



The University Of

Lucknow

MINOR PROJECT

“Restaurant Management System”

Partial Fulfillment of degree for the awards of the degree of

BACHELOR OF COMPUTER APPLICATION

By:-

ASHUTOSH PRASAD (180925080015)

AMAN RATHORE (180920580007)

(Department of B.C.A)

SCHOOL OF MANAGEMNET SCIENCES

(AFFILIATED TO UNIVERSITY OF LUCKNOW)

2017-2020

TABLE OF THE CONTENTS

- **DECLARATION**
- **CERTIFICATE**
- **ACKNOWLEDGEMENT**
- **TITLE**
- **INTRODUCTION OF PROJECT**
- **OBJECTIVE OF PROJECT**
- **TOOLS USED IN PROJECT**
- **IMPLEMENT ENVIRONMENT**
- **SYSTEM DESIGN**
- **MODULES**
- **SYSTEM SPECIFICATION**
- **PYTHON LIBRARY USED IN PROJECT**
- **TKINTER WIDGETS**
- **GEOMETRY MANAGEMENT**
- **CODE**

- **CONCLUSION**
- **PROJECT VIEW**

DECLARATION

We hereby declare that the project work entitled “Restaurant Management System” submitted to Dr.S.A. A RIZVI is a record of an original work done by us. This project work is submitted in the partial fulfilment of the requirement for the award of the degree of the “BACHELOR OF COMPUTER APPLICATION”. The result embodied in this thesis has not been submitted to any other university or institute for the any degree or diploma.

ASHUTOSH PRASAD

AMAN RATHORE

CERTIFICATE

This is to certified that ***Mr. Ashutosh Prasad*** and ***Aman Rathore*** has successfully completed ***Project Based Summer Internship*** in ***Python Programming Language*** from June 12th 2019 to August 2th 2019 conducted at ***Lucknow Learning Centre Of Centre Of MTA India*** and has been awarded the grade A+ on August 10th 2019.

During this period of Project Based Internships they were found innovative, punctual hardworking and inquisitive.

Dr. S.A.A RIZVI
(coordinator)

Ashutosh Prasad
Aman Rathore

Microsoft
Technology Associate



MLISTP19C2715LK



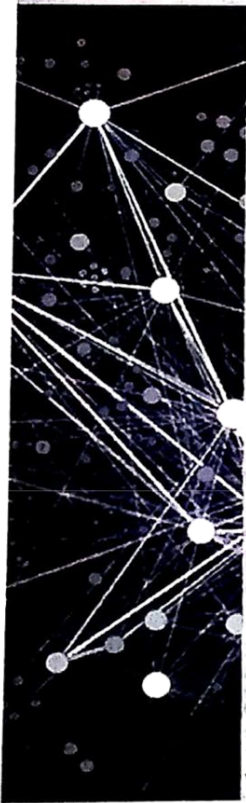
This is to certify that Mr. / Ms. **Aman Rathor** has successfully completed **Project Based Summer Internship in Python Programming** from **June 12th, 2019 to Aug 02nd 2019** conducted at **Lucknow Learning Centre of MTA India** and has been awarded the grade **A+** on Aug 10th, 2019.

During the period of Project Based Internship, he/she was found innovative, punctual, hardworking and inquisitive.

We wish a very bright future to him/her.

Held by
MT Academy India Pvt. Ltd.
Partner- Microsoft Technology Associate
www.mtaindia.co





This is to certify that Mr. / Ms. **Ashutosh Prasad** has successfully completed **Project Based Summer Internship in Python Programming** from **June 12th, 2019** to **Aug 02nd 2019** conducted at **Lucknow Learning Centre of MTA India** and has been awarded the grade **A+** on Aug 10th, 2019.

During the period of Project Based Internship, he/she was found innovative, punctual, hardworking and inquisitive.

We wish a very bright future to him/her.

Held by
MT Academy India Pvt. Ltd.
Partner- Microsoft Technology Associate
www.mtaindia.co



ACKNOWLEDGEMENT

I would like to express my gratitude towards my parents & my institution for their kind cooperation and encouragement which helps me in completion of this project. I would like to express my gratitude **Dr. S.A.A RIZVI** and special thanks to industry persons for giving me such attention and time.

We wish to express our deep sense of gratitude to our Guide Of BCA, School of Management Sciences, Lucknow for his able guidance and Useful suggestions, which helped us in completing the project work in Time. Needless to mention that Dr. S.A.A Rizvi, Head of Department, BCA, who had been a source of inspiration and for his timely guidance in the Conduct of our project work. Finally, yet importantly, we would like to express Our heartfelt thanks to our parents for their blessings, our team members for Their help and wishes for the successful completion of this project

R.M.S

(Restaurant Management System)

INTRODUCTION

INTRO TO PROJECT...

Today all the work at time of “Restaurant Management System” is done manually, which is very slow and consuming, much effort and time. It is required to design of computerized canteen Billing system to speed up and make it easy to use system. The Restaurant Management System an application developed in python is designed to manage billing data during purchased process so as to make the billing process easy for higher purchasing. The aim of processed system is to addresses the limitations of current system. The requirement for the system have been gathered from the defect recorded in the past and also based on the feedback from user of use reducing time in activities. Reduce the taken billing process in the canteen billing system, admitting a bill, verify it etc. Given billing reports smoothly to the customer in centralized way.

Paperless billing report in the canteen with reduces manpower. Reduce the manpower needed to perform all the billing and purchasing task by reducing the paper works needed.

Reduce the time.

Operational efficiency. Improve operational efficiency by improving the quality of process.

OBJECTIVE OF PROJECT

The main objective of project is to automate the process carried out put the organization with the improved performances and realize the vision of the paperless admission. Some of the goals of the system are as below: -

- Manages billing report customer.
- Create billing report the customer.
- Manages the bills of the item which is purchased by customer.
- Effective rather than handwork methods.

TOOLS USED IN PROJECT

Once you have installed python.exe software, there are various options for choosing an environment. Here are the two most common options.: -

- Python shell and IDLE 3.7.



- Vscode



- GIT Bash



IMPLEMENT ENVIRONMENT

For implementing the code, we have to install the python shell and idle. The .exe software of the python is easily downloaded from the python's official website

<https://www.python.org> .After downloading the software install the software, and while installing the software it is necessary to check the box of *Add To PATH*, this will automatically set the path in environment variable.

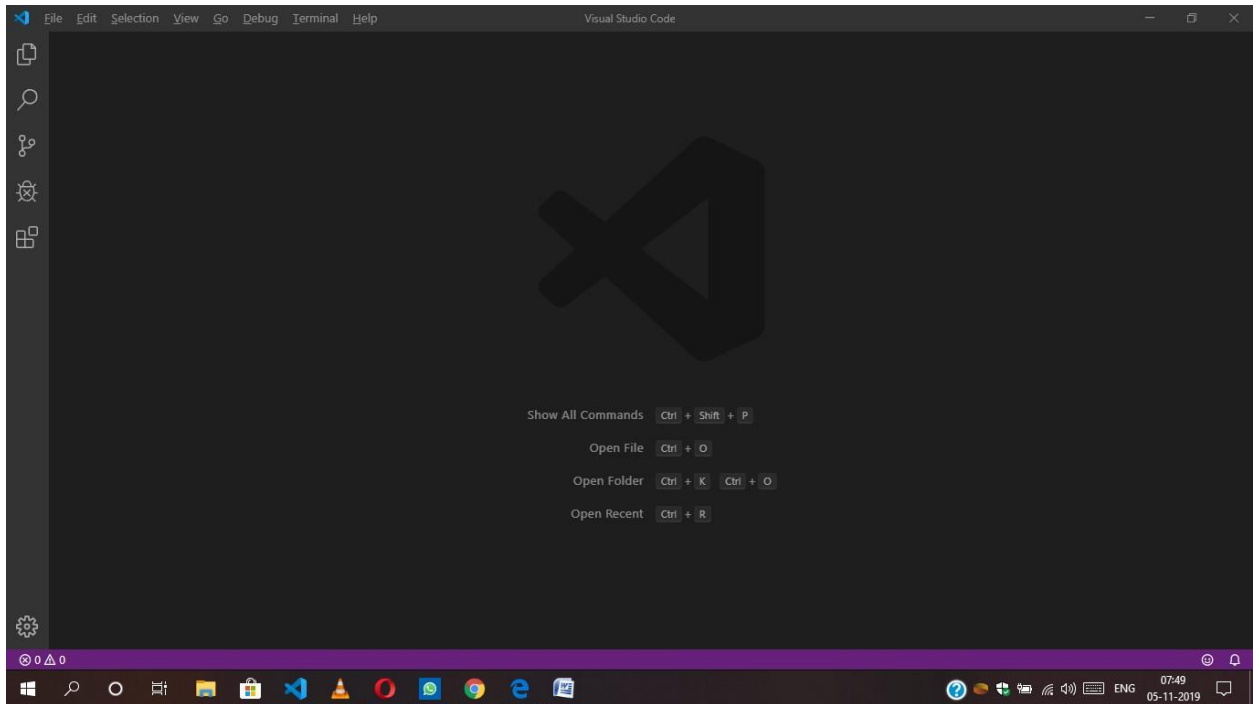
Once we install successfully the python shell&IDLE, we have so many choices to use any text editor for programming. This is the most efficient and easy way to code in VSCODE with GITBASH.

Vscode: -

A Microsoft Visual Studio Code is a platform of coding. It provides an efficient GUI and as well as the suggestions list box for fast programming. In this tool we add the extension of programming languages to interpret the code and run the code.

- Vscode is easy to use.
- It is easy as well as; in we interpret any programming language code at same TAB on adding the extensions.

- Error is easily fixed automatically and suggest tag list box is appearing on every tag or keyword.
- Error detection and compiling speed is fast.



Entrance UI.

Extension download section UI.

On window 8, 8.1, 7, vista, XP and windows 10(2016 build) command prompt does not have the ability to execute the code language itself, so in place of CMD we use Git Bash terminal.

We use Git bash terminal in windows 10 OS because there is very difficult in creating the specified folder and file every time on switching other folders ant etc. This is fix by Git bash, when we integrate the Git bash with the Vscode then there is terminal appears below the Vscode entrance tab. we use the commands to create, read/write, interpret, run, and delete the file.

Mkdir	-	for creating the folder
code	-	for creating the file with extension
del	-	deleting file.
Rm	-	deleting folder.
Python	-	for interpreting the python code and run in terminal.

We can download the Git bash from website <https://git-scm.com/downloads> .

After downloading and installing we have to integrate the git bash terminal with the Vscode text editor.

We use the code for integrate Git in windows. while going: -

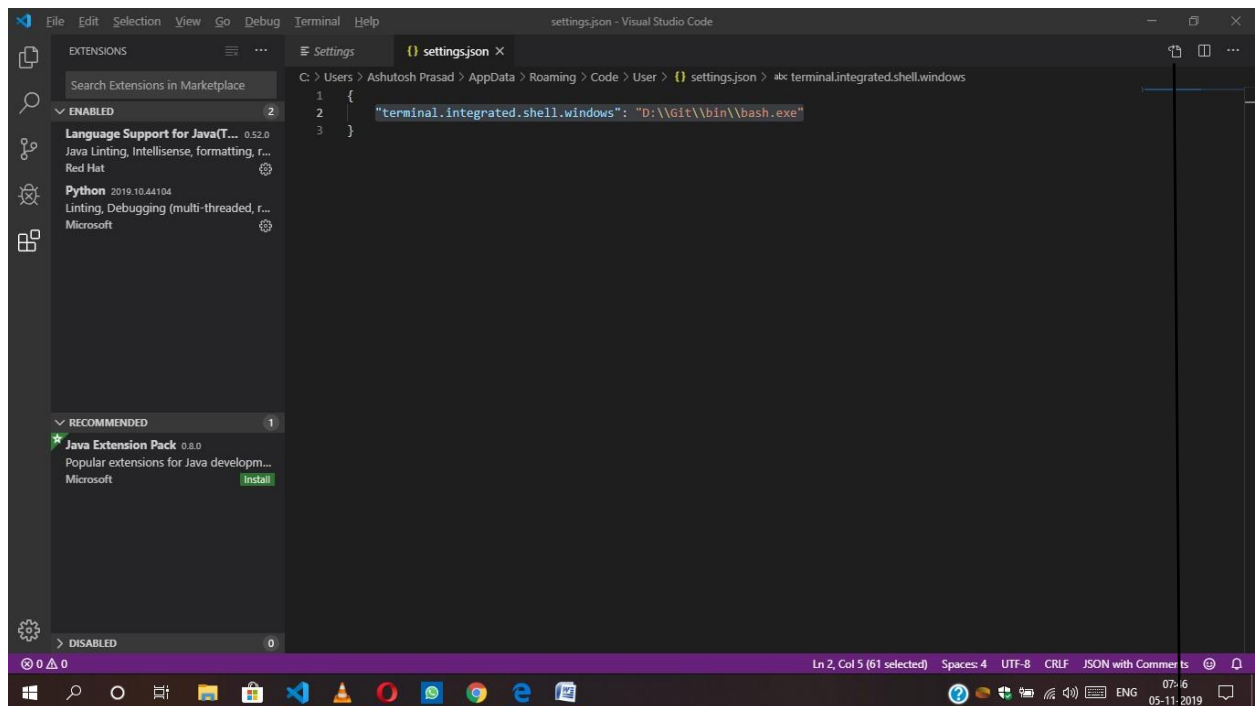
Open file→

Preferences→

Setting→

In the top [] appears click and paste

```
"terminal.integrated.shell.windows": "D:\\Git\\bin\\bash.exe"
```

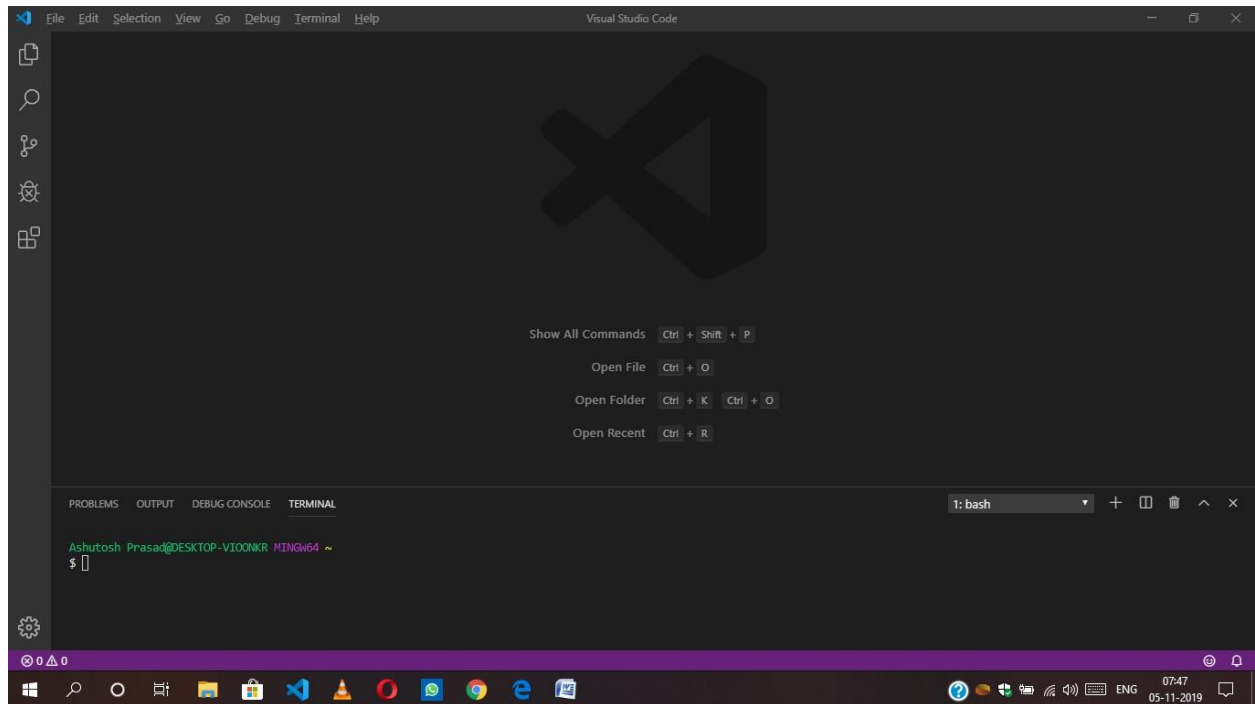


Click

here

Save the setting and closed the Vscode.

When we re-open the Vscode we see the terminal is installed and integrated successfully.



Finishing the environment...

SYSTEM DESIGN

System Design's main aim is to identify the modules that should be in the system, the specifications of these modules and how they interact with each other to produce the desired results. At the end of system design all the major data structures, file formats and the major modules in the system and their specifications are decided. The most creative and challenging phase of the system development process is design phase, it is a solution, a "how to" approach to the creation of the proposed system Design, the first step in the development of an engineered product is initiated only after a clear exposition of expected product functions becomes available. Based on the user requirements and the detailed analysis of a new system, the new system must be designed. This is the phase of system designing. Normally the design proceeds in two stages: preliminary or general design, structure or detailed design.

Preliminary or general design: - In the preliminary or general design, the features of the new system are specified. The costs of implementing these features and benefits to be derived are estimated. If the project is still considered to be feasible, we move to detailed design stage. Structure or detailed design: In the detailed design stage, computer oriented work begins in the earnest. At this stage, the design of the system becomes more structured. Structured design is a blue

print of a computer system solution to a given problem having the same components and inters- relationship among the same components as the original problem. Input, output and processing specifications are drawn up in detail. In the design stage, the programming language and the platform in which the new system will run are also decided. There are several tools and techniques used for designing.

Simple designs are easily understood, easily built, and easily tested. Simplicity is the most important criteria of a design. Other design criteria include the following:

Documentation: - A good design always comes with a set of well-written documents.

Testability: - In a good design, every requirement is testable. A design that cannot be easily tested against its requirements is not acceptable design.

Structure: - A good design presents hierarchical structure that makes logical use of control policies among components.

Modularity: - A good design is modular and exhibits the properties of high cohesiveness and low coupling.

Representation: - A good design should be easily communicated to all interested parties through appropriate abstraction and representation.

Reusability: - a good design should be repeatable and reusable.

MODULES

There are some basic restaurant modules used are as follows,

Price Module - Show the fixed price of the sub modules.
In which we show the price to the customer's satisfaction.

Item	Price
Fries Meal	25
Lunch Meal	40
Burger Meal	35
Pizza Meal	50
Cheese Burger	30
Drinks	35

System Modules - Get the number of meals and calculate the price to be paid by the customer. There is modules are as follows.

Order no

Cost

Service charge

Tax

Subtotal

Total

And some button modules like

Total, reset, exit and have a calculator for calculation of price.

SYSTEM SPECIFICATION

Requirement specification is the part of the project which gives the details about the hardware and software requirement of our project. It also details the features of the programming language used.

HARDWARE REQUIREMENT

Processor - AMD Radeon Processor.
Speed - 2.3 GHz
Memory - 4 GB RAM
Hard drive - minimum 80 GB & above

SOFTWARE REQUIREMENT

Platform - windows 10
Language - Python

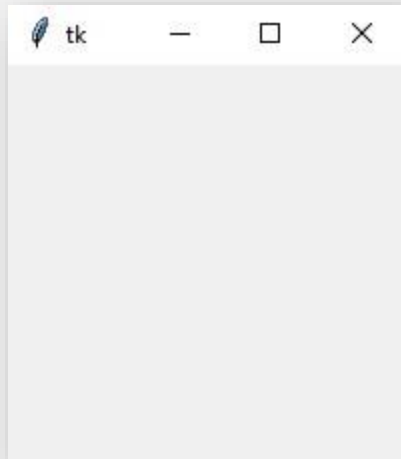
PYTHON LIBRARY USED IN PROJECT

Tkinter – Tkinter is the Python interface to the Tk GUI toolkit shipped with Python. We would look this option in this chapter.



FOR EXAMPLES;

```
import tkinter
Top = Tkinter. Tk()
# Code to add widgets will go here...
top.mainloop ()
```



Tkinter Widgets

Tkinter provides various controls, such as buttons, labels and text boxes used in a GUI application. These controls are commonly called widgets. There are currently 15 types of widgets in Tkinter. We present these widgets as well as a brief description in the following table –

Sr.No.	Operator & Description
1	<u>Button</u> The Button widget is used to display buttons in your application.
2	<u>Canvas</u> The Canvas widget is used to draw shapes, such as lines, ovals, polygons and rectangles, in your application.

3	<p><u>Checkbutton</u></p> <p>The Checkbutton widget is used to display a number of options as checkboxes. The user can select multiple options at a time.</p>
4	<p><u>Entry</u></p> <p>The Entry widget is used to display a single-line text field for accepting values from a user.</p>
5	<p><u>Frame</u></p> <p>The Frame widget is used as a container widget to organize other widgets.</p>
6	<p><u>Label</u></p> <p>The Label widget is used to provide a single-line caption for other widgets. It can also contain images.</p>
7	<p><u>Listbox</u></p> <p>The Listbox widget is used to provide a list of options to a user.</p>
8	<p><u>Menubutton</u></p> <p>The Menubutton widget is used to display menus in your</p>

	application.
9	<u>Menu</u> The Menu widget is used to provide various commands to a user. These commands are contained inside Menubutton.
10	<u>Message</u> The Message widget is used to display multiline text fields for accepting values from a user.
11	<u>Radiobutton</u> The Radiobutton widget is used to display a number of options as radio buttons. The user can select only one option at a time.
12	<u>Scale</u> The Scale widget is used to provide a slider widget.
13	<u>Scrollbar</u> The Scrollbar widget is used to add scrolling capability to various widgets, such as list boxes.
14	<u>Text</u> The Text widget is used to display text in multiple lines.

15	<u>Toplevel</u> The Toplevel widget is used to provide a separate window container.
16	<u>Spinbox</u> The Spinbox widget is a variant of the standard Tkinter Entry widget, which can be used to select from a fixed number of values.
17	<u>PanedWindow</u> A PanedWindow is a container widget that may contain any number of panes, arranged horizontally or vertically.
18	<u>LabelFrame</u> A labelframe is a simple container widget. Its primary purpose is to act as a spacer or container for complex window layouts.
19	<u>tkMessageBox</u> This module is used to display message boxes in your applications.

Geometry Management

All Tkinter widgets have access to specific geometry management methods, which have the purpose of organizing widgets throughout the parent widget area. Tkinter exposes the following geometry manager classes: pack, grid, and place.

- The *pack()* Method – This geometry manager organizes widgets in blocks before placing them in the parent widget.
- The *grid()* Method – This geometry manager organizes widgets in a table-like structure in the parent widget.
- The *place()* Method – This geometry manager organizes widgets by placing them in a specific position in the parent widget.

CODE OF PROJECT

```
from tkinter import*
import random
import time

root = Tk()
root.geometry("1600x700+0+0")
root.title("Restaurant Management System")

Tops = Frame(root,bg="white",width = 1600,height=50,relief=SUNKEN)
Tops.pack(side=TOP)

f1 = Frame(root, width=900, height=700, relief=SUNKEN)
f1.pack(side=LEFT)

f2 = Frame(root, width=400, height=700, relief=SUNKEN)
f2.pack(side=RIGHT)

#-----TIME-----

localtime = time.asctime(time.localtime(time.time()))
```

```
#-----INFO TOP-----
```

```
lblinfo = Label(Tops, font=('ariel',30,'bold'), text="Restaurant  
Management System",
```

```
    fg="steel blue", bd=10,
```

```
    anchor='w')
```

```
lblinfo.grid(row=0,column=0)
```

```
lblinfo = Label(Tops, font=('ariel',20,'bold'), text=localtime, fg="steel  
blue",
```

```
    anchor='w')
```

```
lblinfo.grid(row=1,column=0)
```

```
#-----Calculator-----
```

```
text_Input=StringVar()
```

```
operator = ""
```

```
txtdisplay = Entry(f2,font=('ariel',20,'bold'), textvariable=text_Input,  
bd=5,
```

```
    insertwidth=7, bg="white", justify='right')
```

```
txtdisplay.grid(columnspan=4)
```

```
def btnclick(numbers):  
    global operator  
    operator = operator + str(numbers)  
    text_Input.set(operator)
```

```
def clrdisplay():  
    global operator  
    operator = ""  
    text_Input.set("")
```

```
def eqals():  
    global operator  
    sumup = str(eval(operator))  
  
    text_Input.set(sumup)  
    operator = ""
```

```
def Ref():  
    x = random.randint(12980, 50876)  
    randomRef = str(x)  
    rand.set(randomRef)
```

```
cof = float(Fries.get())
```

```
colfries = float(Largefries.get())
```

```
cob = float(Burger.get())
```

```
cofi = float(Filet.get())
```

```
cochee = float(Cheese_burger.get())
```

```
codr = float(Drinks.get())
```

```
costoffries = cof*25
```

```
costoflargefries = colfries*40
```

```
costofburger = cob*35
```

```
costoffilet = cofi*50
```

```
costofcheeseburger = cochee*50
```

```
costofdrinks = codr*35
```

```
costofmeal = "Rs.",str('%0.2f' % (costoffries + costoflargefries +  
costofburger
```

```
+ costoffilet + costofcheeseburger +  
costofdrinks))
```

```
PayTax = ((costoffries + costoflargefries + costofburger + costoffilet +  
costofcheeseburger + costofdrinks)*0.33)
```

```
Totalcost = (costoffries + costoflargefries + costofburger + costoffilet  
+
```

```
costofcheeseburger + costofdrinks)
```

```
Ser_Charge = ((costoffries + costoflargefries + costofburger +  
costoffilet +
```

```
costofcheeseburger + costofdrinks)/99)
```

```
Service = "Rs.",str('%0.2f'% Ser_Charge)
```

```
OverAllCost = "Rs.",str( PayTax + Totalcost + Ser_Charge)
```

```
PaidTax = "Rs.",str('%0.2f'% PayTax)
```

```
Service_Charge.set(Service)
```

```
cost.set(costofmeal)
```

```
Tax.set(PaidTax)
```

```
Subtotal.set(costofmeal)
```

```
Total.set(OverAllCost)
```

```
def qexit():
```

```
    root.destroy()
```

```
def reset():
```

```
    rand.set("")
```



```
Fries.set("")
Largefries.set("")
Burger.set("")
Filet.set("")
Subtotal.set("")
Total.set("")
Service_Charge.set("")
Drinks.set("")
Tax.set("")
cost.set("")
Cheese_burger.set("")
```

```
btn7 = Button(f2, padx=16, pady=16, bd=4, fg="black",
font=('arial',20,'bold'),
            text="7", bg="powder blue", command=lambda: btnclick(7) )
btn7.grid(row=2, column=0)
```

```
btn8 = Button(f2, padx=16, pady=16, bd=4, fg="black",
font=('arial',20,'bold'),
            text="8", bg="powder blue", command=lambda: btnclick(8) )
```

```
btn8.grid(row=2, column=1)
```

```
btn9 = Button(f2, padx=16, pady=16, bd=4, fg="black",  
font=('arial',20,'bold'),  
text="9", bg="powder blue", command=lambda: btnclick(9) )  
btn9.grid(row=2, column=2)
```

```
Addition = Button(f2, padx=16, pady=16, bd=4, fg="black",  
font=('arial',20,'bold'),  
text="+",bg="powder blue", command=lambda: btnclick("+")  
)  
Addition.grid(row=2, column=3)
```

```
#-----
```

```
btn4 = Button(f2, padx=16, pady=16, bd=4, fg="black",  
font=('arial',20,'bold'),  
text="4", bg="powder blue", command=lambda: btnclick(4) )  
btn4.grid(row=3, column=0)
```

```
btn5 = Button(f2, padx=16, pady=16, bd=4, fg="black",  
font=('arial',20,'bold'),
```

```
text="5", bg="powder blue", command=lambda: btnclick(5) )  
btn5.grid(row=3, column=1)
```

```
btn6 = Button(f2, padx=16, pady=16, bd=4, fg="black",  
font=('ariel',20,'bold'),  
text="6", bg="powder blue", command=lambda: btnclick(6) )  
btn6.grid(row=3, column=2)
```

```
Substraction = Button(f2, padx=16, pady=16, bd=4, fg="black",  
font=('ariel',20,'bold'), text="-", bg="powder blue",  
command=lambda: btnclick("-") )  
Substraction.grid(row=3, column=3)
```

```
#-----
```

```
--
```

```
btn1 = Button(f2, padx=16, pady=16, bd=4, fg="black",  
font=('ariel',20,'bold'),  
text="1", bg="powder blue", command=lambda: btnclick(1) )  
btn1.grid(row=4, column=0)
```

```
btn2 = Button(f2, padx=16, pady=16, bd=4, fg="black",
font=('arial',20,'bold'),
            text="2", bg="powder blue", command=lambda: btnclick(2) )
btn2.grid(row=4, column=1)
```

```
btn3 = Button(f2, padx=16, pady=16, bd=4, fg="black",
font=('arial',20,'bold'),
            text="3", bg="powder blue", command=lambda: btnclick(3) )
btn3.grid(row=4,column=2)
```

```
multiply = Button(f2, padx=16, pady=16, bd=4, fg="black",
font=('arial',20,'bold'),
            text="*", bg="powder blue", command=lambda: btnclick("*") )
multiply.grid(row=4, column=3)
```

```
#-----
```

```
---
```

```
btn0 = Button(f2, padx=16, pady=16, bd=4, fg="black",
font=('arial',20,'bold'),
            text="0", bg="powder blue", command=lambda: btnclick(0) )
btn0.grid(row=5, column=0)
```

```
btnc = Button(f2, padx=16, pady=16, bd=4, fg="black",  
font=('arial',20,'bold'),  
text="c", bg="powder blue", command=clrdisplay)  
btnc.grid(row=5, column=1)
```

```
btnequal = Button(f2, padx=16, pady=16, bd=4, width=16, fg="black",  
font=('arial',20,'bold'), text "=", bg="powder blue",  
command>equals)  
btnequal.grid(columnspan=4)
```

```
Decimal = Button(f2, padx=16, pady=16, bd=4, fg="black",  
font=('arial',20,'bold'),  
text=".", bg="powder blue", command=lambda: btnclick(".") )  
Decimal.grid(row=5, column=2)
```

```
Division = Button(f2, padx=16, pady=16, bd=4, fg="black", font=('arial',  
20 ,'bold'),  
text="/", bg="powder blue", command=lambda: btnclick("/") )  
Division.grid(row=5, column=3)
```

```
status = Label(f2, font=('aria',15,'bold'), width=16, text="-By XXXXXX",  
bd=2,
```

```
relief=SUNKEN)
```

```
status.grid(row=7, columnspan=3)
```

```
#-----
```

```
rand = StringVar()
```

```
Fries = StringVar()
```

```
Largefries = StringVar()
```

```
Burger = StringVar()
```

```
Filet = StringVar()
```

```
Subtotal = StringVar()
```

```
Total = StringVar()
```

```
Service_Charge = StringVar()
```

```
Drinks = StringVar()
```

```
Tax = StringVar()
```

```
cost = StringVar()
```

```
Cheese_burger = StringVar()
```

```
lblreference = Label(f1, font=( 'aria' ,16, 'bold' ),text="Order  
No.",fg="steel blue",bd=10,anchor='w')  
lblreference.grid(row=0,column=0)  
txtreference = Entry(f1,font=('ariel' ,16,'bold'), textvariable=rand ,  
bd=6,insertwidth=4,bg="powder blue" ,justify='right')  
txtreference.grid(row=0,column=1)
```

```
lblfries = Label(f1, font=( 'aria' ,16, 'bold' ),text="Fries Meal",fg="steel  
blue",bd=10,anchor='w')  
lblfries.grid(row=1,column=0)  
txtfries = Entry(f1,font=('ariel' ,16,'bold'), textvariable=Fries ,  
bd=6,insertwidth=4,bg="powder blue" ,justify='right')  
txtfries.grid(row=1,column=1)
```

```
lblLargefries = Label(f1, font=( 'aria' ,16, 'bold' ),text="Lunch  
Meal",fg="steel blue",bd=10,anchor='w')  
lblLargefries.grid(row=2,column=0)  
txtLargefries = Entry(f1,font=('ariel' ,16,'bold'), textvariable=Largefries ,  
bd=6,insertwidth=4,bg="powder blue" ,justify='right')  
txtLargefries.grid(row=2,column=1)
```

```
lblburger = Label(f1, font=( 'aria' ,16, 'bold' ),text="Burger  
Meal",fg="steel blue",bd=10,anchor='w')  
lblburger.grid(row=3,column=0)  
txtburger = Entry(f1,font=('ariel' ,16,'bold'), textvariable=Burger ,  
bd=6,insertwidth=4,bg="powder blue" ,justify='right')  
txtburger.grid(row=3,column=1)
```

```
lblFilet = Label(f1, font=( 'aria' ,16, 'bold' ),text="Pizza Meal",fg="steel  
blue",bd=10,anchor='w')  
lblFilet.grid(row=4,column=0)  
txtFilet = Entry(f1,font=('ariel' ,16,'bold'), textvariable=Filet ,  
bd=6,insertwidth=4,bg="powder blue" ,justify='right')  
txtFilet.grid(row=4,column=1)
```

```
lblCheese_burger = Label(f1, font=( 'aria' ,16, 'bold' ),text="Cheese  
burger",fg="steel blue",bd=10,anchor='w')  
lblCheese_burger.grid(row=5,column=0)  
txtCheese_burger = Entry(f1,font=('ariel' ,16,'bold'),  
textvariable=Cheese_burger , bd=6,insertwidth=4,bg="powder blue"  
,justify='right')
```



```
txtCheese_burger.grid(row=5,column=1)
```

```
#-----
```

```
lblDrinks = Label(f1, font=( 'aria' ,16, 'bold' ),text="Drinks",fg="steel  
blue",bd=10,anchor='w')
```

```
lblDrinks.grid(row=0,column=2)
```

```
txtDrinks = Entry(f1,font=('ariel' ,16,'bold'), textvariable=Drinks ,  
bd=6,insertwidth=4,bg="powder blue" ,justify='right')
```

```
txtDrinks.grid(row=0,column=3)
```

```
lblcost = Label(f1, font=( 'aria' ,16, 'bold' ),text="cost",fg="steel  
blue",bd=10,anchor='w')
```

```
lblcost.grid(row=1,column=2)
```

```
txtcost = Entry(f1,font=('ariel' ,16,'bold'), textvariable=cost ,  
bd=6,insertwidth=4,bg="powder blue" ,justify='right')
```

```
txtcost.grid(row=1,column=3)
```

```
lblService_Charge = Label(f1, font=( 'aria' ,16, 'bold' ),text="Service  
Charge",fg="steel blue",bd=10,anchor='w')
```

```
lblService_Charge.grid(row=2,column=2)
```

```
txtService_Charge = Entry(f1,font=('ariel' ,16,'bold'),  
textvariable=Service_Charge , bd=6,insertwidth=4,bg="powder blue"  
,justify='right')  
txtService_Charge.grid(row=2,column=3)
```

```
lblTax = Label(f1, font=( 'aria' ,16, 'bold' ),text="Tax",fg="steel  
blue",bd=10,anchor='w')  
lblTax.grid(row=3,column=2)  
txtTax = Entry(f1,font=('ariel' ,16,'bold'), textvariable=Tax ,  
bd=6,insertwidth=4,bg="powder blue" ,justify='right')  
txtTax.grid(row=3,column=3)
```

```
lblSubtotal = Label(f1, font=( 'aria' ,16, 'bold' ),text="Subtotal",fg="steel  
blue",bd=10,anchor='w')  
lblSubtotal.grid(row=4,column=2)  
txtSubtotal = Entry(f1,font=('ariel' ,16,'bold'), textvariable=Subtotal ,  
bd=6,insertwidth=4,bg="powder blue" ,justify='right')  
txtSubtotal.grid(row=4,column=3)
```

```
lblTotal = Label(f1, font=( 'aria' ,16, 'bold' ),text="Total",fg="steel  
blue",bd=10,anchor='w')
```

```
lblTotal.grid(row=5,column=2)
```

```
txtTotal = Entry(f1,font=('ariel' ,16,'bold'), textvariable=Total ,
```

```
bd=6,insertwidth=4,bg="powder blue" ,justify='right')
```

```
txtTotal.grid(row=5,column=3)
```

```
#-----buttons-----
```

```
lblTotal = Label(f1,text="-----",fg="white")
```

```
lblTotal.grid(row=6,columnspan=3)
```

```
btnTotal=Button(f1,padx=16,pady=8, bd=10 ,fg="black",font=('ariel'  
,16,'bold'),width=10, text="TOTAL", bg="powder blue",command=Ref)
```

```
btnTotal.grid(row=7, column=1)
```

```
btnreset=Button(f1,padx=16,pady=8, bd=10 ,fg="black",font=('ariel'  
,16,'bold'),width=10, text="RESET", bg="powder blue",command=reset)
```

```
btnreset.grid(row=7, column=2)
```

```
btnexit=Button(f1,padx=16,pady=8, bd=10 ,fg="black",font=('ariel'  
,16,'bold'),width=10, text="EXIT", bg="powder blue",command=qexit)
```

```
btnexit.grid(row=7, column=3)
```

```
def price():  
    roo = Tk()  
    roo.geometry("600x220+0+0")  
    roo.title("Price List")  
    lblinfo = Label(roo, font=('aria', 15, 'bold'), text="ITEM", fg="black",  
bd=5)  
    lblinfo.grid(row=0, column=0)  
    lblinfo = Label(roo, font=('aria', 15, 'bold'), text="_____",  
fg="white", anchor=W)  
    lblinfo.grid(row=0, column=2)  
    lblinfo = Label(roo, font=('aria', 15, 'bold'), text="PRICE", fg="black",  
anchor=W)  
    lblinfo.grid(row=0, column=3)  
    lblinfo = Label(roo, font=('aria', 15, 'bold'), text="Fries Meal",  
fg="steel blue", anchor=W)  
    lblinfo.grid(row=1, column=0)  
    lblinfo = Label(roo, font=('aria', 15, 'bold'), text="25", fg="steel blue",  
anchor=W)  
    lblinfo.grid(row=1, column=3)  
    lblinfo = Label(roo, font=('aria', 15, 'bold'), text="Lunch Meal",  
fg="steel blue", anchor=W)
```

```
lblinfo.grid(row=2, column=0)
```

```
lblinfo = Label(roo, font=('aria', 15, 'bold'), text="40", fg="steel blue",  
anchor=W)
```

```
lblinfo.grid(row=2, column=3)
```

```
lblinfo = Label(roo, font=('aria', 15, 'bold'), text="Burger Meal",  
fg="steel blue", anchor=W)
```

```
lblinfo.grid(row=3, column=0)
```

```
lblinfo = Label(roo, font=('aria', 15, 'bold'), text="35", fg="steel blue",  
anchor=W)
```

```
lblinfo.grid(row=3, column=3)
```

```
lblinfo = Label(roo, font=('aria', 15, 'bold'), text="Pizza Meal",  
fg="steel blue", anchor=W)
```

```
lblinfo.grid(row=4, column=0)
```

```
lblinfo = Label(roo, font=('aria', 15, 'bold'), text="50", fg="steel blue",  
anchor=W)
```

```
lblinfo.grid(row=4, column=3)
```

```
lblinfo = Label(roo, font=('aria', 15, 'bold'), text="Cheese Burger",  
fg="steel blue", anchor=W)
```

```
lblinfo.grid(row=5, column=0)
```

```
lblinfo = Label(roo, font=('aria', 15, 'bold'), text="30", fg="steel blue",  
anchor=W)
```

```
lblinfo.grid(row=5, column=3)
```

```
lblinfo = Label(roo, font=('aria', 15, 'bold'), text="Drinks", fg="steel  
blue", anchor=W)
```

```
lblinfo.grid(row=6, column=0)
```

```
lblinfo = Label(roo, font=('aria', 15, 'bold'), text="35", fg="steel blue",  
anchor=W)
```

```
lblinfo.grid(row=6, column=3)
```

```
roo.mainloop()
```

```
btnprice=Button(f1,padx=16,pady=8, bd=10 ,fg="black",font=('ariel'  
,16,'bold'),width=10, text="PRICE", bg="powder blue",command=price)
```

```
btnprice.grid(row=7, column=0)
```

```
root.mainloop()
```

CONCLUSION

This project has really been faithful and informative. It has made us learn and understand the many trivial concepts of Python Language. As we have used python Tkinter as a GUI it provides various controls, such as buttons, labels and text boxes to build a user friendly application.

The fast growing use of internet confirms the good future and scope of the proposed project.

Finally it has taught us a valuable lifelong lesson about the improvements and working and interacting in a group.

Canteen billing system is developed considering the requirement of people. The software fulfilled most of the basic requirements of the user. The system is not efficient and no additional training is required to work on this software.

Project View

Restaurant Management System

Restaurant Management System

Thu Nov 7 03:12:46 2019

Order No.		Drinks	
Fries Meal		cost	
Lunch Meal		Service Charge	
Burger Meal		Tax	
Pizza Meal		Subtotal	
Cheese burger		Total	

PRICE

TOTAL

RESET

EXIT

7	8	9	+
4	5	6	-
1	2	3	*
0	c	.	/
=			
-By XXXXXX			

We have added a calculator for instant calculation –

