

Automobile Backup Products Booking System

Milestone: Application

Venkata Krishnan

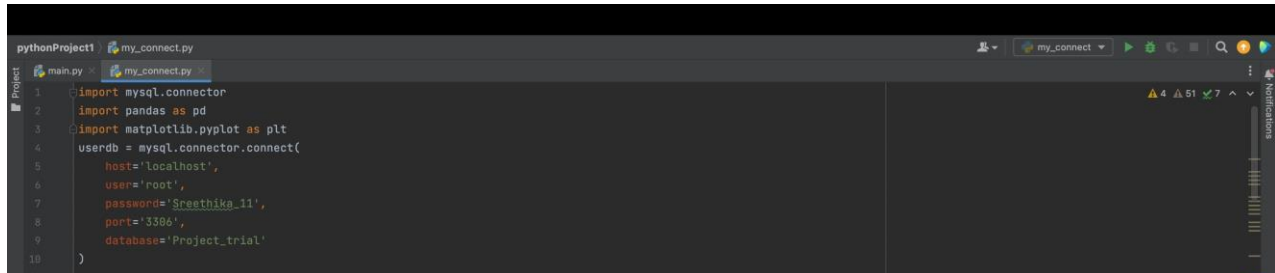
857-210-1207

ravichandran.ve@northeastern.edu

SUMMARY:

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

Below is the code we used for our integration purpose.



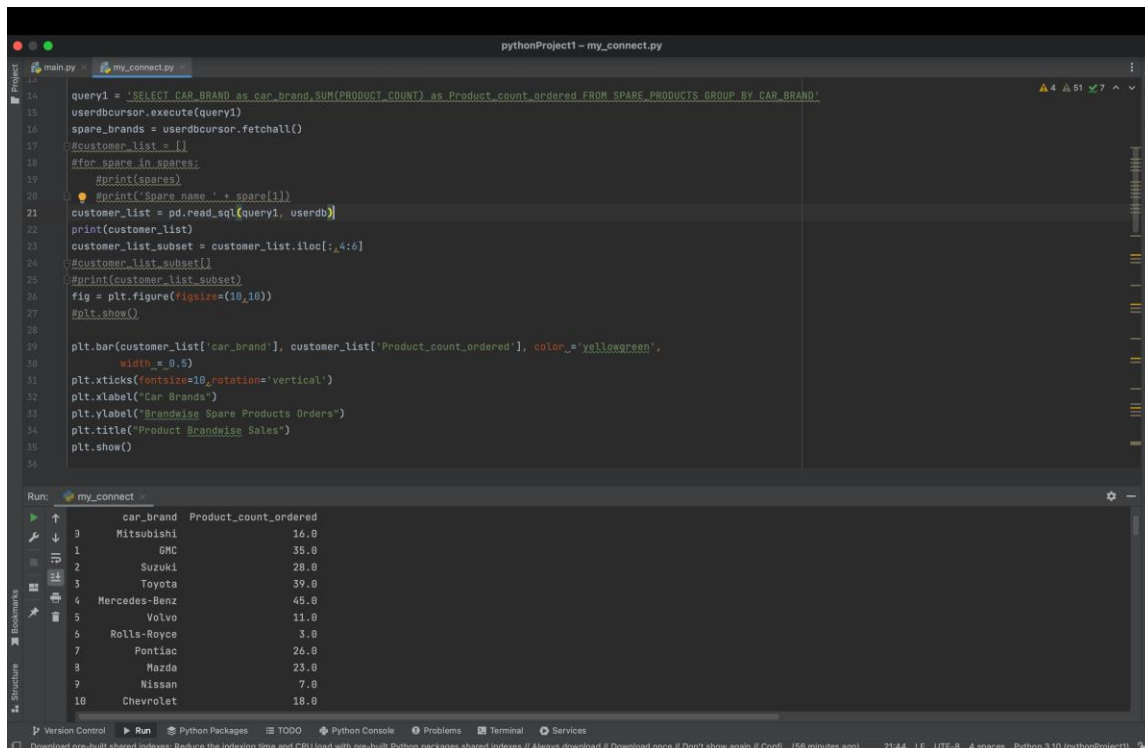
```
pythonProject1 my_connect.py
1 import mysql.connector
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 userdb = mysql.connector.connect(
5     host='localhost',
6     user='root',
7     password='Sreethika_11',
8     port='3306',
9     database='Project_trial'
10 )
```

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.

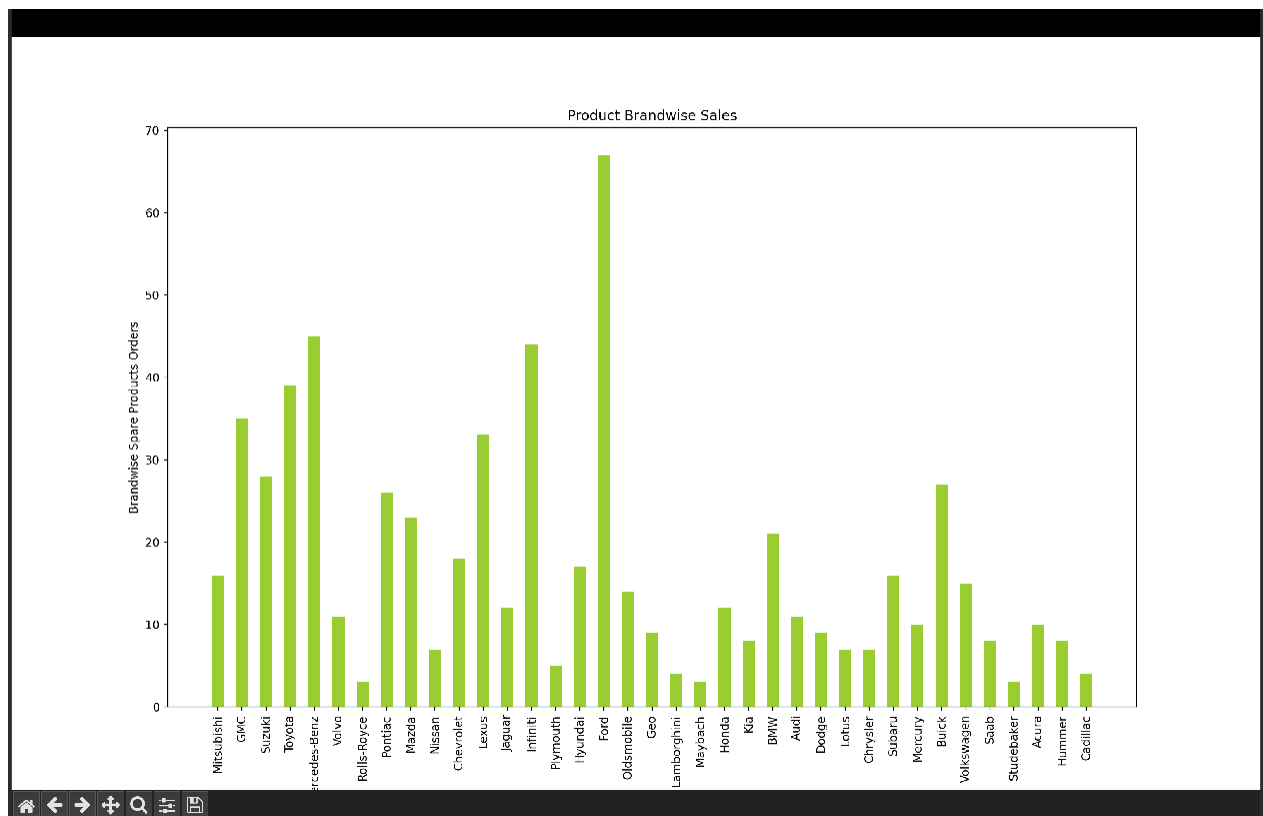
Below is the code we used for this analysis.



```
pythonProject1 - my_connect.py
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(spare)
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
9	Mitsubishi	16.0
1	GMC	35.0
2	Suzuki	28.0
3	Toyota	39.0
4	Mercedes-Benz	45.0
5	Volvo	11.0
6	Rolls-Royce	3.0
7	Pontiac	26.0
8	Mazda	23.0
9	Nissan	7.0
10	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.

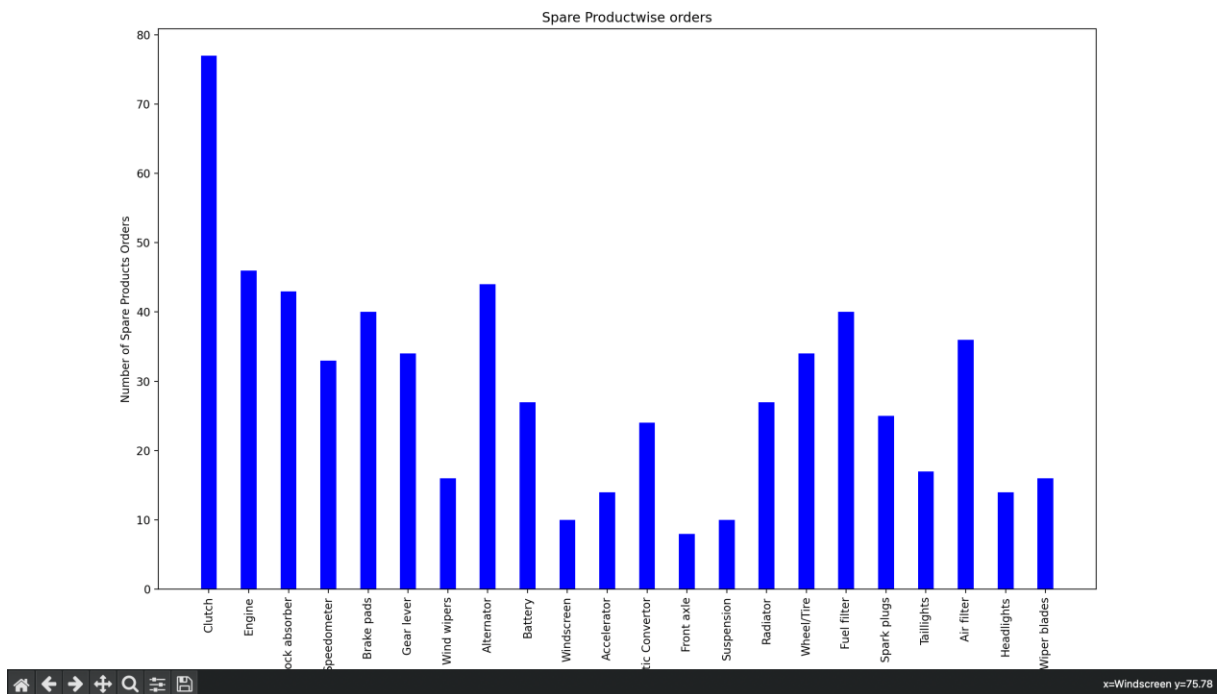
```
pythonProject1 - my_connect.py
main.py x my_connect.py
27 #plt.show()
28
29 plt.bar(customer_list['car_brand'], customer_list['Product_count_ordered'], color='yellowgreen',
30         width=.5)
31 plt.xticks(fontsize=10, rotation='vertical')
32 plt.xlabel("Car Brands")
33 plt.ylabel("Brandwise Spare 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
3	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
5	Wind wipers	16.0
7	Alternator	44.0
8	Battery	27.0
9	Windscreen	10.0
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 // Conf... (58 minutes ago) 45/29 LF UTF-8 4 spaces Python 3.10 (pythonProject1)



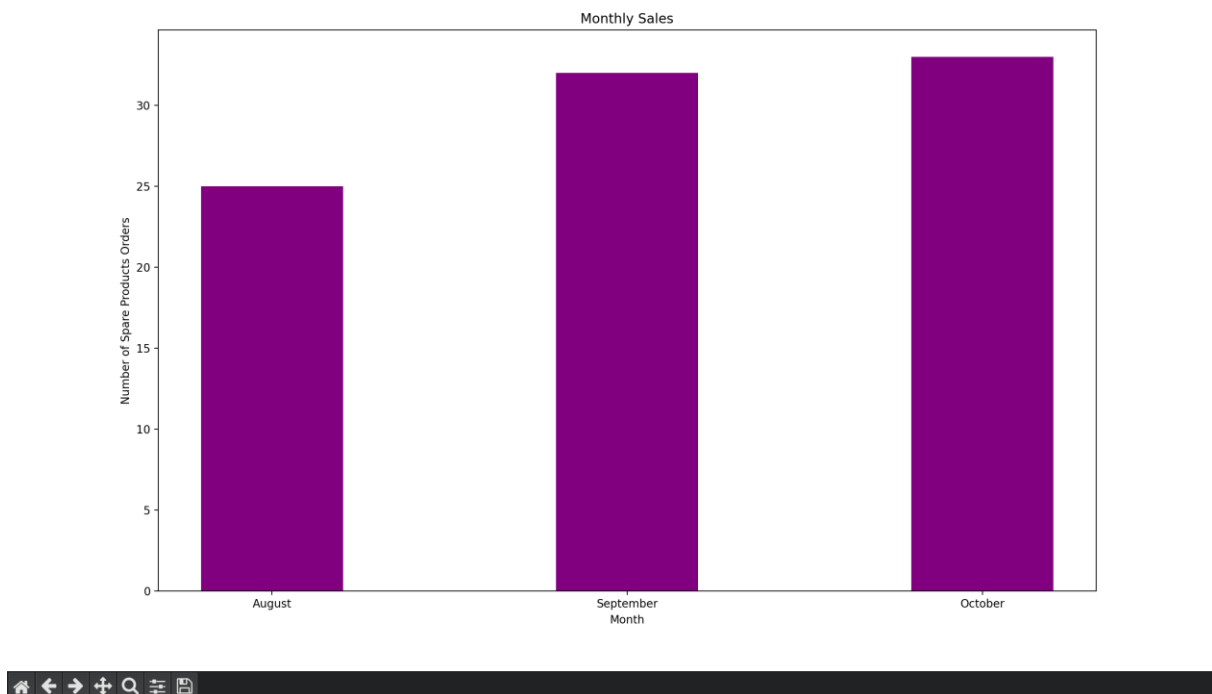
PLOT 3:

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
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
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
19 Air filter 36.0
20 Headlights 14.0
21 Wiper blades 16.0
/Users/ho/PycharmProjects/pythonProject1/my_connect.py:56: UserWarning: pandas only supports SQLAlchemy connectable (engine/connection) or database string URI or sqlite3 DBAPI
monthly_sales = pd.read_sql(query3, userdb)
month count
3 8 25
1 9 32
2 10 33
/Users/ho/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 2022-08-31 1
```

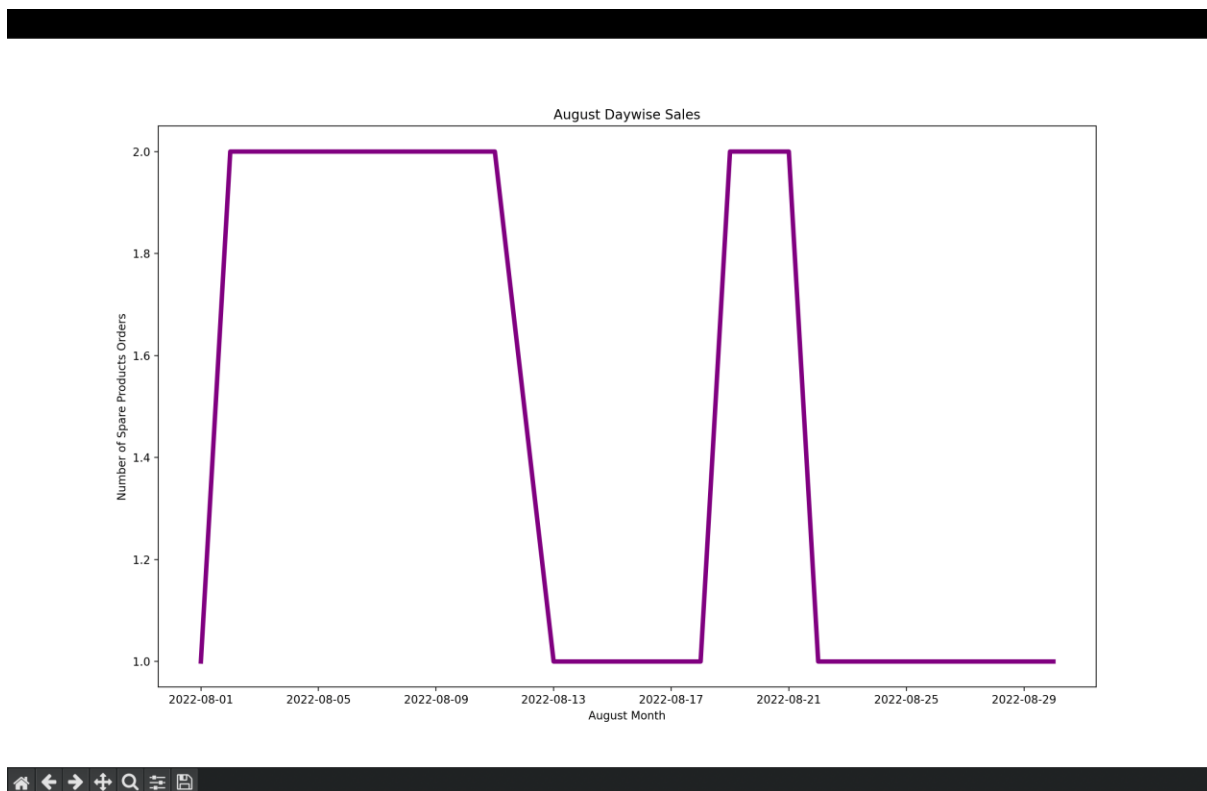


PLOT 4:

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
1: main.py my_connect.py
2: monthly_sales = pd.read_sql(query3, userdb)
3: print(monthly_sales)
4: fig = plt.figure(figsize=(10,10))
5: months = ['August','September','October']
6: plt.bar(monthly_sales['count'],color='purple',width=0.4)
7: plt.xticks(fontsize=10)
8: plt.xlabel("Month")
9: plt.ylabel("Number of Spare Products Orders")
10: plt.title("Monthly Sales")
11: plt.show()
12:
13: 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)'
14: userdbcursor.execute(query4)
15: Aug_orders = userdbcursor.fetchall()
16: august_sales = pd.read_sql(query4, userdb)
17: print(august_sales)
18: fig = plt.figure(figsize=(10,10))
19: plt.plot(august_sales['Date'],august_sales['count'],color='purple',linewidth=4)
20: plt.xticks(fontsize=10)
21: plt.xlabel("August Month")
22: plt.ylabel("Number of Spare Products Orders")
23: plt.title("August Daywise Sales")
24: plt.show()
25:
Run: my_connect
Date count
3 2022-08-01 1
1 2022-08-02 2
2 2022-08-03 2
3 2022-08-08 2
4 2022-08-11 2
5 2022-08-13 1
5 2022-08-14 1
7 2022-08-15 1
8 2022-08-18 1
9 2022-08-19 2
10 2022-08-21 2
11 2022-08-22 1
```



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
main.py my_connect.py
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
77 query5 = 'SELECT DELIVERY_PROVIDER_NAME as Delivery_Service_Provider,DELIVERIES_COUNT_HANDLED as Deliveries_Handled FROM DELIVERY_SERVICE_PROVIDERS'
78 userdbcursor.execute(query5)
79 count_of_deliveries = userdbcursor.fetchall()
80 delivery_count = pd.read_sql(query5, userdb)
81 delivery_count_subset = delivery_count.iloc[:20,:]
82 print(delivery_count_subset)
83 fig = plt.figure(figsize=(10,10))
84 plt.bar(delivery_count_subset['Delivery_Service_Provider'],delivery_count_subset['Deliveries_Handled'],color='blue',width=0.4)
85 plt.xticks(fontsize=10,rotation='vertical')
86 plt.xlabel("Delivery Service Provider")
87 plt.ylabel("Number of Orders Delivered")
88 plt.title("Delivery Service Count")
89 plt.show()

Run: my_connect.py
/Users/De/PycharmProjects/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_count_subset = delivery_count.iloc[:20,:]
0      It 28
1  Y-Solowarm 30
2  Home Ing 33
3  Sonair 61
4  Ronstring 75
5  Ronstring 53
6  Y-rind 46
7  Holdlamis 73
8  Tresom 32
```

