

# ENGINEERING EXPLORATION

Project Report

On

## CANTEEN BILL GENERATOR

By

B Karthikeya

160121733170

Ch Shiva shankar reddy

160121733171

K Mohan kumar

160121733179

P Dileep kumar

160121733191

Under the Mentorship of

**T.Prashanthi**

**Assistant professor**



Branch: **Department of Computer Science Engineering (C-3)**

**Submitted to**

Department of Computer Science and Engineering

**CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY (A)**

(Affiliated to Osmania University)

**Gandipet, Hyderabad- 500075**

**2021 - 2022**



## CERTIFICATE

This is to certify that the project entitled “**CANTEEN BILL GENERATOR**” by the following students has carried out under my mentorship.

B Karthikeya	160121733170
Ch Shiva shankar reddy	160121733171
K Mohan kumar	160121733179
P Dileep kumar	160121733191

**T.Prashanthi**

**ASSISTANT PROFESSOR**

**Department of Computer Science Engineering**

## **ACKNOWLEDGEMENT**

We would like to express our sincere thanks to T.Prashanthi (Mentor),for her valuable guidance and supporting in completing our project. Without her support and suggestions, this project would not have been completed.

We would also take the privilege to thank our management for giving us this opportunity and encouraging us to do such projects.

## **ABSTRACT**

As the technology is growing immensely in recent years, certain improvisations needs to be made. People are tired of making large calculation at the restaurant's bill counters and customers even got tired by waiting at bill counter. To end this and give a solution to restaurant's owners to keep track of their bills and amount, we made a bill generator using python language and oops concept.

We will be using python programming language and object oriented programming concept to generate a bill. we will be using a class which will be having different methods as different categories of food items. Each food category will be allotted a numeric code and each food item will also be allotted a numeric code. Based on the order of the customer bill counter boy will be entering the category code, food item code and quantity of food item. By using the inputs we will be generating a bill and printing it. In addition to that restaurant's owner can keep track of their total bills happened in a day and total amount generated in a day.

In this we will be using the inputs entered based on the customers order and by using the inputs we will be calculating the total amount and amount for each item. by making this project restaurants owners can fastly generate bill for their customers and provide good service to their customers.

## TABLE OF CONTENTS

ACKNOWLEDGEMENT	3
ABSTRACT	4
<b>CHAPTER 1 : INTRODUCTION</b>	
1.1 BACKGROUND	6
1.2 MOTIVATION	6
1.3 PROBLEM STATEMENT	7
1.4 FORMULATION	7
1.5 PROJECT OBJECTIVE	7
<b>CHAPTER 2 : CODE</b>	
2.1 bill_main.py	8
2.2 bill_maker.py	11
2.3 item_code_maker.py	15
<b>CHAPTER 3:RESULT AND DISCUSSIONS</b>	
3.1 CODE EXPLANATION	17
3.2 FINAL OUTPUT	18
<b><u>CHAPTER 4: FUTURE SCOPE</u></b>	20
<b><u>CHAPTER 5: CONCLUSION</u></b>	22
<b>REFERENCES</b>	23

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 BACKGROUND**

The taken project which is about creating a bill generator for a canteen was not the first thing that came to our minds when the EE projects was being discussed. There were many deliberations between us and we were against the projects which included creating a model or a work-piece as it would require bit more time and cannot be completed in the given time period. While n ideas were popping between us we suddenly got an idea of creating a bill generator. In the process we learn new languages and did experiment on them.

### **1.2 MOTIVATION**

The motivation required for this Engineering Exploration project was acquired from a scenario in our college canteen, the receptionist at our canteen used to issue tokens for the different types of food which used to take lot of time and there used to be a big queue in the canteen at lunch time. so in order to help our canteen owner and our beloved friends we started this project. numerous sources mainly being the idea of learning python and different modules in python which opened a gate of imagination in our brains. We can also open the doors of creativity inside us and explore as many options as possible. Our mentor T.prashanthi ma'am also motivated us constantly for exploring new areas in the field and guided us thoroughly.

### **1.3 PROBLEM STATEMENT**

Creating a bill generator program so that canteen owner can print bills and customize his bills easily. To convert analog billing system to digital billing system. To store every bill for future references and to keep track of their bills on daily basis.

### **1.4 FORMULATION**

We used object oriented programming language Python for creating this project. We created a class called as canteen and declared different class methods for different food categories. Every food category (class method) contains list of lists which contains different food items and their prices respective to the particular food category. We created two python files named as billmain.py and bill maker.py in which bill main file has the main implementation of the program which calls the class and its methods by referring to the different numeric characters entered by the user and prints the bill on the screen.

We imported different modules such as tabulate, pandas, datetime for the customization and representation of the bill on the screen And module named as CSV for storing bills in a CSV file for the future reference. We also handled every run time error by try and except concepts for user friendly ecosystem.

In addition to generating a bill, every bill generated will be stored in a CSV file which contains date, bill number and the items purchase for future reference and for the calculation of daily profits and losses. This CSV file can be shared to any one and the canteen owner can access it from anywhere from in this world so that he can keep track of his canteen without personally visiting it.

### **1.5 PROJECT OBJECTIVES**

- To create a program for bill generation of Canteen.
- To convert analogue billing system to digital billing system.
- To store every bill generated in a format for future reference and to keep track of their canteen revenue.

- **CHAPTER 2**

## The Code

## 2.1 bill\_main.py

```
import bill_maker
import csv
import importlib
importlib.reload(bill_maker)

def main() :

    while True :

        print(
            """          Enter 0 for MENU
                    Enter -1 to exit """)


        code = list(map(int,input('Enter the item code :').split('.')))
        if len(code)==1 and code[0]==-1 :
            print("\n','😊'*10,' THANK YOU ','😊'*10)
            print("\t\t\tHAVE A NICE DAY")
            break

        canteen= bill_maker.Canteen()
        if len(code)==1 and code[0]==0 :
            canteen.print_menu()
            continue

        elif len(code)%3!=0 :
            # print('entered code doesn\'t contain sufficient elements')
            print("\t\tINVALID INPUT")
            continue

        else :
```



```

count=len(code)//3
try :

    for i in range(count) :
        if code[3*i]==1 :
            canteen.breakfast(code[3*i+1],code[3*i+2])
        elif code[3*i+0]==2 :
            canteen.veg_starters(code[3*i+1],code[3*i+2])
        elif code[3*i+0]==3 :
            canteen.veg_biryani(code[3*i+1],code[3*i+2])
        elif code[3*i+0]==4 :
            canteen.non_veg_starters(code[3*i+1],code[3*i+2])
        elif code[3*i+0]==5 :
            canteen.non_veg_biryani(code[3*i+1],code[3*i+2])
        elif code[3*i+0]==6 :
            canteen.fast_food(code[3*i+1],code[3*i+2])

    except :
        print("\n\t\t-->COULD'NT FIND ENTERED ITEM")
        print('\t\t-->PLEASE TRY AGAIN\n')
        continue

canteen.print_bill()
canteen.save_data()
# canteen = None

if __name__ == "__main__":
    try :
        f = open("Data.csv",'r')

```

```
except :  
    f = open("Data.csv", 'x')  
    writer = csv.writer(f)  
    writer.writerow(["DATE", "BILL NO", "ITEMS", "TOTAL"])  
    writer.writerow([0,0,0,0])  
  
finally:  
    f = open("Data.csv", 'r')  
    reader = csv.reader(f)  
    for row in reader:  
        if len(row) > 1:  
            bill_maker.Canteen.bill_no = row[1]  
        else:  
            continue  
    f.close()  
main()
```

## 2.2 bill\_maker.py

```
from tabulate import tabulate
import datetime
import csv
class Canteen :
    bill_no = '0'
    def __init__(self):
        self.breakfast_menu = [
            ['Dosa-Masala', 25],
            ['Dosa-Onion', 25],
            ['Dosa-Plain', 25],
            ['Idly', 25],
            ['Poori', 30],
            ['Upma', 30],
            ['Utappa', 25],
            ['Vada', 30],
        ]
        self.veg_starters_menu = [
            ['Veg manchuria',149],
            ['Veg Spring Roll',149],
            ['Chilli Spring Roll',149],
            ['Mushroom 65',199],
            ['Paneer Manchuria',199],
        ]
        self.veg_biryani_menu =[
            ['Veg Biryani-Mini',179],
            ['Veg Biryani-Regular',219],
            ['Veg Biryani-Large',319],
            ['Paneer Biryani',319],
```

```

        ['Paneer Lollipop Biryani',339],
        ['Mushroom Biryani',339],
    ]
    self.non_veg_starters_menu = [
        ['Chicken 65',179],
        ['Chicken Manchuria',149],
        ['Chicken Spring Roll',209],
        ['Chile Fish',289],
        ['Appolo Fish',289],
        ['Prawns Fry',289],
    ]
    self.non_veg_biryani_menu = [
        ['Chicken Biryani-Mini',200],
        ['Chicken Biryani-Regular',379],
        ['Chicken Biryani-Large', 559],
        ['Fish Biryani',359],
        ['Paneer Biryani',359],
        ['Egg Biryani',159],
        ['Egg Masala Biryani',179],
    ]
    self.fast_food_menu = [
        ['Chicken Noodles-Single',70],
        ['Chicken Noodles-Full', 120],
        ['Egg Noodles-Single',60],
        ['Egg Noodles-Full',100],
    ]
    self.total = 0
    self.bill = []
    def breakfast(self,item_code,quantity):

self.bill.append([self.breakfast_menu[item_code][0],quantity,self.breakfast_menu[item_code]
[1]*quantity])
        self.total+=self.breakfast_menu[item_code][1]*quantity

```

```

def veg_starters(self,item_code,quantity):

self.bill.append([self.veg_starters_menu[item_code][0],quantity,self.veg_starters_menu[item_code][1]*quantity])
    self.total += self.veg_starters_menu[item_code][1]*quantity

def veg_biryani(self,item_code,quantity):

self.bill.append([self.veg_biryani_menu[item_code][0],quantity,self.veg_starters_menu[item_code][1]*quantity])
    self.total+=self.veg_starters_menu[item_code][1]*quantity

def non_veg_starters(self,item_code,quantity):

self.bill.append([self.non_veg_starters_menu[item_code][0],quantity,self.non_veg_starters_menu[item_code][1]*quantity])
    self.total+=self.non_veg_starters_menu[item_code][1]*quantity

def non_veg_biryani(self,item_code,quantity):

self.bill.append([self.non_veg_biryani_menu[item_code][0],quantity,self.non_veg_biryani_menu[item_code][1]*quantity])
    self.total+=self.non_veg_biryani_menu[item_code][1]*quantity

def fast_food(self,item_code,quantity):

self.bill.append([self.fast_food_menu[item_code][0],quantity,self.fast_food_menu[item_code][1]*quantity])
    self.total+=self.fast_food_menu[item_code][1]*quantity

def print_bill(self):
    Canteen.bill_no = int(Canteen.bill_no)+1
    print('\n')
    print('='*8,'\tCANTEEN NAME \t','=*8)
    print('\n',"DATE : ",datetime.date.today(),'\t',"BILL NO : ",Canteen.bill_no,'\n')
    print(tabulate(self.bill, headers=["Item", "Quantity", "Ammount"]))

```

```

print('_'*30)
print("TOTAL BILL : \t",self.total)
print("\n",'='*8,' 😊 THANK YOU 😊 ','='*8)
print('=' * 8, ' 😊 VISIT AGAIN 😊 ', '=' * 8, '\n', '\n')

def print_menu(self):
    print("\nBREAKFAST:")
    print(tabulate(self.breakfast_menu, headers=["Item", "Price"]))
    print("\nVEG STARTERS :")
    print(tabulate(self.veg_starters_menu, headers=["Item", "Price"]))
    print("\nVEG BIRYANI :")
    print(tabulate(self.veg_biryani_menu, headers=["Item", "Price"]))
    print("\nNON VEG STARTERS :")
    print(tabulate(self.non_veg_starters_menu, headers=["Item", "Price"]))
    print("\nNON VEG BIRYANI :")
    print(tabulate(self.non_veg_biryani_menu, headers=["Item", "Price"]))
    print("\nFAST FOOD :")
    print(tabulate(self.fast_food_menu, headers=["Item", "Price"]))

def save_data(self):
    with open("Data.csv", 'a') as file :
        writer = csv.writer(file)
        writer.writerow([datetime.date.today(),Canteen.bill_no,self.bill,self.total])

```

### 2.3 item\_code\_maker.py

```
import pandas as pd
import bill_maker
c= bill_maker.Canteen()
Breakfast = { 'Items' : [i[0] for i in c.breakfast_menu],
              'Price' : [i[1] for i in c.breakfast_menu],
              }
Veg_starters = { 'Items' : [i[0] for i in c.veg_starters_menu],
                 'Price' : [i[1] for i in c.veg_starters_menu],
                 }
Veg_biryani = { 'Items' : [i[0] for i in c.veg_biryani_menu],
                'Price' : [i[1] for i in c.veg_biryani_menu],
                }
Non_veg_starters = { 'Items' : [i[0] for i in c.non_veg_starters_menu],
                     'Price' : [i[1] for i in c.non_veg_starters_menu],
                     }
Non_veg_biryani = { 'Items' : [i[0] for i in c.non_veg_biryani_menu],
                    'Price' : [i[1] for i in c.non_veg_biryani_menu],
                    }
Fastfood = { 'Items' : [i[0] for i in c.fast_food_menu],
             'Price' : [i[1] for i in c.fast_food_menu],
             }

print("\n1 : BREAKFAST ")
print(pd.DataFrame(Breakfast))
print("\n2 : VEG STARTERS ")
print(pd.DataFrame(Veg_starters))
print("\n3 : VEG BIRYANI ")
print(pd.DataFrame(Veg_biryani))
print("\n4 : NON-VEG STARTERS ")
```

```
print(pd.DataFrame(Non_veg_starters))  
print("\n5 : NON-VEG BIRYANI ")  
print(pd.DataFrame(Non_veg_biryani))  
print("\n6 : FAST FOOD ")  
print(pd.DataFrame(Fastfood))
```



## **CHAPTER 3**

### **RESULT AND DISCUSSING:**

#### **3.1 CODE EXPLANATION**

The code for the main interface is written in the bill\_main.py file it first prompts you to enter the code which must be in the numeric characters separated by dots depending of the code that you enter it creates an object of the canteen class and navigates to different methods and generates the bill. If the code enter by the user is incorrect it will prompt an error to the user to enter a perfect code and continues the program it will not stop until you enter '-1' and you have to enter '0' for the menu. Every bill generated will be stored in a CSV file including date, bill number, items purchased and total bill.

In addition to these files we made a Python file named as item\_code\_maker.py for the reference of the client (canteen members) which imports module pandas and display the item codes and food category codes.

The canteen class contains different class methods such as food category for the development of the bill such as breakfast, veg starters, non veg starters,vg biryani, non veg biryani, fast food and cafeteria.

## 4.2 FINAL OUTPUT:

```
===== RESTART: C:\Users\karthikeya\OneDrive\Desktop\EE\Bill\bill_main.py =====
      Enter 0 for MENU
      Enter -1 to exit
Ente the item code :|
```

---

Fig 1: prompts to enter item code

## BILL DISPLAY:

```
=====          CANTEEN NAME          =====

  DATE :   2022-08-22          BILL NO :   21

Item                               Quantity          Ammount
-----
Idly                               1              25
Poori                             2              60
Paneer Manchuria                  1             199
Paneer Biryani                    2             398
Chicken 65                        2             358
Chicken Biryani-Regular           3            1137

TOTAL BILL :           2177

=====  😊THANK YOU😊  =====
=====  😊VISIT AGAIN😊 =====
```

---

Fig 2 : final bill display

DATE	BILL NO	ITEMS	TOTAL
0	0	0	0
,75	3	597]]"	597
07-08-2022	3	['Mushroom 65', 1, 199]]	199
07-08-2022	4	['Mushroom 65', 1, 199]]	199
07-08-2022	5	['Mushroom 65', 4, 796]]	796
07-08-2022	6	['Veg Biryani-Large', 4, 596]]	596
07-08-2022	7	['Chicken Biryani-Large', 2, 1118]]	1118
07-08-2022	8	['Poori', 2, 60]]	60
07-08-2022	9	['Dosa-Plain', 3, 75]]	75
07-08-2022	10	['Dosa-Plain', 3, 75]]	75
07-08-2022	11	['Dosa-Plain', 3, 75]]	75
07-08-2022	12	['Idly', 5, 125]]	125
07-08-2022	13	['Chile Fish', 2, 578], ['Veg Biryani-Regular', 2, 298]]	876

Fig 3 : CSV File

## **CHAPTER 4:**

### **FUTURE SCOPE**

In the current 5G technology people no more have patience to stand in a line to place their order or to shout for their orders in a crowded restaurant. Furthermore, the bill generating system for restaurants is a trend that is growing at a fast pace. Major Restaurants today just for the convenience factor and increasing revenue are converting to computerized bill generator.

#### **Why do people prefer bill generator?**

As we all know that food is the basic need in every human life, for which everyone is struggling. Yet even after their efforts, if people are still not happy with their sustenance, then the effort given behind it is completely useless. The reasons might be the people are occupied with their busy schedules, they either can't able to cook or not having time to make food and order their food from outside.

#### **What are the benefits of bill generator?**



1. The food ordering process easier for customers as well as for restaurant owners.
2. Easy order management
3. Less processing time means less waiting time for food orders.
4. Live order tracking.
5. It is very easy to customize the food order.

### **Reason for growth of the food industry**

#### **Scarcity of time**

The young professionals of today feel the desire to do many things. So, in order to do these activities, they decide it better to eat cooked food than cooking it themselves. But, finding a restaurant, ordering food while standing in a queue and eating there takes time. To cut short this time these we created computerized bill generator.

#### **Change of preference**

The people today are exposed to a number of cuisines and no longer restrict themselves to their native cuisine. For example, Indians have started moving from Indian and Chinese cuisine to other cuisines like Italian and Mexican

#### **Easier to get food**

Getting ready-made food is now easier than it was in the past. In fact, food outlets are providing people with meals throughout the day based on the monthly subscription. This arrangement is well suited for the young professionals of today.

## **CHAPTER 5**

### **CONCLUSIONS**

Finally, we have created a computerized bill generator which was better than analog billing system at the beginning. Customers can see the menu, select the food items, order them and can also reserve a table in advance. They can contact the restaurant in case of any issues from the details provided in the bill. At the same time, having a bills reduces reduces time in calculating the revenue and very helpful in reaching out to the customers .

Therefore, it is useful to have a computerized bill generator.

## **REFERENCES**

1. [https://www.youtube.com/watch?v=Edsxf\\_NBFrw](https://www.youtube.com/watch?v=Edsxf_NBFrw)
2. <https://www.youtube.com/watch?v=BsDoLVMnmZs>
3. [https://www.youtube.com/watch?v=B20Getj\\_Zk4](https://www.youtube.com/watch?v=B20Getj_Zk4)
4. [https://drive.google.com/file/d/1XurNb2UThGAn261nwd1SY07Mc\\_jC1h7M/view](https://drive.google.com/file/d/1XurNb2UThGAn261nwd1SY07Mc_jC1h7M/view)
5. <https://en.wikipedia.org/wiki/HTML>