**OSTL MINI-PROJECT**

**RESTAURANT MANAGEMENT SYSTEM**

**-G.Saiprasad -114A1012**

**-Sunil P -116A1058**

**-Shweta S -116A1072**

**Introduction:**

Python is high level language which has plenty of features and is a compilation of best features from many former languages like C, C++, Java etc. We have used tkinter package present in Python which enables us to create GUI based application. Tkinter consists of many inbuilt modules. Tkinter provides us with a variety of common GUI elements which we can use to build our interface – such as buttons, menus and various kinds of entry fields and display areas. We call these elements widgets. We are going to construct a tree of widgets for our GUI – each widget will have a parent widget, all the way up to the root window of our application. For example, a button or a text field needs to be inside some kind of containing window.

The widget classes provide us with a lot of default functionality. They have methods for configuring the GUI’s appearance – for example, arranging the elements according to some kind of layout – and for handling various kinds of user-driven events. Once we have constructed the backbone of our GUI, we will need to customise it by integrating it with our internal application class.

SQLite3 is a very easy to use database engine. It is self-contained, serverless, zero-configuration and transactional. It is very fast and lightweight, and the entire database is stored in a single disk file. It is used in a lot of applications as internal data storage. The Python Standard Library includes a module called "sqlite3" intended for working with this database. This module is a SQL interface compliant with the DB-API 2.0 specification.

**Problem Definition:**

To design a user friendly Restaurant Management system

We have created a Restaurant Management System using tkinter and established the database connectivity using sqlite3. Basically, it provides a GUI to the receptionist in a restaurant to generate bill and automatically generates a pdf as well which gets saved in the same directory under the reference no. generated randomly. On providing the Food Items, their respective prices are fetched from the database and computations are done them to provide the total amount to the user. Our system also generates error on giving wrong input. The GUI and database are synced and also the pdf generation system makes it user friendly.

We have made appealing to the user as well by providing lot of styling and a logo of our restaurant as well.

**MODULES:**

The different types of packages each consisting of following modules:

1. Tkinter package
   1. messagebox

The tkMessageBox module is used to display message boxes in your applications. This module provides a number of functions that you can use to display an appropriate message.

Some of these functions are showinfo, showwarning, showerror, askquestion, askokcancel, askyesno, and askretryignore.

**Syntax**

tkMessageBox.FunctionName(title, message [, options])

* 1. sqlite3:

To use sqlite3 module, you must first create a connection object that represents the database and then optionally you can create a cursor object, which will help you in executing all the SQL statements. SQLite3 is a compact free database you can use easily create and use a database.  Though SQLite3 is not a full- featured database, it supports a surprisingly large set of the SQL standard, and is ideal for those just starting to learn SQL as well for developers that need a simple database engine to plug into their applications.  As such, SQLite has become very popular with smart phone developers.

* 1. time

There is a popular **time** module available in Python which provides functions for working with times and for converting between representations.

syntax: [**time.strptime(str,fmt='%a %b %d %H:%M:%S %Y')**](https://www.tutorialspoint.com/python/time_strptime.htm)

## Directive

* %a - abbreviated weekday name
* %b - abbreviated month name
* %e - day of the month (1 to 31)
* %G - 4-digit year corresponding to the ISO week number (see %V).
* %H - hour, using a 24-hour clock (00 to 23)
* %M - minute
  1. random

This module implements pseudo-random number generators for various distributions.

In Python, just like in almost any other OOP language, chances are that you'll find yourself needing to generate a random at some point. Whether you're just completing an exercise in algorithms to better familiarize yourself with the language, or if you're trying to write more complex code, you can't call yourself a Python coder without knowing how to generate random numbers.

syntax:

random.randint(1,101)

* 1. webbrowser

we have used this to open the pdf file of bill as popup

2) Reportlab

1. reportlab.pdfgen

this is the programming interface to the PDF file format. The Canvas (and its co-workers, TextObject and PathObject) provide everything you need to create PDF output working at a low level - individual shapes and lines of text. Internally, it constructs blocks of page marking operators which match your drawing commands, and hand them over to the pdfbase package for drawing.

1. reportlab.lib

this contains code of interest to application developers which cuts across both of our libraries, such as standard colors, units, and page sizes. It will also contain more drawable and flowable objects in future.

1. Platypus

PLATYPUS stands for "Page Layout and Typography Using Scripts". It provides a higher level of abstraction dealing with paragraphs, frames on the page, and document templates. This is used for multi- page documents

**IMPLEMENTATION:**

The main geometry of the GUI system is of 1600x800 and there are 2 frames. Frame 'f' is the top frame of geometry 1600x200 where the logo, time, day, date are displayed.

To import the logo we used photoimage function which is inbuilt in the tkinter module. The image file is stored in the photo variable. A label is created. The photo variable is passed to the label as a parameter and it is packed.

For time in frame 'f' we have used the time package.

The bottom frame 'f1' is of geometry 1600x600. This is the frame where the main structure for the billing GUI is created with reference no., input on the types of food , amount of quantity ,the total, GST gross total , net total , mode of payment , new bill button, generate bill button, proceed to payment button and submit button are created. We have used various parameters of Label, Entry, OptionMenu, RadioButton to style their display in the GUI. We have used parameters like bg - background colour, bd – border depth, font – to customize the font, anchor, sticky – for allignmet etc.

Here we have used the random function to generate random reference no. In the food item column the various categories of food are displayed using widget of option menu. We have used entry box for the user input for quantity, the prices fetched from the database and the quantity and price are multiplied giving the output for total amount and the total column is automatically updated. In the mode of payment section we have used radiobutton widgets to specify which mode of payment the customer has opted for. Here we have used grid to align and place different types of widgets and entry boxes in this frame ‘f1’.

Once the submit button is clicked the quantity entered and food item selected is passed to the database to fetch its price and calculate the total. If the user accidently enters a non-digit input in the quantity then an error message pops-up saying ‘Invalid Input’ and restricting user to further proceed unless the user rectifies the error.

For back-end we are using sqlite3. We are creating a table with 2 columns – 1 for the food items and another for their respective prices. When a menu item is selected in the GUI from the dropdown list, its rate is fetched from the database. Here the rates are retrieved and passed to the calculation function where the total is calculated multiplying rate and quantity.

In calculation function, the passed rate is fetched and multiplied with quantity. The total, GST, net total, gross total is calculated and are set to their respective text boxes.

The mode of payment is selected from the radiobutton in the GUI and the option selected is displayed in the shell.

Once the ‘generate bill’ button is clicked, the pdf of the bill is generated and pops-up in the user window. For the pdf bill we have used reportlab module. We have fetched the various objects and passed it into the pdf at specified positions and also we have displayed the same logo as in the GUI and even specified its location.

When new bill button is clicked it resets the entire GUI, a new bill reference no. Is generated and all the other fields are cleared for giving input for the new bill.

**Program:**

from tkinter import \*

from tkinter import messagebox

import sqlite3

import time

import random

import webbrowser as wb

from reportlab.pdfgen import canvas

from reportlab.lib.pagesizes import letter

from reportlab.lib.pagesizes import landscape

from reportlab.platypus import Image

root=Tk()

root.geometry("1600x800+0+0")

root.title("Restaurant Management System")

f=Frame(root,width=1600,height=200,bg='White',relief=SUNKEN)

f.pack(side=TOP,fill=BOTH,expand=True)

f1=Frame(root,width=1600,height=600,bg='White',relief=SUNKEN)

f1.pack(side=BOTTOM,fill=BOTH,expand=True)

#==============================Info================================

photo=PhotoImage(file="ambrosia.png")

lblimg=Label(f, image=photo, bg='White')

lblimg.pack()

lblInfo=Label(f,font=('Helvetica',20,'bold'),text="(Pure Veg. Restaurant)",bg='White',bd=5,anchor='center')

lblInfo.pack()

#=============================Time=================================

localtime=time.strftime("%a %b %e %H:%M %G", time.localtime(time.time()))

lblInfo=Label(f,font=('arial',20,'bold'),text=localtime,bd=5,bg='White',anchor='center')

lblInfo.pack()

#===========================Button Function===========================

def ButtonClick():

txtq1=txtquant1.get()

txtq2=txtquant2.get()

txtq3=txtquant3.get()

txtq4=txtquant4.get()

txtq5=txtquant5.get()

if not(txtq1.isdigit() or tkvar1.get()=="Item 1") or not(txtq2.isdigit() or tkvar2.get()=="Item 2") or not(txtq3.isdigit() or tkvar3.get()=="Item 3") or not(txtq4.isdigit() or tkvar4.get()=="Item 4") or not(txtq5.isdigit() or tkvar5.get()=="Item 5"):

messagebox.showerror("Error", "Invalid Input")

else:

i1=tkvar1.get()

i2=tkvar2.get()

i3=tkvar3.get()

i4=tkvar4.get()

i5=tkvar5.get()

insertdb(i1,i2,i3,i4,i5,txtquant1.get(),txtquant2.get(),txtquant3.get(),txtquant4.get(),txtquant5.get())

def PayClick():

val=v.get()

if val==1:

print("You Selected Card Payment Mode")

if val==2:

print("You Selected Cash Payment Mode")

if val==3:

print("You Selected E-Wallet Payment Mode")

#==============================Generate Bill==========================

def BillClick():

#print("Nothing")

sample=rand.get()

if tkvar1.get()=='Item 1':

tkvar1.set("")

if tkvar2.get()=='Item 2':

tkvar2.set("")

if tkvar3.get()=='Item 3':

tkvar3.set("")

if tkvar4.get()=='Item 4':

tkvar4.set("")

if tkvar5.get()=='Item 5':

tkvar5.set("")

if total1.get()=='0.0':

total1.set("")

if total2.get()=='0.0':

total2.set("")

if total3.get()=='0.0':

total3.set("")

if total4.get()=='0.0':

total4.set("")

if total5.get()=='0.0':

total5.set("")

c = canvas.Canvas(rand.get()+'.pdf',pagesize=landscape(letter))

resimg="ambrosia.png"

c.drawImage(resimg,250,450,width=None,height=None)

c.setFont('Helvetica',20,leading=None)

c.drawCentredString(675,450,"(Pure Veg. Restaurant)")

c.setFont('Helvetica',20,leading=None)

c.drawCentredString(300,425,"Ref.No.")

c.setFont('Helvetica',20,leading=None)

c.drawCentredString(400,425,rand.get())

c.setFont('Helvetica',20,leading=None)

c.drawCentredString(300,400,"Date")

c.setFont('Helvetica',20,leading=None)

c.drawCentredString(400,400,time.strftime("%D", time.localtime(time.time())))

c.setFont('Helvetica',20,leading=None)

c.drawCentredString(300,350,"ITEMS")

c.setFont('Helvetica',20,leading=None)

c.drawCentredString(450,350,"QUANTITY")

c.setFont('Helvetica',20,leading=None)

c.drawCentredString(600,350,"PRICE")

c.setFont('Helvetica',20,leading=None)

c.drawCentredString(300,300,tkvar1.get())

c.setFont('Helvetica',20,leading=None)

c.drawCentredString(450,300,txtquant1.get())

c.setFont('Helvetica',20,leading=None)

c.drawCentredString(600,300,total1.get())

c.setFont('Helvetica',20,leading=None)

c.drawCentredString(300,275,tkvar2.get())

c.setFont('Helvetica',20,leading=None)

c.drawCentredString(450,275,txtquant2.get())

c.setFont('Helvetica',20,leading=None)

c.drawCentredString(600,275,total2.get())

c.setFont('Helvetica',20,leading=None)

c.drawCentredString(300,250,tkvar3.get())

c.setFont('Helvetica',20,leading=None)

c.drawCentredString(450,250,txtquant3.get())

c.setFont('Helvetica',20,leading=None)

c.drawCentredString(600,250,total3.get())

c.setFont('Helvetica',20,leading=None)

c.drawCentredString(300,225,tkvar4.get())

c.setFont('Helvetica',20,leading=None)

c.drawCentredString(450,225,txtquant4.get())

c.setFont('Helvetica',20,leading=None)

c.drawCentredString(600,225,total4.get())

c.setFont('Helvetica',20,leading=None)

c.drawCentredString(300,200,tkvar5.get())

c.setFont('Helvetica',20,leading=None)

c.drawCentredString(450,200,txtquant5.get())

c.setFont('Helvetica',20,leading=None)

c.drawCentredString(600,200,total5.get())

c.setFont('Helvetica',20,leading=None)

c.drawCentredString(450,150,"NET TOTAL")

c.setFont('Helvetica',20,leading=None)

c.drawCentredString(600,150,ntotal.get())

c.setFont('Helvetica',20,leading=None)

c.drawCentredString(450,125,"GST")

c.setFont('Helvetica',20,leading=None)

c.drawCentredString(600,125,gst.get())

c.setFont('Helvetica',20,leading=None)

c.drawCentredString(450,100,"GROSS TOTAL")

c.setFont('Helvetica',20,leading=None)

c.drawCentredString(600,100,gtotal.get())

c.setFont('Helvetica',20,leading=None)

c.drawCentredString(600,50,"Signature")

c.showPage()

c.save()

filepath="{}.pdf".format(sample)

wb.open\_new(r'{}.pdf'.format(sample))

#===============================New Bill===========================

def Clear():

rand.set(random.randint(1,10001))

tkvar1.set('Item 1')

tkvar2.set('Item 2')

tkvar3.set('Item 3')

tkvar4.set('Item 4')

tkvar5.set('Item 5')

txtquant1.delete(0,'end')

txtquant2.delete(0,'end')

txtquant3.delete(0,'end')

txtquant4.delete(0,'end')

txtquant5.delete(0,'end')

txtrate1.delete(0,'end')

txtrate2.delete(0,'end')

txtrate3.delete(0,'end')

txtrate4.delete(0,'end')

txtrate5.delete(0,'end')

txttotal1.delete(0,'end')

txttotal2.delete(0,'end')

txttotal3.delete(0,'end')

txttotal4.delete(0,'end')

txttotal5.delete(0,'end')

txtNTotal.delete(0,'end')

txtGST.delete(0,'end')

txtGTotal.delete(0,'end')

v.set(4)

#================================Calculations========================

def CalcTotal(r1,r2,r3,r4,r5):

q1=txtquant1.get()

q2=txtquant2.get()

q3=txtquant3.get()

q4=txtquant4.get()

q5=txtquant5.get()

rate1.set(str(r1))

rate2.set(str(r2))

rate3.set(str(r3))

rate4.set(str(r4))

rate5.set(str(r5))

t1=float(int(q1.strip() or 0)\*int(r1.strip() or 0))

t2=float(int(q2.strip() or 0)\*int(r2.strip() or 0))

t3=float(int(q3.strip() or 0)\*int(r3.strip() or 0))

t4=float(int(q4.strip() or 0)\*int(r4.strip() or 0))

t5=float(int(q5.strip() or 0)\*int(r5.strip() or 0))

nt=(t1+t2+t3+t4+t5)

perc=nt\*5/100

gt=(nt+perc)

total1.set(str(t1))

total2.set(str(t2))

total3.set(str(t3))

total4.set(str(t4))

total5.set(str(t5))

ntotal.set(str(nt))

gst.set(str(perc))

gtotal.set(str(gt))

#===============================Declarations=========================

rand=StringVar()

rate1=StringVar()

rate2=StringVar()

rate3=StringVar()

rate4=StringVar()

rate5=StringVar()

total1=StringVar()

total2=StringVar()

total3=StringVar()

total4=StringVar()

total5=StringVar()

ntotal=StringVar()

gst=StringVar()

gtotal=StringVar()

v=IntVar()

tkvar1=StringVar()

tkvar2=StringVar()

tkvar3=StringVar()

tkvar4=StringVar()

tkvar5=StringVar()

rand.set(random.randint(1,10001))

#============================ReferenceNo============================

lblReference= Label(f1,font=('arial',16,'bold'),text="Reference",bd=5,bg='White',anchor='w')

lblReference.grid(row=0,column=0)

txtReference=Entry(f1,font=('arial',16,'bold'),textvariable=rand,bd=5,bg="#E5E7E9", justify='right')

txtReference.grid(row=0,column=1)

#==========================Dropdown Options==========================

values1={'Veg. Clear Soup','Tomato Soup','Veg Manchow Soup','Veg. Hot & Sour Soup','Sweet Corn Soup','Mushroom Soup'}

values2={'Plain Rice','Jeera Rice','Veg Pulav','Veg Biryani','Peas Pulav','Paneer Pulav','Lemon Rice','Phulka (Tawa Roti)','Puri','Butter Phulka','Paneer Paratha','Stuff Paratha','Butter Tanduri Roti','Garlic Naan','Lachha Paratha','Masala Kulcha','Makke Ki Roti'}

values3={'Plain Dal','Dal Fry','Dal Tadka','Dal Maharani','Dal Makhani','Mix Veg','Veg Kadhai','Veg Handi','Veg Kofta','Chana Masala','Aalu Jeera','Dum Aalu','Aalu Gobi Masala','Paneer Butter Masala','Paneer Kofta','Paneer Bhurji','Palak Paneer','Mutter Paneer','Shahi Paneer','Sarso Ka Saag'}

values4={'Bundi Raita','Mix Veg. Raita','Onion Salad','Green Salad'}

values5={'Banana Milkshake','Chocolate Milkshake','Kiwi Milkshake','Strawberry Milkshake','Blackcurrant Milkshake','Mix Fruit Milkshake','Mango Milkshake','Blue Lagoon','Ginger & Mint','Fruit Punch','Lemon & Blackcurrant','Double Scoop(Choco,Vanilla)','Choco Blast Icecream','Cassata','Malai Kulfi(Pista)','Gulab Jamoon','Rasgulla'}

#initial display on optionmenu widget

tkvar1.set('Item 1')

tkvar2.set('Item 2')

tkvar3.set('Item 3')

tkvar4.set('Item 4')

tkvar5.set('Item 5')

#==========================Column Headings==========================

lblfitem=Label(f1,font=('arial',16,'bold'),text="Food Item",bd=5,bg='White',anchor='w')

lblfitem.grid(row=1,column=1)

lbq1=Label(f1,font=('arial',16,'bold'),text="Quantity",bd=5,bg='White',anchor='w')

lbq1.grid(row=1,column=2)

lbq1=Label(f1,font=('arial',16,'bold'),text="Price",bd=5,bg='White',anchor='w')

lbq1.grid(row=1,column=3)

lbq1=Label(f1,font=('arial',16,'bold'),text="Total",bd=5,bg='White',anchor='w')

lbq1.grid(row=1,column=4)

lblpm=Label(f1,font=('arial',16,'bold'),text="Mode Of Payment",bd=5,bg='White',anchor='w')

lblpm.grid(row=1,column=5)

#=============================Row Headings==========================

lblItem1=Label(f1,font=('arial',16,'bold'),text="Starters",bd=5,bg='White',anchor='w')

lblItem1.grid(row=2,column=0)

lblItem2=Label(f1,font=('arial',16,'bold'),text="Main Course",bd=5,bg='White',anchor='w',)

lblItem2.grid(row=3,column=0)

lblItem3=Label(f1,font=('arial',16,'bold'),text="Side Dish",bd=5,bg='White',anchor='w')

lblItem3.grid(row=4,column=0)

lblItem4=Label(f1,font=('arial',16,'bold'),text="Raita/Salads",bd=5,bg='White',anchor='w')

lblItem4.grid(row=5)

lblItem5=Label(f1,font=('arial',16,'bold'),text="Dessert/Milk Shakes",bd=5,bg='White',anchor='w')

lblItem5.grid(row=6,column=0)

lblNTotal=Label(f1,font=('arial',16,'bold'),text="Net Total",bd=5,bg='White',anchor='w')

lblNTotal.grid(row=7,column=3)

lblGST=Label(f1,font=('arial',16,'bold'),text="GST",bd=5,bg='White',anchor='w')

lblGST.grid(row=8,column=3)

lblGTotal=Label(f1,font=('arial',16,'bold'),text="Gross Total",bd=5,bg='White',anchor='w')

lblGTotal.grid(row=9,column=3)

#=========================Food item option selection=====================

opItem1=OptionMenu(f1,tkvar1,\*values1)

opItem1.config(font=('arial',16,'bold'),bd=5,bg="#E5E7E9",justify='right')

opItem1["menu"].config(font=('arial',16,'bold'),bd=5,bg="#E5E7E9")

opItem1.grid(row=2,column=1,sticky="ew")

opItem2=OptionMenu(f1,tkvar2,\*values2)

opItem2.config(font=('arial',16,'bold'),bd=5,bg="#E5E7E9",justify='right')

opItem2["menu"].config(font=('arial',16,'bold'),bd=5,bg="#E5E7E9")

opItem2.grid(row=3,column=1,sticky="ew")

opItem3=OptionMenu(f1,tkvar3,\*values3)

opItem3.config(font=('arial',16,'bold'),bd=5,bg="#E5E7E9",justify='right')

opItem3["menu"].config(font=('arial',16,'bold'),bd=5,bg="#E5E7E9")

opItem3.grid(row=4,column=1,sticky="ew")

opItem4=OptionMenu(f1,tkvar4,\*values4)

opItem4.config(font=('arial',16,'bold'),bd=5,bg="#E5E7E9",justify='right')

opItem4["menu"].config(font=('arial',16,'bold'),bd=5,bg="#E5E7E9")

opItem4.grid(row=5,column=1,sticky="ew")

opItem5=OptionMenu(f1,tkvar5,\*values5)

opItem5.config(font=('arial',16,'bold'),bd=5,bg="#E5E7E9",justify='right')

opItem5["menu"].config(font=('arial',16,'bold'),bd=5,bg="#E5E7E9")

opItem5.grid(row=6,column=1,sticky="ew")

#============================Quantity Input===========================

txtquant1=Entry(f1,font=('arial',16,'bold'),bd=5,bg="#E5E7E9", justify='right')

txtquant1.grid(row=2,column=2,sticky=W)

txtquant2=Entry(f1,font=('arial',16,'bold'),bd=5,bg="#E5E7E9", justify='right')

txtquant2.grid(row=3,column=2,sticky=W)

txtquant3=Entry(f1,font=('arial',16,'bold'),bd=5,bg="#E5E7E9", justify='right')

txtquant3.grid(row=4,column=2,sticky=W)

txtquant4=Entry(f1,font=('arial',16,'bold'),bd=5,bg="#E5E7E9", justify='right')

txtquant4.grid(row=5,column=2,sticky=W)

txtquant5=Entry(f1,font=('arial',16,'bold'),bd=5,bg="#E5E7E9", justify='right')

txtquant5.grid(row=6,column=2,sticky=W)

#==========================Rate Outputs==============================

txtrate1=Entry(f1,font=('arial',16,'bold'),textvariable=rate1,bd=5,bg="#E5E7E9", justify='right')

txtrate1.grid(row=2,column=3,sticky=W)

txtrate2=Entry(f1,font=('arial',16,'bold'),textvariable=rate2,bd=5,bg="#E5E7E9", justify='right')

txtrate2.grid(row=3,column=3,sticky=W)

txtrate3=Entry(f1,font=('arial',16,'bold'),textvariable=rate3,bd=5,bg="#E5E7E9", justify='right')

txtrate3.grid(row=4,column=3,sticky=W)

txtrate4=Entry(f1,font=('arial',16,'bold'),textvariable=rate4,bd=5,bg="#E5E7E9", justify='right')

txtrate4.grid(row=5,column=3,sticky=W)

txtrate5=Entry(f1,font=('arial',16,'bold'),textvariable=rate5,bd=5,bg="#E5E7E9", justify='right')

txtrate5.grid(row=6,column=3,sticky=W)

#===========================Row Totals==============================

txttotal1=Entry(f1,font=('arial',16,'bold'),textvariable=total1,bd=5,bg="#E5E7E9", justify='right')

txttotal1.grid(row=2,column=4,sticky=W)

txttotal2=Entry(f1,font=('arial',16,'bold'),textvariable=total2,bd=5,bg="#E5E7E9", justify='right')

txttotal2.grid(row=3,column=4,sticky=W)

txttotal3=Entry(f1,font=('arial',16,'bold'),textvariable=total3,bd=5,bg="#E5E7E9", justify='right')

txttotal3.grid(row=4,column=4,sticky=W)

txttotal4=Entry(f1,font=('arial',16,'bold'),textvariable=total4,bd=5,bg="#E5E7E9", justify='right')

txttotal4.grid(row=5,column=4,sticky=W)

txttotal5=Entry(f1,font=('arial',16,'bold'),textvariable=total5,bd=5,bg="#E5E7E9", justify='right')

txttotal5.grid(row=6,column=4,sticky=W)

#================================Final Totals========================

txtNTotal=Entry(f1,font=('arial',16,'bold'),textvariable=ntotal,bd=5,bg="#E5E7E9", justify='right')

txtNTotal.grid(row=7,column=4,sticky=W)

txtGST=Entry(f1,font=('arial',16,'bold'),textvariable=gst,bd=5,bg="#E5E7E9", justify='right')

txtGST.grid(row=8,column=4,sticky=W)

txtGTotal=Entry(f1,font=('arial',16,'bold'),textvariable=gtotal,bd=5,bg="#E5E7E9", justify='right')

txtGTotal.grid(row=9,column=4,sticky=W)

#==================================Payment Modes====================

rb1=Radiobutton(f1,text="Card",variable=v,value=1,font=('arial',16,'bold'),bd=5,bg='White',justify='right')

rb1.grid(row=2,column=5)

rb2=Radiobutton(f1,text="Cash",variable=v,value=2,font=('arial',16,'bold'),bd=5,bg='White',justify='right')

rb2.grid(row=3,column=5)

rb3=Radiobutton(f1,text="E-Wallet",variable=v,value=3,font=('arial',16,'bold'),bd=5,bg='White',justify='right')

rb3.grid(row=4,column=5)

#=================================Buttons===========================

btn1=Button(f1,bd=5,fg="black",font=('arial',16,'bold'),text="Submit",bg="#E5E7E9", justify='right',command=ButtonClick).grid(row=10,column=0)

btn2=Button(f1,bd=5,fg="black",font=('arial',16,'bold'),text="Proceed To Payment",bg="#E5E7E9", justify='right',command=PayClick).grid(row=10,column=1)

btn3=Button(f1,bd=5,fg="black",font=('arial',16,'bold'),text="Generate Bill",bg="#E5E7E9", justify='right',command=BillClick).grid(row=10,column=2)

btn4=Button(f1,bd=5,fg="black",font=('arial',16,'bold'),text="New Bill",bg="#E5E7E9", justify='right',command=Clear).grid(row=10,column=3)

#===================================Database========================

def insertdb(A,B,C,D,E,qa,qb,qc,qd,qe):

conn=sqlite3.connect("Restaurant.db")

c=conn.cursor()

c.execute("drop table Menu")

c.execute("create table Menu(Item varchar(20), Cost varchar(5))")

c.execute("insert into Menu values('%s','%s')" %('Veg. Clear Soup','40'))

c.execute("insert into Menu values('%s','%s')" %('Tomato Soup','50'))

c.execute("insert into Menu values('%s','%s')" %('Veg Manchow Soup','60'))

c.execute("insert into Menu values('%s','%s')" %('Veg. Hot & Sour Soup','60'))

c.execute("insert into Menu values('%s','%s')" %('Sweet Corn Soup','60'))

c.execute("insert into Menu values('%s','%s')" %('Mushroom Soup','65'))

c.execute("insert into Menu values('%s','%s')" %('Plain Rice','50'))

c.execute("insert into Menu values('%s','%s')" %('Jeera Rice','60'))

c.execute("insert into Menu values('%s','%s')" %('Veg Pulav','75'))

c.execute("insert into Menu values('%s','%s')" %('Veg Biryani','80'))

c.execute("insert into Menu values('%s','%s')" %('Peas Pulav','75'))

c.execute("insert into Menu values('%s','%s')" %('Paneer Pulav','90'))

c.execute("insert into Menu values('%s','%s')" %('Lemon Rice','65'))

c.execute("insert into Menu values('%s','%s')" %('Plain Dal','50'))

c.execute("insert into Menu values('%s','%s')" %('Dal Fry','65'))

c.execute("insert into Menu values('%s','%s')" %('Dal Tadka','75'))

c.execute("insert into Menu values('%s','%s')" %('Dal Maharani','85'))

c.execute("insert into Menu values('%s','%s')" %('Dal Makhani','85'))

c.execute("insert into Menu values('%s','%s')" %('Phulka (Tawa Roti)','06'))

c.execute("insert into Menu values('%s','%s')" %('Puri','06'))

c.execute("insert into Menu values('%s','%s')" %('Butter Phulka','09'))

c.execute("insert into Menu values('%s','%s')" %('Paneer Paratha','50'))

c.execute("insert into Menu values('%s','%s')" %('Stuff Paratha','40'))

c.execute("insert into Menu values('%s','%s')" %('Butter Tanduri Roti','15'))

c.execute("insert into Menu values('%s','%s')" %('Garlic Naan','30'))

c.execute("insert into Menu values('%s','%s')" %('Lachha Paratha','20'))

c.execute("insert into Menu values('%s','%s')" %('Masala Kulcha','25'))

c.execute("insert into Menu values('%s','%s')" %('Makke Ki Roti','25'))

c.execute("insert into Menu values('%s','%s')" %('Mix Veg','75'))

c.execute("insert into Menu values('%s','%s')" %('Veg Kadhai','85'))

c.execute("insert into Menu values('%s','%s')" %('Veg Handi','85'))

c.execute("insert into Menu values('%s','%s')" %('Veg Kofta','90'))

c.execute("insert into Menu values('%s','%s')" %('Chana Masala','80'))

c.execute("insert into Menu values('%s','%s')" %('Aalu Jeera','65'))

c.execute("insert into Menu values('%s','%s')" %('Dum Aalu','70'))

c.execute("insert into Menu values('%s','%s')" %('Aalu Gobi Masala','85'))

c.execute("insert into Menu values('%s','%s')" %('Paneer Butter Masala','115'))

c.execute("insert into Menu values('%s','%s')" %('Paneer Kofta','120'))

c.execute("insert into Menu values('%s','%s')" %('Paneer Bhurji','110'))

c.execute("insert into Menu values('%s','%s')" %('Palak Paneer','115'))

c.execute("insert into Menu values('%s','%s')" %('Mutter Paneer','110'))

c.execute("insert into Menu values('%s','%s')" %('Shahi Paneer','115'))

c.execute("insert into Menu values('%s','%s')" %('Sarso Ka Saag','95'))

c.execute("insert into Menu values('%s','%s')" %('Bundi Raita','50'))

c.execute("insert into Menu values('%s','%s')" %('Mix Veg. Raita','60'))

c.execute("insert into Menu values('%s','%s')" %('Onion Salad','25'))

c.execute("insert into Menu values('%s','%s')" %('Green Salad','35'))

c.execute("insert into Menu values('%s','%s')" %('Banana Milkshake','60'))

c.execute("insert into Menu values('%s','%s')" %('Chocolate Milkshake','70'))

c.execute("insert into Menu values('%s','%s')" %('Kiwi Milkshake','80'))

c.execute("insert into Menu values('%s','%s')" %('Strawberry Milkshake','70'))

c.execute("insert into Menu values('%s','%s')" %('Blackcurrant Milkshake','70'))

c.execute("insert into Menu values('%s','%s')" %('Mix Fruit Milkshake','70'))

c.execute("insert into Menu values('%s','%s')" %('Mango Milkshake','70'))

c.execute("insert into Menu values('%s','%s')" %('Blue Lagoon','80'))

c.execute("insert into Menu values('%s','%s')" %('Ginger & Mint','90'))

c.execute("insert into Menu values('%s','%s')" %('Fruit Punch','90'))

c.execute("insert into Menu values('%s','%s')" %('Lemon & Blackcurrant','80'))

c.execute("insert into Menu values('%s','%s')" %('Double Scoop(Choco,Vanilla)','65'))

c.execute("insert into Menu values('%s','%s')" %('Choco Blast Icecream','80'))

c.execute("insert into Menu values('%s','%s')" %('Cassata','70'))

c.execute("insert into Menu values('%s','%s')" %('Malai Kulfi(Pista)','40'))

c.execute("insert into Menu values('%s','%s')" %('Gulab Jamoon','40'))

c.execute("insert into Menu values('%s','%s')" %('Rasgulla','40'))

conn.commit()

if qa=="":

r1=""

else:

c1=c.execute("select \* from Menu where Item=?",(A,))

for row in c1:

r1=row[1]

if qb=="":

r2=""

else:

c2=c.execute("select \* from Menu where Item=?",(B,))

for row in c2:

r2=row[1]

if qc=="":

r3=""

else:

c3=c.execute("select \* from Menu where Item=?",(C,))

for row in c3:

r3=row[1]

if qd=="":

r4=""

else:

c4=c.execute("select \* from Menu where Item=?",(D,))

for row in c4:

r4=row[1]

if qe=="":

r5=""

else:

c5=c.execute("select \* from Menu where Item=?",(E,))

for row in c5:

r5=row[1]

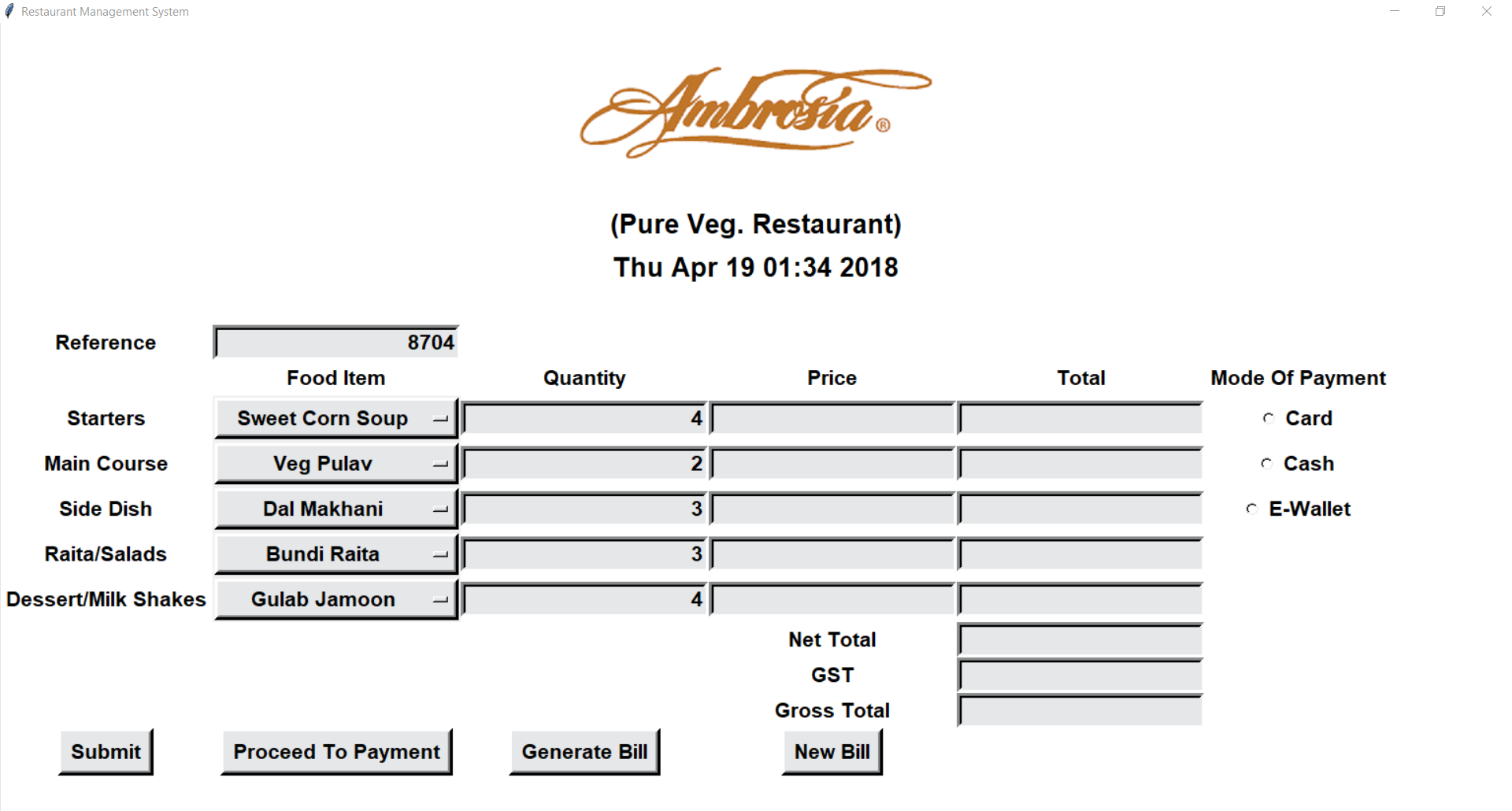
#Calculation Function call

CalcTotal(r1,r2,r3,r4,r5)

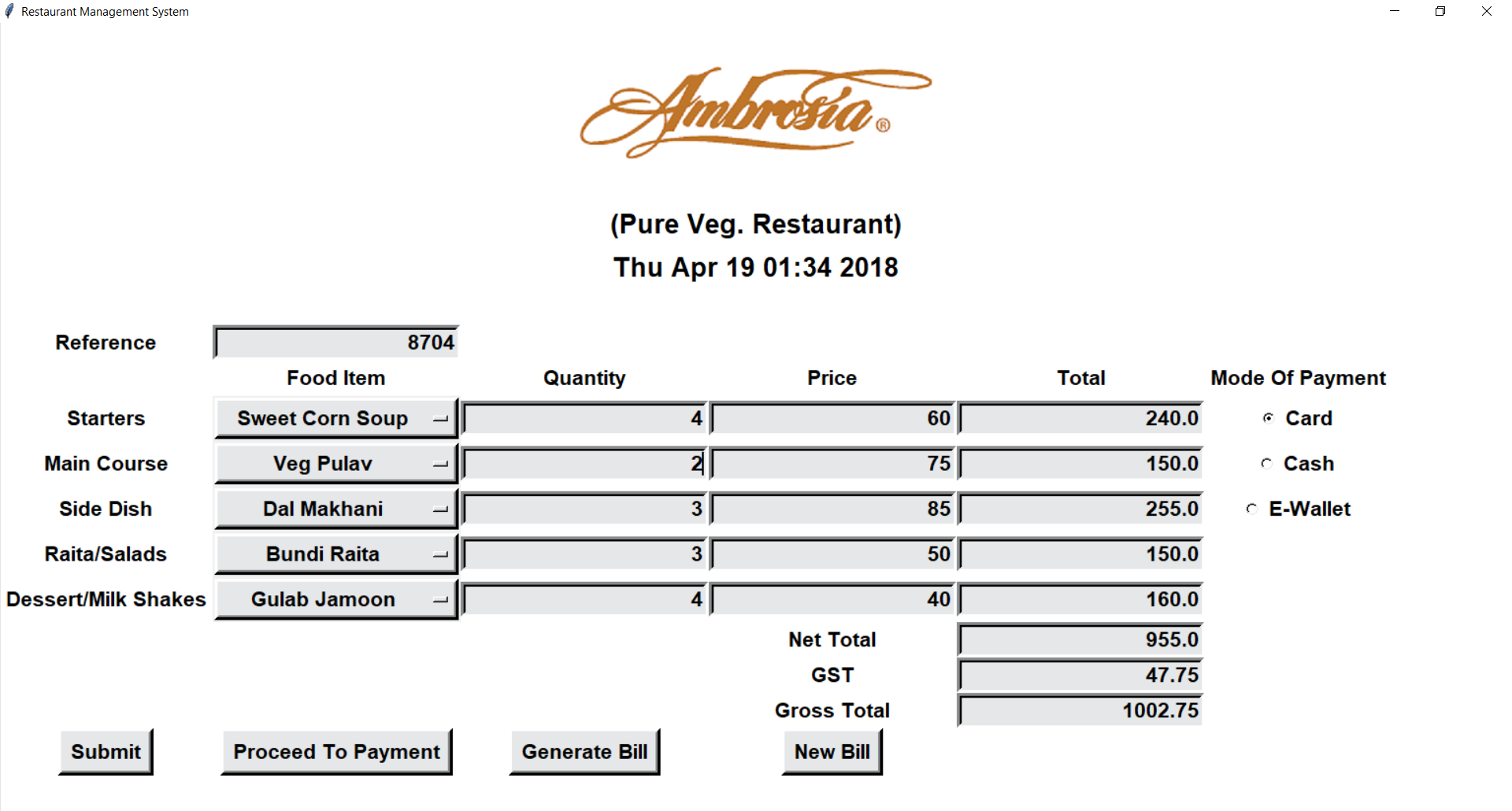
root.mainloop()

**Output:**

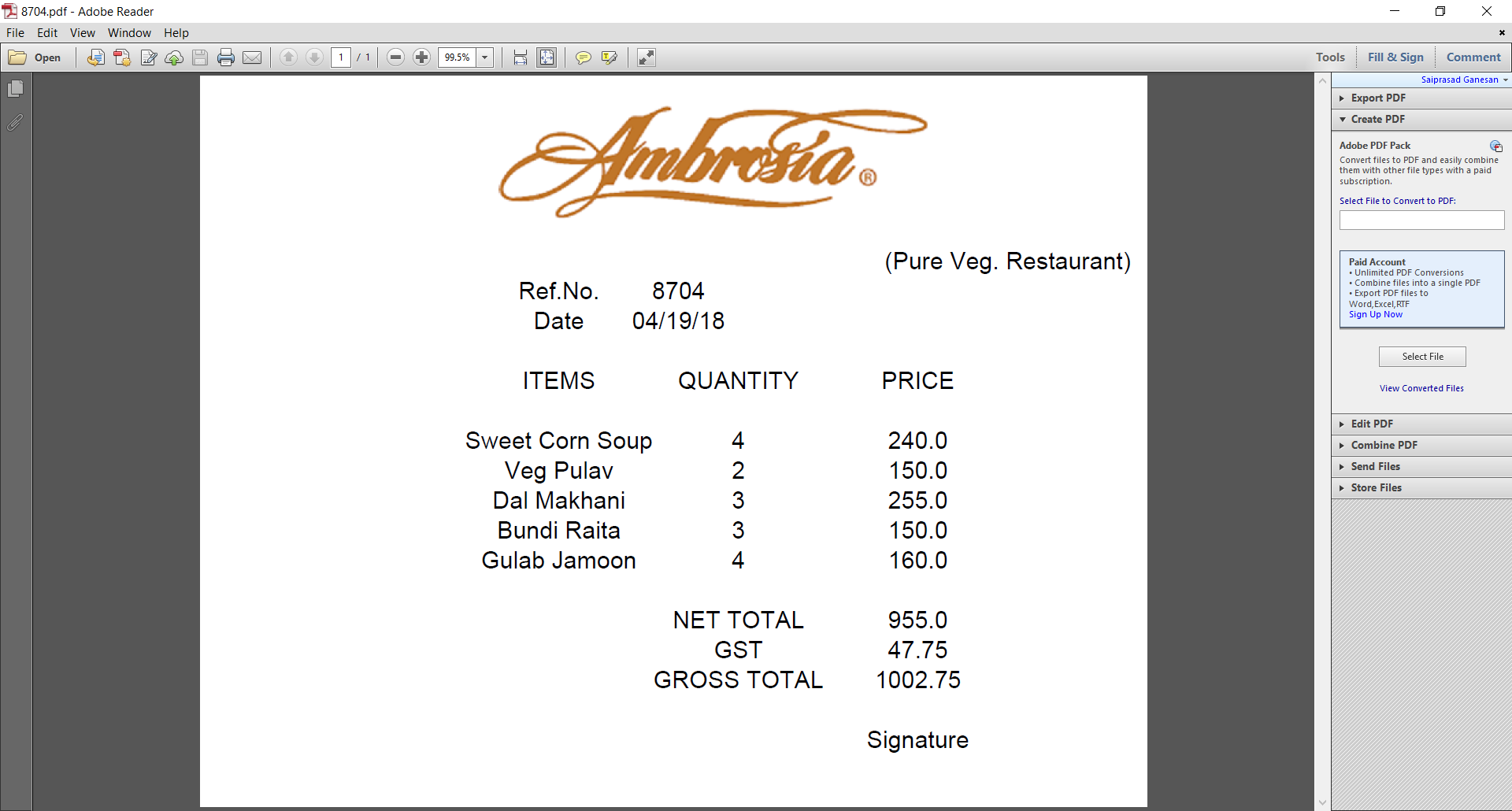
Before Clicking **Submit** Button:



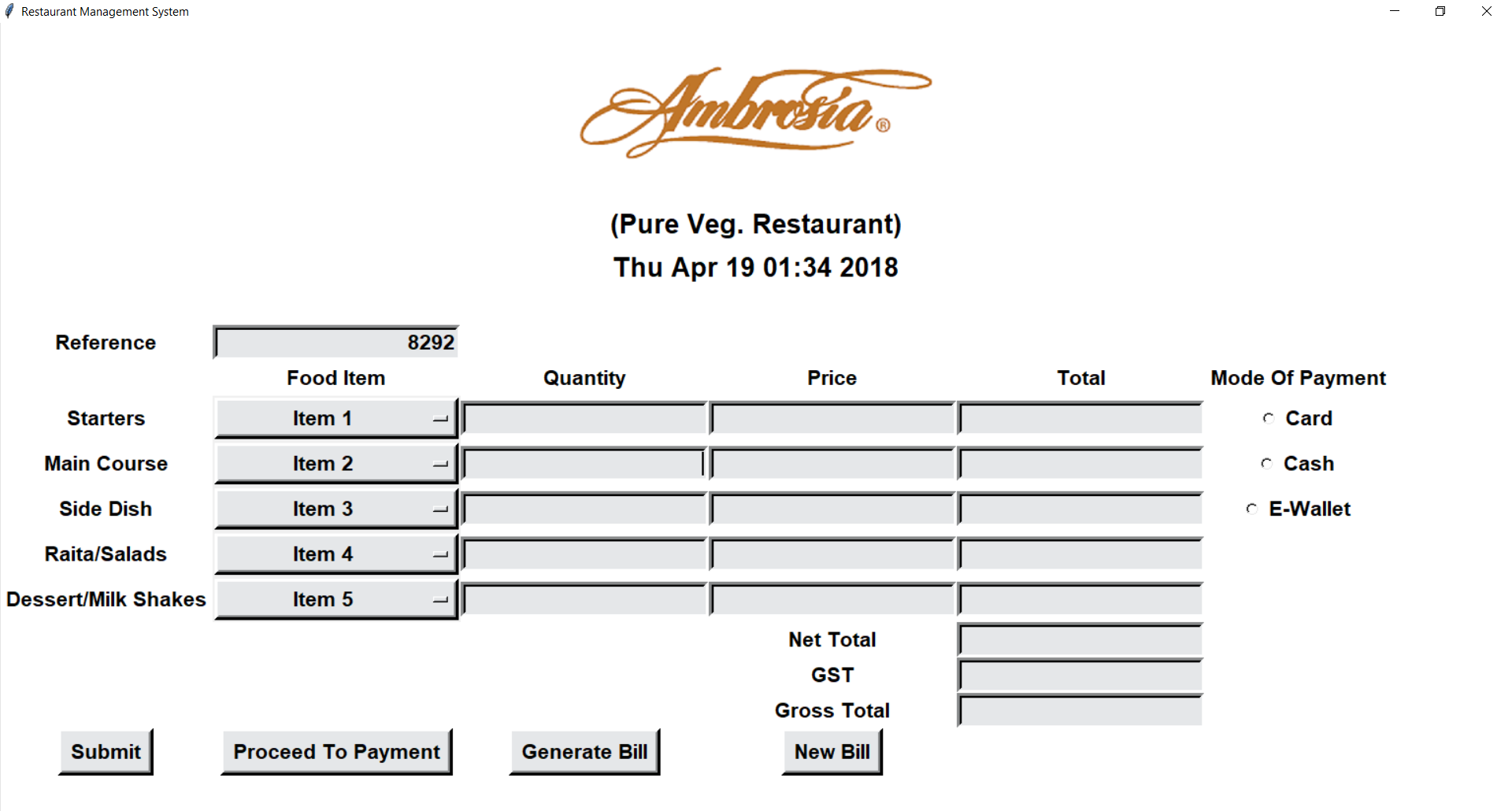
After Clicking **Submit** Button:



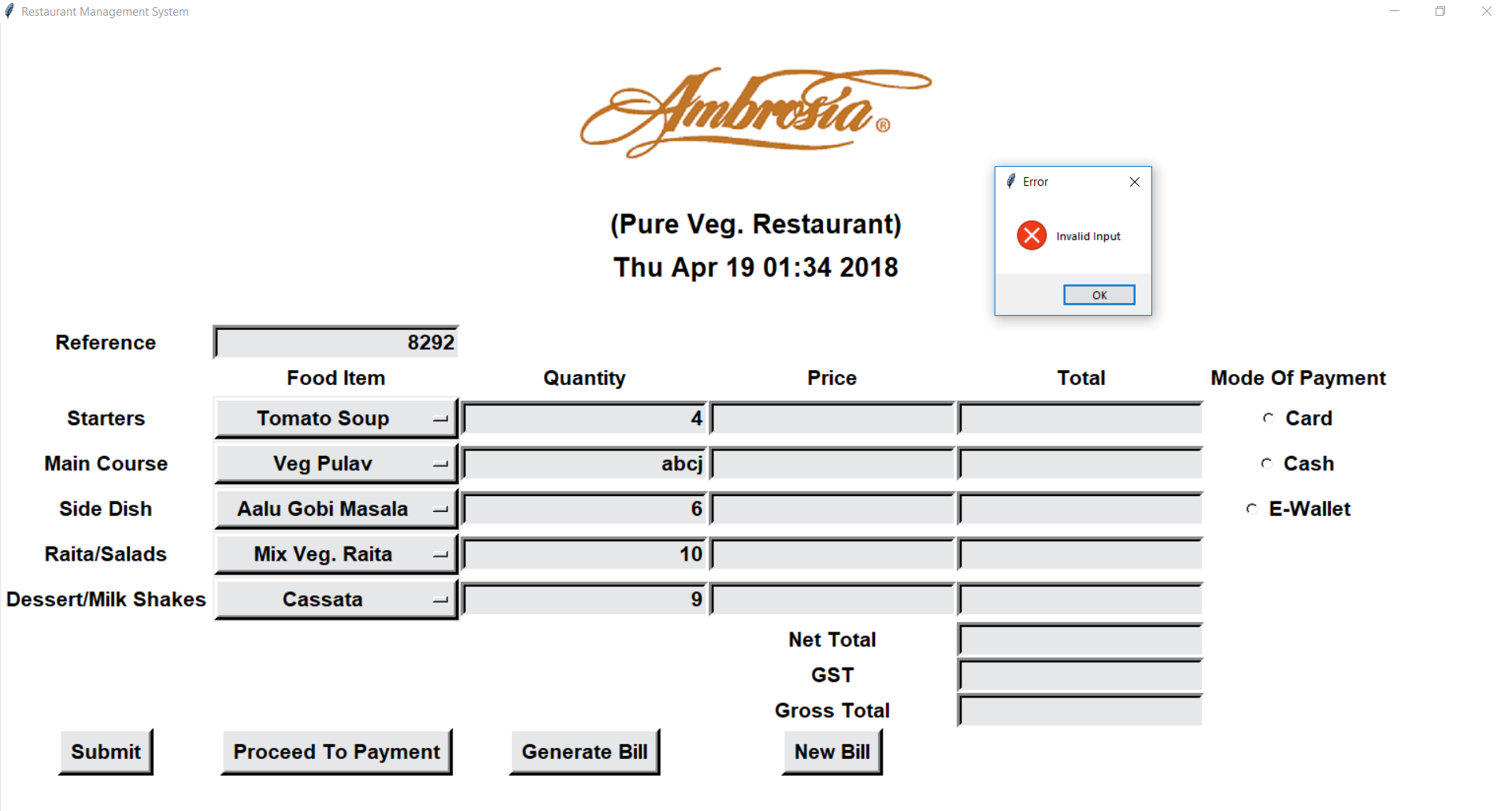
After Clicking **Generate Bill** Button:



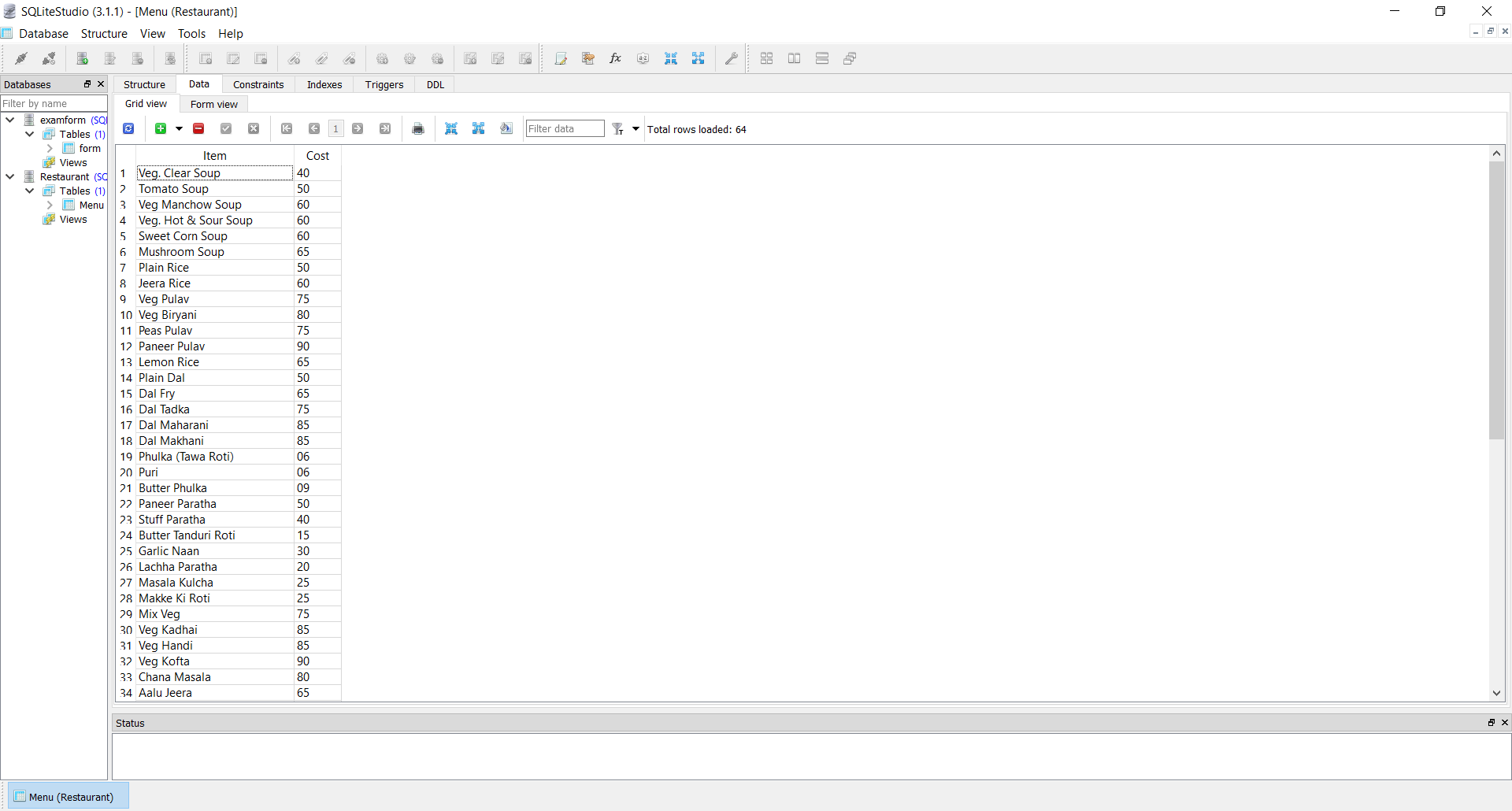
After Clicking **New Bill** Button:



**Error Pop-Up** on clicking **Submit** Button when non integer input is given:



The **Database Restaurant** created and a **table Menu** in it:



**Conclusion:**

We have successfully created a working prototype of Restaurant Management System in Python language with the limited knowledge we possess currently. Graphic User Interface (GUI) is a very interesting aspect of python which is also a lot of fun to work with as we see the results right in front of us and it is a key feature/concept required to make any application user friendly since it deals with the front end. We learnt a lot during the course of creating this system as to the various in-built modules available in tkinter package in python which is the package required to create GUI based applications in Python. Tkinter makes creating GUI really easy with various modules available in it. Also, Python overall as a language has various packages covering a huge range of features. There exist packages like Reportlab for python which enabled to us to create PDF which could be saved in the same directory and opened using the reference number which was generated by random function in python.

Database connectivity is another key aspect of python for which we have used sqlite database which is compatible with python. It is another module that can be imported on installing the package. We are able to successfully connect our GUI with an sqlite Database file which we created and were able to retrieve required data from it. The sqlite queries to create, update, delete tables etc. can be written in python thanks to its compatibility with sqlite.

Finally, I would conclude by saying that it was a great experience working as a team of three in limited time span and creating a descent system capable of handling activities of restaurant management like bill generation. We explored a lot and came to know a lot of new stuff which we didn’t before and hope to take this project to a higher level, make it official and hopefully present it in the market. Python is a highly feature rich, enhanced, high level, open source language with application in various fields in the present to a large extent in the future as well. We are and hope to keep exploring the various aspects of the language in the future.

**References:**

<https://www.tutorialspoint.com/python/index.htm>

<https://pythonspot.com/>

<https://stackoverflow.com/>

<https://pypi.org/>

<https://www.reportlab.com/docs/reportlab-userguide.pdf>

<https://www.google.com/>

<https://www.youtube.com/watch?v=hqC9tioGCi0>

<https://www.youtube.com/watch?v=Ei0fL6j8DtI>

<http://effbot.org/tkinterbook/>

<https://www.python-course.eu/tkinter_dialogs.php>

<https://bytes.com/topic/python/answers/657191-tkinter-resize-tkmessagebox>