Apeejay School

Noida, U.P.

Title: COMPUTER SCIENCE PROJECT

Subtitle: **Grocery Application**

Name: Panav Gupta
RollNo. -

Name: Mohd. Ayaan Ahmad
RollNo. -

INDEX

Acknowledgement

I want to express my deep gratitude to my teacher **Ms. Sujata Bharadwaj**, who gave us the golden opportunity to do this wonderful Computer Science project. Their guidance and encouragement were indispensable in bringing this project to fruition

We also came to know about so many new things, and we are thankful to them. We want to extend our sincere thanks to our family and friends for their unwavering support and motivation throughout this journey.

CERTIFICATE

This is to certify that **Panav Gupta and Mohd. Ayaan Ahmad**, students of class XII have successfully completed their Computer Science project under the guidance of **Ms. Sujata Bhardwaj** (Subject Teacher) during the year 2024-25.

Sign of external
Examiner

Sign of internal
Examiner

PROBLEM DEFINATION

The primary issue that the proposed Grocery Store system aims to solve is the inefficiency and inconvenience associated with manual inventory management, billing, and customer service in traditional grocery stores. This system seeks to streamline the entire process, from stock management to customer checkout, to enhance overall operational efficiency, reduce human errors, and improve customer satisfaction. Additionally, the system will address challenges like tracking stock levels, generating automated bills, inventory management, etcetera.

PROPOSED SYSTEM

The proposed system is a computerized Grocery Store management solution designed to automate various tasks involved in running a grocery store. This system will include features like inventory management, billing, customer management, and report generation. It will enable store owners and employees to keep real-time track of inventory, process sales transactions quickly, and generate detailed reports on sales and stock levels. The system will also offer an intuitive user interface for easy interaction and an underlying database and binary files for storing and retrieving all necessary data.

HARDWARE AND SOFTWARE REQUIRED

Hardware:

- A computer or server with sufficient processing power and storage capacity.
- Receipt printer for printing bills.
- Network router (if the system is to be accessed by multiple users on different machines).
- Backup storage devices for data security.

Software:

- **Operating System:** Windows, Linux, or macOS.
- **Database Management System (DBMS):** MySQL, etc.
- **Programming Language:** Python.
- **IDE (Integrated Development Environment):** Visual Studio, PyCharm, etc
- **Additional Modules:** tkinter(ttk, fileidealog), PIL(ImageTK, Image), pickle, os, shutil, mysql.connecter, random

DESIGN (logical)

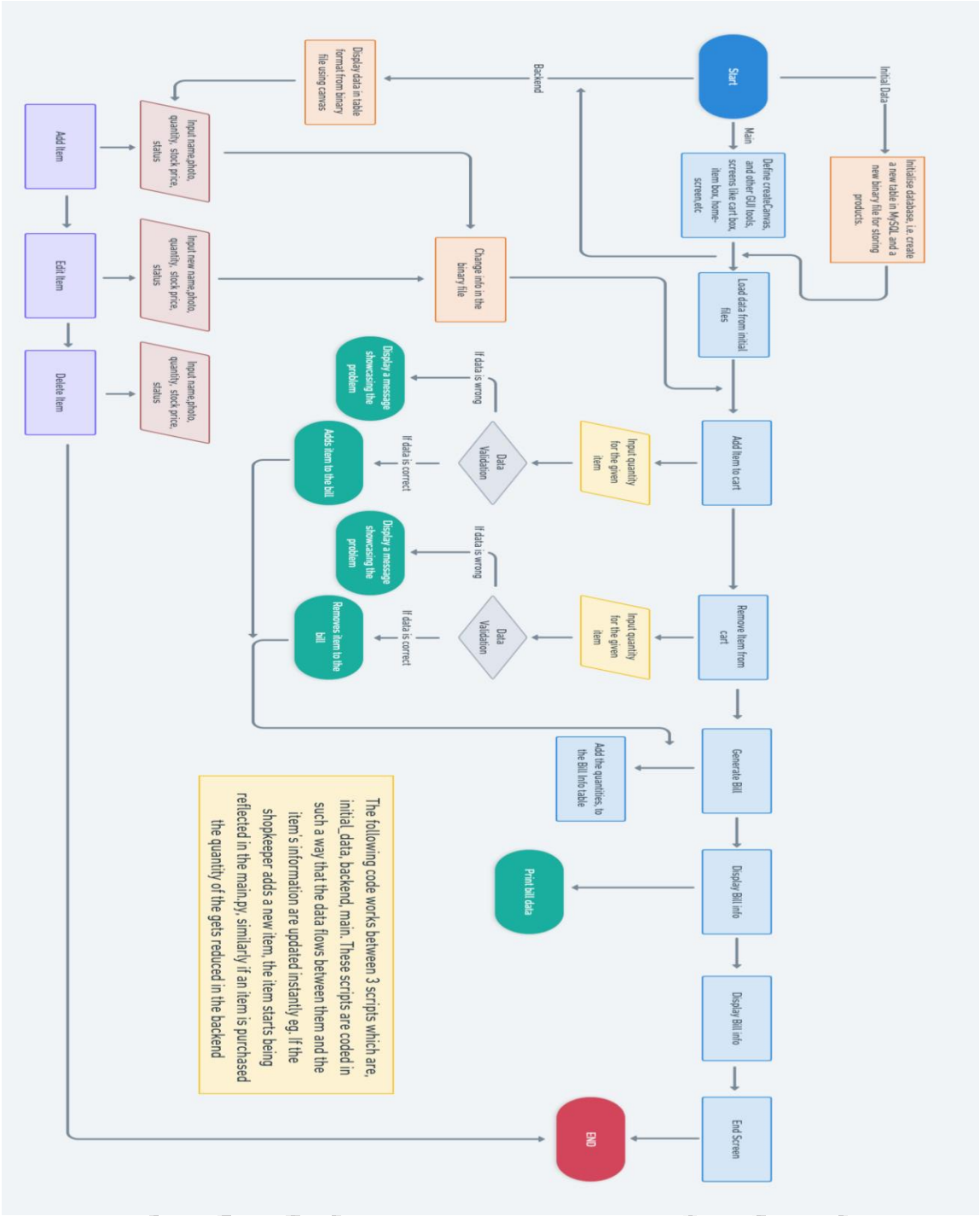
The logical design of the Grocery Store system involves defining the system's architecture and how

different modules will interact with each other. Key components include:

- **Database Design:** Tables for product bills and transactions.
- **Binary file Design:** Files for the inventory of the whole store. (FileName – details.bin)

  Structure – { Item : [stock, price, stock report] }
- **User Interface Design:** Screens for product management, billing, and report generation.
- **DATABASE NAME:** Bill_Info
- **Table Structure:** Name - BillInfo

Field	Type	Null	Key	Default	Extra
SNo	int	YES		NULL	
Name	varchar(30)	YES		NULL	
Qty	int	YES		NULL	
Price	int	YES		NULL	
TotalPrice	int	YES		NULL	



DESIGN (physical)

The physical design involves putting together the actual parts of the system. This includes:

- **Database Setup:** Creating simple tables in a database to store information about products, customers, and sales.

- **User Interface:** Designing basic screens that users will see, such as the product list, billing screen, inventory manager (for owner), etc
- **Hardware:** Setting up the necessary equipment like a computer, printer to make the system work.

SCRIPTS:

```
import pickle
import mysql.connector as sql_con
con = sql_con.connect(host="localhost", username="root", passwd="password")
cur=con.cursor()
sql="drop database bill;"
cur.execute(sql)
SQL = "CREATE DATABASE Bill;"
cur.execute(SQL)
SQL = "USE Bill"
cur.execute(SQL)
con.commit()
SQL = "CREATE TABLE BillInfo(SNo int(3), Name varchar(30), Qty int(3), Price int(4),
TotalPrice int(4)); "
cur.execute(SQL)
con.commit()
f = open("details.txt","wb")
data = {'Apples': [50, 200, 'Sufficient stock'],
        'Banana': [50, 120, 'Sufficient stock'],
        'Broccoli': [50, 80, 'Sufficient stock'],
        'Cauliflower': [50, 60, 'Sufficient stock'],
        'Guava': [50, 50, 'Sufficient stock'],
        'Mango': [50, 350, 'Sufficient stock'],
        'Okra': [50, 60, 'Sufficient stock'],
        'Onion': [50, 40, 'Sufficient stock'],
        'Potato': [50, 25, 'Sufficient stock'],
        'Spinach': [50, 45, 'Sufficient stock'],
        'Tomato': [50, 60, 'Sufficient stock'],
        'Watermelon': [50, 30, 'Sufficient stock']}
pickle.dump(data,f)
f.close()

import gui_lib as G
import tkinter as tk
from tkinter import ttk, filedialog
from PIL import ImageTk, Image
import pickle
import os
import shutil
```

```
h = G.root.winfo_screenheight()
w = G.root.winfo_screenwidth()
G.root.geometry("1280x600")
G.root.title('Edit Window')
G.root.grid_anchor('center')
G.root.resizable(False,False)
G.canvas['background']= 'white'
cenx=w/2
ceny=h/2
f = open("details.txt","rb")
info = pickle.load(f)
n = len(info)
f.close()
def test():
    print(0)
def add():
    add_win = tk.Toplevel()
    add_win.geometry("500x250")
    add_win.title('Add')
    add_win.grid_anchor('center')
    add_win.resizable(False,False)
    add_canvas = tk.Canvas(add_win, width = 500, height = 250, highlightthickness = 0,
bg='#c7c7c7')
    add_canvas.place(x = 0, y = 0)
    add_canvas.photos = []
    G.text('Name:',200,10,canvas=add_canvas,col='black',anchor=tk.W)
    new_name =
G.inputBox(200,25,30,12,bg_col='white',col='black',j='left',canvas=add_canvas)
    G.text('Stock: ',200,70,canvas=add_canvas,col='black',anchor=tk.W)
    new_stock =
G.inputBox(200,85,20,12,bg_col='white',col='black',j='left',canvas=add_canvas)
    G.text('kg',390,100,canvas=add_canvas,col='black',anchor=tk.W,size=12)
    G.text('Cost: ',200,130,canvas=add_canvas,col='black',anchor=tk.W)
    new_cost =
G.inputBox(200,145,20,12,bg_col='white',col='black',j='left',canvas=add_canvas)
    G.text('Rs per kg',390,160,canvas=add_canvas,col='black',anchor=tk.W,size=12)
    global new_path
    new_path = None
    def add_item(img):
        global n
        f = open("details.txt","rb")
        info = pickle.load(f)
        f.close()

       Name = new_name.get()
        Price = int(new_cost.get())
        Stock = int(new_stock.get())
        if Stock < 20 :
            Status = 'Low Stock, Resupply'
        else :
```

```
            Status = 'Sufficient Stock'

        info[Name] = [Stock, Price, Status]
        f = open("details.txt","wb")
        pickle.dump(info,f)
        f.close()
        ext =  os.path.splitext(img)[1]
        filename = os.path.basename(img)
        shutil.copy(img,"Assets")
        assets = "Assets\\"
        os.rename(assets+filename,assets+Name+'.jpg')
        search()
        n+=1
    def addPhoto():
        fileName = filedialog.askopenfilename(parent=add_win)
        G.image(fileName,97,84,194,168,canvas=add_canvas)
        global new_path, photo_button
        new_path =  fileName
        photo_button.destroy()
    global photo_button
    photo_button = G.button('Add
\nphoto',30,80,60,160,canvas=add_canvas,col='blue',bg_col='#c7c7c7',command=addPhoto,f
ont='Consolas 40 bold')
    G.button('OK',430,210,20,40,col='black',bg_col='white', command = lambda:
add_item(new_path), canvas = add_canvas)
def edit(item,data):
    edit_win = tk.Toplevel()
    edit_win.geometry("500x250")
    edit_win.title('Edit')
    edit_win.grid_anchor('center')
    edit_win.resizable(False,False)
    edit_canvas = tk.Canvas(edit_win, width = 500, height = 250, highlightthickness = 0,
bg='#c7c7c7')
    edit_canvas.place(x = 0, y = 0)
    edit_canvas.photos = []
    G.text('Name:',200,10,canvas=edit_canvas,col='black',anchor=tk.W)
    new_name =
G.inputBox(200,25,30,12,bg_col='white',col='black',j='left',canvas=edit_canvas,val=item)
    G.text('Stock: ',200,70,canvas=edit_canvas,col='black',anchor=tk.W)
    new_stock =
G.inputBox(200,85,20,12,bg_col='white',col='black',j='left',canvas=edit_canvas,val=data[0])
    G.text('kg',390,100,canvas=edit_canvas,col='black',anchor=tk.W,size=12)
    G.text('Cost: ',200,130,canvas=edit_canvas,col='black',anchor=tk.W)
    new_cost =
G.inputBox(200,145,20,12,bg_col='white',col='black',j='left',canvas=edit_canvas,val=data[1]
)
    G.text('Rs per kg',390,160,canvas=edit_canvas,col='black',anchor=tk.W,size=12)

    fileName = 'Assets\\'+item+'.jpg'
    G.image(fileName,97,84,194,168,canvas=edit_canvas)
```

```
    global new_path
    new_path = None
    def updatePhoto():
        global new_path
        fileName = filedialog.askopenfilename(parent=edit_win)
        G.image(fileName,97,84,194,168,canvas=edit_canvas)
        new_path = shutil.copy(fileName,"Assets")
        return new_path

    G.button('Change photo',0,140,20,186,col='blue',bg_col='white',command=updatePhoto,
canvas = edit_canvas)

    def update_inv(): #attach toedit button next to each item
        f = open("details.txt","rb")
        info = pickle.load(f)
        f.close()

        global  new_path
        Name = new_name.get()
        Stock = int(new_stock.get())
        Price = int(new_cost.get())


        if Stock < 20 :
          Status = 'Low Stock, Resupply'
        else :
          Status = 'Sufficient Stock'
        if new_path != None:
          print(new_path)
          assets = "Assets\\"
          ext =  os.path.splitext(new_path)[1]
          new_path_rename = assets+Name+'.jpg'
          if os.path.isfile(new_path_rename):
              os.remove(new_path_rename)
              print('clone file detectde')
          if os.path.isfile(new_path):
              os.rename(new_path,new_path_rename)
        else:
          ext =  os.path.splitext(fileName)[1]
          print(fileName)
          assets = "Assets\\"
          if os.path.isfile(fileName):
              os.rename(fileName,assets+Name+'.jpg')
        if item in info:
          del info[item]

        info[Name] = [Stock, Price, Status]
        f = open("details.txt","wb")
        pickle.dump(info,f)
        f.close()
```

```
        edit_win.destroy()
        search()
    G.button('OK',430,210,20,40,col='black',bg_col='white', command = update_inv, canvas =
edit_canvas)

def delete_item(name):
    global n
    f = open("details.txt","rb")
    info = pickle.load(f)
    f.close()
    del info[name]
    os.remove('Assets//'+name+'.jpg')
    f = open("details.txt","wb")
    pickle.dump(info,f)
    f.close()
    search()
    n-=1
def tabulate(name,data,y, bgcol):
    r = G.canvas.create_rectangle(0,y-20,1920,y+30, fill=bgcol,outline=bgcol)
    t1 = G.text(name, 200, y,col='black')
    t2 = G.text(str(data[0])+' kg',400,y,col='black')
    t3 = G.text('₹'+str(data[1])+' per kg',600,y,col='black')
    t4 = G.text(data[2],900,y,col='black')
    b1 = G.button('Edit', 1100,y-10,25,100 ,col='blue',bg_col=bgcol,command=lambda:
edit(name, data))
    b2 = G.button('🗑',1200,y-10,25,25,col='red', bg_col= bgcol, command=lambda:
delete_item(name ))
    G.scroll(600)

    return (b1, b2)

def view_inv(): #default canvas

    print(f"{'Sno' : <5}{'Item_Name' : ^15}{'Stock' : ^10}{'Price' : ^10}{'Status' : ^25}")
    searchbox = G.inputBox(110,5,72,20,bg_col='white',col='black',j='left')
    button_list= []
    global search

    def search():
        f = open("details.txt","rb")
        info = pickle.load(f)
        search_input = searchbox.get().lower()
        y = 120


        #print(n)
        for i in button_list:
            for j in i:
                j.destroy()
        global n
```

```
        G.canvas.create_rectangle(0,y-20,1280,y+50*n,fill='white',outline='white')


        col = '#c7c7c7'
        for k,v in info.items():

            if search_input in k.lower() or search_input==":
                buttons = tabulate(k,v,y,col)
                button_list.append(buttons)

                if col == '#c7c7c7':

                    col = 'white'
                else:
                    col='#c7c7c7'
                y+=50
                print(search_input)
        print(info)
        f.close()
    G.button('⟳',5,5,30,30,command=lambda: edit('Apple',data))
    G.button('Add+',45,5,30,50,command=add)
    G.button('🔍',1200,5,30,30,command=search)
    G.canvas.create_rectangle(0,45,1280,120, fill='black')
    G.text(f"{'Product' : ^12}{'Stock' : ^24}{'Price' : ^12}{'Status' :
^45}",140,75,anchor=tk.W)
    search()

        #print(f'{list(info).index(k)+1: <5}{k : ^15}{info[k][0] : ^10}{info[k][1] :
^10}{info[k][2] : ^25}')


view_inv()
G.root.mainloop()

import tkinter as tk
from tkinter import ttk
from PIL import ImageTk, Image
root = tk.Tk()
def createCanvas():
    global canvas
    canvas = tk.Canvas(root, width = 1280, height = 600, highlightthickness = 0)
    canvas.place(x = 0, y = 0)
    canvas.photos = []

def c(num):
    return num*(w/1920)
createCanvas()
#functions

def image(file,x,y,rw,rh,canvas=canvas):
```

```
    image = Image.open(file)
    resizeImage = image.resize((rw,rh))
    img = ImageTk.PhotoImage(resizeImage)
    canvas.photos.append(img)
    canvas.create_image(x,y, image= img)


def text(text, x, y, col = 'white', size = 15, font = 'Consolas', anchor =
tk.CENTER,canvas=canvas):
    font =(font, int(size))

    text_ = canvas.create_text(x, y, text = text, fill = col, font = font,   anchor = anchor)
    return text_

def dropDown(text, opt, x, y, col = 'white', bg_col = '#232328',canvas=canvas):
    menu = tk.StringVar()
    menu.set(text)
    drop = tk.OptionMenu(canvas, menu,*opt)
    drop.config(bg = bg_col, fg=col, width = 30)
    drop['highlightthickness']=0
    drop['menu'].config(bg = bg_col, fg=col,activebackground=bg_col)
    drop.place(x=x,y=y)

def button(text, x, y,h,w,command, col = 'white', bg_col =
'#232328',font='Consolas',canvas=canvas):
        pixel = tk.PhotoImage(width=1, height=1)
        canvas.photos.append(pixel)
        font =(font, int(15))
        h,w = h,w

        button = tk.Button(canvas, text = text,command =
command,image=pixel,compound='c',relief='flat')
        button.config(background = bg_col, fg = col, font=font,height=h,width=w)
        buttonWindow = canvas.create_window(x, y,  window=button,anchor=tk.NW)
        return  button

def imgButton(buttonImage,x,y,command,canvas=canvas):
        button = tk.Button(canvas, image=buttonImage,command = command)
        button['background'] = '#0e1342'
        buttonWindow = canvas.create_window(x*(w/1920), y*(w/1920), anchor=tk.NW,
window=button)

def inputBox(x, y, w,s,col = 'white', bg_col = '#232328', val='',font='Consolas', j =
'center',canvas=canvas):
    font = (font,int(s))
    entry = tk.Entry(canvas, borderwidth=2,bg = bg_col, fg = col, justify
=j,width=int(w),font=font,relief='sunken')
    entry.insert(0,str(val))
    entryWindow = canvas.create_window(x, y,  window=entry,anchor=tk.NW)
    return entry
```

```
class numEntry(tk.Entry):
    def __init__(self, master=None, **kwargs):
        self.var = tk.StringVar()
        tk.Entry.__init__(self, master, textvariable=self.var, **kwargs)
        self.old_value = ''
        self.var.trace('w', self.check)
        self.get, self.set = self.var.get, self.var.set

    def check(self, *args):
        if self.get().isdigit():
            # the current value is only digits; allow this
            self.old_value = self.get()
        else:
            # there's non-digit characters in the input; reject this
            self.set(self.old_value)

def notif(text,canvas=canvas):
    nlabel = tk.Label(text=text)
    nlabel.configure(font=('Consolas', 18),bg='#000000',fg='white')
    notifWindow = canvas.create_window(cenx, 865, window= nlabel, anchor= tk.CENTER)
    root.after(3000, lambda: nlabel.destroy())

def scroll(span,canvas=canvas):
    bar=ttk.Scrollbar(root,command=canvas.yview, orient=tk.VERTICAL)
    bar.place(relx=0.99, height = span)
    canvas.configure(yscrollcommand=bar.set)
    canvas.configure(yscrollcommand=bar.set)
    canvas.configure(scrollregion=canvas.bbox("all"))

#importing libraries
import tkinter as tk
from tkinter import ttk

from PIL import ImageTk, Image
import random
import pickle
import mysql.connector as sql_con

con = sql_con.connect(host="localhost", username="root", passwd="password",
database="Bill")
cur=con.cursor()
SQL = "TRUNCATE TABLE BillInfo;"
cur.execute(SQL)
con.commit()
#creating window

root = tk.Tk()
h = root.winfo_screenheight()
w = root.winfo_screenwidth()
```

```
root.geometry(str(w)+'x'+str(h))
root.title('')
root.grid_anchor('center')
root.resizable(False,False)
cenx=w/2
ceny=h/2
style = ttk.Style()
style.theme_use('default')
style.configure('TButton', background='#db500b')
money=0
bh,bw=33,33
print(root.winfo_screenwidth())

#canvas
def createCanvas():
    global canvas
    canvas = tk.Canvas(root, width = w, height = h, highlightthickness = 0)
    canvas.place(x = 0, y = 0)
    canvas['background']= '#f0e3c0'
    canvas.create_rectangle(0,0,1920,220, fill='#0e1342')
    canvas.photos = []

createCanvas()

def c(num):
    return num*(w/1920)

#functions

def image(file,x,y,rw,rh):
    image = Image.open(file)
    resizeImage = image.resize((rw,rh))
    img = ImageTk.PhotoImage(resizeImage)
    canvas.photos.append(img)
    canvas.create_image(x,y, image= img)


def text(text, x, y, col = 'white', size = 15, font = 'Consolas', anchor = tk.CENTER):
    font =(font, int(size))

    text_ = canvas.create_text(x, y, text = text, fill = col, font = font,    anchor = anchor)
    return text_

def dropDown(text, opt, x, y, col = 'white', bg_col = '#232328'):
    menu = tk.StringVar()
    menu.set(text)
    drop = tk.OptionMenu(canvas, menu,*opt)
    drop.config(bg = bg_col, fg=col, width = 30)
    drop['highlightthickness']=0
```

```
    drop['menu'].config(bg = bg_col, fg=col,activebackground=bg_col)
    drop.place(x=x,y=y)

def button(text, x, y,h,w,command, col = 'white', bg_col = '#232328',font='Consolas'):
        pixel = tk.PhotoImage(width=1, height=1)
        canvas.photos.append(pixel)
        font =(font, int(15))
        h,w = h,w

        button = tk.Button(canvas, text = text,command = command,image=pixel,compound='c',
relief ='ridge')
        button.config(background = bg_col, fg = col, font=font,height=h,width=w)
        buttonWindow = canvas.create_window(x, y,  window=button,anchor=tk.NW)


def imgButton(buttonImage,x,y,command):
        button = tk.Button(canvas, image=buttonImage,command = command, relief = 'flat')
        button['background'] = '#0e1342'
        buttonWindow = canvas.create_window(x*(w/1920), y*(w/1920), anchor=tk.NW,
window=button)

def inputBox(x, y, w,s,col = 'white', bg_col = '#232328', val='',font='Consolas', j = 'center'):
    font = (font,int(s))
    entry = tk.Entry( borderwidth=2,bg = bg_col, fg = col, justify
=j,width=int(w),font=font,relief='flat')
    entry.insert(0,str(val))
    entryWindow = canvas.create_window(x, y,  window=entry,anchor=tk.NW)
    return entry

class numEntry(tk.Entry):
    def __init__(self, master=None, **kwargs):
        self.var = tk.StringVar()
        tk.Entry.__init__(self, master, textvariable=self.var,relief='flat', **kwargs)
        self.old_value = ''
        self.var.trace('w', self.check)
        self.get, self.set = self.var.get, self.var.set

    def check(self, *args):
        if self.get().isdigit():
            # the current value is only digits; allow this
            self.old_value = self.get()
        else:
            # there's non-digit characters in the input; reject this
            self.set(self.old_value)

def notif(text):
    nlabel = tk.Label(text=text)
    nlabel.configure(font=('Consolas', 18),bg='#000000',fg='white')
    notifWindow = canvas.create_window(cenx, 865, window= nlabel, anchor= tk.CENTER)
    root.after(3000, lambda: nlabel.destroy())
```

```
def scroll():
    bar=ttk.Scrollbar(root,command=canvas.yview, orient=tk.VERTICAL)
    bar.place(relx=0.99, height = 1080)
    canvas.configure(yscrollcommand=bar.set)
    canvas.configure(yscrollcommand=bar.set)
    canvas.configure(scrollregion=canvas.bbox("all"))
#logic
f = open('details.txt','rb')
info = pickle.load(f)
menu = {"Apples":[random.randint(50,150),150.00], "Banana":[random.randint(50,150),
40.00], "Mango": [random.randint(50,150),160.00], "Tomato": [random.randint(50,150),25],
"Watermelon": [random.randint(50,150),45.00], "Guava":[random.randint(50,150),45.00],
"Potato": [random.randint(50,150),20.00],  "Onion":[random.randint(50,150),23.00
],"Broccoli":[random.randint(50,150),55.00], "Spinach":[random.randint(50,150),45.00],
"Cauliflower":[random.randint(50,150),25.00], "Okra":[random.randint(50,150),50.00] }
user_selections = {}
def addCart(item, quantity):
    if quantity.isdigit() and int(quantity) > 0:
        if int(quantity)<= info[item][0]:

            user_selections[item] = int(quantity)
            notif(str(quantity) + ' kg of ' + str(item)+' added to the cart.')
        else:
            notif("Sorry, the requested amount exceeds the stock, the available stock is " +
str(menu[item][0])+" kg.")

    else:
        notif("Please enter a valid quantity!")


def removeCart(item):
    user_selections.pop(item)
    notif(item+ 'removed from cart.')
    cartScreen()
def calc():
    total_price = 0
    for item,quantity in user_selections.items():

                price = info[item][1] * quantity
                total_price += price
    notif('The total price is: ₹'+str(total_price))
    return (str(total_price))
def add(entryWidget):
    n = entryWidget.get()
    entryWidget.delete(0,len(n))
    entryWidget.insert(0,str(int(n)+1))
def sub(entryWidget):
    n = entryWidget.get()
    entryWidget.delete(0,len(n))
```

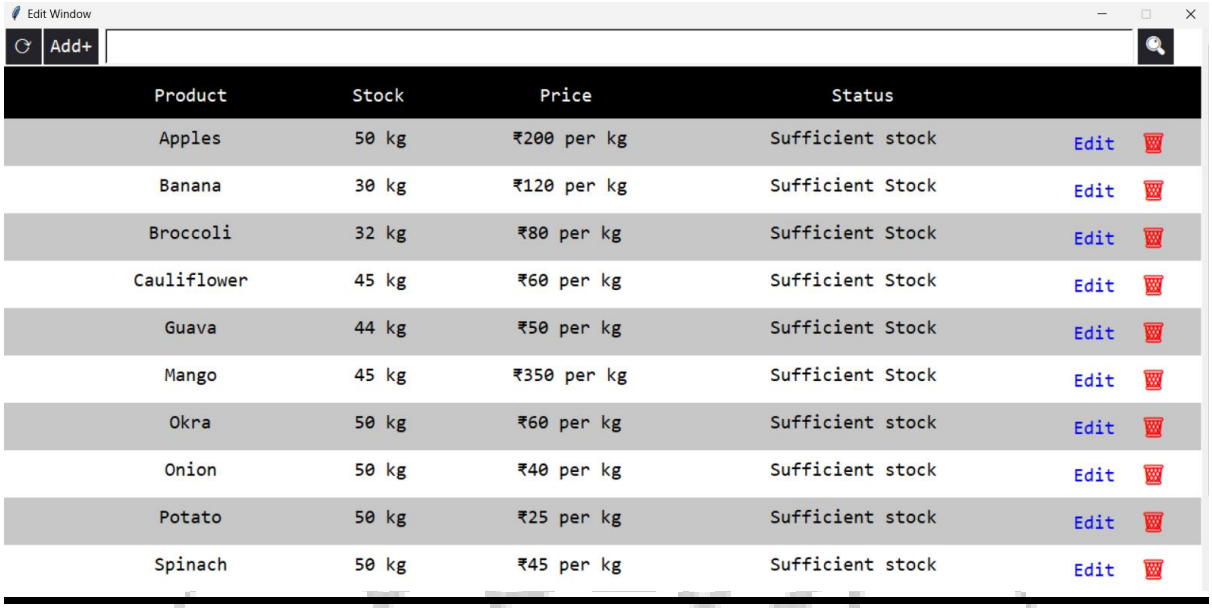
```
    entryWidget.insert(0,str(int(n)-1))
#components
def itemBox(boxText,x,y):
    canvas.create_rectangle(x,y,x+315,y+385,outline='#a1a1a1',width='3')
    text(boxText,x+157.5,y+288.75,size=30,col ="#1e1433")
    text('₹'+str(int(info[boxText][1]))+' per kg',x+157.5,y+315,size=10,col ="#1e1433")
    text('Available stock: '+str(int(info[boxText][0]))+' kg',x+157.5,y+330,size=10,col
="#1e1433")
    fileName = 'Assets\\'+boxText+'.jpg'
    image(fileName,x+157.5,y+136.5,306,262)
    canvas.create_rectangle(x+5.25,y+264.25,x+313.25,y+267.5,outline='#4a4a4a',fill='#4a4a
4a')
    boxQuantity = inputBox(x+52.5,y+341.25,2,24,val=0)
    button('+',x+14,y+341.25,bh,bw,lambda: add(boxQuantity), font='Consolas 16
bold',bg_col="#019404")
    button('-',x+91,y+341.25,bh,bw,lambda: sub(boxQuantity), font='Consolas 16
bold',bg_col="#db000b")
    button('Add to Cart',x+185,y+341,bh,115,lambda:
addCart(boxText,boxQuantity.get()),bg_col="#3d1ab0",font = 'Consolas 11 bold')

def cartBox(item,y):
    fileName = 'Assets\\'+item+'.jpg'
    price = int(info[item][1]* user_selections[item])
    image(fileName,c(100)+96.5,y+85.5,193,169)
    canvas.create_rectangle(c(100),y,w-c(100),y+172.5,outline='#a1a1a1',width='4')
    text(item, 310, y+5, size=30, anchor=tk.NW,col='black')
    text('Quantity:', 310,y+80,size=14,anchor=tk.NW,col='black')
    price = text('₹'+str(price),1800,y+20,size=25,anchor=tk.NE,col='black')
    boxQuantity = inputBox(359,y+115,2,24,val=user_selections[item])
    def addr():
        global money, total_price
        add(boxQuantity)
        if boxQuantity.get().isdigit() and int(boxQuantity.get()) > 0:
            user_selections[item] = int(boxQuantity.get())
            notif('1 more kg of ' + str(item)+' added to the cart.')
        else:
            notif("Please enter a valid quantity!")
        canvas.itemconfig(price, text = '₹'+str(int(info[item][1]* user_selections[item])))
        canvas.itemconfig(money, text='Total price: ₹'+calc())
        cartScreen()
    def subr():
        sub(boxQuantity)
        if boxQuantity.get()=="0":
            removeCart(item)
            return
        elif boxQuantity.get().isdigit() and int(boxQuantity.get()) > 0:
            user_selections[item] = int(boxQuantity.get())
            notif('1 kg of ' + str(item)+' removed from the cart.')
        else:
            notif("Please enter a valid quantity!")
```

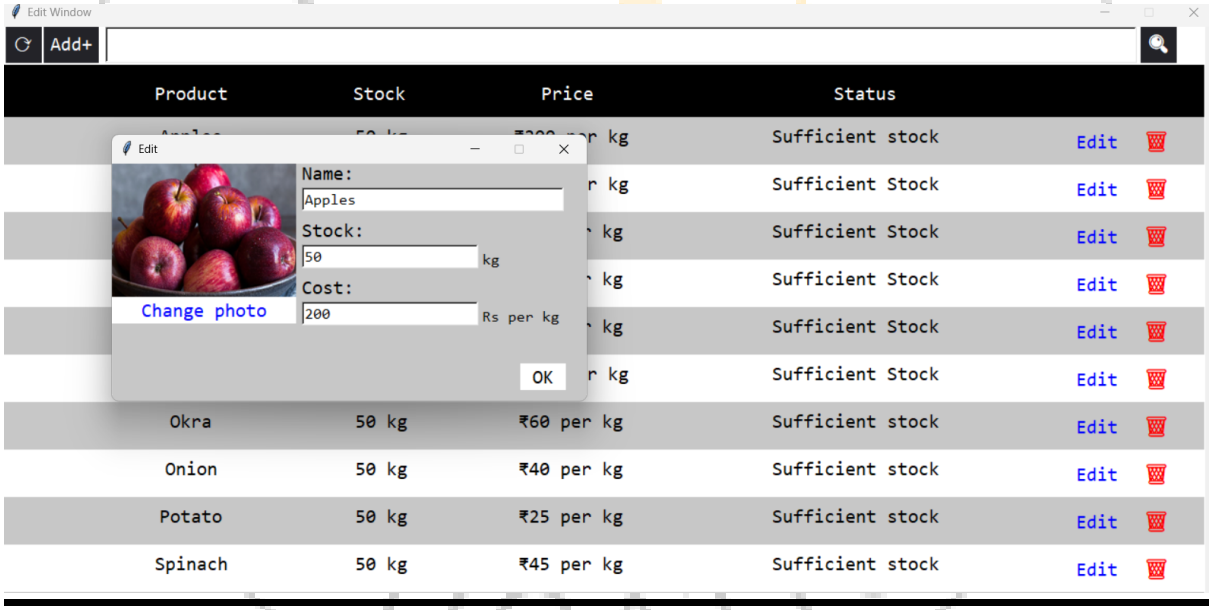
```
        canvas.itemconfig(price, text = '₹'+str(int(info[item][1]* user_selections[item])))
        canvas.itemconfig(money, text='Total price: ₹'+calc())
        cartScreen()
    button('+',316,y+115,35,35,addr,bg_col='#0e1342', font='Consolas 16 bold')
    button('-',400,y+115,35,35,subr,bg_col='#0e1342', font='Consolas 16 bold')
    button('Remove', 1650,y+115,35,130, lambda: removeCart(item),bg_col='red',
font='Consolas 7 bold',col='white')

#pages

def homeScreen():
    createCanvas()
    text("Groceries", cenx, 55, '#dbbe7b', 75, font='Lucida Handwriting')
    text("Fresh fruits and veggies at your doorstep!", cenx, 140, size = 25,font='Lucida Sans')
    cartImage = Image.open('Assets\Cart.png')
    resizeCartImage = cartImage.resize((100,100))
    img = ImageTk.PhotoImage(resizeCartImage)
    canvas.photos.append(img)
    imgButton(img,1750,50,cartScreen)
    #boxes per row
    bpr = root.winfo_screenwidth()//315
    boxx= -10+(w - 315*bpr)/2
    boxy=230
    count = 0
    boxes = len(info)

    #for key,values in info.items():
    for key in info:
        itemBox(key,boxx,boxy)
        boxx+= 315
        count+=1
        boxes -= 1
        if count==bpr:
            if boxes < bpr :
                bpr = boxes
            boxx= -10+(w - 315*bpr)/2
            boxy+=420
            count = 0


    text(',6,boxy+50)
    scroll()
def delivScreen():
    createCanvas()
    x_ = 480*(w/1920)
    x__ = x_ + 600*(w/1920)
    #modifying details.txt, creating table

    text("Delivery", cenx, 100, '#dbbe7b', 75, font='Lucida Handwriting')
```

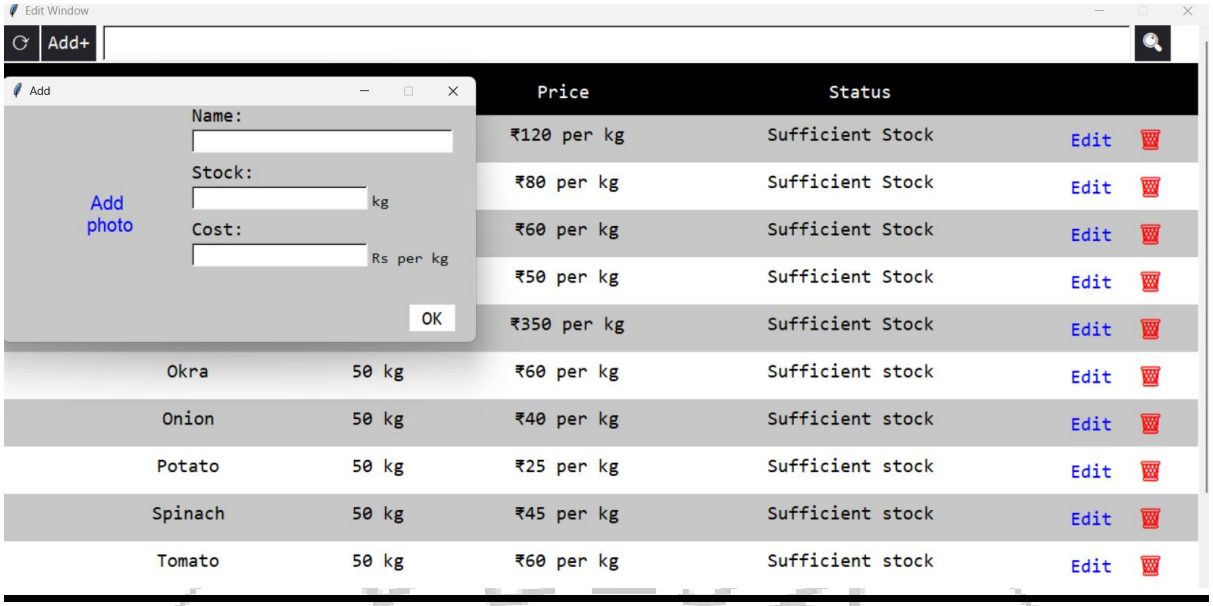
```
    m=calc()
    text("First Name: ",x_,330 , 'black',anchor=tk.W)
    firstname = inputBox(x_, 350,25,20, col ="Black", bg_col= 'White', j='left')
    text("Last Name: ",x_,430 , 'black',anchor=tk.W)
    lastname = inputBox(x_, 450,25,20, col ="Black", bg_col= 'White', j='left')
    text("Phone: ",x_,530,'black',anchor=tk.W)
    phone = numEntry( borderwidth=2,bg = 'white', fg = 'black',width=25,font='Consolas 20')
    phoneWindow = canvas.create_window(x_,550,  window=phone,anchor=tk.NW)
    text("Email: ",x_,630,'black',anchor=tk.W)
    email = inputBox(x_, 650,25,20, col ="Black", bg_col= 'White', j='left')

    text('Home Address:', x__, 330, col = "Black",anchor=tk.W)
    address  = inputBox(x__, 350,25,20, col ="Black", bg_col= 'White', j='left')
    text('City:', x__, 430, col = "Black",anchor=tk.W)
    address  = inputBox(x__, 450,25,20, col ="Black", bg_col= 'White', j='left')
    text('State/UT:', x__, 530, col = "Black",anchor=tk.W)
    states =["Andhra Pradesh","Arunachal Pradesh
","Assam","Bihar","Chhattisgarh","Goa","Gujarat","Haryana","Himachal Pradesh","Jammu
and Kashmir","Jharkhand","Karnataka","Kerala","Madhya
Pradesh","Maharashtra","Manipur","Meghalaya","Mizoram","Nagaland","Odisha","Punjab",
"Rajasthan","Sikkim","Tamil Nadu","Telangana","Tripura","Uttar
Pradesh","Uttarakhand","West Bengal","Andaman and Nicobar
Islands","Chandigarh","Dadra and Nagar Haveli","Daman and
Diu","Lakshadweep","National Capital Territory of Delhi","Puducherry"]
    dropDown('State/UT', states, x__,550, col="Black", bg_col="White")
    text("ZIP Code: ",x__,630,'black',anchor=tk.W)
    zip_ = numEntry( borderwidth=2,bg = 'white', fg = 'black',width=25,font='Consolas 20')
    zipWindow = canvas.create_window(x__,650,  window=zip_,anchor=tk.NW)

    def generateBill():
        f = open('details.txt','rb')
        info = pickle.load(f)
        f.close()
        srn = 1
        for name, qty in user_selections.items():
            price = info[name][1]
            t_price = price*qty
            SQL = "INSERT INTO BillInfo VALUES( {} , '{}' , {} , {} , {} );".format( str(srn),
name, str(qty), str(price), str(t_price) )
            cur.execute(SQL)
            con.commit()
            srn += 1
            info[name][0] = info[name][0] - qty
            if info[name][0]<20:
                info[name][2] = 'Low Stock, Resupply'
            else:
                info[name][2] = 'Sufficient Stock'
            f = open('details.txt','wb')
            pickle.dump(info,f)
            f.close()
```

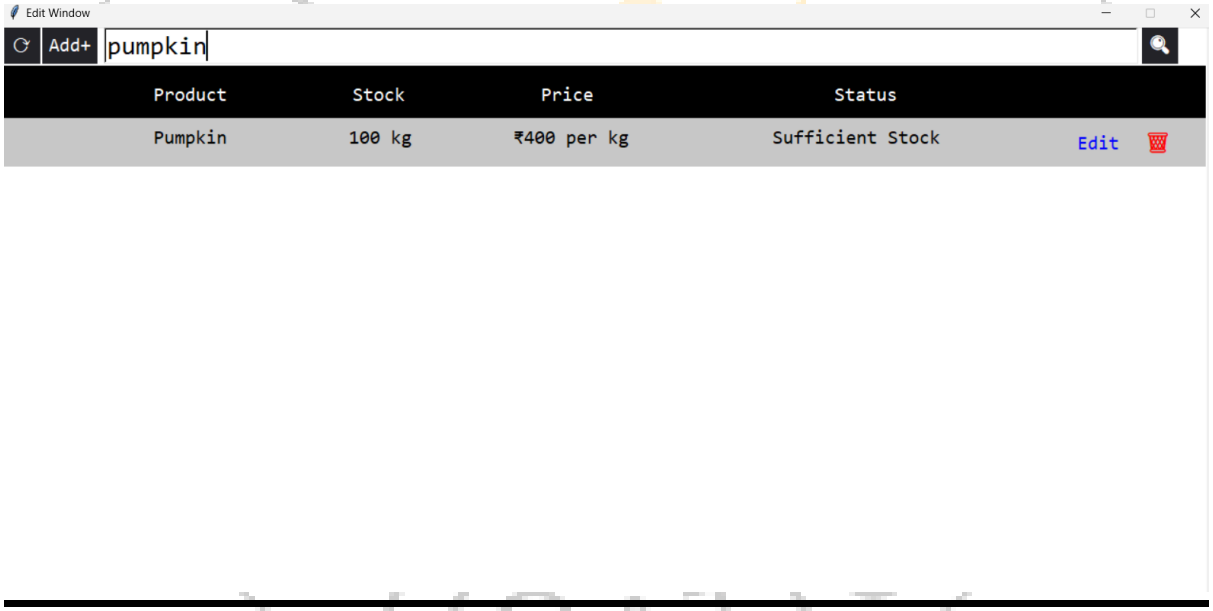
```
        billScreen()
    confirmPurchase = button("Confirm Purchase", (w/1920)*840, 750, 30, 300,
generateBill,col ="Black", bg_col= 'White')
def cartScreen():

    createCanvas()

    text("Your Cart", cenx, 55, '#dbbe7b', 75, font='Lucida Handwriting')
    boxy=230
    count = 0



    for key in user_selections:
        cartBox(key,boxy)
        boxy+=230

    text('Total:'+calc(),100,boxy+50,col='black')
    button('Proceed',100,boxy+120,bg_col='cyan',col='white',h=40,w=130,command=delivScr
een)
    button('Back',c(100),50,bg_col='red',col='white',h=40,w=130,command=homeScreen)
    text(",6,boxy+200)
    scroll()
    print(user_selections)

def billScreen():
    createCanvas()
    text("Bill", cenx, 55, '#dbbe7b', 75, font='Lucida Handwriting')
    y = 500
    SQL = "SELECT * FROM BillInfo"
    cur.execute (SQL)
    table = cur.fetchall()

    text(f"{'Sno' : <5}{'Item_Name' : ^15}{'Quantity' : ^10}{'Price' : ^10}{'Total Price' :
^15}",cenx, 400,size=18,col='Black',font='Monospace')
    for sno,name,qty,price,t_price in table :  #yaha par apne hisaab se print krwa lena
        text(f"{sno : <5}{name : ^15}{qty : ^10}{price : ^10}{t_price : ^15}",cenx,
y,col='Black',font='Monospace',size=18)
        y+=50
    canvas.create_rectangle(cenx-400,350,cenx+400,y)
    text(",6,y+200)
    scroll()
homeScreen()
root.mainloop()
```

INPUT/OUTPUT SCREEN FORMATS


Edit Window
Add+
Product
Stock
Price
Status
Apples 50 kg ₹200 per kg Sufficient stock Edit
Banana 30 kg ₹120 per kg Sufficient Stock Edit
Broccoli 32 kg ₹80 per kg Sufficient Stock Edit
Cauliflower 45 kg ₹60 per kg Sufficient Stock Edit
Guava 44 kg ₹50 per kg Sufficient Stock Edit
Mango 45 kg ₹350 per kg Sufficient Stock Edit
Okra 50 kg ₹60 per kg Sufficient stock Edit
Onion 50 kg ₹40 per kg Sufficient stock Edit
Potato 50 kg ₹25 per kg Sufficient stock Edit
Spinach 50 kg ₹45 per kg Sufficient stock Edit

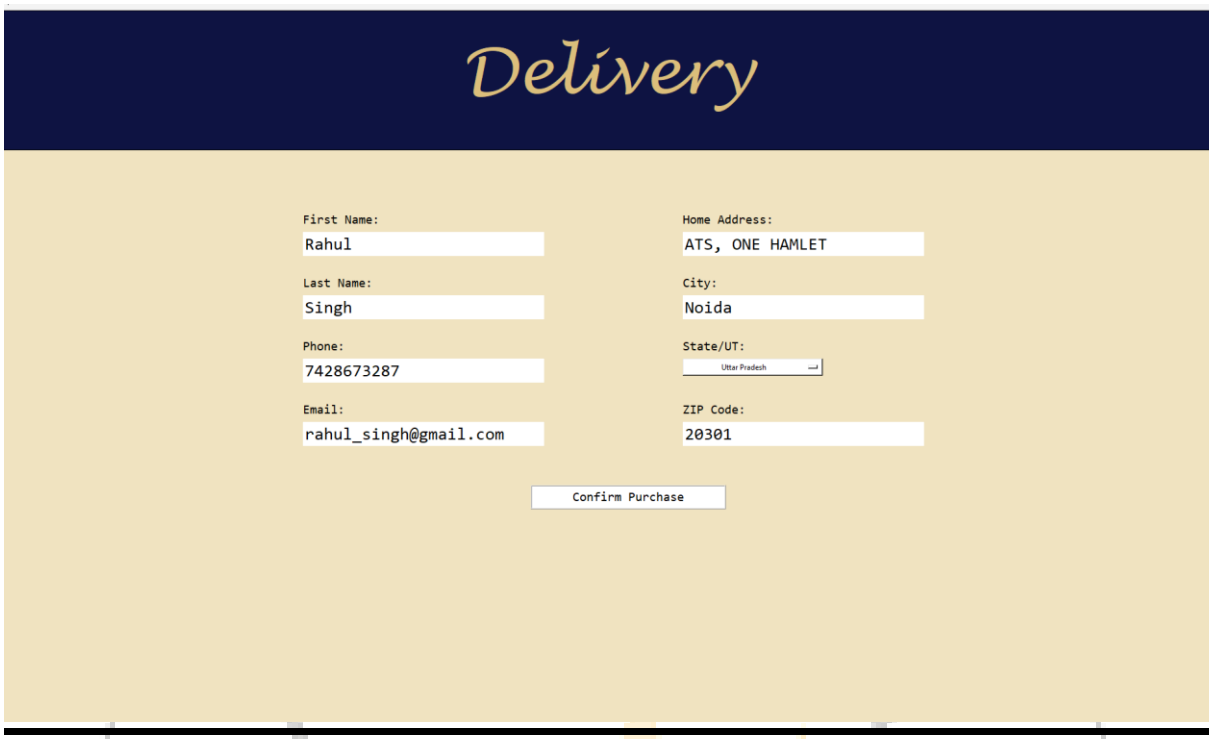

Edit Window
Add+
Product
Stock
Price
Status
Edit
Name:
Apples
Stock:
50
kg
Cost:
200
Rs per kg
Change photo
OK
Okra 50 kg ₹60 per kg Sufficient stock Edit
Onion 50 kg ₹40 per kg Sufficient stock Edit
Potato 50 kg ₹25 per kg Sufficient stock Edit
Spinach 50 kg ₹45 per kg Sufficient stock Edit

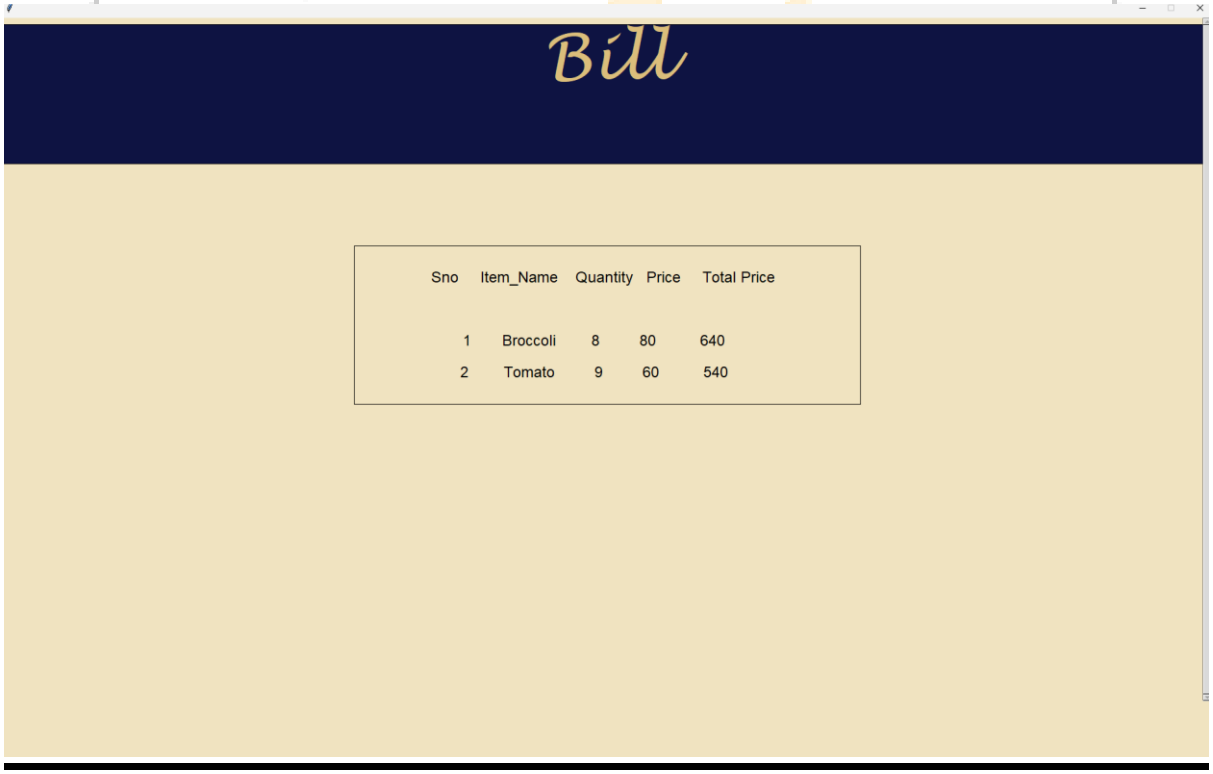

Edit Window
Add+
Add
Name:
Stock:
kg
Cost:
Rs per kg
Add photo
OK
Price
Status
₹120 per kg
Sufficient Stock
Edit
₹80 per kg
Sufficient Stock
Edit
₹60 per kg
Sufficient Stock
Edit
₹50 per kg
Sufficient Stock
Edit
₹350 per kg
Sufficient Stock
Edit
Okra 50 kg ₹60 per kg Sufficient stock Edit
Onion 50 kg ₹40 per kg Sufficient stock Edit
Potato 50 kg ₹25 per kg Sufficient stock Edit
Spinach 50 kg ₹45 per kg Sufficient stock Edit
Tomato 50 kg ₹60 per kg Sufficient stock Edit


Edit Window
Add+
pumpkin
Product
Stock
Price
Status
Pumpkin 100 kg ₹400 per kg Sufficient Stock Edit


Groceries
Fresh fruits and veggies at your doorstep!
Banana
Broccoli
Cauliflower
Guava
Mango
Okra
Onion
Potato
Spinach
Tomato
Watermelon
Apples
Add to Cart


Back
Your Cart
Broccoli
Quantity:
8
Remove
₹640
Tomato
Quantity:
9
Remove
₹540
Total:1180
Proceed
The total price is: ₹1180


Delivery
First Name:
Rahul
Last Name:
Singh
Phone:
7428673287
Email:
rahul_singh@gmail.com
Home Address:
ATS, ONE HAMLET
City:
Noida
State/UT:
Uttar Pradesh
ZIP Code:
20301
Confirm Purchase


Bill
Sno Item_Name Quantity Price Total Price
1 Broccoli 8 80 640
2 Tomato 9 60 540

SCOPE FOR IMPROVEMENT

While the proposed system aims to cover all essential aspects of grocery store management, there is always room for further enhancements. Future improvements could include:

- **Integration with Online Platforms:** Allowing customers to place orders online and pick them up in-store.
- **Mobile Application:** A mobile app version of the system for on-the-go management.
- **Advanced Analytics:** Incorporating AI to predict customer buying patterns and optimize stock levels.
- **Multi-Store Support:** Expanding the system to manage multiple stores from a single interface.

BIBLOGPRAHY / REFRENCES

1. **Websites**:
    - https://stackoverflow.com/
    - https://www.reddit.com/r/learnpython/
    - https://www.geeksforgeeks.org/
    - https://www.tutorialspoint.com/
2. **Videos**:
    - https://www.youtube.com/watch?v=tJxcKyFMTGo&list=PLaL2yxczKLcARpDfF_5JO5Eydb0qwVOBB
    - https://www.youtube.com/watch?v=YXPyB4XeYLA
3. **Books**:
    - Class 12 Comp. Science NCERT
    - Sumita Arora

THANK YOU

-Panav
-Ayaan