Shopping Store Management System

M. Abhiram, 230150015
K. Ashmita, 230150014
B. Cherish, 230150007

1. Project Description

We have implemented a management system where we keep records of all the purchases done by both the customers and the shop admin, and also the inventory database. We have implemented the entire backend of the system using Object Oriented Programming in Python, stored the database in Comma Separated Values (CSV) files and used TkInter, a Python library for implementing a Graphical User Interface (GUI).

2. Database

Products.csv: It stores the information about all the products, in a .csv file. It stores features such as the *product name, stock quantity, price, brand, quantity, etc.*

Transactions.csv: It stores the information about all the transactions, done by the customers in a .csv file. It stores features such as the *order id, total price, discount, taxes, date, time, etc.*

Admin.csv: Similar to the transactions file, it stores the details about all the transactions done by the admin. It stores features such as *product name, quantity, expense, date, time, etc.*

3. Object Oriented Programming in Python

First, we defined a class Shop, which stores all the information about the shop, and defined instance methods for implementing the shop functionalities. When an object is created, all the data from the .csv files are loaded into their respective data structures. The description of the data structures used and the methods defined are given below,

3.1 Data Structures

Products: It is a dictionary, which stores all the product names as the keys, and the value associated with each key is another dictionary which stores the internal details of the corresponding product.

Transactions: It is a list of dictionaries corresponding to each transaction, which contains the details of all the transactions done by the customers.

Admin_Transactions: Similar to the transactions, it is also a list of dictionaries storing the details of the transactions done by the admin.

3.2 Inventory Management

Adding new products : The function *Buy_New_Products* gets called whenever the admin is adding a new product into the inventory. The function appends the transaction details into the admin transactions file and also appends the new product into the products file. Similarly, the internal data structures are also updated.

Restocking the old products : The function *Restock* gets called when the admin is restocking an already existing product. The function first updates the internal data structure used for storing the products information and then re writes the products file. Similar to the above function, it just appends the transaction details into the admin transactions file

Purchasing a Product : The function *buy_products* gets called when a customer places an order. It now has to update the product's stock quantity and store the transaction details, in a similar way as the restock functionality.

3.3 Additional Algorithms

Updating the Discount Rate: The admin has the option to update the discount depending upon the sale.

Monthly Profit Statement: The function *get_monthly_financials* automatically calculates the monthly statistics of all the transactions and returns them in a dictionary.

Zero Stock Alert: The function *Zero_Stock_Alert* finds all the products with stock quantity less than a threshold, which we assumed it to be 10. It then returns all the products and their corresponding stock quantities as a list.

TKinter, A Graphical User Interface

TKinter, is a Python library used for implementing a Graphical User Interface. It is responsible for both the responsiveness and the styling of the application. We can divide the user experience into two components :

Customer UI : The home page lists out all the products and their information in a tabular form. The user can simply click on the product and press add to cart. After the user is satisfied with the chosen
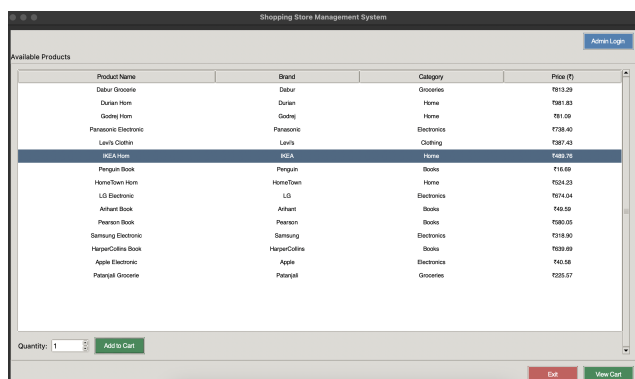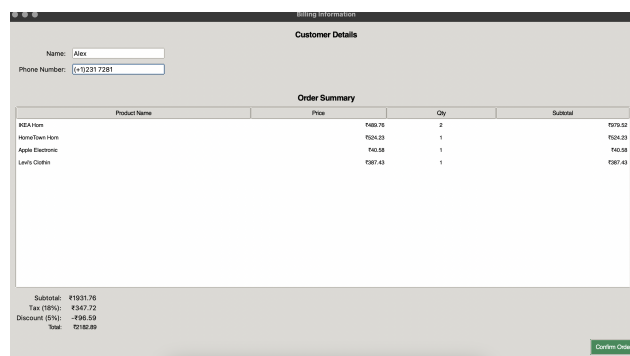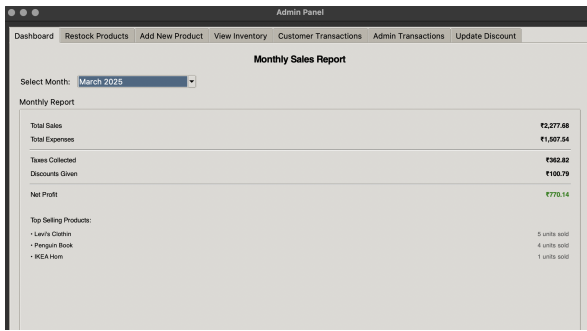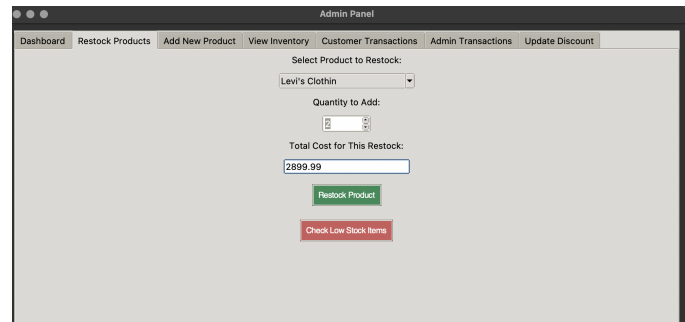

    (a) Home Page   (b) Checkout

Figure 1: Homepage and Cart

products and the prices, the user can checkout by just providing the name and the phone no. A unique 8 digit order id will be created once the order has been placed.

    Admin UI: Once the admin logs in with the correct credentials, the admin dashboard consists of monthly revenue statistics. Apart from the dashboard, there are more tabs through which the admin could navigate to, which consists of viewing the inventory, customer transactions, admin transactions, adding and restocking products, etc.

(a) Monthly Profit/Loss Statement

(b) Restocking Products

(c) Inventory

(d) Customer Transactions

Figure 2: Admin User Interface

Contribution Of Each Member of the Group

We think that all three members of the group have contributed equally to this wonderful project. It took a lot of persistent discussions and brainstorming to come up with the implementation strategy. We collectively worked on implementing the backend using Object Oriented Programming in Python, designing the essential algorithms wherever required and finally designing the Graphical User Interface using TkInter. **Coming together as a group and collaborating with each other, was what made us set the bar higher**.

References Used

- Object Oriented Programming in Python Link

- File Handling in Python Link

- Graphical User Interfaces with Tk Link

- Shop-Management-System–Python-GUI-TkinterLink