KENDRIYA VIDYALAYA

KANJIKODE WEST, PALAKKAD.

CLASS XII A

COMPUTER SCIENCE PROJECT

“GROCERY STORE MANAGEMENT“

DEPARTMENT OF COMPUTER SCIENCE

2021-2022

KENDRIYA VIDYALAYA

KANJIKODE WEST, PALAKKAD.

COMPUTER SCIENCE PROJECT

2021-2022

This is certified to be the bonafied record of the work done by S GIRISH of class XII A Computer Science in the C.S lab during the year 2021-2022.

Submitted for the All India Senior School Certificate Examinationto be held on …… at KENDRIYA VIDYALAYA KANJIKODE WEST, PALAKKAD.

PRINCIPAL Teacher In charge

Internal Examiner External Examiner

ACKNOWLEDGEMENT

I would like to extend my sincere thanks to my Computer Science teacher , head of the department of computer science who has given this wonderful opportunity to do this project of C.S ‘GROCERY STORE MANAGEMENT.

I would like to express my gratitude to our principal who has been there with us supporting in all possible ways .I would like to thank all my teachers and friends who have helped me to make this project a grand success.

INDEX

INTRODUCTION

Python is an easy to use yet powerful object oriented programming language. It is very high level programming language yet as powerful as many other middle level not so high-level language like C++, Java etc..

With the help of Graphical User Interface (GUI) modules like Tkinter, it becomes extremely powerful from creation of buttons to getting input from the user...

This project ‘GROCERY STORE MANAGEMENT basically features file operations and working of files along with storing data in SQL database , graphical user interface, stack,

functions and also features basic of python like List, Tuple and String.

This project would be helpful for running a Store with many items where keeping track of expiry date is a problem, This program automatically finds the item expired from your cart and displays a warning. This program contains all tricks and code blocks from both class 11, 12 and more…

REQUIREMENTS

PROCESSOR: 2.5 GHz

RAM: MINIMUM 4 GB

OPERATING SYST : WINDOWS 7 AND ABOVE

SYSTEM TYPE : 32 OR 64 BIT

Source code

```
from tkinter import *
import mysql.connector as sql
import datetime
import csv
import pickle
login = Tk()
user_ = simpledialog.askstring("LOGIN", "Enter the user name :")
passwd = simpledialog.askstring("LOGIN", "Enter the password :")
if user_=="admin" and passwd=="admin":
   login.destroy()
   db=sql.connect(host="localhost",user="root",passwd="qweasd",database="GSM")
   crsr=db.cursor()

   def addstock(id_,stock):
       global db
       global crsr
       crsr.execute(f"Update stock set stock=stock+{stock} where id={id_}")
       db.commit()

   def modifystock(id_,name,stock,MRP,exp_date,vendor,BatchNo=None):
       crsr.execute(f"Delete FROM stock WHERE id={id_}")
       crsr.execute(f"INSERT INTO Stock VALUES({id_},'{name}',{stock},
{MRP},'{BatchNo}','{exp_date}','{vendor}')")
       db.commit()

   def deletestock():
      crsr.execute(f"Delete FROM stock WHERE id={id_}")
      db.commit()
   def displaystock():
      crsr.execute("SELECT * FROM STOCK")
      head = crsr.column_names
      print("{:<15}{:<25}{:<10}{:<5}{:<10}{:<12}{:<25}{:<5}".format(str(head[0]), str(head[1]), str(head[2]),
str(head[3]), str(head[4]), str(head[5]), str(head[6]), str(head[7])))
      for i in crsr:
         #[101256, 'amul milk', 25, Decimal('22.00'), 'A1', datetime.date(2022, 2, 6), 'amul.co']
         itm = "{:<15}{:<25}{:<10}{:<5}{:<10}{:<12}{:<25}{:<5}".format(str(i[0]), str(i[1]), str(i[2]), str(i[3]),
str(i[4]), str(i[5]), str(i[6]), str(i[7]))
         print(itm)


   #main func
```

```
    con=True
    while con:
        print('''1.add
2.modify
3.delete
4.display
5.quit''')

        x=input("Enter your Choice: ")

        if x=='1':
            id_=int(input("Enter the ID of the item: "))
            stock=int(input("Enter the number of stock: "))
            addstock(id_,stock)

        elif x=='2':
            id_=int(input("Enter the ID of the item: "))
            name=input("Enter the Name of the product: ")
            stock=int(input("Enter the number of stocks: "))
            MRP=float(input("Enter the Price of the Product: "))
            BatchNo=input("Enther the batch number: ")
            exp_date=input("Enter the expiry date of the product: ")
            vendor=input("Enter the vendor: ")
            modifystock(id_,name,stock,MRP, exp_date,vendor,BatchNo)

        elif x=='3':
            id_=int(input("Enter the ID of the item: "))
            deletestock(id_)

        elif x=="4":
            displaystock()
        elif x=="5":
            quit()
        else:
            print("Invalid Choice")

root = Tk()
root.config(bg="black")
root.geometry("1366x768")
root.title("Grocery Store Management")
root.iconbitmap("logo.ico")
logo = PhotoImage(file="logo.gif")
logolabel = Label(root, image=logo, relief="sunken")
logolabel.grid(row=1, column=6, rowspan=4)

## [ BUTTON COMMANDS ]
```

```
def cadd(e=None):
    id_ = identry.get()
    qt_ = qtentry.get()
    identry.delete(0, END)
    qtentry.delete(0, END)
    identry.focus_set()
    ## [ EMPTY DATA ]
    if (id_ == "") or (qt_=="""):
        root = Tk()
        mess = messagebox.showerror('Insufficient Data!', '[ERROR]: The required columns are
empty\nsorry for the inconvineance')
        if mess == messagebox.OK:
            root.destroy()

    elif not ((len(id_) == 6) and id_.isdigit() and qt_.isdigit() and (int(qt_)>=0)):
        root = Tk()
        mess = messagebox.showerror('Value Error!', '[ERROR]: The given value is not in the right
format\nsorry for the inconvineance')
        if mess == messagebox.OK:
            root.destroy()

    else:
    ## [ PULLING DATA FROM SQL ]
        dbm = sql.Connect(user="root", host="localhost", passwd="qweasd", database="GSM")
        crsr = dbm.cursor()
        crsr.execute(f"select * from stock where id = {id_};")
        for i in crsr:
            i=list(i)

        #[101256, 'amul milk', 25, Decimal('22.00'), 'A1', datetime.date(2022, 2, 6), 'amul.co']
    ## [ EXCEPTIONAL HANDLING AND ADDING TO CART ]
        def updatecsv(id_, qt_, item):
            try:
                in_yes = False
                buffer = []
                f = open("cart.csv", 'r')
                while True:
                    y = next(csv.reader(f))
                    if y[0] == id_:
                        in_yes=True
                        y[2] = str(int(y[2]) + int(qt_))
                        y[4] = str(float(y[3])*int(y[2]) + round((float(y[3])*float(y[5])*int(y[2]))/100, 1)).strip('0')
                        buffer.append(y)
                    else:
                        buffer.append(y)
            except StopIteration:
                f.close()
```

```
            f = open("cart.csv", 'w', newline="")
            y = csv.writer(f)
            if in_yes == False:
                buffer.append(item)
            y.writerows(buffer)
            f.close()
        try:
            if datetime.date.today() >= i[5]:
                root = Tk()
                mess = messagebox.showerror('Item expired!', '[ERROR]: The entered item is expired\nsorry
for the inconvineance')
                if mess == messagebox.OK:
                    root.destroy()
            elif int(qt_) > int(i[2]):
                root = Tk()
                mess = messagebox.showerror('Not enough stock!', f'[ERROR]: The required quantity is not
available in stock\n\nOnly {i[2]} item is left in stock\n\nsorry for the inconvineance')
                if mess == messagebox.OK:
                    root.destroy()
            else:
                item = [i[0], i[1], int(qt_), float(i[3]), float(i[3])*int(qt_) +
round((float(i[3])*float(i[7])*int(qt_))/100, 1), i[7]]
                updatecsv(id_, qt_, item)
                crsr.execute(f"update stock set stock.stock = stock - {qt_} where ID = {id_}")
                dbm.commit()
                display()
        except NameError:
            root = Tk()
            mess = messagebox.showerror('Item not found!', '[ERROR]: The entered item is not
found\nplease check the ID of the item again\nsorry for the inconvineance')
            if mess == messagebox.OK:
                root.destroy()


def cclear():
    global item_list
    f = open("cart.csv", "r")
    y = csv.reader(f)
    dbm = sql.Connect(user="root", host="localhost", passwd="qweasd", database="GSM")
    crsr = dbm.cursor()
    try:
        while True:
            item = next(y)
            crsr.execute(f"update stock set stock = stock + {int(item[2])} where id =
{int(item[0])}")#int(item[2])
            dbm.commit()
    except StopIteration:
```

```
        f.close()
        f = open("cart.csv", "w", newline="")
        f.flush()
        f.close()
        item_list = ['                                                                    ', '
', '                                                                    ', '                                                                    ', '
', '                                                                    ', '                                                                    ', '
', '                                                                    ', '                                                                    ', '
', '                                                                    ', '                                                                    ', '
']
        display()

def cpay():
    global usr
    global cntr
    global item_list
    global tcost
    y = open("adminlog.bin", 'rb')
    x = decrypt(pickle.load(y)).split("\n@*@\n")
    f = open("bills/bill.txt", "w")
    #"{:<7}{:<15}{:<25}{:<10}{:<10}{:<10}".format(l[0], l[1], l[2], l[3], l[4], l[5])
    g = open("cart.csv", "r")
    out = x[2].replace("dt\ntm", (str(datetime.datetime.now())[:10:]+"\n"+str(datetime.datetime.now())
[11:19:]))
    k = csv.reader(g)
    slno=1
    try:
        while True:
            item = next(k)
            out += "\n" + "{:<9}{:<15}{:<25}{:<16}{:<10}{:<10}".format(slno, item[0], item[1], item[2], item[3],
item[4])
            slno+=1
    except StopIteration:
        g.close()
        out += "\n\n\n"+"Total Cost :"+ tcost+"\n\n\n"+"\n"+ x[-1].replace("     cashier:cshr\n
counter:cntr", "     cashier:"+usr+"\n     counter:"+cntr)
        f.write(out)
        y.close()
        f.close()
        g = open("cart.csv", "w")
        g.close()
        item_list = ['                                                                    ', '
', '                                                                    ', '                                                                    ', '
', '                                                                    ', '                                                                    ', '
', '                                                                    ', '                                                                    ', '
', '                                                                    ', '                                                                    ', '
']
        tcost = "$0"
        upddisplay()
```

```
def cmod():
    global tcost
    global item_list
    root = Tk()
    mes = simpledialog.askstring("modify item", "Enter the item ID :")
    stk = simpledialog.askstring("modify item", "Enter the new stock :")
    if mes != "":
        if stk == "0":
            f = open("cart.csv", 'r')
            y = csv.reader(f)
            cart = []
            try:
                while True:
                    item = next(y)
                    if mes == item[0]:
                        # putting items back in shelf
                        dbm = sql.Connect(user="root", host="localhost", passwd="qweasd", database="GSM")
                        crsr = dbm.cursor()
                        crsr.execute(f"update stock set stock = stock + {int(item[2])} where id = {int(item[0])}")
                        dbm.commit()
                    else:
                        cart.append(item)
            except StopIteration:
                f.close()
                f = open("cart.csv", "w", newline="")
                z = csv.writer(f)
                z.writerows(cart)
                f.close()
                root.destroy()
                item_list = ['                                                                    ', '
', '                                                                    ', '                                                                    ', '
', '                                                                    ', '                                                                    ', '
', '                                                                    ', '                                                                    ', '
']
                tcost = "$0"
                display()
                upddisplay()
        elif stk.isdigit():
            f = open("cart.csv", 'r')
            y = csv.reader(f)
            cart = []
            try:
                while True:
                    item = next(y)
                    if mes == item[0]:
```

```
                # putting items back in shelf
                dbm = sql.Connect(user="root", host="localhost", passwd="qweasd", database="GSM")
                crsr = dbm.cursor()
                crsr.execute(f"update stock set stock = stock + {int(item[2])} where id = {int(item[0])}")
                dbm.commit()
            else:
                cart.append(item)
        except StopIteration:
            f.close()
            f = open("cart.csv", "w", newline="")
            z = csv.writer(f)
            z.writerows(cart)
            f.close()
            root.destroy()
            item_list = ['                                                                    ', '
', '                                                                    ', '                                                                    ', '
' '                                                                    ' '                                                                    ' '
', '                                                                    ', '                                                                    ', '
', '                                                                    ', '                                                                    ', '
']
            tcost = "$0"
            display()
            id_ = mes
            qt_=stk
            if (id_ == "") or (qt_==""):
                root = Tk()
                mess = messagebox.showerror('Insufficient Data!', '[ERROR]: The required columns are
empty\nsorry for the inconvineance')
                if mess == messagebox.OK:
                    root.destroy()
            elif not ((len(id_) == 6) and id_.isdigit() and qt_.isdigit() and (int(qt_)>=0)):
                root = Tk()
                mess = messagebox.showerror('Value Error!', '[ERROR]: The given value is not in the right
format\nsorry for the inconvineance')
                if mess == messagebox.OK:
                    root.destroy()
            else:
                dbm = sql.Connect(user="root", host="localhost", passwd="qweasd", database="GSM")
                crsr = dbm.cursor()
                crsr.execute(f"select * from stock where id = {id_};")
                for i in crsr:
                    i=list(i)
                def updatecsv(id_, qt_, item):
                    try:
                        in_yes = False
                        buffer = []
                        f = open("cart.csv", 'r')
                        while True:
```

```
                    y = next(csv.reader(f))
                    if y[0] == id_:
                        in_yes=True
                        y[2] = str(int(y[2]) + int(qt_))
                        y[4] = str(float(y[3])*int(y[2]))
                        buffer.append(y)
                    else:
                        buffer.append(y)
                except StopIteration:
                    f.close()
                    f = open("cart.csv", 'w', newline="")
                    y = csv.writer(f)
                    if in_yes == False:
                        buffer.append(item)
                    y.writerows(buffer)
                    f.close()
            try:
                if datetime.date.today() >= i[5]:
                    root = Tk()
                    mess = messagebox.showerror('Item expired!', '[ERROR]: The entered item is
expired\nsorry for the inconvineance')
                    if mess == messagebox.OK:
                        root.destroy()
                elif int(qt_) > int(i[2]):
                    root = Tk()
                    mess = messagebox.showerror('Not enough stock!', f'[ERROR]: The required quantity is
not available in stock\n\nOnly {i[2]} item is left in stock\n\nsorry for the inconvineance')
                    if mess == messagebox.OK:
                        root.destroy()
                else:
                    item = [i[0], i[1], int(qt_), float(i[3]), float(i[3])*int(qt_)]
                    updatecsv(id_, qt_, item)
                    crsr.execute(f"update stock set stock.stock = stock - {qt_} where ID = {id_}")
                    dbm.commit()
                    display()
                    upddisplay()
            except NameError:
                root = Tk()
                mess = messagebox.showerror('Item not found!', '[ERROR]: The entered item is not
found\nplease check the ID of the item again\nsorry for the inconvineance')
                if mess == messagebox.OK:
                    root.destroy()
            except:
                pass
        else:
            root = Tk()
```

```
        mess = messagebox.showerror('Insufficient Data!', '[ERROR]: The required columns are
empty\nsorry for the inconvineance')
        if mess == messagebox.OK:
            root.destroy()
    else:
        root = Tk()
        mess = messagebox.showerror('Insufficient Data!', '[ERROR]: The required columns are
empty\nsorry for the inconvineance')
        if mess == messagebox.OK:
            root.destroy()

def cdel():
    global item_list
    global tcost
    root = Tk()
    mes = simpledialog.askstring("delete item", "Enter the item ID :")
    if mes == '':
        f = open("cart.csv", 'r')
        y = csv.reader(f)
        cart = []
        try:
            while True:
                cart.append(next(y))
        except StopIteration:
            if cart==[]:
                f.close()
            else:
                f.close()
                f = open("cart.csv", 'w', newline="")
                # putting items back in shelf
                dbm = sql.Connect(user="root", host="localhost", passwd="qweasd", database="GSM")
                crsr = dbm.cursor()
                crsr.execute(f"update stock set stock = stock + {int(cart[-1][2])} where id = {int(cart[-1][0])}")
                dbm.commit()

                cart.remove(cart[-1])
                x = csv.writer(f)
                x.writerows(cart)
                f.close()
                root.destroy()
                item_list = ['                                                    ', '
', '                                                    ', '                                                    ', '
', '                                                    ', '                                                    ', '
', '                                                    ', '                                                    ', '
']
                tcost = "$0"
                display()
```

```
    else:
        f = open("cart.csv", 'r')
        y = csv.reader(f)
        cart = []
        try:
            while True:
                item = next(y)
                if mes == item[0]:
                    # putting items back in shelf
                    dbm = sql.Connect(user="root", host="localhost", passwd="qweasd", database="GSM")
                    crsr = dbm.cursor()
                    crsr.execute(f"update stock set stock = stock + {int(item[2])} where id = {int(item[0])}")
                    dbm.commit()
                else:
                    cart.append(item)
        except StopIteration:
            f.close()
            f = open("cart.csv", "w", newline="")
            z = csv.writer(f)
            z.writerows(cart)
            f.close()
            root.destroy()
            item_list = ['                                                                                ', '
', '                                                                                ', '                                                                                ', '
', '                                                                                ', '                                                                                ', '
', '                                                                                ', '                                                                                ', '
']
            tcost = "$0"
            display()




def _quit():
    global root
    cclear()
    root.destroy()

def decrypt(x):
    cipher = ['3', '1', '4', '1', '5', '9', '2', '6', '5', '3', '5', '8', '9', '7', '9', '3', '2', '3', '8', '4', '6', '2', '6', '4', '3', '3',
'8', '3', '2', '7', '9', '5', '0', '2', '8', '8', '4', '1', '9', '7', '1', '6', '9', '3', '9', '9', '3', '7', '5', '1', '0', '5', '8', '2', '0',
'9', '7', '4', '9', '4', '4', '5', '9', '2', '3', '0', '7', '8', '1', '6', '4', '0', '6', '2', '8', '6', '2', '0', '8', '9', '8', '6', '2', '8',
'0', '3', '4', '8', '2', '5', '3', '4', '2', '1', '1', '7', '0', '6', '7', '9']
    if len(x)> 100:
        chunks = [x[i:i+100] for i in range(0, len(x), 100)]
        decrypt = ""
```

```
        for i in chunks:
            i = list(i)
            for j in range(len(i)):
                decrypt += chr(ord(i[j]) - int(cipher[j]))
    else:
        decrypt = ""
        x = list(x)
        for i in range(len(x)):
            decrypt += chr(ord(x[i]) - int(cipher[i]))
    return decrypt


def encrypt(x):
    cipher = ['3', '1', '4', '1', '5', '9', '2', '6', '5', '3', '5', '8', '9', '7', '9', '3', '2', '3', '8', '4', '6', '2', '6', '4', '3', '3',
'8', '3', '2', '7', '9', '5', '0', '2', '8', '8', '4', '1', '9', '7', '1', '6', '9', '3', '9', '9', '3', '7', '5', '1', '0', '5', '8', '2', '0',
'9', '7', '4', '9', '4', '4', '5', '9', '2', '3', '0', '7', '8', '1', '6', '4', '0', '6', '2', '8', '6', '2', '0', '8', '9', '8', '6', '2', '8',
'0', '3', '4', '8', '2', '5', '3', '4', '2', '1', '1', '7', '0', '6', '7', '9']
    if len(x)> 100:
        chunks = [x[i:i+100] for i in range(0, len(x), 100)]
        encrypt = ""
        for i in chunks:
            i = list(i)
            for j in range(len(i)):
                encrypt += chr(ord(i[j]) + int(cipher[j]))
    else:
        encrypt = ""
        x = list(x)
        for i in range(len(x)):
            encrypt += chr(ord(x[i]) + int(cipher[i]))
    return encrypt


## [ INFO BOARD ]

dttm= str(datetime.date.today())
usr="S GIRISH"
cntr="1"
icnt="0"

infodate = Label(root, text="Date:\n"+dttm, fg="red", bg="black", font=("Courier", 15), borderwidth=2,
relief="solid", width=12, height=4)
user = Label(root, text="USER:      \n"+usr, fg="red", bg="black", font=("Courier", 15), borderwidth=2,
relief="solid", width=12, height=4)
counter = Label(root, text="COUNTER:\n"+cntr, fg="red", bg="black", font=("Courier", 15),
borderwidth=2, relief="solid", width=12, height=6)
itemcnt = Label(root, text="No. of item:\n\n\n"+icnt, fg="red", bg="black", font=("Courier", 15),
borderwidth=2, relief="solid", width=12, height=11)
```

```
infodate.grid(row=1, column=0, rowspan=2)
user.grid(row=3, column=0, rowspan=2)
counter.grid(row=5, column=0, rowspan=3)
itemcnt.grid(row=8, column=0, rowspan=5)

## [ CREATING BUTTONS ]

delete = Button(root, text="delete previous item", command=cdel, fg="#47ff33", bg="black",
font=("Courier", 15), width=30, height=4)
add = Button(root, text="add item", command=cadd, fg="#47ff33", bg="black", font=("Courier", 15),
width=25, height=4)
clear = Button(root, text="clear cart", command=cclear, fg="#47ff33", bg="black", font=("Courier", 15),
width=25, height=4)
quit_ = Button(root, text="Quit", command=_quit, fg="#47ff33", bg="black", font=("Courier", 15),
padx=55, pady=33)
modify = Button(root, text="modify item", command=cmod, fg="#47ff33", bg="black", font=("Courier",
15), padx=26, pady=33)
pay = Button(root, text="payment", command=cpay, fg="red", bg="black", font=("Courier", 15),
padx=45, pady=100)

## [ GRID ]


delete.grid(row=0, column=0, columnspan=3)
add.grid(row=0, column=3)
clear.grid(row=0, column=4)
quit_.grid(row=0, column=5)
modify.grid(row=0, column=6)
pay.grid(row=8, column=6, rowspan=5)

## [ CREATE LABELS ]

item_list = ['                                                                    ', '
', '                                                    ', '                                                    ', '
', '                                                    ', '                                                    ', '
', '                                                    ', '                                                    ', '
']
head = "slno     id          name                quantity      MRP      cost    "
tcost = "$0"

header = Label(root, text=head, fg="yellow", bg="black", font=("Courier", 15), borderwidth=2,
relief="sunken", width=83, height=2)
i0 = Label(root, text=item_list[0], fg="white", bg="#1400a8", font=("Courier", 15), borderwidth=2,
relief="sunken", width=83, height=2)
```

```
i1 = Label(root, text=item_list[1], fg="white", bg="#1400a8", font=("Courier", 15), borderwidth=2,
relief="sunken", width=83, height=2)
i2 = Label(root, text=item_list[2], fg="white", bg="#1400a8", font=("Courier", 15), borderwidth=2,
relief="sunken", width=83, height=2)
i3 = Label(root, text=item_list[3], fg="white", bg="#1400a8", font=("Courier", 15), borderwidth=2,
relief="sunken", width=83, height=2)
i4 = Label(root, text=item_list[4], fg="white", bg="#1400a8", font=("Courier", 15), borderwidth=2,
relief="sunken", width=83, height=2)
i5 = Label(root, text=item_list[5], fg="white", bg="#1400a8", font=("Courier", 15), borderwidth=2,
relief="sunken", width=83, height=2)
i6 = Label(root, text=item_list[6], fg="white", bg="#1400a8", font=("Courier", 15), borderwidth=2,
relief="sunken", width=83, height=2)
i7 = Label(root, text=item_list[7], fg="white", bg="#1400a8", font=("Courier", 15), borderwidth=2,
relief="sunken", width=83, height=2)
i8 = Label(root, text=item_list[8], fg="white", bg="#1400a8", font=("Courier", 15), borderwidth=2,
relief="sunken", width=83, height=2)
i9 = Label(root, text=item_list[9], fg="white", bg="#1400a8", font=("Courier", 15), borderwidth=2,
relief="sunken", width=83, height=2)
cost = Label(root, text=tcost, fg="red", bg="#404040", font=("Courier", 27), borderwidth=2,
relief="sunken", width=9, pady=20)

## [ GRID LABEL ]

header.grid(row=1, column=1, columnspan=5)
i0.grid(row=2, column=1, columnspan=5)
i1.grid(row=3, column=1, columnspan=5)
i2.grid(row=4, column=1, columnspan=5)
i3.grid(row=5, column=1, columnspan=5)
i4.grid(row=6, column=1, columnspan=5)
i5.grid(row=7, column=1, columnspan=5)
i6.grid(row=8, column=1, columnspan=5)
i7.grid(row=9, column=1, columnspan=5)
i8.grid(row=10, column=1, columnspan=5)
i9.grid(row=11, column=1, columnspan=5)
cost.grid(row=6, column=6, rowspan=2)


def upddisplay():
    global item_list, tcost, itemcnt, icnt, cost, i0, i1, i2, i3, i4, i5, i6, i7, i8, i9
    i0["text"] = item_list[0]
    i1["text"] = item_list[1]
    i2["text"] = item_list[2]
    i3["text"] = item_list[3]
    i4["text"] = item_list[4]
    i5["text"] = item_list[5]
    i6["text"] = item_list[6]
    i7["text"] = item_list[7]
```

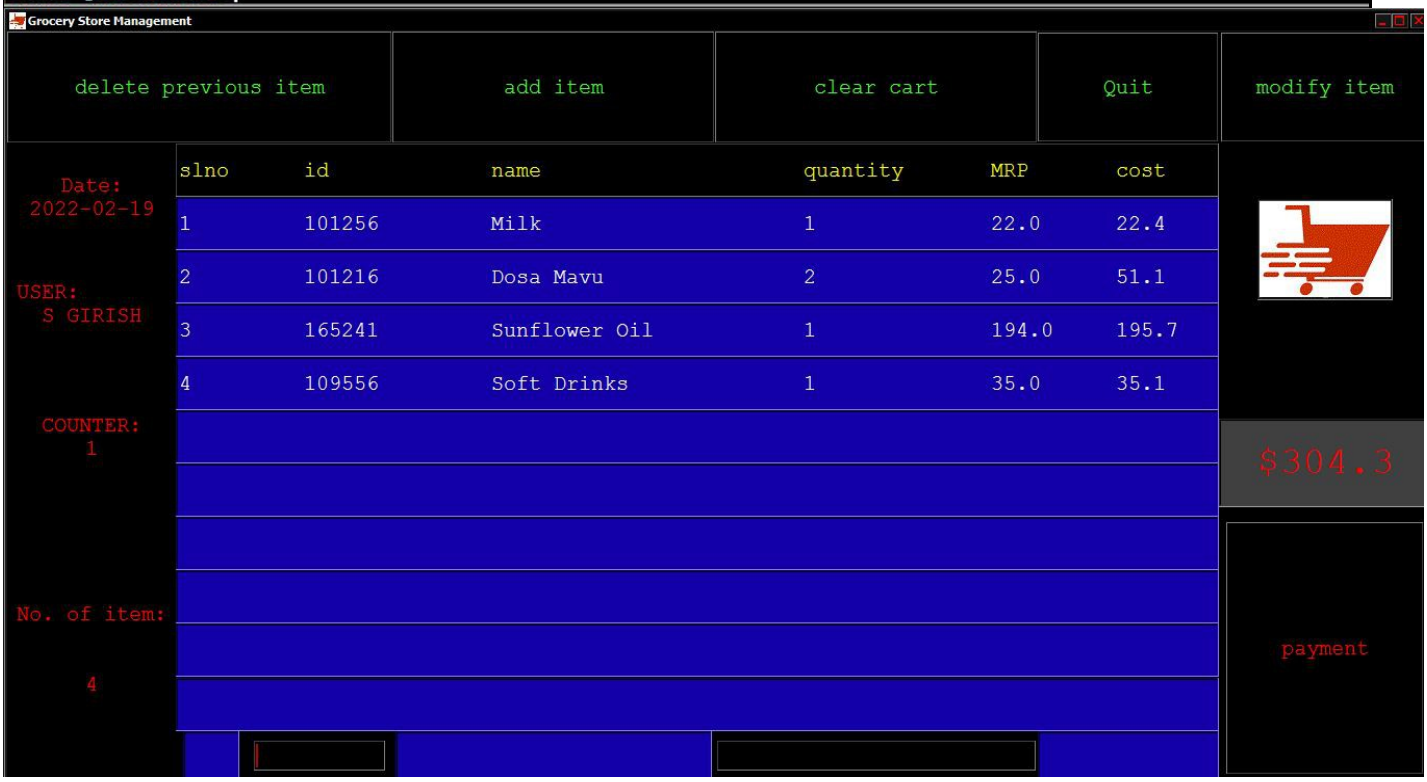
```
    i8["text"] = item_list[8]
    i9["text"] = item_list[9]
    cost["text"] = tcost
    itemcnt["text"] = "No. of item:\n\n\n"+icnt


upddisplay()

## [ DISPLAY ]

def display():
    # item = ['101256', 'amul milk', '2', '22.0', '44.0']
    global item_list
    global tcost
    global icnt
    global line
    buffercost =  0
    f = open("cart.csv", "r")
    item_buffer = []
    k = csv.reader(f)
    try:
        while True:
            item_buffer.append(next(k))
    except StopIteration:
        try:
            f.close()
            icnt = str(len(item_buffer))
            if len(item_buffer)<=10:
                for i in range(len(item_buffer)):
                    item = item_buffer[i]
                    buffercost += float(item[4])
                    item_list[i] = (str(i+1) + (10 - len(str(i+1)))*" " + item[0] + (15-len(item[0]))*" " + item[1] + (25-
len(item[1]))*" " + item[2] + (15-len(item[2]))*" " + item[3] + (10-len(item[3]))*" " + item[4] + (8-
len(item[4]))*" ")
                tcost = "$" + str(buffercost)
                upddisplay()
            else:
                for i in range(len(item_buffer)):
                    if i<10:
                        item = item_buffer[-10+i-line]
                        buffercost += float(item[4])
                        item_list[i] = (str(item_buffer.index(item)+1) + (10 - len(str(item_buffer.index(item)+1)))*"
" + item[0] + (15-len(item[0]))*" " + item[1] + (25-len(item[1]))*" " + item[2] + (15-len(item[2]))*" " +
item[3] + (10-len(item[3]))*" " + item[4] + (8-len(item[4]))*" ")
                    else:
                        item = item_buffer[-10+i-line]
                        buffercost += float(item[4])
```

```
            tcost = "$" + str(buffercost)
            upddisplay()
        except IndexError:
            line -=1


## [ ROW 12 ]

qtentry = Entry(root, fg="red", bg="black", font=("Courier", 15), borderwidth=2, relief="sunken",
width=25)
identry = Entry(root, fg="red", bg="black", font=("Courier", 15), borderwidth=2, relief="sunken",
width=10)

def fcs(e):
    qtentry.focus_set()

identry.grid(row=12, column=2)
qtentry.grid(row=12, column=4)


identry.bind("<Return>", fcs)
qtentry.bind("<Return>", cadd)

psl = ""
pname = ""
pprice = ""


sl = Label(root, text=psl, fg="white", bg="#1400a8", font=("Courier", 15), borderwidth=2,
relief="sunken", padx=25, pady=10)
name = Label(root, text=pname, fg="white", bg="#1400a8", font=("Courier", 15), borderwidth=2,
relief="sunken", padx=150, pady=10)
pric = Label(root, text=pprice, fg="white", bg="#1400a8", font=("Courier", 15), borderwidth=2,
relief="sunken", padx=85, pady=10)


sl.grid(row=12, column=1)
name.grid(row=12, column=3)
pric.grid(row=12, column=5)


line=0
def decrement(e):
    global line
    if line ==0:
        pass
    else:
```

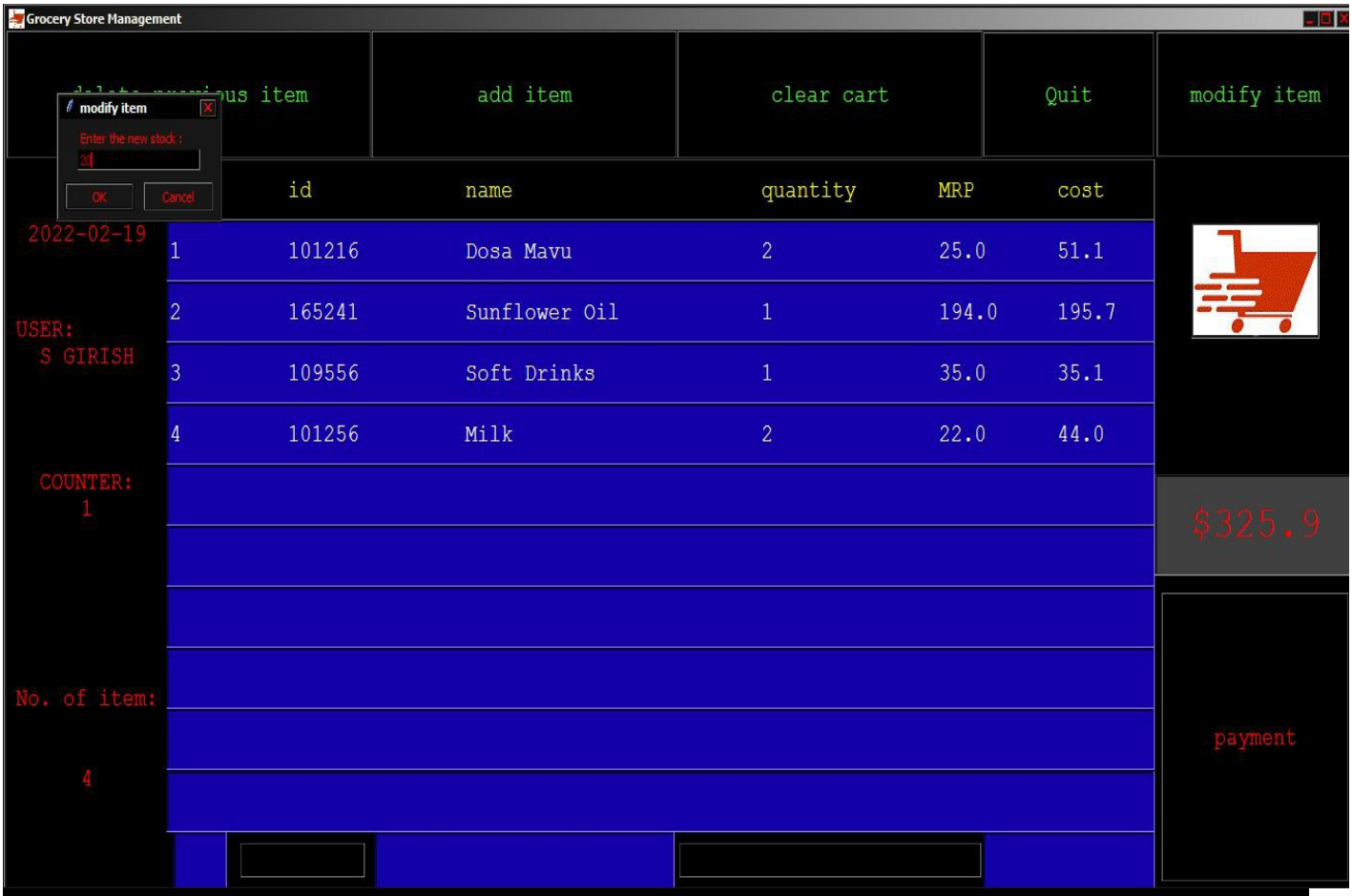
```
        line-=1
        item_list = ['                                                                                  ', '
', '                                                                                  ', '                                                                                  ', '
', '                                                                                  ', '                                                                                  ', '
', '                                                                                  ', '                                                                                  ', '
']
        tcost = "$0"
        display()
def increment(e):
    global line
    f = open("cart.csv", 'r')
    cart=[]
    reader = csv.reader(f)
    try:
        while True:
            cart.append(next(reader))
    except StopIteration:
        f.close()
        pass
    line+=1
    item_list = ['                                                                                  ', '
', '                                                                                  ', '                                                                                  ', '
', '                                                                                  ', '                                                                                  ', '
', '                                                                                  ', '                                                                                  ', '
']
    tcost = "$0"
    display()
root.bind("<w>", increment)
root.bind("<s>", decrement)

identry.focus_set()
root.mainloop()
```

OUTPUT


```
1.add
2.modify
3.delete
4.display
5.quit
Enter your Choice: 4
ID            Name                Stock    MRP  BATCH_NO  EXPDATE     VENDOR                TAX
101216        Dosa Mavu           16       25   1         2022-03-21  Heritage              2.11
101256        Milk                15       22   1         2022-02-28  Amul                  1.74
101425        Dishwash            17       60   1         2023-04-20  Pril                  1.60
109556        Soft Drinks         17       35   1         2022-07-15  Coca-Cola             0.21
110156        Namkeen             17       45   1         2022-12-22  Haldiram              1.01
121454        Chocolates          13       20   1         2022-06-07  Cadbury               0.81
130957        Toothpaste          17       80   1         2023-04-30  Colgate               0.45
132156        Sweets              17       75   1         2022-08-17  Bikanervala           1.11
145236        Wheat Flour         20       54   1         2022-09-17  Aashirvad             0.77
152326        Chips               20       20   1         2022-03-17  Pepsico               0.71
152452        Soap                17       77   1         2023-07-06  Pears                 1.21
165241        Sunflower Oil       17       194  1         2022-08-22  Gold Winner           0.87
187456        Ketchup             20       42   1         2023-02-04  Nestle                0.78
189756        Pulses              18       27   1         2022-05-05  Tata                  0.54
1.add
2.modify
3.delete
4.display
5.quit
Enter your Choice: 1
Enter the ID of the item: 101256
Enter the number of stock: 10
1.add
2.modify
3.delete
4.display
5.quit
Enter your Choice:
```




Grocery Store Management
delete previous item
add item
clear cart
Quit
modify item
Date:
2022-02-19
USER:
S GIRISH
COUNTER:
1
No. of item:
4
slno
id
name
quantity
MRP
cost
1
Milk
1
22.0
22.4
2
Dosa Mavu
2
25.0
51.1
3
165241
Sunflower Oil
1
194.0
195.7
4
109556
Soft Drinks
1
35.0
35.1
delete item
Enter the item ID :
101250
OK
Cancel
$304.3
payment


Grocery Store Management
modify item
Enter the new stock :
20
OK
Cancel
add item
clear cart
Quit
modify item
id
name
quantity
MRP
cost
2022-02-19
1 101216 Dosa Mavu 2 25.0 51.1
2 165241 Sunflower Oil 1 194.0 195.7
3 109556 Soft Drinks 1 35.0 35.1
4 101256 Milk 2 22.0 44.0
USER:
S GIRISH
COUNTER:
1
$325.9
No. of item:
4
payment

bill - Notepad

File Edit Format View Help

Welcome to Store_name
address
phone-xxxxxxxxxx
GSTIN-6ad46516153dfs654s65f4d
Bill of supply
Bill No: 000
2022-02-19
14:22:10

Slno	ID	Name	Quantity	MRP	Price
1	101216	Dosa Mavu	2	25.0	51.1
2	101256	Milk	1	22.0	22.4
3	101425	Dishwash	1	60.0	61.0
4	109556	Soft Drinks	1	35.0	35.1
5	110156	Namkeen	1	45.0	45.5
6	121454	Chocolates	4	20.0	80.6
7	130957	Toothpaste	1	80.0	80.4
8	132156	Sweets	1	75.0	75.8
9	152452	Soap	1	77.0	77.9
10	165241	Sunflower Oil	1	194.0	195.7
11	189756	Pulses	2	27.0	54.3

Total Cost :$779.80

Thank you for shopping at Store_name
for any queries call at xxxxxxxxxx
cashier:S GIRISH
counter:1
shop again between 7AM and 9AM

SQL Database

```
mysql> select * from stock;
+--------+---------------+-------+-----+----------+------------+-------------+------+
| ID     | Name          | Stock | MRP | BATCH_NO | EXPDATE    | VENDOR      | TAX  |
+--------+---------------+-------+-----+----------+------------+-------------+------+
| 101216 | Dosa Mavu     |    16 |  25 |        1 | 2022-03-21 | Heritage    | 2.11 |
| 101256 | Milk          |    15 |  22 |        1 | 2022-02-28 | Amul        | 1.74 |
| 101425 | Dishwash      |    17 |  60 |        1 | 2023-04-20 | Pril        | 1.60 |
| 109556 | Soft Drinks   |    17 |  35 |        1 | 2022-07-15 | Coca-Cola   | 0.21 |
| 110156 | Namkeen       |    17 |  45 |        1 | 2022-12-22 | Haldiram    | 1.01 |
| 121454 | Chocolates    |    13 |  20 |        1 | 2022-06-07 | Cadbury     | 0.81 |
| 130957 | Toothpaste    |    17 |  80 |        1 | 2023-04-30 | Colgate     | 0.45 |
| 132156 | Sweets        |    17 |  75 |        1 | 2022-08-17 | Bikanervala | 1.11 |
| 145236 | Wheat Flour   |    20 |  54 |        1 | 2022-09-17 | Aashirvad   | 0.77 |
| 152326 | Chips         |    20 |  20 |        1 | 2022-03-17 | Pepsico     | 0.71 |
| 152452 | Soap          |    17 |  77 |        1 | 2023-07-06 | Pears       | 1.21 |
| 165241 | Sunflower Oil |    17 | 194 |        1 | 2022-08-22 | Gold Winner | 0.87 |
| 187456 | Ketchup       |    20 |  42 |        1 | 2023-02-04 | Nestle      | 0.78 |
| 189756 | Pulses        |    18 |  27 |        1 | 2022-05-05 | Tata        | 0.54 |
+--------+---------------+-------+-----+----------+------------+-------------+------+
14 rows in set (0.00 sec)

mysql> desc stock;
+----------+--------------+------+-----+---------+-------+
| Field    | Type         | Null | Key | Default | Extra |
+----------+--------------+------+-----+---------+-------+
| ID       | int(11)      | NO   | PRI | NULL    |       |
| Name     | varchar(20)  | NO   |     | NULL    |       |
| Stock    | int(11)      | NO   |     | NULL    |       |
| MRP      | int(11)      | NO   |     | NULL    |       |
| BATCH_NO | int(11)      | NO   |     | NULL    |       |
| EXPDATE  | date         | NO   |     | NULL    |       |
| VENDOR   | varchar(20)  | NO   |     | NULL    |       |
| TAX      | decimal(4,2) | YES  |     | NULL    |       |
+----------+--------------+------+-----+---------+-------+
8 rows in set (0.01 sec)

mysql>
```