**CS631 DATA MANAGEMENT SYSTEM DESIGN**

**The Online Computer Store**

**Project Deliverable 3**

# GROUP MEMBERS

|  |  |
| --- | --- |
| **Shubham Pandkar** | **sp427** |
| **Joul Ammous** | **ja723** |
| **Payal Rane** | **pdr23** |

* **GOAL:**

In this phase of the project, we implement the The Online Computer Store. The android application is implemented using Python, SQLite3 as a backend in order to interact with the database and Python tkinter at the frontend (UI). All the functions that are included in the application are as mentioned

* Users can login with store by providing the username and password.
* The user can sign-up with user’s personal information, credit card details and shipping address.
* Users can link credit card with their account. The account has a card number, a security number, the name of its owner, the billing address, the type of credit card, and an expiry date.
* Basic validation criteria were implemented on registration process.
* A customer can specify shipping address and are identified by the name the customer choses for the address (which is unique among the shipping addresses of this customer) and the ID of the customer. For a shipping address the zip code, street name, street number, city, state and country are provided. If a customer is deleted, we need not keep track of her shipping addresses any longer.
* Each product has its own product ID, a name, a recommended price, and a brief textual description. It also has a unique product type. The quantity of each product in stock is recorded in the database.
* A shopping basket is always non-empty. It is created with the first item added.
* A successful sales transaction is recorded in the database, a customer buys a shopping basket (with products). The price and the quantity of every product in the transaction is available as well as the total amount of the transaction.
* There are three main categories of product types: Desktop, laptops, and printers.
* **IMPLEMENTATION:**

This application satisfies all the needs of a typical online computer store which helps all the people involved in manual purchasing of the computers. The online computer store database application has been made to provide the admin and users with better facilities to perform the functionalities mentioned in the goals of the project. The main purpose of this is to provide easy and convenient access to the users to add a product to cart and purchase it using credit card to a specific shipping address. The registration information can be edited by the user. The user can also view their transactions history.

The admin can generate statistics to analyse its sales and customers. He can generate five types of statistical data based on transaction date range:

1. Most frequently sold products
2. Products sold to highest number of distinct customers
3. Ten best customers in terms of most money spent
4. Maximum basket total per credit card
5. Average sold product price per product type.

We started our process with creating the sql queries in Sqlite3, we first created all the tables without any constraints and determining the primary key and foreign keys. Then we inserted random data, making sure that the inserts are successful.

We then created the front end with Python Tkinter, by creating a simple looking screen for Registration, login page, Main Menu, Online Sale, Sale Statistics and Update user information.

After doing the major work, we connected the front UI with the backend of Sqlite3, so that all the data can be easily accessed by the end user. So, while clicking any button or link it will fetch the details in the backend (DB) and after processing it will display the result to the user.

* **DIFFICULTIES WE FACED:**

* Creating APIs was an issue which we faced since this is the first app that we have developed.
* We also faced an issue with adding data to the cart.
* We faced difficulties in optimizing the queries so as to make the apk of smaller size.
* **NOTE**
* All the constraints including foreign key constraints are satisfied. Eg- If a customer ID is deleted from CUSTOMER, then all the entries related to that customer ID will be deleted from the entire database.
* Two changes in the schema-

1. The laptop is connected to product and additional attribute was added to it. (CPUType). The COMPUTER was labelled as DESKTOP.
2. The shipping address was connected to customer instead of SILVER\_AND\_ABOVE to avoid the exception of regular customer.
3. The status attribute was removed from CUSTOMER as we didn’t find need to identify customer status as it was present in SILVER\_AND\_ABOVE.

* **SQL COMMANDS FOR THE TABLES:**

**# QUERIES FOR CREATING THE TABLES**

**CREATE TABLE IF NOT EXISTS CUSTOMER**

**(CID VARCHAR(20) PRIMARY KEY,FName VARCHAR(20) NOT NULL,**

**LName VARCHAR(20) NOT NULL, Email VARCHAR(20) NOT NULL, Address VARCHAR(50) NOT NULL,**

**Phone VARCHAR(15) NOT NULL, Password VARCHAR(20) NOT NULL)**

**CREATE TABLE IF NOT EXISTS CREDIT\_CARD**

**(CCNumber VARCHAR(16) PRIMARY KEY, SecNumber VARCHAR(4) NOT NULL,**

**OwnerName VARCHAR(40) NOT NULL, CCType VARCHAR(10) NOT NUll, BilAddress VARCHAR(50),**

**ExpDate CHAR(6) NOT NULL,**

**CID VARCHAR(20) REFERENCES CUSTOMER(CID) ON UPDATE CASCADE ON DELETE CASCADE )**

**CREATE TABLE IF NOT EXISTS SHIPPING\_ADDRESS**

**(CID VARCHAR(20) REFERENCES CUSTOMER(CID) ON DELETE CASCADE ON UPDATE CASCADE, SAName VARCHAR(20),**

**RecipientName VARCHAR(20) NOT NULL, Street VARCHAR(10) NOT NULL, SNumber VARCHAR(5),**

**City VARCHAR(10) NOT NULL, Zip VARCHAR(5), State VARCHAR(20), Country VARCHAR(20) NOT NULL,**

**"PRIMARY KEY(CID,SAName)))**

**CREATE TABLE IF NOT EXISTS BASKET**

**(CARTID VARCHAR(10) PRIMARY KEY,**

**CID VARCHAR(10) REFERENCES CUSTOMER(CID) ON UPDATE CASCADE ON DELETE CASCADE)**

**CREATE TABLE IF NOT EXISTS APPEARS\_IN**

**(CARTID VARCHAR(10) REFERENCES BASKET(CARTID) ON UPDATE CASCADE ON DELETE CASCADE ,**

**PRODUCTID VARCHAR(10) REFERENCES PRODUCT(PRODUCTID) ON DELETE CASCADE ON UPDATE CASCADE,**

**Quantity INTEGER(5) NOT NULL, PriceSold DOUBLE(10) NOT NULL, PRIMARY KEY (CARTID, PRODUCTID))**

**CREATE TABLE IF NOT EXISTS PRODUCT**

**(PRODUCTID VARCHAR(10) PRIMARY KEY, PType VARCHAR(20),**

**PName VARCHAR(20), PPrice DOUBLE(10) NOT NULL, Quantity INTEGER(5), Description VARCHAR(20))**

**CREATE TABLE IF NOT EXISTS SILVER\_AND\_ABOVE**

**(CID VARCHAR(20) PRIMARY KEY REFERENCES CUSTOMER(CID) ON UPDATE CASCADE ON DELETE CASCADE,**

**CreditLine DOUBLE(10))**

**CREATE TABLE IF NOT EXISTS TRANS**

**(BID VARCHAR(10) REFERENCES BASKET(CARTID) ON UPDATE CASCADE ON DELETE CASCADE,**

**CCNumber VARCHAR(20) REFERENCES CREDIT\_CARD(CCNumber),**

**CID VARCHAR(20), SAName VARCHAR(10), TDate DATETIME(20) NOT NULL, TTag VARCHAR(10), "**

**PRIMARY KEY (BID, CCNumber, CID, SAName),**

**FOREIGN KEY (CID, SAName) REFERENCES SHIPPING\_ADDRESS(CID, SAName))**

**CREATE TABLE IF NOT EXISTS OFFER\_PRODUCT**

**(PID VARCHAR(10) PRIMARY KEY REFERENCES PRODUCT(PRODUCTID)ON UPDATE CASCADE ON DELETE CASCADE ,**

**OfferPrice DOUBLE(10))**

**CREATE TABLE IF NOT EXISTS PRINTER**

**(PID VARCHAR(10) PRIMARY KEY REFERENCES PRODUCT(PRODUCTID) ON UPDATE CASCADE ON DELETE CASCADE,**

**PrinterType VARCHAR(10), Resolution VARCHAR(10))**

**CREATE TABLE IF NOT EXISTS DESKTOP**

**(PID VARCHAR(10) PRIMARY KEY REFERENCES PRODUCT(PRODUCTID) ON UPDATE CASCADE ON DELETE CASCADE ,**

**CPUType VARCHAR(10))**

**CREATE TABLE IF NOT EXISTS LAPTOP**

**(PID VARCHAR(10) PRIMARY KEY REFERENCES PRODUCT(PRODUCTID) ON UPDATE CASCADE ON DELETE CASCADE,**

**BType VARCHAR(10), WEIGHT DOUBLE(10), CPUType VARCHAR(10))**

**# QUERIES FOR POPULATING THE TABLES**

**INSERT OR IGNORE INTO PRODUCT VALUES**

**('2432424', 'Desktop', 'pc1', 400.00, 5, 'HP')**

**INSERT OR IGNORE INTO DESKTOP VALUES**

**"('2432424', 'sda')**

**INSERT OR IGNORE INTO PRODUCT VALUES**

**('242624', 'Desktop', 'pc2', 500.00, 7, 'Dell')**

**INSERT OR IGNORE INTO DESKTOP VALUES**

**('242624', 'sweeda')**

**INSERT OR IGNORE INTO PRODUCT VALUES**

**"('54532442', 'Printer', 'printer1', 300.00, 10, 'Canon')**

**INSERT OR IGNORE INTO PRINTER VALUES**

**('54532442', 'inkjet', '2400x3200')**

**INSERT OR IGNORE INTO PRODUCT VALUES**

**('1123', 'Printer', 'printer2', 100.00, 20, 'HP')**

**INSERT OR IGNORE INTO PRINTER VALUES**

**"('1123', 'inkjet', '1024x2400')**

**INSERT OR IGNORE INTO PRODUCT VALUES**

**"('12345', 'Laptop', 'pc2', 300.00, 8, 'Asus')**

**INSERT OR IGNORE INTO LAPTOP VALUES**

**"('12345', 'office', 143.00, 'Intel')**

**INSERT OR IGNORE INTO PRODUCT VALUES**

**"('223355', 'Laptop', 'pc5', 600.00, 12, 'Lenovo')**

**INSERT OR IGNORE INTO LAPTOP VALUES**

**"('223355', 'gaming', 130.00, 'Intel')**

**INSERT OR IGNORE INTO PRODUCT VALUES**

**('313131', 'Accessories', 'Mouse', 50.00, 9, 'Razer')**

**INSERT OR IGNORE INTO CUSTOMER VALUES ('jack1', 'jack', 'smith', 'jsmith@gmail.com',**

**'165 South St. Allentown, PA USA', '4847106283', '12345')**

**INSERT OR IGNORE INTO CUSTOMER VALUES ('sandra1', 'sandra', 'smith', 'ssmith1@gmail.com',**

**"'123 North St. Allentown, PA USA', '6167106283', '12345')**

**INSERT OR IGNORE INTO CUSTOMER VALUES ('sam3', 'sam', 'smith', 'ssmith1@gmail.com',**

**'106 Front St. Newark, NJ USA', '2127106283', '111')**

**INSERT OR IGNORE INTO CUSTOMER VALUES ('sam0', 'sam', 'hannah', 's0smith1@gmail.com',**

**'103 East St. Newark, NJ USA', '2127106283', '111')**

**INSERT OR IGNORE INTO CUSTOMER VALUES ('sarah0', 'sarah', 'natoli', 'snatoli@gmail.com',**

**'102 North St. Newark, NJ USA', '3127106283', '111')**

**INSERT OR IGNORE INTO CUSTOMER VALUES ('josh1', 'josh', 'seinfeld', 'jseinfeld@gmail.com',**

**'106 Penn St. newark, NJ USA', '5127106225', '123')**

**INSERT OR IGNORE INTO CUSTOMER VALUES ('ross1', 'ross', 'geller', 'rgeller@gmail.com',**

**'232 Virginia St. Newark, NJ USA', '4842861762', '123')**

**INSERT OR IGNORE INTO SHIPPING\_ADDRESS VALUES**

**('jack1', 'parents', 'john', 'west ave.', '123', 'newark', '18097', 'nj', 'usa')**

**INSERT OR IGNORE INTO SHIPPING\_ADDRESS VALUES**

**('sandra1', 'sandra', 'sandra', 'south ave.', '245', 'newark', '18065', 'nj', 'usa')**

**INSERT OR IGNORE INTO SHIPPING\_ADDRESS VALUES**

**('sam3', 'sam', 'sam', 'north ave.', '632', 'newark', '18092', 'nj', 'usa')**

**INSERT OR IGNORE INTO SHIPPING\_ADDRESS VALUES**

**('sam0', 'parents', 'jacob', 'backer ave.', '102', 'newark', '17062', 'nj', 'usa')**

**INSERT OR IGNORE INTO SHIPPING\_ADDRESS VALUES**

**('sarah0', 'sarah', 'sarah', 'parker st.', '103', 'hoboken', '14087', 'nj', 'usa')**

**INSERT OR IGNORE INTO SHIPPING\_ADDRESS VALUES**

**('josh1', 'josh', 'josh', 'boston st.', '432', 'hoboken', '14098', 'nj', 'usa')**

**INSERT OR IGNORE INTO SHIPPING\_ADDRESS VALUES**

**('ross1', 'ross', 'ross', 'main st.', '510', 'hoboken', '14062', 'nj', 'usa')**

**INSERT OR IGNORE INTO CREDIT\_CARD VALUES**

**('23419378424', '123', 'jack', 'visa', '103 parker st. newark nj', '032024', 'sandra1')**

**INSERT OR IGNORE INTO CREDIT\_CARD VALUES**

**('23341255232', '125', 'sarah', 'visa', '123 parker st. newark nj', '032024', 'sandra1')**

**INSERT OR IGNORE INTO CREDIT\_CARD VALUES**

**('23341255232', '125', 'sarah', 'visa', '123 parker st. newark nj', '032024', 'sandra1')**

**INSERT OR IGNORE INTO CREDIT\_CARD VALUES**

**('23342235', '125', 'ross', 'visa', '123 south st. newark nj', '032024', 'ross1')**

**INSERT OR IGNORE INTO CREDIT\_CARD VALUES**

**('2335215732', '632', 'josh', 'visa', '123 parker st. newark nj', '032024', 'josh1')**

**INSERT OR IGNORE INTO CREDIT\_CARD VALUES**

**('2388532', '098', 'sam', 'visa', '247 penn st. newark nj', '032024', 'sam0')**

**INSERT OR IGNORE INTO CREDIT\_CARD VALUES**

**('23874444543', '098', 'sam', 'visa', '247 penn st. newark nj', '032024', 'sam3')**

**INSERT OR IGNORE INTO CREDIT\_CARD VALUES**

**('3346732244562', '356', 'jack', 'visa', '578 maryland st. newark nj', '032024', 'jack1')**

**INSERT OR IGNORE INTO CREDIT\_CARD VALUES**

**('735435326335', '234', 'sarah', 'visa', '247 penn st. newark nj', '032024', 'sarah0')**

**INSERT OR IGNORE INTO OFFER\_PRODUCT VALUES ('12345', 250)**

**INSERT OR IGNORE INTO OFFER\_PRODUCT VALUES ('223355', 500)**

**INSERT OR IGNORE INTO SILVER\_AND\_ABOVE VALUES ('jack1', 10)**

**INSERT OR IGNORE INTO SILVER\_AND\_ABOVE VALUES ('josh1', 20)**

**INSERT OR IGNORE INTO BASKET VALUES**

**"('121212', 'josh1')**

**INSERT OR IGNORE INTO TRANS VALUES**

**('121212', '2335215732', 'josh1', 'josh', 'April 01, 2022', 'Lost')**

**INSERT OR IGNORE INTO APPEARS\_IN VALUES**

**('121212', '242624', 1, 300.00)**

**INSERT OR IGNORE INTO BASKET VALUES**

**('435', 'sam0')**

**INSERT OR IGNORE INTO TRANS VALUES**

**('435', '2388532', 'sam0', 'parents', 'May 01, 2022', 'Delivered')**

**INSERT OR IGNORE INTO APPEARS\_IN VALUES**

**('435', '1123', 2, 200.00)**

**INSERT OR IGNORE INTO BASKET VALUES**

**('267', 'jack1')**

**INSERT OR IGNORE INTO TRANS VALUES**

**('267', '3346732244562', 'jack1', 'parents', 'March 01, 2022', 'Lost')**

**INSERT OR IGNORE INTO APPEARS\_IN VALUES**

**('267', '223355', 1, 600.00)**

**INSERT OR IGNORE INTO BASKET VALUES**

**('1357', 'ross1')**

**INSERT OR IGNORE INTO TRANS VALUES**

**('1357', '23342235', 'ross1', 'ross', 'March 01, 2022', 'Delivered')**

**INSERT OR IGNORE INTO APPEARS\_IN VALUES**

**('1357', '242624', 3, 1500.00)**

**# QUERIES FOR TESTING (INSERT, UPDATE, DELETE)**

**DELETE FROM PRODUCT WHERE PRODUCTID = '12345'**

**UPDATE PRODUCT SET PRODUCTID = '23222' WHERE PRODUCTID = '12345'**

**DELETE FROM CUSTOMER WHERE CID = 'jack1'**

**INSERT INTO CUSTOMER VALUES ('johnny1', 'johnny', 'cage', 'jcage@gmail.com',**

**'168 Main st. Allentown, PA USA', '4847106283', '12345')**

**DELETE FROM CUSTOMER WHERE CID = 'johnny1'**

**DELETE FROM SHIPPING\_ADDRESS WHERE CID = 'sam0'**

**DELETE FROM CREDIT\_CARD WHERE CID = 'sarah0'**

**UPDATE PRODUCT SET PPrice = 600 where PRODUCTID = '12345'**

**INSERT OR IGNORE INTO PRODUCT VALUES**

**('23134', 'Accessories', 'charger', 20.00, 10, 'Samsung')**

**INSERT OR IGNORE INTO SILVER\_AND\_ABOVE VALUES ('sarah0', 20)**

* **SOURCE CODE:**

**import tkinter as tk**

**from tkinter import \***

**from functools import partial**

**from tkinter import messagebox**

**import sqlite3**

**from tkinter import BOTH, END, LEFT**

**from tkinter import ttk**

**import random**

**import datetime**

**from datetime import date**

**def validateLogin(username, password):**

**username = username.get()**

**password = password.get()**

**global prim**

**prim = username**

**conn = sqlite3.connect('Newark\_IT.db')**

**c = conn.cursor()**

**q1 = (username, password)**

**find = "select \* from CUSTOMER where CID=? and password=?"**

**c.execute(find, q1)**

**result = c.fetchall()**

**conn.close()**

**if result:**

**messagebox.showinfo("Title", "Your login was done successfully")**

**screen1 = tk.Toplevel(tkWindow)**

**screen1.geometry('400x150')**

**screen1.title('SELECTION WINDOW')**

**# ONLINE SALE button**

**online\_sale\_button = Button(screen1, text="ONLINE SALE", command=onlinesale).grid(row=4, column=0)**

**# Sale Statistics button**

**sale\_button = Button(screen1, text="SALE STATISTICS", command=salestats).grid(row=4, column=1)**

**update\_button = Button(screen1, text="UPDATE DETAILS", command=update).grid(row=4, column=2)**

**else:**

**messagebox.showinfo("Title", "Invalid login id password")**

**return**

**def onlinesale():**

**global my\_cart**

**def my\_list():**

**my\_w = tk.Tk()**

**my\_w.geometry("1200x500")**

**my\_w.title("Online Sale")**

**global p\_name**

**global cat1**

**total\_str = tk.DoubleVar()**

**price\_str = tk.DoubleVar()**

**sb1\_str = tk.IntVar()**

**b1\_p\_id = tk.StringVar()**

**cat1 = {1: 'Computer', 2: 'Printer'} # list of categories**

**# Using treeview widget**

**trv = ttk.Treeview(my\_w, selectmode='browse')**

**trv.grid(row=1, column=1, padx=20, pady=20, rowspan=10)**

**# number of columns**

**trv["columns"] = ("1", "2", "3", "4", "5", "6")**

**# Defining heading**

**trv['show'] = 'headings'**

**# width of columns and alignment**

**trv.column("1", width=10, anchor='w')**

**trv.column("2", width=10, anchor='w')**

**trv.column("3", width=200, anchor='w')**

**trv.column("4", width=100, anchor='w')**

**trv.column("5", width=15, anchor='w')**

**trv.column("6", width=250, anchor='w')**

**# respective columns**

**trv.heading("1", text="PRODUCTID")**

**trv.heading("2", text="PRODUCTNAME")**

**trv.heading("3", text="PRODUCTPRICE")**

**trv.heading("4", text="PRODUCTTYPE")**

**trv.heading("5", text="QUANTITY")**

**trv.heading("6", text="DESCRIPTION")**

**def display\_data():**

**conn = sqlite3.connect('Newark\_IT.db')**

**c = conn.cursor()**

**query = "SELECT \* from PRODUCT"**

**r\_set = c.execute(query)**

**for dt in r\_set:**

**trv.insert("", 'end', iid=dt[0], text=dt[0], values=(dt[0], dt[1], dt[2], dt[3], dt[4], dt[5]))**

**conn.close()**

**def update\_total():**

**total\_str.set(int(sb1\_str.get()) \* int(price\_str.get()))**

**def update\_stock(p\_id):**

**global num**

**num = random.randint(1, 100000)**

**conn = sqlite3.connect('Newark\_IT.db')**

**c = conn.cursor()**

**q1 = "INSERT INTO BASKET VALUES (?, ?)"**

**var = (num, prim)**

**c.execute(q1, var)**

**val = (num, p\_id, sb1\_str.get(), total\_str.get())**

**print(val)**

**query = "INSERT INTO APPEARS\_IN VALUES (?, ?, ?, ?)"**

**c.execute(query, val)**

**# q2 = "INSERT INTO TRANS VALUES (?, ?, ?, ?, ?, ?)"**

**# var2 = (num, )**

**q4 = "select CCNumber from CREDIT\_CARD where CID = ?"**

**c.execute(q4, [prim])**

**cc = c.fetchall()**

**q5 = "select SAName from SHIPPING\_ADDRESS where CID = ?"**

**c.execute(q5, [prim])**

**sn = c.fetchall()**

**num1 = random.randint(1, 1000)**

**today = date.today()**

**today1 = today.strftime("%B %d, %Y")**

**q6 = "insert into TRANS values (?,?,?,?,?,?)"**

**vals = (num, cc[0][0], prim, sn[0][0], today1, " ")**

**c.execute(q6, vals)**

**conn.commit()**

**conn.close()**

**messagebox.showinfo("Title", "Order is placed")**

**def data\_show(\*args):**

**p\_id = trv.selection()[0]**

**conn = sqlite3.connect('Newark\_IT.db')**

**c = conn.cursor()**

**# print(p\_id)**

**query = 'SELECT \* FROM PRODUCT WHERE PRODUCTID=?'**

**c.execute(query, [p\_id])**

**row = c.fetchall()**

**# print(row)**

**l\_product.config(text=row[0][2])**

**price\_str.set(row[0][3])**

**b1\_p\_id.set(row[0][0])**

**# print(p\_id(row[4]))**

**if (row[0][4] > 0): # if stock is available**

**sb1.config(state='normal', to=row[0][4])**

**sb1\_str.set(1)**

**total\_str.set(str(row[0][3]))**

**b1.config(state='normal')**

**else:**

**sb1\_str.set(1)**

**sb1.config(state='disabled')**

**total\_str.set(0)**

**b1.config(state='disabled')**

**trv.bind("<<TreeviewSelect>>", data\_show)**

**font1 = ('Times', 24, 'bold')**

**font2\_product = ('Times', 18, 'normal')**

**font\_price = ('Times', 16, 'normal')**

**l\_product = tk.Label(my\_w, text='data ', bg='lightblue', font=font2\_product, width=30)**

**l\_product.grid(row=1, column=2, columnspan=2, padx=20, sticky='w')**

**l\_price = tk.Label(my\_w, text='Price')**

**l\_price.grid(row=3, column=2, padx=10, pady=10)**

**price = tk.Label(my\_w, textvariable=price\_str, font=font\_price)**

**price.grid(row=3, column=3, padx=10, pady=10)**

**l\_stock = tk.Label(my\_w, text='Quantity')**

**l\_stock.grid(row=4, column=2, padx=10, pady=10)**

**sb1 = Spinbox(my\_w, from\_=1, to=10, width=5, font=font1, state='disabled',**

**textvariable=sb1\_str, command=lambda: update\_total())**

**sb1.grid(row=4, column=3, padx=10)**

**l\_total = tk.Label(my\_w, text='Total')**

**l\_total.grid(row=5, column=2, padx=10, pady=10)**

**total = tk.Label(my\_w, textvariable=total\_str, font=font\_price, bg='lightgreen')**

**total.grid(row=5, column=3)**

**b1 = tk.Button(my\_w, text='add to cart and pay'**

**, command=lambda: update\_stock(b1\_p\_id.get()))**

**b1.grid(row=6, column=2, columnspan=2)**

**b2 = tk.Button(my\_w, text='see transactions'**

**, command=lambda: tran\_det())**

**b2.grid(row=7, column=2, columnspan=2)**

**display\_data() # show the top data**

**#### show the cart content ####**

**my\_list()**

**def tran\_det():**

**tra = tk.Toplevel(tkWindow)**

**tra.geometry('1200x500')**

**tra.title('Transactions')**

**trv = ttk.Treeview(tra, selectmode='browse')**

**trv.grid(row=1, column=1, padx=20, pady=20, rowspan=10)**

**# number of columns**

**trv["columns"] = ("1", "2", "3", "4", "5", "6")**

**# Defining heading**

**trv['show'] = 'headings'**

**# width of columns and alignment**

**trv.column("1", width=10, anchor='w')**

**trv.column("2", width=10, anchor='w')**

**trv.column("3", width=200, anchor='w')**

**trv.column("4", width=100, anchor='w')**

**trv.column("5", width=15, anchor='w')**

**trv.column("6", width=250, anchor='w')**

**# respective columns**

**trv.heading("1", text="CARTID")**

**trv.heading("2", text="CARD NUMBER")**

**trv.heading("3", text="USERID")**

**trv.heading("4", text="SHIPPING ADDRESS NAME")**

**trv.heading("5", text="TRANS DATE")**

**trv.heading("6", text="TRANS TAG")**

**conn = sqlite3.connect('Newark\_IT.db')**

**c = conn.cursor()**

**query = "SELECT \* from TRANS where CID = ?"**

**r\_set = c.execute(query, [prim])**

**for dt in r\_set:**

**trv.insert("", 'end', iid=dt[0], text=dt[0], values=(dt[0], dt[1], dt[2], dt[3], dt[4], dt[5]))**

**conn.close()**

**def salestats():**

**sale = tk.Toplevel(tkWindow)**

**sale.title('Sale stats')**

**option1 = tk.StringVar(sale)**

**option1.set("filters") # default value**

**def disp1(option1):**

**go1 = partial(go\_choice\_1, option1)**

**go1\_button = Button(sale, text="GO", command=go1).grid(row=4, column=0)**

**l1 = tk.Label(sale, text='Select One', width=10)**

**l1.grid(row=2, column=1)**

**fill = ["1", "2", "3", "4", "5"]**

**om1 = tk.OptionMenu(sale, option1, \*fill, command=disp1)**

**om1.grid(row=2, column=2)**

**def update():**

**global up**

**up = tk.Toplevel(tkWindow)**

**up.title('USER INFO UPDATE')**

**option2 = tk.StringVar(up)**

**option2.set("Details") # default value**

**def disp2(option2):**

**go2 = go\_choice2(option2)**

**go2\_button = Button(up, text="GO", command=go2).grid(row=4, column=0)**

**l1 = tk.Label(up, text='Select One', width=10)**

**l1.grid(row=2, column=1)**

**det = ["password", "First name", "Last name", "email", "phone", "Card number", "Sec Number", "Owner name",**

**"Card type", "billing address", "shipping address name", "Recipient name", "Shipping address number",**

**"Street", "City", "Country", "State", "Zip"]**

**om1 = tk.OptionMenu(up, option2, \*det, command=disp2)**

**om1.grid(row=2, column=2)**

**def go\_choice\_1(choice):**

**global fil**

**fil = tk.Toplevel(tkWindow)**

**fil.geometry('400x150')**

**fil.title('Filter')**

**fromdataLabel = Label(fil, text="fromdate").grid(row=0, column=0)**

**fromdate = StringVar()**

**usernameEntry = Entry(fil, textvariable=fromdate).grid(row=0, column=1)**

**todateLabel = Label(fil, text="todate").grid(row=1, column=0)**

**todate = StringVar()**

**todateEntry = Entry(fil, textvariable=todate).grid(row=1, column=1)**

**search1 = partial(search\_1, choice, fromdate, todate)**

**Search\_button = Button(fil, text="Search", command=search1).grid(row=2, column=0)**

**def search\_1(choice, fromdate, todate):**

**fromdate = fromdate.get()**

**todate = todate.get()**

**if choice == "1":**

**s1 = tk.Toplevel(tkWindow)**

**s1.geometry('1200x500')**

**s1.title('Transactions')**

**trv = ttk.Treeview(s1, selectmode='browse')**

**trv.grid(row=1, column=1, padx=20, pady=20, rowspan=10)**

**# number of columns**

**trv["columns"] = ("1", "2")**

**# Defining heading**

**trv['show'] = 'headings'**

**# width of columns and alignment**

**trv.column("1", width=100, anchor='w')**

**trv.column("2", width=100, anchor='w')**

**# respective columns**

**trv.heading("1", text="PRODUCTID")**

**trv.heading("2", text="COUNT")**

**conn = sqlite3.connect('Newark\_IT.db')**

**c = conn.cursor()**

**valu = (fromdate, todate)**

**query = "select PRODUCTID, COUNT(Quantity) from APPEARS\_IN INNER JOIN TRANS WHERE CARTID = BID AND TDATE BETWEEN ? AND ? GROUP BY PRODUCTID ORDER BY COUNT(Quantity) DESC"**

**c.execute(query, valu)**

**r\_set = c.fetchall()**

**for dt in r\_set:**

**trv.insert("", 'end', iid=dt[0], text=dt[0], values=(dt[0], dt[1]))**

**conn.close()**

**elif choice == "2":**

**s1 = tk.Toplevel(tkWindow)**

**s1.geometry('1200x500')**

**s1.title('Transactions')**

**trv = ttk.Treeview(s1, selectmode='browse')**

**trv.grid(row=1, column=1, padx=20, pady=20, rowspan=10)**

**# number of columns**

**trv["columns"] = ("1", "2")**

**# Defining heading**

**trv['show'] = 'headings'**

**# width of columns and alignment**

**trv.column("1", width=100, anchor='w')**

**trv.column("2", width=100, anchor='w')**

**# respective columns**

**trv.heading("1", text="PRODUCTID")**

**trv.heading("2", text="COUNT")**

**conn = sqlite3.connect('Newark\_IT.db')**

**c = conn.cursor()**

**valu = (fromdate, todate)**

**query = "select PRODUCTID, COUNT(DISTINCT(CID)) from APPEARS\_IN INNER JOIN TRANS WHERE " \**

**"CARTID = BID AND TDATE BETWEEN ? AND ? GROUP BY PRODUCTID ORDER BY COUNT(DISTINCT(CID)) DESC"**

**c.execute(query, valu)**

**r\_set = c.fetchall()**

**for dt in r\_set:**

**trv.insert("", 'end', iid=dt[0], text=dt[0], values=(dt[0], dt[1]))**

**conn.close()**

**elif choice == "3":**

**s1 = tk.Toplevel(tkWindow)**

**s1.geometry('1200x500')**

**s1.title('Transactions')**

**trv = ttk.Treeview(s1, selectmode='browse')**

**trv.grid(row=1, column=1, padx=20, pady=20, rowspan=10)**

**# number of columns**

**trv["columns"] = ("1", "2")**

**# Defining heading**

**trv['show'] = 'headings'**

**# width of columns and alignment**

**trv.column("1", width=100, anchor='w')**

**trv.column("2", width=100, anchor='w')**

**# respective columns**

**trv.heading("1", text="CUSTOMER ID")**

**trv.heading("2", text="SUM OF MONEY SPENT")**

**conn = sqlite3.connect('Newark\_IT.db')**

**c = conn.cursor()**

**valu = (fromdate, todate)**

**query = "select CID, SUM(PriceSold) from APPEARS\_IN INNER JOIN TRANS WHERE " \**

**"CARTID = BID AND TDATE BETWEEN ? AND ? GROUP BY CID ORDER BY SUM(PriceSold) DESC"**

**c.execute(query, valu)**

**r\_set = c.fetchall()**

**for dt in r\_set:**

**trv.insert("", 'end', iid=dt[0], text=dt[0], values=(dt[0], dt[1]))**

**conn.close()**

**elif choice == "4":**

**s1 = tk.Toplevel(tkWindow)**

**s1.geometry('1200x500')**

**s1.title('Transactions')**

**trv = ttk.Treeview(s1, selectmode='browse')**

**trv.grid(row=1, column=1, padx=20, pady=20, rowspan=10)**

**# number of columns**

**trv["columns"] = ("1", "2")**

**# Defining heading**

**trv['show'] = 'headings'**

**# width of columns and alignment**

**trv.column("1", width=100, anchor='w')**

**trv.column("2", width=100, anchor='w')**

**# respective columns**

**trv.heading("1", text="Credit Card Number")**

**trv.heading("2", text="Max Basket Total")**

**conn = sqlite3.connect('Newark\_IT.db')**

**c = conn.cursor()**

**valu = (fromdate, todate)**

**query = "select CCNumber, MAX(PriceSold) from APPEARS\_IN INNER JOIN TRANS WHERE " \**

**"CARTID = BID AND TDATE BETWEEN ? AND ? GROUP BY CCNumber"**

**c.execute(query, valu)**

**r\_set = c.fetchall()**

**for dt in r\_set:**

**trv.insert("", 'end', iid=dt[0], text=dt[0], values=(dt[0], dt[1]))**

**conn.close()**

**elif choice == "5":**

**s1 = tk.Toplevel(tkWindow)**

**s1.geometry('1200x500')**

**s1.title('Transactions')**

**trv = ttk.Treeview(s1, selectmode='browse')**

**trv.grid(row=1, column=1, padx=20, pady=20, rowspan=10)**

**# number of columns**

**trv["columns"] = ("1", "2")**

**# Defining heading**

**trv['show'] = 'headings'**

**# width of columns and alignment**

**trv.column("1", width=100, anchor='w')**

**trv.column("2", width=100, anchor='w')**

**# respective columns**

**trv.heading("1", text="PRODUCT TYPE")**

**trv.heading("2", text="AVG PRICE SOLD")**

**conn = sqlite3.connect('Newark\_IT.db')**

**c = conn.cursor()**

**valu = (fromdate, todate)**

**query = "select PType, AVG(PriceSold) from APPEARS\_IN NATURAL JOIN PRODUCT " \**

**"WHERE NOT EXISTS (SELECT \* FROM TRANS WHERE TDATE BETWEEN ? AND ?) GROUP BY PTYPE"**

**c.execute(query, valu)**

**r\_set = c.fetchall()**

**for dt in r\_set:**

**trv.insert("", 'end', iid=dt[0], text=dt[0], values=(dt[0], dt[1]))**

**conn.close()**

**else:**

**pass**

**def go\_choice2(choice):**

**up\_win = tk.Toplevel(up)**

**up\_win.geometry('400x150')**

**up\_win.title('Update')**

**if choice == "password":**

**temp = 1**

**passwordLabel = Label(up\_win, text="Password").grid(row=0, column=0)**

**password = StringVar()**

**passwordEntry = Entry(up\_win, textvariable=password, show='\*').grid(row=1, column=1)**

**go3 = partial(go\_choice3, password, temp)**

**go3\_button = Button(up\_win, text="GO", command=go3).grid(row=2, column=0)**

**elif choice == "First name":**

**temp = 2**

**FnameLabel = Label(up\_win, text="First Name").grid(row=0, column=0)**

**Fname = StringVar()**

**FnameEntry = Entry(up\_win, textvariable=Fname).grid(row=0, column=1)**

**go3 = partial(go\_choice3, Fname, temp)**

**go3\_button = Button(up\_win, text="GO", command=go3).grid(row=2, column=0)**

**elif choice == "Last name":**

**temp = 3**

**LnameLabel = Label(up\_win, text="Last Name").grid(row=0, column=0)**

**Lname = StringVar()**

**LnameEntry = Entry(up\_win, textvariable=Lname).grid(row=0, column=1)**

**go3 = partial(go\_choice3, Lname, temp)**

**go3\_button = Button(up\_win, text="GO", command=go3).grid(row=2, column=0)**

**elif choice == "email":**

**temp = 4**

**emailLabel = Label(up\_win, text="email").grid(row=0, column=0)**

**email = StringVar()**

**emailEntry = Entry(up\_win, textvariable=email).grid(row=0, column=1)**

**go3 = partial(go\_choice3, email, temp)**

**go3\_button = Button(up\_win, text="GO", command=go3).grid(row=2, column=0)**

**elif choice == "phone":**

**temp = 5**

**phoneLabel = Label(up\_win, text="phone").grid(row=0, column=0)**

**phone = StringVar()**

**phoneEntry = Entry(up\_win, textvariable=phone).grid(row=0, column=1)**

**go3 = partial(go\_choice3, phone, temp)**

**go3\_button = Button(up\_win, text="GO", command=go3).grid(row=2, column=0)**

**elif choice == "Card number":**

**temp = 6**

**CnumLabel = Label(up\_win, text="Credit card number").grid(row=0, column=0)**

**Cnum = IntVar()**

**CnumEntry = Entry(up\_win, textvariable=Cnum).grid(row=0, column=1)**

**go3 = partial(go\_choice3, Cnum, temp)**

**go3\_button = Button(up\_win, text="GO", command=go3).grid(row=2, column=0)**

**elif choice == "Sec Number":**

**temp = 7**

**SnumLabel = Label(up\_win, text="sec").grid(row=0, column=0)**

**Snum = IntVar()**

**SnumEntry = Entry(up\_win, textvariable=Snum).grid(row=0, column=1)**

**go3 = partial(go\_choice3, Snum, temp)**

**go3\_button = Button(up\_win, text="GO", command=go3).grid(row=2, column=0)**

**elif choice == "Owner name":**

**temp = 8**

**OnameLabel = Label(up\_win, text="Owner Name").grid(row=0, column=0)**

**Oname = StringVar()**

**OnameEntry = Entry(up\_win, textvariable=Oname).grid(row=0, column=1)**

**go3 = partial(go\_choice3, Oname, temp)**

**go3\_button = Button(up\_win, text="GO", command=go3).grid(row=2, column=0)**

**elif choice == "Card type":**

**temp = 9**

**CtypeLabel = Label(up\_win, text="Card type").grid(row=0, column=0)**

**Ctype = StringVar()**

**CtypeEntry = Entry(up\_win, textvariable=Ctype).grid(row=0, column=1)**

**go3 = partial(go\_choice3, Ctype, temp)**

**go3\_button = Button(up\_win, text="GO", command=go3).grid(row=2, column=0)**

**elif choice == "billing address":**

**temp = 10**

**BAddress = Label(up\_win, text="Billing Address").grid(row=0, column=0)**

**BAddress = StringVar()**

**BAddressEntry = Entry(up\_win, textvariable=BAddress).grid(row=0, column=1)**

**go3 = partial(go\_choice3, BAddress, temp)**

**go3\_button = Button(up\_win, text="GO", command=go3).grid(row=2, column=0)**

**elif choice == "shipping address name":**

**temp = 11**

**Sname = Label(up\_win, text="Shipping Address name").grid(row=0, column=0)**

**Sname = StringVar()**

**SnameEntry = Entry(up\_win, textvariable=Sname).grid(row=0, column=1)**

**go3 = partial(go\_choice3, Sname, temp)**

**go3\_button = Button(up\_win, text="GO", command=go3).grid(row=2, column=0)**

**elif choice == "Recipient name":**

**temp = 12**

**RnameLabel = Label(up\_win, text="Recepient Name").grid(row=0, column=0)**

**Rname = StringVar()**

**RnameEntry = Entry(up\_win, textvariable=Rname).grid(row=0, column=1)**

**go3 = partial(go\_choice3, Rname, temp)**

**go3\_button = Button(up\_win, text="GO", command=go3).grid(row=2, column=0)**

**elif choice == "Shipping address number":**

**temp = 13**

**SanumLabel = Label(up\_win, text="Address number").grid(row=0, column=0)**

**Sanum = IntVar()**

**SanumEntry = Entry(up\_win, textvariable=Sanum).grid(row=0, column=1)**

**go3 = partial(go\_choice3, Sanum, temp)**

**go3\_button = Button(up\_win, text="GO", command=go3).grid(row=2, column=0)**

**elif choice == "Street":**

**temp = 14**

**Street = Label(up\_win, text="Street name").grid(row=0, column=0)**

**Street = StringVar()**

**StreetEntry = Entry(up\_win, textvariable=Street).grid(row=0, column=1)**

**go3 = partial(go\_choice3, Street, temp)**

**go3\_button = Button(up\_win, text="GO", command=go3).grid(row=2, column=0)**

**elif choice == "City":**

**temp = 15**

**City = Label(up\_win, text="City").grid(row=0, column=0)**

**City = StringVar()**

**CityEntry = Entry(up\_win, textvariable=City).grid(row=0, column=1)**

**go3 = partial(go\_choice3, City, temp)**

**go3\_button = Button(up\_win, text="GO", command=go3).grid(row=2, column=0)**

**elif choice == "Country":**

**temp = 16**

**Country = Label(up\_win, text="Country").grid(row=0, column=0)**

**Country = StringVar()**

**CountryEntry = Entry(up\_win, textvariable=Country).grid(row=0, column=1)**

**go3 = partial(go\_choice3, Country, temp)**

**go3\_button = Button(up\_win, text="GO", command=go3).grid(row=2, column=0)**

**elif choice == "State":**

**temp = 17**

**State = Label(up\_win, text="State").grid(row=0, column=0)**

**State = StringVar()**

**StateEntry = Entry(up\_win, textvariable=State).grid(row=0, column=1)**

**go3 = partial(go\_choice3, State, temp)**

**go3\_button = Button(up\_win, text="GO", command=go3).grid(row=2, column=0)**

**elif choice == "Zip":**

**temp = 18**

**Zip = Label(up\_win, text="Zip Code").grid(row=0, column=0)**

**Zip = IntVar()**

**ZipEntry = Entry(up\_win, textvariable=Zip).grid(row=0, column=1)**

**go3 = partial(go\_choice3, Zip, temp)**

**go3\_button = Button(up\_win, text="GO", command=go3).grid(row=2, column=0)**

**else:**

**messagebox.showinfo("Title", "Some error occurred")**

**def go\_choice3(new, temp):**

**if temp == 1:**

**x1 = new.get()**

**conn = sqlite3.connect('Newark\_IT.db')**

**c = conn.cursor()**

**n1 = (x1, prim)**

**upd1 = "UPDATE CUSTOMER SET password = ? where CID = ?"**

**c.execute(upd1, n1)**

**conn.commit()**

**conn.close()**

**messagebox.showinfo("Title", "Password Updated")**

**elif temp == 2:**

**x1 = new.get()**

**conn = sqlite3.connect('Newark\_IT.db')**

**c = conn.cursor()**

**n1 = (x1, prim)**

**upd1 = "UPDATE CUSTOMER SET FName = ? where CID = ?"**

**c.execute(upd1, n1)**

**conn.commit()**

**conn.close()**

**messagebox.showinfo("Title", "FIRST NAME Updated")**

**elif temp == 3:**

**x1 = new.get()**

**conn = sqlite3.connect('Newark\_IT.db')**

**c = conn.cursor()**

**n1 = (x1, prim)**

**upd1 = "UPDATE CUSTOMER SET LName = ? where CID = ? "**

**c.execute(upd1, n1)**

**conn.commit()**

**conn.close()**

**messagebox.showinfo("Title", "LAST NAME Updated")**

**elif temp == 4:**

**x1 = new.get()**

**conn = sqlite3.connect('Newark\_IT.db')**

**c = conn.cursor()**

**n1 = (x1, prim)**

**upd1 = "UPDATE CUSTOMER SET Email = ? where CID = ? "**

**c.execute(upd1, n1)**

**conn.commit()**

**conn.close()**

**messagebox.showinfo("Title", "EMAIL Updated")**

**elif temp == 5:**

**x1 = new.get()**

**conn = sqlite3.connect('Newark\_IT.db')**

**c = conn.cursor()**

**n1 = (x1, prim)**

**upd1 = "UPDATE CUSTOMER SET Phone = ? where CID = ? "**

**c.execute(upd1, n1)**

**conn.commit()**

**conn.close()**

**messagebox.showinfo("Title", "PHONE Updated")**

**elif temp == 6:**

**x1 = new.get()**

**conn = sqlite3.connect('Newark\_IT.db')**

**c = conn.cursor()**

**n1 = (x1, prim)**

**upd1 = "UPDATE CREDIT\_CARD SET CCNumber = ? where CID = ?"**

**c.execute(upd1, n1)**

**conn.commit()**

**conn.close()**

**messagebox.showinfo("Title", "Credit card number Updated")**

**elif temp == 7:**

**x1 = new.get()**

**conn = sqlite3.connect('Newark\_IT.db')**

**c = conn.cursor()**

**n1 = (x1, prim)**

**upd1 = "UPDATE CREDIT\_CARD SET SecNumber = ? where CID = ?"**

**c.execute(upd1, n1)**

**conn.commit()**

**conn.close()**

**messagebox.showinfo("Title", "SEC NUMBER Updated")**

**elif temp == 8:**

**x1 = new.get()**

**conn = sqlite3.connect('Newark\_IT.db')**

**c = conn.cursor()**

**n1 = (x1, prim)**

**upd1 = "UPDATE CREDIT\_CARD SET OwnerName = ? where CID = ?"**

**c.execute(upd1, n1)**

**conn.commit()**

**conn.close()**

**messagebox.showinfo("Title", "Owner name Updated")**

**elif temp == 9:**

**x1 = new.get()**

**conn = sqlite3.connect('Newark\_IT.db')**

**c = conn.cursor()**

**n1 = (x1, prim)**

**upd1 = "UPDATE CREDIT\_CARD SET CCType = ? where CID = ?"**

**c.execute(upd1, n1)**

**conn.commit()**

**conn.close()**

**messagebox.showinfo("Title", "Card Type Updated")**

**elif temp == 10:**

**x1 = new.get()**

**conn = sqlite3.connect('Newark\_IT.db')**

**c = conn.cursor()**

**n1 = (x1, prim)**

**upd1 = "UPDATE CREDIT\_CARD SET BilAddress = ? where CID = ?"**

**c.execute(upd1, n1)**

**upd2 = "UPDATE CUSTOMER SET Address = ? where CID = ?"**

**c.execute(upd2, n1)**

**conn.commit()**

**conn.close()**

**messagebox.showinfo("Title", "Billing address Updated")**

**elif temp == 11:**

**x1 = new.get()**

**conn = sqlite3.connect('Newark\_IT.db')**

**c = conn.cursor()**

**n1 = (x1, prim)**

**upd1 = "UPDATE SHIPPING\_ADDRESS SET SAName = ? where CID = ? "**

**c.execute(upd1, n1)**

**conn.commit()**

**conn.close()**

**messagebox.showinfo("Title", "shipping address name Updated")**

**elif temp == 12:**

**x1 = new.get()**

**conn = sqlite3.connect('Newark\_IT.db')**

**c = conn.cursor()**

**n1 = (x1, prim)**

**upd1 = "UPDATE SHIPPING\_ADDRESS SET RecipientName = ? where CID = ?"**

**c.execute(upd1, n1)**

**conn.commit()**

**conn.close()**

**messagebox.showinfo("Title", "Recipient name Updated")**

**elif temp == 13:**

**x1 = new.get()**

**conn = sqlite3.connect('Newark\_IT.db')**

**c = conn.cursor()**

**n1 = (x1, prim)**

**upd1 = "UPDATE SHIPPING\_ADDRESS SET SNumber = ? where CID = ?"**

**c.execute(upd1, n1)**

**conn.commit()**

**conn.close()**

**messagebox.showinfo("Title", "Shipping address number Updated")**

**elif temp == 14:**

**x1 = new.get()**

**conn = sqlite3.connect('Newark\_IT.db')**

**c = conn.cursor()**

**n1 = (x1, prim)**

**upd1 = "UPDATE SHIPPING\_ADDRESS SET Street = ? where CID = ?"**

**c.execute(upd1, n1)**

**conn.commit()**

**conn.close()**

**messagebox.showinfo("Title", "Street Updated")**

**elif temp == 15:**

**x1 = new.get()**

**conn = sqlite3.connect('Newark\_IT.db')**

**c = conn.cursor()**

**n1 = (x1, prim)**

**upd1 = "UPDATE SHIPPING\_ADDRESS SET City = ? where CID = ? "**

**c.execute(upd1, n1)**

**conn.commit()**

**conn.close()**

**messagebox.showinfo("Title", "City Updated")**

**elif temp == 16:**

**x1 = new.get()**

**conn = sqlite3.connect('Newark\_IT.db')**

**c = conn.cursor()**

**n1 = (x1, prim)**

**upd1 = "UPDATE SHIPPING\_ADDRESS SET Country = ? where CID = ? "**

**c.execute(upd1, n1)**

**conn.commit()**

**conn.close()**

**messagebox.showinfo("Title", "Country Updated")**

**elif temp == 17:**

**x1 = new.get()**

**conn = sqlite3.connect('Newark\_IT.db')**

**c = conn.cursor()**

**n1 = (x1, prim)**

**upd1 = "UPDATE SHIPPING\_ADDRESS SET State = ? where CID = ?"**

**c.execute(upd1, n1)**

**conn.commit()**

**conn.close()**

**messagebox.showinfo("Title", "State Updated")**

**elif temp == 18:**

**x1 = new.get()**

**conn = sqlite3.connect('Newark\_IT.db')**

**c = conn.cursor()**

**n1 = (x1, prim)**

**upd1 = "UPDATE SHIPPING\_ADDRESS SET Zip = ? where CID = ?"**

**c.execute(upd1, n1)**

**conn.commit()**

**conn.close()**

**messagebox.showinfo("Title", "Zip Updated")**

**else:**

**pass**

**def validateRegister(username, password, Fname, Lname, email, phone, Cnum, Snum, Oname, Ctype, Expdate,**

**BAddress, Sname, Rname, Sanum, Street, City, State, Country, Zip):**

**username = username.get()**

**password = password.get()**

**Fname = Fname.get()**

**Lname = Lname.get()**

**email = email.get()**

**phone = phone.get()**

**Cnum = Cnum.get()**

**Snum = Snum.get()**

**Oname = Oname.get()**

**Ctype = Ctype.get()**

**Expdate = Expdate.get()**

**BAddress = BAddress.get()**

**Sname = Sname.get()**

**Rname = Rname.get()**

**Sanum = Sanum.get()**

**Street = Street.get()**

**City = City.get()**

**State = State.get()**

**Country = Country.get()**

**Zip = Zip.get()**

**global username1**

**global password1**

**global Fname1**

**global Lname1**

**global email1**

**global phone1**

**global Cnum1**

**global Snum1**

**global Oname1**

**global Ctype1**

**global BAddress1**

**global Sname1**

**global Rname1**

**global Sanum1**

**global Street1**

**global City1**

**global State1**

**global Country1**

**global Zip1**

**username1 = username**

**password1 = password**

**Fname1 = Fname**

**Lname1 = Lname**

**email1 = email**

**phone1 = phone**

**Cnum1 = Cnum**

**Snum1 = Snum**

**Oname1 = Oname**

**Ctype1 = Ctype**

**BAddress1 = BAddress**

**Sname1 = Sname**

**Rname1 = Rname**

**Sanum1 = Sanum**

**Street1 = Street**

**City1 = City**

**State1 = State**

**Country1 = Country**

**Zip1 = Zip**

**if username == "":**

**messagebox.showinfo("Title", "Please Enter username Name")**

**elif password == "":**

**messagebox.showinfo("Title", "Please Enter Password")**

**elif Fname == "":**

**messagebox.showinfo("Title", "Please Enter First Name")**

**elif Lname == "":**

**messagebox.showinfo("Title", "Please Enter Last Name")**

**elif email == "":**

**messagebox.showinfo("Title", "Please Enter Email id")**

**elif phone == 0:**

**messagebox.showinfo("Title", "Please Enter Phone Number")**

**elif Cnum == 0:**

**messagebox.showinfo("Title", "Please Enter Card Number")**

**elif Snum == 0:**

**messagebox.showinfo("Title", "Please Enter Sec Number")**

**elif Oname == "":**

**messagebox.showinfo("Title", "Please Enter Owner Name")**

**elif Ctype == "":**

**messagebox.showinfo("Title", "Please Enter Card Type")**

**elif Expdate == "":**

**messagebox.showinfo("Title", "Please Enter Expiration Date")**

**elif BAddress == "":**

**messagebox.showinfo("Title", "Please Enter Billing Address")**

**elif Sname == "":**

**messagebox.showinfo("Title", "Please Enter Shipping Address Name")**

**elif Rname == "":**

**messagebox.showinfo("Title", "Please Enter Recipient Name")**

**elif Sanum == "":**

**messagebox.showinfo("Title", "Please Enter Shipping Address Number")**

**elif Street == "":**

**messagebox.showinfo("Title", "Please Enter Street")**

**elif City == "":**

**messagebox.showinfo("Title", "Please Enter City")**

**elif State == "":**

**messagebox.showinfo("Title", "Please Enter State")**

**elif Country == "":**

**messagebox.showinfo("Title", "Please Enter Country")**

**elif Zip == "":**

**messagebox.showinfo("Title", "Please Enter Zip Code")**

**else:**

**conn = sqlite3.connect('Newark\_IT.db')**

**c = conn.cursor()**

**d1 = (username, Fname, Lname, email, BAddress, phone, password)**

**save = "insert into CUSTOMER values (?,?,?,?,?,?,?)"**

**c.execute(save, d1)**

**conn.commit()**

**d2 = (Cnum, Snum, Oname, Ctype, BAddress, Expdate, username)**

**save1 = "insert into CREDIT\_CARD values (?,?,?,?,?,?,?)"**

**c.execute(save1, d2)**

**conn.commit()**

**d3 = (username, Sname, Rname, Street, Sanum, City, Zip, State, Country)**

**save2 = "insert into SHIPPING\_ADDRESS values (?,?,?,?,?,?,?,?,?)"**

**c.execute(save2, d3)**

**conn.commit()**

**num = random.randint(1, 100000)**

**d4 = (num, username)**

**save3 = "insert into BASKET values (?,?)"**

**c.execute(save3, d4)**

**conn.commit()**

**conn.close()**

**messagebox.showinfo("Title", "Registered Successfully")**

**screen.destroy()**

**return**

**def RegisterPage():**

**global screen**

**screen = tk.Toplevel(tkWindow)**

**screen.geometry('1000x500')**

**screen.title('Register')**

**# username label and text entry box**

**usernameLabel = Label(screen, text="User Name").grid(row=0, column=0)**

**username = StringVar()**

**usernameEntry = Entry(screen, textvariable=username).grid(row=0, column=1)**

**# password label and password entry box**

**passwordLabel = Label(screen, text="Password").grid(row=1, column=0)**

**password = StringVar()**

**passwordEntry = Entry(screen, textvariable=password, show='\*').grid(row=1, column=1)**

**# first name**

**FnameLabel = Label(screen, text="First Name").grid(row=2, column=0)**

**Fname = StringVar()**

**FnameEntry = Entry(screen, textvariable=Fname).grid(row=2, column=1)**

**# last name**

**LnameLabel = Label(screen, text="Last Name").grid(row=2, column=2)**

**Lname = StringVar()**

**LnameEntry = Entry(screen, textvariable=Lname).grid(row=2, column=3)**

**# email**

**emailLabel = Label(screen, text="email").grid(row=3, column=0)**

**email = StringVar()**

**emailEntry = Entry(screen, textvariable=email).grid(row=3, column=1)**

**# phone**

**phoneLabel = Label(screen, text="phone").grid(row=4, column=0)**

**phone = StringVar()**

**phoneEntry = Entry(screen, textvariable=phone).grid(row=4, column=1)**

**# credit card label and credit card entry box**

**CnumLabel = Label(screen, text="Credit card number").grid(row=5, column=0)**

**Cnum = IntVar()**

**CnumEntry = Entry(screen, textvariable=Cnum).grid(row=5, column=1)**

**SnumLabel = Label(screen, text="sec").grid(row=6, column=0)**

**Snum = IntVar()**

**SnumEntry = Entry(screen, textvariable=Snum).grid(row=6, column=1)**

**OnameLabel = Label(screen, text="Owner Name").grid(row=6, column=2)**

**Oname = StringVar()**

**OnameEntry = Entry(screen, textvariable=Oname).grid(row=6, column=3)**

**CtypeLabel = Label(screen, text="Card type").grid(row=7, column=0)**

**Ctype = StringVar()**

**CtypeEntry = Entry(screen, textvariable=Ctype).grid(row=7, column=1)**

**ExpdateLabel = Label(screen, text="Expiration Date").grid(row=7, column=2)**

**Expdate = StringVar()**

**ExpdateEntry = Entry(screen, textvariable=Expdate).grid(row=7, column=3)**

**BAddress = Label(screen, text="Billing Address").grid(row=8, column=0)**

**BAddress = StringVar()**

**BAddressEntry = Entry(screen, textvariable=BAddress).grid(row=8, column=1)**

**# shipping address label and shipping address entry box**

**SAddress = Label(screen, text="Shipping Address").grid(row=9, column=0)**

**Sname = Label(screen, text="Shipping Address name").grid(row=10, column=0)**

**Sname = StringVar()**

**SnameEntry = Entry(screen, textvariable=Sname).grid(row=10, column=1)**

**RnameLabel = Label(screen, text="Recepient Name").grid(row=10, column=2)**

**Rname = StringVar()**

**RnameEntry = Entry(screen, textvariable=Rname).grid(row=10, column=3)**

**SanumLabel = Label(screen, text="Address number").grid(row=10, column=4)**

**Sanum = IntVar()**

**SanumEntry = Entry(screen, textvariable=Sanum).grid(row=10, column=5)**

**Street = Label(screen, text="Street name").grid(row=11, column=0)**

**Street = StringVar()**

**StreetEntry = Entry(screen, textvariable=Street).grid(row=11, column=1)**

**City = Label(screen, text="City").grid(row=11, column=2)**

**City = StringVar()**

**CityEntry = Entry(screen, textvariable=City).grid(row=11, column=3)**

**State = Label(screen, text="State").grid(row=11, column=4)**

**State = StringVar()**

**StateEntry = Entry(screen, textvariable=State).grid(row=11, column=5)**

**Country = Label(screen, text="Country").grid(row=12, column=0)**

**Country = StringVar()**

**CountryEntry = Entry(screen, textvariable=Country).grid(row=12, column=1)**

**Zip = Label(screen, text="Zip Code").grid(row=12, column=2)**

**Zip = StringVar()**

**ZipEntry = Entry(screen, textvariable=Zip).grid(row=12, column=3)**

**validateregister = partial(validateRegister, username, password, Fname, Lname, email, phone, Cnum, Snum, Oname,**

**Ctype, Expdate, BAddress, Sname, Rname, Sanum, Street, City, State, Country,**

**Zip)**

**register\_button = Button(screen, text="Register", command=validateregister).grid(row=13, column=3)**

**def main\_screen():**

**# window**

**conn = sqlite3.connect('Newark\_IT.db')**

**c = conn.cursor()**

**c.execute("PRAGMA foreign\_keys = ON")**

**conn.commit()**

**conn.close()**

**global tkWindow**

**tkWindow = Tk()**

**tkWindow.geometry('400x150')**

**tkWindow.title('Login')**

**# username label and text entry box**

**usernameLabel = Label(tkWindow, text="User Name").grid(row=0, column=0)**

**username = StringVar()**

**usernameEntry = Entry(tkWindow, textvariable=username).grid(row=0, column=1)**

**# password label and password entry box**

**passwordLabel = Label(tkWindow, text="Password").grid(row=1, column=0)**

**password = StringVar()**

**passwordEntry = Entry(tkWindow, textvariable=password, show='\*').grid(row=1, column=1)**

**validatelogin = partial(validateLogin, username, password)**

**# login button**

**login\_button = Button(tkWindow, text="Login", command=validatelogin).grid(row=4, column=0)**

**# Register button**

**register\_button = Button(tkWindow, text="Register", command=RegisterPage).grid(row=4, column=1)**

**tkWindow.mainloop()**

**main\_screen()**

* **SCREENSHOTS:**

1. **Login Page**

Graphical user interface, text, application, chat or text message

Description automatically generated

1. **Registration page**

Graphical user interface

Description automatically generated

1. **Main-screen**

Graphical user interface, text, application, chat or text message

Description automatically generated

1. **Update User Information**

A screenshot of a video game

Description automatically generated with medium confidence

1. **Online Sale**

Graphical user interface, application, website

Description automatically generated

1. **Sale Statistics**

A screenshot of a video game

Description automatically generated

1. **User Information Update**

