

PROJECT REPORT

RESTAURANT BILLING SYSTEM

(<https://github.com/surbhirajpal/project>)

INDUSTRIAL TRAINING
(SUMMER TRAINING)

SUBMITTED BY:

SURBHI RAJPAL (UE168108)



DEPARTMENT OF
INFORMATION TECHNOLOGY

UNIVERSITY INSTITUTE OF ENGINEERING
AND TECHNOLOGY

UNIVERSITY INSTITUTE OF ENGINEERING AND TECHNOLOGY PANJAB UNIVERSITY CHANDIGARH



CANDIDATE'S DECLARATION

I hereby certify that the work presented in the project entitled **RESTAURANT BILLING SYSTEM** as per the requirement of the subject summer training for the award of the degree bachelor of engineering in information technology, **University Institute Of Engineering And Technology, Panjab University Chandigarh** is an authentic record of my own work carried out under the mentors of Acadview, Chandigarh .

Date:

Place: Chandigarh

SURBHI RAJPAL

UE168108

Chandigarh

Acknowledgement

This major project would not have been possible without the valuable assistance of many people to whom I am indebted, in particular, my mentor of Acadview, Chandigarh.

He guided me through the training, provided helpful insights and motivated to work harder. His constant guidance and willingness to share his vast knowledge made me understand this project and its manifestation in great depths which helped in completing the task assigned.

I would also like to thank **Department of Information Technology, University Institute of Engineering and Technology** for providing me with the best knowledge and helping me in every doubt I had.

TABLE OF CONTENT

1. Introduction.....	1
1.1. The Python Language.....	1
1.2. The Workflow.....	2
2. Restaurant Billing System.....	3
2.1. Objective.....	3
2.2. Components of Application.....	3
2.3. Scenario.....	10
3. System Development.....	12
3.1. Welcome Window.....	12
3.2. Login Form.....	12
3.3. Main Window.....	14
3.4. Price List Window.....	15
3.5. Generated Bill.....	15
3.6. Database Table.....	16
4. Conclusion.....	17
5. Drawbacks.....	18
6. References.....	19

CHAPTER - 1

INTRODUCTION

1.1. The Python Language

Python is a general purpose, dynamic, high level and interpreted programming language. It supports Object Oriented programming approach to develop applications. It is simple and easy to learn and provides lots of high-level data structures. Python is easy to learn yet powerful and versatile scripting language which makes it attractive for Application Development.

Python's syntax and dynamic typing with its interpreted nature, makes it an ideal language for scripting and rapid application development. Python supports multiple programming pattern, including object oriented, imperative and functional or procedural programming styles. Python is not intended to work on special area such as web programming. That is why it is known as multipurpose because it can be used with web, enterprise, 3D CAD etc. We don't need to use data types to declare variable because it is dynamically typed so we can write `a=10` to assign an integer value in an integer variable. Python makes the development and debugging fast because there is no compilation step included in python development and edit-test-debug cycle is very fast.

Some specific features of Python are as follows:

- An interpreted (as opposed to compiled) language. Contrary to e.g. C or Fortran, one does not compile Python code before executing it. In addition, Python can be used **interactively**: many Python interpreters are available, from which commands and scripts can be executed.
- A free software released under an **open-source** license: Python can be used and distributed free of charge, even for building commercial software.
- Multi-platform: Python is available for all major operating systems, Windows, Linux/Unix, MacOS X, most likely your mobile phone OS, etc.
- A very readable language with clear non-verbose syntax
- A language for which a large variety of high-quality packages are available for various applications, from web frameworks to scientific computing.
- A language very easy to interface with other languages, in particular C and C++.
- Python is an object-oriented language, with dynamic typing (the same variable can contain objects of different types during the course of a program).

Python Strength:

- **Batteries included** Rich collection of already existing **bricks** of classic numerical methods, plotting or data processing tools. We don't want to

re-program the plotting of a curve, a Fourier transform or a fitting algorithm. Don't reinvent the wheel!

- **Easy to learn** Most scientists are not payed as programmers, neither have they been trained so. They need to be able to draw a curve, smooth a signal, do a Fourier transform in a few minutes.
- **Easy communication** To keep code alive within a lab or a company it should be as readable as a book by collaborators, students, or maybe customers. Python syntax is simple, avoiding strange symbols or lengthy routine specifications that would divert the reader from mathematical or scientific understanding of the code.
- **Efficient code** Python numerical modules are computationally efficient. But needless to say that a very fast code becomes useless if too much time is spent writing it. Python aims for quick development times and quick execution times.
- **Universal** Python is a language used for many different problems. Learning Python avoids learning a new software for each new problem.

1.2. The workflow: interactive environments and text editors

Python is a general-purpose language. As such, there is not one blessed environment to work in, and not only one way of using it. Although this makes it harder for beginners to find their way, it makes it possible for Python to be used for programs, in web servers, or embedded devices.

As you move forward, it will be important to not only work interactively, but also to create and reuse Python files. For this, a powerful code editor will get you far. Here are several good easy-to-use editors:

- [Spyder](#): integrates an IPython console, a debugger, a profiler...
- [PyCharm](#): integrates an IPython console, notebooks, a debugger...
(freely available, but commercial)
- [Atom](#)

CHAPTER-2

RESTAURANT BILLING SYSTEM

2.1. Objective

Restaurant billing system is very much needed. It is found that the billing system and maintenance of records of the cashier is done manually using a file based method in some restaurants. Therefore we gave our attention to avoid this weakness by building a computerized system to replace the file based system that will help the restaurant to give a more efficient, effective, productive service to the customer. It reduces a lot of hardwork and complexity. Nowadays restaurants are filled with people so it's very difficult to manage all the work manually. That's why these kind of softwares are required.

Main objective was to develop a computerized system to store and retrieve billing records of customers. To computerize the file based system I designed a system which enables easy access to billing records and restaurant records etc.. This application is built in Python to maintain the orders and calculate the prices of orders at the restaurant. It allows a user to calculate the price, subtotal, taxes and total amount of the order by managing the quantities and prices of the items. A special feature of this system is the login method which requires a username and a password which increases the security of the stored data. Throughout the system we gave special attention to build interfaces that are user friendly which will make our system a successful one.

2.2. Components of the Application

2.2.1. Libraries Imported

- Tkinter - Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.
- sqlite3 - SQLite is a C library that provides a lightweight disk-based database that doesn't require a separate server process and allows accessing the database using a nonstandard variant of the SQL query language. Some applications can use SQLite for internal data storage. It's also possible to prototype an application using SQLite and then port the code to a larger database such as PostgreSQL or Oracle. To use the module, you must first

create a **Connection** object that represents the database. Here the data will be stored in the example.db file:

```
import sqlite3
conn = sqlite3.connect('example.db')
```

- Time - This module provides various time-related functions. For related functionality, the datetime and calendar modules can also be used.
- PIL - Python Imaging Library (abbreviated as PIL) (in newer versions known as Pillow) is a **free library** for the **Python programming language** that adds support for opening, manipulating, and saving many different image file formats. It is available for Windows, Mac OS X and Linux.
- Os - The OS module in Python provides a way of using operating system dependent functionality. The functions that the OS module provides allows you to interface with the underlying operating system that Python is running on – be that Windows, Mac or Linux. You can find important information about your location or about the process.

2.2.2. Front End

The GUI of this application is built using Python Tkinter library.

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps –

- Import the Tkinter module.
- Create the GUI application main window.
- Add widgets to the GUI application.
- Enter the main event loop to take action against each event triggered by the user.

Importing tkinter is same as importing any other module in the python code. Note that the name of the module in Python 2.x is 'Tkinter' and in Python 3.x is 'tkinter'.

```
import tkinter
```

There are two main methods used you the user need to remember while creating the Python application with GUI.

- Tk(screenName=None, baseName=None, className='Tk', useTk=1): To create a main window, tkinter offers a method 'Tk(screenName=None, baseName=None, className='Tk', useTk=1)'. To

change the name of the window, you can change the className to the desired one. The basic code used to create the main window of the application is:

```
m=tkinter.Tk() where m is the name of the main window object or  
m=Tk()
```

- `mainloop()`: There is a method known by the name `mainloop()` is used when you are ready for the application to run. `mainloop()` is an infinite loop used to run the application, wait for an event to occur and process the event till the window is not closed.

```
m.mainloop()
```

```
import tkinter  
m = tkinter.Tk()  
...  
widgets are added here  
...  
m.mainloop()
```

Tkinter also offers access to the geometric configuration of the widgets which can organize the widgets in the parent windows. There are mainly three geometry manager classes class.

- `pack()` method:It organizes the widgets in blocks before placing in the parent widget.
- `grid()` method:It organizes the widgets in grid (table-like structure) before placing in the parent widget.
- `place()` method:It organizes the widgets by placing them on specific positions directed by the programmer.

There are a number of widgets used in this tkinter application. Some of the major widgets are explained below:

1. Button:

To add a button in your application, this widget is used.

The general syntax is:

```
w=Button(master, option=value)
```

master is the parameter used to represent the parent window.

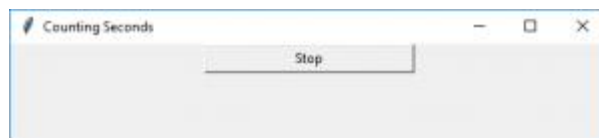
There are number of options which are used to change the format of the Buttons. Number of options can be passed as parameters separated by commas. Some of them are listed below.

- `activebackground`: to set the background color when button is under the cursor.
- `activeforeground`: to set the foreground color when button is under the cursor.
- `bg`: to set he normal background color.

- command: to call a function.
- font: to set the font on the button label.
- image: to set the image on the button.
- width: to set the width of the button.
- height: to set the height of the button.

```
import tkinter as tk
r = tk.Tk()
r.title('Counting Seconds')
button = tk.Button(r, text='Stop', width=25,
command=r.destroy)
button.pack()
r.mainloop()
```

Output:



2. Entry:

It is used to input the single line text entry from the user. For multi-line text input, Text widget is used.

The general syntax is:

```
w=Entry(master, option=value)
```

master is the parameter used to represent the parent window.

There are number of options which are used to change the format of the widget. Number of options can be passed as parameters separated by commas. Some of them are listed below.

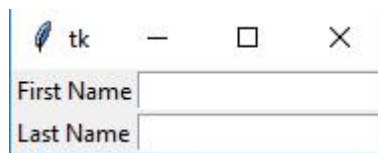
- bd: to set the border width in pixels.
- bg: to set the normal background color.
- cursor: to set the cursor used.
- command: to call a function.
- highlightcolor: to set the color shown in the focus highlight.
- width: to set the width of the button.
- height: to set the height of the button.

```

from tkinter import *
master = Tk()
Label(master, text='First
Name').grid(row=0)
Label(master, text='Last
Name').grid(row=1)
e1 = Entry(master)
e2 = Entry(master)
e1.grid(row=0, column=1)
e2.grid(row=1, column=1)
mainloop()

```

Output:



3. Frame:

It acts as a container to hold the widgets. It is used for grouping and organizing the widgets.

The general syntax is:

```
w = Frame(master, option=value)
```

master is the parameter used to represent the parent window.

There are number of options which are used to change the format of the widget. Number of options can be passed as parameters separated by commas. Some of them are listed below.

- highlightcolor: To set the color of the focus highlight when widget has to be focused.
- bd: to set the border width in pixels.
- bg: to set the normal background color.
- cursor: to set the cursor used.
- width: to set the width of the widget.
- height: to set the height of the widget.

```

from tkinter import *

root = Tk()
frame = Frame(root)
frame.pack()
bottomframe = Frame(root)
bottomframe.pack( side = BOTTOM )
redbutton = Button(frame, text = 'Red', fg = 'red')
redbutton.pack( side = LEFT)
greenbutton = Button(frame, text = 'Brown', fg='brown')
greenbutton.pack( side = LEFT )
bluebutton = Button(frame, text = 'Blue', fg = 'blue')
bluebutton.pack( side = LEFT )
blackbutton = Button(bottomframe, text = 'Black', fg='black')
blackbutton.pack( side = BOTTOM)
root.mainloop()

```

Output:



4. Label:

It refers to the display box where you can put any text or image which can be updated any time as per the code.

The general syntax is:

```
w=Label(master, option=value)
```

master is the parameter used to represent the parent window.

There are number of options which are used to change the format of the widget. Number of options can be passed as parameters separated by commas. Some of them are listed below.

- bg: to set the normal background color.
- bg to set the normal background color.
- command: to call a function.
- font: to set the font on the button label.
- image: to set the image on the button.
- width: to set the width of the button.
- Height: to set the height of the button.

```

from tkinter import *
root = Tk()
w = Label(root, text='GeeksForGeeks.org!')
w.pack()
root.mainloop()

```

Output:



5. tkMessageBox :

The tkMessageBox module is used to display message boxes in your applications. This module provides a number of functions that you can use to display an appropriate message. Some of these functions are showinfo, showwarning, showerror, askquestion, askokcancel, askyesno, and askretryignore.

The general syntax is –

```
tkMessageBox.FunctionName(title, message [, options])
```

Parameters

- FunctionName – This is the name of the appropriate message box function.
- title – This is the text to be displayed in the title bar of a message box.
- message – This is the text to be displayed as a message.
- options – options are alternative choices that you may use to tailor a standard message box. Some of the options that you can use are default and parent. The default option is used to specify the default button, such as ABORT, RETRY, or IGNORE in the message box. The parent option is used to specify the window on top of which the message box is to be displayed.

```

from tkinter import *
root = Tk()
def hello():
    tkMessageBox.showinfo("Say Hello", "Hello World")
B1=Button(root,text= "Say Hello", command=hello)
B1.pack()
root.mainloop()

```

Output:



2.2.3. Backend Calculations:

The program has a prespecified menu with their prices . The user specifies the number of portions he like to order and the backend calculates the total price including the service charges and tax. The service charges and tax rates are to be specified by the programmer.

Workflow:

- Prepare a menu and price list for the restaurant.
- A GUI interface for the billing system using tkinter .
- Input boxes for the number of portions of different items in the menu.
- Classes in the backend for different functions like calculate taxes, service charges and total price of the meal including all taxes.
- Button operations for total, price list, reset and exit.

2.3. Scenario

The newly designed and developed computerized system to be implemented in the restaurant consists of about four interfaces created with the help of tkinter (A python library). Firstly there is a Welcome Form. In it There are two buttons, one is to login and the other is to cancel. Once you login then you go into the second interface which is the login form. In the login form you have to enter your username and password to proceed forward. Once you enter the correct username and password you go to the main window. In the main form there are five buttons.

- Price List -Clicking this button will give you price list of menu offered.
- Total Amount - Clicking this button will calculate total amount to be paid by customer.
- Reset - Clicking this button will reset all quantities offered by user and give zero default value to them.
- Exit - Clicking this button will close the application window.
- Generate Bill - Clicking this button will generate the bill of current order.

With these five buttons there are fourteen labels of order,menu and payments and fourteen entries boxes out of which order no,cost,service charge,tax,subtotal,total is disabled. In enabled entries,user is required to fill the quantity of food ordered.The order record is connected to the database.

CHAPTER - 3

SYSTEM DEVELOPMENT

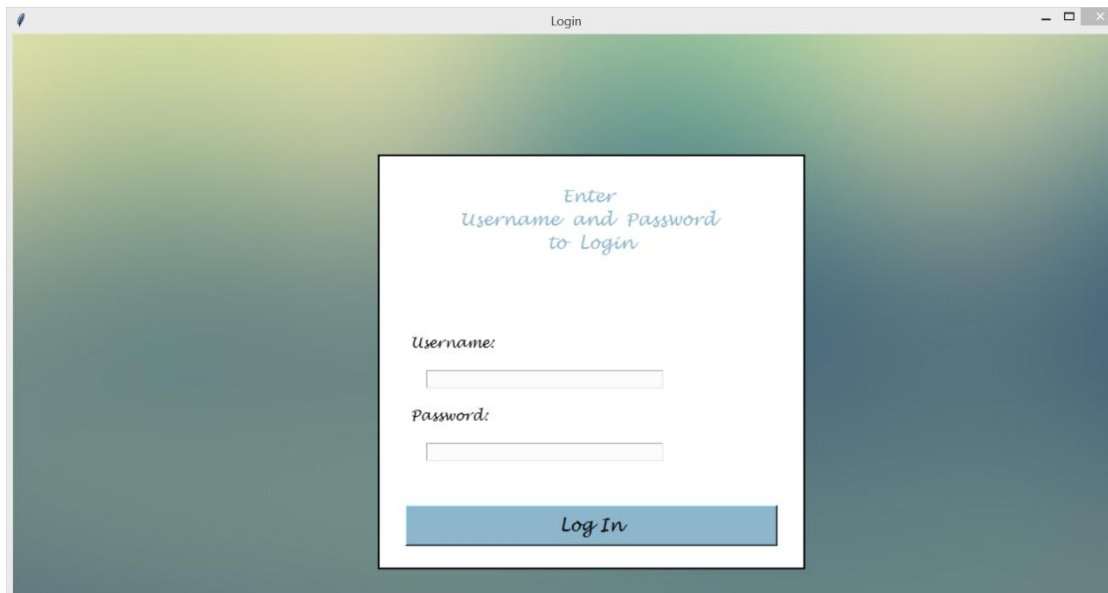
3.1. Welcome Window



In this welcome window, there are two buttons.

- LOGIN - Clicking this button will open another interface for login.
- CANCEL - Clicking this button will close this window.

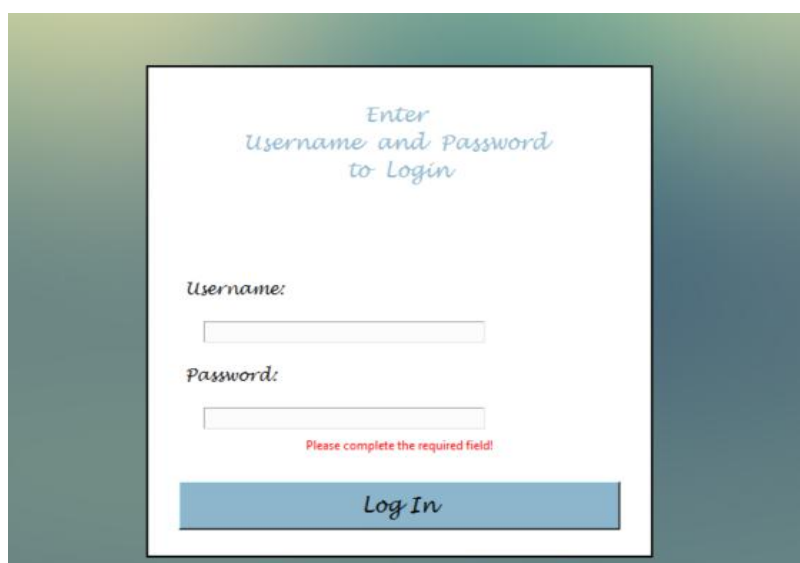
3.2. Login Form



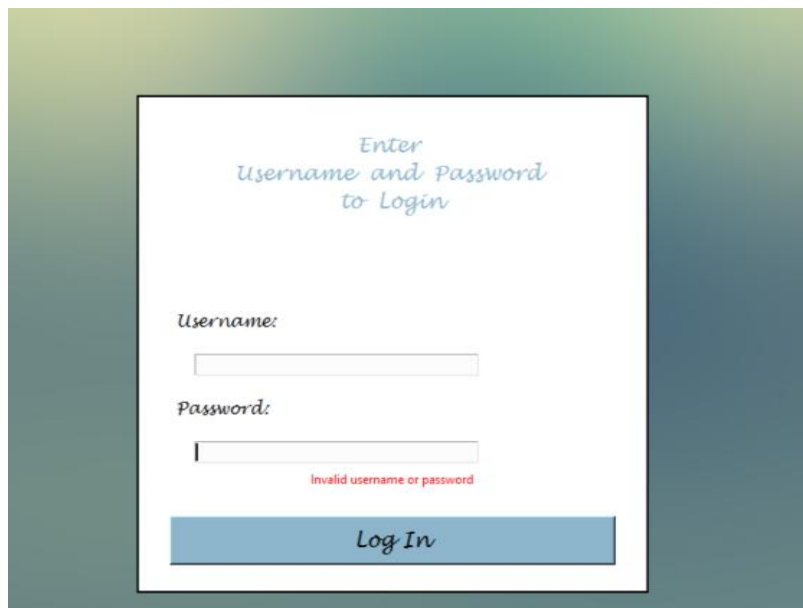
In this Login Window, you have to enter correct username and password to login. I have linked three usernames and passwords for login with database. Currently only these username and password will help you to login. I will definitely work for sign up where user can enter his/her own username and password and link it to database in my further version. Username and password linked with this application is as follows:

- Username1 - admin
Password1 - admin123
- Username2 - surbhi
Password2 - surbhi123
- Username3 - rollno
Password3 - ue168108

Clicking login with incomplete information will show error as follows:



Clicking login with incorrect information will clear both username and password and show error as follows:



Enter Username and Password to Login

Username:

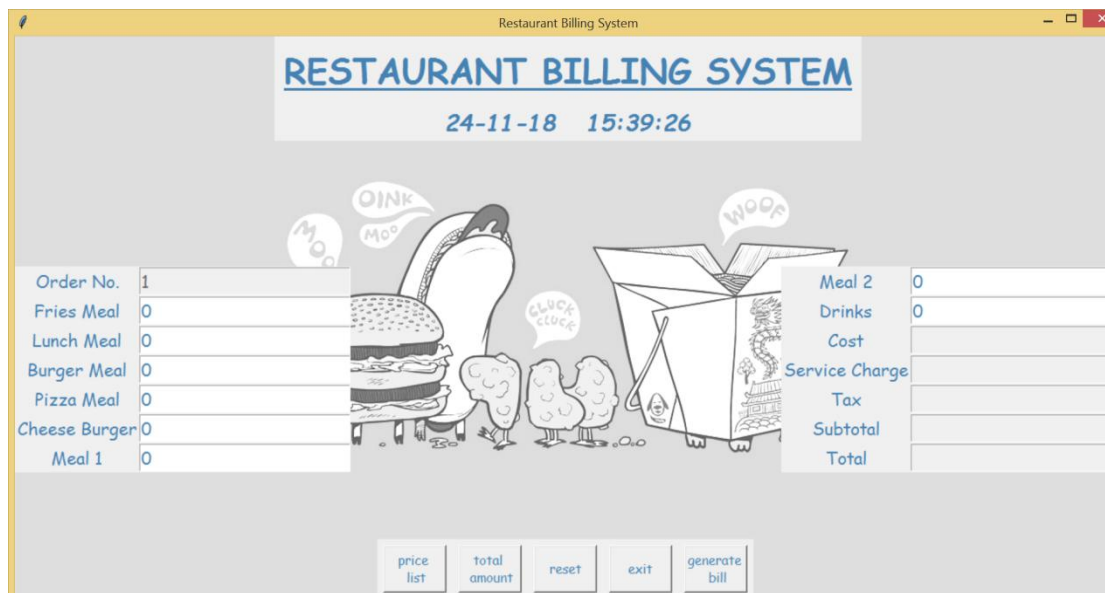
Password:

Invalid username or password

Log In

Clicking login with correct information will open a new interface.

3.3. Main Window



Restaurant Billing System

24-11-18 15:39:26

RESTAURANT BILLING SYSTEM

24-11-18 15:39:26

Order No. 1

Fries Meal 0

Lunch Meal 0

Burger Meal 0

Pizza Meal 0

Cheese Burger 0

Meal 1 0

Meal 2 0

Drinks 0

Cost

Service Charge

Tax

Subtotal

Total

price list total amount reset exit generate bill

This main window displays current system date and time on top. User has to enter quantities required for food item and click total amount button for getting total amount and simultaneously saving it in database with a pop-up message box.

3.4. Price List Window

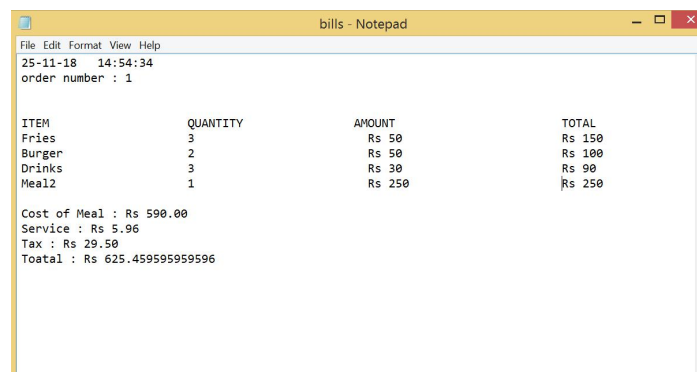


The screenshot shows a window titled "Price List" with a yellow border. Inside, the text "PRICE LIST" is underlined. Below it, there are two columns: "ITEMS" and "PRICE". The items listed are FRIES MEAL, LUNCH MEAL, BURGER MEAL, PIZZA MEAL, CHESSE BURGER, DRINKS, MEAL 1- FRIES MEAL+BURGER MEAL+DRINKS, and MEAL 2- PIZZA MEAL+CHESSE BURGER+DRINKS. The corresponding prices are Rs 50, Rs 150, Rs 50, Rs 200, Rs 70, Rs 30, Rs 200, and Rs 250.

ITEMS	PRICE
FRIES MEAL	Rs 50
LUNCH MEAL	Rs 150
BURGER MEAL	Rs 50
PIZZA MEAL	Rs 200
CHESSE BURGER	Rs 70
DRINKS	Rs 30
MEAL 1- FRIES MEAL+BURGER MEAL+DRINKS	Rs 200
MEAL 2- PIZZA MEAL+CHESSE BURGER+DRINKS	Rs 250

Clicking on price list button will open this price list window. This price list displays prices of menu offered. This program has a prespecified menu with their prices.

3.5. Generated Bill



The screenshot shows a Notepad window titled "bills - Notepad". The text inside is as follows:

```
25-11-18 14:54:34
order number : 1

ITEM          QUANTITY      AMOUNT      TOTAL
Fries         3              Rs 50       Rs 150
Burger        2              Rs 50       Rs 100
Drinks        3              Rs 30       Rs 90
Meal2         1              Rs 250      Rs 250

Cost of Meal : Rs 590.00
Service : Rs 5.96
Tax : Rs 29.50
Toatal : Rs 625.459595959596
```

Clicking on generate bill button in main window generates bill in text document.

3.6. Database Table

SQLite Studio - bills.db													
File Edit View Language Help													
Database Browser Browse Data Execute SQL													
Select Table: ORDERS													
Refresh New Record Delete Record Save Changes Search: Advanced Search													
ER_NUM	RIES_MEA	JNCH_MEA	JRGER_MEA	IZZA_MEA	EESE_BUR	DRINKS	MEAL_1	MEAL_2	COST	VICE_CHA	TAX	SUBTOTAL	TOTAL
1	1	100	0	0	200	0	60	0	360.00	3.64	18.00	360.00	381.6363...
2	2	0	0	50	0	0	60	200	0	310.00	3.13	15.50	310.00 328.6313...
3	3	100	0	0	0	280	120	0	500.00	5.05	25.00	500.00	530.0505...
4	4	0	0	0	0	70	0	0	70.00	0.71	3.50	70.00	74.20707...

Clicking on total amount button in main window will create a database table if not created in the same folder with “bills.db” name and extension. In this database created order number is primary key and will never be null and autoincrement after every insertion.

CHAPTER - 4

CONCLUSION

This training and project work made coding much easier than what it was before for me. Python is a programming language used almost everywhere to build the most common applications that we use today like, YouTube. I am able to create a computerized system for restaurant billing system. This system able to store billing records securely and access to them whenever required.

Problems faced by the existing system are:

- Inconsistency
- Low speed to data access
- Time wastage
- Poor performance
- Lack maintainability
- Non efficient representation of data.
- Requires a large storage space

The scope of my project in building a computerized system is handling restaurant records storing of restaurant records and enabling to view the records when desired. The employees are given limited access in the system in order to safe guard the privacy and security of the records.

Advantages of the new system are:

- Fast storage and retrieval of data.
- Increased accuracy of data.
- User friendly interfaces.
- Reduce human energy.
- Reduce time waste.
- Increased privacy due to security.
- Reduce storage space.
- Prevent data loss by backups stored in several locations.
- High efficiency through out the system.
- Increase the productivity of the company.

Requirement of the project

- Hardware requirement - Here is the recommended hardware requirement for this software to run efficiently.
 - 1) Intel core i3 or higher processor
 - 2) 10 MB RAM
 - 3) 15 MB free hard disc space
 - 4) SVGA monitors / Laptops
- Software requirement - This software comes under application software. So the necessary software for this is
 - 1) Windows operating system family.
 - 2) Python (version 3.0 or above)
 - 3) Python IDLE (Pycharm)
 - 4) Sqlite3 or any other database
 - 5) Sqlite Studio(to view database)

CHAPTER - 5

DRAWBACKS

Due to time constraint it is possible that some points might remain uncovered by me. In future I will update my software to give valuable information left at present. Though the system has been designed according to the requirements of the users it has its own limitation. Thus the limitation of system that I believe are:

- No facility to create new username and password.
- If project is already running, running it again in new window will create an error.
- If project is running on same window, order number is automatically incremented but order number will be initialized to 1 every time the project is executed .

However, this error is not generated in database i.e. In database orders are properly stored with their proper order number.

- After clicking on generate bill button in main window, .txt file opens. In this text file, changes can be done by the user while it must be disabled and bill is generated of only current order.

CHAPTER-6

REFERENCES

- <https://acadview.com/>
- <https://docs.python.org/3/>
- <https://stackoverflow.com/>
- <https://www.tutorialspoint.com/python3/>
- https://www.tutorialspoint.com/python/python_gui_programming.htm
- <https://www.geeksforgeeks.org/python-gui-tkinter>
- <https://docs.python.org/2/library/tkinter.html>