**Index**

* *Abstract*
* *Design-front and back end*
* *Source code*
* *Conclusion*
* *Bibliography*

**ABSTRACT**

**Kiosk for Ordering Food** is a project that provides a system that processes the management of billing and data handling in a restaurant for full-fledged and efficient working. It helps the consumer to directly order the food, view the menu, and register themselves as a member of the restaurant.

It also provides the manager of the restaurant who will have all the responsibilities to insert new items to the menu. This is safeguarded by an encryption which will prevent anyone from accessing the portal. The system is capable of creating proper bills for the customers, keeping a record of them in an appropriate manner and displaying menu. There are features such specific bill number and varied items and categories.

The back end of project is MySQL database where all information related to the categories, the items and the bill is saved that can be put to use at any time. The items in menu are stored in the table Itemsmaster, bills generated are stored in the Table Bill, and the customer’s details in Table MemberDetails. The project uses database connectivity where details of the restaurant are stored. The front end which is the user interface is the GUI form designed in Python with the help of Tkinter module.

It provides a complete picture of a self-sufficient working restaurant kiosk that can be incorporated by many established and upcoming restaurants and reducing the long waiting lines.

**Front End**

It is the user interface where the data is collected. It has GUI elements allowing applications to collect data.

In this project, tkinter has been used for GUI feature. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter outputs the fastest and easiest way to create the GUI applications.

**Back End**

It is the database where the data is actually stored. Data is mainly stored in DBMS (MySQL). Database name: food\_chain

This was created and used by the use of the following MySQL query –

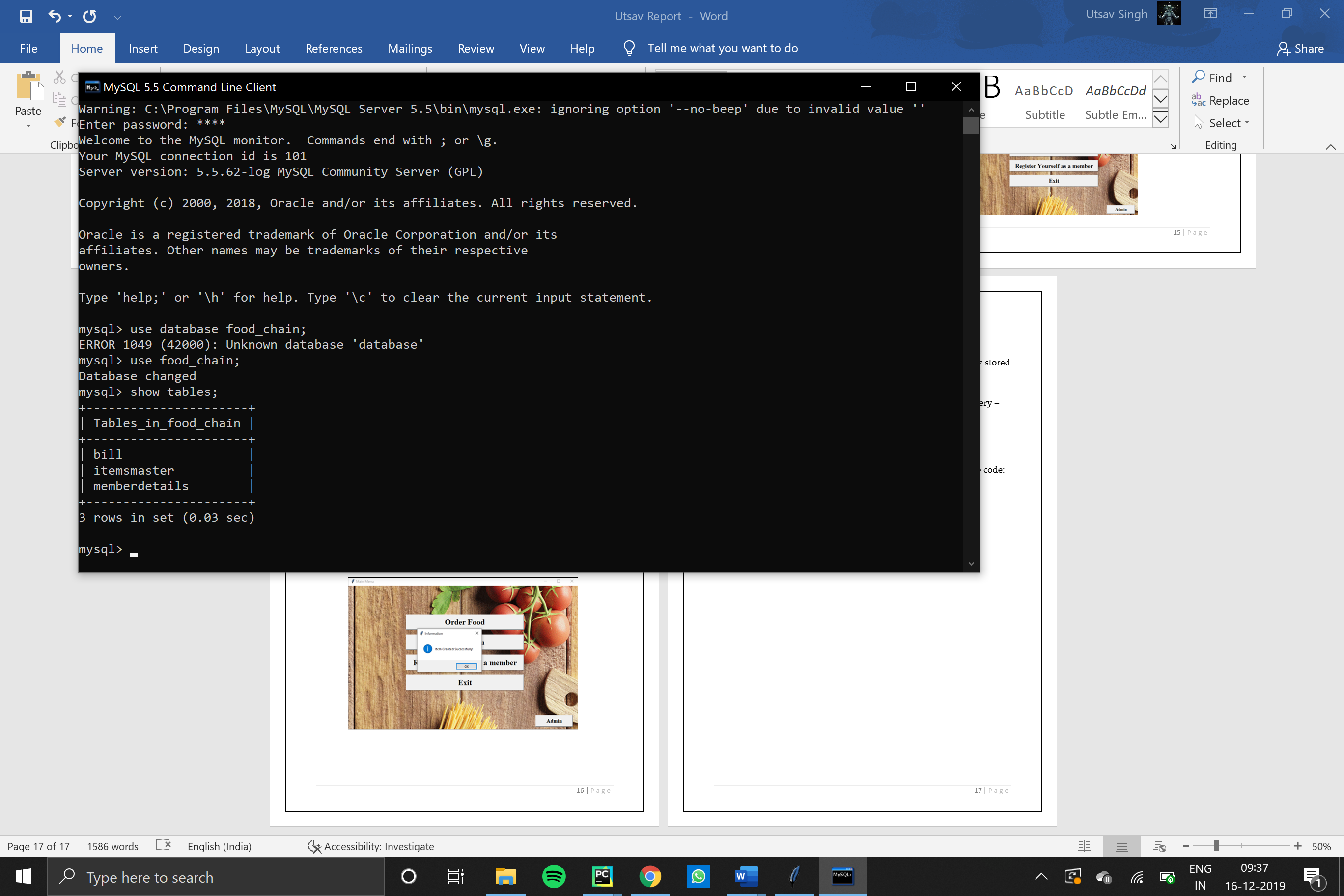
create database food\_chain;

use food\_chain;

The database contains three tables which can be seen by using the code:

use food\_chain;

show tables;



The three tables are:

1. bill table
2. itemsmaster table
3. memberdetails table
4. Table bill:

This table contains the information of each bill which has been generated by the system. The code for this table is:

create table bill

(

billno int(3) primary key,

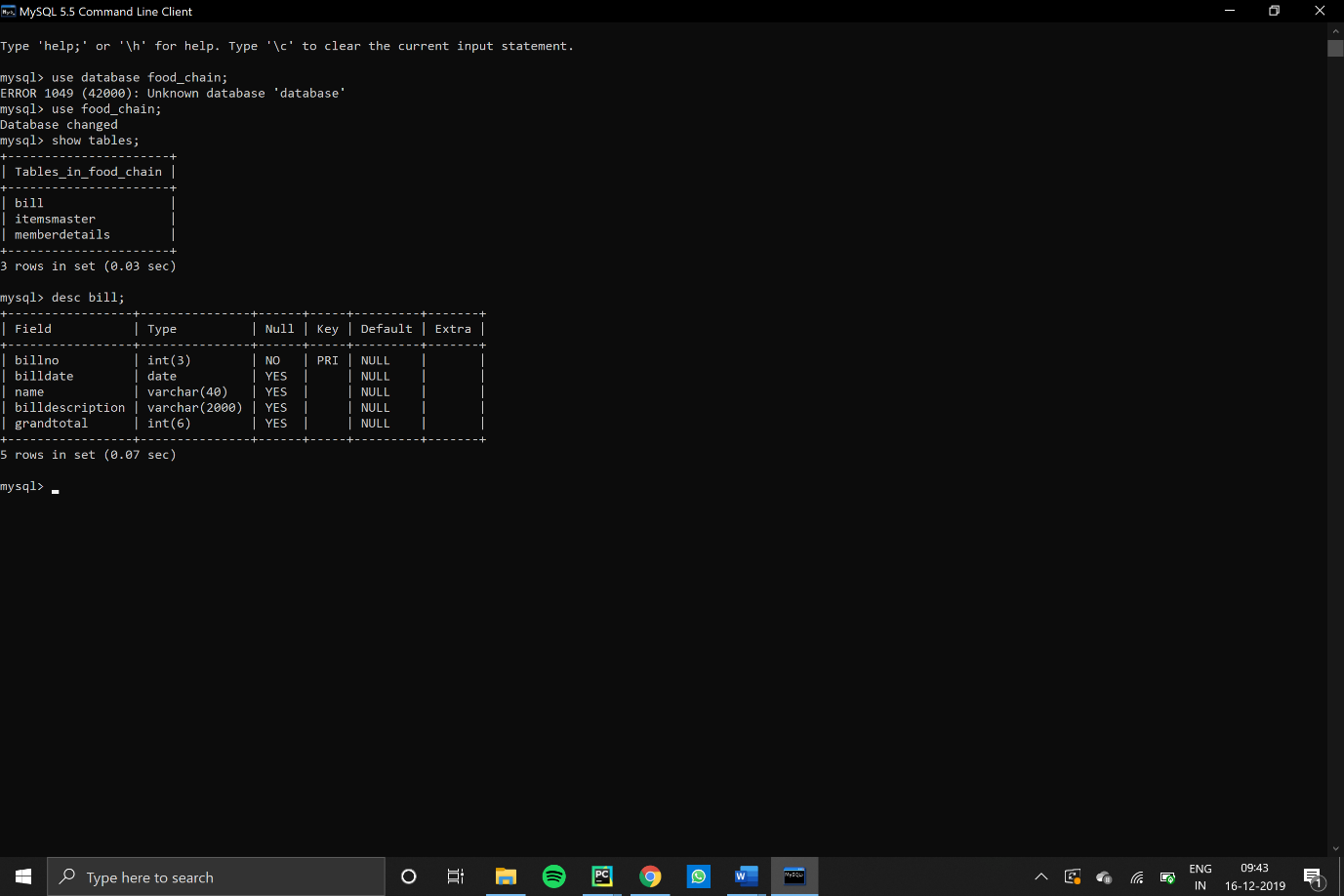
billdate date,

name varchar(40),

billdescription varchar(2000),

grandtotal int(6)

);



1. Table Itemsmaster”

This table contains all the items input by the manager. The code to create the table is:

create table itemsmaster

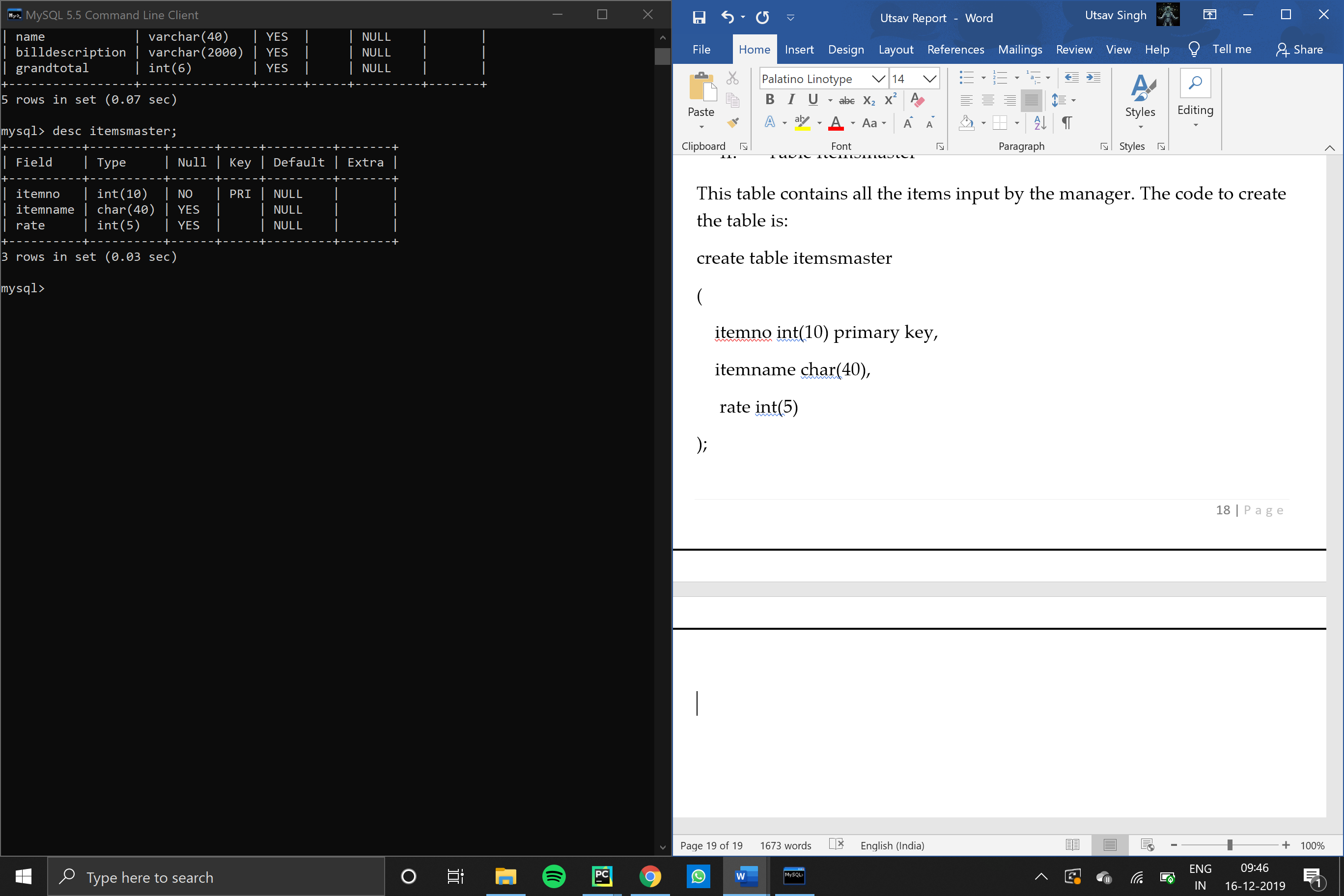
(

itemno int(10) primary key,

itemname char(40),

rate int(5)

);



1. Table memberdetails

This table contains the information entered by the user in the Register window. The code for this table is:

(

firstname varchar(40),

lastname varchar(40),

mobile varchar(40),

address varchar(300)

);

**Source Code**

**Python**

**Main Menu Screen:**

**from** tkinter **import** \*  
**from** tkinter **import** Button  
  
*#\*\*\*\*\*\*importing required modules\*\*\*\*\*\*\*\**

**import** Register  
**import** Manager  
**import** order  
**import** ShowMenu  
  
MainWindow = Tk()

MainWindow.title('Main Menu')

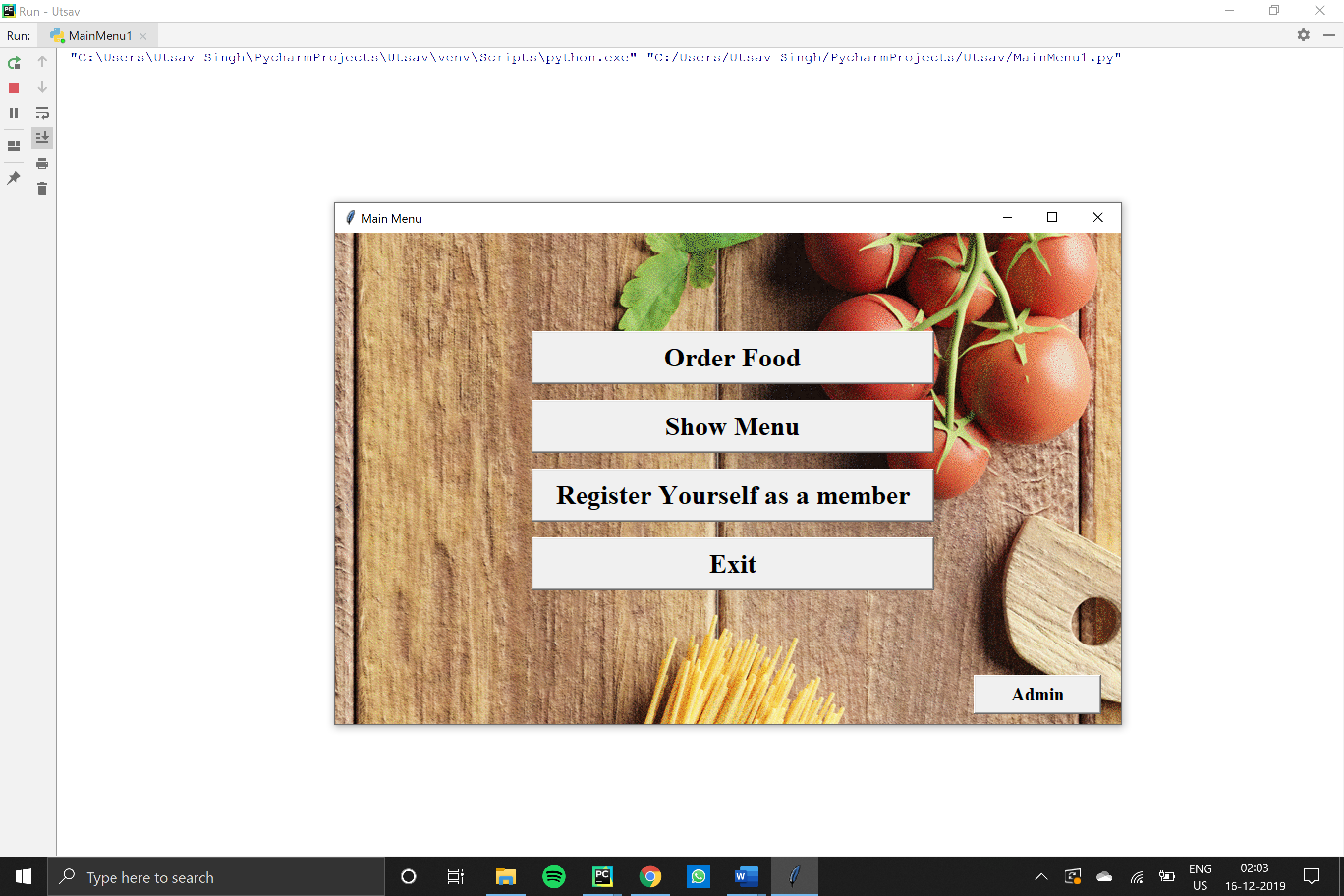
*#\*\*\*\*\*\*\*\*\*\*adding background image\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

picture = PhotoImage(file = **"C:\\Users\\Utsav Singh\\Desktop\\IP Project\\Pic.gif"**)  
heading = Label(MainWindow, text = **'Welcome to XYZ'**, font = **'times 26 bold underline'**, image = picture)  
heading.pack()  
  
*#\*\*\*\*\*\*\*\*\*\*\*\*\*\*creating buttons\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

btnOrderFood = Button(MainWindow, text = **'Order Food'** ,width = 25, font = **'times 20 bold'**, activebackground = **'red'**, activeforeground = **'white'**, command = order.showbillform)  
btnShowMenu = Button(MainWindow, text = **'Show Menu'**,width = 25, font = **'times 20 bold'**,activebackground = **'red'**, activeforeground = **'white'**, command = ShowMenu.ShowMenu)  
btnRegisterMember = Button(MainWindow, text = **'Register Yourself as a member'**,width = 25, font = **'times 20 bold'**, activebackground = **'red'**, activeforeground =**'white'**,command = Register.registerform,)  
btnExit = Button(MainWindow, text = **'Exit'**,width = 25, font = **'times 20 bold'**, activebackground = **'red'**, activeforeground = **'white'**, command = MainWindow.destroy)  
btnManager = Button(MainWindow, text = **'Admin'**, width = 10, font = **'times 15 bold'**, activebackground = **'red'**, activeforeground =**'white'**, command = Manager.check\_password)

*#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*placing buttons on screen\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**btnOrderFood.place(x = 200, y = 100)  
btnShowMenu.place(x = 200, y = 170)  
btnRegisterMember.place (x = 200, y = 240)  
btnExit.place(x = 200, y= 310)  
btnManager.place (x = 650,y = 450)

*#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*MainWindow Geometry\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**w = 800  
h = 500  
screenwidth = MainWindow.winfo\_screenmmwidth()  
screenheight = MainWindow.winfo\_screenheight()  
x = str(int(screenwidth/2 - w / 2))  
y = str(int(screenheight/2 - h/2))  
s = **'800x500'** + x + **'+'** + y  
MainWindow.geometry(s)  
MainWindow.resizable(width = **False**, height = **True**)  
  
MainWindow.mainloop()

***Output:***

**Order Food Screen:**

*#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*importing required modules\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****from** tkinter **import** \*  
**from** tkinter **import** ttk  
**import** tkinter.messagebox  
**import** mysql.connector **as** connector  
**import** pandas  
  
*#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*user-defined functions\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****def** showbillform():  
 **def** loaditemsinitemscombo():  
 mydb = connector.connect(host=**"localhost"**, user=**"root"**, passwd=**"root"**, database =

**"food\_chain"** )  
 mycursor = mydb.cursor()  
 mycursor.execute(**'Select itemname from itemsmaster'**)  
 result = mycursor.fetchall()  
 DF = pandas.DataFrame(result, columns=[**'itemname'**])  
 **return** list(DF[**'itemname'**])  
  
 **def** setselecteditem(event):  
 txtselecteditem.delete(0, END)  
 txtselecteditem.insert(0, itemscombo.get())  
  
 mydb = connector.connect(host=**"localhost"**, user=**"root"**, passwd=**"root"**, database =

**"food\_chain"** )  
 mycursor = mydb.cursor()  
 mycursor.execute(**"Select rate from itemsmaster where itemname = '{}'"**.format(

txtselecteditem.get() ))  
 result = mycursor.fetchall()  
 DF = pandas.DataFrame(result, columns=[**'rate'**])  
 txtrate.delete(0, END)  
 txtrate.insert(0, list(DF[**'rate'**]))  
  
 **def** calculateitemamount():  
 rate = int(txtrate.get())  
 qty = int(qtyspin.get())  
 amount = rate \* qty  
 txtamt.delete(0, END)  
 txtamt.insert(0, amount)  
  
 **def** additemdatatobill():  
 txtdata.config(state=**'normal'**)  
 item = txtselecteditem.get()  
 rate = txtrate.get()  
 qty = qtyspin.get()  
 amount = txtamt.get()  
 txtdata.insert(INSERT, item + **'\t\t'** + rate + **'\t\t'** + qty + **'\t\t'** + amount + **'\n\n'**)  
 txtdata.config(state=**'disabled'**)  
 grandtotal = int(txtgrandtotal.get()) + int(txtamt.get())  
 txtgrandtotal.delete(0, END)  
 txtgrandtotal.insert(0, grandtotal)  
  
 **def** savebill():  
 billdate = txtbilldate.get()  
 name = txtname.get()  
 billdescription = txtdata.get(1.0, END)  
 grandtotal = txtgrandtotal.get()  
 mydb = connector.connect(host=**"localhost"**, user=**"root"**, passwd=**"root"**,

database = **"food\_chain"**)  
 mycursor = mydb.cursor()  
 q = **"insert into bill values({},'{}','{}','{}',{})"** q = q.format(c, billdate, name, billdescription, grandtotal)  
 mycursor.execute(q)  
 mydb.commit()  
 tkinter.messagebox.showinfo(**'Information'**, **'Order Received!'**)  
 mainwindow.destroy()  
  
 *# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Window Layout \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** mainwindow = Tk()  
 mainwindow.title(**'Order Food'**)  
 mainwindow.geometry(**'800x640'**)  
  
 headinglabel = Label(mainwindow, text=**"Generate Bill"**, font=**'times 24 bold underline'**)  
 selectlabel = Label(mainwindow, text=**"Select Item"**, font=**'times 12 bold'**)  
 txtdata = Text(mainwindow, width=70, height=12)  
 txtdata.insert(INSERT, **'Item Name'** + **'\t\t'** + **'Rate'** + **'\t\t'** + **'Qty'** + **'\t\t'** + **'Amount'** +

**'\n\n'**)  
 txtselecteditem = Entry(mainwindow, width=25)  
 ratelabel = Label(mainwindow, text=**"Rate"**, font=**'times 12 bold'**)  
 qtylabel = Label(mainwindow, text = **'Quantity'**, font = **'times 12 bold'**)  
 qtyspin = Spinbox(mainwindow, from\_=0,to = 50)  
 amountlabel = Label(mainwindow, text=**"Amount"**, font=**'times 12 bold'**)  
 txtrate = Entry(mainwindow)  
 txtamt = Entry(mainwindow)  
 btncalculate = Button(mainwindow, text=**'Calculate'**, command=calculateitemamount)  
 btnaddtobill = Button(mainwindow, text=**'Add Selected Item To Bill'**, width=40,

font=**'times 14 bold'**, command=additemdatatobill)  
 btnsavebill = Button(mainwindow, text=**'Order Food'**, font=**'times 14 bold'**, width=20,

command=savebill)  
 grandtotallabel = Label(mainwindow, text=**"Grand Total"**, font=**'times 12 bold'**)  
 txtgrandtotal = Entry(mainwindow)  
  
 mydb = connector.connect(host=**"localhost"**, user=**"root"**, passwd=**"root"**,

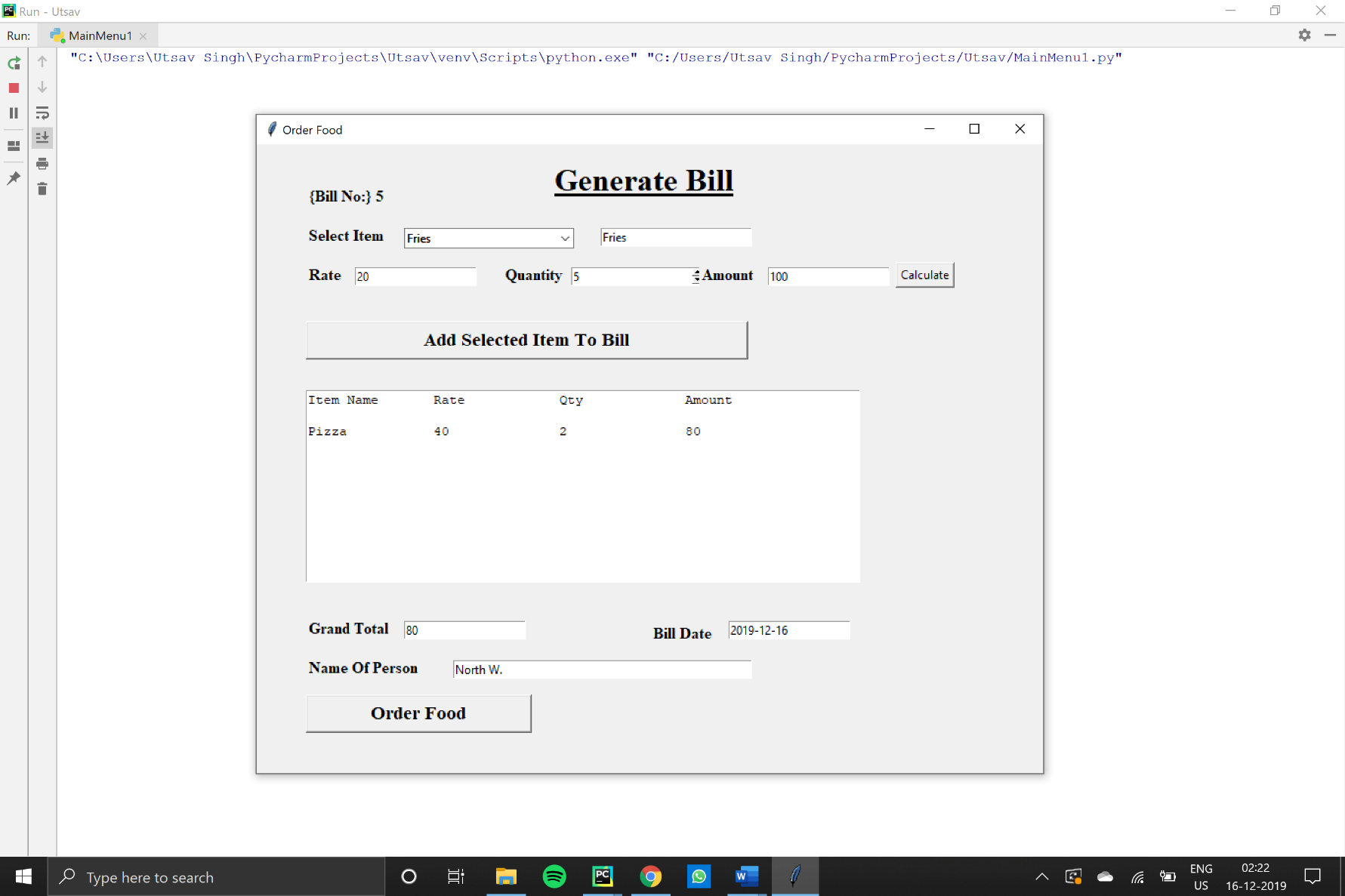
database= **"food\_chain"**)  
 mycursor = mydb.cursor()  
 a = **"select \* from bill"** mycursor.execute(a)  
 b = mycursor.fetchall()  
 DF = pandas.DataFrame(b, columns=[**'billno'**, **'billdate'**, **'name'**, **'billdescription'**,

**'grandtotal'**])  
 c = max(DF[**'billno'**])  
 c = c + 1  
 billnolabel = Label(mainwindow, text=(**'Bill No:'**,c), font=**'times 12 bold'**)  
 itemscombo = ttk.Combobox(mainwindow, values=loaditemsinitemscombo(), width=25,

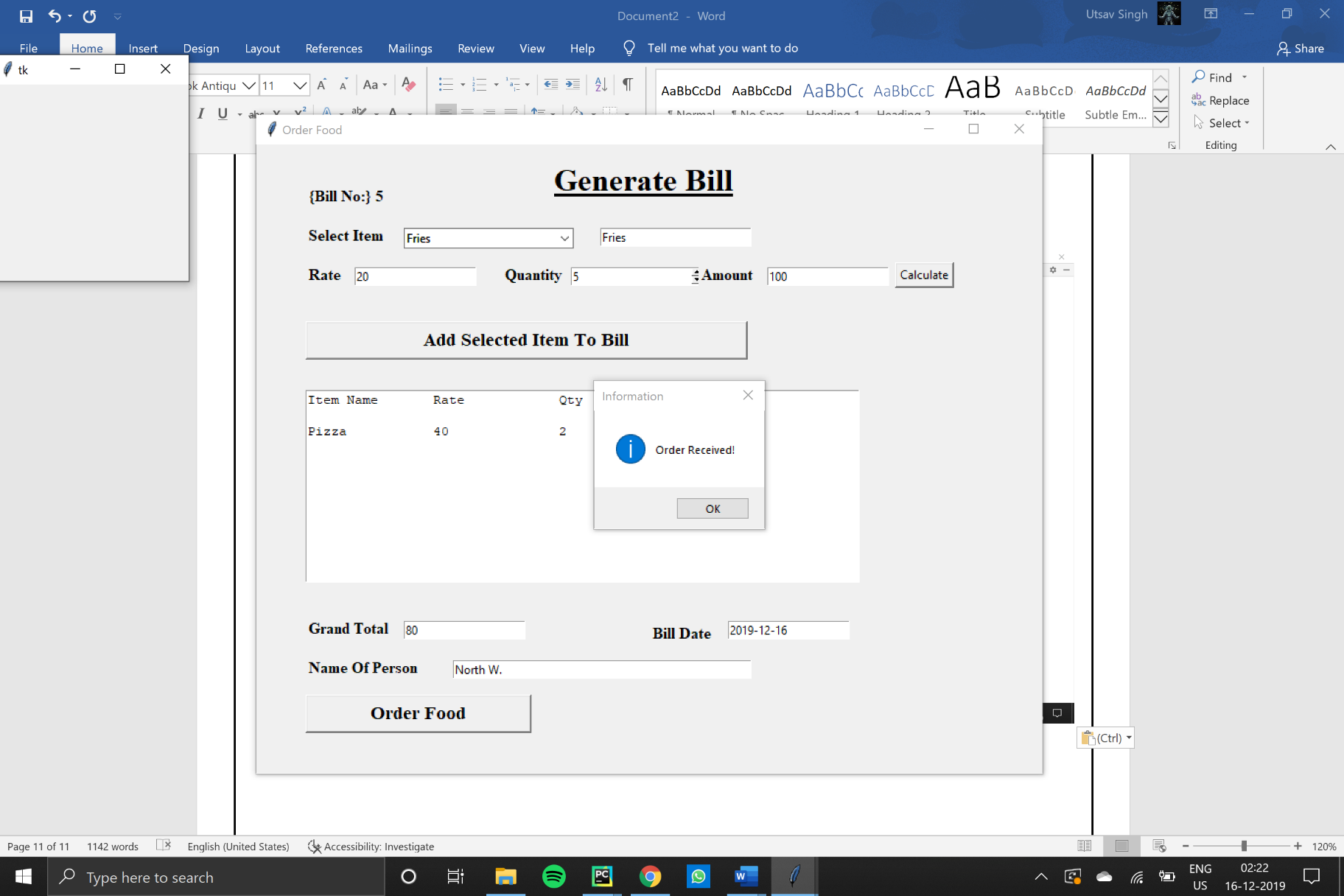
state=**'readonly'**)  
 itemscombo.current(0)  
 itemscombo.bind(**"<<ComboboxSelected>>"**, setselecteditem)  
 billdatelabel = Label(mainwindow, text=**"Bill Date"**, font=**'times 12 bold'**)  
 txtbilldate = Entry(mainwindow)  
 namelabel = Label(mainwindow, text=**"Name Of Person"**, font=**'times 12 bold'**)  
 txtname = Entry(mainwindow, width=50)  
  
 headinglabel.place(x=300, y=15)  
 selectlabel.place(x=50, y=80)  
 txtdata.place(x=50, y=250)  
 itemscombo.place(x=150, y=85)  
 txtselecteditem.place(x=350, y=85)  
 ratelabel.place(x=50, y=120)  
 txtrate.place(x=100, y=125)  
 qtylabel.place(x=250, y=120)  
 qtyspin.place(x=320, y=125)  
 amountlabel.place(x=450, y=120)  
 txtamt.place(x=520, y=125)  
 btncalculate.place(x=650, y=120)  
 btnaddtobill.place(x=50, y=180)  
 grandtotallabel.place(x=50, y=480)  
 txtgrandtotal.place(x=150, y=485)  
 txtgrandtotal.insert(0, 0)  
 billnolabel.place(x=50, y=40)  
 billdatelabel.place(x=400, y=485)  
 txtbilldate.place(x=480, y=485)  
 namelabel.place(x=50, y=520)  
 txtname.place(x=200, y=525)  
 btnsavebill.place(x=50, y=560)  
  
 *# \*\*\*\* Show Current Date in Bill Date Text Field \*\*\*\*\** mydb = connector.connect(host=**"localhost"**, user=**"root"**, passwd=**"root"**, database =

**"food\_chain"**)  
 mycursor = mydb.cursor()  
 mycursor.execute(**'Select curdate()'**)  
 result = mycursor.fetchall()  
 DF = pandas.DataFrame(result, columns=[**'Date'**])  
 txtbilldate.insert(0, list(DF[**'Date'**]))  
 mydb.close()  
 *# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** mainwindow.mainloop()

***Output:***



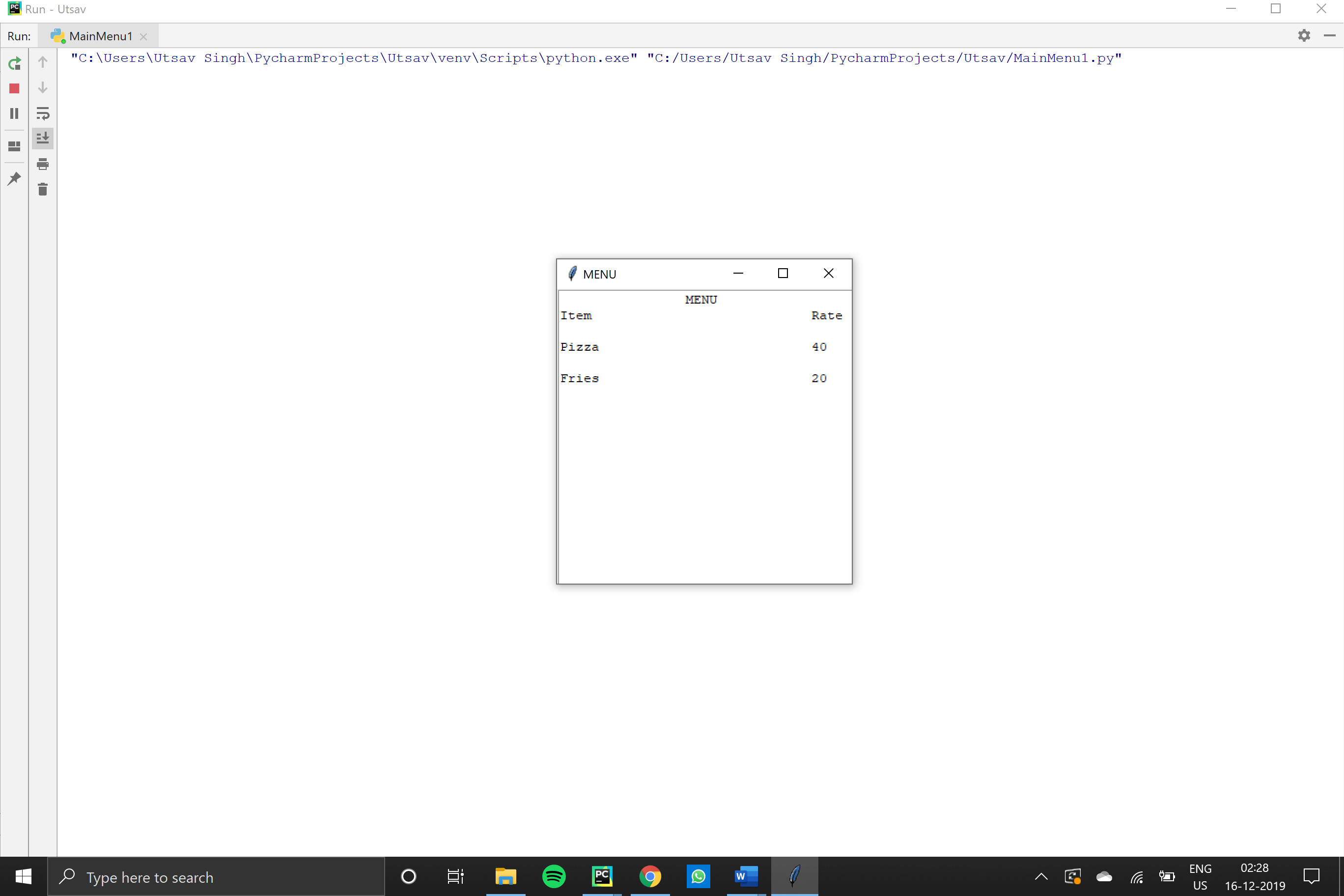
***After clicking “Order Food”:***



**Show Menu Screen:**

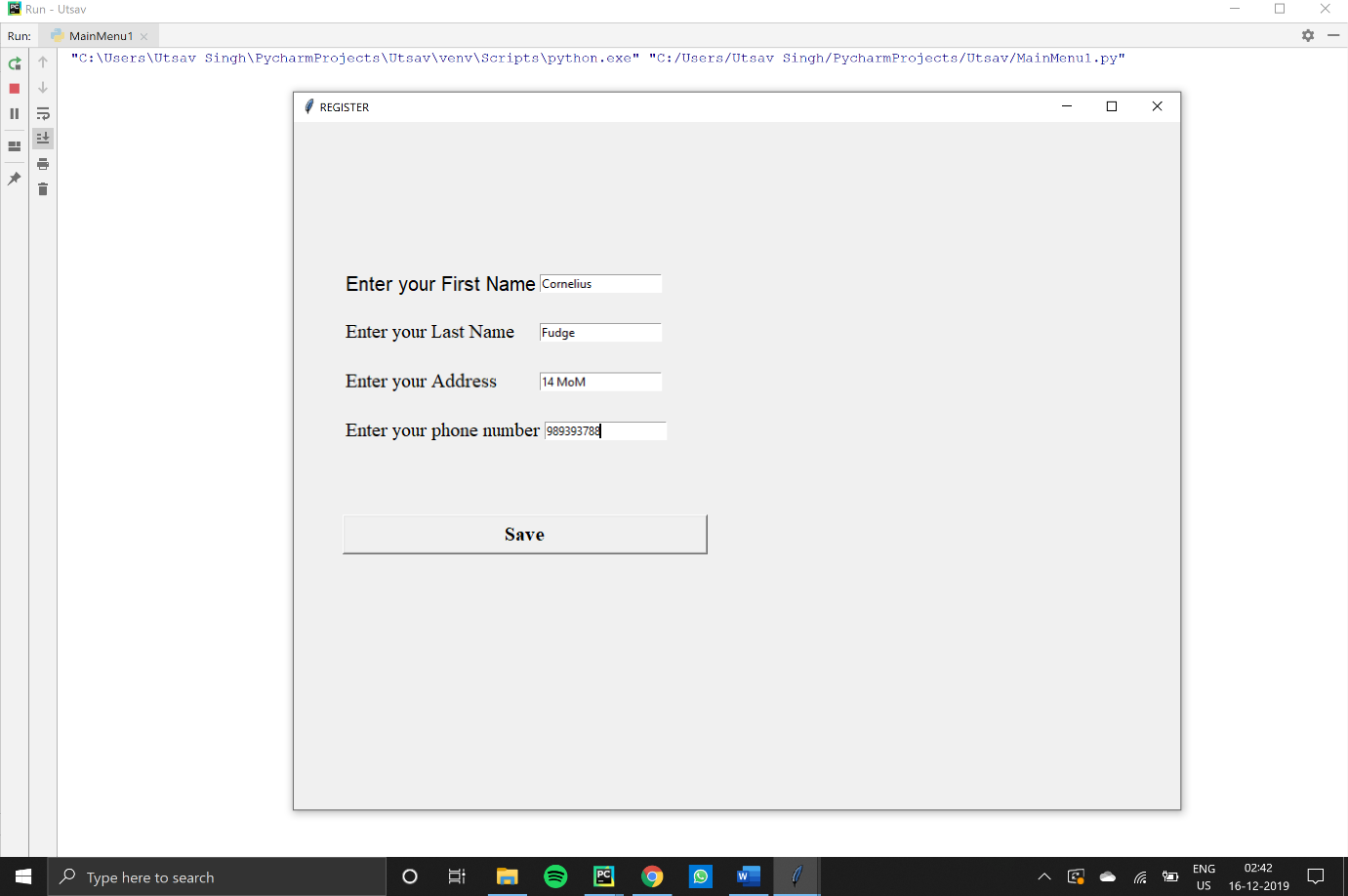
**from** tkinter **import** \*  
**import** mysql.connector **as** connector  
**import** pandas  
  
  
**def** ShowMenu():  
  
 window = Tk()  
 window.title(**'MENU'**)  
 mydb = connector.connect(host = **'localhost'**, user = **'root'**, password = **'root'**, database =

**'food\_chain'**)  
 mycursor = mydb.cursor()  
 mycursor.execute(**'select itemname, rate from itemsmaster'**)  
 itemname = mycursor.fetchall()  
 DF = pandas.DataFrame(itemname, columns=[**'Itemname'**, **'Rate'**])  
  
 text = Text(window)  
 text.place(x = 1, y = 1)  
 text.insert(INSERT,**'\t\tMENU\t\t\n'**)  
 text.insert(INSERT, **'Item\t\t\t\tRate\n\n'**)  
 **for** ri,rd **in** DF.iterrows():  
 a = DF.loc[ri,**'Itemname'**]  
 b = DF.loc[ri, **'Rate'**]  
 text.insert(INSERT, a)  
 text.insert(INSERT, **'\t\t\t\t'**)  
 text.insert(INSERT, b)  
 text.insert(INSERT, **'\n\n'**)  
  
 window.geometry(**'300x300'**)  
 window.mainloop()

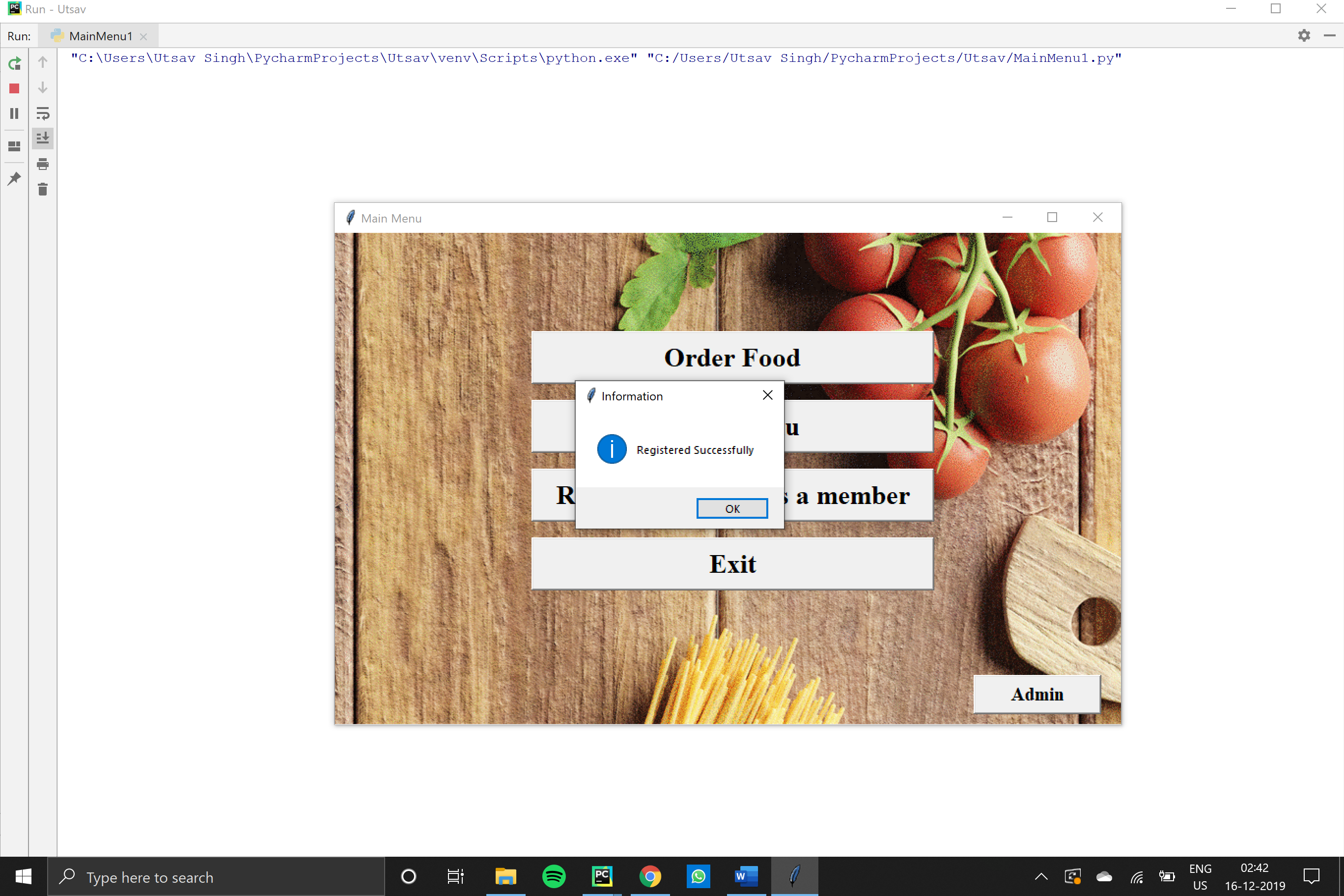


**Register Yourself Screen:**

**from** tkinter **import** \*  
**import** tkinter.messagebox  
**import** mysql.connector **as** connector  
  
**def** registerform():  
 **def** newmember():  
 mydb = connector.connect(host = **'localhost'**, user = **'root'**, password = **'root'**, database = **'Food\_Chain'**)  
 mycursor = mydb.cursor()  
  
 firstname = txtfirstname.get()  
 lastname = txtlastname.get()  
 address = txtaddress.get()  
 mobile = txtmobile.get()  
  
  
 qry = **"insert into memberdetails values('{}','{}','{}',{})"**.format(firstname, lastname,address, mobile)  
 mycursor.execute(qry)  
 mydb.commit()  
 mainwindow.destroy()  
 tkinter.messagebox.showinfo(**'Information'**, **'Registered Successfully'**)  
  
 mainwindow = Tk()  
 mainwindow.geometry(**'900x700'**)  
 mainwindow.title(**"REGISTER"**)  
  
 lblfirstname = Label(mainwindow, text=**'Enter your First Name'**, font=**'time 15'**)  
 lbllastname = Label(mainwindow, text=**'Enter your Last Name'**, font=**'times 15'**)  
 lbladdress = Label(mainwindow, text=**'Enter your Address'**, font=**'times 15'**)  
 lblmobile = Label(mainwindow, text=**'Enter your phone number'**, font=**'times 15'**)  
  
 txtfirstname =Entry(mainwindow)  
 txtlastname = Entry(mainwindow)  
 txtaddress = Entry(mainwindow)  
 txtmobile =Entry(mainwindow)  
  
 lblfirstname.place(x=50, y=150)  
 lbllastname.place(x=50, y=200)  
 lbladdress.place(x = 50, y = 250)  
 lblmobile.place(x=50, y=300)  
  
 txtfirstname.place(x=250, y=155)  
 txtlastname.place(x=250, y=205)  
 txtaddress.place(x = 250, y = 255)  
 txtmobile.place(x=255, y=305)  
  
 savebutton = Button(mainwindow, text=**'Save'**, font=**'times 15 bold'**, width=30, command = newmember)  
 savebutton.place(x=50, y=400)  
  
 txtfirstname.focus()  
 mainwindow.mainloop()

***Output:***

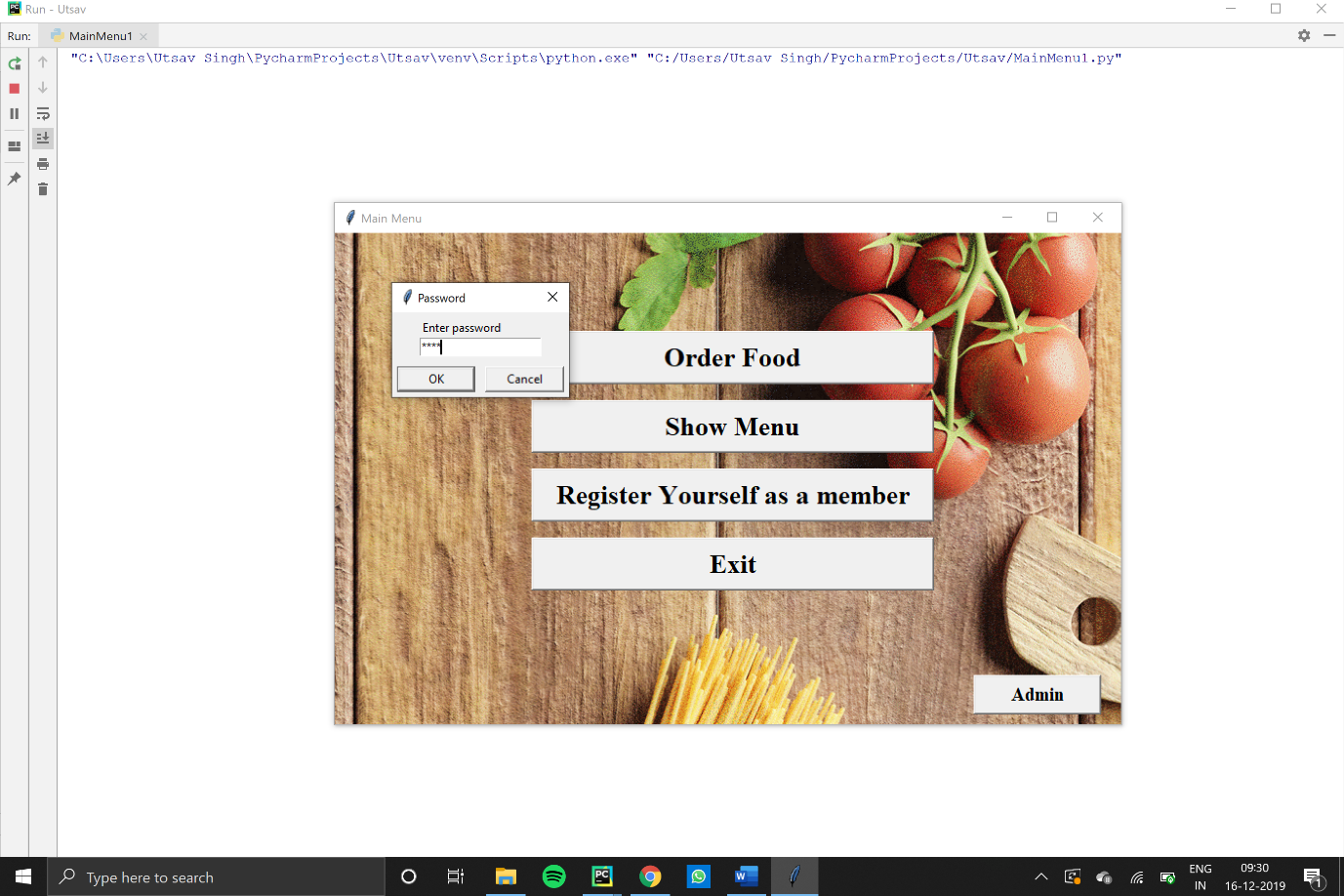
***After clicking “Save”:***



**Manager:**

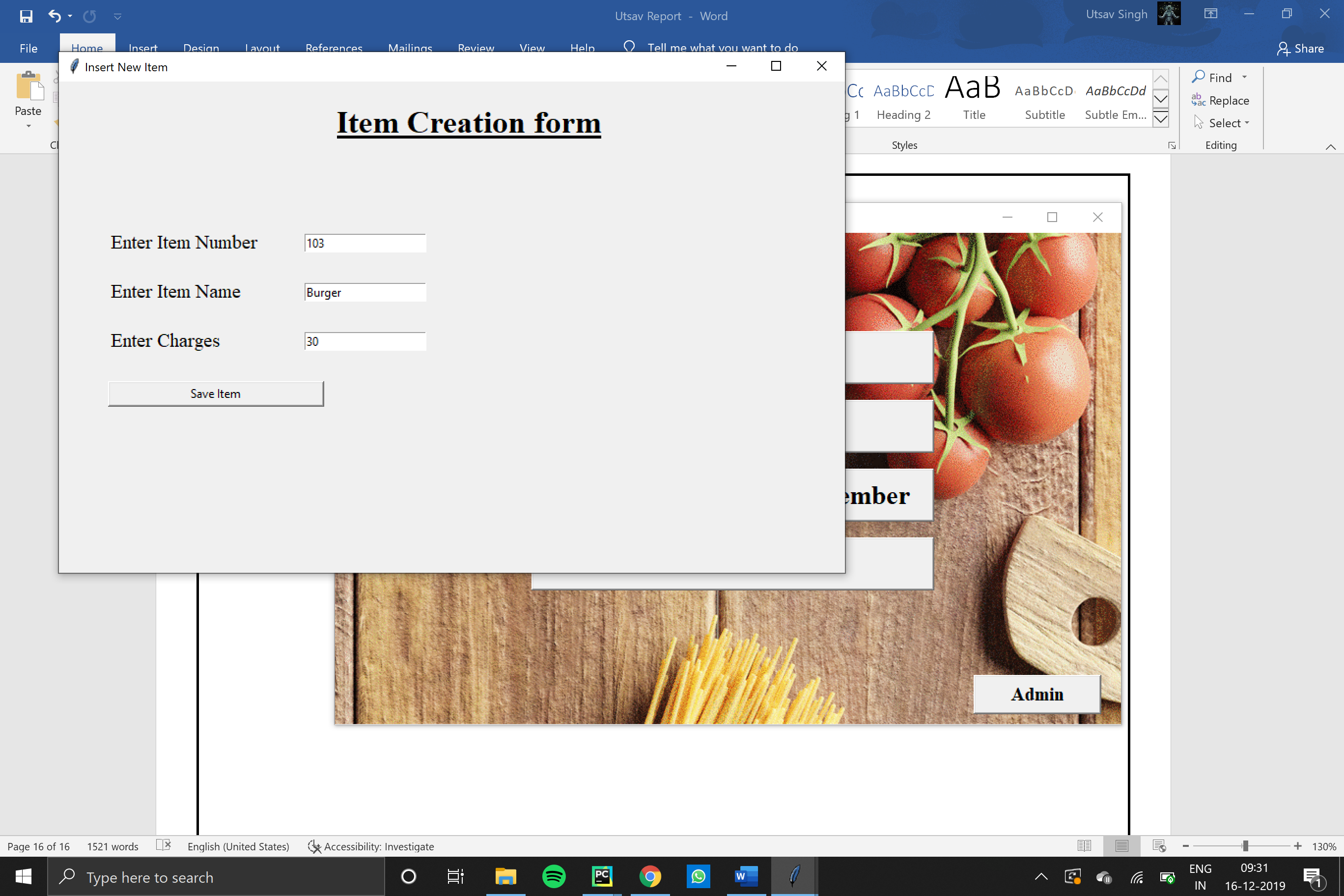
**from** tkinter **import** \*  
**import** tkinter.messagebox  
**import** mysql.connector **as** connector  
**from** tkinter.simpledialog **import** askstring  
  
**def** check\_password():  
 pswd = askstring(**'Password'**, **'Enter password'**, show=**'\*'**)  
 **if** pswd == **'1234'**:  
 **def** insertnewitem():  
 mydb = connector.connect(host=**'localhost'**, user=**'root'**, password=**'root'**, database=**'food\_chain'**)  
  
 mycursor = mydb.cursor()  
  
 itemno = txtitemno.get()  
 itemname = txtitemname.get()  
 itemrate = txtitemrate.get()  
 qry = **"insert into itemsmaster values({},'{}',{})"**.format(itemno, itemname, itemrate)  
 mycursor.execute(qry)  
 mydb.commit()  
 mainwindow.destroy()  
 tkinter.messagebox.showinfo(**'Information'**, **'Item Created Successfully!'**)

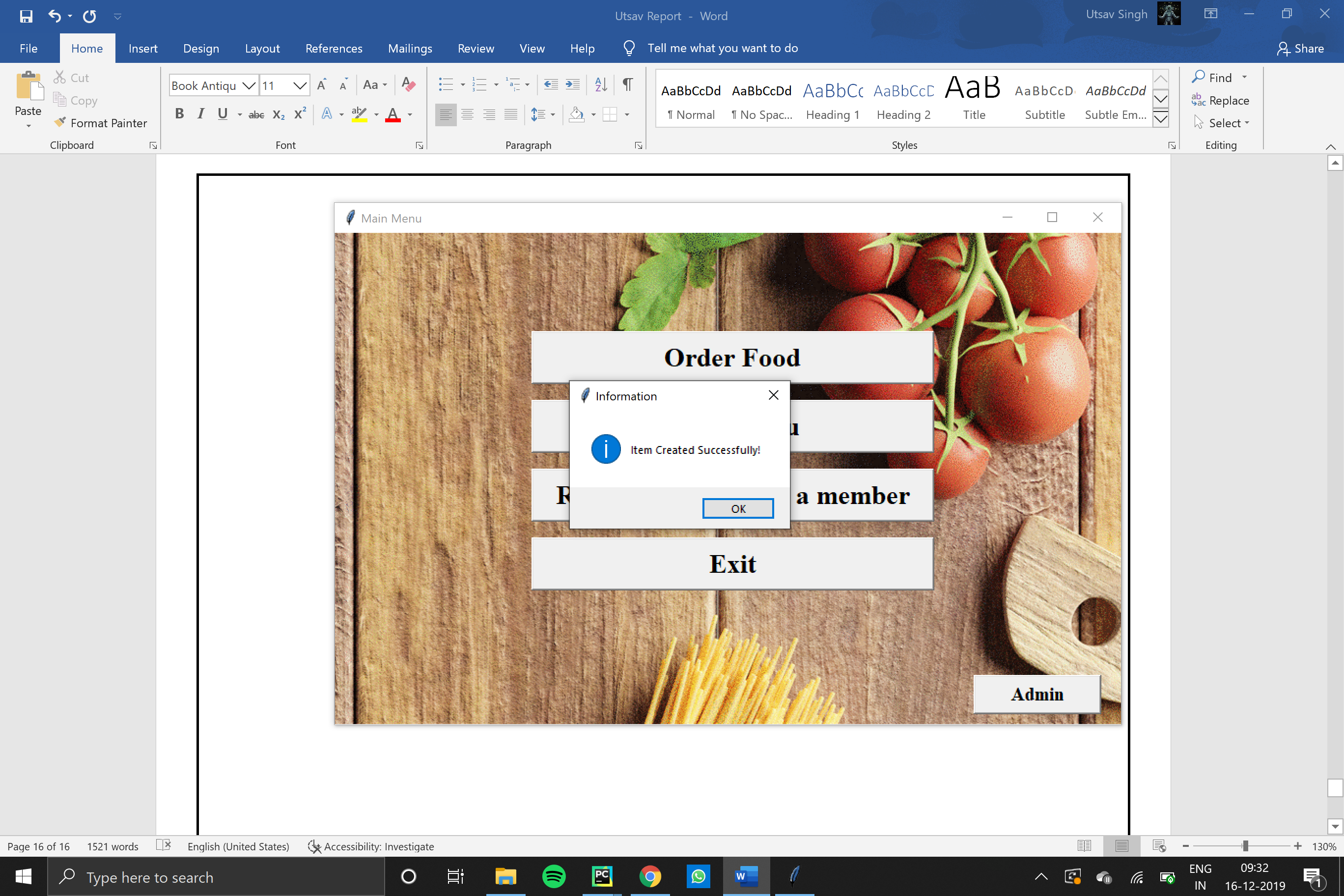
mainwindow = Tk()  
 mainwindow.geometry(**'800x500'**)  
  
 headinglabel = Label(mainwindow, text=**"Item Creation form"**, font=**'times 24 bold underline'**)  
 headinglabel.place(x=280, y=20)  
  
 lblitemno = Label(mainwindow, text=**'Enter Item Number'**, font=**'times 15'**)  
 lblitemname = Label(mainwindow, text=**'Enter Item Name'**, font=**'times 15'**)  
 lblitemrate = Label(mainwindow, text=**'Enter Charges'**, font=**'times 15'**)  
  
 txtitemno = Entry(mainwindow)  
 txtitemname = Entry(mainwindow)  
 txtitemrate = Entry(mainwindow)  
  
 lblitemno.place(x=50, y=150)  
 lblitemname.place(x=50, y=200)  
 lblitemrate.place(x=50, y=250)  
  
 txtitemno.place(x=250, y=155)  
 txtitemname.place(x=250, y=205)  
 txtitemrate.place(x=250, y=255)  
  
 savebutton = Button(mainwindow, text=**'Save Item'**, width=30, command=insertnewitem)  
 savebutton.place(x=50, y=305)  
 mainwindow.title(**'Insert New Item'**)  
  
 txtitemno.focus()  
 mainwindow.mainloop()  
  
 **else**:  
 tkinter.messagebox.showinfo(**"Access Denied"**,**"Wrong Password!"**)



***Output***

Item Creation Form:



After clicking Save Item:

**Conclusion**

**Present scope:**

The project “Kiosk for Ordering Food” will help in reducing human labour and efforts in the efficient management of a restaurant.

It will increase the working efficiency and help in data handling.

Its main features are:

* Taking digital orders
* Printing bills
* Maintaining records
* Reduce human effort
* User friendly

**Future scope:**

In the future there lies scope of a more productive and coherent venture by including more forms like Take Away, which would help in making parcel facilities easier in a restaurant. Concepts of HOME DELIVERY also would be incorporated.

As many restaurants have already adapted the idea of billing softwares, this project also aims to make the billing and data handling in restaurants more manageable, easier and user friendly.

**Bibliography**

The following sources were used as a reference for the successful completion of the project not to forget the immense help provided by the computer department.

The following sites were used-

* <http://stackoverflow.com>
* <http://www.tutorialspoint.com/sql.htm>
* <https://forums.netbeans.org>
* <https://en.wikipedia.org>
* <http://cbseipprojects.com>