VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi-560018

File Structures

Mini Project Report

On

" JST RETAIL SHOP "

Submitted in partial fulfilment of the requirement of VI semester File Structures Laboratory

Submitted by,

SIDDHARTH SHUKLA	**1DT19IS166**
TEJASWINI MOHANTY	**1DT19IS170**

Under the guidance of

Mrs. Supriya R.K
Asst. Professor
Dept. of ISE
DSATM, Bangalore.

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

DAYANANDA SAGAR ACADEMY OF TECHNOLOGY AND MANAGEMENT
(Affiliated to Visvesvaraya Technological University, Belagavi & Approved by AICTE, New Delhi)
(Accredited by NBA, New Delhi)
Opp. Art of Living, Udayapura, Kanakapura Road, Bangalore- 560082

2021-22

DAYANANDA SAGAR ACADEMY OF TECHNOLOGY AND MANAGEMENT

(Affiliated to Visvesvaraya Technological University, Belagavi & Approved by AICTE, New Delhi)
(Accredited by NBA, New Delhi)
Opp. Art of Living, Udayapura, Kanakapura Road, Bangalore- 560082

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

Certificate

This is to certify that the mini-project work entitled **"JST RETAIL SHOP"** is carried out by SIDDHARTH SHUKLA (1DT19IS166) and TEJASWINI MOHANTY (1DT19IS170) in partial fulfilment of the requirement of VI semester File Structures Laboratory in **Information Science and Engineering** of the **Visvesvaraya Technological University, Belagavi** during the year 2021-2022. It is certified that all the corrections/ suggestions indicated for the given internal assessment have been incorporated in the report. This report has been approved as it satisfies the academic requirements with respect to the mini-project work.

Signature of the Guide

Mrs. Supriya R.K
Asst. Professor,
Dept. of ISE
DSATM, Bangalore.

Signature of the HOD

Dr. Sumithra Devi K A
Dean Academics, Professor & Head
Dept. of ISE
DSATM, Bangalore.

External Viva

Name of the Examiners **Signature with date**

1.

2.

ABSTRACT

This will estimate the total invoice at the expected discount rate after all product components have been selected. Finally, the system generates an invoice receipt in .txt format that includes the customer's name, date and time of purchase, quantity of product items, unit price and total, discountable amount, and payment amount. The billing system helps automate time-consuming tasks such as billing, product tracking, and other accounting documents. In addition, such software automatically calculates and applies taxes on taxable products, so you don't have to worry about every charges. This software helps sellers manage different types of records for their customers. This product helps users work in a highly effective and efficient environment. It consumes a lot of time and energy that can be used more effectively Productive activity. Apart from that, as the customer's power grows, management tasks Notifying individual customers is certainly a tedious task. In a manual system There are many inefficiencies faced by sellers. Information retrieval Foreground problem.

TABLE OF CONTENTS

ACKNOWLEDGEMENT

The completion of the project brings a sense of satisfaction, but it is never complete without thanking the persons responsible for its successful completion.

We express our profound gratitude to **Dr. M. Ravishankar, Principal, DSATM, Bangalore,** for providing the necessary facilities and an ambient environment to work.

We are grateful to **Dr. Sumithra Devi K A, Dean Academics, Professor & Head, Department of Information Science and Engineering, DSATM, Bangalore,** for her valuable suggestions and advice throughout our work period.

We would like to express our deepest gratitude and sincere thanks to our guide **Mrs. Supriya R K, Assistant Professor, Department of Information Science and Engineering, DSATM, Bangalore,** for her keen interest and encouragement in the project whose guidance made the project into reality.

We would like to thank all the staff members of the **Department of Information Science and Engineering** for their support and encouragement during the course of this project.

Definitely most, we would like to thank our fellow senior Mr. Jaisal Srivastava for his much-needed expert guidance to help us walk through every phase of this project and help make it a success. Without his help and guidance this assessment would have been impossible to complete.

TEJASWINI MOHANTY
(1DT19IS170)
SIDDHARTH SHUKLA
(1DT19IS166).

CHAPTER 1

INTRODUCTION

The **JST RETAIL SHOP** is primarily concerned with dealing with customer bill, including purchase quantity and discount. The system allows the user to first enter the customer's name, after which the user can simply enter the item's name and quantity desired.

This estimates the total due bill with the predicted discount percentage after selecting all of the product components. Finally, the system generates an invoice receipt in.txt format that includes the customer's name, date, and time of purchase, product item quantity, unit price, and total, discountable amount, and payment amount.

Billing systems aid in the automation of time-consuming tasks such as invoice generation, product tracking, and other accounting documents, among others. Furthermore, such software calculates and applies tax on taxable products automatically, so you don't have to worry about it every time you bill.

This software will help salespersons in managing the various types of records pertaining to customer. The product will help the user to work in a highly effective and efficient environment. It consumes the considerable time and energy that could be utilized in the better productive activities. Apart From that, with increasing customer strength, the task of managing information of each individual customer is indeed a cumbersome task. In The manual system, there are number of inefficiencies that a salesperson faces. The Information retrieval is one of the foremost problems.

CHAPTER 2

REQUIREMENT ANALYSIS

The requirement analysis specifies the requirements needed to develop a graphic project. In this phase, we collect the requirements needed for designing the project. The requirements collected are then analyzed and carried to the next phase.

Software requirements:

Operating System: windows 7 or above

Language used: Python 3.10 or above

Software used: Visual Studio Code

Hardware requirements:

Processor: Intel ® Core™ i3-2370 CPU @2.40GHz or above

RAM: 8 GB or more

Hard disk: 10 GB or more

CHAPTER 3

DESIGN

Use Case Diagram:

The below UML diagram depicts that the User can have a access to all the operations of the bill of the Retail Shop.

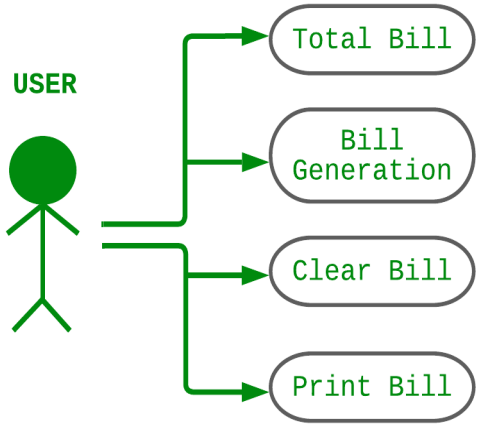


Fig 3.1: Use Case Diagram of JST retail shop.

A use case diagram is a dynamic or behavior diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system need stopper form.

A use case diagram of JST retail shop shown above shows that the user can take the total of bill, generate the bill, clear the bill and also print the bill.

CHAPTER 4

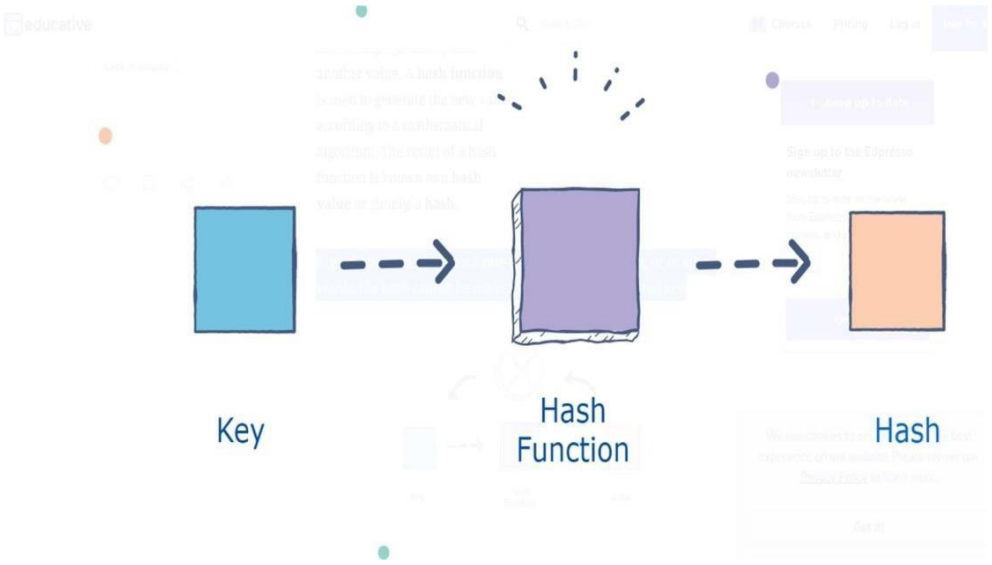
IMPLEMENTATION

4.1 INTRODUCTION TO PYTHON AND TKINTER:

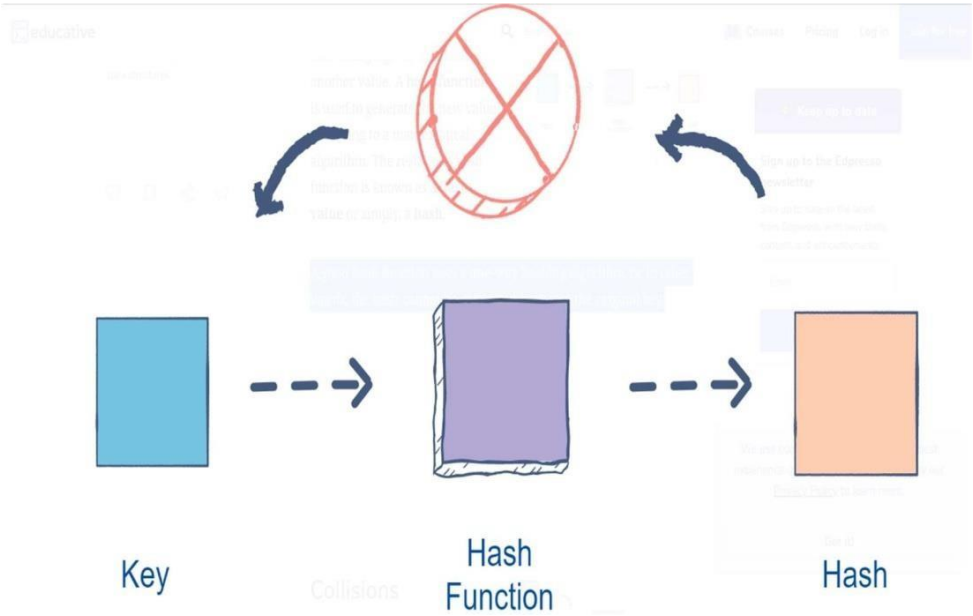
Python and Tkinter Programming presents the elements of typical Python and Tkinter applications in a straight-forward fashion. Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms. The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python Web site, https://www.python.org/, and may be freely distributed. The same site also contains distributions of and pointers to many free third party Python modules, programs and tools, and additional documentation. The Python interpreter is easily extended with new functions and data types implemented in C or C++ (or other languages callable from C). Python is also suitable as an extension language for customizable applications. Sample code illustrates each element. Complete applications that go far beyond the fill-the-form class of graphical user interfaces are presented; here you will find examples of complex controls, drawn interfaces and photorealistic panels. The code can readily be used as templates for new applications. Extensions to Python (such as ODBC) are examined as well. 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. Tkinter provides various controls, such as buttons, labels and text boxes used in a GUI application. These controls are commonly called widgets. Tkinter is the most commonly used library for developing GUI (Graphical User Interface) in Python. It is a standard Python interface to the Tk GUI toolkit shipped with Python. As Tk and Tkinter are available on most of the Unix platforms as well as on the Windows system, developing GUI applications with Tkinter becomes the fastest and easiest.

4.2 FILE STRUCTURECONCEPTUSED: HASHING

Hashing is the process of converting a given key into another value. A hash function is used to generate the new value according to a mathematical algorithm. The result of a hash function is known as a hash value or simply a hash.

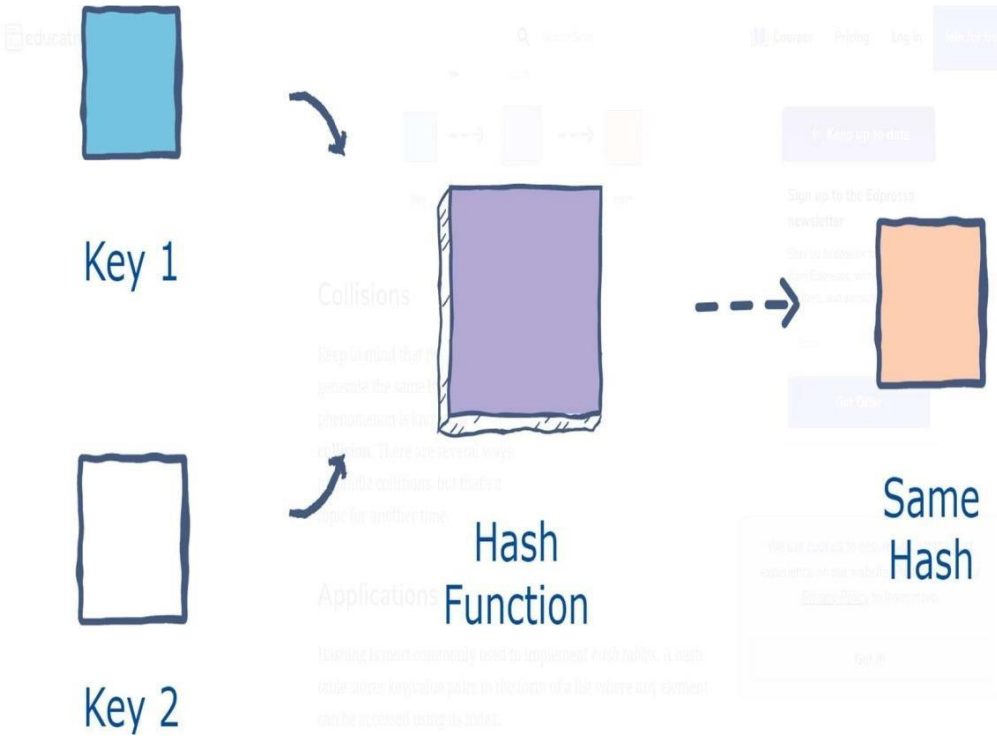


A good hash function uses a one-way hashing algorithm, or in other words, the hash cannot be converted back into the original key.



Collisions

Keep in mind that two keys can generate the same hash. This phenomenon is known as a collision. There are several ways to handle collisions, but that's a topic for another time.



Applications:

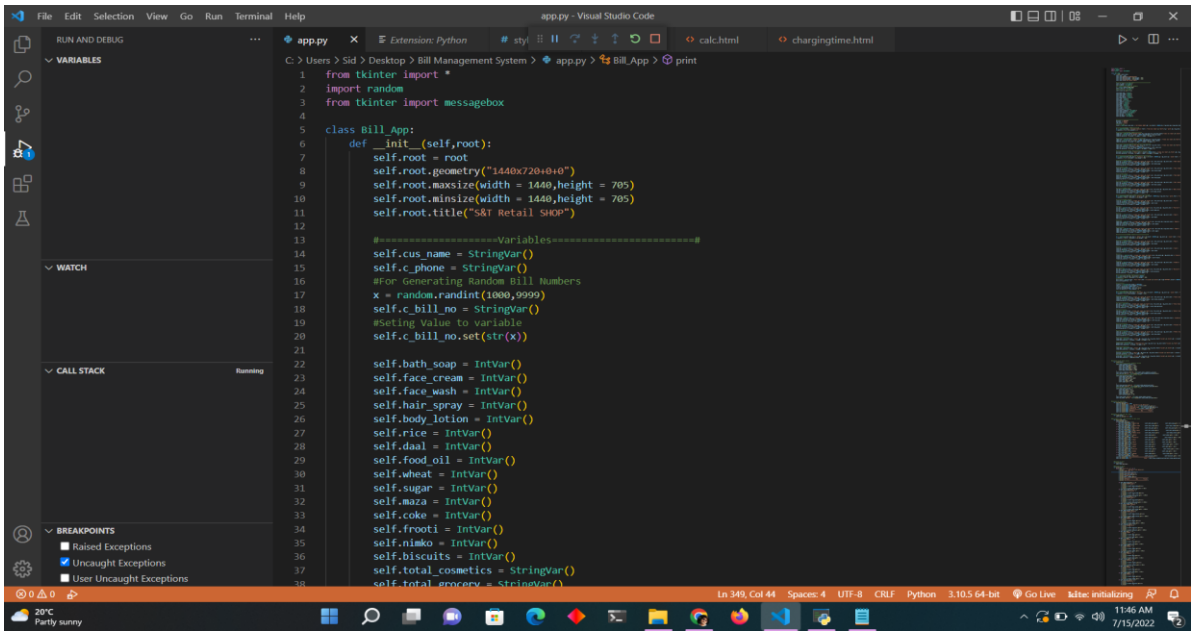
Hashing is most commonly used to implement hash tables. A hash table stores key/value pairs in theform of a list where any element can be accessed using its index. Since there is no limit to thenumber of key/value pairs, we can use a hash function to map the keys to the size of the table;the hash value becomes the index for a given element. A simple hashing approach would be totake the modular of the key (assuming it's numerical) against the table's size:

Index=key MOD table Size

This will ensure that the hash is always within the limits of the table size.

4.3 MODULES

1. Bill app declarations

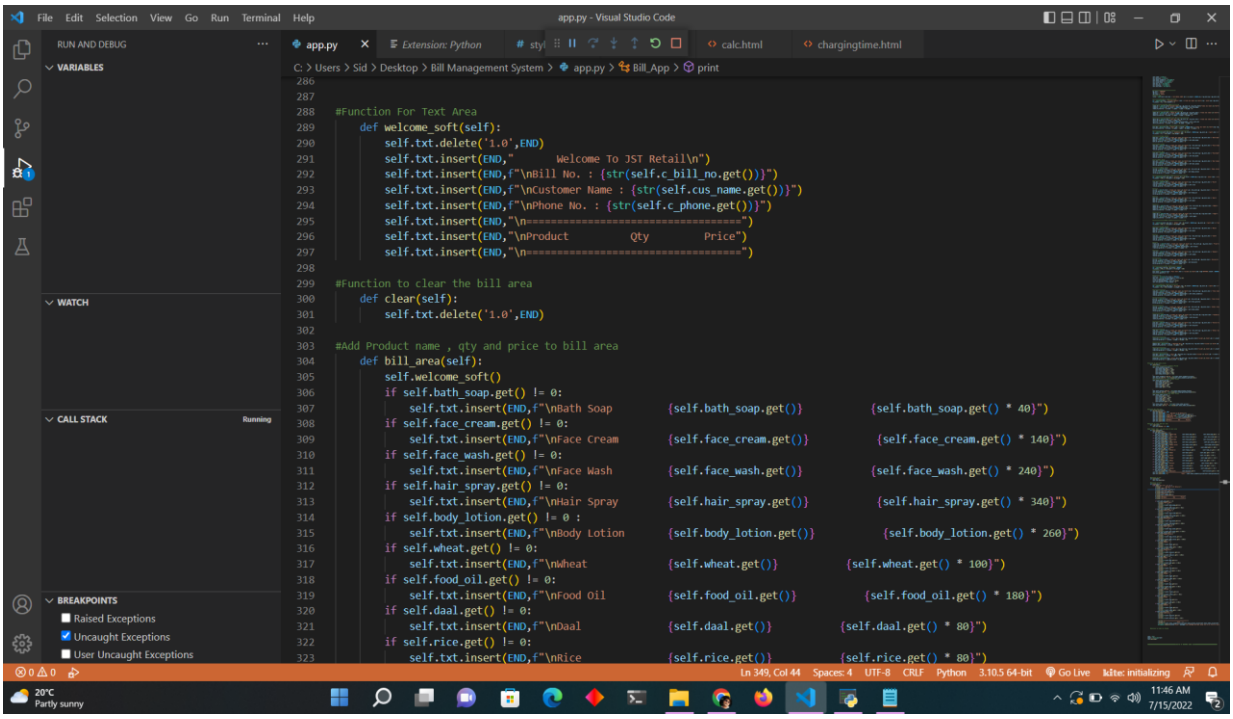


```
from tkinter import *
import random
from tkinter import messagebox

class Bill_App:
    def __init__(self,root):
        self.root = root
        self.root.geometry("1440x720+0+0")
        self.root.maxsize(width = 1440,height = 705)
        self.root.minsize(width = 1440,height = 705)
        self.root.title("S&T Retail SHOP")

        #=================Variables=====================#
        self.cus_name = StringVar()
        self.c_phone = StringVar()
        #For Generating Random Bill Numbers
        x = random.randint(1000,9999)
        self.c_bill_no = StringVar()
        #Seting Value to variable
        self.c_bill_no.set(str(x))

        self.bath_soap = IntVar()
        self.face_cream = IntVar()
        self.face_wash = IntVar()
        self.hair_spray = IntVar()
        self.body_lotion = IntVar()
        self.rice = IntVar()
        self.daal = IntVar()
        self.food_oil = IntVar()
        self.wheat = IntVar()
        self.sugar = IntVar()
        self.maza = IntVar()
        self.coke = IntVar()
        self.frooti = IntVar()
        self.nimko = IntVar()
        self.biscuits = IntVar()
        self.total_cosmetics = StringVar()
        self.total_grocery = StringVar()
```

Fig 4.3.1: "Bill App Declarations"

The initial declarations of variables and parameters are given in the initial part of the Coding process.

2. Functions of text area and bill area:



```
#Function For Text Area
    def welcome_soft(self):
        self.txt.delete('1.0',END)
        self.txt.insert(END,"       Welcome To JST Retail\n")
        self.txt.insert(END,f"\nBill No. : {str(self.c_bill_no.get())}")
        self.txt.insert(END,f"\nCustomer Name : {str(self.cus_name.get())}")
        self.txt.insert(END,f"\nPhone No. : {str(self.c_phone.get())}")
        self.txt.insert(END,"\n================================")
        self.txt.insert(END,"\nProduct          Qty          Price")
        self.txt.insert(END,"\n================================")

#Function to clear the bill area
    def clear(self):
        self.txt.delete('1.0',END)

#Add Product name , qty and price to bill area
    def bill_area(self):
        self.welcome_soft()
        if self.bath_soap.get() != 0:
            self.txt.insert(END,f"\nBath Soap        {self.bath_soap.get()}          {self.bath_soap.get() * 40}")
        if self.face_cream.get() != 0:
            self.txt.insert(END,f"\nFace Cream       {self.face_cream.get()}          {self.face_cream.get() * 140}")
        if self.face_wash.get() != 0:
            self.txt.insert(END,f"\nFace Wash        {self.face_wash.get()}          {self.face_wash.get() * 240}")
        if self.hair_spray.get() != 0:
            self.txt.insert(END,f"\nHair Spray       {self.hair_spray.get()}          {self.hair_spray.get() * 340}")
        if self.body_lotion.get() != 0 :
            self.txt.insert(END,f"\nBody Lotion      {self.body_lotion.get()}          {self.body_lotion.get() * 260}")
        if self.wheat.get() != 0:
            self.txt.insert(END,f"\nWheat            {self.wheat.get()}          {self.wheat.get() * 100}")
        if self.food_oil.get() != 0:
            self.txt.insert(END,f"\nFood Oil         {self.food_oil.get()}          {self.food_oil.get() * 180}")
        if self.daal.get() != 0:
            self.txt.insert(END,f"\nDaal             {self.daal.get()}          {self.daal.get() * 80}")
        if self.rice.get() != 0:
            self.txt.insert(END,f"\nRice             {self.rice.get()}          {self.rice.get() * 80}")
```

Fig 4.3.2: "Text area and the Bill area"

This image shows the functions of the text area and the bill area tab along with the updating function of the bill.

3. Function for Total price:

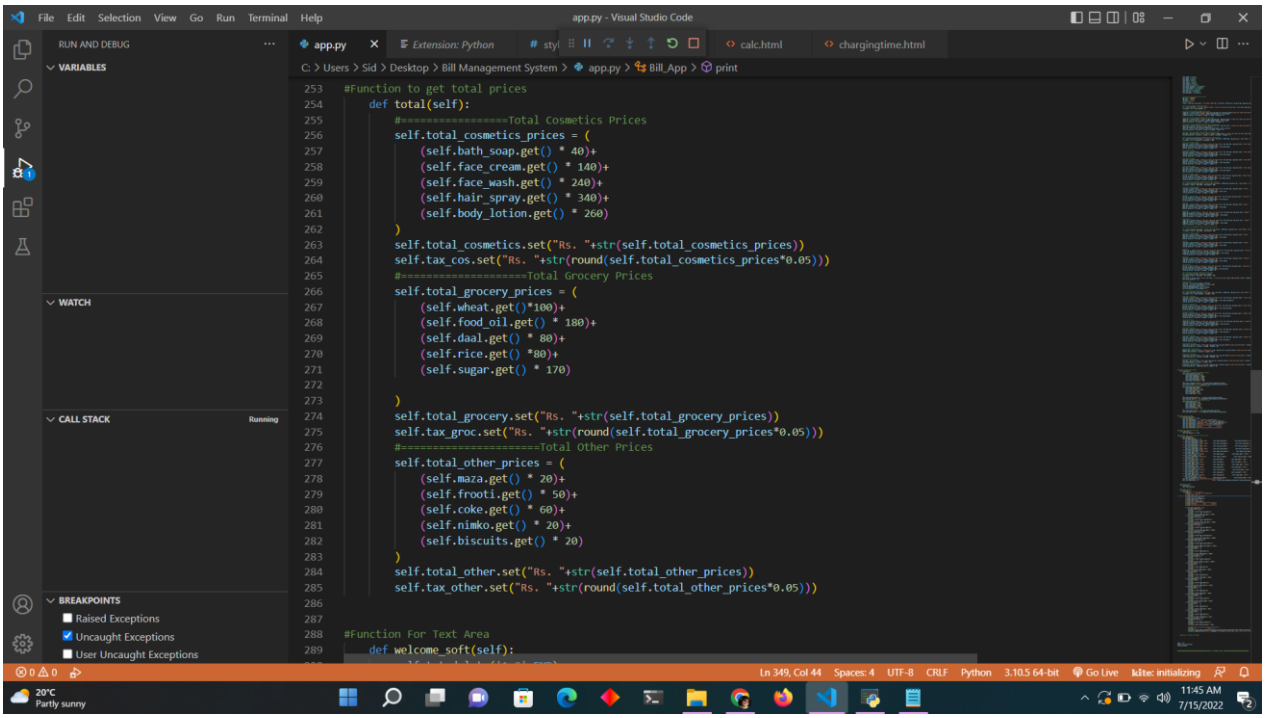


Fig 4.3.3: "Total price of bill"

This tab shows the total price of all the items that is being added by the user in the list for the purpose of buying.

4. Printing function of Bill:

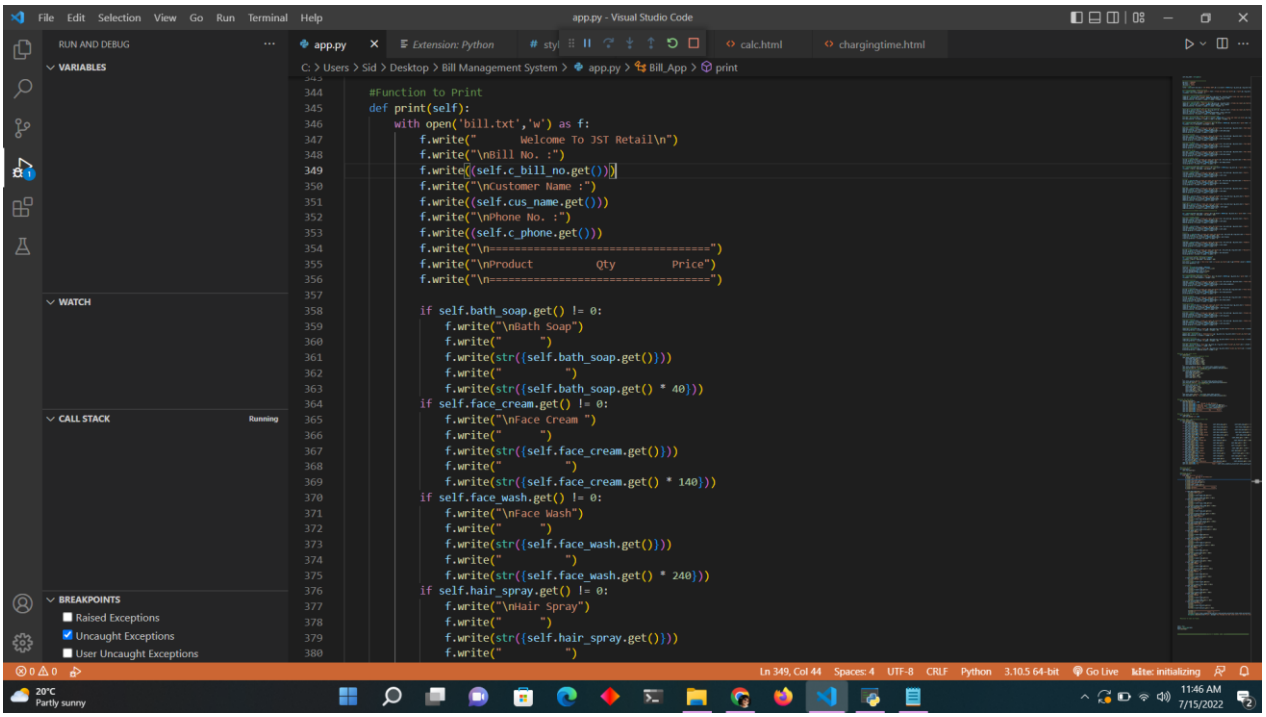


Fig 4.3.4:"Printing of bill"

CHAPTER 5

SNAPSHOTS

Fig 5.1: "Home Page"

It is the initial page of the project that the user can view.

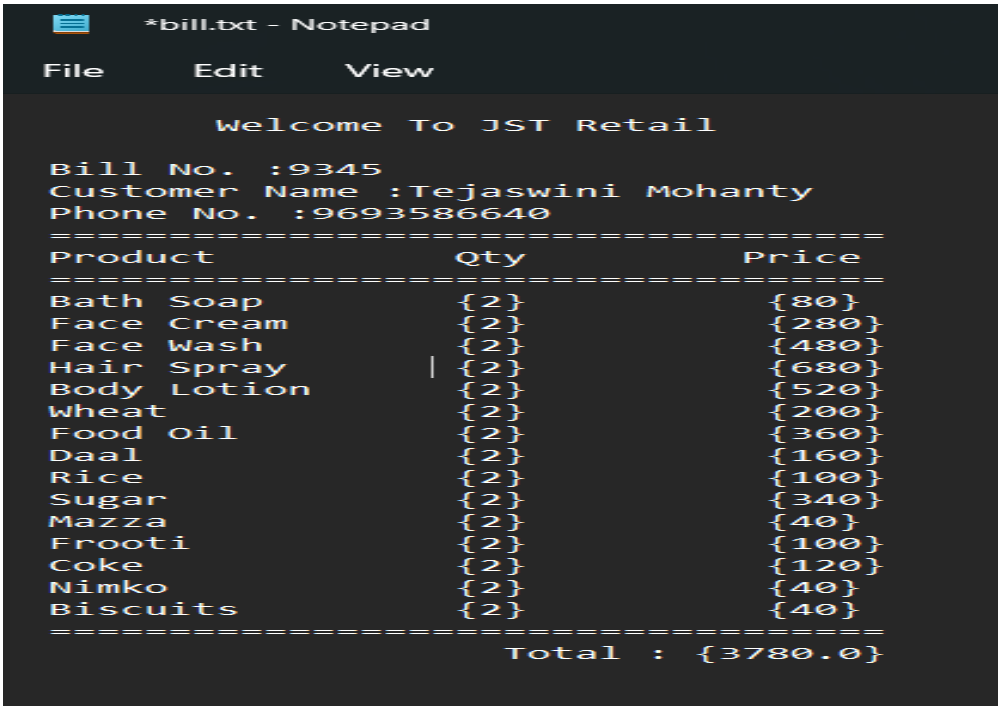


Fig 5.2: "Bill Layout"

It is the billing page of the project that the user will get after adding the quantity of items.

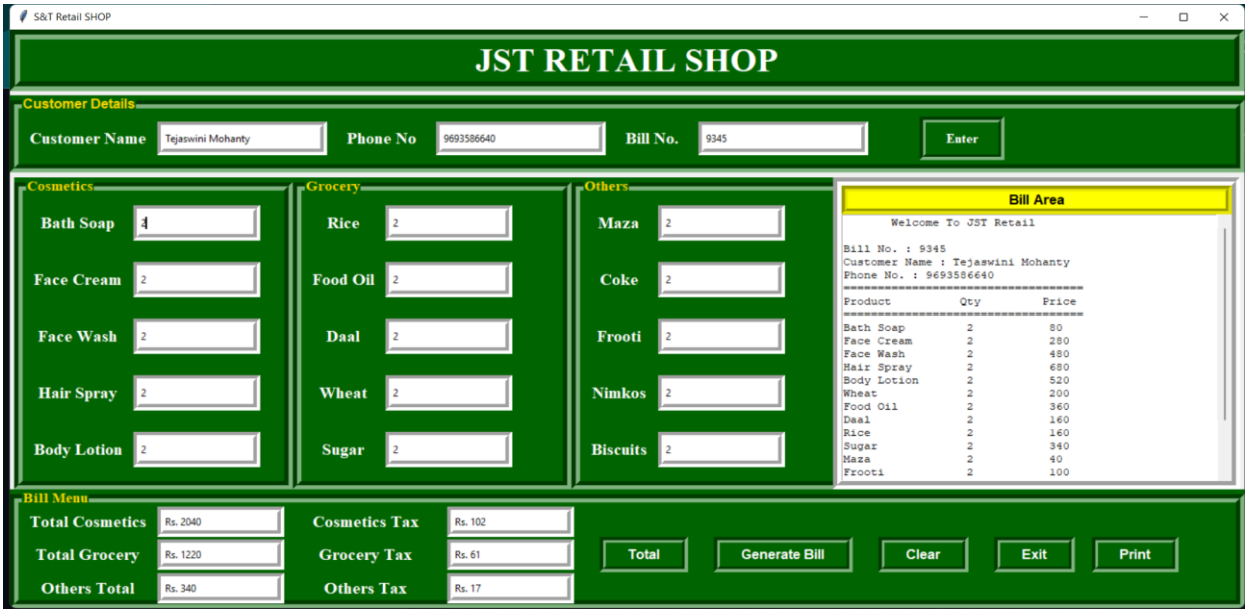


Fig 5.3: "Updating Detail"

It is the initial page of the project that the user can update the items.

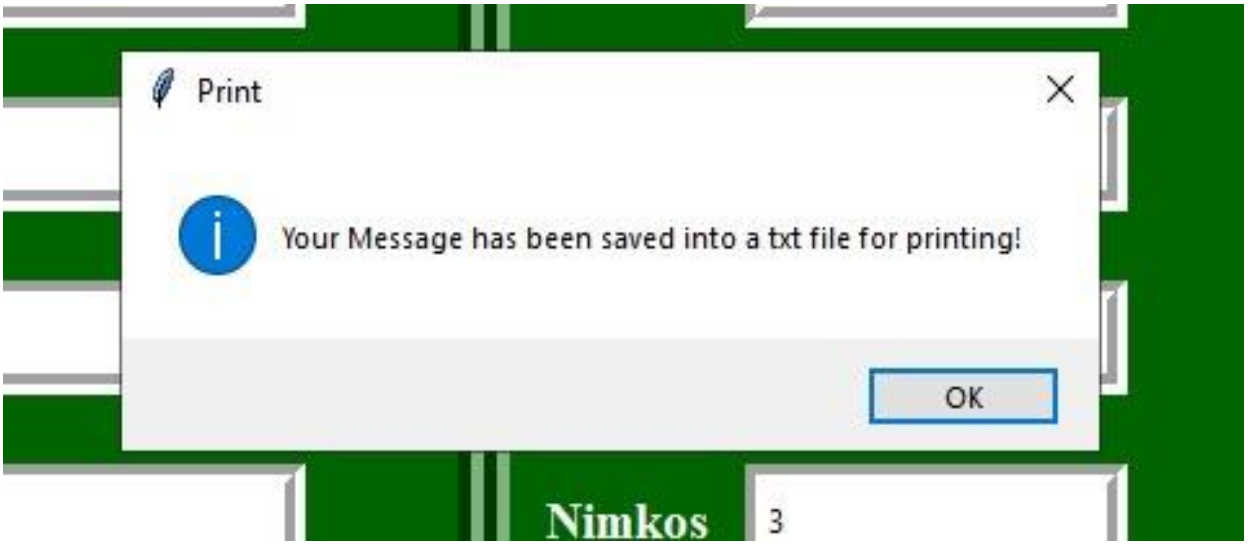


Fig 5.4: "TXT file"

This is the TXT file that gets generated when we execute the project.

CONCLUSION AND FUTURE ENHANCEMENT

After we have completed the project we are sure the problems in the existing system world overcome. The “JST RETAIL SHOP” process made computerized to reduce human errors & to increase the efficiency.

The main focus of this project is to less human efforts. The maintenance of the records is made efficient, as all the records are stored in the Access File, through which data can be retrieved easily. The editing is also made simples. The user has to just type in the required field & process the update button to update the desired field. The user are given a particular unique id no. so that they can be access correctly & without errors. Our main aim of the project is to get the correct information about the bill.

In the future, we would like to improve the following:

- Use more improved GUI for the same implementations.
- Use better file structures concepts for the betterment of the applications.
- Use latest technologies to make the project look impressive and attractive.
- Can add SignUp and LogIn for more security of users.
- Can add search for easy searching of products.

REFERENCES

1. The complete reference - Python GUI Programming with Tkinter by Alan D. Moore

2. Python Programming by Nicholas Ayden

3. Modern Tkinter for Busy Python Developers by Mark Roseman

4. Wikipedia:

- https://en.wikipedia.org/wiki/Tkinter

5. Google.

- https://Phone-Book/blob/master/phonebook.py