particulars of the vehicle data, which will then be used to populate data on the web. The manager of the branch can add new records, update existing records, and delete records from the tables that are associated with his branch.

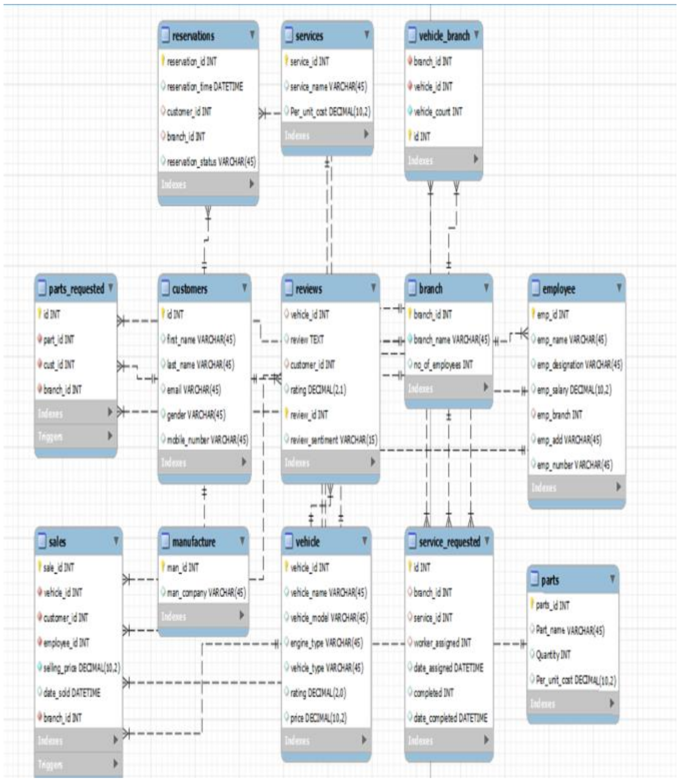
Customers can book their appointment online to visit the branch. A form is provided for users to book an appointment. And managers will update the status of the appointment according to the availability of the employees. Whenever a service is added to the table, an employee other than the manager gets assigned to the task automatically based on the employees based on the data.

In addition, we will design a survey for customers to fill out and submit their feedback regarding a specific vehicle to assist them in making an informed purchase decision. We are going to use a stored procedure to query the data to analyze the performance of each worker and location. Also, we have implemented a trigger that will increase the salary of an employee based on the performance of the sales. If the performance of that employee meets certain requirements in which will also be determined by the services they have provided, it will be rated high automatically.

The system will also keep track of the insurances that are opted for by customers while buying a vehicle. This data will be used for recommending new customers who are looking for vehicle insurance. In addition, we will set up a trigger that will activate if any of our customers' insurance policies are close to reaching their expiration dates.



Schema -Diagram



ER-Diagram

The database system consists of 12 tables.

1. The **Vehicle Table** consists of detailed documentation of the vehicle model and name, engine type, and price of each vehicle and holds the rating of every vehicle type which was eventually given by the customers who bought it and used it for some time on a scale of 1 to 5 where 1 being the lowest and 5 being the highest.

2. The **Branch Table** consists of comprehensive information about each branch and the employees working in a particular branch.

3. The **Vehicle_Branch** table holds data for each branch that the number of vehicles in that branch.

4. The **Services Table** has the data on various types of services that would be accomplished in a particular branch and the details of the pricing of that service.

5. The **Parts** table contains details of all the names of the parts available in a specific branch and the quantity it holds along with the per unit cost of that part_id.

6. The **Reservation table** will preserve the data of customers who book an appointment at a particular branch where a manager can approve/decline the request as per the employees working on that day. It also stores the data of the customer who has requested a reservation of the vehicle, and at which branch the reservation has been requested along with the reservation status.

7. The **Employee table** keeps the data of the employee's name, designation of an employee he held in a particular branch, the salary was given to a particular employee, and the branch they are working under. Each branch has a manager who has access to update, delete, and add data to different tables in their branch.

8. The **Sales table** consists of data on all the vehicles sold with the information of who sold the vehicle, how much they sold it for, and in which branch the vehicle was sold. This data will be later used to look into individual performances of an employee and branch.

9. The **Parts_requested** table has the data of an id of a particular part and details of the customer who requested it along with the branch details where it has been requested for.

10. **The Manufacturer** table consists of the name of a manufacturer who manufactured the vehicle in the dealer’s showroom and an id will be given to each manufacturer.

11. **The Service Requested** table has all the information about each service provided by a branch like a branch where the service has been requested. It also has employee details who has

been appointed by the manager to oversee the service. It even holds the date of the service assigned and the completed date of it.

12. The Reviews table holds the documentation of the reviews given by a customer for a vehicle, details of the customer who has given the review and rating column for the vehicle and has included a sentiment analysis tool Textblob to analyze the sentiment of the customer based on the reviews drafted by the customers.

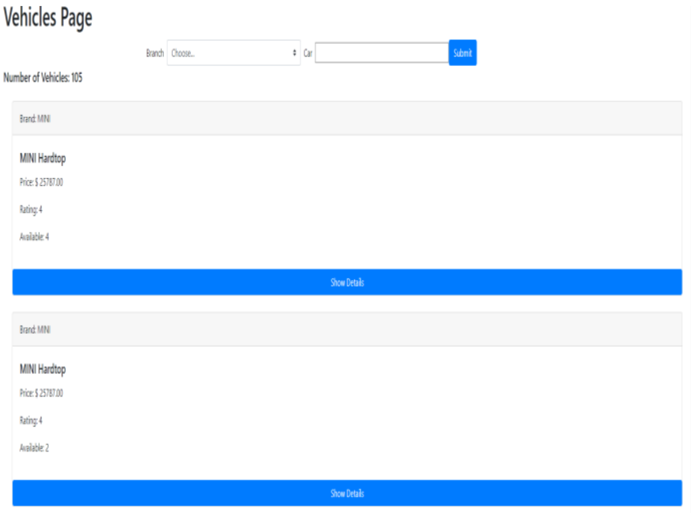
13. The customers table consists of data of name, email address, gender and mobile number of every customer who have purchased a vehicle from the dealer.

5. Application Description

The whole data is stored in a SQL server. And this data is being retrieved from the same SQL database using SQL queries. We have developed a web interface for employees and customers, where customers can go to the web page to buy a particular vehicle, book an appointment, request car services (like an Oil change, or car wash) or buy parts. On the other hand, employees can update the details of a Vehicle, Services, and Parts available in a particular branch.

I. User:

A user can first look at all the vehicles available, filter them by branch, or even search by a particular vehicle name.



Vehicles Page

Branch Choose... Car

Number of Vehicles: 105

Brand: MINI

MINI Hardtop

Price: $ 25787.00

Rating: 4

Available: 4

Brand: MINI

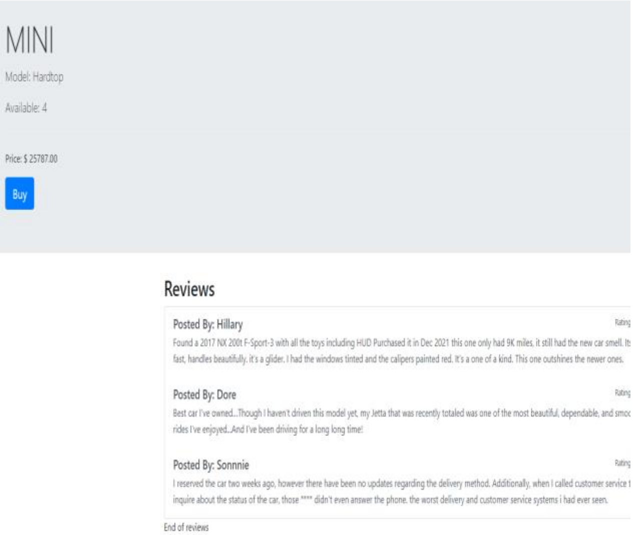
MINI Hardtop

Price: $ 25787.00

Rating: 4

Available: 2

If a user is interested in looking to purchase a vehicle, they can click on "Show Details" and get complete details about the availability, price, and reviews provided by other customers.



MINI

Model: Hardtop

Available: 4

Price: $ 25787.00

Reviews

Posted By: Hillary

Found a 2017 NX 200t F-Sport-3 with all the toys including HUD Purchased it in Dec 2021 this one only had 9K miles. it still had the new car smell. Its fast, handles beautifully. it's a glider. I had the windows tinted and the calipers painted red. It's a one of a kind. This one outshines the newer ones.

Posted By: Dore

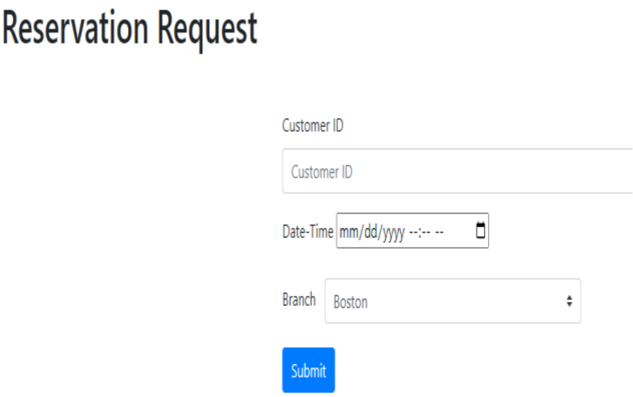
Best car I've owned...Though I haven't driven this model yet, my Jetta that was recently totaled was one of the most beautiful, dependable, and smoo rides I've enjoyed...And I've been driving for a long long time!

Posted By: Sonnnie

I reserved the car two weeks ago, however there have been no updates regarding the delivery method. Additionally, when I called customer service t inquire about the status of the car, those **** didn't even answer the phone. the worst delivery and customer service systems i had ever seen.

End of reviews

If the customer is ready to buy a vehicle, they can visit the store by requesting an appointment.



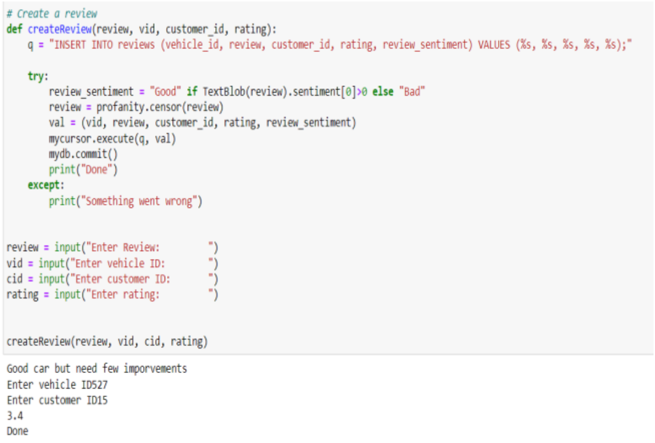
Reservation Request

Customer ID

Date-Time mm/dd/yyyy --:-- --

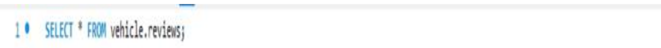
Branch Boston

A customer can even add a review for a particular vehicle. Before a review is added to our database, using **Textblob** we would estimate the **sentiment of the review** (i.e whether it is a good review or a bad review) and we also removed any inappropriate words in the reviews using Profanity.

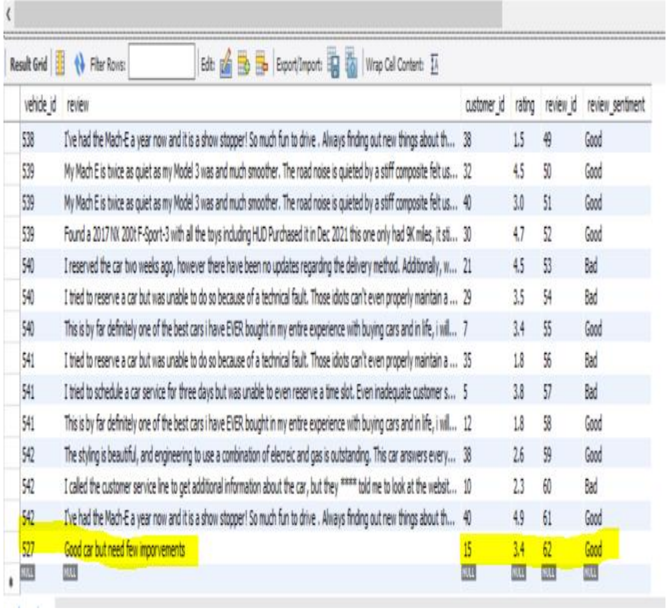


```
# Create a review
def createReview(review, vid, customer_id, rating):
    q = "INSERT INTO reviews (vehicle_id, review, customer_id, rating, review_sentiment) VALUES (%s, %s, %s, %s, %s);"

    try:
        review_sentiment = "Good" if TextBlob(review).sentiment[0]>0 else "Bad"
        review = profanity.censor(review)
        val = (vid, review, customer_id, rating, review_sentiment)
        mycursor.execute(q, val)
        mydb.commit()
        print("Done")
    except:
        print("Something went wrong")


review = input("Enter Review:        ")
vid = input("Enter vehicle ID:      ")
cid = input("Enter customer ID:     ")
rating = input("Enter rating:        ")


createReview(review, vid, cid, rating)
```

```
Good car but need few imporvements
Enter vehicle ID527
Enter customer ID15
3.4
Done
```



```
SELECT * FROM vehicle.reviews;
```

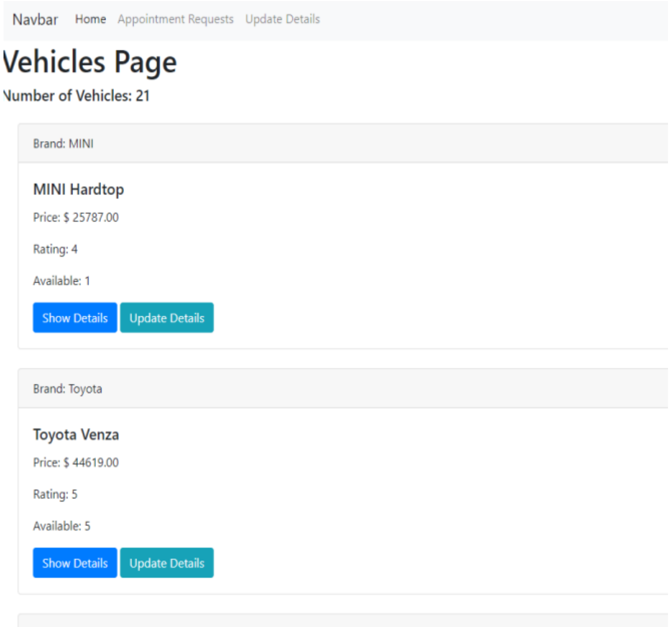


vehicle_id	review	customer_id	rating	review_id	review_sentiment
538	I've had the Mach-E a year now and it is a show stopper! So much fun to drive . Always finding out new things about th...	38	1.5	49	Good
539	My Mach E is twice as quiet as my Model 3 was and much smoother. The road noise is quieted by a stiff composite felt us...	32	4.5	50	Good
539	My Mach E is twice as quiet as my Model 3 was and much smoother. The road noise is quieted by a stiff composite felt us...	40	3.0	51	Good
539	Found a 2017 NX 200t F-Sport-3 with all the toys including HUD Purchased it in Dec 2021 this one only had 9K miles, it sti...	30	4.7	52	Good
540	I reserved the car two weeks ago, however there have been no updates regarding the delivery method. Additionally, w...	21	4.5	53	Bad
540	I tried to reserve a car but was unable to do so because of a technical fault. Those idiots can't even properly maintain a ...	29	3.5	54	Bad
540	This is by far definitely one of the best cars i have EVER bought in my entire experience with buying cars and in life, i wil...	7	3.4	55	Good
541	I tried to reserve a car but was unable to do so because of a technical fault. Those idiots can't even properly maintain a ...	35	1.8	56	Bad
541	I tried to schedule a car service for three days but was unable to even reserve a time slot. Even inadequate customer s...	5	3.8	57	Bad
541	This is by far definitely one of the best cars i have EVER bought in my entire experience with buying cars and in life, i wil...	12	1.8	58	Good
542	The styling is beautiful, and engineering to use a combination of elecreic and gas is outstanding. This car answers every...	38	2.6	59	Good
542	I called the customer service line to get additional information about the car, but they **** told me to look at the websit...	10	2.3	60	Bad
542	I've had the Mach-E a year now and it is a show stopper! So much fun to drive . Always finding out new things about th...	40	4.9	61	Good
527	Good car but need few imporvements	15	3.4	62	Good
NULL	NULL	NULL	NULL	NULL	NULL

II. Employee:

Every branch will have many employees. Each employee has a designation such as a manager, or Worker. As the manager has higher authority, there will be few tasks that can only be done by the manager. For instance, a manager can update the number of vehicles in their branch, whereas a worker cannot. A manager can assign a service to an employee of that branch.

After logging in, the employee can see details for their branch only.



Vehicles Page

Number of Vehicles: 21

Brand: MINI

MINI Hardtop

Price: $ 25787.00

Rating: 4

Available: 1

Brand: Toyota

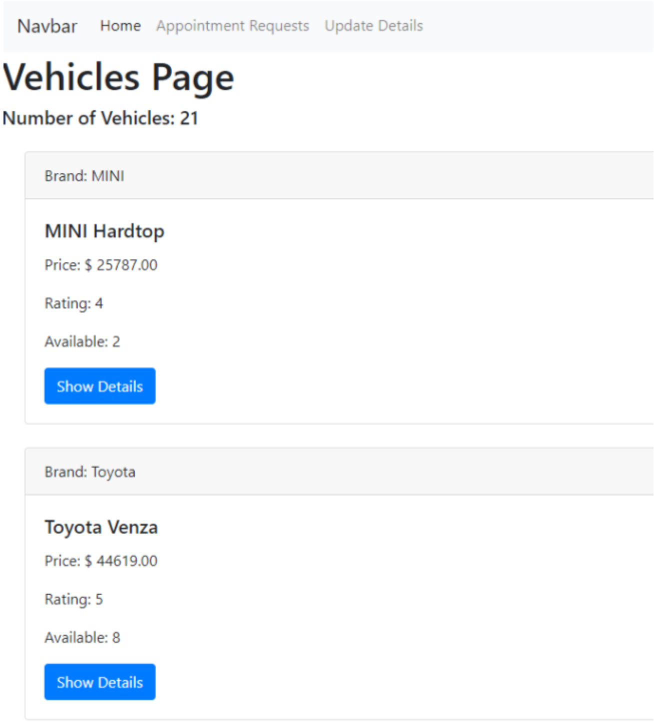
Toyota Venza

Price: $ 44619.00

Rating: 5

Available: 5

Landing Page of Manager



Vehicles Page

Number of Vehicles: 21

Brand: MINI

MINI Hardtop

Price: $ 25787.00

Rating: 4

Available: 2

Brand: Toyota

Toyota Venza

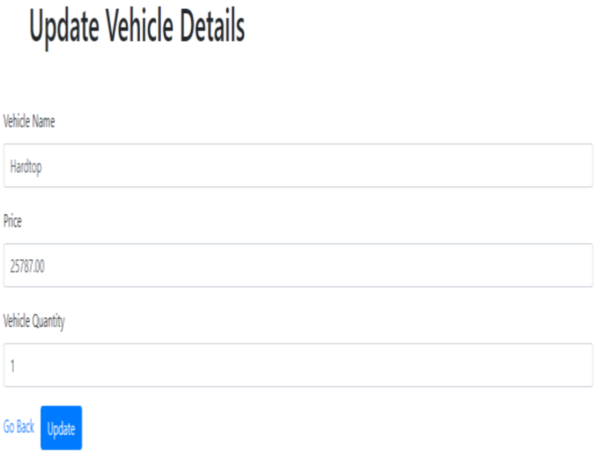
Price: $ 44619.00

Rating: 5

Available: 8

Landing Page of workers

Only the manager can update details about a vehicle, like a name, price, and several available units.



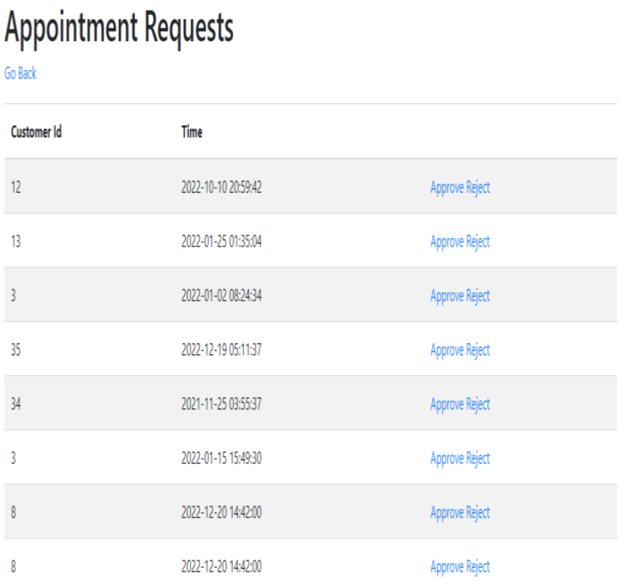
Update Vehicle Details

Vehicle Name: Hardtop

Price: 25787.00

Vehicle Quantity: 1

An employee whether the manager or the worker can approve or reject an appointment based on the requirements.

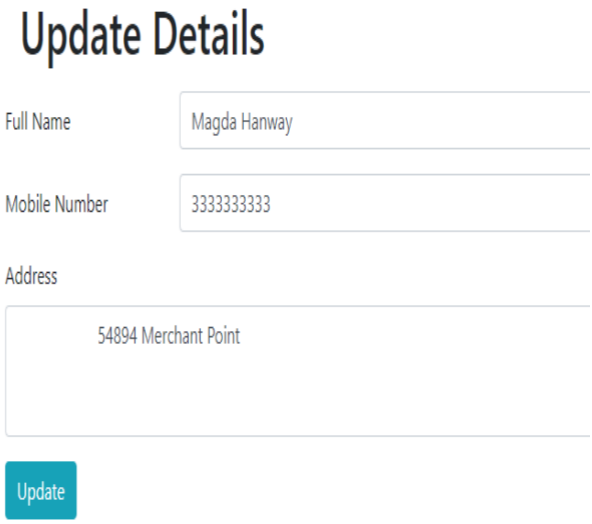


Appointment Requests

Go Back

Customer Id	Time	
12	2022-10-10 20:59:42	Approve Reject
13	2022-01-25 01:35:04	Approve Reject
3	2022-01-02 08:24:34	Approve Reject
35	2022-12-19 05:11:37	Approve Reject
34	2021-11-25 03:55:37	Approve Reject
3	2022-01-15 15:49:30	Approve Reject
8	2022-12-20 14:42:00	Approve Reject
8	2022-12-20 14:42:00	Approve Reject

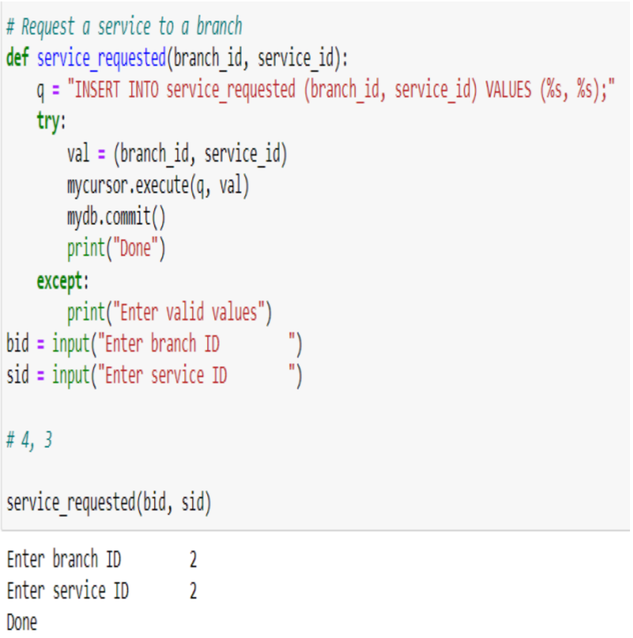
Any employee can update their details, like their name, address, and phone number.



Update Details

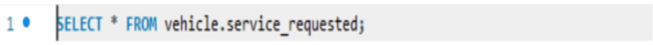
Full Name	Magda Hanway
Mobile Number	3333333333
Address	54894 Merchant Point

Anyone can request a service in a branch, they need not be the customers.

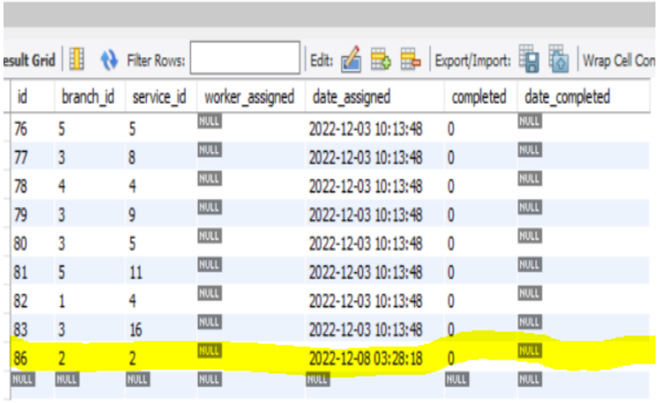


```
# Request a service to a branch
def service_requested(branch_id, service_id):
    q = "INSERT INTO service_requested (branch_id, service_id) VALUES (%s, %s);"
    try:
        val = (branch_id, service_id)
        mycursor.execute(q, val)
        mydb.commit()
        print("Done")
    except:
        print("Enter valid values")
bid = input("Enter branch ID       ")
sid = input("Enter service ID      ")

# 4, 3

service_requested(bid, sid)
```

```
Enter branch ID        2
Enter service ID       2
Done
```

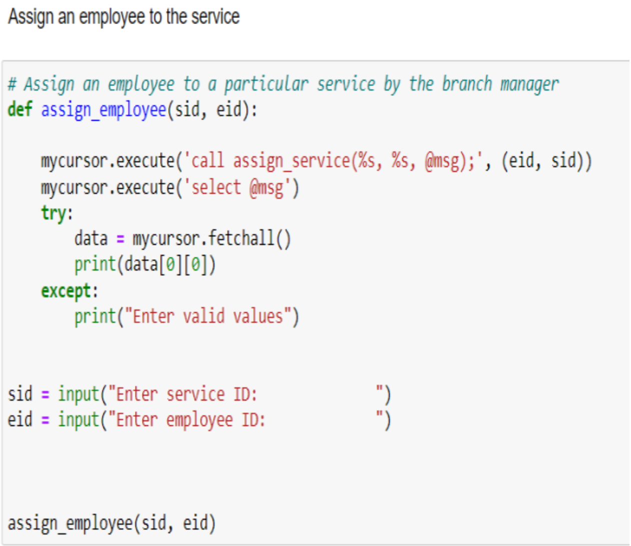


```
SELECT * FROM vehicle.service_requested;
```



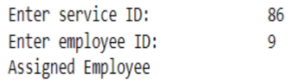
id	branch_id	service_id	worker_assigned	date_assigned	completed	date_completed
76	5	5	NULL	2022-12-03 10:13:48	0	NULL
77	3	8	NULL	2022-12-03 10:13:48	0	NULL
78	4	4	NULL	2022-12-03 10:13:48	0	NULL
79	3	9	NULL	2022-12-03 10:13:48	0	NULL
80	3	5	NULL	2022-12-03 10:13:48	0	NULL
81	5	11	NULL	2022-12-03 10:13:48	0	NULL
82	1	4	NULL	2022-12-03 10:13:48	0	NULL
83	3	16	NULL	2022-12-03 10:13:48	0	NULL
86	2	2	NULL	2022-12-08 03:28:18	0	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Now only the manager of that branch can assign one of their employees to work on this service.

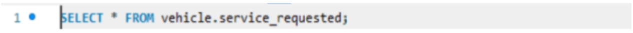


Assign an employee to the service

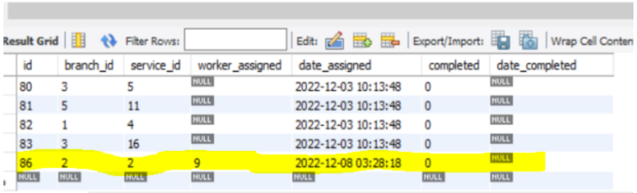
```
# Assign an employee to a particular service by the branch manager
def assign_employee(sid, eid):

    mycursor.execute('call assign_service(%s, %s, @msg);', (eid, sid))
    mycursor.execute('select @msg')
    try:
        data = mycursor.fetchall()
        print(data[0][0])
    except:
        print("Enter valid values")


sid = input("Enter service ID:              ")
eid = input("Enter employee ID:             ")


assign_employee(sid, eid)
```



```
Enter service ID:              86
Enter employee ID:             9
Assigned Employee
```



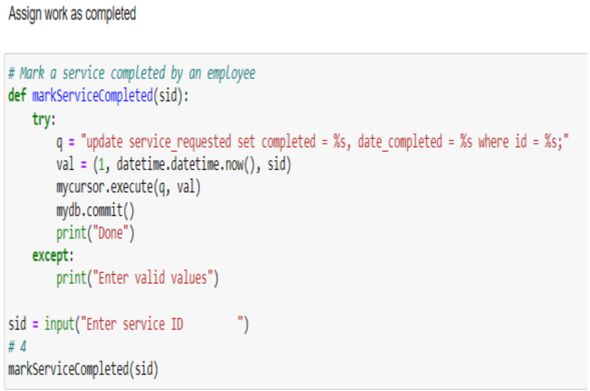
```
SELECT * FROM vehicle.service_requested;
```



Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Conten

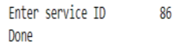
id	branch_id	service_id	worker_assigned	date_assigned	completed	date_completed
80	3	5	NULL	2022-12-03 10:13:48	0	NULL
81	5	11	NULL	2022-12-03 10:13:48	0	NULL
82	1	4	NULL	2022-12-03 10:13:48	0	NULL
83	3	16	NULL	2022-12-03 10:13:48	0	NULL
86	2	2	9	2022-12-08 03:28:18	0	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL

And once the service is completed the worker assigned to that service can mark it completed.

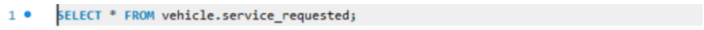


Assign work as completed

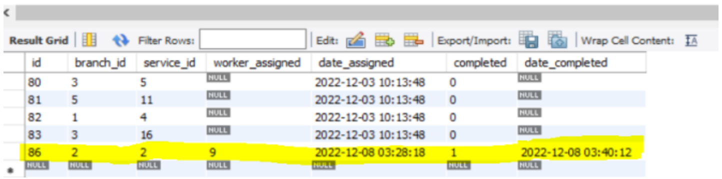
```
# Mark a service completed by an employee
def markServiceCompleted(sid):
    try:
        q = "update service_requested set completed = %s, date_completed = %s where id = %s;"
        val = (1, datetime.datetime.now(), sid)
        mycursor.execute(q, val)
        mydb.commit()
        print("Done")
    except:
        print("Enter valid values")

sid = input("Enter service ID        ")
# 4
markServiceCompleted(sid)
```



```
Enter service ID        86
Done
```



```
SELECT * FROM vehicle.service_requested;
```



Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:

id	branch_id	service_id	worker_assigned	date_assigned	completed	date_completed
80	3	5	NULL	2022-12-03 10:13:48	0	NULL
81	5	11	NULL	2022-12-03 10:13:48	0	NULL
82	1	4	NULL	2022-12-03 10:13:48	0	NULL
83	3	16	NULL	2022-12-03 10:13:48	0	NULL
86	2	2	9	2022-12-08 03:28:18	1	2022-12-08 03:40:12
NULL	NULL	NULL	NULL	NULL	NULL	NULL

III. Triggers:

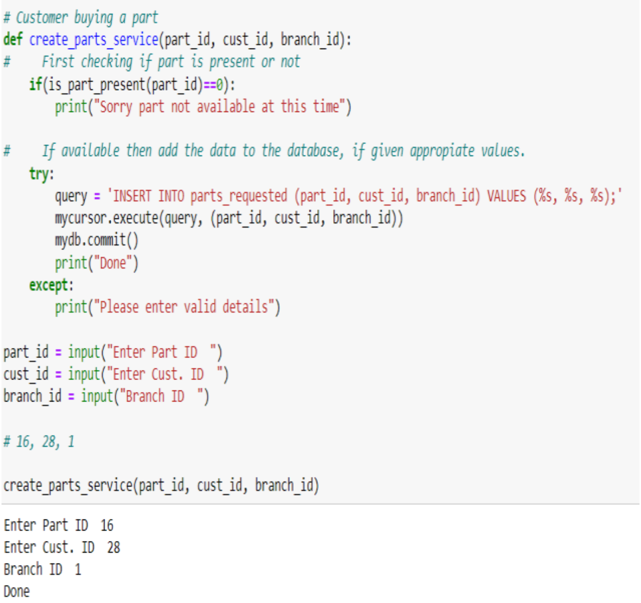
1. Customers can purchase different vehicles available. Whenever a particular vehicle is bought by the customer the vehicle count in that branch gets decreased.

2. Customers can buy different parts available. Whenever a particular part is bought by the customer the part_id, and customer_id gets stored in the parts_requested table and the

number of the parts present in the inventory gets reduced accordingly.

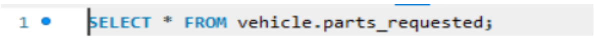
Output:

Customer with ID 28, purchased a part with ID 16.

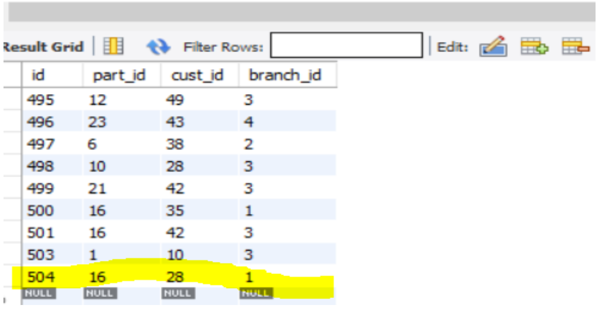


```
# Customer buying a part
def create_parts_service(part_id, cust_id, branch_id):
#     First checking if part is present or not
    if(is_part_present(part_id)==0):
        print("Sorry part not available at this time")

#     If available then add the data to the database, if given appropiate values.
    try:
        query = 'INSERT INTO parts_requested (part_id, cust_id, branch_id) VALUES (%s, %s, %s);'
        mycursor.execute(query, (part_id, cust_id, branch_id))
        mydb.commit()
        print("Done")
    except:
        print("Please enter valid details")

part_id = input("Enter Part ID  ")
cust_id = input("Enter Cust. ID  ")
branch_id = input("Branch ID  ")

# 16, 28, 1

create_parts_service(part_id, cust_id, branch_id)
```

```
Enter Part ID  16
Enter Cust. ID  28
Branch ID  1
Done
```

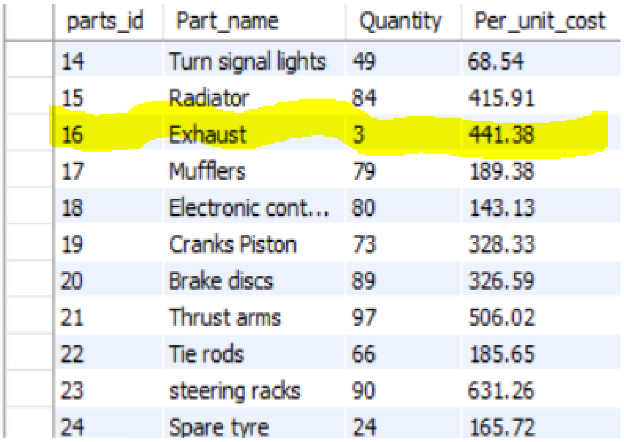


```
SELECT * FROM vehicle.parts_requested;
```



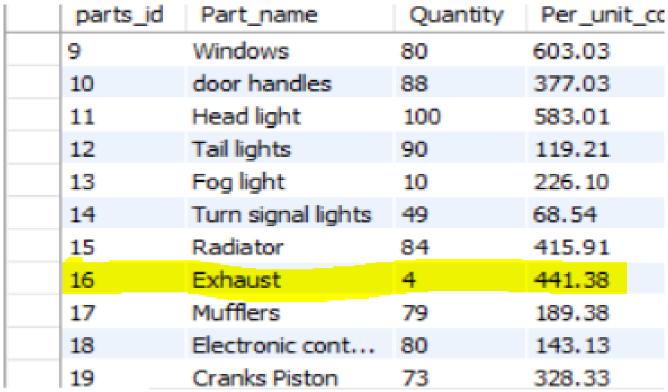
id	part_id	cust_id	branch_id
495	12	49	3
496	23	43	4
497	6	38	2
498	10	28	3
499	21	42	3
500	16	35	1
501	16	42	3
503	1	10	3
504	16	28	1
NULL	NULL	NULL	NULL

Quantity of part_id = 16, After purchase.



parts_id	Part_name	Quantity	Per_unit_cost
14	Turn signal lights	49	68.54
15	Radiator	84	415.91
16	Exhaust	3	441.38
17	Mufflers	79	189.38
18	Electronic cont...	80	143.13
19	Cranks Piston	73	328.33
20	Brake discs	89	326.59
21	Thrust arms	97	506.02
22	Tie rods	66	185.65
23	steering racks	90	631.26
24	Spare tyre	24	165.72

Quantity of part_id = 16, Before purchase.



parts_id	Part_name	Quantity	Per_unit_cc
9	Windows	80	603.03
10	door handles	88	377.03
11	Head light	100	583.01
12	Tail lights	90	119.21
13	Fog light	10	226.10
14	Turn signal lights	49	68.54
15	Radiator	84	415.91
16	Exhaust	4	441.38
17	Mufflers	79	189.38
18	Electronic cont...	80	143.13
19	Cranks Piston	73	328.33

IV. Stored Procedures:

In total, we have implemented 5 stored procedures.

1. Given the employee id, it returns all the details of that employee's performance. (As the name, branch, salary, Services done, revenue generated, and profit generated).

2. A stored procedure to assign a service by the manager to an employee working in that branch.

3. A stored procedure to update employee salary by only the manager, used multiple functions inside this stored procedure to update the salary of the employee.

4. A stored procedure to update vehicle count in a particular branch.

5. A stored procedure to create a reservation.

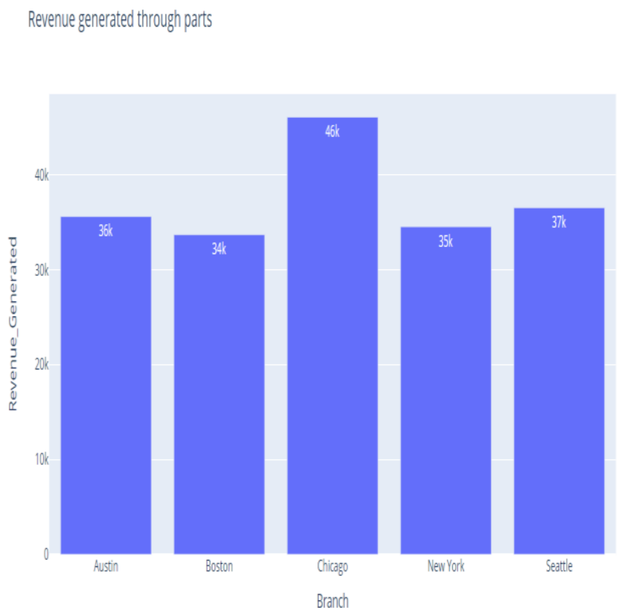
V. Views:

Created a view with all the sales information of all vehicles, parts, and services across all the branches.

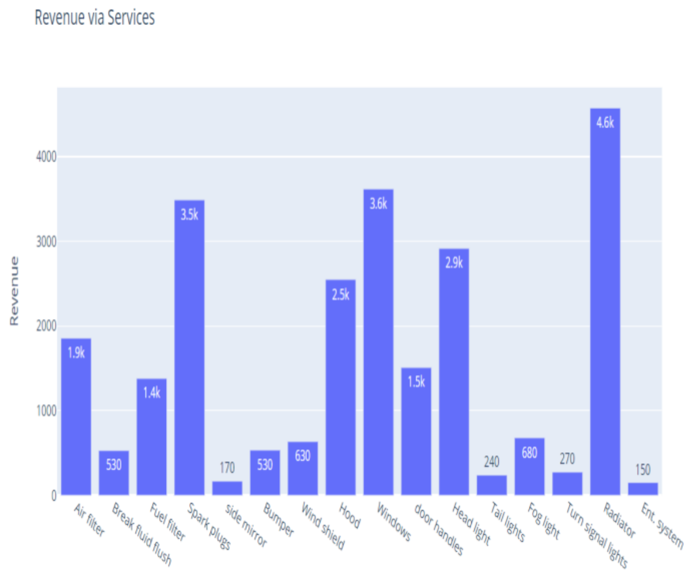
Using this view created multiple plots to visualize the revenue generated through Vehicles, Parts, and Services. And from this, we have even interpreted the sales made by each employee.

VI. Visualizations:

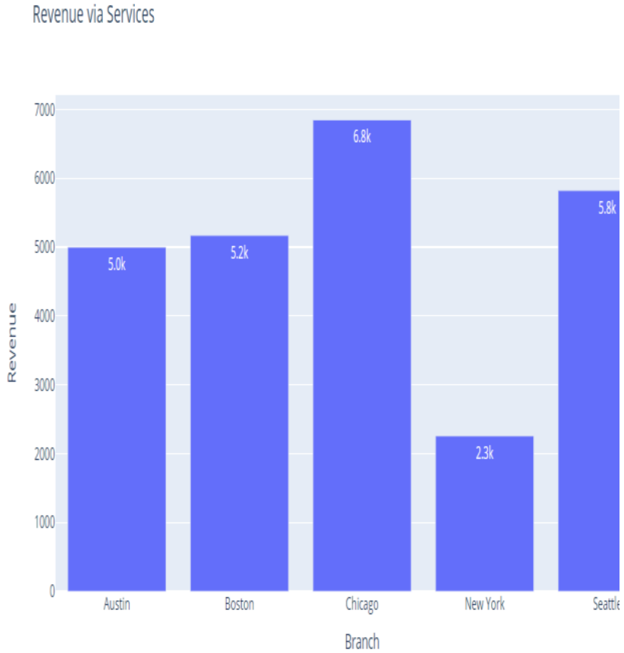
Each Branch Revenue is Generated through Parts.



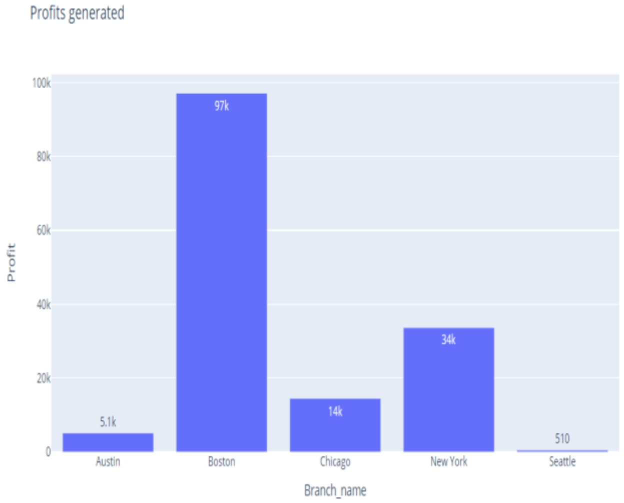
Revenue Generated through Services.



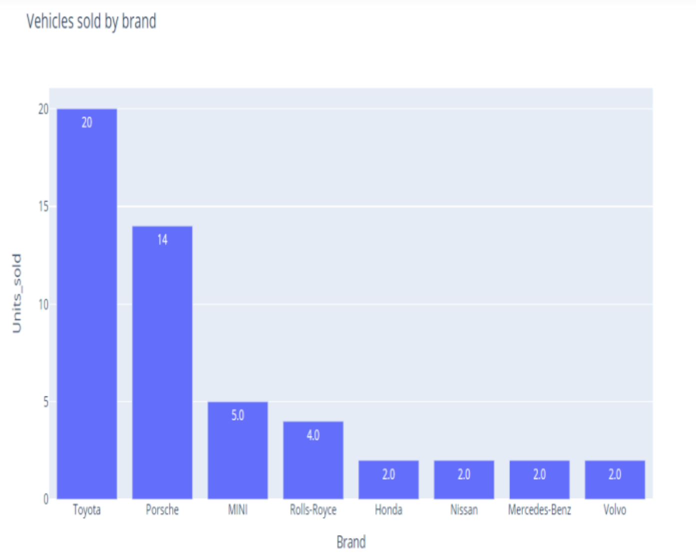
Each Branch Revenue through Services.



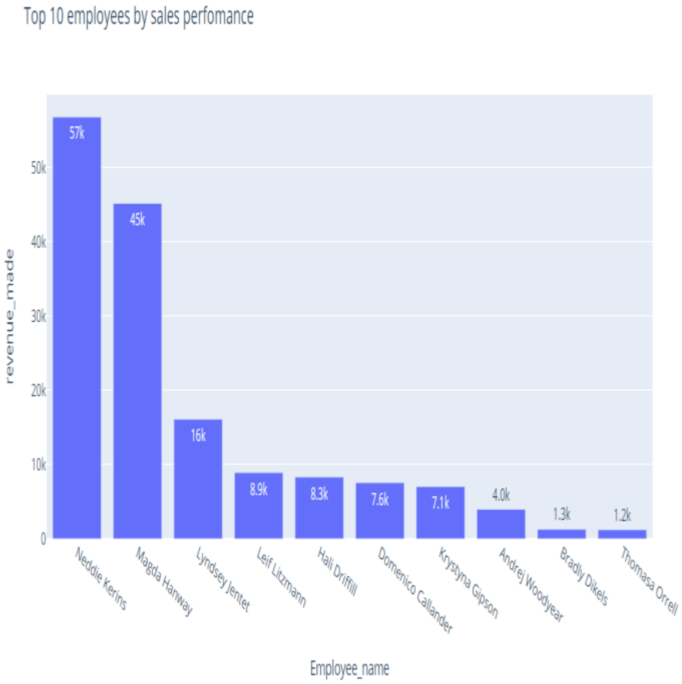
Each Branch Profits are Generated through Selling Vehicles.



No. of Brands sold across all the branches



Top 10 Employees by Sales Performance.



6. Conclusion:

We have developed this project vehicle management system to digitalize the entire records to access it better, which improves the business efficacy alongside commencing with the clients in a better way. This has effectively included accessing data and manipulating it in the comfort of managing it more efficiently.

This also provides a hassle-free graphical interface which shows to be one step ahead when compared to the currently existing systems. This system will substantially overcome the hindrance in communication with each other and in addition updating and adding data in a system will also become uncomplicated.

This system also has enough scope for refinement in the future. Instead of the manager assigning work to workers manually, we can look into the worker timetable and look for workers with free time and assign the work.

We learned things on how to connect tables, views, stored procedures, and functions together. We also learned how to connect a web application with a database. If we had more time, we would add more features and develop a chatbot and better user experience for our application. We recommend that future DS5110 students to include more use cases and develop an end to end application.