



VIT[®]

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering (SCOPE)

B. Tech. Winter 2020-2021

CSE2004: Database Management Systems

TITLE: ONLINE FOOD DELIVERY SYSTEM

Course Faculty

M.Braveen

Batch Members

1.Chandrakiran Reddy(20BCE1936)

2.Ashwin KS(20BCE1733)

3.Kesav Santhosh(20BCE1105)

A project report submitted to

M BRAVEEN

SCOPE

Vandalur – Kelambakkam Road

Chennai – 600127

JUN 2021

Table of Contents

S. NO.	TITLE	PAGE NUMBER
1	ABSTRACT	3
2	ACKNOWLEDGEMENT	4
3	INTRODUCTION	5
4	ER DIAGRAM	6
5	ER TO RELATIONAL MAPPING	7
6	IMPLEMENTATION	8
7	EXPERIMENTAL SETUP(HARDWARE & SOFTWARE SPECS.)	9
8	CONCLUSION AND FUTURE WORK	10
9	REFERENCES	11
10	SQL QUERIES	12
11	RELEVANT SCREENSHOTS	16
12	SOURCE CODE	21

Abstract

The working areas used to have everything as manual ,until the Covid crisis . Now everything is online . So , through this project we are trying to bring food management also online . Through this project ” Online food ordering system ” we are providing a platform for users to order food online with less time and efforts . The platform modifies , records , stores and channelizes the transaction by computer . It is even more easier than ordering food in a restaurant/hotel . Computer is reliable in today’s era and there is no chance of duplication , so the working will be smoother. There is no chaos and confusion while using the online system compared to the book system.

ACKNOWLEDGEMENT

We wish to express our sincere thanks and deep sense of gratitude to our project guide, M BRAVEEN, SCOPE, for his consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work.

We also take this opportunity to thank all the faculty of the School for their support and their wisdom imparted to us throughout the course.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

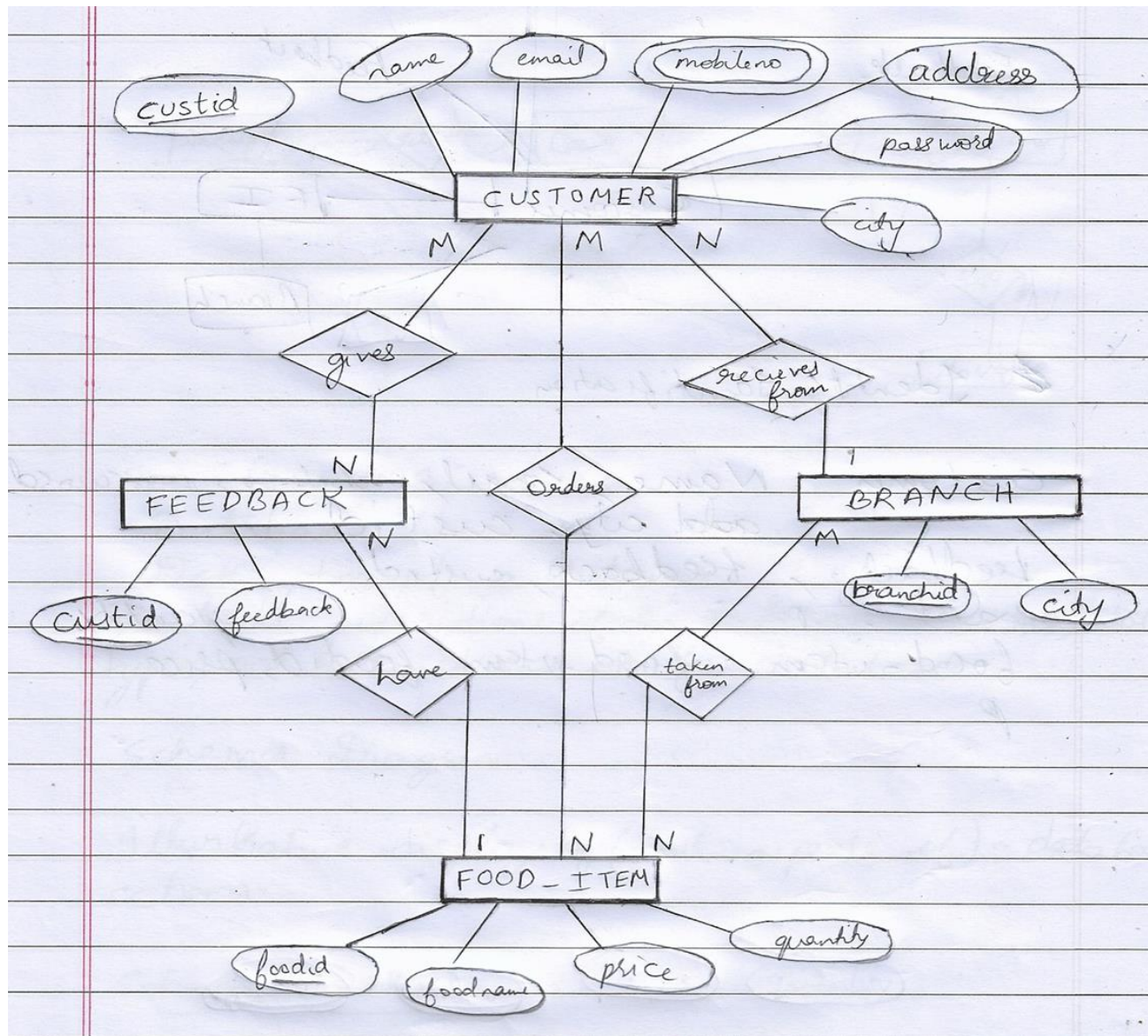
Introduction

The Online Ordering System can be defined as a simple and convenient way for customers to purchase food online , without having to go to the restaurant . This system is enabled by the internet – it is the internet that connects the restaurant or the food company on one hand, and the customer on other hand .

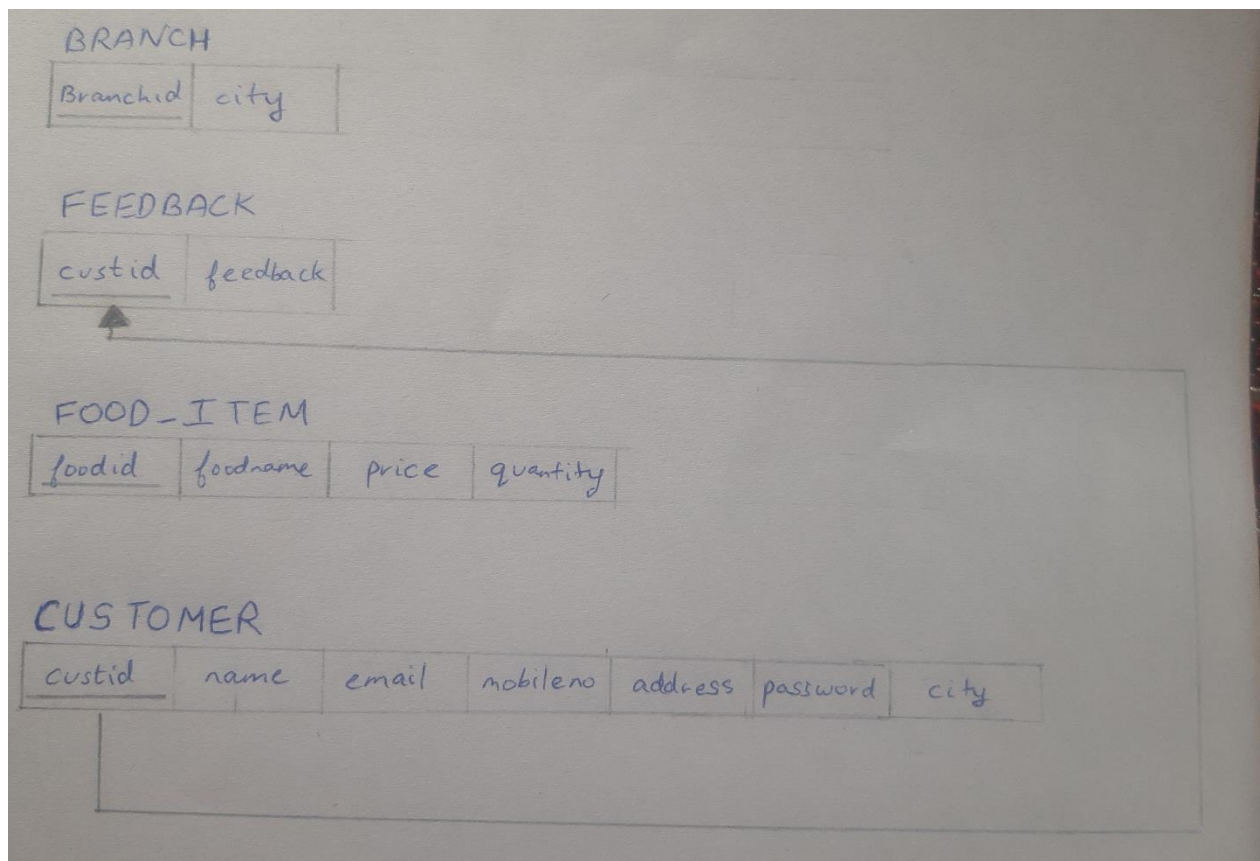
Therefore , as per this system , the customer visits the restaurant's app or website , browses through the various food items , combos and cuisines available there and goes ahead and selects and purchases the items he or she needs . These items will then be delivered to the customer at his or her doorstep at the time they choose by a delivery person .

Payments for such online orders can be made through debit cards , credit cards , cash or card on delivery , or even through digital wallets . This system for online food delivery is completely safe , secure and is a very popular method that is revolutionizing the way in which the food industry operates.

ER Diagram



ER to Relational mapping



IMPLEMENTATION(DESCRIPTION OF TABLES)

```
mysql> desc feedback;
```

Field	Type	Null	Key	Default	Extra
custid	int(11)	NO	PRI	NULL	
feedback	varchar(15)	YES		NULL	

```
2 rows in set (0.01 sec)
```

```
mysql> desc food_item;
```

Field	Type	Null	Key	Default	Extra
foodid	int(11)	NO	PRI	NULL	
foodname	varchar(20)	YES		NULL	
price	int(11)	YES		NULL	
quantity	int(11)	YES		NULL	

```
4 rows in set (0.01 sec)
```

```
mysql> desc otptable;
```

Field	Type	Null	Key	Default	Extra
otpid	int(11)	YES		NULL	
otp	varchar(10)	YES		NULL	

```
2 rows in set (0.01 sec)
```

```
mysql> desc phoneno;
```

Field	Type	Null	Key	Default	Extra
phoneid	int(11)	YES		NULL	
phonenum	varchar(20)	YES		NULL	

```
2 rows in set (0.01 sec)
```

```
mysql>
```


EXPERIMENTAL SETUP

Hardware used

- Mouse
- Printer
- Laptop (8 GB RAM, i5 Processor, 1TB ROM,
- Pen Drive

Software used

- Python IDLE version 3.9
- Mysql server 5.1

CONCLUSION AND FUTURE WORK

Today the internet and its boom have created a new economic scenario that not only stresses on the classical concept of the “product” but also on the modern concept of “service” . As people become more accustomed to using the internet , they view ordering food products online as a time-saving and cost-saving experience , which is the very essence of our online food ordering system . Maintaining the deliverable goods as well as services through single or multiple windows is also on the agenda . To play an Anchor’s role in the Food Ordering System and to consolidate the entity of hotel which facilitates food orders , helping it merge to the larger society as an institution of importance. Developing the distinct function of the food ordering centre towards the order and delivery of food items . Building up – and inspiring – a body of focused research into the architecture, aesthetic and design aspects of hotel with food order facility . Encouraging the development and spread of sophisticated management paradigm and maintenance method . Being a knowledge resource centre that collects , analyses and disseminates information on techniques of profitable operation .

REFERENCES

- A Textbook of Computer Science with Python for Class 11 and 12 by Sumita Arora
- www.geeksforgeeks.org
- <https://www.w3resource.com/>
- <https://github.com/>
- <https://stackoverflow.com/>
- <https://www.tutorialspoint.com/>
- <https://wiki.python.org/moin/TkInter>
- Faculty

APPENDIX I(SQL QUERIES)

```
mysql> create table branch(branchid int primary key,city  
varchar(15));
```

Query OK, 0 rows affected (0.02 sec)

```
mysql> insert into branch values(1,'chennai');
```

Query OK, 1 row affected (0.00 sec)

```
mysql> insert into branch values(2,'mumbai');
```

Query OK, 1 row affected (0.00 sec)

```
mysql> insert into branch values(3,'delhi');
```

Query OK, 1 row affected (0.01 sec)

```
mysql> insert into branch values(4,'kolkata');
```

Query OK, 1 row affected (0.00 sec)

```
mysql> insert into branch values(5,'bangalore');
```

Query OK, 1 row affected (0.01 sec)

```
mysql> create table customer(custid int primary key,name  
varchar(10),email varchar(20),mobile_no int,password  
varchar(10),address varchar(20),city varchar(10));
```

Query OK, 0 rows affected (0.03 sec)

```
mysql> create table feedback(custid int primary key,feedback  
varchar(15));
```

Query OK, 0 rows affected (0.03 sec)

```
mysql> insert into feedback values(1,'good');
```

Query OK, 1 row affected (0.01 sec)

```
mysql> create table food_item(foodid int primary key,foodname  
varchar(20),price int,quantity int);
```

Query OK, 0 rows affected (0.02 sec)

```
mysql> insert into food_item values(1,'veg pizza',59,26);
```

Query OK, 1 row affected (0.01 sec)

```
mysql> insert into food_item values(2,'burger',63,43);
```

Query OK, 1 row affected (0.00 sec)

```
mysql> insert into food_item values(3,'vada',63,43);
```

Query OK, 1 row affected (0.01 sec)

```
mysql> insert into food_item values(4,'idli',18,96);
```

Query OK, 1 row affected (0.01 sec)

```
mysql> insert into food_item values(5,'masala dosa',40,38);
```

Query OK, 1 row affected (0.01 sec)

```
mysql> insert into food_item values(6,'poori',55,100);
```

Query OK, 1 row affected (0.00 sec)

```
mysql> drop table branch1;
```

Query OK, 0 rows affected (0.01 sec)

```
mysql> drop table customers;
```

Query OK, 0 rows affected (0.01 sec)

```
mysql> drop table feedback1;
```

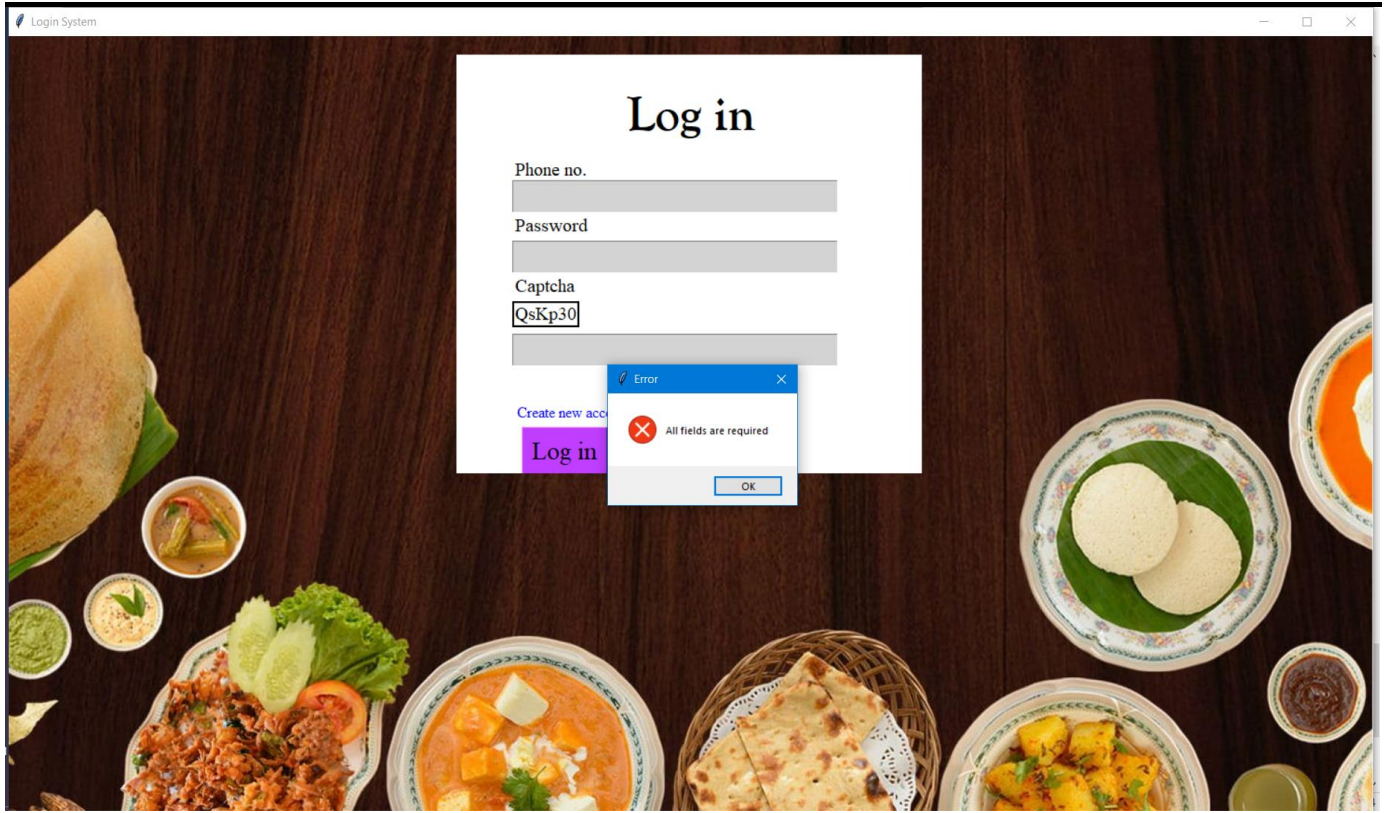
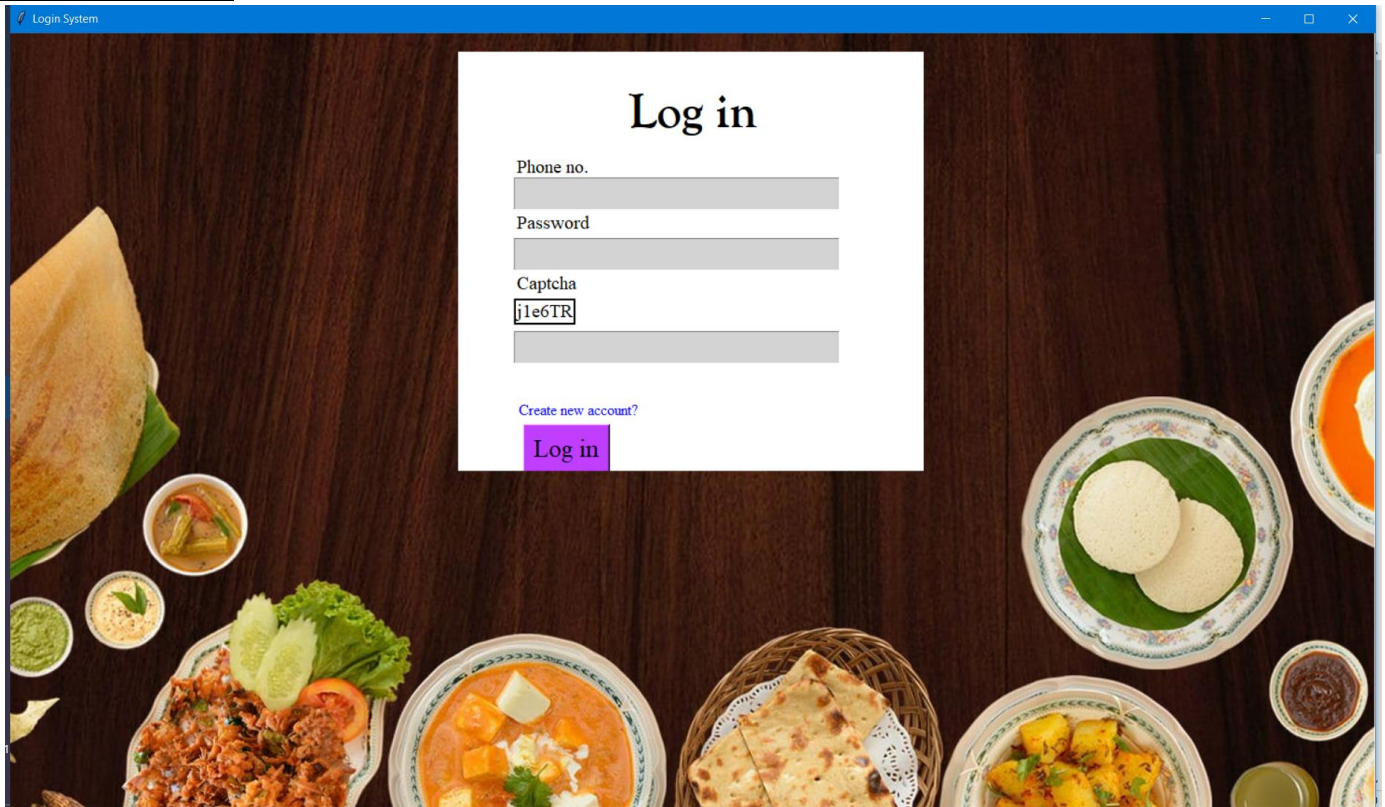
Query OK, 0 rows affected (0.01 sec)

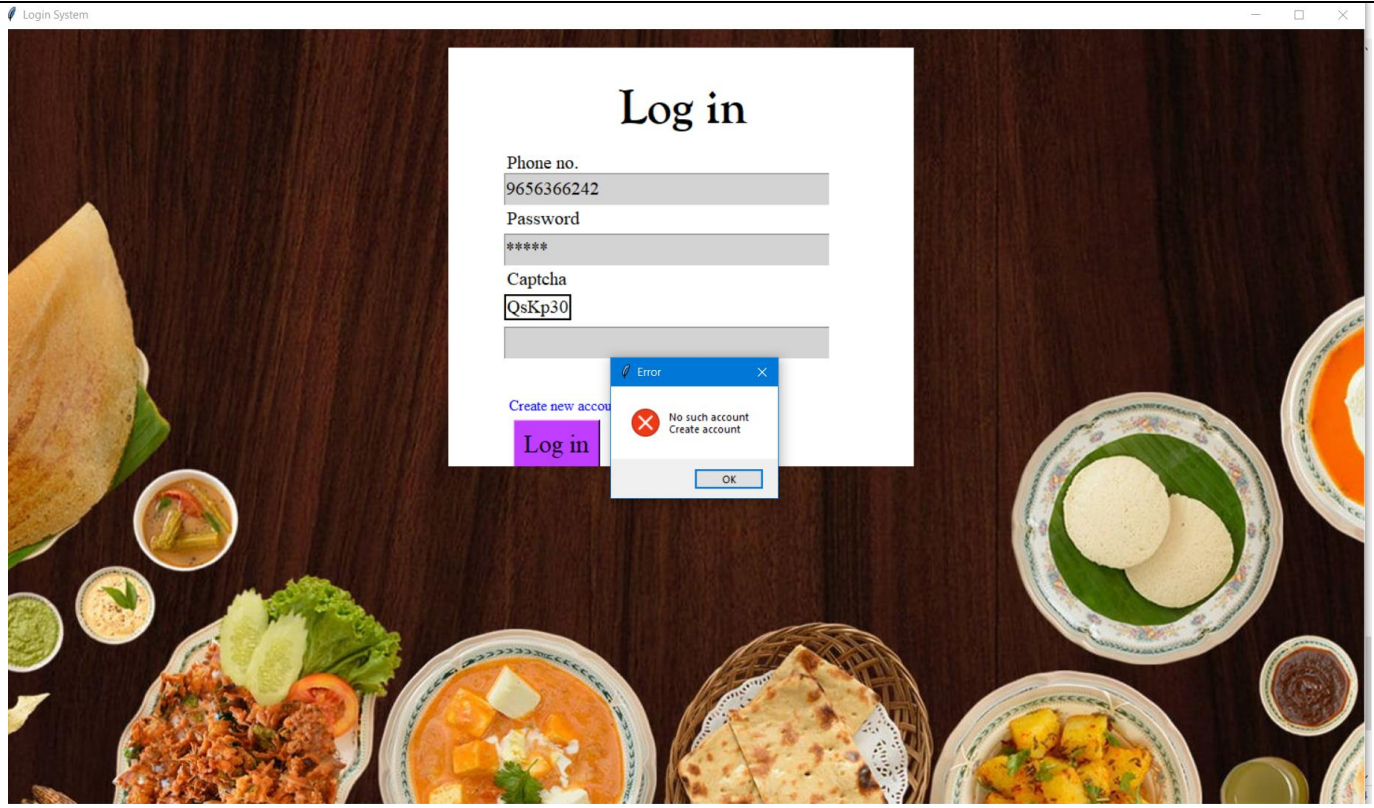
```
mysql> drop table food_item1;
```

Query OK, 0 rows affected (0.01 sec)

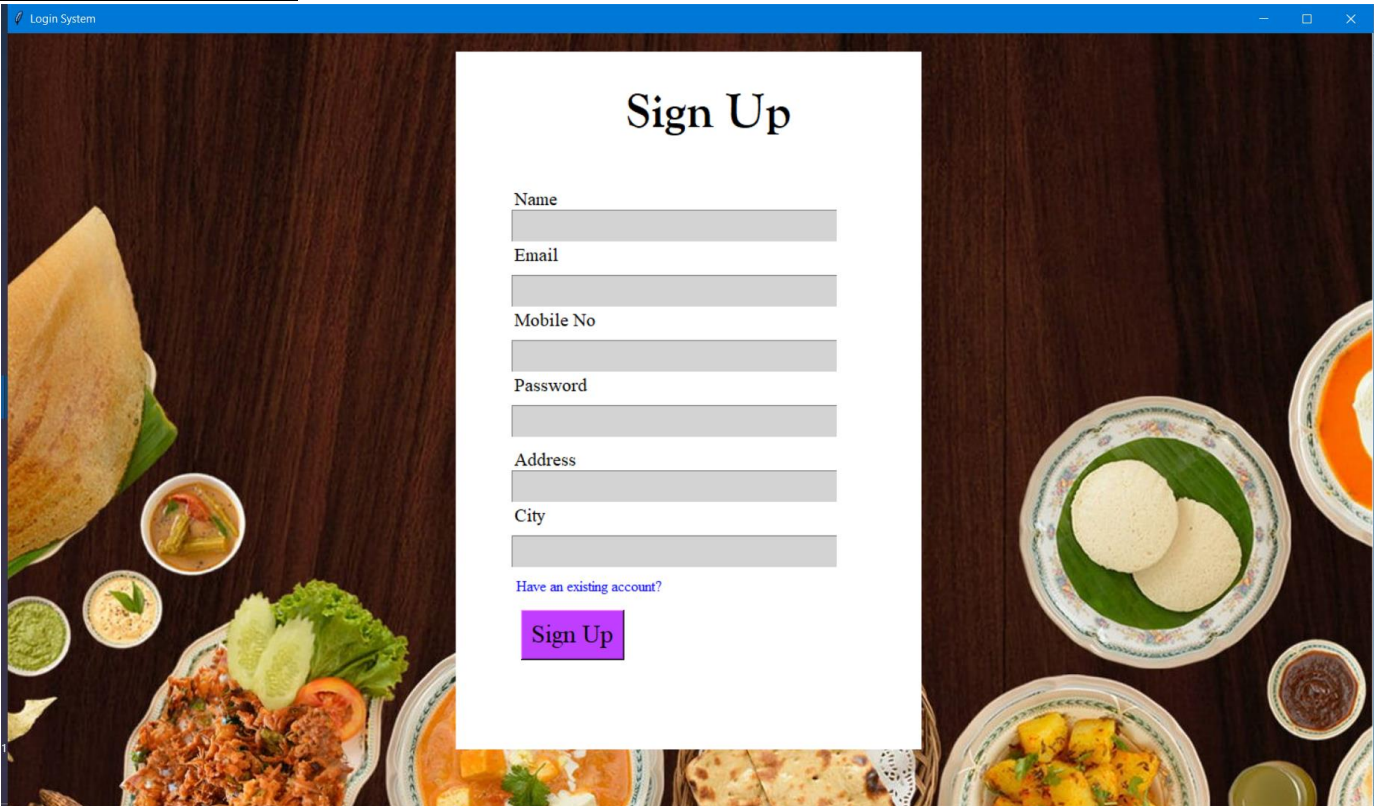
APPENDIX II (Screenshots)

LOGIN PAGE

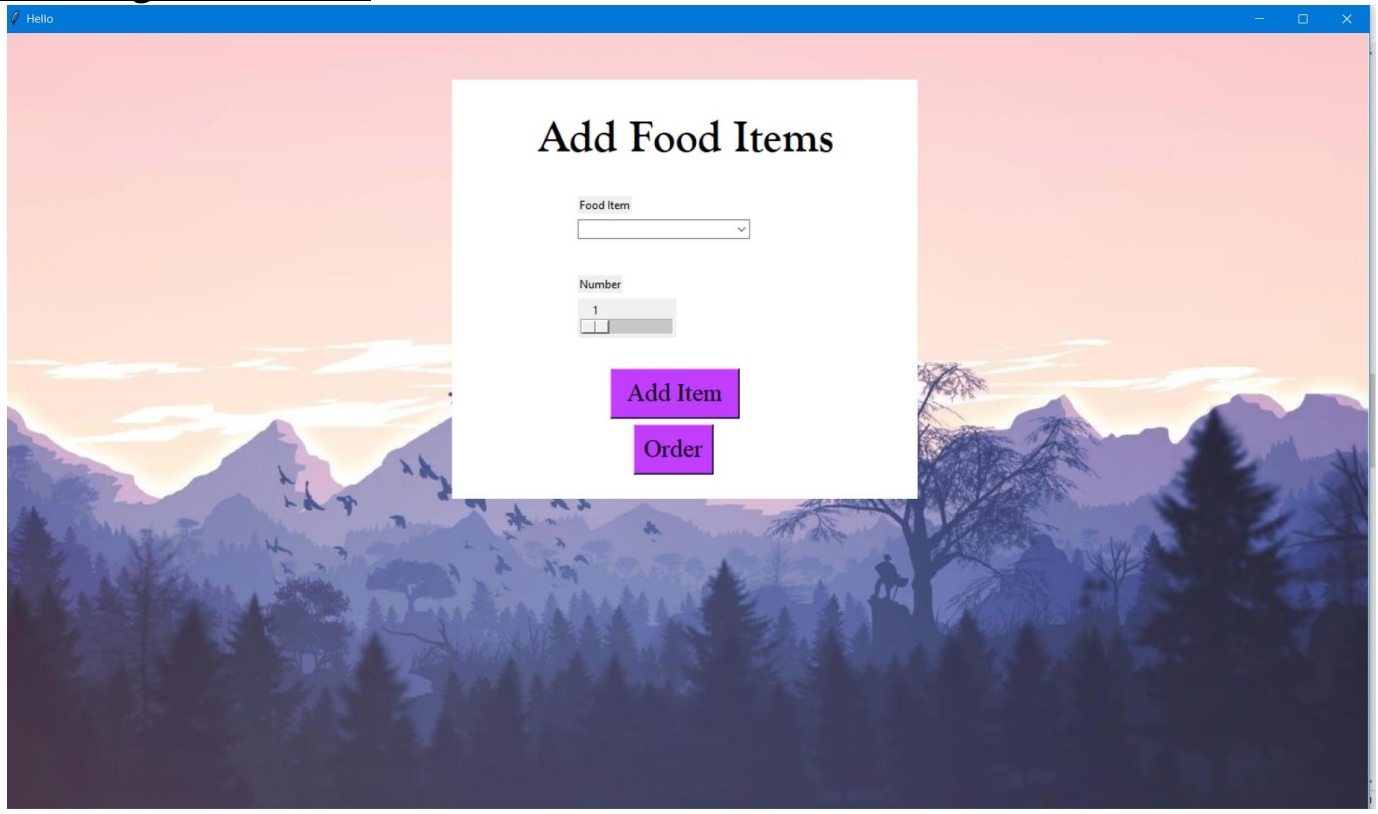




SIGN UP PAGE



Selecting food items



Add Food Items

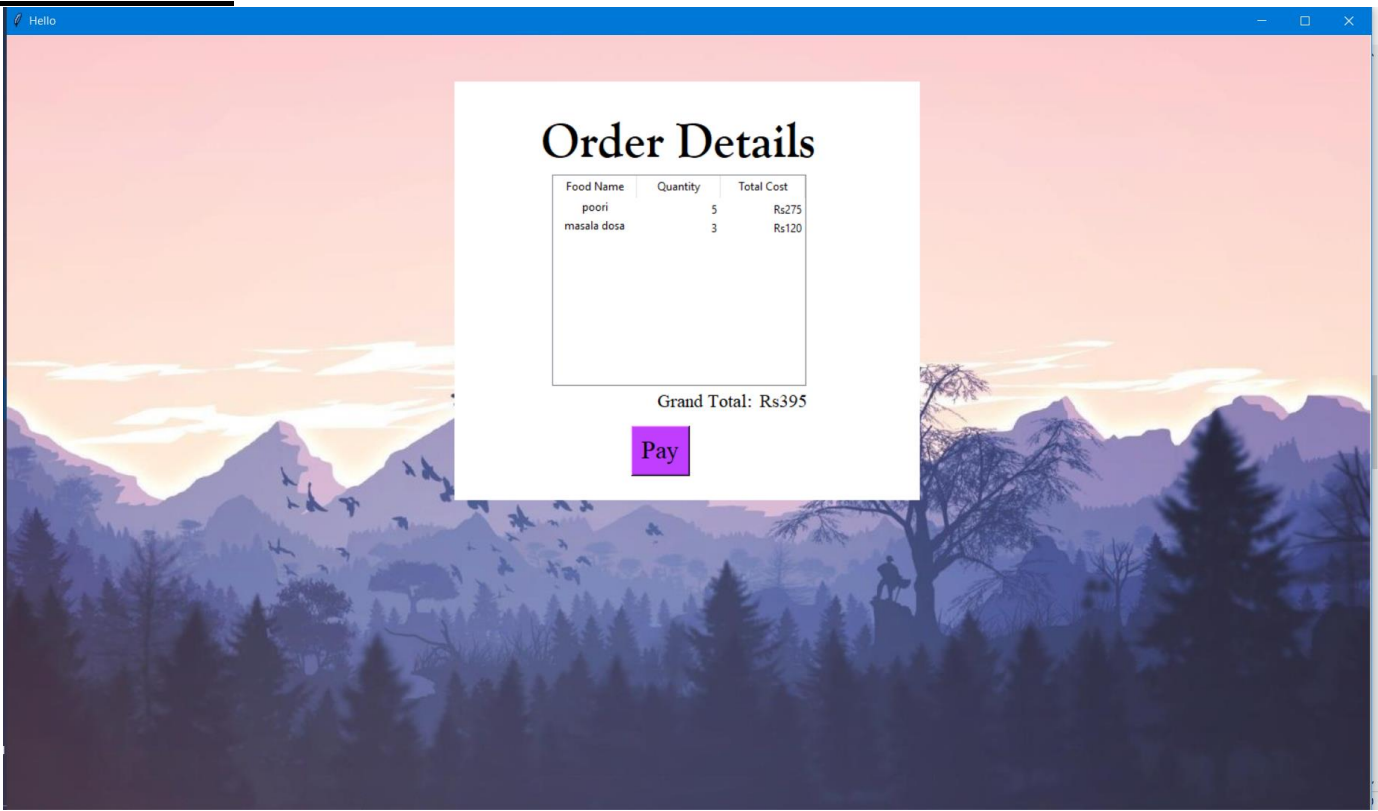
Food Item

Number

Add Item

Order

Order details



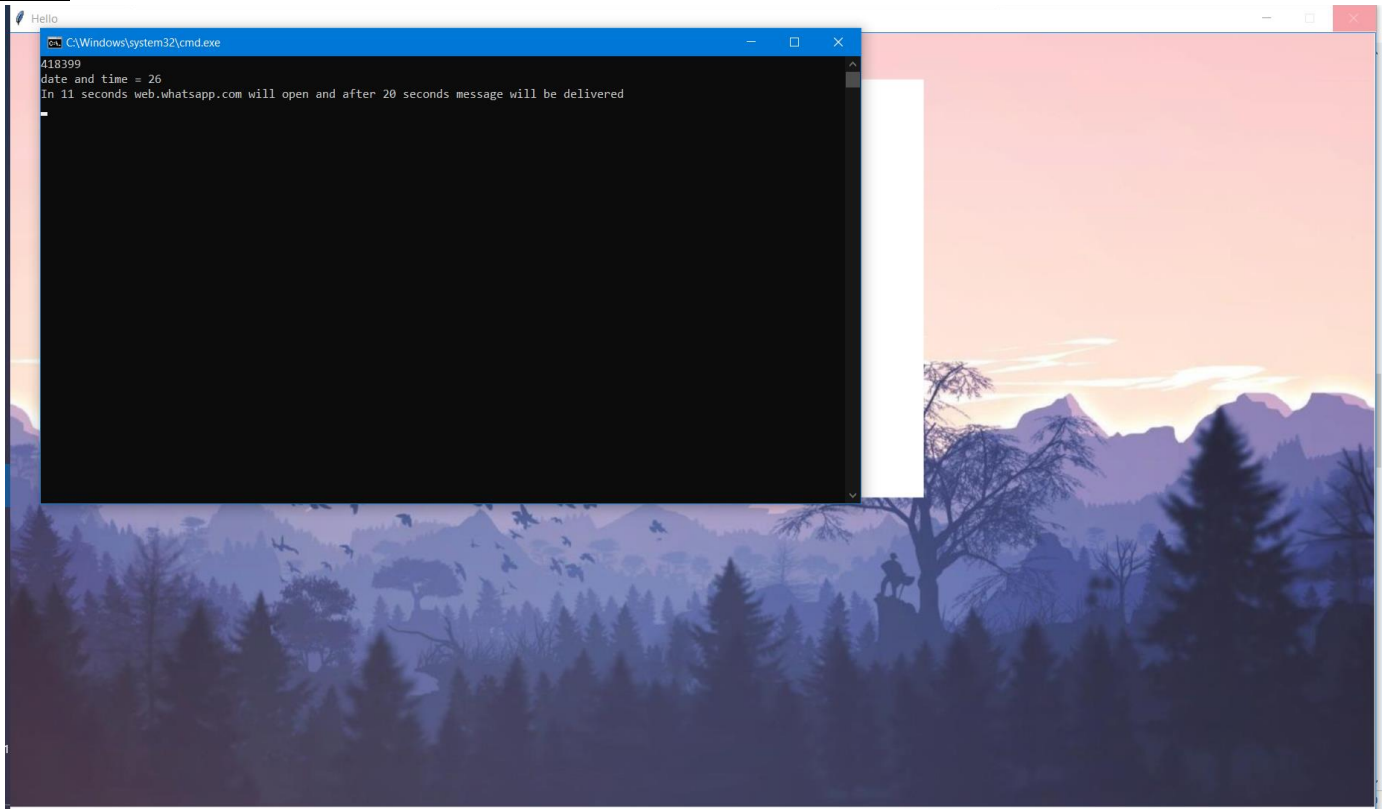
Order Details

Food Name	Quantity	Total Cost
poori	5	Rs275
masala dosa	3	Rs120

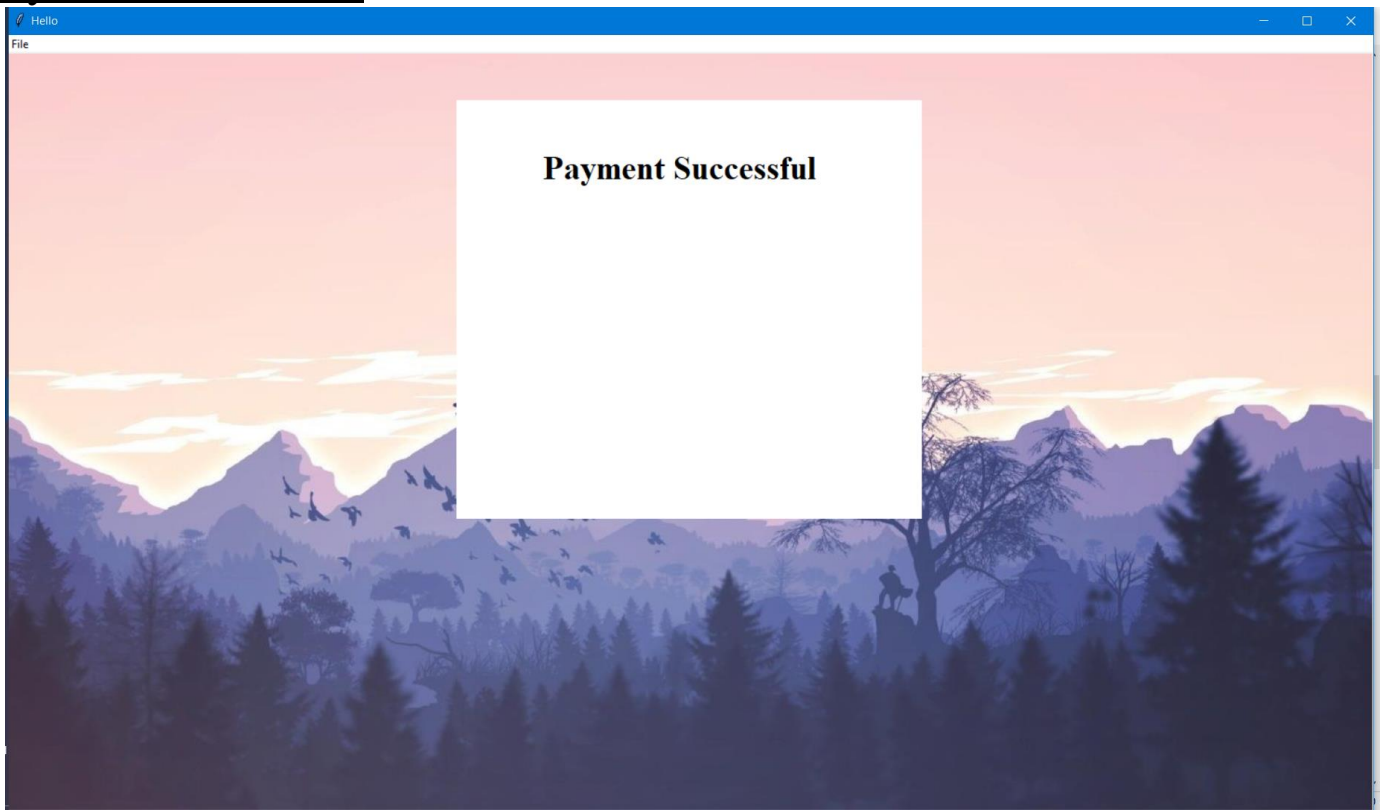
Grand Total: Rs395

Pay

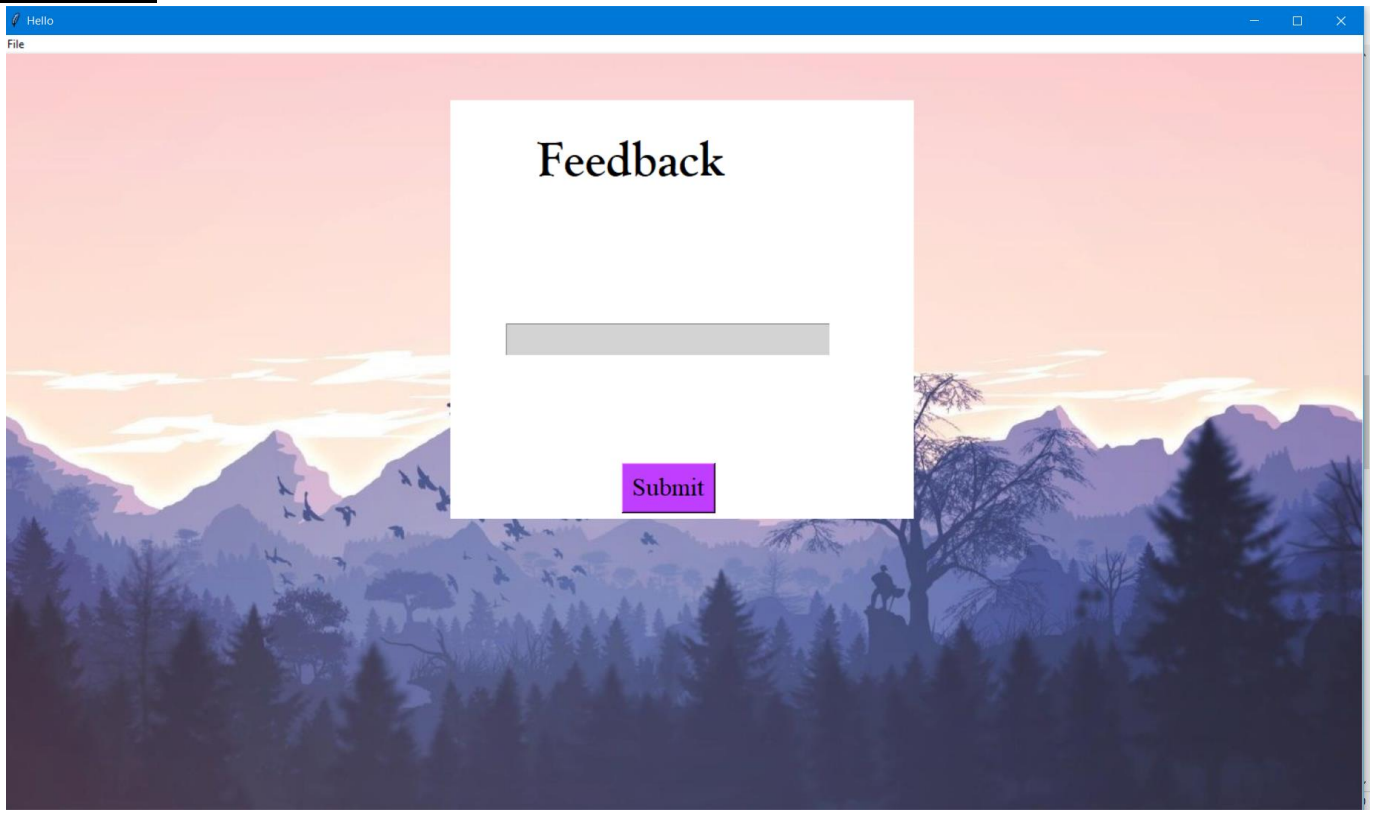
OTP



Payment Successful



Feedback



SOURCE CODE

```
#import statements
import mysql.connector as myconnect
import random
con=myconnect.connect(host='localhost',user='root',password='kesavmelo2002',database='food')
from tkinter import *
from PIL import ImageTk,Image
from tkinter import messagebox,ttk
import os
class Login:
    def __init__(self,root): #PAGE 1
        self.root=root
        self.root.title("Login System")
        self.root.geometry("1470x1000")
        b=chr(random.choice(range(48,58)))+chr(random.choice(range(48,58)))+chr(random.choice(range(65,91)))+chr(random.choice(range(65,91)))+chr(random.choice(range(97,123)))+chr(random.choice(range(97,123)))
        b=list(b)
        random.shuffle(b)
        print(b)
        global c
        c=""
        for i in range(0,len(b)):
            c=c+b[i]
        print(c)
        self.bg=ImageTk.PhotoImage(file="images/image2.jpg")
        self.bg_image=Label(self.root,image=self.bg).place(x=0,y=0,relwidth=0.999,relheight=0.999)
        Frame_login=Frame(self.root,bg="white")
        Frame_login.place(x=485,y=20,height=450,width=500)
        title=Label(Frame_login,text="Log in",font=("Goudy old style",40,"bold"),fg="black",bg="white").place(x=180,y=30)
        title=Label(Frame_login,text="Phone no.",font=("times new roman",15),fg="black",bg="white").place(x=60,y=110)
```

```

self.txt_user=Entry(Frame_login,font=("times new
roman",15),bg="lightgray")
self.txt_user.place(x=60,y=135,width=350,height=35)
title=Label(Frame_login,text="Password",font=("times new
roman",15),fg="black",bg="white").place(x=60,y=170)
self.txt_pass=Entry(Frame_login,show='*',font=("times new
roman",15),bg="lightgray")
self.txt_pass.place(x=60,y=200,width=350,height=35)
title=Label(Frame_login,text="Captcha",font=("times new
roman",15),fg="black",bg="white").place(x=60,y=235)
title=Label(Frame_login,text=c,font=("times new
roman",15),fg="black",bg="white",relief="solid").place(x=60,y=265)
self.txt_captcha=Entry(Frame_login,font=("times new
roman",15),bg="lightgray")
self.txt_captcha.place(x=60,y=300,width=350,height=35)
forget_btn=Button(Frame_login,command=self.forget_function,cursor="hand2",
text="Create new account?",bg="white",fg="blue",bd=0,font=("times new
roman",12)).place(x=60,y=370)
login_btn=Button(Frame_login,command=self.login_function,cursor="hand2",
text="Log in",bg="DarkOrchid1",fg="black",font=("times new
roman",20)).place(x=70,y=400)

```

```

def login_function(self): #when clicking login button
    if(self.txt_user.get()!=""):
        a=int(self.txt_user.get())
    else:
        a=0
    b=self.txt_pass.get()
    global c
    query1="select * from customer where mobile_no={ }".format(a)
    mycur1=con.cursor()
    mycur1.execute(query1)
    row1=mycur1.fetchall()
    rc1=mycur1.rowcount
    query2="select * from customer where password='{ }'".format(b)
    mycur2=con.cursor()
    mycur2.execute(query2)
    row2=mycur2.fetchall()

```

```

rc2=mycur2.rowcount
if(self.txt_pass.get()==" or self.txt_user.get()==""):
    messagebox.showerror("Error","All fields are
required",parent=self.root)
elif(rc1==0 or rc2==0):
    messagebox.showerror("Error","No such account\nCreate
account",parent=self.root)
elif(row1==row2 and self.txt_captcha.get()==c):
    print("hello")
    self.page_function()
elif(self.txt_captcha.get()!=c):
    messagebox.showerror("Error","Wrong Captcha",parent=self.root)
global m0,m1
m0=[]
m1=[]

def exist_function(self): #PAGE 1
    self.root=root
    self.root.title("Login System")
    self.root.geometry("1470x1000")
    #image
b=chr(random.choice(range(48,58)))+chr(random.choice(range(48,58)))+chr(r
andom.choice(range(65,91)))+chr(random.choice(range(65,91)))+chr(random.c
hoice(range(97,123)))+chr(random.choice(range(97,123)))
    b=list(b)
    random.shuffle(b)
    print(b)
    global c
    c=""
    for i in range(0,len(b)):
        c=c+b[i]
    print(c)
    self.bg=ImageTk.PhotoImage(file="images/image2.jpg")
self.bg_image=Label(self.root,image=self.bg).place(x=0,y=0,relwidth=0.999,re
lheight=0.999)
    Frame_login=Frame(self.root,bg="white")
    Frame_login.place(x=485,y=20,height=450,width=500)
    title=Label(Frame_login,text="Log in",font=("Goudy old

```

```

style",40,"bold"),fg="black",bg="white").place(x=180,y=30)
    title=Label(Frame_login,text="Phone no.",font=("times new
roman",15),fg="black",bg="white").place(x=60,y=110)
    self.txt_user=Entry(Frame_login,font=("times new
roman",15),bg="lightgray")
    self.txt_user.place(x=60,y=135,width=350,height=35)
    title=Label(Frame_login,text="Password",font=("times new
roman",15),fg="black",bg="white").place(x=60,y=170)
    self.txt_pass=Entry(Frame_login,show='*',font=("times new
roman",15),bg="lightgray")
    self.txt_pass.place(x=60,y=200,width=350,height=35)
    title=Label(Frame_login,text="Captcha",font=("times new
roman",15),fg="black",bg="white").place(x=60,y=235)
    title=Label(Frame_login,text=c,font=("times new
roman",15),fg="black",bg="white",relief="solid").place(x=60,y=265)
    self.txt_captcha=Entry(Frame_login,font=("times new
roman",15),bg="lightgray")
    self.txt_captcha.place(x=60,y=300,width=350,height=35)
forget_btn=Button(Frame_login,command=self.forget_function,cursor="hand2",
text="Create new account?",bg="white",fg="blue",bd=0,font=("times new
roman",12)).place(x=60,y=370)
login_btn=Button(Frame_login,command=self.login_function,cursor="hand2",
text="Log in",bg="DarkOrchid1",fg="black",font=("times new
roman",20)).place(x=70,y=400)

def forget_function(self): #when clicking sign up button
    self.root=root
    self.root.title("Login System")
    self.root.geometry("1470x1000")
    #image
    self.bg=ImageTk.PhotoImage(file="images/image2.jpg")
self.bg_image=Label(self.root,image=self.bg).place(x=0,y=0,relwidth=0.999,relheight=0.999)
    Frame_login=Frame(self.root,bg="white")
    Frame_login.place(x=485,y=20,height=750,width=500)
    title=Label(Frame_login,text="Sign Up",font=("Goudy old
style",40,"bold"),fg="black",bg="white").place(x=180,y=30)
    title=Label(Frame_login,text="Name",font=("times new

```



```

roman",15),fg="black",bg="white").place(x=60,y=145)
    self.txt_name=Entry(Frame_login,font=("times new
roman",15),bg="lightgray")
    self.txt_name.place(x=60,y=170,width=350,height=35)
    title=Label(Frame_login,text="Email",font=("times new
roman",15),fg="black",bg="white").place(x=60,y=205)
    self.txt_email=Entry(Frame_login,font=("times new
roman",15),bg="lightgray")
    self.txt_email.place(x=60,y=240,width=350,height=35)
    title=Label(Frame_login,text="Mobile No",font=("times new
roman",15),fg="black",bg="white").place(x=60,y=275)
    self.txt_mobilenos=Entry(Frame_login,font=("times new
roman",15),bg="lightgray")
    self.txt_mobilenos.place(x=60,y=310,width=350,height=35)
    title=Label(Frame_login,text="Password",font=("times new
roman",15),fg="black",bg="white").place(x=60,y=345)
    self.txt_pass=Entry(Frame_login,show="*",font=("times new
roman",15),bg="lightgray")
    self.txt_pass.place(x=60,y=380,width=350,height=35)
    title=Label(Frame_login,text="Address",font=("times new
roman",15),fg="black",bg="white").place(x=60,y=425)
    self.txt_address=Entry(Frame_login,font=("times new
roman",15),bg="lightgray")
    self.txt_address.place(x=60,y=450,width=350,height=35)
    title=Label(Frame_login,text="City",font=("times new
roman",15),fg="black",bg="white").place(x=60,y=485)
    self.txt_city=Entry(Frame_login ,font=("times new
roman",15),bg="lightgray")
    self.txt_city.place(x=60,y=520,width=350,height=35)
forget_btn=Button(Frame_login,command=self.exist_function,cursor="hand2",
text="Have an existing account?",bg="white",fg="blue",bd=0,font=("times
new roman",12)).place(x=60,y=560)
login_btn=Button(Frame_login,command=self.signup_function,cursor="hand2",
text="Sign Up",bg="DarkOrchid1",fg="black",font=("times new
roman",20)).place(x=70,y=600)

```

```

def signup_function(self): #PAGE 2

```

```

    if(self.txt_name.get()==" or self.txt_email.get()==" or

```

```

self.txt_mobilenumber.get()==" or self.txt_password.get()==" or
self.txt_address.get()==" or self.txt_city.get()==""):
    messagebox.showerror("Error","All fields are
required",parent=self.root)
    a3=int(self.txt_mobilenumber.get())
    a4=self.txt_password.get()
    query1="select * from customer where mobile_no={ }".format(a3)
    mycur1=con.cursor()
    mycur1.execute(query1)
    row1=mycur1.fetchall()
    rc1=mycur1.rowcount
    query2="select * from customer where password='{ }'".format(a4)
    mycur2=con.cursor()
    mycur2.execute(query2)
    row2=mycur2.fetchall()
    rc2=mycur2.rowcount
    if(rc1!=0 or rc2!=0):
        messagebox.showerror("Error","Account already exist with this mobile
no/password",parent=self.root)
    else:
        a1=self.txt_name.get()
        a2=self.txt_email.get()
        a3=int(self.txt_mobilenumber.get())
        a4=self.txt_password.get()
        a5=self.txt_address.get()
        a6=self.txt_city.get()
        query="select max(custid) from customer"
        mycur=con.cursor()
        mycur.execute(query)
        row1=mycur.fetchall()
        h=row1[0][0]+1
        query="insert into customer
values( {}, '{}', '{}', {}, {}, {}, {}, {} )".format(h,a1,a2,a3,a4,a5,a6)
        print("Data successfully added\n")
        mycur=con.cursor()
        mycur.execute(query)
        con.commit()
        mycur.close()

```

```

def page_function(self): #NEW PAGE
    self.root=root
    self.root.title("Hello")
    self.root.geometry("1470x1000")
    #image
    self.bg=ImageTk.PhotoImage(file="images/background.jpg")
self.bg_image=Label(self.root,image=self.bg).place(x=0,y=0,relwidth=0.999,relheight=0.999)
    #img = ImageTk.PhotoImage(Image.open("D:/project/images/pizza.jpg"))
    Frame_login=Frame(self.root,bg="white")
    Frame_login.place(x=485,y=50,height=450,width=500)
    title=Label(Frame_login,text="Add Food Items",font=("Goudy old style",36,"bold"),fg="black",bg="white").place(x=90,y=30)
    label = ttk.Label(root,text="Food Item").place(x=620,y=175)
    # create a combobox
    n= StringVar()
    global food_cb,o1
    o1
    food_cb = ttk.Combobox(root,width=27,textvariable=n)
    food_cb.place(x=620,y=200)
    query1="select foodname from food_item"
    mycurl=con.cursor()
    mycurl.execute(query1)
    row1=mycurl.fetchall()
    print(row1)
    global l
    global w
    l=[]
    for i in row1:
        l+=[i][0]
    print(l)
    food_cb['values'] = l
    food_cb.current()
    label = ttk.Label(root,text="Number").place(x=620,y=260)
    w = Scale(root, from_=1, to=10, orient=HORIZONTAL)
    w.place(x=620,y=285)
    global k

```

```

k=0
add_btn=Button(Frame_login,command=self.addfood,cursor="hand2",text="
Add Item ",bg="DarkOrchid1",fg="black",font=("times new
roman",20)).place(x=170,y=310)
login_btn=Button(Frame_login,command=self.bill,cursor="hand2",text="Orde
r",bg="DarkOrchid1",fg="black",font=("times new
roman",20)).place(x=195,y=370)
print(w)

def addfood(self): #submit button for 2nd page
    global food_cb,w,l,k,m0,m1
    for i in range(0,len(l)):
        if food_cb.get()==l[i]:
            m0+=l[i]
            m1+=w.get()
            k+=1
    print(m0,m1)

def bill(self):
    global m0,m1
    self.root=root
    self.root.title("Hello")
    self.root.geometry("1470x1000")
    #image
    self.bg=ImageTk.PhotoImage(file="images/background.jpg")
self.bg_image=Label(self.root,image=self.bg).place(x=0,y=0,relwidth=0.999,relheight=0.999)
    #img = ImageTk.PhotoImage(Image.open("D:/project/images/pizza.jpg"))
    Frame_login=Frame(self.root,bg="white")
    Frame_login.place(x=485,y=50,height=450,width=500)
    title=Label(Frame_login,text="Order Details",font=("Goudy old
style",40,"bold"),fg="black",bg="white").place(x=90,y=30)
    # Using treeview widget
    treev = ttk.Treeview(Frame_login, selectmode ='browse')
    treev.place(x=105,y=100)
    treev["columns"] = ("1", "2", "3")
    # Defining heading
    treev['show'] = 'headings'

```

```

# Assigning the width and anchor to the
# respective columns
treev.column("1", width = 90, anchor ='c')
treev.column("2", width = 90, anchor ='se')
treev.column("3", width = 90, anchor ='se')
# Assigning the heading names to the
# respective columns
treev.heading("1", text ="Food Name")
treev.heading("2", text ="Quantity")
treev.heading("3", text ="Total Cost")
# Inserting the items and their features to the
# columns built
l1=0
for i in range(len(m0)):
    query2="select price from food_item where
foodname='{ }'".format(m0[i])
    mycur1=con.cursor()
    mycur1.execute(query2)
    row1=mycur1.fetchone()
    k=m1[i]*row1[0]
    treev.insert("",'end',text ="L1",values =(m0[i],m1[i],"Rs"+str(k)))
    l1+=k
    l2="Rs"+str(l1)
    title=Label(Frame_login,text="Grand Total:",font=("times new
roman",15),fg="black",bg="white").place(x=215,y=330)
    title=Label(Frame_login,text=l2,font=("times new
roman",15),fg="black",bg="white").place(x=325,y=330)
add_btn=Button(Frame_login,command=self.pay,cursor="hand2",text="Pay",b
g="DarkOrchid1",fg="black",font=("times new
roman",20)).place(x=190,y=370)

def pay(self):
    global m0,m1
    global d1,d2
    self.root=root
    self.root.title("Hello")
    self.root.geometry("1470x1000")
    #image

```

```

global x,o1,d1,d2
d1=0
d2=0
self.bg=ImageTk.PhotoImage(file="images/background.jpg")
self.bg_image=Label(self.root,image=self.bg).place(x=0,y=0,relwidth=0.999,relheight=0.999)
    Frame_login=Frame(self.root,bg="white")
    Frame_login.place(x=485,y=50,height=450,width=500)
    title=Label(Frame_login,text="PayTM Mobile number",font=("times new roman",15),fg="black",bg="white").place(x=60,y=145)
    self.abc=Entry(Frame_login,font=("times new roman",15),bg="lightgray")
    self.abc.place(x=60,y=170,width=350,height=35)
    title=Label(Frame_login,text="Otp",font=("times new roman",15),fg="black",bg="white").place(x=60,y=205)
    self.bcd=Entry(Frame_login,font=("times new roman",15),bg="lightgray")
    self.bcd.place(x=60,y=240,width=350,height=35)
    title=Label(Frame_login,text="Order Details",font=("Goudy old style",40,"bold"),fg="black",bg="white").place(x=90,y=30)
    add_btn=Button(Frame_login,command=self.OTP,cursor="hand2",text="Send OTP",bg="DarkOrchid1",fg="black",font=("times new roman",20)).place(x=170,y=310)
    sub_btn=Button(Frame_login,command=self.DONE,cursor="hand2",text="Submit",bg="DarkOrchid1",fg="black",font=("times new roman",20)).place(x=185,y=390)

def OTP(self):
    global o1,o2
    query="select max(phoneid) from phoneno"
    mycur=con.cursor()
    mycur.execute(query)
    row1=mycur.fetchall()
    h2=row1[0][0]+2
    query="insert into phoneno values({},{})".format(h2,self.abc.get())
    mycur=con.cursor()
    mycur.execute(query)
    con.commit()

```

```

mycur.close()
os.system('whatsapp.py')
query1="select max(otpid) from otpable"
mycur=con.cursor()
mycur.execute(query1)
row1=mycur.fetchall()
query2="select otp from otpable where otpid={ }".format(row1[0][0])
mycur=con.cursor()
mycur.execute(query2)
row2=mycur.fetchall()
o1=row2[0][0]
print(o1)

```

```

def DONE(self):
    global o1
    print(self.abc.get(),self.bcd.get())
    if(self.abc.get()!=" and self.bcd.get()==o1):
        self.root=root
        self.root.title("Hello")
        self.root.geometry("1470x1000")
        #image
        self.bg=ImageTk.PhotoImage(file="images/background.jpg")
self.bg_image=Label(self.root,image=self.bg).place(x=0,y=0,relwidth=0.999,relheight=0.999)
        #img =
ImageTk.PhotoImage(Image.open("D:/project/images/pizza.jpg"))
        Frame_login=Frame(self.root,bg="white")
        Frame_login.place(x=485,y=50,height=450,width=500)
        title=Label(Frame_login,text="Payment Successful",font=("Times New Roman",26,"bold"),fg="black",bg="white").place(x=90,y=50)
        menubar = Menu(root)
        # Adding File Menu and commands
        file = Menu(menubar, tearoff = 0)
        menubar.add_cascade(label = 'File', menu = file)
        file.add_command(label = 'Feedback', command=self.feedback)
        file.add_command(label = 'Log Out', command=self.exist_function)
        root.config(menu=menubar)

```

```

def feedback(self):
    self.root=root
    self.root.title("Hello")
    self.root.geometry("1470x1000")
    self.bg=ImageTk.PhotoImage(file="images/background.jpg")
self.bg_image=Label(self.root,image=self.bg).place(x=0,y=0,relwidth=0.999,relheight=0.999)
    Frame_login=Frame(self.root,bg="white")
    Frame_login.place(x=485,y=50,height=450,width=500)
    title=Label(Frame_login,text="Feedback",font=("Goudy old style",40,"bold"),fg="black",bg="white").place(x=90,y=30)
    self.feed=Entry(Frame_login,font=("times new roman",15),bg="lightgray")
    self.feed.place(x=60,y=240,width=350,height=35)
    sub_btn=Button(Frame_login,command=self.fil,cursor="hand2",text="Submit",bg="DarkOrchid1",fg="black",font=("times new roman",20)).place(x=185,y=390)

def fil(self):
    f = open("demofile.txt","a")
    f.write(self.feed.get())
    f.write("\n")
    f.close()

l2=[]
o1=0
food_cb={ }
root=Tk()
obj=Login(root)
root.mainloop()

```


THANK YOU