**Project On  
Online Shopping Data Management**

By-Pranav Kumar Varshney

Class-XII-B2

Board Roll No:-

**Certificate**

This is to certify that Pranav Kumar Varshney, student of class XII has successfully completed the project on the topic ‘Online Shopping’ under the guidance of Mrs. Taruna Mongia during the year 2020-21.

(Mrs. Taruna Mongia)

Informatics Practices Teacher

**ACKNOWLEDGEMENT**

I greatly acknowledge the support and the encouragement of my Informatics Practices Teacher, Mrs.Taruna Mongia, whose efforts and teachings helped a lot while completing this practical file. I am also grateful to her for guiding me and giving me suggestions at different stages and helping me alot at every step whenever I needed it while completing this practical file.

I would also like to thank my partner Pranav Kumar Varshney as well who helped and supported me throughout this project and without whom this project would have been impossible.

I also would like to thank my parents for providing me all the technical supplies and being so supportive.

**Objectives of The Project**

**Objectives of *Online Shop Data Management System* are**

* To Make it Easy for the Staff to check Data as they can find it all at one place.
* To Make all the Records Available in a Structured Manner.
* To Make Online Shopping Services 24/7 available.
* To Ease the Retrieval of information and save time.
* To Eliminate Paperwork.
* To Provide Easy Customer Service.
* Easy to Update the Data after any Transaction.
* Redundancy and unnecessary Replication of Data is Eliminated.
* Cost Efficient.
* Operation Efficient

**Introduction**

Online shopping is a form of electronic commerce which allows consumers to directly buy goods or services from a seller over the Internet using a web browser or a mobile app. Consumers find a product of interest by visiting the website of the retailer directly or by searching among alternative vendors using a shopping search engine, which displays the same product's availability and pricing at different e-retailers. As of 2020, customers can shop online using a range of different computers and devices, including desktop computers, laptops, tablet computers and smartphones.

An online shop evokes the physical analogy of buying products or services at a regular "bricks-and-mortar" retailer or shopping center; the process is called business-to-consumer (B2C) online shopping. When an online store is set up to enable businesses to buy from other businesses, the process is called business-to-business (B2B) online shopping. A typical online store enables the customer to browse the firm's range of products and services, view photos or images of the products, along with information about the product specifications, features and prices.

Online stores usually enable shoppers to use "search" features to find specific models, brands or items. Online customers must have access to the Internet and a valid method of payment in order to complete a transaction, such as a credit card, an Interact-enabled debit card, or a service such as PayPal. For physical products (e.g., paperback books or clothes), the e-tailer ships the products to the customer; for digital products, such as digital audio files of songs or software, the e-tailer usually sends the file to the customer over the Internet. The largest of these online retailing corporations are Alibaba, Amazon.com, and eBay.

**Software Specifications**

* Frontend: Jupyter Notebook
* Language: Python 3.8
* Backend: MySQL 5.7 server edition
* Operating System: Windows 10

**Table Structure**

**Table-1**

**Customer Information**

|  |  |  |
| --- | --- | --- |
| ***Column Name*** | ***Data Type*** | ***Constraint*** |
| Customer Name | Char(20) |  |
| Customer Phone No | Bigint(20) |  |
| Customer Id | Char(9) | Primary Key |
| Customer Address | Char(100) |  |
| No. of Order Placed | int(11) |  |
| No. of order Cancelled | int(11) |  |

**Table-2**

**Stock**

|  |  |  |
| --- | --- | --- |
| ***Column Name*** | ***Data Type*** | ***Constraint*** |
| Product Name | Char(20) |  |
| Product ID | Char(7) | Primary Key |
| Stock | Int(11) |  |
| Product Weight | Char(20) |  |
| Product Price | Char(11) |  |

**Table-3**

**Delivery**

|  |  |  |
| --- | --- | --- |
| ***Column Name*** | ***Data Type*** | ***Constraint*** |
| Order Id | Char(7) | Primary Key |
| Delivery Time | Int(11) |  |
| Product Received | Char(1) |  |
| Product Returned | Char(1) |  |
| Customer Id | Char(9) | Foreign Key |
| Product Id | Char(7) | Foreign Key |

**Table-4**

**Bill Information**

|  |  |  |
| --- | --- | --- |
| ***Column Name*** | ***Data Type*** | ***Constraint*** |
| Bill Id | char(7) | Primary Key |
| Amount Paid | int(11) |  |
| Customer Id | char(9) | Foreign Key |
| Order Id | char(7) | Foreign Key |

**Program Code**

import mysql.connector

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

mydb=mysql.connector.connect(host="localhost",user="root",passwd="root",database="online\_shopping")

mycursor=mydb.cursor()

n=input("To Go To Main Menu enter 'Yes',else 'No'")

while(n=='Yes')or (n=='yes'):

m=int(input("Welcome To Main Menu\nEnter 0 To Display Data\nEnter 1 To Add a New Row\nEnter 2 To Update Data\nEnter 3 To Graphically Analyse Data\nEnter 4 To Delete a Row\nEnter 5 to export data in .csv format\nEnter 6 to Logout"))

if(m==0):

print("Type 'C' for Customer Information\nType 'S' for Stock information\nType 'D' for Delivery Info\nType 'B' FOR Bill Info ")

T=input()

if (T=='s')or (T=='S'):

mycursor.execute("select \* from stock")

myrecords=mycursor.fetchall()

print(pd.DataFrame(myrecords,columns=['Product Name','Product Id','Stock','Product Weight','Product Price']))

elif T=='c'or T=='C':

mycursor.execute("select \* from customer\_info")

myrecords=mycursor.fetchall()

print(pd.DataFrame(myrecords,columns=['Customer Name','Customer Phone No','Customer Id','Customer Address','Order Placed','Order Cancelled']))

elif T=='d'or T=='D':

mycursor.execute("select \* from delivery")

myrecords=mycursor.fetchall()

print(pd.DataFrame(myrecords,columns=['Order Id','Delivery Time','Product Recieved','Product Returned','Customer Id','Product Id']))

elif T=='b'or T=='B':

mycursor.execute("select \* from bill\_info")

myrecords=mycursor.fetchall()

print(pd.DataFrame(myrecords,columns=['Bill Id','Amount Paid','Customer Id','Order Id']))

elif m==1:

print("Type 'C' to add a new Row to Customer Information Table\nType 'S' to add a new Row to Stock information Table\nType 'D' to add a new Row to Delivery Info Table\nType 'B' to add a new Row to Bill Info Table")

t=input()

if (t=='s') or (t=='S'):

s="Rs."

Pn=input("Enter Product Name")

Pi=input("Enter Product Id")

S=int(input("Enter Stock amount"))

Pd=input("Enter Weight of the Product in g")

Pr=int(input("Enter Price of The Product"))

mycursor.execute("insert into stock values('"+Pn+"','"+Pi+"','"+str(S)+"','"+Pd+"','"+str(Pr)+"')")

mydb.commit()

elif (t=='c') or (t=='C'):

Cn=input("Enter Customer Name")

Cp=int(input("Enter Customer Phno"))

Ci=input("Enter Customer Id")

Add=input("Customer Address")

OP=input("No. of Order Placed")

OC=input("No. of Order Cancelled")

mycursor.execute("insert into customer\_info values('"+Cn+"','"+str(Cp)+"','"+Ci+"','"+Add+"','"+OP+"','"+OC+"')")

mydb.commit()

elif t=='d' or t=='D':

oi=input("Enter Order Id")

dt=int(input("Enter Delivery Time"))

p=input("Enter y if product was recieved n if product was not recieved")

pr=input("Enter y if returned else n")

Ci=input("Enter Customer id")

pi=input("Enter product id")

mycursor.execute("insert into delivery values('"+oi+"','"+str(dt)+"','"+p+"','"+pr+"','"+Ci+"','"+pi+"')")

mydb.commit()

elif t=='b'or t=='B':

bi=input("Enter Bill Id")

a=int(input("Enter Amount Paid"))

ci=input("Enter Customer Id")

oi=input("Enter Order Id")

mycursor.execute("insert into bill\_info values('"+bi+"','"+str(a)+"','"+ci+"','"+oi+"')")

mydb.commit()

elif m==2:

print("Type 'C' to update data in Customer Information Table\nType 'S' to update data in Stock information Table\nType 'D' to update data in Delivery Info Table\nType 'B' to update Data in Bill Info Table")

t=input()

if t=='s'or t=='S':

mycursor.execute("select \* from stock")

myrecords=mycursor.fetchall()

print(pd.DataFrame(myrecords,columns=['Product Name','Product Id','Stock','Product Weight','Product Price']))

col=input("Enter column name")

data=input("Enter the changed value")

con=input("Enter the product id")

mycursor.execute("update stock set "+col+"='"+data+"' where product\_id='"+con+"';")

mydb.commit()

print("Data succesfully changed")

if t=='c'or t=='C':

mycursor.execute("select \* from customer\_info")

myrecords=mycursor.fetchall()

print(pd.DataFrame(myrecords,columns=['Customer Name','Customer Phone No','Customer Id','Customer Address','Order Placed','Order Cancelled']))

col=input("Enter column name")

data=input("Enter the changed value")

con=input("Enter the customer id")

mycursor.execute("update customer\_info set "+col+"='"+data+"' where customer\_id='"+con+"';")

mydb.commit()

print("Data succesfully changed")

if t=='d' or t=='D':

mycursor.execute("select \* from delivery")

myrecords=mycursor.fetchall()

print(pd.DataFrame(myrecords,columns=['Order Id','Delivery Time','Product Recieved','Product Returned','Customer Id','Product Id']))

col=input("Enter column name")

data=input("Enter the changed value")

con=input("Enter the order id")

mycursor.execute("update delivery set "+col+"='"+data+"' where order\_id='"+con+"';")

mydb.commit()

print("Data succesfully changed")

if t=='b'or t=='B':

mycursor.execute("select \* from bill\_info")

myrecords=mycursor.fetchall()

print(pd.DataFrame(myrecords,columns=['Bill Id','Amount Paid','Customer Id','Order Id']))

data=input("Enter the changed Amount Paid")

con=input("Enter the bill id")

mycursor.execute("update bill\_info set amount\_paid='"+data+"' where bill\_id='"+con+"';")

mydb.commit()

print("Data succesfully changed")

elif m==3:

print("Type 'C' to See which Customer has Placed Most number of orders\nType 'S' for Stock Chart\nType 'B' for Amount Paid Chart ")

T=input()

if (T=='s')or (T=='S'):

mycursor.execute("select \* from stock;")

myrecords=mycursor.fetchall()

x=[]

y=[]

for i in myrecords:

x.append(i[0])

y.append(i[2])

i=np.arange(len(x))

plt.xticks(i,x,rotation=30)

plt.title("Stock Chart")

plt.xlabel("Name of Product")

plt.ylabel("Stock")

plt.bar(x,y)

plt.show()

elif (T=='c')or (T=='C'):

mycursor.execute("select \* from customer\_info;")

myrecords=mycursor.fetchall()

x=[]

y=[]

for i in myrecords:

x.append(i[0])

y.append(i[4])

i=np.arange(len(x))

plt.xticks(i,x,rotation=30)

plt.title("Customer Who Order's The Most")

plt.ylabel("Order Placed")

plt.xlabel("Customer Name")

plt.bar(x,y)

plt.show()

elif (T=='b')or (T=='B'):

mycursor.execute("select \* from bill\_info;")

myrecords=mycursor.fetchall()

x=[]

y=[]

for i in myrecords:

x.append(i[0])

y.append(i[1])

i=np.arange(len(x))

plt.xticks(i,x,rotation=30)

plt.title("Amount Paid Chart")

plt.xlabel("Bill Id")

plt.ylabel("Amount")

plt.bar(x,y)

plt.show()

elif m==4:

print("Type 'C' for Customer Information\nType 'S' for Stock information\nType 'D' for Delivery Info\nType 'B' FOR Bill Info ")

T=input()

if (T=='s')or (T=='S'):

con=input("Enter Product Id")

mycursor.execute("Delete From Stock where Product\_Id='"+con+"'")

if (T=='c')or (T=='C'):

con=input("Enter Customer Id")

mycursor.execute("Delete From Customer\_Id where Customer\_Id='"+con+"'")

if (T=='d')or (T=='D'):

con=input("Enter Order Id")

mycursor.execute("Delete From Delivery where Order\_Id='"+con+"'")

if (T=='b')or (T=='B'):

con=input("Enter Bill Id")

mycursor.execute("Delete From Bill\_Info where Bill\_Id='"+con+"'")

elif m==5:

print("Type 'C' for Customer Information\nType 'S' for Stock information\nType 'D' for Delivery Info\nType 'B' FOR Bill Info ")

T=input()

if (T=='s')or (T=='S'):

mycursor.execute("select \* from stock")

myrecords=mycursor.fetchall()

df=pd.DataFrame(myrecords,columns=['Product Name','Product Id','Stock','Product Weight','Product Price'])

df.to\_csv(r"D:\CSV\Stock.csv")

elif T=='c'or T=='C':

mycursor.execute("select \* from customer\_info")

myrecords=mycursor.fetchall()

df=pd.DataFrame(myrecords,columns=['Customer Name','Customer Phone No','Customer Id','Customer Address','Order Placed','Order Cancelled'])

df.to\_csv(r"D:\CSV\Customer Info.csv")

elif T=='d'or T=='D':

mycursor.execute("select \* from delivery")

myrecords=mycursor.fetchall()

df=pd.DataFrame(myrecords,columns=['Order Id','Delivery Time','Product Recieved','Product Returned','Customer Id','Product Id'])

df.to\_csv(r"D:\CSV\Delivery.csv")

elif T=='b'or T=='B':

mycursor.execute("select \* from bill\_info")

myrecords=mycursor.fetchall()

df=pd.DataFrame(myrecords,columns=['Bill Id','Order Id','Customer Id','Amount Paid'])

df.to\_csv(r"D:\CSV\Bill Info.csv")

elif m==6:

print("You have succesfully Logged Out")

break

else:

print("You have succesfully Logged Out")

**Menu Options**

Menu Option-1

To Display Data

Menu Option-2

To Add a New Row

Menu Option-3

To Update Data

Menu Option-4

To Graphically Analyze Data

Menu Option-5

To Delete a Row

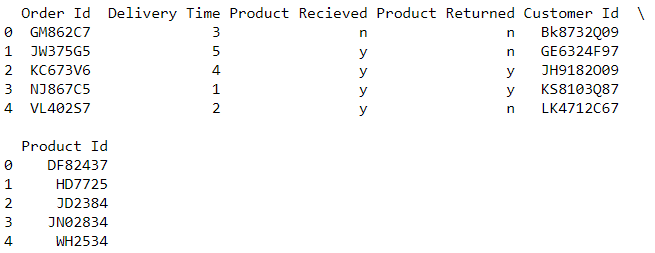
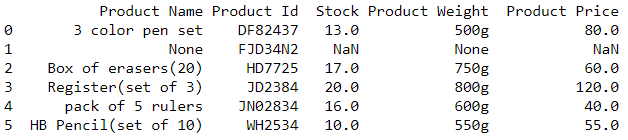
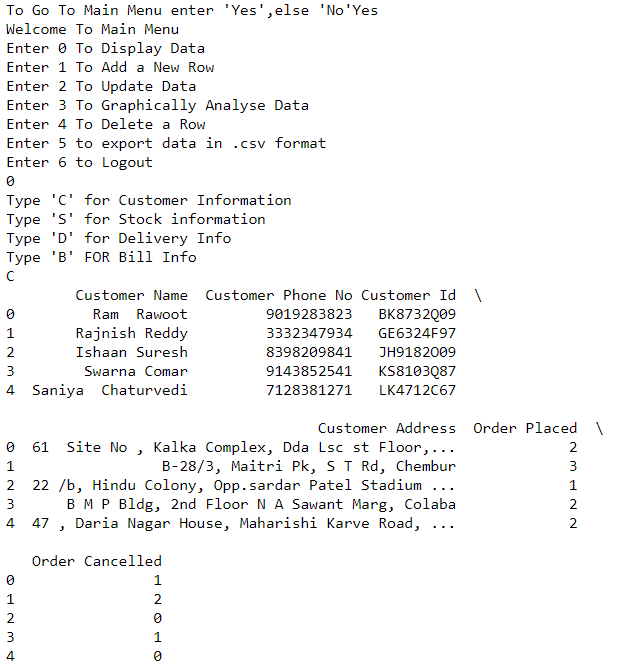
Menu Option-6

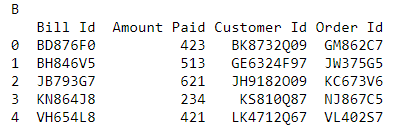
To Export Data to CSV File

Menu Option-7

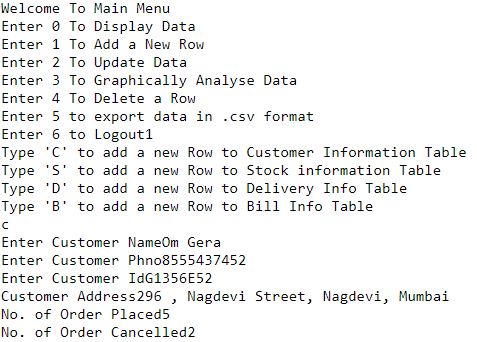
To LogOut

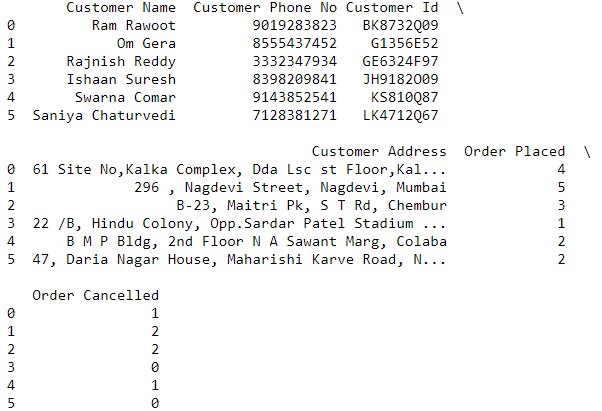
**Outputs**

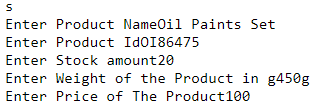
Menu Option 1****

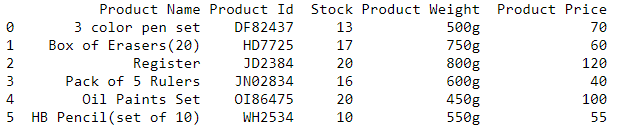
****

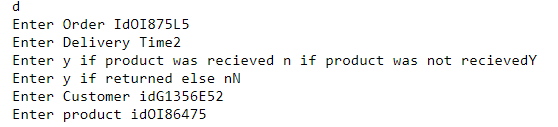
Menu Option 2

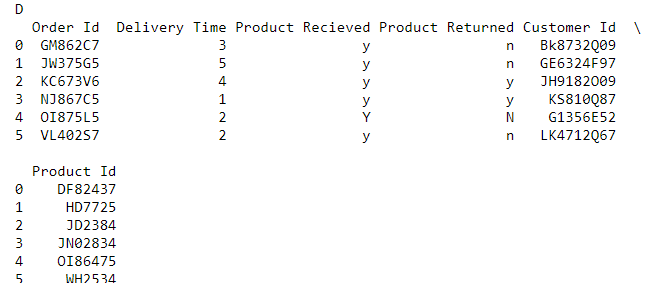
****

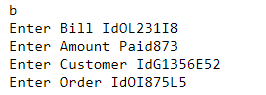
****

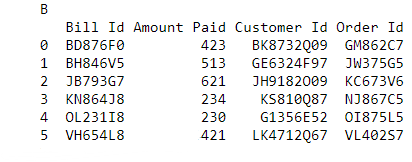
****

****

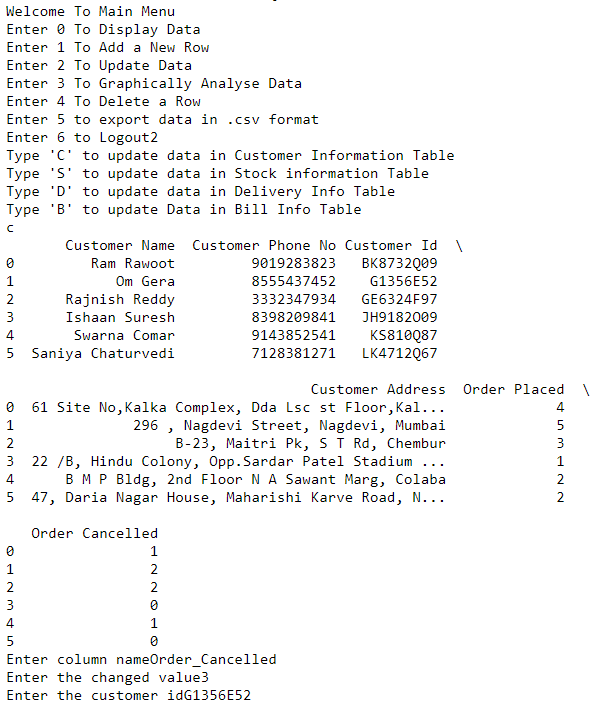
****

****

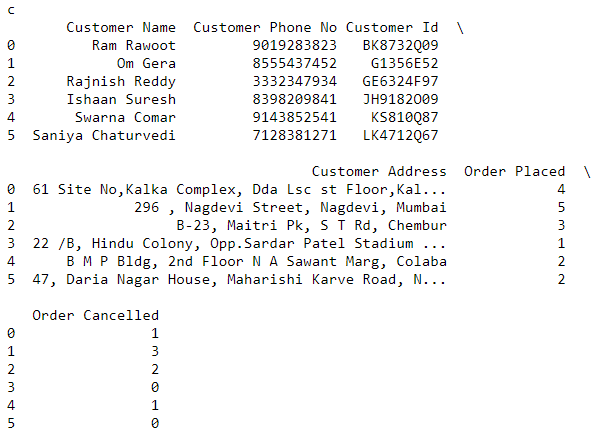
****

****

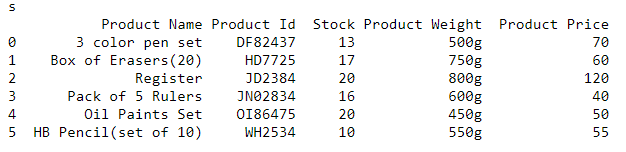
Menu Option 3



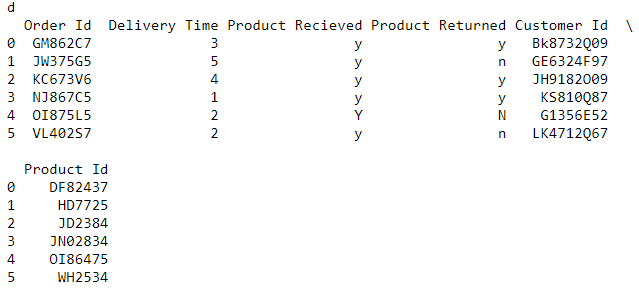




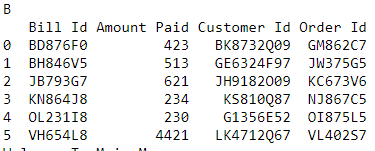


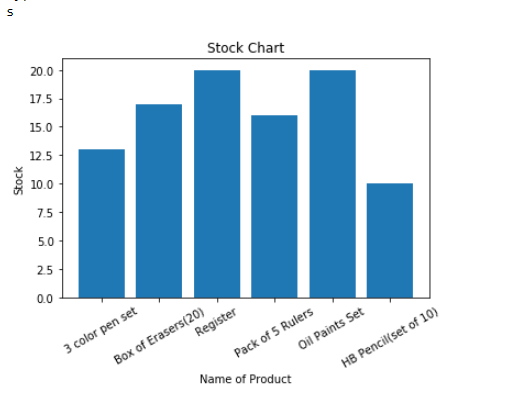
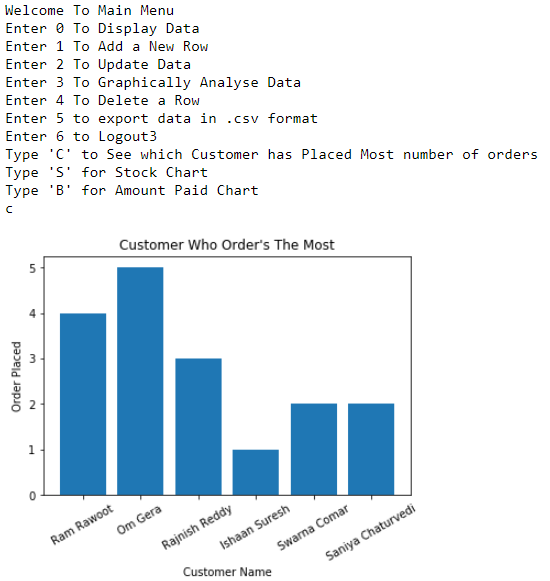


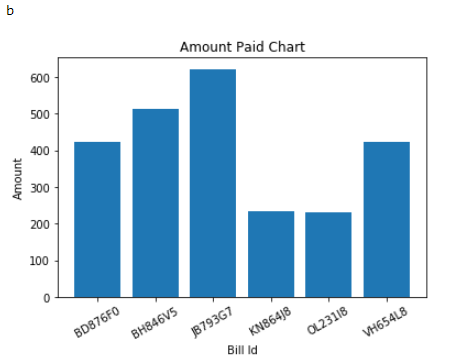




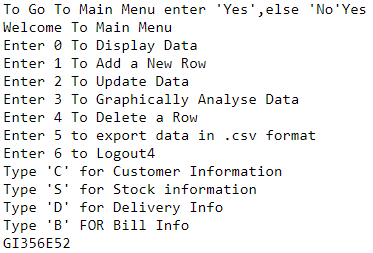


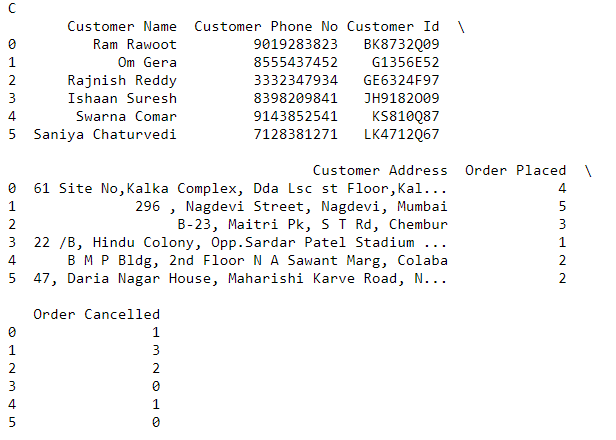


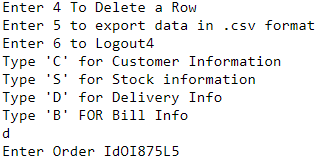
Menu Option 4

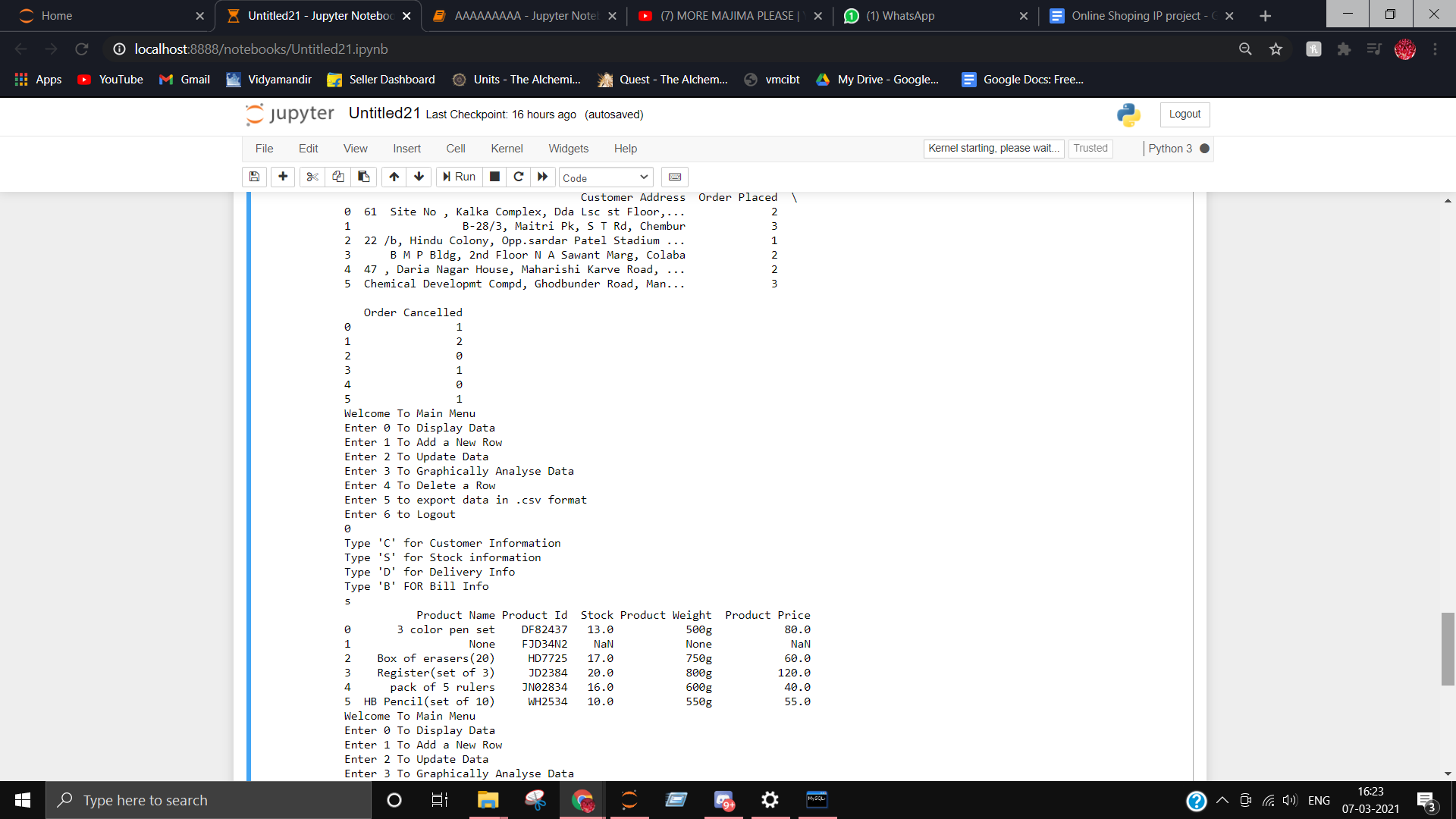
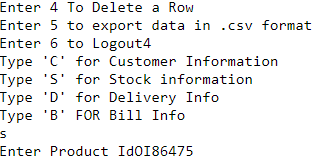
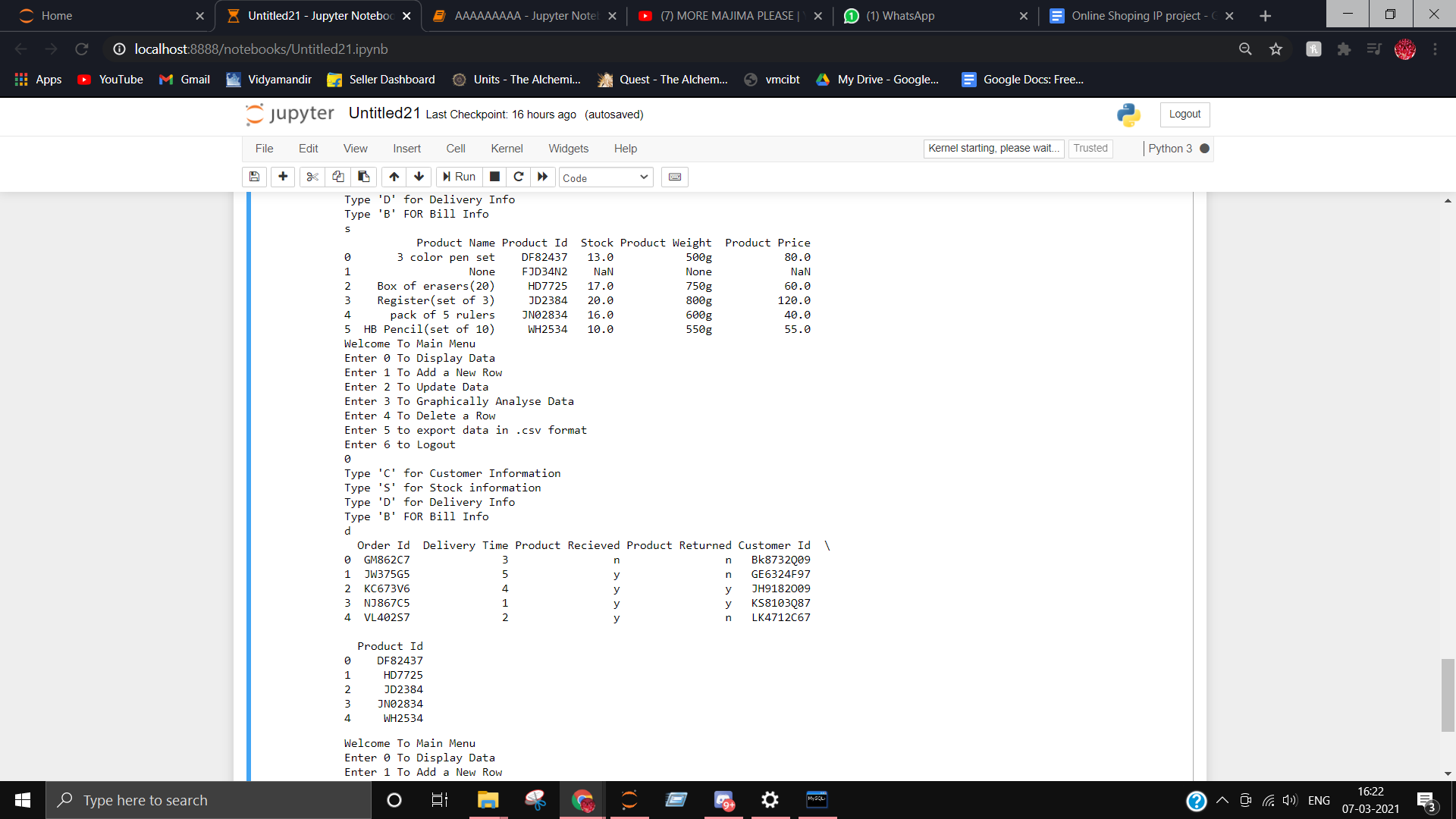


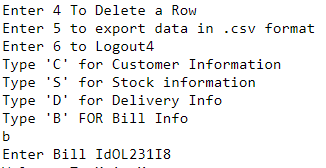
Menu Option 5

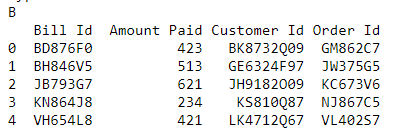




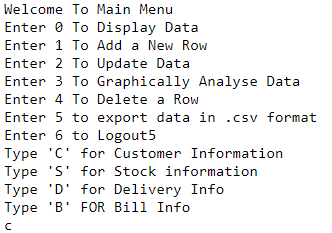


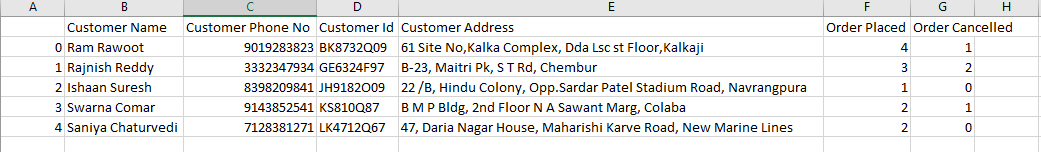


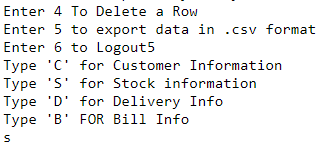


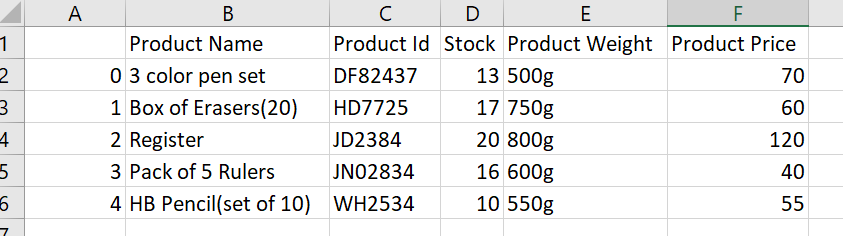


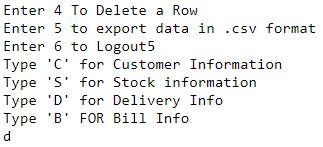
Menu Option 6

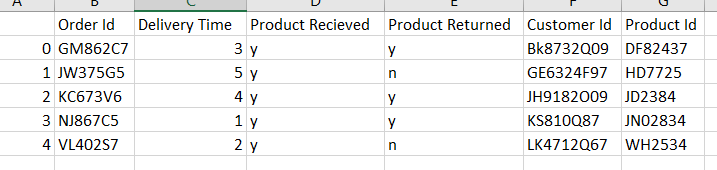


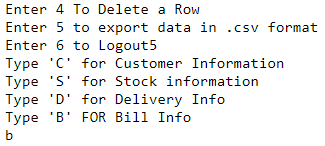


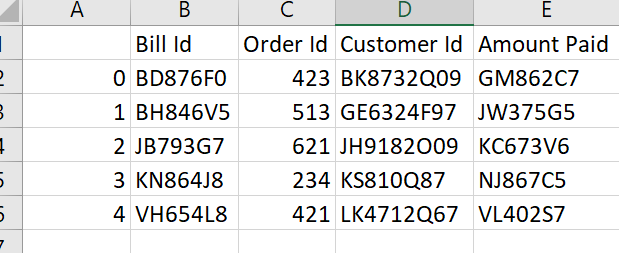












Menu Option 7

