

The background of the entire page is a repeating pattern of various food items, including burgers, fries, coffee cups, and containers, rendered in a sketchy, brown line-art style. A large, bold, orange diagonal line runs from the top-left towards the bottom-right, crossing the page. A grey rectangular box is positioned horizontally across the middle of the page, containing the title. Another white rectangular box is located in the lower-left area, containing the author's name.

E-FOOD DELIVERY DATABASE

By
Arjun Sekar
Gautham US

Contents

S.No.	Topic	Page No.
1	BONAFIDE	3
2	ACKNOWLEDGEMENT	4
3	PYTHON OVERVIEW	5
4	PROJECT DESCRIPTION	7
5	IMPORT MODULES	8
6	FILES USED	9
7	FUNCTIONS USED	10
7	TABLES USED	13
8	SOURCE CODE	15
9	SAMPLE OUTPUTS	62
10	CONCLUSION	70
11	BIBLIOGRAPHY	71



Chettinad Vidyashram

(Affiliated to Central Board of Secondary Education, New Delhi)
(Chettinad House, R.A.Puram, Chennai – 600 028)

COMPUTER SCIENCE

Certified to be the Bonafide Record of work done by

_____ of Std XII Sec _____

in the Computer Science Lab of the CHETTINAD VIDYASHRAM,
CHENNAI, during the year 2021 – 2022.

Date:

Teacher-in-charge

REGISTER NO. _____

Submitted for All India Senior Secondary Practical Examination in

Computer Science held on _____ at

Chettinad Vidyashram, Chennai – 600 028.

Principal

Internal Examiner

External Examiner

Acknowledgement

I would like to express my sincere thanks to Meena Aunty, Principal Mrs. S.Amudhalakshmi for their encouragement and support to work on this Project. I am grateful to my computer science teacher UMA MAGESWARI R and to the computer science department for the constant guidance and support to complete the project.

Python Overview

Python is a high-level, interpreted, interactive and object-oriented scripting language. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is interpreted** – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is interactive** – You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

Evolution of Python

Version 1.0 was launched in January year 1994. Its main release aspects included several new elements as well as useful software design tools, which included lambda, map, and filter.

Version 2.0 was launched in October 2000, featuring a new list translation function plus a waste collection system.

In December 2008, version 3.0 was launched. It includes several new characteristics and upgrades, together with several discontinued features.

Python 4.0 is supposed to be available by 2023 once version 3.9 is launched. It should come along with elements helping programmers quickly shift from version 3 to version 4.

Features of Python

- **Easy-to-learn** – Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** – Python code is more clearly defined and visible to the eyes.

- **Easy-to-maintain** – Python's source code is fairly easy to maintain.
- **A broad standard library** – Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** – Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** – Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** – You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** – Python provides interfaces to all major commercial databases.
- **Scalable** – Python provides a better structure and support for large programs than shell scripting.
- **GUI Programming** – Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

Getting Python

The most up-to-date and current source code, binaries, documentation, news, etc., are available on the official website of Python (<https://www.python.org>). The website also contains detailed documentation available in HTML, PDF and PostScript formats.

Uses of Python

- Python is often used as a support language for software developers, for build control and management, testing, and in many other ways.
- Python is a superb language for teaching programming, both at the introductory level and in more advanced courses.
- Python is widely used in scientific and numeric computing.
- Python is used in various fields like game development, web development, language development and so on.

Project Description

History of e-food delivery

The first online food order was a pizza from Pizza Hut in 1994. The online food ordering market has increased in the U.S with 40 percent of U.S adults having ordered their food online once. The online food ordering market includes foods prepared by restaurants, prepared by independent people, and groceries being ordered online and then picked up or delivered.

The first online food ordering service, World Wide Waiter (now known as Waiter.com), was founded in 1995. The site originally serviced only northern California, later expanding to several additional cities in the United States.

By 2015, online ordering began overtaking phone ordering. In 2015, China's online food ordering and delivery market grew from 0.15 billion yuan to 44.25 billion yuan. As of September 2016, online delivery accounted for about 3 percent of the 61 billion U.S. restaurant transactions.

According to research conducted by the NDP Group, online restaurant ordering is growing 300% faster than dine-in traffic. Online ordering has started to become the norm, thanks to the convenience, accuracy, and ability to integrate payments. At scale, ubiquitous on-demand and subscription delivery of prepared food could potentially spell the end of cooking at home.

Project Overview

This project consists of an e-food delivery application created using Python and SQL, which can be used to view the menu and place food orders online. The platform for this application has been created using a toolkit of python called tkinter, to provide a graphical user interface and MySQL software to create and handle databases. This application requires new users to create an account while the existing customers can use their login-id and password to view the items in the menu and place an order. The application provides cash and card payment options and has functionality to check the entered card details against the stored card details from database for validation. The application also keeps track of the user's order history and helps the user search for past orders by date. Overall this application enables the users to place food orders online quickly and seamlessly.

Advantages of e-food ordering system

- Safer and healthier for customers and businesses
- Less room for error
- Increases the number of customers you can service
- Increases customer loyalty

- Highly customisable
- Encourages higher customer spend
- Reduces costs

Import Modules

- `from tkinter import *`
- `import tkinter as tk`
- `from tkinter import ttk`
- `from tkinter import messagebox`
- `import mysql.connector`
- `import random`
- `import time`
- `from datetime import *`
- `import pymysql`
- `import tkinter.messagebox`
- `from tkinter.ttk import Combobox`
- `import os`

Files Used

Filename	Purpose of file
full.py	Contains code for Main Screen , login page and registration page
menu.py	Contains code for menu page
order_history.py	Contains code for order history page

User-defined Functions

Function name	Purpose
update(rows)	To update order history table
only_numbers()	To validate if given input is only numeric datatype.
search()	To search items in order history table by date.
only_chars()	To validate if given input is only character datatype.
validateAllFields()	To make entry in every field compulsory in the registration page.
login()	To display the available options in login page.
order_open()	To close the login page after logging in successfully.
order_open1()	To close the registration page after registering successfully.
failed_destroy()	To close the popup message upon invalid/incorrect entries in login page.
logged()	To display popup message upon logging in successfully.
failed()	To display popup message upon invalid/incorrect entries in login page.
login_verification()	To validate login credentials and insert cid into login table.
register_user()	To insert record with given entries into the database.
registered()	To display popup message upon registering successfully.
register()	To display the available options in the registration page.
main_display()	To display the main screen.

iExit():	To send a logout confirmation message
Reset():	To reset all selections in the menu
chkSpicychicken()	To enable the quantity entry of Spicy Chicken Double patty burger check button is ticked
chkPaneerDoublePatty()	To enable the quantity entry of Paneer Double Patty burger when check button is ticked
chkVegBurger()	To enable the quantity entry of Veg Burger when check button is ticked
chkPaneerwrap()	To enable the quantity entry of Paneer wrap when check button is ticked
chkaloowrap()	To enable the quantity entry of aloo wrap when check button is ticked
chkcheesecorn()	To enable the quantity entry of cheese corn sandwich when check button is ticked
chkchickentandoori()	To enable the quantity entry of chicken tandoori sandwich when check button is ticked
chkveggrilled()	To enable the quantity entry of veg grilled sandwich when check button is ticked
chkExtracheese()	To enable the quantity entry of extra cheese sandwich when check button is ticked
chkchocolava()	To enable the quantity entry of choco lava when check button is ticked
chkstrawberry()	To enable the quantity entry of strawberry ice cream when check button is ticked
chkpancakes()	To enable the quantity entry of pancakes & syrup when check button is ticked

chkchocolate()	To enable the quantity entry of chocolate ice cream when check button is ticked
chkvanilla()	To enable the quantity entry of vanilla milkshake when check button is ticked
chkcoke()	To enable the quantity entry of coke when check button is ticked
chksprite()	To enable the quantity entry of sprite burger when check button is ticked
chkmiranda()	To enable the quantity entry of miranda when check button is ticked
chk7up()	To enable the quantity entry of 7up when check button is ticked
chkhashbrown()	To enable the quantity entry of hash brown when check button is ticked
chkgarlicbread()	To enable the quantity entry of garlic bread when check button is ticked
chkfriesm()	To enable the quantity entry of fries(M) when check button is ticked
chkfriesl()	To enable the quantity entry of fries(L) when check button is ticked
total()	To calculate total of the quantities of the selected items
orderpage()	To show order summary
ok()	To store the order summary in a table
sel()	To display information on the order summary
only_numbers(char)	To check for characters other than numbers and display error

Tables used:

E-foodelivery database:

Tables_in_efooddelivery	
customer	
foodmenu	
login	
order_history	

Table: Customer

```
mysql> select * from customer order by cid;
```

cid	first_name	last_name	address	city	postal_code	nearby_landmark	login_id	password	cd_no	cvv
1	1	1	1	1	1	1	1	1	1	1
2	sudha	umashankar	velachery	chennai	600042	kb bank	9840364124	1431	1234567890123456	123

Table: foodmenu

```
mysql> select * from foodmenu;
```

item_no	item	category	price
1	Spicy chicken double patty burger	Burgers&Wraps	250
2	Panner double patty burger	Burgers&Wraps	246
3	Veg burger	Burgers&Wraps	160
4	Panner wrap	Burgers&Wraps	125
5	aloo wrap	Burgers&Wraps	100
6	cheese corn delight sandwich	Sandwich	250
7	chicken tandoori sandwich	Sandwich	270
8	veg grilled sandwich	Sandwich	220
9	Extra cheese panner sandwich	Sandwich	280
10	choco lava	desserts	80
11	strawberry icecream	desserts	90
12	pancakes &syrup	desserts	110
13	chocolate icecream	desserts	100
14	vanilla milkshake	desserts	60
15	coke	drinks	92
16	sprite	drinks	82
17	miranda	drinks	85
18	7up	drinks	73
19	hash brown	sides	55
20	garlic bread	sides	65
21	french fries(m)	sides	75
22	french fries(L)	sides	90

Table: order_history

```
mysql> select * from order_history order by cid;
```

cid	itemname	price	quantity	amount	Date	time
2	Spicy chickendoublepatty burger	250	2.0	500.0	06-09-2021	19:13:36
2	Spicy chickendoublepatty burger	250	2.0	500.0	06-09-2021	22:44:51
2	Spicy chickendoublepatty burger	250	12.0	3000.0	06-09-2021	22:45:32
2	Spicy chickendoublepatty burger	250	6.0	1500.0	06-09-2021	22:48:06
2	Spicy chickendoublepatty burger	250	3.0	750.0	07-09-2021	12:20:54
2	Spicy chickendoublepatty burger	250	2.0	500.0	07-09-2021	23:05:15
2	french fries(m)	75	2.0	150.0	07-09-2021	23:06:36
2	Panner double patty burger	246	3.0	738.0	08-09-2021	01:52:54
2	Spicy chickendoublepatty burger	250	2.0	500.0	08-09-2021	07:35:55
2	Panner wrap	125	3.0	375.0	08-09-2021	07:35:55
2	coke	92	2.0	184.0	08-09-2021	09:17:26
2	Spicy chickendoublepatty burger	250	2.0	500.0	08-09-2021	09:51:53
2	Panner wrap	125	1.0	125.0	09-09-2021	14:24:38
2	Spicy chickendoublepatty burger	250	2.0	500.0	06-09-2021	22:43:50
2	Spicy chickendoublepatty burger	250	2.0	500.0	06-09-2021	22:38:00
2	Panner wrap	125	1.0	125.0	06-09-2021	19:13:36
2	veg grilled sandwich	220	2.0	440.0	06-09-2021	19:13:36
2	choco lava	250	1.0	80.0	06-09-2021	19:13:36
2	sprite	82	3.0	246.0	06-09-2021	19:13:36
2	hash brown	55	2.0	110.0	06-09-2021	19:13:36
2	Panner double patty burger	246	1.0	246.0	06-09-2021	19:19:12
2	sprite	82	1.0	82.0	06-09-2021	19:19:12
2	Panner double patty burger	246	4.0	984.0	06-09-2021	19:51:34
2	sprite	82	2.0	164.0	06-09-2021	22:21:22
2	Spicy chickendoublepatty burger	250	3.0	750.0	06-09-2021	22:32:39
2	Spicy chickendoublepatty burger	250	2.0	500.0	06-09-2021	22:34:02
2	Spicy chickendoublepatty burger	250	3.0	750.0	06-09-2021	22:37:13
2	Veg burger	160	1.0	160.0	17-09-2021	10:39:38

Note: Login table (by default) is an empty table. This stores the cid temporarily during program execution.

Source Code

File:full.py

```
import mysql.connector
import pymysql
import tkinter.messagebox
from tkinter.ttk import Combobox
from tkinter import *
import os

# connecting to the database
connectiondb = mysql.connector.connect(host="localhost", user="root",
passwd="root", charset='utf8',database="efooddelivery")
cursordb = connectiondb.cursor(buffered=True)
cursordb.execute("truncate table login")

def only_numbers(char):
    if char.isdigit():
        return TRUE
    elif char == " ":
        return TRUE
    else:
        messagebox.showinfo('Information', 'Only digits are allowed !')
        return False

def only_chars(char):
    if char.isalpha():
        return TRUE
    elif char == " ":
        return TRUE
    else:
        messagebox.showinfo('Information', 'Enter Valid Name. Numbers not allowed
```



```

        here !')

        return False

# Function for validating all other user input fields
def validateAllFields():
    print('.....mobilenumber get...',mobilenumber.get())
    mobile_no_sql = "select count(*) from customer where login_id = %s"
    cursordb.execute(mobile_no_sql, [(mobilenumber.get())])
    mobile_no_result = cursordb.fetchone()
    mobile_no = mobile_no_result[0]
    print('.....in validate fields.....',mobile_no)
    if firstname.get() == "":
        messagebox.showinfo('Information', 'Please enter first name to Proceed')
    elif lastname.get() == "":
        messagebox.showinfo('Information', 'Please enter last name to Proceed')
    elif address.get() == "":
        messagebox.showinfo('Information', 'Please enter address to Proceed')
    elif city.get() == "":
        messagebox.showinfo('Information', 'Please enter city to Proceed')
    elif len(postalcode.get()) != 6:
        messagebox.showinfo('Information', 'Postal code should be 6 digits')
    elif nearbylandmark.get() == "":
        messagebox.showinfo('Information', 'Please enter nearbylandmark to Proceed')
    elif len(mobilenumber.get()) != 10:
        messagebox.showinfo('Information', 'Phone number should be 10 digits')
    elif len(mobilenumber.get()) == 10 and mobile_no == 1:
        print('inside not None check')
        messagebox.showinfo('Information', 'Phone number already exists !')
    elif len(cd_no.get()) != 16:
        messagebox.showinfo('Information', 'Card Number should be 16 digits')

```

```

elif len(cvv.get()) != 3:
    messagebox.showinfo('Information', 'CVV should be 3 digits')
else:
    register_user()
def login():
    global root2
    root2 = Toplevel(root)
    root2.title("Account Login")
    root2.geometry("1600x800")
    root2.config(bg="white")
    global username_verification
    global password_verification
    Label(root2, text='Please Enter your Account Details', bd=15, font=('Trajan Pro',
    18, 'bold'), relief="groove", fg="white",bg="orange", width=500).pack()
    username_verification = StringVar()
    password_verification = StringVar()
    Label(root2, text="").pack()
    Label(root2, text="Registered Mobile Number :", fg="black", font=('Trajan Pro',
    12, 'bold')).pack()
    Entry(root2, textvariable=username_verification).pack()
    Label(root2, text="").pack()
    Label(root2, text="Password :", fg="black", font=('Trajan Pro', 12, 'bold')).pack()
    Entry(root2, textvariable=password_verification, show="*").pack()
    Label(root2, text="").pack()
    Button(root2, text="Login", bg="orange", fg='white', relief="groove", font=('Trajan
    Pro', 12, 'bold'),command=login_verification).pack()
    Label(root2, text="")
def order_open():
    logged_message.destroy()

```

```

os.system('menu.py')

def order_open1():
    registered_message.destroy()
    os.system('menu.py')
def failed_destroy():
    failed_message.destroy()
def logged(cid_pass):
    root2.destroy()
    username_verification.set("")
    password_verification.set("")
    cid_pass1 = cid_pass
    print(cid_pass1)
    global logged_message
    logged_message = Toplevel()
    logged_message.title("Welcome")
    logged_message.geometry("500x125")
    Label(logged_message, text="Login Successful!... Welcome", fg="green",
    font="bold").pack()
    Label(logged_message, text="").pack()
    Button(logged_message, text="Place order", bg="orange", fg='white',
    relief="groove", font=('Trajan Pro', 12, 'bold'),command=order_open).pack()
def failed():
    global failed_message
    failed_message = Toplevel(root2)
    failed_message.title("Invalid Message")
    failed_message.geometry("500x125")
    Label(failed_message, text="Invalid Username or Password", fg="red",
    font="bold").pack()

```

```

Label(failed_message, text="").pack()
Button(failed_message, text="Ok", bg="orange", fg='white', relief="groove",
font=('Trajan Pro', 12, 'bold'),command=failed_destroy).pack()
def CID(cid):
    LoginCID.cid = cid
    print('...inside global...',LoginCID.cid)
def login_verification():
    user_verification = username_verification.get()
    pass_verification = password_verification.get()
    sql = "select * from customer where login_id = %s and password = %s"
    cursordb.execute(sql, [(user_verification), (pass_verification)])
    results = cursordb.fetchall()
    if results:
        for i in results:
            cid_sql = "select cid from customer where login_id = %s and password = %s"
            cursordb.execute(cid_sql, [(user_verification), (pass_verification)])
            cidresult = cursordb.fetchone()
            print(cidresult)
            cid_pass=cidresult[0]
            print(cid_pass)
            cursordb.execute("insert into login (cid) values(%s)" %(cid_pass))
            connectiondb.commit()
            logged(cid_pass)
            break
    else:
        failed()

def register_user():
    firstname_info = firstname.get()

```

```

lastname_info = lastname.get()
addressss_info = address.get()
city_info = city.get()
postalcode_info = postalcode.get()
nearbylandmark_info = nearbylandmark.get()
mobilenos_info = mobilenos.get()
password_info = password.get()
cd_no_info = cd_no.get()
cvv_info = cvv.get()
global cid
cursordb.execute("select max(cid)+1 from customer")
cid = cursordb.fetchone()[0]
print(cid)
mySql_insert_query = """INSERT INTO customer
(cid,first_name,last_name,address,city,
postal_code,nearby_landmark,login_id,password,cd_no,cvv)
VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s,%s,%s) """
record = (cid, firstname_info, lastname_info, addressss_info, city_info,
postalcode_info, nearbylandmark_info, mobilenos_info,
password_info,cd_no_info,cvv_info)
cursordb.execute(mySql_insert_query,record)
connectiondb.commit()
cid_sql = "select cid from customer where login_id = %s"
cursordb.execute(cid_sql, [(mobilenos_info)])
cidresult = cursordb.fetchone()
print('.....register user...',cidresult)
cid_pass = cidresult[0]
print(cid_pass)
cursordb.execute("insert into login (cid) values(%s)" % (cid_pass))

```

```

connectiondb.commit()
firstname_entry.delete(0, END)
lastname_entry.delete(0, END)
address_entry.delete(0, END)
city_entry.delete(0, END)
postalcode_entry.delete(0, END)
nearbylandmark_entry.delete(0, END)
mobilenos_entry.delete(0, END)
password_entry.delete(0, END)
cd_no_entry.delete(0,END)
cvv_entry.delete(0,END)
registered()
def registered():
    register_screen.destroy()
    global registered_message
    registered_message = Toplevel(root)
    registered_message.title("Welcome")
    registered_message.geometry("500x125")
    Label(registered_message, text="Registered Successfully!... Welcome",
    fg="green",font="bold").pack()
    Label(registered_message, text="").pack()
    Button(registered_message, text="Place order", bg="orange", fg='white',
    relief="groove", font=('Trajan Pro', 12, 'bold'),command=order_open1).pack()
def register():
    global register_screen
    register_screen = Toplevel()
    register_screen.title("Register")
    register_screen.geometry("1000x800")
    global firstname

```

global lastname
global address
global city
global postalcode
global nearbylandmark
global mobileno
global password
global cd_no
global cvv
global firstname_entry
global lastname_entry
global address_entry
global city_entry
global postalcode_entry
global nearbylandmark_entry
global mobileno_entry
global password_entry
global cd_no_entry
global cvv_entry
firstname = StringVar()
lastname = StringVar()
address = StringVar()
city = StringVar()
postalcode = StringVar()
nearbylandmark = StringVar()
mobileno = StringVar()
password = StringVar()
cd_no=StringVar()
cvv=StringVar()


```

Label(register_screen, text="Please enter details below", bg="orange", font=('Trajan
Pro',20,'bold')).place(x=0,y=0)
Label(register_screen, text="").place()
firstname_label = Label(register_screen, text="First Name * ", font=('Trajan
Pro',12))
firstname_label.place(x=10,y=100)
firstname_entry = Entry(register_screen, textvariable=firstname)
firstname_entry.place(x=10,y=125)
valid_firstname= register_screen.register(only_chars)
firstname_entry.config(validate="key", validatecommand=(valid_firstname, '%P'))
lastname_label = Label(register_screen, text="Last Name * ", font=('Trajan
Pro',12))
lastname_label.place(x=10,y=200)
lastname_entry = Entry(register_screen, textvariable=lastname)
lastname_entry.place(x=10,y=225)
valid_lastname= register_screen.register(only_chars)
lastname_entry.config(validate="key", validatecommand=(valid_lastname, '%P'))
address_label = Label(register_screen, text="Address * ", font=('Trajan Pro',12))
address_label.place(x=10,y=300)
address_entry = Entry(register_screen, textvariable=address)
address_entry.place(x=10,y=325)
city_label = Label(register_screen, text="City * ", font=('Trajan Pro',12))
city_label.place(x=10,y=400)
city_entry = Entry(register_screen, textvariable=city)
city_entry.place(x=10,y=425)
postalcode_label = Label(register_screen, text="Postal Code * ", font=('Trajan
Pro',12))
postalcode_label.place(x=10,y=500)
postalcode_entry = Entry(register_screen, textvariable=postalcode)

```

```

postalcode_entry.place(x=10,y=525)
valid_postalcode = register_screen.register(only_numbers)
postalcode_entry.config(validate="key", validatecommand=(valid_postalcode,
'%P'))
nearbylandmark_label = Label(register_screen, text="Nearby Landmark *
",font=('Trajan Pro',12))
nearbylandmark_label.place(x=10,y=600)
nearbylandmark_entry = Entry(register_screen, textvariable=nearbylandmark)
nearbylandmark_entry.place(x=10,y=625)
mobilenumber_label = Label(register_screen, text="Mobile Number * ",font=('Trajan
Pro',12))
mobilenumber_label.place(x=500,y=100)
mobilenumber_entry = Entry(register_screen, textvariable=mobilenumber)
mobilenumber_entry.place(x=500,y=125)
valid_mobilenumber = register_screen.register(only_numbers)
mobilenumber_entry.config(validate="key", validatecommand=(valid_mobilenumber, '%P'))
password_label = Label(register_screen, text="Password * ",font=('Trajan Pro',12))
password_label.place(x=500,y=200)
password_entry = Entry(register_screen, textvariable=password, show='*')
password_entry.place(x=500,y=225)
cd_number_label = Label(register_screen, text="Card Number * ",font=('Trajan
Pro',12))
cd_number_label.place(x=500,y=300)
cd_number_entry = Entry(register_screen, textvariable=cd_number)
cd_number_entry.place(x=500,y=325)
valid_cd_number = register_screen.register(only_numbers)
cd_number_entry.config(validate="key", validatecommand=(valid_cd_number, '%P'))
cvv_label = Label(register_screen, text="cvv * ",font=('Trajan Pro',12))
cvv_label.place(x=500,y=400)

```

```

cvv_entry = Entry(register_screen, textvariable=cvv, show="*")
cvv_entry.place(x=500, y=425)
valid_cvv = register_screen.register(only_numbers)
cvv_entry.config(validate="key", validatecommand=(valid_cvv, '%P'))
Label(register_screen, text="").place()
Button(register_screen, text="Register", height="1", width="20", bg="orange",
fg='white', relief="groove", font=("Trajan Pro", 13),
command=validateAllFields).place(x=700, y=700)
def main_display():
    global root
    root = Tk()
    root.title("Yummy corner")
    root.geometry("1000x800")
    Label(root, text="Welcome to Yummy Corner", bg="orange", width="500",
height="5", font=("Trajan Pro", 18)).pack()
    Label(root, text="").pack()
    Button(root, text="Login", height="5", width="20", bg="orange", fg='white',
relief="groove", font=("Trajan Pro", 13), command=login).place(x=400, y=300)
    Label(root, text="").pack()
    Button(root, text="Register", height="5", width="20", bg="orange", fg='white',
relief="groove", font=("Trajan Pro", 13), command=register).place(x=400, y=600)
    Label(root, text="").pack()
main_display()
root.mainloop()

```

File: menu.py

```

from tkinter import*
import random
import time
from datetime import*

```

```

from tkinter import Tk,StringVar,ttk,messagebox
import mysql.connector
import os

# connecting to the database
connectiondb = mysql.connector.connect(host="localhost", user="root",
passwd="root", charset='utf8', database="efooddelivery")
cursordb = connectiondb.cursor(buffered=True)

global cid
root=Tk()
root.geometry("1800x1050")
root.title(("Yummy Corner"))
cid_sql = "select cid from login"
cursordb.execute(cid_sql)
cid_temp = cursordb.fetchone()
cid=cid_temp[0]
print('....inside menu...',cid)

=====FRAMES=====

tops=Frame(root,width=1800,height=50,bd=10,relief="raise",bg="orange")
tops.pack(side=TOP)

Labelmenu=Label(root,text="MENU",fg="white",relief="raised",font=("Calibri",25,"
bold"),width=50,bg="blue",bd=1)
Labelmenu.place(x=500,y=4)

topleft1=Frame(root,width=600,height=1000,bd=1,relief="raise")
topleft1.pack(side=LEFT)

topleft2=Frame(topleft1,width=600,height=350,bd=1,relief="raise")
topleft2.pack(side=TOP)

topleft3=Frame(topleft1,width=600,height=350,bd=1,relief="raise")
topleft3.pack(side=BOTTOM)

topleft4=Frame(topleft1,width=600,height=350,bd=1,relief="raise")

```

```

topleft4.pack(side=BOTTOM)
topleft5=Frame(root,width=1200,height=350,bd=1,relief="raise")
topleft5.pack(side=TOP)
topleft6=Frame(topleft5,width=600,height=350,bd=1,relief="raise")
topleft6.pack(side=LEFT)
topleft6=Frame(topleft5,width=600,height=350,bd=1,relief="raise")
topleft6.pack(side=RIGHT)
topright2=Frame(root,width=1200,height=700,relief="raise",bd=20)
topright2.pack(side=RIGHT)

=====VARIABLES=====

var1=IntVar()
var2=IntVar()
var3=IntVar()
var4=IntVar()
var5=IntVar()
var6=IntVar()
var7=IntVar()
var8=IntVar()
var9=IntVar()
var10=IntVar()
var11=IntVar()
var12=IntVar()
var13=IntVar()
var14=IntVar()
var15=IntVar()
var16=IntVar()
var17=IntVar()
var18=IntVar()
var19=IntVar()

```

```
var20=IntVar()
var21=IntVar()
var22=IntVar()
var23=StringVar()
var1.set(0)
var2.set(0)
var3.set(0)
var4.set(0)
var5.set(0)
var6.set(0)
var7.set(0)
var8.set(0)
var9.set(0)
var10.set(0)
var11.set(0)
var12.set(0)
var13.set(0)
var14.set(0)
var15.set(0)
var16.set(0)
var17.set(0)
var18.set(0)
var19.set(0)
var20.set(0)
var21.set(0)
var22.set(0)
var30 = IntVar()
varcvv = StringVar()
varSpicychickendoublepattyburger=StringVar()
```

```
varPannerdoublepattyburger=StringVar()
varVegburger=StringVar()
varPannerwrap=StringVar()
varaloowrap=StringVar()
varcheesecorndelightsandwich=StringVar()
varchickentandoorisandwich=StringVar()
varveggrilledsandwich=StringVar()
varExtracheesepannersandwich=StringVar()
varchocolava=StringVar()
varstrawberryicecream=StringVar()
varpancakessyrup=StringVar()
varchocolateicecream=StringVar()
varvanillamilkshake=StringVar()
varcoke=StringVar()
varsprite=StringVar()
varmiranda=StringVar()
var7up=StringVar()
varhashbrown=StringVar()
vargarlicbread=StringVar()
varfrenchfriesm=StringVar()
varfrenchfriesL=StringVar()
vartotal=StringVar()
varsubTotal=IntVar()
varsgst=StringVar()
varcgst=StringVar()
varSpicychickendoublepattyburger.set("0")
varPannerdoublepattyburger.set("0")
varVegburger.set("0")
varPannerwrap.set("0")
```



```

varaloowrap.set("0")
varcheesecorndelightsandwich.set("0")
varchickentandoorisandwich.set("0")
varveggrilledsandwich.set("0")
varExtracheesepannersandwich.set("0")
varchocolava.set("0")
varstrawberryicecream.set("0")
varpancakessyrup.set("0")
varchocolateicecream.set("0")
varvanillamilkshake.set("0")
varcoke.set("0")
varsprite.set("0")
varmiranda.set("0")
var7up.set("0")
varhashbrown.set("0")
vargarlicbread.set("0")
varfrenchfriesm.set("0")
varfrenchfriesL.set("0")
vartotal.set("0")
varsgst.set("0")
varcgst.set("0")

```

=====DEFINITIONS=====

```

def iExit():
    global root
    top = Toplevel()
    top.title("Confirm Logout")
    top.geometry("400x100")
    lbl2=Label(top,text="Are you sure you want to Logout?").pack()
    but1 = Button(top, text="Yes",

```

```

command=exit,bg="orange",fg="white",width=10).place(x=100, y=50)
but1 = Button(top, text="No",
command=top.destroy,bg="orange",fg="white",width=10).place(x=200, y=50)
root.mainloop()
top.mainloop()
def Reset():
    varSpicychickendoublepattyburger.set("0")
    varPannerdoublepattyburger.set("0")
    varVegburger.set("0")
    varPannerwrap.set("0")
    varaloowrap.set("0")
    varcheesecorndelightsandwich.set("0")
    varchickentandoorisandwich.set("0")
    varveggrilledsandwich.set("0")
    varExtracheesepannersandwich.set("0")
    varchocolava.set("0")
    varstrawberryicecream.set("0")
    varpancakessyrup.set("0")
    varchocolateicecream.set("0")
    varvanillamilkshake.set("0")
    varcoke.set("0")
    varsprite.set("0")
    varmiranda.set("0")
    var7up.set("0")
    varhashbrown.set("0")
    vargarlicbread.set("0")
    varfrenchfriesm.set("0")
    varfrenchfriesL.set("0")
    vartotal.set("0")

```

```
varsubTotal.set("0")
varcgst.set("0")
varsgst.set("0")
txtSpicychickendoublepattyburger.configure(state=DISABLED)
txtPannerdoublepattyburger.configure(state=DISABLED)
txtVegburger.configure(state=DISABLED)
txtaloowrap.configure(state=DISABLED)
txtPannerwrap.configure(state=DISABLED)
txtcheesecorndelightsandwich.configure(state=DISABLED)
txtchickentandoorisandwich.configure(state=DISABLED)
txtveggrilledsandwich.configure(state=DISABLED)
txtExtracheesepannersandwich.configure(state=DISABLED)
txtchocolava.configure(state=DISABLED)
txtstrawberryicecream.configure(state=DISABLED)
txtpancakessyrup.configure(state=DISABLED)
txtchocolateicecream.configure(state=DISABLED)
txtvanillamilkshake.configure(state=DISABLED)
txtcoke.configure(state=DISABLED)
txtsprite.configure(state=DISABLED)
txtmiranda.configure(state=DISABLED)
txt7up.configure(state=DISABLED)
txthashbrown.configure(state=DISABLED)
txtgarlicbread.configure(state=DISABLED)
txtfrenchfriesm.configure(state=DISABLED)
txtfrenchfriesL.configure(state=DISABLED)
Spicychickendoublepattyburger.deselect()
Pannerdoublepattyburger.deselect()
Pannerwrap.deselect()
Vegburger.deselect()
```

```

aloowrap.deselect()
cheesecorndelightsandwich.deselect()
chickentandoorisandwich.deselect()
veggrilledsandwich.deselect()
Extracheesepannersandwich.deselect()
chocolava.deselect()
strawberryicecream.deselect()
chocolateicecream.deselect()
pancakessyrup.deselect()
vanillamilkshake.deselect()
coke.deselect()
sprite.deselect()
miranda.deselect()
sevenup.deselect()
hashbrown.deselect()
garlicbread.deselect()
frenchfriesm.deselect()
frenchfriesL.deselect()
def chkSpicychicken():
    if var1.get() == 1:
        txtSpicychickendoublepattyburger.configure(state=NORMAL)
    elif var1.get() == 0:
        txtSpicychickendoublepattyburger.configure(state=DISABLED)
def chkPaneerDoublePatty():
    if var2.get() == 1:
        txtPannerdoublepattyburger.configure(state=NORMAL)
    elif var2.get() == 0:
        txtPannerdoublepattyburger.configure(state=DISABLED)
def chkVegBurger():

```

```

if var3.get() == 1:
    txtVegburger.configure(state=NORMAL)
elif var3.get() == 0:
    txtVegburger.configure(state=DISABLED)
def chkPaneerwrap():
    if var4.get() == 1:
        txtPannerwrap.configure(state=NORMAL)
    elif var4.get() == 0:
        txtPannerwrap.configure(state=DISABLED)
def chkaloowrap():
    if var5.get() == 1:
        txtaloowrap.configure(state=NORMAL)
    elif var5.get() == 0:
        txtaloowrap.configure(state=DISABLED)
def chkcheesecorn():
    if var6.get() == 1:
        txtcheesecorndelightsandwich.configure(state=NORMAL)
    elif var6.get() == 0:
        txtcheesecorndelightsandwich.configure(state=DISABLED)
def chkchickentandoori():
    if var7.get() == 1:
        txtchickentandoorisandwich.configure(state=NORMAL)
    elif var7.get() == 0:
        txtchickentandoorisandwich.configure(state=DISABLED)
def chkveggrilled():
    if var8.get() == 1:
        txtveggrilledsandwich.configure(state=NORMAL)
    elif var8.get() == 0:
        txtveggrilledsandwich.configure(state=DISABLED)

```

```

def chkExtracheese():
    if var9.get() == 1:
        txtExtracheesepannersandwich.configure(state=NORMAL)
    elif var9.get() == 0:
        txtExtracheesepannersandwich.configure(state=DISABLED)
def chkchocolava():
    if var10.get() == 1:
        txtchocolava.configure(state=NORMAL)
    elif var10.get() == 0:
        txtchocolava.configure(state=DISABLED)
def chkstrawberry():
    if var11.get() == 1:
        txtstrawberryicecream.configure(state=NORMAL)
    elif var11.get() == 0:
        txtstrawberryicecream.configure(state=DISABLED)
def chkpancakes():
    if var12.get() == 1:
        txtpancakessyrup.configure(state=NORMAL)
    elif var12.get() == 0:
        txtpancakessyrup.configure(state=DISABLED)
def chkchocolate():
    if var13.get() == 1:
        txtchocolateicecream.configure(state=NORMAL)
    elif var13.get() == 0:
        txtchocolateicecream.configure(state=DISABLED)
def chkvanilla():
    if var14.get() == 1:
        txtvanillamilkshake.configure(state=NORMAL)
    elif var14.get() == 0:

```

```

        txtvanillamilkshake.configure(state=DISABLED)
def chkcoke():
    if var15.get() == 1:
        txtcoke.configure(state=NORMAL)
    elif var15.get() == 0:
        txtcoke.configure(state=DISABLED)
def chksprite():
    if var16.get() == 1:
        txtsprite.configure(state=NORMAL)
    elif var16.get() == 0:
        txtsprite.configure(state=DISABLED)
def chkmiranda():
    if var17.get() == 1:
        txtmiranda.configure(state=NORMAL)
    elif var17.get() == 0:
        txtmiranda.configure(state=DISABLED)
def chk7up():
    if var18.get() == 1:
        txt7up.configure(state=NORMAL)
    elif var18.get() == 0:
        txt7up.configure(state=DISABLED)

def chkhashbrown():
    if var19.get() == 1:
        txthashbrown.configure(state=NORMAL)
    elif var19.get() == 0:
        txt7up.configure(state=DISABLED)
def chkgarlicbread():
    if var20.get() == 1:

```



```

        txtgarlicbread.configure(state=NORMAL)
    elif var20.get() == 0:
        txt7up.configure(state=DISABLED)
def chkfriesm():
    if var21.get() == 1:
        txtfrenchfriesm.configure(state=NORMAL)
    elif var21.get() == 0:
        txt7up.configure(state=DISABLED)
def chkfriesl():
    if var22.get() == 1:
        txtfrenchfriesL.configure(state=NORMAL)
    elif var22.get() == 0:
        txtfrenchfriesL.configure(state=DISABLED)
def total():
    global meal1,meal2,meal3,meal4,meal5,meal6,meal7,meal8,meal9,meal10,meal11,
    meal12,meal13,meal14,meal15,meal16,meal17,meal18,meal19,meal20,meal21,
    meal22,total2
    meal1 = float(varSpicychickendoublepattyburger.get())
    meal2 = float(varPannerdoublepattyburger.get())
    meal3 = float(varVegburger.get())
    meal4 = float(varPannerwrap.get())
    meal5 = float(varaloowrap.get())
    meal6 = float(varcheesecorndelightsandwich.get())
    meal7= float(varchickentandoorisandwich.get())
    meal8 = float(varveggrilledsandwich.get())
    meal9 = float(varExtracheesepannersandwich.get())
    meal10 = float(varchocolava.get())
    meal11 = float(varstrawberryicecream.get())
    meal12 = float(varpancakessyrup.get())

```

```

meal13 = float(varchocolateicecream.get())
meal14 = float(varvanillamilkshake.get())
meal15 = float(varcoke.get())
meal16 = float(varsprite.get())
meal17 = float(varmiranda.get())
meal18 = float(var7up.get())
meal19 = float(varhashbrown.get())
meal20 = float(vargarlicbread.get())
meal21 = float(varfrenchfriesm.get())
meal22 = float(varfrenchfriesL.get())
subTotal=((meal1*250)+(meal2*246)+(meal3*160)+(meal4*125)+(meal5*100)+
(meal6*250)+(meal7*270)+(meal8*220)+(meal9*280)+(meal10*80)+(meal11*90)
+(meal12*110)+(meal13*100)+(meal14*92)+(meal15*82)+(meal16*85)+
(meal17*73)+(meal18*73)+(meal19*55)+(meal20*65)+(meal21*75)+
(meal22*90))
a=str(subTotal)
varsubTotal.set(a)
cgst = (subTotal * 9 / 100)
sgst = (subTotal * 9 / 100)
varcgst.set(cgst)
varsgst.set(sgst)
total2 = cgst + sgst + subTotal
vartotal.set(total2)
def orderpage():
    top2 = Toplevel()
    top2.title("Place order")
    top2.geometry("800x700")
    wrapper1 = LabelFrame(top2, text="Order Summary:", font=("arial", 11, 'italic'))
    wrapper2 = LabelFrame(top2, text="Select Payment Method:", font=("arial", 13,

```

```

'italic'))
wrapper1.pack(fill="both", expand='yes', padx=20, pady=10)
wrapper2.pack(fill="both", expand='yes', padx=20, pady=10)
style = ttk.Style()
style.configure("Treeview.Heading", font=("rockwell", 15, "bold"), fg="blue")
trv = ttk.Treeview(wrapper1, columns=(1, 2, 3), show="headings", height=15)
trv.pack()
trv.heading(1, text="Item Name:")
trv.heading(2, text="Price:")
trv.heading(3, text="Quantity:")
def ok():
    connectiondb = mysql.connector.connect(host="localhost", user="root",
    passwd="root", charset='utf8', database="efooddelivery")
    cursordb = connectiondb.cursor(buffered=True)
    # fetch current login cid
    cid_sql = "select cid from login"
    cursordb.execute(cid_sql)
    cid_result = cursordb.fetchone()
    cid_fetched = cid_result[0]
    print('.....fetch cid stored.....', cid_fetched)
    # fetch cvv stored in customer record
    cvv_sql = "select cvv from customer where cid = %s"
    cursordb.execute(cvv_sql, [(cid_fetched)])
    cvv_result = cursordb.fetchone()
    cvv_stored = cvv_result[0]
    print('.....fetch cvv stored.....', cvv_stored)
    global varcvv, var30
    print('.....entered cvv.....', varcvv.get())
    if len(varcvv.get()) > 3:

```

```

        messagebox.showinfo('Information', 'CVV should be 3 digits')

        return

elif len(varcvv.get()) == 3 and str(cvv_stored) != varcvv.get():

    print('inside not cvv comparison')

    messagebox.showinfo('Information', 'CVV entered is not correct. Please
    check and enter again !')

    return

d = str(date.today())
dt = d.split("-")
a, b, c = dt
e = c + "-" + b + "-" + a
print(e)

if var1.get() == 1:

    orderline1_insert_query = """INSERT INTO Order_History (cid, itemname,
    price, quantity, amount, date,time) VALUES (%s, %s, %s, %s, %s, %s,%s)"""
    record = (cid, 'Spicy chickendoublepatty burger', '250', str(meal1), str(250 *
    (meal1)), e, datetime.now().strftime("%H:%M:%S"))
    cursordb.execute(orderline1_insert_query, record)
    connectiondb.commit()

    varSpicychickendoublepattyburger.set("0")
    txtSpicychickendoublepattyburger.configure(state=DISABLED)
    Spicychickendoublepattyburger.deselect()

if var2.get() == 1:

    orderline2_insert_query = """INSERT INTO Order_History (cid, itemname,
    price, quantity, amount, date,time)VALUES (%s, %s, %s, %s, %s, %s,%s) """
    record = (cid, 'Panner double patty burger', '246', str(meal2), str(246 *
    (meal2)), e, datetime.now().strftime("%H:%M:%S"))
    cursordb.execute(orderline2_insert_query, record)
    connectiondb.commit()

```

```

varPannerdoublepattyburger.set("0")
txtPannerdoublepattyburger.configure(state=DISABLED)
Pannerdoublepattyburger.deselect()
if var3.get() == 1:
    orderline3_insert_query = """INSERT INTO Order_History (cid, itemname,
    price, quantity, amount, date,time) VALUES (%s, %s, %s, %s, %s, %s,%s)
    """
    record = (cid, 'Veg burger', '160', str(meal3), str(160 * (meal3)), e,
    datetime.now().strftime("%H:%M:%S"))
    cursordb.execute(orderline3_insert_query, record)
    connectiondb.commit()
    varVegburger.set("0")
    txtVegburger.configure(state=DISABLED)
    Vegburger.deselect()
if var4.get() == 1:
    orderline4_insert_query = """INSERT INTO Order_History (cid, itemname,
    price, quantity, amount, date,time)VALUES (%s, %s, %s, %s, %s, %s,%s) """
    record = (cid, 'Panner wrap', '125', str(meal4), str(125 * (meal4)), e,
    datetime.now().strftime("%H:%M:%S"))
    cursordb.execute(orderline4_insert_query, record)
    connectiondb.commit()
    varPannerwrap.set("0")
    txtPannerwrap.configure(state=DISABLED)
    Pannerwrap.deselect()
if var5.get() == 1:
    orderline5_insert_query = """INSERT INTO Order_History (cid, itemname,
    price, quantity, amount, date,time) VALUES (%s, %s, %s, %s, %s, %s,%s)
    """
    record = (cid, 'aloo wrap', '100', str(meal5), str(100 * (meal5)), e,

```

```

datetime.now().strftime("%H:%M:%S"))
cursordb.execute(orderline5_insert_query, record)
connectiondb.commit()
varaloowrap.set("0")
txtaloowrap.configure(state=DISABLED)
aloowrap.deselect()
if var6.get() == 1:
    orderline6_insert_query = """INSERT INTO Order_History (cid, itemname,
    price, quantity, amount, date,time)VALUES (%s, %s, %s, %s, %s, %s,%s) """
    record = (cid, 'cheese corn delight sandwich', '250', str(meal6), str(250 *
    (meal6)), e,datetime.now().strftime("%H:%M:%S"))
    cursordb.execute(orderline6_insert_query, record)
    connectiondb.commit()
    varcheesecorndelightsandwich.set("0")
    txtcheesecorndelightsandwich.configure(state=DISABLED)
    cheesecorndelightsandwich.deselect()
if var7.get() == 1:
    orderline7_insert_query = """INSERT INTO Order_History (cid, itemname,
    price, quantity, amount, date,time)VALUES (%s, %s, %s, %s, %s, %s,%s)
    """
    record = (cid, 'chicken tandoori sandwich', '270', str(meal7), str(270 *
    (meal7)), e,datetime.now().strftime("%H:%M:%S"))
    cursordb.execute(orderline7_insert_query, record)
    connectiondb.commit()
    varchickentandoorisandwich.set("0")
    txtchickentandoorisandwich.configure(state=DISABLED)
    chickentandoorisandwich.deselect()
if var8.get() == 1:
    orderline8_insert_query = """INSERT INTO Order_History (cid, itemname,

```

```

price, quantity, amount, date,time VALUES (%s, %s, %s, %s, %s, %s,%s) """
record = ( cid, 'veg grilled sandwich ', '220', str(meal8), str(220 * (meal8)), e,
datetime.now().strftime("%H:%M:%S"))
cursordb.execute(orderline8_insert_query, record)
connectiondb.commit()
varveggrilledsandwich.set("0")
txtveggrilledsandwich.configure(state=DISABLED)
veggrilledsandwich.deselect()
if var9.get() == 1:
    orderline9_insert_query = """INSERT INTO Order_History (cid, itemname,
price, quantity, amount, date,time)VALUES (%s, %s, %s, %s, %s, %s,%s) """
    record = (cid, 'Extra cheese panner sandwich', '280', str(meal9), str(280 *
(meal9)), e,datetime.now().strftime("%H:%M:%S"))
    cursordb.execute(orderline9_insert_query, record)
    connectiondb.commit()
    varExtracheesepannersandwich.set("0")
    txtExtracheesepannersandwich.configure(state=DISABLED)
    Extracheesepannersandwich.deselect()
if var10.get() == 1:
    orderline10_insert_query = """INSERT INTO Order_History (cid, itemname,
price, quantity, amount, date,time)VALUES (%s, %s, %s, %s, %s, %s,%s) """
    record = (cid, 'choco lava', '250', str(meal10), str(80 * (meal10)), e,
datetime.now().strftime("%H:%M:%S"))
    cursordb.execute(orderline10_insert_query, record)
    connectiondb.commit()
    varchocolava.set("0")
    txtchocolava.configure(state=DISABLED)
    chocolava.deselect()
if var11.get() == 1:

```

```

orderline11_insert_query = """INSERT INTO Order_History (cid, itemname,
price, quantity, amount, date,time)VALUES (%s, %s, %s, %s, %s, %s,%s) """
record = (cid, 'strawberry icecream', '90', str(meal11), str(90 * (meal11)), e,
datetime.now().strftime("%H:%M:%S"))
cursordb.execute(orderline11_insert_query, record)
connectiondb.commit()
varstrawberryicecream.set("0")
txtstrawberryicecream.configure(state=DISABLED)
strawberryicecream.deselect()
if var12.get() == 1:
    orderline12_insert_query = """INSERT INTO Order_History (cid, itemname,
price, quantity, amount, date,time)VALUES (%s, %s, %s, %s, %s, %s,%s) """
    record = (cid, 'pancakes & syrup', '110', str(meal12), str(110 * (meal12)), e,
datetime.now().strftime("%H:%M:%S"))
    cursordb.execute(orderline12_insert_query, record)
    connectiondb.commit()
    varpancakessyrup.set("0")
    txtpancakessyrup.configure(state=DISABLED)
    pancakessyrup.deselect()
if var13.get() == 1:
    orderline13_insert_query = """INSERT INTO Order_History (cid, itemname,
price, quantity, amount, date,time)VALUES (%s, %s, %s, %s, %s, %s,%s) """
    record = (cid, 'chocolate icecream', '100', str(meal13), str(100 * (meal13)), e,
datetime.now().strftime("%H:%M:%S"))
    cursordb.execute(orderline13_insert_query, record)
    connectiondb.commit()
    varchocolateicecream.set("0")
    txtchocolateicecream.configure(state=DISABLED)
    chocolateicecream.deselect()

```



```

if var14.get() == 1:
    orderline14_insert_query = """INSERT INTO Order_History (cid, itemname,
    price, quantity, amount, date,time)VALUES (%s, %s, %s, %s, %s, %s,%s) """
    record = (cid, 'vanilla milkshake', '60', str(meal14), str(60 * (meal14)), e,
    datetime.now().strftime("%H:%M:%S"))
    cursordb.execute(orderline14_insert_query, record)
    connectiondb.commit()
    varvanillamilkshake.set("0")
    txtvanillamilkshake.configure(state=DISABLED)
    vanillamilkshake.deselect()

if var15.get() == 1:
    orderline15_insert_query = """INSERT INTO Order_History (cid, itemname,
    price, quantity, amount, date,time)VALUES (%s, %s, %s, %s, %s, %s,%s) """
    record = (cid, 'coke', '92', str(meal15), str(92 * (meal15)), e,
    datetime.now().strftime("%H:%M:%S"))
    cursordb.execute(orderline15_insert_query, record)
    connectiondb.commit()
    varcoke.set("0")
    txtcoke.configure(state=DISABLED)
    coke.deselect()

if var16.get() == 1:
    orderline16_insert_query = """INSERT INTO Order_History (cid, itemname,
    price, quantity, amount, date,time)VALUES (%s, %s, %s, %s, %s, %s,%s)
    """
    record = (cid, 'sprite', '82', str(meal16), str(82 * (meal16)), e,
    datetime.now().strftime("%H:%M:%S"))
    cursordb.execute(orderline16_insert_query, record)
    connectiondb.commit()

```

```

    varsprite.set("0")
    txtsprite.configure(state=DISABLED)
    sprite.deselect()
if var17.get() == 1:
    orderline17_insert_query = """INSERT INTO Order_History (cid, itemname,
    price, quantity, amount, date,time)VALUES (%s, %s, %s, %s, %s, %s,%s) """
    record = (cid, 'miranda', '85', str(meal17), str(85 * (meal17)), e,
    datetime.now().strftime("%H:%M:%S"))
    cursordb.execute(orderline17_insert_query, record)
    connectiondb.commit()
    varmiranda.set("0")
    txtmiranda.configure(state=DISABLED)
    miranda.deselect()
if var18.get() == 1:
    orderline18_insert_query = """INSERT INTO Order_History (cid, itemname,
    price, quantity, amount, date,time)VALUES (%s, %s, %s, %s, %s, %s,%s)
    """
    record = (cid, '7up', '73', str(meal18), str(73 * (meal18)), e,
    datetime.now().strftime("%H:%M:%S"))
    cursordb.execute(orderline18_insert_query, record)
    connectiondb.commit()
    var7up.set("0")
    txt7up.configure(state=DISABLED)
    sevenup.deselect()
if var19.get() == 1:
    orderline19_insert_query = """INSERT INTO Order_History (cid, itemname,
    price, quantity, amount, date,time)VALUES (%s, %s, %s, %s, %s, %s,%s)
    """
    record = (cid, 'hash brown', '55', str(meal19), str(55 * (meal19)), e,

```

```

datetime.now().strftime("%H:%M:%S"))
cursordb.execute(orderline19_insert_query, record)
connectiondb.commit()
varhashbrown.set("0")
txthashbrown.configure(state=DISABLED)
hashbrown.deselect()
if var20.get() == 1:
    orderline20_insert_query = """INSERT INTO Order_History (cid, itemname,
    price, quantity, amount, date,time)VALUES (%s, %s, %s, %s, %s, %s,%s) """
    record = (cid, 'garlic bread', '65', str(meal20), str(65 * (meal20)), e,
    datetime.now().strftime("%H:%M:%S"))
    cursordb.execute(orderline20_insert_query, record)
    connectiondb.commit()
    vargarlicbread.set("0")
    txtgarlicbread.configure(state=DISABLED)
    garlicbread.deselect()
if var21.get() == 1:
    orderline21_insert_query = """INSERT INTO Order_History (cid, itemname,
    price, quantity, amount, date,time)VALUES (%s, %s, %s, %s, %s, %s,%s)
    """
    record = (cid, 'french fries(m)', '75', str(meal21), str(75 * (meal21)), e,
    datetime.now().strftime("%H:%M:%S"))
    cursordb.execute(orderline21_insert_query, record)
    connectiondb.commit()
    varfrenchfriesm.set("0")
    txtfrenchfriesm.configure(state=DISABLED)
    frenchfriesm.deselect()
if var22.get() == 1:
    orderline22_insert_query = """INSERT INTO Order_History (cid,

```

```

        itemname, price, quantity, amount, date,time)VALUES (%s, %s, %s, %s, %s,
        %s,%s) """

record = (cid, 'french fries(L)', '90', str(meal22), str(90 * (meal22)), e,
datetime.now().strftime("%H:%M:%S"))

cursordb.execute(orderline22_insert_query, record)

connectiondb.commit()

varfrenchfriesL.set("0")

txtfrenchfriesL.configure(state=DISABLED)

frenchfriesL.deselect()

vartotal.set("0")

varsubTotal.set("0")

varcgst.set("0")

varsgst.set("0")

messagebox.showinfo("Ordered confirmed", "Thank you for ordering in
Yummy corner! Visit us again! ")

top2.destroy()

def sel():

    if var30.get() == 2:

        Label(wrapper2, text="Enter your cvv:", fg="black", font=('Trajan Pro', 12,
        'bold')).place(x=100,y=80)

        cvv = Entry(wrapper2, textvariable=varcvv, font=('arial', 12, 'bold'), width=6,
        justify='left')

        cvv.place(x=100, y=100)

        print(var30.get())

        R1 = Radiobutton(wrapper2, text="Cash", variable=var30, value=1,
        command=sel)

        R1.place(x=100, y=30)

        R2 = Radiobutton(wrapper2, text="Credit Card", variable=var30, value=2,
        command=sel)

```

```

R2.place(x=100, y=60)
order=[]
a = 1
if var1.get() == 1:
    order.append(["Spicy Chicken Double Patty
    Burger",250,txtSpicychickendoublepattyburger.get()])
if var2.get() == 1:
    order.append(["Paneer Double patty burger", 246,
    txtPannerdoublepattyburger.get()])
if var3.get() == 1:
    order.append(["Veg Burger", 160, txtVegburger.get()])
if var4.get() == 1:
    order.append(["Paneer wrap", 125, txtPannerwrap.get()])
if var5.get() == 1:
    order.append(["aloo wrap", 100, txtaloowrap.get()])
if var6.get() == 1:
    order.append(["cheese corn delight sandwich", 250,
    txtcheesecorndelightsandwich.get()])
if var7.get() == 1:
    order.append(["chicken tandoori sandwich", 270,
    txtchickentandoorisandwich.get()])
if var8.get() == 1:
    order.append(["veg grilled sandwich", 220, txtveggrilledsandwich.get()])
if var9.get() == 1:
    order.append(["Extra cheese paneer sandwich", 280,
    txtExtracheesepannersandwich.get()])
if var10.get() == 1:
    order.append(["choco lava", 80, txtchocolava.get()])
if var11.get() == 1:

```

```

        order.append(["strawberry icecream",90,txtstrawberryicecream.get()])
if var12.get() == 1:
    order.append(["pancakes & syrup", 110, txtpancakessyrup.get()])
if var13.get() == 1:
    order.append(["chocolate icecream", 100, txtchocolateicecream.get()])
if var14.get() == 1:
    order.append(["vanilla milkshake", 60, txtvanillamilkshake.get()])
if var15.get() == 1:
    order.append(["coke", 92, txtcoke.get()])
if var16.get() == 1:
    order.append(["sprite", 82, txtsprite.get()])
if var17.get() == 1:
    order.append(["miranda", 85, txtmiranda.get()])
if var18.get() == 1:
    order.append(["7up", 73, txt7up.get()])
if var19.get() == 1:
    order.append(["hash brown", 55, txthashbrown.get()])
if var20.get() == 1:
    order.append(["garlic bread", 65, txtgarlicbread.get()])
if var21.get() == 1:
    order.append(["french fries(m)", 75, txtfrenchfriesm.get()])
if var22.get() == 1:
    order.append(["french fries(L)", 90, txtfrenchfriesL.get()])
for i in range(len(order)):
    col1 = order[i][0]
    col2 = order[i][1]
    col3 = order[i][2]
    trv.insert("", "end", values=(col1, col2,col3))
myButton10 = Button(wrapper2, text="Confirm Order", height=2, width=18,

```

```

        command=ok,font=('Trajan Pro',14,'bold'),bg="orange",fg="white")
        myButton10.place(x=450,y=40)
        top2.mainloop()
def only_numbers(char):
    if char.isdigit():
        return TRUE
    elif char == " ":
        return TRUE
    else:
        messagebox.showinfo('Information', 'Only digits are allowed')
        return False

=====FRAME 1=====

lblBurgersWraps=Label(topleft2,font=('Verdana',18,'bold'),text='Burgers &
Wraps',fg="goldenrod")
lblBurgersWraps.place(x=180,y=9)
Spicychickendoublepattyburger=Checkbutton(topleft2,text="Spicy chicken double
pattyburger\t₹250",variable=var1,onvalue=1,offvalue=0,font=('arial',12,'bold'),
command=chkSpicychicken)
Spicychickendoublepattyburger.place(x=1,y=60)
txtSpicychickendoublepattyburger=Entry(topleft2,
textvariable=varSpicychickendoublepattyburger,font=('arial',12,'bold'),width=6,justify
='left',state=DISABLED)
txtSpicychickendoublepattyburger.place(x=410,y=64)
valid_postalcode = root.register(only_numbers)
txtSpicychickendoublepattyburger.config(validate="key",validatecommand=(valid_po
stalcode, '%P'))
Pannerdoublepattyburger=Checkbutton(topleft2,text="Paneer double patty
burger\t₹246",variable=var2,onvalue=1,offvalue=0,font=('arial',12,'bold'),command
=

```

```

chkPaneerDoublePatty)
Pannerdoublepattyburger.place(x=1,y=100)
txtPannerdoublepattyburger=Entry(topsleft2,
textvariable=varPannerdoublepattyburger,font=('arial',12,'bold'),width=6,justify='left',
state=DISABLED)
txtPannerdoublepattyburger.place(x=410,y=104)
valid_postalcode = root.register(only_numbers)
txtPannerdoublepattyburger.config(validate="key",validatecommand=(
valid_postalcode, '%P'))
Vegburger=Checkbutton(topsleft2,text="Vegburger\t\t₹160",variable=var3,onvalue=
1,offvalue=0,font=('arial',12,'bold'),command=chkVegBurger)
Vegburger.place(x=1,y=140)
txtVegburger=Entry(topsleft2,textvariable=varVegburger,font=('arial',12,'bold'),width
=6,justify='left',state=DISABLED)
txtVegburger.place(x=410,y=144)
valid_postalcode =root.register(only_numbers)
txtVegburger.config(validate="key",validatecommand=(valid_postalcode, '%P'))
Pannerwrap=Checkbutton(topsleft2,text="Pannerwrap\t\t₹125",variable=var4,
onvalue=1,offvalue=0,font=('arial',12,'bold'),command=chkPaneerwrap)
Pannerwrap.place(x=1,y=180)
txtPannerwrap=Entry(topsleft2,textvariable=varPannerwrap,font=('arial',12,'bold'),
width=6,justify='left',state=DISABLED)
txtPannerwrap.place(x=410,y=184)
valid_postalcode = root.register(only_numbers)
txtPannerwrap.config(validate="key",validatecommand=(valid_postalcode, '%P'))
aloowrap=Checkbutton(topsleft2,text="aloowrap\t\t₹100",variable=var5,onvalue=1,
offvalue=0,font=('arial',12,'bold'),command=chkaloowrap)
aloowrap.place(x=1,y=220)
txtaloowrap=Entry(topsleft2,textvariable=varaloowrap,font=('arial',12,'bold'),width=6,

```



```

justify='left',state=DISABLED)
txtaloowrap.place(x=410,y=224)
valid_postalcode = root.register(only_numbers)
txtaloowrap.config(validate="key",validatecommand=(valid_postalcode, '%P'))
=====FRAME 2=====
lblSandwich=Label(root,font=('Verdana',18,'bold'),text='Sandwich',fg="goldenrod")
lblSandwich.place(x=200,y=410)
cheesecorndelightsandwich=Checkbutton(root,text="cheese corn
delightsandwich\t₹250",variable=var6,onvalue=1,offvalue=0,font=('arial',12,'bold'),
command=chkcheesecorn)
cheesecorndelightsandwich.place(x=1,y=461)
txtcheesecorndelightsandwich=Entry(root,textvariable=varcheesecorndelightsandwich
,font=('arial',12,'bold'),width=6,justify='left',state=DISABLED)
txtcheesecorndelightsandwich.place(x=410,y=461)
valid_postalcode = root.register(only_numbers)
txtcheesecorndelightsandwich.config(validate="key",validatecommand=(valid_postal
code, '%P'))
chickentandoorisandwich=Checkbutton(root,text="chicken tandoori
sandwich\t₹270",variable=var7,onvalue=1,offvalue=0,font=('arial',12,'bold'),
command=chkchickentandoori)
chickentandoorisandwich.place(x=1,y=511)

txtchickentandoorisandwich=Entry(root,textvariable=varchickentandoorisandwich,
font=('arial',12,'bold'),width=6,justify='left',state=DISABLED)
txtchickentandoorisandwich.place(x=410,y=511)
valid_postalcode = root.register(only_numbers)
txtchickentandoorisandwich.config(validate="key",validatecommand=(valid_postalco
de, '%P'))
veggrilledsandwich=Checkbutton(root,text="veg grilled

```

```

sandwich\t\t₹220",variable=var8,onvalue=1,offvalue=0,font=('arial',12,'bold'),
command=chkveggrilled)
veggrilledsandwich.place(x=1,y=561)
txtveggrilledsandwich=Entry(root,textvariable=varveggrilledsandwich,font=('arial',12,
'bold'),width=6,justify='left',state=DISABLED)
txtveggrilledsandwich.place(x=410,y=561)
valid_postalcode = root.register(only_numbers)
txtveggrilledsandwich.config(validate="key",validatecommand=(valid_postalcode,
'%P'))
Extracheesepannerssandwich=Checkbutton(root,text="Extra cheese paneer
sandwich\t\t₹280",variable=var9,onvalue=1,offvalue=0,font=('arial',12,'bold'),
command=chkExtracheese)
Extracheesepannerssandwich.place(x=1,y=611)
txtExtracheesepannerssandwich=Entry(root,textvariable=
varExtracheesepannerssandwich,font=('arial',12,'bold'),width=6,justify='left',state=
DISABLED)
txtExtracheesepannerssandwich.place(x=410,y=611)
valid_postalcode = root.register(only_numbers)
txtExtracheesepannerssandwich.config(validate="key",validatecommand=(valid_postal
code, '%P'))

```

=====FRAME 3=====

```

lblDesserts=Label(root,font=('Verdana',18,'bold'),text='Desserts',fg="goldenrod")
lblDesserts.place(x=200,y=710)
chocolava=Checkbutton(root,text="chocolava\t\t₹80",variable=var10,onvalue=1,
offvalue=0,font=('arial',12,'bold'),command=chkchocolava)
chocolava.place(x=1,y=770)
txtchocolava=Entry(root,textvariable=varchocolava,font=('arial',12,'bold'),width=6,
justify='left',state=DISABLED)

```

```

txtchocolava.place(x=410,y=770)
valid_postalcode = root.register(only_numbers)
txtchocolava.config(validate="key",validatecommand=(valid_postalcode, '%P'))
strawberryicecream=Checkbutton(root,text="strawberryicecream\t₹90",
variable=var11,onvalue=1,offvalue=0,font=('arial',12,'bold'),
command=chkstrawberry)
strawberryicecream.place(x=1,y=820)
txtstrawberryicecream=Entry(root,textvariable=varstrawberryicecream,font=('arial',12
,'bold'),width=6,justify='left',state=DISABLED)
txtstrawberryicecream.place(x=410,y=820)
valid_postalcode = root.register(only_numbers)
txtstrawberryicecream.config(validate="key",validatecommand=(valid_postalcode,
'%P'))
pancakessyrup=Checkbutton(root,text="pancakes &
syrup\t₹110",variable=var12,onvalue=1,offvalue=0,font=('arial',12,'bold'),
command=chkpancakes)
pancakessyrup.place(x=1,y=870)
txtpancakessyrup=Entry(root,textvariable=varpancakessyrup,font=('arial',12,'bold'),
width=6,justify='left',state=DISABLED)
txtpancakessyrup.place(x=410,y=870)
valid_postalcode = root.register(only_numbers)
txtpancakessyrup.config(validate="key",validatecommand=(valid_postalcode, '%P'))
chocolateicecream=Checkbutton(root,text="chocolate
icecream\t₹100",variable=var13,onvalue=1,offvalue=0,font=('arial',12,'bold'),
command=chkchocolate)
chocolateicecream.place(x=1,y=920)
txtchocolateicecream=Entry(root,textvariable=varchocolateicecream,font=('arial',12,'b
old'),width=6,justify='left',state=DISABLED)
txtchocolateicecream.place(x=410,y=920)

```

```

valid_postalcode = root.register(only_numbers)
txtchocolateicecream.config(validate="key",validatecommand=(valid_postalcode,
'%P'))
vanillamilkshake=Checkbutton(root,text="vanilla
milkshake\t\t₹60",variable=var14,onvalue=1,offvalue=0,font=('arial',12,'bold'),
command=chkvanilla)
vanillamilkshake.place(x=1,y=970)
txtvanillamilkshake=Entry(root,textvariable=varvanillamilkshake,font=('arial',12,'bold
'),width=6,justify='left',state=DISABLED)
txtvanillamilkshake.place(x=410,y=970)
valid_postalcode = root.register(only_numbers)
txtvanillamilkshake.config(validate="key",validatecommand=(valid_postalcode,
'%P'))
=====FRAME 4=====
lblDrinks=Label(root,font=('Verdana',18,'bold'),text='Drinks',fg="goldenrod")
lblDrinks.place(x=810,y=50)
coke=Checkbutton(root,text="coke\t\t₹92",variable=var15,onvalue=1,offvalue=0,
font=('arial',12,'bold'),command=chkcoke)
coke.place(x=611,y=120)
txtcoke=Entry(root,textvariable=varcoke,font=('arial',12,'bold'),width=6,justify='left',
state=DISABLED)
txtcoke.place(x=950,y=124)
valid_postalcode = root.register(only_numbers)
txtcoke.config(validate="key",validatecommand=(valid_postalcode, '%P'))
sprite=Checkbutton(root,text="sprite\t\t₹82",variable=var16,onvalue=1,offvalue=0,
font=('arial',12,'bold'),command=chksprite)
sprite.place(x=611,y=170)
txtsprite=Entry(root,textvariable=varsprite,font=('arial',12,'bold'),width=6,justify='left'
,state=DISABLED)

```

```

txtsprite.place(x=950,y=174)
valid_postalcode = root.register(only_numbers)
txtsprite.config(validate="key",validatecommand=(valid_postalcode, '%P'))
miranda=Checkbutton(root,text="miranda\t\t₹85",variable=var17,onvalue=1,offvalue
=0,font=('arial',12,'bold'),command=chkmiranda)
miranda.place(x=611,y=220)
txtmiranda=Entry(root,textvariable=varmiranda,font=('arial',12,'bold'),width=6,justify
='left',state=DISABLED)
txtmiranda.place(x=950,y=224)
valid_postalcode = root.register(only_numbers)
txtmiranda.config(validate="key",validatecommand=(valid_postalcode, '%P'))
sevenup=Checkbutton(root,text="7up\t\t₹73",variable=var18,onvalue=1,offvalue=0,
font=('arial',12,'bold'),command=chk7up)
sevenup.place(x=611,y=270)
txt7up=Entry(root,textvariable=var7up,font=('arial',12,'bold'),width=6,justify='left',
state=DISABLED)
txt7up.place(x=950,y=274)
valid_postalcode = root.register(only_numbers)
txt7up.config(validate="key",validatecommand=(valid_postalcode, '%P'))
=====FRAME 5=====
lblSides=Label(root,font=('Verdana',18,'bold'),text='Sides',fg="goldenrod")
lblSides.place(x=1420,y=50)
hashbrown=Checkbutton(root,text="hashbrown\t\t₹55",variable=var19,onvalue=1,
offvalue=0,font=('arial',12,'bold'),command=chkhashbrown)
hashbrown.place(x=1221,y=120)
txthashbrown=Entry(root,textvariable=varhashbrown,font=('arial',12,'bold'),width=6,
justify='left',state=DISABLED)
txthashbrown.place(x=1580,y=124)
valid_postalcode = root.register(only_numbers)

```

```

txthashbrown.config(validate="key",validatecommand=(valid_postalcode, '%P'))
garlicbread=Checkbutton(root,text="garlicbread\t\t₹65",variable=var20,onvalue=1,
offvalue=0,font=('arial',12,'bold'),command=chkgarlicbread)
garlicbread.place(x=1221,y=170)
txtgarlicbread=Entry(root,textvariable=vargarlicbread,font=('arial',12,'bold'),width=6,
justify='left',state=DISABLED)
txtgarlicbread.place(x=1580,y=174)
valid_postalcode = root.register(only_numbers)
txtgarlicbread.config(validate="key",validatecommand=(valid_postalcode, '%P'))
frenchfriesm=Checkbutton(root,text="frenchfries(m)\t\t₹75",variable=var21,
onvalue=1,offvalue=0,font=('arial',12,'bold'),command=chkfriesm)
frenchfriesm.place(x=1221,y=220)
txtfrenchfriesm=Entry(root,textvariable=varfrenchfriesm,font=('arial',12,'bold'),width
=6,justify='left',state=DISABLED)
txtfrenchfriesm.place(x=1580,y=224)

```

```

valid_postalcode = root.register(only_numbers)
txtfrenchfriesm.config(validate="key",validatecommand=(valid_postalcode, '%P'))
frenchfriesL=Checkbutton(root,text="frenchfries(L)
\t\t₹90",variable=var22,onvalue=1
,offvalue=0,font=('arial',12,'bold'),command=chkfriesl)
frenchfriesL.place(x=1221,y=270)
txtfrenchfriesL=Entry(root,textvariable=varfrenchfriesL,font=('arial',12,'bold'),
width=6,justify='left',state=DISABLED)
txtfrenchfriesL.place(x=1580,y=274)
valid_postalcode = root.register(only_numbers)
txtfrenchfriesL.config(validate="key",validatecommand=(valid_postalcode, '%P'))

```

=====FRAME 6=====

```

lblSubTotal=Label(root,text="Sub Total:",font=('trajan
pro',30,'bold')).place(x=1210,y=500)
txtsubTotal=Entry(root,state=DISABLED,width=8,textvariable=varsubTotal,
font=('trajan pro ',20,'bold'))
txtsubTotal.place(x=1450,y=510)
lbldgst=Label(root,text="CGST:",font=('trajan pro ',30,'bold')).place(x=1210,y=600)
txtdgst = Entry(root, state=DISABLED,width=8,textvariable=vardgst,
font=('trajan pro ',20,'bold'))
txtdgst.place(x=1450, y=610)
lbldgst=Label(root,text="SGST:",font=('trajan pro ',30,'bold')).place(x=1210,y=700)
txtdgst = Entry(root, state=DISABLED,width=8,textvariable=varsdgst,font=('trajan pro
',20,'bold'))
txtdgst.place(x=1450, y=710)
lbldtotal=Label(root,text="Total:",font=('trajan pro ',30,'bold')).place(x=1210,y=800)
txtdtotal = Entry(root, state=DISABLED,width=8,textvariable=vardtotal,font=('trajan
pro ',20,'bold'))
txtdtotal.place(x=1450, y=810)

```

=====BUTTONS=====

```

myButton=Button(root,text="PlaceOrder",height=2,width=20,command=orderpage,
font=('Trajan Pro',15,'bold'),bg="orange",fg="white")
myButton.place(x=1200,y=870)
myButton2=Button(root,text="OrderHistory",height=2,width=20,
command=lambda:os.system('order_history.py'),font=('Trajan
Pro',15,'bold'),bg="orange",fg="white")
myButton2.place(x=610,y=750)
myButton4=Button(root,text="Reset",height=2,width=20,font=('Trajan
Pro',15,'bold'),bg="orange",fg="white",command=Reset)
myButton4.place(x=610,y=620)
myButton3=Button(root,text="Logout",height=2,width=20,font=('Trajan

```

```

Pro',15,'bold'),bg="orange",fg="white",command=iExit)
myButton3.place(x=610,y=880)
myButton3=Button(root,text="Check Total",height=2,width=20,font=('Trajan
Pro',15,'bold'),bg="orange",fg="white",command=total)
myButton3.place(x=610,y=490)
root.mainloop()

```

File:order_history.py

```

from tkinter import *
import tkinter as tk
from tkinter import ttk
from tkinter import messagebox
import mysql.connector
connectiondb = mysql.connector.connect(host="localhost", user="root",
passwd="root", charset='utf8',database="efooddelivery")
cursordb = connectiondb.cursor(buffered=True)
cursordb.execute('use efooddelivery')
cid_sql = "select cid from login"

```

```

cursordb.execute(cid_sql)
cid_temp = cursordb.fetchone()
cid=cid_temp[0]
print('....inside menu...',cid)
def update(rows):
    trv.delete(*trv.get_children())
    for i in rows:
        trv.insert("", "end", values=i)
def search():
    v = q.get()

```



```

# date = '16-06-1998'
cond1 = (len(v) == 10)
cond2 = (len(v.split('-')) == 3)
if cond1 and cond2:
    dd, mm, yyyy = v.split('-')
    cond3 = (1 <= int(dd) <= 31)
    cond4 = (1 <= int(mm) <= 12)
    cond5 = (1000 <= int(yyyy) <= 2021)
    if cond3 and cond4 and cond5:
        query = "select * from Order_History where date='{ }' and
        cid={ }".format(v,cid)
        cursordb.execute(query)
        rows = cursordb.fetchall()
        answer.config(text=' ')
        if len(rows) == 0:
            answer.config(text='No matching results')
            print("line44")
        if mm == "02" and (int(dd) == 30 or int(dd) == 31):
            answer.config(text='Oops!! Invailld Date Entered ')
            print("line47")
        update(rows)
    else:
        answer.config(text='Oops!! Invailld Date Entered ')
        print("line51")
else:
    answer.config(text='Enter correct date in DD-MM-YYYY format')
    print("line54")
root = Tk()
q = StringVar()

```

```

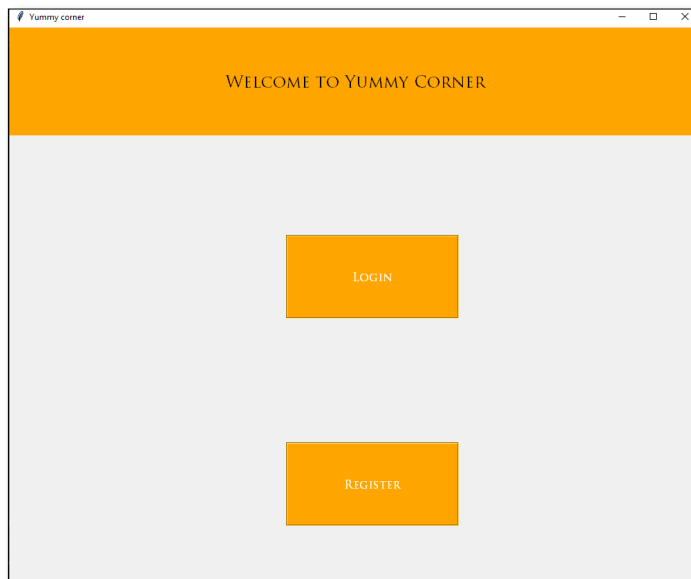
wrapper1 = LabelFrame(root, text="Order History:", font=("arial", 11, 'italic'))
wrapper2 = LabelFrame(root, text="Search:", font=("arial", 13, 'italic'))
wrapper1.pack(fill="both", expand='yes', padx=20, pady=10)
wrapper2.pack(fill="both", expand='yes', padx=20, pady=10)
style = ttk.Style()
style.configure("Treeview.Heading", font=("rockwell", 15, "bold"), fg="blue")
trv = ttk.Treeview(wrapper1, columns=(1, 2, 3, 4, 5, 6,7), show="headings",
height=15)
trv.pack()
trv.heading(1, text="Customer_id:")
trv.heading(2, text="Item_Name:")
trv.heading(3, text="Price:")
trv.heading(4, text="Quantity:")
trv.heading(5, text="Amount:")
trv.heading(6, text="Date:")
trv.heading(7, text="Time:")
query = "select * from Order_History where cid={}".format(cid)
cursordb.execute(query)
rows = cursordb.fetchall()
update(rows)
# search:
lbl = Label(wrapper2, text="Enter Date To Search:", font=("arial", 12, 'bold'))
lbl.pack(side=tk.LEFT, padx=10)
entry = Entry(wrapper2, textvariable=q)
entry.pack(side=tk.LEFT, padx=6)
butt = Button(wrapper2, text='Search', command=search)
butt.pack(side=tk.LEFT, padx=6)
answer = Label(wrapper1, text="", font=("arial", 16, 'bold'))
answer.pack(pady=20)

```

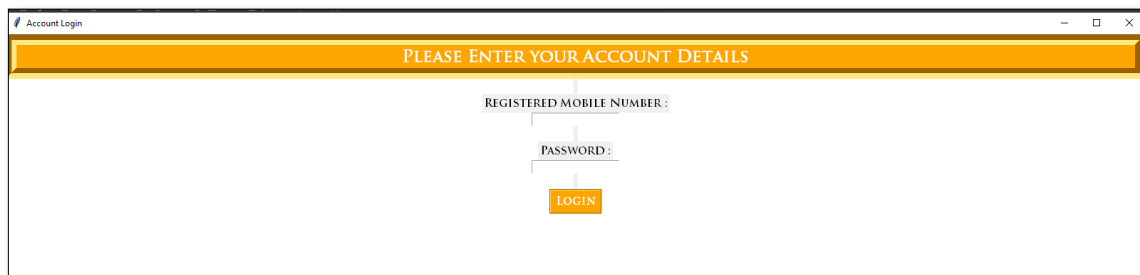
```
root.title("Order history")
root.geometry("800x700")
root.mainloop()
```

Output

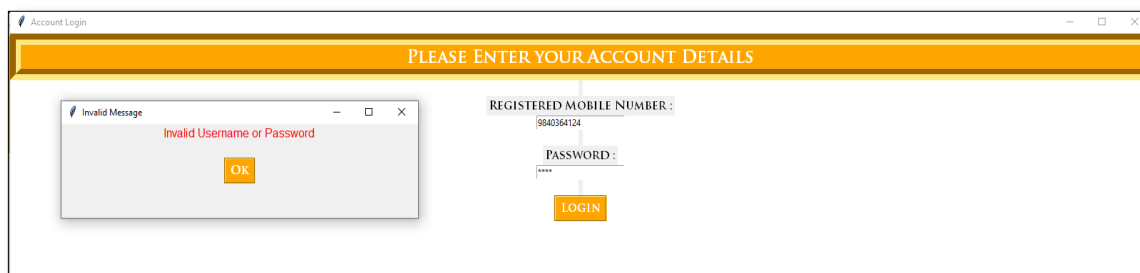
Main Screen



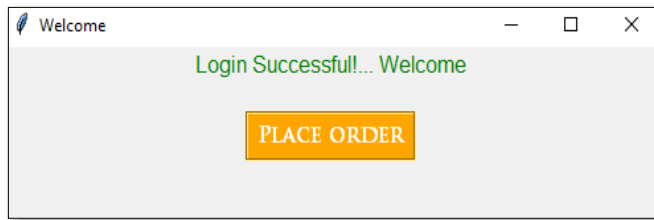
Case1: If user is a registered customer, then on clicking “Login” button on main screen, Login page is displayed.



Invalid username or password popup message



Upon entering valid details, customer is Logged in and below popup message is displayed



Upon clicking “PLACE ORDER” button, user will be taken to Menu Screen

Case2: If user is a registering for the first time, then on clicking “Register” button on main screen, Registration page is displayed.

 A screenshot of a web application window titled 'Register'. The window has a light gray background. At the top, there is a yellow banner with the text 'PLEASE ENTER DETAILS BELOW' in black capital letters. Below the banner, there are several input fields arranged in two columns. The fields are labeled: 'FIRST NAME *', 'LAST NAME *', 'ADDRESS *', 'CITY *', 'POSTAL CODE *', 'NEARBY LANDMARK *', 'MOBILE NUMBER *', 'PASSWORD *', 'CARD NUMBER *', and 'CVV *'. Each field has an asterisk indicating it is compulsory. A small orange button labeled 'REGISTER' is located at the bottom right. A validation error popup is displayed in the center of the form. The popup is titled 'Information' and contains a blue circular icon with a white exclamation mark. The text inside the popup says 'Please enter first name to Proceed'. There is an 'OK' button at the bottom of the popup.

Entry in every field is compulsory as shown above.

 A screenshot of the same 'Register' form as above, but with sample data entered into all the compulsory fields. The data is as follows:

Field	Value
FIRST NAME *	Gautham
LAST NAME *	US
ADDRESS *	S2_vasantham apt
CITY *	chennai
POSTAL CODE *	600042
NEARBY LANDMARK *	kb bank
MOBILE NUMBER *	9500026205
PASSWORD *	*****
CARD NUMBER *	1234567890123454
CVV *	***

 The 'REGISTER' button remains at the bottom right.

Now the above entries are inserted into the customer table upon clicking the “Register” button. If another customer has entered the same “MOBILE NUMBER” the following popup is displayed:

The image shows a web application window titled "Register". At the top, there is a yellow banner with the text "PLEASE ENTER DETAILS BELOW". The form contains several input fields arranged in two columns. The left column includes fields for "FIRST NAME *" (containing "Sai"), "LAST NAME *" (containing "anuraag"), "ADDRESS *" (containing "adyar"), "CITY *" (containing "chennai"), "POSTAL CODE *" (containing "600028"), and "NEARBY LANDMARK *" (containing "rohini theatre"). The right column includes fields for "MOBILE NUMBER" (containing "9500026205"), "PASSWORD *" (containing "*****"), "CARD NUMBER *" (containing "1234567890111111"), and "CVV *" (containing "*****"). A yellow "REGISTER" button is located at the bottom right of the form. An "Information" popup dialog is displayed in the center, with a blue information icon and the message "Phone number already exists !". The dialog has an "OK" button.

Field Label	Value
FIRST NAME *	Sai
LAST NAME *	anuraag
ADDRESS *	adyar
CITY *	chennai
POSTAL CODE *	600028
NEARBY LANDMARK *	rohini theatre
MOBILE NUMBER	9500026205
PASSWORD *	*****
CARD NUMBER *	1234567890111111
CVV *	*****

Once the valid details are entered the following popup message is displayed.

The image shows a web application window titled "Welcome". The main content area displays the text "Registered Successfully!... Welcome" in green. Below this text is a yellow button with the text "PLACE ORDER".

Menu Screen(item selected and total is calculated for the same)

Yummy Corner

MENU

Burgers & Wraps

☐ Spicy chicken double patty burger ₹250

☐ Paneer double patty burger ₹246

☒ Veg burger ₹160

☐ Panner wrap ₹125

☐ aloo wrap ₹100

Drinks

☐ coke ₹92

☐ sprite ₹82

☐ miranda ₹85

☐ 7 up ₹73

Sides

☐ hash brown ₹55

☐ garlic bread ₹65

☐ french fries(m) ₹75

☐ french fries(L) ₹90

Sandwich

☐ cheese corn delight sandwich ₹250

☐ chicken tandoori sandwich ₹270

☐ veg grilled sandwich ₹220

☐ Extra cheese paneer sandwich ₹280

Desserts

☐ choco lava ₹80

☐ strawberry icecream ₹90

☐ pancakes & syrup ₹110

☐ chocolate icecream ₹100

☐ vanilla milkshake ₹60

CHECK TOTAL

RESET

ORDER HISTORY

LOGOUT

Sub Total:

160.0

CGST:

14.4

SGST:

14.4

Total:

188.8

PLACE ORDER

Order summary page

Yummy Corner

MENU

Burgers & Wraps

☐ Spicy chicken double patty burger ₹250

☐ Paneer double patty burger ₹246

☒ Veg burger ₹160

☐ Panner wrap ₹125

☐ aloo wrap ₹100

Drinks

☐ coke ₹92

☐ sprite ₹82

☐ miranda ₹85

☐ 7 up ₹73

Sides

☐ hash brown ₹55

☐ garlic bread ₹65

☐ french fries(m) ₹75

☐ french fries(L) ₹90

Sandwich

☐ cheese corn delight sandwich ₹250

☐ chicken tandoori sandwich ₹270

☐ veg grilled sandwich ₹220

☐ Extra cheese paneer sandwich ₹280

Desserts

☐ choco lava ₹80

☐ strawberry icecream ₹90

☐ pancakes & syrup ₹110

☐ chocolate icecream ₹100

☐ vanilla milkshake ₹60

CHECK TOTAL

RESET

ORDER HISTORY

LOGOUT

Sub Total:

160.0

CGST:

14.4

SGST:

14.4

Total:

188.8

PLACE ORDER

Order Summary

Item Name:

Veg Burger

Price:

160

Quantity:

1

Select Payment Method:

☐ Cash
 ☐ Credit Card

CONFIRM ORDER

67 | Page

Yummy Corner

MENU

Burgers & Wraps

Spicy chicken

Paneer double

☒ Veg burger

Paneer wrap

aloo wrap

cheese corn d

chicken tando

veg grilled san

Extra cheese p

choco lava

strawberry icec

pancakes & syrup

chocolate icecream

vanilla milkshake

Drinks

coke

sprite

miranda

7 up

Sides

hash brown

garlic bread

french fries(m)

french fries(L)

Order Summary

Item Name:	Price:	Quantity:
Veg Burger	160	1

Select Payment Method:

☐ Cash
 ☒ Credit Card

ENTER YOUR CVV: 1234

CONFIRM ORDER

Sub Total:

CGST:

SGST:

Total:

160.0

14.4

14.4

188.8

LOGOUT

PLACE ORDER

Wrong CVV entered in the entry given in order summary page

Popup giving error message

Yummy Corner

MENU

Burgers & Wraps

Spicy chicken double patty burger

Paneer double patty burger

☒ Veg burger

Paneer wrap

aloo wrap

cheese corn delight sandwich

chicken tandoori sandwich

veg grilled sandwich

Extra cheese paneer sandwich

choco lava

strawberry icecream

pancakes & syrup

chocolate icecream

vanilla milkshake

Drinks

coke

sprite

miranda

7 up

Sides

hash brown

garlic bread

french fries(m)

french fries(L)

Order Summary

Item Name:	Price:	Quantity:
Veg Burger	160	1

Select Payment Method:

☐ Cash
 ☒ Credit Card

ENTER YOUR CVV: 1234

CONFIRM ORDER

Sub Total:

CGST:

SGST:

Total:

160.0

14.4

14.4

188.8

LOGOUT

PLACE ORDER

Information

CVV entered is not correct. Please check and enter again!

OK

Yummy Corner

MENU

Burgers & Wraps

Spicy chicken double patty burger ₹250 0

Paneer double patty burger ₹246 0

Veg burger ₹160 0

Panner wrap ₹125 0

aloo wrap ₹100 0

Drinks

coke ₹92 0

sprite ₹82 0

miranda ₹85 0

7 up ₹73 0

Sides

hash brown ₹55 0

garlic bread ₹65 0

french fries(m) ₹75 0

french fries(L) ₹90 0

Sandwich

cheese corn delight sandwich ₹250 0

chicken tandoori sandwich ₹270 0

veg grilled sandwich ₹220 0

Extra cheese paneer sandwich ₹280 0

Desserts

choco lava ₹80 0

strawberry icecream ₹90 0

pancakes & syrup ₹110 0

chocolate icecream ₹100 0

vanilla milkshake ₹60 0

CHECKOUT

RESET

ORDER HISTORY

LOGOUT

Ordered confirmed

Thank you for ordering in Yummy corner! Visit us again!

OK

Sub Total: 0

CGST: 0

SGST: 0

Total: 0

PLACE ORDER

Order is placed upon giving the correct cvv with a pop message confirming the same.

Order History page

Order history

Order History:

	Customer_id:	Item_Name:	Price:	Quantity:	Amount:	Date:	Time:
2		Spicy chicken double patty burger	250	2.0	500.0	06-09-2021	19:13:36
2		Panner wrap	125	1.0	125.0	06-09-2021	19:13:36
2		veg grilled sandwich	220	2.0	440.0	06-09-2021	19:13:36
2		choco lava	250	1.0	80.0	06-09-2021	19:13:36
2		sprite	82	3.0	246.0	06-09-2021	19:13:36
2		hash brown	55	2.0	110.0	06-09-2021	19:13:36
2		Panner double patty burger	246	1.0	246.0	06-09-2021	19:19:12
2		sprite	82	1.0	82.0	06-09-2021	19:19:12
2		Panner double patty burger	246	4.0	984.0	06-09-2021	19:51:34
2		sprite	82	2.0	164.0	06-09-2021	22:21:22
2		Spicy chicken double patty burger	250	3.0	750.0	06-09-2021	22:32:39
2		Spicy chicken double patty burger	250	2.0	500.0	06-09-2021	22:34:02
2		Spicy chicken double patty burger	250	3.0	750.0	06-09-2021	22:37:13
2		Spicy chicken double patty burger	250	2.0	500.0	06-09-2021	22:38:00
2		Spicy chicken double patty burger	250	2.0	500.0	06-09-2021	22:43:50

Search:

Enter Date To Search: Search

On giving any valid date, the page shows that particular customers's order on the given date.

Order history

Order History:

	Customer_id:	Item_Name:	Price:	Quantity:	Amount:	Date:	Time:
2		Spicy chickendoublepatty burger	250	3.0	750.0	07-09-2021	12:30:54
2		Spicy chickendoublepatty burger	250	2.0	500.0	07-09-2021	23:05:15
2		French fries(m)	75	2.0	150.0	07-09-2021	23:06:36

Search:

Enter Date To Search: 07-09-2021 Search

Invalid characters given in entry specified for date of order

Order history

Order History:

	Customer_id:	Item_Name:	Price:	Quantity:	Amount:	Date:	Time:
2		Spicy chickendoublepatty burger	250	2.0	500.0	06-09-2021	19:13:36
2		Panner wrap	125	1.0	125.0	06-09-2021	19:13:36
2		veg grilled sandwich	220	2.0	440.0	06-09-2021	19:13:36
2		choco lava	250	1.0	80.0	06-09-2021	19:13:36
2		sprite	82	3.0	246.0	06-09-2021	19:13:36
2		hash brown	55	2.0	110.0	06-09-2021	19:13:36
2		Panner double patty burger	246	1.0	246.0	06-09-2021	19:19:12
2		sprite	82	1.0	82.0	06-09-2021	19:19:12
2		Panner double patty burger	246	4.0	984.0	06-09-2021	19:51:34
2		sprite	82	2.0	164.0	06-09-2021	22:21:22
2		Spicy chickendoublepatty burger	250	3.0	750.0	06-09-2021	22:32:39
2		Spicy chickendoublepatty burger	250	2.0	500.0	06-09-2021	22:34:02
2		Spicy chickendoublepatty burger	250	3.0	750.0	06-09-2021	22:37:13
2		Spicy chickendoublepatty burger	250	2.0	500.0	06-09-2021	22:38:00
2		Spicy chickendoublepatty burger	250	2.0	500.0	06-09-2021	22:43:50

Enter correct date in DD-MM-YYYY format

Search:

Enter Date To Search: 07/09/21 Search

Invalid date (Eg:30/2/2021)

Order history

Order History:

Customer_id:	Item_Name:	Price:	Quantity:	Amount:	Date:	Time:
--------------	------------	--------	-----------	---------	-------	-------

Oops!! Invalid Date Entered

Search:

Enter Date To Search: 30-02-2021 Search

Pop Window with confirmation to Logout

Yummy Corner

MENU

Burgers & Wraps

- ☐ Spicy chicken double patty burger ₹250 0
- ☐ Paneer double patty burger ₹246 0
- ☐ Veg burger ₹160 0
- ☐ Panner wrap ₹125 0
- ☐ aloo wrap ₹100 0

Drinks

- ☐ coke ₹92 0
- ☐ sprite ₹82 0
- ☐ miranda ₹85 0
- ☐ 7 up ₹73 0

Sides

- ☐ hash brown ₹55 0
- ☐ garlic bread ₹65 0
- ☐ french fries(m) ₹75 0
- ☐ french fries(L) ₹90 0

Sandwich

- ☐ cheese corn delight sandwich ₹250 0
- ☐ chicken tandoori sandwich ₹270 0
- ☐ veg grilled sandwich ₹220 0
- ☐ Extra cheese paneer sandwich ₹280 0

Desserts

- ☐ choco lava ₹80 0
- ☐ strawberry icecream ₹90 0
- ☐ pancakes & syrup ₹110 0
- ☐ chocolate icecream ₹100 0
- ☐ vanilla milkshake ₹60 0

CHECK TOTAL

RESET

LOGOUT

Sub Total: 0

CGST: 0

SGST: 0

Total: 0

PLACE ORDER

Confirm Logout

Are you sure you want to Logout?

Yes No

Conclusion

Online e-food delivery has a lot of benefits and it has become popular and being developed in the recent years. Online delivery is more convenient, cost effective and has a wider reach towards people, E-food delivery database has a great scope in the near future, when developed it can be used as a full-fledged food ordering application with an active order tracking facility like Swiggy and Zomato.

Bibliography

- <https://www.geeksforgeeks.org/python-tkinter-tutorial/>
- <https://www.youtube.com/watch?v=je1LcqXQAhU>
- <https://www.w3schools.com/>
- <https://www.python.org/>
- <https://www.wikipedia.org/>