By:

NAME:PRACHI PATHAK

STD:12-A

ROLL NO.

***FASHION STORE***

|  |  |
| --- | --- |
| Image result for fashion store logo design |  |

***ACKNOWLEDGEMENT***

***I hereby thank the school and the staff of the computer department for all the guidance and support extended to me during the course of this project.***

***My special thanks are due to the Principal for providing us with a right platform***

***and environment to complete the project.***

***My sincere thanks to Ms. Madhurima Kashyap our Computer science teacher for her able guidance , unflinching support and the affectionate touch without which this project would not have been a success.***

***Finally I thank all those who have directly or indirectly contributed to the success***

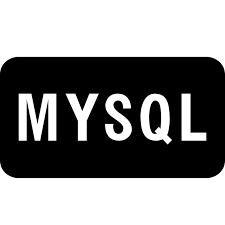
***of this project.***

**HARDWARE REQUIREMENT :**

***LAPTOP,CD***

**SOFTWARE REQUIREMENT :**

***MYSQL,PYTHON,TKINTER***

******

***A database is a systematic collection of data. A database management system is needed to store, access, delete, or otherwise organize data in a database.***

***MySQL is an open source database management system.. You can freely download, modify, and use open source software without having to pay any fees/royalty to the original author.***

***MySQL is a relational database management system (RDBMS), which manages a relational database. A relational database stores information in several tables instead of storing it as a large table. This structure imparts flexibility to the database. MySQL uses the standardized Structured Query Language (SQL) to manage the database.***

***MySQL is developed and distributed by MySQL AB, a company founded by the MySQL developers. In 2008, Sun Microsystems acquired MySQL AB. In 2010, Oracle acquired Sun Microsystems and so MySQL is now owned by Oracle Corp.***

## *. Main Features in MySQL :*

***MySQL was designed to achieve goals such as speed, robustness, and ease of use. The features of MySQL are as follows:***

***Cross-platform compatibility, that is, works on multiple operating systems.***

***Can use multiple CPUs where available, boosting performance.***

***Supports the use of all MySQL operators and functions in the SELECT statement and the WHERE clause.***

***Supports the use of tables from different databases in one statement.***

***Supports table and column aliases.***

***Displays information about databases, tables, and indexes using the SHOW command.***

***Displays query resolution information using the EXPLAIN command.***

***Provides full support for SQL GROUP BY and ORDER BY clauses and group functions as well as left and right outer joins.***

***Supports the use of function names as table or column name.***

## *Advantages of using MySQL:*

***Reliable: Different users have found MySQL even under very heavy loads and large database sizes.***

***Ease of Use: MySQL has a modular and flexible architecture that makes it easy to manage and customize.***

***Cross Platform Support: MySQL is available on more than twenty different operating systems including Linux, UNIX, and Microsoft Windows.***

### ***It’s Incredibly Inexpensive :Depending on what you plan to use it for, a MySQL implementation could***[***range in price from free to***](http://www.mysql.com/products/) ***what you plan to use it for, a MySQL implementation could***[***range in price from free to $10,000***](http://www.mysql.com/products/)***or more. Either way, it’s significantly less expensive than most other database options on the market (save for MySQL’s open-source competitors).***

## *Disadvantages of using MySQL:*

***MySQL does not support a very large database size as efficiently.***

***It’s Got A Few Stability Issues***

***It Suffers From Relatively Poor Performance Scaling.***

**INTRODUCTION:**

***The project “Store Management” has been designed to make the management of the products easier. It has features to update the data, display performance of the store and display the bill of the customer. MySQL and file handling has been used in the project to enhance the performance and efficiency of the project. Overall, “Store Management” is a project which helps the user to handle and keep record of their products in an easier way.***

**AIM AND OBJECTIVE:**

***1. To maintain and store data of the products like***

***type, cost, quantity, sold etc.***

***2. To update data by adding new products or***

***deleting old products.***

***3. To keep track of the progress by displaying***

***performance in format of graph***

***4. To create bill of customers.***

***5. To store in notepad and display bill of customer.***

***6. To provide efficient access to the various data of the product.***

******

***Python is an interpreted, high-level, general-purpose programming language created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. It is dynamically typed and garbage-collected.***

***Python is an object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.***

# **Features and advantages of Python Language :**

## *Simple and easy to Learn : The syntaxes of python language are very simple. Anybody can remember the python language syntaxes, rules and regulations very easily.*

***The elegant syntaxes of the Python language make the people learn python most comfortably.***

***Without having any other programming languages knowledge, we can learn python directly.***

***The simple and powerful syntax of python language makes the programmers express their business logic is fewer lines of code.***

## *Platform Independent: Like Java programs, Python programs are also platform independent.*

***Once we write a Python program, it can run on any platform without rewriting once again.***

***Python uses PVM to convert python code to machine understandable code.***

## *Portability: The python applications or portable. The applications which are developed by python on one platform are going to execute any platform without making any changes (re-compiling is not required).*

***To achieve the portability feature for the every operating system a separate python software is developed for every version.***

## *Free and Open Source & Redistribution: Python is an open source language so that anyone can use the python software without purchasing the licence.*

***Anyone can read the python source code and can able to do the modification in the source code and also we can redistribute that code to others.***

## *High-Level Language: Python is a high-level language.*

***While developing python applications, developers no need to bother about memory management.***

**Disadvantages of python:**

* ***Speed. Python is slower than C or C++.***
* ***Mobile Development. Python is not a very good language for mobile development.***
* ***Memory Consumption. Python is not a good choice for memory intensive tasks.***
* ***Database Access. Python has limitations with database access.***
* ***Runtime Errors.***

**TABLE STRUCTURE**

**create table if not exists inventory**

**(sno int primary key,**

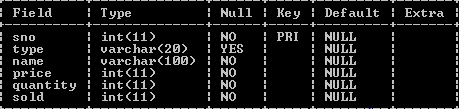
**type varchar(20) check(type in('Tops','Bottoms','Accessories')),**

**name varchar(100) not null**

**price int not null,**

**quantity int not null,**

**sold int not null);**

****

# ***CODE***

# #%%

# import mysql.connector

# con=mysql.connector.connect(host='localhost',user='root',password='arya2002')

# cursor=con.cursor()

# cursor.execute('create database if not exists store;')

# cursor.execute('use store;')

# cursor.execute('''create table if not exists inventory1(sno int primary key,

# type varchar(20) check(type in('Tops','Bottoms','Accessories')),

# name varchar(100) not null,

# quantity int not null,

# costprice int not null,

# saleprice int not null,

# totalcostprice int,

# totalsaleprice int,

# assumedprofit int,

# 

# sold int not null);''')

# import tkinter as tk

# import tkinter.messagebox

# root=tk.Tk()

# root.title('Store management')

# root.geometry('800x220')

# root.configure(background='white')

# qry='''update inventory1 set totalcostprice=costprice\*quantity;'''

# cursor.execute(qry)

# qry='''update inventory1 set totalsaleprice=saleprice\*quantity;'''

# cursor.execute(qry)

# qry='''update inventory1 set assumedprofit=totalsaleprice-totalcostprice;'''

# cursor.execute(qry)

# pdlist=[]

# pdqtylist=[]

# pdpricelist=[]

# pdidlist=[]

# def add():

# 

# sno=e1.get()

# typ=e2.get()

# nm=e3.get()

# qty=e4.get()

# cp=e5.get()

# sp=e6.get()

# sld=e7.get()

# e1.delete(0,tk.END)

# e2.delete(0,tk.END)

# e3.delete(0,tk.END)

# e4.delete(0,tk.END)

# e5.delete(0,tk.END)

# e6.delete(0,tk.END)

# e7.delete(0,tk.END)

# 

# print(sno)

# qry='''select sno from inventory1;'''

# cursor.execute(qry)

# d=cursor.fetchall()

# print(d)

# l=[]

# for i in d:

# l.append(int((i[0])))

# print(l)

# if int(sno) in l:

# tkinter.messagebox.showinfo('Error','Product with same sno already in inventory')

# if typ not in ['Tops','Bottoms','Accessories']:

# tkinter.messagebox.showinfo('Error','Enter a valid product type')

# if qty.isdigit()==False or cp.isdigit()==False or sp.isdigit()==False:

# tkinter.messagebox.showinfo('Error','The quantity,costprice,saleprice should be numbers')

# if cp>sp:

# tkinter.messagebox.showinfo('Error','The saleprice is less than costprice!')

# 

# 

# else:

# 

# w='''insert into inventory1 (sno,type,name,quantity,costprice,saleprice,sold)

# values({},'{}','{}',{},{},{},{})'''.format(sno,typ,nm,qty,cp,sp,sld)

# cursor.execute(w)

# con.commit()

# qry='''select sno from inventory1;'''

# cursor.execute(qry)

# d=cursor.fetchall()

# print(d)

# l=[]

# for i in d:

# l.append(int((i[0])))

# #print(l)

# 

# def display():

# top=tk.Toplevel()

# top.configure(background='lightblue')

# cursor.execute('select\*from inventory1;')

# disp=cursor.fetchall()

# a=''

# for row in disp:

# a+=str(row)

# a+='\n'

# mylabel=tk.Label(top,text=a,bg='white',fg='black')

# mylabel.grid()

# def graph():

# top=tk.Toplevel()

# import matplotlib

# matplotlib.use('TkAgg')

# from matplotlib.figure import Figure

# from matplotlib.backends.backend\_tkagg import FigureCanvasTkAgg

# figure=Figure(figsize=(5,4),dpi=100)

# plot=figure.add\_subplot(1,1,1)

# canvas=FigureCanvasTkAgg(figure,top)

# canvas.get\_tk\_widget().grid(row=0,column=0)

# 

# cursor.execute('select\*from inventory1;')

# dis=cursor.fetchall()

# print(dis)

# 

# import numpy as np

# li=[]

# for i in dis:

# li.append(i[8]) #sold

# name=[]

# for i in dis:

# name.append(i[2])

# print(name)

# ww=np.arange(len(name))

# plot.bar(ww,li,color='r',width=0.1)

# plot.set\_title('Performance')

# plot.set\_xticks(ww)

# plot.set\_xticklabels(name)

# plot.set\_xlabel('Product Name')

# plot.set\_ylabel('Assumed profit')

# def update():

# sno=e1.get()

# typ=e2.get()

# nm=e3.get()

# qty=e4.get()

# cp=e5.get()

# sp=e6.get()

# sld=e7.get()

# e1.delete(0,tk.END)

# e2.delete(0,tk.END)

# e3.delete(0,tk.END)

# e4.delete(0,tk.END)

# e5.delete(0,tk.END)

# e6.delete(0,tk.END)

# e7.delete(0,tk.END)

# q='''update inventory1

# set type='{}',name='{}',quantity={},costprice ={},saleprice={},sold={}

# where sno={};'''.format(typ,nm,qty,cp,sp,sld,sno)

# cursor.execute(q)

# con.commit()

# def deleterec():

# rec=e1.get()

# if int(rec) in l:

# d='''delete from inventory1

# where sno={};'''.format(rec)

# cursor.execute(d)

# con.commit()

# else:

# tkinter.messagebox.showinfo('Error','Entered sno. not in inventory')

# def clearfields():

# e1.delete(0,tk.END)

# e2.delete(0,tk.END)

# e3.delete(0,tk.END)

# e4.delete(0,tk.END)

# e5.delete(0,tk.END)

# e6.delete(0,tk.END)

# clrbutn=tk.Button(root,text='Clear all entry fields',bg='orange',command=clearfields)

# clrbutn.grid(row=0,column=3)

# def bill():

# top=tk.Toplevel()

# top.geometry("1000x700")

# top.title('Billing')

# 

# left=tk.Frame(top,width=500,height=768,bg='white')

# left.grid(row=0,column=0,columnspan=4,rowspan=12)

# 

# right=tk.Frame(top,width=500,height=768,bg='lightblue')

# right.grid(row=0,column=5,columnspan=4,rowspan=12)

# 

# heading=tk.Label(top,text='PERSONA STORE',font=('Arial 15 bold'),bg='white')

# heading.grid(row=0,column=1,columnspan=2)

# 

# pr=tk.Label(top,text='Products',font=('Arial 15 bold'),bg='lightblue',fg='white')

# pr.grid(row=0,column=6)

# 

# qt=tk.Label(top,text='Quantity',font=('Arial 15 bold'),bg='lightblue',fg='white')

# qt.grid(row=0,column=7)

# 

# amt=tk.Label(top,text='Amount',font=('Arial 15 bold'),bg='lightblue',fg='white')

# amt.grid(row=0,column=8)

# 

# ensno=tk.Label(top,text='Enter sno',font=('Arial 15 bold'),bg='white')

# ensno.grid(row=1,column=0)

# 

# ensnoe=tk.Entry(top,width=25)

# ensnoe.grid(row=1,column=1)

# ensnoe.focus()

# 

# 

# prname=tk.Label(top,text='',font=('Arial 15 bold'),bg='white')

# prname.grid(row=3,column=0)

# 

# pprice=tk.Label(top,text='',font=('Arial 15 bold'),bg='white')

# pprice.grid(row=4,column=0)

# 

# def ajax():

# a=int(ensnoe.get())

# 

# qry1='''select sno from inventory1'''

# cursor.execute(qry1)

# er=cursor.fetchall()

# #print(er)

# sn=[]

# for i in er:

# sn.append(i[0])

# # print(sn)

# if a not in sn:

# tkinter.messagebox.showinfo('Error','Invalid Serial No.')

# else:

# print('Search button working')

# 

# 

# 

# t=()

# t+=(a,)

# print(t)

# qry='''select name,saleprice

# from inventory1 where sno={};'''.format(a)

# #getsno=ensnoe.get()

# cursor.execute(qry)

# re2=cursor.fetchall()

# 

# 

# print(re2)

# for r in re2:

# getnm=r[0]

# getprice=r[1]

# prname.configure(text='product name:'+str(getnm))

# pprice.configure(text='Price'+str(getprice))

# 

# qtyl=tk.Label(top,text='Enter qty:',font=('Arial 15 bold'),bg='white')

# qtyl.grid(row=5,column=0)

# 

# qtye=tk.Entry(top,text='Enter qty:',font=('Arial 15 bold'),bg='white')

# qtye.grid(row=5,column=1)

# qtye.focus()

# changel=tk.Label(top,text='Given Amount:',font=('Arial 15 bold'),bg='white')

# changel.grid(row=8,column=0)

# 

# changee=tk.Entry(top,width=15)

# changee.grid(row=8,column=1)

# 

# 

# 

# billbtn=tk.Button(top,text='Generate bill',width=20,height=2,bg='orange')

# billbtn.grid(row=11,column=1)

# 

# def finorder():

# qtyreq=int(qtye.get())

# pdid=ensnoe.get()

# 

# qry='''select quantity from inventory1 where sno={}'''.format(pdid)

# cursor.execute(qry)

# d=cursor.fetchall()

# #print(d[0][0 ])

# qtyinstock=int(d[0][0])

# if qtyinstock<qtyreq:

# tkinter.messagebox.showinfo('Error','Not enough item in stock.')

# 

# else:

# 

# finalprice=float(qtyreq)\*float(getprice)

# if pdid in pdidlist:

# tkinter.messagebox.showinfo('Error','Item is already added')

# else:

# 

# pdlist.append(getnm)

# pdqtylist.append(qtyreq)

# pdpricelist.append(finalprice)

# pdidlist.append(pdid)

# print(pdlist)

# print(pdqtylist)

# print(pdpricelist)

# count=0

# 

# for p in pdlist:

# count+=1

# tempname=tk.Label(top,text=str(p),font=('arial 15 bold'),bg='lightblue',fg='white')

# tempname.grid(row=0+count,column=6)

# count=0

# for p in pdqtylist:

# count+=1

# tempname=tk.Label(top,text=str(p),font=('arial 15 bold'),bg='lightblue',fg='white')

# tempname.grid(row=0+count,column=7)

# count=0

# for p in pdpricelist:

# count+=1

# tempname=tk.Label(top,text=str(p),font=('arial 15 bold'),bg='lightblue',fg='white')

# tempname.grid(row=0+count,column=8)

# totprice=sum(pdpricelist)

# totallabel=tk.Label(top,text='Total:'+str(totprice),bg='lightblue',font=('arial 15 bold'),fg='white')

# totallabel.grid(row=10,column=8)

# def change():

# givenamt=changee.get()

# if float(givenamt)<totprice:

# tkinter.messagebox.showinfo('Error','Amount received is less than price')

# else:

# change=float(givenamt)-totprice

# changecal=tk.Label(top,text='Change:'+str(change),bg='lightblue',fg='white',font=('Arial 15 bold'))

# changecal.grid(row=10,column=7)

# 

# changebtn=tk.Button(top,text='Calculate change',width=15,height=2,bg='orange',command=change)

# changebtn.grid(row=9,column=1)

# buybtn=tk.Button(top,text='Add to cart',width=15,height=2,bg='orange',command=finorder)

# buybtn.grid(row=7,column=1)

# 

# totallabel=tk.Label(top,text='',bg='lightblue',font=('arial 15 bold'),fg='white')

# totallabel.grid(row=10,column=6)

# searchbtn=tk.Button(top,text='Search',width=15,height=1,bg='orange',command=ajax)

# searchbtn.grid(row=2,column=1)

# cursor.execute('select\*from inventory1;')

# dis=cursor.fetchall()

# 

# e1=tk.Entry(root,width=50)

# e1.grid(row=0,column=1)

# e2=tk.Entry(root,text='Enter pr type',width=50)

# e2.grid(row=2,column=1)

# e3=tk.Entry(root,text='Enter pr name',width=50)

# e3.grid(row=4,column=1)

# e4=tk.Entry(root,text='Enter qty',width=50)

# e4.grid(row=6,column=1)

# e5=tk.Entry(root,text='Enter costprice',width=50)

# e5.grid(row=8,column=1)

# e6=tk.Entry(root,text='Enter saleprice',width=50)

# e6.grid(row=10,column=1)

# e7=tk.Entry(root,text='Enter qty sold',width=50)

# e7.grid(row=12,column=1)

# e1lab=tk.Label(root,text='Enter sno:',bg='white',fg='black')

# e1lab.grid(row=0,column=0)

# e2lab=tk.Label(root,text='Enter pr type:',bg='white',fg='black')

# e2lab.grid(row=2,column=0)

# e3lab=tk.Label(root,text='Enter pr nmae :',bg='white',fg='black')

# e3lab.grid(row=4,column=0)

# e4lab=tk.Label(root,text='Enter qty',bg='white',fg='black')

# e4lab.grid(row=6,column=0)

# e5lab=tk.Label(root,text='Enter cost price:',bg='white',fg='black')

# e5lab.grid(row=8,column=0)

# e6lab=tk.Label(root,text='Enter sale price:',bg='white',fg='black')

# e6lab.grid(row=10,column=0)

# e7lab=tk.Label(root,text='Enter qty sold:',bg='white',fg='black')

# e7lab.grid(row=12,column=0)

# mybutton=tk.Button(root,text='Click me to display items',state='active',command=display,bg='orange')

# mybutton.grid(row=14,column=1)

# mybutton2=tk.Button(root,text='Click after entering sno to delete record',state='active',command=deleterec,bg='orange')

# mybutton2.grid(row=14,column=3)

# mybutton1=tk.Button(root,text='Click to add new items',state='active',command=add,bg='orange')

# mybutton1.grid(row=14,column=0)

# mygraphbtn=tk.Button(text='Click to see the performance graph',command=graph,bg='orange')

# mygraphbtn.grid(row=16,column=3)

# mybutton3=tk.Button(root,text='Click to update the data for entered sno.',command=update,bg='orange')

# mybutton3.grid(row=16,column=0)

# mybutton4=tk.Button(root,text='Click to generate a bill',bg='orange',command=bill)

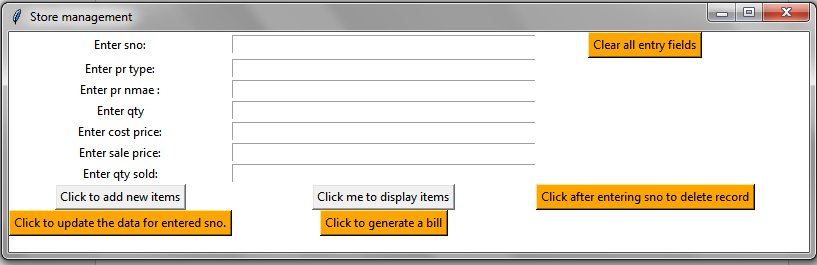
# mybutton4.grid(row=16,column=1)

# root.mainloop()

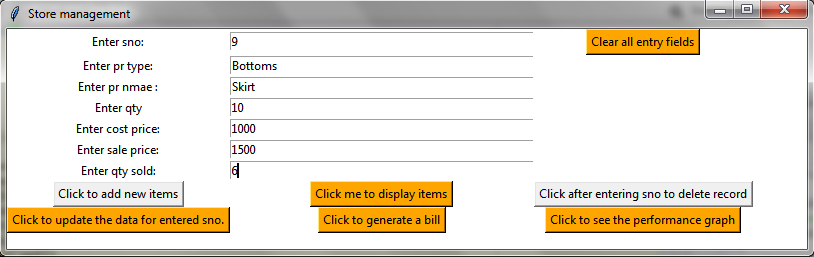
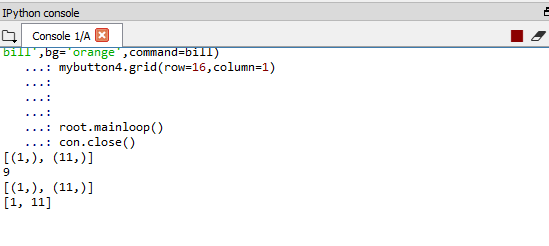
# con.close()

# **OUTPUT**

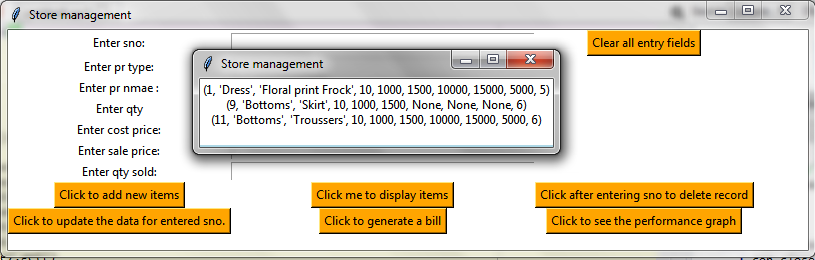
1. **STORE MANAGEMENT :**



1. **OUTPUT FOR ADDING THE PRODUCT:**

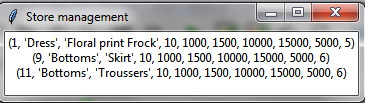


1. **OUTPUT FOR DISPLAYING ITEMS:**

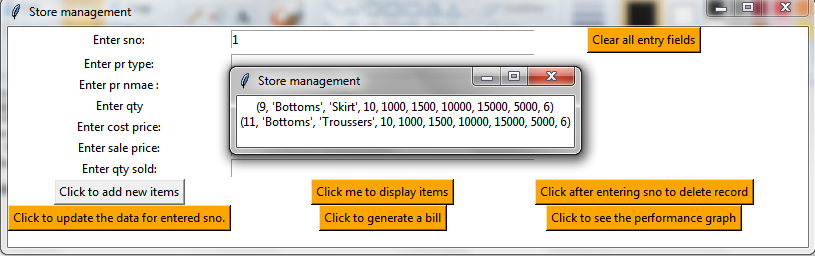
****

1. **OUTPUT FOR DELETING THE RECORD:**

**Before deleting record**

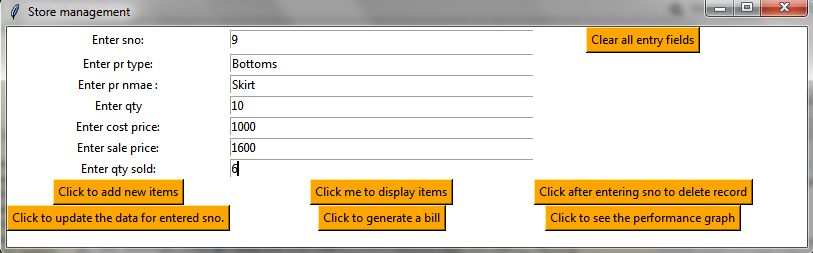
****

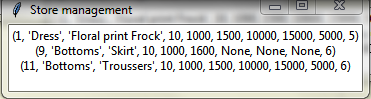
**After deleting record**

****

1. **OUTPUT FOR UPDATING THE DATA:**

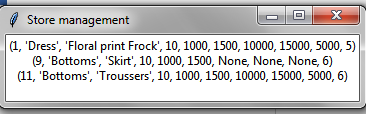
**Record entered:**

****

****

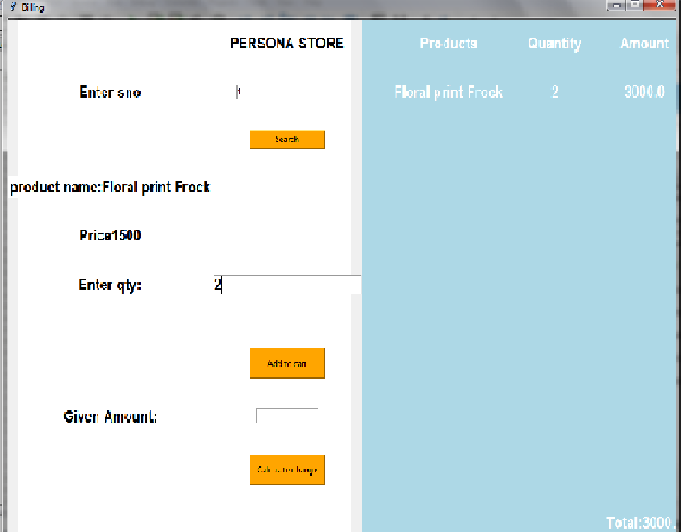
**Record after updating it:**

****

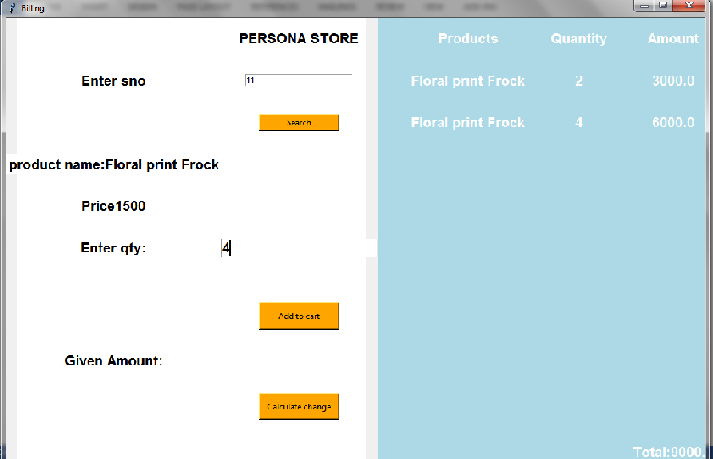
****

1. **OUTPUT FOR GENERATING A BILL:**

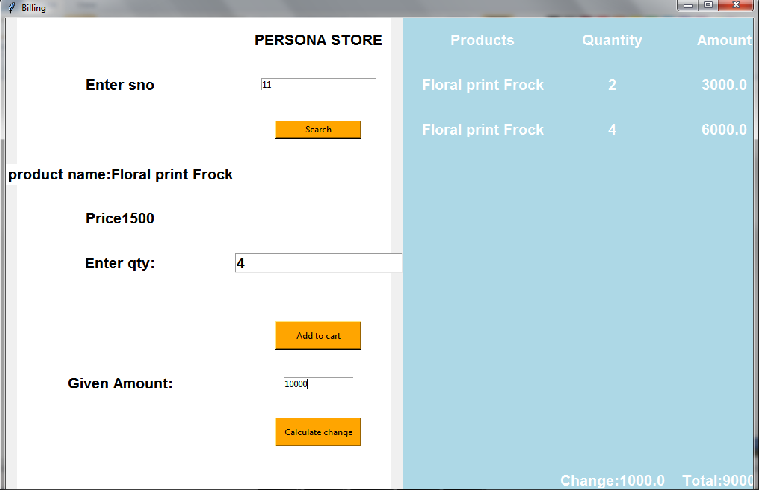
**Entered 1st product :**

****

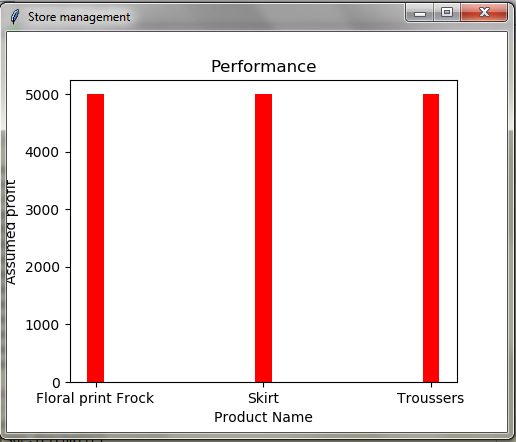
**Entered 2nd product:**

****

**Total bill for both products along with the change displayed:**

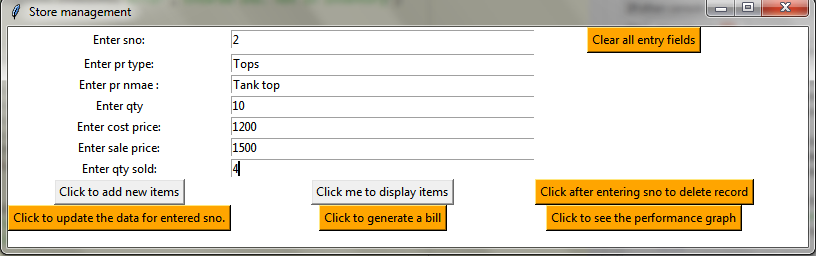
****

1. **OUTPUT FOR THE PERFORMANCE GRAPH:**



1. **OUTPUT FOR CLEARING THE RECORD:**

**Before**

****

**After**

****

# **DECLARATION**

I PRACHI PATHAK, Roll No. a student of DPS , Navi Mumbai humbly submit that I have completed the project work as described in this report by my own skill and study as per the instructions of my teacher Ms. Madhurima Kashyap and that I have not copied the report or its any appreciable parts from any other literature in contravention of the academic ethics.

DATE: SIGNATURE OF THE STUDENT: