A Project Report On

***SUPER MARKET BILLING***



**Submitted By**

DARSHIL KUMAR

Class: XII A

**Under the Guidance of**

Mr. Satpal Singh

Vanasthali Public School

A-32 B, Sector 56, Noida, Uttar Pradesh-201301

**Contents**

1. **Certificate**
2. **Declaration**
3. **Acknowledgement**
4. **Module Imported**
5. **Coding**
6. **Output**
7. **Bibliography**

**Certificate**

This is to certify that **DARSHIL KUMAR** of Class **XII A** has prepared the report on the Project entitled “**SUPER MARKET BILLING**”. The report is the result of his efforts & endeavours. The report is found worthy of acceptance as final project report for the subject Computer Science of Class XII. He has prepared the report under my guidance.

**Mr. Satpal Singh**

**Computer Teacher**

**Certificate**

The project report entitled “**Super Market Billing**” submitted by **Darshil Kumar** of class **XII A** for the CBSE Senior Secondary Examination class XII of computer Science at **Vanasthali Public School, Mayur Vihar** has been examined.

**SIGNATURE OF EXAMINER**

**Declaration**

I hereby declare that the project work entitled “**Super Market Billing**”, submitted to **Vanasthali Public School, Mayur Vihar**  is prepared by me. All the coding is result of my personal efforts.

**Darshil Kumar**

**Class: XII A**

**Acknowledgement**

I would like to express a deep sense of thanks & gratitude to my project guide **Mr. Satpal Singh Sir** for guiding me immensely through the course of the project. He always evinced keen interest in my work. His constructive advice & constant motivation have been responsible for the successful completion of this project.

I also thanks to my parents for their motivation & support for compilation of this project.

**Last but not the least, I would like to thank all those who had helped directly or indirectly towards the completion of this project.**

**Darshil Kumar**

**Class: XII A**

**Working Description**

* To start the main program the User is asked about the Password which is the Password of SQL i.e. 123456(in this case)
* After this the program creates the database and the table for our program.
* Then the menu appears in which we can choose options for carrying out the mentioned operation.
* The User can view the products which are in the database(Supermarket) using the View Product Option(1).
* The user can also enter the information (Pno, PName, Brand, MRP, SellingP) of the product(Only If there is no entry with same Pno) in database using the Add product option(2).
* The user can also remove a record using Pno, PName or Brand of the product in database using the Remove product option(3).
* The user can also update the pno, name, brand, mrp, sellingp, of the any record using the Update Products option(4).
* The user can also generate billing by adding the pno and quantity of the products the customer had brought and at last it will generate billing with total price, total discount, amount to be paid.

**Module Imported**

* **mysql.connector**
* To establish a connection between python(front end) and mysql(back end) and carry out the queries needed as per the program.
* **datetime**
* To capture system date and time and use it to display current date and time in the program.

**Coding**

import mysql.connector as con

import datetime

while True:

try:

pswd=input("Enter SQL Password:")

dbobj=con.connect(host="localhost",user="root",password=pswd,charset='utf8')

print("Connecting....")

break

except :

print("\*\*\*WRONG PASSWORD\*\*\*")

crsr=dbobj.cursor()

now = datetime.datetime.now()

def dbtable():

crsr.execute("create database if not exists supermarket")

crsr.execute("use supermarket")

crsr.execute("create table if not exists product(PNO int,PNAME char(15),BRAND char(15),MRP int,SELLINGP int,primary key(PNO))")

def intro():

print()

print("\_"\*85)

print("""

DARSHIL SUPEMARKET {}

0)EXIT

1)VIEW PRODUCTS

2)ADD PRODUCT

3)REMOVE PRODUCT

4)UPDATE PRODUCTS

5)GENERATE BILLING

""".format(now.strftime('%d-%m-%Y %H:%M')))

def intinp(stmnt):

y=0

while y==0:

try:

x=input(stmnt)

if bool(int(x))==True:

y=1

return int(x)

except:

print("\*\*\*\*Integer Value Required\*\*\*\*")

def viewproduct():

print()

print("-"\*85)

crsr.execute("desc product;")

recs=crsr.fetchall()

print("%3s %-15s %-15s %5s %5s"%(recs[0][0],recs[1][0],recs[2][0],recs[3][0],recs[4][0]))

crsr.execute("select \* from product;")

recs=crsr.fetchall()

for rec in recs:

print("%3s %-15s %-15s %5s %5s"%(rec[0],rec[1],rec[2],rec[3],rec[4]))

def addproduct():

print("""

====================================ADD PRODUCT======================================

""")

try:

n=intinp("Enter no. of records to be added:")

for w in range (n):

try:

pno=intinp("Enter PNO:")

pname=input("Enter PNAME:")

brand=input("Enter BRAND:")

mrp=intinp("Enter MRP:")

sellingp=intinp("Enter SELLINGP:")

record=(str(pno),pname,brand,str(mrp),str(sellingp))

print(record)

print()

confirm=input("Confirm(y/n):")

print()

try:

if confirm=="y" or confirm=="Y":

crsr.execute("insert into product values {}".format(record))

crsr.execute("commit")

print()

except con.errors.IntegrityError:

print("\*\*\*\*DUPLICATE KEY NOT ALLOWED\*\*\*\*")

except ValueError:

print("\_\_\_\_\_Wrong values\_\_\_\_\_")

except ValueError:

print("\_\_\_\_\_Wrong values\_\_\_\_\_")

def remove(by,val):

try:

crsr.execute("select \* from product where {}='{}'".format(by,val))

rec=crsr.fetchall()

if rec==[]:

print("EMPTY RECORD")

else:

print("Record:")

print(rec)

confirm=input("Confirm(y/n):")

if confirm=="y" or confirm=="Y":

if by=="pno":

crsr.execute("delete from product where pno ={}".format(val))

elif by=="pname":

crsr.execute("delete from product where pname = '{}'".format(val))

elif by=="brand":

crsr.execute("delete from product where brand ='{}'".format(val))

crsr.execute("commit")

print("RECORD DELETED")

return

else:

pass

except:

print("\_\_\_\_\_Wrong values\_\_\_\_\_")

def removeproduct():

print("""

===================================REMOVE PRODUCT====================================

""")

print(""""

MODES:

0)EXIT

1)PNO

2)PNAME

3)BRAND """)

print("\_"\*85)

y=0

data=""

while y==0:

try:

print()

mod=input("Enter Mode:")

if mod=="0":

y=1

break

elif mod=="1" or mod=="pno" or mod=="PNO":

z=1

while z==1:

try:

val=intinp("Enter PNO")

if val==0:

z=0

else:

remove("pno",val)

z=0

except:

print("\_\_\_\_\_Wrong values\_\_\_\_\_")

elif mod=="2" or mod=="PNAME" or mod=="pname":

z=1

while z==1:

try:

data=input("Enter PNAME:")

if data=="0":

z=1

else:

z=0

remove("pname",data)

except:

print("\_\_\_\_\_Wrong values\_\_\_\_\_")

elif mod=="3" or mod=="BRAND" or mod=="brand":

z=1

while z==1:

try:

data=input("Enter BRAND:")

if data=="0":

z=1

else:

z=0

remove("brand",data)

except:

print("\_\_\_\_\_Wrong values\_\_\_\_\_")

else:

print("WRONG MODE SELECTED...")

except ValueError:

print("\_\_\_\_\_Wrong values\_\_\_\_\_")

def billing():

print("""

=======================================BILLING=======================================

""")

viewproduct()

pnolist=[]

qtylist=[]

y=0

sm=0

print("ENTER PNO AS 0 TO EXIT")

print()

while y==0:

try:

no=int(input("Enter PNO:"))

if no==0:

sm=0

disc=0

tdisc=0

smprice=0

print("-"\*85)

crsr.execute("desc product;")

recs=crsr.fetchall()

print("%3s %-15s %-15s %4s %-3s %-5s %5s %5s"%(recs[0][0],recs[1][0],recs[2][0],recs[3][0],"QTY","PRICE","DISC","COST"))

for i in range(len(pnolist)):

crsr.execute("select \* from product where pno={}".format(pnolist[i]))

rec=crsr.fetchone()

cost=int(rec[4])\*qtylist[i]

mrp=int(rec[3])

tmrp=mrp\*qtylist[i]

sm=sm+cost

disc=tmrp-cost

tdisc=tdisc+disc

smprice=smprice+tmrp

print("%3s %-15s %-15s %4s %3s %5s %5s %5s"%(rec[0],rec[1],rec[2],rec[3],qtylist[i],rec[3]\*qtylist[i],disc,cost))

print("%58s %6s"%("TOTAL PRICE:",smprice))

print("%58s %6s"%("TOTAL DISCOUNT:",tdisc))

print("%58s %6s"%("AMOUNT TO BE PAID:",sm))

y=1

break

else:

z=0

while z==0:

crsr.execute("select \* from product where pno={}".format(no))

rec=crsr.fetchone()

if rec==None:

print("No such product available")

z=1

break

qty=int(input("Enter Qty:"))

cost=int(rec[4])\*qty

sm=sm+cost

crsr.execute("desc product;")

recs=crsr.fetchall()

print("%3s %-15s %-15s %5s %8s %5s %4s"%(recs[0][0],recs[1][0],recs[2][0],recs[3][0],recs[4][0],"QTY","COST"))

print("%3s %-15s %-15s %5s %8s %5s %4s"%(rec[0],rec[1],rec[2],rec[3],rec[4],qty,cost))

print()

pnolist.append(no)

qtylist.append(qty)

break

except ValueError:

print("\_\_\_\_\_Wrong values\_\_\_\_\_")

def update(mod,val):

try:

crsr.execute("select \* from product where {}='{}'".format(mod,val))

rec=crsr.fetchall()

if rec==None:

print("EMPTY RECORD")

else:

print("Record:")

print(rec)

confirm=input("Confirm(y/n):")

if confirm=="y" or confirm=="Y":

if mod=="pno":

newrec=intinp("Enter NEW PNO:")

crsr.execute("update product set pno={} where pno={}".format(newrec,val))

elif mod=="pname":

newrec=input("Enter NEW PNAME:")

k="update product set pname= '"+ newrec+"' where pname= '"+val + "'"

crsr.execute(k)

elif mod=="brand":

newrec=input("Enter NEW BRAND:")

k="update product set brand= '"+ newrec+"' where brand= '"+val + "'"

crsr.execute(k)

elif mod=="mrp":

newrec=intinp("Enter NEW MRP:")

crsr.execute("update product set mrp={} where mrp={}".format(newrec,val))

elif mod=="sellingp":

newrec=intinp("Enter NEW SELLINGP:")

crsr.execute("update product set sellingp={} where sellingp={}".format(newrec,val))

crsr.execute("commit")

print("RECORD UPDATED")

return

else:

pass

except:

print("\_\_\_\_\_Wrong values\_\_\_\_\_")

def updateproduct():

print("""

===================================UPDATE PRODUCT====================================

""")

print("""

0)EXIT

1)PNO

2)PNAME

3)BRAND

4)MRP

5)SELLINGP

""")

y=0

data=""

while y==0:

try:

print()

mod=input("Enter Mode:")

if mod=="0":

y=1

break

elif mod=="1" or mod=="pno" or mod=="PNO":

z=1

while z==1:

try:

val=intinp("Enter CURRENT PNO: ")

if val==0:

z=0

else:

update("pno",val)

z=0

except:

print("\_\_\_\_\_Wrong values\_\_\_\_\_")

elif mod=="2" or mod=="PNAME" or mod=="pname":

z=1

while z==1:

try:

data=input("Enter PNAME:")

if data=="0":

z=1

else:

z=0

update("pname",data)

except:

print("\_\_\_\_\_Wrong values\_\_\_\_\_")

elif mod=="3" or mod=="BRAND" or mod=="brand":

z=1

while z==1:

try:

data=input("Enter BRAND:")

if data=="0":

z=1

else:

z=0

update("brand",data)

except:

print("\_\_\_\_\_Wrong values\_\_\_\_\_")

elif mod=="4" or mod=="MRP" or mod=="mrp":

z=1

while z==1:

try:

data=intinp("Enter CURRENT MRP:")

if data=="0":

z=1

else:

z=0

update("mrp",data)

except:

print("\_\_\_\_\_Wrong values\_\_\_\_\_")

elif mod=="5" or mod=="SELLINGP" or mod=="sellingp":

z=1

while z==1:

try:

data=intinp("Enter CURRENT SELLINGP:")

if data=="0":

z=1

else:

z=0

update("sellingp",data)

except:

print("\_\_\_\_\_Wrong values\_\_\_\_\_")

else:

print("WRONG MODE SELECTED...")

except ValueError:

print("\_\_\_\_\_Wrong values\_\_\_\_\_")

def main():

dbtable()

while True:

input()

intro()

option=input("Enter Option")

if option=="1":

viewproduct()

elif option=="2":

addproduct()

elif option=="3":

removeproduct()

elif option=="4":

updateproduct()

elif option=="5":

billing()

elif option=="0":

print("Thanks for visiting...")

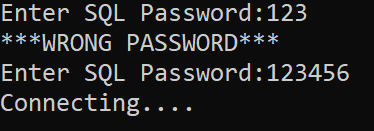
break

else:

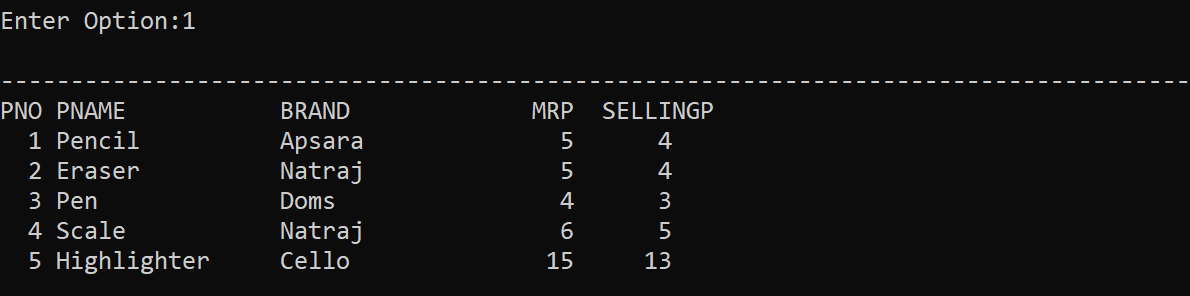
print("Try Again")

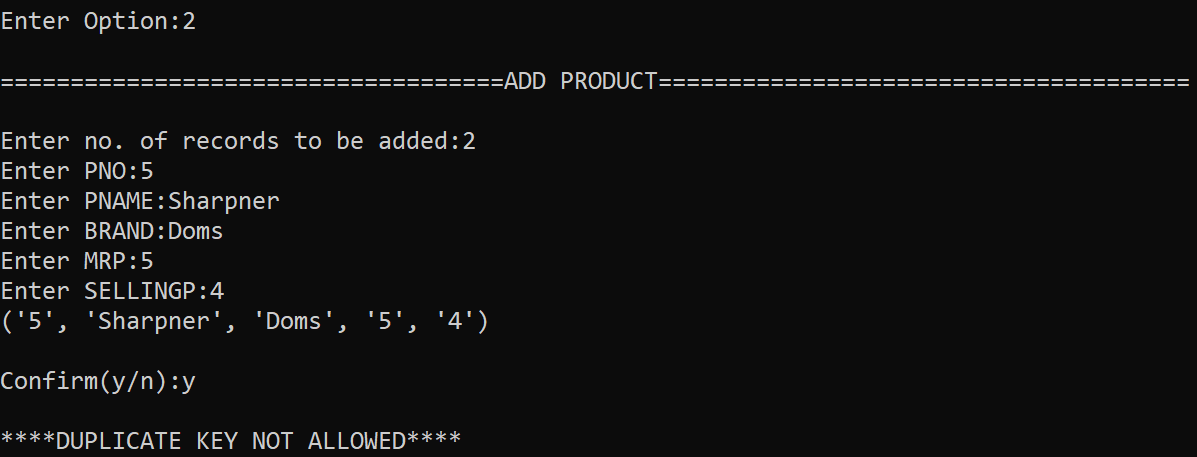
main()

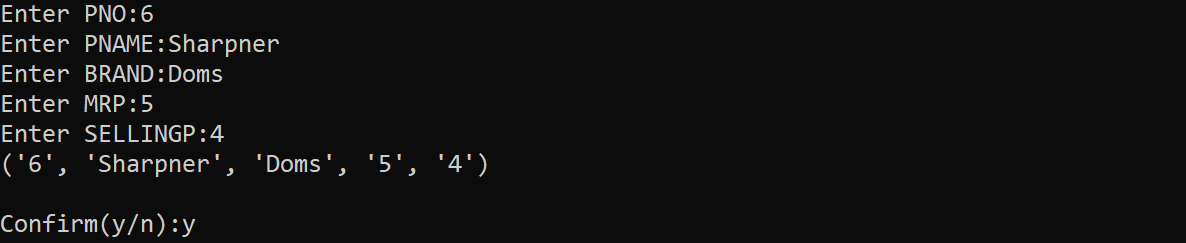
**Output**

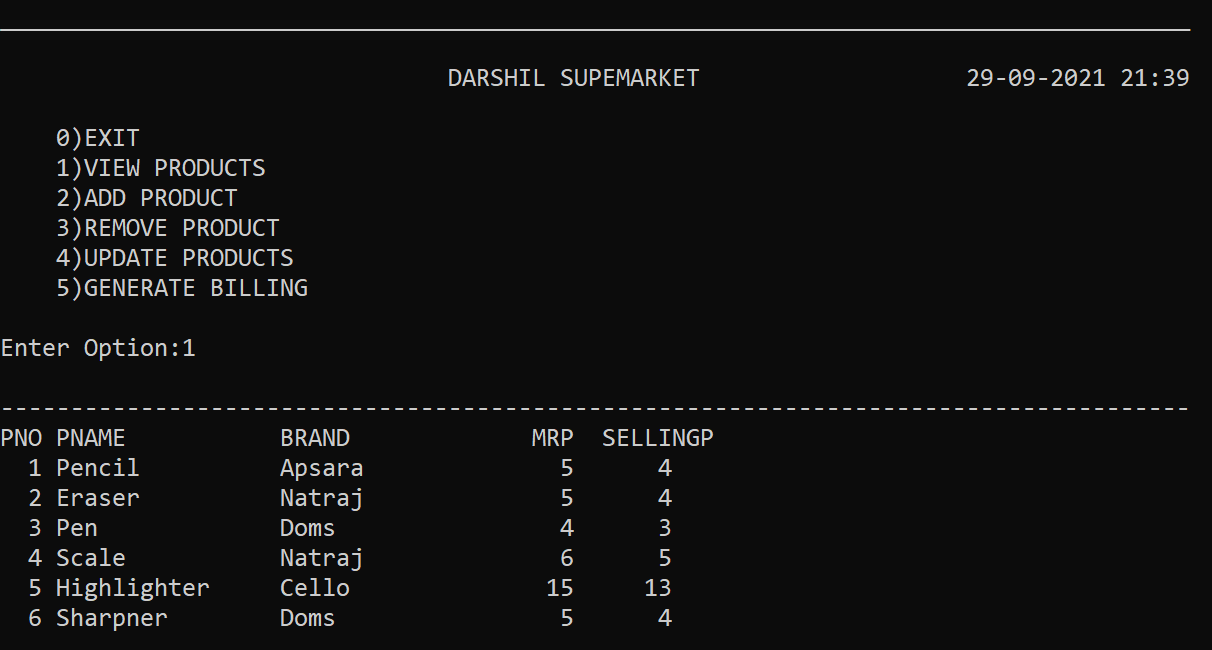


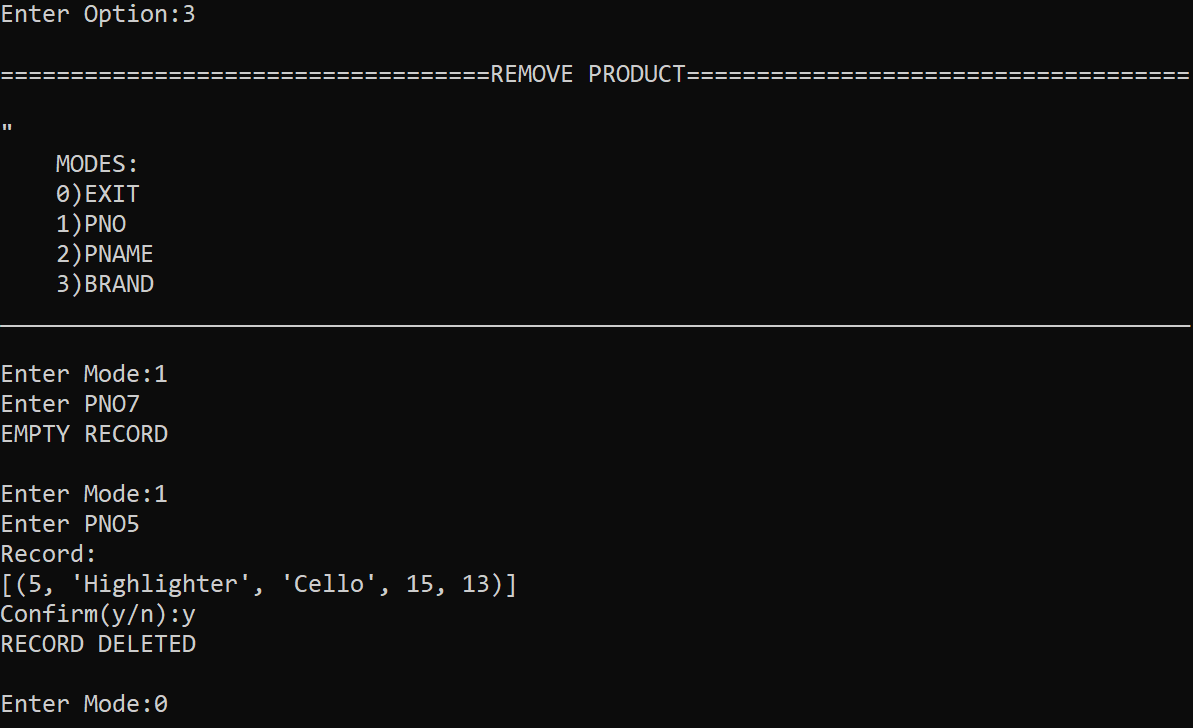


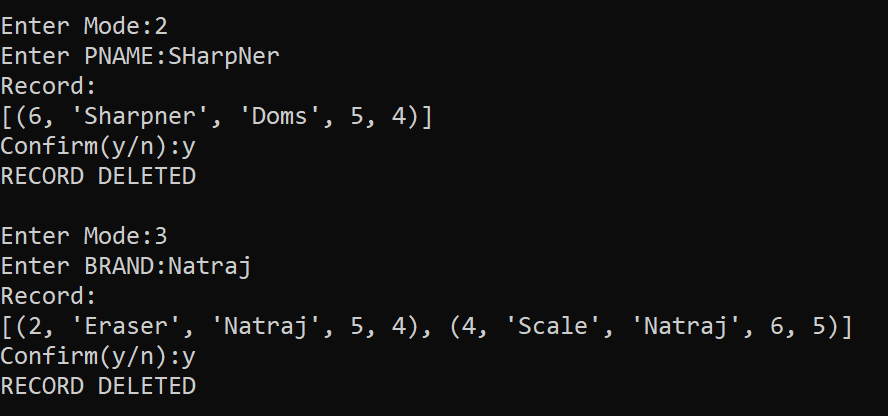


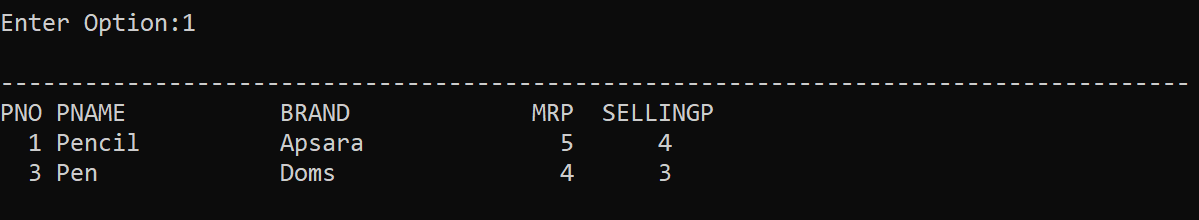


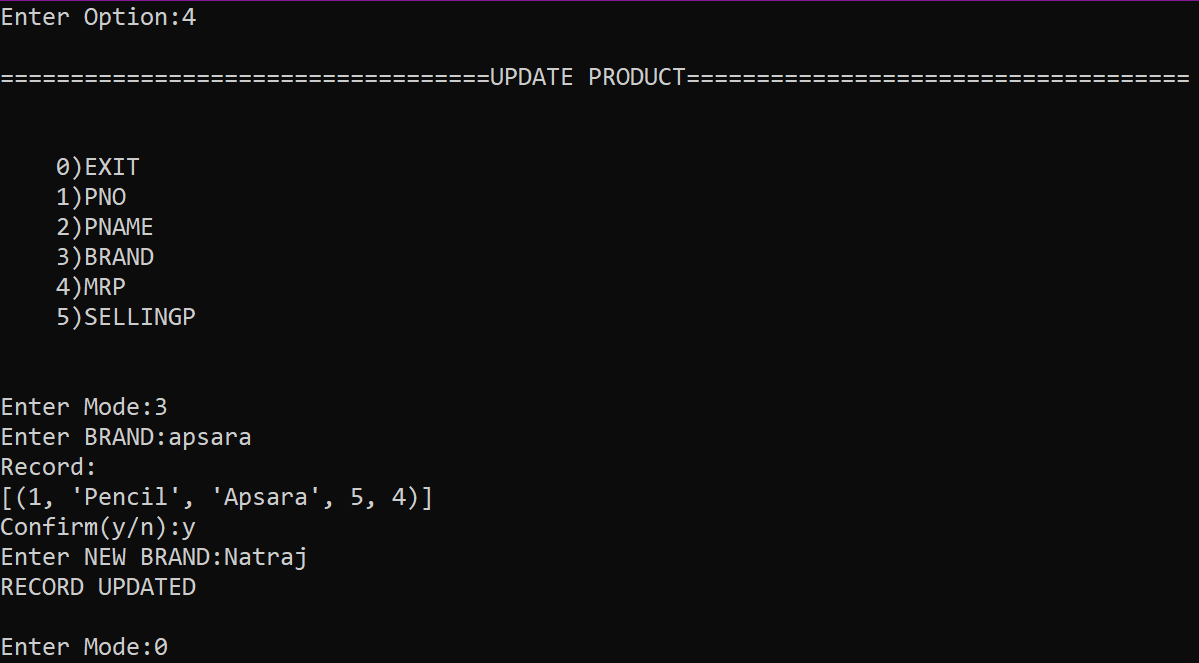


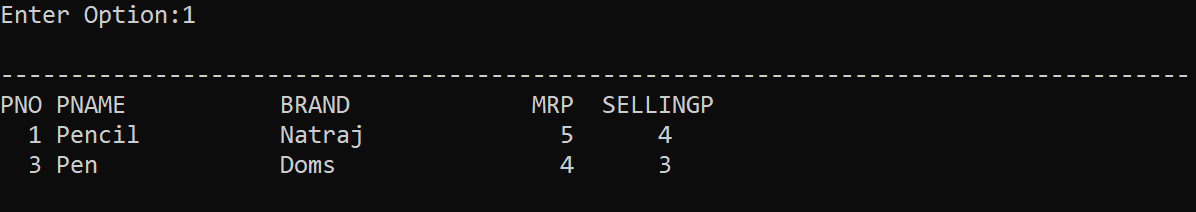


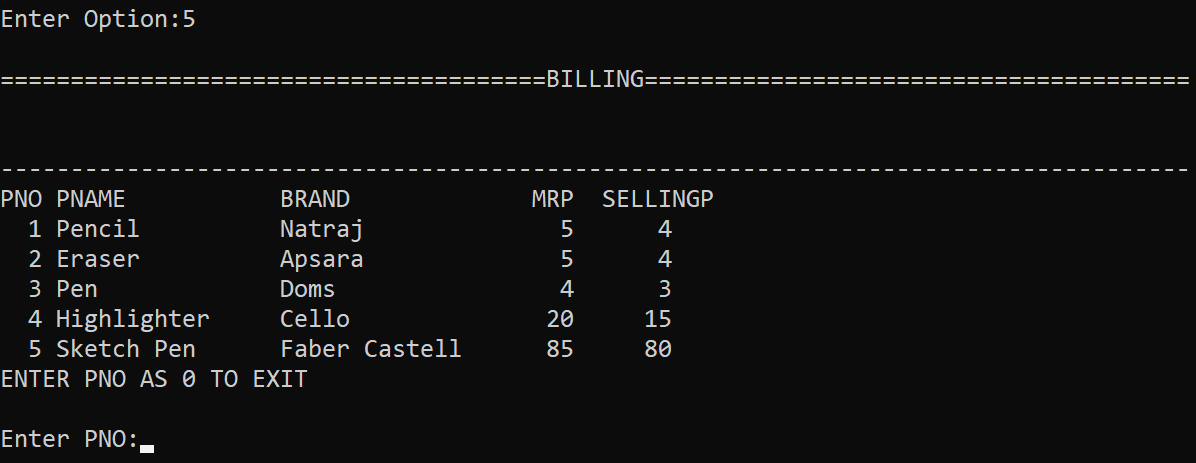


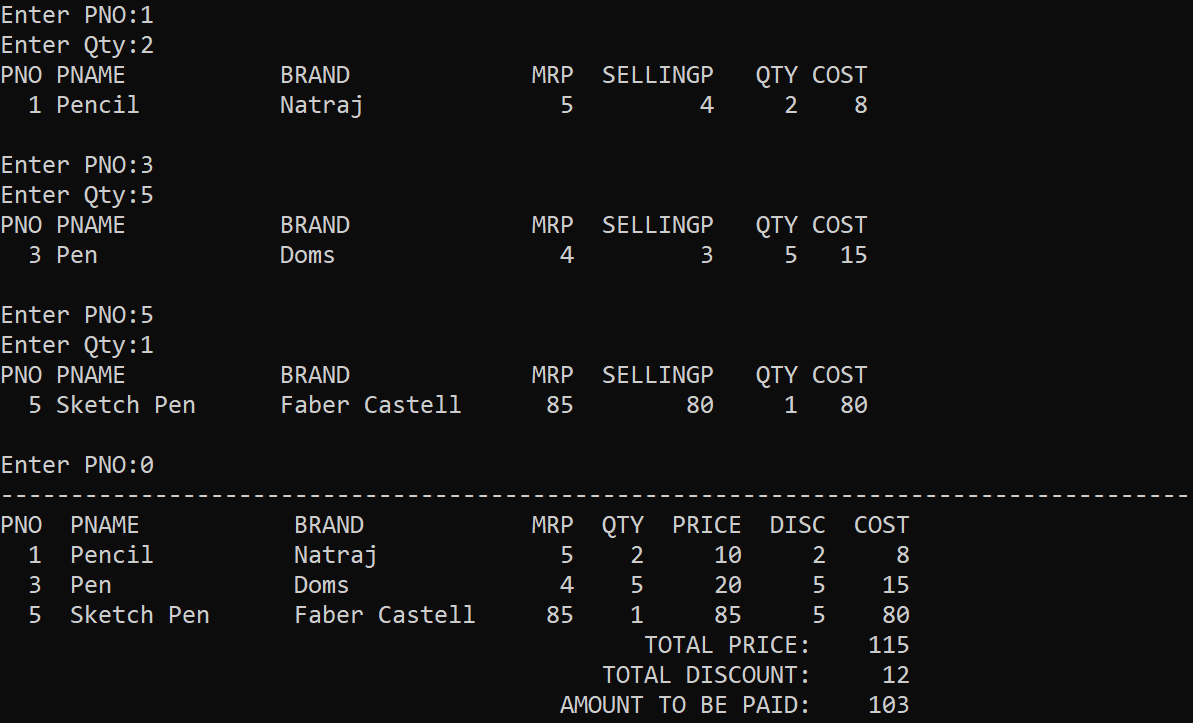












**Bibliography**

* <http://www.google.com/>
* <https://www.w3schools.com/>
* Computer Science with Python by Sumita Arora
* <http://www.slideshare.net/>