CASE STUDY: FOOD BOX

TEAM SYNTAX ERROR

A report on the team project ‘FOOD BOX’ by team Syntax Error, consisting of a brief summary, input code snippets, output screenshots, liabilities and future developmental plans.

MODERN SENIOR SECONDARY SCHOOL

CLASS 12-A

2017-18

***CONTENTS***

[**ACKNOWLEDGEMENTS**](#_Toc498890864) 2

[**SYSTEM REQUIREMENTS**](#_Toc498890865) 3

[**INTRODUCTION**](#_Toc498890866) 4

**PROGRAM STRUCTURE\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**­5

**SOURCE CODE\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** 5

**OUTPUT SNIPPETS\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** 5

**EXTERNAL MODULES\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**7

**FUTURE DEVELOPMENT\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** 8

**BIBLIOGRAPHY\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**9

**THE Team Members of the Syntax Error Team are:**

1. **V.Vignesh - XII A**
2. **Nithin Narayanan.G - XII A**
3. **Aakash Murugan - XII A**

# **ACKNOWLEDGEMENTS**

Completing this project would not have been possible without the help and able guidance of our teacher, Mrs.L.Padmavathy , who supported us throughout the course of our project .

By providing us access to the laboratory and encouraging us to excel, she has supported us in every step of the way. We are very much obliged to her.

Personally, at this juncture, I extend my gratitude to my team members :

Aakash Murugan and

Nithin Narayanan.G

Also, we are grateful for getting this opportunity to work on such a product, and to “TCS” and our school for providing us such a wonderful opportunity.

# **SYSTEM REQUIREMENTS**

***\*Required modules are provided in the project file with instructions for installation***

# **INTRODUCTION**

The aim of the project,**“FOOD BOX”**, which is an application for ordering food and managing the stocks of food. The application consists of two different modules:

1. **INVENTORY**
2. **SALES**

There are two different access LEVELS:

1. **ADMINISTRATOR**
2. **USER**

The ***Inventory*** module is accessible only through the ***Administrator*** level; The ***Sales*** module through *both****Administrator*** and ***User*** levels.

The ***Inventory*** module consists of the ***available food items and relevant details*** such as quantity available, price, food type and food code. The Administrator will be able to create, modify and delete items from the list of food items available.

The ***Sales*** module offers the ***option to order food*** from the list of available food items, depending upon the quantity in Inventory. The User will be able to see the price of each food item available and order the food he wishes to. ***Relevant changes*** will be made in the ***Inventory*** after a purchase by the User has been completed.

At the ***end of the day***, a comprehensive ***report*** will be generated, accessible only by the ***Administrator***, which provides details of ***total sales*** and current ***state of the Inventory***.

**PROGRAM STRUCTURE**

The program consists of a core database, an Excel worksheet, ‘***master\_sheet.xlsx***’, which contains all data with total of 5 individual worksheets within. Modification of these worksheets, *i.e*. to create, modify or delete food items is possible ***only through Inventory*** module, which is accessible only through Administrator level.

The ***Sales*** module ***procures data*** from the main Inventory worksheet and ***displays*** them for the User to order, and based on that order, the worksheets shall be ***aptly modified*** by the program to reflect these changes.

All computations and calculations are carried out in MS Excel and the results are fetched from those worksheets.

The ***User/Administrator access level differentiation*** is created by using 2 different worksheets for storing data on various parameters and accessing them while necessary. The two basic modules Sales and Inventory are created using two ***classes*** and the main program is an interlinked ***instance*** of these two classes.

Each class further contains various functions required to perform the given tasks and are invoked ***within the class.***

The main opening page is a ***Tk( ) window*** with further links to Inventory and Sales module, which are instances of the class.

The ***GUI*** part of the program is achieved through an inbuilt module in ***Python*** known as ***Tkinter.***

The ***access*** and ***modification*** of ***Excel worksheets*** is done through another ***external module*** called as ***openpyxl*** for which the installation requires a pip module.

**SOURCE CODE**

#Case Study-Food box

#Done By

#Syntax error team

from Tkinter import \*

from openpyxl import \*

import os

import tkMessageBox

class Inventory():

def \_\_init\_\_(self):

self.number=0

self.value1=0 self.color=['blue','red','orange','green','navy','firebrick','cyan','turquoise']

def inventory(self):

inv =Tk()

inv.title('INVENTORY')

inv.configure(bg='skyblue1')

wb=load\_workbook('master\_sheet.xlsx')

ws=wb['Inventory']; row=ws.max\_row

ws1=wb['Sales']

ws2=wb['Report']

fori in range(1,row+1):

s1='a'+str(i);a1=str(ws[s1].value)

s2='b'+str(i);a2=str(ws[s2].value)

s3='c'+str(i);a3=str(ws[s3].value)

s4='d'+str(i);a4=str(ws[s4].value)

s5='e'+str(i);a5=str(ws[s5].value)

if a1!='None' and a2!='None' and a3!='None' and a4!='None' and a5!='None':

Label(inv,text=a1,bg='skyblue1',fg='black',font='bold').grid(row=i,column=0,sticky=W)

Label(inv,text=a2,bg='skyblue1',fg='black',font='bold').grid(row=i,column=1,sticky=W)

Label(inv,text=a3,bg='skyblue1',fg='black',font='bold').grid(row=i,column=2,sticky=W)

Label(inv,text=a4,bg='skyblue1',fg='black',font='bold').grid(row=i,column=3,sticky=W)

Label(inv,text=a5,bg='skyblue1',fg='black',font='bold').grid(row=i,column=4,sticky=W)

num=i

Button(inv,text='ADD FOOD',bg='yellow',fg='black',command=lambda: self.add\_new(wb,ws,ws1,ws2,inv)).grid(row=num+1,column=0)

Button(inv,text='MODIFY FOOD',bg='yellow',fg='black',command=lambda: self.modify\_food(wb,ws,ws1,ws2,inv)).grid(row=num+1,column=1)

Button(inv,text='DELETE FOOD',bg='yellow',fg='black',command=lambda: self.delete(wb,ws,ws1,ws2,inv)).grid(row=num+1,column=2)

Button(inv,text='SEARCH BY NAME',bg='yellow',fg='black',command=lambda: self.search\_by\_name(wb,ws,ws1,ws2)).grid(row=num+1,column=3)

Button(inv,text='SEARCH BY FOOD NUMBER',bg='yellow',fg='black',command=lambda: self.search\_by\_fno(wb,ws,ws1,ws2)).grid(row=num+1,column=4)

inv.mainloop()

returninv

defadd\_new(self,wb,ws,ws1,ws2,inv):

def b1():

flag=True

maxrow=ws.max\_row

foodtype=e1.get()

foodno=e2.get()

foodname=e3.get()

qty=e4.get()

price=e5.get()

iffoodtype=='' or foodno=='' or foodname=='' or qty=='' or price=='' :

tkMessageBox.showerror('Error','All fields are mandatory!')

else:

fori in range(2,maxrow+1):

ifstr(foodno) == str(ws['b'+str(i)].value):

tkMessageBox.showerror('Error','Food Number already Exists. \n Choose a new one')

flag=False

if flag:

iffoodtype.upper()!="CHINESE" or foodtype.upper()!="CONTINENTAL" or foodtype.upper()!="SOUTH" or foodtype.upper()=="NORTH":

ifqty>=0 and price>=0:

ws.append([foodtype,foodno,foodname,qty,price])

ws1.append([foodtype,foodno,foodname,qty,price])

ws2.append([foodtype,foodno,foodname,qty,price])

wb.save("master\_sheet1.xlsx")

os.remove("master\_sheet.xlsx")

os.rename("master\_sheet1.xlsx","master\_sheet.xlsx")

tkMessageBox.showinfo('Success','New Food added')

inv.destroy()

self.inventory()

else:

tkMessageBox.showerror('Error','Enter a valid foodtype')

else:

tkMessageBox.showerror('Error','Values below 0 cannot be accepted')

add=Toplevel()

e1=StringVar()

e2=StringVar()

e3=StringVar()

e4=StringVar()

e5=StringVar()

Label(add,text="Enter the details")

Label(add,text="Food Type").grid(row=0,column=0,sticky=S,pady=3)

Label(add,text="Food Number").grid(row=1,column=0,sticky=S,pady=3)

Label(add,text="Food Name").grid(row=2,column=0,sticky=S,pady=3)

Label(add,text="Quantity").grid(row=3,column=0,sticky=S,pady=3)

Label(add,text="Price").grid(row=4,column=0,sticky=S,pady=3)

Entry(add,textvariable=e1).grid(row=0,column=1,sticky=S,pady=3)

Entry(add,textvariable=e2).grid(row=1,column=1,sticky=S,pady=3)

Entry(add,textvariable=e3).grid(row=2,column=1,sticky=S,pady=3)

Entry(add,textvariable=e4).grid(row=3,column=1,sticky=S,pady=3)

Entry(add,textvariable=e5).grid(row=4,column=1,sticky=S,pady=3)

btn1=Button(add,text="ADD FOOD",command=b1).grid(row=5,column=2,sticky=W)

add.mainloop()

defmodify\_food(self,wb,ws,ws1,ws2,inv):

def modify():

foodtype=e1.get()

foodno=e2.get()

foodname=e3.get()

qty=e4.get()

price=e5.get()

foodno\_existing=e6.get()

foodname\_existing=e7.get()

a=0

maxrow=ws.max\_row

iffoodtype=='' or foodno=='' or foodname=='' or qty=='' or price=='' or foodno\_existing=='' or foodname\_existing=='' :

tkMessageBox.showerror('Error','All fields are mandatory!')

else:

fori in range(2,maxrow+1):

if str(foodname\_existing)==str(ws['c'+str(i)].value) and str(foodno\_existing)==str(ws['b'+str(i)].value):

tkMessageBox.showinfo('Success','Given food item exists')

self.number=str(i)

a=1

break

else:

tkMessageBox.showerror('Error','Food item does not exist. \n Enter a valid food item to modify')

modify\_main.destroy()

if a :

ws['a'+self.number]=foodtype

ws['b'+self.number]=foodno

ws['c'+self.number]=foodname

ws['d'+self.number]=qty

ws['e'+self.number]=price

ws1['a'+self.number]=foodtype

ws1['b'+self.number]=foodno

ws1['c'+self.number]=foodname

ws1['d'+self.number]=qty

ws1['e'+self.number]=price

ws2['a'+self.number]=foodtype

ws2['b'+self.number]=foodno

ws2['c'+self.number]=foodname

ws2['d'+self.number]=qty

ws2['e'+self.number]=price

wb.save("master\_sheet1.xlsx")

os.remove("master\_sheet.xlsx")

os.rename("master\_sheet1.xlsx","master\_sheet.xlsx")

tkMessageBox.showinfo('Success','Food modified')

inv.destroy()

self.inventory()

modify\_main=Toplevel()

e1=StringVar()

e2=StringVar()

e3=StringVar()

e4=StringVar()

e5=StringVar()

e6=StringVar()

e7=StringVar()

maxrow=ws.max\_row

Label(modify\_main,text="Enter the details").grid(row=0,column=0,sticky=S)

Label(modify\_main,text="Existing Food Number").grid(row=1,column=0,sticky=S)

Label(modify\_main,text="Existing Food Name").grid(row=2,column=0,sticky=S)

Label(modify\_main,text="Food Type").grid(row=3,column=0,sticky=S)

Label(modify\_main,text="Food Number").grid(row=4,column=0,sticky=S)

Label(modify\_main,text="Food Name").grid(row=5,column=0,sticky=S)

Label(modify\_main,text="Quantity").grid(row=6,column=0,sticky=S)

Label(modify\_main,text="Price").grid(row=7,column=0,sticky=S)

Entry(modify\_main,textvariable=e6).grid(row=1,column=1,sticky=S)

Entry(modify\_main,textvariable=e7).grid(row=2,column=1,sticky=S)

Entry(modify\_main,textvariable=e1).grid(row=3,column=1,sticky=S)

Entry(modify\_main,textvariable=e2).grid(row=4,column=1,sticky=S)

Entry(modify\_main,textvariable=e3).grid(row=5,column=1,sticky=S)

Entry(modify\_main,textvariable=e4).grid(row=6,column=1,sticky=S)

Entry(modify\_main,textvariable=e5).grid(row=7,column=1,sticky=S)

Button(modify\_main,text="MODIFY FOOD",command=modify).grid(row=8,column=2,sticky=S)

modify\_main.mainloop()

def delete(self,wb,ws,ws1,ws2,inv):

defdelete\_food():

check=0;whatrow=0

foodno=e1.get()

foodname=e2.get()

maxrow=ws.max\_row

fori in range(2,maxrow+1):

if str(foodno)==str(ws['b'+str(i)].value) and str(foodname)==str(ws['c'+str(i)].value):

check=1

whatrow=i

break

else:

tkMessageBox.showerror('Error','Food item does not exist. \n Enter a valid food item to delete')

delete.destroy()

if check==1:

inv\_sheet=wb.create\_sheet('inv\_sheet')

sales\_sheet=wb.create\_sheet('sales\_sheet')

report\_sheet=wb.create\_sheet('report\_sheet')

fori in range(1,maxrow+1):

ifi != int(whatrow):

inv\_sheet['a'+str(i)].value= ws['a'+str(i)].value

inv\_sheet['b'+str(i)].value= ws['b'+str(i)].value

inv\_sheet['c'+str(i)].value= ws['c'+str(i)].value

inv\_sheet['d'+str(i)].value= ws['d'+str(i)].value

inv\_sheet['e'+str(i)].value= ws['e'+str(i)].value

sales\_sheet['a'+str(i)].value= ws1['a'+str(i)].value

sales\_sheet['b'+str(i)].value= ws1['b'+str(i)].value

sales\_sheet['c'+str(i)].value= ws1['c'+str(i)].value

sales\_sheet['d'+str(i)].value= ws1['d'+str(i)].value

sales\_sheet['e'+str(i)].value= ws1['e'+str(i)].value

report\_sheet['a'+str(i)].value= ws2['a'+str(i)].value

report\_sheet['b'+str(i)].value= ws2['b'+str(i)].value

report\_sheet['c'+str(i)].value= ws2['c'+str(i)].value

report\_sheet['d'+str(i)].value= ws2['d'+str(i)].value

report\_sheet['e'+str(i)].value= ws2['e'+str(i)].value

tkMessageBox.showinfo('Success','Food Item Deleted')

wb.remove\_sheet(ws)

wb.remove\_sheet(ws1)

wb.remove\_sheet(ws2)

inv\_sheet.title = 'Inventory'

sales\_sheet.title = 'Sales'

report\_sheet.title = 'Report'

wb.save('master\_sheet.xlsx')

inv.destroy()

self.inventory()

delete=Toplevel()

e1=StringVar()

e2=StringVar()

Label(delete,text="Enter the details")

Label(delete,text="Food Number").grid(row=0,column=0,sticky=S)

Label(delete,text="Food Name").grid(row=1,column=0,sticky=S)

Entry(delete,textvariable=e1).grid(row=0,column=1,sticky=S)

Entry(delete,textvariable=e2).grid(row=1,column=1,sticky=S)

b1=Button(delete,text="Delete",command=delete\_food).grid(row=4,column=4,sticky=W)

delete.mainloop()

defsearch\_by\_name(self,wb,ws,ws1,ws2):

def search1():

foodname=e1.get()

maxrow=ws.max\_row

fori in range(2,maxrow+1):

if str(foodname)==str(ws['c'+str(i)].value):

Label(sbn,text="Food Type").grid(row=8,column=0,sticky=S)

Label(sbn,text="Food No").grid(row=8,column=1,sticky=S)

Label(sbn,text="Food Name").grid(row=8,column=2,sticky=S)

Label(sbn,text="Quantity").grid(row=8,column=3,sticky=S)

Label(sbn,text="Price").grid(row=8,column=4,sticky=S)

Label(sbn,text=str(ws['a'+str(i)].value)).grid(row=9,column=0,sticky=S)

Label(sbn,text=str(ws['b'+str(i)].value)).grid(row=9,column=1,sticky=S)

Label(sbn,text=str(ws['c'+str(i)].value)).grid(row=9,column=2,sticky=S)

Label(sbn,text=str(ws['d'+str(i)].value)).grid(row=9,column=3,sticky=S)

Label(sbn,text=str(ws['e'+str(i)].value)).grid(row=9,column=4,sticky=S)

break

else:

tkMessageBox.showerror('Error','Food item does not exist')

sbn.destroy()

sbn=Toplevel()

e1=StringVar()

Label(sbn,text="Enter Food Name").grid(row=1,column=0,sticky=S)

Entry(sbn,textvariable=e1).grid(row=1,column=1,sticky=S)

Button(sbn,text="Search for food",command=search1).grid(row=6,column=1,sticky=W)

sbn.mainloop()

defsearch\_by\_fno(self,wb,ws,ws1,ws2):

def search2():

foodno=e1.get()

maxrow=ws.max\_row

fori in range(2,maxrow+1):

if str(foodno)==str(ws['b'+str(i)].value):

Label(sbf,text="Food Type").grid(row=8,column=0,sticky=S)

Label(sbf,text="Food No").grid(row=8,column=1,sticky=S)

Label(sbf,text="Food Name").grid(row=8,column=2,sticky=S)

Label(sbf,text="Quantity").grid(row=8,column=3,sticky=S)

Label(sbf,text="Price").grid(row=8,column=4,sticky=S)

Label(sbf,text=str(ws['a'+str(i)].value)).grid(row=9,column=0,sticky=S)

Label(sbf,text=str(ws['b'+str(i)].value)).grid(row=9,column=1,sticky=S)

Label(sbf,text=str(ws['c'+str(i)].value)).grid(row=9,column=2,sticky=S)

Label(sbf,text=str(ws['d'+str(i)].value)).grid(row=9,column=3,sticky=S)

Label(sbf,text=str(ws['e'+str(i)].value)).grid(row=9,column=4,sticky=S)

break

else:

tkMessageBox.showerror('Error','Food item does not exist')

sbf.destroy()

sbf=Toplevel()

e1=StringVar()

Label(sbf,text="Enter Food Number").grid(row=1,column=0,sticky=S)

Entry(sbf,textvariable=e1).grid(row=1,column=1,sticky=S)

Button(sbf,text="Search for food",command=search2).grid(row=6,column=1,sticky=W)

sbf.mainloop()

access\_inventory=Inventory()

class Sales(Inventory):

def \_\_init\_\_(self):

self.value1=0

self.wb=load\_workbook('master\_sheet.xlsx')

self.ws\_inv=self.wb['Inventory']

self.ws\_sales=self.wb['Sales']

self.ws\_report=self.wb['Report']

self.dict1={}

defsave\_data(self):

self.wb.save('master\_sheet1.xlsx')

os.remove('master\_sheet.xlsx')

os.rename('master\_sheet1.xlsx','master\_sheet.xlsx')

defsales\_module\_user(self):

definc\_and\_dec(x,y,z,action):

if action=='add':

if x<int(self.ws\_inv['d'+str(y)].value):

z['text']=str(x+1)

z.grid(row=y,column=5,sticky=W)

else:

tkMessageBox.showerror("Error","This is the maximum stock in the inventory",parent=sales2)

elif action=='remove':

ifint(z['text']) >=1:

z['text']=str(x-1)

z.grid(row=y,column=5,sticky=W)

self.dict1[y]=int(z['text'])

sales2 = Tk()

sales2.geometry('500x305+500+200')

sales2.maxsize(height=500,width=500)

sales2.title('SALES')

sales2.configure(bg='white')

Label(sales2,text='Food Type',borderwidth=2,bg='yellow',relief='ridge').grid(row=0,column=0,columnspan=2,sticky=W)

Label(sales2,text='Food Number',borderwidth=2,bg='yellow',relief='ridge').grid(row=0,column=1,columnspan=2,sticky=W)

Label(sales2,text='Food Name',borderwidth=2,bg='yellow',relief='ridge').grid(row=0,column=2,columnspan=3,sticky=W)

Label(sales2,text='Price',borderwidth=2,bg='yellow',relief='ridge').grid(row=0,column=3,columnspan=1,sticky=W)

Label(sales2,text='Order',borderwidth=2,bg='yellow',relief='ridge').grid(row=0,column=5,columnspan=3,sticky=W)

maxrow=self.ws\_inv.max\_row

fori in range(2,maxrow+1):

s1='a'+str(i);ftype=str(self.ws\_inv[s1].value)

s2='b'+str(i);fno=str(self.ws\_inv[s2].value)

s3='c'+str(i);fname=str(self.ws\_inv[s3].value)

s4='d'+str(i);qty=str(self.ws\_inv[s4].value)

s5='e'+str(i);price=str(self.ws\_inv[s5].value)

ifqty != '0' and ftype!='None' and fno!='None' and fname!='None' and qty!='None' and price!='None':

Label(sales2,text=ftype,font="bold",bg='white',fg='black').grid(row=i,column=0,sticky=W)

Label(sales2,text=fno,font="bold",bg='white',fg='black').grid(row=i,column=1,sticky=W)

Label(sales2,text=fname,font="bold",fg='black',bg='white').grid(row=i,column=2,sticky=W)

Label(sales2,text=price,font="bold",bg='white',fg='black').grid(row=i,column=3,sticky=W)

l1=Label(sales2,text='0');l1.grid(row=i,column=5,sticky=E)

Button(sales2,text='-',bg='orange red',command=lambda c=i,d=l1: inc\_and\_dec(int(d['text']),c,d,'remove')).grid(row=i,column=4,sticky=E)

Button(sales2,text='+',bg='green2',command=lambda a=i,b=l1: inc\_and\_dec(int(b['text']),a,b,'add')).grid(row=i,column=6,sticky=E)

self.dict1[i]=0

defProceedToCheckout():

self.Checkout()

Button(sales2,text="Proceed to Checkout",command=ProceedToCheckout,bg='yellow').grid(row=maxrow+1,column=1)

sales2.mainloop()

def Checkout(self):

row=self.ws\_sales.max\_row

fori in range(2,row+1):

string='d'+str(i)

self.ws\_sales[string]=self.dict1[i]

self.save\_data()

self.wb.close()

self.lastpage()

deflastpage(self):

lastpage=Tk()

maxrow=self.ws\_sales.max\_row

Label(lastpage,text='Food Type',borderwidth=2,relief='ridge').grid(row=0,column=0,sticky=W)

Label(lastpage,text='Food Number',borderwidth=2,relief='ridge').grid(row=0,column=1,sticky=W)

Label(lastpage,text='Food Name',borderwidth=2,relief='ridge').grid(row=0,column=2,sticky=W)

Label(lastpage,text='Price',borderwidth=2,relief='ridge').grid(row=0,column=3,sticky=W)

count=0

fori in range(2,maxrow+1):

s1='a'+str(i);ftype=str(self.ws\_sales[s1].value)

s2='b'+str(i);fno=str(self.ws\_sales[s2].value)

s3='c'+str(i);fname=str(self.ws\_sales[s3].value)

s4='d'+str(i);order=str(self.ws\_sales[s4].value)

s5='e'+str(i);price=str(self.ws\_sales[s5].value)

string ='d'+ str(i)

ifint(self.ws\_sales[string].value)!=0:

Label(lastpage,text=ftype).grid(row=i,column=0,sticky=W)

Label(lastpage,text=fno).grid(row=i,column=1,sticky=N+E+W+S)

Label(lastpage,text=fname).grid(row=i,column=2,sticky=W)

Label(lastpage,text=int(order)\*int(price)).grid(row=i,column=3,sticky=W)

count+=int(order)\*int(price)

defpay\_bill():

lastpage.destroy()

self.report()

Label(lastpage,text="TOTAL PRICE:"+ str(count)).grid(row=maxrow+1,column=3)

Button(lastpage,text="GO BACK AND ORDER MORE FOOD",bg='red',command=lastpage.destroy).grid(row=maxrow+2,column=0)

Button(lastpage,text="PAY BILL AND ENJOY YOUR MEAL",bg='red',command=pay\_bill).grid(row=maxrow+2,column=3)

def report(self):

wb1=load\_workbook('master\_sheet.xlsx')

ws1=wb1['Report']

ws2=wb1['Sales']

ws3=wb1['Inventory']

maxrow=self.ws\_report.max\_row

fori in range(2,maxrow+1):

Ordered\_food=int(self.ws\_sales['d'+str(i)].value)

self.ws\_inv['d'+str(i)].value=str(int(self.ws\_inv['d'+str(i)].value)-Ordered\_food)

self.ws\_report['d'+str(i)].value=str(int(self.ws\_report['d'+str(i)].value)+Ordered\_food)

self.ws\_sales['d'+str(i)].value=0

thankyou=Toplevel()

thankyou.configure(bg="blue")

thankyou.geometry("500x500+500+200")

thankyou.maxsize(width=500,height=500)

Label(thankyou,text="Thank you for coming.",font=('Times',40),bg="yellow",fg="red").place(x=1,y=150)

Label(thankyou,text="Enjoy your meal",font=('Times',40),bg="yellow",fg="red").place(x=1,y=217)

access\_sales=Sales()

deflogin\_page(x):

def validate():

username=a.get()

password=b.get()

wb=load\_workbook('master\_sheet.xlsx')

if x==1:

ws1=wb['Admin']; row1=ws1.max\_row

fori in range(1,row1+1):

string='b'+str(i); string1='c'+str(i)

if (ws1[string].value==username) and (ws1[string1].value==password):

access\_inventory.inventory()

break

#ADMIN LOGIN INTO INVENTORY

else:

tkMessageBox.showerror('Error','Wrong username or password',parent=root\_login)

elif x==0:

ws1=wb['Admin']; row1=ws1.max\_row

ws2=wb['User']; row2=ws2.max\_row

fori in range(1,row1+1):

string='b'+str(i); string1='c'+str(i)

if (ws1[string].value==username) and (ws1[string1].value==password):

#ADMIN LOGIN INTO SALES

access\_sales.sales\_module\_user()

break

for j in range(1,row2+1):

string='b'+str(j); string1='c'+str(j)

if (ws2[string].value==username) and (ws2[string1].value==password):

#USER LOGIN INTO SALES

access\_sales.sales\_module\_user()

break

else:

tkMessageBox.showerror('Error','Wrong username or password',parent=root)

wb.close()

root\_login=Toplevel()

root\_login.geometry('1050x500')

w = 1050

h = 500

ws = root\_login.winfo\_screenwidth()

hs = root\_login.winfo\_screenheight()

x1 = (ws/2) - (w/2)

y1 = (hs/2) - (h/2)

root\_login.geometry('%dx%d+%d+%d' % (w, h, x1, y1))

root\_login.maxsize(height=500,width=1050)

root\_login.configure(bg='gold')

a=StringVar()

b=StringVar()

img=PhotoImage(file='csfb.gif')

Label(root\_login,image=img,bg='gold',height=400,width=600).grid(row=0,column=0,rowspan=30)

Label(root\_login,text='Username: ',font=('Times',20),bg='gold',fg='blue2').grid(row=20,column=2,sticky=W)

Entry(root\_login,font=('Times',20),textvariable=a).grid(row=20,column=3)

Label(root\_login,text='Password: ',font=('Times',20),bg='gold',fg='blue2').grid(row=21,column=2,sticky=W)

Entry(root\_login,font=('Times',20),show='\*',textvariable=b).grid(row=21,column=3)

Button(root\_login,text='LOGIN',bg='medium blue',fg='gold',height=3,width=15,command=validate).grid(row=24,column=3)

root\_login.mainloop()

defsign\_up\_now():

root1=Toplevel(root)

root1.configure(bg='DeepSkyBlue2')

w = 650

h = 330

ws = root1.winfo\_screenwidth()

hs = root1.winfo\_screenheight()

x2 = (ws/2) - (w/2)

y2 = (hs/2) - (h/2)

root1.geometry('%dx%d+%d+%d' % (w, h, x2, y2))

root1.maxsize(height=330,width=650)

user1=StringVar()

pass1=StringVar()

pass2=StringVar()

name1=StringVar()

v=IntVar()

def signup():

name=name1.get()

user=user1.get()

pass\_1=pass1.get()

pass\_2=pass2.get()

person=v.get()

defadd\_user(name,user,pass\_1):

wb=load\_workbook('master\_sheet.xlsx')

ws=wb['User']

ws.append([name,user,pass\_1])

wb.save('master\_sheet1.xlsx')

os.remove('master\_sheet.xlsx')

os.rename('master\_sheet1.xlsx','master\_sheet.xlsx')

wb.close()

defadd\_admin(name,user,pass\_1):

wb=load\_workbook('master\_sheet.xlsx')

ws=wb['Admin']

ws.append([name,user,pass\_1])

wb.save('master\_sheet1.xlsx')

os.remove('master\_sheet.xlsx')

os.rename('master\_sheet1.xlsx','master\_sheet.xlsx')

wb.close()

if (len(name)>0 and len(user)>0):

iflen(pass\_1)>=8:

if (pass\_1==pass\_2) and person==0:

add\_user(name,user,pass\_1)

tkMessageBox.showinfo('Success','Signupsuccessful!',parent=root1)

root1.destroy()

elif (pass\_1==pass\_2) and person==1:

add\_admin(name,user,pass\_1)

tkMessageBox.showinfo('Success','Signupsuccessful!',parent=root1)

root1.destroy()

else:

tkMessageBox.showerror('Error','Passwords don\'t match',parent=root1)

else:

tkMessageBox.showerror('Error','Password must be atleast 8 characters long',parent=root1)

else:

tkMessageBox.showerror('Error','No name or username given',parent=root1)

Label(root1,text='WELCOME NEW USER',justify=CENTER,font=('Times',30),fg='gray1',bg='DeepSkyBlue2').grid(row=0,column=0,rowspan=2,columnspan=3,sticky=N+E+W+S)

Label(root1,text='Enter your name: ',font=('Times',20),bg='DeepSkyBlue2',fg='gray1').grid(row=3,column=0,sticky=W)

Entry(root1,font=('Times',20),textvariable=name1,bg='white').grid(row=3,column=1)

Label(root1,text='Enter your preferred username: ',font=('Times',20),bg='DeepSkyBlue2',fg='gray1').grid(row=4,column=0,sticky=W)

Entry(root1,font=('Times',20),textvariable=user1,bg='white').grid(row=4,column=1)

Label(root1,text='Enter password: ',font=('Times',20),bg='DeepSkyBlue2',fg='gray1').grid(row=5,column=0,sticky=W)

Entry(root1,font=('Times',20),show='\*',textvariable=pass1,bg='white').grid(row=5,column=1)

Label(root1,font=('Times',20),text='Confirm password: ',bg='DeepSkyBlue2',fg='gray1').grid(row=6,column=0,rowspan=2,sticky=W)

Entry(root1,font=('Times',20),show='\*',textvariable=pass2,bg='white').grid(row=6,column=1,rowspan=2)

Button(root1,text='SIGN UP',fg='yellow2',bg='red2',height=3,width=15,command=signup).grid(row=16,column=1,rowspan=2,pady=4)

Radiobutton(root1,text='I am an Administrator',font=('Times',10),bg='DeepSkyBlue2',fg='blue',variable=v,value=1).grid(row=13,column=1,pady=4,sticky=W)

Radiobutton(root1,text='I am a User',font=('Times',10),bg='DeepSkyBlue2',fg='blue',variable=v,value=0).grid(row=10,column=1,pady=4,sticky=W)

root1.mainloop()

root=Tk()

#-------------------Positioning of the root-----------------------------

w = 1000

h = 600

ws = root.winfo\_screenwidth()

hs = root.winfo\_screenheight()

x3 = (ws/2) - (w/2)

y3 = (hs/2) - (h/2)

root.geometry('%dx%d+%d+%d' % (w, h, x3, y3))

root.maxsize(height=600,width=1000)

#--------------------------------------------------------------------------

root.title('Welcome to the Food Box')

root.configure(bg='blue2')

img=PhotoImage(file='welcome-food1.gif')

Label(root,image=img,bg='blue2',height=600,width=600).grid(row=0,column=0,rowspan=10)

Button(root,text='Sales',font=('Times',20),fg='black',bg='firebrick1',height=5,width=25,command=lambda: login\_page(x=0)).grid(row=2,column=1)

Button(root,text='Inventory\n(Only Administrator Access)',fg='black',bg='gold',font=('Times',20),height=5,width=25,command=lambda: login\_page(x=1)).grid(row=3,column=1)

Button(root,text='New to Food Box?\nSign Up Now',font=('Times',20),fg='black',bg='green',height=5,width=25,command=sign\_up\_now).grid(row=4,column=1)

root.mainloop()

**OUTPUT SNIPPETS**

**FUTURE DEVELOPMENT**

Scope for future development in this project is high. ***Linking between various vendors***, hence acting as a platform for competitive food selling, *i.e.* giving the User a choice of ***ordering the same food*** item from ***different vendors***, based on two parameters, ***price*** and ***previous experience.***

If used as a ***real commercial software***, providing efficient billing systems and option to pay using ***online banking or e-wallets*** such as ***Paypal*** or ***Paytm*** already having a wide reach, will ***increase the magnitude of users*** commercially.

Future development would also involve a more efficient and secure database management system, security being of main concern here, as sufficient security measures are not necessary for such a project of such small scale, but holding great promises in the future.

**BIBLIOGRAPHY**

We have referred to many sources some in the Internet, to develop our project .We have listed them below:

* ***Python for Class XII ,*** by Sumita Arora
* ***Tkinter Documentation***
* ***Openpyxl Documentation***

And various other random websites to clear our queries based on functionality of the modules.