

**2020-2021**

**ACCOUNTING SOFTWARE**

*SUBMITTED TO:- SUBMITTED BY:-*

**Mr. Rajneesh Sharma Ranvir Singh Chhabra**

**Safiullah Saif**

**CS Project**

**Index**

1. Certificate
2. Acknowledgement
3. Introduction
4. System Analysis
5. System Requirements
6. Data Dictionary
7. Advantages and Limitations
8. Future Enhancement
9. Input/Output
10. Conclusion

**CERTIFICATE**

This is to certify that this project report entitled

**“Accounting Software”**

has been prepared by

**Ranvir Singh Chhabra and Safiullah Saif**

Students of Class XII. The system has been approved by the Department of Computer science, **Daly College, Indore** (M.P.) and the work has been done under my guidance. This work is up to the mark of satisfaction. We wish them success in every aspect of life. They have performed this project on their own. They have also put in sufficient periods for completion. This project has been completed as per rules of the Central Board of Secondary Examination Syllabus and can be considered as the fulfillment of the A.I.S.S.C.E. Examination.

**Date:**

**Mr. Rajesh Nandwal**

**Head, Department of Computer Science,**

**Daly College, Indore**

**Indore**

**Acknowledgement**

We feel immense pleasure and deep feeling of gratitude towards ***Mr Rajesh Nandwal*** **(H.O.D |Comp. Sc.)** of **Daly College, Indore** for his skillful guidance, constructive and valuable suggestions and encouraging cooperation for my project which not merely helped but enabled me to give effort towards this project.

We are also thankful to you ***Mr. Neeraj Kumar Bedhotiya*, Principal, Daly College, Indore** for his encouragement and valuable suggestions given to us from time to time.We also extend our thanks to all our colleagues and friends for their valuable assistance and kind cooperation during the course of this investigation.

**Date:**

**Place: Daly College, Indore**

**Safiullah Saif and Ranvir Singh Chhabra**

**INTRODUCTION**

This project is based on the idea of accounting in a general shop/market. It has been made to suit the user as it is very simple and user friendly.

It has been made attractive by using Python, MySQL, Pycharm, IDLE and Tkinter. **Python** is an [interpreted](https://en.wikipedia.org/wiki/Interpreted_language), [high-level](https://en.wikipedia.org/wiki/High-level_programming_language) and [general-purpose programming language](https://en.wikipedia.org/wiki/General-purpose_programming_language) used worldwide. Python was created in the late 1980s, and first released in 1991, by [Guido van Rossum](https://en.wikipedia.org/wiki/Guido_van_Rossum) as a successor to the [ABC programming language](https://en.wikipedia.org/wiki/ABC_(programming_language)).It is used for web development, software development, mathematics, system scripting etc. As of December 2020, Python ranked third in [TIOBE](https://en.wikipedia.org/wiki/TIOBE_index)’s index of most popular programming languages, behind [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [Java](https://en.wikipedia.org/wiki/Java_(programming_language)).

Python, with the help of Tkinter made making of such a complicated and dynamic application, which is easy to work upon.

**System Analysis**

The following project is designed to computerize the existing manual system.

Also if records are maintained on paper, it will be difficult to store records relatively for a longer duration of time.

In this project, we have prepared a manual input form of various products purchased and store them in the form of a table and then accordingly generate an invoice in the form of a pdf which can be printed.

The list of items available in the store are stored in a database. This database can be accessed through different programs from the main menu which can be used to perform tasks such as viewing, addition and deletion of data.

The maintained table of items consists of the following information:

* Product Code
* Product Category
* Product Name
* Product Price

At the end, a bill is generated in the form of a pdf consisting of the following:

* List of products purchased along with their code, name, quantity, price per unit
* Invoice Number, date of issue, user and seller information
* Total amount payable

The bill also gives an option of mode of payment either through cash ,card or digital payment.

**System Details**

These are the specifications of the system on which this program was built.

***Software***

* Microsoft Windows 10
* Python 3.0
* PyCharm IDLE
* MySQL
* MYSQL Workbench 8.0 CE
* Tkinter
* FPDF plugin For PyCharm

***Hardware***

* Min. processor clock speed: 1.3 GHz
* Min. 2 GB RAM
* Min. 10 GB HDD

**Advantages and Limitations**

There are various advantages of this program which are:

* It performs various functions instead of only storing the records.
* It shows the list of all the items present in the store.
* It automatically performs all the calculations such as multiplication of item quantity and price per unit as well as calculation of grand total.
* It gives the option of deletion of an item from the database through the program itself.
* It also ensures security as only registered employees are allowed to access the program with the help of username and password.
* It provides a list of items to the user, before the invoice is generated.
* It also provides an option to create a new account in case a new employee is hired.
* It automatically generates a pdf on the basis of the items purchased by the customer.
* In case a new product is launched, the user can append the details of this product into the database from the program itself.
* In case the sales of a product are discontinued, the user can delete this product from the database from the program itself.

The limitations are:

* It does not give an option to provide a discount as per category.
* It does not provide the facility of the addition of taxes without changing it in the database.

**Future Enhancement**

There are various features which can be added in the program. For example:

* Discount on items
* More categories of items
* It can use graphics which makes it very attractive.
* Addition of points to the user’s account.

There are various other things which can be upgraded in this program:

* A Larger database
* More categories of items
* Keeping a record of the expiry date of items by creating a seperate table

**Input**

**# Start.py**

from tkinter import \*

import tkinter.font as font

import os

from tkinter import messagebox

import pickle

top = Tk()

top.geometry("400x275")

top.title("User Verification")

top.configure(bg="deep sky blue")

myfont = font.Font(family="Times", size=30, weight=font.BOLD)

myfont1 = font.Font(family="Times", size=15)

name\_var = StringVar()

passw\_var = StringVar()

def fun\_login():

file1 = open('updation.txt', 'rb+')

n = 0

name = name\_entry.get()

passw = passw\_entry.get()

if name == '' or passw == '':

messagebox.showerror('Error', 'Please complete the required fields')

else:

file1.seek(0)

try:

while True:

s = pickle.load(file1)

if name in s and s[name] == passw:

n = 1

except EOFError:

file1.close()

if n == 1:

top.destroy()

os.system('main1.py')

else:

messagebox.showerror("Error", "Incorrect username or password")

name\_var.set("")

passw\_var.set("")

def fun\_new():

top.destroy()

os.system('new.py')

desc\_label = Label(top, text="Login", font=myfont, bg="deep sky blue")

name\_label = Label(top, text="Name:", font=myfont1, bg="deep sky blue")

name\_entry = Entry(top, textvariable=name\_var, font=myfont1)

passw\_label = Label(top, text="Password:", font=myfont1, bg="deep sky blue")

passw\_entry = Entry(top, textvariable=passw\_var, show='\*', font=myfont1)

sub\_btn = Button(top, text="Submit", command=fun\_login, font=myfont1)

new\_btn = Button(top, text="Create new ID", command=fun\_new, font=myfont1)

desc\_label.grid(row=0, column=0, pady=10, padx=30)

name\_label.grid(row=3, column=0)

name\_entry.grid(row=3, column=1, pady=10)

passw\_label.grid(row=4, column=0)

passw\_entry.grid(row=4, column=1)

sub\_btn.grid(row=5, column=1, pady=10)

new\_btn.grid(row=6, column=1, pady=10)

top.mainloop()

**# new.py**

from tkinter import \*

import pickle

from tkinter import messagebox

import os

import tkinter.font as font

top = Tk()

top.geometry("400x250")

top.title("Create Account")

myfont = font.Font(family="Times", size=25, weight=font.BOLD)

name\_var = StringVar()

email\_var = StringVar()

passw\_var = StringVar()

passw2\_var = StringVar()

def fun\_calcualte():

n = 0

name = name\_entry.get()

email = email\_entry.get()

passw = passw\_entry.get()

passw2 = passw2\_entry.get()

if name == '' or email == '' or passw == '' or passw2 == '':

messagebox.showerror('Error', 'Please complete the required fields')

else:

if passw != passw2:

passw\_var.set("")

passw2\_var.set("")

messagebox.showerror('Error', 'Passwords do not match')

else:

try:

file1 = open('updation.txt', 'ab+')

file1.seek(0)

while True:

s = pickle.load(file1)

if name in s:

n = 1

except EOFError:

file1.close()

file1 = open('updation.txt', 'ab+')

if n == 1:

messagebox.showerror('Error', 'Account already exists')

file1.seek(0)

else:

dict1 = {name: passw}

pickle.dump(dict1, file1)

file1.flush()

top.destroy()

os.system('start.py')

name\_var.set("")

email\_var.set("")

passw\_var.set("")

passw2\_var.set("")

name\_label = Label(top, text="Name:")

name\_entry = Entry(top, textvariable=name\_var)

email\_label = Label(top, text="Email:")

email\_entry = Entry(top, textvariable=email\_var)

passw\_label = Label(top, text="Select Password:")

passw\_entry = Entry(top, textvariable=passw\_var, show='\*')

passw2\_label = Label(top, text="Confirm Password:")

passw2\_entry = Entry(top, textvariable=passw2\_var, show='\*')

top\_label = Label(top, text=" Create Account", font=myfont)

sub\_btn = Button(top, text="Submit", command=fun\_calcualte, activebackground="pink", activeforeground="blue")

top\_label.grid(row=0, column=0, pady=10)

name\_label.grid(row=1, column=0, pady=5)

name\_entry.grid(row=1, column=1, pady=5)

email\_label.grid(row=2, column=0, pady=5)

email\_entry.grid(row=2, column=1, pady=5)

passw\_label.grid(row=3, column=0, pady=5)

passw\_entry.grid(row=3, column=1, pady=5)

passw2\_label.grid(row=4, column=0, pady=5)

passw2\_entry.grid(row=4, column=1, pady=5)

sub\_btn.grid(row=5, column=1, pady=5)

top.mainloop()

**# main.py**

from tkinter import \*

import tkinter.font as font

import os

top = Tk()

myfont = font.Font(family="Times", size=35, weight=font.BOLD)

myfont1 = font.Font(family="Times", size=25, weight=font.BOLD)

myfont2 = font.Font(family="Times", size=15, weight=font.BOLD)

top.geometry("575x350")

top.configure(bg="deep sky blue")

top.title("Main Menu")

def fun\_items():

top.destroy()

os.system('view.py')

def fun\_bill():

top.destroy()

os.system('bill2.py')

def fun\_add\_items():

top.destroy()

os.system('items.py')

def fun\_delete\_items():

top.destroy()

os.system('deletion.py')

def fun\_exit():

top.destroy()

items = Button(top, text="View Items ", command=fun\_items, activeforeground="dark blue", activebackground="light blue",

pady=3, padx=30, relief=RAISED, font=myfont1, width=10, height=1, bg="darkolivegreen1")

bill = Button(top, text="Bill ", command=fun\_bill, activeforeground="dark blue", activebackground="light blue",

pady=3, padx=30, relief=RAISED, font=myfont1, width=10, height=1, bg="darkolivegreen1")

add\_items = Button(top, text="New Items ", command=fun\_add\_items, activeforeground="dark blue",

activebackground="light blue", pady=3, padx=30, relief=RAISED, font=myfont1,

width=10, height=1, bg="darkolivegreen1")

delete\_items = Button(top, text="Delete Items ", command=fun\_delete\_items, activeforeground="dark blue",

activebackground="light blue", pady=3, padx=30, relief=RAISED,

font=myfont1, width=10, height=1, bg="darkolivegreen1")

exit = Button(top, text="Exit", command=fun\_exit, activeforeground="dark blue",

activebackground="light blue", pady=3, padx=30, relief=RAISED,

font=myfont2, width=7, height=1, bg="orange red", fg="white")

top\_label = Label(top, text="Main Menu", font=myfont, pady=10, padx=10, bg="white")

top\_label.grid(row=0, column=0, pady=10, columnspan=2)

items.grid(row=2, column=0, padx=10)

bill.grid(row=2, column=1, padx=10)

add\_items.grid(row=3, column=0, pady=10)

delete\_items.grid(row=3, column=1, padx=10)

exit.grid(row=4, column=0, padx=10, columnspan=2, pady=20)

top.mainloop()

**# items.py**

from tkinter import \*

import mysql.connector as sqltor

from tkinter import messagebox

from tkinter import ttk

import tkinter.font as font

import os

window = Tk()

window.title("Enter New Items")

window.configure(bg="goldenrod1")

window.minsize(400, 290)

myfont = font.Font(family="Times", size=25, weight=font.BOLD)

con = sqltor.connect(

host='localhost',

user='root',

passwd='1234',

database='test'

)

cursor = con.cursor()

def main\_menu():

window.destroy()

os.system('main1.py')

def add\_values():

v1 = ITEMNAME.get()

v2 = ITEMCODE.get()

v3 = n.get()

v4 = ITEMPRICE.get()

if type(float(v4)) == float or type(v4) == int:

if v1 == "" or v2 == "" or v3 == "" or v4 == "":

messagebox.showerror("Error", "Please Complete the Required Fields")

else:

st = "SELECT \* FROM items WHERE productname = '%s'" % (v1,)

cursor.execute(st)

y = cursor.fetchone()

if y is not None:

messagebox.showerror("Error", "Item Already Exisits")

else:

cursor.execute("INSERT INTO items (productcode,productcategory,productname,productprice) "

"VALUES ('%s','%s','%s','%s');" % (v2, v3, v1, v4))

ITEMCODE.set("")

n.set("")

ITEMNAME.set("")

ITEMPRICE.set("")

messagebox.showinfo("Success", "Item Successfully Added!")

con.commit()

else:

messagebox.showerror("Error", "Price should be a Numeric Value")

ITEMNAME = StringVar()

ITEMCODE = StringVar()

ITEMCATEGORY = StringVar()

ITEMPRICE = StringVar()

n = StringVar()

label1 = Label(window, text=" Add Products", font=myfont).grid(row=0, column=0, pady=10)

iname = Label(window, text="Enter Item Name: ").grid(row=1, column=0, pady=5)

e1 = Entry(window, textvariable=ITEMNAME).grid(row=1, column=1, pady=5)

icode = Label(window, text="Enter Item Code: ").grid(row=2, column=0, pady=5)

e2 = Entry(window, textvariable=ITEMCODE).grid(row=2, column=1, pady=5)

icategory = Label(window, text="Select Item Category: ").grid(row=3, column=0, pady=5)

e3 = ttk.Combobox(window, width=18, textvariable=n, state='readonly', values=[' Food', ' Clothing', ' Electronics'

, 'Sports', ' Daily Essentials']).grid(row=3, column=1, pady=5)

iprice = Label(window, text="Enter Item Price: ").grid(row=4, column=0, pady=5)

e4 = Entry(window, textvariable=ITEMPRICE).grid(row=4, column=1, pady=5)

submit = Button(window, text="Add Item", command=add\_values).grid(row=5, column=1, pady=5)

mainmenu = Button(window, text="Return to Main Menu", command=main\_menu).grid(row=6, column=1, pady=5)

window.mainloop()

con.commit()

**# view.py**

from tkinter import \*

import tkinter

import mysql.connector as sqltor

import tkinter.font as font

import os

from tkinter import ttk

window = Tk()

window.title("Items")

window.configure(bg="deep sky blue")

window.minsize(850, 450)

main\_frame = ttk.Frame(window)

main\_frame.pack(fill=BOTH, expand=1)

my\_canvas = tkinter.Canvas(main\_frame)

my\_canvas.pack(side=LEFT, fill=BOTH, expand=1)

my\_scrollbar = ttk.Scrollbar(main\_frame, orient=VERTICAL, command=my\_canvas.yview)

my\_scrollbar.pack(side=RIGHT, fill=Y)

my\_canvas.configure(yscrollcommand=my\_scrollbar.set)

my\_canvas.bind('<Configure>', lambda e: my\_canvas.configure(scrollregion=my\_canvas.bbox("all")))

second\_frame = ttk.Frame(my\_canvas)

my\_canvas.create\_window((0, 0), window=second\_frame, anchor="nw")

myfont = font.Font(family="Times", size=18, weight=font.BOLD)

myfont1 = font.Font(family="Times", size=16)

con = sqltor.connect(

host='localhost',

user='root',

passwd='1234',

database='test')

def main\_menu():

window.destroy()

os.system('main1.py')

cursor = con.cursor()

cursor.execute("SELECT \* FROM items")

i = 1

o1 = Label(second\_frame, text="Product Code", bg='green yellow', width=14, font=myfont)\

.grid(row=0, column=0, padx=2)

o2 = Label(second\_frame, text="Product Category", bg='green yellow', width=14, font=myfont)\

.grid(row=0, column=1, padx=2)

o3 = Label(second\_frame, text="Product Name", bg='green yellow', width=14, font=myfont)\

.grid(row=0, column=2, padx=2)

o4 = Label(second\_frame, text="Product Price", bg='green yellow', width=14, font=myfont)\

.grid(row=0, column=3, padx=2)

for x in cursor:

for j in range(len(x)):

e = Entry(second\_frame, width=18, fg='navy', bg='sky blue', font=myfont1)

e.grid(row=i + 1, column=j)

e.insert(END, x[j])

i = i+1

mainmenu = Button(window, text="Return to Main Menu", command=main\_menu, fg='white', bg='dark blue',

font=myfont).pack(side=BOTTOM, pady=10)

window.mainloop()

**# deletion.py**

from tkinter import messagebox

import tkinter.font as font

window = Tk()

window.title("Delete Items")

window.configure(bg="deep sky blue")

window.minsize(370, 250)

myfont = font.Font(family="Times", size=28, weight=font.BOLD)

myfont1 = font.Font(family="Times", size=15)

con = sqltor.connect(

host='localhost',

user='root',

passwd='1234',

database='test'

)

cursor = con.cursor()

if con.is\_connected():

print('hi')

var = StringVar()

def delete\_fun():

var2 = var.get()

print(var2)

st = "SELECT \* FROM items WHERE productcode = '%s'" % (var2,)

cursor.execute(st)

y = cursor.fetchone()

if y is not None:

st1 = "DELETE FROM items WHERE productcode = '%s'" % (var2,)

cursor.execute(st1)

con.commit()

messagebox.showinfo("Success", "Item Successfully Deleted!")

else:

messagebox.showerror("Error", "Please Enter a Valid Product Code")

label1 = Label(window, text="Delete Items", font=myfont, bg="deep sky blue").grid(row=1, column=1, columnspan=2, pady=15)

label2 = Label(window, text="Enter Item Code: ", font=myfont1, bg="deep sky blue").grid(row=2, column=1)

text1 = Entry(window, textvariable=var, font=myfont1).grid(row=2, column=2, pady=15)

button1 = Button(window, text="Delete an Item", command=delete\_fun, font=myfont1).grid(row=3, column=1, columnspan=2, pady=15)

window.mainloop()

**# bill2.py**

from tkinter import \*

import tkinter.font as font

import mysql.connector as sqltor

import tkinter.ttk as ttk

import random

from datetime import date

from fpdf import FPDF

from tkinter import messagebox

import os

root = Tk()

root.geometry("375x200")

root.configure(bg="deep sky blue")

root.title("User Information")

def on\_closing():

if messagebox.askokcancel("Quit", "Do you want to quit?"):

sys.exit()

root.protocol("WM\_DELETE\_WINDOW", on\_closing)

myfont = font.Font(family="Times", size=25, weight=font.BOLD)

myfont5 = font.Font(family="Times", size=12)

myfont6 = font.Font(family="Times", size=12, weight=font.BOLD)

myfont4 = font.Font(family="Times", size=10)

name\_var = StringVar()

phone\_var = StringVar()

try:

def sub\_fun():

global v1, v2

v1 = name\_var.get()

v2 = phone\_var.get()

root.destroy()

except:

messagebox.showerror('Error', 'Please Enter A Valid Phone Number')

label1=Label(root, text= "Customer Information", bg="deep sky blue", font=myfont)

name\_label = Label(root, text="Enter Name:", bg="deep sky blue")

name\_entry = Entry(root, textvariable=name\_var)

phone\_label = Label(root, text="Enter Phone Number:", bg="deep sky blue")

phone\_entry = Entry(root, textvariable=phone\_var)

sub\_btn = Button(root, text="Submit", command=sub\_fun, activeforeground="dark blue", activebackground="light blue")

label1.grid(row=0, column=0,pady=5, columnspan=2)

name\_label.grid(row=1, column=0, pady=5)

name\_entry.grid(row=1, column=1, pady=5)

phone\_label.grid(row=2, column=0, pady=5)

phone\_entry.grid(row=2, column=1, pady=5)

sub\_btn.grid(row=3, column=1, pady=5)

root.mainloop()

con = sqltor.connect(

host='localhost',

user='root',

passwd='1234',

database='test')

cursor = con.cursor()

top = Tk()

top.geometry("1175x500")

top.configure(bg="cadetblue2")

top.title("Add Items")

item\_var = StringVar()

quant\_var = StringVar()

item\_list = []

grand\_total = 0

def add\_fun():

global item\_list

global grand\_total

v1 = item\_var.get()

v2 = quant\_var.get()

st = "SELECT productname, productprice, productcode FROM items WHERE productcode = '%s'" % (v1,)

cursor.execute(st)

y = cursor.fetchone()

try:

global x

x = [y[2], y[0], v2, y[1]]

price = float(v2)\*float(y[1])

x.append(price)

grand\_total += price

tree.insert('', 'end', values=x)

item\_list.append(x)

except TypeError:

messagebox.showerror('Error', 'Please Enter An Integer In ProductCode and Quantity')

def main\_fun():

top.destroy()

os.system('main1.py')

def del\_fun():

try:

selected\_item = tree.selection()[0]

tree.delete(selected\_item)

except IndexError:

messagebox.showerror('Error', 'Please Select An Item before its Deletion')

def gen\_fun():

global y1

y1 = str(var.get())

if y1 == "0":

messagebox.showerror("Error", "Please Select a Mode of Payment")

return

pdf = FPDF(orientation='P', format='A4')

pdf.add\_page()

pdf.set\_font("Arial", size=30, style='B')

pdf.cell(200, 13, txt="INVOICE", ln=1)

pdf.set\_font("Arial", size=15, style='B')

pdf.cell(10, 12, txt="Invoice Number Date of Issue", ln=2, align="L")

x = str(random.randint(100000000, 999999999))

y = str(date.today())

z = x + " " + y

pdf.set\_font("Arial", size=15)

pdf.cell(10, 14, txt=z, ln=1, align="L")

pdf.set\_font("Arial", size=15, style='B')

# pdf.cell(200, 10, txt='', ln=1, align="L")

pdf.cell(20, 12, txt="Bill To : Seller's Name : ABC Supermarket",

ln=2, align="L")

pdf.set\_font("Arial", size=15)

# pdf.cell(200, 10, txt='', ln=1, align="L")

pdf.cell(200, 12, txt=v1, ln=1, align="L")

pdf.cell(200, 14, txt=v2, ln=1, align="L")

pdf.cell(200, 14, txt="", ln=1, align="L")

pdf.image("ball.png", x=165, y=5, w=40, h=40)

def simple\_table(spacing=3):

pdf.set\_font("Arial", size=12, style="B")

x = ['Item Code', 'Item Name', 'Quantity', 'Price Per Unit', 'Total Price']

col\_width = pdf.w / 5.5

row\_height = pdf.font\_size

for b in x:

pdf.cell(col\_width, row\_height \* spacing, txt=b, border=1)

pdf.ln(row\_height \* spacing)

pdf.set\_font("Arial", size=12)

for row in item\_list:

for item in row:

pdf.cell(col\_width, row\_height \* spacing,

txt=str(item), border=1)

pdf.ln(row\_height \* spacing)

simple\_table()

pdf.cell(175, 10, txt="Grand Total : "+str(grand\_total), ln=1, align="R")

pdf.set\_font("Arial", size=15, style='B')

pdf.cell(200, 10, txt='', ln=1, align="L")

pdf.cell(200, 10, txt="Mode Of Payment ", ln=2, align="L")

pdf.set\_font("Arial", size=12)

pdf.cell(100, 10, txt=y1, ln=1, align="L")

pdf.cell(200, 10, txt='', ln=1, align="L")

pdf.output("simple\_demo.pdf")

os.system('simple\_demo.pdf')

item\_label = Label(top, text="Enter Item Code:", bg="cadetblue2", font=myfont5)

item\_entry = Entry(top, textvariable=item\_var, font=myfont5)

quant\_label = Label(top, text="Enter Quantity:", bg="cadetblue2", font=myfont5)

quant\_entry = Entry(top, textvariable=quant\_var, font=myfont5)

add\_btn = Button(top, text="Add Item", command=add\_fun, activeforeground="dark blue", activebackground="light blue", font=myfont5)

gen\_btn = Button(top, text="Generate Invoice", command=gen\_fun, activeforeground="dark blue",

activebackground="light blue", font=myfont6)

del\_btn = Button(top, text="Delete", command=del\_fun, font=myfont5)

main\_btn = Button(top, text="Main Menu", command=main\_fun, activeforeground="dark blue",

activebackground="light blue", font=myfont5)

invoice\_header = ['Item Code', 'Item Name', 'Quantity', 'Price per Unit']

tree = ttk.Treeview(columns=invoice\_header, show="headings")

vsb = ttk.Scrollbar(orient="vertical", command=tree.yview)

tree.configure(yscrollcommand=vsb.set)

vsb.grid(column=5, row=0, sticky='ns', rowspan=3) # align the scrollbar correctly

label1 = Label(top, text="Payment Mode:", bg="cadetblue2", font=myfont5)

var = StringVar()

var.set("0")

R1 = Radiobutton(top, text="Cash", variable=var, value="Cash", bg="cadetblue2", font=myfont5)

R2 = Radiobutton(top, text="Card", variable=var, value="Card", bg="cadetblue2", font=myfont5)

R3 = Radiobutton(top, text="Digital Payment", variable=var, value="Digital Payment", bg="cadetblue2", font=myfont5)

label = Label(top, bg="cadetblue2")

label.grid(row=8, column=8)

for col in invoice\_header:

tree.heading(col, text=col)

tree.column(0, width=200)

tree.column(1, width=100)

tree.column(2, width=150)

item\_label.grid(row=0, column=0, pady=5, padx=10)

item\_entry.grid(row=0, column=1, pady=5)

quant\_label.grid(row=1, column=0, pady=5)

quant\_entry.grid(row=1, column=1, pady=5, padx=50)

add\_btn.grid(row=2, column=0, pady=5)

gen\_btn.grid(row=2, column=1, pady=5)

tree.grid(row=0, column=2, pady=5, rowspan=3, sticky='nsew', columnspan=3)

del\_btn.grid(row=3, column=2, pady=1, padx=10)

main\_btn.grid(row=5, column=2)

label1.grid(row=4, column=0, pady=1, padx=10)

R1.grid(row=4, column=1, pady=40, padx=10)

R2.grid(row=4, column=2, pady=1, padx=10)

R3.grid(row=4, column=3, pady=1, padx=10)

R1.deselect()

R2.deselect()

R3.deselect()

top.mainloop()

**List of MySQL Commands:-**

create database if not exists test;

use test;

CREATE TABLE items(productcode CHAR(5) PRIMARY KEY, productcategory VARCHAR(15), productname VARCHAR(20),productprice float);

insert into items values("100","Electronics","iPhone 12","80000");

insert into items values("101","Electronics","Galaxy S20","85000");

insert into items values("102","Electronics","iPad Pro","70000");

insert into items values("103","Electronics","PS4","45000");

insert into items values("104","Electronics","Airpods Pro","25000");

insert into items values("105","Electronics","Razer Laptop","95000");

insert into items values("106","Electronics","Dell Laptop","85000");

insert into items values("107","Electronics","HP Laptop","75000");

insert into items values("108","Electronics","Macbook Air","95000");

insert into items values("109","Electronics","iPhone 12 Mini","75000");

insert into items values("110","Electronics","iPhone 12 Pro","120000");

insert into items values("111","Electronics","Oneplus 8 Pro","50000");

insert into items values("112","Electronics","Samsung Refrigerator","95000");

insert into items values("113","Electronics","Samsung OLED TV","95000");

insert into items values("114","Electronics","JBL Headphones","5000");

insert into items values("115","Electronics","LG OLED TV","95000");

insert into items values("116","Electronics","LG Air Conditioner","45000");

insert into items values("117","Electronics","Bose Speaker Set","50000");

insert into items values("118","Electronics","Lenovo Laptop","40000");

insert into items values("119","Electronics","Sony LED TV","95000");

insert into items values("120","Electronics","iPhone SE","35000");

insert into items values("121","Food","Doritos","35");

insert into items values("122","Food","Lays Chips","20");

insert into items values("123","Food","Hide and Seek","50");

insert into items values("123","Food","Bourbon","50");

insert into items values("125","Food","Kissan Jam","150");

insert into items values("126","Food","Amul Butter","200");

insert into items values("127","Food","Dairy Milk","20");

insert into items values("128","Food","Apple","50");

insert into items values("129","Food","Kitkat","25");

insert into items values("130","Food","Munch","20");

insert into items values("131","Food","Perk","10");

insert into items values("132","Food","Orange","50");

insert into items values("133","Food","Lemon","70");

insert into items values("134","Food","Bananas","55");

insert into items values("135","Food","Maggi Noodles","50");

insert into items values("136","Food","Chocolate IceCream","100");

insert into items values("137","Food","Strawberry IceCream","100");

insert into items values("138","Food","Iced Tea","100");

insert into items values("139","Food","Coca Cola","20");

insert into items values("140","Food","Pepsi","25");

insert into items values("141","Clothing","Socks","200");

insert into items values("142","Clothing","Jeans","2000");

insert into items values("143","Clothing","T Shirts","1500");

insert into items values("144","Clothing","Vest","300");

insert into items values("145","Clothing","Sweater","2000");

insert into items values("146","Clothing","Shirt","2000");

insert into items values("147","Clothing","Trousers","2500");

insert into items values("148","Clothing","Jacket","5000");

insert into items values("149","Clothing","Shoes","3000");

insert into items values("150","Clothing","Caps","1000");

insert into items values("151","Clothing","Sneakers","2500");

insert into items values("152","Clothing","Crocs","1500");

insert into items values("153","Clothing","Flip Flops","1500");

insert into items values("154","Clothing","Hat","1500");

insert into items values("155","Clothing","Shorts","1500");

insert into items values("156","Clothing","Capris","2000");

insert into items values("157","Clothing","Muffler","1000");

insert into items values("158","Clothing","Sweatshirt","3000");

insert into items values("159","Clothing","Skirt","1500");

insert into items values("160","Clothing","Lowers","1500");

insert into items values("161","Sports","Yoga Mat","1500");

insert into items values("162","Sports","Cricket Helmet","1000");

insert into items values("163","Sports","Badminton Racquet","3000");

insert into items values("164","Sports","Tennis Racquet","5000");

insert into items values("165","Sports","Golf Stick","5000");

insert into items values("166","Sports","Cricket Bat","5000");

insert into items values("167","Sports","Football","3000");

insert into items values("168","Sports","Squash Racquet","5000");

insert into items values("169","Sports","Cricket Ball","500");

insert into items values("170","Sports","Tennis Ball","500");

insert into items values("171","Sports","Squash Ball","500");

insert into items values("172","Sports","Swimming Trunks","1500");

insert into items values("173","Sports","Swimming Cap","1000");

insert into items values("174","Sports","Swimming Goggles","500");

insert into items values("175","Sports","Badminton Shuttle","500");

insert into items values("176","Sports","Golf Ball","2000");

insert into items values("177","Sports","Baseball Bat","5000");

insert into items values("178","Sports","Golf Gloves","1500");

insert into items values("179","Sports","Skipping Rope","500");

insert into items values("180","Sports","Basketball","1000");

insert into items values("181","Daily Essentials","Milk","500");

insert into items values("182","Daily Essentials","Bread","500");

insert into items values("183","Daily Essentials","Soap","300");

insert into items values("184","Daily Essentials","Detergent","500");

insert into items values("185","Daily Essentials","Shampoo","400");

insert into items values("186","Daily Essentials","Oil","400");

insert into items values("187","Daily Essentials","Wheat","400");

insert into items values("188","Daily Essentials","Rice","400");

insert into items values("189","Daily Essentials","Perfume","1000");

insert into items values("190","Daily Essentials","Maize","500");

insert into items values("191","Daily Essentials","Salt","300");

insert into items values("192","Daily Essentials","Sugar","300");

insert into items values("193","Daily Essentials","Deodorant","1000");

insert into items values("194","Daily Essentials","Toothbrush","200");

insert into items values("195","Daily Essentials","Toothpaste","300");

insert into items values("196","Daily Essentials","Body Wash","600");

insert into items values("197","Daily Essentials","Water Bottle","700");

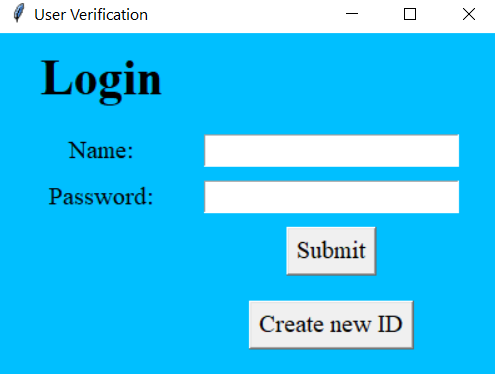
insert into items values("198","Daily Essentials","Soda","200");

insert into items values("199","Daily Essentials","Dal","300");

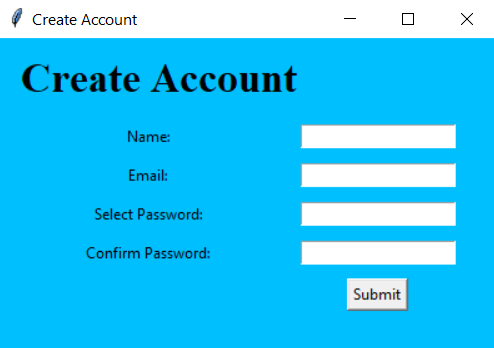
insert into items values("200","Daily Essentials","Baby Powder","300");

**Outputs:-**

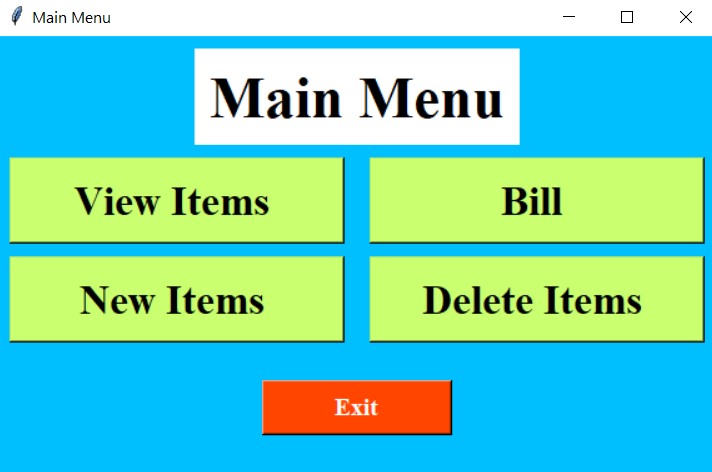
**start.py**



**new.py**

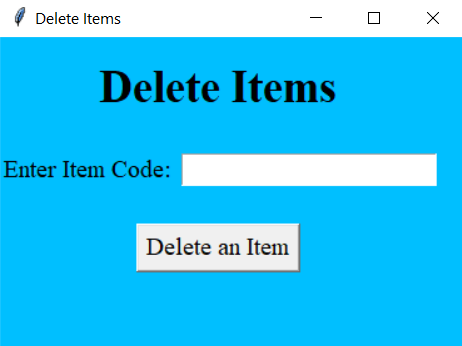


**main.py**

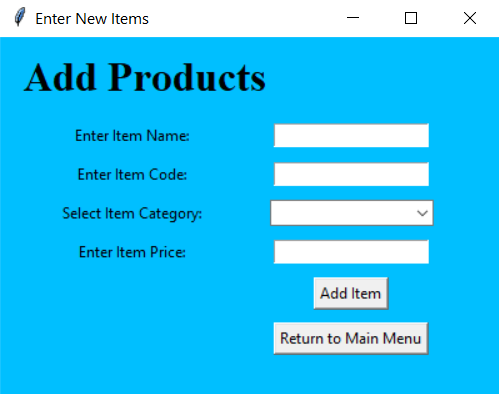


**view.py**

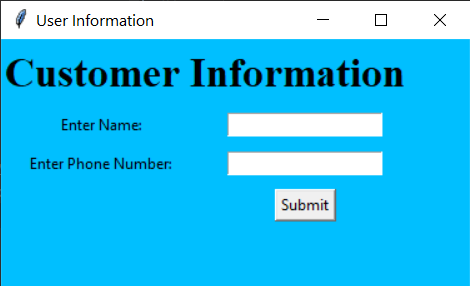
**deletion.py**

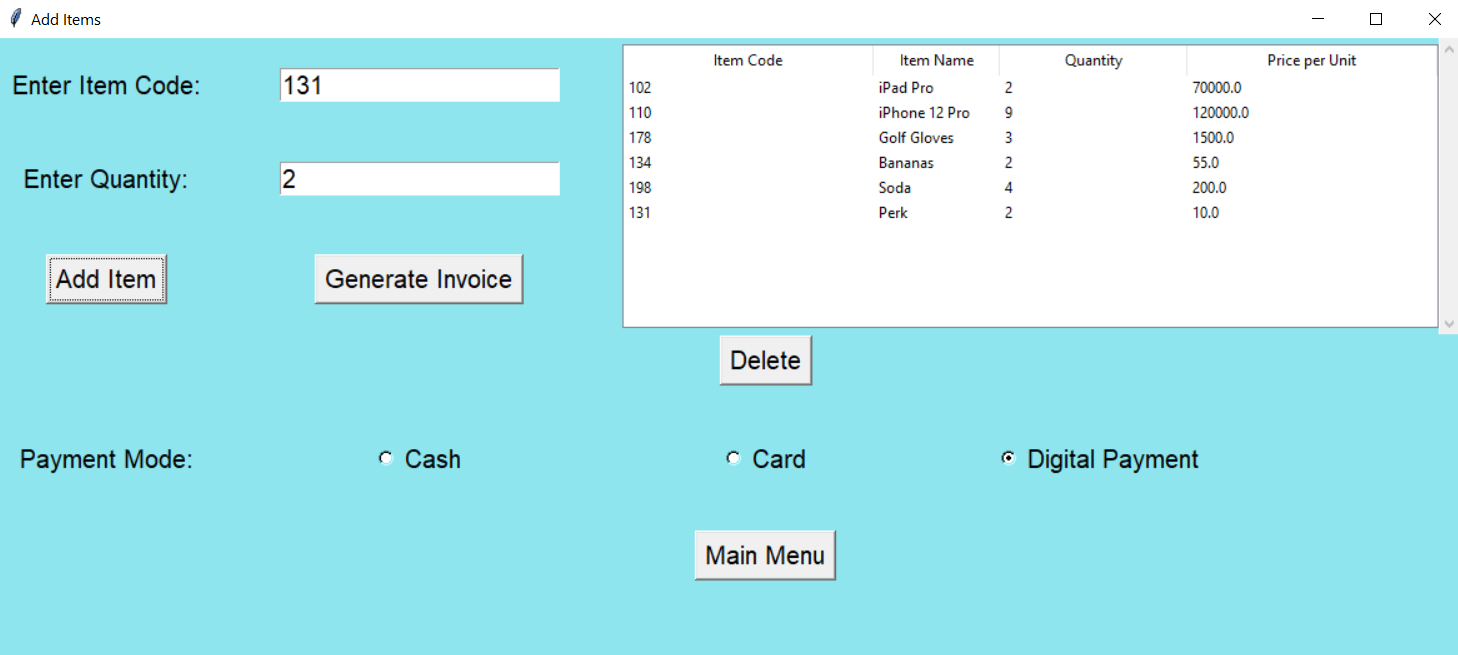


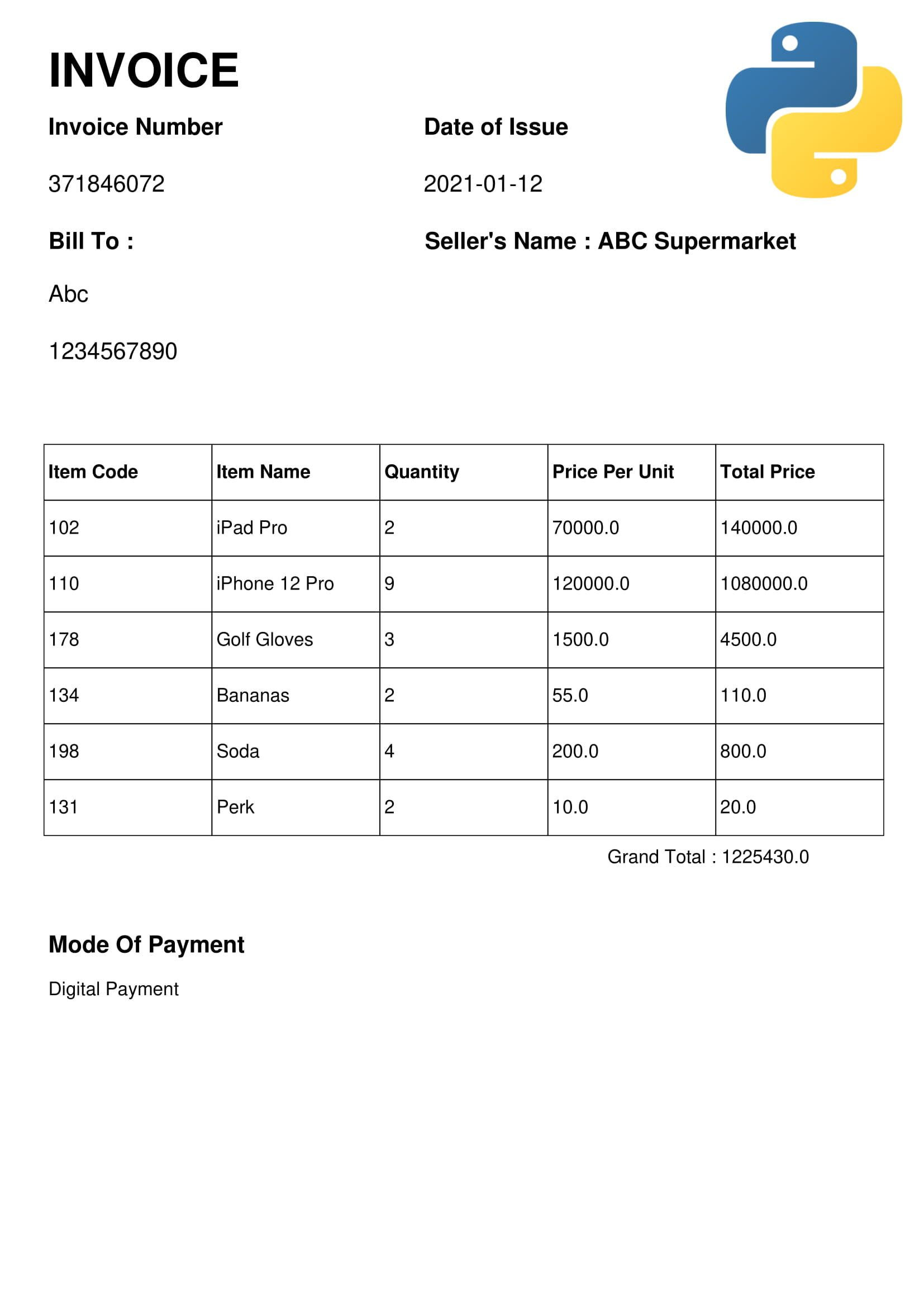
**Items.py**



**bill2.py**





**Generated Invoice (Main Output)**

**Conclusion**

By making this project I was able to make my doubts clear about the two most difficult topics of the syllabus. I learnt a new way of programming and it also helped me to learn new methods to perform the same task differently and efficiently.

This software makes accounting easier, quicker and much more efficient than performing the task manually. It saves time and money, the two most precious things which cannot be wasted in this century.

If taken in use, it will improve database management and help in planning sales accordingly.