**PYTHON FINAL PROJECT**

**Smart Mart Application System**

**User Manual**

**Submitted By:**

**Harkaran Singh(C0768337)**

**Dimple Verma(C0777178)**

**Anand Hirulkar(C0772915)**

**Chippy Baby(C0777183)**

**Kiranpreet Kaur(C0771720)**

**INTRODUCTION**

Smart-Mart is a convenient store that is located in Toronto, Canada. This application is developed for the purpose of keeping the track of the inventory items and managing the billing process of the Smart Mart Store. The purpose of the application is to ease the process of keeping the record of the items sold and generating the required reports of the items in stock or sold.

**Hardware Requirements:**

The application used tkinter module of python to develop the Graphical User Interface for the user and MySQL database for saving the data in tables. Below are the hardware requirements for running the application:

Python/Tkinter:

**Minimum System Requirements**

* Processors: Intel Atom® processor or Intel® Core™ i3 processor
* Disk space: 1 GB
* Operating systems: Windows\* 7 or later, macOS, and Linux
* Python\* versions: 2.7.X, 3.6.X

MySQL:

**Minimum Hardware Requirements**

* 2 CPU Cores
* 2 GB RAM
* Disk I/O subsystem applicable to a write-intensive database

**Software Requirements:**

The application uses the below software programs:

Python 3.5 or later (earlier versions not supported on windows XP or earlier)

MySQL Database software with additional set of requirements mentioned below:

* Microsoft .NET Framework 4.5.2
* Microsoft Visual C++ Redistributable for Visual Studio 2019
* Microsoft Windows 10 or Windows Server 2019

**Each Python File Description/Usage of Application:**

There are total of 7 python files used in this project. Each python files serve a specific purpose. The project is divided into multiple files to serve the purpose of easy understanding for the user. Below is list of the files available in the python project:

**dbconnext.txt:**

This file contains the information about the database connection.

The file contains the information as

hostname=localhost

username=root

password=password

database=smartmart

A new user using the application is not required to provide manually in each file the value. Instead this alone file can be updated to connect to database.

The first file with which the application starts is **login.py**

**login.py File :**

The login.py file allows the user to login to the application. The username and password is saved in database. There are 2 default usernames and passwords as mentioned below:

Username: username

Password: username

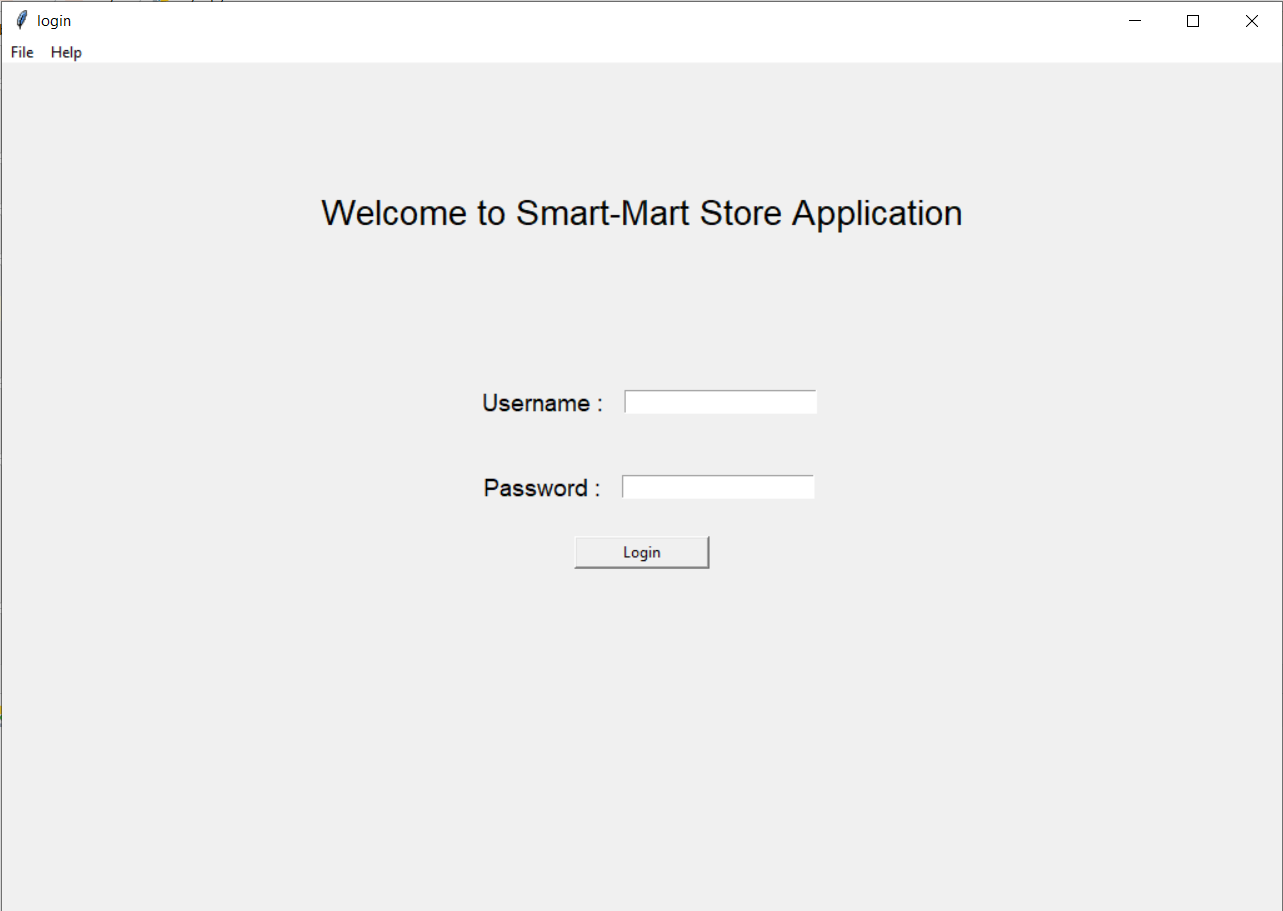
OR

Username: login

Password: login

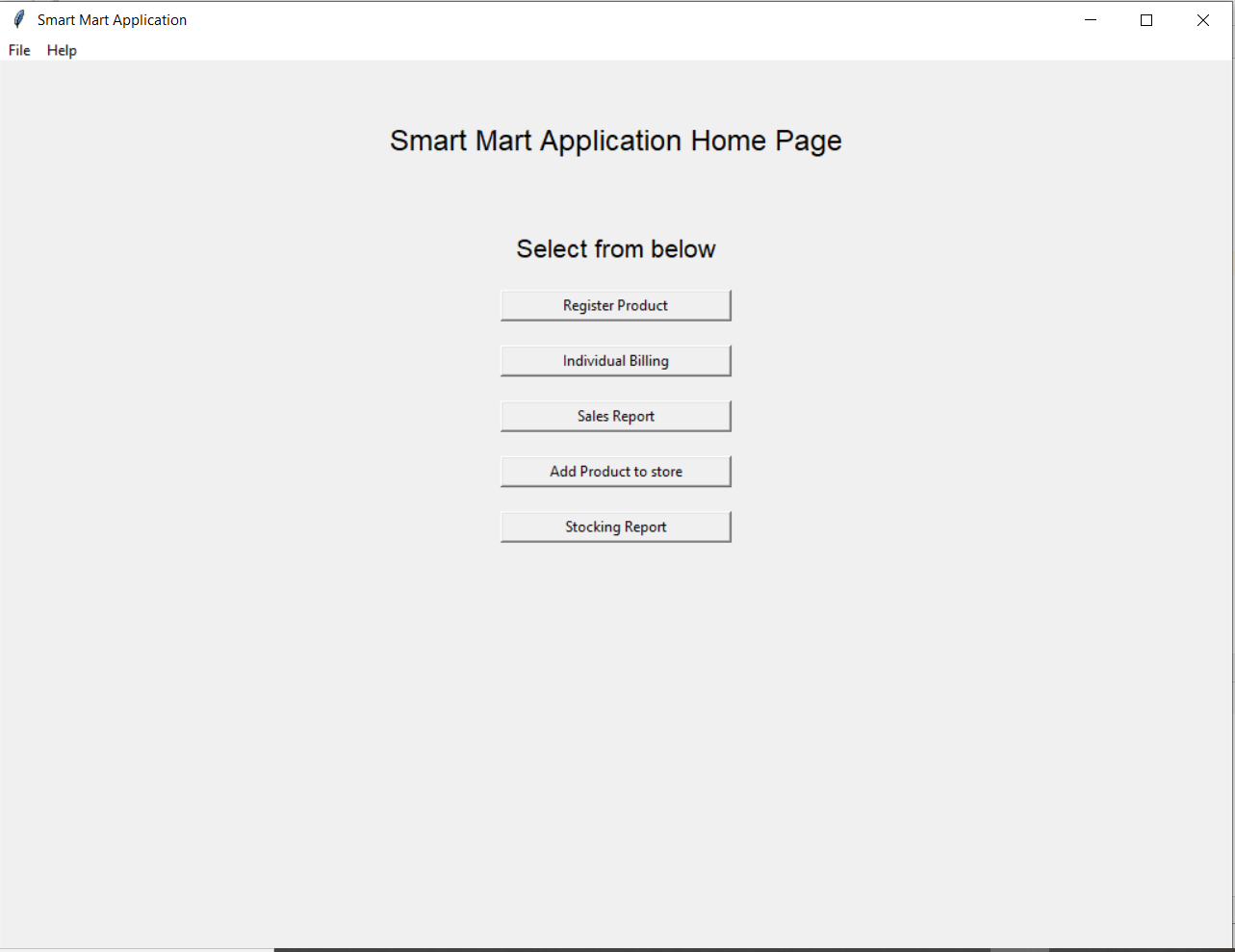
This page allows the user to enter into the application on successful login.

Login Page :



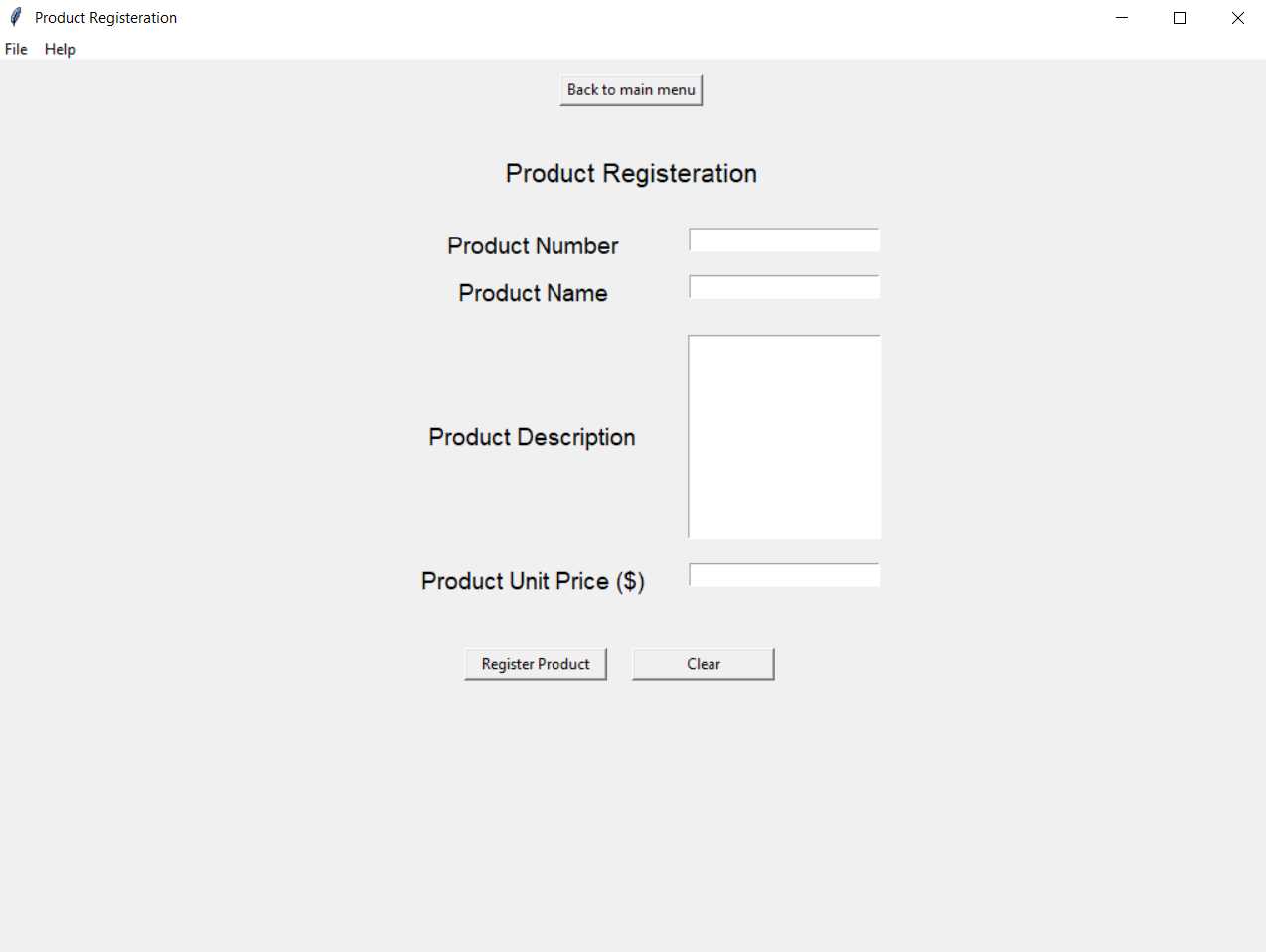
**Smartmart.py:**

The smartmart.py file serves the purpose of providing the user with multiple buttons to choose from the operation the user want to perform. The page is the centre link between all of the application pages.



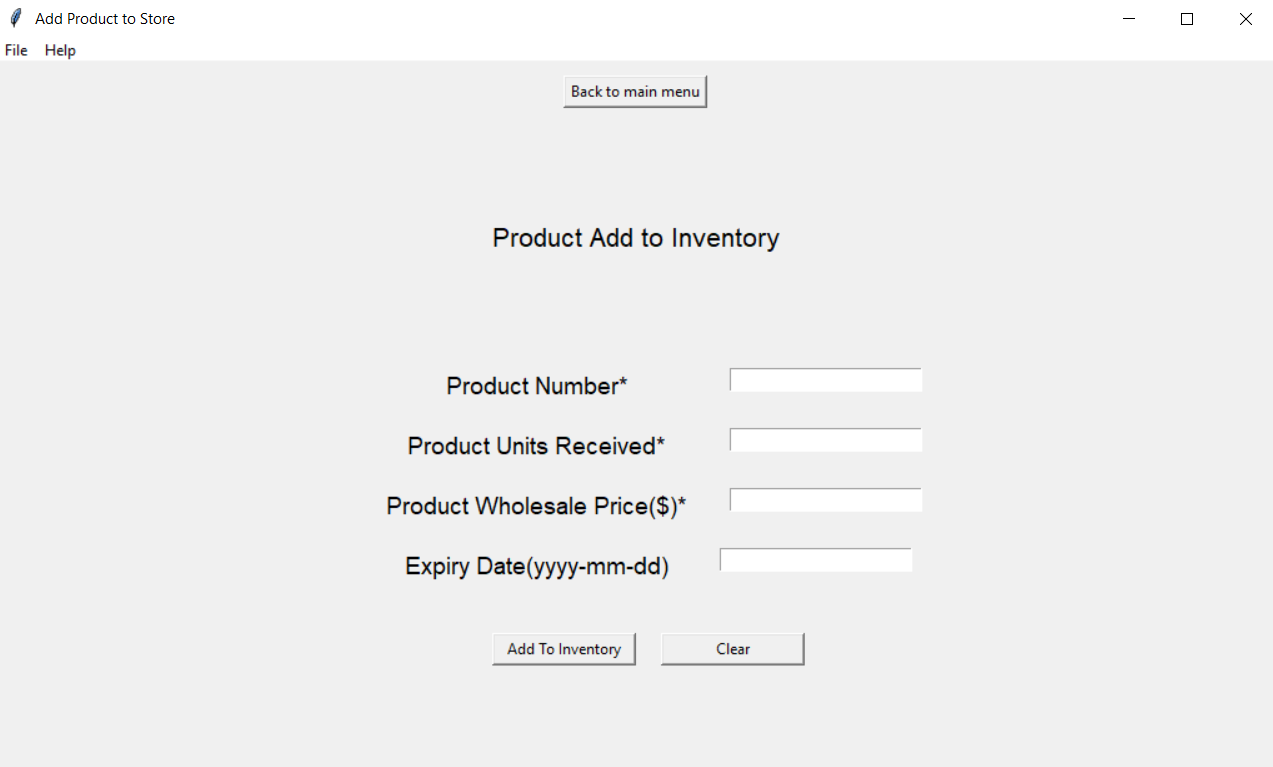
**registerproduct.py:**

Product Registration page helps the user to register a new product into the store. The product can only be added to the inventory once it has been registered. For registering the product user needs to provide the entries as shown in the image.



**addtoinventory.py**

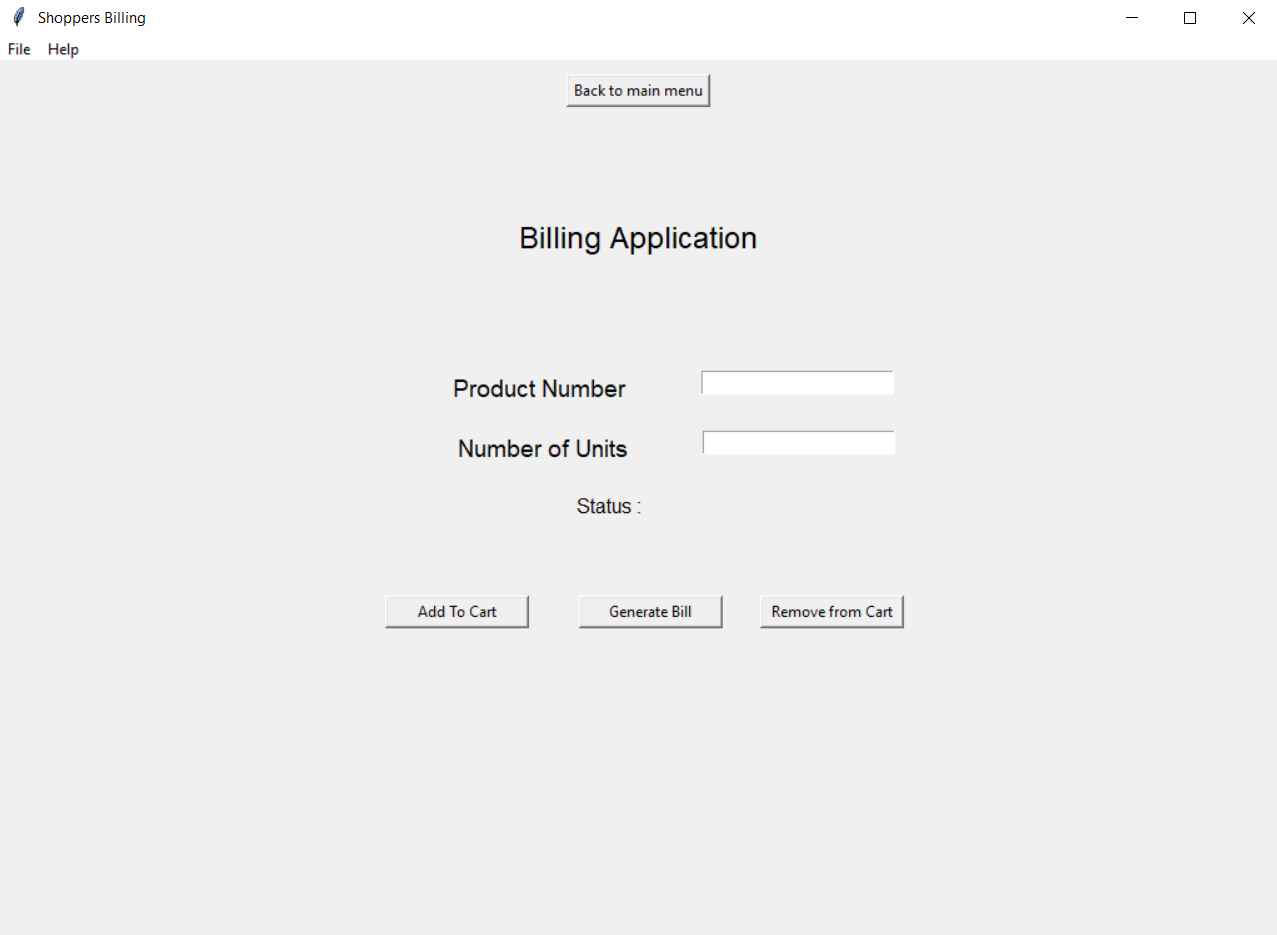
Once the product has been registered into the store, we can add the product into the inventory by providing the units received in shipment and the wholesale price of the product. The product that once added into the store can be used in the billing process. Note that the expiry date column is not mandatory to provide as per the application requirements.



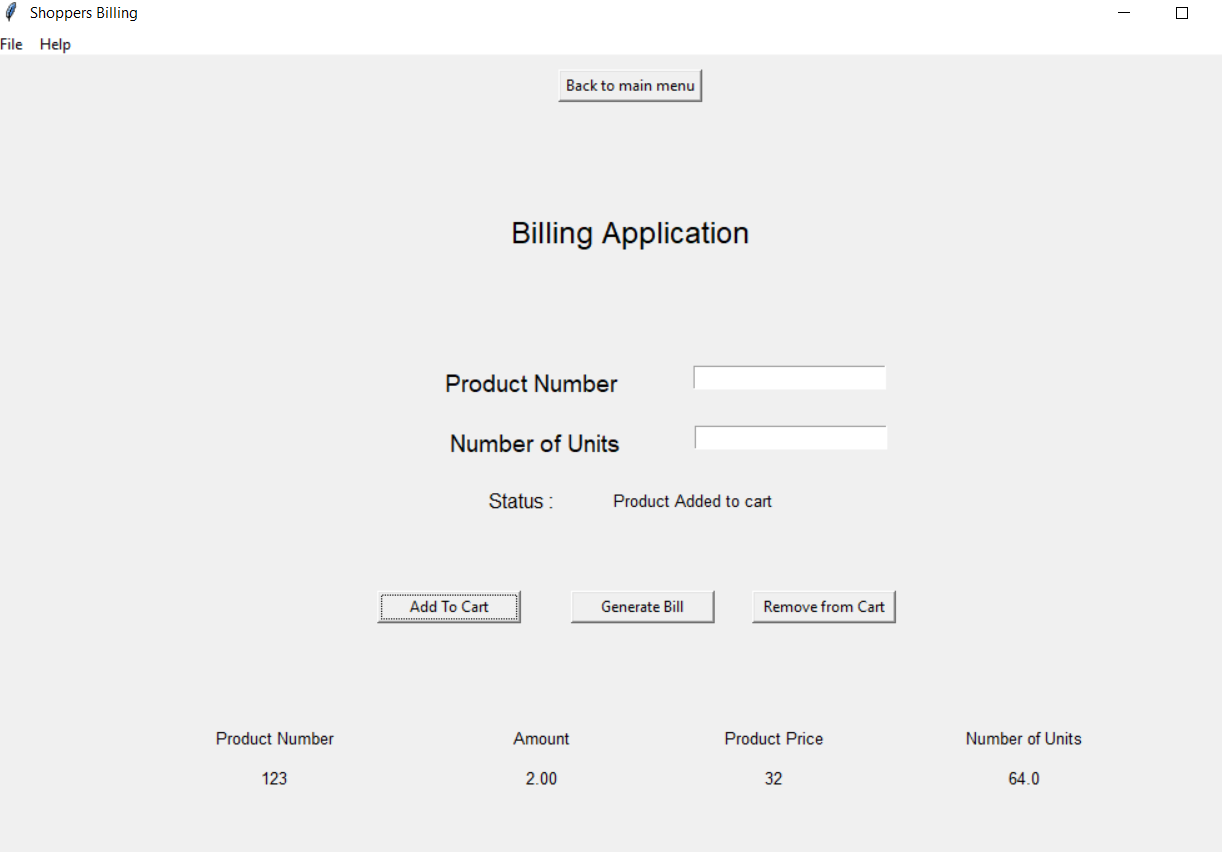
**individualbilling.py:**

Individual Billing page allows a user to perform the billing operation by adding multiple items to the cart and then generate a bill. The items added to the cart are shown below once added.

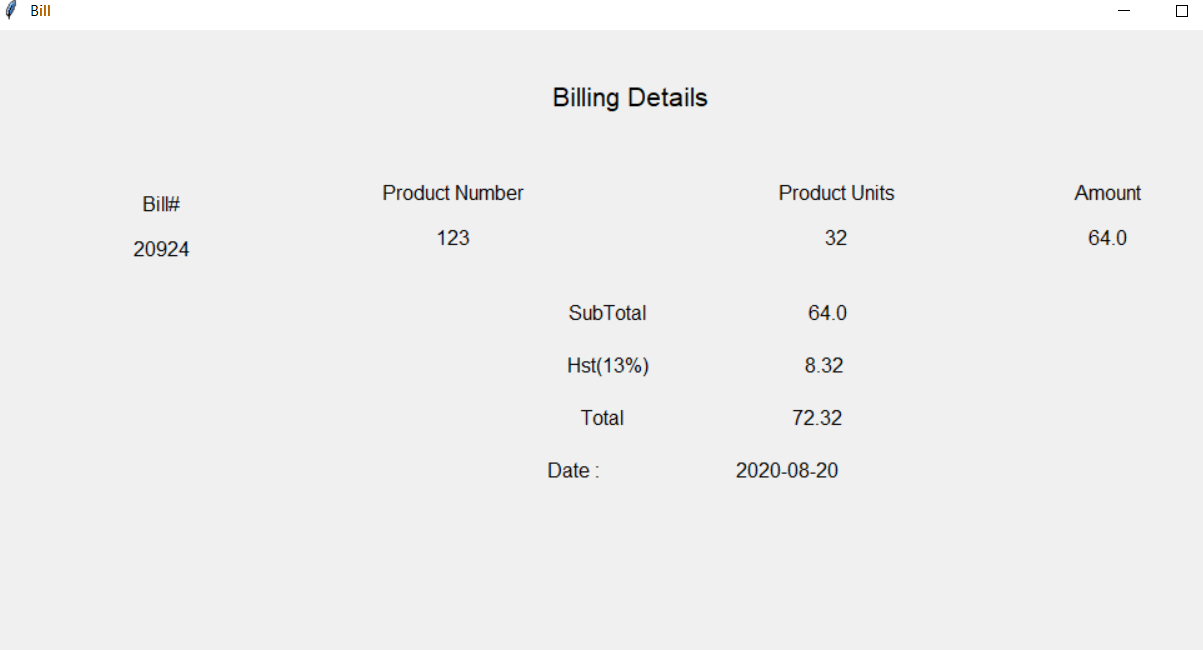
**Billing Application:**



**Items added to cart:**

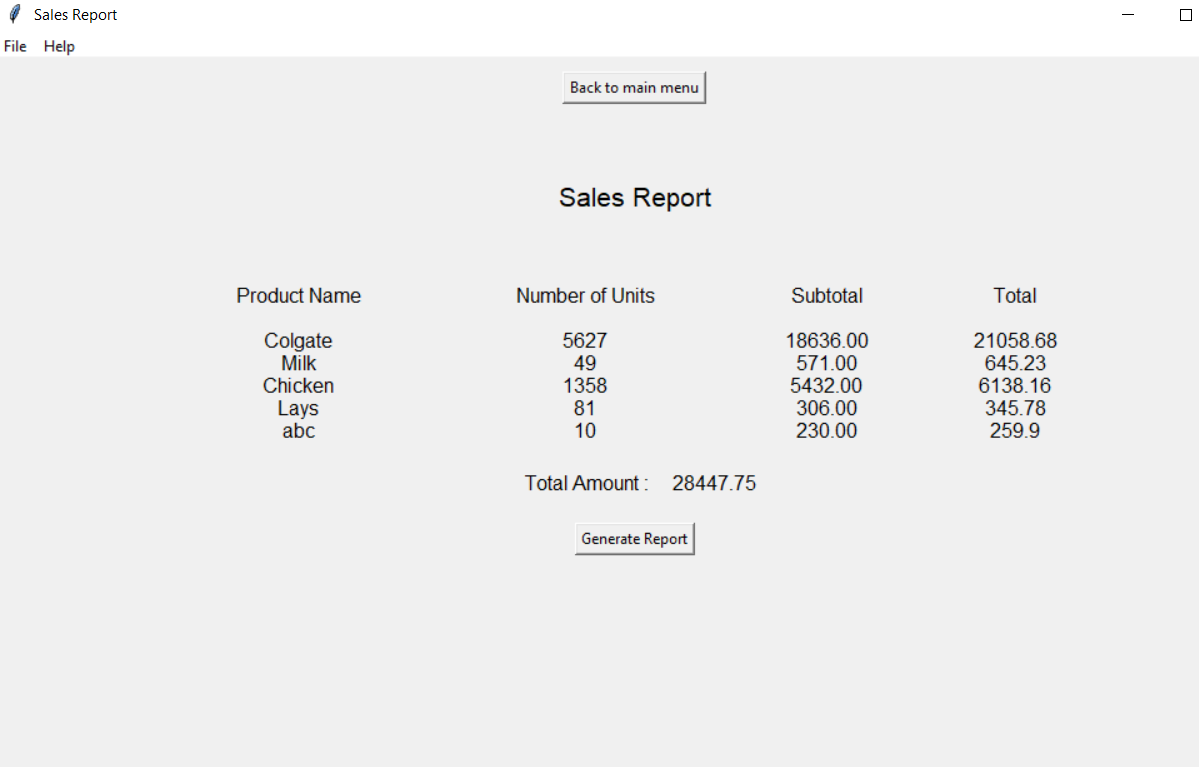


**Bill Generated for the items:**



**salesreport.py**

This page provides with the information of the products sold and the total amount of money that has been made of the sold products. It provides a button to generate the report upon clicking of which the report generates.



**stockingreport.py:**

This page provides with the information of total products available in the inventory and if the quantity of the product is less than 10, the product is added to the order list.

