

CANTEEN MANAGEMENT SYSTEM

A Course Project Report submitted to the
**VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND
TECHNOLOGY**
in partial fulfilment of the requirements for the award of the degree of
BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE & ENGINEERING

Submitted by

K.LASYA NEHA

21071A6229

Under the Guidance of
Mrs.E.Lalitha



VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institute, NAAC Accredited with 'A++' Grade (CGPA: 3.73/4.0)

NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT B.Tech. Programmes

Approved by AICTE, New Delhi, Affiliated to JNTU-H, Recognised as "College with Potential for Excellence" by UGC VignanaJyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad TS 500 090 India

**VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI
INSTITUTE OF ENGINEERING AND TECHNOLOGY**

An Autonomous Institute, NAAC Accredited with 'A++' Grade (CGPA: 3.73/4.0)

NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT B.Tech. Programmes

Approved by AICTE, New Delhi, Affiliated to JNTU-H, Recognised as "College with Potential for Excellence" by UGC

VignanaJyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad TS 500 090 India



CERTIFICATE

This is to certify that B.SAI NIKHITHA (21071A0508), B.VINAYA (21071A0509), P.AVINASH (21071A0548) & M.SAI KARTHIKEYA (22075A0505) have successfully completed their project work at Computer Science & Engineering Department of Vallurupalli Nageswara Rao Vignana Jyothi Institute of Engineering and Technology, Hyderabad entitled "CANTEEN MANAGAEMENT SYSTEM" in partial fulfilment of the requirements for the award of B.Tech during the academic year 2022-2023. This work is carried out under my supervision and has not been submitted to any other University/ Institute for award of any degree/ diploma.

Project Guide

Mrs.E.Lalitha

Assistant Prof.and Internal Guide

Dept.of CSE-CYS,DS and AI&DS

VNR VJIET

Head of Department

Dr.Mr.Rajasekhar

Prof.and Head

Dept.of CSE-CYS,DS and AI&DS

VNR VJIET

External Examiners

DECLARATION

This is to certify that the project work entitled “**CANTEEN MANAGEMENT SYSTEM**” submitted in VNR Vignana Jyothi Institute of Engineering & Technology in partial fulfilment of requirement for the award of Bachelor of Technology in Computer Science and Engineering is a bonafide report of the work carried out by us under the guidance and supervision of Mr.Ch.Sri Sumanth, Assistant Professor, Department of CSE, VNRVJIET. To the best of our knowledge, this report has not been submitted in any form to any university or institution for the award of any degree or diploma.

Submission Dt:

K.LASYA NEHA

21071A6229

CSE-CYS

ACKNOWLEDGEMENT

An endeavour over a long period can be successful only with the advice and support of many well-wishers. We take this opportunity to express our gratitude and appreciation to all of them. We are indebted to our venerable principal Dr. C. D. Naidu for this unflinching devotion, which lead us to complete this project. The support, encouragement given by him and his motivation led us to complete this project. We wish to express our profound gratitude to Dr. S. Nagini, Associate Professor and HOD CSE Department, VNR Vignana Jyothi Institute of Engineering and Technology for their constant and dedicated service to brighten our carrier. With a great pleasure we express our gratitude to the internal guide **Mr. Ch Sri Sumanth**, Assistant Professor, CSE for his timely help, constant guidance, cooperation, support and encouragement throughout this project. Finally, we wish to express our deep sense of gratitude and sincere thanks to our parents, friends and all our well-wishers who have technically and non-technically contributed for the successful completion of our course-based project.

K.LASYA NEHA

21071A6229

INDEX

S.NO	TOPIC	PAGE
1	PROBLEM STATEMENT	6
2	ABSTRACT	6
3	INTRODUCTION	7
4	METHODOLOGY	8-10
5	EVALUATION AND CASE DESCRIPTION	11
6	IMPLEMENTATION	12-24
7	FEATURES	25
8	FUTURE SCOPE	25
9	CONCLUSIONS	25
10	REFERENCES	26

1.PROBLEM STATEMENT-

VNRVJIET Canteen provides services to students and staff. It serves its customers by serving various items including food, break-fast, snacks, drinks etc. All the items are available at a reasonable rate. Traditionally, the method in which customers specify their desired menu and get supplied is a time consuming process. After some days, the menu(written on board) lost its worthy look and attractiveness. Due to rush in the canteen sometimes there can missing tally of money.

Instead its better to use system and have efficient use of time and boost sales through user friendliness and fast service.

2.ABSTRACT

The purpose of Canteen System is to automate the existing manual system by the help of computerised systems and full-fledged computer software by fulfilling their requirements. It can lead to error free, secure, reliable and fast management system.

Basically the project describes how to manage and provide better services for the students/staff.

Our canteen system provides many features which includes

- Items available in the canteen along with rates.
- Add/update items and rates.
- Students/staff can order the items and get the token/bill.
- Customers may search for items available according to rate/type.
- Only admin can access the system by logging in and can check the daily sales.
- Everything is stored, data is never lost.

3.INTRODUCTION

Canteen management system is to provide fast services to their college students, Staffs etc. Usually, People have to go to canteen and order the foods and they have to wait in queue for a long time to get the orders, But with the help of this you just have to follow a very simple process to order your stuffs. And you need not to wait in the long queue. A canteen facility is a supplementary system that is provided by organizations for their employees/students. Organizations with large numbers of employees cannot handle a canteen with manual processes.

Currently, the way schools/college canteens work is that you pay for the food and you wait for the food in line but the problem is that all the students in a facility have the same time slot for the break hence a group of people end up rushing towards the canteen at the same time this creates a lot of inconvenience for the canteen staff as well as the students and since the time is limited some students end up not eating food or wasting their food. The digitalization of the canteen system will help provide better service to the users and the time consumption will be reduced. The updation and deletion of any item can be done. The online system will be helpful for the food makers to prepare the food as early as possible. As a result, there will be quick service to the customers. No queues can be formed for waiting for the food. The updation of the data to the database will be monitored by the admin. The user's data like recognizing the regular users to the canteen will be done and sent to the database. The security of data is done by the encrypted format and server databases of the institution

Our canteen management system provides a friendly User Interface for numerous food outlets, menu design, billing features and lots more.Implementation of such a system makes the operation of the kitchen and the whole of the canteen as effective and quick as possible

4.METHODOLOGY

The modules used-

- **Tkinter**

It's a standard GUI library for python. Python when combined with tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit. Tkinter provides various controls, such as buttons, labels and text boxes used in a GUI application. These controls are commonly called widgets. The widgets we used for our system are button, checkbox, entry, message, tkMessageBox.

- **PIL**

python imaging library

Python interpreter with image editing capabilities. The Python Imaging Library is best suited for image archival and batch processing applications. Python pillow package can be used for creating thumbnails, converting from one format to another and print images, etc.

- **Openpyxl**

Python provides the Openpyxl module, which is used to deal with Excel files without involving third-party Microsoft application software. By using this module, we can have control over excel without open the application. It is used to perform excel tasks such as read data from excel file, or write data to the excel file, draw some charts, accessing excel sheet, renaming sheet, modification (adding and deleting) in excel sheet, formatting, styling in the sheet, and any other task.

- **Random**

Python Random module is an in-built module of Python which is used to generate random numbers. These are pseudo-random numbers means these are not truly random. This module can be used to perform random actions such as generating random numbers, print random a value for a list or string, etc.

- **Datetime**

Python Datetime module supplies classes to work with date and time. These classes provide a number of functions to deal with dates, times and time intervals. Date and datetime are an object in Python, so when you manipulate them, you are actually manipulating objects and not string or timestamps.

The DateTime module is categorized into 6 main classes –

- ❖ **date** – An idealized naive date, assuming the current Gregorian calendar always was, and always will be, in effect. Its attributes are year, month and day.

- ❖ **time** – An idealized time, independent of any particular day, assuming that every day has exactly $24 \times 60 \times 60$ seconds. Its attributes are hour, minute, second, microsecond, and tzinfo.
- ❖ **datetime** – Its a combination of date and time along with the attributes year, month, day, hour, minute, second, microsecond, and tzinfo.
- ❖ **timedelta** – A duration expressing the difference between two date, time, or datetime instances to microsecond resolution.
- ❖ **tzinfo** – It provides time zone information objects.
- ❖ **timezone** – A class that implements the tzinfo abstract base class as a fixed offset from the UTC



VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

CANTEEN BILLING SYSTEM

LOG IN



5.EVALUATION AND CASE DEMONSTRATION

The following section demonstrates the practical application of the proposed method. In order to demonstrate practical application, Canteen Management System of VNR Vignana Jyothi Institute of Engineering and Technology has been chosen. A heavy crowd of students, faculty have always been seen in VNR canteen. In VNRVJIET the food selection and payment are not done through computerised payment system. Not having a computerised payment system is the problem addressed and hence we want to build an interface aiming to find solution for the problem addressed above.

It is a food token system enabling users to choose food items and pay using QR scanner instead of physical money. The system is developed as a desktop or mobile website where users login to the System. Canteen Management System includes the following features login page, orders page. The details of the students and the faculty are stored in an excel file where details includes University Registration Number and a Password with security provided. There is an other excel file that includes the menu of the canteen along with the prices. These features helps in establishing a user friendly interface. The interface contains list of food items divided into categories as Tiffins, Main course, Fastfood, Desserts, Beverages and Bakery items.

To develop this system Visual Studio Code as the Computing environment. We have imported modules such as Tkinter, PIL, openpyxl, Random, Date and Time.

Students can order their food via their phones beforehand from the website in which the user has to enter the college registration id and then they are ready to order via e-menu and do the payments as well and this helps the college to keep a track on the canteens transactions.[in case of any frauds]

The online canteen food ordering system contains e-menu cards that contain the details of the food. It will provide the list of their various items menu list. The customer can select the desired item and can pay the amount. Immediately after booking, the canteen people will get the information of the order and they will prepare the order. The food will be ready in advance and the customers need not wait near the delivery place.

6.IMPLEMENTATION

```
from tkinter import *
from tkinter import messagebox
from PIL import ImageTk, Image
from openpyxl import load_workbook as lw
from random import randint as ri
from datetime import datetime

def username(*event):
    if username_entry.get() == 'Username':
        username_entry.delete(0, END)
        username_entry.focus()
    if password_entry.get() != 'Password':
        login_button.config(state=NORMAL)

def password(*event):
    if password_entry.get() == 'Password':
        password_entry.delete(0, END)
        password_entry.config(show='*')
        password_entry.focus()
    if username_entry.get() != 'Username':
        login_button.config(state=NORMAL)

def tab2(*event):
    credentials = lw('CmsCd.xlsx')
    credentials = credentials['Credentials']
    for i in range(2, 6):
        if username_entry.get() == str(credentials.cell(i, 2).value):
            if password_entry.get() == str(credentials.cell(i, 3).value):
                login_button.focus_set()
                mainframe1.pack_forget()
                mainframe2.pack()
                t2.set('')
            else:
                messagebox.showerror('ERROR', 'Invalid Password')
                break
        else:
            messagebox.showerror('ERROR', 'Invalid Username')

def total(*event):
    global menu_list
    s = 0
```

```

for category, quantity in menu_list.items():
    sheet = Menu_Data[category]
    for i in range(6):
        c = quantity[i].get()
        if c == '':
            continue
        for j in c: # To eliminate spaces before first alphanumeric
character(if any)
            if j == ' ':
                quantity[i].delete(0, 1)
            else:
                break
        c = quantity[i].get()
        if not c.isnumeric():
            messagebox.showerror('ERROR', 'Invalid input(s)')
            return
        elif int(c) == 0:
            continue
        elif int(c) > 100:
            messagebox.showinfo('NOTICE', 'Item limit(100) exceeded')
            return
        s += float(c) * int(sheet.cell(i + 2, 2).value[:-2])
        k = 0
        for j in c: # To eliminate zeroes before first digit(if any)
            if j == '0':
                quantity[i].delete(0, 1)
            else:
                break

if s == 0:
    messagebox.showerror('ERROR', 'No item selected')
    return

subtotal_entry.config(state=NORMAL)
serviceTax_entry.config(state=NORMAL)
total_entry.config(state=NORMAL)

subtotal_entry.delete(0, END)
serviceTax_entry.delete(0, END)
total_entry.delete(0, END)

subtotal_entry.insert(0, '{:>7.2f}'.format(s))
serviceTax_entry.insert(0, '{:>7.2f}'.format(5 * s / 100))
total_entry.insert(0, '{:>7.2f}'.format(s + 5 * s / 100))

subtotal_entry.config(state=DISABLED)
serviceTax_entry.config(state=DISABLED)
total_entry.config(state=DISABLED)

```

```

        subtotal_label.focus_set()

def invoice():
    global menu_list
    total() # estimate total before invoice (if not totaled before)
    if subtotal_entry.get() == '':
        return
    receipt.config(state=NORMAL)
    receipt.delete('1.0', END)

    bill = ri(1000, 10000)
    date = datetime.now().strftime('%d/%m/%Y')
    time = datetime.now().strftime('%H:%M:%S')

    receipt.insert(END,
                    'BILL:{}\t\t{}\t\t{}\n'.format(str(bill).ljust(13, ' '),
date.center(9, ' '), time.center(9, ' ')))
    receipt.insert(END, '*' * 70)
    receipt.insert(END, '\n')
    receipt.insert(END,
                    '{}\t\t{}\t\t{}\n'.format('ITEMS'.ljust(18, ' '),
'QUANTITY'.center(9, ' '), 'PRICE'.center(9, ' ')))
    receipt.insert(END, '*' * 70)
    receipt.insert(END, '\n')
    s = 0
    for category, quantity in menu_list.items():
        sheet = Menu_Data[category]
        for i in range(6):
            c = quantity[i].get()
            if c == '':
                continue
            if int(c) != 0:
                t = '{:>7.2f}'.format(float(c) * int(sheet.cell(i + 2,
2).value[: -2]))
                receipt.insert(END, '{}\t\t{}\t\t{}\n'.format(sheet.cell(i +
2, 1).value.ljust(18, ' '),
                                                                    str(c).center(9,
' '),
                                                                    ('Rs. ' +
str(t)).ljust(9, ' ')))
                receipt.insert(END, '*' * 70)
                receipt.insert(END, '\n{}\t\t{}\t\t{}\n'.format('Sub Total:'.ljust(18, ' '),
str(' ').center(9, ' '),
                                                                    ('Rs. ' +
subtotal_entry.get()).center(9, ' ')))
                receipt.insert(END, '\n{}\t\t{}\t\t{}\n'.format('Service Tax:'.ljust(18, ' '),
str(' ').center(9, ' '),

```

```

('Rs. ' +
serviceTax_entry.get()).center(9, ' '))
    receipt.insert(END, '\n{}\t\t{}\t\t{}\n'.format('Total:'.ljust(18, ' '),
str(' ').center(9, ' ')),
('Rs. ' +
total_entry.get()).center(9, ' '))
    receipt.insert(END, '*' * 70)
    receipt.config(state=DISABLED)
    button_save.config(state=NORMAL)

def save():
    if len(receipt.get('1.0', 'end-1c')) == 0:
        return
    file = open("canteendb.txt", 'a')
    file.write(receipt.get('1.0', END))
    file.close()
    messagebox.showinfo('Information', 'Your Invoice is saved successfully')

def reset(*event):
    for i in menu_list.values():
        for j in i:
            j.delete(0, END)
    subtotal_label.focus_set()

    subtotal_entry.config(state=NORMAL)
    serviceTax_entry.config(state=NORMAL)
    total_entry.config(state=NORMAL)
    receipt.config(state=NORMAL)

    subtotal_entry.delete(0, END)
    serviceTax_entry.delete(0, END)
    total_entry.delete(0, END)
    receipt.delete('1.0', END)

    subtotal_entry.config(state=DISABLED)
    serviceTax_entry.config(state=DISABLED)
    total_entry.config(state=DISABLED)
    receipt.config(state=DISABLED)

    button_save.config(state=DISABLED)

def logout():
    if messagebox.askyesno('LOGOUT', 'Are you sure you want to Logout ?'):
        reset()
        close_calci()
        mainframe2.pack_forget()

```

```

        mainframe1.pack()
        password_entry.focus()

def calculator():
    if calci.wm_state() == 'withdrawn' or calci.wm_state() == 'iconic':
        calci.deiconify()
        calci.focus_force()

def close_calci():
    calci.withdraw()
    clear()

def close_root():
    if mainframe2.winfo_ismapped():
        if messagebox.askyesno('QUIT', 'Are you sure you want to Quit ?'):
            calci.destroy()
            root.destroy()
    else:
        calci.destroy()
        root.destroy()

def clear():
    Disp_box.delete(0, END)

def backspace():
    global flag, exp
    s = Disp_box.get()
    clear()
    if flag == 0:
        Disp_box.insert(0, s[:-1])
    else:
        Disp_box.insert(0, exp[:-1])
    flag = 0

def click_number(n):
    global flag
    if flag == 2 or (flag == 1 and n not in operators):
        clear()
    s = Disp_box.get()
    if (len(s) == 1 and s[-1] == '0') or (len(s) > 1 and (s[-1] == '0' and s[-2] in operators)):

```



```

        backspace()
    flag = 0
    Disp_box.insert(END, str(n))

def click_operator(n):
    global flag
    if flag == 2:
        clear()
    s = Disp_box.get()
    if (len(s) == 0 and n != '-') or (len(s) == 1 and s[-1] == '-'):
        return
    elif s != '':
        if n != '-':
            if s[-1] in operators:
                backspace()
            if s[-2] in operators:
                backspace()
        else:
            if s[-1] in operators[:2]:
                backspace()
    flag = 0
    Disp_box.insert(END, n)

def click_dot(n):
    global flag
    if flag == 2 or (flag == 1 and n not in operators):
        clear()
    s = Disp_box.get()
    if s != '':
        if s[-1] == '.':
            return
        elif s[-1] in operators:
            n = '0' + n
    else:
        n = '0' + n
    Disp_box.insert(END, n)
    flag = 0

def equate(*event):
    global flag, exp
    s = exp = Disp_box.get()
    if len(s) == 0 or (len(s) > 0 and s[-1] in operators):
        return
    s = s.replace('x', '*')
    s = s.replace('÷', '/')
    try:

```

```

        s = eval(s)
        flag = 1
        if type(s) == float and s.is_integer():
            s = int(s)
        except ZeroDivisionError:
            s = 'Cannot divide by zero'
            flag = 2
        except (NameError, SyntaxError, TypeError):
            s = 'ERROR'
            flag = 2

    clear()
    Disp_box.insert(0, s)

# Driver code

root = Tk()
root.title('VNR CANTEEN BILLING SYSTEM')
root.geometry('900x600')
root.state('zoomed')
root.config(bg='#000080')
screen_width = root.winfo_screenwidth()
screen_height = root.winfo_screenheight()

login_image = ImageTk.PhotoImage(Image.open('Cms1.png').resize((screen_width,
screen_height), Image.Resampling.LANCZOS))
calci_icon = ImageTk.PhotoImage(Image.open('Cms2.png').resize((35, 35),
Image.Resampling.LANCZOS))
Menu_Data = lw('Cmsmenu.xlsx')

mainframe1 = Frame(root)
mainframe1.pack()

login_page = Label(mainframe1, image=login_image, bd=0)
login_page.grid(row=0, column=0)

login_label = Label(mainframe1, text='LOG IN', font=('Inter', 40, 'bold'),
bg='#FFFFFF')
login_label.place(x=560, y=400)

t1 = StringVar()
t2 = StringVar()
t1.set('Username')
t2.set('Password')

username_entry = Entry(mainframe1, font=('Inter', 25), textvariable=t1,
bg='#ADADFF', bd=7, width=23, relief=RIDGE)
username_entry.place(x=560, y=510)

```

```

password_entry = Entry(mainframe1, font=('Inter', 25), textvariable=t2,
                        bg='#ADADFF', bd=7, width=23, relief=RIDGE,
                        show='')
password_entry.place(x=560, y=570)

login_label.focus_set()
login_label.bind('<Tab>', username)
username_entry.bind('<Button-1>', username)
username_entry.bind('<Tab>', password)
username_entry.bind('<Return>', password)
password_entry.bind('<Button-1>', password)
password_entry.bind('<Return>', tab2)

login_button = Button(mainframe1, font=('Inter', 24), text='Log in',
                      command=tab2, bg='#00A3FF', width=22,
                      state=DISABLED)
login_button.place(x=560, y=660)

mainframe2 = Frame(root, bg='#000080')
# mainframe2.pack()

# Frames

Top_Frame = Frame(mainframe2, bg='#000080')
Top_Frame.pack(side=TOP)

title_label = Label(Top_Frame, text='VNR Canteen Management System',
                    font=('Calibri', 50, 'bold'), bg='#000080',
                    fg='Yellow', width=screen_width // 35, padx=4, bd=12,
                    relief=RIDGE)
title_label.grid(row=0, column=0)

Left_frame = Frame(mainframe2, bd=12, relief=RIDGE, bg='#000080')
Left_frame.pack(side=LEFT)

menu_frame = Frame(Left_frame, bg='#000000')
menu_frame.pack(side=LEFT)

Fastfood_frame = LabelFrame(menu_frame, text='FASTFOOD', font=('Calibri', 20,
'bold'), bg='#000000', fg='yellow', bd=10)
Fastfood_frame.grid(row=0, column=0, padx=3)

Bakeryitems_frame = LabelFrame(menu_frame, text='BAKERY ITEMS',
font=('Calibri', 20, 'bold'), bg='#000000', fg='yellow', bd=10)
Bakeryitems_frame.grid(row=0, column=1, padx=3)

```

```

MainCourse_frame = LabelFrame(menu_frame, text='MAIN COURSE', font=('Calibri',
20, 'bold'), bg='#000000', fg='yellow',
                                bd=10)
MainCourse_frame.grid(row=0, column=2, padx=3)

Tiffins_frame = LabelFrame(menu_frame, text='TIFFINS', font=('Calibri', 20,
'bold'), bg='#000000', fg='yellow', bd=10)
Tiffins_frame.grid(row=1, column=0, padx=3, pady=3)

Beverages_frame = LabelFrame(menu_frame, text='BEVERAGES', font=('Calibri',
20, 'bold'), bg='#000000', fg='yellow',
                                bd=10)
Beverages_frame.grid(row=1, column=1, padx=3)

Desserts_frame = LabelFrame(menu_frame, text='DESSERTS', font=('Calibri', 20,
'bold'), bg='#000000', fg='yellow', bd=10)
Desserts_frame.grid(row=1, column=2, padx=3)

Right_outer_frame = Frame(mainframe2, bg='#000080', relief=RIDGE, bd=12)
Right_outer_frame.pack(side=RIGHT)

Right_frame = Frame(Right_outer_frame, bg='DarkOrange', relief=RIDGE)
Right_frame.pack()

receipt_frame = Frame(Right_frame, bd=10, relief=RIDGE)
receipt_frame.pack()

amount_frame = Frame(Right_frame, bg='Gray')
amount_frame.pack(fill=X)

buttons_frame = Frame(Right_frame, bd=10, relief=RIDGE, bg='DarkOrange')
buttons_frame.pack()

# Menu

menu_frame_list = [Fastfood_frame, Bakeryitems_frame, MainCourse_frame,
Tiffins_frame, Beverages_frame, Desserts_frame]

menu_list = {'FASTFOOD': [], 'BAKERY ITEMS': [], 'MAIN COURSE': [], 'TIFFINS':
[], 'BEVERAGES': [], 'DESSERTS': []}

for i in menu_frame_list:
    text = i.cget('text')
    sheet = Menu_Data[text]
    for j in range(2, sheet.max_row+1):
        item_label = Label(i, text=sheet.cell(j, 1).value.ljust(19, ' '),
font=('Calibri', 15), bg='#000000',
                                fg='Yellow')

```

```

        item_label.grid(row=j, column=0, sticky=W, padx=3)
        rate_label = Label(i, text=sheet.cell(j, 2).value.rjust(5, ' '),
font=('Calibri', 15), bg='#000000',
                        fg='Yellow', width=screen_width // 307)
        rate_label.grid(row=j, column=1, sticky=E, padx=3, pady=screen_height
// 87)
        entry_box = Entry(i, font=('Arial', 15), bd=7, width=4, justify=RIGHT,
bg='#ADADFF', fg='red')
        entry_box.grid(row=j, column=2, padx=3)
        entry_box.bind('<Return>', total)
        menu_list[text].append(entry_box)
    else:
        entry_box.bind('<Tab>', total)

# Right Part

y = Scrollbar(receipt_frame, orient='vertical')
y.grid(row=0, column=1, sticky=N + S)

receipt = Text(receipt_frame, font=('Helvetica', 14), width=screen_width //
36, height=screen_height // 52, bd=10,
                yscrollcommand=y.set, state=DISABLED)
receipt.grid(row=0, column=0)

y.config(command=receipt.yview)

subtotal_label = Label(amount_frame, text='Sub Total ', font=('Calibri', 20,
'bold'), bg='Gray', fg='#000000')
subtotal_label.grid(row=0, column=0, padx=20, pady=14)

serviceTax_label = Label(amount_frame, text='Service Tax ', font=('Calibri',
20, 'bold'), bg='Gray', fg='#000000')
serviceTax_label.grid(row=1, column=0, padx=20, pady=14)

total_label = Label(amount_frame, text='Total ', font=('Calibri', 20, 'bold'),
bg='Gray', fg='#000000')
total_label.grid(row=2, column=0, padx=20, pady=13)

subtotal_entry = Entry(amount_frame, font=('Calibri', 20, 'bold'),
state=DISABLED, justify=RIGHT, bd=5, relief=RIDGE,
                        bg='#F0F0F8')
subtotal_entry.grid(row=0, column=1)

serviceTax_entry = Entry(amount_frame, font=('Calibri', 20, 'bold'),
state=DISABLED, justify=RIGHT, bd=5, relief=RIDGE,
                        bg='#F0F0F8')
serviceTax_entry.grid(row=1, column=1)

```

```

total_entry = Entry(amount_frame, font=('Calibri', 20, 'bold'),
state=DISABLED, justify=RIGHT, bd=5, relief=RIDGE,
                    bg='#F0F0F8')
total_entry.grid(row=2, column=1)

calculator_button = Button(amount_frame, image=calci_icon, command=calculator)
calculator_button.grid(row=2, column=2, rowspan=2, padx=3)

button_total = Button(buttons_frame, text='Total', font=('Calibri', 18,
'bold'), bg='DarkOrange', fg='#000000', bd=5,
                    padx=9, command=total)
button_total.grid(row=0, column=0)

button_invoice = Button(buttons_frame, text='Invoice', font=('Calibri', 18,
'bold'), bg='DarkOrange', fg='#000000',
                    bd=5, padx=8, command=invoice)
button_invoice.grid(row=0, column=1)

button_save = Button(buttons_frame, text='Save', font=('Calibri', 18, 'bold'),
bg='DarkOrange', fg='#000000', bd=5,
                    padx=10, command=save, state=DISABLED)
button_save.grid(row=0, column=2)

button_reset = Button(buttons_frame, text='Reset', font=('Calibri', 18,
'bold'), bg='DarkOrange', fg='#000000', bd=5,
                    padx=9, command=reset)
button_reset.grid(row=0, column=3)

button_logout = Button(buttons_frame, text='Logout', font=('Calibri', 18,
'bold'), bg='DarkOrange', fg='#000000', bd=5,
                    padx=7, command=logout)
button_logout.grid(row=0, column=4)

# Calculator Window

calci = Tk()
calci.resizable(0, 0)
calci.title('Calculator')
calci.geometry('392x380+590+250')
operators = '-+×÷'
exp = ''
flag = 0

Disp_box = Entry(calci, width=24, borderwidth=15, justify=RIGHT, bg='#FFFC1',
font=('Inter', 20))
Disp_box.grid(row=0, column=0, columnspan=4)

```

```

button_clear = Button(calci, text='CLEAR', width=24, padx=11, height=2,
font=('Inter', 15), bg='gray',
                        activebackground='gray', command=clear)
button_backspace = Button(calci, text='DEL', width=8, height=2, font=('Inter',
15), bg='gray',
                        activebackground='gray', command=backspace)

button_add = Button(calci, text='+', width=8, pady=13, font=('Inter', 15),
bg='gray', activebackground='gray',
                        command=lambda: click_operator('+'))
button_sub = Button(calci, text='-', width=8, height=2, font=('Inter', 15),
bg='gray', activebackground='gray',
                        command=lambda: click_operator('-'))
button_div = Button(calci, text='÷', width=8, height=2, font=('Inter', 15),
bg='gray', activebackground='gray',
                        command=lambda: click_operator('÷'))
button_mul = Button(calci, text='×', width=8, height=2, font=('Inter', 15),
bg='gray', activebackground='gray',
                        command=lambda: click_operator('×'))

button_1 = Button(calci, text='1', width=8, height=2, font=('Inter', 15),
bg='cyan', activebackground='cyan',
                        command=lambda: click_number('1'))
button_2 = Button(calci, text='2', width=8, height=2, font=('Inter', 15),
bg='cyan', activebackground='cyan',
                        command=lambda: click_number('2'))
button_3 = Button(calci, text='3', width=8, height=2, font=('Inter', 15),
bg='cyan', activebackground='cyan',
                        command=lambda: click_number('3'))
button_4 = Button(calci, text='4', width=8, height=2, font=('Inter', 15),
bg='cyan', activebackground='cyan',
                        command=lambda: click_number('4'))
button_5 = Button(calci, text='5', width=8, height=2, font=('Inter', 15),
bg='cyan', activebackground='cyan',
                        command=lambda: click_number('5'))
button_6 = Button(calci, text='6', width=8, height=2, font=('Inter', 15),
bg='cyan', activebackground='cyan',
                        command=lambda: click_number('6'))
button_7 = Button(calci, text='7', width=8, height=2, font=('Inter', 15),
bg='cyan', activebackground='cyan',
                        command=lambda: click_number('7'))
button_8 = Button(calci, text='8', width=8, height=2, font=('Inter', 15),
bg='cyan', activebackground='cyan',
                        command=lambda: click_number('8'))
button_9 = Button(calci, text='9', width=8, height=2, font=('Inter', 15),
bg='cyan', activebackground='cyan',
                        command=lambda: click_number('9'))

```

```

button_0 = Button(calci, text='0', width=8, pady=13, font=('Inter', 15),
bg='cyan', activebackground='cyan',
                command=lambda: click_number('0'))

button_dot = Button(calci, text='.', width=8, pady=13, font=('Inter', 15),
bg='cyan', activebackground='cyan',
                command=lambda: click_dot('.'))
button_equal = Button(calci, text='=', width=8, pady=13, font=('Inter', 15),
bg='blue', activebackground='blue',
                command=equate)

button_clear.grid(row=1, column=0, columnspan=3)
button_backspace.grid(row=1, column=3)

button_7.grid(row=2, column=0)
button_8.grid(row=2, column=1)
button_9.grid(row=2, column=2)
button_div.grid(row=2, column=3)

button_4.grid(row=3, column=0)
button_5.grid(row=3, column=1)
button_6.grid(row=3, column=2)
button_mul.grid(row=3, column=3)

button_1.grid(row=4, column=0)
button_2.grid(row=4, column=1)
button_3.grid(row=4, column=2)
button_sub.grid(row=4, column=3)

button_dot.grid(row=5, column=0)
button_0.grid(row=5, column=1)
button_equal.grid(row=5, column=2)
button_add.grid(row=5, column=3)

calci.bind('<Return>', equate)
calci.withdraw()

calci.protocol('WM_DELETE_WINDOW', close_calci)
root.protocol('WM_DELETE_WINDOW', close_root)

calci.mainloop()
root.mainloop()

```


7.FEATURES

Our Canteen Billing System has the following salient features

- Show the items available in the canteen along with rates.
- Admin can add/update items and rates.
- Type of users: anonymous, admin, student, faculty.
- Customers may order the items and get the token / bill
- Customers may search for the items available according to rate, type
- The admin can see the daily sales and all the bills are stored in a file
- Summary reports can be generated by the analysis of the owner
- Everything is stored, data is never lost.
- Only admin can access the System by logging in

8.FUTURE SCOPE

The future of Canteen System is going to continue to evolve. This project could possibly be even extended by using sound effects and adding a few options such as themes, other options etc. We can further add filter options, voice activated system to take orders for advancing this program. Further this system can be integrated with the payment merchant for seamless transaction. Also, this project can be extended by using student accounts to handle payments and have bonus points for more attractive and user-friendly experience. We also plan on adding Summary report features where without the intervention of the owner, sales statistics graphs and predictor can be generated.

9.CONCLUSION

- The development of Canteen Management System involved many phases. The approach used is a top-down one concentrating on what first then how and moving to successive levels of details.
- The first phase started with a detailed study of the problems and prospects of ordering in Foods.
- This Software is efficient in maintaining customer's details and can easily perform operations on platform.

10.REFERENCES

- 1) International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
- 2) <https://www.acssonaicollege.com/wp-content/uploads/2022/01/Devhare-Vanita-Dhokne.pdf>
- 3) <https://www.ijitee.org/wp-content/uploads/papers/v9i7/G5095059720.pdf>
- 4) <https://www.starlinkindia.com/wp-content/uploads/2017/01/CANTEEN-MANAGEMENT-SYSTEM.pdf>