### SOURCE CODE

# Importing modules

from tkinter import \*

import tkinter.font as font

import datetime as calender

from resmag import resmanag\_system\_backend as res

from tkinter import messagebox

from tkinter import ttk

from math import \*

# Setting up tkinter window

root = Tk()

root.geometry('1750x700')

root.title('Restaurant management system')

root.config(bg="wheat1")

root.iconbitmap(r'C:\Users\hello\Downloads\favicon\_io\favicon.ico')

# String variables

data1 = StringVar()

data2 = StringVar()

data3 = StringVar()

data4 = StringVar()

data5 = StringVar()

data6 = StringVar()

data7 = StringVar()

# Important setups

current\_date = calender.date.today()

data4.set(current\_date)

total\_list = []

grand\_total = 0

earning\_list = []

earning\_total = 0

early\_value = res.data\_all()

early\_list = []

for bill\_data in early\_value:

early\_list.append(bill\_data[0])

# Dishes in form of list

dishes = ["French Fries", "Burger", "Pizza", "Coffee", "Sandwich", "Noodles", "Rice", "Soup", "Curry", "Hot Dog"]

# Defining functions

def text\_insert():

"""This method inserts the default data into the bill area."""

text\_area.insert(END, '\t\t\t\tProgrammer Foundation Restaurant')

text\_area.insert(END, '\n\t\t\t\t Mountain view, California')

text\_area.insert(END, '\n\t\t\t\t contact:- 1234567890')

text\_area.insert(END, '\n'

'==========================================================='

'=================================')

def value\_generate():

"""This functions generates the bill of the customer"""

ph\_no = len(data3.get())

if data1.get() == '' or data2.get() == '' or ph\_no != 10:

messagebox.showerror("Error!", "Please enter all the fields correctly.")

else:

text\_area.insert(END, f'Bill Number:-{data1.get()}')

text\_area.insert(END, f'\nCustomer Name:- {data2.get()}')

text\_area.insert(END, f'\nCustomer Contact:- {data3.get()}')

text\_area.insert(END, f'\nDate:- {data4.get()}')

text\_area.insert(END, '\n'

'======================================================================'

'======================')

text\_area.insert(END, 'Product Name\t\t Quantity\t\t\t per cost\t\t\t Total')

text\_area.insert(END, '\n'

'===================================================================='

'========================')

def value\_clear():

"""This method deletes all the data from the entry boxes"""

data1.set('')

data2.set('')

data3.set('')

data4.set('')

data5.set('')

data6.set('')

data7.set('')

def value\_reset():

"""This function resets the bill area"""

global grand\_total

total\_list.clear()

grand\_total = 0

text\_area.delete('1.0', END)

text\_insert()

def value\_add():

"""This method adds up the order and shows in the bill area in the format of a table"""

if data6.get() == '' or data7.get() == '':

messagebox.showerror("Error", "Please enter all the details.")

else:

no\_qtn = int(data6.get())

no\_cost = int(data7.get())

total\_cost = no\_qtn \* no\_cost

total\_list.append(total\_cost)

text\_area.insert(END,

f'\n{data5.get()}\t\t {data6.get()}\t\t\t {data7.get()}\t\t {total\_cost}')

def value\_total():

"""This function makes a total of all the order"""

global grand\_total

for items in total\_list:

grand\_total += items

text\_area.insert(END, '\n'

'==============================================================='

'=============================')

text\_area.insert(END, f'\t\t\t\t\t Grand Total:- {grand\_total}')

text\_area.insert(END, '\n'

'=============================================================='

'==============================')

def value\_save():

"""This function saves the user data into the MySql database"""

ask\_confirm = messagebox.askyesno("Confirm", "Do you really want to save?")

if ask\_confirm > 0:

if data1.get() == '' or data4.get() == '' or data2.get() == '' or data5.get() == '' or data6.get() == '' or \

data7.get() == '':

messagebox.showerror("Error", "Please enter all the details.")

else:

try:

val\_date = data4.get().split('-')

val\_sql\_date = int(val\_date[0] + val\_date[1] + val\_date[2])

exact\_value = res.data\_insert(data1.get(), data2.get(), data3.get(), val\_sql\_date, data5.get(),

data6.get(),

data7.get())

if exact\_value == "successful":

messagebox.showinfo("Message", "Data has been saved successfully")

else:

messagebox.showerror("error", exact\_value)

except Exception as error:

messagebox.showerror("Error", error)

else:

return

def value\_restore():

"""This method takes the bill no as key and restores the respective records in the entry widgets from the

database"""

if data1.get() == '':

messagebox.showerror("error", "please enter the bill number!")

else:

key = data1.get()

try:

restore\_val = res.data\_search(key)

if restore\_val is None:

messagebox.showerror("error", "Details not found on this bill number!")

else:

print(restore\_val)

data2.set(restore\_val[1])

data3.set(restore\_val[2])

data4.set(restore\_val[3])

data5.set(restore\_val[4])

data6.set(restore\_val[5])

data7.set(restore\_val[6])

except Exception as error:

messagebox.showerror("error",error)

print(error)

def value\_delete():

"""This method takes the bill number as key and deletes the respective record from the database."""

if data1.get() == '':

messagebox.showerror("error", "please enter the bill number!")

else:

del\_confirm = messagebox.askyesno("Warning", "Do you really want to delete this record!")

if del\_confirm > 0:

try:

key = int(data1.get())

if key in early\_list:

res.data\_delete(key)

messagebox.showinfo("message", "Record has been successfully deleted.")

else:

messagebox.showwarning("Not Found", "Bill number not found! Enter a valid one.")

except Exception as error:

messagebox.showerror("error", error)

print(error)

else:

return

def value\_show\_all():

"""This method inserts all the records from the database into the bill area in a tabular format"""

global earning\_total

fetcher\_value = res.data\_all()

text\_area.insert(END, '\n'

'======================================================='

'=====================================')

text\_area.insert(END, 'bill no\t Cust Name\t Cust contact\t date\t item\t '

'quantity\t per cost')

text\_area.insert(END, '\n'

'===================================================='

'========================================')

for data in fetcher\_value:

earn\_multiple = data[5] \* data[6]

earning\_list.append(earn\_multiple)

text\_area.insert(END, f'\n{data[0]}\t {data[1]}\t {data[2]}\t {data[3]}\t {data[4]}\t '

f'{data[5]}\t\t\t {data[6]} ')

for items in earning\_list:

earning\_total += items

text\_area.insert(END, '\n'

'========================================================'

'====================================')

text\_area.insert(END, f'\t\t\t\t\t Total Earning:- ₹{earning\_total}')

text\_area.insert(END, '\n'

'========================================================'

'====================================')

def all\_exit():

wish1 = messagebox.askyesno("Confirm", "Do you want to exit?")

if wish1 > 0:

if data1.get() != '' or data2.get() != '' or data3.get() != '':

wish2 = messagebox.askyesno("Confirm", "Would you like to save the data before exiting?")

if wish2 > 0:

return

else:

root.destroy()

else:

root.destroy()

else:

return

# Creating frames and widgets

title\_label = Label(root, text='Restaurant management system', font=('Ariel', 35, 'bold'), bg='light gray', bd=8,

relief=GROOVE)

title\_label.pack(side=TOP, fill=X)

bill\_frame = LabelFrame(root, text='Enter Details', bg='light gray', font=("Ariel", 15), bd=7, relief=GROOVE)

bill\_frame.place(x=20, y=80, width=500, height=620)

# Input area labels

label\_data1 = Label(bill\_frame, text='Bill no.', font=('Ariel', 15), bg='light gray')

label\_data1.grid(row=0, column=0, sticky=W)

label\_data2 = Label(bill\_frame, text='Customer name', font=('Ariel', 15), bg='light gray')

label\_data2.grid(row=1, column=0, sticky=W)

label\_data3 = Label(bill\_frame, text='Customer contact', font=('Ariel', 15), bg='light gray')

label\_data3.grid(row=2, column=0, sticky=W)

label\_data4 = Label(bill\_frame, text='Date', font=('Ariel', 15), bg='light gray')

label\_data4.grid(row=3, column=0, sticky=W)

label\_data5 = Label(bill\_frame, text='Item purchased', font=('Ariel', 15), bg='light gray')

label\_data5.grid(row=4, column=0, sticky=W)

label\_data6 = Label(bill\_frame, text='Item quantity', font=('Ariel', 15), bg='light gray')

label\_data6.grid(row=5, column=0, sticky=W)

label\_data7 = Label(bill\_frame, text='cost of one', font=('Ariel', 15), bg='light gray')

label\_data7.grid(row=6, column=0, sticky=W)

# Entry widgets

entry\_data1 = Entry(bill\_frame, font=('Ariel', 15), bd=5, textvar=data1, bg="AntiqueWhite1")

entry\_data1.grid(row=0, column=1, padx=2, pady=2)

entry\_data2 = Entry(bill\_frame, font=('Ariel', 15), bd=5, textvar=data2, bg="AntiqueWhite1")

entry\_data2.grid(row=1, column=1, padx=2, pady=2)

entry\_data3 = Entry(bill\_frame, font=('Ariel', 15), bd=5, textvar=data3, bg="AntiqueWhite1")

entry\_data3.grid(row=2, column=1, padx=2, pady=2)

entry\_data4 = Entry(bill\_frame, font=('Ariel', 15), bd=5, textvar=data4, bg="AntiqueWhite1")

entry\_data4.grid(row=3, column=1, padx=2, pady=2)

combo\_data5 = ttk.Combobox(bill\_frame, values=dishes, width=23, font=('Ariel', 13), textvariable=data5)

combo\_data5.grid(row=4, column=1, padx=2, pady=2)

entry\_data6 = Entry(bill\_frame, font=('Ariel', 15), bd=5, textvar=data6, bg="AntiqueWhite1")

entry\_data6.grid(row=5, column=1, padx=2, pady=2)

entry\_data7 = Entry(bill\_frame, font=('Ariel', 15), bd=5, textvar=data7, bg="AntiqueWhite1")

entry\_data7.grid(row=6, column=1, padx=2, pady=2)

# Options area frame and buttons

button\_frame = LabelFrame(bill\_frame, bd=5, text='Options', bg='light gray', font=('Ariel', 15))

button\_frame.place(x=20, y=275, width=390, height=305)

button1 = Button(button\_frame, bd=2, text='Add', font=('Ariel', 12), width=12, height=3, command=value\_add,

bg="bisque2")

button1.grid(row=0, column=0, padx=4, pady=2)

button2 = Button(button\_frame, bd=2, text='Generate', font=('Ariel', 12), width=12, height=3, command=value\_generate,

bg="bisque2")

button2.grid(row=0, column=1, padx=4, pady=2)

button3 = Button(button\_frame, bd=2, text='Clear', font=('Ariel', 12), width=12, height=3, command=value\_clear,

bg="bisque2")

button3.grid(row=0, column=2, padx=4, pady=2)

button4 = Button(button\_frame, bd=2, text='Total', font=('Ariel', 12), width=12, height=3, command=value\_total,

bg="bisque2")

button4.grid(row=1, column=0, padx=4, pady=2)

button5 = Button(button\_frame, bd=2, text='Reset', font=('Ariel', 12), width=12, height=3, command=value\_reset,

bg="bisque2")

button5.grid(row=1, column=1, padx=4, pady=2)

button6 = Button(button\_frame, bd=2, text='Save', font=('Ariel', 12), width=12, height=3, command=value\_save,

bg="bisque2")

button6.grid(row=1, column=2, padx=4, pady=2)

button7 = Button(button\_frame, bd=2, text='Restore', font=('Ariel', 12), width=12, height=3, command=value\_restore,

bg="bisque2")

button7.grid(row=2, column=0, padx=4, pady=2)

button8 = Button(button\_frame, bd=2, text='Delete', font=('Ariel', 12), width=12, height=3, command=value\_delete,

bg="bisque2")

button8.grid(row=2, column=1, padx=4, pady=2)

button9 = Button(button\_frame, bd=2, text='Show all', font=('Ariel', 12), width=12, height=3, command=value\_show\_all,

bg="bisque2")

button9.grid(row=2, column=2, padx=4, pady=2)

button10 = Button(button\_frame, bd=2, text='Exit', font=('Ariel', 12), width=40, height=2, command=all\_exit,

bg="bisque2")

button10.grid(row=3, column=0, padx=4, pady=2, columnspan=3)

# Creating calculator

calc\_frame = Frame(root, bd=3, bg='light gray', relief=GROOVE)

calc\_frame.place(x=550, y=80, width=780, height=400)

# defining applicable function

def click(event):

text = event.widget.cget("text")

if text == "=":

"""when = is pressed the expression will be evaluated """

if f\_value.get().isdigit():

value = int(f\_value.get())

else:

try:

value = eval(input\_field.get())

except Exception as error:

print(error)

value = "Error"

text\_area.insert(END, f"\n{f\_value.get()} = {value}")

f\_value.set(value)

input\_field.update()

elif text == "clear":

"""when clear is pressed the input field will be reset """

f\_value.set("")

input\_field.update()

else:

f\_value.set(f\_value.get() + text)

input\_field.update()

Font1 = font.Font(family="Times", size=15, weight="bold")

f\_value = StringVar()

f\_value.set("")

input\_field = Entry(calc\_frame, textvar=f\_value, width=20, borderwidth=5, bg="AntiqueWhite1", font=Font1)

input\_field.grid(row=0, column=2, columnspan=3, padx=20, pady=10)

# numeral buttons

button1 = Button(calc\_frame, text="1", padx=40, pady=20, bg="MistyRose2", font=Font1)

button2 = Button(calc\_frame, text="2", padx=55, pady=20, bg="MistyRose2", font=Font1)

button3 = Button(calc\_frame, text="3", padx=45, pady=20, bg="MistyRose2", font=Font1)

button4 = Button(calc\_frame, text="4", padx=40, pady=20, bg="MistyRose2", font=Font1)

button5 = Button(calc\_frame, text="5", padx=55, pady=20, bg="MistyRose2", font=Font1)

button6 = Button(calc\_frame, text="6", padx=45, pady=20, bg="MistyRose2", font=Font1)

button7 = Button(calc\_frame, text="7", padx=40, pady=20, bg="MistyRose2", font=Font1)

button8 = Button(calc\_frame, text="8", padx=55, pady=20, bg="MistyRose2", font=Font1)

button9 = Button(calc\_frame, text="9", padx=45, pady=20, bg="MistyRose2", font=Font1)

button0 = Button(calc\_frame, text="0", padx=55, pady=20, bg="MistyRose2", font=Font1)

# Operator buttons

button\_plus = Button(calc\_frame, text="+", padx=50, pady=20, bg="MistyRose2", font=Font1)

button\_minus = Button(calc\_frame, text="-", padx=50, pady=20, bg="MistyRose2", font=Font1)

button\_into = Button(calc\_frame, text="\*", padx=50, pady=20, bg="MistyRose2", font=Font1)

button\_divide = Button(calc\_frame, text="/", padx=50, pady=20, bg="MistyRose2", font=Font1)

button\_equal = Button(calc\_frame, text="=", padx=40, pady=20, bg="MistyRose2", font=Font1)

button\_clear = Button(calc\_frame, text="clear", padx=25, pady=20, bg="MistyRose2", font=Font1)

button\_point = Button(calc\_frame, text=".", padx=45, pady=20, bg="MistyRose2", font=Font1)

button\_two\_zero = Button(calc\_frame, text="00", padx=45, pady=20, bg="MistyRose2", font=Font1)

button\_sin = Button(calc\_frame, text="sin", padx=55, pady=20, bg="MistyRose2", font=Font1)

button\_cos = Button(calc\_frame, text="cos", padx=55, pady=20, bg="MistyRose2", font=Font1)

button\_tan = Button(calc\_frame, text="tan", padx=55, pady=20, bg="MistyRose2", font=Font1)

button\_br1 = Button(calc\_frame, text="(", padx=65, pady=20, bg="MistyRose2", font=Font1)

button\_br2 = Button(calc\_frame, text=")", padx=65, pady=20, bg="MistyRose2", font=Font1)

button\_sqrt = Button(calc\_frame, text="sqrt", padx=41, pady=20, bg="MistyRose2", font=Font1)

button\_exp = Button(calc\_frame, text="exp", padx=43, pady=20, bg="MistyRose2", font=Font1)

button\_log10 = Button(calc\_frame, text="log10", padx=35, pady=20, bg="MistyRose2", font=Font1)

button\_pi = Button(calc\_frame, text="pi", padx=50, pady=20, bg="MistyRose2", font=Font1)

button\_fac = Button(calc\_frame, text="factorial", padx=21, pady=20, bg="MistyRose2", font=Font1)

# \_\_binding buttons\_\_

button1.grid(row=1, column=1)

button1.bind("<Button-1>", click)

button2.grid(row=1, column=2)

button2.bind("<Button-1>", click)

button3.grid(row=1, column=3)

button3.bind("<Button-1>", click)

button4.grid(row=2, column=1)

button4.bind("<Button-1>", click)

button5.grid(row=2, column=2)

button5.bind("<Button-1>", click)

button6.grid(row=2, column=3)

button6.bind("<Button-1>", click)

button7.grid(row=3, column=1)

button7.bind("<Button-1>", click)

button8.grid(row=3, column=2)

button8.bind("<Button-1>", click)

button9.grid(row=3, column=3)

button9.bind("<Button-1>", click)

button0.grid(row=4, column=2)

button0.bind("<Button-1>", click)

button\_plus.grid(row=0, column=5)

button\_plus.bind("<Button-1>", click)

button\_minus.grid(row=1, column=5)

button\_minus.bind("<Button-1>", click)

button\_into.grid(row=2, column=5)

button\_into.bind("<Button-1>", click)

button\_divide.grid(row=3, column=5)

button\_divide.bind("<Button-1>", click)

button\_two\_zero.grid(row=4, column=5)

button\_two\_zero.bind("<Button-1>", click)

button\_equal.grid(row=4, column=1)

button\_equal.bind("<Button-1>", click)

button\_clear.grid(row=0, column=1)

button\_clear.bind("<Button-1>", click)

button\_point.grid(row=4, column=3)

button\_point.bind("<Button-1>", click)

button\_sin.grid(row=0, column=6)

button\_sin.bind("<Button-1>", click)

button\_cos.grid(row=1, column=6)

button\_cos.bind("<Button-1>", click)

button\_tan.grid(row=2, column=6)

button\_tan.bind("<Button-1>", click)

button\_br1.grid(row=3, column=6)

button\_br1.bind("<Button-1>", click)

button\_br2.grid(row=4, column=6)

button\_br2.bind("<Button-1>", click)

button\_sqrt.grid(row=0, column=0)

button\_sqrt.bind("<Button-1>", click)

button\_exp.grid(row=1, column=0)

button\_exp.bind("<Button-1>", click)

button\_log10.grid(row=2, column=0)

button\_log10.bind("<Button-1>", click)

button\_pi.grid(row=3, column=0)

button\_pi.bind("<Button-1>", click)

button\_fac.grid(row=4, column=0)

button\_fac.bind("<Button-1>", click)

# Creating text area

text\_frame = LabelFrame(root, text='Bill Area', font=('Ariel', 15), bg='light gray', bd=8, relief=GROOVE)

text\_frame.place(x=550, y=485, width=778, height=210)

y\_scroll = Scrollbar(text\_frame, orient=VERTICAL)

text\_area = Text(text\_frame, bg='white', yscrollcommand=y\_scroll.set)

y\_scroll.config(command=text\_area.yview)

y\_scroll.pack(side=RIGHT, fill=Y)

text\_area.pack(fill=BOTH, expand=TRUE)

text\_insert()

root.mainloop()