A
Mini Project Report
On

Restaurant Billing System

For
Partial fulfillment of award of the
B.Tech. Degree
in
Information Technology

(Session 2023-24)

Name of Supervisor

Mr. Abhishek Choudhary

Assistant Professor

Team Member(s)

Saurabh Verma (2201920139016)
Madhuresh (2201920139010)

Department of Information Technology

G.L. Bajaj Institute of Technology and Management

Greater Noida-201306

GL BAJAJ
Institute of Technology & Management

Department of Information Technology

Declaration

I/We herewith declare that the project work conferred during this report entitled "**Restaurant Billing System"** , in partial fulfillment of the necessity for the award of the degree of Bachelor of Technology in Information Technology, submitted to A.P.J. Abdul Kalam Pradesh Technical University, Uttar Pradesh, is an authentic record of my/our own work distributed in Department of Information Technology & Engineering, G.L. Bajaj Institute of Technology & Management, Greater Noida. It contains no material antecedently printed or written by another person except wherever due acknowledgement has been created within the text. The project work reported during this report has not been submitted by me/us for award of the other degree or certification.

Signature:

Name: Saurabh Verma

Roll No: 2201920139016

Signature:

Name: Madhuresh

Roll No: 2201920139010

Date:

Place: Greater Noida

GL BAJAJ
Institute of Technology & Management

Department of Information Technology

Certificate

This is to certify that Project Report entitled "**Restaurant Billing System"** that is submitted by **Saurabh Verma (2201920139016), Madhuresh (2201920139010)** in partial fulfillment of the necessity for the award of degree B. Tech. in Department of Information Technology of Abdul Kalam Technical University, are record of the candidate own work distributed by him below my/our oversight. The matter embodied during this thesis is original and has not been submitted for the award of the other degree.

Date:

Mr. Abhishek Choudhary
(Assistant Professor)

Dr. P.C. Vashist
Head of Department

GL BAJAJ
Institute of Technology & Management

Department of Information Technology

Acknowledgement

We would like to express our sincere thanks to our project supervisor **Mr. Abhishek Choudhary** and our Head of department **Dr. P.C. Vashist** for their invaluable guidance and suggestions. This project helped to us to understand the concept of machine learning and IOT. This project enriches our knowledge and experience of working in a team and a live project.

Lastly, we would like to thank all the faculties for providing their valuable time whenever needed for helping us carry on with our project.

TABLE OF CONTENT

ABSTRACT

A billing system can be very useful within a business environment. Instead of making bills manually or to sum up the total manually, it is very much time consuming and also may have some human errors.
like adding up the wrong total or adding wrong items into the bill. When making a handwritten bill the owner and customer both have to repeatedly check the total, items added, etc. It also sometimes results in to a Bad Impression towards the Restaurant from a Customer. Ideally, user should be able to generate bill without any mistakes and quickly, enabling them to fasten or improve their process. To overcome this problem, we have come up with this project, that is, Restaurant Billing System Using Python.

A simple project based on Restaurant/Cafe Billing System which uses Python Language with Tkinter Library for GUI. Following Python with Tkinter Library project contains the least, but important features which can be in use for the first-year IT students for their college projects. It has features that will allow all the users to interact in a way that the restaurant manager interacts with their customers regarding their billing payments. This system as well as the python application's concept is all clear, it's the same as real-life scenarios and well-implemented on it.

INTRODUCTION

Restaurant Billing System Using Python can be very useful within a business environment. Instead of doing manual work for making up a bill at Restaurant, which gets tiring and time consuming, you can generate a bill including tax and service charges in just few clicks. When making up a bill manually at a Restaurant may contain some human errors like adding wrong items into the bill or summing up their total also may end up wrong, it also sometimes results into a Bad Impression towards the Restaurant from a Customer. Ideally, user should be able to generate bill without any mistakes and quickly, enabling them to fasten or improve their process. To overcome this problem, we have come up with this project, that is, Restaurant Billing System Using Python.

The Restaurant Billing System Using Python is very useful to small business or restaurant or cafe or food truck owners. This helps the owner to fasten the process which is bug free and easy to use. It also has a calculator to ease the use of the user. This project firstly has the menu and then adds up the selected items by customer and sums up the total of all items adds tax and service charges and displays total. To perform any other operation like division, multiplication, etc .

Moving on, this restaurant/cafe system project in Python focuses mainly on dealing with customer's payment details with their respective food orders and amounts. Also, the system allows the selection of food and drink items for calculation and entering the quantities. But here, the project only contains Admin Panel. In an overview of this app, the system user has to select a particular food and drink item, enter a certain quantity and generate the total cost. In addition, the system generates the total bill amount with tax. Besides, the system also generates a bill receipt with a reference number. Additionally, the system also contains a mini calculator where the user can perform simple mathematics for calculation too. So with it, this simple project can perform all the important tasks for calculations of the total bill amount of the customer.

Last but not least, a clean and simple GUI is presented with simple color combinations for a greater user experience while using this restaurant billing system project in Python. For its UI elements, a standard GUI library; Tkinter is on board. Presenting a new restaurant/cafe billing system in Python project which includes a user panel that contains all the essential features to follow up, and a knowledgeable resource for learning purposes.

PROBLEM DEFINITION

The Restaurant industry is enlarging rapidly and restaurant owners are keen to improve every section of their business. Though much attention is paid to digitalizing the restaurant management and the menu, but not many business owners realize the importance of applying digital billing software in the restaurant. The customers' experience at your restaurant includes the billing and payment experiences too. Billing software provides some exclusive features that ease up the restaurant services. It upgrades the billing process and uplift the customers' experience. It enables customers to pay bills more easily. The software can generate detailed bills that eliminate the need to calculate bills separately when the guests wish to know total GST amount. Apart from billing, the software enables you to organize a number of processes at the restaurant. It makes your system more effective and helps you provide faster and easy services to the customers. So many times, customers leave unhappy due to improper billing. When the crowd is vast in the restaurant, it might take you some time to generate manual bills that may leave your customers unsatisfied. This is where the automated billing system can be used. It generates digital bills automatically and allows customers to make quick payments.

PURPOSE

The Restaurant Management System helps the restaurant manager to manage the restaurant more effectively and efficiently by computerizing meal ordering, billing and inventory control.

EXISTING SYSTEM

There is always a need of a sytem that will perform easy billing calculation in a grocery store. This system will reduce the manual operation required to maintain all the bills. And also generates bill receipt with unique bill number.

EXISTING SYSTEM DRAWBACKS

- Time consuming.
- Human error.
- No backup records in case of loss or damage.
- May require specialized knowledge to maintain.

PROPOSED SYSTEM

Since many restaurant or café owners make bills for their customers manually with a pen paper. This sometimes results into an error of total or wrong items added or some items missing in bill or extra items added. This may end up by building up a bad impression of customer towards the Café or restaurant. So, to overcome this problem we've come up with this helpful project named Restaurant Billing System Using Python. We all love going to cafes or restaurants but when it takes time for them to make a bill or if they Make wrong bill then it's time consuming. So, to avoid all such chaos our project will help in All possible terms.

PROPOSED SYSTEM ADVANTAGES

1. Improve Customer Relationships

2. Tracking Sales

3. Automatic Analysis

4. Employee Satisfaction

5. Reduction of Errors

6. Enhanced Productivity

7. Effective Use of Human Resource

8. Reduce the paper work

9. Customer satisfies

SYSTEM REQUIRMENTS AND SPECIFICATIONS

Here we are including the software's and hardware's used for developing the project and implementing the project

A. Software Requirements

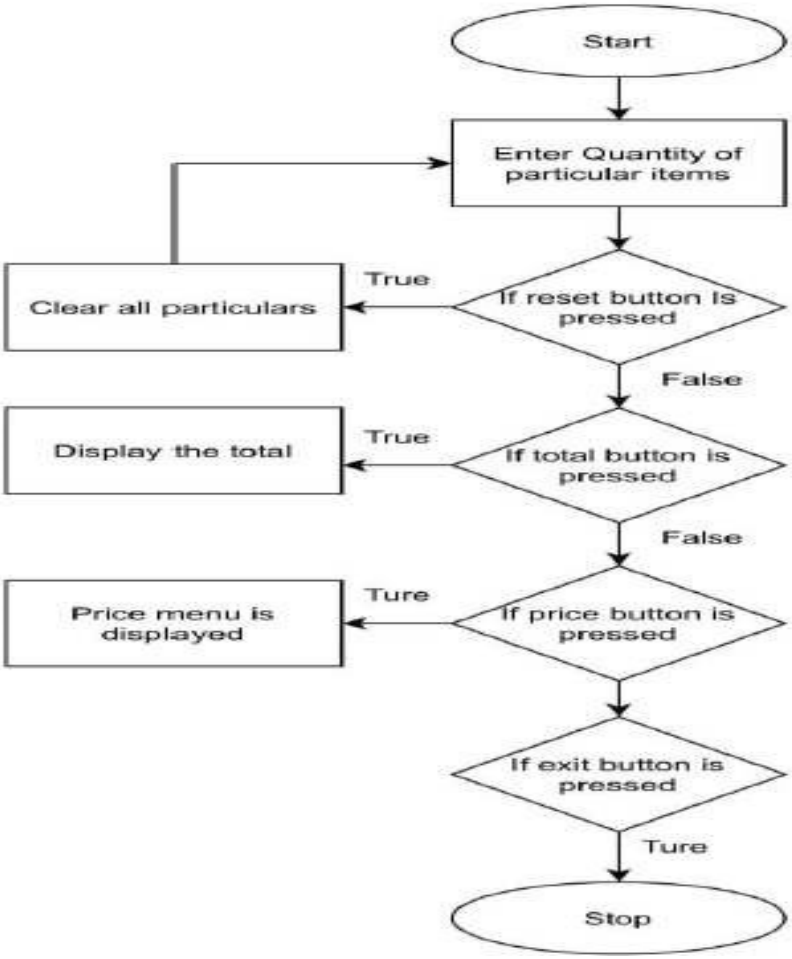
1. Python 3.9
2. Notepad
3. Any OS

B. Hardware Requirements

1. 2 GB RAM or above
2. Intel i3 Processor or above
3. 32 Bit System or above

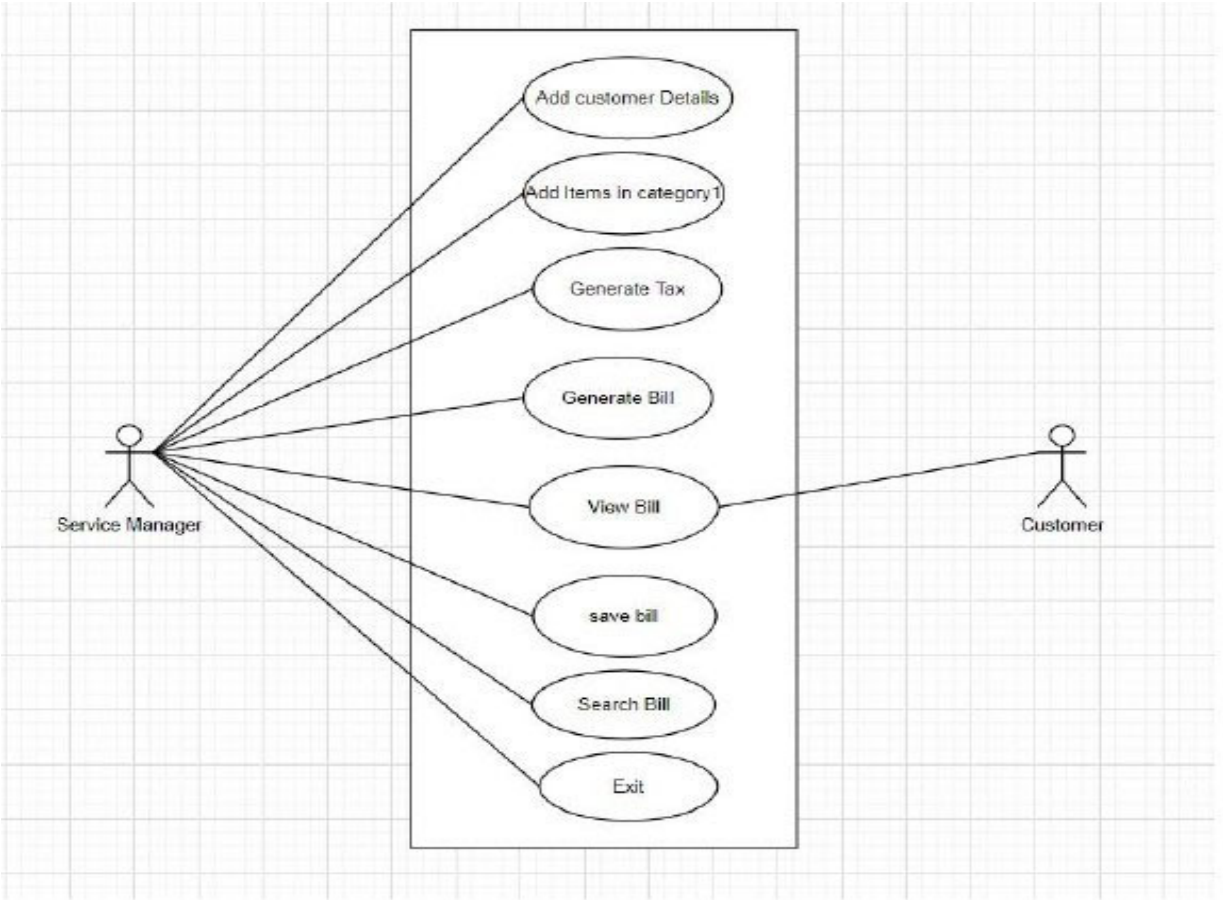
IMPLEMENTATION

ARCHITECTURE OR FLOW DIAGRAM



USE CASE DIAGRAM

This is use case diagram is a graphical depiction of the interactions among the elements of Restaurant Billing System.It represents the methodology used in system analysis to identify,clarify and organize system requirements of restaurant billing System.The main actors of Restaurant Billing System in this use case diagram are :-Service Manager and Customer who performs different types of use cases such as adding customer details,adding quantities for the items,generating tax,generates the bill in bill area,view bill,save the bill,searches the bill .



SOURCE CODE

```
from tkinter import *
import random
class Bill_App:
    def __init__(self,root):
        self.root = root
        self.root.geometry("1300x700+0+0")
        self.root.maxsize(width = 1280,height = 700)
        self.root.minsize(width = 1280,height = 700)
        self.root.title("Billing Software")

        #=====================Variables=========================#
        self.cus_name = StringVar()
        self.c_phone = StringVar()
        #For Generating Random Bill Numbers
        x = random.randint(1000,9999)
        self.c_bill_no = StringVar()
        #Seting Value to variable
        self.c_bill_no.set(str(x))

        self.Gobi_Manchureian= IntVar()
        self.Chilli_Chicken = IntVar()
        self.Chicken_Lolipop = IntVar()
        self.Paneer_Tikkas = IntVar()
        self.Chicken_Curry_Rice = IntVar()
        self.Mutton_Curry_Rice = IntVar()
        self.Prawn_Curry_Rice = IntVar()
```

```
        self.Egg_Curry_Rice = IntVar()
        self.Butter_Naan = IntVar()
        self.Butter_Roti = IntVar()
        self.Cheese_Garlic_Naan = IntVar()
        self.Keema_Naan = IntVar()
        self.Channa_Masala = IntVar()
        self.Kadhai_Paneer = IntVar()
        self.Mushroom_Curry = IntVar()
        self.Mutton_Chicken = IntVar()
        self.Mutton_Biryani = IntVar()
        self.Chicken_Biryani = IntVar()
        self.EggVeg_Biryani = IntVar()
        self.JeeraSteamed_Rice = IntVar()
        self.Water_Bottle = IntVar()
        self.SaltSweet_Lassi = IntVar()
        self.Juices = IntVar()
        self.Lemon_SodaWater = IntVar()
        self.total_bill= StringVar()



        #==================================
        bg_color = "#8B7D6B"
        fg_color = "white"
        lbl_color = 'white'
        #Title of App
            title = Label(self.root,text = "RESTAURANT BILLING SYSTEM ",bd = 12,relief =     GROOVE,fg =
fg_color,bg = bg_color,font=("times new roman",30,"bold"),pady = 3).pack(fill = X)
```

```
#==========Customers Frame==========#

        F1 = LabelFrame(text = "Customer Details",font = ("time new roman",12,"bold"),fg = "gold",bg = bg_color,relief = GROOVE,bd = 10)

    F1.place(x = 0,y = 80,relwidth = 1)


    #===============Customer Name===========#

            cname_lbl = Label(F1,text="Customer Name",bg = bg_color,fg = fg_color,font=("times new roman",15,"bold")).grid(row = 0,column = 0,padx = 10,pady = 5)

    cname_en = Entry(F1,bd = 8,relief = GROOVE,textvariable = self.cus_name)

    cname_en.grid(row = 0,column = 1,ipady = 4,ipadx = 30,pady = 5)


    #================Customer Phone==============#

            cphon_lbl = Label(F1,text = "Phone No",bg = bg_color,fg = fg_color,font = ("times new roman",15,"bold")).grid(row = 0,column = 2,padx = 20)

    cphon_en = Entry(F1,bd = 8,relief = GROOVE,textvariable = self.c_phone)

    cphon_en.grid(row = 0,column = 3,ipady = 4,ipadx = 30,pady = 5)


    #====================Customer Bill No==================#

    cbill_lbl = Label(F1,text = "Bill No.",bg = bg_color,fg = fg_color,font = ("times new roman",15,"bold"))

    cbill_lbl.grid(row = 0,column = 4,padx = 20)

    cbill_en = Entry(F1,bd = 8,relief = GROOVE,textvariable = self.c_bill_no)

    cbill_en.grid(row = 0,column = 5,ipadx = 30,ipady = 4,pady = 5)


    #==================Straters====================#

     F2 = LabelFrame(self.root,text = 'Straters',bd = 10,relief = GROOVE,bg = bg_color,fg = "gold",font = ("times new roman",13,"bold"))

    F2.place(x = 5,y = 180,width = 325,height = 220)
```

```
        gobi_lbl = Label(F2,font = ("times new roman",12,"bold"),fg = lbl_color,bg = bg_color,text = "Gobi Manchureian")

    gobi_lbl.grid(row = 0,column = 0,padx = 5,pady = 10)

    gobi_en = Entry(F2,bd = 8,relief = GROOVE,textvariable = self.Gobi_Manchureian)

    gobi_en.grid(row = 0,column = 1,ipady = 2,ipadx = 1)

    face_lbl = Label(F2,font = ("times new roman",12,"bold"),fg = lbl_color,bg = bg_color,text = "Chilli Chicken")

    face_lbl.grid(row = 1,column = 0,padx = 5,pady = 10)

    face_en = Entry(F2,bd = 8,relief = GROOVE,textvariable = self.Chilli_Chicken)

    face_en.grid(row = 1,column = 1,ipady = 2,ipadx = 1)

        wash_lbl = Label(F2,font = ("times new roman",12,"bold"),fg = lbl_color,bg = bg_color,text = "Chicken Lolipop")

    wash_lbl.grid(row = 2,column = 0,padx = 5,pady = 10)

    wash_en = Entry(F2,bd = 8,relief = GROOVE,textvariable = self.Chicken_Lolipop)

    wash_en.grid(row = 2,column = 1,ipady = 2,ipadx = 1)

    hair_lbl = Label(F2,font = ("times new roman",12,"bold"),fg = lbl_color,bg = bg_color,text = "Paneer Tikkas")

    hair_lbl.grid(row = 3,column = 0,padx = 5,pady = 10)

    hair_en = Entry(F2,bd = 8,relief = GROOVE,textvariable = self.Paneer_Tikkas)

    hair_en.grid(row = 3,column = 1,ipady = 2,ipadx = 1)




    F2 = LabelFrame(self.root,text = 'Specials',bd = 10,relief = GROOVE,bg = bg_color,fg = "gold",font = ("times new roman",13,"bold"))

    F2.place(x = 5,y = 400,width = 325,height = 380)



     gobi_lbl = Label(F2,font = ("times new roman",12,"bold"),fg = lbl_color,bg = bg_color,text = "Chicken Curry Rice")

    gobi_lbl.grid(row = 0,column = 0,padx = 5,pady = 10)

    gobi_en = Entry(F2,bd = 8,relief = GROOVE,textvariable = self.Chicken_Curry_Rice)

    gobi_en.grid(row = 0,column = 1,ipady = 2,ipadx = 1)
```

```
face_lbl = Label(F2,font = ("times new roman",12,"bold"),fg = lbl_color,bg = bg_color,text = "Mutton Curry Rice")

face_lbl.grid(row = 1,column = 0,padx = 5,pady = 10)

face_en = Entry(F2,bd = 8,relief = GROOVE,textvariable = self.Mutton_Curry_Rice)

face_en.grid(row = 1,column = 1,ipady = 2,ipadx = 1)

wash_lbl = Label(F2,font = ("times new roman",12,"bold"),fg = lbl_color,bg = bg_color,text = "Prawn Curry Rice")

wash_lbl.grid(row = 2,column = 0,padx = 5,pady = 10)

wash_en = Entry(F2,bd = 8,relief = GROOVE,textvariable = self.Prawn_Curry_Rice)

wash_en.grid(row = 2,column = 1,ipady = 2,ipadx = 1)

hair_lbl = Label(F2,font = ("times new roman",12,"bold"),fg = lbl_color,bg = bg_color,text = "Egg Curry Rice")

hair_lbl.grid(row = 3,column = 0,padx = 5,pady = 10)

hair_en = Entry(F2,bd = 8,relief = GROOVE,textvariable = self.Egg_Curry_Rice)

hair_en.grid(row = 3,column = 1,ipady = 2,ipadx = 1)




F2 = LabelFrame(self.root,text = 'Main Course-INDIAN BREADS',bd = 10,relief = GROOVE,bg = bg_color,fg = "gold",font = ("times new roman",13,"bold"))

F2.place(x = 330,y = 180,width = 325,height = 220)


gobi_lbl = Label(F2,font = ("times new roman",12,"bold"),fg = lbl_color,bg = bg_color,text = "Butter Naan")

gobi_lbl.grid(row = 0,column = 0,padx = 5,pady = 10)

gobi_en = Entry(F2,bd = 8,relief = GROOVE,textvariable = self.Butter_Naan)

gobi_en.grid(row = 0,column = 1,ipady = 2,ipadx = 1)

face_lbl = Label(F2,font = ("times new roman",12,"bold"),fg = lbl_color,bg = bg_color,text = "Butter Roti")

face_lbl.grid(row = 1,column = 0,padx = 5,pady = 10)

face_en = Entry(F2,bd = 8,relief = GROOVE,textvariable = self.Butter_Roti)

face_en.grid(row = 1,column = 1,ipady = 2,ipadx = 1)
```

```
wash_lbl = Label(F2,font = ("times new roman",12,"bold"),fg = lbl_color,bg = bg_color,text = "Cheese Garlic Naan")

wash_lbl.grid(row = 2,column = 0,padx = 5,pady = 10)

wash_en = Entry(F2,bd = 8,relief = GROOVE,textvariable = self.Cheese_Garlic_Naan)

wash_en.grid(row = 2,column = 1,ipady = 2,ipadx = 1)

hair_lbl = Label(F2,font = ("times new roman",12,"bold"),fg = lbl_color,bg = bg_color,text = "Keema Naan")

hair_lbl.grid(row = 3,column = 0,padx = 5,pady = 10)

hair_en = Entry(F2,bd = 8,relief = GROOVE,textvariable = self.Keema_Naan)

hair_en.grid(row = 3,column = 1,ipady = 2,ipadx = 1)
```

```
F2 = LabelFrame(self.root,text = 'MAIN COURSE - VEG& NON VEG',bd = 10,relief = GROOVE,bg = bg_color,fg = "gold",font = ("times new roman",13,"bold"))

F2.place(x = 330,y = 400,width = 325,height = 380)
```

```
gobi_lbl = Label(F2,font = ("times new roman",12,"bold"),fg = lbl_color,bg = bg_color,text = "Channa Masala")

gobi_lbl.grid(row = 0,column = 0,padx = 5,pady = 10)

gobi_en = Entry(F2,bd = 8,relief = GROOVE,textvariable = self.Channa_Masala)

gobi_en.grid(row = 0,column = 1,ipady = 2,ipadx = 1)

face_lbl = Label(F2,font = ("times new roman",12,"bold"),fg = lbl_color,bg = bg_color,text = "Kadhai Paneer")

face_lbl.grid(row = 1,column = 0,padx = 5,pady = 10)

face_en = Entry(F2,bd = 8,relief = GROOVE,textvariable = self.Kadhai_Paneer)

face_en.grid(row = 1,column = 1,ipady = 2,ipadx = 1)

wash_lbl = Label(F2,font = ("times new roman",12,"bold"),fg = lbl_color,bg = bg_color,text = "Mushroom Curry")

wash_lbl.grid(row = 2,column = 0,padx = 5,pady = 8)

wash_en = Entry(F2,bd = 8,relief = GROOVE,textvariable = self.Mushroom_Curry)

wash_en.grid(row = 2,column = 1,ipady = 2,ipadx = 0)
```

```
        hair_lbl = Label(F2,font = ("times new roman",12,"bold"),fg = lbl_color,bg = bg_color,text = "Mutton/Chicken")

    hair_lbl.grid(row = 3,column = 0,padx = 5,pady = 8)

    hair_en = Entry(F2,bd = 8,relief = GROOVE,textvariable = self.Mutton_Chicken)

    hair_en.grid(row = 3,column = 1,ipady = 2,ipadx = 0)



     F2 = LabelFrame(self.root,text = 'INDIAN RICE & BIRYANI',bd = 10,relief = GROOVE,bg = bg_color,fg = "gold",font = ("times new roman",13,"bold"))

    F2.place(x = 655,y = 180,width = 325,height = 220)



        gobi_lbl = Label(F2,font = ("times new roman",12,"bold"),fg = lbl_color,bg = bg_color,text = "Mutton Biryani")

    gobi_lbl.grid(row = 0,column = 0,padx = 5,pady = 10)

    gobi_en = Entry(F2,bd = 8,relief = GROOVE,textvariable = self.Mutton_Biryani)

    gobi_en.grid(row = 0,column = 1,ipady = 2,ipadx = 1)

        face_lbl = Label(F2,font = ("times new roman",12,"bold"),fg = lbl_color,bg = bg_color,text = "Chicken Biryani")

    face_lbl.grid(row = 1,column = 0,padx = 5,pady = 10)

    face_en = Entry(F2,bd = 8,relief = GROOVE,textvariable = self.Chicken_Biryani)

    face_en.grid(row = 1,column = 1,ipady = 2,ipadx = 1)

       wash_lbl = Label(F2,font = ("times new roman",12,"bold"),fg = lbl_color,bg = bg_color,text = "Egg/Veg Biryani")

    wash_lbl.grid(row = 2,column = 0,padx = 5,pady = 10)

    wash_en = Entry(F2,bd = 8,relief = GROOVE,textvariable = self.EggVeg_Biryani)

    wash_en.grid(row = 2,column = 1,ipady = 2,ipadx = 1)

     hair_lbl = Label(F2,font = ("times new roman",12,"bold"),fg = lbl_color,bg = bg_color,text = "Jeera/Steamed Rice")

    hair_lbl.grid(row = 3,column = 0,padx = 5,pady = 10)

    hair_en = Entry(F2,bd = 8,relief = GROOVE,textvariable = self.JeeraSteamed_Rice)

    hair_en.grid(row = 3,column = 1,ipady = 2,ipadx = 1)
```

```
F2 = LabelFrame(self.root,text = 'Cold Beverages',bd = 10,relief = GROOVE,bg = bg_color,fg = "gold",font = ("times new roman",13,"bold"))

F2.place(x = 655,y = 400,width = 325,height = 380)


gobi_lbl = Label(F2,font = ("times new roman",12,"bold"),fg = lbl_color,bg = bg_color,text = "Water Bottle")

gobi_lbl.grid(row = 0,column = 0,padx = 5,pady = 10)

gobi_en = Entry(F2,bd = 8,relief = GROOVE,textvariable = self.Water_Bottle)

gobi_en.grid(row = 0,column = 1,ipady = 2,ipadx = 1)

   face_lbl = Label(F2,font = ("times new roman",12,"bold"),fg = lbl_color,bg = bg_color,text = "Salt/Sweet Lassi")

face_lbl.grid(row = 1,column = 0,padx = 5,pady = 10)

face_en = Entry(F2,bd = 8,relief = GROOVE,textvariable = self.SaltSweet_Lassi)

face_en.grid(row = 1,column = 1,ipady = 2,ipadx = 1)

wash_lbl = Label(F2,font = ("times new roman",12,"bold"),fg = lbl_color,bg = bg_color,text = "Juices")

wash_lbl.grid(row = 2,column = 0,padx = 5,pady = 8)

wash_en = Entry(F2,bd = 8,relief = GROOVE,textvariable = self.Juices)

wash_en.grid(row = 2,column = 1,ipady = 2,ipadx = 0)

    hair_lbl = Label(F2,font = ("times new roman",12,"bold"),fg = lbl_color,bg = bg_color,text = "Lemon Soda/Water")

hair_lbl.grid(row = 3,column = 0,padx = 5,pady = 8)

hair_en = Entry(F2,bd = 8,relief = GROOVE,textvariable = self.Lemon_SodaWater)

hair_en.grid(row = 3,column = 1,ipady = 2,ipadx = 0)




#==================Bill Aera================#

F3 = Label(self.root,bd = 10,relief = GROOVE)

F3.place(x = 960,y = 180,width = 325,height = 450)

#==========
```

```
bill_title = Label(F3,text = "Bill Area",font = ("Lucida",13,"bold"),bd= 7,relief = GROOVE)

bill_title.pack(fill = X)


#===========

scroll_y = Scrollbar(F3,orient = VERTICAL)

self.txt = Text(F3,yscrollcommand = scroll_y.set)

scroll_y.pack(side = RIGHT,fill = Y)

scroll_y.config(command = self.txt.yview)

self.txt.pack(fill = BOTH,expand = 1)


#==========Buttons Frame==========#

    F4 = LabelFrame(self.root,text = 'Bill Menu',bd = 10,relief = GROOVE,bg = bg_color,fg = "gold",font = ("times new roman",13,"bold"))

F4.place(x = 0,y = 600,relwidth = 1,height = 145)


#==================

cosm_lbl = Label(F4,font = ("times new roman",15,"bold"),fg = lbl_color,bg = bg_color,text = "Total")

cosm_lbl.grid(row = 0,column = 0,padx = 10,pady = 0)

cosm_en = Entry(F4,bd = 8,relief = GROOVE,textvariable = self.total_bill)

cosm_en.grid(row = 0,column = 1,ipady = 2,ipadx = 5)


#=======================

    genbill_btn = Button(F4,text = "Generate Bill",bg = bg_color,fg = fg_color,font=("lucida",12,"bold"),bd = 7,relief = GROOVE,command = self.bill_area)

genbill_btn.grid(row = 0,column = 2,ipadx = 20,padx = 30)


#====================

    clear_btn = Button(F4,text = "Clear",bg = bg_color,fg = fg_color,font=("lucida",12,"bold"),bd = 7,relief = GROOVE,command = self.clear)
```

```
        clear_btn.grid(row = 0,column = 3,ipadx = 20,padx = 30)


        #======================

            exit_btn = Button(F4,text = "Exit",bg = bg_color,fg = fg_color,font=("lucida",12,"bold"),bd = 7,relief = GROOVE,command = self.exit)

        exit_btn.grid(row = 0,column = 4,ipadx = 20,padx = 30)


#Function to get total prices

    def total(self):

        #==================Total

        sm=0

        sm = (sm+

            (self.Gobi_Manchureian.get() * 189)+

            (self.Chilli_Chicken.get() * 240)+

            (self.Chicken_Lolipop.get() * 240)+

            (self.Paneer_Tikkas.get() * 200)+

            (self.Chicken_Curry_Rice.get() * 300)+

            (self.Mutton_Curry_Rice.get()*350)+

            (self.Prawn_Curry_Rice.get() * 300)+

            (self.Egg_Curry_Rice.get() * 400)+

            (self.Butter_Naan.get() *45)+

            (self.Butter_Roti.get() * 35)+

            (self.Cheese_Garlic_Naan.get() * 80)+

            (self.Keema_Naan.get() * 50)+

            (self.Channa_Masala.get() * 160)+

            (self.Kadhai_Paneer.get() * 240)+

            (self.Mutton_Chicken.get() * 280)+

            (self.Mutton_Biryani.get() *320)+
```

```
            (self.Chicken_Biryani.get() * 270)+
            (self.EggVeg_Biryani.get() * 200)+
            (self.JeeraSteamed_Rice.get() * 150)+
            (self.Water_Bottle.get() * 20)+
            (self.SaltSweet_Lassi.get() * 40)+
            (self.Juices.get() * 80)+
            (self.Lemon_SodaWater.get() * 45)
        )
        self.total_bill.set("Rs. "+str(sm))
        return sm


#Function For Text Area
    def welcome_soft(self):
        self.txt.delete('1.0',END)
        self.txt.insert(END,"        Welcome To Restaurant \n")
        self.txt.insert(END,f"\nBill No. : {str(self.c_bill_no.get())}")
        self.txt.insert(END,f"\nCustomer Name : {str(self.cus_name.get())}")
        self.txt.insert(END,f"\nPhone No. : {str(self.c_phone.get())}")
        self.txt.insert(END,"\n===================================")
        self.txt.insert(END,"\nProduct             Qty       Price")
        self.txt.insert(END,"\n===================================")


#Function to clear the bill area
    def clear(self):
        self.txt.delete('1.0',END)


#Add Product name , qty and price to bill area
```

```
    def bill_area(self):


        self.welcome_soft()
        if self.Gobi_Manchureian.get() != 0:
                                        self.txt.insert(END,f"\nGobiManchureian                    {self.Gobi_Manchureian.get()}
{self.Gobi_Manchureian.get() * 189}")
        if self.Chilli_Chicken.get() != 0:
            self.txt.insert(END,f"\nChilliChicken        {self.Chilli_Chicken.get()}        {self.Chilli_Chicken.get() * 240}")
        if self.Chicken_Lolipop.get() != 0:
             self.txt.insert(END,f"\nChickenLolipop        {self.Chicken_Lolipop.get()}        {self.Chicken_Lolipop.get() *
240}")
        if self.Paneer_Tikkas.get() != 0:
            self.txt.insert(END,f"\nPaneerTikkas        {self.Paneer_Tikkas.get()}        {self.Paneer_Tikkas.get() * 200}")
        if self.Chicken_Curry_Rice.get() != 0 :
                                        self.txt.insert(END,f"\nChickenCurryRice                {self.Chicken_Curry_Rice.get()}
{self.Chicken_Curry_Rice.get() * 300}")
        if self.Mutton_Curry_Rice.get() != 0:
                                        self.txt.insert(END,f"\nMuttonCurryRice                    {self.Mutton_Curry_Rice.get()}
{self.Mutton_Curry_Rice.get()*350}")
        if self.Prawn_Curry_Rice.get() != 0:
                                        self.txt.insert(END,f"\nPrawnCurryRice                    {self.Prawn_Curry_Rice.get()}
{self.Prawn_Curry_Rice.get() * 300}")
        if self.Egg_Curry_Rice.get() != 0:
             self.txt.insert(END,f"\nEggCurryRice        {self.Egg_Curry_Rice.get()}        {self.Egg_Curry_Rice.get() *
400}")
        if self.Butter_Naan.get() != 0:
            self.txt.insert(END,f"\nButterNaan        {self.Butter_Naan.get()}        {self.Butter_Naan.get() *45}")
        if self.Butter_Roti.get() != 0:
            self.txt.insert(END,f"\nButterRoti        {self.Butter_Roti.get()}        {self.Butter_Roti.get() * 35}")
        if self.Cheese_Garlic_Naan.get() != 0:
```

```
                self.txt.insert(END,f"\nCheeseGarlicNaan          {self.Cheese_Garlic_Naan.get()}
{self.Cheese_Garlic_Naan.get() * 80}")
    if self.Keema_Naan.get() != 0:
        self.txt.insert(END,f"\nKeemaNaan          {self.Keema_Naan.get()}        {self.Keema_Naan.get() * 50}")
    if self.Channa_Masala.get() != 0:
        self.txt.insert(END,f"\nChannaMasala          {self.Channa_Masala.get()}          {self.Channa_Masala.get() *
160}")
    if self.Kadhai_Paneer.get() != 0:
        self.txt.insert(END,f"\nKadhaiPaneer          {self.Kadhai_Paneer.get()}          {self.Kadhai_Paneer.get() *
240}")
    if self.Mutton_Chicken.get() != 0:
        self.txt.insert(END,f"\nMuttonChicken          {self.Mutton_Chicken.get()}          {self.Mutton_Chicken.get() *
280}")
    if self.Mutton_Biryani.get() != 0:
        self.txt.insert(END,f"\nMuttonBiryani          {self.Mutton_Biryani.get()}          {self.Mutton_Biryani.get()
*320}")
    if self.Chicken_Biryani.get() != 0:
        self.txt.insert(END,f"\nChickenBiryani          {self.Chicken_Biryani.get()}          {self.Chicken_Biryani.get() *
270}")
    if self.EggVeg_Biryani.get() != 0:
        self.txt.insert(END,f"\nEgg/VegBiryani          {self.EggVeg_Biryani.get()}          {self.EggVeg_Biryani.get()  *
200}")
    if self.JeeraSteamed_Rice.get() != 0:
                self.txt.insert(END,f"\nJeera/SteamedRice          {self.JeeraSteamed_Rice.get()}
{self.JeeraSteamed_Rice.get() * 150}")
    if self.Water_Bottle.get() != 0:
        self.txt.insert(END,f"\nWaterBottle          {self.Water_Bottle.get()}          {self.Water_Bottle.get() * 20}")
    if self.SaltSweet_Lassi.get() != 0:
        self.txt.insert(END,f"\nSaltSweetLassi          {self.SaltSweet_Lassi.get()}          {self.SaltSweet_Lassi.get() *
40}")
    if self.Juices.get() != 0:
        self.txt.insert(END,f"\nJuices          {self.Juices.get()}          {self.Juices.get() * 80}")
```

```
        if self.Lemon_SodaWater.get() != 0:
            self.txt.insert(END,f"\nLemonSoda/Water  {self.Lemon_SodaWater.get()}          {self.Lemon_SodaWater.get()
* 45}")


        self.txt.insert(END,"\n==================================")
        self.txt.insert(END,f"\n                    Total : {self.total()}")



    #Function to exit
    def exit(self):
        self.root.destroy()


    #Function To Clear All Fields




root = Tk()
object = Bill_App(root)
root.mainloop()
```

OUTPUT(SCREENSHOTS)

Screenshot1:-Intially the application starts

Screen shot2:- Taking the order from user

Screen shot 3:- After clicking generate bill

Screen shot 4:-After clicking the clear button

CONCLUSION

The urge for the digital restaurant management systems is increasing day by date. Restaurant Billing System Using Python is a perfect solution for this. Through this the ease of access and flexibility of the day to day works in the restaurant is made simpler. The features such as bill number, CGST and SGST make this software user friendly. Both the management side and worker site can manage the data easily using such a system. It is very good and reliable system which can be in corporate to the chain of hotels so can easily maintained and addressed.

FUTURE WORK

In future, this application can be updated with some more food items. Many other latest features will be added. Project will surely be enhanced with respect to looks and appearance and also as per user requirements. Many more functionalities will be added. Some enhancement can also be done with calculator. For now, this application generates the bill but with respect to future application it will be enhanced that it will also print a bill. It can also be used on a large scale. Many more modification can dowith menu or prices or tax as well. It will be easy to use and bug free to all future or upcoming users. This can also be enhanced in future as per customer requirements. Many more features can be added. This will surely help users instead of making a bill manually.

REFERENCES

[1] Noor Azah Samsudin, Shamsul Kamal AhmadKhalid, Mohd Fikry Akmal Mohd Kohar, Zulkifli Senin,

Mohd Nor Ihkasan; "A Customizable Wireless Food Ordering System with Real-Time Customer Feedback.";2011 IEEE Symposium on Wireless Technology and Applications (ISWTA), September 25-28, 2011, Langkawi , Malaysia

[2] Sakari Pieska, Markus Liuska, Juhana Jauhiainen, and Antti Auno of Centria University of Applied Sciences Ylivieska; "Intelligent Restaurant System Smart Menu That Digital Technology"; coginfocom 2013 4th IEEE International Conference on Cognitive Info communications December 2-5, 2013, Budapest,Hungary.

[3] Ching-suchang, Che-chen Kung, Tan-hsu Tan,"Development and Implementation of an E-Restaurantfor Customer-Centric Service Using Wlan And Rfid Technology", proceedings of the Seventh International Conference On Machine Learning And Cybernetics, Kunming, 12-15 July 2008