Python Lab
Lab - Assignment 01

Design a Simple Python Application that should be based on real world applications:
Can be 1 from these topics or your choice
1)Web Development
2)Game Development
3)Artificial Intelligence and Machine Learning
5)Desktop GUI
6)Image Processing
7)Text Processing
8)Web Scraping Applications
9)Data Science and Data Visualization

Roll No.	01
Name	Aamir Ansari
Class	D10-A
Subject	Python Lab
LO Mapped	LO1: Understand the structure, syntax, and semantics of the Python language LO2:Interpret advanced data types and functions in python LO3:illustrate the concepts of object-oriented programming as used in Python LO4:Create Python applications using modules, packages, multithreading and exception handling. LO5: Gain proficiency in evaluate database operations in python LO6:Design and Develop cost-effective robust applications using the latest Python trends and technologies

AIM:

To design a Simple Python Application that should be based on real world applications

INTRODUCTION:

Python is easy to use, powerful, and versatile, making it a great choice for beginners and experts alike. Python's readability makes it a great first programming language — it allows you to think like a programmer and not waste time with confusing syntax. Python is accessible by design, making it one of the fastest languages in terms of speed of development. A user-friendly environment in the hands of your development team means less time wrestling with your building tool and more time spent actually building.

Advantages of Python for GUI development
There are many advantages of Python that help you get results fast within the field of web development:

1. Python has a large selection of pre-built libraries for just about anything. Scientific computing, image processing, data processing, machine learning, deep learning—you name it, Python has it.
2. Python code takes less time to write due to its simple and clean syntax. Because of this, code written in Python lends itself very well to creating quick prototypes.
3. Python accelerates the ROI of commercial projects. The reason behind this is similar to the previous point: you can write and ship your code faster. This is especially important for startups.
4. Python has a built-in framework for unit tests. This helps you ship bug-free code.

In addition to Python's standard features, one of its major strengths in web development is the variety of web frameworks it offer
Python provides various options for developing graphical user interfaces (GUIs). Most important are listed below.

1) Tkinter − Tkinter is the Python interface to the Tk GUI toolkit shipped with Python. We would look this option in this chapter.

2) wxPython − This is an open-source Python interface for wxWindows http://wxpython.org.

3) JPython − JPython is a Python port for Java which gives Python scripts seamless access to Java class libraries on the local machine http://www.jython.org.

There are many other interfaces available, which you can find them on the net.

TKINTER PROGRAMMING

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.
Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps −

1. Import the Tkinter module.
2. Create the GUI application main window.
3. Add one or more of the above-mentioned widgets to the GUI application.
4. Enter the main event loop to take action against each event triggered by the user.

BILLING SYSTEM:

A billing system which is used at the hotels,restaurants,etc. Can be made easily with the help of Python. We have made a Food billing system.
Functionalities added are:

1. Calculate bill as per the order
2. Has a date marked on every bill
3. Generate a detailed bill of the order
4. A database which allows us to see the previous orders
5. A stock tab to view the remaining products

This system can prove its efficiency in other fields too just by changing the name of products and their prices.

RESULTS:

Code:

```
from tkinter import *
from tkinter import ttk
from tkcalendar import DateEntry
from tkinter import messagebox
import sqlite3  as db

def connection():
   connectObj = db.connect("shopManagement.db")
   cur = connectObj.cursor()
   sql = '''
   create table if not exists sellings (
      date string,
      product string,
      price number,
      quantity number,
      total number
      )
   '''
   cur.execute(sql)
   connectObj.commit()

connection()
window=Tk()
window.title("Food ordering System")
tabs = ttk.Notebook(window)
root= ttk.Frame(tabs)
root2=ttk.Frame(tabs)

tabs.add(root, text ='Sell')
tabs.add(root2, text ='Stock')
tabs.pack(expand = 1, fill ="both")


#---------------------------------------------tab1 ---------------------------------

def GenerateBill():
   connectObj = db.connect("shopManagement.db")
   cur = connectObj.cursor()

   global billarea
   if p1quantity.get()==0 and p2quantity.get()==0 and p3quantity.get()==0 and
p4quantity.get()==0:
      messagebox.showerror("Error","No product purchased")
   else:
      billarea.delete('1.0',END)
      billarea.insert(END,"\t|| Food ordering System ||")
```

```
    billarea.insert(END,"\n__________________________________________\n")
    billarea.insert(END,"\nDate\t Products\tPrice\t   QTY\t Total")

billarea.insert(END,"\n==========================================")

    price= IntVar()
    price2=IntVar()
    price3=IntVar()
    price4=IntVar()

    print(dateE.get())
    price=price2=price3=price4=0

    if p1quantity.get()!=0:
        price=p1quantity.get()*p1price.get()
        print(price)
        billarea.insert(END,f"\n{dateE.get()}\t Biryani \t{p1price.get()}\t
{p1quantity.get()}\t {price}")

        sql = '''
        INSERT INTO Sellings VALUES
        (?, ?, ?, ?,?)
        '''
        cur.execute(sql,(dateE.get(),'Product-1',p1price.get(),p1quantity.get(),price))
        connectObj.commit()

    if p2quantity.get()!=0:
        price2=(p2quantity.get()*p2price.get())
        print(price2)
        billarea.insert(END,f"\n{dateE.get()}\t Pulao \t{p2price.get()}\t
{p2quantity.get()}\t {price2}")

        sql = '''
        INSERT INTO Sellings VALUES
        (?, ?, ?, ?,?)
        '''
        print(dateE.get(),'Product-2',p2price.get(),p2quantity.get(),price2)
        cur.execute(sql,(dateE.get(),'Product-2',p2price.get(),p2quantity.get(),price2))
        connectObj.commit()

    if p3quantity.get()!=0:
        price3=p3quantity.get()*p1price.get()
        print(price3)
        billarea.insert(END,f"\n{dateE.get()}\t Pani puri \t{p3price.get()}\t
{p3quantity.get()}\t {price3}")

        sql = '''
        INSERT INTO Sellings VALUES
        (?, ?, ?, ?,?)
        '''
```

```
            cur.execute(sql,(dateE.get(),'Product-3',p3price.get(),p3quantity.get(),price3))
            connectObj.commit()

        if p4quantity.get()!=0:
            price4=p4quantity.get()*p1price.get()
            billarea.insert(END,f"\n{dateE.get()}\t Paneer tikka \t{p4price.get()}\t
{p4quantity.get()}\t {price4}")

            sql = '''
            INSERT INTO Sellings VALUES
            (?, ?, ?, ?,?)
            '''
            cur.execute(sql,(dateE.get(),'Product-4',p4price.get(),p4quantity.get(),price4))
            connectObj.commit()

        Totalprice=IntVar()
        Totalprice=price+price2+price3+price4

        Totalquantity=IntVar()

Totalquantity=p1quantity.get()+p2quantity.get()+p3quantity.get()+p4quantity.get()
        billarea.insert(END,f"\nTotal \t \t  \t{Totalquantity}\t {Totalprice}")


def view():
    connectObj = db.connect("shopManagement.db")
    cur = connectObj.cursor()

    sql = 'Select * from Sellings'
    cur.execute(sql)

    rows=cur.fetchall()
    viewarea.insert(END,f"Date\t Product\t  Price of 1\t  Quantity\t  Price\n")


    for i in rows:
        allrows=""
        for j in i:
            allrows+=str(j)+'\t'
        allrows+='\n'
        viewarea.insert(END,allrows)

dateL=Label(root,text="Date",bg="DodgerBlue2",width=12,font=('arial',15,'bold'))
dateL.grid(row=0,column=0,padx=7,pady=7)

dateE=DateEntry(root,width=12,font=('arial',15,'bold'))
dateE.grid(row=0,column=1,padx=7,pady=7)

l=Label(root, text="Product",font=('arial',15,'bold'),bg="DodgerBlue2",width=12)
l.grid(row=1,column=0,padx=7,pady=7)
```

```
l=Label(root, text="Price",font=('arial',15,'bold'),bg="DodgerBlue2",width=12)
l.grid(row=1,column=1,padx=7,pady=7)

l=Label(root, text="Quantity",font=('arial',15,'bold'),bg="DodgerBlue2",width=12)
l.grid(row=1,column=2,padx=7,pady=7)

#----product 1---------------------------------------------------
p1name=StringVar()
p1name.set('Biryani')

p1price=IntVar()
p1price.set(100)

p1quantity=IntVar()
p1quantity.set(0)

l=Label(root, text=p1name.get(),font=('arial',15,'bold'),width=12)
l.grid(row=2,column=0,padx=7,pady=7)

l=Label(root, text=p1price.get(),font=('arial',15,'bold'),width=12)
l.grid(row=2,column=1,padx=7,pady=7)

t=Entry(root,textvariable=p1quantity,font=('arial',15,'bold'),width=12)
t.grid(row=2,column=2,padx=7,pady=7)

#----product 2------------------------------------------------------------
p2name=StringVar()
p2name.set('Pulao')

p2price=IntVar()
p2price.set(200)

p2quantity=IntVar()
p2quantity.set(0)

l=Label(root, text=p2name.get(),font=('arial',15,'bold'),width=12)
l.grid(row=3,column=0,padx=7,pady=7)

l=Label(root, text=p2price.get(),font=('arial',15,'bold'),width=12)
l.grid(row=3,column=1,padx=7,pady=7)

t=Entry(root,textvariable=p2quantity,font=('arial',15,'bold'),width=12)
t.grid(row=3,column=2,padx=7,pady=7)

#----product 3----
p3name=StringVar()
p3name.set('Pani puri')

p3price=IntVar()
```

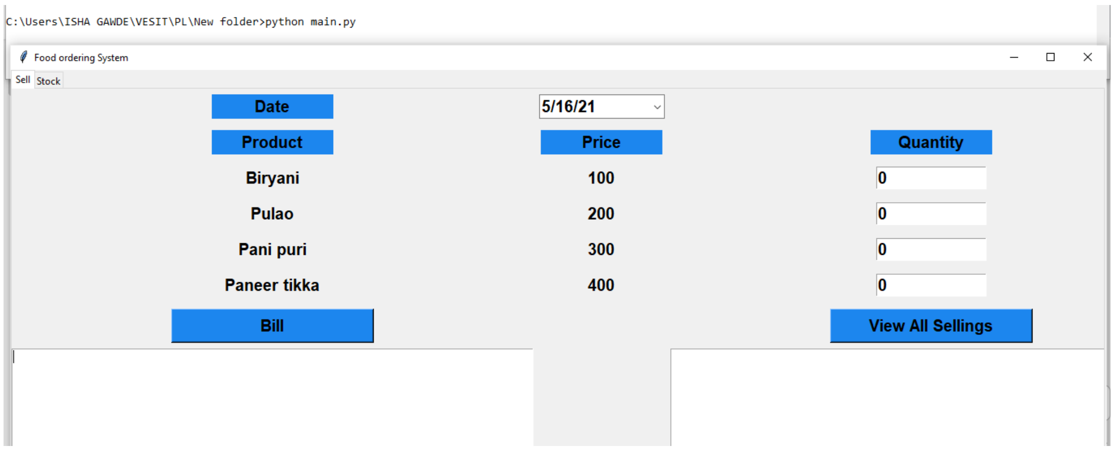
```
p3price.set(300)

p3quantity=IntVar()
p3quantity.set(0)

l=Label(root, text=p3name.get(),font=('arial',15,'bold'),width=12)
l.grid(row=4,column=0,padx=7,pady=7)

l=Label(root, text=p3price.get(),font=('arial',15,'bold'),width=12)
l.grid(row=4,column=1,padx=7,pady=7)

t=Entry(root,textvariable=p3quantity,font=('arial',15,'bold'),width=12)
t.grid(row=4,column=2,padx=7,pady=7)

#----product 4----
p4name=StringVar()
p4name.set('Paneer tikka')

p4price=IntVar()
p4price.set(400)

p4quantity=IntVar()
p4quantity.set(0)

l=Label(root, text=p4name.get(),font=('arial',15,'bold'),width=12)
l.grid(row=5,column=0,padx=7,pady=7)

l=Label(root, text=p4price.get(),font=('arial',15,'bold'),width=12)
l.grid(row=5,column=1,padx=7,pady=7)

t=Entry(root,textvariable=p4quantity,font=('arial',15,'bold'),width=12)
t.grid(row=5,column=2,padx=7,pady=7)

#-----------------------bill------------------------
billarea=Text(root)

submitbtn=Button(root,command=GenerateBill,text="Bill",
font=('arial',15,'bold'),bg="DodgerBlue2",width=20 )

submitbtn.grid(row=6,column=0,padx=7,pady=7)

viewbtn=Button(root,command=view,text="View All Sellings",
font=('arial',15,'bold'),bg="DodgerBlue2",width=20 )

viewbtn.grid(row=6,column=2,padx=7,pady=7)

billarea.grid(row=9,column=0)
viewarea=Text(root)
viewarea.grid(row=9,column=2)
#-------------------------------------------tab2 --------------------------------
```

```
def connection2():
    connectObj2 = db.connect("shopManagement.db")
    cur = connectObj2.cursor()
    sql = '''
    create table if not exists stocks (
        date string,
        product string,
        price number,
        quantity number
        )
    '''
    cur.execute(sql)
    connectObj2.commit()

connection2()

def addStock():
    global dateE2,qty,name,price

    connectObj = db.connect("shopManagement.db")
    cur = connectObj.cursor()
    sql = '''
            INSERT INTO stocks VALUES
            (?, ?, ?, ?)
            '''
    cur.execute(sql,(dateE2.get(),name.get(),price.get(),qty.get()))
    connectObj.commit()

def viewStock():
    connectObj = db.connect("shopManagement.db")
    cur = connectObj.cursor()

    sql = 'Select * from stocks'
    cur.execute(sql)

    rows=cur.fetchall()
    viewarea2.insert(END,f"Date \tProduct\t  Price\t  Quantity\t \n")

    for i in rows:
        allrows=""
        for j in i:
            allrows+=str(j)+'\t'
        allrows+='\n'
        viewarea2.insert(END,allrows)

dateL=Label(root2,text="Date",bg="DodgerBlue2",width=12,font=('arial',15,'bold'))
dateL.grid(row=0,column=0,padx=7,pady=7)

dateE2=DateEntry(root2,width=12,font=('arial',15,'bold'))
dateE2.grid(row=0,column=1,padx=7,pady=7)
```

```
l=Label(root2, text="Product",font=('arial',15,'bold'),bg="DodgerBlue2",width=12)
l.grid(row=1,column=0,padx=7,pady=7)

l=Label(root2, text="Price",font=('arial',15,'bold'),bg="DodgerBlue2",width=12)
l.grid(row=2,column=0,padx=7,pady=7)

l=Label(root2, text="Quantity",font=('arial',15,'bold'),bg="DodgerBlue2",width=12)
l.grid(row=3,column=0,padx=7,pady=7)

name=StringVar()
price=IntVar()
qty=IntVar()

Name=Entry(root2,textvariable=name,font=('arial',15,'bold'),width=12)
Name.grid(row=1,column=1,padx=7,pady=7)

Price=Entry(root2,textvariable=price,font=('arial',15,'bold'),width=12)
Price.grid(row=2,column=1,padx=7,pady=7)

Qty=Entry(root2,textvariable=qty,font=('arial',15,'bold'),width=12)
Qty.grid(row=3,column=1,padx=7,pady=7)

addbtn=Button(root2,command=addStock,text="Add",
font=('arial',15,'bold'),bg="DodgerBlue2",width=20)

addbtn.grid(row=4,column=1,padx=7,pady=7)

viewarea2=Text(root2)
viewarea2.grid(row=5,column=0,columnspan=2)

viewbtn2=Button(root2,command=viewStock,text="View Stock",
font=('arial',15,'bold'),bg="DodgerBlue2",width=20 )

viewbtn2.grid(row=4,column=0,padx=7,pady=7)

mainloop()
```

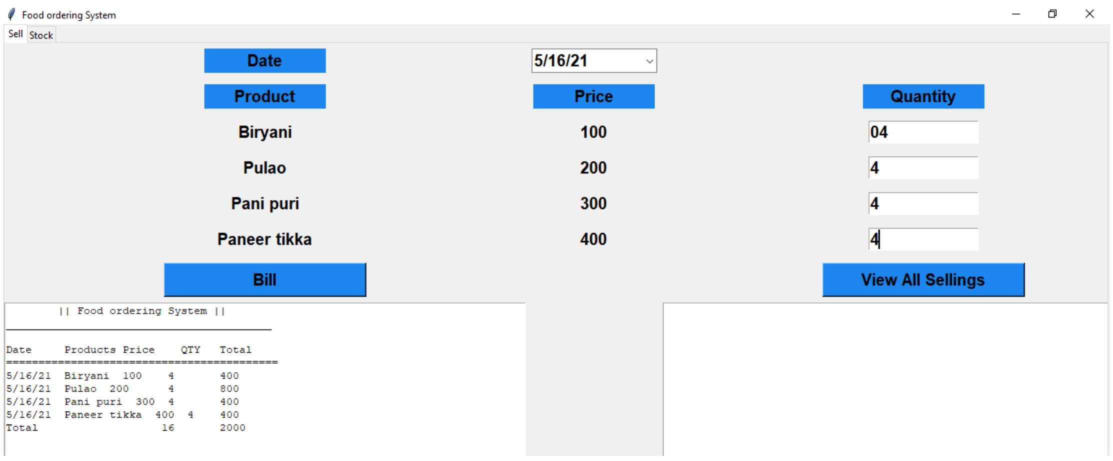
(Database is attached with the file)

<u>Output:</u>



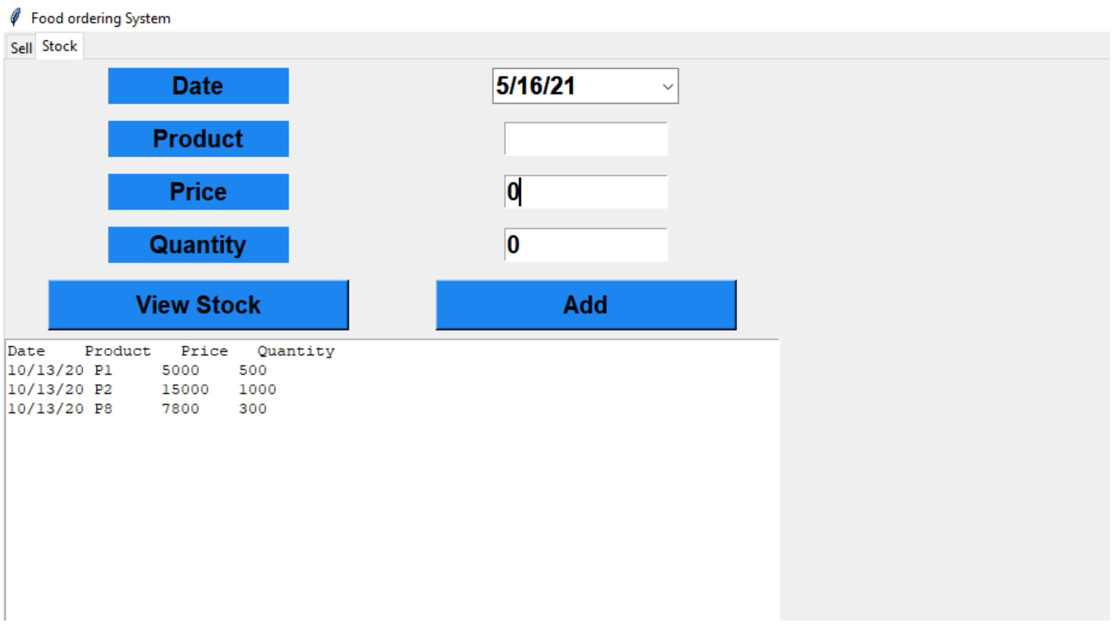
```
C:\Users\ISHA GAWDE\VESIT\PL\New folder>python main.py
```

```
Date            5/16/21
Product         Price       Quantity
Biryani         100         0
Pulao           200         0
Pani puri       300         0
Paneer tikka    400         0
```



```
Date            5/16/21
Product         Price       Quantity
Biryani         100         04
Pulao           200         4
Pani puri       300         4
Paneer tikka    400         4
```

```
        || Food ordering System ||
______________________________________

Date     Products Price    QTY   Total
==========================================
5/16/21  Biryani  100    4       400
5/16/21  Pulao  200      4       800
5/16/21  Pani puri  300  4       400
5/16/21  Paneer tikka  400  4    400
Total                   16       2000
```



```
Date        5/16/21
Product
Price       0
Quantity    0
```

```
Date      Product   Price    Quantity
10/13/20 P1      5000     500
10/13/20 P2      15000    1000
10/13/20 P8      7800     300
```

CONCLUSION:

We have successfully developed a Game Ordering System , where we can track the inventories, products, sales, etc. Based on the requirements you can add more features in this project. It connects the python program to a database and creates a sellings table with five attributes