

Automobile Backup Products Booking System

Milestone: Application

Group 3

Aarthi Thinakaran

Venkata Krishnan

857-390-4725

857-210-1207

thinakaran.a@northeastern.edu

ravichandran.ve@northeastern.edu

Percentage of Effort Contributed by Student1: _____ 50% _____

Percentage of Effort Contributed by Student2: _____ 50% _____

Signature of Student 1: _____ Aarthi Thinakaran _____

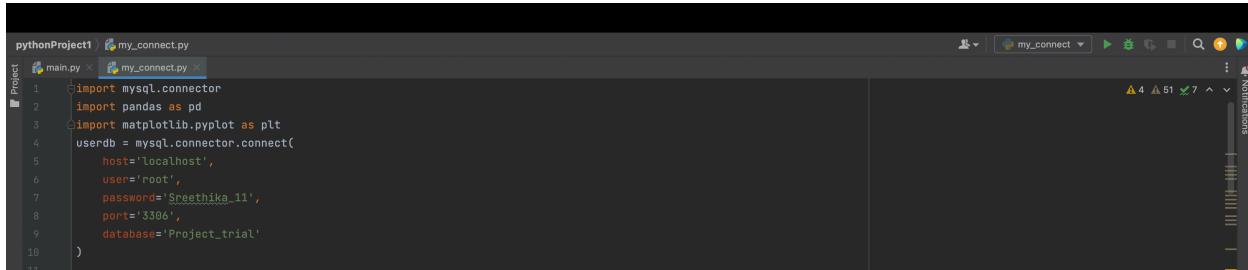
Signature of Student 2: _____ Venkata Krishnan _____

Submission Date: _____ 26 November 2022 _____

SUMMARY:

We have implemented the integration of MySQL and Python using the PyCharm IDE.

Below is the code we used for our integration purpose.



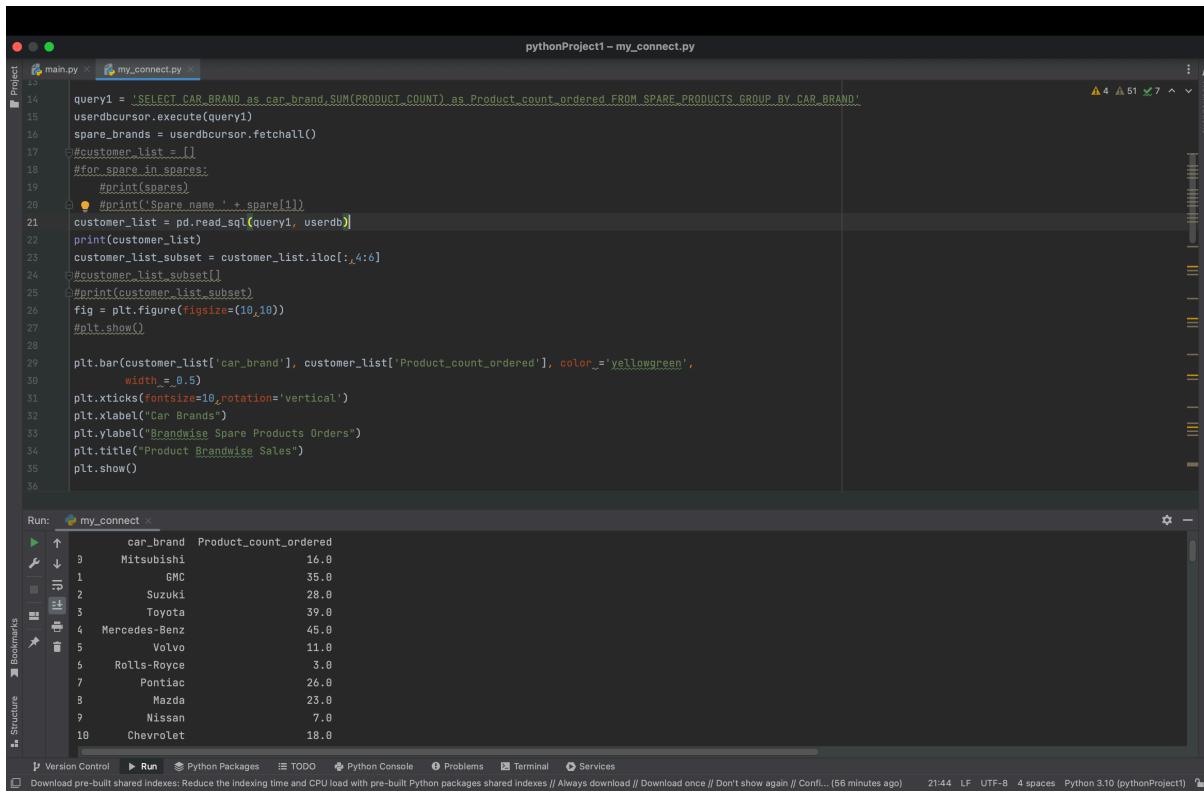
```
pythonProject1 my_connect.py
main.py my_connect.py
1 import mysql.connector
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 userdb = mysql.connector.connect(
5     host='localhost',
6     user='root',
7     password='Sreethika_11',
8     port='3306',
9     database='Project_trial'
10 )
```

We have created different plots of analysis using the data from our tables in the MySQL database.

PLOT 1:

We have showed an analysis on the number of automobile spare products ordered from each car brand.

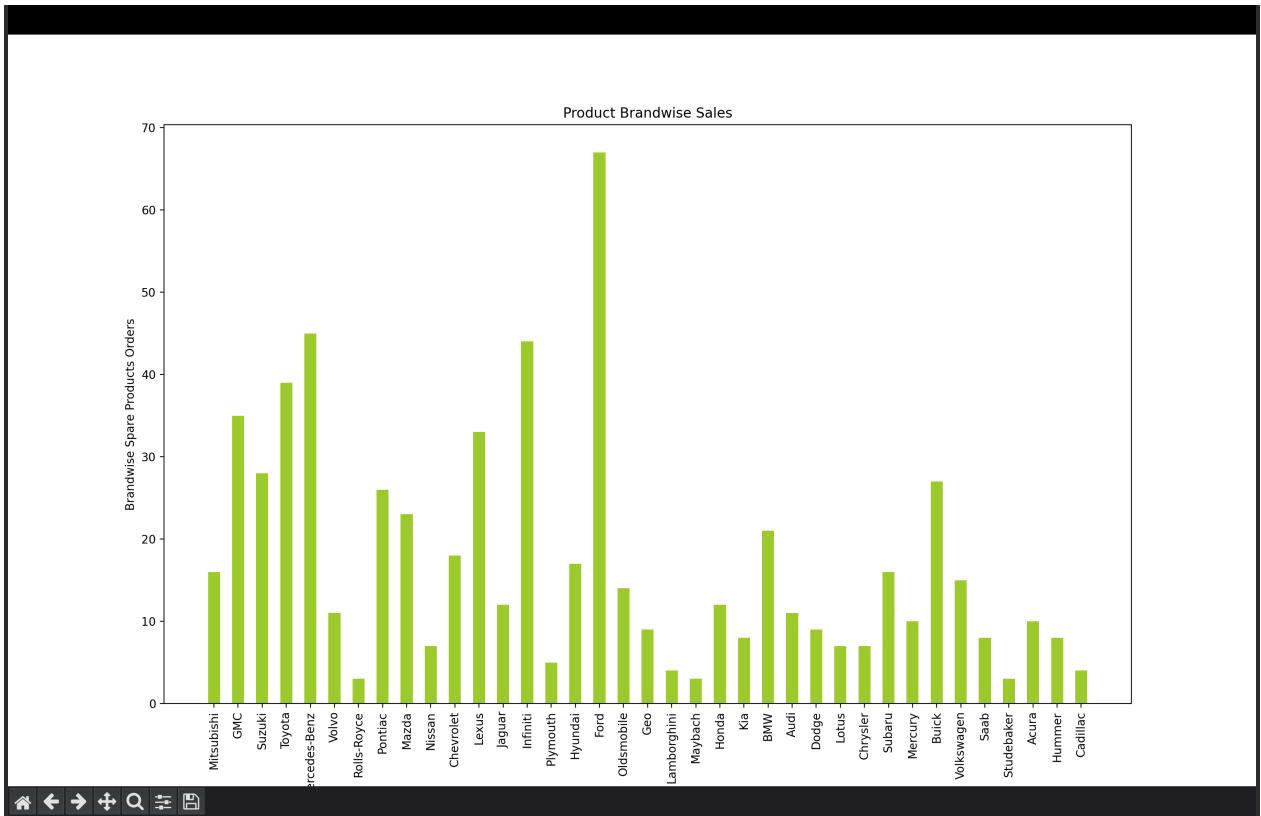
Below is the code we used for this analysis.



```
pythonProject1 - my_connect.py
main.py my_connect.py
14 query1 = 'SELECT CAR_BRAND as car_brand,SUM(PRODUCT_COUNT) as Product_count_ordered FROM SPARE_PRODUCTS GROUP BY CAR_BRAND'
15 userdbcursor.execute(query1)
16 spare_brands = userdbcursor.fetchall()
17 #customer_list = []
18 #for spare in spares:
19 #    #print(spares)
20 #    #print('Spare name ' + spare[1])
21 customer_list = pd.read_sql(query1, userdb)
22 print(customer_list)
23 customer_list_subset = customer_list.iloc[:,4:6]
24 #customer_list_subset[]
25 #print(customer_list_subset)
26 fig = plt.figure(figsize=(10,10))
27 #plt.show()
28
29 plt.bar(customer_list['car_brand'], customer_list['Product_count_ordered'], color='yellowgreen',
30         width=0.5)
31 plt.xticks(fontsize=10, rotation='vertical')
32 plt.xlabel("Car Brands")
33 plt.ylabel("Brandwise Spare Products Orders")
34 plt.title("Product Brandwise Sales")
35 plt.show()
36
```

Run: my_connect

car_brand	Product_count_ordered
Mitsubishi	16.0
GMC	55.0
Suzuki	28.0
Toyota	39.0
Mercedes-Benz	45.0
Volvo	11.0
Rolls-Royce	3.0
Pontiac	26.0
Mazda	23.0
Nissan	7.0
Chevrolet	18.0



PLOT 2:

We have showed an analysis on the amount of customer orders made based on the type of automobile spare product.

Below is the code we used for this analysis.

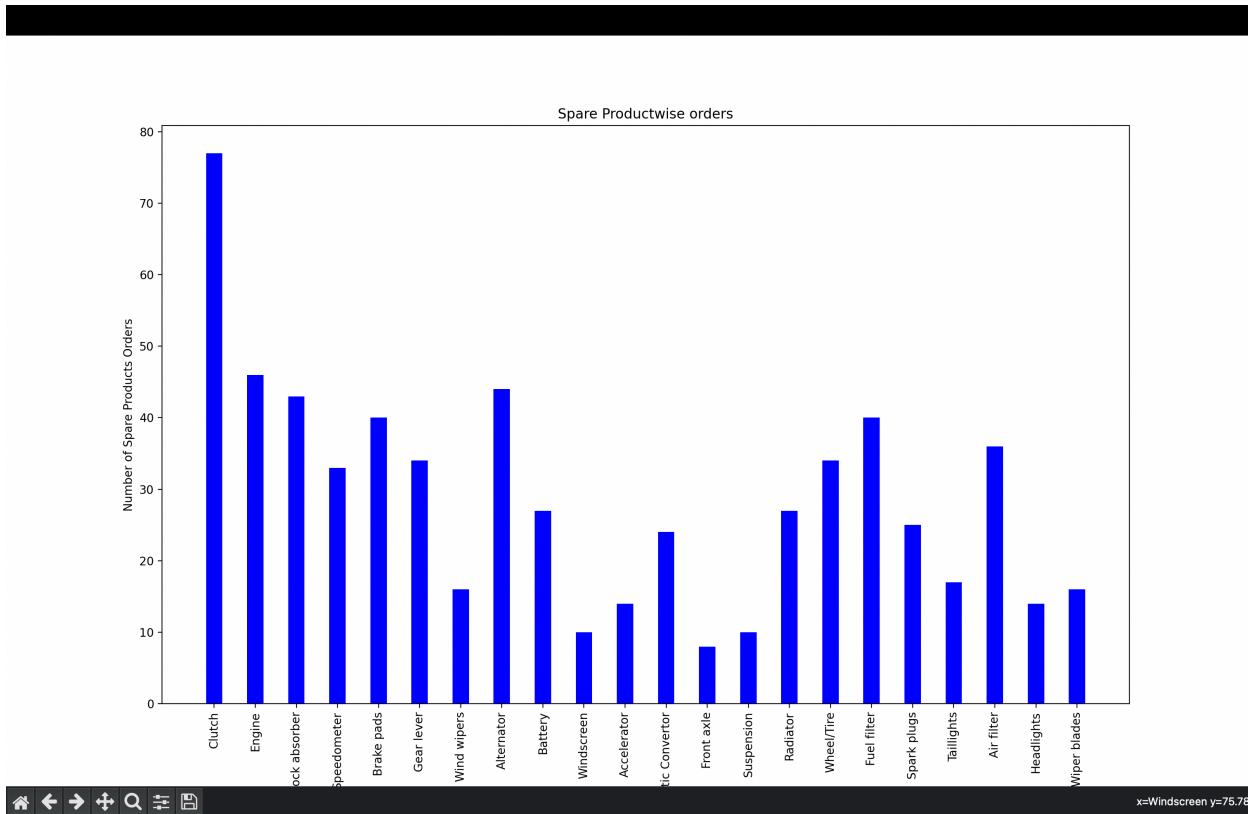
```

pythonProject1 - my_connect.py
main.py x my_connect.py
Project: pythonProject1 - my_connect.py
27     #plt.show()
28
29     plt.bar(customer_list['car_brand'], customer_list['Product_count_ordered'], color='yellowgreen',
30             width=.5)
31     plt.xticks(fontsize=10, rotation='vertical')
32     plt.xlabel('Car Brands')
33     plt.ylabel('Brandwise Spare Products Orders')
34     plt.title("Product Brandwise Sales")
35     plt.show()
36
37     query2 = "SELECT SPARE_NAME as Spare_Product_Name, SUM(PRODUCT_COUNT) as Product_count_ordered FROM SPARE_PRODUCTS GROUP BY SPARE_NAME"
38     userdbcursor.execute(query2)
39     spares = userdbcursor.fetchall()
40     spare_products = pd.read_sql(query2, userdb)
41     print(spare_products)
42     fig = plt.figure(figsize=(10,10))
43     plt.bar(spare_products['Spare_Product_Name'],spare_products['Product_count_ordered'],color='blue',width=0.4)
44     plt.xticks(fontsize=10, rotation='vertical')
45     plt.xlabel("Spare Products")
46     plt.ylabel("Number of Spare Products Orders")
47     plt.title("Spare Productwise orders")
48     plt.show()
49
Run: my_connect <
      Spare_Product_Name  Product_count_ordered
0           Clutch            77.0
1          Engine            46.0
2   Shock absorber            43.0
3    Speedometer            33.0
4    Brake pads            40.0
5    Gear lever            34.0
6    Wind wipers            16.0
7    Alternator            44.0
8      Battery            27.0
9   Windscreen            18.0
10   Accelerator            14.0

```

Version Control Run Python Packages TODO Python Console Problems Terminal Services

Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packages shared indexes // Always download // Download once // Don't show again // Configuration (58 minutes ago) 45:29 LF UTF-8 4 spaces Python 3.10 (pythonProject1)



PLOT 3:

We have plotted an analysis of the monthly sales made with respect to the customer orders.

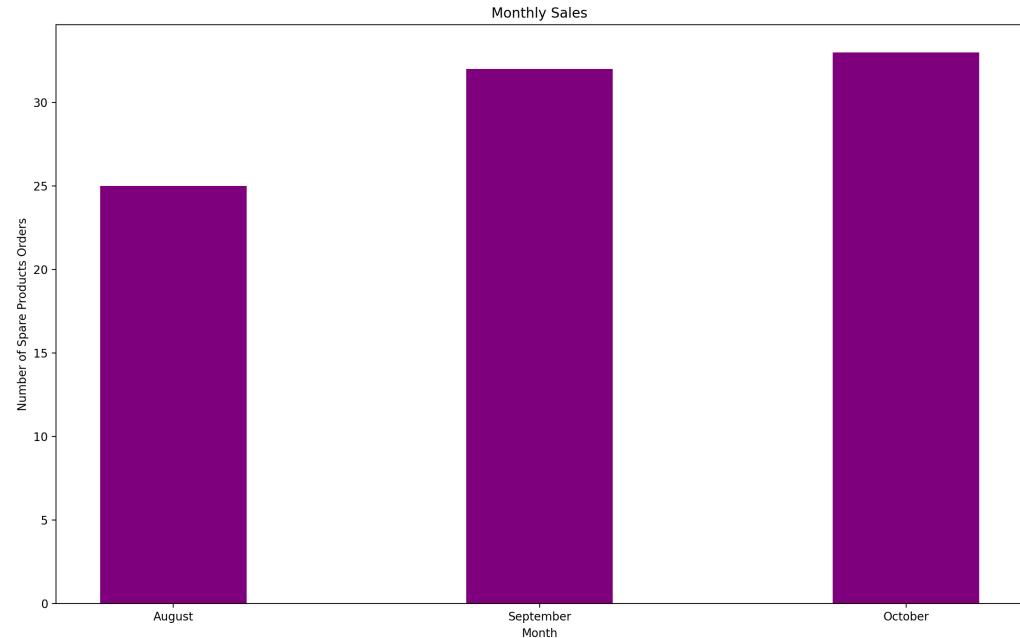
Below is the code we used for this analysis.

```
pythonProject1 - my_connect.py
main.py x my_connect.py ...
40:     spare_products = pd.read_sql(query2, userdb)
41:     print(share_products)
42:     fig = plt.figure(figsize=(10,10))
43:     plt.bar(share_products['Spare_Product_Name'],share_products['Product_count_ordered'],color='blue',width=0.4)
44:     plt.xticks(fontsize=10,rotation='vertical')
45:     plt.xlabel('Spare Products')
46:     plt.ylabel("Number of Spare Products Orders")
47:     plt.title("Spare Productwise orders")
48:     plt.show()
49:
50: query3 = 'SELECT MONTH(order_date) as month,COUNT(order_id) as count FROM CUSTOMER_BOOKINGS GROUP BY MONTH(order_date),YEAR(order_date)'
51: userdbcursor.execute(query3)
52: no_of_orders = userdbcursor.fetchall()
53: monthly_sales = pd.read_sql(query3, userdb)
54: print(monthly_sales)
55: fig = plt.figure(figsize=(10,10))
56: months = ['August','September','October']
57: plt.bar(months,monthly_sales['count'],color='purple',width=0.4)
58: plt.xticks(fontsize=10)
59: plt.xlabel("Month")
60: plt.ylabel("Number of Spare Products Orders")
61: plt.title("Monthly Sales")
62: plt.show()

Run: my_connect x
  17 Air filter          36.0
  20 Headlights          14.0
  21 Wiper blades         16.0
  month count
  3    8    25
  1    9    32
  2   10    33
  /Users/HQ/PycharmProjects/pythonProject1/my_connect.py:54: UserWarning: pandas only supports SQLAlchemy connectable (engine/connection) or database string URI or sqlite3 DBAPI
  monthly_sales = pd.read_sql(query3, userdb)
  months
  /Users/HQ/PycharmProjects/pythonProject1/my_connect.py:70: UserWarning: pandas only supports SQLAlchemy connectable (engine/connection) or database string URI or sqlite3 DBAPI
  august_sales = pd.read_sql(query4, userdb)
  Date count
  3  2022-08-01      1
  /Users/HQ/PycharmProjects/pythonProject1/my_connect.py:70: UserWarning: pandas only supports SQLAlchemy connectable (engine/connection) or database string URI or sqlite3 DBAPI
  august_sales = pd.read_sql(query4, userdb)

Project: pythonProject1 - my_connect.py
  main.py x my_connect.py ...
  Run: my_connect x
  17 Air filter          36.0
  20 Headlights          14.0
  21 Wiper blades         16.0
  month count
  3    8    25
  1    9    32
  2   10    33
  /Users/HQ/PycharmProjects/pythonProject1/my_connect.py:54: UserWarning: pandas only supports SQLAlchemy connectable (engine/connection) or database string URI or sqlite3 DBAPI
  monthly_sales = pd.read_sql(query3, userdb)
  months
  /Users/HQ/PycharmProjects/pythonProject1/my_connect.py:70: UserWarning: pandas only supports SQLAlchemy connectable (engine/connection) or database string URI or sqlite3 DBAPI
  august_sales = pd.read_sql(query4, userdb)
  Date count
  3  2022-08-01      1
  /Users/HQ/PycharmProjects/pythonProject1/my_connect.py:70: UserWarning: pandas only supports SQLAlchemy connectable (engine/connection) or database string URI or sqlite3 DBAPI
  august_sales = pd.read_sql(query4, userdb)

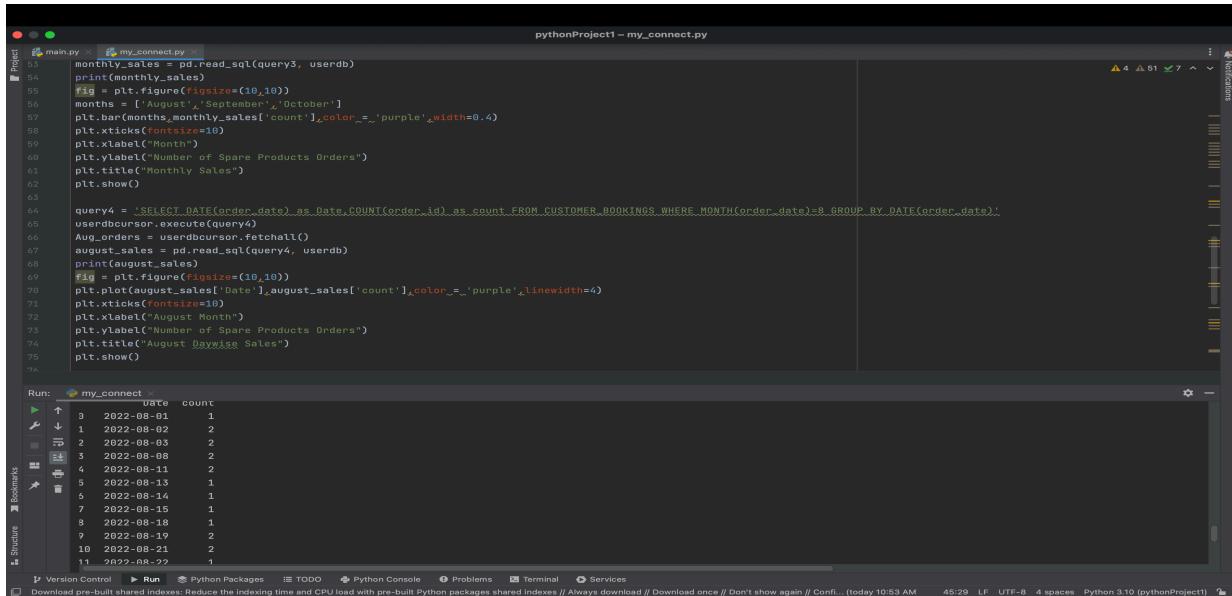
Version Control  Run  Python Packages  TODO  Python Console  Problems  Terminal  Services
Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packages shared indexes // Always download // Download once // Don't show again // Configure... (1 hour ago)  45:29  LF  UTF-8  4 spaces  Python 3.10 (pythonProject1)  99%
```



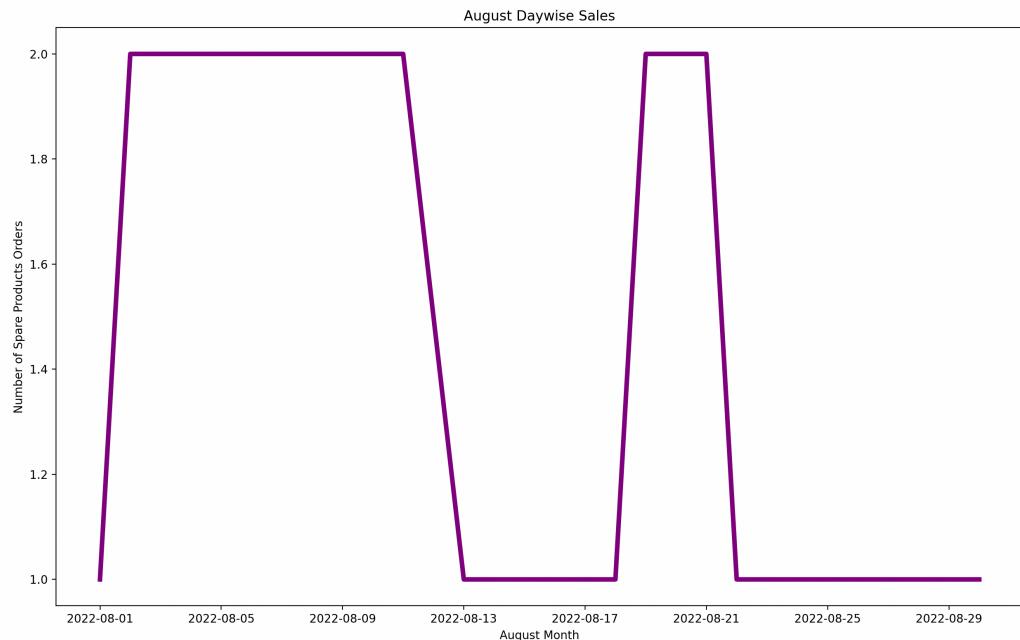
PLOT 4:

We have plotted an analysis on the daily sales for the month of ‘August’.

Below is the code we used for this analysis.



```
pythonProject1 - my_connect.py
53 monthly_sales = pd.read_sql(query3, userdb)
54 print(monthly_sales)
55 fig, ax = plt.subplots(figsize=(10,10))
56 months = ['August','September','October']
57 plt.bar(months,monthly_sales['count'],color='purple',width=0.4)
58 plt.xticks(fontsize=10)
59 plt.xlabel("Month")
60 plt.ylabel("Number of Spare Products Orders")
61 plt.title("Monthly Sales")
62 plt.show()
63
64 query4 = "SELECT DATE(order_date).as.Date,COUNT(order_id).as.count FROM CUSTOMER_BOOKINGS WHERE MONTH(order_date)=8 GROUP BY DATE(order_date)"
65 userdbcursor.execute(query4)
66 Aug_orders = userdbcursor.fetchall()
67 august_sales = pd.read_sql(query4, userdb)
68 print(august_sales)
69 fig = plt.figure(figsize=(10,10))
70 plt.plot(august_sales['Date'],august_sales['count'],color='purple',linewidth=4)
71 plt.xticks(fontsize=10)
72 plt.xlabel("August Month")
73 plt.ylabel("Number of Spare Products Orders")
74 plt.title("August Daywise Sales")
75 plt.show()
76
```



PLOT 5:

We have made an analysis on number of deliveries made by each Delivery Service Provider in the database.

Below is the code we used for this analysis.

```
pythonProject1 | my_connect.py
Project: my_connect.py
File: my_connect.py
47:         august_sales = pd.read_sql(query4, userdb)
48:         print(august_sales)
49:         fig = plt.figure(figsize=(10,10))
50:         plt.plot(august_sales['Date'],august_sales['count'],color='purple',linewidth=4)
51:         plt.xticks(fontsize=10)
52:         plt.xlabel("August Month")
53:         plt.ylabel("Number of Spare Products Orders")
54:         plt.title("August Daywise Sales")
55:         plt.show()
56:
57:
58: query5 = "SELECT DELIVERY_PROVIDER_NAME AS Delivery_Service_Provider,DELIVERIES_COUNT HANDLED AS Deliveries_Handled FROM DELIVERY_SERVICE_PROVIDERS"
59: userdbcursor.execute(query5)
60: count_of_deliveries = userdbcursor.fetchall()
61: delivery_count = pd.read_sql(query5, userdb)
62: delivery_count_subset = delivery_count.iloc[:20,:]
63: print(delivery_count_subset)
64: fig = plt.figure(figsize=(10,10))
65: delivery_count_subset['Delivery_Service_Provider'],delivery_count_subset['Deliveries_Handled'],color='blue',width=0.4)
66: plt.xticks(rotation=90,vertical)
67: plt.xlabel("Delivery Service Provider")
68: plt.ylabel("Number of Orders Delivered")
69: plt.title("Delivery Service Count")
70: plt.show()

Run: my_connect
/Users/mu/vcharmProjects/pythonProject1/my_connect.py:86: UserWarning: pandas only supports SQLAlchemy connectable (engine/connection) or database string URI or sqlite3 DBAPI
  delivery_count = pd.read_sql(query5, userdb)
Delivery_Service_Provider  Deliveries_Handled
0                  It            28
1      Y-Solwarm            30
2     Home Ing            33
3        Sonar             61
4      Ronstring            75
5      Ronstring            53
6        Yfind              46
7      Holdlamis            73
8       Tresom              32
```

